User's Manual

16

R9A02G015

User's Manual: Hardware

ASSP (USB Power Delivery Controller)

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

Renesas Electronics
www.renesas.com

Rev.1.00 Mar 2019

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 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

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.

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the R9A02G015 and design and develop application systems and programs for these devices. The target products are as follows.

- 32-pin: R9A02G0150/R9A02G0151

Purpose

This manual is intended to give users an understanding of the functions described in the **Organization** below.

Organization

The R9A02G015 manual is separated into three parts: this manual, Data Sheet, and the software edition (common to the RL78 family).

R9A02G015
User's Manual: Hardware
(This Manual)

- Pin functions
- Internal block functions
- Interrupts
- Other on-chip peripheral functions

RL78 Family
User's Manual: Software

- CPU functions
- Instruction set
- Explanation of each instruction

R9A02G015
Data Sheet

- Electrical specifications
- Package drawings

How to Read This Manual

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler.
- To know details of the R9A02G015 (built in RL78 CPU core) instructions:
  → Refer to the separate document **RL78 Family User's Manual Software (R01US0015E)**.

Conventions	Data significance:	Higher digits on the left and lower digits on the right
	Active low representations:	$\overline{\times\times\times}$ (overscore over pin and signal name) or xxxB
	Note:	Footnote for item marked with Note in the text
	Caution:	Information requiring particular attention
	Remark:	Supplementary information
	Numerical representations:	Binary.....×××× or ××××B
		Decimal.....××××
		Hexadecimal .....××××H

Related Documents The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Documents Related to Devices

Document Name	Document No.
R9A02G015 User's Manual: Hardware	This manual
R9A02G015 Data Sheet	R19DS0101E
RL78 Family User's Manual: Software	R01US0015E

Documents Related to Flash Memory Programming (User's Manual)

Document Name	Document No.
PG-FP6 Flash Memory Programmer User's Manual	R20UT4025E
E1 Emulator R0E000010KCE00 E20 Emulator R0E000200KCT00 User's Manual	R20UT0398E
E2 Emulator RTE0T00020KCE00000R User's Manual	R20UT3538E
E2 Emulator Lite RTE0T0002LKCE00000R User's Manual	R20UT3240E
Renesas Flash Programmer Flash Memory Programming Software User's Manual	R20UT4066E
Renesas Flash Development Toolkit User's Manual	R20UT0508E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

Other Documents

Document Name	Document No.
Renesas Microcontrollers RL78 Family	R01CP0003E
Semiconductor Package Mount Manual	R50ZZ0003E
Semiconductor Reliability Handbook	R51ZZ0001E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

All trademarks and registered trademarks are the property of their respective owners.

EEPROM is a trademark of Renesas Electronics Corporation.

SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

Ultra-low power consumption technology

- V_{DD} = single power supply voltage of 2.7 to 5.5 V
- HALT mode
- STOP mode
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: 0.04167 μs: @ 24 MHz operation with high-speed on-chip oscillator
- Multiply/divide/multiply & accumulate instructions are supported.
- Address space: 1 MB
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 7 KB

Code flash memory

- Code flash memory: 128 KB
- Block size: 1 KB
- Prohibition of block erase and rewriting (security function)
- On-chip debug function
- Self-programming (with boot swap function/flash shield window function)

Data flash memory

- Data flash memory: 2 KB
- Back ground operation (BGO): Instructions can be executed from the program memory while rewriting the data flash memory.
- Number of rewrites: 1,000,000 times (TYP.)
- Voltage of rewrites: V_{DD} = 2.7 to 5.5 V

High-speed on-chip oscillator

- Select from 48 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, and 1 MHz
- High accuracy: ±1.0% (V_{DD} = 2.7 to 5.5 V, T_A = -20 to +85°C)

Operating ambient temperature

- T_A = -40 to +85°C (A: Consumer applications)

Power management and reset function

- On-chip power-on-reset (POR) circuit
- On-chip voltage detector (LVD) (Select reset from 6 levels)

USB

- Complying with USB Specification Revision 2.0, incorporating host/function controller
- Corresponding to full-speed transfer (12 Mbps) and low-speed (1.5 Mbps)
- Complying with Battery Charging Specification Revision 1.2
- Compliant with the 2.1 A/1.0 A charging mode.

Serial interfaces

- CSI: 2 channels
- UART: 1 channel
- Simplified I^2C: 2 channels
- I^2C: 2 or 3 channels

Timers

- 16-bit timer: 8 channels
- 12-bit interval timer: 1 channel
- Watchdog timer: 1 channel

A/D converter

- 8/10-bit resolution A/D converter (V_{DD} = 2.7 to 5.5 V)
- Analog input: 8 channels
- Internal reference voltage (1.45 V) and temperature sensor

I/O ports

- I/O port: 23 or 28 (N-ch open drain I/O [withstand voltage of 6 V]: 5, N-ch open drain I/O [V_{DD} withstand voltage]: 8 or 13)
- Can be set to N-ch open drain, TTL input buffer, and on-chip pull-up resistor
- On-chip clock output/buzzer output controller

Others

- On-chip BCD (binary-coded decimal) correction circuit

Remark The functions mounted depend on the product. See **1.6 Outline of Functions**.

○ ROM, RAM capacities

Flash ROM	Data flash	RAM	R9A02G015	
			32 pins (with USB)	32 pins (without USB)
128 KB	2 KB	7 KB Note	R9A02G0150	R9A02G0151

Note The flash library uses RAM in self-programming and rewriting of the data flash memory.
The target products and start address of the RAM areas used by the flash library are shown below.

R9A02G0150/R9A02G0151: Start address FE300H

1.2 Ordering Information

Figure 1 - 1 Part Number and Package of R9A02G015

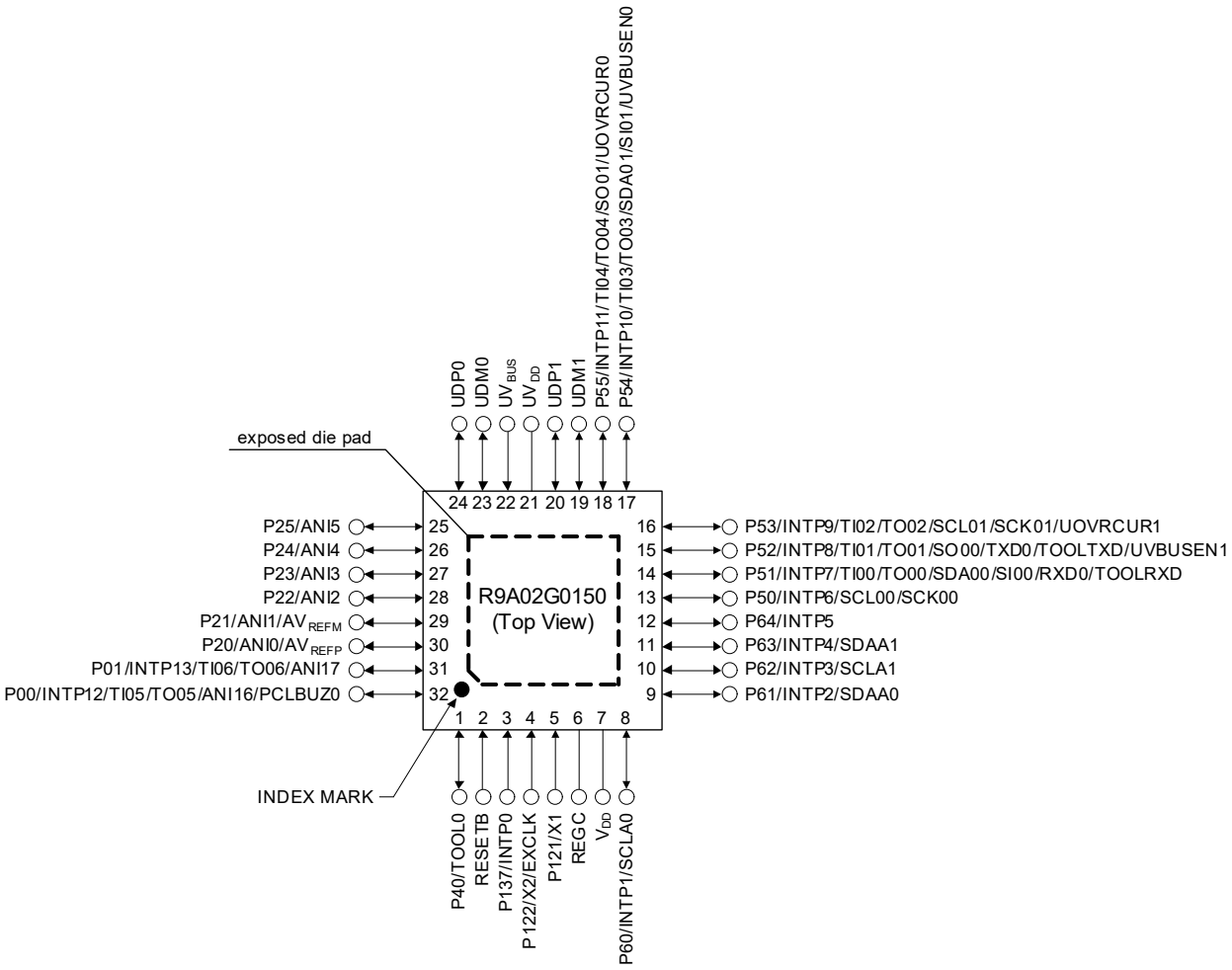
Pin count	Package	Ordering Part Number	Remarks
32 pins	32-pin QFN (4 × 4 mm, 0.4 mm pitch)	R9A02G015020GNP#AC0	Product with USB (R9A02G0150)
		R9A02G015120GNP#AC0	Product without USB (R9A02G0151)

Caution The ordering part numbers represent the numbers at the time of publication. For the latest ordering part numbers, refer to the target product page of the Renesas Electronics website.

1.3 Pin Configuration (Top View)

1.3.1 32-pin product (with USB)

• 32-pin QFN (4 × 4 mm, 0.4 mm pitch)



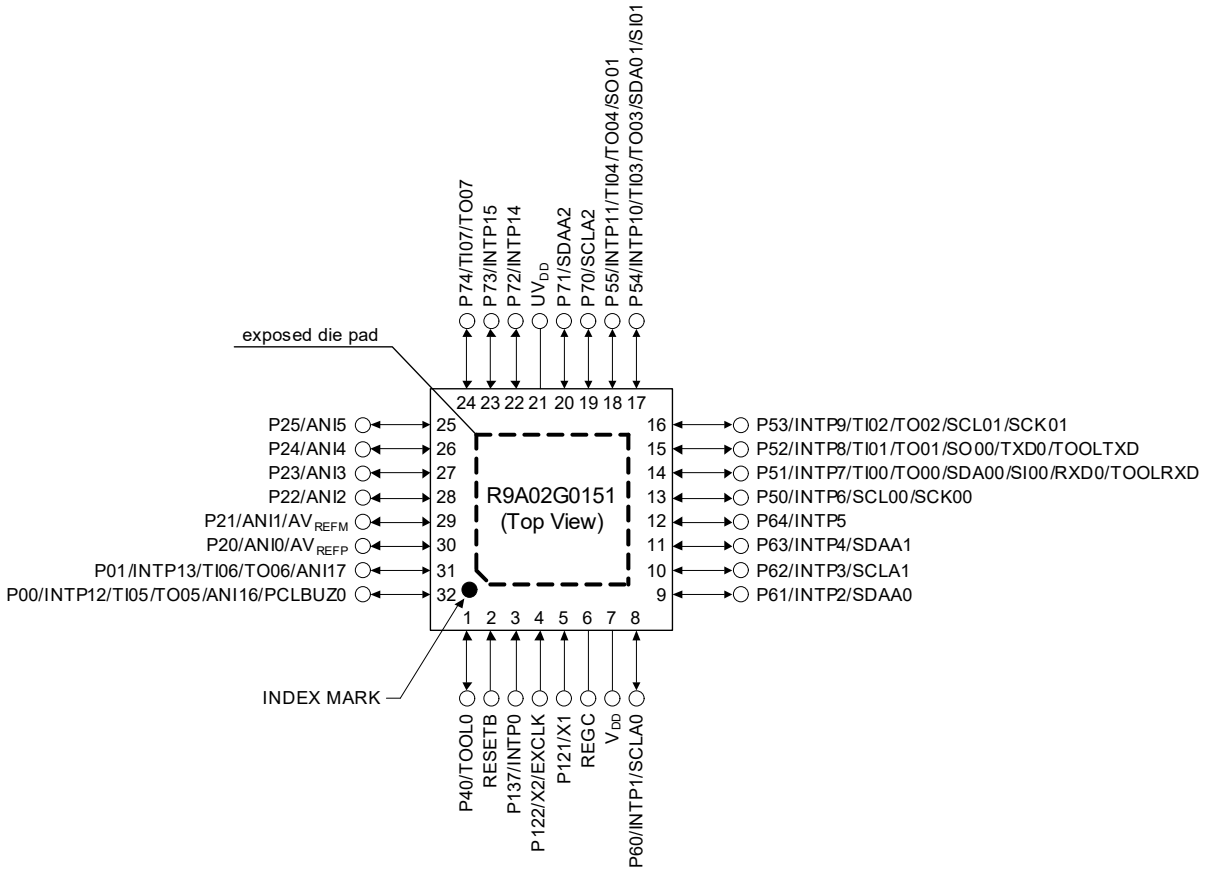
Caution 1. Connect the exposed die pad (Vss) to ground.

Caution 2. Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).

Remark For pin identification, see **1.4 Pin Identification**.

1.3.2 32-pin product (without USB)

• 32-pin QFN (4 × 4 mm, 0.4 mm pitch)



Caution 1. Connect the exposed die pad (Vss) to ground.

Caution 2. Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).

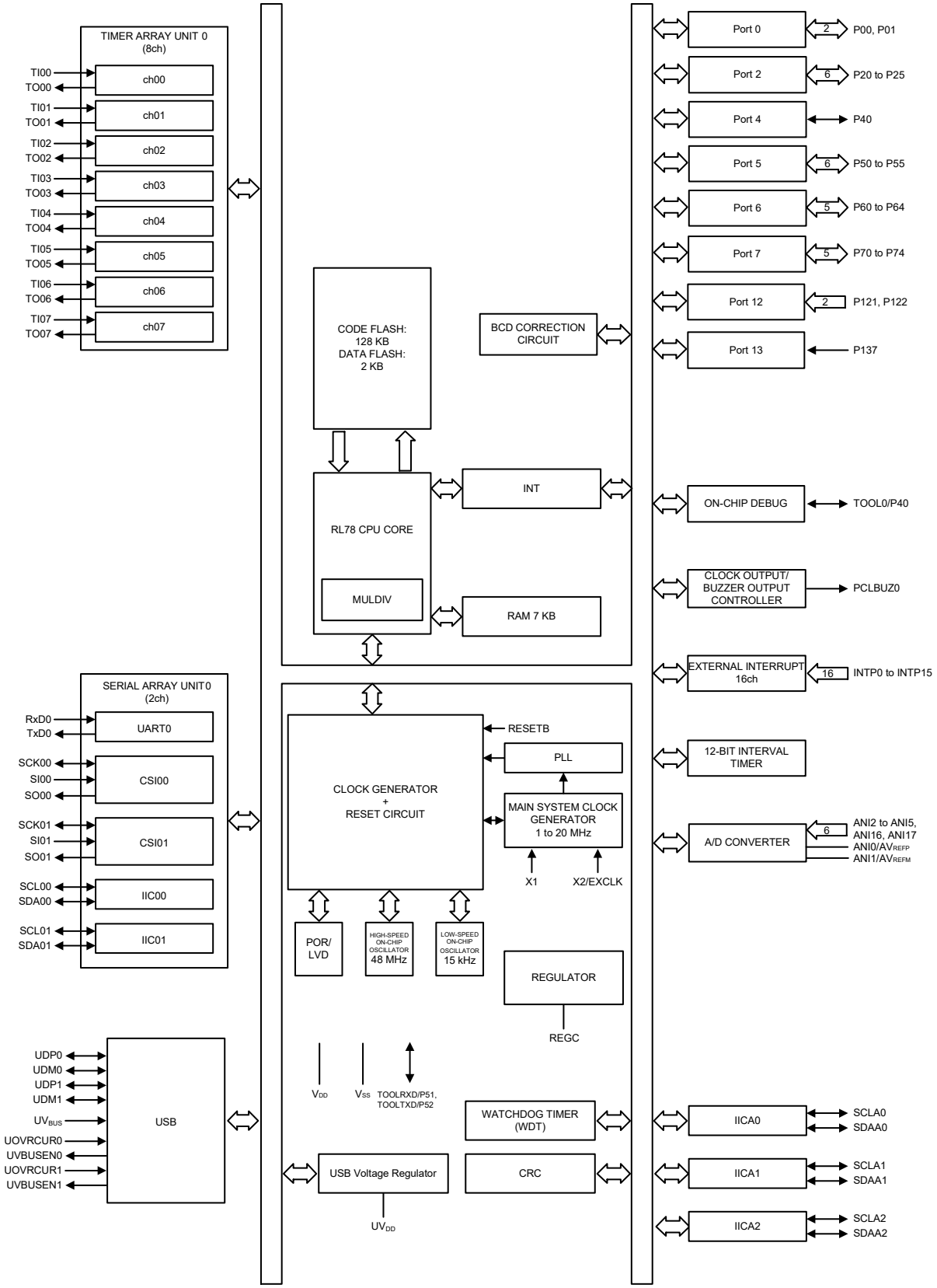
Remark For pin identification, see **1.4 Pin Identification**.

1.4 Pin Identification

ANI0 to ANI5, ANI16, ANI17:	Analog input
AVREFM:	A/D converter reference potential (- side) input
AVREFP:	A/D converter reference potential (+ side) input
EXCLK:	External clock input (main system clock)
INTP0 to INTP15:	External interrupt input
P00, P01:	Port 0
P20 to P25:	Port 2
P40:	Port 4
P50 to P55:	Port 5
P60 to P64:	Port 6
P70 to P74:	Port 7
P121, P122:	Port 12
P137:	Port 13
PCLBUZ0:	Programmable clock output/buzzer output
REGC:	Regulator capacitance
RESETB:	Reset
RxD0:	Receive data
SCK00, SCK01:	Serial clock input/output
SCLA0 to SCLA2, SCL00, SCL01:	Serial clock input/output
SDAA0 to SDAA2, SDA00, SDA01:	Serial data input/output
SI00, SI01:	Serial data input
SO00, SO01:	Serial data output
TI00 to TI07:	Timer input
TO00 to TO07:	Timer output
TOOL0:	Data input/output for tool
TOOLRXD, TOOLTXD:	Data input/output for external device
TxD0:	Transmit data
UDM0, UDM1, UDP0, UDP1:	USB Input/Output
UOVRCUR0, UOVRCUR1:	USB Input
UVBUSEN0, UVBUSEN1:	USB Output
UVDD:	USB Power Supply/USB Regulator Capacitance
UVBUS:	USB Input/USB Power Supply (USB Optional BC)
VDD:	Power supply
VSS:	Ground
X1, X2:	Crystal oscillator (main system clock)

1.5 Block Diagram

1.5.1 32-pin products



1.6 Outline of Functions

(1/2)

Item		32-pin (with USB) R9A02G0150	32-pin (without USB) R9A02G0151
Code flash memory (KB)		128 KB	
Data flash memory (KB)		2 KB	
RAM		7KB Note 1	
Address space		1 MB	
Main system clock	High-speed system clock (f_{MX})	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) HS (High-speed main) mode: 1 to 20 MHz (V_{DD} = 2.7 to 5.5 V)	
	High-speed on-chip oscillator clock (f_{IH}) Max: 24 MHz	HS (High-speed main) mode: 1 to 24 MHz (V_{DD} = 2.7 to 5.5 V)	
	PLL clock	6, 12, 24 MHz Note 2: V_{DD} = 2.7 to 5.5 V	
Subsystem clock	Low-speed on-chip oscillator clock (f_{IL})	15 kHz (TYP.): V_{DD} = 2.7 to 5.5 V	
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)	
Minimum instruction execution time		0.04167 μs (High-speed on-chip oscillator clock: f_{HOCO} = 48MHz/f_{IH} = 24 MHz operation)	
		0.04167 μs (PLL clock: f_{PLL} = 48 MHz/f_{IH} = 24 MHz Note 2 operation)	
		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)	
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits, 16 bits × 16 bits), Division (16 bits ÷ 16 bits, 32 bits ÷ 32 bits) • Multiplication and Accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (Set, reset, test, and Boolean operation), etc.	
I/O port	Total	23	28
	CMOS I/O	15	20
	CMOS input	3	
	N-ch open-drain I/O (6 V tolerance)	5	
Timer	16-bit timer	8 channels	
	Watchdog timer	1 channel	
	12-bit interval timer	1 channel	
	Timer output	7	8
Clock output/buzzer output		1	
		2.93 kHz, 5.86 kHz, 11.7 kHz, 1.5 MHz, 3 MHz, 6 MHz (Main system clock: f_{MAIN} = 24 MHz operation)	
10-bit resolution A/D converter		8 channels	
Serial interface		CSI: 2 channels/UART: 1 channel/simplified I^2C: 2 channels	
	I^2C bus	2 channels	3 channels
USB	Host controller	2 channels	—
	Function controller	1channel	—
Vectored interrupt sources	Internal	22	21
	External	14	16
Reset		• Reset by RESETB pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detector • Internal reset by illegal instruction execution Note 3 • Internal reset by RAM parity error • Internal reset by illegal-memory access	
Power-on-reset circuit		• Power-on-reset: 1.51 ± 0.04 V (T_A = -40 to +85°C) • Power-down-reset: 1.50 ± 0.04 V (T_A = -40 to +85°C)	
Voltage detector	Power on	2.81 V to 4.06 V (6 stages)	
	Power down	2.75 V to 3.98 V (6 stages)	
On-chip debug function		Provided (Enable to tracing)	

(2/2)

Item	32-pin (with USB)	32-pin (without USB)
	R9A02G0150	R9A02G0151
Power supply voltage	V_{DD} = 2.7 to 5.5 V	
Operating ambient temperature	T_A = -40 to +85°C	

Note 1. The flash library uses RAM in self-programming and rewriting of the data flash memory.
The target products and start address of the RAM areas used by the flash library are shown below.

R9A02G0150/R9A02G0151: Start address FE300H

Note 2. In the PLL clock 48 MHz operation, the system clock is 2/4/8 dividing ratio.

Note 3. The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution is not issued by emulation with the in-circuit emulator or on-chip debug emulator.

CHAPTER 2 PIN FUNCTIONS

2.1 Port Functions

Pin I/O buffer power supplies are shown below.

Table 2 - 1 Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
V_{DD}	All pins other than UDP0, UDM0, UDP1, and UDM1
UV_{DD}	UDP0, UDM0, UDP1, UDM1

Set in each port I/O, buffer, pull-up resistor is also valid for alternate functions.

2.1.1 32-pin Products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	8-3-4	I/O	Analog input port	INTP12/TI05/TO05/ANI16/PCLBUZ0	Port 0. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P00 and P01 can be set to TTL input buffer. Output of P00 and P01 can be set to N-ch open-drain output (V_{DD} tolerance). P00 and P01 can be set to analog input. Note 1
P01				INTP13/TI06/TO06/ANI17	
P20	4-3-5	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 5-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input. Note 1
P21				ANI1/AV_{REFM}	
P22				ANI2	
P23				ANI3	
P24				ANI4	
P25				ANI5	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.
P50	8-1-4	I/O	Input port	INTP6/SCL00/SCK00	Port 5. 6-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P50 to P55 can be set to TTL input buffer. Output of P50 to P55 can be set to N-ch open-drain output (V_{DD} tolerance).
P51				INTP7/TI00/TO00/SDA00/SI00/ RXD0/TOOLRXD	
P52				INTP8/TI01/TO01/SO00/TXD0/ TOOLTXD/UVBUSEN1 Note 2	
P53				INTP9/TI02/TO02/SCL01/SCK01/ UOVRCUR1 Note 2	
P54				INTP10/TI03/TO03/SDA01/SI01/ UVBUSEN0 Note 2	
P55				INTP11/TI04/TO04/SO01/ UOVRCUR0 Note 2	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P60	12-1-3	I/O	Input port	INTP1/SCLA0	Port 6. 5-bit I/O port. Input/output can be specified in 1-bit units. Output can be set to N-ch open-drain output (6 V tolerance).
P61				INTP2/SDAA0	
P62				INTP3/SCLA1	
P63				INTP4/SDAA1	
P64	12-1-2			INTP5	
P70 Note 3	8-1-4	I/O	Input port	SCLA2	Port 7. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P70 to P74 can be set to TTL input buffer. Output of P70 to P74 can be set to N-ch open-drain output (V_{DD} tolerance).
P71 Note 3				SDAA2	
P72 Note 3				INTP14	
P73 Note 3				INTP15	
P74 Note 3				TI07/TO07	
P121	2-2-1	Input	Input port	X1	Port 12. 2-bit input-only port.
P122				X2/EXCLK	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input-only port.
RESETB	2-1-1	Input	—	—	Input-only pin for external reset. Connect to V_{DD} directly or via a resistor when external reset is not used.
UDP0 Note 2	18-11-1	I/O	—	—	D+ I/O pin of USB port 0. This pin should be connected to the D+ pin of the USB bus.
UDM0 Note 2	18-11-1	I/O	—	—	D- I/O pin of USB port 0. This pin should be connected to the D- pin of the USB bus.
UV_{BUS} Note 2	17-11-1	Input	—	—	USB cable connection monitor pin. This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected during operation as a function controller.
UDP1 Note 2	18-11-1	I/O	—	—	D+ I/O pin of USB port 1. This pin should be connected to the D+ pin of the USB bus.
UDM1 Note 2	18-11-1	I/O	—	—	D- I/O pin of USB port 1. This pin should be connected to the D- pin of the USB bus.

Note 1. Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).

Note 2. This function has not been mounted in the R9A02G0151 product.

Note 3. This function has not been mounted in the R9A02G0150 product.

2.2 Functions other than port pins

(1/2)

Function Name	I/O	Function	Product with USB	Product without USB
			32-pin	32-pin
ANI0	Input	A/D converter analog input	✓	✓
ANI1			✓	✓
ANI2			✓	✓
ANI3			✓	✓
ANI4			✓	✓
ANI5			✓	✓
ANI16			✓	✓
ANI17			✓	✓
EXCLK	Input	External clock input for main system clock	✓	✓
INTP0	Input	External interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.	✓	✓
INTP1			✓	✓
INTP2			✓	✓
INTP3			✓	✓
INTP4			✓	✓
INTP5			✓	✓
INTP6			✓	✓
INTP8			✓	✓
INTP9			✓	✓
INTP10			✓	✓
INTP11			✓	✓
INTP12			✓	✓
INTP13			✓	✓
INTP14			—	✓
INTP15			—	✓
PCLBUZ0	Output	Clock output/buzzer output	✓	✓
REGC	—	Pin for connecting regulator output stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.	✓	✓
RESETB	Input	This is the active-low system reset input pin. When the external reset pin is not used, connect this pin directly or via a resistor to V_{DD}.	✓	✓
RxD0	Input	Serial data input pins of serial interface UART0	✓	✓
TxD0	Output	Serial data output pins of serial interface UART0	✓	✓
SCK00	I/O	Clock input/output for CSI00, CSI01	✓	✓
SCK01			✓	✓
SCLA0	I/O	Clock input/output for IICA0, IICA1, IICA2	✓	✓
SCLA1			✓	✓
SCLA2			—	✓
SCL00	Output	Clock output for IIC00, IIC01	✓	✓
SCL01			✓	✓
SDAA0	I/O	Serial data I/O for IICA0, IICA1, IICA2	✓	✓
SDAA1			✓	✓
SDAA2			—	✓
SDA00	I/O	Serial data I/O for IIC00, IIC01	✓	✓
SDA01			✓	✓

(2/2)

Function Name	I/O	Function	Product with USB	Product without USB
			32-pin	32-pin
SI00	Input	Serial data input to CSI00, CSI01	✓	✓
SI01			✓	✓
SO00	Output	Serial data output from CSI00, CSI01	✓	✓
SO01			✓	✓
TI00	Input	The pins for inputting an external count clock/capture trigger to 16-bit timers 00 to 07	✓	✓
TI01			✓	✓
TI02			✓	✓
TI03			✓	✓
TI04			✓	✓
TI05			✓	✓
TI06			✓	✓
TI07			—	✓
TO00	Output	Timer output pins of 16-bit timers 00 to 07	✓	✓
TO01			✓	✓
TO02			✓	✓
TO03			✓	✓
TO04			✓	✓
TO05			✓	✓
TO06			✓	✓
TO07			—	✓
X1	—	Resonator connection for main system clock	✓	✓
X2	—		✓	✓
V_{DD}	—	Positive power supply for pins other than USB data pins	✓	✓
AV_{REFP}	Input	A/D converter reference potential (+ side) input	✓	✓
AV_{REFM}	Input	A/D converter reference potential (- side) input	✓	✓
V_{SS}	—	Ground potential for all pins	✓	✓
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programming	✓	✓
TOOLTxD	Output	UART transmission pin for the external device connection used during flash memory programming	✓	✓
TOOL0	I/O	Data I/O for flash memory programmer/debugger	✓	✓
UV_{DD}	—	Power supply for USB	✓	✓
UV_{BUS}	Input	VBUS input	✓	—
UDP0	I/O	USB data input/output (+ side) (USB port 0)	✓	—
UDM0	I/O	USB data input/output (- side) (USB port 0)	✓	—
UDP1	I/O	USB data input/output (+ side) (USB port 1)	✓	—
UDM1	I/O	USB data input/output (- side) (USB port 1)	✓	—
UVBUSEN0	Output	VBUS supply permit output (for USB port 0)	✓	—
UOVRCUR0	Input	Overcurrent detection input (for USB port 0)	✓	—
UVBUSEN1	Output	VBUS supply permit output (for USB port 1)	✓	—
UOVRCUR1	Input	Overcurrent detection input (for USB port 1)	✓	—

Caution After reset release, the relationships between P40/TOOL0 and the operating mode are as follows.

Table 2 - 2 Relationships Between P40/TOOL0 and Operation Mode After Reset Release

P40/TOOL0	Operating mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

For details, see 22.4 Programming Method.

Remark Use bypass capacitors (about 0.1 μF) as noise and latch up countermeasures with relatively thick wires at the shortest distance to V_{DD} to V_{SS} lines.

2.3 Connection of Unused Pins

Table 2 - 3 shows the Connection of Unused Pins.

Remark The mounted pins depend on the product. Refer to **1.3 Pin Configuration (Top View)** and **2.1 Port Functions**.

Table 2 - 3 Connection of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P00, P01	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P20 to P25	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P40/TOOL0	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P50 to P55	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P60 to P64	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Set the port's output latch to 0 and leave the pins open, or set the port's output latch to 1 and independently connect the pins to V_{DD} or V_{SS} via a resistor.
P70 to P74	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121, P122	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P137	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
RESETB	Input	Connect directly or via a resistor to V_{DD}.
REGC	—	Connect to V_{SS} via capacitor (0.47 to 1 μF).
UV_{DD}	—	• When the USB power supply is not used, connect directly to V_{DD}, or draw power from an external 3.3 V supply. • When the USB power is generated in the chip, connect to V_{SS} via capacitor (0.33 μF).
UV_{BUS} Note	Input	Independently connect to V_{SS} via a resistor.
UDM0 Note, UDP0 Note	I/O	Input: Independently connect to UV_{DD} or V_{SS} via a resistor. Output: Leave open.
UDM1 Note, UDP1 Note	I/O	

Note This function is not mounted in the R9A02G0151 product.

2.4 Pin Block Diagrams

For the pin types listed in 2.1.1 32-pin Products, pin block diagrams are shown in Figures 2 - 1 to 2 - 11.

Figure 2 - 1 Pin Block Diagram of Pin Type 2-1-1

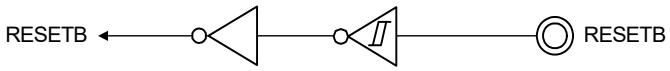
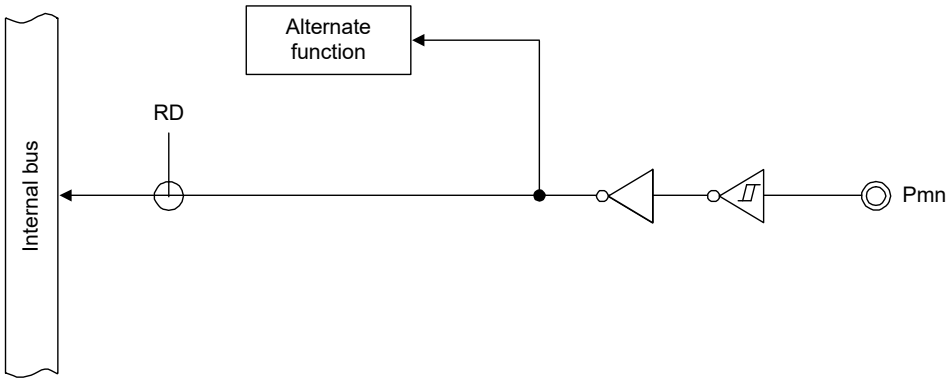
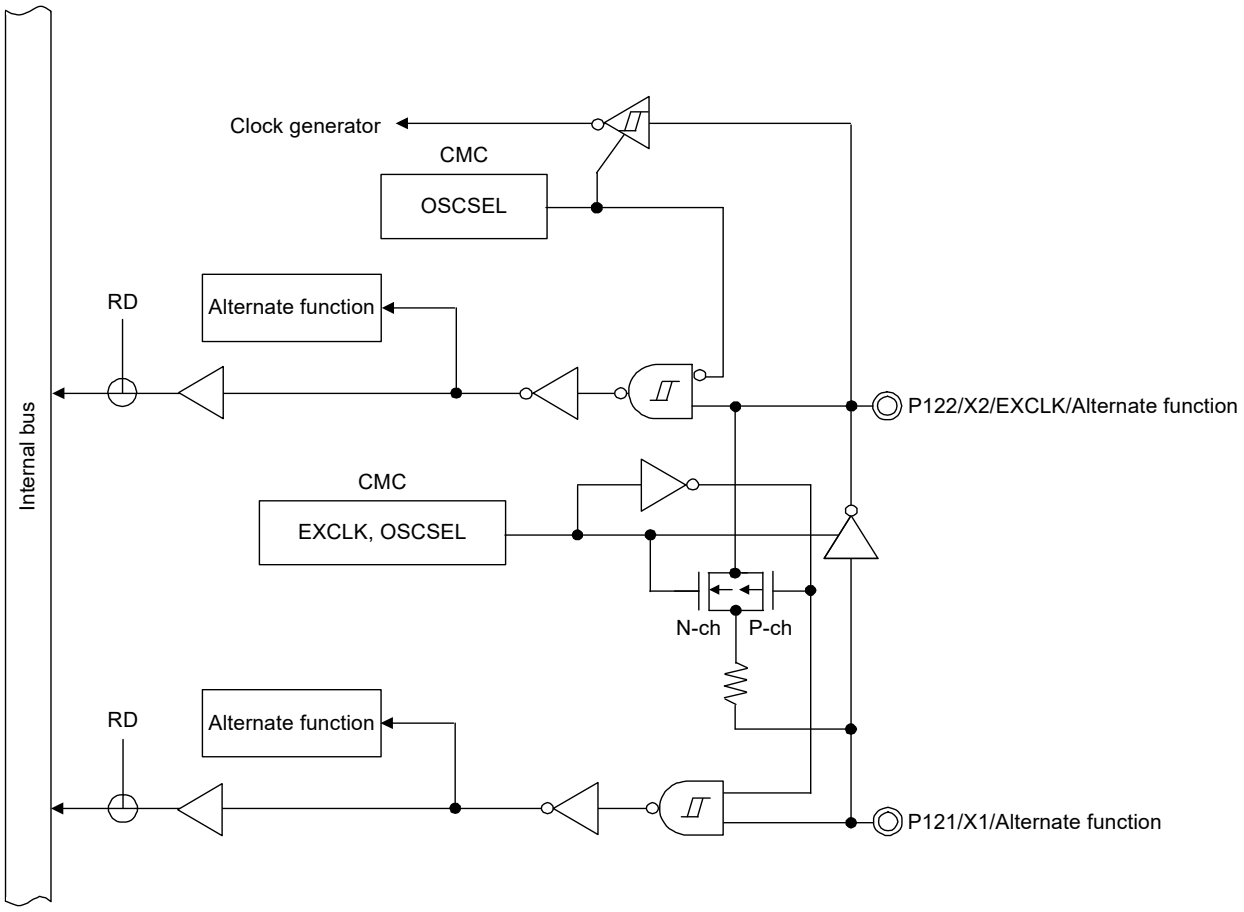


Figure 2 - 2 Pin Block Diagram of Pin Type 2-1-2



Remark Refer to **2.1 Port Functions** for alternate functions.

Figure 2 - 3 Pin Block Diagram of Pin Type 2-2-1



Remark Refer to **2.1 Port Functions** for alternate functions.

Figure 2 - 4 Pin Block Diagram of Pin Type 4-3-5

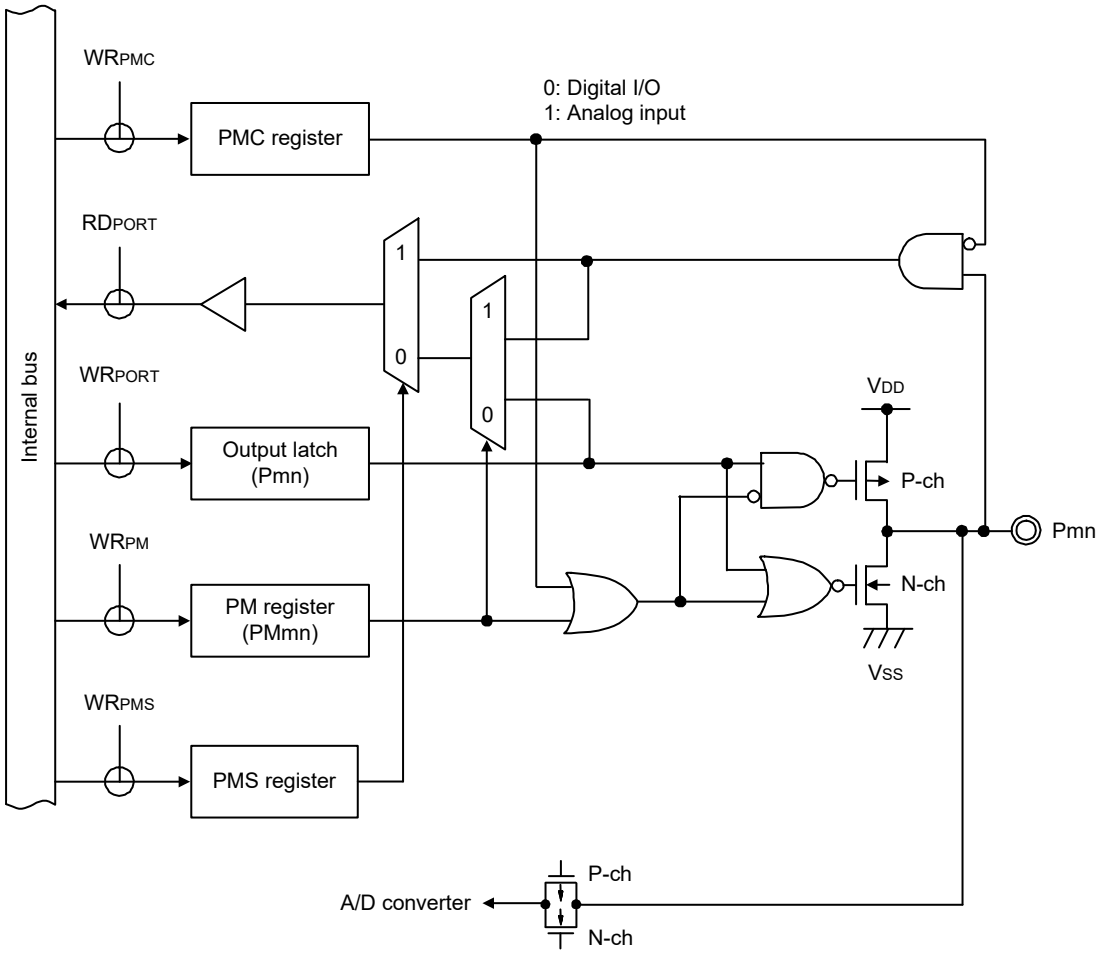
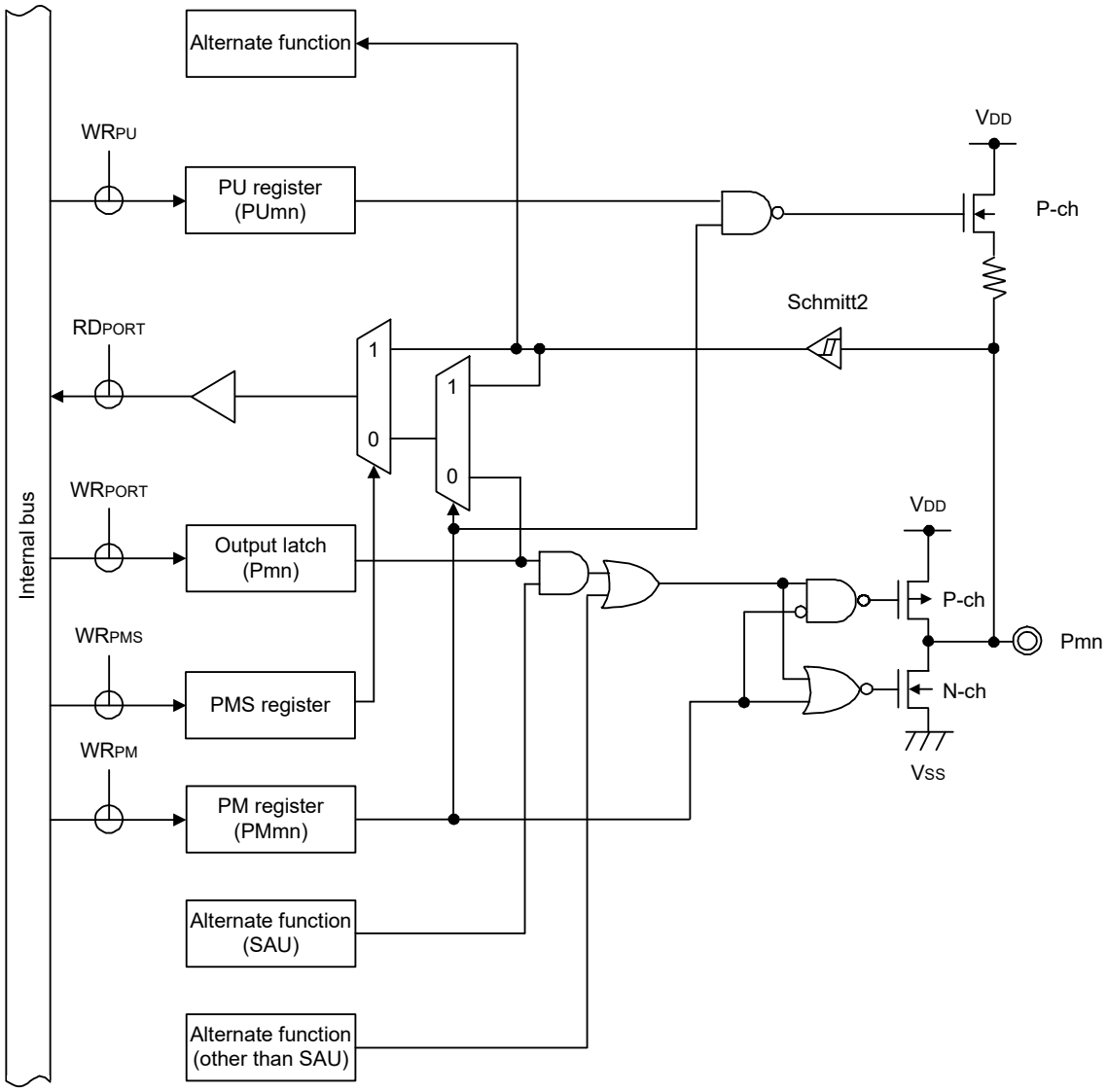
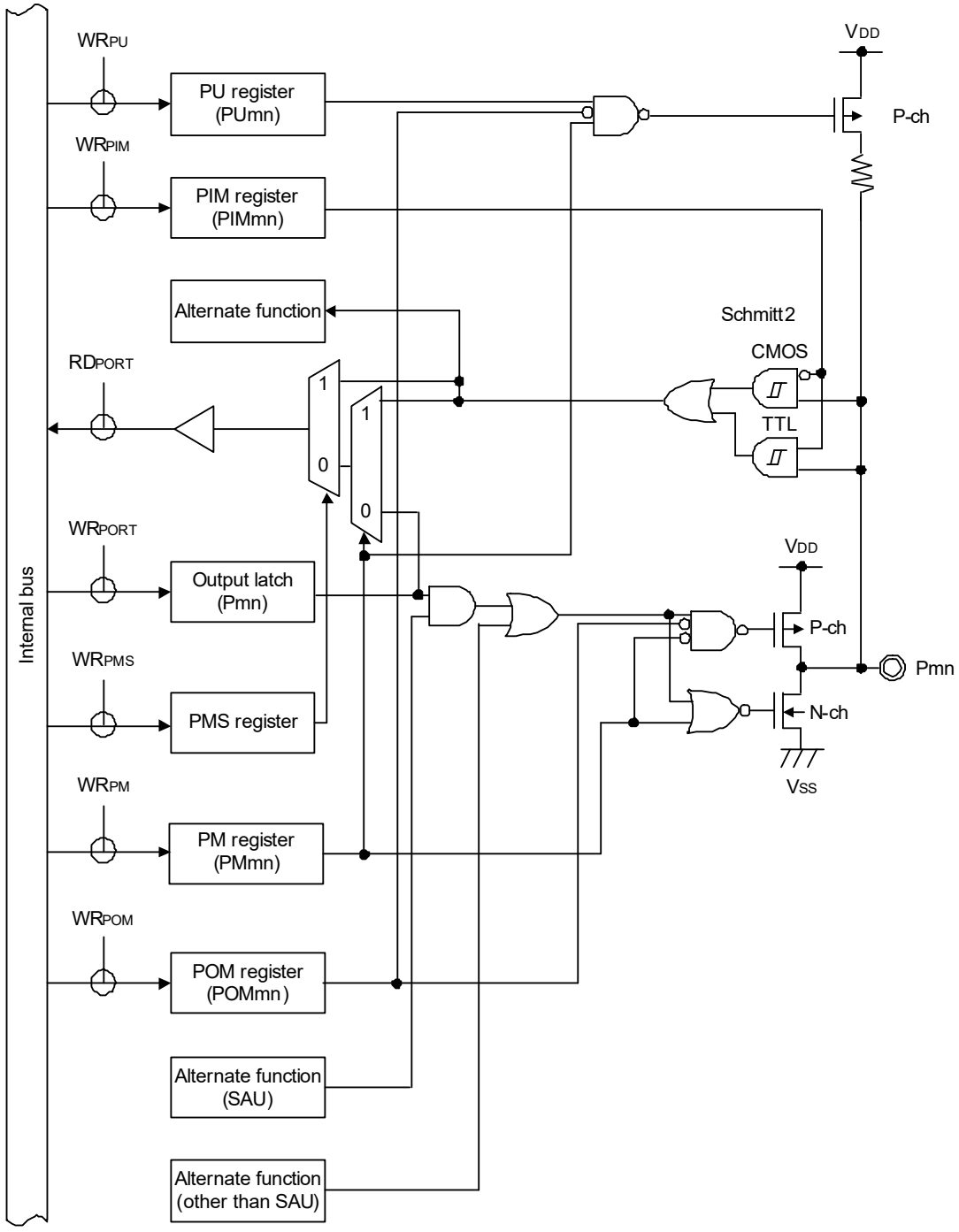


Figure 2 - 5 Pin Block Diagram of Pin Type 7-1-3



Remark 1. Refer to **2.1 Port Functions** for alternate functions.
Remark 2. SAU: Serial array unit

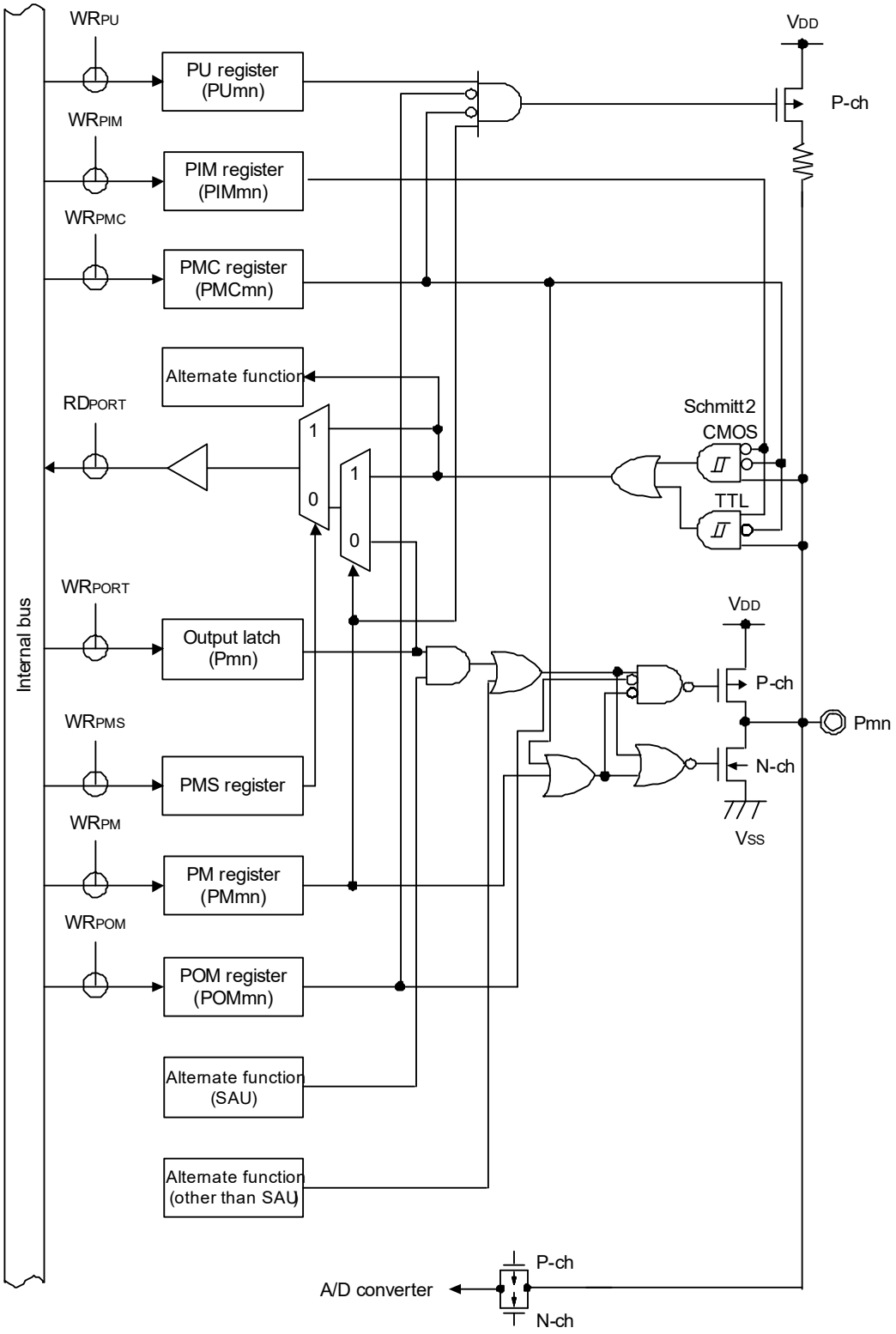
Figure 2 - 6 Pin Block Diagram of Pin Type 8-1-4



Remark 1. Refer to **2.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

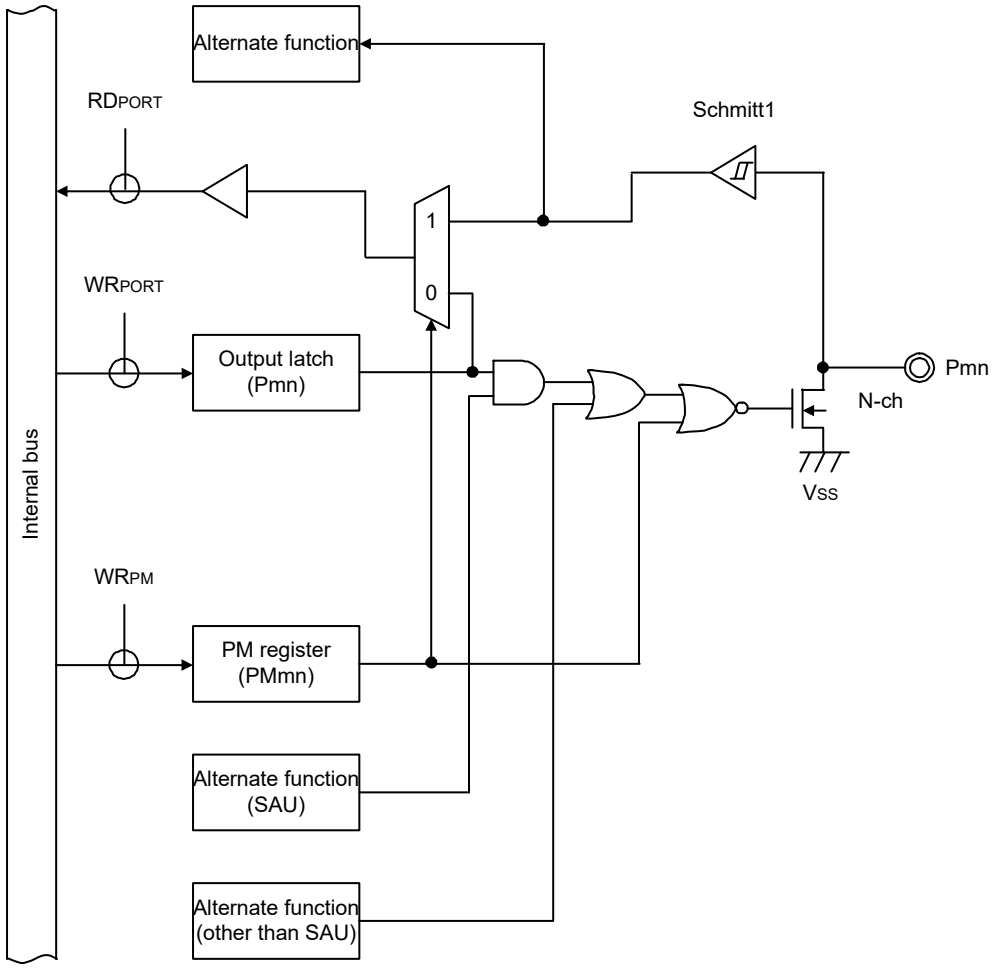
Figure 2 - 7 Pin Block Diagram of Pin Type 8-3-4



Remark 1. Refer to **2.1 Port Functions** for alternate functions.

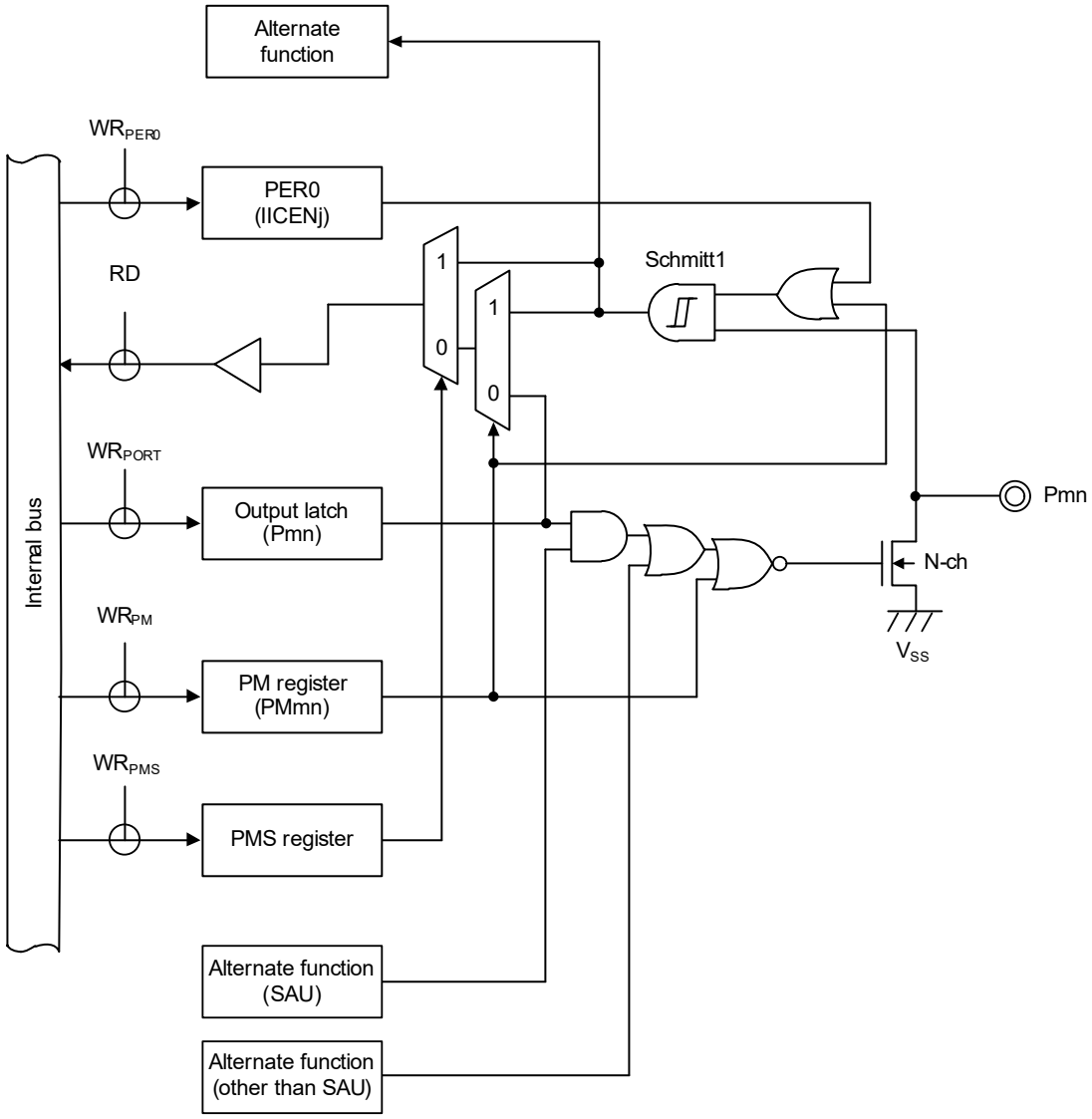
Remark 2. SAU: Serial array unit

Figure 2 - 8 Pin Block Diagram of Pin Type 12-1-2



Remark 1. Refer to **2.1 Port Functions** for alternate functions.
Remark 2. SAU: Serial array unit

Figure 2 - 9 Pin Block Diagram of Pin Type 12-1-3



Remark 1. Refer to **2.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

Figure 2 - 10 Pin Block Diagram of Pin Type 17-11-1

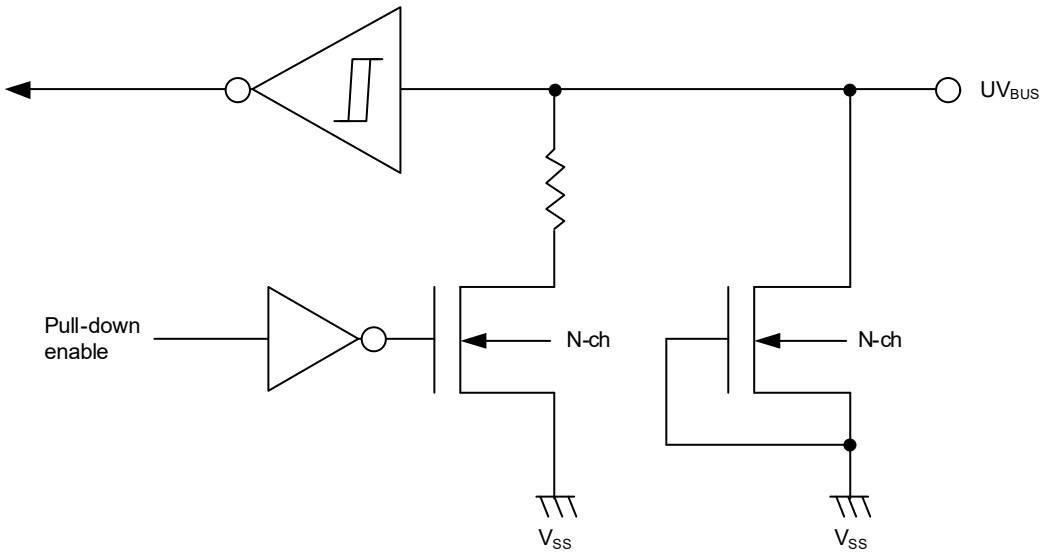
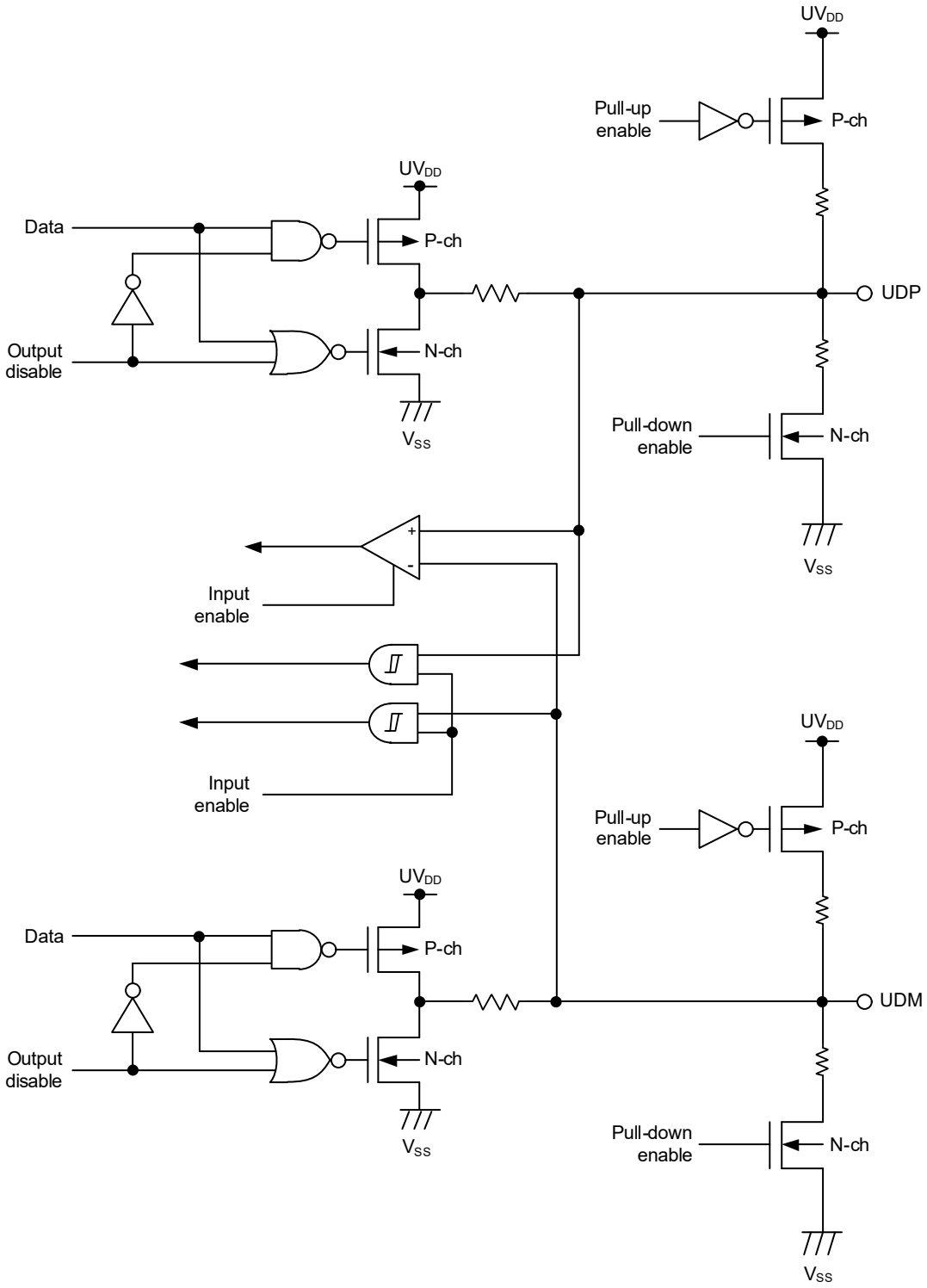


Figure 2 - 11 Pin Block Diagram of Pin Type 18-11-1



CHAPTER 3 CPU ARCHITECTURE

The R9A02G015 has the RL78-S3 CPU core.
The CPU core in the RL78-S3 employs the Harvard architecture which has independent instruction fetch bus, address bus and data bus. In addition, through the adoption of three-stage pipeline control of fetch, decode, and memory access, the operation efficiency is remarkably improved over the conventional CPU core. The CPU core features high performance and highly functional instruction processing, and can be suited for use in various applications that require high speed and highly functional processing.

- 3-stage pipeline CISC architecture
- Address space: 1 Mbyte
- Minimum instruction execution time: One instruction per clock cycle
- General-purpose registers: Eight 8-bit registers
- Type of instruction: 81

The following multiply/divide instructions are available only in the RL78-S3 CPU core.
MULHU (unsigned 16-bit multiplication)
MULH (signed 16-bit multiplication)
DIVHU (unsigned 16-bit division)
DIVWU (unsigned 32-bit division)

MACHU (unsigned multiplication/accumulation (16 bits × 16 bits) + 32 bits)
MACH (signed multiplication/accumulation (16 bits × 16 bits) + 32 bits)

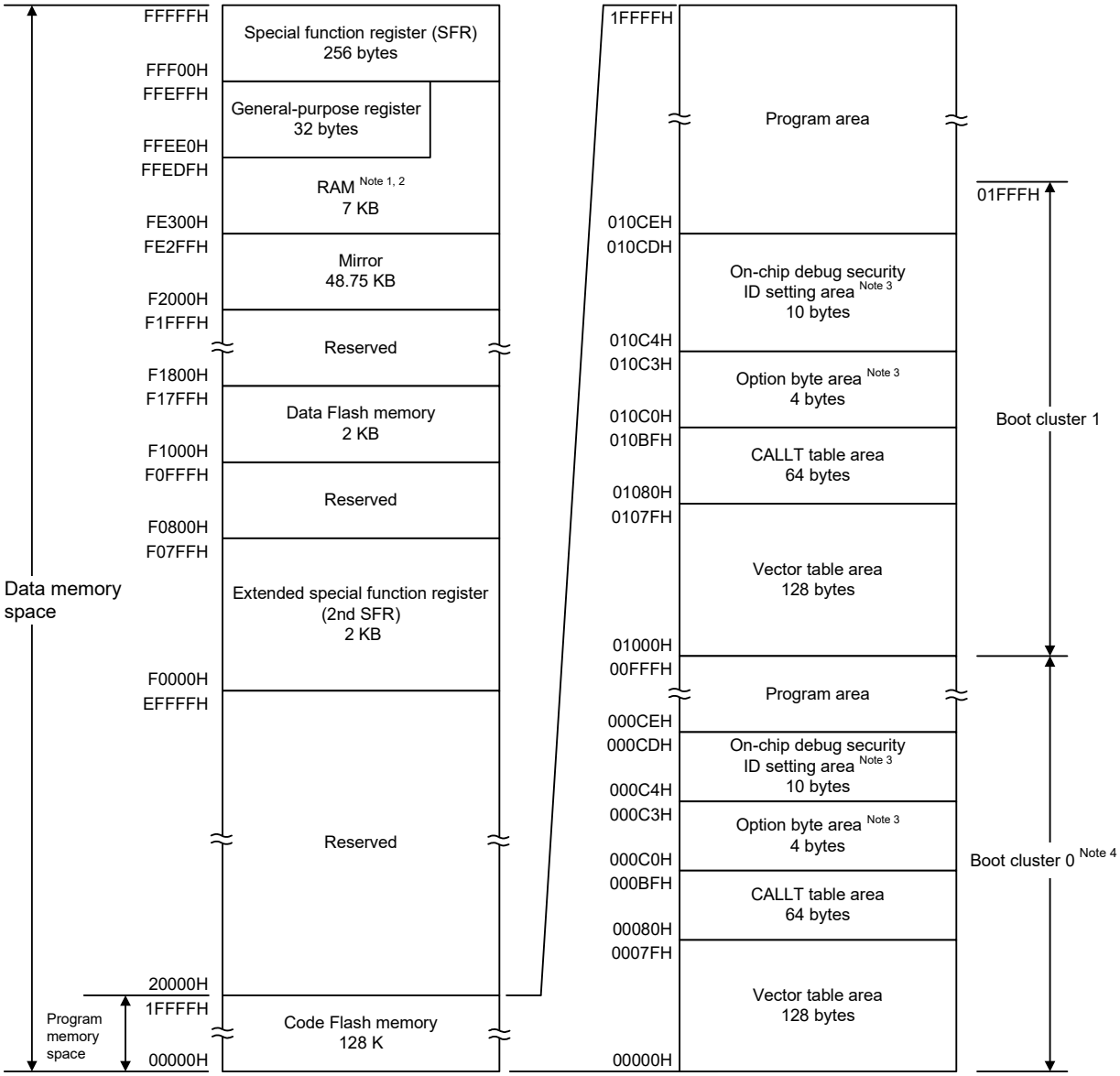
- Data allocation: Little endian

The R9A02G015 supports an OCD trace function.

3.1 Memory Space

Products in the R9A02G015 can access a 1 MB address space. Figures 3 - 1 shows the memory maps.

Figure 3 - 1 Memory Map



Note 1. Do not allocate the stack area, data buffers, branch destinations in the processing of vectored interrupts to the area from FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.

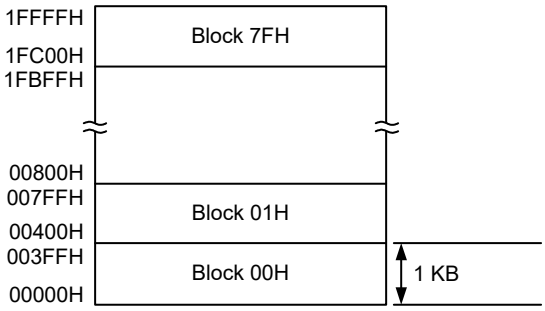
Note 2. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 3. When boot swap is not used:Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

Note 4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **22.7 Security Settings**).

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively.**
Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 19.3.3 RAM parity error detection function.

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3 - 1 Correspondence Between Address Values and Block Numbers in Flash Memory**.



Correspondence between the address values and block numbers in the flash memory are shown below.

Table 3 - 1 Correspondence Between Address Values and Block Numbers in Flash Memory

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	00H	08000H to 083FFH	20H	10000H to 103FFH	40H	18000H to 183FFH	60H
00400H to 007FFH	01H	08400H to 087FFH	21H	10400H to 107FFH	41H	18400H to 187FFH	61H
00800H to 00BFFH	02H	08800H to 07BFFH	22H	10800H to 10BFFH	42H	18800H to 17BFFH	62H
00C00H to 00FFFH	03H	08C00H to 08FFFH	23H	10C00H to 10FFFH	43H	18C00H to 18FFFH	63H
01000H to 013FFH	04H	09000H to 093FFH	24H	11000H to 113FFH	44H	19000H to 193FFH	64H
01400H to 017FFH	05H	09400H to 097FFH	25H	11400H to 117FFH	45H	19400H to 197FFH	65H
01800H to 01BFFH	06H	09800H to 09BFFH	26H	11800H to 11BFFH	46H	19800H to 19BFFH	66H
01C00H to 01FFFH	07H	09C00H to 09FFFH	27H	11C00H to 11FFFH	47H	19C00H to 19FFFH	67H
02000H to 023FFH	08H	0A000H to 0A3FFH	28H	12000H to 123FFH	48H	1A000H to 1A3FFH	68H
02400H to 027FFH	09H	0A400H to 0A7FFH	29H	12400H to 127FFH	49H	1A400H to 1A7FFH	69H
02800H to 02BFFH	0AH	0A800H to 0ABFFH	2AH	12800H to 12BFFH	4AH	1A800H to 1ABFFH	6AH
02C00H to 02FFFH	0BH	0AC00H to 0AFFFH	2BH	12C00H to 12FFFH	4BH	1AC00H to 1AFFFH	6BH
03000H to 033FFH	0CH	0B000H to 0B3FFH	2CH	13000H to 133FFH	4CH	1B000H to 1B3FFH	6CH
03400H to 037FFH	0DH	0B400H to 0B7FFH	2DH	13400H to 137FFH	4DH	1B400H to 1B7FFH	6DH
03800H to 03BFFH	0EH	0B800H to 0BBFFH	2EH	13800H to 13BFFH	4EH	1B800H to 1BBFFH	6EH
03C00H to 03FFFH	0FH	0BC00H to 0BFFFH	2FH	13C00H to 13FFFH	4FH	1BC00H to 1BFFFH	6FH
04000H to 043FFH	10H	0C000H to 0C3FFH	30H	14000H to 143FFH	50H	1C000H to 1C3FFH	70H
04400H to 047FFH	11H	0C400H to 0C7FFH	31H	14400H to 147FFH	51H	1C400H to 1C7FFH	71H
04800H to 04BFFH	12H	0C800H to 0CBFFH	32H	14800H to 14BFFH	52H	1C800H to 1CBFFH	72H
04C00H to 04FFFH	13H	0CC00H to 0CFFFH	33H	14C00H to 14FFFH	53H	1CC00H to 1CFFFH	73H
05000H to 053FFH	14H	0D000H to 0D3FFH	34H	15000H to 153FFH	54H	1D000H to 1D3FFH	74H
05400H to 057FFH	15H	0D400H to 0D7FFH	35H	15400H to 157FFH	55H	1D400H to 1D7FFH	75H
05800H to 05BFFH	16H	0D800H to 0DBFFH	36H	15800H to 15BFFH	56H	1D800H to 1DBFFH	76H
05C00H to 05FFFH	17H	0DC00H to 0DFFFH	37H	15C00H to 15FFFH	57H	1DC00H to 1DFFFH	77H
06000H to 063FFH	18H	0E000H to 0E3FFH	38H	16000H to 163FFH	58H	1E000H to 1E3FFH	78H
06400H to 067FFH	19H	0E400H to 0E7FFH	39H	16400H to 167FFH	59H	1E400H to 1E7FFH	79H
06800H to 06BFFH	1AH	0E800H to 0EBFFH	3AH	16800H to 16BFFH	5AH	1E800H to 1EBFFH	7AH
06C00H to 06FFFH	1BH	0EC00H to 0EFFFH	3BH	16C00H to 16FFFH	5BH	1EC00H to 1EFFFH	7BH
07000H to 073FFH	1CH	0F000H to 0F3FFH	3CH	17000H to 173FFH	5CH	1F000H to 1F3FFH	7CH
07400H to 077FFH	1DH	0F400H to 0F7FFH	3DH	17400H to 177FFH	5DH	1F400H to 1F7FFH	7DH
07800H to 07BFFH	1EH	0F800H to 0FBFFH	3EH	17800H to 17BFFH	5EH	1F800H to 1FBFFH	7EH
07C00H to 07FFFH	1FH	0FC00H to 0FFFFH	3FH	17C00H to 17FFFH	5FH	1FC00H to 1FFFFH	7FH

3.1.1 Internal program memory space

The internal program memory space stores the program and table data.

The R9A02G015 products incorporate internal ROM (flash memory), as shown below.

Table 3 - 2 Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R9A02G0150	Flash memory	131072 × 8 bits (00000H to 1FFFFH)
R9A02G0151		

The internal program memory space is divided into the following areas.

(1) Vector table area

The 128-byte area 00000H to 0007FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area. Furthermore, the interrupt jump address is a 64 K address of 00000H to 0FFFFH, because the vector code is assumed to be 2 bytes.

Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

To use the boot swap function, set a vector table also at 01000H to 0107FH.

Table 3 - 3 lists the vector table. "√" indicates an interrupt source which is supported. "—" indicates an interrupt source which is not supported.

Table 3 - 3 Vector Table (1/2)

Vector Table Address	Interrupt Source	With USB	Without USB
		32-pin	32-pin
00000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE	✓	✓
00004H	INTWDT1	✓	✓
00006H	INTLVI	✓	✓
00008H	INTP0	✓	✓
0000AH	INTP1	✓	✓
0000CH	INTP2	✓	✓
0000EH	INTP3	✓	✓
00010H	INTP4	✓	✓
00012H	INTP5	✓	✓
00014H	INTP6	✓	✓
00016H	INTST0/INTCSI00/INTIIC00	✓	✓
00018H	INTSR0/INTCSI01/INTIIC01	✓	✓
0001EH	INTSRE0	✓	✓
00020H	INTTM00	✓	✓
00026H	INTTM01H	✓	✓
00028H	INTTM03H	✓	✓
0002AH	INTTM01	✓	✓
0002CH	INTTM02	✓	✓
0002EH	INTTM03	✓	✓
00030H	INTIICA0	✓	✓
00032H	INTIICA1	✓	✓
00034H	INTAD	✓	✓

Table 3 - 3 Vector Table (2/2)

Vector Table Address	Interrupt Source	With USB	Without USB
		32-pin	32-pin
00038H	INTIT	✓	✓
0003CH	INTUSB	✓	✓
0003EH	INTRSUM	✓	✓
00040H	INTIICA2	—	✓
00044H	INTTM04	✓	✓
00046H	INTTM05	✓	✓
00048H	INTTM06	✓	✓
0004AH	INTTM07	✓	✓
0004CH	INTP7	✓	✓
0004EH	INTP8	✓	✓
00050H	INTP9	✓	✓
00052H	INTP10	✓	✓
00054H	INTP11	✓	✓
00056H	INTP12	✓	✓
00058H	INTP13	✓	✓
0005AH	INTP14	—	✓
0005CH	INTP15	—	✓
00062H	INTFL	✓	✓
0007EH	BRK	✓	✓

(2) CALLT instruction table area

The 64-byte area 00080H to 000BFH can store the subroutine entry address of a 2-byte call instruction (CALLT). Set the subroutine entry address to a value in a range of 00000H to 0FFFFH (because an address code is 2 bytes).

To use the boot swap function, set a CALLT instruction table also at 01080H to 010BFH.

(3) Option byte area

A 4-byte area of 000C0H to 000C3H can be used as an option byte area. Set the option byte at 010C0H to 010C3H when the boot swap is used. For details, see **CHAPTER 21 OPTION BYTE**.

(4) On-chip debug security ID setting area

A 10-byte area of 000C4H to 000CDH and 010C4H to 010CDH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 000C4H to 000CDH when the boot swap is not used and at 000C4H to 000CDH and at 010C4H to 010CDH when the boot swap is used. For details, see **CHAPTER 23 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The R9A02G015 mirrors the code flash area of 00000H to 0FFFFH, to F0000H to FFFFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

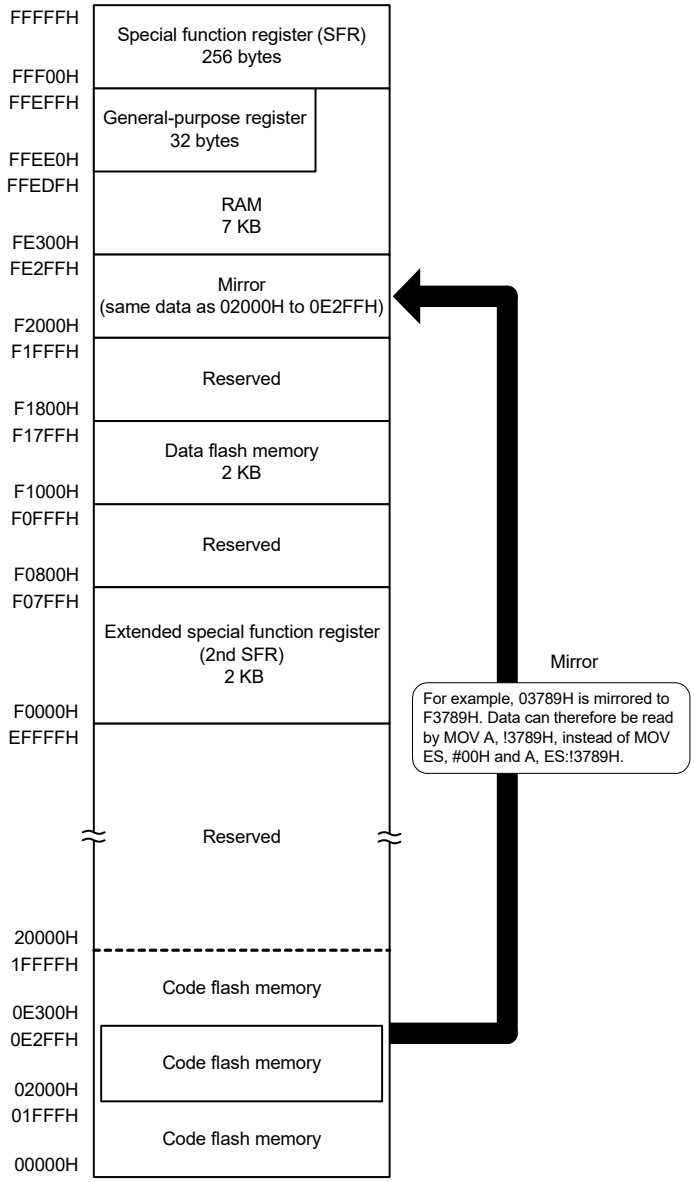
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the special function register (SFR), extended special function register (2nd SFR), RAM, data flash memory, and use prohibited areas.

See **3.1 Memory Space** for the mirror area of each product.

The mirror area can only be read and no instruction can be fetched from this area.

The following show examples.

Example



The PMC register is described below.

• Processor mode control register (PMC)
This register sets the flash memory space for mirroring to area from F0000H to FFFFFH.
The PMC register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Figure 3 - 2 Format of Configuration of Processor mode control register (PMC)

Address: FFFFEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
PMC	0	0	0	0	0	0	0	MAA

MAA	Selection of flash memory space for mirroring to area from F0000H to FFFFFH
0	00000H to 0FFFFH is mirrored to F0000H to FFFFFH
1	Setting prohibited

Caution 1. Be sure to clear bit 0 (MAA) of this register to 0 (default value).
Caution 2. After setting the PMC register, wait for at least one instruction and access the mirror area.

3.1.3 Internal data memory space

The R9A02G015 products incorporate the following RAMs.

Table 3 - 4 Internal RAM Capacity

Part Number	Internal RAM
R9A02G0150	7168 × 8 bits (FE300H to FFEFFH)
R9A02G0151	7168 × 8 bits (FE300H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are fetched (it is prohibited to use the general-purpose register area for fetching instructions). Four general-purpose register banks consisting of eight 8-bit registers per bank are assigned to the 32-byte area of FFEE0H to FFEFFH of the internal RAM area.
The internal RAM is used as stack memory.

Caution 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.
Caution 2. Do not allocate the stack area, data buffers, branch destinations in the processing of vectored interrupts to the area from FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.
Caution 3. The flash library uses RAM in self-programming and rewriting of the data flash memory. The target products and start address of the RAM areas used by the flash library are shown below.
R9A02G0150/R9A02G0151: Start address FE300H
Caution 4. The internal RAM area in the following products cannot be used as stack memory when using the on-chip debugging trace function.
R9A02G0150/R9A02G0151: FE300H to FEAFFH

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH (see **Tables 3 - 5** in **3.2.4 Special function registers (SFRs)**).

Caution Do not access addresses to which SFRs are not assigned.

3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area F0000H to F07FFH (see **Tables 3 - 6** in **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**).

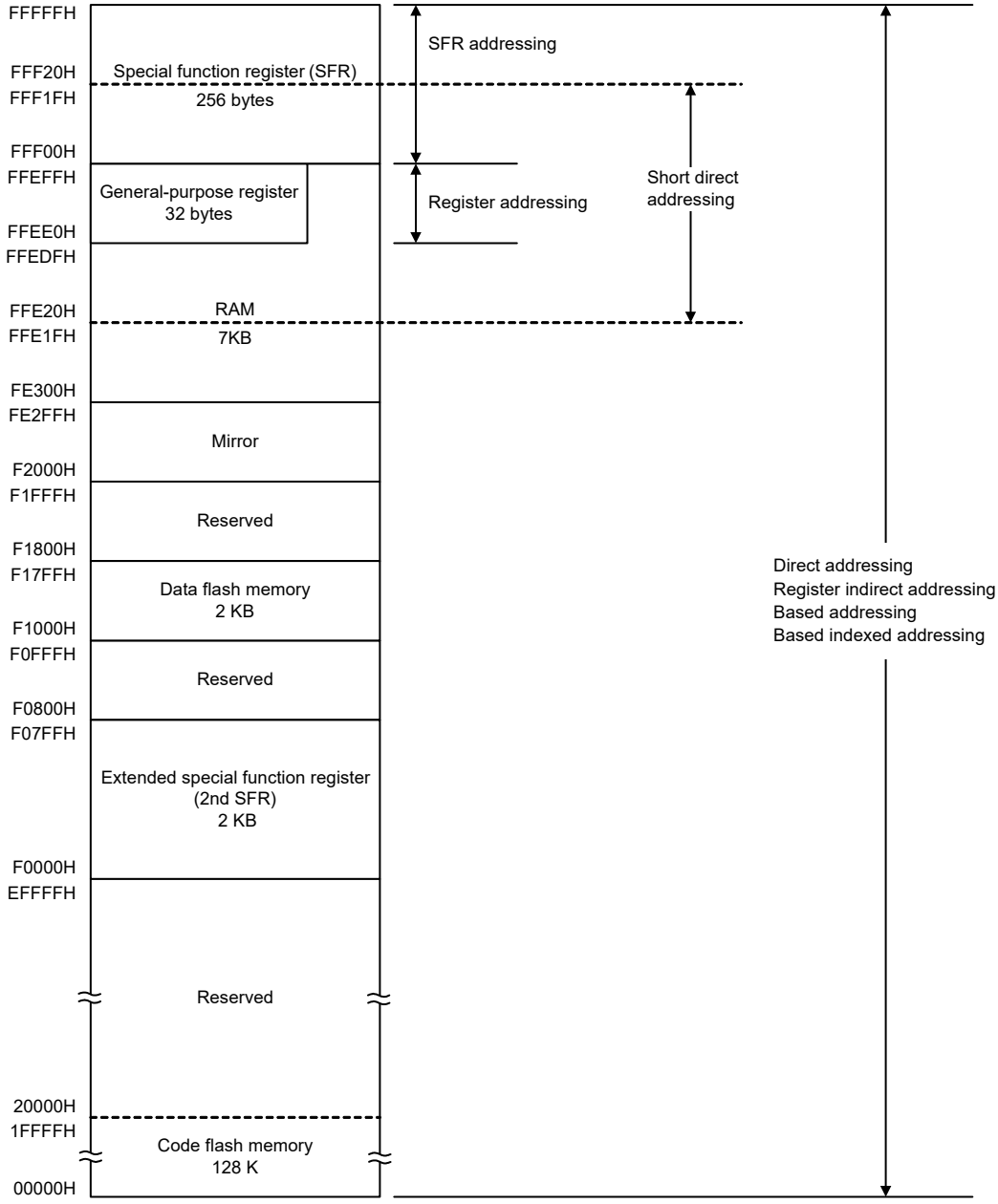
Caution Do not access addresses to which extended SFRs are not assigned.

3.1.6 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the R9A02G015, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figure 3 - 3 shows correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3 - 3 Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The R9A02G015 products incorporate the following processor registers.

3.2.1 Control registers

The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

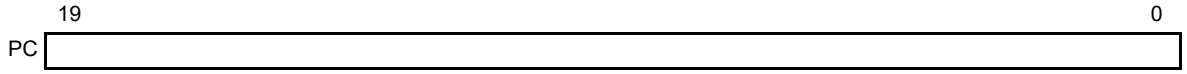
(1) Program counter (PC)

The program counter is a 20-bit register that holds the address information of the next program to be executed.

In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.

Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 3 - 4 Format of Program Counter



(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags set/reset by instruction execution. Program status word contents are stored in the stack area upon vectored interrupt request is acknowledged or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions. Reset signal generation sets the PSW register to 06H.

Figure 3 - 5 Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

This flag controls the interrupt request acknowledge operations of the CPU.

When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISP0), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

(b) Zero flag (Z)

When the operation result is zero, this flag is set (1). It is reset (0) in all other cases.

(c) Register bank select flags (RBS0, RBS1)

These are 2-bit flags to select one of the four register banks.

In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

(d) Auxiliary carry flag (AC)

If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (1). It is reset (0) in all other cases.

(e) In-service priority flags (ISP1, ISP0)

This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **14.3.3**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit operation instruction execution.

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal RAM area can be set as the stack area.

Figure 3 - 6 Format of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	0

In stack addressing through a stack pointer, the SP is decremented ahead of write (save) to the stack memory and is incremented after read (restore) from the stack memory.

Caution 1. Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

Caution 2. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Caution 3. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

Caution 4. Use of the RAM areas of the following products is prohibited when performing self-programming and rewriting the data flash memory, because these areas are used for each library.
R9A02G0150/R9A02G0151: FE300H to FE709H

Caution 5. The internal RAM area in the following products cannot be used as stack memory when using the on-chip debugging trace function.
R9A02G0150/R9A02G0151: FE300H to FEAFFH

3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FFEE0H to FFEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

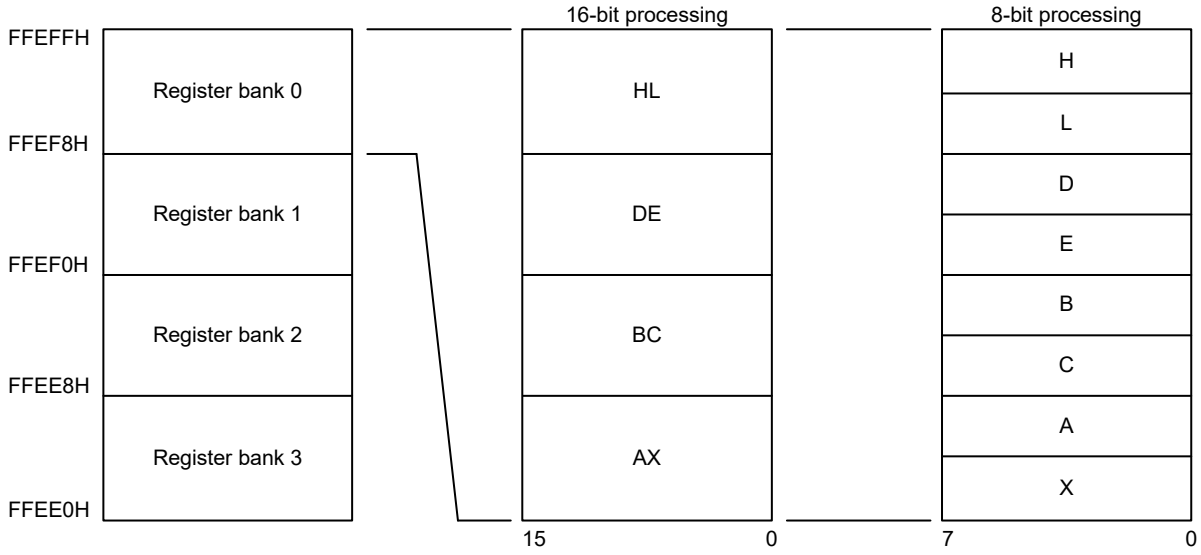
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupt processing for each bank.

Caution It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Figure 3 - 7 Configuration of General-Purpose Registers

(a) Function name



3.2.3 ES and CS registers

The ES register and CS register are used to specify the higher address for data access and when a branch instruction is executed (register direct addressing), respectively.
The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

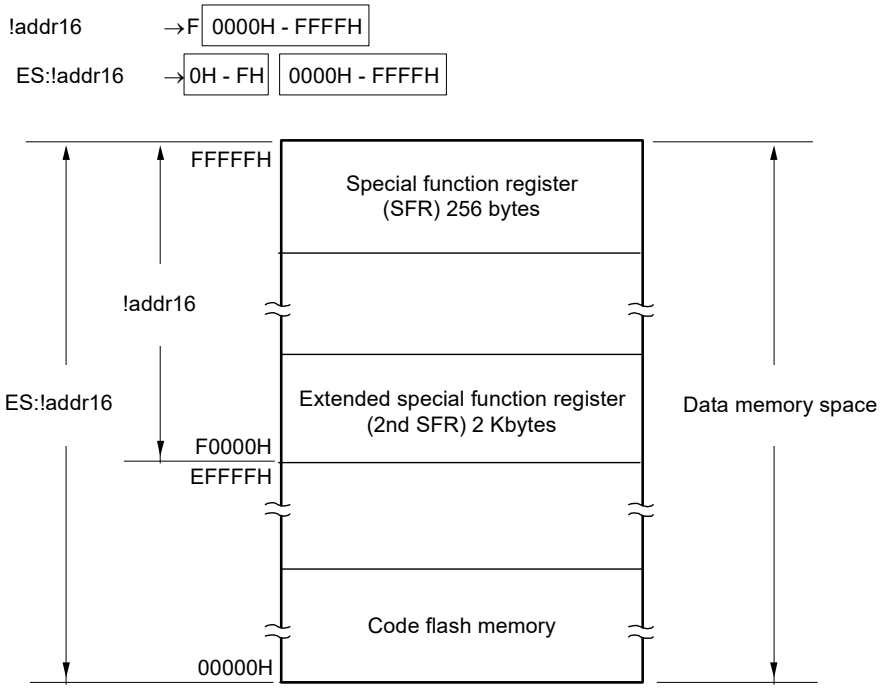
Figure 3 - 8 Configuration of ES and CS Registers

	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0

	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CS2	CS1	CS0

Though the data area which can be accessed with 16-bit addresses is the 64 Kbytes from F0000H to FFFFFH, using the ES register as well extends this to the 1 Mbyte from 00000H to FFFFFH.

Figure 3 - 9 Extension of Data Area Which Can Be Accessed



3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each SFR has a special function.
SFRs are allocated to the FFF00H to FFFFFH area.
SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.
Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit manipulation instruction operand (sfr.bit).
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>, <Bit number> or <Address>, <Bit number>
- 8-bit manipulation
  Describe the symbol defined by the assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol defined by the assembler for the 16-bit manipulation instruction operand (sfrp). When specifying an address, describe an even address.

Tables 3 - 5 gives lists of the SFRs. The meanings of items in the table are as follows.

- Symbol
  This item indicates the address of a special function register. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  This item indicates whether the corresponding SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "—" indicates a bit unit for which manipulation is not possible.
- After reset
  This item indicates each register status upon reset signal generation.

Caution **Do not access addresses to which SFRs are not assigned.**

Remark For extended SFRs (2nd SFRs), see **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 3 - 5 Special Function Register (SFR) List (1/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF00H	Port register 0	P0		R/W	✓	✓	—	00H
FFF02H	Port register 2	P2		R/W	✓	✓	—	00H
FFF04H	Port register 4	P4		R/W	✓	✓	—	00H
FFF05H	Port register 5	P5		R/W	✓	✓	—	00H
FFF06H	Port register 6	P6		R/W	✓	✓	—	00H
FFF07H	Port register 7 Note 1	P7		R/W	✓	✓	—	00H
FFF0CH	Port register 12	P12		R/W	✓	✓	—	Undefined
FFF0DH	Port register 13	P13		R/W	✓	✓	—	Undefined
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	—	✓	✓	0000H
FFF11H		—			—	—		
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	—	✓	✓	0000H
FFF13H		—			—	—		
FFF18H	Timer data register 00	TDR00		R/W	—	—	✓	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	—	✓	✓	00H
FFF1BH		TDR01H			—	✓		00H
FFF1EH	10-bit A/D conversion result register	ADCR		R	—	—	✓	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	—	✓	—	00H
FFF20H	Port mode register 0	PM0		R/W	✓	✓	—	FFH
FFF22H	Port mode register 2	PM2		R/W	✓	✓	—	FFH
FFF24H	Port mode register 4	PM4		R/W	✓	✓	—	FFH
FFF25H	Port mode register 5	PM5		R/W	✓	✓	—	FFH
FFF26H	Port mode register 6	PM6		R/W	✓	✓	—	FFH
FFF27H	Port mode register 7 Note 1	PM7		R/W	✓	✓	—	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	✓	✓	—	00H
FFF31H	Analog input channel specification register	ADS		R/W	✓	✓	—	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	✓	✓	—	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	✓	✓	—	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	✓	✓	—	00H
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	✓	✓	—	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	✓	✓	—	00H
FFF50H	IICA shift register 0	IICA0		R/W	—	✓	—	00H
FFF51H	IICA status register 0	IICS0		R	✓	✓	—	00H
FFF52H	IICA flag register 0	IICF0		R/W	✓	✓	—	00H
FFF54H	IICA shift register 1	IICA1		R/W	—	✓	—	00H
FFF55H	IICA status register 1	IICS1		R	✓	✓	—	00H
FFF56H	IICA flag register 1	IICF1		R/W	✓	✓	—	00H
FFF60H	IICA shift register 2 Note 1	IICA2		R/W	—	✓	—	00H
FFF61H	IICA status register 2 Note 1	IICS2		R	✓	✓	—	00H
FFF62H	IICA flag register 2 Note 1	IICF2		R/W	✓	✓	—	00H

Table 3 - 5 Special Function Register (SFR) List (2/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF64H	Timer data register 02	TDR02		R/W	—	—	✓	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	—	✓	✓	00H
FFF67H		TDR03H			—	✓		00H
FFF68H	Timer data register 04	TDR04		R/W	—	—	✓	0000H
FFF69H								
FFF6AH	Timer data register 05	TDR05		R/W	—	—	✓	0000H
FFF6BH								
FFF6CH	Timer data register 06	TDR06		R/W	—	—	✓	0000H
FFF6DH								
FFF6EH	Timer data register 07	TDR07		R/W	—	—	✓	0000H
FFF6FH								
FFF90H	12-bit interval timer control register	ITMC		R/W	—	—	✓	0FFFH
FFF91H								
FFFA0H	Clock operation mode control register	CMC		R/W	—	✓	—	00H
FFFA1H	Clock operation status control register	CSC		R/W	✓	✓	—	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	✓	✓	—	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	—	✓	—	07H
FFFA4H	System clock control register	CKC		R/W	✓	✓	—	00H
FFFA5H	Clock output select register 0	CKS0		R/W	✓	✓	—	00H
FFFA8H	Reset control flag register	RESF		R	—	✓	—	Undefined Note 2
FFFA9H	Voltage detection register	LVIM		R/W	✓	✓	—	00H Note 2
FFFAAH	Voltage detection level register	LVIS		R/W	✓	✓	—	Note 4
FFFABH	Watchdog timer enable register	WDTE		R/W	—	✓	—	1AH/9AH Note 3
FFFACH	CRC input register	CRCIN		R/W	—	✓	—	00H
FFFD0H	Interrupt request flag register 2	IF2L	IF2	R/W	✓	✓	—	00H
FFFD1H		IF2H		R/W	✓	✓	—	00H
FFFD4H	Interrupt mask flag register 2	MK2L	MK2	R/W	✓	✓	—	FFH
FFFD5H		MK2H		R/W	✓	✓	—	FFH
FFFD8H	Priority specification flag register 02	PR02L	PR02	R/W	✓	✓	—	FFH
FFFD9H		PR02H		R/W	✓	✓	—	FFH
FFFDCH	Priority specification flag register 12	PR12L	PR12	R/W	✓	✓	—	FFH
FFFDDH		PR12H		R/W	✓	✓	—	FFH
FFFE0H	Interrupt request flag register 0	IF0L	IF0	R/W	✓	✓	✓	00H
FFFE1H		IF0H		R/W	✓	✓		00H
FFFE2H	Interrupt request flag register 1	IF1L	IF1	R/W	✓	✓	✓	00H
FFFE3H		IF1H		R/W	✓	✓		00H
FFFE4H	Interrupt mask flag register 0	MK0L	MK0	R/W	✓	✓	✓	FFH
FFFE5H		MK0H		R/W	✓	✓		FFH
FFFE6H	Interrupt mask flag register 1	MK1L	MK1	R/W	✓	✓	✓	FFH
FFFE7H		MK1H		R/W	✓	✓		FFH
FFFE8H	Priority specification flag register 00	PR00L	PR00	R/W	✓	✓	✓	FFH
FFFE9H		PR00H		R/W	✓	✓		FFH
FFFEAH	Priority specification flag register 01	PR01L	PR01	R/W	✓	✓	✓	FFH
FFFEBH		PR01H		R/W	✓	✓		FFH

Table 3 - 5 Special Function Register (SFR) List (3/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFECH	Priority specification flag register 10	PR10L	PR10	R/W	✓	✓	✓	FFH
FFFEDH		PR10H		R/W	✓	✓		FFH
FFFEEH	Priority specification flag register 11	PR11L	PR11	R/W	✓	✓	✓	FFH
FFFEFH		PR11L		R/W	✓	✓		FFH
FFFF0H	Multiply and accumulation register (L)	MACRL		R/W	✓	✓	✓	0000H
FFFF1H								
FFFF2H	Multiply and accumulation register (H)	MACRH		R/W	—	—	✓	0000H
FFFF3H								
FFFFEH	Processor mode control register	PMC		R/W	✓	✓	—	00H

Note 1. This register is incorporated with R9A02G0151, but is not incorporated with R9A02G0150.

Note 2. These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM parity error	Reset by illegal-memory access	Reset by LVD
RESF	TRAP	Cleared (0)		Set (1)	Held			Held
	WDTRF			Held	Set (1)	Held		
	RPERF			Held		Set (1)	Held	
	IAWRF			Held			Set (1)	
	LVIRF			Held				Set (1)
LVIM	LVISEN	Cleared (0)						Held
	LVIOMSK	Held						
	LVIF							

Note 3. The reset value of the WDTE register is determined by the setting of the option byte.

Note 4. The reset value of the LVIS register is determined by the setting of the option byte.

Remark For extended SFRs (2nd SFRs), see **Tables 3 - 6 Extended Special Function Register (2nd SFR) List**.

3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)

Unlike a general-purpose register, each extended SFR (2nd SFR) has a special function.
Extended SFRs are allocated to the F0000H to F07FFH area. SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.
Extended SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.
Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit manipulation instruction operand (!addr16.bit)
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>, <Bit number> or <Address>, <Bit number>
- 8-bit manipulation
  Describe the symbol defined by the assembler for the 8-bit manipulation instruction operand (!addr16). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol defined by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Tables 3 - 6 gives lists of the extended SFRs. The meanings of items in the table are as follows.

- Symbol
  This item indicates the address of an extended SFR. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  This item indicates whether the corresponding extended SFR can be read or written.
  R/W:Read/write enable
  R:Read only
  W:Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "—" indicates a bit unit for which manipulation is not possible.
- After reset
  This item indicates each register status upon reset signal generation.

Caution Do not access addresses to which extended SFRs are not assigned.

Remark For SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.

Table 3 - 6 Extended Special Function Register (2nd SFR) List (1/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0010H	A/D converter mode register 2	ADM2		R/W	✓	✓	—	00H
F0011H	Conversion result comparison upper limit setting register	ADUL		R/W	—	✓	—	FFH
F0012H	Conversion result comparison lower limit setting register	ADLL		R/W	—	✓	—	00H
F0013H	A/D test register	ADTES		R/W	—	✓	—	00H
F0030H	Pull-up resistor option register 0	PU0		R/W	✓	✓	—	00H
F0034H	Pull-up resistor option register 4	PU4		R/W	✓	✓	—	00H
F0035H	Pull-up resistor option register 5	PU5		R/W	✓	✓	—	00H
F0037H	Pull-up resistor option register 7 Note 1	PU7		R/W	✓	✓	—	00H
F0040H	Port input mode register 0	PIM0		R/W	✓	✓	—	00H
F0045H	Port input mode register 5	PIM5		R/W	✓	✓	—	00H
F0047H	Port input mode register 7 Note 1	PIM7		R/W	✓	✓	—	00H
F0050H	Port output mode register 0	POM0		R/W	✓	✓	—	00H
F0055H	Port output mode register 5	POM5		R/W	✓	✓	—	00H
F0057H	Port output mode register 7 Note 1	POM7		R/W	✓	✓	—	00H
F0060H	Port mode control register 0	PMC0		R/W	✓	✓	—	FFH
F0062H	Port mode control register 2	PMC2		R/W	✓	✓	—	FFH
F0070H	Noise filter enable register 0	NFEN0		R/W	✓	✓	—	00H
F0071H	Noise filter enable register 1	NFEN1		R/W	✓	✓	—	00H
F0074H	Timer input select register 0	TIS0		R/W	—	✓	—	00H
F0077H	Invalid memory access detection control register	IAWCTL		R/W	—	✓	—	00H
F007BH	Port mode selection register	PMS		R/W	✓	✓	—	00H
F0090H	Data flash control register	DFLCTL		R/W	✓	✓	—	00H
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV		R/W	—	✓	—	Undefined Note 2
F00F0H	Peripheral enable register 0	PER0		R/W	✓	✓	—	00H
F00F1H	Peripheral reset control register 0	PRR0		R/W	✓	✓	—	00H
F00F3H	Subsystem clock supply mode control register	OSMC		R/W	✓	✓	—	00H
F00F5H	RAM parity error control register	RPECTL		R/W	✓	✓	—	00H
F00FCH	Peripheral enable register 2	PER2		R/W	✓	✓	—	00H
F00FDH	Peripheral reset control register 2	PRR2		R/W	✓	✓	—	00H
F00FEH	BCD adjust result register	BCDADJ		R	—	✓	—	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	—	✓	✓	0000H
F0101H		—		R	—	—		
F0102H	Serial status register 01	SSR01L	SSR01	R	—	✓	✓	0000H
F0103H		—		R	—	—		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	—	✓	✓	0000H
F0109H		—			—	—		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	—	✓	✓	0000H
F010BH		—			—	—		
F0110H	Serial mode register 00	SMR00		R/W	—	—	✓	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	—	—	✓	0020H
F0113H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	—	—	✓	0087H
F0119H								

Table 3 - 6 Extended Special Function Register (2nd SFR) List (2/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F011AH	Serial communication operation setting register 01	SCR01		R/W	—	—	✓	0087H
F011BH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	✓	✓	✓	0000H
F0121H		—			—	—		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	✓	✓	✓	0000H
F0123H		—			—	—		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	✓	✓	✓	0000H
F0125H		—			—	—		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	—	✓	✓	0000H
F0127H		—			—	—		
F0128H	Serial output register 0	SO0		R/W	—	—	✓	0303H
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	✓	✓	✓	0000H
F012BH		—			—	—		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	—	✓	✓	0000H
F0135H		—			—	—		
F0138H	Serial standby control register 0	SSC0L	SSC0	R/W	—	✓	✓	0000H
F0139H		—			—	—		
F0180H	Timer counter register 00	TCR00		R	—	—	✓	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	—	—	✓	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	—	—	✓	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	—	—	✓	FFFFH
F0187H								
F0188H	Timer counter register 04	TCR04		R	—	—	✓	FFFFH
F0189H								
F018AH	Timer counter register 05	TCR05		R	—	—	✓	FFFFH
F018BH								
F018CH	Timer counter register 06	TCR06		R	—	—	✓	FFFFH
F018DH								
F018EH	Timer counter register 07	TCR07		R	—	—	✓	FFFFH
F018FH								
F0190H	Timer mode register 00	TMR00		R/W	—	—	✓	0000H
F0191H								
F0192H	Timer mode register 01	TMR01		R/W	—	—	✓	0000H
F0193H								
F0194H	Timer mode register 02	TMR02		R/W	—	—	✓	0000H
F0195H								
F0196H	Timer mode register 03	TMR03		R/W	—	—	✓	0000H
F0197H								
F0198H	Timer mode register 04	TMR04		R/W	—	—	✓	0000H
F0199H								
F019AH	Timer mode register 05	TMR05		R/W	—	—	✓	0000H
F019BH								

Table 3 - 6 Extended Special Function Register (2nd SFR) List (3/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F019CH	Timer mode register 06	TMR06		R/W	—	—	✓	0000H
F019DH								
F019EH	Timer mode register 07	TMR07		R/W	—	—	✓	0000H
F019FH								
F01A0H	Timer status register 00	TSR00L	TSR00	R	—	✓	✓	0000H
F01A1H		—			—	—		
F01A2H	Timer status register 01	TSR01L	TSR01	R	—	✓	✓	0000H
F01A3H		—			—	—		
F01A4H	Timer status register 02	TSR02L	TSR02	R	—	✓	✓	0000H
F01A5H		—			—	—		
F01A6H	Timer status register 03	TSR03L	TSR03	R	—	✓	✓	0000H
F01A7H		—			—	—		
F01A8H	Timer status register 04	TSR04L	TSR04	R	—	✓	✓	0000H
F01A9H		—			—	—		
F01AAH	Timer status register 05	TSR05L	TSR05	R	—	✓	✓	0000H
F01ABH		—			—	—		
F01ACH	Timer status register 06	TSR06L	TSR06	R	—	✓	✓	0000H
F01ADH		—			—	—		
F01AEH	Timer status register 07	TSR07L	TSR07	R	—	✓	✓	0000H
F01AFH		—			—	—		
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	✓	✓	✓	0000H
F01B1H		—			—	—		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	✓	✓	✓	0000H
F01B3H		—			—	—		
F01B4H	IICCTL01	TT0L	TT0	R/W	✓	✓	✓	0000H
F01B5H		—			—	—		
F01B6H	Timer clock select register 0	TPS0		R/W	—	—	✓	0000H
F01B7H								
F01B8H	Timer output register 0	TO0L	TO0	R/W	—	✓	✓	0000H
F01B9H		SYSCFG1			—	—		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	✓	✓	✓	0000H
F01BBH		—			—	—		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	—	✓	✓	0000H
F01BDH		—			—	—		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	—	✓	✓	0000H
F01BFH		—			—	—		
F0230H	IICA control register 00	IICCTL00		R/W	✓	✓	—	00H
F0231H	IICA control register 01	IICCTL01		R/W	✓	✓	—	00H
F0232H	IICA low-level width setting register 0	IICWL0		R/W	—	✓	—	FFH
F0233H	IICA high-level width setting register 0	IICWH0		R/W	—	✓	—	FFH
F0234H	Slave address register 0	SVA0		R/W	—	✓	—	00H
F0238H	IICA control register 10	IICCTL10		R/W	✓	✓	—	00H
F0239H	IICA control register 11	IICCTL11		R/W	✓	✓	—	00H
F023AH	IICA low-level width setting register 1	IICWL1		R/W	—	✓	—	FFH
F023BH	IICA high-level width setting register 1	IICWH1		R/W	—	✓	—	FFH
F023CH	Slave address register 1	SVA1		R/W	—	✓	—	00H

Table 3 - 6 Extended Special Function Register (2nd SFR) List (4/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0240H	IICA control register 20 Note 1	IICCTL20		R/W	✓	✓	—	00H
F0241H	IICA control register 21 Note 1	IICCTL21		R/W	✓	✓	—	00H
F0242H	IICA low-level width setting register 2 Note 1	IICWL2		R/W	—	✓	—	FFH
F0243H	IICA high-level width setting register 2 Note 1	IICWH2		R/W	—	✓	—	FFH
F0244H	Slave address register 2 Note 1	SVA2		R/W	—	✓	—	00H
F02E5H	PLL control register	DSCCTL		R/W	✓	✓	—	00H
F02E6H	Main clock control register	MCKC		R/W	✓	✓	—	00H
F02F0H	Flash memory CRC control register	CRC0CTL		R/W	✓	✓	—	00H
F02F2H	Flash memory CRC operation result register	PGCRCL		R/W	—	—	✓	0000H
F02F3H								
F02FAH	CRC data register	CRCD		R/W	—	—	✓	0000H
F02FBH								
F0400H	System configuration control register Note 3	SYSCFG		R/W	—	—	✓	0000H
F0401H								
F0402H	System configuration control register 1 Note 3	SYSCFG1		R/W	—	—	✓	0000H
F0403H								
F0404H	System configuration status register 0 Note 3	SYSSTS0		R	—	—	✓	0000H
F0405H								
F0406H	System configuration status register 1 Note 3	SYSSTS1		R	—	—	✓	0000H
F0407H								
F0408H	Device state control register 0 Note 3	DVSTCTR0		R/W	—	—	✓	0000H
F0409H								
F040AH	Device state control register 1 Note 3	DVSTCTR1		R/W	—	—	✓	0000H
F040BH								
F0414H	CFIFO port register Note 3	CFIFOML	CFIFOM	R/W	—	✓	✓	0000H
F0415H		—			—	—		
F0418H	D0FIFO port register Note 3	D0FIFOML	D0FIFOM	R/W	—	✓	✓	0000H
F0419H		—			—	—		
F041CH	D1FIFO port register Note 3	D1FIFOML	D1FIFOM	R/W	—	✓	✓	0000H
F041DH		—			—	—		
F0420H	CFIFO port selection register Note 3	CFIFOSEL		R/W	—	—	✓	0000H
F0421H								
F0422H	CFIFO port control register Note 3	CFIFOCTR		R/W	—	—	✓	0000H
F0423H								
F0428H	D0FIFO port selection register Note 3	D0FIFOSEL		R/W	—	—	✓	0000H
F0429H								
F042AH	D0FIFO port control register Note 3	D0FIFOCTR		R/W	—	—	✓	0000H
F042BH								
F042CH	D1FIFO port selection register Note 3	D1FIFOSEL		R/W	—	—	✓	0000H
F042DH								
F042EH	D1FIFO port control register Note 3	D1FIFOCTR		R/W	—	—	✓	0000H
F042FH								
F0430H	Interrupt enable register 0 Note 3	INTENB0		R/W	—	—	✓	0000H
F0431H								
F0432H	Interrupt enable register 1 Note 3	INTENB1		R/W	—	—	✓	0000H
F0433H								

Table 3 - 6 Extended Special Function Register (2nd SFR) List (5/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0434H	Interrupt enable register 2 Note 3	INTENB2	R/W	—	—	✓	0000H
F0435H							
F0436H	BRDY interrupt enable register Note 3	BRDYENB	R/W	—	—	✓	0000H
F0437H							
F0438H	NRDY interrupt enable register Note 3	NRDYENB	R/W	—	—	✓	0000H
F0439H							
F043AH	BEMP interrupt register Note 3	BEMPENB	R/W	—	—	✓	0000H
F043BH							
F043CH	SOF output configuration register Note 3	SOFCFG	R/W	—	—	✓	0000H
F043DH							
F0440H	Interrupt status register 0 Note 3	INTSTS0	R/W	—	—	✓	00000000 X0000000B
F0441H							
F0442H	Interrupt status register 1 Note 3	INTSTS1	R/W	—	—	✓	XX0X0000 00000000B
F0443H							
F0444H	Interrupt status register 2 Note 3	INTSTS2	R/W	—	—	✓	X00X0000 00000000B
F0445H							
F0446H	BRDY interrupt status register Note 3	BRDYSTS	R/W	—	—	✓	0000H
F0447H							
F0448H	NRDY Interrupt status register Note 3	NRDYSTS	R/W	—	—	✓	0000H
F0449H							
F044AH	BEMP interrupt status register Note 3	BEMPSTS	R/W	—	—	✓	0000H
F044BH							
F044CH	Frame number register Note 3	FRMNUM	R/W	—	—	✓	0000H
F044DH							
F0450H	USB address register Note 3	USBADDR	R	—	—	✓	0000H
F0451H							
F0454H	USB request type register Note 3	USBREQ	Note 4	—	—	✓	0000H
F0455H							
F0456H	USB request value register Note 3	USBVAL	Note 4	—	—	✓	0000H
F0457H							
F0458H	USB request index register Note 3	USBINDX	Note 4	—	—	✓	0000H
F0459H							
F045AH	USB request length register Note 3	USBLENG	Note 4	—	—	✓	0000H
F045BH							
F045CH	DCP configuration registerNote 3	DCPCFG	R/W	—	—	✓	0000H
F045DH							
F045EH	DCP maximum packet size register Note 3	DCPMAXP	R/W	—	—	✓	0000H
F045FH							
F0460H	DCP control register Note 3	DCPCTR	R/W	—	—	✓	0000H
F0461H							
F0464H	Pipe window selection register Note 3	PIPESEL	R/W	—	—	✓	0000H
F0465H							
F0468H	Pipe configuration register Note 3	PIPECFG	R/W	—	—	✓	0000H
F0469H							
F046CH	Pipe maximum packet size register Note 3	PIPEMAXP	R/W	—	—	✓	0000H/ 0040H
F046DH							

Table 3 - 6 Extended Special Function Register (2nd SFR) List (6/6)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F046EH	Pipe interval control register Note 3	PIPEPERI	R/W	—	—	✓	0000H
F046FH							
F0476H	Pipe 4 control register Note 3	PIPE4CTR	R/W	—	—	✓	0000H
F0477H							
F0478H	Pipe 5 control register Note 3	PIPE5CTR	R/W	—	—	✓	0000H
F0479H							
F047AH	Pipe 6 control register Note 3	PIPE6CTR	R/W	—	—	✓	0000H
F047BH							
F047CH	Pipe 7 control register Note 3	PIPE7CTR	R/W	—	—	✓	0000H
F047DH							
F049CH	Pipe 4 transaction counter enable register Note 3	PIPE4TRE	R/W	—	—	✓	0000H
F049DH							
F049EH	Pipe 4 transaction counter register Note 3	PIPE4TRN	R/W	—	—	✓	0000H
F049FH							
F04A0H	Pipe 5 transaction counter enable register Note 3	PIPE5TRE	R/W	—	—	✓	0000H
F04A1H							
F04A2H	Pipe 5 transaction counter register Note 3	PIPE5TRN	R/W	—	—	✓	0000H
F04A3H							
F04B0H	BC control register 0 Note 3	USBBCCTRL0	R/W	—	—	✓	0000H
F04B1H							
F04B4H	BC control register 1 Note 3	USBBCCTRL1	R/W	—	—	✓	0000H
F04B5H							
F04B8H	BC option control register 0 Note 3	USBBCOPT0	R/W	—	—	✓	0000H
F04B9H							
F04BCH	BC option control register 1 Note 3	USBBCOPT1	R/W	—	—	✓	0000H
F04BDH							
F04C4H	USB clock selection register Note 3	UCKSEL	R/W	—	—	✓	0000H
F04C5H							
F04CCH	USB module control register Note 3	USBMC	R/W	—	—	✓	0000H
F04CDH							
F04D0H	Device address 0 configuration register Note 3	DEVADD0	R/W	—	—	✓	0000H
F04D1H							
F04D2H	Device address 1 configuration register Note 3	DEVADD1	R/W	—	—	✓	0000H
F04D3H							
F04D4H	Device address 2 configuration register Note 3	DEVADD2	R/W	—	—	✓	0000H
F04D5H							
F04D6H	Device address 3 configuration register Note 3	DEVADD3	R/W	—	—	✓	0000H
F04D7H							
F04D8H	Device address 4 configuration register Note 3	DEVADD4	R/W	—	—	✓	0000H
F04D9H							
F04DAH	Device address 5 configuration register Note 3	DEVADD5	R/W	—	—	✓	0000H
F04DBH							

Note 1. This register is incorporated with R9A02G0151, but is not incorporated with R9A02G0150.

Note 2. The value after a reset is a value set in FRQSEL2 to FRQSEL0 of the option byte 000C2H.

Note 3. This register is incorporated with R9A02G0150, but is not incorporated with R9A02G0151.

Note 4. When the function controller is selected, the register can only be read. Note that writing the register is prohibited. On the other hand, when the host controller is selected, the register can be read and write.

Remark For SFRs in the SFR area, see **Tables 3 - 5 Special Function Register (SFR) List**.

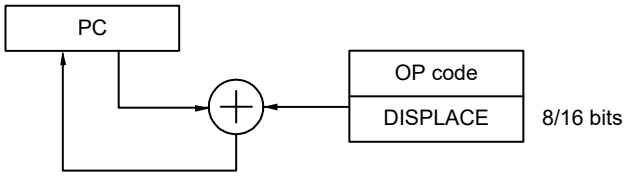
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

Relative addressing stores in the program counter (PC) the result of adding a displacement value included in the instruction word (signed complement data: -128 to +127 or -32768 to +32767) to the program counter (PC)'s value (the start address of the next instruction), and specifies the program address to be used as the branch destination. Relative addressing is applied only to branch instructions.

Figure 3 - 10 Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

Immediate addressing stores immediate data of the instruction word in the program counter, and specifies the program address to be used as the branch destination.

For immediate addressing, CALL !!addr20 or BR !!addr20 is used to specify 20-bit addresses and CALL !addr16 or BR !addr16 is used to specify 16-bit addresses. 0000 is set to the higher 4 bits when specifying 16-bit addresses.

Figure 3 - 11 Example of CALL !!addr20/BR !!addr20

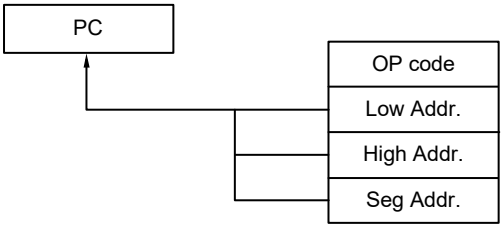
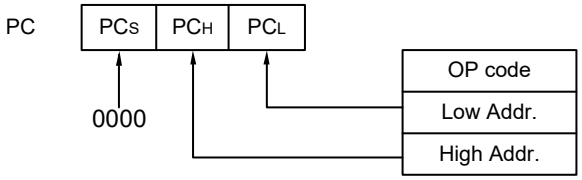


Figure 3 - 12 Example of CALL !addr16/BR !addr16



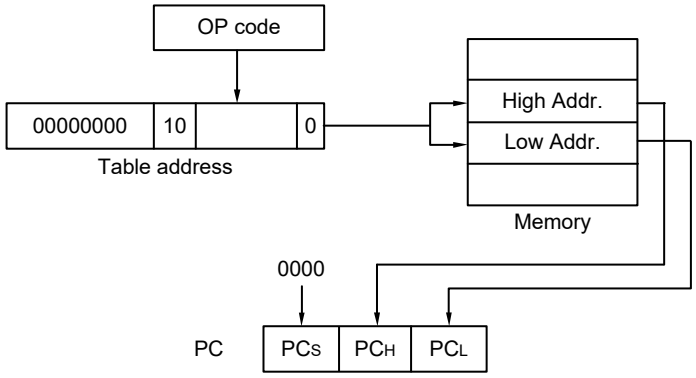
3.3.3 Table indirect addressing

[Function]

Table indirect addressing specifies a table address in the CALLT table area (0080H to 00BFH) with the 5-bit immediate data in the instruction word, stores the contents at that table address and the next address in the program counter (PC) as 16-bit data, and specifies the program address. Table indirect addressing is applied only for CALLT instructions.

In the R9A02G015, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

Figure 3 - 13 Outline of Table Indirect Addressing

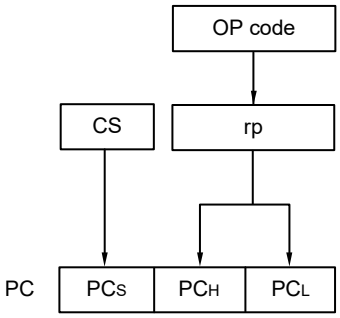


3.3.4 Register indirect addressing

[Function]

Register direct addressing stores in the program counter (PC) the contents of a general-purpose register pair (AX/BC/DE/HL) and CS register of the current register bank specified with the instruction word as 20-bit data, and specifies the program address. Register indirect addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3 - 14 Outline of Register Indirect Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

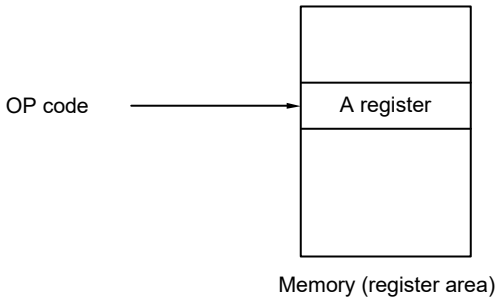
[Function]

Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

Implied addressing can be applied only to MULU X.

Figure 3 - 15 Outline of Implied Addressing



3.4.2 Register addressing

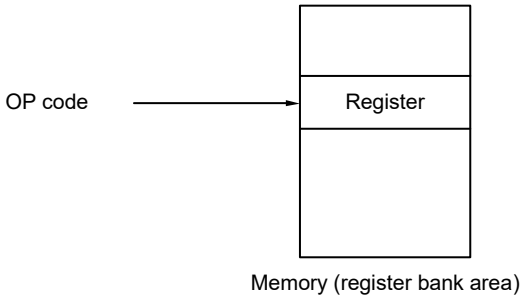
[Function]

Register addressing accesses a general-purpose register as an operand. The instruction word of 3-bit long is used to select an 8-bit register and the instruction word of 2-bit long is used to select a 16-bit register.

[Operand format]

Identifier	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

Figure 3 - 16 Outline of Register Addressing



3.4.3 Direct addressing

[Function]

Direct addressing uses immediate data in the instruction word as an operand address to directly specify the target address.

[Operand format]

Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3 - 17 Example of !addr16

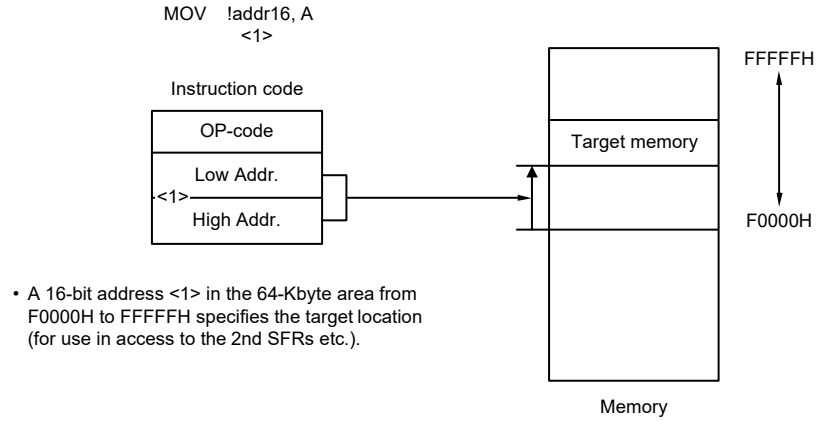
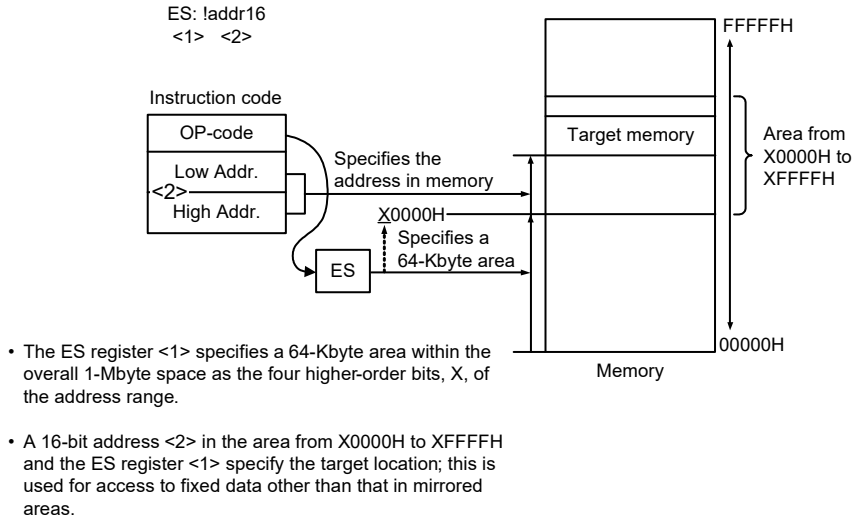


Figure 3 - 18 Example of ES:!addr16



3.4.4 Short direct addressing

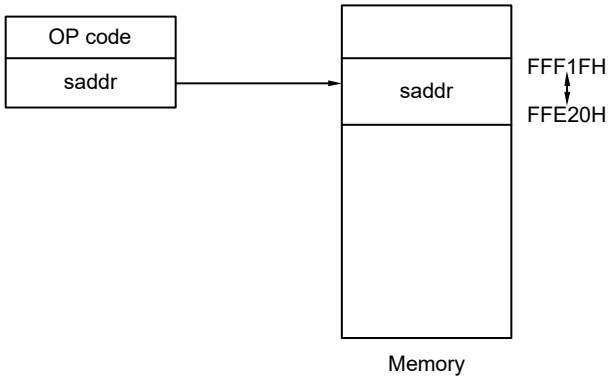
[Function]

Short direct addressing directly specifies the target addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFE20H to FFF1FH.

[Operand format]

Identifier	Description
SADDR	Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (only the space from FFE20H to FFF1FH is specifiable)
SADDRP	Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (even address only) (only the space from FFE20H to FFF1FH is specifiable)

Figure 3 - 19 Outline of Short Direct Addressing



Remark SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data.
Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

3.4.5 SFR addressing

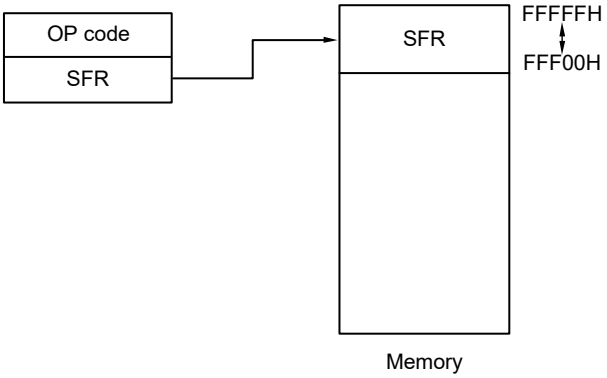
[Function]

SFR addressing directly specifies the target SFR addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFF00H to FFFFFH.

[Operand format]

Identifier	Description
SFR	SFR name
SFRP	16-bit-manipulatable SFR name (even address)

Figure 3 - 20 Outline of SFR Addressing



3.4.6 Register indirect addressing

[Function]

Register indirect addressing directly specifies the target addresses using the contents of the register pair specified with the instruction word as an operand address.

[Operand format]

Identifier	Description
—	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 21 Example of [DE], [HL]

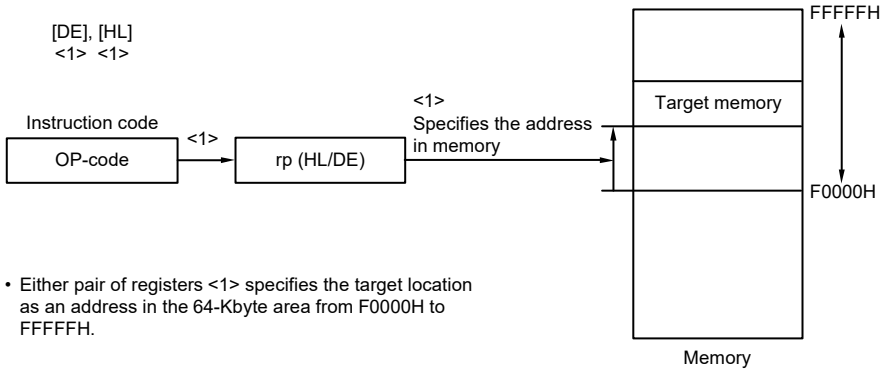
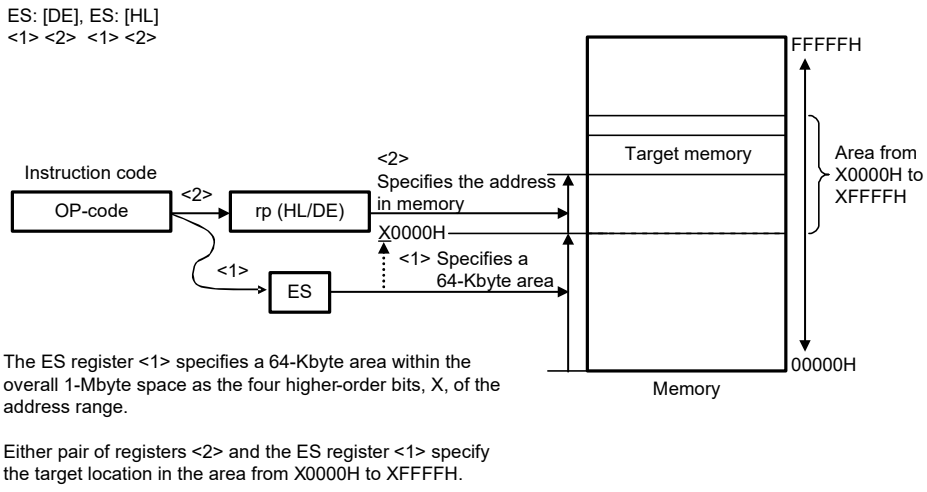


Figure 3 - 22 Example of ES:[DE], ES:[HL]



3.4.7 Based addressing

[Function]

Based addressing uses the contents of a register pair specified with the instruction word or 16-bit immediate data as a base address, and 8-bit immediate data or 16-bit immediate data as offset data. The sum of these values is used to specify the target address.

[Operand format]

Identifier	Description
—	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
—	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
—	word[BC] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[HL + byte], ES:[DE + byte] (higher 4-bit addresses are specified by the ES register)
—	ES:word[B], ES:word[C] (higher 4-bit addresses are specified by the ES register)
—	ES:word[BC] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 23 Example of [SP+byte]

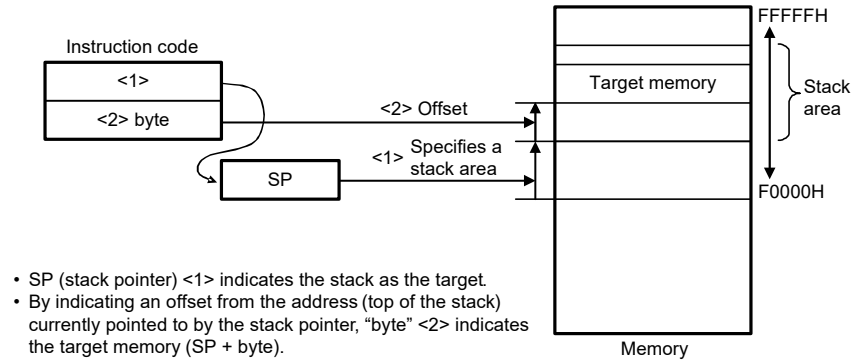


Figure 3 - 24 Example of [HL + byte], [DE + byte]]

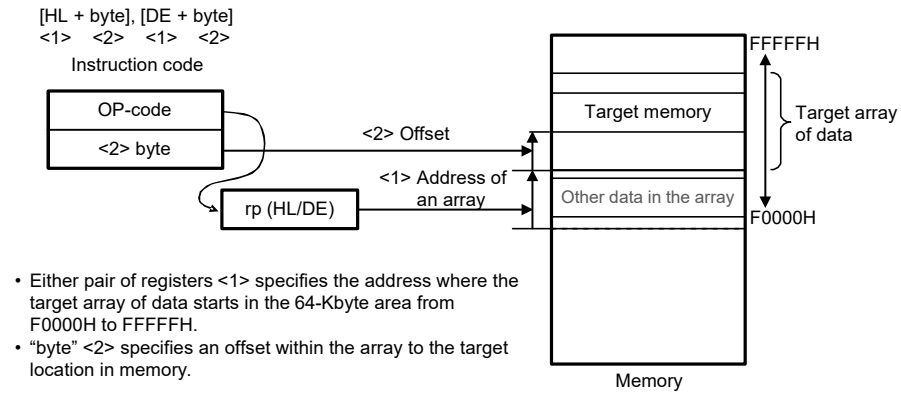
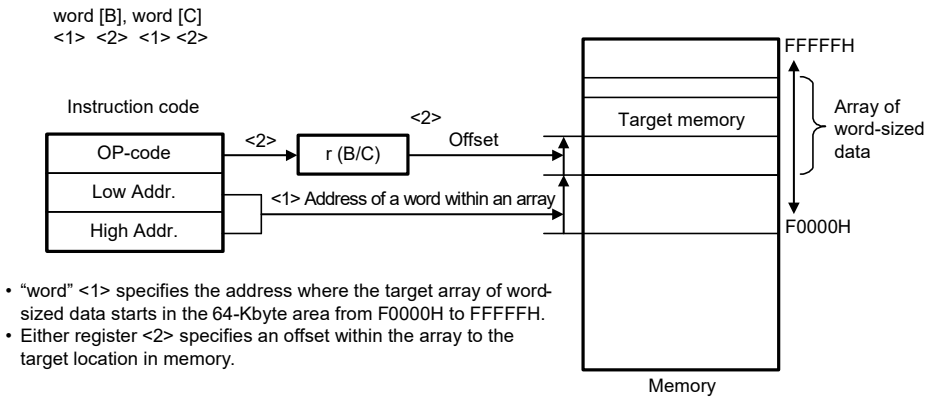
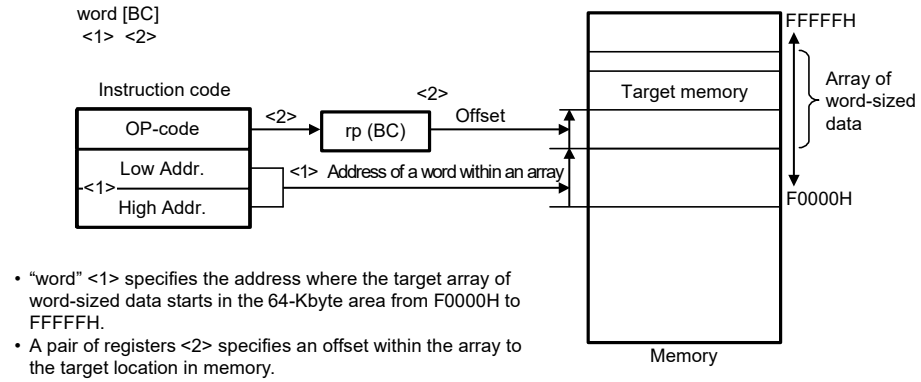


Figure 3 - 25 Example of word [B], word [C]



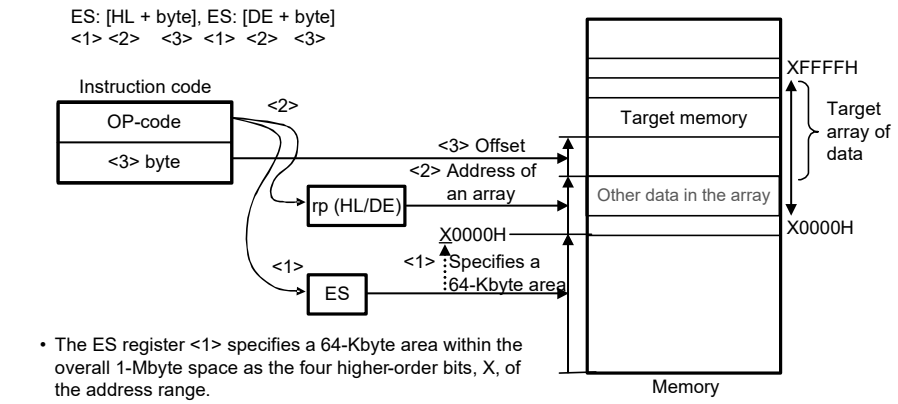
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3 - 26 Example of word [BC]



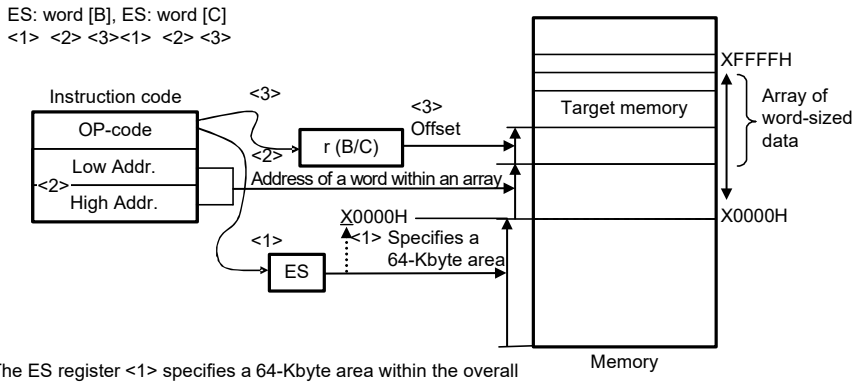
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

Figure 3 - 27 Example of [HL + byte], [DE + byte]]



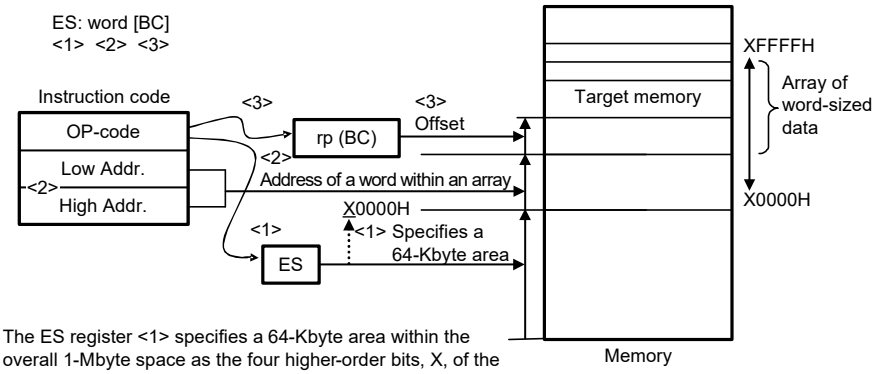
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- "byte" <3> specifies an offset within the array to the target location in memory.

Figure 3 - 28 Example of word [B], word [C]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- "word" <2> specifies the address where the target array of word-sized data starts in the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

Figure 3 - 29 Example of word [BC]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- "word" <2> specifies the address where the target array of word-sized data starts in the 64-Kbyte area specified in the ES register <1>.
- A pair of registers <3> specifies an offset within the array to the target location in memory.

3.4.8 Based indexed addressing

[Function]

Based indexed addressing uses the contents of a register pair specified with the instruction word as the base address, and the content of the B register or C register similarly specified with the instruction word as offset address. The sum of these values is used to specify the target address.

[Operand format]

Identifier	Description
—	[HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 30 Example of [HL + B], [HL + C]

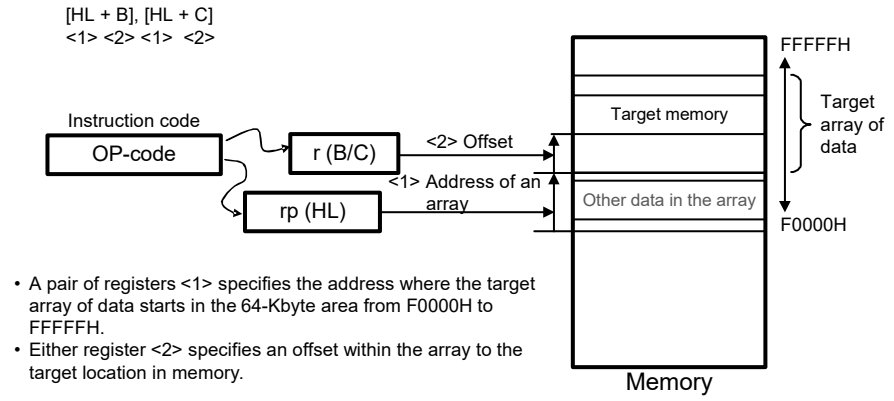
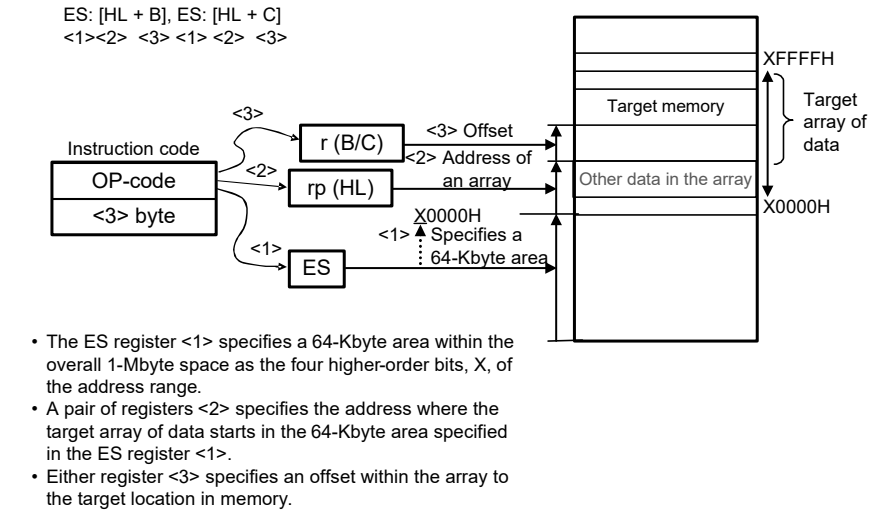


Figure 3 - 31 Example of ES:[HL + B], ES:[HL + C]



3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) values. This addressing is automatically employed when the PUSH, POP, subroutine call, and return instructions are executed or the register is saved/restored upon generation of an interrupt request.

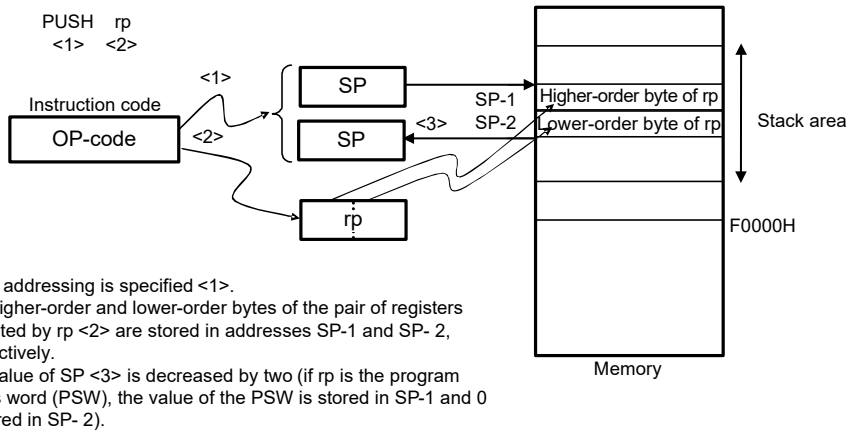
Only the internal RAM area can be set as the stack area.

[Operand format]

Identifier	Description
—	PUSH PSW AX/BC/DE/HL POP PSW AX/BC/DE/HL CALL/CALLT RET BRK RETB (Interrupt request generated) RETI

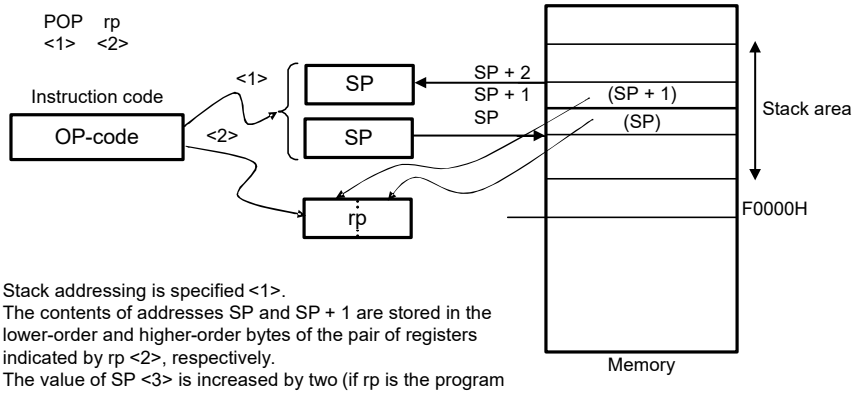
The data to be saved/restored by each stack operation is shown in Figures 3 - 32 to 3 - 37.

Figure 3 - 32 Example of PUSH rp



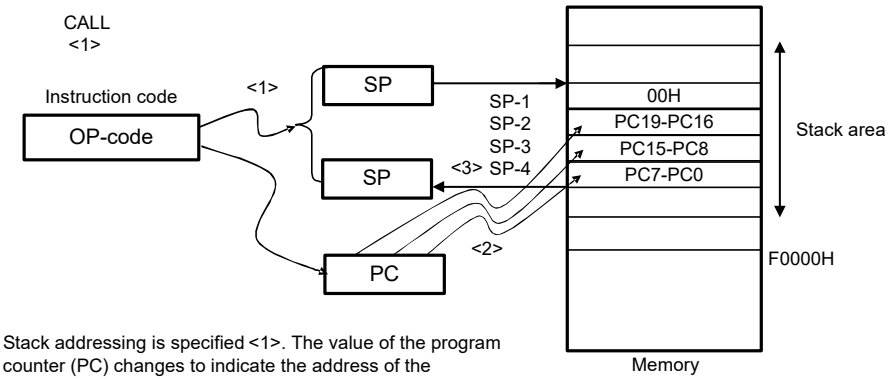
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP-1 and SP- 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP-1 and 0 is stored in SP- 2).

Figure 3 - 33 Example of POP



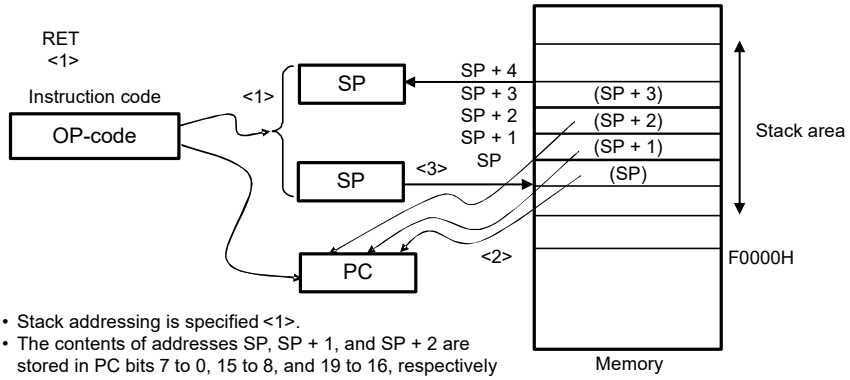
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP + 1 is stored in the PSW).

Figure 3 - 34 Example of CALL, CALLT



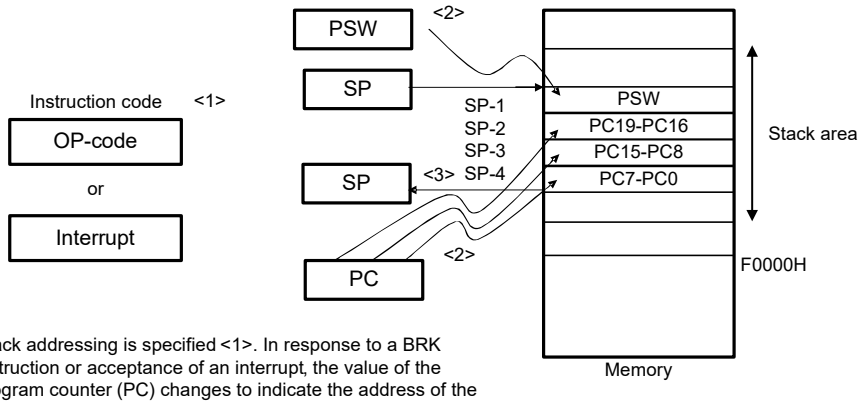
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP-1, SP-2, SP-3, and SP-4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3 - 35 Example of RET



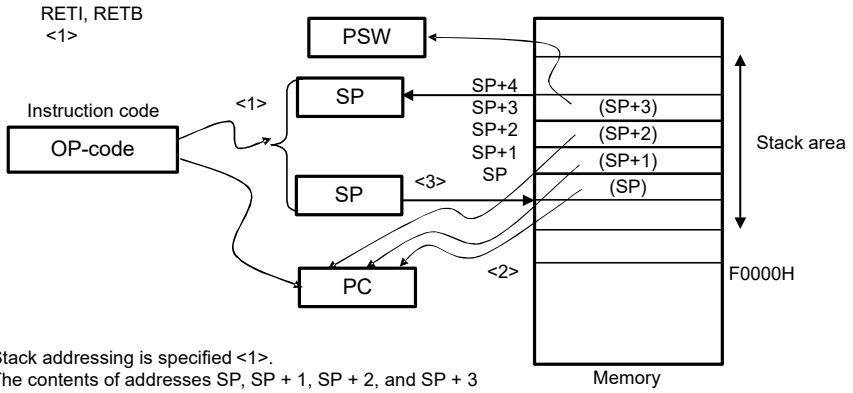
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3 - 36 Example of Interrupt, BRK



- Stack addressing is specified <1>. In response to a BRK instruction or acceptance of an interrupt, the value of the program counter (PC) changes to indicate the address of the next instruction.
- The values of the PSW, PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP-1, SP-2, SP-3, and SP-4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3 - 37 Example of RETI, RETB



- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, SP + 2, and SP + 3 are stored in PC bits 7 to 0, 15 to 8, 19 to 16, and the PSW, respectively <2>.
- The value of SP <3> is increased by four.

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The R9A02G015 is provided with digital I/O ports, which enable variety of control operations.
In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 2 PIN FUNCTIONS**.

4.2 Port Configuration

Ports include the following hardware.

Table 4 - 1 Port Configuration

Item	Configuration
Control registers	Port mode registers (PM0, PM2, PM4 to PM7) Port registers (P0, P2, P4 to 7, P12, P13) Pull-up resistor option registers (PU0, PU4, PU5, PU7) Port input mode registers (PIM0, PIM5, PIM7) Port output mode registers (POM0, POM5, POM7) Port mode control registers (PMC0, PMC2)
Port	• 32-pin product (with USB) Total: 23 (CMOS I/O: 15, CMOS input: 3, N-ch open drain I/O: 5) • 32-pin product (without USB) Total: 28 (CMOS I/O: 20, CMOS input: 3, N-ch open drain I/O: 5)
Pull-up resistor	• 32-pin product (with USB) : 9 • 32-pin product (without USB): 14

4.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 and P01 are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).
To use P00 and P01 as digital input/output pins, set them in the digital I/O mode by using port mode control register 0 (PMC0) (can be specified in 1-bit units).
This port can also be used for external interrupt request input, timer I/O, A/D converter analog input, and buzzer output.
Reset signal generation sets port 0 to analog input port.

4.2.2 Port 2

Port 2 is an I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units using port mode register 2 (PM2).
This port can also be used for A/D converter analog input.
To use P20 to P25 as analog input pins, set them to analog input using port mode control register 2 (PMC2) (can be specified in 1-bit units).
Reset signal generation sets port 2 to analog input port.

4.2.3 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).
This port can also be used for data I/O for a flash memory programmer/debugger.
Reset signal generation sets port 4 to input port.

4.2.4 Port 5

Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P55 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).
Input to the P50 to P55 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 5 (PIM5).
Output from the P50 to P55 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 5 (POM5).
This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, programming UART transmission/reception, timer I/O, and interface with USB connector.
Reset signal generation sets port 5 to input port.

4.2.5 Port 6

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6).
The output of the P60 to P64 pins is N-ch open-drain output (6 V tolerance).
This port can also be used for external interrupt request input, serial interface data I/O, and clock I/O.
Reset signal generation sets port 6 to input port.

4.2.6 Port 7

Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P74 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).
Input to the P70 to P74 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 7 (PIM7).
Output from the P70 to P74 pins can be specified as N-ch open-drain output (VDD tolerance) in 1-bit units using port output mode register 7 (POM7).
This port can also be used for external interrupt request input, timer I/O, serial interface data I/O, and clock I/O.
Reset signal generation sets port 7 to input port.

4.2.7 Port 12

P121 and P122 are 2-bit input ports.
This port can also be used for connecting a resonator for the main system clock and external clock input for the main system clock.
Reset signal generation sets P121 and P122 to input port.

4.2.8 Port 13

P137 is a 1-bit input-only port.

P137 is fixed an input ports.

This port can also be used for external interrupt request input.

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control registers (PMCxx)

Caution Which registers and bits are included depends on the product. For registers and bits mounted on each product, see Tables 4 - 2. Be sure to set bits that are not mounted to their initial values.

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (1/2)

Port		Bit Name						Product with USB	Product without USB
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register	32-pin	32-pin
Port 0	0	PM00	P00	PU00	PIM00	POM00	PMC00	✓	✓
	1	PM01	P01	PU01	PIM01	POM01	PMC01	✓	✓
	2	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
Port 2	0	PM20	P20	—	—	—	PMC20	✓	✓
	1	PM21	P21	—	—	—	PMC21	✓	✓
	2	PM22	P22	—	—	—	PMC22	✓	✓
	3	PM23	P23	—	—	—	PMC23	✓	✓
	4	PM24	P24	—	—	—	PMC24	✓	✓
	5	PM25	P25	—	—	—	PMC25	✓	✓
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
Port 4	0	PM40	P40	PU40	—	—	—	✓	✓
	1	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (2/2)

Port		Bit Name						Product with USB	Product without USB
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register	32-pin	32-pin
Port 5	0	PM50	P50	PU50	PIM50	POM50	—	✓	✓
	1	PM51	P51	PU51	PIM51	POM51	—	✓	✓
	2	PM52	P52	PU52	PIM52	POM52	—	✓	✓
	3	PM53	P53	PU53	PIM53	POM53	—	✓	✓
	4	PM54	P54	PU54	PIM54	POM54	—	✓	✓
	5	PM55	P55	PU55	PIM55	POM55	—	✓	✓
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
Port 6	0	PM60	P60	—	—	—	—	✓	✓
	1	PM61	P61	—	—	—	—	✓	✓
	2	PM62	P62	—	—	—	—	✓	✓
	3	PM63	P63	—	—	—	—	✓	✓
	4	PM64	P64	—	—	—	—	✓	✓
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
Port 7	0	PM70	P70	PU70	PIM70	POM70	—	—	✓
	1	PM71	P71	PU71	PIM71	POM71	—	—	✓
	2	PM72	P72	PU72	PIM72	POM72	—	—	✓
	3	PM73	P73	PU73	PIM73	POM73	—	—	✓
	4	PM74	P74	PU74	PIM74	POM74	—	—	✓
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
Port 12	0	—	—	—	—	—	—	—	—
	1	—	P121	—	—	—	—	✓	✓
	2	—	P122	—	—	—	—	✓	✓
	3	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
Port 13	0	—	—	—	—	—	—	—	—
	1	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	—	✓	✓

4.3.1 Port mode registers (PMxx)

These registers specify input or output mode for the port in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register by referencing **4.5 Register Settings When Using Alternate Function**.

Figure 4 - 1 Format of Port mode register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FFF20H	FFH	R/W
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	PM64	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	1	1	1	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 2, 4 to 7; n = 0 to 5)
0	Output mode (the pin functions as an output port (output buffer on))
1	Input mode (the pin functions as an input port (output buffer off))

Caution Be sure to set bits that are not mounted to their initial values.

4.3.2 Port registers (Pxx)

These registers set the output latch value of a port.

If the data is read in the input mode, the pin level is read. If it is read in the output mode, the output latch value is read Note.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Note When P00, P01, and P20 to P25 are set to the analog function, if a port is read in input mode, the read value is always 0, not the pin level.

Figure 4 - 2 Format of Port register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	0	P01	P00	FFF00H	00H (output latch)	R/W
P2	0	0	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P4	0	0	0	0	0	0	0	P40	FFF04H	00H (output latch)	R/W
P5	0	0	P55	P54	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	P64	P63	P62	P61	P60	FFF06H	00H (output latch)	R/W
P7	0	0	0	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P12	0	0	0	0	0	P122	P121	0	FFF0CH	Undefined	R/W Note 1
P13	P137	0	0	0	0	0	0	0	FFF0DH	Note 2	R/W Note 1

Pmn	m = 0, 2, 4 to 7, 12, 13; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note 1. P121, P122, and P137 are read-only.

Note 2. P137: Undefined

Caution Be sure to set bits that are not mounted to their initial values.

4.3.3 Pull-up resistor option registers (PUxx)

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode (PMmn = 1 and POMmn = 0) for the pins to which the use of an on-chip pull-up resistor has been specified in these registers. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of these registers. Similarly, on-chip pull-up resistors cannot be connected to the pins used as alternate-function output pins and the pins set to the analog function.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H (Only PU4 is set to 01H).

Figure 4 - 3 Format of Pull-up resistor option register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0	PU01	PU00	F0030H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU7	0	0	0	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 4, 5, 7; n = 0 to 5)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Caution Be sure to set bits that are not mounted to their initial values.

4.3.4 Port input mode registers (PIMxx)

These registers set the input buffer in 1-bit units.

TTL input buffer can be selected during serial communication with an external device of the different potential.

Port input mode registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 4 - 4 Format of Port input mode register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	0	0	PIM01	PIM00	F0040H	00H	R/W
PIM5	0	0	PIM55	PIM54	PIM53	PIM52	PIM51	PIM50	F0045H	00H	R/W
PIM7	0	0	0	PIM74	PIM73	PIM72	PIM71	PIM70	F0047H	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 5, 7; n = 0 to 5)
0	Normal input buffer
1	TTL input buffer

Caution Be sure to set bits that are not mounted to their initial values.

4.3.5 Port output mode registers (POMxx)

These registers set the output mode in 1-bit units.

N-ch open-drain output (V_{DD} tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA00 and SDA01 pins during simplified I^2C communication with an external device of the same potential.

In addition, POMxx register is set with PUxx register, whether or not to use the on-chip pull-up resistor.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open drain output (V_{DD} tolerance) mode is set.

Figure 4 - 5 Format of Port output mode register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	0	0	0	POM01	POM00	F0050H	00H	R/W
POM5	0	0	POM55	POM54	POM53	POM52	POM51	POM50	F0055H	00H	R/W
POM7	0	0	0	POM74	POM73	POM72	POM71	POM70	F0057H	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 5, 7; n = 0 to 5)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

Caution Be sure to set bits that are not mounted to their initial values.

4.3.6 Port mode control registers (PMCxx)

These registers set the P00, P01, P20 to P25 digital I/O/analog input in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to FFH.

Figure 4 - 6 Format of Port mode control register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	1	1	1	1	1	1	PMC01	PMC00	F0060H	FFH	R/W
PMC2	1	1	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20	F0062H	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 0, 2; n = 0 to 5)
0	Digital I/O (alternate function other than analog input)
1	Analog input

Caution Be sure to set bits that are not mounted to their initial values.

4.4 Port Function Operations

Port operations differ depending on whether the input or output mode is set, as shown below.

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.

Once data is written to the output latch, it is retained until data is written to the output latch again.

The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change.

Once data is written to the output latch, it is retained until data is written to the output latch again.

The data of the output latch is cleared when a reset signal is generated.

4.4.2 Reading from I/O port

(1) Output mode

The output latch contents are read by a transfer instruction. The output latch contents do not change.

(2) Input mode

The pin status is read by a transfer instruction. The output latch contents do not change.

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.

Once data is written to the output latch, it is retained until data is written to the output latch again.

The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change. Therefore, byte data can be written to the ports used for both input and output.

The data of the output latch is cleared when a reset signal is generated.

4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers

It is possible to connect an external device operating on a different potential (1.8 V, 2.5 V) by switching I/O buffers with the port input mode register (PIMxx) and port output mode register (POMxx).
When receiving input from an external device with a different potential (1.8 V, 2.5 V), set the port input mode registers 0, 5, and 7 (PIM0, PIM5, and PIM7) on a bit-by-bit basis to enable normal input (CMOS)/TTL input buffer switching.
When outputting data to an external device with a different potential (1.8 V, 2.5 V), set the port output mode registers 0, 5, and 7 (POM0, POM5, and POM7) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain (V_{DD} tolerance) switching.

The connection of a serial interface is described in the following.

(1) Setting procedure when using input pins of UART0, CSI00, and CSI01 functions for the TTL input buffer

In case of UART0:	P51
In case of CSI00:	P50, P51
In case of CSI01:	P53, P54

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> Set the corresponding bit of the PIM0, PIM5, and PIM7 registers 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/CSI mode.

(2) Setting procedure when using output pins of UART0, CSI00, and CSI01 functions in N-ch open-drain output mode

In case of UART0:	P52
In case of CSI00:	P50, P52
In case of CSI01:	P53, P55

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM0, POM5, and POM7 registers to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Enable the operation of the serial array unit and set the mode to the UART/CSI mode.
<6> Set the corresponding bit of the PM0, PM5, and PM7 registers to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

(3) Setting procedure when using I/O pins of IIC00 and IIC01 functions with a different potential (1.8 V, 2.5 V)

In case of simplified IIC00: P50, P51
In case of simplified IIC01: P53, P54

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM0, POM5 and POM7 registers to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Set the corresponding bit of the PIM0, PIM5 and PIM7 registers 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.
<6> Enable the operation of the serial array unit and set the mode to the simplified I^2C mode.
<7> Set the corresponding bit of the PM0, PM5 and PM7 registers 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.

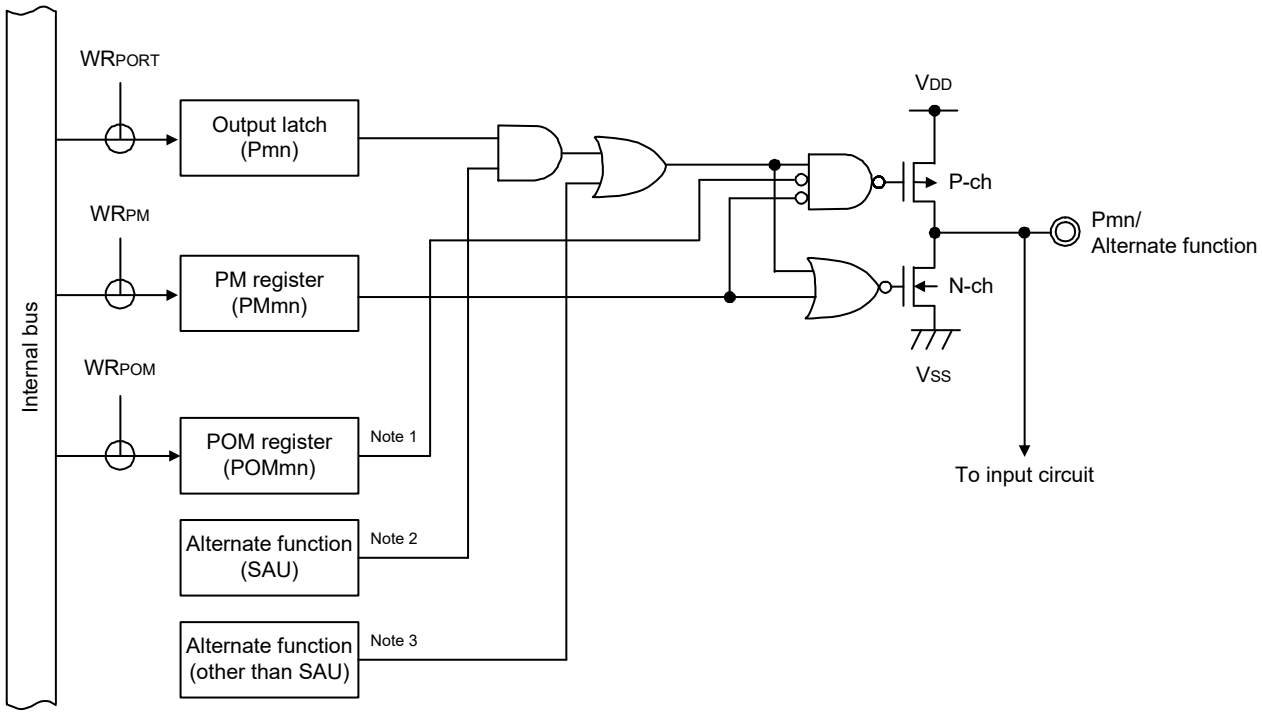
4.5 Register Settings When Using Alternate Function

4.5.1 Basic concept when using alternate function

In the beginning, for a pin also assigned to be used for analog function, use the port mode control register (PMCxx) to specify whether to use the pin for analog function or digital input/output.

Figure 4 - 7 shows the basic configuration of an output circuit for pins used for digital input/output. The output of the output latch for the port and the output of the alternate SAU function are input to an AND gate. The output of the AND gate is input to an OR gate. The output of an alternate function other than SAU (Timer, clock/buzzer output, etc.) is connected to the other input pin of the OR gate. When such kind of pins are used by the port function or an alternate function, the unused alternate function must not hinder the output of the function to be used. An idea of basic settings for this kind of case is shown in Table 4 - 3.

Figure 4 - 7 Basic Configuration of Output Circuit for Pins



Note 1. When there is no POM register, this signal should be considered to be low level (0).
Note 2. When there is no alternate function, this signal should be considered to be high level (1).
Note 3. When there is no alternate function, this signal should be considered to be low level (0).

Table 4 - 3 Concept of Basic Settings

Output Function of Used Pin	Output Settings of Unused Alternate Function		
	Output Function for Port	Output Function for SAU	Output Function for other than SAU
Output function for port	—	Output is high (1)	Output is low (0)
Output function for SAU	High (1)	—	Output is low (0)
Output function for other than SAU	Low (0)	Output is high (1)	Output is low (0) Note

Note Since more than one output function other than SAU may be assigned to a single pin, the output of an unused alternate function must be set to low level (0). For details on the setting method, see **4.5.2 Register settings for alternate function whose output function is not used**.

4.5.2 Register settings for alternate function whose output function is not used

When the output of an alternate function of the pin is not used, the following settings should be made.

(1) SOp = 1, TxDq = 1 (settings when the serial output (SOp/TxDq) of SAU is not used)
When the serial output (SOp/TxDq) is not used, such as, a case in which only the serial input of SAU is used, set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled) and set the SOmn bit in serial output register m (SOm) to 1 (high). These are the same settings as the initial state.

(2) SCKp = 1, SDAr = 1, SCLr = 1 (settings when channel n in SAU is not used)
When SAU is not used, set bit n (SEmn) in serial channel enable status register m (SEm) to 0 (operation stopped state), set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled), and set the SOmn and CKOmn bits in serial output register m (SOm) to 1 (high). These are the same settings as the initial state.

(3) TOmn = 0 (settings when the output of channel n in TAU is not used)
When the TOmn output of TAU is not used, set the bit in timer output enable register 0 (TOE0) which corresponds to the unused output to 0 (output disabled) and set the bit in timer output register 0 (TO0) to 0 (low). These are the same settings as the initial state.

(4) SDAAn = 0, SCLAn = 0 (setting when IICA is not used)
When IICA is not used, set the IICEn bit in IICA control register n0 (IICCTLn0) to 0 (operation stopped). This is the same setting as the initial state.

(5) PCLBUZ0 = 0 (setting when clock/buzzer output is not used)
When the clock/buzzer output is not used, set the PCLOEn bit in clock output select register n (CKSn) to 0 (output disabled). This is the same setting as the initial state.

4.5.3 Register setting examples for used port and alternate functions

Register setting examples for used port and alternate functions are shown in Tables 4 - 4 to 4 - 11. The registers used to control the port functions should be set as shown in Tables 4 - 4 to 4 - 11. See the following remark for legends used in Tables 4 - 4 to 4 - 11.

Remark —: Not supported
×: Don't care
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch

Table 4 - 4 Setting Examples of Registers When Using P00 to P01 Pin Function

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	
	Function Name	I/O					SAU Output Function	Other than SAU	with USB	without USB
P00	P00	Input	×	0	1	×	—	×	✓	✓
		Output	0	0	0	0/1	—	TO05 = 0 PCLBUZ0 = 0		
		N ch-OD Output	1	0	0	0/1	—			
	INTP12	Input	×	0	1	×	—	×	✓	✓
	TI05	Input	×	0	1	×	—	×	✓	✓
	TO05	Output	0	0	0	0	—	PCLBUZ0 = 0	✓	✓
	ANI16	Analog input	×	1	1	×	—	×	✓	✓
	PCLBUZ0	Output	×	0	0	0/1	—	TO05 = 0	✓	✓
P01	P01	Input	×	0	1	×	—	×	✓	✓
		Output	0	0	0	0/1	—	TO06 = 0		
		N ch-OD Output	1	0	0	0/1	—			
	INTP13	Input	×	0	1	×	—	×	✓	✓
	TI06	Input	×	0	1	×	—	×	✓	✓
	TO06	Output	0	0	0	0	—	×	✓	✓
	ANI17	Analog input	×	1	1	×	—	×	✓	✓

Table 4 - 5 Setting Examples of Registers When Using P20 to 25 Pin Function

Pin Name	Used Function		ADM2	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	
	Function Name	I/O						SAU Output Function	Other than SAU	with USB	without USB
P20	P20	Input	×	—	0	1	×	—	—	✓	✓
		Output	×	—	0	0	0/1	—	—		
	ANI0	Analog input	00x0xx0x, 10x0xx0x	—	1	1	×	—	—	✓	✓
	AVREFP	Reference power supply	01x0xx0x	—	1	1	×	—	—	✓	✓
P21	P21	Input	×	—	0	1	×	—	—	✓	✓
		Output	×	—	0	0	0/1	—	—		
	ANI1	Analog input	xx00xx0x	—	1	1	×	—	—	✓	✓
	AVREFM	Reference power supply	xx10xx0x	—	1	1	×	—	—	✓	✓
P22	P22	Input	—	—	0	1	×	—	—	✓	✓
		Output	—	—	0	0	0/1	—	—		
	ANI2	Analog input	—	—	1	1	×	—	—	✓	✓
P23	P23	Input	—	—	0	1	×	—	—	✓	✓
		Output	—	—	0	0	0/1	—	—		
	ANI3	Analog input	—	—	1	1	×	—	—	✓	✓
P24	P24	Input	—	—	0	1	×	—	—	✓	✓
		Output	—	—	0	0	0/1	—	—		
	ANI4	Analog input	—	—	1	1	×	—	—	✓	✓
P25	P25	Input	—	—	0	1	×	—	—	✓	✓
		Output	—	—	0	0	0/1	—	—		
	ANI5	Analog input	—	—	1	1	×	—	—	✓	✓

Table 4 - 6 Setting Examples of Registers When Using P40 Pin Function

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	
	Function Name	I/O					SAU Output Function	Other than SAU	with USB	without USB
P40	P40	Input	—	—	1	×	—	—	✓	✓
		Output	—	—	0	0/1	—	—		

Table 4 - 7 Setting Examples of Registers When Using P50 to P55 Pin Function (1/2)

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	
	Function Name	I/O					SAU Output Function	Other than SAU	with USB	without USB
P50	P50	Input	0	—	1	×	×	—	✓	✓
		Output	0	—	0	0/1	SCK00/SCL00 = 1	—		
		N ch-OD output	1	—	0	0/1		—		
	INTP6	Input	×	—	1	×	×	—	✓	✓
	SCL00	Output	0/1	—	0	1	×	—	✓	✓
	SCK00	Input	×	—	1	×	×	—	✓	✓
		Output	0/1	—	0	1	×	—		
P51	P51	Input	0	—	1	×	×	×	✓	✓
		Output	0	—	0	0/1	SDA00 = 1	TO00 = 0		
		N ch-OD output	1	—	0	0/1				
	INTP7	Input	×	—	1	×	×	×	✓	✓
	TI00	Input	×	—	1	×	×	×	✓	✓
	TO00	Output	0	—	0	0	SDA00 = 1	×	✓	✓
	SDA00	I/O	1	—	0	1	×	TO00 = 0	✓	✓
	SI00	Input	×	—	1	×	×	×	✓	✓
	RXD0	Input	×	—	1	×	×	×	✓	✓
P52	P52	Input	0	—	1	×	×	×	✓	✓
		Output	0	—	0	0/1	SO00 = 1 TXD0 = 1	TO01 = 0 UVBUSEN1 = 0		
		N ch-OD output	1	—	0	0/1				
	INTP8	Input	×	—	1	×	×	×	✓	✓
	TI01	Input	×	—	1	×	×	×	✓	✓
	TO01	Output	0	—	0	0	SO00 = 1	×	✓	✓
	SO00	Output	0/1	—	0	1	×	TO01 = 0 UVBUSEN1 = 0	✓	✓
	TXD0	Output	0/1	—	0	1	SO00 = 1		✓	✓
	UVBUSEN1	Output	—	—	1	×		TO01 = 0	✓	×
P53	P53	Input	0	—	1	×	×	×	✓	✓
		Output	0	—	0	0/1	SCK01/SCL01 = 1	TO02 = 0		
		N ch-OD output	1	—	0	0/1				
	INTP9	Input	×	—	1	×	×	×	✓	✓
	TI02	Input	×	—	1	×	×	×	✓	✓
	TO02	Output	0	—	0	0	SCK01/SCL01 = 1	×	✓	✓
	SCL01	Output	0/1	—	0	1	×	TO02 = 0	✓	✓
	SCK01	Input	×	—	1	×	×	×	✓	✓
		Output	0/1	—	0	1	×	TO02 = 0		
	UVOVRCUR1	Input	×	—	1	×	×	×	✓	×
P54	P54	Input	0	—	1	×	×	×	✓	✓
		Output	0	—	0	0/1	SDA01 = 1	TO03 = 0 UVBUSEN0 = 0		
		N ch-OD output	1	—	0	0/1				
	INTP10	Input	×	—	1	×	×	×	✓	✓
	TI03	Input	×	—	1	×	×	×	✓	✓
	TO03	Output	0	—	0	0	SDA01 = 1	UVBUSEN0 = 0	✓	✓
	SDA01	I/O	1	—	0	1	×	TO03 = 0 UVBUSEN0 = 0	✓	✓
	SI01	Input	×	—	1	×	×	×	✓	✓
	UVBUSEN0	Output	—	—	1	×	SDA01 = 1	TO03 = 0	✓	×

Table 4 - 7 Setting Examples of Registers When Using P50 to P55 Pin Function (2/2)

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	
	Function Name	I/O					SAU Output Function	Other than SAU	with USB	without USB
P55	P55	Input	0	—	1	×	×	×	✓	✓
		Output	0	—	0	0/1	SO01 = 1	TO04 = 0		
		N ch-OD output	1	—	0	0/1				
	INTP11	Input	×	—	1	×	×	×	✓	✓
	TI04	Input	×	—	1	×	×	×	✓	✓
	TO04	Output	0	—	0	0	SO01 = 1	×	✓	✓
	SO01	Output	0/1	—	0	1	×	TO04 = 0	✓	✓
	UVOVRCUR0	Input	×	—	1	×	SO01 = 1	×	✓	×

Table 4 - 8 Setting Examples of Registers When Using P60 to P64 Pin Function

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	
	Function Name	I/O					SAU Output Function	Other than SAU	with USB	without USB
P60	P60	Input	—	—	1	×	—	×	✓	✓
		Output	—	—	0	0/1	—	SCLA0 = 0		
	INTP1	Input	—	—	1	×	—	×	✓	✓
	SCLA0	I/O	—	—	0	0	—	×	✓	✓
P61	P61	Input	—	—	1	×	—	×	✓	✓
		Output	—	—	0	0/1	—	SDAA0 = 0		
	INTP2	Input	—	—	1	×	—	×	✓	✓
	SDAA0	I/O	—	—	0	0	—	×	✓	✓
P62	P62	Input	—	—	1	×	—	×	✓	✓
		Output	—	—	0	0/1	—	SCLA1 = 0		
	INTP3	Input	—	—	1	×	—	×	✓	✓
	SCLA1	I/O	—	—	0	0	—	×	✓	✓
P63	P63	Input	—	—	1	×	—	×	✓	✓
		Output	—	—	0	0/1	—	SDAA1 = 0		
	INTP4	Input	—	—	1	×	—	×	✓	✓
	SDAA1	I/O	—	—	0	0	—	×	✓	✓
P64	P64	Input	—	—	1	×	—	—	✓	✓
		Output	—	—	0	0/1	—	—		
	INTP5	Input	—	—	1	×	—	—	✓	✓

Table 4 - 9 Setting Examples of Registers When Using P70 to P74 Pin Function

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	
	Function Name	I/O					SAU Output Function	Other than SAU	with USB	without USB
P70	P70	Input	×	—	1	×	—	×	×	✓
		Output	0	—	0	0/1	—	SCLA2 = 0		
		N ch-OD output	1	—	0	0/1	—			
	SCLA2	I/O	1	—	0	0	—	×	×	✓
P71	P71	Input	×	—	1	×	—	×	×	✓
		Output	0	—	0	0/1	—	SDAA2 = 0		
		N ch-OD output	1	—	0	0/1	—			
	SDAA2	I/O	1	—	0	0	—	×	×	✓
P72	P72	Input	×	—	1	×	—	—	×	✓
		Output	0	—	0	0/1	—	—		
		N ch-OD output	1	—	0	0/1	—	—		
	INTP14	Input	×	—	1	×	—	—	×	✓
P73	P73	Input	×	—	1	×	—	—	×	✓
		Output	0	—	0	0/1	—	—		
		N ch-OD output	1	—	0	0/1	—	—		
	INTP15	Input	×	—	1	×	—	—	×	✓
P74	P74	Input	×	—	1	×	—	×	×	✓
		Output	0	—	0	0/1	—	TO07 = 0		
		N ch-OD output	1	—	0	0/1	—			
	TI07	Input	×	—	1	×	—	×	×	✓
	TO07	Output	0	—	0	0	—	×	×	✓

Table 4 - 10 Setting Examples of Registers When Using P121 to P122 Pin Function

Pin Name	Used Function		CMC (EXCLK, OSCSEL)	Pxx	32-pin	
	Function Name	I/O			with USB	without USB
P121	P121	Input	00/10/11	×	✓	✓
	X1	—	01	—	✓	✓
P122	P122	Input	00/10	×	✓	✓
	X2	—	01	—	✓	✓
	EXCLK	Input	11	×	✓	✓

Table 4 - 11 Setting Examples of Registers When Using P137 Pin Function

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	
	Function Name	I/O					SAU Output Function	Other than SAU	with USB	without USB
P137	P137	Input	—	—	—	×	—	—	✓	✓
	INTP0	Input	—	—	—	×	—	—	✓	✓

4.6 Cautions When Using Port Function

4.6.1 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn)

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit. Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

<Example> When P20 is an output port, P21 to P25 are input ports (all pin statuses are high level), and the port latch value of port 1 is 00H, if the output of output port P20 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 is 3FH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMnm bit is 1 are the output latch and pin status, respectively.

A 1-bit manipulation instruction is executed in the following order in the R9A02G015.

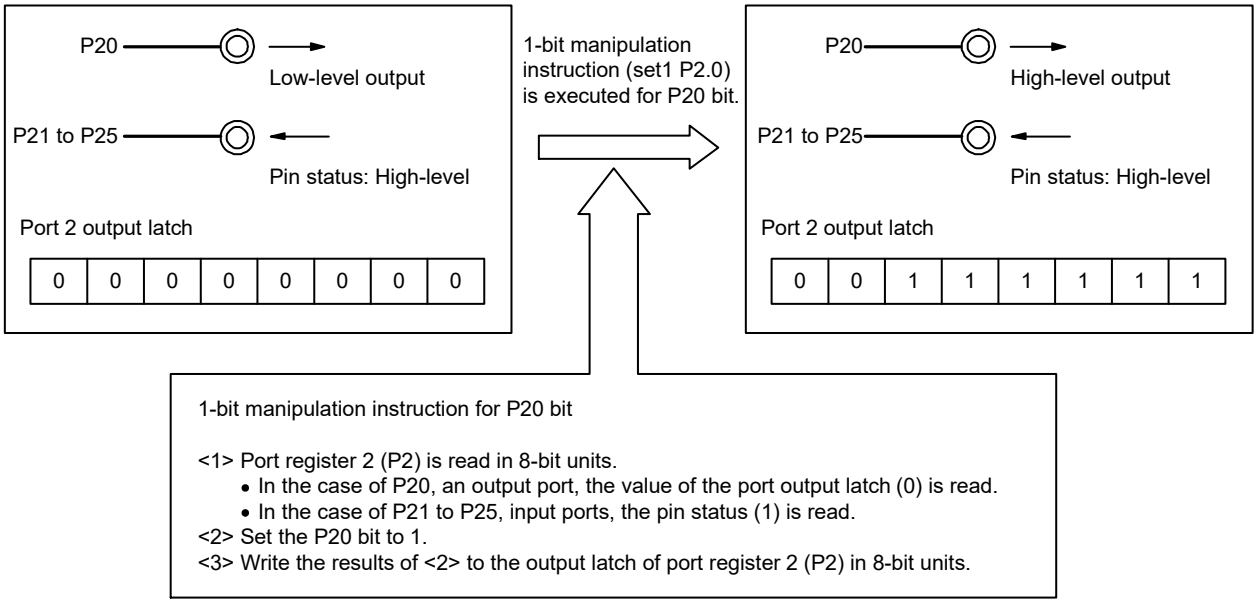
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P20, which is an output port, is read, while the pin statuses of P21 to P25, which are input ports, are read. If the pin statuses of P21 to P25 are high level at this time, the read value is 3EH.

The value is changed to 3FH by the manipulation in <2>.

3FH is written to the output latch by the manipulation in <3>.

Figure 4 - 8 Bit Manipulation Instruction (P20)



4.6.2 Notes on specifying the pin settings

For an output pin to which multiple functions are assigned, the output of the unused alternate functions must be set to its initial state so as to prevent conflicting outputs. For details about the alternate function output, see **4.5 Register Settings When Using Alternate Function**.
No specific setting is required for input pins because the output of their alternate functions is disabled (the buffer output is Hi-Z).

Disabling the unused functions, including blocks that are only used for input or do not have I/O, is recommended for lower power consumption.

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following two kinds of system clocks and clock oscillators are selectable.

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of fX = 1 to 20 MHz by connecting a resonator to X1 and X2 pins.
Oscillation can be stopped by executing the STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

<2> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from amoung fHOCO = 48, 24, 16, 12, 8, 6, 4, 3, 2 or 1MHz (TYP.) by using the option byte (000C2H). When selecting 48MHz for the high-speed on-chip oscillator clock fHOCO (FRQSEL4 = 1 in the option byte (000C2H)) and the CKSELR bit (bit 0 of the MCKC register) is set to 0, a clock generated by dividing the fHOCO frequency by 2, 4, or 8 (divided by 2 is the default) by setting the RDIV0 and RDIV1 bits (bits 1 and 2 of the MCKC register) is selected as the main system clock source (fIH). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock.
When selecting the high-speed on-chip oscillator as the USB clock (this setting is only available when using the USB function controller), be sure to select 48 MHz as the fHOCO frequency and then select the main system clock source (fIH) with fHOCO divided as the main system clock (fMAIN).
Oscillation can be stopped by executing the STOP instruction or setting the HIOSTOP bit (bit 0 of the CSC register).
The frequency specified by using an option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 5 - 10 Format of High-speed on-chip oscillator frequency select register (HOCODIV)**.

<3> High-speed system clock multiplication function using PLL (phase locked loop)

This clock function is mainly used for clock supply to the USB host/function controller. Set the DSCCTL register so that the PLL oscillation frequency (fPLL) is 48 MHz. When the CKSELR bit is set to 1, a clock generated by dividing the fPLL frequency by 2, 4, or 8 by setting the RDIV0 and RDIV1 bits is selected as the main system clock source (fIH). When selecting the PLL clock as the USB source clock, be sure to select the main system clock source (fIH) with fPLL divided as the main system clock (fMAIN).
The PLL can be operated or stopped by setting the DSCON bit (bit 0 in the DSCCTL register). For details on PLL settings and the connection with the USB clock, see **Figure 5 - 13** and **Table 5 - 3**.

An external main system clock (fEX = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of the MSTOP bit.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or high-speed on-chip oscillator clock or PLL clock can be selected by setting of the MCM0 bit (bit 4 of the system clock control register (CKC)).

(2) Low-speed on-chip oscillator clock (Low-speed on-chip oscillator)

This circuit oscillates a clock of f_{IL} = 15 kHz (TYP.).

The low-speed on-chip oscillator clock cannot be used as the CPU clock.

Only the following peripheral hardware runs on the low-speed on-chip oscillator clock.

- Watchdog timer
- 12-bit interval timer

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC), or both are set to 1.

However, when WDTON = 1, WUTMMCK0 = 0, and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, oscillation of the low-speed on-chip oscillator stops if the HALT or STOP instruction is executed.

Remark f_X: X1 clock oscillation frequency

f_{IH}: Main system clock source frequency divided by 2, 4, or 8 when the high-speed on-chip oscillator clock or the PLL clock is selected (24 MHz max.)

f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)

f_{EX}: External main system clock frequency

f_{IL}: Low-speed on-chip oscillator clock frequency

f_{PLL}: PLL oscillation frequency

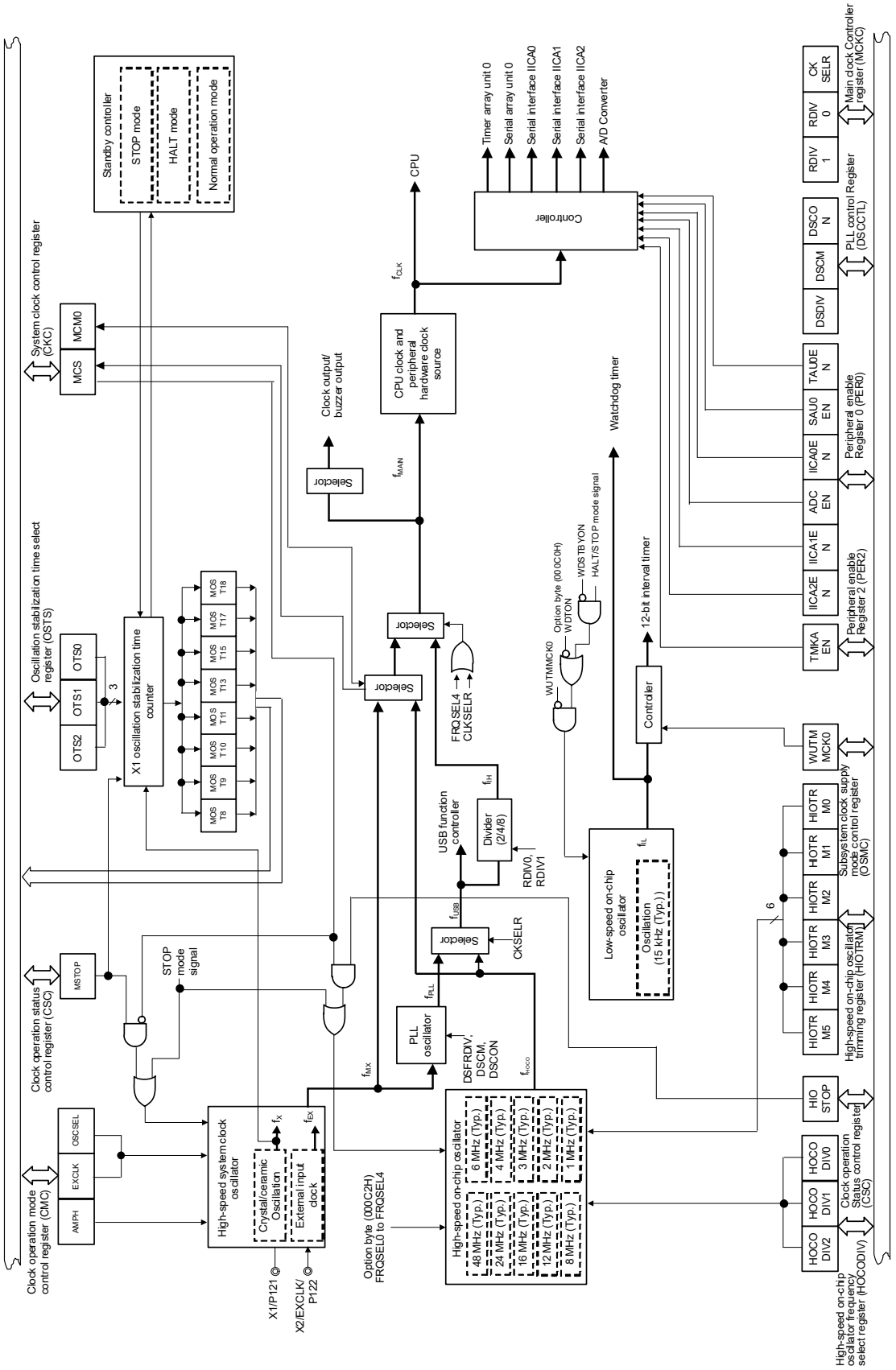
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5 - 1 Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) System clock control register (CKC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) Peripheral enable registers 0, 2 (PER0, PER2) Subsystem clock supply mode control register (OSMC) High-speed on-chip oscillator frequency select register (HOCODIV) High-speed on-chip oscillator trimming register (HIOTRM) PLL control register (DSCCTL) Main clock control register (MCKC) USB clock selection register (UCKSEL)
Oscillators	X1 oscillator High-speed on-chip oscillator Low-speed on-chip oscillator PLL oscillator

Figure 5 - 1 Block Diagram of Clock Generator



(**Note** and **Remark** are listed on the next page.)

Note Be sure to select f_{IH} when using the USB host/function controller.

Remark f_X: X1 clock oscillation frequency
f_{IH}: Main system clock source frequency when the high-speed on-chip oscillator clock divided 1, 2, 4, or 8, or the PLL clock divided by 2, 4, or 8 is selected (24 MHz max.)
f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{PLL}: PLL oscillation frequency
f_{USB}: USB clock frequency

5.3 Registers Controlling Clock Generator

The following registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable registers 0, 2 (PER0, PER2)
- Subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)
- PLL control register (DSCCTL)
- Main clock control register (MCKC)
- USB clock selection register (UCKSEL)

5.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121, X2/EXCLK/P122 pins, and to select a gain of the oscillator.

The CMC register can be written only once by an 8-bit memory manipulation instruction after reset release. This register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5 - 2 Format of Clock Operation Mode Control Register (CMC)

Address: FFFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	0	0	0	0	0	AMPH

EXCLK	OSCSEL	High-speed system clock pin operation mode	X1/P121 pin	X2/EXCLK/P122 pin
0	0	Input port mode	Input port	
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

AMPH	Control of X1 clock oscillation frequency
0	1 MHz ≤ fX ≤ 10 MHz
1	10 MHz < fX ≤ 20 MHz

Caution 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. When using the CMC register with its initial value (00H), be sure to set the register to 00H after a reset ends in order to prevent malfunction due to a program loop. Such a malfunction becomes unrecoverable when a value other than 00H is mistakenly written.

Caution 2. After reset release, set the CMC register before X1 oscillation is started as set by the clock operation status control register (CSC).

Caution 3. Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz.

Caution 4. Specify the settings for the AMPH bit while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX}).

Caution 5. Although the maximum system clock frequency is 24 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

Remark fX: X1 clock frequency

5.3.2 System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and a main system clock.

The CKC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 5 - 3 Format of System Clock Control Register (CKC)

Address: FFFA4H After reset: 00H R/W **Note 1**

Symbol	7	6	<5>	<4>	3	2	1	0
CKC	0	0	MCS	MCM0	0	0	0	0

MCS	Status of Main system clock (f_{MAIN})
0	High-speed on-chip oscillator clock (f_{HOCO})
1	High-speed system clock (f_{MX})

MCM0 **Note 2**	Main system clock (f_{MAIN}) operation control
0	Selects the high-speed on-chip oscillator clock (f_{HOCO}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

Note 1. Bit 5 is read-only.

Note 2. Don't set the MCM0 bit to 1 while the FRQSEL4 bit of the option byte (000C2H) or the CKSELR bit of the MCKC register is set to 1.

Caution Be sure to clear bit7, bit6 and bit3 to 0 to 0.

Remark f_{IH}: Main system clock source frequency divided by 2, 4, or 8 when the high-speed on-chip oscillator clock or the PLL clock is selected (24 MHz max.)

f_{MX}: High-speed system clock frequency

f_{MAIN}: Main system clock frequency

5.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock and high-speed on-chip oscillator.

The CSC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to C0H.

Figure 5 - 4 Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H After reset: C0H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
CSC	MSTOP	1	0	0	0	0	0	HIOSTOP

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Input port mode
0	X1 oscillator operating	External clock from EXCLK pin is valid	Input port
1	X1 oscillator stopped	External clock from EXCLK pin is invalid	

HIOSTOP	High-speed on-chip oscillator clock operation control
0	High-speed on-chip oscillator operating
1	High-speed on-chip oscillator stopped

Caution 1. After reset release, set the clock operation mode control register (CMC) before setting the CSC register.

Caution 2. Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after releasing reset. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.

Caution 3. To start X1 oscillation as set by the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).

Caution 4. Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the OSC register.

Caution 5. The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as Table 5 - 2.
Before stopping the clock oscillation, check the conditions before the clock oscillation is stopped.

Table 5 - 2 Stopping Clock Method

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock External main system clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock. (MCS = 0)	MSTOP = 1
High-speed on-chip oscillator clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed on-chip oscillator clock. (MCS = 1)	HIOSTOP = 1

5.3.4 Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.
The X1 clock oscillation stabilization time can be checked in the following case,

- If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.
When reset signal is generated, the STOP instruction and MSTOP (bit 7 of clock operation status control register (CSC)) = 1 clear the OSTC register to 00H.

Remark The oscillation stabilization time counter starts counting in the following cases.
- When oscillation of the X1 clock starts (EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- When the STOP mode is released

Figure 5 - 5 Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

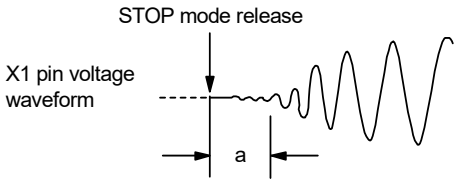
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204 μs min.	102 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819 μs min.	409 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.63 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.1 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	26.2 ms min.	13.1 ms min.

Caution 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

Caution 2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).
In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.**
  (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)

Caution 3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).



Remark fx: X1 clock oscillation frequency

5.3.5 Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time.

When the X1 clock is made to oscillate by clearing the MSTOP bit to start the X1 oscillation circuit operating, actual operation is automatically delayed for the time set in the OSTS register.

When switching the CPU clock from the high-speed on-chip oscillator clock to the X1 clock, and when using the high-speed on-chip oscillator clock for switching the X1 clock from the oscillating state to STOP mode, use the oscillation stabilization time counter status register (OSTC) to confirm that the desired oscillation stabilization time has elapsed after release from the STOP mode. That is, use the OSTC register to check that the oscillation stabilization time corresponding to its setting has been reached.

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

Reset signal generation sets the OSTS register to 07H.

Figure 5 - 6 Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				fx = 10 MHz	fx = 20 MHz
0	0	0	2^8/fx	25.6 μs	12.8 μs
0	0	1	2^9/fx	51.2 μs	25.6 μs
0	1	0	2^{10}/fx	102 μs	51.2 μs
0	1	1	2^{11}/fx	204 μs	102 μs
1	0	0	2^{13}/fx	819 μs	409 μs
1	0	1	2^{15}/fx	3.27 ms	1.63 ms
1	1	0	2^{17}/fx	13.6 ms	6.55 ms
1	1	1	2^{18}/fx	26.2 ms	13.1 ms

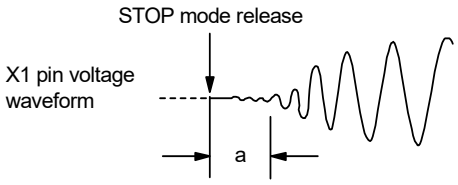
Caution 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

Caution 2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register.

In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)**

Caution 3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).



Remark fx: X1 clock oscillation frequency

5.3.6 Peripheral enable registers 0, 2 (PER0, PER2)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use the peripheral functions below, which are controlled by this register, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- 12-bit interval timer
- A/D converter
- Serial Interface IICA0
- Serial interface IICA1
- Serial interface IICA2 **Note**
- Serial array unit 0
- Timer array unit 0

The PER0 and PER2 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5 - 7 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	<2>	1	<0>
PER0	IICA2EN **Note**	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

IICA2EN Note	Control of serial interface IICA2 input clock supply
0	Stops input clock supply. • SFR used by the serial interface IICA2 cannot be written. • The serial interface IICA2 is in the reset status.
1	Enables input clock supply. • SFR used by the serial interface IICA2 can be read and written.

IICA1EN	Control of serial interface IICA1 input clock supply
0	Stops input clock supply. • SFR used by the serial interface IICA1 cannot be written. • The serial interface IICA1 is in the reset status.
1	Enables input clock supply. • SFR used by the serial interface IICA1 can be read and written.

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFR used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFR used by the A/D converter can be read and written.

IICA0EN	Control of serial interface IICA0 input clock supply
0	Stops input clock supply. • SFR used by the serial interface IICA0 cannot be written. • The serial interface IICA0 is in the reset status.
1	Enables input clock supply. • SFR used by the serial interface IICA0 can be read and written.

SAU0EN	Control of serial array unit 0 input clock supply
0	Stops input clock supply. • SFR used by the serial array unit 0 cannot be written. • The serial array unit 0 is in the reset status.
1	Enables input clock supply. • SFR used by the serial array unit 0 can be read and written.

TAU0EN	Control of timer array unit 0 input clock supply
0	Stops input clock supply. • SFR used by timer array unit 0 cannot be written. • Timer array unit 0 is in the reset status.
1	Enables input clock supply. • SFR used by timer array unit 0 can be read and written.

Note This function has not been mounted in the R9A02G0150 product.

Caution Be sure to clear bit 1, 3 to 0.

Figure 5 - 8 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER2	TMKAEN	0	0	0	0	0	0	0

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. • SFRs used by the 12-bit interval timer cannot be written. • The 12-bit interval timer is in the reset status.
1	Enables input clock supply. • SFRs used by the 12-bit interval timer can be read and written.

Caution Be Sure to clear bits 6 to 0 to 0.

5.3.7 Subsystem clock supply mode control register (OSMC)

The OSMC register can be used to select the count clock of the 12-bit interval timer.

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

Reset signal generation clears this register to 00H.

Figure 5 - 9 Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of count clock for 12-bit interval timer.
0	No clock supplied
1	Low-speed on-chip oscillator clock (f_{IL})

5.3.8 High-speed on-chip oscillator frequency select register (HOCODIV)

The frequency of the high-speed on-chip oscillator which is set by an option byte (000C2H) can be changed by using high-speed on-chip oscillator frequency select register (HOCODIV). However, the selectable frequency depends on the FRQSEL4 and FRQSEL3 bits of the option byte (000C2H).

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

Reset signal generation clears this register to the value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H).

Figure 5 - 10 Format of High-speed on-chip oscillator frequency select register (HOCODIV)

Address: F00A8H After reset: The value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H) R/W

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator clock frequency		
			FRQSEL4 = 0		FRQSEL4 = 1
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0
0	0	0	f_{IH} = 24 MHz	Setting prohibited	f_{IH} = 24/12/6 MHz Note 1 f_{HOCO} = 48 MHz Note 2
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz	f_{IH} = 12/6/3 MHz Note 1 f_{HOCO} = 24 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz	f_{IH} = 6/3 MHz Note 1 f_{HOCO} = 12 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz	f_{IH} = 3 MHz f_{HOCO} = 6 MHz
1	0	0	Setting prohibited	f_{IH} = 2 MHz	Setting prohibited
1	0	1	Setting prohibited	f_{IH} = 1 MHz	Setting prohibited
Other than above			Setting prohibited		

Note 1. See the MCKC register for division ratio settings.

Note 2. When using the high-speed on-chip oscillator clock to operate the USB/function controller, be sure to set f_{HOCO} = 48 MHz.

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
1	1	HS (high-speed main) mode	1 to 24 MHz	2.7 to 3.6 V

Caution 1. Set the HOCODIV register with the high-speed on-chip oscillator clock (f_{IH}) selected as the CPU/peripheral hardware clock (f_{CLK}).

Caution 2. After the frequency is changed with the HOCODIV register, the frequency is switched after the following transition time has elapsed.

- **Operation for up to three clocks at the pre-change frequency**
- **CPU/peripheral hardware clock wait at the post-change frequency for up to three clocks**

Caution 3. To change the frequency of the high-speed on-chip oscillator when X1 oscillation or external oscillation input is set for the system clock, stop the high-speed on-chip oscillator by setting bit 0 (HIOSTOP) of the CSC register to 1 and then change the frequency.

5.3.9 High-speed on-chip oscillator trimming register (HIOTRM)

This register is used to adjust the accuracy of the high-speed on-chip oscillator.
With self-measurement of the high-speed on-chip oscillator frequency via a timer using high-accuracy external clock input (timer array unit), and so on, the accuracy can be adjusted.
The HIOTRM register can be set by an 8-bit memory manipulation instruction.

Caution The frequency will vary if the temperature and V_{DD} pin voltage change after accuracy adjustment. When the temperature and V_{DD} voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

Figure 5 - 11 Format of High-Speed On-Chip Oscillator Trimming Register (HIOTRM)

Address: F00A0H After reset: **Note** R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	High-speed on-chip oscillator
0	0	0	0	0	0	Minimum speed
0	0	0	0	0	1	
0	0	0	0	1	0	
0	0	0	0	1	1	
0	0	0	1	0	0	
			• • •			
1	1	1	1	1	0	
1	1	1	1	1	1	Maximum speed

Note The value after a reset is adjusted at the time of shipment.

Caution 1. The HIOTRM register holds a six-bit value used to adjust the high-speed on-chip oscillator with an increment of 1 corresponding to an increase of frequency by about 0.05%.

Caution 2. For the usage example of the HIOTRM register, see the application note for RL78 MCU series High-speed On-chip Oscillator (HOCO) Clock Frequency Correction (R01AN0464).

5.3.10 PLL control register (DSCCTL)

This register is used to control the operations of the PLL oscillator.

The DSCCTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5 - 12 Format of PLL Control Register (DSCCTL)

Address: F02E5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DSCCTL	0	0	0	0	0	DSFRDIV	DSCM	DSCON

DSFRDIV	PLL reference clock divider control
0	No division
1	Divided by 2

Remark PLL reference clock is the high-speed system clock (f_{MX}).

DSCM	PLL multiplication selection
0	12 times (6 times)
1	16 times (8 times)

Remark The frequency is devided by 2 in the last stage of the PLL oscillator, therefore the multiplication ratio becomes the value in parentheses.

DSCON	PLL oscillation and output control
0	Stop
1	Ocsillation, output

Caution 1. Be sure to clear bits 3 to 7 to 0.

Caution 2. Be sure to set the DSCON bit to 0 before changing DSFRDIV and DSCM.

Caution 3. Do not set the DSCON bit to 0 while the PLL clock is selected as the system clock.

The combination which user can select as the USB clock when the PLL is used is shown below.

Figure 5 - 13 Relationship between the PLL and the USB Clock

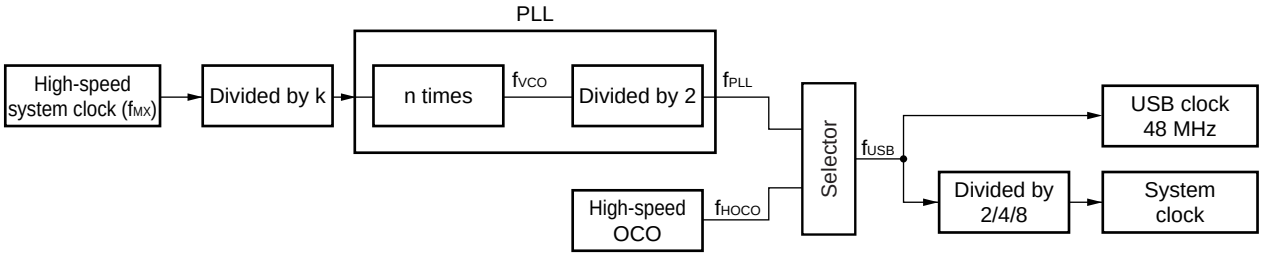


Table 5 - 3 USB Clock Frequency Setting Example

High-speed system clock (fMX)	Divided by k DSFRDIV	n times DSCM		Frequency after n times (fVCO)	USB clock (fUSB)
16 MHz	Divided by 2	0	12 times	96 MHz	48 MHz
12 MHz	Divided by 2	1	16 times	96 MHz	48 MHz
8 MHz	No division	0	12 times	96 MHz	48 MHz
6 MHz	No division	1	16 times	96 MHz	48 MHz

5.3.11 Main clock control register (MCKC)

This register is used to control the operations of the main clock.

The MCKC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5 - 14 Format of Main Clock Control Register (MCKC)

Address: F02E6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCKC	0	0	0	0	0	RDIV1	RDIV0	CKSELR

RDIV1	RDIV0	High-speed on-chip oscillator clock/PLL clock division ratio (divided by 2/4/8) selection
0	0	Divided by 2
0	1	Divided by 4
1	0	Divided by 8
1	1	Setting prohibited

CKSELR	High-speed on-chip oscillator clock/PLL clock selection
0	High-speed on-chip oscillator clock (f_{HOCO})
1	PLL clock (f_{PLL})

Caution 1. When selecting the high-speed on-chip oscillator clock as the USB clock (f_{USB}), set the UCKSELC bit in the USB clock select register (UCKSEL) to 1.

Caution 2. When the clock is switched between the PLL clock (f_{PLL}) and the high-speed on-chip oscillator clock (f_{HOCO}), both clocks must be oscillating.

Caution 3. Be sure to clear bits 3 to 7 to 0.

5.3.12 USB clock selection register (UCKSEL)

The UCKSEL register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 5 - 15 Format of USB clock selection register (UCKSEL)

Address: F04C4H, F04C5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UCKSEL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	UCKSELC

UCKSELC	USB clock selection
0	High-speed on-chip oscillator clock (f_{HOCO}) is not selected as USB clock
1	High-speed on-chip oscillator clock (f_{HOCO}) is selected as USB clock

Caution 1. When selecting the high-speed on-chip oscillator clock (f_{HOCO}) as the USB clock, set UCKSELC = 1 and the CKSELR bit in the MCKC register to 0 at the same time.

Caution 2. The USB clock select register can be rewritten only when the USB is disconnected.

Caution 3. The high-speed on-chip oscillator clock can be selected only when T_A = -20 to +85°C.

Caution 4. If the high-speed on-chip oscillator clock is selected as the USB clock, when the USB is suspended, perform USB suspended processing while the high-speed on-chip oscillator clock is selected (UCKSELC = 1).

Caution 5. If the high-speed on-chip oscillator clock is selected as the USB clock, when the USB is disconnected, perform USB stopped processing (including setting DPRPU = 0) before setting UCKSELC = 0.

5.4 System Clock Oscillator

5.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 20 MHz) connected to the X1 and X2 pins.

An external clock can also be input. In this case, input the clock signal to the EXCLK pin.

To use the X1 oscillator, set bits 7 and 6 (EXCLK, OSCSEL) of the clock operation mode control register (CMC) as follows.

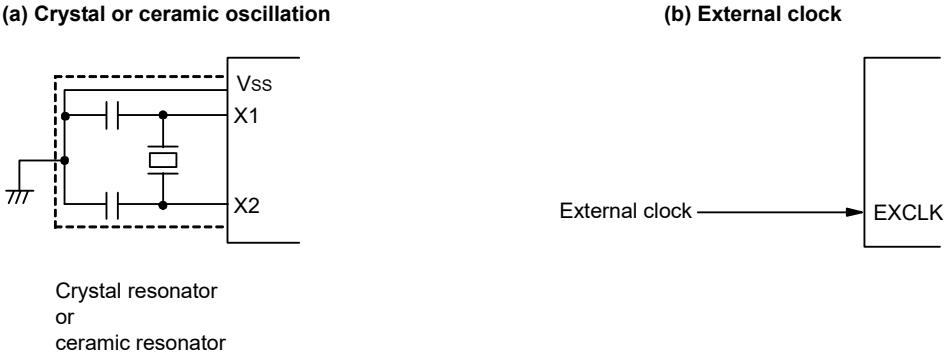
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as input port pins, either, see **Table 2 - 3 Connection of Unused Pins**.

Figure 5 - 16 shows an example of the external circuit of the X1 oscillator.

Figure 5 - 16 Example of External Circuit of X1 Oscillator



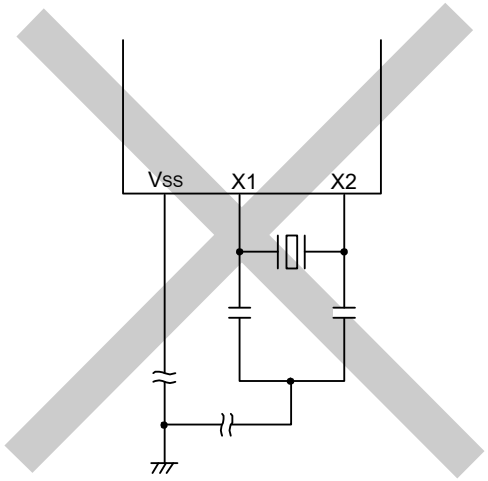
Caution When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the Figure 5 - 16 to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as Vss. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

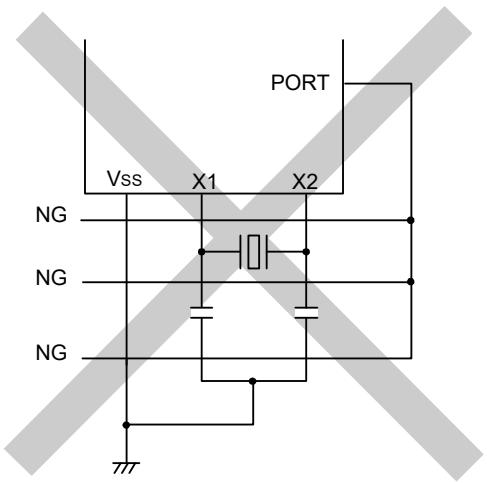
Figure 5 - 17 shows examples of incorrect resonator connection.

Figure 5 - 17 Examples of Incorrect Resonator Connection (1/2)

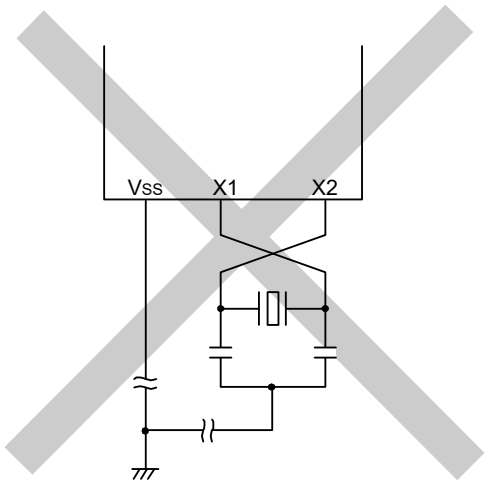
(a) Too long wiring



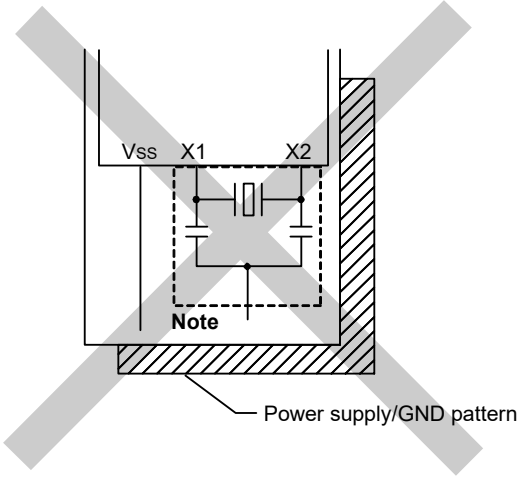
(b) Crossed signal line



(c) The X1 and X2 signal line wires cross.



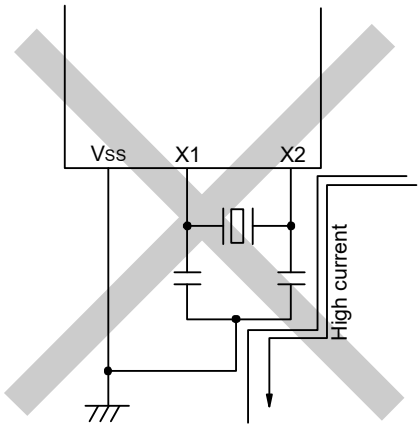
(d) A power supply/GND pattern exists under the X1 and X2 wires.



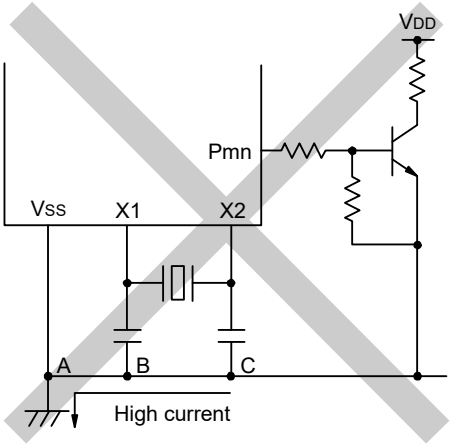
Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

Figure 5 - 17 Examples of Incorrect Resonator Connection (2/2)

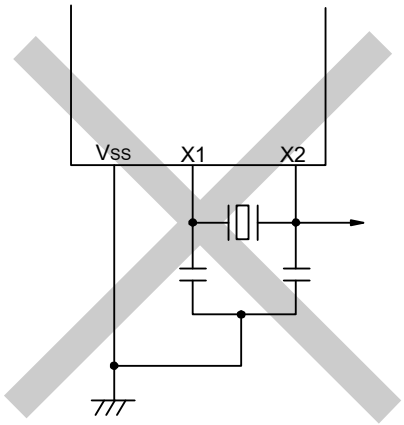
(e) Wiring near high alternating current



(f) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)



(g) Signals are fetched



5.4.2 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the R9A02G015. Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC). The high-speed on-chip oscillator automatically starts oscillating after reset release.

5.4.3 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the R9A02G015.
The low-speed on-chip oscillator clock is used only as the watchdog timer, and 12-bit interval timer clock. The low-speed on-chip oscillator clock cannot be used as the CPU clock.
This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC), or both are set to 1.
Unless the watchdog timer is stopped and WUTMMCK0 is a value other than zero, oscillation of the low-speed on-chip oscillator continues. Note that only when the watchdog timer is operating and the WUTMMCK0 bit is 0, oscillation of the low-speed on-chip oscillator will stop while the WDSTBYON bit is 0 and operation is in the HALT, STOP, or SNOOZE mode. While the watchdog timer operates, the low-speed on-chip oscillator clock does not stop even if the program freezes.

5.4.4 PLL (Phase Locked Loop)

The PLL circuit is incorporated in the R9A02G015.
The PLL can be used to multiply the high-speed system clock.
Operation of the PLL circuit can be controlled by using bit 0 (DSCON) of the PLL control register (DSCCTL).

Caution When switching from PLL mode to the internal high-speed oscillation clock and the high-speed system clock, stop the function (USB function controller) that provides the PLL output clock (f_{PLL}).

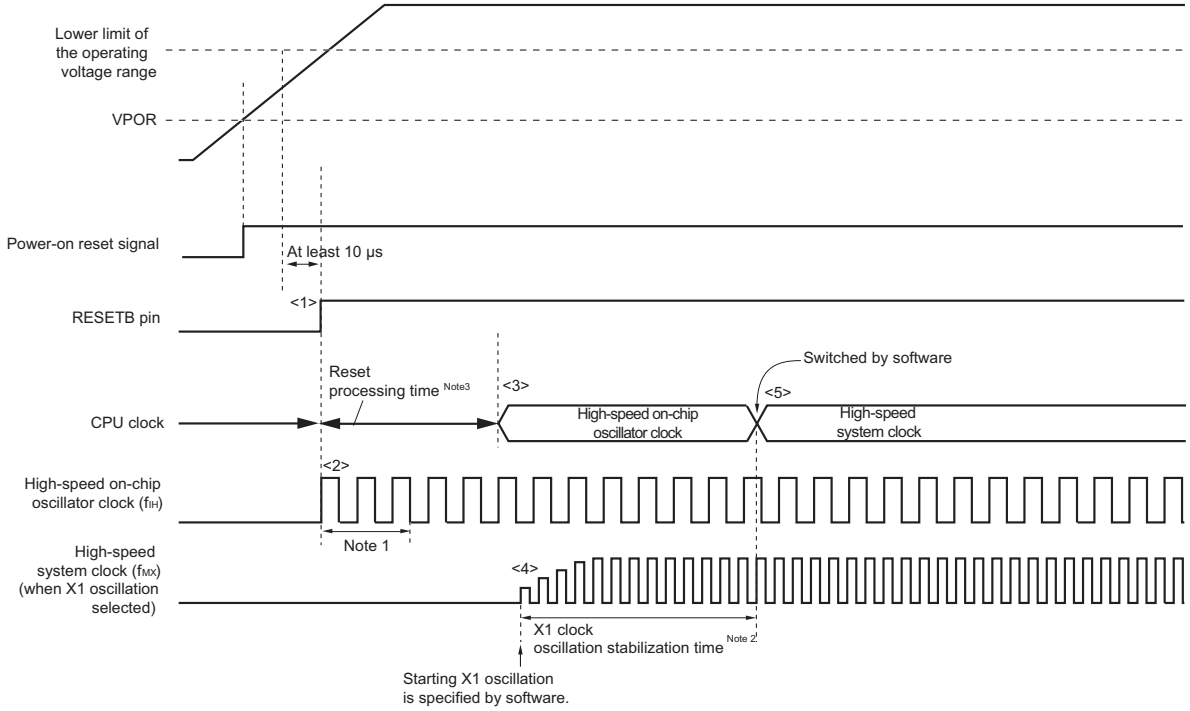
5.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5 - 1**).

- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - High-speed on-chip oscillator clock f_{HOCO}
  - PLL clock f_{PLL}
- Low-speed on-chip oscillator clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the R9A02G015.
When the power supply voltage is turned on, the clock generator operation is shown in **Figure 5 - 18**.

Figure 5 - 18 Clock Generator Operation When Power Supply Voltage Is Turned On



<1> When the power is turned on, an internal reset signal is generated by the power-on-reset (POR) circuit. Note that the reset state is maintained after a reset by the voltage detector or an external reset until the voltage reaches the range of operating voltage described in **2.4 AC Characteristics** of the R9A02G015 Data Sheet (R19DS0101E) (the above figure is an example when the external reset is in use).

<2> When the reset is released, the high-speed on-chip oscillator automatically starts oscillation.

<3> The CPU starts operation on the high-speed on-chip oscillator clock after waiting for the voltage to stabilize and a reset processing have been performed after reset release.

<4> Set the start of oscillation of the X1 clock via software (see **5.6.2 Example of setting X1 oscillation clock**).

<5> When switching the CPU clock to the X1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **5.6.2 Example of setting X1 oscillation clock**).

Note 1. The reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. When releasing a reset, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC).

Note 3. For the reset processing time, see **CHAPTER 17 POWER-ON-RESET CIRCUIT**.

Caution It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.

5.6 Controlling Clock

5.6.1 Example of setting high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 48, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using FRQSEL0 to FRQSEL4 of the option byte (000C2H). In addition, oscillation can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

Option byte (000C2H)	7	6	5	4	3	2	1	0
	CMODE1 1	CMODE0 1	 1	FRQSEL4 1	FRQSEL3 0	FRQSEL2 0	FRQSEL1 0	FRQSEL0 0/1

CMODE1	CMODE0	Setting of flash operation mode
1	1	HS (high speed main) mode V_{DD} = 2.7 V to 5.5 V @ 1 MHz to 24 MHz
Other than above		Setting prohibited

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator	
					f_{HOCO}	f_{IH}
1	0	0	0	0	48 MHz **Note 1**	24/12/6 MHz **Note 2**
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than above					Setting prohibited	

Note 1. When you use PLL, set it in FRQSEL4 = 0, and, please do not choose 48 MHz.

Note 2. See the MCKC register for division ratio settings.

[High-speed on-chip oscillator frequency select register (HOCODIV) setting]

Address: F00A8H

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator clock frequency		
			FRQSEL4 = 0		FRQSEL4 = 1
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0
0	0	0	f_{IH} = 24 MHz	Setting prohibited	f_{IH} = 24/12/6 MHz **Note** f_{HOCO} = 48 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz	f_{IH} = 12/6/3 MHz **Note** f_{HOCO} = 24 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz	f_{IH} = 6/3 MHz **Note** f_{HOCO} = 12 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz	f_{IH} = 3 MHz f_{HOCO} = 6 MHz
1	0	0	Setting prohibited	f_{IH} = 2 MHz	Setting prohibited
1	0	1	Setting prohibited	f_{IH} = 1 MHz	Setting prohibited
Other than above			Setting prohibited		

Note See the MCKC register for division ratio settings.

5.6.2 Example of setting X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the X1 oscillation clock, set the oscillator and start oscillation by using the oscillation stabilization time select register (OSTS) and clock operation mode control register (CMC) and clock operation status control register (CSC) and wait for oscillation to stabilize by using the oscillation stabilization time counter status register (OSTC). After the oscillation stabilizes, set the X1 oscillation clock to f_{CLK} by using the system clock control register (CKC).

[Register settings] Set the register in the order of <1> to <5> below.

<1> Set (1) the OSCSEL bit of the CMC register, except for the cases where fx > 10 MHz, in such cases set (1) the AMPH bit, to operate the X1 oscillator.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 1	0	0	0	0	0	AMPH 0/1

AMPH bit: Set this bit to 0 if the X1 oscillation clock is 10 MHz or less.

<2> Using the OSTS register, select the oscillation stabilization time of the X1 oscillator at releasing of the STOP mode.

Example: Setting values when a wait of at least 102 μs is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2 0	OSTS1 1	OSTS0 0

<3> Clear (0) the MSTOP bit of the CSC register to start oscillating the X1 oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP 0	1	0	0	0	0	0	HIOSTOP 0

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.

Example: Wait until the bits reach the following values when a wait of at least 102.4 μs is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTC	MOST8 1	MOST9 1	MOST10 1	MOST11 0	MOST13 0	MOST15 0	MOST17 0	MOST18 0

<5> Use the MCM0 bit of the CKC register to specify the X1 oscillation clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	0	0	MCS 0	MCM0 1	0	0	0	0

Caution Set the HOCODIV register within the operable voltage range of the flash operation mode set in the option byte (000C2H) before and after the frequency change.

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
1	1	HS (high-speed main) mode	1 MHz to 24 MHz	2.7 V to 5.5 V

5.6.3 Example of setting PLL circuit

After setting the high-speed system clock (see **5.6.2 Example of setting X1 oscillation clock**), use the PLL control register (DSCCTL) to control the PLL circuit.

[Register settings] Set the register in the order of <1> to <5> below.

<1> Set the HIOSTOP bit in the CSC register to make the high-speed on-chip oscillator run.

	7	6	5	4	3	2	1	0
CSC	0/1	0/1	0	0	0	0	0	HIOSTOP 0 Note 1

<2> Set the DSFRDIV bit and DSCM bit in the DSCCTL register to set the PLL multiplication and division.

	7	6	5	4	3	2	1	0
DSCCTL	0	0	0	0	0	DSFRDIV 0/1	DSCM 0/1	DSCON 0

<3> Set the RDIV1, RDIV0 bits of the MCKC register to set the division of the system clock.

	7	6	5	4	3	2	1	0
MCKC	0	0	0	0	0	RDIV1 0/1	RDIV0 0/1	CKSELR 0

<4> After wait more than 1 μs, set (1) the DSCON bit of the DSCCTL register to operate the PLL circuit Note 2.

	7	6	5	4	3	2	1	0
DSCCTL	0	0	0	0	0	DSFRDIV 0/1	DSCM 0/1	DSCON 1

<5> Set (1) the CKSELR bit of the MCKC register to select PLL output for the system clock.

	7	6	5	4	3	2	1	0
MCKC	0	0	0	0	0	RDIV1 0/1	RDIV0 0/1	CKSELR 1

<6> Use software to set up a wait of 135 µs. **Note 3**

<7> Set the HIOSTOP bit in the CSC register to stop the high-speed on-chip oscillator. **Note 2**

	7	6	5	4	3	2	1	0
CSC	0/1	0/1	0	0	0	0	0	HIOSTOP 0 **Note 1**

<8> When the PLL clock frequency divided by 2, 4, or 8 is selected as the main system clock (f_{MAIN}), set the MCM0 bit in the CKC register to select the source for deriving the main system clock as a signal with a frequency (f_{IH}) of up to 24 MHz.

	7	6	5	4	3	2	1	0
CKC	0/1	0/1	MCS 0	MCM0 0	0	0	0	0

Note 1. No setting is required to change to the PLL while the CKSELR bit is 1.
When setting the CKSELR bit to 1, ensure that the high-speed on-chip oscillator is running.

Note 2. After oscillation by the X1 oscillator clock has become stable, allow at least 1 µs to elapse before starting the PLL. When restarting the PLL after it has been stopped, wait for at least 4 µs before using it in operations.

Note 3. Wait for 40 µs for oscillation by the oscillator clock to become stabled if the HIOSTOP bit is not set to 0.

5.6.4 CPU clock status transition diagram

Figure 5 - 19 shows the CPU clock status transition diagram of this product.

Figure 5 - 19 CPU Clock Status Transition Diagram

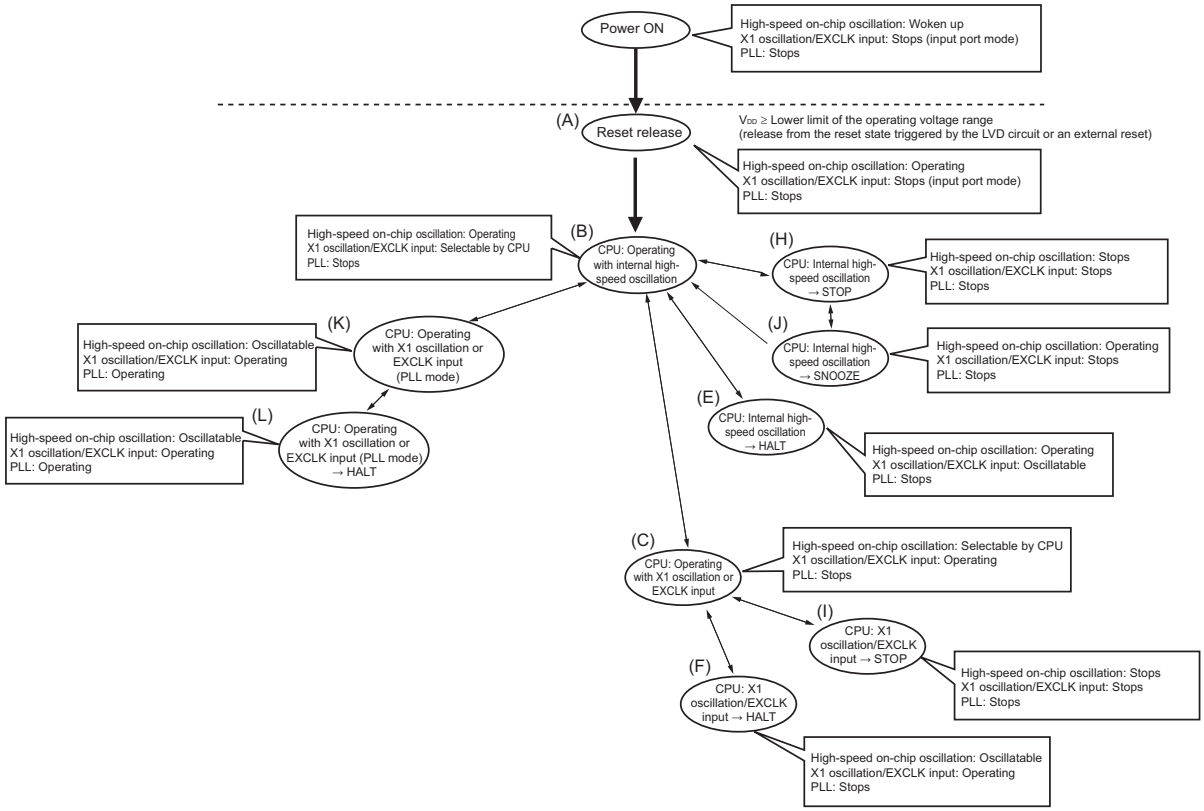


Table 5 - 4 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5 - 4 CPU Clock Transition and SFR Register Setting Examples (1/4)

(1) CPU operating with high-speed on-chip oscillator clock (B) after reset release (A)

Status Transition	SFR Register Setting
(A) → (B)	SFR registers do not have to be set (default status after reset release).

(2) CPU operating with high-speed system clock (C) after reset release (A)

(The CPU operates with the high-speed on-chip oscillator clock immediately after a reset release (B).)

(Setting sequence of SFR registers) ⟶

Setting Flag of SFR Register / Status Transition	CMC Register Note 1			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A) → (B) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	0	1	0	**Note 2**	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	**Note 2**	0	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	×	**Note 2**	0	Must not be checked	1

Note 1. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release.

Note 2. Set the oscillation stabilization time as follows.

- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see ELECTRICAL SPECIFICATIONS in the R9A02G015 Data Sheet (R19DS0101E)).

Table 5 - 4 CPU Clock Transition and SFR Register Setting Examples (2/4)

(3) CPU clock changing from high-speed on-chip oscillator clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note 1			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(B) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	0	1	0	**Note 2**	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	**Note 2**	0	Must be checked	1
(B) → (C) (external main clock)	1	1	×	**Note 2**	0	Need not be checked	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

Note 1. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release. This setting is not necessary if it has already been set.

Note 2. Set the oscillation stabilization time as follows.

- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see ELECTRICAL SPECIFICATIONS in the R9A02G015 Data Sheet (R19DS0101E)).

Table 5 - 4 CPU Clock Transition and SFR Register Setting Examples (3/4)

(4) CPU clock changing from high-speed system clock (C) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		MCM0
(C) → (B)	0	18 μs to 135 μs	0

Unnecessary if the CPU is operating with the high-speed on-chip oscillator clock

Remark The oscillation accuracy stabilization time changes according to the temperature conditions and the period over which the HIOSTOP bit is 0.

(5) CPU clock changing from high-speed on-chip oscillator clock (B) to high-speed system clock (PLL mode) (K)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note 1			OSTS Register	CSC Register	OSTC Register	DSCCTL Register	
	EXCLK	OSCSEL	AMPH		MSTOP		DSFRDIV	DSCM
(B) → (K) (divided by 2)	0/1	1	0/1	**Note 2**	0	Must be checked	0/1	0/1
(B) → (K) (divided by 4)	0/1	1	0/1	**Note 2**	0	Must be checked	0/1	0/1
(B) → (K) (divided by 8)	0/1	1	0/1	**Note 2**	0	Must be checked	0/1	0/1

→

MCKC Register		Waiting for Oscillation Stabilization	DSCCTL Register	Waiting for Oscillation Stabilization	MCKC Register
RDIV1	RDIV0		DSCON		CKSELR
0	0	1 μs	1	40 μs	1
0	1		1		1
1	0		1		1

Note 1. Writing to the clock operating mode control register (CMC) can only proceed once and must be by an 8-bit memory manipulation instruction after release from the reset state.

Note 2. Set the oscillation stabilization time in the oscillation stabilization time select register (OSTS) as follows.

- Desired oscillation stabilization time setting of the oscillation stabilization time counter status register (OSTC) ≤ Oscillation stabilization time set in the OSTS register

Caution Completion of clock switching after the CKSELR bit has been set to 1 requires up to 2 clock cycles when the FRQSEL4 bit is 1, and up to 10 clock cycles when the FRQSEL4 bit is 0. Until the clock switching is completed, do not stop the high-speed on-chip oscillator.

Remark (A) to (L) in Table 5 - 4 correspond to (A) to (L) in Figure 5 - 19.

Table 5 - 4 CPU Clock Transition and SFR Register Setting Examples (4/4)

(6) CPU clock changing from high-speed system clock (PLL mode) (K) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Waiting for Oscillation Stabilization	MCKC Register	Waiting for Oscillation Stabilization	DSCCTL Register
	HIOSTOP		CKSELR		DSCON
(K) → (B) FRQSEL4 = 0	0	18 to 65 μs	0	256 clocks	0
(K) → (B) FRQSEL4 = 1		18 ro 135 μs		16 clocks	

(7) • HALT mode (E) set while CPU is operating with high-speed on-chip oscillator clock (B)
• HALT mode (F) set while CPU is operating with high-speed system clock (C)
• HALT mode (L) set while CPU is operating with high-speed system clock (PLL mode) (K)

Status Transition	Setting
(B) → (E) (C) → (F) (K) → (L)	Executing HALT instruction

(8) STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

Status Transition		Setting		
(B) → (H)		Stopping peripheral functions that cannot operate in STOP mode	—	Executing STOP instruction
(C) → (I)	In X1 oscillation		Sets the OSTS register	
	External main system clock		—	

Remark (A) to (L) in Table 5 - 4 correspond to (A) to (L) in Figure 5 - 19.

(9) CPU changing from STOP mode (H) to SNOOZE mode (J)

For details about the setting for switching from the STOP mode to the SNOOZE mode, see **10.8 SNOOZE Mode Function**, and **11.5.7 SNOOZE mode function**.

(10) STOP mode (I) set while CPU is operating with high-speed system clock (PLL mode) (K)

Switch to high-speed on-chip oscillator clock operation from PLL mode, stop the PLL (DSCON = 0), and then execute the STOP instruction.

Remark (A) to (L) in Table 5 - 4 correspond to (A) to (L) in Figure 5 - 19.

5.6.5 Condition before changing CPU clock and processing after changing CPU clock

Condition before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 5 - 5 Changing CPU Clock (1/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	After checking that the CPU clock is switched to the clock after change, operating current can be reduced by stopping high-speed on-chip oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	PLL clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time Or enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0 Oscillation of PLL • DSCON = 1	
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation accuracy stabilization time	After checking that the CPU clock is switched to the clock after change, X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible (To change the clock, set it again after executing reset once.)	—
	PLL clock	Oscillation of PLL • DSCON = 1 Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	—

Table 5 - 5 Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation accuracy stabilization time	External main system clock input can be disabled (MSTOP = 1).
	X1 clock	Transition not possible	—
	PLL clock	Oscillation of PLL • DSCON = 1 Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	—
PLL clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator • HIOSTOP = 0	Operating current can be reduced by stopping PLL (DSCON = 0).
	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	

5.6.6 Time required for switchover of CPU clock and system clock

By setting bit 4 (MCM0) of the system clock control register (CKC), the CPU clock can be switched, and main system clock can be switched (between the high-speed on-chip oscillator clock and the high-speed system clock).

The actual switchover operation is not performed immediately after rewriting to the CKC register; operation continues on the pre-switchover clock for several clocks (see Table 5 - 6 and Table 5 - 7).

Whether the main system clock is operating on the high-speed system clock or high-speed on-chip oscillator clock can be ascertained using bit 5 (MCS) of the CKC register.

When the CPU clock is switched, the peripheral hardware clock is also switched.

Table 5 - 6 Maximum Time Required for System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	◀——▶	f_{MX}	See Table 5 - 7

Table 5 - 7 Maximum Number of Clocks Required for $f_{IH} \leftrightarrow f_{MX}$

Set Value Before Switchover		Set Value After Switchover	
MCM0		MCM0	
		0 ($f_{MAIN} = f_{IH}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$		2 clock
	$f_{MX} < f_{IH}$		2 f_{IH}/f_{MX} clock
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{IH}$	2 f_{MX}/f_{IH} clock	
	$f_{MX} < f_{IH}$	2 clock	

Remark 1. The number of clocks listed in Table 5 - 7 is the number of CPU clocks before switchover.

Remark 2. Calculate the number of clocks in Table 5 - 7 by removing the decimal portion.

Example When switching the main system clock from the high-speed on-chip oscillator clock (when 12 MHz selected) to the high-speed system clock (@ oscillation with f_{IH} = 12 MHz, f_{MX} = 10 MHz)

1 + f_{IH}/f_{MX} = 1 + 12/10 = 1 + 1.2 = 2.2 → 2 clocks

5.6.7 Conditions before clock oscillation is stopped

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

Before stopping the clock oscillation, check the conditions before the clock oscillation is stopped.

Table 5 - 8 Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
High-speed on-chip oscillator clock	MCS = 1 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
X1 clock External main system clock	MCS = 0 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1

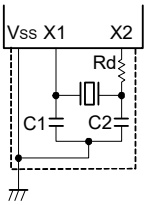
5.7 Resonator and Oscillator Constants

For the resonators for which operation has been verified and their oscillator constants, see the target product page on the Renesas Web site.

Caution 1. The constants for these oscillator circuits are reference values based on specific environments set up for evaluation by the manufacturers. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board. Furthermore, if you are switching from a different product to this microcontroller, and whenever you change the board, again request evaluation by the manufacturer of the oscillator circuit mounted on the new board.

Caution 2. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the R9A02G015 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Figure 5 - 20 External Oscillation Circuit Example



X1 oscillation:

As of March, 2013 (1/2)

Manufacturer	Resonator	Part Number **Note 3**	SMD/ Lead	Frequency (MHz)	Flash operation mode **Note 1**	Recommended Circuit Constants **Note 2** (reference)			Oscillation Voltage Range (V)	
						C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Murata Manufacturing Co., Ltd.	Ceramic resonator	CSTCC2M00G56-R0	SMD	2	LV	(47)	(47)	0	1.6	5.5
		CSTCR4M00G55-R0	SMD	4		(39)	(39)	0		
		CSTLS4M00G53-B0	Lead			(15)	(15)	0		
		CSTCC2M00G56-R0	SMD	2	LS	(47)	(47)	0	1.8	5.5
		CSTCR4M00G55-R0	SMD	4		(39)	(39)	0		
		CSTLS4M00G53-B0	Lead			(15)	(15)	0		
		CSTCR4M19G55-R0	SMD	4.194		(39)	(39)	0		
		CSTLS4M19G53-B0	Lead			(15)	(15)	0		
		CSTCR4M91G53-R0	SMD	4.915		(15)	(15)	0		
		CSTLS4M91G53-B0	Lead			(15)	(15)	0		
		CSTCR5M00G53-R0	SMD	5		(15)	(15)	0		
		CSTLS5M00G53-B0	Lead			(15)	(15)	0		
		CSTCR6M00G53-R0	SMD	6		(15)	(15)	0		
		CSTLS6M00G53-B0	Lead			(15)	(15)	0		
		CSTCE8M00G52-R0	SMD	8		(10)	(10)	0		
		CSTLS8M00G53-B0	Lead			(15)	(15)	0		
		CSTCE8M38G52-R0	SMD	8.388	HS	(10)	(10)	0	2.4	5.5
		CSTLS8M38G53-B0	Lead			(15)	(15)	0		
		CSTCE10M0G52-R0	SMD	10		(10)	(10)	0		
		CSTLS10M0G53-B0	Lead			(15)	(15)	0		
		CSTCE12M0G52-R0	SMD	12		(10)	(10)	0		
		CSTCE16M0V53-R0	SMD	16		(15)	(15)	0		
		CSTLS16M0X51-B0	Lead			(5)	(5)	0		
		CSTCE20M0V51-R0	SMD	20		(5)	(5)	0	2.7	5.5
		CSTLS20M0X51-B0	Lead			(5)	(5)	0		

Note 1. Set the flash operation mode by using CMODE1 and CMODE0 bits of the option byte (000C2H).

Note 2. Values in parentheses in the C1 and C2 columns indicate an internal capacitance.

Note 3. Products supporting 105°C operation have different part numbers. For details, contact Murata Co., Ltd (http://www.murata.co.jp).

Remark Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.
HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz

As of March, 2013 (2/2)

Manufacturer	Resonator	Part Number	SMD/ Lead	Frequency (MHz)	Flash operation mode **Note 1**	Recommended Circuit Constants **Note 2** (reference)			Oscillation Voltage Range (V)	
						C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Nihon Dempa Kogyo Co., Ltd.	Crystal resonator	NX8045GB **Note 3**	SMD	8	**Note 3**					
		NX5032GA **Note 3**	SMD	16						
		NX3225HA **Note 3**	SMD	20						
Kyocera Crystal Device Co., Ltd.	Crystal resonator	CX8045GB04000D0PPTZ1 **Note 4**	SMD	4	LV	12	12	0	1.6	5.5
					LS				1.6	5.5
		CX8045GB04915D0PPTZ1 **Note 4**	SMD	4.915	LS	12	12	0	1.8	5.5
		CX8045GB08000D0PPTZ1 **Note 4**	SMD	8	LS	12	12	0	1.8	5.5
		CX8045GB10000D0PPTZ1 **Note 4**	SMD	10	HS	12	12	0	2.4	5.5
		CX3225GB12000B0PPTZ1 **Note 4**	SMD	12	HS	5	5	0	2.4	5.5
		CX3225GB16000B0PPTZ1 **Note 4**	SMD	16	HS	5	5	0	2.4	5.5
		CX3225SB20000B0PPTZ1 **Note 4**	SMD	20	HS	5	5	0	2.7	5.5
RIVER ELETEC CORPORATION	Crystal resonator	FCX-03-8.000MHZ-J21140 **Note 5**	SMD	8	HS	3	3	0	2.4	5.5
		FCX-04C-10.000MHZ-J21139 **Note 5**	SMD	10	HS	4	4	0	2.4	5.5
		FCX-05-12.000MHZ-J21138 **Note 5**	SMD	12	HS	6	6	0	2.4	5.5
		FCX-06-16.000MHZ-J21137 **Note 5**	SMD	16	HS	4	4	0	2.4	5.5

Note 1. Set the flash operation mode by using CMODE1 and CMODE0 bits of the option byte (000C2H).

Note 2. This resonator supports operation at up to 85°C. Contact crystal oscillator manufacturers with regard to products supporting operation at up to 105°C.

Note 3. When using this resonator, for details about the matching, contact Nihon Dempa Kogyo Co., Ltd (http://www.ndk.com/en).

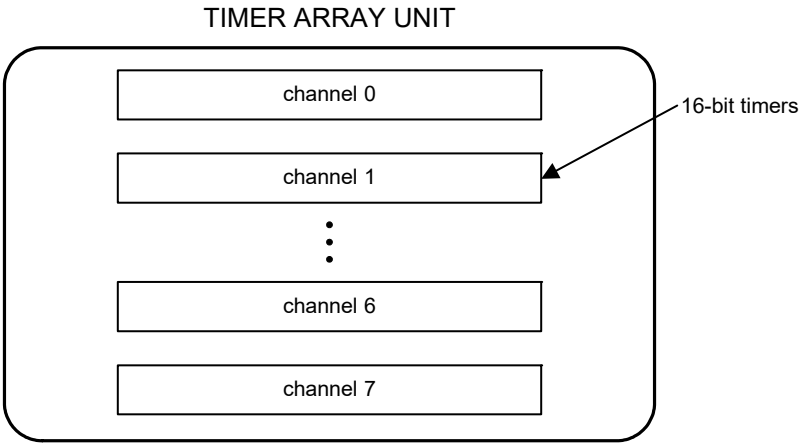
Note 4. When using this resonator, for details about the matching, contact Kyocera Crystal Device Co., Ltd. (http://www.kyocera-crystal.jp, http://www.kyocera.co.jp).

Note 5. When using this resonator, for details about the matching, contact RIVER ELETEC CORPORATION (http://www.river-ele.co.jp).

Remark Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.
HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz

CHAPTER 6 TIMER ARRAY UNIT

The timer array unit has eight 16-bit timers.
Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.



For details about each function, see the table below.

Independent channel operation function	Simultaneous channel operation function
• Interval timer (→ refer to **6.8.1**) • Square wave output (→ refer to **6.8.1**) • External event counter (→ refer to **6.8.2**) • Input pulse interval measurement (→ refer to **6.8.3**) • Measurement of high-/low-level width of input signal (→ refer to **6.8.4**) • Delay counter (→ refer to **6.8.5**)	• One-shot pulse output (→ refer to **6.9.1**) • PWM output (→ refer to **6.9.2**) • Multiple PWM output (→ refer to **6.9.3**)

It is possible to use the 16-bit timer of channels 1 and 3 as two 8-bit timers (higher and lower). The functions that can use channels 1 and 3 as 8-bit timers are as follows:

- Interval timer (upper or lower 8-bit timer)/square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

6.1 Functions of Timer Array Unit

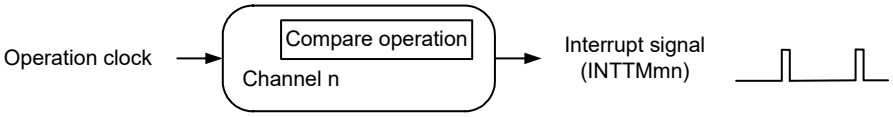
Timer array unit has the following functions.

6.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

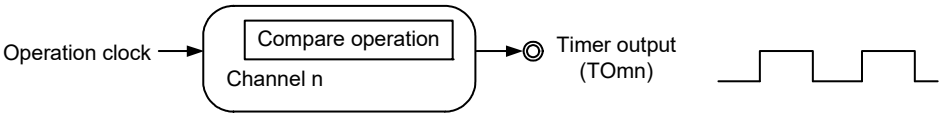
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



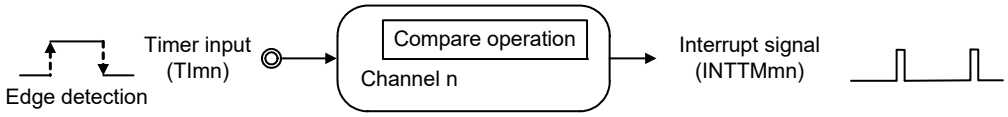
(2) Square wave output

A toggle operation is performed each time INTTMmn interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOmn).



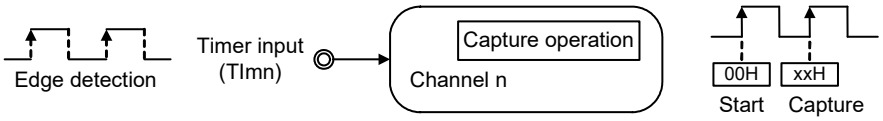
(3) External event counter

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 (TImn) has reached a specific value.



(4) Input pulse interval measurement

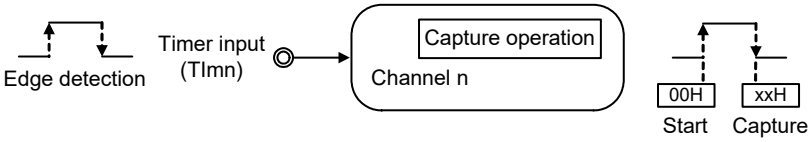
Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



(**Remark** is listed on the next page.)

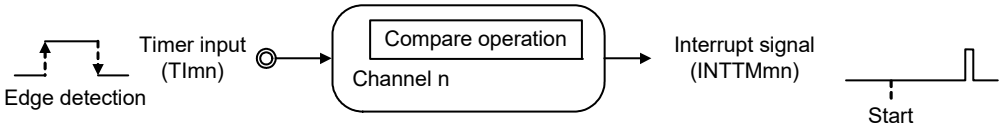
(5) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TImn), 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.



(6) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TImn), and an interrupt is generated after any delay period.



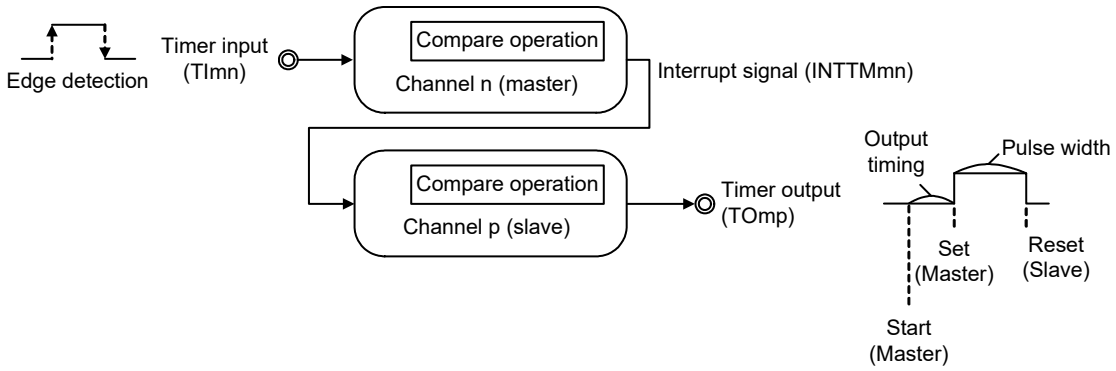
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel), channels can be used for the following purposes.

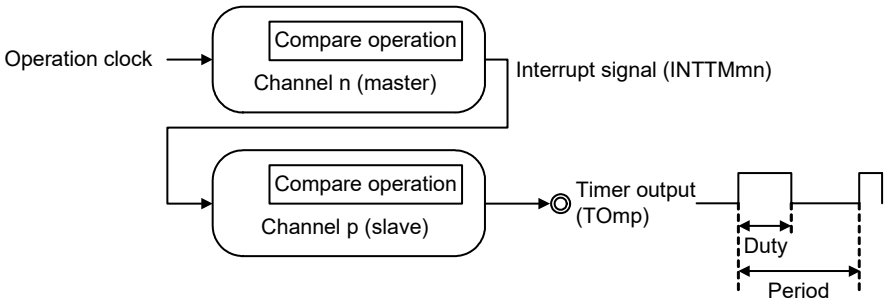
(1) One-shot pulse output

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



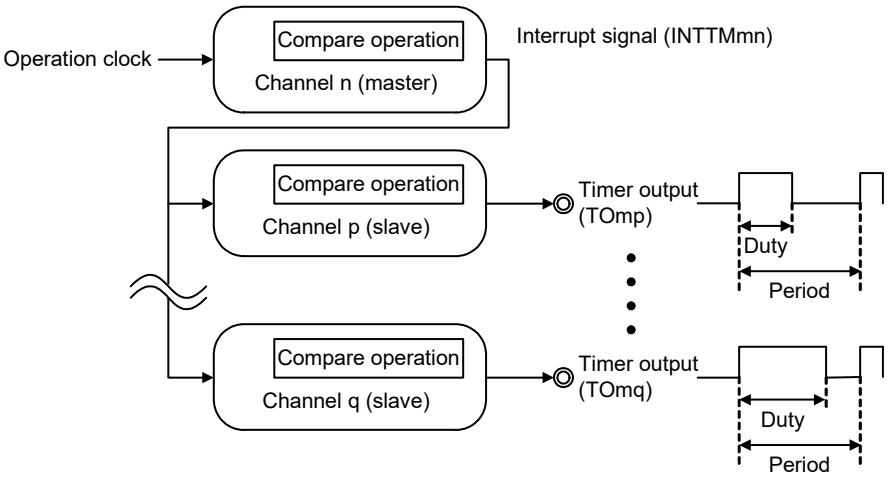
(2) PWM (Pulse Width Modulation) output

Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



(3) 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.



Caution For details about the rules of simultaneous channel operation function, see 6.4.1 Basic rules of simultaneous channel operation function.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7),
p, q: Slave channel number (n < p < q ≤ 7)

6.1.3 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.

Caution There are several rules for using 8-bit timer operation function.
For details, see 6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 6 - 1 Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer count register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI07 Note 1
Timer output	TO00 to TO07 Note 2
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer input select register 0 (TIS0) • Timer output enable register m (TOEm) • Timer output register m (TOm) • Timer output level register m (TOLm) • Timer output mode register m (TOMm)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Noise filter enable register 1 (NFEN1) • Port mode control register (PMCxx) Note 3 • Port mode register (PMxx) Note 3 • Port register (Pxx) Note 3

Note 1. TI07 is incorporated with R9A02G0151, but is not incorporated with R9A02G0150.

Note 2. TO07 is incorporated with R9A02G0151, but is not incorporated with R9A02G0150.

Note 3. The Port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. For details, see **4.5.3 Register setting examples for used port and alternate functions**.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

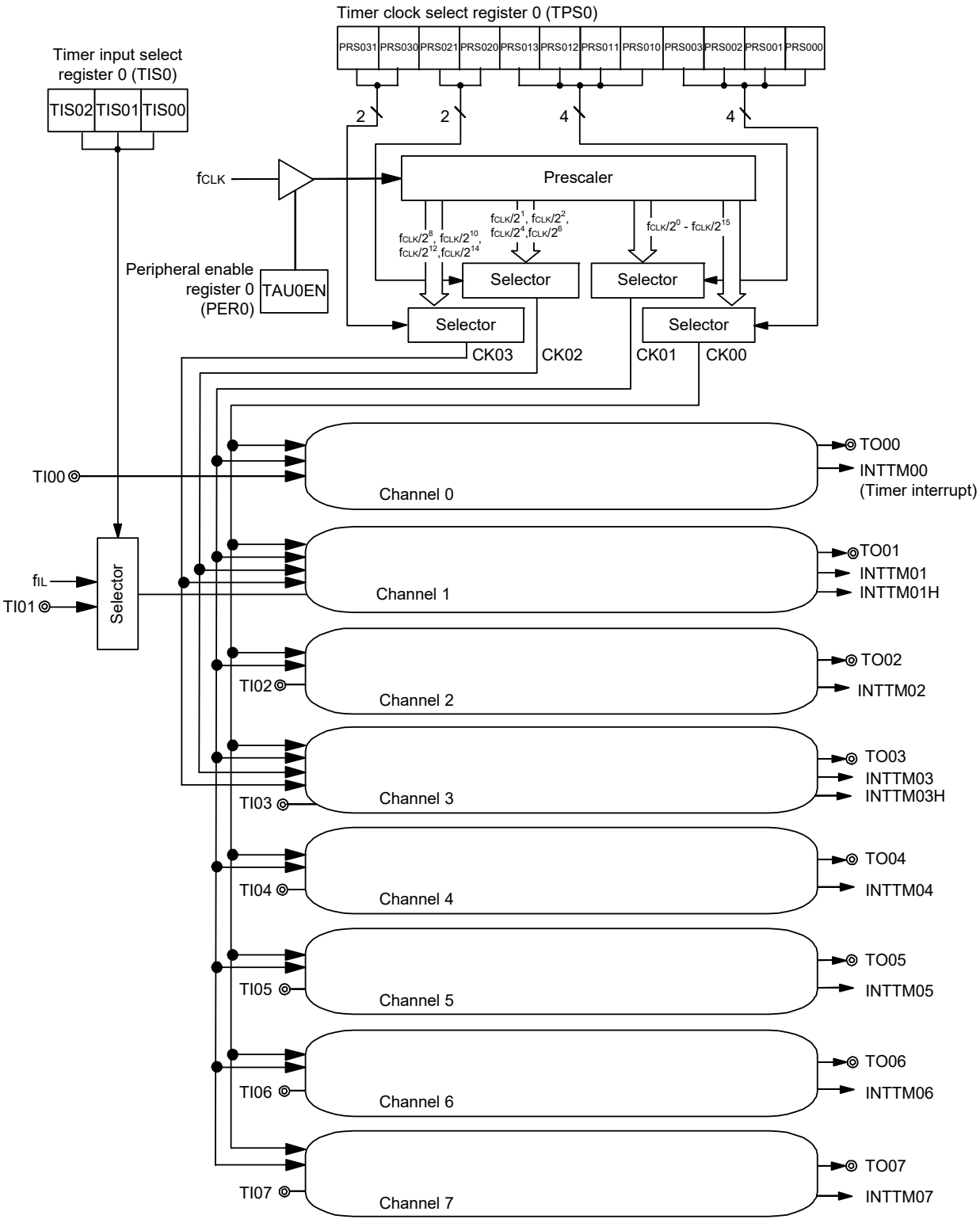
The port pins alternatively used as timer I/O pins in each timer array unit channel depend on the product.

Table 6 - 2 Timer I/O Pins provided in Each Product

Timer array unit channels	with USB	without USB
Channel 0	TI00/TO00	
Channel 1	TI01/TO01	
Channel 2	TI02/TO02	
Channel 3	TI03/TO03	
Channel 4	TI04/TO04	
Channel 5	TI05/TO05	
Channel 6	TI06/TO06	
Channel 7	—	TI07/TO07

Figure 6 - 1 shows the block diagram of the timer array unit.

Figure 6 - 1 Entire Configuration of Timer Array Unit 0



Remark fIL: Low-speed on-chip oscillator clock frequency

Figure 6 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit

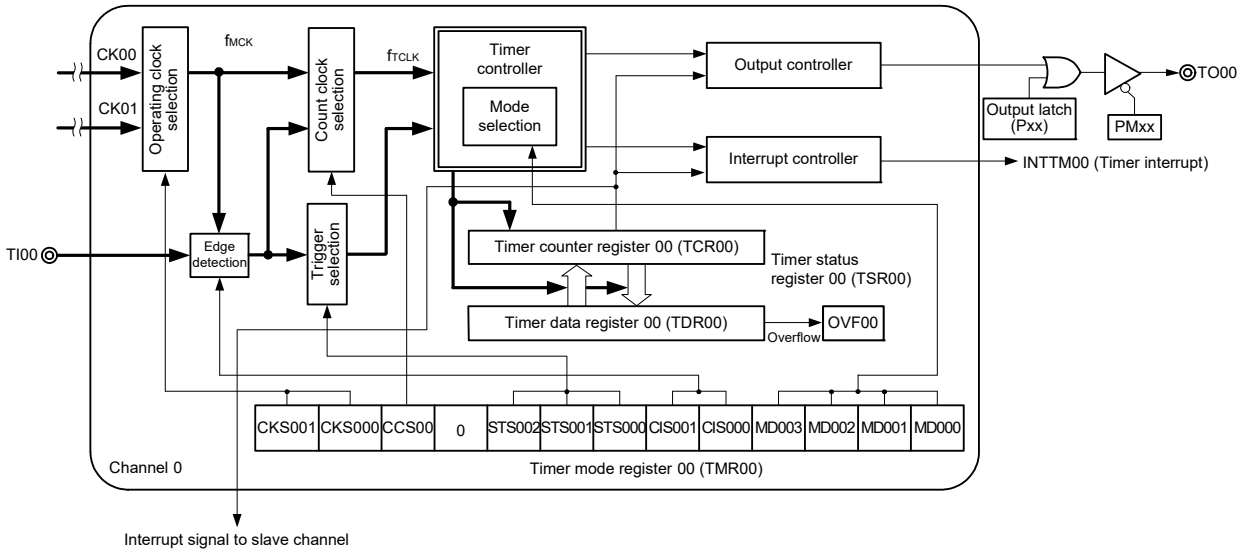


Figure 6 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit

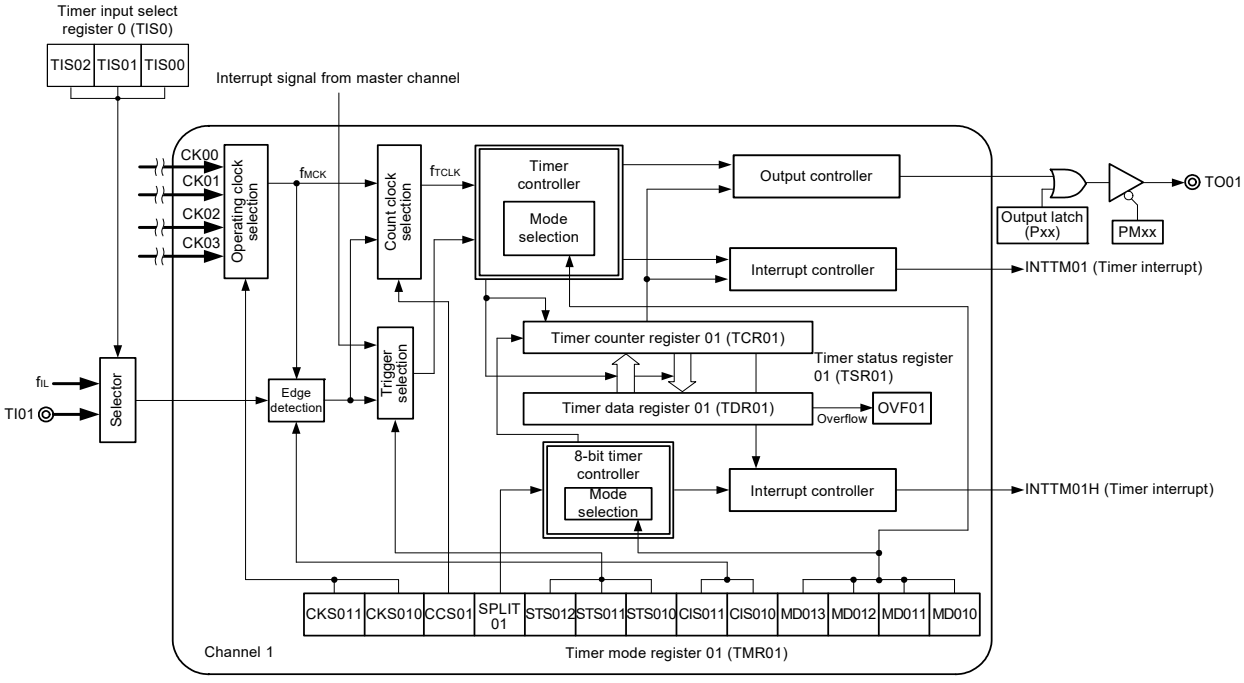
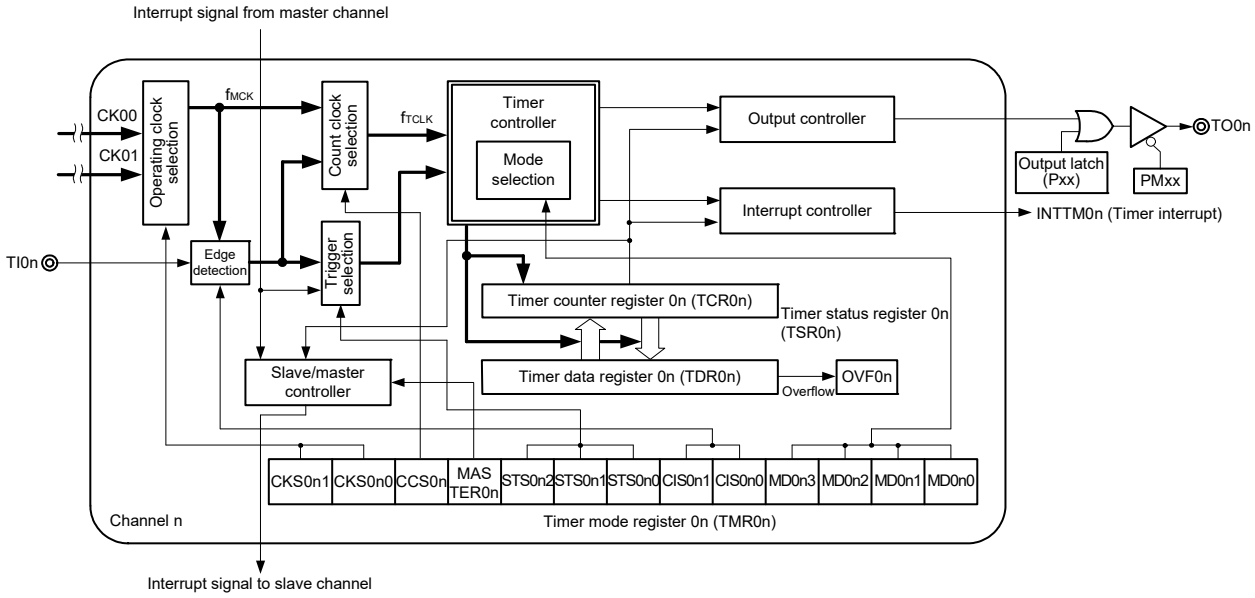


Figure 6 - 4 Internal Block Diagram of Channel n of Timer Array Unit



Remark n = 2, 4, 6

Figure 6 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit

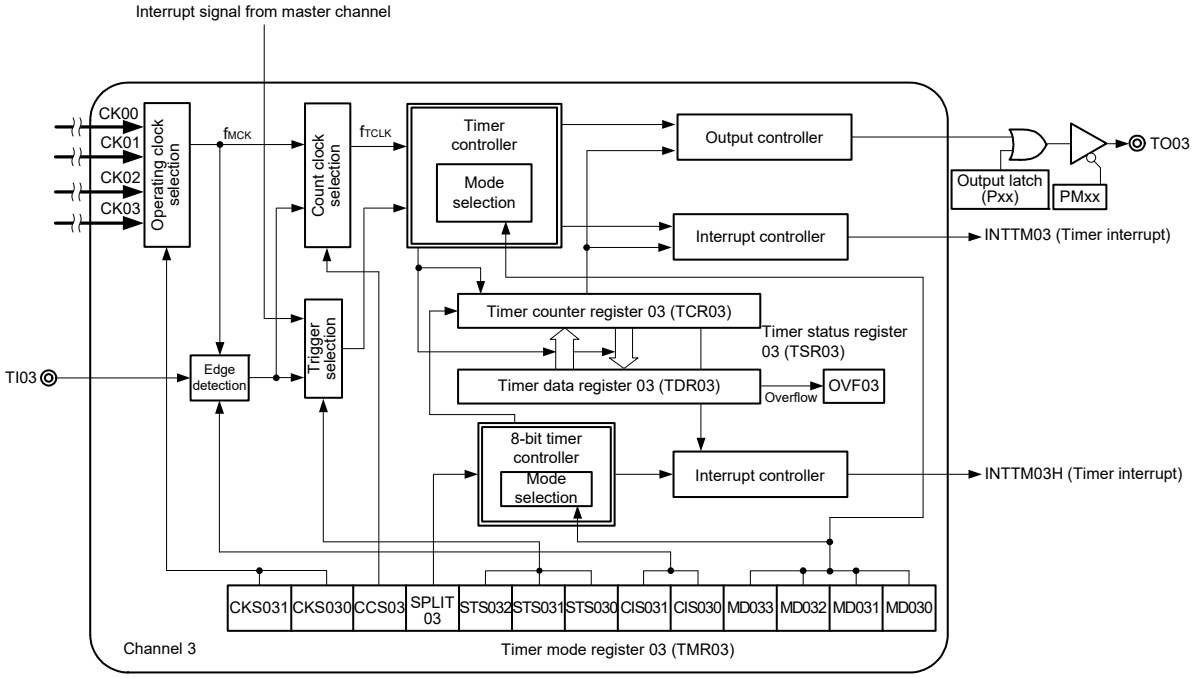


Figure 6 - 6 Internal Block Diagram of Channel 5 of Timer Array Unit

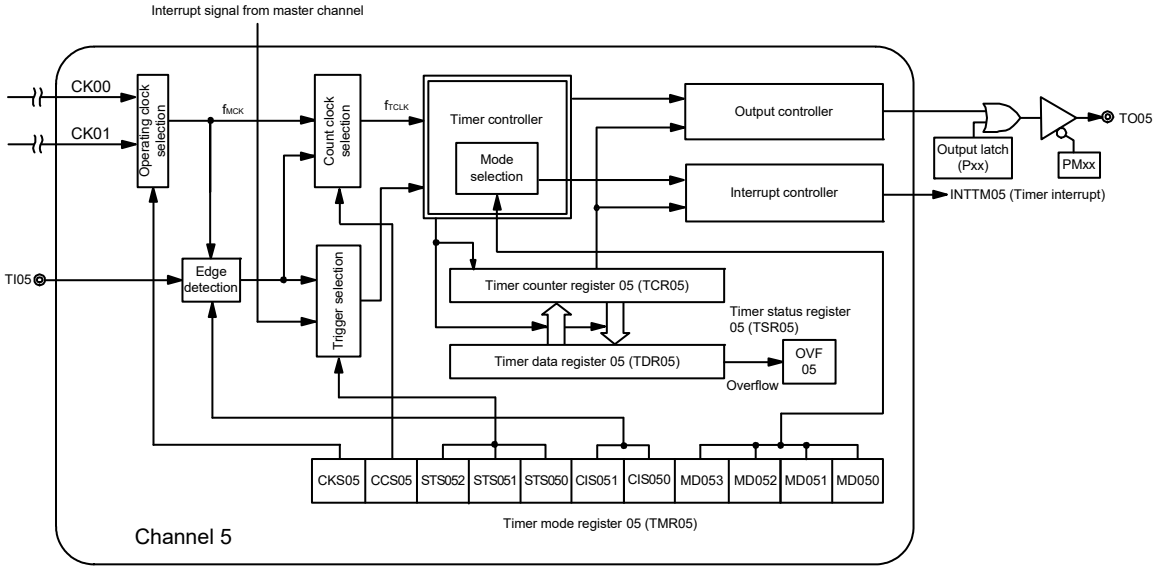
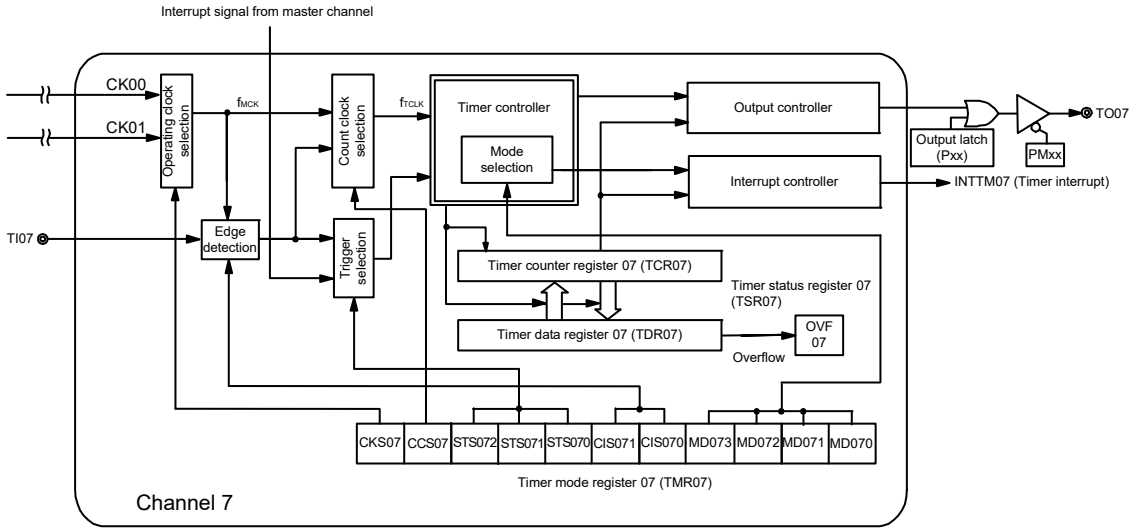


Figure 6 - 7 Internal Block Diagram of Channel 7 of Timer Array Unit



6.2.1 Timer count register mn (TCRmn)

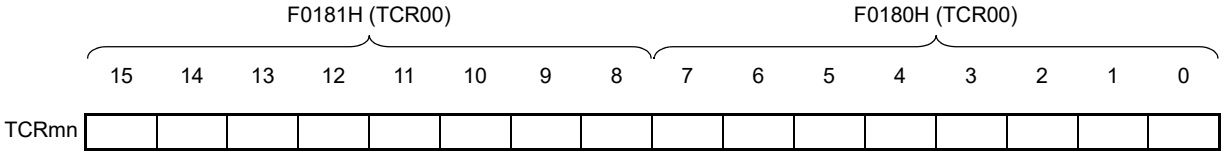
The TCRmn register is a 16-bit read-only register and is used to count clocks.

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 MDmn3 to MDmn0 bits of timer mode register mn (TMRmn) (refer to **6.3.3 Timer mode register mn (TMRmn)**).

Figure 6 - 8 Format of Timer count register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07) After reset: FFFFH R



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

The count value can be read by reading timer count register mn (TCRmn).

The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAUmEN bit of peripheral enable register 0 (PER0) is cleared
- 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 0000H in the following cases.
- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register mn (TDRmn) even when the TCRmn register is read.

The TCRmn register read value differs as follows according to operation mode changes and the operating status.

Table 6 - 3 Timer Count Register mn (TCRmn) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer count register mn (TCRmn) Read Value Note			
		Value if the operation mode was changed after releasing reset	Value if the operation was restarted after count operation paused (TTmn = 1)	Value if the operation mode was changed after count operation paused (TTmn = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	—
Capture mode	Count up	0000H	Value if stop	Undefined	—
Event counter mode	Count down	FFFFH	Value if stop	Undefined	—
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of TDRmn register + 1

Note This indicates the value read from the TCRmn register when channel n has stopped operating as a timer (TEmn = 0) and has been enabled to operate as a counter (TSmn = 1). The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.2.2 Timer data register mn (TDRmn)

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 MDmn3 to MDmn0 bits of timer mode register mn (TMRmn).

The value of the TDRmn register can be changed at any time.

This register can be read or written in 16-bit units.

In addition, for the TDRm1 and TDRm3 registers, while in the 8-bit timer mode (when the SPLIT bits of timer mode registers 01 and 03 (TMRm1, TMRm3) are 1), it is possible to rewrite the data in 8-bit units, with TDRm1H and TDRm3H used as the higher 8 bits, and TDRm1L and TDRm3L used as the lower 8 bits. However, reading is only possible in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 6 - 9 Format of Timer data register mn (TDRmn) (n = 0, 2, 4 to 7)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02) After reset: 0000H R/W
FFF68H, FFF69H (TDR04) to FFF6EH, FFF6FH (TDR07)

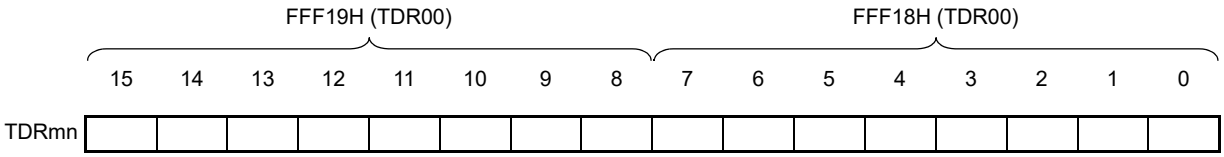
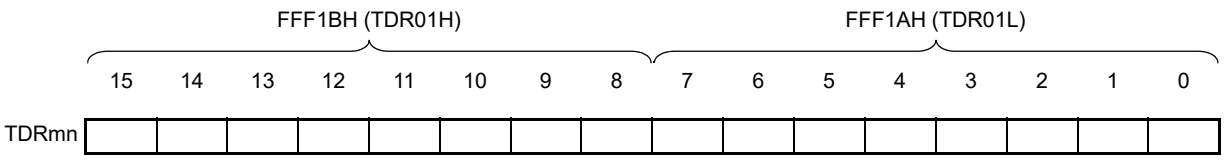


Figure 6 - 10 Format of Timer data register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) After reset: 00H R/W



(i) When timer data register mn (TDRmn) is used as compare register

Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

Caution The TDRmn 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 mn (TDRmn) is used as capture register

The count value of timer count register mn (TCRmn) is captured to the TDRmn register when the capture trigger is input.

A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by timer mode register mn (TMRmn).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer input select register 0 (TIS0)
- Timer output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Noise filter enable register 1 (NFEN1)
- Port mode control register (PMCxx)
- Port mode register (PMxx)
- Port register (Pxx)

Caution **Which registers and bits are included depends on the product. Be sure to set bits that are not mounted to their initial values.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.1 Peripheral enable register 0 (PER0)

This registers is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the timer array unit 0 is used, be sure to set bit 0 (TAU0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6 - 11 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	IICA2EN Note	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

TAU0EN	Control of timer array unit input clock
0	Stops supply of input clock. • SFR used by the timer array unit cannot be written. • The timer array unit is in the reset status.
1	Supplies input clock. • SFR used by the timer array unit can be read/written.

Note This bit is incorporated with R9A02G0151, but is not incorporated with R9A02G0150.

Caution 1. When setting the timer array unit, be sure to set the following registers first while the TAUmEN bit is set to 1. If TAUmEN = 0, the values of the registers which control the timer array unit are cleared to their initial values and writing to them is ignored (except for the timer input select register (TIS0), noise filter enable register 1 (NFEN1), port mode control register 0 (PMC0), port mode registers 0, 5, 7 (PM0, PM5, PM7), and port registers 0, 5, 7 (P0, P5, P7)).

- **Timer clock select register m (TPSm)**
- **Timer mode register mn (TMRmn)**
- **Timer status register mn (TSRmn)**
- **Timer channel enable status register m (TEm)**
- **Timer channel start register m (TSm)**
- **Timer channel stop register m (TTm)**
- **Timer output enable register m (TOEm)**
- **Timer output register m (TOm)**
- **Timer output level register m (TOLm)**
- **Timer output mode register m (TOMm)**

Caution 2. Be sure to clear bits 1 and 6 to 0.

6.3.2 Timer clock select register m (TPSm)

The TPSm register is a 16-bit register that is used to select two types or four types of operation clocks (CKm0, CKm1, CKm2, CKm3) that are commonly supplied to each channel. CKm0 is selected by using bits 3 to 0 of the TPSm register, and CKm1 is selected by using bits 7 to 4 of the TPSm register. In addition, only for channels 1 and 3, CKm2 and CKm3 can be also selected. CKm2 is selected by using bits 9 and 8 of the TPSm register, and CKm3 is selected by using bits 13 and 12 of the TPSm register.
Rewriting of the TPSm register during timer operation is possible only in the following cases.

If the PRSm00 to PRSm03 bits can be rewritten (n = 0 to 7):
All channels for which CKm0 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).
If the PRSm10 to PRSm13 bits can be rewritten (n = 0 to 7):
All channels for which CKm2 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).
If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):
All channels for which CKm1 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 0) are stopped (TEmn = 0).
If the PRSm30 and PRSm31 bits can be rewritten (n = 1, 3):
All channels for which CKm3 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0000H.

Figure 6 - 12 Format of Timer clock select register m (TPSm) (1/2)

Address: F01B6H, F01B7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRSm 31	PRSm 30	0	0	PRSm 21	PRSm 20	PRSm 13	PRSm 12	PRSm 11	PRSm 10	PRSm 03	PRSm 02	PRSm 01	PRSm 00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operation clock (CKmk) Note (k = 0, 1)					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61.0 Hz	153 Hz	305 Hz	610 Hz	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).

Caution 1. Be sure to clear bits 15, 14, 11, and 10 to "0".

Caution 2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0, m = 0 to 7), interrupt requests output from timer array units cannot be used.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. The above $f_{CLK}/2^r$ is not a signal which is simply divided f_{CLK} by 2r, but a signal which becomes high level for one period of f_{CLK} from its rising edge (r = 1 to 15). For details, see **6.5.1 Count clock (f_{TCLK})**.

Figure 6 - 13 Format of Timer clock select register m (TPSm) (2/2)

Address: F01B6H, F01B7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRSm31	PRSm30	0	0	PRSm21	PRSm20	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRS m21	PRS m20	Selection of operation clock (CKm2) Note					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	1	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
1	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
1	1	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156.2 kHz	313 kHz	375 kHz

PRS m31	PRS m30	Selection of operation clock (CKm3) Note					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
0	1	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop the timer array unit (TTm = 00FFH).
The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected.

Caution Be sure to clear bits 15, 14, 11, and 10 to "0".

By using channels 1 and 3 in the 8-bit timer mode and specifying CKm2 or CKm3 as the operation clock, the interval times shown in Table 6 - 4 can be achieved by using the interval timer function.

Table 6 - 4 Interval Times Available for Operation Clock CKSm2 or CKSm3

Clock		Interval time Note (f_{CLK} = 20 MHz)			
		16 µs	160 µs	1.6 ms	16 ms
CKm2	$f_{CLK}/2$	✓	—	—	—
	$f_{CLK}/2^2$	✓	—	—	—
	$f_{CLK}/2^4$	✓	✓	—	—
	$f_{CLK}/2^6$	✓	✓	—	—
CKm3	$f_{CLK}/2^8$	—	✓	✓	—
	$f_{CLK}/2^{10}$	—	✓	✓	—
	$f_{CLK}/2^{12}$	—	—	✓	✓
	$f_{CLK}/2^{14}$	—	—	✓	✓

Note The margin is within 5%.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. For details of a signal of $f_{CLK}/2^i$ selected with the TPSm register, see **6.5.1 Count clock (f_{TCLK})**.

6.3.3 Timer mode register mn (TMRmn)

The TMRmn 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/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 and one-count).

Rewriting the TMRmn register is prohibited when the register is in operation (when TEmn = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEmn = 1) (for details, see **6.7 Timer Input (TImn) Control** and **6.9 Simultaneous Channel Operation Function of Timer Array Unit**).

The TMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Caution The bits mounted depend on the channels in bit 11 of TMRmn register.
TMRm2, TMRm4, TMRm6: MASTERmn bit (n = 2, 4, 6)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0, TMRm5, TMRm7: Fixed to 0

Figure 6 - 14 Format of Timer mode register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn1	CKSmn0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CKm0 set by timer clock select register m (TPSm)
0	1	Operation clock CKm2 set by timer clock select register m (TPSm)
1	0	Operation clock CKm1 set by timer clock select register m (TPSm)
1	1	Operation clock CKm3 set by timer clock select register m (TPSm)
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 CCSmn bit. The operation clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

CCSmn	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
1	Valid edge of input signal input from the TImn pin In channel 1, Valid edge of input signal selected by TIS0
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Note Bit 11 is fixed at 0 of read only, write is ignored.

Caution 1. Be sure to clear bits 13, 5, and 4 to "0".

Caution 2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn0 and CKSmn1 bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 15 Format of Timer mode register mn (TMRmn) (2/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

(Bit 11 of TMRmn (n = 2, 4, 6))

MASTERmn	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.
Only channel 2, 4, 6 can be set as a master channel (MASTERmn = 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 MASTERmn bit to 0 for a channel that is used with the independent channel operation function.	

(Bit 11 of TMRmn (n = 1, 3))

SPLITmn	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.

STSmn2	STSmn1	STSmn0	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 TImn pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of the TImn 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).
Other than above			Setting prohibited

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 16 Format of Timer mode register mn (TMRmn) (3/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CISmn1	CISmn0	Selection of TImn pin input valid edge (n = 0, 1)
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
If both the edges are specified when the value of the STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B.		

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 17 Format of Timer mode register mn (TMRmn) (4/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0 Note 1	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

MDmn3	MDmn2	MDmn1	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer / Square wave output / PWM output (master)	Counting down
0	1	0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	Event counter mode	External event counter	Counting down
1	0	0	One-count mode	Delay counter / One-shot pulse output / PWM output (slave)	Counting down
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than above			Setting prohibited		
The operation of each mode varies depending on MDmn0 bit (see the table below).					

Operation mode (Value set by the MDmn3 to MDmn1 bits (see the table above))	MDmn0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0) • Capture mode (0, 1, 0)	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 (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode Note 2 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
	1	Start trigger is valid during counting operation Note 3. At that time, interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	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, either.

Note 1. Bit 11 is fixed at 0 of read only, write is ignored.

Note 2. In one-count mode, interrupt output (INTTMmn) when starting a count operation and TOmn output are not controlled.

Note 3. If the start trigger (TSmn = 1) is issued during operation, the counter is initialized, an interrupt is generated, and recounting is started (does not occur the interrupt request).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.4 Timer status register mn (TSRmn)

The TSRmn register indicates the overflow status of the counter of channel n.

The TSRmn register is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). See **Table 6 - 5** for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSRmn register can be set with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

Figure 6 - 18 Format of Timer status register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	Counter overflow status of channel n
0	Overflow does not occur.
1	Overflow occurs.
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Table 6 - 5 OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF bit	Set/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	

Remark The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

6.3.5 Timer channel enable status register m (TEm)

The TEm register is used to enable or stop the timer operation of each channel.

Each bit of the TEm register corresponds to each bit of the timer channel start register m (TSm) and the timer channel stop register m (TTm). When a bit of the TSm register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TTm register is set to 1, the corresponding bit of this register is cleared to 0.

The TEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 19 Format of Timer channel enable status register m (TEm)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	TEHm3	0	TEHm1	0	TEm7	TEm6	TEm5	TEm4	TEm3	TEm2	TEm1	TEm0

TEHm3	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.

TEHm1	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.

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TEm1 and TEm3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.6 Timer channel start register m (TSm)

The TSm register is a trigger register that is used to initialize timer count register mn (TCRmn) 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 m (TEm) is set to 1. The TSmn, TSHm1, TSHm3 bits are immediately cleared when operation is enabled (TEmn, TEHm1, TEHm3 = 1), because they are trigger bits.

The TSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSm register can be set with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 20 Format of Timer channel start register m (TSm)

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	TSHm3	0	TSHm1	0	TSm7	TSm6	TSm5	TSm4	TSm3	TSm2	TSm1	TSm0

TSHm3	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 TEHm3 bit is set to 1 and the count operation becomes enabled. The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6 - 6** in **6.5.2 Start timing of counter**).

TSHm1	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 TEHm1 bit is set to 1 and the count operation becomes enabled. The TCRm1 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6 - 6** in **6.5.2 Start timing of counter**).

TSmn	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TEmn bit is set to 1 and the count operation becomes enabled. The TCRmn register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6 - 6** in **6.5.2 Start timing of counter**). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TSm1 and TSm3 when channel 1 or 3 is in the 8-bit timer mode.

Caution 1. Be sure to clear bits 15 to 12, 10, and 8 to "0".

Caution 2. When switching from a function that does not use TImn pin input to one that does, the following wait period is required from when timer mode register mn (TMRmn) is set until the TSmn (TSHm1, TSHm3) bit is set to 1.
When the TImn pin noise filter is enabled (TNFENmn = 1): Four cycles of the operation clock (f_{MCK})
When the TImn pin noise filter is disabled (TNFENmn = 0): Two cycles of the operation clock (f_{MCK})

Remark 1. When the TSm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.7 Timer channel stop register m (TTm)

The TTm 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 m (TEm) is cleared to 0. The TTmn, TTHm1, TTHm3 bits are immediately cleared when operation is stopped (TEmn, TEHm1, TEHm3 = 0), because they are trigger bits.

The TTm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TTm register can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 21 Format of Timer channel stop register m (TTm)

Address: F01B4H, F01B5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	TTHm3	0	TTHm1	0	TTm7	TTm6	TTm5	TTm4	TTm3	TTm2	TTm1	TTm0

TTHm3	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	TEHm3 bit is cleared to 0 and the count operation is stopped.

TTHm1	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	TEHm1 bit is cleared to 0 and the count operation is stopped.

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	TEmn bit is cleared to 0 and the count operation is stopped. This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 is in the 8-bit timer mode.

Caution Be sure to clear bits 15 to 12, 10, and 8 of the TTm register to "0".

Remark 1. When the TTm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.8 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channel 0 and 1 timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6 - 22 Format of Timer input select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
Other than above			Setting prohibited

Caution At least 1/f_{MCK} + 10 ns is necessary as the high-level and low-level widths of the timer input to be selected.

6.3.9 Timer output enable register m (TOEm)

The TOEm 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 TOmn bit of timer output register m (TOm) 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 (TOmn).

The TOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TOEmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 23 Format of Timer output enable register m (TOEm)

Address: F01BAH, F01BBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEm	0	0	0	0	0	0	0	0	TOEm7	TOEm6	TOEm5	TOEm4	TOEm3	TOEm2	TOEm1	TOEm0

TOEmn	Timer output enable/disable of channel n
0	Timer output is disabled. Timer operation is not applied to the TOmn bit and the output is fixed. Writing to the TOmn bit is enabled and the level set in the TOmn bit is output from the TOmn pin.
1	Timer output is enabled. Timer operation is applied to the TOmn bit and an output waveform is generated. Writing to the TOmn bit is ignored.

Caution Be sure to clear bits 15 to 8 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.10 Timer output register m (TOm)

The TOm 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 (TOmn) of each channel.

The TOmn bit of this register can be rewritten by software only when timer output is disabled (TOEmn = 0). When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P51/TI00/TO00, P52/TI01/TO01, P53/TI02/TO02, P54/TI03/TO03, P55/TI04/TO04, P00/TI05/TO05, P01/TI06/TO06, P74/TI07/TO07 pin as a port function pin, set the corresponding TOmn bit to “0”.

The TOm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOm register can be set with an 8-bit memory manipulation instruction with TOmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 24 Format of Timer output register m (TOm)

Address: F01B8H, F01B9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOm	0	0	0	0	0	0	0	0	TOm7	TOm6	TOm5	TOm4	TOm3	TOm2	TOm1	TOm0

TOmn	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution Be sure to clear bits 15 to 8 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.11 Timer output level register m (TOLm)

The TOLm register is a register that 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 (TOEmn = 1) in the Slave channel output mode (TOMmn = 1). In the master channel output mode (TOMmn = 0), this register setting is invalid.

The TOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOLm register can be set with an 8-bit memory manipulation instruction with TOLmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 25 Format of Timer output level register m (TOLm)

Address: F01BCH, F01BDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	TOLm7	TOLm6	TOLm5	TOLm4	TOLm3	TOLm2	TOLm1	0

TOLmn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative logic output (active-low)

Caution Be sure to clear bits 15 to 8, and 0 to “0”.

Remark 1. 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.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.12 Timer output mode register m (TOMm)

The TOMm 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 (TOEmn = 1).

The TOMm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOMm register can be set with an 8-bit memory manipulation instruction with TOMmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 26 Format of Timer output mode register m (TOMm)

Address: F01BEH, F01BFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	0	TOMm7	TOMm6	TOMm5	TOMm4	TOMm3	TOMm2	TOMm1	0

TOMmn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTMmn)
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Caution **Be sure to clear bits 15 to 8, and 0 to “0”.**

Remark 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
(For details of the relation between the master channel and slave channel, refer to **6.4.1 Basic rules of simultaneous channel operation function**.)

6.3.13 Noise filter enable register 1 (NFEN1)

The NFEN1 register is 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 Note.

The NFEN1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note For details, see **6.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)**, **6.5.2 Start timing of counter**, and **6.7 Timer Input (TImn) Control**.

Figure 6 - 27 Format of Noise filter enable register 1 (NFEN1)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	Enable/disable using noise filter of TI07 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN06	Enable/disable using noise filter of TI06 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00 pin input signal
0	Noise filter OFF
1	Noise filter ON

6.3.14 Registers that control port functions of timer input/output pins

Using the timer array unit requires setting of the registers that control the port functions for the port pins with which the timer array unit pin functions for the target channel are multiplexed (port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, and **4.3.6 Port mode control registers (PMCxx)**.
The port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx) to be set depend on the product. For details, see **4.5.3 Register setting examples for used port and alternate functions**.

Using a port pin which is multiplexed with a timer output pin function (e.g. P51/TI00/TO00, P53/TI02/TO02) for timer output requires setting the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example When P53/TO02/TI02 is to be used for timer output
Set the PM53 bit of port mode register 0 to 0.
Set the P53 bit of port register 0 to 0.

Using a port pin which is multiplexed with a timer input pin function (e.g. P51/TI00/TO00, P53/TI02/TO02) for timer input requires setting the corresponding bit in the port mode register (PMxx) to 1. At this time, the value of the corresponding bit in the port register (Pxx) may be 0 or 1.

Example When P53/TO02/TI02 is to be used for timer input
Set the PM53 bit of port mode register 0 to 1.
Set the P53 bit of port register 0 to 0 or 1.

Remark The P00/TI05/TO05/ANI16 and P01/TI06/TO06 pins are multiplexed with an analog input pin function. When using the timer I/O function, be sure to set the corresponding bit of the PMC0 register which switches between digital I/O and analog input to “0”.

6.4 Basic Rules of Timer Array Unit

6.4.1 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 (channel 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.

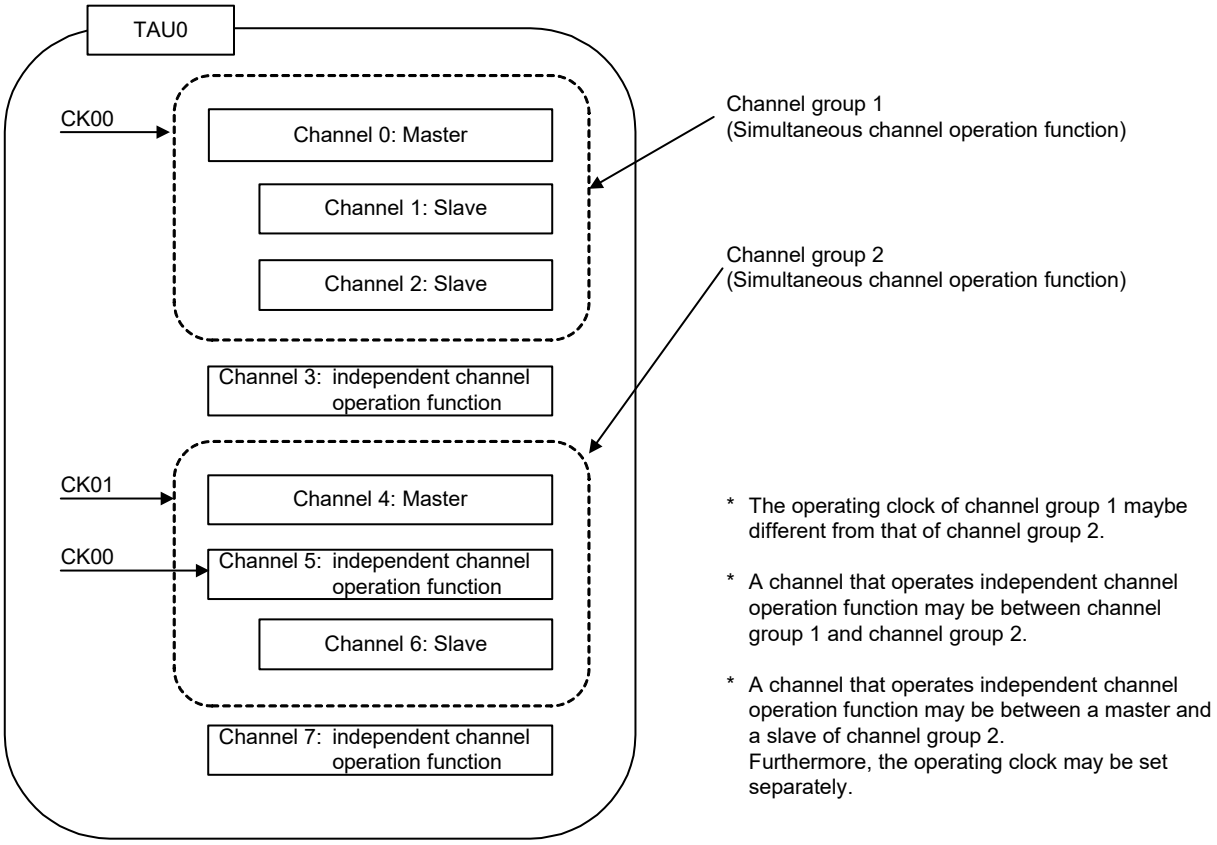
(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 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 CKSmn0, CKSmn1 bits (bit 15, 14 of timer mode register mn (TMRmn)) 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 INTTMmn (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTMmn (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 (TSmn) of the channels in combination must be set at the same time.
(11) During the counting operation, a TSmn bit of a master channel or TSmn bits of all channels which are operating simultaneously can be set. It cannot be applied to TSmn bits of slave channels alone.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.
(13) CKm2/CKm3 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 m0 (TMRm0) has no master bit (it is fixed as “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 **6.4.1 Basic rules of simultaneous channel operation function** do not apply to the channel groups.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Example



6.4.2 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 mn (TMRmn) 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 INTTMm1H/INTTMm3H (an interrupt) (which is the same operation performed when MDmn0 is set to 1).

(5) The operation clock of the higher 8 bits is selected according to the CKSmn1 and CKSmn0 bits of the lower-bit TMRmn register.

(6) For the higher 8 bits, the TSHm1/TSHm3 bit is manipulated to start channel operation and the TTHm1/TTHm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEHm1/TEHm3 bit.

(7) The lower 8 bits operate according to the TMRmn register settings. The following three functions support operation of the lower 8 bits:
- Interval timer function
- External event counter function
- Delay count function

(8) For the lower 8 bits, the TSm1/TSm3 bit is manipulated to start channel operation and the TTm1/TTm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEm1/TEm3 bit.

(9) During 16-bit operation, manipulating the TSHm1, TSHm3, TTHm1, and TTHm3 bits is invalid. The TSm1, TSm3, TTm1, and TTm3 bits are manipulated to operate channels 1 and 3. The TEHm3 and TEHm1 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.

Remark m: Unit number (m = 0), n: Channel number (n = 1, 3)

6.5 Operation of Counter

6.5.1 Count clock (fTCLK)

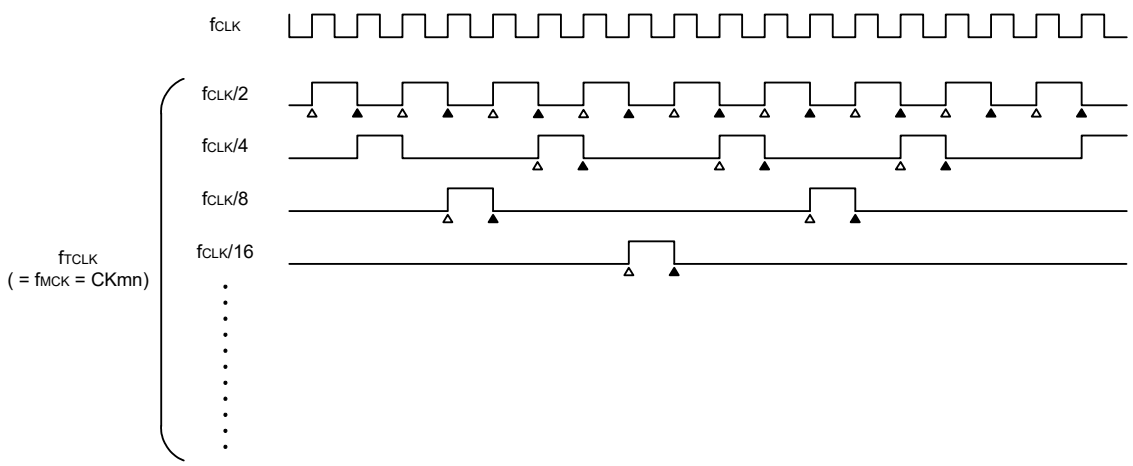
The count clock (fTCLK) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

- Operation clock (fMCK) specified by the CKSmn0 and CKSmn1 bits
- Valid edge of input signal input from the TImn pin

Because the timer array unit is designed to operate in synchronization with fCLK, the timings of the count clock (fTCLK) are shown below.

(1) When operation clock (fMCK) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)
The count clock (fTCLK) is between fCLK to fCLK /2^{15} by setting of timer clock select register m (TPSm). When a divided fCLK is selected, however, the clock selected in TPSmn register, but a signal which becomes high level for one period of fCLK from its rising edge. When a fCLK is selected, fixed to high level. Counting of timer count register mn (TCRmn) delayed by one period of fCLK from rising edge of the count clock, because of synchronization with fCLK. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

Figure 6 - 28 Timing of fCLK and count clock (fTCLK) (When CCSmn = 0)



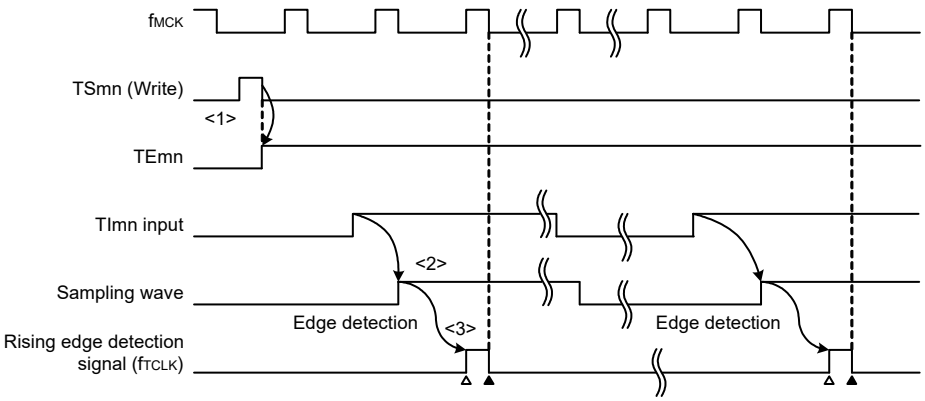
Remark 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter
Remark 2. fCLK: CPU/peripheral hardware clock

(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 periods of f_{MCK} from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clocks).

Counting of timer count register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at valid edge of input signal via the TImn pin”, as a matter of convenience.

Figure 6 - 29 Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 1, noise filter unused)



<1> Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TImn pin.
<2> The rise of input signal via the TImn 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.

Remark 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter

Remark 2. f_{CLK}: CPU/peripheral hardware clock
f_{MCK}: Operation clock of channel n

Remark 3. The waveform of the input signal via TImn pin of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter, and the one-shot pulse output are the same as that shown in Figure 6 - 29.

6.5.2 Start timing of counter

Timer count register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSm).

Operations from count operation enabled state to timer count Register mn (TCRmn) count start is shown in Table 6 - 6.

Table 6 - 6 Operations from Count Operation Enabled State to Timer count Register mn (TCRmn) Count Start

Timer operation mode	Operation when TSmn = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TSmn = 1) until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (1) Operation of interval timer mode**).
• Event counter mode	Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register. If detect edge of TImn input. The subsequent count clock performs count down operation. (see **6.5.3 (2) Operation of event counter mode**).
• Capture mode	No operation is carried out from start trigger detection (TSmn = 1) until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (3) Operation of capture mode (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (4) Operation of one-count mode**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (5) Start timing in capture & one-count mode (when high-level width is measured)**).

6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer count register mn (TCRmn) holds the initial value until count clock generation.

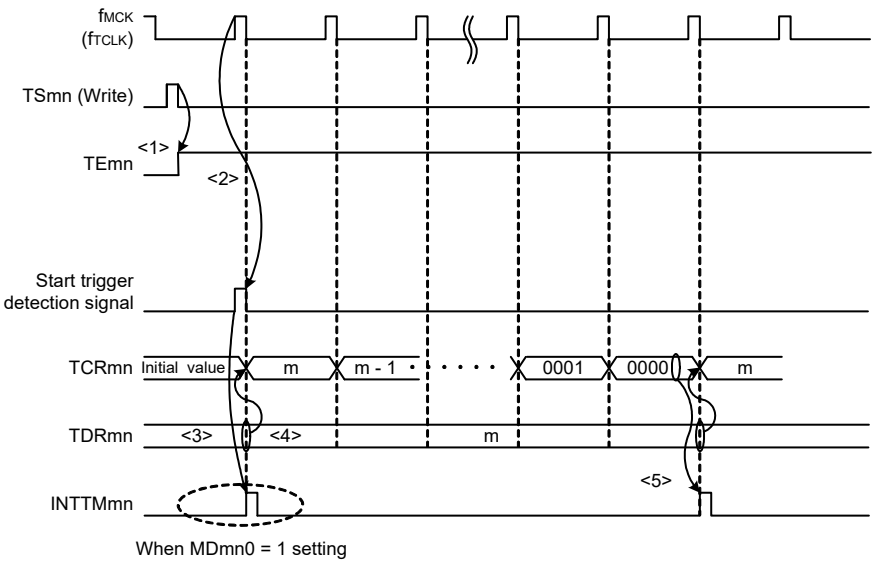
<2> A start trigger is generated at the first count clock after operation is enabled.

<3> When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting starts in the interval timer mode.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

Figure 6 - 30 Operation Timing (In Interval Timer Mode)



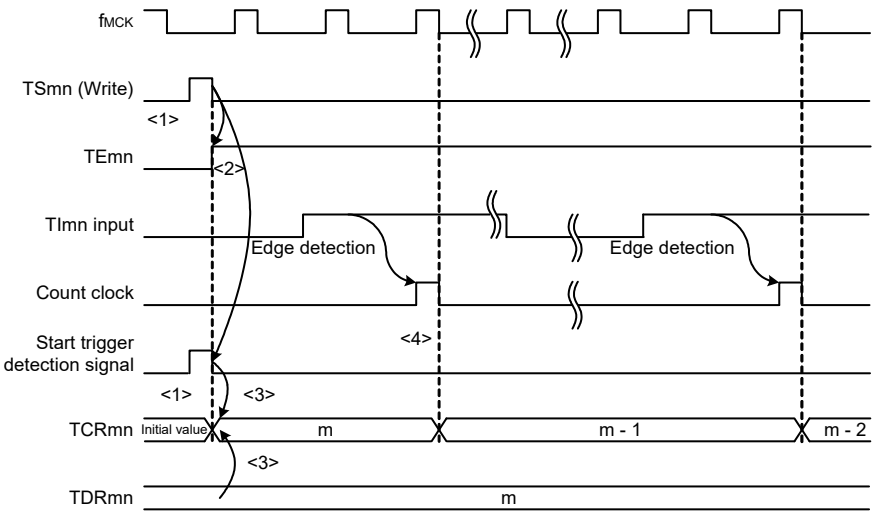
Caution **In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark fMCK, the start trigger detection signal, and INTTMmn become active between one clock in synchronization with fCLK.

(2) Operation of event counter mode

<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).
<2> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<3> As soon as 1 has been written to the TSmn bit and 1 has been set to the TEmn bit, the value of timer data register mn (TDRmn) is loaded to the TCRmn register to start counting.
<4> After that, the TCRmn register value is counted down according to the count clock of the valid edge of the TImn input.

Figure 6 - 31 Operation Timing (In Event Counter Mode)

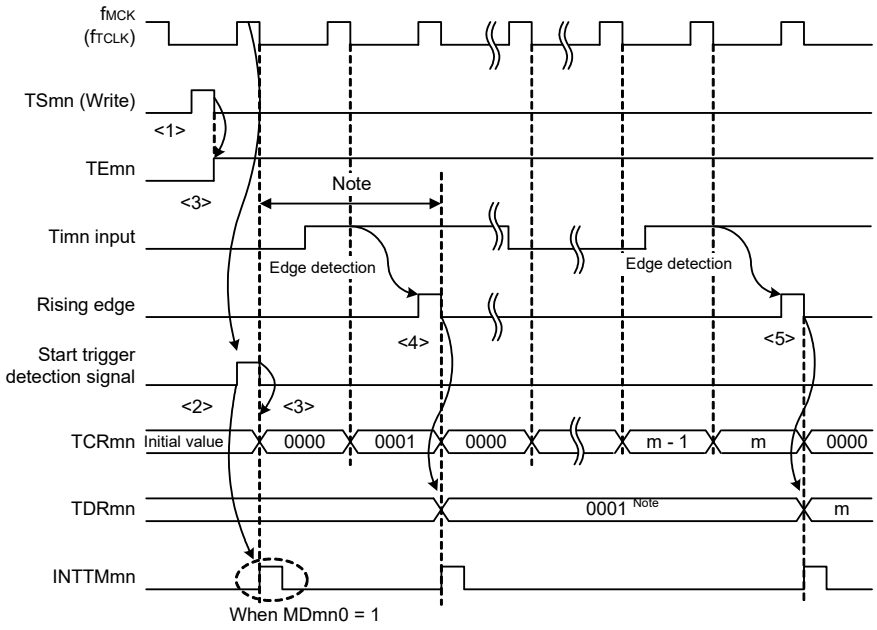


Remark The above figure shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(3) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<2> Timer count register mn (TCRmn) 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 0000H is loaded to the TCRmn register and counting starts in the capture mode. (When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.)
<4> On detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated. However, this capture value is no meaning. The TCRmn register keeps on counting from 0000H.
<5> On next detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6 - 32 Operation Timing (In Capture Mode: Input Pulse Interval Measurement)



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, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Caution **In the first cycle operation of count clock after writing the TSmn 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 MDmn0 = 1.**

Remark The above figure shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK 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 (fMCK).

(4) Operation of one-count mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

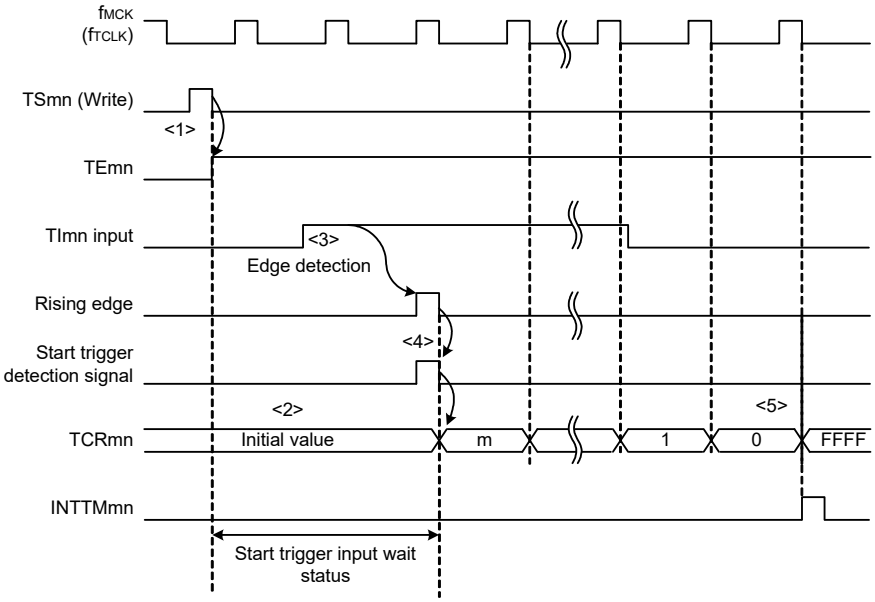
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and count starts.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of the TCRmn register becomes FFFFH and counting stops.

Figure 6 - 33 Operation Timing (In One-count Mode)



Remark The above figure shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK 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 (fMCK).

(5) Start timing in capture & one-count mode (when high-level width is measured)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).

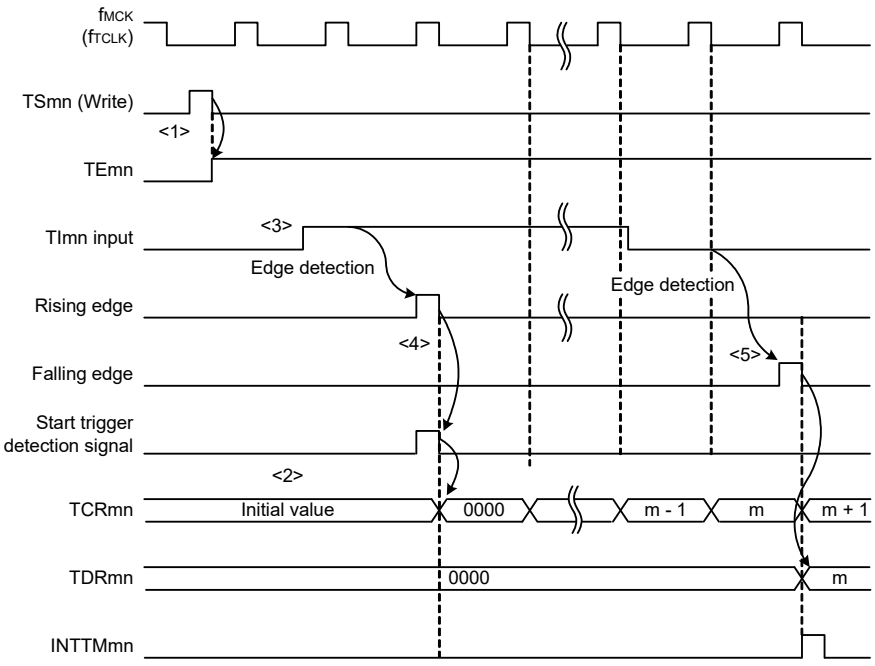
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of 0000H is loaded to the TCRmn register and count starts.

<5> On detection of the falling edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6 - 34 Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

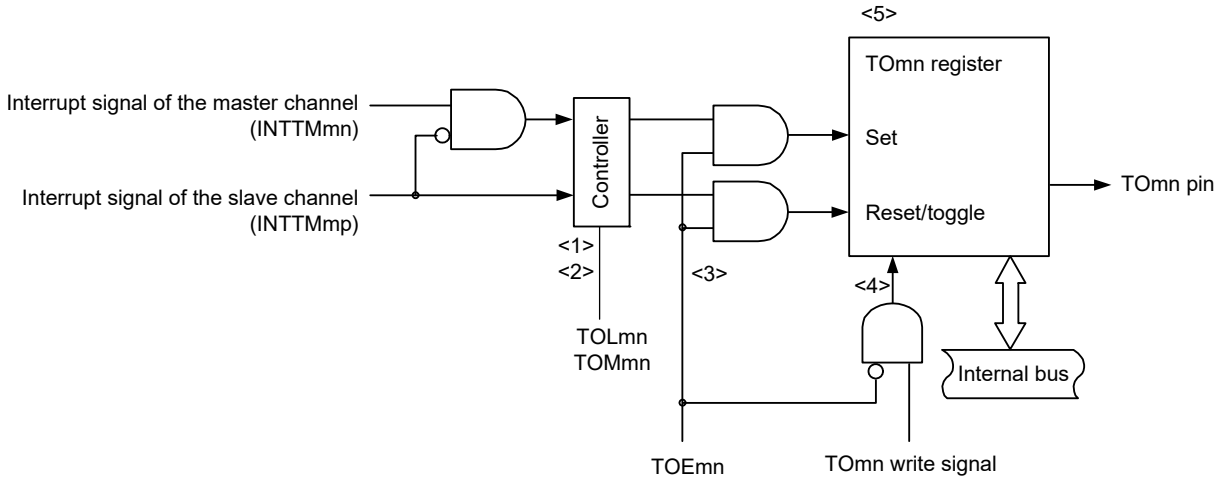


Remark The above figure shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

6.6 Channel Output (TOmn pin) Control

6.6.1 TOmn pin output circuit configuration

Figure 6 - 35 Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:

When TOLmn = 0: Positive logic output (INTTMmn → set, INTTM0p → reset)
When TOLmn = 1: Negative logic output (INTTMmn → reset, INTTM0p → set)

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation is stopped (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TOm register.

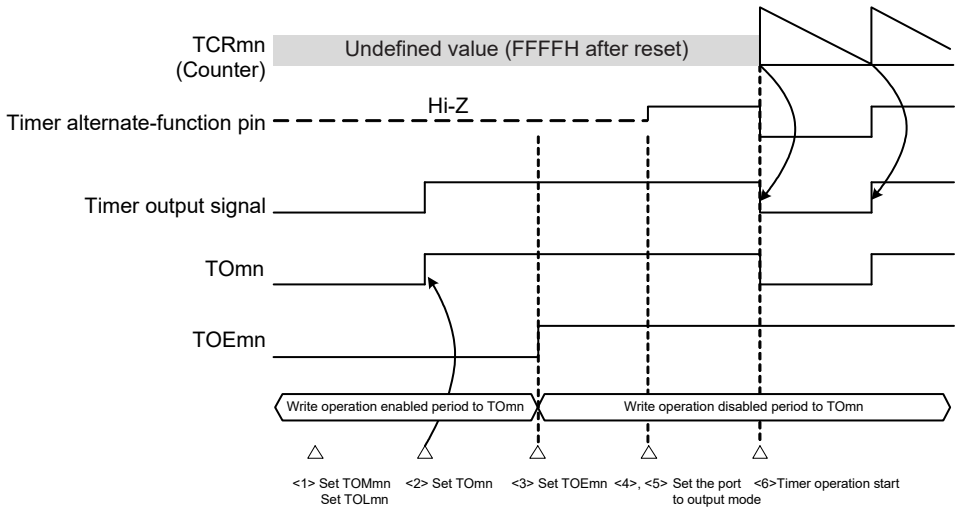
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

Remark 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

6.6.2 TOmn Pin Output Setting

The following figure shows the procedure and status transition of the TOmn output pin from initial setting to timer operation start.

Figure 6 - 36 Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn bit (0: Positive logic output, 1: Negative logic output)

<2> The timer output signal is set to the initial status by setting timer output register m (TOm).

<3> The timer output operation is enabled by writing 1 to the TOEmn bit (writing to the TOm register is disabled).

<4> The port is set to digital I/O by port mode control register (PMCxx).

<5> The port I/O setting is set to output (see **6.3.14 Registers that control port functions of timer input/output pins**).

<6> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.6.3 Cautions on Channel Output Operation

(1) Changing values set in the registers TOm, TOEm, and TOLm during timer operation

Since the timer operations (operations of timer count register mn (TCRmn) and timer data register mn (TDRmn)) are independent of the TOmn output circuit and changing the values set in timer output register m (TOm), timer output enable register m (TOEm), and timer output level register m (TOLm) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOmn pin by timer operation, however, set the TOm, TOEm, TOLm, and TOMm registers to the values stated in the register setting example of each operation shown by 6.7 and 6.8.

When the values set to the TOEm, and TOMm registers (but not the TOm register) are changed close to the occurrence of the timer interrupt (INTTMmn) of each channel, the waveform output to the TOmn pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) occurs.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

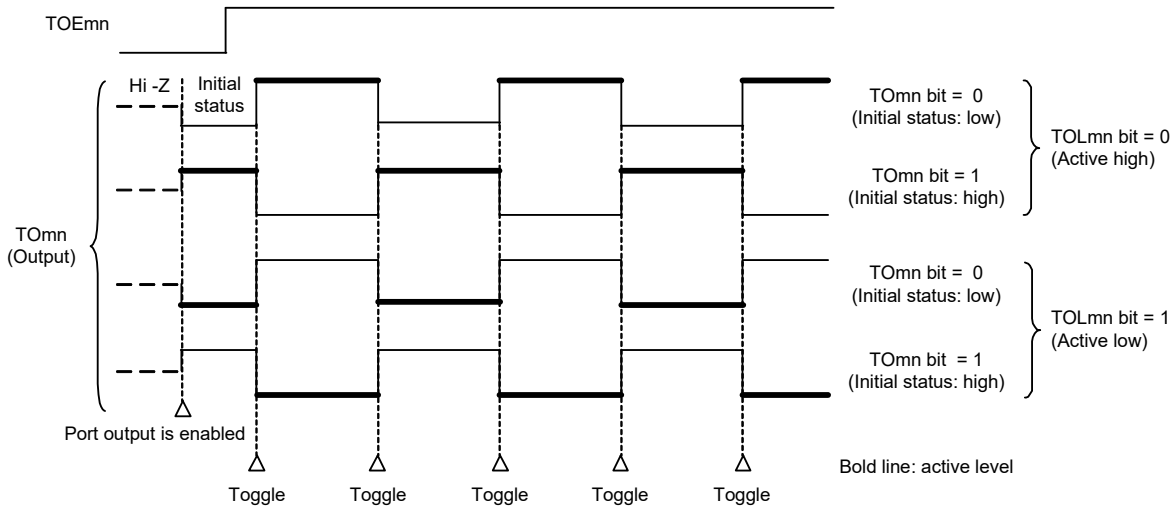
(2) Default level of TOmn pin and output level after timer operation start

The change in the output level of the TOmn pin when timer output register m (TOm) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOMmn = 0) setting

The setting of timer output level register m (TOLm) is invalid when master channel output mode (TOMmn = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TOmn pin is reversed.

Figure 6 - 37 TOmn Pin Output Status at Toggle Output (TOMmn = 0)

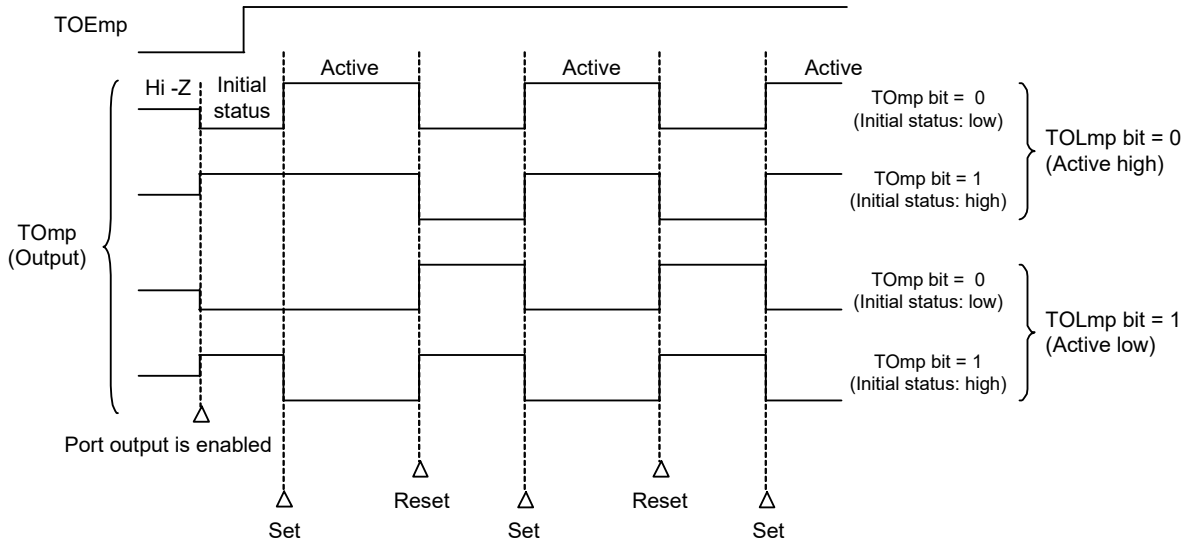


Remark 1. Toggle: Reverse TOmn pin output status

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

(b) When operation starts with slave channel output mode (TOMmp = 1) setting (PWM output))
When slave channel output mode (TOMmp = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 6 - 38 TOmp Pin Output Status at PWM Output (TOMmp = 1)



Remark 1. 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.

Remark 2. m: Unit number (m = 0), p: Channel number (p = 1 to 7)

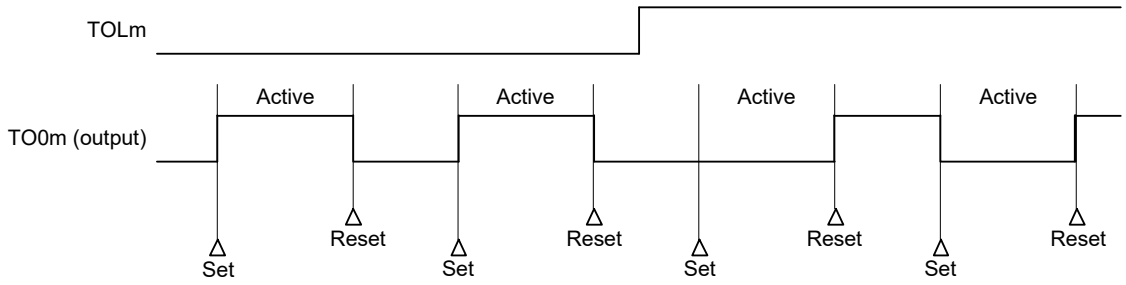
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(a) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOmn pin change condition. Rewriting the TOLm register does not change the output level of the TOmn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

Figure 6 - 39 Operation when TOLm Register Has Been Changed Contents during Timer Operation



Remark 1. 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.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TOmn pin/TOmn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

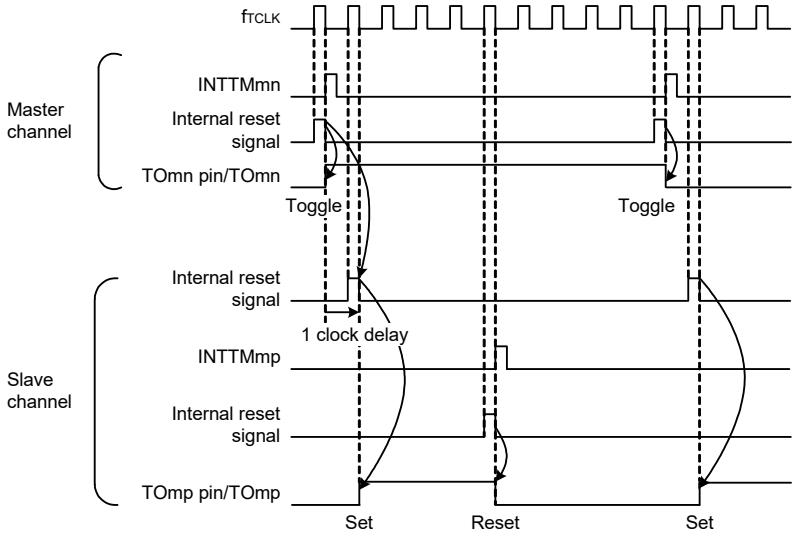
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6 - 40 shows the set/reset operating statuses where the master/slave channels are set as follows.

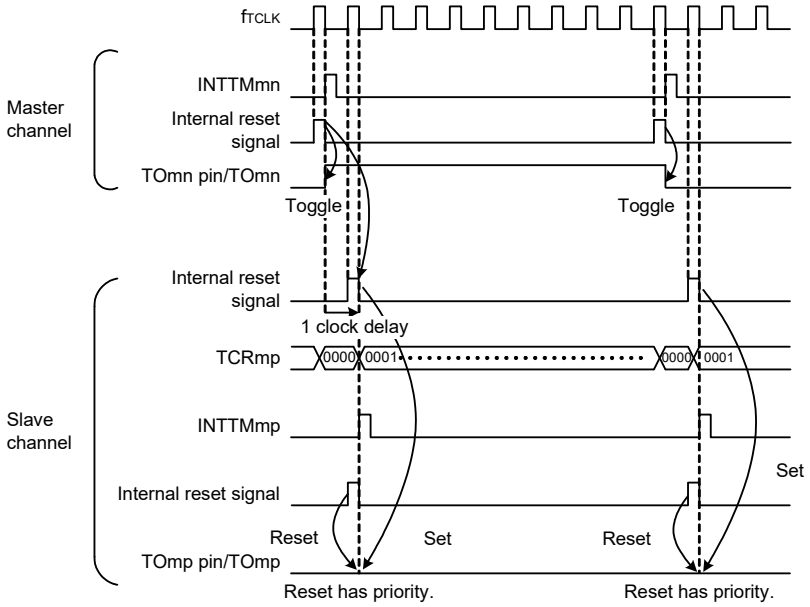
Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 6 - 40 Set/Reset Timing Operating Statuses

(1) Basic operation timing



(2) Operation timing when 0% duty



Remark 1. Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

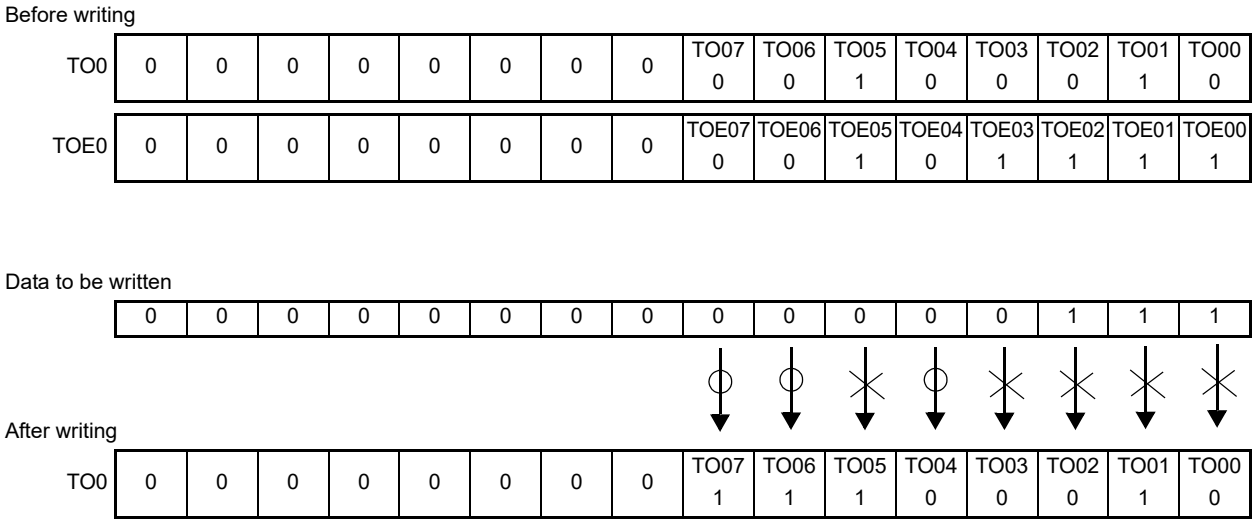
Remark 2. m: Unit number (m = 0)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

6.6.4 Collective manipulation of TOmn bit

In timer output register m (TOm), the setting bits for all the channels are located in one register in the same way as timer channel start register m (TSm). Therefore, the TOmn bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TOmn bits (TOEmn = 0) that correspond to the relevant bits of the channel used to perform output (TOmn).

Figure 6 - 41 Example of TO0n Bit Collective Manipulation



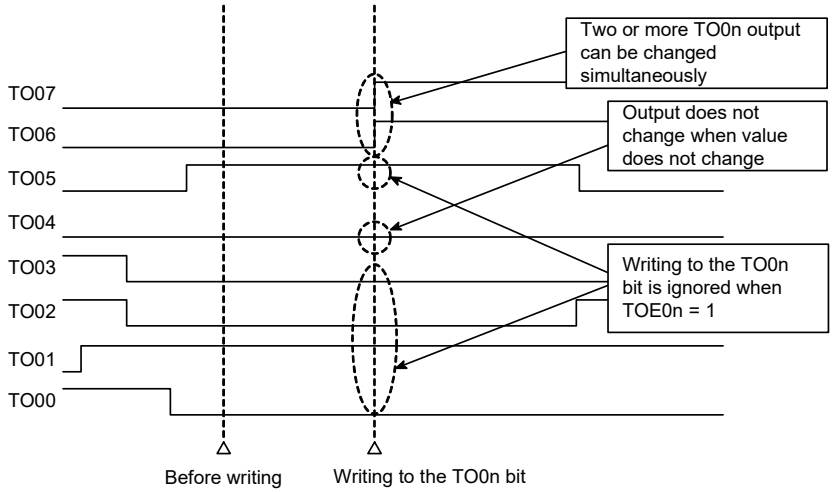
									TO07	TO06	TO05	TO04	TO03	TO02	TO01	TO00
TO0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
TOE0	0	0	0	0	0	0	0	0	TOE07	TOE06	TOE05	TOE04	TOE03	TOE02	TOE01	TOE00
									0	0	1	0	1	1	1	1

Data to be written	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

									TO07	TO06	TO05	TO04	TO03	TO02	TO01	TO00
After writing TO0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0

Writing is done only to the TOmn bit with TOEmn = 0, and writing to the TOmn bit with TOEmn = 1 is ignored. TOmn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to the TOmn bit, it is ignored and the output change by timer operation is normally done.

Figure 6 - 42 TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.6.5 Timer Interrupt and TOmn Pin Output at Operation Start

In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

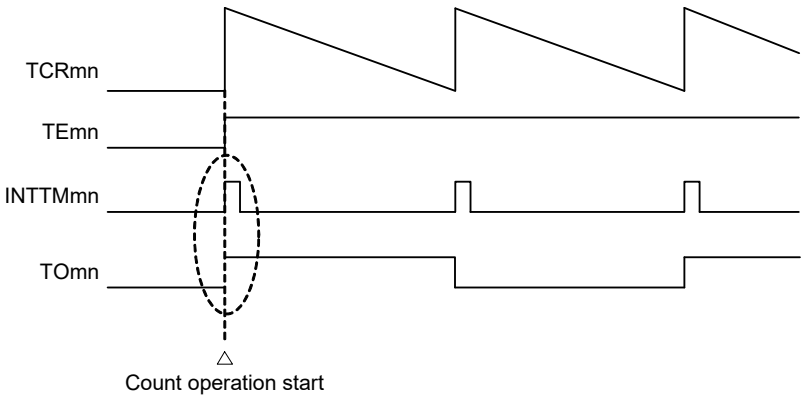
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled.

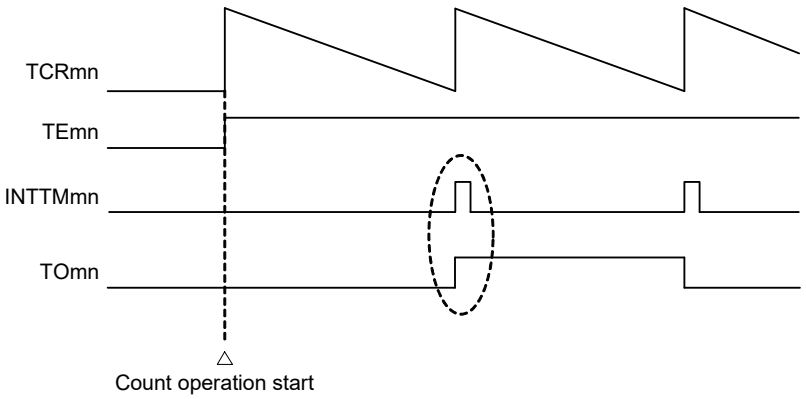
Figure 6 - 43 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 6 - 43 Operation examples of timer interrupt at count operation start and TOmn output

(a) When MDmn0 is set to 1



(b) When MDmn0 is set to 0



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOmn does not change either. After counting one cycle, INTTMmn is output and TOmn performs a toggle operation.

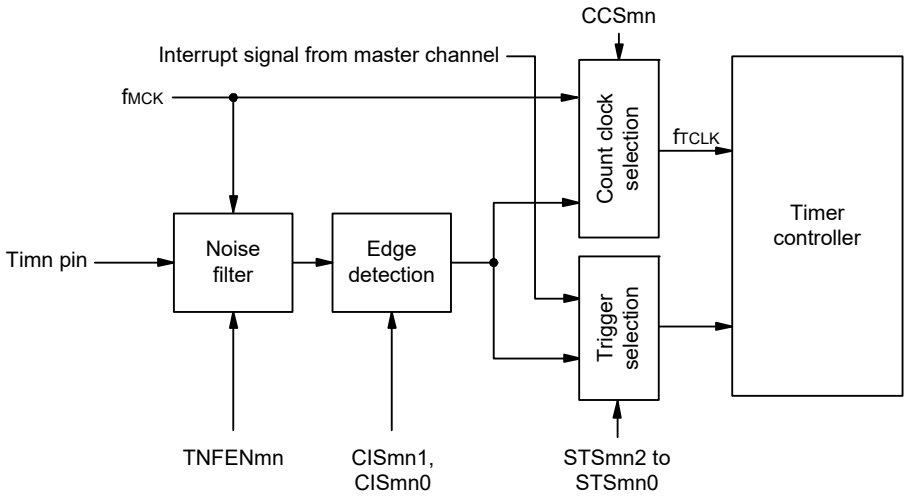
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.7 Timer Input (TImn) Control

6.7.1 TImn 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. The following shows the configuration of the input circuit.

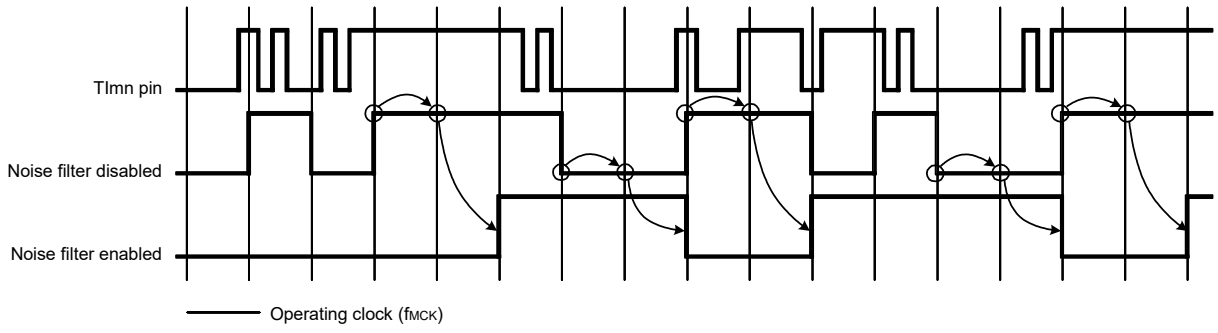
Figure 6 - 44 Input Circuit Configuration



6.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (fMCK) for channel n. When the noise filter is enabled, after synchronization with the operating clock (fMCK) for channel n, whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.

Figure 6 - 45 Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



Caution The input waveforms to the TImn pin are shown to explain the operation when the noise filter is enabled or disabled.

6.7.3 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 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) 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 (TSm).

(2) Noise filter is enabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) 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 (TSm).

6.8 Independent Channel Operation Function of Timer Array Unit

6.8.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

(2) Operation as square wave output

TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

- Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
- Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

Timer count register mn (TCRmn) operates as a down counter in the interval timer mode.

The TCRmn register loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1. If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and TOmn is toggled.

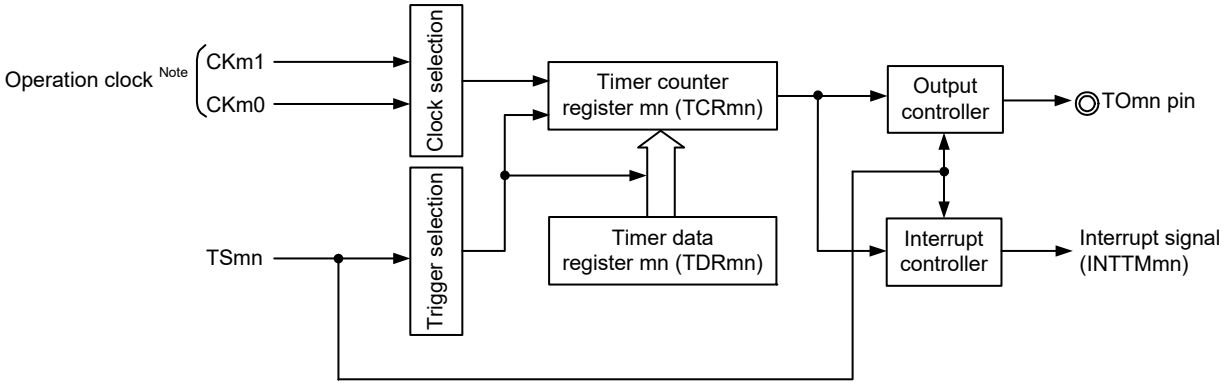
After that, the TCRmn register count down in synchronization with the count clock.

When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

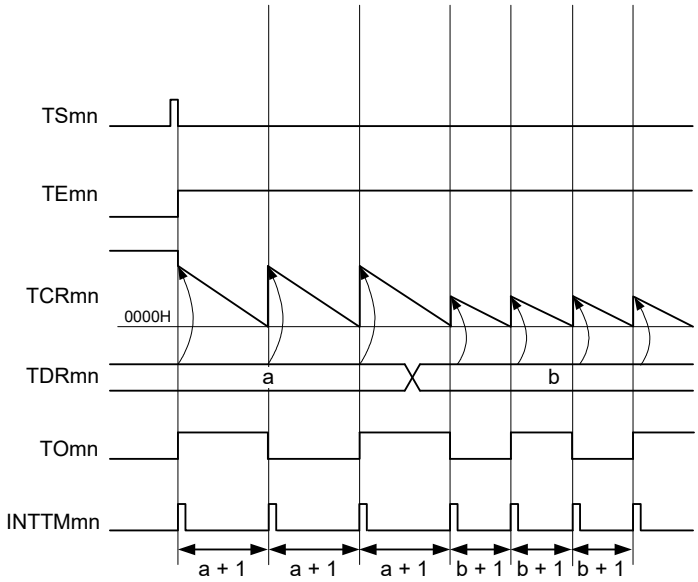
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 46 Block Diagram of Operation as Interval Timer/Square Wave Output



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6 - 47 Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)

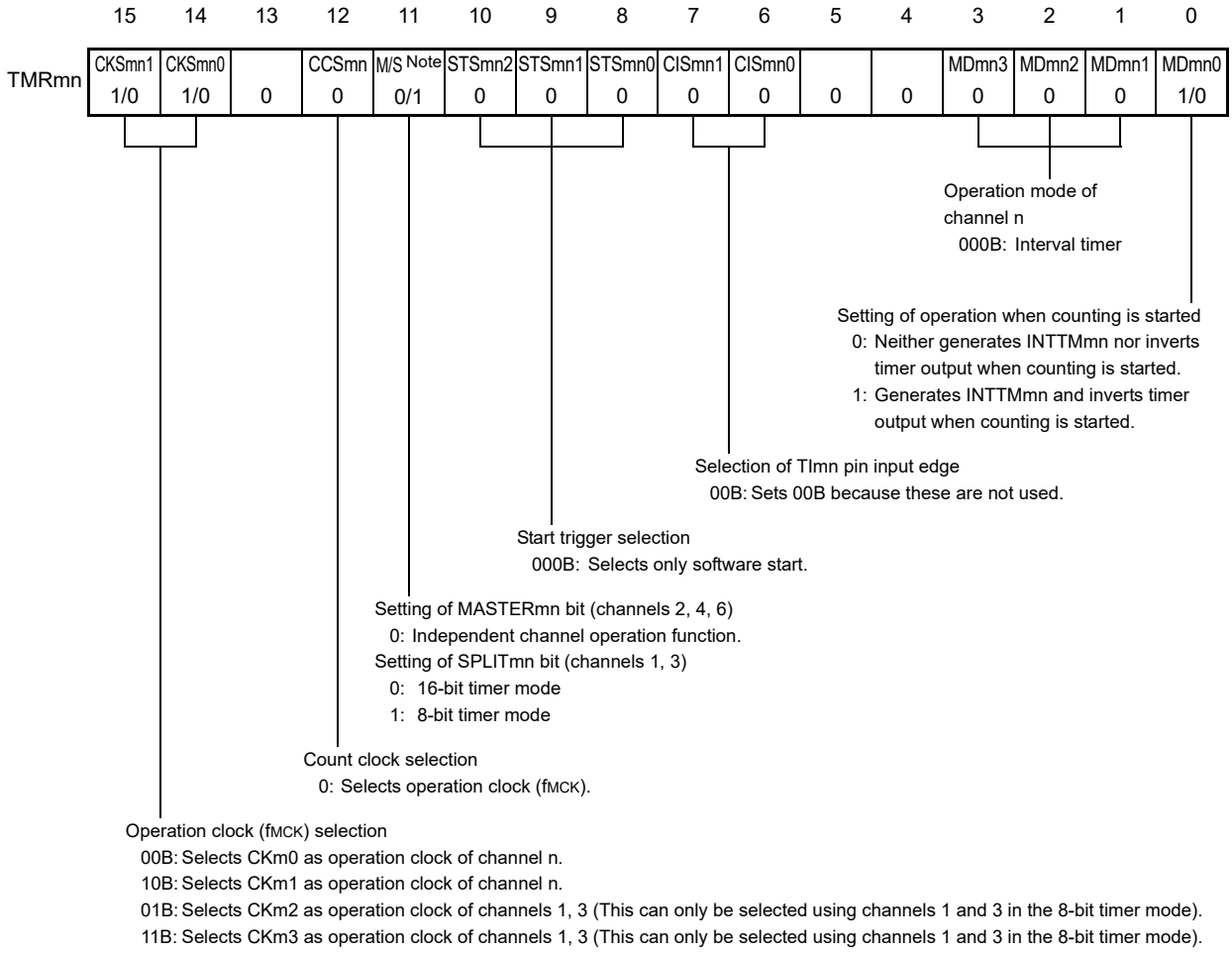


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

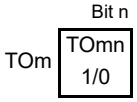
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
TOmn: TOmn pin output signal

Figure 6 - 48 Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output

(a) Timer mode register mn (TMRmn)

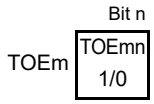


(b) Timer output register m (TOm)



0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

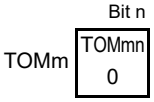


0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)



0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 49 Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOmn output Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. →	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOEmn bit to 1 and enables operation of TOmn. →	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmn pin outputs the TOmn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.). Sets the TSmn (TSHm1, TSHm3) bit to 1. → The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn). INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed. Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn (TTHm1, TTHm3) bit is set to 1. → The TTmn (TTHm1, TTHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3), and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized but holds current status.
	The TOEmn bit is cleared to 0 and value is set to the TOmn bit. →	The TOmn pin outputs the TOmn bit set level.

Operation is resumed.

(**Remark** is listed on the next page.)

Figure 6 - 50 Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmn pin output level Clears the TOmn bit to 0 after the value to be held is set to the port register. →	The TOmn pin output level is held by port function.
	When holding the TOmn pin output level is not necessary Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

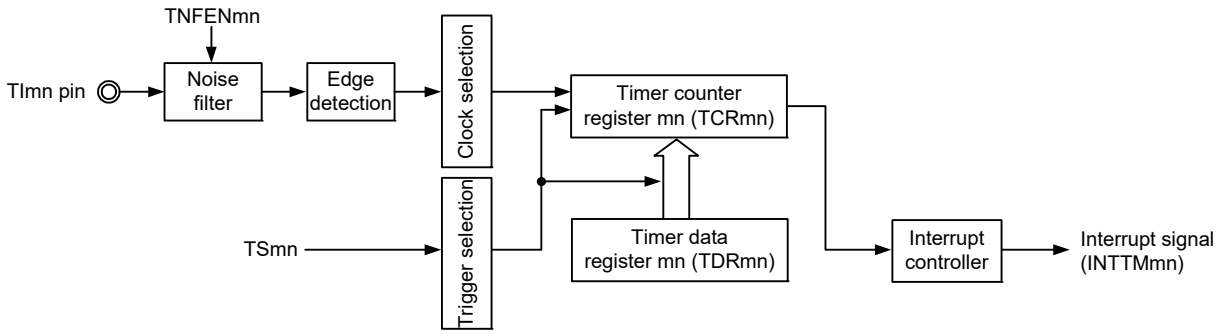
6.8.2 Operation as external event counter

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 TImn 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 TDRmn + 1

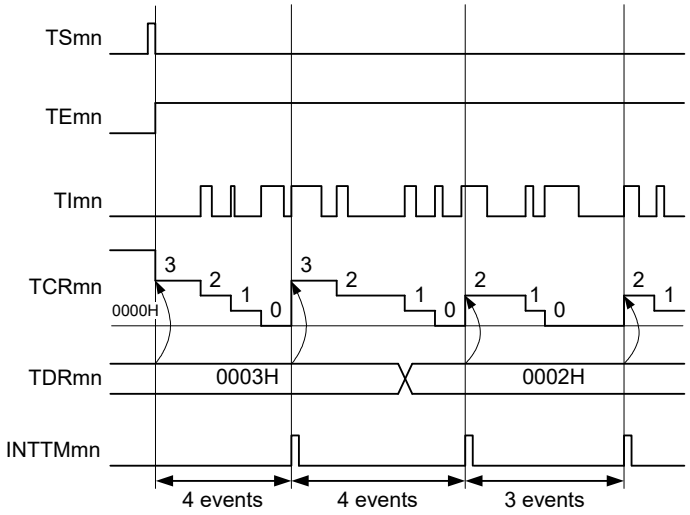
Timer count register mn (TCRmn) operates as a down counter in the event counter mode.
The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) to 1.
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.
After that, the above operation is repeated.
An irregular waveform that depends on external events is output from the TOmn pin. Stop the output by setting the TOEmn bit of timer output enable register m (TOEm) to 0.
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 6 - 51 Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 52 Example of Basic Timing of Operation as External Event Counter

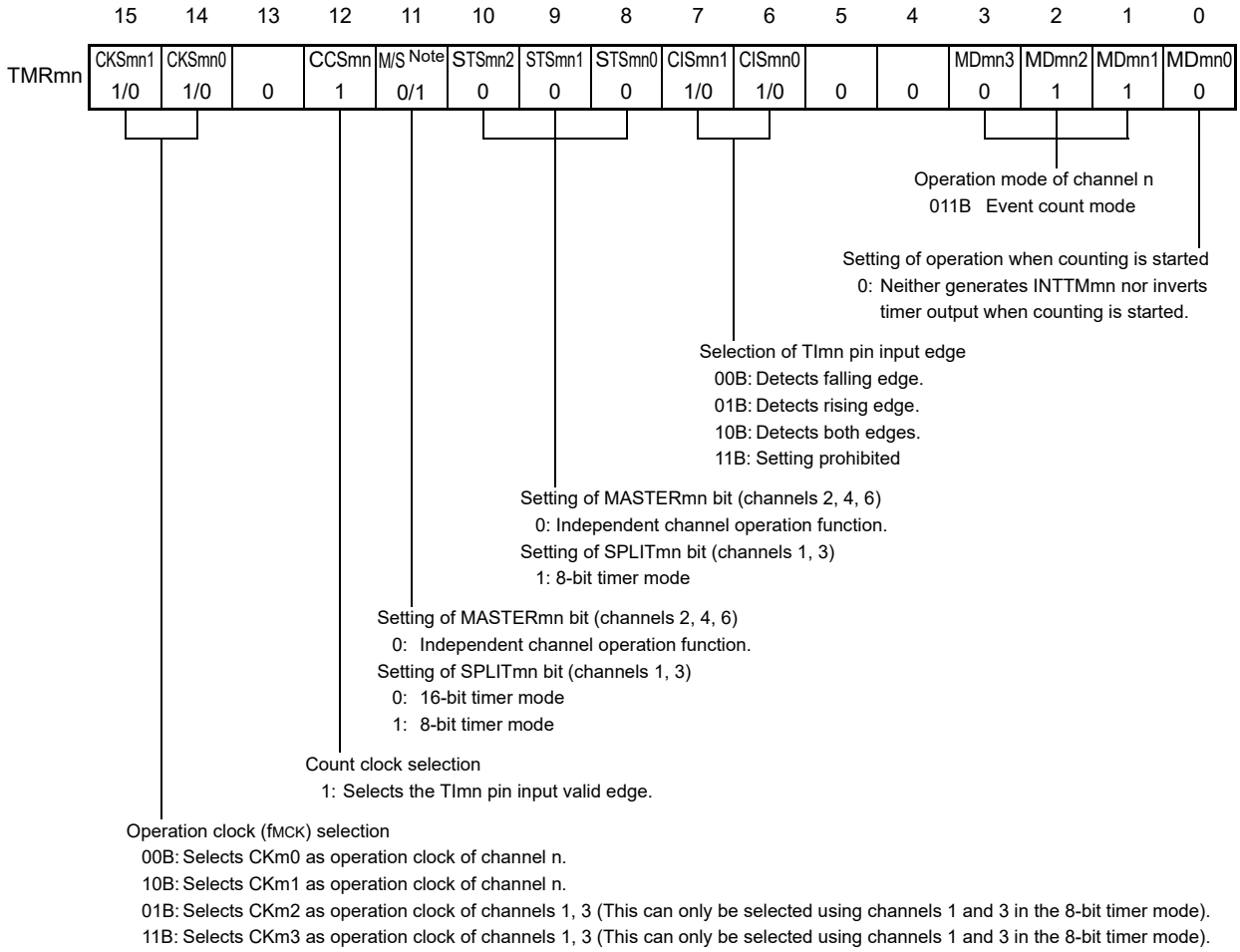


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

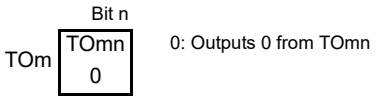
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 6 - 53 Example of Set Contents of Registers in External Event Counter Mode

(a) Timer mode register mn (TMRmn)

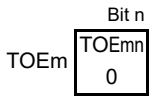


(b) Timer output register m (TOm)



0: Outputs 0 from TOmn

(c) Timer output enable register m (TOEm)

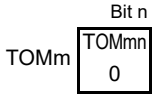


0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)



0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 54 Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets number of counts to timer data register mn (TDRmn). Clears the TOEmn bit of timer output enable register m (TOEm) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn) and detection of the TImn pin input edge is awaited.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.8.3 Operation as input pulse interval measurement

The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured. In addition, the count value can be captured by using software operation (TSmn = 1) as a capture trigger while the TEmn bit is set to 1.

The pulse interval can be calculated by the following expression.

TImn input pulse interval = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of up to one operating clock cycle occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TCRmn register counts up from 0000H in synchronization with the count clock.

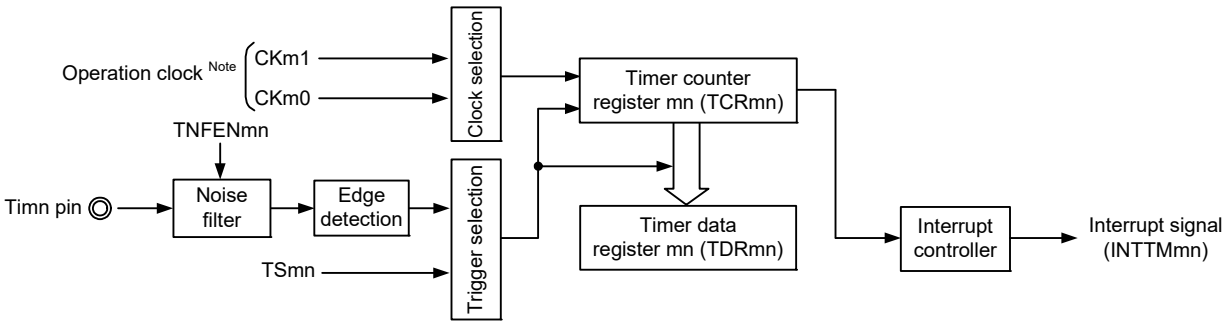
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) 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 TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status 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 TSRmn 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 STSmn2 to STSmn0 bits of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

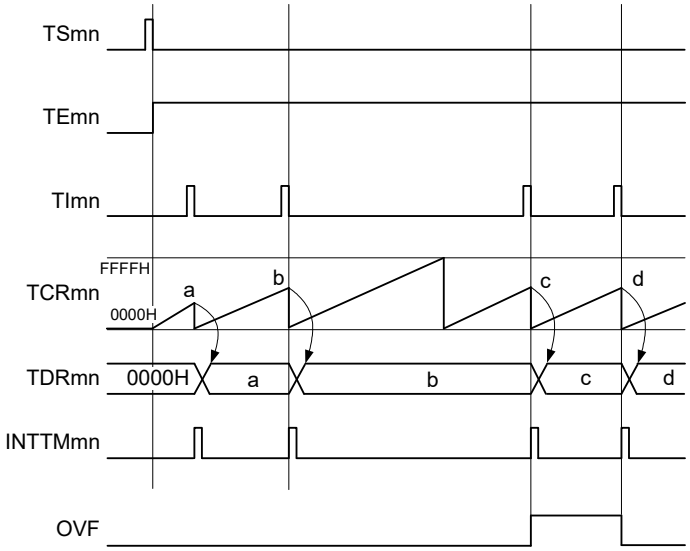
Figure 6 - 55 Block Diagram of Operation as Input Pulse Interval Measurement



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 56 Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)

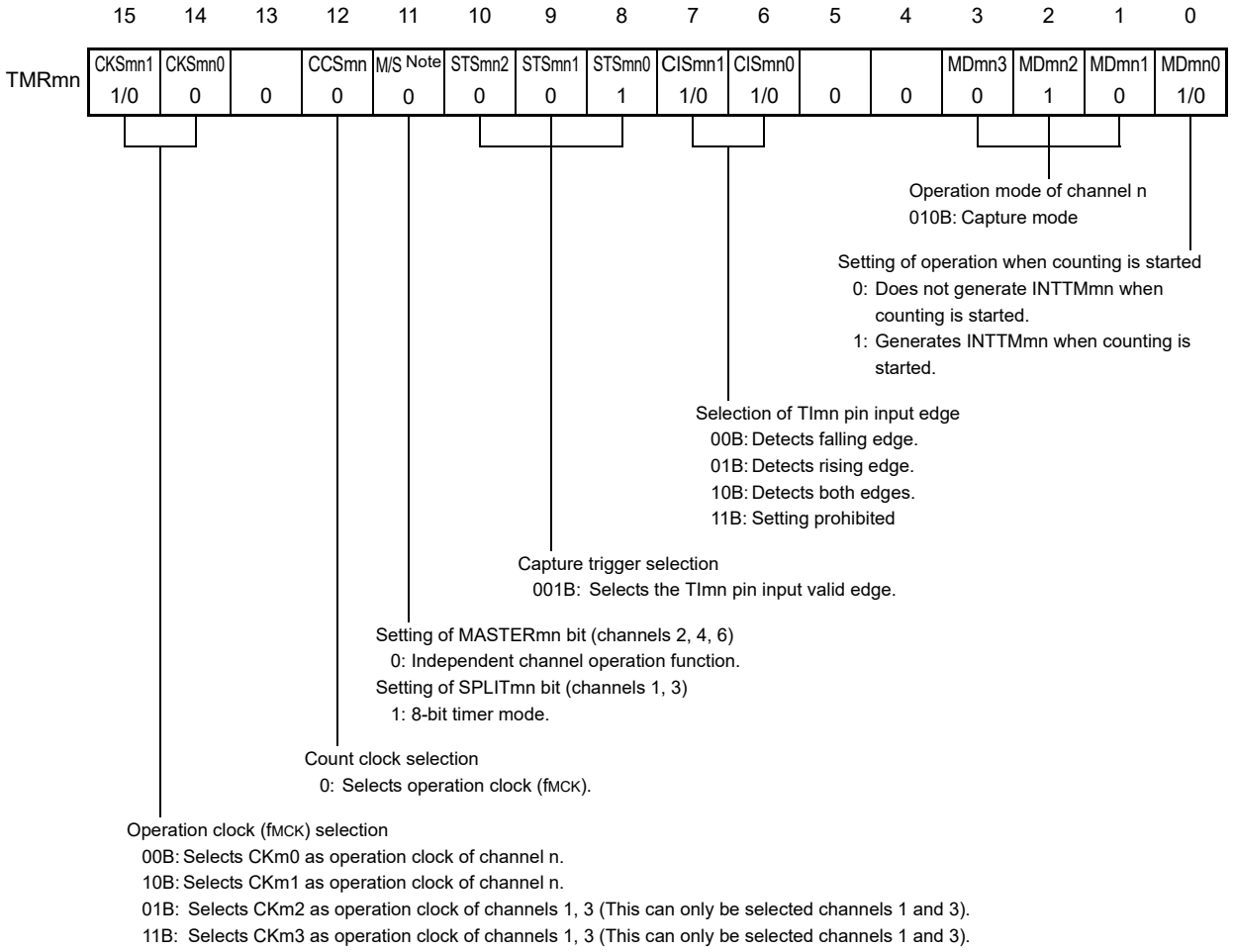


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

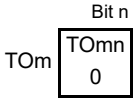
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 6 - 57 Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)

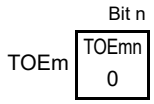


(b) Timer output register m (TOm)



0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

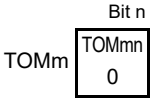


0: Stops TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)



0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 58 Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Timer count register mn (TCRmn) is cleared to 0000H at the count clock input. When the MDmn0 bit of the TMRmn register is 1, INTTMmn is generated.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts up from 0000H. When the valid edge of the TImn pin input is detected or the TSmn bit is set to 1, the count value is transferred (captured) to timer data register mn (TDRmn). At the same time, the TCRmn register is cleared to 0000H, and the INTTMmn signal is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.8.4 Operation as input signal high-/low-level width measurement

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

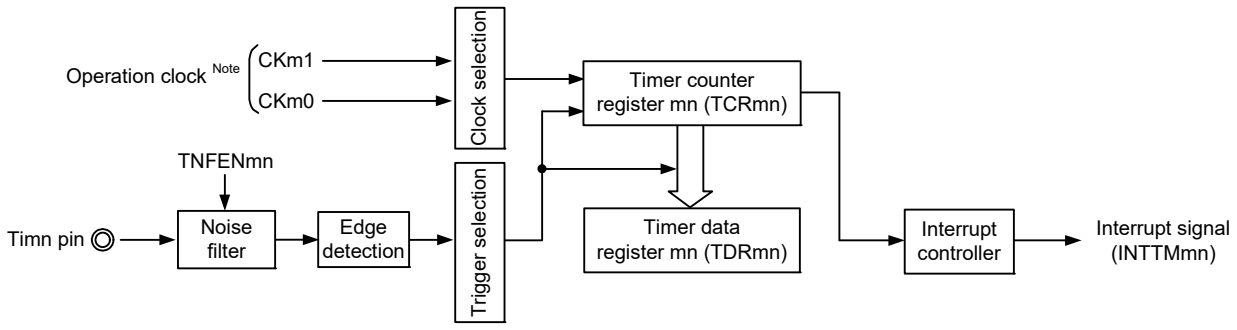
Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error equivalent to one operation clock occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture & one-count mode.
When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.
When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value “value transferred to the TDRmn register + 1”, and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status 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 TSRmn 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 TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.
Because this function is used to measure the signal width of the TImn pin input, the TSmn bit cannot be set to 1 while the TEmn bit is 1.

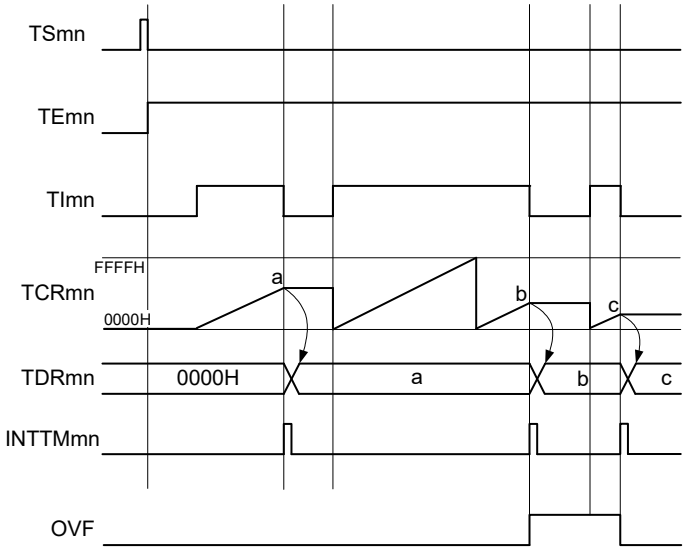
CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 6 - 59 Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Note For channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6 - 60 Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement

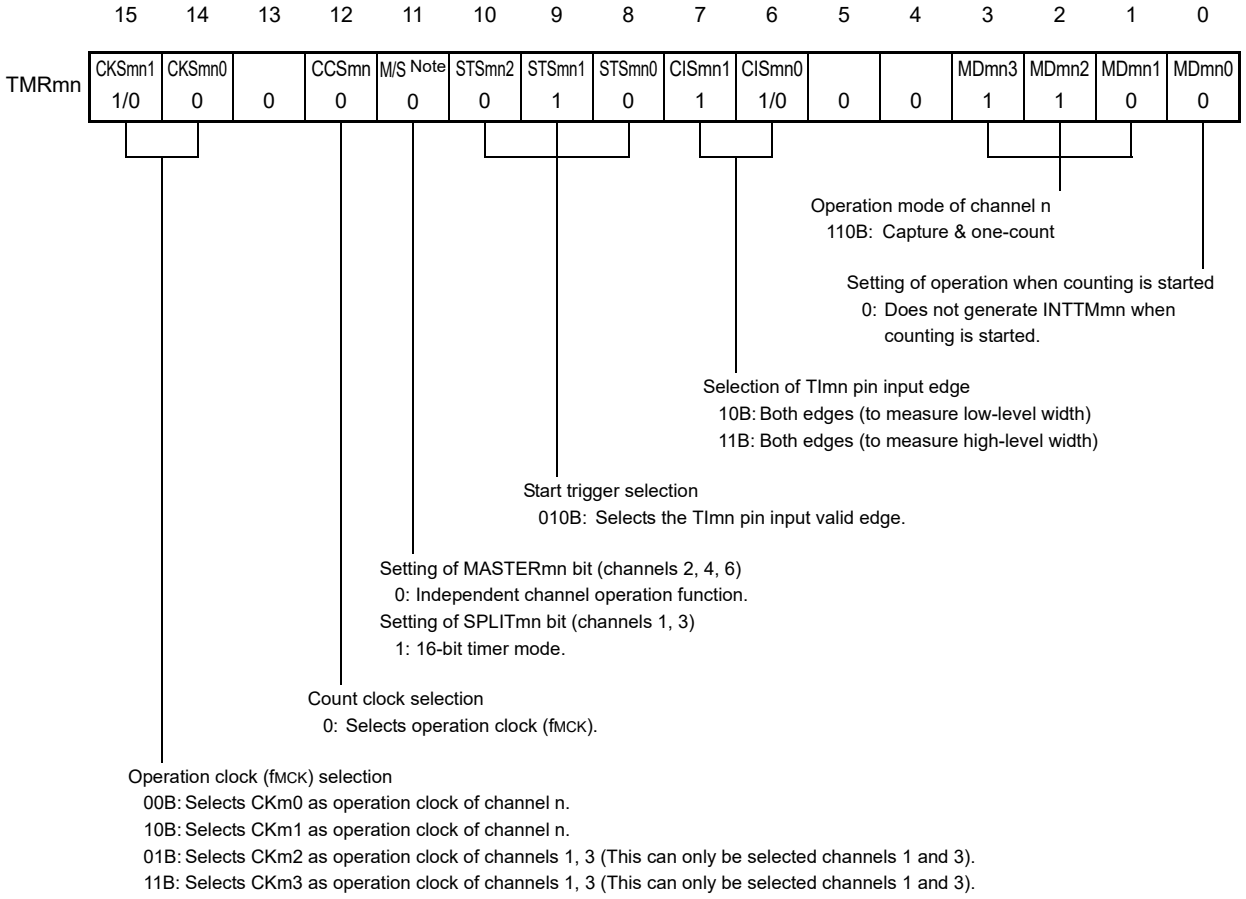


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

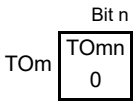
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 6 - 61 Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register mn (TMRmn)

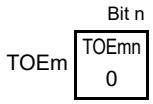


(b) Timer output register m (TOm)



0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

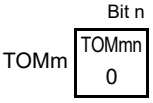


0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)



0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 62 Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects the TImn pin input count start valid edge. →	Clears timer count register mn (TCRmn) to 0000H and starts counting up.
During operation	Set value of the TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.8.5 Operation as delay counter

It is possible to start counting down when the valid edge of the TImn pin input is detected (an external event), and then generate INTTMmn (a timer interrupt) after any specified interval.

It is also possible to start counting down and generate INTTMmn (timer interrupt) at any interval by setting TSmn to 1 by software while TEmn = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

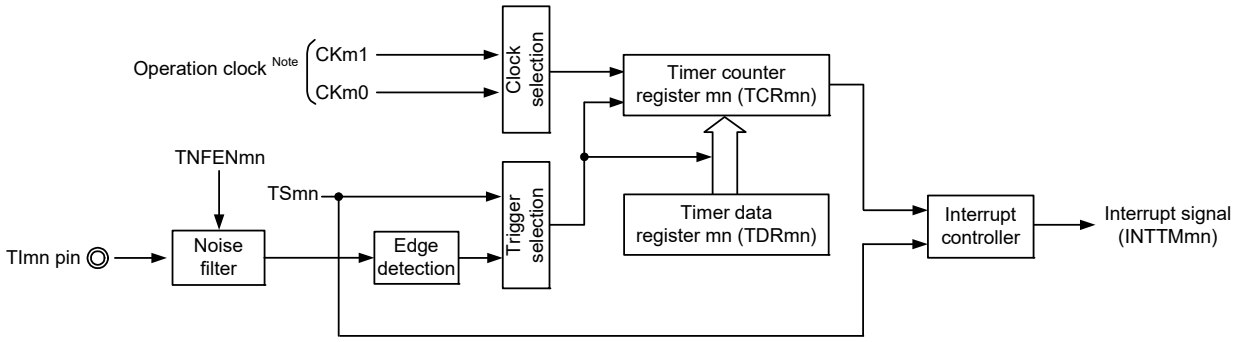
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

When the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1, the TEmn, TEHm1, TEHm3 bits are set to 1 and the TImn pin input valid edge detection wait status is set.

Timer count register mn (TCRmn) starts operating upon TImn pin input valid edge detection and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next TImn pin input valid edge is detected.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

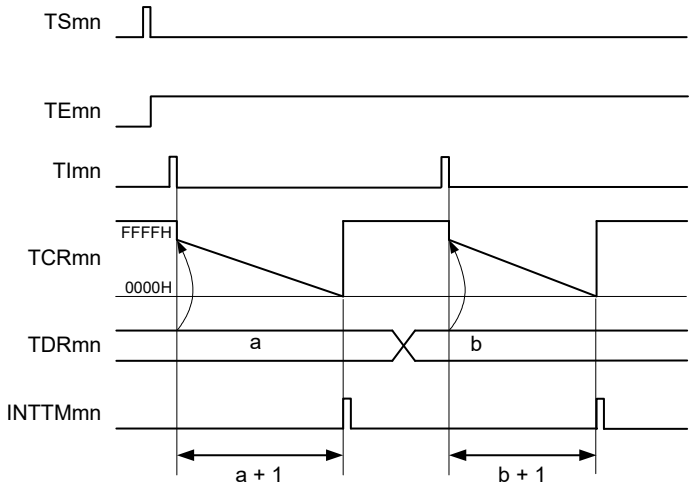
Figure 6 - 63 Block Diagram of Operation as Delay Counter



Note For using channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 64 Example of Basic Timing of Operation as Delay Counter

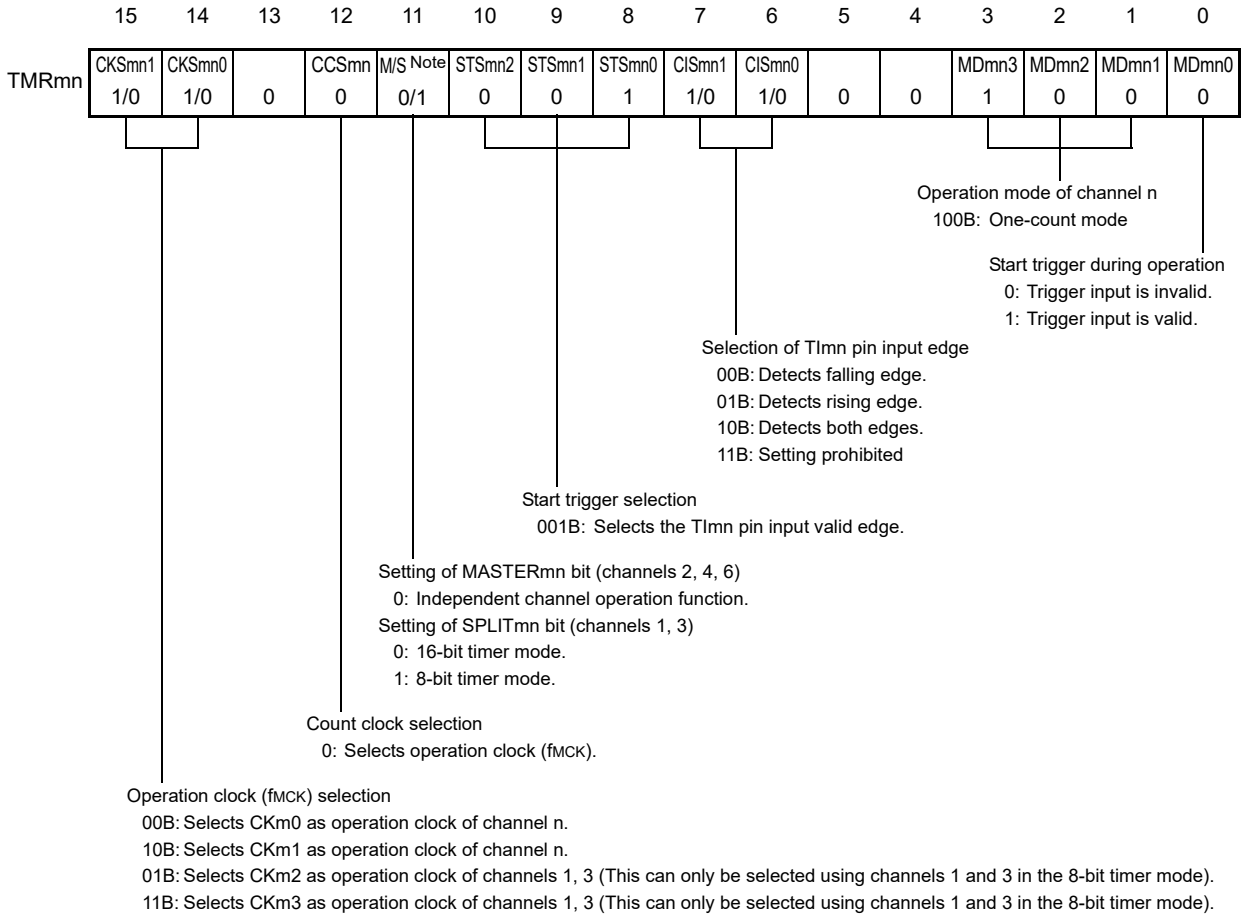


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

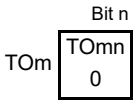
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 6 - 65 Example of Set Contents of Registers to Delay Counter

(a) Timer mode register mn (TMRmn)

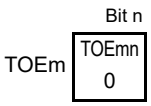


(b) Timer output register m (TOm)



0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

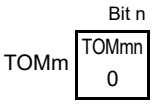


0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)



0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6 - 66 Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). INTTMmn output delay is set to timer data register mn (TDRmn). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1) wait status is set.
	Detects the TImn pin input valid edge. →	Value of the TDRmn register is loaded to the timer count register mn (TCRmn).
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used.	The counter (TCRmn) counts down. When the count value of TCRmn reaches 0000H, the INTTMmn output is generated, and the count operation stops until the next start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1).
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.9 Simultaneous Channel Operation Function of Timer Array Unit

6.9.1 Operation as 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 TImn pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDRmn (master) + 2} × Count clock period
Pulse width = {Set value of TDRmp (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer count register mn (TCRmn) of the master channel starts operating upon start trigger detection and loads the value of timer data register mn (TDRmn).

The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

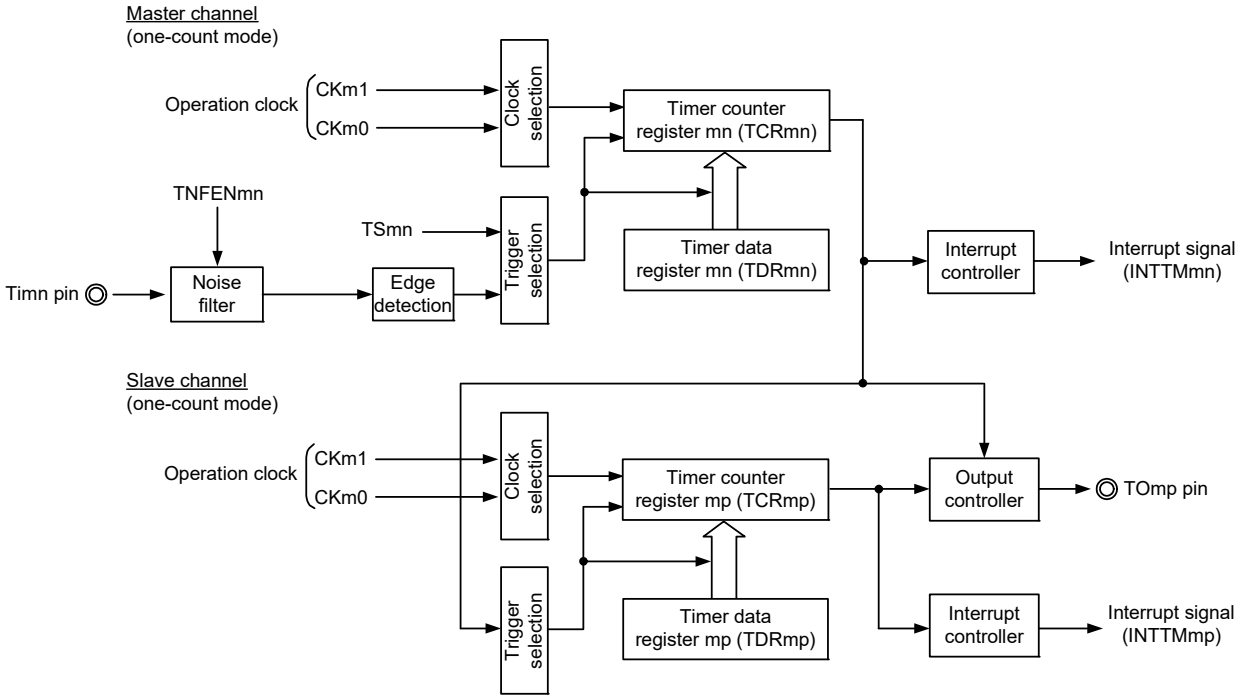
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution **The timing of loading of timer data register mn (TDRmn) of the master channel is different from that of the TDRmp register of the slave channel. If the TDRmn and TDRmp registers are rewritten during counting, therefore, an illegal waveform may be output in conflict with the timing of loading. Rewrite the TDRmn register after INTTMmn is generated and the TDRmp register after INTTMmp is generated.**

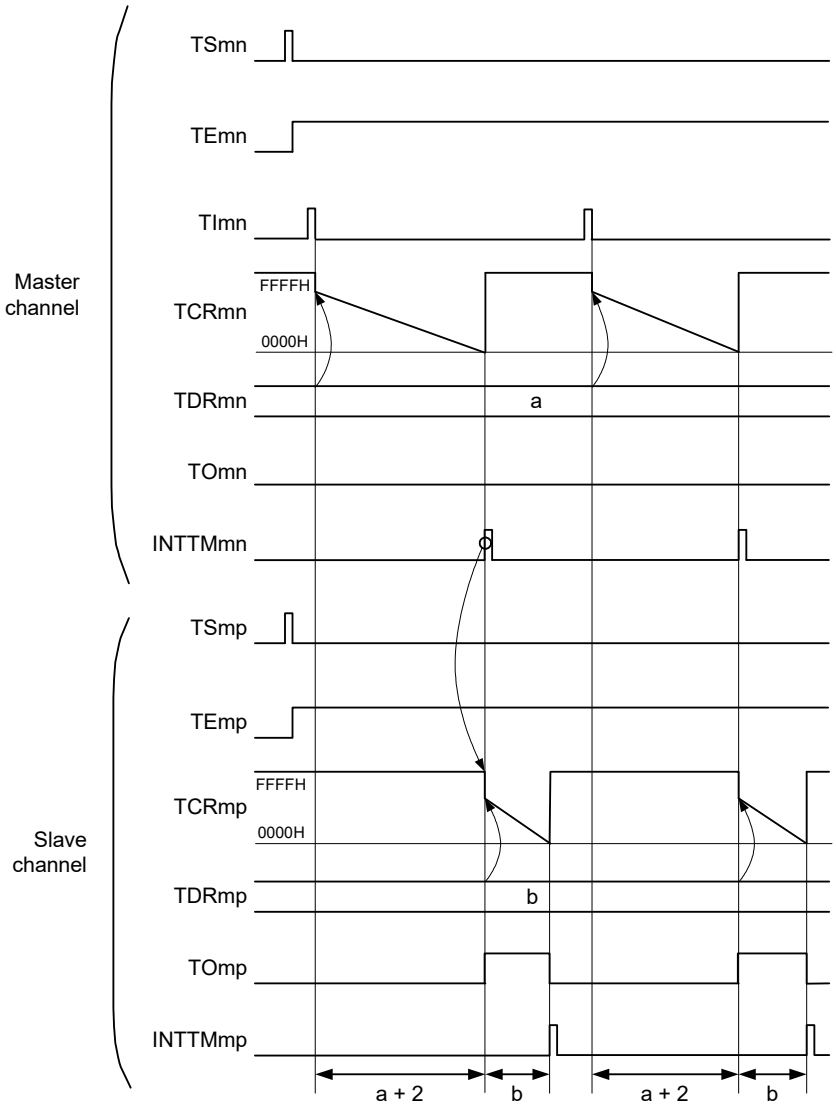
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6 - 67 Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6))
p: Slave channel number (n < p ≤ 7)

Figure 6 - 68 Example of Basic Timing of Operation as One-Shot Pulse Output Function

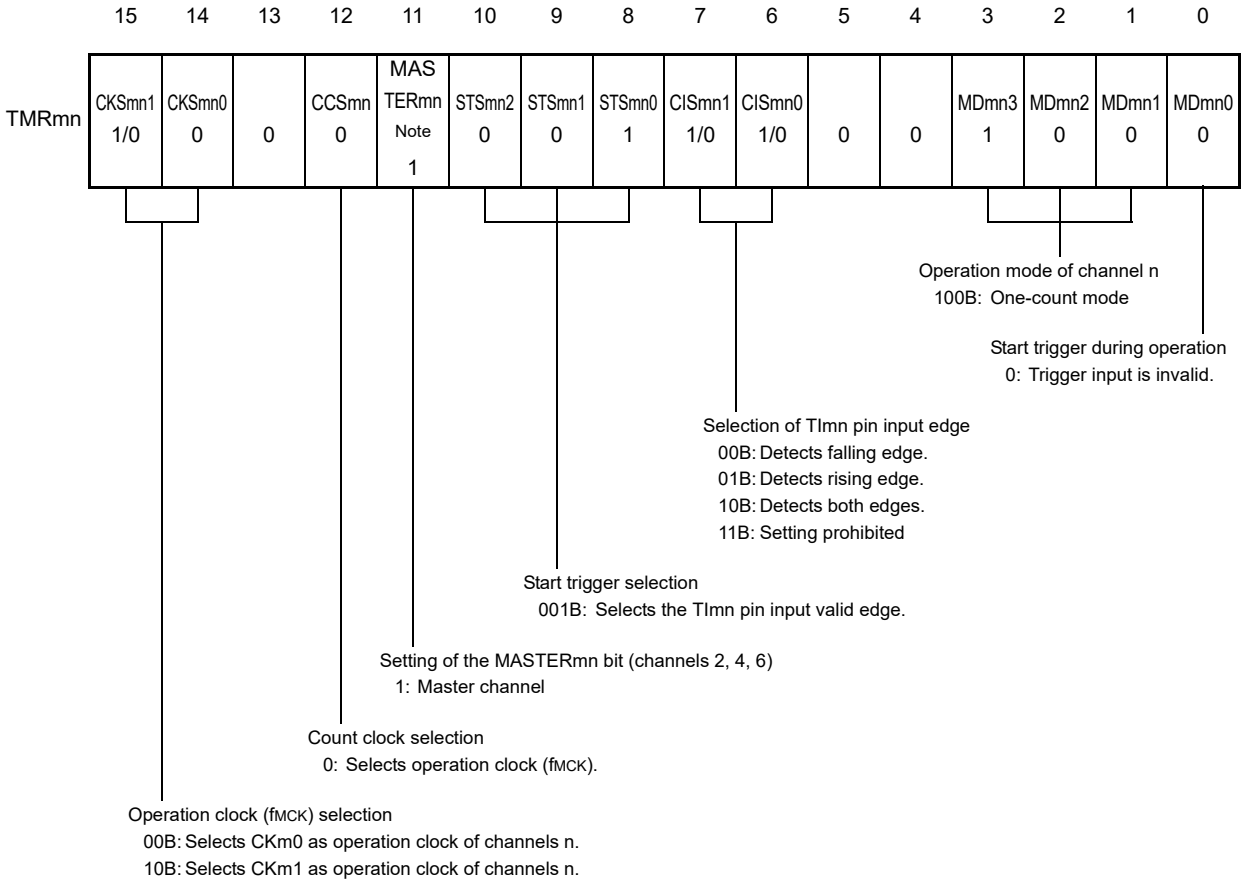


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Remark 2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TImn, TImp: TImn and TImp pins input signal
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 6 - 69 Example of Set Contents of Registers
When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

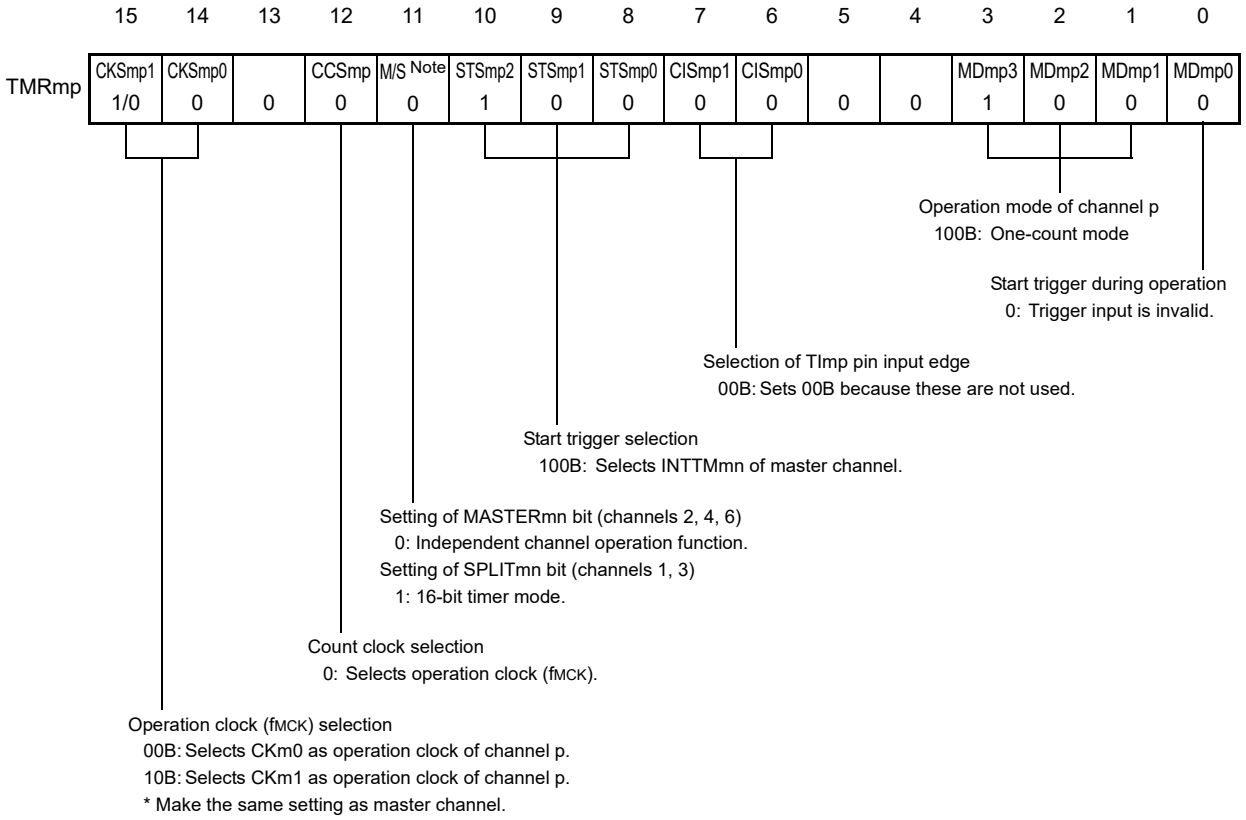
	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

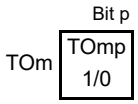
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)

Figure 6 - 70 Example of Set Contents of Registers
When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)

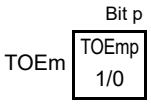


(b) Timer output register m (TOm)



0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

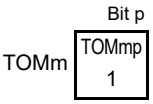


0: Stops the TOmp output operation by counting operation.
1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)



1: Sets the slave channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6 - 71 Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable registers 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1 (NFEN1) to 1. Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register mn (TDRmn) of the master channel, and a pulse width is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

(**Remark** is listed on the next page.)

Figure 6 - 72 Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). →	The TEmn and TEmp bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1) wait status. Counter stops operating.
	The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	
	Count operation of the master channel is started by start trigger detection of the master channel. • Detects the TImn pin input valid edge. • Sets the TSmn bit of the master channel to 1 by software Note. Note Do not set the TSmn bit of the slave channel to 1. →	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers by slave channel can be changed.	Master channel loads the value of the TDRmn register to timer count register mn (TCRmn) by the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next start trigger detection. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. →	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. →	The TOmp pin output level is held by port function.
	When holding the TOmp pin output level is not necessary Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

6.9.2 Operation as 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 TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

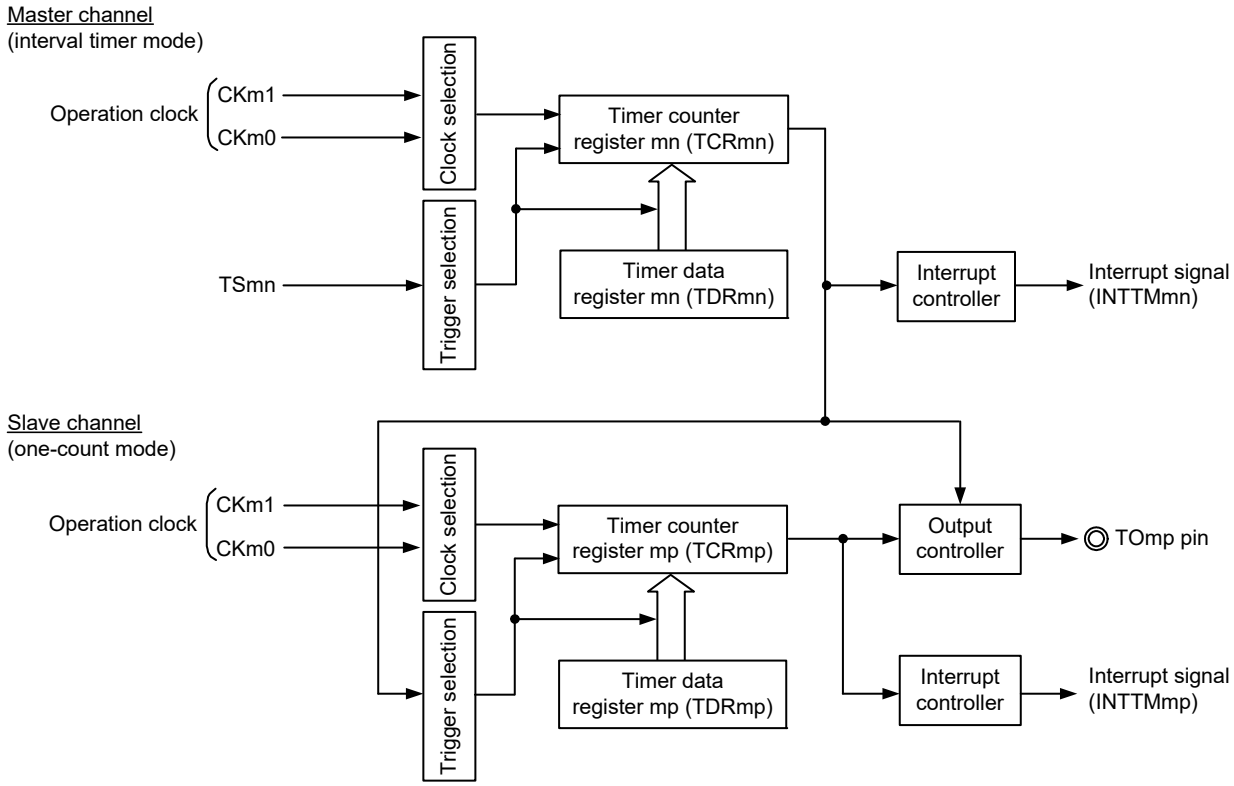
Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer count register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTm) is set to 1.
If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.
The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn 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 0000H is the PWM output (TOmp) duty.
PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

Caution **To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.**

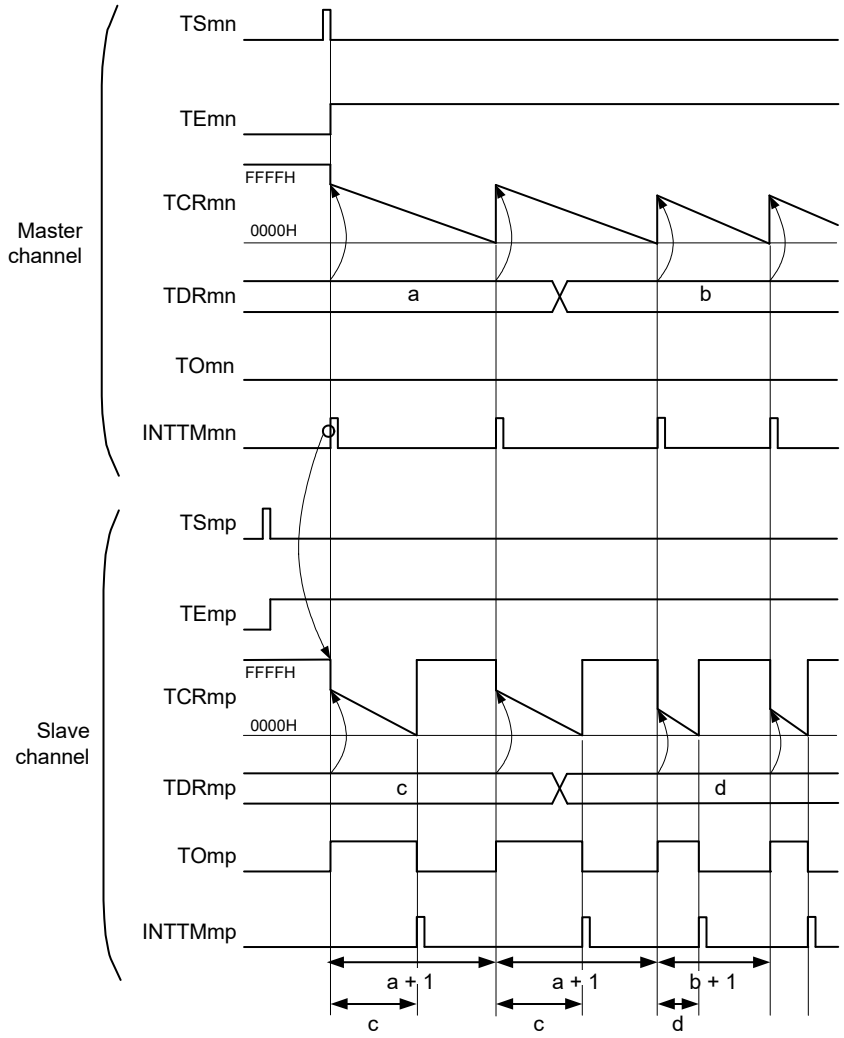
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6 - 73 Block Diagram of Operation as PWM Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6 - 74 Example of Basic Timing of Operation as PWM Function

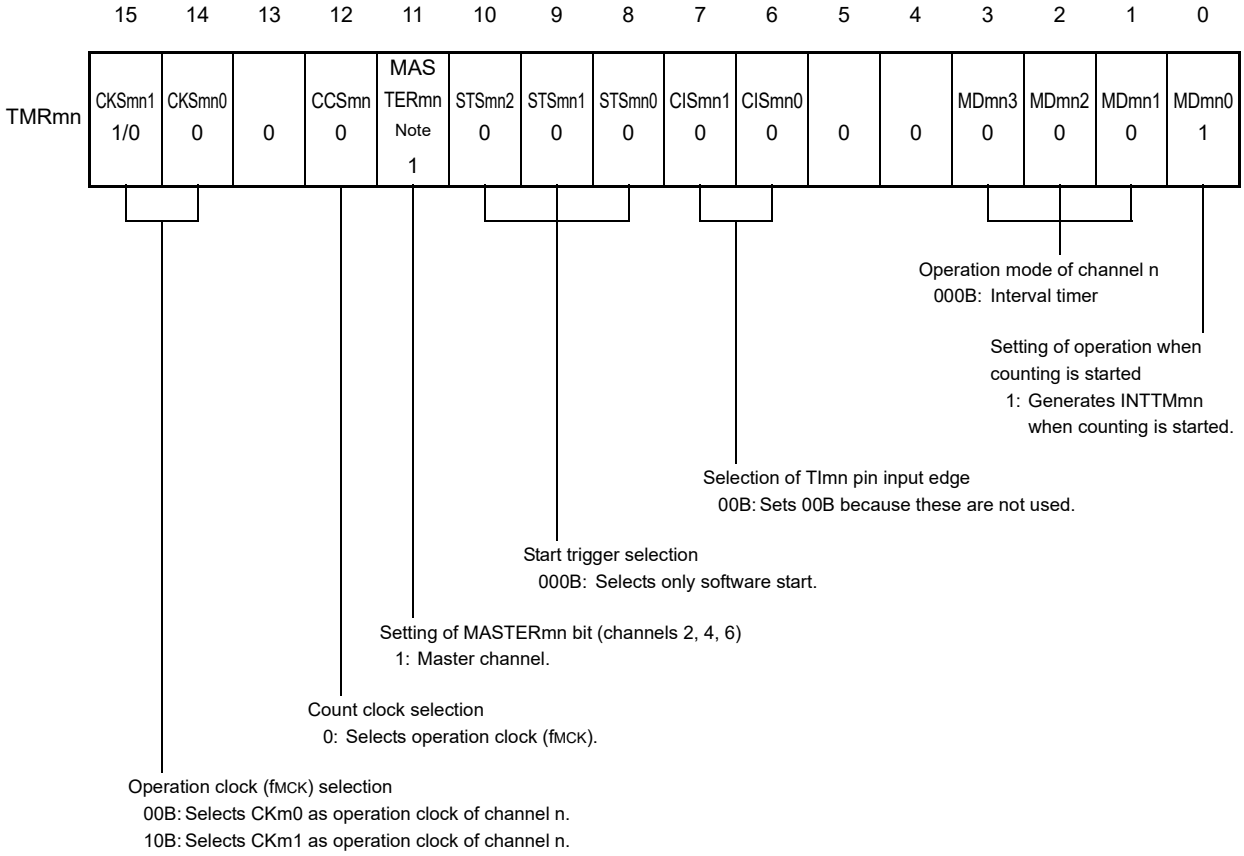


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

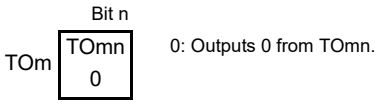
Remark 2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 6 - 75 Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

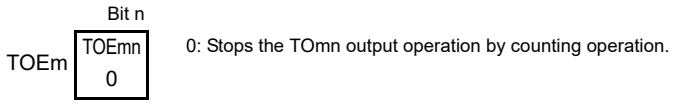
(a) Timer mode register mn (TMRmn)



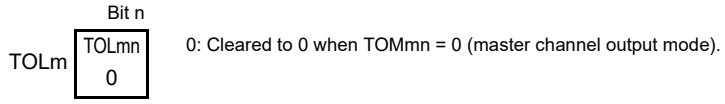
(b) Timer output register m (TOm)



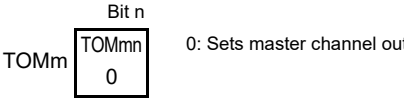
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)

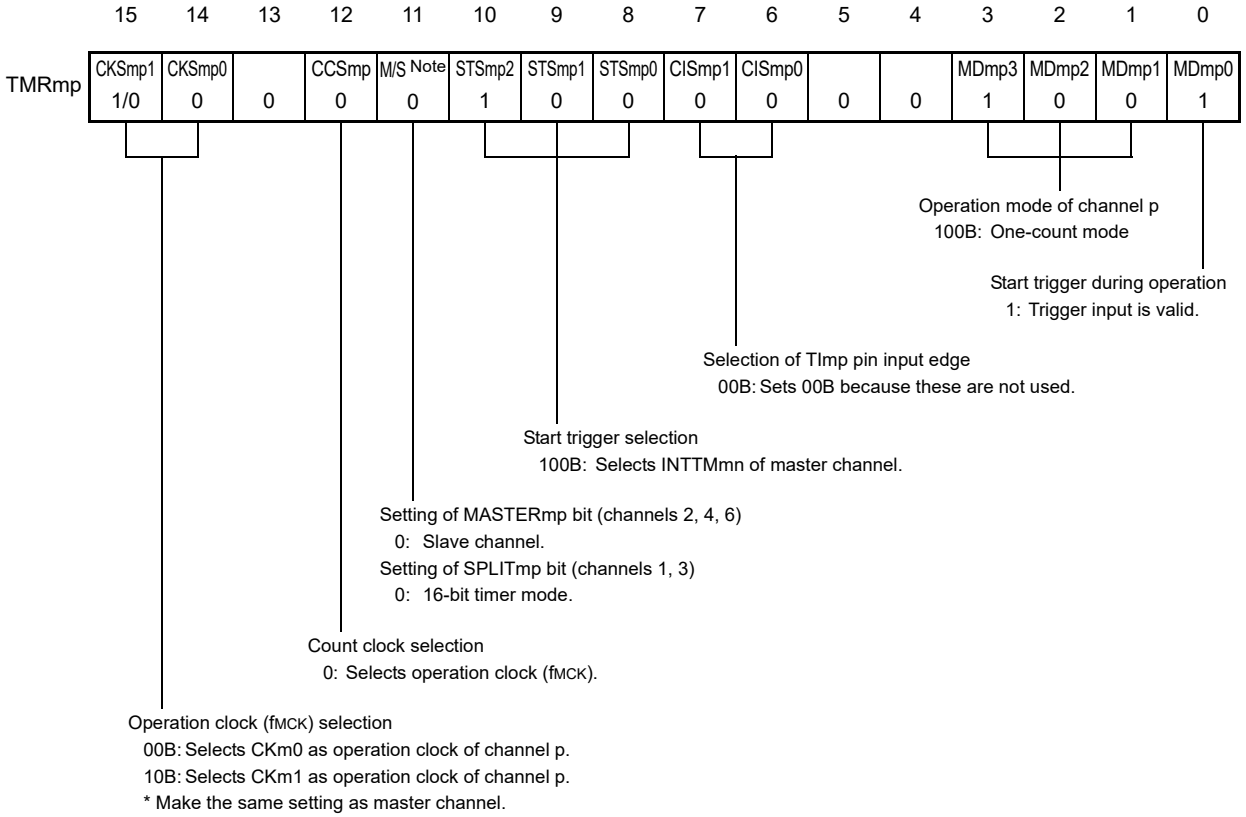


Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

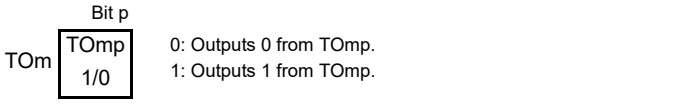
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)

Figure 6 - 76 Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

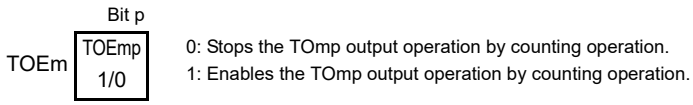
(a) Timer mode register mp (TMRmp)



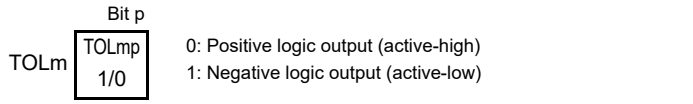
(b) Timer output register m (TOm)



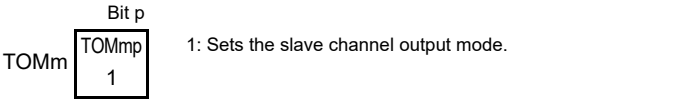
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2, TMRm4 TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6 - 77 Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

(**Remark** is listed on the next page.)

Figure 6 - 78 Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp = 1 → When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. →	The TOmp pin output level is held by port function.
	When holding the TOmp pin output level is not necessary Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

6.9.3 Operation as 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 TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer count register mn (TCRmn) of the master channel operates in the interval timer mode and counts the periods.

The TCRmp register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

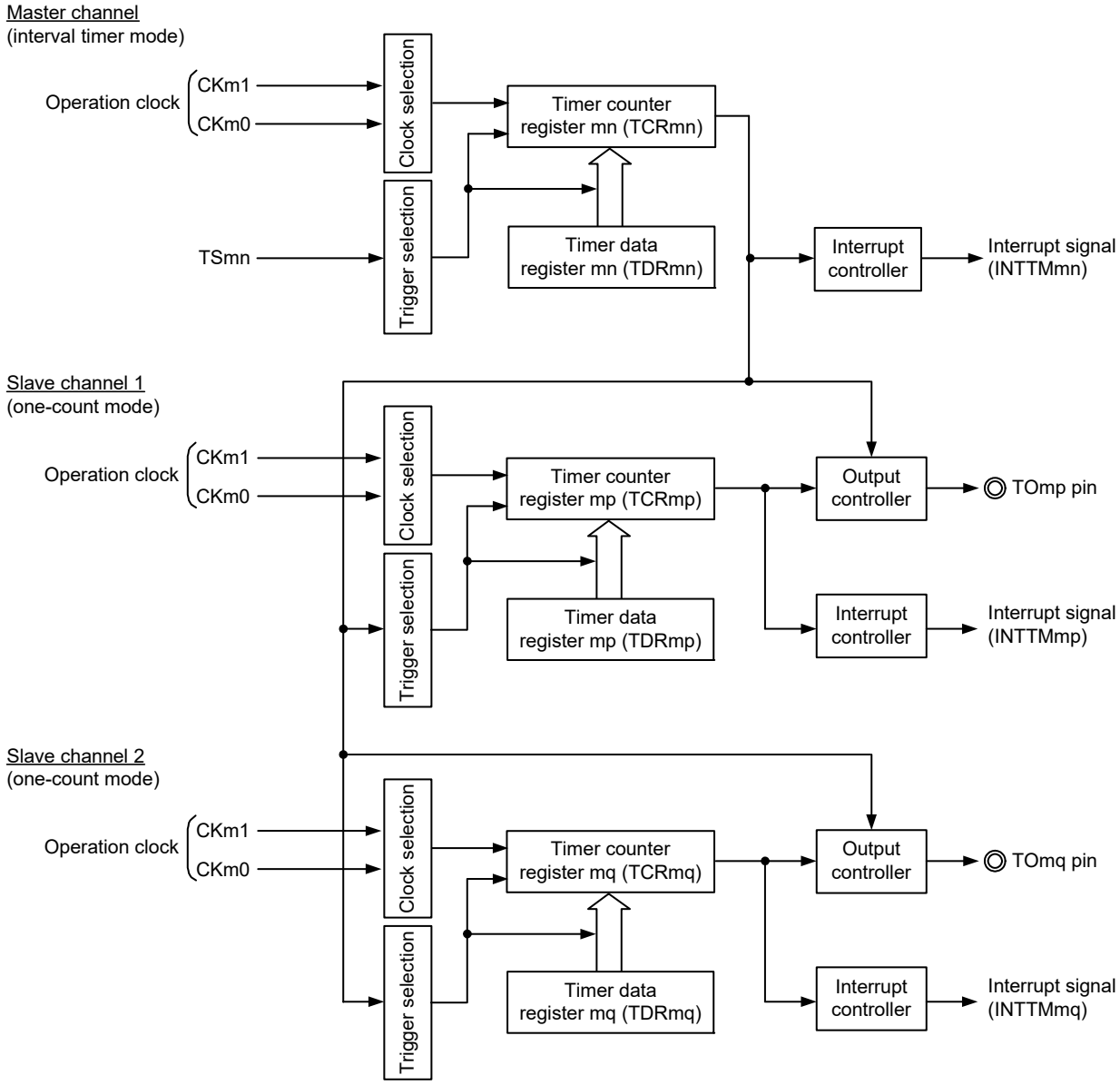
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.

When channel 0 is used as the master channel as above, up to three types of PWM signals can be output at the same time.

Caution **To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).**

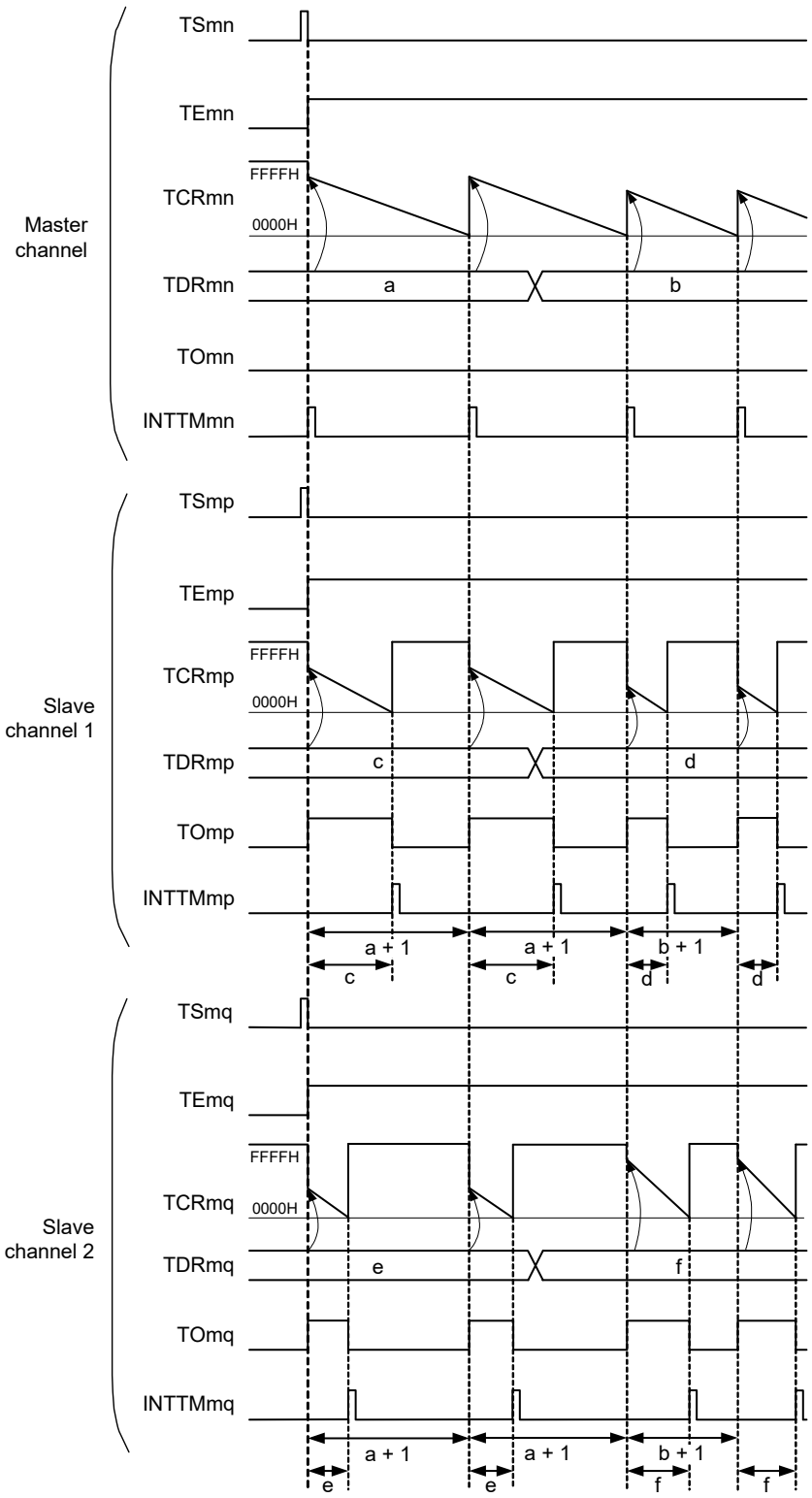
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6 - 79 Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6 - 80 Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs) (1/2)



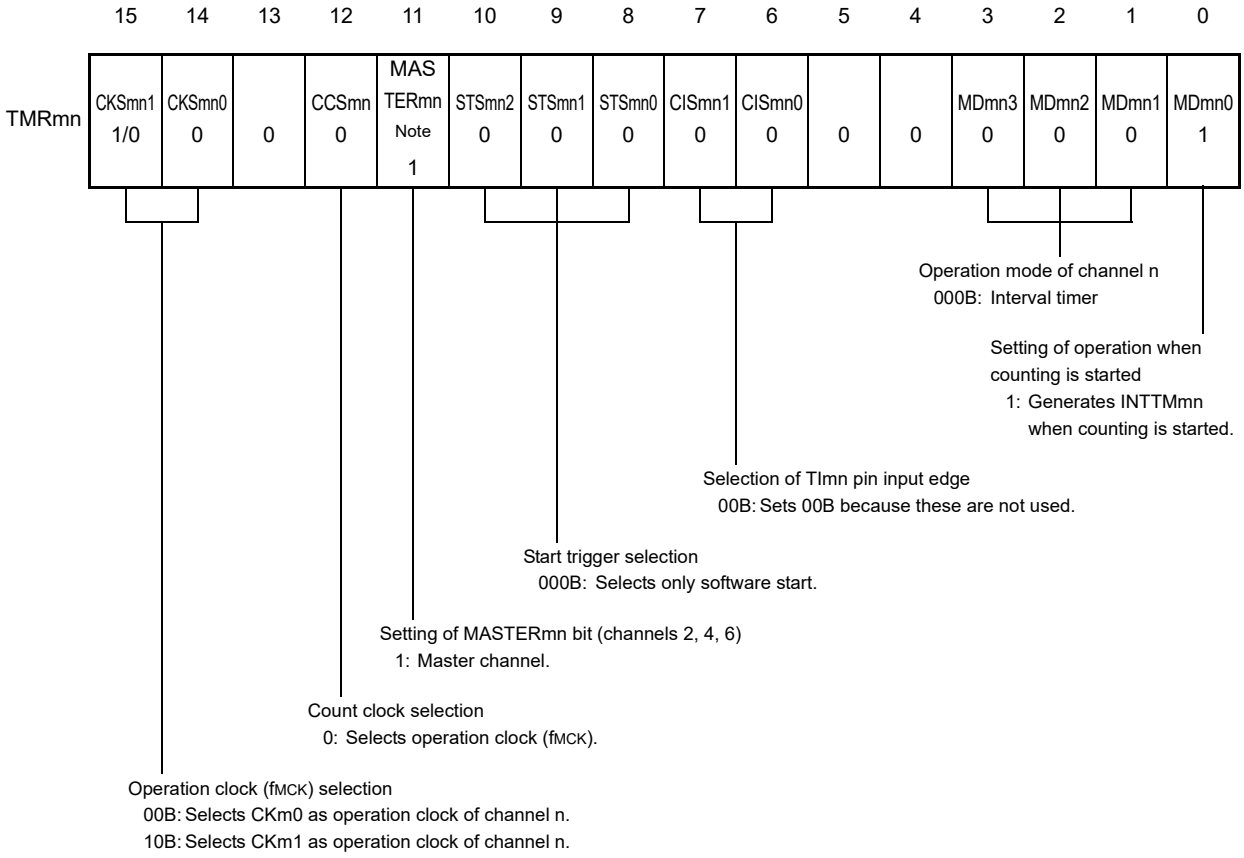
(**Remark** is listed on the next page.)

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Remark 2. TSmn, TSmp, TSmq: Bit n, p, q of timer channel start register m (TSm)
TEmn, TEmp, TEmq: Bit n, p, q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq: Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq: TOmn, TOmp, and TOmq pins output signal

Figure 6 - 81 Example of Set Contents of Registers
When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

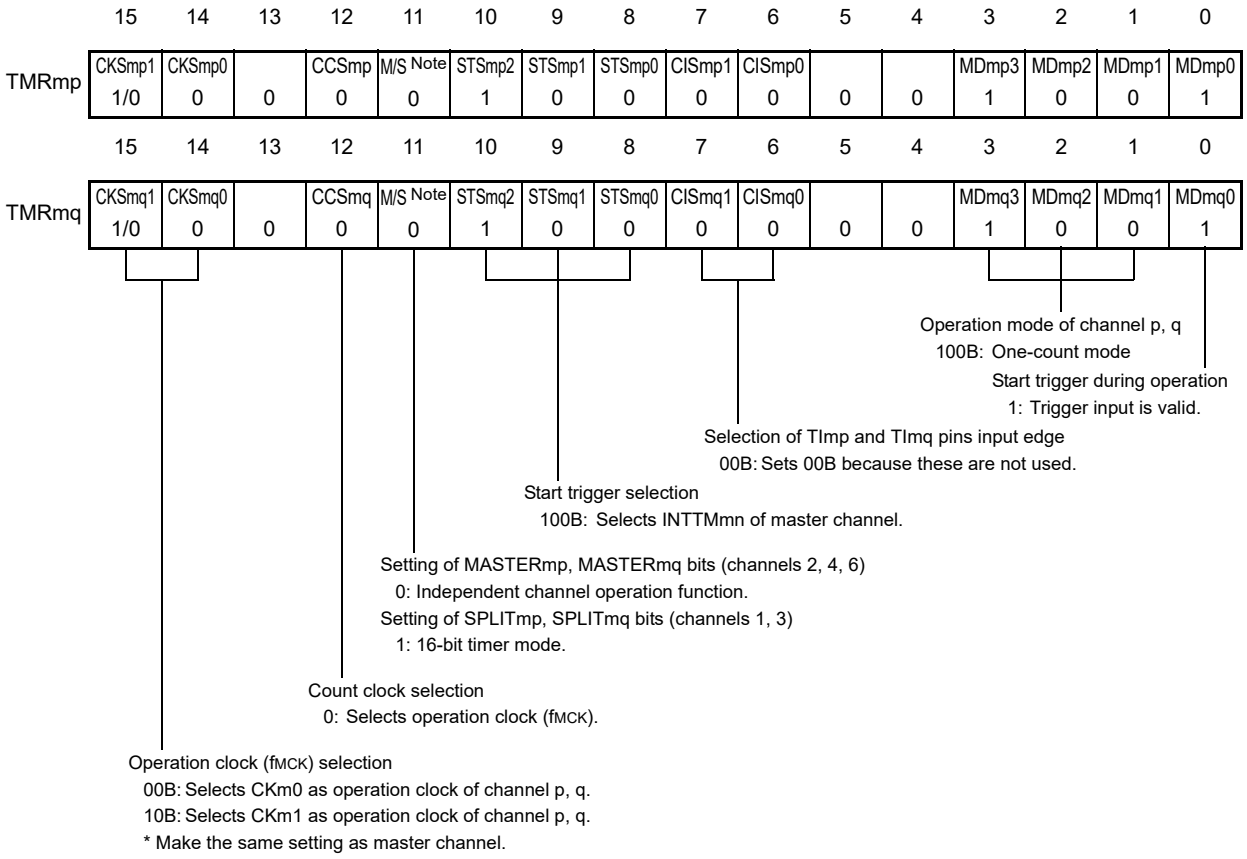
	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)

Figure 6 - 82 Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register mp, mq (TMRmp, TMRmq)



(b) Timer output register m (TOm)

Bit q Bit p

TOm	TOmq	TOmp
	1/0	1/0

0: Outputs 0 from TOmp or TOmq.
1: Outputs 1 from TOmp or TOmq.

(c) Timer output enable register m (TOEm)

Bit q Bit p

TOEm	TOEmq	TOEmp
	1/0	1/0

0: Stops the TOmp or TOmq output operation by counting operation.
1: Enables the TOmp or TOmq output operation by counting operation.

(d) Timer output level register m (TOLm)

Bit q Bit p

TOLm	TOLmq	TOLmp
	1/0	1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

Bit q Bit p

TOMm	TOMmq	TOMmp
	1	1

1: Sets the slave channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmp, MASTERmq bit
TMRm1, TMRm3: SPLITmp, SPLIT0q bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6 - 83 Operation Procedure When Multiple PWM Output Function Is Used (output two types of PWMs) (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp, mq (TMRmn, TMRmp, TMRmq) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Clears the TOLmp and TOLmq bits to 0. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmp and TOmq pins go into Hi-Z output state. The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. →	TOmp and TOmq do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.

(**Remark** is listed on the next page.)

Figure 6 - 84 Operation Procedure When Multiple PWM Output Function Is Used (output two types of PWMs) (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	(Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. →	TEmn = 1, TEmp, TEmq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
	The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	
During operation	Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSRmq registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel 1, the values of the TDRmp register are transferred to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDRmq register are transferred to TCRmq register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. →	TEmn, TEmp, TEmq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOmq output are not initialized but hold current status.
	The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	
	The TOEmp and TOEmq bits of slave channels are cleared to 0 and value is set to the TOmp and TOmq bits. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.
TAU stop	To hold the TOmp and TOmq pin output levels Clears the TOmp and TOmq bits to 0 after the value to be held is set to the port register. →	The TOmp and TOmq pin output levels are held by port function.
	When holding the TOmp and TOmq pin output levels are not necessary Setting not required	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integer greater than n)

6.10 Cautions When Using Timer Array Unit

6.10.1 Cautions When Using Timer output

Depends on products, a pin is assigned a timer output and other alternate functions. In this case, outputs of the other alternate functions must be set in initial status.

For details, see **4.5 Register Settings When Using Alternate Function**.

CHAPTER 7 12-BIT INTERVAL TIMER

7.1 Functions of 12-bit Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

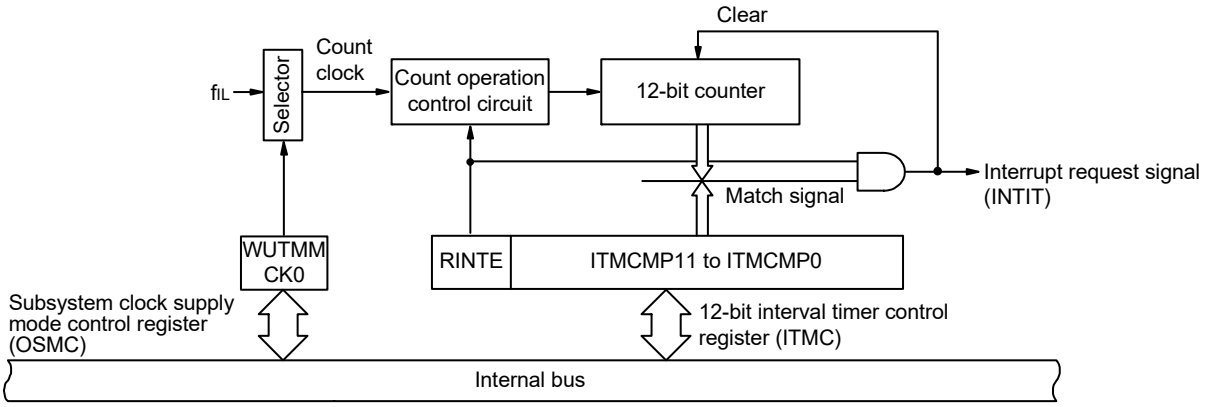
7.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 7 - 1 Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 2 (PER2)
	Peripheral reset control register 2 (PRR2)
	Subsystem clock supply mode control register (OSMC)
	12-bit interval timer control register (ITMC)

Figure 7 - 1 Block Diagram of 12-bit Interval Timer



7.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Subsystem clock supply mode control register (OSMC)
- 12-bit interval timer control register (ITMC)

7.3.1 Peripheral enable register 2 (PER2)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.
When the 12-bit interval timer is used, be sure to set bit 7 (TMKAEN) of this register to 1.
The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 7 - 2 Format of Peripheral enable register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
PER2	TMKAEN	0	0	0	0	0	0	0

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the 12-bit interval timer cannot be written. The read value is 0H. However, the SFR is not initialized. Note
1	Enables input clock supply. • SFR used by the 12-bit interval timer can be read and written.

Note To initialize the 12-bit interval timer and the SFR used by the 12-bit interval timer, use bit 7 (TMKARES) of PRR2.

Caution 1. Be sure to clear the following bits to 0.
Bits 0 to 6

Caution 2. Do not change the target bit in the PER2 register while operation of each peripheral function is enabled. Change the setting specified by PER2 while operation of each peripheral function assigned to PER2 is stopped.

7.3.2 Peripheral reset control register 2 (PRR2)

This register is used for individual reset control of each peripheral hardware.

This MCU controls reset and reset release of each peripheral hardware supported by the PRR2 register.

To reset the 12-bit interval timer, be sure to set bit 7 (TMKARES) to 1.

The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 3 Format of Peripheral reset control register 2 (PRR2)

Address: F00FDH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
PRR2	TMKARES	0	0	0	0	0	0	0

TMKARES	Reset control of 12-bit interval timer
0	12-bit interval timer reset release
1	12-bit interval timer reset state

7.3.3 Subsystem clock supply mode control register (OSMC)

Be sure to set the WUTMMCK0 bit to 1 while using 12-bit interval timer.

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

Reset signal generation clears this register to 00H.

Figure 7 - 4 Format of Subsystem clock supply mode control register (OSMC)

Address: F00F3H After reset: 00H R/W Note

Symbol	7	6	5	<4>	3	2	1	0
OSMC	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	—
0	Do not set the WUTMMCK0 bit to 0 while using 12-bit interval timer
1	Low-speed on-chip oscillator clock

Note Be sure to set bits 0, 1, 5, 6 and 7 to 0. Bits 2 and 3 are read-only, write is ignored.

7.3.4 12-bit interval timer control register (ITMC)

This register is used to set up the starting and stopping of the 12-bit interval timer operation and to specify the timer compare value.

The ITMC register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0FFFH.

Figure 7 - 5 Format of 12-bit interval timer control register (ITMC)

Address: FFF90H After reset: 0FFFH R/W

Symbol	15	14	13	12	11 to 0
ITMC	RINTE	0	0	0	ITCMP11 to ITCMP0

RINTE	12-bit interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITCMP11 to ITCMP0	Specification of the 12-bit interval timer compare value
001H	These bits generate a fixed-cycle interrupt (count clock cycles x (ITCMP setting + 1)).
• • •	
FFFH	
000H	Setting prohibit
Example interrupt cycles when 001H or FFFH is specified for ITCMP11 to ITCMP0 • ITCMP11 to ITCMP0 = 001H, count clock: when f_{IL} = 15 kHz 1/15 [kHz] × (1 + 1) = 0.133333 [ms] ≅ 133.33 [μs] • ITCMP11 to ITCMP0 = FFFH, count clock: when f_{IL} = 15 kHz 1/15 [kHz] × (4095 + 1) ≅ 273.07 [ms]	

Caution 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the ITIF flag, and then enable the interrupt servicing.

Caution 2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

Caution 3. When setting the RINTE bit after returned from standby mode and entering standby mode again, confirm that the written value of the RINTE bit is reflected, or wait that more than one clock of the count clock has elapsed after returned from standby mode. Then enter standby mode.

Caution 4. Only change the setting of the ITCMP11 to ITCMP0 bits when RINTE = 0.
However, it is possible to change the settings of the ITCMP11 to ITCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

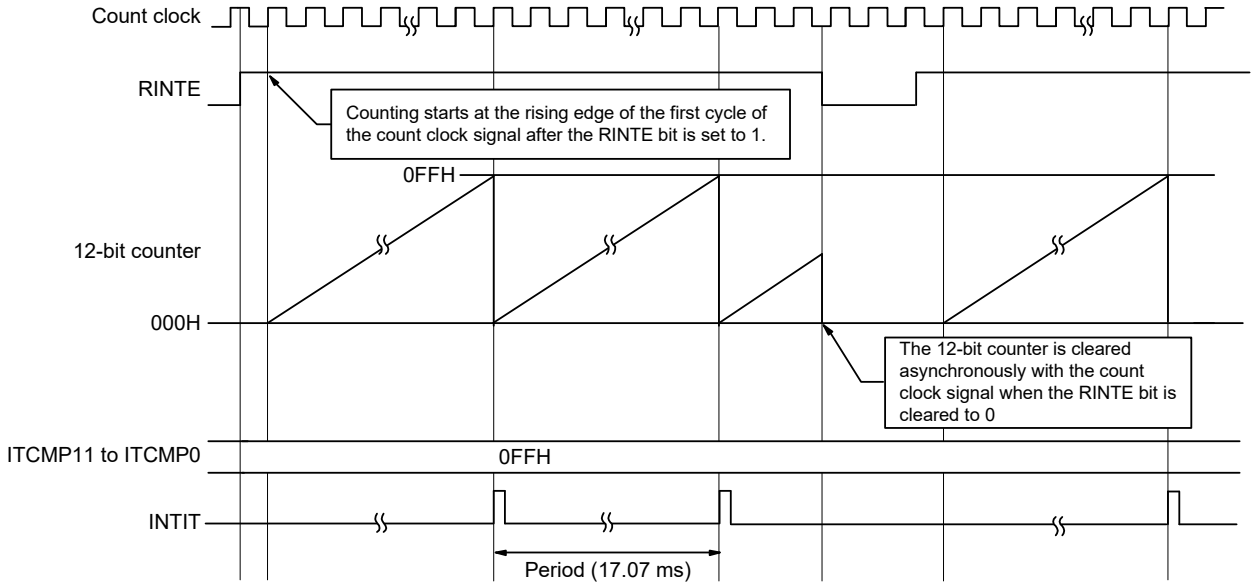
7.4 12-bit Interval Timer Operation

7.4.1 12-bit interval timer operation timing

The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate an 12-bit interval timer that repeatedly generates interrupt requests (INTIT).
When the RINTE bit is set to 1, the 12-bit counter starts counting.
When the 12-bit counter value matches the value specified for the ITCMP11 to ITCMP0 bits, the 12-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the 12-bit interval timer is as follows.

Figure 7 - 6 12-bit Interval Timer Operation Timing (ITCMP11 to ITCMP0 = 0FFH, count clock: f_{IL} = 15 kHz)

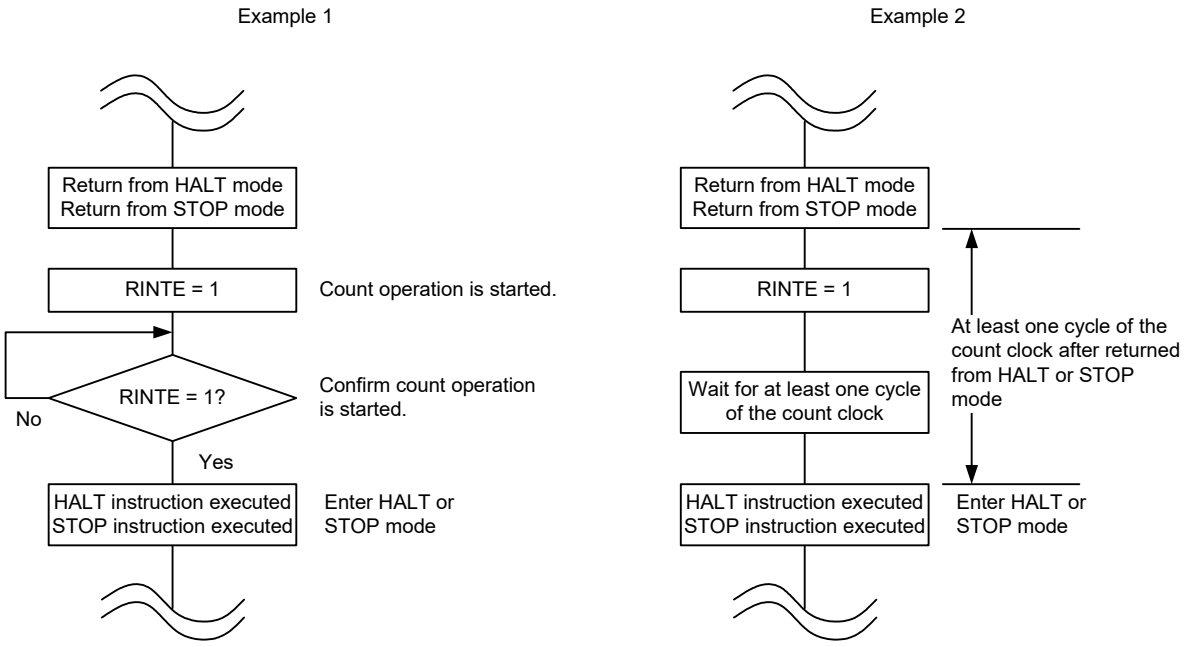


7.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

When setting the RINTE bit after returned from HALT or STOP mode and entering HALT or STOP mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock.
Then, enter HALT or STOP mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter HALT or STOP mode (see **Example 1** in **Figure 7 - 7**).
- After setting RINTE to 1, wait for at least one cycle of the count clock and then enter HALT or STOP mode (see **Example 2** in **Figure 7 - 7**).

Figure 7 - 7 Procedure of entering to HALT or STOP mode after setting RINTE to 1



CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

8.1 Functions of Clock Output/Buzzer Output Controller

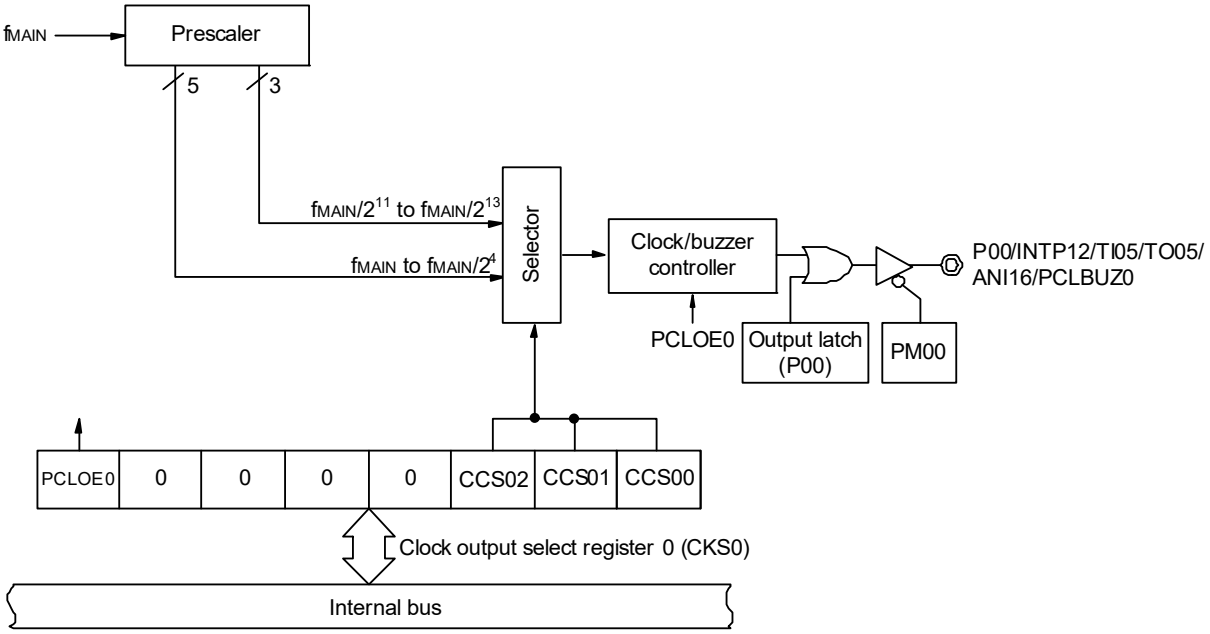
The clock output controller is intended for clock output for supply to peripheral ICs. Buzzer output is a function to output a square wave of buzzer frequency.

The PCLBUZ0 pin outputs a clock selected by clock output select register n (CKSn).

Figure 8 - 1 shows the Block Diagram of Clock Output/Buzzer Output Controller.

Remark n = 0

Figure 8 - 1 Block Diagram of Clock Output/Buzzer Output Controller



8.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 8 - 1 Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Port mode register 0 (PM0) Port register 0 (P0)

8.3 Registers Controlling Clock Output/Buzzer Output Controller

8.3.1 Clock output select registers n (CKSn)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZ0), and set the output clock.

Select the clock to be output from the PCLBUZ0 pin by using the CKSn register.

The CKSn register are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 8 - 2 Format of Clock output select registers n (CKSn)

Address: FFFA5H (CKS0) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	0	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZ0 pin output enable/disable specification
0	Output disable (default)
1	Output enable

CCSn2	CCSn1	CCSn0	PCLBUZn pin output clock selection				
				f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 24 MHz
0	0	0	f_{MAIN}	5 MHz	10 MHz	Setting prohibited	Setting prohibited
0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz	Setting prohibited
0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	3 MHz
1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	1.5 MHz
1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz

Caution 1. Change the output clock after disabling clock output (PCLOEn = 0).
Caution 2. To shift to STOP mode, set PCLOEn = 0 before executing the STOP instruction.

Remark 1. n = 0
Remark 2. f_{MAIN}: Main system clock frequency

8.3.2 Registers controlling port functions of pins to be used for clock or buzzer output

Using a port pin for clock or buzzer output requires setting of the registers that control the port functions multiplexed on the target pin (port mode register (PMxx), port register (Pxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

Specifically, using a port pin with a multiplexed clock or buzzer output function (e.g. P00/INTP12/TI05/TO05/ANI16/PCLBUZ0) for clock or buzzer output, requires setting the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

8.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.

The PCLBUZ0 pin outputs a clock/buzzer selected by the clock output select register 0 (CKS0).

8.4.1 Operation as output pin

The PCLBUZ0 pin is output as the following procedures.

<1> Set 0 in the bit of the port mode register (PMxx) and port register (Pxx) which correspond to the port which has a pin used as the PCLBUZ0 pin.

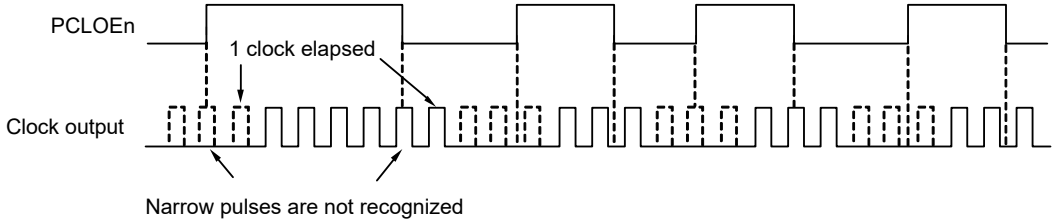
<2> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2) of the clock output select register (CKSn) of the PCLBUZ0 pin (output in disabled status).

<3> Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Remark 1. The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOEn bit) is switched. At this time, pulses with a narrow width are not output. Figure 8 - 3 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.

Remark 2. n = 0

Figure 8 - 3 Timing of Outputting Clock from PCLBUZ0 Pin



8.5 Cautions of clock output/buzzer output controller

If STOP mode is entered within 1.5 clock cycles output from the PCLBUZ0 pin after the output is disabled (PCLOEn = 0), the PCLBUZ0 output width becomes shorter.

CHAPTER 9 WATCHDOG TIMER

9.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the option byte (000C0H).
The watchdog timer operates on the low-speed on-chip oscillator clock (f_{IL}).
The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.
Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1.
For details of the RESF register, see **CHAPTER 16 RESET FUNCTION**.
When 75% + 1/2 f_{IL} of the overflow time is reached, an interval interrupt can be generated.

9.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9 - 1 Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

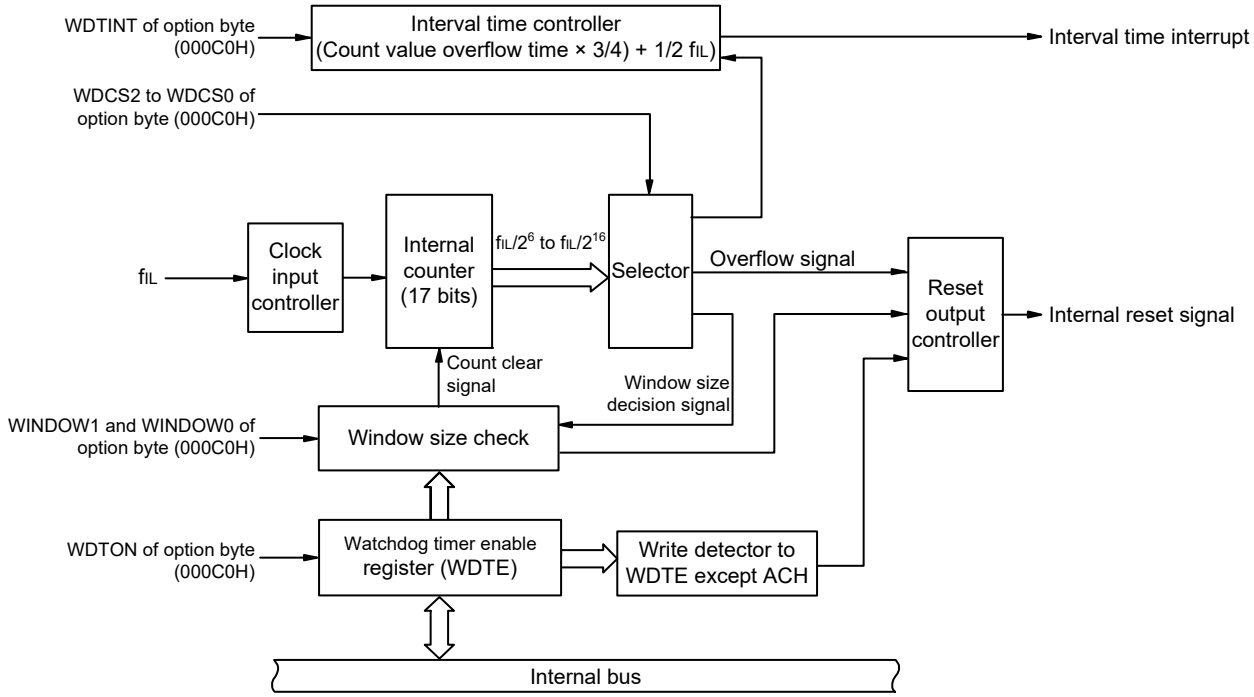
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 9 - 2 Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the option byte, see **CHAPTER 21 OPTION BYTE**.

Figure 9 - 1 Block Diagram of Watchdog Timer



Remark fIL: Low-speed on-chip oscillator clock

9.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

9.3.1 Watchdog timer enable register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and starts counting again.

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

Reset signal generation sets this register to 9AH or 1AH Note.

Figure 9 - 2 Format of Watchdog timer enable register (WDTE)

Address: FFFABH After reset: 1AH/9AH Note R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Caution 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

Caution 2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

Caution 3. The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

9.4 Operation of Watchdog Timer

9.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

• Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 21**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

• Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **9.4.2** and **CHAPTER 21**).
• Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **9.4.3** and **CHAPTER 21**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to the WDTE register, an internal reset signal is generated.
   An internal reset signal is generated in the following cases.

• If a 1-bit manipulation instruction is executed on the WDTE register
• If data other than “ACH” is written to the WDTE register

Caution 1.When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

Caution 2. After “ACH” is written to the WDTE register, an error of up to 2 clocks (f_{IL}) may occur before the watchdog timer is cleared.

Caution 3. The watchdog timer can be cleared immediately before the count value overflows.

Caution 4. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		
In SNOOZE mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.
When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

9.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) during the window open period before the overflow time.

The following overflow times can be set.

Table 9 - 3 Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)

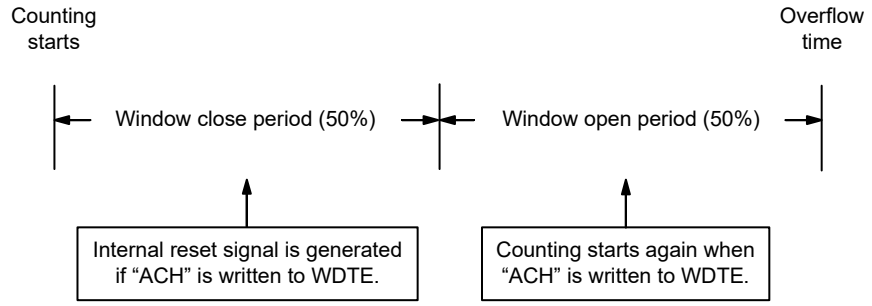
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

9.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 50%



Caution When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period can be set is as follows.

Table 9 - 4 Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	Setting prohibited
0	1	50%
1	0	75% Note
1	1	100%

Note When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)	1899.59 ms to 2570.04 ms

Caution When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.

Remark If the overflow time is set to $2^9/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^9/f_{IL}$ (MAX.) = 2^9/17.25 kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^9/f_{IL}$ (MIN.) × (1 - 0.5) = 0 to 2^9/12.75 kHz × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{IL}$ (MIN.) × (1 - 0.5) to $2^9/f_{IL}$ (MAX.) = 2^9/12.75 kHz × 0.5 to 2^9/17.25 kHz = 20.08 to 29.68 ms

9.4.4 Setting watchdog timer interval interrupt

Setting bit 7 (WDTINT) of an option byte (000C0H) can generate an interval interrupt (INTWDTI) when 75% + 1/2 f_{IL} of the overflow time is reached.

Table 9 - 5 Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2 f_{IL} of overflow time is reached.

Caution **When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.**
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

Remark The watchdog timer continues counting even after INTWDTI is generated (until ACH is written to the watchdog timer enable register (WDTE)). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

CHAPTER 10 A/D CONVERTER

The number of analog input channels of the A/D converter is shown below.

	with USB	without USB
	32-pin	32-pin
Analog input channels	8 ch (ANI0 to ANI5, ANI16 and ANI17)	8 ch (ANI0 to ANI5, ANI16 and ANI17)

10.1 Function of A/D Converter

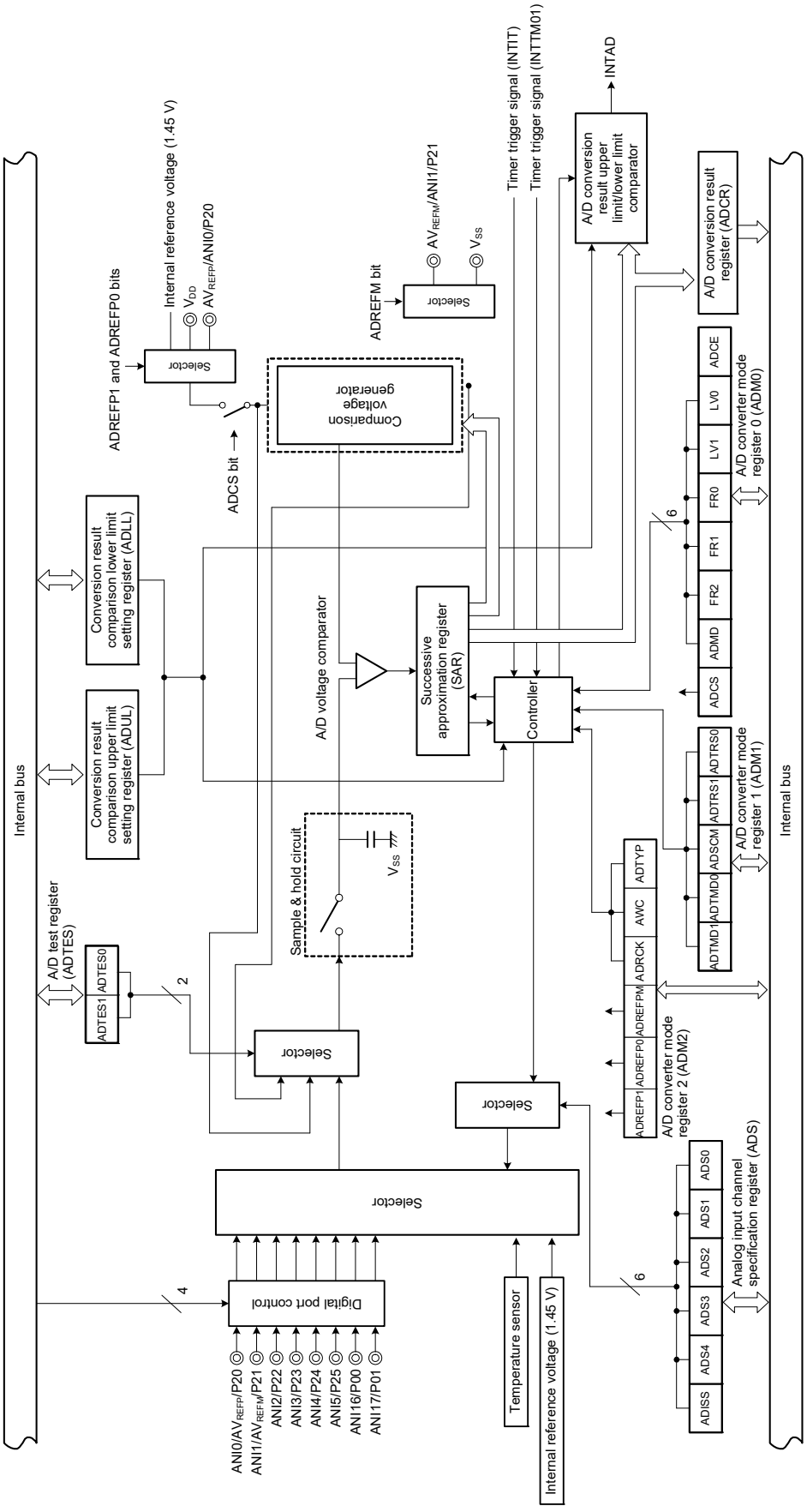
The A/D converter is used to convert analog input signals into digital values, and is configured to control analog inputs, including up to 8 channels of A/D converter analog inputs (ANI0 to ANI5, ANI16 and ANI17). 10-bit or 8-bit resolution can be selected by the ADTYP bit of the A/D converter mode register 2 (ADM2).
The A/D converter has the following function.

- 10-bit/8-bit resolution A/D conversion
  10-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI5, ANI16 and ANI17. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power is turned on by detecting a hardware trigger while the system is off and in the conversion standby state, and conversion is then started automatically after the stabilization wait time passes. When using the SNOOZE mode function, specify the hardware trigger wait mode.
Channel selection mode	Select mode	A/D conversion is performed on the analog input of one selected channel.
	Scan mode	A/D conversion is performed on the analog input of four channels in order. Four consecutive channels can be selected from ANI0 to ANI5, ANI16 and ANI17 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.
Operation voltage mode	Standard 1 or standard 2 mode	Conversion is done in the operation voltage range of 2.7 V ≤ V_{DD} ≤ 5.5 V.
Sampling time selection	Sampling clock cycles: 7 f_{AD}	The sampling time in standard 1 mode is seven cycles of the conversion clock (f_{AD}). Select this mode when the output impedance of the analog input source is high and the sampling time should be long.
	Sampling clock cycles: 5 f_{AD}	The sampling time in standard 2 mode is five cycles of the conversion clock (f_{AD}). Select this mode when enough sampling time is ensured (for example, when the output impedance of the analog input source is low).

Figure 10 - 1 Block Diagram of A/D Converter



10.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI5, ANI16 and ANI17

These are the analog input pins of the 8 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 bit of the SAR is reset.

After that, bit 8 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 9, to which the result has been already set.

Bit 9 = 0: (1/4 AV_{REF})
Bit 9 = 1: (3/4 AV_{REF})

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 8 = 1
Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

Remark AVREF: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

(4) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR register is a register that sets voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 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 (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register 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 (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(8) Controller

This circuit controls the conversion time of an input analog 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 INTAD through the A/D conversion result upper limit/lower limit comparator.

(9) AV_{REFP} pin

This pin inputs an external reference voltage (AV_{REFP}).

If using AV_{REFP} as the + side reference voltage of the A/D converter, set the ADREFP1 and ADREFP0 bits of A/D converter mode register 2 (ADM2) to 0 and 1, respectively.

The analog signals input to ANI0 to ANI5, ANI16 and ANI17 are converted to digital signals based on the voltage applied between AVREFP and the - side reference voltage (AV_{REFM}/V_{SS}).

In addition to AV_{REFP}, it is possible to select V_{DD} or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

(10) AV_{REFM} pin

This pin inputs an external reference voltage (AV_{REFM}). If using AV_{REFM} as the – side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AV_{REFM}, it is possible to select V_{SS} as the – side reference voltage of the A/D converter.

10.3 Registers Controlling A/D Converter

The A/D converter is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- Port mode control registers 0, 2 (PMC0, PMC2)
- Port mode registers 0, 2 (PM0, PM2)

10.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 2 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	IICA2EN Note	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFR used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFR used by the A/D converter can be read/written.

Note This bit is incorporated with R9A02G0151, but is not incorporated with R9A02G0150.

Caution 1. When setting the A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1. If ADCEN = 0, the values of the A/D converter control registers are cleared to their initial values and writing to them is ignored (except for port mode registers 0 and 2 (PM0, PM2), port mode control registers 0, and 2 (PMC0, PMC2)).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **10-bit A/D conversion result register (ADCR)**
- **8-bit A/D conversion result register (ADCRH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES).**

Caution 2. Be sure to clear the bits 1, 3 to 0.

10.3.2 A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 3 Format of A/D Converter Mode Register 0 (ADM0)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	ADMD	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation [When read] Conversion stopped/standby status
1	Enables conversion operation [When readNote 2 Note 2] While in the software trigger mode: Conversion operation status While in the hardware trigger wait mode: A/D power supply stabilization wait status + conversion operation status

ADMD	Specification of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Note 1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Table 10 - 3 A/D Conversion Time Selection**.

Note 2. While in the software trigger 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 from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Caution 1. Change the ADMD, FR2 to FR0, LV1, and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. Do not set the ADCS bit to 1 and the ADCE bit to 0 at the same time.

Caution 3. 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 set these bits in the order described in 11.7 A/D Converter Setup Flowchart.

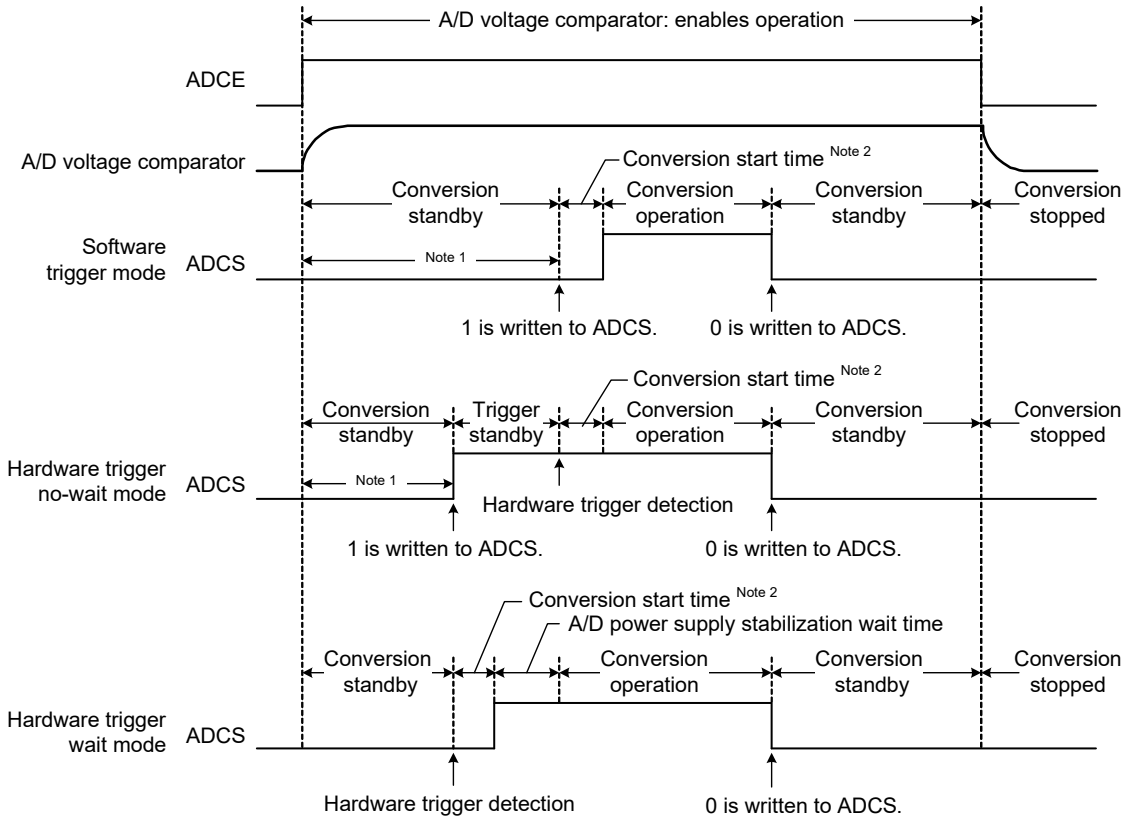
Table 10 - 1 Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Conversion stopped state
0	1	Conversion standby state
1	0	Setting prohibited
1	1	Conversion-in-progress state

Table 10 - 2 Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	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.
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.

Figure 10 - 4 Timing Chart When A/D Voltage Comparator Is Used



Note 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer to stabilize the internal circuit.

Note 2. The following time is the maximum amount of time necessary to start conversion.

ADM0			Conversion Clock (f_{AD})	Conversion Start Time (Number of fCLK Clocks)	
FR2	FR1	FR0		Software trigger mode/ Hardware trigger no wait mode	Hardware trigger wait mode
0	0	0	f_{CLK}/64	63	1
0	0	1	f_{CLK}/32	31	
0	1	0	f_{CLK}/16	15	
0	1	1	f_{CLK}/8	7	
1	0	0	f_{CLK}/6	5	
1	0	1	f_{CLK}/5	4	
1	1	0	f_{CLK}/4	3	
1	1	1	f_{CLK}/2	1	

However, for the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 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.

Caution 1. 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 status.

Caution 2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

Caution 3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).

Caution 4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:

Hardware trigger no wait mode: 2 fCLK clock + A/D conversion time
Hardware trigger wait mode: 2 fCLK clock + stabilization wait time + A/D conversion time

Remark fCLK: CPU/peripheral hardware clock frequency

Table 10 - 3 A/D Conversion Time Selection (1/2)

(1) When there is no A/D power supply stabilization wait time

Normal mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (fAD)	Number of Conversion Clock Note 1	Conversion Time	Conversion Time at 10-Bit Resolution				
									2.7 V ≤ VDD ≤ 5.5 V				
FR2	FR1	FR0	LV1	LV0					fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 24 MHz
0	0	0	1	0	Normal 1	fCLK/64	19 fAD (number of sampling clock: 7 fAD)	1216/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				fCLK/32		608/fCLK				38 μs	25.3333 μs
0	1	0				fCLK/16		304/fCLK			38 μs	19 μs	12.6667 μs
0	1	1				fCLK/8		152/fCLK		38 μs	19 μs	9.5 μs	6.3333 μs
1	0	0				fCLK/6		114/fCLK		28.5 μs	14.25 μs	7.125 μs	4.7500 μs
1	0	1				fCLK/5		95/fCLK		23.75 μs	11.875 μs	5.9375 μs	3.9583 μs
1	1	0				fCLK/4		76/fCLK		19 μs	9.5 μs	4.75 μs	3.1667 μs Note 2, 3
1	1	1				fCLK/2		38/fCLK	38 μs	9.5 μs	4.75 μs	2.375 μs Note 2, 3	Setting prohibited
0	0	0	1	1	Normal 2	fCLK/64	17 fAD (number of sampling clock: 5 fAD)	1088/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				fCLK/32		544/fCLK				34 μs	22.6667 μs
0	1	0				fCLK/16		272/fCLK			34 μs	17 μs	11.3333 μs
0	1	1				fCLK/8		136/fCLK		34 μs	17 μs	8.5 μs	5.6667 μs
1	0	0				fCLK/6		102/fCLK		25.5 μs	12.75 μs	6.375 μs	4.2500 μs
1	0	1				fCLK/5		85/fCLK		21.25 μs	10.625 μs	5.3125 μs	3.5417 μs Note 2
1	1	0				fCLK/4		68/fCLK		17 μs	8.5 μs	4.25 μs	2.8333 μs Note 2, 3
1	1	1				fCLK/2		34/fCLK	34 μs	8.5 μs	4.25 μs	2.125 μs Note 2, 3	Setting prohibited

Note 1. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (fAD).

Note 2. This value is prohibited when using the temperature sensor

Note 3. Setting prohibited in the 3.6 V

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (tCONV) described in 2.6.1 A/D converter characteristics of the R9A02G015 Data Sheet (R19DS0101E).

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark fCLK: CPU/peripheral hardware clock frequency

Table 10 - 4 A/D Conversion Time Selection (2/2)

(2) When there is A/D power supply stabilization wait time

Normal mode 1, 2 (hardware trigger wait mode Note 1)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (fAD)	Number of Stabilization Wait Clock	Number of Conversion Clock Note 2	Stabilization Wait Time+ Conversion Time	Conversion Time at 10-Bit Resolution				
										2.7 V ≤ VDD ≤ 5.5 V				
FR2	FR1	FR0	LV1	LV0						fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 24 MHz
0	0	0	0	0	Normal 1	fCLK/64	8 fAD	19 fAD (number of sampling clock: 7 fAD)	1728/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				fCLK/32			864/fCLK				54 μs	36.0 μs
0	1	0				fCLK/16			432/fCLK			54 μs	27 μs	18.0 μs
0	1	1				fCLK/8			216/fCLK		54 μs	27 μs	13.5 μs	9.0 μs
1	0	0				fCLK/6			162/fCLK		40.5 μs	20.25 μs	10.125 μs	6.8 μs
1	0	1				fCLK/5			135/fCLK		33.75 μs	16.875 μs	8.4375 μs	5.6 μs
1	1	0				fCLK/4			108/fCLK		27 μs	13.5 μs	6.75 μs	4.5 μs Note 3, 4
1	1	1				fCLK/2			54/fCLK	54 μs	13.5 μs	6.75 μs	3.375 μs Note 3, 4	Setting prohibited
0	0	0	0	1	Normal 2	fCLK/64	8 fAD	17 fAD (number of sampling clock: 5 fAD)	1600/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				fCLK/32			800/fCLK				50 μs	33.3333 μs
0	1	0				fCLK/16			400/fCLK			50 μs	25 μs	16.6667 μs
0	1	1				fCLK/8			200/fCLK		50 μs	25 μs	12.5 μs	8.3333 μs
1	0	0				fCLK/6			150/fCLK		37.5 μs	18.75 μs	9.375 μs	6.2500 μs
1	0	1				fCLK/5			125/fCLK		31.25 μs	15.625 μs	7.8125 μs	5.2083 μs Note 3
1	1	0				fCLK/4			100/fCLK		25 μs	12.5 μs	6.25 μs	4.1667 μs Note 3, 4
1	1	1				fCLK/2			50/fCLK	50 μs	12.5 μs	6.25 μs	3.125 μs Note 3, 4	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 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 (see table 11-3 (1/4)).

Note 2. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (fAD).

Note 3. This value is prohibited when using the temperature sensor

Note 4. Setting prohibited in the 3.6 V

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (tCONV) described in 2.6.1 A/D converter characteristics of the R9A02G015 Data Sheet (R19DS0101E). Note that the conversion time (tCONV) does not include the A/D power supply stabilization wait time.

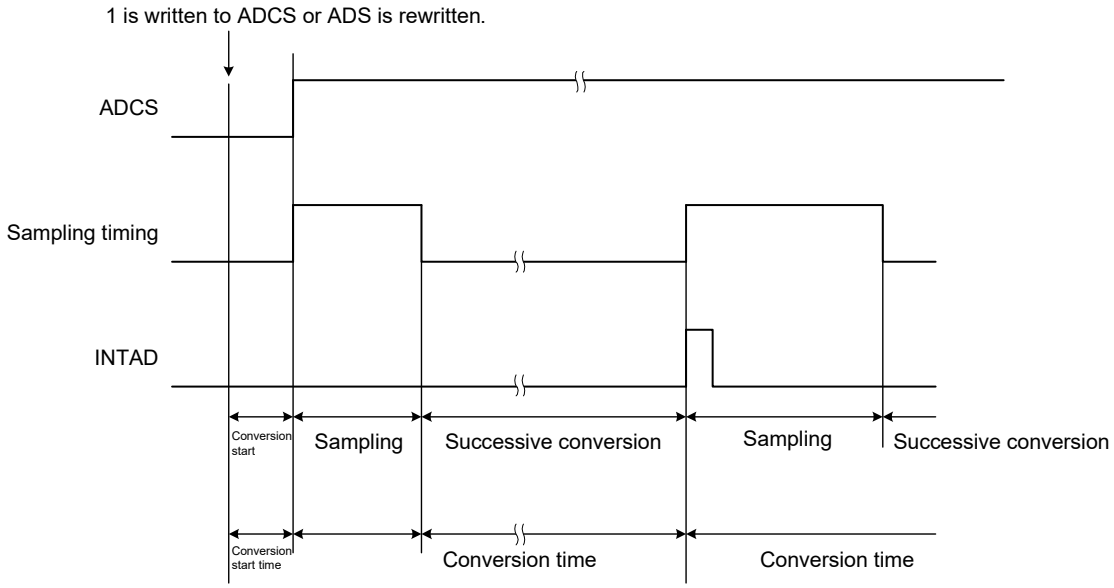
Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time, including the A/D power supply stabilization wait time from the hardware trigger detection.

Remark fCLK: CPU/peripheral hardware clock frequency

Figure 10 - 5 A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



10.3.3 A/D converter mode register 1 (ADM1)

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal.

The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 6 Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF32H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	—	Software trigger mode
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	End of timer channel 1 count or capture interrupt signal (INTTM01)
0	1	Setting prohibited
1	0	Setting prohibited
1	1	12-bit interval timer interrupt signal (INTIT)

Caution 1. Rewrite the value of the ADM1 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 f_{CLK} clock + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

Caution 3. In modes other than SNOOZE mode, input of the next INTIT will not be recognized as a valid hardware trigger for up to four f_{CLK} cycles after the first INTIT is input.

10.3.4 A/D converter mode register 2 (ADM2)

This register is used to select the + side or - side reference voltage 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.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 7 Format of A/D Converter Mode Register 2 (ADM2) (1/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	0	Supplied from V_{DD}
0	1	Supplied from P20/AV_{REFP}/ANI0
1	0	Supplied from the internal reference voltage (1.45 V) Note
1	1	Setting prohibited

- When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures.
  (1) Set ADCE = 0
  (2) Change the values of ADREFP1 and ADREFP0
  (3) Reference voltage stabilization wait time (A)
  (4) Set ADCE = 1
  (5) Reference voltage stabilization wait time (B)
  When ADREFP1 and ADREFP0 are set to 1 and 0, the setting is changed to A = 5 μs, B = 1 μs.
  When ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1, A needs no wait and B = 1 μs.
- When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the temperature sensor output and internal reference voltage output.
  Be sure to perform A/D conversion while ADISS = 0.

ADREFM	Selection of the – side reference voltage source of the A/D converter
0	Supplied from V_{SS}
1	Supplied from P21/AV_{REFM}/ANI1

ADRCK	Selection of the – side reference voltage source of the A/D converter
0	The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (<1>).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (<2>) or the ADUL register < the ADCR register (<3>).
Figures 10 - 8 shows the generation range of the interrupt signal (INTAD) for <1> to <3>.	

Note This setting can be used only in HS (high-speed main) mode.

Caution 1. Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode.

Caution 3. When using AV_{REFP} and AV_{REFM}, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

Figure 10 - 8 Format of A/D Converter Mode Register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).

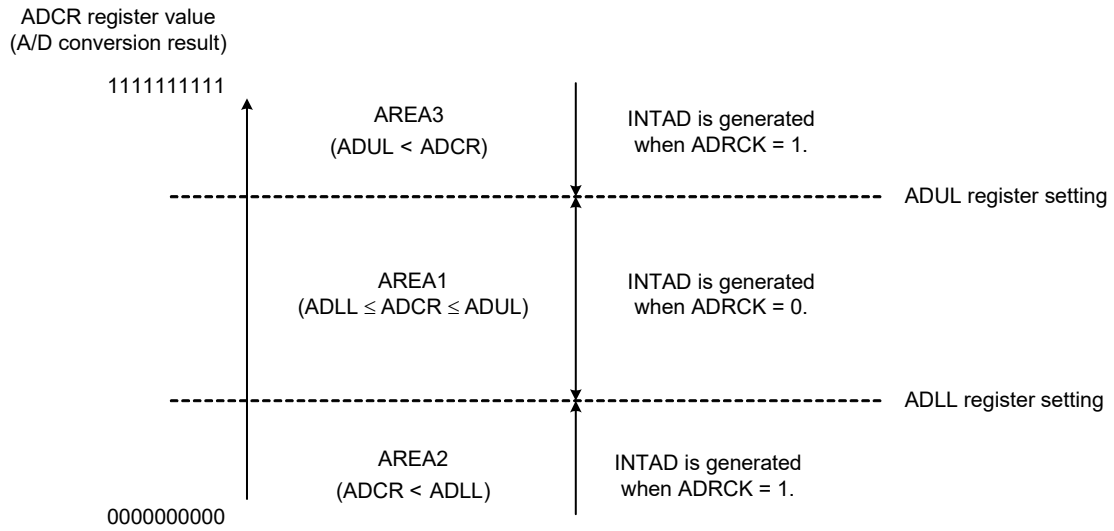
- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited.
- Using the SNOOZE mode function in the sequential conversion mode is prohibited.
- When using the SNOOZE mode function, specify a hardware trigger interval of at least "shift time to SNOOZE mode Note + conversion start time + A/D power supply stabilization wait time + A/D conversion time +2 f_{CLK} clock"
- Even when using SNOOZE mode, be sure to set the AWC bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode.
  Also, be sure to change the AWC bit to 0 after returning from STOP mode to normal operation mode.
  If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode.

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note Refer to "From STOP to SNOOZE" in 19.3.3 SNOOZE mode.

Caution **Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCS bit of A/D converter mode register 0 (ADM0) being 0).**

Figure 10 - 9 ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD does not occur, the A/D conversion result is not stored in the ADCR or ADCRH register.

10.3.5 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH Note.

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see Figure 11-8), the result is not stored.

Figure 10 - 10 Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Caution 1. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (ADCR1 and ADCR0).

Caution 2. When the ADCR register is accessed in 16-bit units, the higher 10 bits of the conversion result are read in order starting at bit 15.

10.3.6 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored Note.

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

Reset signal generation clears this register to 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see Figure 10 - 8), the result is not stored.

Figure 10 - 11 Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution 1. When writing to the A/D converter mode register 0 (ADM0) and analog input channel specification register (ADS), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0 and ADS. Using timing other than the above may cause an incorrect conversion result to be read.

10.3.7 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 12 Format of Analog Input Channel Specification Register (ADS) (1/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin
0	1	0	0	0	0	ANI16	P00/ANI16 pin
0	1	0	0	0	1	ANI17	P01/ANI17 pin
1	0	0	0	0	0	–	Temperature sensor output Note
1	0	0	0	0	1	–	Internal reference voltage output (1.45 V) Note
Other than the above						Setting prohibited	

Note Operation is possible only in HS (high-speed main) mode.

Figure 10 - 13 Format of Analog Input Channel Specification Register (ADS) (2/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
						Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
Other than the above						Setting prohibited			

Caution 1. Be sure to clear bits 5 and 6 to 0.

Caution 2. Set a channel to be used for A/D conversion in the input mode by using port mode registers 0, 2 (PM0, PM2).

Caution 3. Do not set the pin that is set by port mode control registers 0, 2 (PMC0, PMC2) as digital I/O by the ADS register.

Caution 4. Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 5. If using AV_{REFP} as the + side reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 6. If using AV_{REFM} as the - side reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.

Caution 7. If the ADISS bit is set to 1, the internal reference voltage output (1.45 V) 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 10.7.4 Setup when using temperature sensor (example for software trigger mode and one-shot conversion mode).

Caution 8. Do not set the ADISS bit to 1 when shifting to STOP mode.

10.3.8 Conversion result comparison upper limit setting register (ADUL)

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 (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in Figure 10 - 8).
The ADUL register can be set by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.

Caution When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL register.

Figure 10 - 14 Format of Conversion Result Comparison Upper Limit Setting Register (ADUL)

Address: F0011H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

10.3.9 Conversion result comparison lower limit setting register (ADLL)

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 (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in Figure 10 - 8).
The ADLL register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 10 - 15 Format of Conversion Result Comparison Lower Limit Setting Register (ADLL)

Address: F0012H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Caution 1. When A/D conversion with 10-bit resolution is selected, the eight higher-order bits of the 10-bit A/D conversion result register (ADCR) are compared with the values in the ADUL and ADLL registers.
Caution 2. Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).
Caution 3. The setting of the ADUL registers must be greater than that of the ADLL register.

10.3.10 A/D test register (ADTES)

This register is used to select the + side reference voltage or - side reference voltage for the converter, an analog input
channel (ANIxx), the temperature sensor output, or the internal reference voltage output (1.45 V) 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.

The ADTES register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 10 - 16 Format of A/D Test Register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output Note/internal reference voltage output (1.45 V) Note (This is specified using the 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 ADREFP1 or ADREFP0 bit of the ADM2 register)
Other than the above		Setting prohibited

Note The temperature sensor output and internal reference voltage output (1.45 V) can be selected only in the HS (high-speed main) mode.

10.3.11 Registers controlling port function of analog input pins

Set up the registers for controlling the functions of the ports shared with the analog input pins of the A/D converter (port mode registers (PMxx), port mode control registers (PMCxx). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.6 Port mode control registers (PMCxx)**.

When using the ANI0 to ANI5, ANI16 and ANI17 pins for analog input of the A/D converter, set the port mode register (PMxx) bit and port mode control register (PMCxx) bit corresponding to each port to 1.

10.4 A/D Converter Conversion 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 9 of the successive approximation register (SAR) is set. 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 bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB bit is reset to 0.

<5> Next, bit 8 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 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.

- Sampled voltage ≤ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched. At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

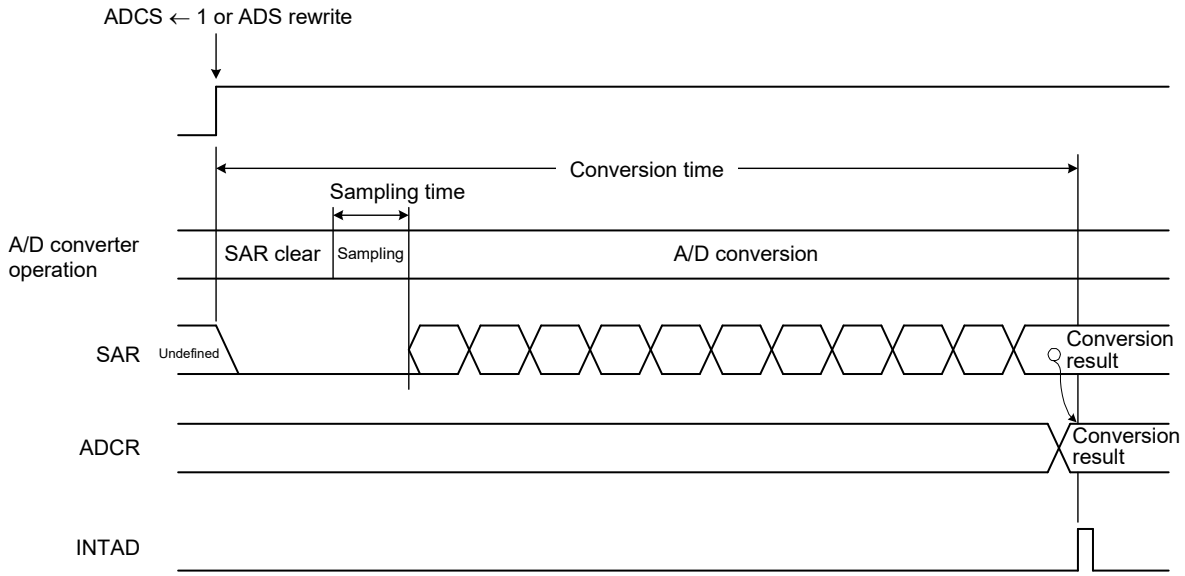
<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0 [Note].
To stop the A/D converter, clear the ADCS bit to 0.

Note 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.

Remark 1. Two types of the A/D conversion result registers are available.
- ADCR register (16 bits): Store 10-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

Remark 2. AV_{REF}: The + side reference voltage of the A/D converter.This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

Figure 10 - 17 Conversion Operation of A/D Converter (Software Trigger 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.
Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

10.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI5, ANI16 and ANI17) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

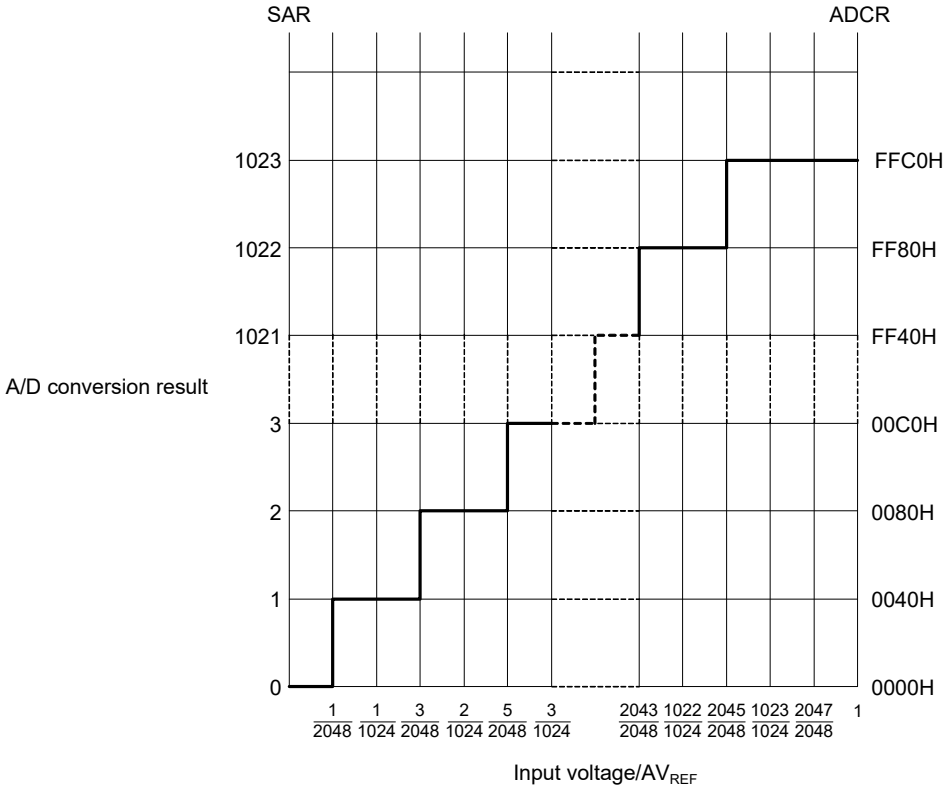
or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses
V_{AIN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: A/D conversion result register (ADCR) value
SAR: Successive approximation register

Figure 10 - 18 shows the Relationship Between Analog Input Voltage and A/D Conversion Result.

Figure 10 - 18 Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

10.6 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 **10.7 A/D Converter Setup Flowchart**.

10.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), 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 register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) 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.

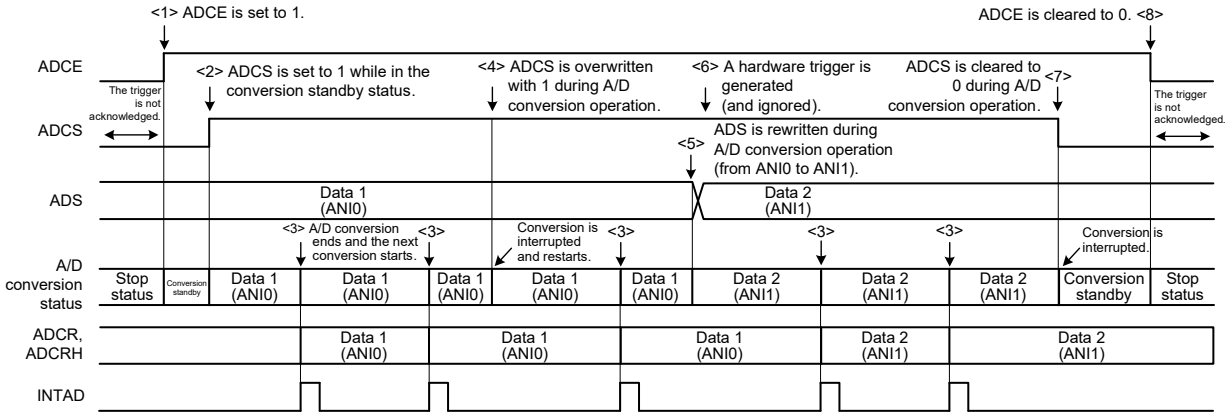
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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 system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 10 - 19 Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



10.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), 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 register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

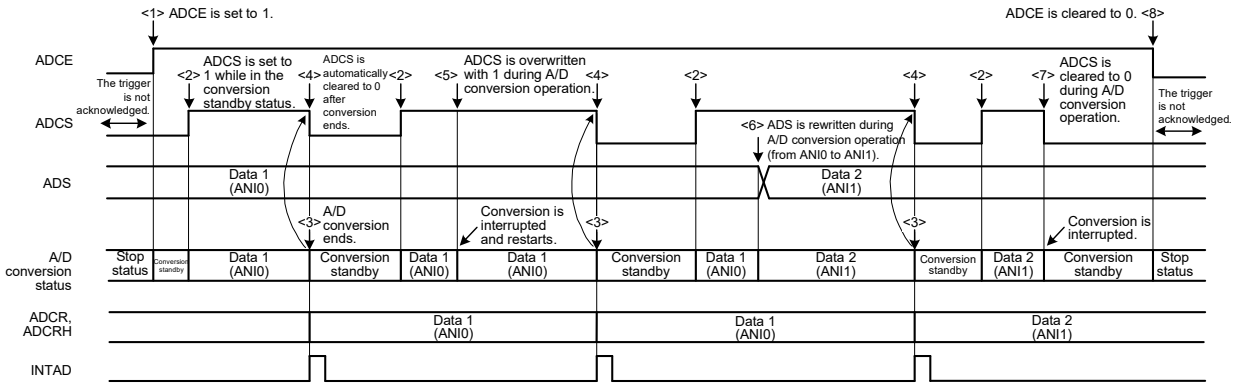
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 10 - 20 Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



10.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), 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 (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel 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.

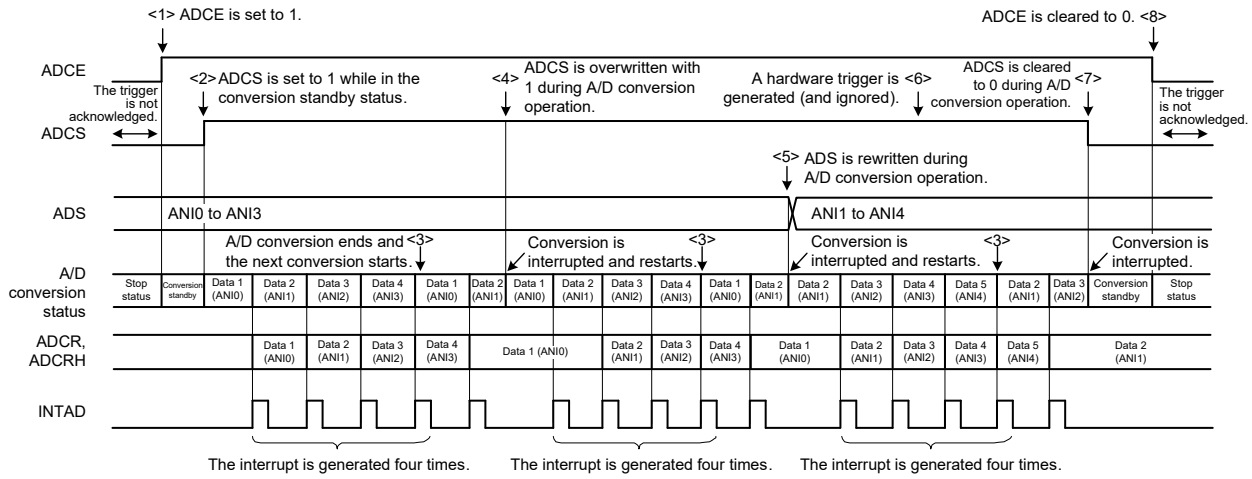
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<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 system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 10 - 21 Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



10.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), 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 (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

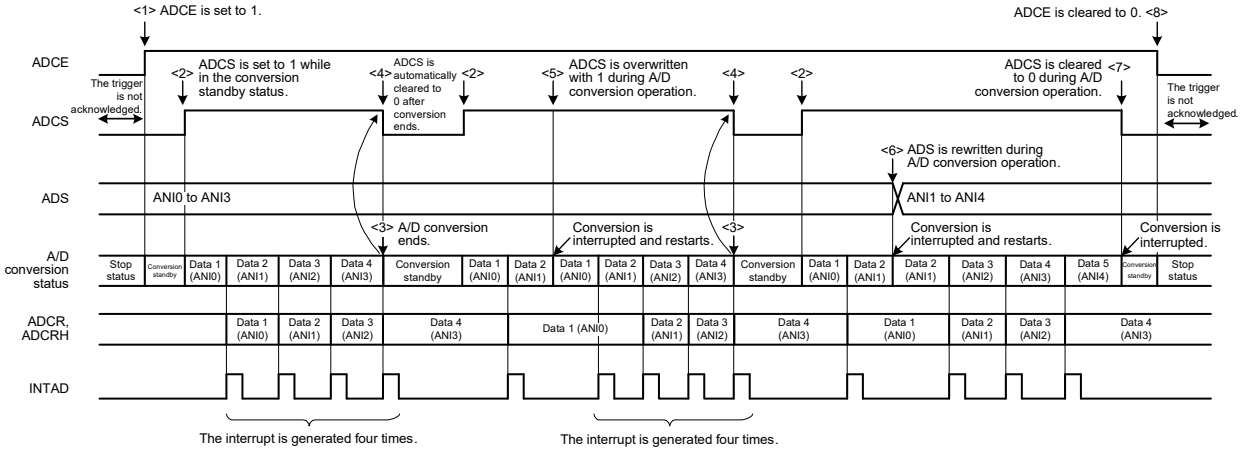
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<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 system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 10 - 22 Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



10.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, 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 register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) 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.

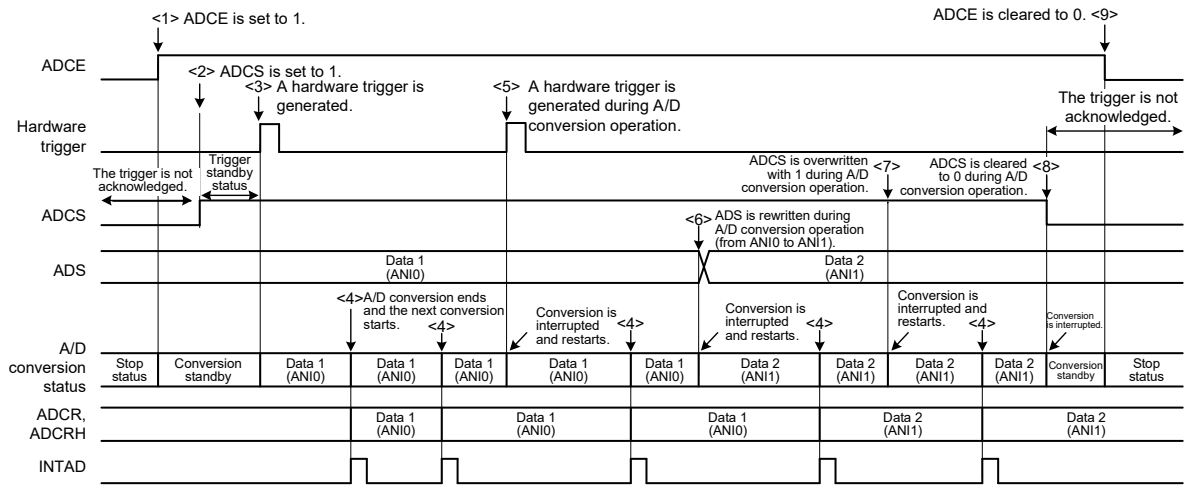
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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 system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10 - 23 Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



10.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, 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 register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<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.

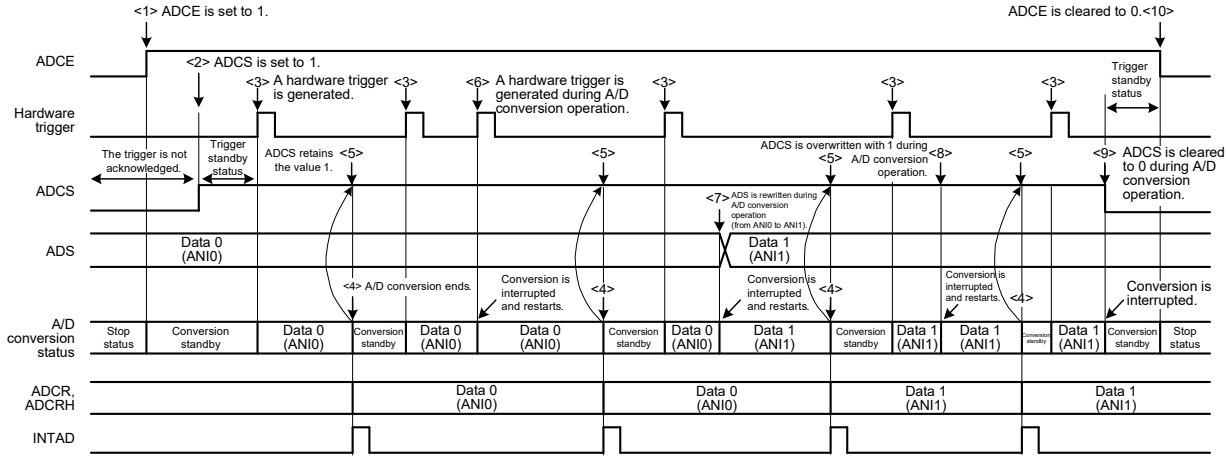
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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 system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10 - 24 Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



10.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, 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 (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel 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.

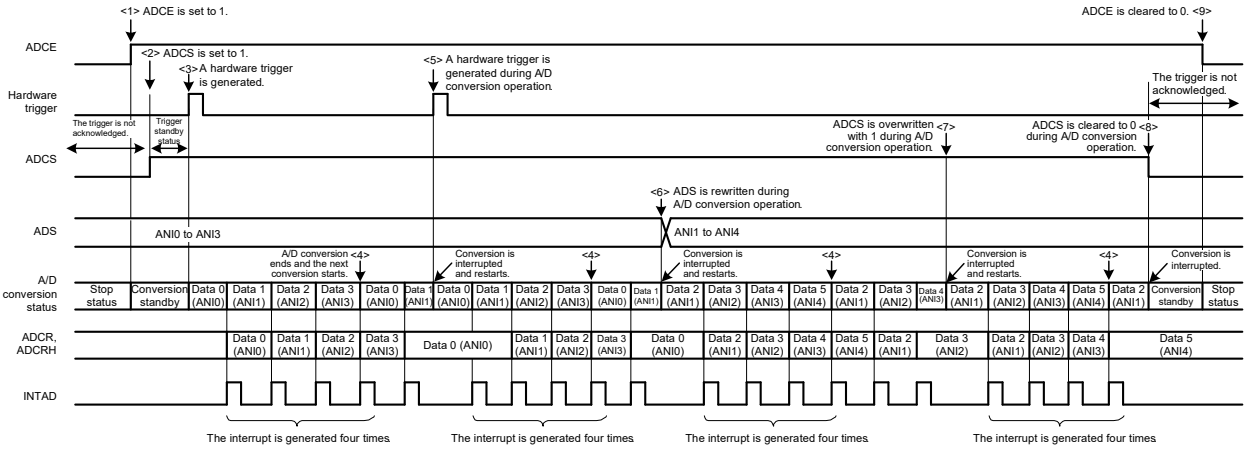
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<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 system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 10 - 25 Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



10.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, 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 (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<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.

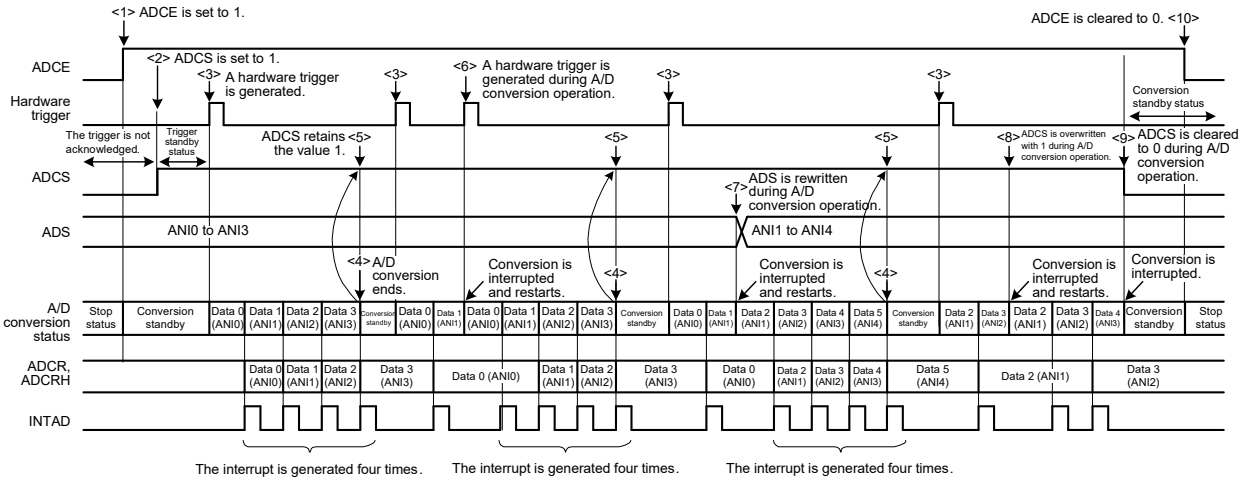
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<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 system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10 - 26 Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



10.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, 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 register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

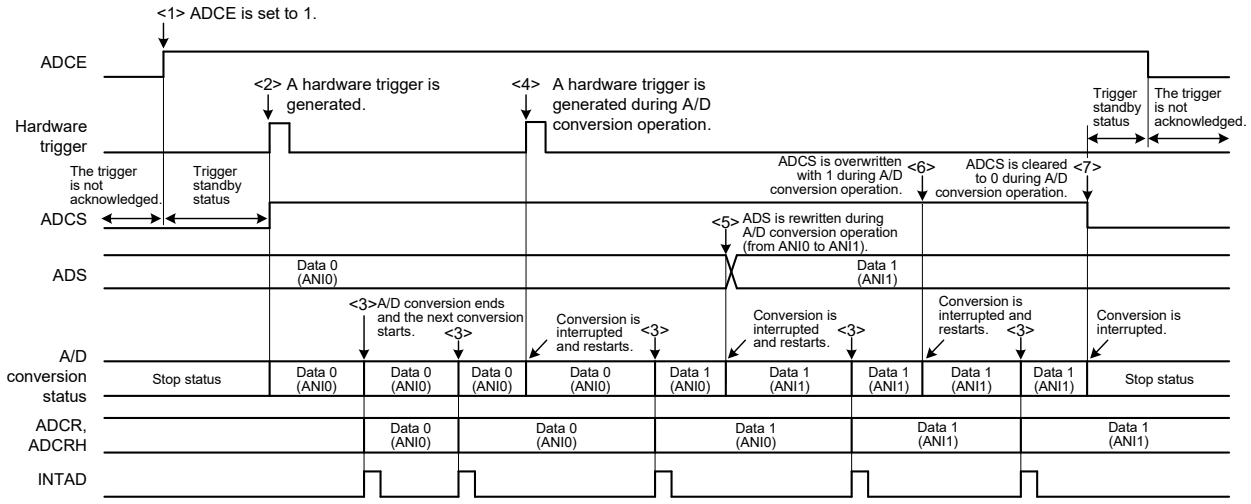
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10 - 27 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



10.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, 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 register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

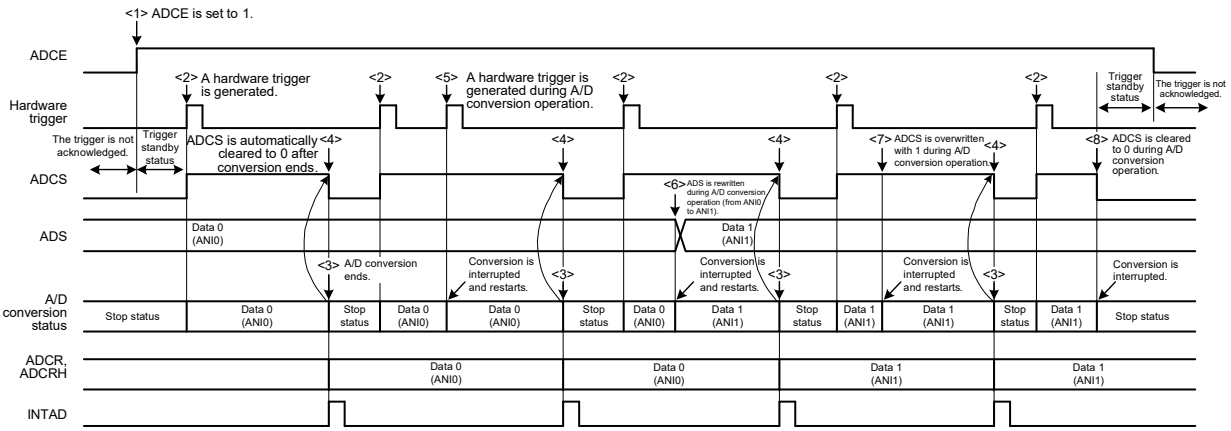
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10 - 28 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



10.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, 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 (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

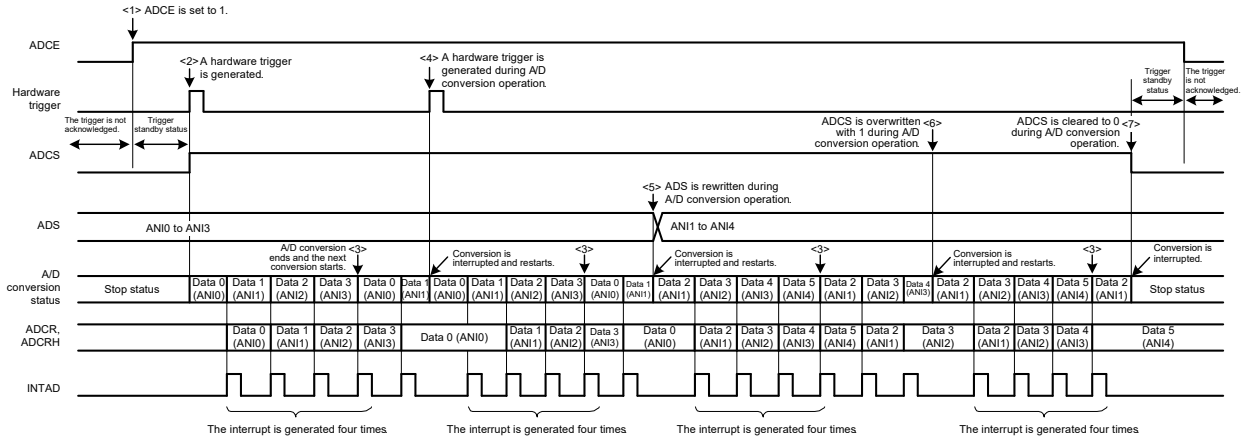
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<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 system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10 - 29 Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



10.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, 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 (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

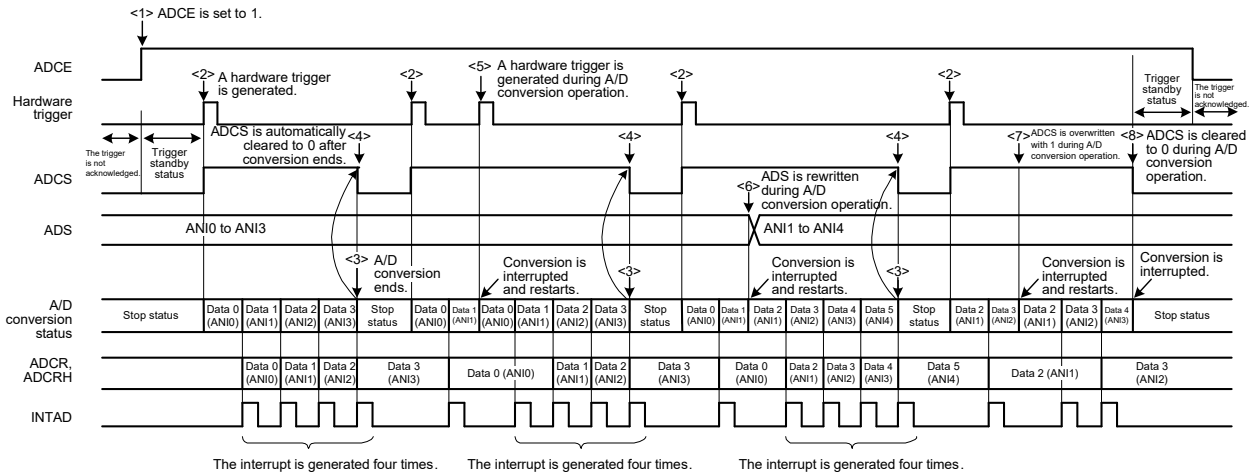
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<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 system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 10 - 30 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing

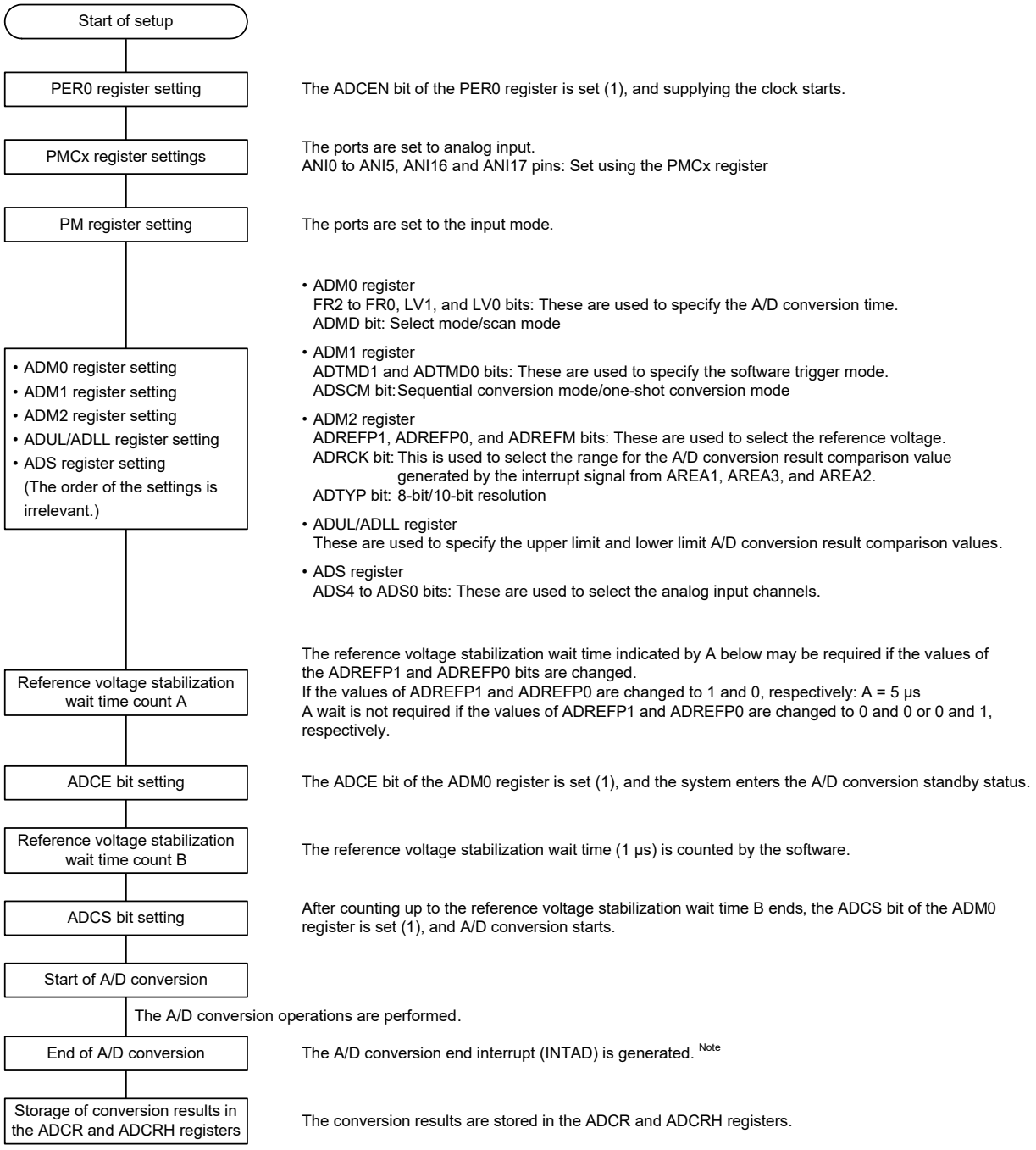


10.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

10.7.1 Setting up software trigger mode

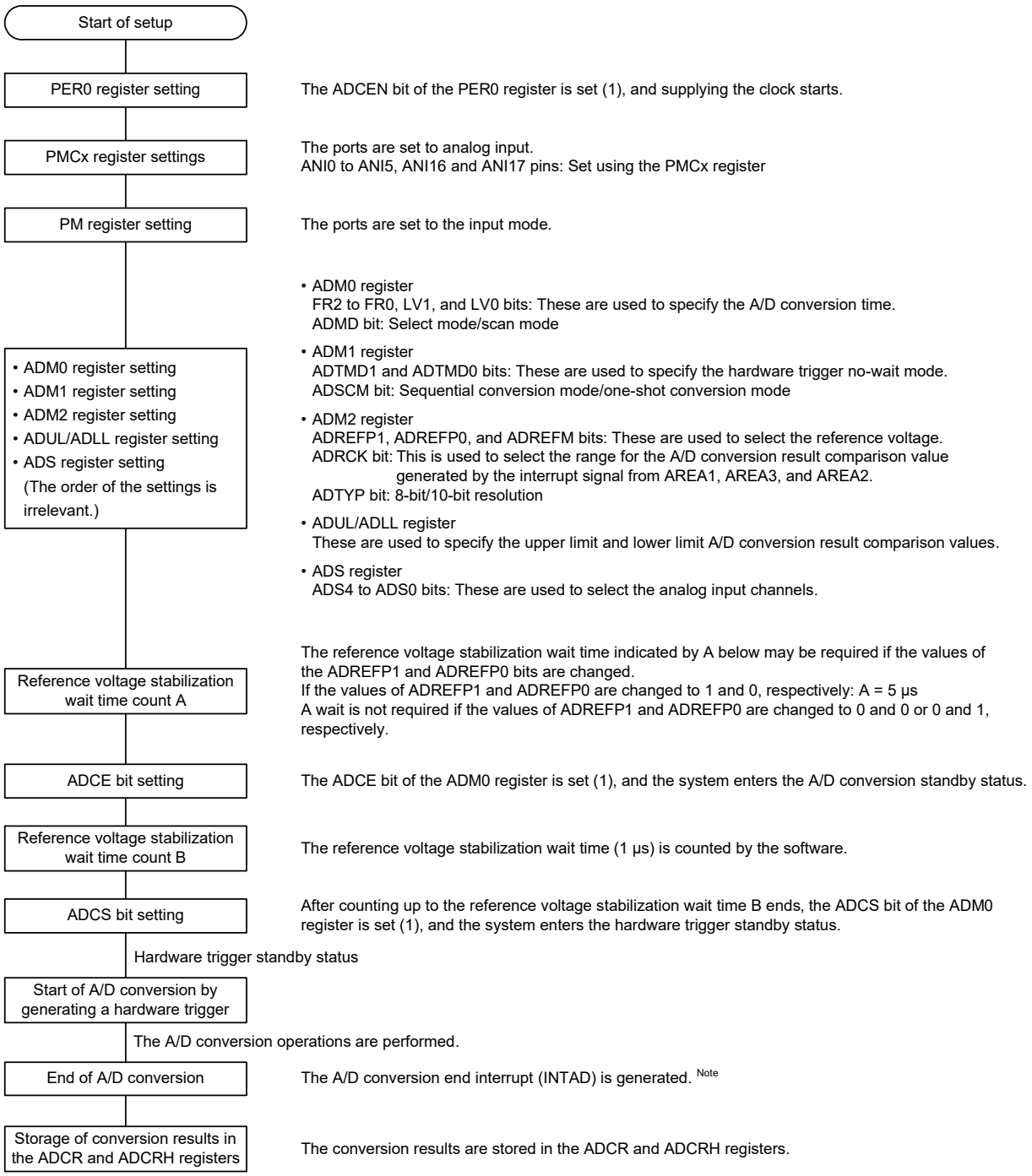
Figure 10 - 31 Setting up Software Trigger Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

10.7.2 Setting up hardware trigger no-wait mode

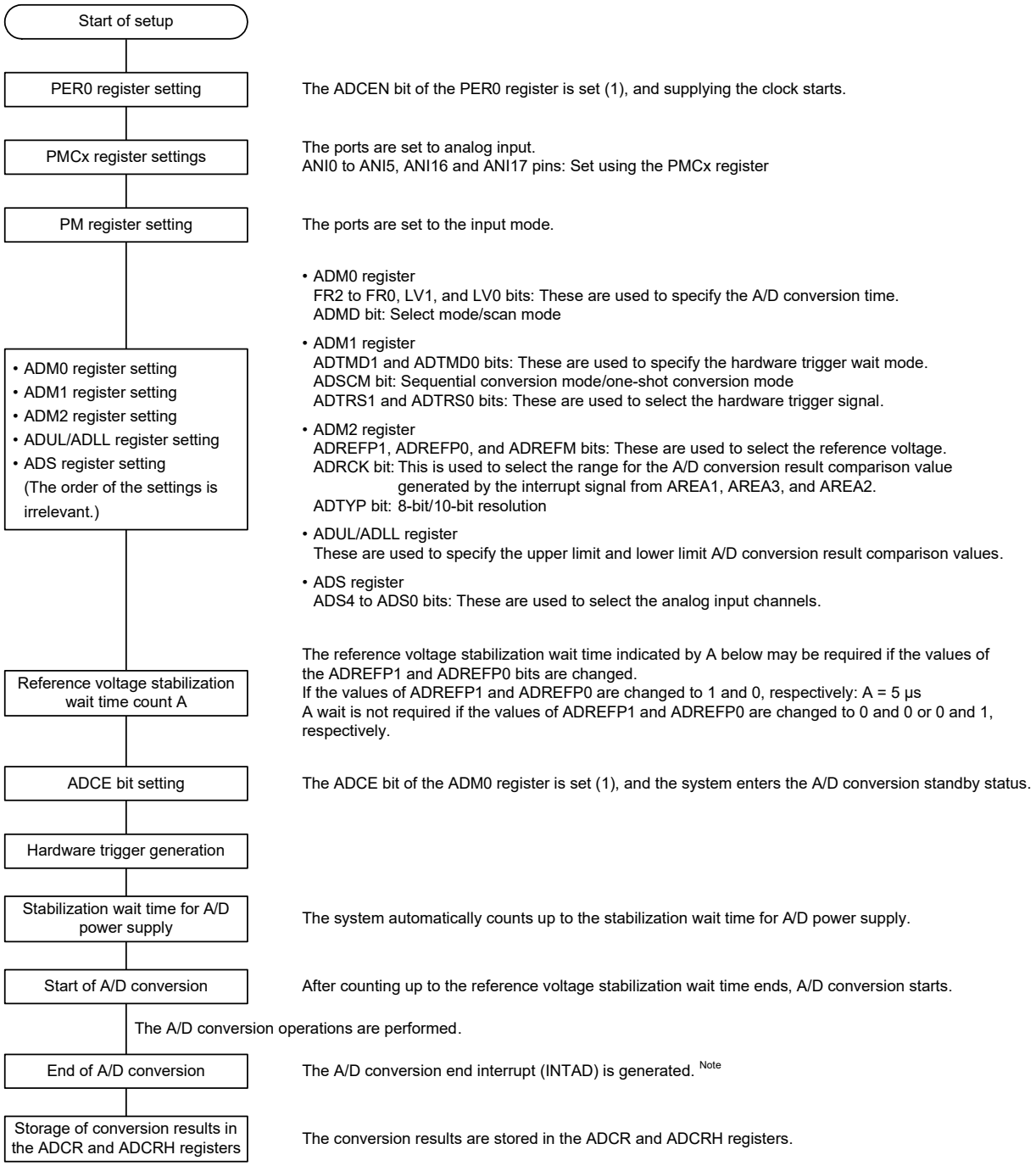
Figure 10 - 32 Setting up Hardware Trigger No-Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

10.7.3 Setting up hardware trigger wait mode

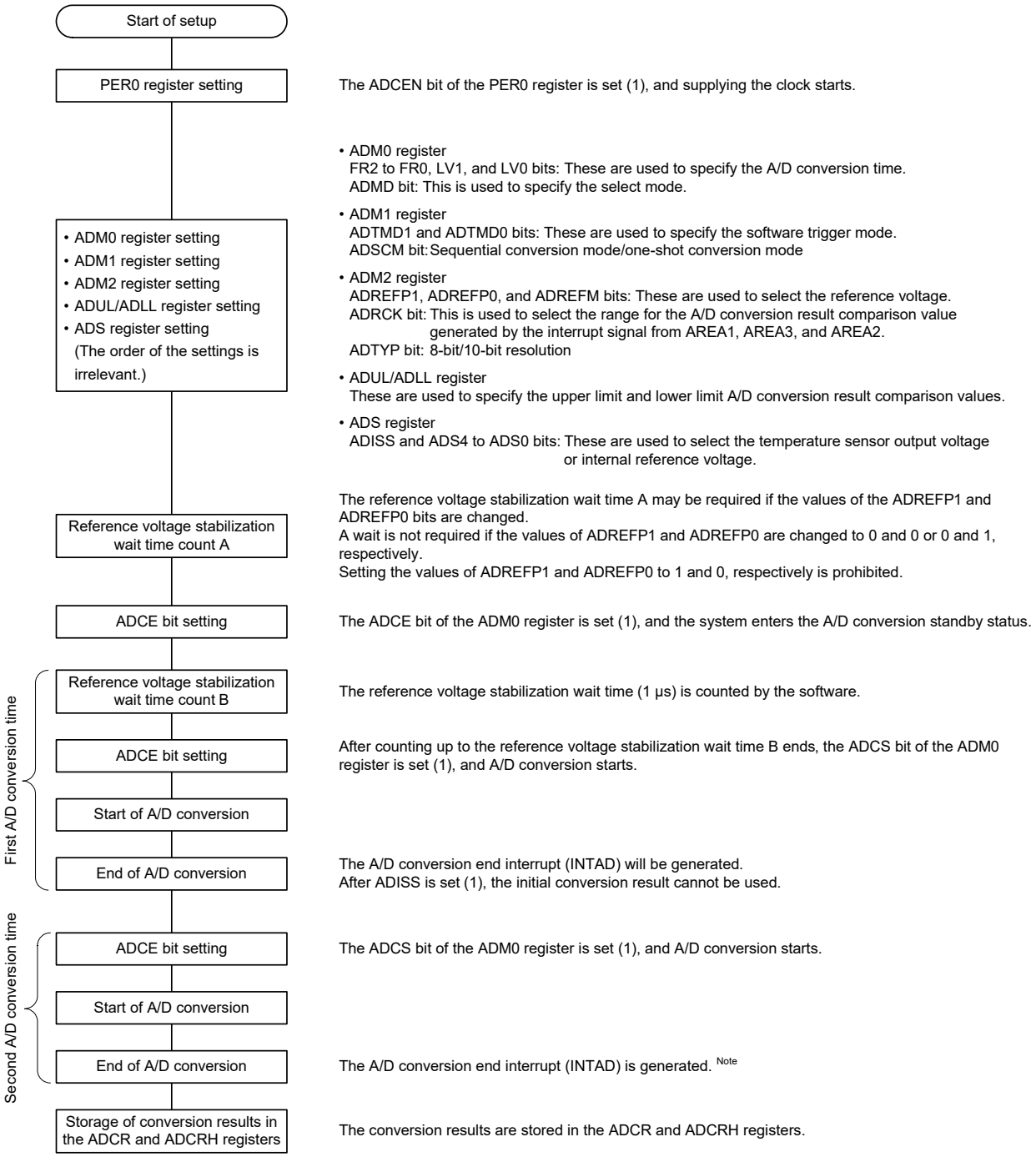
Figure 10 - 33 Setting up Hardware Trigger Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

10.7.4 Setup when using temperature sensor (example for software trigger mode and one-shot conversion mode)

Figure 10 - 34 Setup when temperature sensor output/internal reference voltage output is selected

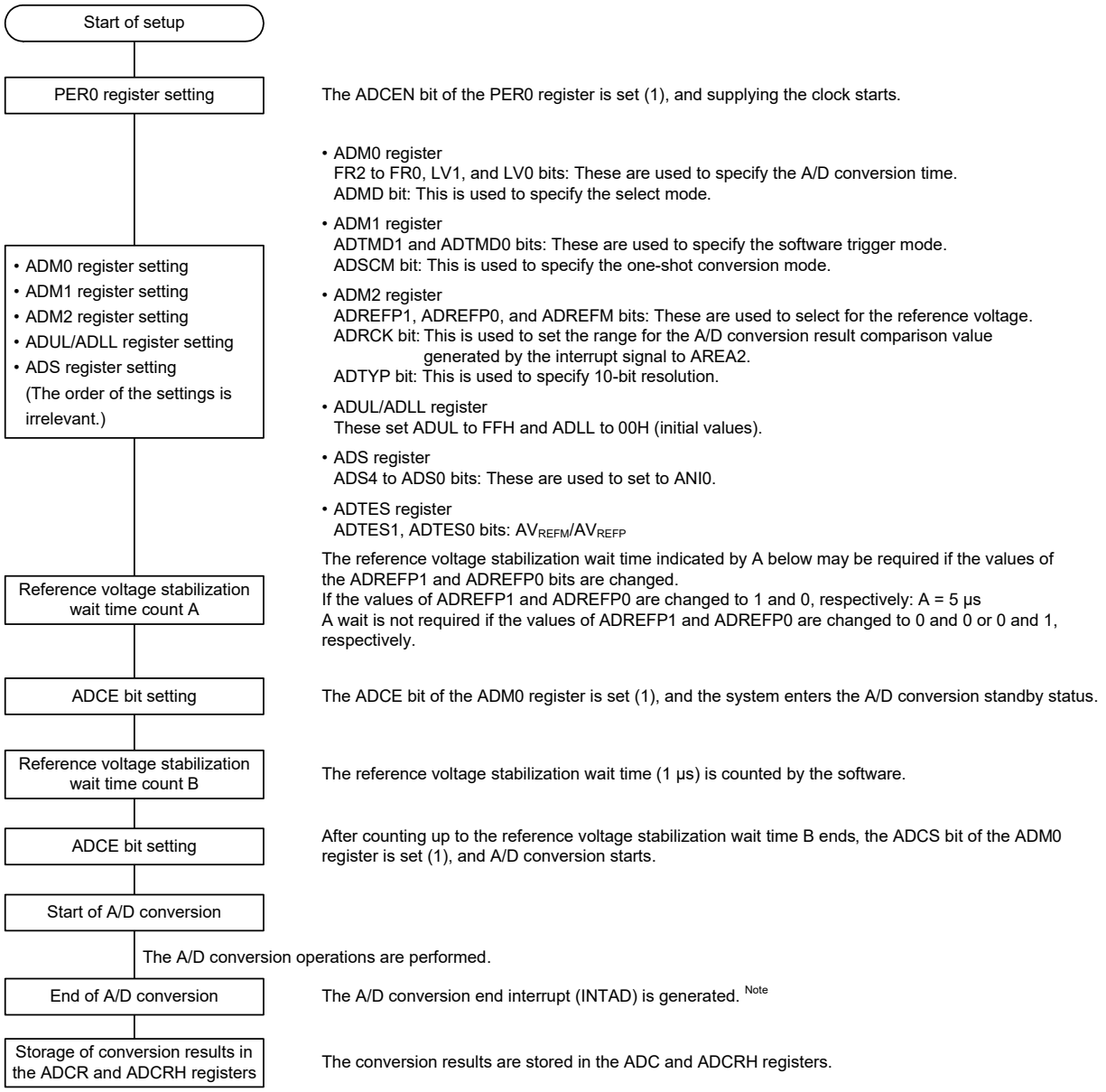


Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

Caution Operation is possible only in HS (high-speed main) mode.

10.7.5 Setting up test mode

Figure 10 - 35 Setting up Test Trigger Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

Caution For the procedure for testing the A/D converter, see 23.3.8 A/D test function.

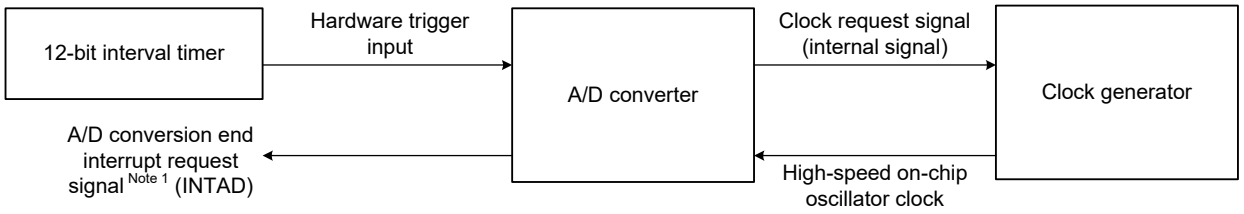
10.8 SNOOZE Mode Function

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode, A/D conversion can be performed without operating the CPU by inputting a hardware trigger. This is effective for reducing the operation current.
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the 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)

Caution That the SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 10 - 36 Block Diagram When Using SNOOZE Mode Function



When using the SNOOZE mode function, the initial setting of each register is specified before switching to the STOP mode (for details about these settings, see **10.7.3 Setting up hardware trigger wait mode** Note 2). Just before move to STOP mode, bit 2 (AWC) of A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.
If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the system 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 Note 1.

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.
Note 2. Be sure to set the ADM1 register to E2H or E3H.

Remark The hardware trigger is INTIT.
Specify the hardware trigger by using the A/D Converter Mode Register 1 (ADM1).

(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 and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

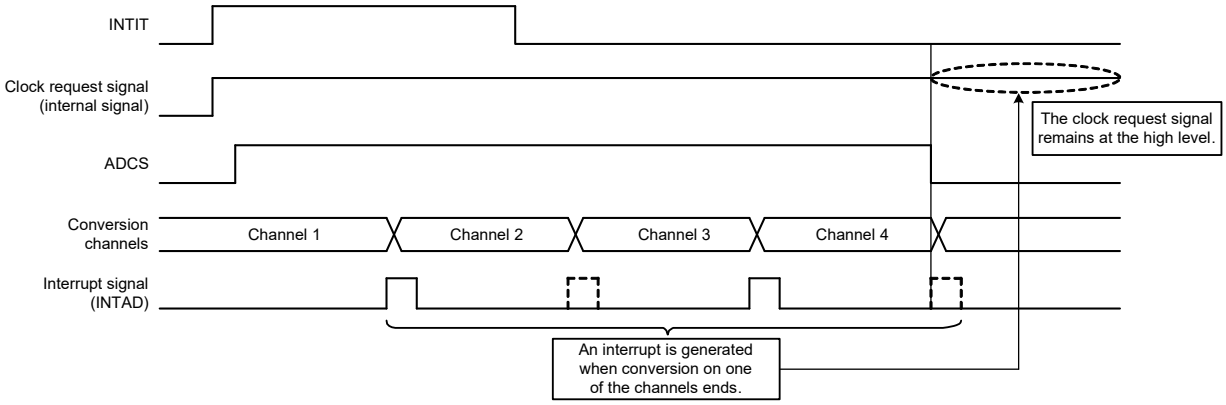
• While in the select mode

When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD) is generated, 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 A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the clock request signal remains at the high level, and 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 A/D converter mode register 2 (ADM2) to 0. 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 10 - 37 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 and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

• While in the select mode

If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

• While in the scan mode

If the A/D conversion end interrupt request signal (INTAD) is not generated even once during A/D conversion of the four channels, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

Figure 10 - 38 Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)

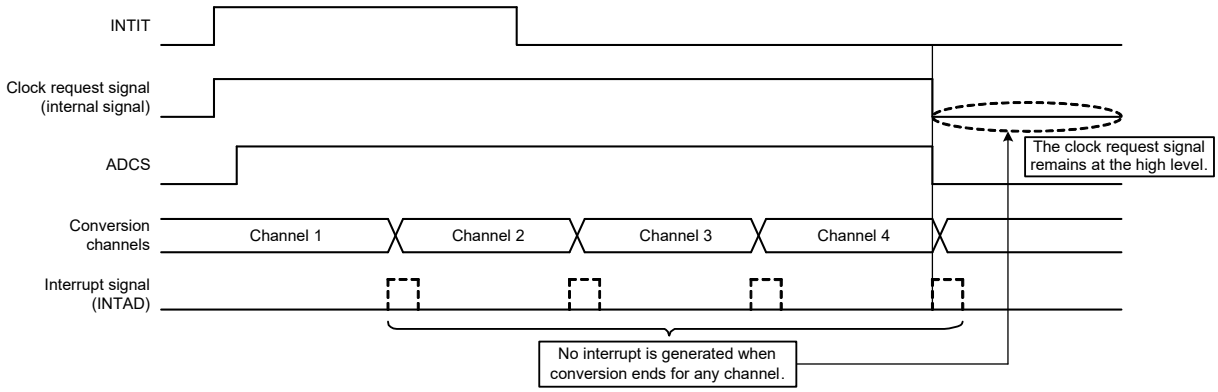
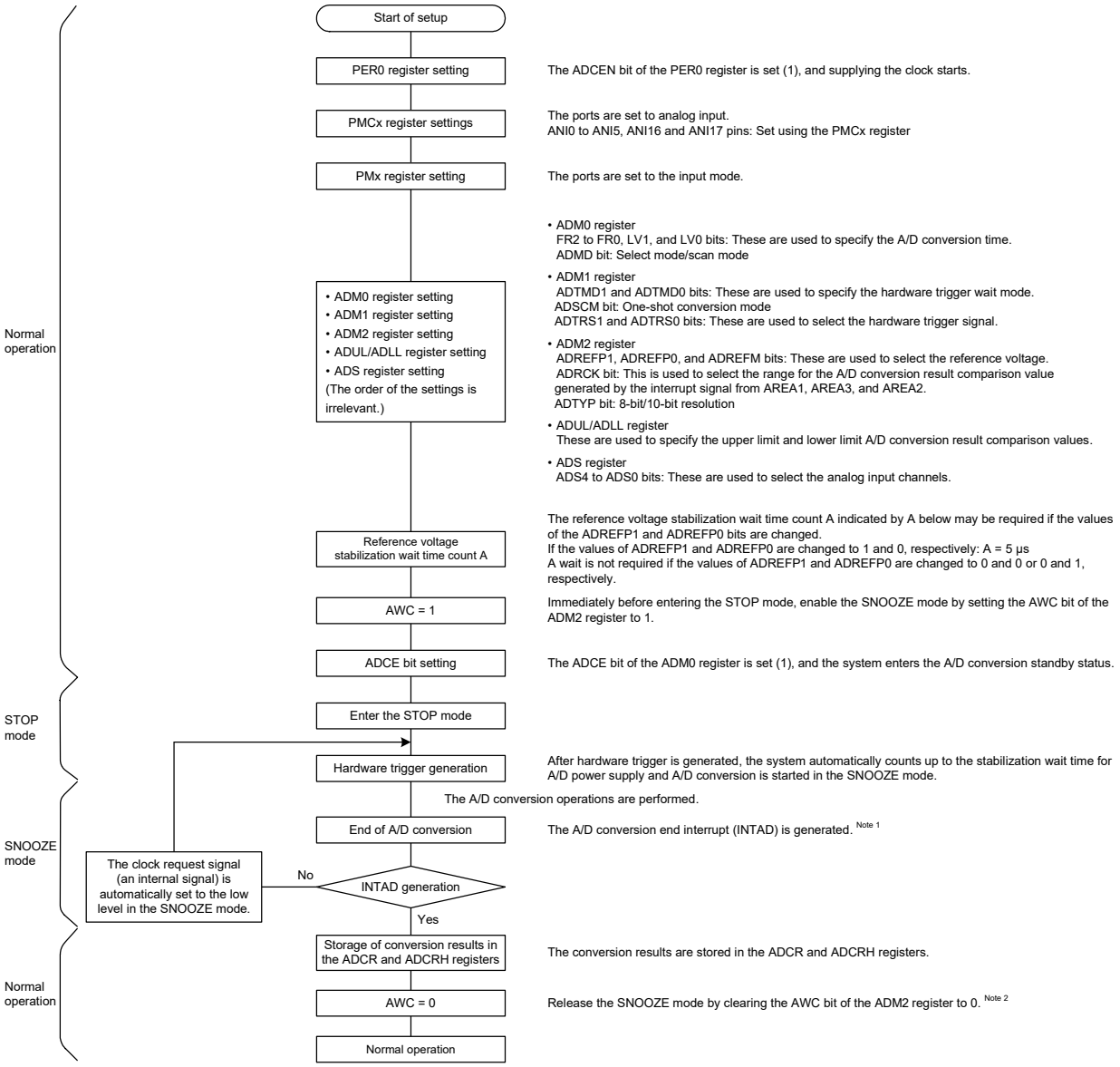


Figure 10 - 39 Flowchart for Setting up SNOOZE Mode



Note 1. If the A/D conversion end interrupt request signal (INTAD) is not generated by setting ADRCK bit and ADUL/ADLL register, the result is not stored in the ADCR and ADCRH registers.
The system enters the STOP 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 spite of the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

10.9 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 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

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, and differential linearity errors that are 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 10 - 40 Overall Error

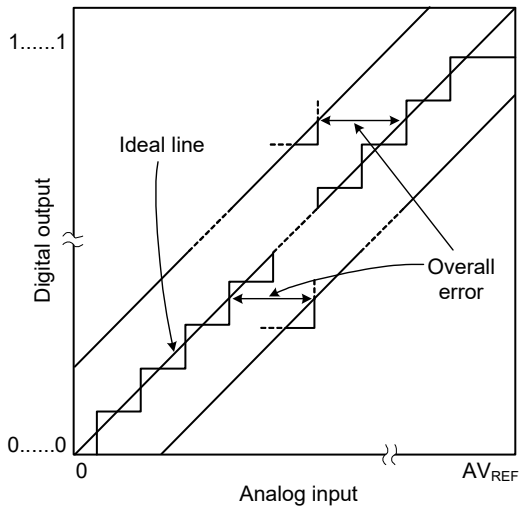
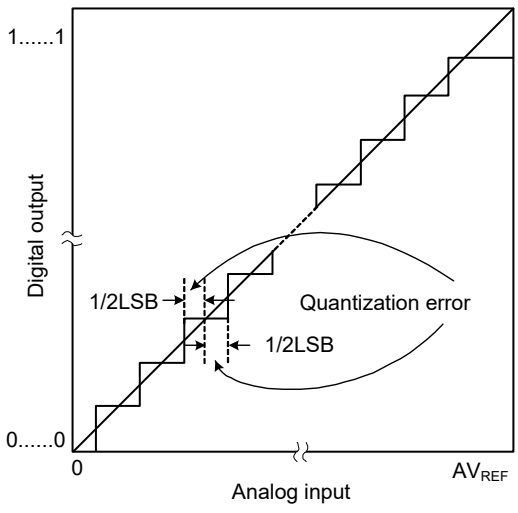


Figure 10 - 41 Quantization Error



(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.....000 to 0.....001.

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……001 to 0……010.

(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.....110 to 1.....111.

(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.

Figure 10 - 42 Zero-Scale Error

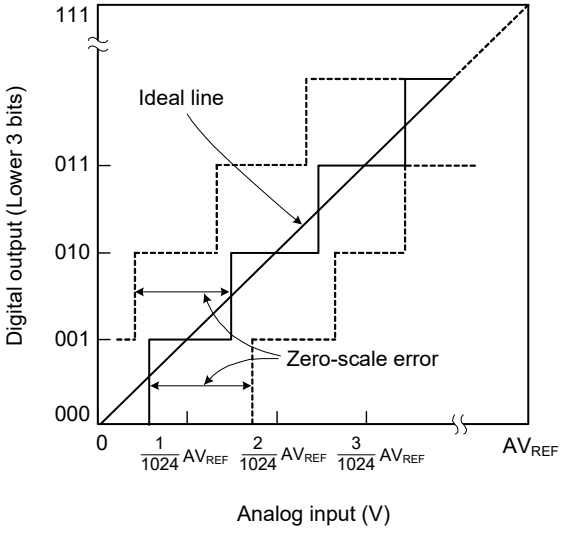


Figure 10 - 43 Full-Scale Error

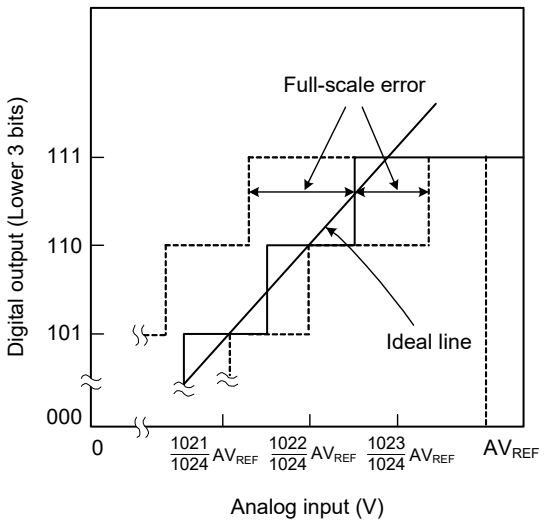


Figure 10 - 44 Integral Linearity Error

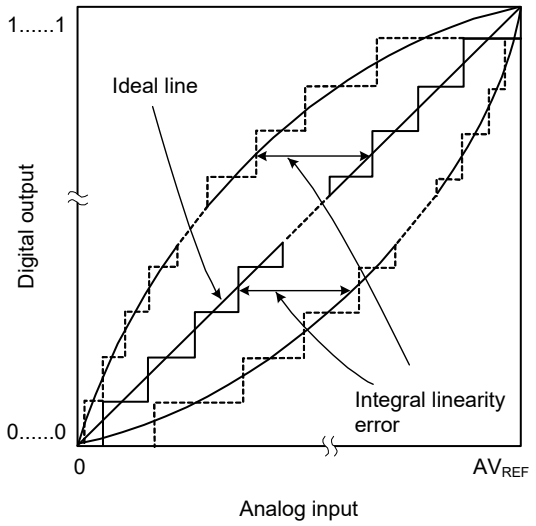
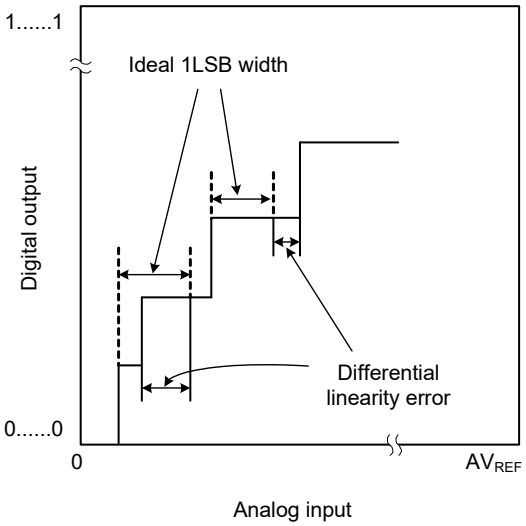


Figure 10 - 45 Differential Linearity Error



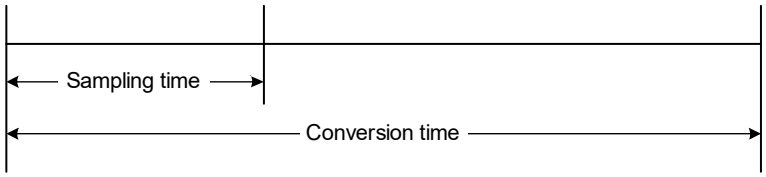
(8) Conversion time

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.



10.10 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP 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 status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

(2) Input range of ANI0 to ANI5, ANI16 and ANI17 pins

Observe the rated range of the ANI0 to ANI5, ANI16 and ANI17 pins input voltage. If a voltage exceeding V_{DD} and AV_{REFP} or below V_{SS} and AV_{REFM} or lower (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 (1.45 V) is selected as the reference voltage for the + side of the A/D converter, do not input voltage exceeding internal reference voltage (1.45 V) to a pin selected by the ADS register. However, it is no problem that a pin not selected by the ADS register is input voltage exceeding the internal reference voltage (1.45 V).

Caution The internal reference voltage (1.45 V) can be selected only in HS (high-speed main) mode.

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion

The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between the ADCR or ADCRH register write and the A/D converter mode register 0 (ADM0) write, the analog input channel specification register (ADS) write upon the end of conversion

The ADM0, ADS registers write has priority. The ADCR or ADCRH register write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REFP}, V_{DD}, ANI0 to ANI5, ANI16 and ANI17.

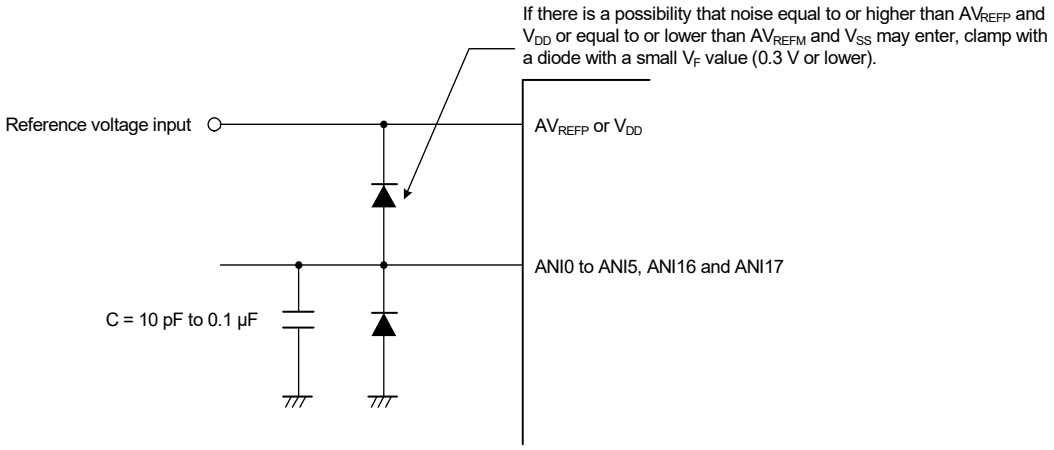
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external capacitor as shown in Figure 10 - 44 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 10 - 46 Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> The analog input pins (ANI0 to ANI5, ANI16 and ANI17) are also used as input port pins (P20 to P25, P00 and P01).
When A/D conversion is performed with any of the ANI0 to ANI5, ANI16 and ANI17 pins selected, do not access P20 to P25, P00 and P01 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

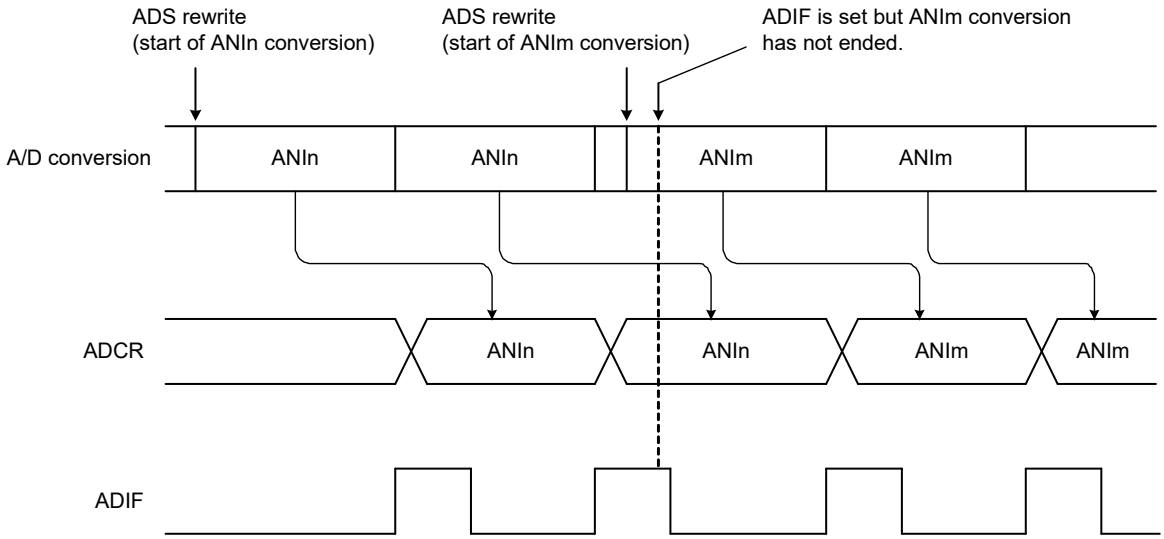
(6) Input impedance of analog input (ANIn) pins

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, and on the other states.
To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI5, ANI16 and ANI17 pins (see Figure 10 - 45).
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 conversion.

(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) 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 ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.
When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

Figure 10 - 47 Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger 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 after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS) and port mode control register (PMC), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, or PMC register. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 10 - 48 Internal Equivalent Circuit of ANIn Pin

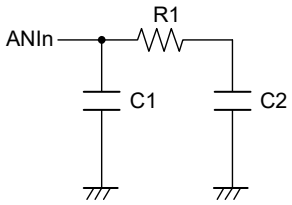


Table 10 - 5 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REFP}, V_{DD}	ANIn Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
3.6 V ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI5	14	8	2.5
	ANI16 and ANI17	18	8	7.0
2.7 V ≤ V_{DD} < 3.6 V	ANI0 to ANI5	39	8	2.5
	ANI16 and ANI17	53	8	7.0

Remark 1. The resistance and capacitance values shown in Table 10 - 5 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AV_{REFP} and V_{DD} voltages stabilize.

CHAPTER 11 SERIAL ARRAY UNIT

The serial array unit has two serial channels. All channels can achieve UART, 3-wire serial (CSI) and simplified I^2C. Function assignment of each channel supported by the R9A02G015 is as shown below.

• 32-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01

11.1 Functions of Serial Array Unit

Each serial interface supported by the R9A02G015 has the following features.

11.1.1 3-wire serial I/O (CSI00, CSI01)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel.
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 **11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01) Communication**.

[Data transmission/reception]
• Data length of 7 or 8 bits
• Phase control of transmit/receive data
• MSB/LSB first selectable
[Clock control]
• Master/slave selection
• Phase control of I/O clock
• Setting of transfer period by prescaler and internal counter of each channel
• Maximum transfer rate Note
  During master communication: Max. f_{CLK}/2 (CSI00 only)
  Max. f_{CLK}/4
  During slave communication: Max. f_{MCK}/6
[Interrupt function]
• Transfer end interrupt/buffer empty interrupt
[Error detection flag]
• Overrun error

CSIs of the following channels supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only the following CSIs can be specified.
• CSI00

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E).

11.1.2 UART (UART0)

This is a start-stop synchronization 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).
For details about the settings, see **11.6 Operation of UART (UART0) Communication**.

[Data transmission/reception]
• Data length of 7, 8, or 9 bits
• Select the MSB/LSB first
• Level setting of transmit/receive data and select of reverse
• Parity bit appending and parity check functions
• Stop bit appending
[Interrupt function]
• Transfer end interrupt/buffer empty interrupt
• Error interrupt in case of framing error, parity error, or overrun error
[Error detection flag]
• Framing error, parity error, or overrun error

In addition, UART reception supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. UART can be specified when the high-speed on-chip oscillator clock (f_{IH}) is selected for the CPU/peripheral hardware clock (f_{CLK}) in the SNOOZE mode.

11.1.3 Simplified I²C (IIC00, IIC01)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **11.7 Operation of Simplified I²C (IIC00, IIC01) Communication**.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function Note and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error or overrun error

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See the processing flow in **11.7.3 (2)** for details.

11.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 11 - 1 Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits or 9 bits
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn) Note
Serial clock I/O	SCK00, SCK01 pins (for 3-wire serial I/O) and SCL00, SCL01 pins (for simplified I^2C)
Serial data input	SI00, SI01 pins (for 3-wire serial I/O), RxD0 pin (for UART)
Serial data output	SO00, SO01 pins (for 3-wire serial I/O), TxD0 pin (for UART)
Serial data I/O	SDA00, SDA01 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Peripheral reset control register 0 (PRR0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Serial standby control register m (SSCm) • Noise filter enable register 0 (NFEN0)
	<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)
	• Port input mode register 5 (PIM5) • Port output mode register 5 (POM5) • Port mode register 5 (PM5) • Port register 5 (P5)

(**Note** and **Remark** are listed on the next page.)

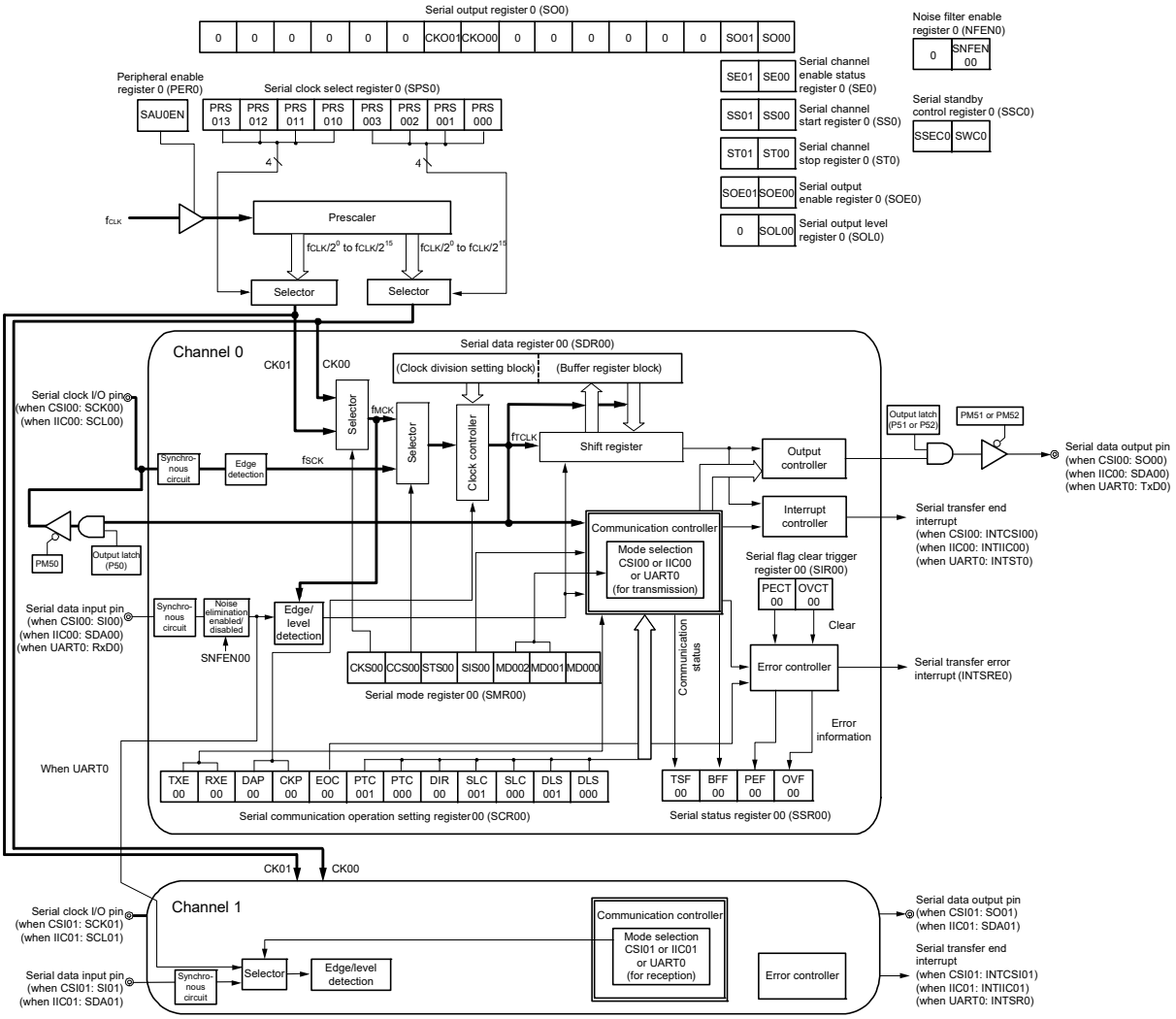
Note The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFRs, depending on the communication mode.

- CSIp communication ..... SIOp (CSIp data register)
- UARTq reception ..... RXDq (UARTq receive data register)
- UARTq transmission ..... TXDq (UARTq transmit data register)
- IICr communication ..... SIOr (IICr data register)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00, 01)

Figure 11 - 1 shows the Block Diagram of Serial Array Unit 0.

Figure 11 - 1 Block Diagram of Serial Array Unit 0



11.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa.

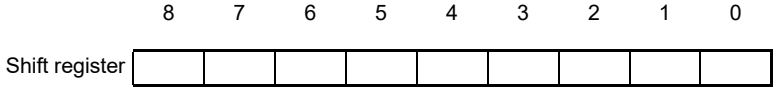
In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used Note.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8/9 bits of serial data register mn (SDRmn).



Note Only the following UARTs can be specified for the 9-bit data length.
• UART0

11.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) Note or bits 7 to 0 (lower 8 bits) function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 8/9 bits. When data is to be transmitted, set transmit data to be transferred to the shift register to the lower 8/9 bits.

The data stored in the lower 8/9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

• 7-bit data length (stored in bits 0 to 6 of SDRmn register)
• 8-bit data length (stored in bits 0 to 7 of SDRmn register)
• 9-bit data length (stored in bits 0 to 8 of SDRmn register)

The SDRmn register can be read or written in 16-bit units.

The lower 8/9 bits of the SDRmn register can be read or written Note as the following SFRs, depending on the communication mode.

• CSIp communication..... SIOp (CSIp data register)
• UARTq reception ..... RXDq (UARTq receive data register)
• UARTq transmission..... TXDq (UARTq transmit data register)
• IICr communication..... SIOr (IICr data register)

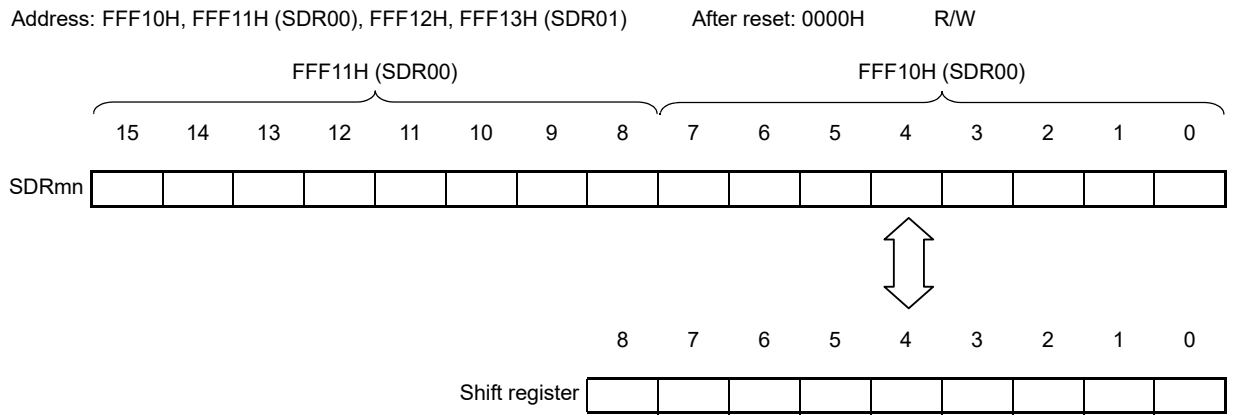
Reset signal generation clears the SDRmn register to 0000H.

Note Using an 8-bit memory manipulation instruction to write to the SDRmn[7:0] bits while operation is stopped (SEmn = 0) is prohibited (doing so clears the SDRmn[15:9] bits to 0).

Remark 1. After data is received, 0 is stored in bits 0 to 8 in bit portions that exceed the data length.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01),
q: UART number (q = 0), r: IIC number (r = 00, 01)

Figure 11 - 2 Format of Serial data register mn (SDRmn) (mn = 00, 01)



Remark For the function of the higher 7 bits of the SDRmn register, see **11.3 Registers Controlling Serial Array Unit**.

11.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Serial standby control register m (SSCm)
- Noise filter enable register 0 (NFEN0)
- Port input mode register 5 (PIM5)
- Port output mode register 5 (POM5)
- Port mode register 5 (PM5)
- Port register 5 (P5)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

11.3.1 Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PER0 register to 00H.

Figure 11 - 3 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	IICA2EN Note	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

SAUmEN	Control of serial array unit m input clock supply
0	Stops supply of input clock. • SFR used by serial array unit m cannot be written.
1	Enables input clock supply. • SFR used by serial array unit m can be read/written.

Note This bit is incorporated with R9A02G0151, but is not incorporated with R9A02G0150.

Caution 1. When setting serial array unit m, be sure to first set the following registers with the SAUmEN bit set to 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored (except for the noise filter enable register 0 (NFEN0), port input mode register 5 (PIM5), port output mode register 5 (POM5), port mode register 5 (PM5), and port register 5 (P5).

- **Serial clock select register m (SPSm)**
- **Serial mode register mn (SMRmn)**
- **Serial communication operation setting register mn (SCRmn)**
- **Serial data register mn (SDRmn)**
- **Serial flag clear trigger register mn (SIRmn)**
- **Serial status register mn (SSRmn)**
- **Serial channel start register m (SSm)**
- **Serial channel stop register m (STm)**
- **Serial channel enable status register m (SEm)**
- **Serial output enable register m (SOEm)**
- **Serial output level register m (SOLm)**
- **Serial output register m (SOm)**
- **Serial standby control register m (SSCm)**

Caution 2. Be sure to clear the following bits to 0.
Bits 1 and 3

11.3.2 Peripheral reset control register 0 (PRR0)

This register is used for individual reset control of each peripheral hardware.

The R9A02G015 controls reset and reset release of each peripheral hardware supported by the PRR0 register.

To reset the serial array unit, be sure to set bit 2 (SAU0RES) to 1.

The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PRR0 register to 00H.

Figure 11 - 4 Format of Peripheral reset control register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PRR0	IICA2RES[Note]	IICA1RES	ADCRES	IICA0RES	0	SAU0RES	0	TAU0RES

SAU0RES	Reset control of serial array unit
0	Serial array unit reset release
1	Serial array unit reset state

Note This bit is incorporated with R9A02G0151, but is not incorporated with R9A02G0150.

11.3.3 Serial clock select register m (SPSm)

The SPSm register 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 bits 7 to 4 of the SPSm register, and CKm0 is selected by bits 3 to 0.

Rewriting the SPSm register is prohibited when the register is in operation (when SEmn = 1).

The SPSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SPSm register can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears the SPSm register to 0000H.

Figure 11 - 5 Format of Serial clock select register m (SPSm)

Address: F0126H, F0127H (SPS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0		Section of operation clock (CKmk) Note				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Caution Be sure to clear bits 15 to 8 to 0.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency
Remark 2. m: Unit number (m = 0)
Remark 3. k = 0, 1

11.3.4 Serial mode register mn (SMRmn)

The SMRmn register is a register that sets 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, an operation mode (CSI, 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 SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

The SMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SMRmn register to 0020H.

Figure 11 - 6 Format of Serial mode register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0112H, F0113H (SMR01), After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn Note	0	SISmn0 Note	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	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
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCSmn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in CSI mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STSmn Note	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^2C).
1	Valid edge of the RxDq pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Note The SMR01 register only.

Caution **Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00 register) to 0. Be sure to set bit 5 to 1.**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00, 01)

Figure 11 - 7 Format of Serial mode register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0112H, F0113H (SMR01) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn Note	0	SIS mn0 Note	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0 Note	Controls inversion of level of receive data of channel n 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.

MD mn2	MD mn1	Setting of operation mode of channel n
0	0	CSI mode
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note The SMR01 register only.

Caution **Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00 register) to 0. Be sure to set bit 5 to 1.**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00, 01)

11.3.5 Serial communication operation setting register mn (SCRmn)

The SCRmn register is a communication operation setting register of channel n. It is used to set a data transmission/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 SCRmn register is prohibited when the register is in operation (when SEmn = 1).

The SCRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SCRmn register to 0087H.

Figure 11 - 8 Format of Serial communication operation setting register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011AH, F011BH (SCR01) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 Note 1	SLC mn0	0	1	DLSm n1	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in CSI mode	Type
0	0	SCKp SOp D7 D6 D5 D4 D3 D2 D1 D0 SIp input timing	1
0	1	SCKp SOp D7 D6 D5 D4 D3 D2 D1 D0 SIp input timing	2
1	0	SCKp SOp D7 D6 D5 D4 D3 D2 D1 D0 SIp input timing	3
1	1	SCKp SOp D7 D6 D5 D4 D3 D2 D1 D0 SIp input timing	4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I²C mode.			

EOC mn	Mask control of error interrupt signal (INTSREx (x = 0))
0	Disables generation of error interrupt INTSREx (INTSRx is generated).
1	Enables generation of error interrupt INTSREx (INTSRx is not generated if an error occurs).
Set EOCmn = 0 in the CSI mode, simplified I²C mode, and during UART transmission Note 2.	

Note 1. The SCR00 register only.

Note 2. When using CSImn not with EOCmn = 0, error interrupt INTSREn may be generated.

Caution **Be sure to clear bits 3, 6, and 11 to 0 (Also clear bit 5 of the SCR01 register to 0). Be sure to set bit 2 to 1.**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 11 - 9 Format of Serial communication operation setting register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) to F011AH, F011BH (SCR01) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCmn1 Note 1	SLC mn0	0	1	DLSmn1	DLS mn0

PTC mn1	PTC mn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity Note 2.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the CSI mode and simplified I^2C mode.			

DIR mn	Selection of data transfer sequence in CSI and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I^2C mode.	

SLCmn1 Note 1	SLC mn0	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 (mn = 00 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) or 2 bits (SLCmn1, SLCmn0 = 1, 0) during UART transmission.		

DLSmn1	DLS mn0	Setting of data length in CSI and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDRmn register) (settable in UART mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited
Be sure to set DLSmn1, DLSmn0 = 1, 1 in the simplified I^2C mode.		

Note 1. The SCR00 register only.

Note 2. 0 is always added regardless of the data contents.

Caution **Be sure to clear bits 3, 6, and 11 to 0 (Also clear bit 5 of the SCR01 register to 0). Be sure to set bit 2 to 1.**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

11.3.6 Serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n.

Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 as a transmit/receive buffer register, and bits 15 to 9 (higher 7 bits) are used as a register that sets the division ratio of the operation clock (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of the SDRmn register is used as the transfer clock.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00 and SDR01 to 0000000B. The input clock f_{SCK} (slave transfer in CSI mode) from the SCKp pin is used as the transfer clock.

The lower 8/9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.

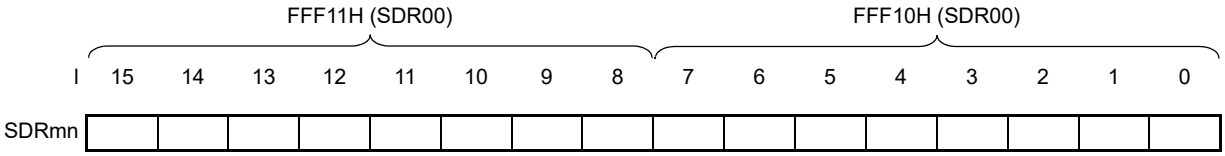
The SDRmn register can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8/9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0.

Reset signal generation clears the SDRmn register to 0000H.

Figure 11 - 10 Format of Serial data register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W



SDRmn[15:9]							Transfer clock set by dividing the operating clock
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

Caution 1. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

Caution 2. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.

Caution 3. Using an 8-bit memory manipulation instruction to write to the SDRmn[7:0] bits while operation is stopped (SEmn = 0) is prohibited (doing so clears the SDRmn[15:9] bits to 0).

Remark 1. For the function of the lower 8/9 bits of the SDRmn register, see **11.2 Configuration of Serial Array Unit**.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

11.3.7 Serial flag clear trigger register mn (SIRmn)

The SIRmn register is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.

The SIRmn register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears the SIRmn register to 0000H.

Figure 11 - 11 Format of Serial flag clear trigger register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010AH, F010BH (SIR01) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn Note	PECTmn	OVCTmn

FECTmn Note	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Note The SIR01 register only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00 register) to 0.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

Remark 2. When the SIRmn register is read, 0000H is always read.

11.3.8 Serial status register mn (SSRmn)

The SSRmn register is a register that indicates the communication status and error occurrence status of channel n. The errors indicated by this register are a framing error, parity error, and overrun error.

The SSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be set with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears the SSRmn register to 0000H.

Figure 11 - 12 Format of Serial status register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0102H, F0103H (SSR01) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn Note	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> • 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 STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TXEmn 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 RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Note The SSR01 register only.

Caution **When the CSI is performing reception operations in the SNOOZE mode (SWCm = 1), the BFFmn flag will not change.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 11 - 13 Format of Serial status register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0102H, F0103H (SSR01) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn Note	PEF mn	OVF mn

FEF mn Note	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear condition> • 1 is written to the FECTmn bit of the SIRmn register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEF mn	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I^2C transmission).
<Clear condition> • 1 is written to the PECTmn bit of the SIRmn register. <Set 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 channel at the ACK reception timing during I^2C transmission (ACK is not detected).	

OVF mn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> • 1 is written to the OVCTmn bit of the SIRmn register. <Set 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 RXEmn 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 CSI mode.	

Note The SSR01 register only.

Caution 1. If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVEmn = 1) is detected.

Caution 2. When the CSI is performing reception operations in the SNOOZE mode (SWCm = 1), the OVFmn flag will not change.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

11.3.9 Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

The SSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSm register can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

Figure 11 - 14 Format of Serial channel start register m (SSm)

Address: F0122H, F0123H (SS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS01	SS00

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status Note.

Note If set the SSmn = 1 to during a communication operation, will wait status to stop the communication. At this time, holding status value of control register and shift register, SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution 1. Be sure to clear bits 15 to 2 of the SS0 register to 0.

Caution 2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

Remark 2. When the SSm register is read, 0000H is always read.

11.3.10 Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

The STm register can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of the STm register can be set with a 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears the STm register to 0000H.

Figure 11 - 15 Format of Serial channel stop register m (STm)

Address: F0124H, F0125H (ST0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST01	ST00

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation Note.

Note Holding status value of the control register and shift register, the SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution Be sure to clear bits 15 to 2 of the ST0 register to 0.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

Remark 2. When the STm register is read, 0000H is always read.

11.3.11 Serial channel enable status register m (SEm)

The SEm register indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1.

When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

Channel n that stops operation can set the value of the CKOmn bit of the SOm register by software and output its value from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears the SEm register to 0000H.

Figure 11 - 16 Format of Serial channel enable status register m (SEm)

Address: F0120H, F0121H (SE0) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE01	SE00

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

11.3.12 Serial output enable register m (SOEm)

The SOEm register is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel n that enables serial output cannot rewrite by software the value of the SOmn bit of serial output register m (SOm) to be described below, 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 SOmn bit value of the SOm register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and stop condition can be created by software.

The SOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears the SOEm register to 0000H.

Figure 11 - 17 Format of Serial output enable register m (SOEm)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE01	SOE00

SOEmn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 15 to 2 of the SOE0 register, bits 15 to 2 to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

11.3.13 Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.

To use a pin for the serial interface as a port function pin other than a serial interface function pin, set the corresponding the CKOmn and SOmn bits to 1.

The SOm register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears the SOm register to 0303H.

Figure 11 - 18 Format of Serial output register m (SOm)

Address: F0128H, F0129H After reset: 0303H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	CKO 01	CKO 00	0	0	0	0	0	0	SO 01	SO 00

CKO mn	Serial clock output of channel n
0	Serial clock output value is 0.
1	Serial clock output value is 1.

SO mn	Serial data output of channel n
0	Serial data output value is 0.
1	Serial data output value is 1.

Caution **Be sure to clear bits 15 to 10 and 7 to 2 of the SO0 register to 0.**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

11.3.14 Serial output level register m (SOLm)

The SOLm register is a register that is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0 for corresponding bit in the CSI mode and simplifies I2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEmn = 1).

The SOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears the SOLm register to 0000H.

Figure 11 - 19 Format of Serial output level register m (SOLm)

Address: F0134H, F0135H (SOL0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL00

SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

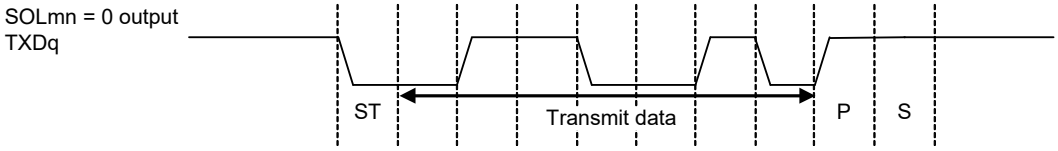
Caution Be sure to clear bits 15 to 1 of the SOL0 register to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0)

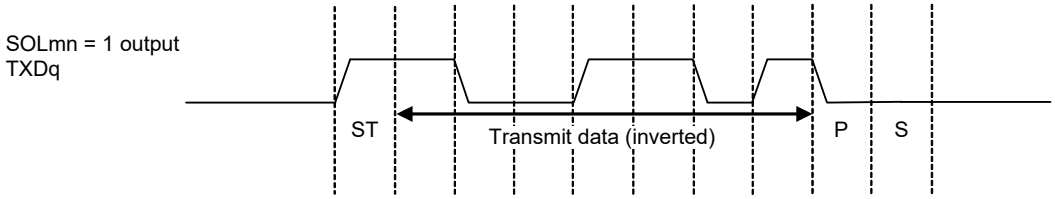
Figure 11 - 20 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 11 - 20 Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0), n: Channel number (n = 0)

11.3.15 Serial standby control register m (SSCm)

The SSC0 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data.

The SSCm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSCm register can be set with an 8-bit memory manipulation instruction with SSCmL.

Reset signal generation clears the SSCm register to 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- **When using CSI00: Up to 1 Mbps**
- **When using UART0: 4800 bps only**

Figure 11 - 21 Format of Serial standby control register m (SSCm)

Address: F0138H (SSC0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSECm	SWCm

SSECm	Selection of whether to enable or disable the generation of communication error interrupts in the SNOOZE mode
0	Enable the generation of error interrupts (INTSRE0).
1	Disable the generation of error interrupts (INTSRE0).

- The SSECm bit can be set to 1 or 0 only when both the SWCm and EOCmn bits are set to 1 during UART reception in the SNOOZE mode. In other cases, clear the SSECm bit to 0.
- Setting SSECm, SWCm = 1, 0 is prohibited.

SWCm	Setting of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

- When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion 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 is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Even when using SNOOZE mode, be sure to set the SWCm bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode.
  Also, be sure to change the SWCm bit to 0 after returning from STOP mode to normal operation mode.

Caution Setting SSECm, SWCm = 1, 0 is prohibited.

Figure 11 - 22 Interrupt in UART Reception Operation in SNOOZE Mode

EOCmn Bit	SSECm Bit	Reception Ended Successfully	Reception Ended in an Error
0	0	INTSRx is generated.	INTSRx is generated.
0	1	INTSRx is generated.	INTSRx is generated.
1	0	INTSRx is generated.	INTSREx is generated.
1	1	INTSRx is generated.	No interrupt is generated.

11.3.16 Noise filter enable register 0 (NFEN0)

The NFEN0 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 CSI 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 OFF, only synchronization is performed with the Operation clock of target channel (f_{MCK}).

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 register to 00H.

Figure 11 - 23 Format of Noise filter enable register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	0	0	SNFEN00

SNFEN00	Use of noise filter of RxD0 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear the SNFEN00 bit to 0 to use the other than RxD0 pin.	

Caution Be sure to clear bits 7 to 1 to 0.

11.3.17 Registers controlling port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), port register (Pxx), port input mode register (PIMxx), and port output mode register (POMxx)).

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode registers (PIMxx)**, and **4.3.5 Port output mode registers (POMxx)**.

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P52/INTP8/TI01/TO01/SO00/TXD0/TOOLTXD/UVBUSEN1) for serial data or serial clock output, requires setting the corresponding bits in the port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) to 1.

When using the port pin in N-ch open-drain output (V_{DD} tolerance) mode, set the corresponding bit in the port output mode register (POMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V, or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Example When P52/INTP8/TI01/TO01/SO00/TXD0/TOOLTXD/UVBUSEN1 is to be used for serial data output
Set the PM52 bit of port mode register 5 to 0.
Set the P52 bit of port register 5 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P51/INTP7/TI00/TO00/SDA00/SI00/RXD0/TOOLRXD) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PMxx) to 1. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

When the TTL input buffer is selected, set the corresponding bit in the port input mode register (PIMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Example When P51/INTP7/TI00/TO00/SDA00/SI00/RXD0/TOOLRXD is to be used for serial data input
Set the PM51 bit of port mode register 5 to 1.
Set the P51 bit of port register 5 to 0 or 1.

11.4 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.

11.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0.

Figure 11 - 24 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0)... Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	IICA2EN Note	IICA1EN	ADCEN	IICA0EN		SAU0EN		TAU0EN
	×	×	×	×	0	0/1	0	×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Note This bit is incorporated with R9A02G0151, but is not incorporated with R9A02G0150.

Caution 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored. Note that this does not apply to the following registers.
- **Noise filter enable register 0 (NFEN0)**
- **Port input mode register 5 (PIM5)**
- **Port output mode register 5 (POM5)**
- **Port mode register 5 (PM5)**
- **Port register 5 (P5)**

Caution 2. Be sure to clear the following bits to 0.
Bits 1 and 3

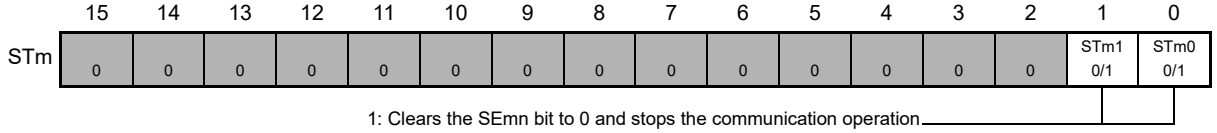
Remark ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

11.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

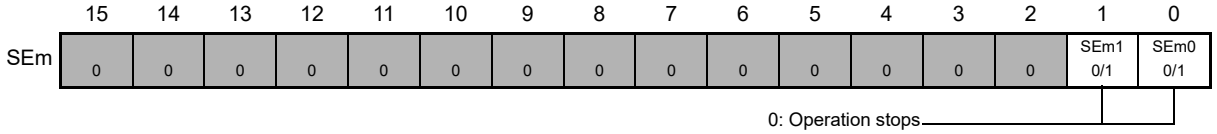
Figure 11 - 25 Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm)... This register is a trigger register that is used to enable stopping communication/count by each channel.



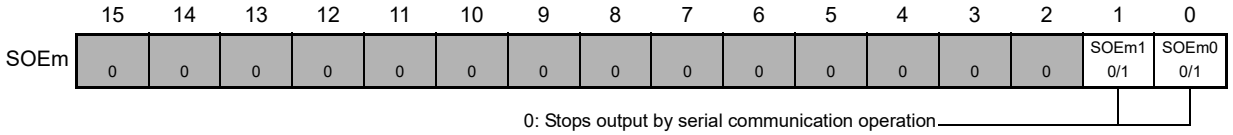
* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm)... This register indicates whether data transmission/ reception operation of each channel is enabled or stopped.



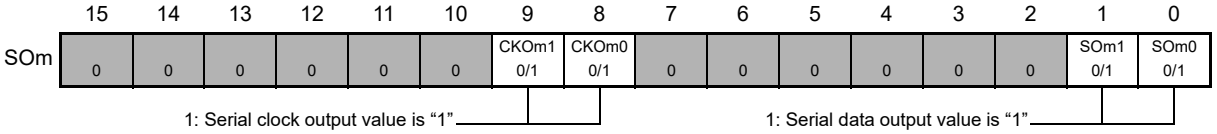
* The SEm register is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm)... This register is a register that is used to enable or stop output of the serial communication operation of each channel.



* For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

(d) Serial output register m (SOm)... This register is a buffer register for serial output of each channel.



* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to “1”.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

Remark 2. ■: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]
• Data length of 7 or 8 bits
• Phase control of transmit/receive data
• MSB/LSB first selectable
[Clock control]
• Master/slave selection
• Phase control of I/O clock
• Setting of transfer period by prescaler and internal counter of each channel
• Maximum transfer rate Note
  During master communication: Max. f_{CLK}/2 (CSI00 only)
  Max. f_{CLK}/4
  During slave communication: Max. f_{MCK}/6
[Interrupt function]
• Transfer end interrupt/buffer empty interrupt
[Error detection flag]
• Overrun error

CSIs of the following channels supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only following CSIs can be specified.
• CSI00

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E).

The channels supporting 3-wire serial I/O (CSI00, CSI01) are channels 0 and 1 of SAU0.

• 32-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01

3-wire serial I/O (CSI00, CSI01) performs the following seven types of communication operations.

- Master transmission (See **11.5.1**.)
- Master reception (See **11.5.2**.)
- Master transmission/reception (See **11.5.3**.)
- Slave transmission (See **11.5.4**.)
- Slave reception (See **11.5.5**.)
- Slave transmission/reception (See **11.5.6**.)
- SNOOZE mode function (CSI00 only) (See **11.5.7**.)

11.5.1 Master transmission

Master transmission is that the R9A02G015 microcontroller outputs a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SO00	SCK01, SO01
Interrupt	INTCSI00	INTCSI01
	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 Note	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)).

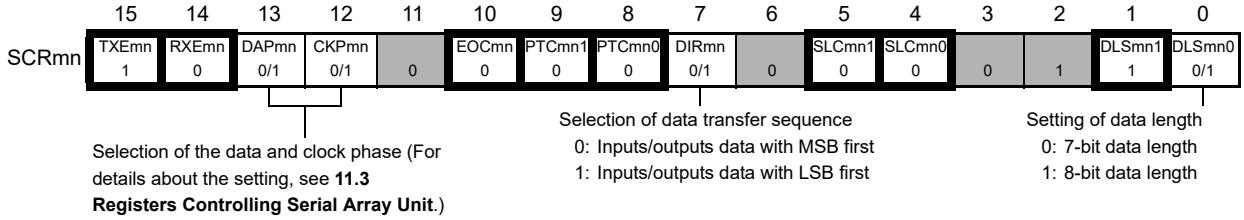
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

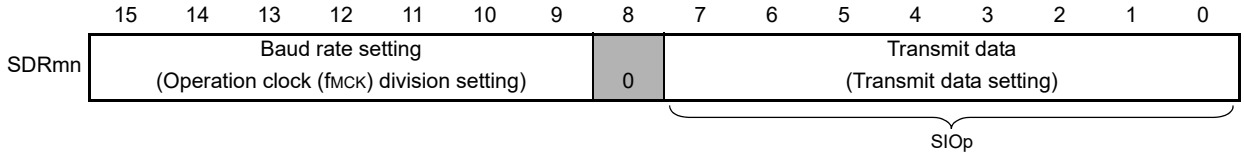
Figure 11 - 26 Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

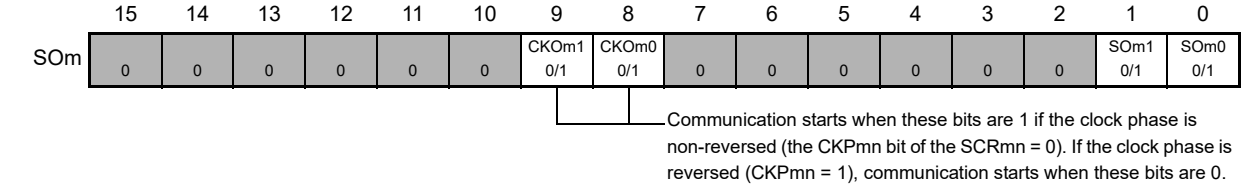
(b) Serial communication operation setting register mn (SCRmn)



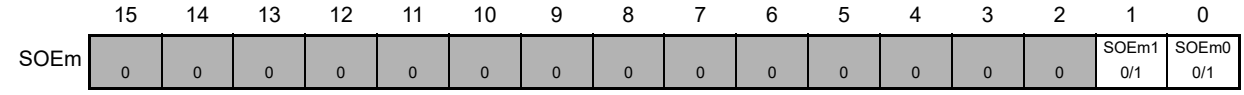
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



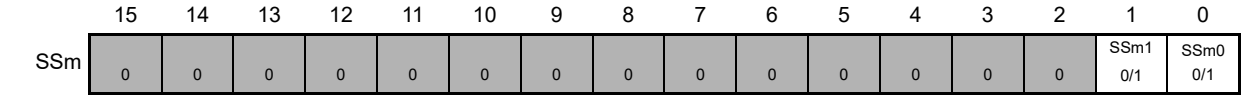
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.



(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the CSI master transmission mode,

▒: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11 - 27 Initial Setting Procedure for Master Transmission

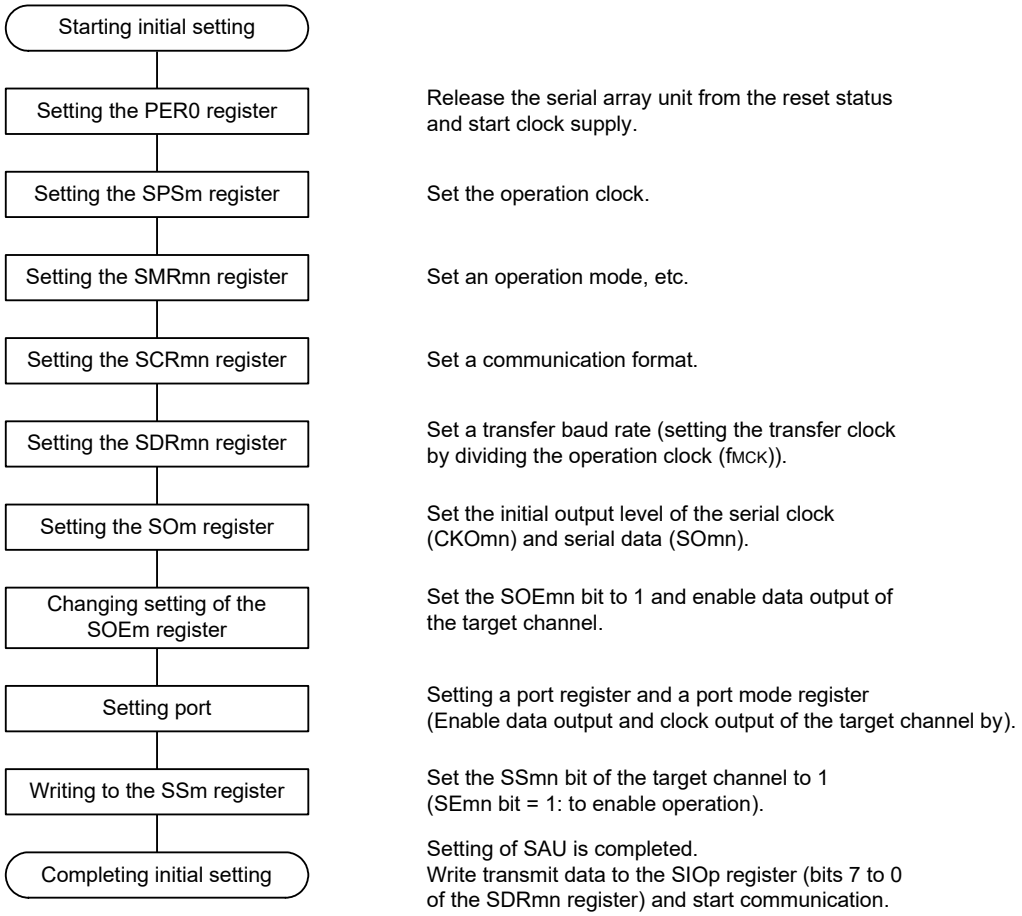


Figure 11 - 28 Procedure for Stopping Master Transmission

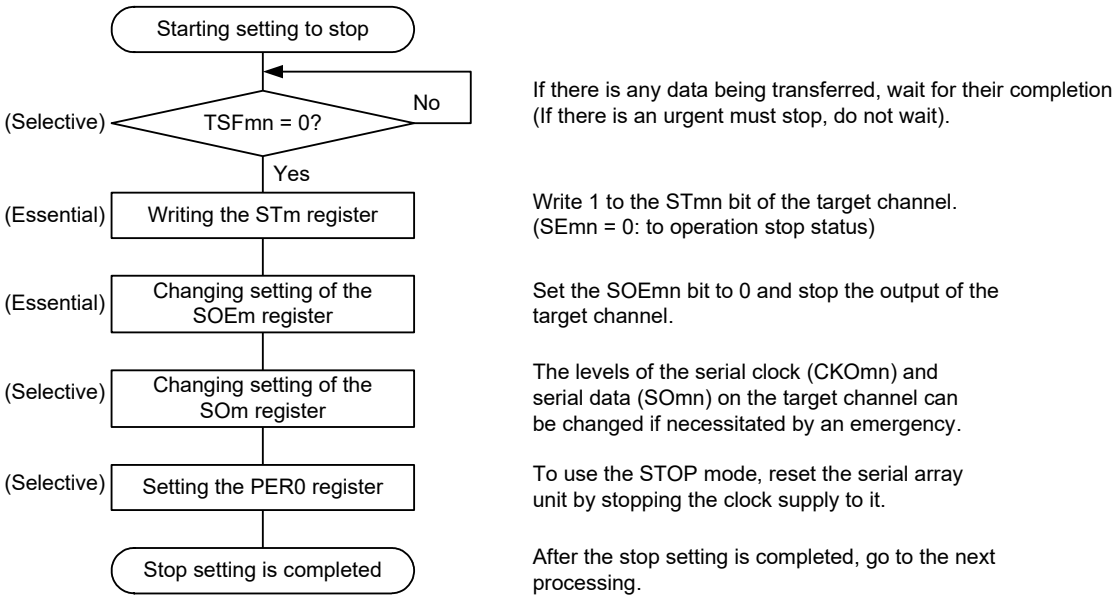
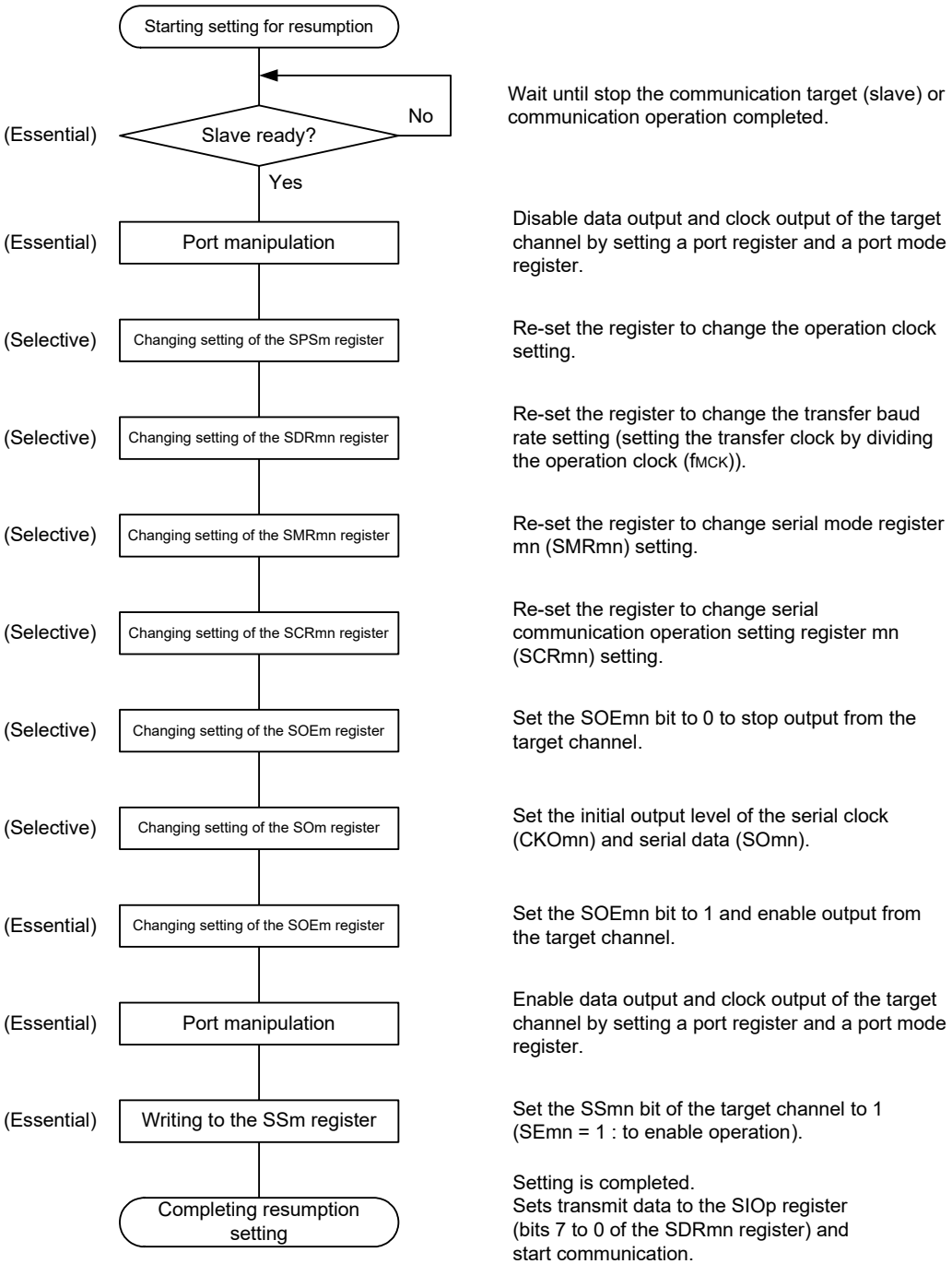


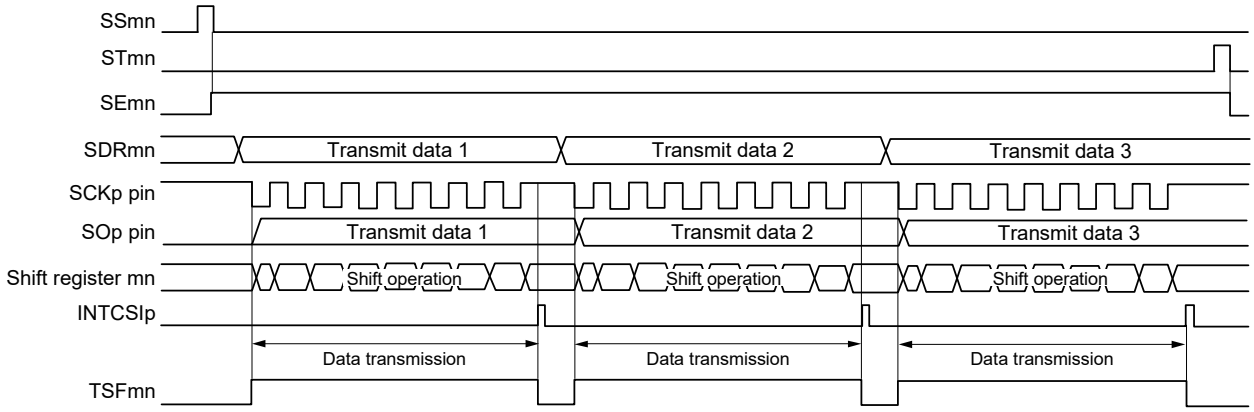
Figure 11 - 29 Procedure for Resuming Master Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

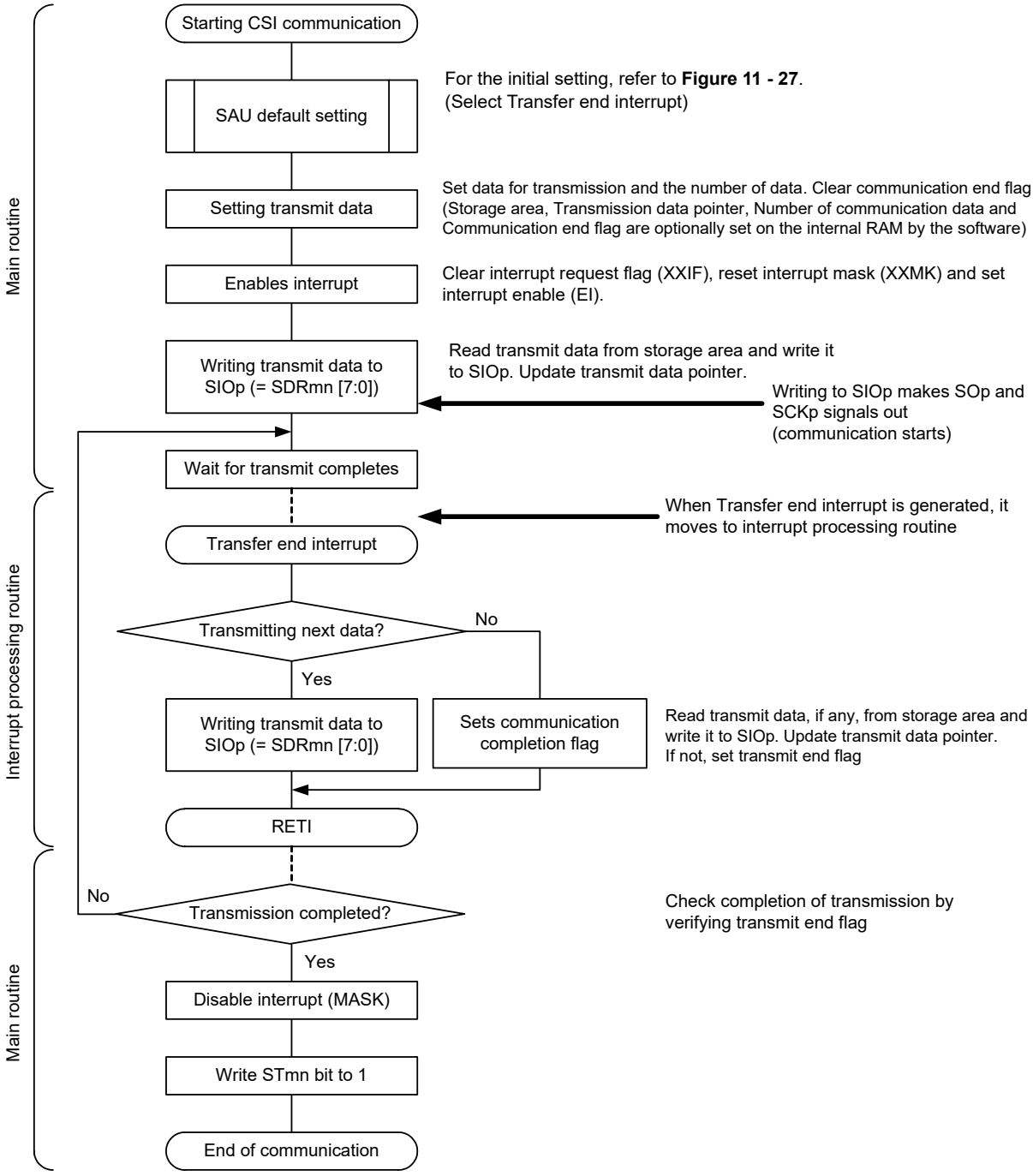
(3) Processing flow (in single-transmission mode)

Figure 11 - 30 Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



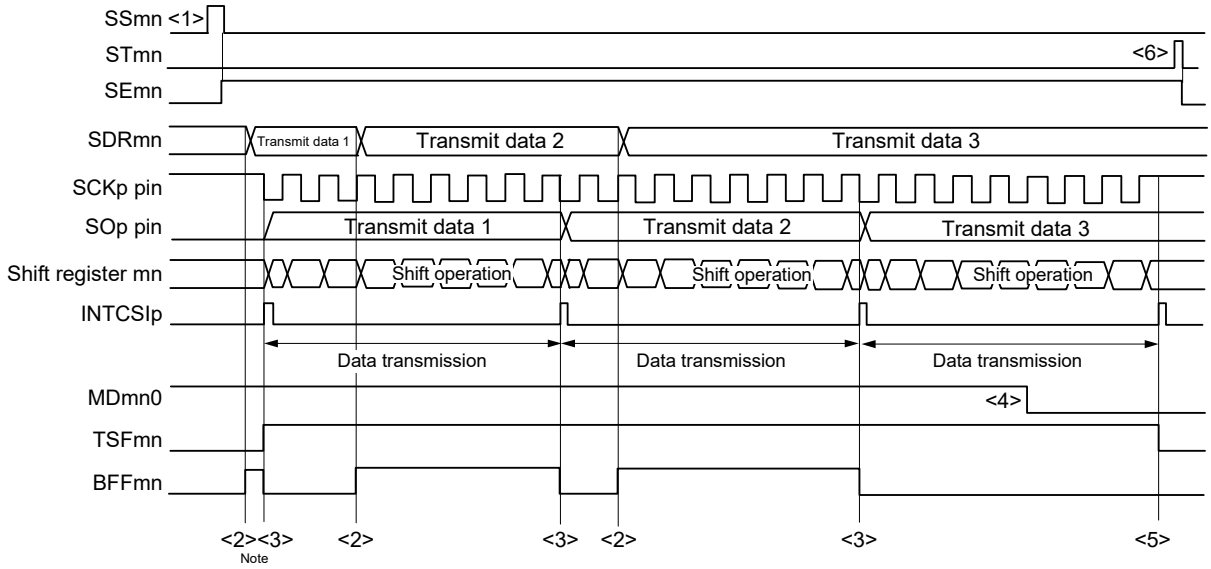
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 11 - 31 Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 11 - 32 Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

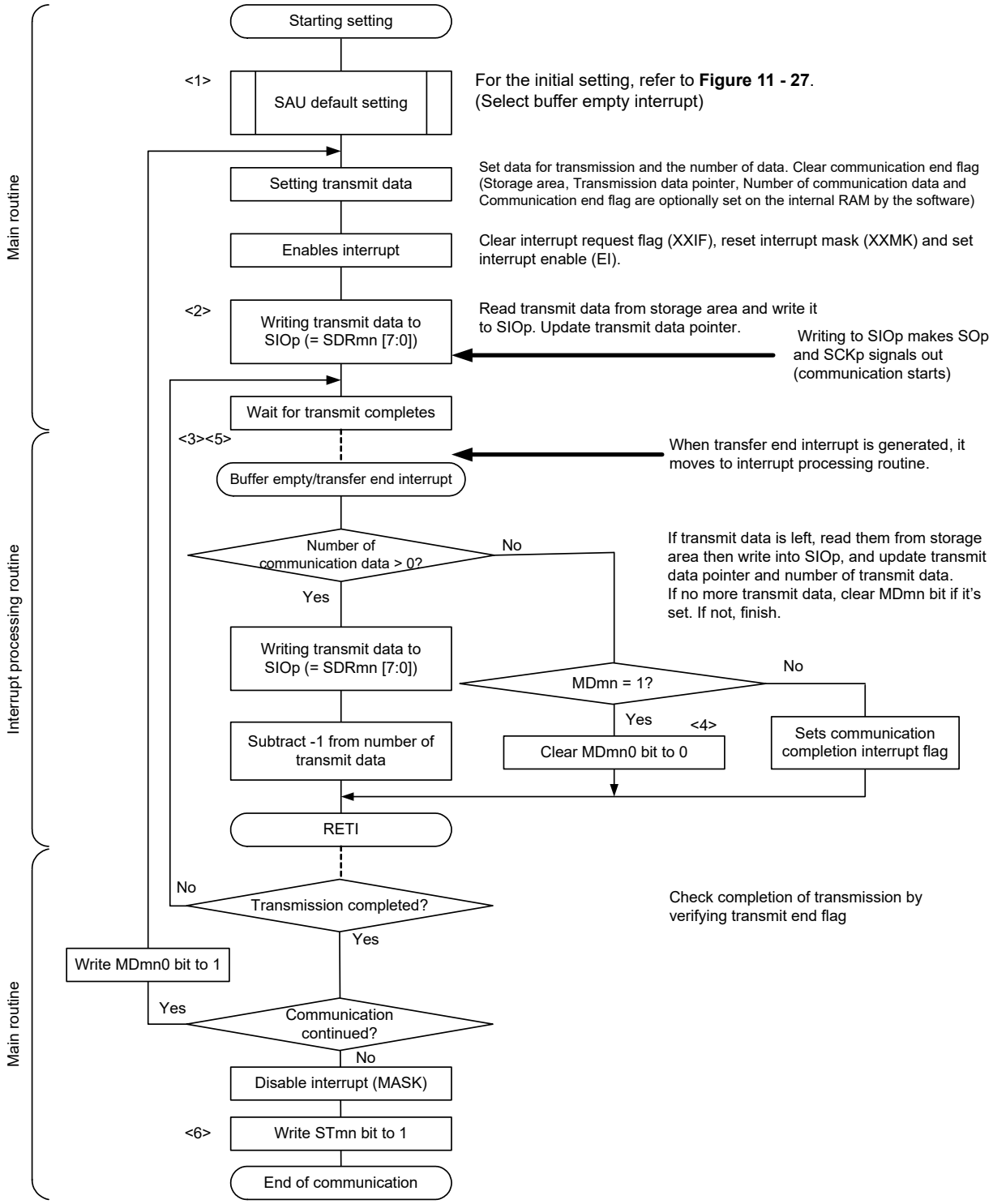


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 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 will be rewritten before the transfer end interrupt of the last transmit data.**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 11 - 33 Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 11 - 32 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

11.5.2 Master reception

Master reception is that the R9A02G015 microcontroller outputs a transfer clock and receives data from other device.

3-Wire Serial I/O	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	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 (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate Note	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)).

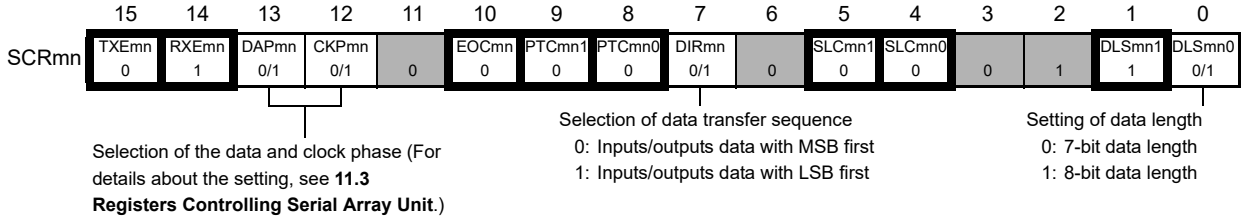
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

(1) Register setting

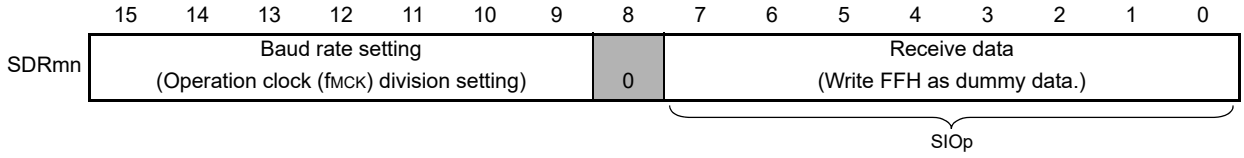
Figure 11 - 34 Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

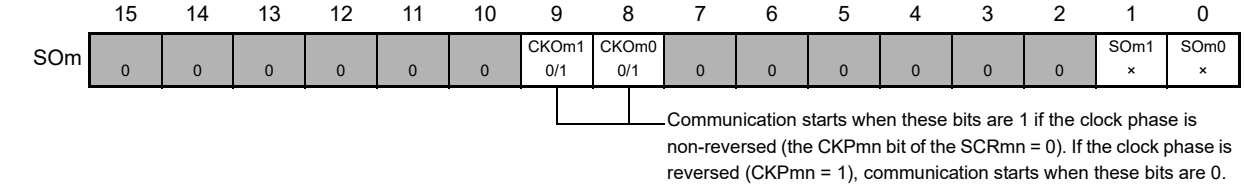
(b) Serial communication operation setting register mn (SCRmn)



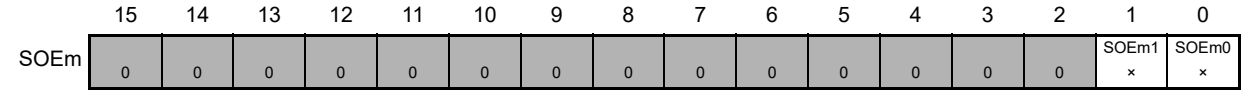
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



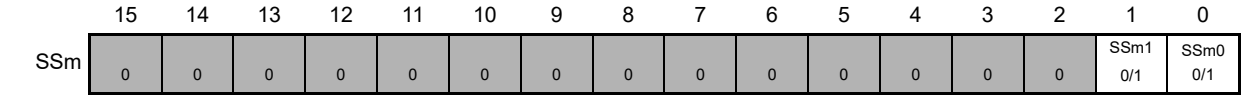
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(e) Serial output enable register m (SOEm)... The register that not used in this mode.



(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the CSI master reception mode,

▒: Setting disabled (set to the initial value)

×: 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

Figure 11 - 35 Initial Setting Procedure for Master Reception

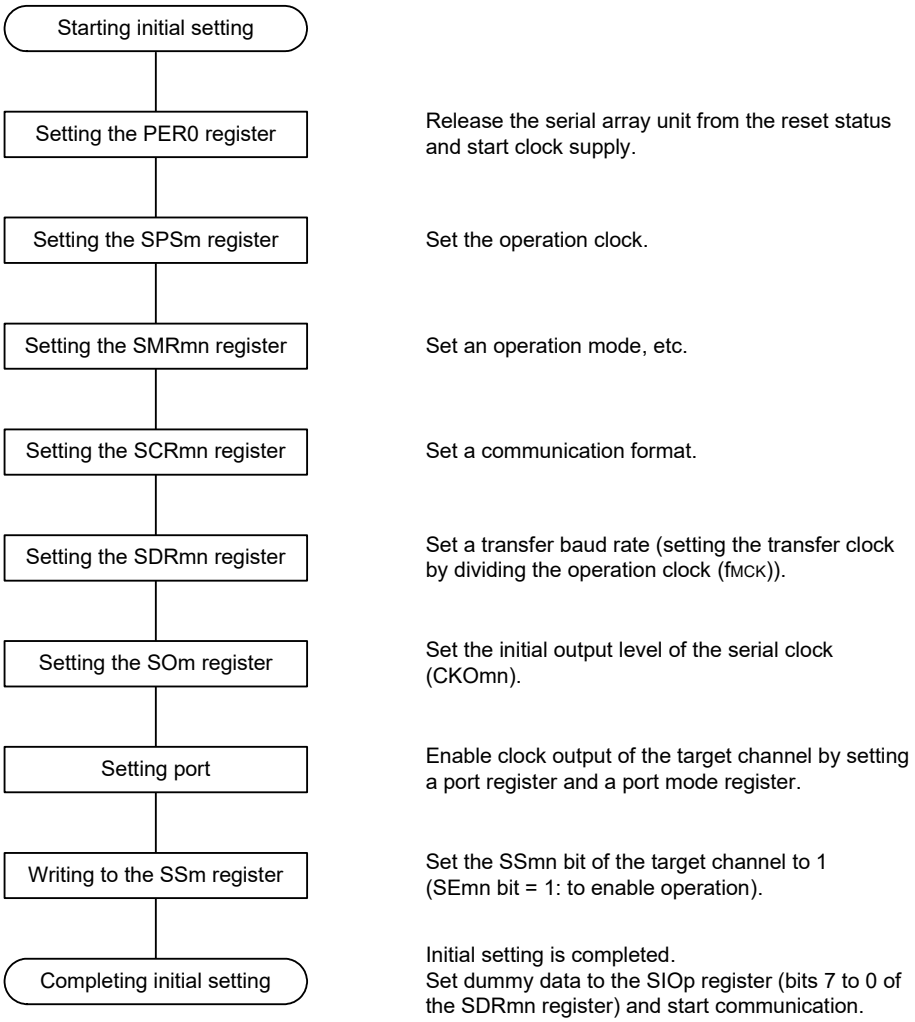


Figure 11 - 36 Procedure for Stopping Master Reception

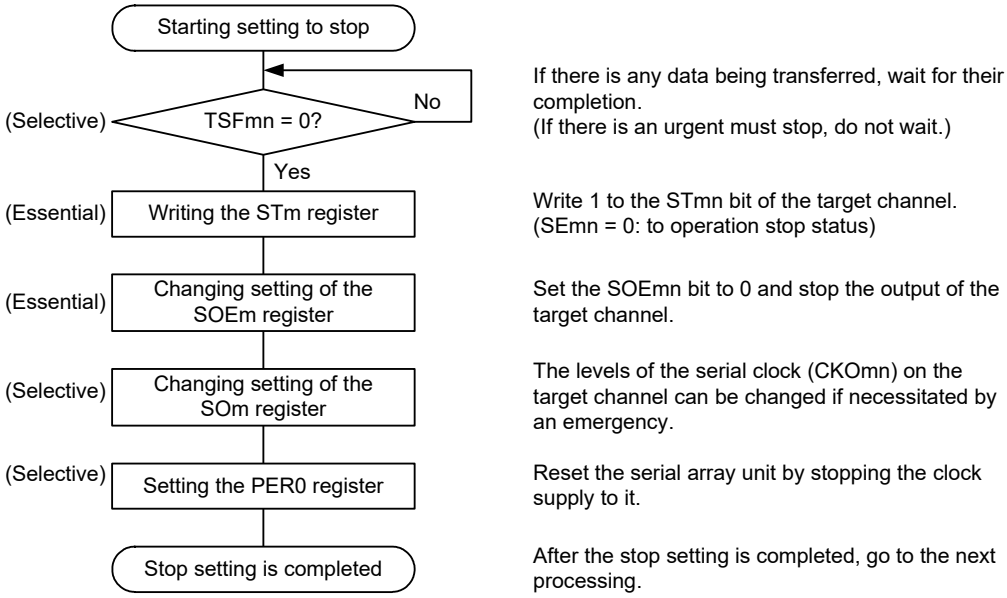
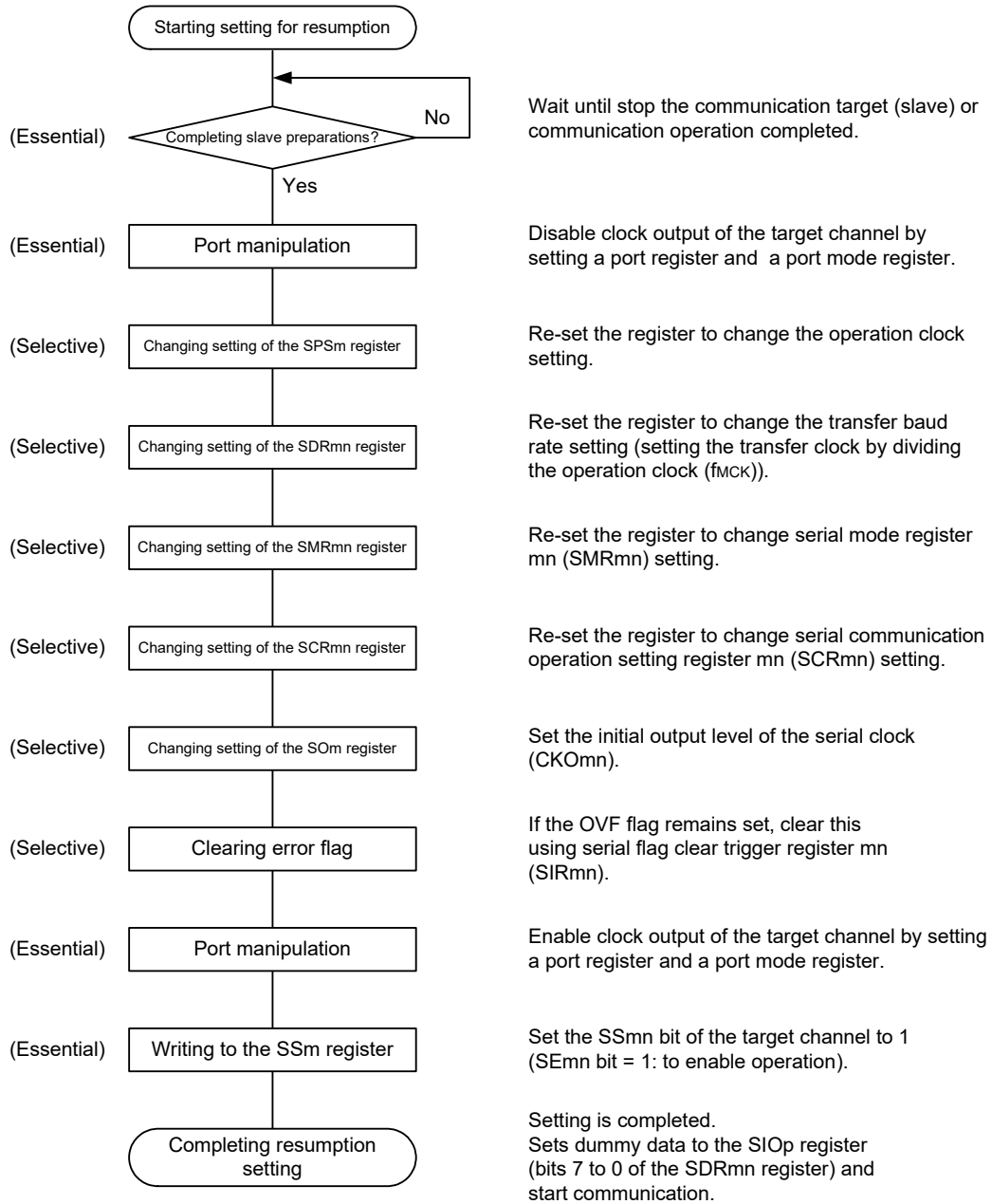


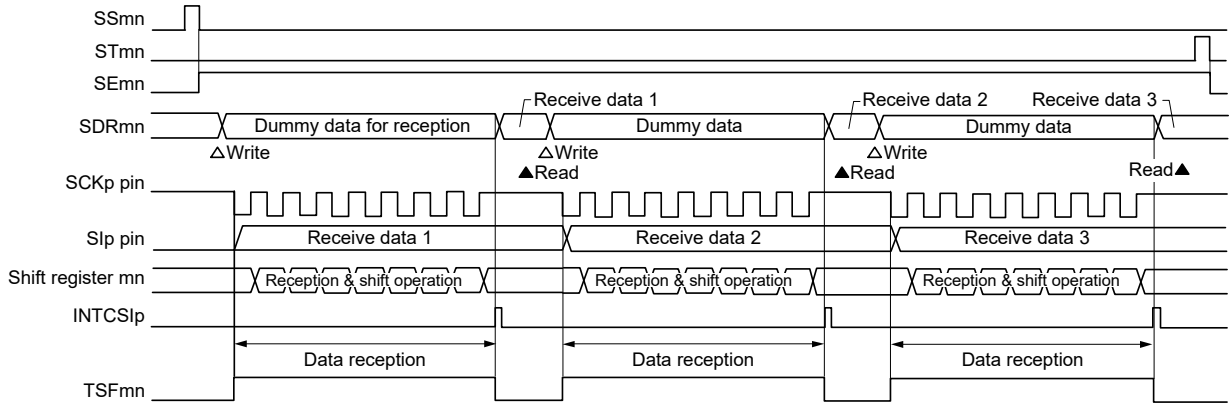
Figure 11 - 37 Procedure for Resuming Master Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

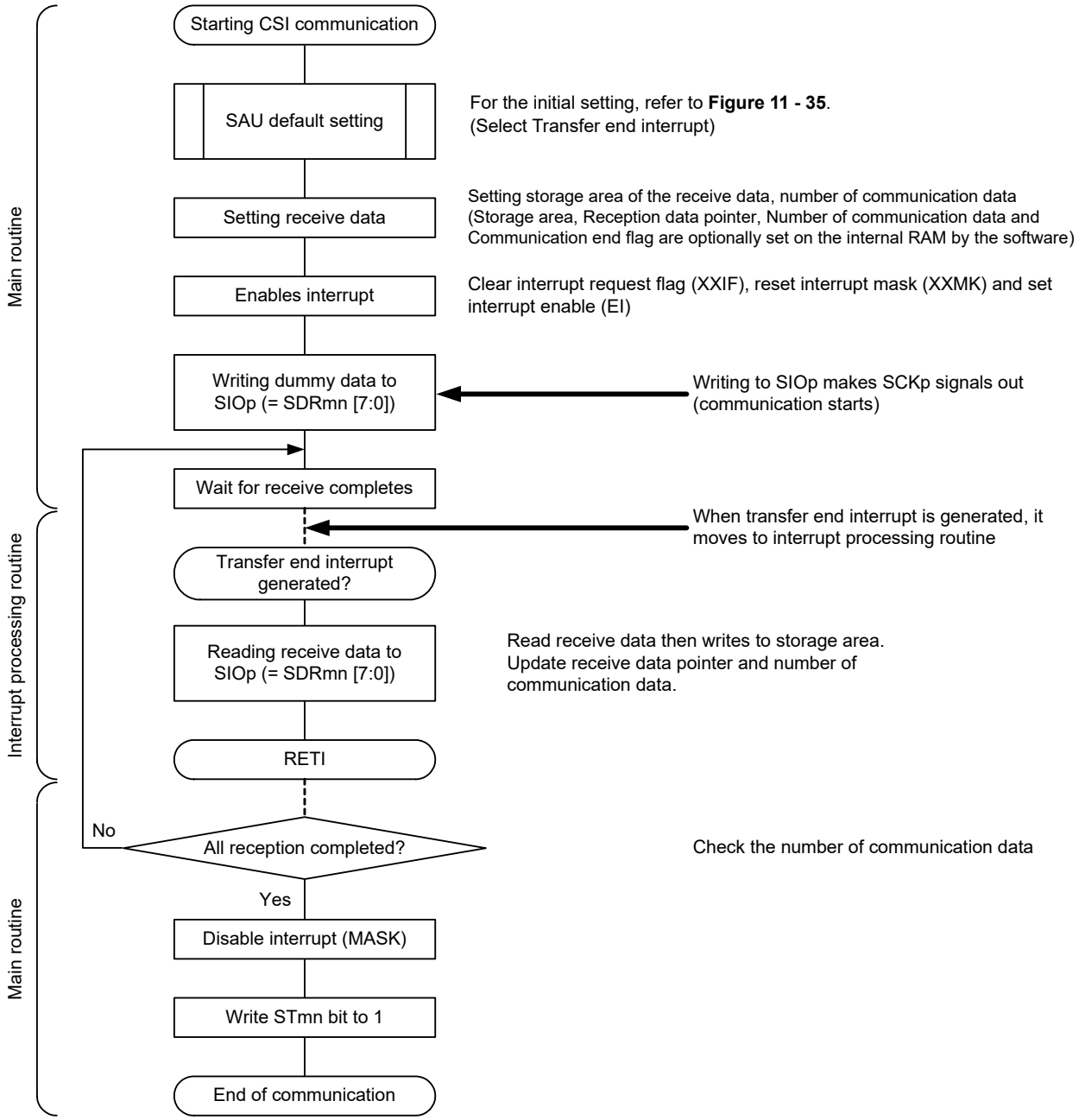
(3) Processing flow (in single-reception mode)

Figure 11 - 38 Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



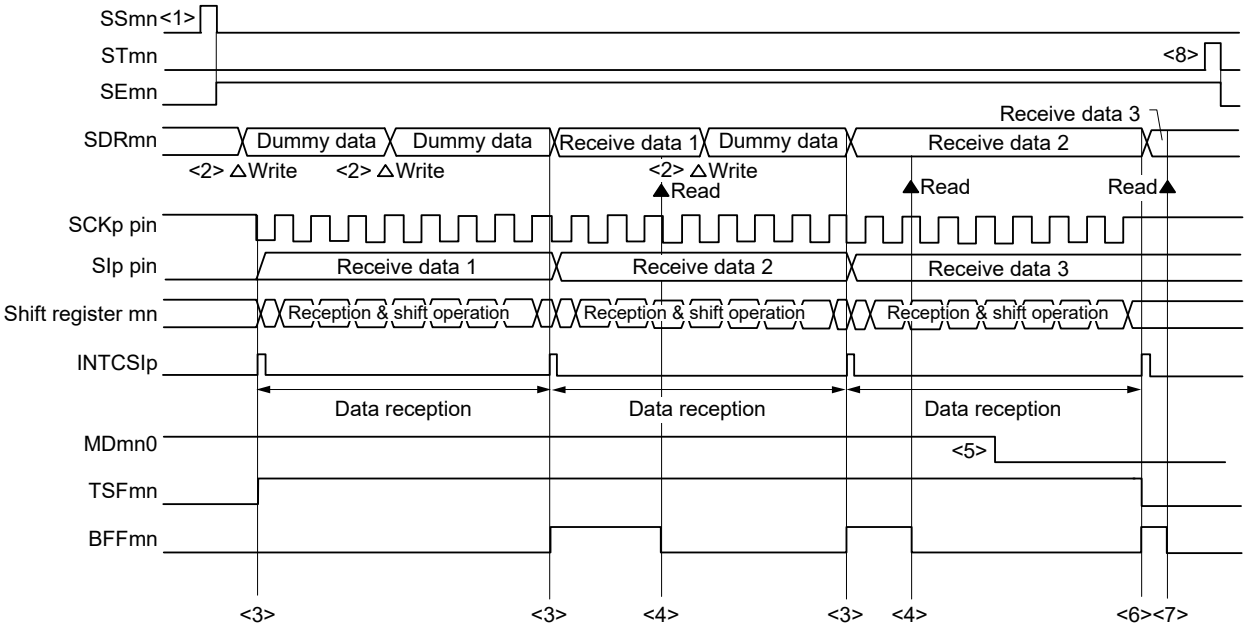
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 11 - 39 Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 11 - 40 Timing Chart of Master Reception (in Continuous Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

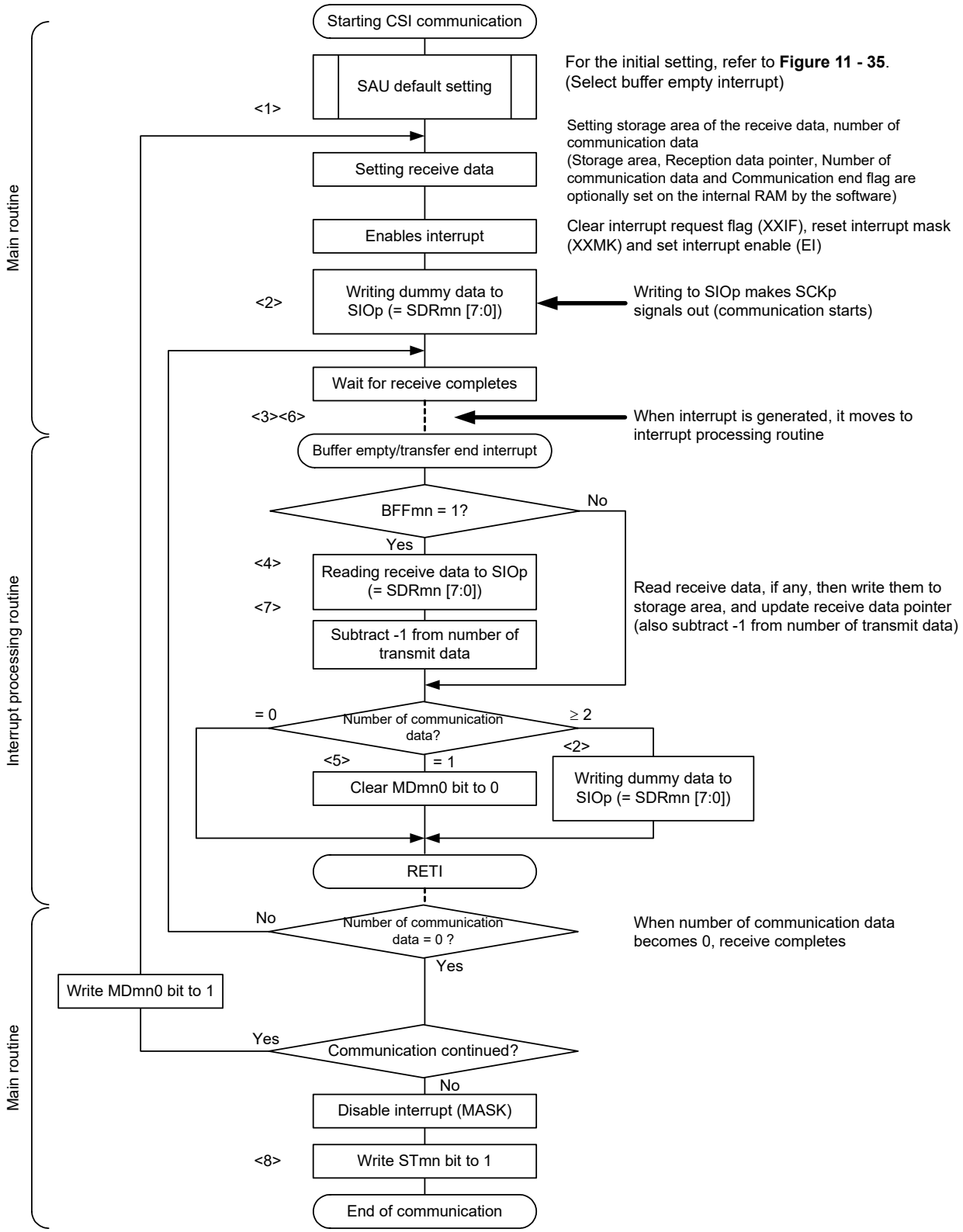


Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 11 - 41 Flowchart of Master Reception (in Continuous Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 11 - 41 Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 11 - 40 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

11.5.3 Master transmission/reception

Master transmission/reception is that the R9A02G015 microcontroller outputs a transfer clock and transmits/receives data to/from another device.

3-Wire Serial I/O	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	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 (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate Note	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)).

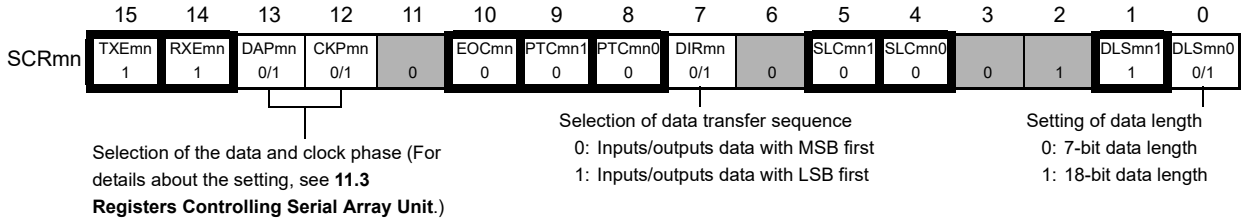
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

(1) Register setting

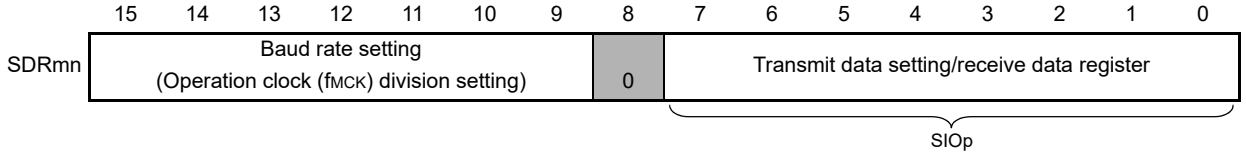
Figure 11 - 42 Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

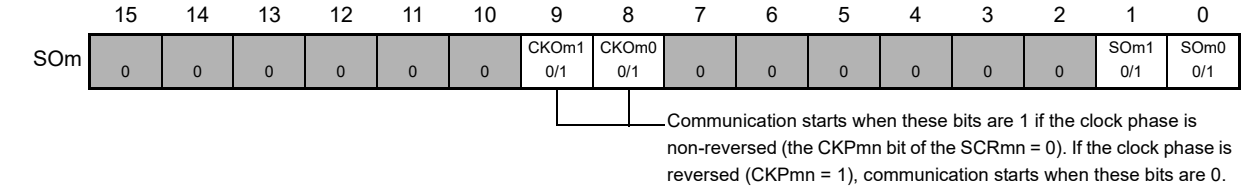
(b) Serial communication operation setting register mn (SCRmn)



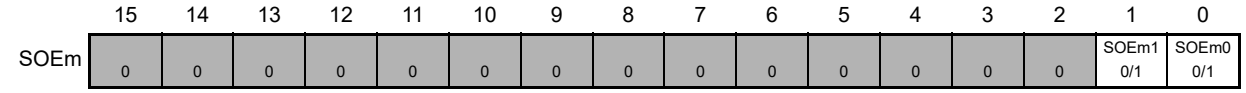
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



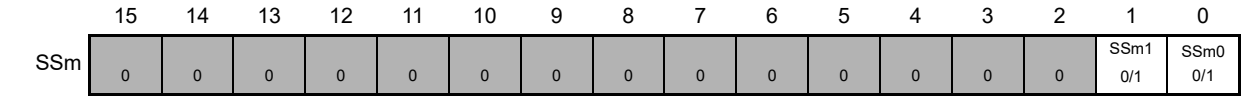
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.



(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the CSI master transmission/reception mode,

▒: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11 - 43 Initial Setting Procedure for Master Transmission/Reception

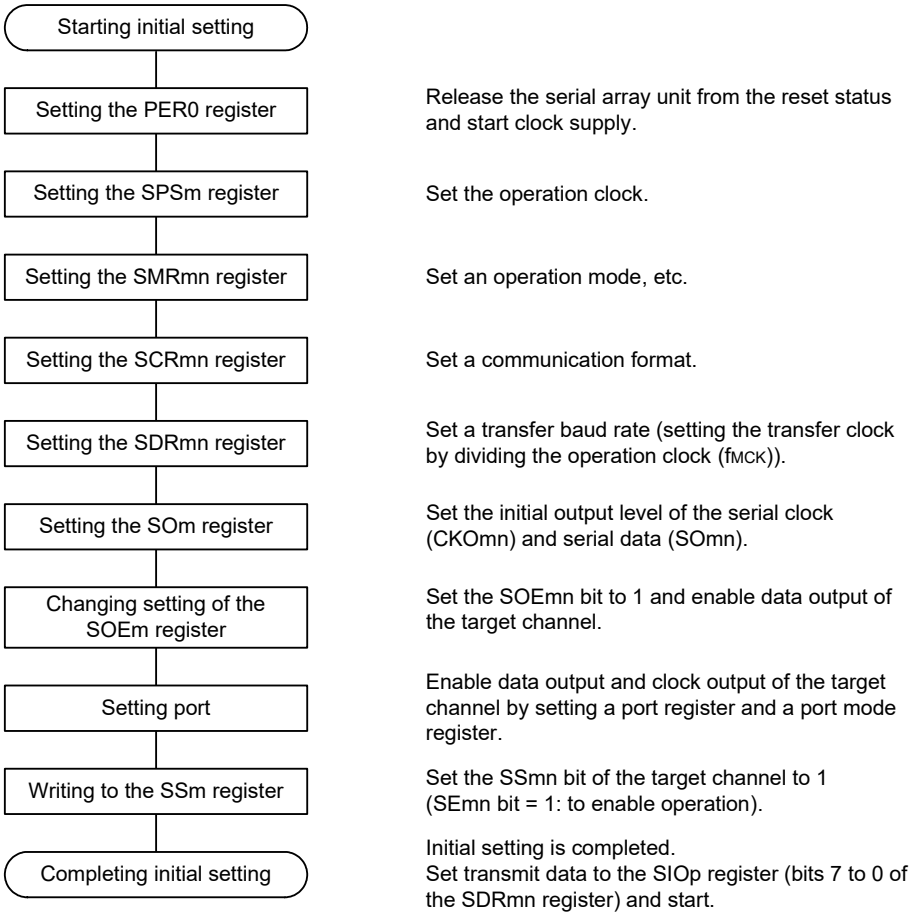


Figure 11 - 44 Procedure for Stopping Master Transmission/Reception

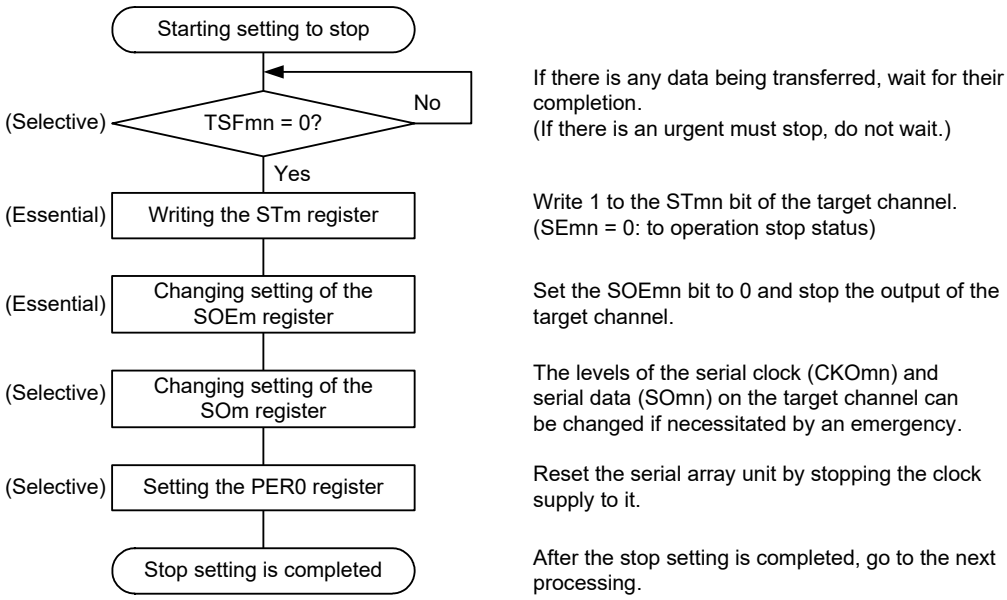
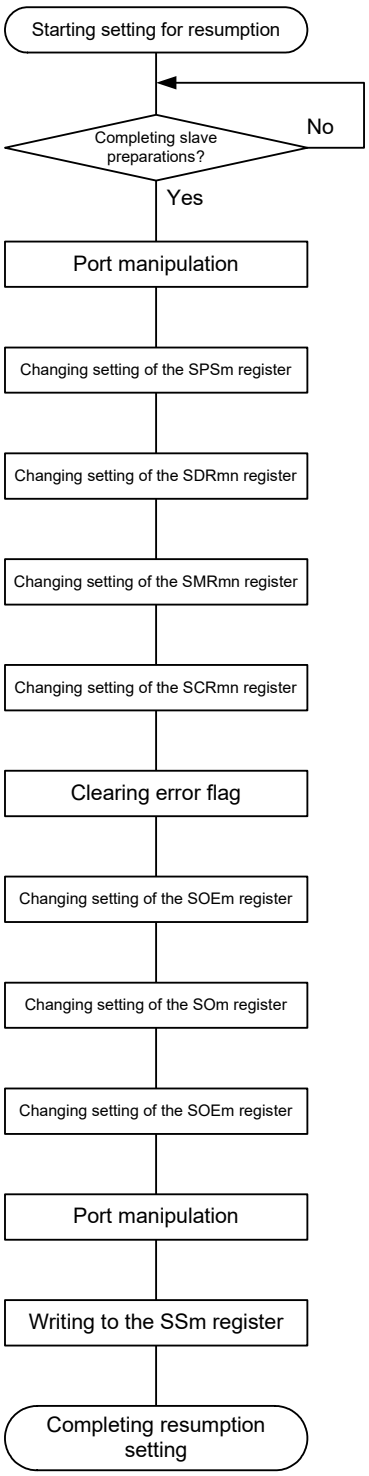


Figure 11 - 45 Procedure for Resuming Master Transmission/Reception



Wait until stop the communication target (slave) or communication operation completed.

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

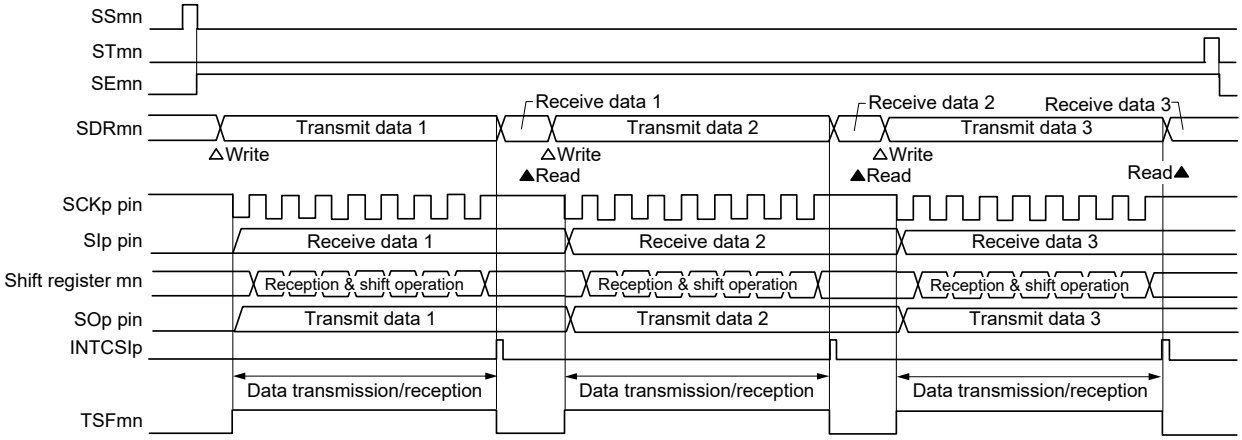
Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

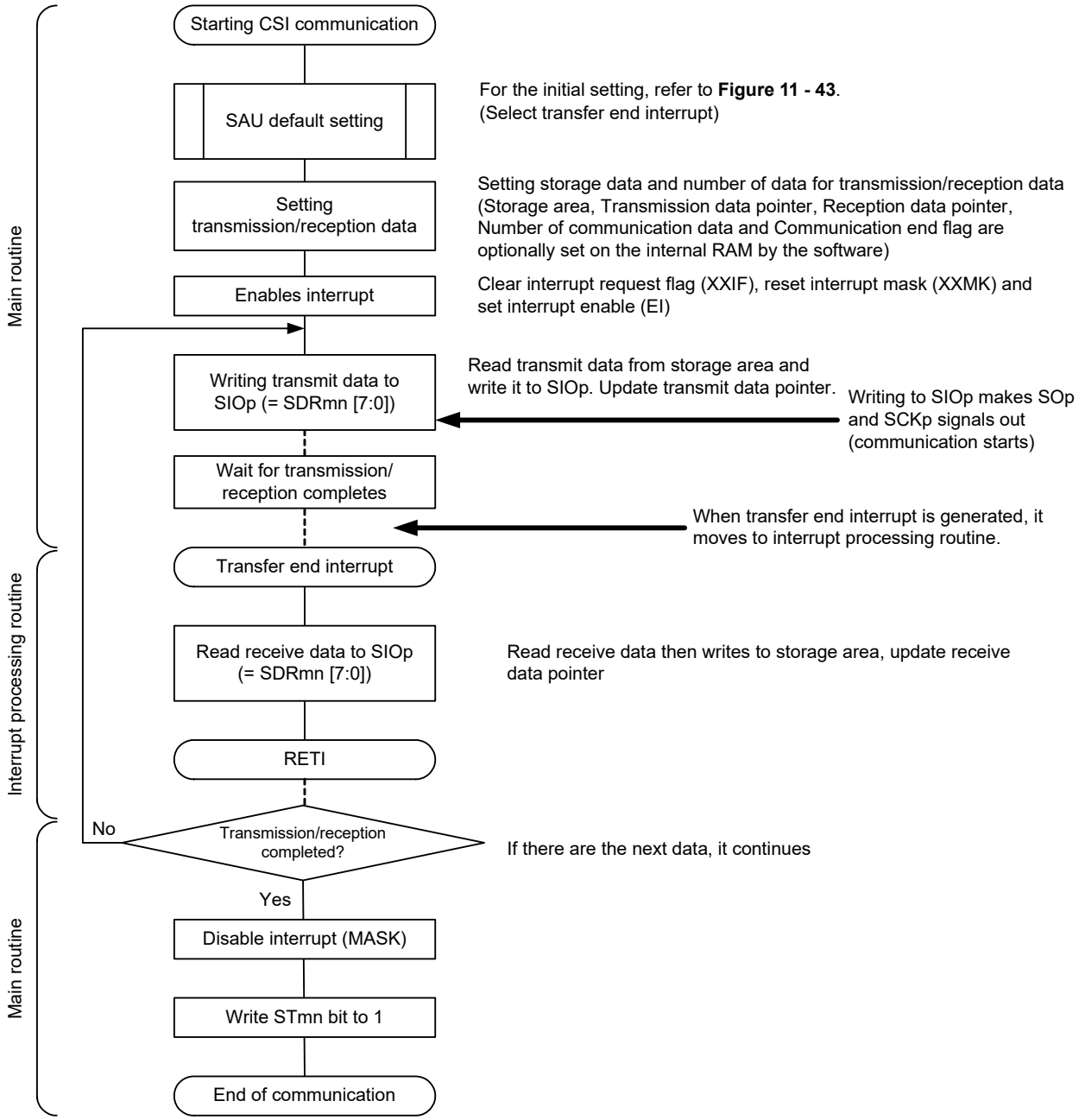
(3) Processing flow (in single-transmission/reception mode)

Figure 11 - 46 Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



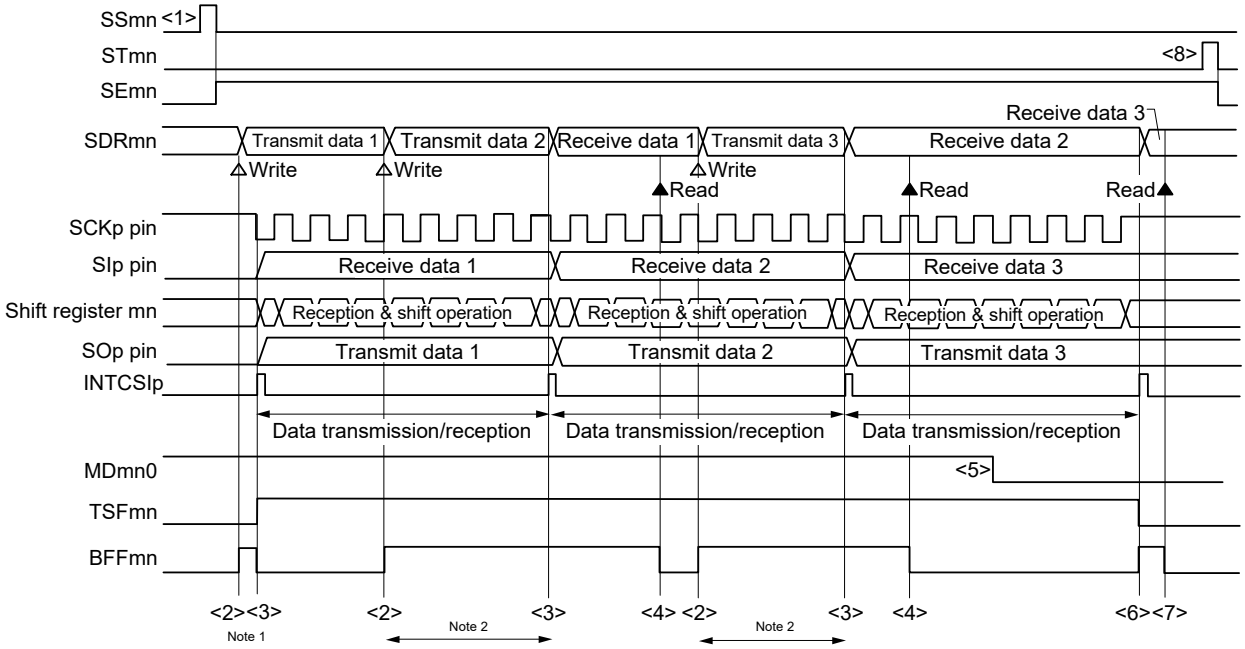
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 11 - 47 Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 11 - 48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

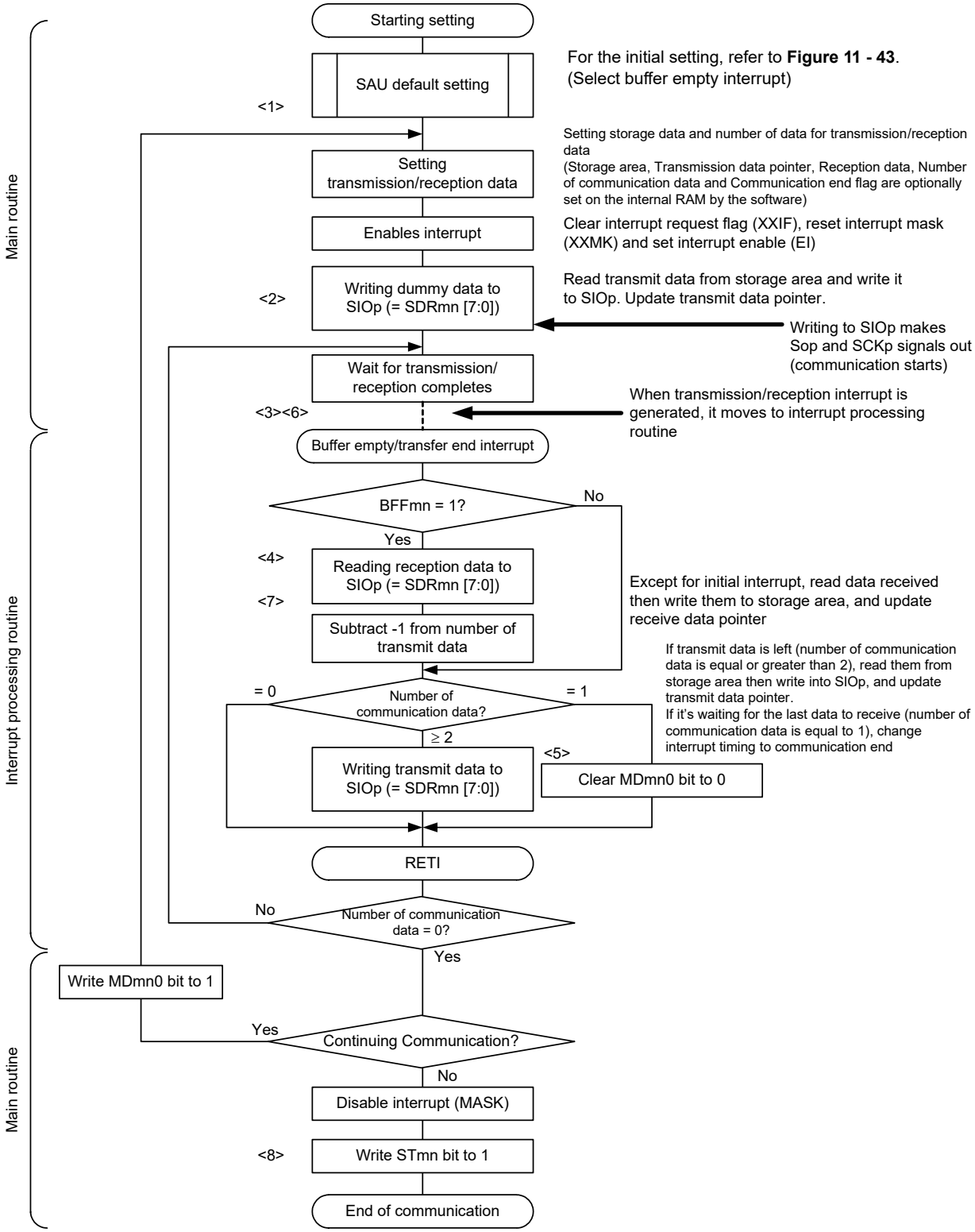
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 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.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 11 - 49 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 11 - 49 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 11 - 48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

11.5.4 Slave transmission

Slave transmission is that the R9A02G015 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SO00	SCK01, SO01
Interrupt	INTCSI00	INTCSI01
	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 (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2.	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note 1. Because the external serial clock input to the SCK00 and SCK01 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 in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)).

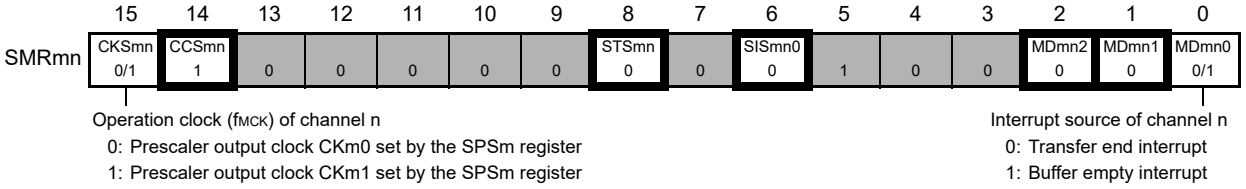
Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

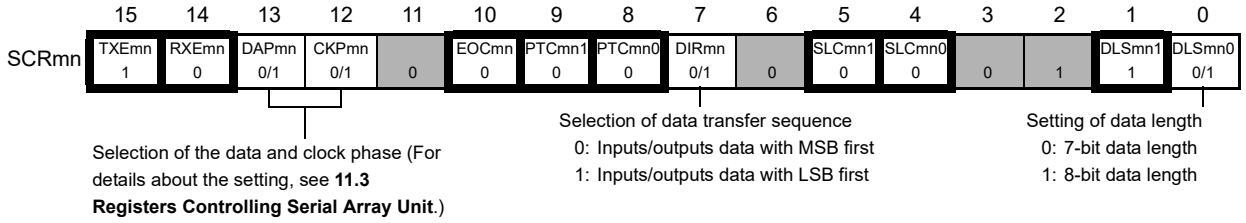
(1) Register setting

Figure 11 - 50 Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01)

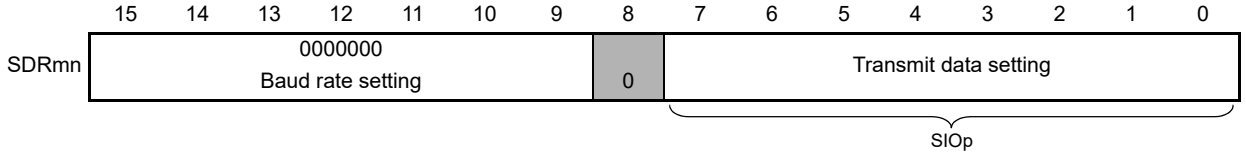
(a) Serial mode register mn (SMRmn)



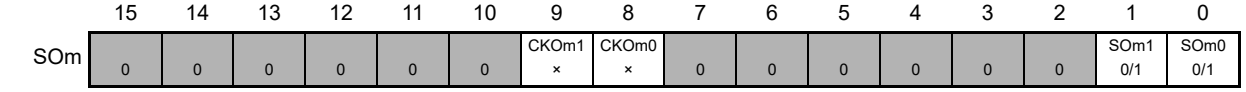
(b) Serial communication operation setting register mn (SCRmn)



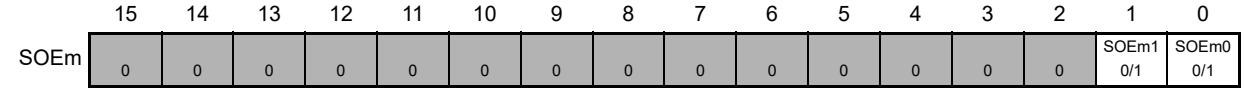
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



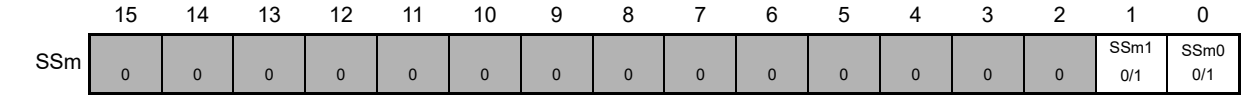
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.



(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the CSI slave transmission mode,

▒: Setting disabled (set to the initial value)

×: 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

Figure 11 - 51 Initial Setting Procedure for Slave Transmission

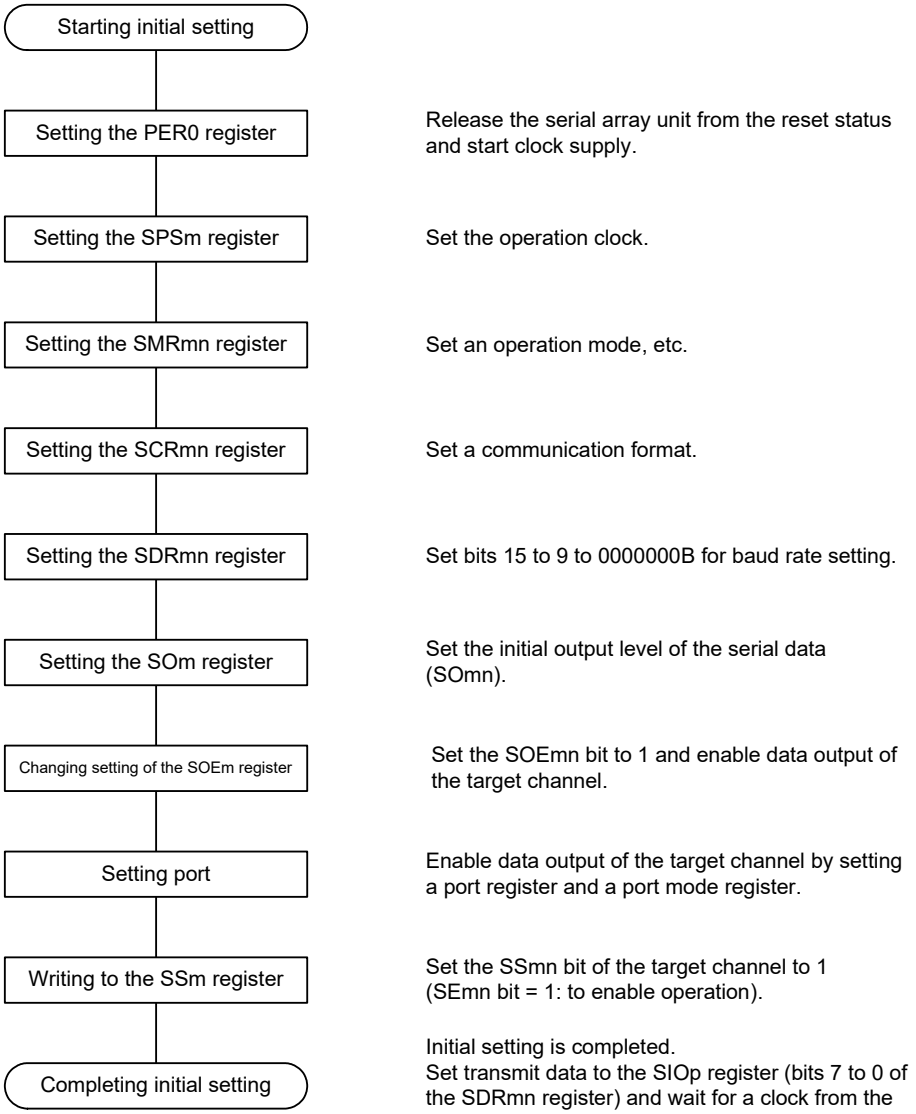


Figure 11 - 52 Procedure for Stopping Slave Transmission

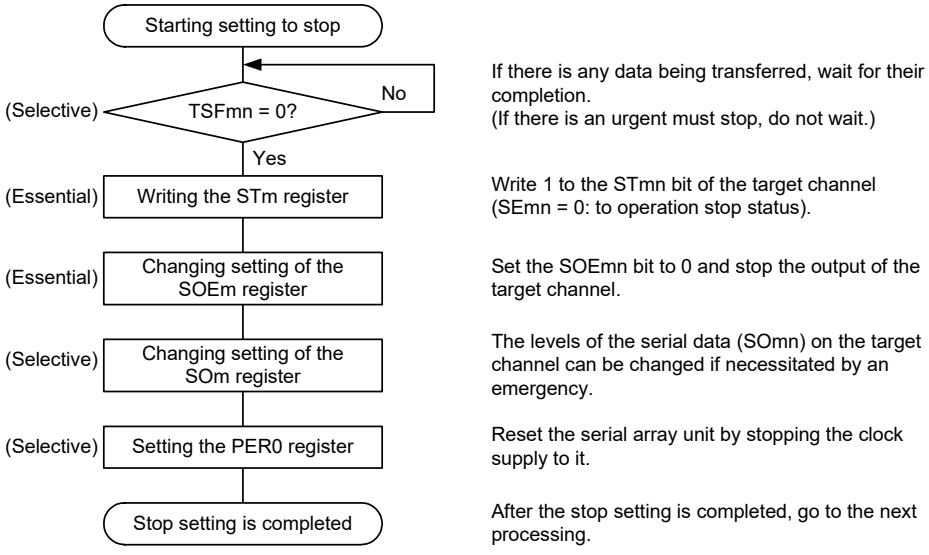
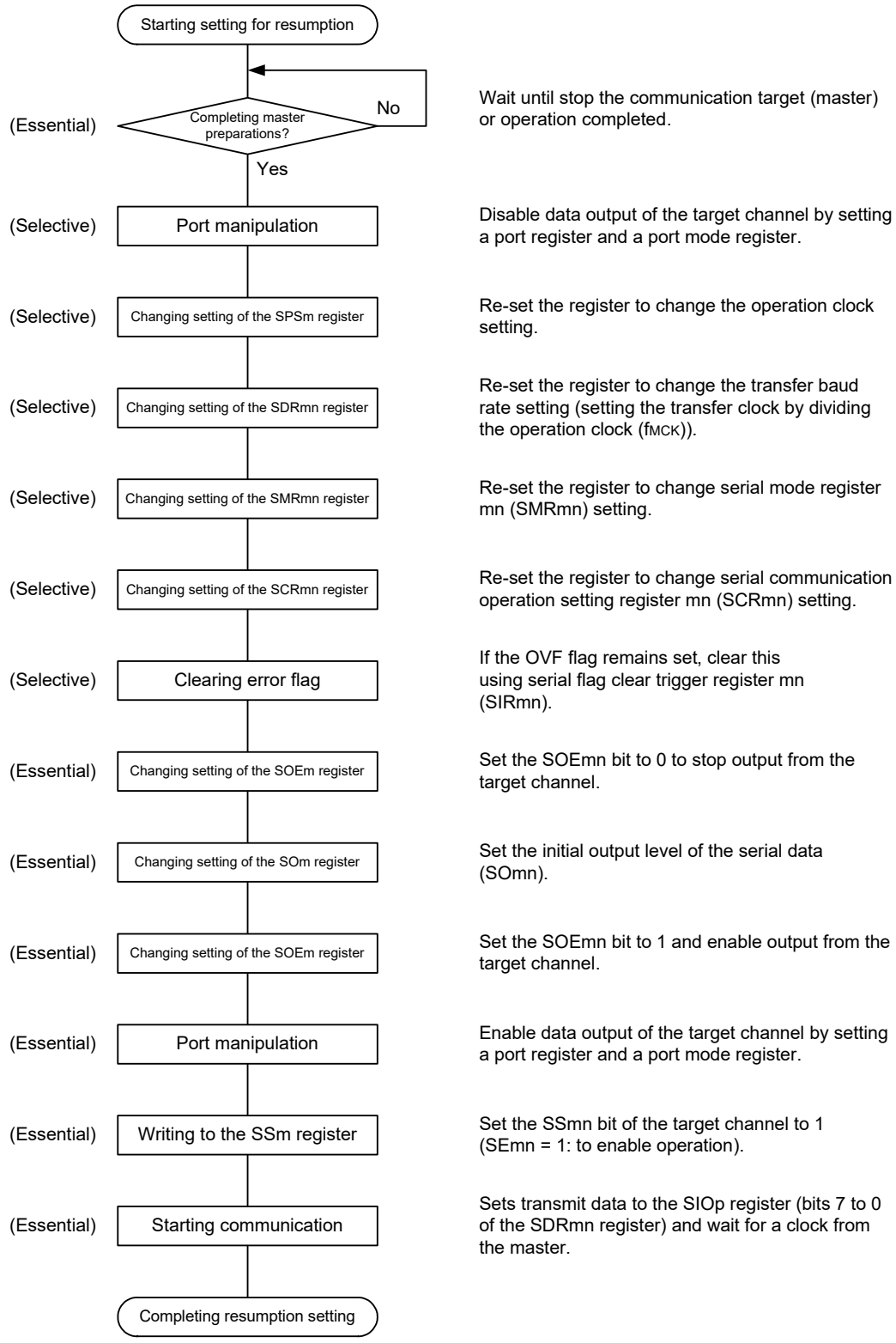


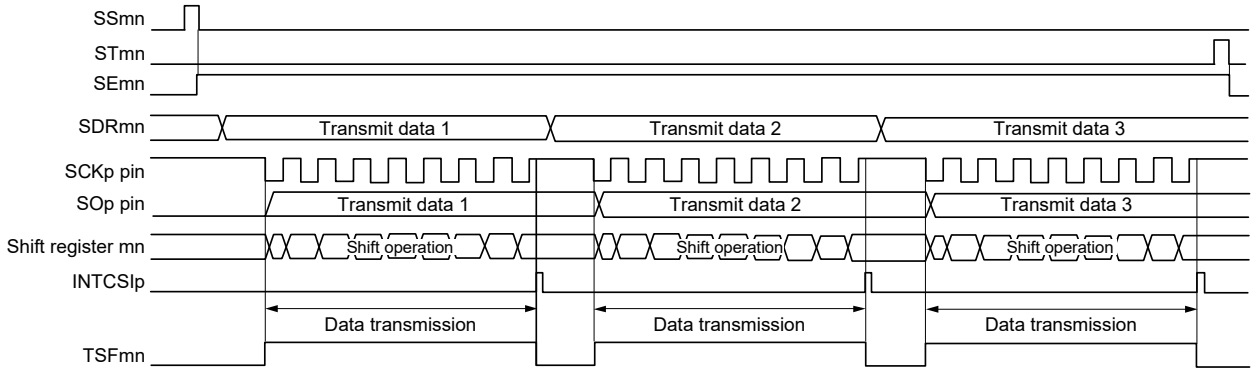
Figure 11 - 53 Procedure for Resuming Slave Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

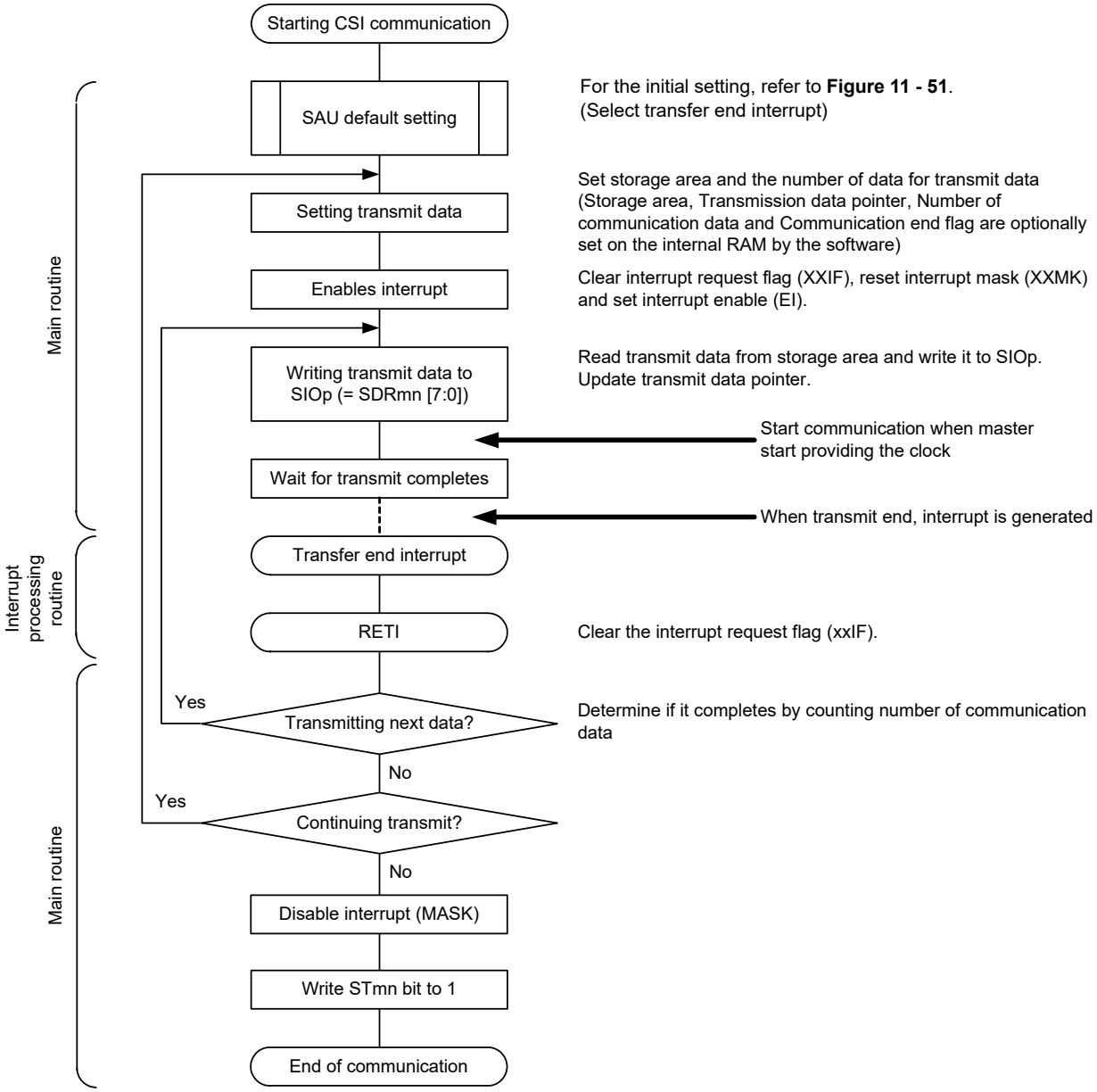
(3) Processing flow (in single-transmission mode)

Figure 11 - 54 Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



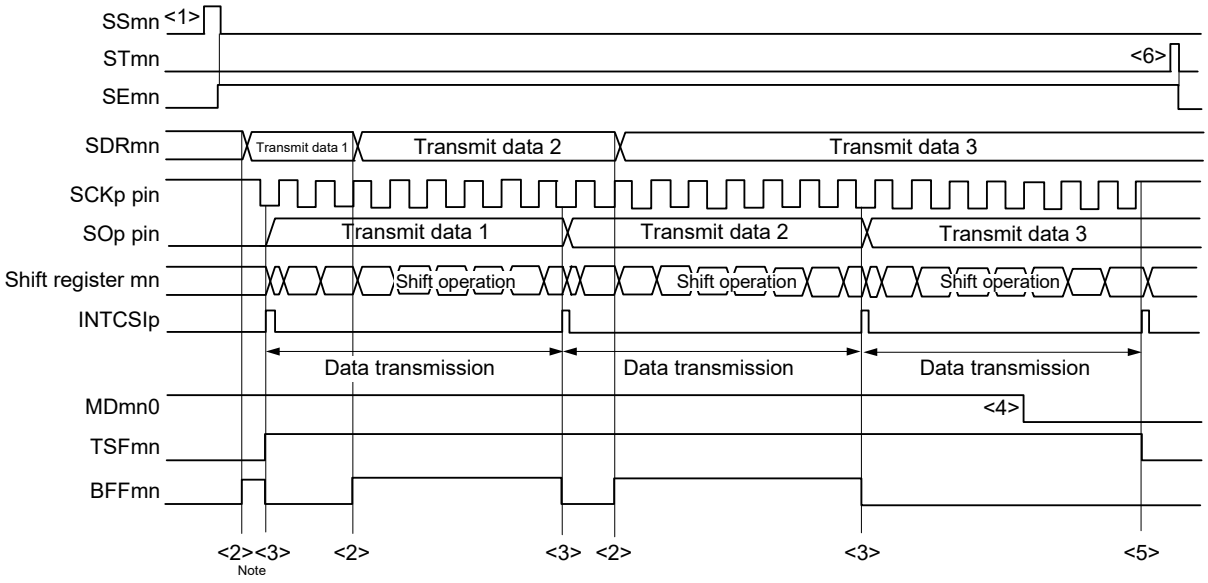
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 11 - 55 Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 11 - 56 Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

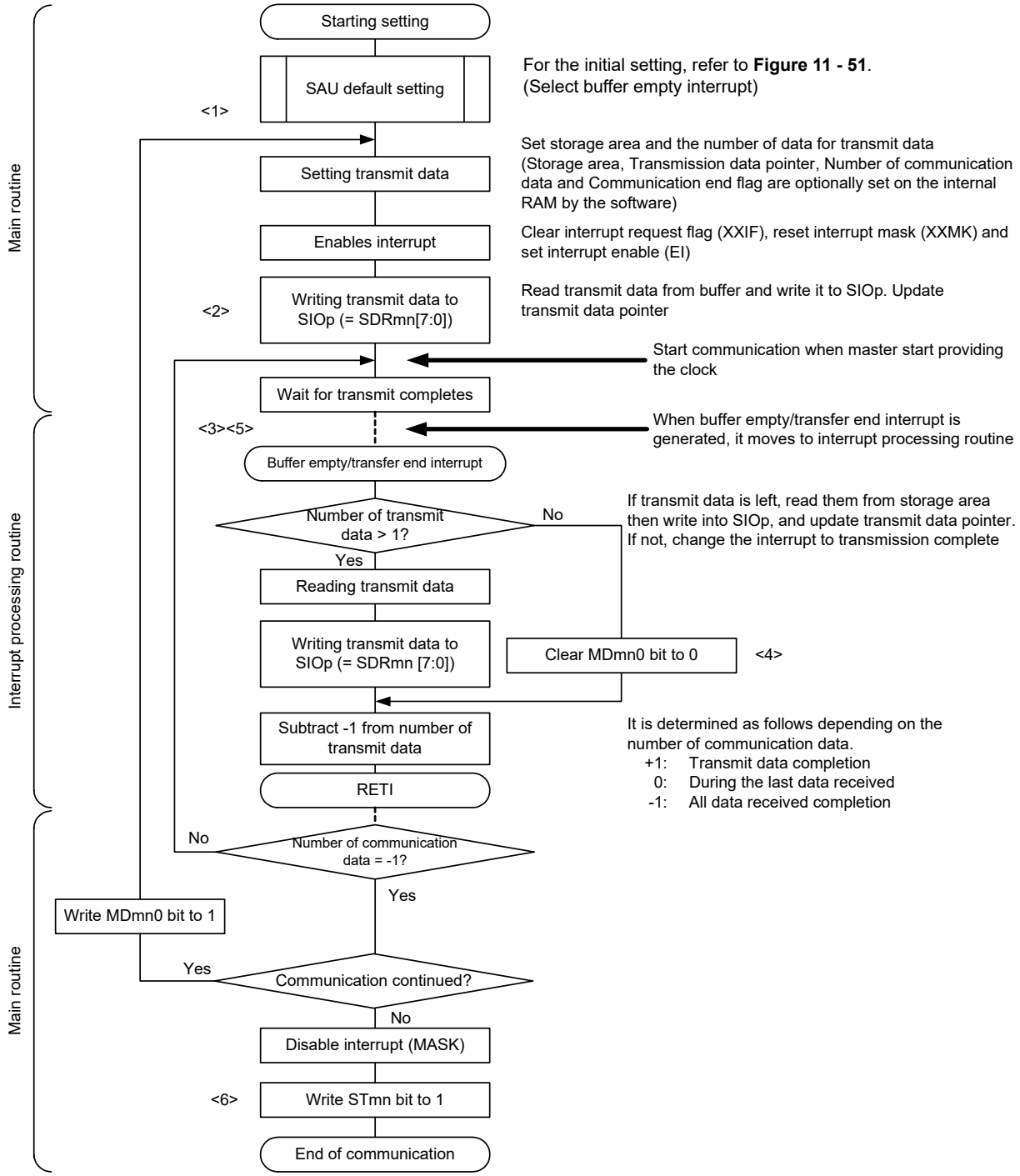


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 11 - 57 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 11 - 56 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

11.5.5 Slave reception

Slave reception is that the R9A02G015 microcontroller receives data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note 1. Because the external serial clock input to the SCK00 and SCK011 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 in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 11 - 58 Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

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

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **11.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Receive data							

SIOp

(d) Serial output register m (SOm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 ×

(e) Serial output enable register m (SOEm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the CSI slave reception mode,
■: Setting disabled (set to the initial value)
×: 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

Figure 11 - 59 Initial Setting Procedure for Slave Reception

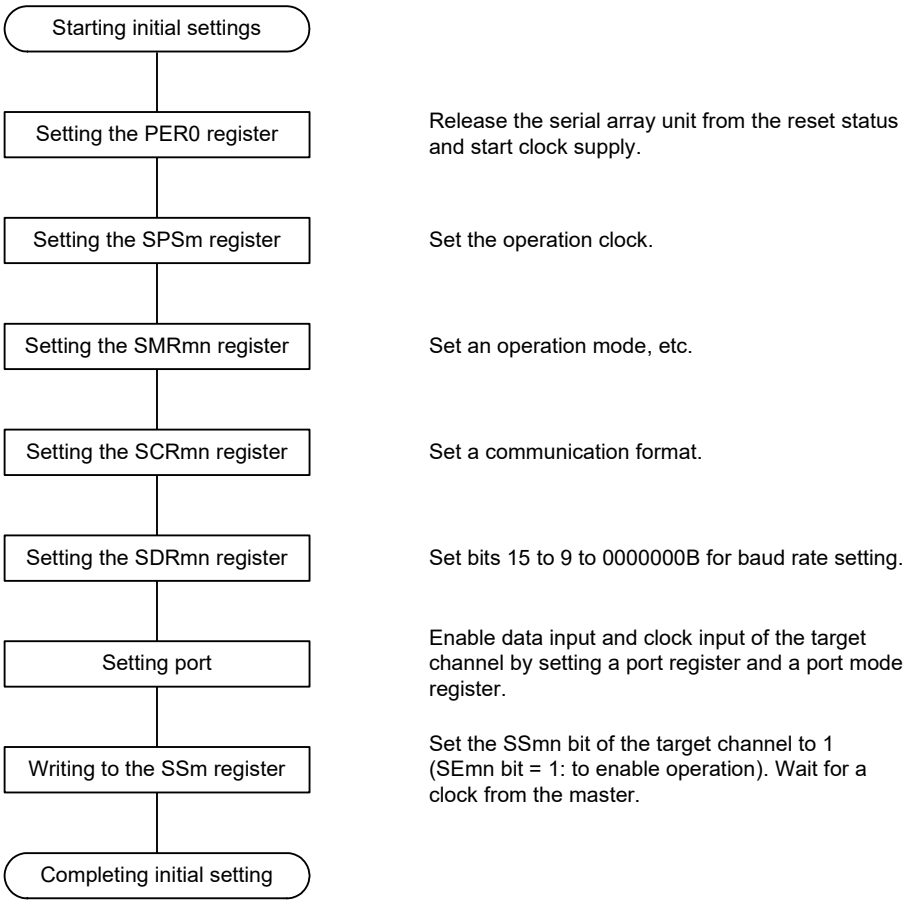


Figure 11 - 60 Procedure for Stopping Slave Reception

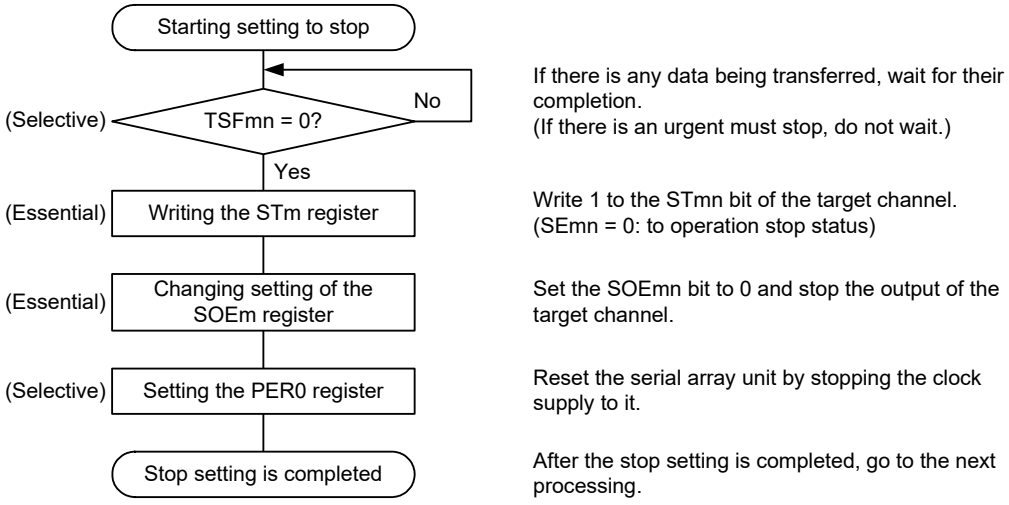
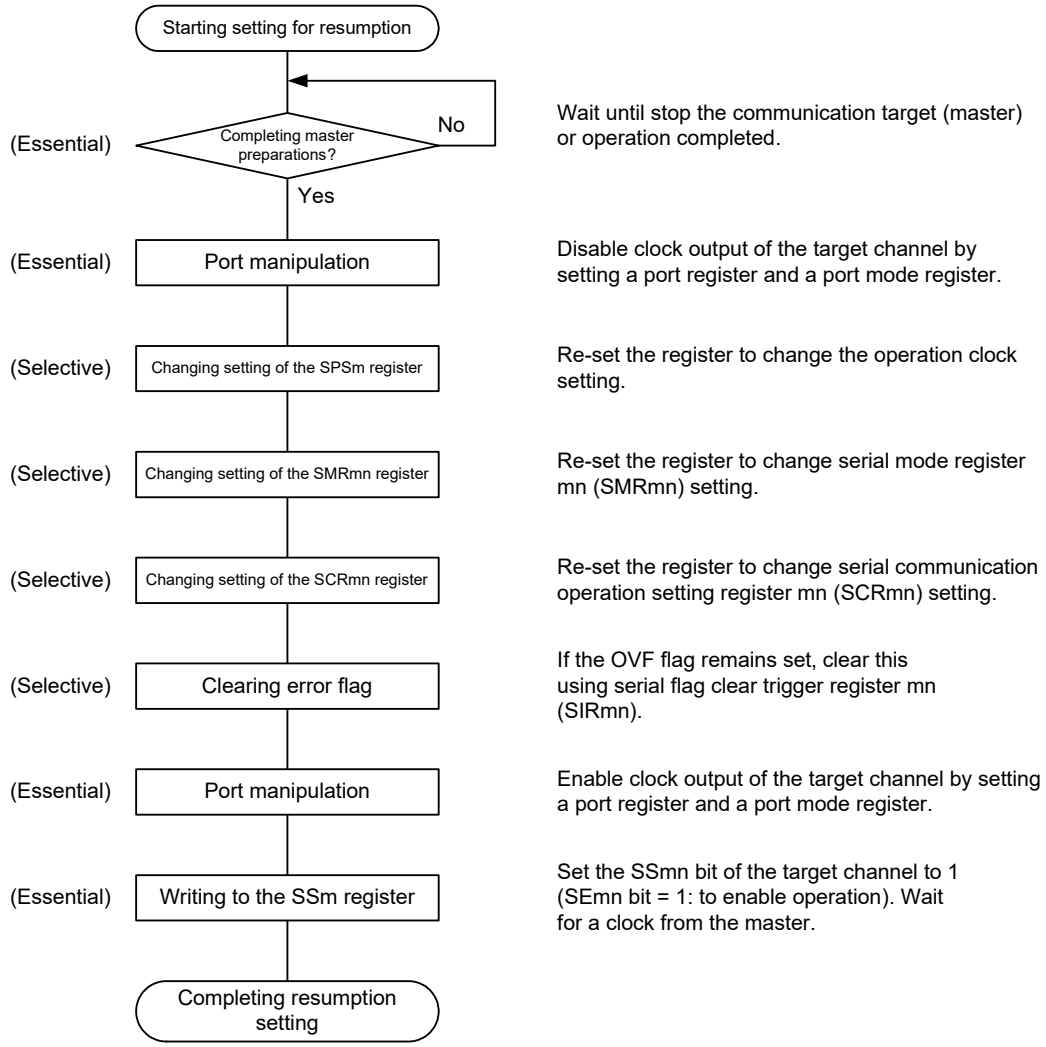


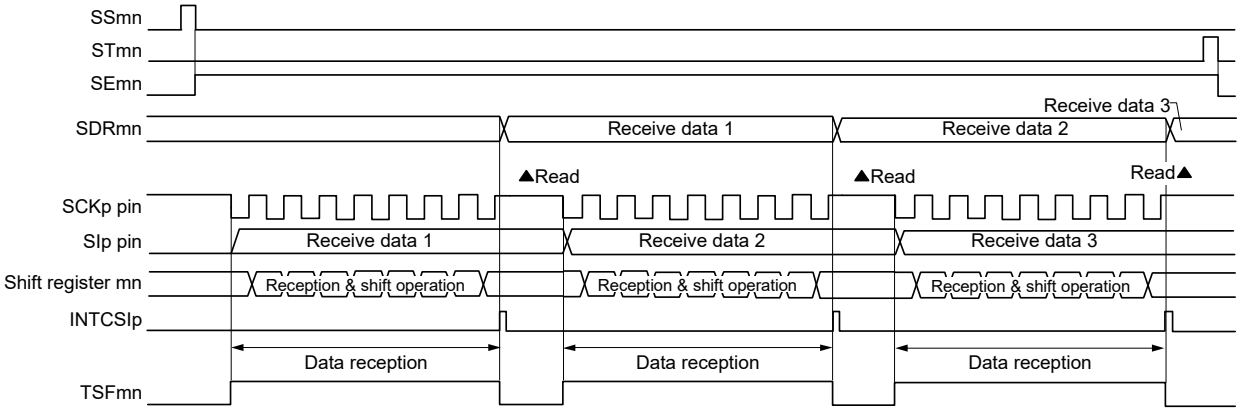
Figure 11 - 61 Procedure for Resuming Slave Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

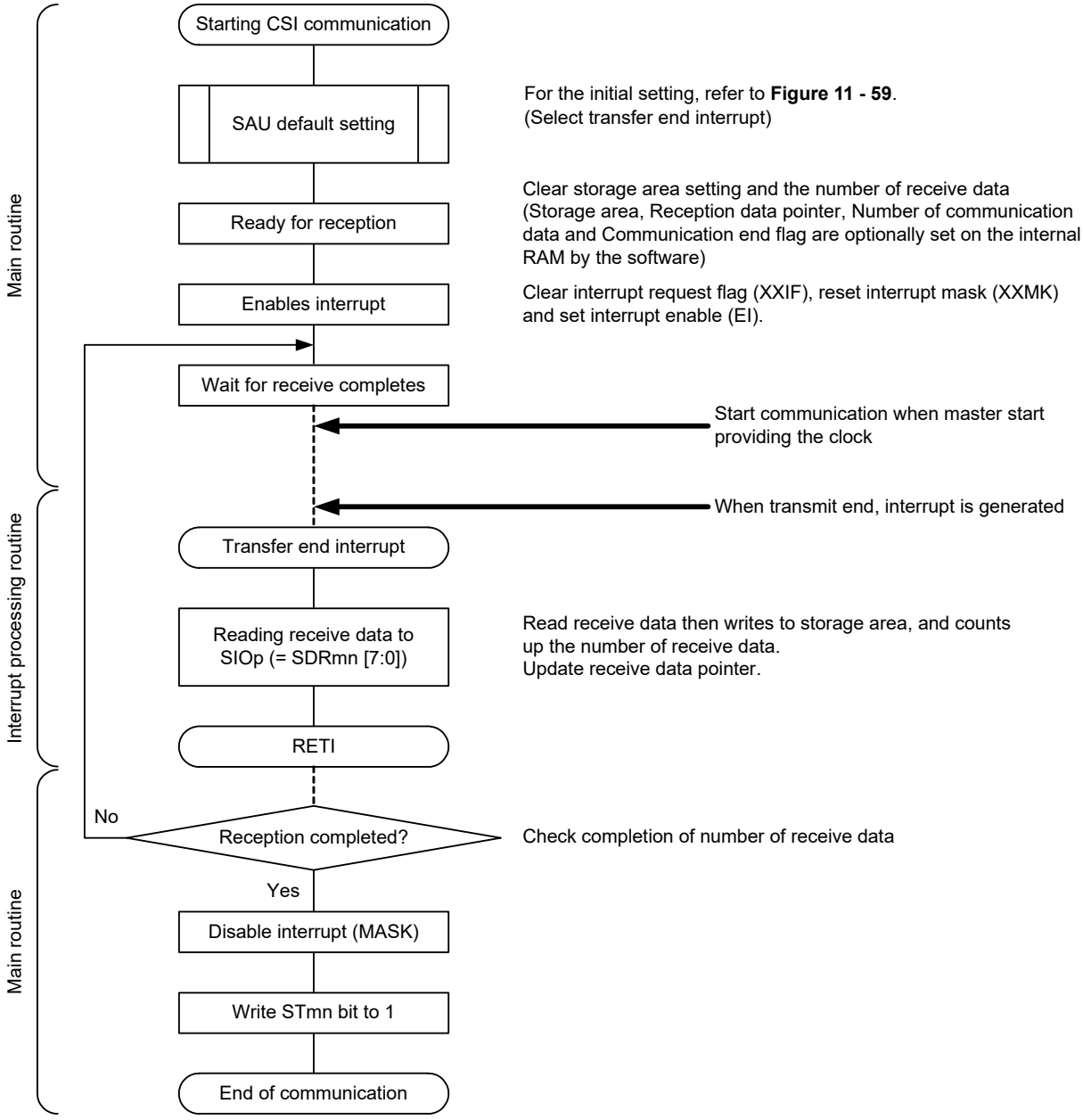
(3) Processing flow (in single-reception mode)

Figure 11 - 62 Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 11 - 63 Flowchart of Slave Reception (in Single-Reception Mode)



11.5.6 Slave transmission/reception

Slave transmission/reception is that the R9A02G015 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	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 (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2.	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note 1. Because the external serial clock input to the SCK00 and SCK01 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 in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)).

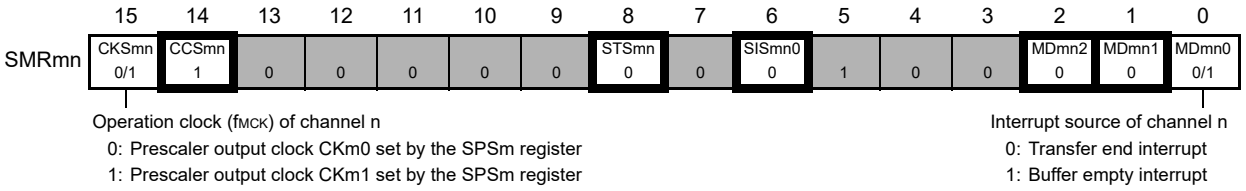
Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

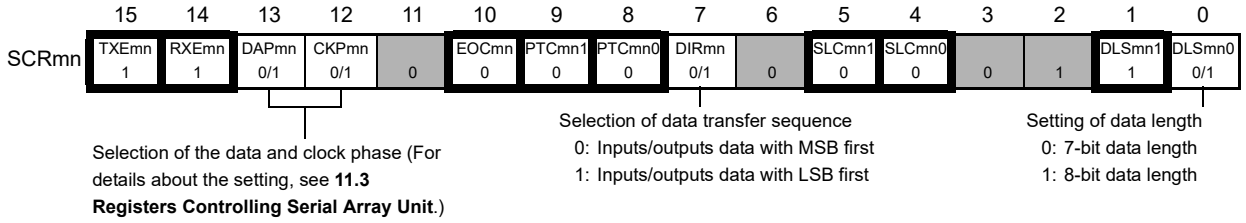
(1) Register setting

Figure 11 - 64 Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01)

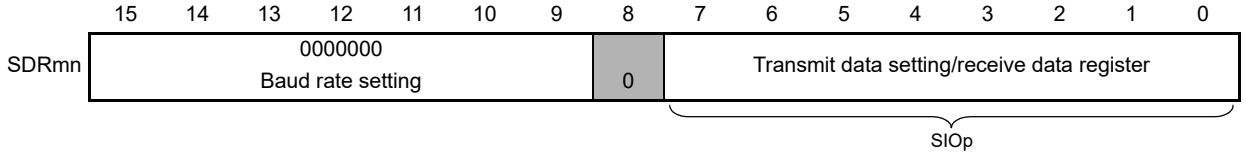
(a) Serial mode register mn (SMRmn)



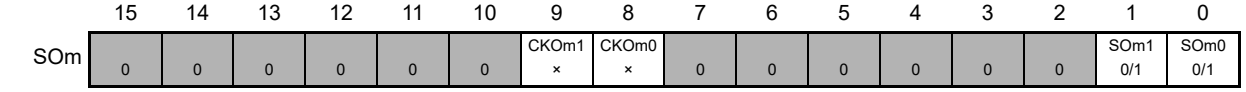
(b) Serial communication operation setting register mn (SCRmn)



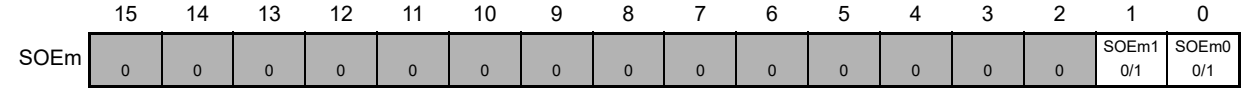
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



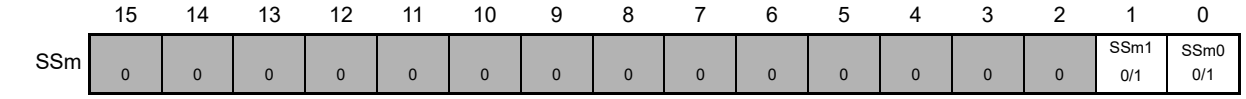
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.



(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the CSI master transmission/reception mode

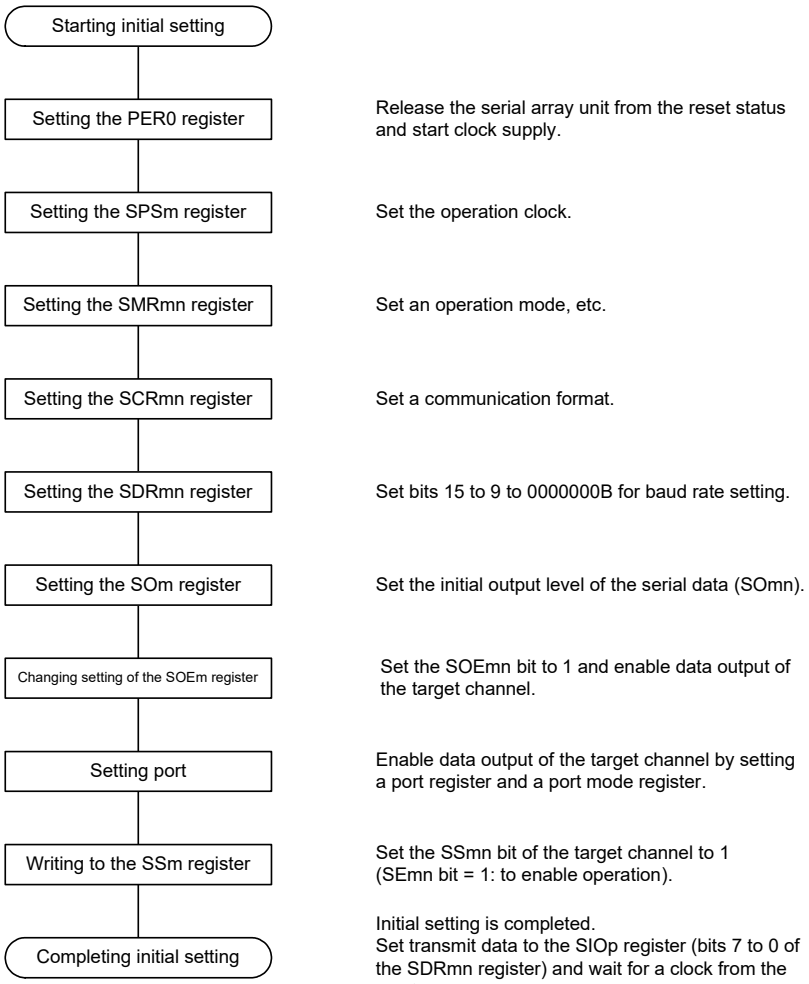
▒: Setting disabled (set to the initial value)

×: 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

Figure 11 - 65 Initial Setting Procedure for Slave Transmission/Reception



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Figure 11 - 66 Procedure for Stopping Slave Transmission/Reception

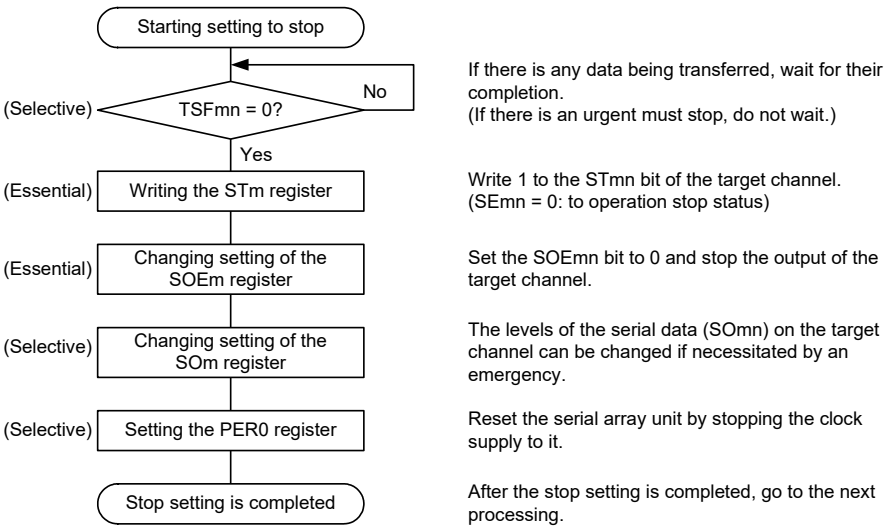
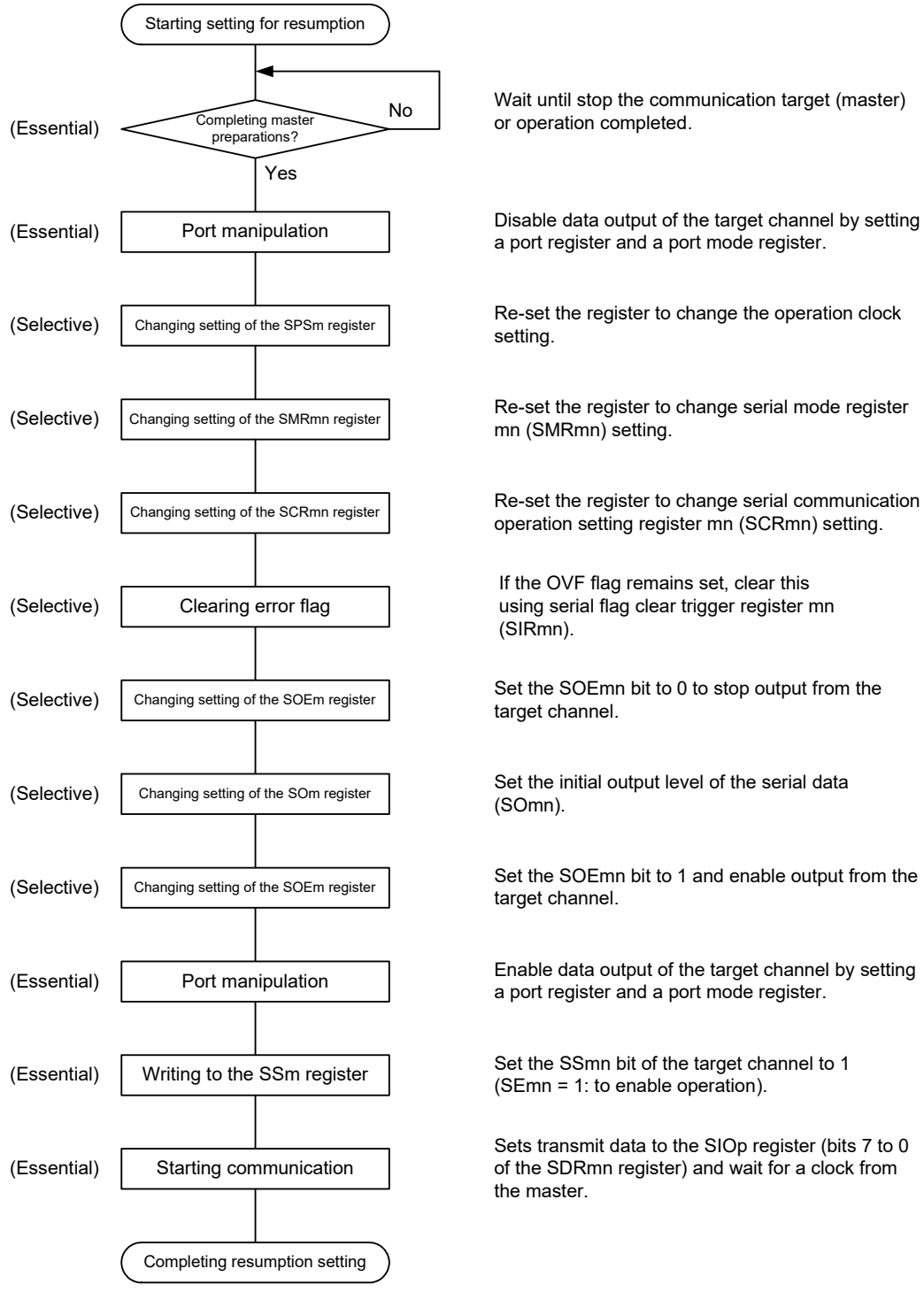


Figure 11 - 67 Procedure for Resuming Slave Transmission/Reception

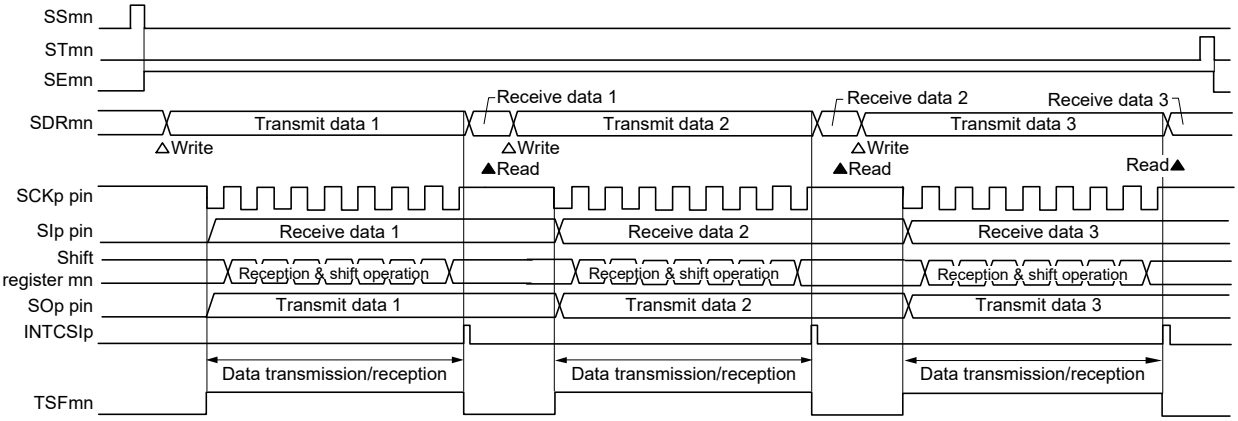


Caution 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

Caution 2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

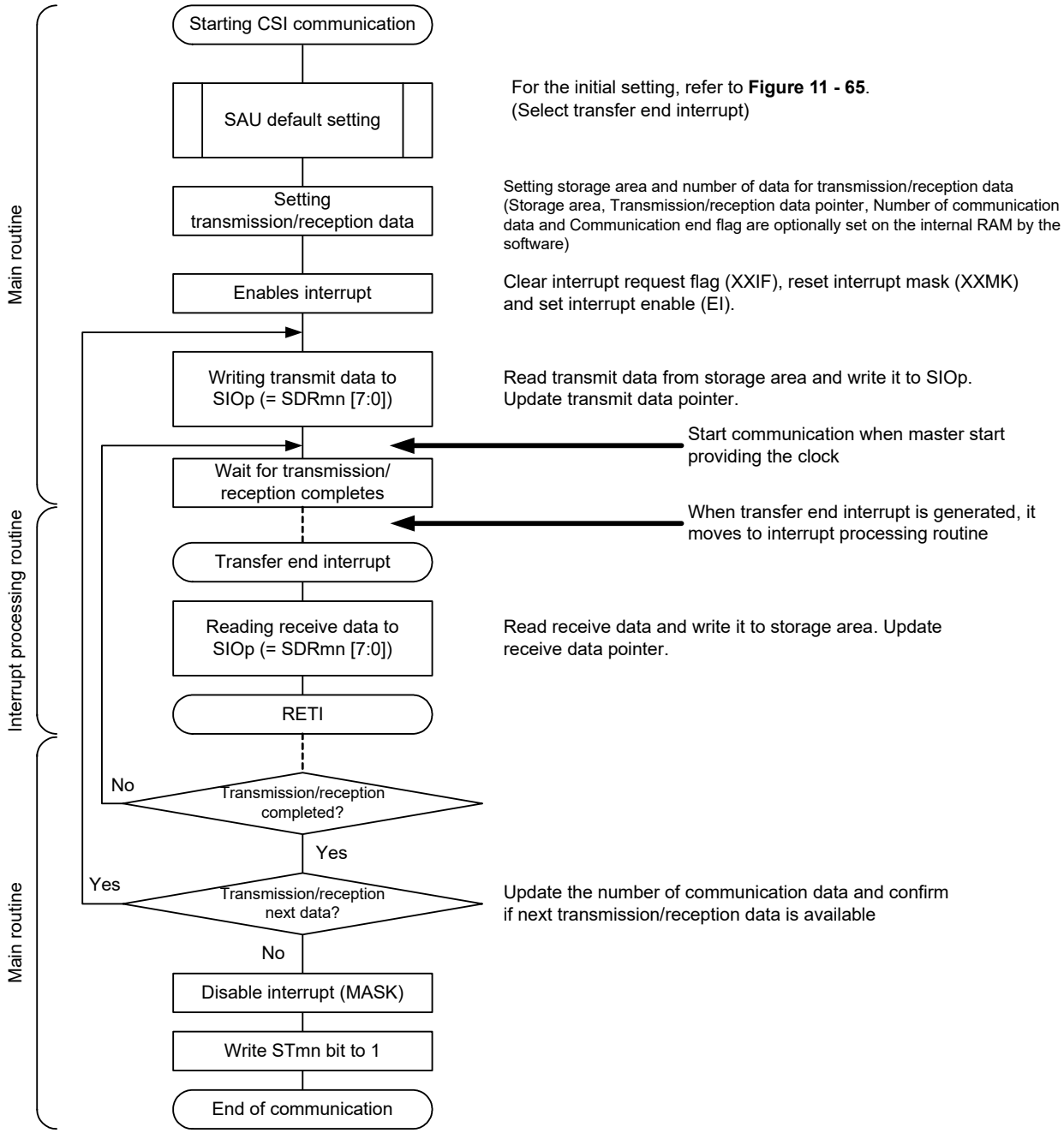
(3) Processing flow (in single-transmission/reception mode)

Figure 11 - 68 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

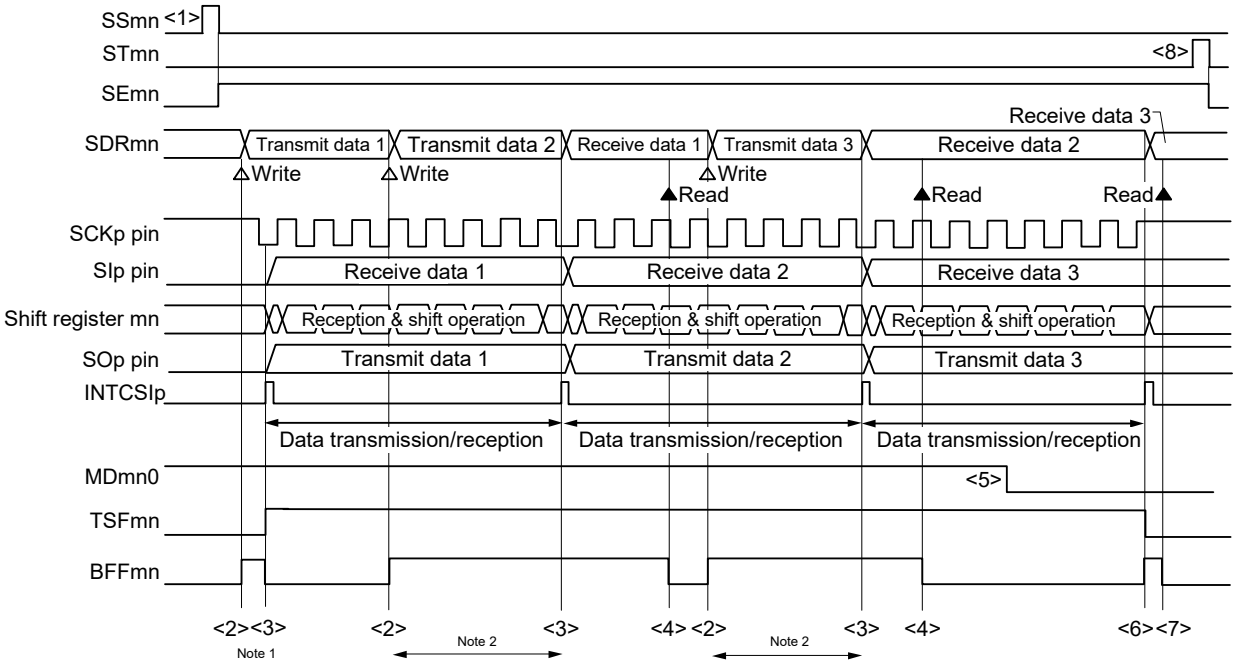
Figure 11 - 69 Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11 - 70 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

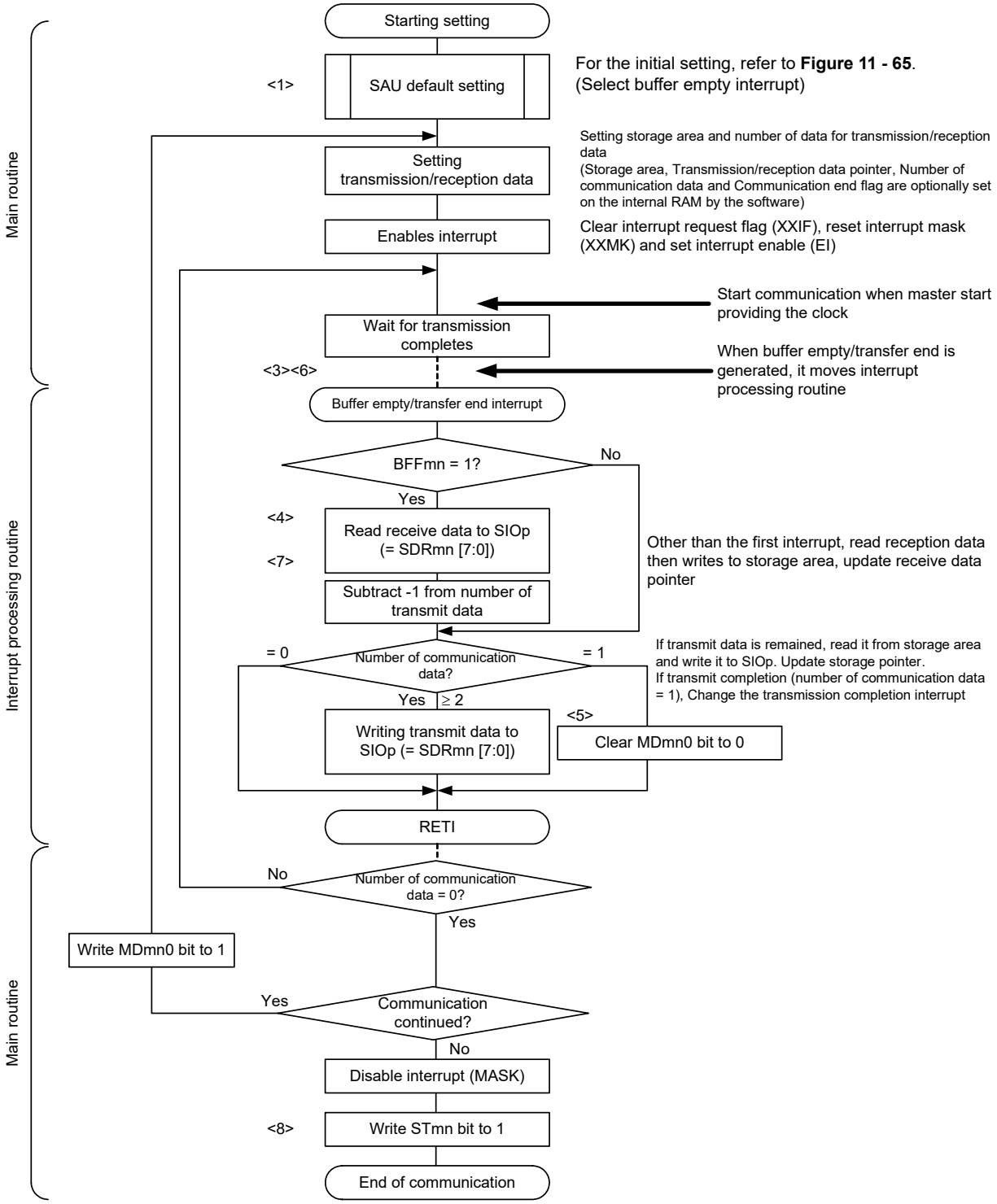
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 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.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 11 - 71 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 11 - 71 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 11 - 70 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

11.5.7 SNOOZE mode function

SNOOZE mode makes CSI operate reception by SCKp pin input detection while the STOP mode. Normally CSI stops communication in the STOP mode. But, using the SNOOZE mode makes reception CSI operate unless the CPU operation by detecting SCKp pin input. Only the following channels can be set to the SNOOZE mode.

• CSI00

When using the CSI in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 11 - 73** and **Figure 11 - 75 Flowchart of SNOOZE Mode Operation**).

• When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SSm0 bit of serial channel start register m (SSm) to 1.

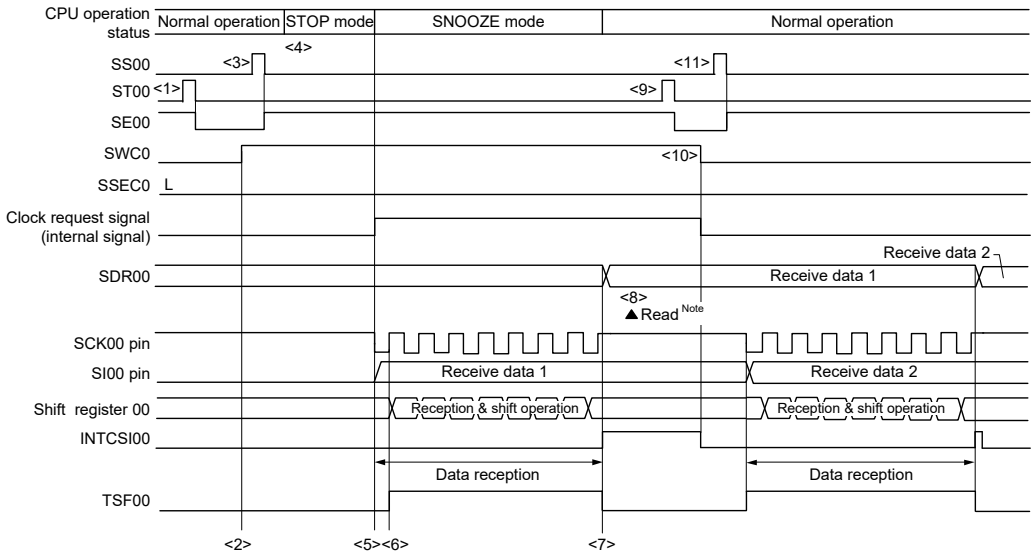
The CPU shifts to the SNOOZE mode on detecting the valid edge of the SCKp signal following a transition to the STOP mode. A CSIp starts reception on detecting input of the serial clock on the SCKp pin.

Caution 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock (f_{IH}) is selected for f_{CLK}.

Caution 2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 11 - 72 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

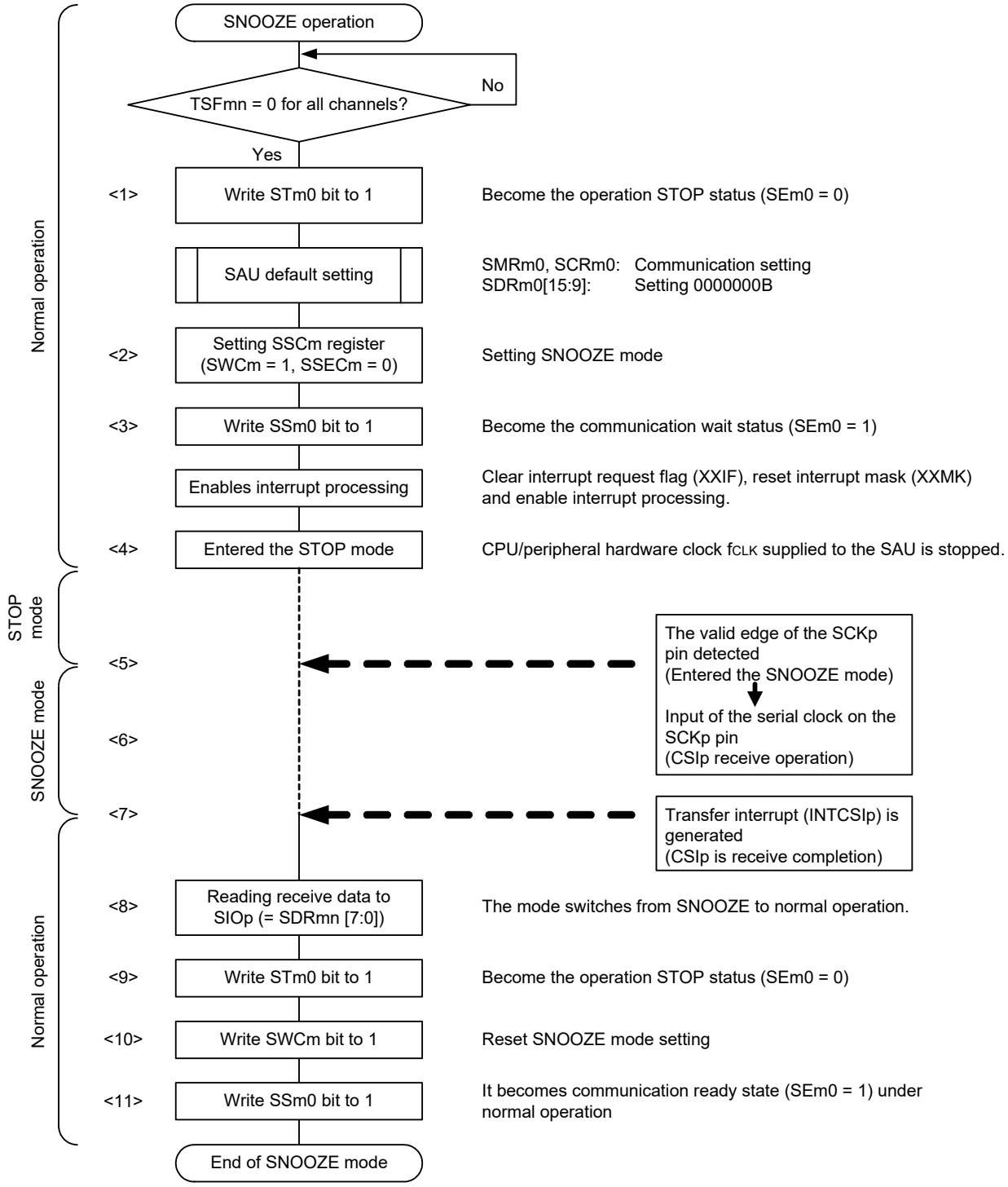
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Caution 2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 11 - 73 Flowchart of SNOOZE Mode Operation (once startup).

Remark 2. m = 0; p = 00

Figure 11 - 73 Flowchart of SNOOZE Mode Operation (once startup)

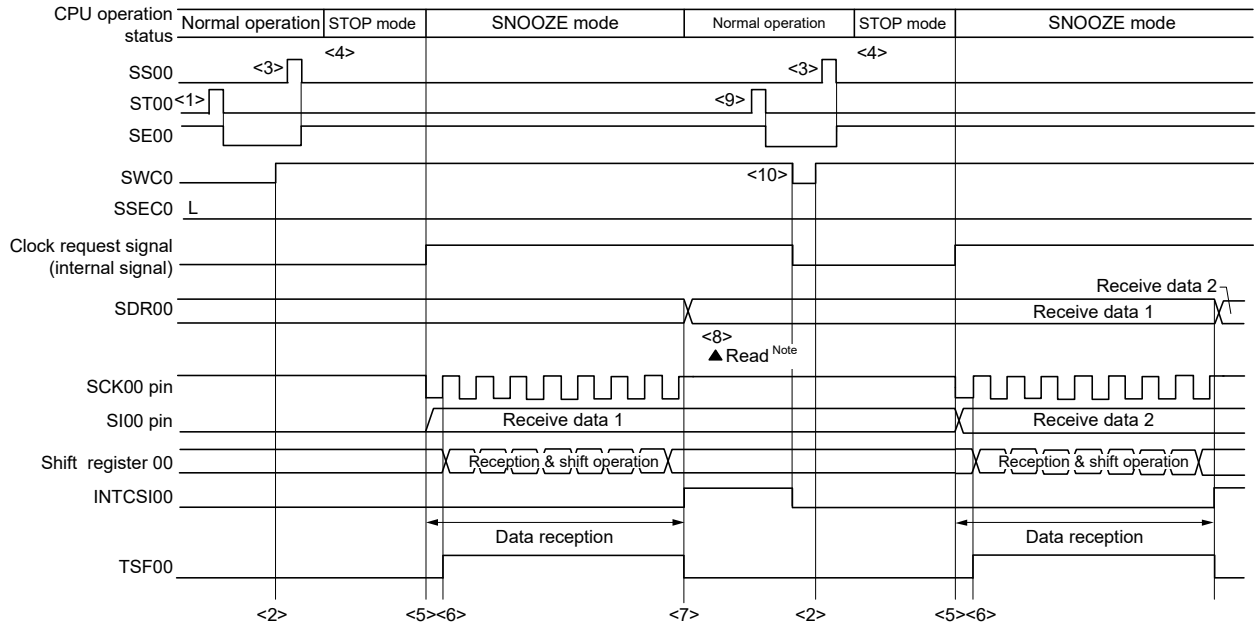


Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 11 - 72 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m = 0; p = 00

(2) SNOOZE mode operation (continuous startup)

Figure 11 - 74 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

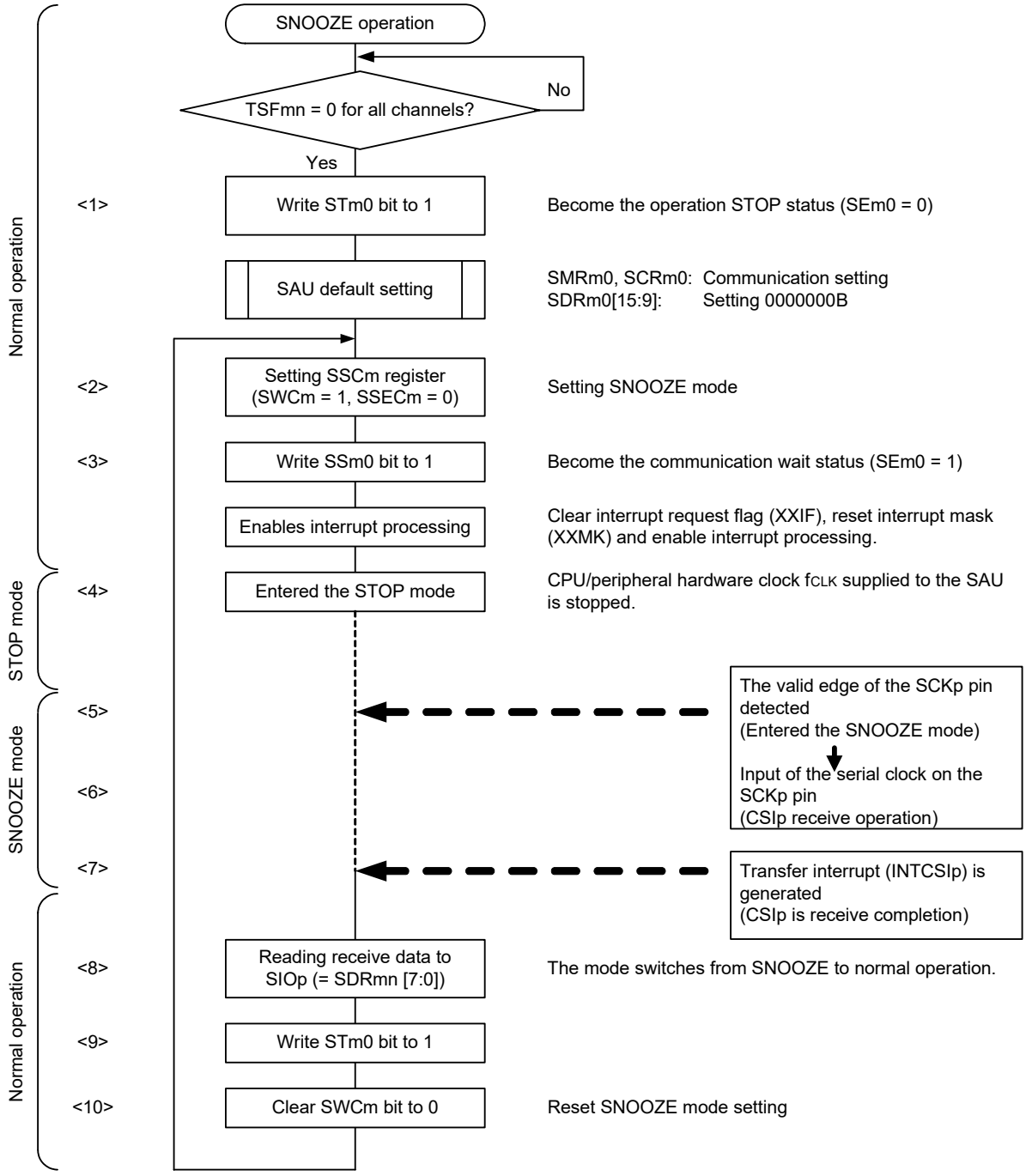
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE release).

Caution 2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 11 - 75 Flowchart of SNOOZE Mode Operation (continuous startup).

Remark 2. m = 0; p = 00

Figure 11 - 75 Flowchart of SNOOZE Mode Operation (continuous startup)



Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 11 - 74 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m = 0; p = 00

11.5.8 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI01) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master} Note [Hz]

Note The permissible maximum transfer clock frequency is f_{MCK}/6.

Remark The value of SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11 - 2 Selection of Operation Clock For 3-Wire Serial I/O

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

11.5.9 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01) communication is described in Figure 11 - 76.

Figure 11 - 76 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn 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.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

11.6 Operation of UART (UART0) Communication

This is a start-stop synchronization 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).

[Data transmission/reception]
- Data length of 7, 8, or 9 bits
- Select the MSB/LSB first
- Level setting of transmit/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/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]
- Framing error, parity error, or overrun error

In addition, UARTs of the following channels supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only the following UART can be specified.
- UART0

UART0 uses channels 0 and 1 of SAU0.

• 32-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01

Select any function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, these channels cannot be used for CSI00 and IIC00.

Caution 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 **11.6.1**.)
• UART reception (See **11.6.2**.)

11.6.1 UART transmission

UART transmission is an operation to transmit data from the R9A02G015 microcontroller to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0
Target channel	Channel 0 of SAU0
Pins used	TxD0
Interrupt	INTST0
	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 Note 1
Transfer rate	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] Note 2
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 bits
Data direction	MSB or LSB first

Note 1. Only the following UARTs can be specified for the 9-bit data length.
• UART0

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)). For UART transmission, use the high-speed system clock or high-speed on-chip oscillator.

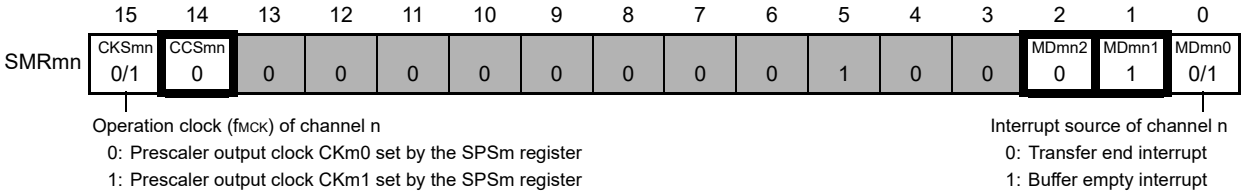
Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

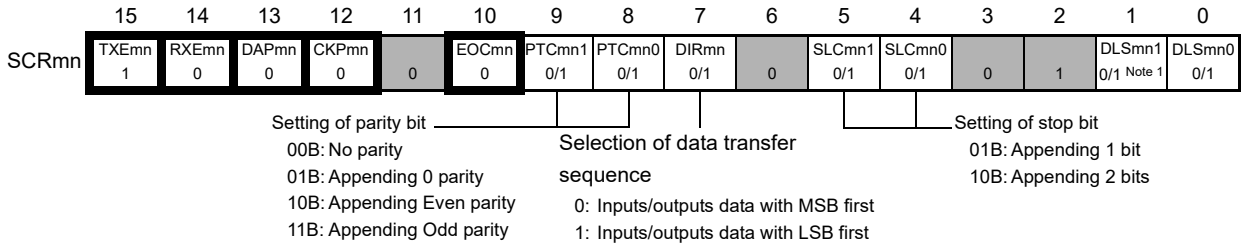
(1) Register setting

Figure 11 - 77 Example of Contents of Registers for UART Transmission of UART (UART0) (1/2)

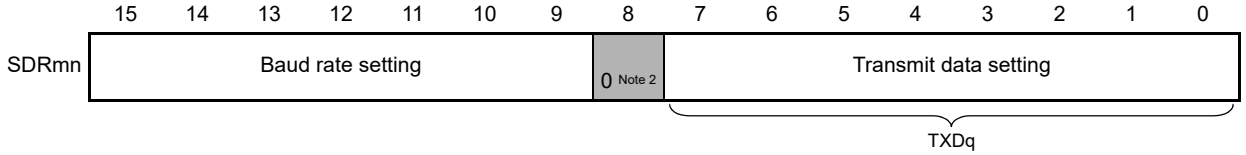
(a) Serial mode register mn (SMRmn)



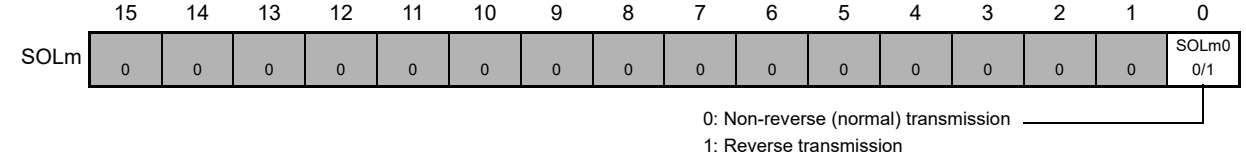
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



(d) Serial output level register m (SOLm)... Sets only the bits of the target channel.



Note 1. Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.

Note 2. When performing 9-bit communication, bits 0 to 8 of the SDRm0 register are used to specify the transmission data.

- UART0

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0), mn = 00

Remark 2. ■: Setting is fixed in the UART transmission mode

▒: Setting disabled (set to the initial value)

×: 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

Figure 11 - 78 Example of Contents of Registers for UART Transmission of UART (UART0) (2/2)

(e) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 0/1 Note

0: Serial data output value is “0”
1: Serial data output value is “1”

(f) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 0/1

(g) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

Note Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0), mn = 00

Remark 2. : Setting disabled (set to the initial value)
×: 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

Figure 11 - 79 Initial Setting Procedure for UART Transmission

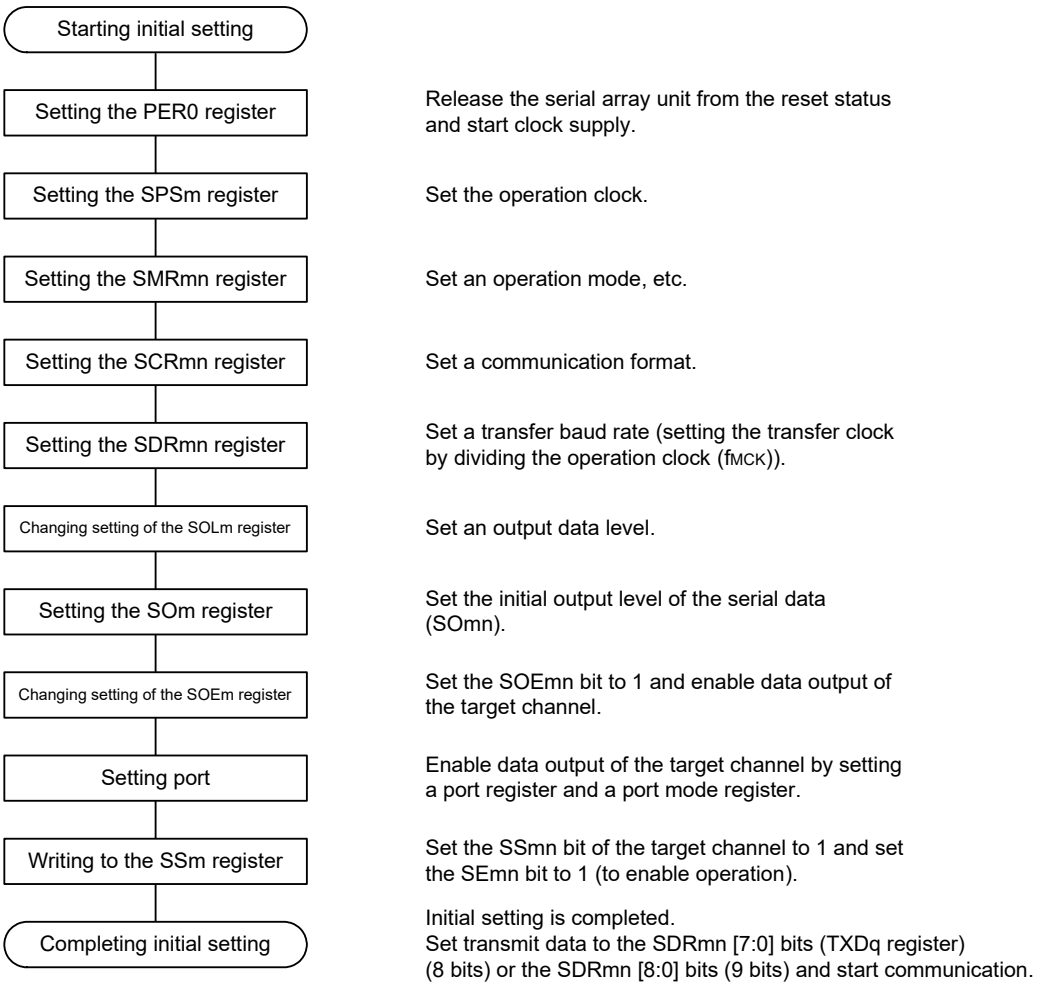


Figure 11 - 80 Procedure for Stopping UART Transmission

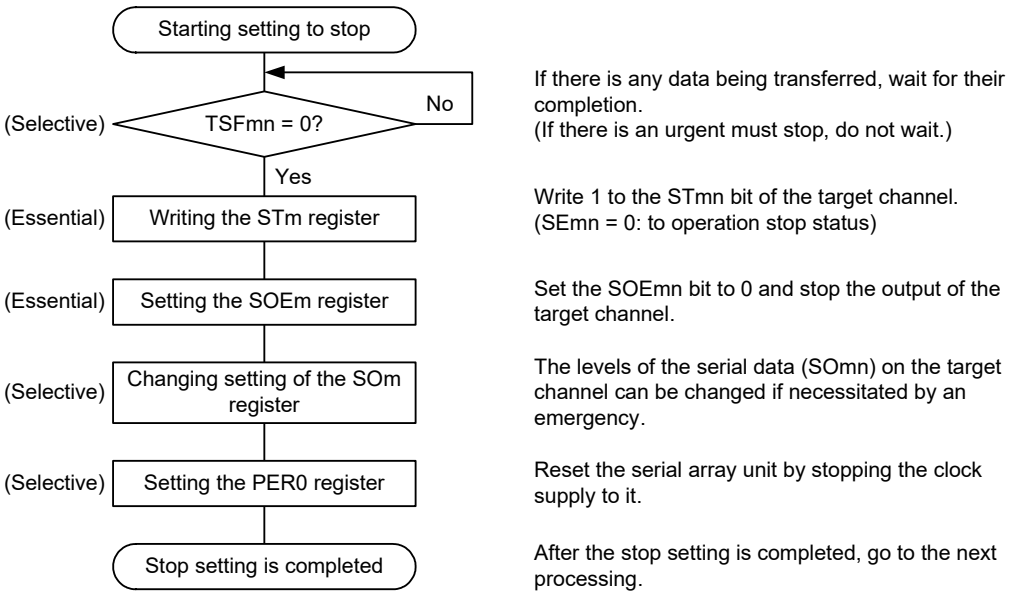
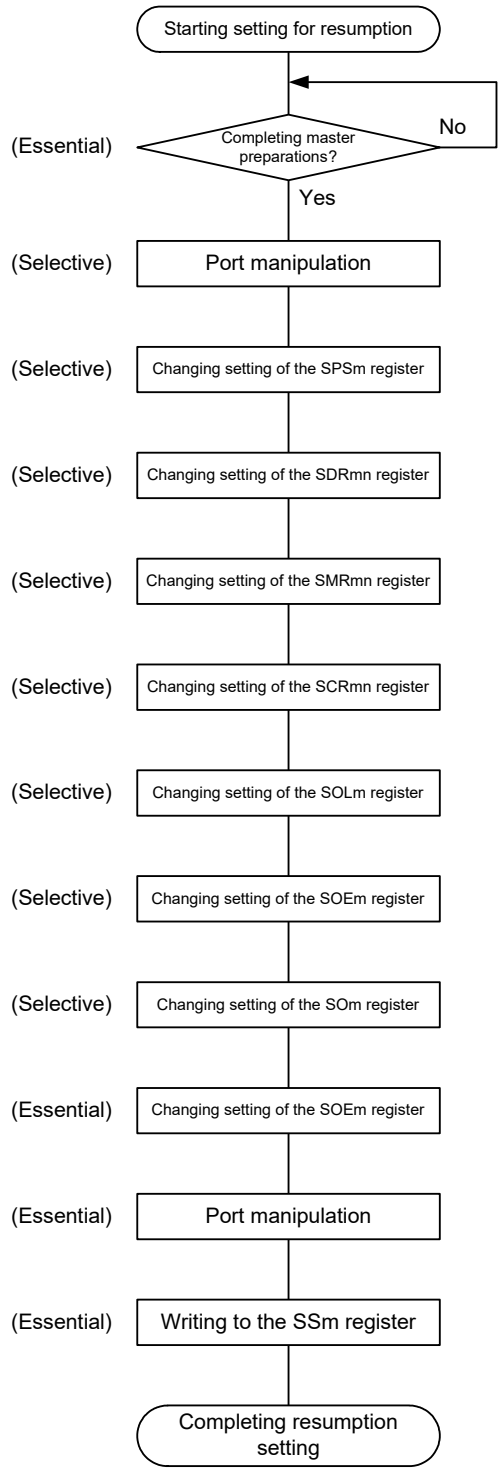


Figure 11 - 81 Procedure for Resuming UART Transmission



Wait until stop the communication target or communication operation completed

Disable data output of the target channel by setting a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change the serial communication operation setting register mn (SCRmn) setting.

Re-set the register to change serial output level register m (SOLm) setting.

Clear the SOEmn bit to 0 and stop output.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output.

Enable data output of the target channel by setting a port register and a port mode register.

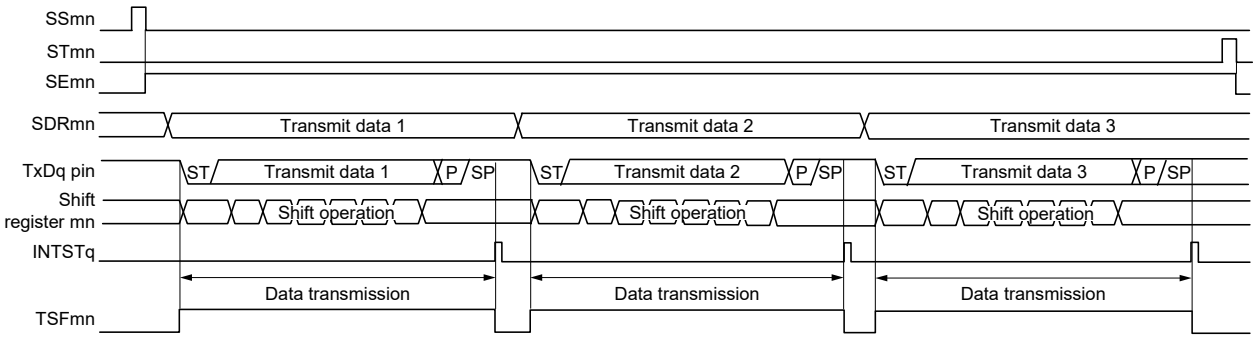
Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

Setting is completed.
Set transmit data to the SDRmn [7:0] bits (TXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

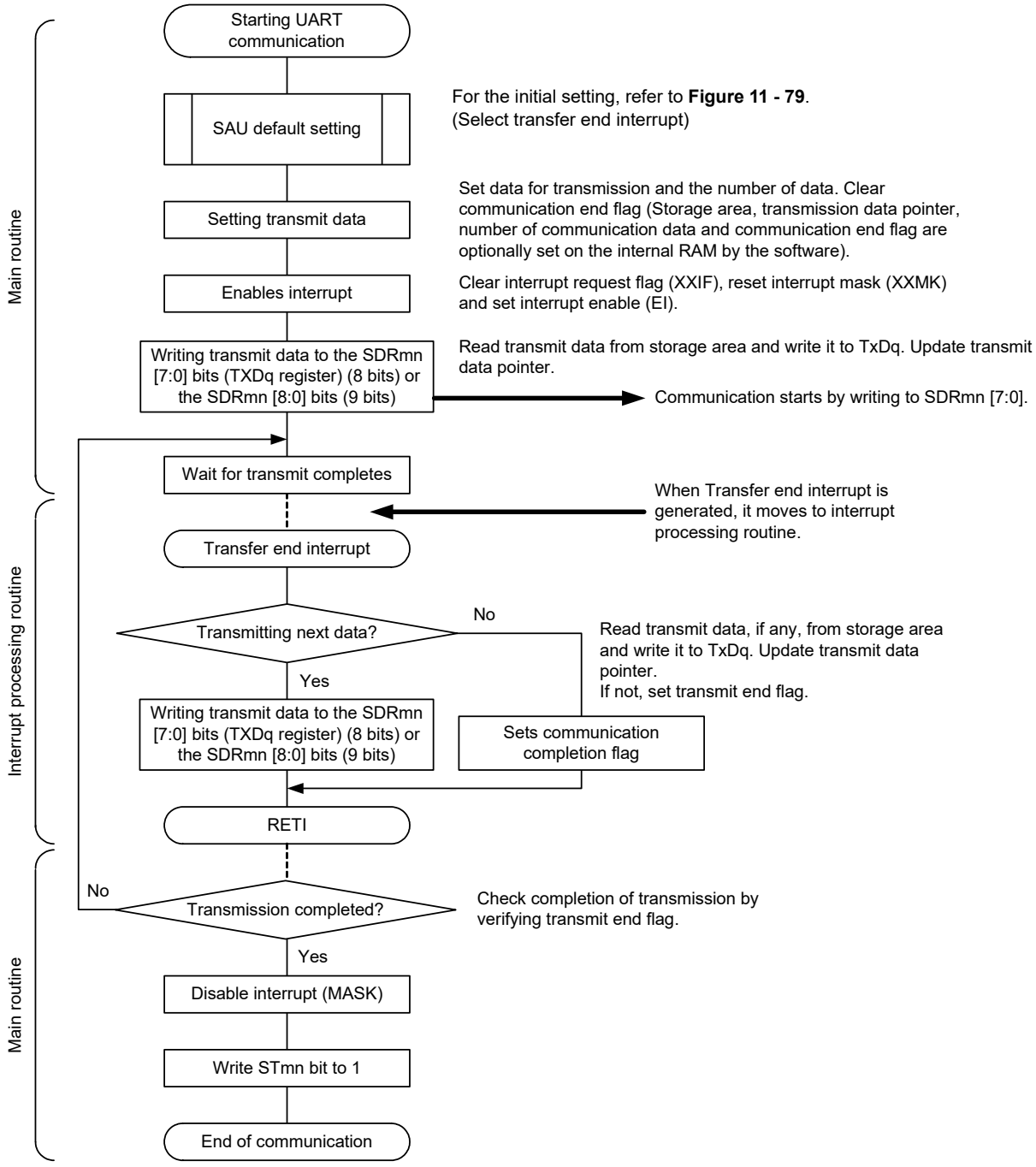
(3) Processing flow (in single-transmission mode)

Figure 11 - 82 Timing Chart of UART Transmission (in Single-Transmission Mode)



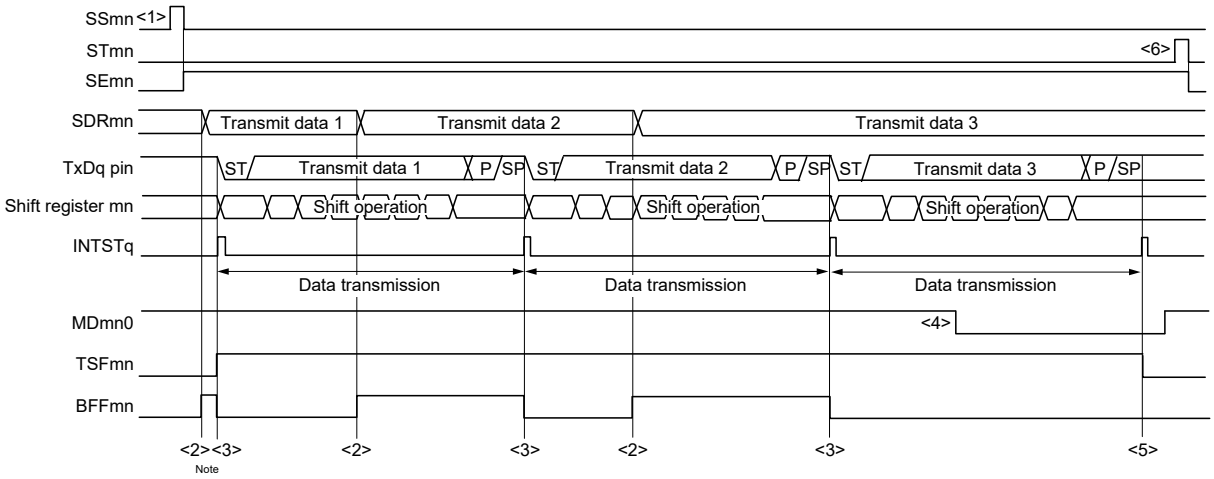
Remark m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0), mn = 00

Figure 11 - 83 Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 11 - 84 Timing Chart of UART Transmission (in Continuous Transmission Mode)

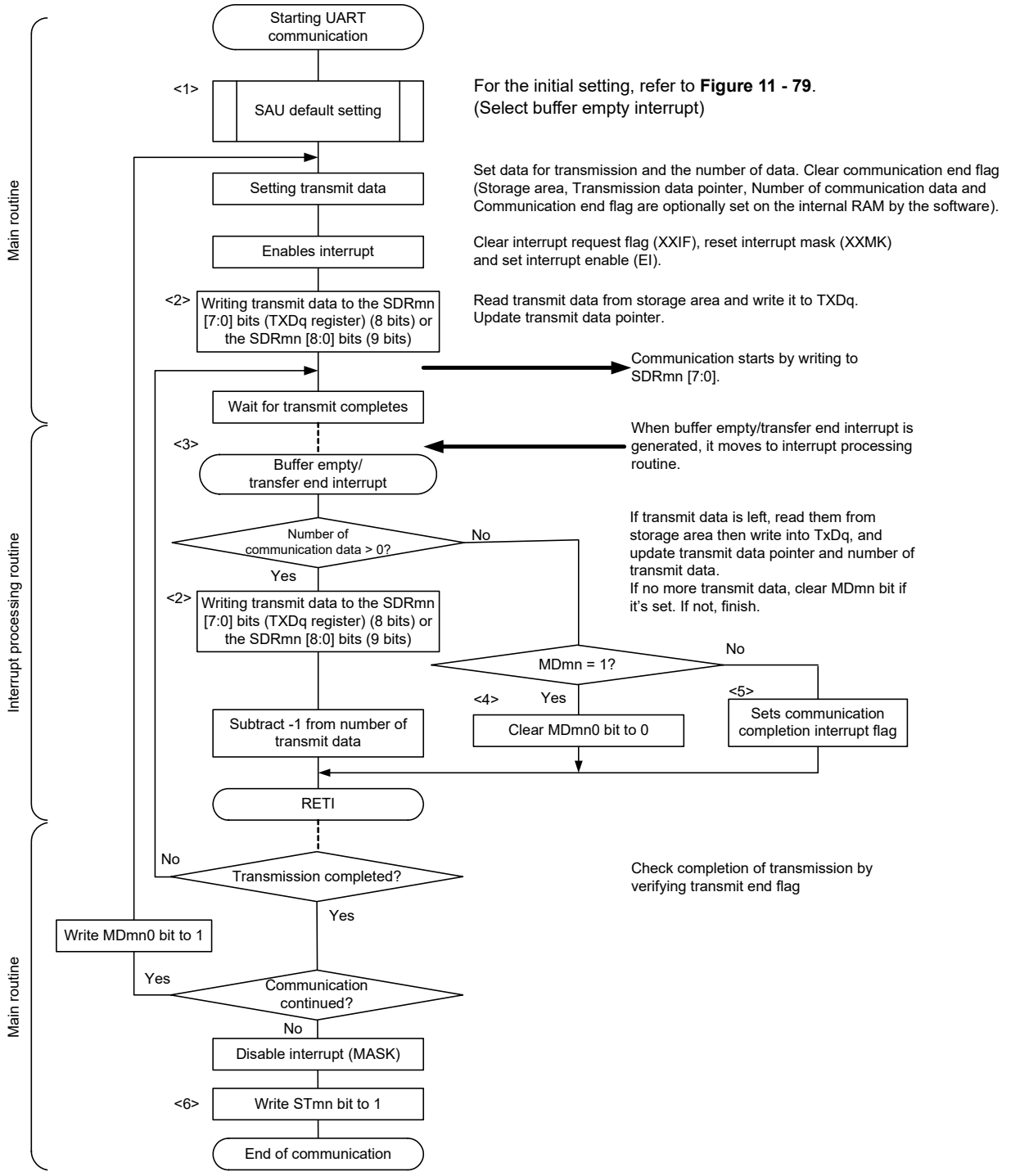


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0), mn = 00

Figure 11 - 85 Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 11 - 84 Timing Chart of UART Transmission (in Continuous Transmission Mode).

11.6.2 UART reception

UART reception is an operation wherein the R9A02G015 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 SMR register of both the odd- and even-numbered channels must be set.

UART	UART0
Target channel	Channel 1 of SAU0
Pins used	RxD0
Interrupt	INTSR0
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error interrupt	INTSRE0
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)
Transfer data length	7, 8 or 9 bits Note 1
Transfer rate Note 2	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)
Parity bit	The following selectable • No parity bit (no parity check) • Appending 0 parity (no parity check) • Appending even parity • Appending odd parity
Stop bit	Appending 1 bit
Data direction	MSB or LSB first

Note 1. Only the following UARTs can be specified for the 9-bit data length.
• UART0

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)). For UART reception, use the high-speed system clock or high-speed on-chip oscillator.

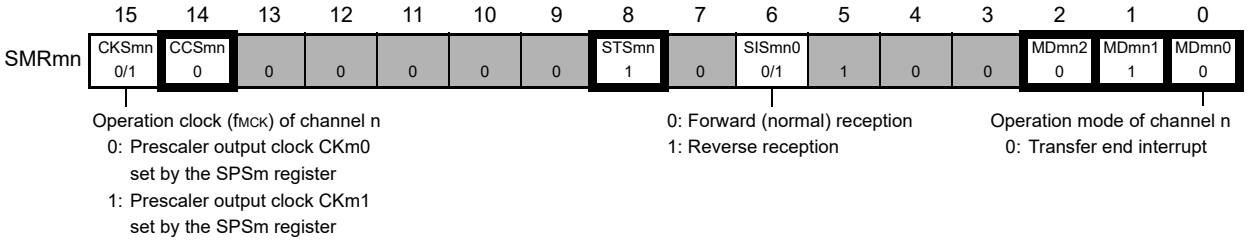
Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01

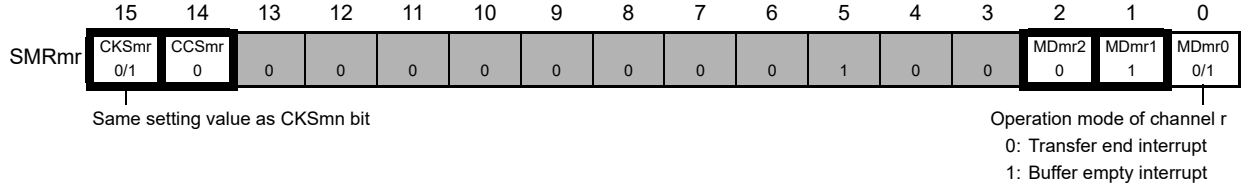
(1) Register setting

Figure 11 - 86 Example of Contents of Registers for UART Reception of UART (UART0) (1/2)

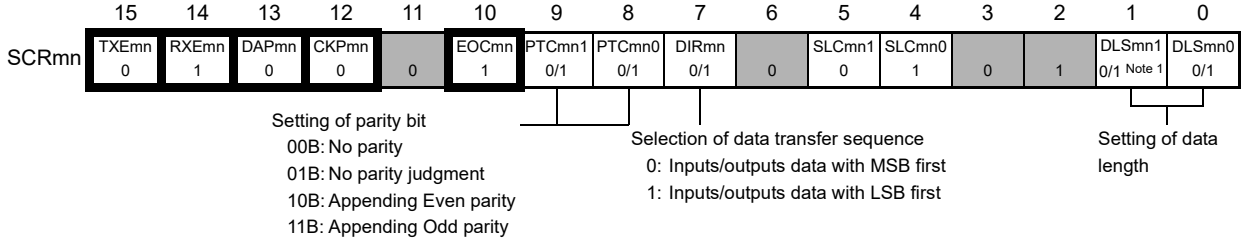
(a) Serial mode register mn (SMRmn)



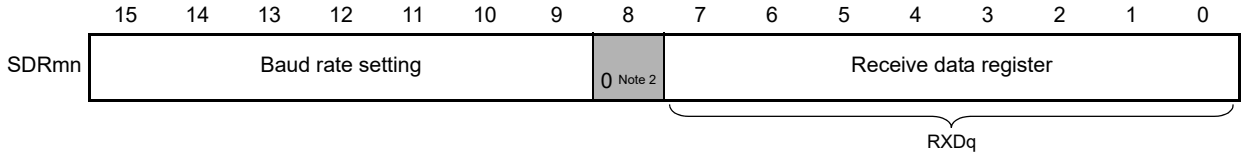
(b) Serial mode register mr (SMRmr)



(c) Serial communication operation setting register mn (SCRmn)



(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



Note 1. Only provided for the SCR01 register. This bit is fixed to 1 for the other registers.

Note 2. When performing 9-bit communication, bits 0 to 8 of the SDRm1 register are used to specify the receive data. Only the following UART can be specified for the 9-bit data length.
• UART0

Caution For the UART reception, be sure to set the SMRmr register of channel r that is to be paired with channel n.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01, r: Channel number (r = n - 1), q: UART number (q = 0)

Remark 2. : Setting is fixed in the UART reception mode,
: Setting disabled (set to the initial value)
×: 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

Figure 11 - 87 Example of Contents of Registers for UART Reception of UART (UART0) (2/2)

(e) Serial output register m (SOm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 ×

(f) Serial output enable register m (SOEm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

(g) Serial channel start register m (SSm)... Sets only the bits of the target channel is 1.

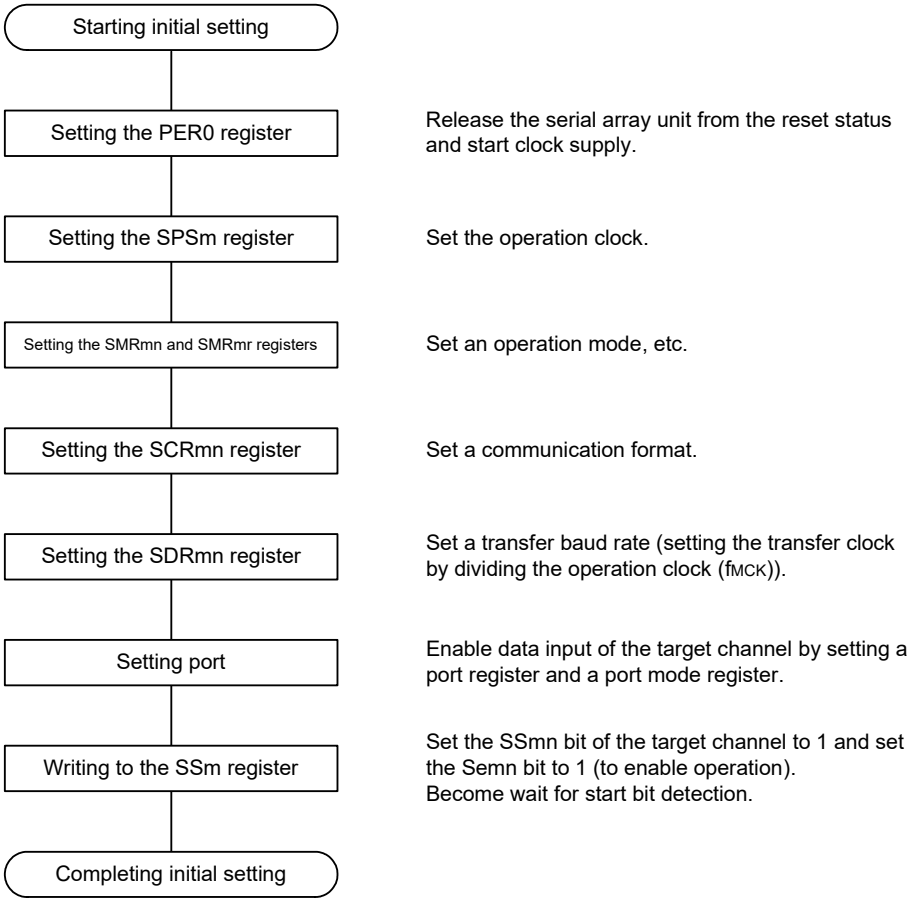
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 ×

Remark 1. m: Unit number (m = 0)

Remark 2. : Setting disabled (set to the initial value)
×: 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

Figure 11 - 88 Initial Setting Procedure for UART Reception



Caution Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more fMCK clocks have elapsed.

Figure 11 - 89 Procedure for Stopping UART Reception

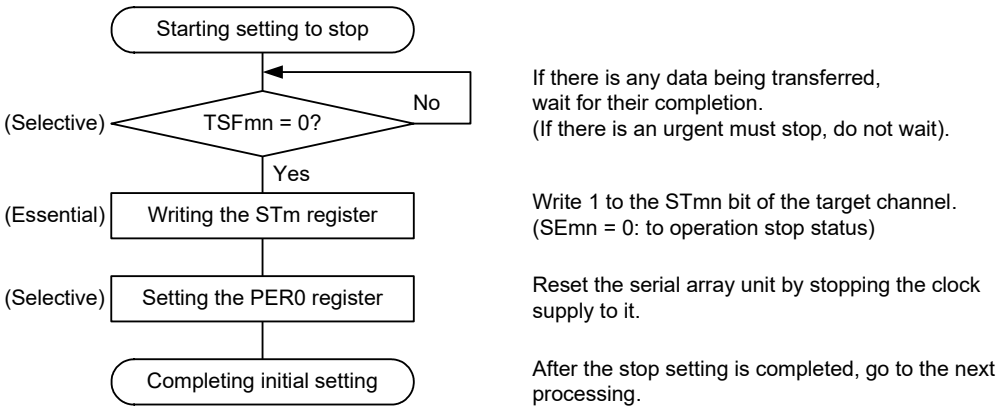
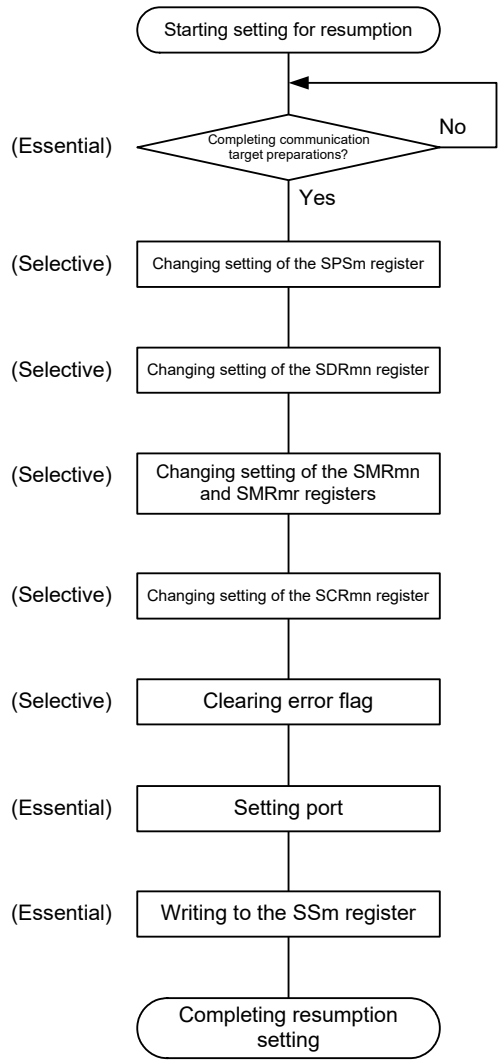


Figure 11 - 90 Procedure for Resuming UART Reception



Stop the target for communication or wait until completes its communication operation.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the registers to change serial mode registers mn, mr (SMRmn, SMRmr) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register mn (SIRmn).

Enable data input of the target channel by setting a port register and a port mode register.

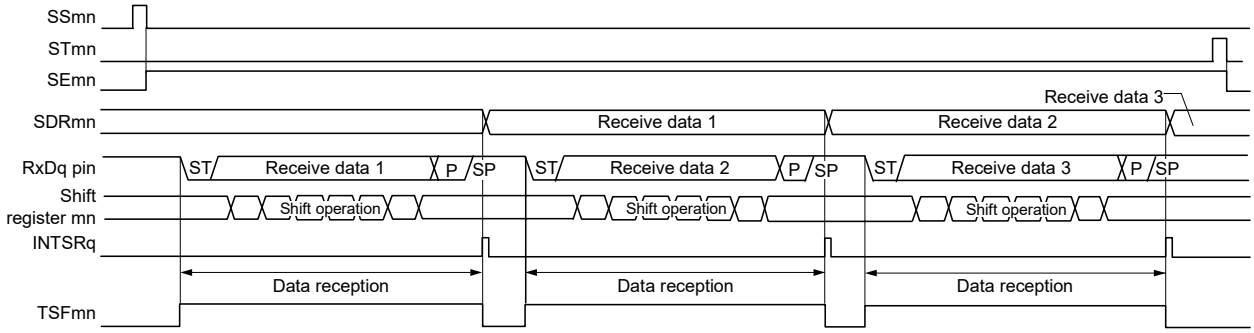
Set the SSmn bit of the target channel to 1 and set the Semn bit to 1 (to enable operation).
Become wait for start bit detection.

Caution After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of f_{MCK}.

Remark If PER0 is rewritten while stopping the communication target and the clock supply is stopped, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

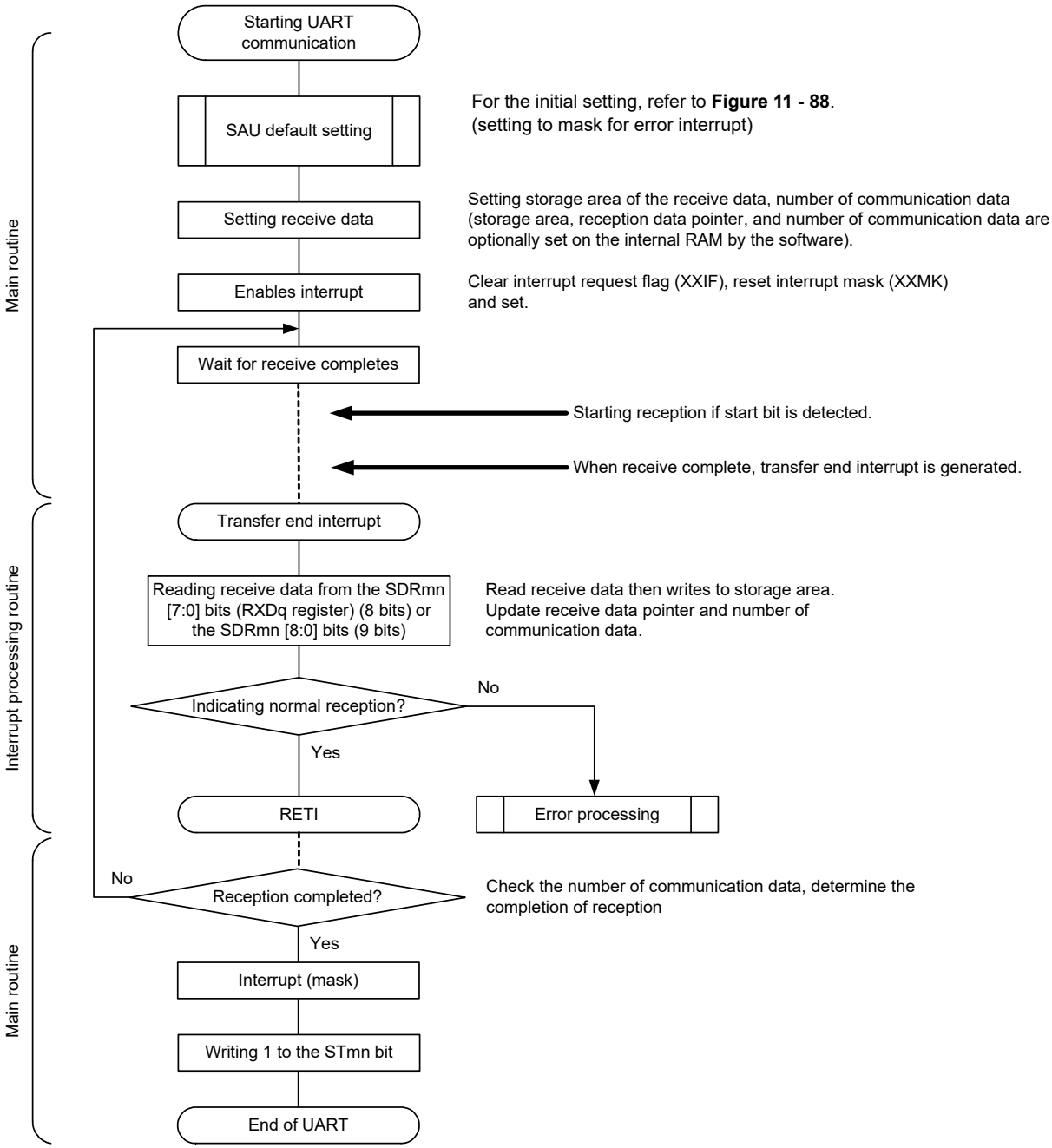
(3) Processing flow

Figure 11 - 91 Timing Chart of UART Reception



Remark m: Unit number (m = 0), n: Channel number (n = 1), mn = 01, r: Channel number (r = n - 1), q: UART number (q = 0)

Figure 11 - 92 Flowchart of UART Reception



11.6.3 SNOOZE mode function

The SNOOZE mode makes the UART perform reception operations upon RxDq pin input detection while in the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception operations without CPU operation. Only the following UART can be specified.

• UART0

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode (See **Figures 11 - 95** and **11 - 97 Flowchart of SNOOZE Mode Operation**).

• In the SNOOZE mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPSm register and bits 15 to 9 of the SDRmn register with reference to Table 11 - 3.

• Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.

• When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm1 bit of serial channel start register m (SSm) to 1.

A UARTq starts reception in SNOOZE mode on detecting input of the start bit on the RxDq pin following a transition of the CPU to the STOP mode.

Caution 1. SNOOZE mode can be used only when the high-speed on-chip oscillator clock (f_{IH}) is selected as f_{CLK}.

Caution 2. The transfer rate in the SNOOZE mode is only 4800 bps.

Caution 3. When SWCm = 1, UARTq can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP 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 SWCm bit has been set to 1, the reception operation is started before the STOP mode is entered**
- **When the reception operation is started while another function is in the SNOOZE mode**
- **When after returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWCm bit is returned to 0**

Caution 4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

Caution 5. When using UART reception to transition from STOP mode to SNOOZE mode, use the high-speed on-chip oscillator.

Table 11 - 3 Baud Rate Setting for UART Reception in SNOOZE Mode

High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn [15:9]	Maximum Permissible Value	Minimum Permissible Value
24 MHz ± 1.0% Note	$f_{CLK}/2^5$	79	1.60%	-2.18%
16 MHz ± 1.0% Note	$f_{CLK}/2^4$	105	2.27%	-1.53%
12 MHz ± 1.0% Note	$f_{CLK}/2^4$	79	1.60%	-2.19%
8 MHz ± 1.0% Note	$f_{CLK}/2^3$	105	2.27%	-1.53%
6 MHz ± 1.0% Note	$f_{CLK}/2^3$	79	1.60%	-2.19%
4 MHz ± 1.0% Note	$f_{CLK}/2^2$	105	2.27%	-1.53%
3 MHz ± 1.0% Note	$f_{CLK}/2^2$	79	1.60%	-2.19%
2 MHz ± 1.0% Note	$f_{CLK}/2$	105	2.27%	-1.54%
1 MHz ± 1.0% Note	f_{CLK}	105	2.27%	-1.57%

Note 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.

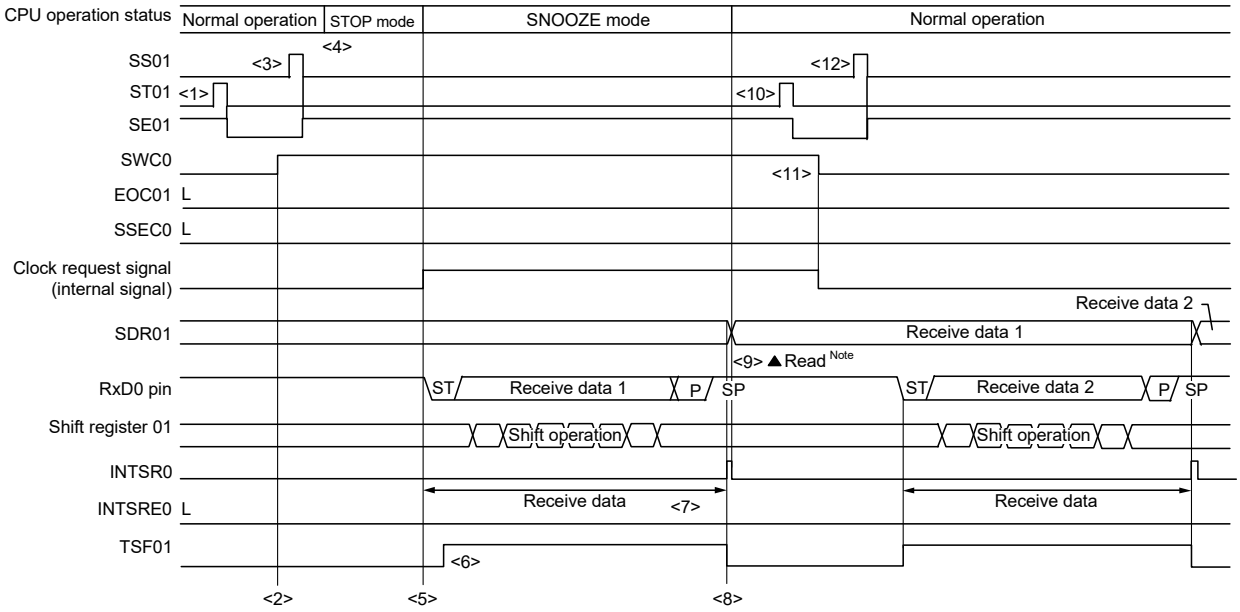
- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value - 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ± 2.0%, perform (Maximum permissible value - 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Remark 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 (EOCm1 = 0, SSECm = 0/1)

Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

Figure 11 - 93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1.

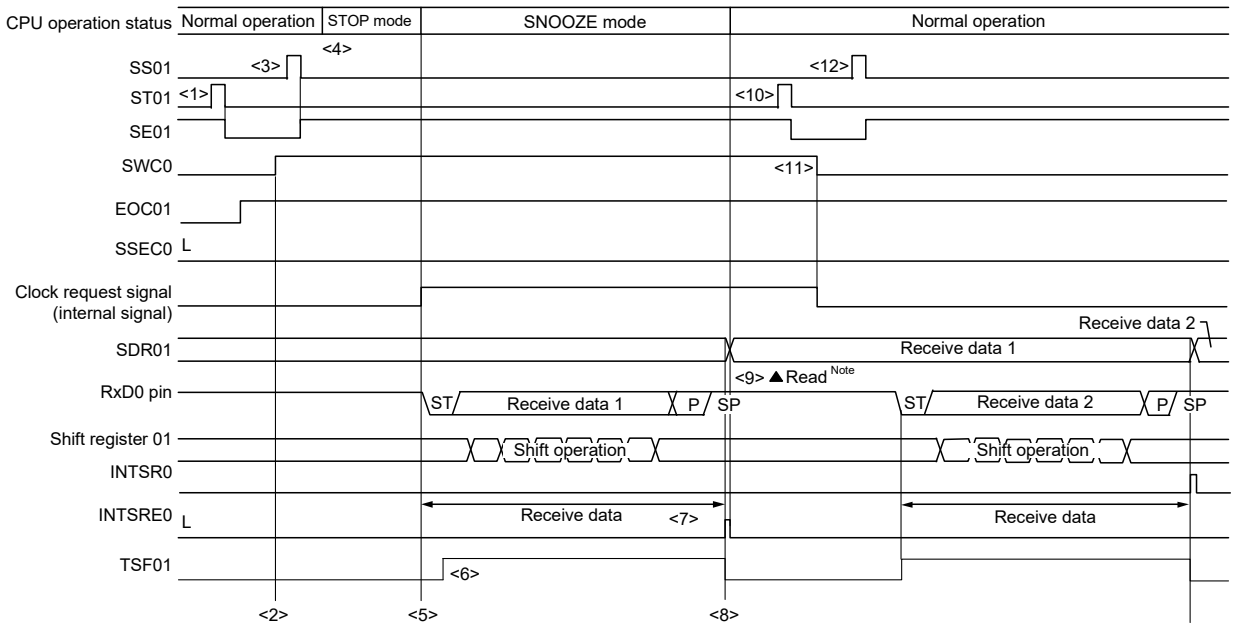
Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).**
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 11 - 95 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

(2) SNOOZE mode operation (EOCm1 = 1, SSECm = 0: Error interrupt (INTSREq) generation is enabled)
Because EOCm1 = 1 and SSECm = 0, an error interrupt (INTSREq) is generated when a communication error occurs.

Figure 11 - 94 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



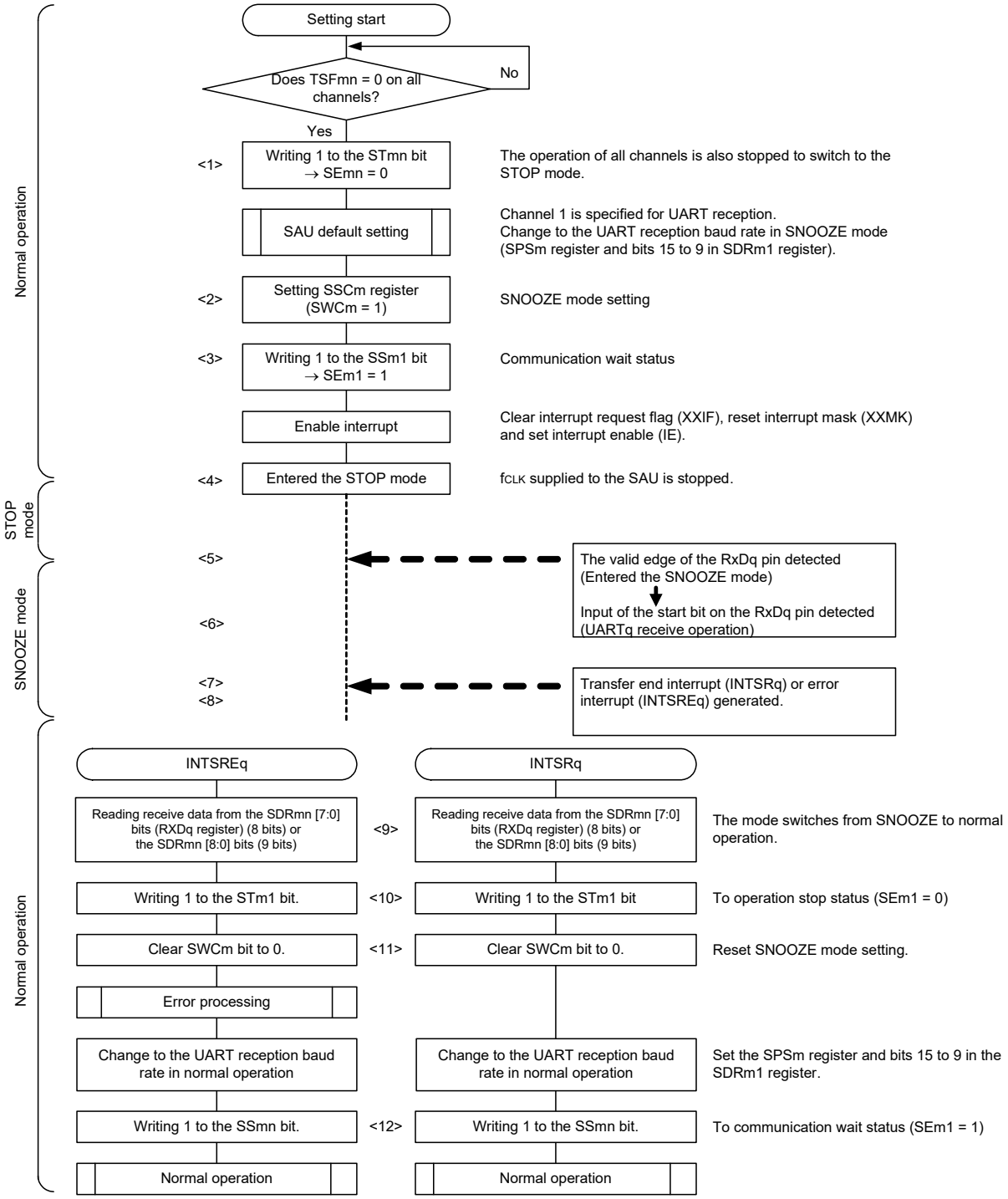
Note Read the received data when SWCm = 1.

Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).**
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 11 - 95 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

Figure 11 - 95 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)



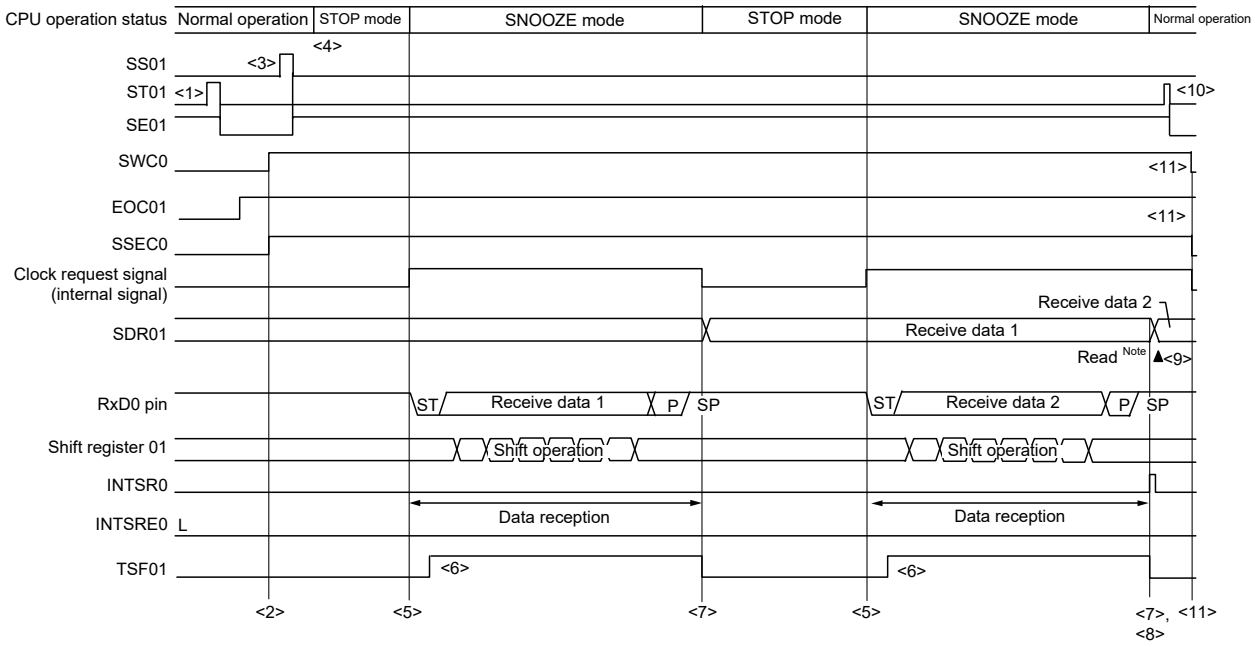
Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 11 - 93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1) and Figure 11 - 94 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

(3) SNOOZE mode operation (EOCm1 = 1, SSECm = 1: Error interrupt (INTSREq) generation is stopped)

Because EOCm1 = 1 and SSECm = 1, an error interrupt (INTSREq) is not generated when a communication error occurs.

Figure 11 - 96 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Only read received data while SWCm = 1.

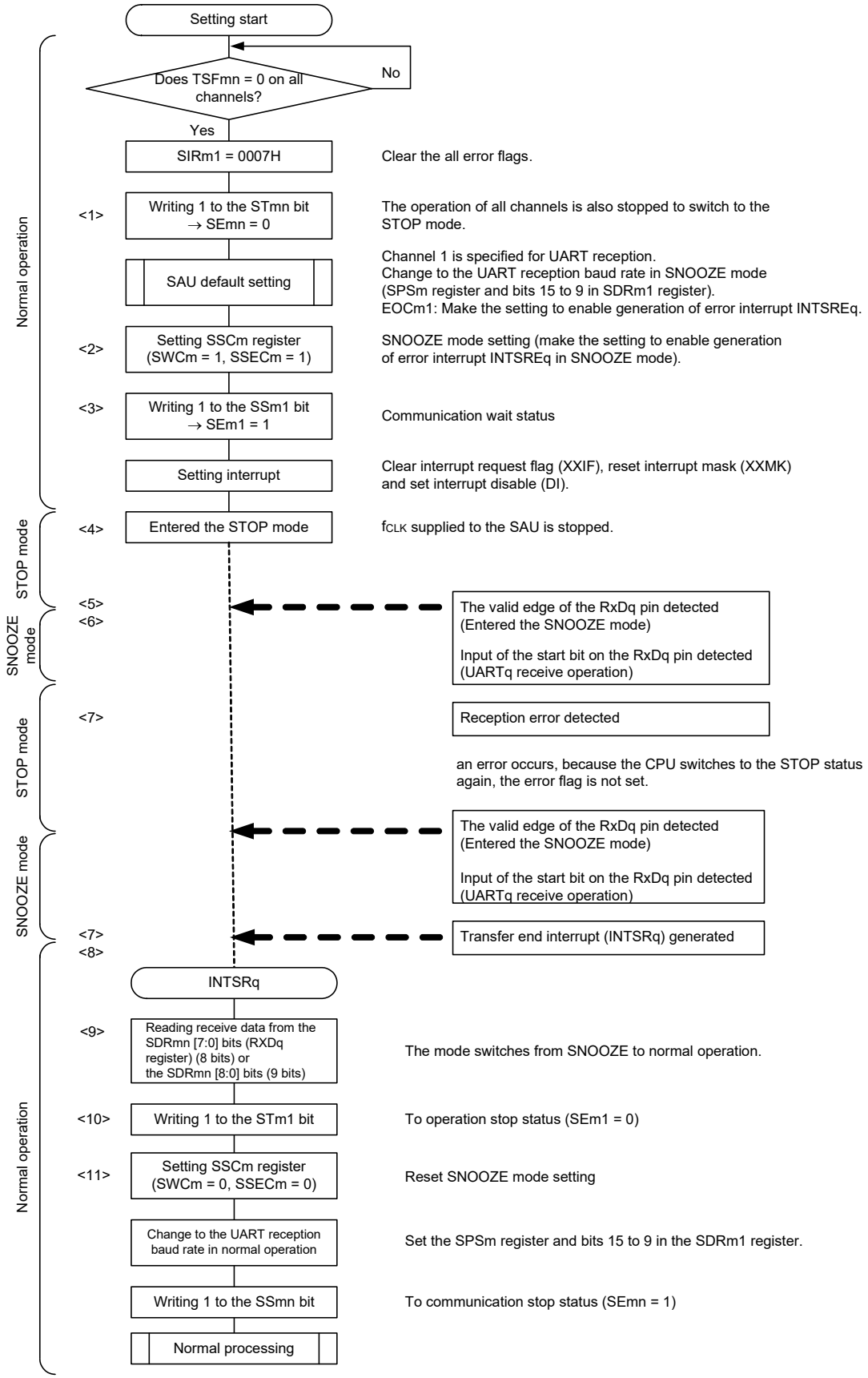
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Caution 2. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 11 - 97 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0; q = 0

Figure 11 - 97 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



(**Caution** and **Remarks** are listed on the next page.)

Caution **If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).**

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 11 - 96 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0; q = 0

11.6.4 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0 to UART3) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remark 1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11 - 4 Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01

(2) Baud rate error during transmission

The baud rate error of UART (UART0) 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 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 24 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	77	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	77	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	77	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	77	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	77	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	77	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	77	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	47	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	77	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	77	76923.1 bps	+0.16%
153600 bps	f_{CLK}	77	153846 bps	+0.16%
312500 bps	f_{CLK}	37	315789 bps	±1.05%

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0) 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.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **11.6.4 (1) Baud rate calculation expression**.)

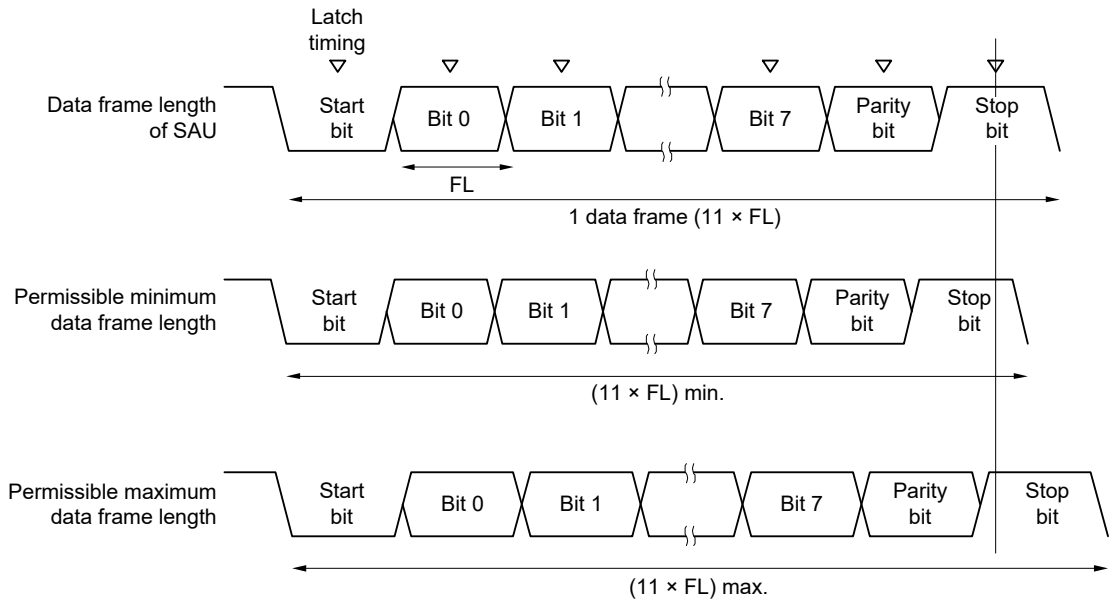
k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]

= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01

Figure 11 - 98 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 11 - 98, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 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.

11.6.5 Procedure for processing errors that occurred during UART (UART0) communication

The procedure for processing errors that occurred during UART (UART0) communication is described in Figures 11 - 99 and 11 - 100.

Figure 11 - 99 Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn) →	The BFFmn 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.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 11 - 100 Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn 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.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operating.	
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.
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

11.7 Operation of Simplified I²C (IIC00, IIC01) 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 communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the 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.

[Data transmission/reception]
- Master transmission, master reception (only master function with a single master)
- ACK output function Note 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.)
- Generation of start condition and stop condition for software

[Interrupt function]
- Transfer end interrupt

[Error detection flag]
- Overrun error
- ACK error

* [Functions not supported by simplified I²C]
- Slave transmission, slave reception
- Multi-master function (arbitration loss detection function)
- Wait detection function

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn (SOEm register) bit and serial communication data output is stopped. See the processing flow in **11.7.3 (2)** for details.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

The channel supporting simplified I2C (IIC00, IIC01) is channels 0 and 1 of SAU0.

• 32-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01

Simplified I2C (IIC00, IIC01) performs the following four types of communication operations.

- Address field transmission (See **11.7.1**.)
- Data transmission (See **11.7.2**.)
- Data reception (See **11.7.3**.)
- Stop condition generation (See **11.7.4**.)

11.7.1 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.

Simplified I^2C	IIC00	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	ACK error detection flag (PEFmn)	
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)	
Transfer rate Note 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 I^2C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)	
Data level	Non-reversed output (default: high level)	
Parity bit	No parity bit	
Stop bit	Appending 1 bit (for ACK reception timing)	
Data direction	MSB first	

Note 1. To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)).

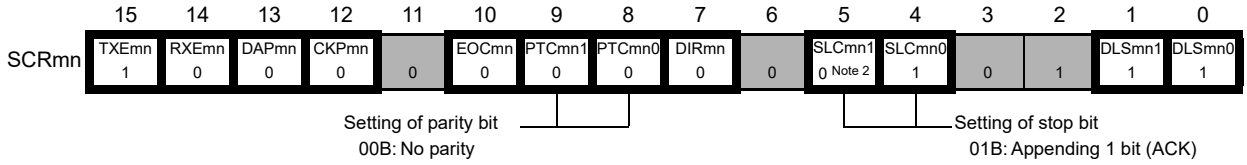
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

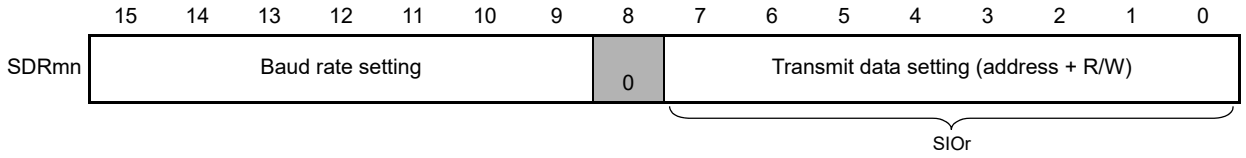
Figure 11 - 101 Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC01)

(a) Serial mode register mn (SMRmn)

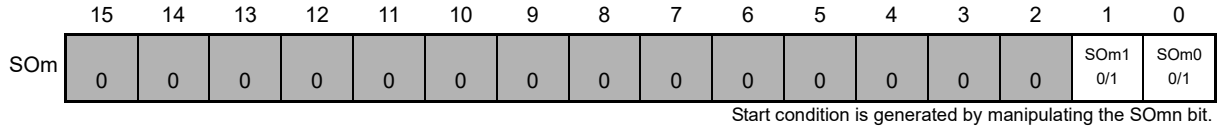
(b) Serial communication operation setting register mn (SCRmn)



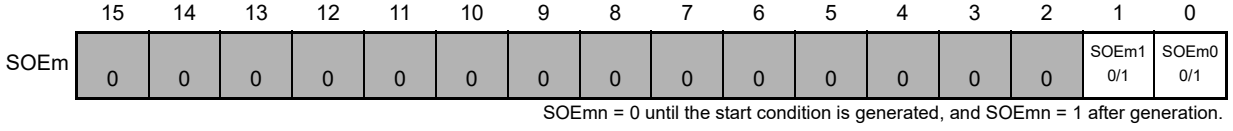
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)



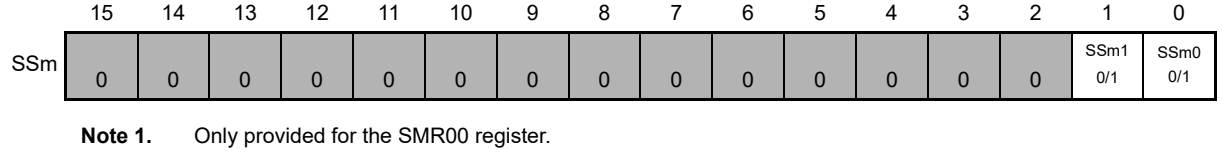
(d) Serial output register m (SOm)



(e) Serial output enable register m (SOEm)



(f) Serial channel start register m (SSm)... Sets only the bits of the target channel is 1.



Note 1. Only provided for the SMR00 register.

Note 2. Only provided for the SCR00 register.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Remark 2. : Setting is fixed in the IIC mode,

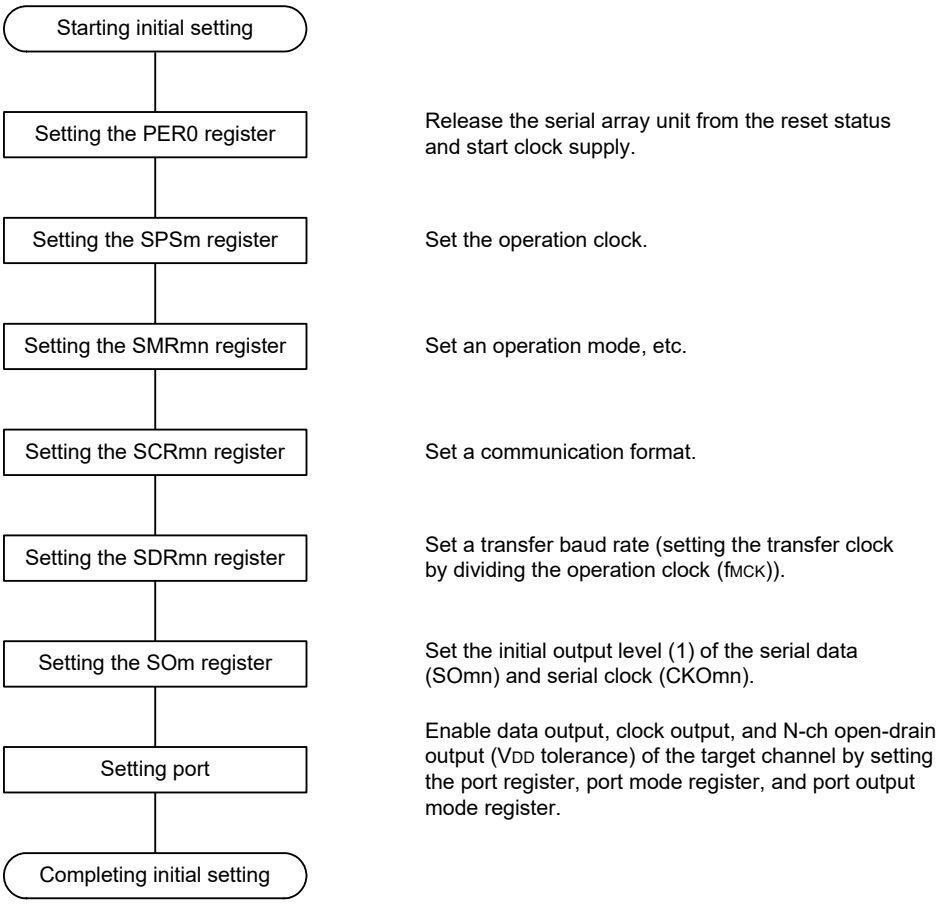
: Setting disabled (set to the initial value)

×: 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

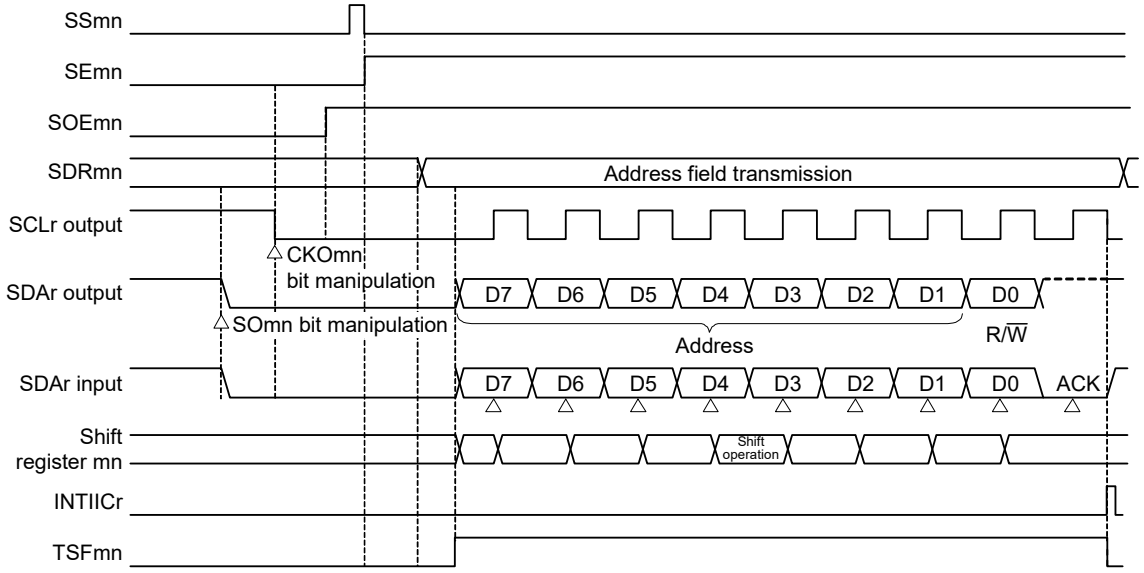
Figure 11 - 102 Initial Setting Procedure for Address Field Transmission



Remark At the end of the initial setting, the simplified I^2C (IIC00, IIC01) must be set so that output is disabled and operations are stopped.

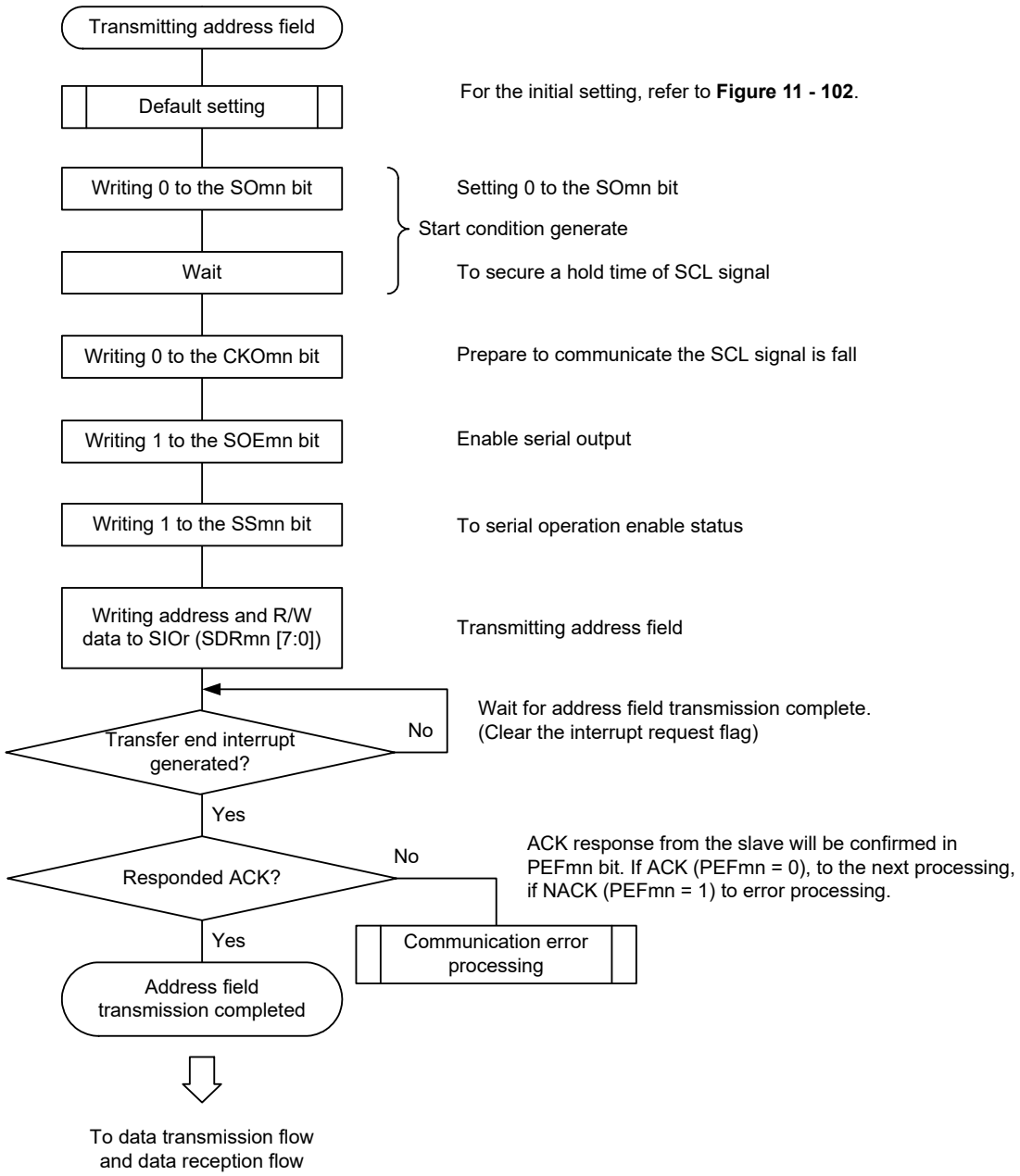
(3) Processing flow

Figure 11 - 103 Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Figure 11 - 104 Flowchart of Address Field Transmission



11.7.2 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.

Simplified I^2C	IIC00	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	ACK error flag (PEFmn)	
Transfer data length	8 bits	
Transfer rate Note 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 I^2C. • 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 1. To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 11 - 105 Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC01)

(a) Serial mode register mn (SMRmn)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 Note 1	0	SISmn0 0 Note 1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(b) Serial communication operation setting register mn (SCRmn)... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 Note 2	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)... During data transmission/reception, valid only lower 8-bits (SIOr).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting Note 3							0	Transmit data setting							
									SIOr							

(d) Serial output register m (SOm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1 Note 4	CKOm0 0/1 Note 4	0	0	0	0	0	0	SOm1 0/1 Note 4	SOm0 0/1 Note 4

(e) Serial output enable register m (SOEm)... Do not manipulate this register during data transmission/reception

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 1	SOEm0 1

(f) Serial channel start register m (SSm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Note 1. Only provided for the SMR01 register.

Note 2. Only provided for the SCR00 register.

Note 3. Because the setting is completed by address field transmission, setting is not required.

Note 4. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the IIC mode,

▒: Setting disabled (set to the initial value)

×: 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) Processing flow

Figure 11 - 106 Timing Chart of Data Transmission

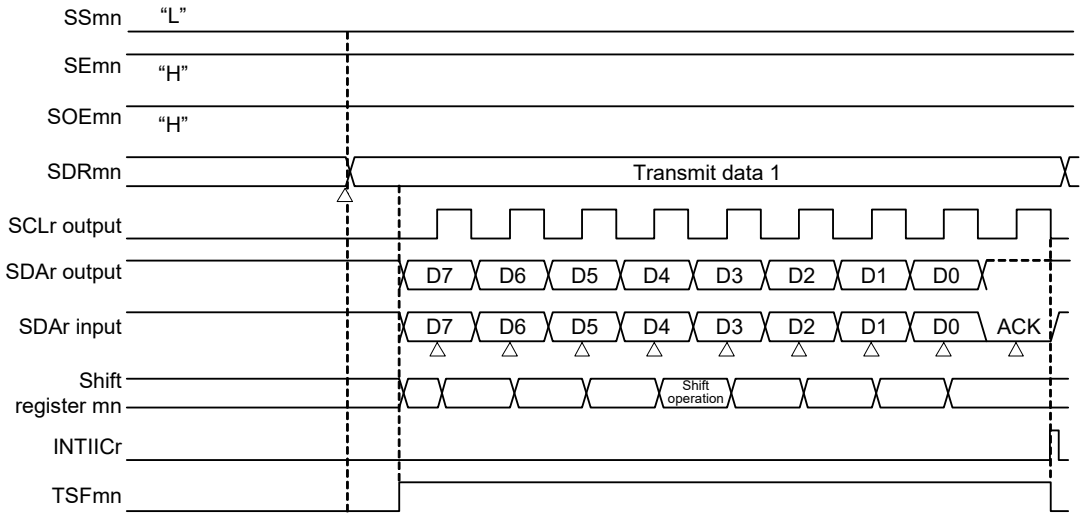
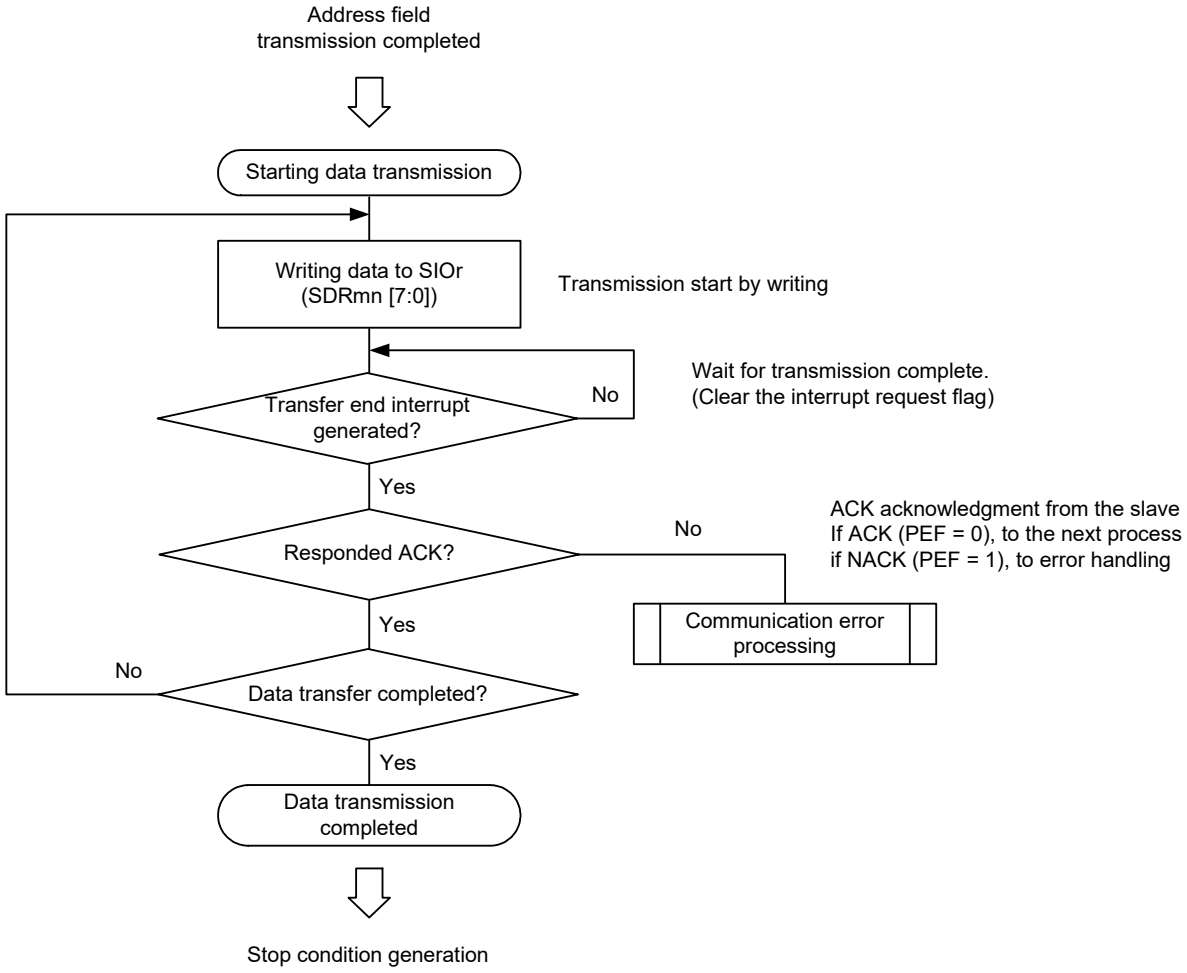


Figure 11 - 107 Flowchart of Data Transmission



11.7.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	8 bits	
Transfer rate Note 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 I^2C. • 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 1. To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **ELECTRICAL SPECIFICATIONS** in the R9A02G015 Data Sheet (R19DS0101E)).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 11 - 108 Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01)

(a) Serial mode register mn (SMRmn)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 Note 1	0	SISmn0 0 Note 1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(b) Serial communication operation setting register mn (SCRmn)... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 Note 2	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting Note 3							0	Dummy transmit data setting (FFH)							

SIOr

(d) Serial output register m (SOm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1 Note 4	CKOm0 0/1 Note 4	0	0	0	0	0	0	SOm1 0/1 Note 4	SOm0 0/1 Note 4

(e) Serial output enable register m (SOEm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Note 1. Only provided for the SMR01 register.

Note 2. Only provided for the SCR00 register.

Note 3. The baud rate setting is not required because the baud rate has already been set when the address field was transmitted.

Note 4. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the IIC mode,

■: Setting disabled (set to the initial value)

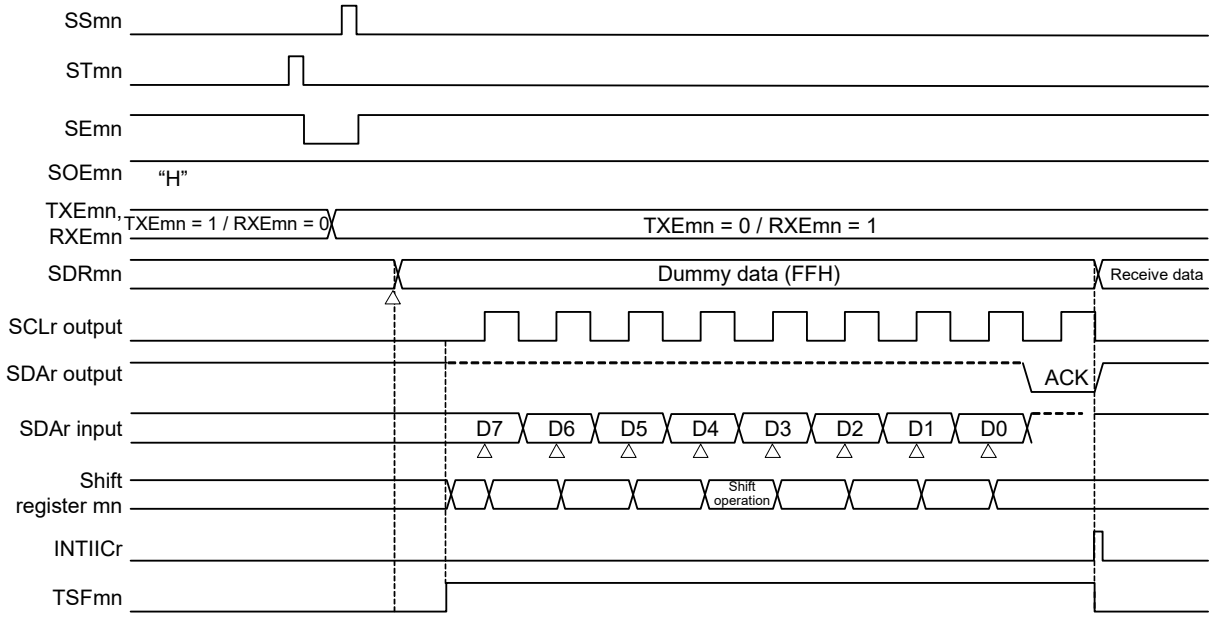
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

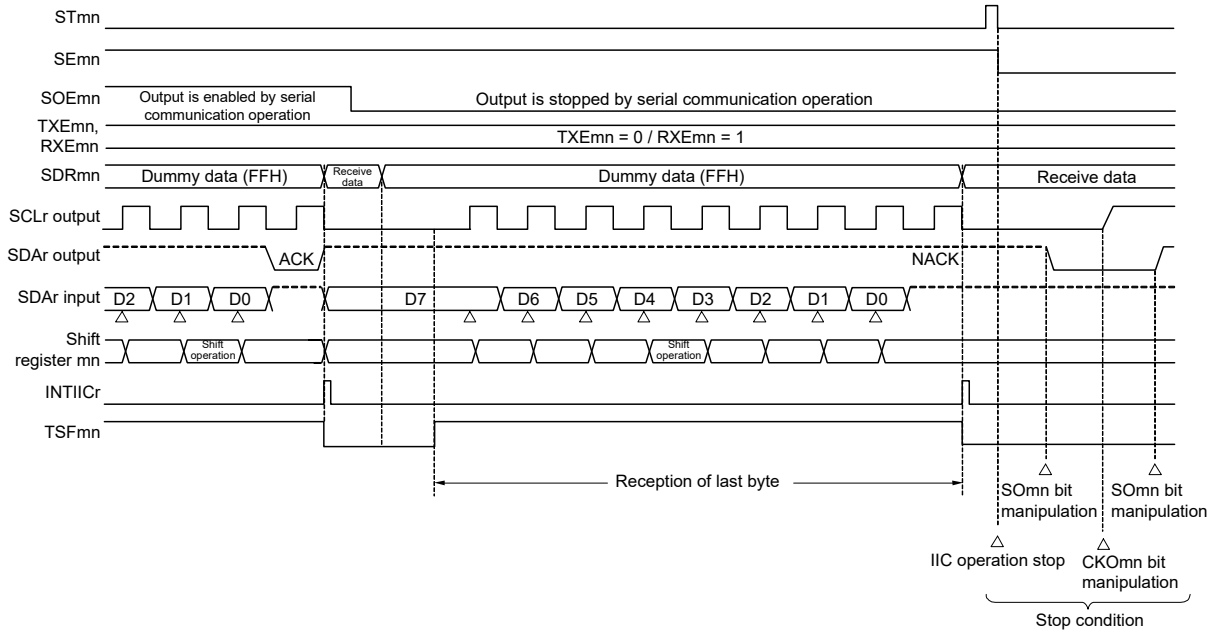
(2) Processing flow

Figure 11 - 109 Timing Chart of Data Reception

(a) When starting data reception

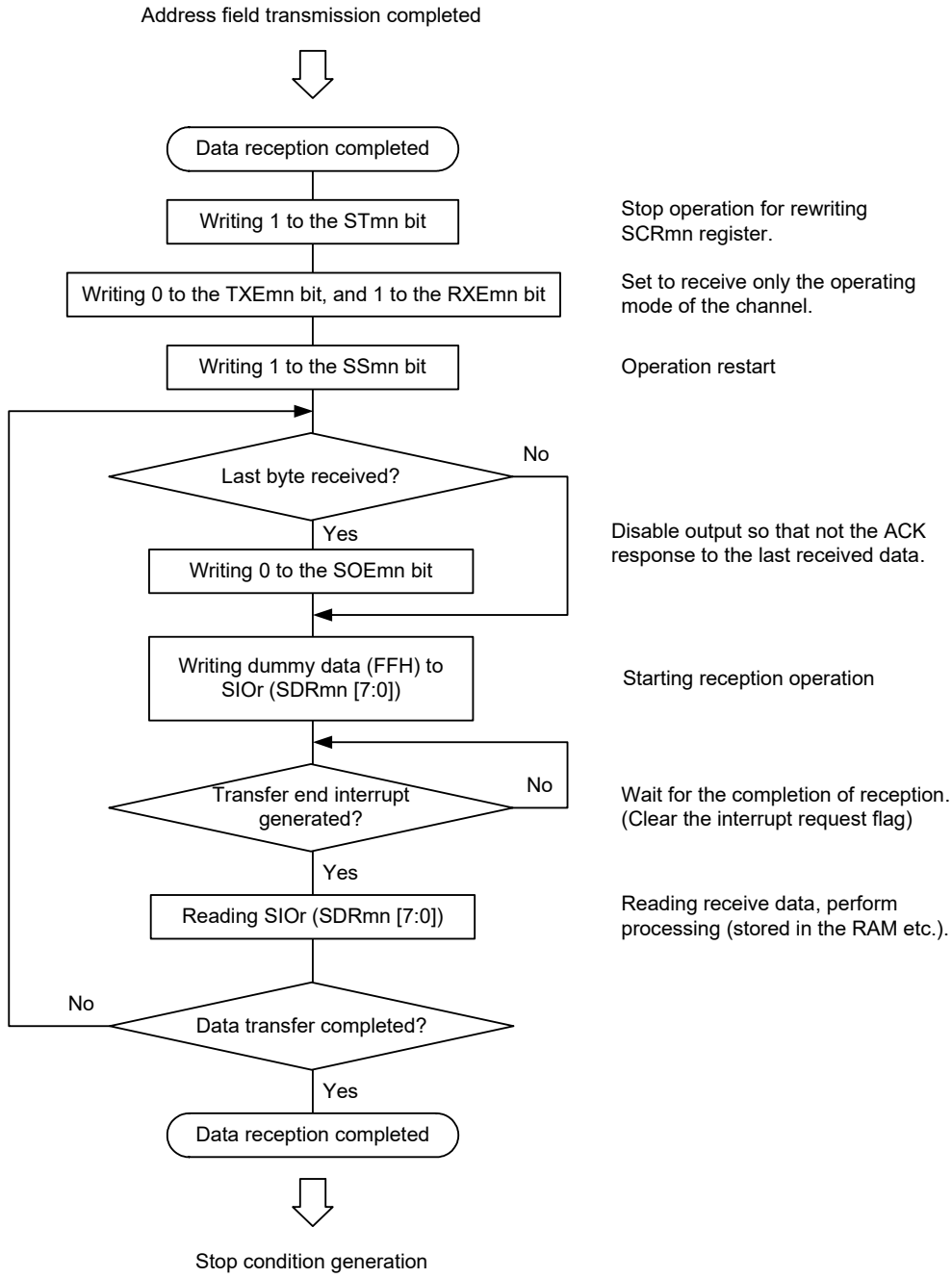


(b) When receiving last data



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Figure 11 - 110 Flowchart of Data Reception



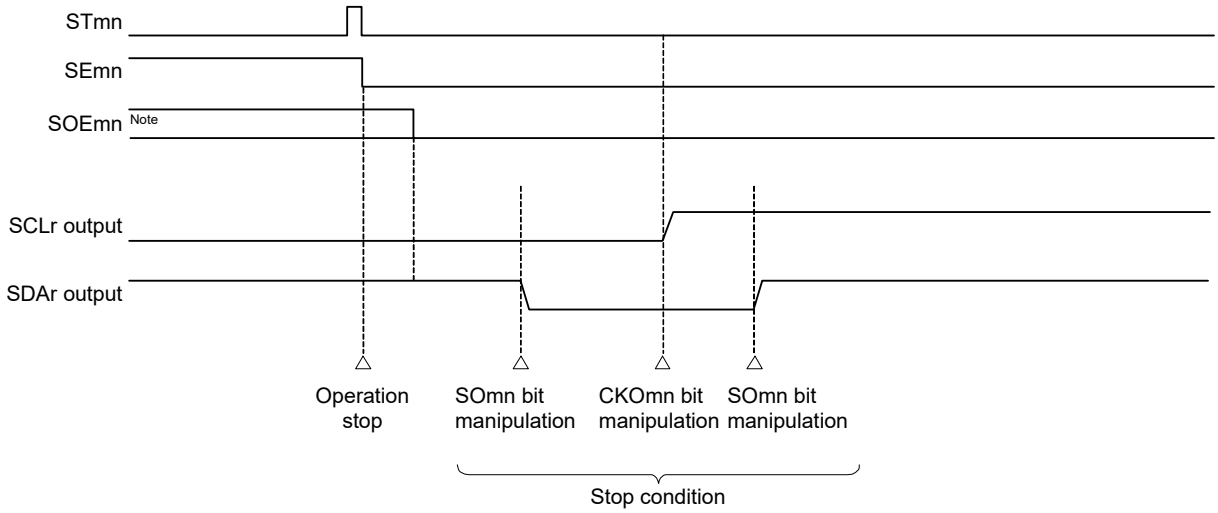
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 to the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.

11.7.4 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.

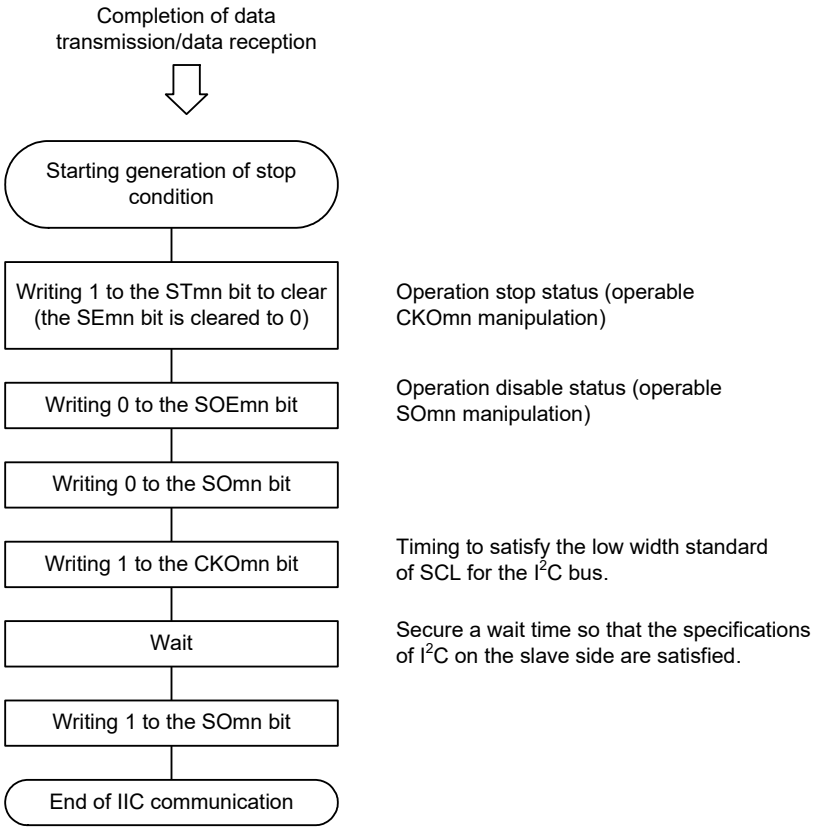
(1) Processing flow

Figure 11 - 111 Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 11 - 112 Flowchart of Stop Condition Generation



11.7.5 Calculating transfer rate

The transfer rate for simplified I²C (IIC00, IIC01) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution SDRmn[15:9] must not be set to 0000000B. Be sure to set a value of 0000001B or greater for SDRmn[15:9]. The duty ratio of the SCL signal output by the simplified I²C is 50%. The I²C 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 I²C bus specifications. In addition, it may be required to set an appropriate transfer rate so that the data setup time (reception) of R9A02G015 meet the I2C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I²C bus specifications.

Remark 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11 - 5 Selection of Operation Clock For Simplified I²C

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.7 kHz
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

Here is an example of setting an I²C transfer rate where $f_{MCK} = f_{CLK}$ = 24 MHz.

I²C Transfer Mode (Desired Transfer Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	59	100 kHz	0.0%
400 kHz	f_{CLK}	31	375 kHz	6.25% Note
1 MHz	f_{CLK}	14	0.80 MHz	20.0% Note

Note The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

11.7.6 Procedure for processing errors that occurred during simplified I²C (IIC00, IIC01) communication

The procedure for processing errors that occurred during simplified I²C (IIC00, IIC01) communication is described in **Figures 11 - 113** and **11 - 114**.

Figure 11 - 113 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn 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.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
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.

Figure 11 - 114 Processing Procedure in Case of ACK error in Simplified I²C Mode

Software Manipulation	Hardware Status	Remark
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	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.
Creates stop condition.		
Creates start condition.		
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

CHAPTER 12 SERIAL INTERFACE IICA

The number of channels of the serial Interface IICA differs, depending on the product.

	With USB	Without USB
	32-pin	32-pin
Channels	2 ch	3 ch

12.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLAn) line and a serial data bus (SDAAn) line.

This mode complies with the I^2C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since the SCLAn and SDAAn pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

(3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICAn) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUPn bit of IICA control register n1 (IICCTLn1).

Figure 12 - 1 shows a block diagram of serial interface IICA

Remark n = 0 to 2

Figure 12 - 1 Block Diagram of Serial Interface IICA

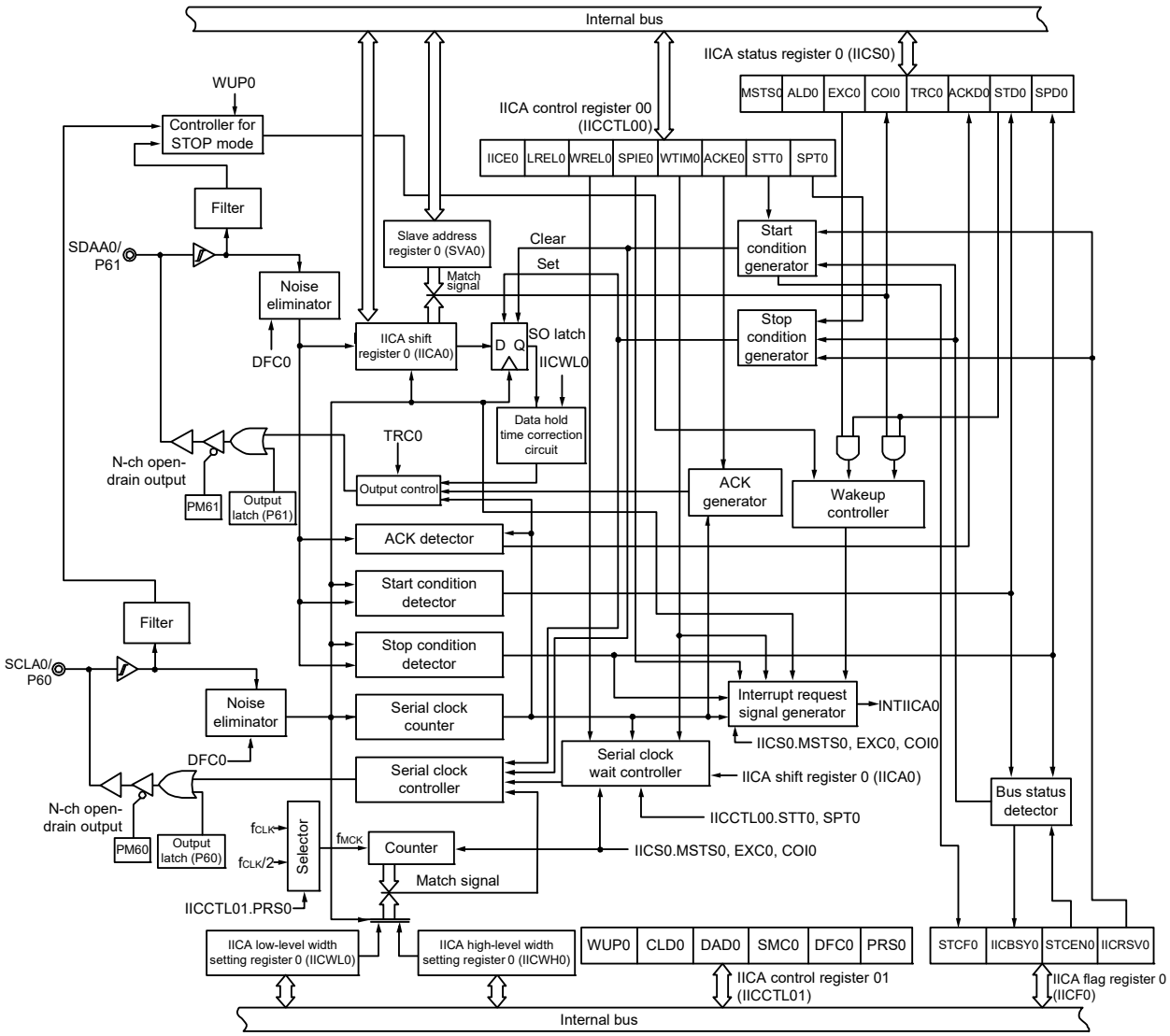
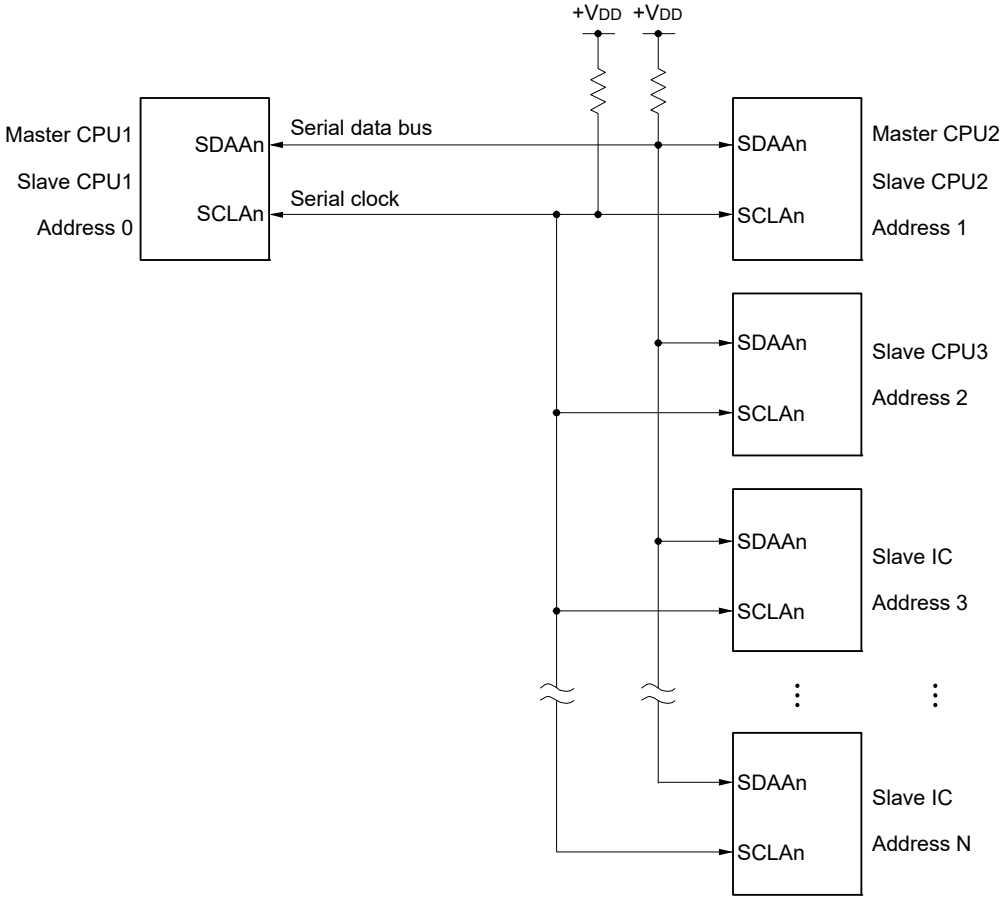


Figure 12 - 2 shows a serial bus configuration example.

Figure 12 - 2 Serial Bus Configuration Example Using I²C Bus



Remark n = 0 to 2

12.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 12 - 1 Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register n (IICAn) Slave address register n (SVAn)
Control registers	Peripheral enable register 0 (PER0) IICA control register n0 (IICCTLn0) IICA status register n (IICSn) IICA flag register n (IICFn) IICA control register n1 (IICCTLn1) IICA low-level width setting register n (IICWLn) IICA high-level width setting register n (IICWHn) Port mode registers 6, 7 (PM6, PM7) Port registers 6, 7 (P6, P7)

Remark n = 0 to 2

(1) IICA shift register n (IICAn)

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 and reading operations to the IICAn register.

Cancel the wait state and start data transfer by writing data to the IICAn register during the wait period.

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

Reset signal generation clears IICAn to 00H.

Figure 12 - 3 Format of IICA shift register n (IICAn)

Address: FFF50H (IICA0), FFF54H (IICA1), FFF60H (IICA2) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICAn								

Caution 1. Do not write data to the IICAn register during data transfer.

Caution 2. Write or read the IICAn register only during the wait period. Accessing the IICAn register in a communication state other than during the wait period is prohibited. When the device serves as the master, however, the IICAn register can be written only once after the communication trigger bit (STTn) is set to 1.

Caution 3. When communication is reserved, write data to the IICAn register after the interrupt triggered by a stop condition is detected.

Remark n = 0 to 2

(2) Slave address register n (SVAn)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

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

However, rewriting to this register is prohibited while STDn = 1 (while the start condition is detected).

Reset signal generation clears the SVAn register to 00H.

Figure 12 - 4 Format of Slave address register n (SVAn)

Address: F0234H (SVA0), F023CH (SVA1), F0244H (SVA2) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0 Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDAAn pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICAn) when the address received by this register matches the address value set to the slave address register n (SVAn) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(6) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICAn).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by the WTIMn bit)
- Interrupt request generated when a stop condition is detected (set by the SPIEn bit)

Remark WTIMn bit: Bit 3 of IICA control register n0 (IICCTLn0)
SPIEn bit: Bit 4 of IICA control register n0 (IICCTLn0)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLAn pin from a sampling clock.

(8) Serial clock wait controller

This circuit controls the wait timing.

Remark n = 0 to 2

(9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(10) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(11) Start condition generator

This circuit generates a start condition when the STTn bit is set to 1.

However, in the communication reservation disabled status (IICRSVn bit = 1), when the bus is not released (IICBSYn bit = 1), start condition requests are ignored and the STCFn bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPTn bit is set to 1.

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCENn bit.

Remark 1. STTn bit: Bit 1 of IICA control register n0 (IICCTLn0)
SPTn bit: Bit 0 of IICA control register n0 (IICCTLn0)
IICRSVn bit: Bit 0 of IICA flag register n (IICFn)
IICBSYn bit: Bit 6 of IICA flag register n (IICFn)
STCFn bit: Bit 7 of IICA flag register n (IICFn)
STCENn bit: Bit 1 of IICA flag register n (IICFn)

Remark 2. n = 0 to 2

12.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following eight registers.

- Peripheral enable register 0 (PER0)
- IICA control register n0 (IICCTLn0)
- IICA flag register n (IICFn)
- IICA status register n (IICSn)
- IICA control register n1 (IICCTLn1)
- IICA low-level width setting register n (IICWLn)
- IICA high-level width setting register n (IICWHn)
- Port mode registers 6, 7 (PM6, PM7)
- Port registers 6, 7 (P6, P7)

Remark n = 0 to 2

12.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial interface IICAn is used, be sure to set bits 7, 6, 4 (IICA2EN, IICA1EN, IICA0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12 - 5 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	IICA2EN Note	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

IICAnEN	Control of serial interface IICAn input clock supply
0	Stops input clock supply. • SFR used by serial interface IICAn cannot be written. • Serial interface IICAn is in the reset status.
1	Enables input clock supply. • SFR used by serial interface IICAn can be read/written.

Note 32-pin product without USB only.

Caution 1. When setting serial interface IICA, be sure to set the following registers first while the IICAnEN bit is set to 1. If IICAnEN = 0, the control registers of serial interface IICA are set to their initial values, and writing to them is ignored (except for port mode registers 6, 7 (PM6, PM7), and port registers 6, 7 (P6, P7)).

- **IICA control register n0 (IICCTLn0)**
- **IICA flag register n (IICFn)**
- **IICA status register n (IICSn)**
- **IICA control register n1 (IICCTLn1)**
- **IICA low-level width setting register n (IICWLn)**
- **IICA high-level width setting register n (IICWHn)**

Caution 2. Be sure to clear the following bits to 0.
32-pin product with USB: bit 7

Remark n = 0 to 2

12.3.2 IICA control register n0 (IICCTLn0)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

The IICCTLn0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIEn, WTIMn, and ACKEn bits while IICEn = 0 or during the wait period. These bits can be set at the same time when the IICEn bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Remark n = 0 to 2

Figure 12 - 6 Format of IICA control register n0 (IICCTLn0) (1/4)

Address: F0230H (IICCTL00), F0238H (IICCTL10), F0240H (IICCTL20) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I^2C operation enable
0	Stop operation. Reset the IICA status register n (IICSn) Note 1. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLAn and SDAAn lines are at high level.	
Condition for clearing (IICEn = 0)	Condition for setting (IICEn = 1)
• Cleared by instruction • Reset	• Set by instruction

LRELn Notes 2, 3	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLAn and SDAAn lines are set to high impedance. The following flags of IICA control register n0 (IICCTLn0) and the IICA status register n (IICSn) are cleared to 0. • STTn • SPTn • MSTSn • EXCn • COIn • TRCn • ACKDn • STDn
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LRELn = 0)	Condition for setting (LRELn = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WRELn Notes 2, 3	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When the WRELn bit is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRCn = 1), the SDAAn line goes into the high impedance state (TRCn = 0).	
Condition for clearing (WRELn = 0)	Condition for setting (WRELn = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Note 1. The IICA shift register n (IICAn), the STCFn and IICBSYn bits of the IICA flag register n (IICFn), and the CLDn and DADn bits of IICA control register n1 (IICCTLn1) are reset.

Note 2. The signal of this bit is invalid while IICEn is 0.

Note 3. When the LRELn and WRELn bits are read, 0 is always read.

Caution **If the operation of I^2C is enabled (IICEn = 1) when the SCLAn line is high level, the SDAAn line is low level, and the digital filter is turned on (DFCn bit of IICCTLn1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LRELn bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICEn = 1).**

Remark n = 0 to 2

Figure 12 - 7 Format of IICA control register n0 (IICCTLn0) (2/4)

SPIEn Note 1	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the WUPn bit of IICA control register n1 (IICCTLn1) is 1, no stop condition interrupt will be generated even if SPIEn = 1.	
Condition for clearing (SPIEn = 0)	Condition for setting (SPIEn = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIMn Note 1	Control of wait and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.
An interrupt is generated at the falling edge of the ninth clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an acknowledge (ACK) is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIMn = 0)	Condition for setting (WTIMn = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKEn Notes 1, 2	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAAn line is set to low level.
Condition for clearing (ACKEn = 0)	Condition for setting (ACKEn = 1)
• Cleared by instruction • Reset	• Set by instruction

Note 1. The signal of this bit is invalid while IICEn is 0. Set this bit during that period.

Note 2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Remark n = 0 to 2

Figure 12 - 8 Format of IICA control register n0 (IICCTLn0) (3/4)

STTn Notes 1, 2	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSYn = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICRSVn = 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 (IICRSVn = 1) Even if this bit is set (1), the STTn bit is cleared and the STTn clear flag (STCFn) is set (1). No start condition is generated. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPTn).
- Once STTn is set (1), setting it again (1) before the clear condition is met is not allowed.

Condition for clearing (STTn = 0)	Condition for setting (STTn = 1)
• Cleared by setting the STTn bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LRELn = 1 (exit from communications) • When IICEn = 0 (operation stop) • Reset	• Set by instruction

Note 1. The signal of this bit is invalid while IICEn is 0.
Note 2. The STTn bit is always read as 0.

Remark 1. Bit 1 (STTn) becomes 0 when it is read after data setting.
Remark 2. IICRSVn: Bit 0 of IICA flag register n (IICFn)
STCFn: Bit 7 of IICA flag register n (IICFn)
Remark 3. n = 0 to 2

Figure 12 - 9 Format of IICA control register n0 (IICCTLn0) (4/4)

SPTn Note	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer).

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as start condition trigger (STTn).
- The SPTn bit can be set to 1 only when in master mode.
- When the WTIMn bit has been cleared to 0, if the SPTn bit is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. The WTIMn bit should be changed from 0 to 1 during the wait period following the output of eight clocks, and the SPTn bit should be set to 1 during the wait period that follows the output of the ninth clock.
- Once SPTn is set (1), setting it again (1) before the clear condition is met is not allowed.

Condition for clearing (SPTn = 0)	Condition for setting (SPTn = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When IICEn = 0 (operation stop) • Reset	• Set by instruction

Note When the SPTn register is read, 0 is always read.

Caution **When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission status), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and wait is canceled, after which the TRCn bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the wait performed while the TRCn bit is 1 (transmission status) by writing to the IICA shift register n.**

Remark n = 0 to 2

12.3.3 IICA status register n (IICSn)

This register indicates the status of I^2C.

The IICSn register is read by a 1-bit or 8-bit memory manipulation instruction only when STTn = 1 and during the wait period.

Reset signal generation clears this register to 00H.

Caution Reading the IICSn register while the address match wakeup function is enabled (WUPn = 1) in STOP mode is prohibited. When the WUPn bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICAn interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIEn = 1) the interrupt generated by detecting a stop condition and read the IICSn register after the interrupt has been detected.

Remark STTn: bit 1 of IICA control register n0 (IICCTLn0)
WUPn: bit 7 of IICA control register n1 (IICCTLn1)

Figure 12 - 10 Format of IICA status register n (IICSn) (1/3)

Address: FFF51H (IICS0), FFF55H (IICS1), FFF61H (IICS2) After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

MSTSn	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTSn = 0)	Condition for setting (MSTSn = 1)
• When a stop condition is detected • When ALDn = 1 (arbitration loss) • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALDn	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 MSTSn bit is cleared.

Condition for clearing (ALDn = 0)	Condition for setting (ALDn = 1)
• Automatically cleared after the IICSn register is read Note • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the IICSn register. Therefore, when using the ALDn bit, read the data of this bit before the data of the other bits.

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0 to 2

Figure 12 - 11 Format of IICA status register n (IICSn) (2/3)

EXCn	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXCn = 0)	Condition for setting (EXCn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COIn	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COIn = 0)	Condition for setting (COIn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register n (SVAn)) (set at the rising edge of the eighth clock).

TRCn	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDAAn line is set for high impedance.
1	Transmit status. The value in the SOn latch is enabled for output to the SDAAn line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRCn = 0)	Condition for setting (TRCn = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Cleared by WRELn = 1 Note (wait cancel) • When the ALDn bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTSn, EXCn, COIn = 0) <Master> • When “1” is output to the first byte’s LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When “0” is input to the first byte’s LSB (transfer direction specification bit)	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Note When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission status), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and wait is canceled, after which the TRCn bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the wait performed while the TRCn bit is 1 (transmission status) by writing to the IICA shift register n.

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0 to 2

Figure 12 - 12 Format of IICA status register n (IICSn) (3/3)

ACKDn	Detection of acknowledge (ACK)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKDn = 0)	Condition for setting (ACKDn = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• After the SDAAn line is set to low level at the rising edge of SCLAn line's ninth clock

STDn	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.

Condition for clearing (STDn = 0)	Condition for setting (STDn = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPDn	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPDn = 0)	Condition for setting (SPDn = 1)
• 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 WUPn bit changes from 1 to 0 • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)
Remark 2. n = 0 to 2

12.3.4 IICA flag register n (IICFn)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.
The IICFn register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STTn clear flag (STCFn) and I^2C bus status flag (IICBSYn) bits are read-only.
The IICRSVn bit can be used to enable/disable the communication reservation function.
The STCENn bit can be used to set the initial value of the IICBSYn bit.
The IICRSVn and STCENn bits can be written only when the operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) = 0). When operation is enabled, the IICFn register can be read.
Reset signal generation clears this register to 00H.

Figure 12 - 13 Format of IICA flag register n (IICFn)

Address: FFF52H (IICF0), FFF56H (IICF1), FFF62H (IICF2) After reset: 00H R/W Note

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STTn flag

Condition for clearing (STCFn = 0)	Condition for setting (STCFn = 1)
• Cleared by STTn = 1 • When IICEn = 0 (operation stop) • Reset	• Generating start condition unsuccessful and the STTn bit cleared to 0 when communication reservation is disabled (IICRSVn = 1).

IICBSYn	I^2C bus status flag
0	Bus release status (communication initial status when STCENn = 1)
1	Bus communication status (communication initial status when STCENn = 0)

Condition for clearing (IICBSYn = 0)	Condition for setting (IICBSYn = 1)
• Detection of stop condition • When IICEn = 0 (operation stop) • Reset	• Detection of start condition • Setting of the IICEn bit when STCENn = 0

STCENn	Initial start enable trigger
0	After operation is enabled (IICEn = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICEn = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCENn = 0)	Condition for setting (STCENn = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSVn	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSVn = 0)	Condition for setting (IICRSVn = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Caution 1. Write to the STCENn bit only when the operation is stopped (IICEn = 0).

Caution 2. As the bus release status (IICBSYn = 0) is recognized regardless of the actual bus status when STCENn = 1, when generating the first start condition (STTn = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

Caution 3. Write to IICRSVn only when the operation is stopped (IICEn = 0).

Remark 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0 to 2

12.3.5 IICA control register n1 (IICCTLn1)

This register is used to set the operation mode of I^2C and detect the statuses of the SCLAn and SDAAn pins.

The IICCTLn1 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLDn and DADn bits are read-only.

Set the IICCTLn1 register, except the WUPn bit, while operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation clears this register to 00H.

Figure 12 - 14 Format of IICA control register n1 (IICCTLn1) (1/2)

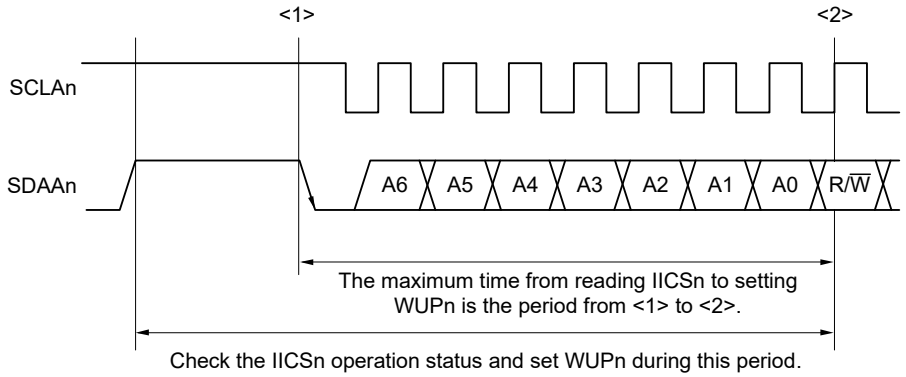
Address: F0231H (IICCTL01), F0239H (IICCTL11), F0241H (IICCTL21) After reset: 00H R/W Note 1

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.
To shift to STOP mode when WUPn = 1, execute the STOP instruction at least three fMCK clocks after setting (1) the WUPn bit (see **Figure 12 - 30 Flow When Setting WUPn = 1**). Clear (0) the WUPn bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUPn bit. (The wait must be released and transmit data must be written after the WUPn bit has been cleared (0).) The interrupt timing when the address has matched or when an extension code has been received, while WUPn = 1, is identical to the interrupt timing when WUPn = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUPn = 1, a stop condition interrupt is not generated even if the SPIEn bit is set to 1.	
Condition for clearing (WUPn = 0)	Condition for setting (WUPn = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when the MSTSn, EXCn, and COIn bits are “0”, and the STDn bit also “0” (communication not entered)) Note 2

Note 1. Bits 4 and 5 are read-only.

Note 2. The status of the IICA status register n (IICSn) must be checked and the WUPn bit must be set during the period shown below.



Remark n = 0 to 2

Figure 12 - 15 Format of IICA control register n1 (IICCTLn1) (2/2)

CLDn	Detection of SCLAn pin level (valid only when IICEn = 1)
0	The SCLAn pin was detected at low level.
1	The SCLAn pin was detected at high level.

Condition for clearing (CLDn = 0)	Condition for setting (CLDn = 1)
• When the SCLAn pin is at low level • When IICEn = 0 (operation stop) • Reset	• When the SCLAn pin is at high level

DADn	Detection of SDAAn pin level (valid only when IICEn = 1)
0	The SDAAn pin was detected at low level.
1	The SDAAn pin was detected at high level.

Condition for clearing (DADn = 0)	Condition for setting (DADn = 1)
• When the SDAAn pin is at low level • When IICEn = 0 (operation stop) • Reset	• When the SDAAn pin is at high level

SMCn	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).

DFCn	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
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 DFCn bit being set (1) or cleared (0).	

PRSn	IICA operation clock (f_{MCK}) control
0	Selects f_{CLK} (1 MHz ≤ f_{CLK} ≤ 20 MHz)
1	Selects $f_{CLK}/2$ (20 MHz ≤ f_{CLK})

Caution 1. The maximum operating frequency of the IICA operating clock (f_{MCK}) is 20 MHz (Max.). Only when f_{CLK} exceeds 20 MHz, set bit 0 (PRSn) of IICA control register n1 (IICCTLn1) to 1.

Caution 2. Note the minimum f_{CLK} operating frequency when setting the transfer clock. The minimum f_{CLK} operating frequency for serial interface IICA is determined according to the mode.

Fast mode: f_{CLK} = 3.5 MHz (MIN.)

Fast mode plus: f_{CLK} = 10 MHz (MIN.)

Normal mode: f_{CLK} = 1 MHz (MIN.)

Remark 1. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0 to 2

12.3.6 IICA low-level width setting register n (IICWLn)

This register is used to set the low-level width (tLOW) of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal.

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

Set the IICWLn register while operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWLn register, see **12.4.2 Setting transfer clock by using IICWLn and IICWHn registers**.

The data hold time is one-quarter of the time set by the IICWLn register.

Figure 12 - 16 Format of IICA low-level width setting register n (IICWLn)

Address: F0232H (IICWL0), F023AH (IICWL1), F0242H (IICWL2) After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWLn								

12.3.7 IICA high-level width setting register n (IICWHn)

This register is used to set the high-level width of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal.

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

Set the IICWHn register while operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

Figure 12 - 17 Format of IICA high-level width setting register n (IICWHn)

Address: F0233H (IICWH0), F023BH (IICWH1), F0243H (IICWH2) After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWHn								

Remark 1. For setting procedures of the transfer clock on master side and of the IICWLn and IICWHn registers on slave side, see **12.4.2 (1)** and **12.4.2 (2)**, respectively.

Remark 2. n = 0 to 2

12.3.8 Port mode registers 6, 7 (PM6, PM7)

These registers set the input/output of ports 6 and 7 in 1-bit units.

When using the P60/SCLA0 (P62/SCLA1, P70/SCLA2) pin as clock I/O and the P61/SDAA0 (P63/SDAA1, P71/SDAA2) pin as serial data I/O, clear PM60 (PM62, PM70) and PM61 (PM63, PM71), and the output latches of P60 (P62, P70) and P61 (P63, P71) to 0.

Set the IICEn bit (bit 7 of IICA control register n0 (IICCTLn0)) to 1 before setting the output mode because the P60/SCLA0 (P62/SCLA1, P70/SCLA2) and P61/SDAA0 (P63/SDAA1, P71/SDAA2) pins output a low level (fixed) when the IICEn bit is 0.

The PM6 and PM7 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 12 - 18 Format of Port mode register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	PM63	PM62	PM61	PM60

PM6x	P6x pin I/O mode selection (x = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 12 - 19 Format of Port mode register 7 (PM7)

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	1	1	1	1	PM71	PM70

PM7x	P7x pin I/O mode selection (x = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

12.4 I²C Bus Mode Functions

12.4.1 Pin configuration

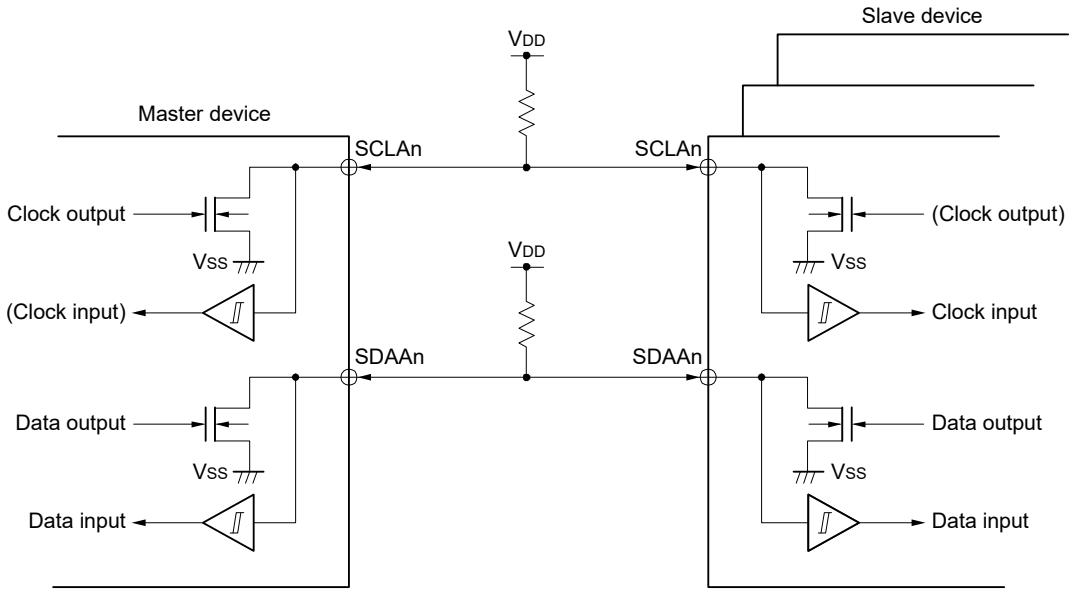
The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are configured as follows.

(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 12 - 20 Pin Configuration Diagram



Remark n = 0 to 2

12.4.2 Setting transfer clock by using IICWLn and IICWHn registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{MCK}}{IICWL + IICWH + f_{MCK}(t_R + t_F)}$$

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

$$IICWLn = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

• When the standard mode

$$IICWLn = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

• When the fast mode plus

$$IICWLn = \frac{0.50}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.50}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

(2) Setting IICWLn and IICWHn registers on slave side
(The fractional parts of all setting values are truncated.)

• When the fast mode

$IICWLn = 1.3\ \mu s \times f_{MCK}$

$IICWHn = (1.2\ \mu s - t_R - t_F) \times f_{MCK}$

• When the standard mode

$IICWLn = 4.7\ \mu s \times f_{MCK}$

$IICWHn = (5.3\ \mu s - t_R - t_F) \times f_{MCK}$

• When the fast mode plus

$IICWLn = 0.50\ \mu s \times f_{MCK}$

$IICWHn = (0.50\ \mu s - t_R - t_F) \times f_{MCK}$

Caution 1. The maximum operating frequency of the IICA operating clock (f_{MCK}) is 20 MHz (Max.). Only when f_{CLK} exceeds 20 MHz, set bit 0 (PRSn) of IICA control register n1 (IICCTLn1) to 1.

Caution 2. Note the minimum f_{CLK} operating frequency when setting the transfer clock. The minimum f_{CLK} operating frequency for serial interface IICA is determined according to the mode.

Fast mode: f_{CLK} = 3.5 MHz (MIN.)

Fast mode plus: f_{CLK} = 10 MHz (MIN.)

Normal mode: f_{CLK} = 1 MHz (MIN.)

(**Remarks** are listed on the next page.)

Remark 1. 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.

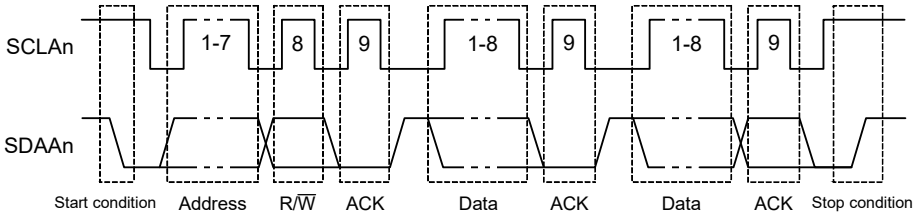
Remark 2. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
t_R: SDAAn and SCLAn signal rising times
f_{MCK}: IICA operating clock frequency

Remark 3. n = 0 to 2

12.5 I²C Bus Definitions and Control Methods

The I²C bus's serial data communication format and the signals used by the I²C bus are described below. Figure 12 - 21 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 12 - 21 I²C Bus Serial Data Transfer Timing

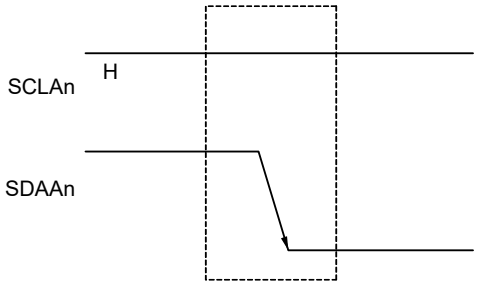


The master device generates the start condition, slave address, and stop condition.
The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).
The serial clock (SCLAn) is continuously output by the master device. However, in the slave device, the SCLAn pin low level period can be extended and a wait can be inserted.

12.5.1 Start conditions

A start condition is met when the SCLAn pin is at high level and the SDAAn pin changes from high level to low level. The start conditions for the SCLAn pin and SDAAn pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 12 - 22 Start Conditions



A start condition is output when bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set (1) after a stop condition has been detected (SPDn: Bit 0 of the IICA status register n (IICSn) = 1). When a start condition is detected, bit 1 (STDn) of the IICSn register is set (1).

Remark n = 0 to 2

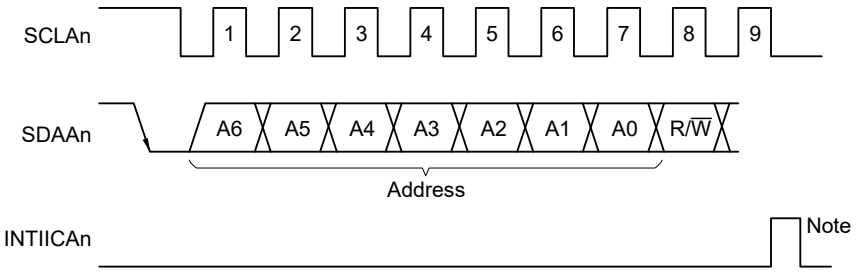
12.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register n (SVAn). If the address data matches the SVAn register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 12 - 23 Address



Note INTIICAn is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **12.5.3 Transfer direction specification** are written to the IICA shift register n (IICAn). The received addresses are written to the IICAn register.

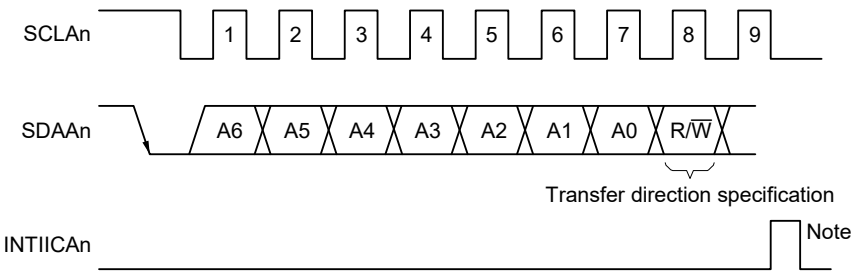
The slave address is assigned to the higher 7 bits of the IICAn register.

12.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

Figure 12 - 24 Transfer Direction Specification



Note INTIICAn is not issued if data other than a local address or extension code is received during slave device operation.

Remark n = 0 to 2

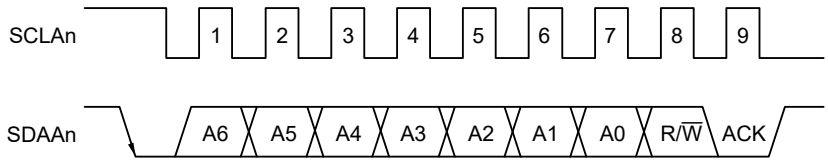
12.5.4 Acknowledge (ACK)

ACK is used to check the status of serial data at the transmission and reception sides.
The reception side returns ACK each time it has received 8-bit data.
The transmission side usually receives ACK after transmitting 8-bit data. When ACK is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether ACK has been detected can be checked by using bit 2 (ACKDn) of the IICA status register n (IICSn).
When the master receives the last data item, it does not return ACK and instead generates a stop condition. If a slave does not return ACK after receiving data, the master outputs a stop condition or restart condition and stops transmission. If ACK is not returned, the possible causes are as follows.

<1> Reception was not performed normally.
<2> The final data item was received.
<3> The reception side specified by the address does not exist.

To generate ACK, the reception side makes the SDAAn line low at the ninth clock (indicating normal reception).
Automatic generation of ACK is enabled by setting bit 2 (ACKEn) of IICA control register n0 (IICCTLn0) to 1. Bit 3 (TRCn) of the IICSn register is set by the data of the eighth bit that follows 7-bit address information. Usually, set the ACKEn bit to 1 for reception (TRCn = 0).
If a slave can receive no more data during reception (TRCn = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKEn bit to 0, that it will not receive any more data.
When the master does not require the next data item during reception (TRCn = 0), it must clear the ACKEn bit to 0 so that ACK is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 12 - 25 ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKEn bit.
When an address other than that of the local address is received, ACK is not generated (NACK).
When an extension code is received, ACK is generated if the ACKEn bit is set to 1 in advance.
How ACK is generated when data is received differs as follows depending on the setting of the wait timing.

- When 8-clock wait state is selected (bit 3 (WTIMn) of IICCTLn0 register = 0):
  By setting the ACKEn bit to 1 before releasing the wait state, ACK is generated at the falling edge of the eighth clock of the SCLAn pin.
- When 9-clock wait state is selected (bit 3 (WTIMn) of IICCTLn0 register = 1):
  ACK is generated by setting the ACKEn bit to 1 in advance.

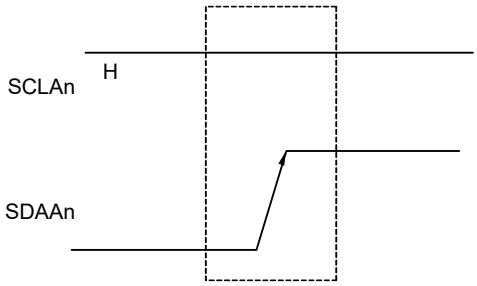
Remark n = 0 to 2

12.5.5 Stop condition

When the SCLAn pin is at high level, changing the SDAAn pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 12 - 26 Stop Condition



A stop condition is generated when bit 0 (SPTn) of IICA control register n0 (IICCTLn0) is set to 1. When the stop condition is detected, bit 0 (SPDn) of the IICA status register n (IICSn) is set to 1 and INTIICAn is generated when bit 4 (SPIEn) of the IICCTLn0 register is set to 1.

Remark n = 0 to 2

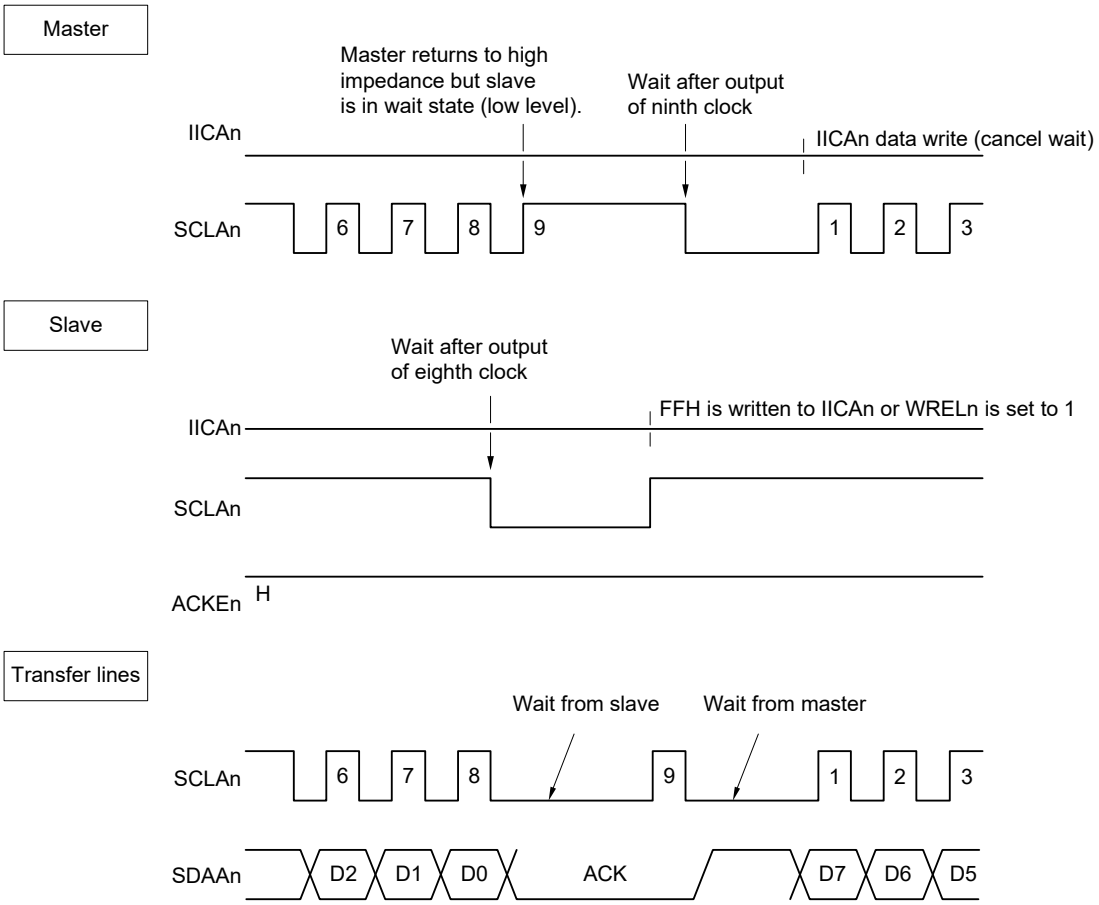
12.5.6 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCLAn pin to low level notifies the communication partner of the wait state. When wait state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 12 - 27 Wait (1/2)

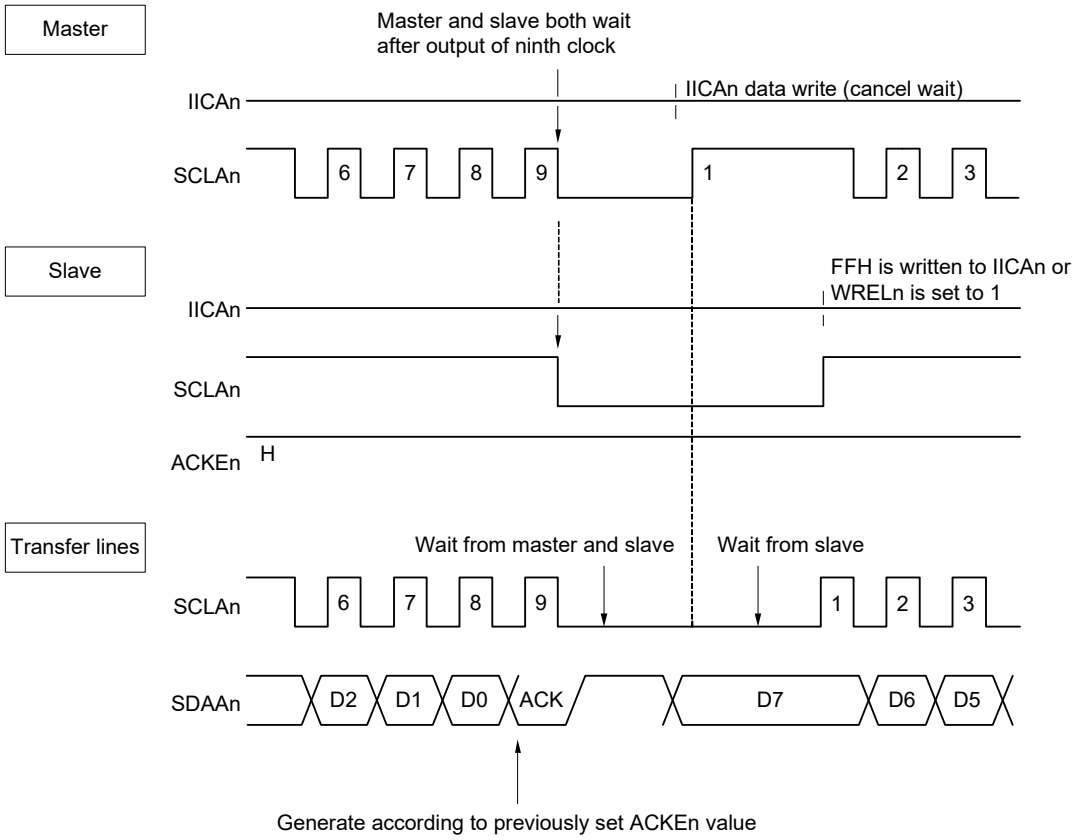
(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKEn = 1)



Remark n = 0 to 2

Figure 12 - 28 Wait (2/2)

(2) When master and slave devices both have a nine-clock wait
(master transmits, slave receives, and ACKEn = 1)



Remark ACKEn: Bit 2 of IICA control register n0 (IICCTLn0)
WRELn: Bit 5 of IICA control register n0 (IICCTLn0)

A wait may be automatically generated depending on the setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0).

Normally, the receiving side cancels the wait state when bit 5 (WRELn) of the IICCTLn0 register is set to 1 or when FFH is written to the IICA shift register n (IICAn), and the transmitting side cancels the wait state when data is written to the IICAn register.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STTn) of the IICCTLn0 register to 1
- By setting bit 0 (SPTn) of the IICCTLn0 register to 1

Remark n = 0 to 2

12.5.7 Canceling wait

The I2C usually cancels a wait state by the following processing.

- Writing data to the IICA shift register n (IICAn)
- Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (canceling wait)
- Setting bit 1 (STTn) of the IICCTLn0 register (generating start condition) Note
- Setting bit 0 (SPTn) of the IICCTLn0 register (generating stop condition) Note

Note Master only

When the above wait canceling processing is executed, the I2C cancels the wait state and communication is resumed.
To cancel a wait state and transmit data (including addresses), write the data to the IICAn register.
To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WRELn) of the IICCTLn0 register to 1.
To generate a restart condition after canceling a wait state, set bit 1 (STTn) of the IICCTLn0 register to 1.
To generate a stop condition after canceling a wait state, set bit 0 (SPTn) of the IICCTLn0 register to 1.
Execute the canceling processing only once for one wait state.
If, for example, data is written to the IICAn register after canceling a wait state by setting the WRELn 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, communication is stopped if the IICEn bit is cleared to 0 when communication has been aborted, so that the wait state can be canceled.
If the I2C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LRELn) of the IICCTLn0 register, so that the wait state can be canceled.

Caution If a processing to cancel a wait state is executed when WUPn = 1, the wait state will not be canceled.

Remark n = 0 to 2

12.5.8 Interrupt request (INTIICAn) generation timing and wait control

The setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0) determines the timing by which INTIICAn is generated and the corresponding wait control, as shown in Table 12 - 2.

Table 12 - 2 INTIICAn Generation Timing and Wait Control

WTIMn	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Note 1. The slave device's INTIICAn signal and wait period occurs at the falling edge of the ninth clock 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 register's bit 2 (ACKEn). For a slave device that has received an extension code, INTIICAn occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICAn is generated at the falling edge of the 9th clock, but wait does not occur.

Note 2. If the received address does not match the contents of the slave address register n (SVAn) and extension code is not received, neither INTIICAn nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception
- Slave device operation: Interrupt and wait timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIMn bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIMn bit.

(2) During data reception
- Master/slave device operation: Interrupt and wait timing are determined according to the WTIMn bit.

(3) During data transmission
- Master/slave device operation: Interrupt and wait timing are determined according to the WTIMn bit.

Remark n = 0 to 2

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to the IICA shift register n (IICAn)
- Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (canceling wait)
- Setting bit 1 (STTn) of IICCTLn0 register (generating start condition) Note
- Setting bit 0 (SPTn) of IICCTLn0 register (generating stop condition) Note

Note Master only.

When an 8-clock wait has been selected (WTIMn = 0), the presence/absence of ACK generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIICAn is generated when a stop condition is detected (only when SPIEn = 1).

12.5.9 Address match detection method

In I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request (INTIICAn) occurs when the address set to the slave address register n (SVAn) matches the slave address sent by the master device, or when an extension code has been received.

12.5.10 Error detection

In I^2C bus mode, the status of the serial data bus (SDAAn) during data transmission is captured by the IICA shift register n (IICAn) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

Remark n = 0 to 2

12.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXCn) is set to 1 for extension code reception and an interrupt request (INTIICAn) is issued at the falling edge of the eighth clock. The local address stored in the slave address register n (SVAn) is not affected.

(2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer when the SVAn register is set to 11110xx0. Note that INTIICAn occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXCn = 1
- Seven bits of data match: COIn = 1

Remark EXCn: Bit 5 of IICA status register n (IICSn)
COIn: Bit 4 of IICA status register n (IICSn)

(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 while a slave device is operating, 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 bit 6 (LRELn) of IICA control register n0 (IICCTLn0) to 1 to set the standby mode for the next communication operation.

Table 12 - 3 Bit Definitions of Major Extension Codes

Slave Address	R/W Bit	Description
0000 000	0	General call address
1111 0xx	0	10-bit slave address specification (during address authentication)
1111 0xx	1	10-bit slave address specification (after address match, when read command is issued)

Remark 1. See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.
Remark 2. n = 0 to 2

12.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STTn bit is set to 1 before the STDn bit is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

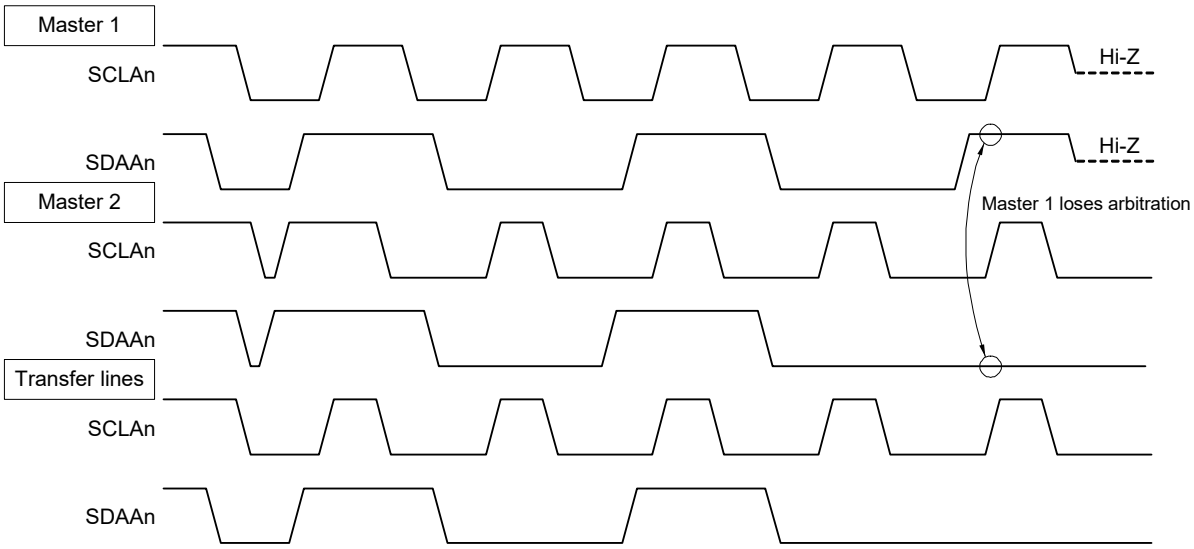
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in the IICA status register n (IICSn) is set (1) via the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALDn = 1 setting that has been made by software.

For details of interrupt request timing, see **12.5.8 Interrupt request (INTIICAn) generation timing and wait control**.

Remark STDn: Bit 1 of IICA status register n (IICSn)
STTn: Bit 1 of IICA control register n0 (IICCTLn0)

Figure 12 - 29 Arbitration Timing Example



Remark n = 0 to 2

Table 12 - 4 Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer Note 1
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During ACK transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIEn = 1) Note 2
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer Note 1
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIEn = 1) Note 2
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer Note 1
When SCLAn is at low level while attempting to generate a restart condition	

Note 1. When the WTIMn bit (bit 3 of IICA control register n0 (IICCTLn0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIMn = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

Note 2. When there is a chance that arbitration will occur, set SPIEn = 1 for master device operation.

Remark 1. SPIEn: Bit 4 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0 to 2

12.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICAn) when a local address and extension code have been received.

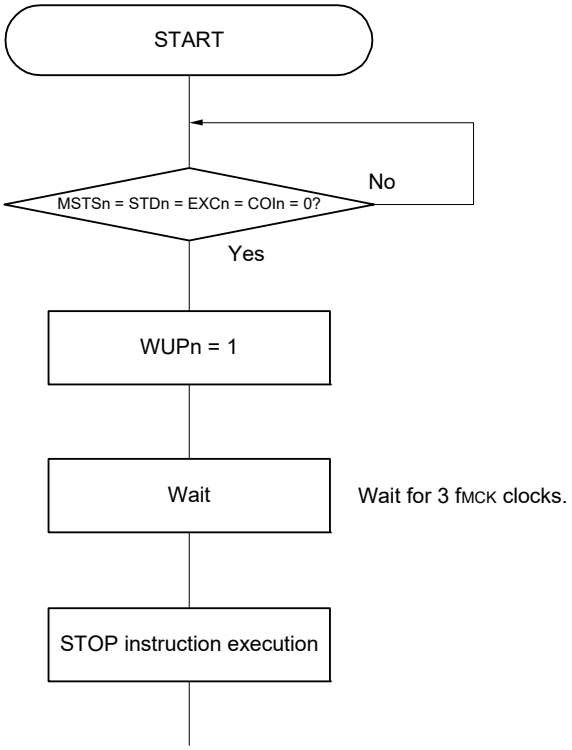
This function makes processing more efficient by preventing unnecessary INTIICAn signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

To use the wakeup function in the STOP mode, set the WUPn bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICAn) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUPn bit after this interrupt has been generated.

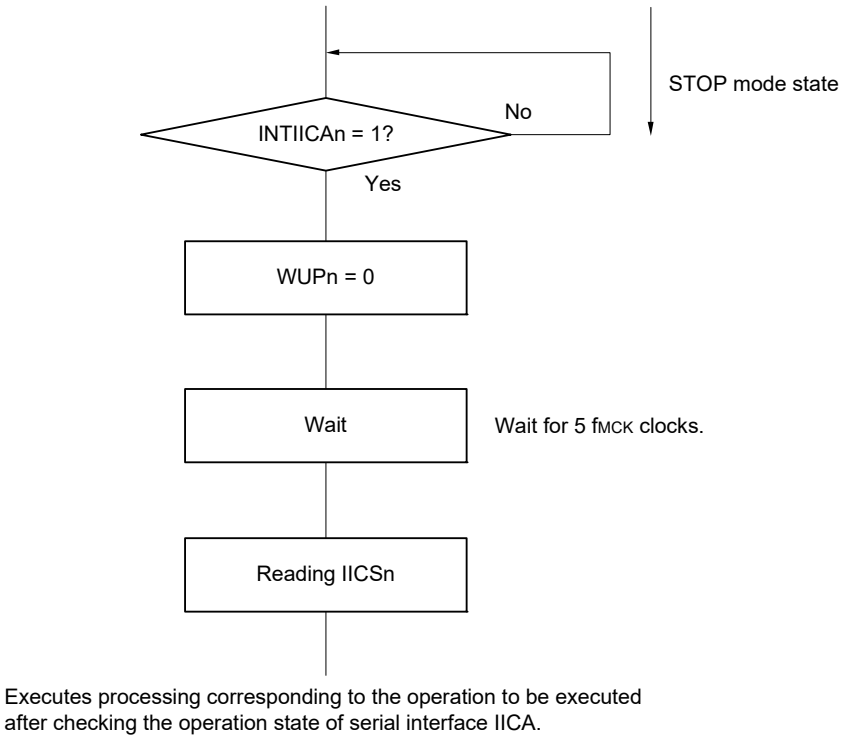
Figure 12 - 30 shows the flow for setting WUPn = 1 and Figure 12 - 31 shows the flow for setting WUPn = 0 upon an address match.

Figure 12 - 30 Flow When Setting WUPn = 1



Remark n = 0 to 2

Figure 12 - 31 Flow When Setting WUPn = 0 upon Address Match (Including Extension Code Reception)

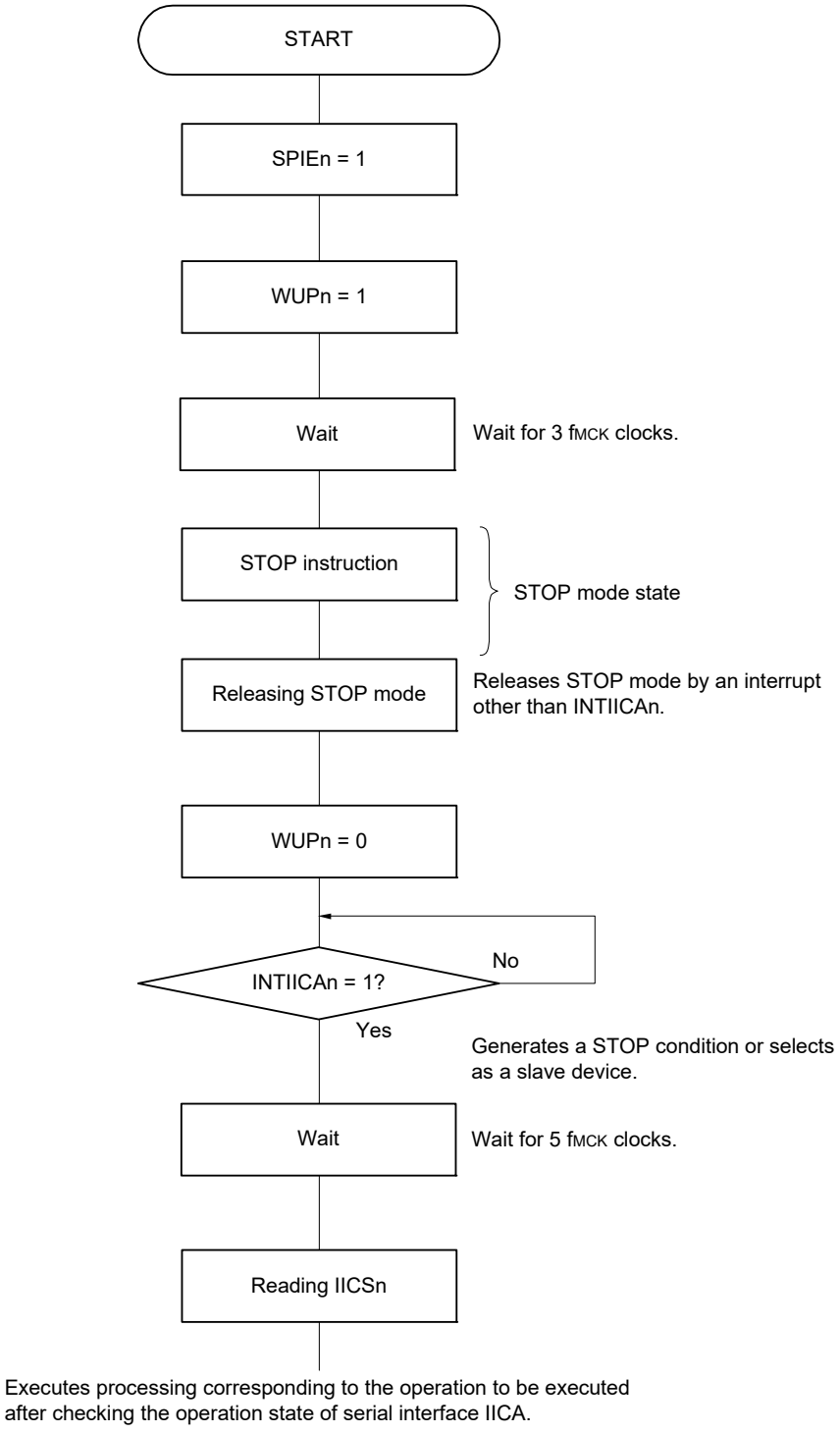


Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICAn) generated from serial interface IICA.

- When operating as the master device for the next IIC communication: Flow shown in Figure 12 - 32
- When operating as a slave device for the next IIC communication:
  - When the INTIICAn interrupt is used to return from the mode:
    - Same as the flow in Figure 12 - 31
  - When an interrupt other than the INTIICAn interrupt is used to return from the mode:
    - Continue operation while WUPn = 1 until an INTIICAn interrupt is generated.

Remark n = 0 to 2

Figure 12 - 32 When Operating as Master Device after Releasing STOP Mode other than by INTIICAn



Remark n = 0 to 2

12.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of IICA control register n0 (IICCTLn0) to 1 and saving communication).

If bit 1 (STTn) of the IICCTLn0 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register n (IICAn) after bit 4 (SPIEn) of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid.
When the STTn bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released.....a start condition is generated
- If the bus has not been released (standby mode).....communication reservation

Check whether the communication reservation operates or not by using the MSTSn bit (bit 7 of the IICA status register n (IICSn)) after the STTn bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

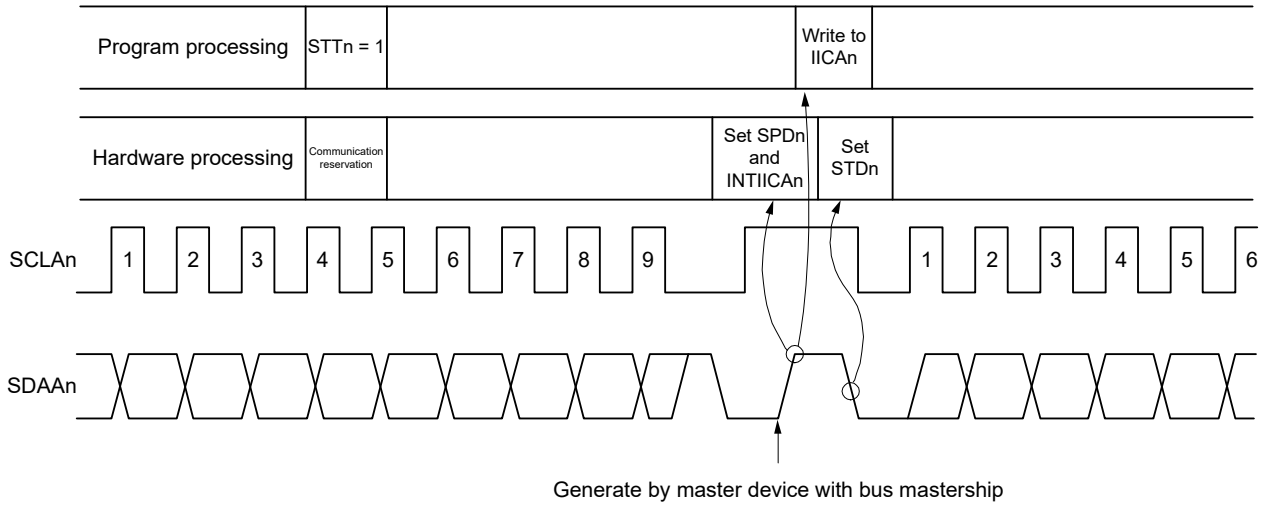
Wait time from setting STTn = 1 to checking the MSTSn flag:
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + $t_F \times 2$

Remark 1. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operating clock frequency

Remark 2. n = 0 to 2

Figure 12 - 33 shows the Communication Reservation Timing.

Figure 12 - 33 Communication Reservation Timing



Remark IICAn: IICA shift register n
STTn: Bit 1 of IICA control register n0 (IICCTLn0)
STDn: Bit 1 of IICA status register n (IICSn)
SPDn: Bit 0 of IICA status register n (IICSn)

Communication reservations are accepted via the timing shown in Figure 12 - 34. After bit 1 (STDn) of the IICA status register n (IICSn) is set to 1, a communication reservation can be made by setting bit 1 (STTn) of IICA control register n0 (IICCTLn0) to 1 before a stop condition is detected.

Figure 12 - 34 Timing for Accepting Communication Reservations

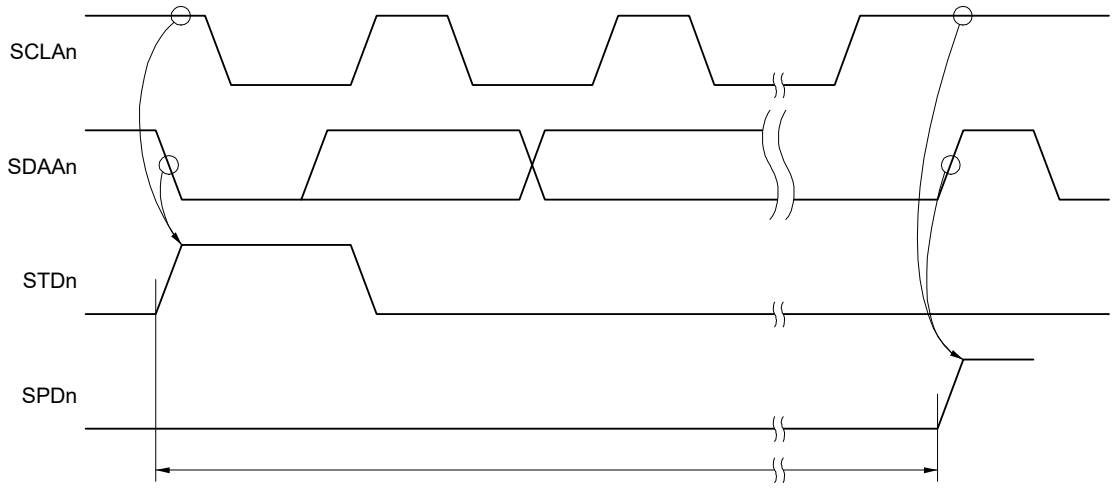
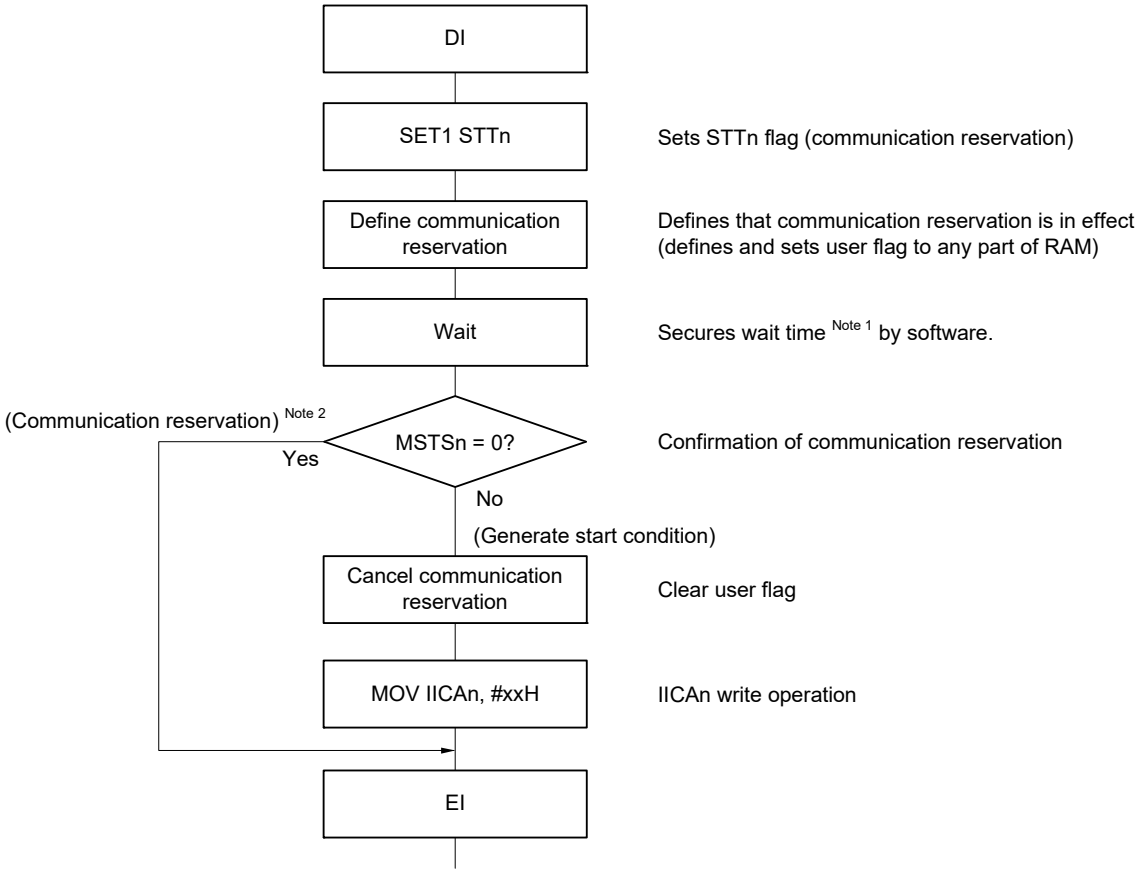


Figure 12 - 35 shows the Communication Reservation Protocol.

Remark n = 0 to 2

Figure 12 - 35 Communication Reservation Protocol



Note 1. The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + $t_F \times 2$

Note 2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

Remark1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)
MSTSn: Bit 7 of IICA status register n (IICSn)
IICAn: IICA shift register n
IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operating clock frequency

Remark2. n = 0 to 2

(2) When communication reservation function is disabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 1)
When bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of the IICCTLn0 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCFn (bit 7 of the IICFn register). It takes up to five f_{MCK} clocks until the STCFn bit is set to 1 after setting STTn = 1. Therefore, secure this time by software.

Remark n = 0 to 2

12.5.15 Cautions

(1) When STCENn = 0

Immediately after I2C operation is enabled (IICEn = 1), the bus communication status (IICBSYn = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IICA control register n1 (IICCTLn1).
<2> Set bit 7 (IICEn) of IICA control register n0 (IICCTLn0) to 1.
<3> Set bit 0 (SPTn) of the IICCTLn0 register to 1.

(2) When STCENn = 1

Immediately after I2C operation is enabled (IICEn = 1), the bus released status (IICBSYn = 0) is recognized regardless of the actual bus status. To generate the first start condition (STTn = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I2C communications are already in progress

If I2C 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 macro of I2C 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, ACK is returned, but this interferes with other I2C communications. To avoid this, start I2C in the following sequence.

<1> Clear bit 4 (SPIEn) of the IICCTLn0 register to 0 to disable generation of an interrupt request signal (INTIICAn) when the stop condition is detected.
<2> Set bit 7 (IICEn) of the IICCTLn0 register to 1 to enable the operation of I2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LRELn) of the IICCTLn0 register to 1 before ACK is returned (4 to 72 f_{MCK} clocks after setting the IICEn bit to 1), to forcibly disable detection.

(4) Setting the STTn and SPTn bits (bits 1 and 0 of the IICCTLn0 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIEn bit (bit 4 of the IICCTLn0 register) 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 SPIEn bit to 1 when the MSTSn bit (bit 7 of the IICA status register n (IICSn)) is detected by software.

Remark n = 0 to 2

12.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the R9A02G015 as the master in a single master system is shown below.
This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In the I^2C bus multimaster system, whether the bus is released or used cannot be judged by the I^2C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the R9A02G015 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 the R9A02G015 looses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

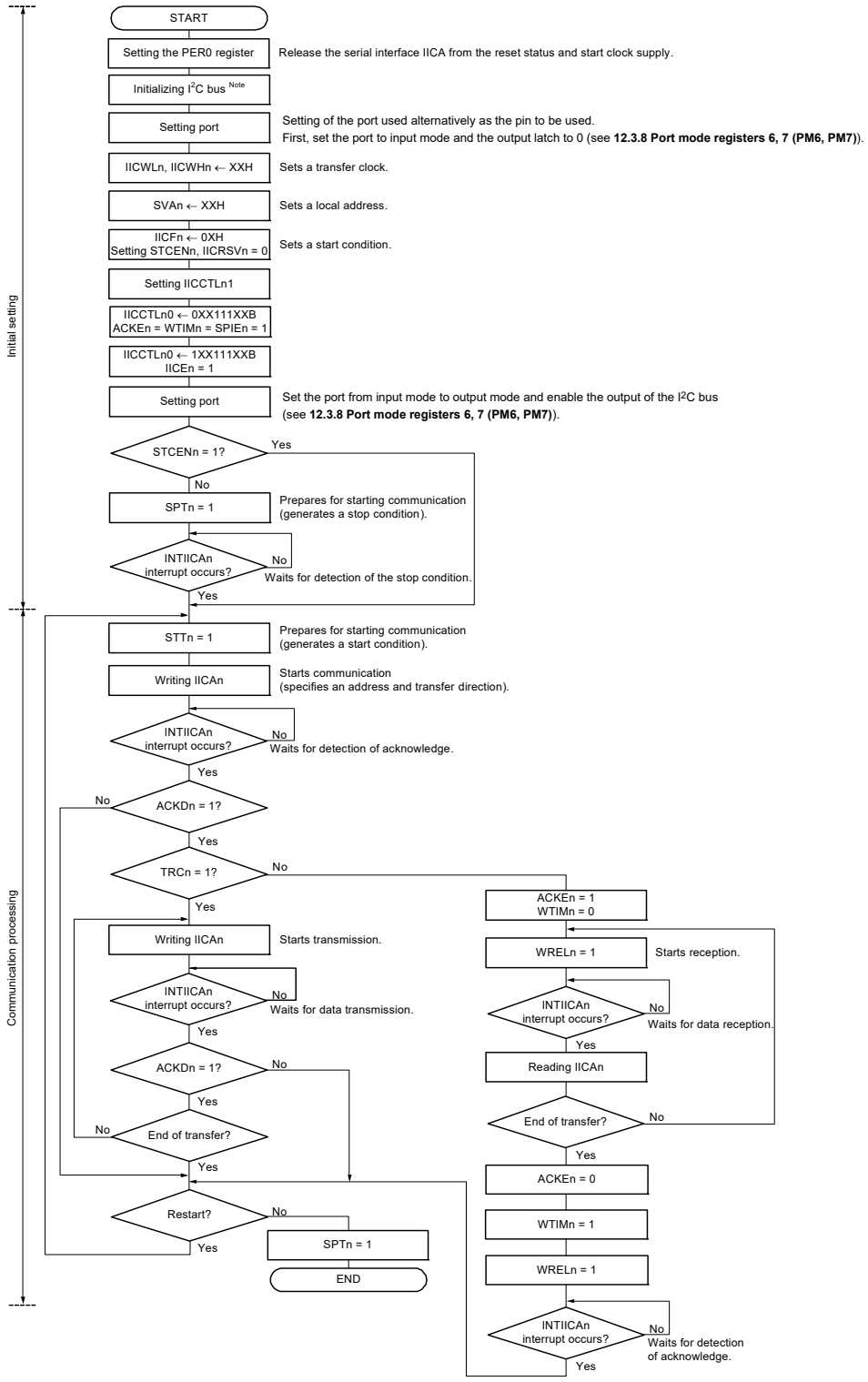
(3) Slave operation

An example of when the R9A02G015 is used as the I^2C bus slave is shown below.
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICAn interrupt occurrence (communication waiting). When an INTIICAn 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.

Remark n = 0 to 2

(1) Master operation in single master system

Figure 12 - 36 Master Operation in Single-Master System



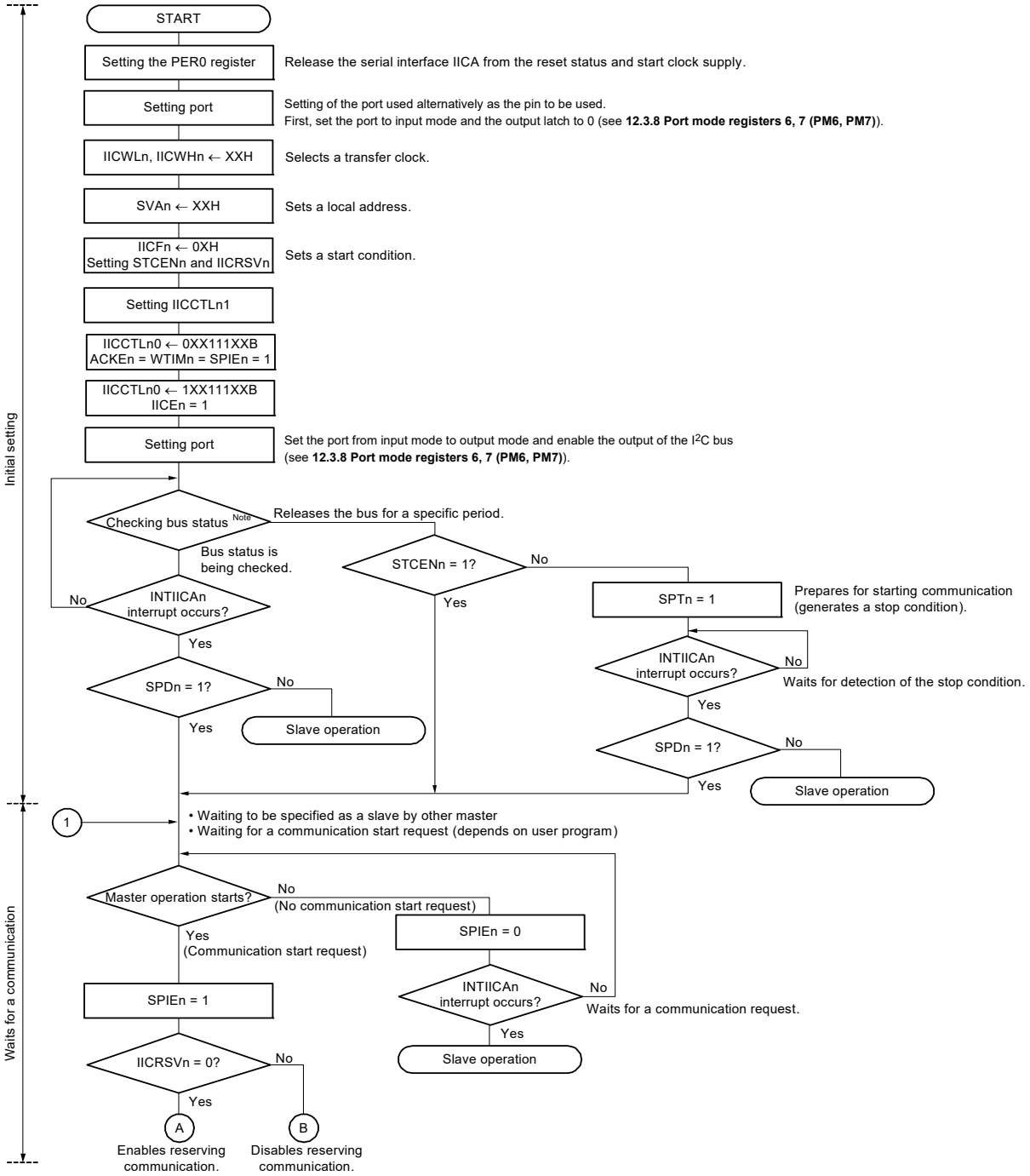
Note Release (SCLAn and SDAAn pins = high level) the I2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

Remark1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark2. n = 0 to 2

(2) Master operation in multimaster system

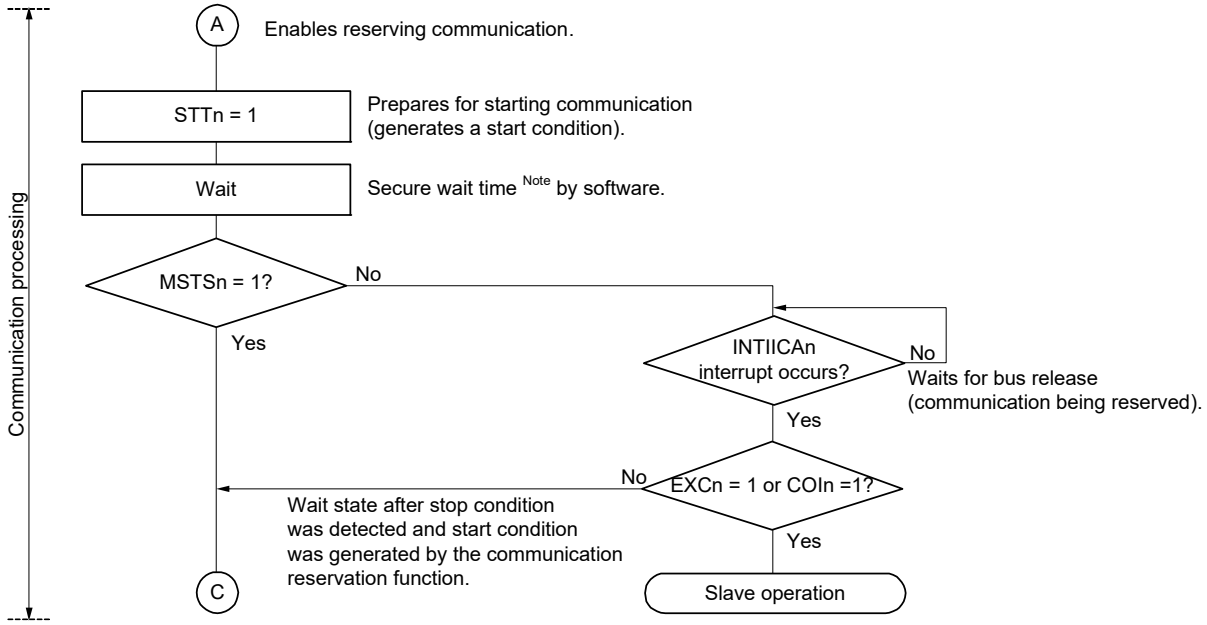
Figure 12 - 37 Master Operation in Multi-Master System (1/3)



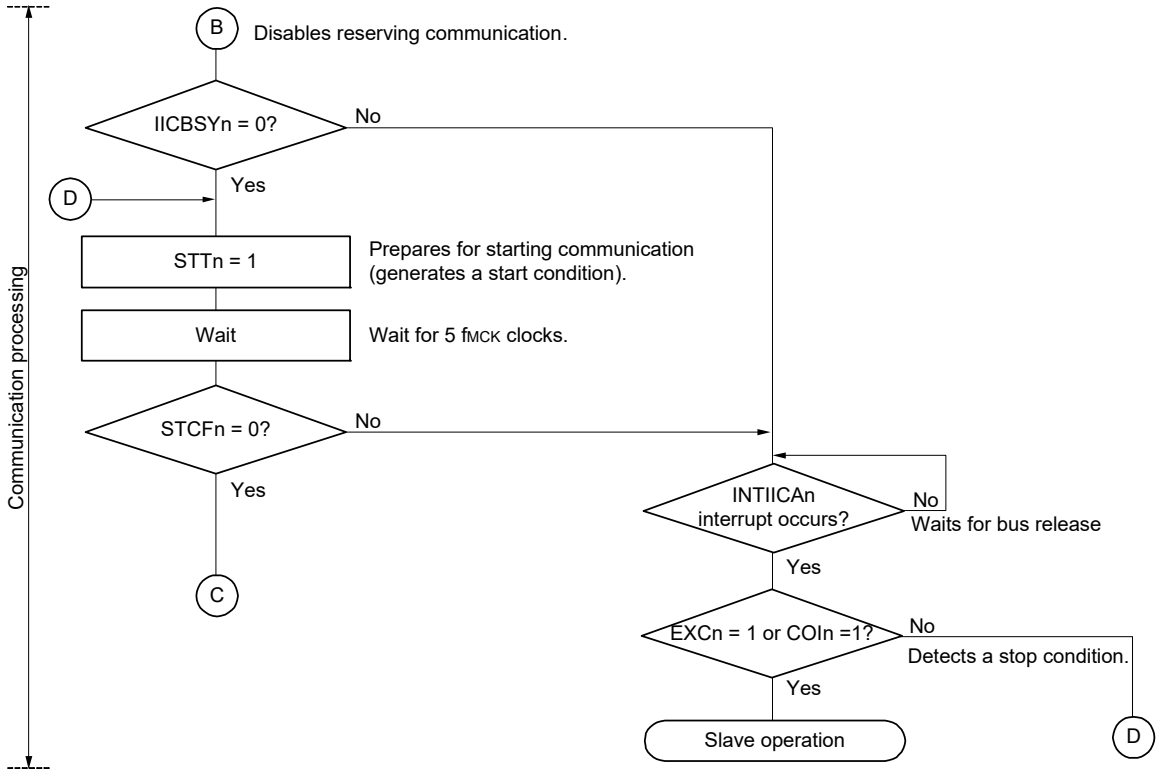
Note Confirm that the bus is released (CLDn bit = 1, DADn bit = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I^2C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

Remark n = 0 to 2

Figure 12 - 38 Master Operation in Multi-Master System (2/3)



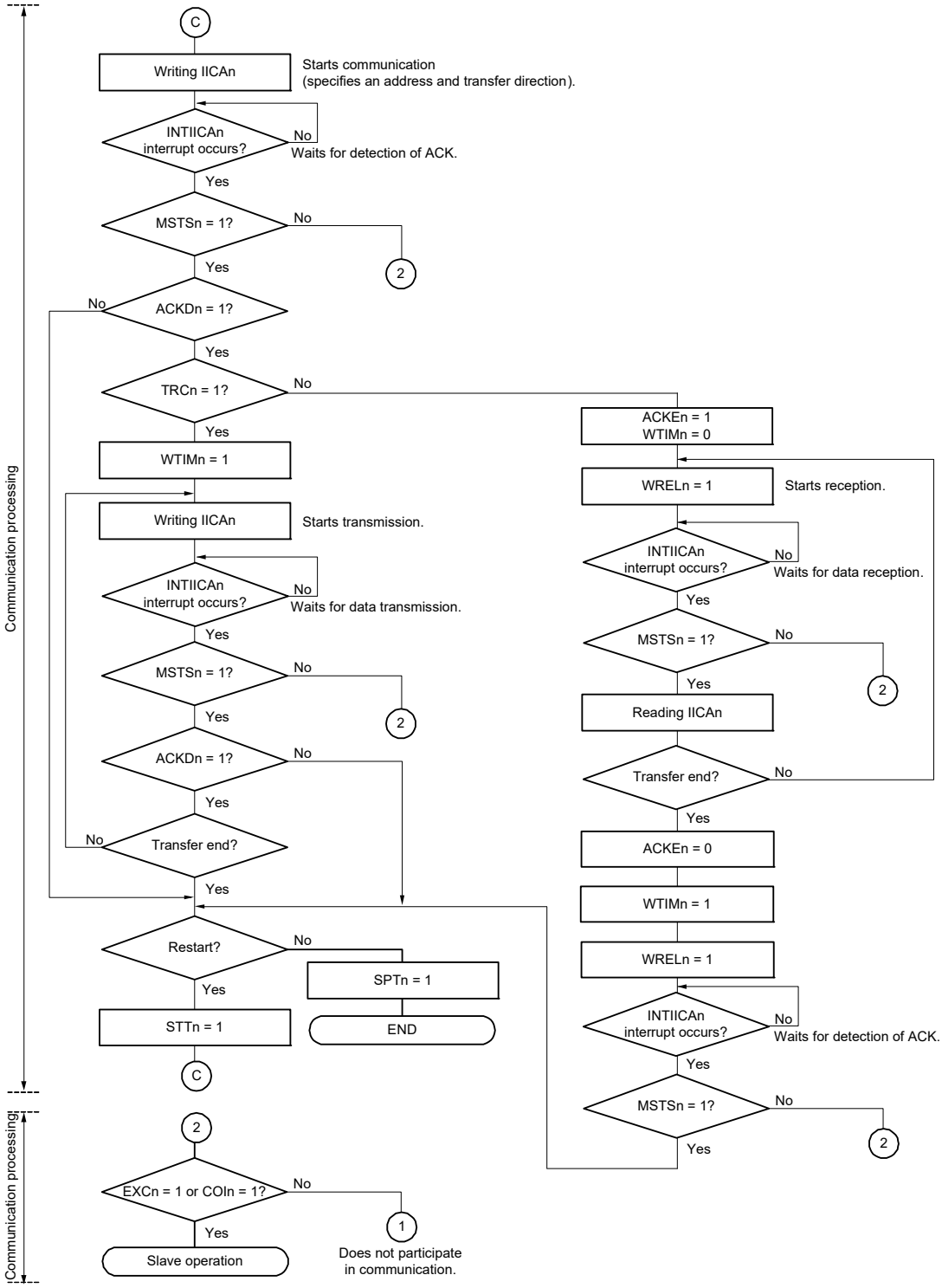
Note The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + $t_F \times 2$



Remark1. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operating clock frequency

Remark2. n = 0 to 2

Figure 12 - 39 Master Operation in Multi-Master System (3/3)



Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark 2. To use the device as a master in a multi-master system, read the MSTSn bit each time interrupt INTIICAn has occurred to check the arbitration result.

Remark 3. To use the device as a slave in a multi-master system, check the status by using the IICA status register n (IICSn) and IICA flag register n (IICFn) each time interrupt INTIICAn has occurred, and determine the processing to be performed next.

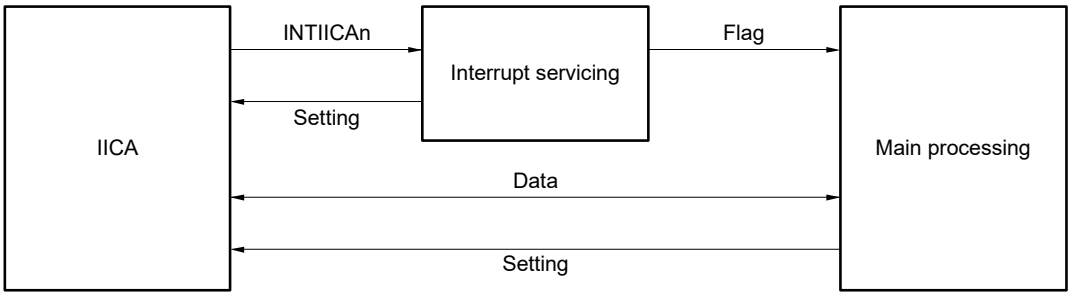
Remark 4. n = 0 to 2

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICAn 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 extension code is not supported for data communication. It is also assumed that the INTIICAn interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICAn.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- •Clear mode: Status in which data communication is not performed
- •Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICAn 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 TRCn bit.

Remark n = 0 to 2

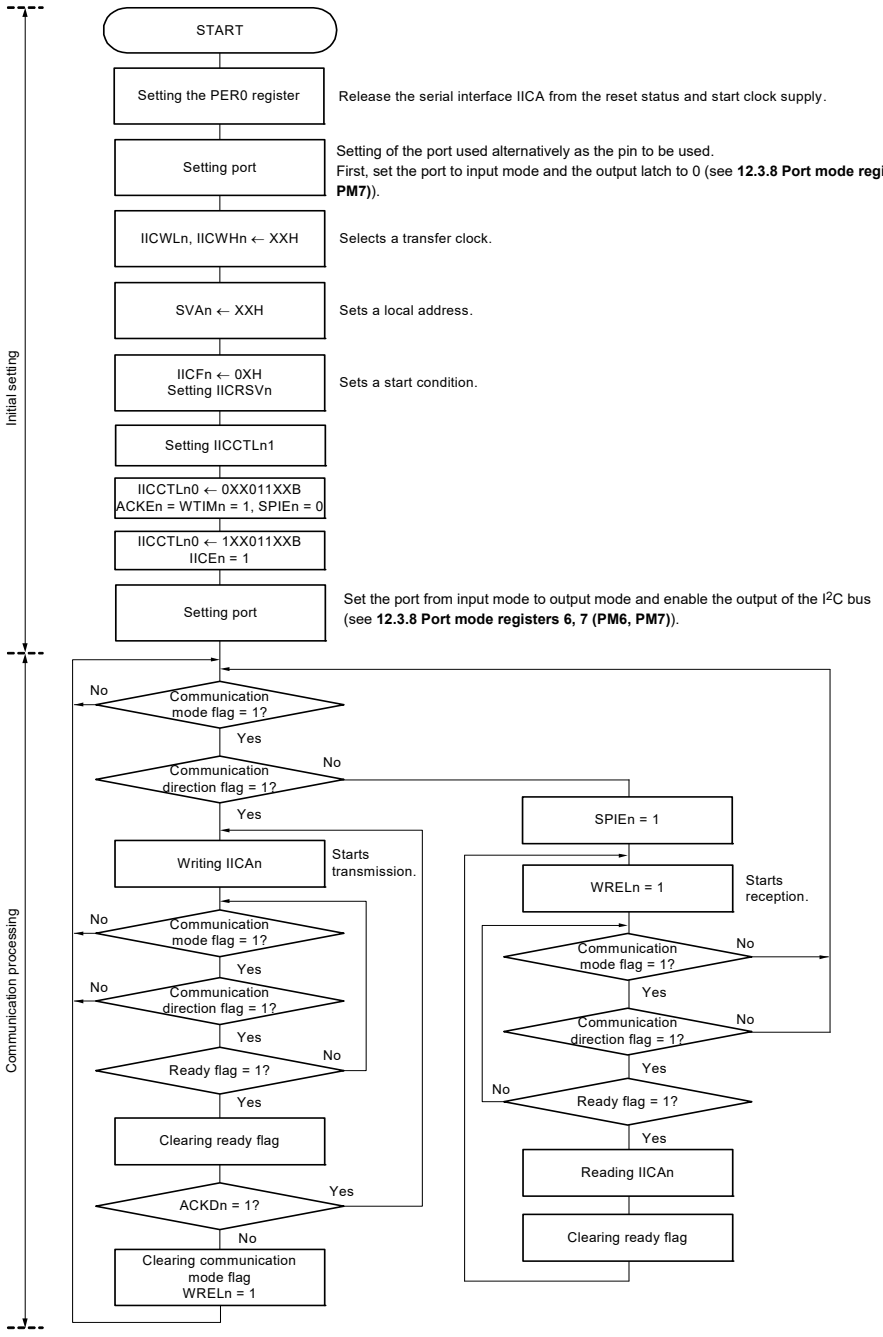
The main processing of the slave operation is explained next.

Start serial 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 no longer returns ACK. If ACK is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, ACK is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 12 - 40 Slave Operation Flowchart (1)



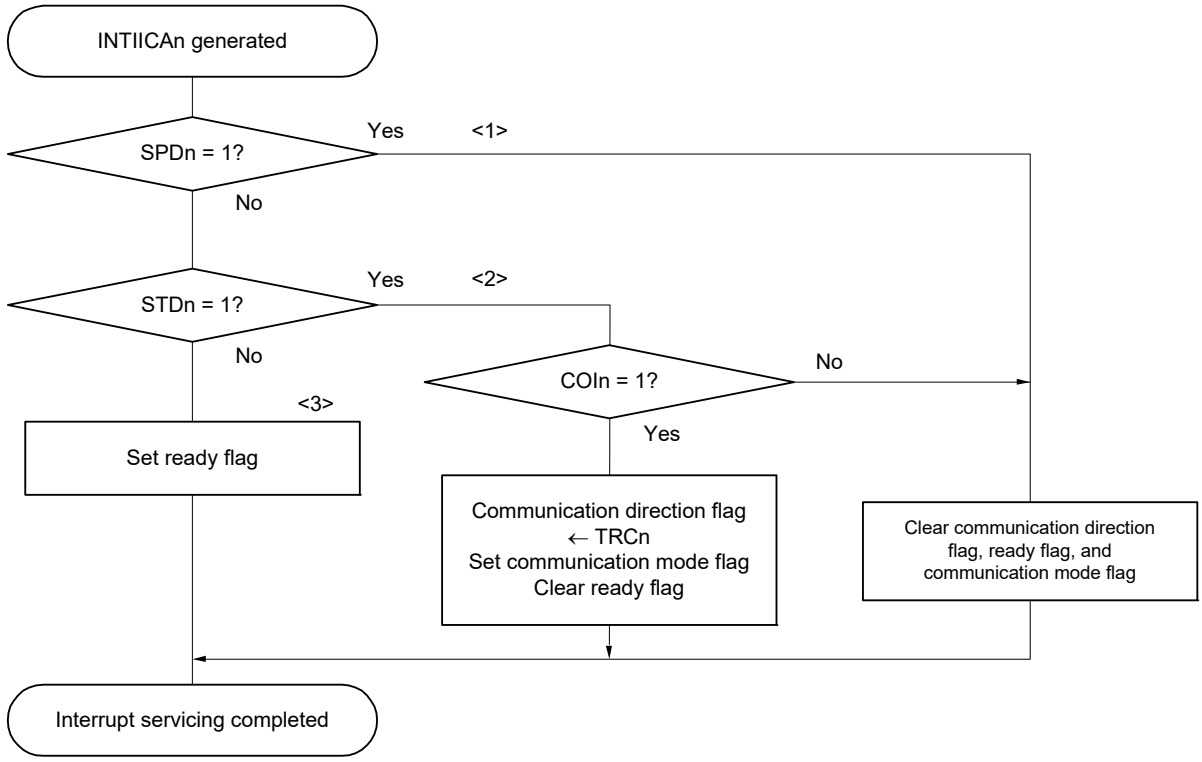
Remark1. Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.
Remark2. n = 0 to 2

An example of the processing procedure of the slave with the INTIICAn interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICAn 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 cancelled, and processing returns from the interrupt (the ready flag is cleared).
<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in Figure 12 - 41 Slave Operation Flowchart (2).

Figure 12 - 41 Slave Operation Flowchart (2)



Remark n = 0 to 2

12.5.17 Timing of I²C interrupt request (INTIICAn) occurrence

The timing of transmitting or receiving data and generation of interrupt request signal INTIICAn, and the value of the IICA status register n (IICSn) when the INTIICAn signal is generated are shown below.

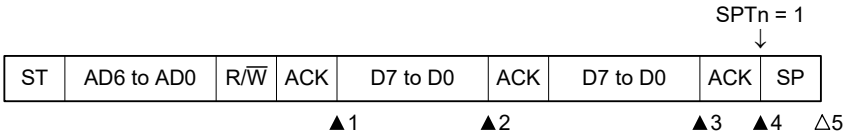
Remark 1. ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
ACK: Acknowledge
D7 to D0: Data
SP: Stop condition

Remark 2. n = 0 to 2

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B

▲3: IICSn = 1000×000B (Sets the WTIMn bit to 1) Note

▲4: IICSn = 1000××00B (Sets the SPTn bit to 1) Note

△5: IICSn = 00000001B

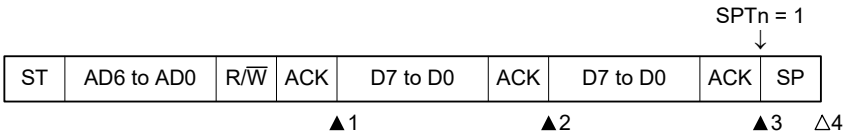
Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B

▲3: IICSn = 1000××00B (Sets the SPTn bit to 1)

△4: IICSn = 00000001B

Remark ▲: Always generated

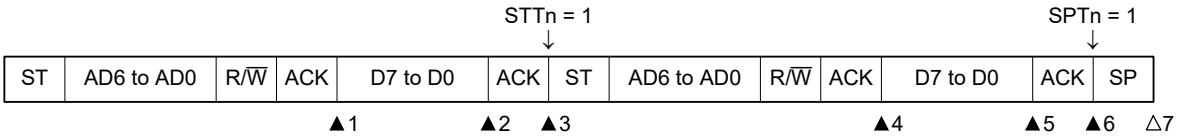
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0 to 2

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1) Note 1

▲3: IICSn = 1000××00B (Clears the WTIMn bit to 0 Note 2, sets the STTn bit to 1)

▲4: IICSn = 1000×110B

▲5: IICSn = 1000×000B (Sets the WTIMn bit to 1) Note 3

▲6: IICSn = 1000××00B (Sets the SPTn bit to 1)

△7: IICSn = 00000001B

Note 1. To generate a start condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Note 2. Clear the WTIMn bit to 0 to restore the original setting.

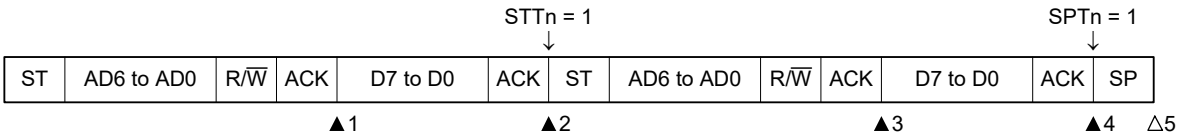
Note 3. To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000××00B (Sets the STTn bit to 1)

▲3: IICSn = 1000×110B

▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)

△5: IICSn = 00000001B

Remark ▲: Always generated

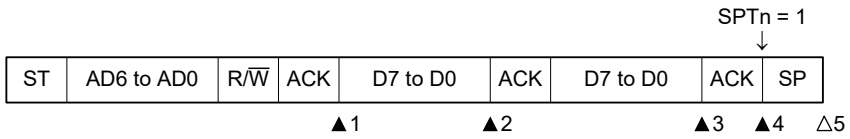
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0 to 2

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIMn = 0



▲1: IICSn = 1010×110B

▲2: IICSn = 1010×000B

▲3: IICSn = 1010×000B (Sets the WTIMn bit to 1) Note

▲4: IICSn = 1010××00B (Sets the SPTn bit to 1)

△5: IICSn = 00000001B

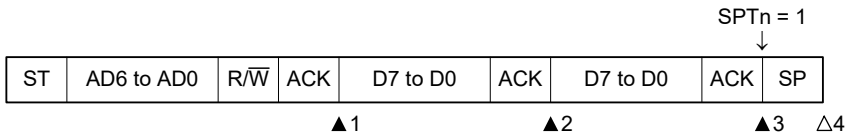
Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1010×110B

▲2: IICSn = 1010×100B

▲3: IICSn = 1010××00B (Sets the SPTn bit to 1)

△4: IICSn = 00001001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

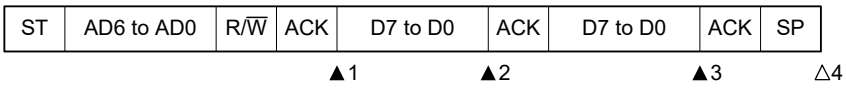
×: Don't care

Remark n = 0 to 2

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) When WTIMn = 0



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×000B

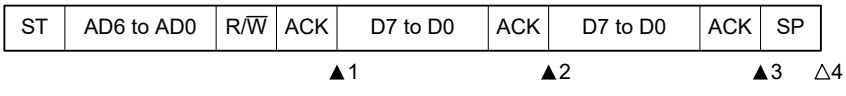
△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×100B

▲3: IICSn = 0001××00B

△4: IICSn = 00000001B

Remark ▲: Always generated

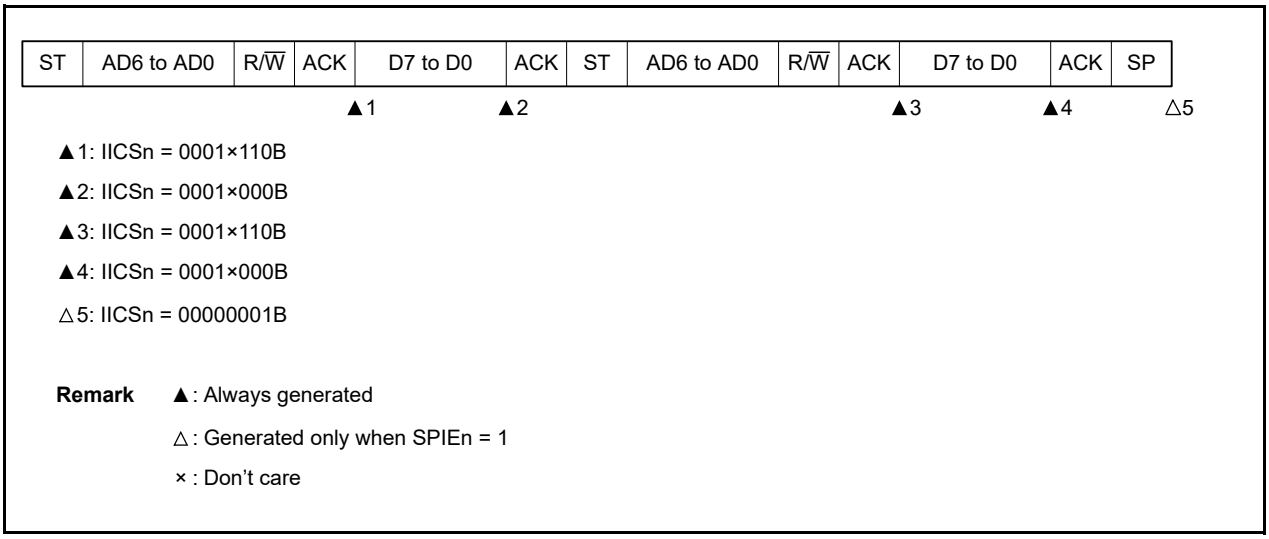
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0 to 2

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, matches with SVAn)



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001×000B

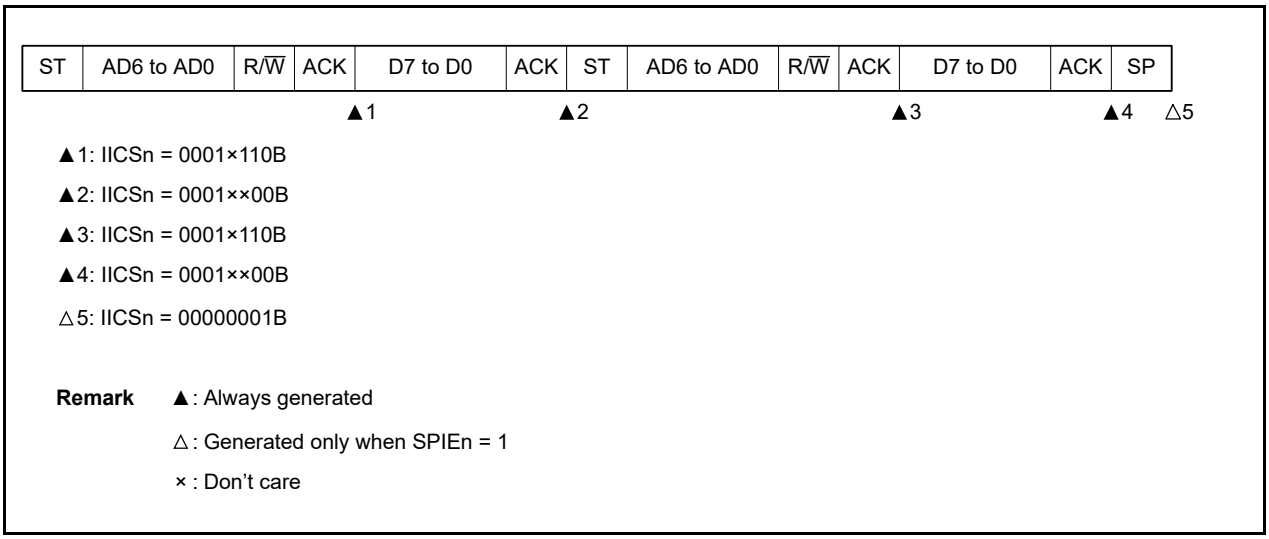
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, matches with SVAn)



▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

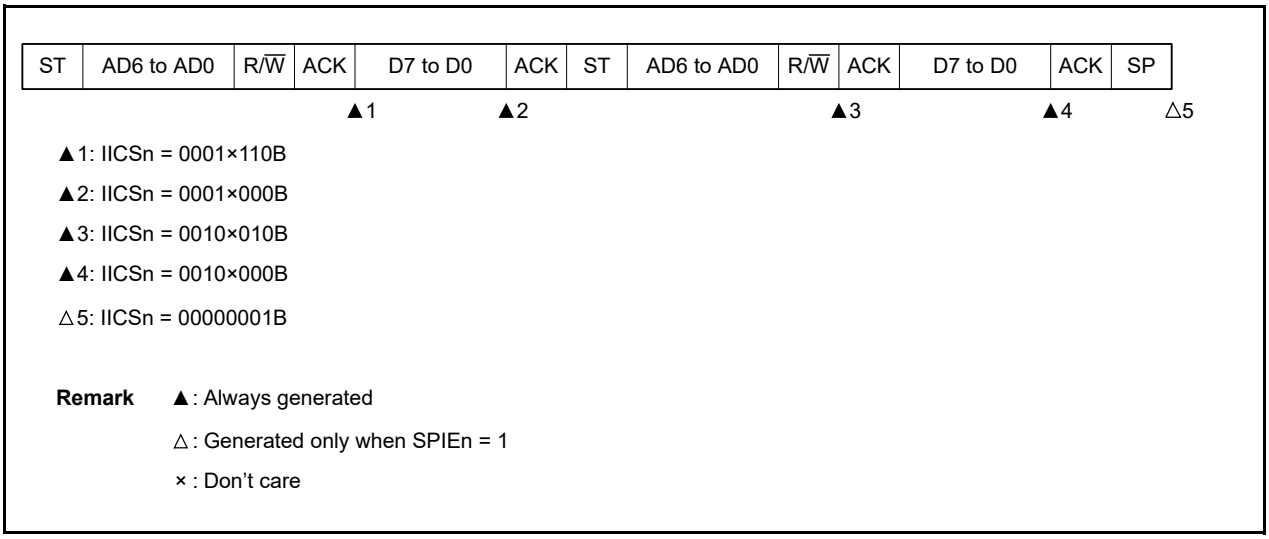
△: Generated only when SPIEn = 1

×: Don't care

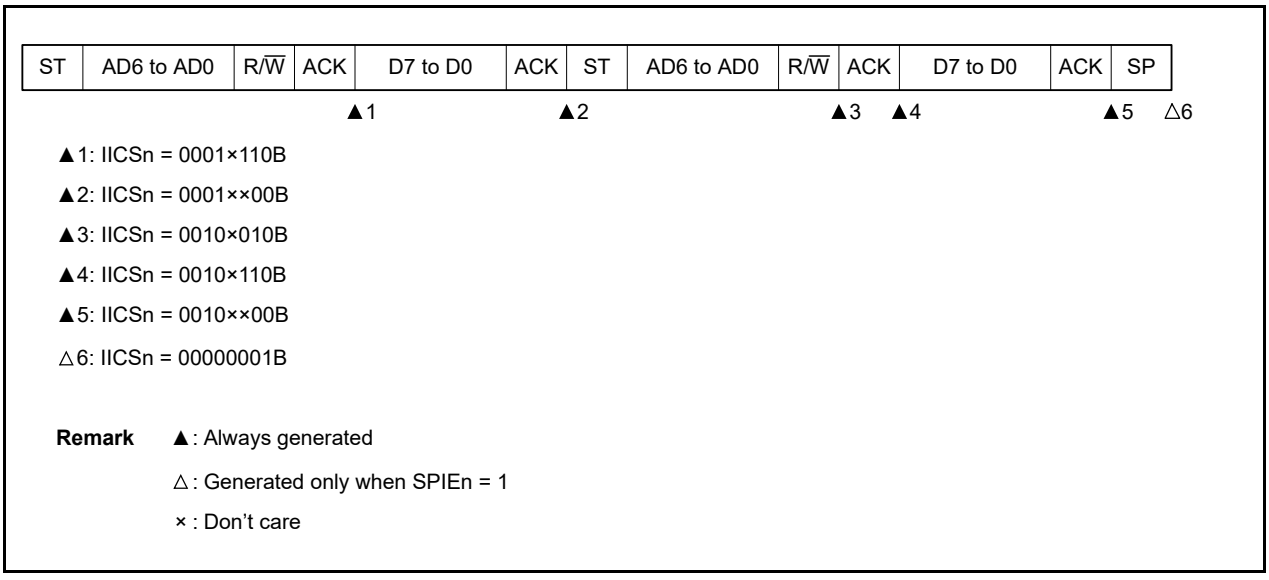
Remark n = 0 to 2

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= extension code))



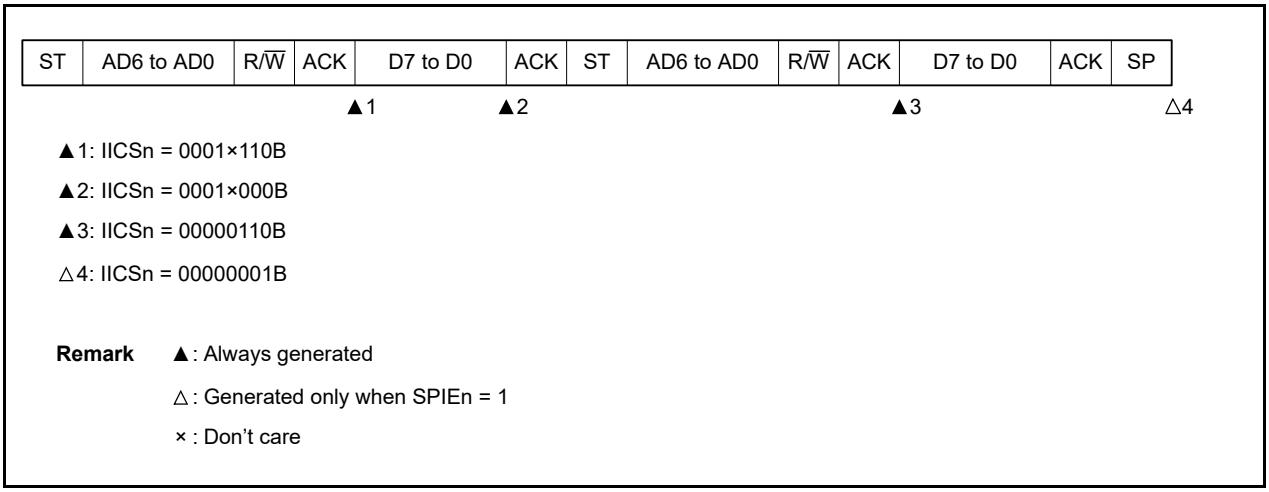
(ii) When WTIMn = 1 (after restart, does not match address (= extension code))



Remark n = 0 to 2

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= not extension code))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 00000110B

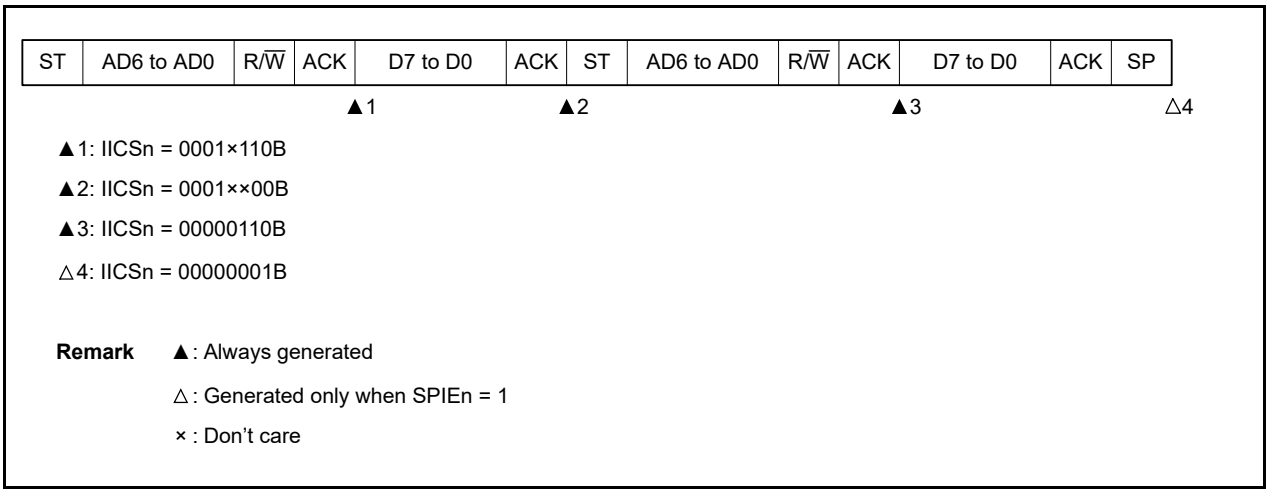
△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, does not match address (= not extension code))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 00000110B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

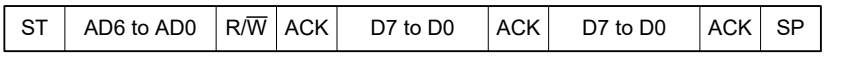
Remark n = 0 to 2

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

(i) When WTIMn = 0



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×000B

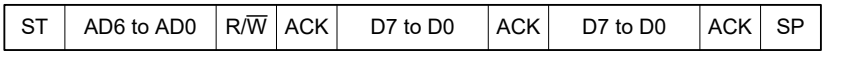
△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010×100B

▲4: IICSn = 0010××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

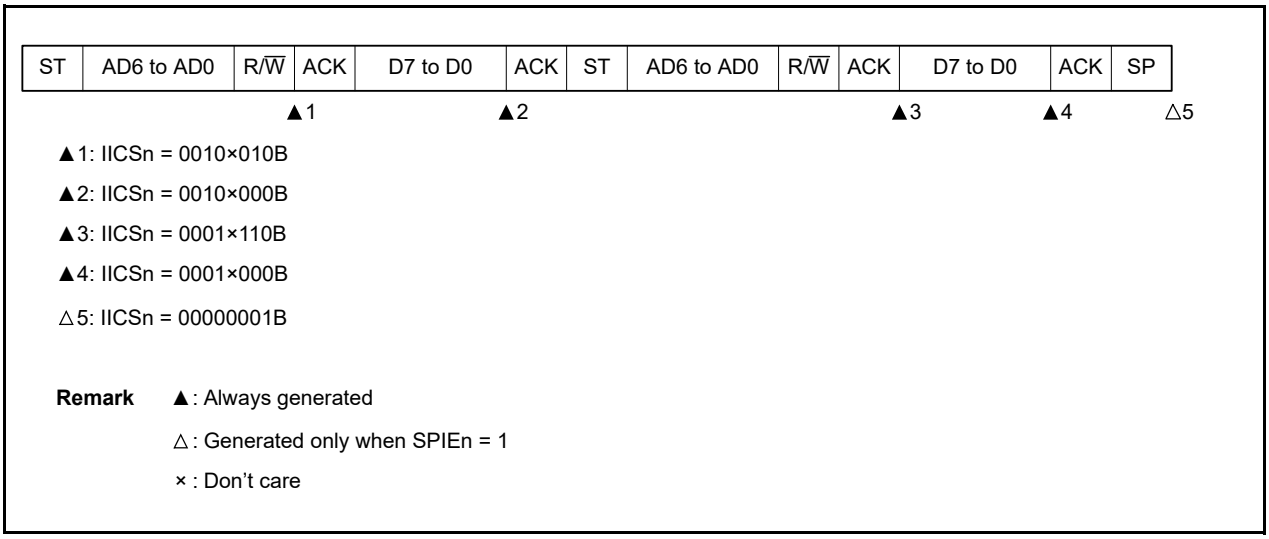
△: Generated only when SPIEn = 1

×: Don't care

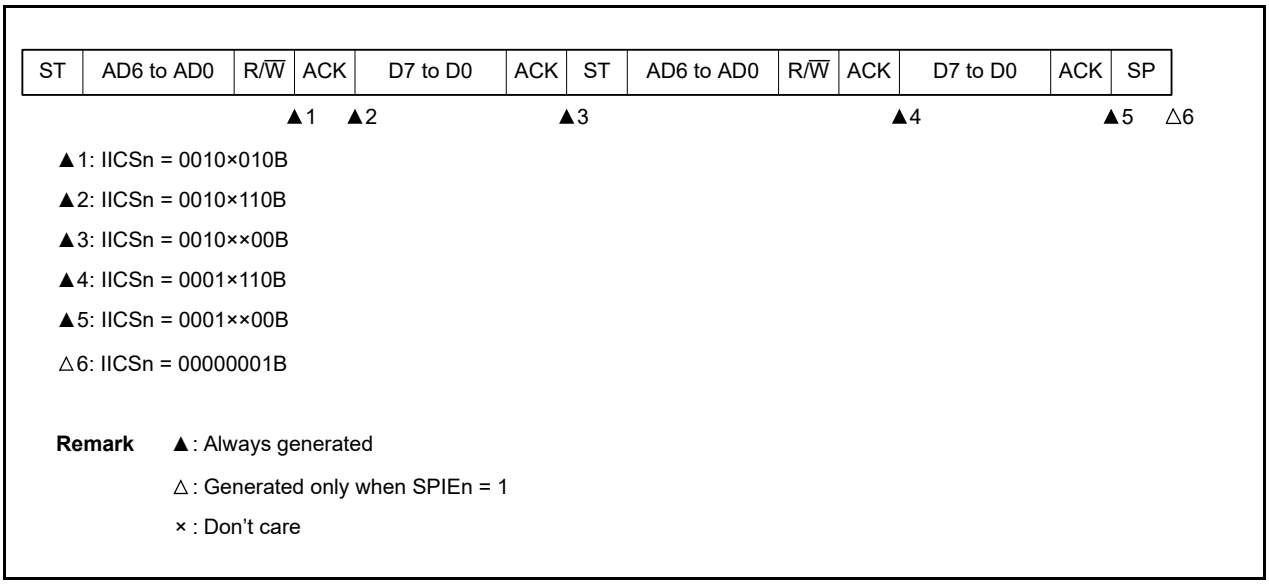
Remark n = 0 to 2

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, matches SVAn)



(ii) When WTIMn = 1 (after restart, matches SVAn)



Remark n = 0 to 2

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, extension code reception)

▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×000B

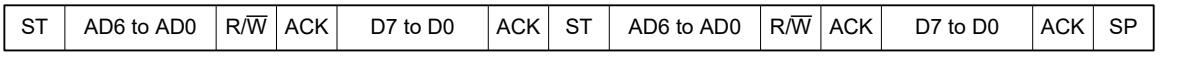
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, extension code reception)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010××00B

▲4: IICSn = 0010×010B

▲5: IICSn = 0010×110B

▲6: IICSn = 0010××00B

△7: IICSn = 00000001B

Remark ▲: Always generated

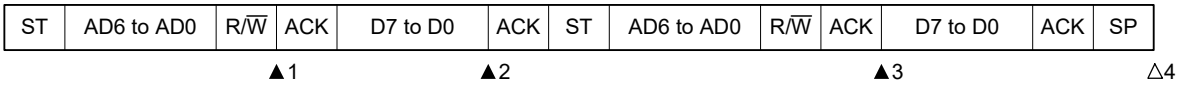
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0 to 2

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= not extension code))



▲1: IICSn = 0010 × 010B

▲2: IICSn = 0010 × 000B

▲3: IICSn = 00000 × 10B

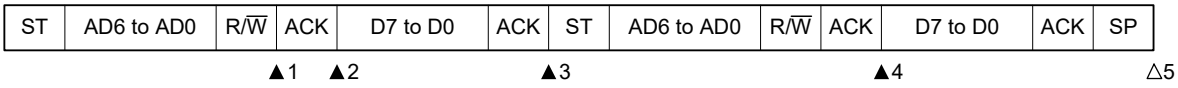
△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, does not match address (= not extension code))



▲1: IICSn = 0010 × 010B

▲2: IICSn = 0010 × 110B

▲3: IICSn = 0010 × × 00B

▲4: IICSn = 00000 × 10B

△5: IICSn = 00000001B

Remark ▲: Always generated

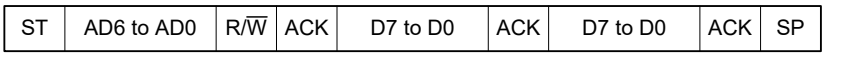
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0 to 2

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICSn = 00000001B

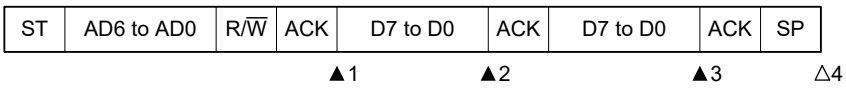
Remark △: Generated only when SPIEn = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

(i) When WTIMn = 0



▲1: IICSn = 0101×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×000B

△4: IICSn = 00000001B

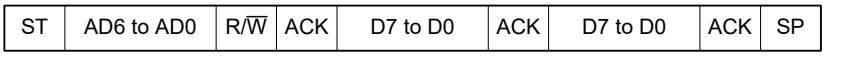
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0 to 2

(ii) When WTIMn = 1



▲1: IICSn = 0101×110B

▲2: IICSn = 0001×100B

▲3: IICSn = 0001××00B

△4: IICSn = 00000001B

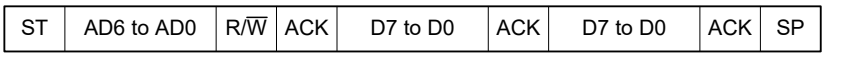
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIMn = 0



▲1: IICSn = 0110×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×000B

△4: IICSn = 00000001B

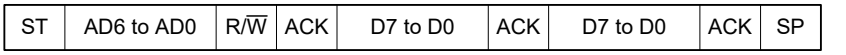
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0 to 2

(ii) When WTIMn = 1



▲1 ▲2 ▲3 ▲4 △5

▲1: IICSn = 0110×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010×100B

▲4: IICSn = 0010××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

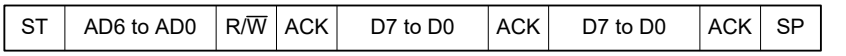
△: Generated only when SPIEn = 1

×: Don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data (when WTIMn = 1)



▲1 △2

▲1: IICSn = 01000110B

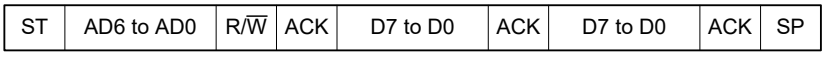
△2: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

Remark n = 0 to 2

(b) When arbitration loss occurs during transmission of extension code



▲1 △2

▲1: IICSn = 0110×010B

Sets LRELn = 1 by software

△2: IICSn = 00000001B

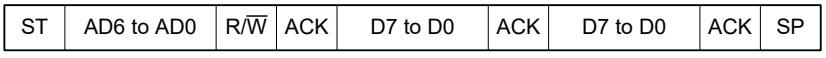
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(c) When arbitration loss occurs during transmission of data

(i) When WTIMn = 0



▲1 ▲2 △3

▲1: IICSn = 10001110B

▲2: IICSn = 01000000B

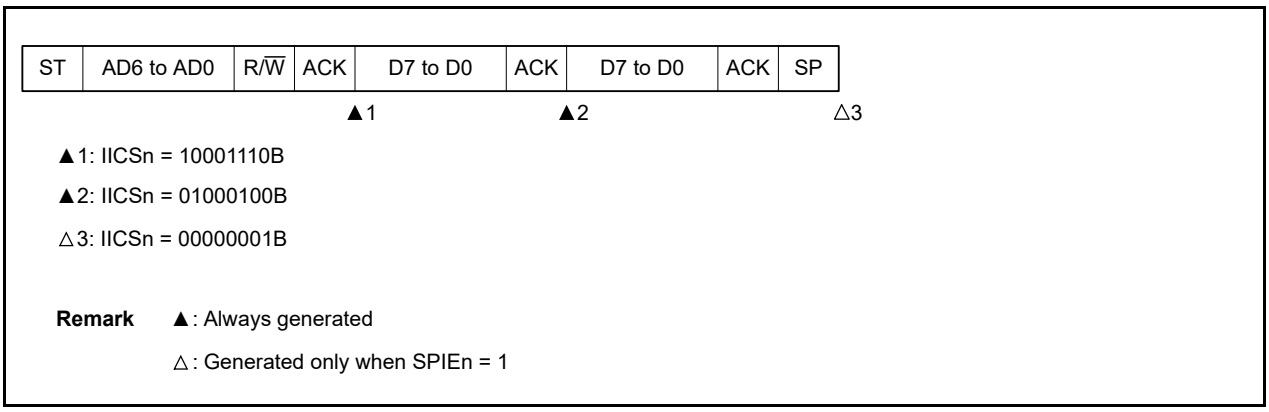
△3: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

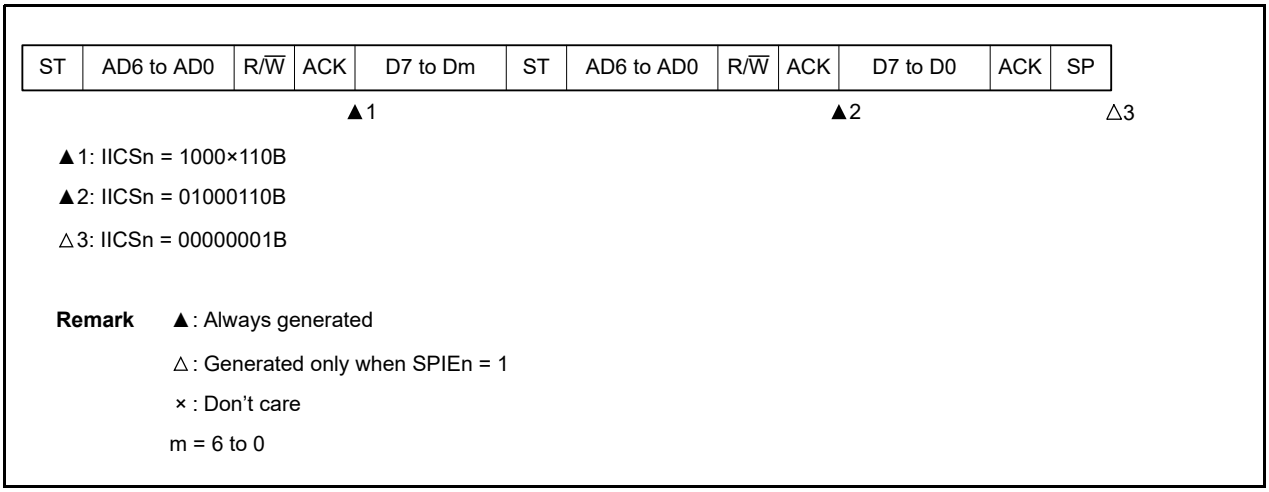
Remark n = 0 to 2

(ii) When WTIMn = 1



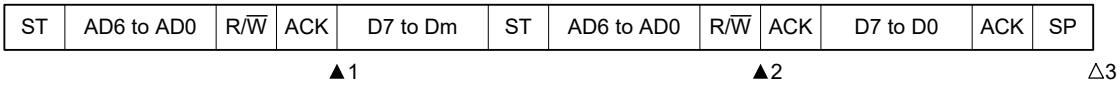
(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatches with SVAn)



Remark n = 0 to 2

(ii) Extension code



▲1: IICSn = 1000×110B

▲2: IICSn = 01100010B

Sets LRELn = 1 by software

△3: IICSn = 00000001B

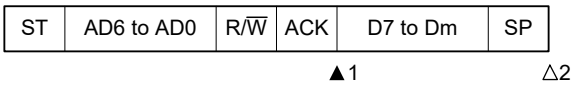
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

m = 6 to 0

(e) When loss occurs due to stop condition during data transfer



▲1: IICSn = 10000110B

△2: IICSn = 01000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

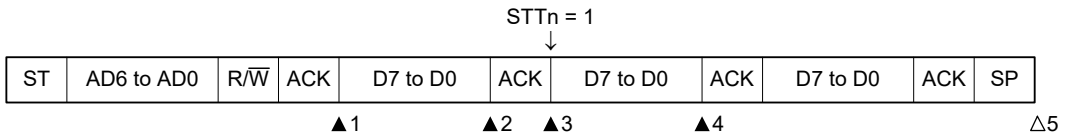
×: Don't care

m = 6 to 0

Remark n = 0 to 2

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)

▲4: IICSn = 01000000B

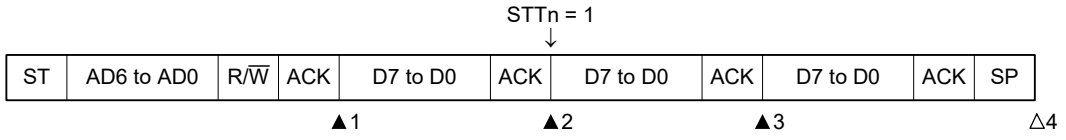
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B (Sets the STTn bit to 1)

▲3: IICSn = 01000100B

△4: IICSn = 00000001B

Remark ▲: Always generated

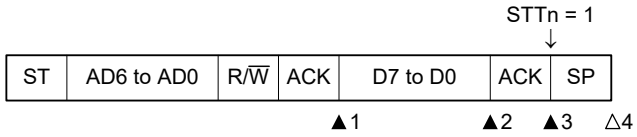
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0 to 2

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000××00B (Sets the STTn bit to 1)

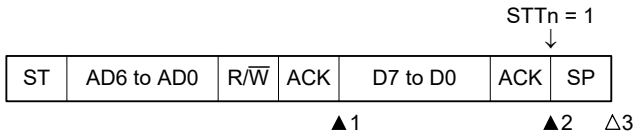
△4: IICSn = 01000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000××00B (Sets the STTn bit to 1)

△3: IICSn = 01000001B

Remark ▲: Always generated

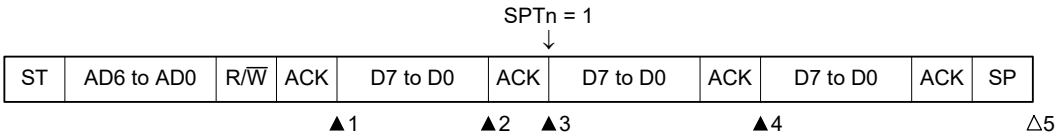
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0 to 2

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)

▲4: IICSn = 01000100B

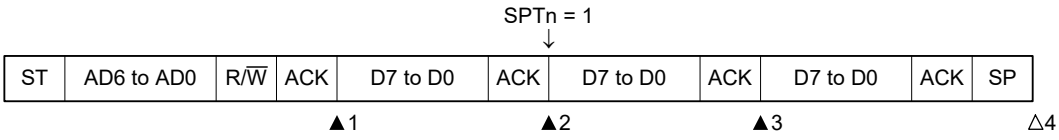
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B (Sets the SPTn bit to 1)

▲3: IICSn = 01000100B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0 to 2

12.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRCn 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.

Figures 12 - 42 to 12 - 48 show timing charts of the data communication.

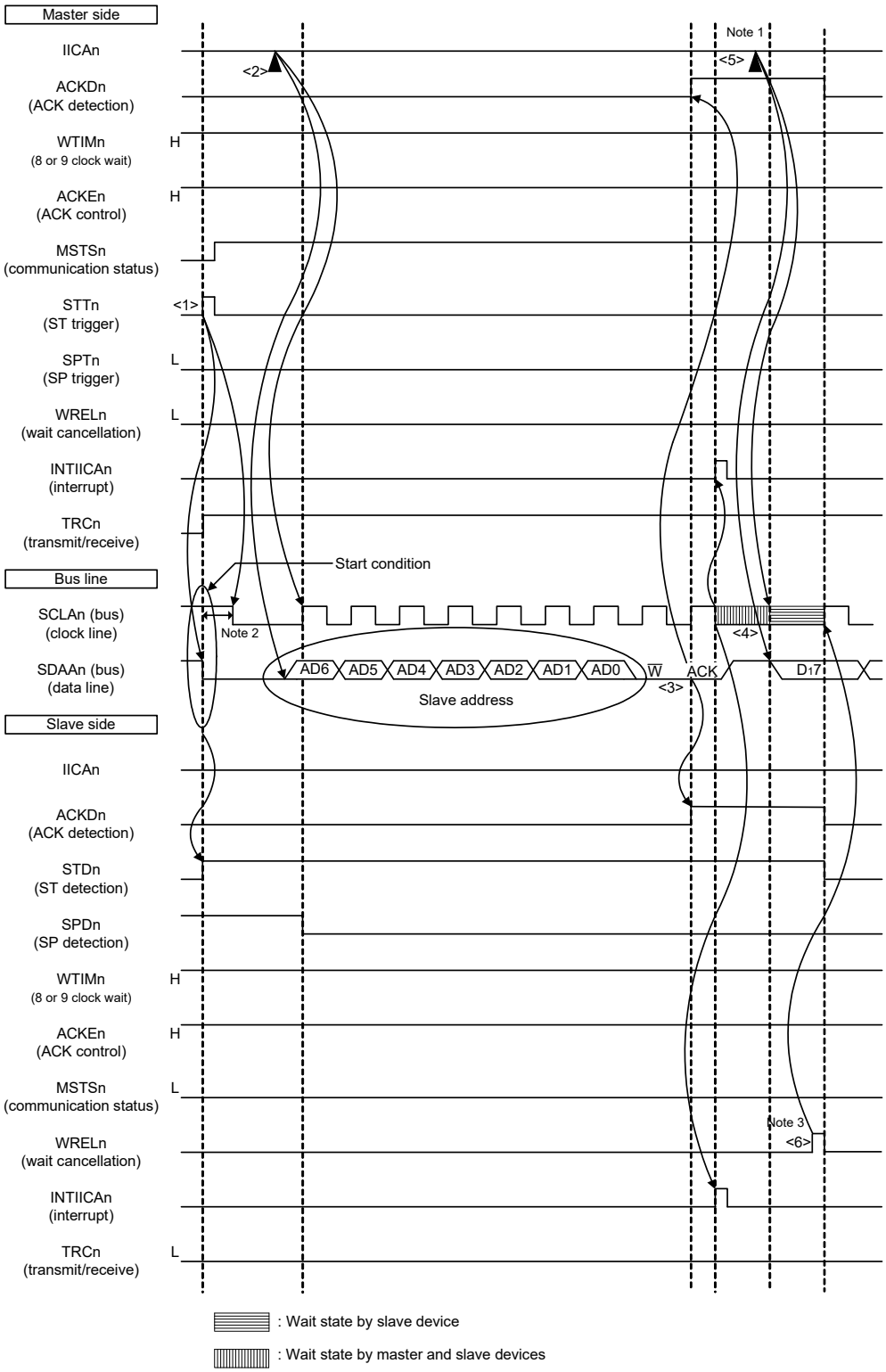
The IICA shift register n (IICAn)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin.

Data input via the SDAAn pin is captured into IICAn at the rising edge of SCLAn.

Remark n = 0 to 2

Figure 12 - 42 Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/4)

(1) Start condition ~ address ~ data



Note 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a master device.
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 releasing wait state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.
Remark n = 0 to 2

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 12 - 42 are explained below.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 changes SDAAn from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communication status (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The master device writes the data to transmit to the IICAn register and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WRELn = 1), the master device starts transferring data to the slave device.

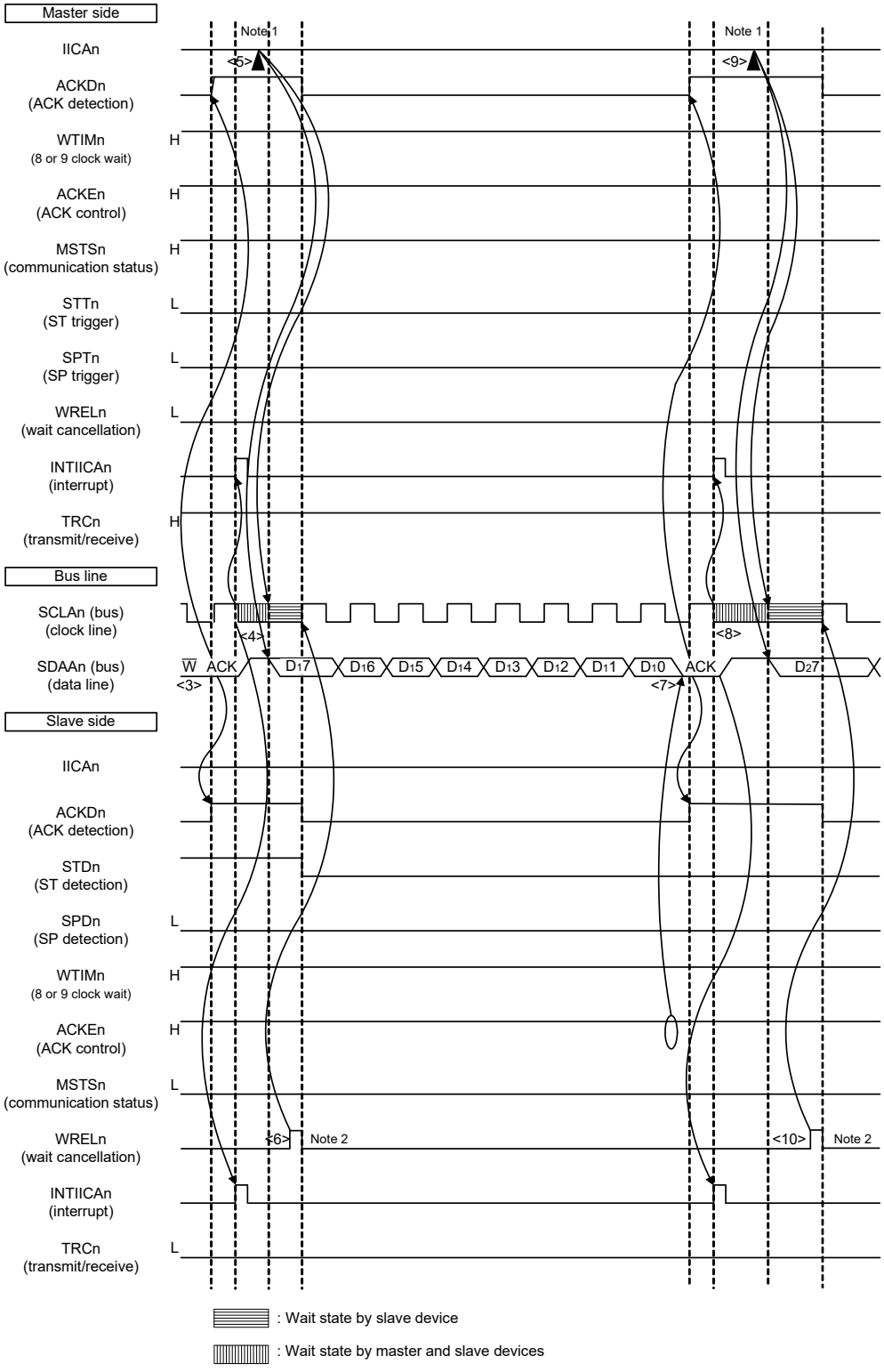
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in Figures 12 - 42 to 12 - 44 represent the entire procedure for communicating data using the I^2C bus. Figure 12 - 42 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12 - 43 (3) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12 - 44 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0 to 2

Figure 12 - 43 Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/4)

(3) Address ~ data ~ data



Note 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a master device.
Note 2. For releasing wait state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.
Remark n = 0 to 2

The meanings of <3> to <10> in (3) Address ~ data ~ data in Figure 12 - 43 are explained below.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WRELn = 1), the master device starts transferring data to the slave device.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<9> The master device writes the data to transmit to the IICAn register and releases the wait status that it set by the master device.

<10>The slave device reads the received data and releases the wait status (WRELn = 1). The master device then starts transferring data to the slave device.

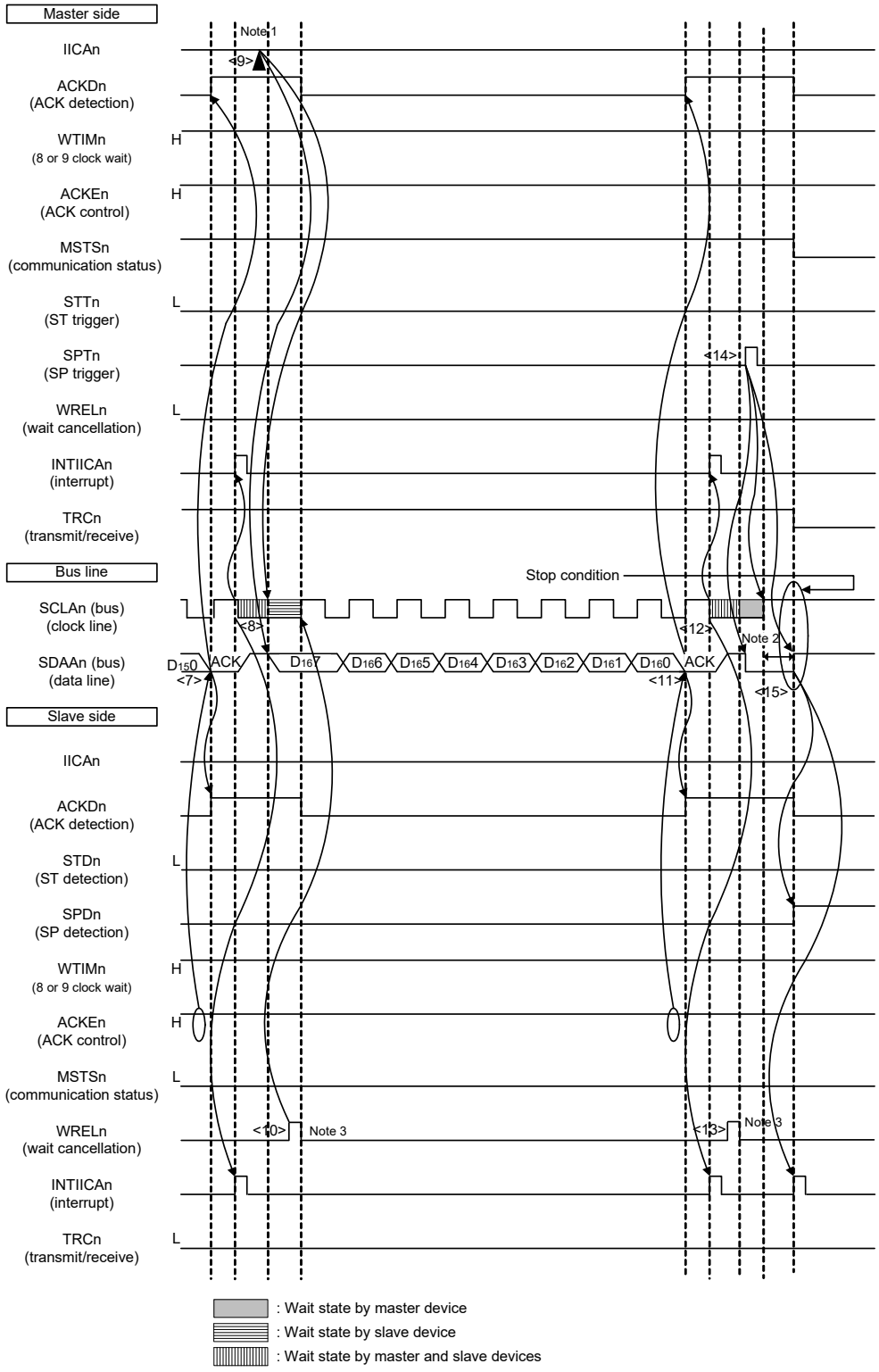
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in Figures 12 - 42 to 12 - 44 represent the entire procedure for communicating data using the I^2C bus. Figure 12 - 42 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12 - 43 (3) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12 - 44 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0 to 2

Figure 12 - 44 Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/4)

(3) Data ~ data ~ stop condition



Note 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a master device.
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 releasing wait state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.
Remark n = 0 to 2

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 12 - 44 are explained below.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the wait status that it set by the master device.

<10>The slave device reads the received data and releases the wait status (WRELn = 1). The master device then starts transferring data to the slave device.

<11>When data transfer is complete, the slave device (ACKEn =1) sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<12>The master device and slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<13>The slave device reads the received data and releases the wait status (WRELn = 1).

<14> By the master device setting a stop condition trigger (SPTn = 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 (i.e. SCLAn =1 changes SDAAn from 0 to 1).

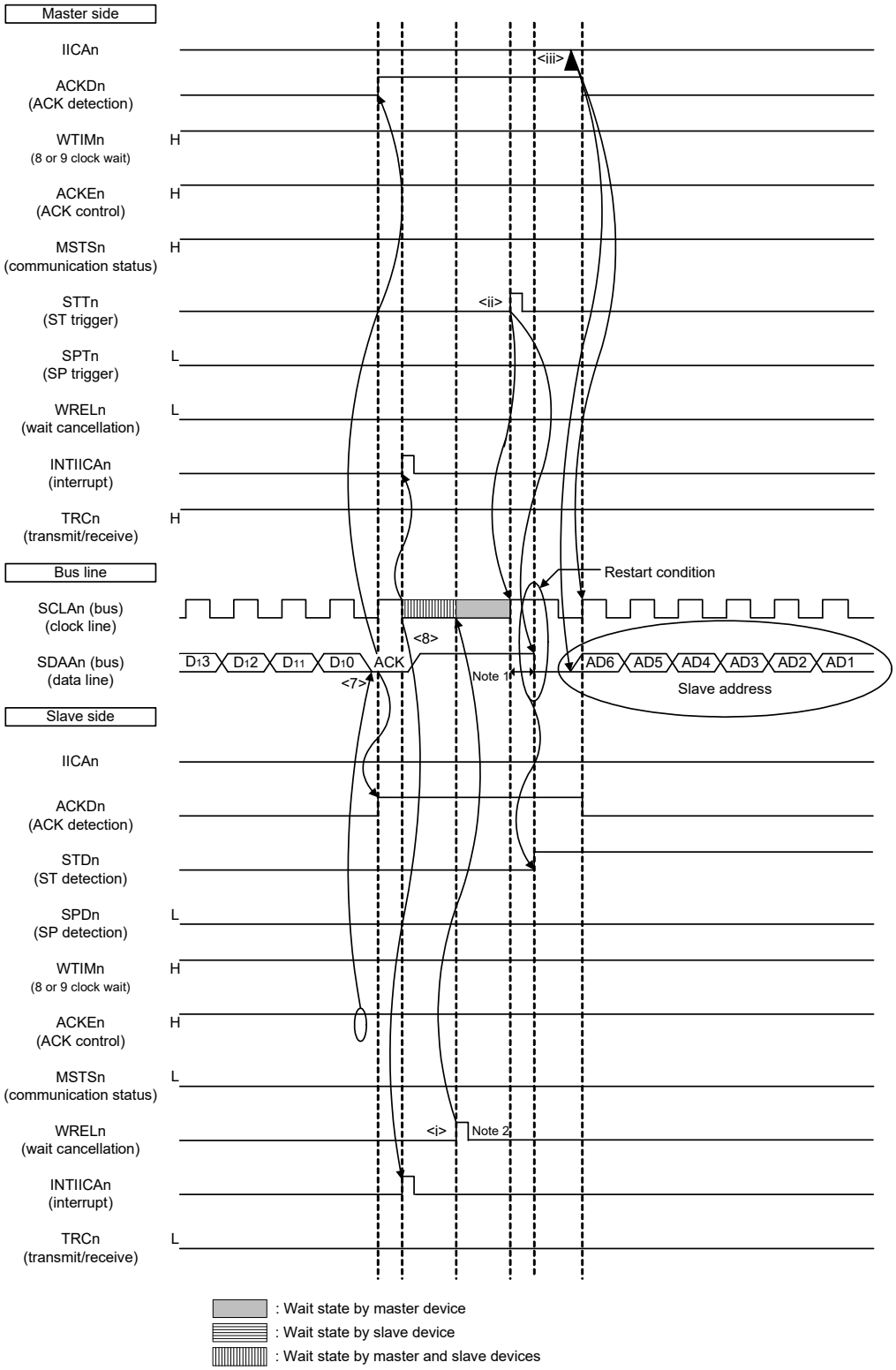
<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <15> in Figures 12 - 42 to 12 - 44 represent the entire procedure for communicating data using the I^2C bus. Figure 12 - 42 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12 - 43 (3) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12 - 44 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0 to 2

Figure 12 - 45 Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (4/4)

(3) Data ~ restart condition ~ address



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 releasing wait state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0 to 2

The following describes the operations in Figure 12 - 45 (3) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <iii>, the data transmission step.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<i> The slave device reads the received data and releases the wait status (WRELn = 1).

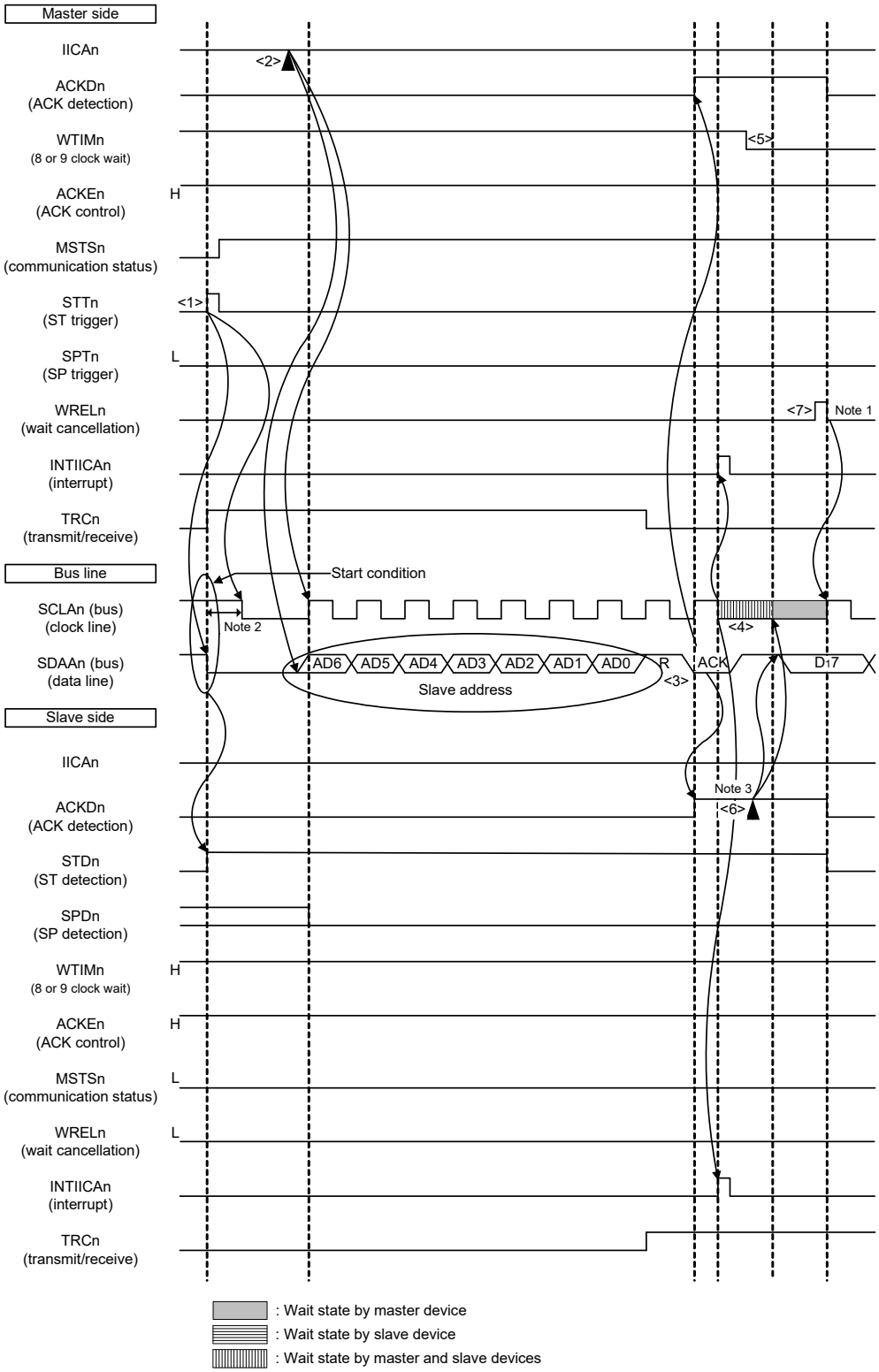
<ii> The start condition trigger is set again by the master device (STTn = 1) and a start condition (i.e. SCLAn =1 changes SDAAn from 1 to 0) is generated once the bus clock line goes high (SCLAn = 1) and the bus data line goes low (SDAAn = 0) after the restart condition setup time has elapsed. When 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 (IICAn) enables the slave address to be transmitted.

Remark n = 0 to 2

Figure 12 - 46 Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



Note 1. For releasing wait state during reception of a master device, write “FFH” to IICAn or set the WRELn 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. Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a slave device.
Remark n = 0 to 2

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 12 - 46 are explained below.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn =1 changes SDAAn from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communication status (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIMn = 0).

<6> The slave device writes the data to transmit to the IICAn register and releases the wait status that it set by the slave device.

<7> The master device releases the wait status (WRELn = 1) and starts transferring data from the slave device to the master device.

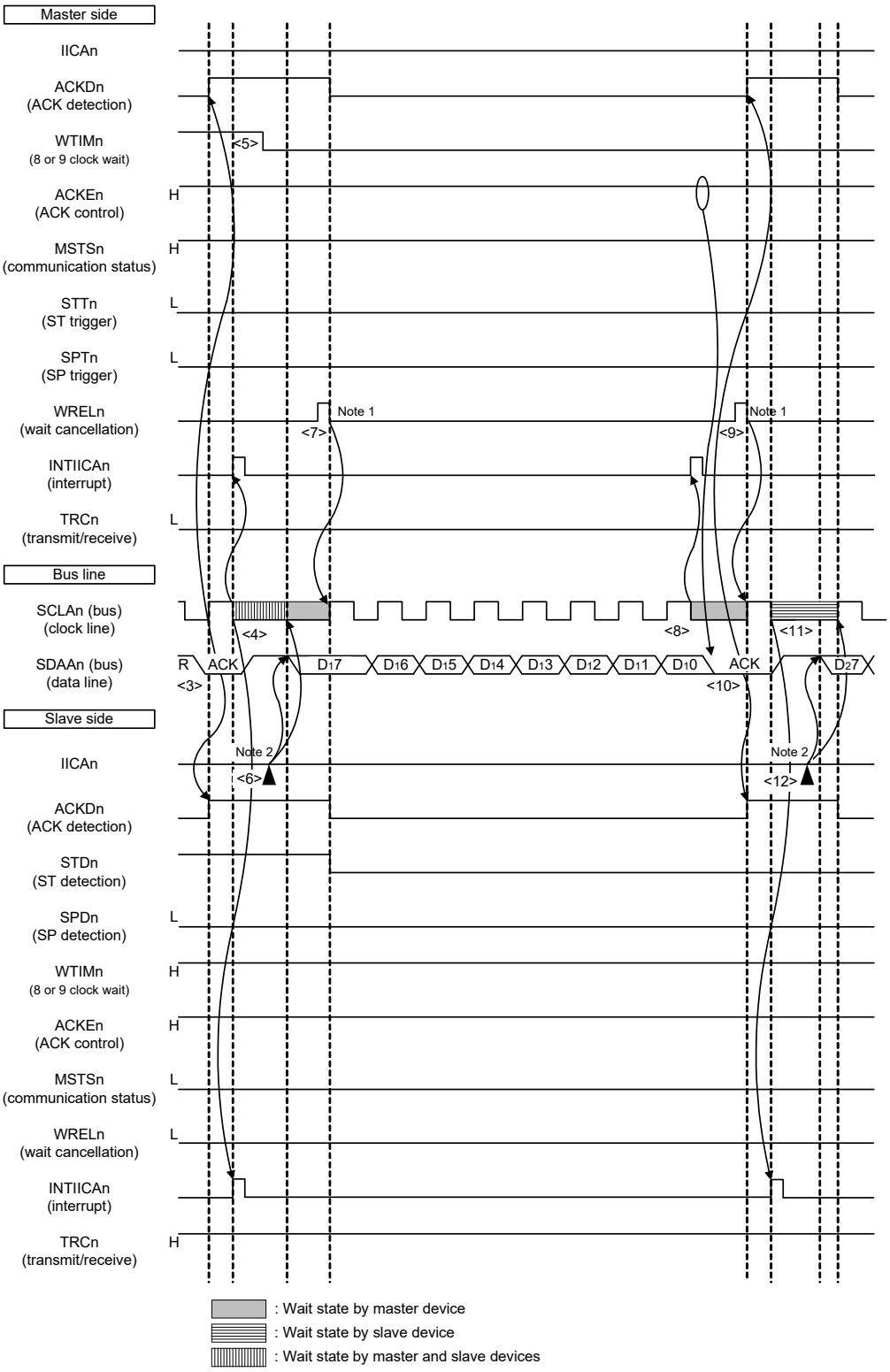
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in Figures 12 - 46 to 12 - 48 represent the entire procedure for communicating data using the I^2C bus. Figure 12 - 46 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12 - 47 (3) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12 - 48 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0 to 2

Figure 12 - 47 Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

(3) Address ~ data ~ data



Note 1. For releasing wait state during reception of a master device, write “FFH” to IICAn or set the WRELn bit.
Note 2. Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a slave device.
Remark n = 0 to 2

The meanings of <3> to <12> in (3) Address ~ data ~ data in Figure 12 - 47 are explained below.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The master device changes the timing of the wait status to the 8th clock (WTIMn = 0).

<6> The slave device writes the data to transmit to the IICA shift register n (IICAn) and releases the wait status that it set by the slave device.

<7> The master device releases the wait status (WRELn = 1) and starts transferring data from the slave device to the master device.

<8> The master device sets a wait status (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 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 wait status (WRELn = 1).

<10>The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.

<11>The slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).

<12>By the slave device writing the data to transmit to the IICAn register, the wait status set by the slave device is released. The slave device then starts transferring data to the master device.

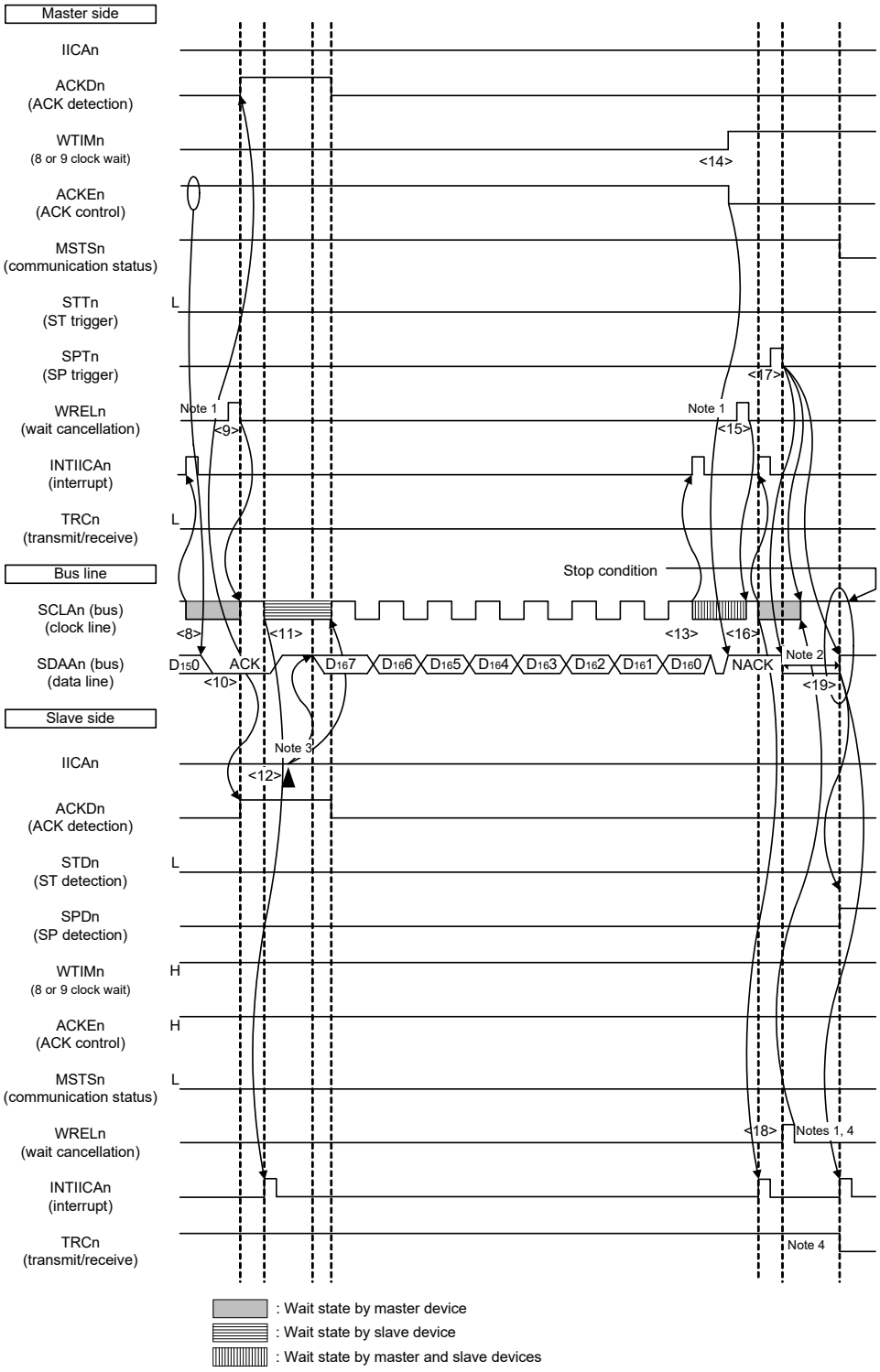
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in Figures 12 - 46 to 12 - 48 represent the entire procedure for communicating data using the I^2C bus. Figure 12 - 46 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12 - 47 (3) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12 - 48 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0 to 2

Figure 12 - 48 Example of Slave to Master Communication
(When 8-Clock and 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/3)

(3) Data ~ data ~ stop condition



Note 1. To cancel a wait state, write “FFH” to IICAn or set the WRELn 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. Write data to IICAn, not setting the WRELn bit, in order to cancel a wait state during transmission by a slave device.
Note 4. If a wait state during transmission by a slave device is canceled by setting the WRELn bit, the TRCn bit will be cleared.
Remark n = 0 to 2

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 12 - 48 are explained below.

<8> The master device sets a wait status (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 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 wait status (WRELn = 1).

<10>The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.

<11>The slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).

<12>By the slave device writing the data to transmit to the IICA register, the wait status 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 (INTIICAn: end of transfer) at the falling edge of the 8th clock, and sets a wait status (SCLAn = 0). Because ACK control (ACKEn = 1) is performed, the bus data line is at the low level (SDAAn = 0) at this stage.

<14>The master device sets NACK as the response (ACKEn = 0) and changes the timing at which it sets the wait status to the 9th clock (WTIMn = 1).

<15>If the master device releases the wait status (WRELn = 1), the slave device detects the NACK (ACKDn = 0) at the rising edge of the 9th clock.

<16>The master device and slave device set a wait status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<17> When the master device issues a stop condition (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the master device releases the wait status. The master device then waits until the bus clock line is set (SCLAn = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the wait status (WRELn = 1) to end communication. Once the slave device releases the wait status, 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 (i.e. SCLAn =1 changes SDAAn from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <19> in Figures 12 - 46 to 12 - 48 represent the entire procedure for communicating data using the I^2C bus. Figure 12 - 46 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12 - 47 (3) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12 - 48 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0 to 2

CHAPTER 13 USB 2.0 HOST/FUNCTION MODULE (USB)

The availability of each pin of USB port, VBUS supply enable output, and overcurrent detection input differs, depending on the product.

	R9A02G0150	R9A02G0151
UDP0 pin	✓	—
UDM0 pin	✓	—
UVBUS pin	✓	—
UDP1 pin	✓	—
UDM1 pin	✓	—
UVBUSEN0 pin	✓	—
UVBUSEN1 pin	✓	—
UOVRCUR0 pin	✓	—
UOVRCUR1 pin	✓	—
UVDD pin	✓	✓Note

Note Connect directory to VDD, or draw power from an external 3.3 V supply.

13.1 Functions of USB 2.0 Host/Function Module

The R9A02G0150 incorporates a USB 2.0 host/function module (USB module) compliant to USB (Universal Serial Bus) Specification 2.0. The USB module provides capabilities as a host/function controller which supports full-speed (12 Mbps) and low-speed (1.5 Mbps) transfer.

The USB module can also detect battery charging (hereafter BC) connection during host/function controller operation compliant to USB Battery Charging Specification Revision 1.2.

The USB host/function module can detect connection of USB host/function devices compliant with the 2.1A/1.0A charging mode prescribed in USB power supply component specification.

Table 13 - 1 lists the USB Specifications.

Table 13 - 1 USB module Specifications

Item	Specifications
Features	• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. • The USB host controller and USB function controller are incorporated (can be switched by software). • Operable with self-power mode and bus-power mode.
	Features of the USB host controller - Two ports are provided. - Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported - Automatic scheduling for SOF and packet transmissions - Programmable intervals for interrupt transfers - On-chip D+/D- pin pull-down resistor
	Features of the USB function controller - One port is provided. - Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported - Control transfer stage control function - Device state control function - Auto response function for SET_ADDRESS request - SOF interpolation function - On-chip D+/D- pin pull-up resistor
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer
Pipe configuration	• Buffer memory for USB communications is provided for 448 byte. • Up to five pipes can be selected (including the default control pipe). • Usable pipe numbers are 0, 4 to 7. • Endpoint numbers can be assigned flexibly to PIPE4 to PIPE7.
	Transfer conditions that can be set for each pipe: • PIPE0: Control transfer only (default control pipe: DCP) Buffer size: 64 byte (single buffer) • PIPE4, PIPE5: Bulk transfer only Buffer size: 64 byte (double buffer can be specified) • PIPE6, PIPE7: Interrupt transfer only Buffer size: 64 byte (single buffer)
Others	• Reception ending function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • NAK setting function for response PID generated by end of transfer (SHTNAK) • Automatic buffer memory clearing function after reading data in the pipe specified by the DnFIFO (n = 0, 1) port (DCLRM) • USB Battery Charging is supported (USB Battery Charging Specification Revision 1.2) - Host (charging downstream port, dedicated charging port) BC connection detection function (two ports) is supported - Function (portable device) BC connection detection function (two ports) is supported • Battery charging connection detection optional functions supported - USB host controller supports 2.1A/1.0A charging mode prescribed in USB power supply component specification (2 ports) - USB function controller supports 2.1A/1.0A charging mode prescribed in USB power supply component specification (2 ports) • Optional functions for battery charging connection detection are provided - USB port voltage output function (four patterns) - USB port voltage detection function (16 stages)

13.2 Configuration of USB 2.0 Host/Function Module

The USB module consists of the following hardware.

- USB 2.0 Host/Function Controller
  This controls the host/function supporting full-speed (12 Mbps) and low-speed (1.5 Mbps) transfer.

- USB transceiver
  This is a USB transceiver of one alternate port (USB port 0) for the host/function, and of one dedicated port (USB port 1) for the host.
  This transceiver contains an on-chip pull-down resistor (for the host)/pull-up resistor (for the function) for transfer rate detection.
  An external power supply (via the UV_{DD} pin) or the internal power supply for the USB can be selected as the power supply for the USB transceiver (hereafter USB power supply).

- Buffer memory for USB communications and FIFO/memory controller
  Up to five pipes can be used. End point numbers can be assigned flexibly to PIPE4 to PIPE7 according to peripheral devices and a user system for communication.

- Battery charging detection/controller
  This processes BC connection detection during host /function controller operation compliant to Battery Charging Specification Revision 1.2.

- Various registers
  There are various registers for control, monitoring, and transmitting/receiving data.
  See Table 13 - 3.

- Various pins
  USB port I/O pins (UDP0, UDM0, UDP1 , UDM1)
  VBUS input pin (UV_{BUS})
  USB transceiver power supply pin (UV_{DD})
  VBUS supply control output pins (UVBUSEN0, UVBUSEN1)
  Overcurrent detection input pins (UOVRCUR0, UOVRCUR1)
  See Table 13 - 2 for details.

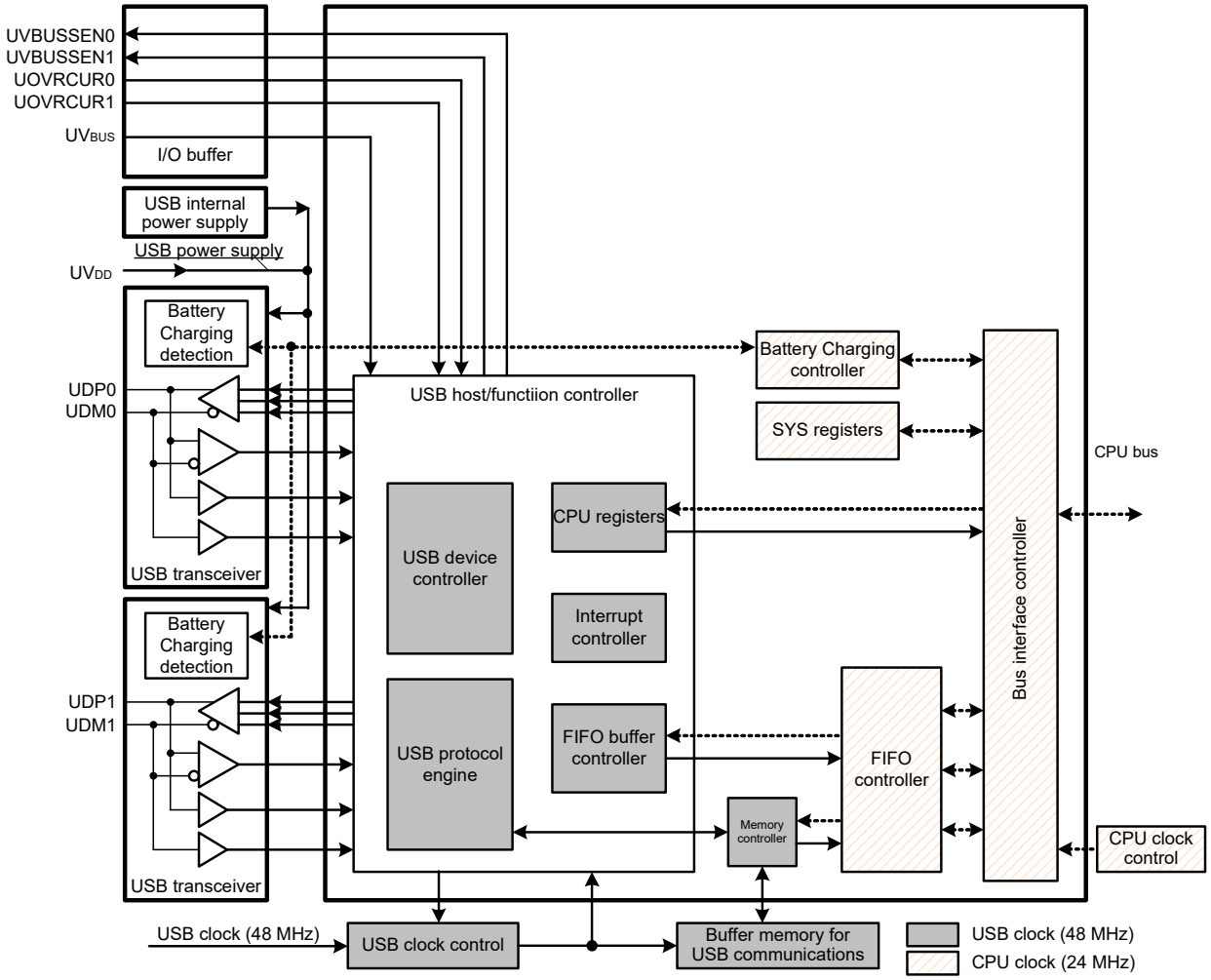
- Others
  The clock controller operates or stops various clocks to be used by the USB module and divides their frequencies. The bus interface controller controls access between the CPU and each register of the USB module.

Table 13 - 2 lists the USB Module I/O Pins and Figure 13 - 1 the USB Module Block Diagram.

Table 13 - 2 USB module I/O Pins

Pin Name	I/O	Function
UDP0	I/O	D+ I/O pin of USB port 0 This pin should be connected to the D+ pin of the USB bus.
UDM0	I/O	D- I/O pin of USB port 0 This pin should be connected to the D- pin of the USB bus.
UV_{BUS}	Input	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected during operation as a function controller.
UDP1	I/O	D+ I/O pin of USB port 1 This pin should be connected to the D+ pin of the USB bus.
UDM1	I/O	D- I/O pin of USB port 1 This pin should be connected to the D- pin of the USB bus.
UVBUSEN0	Output	VBUS (5 V) supply enable signal to the external power supply IC for USB port 0
UVBUSEN1	Output	VBUS (5 V) supply enable signal to the external power supply IC for USB port 1
UOVRCUR0	Input	An external overcurrent detection signal for USB port 0 should be connected to this pin.
UOVRCUR1	Input	An external overcurrent detection signal for USB port 1 should be connected to this pin.
UV_{DD}	I/O	Power supply (USB power supply) pin for the USB transceiver. This pin should be connected to an external power supply. When using the internal power supply for the USB, a capacitor of 0.33 µF should be connected between this pin and V_{SS}.

Figure 13 - 1 Block diagram of USB module



13.3 Registers Used in USB 2.0 Host/Function Module

Table 13 - 3 lists the USB Registers.

Table 13 - 3 List of USB Registers (1/2)

Register	Name Symbol	After Reset	Address	Access Size
System Configuration Control Register	SYSCFG	0000H	F0400H, F0401H	16
System Configuration Control Register 1	SYSCFG1	0000H	F0402H, F0403H	16
System Configuration Status Register 0	SYSSTS0	X00X0000 00X00000B	F0404H, F0405H	16
System Configuration Status Register 1	SYSSTS1	X00X0000 00X00000B	F0406H, F0407H	16
Device State Control Register 0	DVSTCTR0	0000H	F0408H, F0409H	16
Device State Control Register 1	DVSTCTR1	0000H	F040AH, F040BH	16
CFIFO Port Register	CFIFOM	0000H	F0414H, F0415H	8, 16
D0FIFO Port Register	D0FIFOM	0000H	F0418H, F0419H	8, 16
D1FIFO Port Register	D1FIFOM	0000H	F041CH, F041DH	8, 16
CFIFO Port Select Register	CFIFOSEL	0000H	F0420H, F0421H	16
CFIFO Port Control Register	CFIFOCTR	0000H	F0422H, F0423H	16
D0FIFO Port Select Register	D0FIFOSEL	0000H	F0428H, F0429H	16
D0FIFO Port Control Register	D0FIFOCTR	0000H	F042AH, F042BH	16
D1FIFO Port Select Register	D1FIFOSEL	0000H	F042CH, F042DH	16
D1FIFO Port Control Register	D1FIFOCTR	0000H	F042EH, F042FH	16
Interrupt Enable Register 0	INTENB0	0000H	F0430H, F0431H	16
Interrupt Enable Register 1	INTENB1	0000H	F0432H, F0433H	16
Interrupt Enable Register 2	INTENB2	0000H	F0434H, F0435H	16
BRDY Interrupt Enable Register	BRDYENB	0000H	F0436H, F0437H	16
NRDY Interrupt Enable Register	NRDYENB	0000H	F0438H, F0439H	16
BEMP Interrupt Enable Register	BEMPENB	0000H	F043AH, F043BH	16
SOF Output Configuration Register	SOFCFG	0000H	F043CH, F043DH	16
Interrupt Status Register 0	INTSTS0	00000000 X0000000B	F0440H, F0441H	16
Interrupt Status Register 1	INTSTS1	XX0X0000 00000000B	F0442H, F0443H	16
Interrupt Status Register 2	INTSTS2	X00X0000 00000000B	F0444H, F0445H	16
BRDY Interrupt Status Register	BRDYSTS	0000H	F0446H, F0447H	16
NRDY Interrupt Status Register	NRDYSTS	0000H	F0448H, F0449H	16
BEMP Interrupt Status Register	BEMPSTS	0000H	F044AH, F044BH	16
Frame Number Register	FRMNUM	0000H	F044CH, F044DH	16
USB Address Register	USBADDR	0000H	F0450H, F0451H	16
USB Request Type Register	USBREQ	0000H	F0454H, F0455H	16
USB Request Value Register	USBVAL	0000H	F0456H, F0457H	16
USB Request Index Register	USBINDX	0000H	F0458H, F0459H	16
USB Request Length Register	USBLENG	0000H	F045AH, F045BH	16
DCP Configuration Register	DCPCFG	0000H	F045CH, F045DH	16
DCP Maximum Packet Size Register	DCPMAXP	0040H	F045EH, F045FH	16
DCP Control Register	DCPCTR	0040H	F0460H, F0461H	16
Pipe Window Select Register	PIPESEL	0000H	F0464H, F0465H	16
Pipe Configuration Register	PIPECFG	0000H	F0468H, F0469H	16
Pipe Maximum Packet Size Register	PIPEMAXP	0000H/0040H[Note]	F046CH, F046DH	16
Pipe Cycle Control Register	PIPEPERI	0000H	F046EH, F046FH	16

Table 13 - 3 List of USB Registers (2/2)

Register	Name Symbol	After Reset	Address	Access Size
Pipe 4 Control Register	PIPE4CTR	0000H	F0476H, F0477H	16
Pipe 5 Control Register	PIPE5CTR	0000H	F0478H, F0479H	16
Pipe 6 Control Register	PIPE6CTR	0000H	F047AH, F047BH	16
Pipe 7 Control Register	PIPE7CTR	0000H	F047CH, F047DH	16
Pipe 4 Transaction Counter Enable Register	PIPE4TRE	0000H	F049CH, F049DH	16
Pipe 4 Transaction Counter Register	PIPE4TRN	0000H	F049EH, F049FH	16
Pipe 5 Transaction Counter Enable Register	PIPE5TRE	0000H	F04A0H, F04A1H	16
Pipe 5 Transaction Counter Register	PIPE5TRN	0000H	F04A2H, F04A3H	16
BC Control Register 0	USBBCCTRL0	0000H	F04B0H, F04B1H	16
BC Control Register 1	USBBCCTRL1	0000H	F04B4H, F04B5H	16
BC Option Control Register 0	USBBCOPT0	0000H	F04B8H, F04B9H	16
BC Option Control Register 1	USBBCOPT1	0000H	F04BCH, F04BDH	16
USB Clock Selection Register	UCKSEL	0000H	F04C4H, F04C5H	16
USB Module Control Register	USBMC	0002H	F04CCH, F04CDH	16
Device Address 0 Configuration Register	DEVADD0	0000H	F04D0H, F04D1H	16
Device Address 1 Configuration Register	DEVADD1	0000H	F04D2H, F04D3H	16
Device Address 2 Configuration Register	DEVADD2	0000H	F04D4H, F04D5H	16
Device Address 3 Configuration Register	DEVADD3	0000H	F04D6H, F04D7H	16
Device Address 4 Configuration Register	DEVADD4	0000H	F04D8H, F04D9H	16
Device Address 5 Configuration Register	DEVADD5	0000H	F04DAH, F04DBH	16

Note The initial value of this register differs according to the setting of the PIPESEL3 to PIPESEL0 bits. The initial value is 0000H when the pipe is not selected and 0040H when selected.

Table 13 - 4 Registers Initialized by Writing USBE = 0 (When Function Controller Function is Selected)

Register	Symbol	Remarks
SYSSTS0	LNST1, LNST0	The value is retained when the host controller function is selected.
SYSSTS1	LNST1, LNST0	The value is retained when the host controller function is selected.
DVSTCTR0	RHST2 to RHST0	
DVSTCTR1	RHST2 to RHST0	
INTSTS0	DVSQ2 to DVSQ0	The value is retained when the host controller function is selected.
USBADDR	USBADDR	The value is retained when the host controller function is selected.
USBREQ	BREQUEST, BMREQUESTTYPE	The value is retained when the host controller function is selected.
USBVAL	WVALUE	The value is retained when the host controller function is selected.
USBINDX	WINDEX	The value is retained when the host controller function is selected.
USBLENG	WLENGTH	The value is retained when the host controller function is selected.

Table 13 - 5 Registers Initialized by Writing USBE = 0 (When Host Controller Function is Selected)

Register	Symbol	Remarks
DVSTCTR0	RHST2 to RHST0	
DVSTCTR1	RHST2 to RHST0	
FRMNUM	FRNM	The value is retained when the function controller function is selected.

13.3.1 System configuration control register (SYSCFG), system configuration control register 1 (SYSCFG1)

Figure 13 - 2 Format of System Configuration Control Register (SYSCFG)

Address: F0400H, F0401H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSCFG	—	—	—	—	—	SCKE	—	CNEN	—	DCFM	DRPD	DPRPU	DMRPU	—	—	USBE

Bits 15 to 11	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

SCKE	USB module clock enable	R/W
0	Stops supplying the clock signal to the USB module.	R/W
1	Enables supplying the clock signal to the USB module.	
Stops or enables supplying 48-MHz clock signals to the USB module. After setting the SCKE bit to 1, be sure to read this bit to confirm that it is actually set to 1. When the SCKE bit is 0, the USB module registers that can be read and written are SYSCFG, SYSCFG1, USBBCCTRL0, USBBCCTRL1, USBBCOPT0, USBBCOPT1, and USBMC.		

Bit 9	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

CNEN	USB port 0 single end receiver enable	R/W
0	Single end receiver operation is disabled.	R/W
1	Single end receiver operation is enabled.	
Enables or disables the single end receiver. Setting the CNEN bit is set to 1 allows the USB module to enable the single end receiver of USB port 0 and set the LNST bit to monitor the status of the D+/D- lines. The CNEN bit is used to monitor LNST when the USB module operates as a Portable Device for Battery Charging.		

Bit 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DCFM	Controller function select	R/W
0	Function controller function is selected.	R/W
1	Host controller function is selected.	
Selects the function of the USB module. The DCFM bit should be modified while DPRPU is 0 and DRPD is 0.		

DRPD	USB port 0 D+/D- line resistor control	R/W
0	Pulling down the lines is disabled.	R/W
1	Pulling down the lines is enabled.	
Enables or disables pulling down the D+/D- lines of USB port 0 when the host controller function is selected. The DRPD bit should be set to 1 if the host controller function is selected, and should be set to 0 if the function controller function, host BC connection detection function, or function BC connection detection function is selected.		

DPRPU	USB port 0 D+ line resistor control Note	R/W
0	Pulling up the line is disabled.	R/W
1	Pulling up the line is enabled.	
Enables or disables pulling up the D+ line when the function controller function is selected. Setting the DPRPU bit to 1 when the function controller function is selected allows the USB module to enable pulling up the D+ line of USB port 0, thus notifying the USB host of connection as a full-speed device. Modifying the DPRPU bit from 1 to 0 allows the USB module to disable pulling up the D+ line of USB port 0, thus notifying the USB host of disconnection. The DPRPU bit should be set to 1 if the function controller function is selected, and should be set to 0 if the function controller function, host BC connection detection function, or function BC connection detection function is selected.		

DMRPU	USB port 0 D- line resistor control Note	R/W
0	Pulling up the line is disabled.	R/W
1	Pulling up the line is enabled.	
Enables or disables pulling up the D- line when the function controller function is selected. Setting the DMRPU bit to 1 when the function controller function is selected allows the USB module to enable pulling up the D- line of USB port 0, thus notifying the USB host of connection as a low-speed device. Modifying the DMRPU bit from 1 to 0 allows the USB module to disable pulling up the D- line of USB port 0, thus notifying the USB host of disconnection. The DMRPU bit should be set to 1 if the function controller function is selected, and should be set to 0 if the function controller function, host BC connection detection function, or function BC connection detection function is selected.		

Bits 2, 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

USBE	USB module clock enable	R/W
0	USB module operation is disabled.	R/W
1	USB module operation is enabled.	
Enables or disables operation of the USB module. Modifying the USBE bit from 1 to 0 initializes some register bits as listed in Table 13 - 4 and 13 - 5. This bit should be modified while SCKE is 1. When the host controller function is selected, USBE should be set to 1 after setting DRPD to 1, eliminating LNST bit chattering, and checking that the USB bus state has been settled.		

Note Setting the DMRPU and DPRPU bits simultaneously to 1 (to enable pulling up the line) is prohibited.

Figure 13 - 3 Format of System Configuration Control Register 1 (SYSCFG1)

Address: F0402H, F0403H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSCFG1	—	—	—	—	—	—	—	CNEN	—	—	DRPD	—	—	—	—	—

Bit 15	Reserved	R/W
—	The write value must be 0. The read value is 0.	—

Bits 14 to 9	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

CNEN	USB port 1 single end receiver enable	R/W
0	Single end receiver operation is disabled.	R/W
1	Single end receiver operation is enabled.	
Enables or disables the single end receiver. Setting the CNEN bit is set to 1 allows the USB module to enable the single end receiver of USB port 1 and set the LINST bit to monitor the status of the D+/D- lines. The CNEN bit is used to monitor LNST when the USB module operates as a Portable Device for Battery Charging.		

Bits 7, 6	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DRPD		R/W
0	Pulling down the lines is disabled.	R/W
1	Pulling down the lines is enabled.	
Enables or disables pulling down the D+/D- lines of USB port 1 when the host controller function is selected. This bit should be set to 1 if the function controller function is selected, and should be set to 0 if the function controller function, host BC connection detection function, or function BC connection detection function is selected.		

Bits 4 to 0	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

13.3.2 System configuration status register n (SYSSTSn) (n = 0, 1)

Figure 13 - 4 Format of System Configuration Status Register n (SYSSTSn) (n = 0, 1)

Address: F0404H, F0405H (SYSSTS0), F0406H, F0407H (SYSSTS1) After reset: X00X0000 00X00000B

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSSTSn	OVCMON1	—	—	—	—	—	—	—	—	HTACT	—	—	—	—	LNST1	LNST0

OVCMON1	External UOVRCURn input pin monitor Note	R/W
These bits indicate the status of overcurrent from an external power-supply IC. The OVCMON1 bit indicates the status of the UOVRCURn pin.		R

Bit 13	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

Bit 12	Reserved	R/W
—	The write value must be 0. The read value is undefined.	—

Bits 11 to 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

HTACT	USB port n USB host sequencer status monitor	R/W
0	Host sequencer of the USB module is completely stopped.	R
1	Host sequencer of the USB module is not completely stopped.	
This bit is 0 when the host sequencer of the USB module is completely stopped. Make sure the HTACT bit is 0 when stopping the clock supply to the USB module.		

Bit 5	Reserved	R/W
—	The write value must be 0. The read value is undefined.	—

Bits 4 to 2	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

LNST1	LNST0	USB port n USB data line status monitor	R/W
0	0	SE0	R
0	1	J-State (full speed)/K-State (low speed)	
1	0	K-State (full speed)/J-State (low speed)	
1	1	SE1	
These bits indicate the status of the USB data bus lines (D+ and D-). The LNST1 and LNST0 bits should be read after enabling pull-down of the line (DRPD = 1).			

Note The read value depends on the status of the UOVRCURn pin.

13.3.3 Device state control register n (DVSTCTRn) (n = 0, 1)

Figure 13 - 5 Format of Device State Control Register 0 (DVSTCTR0)

Address: F0408H, F0409H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVSTCTR0	—	—	—	—	—	—	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST2	RHST1	RHST0

Bit 15	Reserved	R/W
—	The write value must be 0. The read value is 0.	—

Bits 14 to 12	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

Bits 11, 10	Reserved	R/W
—	The write value must be 0. The read value is 0.	—

VBUSEN	USB port 0 UVBUSEN0 output pin control	R/W
The VBUSEN bit value is output as the status of the external UVBUSEN0 pin without change.		R/W

WKUP	USB port 0 wakeup output	R/W
0	Remote wakeup signal is not output.	R/W Note
1	Remote wakeup signal is output.	
Enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller function is selected. The USB module controls the output time of a remote wakeup signal. When this bit is set to 1, the USB module clears this bit to 0 after outputting the 10-ms K-state. According to the USB Specifications, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. Even if the USB module writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms. Do not write 1 to this bit, unless the device state is in the suspended state (DVSQ2 to DVSQ0 in the INTSTS0 register = 1xxB) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to WKUP while SCKE in the SYSCFG register = 1). This bit should be set to 0 if the host controller function is selected.		

RWUPE	USB port 0 wakeup detection enable	R/W
0	Downstream port wakeup is disabled.	R/W
1	Downstream port wakeup is enabled.	
Enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller function is selected. With this bit set to 1, on detecting the remote wakeup signal, the USB module detects the resume signal (K-state for 2.5 µs) from the downstream port device and performs the resume processing (drives the port to the K-state). With this bit set to 0, the USB module ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port. While the PWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SCKE should be set to 1). This bit should be set to 0 if the function controller function is selected.		

USBRST	USB port 0 USB bus reset output	R/W
0	USB bus reset signal is not output.	R/W
1	USB bus reset signal is output.	
Controls the USB bus reset signal output when the host controller function is selected. When the host controller function is selected, setting this bit to 1 allows the USB module to drive SE0 of the USB port to reset the USB bus. The USB module continues outputting SE0 while USBRST is 1 (until software sets USBRST to 0). The USBRST bit should be 1 (= USB bus reset period) for the time defined by the USB Specifications 2.0. Writing 1 to this bit during communication (UACT = 1) or during the resume processing (RESUME = 1) prevents the USB module from starting the USB bus reset processing until UACT and RESUME become 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to USBRST). This bit should be set to 0 if the function controller function is selected.		

RESUME	USB port 0 resume output	R/W
0	Resume signal is not output.	R/W
1	Resume signal is output.	
Controls the resume signal output when the host controller function is selected. Setting the RESUME bit to 1 allows the USB module to drive the port to the K-state and output the resume signal. The USB module continues outputting K-state while RESUME is 1 (until software sets RESUME to 0). The RESUME bit should be 1 (= resume period) for the time defined by the USB Specifications 2.0. This bit should be set to 1 in the suspended state. Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to RESUME). This bit should be set to 0 if the function controller function is selected.		

UACT	USB port 0 USB bus enable	R/W
0	Downstream port is disabled (SOF transmission is disabled).	R/W
1	Downstream port is enabled (SOF transmission is enabled).	
Enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller function is selected. With this bit set to 1, the USB module puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception. This module starts outputting SOF packets within one frame after software has written 1 to UACT. With this bit set to 0, the USB module enters the idle state after outputting SOF packets. The USB module sets the UACT bit to 0 on any of the following conditions. • A DTCH interrupt is detected during communication (while UACT = 1). • An EOFERR interrupt is detected during communication (while UACT = 1). Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to USBRST) or at the end of the resume processing from the suspended state (writing 0 to RESUME). This bit should be set to 0 if the function controller function is selected.		

Bit 3	Reserved	R/W
—	Nothing is assigned. The write value must be 0. The read value is 0.	—

RHST2	RHST1	RHST0	USB port 0 USB bus reset status	R/W
• When the host controller function is selected				R
0	0	0	Communication speed not determined (powered state or no connection)	
1	X	X	USB bus reset in progress	
0	0	1	Low-speed connection	
0	1	0	Full-speed connection	
• When the function controller function is selected				
0	0	0	Communication speed not determined	
1	X	X	USB bus reset in progress	
0	0	1	Low-speed connection	
0	1	0	Full-speed connection	
The RHST[2:0] bits indicate the status of the USB bus reset. When the host controller function is selected, these bits indicate 100B after software has written 1 to USBRST. The USB module fixes the value of these bits when software writes 0 to USBRST and the USB module completes SE0 driving. When the function controller function is selected, a DVST interrupt is generated as soon as the USB module detects the USB bus reset and then these bits are fixed to 010B.				

Note Only 1 can be written.

Remark x = Don't care

Figure 13 - 6 Format of Device State Control Register 1 (DVSTCTR1)

Address: F040AH, F040BH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVSTCTR1	—	—	—	—	—	—	VBUSEN	—	RWUPE	USBRST	RESUME	UACT	—	RHST2	RHST1	RHST0

Bits 15, 14	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

Bit 13	Reserved	R/W
—	The write value must be 0. The read value is 0.	—

Bits 12 to 10	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

VBUSEN	USB port 1 UVBUSEN1 output pin control	R/W
The VBUSEN bit value is output as the status of the external UVBUSEN1 pin without change.		R/W

Bit 8	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

RWUPE	USB port 1 wakeup detection enable	R/W
0	Downstream port wakeup output is disabled.	R/W
1	Downstream port wakeup is enabled.	
Enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller function is selected. With this bit set to 1, on detecting the remote wakeup signal, the USB module detects the resume signal (K-state for 2.5 µs) from the downstream port device and performs the resume processing (drives the port to the K-state). With this bit set to 0, the USB module ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port. While the PWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SCKE should be set to 1). This bit should be set to 0 if the function controller function is selected.		

USBRST	USB port 1 USB bus reset output	R/W
0	USB bus reset signal is not output.	R/W
1	USB bus reset signal is not output.	
Controls the USB bus reset signal output when the host controller function is selected. When the host controller function is selected, setting this bit to 1 allows the USB module to drive SE0 of the USB port to reset the USB bus. The USB module continues outputting SE0 while USBRST is 1 (until software sets USBRST to 0). The USBRST bit should be 1 (= USB bus reset period) for the time defined by the USB Specifications 2.0. Writing 1 to this bit during communication (UACT = 1) or during the resume processing (RESUME = 1) prevents the USB module from starting the USB bus reset processing until both UACT and RESUME become 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to USBRST). This bit should be set to 0 if the function controller function is selected.		

RESUME	USB port 1 resume output	R/W
0	Resume signal is not output.	R/W
1	Resume signal is output.	
Controls the resume signal output when the host controller function is selected. Setting the RESUME bit to 1 allows the USB module to drive the port to the K-state and output the resume signal. The USB module continues outputting K-state while RESUME is 1 (until software sets RESUME to 0). The RESUME bit should be 1 (= resume period) for the time defined by the USB Specifications 2.0. This bit should be set to 1 in the suspended state. Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to RESUME). This bit should be set to 0 if the function controller function is selected.		

UACT	USB port 1 USB bus enable	R/W
0	Downstream port is disabled (SOF transmission is disabled).	R/W
1	Downstream port is enabled (SOF transmission is enabled).	
Enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller function is selected. With this bit set to 1, the USB module puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception. This module starts outputting SOF packets within one frame after software has written 1 to the UACT bit. With this bit set to 0, the USB module enters the idle state after outputting SOF packets. The USB module sets the UACT bit to 0 on any of the following conditions. A DTCH interrupt is detected during communication (while UACT = 1). An EOFERR interrupt is detected during communication (while UACT = 1). Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to USBRST) or at the end of the resume processing from the suspended state (writing 0 to RESUME). This bit should be set to 0 if the function controller function is selected.		

Bit 3	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

RHST2	RHST1	RHST0	USB port 1 USB bus reset status	R/W
• When the host controller function is selected				R
0	0	0	Communication speed not determined (powered state or no connection)	
1	X	X	USB bus reset in progress	
0	0	1	Low-speed connection	
0	1	0	Full-speed connection	
The RHST[2:0] bits indicate 000B when the function controller function is selected. These bits indicate the status of the USB bus reset when the host controller function is selected. When the host controller function is selected, these bits indicate 100B after software has written 1 to USBRST. The USB module fixes the value of these bits when software writes 0 to USBRST and the USB module completes SE0 driving.				

Remark x = Don't care

13.3.4 CFIFO port register (CFIFOM), DnFIFO port register (DnFIFOM) (n = 0, 1)

Figure 13 - 7 Format of CFIFO Port Register (CFIFOM)

Address: F0414H, F0415H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIFOM	CFIFO[15:0]															

CFIFO[15:0]	CFIFO port	R/W
Accessing the CFIFO bits allow reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer. The CFIFO port register can be accessed only while the FRDY bit in the CFIFO port control register (CFIFOCTR) is 1. The valid bits in the CFIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Table 13 - 6 and 13 - 7.		R/W

Caution 1. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed.

Caution 2. The same pipe should not be assigned to different FIFO ports.

Caution 3. There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

Table 13 - 6 Endian Operation in 16-Bit Access

BIGEND Bit in CFIFO Port Register, DnFIFO Port Register	Bits 15 to 8	Bits 7 to 0
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 13 - 7 Endian Operation in 8-Bit Access

BIGEND Bit in CFIFO Port Register, DnFIFO Port Register	Bits 15 to 8	Bits 7 to 0
0	Access prohibited	N + 0 data
1	Access prohibited	N + 0 data

Figure 13 - 8 Format of DnFIFO Port Register (DnFIFOM) (n = 0, 1)

Address: F0418H, F0419H (D0FIFOM), F041CH, F041DH (D1FIFOM) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DnFIFOM	DnFIFO[15:0]															

DnFIFO[15:0]	FIFO port	R/W
Addresses for CPU transfers using the DnFIFO port. Accessing the DnFIFO bits allow reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer. The DnFIFO port register can be accessed only while the FRDY bit in the DnFIFO port control register (DnFIFOCTR) is 1. The valid bits in the DnFIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Table 13 - 6 and 13 - 7.		R/W

Caution 1. The FIFO buffer for DCP (control transfer) cannot be accessed.

Caution 2. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed.

Caution 3. The same pipe should not be assigned to different FIFO ports.

Caution 4. There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

13.3.5 CFIFO port select register (CFIFOSEL), DnFIFO port select register (DnFIFOSEL) (n = 0, 1)

Figure 13 - 9 Format of CFIFO Port Select Register (CFIFOSEL)

Address: F0420H, F0421H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIFOSEL	RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0

RCNT	Read count mode	R/W
0	When all of the receive data has been read from the CFIFO, 0 is written to the DTLN bit. (In double buffer mode, the DTLN bit value is cleared when all the data has been read from only a single plane.)	R/W
1	The DTLN bit is decremented each time the receive data is read from the CFIFO.	
Specifies the read mode for the value in the DTLN[8:0] bits in the CFIFOCTR register.		

REW	Buffer pointer rewind	R/W
0	Disable (the buffer pointer is not rewound.)	R/W Note 1
1	The buffer pointer is rewound.	
Specifies whether or not to rewind the buffer pointer. When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed). Do not set the REW bit to 1 simultaneously with modifying the CURPIPE3 to CURPIPE0 bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1. To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.		

Bits 13 to 11	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

MBW	CFIFO port access bit width	R/W
0	8-bit width	R/W
1	16-bit width	
Specifies the bit width for accessing the CFIFO port. When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read. When the selected pipe is in the receiving direction, set the CURPIPE3 to CURPIPE0 and MBW bits simultaneously. If the size of data to be read is an odd number of bytes when 16-bit width is selected, delete unnecessary bytes after reading the data in words. When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory. An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.		

Bit 9	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

BIGEND	CFIFO port endian control	R/W
0	Little endian	R/W
1	Big endian	
Specifies the byte endian for the CFIFO port.		

Bits 7, 6	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

ISEL	CFIFO port access direction when DCP is selected	R/W
0	Reading from the buffer memory is selected	R/W
1	Writing to the buffer memory is selected	
After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. Set this bit and the CURPIPE3 to CURPIPE0 bits simultaneously.		

Bit 4	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0	CFIFO port access pipe specification Note 2	R/W
0	0	0	0	DCP (Default control pipe)	R/W
0	1	0	0	Pipe 4	
0	1	0	1	Pipe 5	
0	1	1	0	Pipe 6	
0	1	1	1	Pipe 7	
Other than above				Do not set.	
The CURPIPE3 to CURPIPE0 bits specify the pipe number using which data is read or written through the CFIFO port. After writing to the CURPIPE3 to CURPIPE0 bits, read these bits to check that the written value agrees with the read value before proceeding to the next process. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.					

Note 1. Only 0 can be read.

Note 2. The same pipe number should not be set by the CURPIPE3 to CURPIPE0 bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

Figure 13 - 10 Format of DnFIFO port select register (DnFIFOSEL) (n = 0, 1)

Address: F0428H, F0429H (D0FIFOSEL), F042CH, F042DH (D1FIFOSEL) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DnFIFOSEL	RCNT	REW	DCLRM	—	—	MBW	—	BIGEND	—	—	—	—	CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0

RCNT	Read count mode	R/W
0	When all of the receive data has been read from the DnFIFO, 0 is written to the DTLN[8:0] bits. (In double buffer mode, the DTLN[8:0] bit value is cleared when all the data has been read from only a single plane.)	R/W
1	The DTLN[8:0] bits are decremented each time the receive data is read from the DnFIFO.	
Specifies the read mode for the value in the DTLN[8:0] bits in the DnFIFOCTR register. When accessing DnFIFO with the BFRE bit set to 1, set the RCNT bit to 0.		

REW	Buffer pointer rewind	R/W
0	Disable (the buffer pointer is not rewound.)	R/W Note 1
1	The buffer pointer is rewound.	
Specifies whether or not to rewind the buffer pointer. When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed). Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, be sure to check that FRDY is 1. When accessing the DnFIFO with the BFRE bit set to 1, do not set the REW bit to 1 while reading of the short packet data is completed. To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.		

DCLRM	Auto buffer memory clear mode accessed after specified pipe data is read	R/W
0	Auto buffer clear mode is disabled.	R/W
1	Auto buffer clear mode is enabled.	
Enables or disables the buffer memory to be cleared automatically after data has been read out using the selected pipe. With the DCLRM bit set to 1, the USB module sets BCLR to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while BFRE in the PIPECFG register is 1. When using the USB module with BRDYM in the SOFCFG register set to 1, set the DCLRM bit to 0.		

Bit 12	Reserved	R/W
—	The write value must be 0. The read value is 0.	—

Bit 11	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

MBW	DnFIFO port access bit width	R/W
0	8-bit width	R/W
1	16-bit width	
Specifies the bit width for accessing the DnFIFO port. When the selected pipe is in the receiving direction, once reading data is started after setting the MBW bit, this bit should not be modified until all the data has been read. When the selected pipe is in the receiving direction, set the CURPIPE and MBW bits simultaneously. If the size of data to be read is an odd number of bytes when 16-bit width is selected, delete unnecessary bytes after reading the data in words. When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory. An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.		

Bit 9	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

BIGEND	DnFIFO port endian control	R/W
0	Little endian	R/W
1	Big endian	
Specifies the byte endian for the DnFIFO port.		

Bit 7	Reserved	R/W
—	The write value must be 0. The read value is 0.	—

Bits 6 to 4	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0	DnFIFO port access pipe specification Note 2	R/W
0	0	0	0	No pipe specified	R/W
0	1	0	0	Pipe 4	
0	1	0	1	Pipe 5	
0	1	1	0	Pipe 6	
0	1	1	1	Pipe 7	
Other than above				Do not set.	
The CURPIPE3 to CURPIPE0 bits specify the pipe number using which data is read or written through the DnFIFO port. After writing to CURPIPE3 to CURPIPE0 bits, then read these bits to check that the written value agrees with the read value before proceeding to the next process. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.					

Note 1. Only 0 can be read.

Note 2. The same pipe number should not be set by the CURPIPE3 to CURPIPE0 bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

13.3.6 CFIFO port control register (CFIFOCTR), DnFIFO port control register (DnFIFOCTR) (n = 0, 1)

Figure 13 - 11 Format of CFIFO Port Control Register (CFIFOCTR)

Address: F0422H, F0423H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIFOCTR	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								

BVAL	Buffer memory valid flag	R/W
0	Invalid	R/W Note 1
1	Writing ended	
This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE3 to CURPIPE0 bits (selected pipe). When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB module switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission. • To transmit a short packet, set the BVAL bit to 1 after data has been written. • To transmit a zero-length packet, set the BVAL bit to 1 while the FIFO buffer is empty. When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB module sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission. Writing 1 to the BVAL bit should be done while FRDY is 1 (set by the USB module). When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.		

BCLR	CPU buffer clear	R/W
0	Invalid	R/W Note 2
1	Clears the buffer memory on the CPU side.	
This bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe. When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB module clears only one plane of the FIFO buffer even when both planes are read-enabled. When the selected pipe is the DCP, setting BCLR to 1 allows the USB module to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the PID bits for the DCP control register to NAK before setting BCLR to 1. When the selected pipe is in the transmitting direction, if 1 is written to the BVAL and BCLR bits simultaneously, the USB module clears the data that has been written before it, enabling transmission of a zero-length packet. When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while FRDY in the FIFO port control register is 1 (set by the USB module).		

FRDY	FIFO port ready	R/W
0	FIFO port access is disabled.	R
1	FIFO port access is enabled.	
Indicates whether the FIFO port can be accessed by the CPU. In the following cases, the USB module sets FRDY to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1 to clear the FIFO buffer, and enable transmission and reception of the next data. • A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty. • A short packet is received and the data is completely read while BFRE in the PIPECFG register is 1.		

Bits 12 to 9	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DTLN[8:0]	Receive data length	R/W
The DTLN[8:0] bits indicate the length of the receive data. While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the RCNT bit value as described below. • RCNT = 0 The USB module sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU has read all the received data from a single FIFO buffer plane. While BFRE in the PIPECFG register is 1, these bits retain the length of the receive data until BCLR is set to 1 even after all the data has been read. • RCNT = 1 The USB module decrements the value indicated by these bits each time data is read from the FIFO buffer. (The value is decremented by one when MBW is 0, and by two when MBW is 1.) The USB module sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB module sets the DTLN[8:0] bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.		R

Note 1. Only 1 can be written.

Note 2. Only 0 can be read and 1 can be written.

Figure 13 - 12 Format of DnFIFO Port Control Register (DnFIFOCTR) (n = 0, 1)

Address: F042AH, F042BH (D0FIFOCTR), F042EH, F042FH (D1FIFOCTR) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DnFIFOCTR	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								

BVAL	Buffer memory valid flag	R/W
0	Invalid	R/W Note 1
1	Writing ended	

This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE3 to CURPIPE0 bits (selected pipe).
When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB module switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
• To transmit a short packet, set the BVAL bit to 1 after data has been written.
• To transmit a zero-length packet, set the BVAL bit to 1 while the FIFO buffer is empty.
When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB module sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
Writing 1 to the BVAL bit should be done while FRDY is 1 (set by the USB module).
When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.

BCLR	CPU buffer clear	R/W
0	Invalid	R/W Note 2
1	Clears the buffer memory on the CPU side.	

This bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.
When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB module clears only one plane of the FIFO buffer even when both planes are read-enabled.
When the selected pipe is in the transmitting direction, if 1 is written to the BVAL and BCLR bits simultaneously, the USB module clears the data that has been written before it, enabling transmission of a zero-length packet.
When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while FRDY is 1 (set by the USB module).

FRDY	FIFO port ready	R/W
0	FIFO port access is disabled.	R
1	FIFO port access is enabled.	

Indicates whether the FIFO port can be accessed by the CPU.
In the following cases, the USB module sets FRDY to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.
• A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
• A short packet is received and the data is completely read when BFRE is set to 1.

Bits 12 to 9	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DTLN[8:0]	Receive data length	R/W
The DTLN[8:0] bits indicate the length of the receive data. While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the RCNT bit value as described below. • RCNT = 0 The USB module sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU has read all the received data from a single FIFO buffer plane. While the BFRE bit in the PIPECFG register is 1, these bits retain the length of the receive data until BCLR is set to 1 even after all the data has been read. • RCNT = 1 The USB module decrements the value indicated by these bits each time data is read from the FIFO buffer. (The value is decremented by one when MBW is 0, and by two when MBW is 1.) The USB module sets DTLN to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB module sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.		R/W

Note 1. Only 1 can be written.

Note 2. Only 0 can be read and 1 can be written.

13.3.7 Interrupt enable register 0 (INTENB0)

Figure 13 - 13 Format of Interrupt Enable Register 0 (INTENB0)

Address: F0430H, F0431H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENB0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—

VBSE	VBUS interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the VBINT interrupt is detected.		

RSME	Resume interrupt enable Note	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the RESM interrupt is detected.		

SOFE	Frame number update interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the SOFR interrupt is detected.		

DVSE	Device state transition interrupt enable Note	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the DVST interrupt is detected.		

CTRE	Control transfer stage transition interrupt enable Note	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the CTRT interrupt is detected.		

BEMPE	Buffer empty interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the BEMP interrupt is detected.		

NRDYE	Buffer Not Ready Response Interrupt Enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the NRDY interrupt is detected.		

BRDYE	Buffer Ready Interrupt Enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the BRDY interrupt is detected.		

Bits 7 to 0	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

Note The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller function is selected; do not set these bits to 1 (interrupt output enabled) when the host controller function is selected.

13.3.8 Interrupt enable register n (INTENBn) (n = 1, 2)

Figure 13 - 14 Format of Interrupt Enable Register 1 (INTENB1)

Address: F0432H, F0433H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENB1	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	PDDETINTE

OVRCRE	USB port 0 overcurrent input change interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the OVRCR interrupt is detected.		

BCHGE	USB port 0 USB bus change interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the BCHG interrupt is detected.		

Bit 13	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DTCHE	USB port 0 disconnection detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the DTCH interrupt is detected.		

ATTCHE	USB port 0 connection detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the ATTCH interrupt is detected.		

Bits 10 to 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

EOFERRE	USB port 0 EOF error detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the EOFERR interrupt is detected.		

SIGNE	Setup transaction error interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the SIGN interrupt is detected.		

SACKE	Setup transaction normal response interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the SACK interrupt is detected.		

Bits 3 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PDDETINTE	USB port 0 Portable Device detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the PDDETINT interrupt is detected.		

Caution The bits in INTENB1 can be set to 1 only when the host controller function is selected; do not set these bits to 1 (interrupt output enabled) when the function controller function is selected.

Figure 13 - 15 Format of Interrupt Enable Register 2 (INTENB2)

Address: F0434H, F0435H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENB2	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	—	—	—	—	—	PDDETINTE

OVRCRE	USB port 1 Overcurrent input change interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the OVRCR interrupt is detected.		

BCHGE	USB port 1 USB bus change interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the BCHG interrupt is detected.		

Bit 13	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DTCHE	USB port 1 disconnection detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the DTCH interrupt is detected.		

ATTCHE	USB port 1 connection detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the ATTCH interrupt is detected.		

Bits 10 to 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

EOFERRE	USB port 1 EOF error detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the EOFERR interrupt is detected.		

Bits 5 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PDDETINTE	USB port 1 Portable Device detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the PDDETINT interrupt is detected.		

Caution The bits in INTENB2 can be set to 1 only when the host controller function is selected; do not set these bits to 1 (interrupt output enabled) when the function controller function is selected.

13.3.9 BRDY interrupt enable register (BRDYENB)

Figure 13 - 16 Format of BRDY Interrupt Enable Register (BRDYENB)

Address: F0436H, F0437H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRDYENB	—	—	—	—	—	—	—	—	PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	—	—	—	PIPE0BRDYE

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PIPEnBRDYE	BRDY interrupt enable for PIPEn	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	

Remark n = 7 to 4, 0

13.3.10 NRDY interrupt enable register (NRDYENB)

Figure 13 - 17 Format of NRDY Interrupt Enable Register (NRDYENB)

Address: F0438H, F0439H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDYENB	—	—	—	—	—	—	—	—	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	—	—	—	PIPE0NRDYE

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PIPEnNRDYE	NRDY interrupt enable for PIPEn	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	

Remark n = 7 to 4, 0

13.3.11 BEMP interrupt enable register (BEMPENB)

Figure 13 - 18 Format of BEMP Interrupt Enable Register (BEMPENB)

Address: F043AH, F043BH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEMPENB	—	—	—	—	—	—	—	—	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	—	—	—	PIPE0BEMPE

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PIPEnBEMPE	BEMP interrupt enable for PIPEn	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	

Remark n = 7 to 4, 0

13.3.12 SOF output configuration register (SOFCFG)

Figure 13 - 19 Format of SOF Output Configuration Register (SOFCFG)

Address: F043CH, F043DH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOFCFG	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—

Bits 15 to 9	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

TRNENSEL	Transaction-enabled time select	R/W
0	For non-low-speed communication	R/W
1	For low-speed communication	
This bit selects, for full-speed or low-speed communication, the transaction-enabled time in which the USB module issues tokens in a frame via the port. The TRNENSEL bit is valid only when the host controller function is selected. This bit should be set to 0 if the function controller function is selected.		

Bit 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

BRDYM	BRDY interrupt status clear timing for each pipe	R/W
0	Software clears the status.	R/W
1	The USB module clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	
Specifies the timing for clearing the BRDY interrupt status for each pipe.		

Bit 5	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

EDGESTS	Edge interrupt output status monitor Note	R/W
0	The edge interrupt output signal is not in the middle of the edge processing.	R
1	The edge interrupt output signal is in the middle of the edge processing.	
Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.		

Bits 3 to 0	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

Note Make sure the EDGESTS bit is 0 when stopping the clock supply to the USB module.

13.3.13 Interrupt status register 0 (INTSTS0)

Figure 13 - 20 Format of Interrupt Status Register 0 (INTSTS0)

Address: F0440H, F0441H After reset: 00000000 X0000000B

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTSTS0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ2	DVSQ1	DVSQ0	VALID	CTSQ2	CTSQ1	CTSQ0

VBINT	VBUS interrupt status Note 1	R/W
0	VBUS interrupts are not generated.	R/W Note 2
1	VBUS interrupts are generated.	
The USB module sets the VBINT bit to 1 on detecting a level change (high to low or low to high) in the UV_{BUS} pin input value. The USB module sets the VBSTS bit to indicate the VBUS pin input value. When the UV_{BUS} interrupt is generated, use software to repeat reading the VBSTS bit until the same value is read three or more times, and eliminate chattering.		

RESM	Resume interrupt status Note 1, 3	R/W
0	Resume interrupts are not generated.	R/W Note 2
1	Resume interrupts are generated.	
When the function controller function is selected, the USB module sets the RESM bit to 1 on detecting the falling edge of the signal on the USB_DP pin in the suspended state (DVSQ2 to DVSQ2 = 1xxB). When the host controller function is selected, the read value is invalid.		

SOFR	Frame number refresh interrupt status	R/W
0	SOF interrupts are not generated.	R/W Note 2
1	SOF interrupts are generated.	
(1) When the host controller function is selected The USB module sets the SOFR bit to 1 on updating the frame number when software has set the UACT bit in DVSTCTR0 to 1. (A frame number refresh interrupt is detected every 1 ms.) (2) When the function controller function is selected The USB module sets the SOFR bit to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.) The USB module can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.		

DVST	Device state transition interrupt status Note 3	R/W
0	Device state transition interrupts are not generated.	R/W Note 2
1	Device state transition interrupts are generated.	
When the function controller function is selected, the USB module updates the DVSQ2 to DVSQ2 value and sets the DVST bit to 1 on detecting a change in the device state. When a device state transition interrupt is generated, clear the status before the USB module detects the next device state transition. When the host controller function is selected, the read value is invalid.		

CTRT	Control transfer stage transition interrupt status Note 3	R/W
0	Control transfer stage transition interrupts are not generated	R/W Note 2
1	Control transfer stage transition interrupts are generated.	
When the function controller function is selected, the USB module updates the CTSQ2 to CTSQ0 value and sets the CTRT bit to 1 on detecting a change in the control transfer stage. When a control transfer stage transition interrupt is generated, clear the status before the USB module detects the next control transfer stage transition. When the host controller function is selected, the read value is invalid.		

BEMP	Buffer empty interrupt status	R/W
0	BEMP interrupts are not generated.	R
1	BEMP interrupts are generated.	
Indicates the BEMP interrupt status. The USB module sets the BEMP bit to 1 when at least one PIPEnBEMP bit in the BEMPSTS register is set to 1 among the PIPEnBEMP bits corresponding to the PIPEnBEMPE bits in the BEMPENB register to which 1 has been set (when the USB module detects the BEMP interrupt status in at least one pipe among the pipes for which software enables the BEMP interrupt output). For the conditions for PIPEnBEMP status assertion, refer to **13.4.3.3 BEMP interrupt**. The USB module clears the BEMP bit to 0 when software writes 0 to all the PIPEnBEMP bits corresponding to the PIPEnBEMPE bits to which 1 has been set. The BEMP bit cannot be cleared to 0 even if software writes 0 to this bit.		

NRDY	Buffer not ready interrupt status	R/W
0	NRDY interrupts are not generated.	R
1	NRDY interrupts are generated.	
Indicates the NRDY interrupt status. The USB module sets the NRDY bit to 1 when at least one PIPEnNRDY bit in NRDYSTS is set to 1 among the PIPEnNRDY bits corresponding to the PIPEnNRDYE bits in NRDYENB to which 1 has been set (when the USB module detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output). For the conditions for PIPEnNRDY status assertion, refer to **13.4.3.2 NRDY interrupt**. The USB module clears the NRDY bit to 0 when software writes 0 to all the PIPEnNRDY bits corresponding to the PIPEnNRDYE bits to which 1 has been set. The NRDY bit cannot be cleared to 0 even if software writes 0 to this bit.		

BRDY	Buffer ready interrupt status	R/W
0	BRDY interrupts are not generated.	R
1	BRDY interrupts are generated.	
Indicates the BRDY interrupt status. The USB module sets the BRDY bit to 1 when at least one PIPEnBRDY bit in the BRDYSTS register is set to 1 among the PIPEnBRDY bits corresponding to the PIPEnBRDYE bits in the BRDYENB register to which 1 has been set (when the USB module detects the BRDY interrupt status in at least one pipe among the pipes for which software enables the BRDY interrupt output). For the conditions for PIPEnBRDY status assertion, refer to **13.4.3.1 BRDY interrupt**. The USB module clears the BRDY bit to 0 when software writes 0 to all the PIPEnBRDY bits corresponding to the PIPEnBRDYE bits to which 1 has been set. The BRDY bit cannot be cleared to 0 even if software writes 0 to this bit.		

VBSTS	VBUS interrupt status Note 4	R/W
0	UVBUS pin is low.	R
1	UVBUS pin is high.	

DVSQ2	DVSQ1	DVSQ1	Device state	R/W
0	0	0	Powered state	R
0	0	1	Default state	
0	1	0	Address state	
0	1	1	Configured state	
1	X	X	Suspended state	
These bits indicate the device status. When the host controller function is selected, the read value is invalid.				

VALID	USB request reception	R/W
0	Not detected	R/W Note 2
1	Setup packet reception	
Indicates the USB request reception status. When the host controller function is selected, the read value is invalid.		

CTSQ2	CTSQ1	CTSQ0	Control transfer stage	R/W
0	0	0	Idle or setup stage	R
0	0	1	Control read data stage	
0	1	0	Control read status stage	
0	1	1	Control write data stage	
1	0	0	Control write status stage	
1	0	1	Control write (no data) status stage	
1	1	0	Control transfer sequence error	
1	1	1	Do not set.	
These bits indicate the control transfer stage status. When the host controller function is selected, the read value is invalid.				

Note 1. A change in the status indicated by the VBINT and RESM bits can be detected even while the clock supply is stopped (SCKE = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

Note 2. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bit, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Note 3. A change in the status of the RESM, DVST, and CTRT bits occur only when the function controller function is selected; set the corresponding interrupt enable bits to 0 (disabled) when the host controller function is selected.

Note 4. The value after reset depends on the value of the UVBUS pin. This bit is 1 when the UVBUS pin input is high level and 0 when the input is low level.

13.3.14 Interrupt status register n (INTSTSn) (n = 1, 2)

Figure 13 - 21 Format of Interrupt Status Register 1 (INTSTS1)

Address: F0442H, F0443H After reset: XX0X0000 00000000B

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTSTS1	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	PDDETINT

OVRCR	USB port 0 overcurrent input change interrupt status Note 1	R/W
0	OVRCR interrupts are not generated.	R/W Note 2
1	OVRCR interrupts are generated.	
Indicates the status of the UOVRCUR0 input pin change interrupt. The USB module detects the OVRCR interrupt when a change (high to low or low to high) occurs in at least one of the input values to the UOVRCUR0 pin, and sets the OVRCR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt.		

BCHG	USB port 0 USB bus change interrupt status Note 1	R/W
0	BCHG interrupts are not generated.	R/W Note 2
1	BCHG interrupts are generated.	
Indicates the status of the USB bus change interrupt. The USB module detects the BCHG interrupt when a change in the full-speed/low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. The USB module sets the LNST bits in the SYSSTS0 register to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST bits until the same value is read three or more times, and eliminate chattering. A change in the USB bus state can be detected even while the internal clock supply is stopped. When the function controller function is selected, the read value is invalid.		

Bit 13	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DTCH	USB port 0 USB disconnection detection interrupt status	R/W
0	DTCH interrupts are not generated.	R/W Note 2
1	DTCH interrupts are generated.	
Indicates the status of the USB disconnection detection interrupt when the host controller function is selected. The USB module detects the disconnection detection interrupt on detecting USB bus disconnection, and sets the DTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. The USB module detects bus disconnection based on the USB Specification 2.0. After detecting the disconnection detection interrupt, the USB module controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried out for the USB port and make a transition to the wait state for bus connection to the USB port (wait state for connection detection interrupt generation). • Modifies the UACT bit for the port in which a disconnection detection interrupt has been detected to 0. • Puts the port in which a disconnection detection interrupt has been generated into the idle state. When the function controller function is selected, the read value is invalid.		

ATTCH	USB port 0 connection detection interrupt status	R/W
0	Connection detection interrupts are not generated.	R/W Note 2
1	Connection detection interrupts are generated.	
Indicates the status of the ATTCH interrupt when the host controller function is selected. The USB module detects the ATTCH interrupt on detecting J-state or K-state of the full-speed signal level for 2.5 µs, and sets the ATTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. Specifically, the USB module detects the connection detection interrupt on any of the following conditions. • K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs. • J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs. When the function controller function is selected, the read value is invalid.		

Bits 10 to 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

EOFERR	USB port 0 EOF error detection interrupt status	R/W
0	EOFERR interrupts are not generated.	R/W Note 2
1	EOFERR interrupts are generated.	
Indicates the status of the EOFERR interrupt when the host controller function is selected. The USB module detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by the USB Specifications 2.0, and sets the EOFERR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the EOFERR interrupt. After detecting the EOFERR interrupt, the USB module controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried for the USB port and perform re-enumeration of the USB port. • Modifies the UACT bit for the port in which an EOFERR interrupt has been detected to 0. • Puts the port in which an EOFERR interrupt has been generated into the idle state. When the function controller function is selected, the read value is invalid.		

SIGN	Setup transaction error interrupt status	R/W
0	SIGN interrupts are not generated.	R/W Note 2
1	SIGN interrupts are generated.	
Indicates the status of the setup transaction error interrupt when the host controller function is selected. The USB module detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets the SIGN bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the SIGN interrupt. Specifically, the USB module detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions. • Timeout is detected by the USB module when the peripheral device has returned no response. • A damaged ACK packet is received. • A handshake other than ACK (NAK, NYET, or STALL) is received. When the function controller function is selected, the read value is invalid.		

SACK	Setup transaction normal response interrupt status	R/W
0	SACK interrupts are not generated.	R/W Note 2
1	SACK interrupts are generated.	
Indicates the status of the setup transaction normal response interrupt when the host controller function is selected. The USB module detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB module, and sets the SACK bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the SACK interrupt. When the function controller function is selected, the read value is invalid.		

Bits 3 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PDDETINT	USB port 0 Portable Device detection interrupt status	R/W
0	PDDETINT interrupts are not generated.	R/W Note 2
1	PDDETINT interrupts are generated.	
Indicates the status of the Portable Device detection interrupt when the host controller function is selected. The USB module detects when a change (high to low or low to high) occurs in the input value to the USB transceiver VDPDET pin, and sets this bit to 1. The USB module indicates the input value to the USB transceiver VDPDET pin to the PDDETSTS bit. When the PDDETINT interrupt is generated, use software to repeat reading the PDDETSTS bit until the same value is read three or more times, and eliminate chattering.		

Note 1. A change in the status indicated by the OVRCR or BCHG bit can be detected even while the clock supply is stopped (SCKE = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after enabling the clock supply. No interrupts other than those indicated by the BCHG and OVRCR bits can be detected while the clock supply is stopped (SCKE = 0).

Note 2. To clear the status indicated by the bits in the INTSTS1 register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Figure 13 - 22 Format of Interrupt Status Register 2 (INTSTS2)

Address: F0444H, F0445H After reset: X00X0000 00000000B

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTSTS2	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	—	—	—	—	—	PDDETINT

OVRCR	USB port 1 overcurrent input change interrupt status Note 1	R/W
0	OVRCR interrupts are not generated.	R/W Note 2
1	OVRCR interrupts are generated.	
Indicates the status of the UOVRCUR1 input pin change interrupt. The USB module detects the overcurrent interrupt when a change (high to low or low to high) occurs in at least one of the input values to the UOVRCUR1 pin, and sets the OVRCR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt.		

BCHG	USB port 1 USB bus change interrupt status Note 1	R/W
0	BCHG interrupts are not generated.	R/W Note 2
1	BCHG interrupts are generated.	
Indicates the status of the USB bus change interrupt. The USB module detects the BCHG interrupt when a change in the full-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. The USB module sets the LNST bits in SYSSTS1 to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST bits until the same value is read three or more times, and eliminate chattering. A change in the USB bus state can be detected even while the internal clock supply is stopped. When the function controller function is selected, the read value is invalid.		

Bit 13	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DTCH	USB port 1 disconnection detection interrupt status	R/W
0	DTCH interrupts are not generated.	R/W Note 2
1	DTCH interrupts are generated.	
Indicates the status of the USB disconnection detection interrupt when the host controller function is selected. The USB module detects the disconnection detection interrupt on detecting USB bus disconnection, and sets the DTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. The USB module detects bus disconnection based on the USB Specifications 2.0. After detecting the disconnection detection interrupt, the USB module controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried out for the USB port and make a transition to the wait state for bus connection to the USB port (wait state for connection detection interrupt generation). • Modifies the UACT bit for the port in which a disconnection detection interrupt has been detected to 0. • Puts the port in which a disconnection detection interrupt has been generated into the idle state. When the function controller function is selected, the read value is invalid.		

ATTCH	USB port 1 connection detection interrupt status	R/W
0	Connection detection interrupts are not generated.	R/W Note 2
1	Connection detection interrupts are generated.	
Indicates the status of the ATTCH interrupt when the host controller function is selected. The USB module detects the ATTCH interrupt on detecting J-state or K-state of the full-speed signal level for 2.5 µs, and sets the ATTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. Specifically, the USB module detects the connection detection interrupt on any of the following conditions. K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs. J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs. When the function controller function is selected, the read value is invalid.		

Bits 10 to 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

EOFERR	USB port 1 EOF error detection interrupt status	R/W
0	EOFERR interrupts are not generated.	R/W Note 2
1	EOFERR interrupts are generated.	
Indicates the status of the EOFERR interrupt when the host controller function is selected. The USB module detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by the USB Specifications 2.0, and sets the EOFERR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the EOFERR interrupt. After detecting the EOFERR interrupt, the USB module controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried for the USB port and perform re-enumeration of the USB port. • Modifies the UACT bit for the port in which an EOFERR interrupt has been detected to 0. • Puts the port in which an EOFERR interrupt has been generated into the idle state. When the function controller function is selected, the read value is invalid.		

Bits 5 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PDDETINT	USB port 1 portable Device detection interrupt status	R/W
0	PDDETINT interrupts are not generated.	R/W Note 2
1	PDDETINT interrupts are generated.	
Indicates the status of the Portable Device detection interrupt when the host controller function is selected. The USB module detects when a change (high to low or low to high) occurs in the input value to the USB transceiver VDPDET pin, and sets this bit to 1. The USB module indicates the input value to the USB transceiver VDPDET pin to the PDDETSTS bit. When the PDDETINT interrupt is generated, use software to repeat reading the PDDETSTS bit until the same value is read three or more times, and eliminate chattering.		

Note 1. A change in the status indicated by the OVRCR or BCHG bit can be detected even while the clock supply is stopped (SCKE = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after enabling the clock supply. No interrupts other than those indicated by the BCHG and OVRCR bits can be detected while the clock supply is stopped (SCKE = 0).

Note 2. To clear the status indicated by the bits in the INTSTS2 register, write 0 only to the bits to be cleared; write 1 to the other bits.

13.3.15 BRDY interrupt status register (BRDYSTS)

Figure 13 - 23 Format of BRDY Interrupt Status Register (BRDYSTS)

Address: F0446H, F0447H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRDYSTS	—	—	—	—	—	—	—	—	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	—	—	—	PIPE0 BRDY

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PIPEnBRDY	BRDY interrupt status for PIPEn Note 1	R/W
0	Interrupts are not generated.	R/W Note 2
1	Interrupts are generated.	

Note 1. When BRDYM in SOFCGFG is 0, clearing BRDY interrupts should be done before accessing the FIFO.

Note 2. When BRDYM in SOFCGFG is 0, to clear the status indicated by the bits in BRDYSTS, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Remark n = 7 to 4, 0

13.3.16 NRDY interrupt status register (NRDYSTS)

Figure 13 - 24 Format of NRDY Interrupt Status Register (NRDYSTS)

Address: F0448H, F0449H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDYSTS	—	—	—	—	—	—	—	—	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	—	—	—	PIPE0 NRDY

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PIPEnNRDY	NRDY interrupt status for PIPEn	R/W
0	Interrupts are not generated.	R/W Note
1	Interrupts are generated.	

Note To clear the status indicated by the bits in the NRDYSTS register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Remark n = 7 to 4, 0

13.3.17 BEMP interrupt status register (BEMPSTS)

Figure 13 - 25 Format of BEMP Interrupt Status Register (BEMPSTS)

Address: F044AH, F044BH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEMPSTS	—	—	—	—	—	—	—	—	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	—	—	—	PIPE0 BEMP

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PIPEnBEMP	BEMP interrupt status for PIPEn	R/W
0	Interrupts are not generated.	R/W Note
1	Interrupts are generated.	

Note To clear the status indicated by the bits in the BEMPSTS register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Remark n = 7 to 4, 0

13.3.18 Frame number register (FRMNUM)

Figure 13 - 26 Format of Frame Number Register (FRMNUM)

Address: F044CH, F044DH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRMNUM	—	—	—	—	—	FRNM[10:0]										

Bits 15 to 11	Reserved	R/W
—	The write value must be 0. The read value is 0.	—

FRNM[10:0]	Frame number	R/W
The USB module sets the FRNM[10:0] bits to indicate the latest frame number, which is updated every time an SOF packet is issued or received (every 1 ms). Repeat reading the FRNM[10:0] bits until the same value is read twice.		R

13.3.19 USB address register (USBADDR)

Figure 13 - 27 Format of USB Address Register (USBADDR)

Address: F0450H, F0451H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBADDR	—	—	—	—	—	—	—	—	—	USBADDR[6:0]						

Bits 15 to 12	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

Bits 11 to 8	Reserved	R/W
—	The write value must be 0. The read value is 0.	—

Bit 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

USBADDR[6:0]	USB address	R/W
The current USB address value can be read. This register is not used while in the host mode.		R

13.3.20 USB request type register (USBREQ)

Figure 13 - 28 Format of USB Request Type Register (USBREQ)

Address: F0454H, F0455H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBREQ	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							

BREQUEST[7:0]	Request	R/W
These bits store the USB request bRequest value. [When the host controller function is selected] The USB request data value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1. [When the function controller function is selected] These bits indicate the USB request data value received during the setup transaction. Writing to these bits is invalid.		R/W Note

BMREQUESTTYPE[7:0]	Request type	R/W
These bits store the USB request bmRequestType value. [When the host controller function is selected] The USB request data value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1. [When the function controller function is selected] These bits indicate the USB request data value received during the setup transaction. Writing to these bits is invalid.		R/W Note

Note When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

13.3.21 USB request value register (USBVAL)

Figure 13 - 29 Format of USB Request Value Register (USBVAL)

Address: F0456H, F0457H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBVAL	WVALUE[15:0]															

WVALUE[15:0]	Value	R/W
These bits store the USB request wValue value. [When the host controller function is selected] The USB request wValue value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1. [When the function controller function is selected] These bits indicate the USB request wValue value received during the setup transaction. Writing to these bits is invalid.		R/W Note

Note When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

13.3.22 USB request index register (USBINDX)

Figure 13 - 30 Format of USB Request Index Register (USBINDX)

Address: F0458H, F0459H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBINDX	WINDEX[15:0]															

WINDEX[15:0]	Index	R/W
These bits store the USB request wIndex value. [When the host controller function is selected] The USB request wIndex value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ in the DCPCTR register is 1. [When the function controller function is selected] These bits indicate the USB request wIndex value received during the setup transaction. Writing to these bits is invalid.		R/W Note

Note When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

13.3.23 USB request length register (USBLENG)

Figure 13 - 31 Format of USB Request Length Register (USBLENG)

Address: F045AH, F045BH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBLENG	WLENGTH[15:0]															

WLENGTH[15:0]	Length	R/W
These bits store the USB request wLength value. [When the host controller function is selected] The USB request wLength value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ in the DCPCTR register is 1. [When the function controller function is selected] These bits indicate the USB request wLength value received during the setup transaction. Writing to these bits is invalid.		R/W Note

Note When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

13.3.24 DCP configuration register (DCPCFG)

Figure 13 - 32 Format of DCP Configuration Register (DCPCFG)

Address: F045CH, F045DH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCPCFG	—	—	—	—	—	—	—	—	SHTNAK	—	—	DIR	—	—	—	—

Bits 15 to 8	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

SHTNAK	Pipe disabled at end of transfer Note	R/W
0	Pipe continued at the end of transfer	R/W
1	Pipe disabled at the end of transfer	
Specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction. The SHTNAK bit is valid when the selected pipe in the receiving direction. When the SHTNAK bit is set to 1, the USB module modifies the PID bits for the DCP to NAK on determining the end of the transfer. The USB module determines that the transfer has ended on the following condition. • A short packet (including a zero-length packet) is successfully received.		

Bits 6, 5	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DIR	Transfer direction Note	R/W
0	Data receiving direction	R/W
1	Data transmitting direction	
When the host controller function is selected, the DIR bit sets the transfer direction of the data stage and status stage. When the function controller function is selected, the DIR bit should be set to 0.		

Bits 3 to 0	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

Note Modify these bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

13.3.25 DCP maximum packet size register (DCPMAXP)

Figure 13 - 33 Format of DCP Maximum Packet Size Register (DCPMAXP)

Address: F045EH, F045FH After reset: 0040H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCPMAXP	—	DEVSEL2	DEVSEL1	DEVSEL0	—	—	—	—	—	MXPS[6:0]						

Bit 15	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DEVSEL2	DEVSEL1	DEVSEL0	Device select Note 1	R/W
0	0	0	USB address 000	R/W
0	0	1	USB address 001	
0	1	0	USB address 010	
0	1	1	USB address 011	
1	0	0	USB address 100	
1	0	1	USB address 101	
Other than above			Do not set.	
When the host controller function is selected, the DEVSEL2 to DEVSEL0 bits specify the address of the peripheral device which is the communication target during control transfer. The DEVSEL2 to DEVSEL0 bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in these bits. For example, before setting DEVSEL2 to DEVSEL0 = 010B, the address should be set to DEVADD2. When the function controller function is selected, these bits should be set to 000B.				

Bits 11 to 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

MXPS[6:0]	Maximum packet size Note 2	R/W
These bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 40H (64 bytes). These bits should be set to the value based on the USB Specifications. While the MXPS bits are 0, do not write to the FIFO buffer or do not set PID to BUF.		R/W

Note 1. Modify the DEVSEL bits while PID is NAK and the SUREQ bit is 0. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 2. Modify the MXPS[6:0] bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE bits, clear the buffer by setting BCLR to 1.

13.3.26 DCP control register (DCPCTR)

Figure 13 - 34 Format of DCP Control Register (DCPCTR)

Address: F0460H, F0461H After reset: 0040H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCPCTR	BSTS	SUREQ	—	—	SUREQCLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID1	PID0

BSTS	Buffer status	R/W
0	Buffer access is enabled.	R
1	Buffer access is disabled.	
Indicates whether DCP FIFO buffer access is enabled or disabled. The meaning of the BSTS bit depends on the ISEL bit setting as follows. • When ISEL is 0, the BSTS bit indicates whether the received data can be read from the buffer. • When ISEL is 1, the BSTS bit indicates whether the data to be transmitted can be written to the buffer.		

SUREQ	Setup token transmission	R/W
0	Invalid	R/W Note 1
1	Transmits the setup packet.	
The USB module transmits the setup packet by setting the SUREQ bit to 1 when the host controller function is selected. After completing the setup transaction process, the USB module generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0. The USB module also clears the SUREQ bit to 0 when software sets the SUREQCLR bit to 1. Before setting the SUREQ bit to 1, set the DEVSEL bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the desired USB request in the setup transaction. Before setting SUREQ to 1, check that the PID bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not modify the DEVSEL bits, the USBREQ, USBVAL, USBINDX, or USBLENG register until the setup transaction is completed (SUREQ = 1). Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0. When the function controller function is selected, be sure to write 0 to the SUREQ bit.		

Bits 13, 12	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

SUREQCLR	SUREQ bit clear	R/W
0	Invalid	R/W Note 2
1	0 is written to the SUREQ bit.	
When the host controller function is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit always indicates 0. Set the SUREQCLR bit to 1 through software when communication has stopped with SUREQ being 1 during the setup transaction. However, for normal setup transactions, the USB module automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary. Controlling the SUREQ bit through the SUREQCLR bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected. When the function controller function is selected, be sure to write 0 to the SUREQCLR bit.		

Bits 10, 9	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

SQCLR	Toggle bit clear Note 2	R/W
0	Invalid	R/W Note 3
1	Specifies DATA0.	
Specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit always indicates 0. Do not set the SQCLR and SQSET bits to 1 simultaneously.		

SQSET	Toggle bit set Note 2	R/W
0	Invalid	R/W Note 3
1	Specifies DATA1.	
Specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. Do not set the SQCLR and SQSET bits to 1 simultaneously.		

SQMON	Sequence toggle bit monitor	R/W
0	DATA0	R
1	DATA1	
Indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The USB module allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction. When the function controller function is selected, the USB module sets the SQMON bit to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet. When the function controller function is selected, the USB module does not reference the SQMON bit during the IN/OUT transaction of the status stage, and does not allow the SQMON bit to toggle upon normal completion.		

PBUSY	Pipe busy	R/W
0	DCP is not used for the transaction.	R
1	DCP is used for the transaction.	
Indicates whether DCP is used or not for the transaction when USB changes the PID bits from BUF to NAK. The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction. Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible. For details, refer to **13.4.4.1 Pipe control register switching procedures**.		

Bits 4, 3	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

CCPL	Control transfer end enable	R/W
0	Completion of control transfer is disabled.	R/W
1	Completion of control transfer is enabled.	

When the function controller function is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.
When software sets the CCPL bit to 1 while the corresponding PID bits are set to BUF, the USB module completes the control transfer stage.
Specifically, during control read transfer, the USB module transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB module operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit.
The USB module modifies the CCPL bit from 1 to 0 on receiving a new setup packet.
Software cannot write 1 to the CCPL bit while VALID is 1.
When the host controller function is selected, be sure to write 0 to the CCPL bit.

PID1	PID0	Response PID	R/W
0	0	NAK response	R/W
0	1	BUF response (depending on the buffer state)	
1	0	STALL response	
1	1	STALL response	

The PID1 and PID0 bits control the response type of the USB module during control transfer.

[When the host controller function is selected]
Modify the setting of the PID1 and PID0 bits from NAK to BUF using the following procedure.
- When the transmitting direction is set
  Write all the transmit data to the FIFO buffer while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, the USB module executes the OUT transaction.
- When the receiving direction is set
  Check that the FIFO buffer is empty (or empty the buffer) while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, the USB module executes the IN transaction.

The USB module modifies the setting of the PID1 and PID0 bits as follows.
- The USB module sets PID to STALL (11B) on receiving the data of a size exceeding the maximum packet size when software has set the PID1 and PID0 bits to BUF.
- The USB module sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times.
- The USB module also sets PID to STALL (11B) on receiving the STALL handshake.

[When the function controller function is selected]
The USB module modifies the setting of the PID1 and PID0 bits as follows.
- The USB module modifies the PID1 and PID0 bits to NAK on receiving the setup packet. Here, the USB module sets VALID to 1. Software cannot modify the setting of the PID1 and PID0 bits until software sets VALID to 0.
- The USB module sets PID to STALL (11B) on receiving the data of a size exceeding the maximum packet size when software has set the PID1 and PID0 bits to BUF.
- The USB module sets PID to STALL (1xB) on detecting the control transfer sequence error.
- The USB module sets PID to NAK on detecting the USB bus reset.

The USB module does not reference to the setting of the PID1 and PID0 bits while the SET_ADDRESS request is processed (auto processing).

Note 1. Only 1 can be written.
Note 2. Write 1 to the SQSET and SQCLR bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.
Note 3. This bit is always read as 0. Only 1 can be written.

13.3.27 Pipe window select register (PIPESEL)

Figure 13 - 35 Format of Pipe Window Select Register (PIPESEL)

Address: F0464H, F0465H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPESEL	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL3	PIPESEL2	PIPESEL1	PIPESEL0

Bits 15 to 4	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PIPESEL3	PIPESEL2	PIPESEL1	PIPESEL0	Pipe window select	R/W
0	0	0	0	No pipe selected	R/W
0	1	0	0	PIPE4	
0	1	0	1	PIPE5	
0	1	1	0	PIPE6	
0	1	1	1	PIPE7	
Other than above				Do not set.	
The PIPESEL3 to PIPESEL0 bits select the pipe number corresponding to the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers which data are written to or read from. Selecting a pipe number through these bits allows writing to and reading from the PIPECFG, PIPEMAXP, and PIPEPERI registers which correspond to the selected pipe number. When the PIPESEL3 to PIPESEL0 bits are set to 0000B, 0 is read from all of the bits in the PIPECFG, PIPEMAXP, and PIPEPERI registers. Writing to these bits is invalid.					

Caution After selecting the pipe using the PIPESEL register, functions of the pipe should be set using the PIPECFG, PIPEMAXP, and PIPEPERI registers. The PIPEnCTR, PIPEnTRE, and PIPEnTRN registers can be set regardless of the pipe selection in the PIPESEL register.

13.3.28 Pipe configuration register (PIPECFG)

Figure 13 - 36 Format of Pipe Configuration Register (PIPECFG)

Address: F0468H, F0469H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPECFG	TYPE1	TYPE0	—	—	—	BFRE	DBLB	—	SHTNAK	—	—	—	DIR	EPNUM[3:0]		

TYPE1	TYPE0	Transfer type Note 1	R/W
PIPE4, PIPE5			R/W
0	0	Pipe not used	
0	1	Bulk transfer	
1	0	Do not set.	
1	1	Do not set.	
PIPE6, PIPE7			
0	0	Pipe not used	
0	1	Do not set.	
1	0	Interrupt transfer	
1	1	Do not set.	
These bits select the transfer type for the pipe selected by the PIPESEL3 to PIPESEL0 bits (selected pipe). Before setting PID to BUF for the selected pipe (before starting USB communication using the selected pipe), be sure to set the TYPE1 and TYPE0 bits to a value other than 00B.			

Bits 13 to 11	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

BFRE	BRDY interrupt operation specification Note 2, 3	R/W
0	BRDY interrupt upon transmitting or receiving data	R/W
1	BRDY interrupt upon completion of reading data	
Specifies the BRDY interrupt generation timing from the USB module to the CPU with respect to the selected pipe. When software has set the BFRE bit to 1 and the selected pipe is in the receiving direction, the USB module detects the transfer completion and generates the BRDY interrupt on having read the relevant packet. When the BRDY interrupt is generated with the above conditions, software needs to write 1 to the BCLR bit. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit. When software has set the BFRE bit to 1 and the selected pipe is in the transmitting direction, the USB module does not generate the BRDY interrupt. For details, refer to 13.4.3.1 BRDY interrupt		

DBLB	Double buffer mode Note 2, 3	R/W
0	Single buffer	R/W
1	Double buffer	
Selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The DBLB bit is valid when PIPE4 and PIPE5 are selected.		

Bit 8	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

SHTNAK	Pipe disabled at end of transfer Note 1	R/W
0	Pipe continued at the end of transfer	R/W
1	Pipe disabled at the end of transfer	
Specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction. The SHTNAK bit is valid when the selected pipe is PIPE4 and PIPE5 in the receiving direction. When software has set the SHTNAK bit to 1 for the selected pipe in the receiving direction, the USB module modifies the PID bits corresponding to the selected pipe to NAK on determining the end of the transfer. The USB module determines that the transfer has ended on any of the following conditions. • A short packet (including a zero-length packet) is successfully received. • The transaction counter is used and the number of packets specified by the counter are successfully received.		

Bits 6, 5	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DIR	Transfer direction Note 2, 3	R/W
0	Receiving direction	R/W
1	Transmitting direction	
Specifies the transfer direction for the selected pipe. When software has set the DIR bit to 0, the USB module uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB module uses the selected pipe in the transmitting direction.		

EPNUM[3:0]	Endpoint number Note 1	R/W
These bits specify the endpoint number for the selected pipe. Setting 0000B means unused pipe. During function controller operation, do not allow the combination of the settings of the DIR and EPNUM bits to be the same for two or more pipes (EPNUM = 0000B can be set for all of the pipes).		R/W

Note 1. Modify the TYPE1 and TYPE0, SHTNAK, and EPNUM bits while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 2. Modify the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected by the CURPIPE3 to CURPIPE0 bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the state described in the above note 2.

13.3.29 Pipe maximum packet size register (PIPEMAXP)

Figure 13 - 37 Format of Pipe Maximum Packet Size Register (PIPEMAXP)

Address: F046CH, F046DH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEMAXP	—	DEVSEL2	DEVSEL1	DEVSEL0	—	—	—	MXPS[8:0]								

Bit 15	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DEVSEL2	DEVSEL1	DEVSEL0	Device select Note 1	R/W
0	0	0	USB address 000	R/W
0	0	1	USB address 001	
0	1	0	USB address 010	
0	1	1	USB address 011	
1	0	0	USB address 100	
1	0	1	USB address 101	
Other than above			Do not set.	
When the host controller function is selected, the DEVSEL2 to DEVSEL0 bits specify the USB device address of the peripheral device which is the communication target. The DEVSEL2 to DEVSEL0 bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in these bits. For example, before setting the DEVSEL bits to 010B, the address should be set to DEVADD2. When the function controller function is selected, these bits should be set to 000B.				

Bits 11 to 9	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

MXPS[8:0]	Maximum packet size Note 2	R/W
These bits specify the maximum data payload (maximum packet size). The range of values can be set for each pipe is shown as follows: PIPE4, PIPE5: 8 bytes (008H), 16 bytes (010H), 32 bytes (020H), 64 bytes (040H) (Bits [8:7] and [2:0] are not provided.) PIPE6, PIPE7: 1 byte (001H) to 64 bytes (040H) (Bits [8:7] are not provided.) The MXPS bits should be set to the appropriate value for each transfer type based on the USB Specifications. While the MXPS bits are 0, do not write to the FIFO buffer or set PID to BUF.		R/W

Note 1. Modify the DEVSEL bits while PID is NAK and the SUREQ bit is 0. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 2. Modify the MXPS bits while PID is NAK and before the pipe is selected by the CURPIPE bits.
Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Caution The value of the PIPEMAXP register after reset differs depending on when no pipe is selected and when a pipe is selected with the PIPESEL3 to PIPESEL0 bits in the PIPESEL register.
The value after reset is 0000H when no pipe is selected and 0040H when a pipe is selected.

13.3.30 Pipe cycle control register (PIPEPERI)

Figure 13 - 38 Format of Pipe Cycle Control Register (PIPEPERI)

Address: F046EH, F046FH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEPERI	—	—	—	—	—	—	—	—	—	—	—	—	—	IITV[2:0]		

Bits 15 to 13	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

Bit 12	Reserved	R/W
—	The write value must be 0. The read value is 0.	—

Bits 11 to 3	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

IITV[2:0]	Interval error detection interval Note	R/W
Specifies the interval error detection timing for the selected pipe in terms of frames, which is expressed as n-th power of 2. For details on the functions, refer to **13.4.8 Interrupt transfers (PIPE6, PIPE7)**. Before modifying the IITV[2:0] bits after USB communication has been completed with the IITV[2:0] bits set to a certain value, set PID to NAK and then set ACLRM to 1 to initialize the interval timer. The IITV[2:0] bits are invalid for PIPE4 and PIPE5. Set these bits to 000B for PIPE4 and PIPE5.		R/W

Note Modify the IITV bits while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

13.3.31 PIPEn control registers (PIPEnCTR) (n = 4 to 7)

Figure 13 - 39 Format of PIPEn Control Registers (PIPEnCTR) (n = 4, 5)

Address: F0476H, F0477H (PIPE4CTR), F0478H, F0479H (PIPE5CTR) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEnCTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID1	PID0

BSTS	Buffer status	R/W
0	Buffer access by the CPU is disabled.	R
1	Buffer access by the CPU is enabled.	
Indicates the FIFO buffer status for the relevant pipe. The meaning of the BSTS bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in Table 13 - 8.		

INBUFM	Transmit buffer monitor	R/W
0	There is no data to be transmitted in the buffer memory.	R
1	There is data to be transmitted in the buffer memory.	
Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction. When the relevant pipe is in the transmitting direction (DIR = 1), the USB module sets the INBUFM bit to 1 when software completes writing data to at least one FIFO buffer plane. The USB module sets the INBUFM bit to 0 when the USB module completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (DBLB = 1), the USB module sets the INBUFM bit to 0 when the USB module completes transmitting the data from the two FIFO buffer planes before software completes writing data to one FIFO buffer plane. The INBUFM bit indicates the same value as the BSTS bit when the relevant pipe is in the receiving direction (DIR = 0).		

Bits 13 to 11	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

ATREPM	Auto response mode Note 1	R/W
0	Auto response is disabled.	R/W
1	Auto response is enabled.	
Enables or disables auto response mode for the relevant pipe. When the function controller function is selected and the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1. When the ATREPM bit is set to 1, the USB module responds to the token from the USB host as described below. [When the relevant pipe is for bulk IN transfer (TYPE bits = 01B and DIR = 1)] When ATREPM is 1 and PID is BUF, the USB module transmits a zero-length packet in response to the IN token. The USB module updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USB module receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.). In this case, the USB module does not generate the BRDY or BEMP interrupt. [When the relevant pipe is for bulk OUT transfer (TYPE bits = 01B and DIR = 0)] When ATREPM is 1 and PID is BUF, the USB module returns NAK in response to the OUT token and generates the NRDY interrupt. For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. When the host controller function is selected, be sure to set the ATREPM bit to 0.		

ACLRM	Auto buffer clear mode Note 2	R/W
0	Disabled	R/W
1	Enabled (all buffers are initialized)	
Enables or disables auto buffer clear mode for the relevant pipe. To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously. Table 13 - 9 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.		

SQCLR	Toggle bit clear Note 1	R/W
0	Invalid	R/W Note 3
1	Specifies DATA0.	
The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQCLR bit to 1 through software allows the USB module to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQCLR bit to 0.		

SQSET	Toggle bit set Note 1	R/W
0	Invalid	R/W Note 3
1	Specifies DATA1.	
The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQSET bit to 1 through software allows the USB module to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQSET bit to 0.		

SQMON	Toggle bit monitor	R/W
0	DATA0	R
1	DATA1	
Indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe. The USB module allows the SQMON bit to toggle upon normal completion of the transaction of the relevant pipe. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.		

PBUSY	Pipe busy	R/W
0	The relevant pipe is not used for the transaction.	R
1	The relevant pipe is used for the transaction.	
Indicates whether the relevant pipe is being currently used or not for the transaction. The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction. Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible. For details, refer to **13.4.4.1 Pipe control register switching procedures**.		

Bits 4 to 2	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PID1	PID0	Response PID	R/W
0	0	NAK response	R/W
0	1	BUF response (depending on the buffer state)	
1	0	STALL response	
1	1	STALL response	

The PID1 and PID0 bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID1 and PID0 bits is NAK. Modify the setting of these bits to BUF to use the relevant pipe for USB transfer. Tables 13 - 10 and 13 - 11 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB module depending on the PID bit setting.
After modifying the setting of the PID1 and PID0 bits through software from BUF to NAK during USB communication using the relevant pipe, check that PBUSY is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state.
The USB module modifies the setting of the PID1 and PID0 bits in the following cases.

- The USB module sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1.
- The USB module sets PID to STALL (11B) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB module sets PID to NAK on detecting a USB bus reset when the function controller function is selected.
- The USB module sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller function is selected.
- The USB module sets PID to STALL (11B) on receiving the STALL handshake when the host controller function is selected.

To specify each response type, set the PID1 and PID0 bits as follows.

- To make a transition from NAK (00B) to STALL, write 10B.
- To make a transition from BUF (01B) to STALL, write 11B.
- To make a transition from STALL (11B) to NAK, write 10B and then 00B.
- To make a transition from STALL to BUF, write 00B (NAK) and then 01B (BUF).

Note 1. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 2. Modify the ATREPM bit while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 3. Only 0 can be read and 1 can be written.

Table 13 - 8 Operation of BSTS Bit

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Bit Function
0	0	0	1 when the received data can be read from the FIFO buffer; 0 when the received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	1 when the received data can be read from the FIFO buffer; 0 when software has set BCLR to 1 after the received data has been completely read from the FIFO buffer.
		1	1 when the received data can be read from the FIFO buffer; 0 when the received data has been completely read from the FIFO buffer.
1	0	0	1 when the transmit data can be written to the FIFO buffer; 0 when the transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

Table 13 - 9 Information Cleared by USB Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected)	
2	Internal flags concerning the BFRE bit	When the BFRE setting is modified
3	FIFO buffer toggle control	When the DBLB setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

Table 13 - 10 Operation of USB Module depending on PID Bit Setting (When Host Controller Function is Selected)

PID Bits (PID1 and PID0)	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB Module
00 (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01 (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while UACT is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens while UACT is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
10 (STALL) or 11 (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

Table 13 - 11 Operation of USB Module depending on PID Bit Setting (When Function Controller Function is Selected)

PID Bits (PID1 and PID0)	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB Module
00 (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host For the operation when ATREPM is 1, refer to the description of the ATREPM bit.
01 (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
10 (STALL) or 11 (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.

Figure 13 - 40 Format of PIPEn Control Registers (PIPEnCTR) (n = 6, 7)

Address: F047AH, F047BH (PIPE6CTR), F047CH, F047DH (PIPE7CTR) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEnCTR	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID1	PID0

BSTS	Buffer status	R/W
0	Buffer access by the CPU is disabled.	R
1	Buffer access by the CPU is enabled.	
Indicates the FIFO buffer status for the relevant pipe. The meaning of the BSTS bit depends on the settings of the DIR, BFRE, and DCLR bits as shown in Table 13 - 8.		

Bits 14 to 10	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

ACLRM	Auto buffer clear mode Note 1	R/W
0	Auto buffer clear mode is disabled.	R/W
1	Auto buffer clear mode is enabled (all buffers are initialized)	
Enables or disables auto buffer clear mode for the relevant pipe. To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously. Table 13 - 12 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.		

SQCLR	Toggle bit clear Note 2	R/W
0	Invalid	R/W Note 3
1	Specifies DATA0.	
The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQCLR bit to 1 through software allows the USB module to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQCLR bit to 0.		

SQSET	Toggle bit set Note 2	R/W
0	Invalid	R/W Note 3
1	Specifies DATA1.	
The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQSET bit to 1 through software allows the USB module to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQSET bit to 0.		

SQMON	Toggle bit monitor	R/W
0	DATA0	R
1	DATA1	
Indicates the value of the sequence toggle bit for the next transaction of the relevant pipe. When the relevant pipe is not for the isochronous transfer, the USB module allows the SQMON bit to toggle upon normal completion of the transaction of the relevant pipe. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.		

PBUSY	Pipe busy	R/W
0	The relevant pipe is not used for the transaction.	R
1	The relevant pipe is used for the transaction.	
Indicates whether the relevant pipe is being currently used or not for the transaction. The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction. Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible. For details, refer to **13.4.4.1 Pipe control register switching procedures**.		

Bits 4 to 2	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PID1	PID0	Response PID	R/W
0	0	NAK response	R/W
0	1	BUF response (depending on the buffer state)	
1	0	STALL response	
1	1	STALL response	

The PID1 and PID0 bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID1 and PID0 bits is NAK. Modify the setting of these bits to BUF to use the relevant pipe for USB transfer. Tables 13 - 10 and 13 - 11 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB module depending on the PID bit setting.
After modifying the setting of the PID1 and PID0 bits through software from BUF to NAK during USB communication using the relevant pipe, check that PBUSY is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state.
The USB module modifies the setting of the PID1 and PID0 bits in the following cases.

- The USB module sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1.
- The USB module sets PID to STALL (11B) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB module sets PID to NAK on detecting a USB bus reset when the function controller function is selected.
- The USB module sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller function is selected.
- The USB module sets PID to STALL (11B) on receiving the STALL handshake when the host controller function is selected.

To specify each response type, set the PID1 and PID0 bits as follows.

- To make a transition from NAK (00B) to STALL, write 10B.
- To make a transition from BUF (01B) to STALL, write 11B.
- To make a transition from STALL (11B) to NAK, write 10B and then 00B.
- To make a transition from STALL to BUF, write 00B (NAK) and then 01B (BUF).

Note 1. Modify the ACLRM bit while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 2. Write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 3. Only 0 can be read and 1 can be written.

Table 13 - 12 Information Cleared by USB Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	
2	The interval count value when the selected pipe is for interrupt transfer and the host controller function is selected	When the interval count value is to be reset
3	Internal flags concerning the BFRE bit	When the BFRE setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

13.3.32 PIPEn transaction counter enable registers (PIPEnTRE) (n = 4, 5)

Figure 13 - 41 Format of PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 4, 5)

Address: F049CH, F049DH (PIPE4TRE), F04A0H, F04A1H (PIPE5TRE) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEnTRE	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—

Bits 15 to 10	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

TRENB	Transaction counter enable	R/W
0	Transaction counter is disabled.	R/W
1	Transaction counter is enabled.	
Enables or disables the transaction counter. For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the TRNCNT bits through software allows the USB module to control hardware as described below on having received the number of packets equal to the setting of the TRNCNT bits. • While SHTNAK is 1, the USB module modifies the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the setting of the TRNCNT bits. • While BFRE is 1, the USB module asserts the BRDY interrupt on having received the number of packets equal to the setting of the TRNCNT bits and then reading out the last received data. For the pipe in the transmitting direction, set the TRENB bit to 0. When the transaction counter is not used, set the TRENB bit to 0. When the transaction counter is used, set the TRNCNT bits before setting the TRENB bit to 1. Set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.		

TRCLR	Transaction counter clear	R/W
0	Invalid	R/W
1	The current counter value is cleared.	
Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.		

Bits 7 to 0	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

Caution **Modify each bit in the PIPEnTRE register while CSST is 1 and PID is NAK.**
Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

13.3.33 PIPEn transaction counter registers (PIPEnTRN) (n = 4, 5)

Figure 13 - 42 Format of PIPEn Transaction Counter Registers (PIPEnTRN) (4, 5)

Address: F049EH, F049FH (PIPE4TRN), F04A2H, F04A3H (PIPE5TRN) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEnTRN	TRNCNT[15:0]															

TRNCNT[15:0]	Transaction counter	R/W
[When written to] Specifies the number of transactions. [When read from] Indicates the specified number of transactions if TRENB is 0. Indicates the number of currently counted transactions if TRENB is 1. The USB module increments the value of the TRNCNT bits by one when all of the following conditions are satisfied on receiving the packet. • TRENB = 1 • (TRNCNT set value ≠ current counter value + 1) on receiving the packet. • The payload of the received packet agrees with the setting of the MXPS bits. The USB module clears the value of the TRNCNT bits to 0 when any of the following conditions are satisfied. (1) All of the following conditions are satisfied. • TRENB = 1 • (TRNCNT set value = current counter value + 1) on receiving the packet. • The payload of the received packet agrees with the setting of the MXPS bits. (2) All of the following conditions are satisfied. • TRENB = 1 • The USB module has received a short packet. (3) All of the following conditions are satisfied. • TRENB = 1 • Software has set the TRCLR bit to 1. For the pipe in the transmitting direction, set the TRNCNT bits to 0. When the transaction counter is not used, set the TRNCNT bits to 0. Setting the number of transactions to be transferred to the TRNCNT bits is only enabled when the TRENB bit in the PIPEnTRE register is 0. To modify the number of transactions to be transferred, set the TRCLR bit in the PIPEnTRE register to 1 (to clear the current counter value) before setting TRENB to 1.		R/W

13.3.34 BC control register n (USBBCCTRLn) (n = 0, 1)

Figure 13 - 43 Format of BC Control Register 0 (USBBCCTRL0)

Address: F04B0H, F04B1H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBBCCTRL0	—	—	—	—	—	—	PDDETSTS0	CHGDETSTS0	BATCHGE0	DCPMODE0	VDMSRCE0	IDPSINKE0	VDPSRCE0	IDMSINKE0	IDPSRCE0	RPDME0

Bits 15 to 10	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PDDETSTS0	V_{DP_SRC} (0.6 V) input detection flag for UDP0 pin	R/W
0	Not detected	R
1	Detected	
Detects V_{DP_SRC} (0.6 V) is applied to the UDP0 pin from the connected device. (Detects that the voltage applied to UDP0 is in the range of V_{DAT_REF} to V_{IH} (UDP0).) To use this bit for detection, set the CNEN bit (bit 8) in the SYSCFG register to 1, and enable the single end receiver of USB port 0. Valid when IDPSINKE0 is 1.		

CHGDETSTS0	V_{DM_SRC} (0.6 V) input detection flag for UDM0 pin	R/W
0	Not detected	R
1	Detected	
Detects V_{DM_SRC} (0.6 V) is applied to the UDM0 pin from the connected device. (Detects that the voltage applied to UDM0 is in the range of V_{DAT_REF} to V_{IH} (UDM0).) To use this bit for detection, set the CNEN bit (bit 8) in the SYSCFG register to 1, and enable the single end receiver of USB port 0. Valid when IDMSINKE0 is 1.		

BATCHGE0	USB port 0 BC connection detection operation enable	R/W
0	Operation disabled	R/W
1	Operation enabled	
When this bit is set to enabled, each bit setting of VDPSRCE0, VDMSRCE0, IDPSINKE0, IDMSINKE0, and IDPSRCE0 is valid BC connection detection can be operated via USB port 0.		

DCPMODE0	Dedicated charging port resistor connection control for UDP0/UDM0 pins	R/W
0	Resistor disabled	R/W
1	Resistor enabled	
Connects the resistor (R_{DCP_DAT}) between the UDP0 and UDM0 pins used for the host (dedicated charging port) BC connection detection function.		

VDMSRCE0	UDM0 pin V_{DM_SRC} (0.6 V) output control	R/W
0	V_{DM_SRC} output disabled	R/W
1	V_{DM_SRC} output enabled (0.6 V output)	
Controls the V_{DM_SRC} output.		

IDPSINKE0	UDP0 pin V_{DP_SRC} (0.6 V) input detection (comparator and sink) control	R/W
0	UDP0 pin (0.6 V) input detection disabled	R/W
1	UDP0 pin (0.6 V) input detection enabled	
Controls the I_{DP_SINK} (sink current) used for the 0.6 V input detection circuit (comparator) and detection for the UDP0 pin.		

VDPSRCE0	UDP0 pin V_{DP_SRC} (0.6 V) output control	R/W
0	V_{DP_SRC} output disabled	R/W
1	V_{DP_SRC} output enabled (0.6 V output)	
Controls the V_{DP_SRC} output.		

IDMSINKE0	UDM0 pin V_{DM_SRC} (0.6 V) input detection (comparator and sink) control	R/W
0	UDM0 pin (0.6 V) input detection disabled	R/W
1	UDM0 pin (0.6 V) input detection enabled	
Controls the I_{DM_SINK} (sink current) used for the 0.6 V input detection circuit (comparator) and detection for the UDM0 pin.		

IDPSRCE0	UDP0 pin I_{DP_SRC} (10 µA) output control	R/W
0	I_{DP_SRC} output disabled	R/W
1	I_{DP_SRC} output enabled (10 µA output)	
Controls the I_{DP_SRC} output.		

RPDME0	UDM0 pull-down control	R/W
0	Pulling down disabled	R/W
1	Pulling down enabled	
Only the UDM0 pin can be pulled down (R_{PD}) by setting this bit.		

Figure 13 - 44 Format of BC Control Register 1 (USBBCCTRL1)

Address: F04B4H, F04B5H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBBCCTRL1	—	—	—	—	—	—	PDDETSTS1	CHGDETSTS1	BATCHGE1	DCPMODE1	VDMSRCE1	IDPSINKE1	VDPSRCE1	IDMSINKE1	IDPSRCE1	RPDME1

Bits 15 to 10	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PDDETSTS1	V_{DP_SRC} (0.6 V) input detection flag for UDP1 pin	R/W
0	Not detected	R/W
1	Detected	
Detects V_{DP_SRC} (0.6 V) is applied to the UDP1 pin from the connected device. (Detects that the voltage applied to UDP1 is in the range of V_{DAT_REF} to V_{IH} (UDP1).) To use this bit for detection, set the CNEN bit (bit 8) in the SYSCFG1 register to 1, and enable the single end receiver of USB port 1. Valid when IDPSINKE1 is 1.		

CHGDETSTS1	V_{DM_SRC} (0.6 V) input detection flag for UDM1 pin	R/W
0	Not detected	R
1	Detected	
Detects V_{DM_SRC} (0.6 V) is applied to the UDM1 pin from the connected device. (Detects that the voltage applied to UDM1 is in the range of V_{DAT_REF} to V_{IH} (UDM1).) To use this bit for detection, set the CNEN bit (bit 8) in the SYSCFG register to 1, and enable the single end receiver of USB port 1. Valid when IDMSINKE1 is 1.		

BATCHGE1	USB port 1 BC connection detection operation enable	R/W
0	Operation disabled	R/W
1	Operation enabled	
When this bit is set to enabled, each bit setting of VDMSRCE1, IDPSINKE1 is valid BC connection detection can be operated via USB port 1.		

DCPMODE1	Dedicated charging port resistor connection control for UDP1/UDM1 pins	R/W
0	Resistor disabled	R/W
1	Resistor enabled	
Connects the resistor (R_{DCP_DAT}) between the UDP1 and UDM1 pins used for the host (dedicated charging port) BC connection detection function.		

VDMSRCE1	UDM1 pin V_{DM_SRC} (0.6 V) output control	R/W
0	V_{DM_SRC} output disabled	R/W
1	V_{DM_SRC} output enabled (0.6 V output)	
Controls the V_{DM_SRC} output.		

IDPSINKE1	UDP1 pin V_{DP_SRC} (0.6 V) input detection (comparator and sink) control	R/W
0	UDP1 pin (0.6 V) input detection disabled	R/W
1	UDP1 pin (0.6 V) input detection enabled	
Controls the I_{DP_SINK} (sink current) used for the 0.6 V input detection circuit (comparator) and detection for the UDP1 pin.		

VDPSRCE1	UDP1 pin V_{DP_SRC} (0.6 V) output control	R/W
0	V_{DP_SRC} output disabled	R/W
1	V_{DP_SRC} output enabled (0.6 V output)	
Controls the V_{DP_SRC} output.		

IDMSINKE1	UDM1 pin V_{DM_SRC} (0.6 V) input detection (comparator and sink) control	R/W
0	UDM1 pin (0.6 V) input detection disabled	R/W
1	UDM1 pin (0.6 V) input detection enabled	
Controls the I_{DM_SINK} (sink current) used for the 0.6 V input detection circuit (comparator) and detection for the UDM1 pin.		

IDPSRCE1	UDP1 pin I_{DP_SRC} (10 μA) output control	R/W
0	I_{DP_SRC} output disabled	R/W
1	I_{DP_SRC} output enabled (10 μA output)	
Controls the I_{DP_SRC} output.		

RPDME1	UDM1 pull-down control	R/W
0	Pulling down disabled	R/W
1	Pulling down enabled	
Only the UDM1 pin can be pulled down (R_{PD}) by setting this bit.		

13.3.35 BC option control register n (USBBCOPTn) (n = 0, 1)

Figure 13 - 45 Format of BC Option Control Register 0 (USBBCOPT0)

Address: F04B8H, F04B9H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBBCOPT0	—	—	—	—	—	—	DMCUSDET0	DPCUSDET0	—	CUSDETE0	VDOUTE0	—	VDSEL03	VDSEL02	VDSEL01	VDSEL00

Bits 15, 14	Reserved	R/W
—	The write value must be 0.	—

Bits 13 to 10	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DMCUSDET0 Note	UDM0 voltage detectiion (Option BC: Host/functiion)	R/W
0	• Host UDM0 pin output voltage is maintained (Within the comparison voltage range selected by the VDSEL0x bit). • Function UDM0 pin voltage is lower than the comparison voltage selected by the VDSEL0x bit.	R
1	• Host UDM0 pin output voltage change is detected (When the comparison voltage range selected by the VDSEL0x bit is exceeded). • Function UDM0 pin voltage exceeds the comparison voltage selected by the VDSEL0x bit.	

DPCUSDET0 Note	UDP0 voltage detectiion (Option BC: Host/functiion)	R/W
0	• Host UDP0 pin output voltage is maintained (Within the comparison voltage range selected by the VDSEL0x bit). • Function UDP0 pin voltage is lower than the comparison voltage selected by the VDSEL0x bit.	R
1	• Host UDP0 pin output voltage change is detected (When the comparison voltage range selected by the VDSEL0x bit is exceeded). • Function UDP0 pin voltage exceeds the comparison voltage selected by the VDSEL0x bit.	

Bit 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

CUSDETE0	Option voltage detectiion circuit control (Option BC: Host/functiion)	R/W
0	Disabled	R/W
1	Enabled	

VDOUTE0	Option voltage output control (Option BC: Host)	R/W
0	Disabled	R/W
1	Enabled	

Bit 4	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

VDSEL03 to VDSEL00	UDP0/UDM0 pin option output voltage and comparison voltage select (Option BC: Host/functiion)	R/W
Select UDP0/UDM0 pin output voltage and comparison voltage value (Refer to Remark)		R/W

Note Valid when CUSDETE0 = 1

Remark The option output voltage values of the UDP0 and UDM0 pins and the comparison voltage values of the option BC detection circuit when UV_{BUS} = 5.0 V are shown below. The following voltages vary in proportion to the UV_{BUS} input voltage.

VDOUTE0	VDSEL03	VDSEL02	VDSEL01	VDSEL00	Output voltage (V)		Comparison voltage (V) (Valid when CUSDETE0 = 1)	
					UDP0	UDM0	UDP0	UDM0
1 (Host)	1	0	0	0	2.00	2.00	1.60 to 2.60	1.60 to 2.60
	1	0	0	1	2.68	2.00	2.45 to 2.80	1.60 to 2.60
	1	0	1	0	2.00	2.68	1.60 to 2.60	2.45 to 2.80
	1	1	0	0	3.30	3.30	—	—
0 (Function)	0	0	0	0	—		1.60	
	0	0	0	1	—		1.70	
	0	0	1	0	—		1.85	
	0	0	1	1	—		2.00	
	0	1	0	0	—		2.15	
	0	1	0	1	—		2.30	
	0	1	1	0	—		2.45	
	0	1	1	1	—		2.60	
	1	0	0	0	—		2.80	
	1	0	0	1	—		3.00	
	1	0	1	0	—		3.20	
	1	0	1	1	—		3.40	
	1	1	0	0	—		3.60	
	1	1	0	1	—		3.80	
	1	1	1	0	—		4.00	
	1	1	1	1	—		4.20	
Other than above					Setting prohibited			

Figure 13 - 46 Format of BC Option Control Register 1 (USBBCOPT1)

Address: F04BCH, F04BDH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBBCOPT1	—	—	—	—	—	—	DMCUSDET1	DPCUSDET1	—	CUSDETE1	VDOUTE1	—	VDSEL13	VDSEL12	VDSEL11	VDSEL10

Bits 15, 14	Reserved	R/W
—	The write value must be 0.	—

Bits 13 to 10	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

DMCUSDET1 Note	UDM1 voltage detectiion (Option BC: Host/functiion)	R/W
0	• Host UDM1 pin output voltage is maintained (Within the comparison voltage range selected by the VDSEL1x bit). • Function UDM1 pin voltage is lower than the comparison voltage selected by the VDSEL1x bit.	R
1	• Host UDM1 pin output voltage change is detected (When the comparison voltage range selected by the VDSEL1x bit is exceeded). • Function UDM1 pin voltage exceeds the comparison voltage selected by the VDSEL1x bit.	

DPCUSDET1 Note	UDP1 voltage detectiion (Option BC: Host/functiion)	R/W
0	• Host UDP1 pin output voltage is maintained (Within the comparison voltage range selected by the VDSEL1x bit). • Function UDP1 pin voltage is lower than the comparison voltage selected by the VDSEL1x bit.	R
1	• Host UDP1 pin output voltage change is detected (When the comparison voltage range selected by the VDSEL1x bit is exceeded). • Function UDP1 pin voltage exceeds the comparison voltage selected by the VDSEL1x bit.	

Bit 7	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

CUSDETE1	Option voltage detectiion circuit control (Option BC: Host)	R/W
0	Disabled	R/W
1	Enabled	

VDOUTE1	Option voltage output control (Option BC: Host)	R/W
0	Disabled	R/W
1	Enabled	

Bit 4	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

VDSEL13 to VDSEL10	UDP1/UDM1 pin option output voltage and comparison voltage select (Option BC: Host)	R/W
Select UDP1/UDM1 pin output voltage and comparison voltage value (Refer to Remark)		R/W

Note Valid when CUSDETE1 = 1

Remark The option output voltage values of the UDP1 and UDM1 pins and the comparison voltage values of the option BC detection circuit when UV_{BUS} = 5.0 V are shown below. The following voltages vary in proportion to the UV_{BUS} input voltage.

VDOUTE1	VDSEL13	VDSEL12	VDSEL11	VDSEL10	Output voltage (V)		Comparison voltage (V) (Valid when CUSDETE1 = 1)	
					UDP1	UDM1	UDP1	UDM1
1 (Host)	1	0	0	0	2.00	2.00	1.60 to 2.60	1.60 to 2.60
	1	0	0	1	2.68	2.00	2.45 to 2.80	1.60 to 2.60
	1	0	1	0	2.00	2.68	1.60 to 2.60	2.45 to 2.80
	1	1	0	0	3.30	3.30	—	—
0 (Function)	0	0	0	0	—		1.60	
	0	0	0	1	—		1.70	
	0	0	1	0	—		1.85	
	0	0	1	1	—		2.00	
	0	1	0	0	—		2.15	
	0	1	0	1	—		2.30	
	0	1	1	0	—		2.45	
	0	1	1	1	—		2.60	
	1	0	0	0	—		2.80	
	1	0	0	1	—		3.00	
	1	0	1	0	—		3.20	
	1	0	1	1	—		3.40	
	1	1	0	0	—		3.60	
	1	1	0	1	—		3.80	
	1	1	1	0	—		4.00	
	1	1	1	1	—		4.20	
Other than above					Setting prohibited			

13.3.36 USB clock selection register (UCKSEL)

Figure 13 - 47 Format of USB clock selection register (UCKSEL)

Address: F04C4H, F04C5H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UCKSEL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	UCKSELC

UCKSELC	USB clock selection	R/W
0	High-speed on-chip oscillator clock (f_{HOCO}) is not selected as USB clock	R/W
1	High-speed on-chip oscillator clock (f_{HOCO}) is selected as USB clock	

Caution 1. When selecting the high-speed on-chip oscillator clock (f_{HOCO}) as the USB clock, set UCKSELC = 1 and the CKSELR bit in the MCKC register to 0 at the same time.

Caution 2. The USB clock select register can be rewritten only when the USB is disconnected.

Caution 3. The high-speed on-chip oscillator clock can be selected only when TA = -20 to +85°C.

Caution 4. If the high-speed on-chip oscillator clock is selected as the USB clock, when the USB is suspended, perform USB suspended processing while the high-speed on-chip oscillator clock is selected (UCKSELC = 1).

Caution 5. If the high-speed on-chip oscillator clock is selected as the USB clock, when the USB is disconnected, perform USB stopped processing (including setting DPRPU = 0) before setting UCKSELC = 0.

13.3.37 USB module control register (USBMC)

Figure 13 - 48 Format of USB Module Control Register (USBMC)

Address: F04CCH, F04CDH After reset: 0002H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBMC	—	—	—	—	—	—	—	—	VBRPDCUT	—	—	—	—	—	PXXCON	VDDUSBE

Bits 15 to 8	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

VBRPDCUT Note 1	UVBUS pin pull-down resistor control	R/W
0	Pull-down resistor is valid.	R/W
1	Pull-down resistor is invalid.	

Bits 6 to 2	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

PXXCON	VDDUSBE	USB internal power supply control	R/W
0	0	The internal power supply for USB is stopped UVDD pin input is pull-down setting Note 2	R/W
0	1	The internal power supply for USB is stopped Note 3 Setting for when power (3.3 V) is applied externally via the UVDD pin Note 3 Be sure to specify this setting when using the BC connection detection function (except the optional BC connection detection functions) as well as the above settings Note 4	
1	0	The internal power supply for USB is stopped Setting for when power (3.3 V) is applied externally via the UVDD pin Note 3	
1	1	The internal power supply for USB is turned on. Power (3.3 V) is generated internal USB power supply Note 3, 4	
Controlling to the USB internal power supply.			

Note 1. Be sure to set VBRPDCUT = 1 (pull-down resistor is invalid) when this pin is used as VBUS power supply in the BC option (host), etc. Set VBRPDCUT = 0 (pull-down resistor is valid) when the VBUS input is used, or this pin is not used.

Note 2. To prevent malfunction caused by a floating USB power supply, be sure to specify this setting when power is not being applied externally from the UVDD pin or when the internal USB power supply might stop.

Note 3. When PXXCON = 0 and VDDUSBE = 1 or PXXCON = 1 and VDDUSBE = 0, the high level of the UDP and UDM pins is based on the level of the external power supply input from the UVDD pin.
When PXXCON = 1 and VDDUSBE = 1, the high level of the UDP and UDM pins is based on the voltage generated by the internal USB power supply (3.3 V), and 3.3 V is output from the UVDD pin.

Note 4. Specify PXXCON = 0 and VDDUSBE = 1 when using the BC connection detection function (except the optional BC connection detection functions) with external power being applied to the UVDD pin. Specify PXXCON = 1 and VDDUSBE = 1 when using the BC connection detection function (except the optional BC connection detection functions) with the internal USB power supply. Note that the temperature sensor cannot be used and A/D conversion that uses the internal reference voltage cannot be performed when using the internal USB power supply or using the BC connection detection function (except the optional BC connection detection functions) (VDDUSBE = 1).

13.3.38 Device address n configuration registers (DEVADDn) (n = 0 to 5)

Figure 13 - 49 Format of Device Address n Configuration Registers (DEVADDn) (n = 0 to 5)

Address: F04D0H, F04D1H (DEVADD0), F04D2H, F04D3H (DEVADD1), F04D4H, F04D5H (DEVADD2), F04D6H, F04D7H (DEVADD3), F04D8H, F04D9H (DEVADD4), F04DAH, F04DBH (DEVADD5)

After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVADDn	—	—	—	—	—	—	—	—	USBSPD1	USBSPD0	—	—	—	—	—	RTPORT

Bits 15 to 8	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

USBSPD1	USBSPD0	Transfer speed of communication target device	R/W
0	0	The DEVADDn register is not used	R/W
0	1	Low speed	
1	0	Full speed	
1	1	Do not set.	
These bits specify the USB transfer speed of the communication target peripheral device. When the host controller function is selected, the USB module refers to the setting of the USBSPD[1:0] bits to generate packets. When the function controller function is selected, set these bits to 00B.			

Bits 5 to 1	Nothing is assigned	R/W
—	The write value must be 0. The read value is 0.	—

RTPORT	Route hub port number	R/W
0	USB port 0	R/W
1	USB port 1	
Specifies the corresponding device is connected to which port. When the host controller function is selected, the USB module references to the setting of the RTPORT bit and generates a packet. When the function controller function is selected, this setting is ignored.		

13.4 Operation

13.4.1 System control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

13.4.1.1 Starting operation

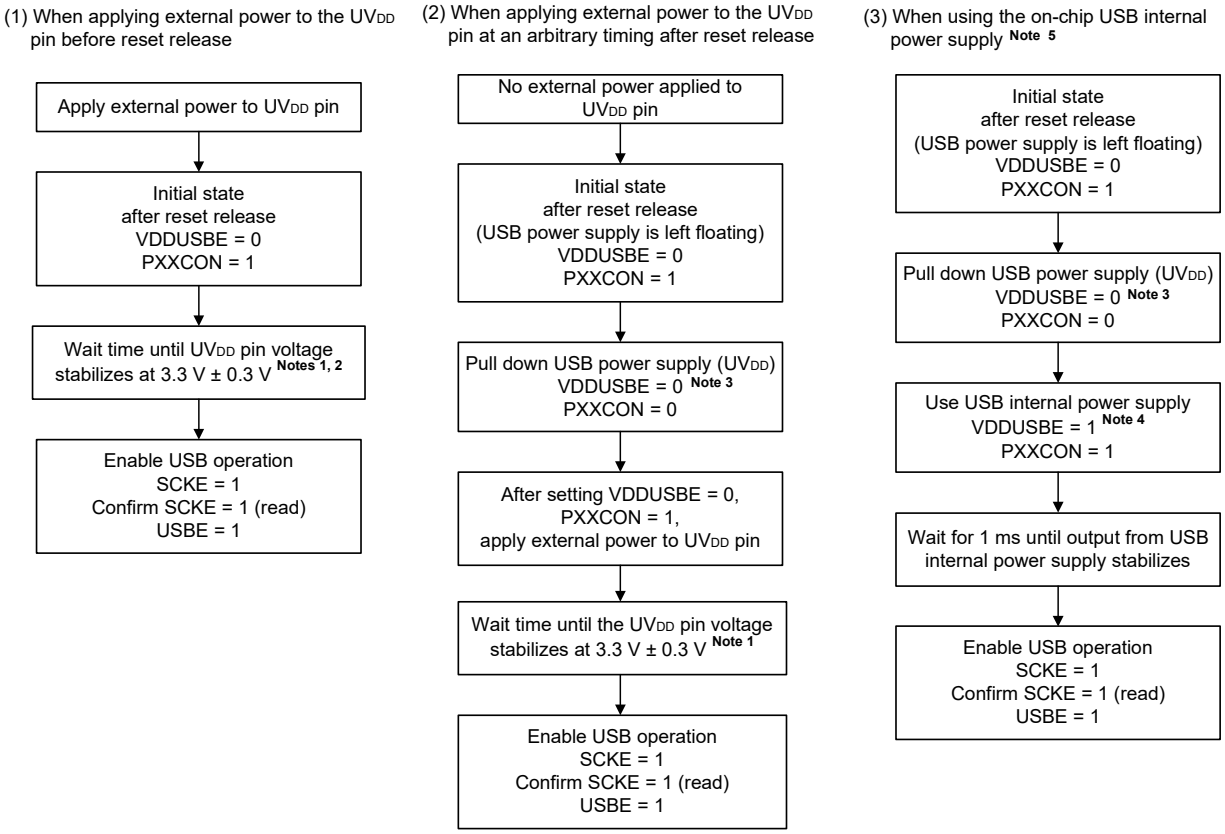
The source for the USB power supply (UV_{DD}) can be applied either externally from the UV_{DD} pin or from the on-chip USB internal power supply.

When no external power is applied to the UV_{DD} pin, the initial state after releasing the USB power supply reset is a floating state. To prevent erroneous operation caused by the floating state, pull down the USB power supply (UV_{DD}) by setting the register (VDDUSBE bit = 0, PXXCON bit = 0) until applying external power to the UV_{DD} pin or using the USB internal power supply.

When using the on-chip USB internal power supply, it is necessary to connect an external 0.33 µF stabilization capacitance (for V_{SS}) to the UV_{DD} pin. While the USB internal power supply is used, A/D conversion cannot be performed using the temperature sensor or internal reference voltage at the same time.

Figure 13 - 50 shows the flow for turning on the USB power when using the USB host/function controller. For details about the flow for turning on the USB power when using the USB host/function controller and the BC connection detection function (except the optional BC connection detection functions), see the application note.

Figure 13 - 50 Flow for Turning on USB Power



Note 1. The wait time depends on the characteristics of an external power supply to be used.

Note 2. No wait time is required if the UV_{DD} pin voltage stabilizes during the reset release sequence.

Note 3. To prevent malfunction caused by a floating USB power supply, be sure not to change this setting when power is not being applied externally from the UV_{DD} pin or when the internal USB power supply might be stop.

Note 4. When the USB internal power supply is used, A/D conversion cannot be performed using the temperature sensor or internal reference voltage.

Note 5. When using the USB internal power supply, connect an external 0.33 μF stabilization capacitance (for V_{SS}) to the UV_{DD} pin.

13.4.1.2 Controller function selection

For the USB module, the host controller function or function controller function can be selected using the DCFM bit in the SYSCFG register. The DCFM bit should be modified in the initial settings immediately after a power-on reset or in the D+ pull-up-disabled (DPRPU = 0) and D+ and D- pull-down-disabled (DRPD = 0) state.

Table 13 - 13 shows USB Port Function Selection.

Table 13 - 13 USB Port Function Selection

When the host controller function is selected		
USB port 0	USB port 1	Remark
Full or low	Full or low	The transfer scheduling is common to USB port 0 and USB port 1. The output is driven to both USB port 0 and USB port 1.
When the function controller function is selected		
USB port 0	USB port 1	Remark
Full or low	Not used	USB port 1 is invalid.

13.4.1.3 Controlling USB data bus resistors

The USB module contains pull-up and pull-down resistors of the D+ and D- lines. Pull up or pull down these lines by setting the DPRPU and DRPD bits in the SYSCFG register.

When the function controller function is selected, confirm that connection to the USB host is made, then set the DPRPU bit in the SYSCFG register to 1 and pull up the D+ (full speed)/D- (low speed) line.

When the DPRPU bit in the SYSCFG register is set to 1 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of connection.

When the host controller function is selected, set the DRPD bit (SYSCFG or SYSCFG1 register) for the used port to 1 and pull down the D+/D- lines.

Table 13 - 14 lists the settings for Controlling USB Data Bus Resistors of USB port 0 and Table 13 - 15 lists the settings for Controlling USB Data Bus Resistors of USB port 1.

Table 13 - 14 Controlling USB Data Bus Resistors of USB port 0

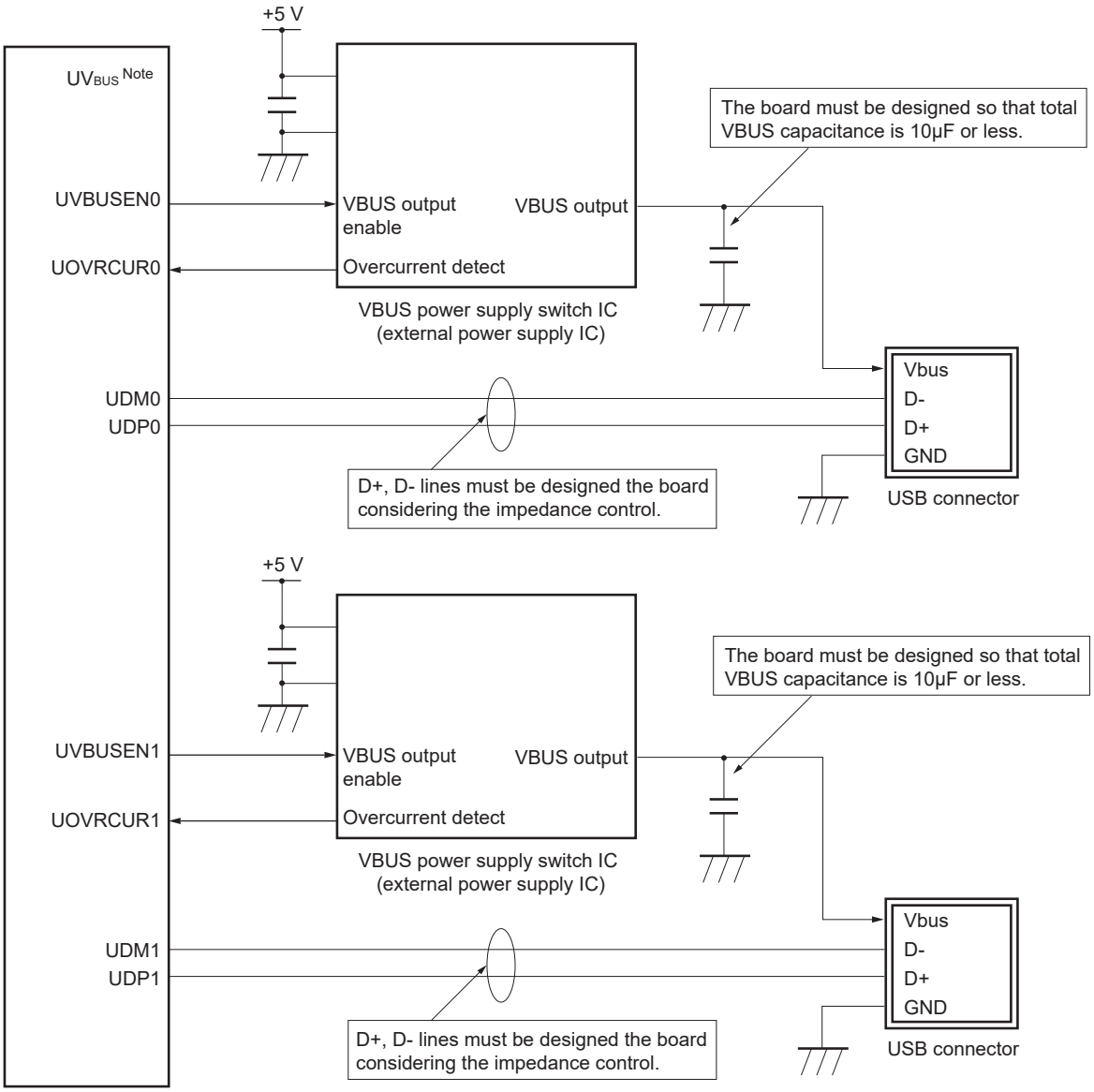
Settings			USB Data Bus Resistor Control		
DRPD	DPRPU	DMRPU	D- Line	D+ Line	Remarks
0	0	0	Open	Open	When USB port 0 is not used
0	1	0	Open	Pull-up	Set to this state when operating as the function controller (full speed).
0	0	1	Pull-up	Open	Set to this state when operating as the function controller (low speed).
1	0	0	Pull-down	Pull-down	Set to this state when operating as the host controller.
Other than above			—	—	Setting prohibited

Table 13 - 15 Controlling USB Data Bus Resistors of USB port 1

Settings	USB Data Bus Resistor Control		
DRPD	D- Line	D+ Line	Remarks
0	Open	Open	When USB port 1 is not used
1	Pull-Down	Pull-Down	Set to this state when operating as the function controller.

Figures 13 - 51 to 13 - 55 show the examples of USB external connection circuit.

Figure 13 - 51 USB Connector Host Connection Example



Note The UVBUS input pin is not used when the host controller function is selected.
The UVBUS pin is connected to an on-chip pulled-down resistor while the VBRPDCUT bit (bit 7 in the USBMC register) is in the default state (0), in order to prevent the pin level from becoming unstable when the pin is left open.
To fix the voltage of the UVBUS pin by applying an external voltage, set the VBRPDCUT bit to 1 and disconnect this on-chip resistor to prevent unnecessary current consumption.
When using the battery charging connection detection optional functions, the VBUS voltage to be connected to the USB connector must be applied to the UVBUS pin. In this case, set the VBRPDCUT bit to 1 and cut off the internal pull-down resistor of the UVBUS pin.

Figure 13 - 52 USB Connector Function Connection Example in Self-powered Mode (5 V)

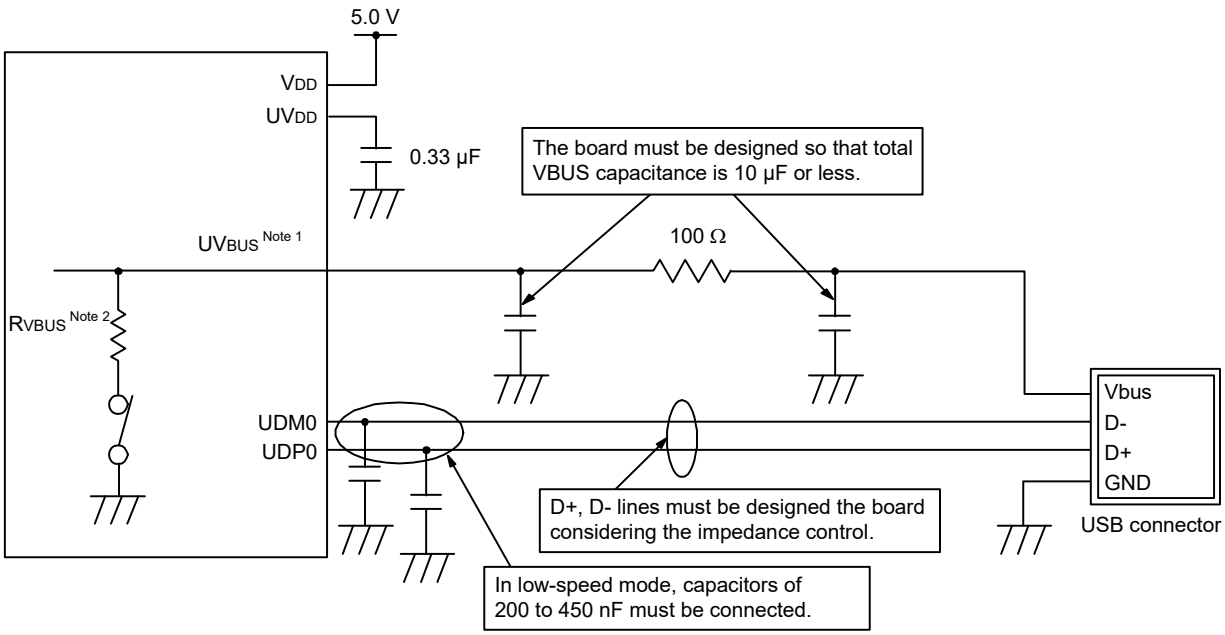
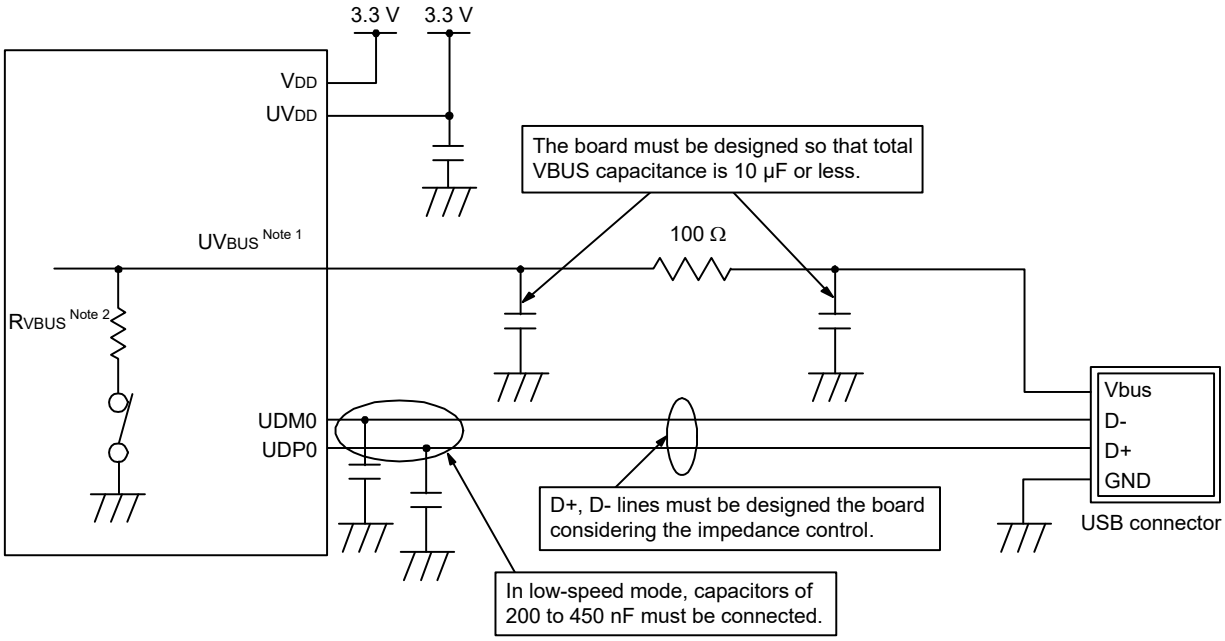


Figure 13 - 53 USB Connector Function Connection Example in Self-powered Mode (3.3 V)



Note 1. 5 V torelant

Note 2. Set the VBRPDCUT bit (bit 7 in the USBMC register) to 0 and connect the on-chip pull-down resistor of the UVBUS pin.

Figure 13 - 54 USB Connector Function Connection Example in Bus-powered Mode (5 V)

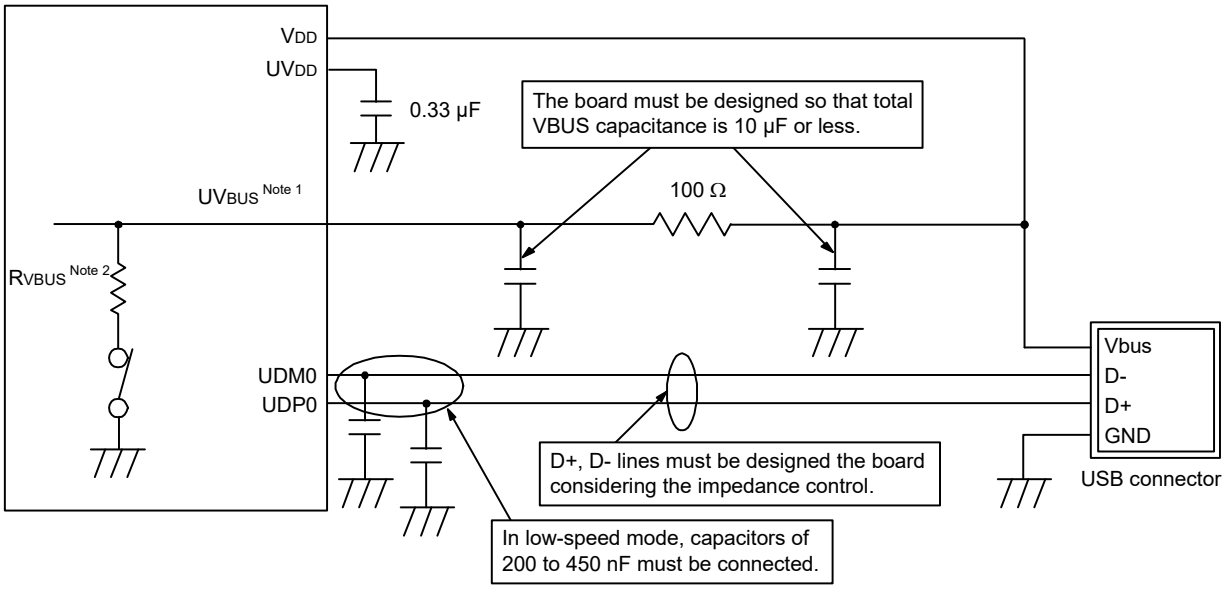
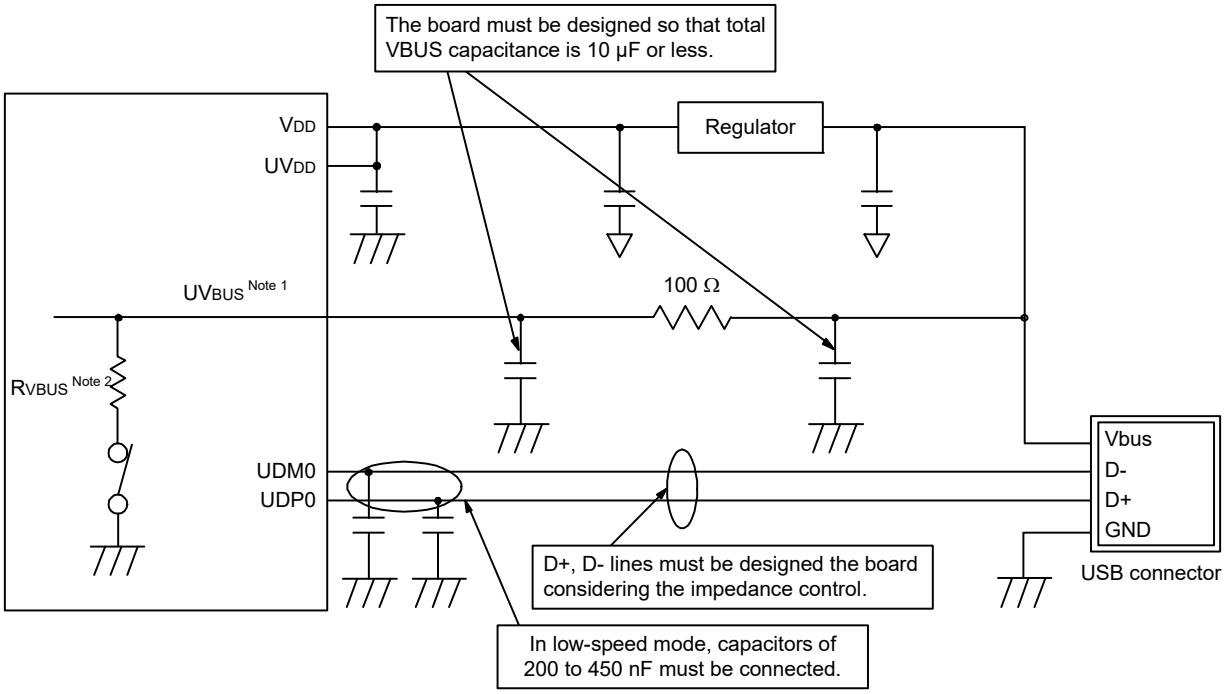


Figure 13 - 55 USB Connector Function Connection Example in Bus-powered Mode (3.3 V)



Note 1. 5 V torelant

Note 2. Set the VBRPDCUT bit (bit 7 in the USBMC register) to 0 and connect the on-chip pull-down resistor of the UVBUS pin.

13.4.2 Interrupt sources

Table 13 - 16 lists the Interrupt Sources in the USB module.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, the USB issues a USB interrupt request to the interrupt controller and an USB interrupt will be generated.

Table 13 - 16 Interrupt Sources (1/2)

Bit to be Set	Name	Interrupt Source	Function That Generates Interrupt	Status Flag
VBINT	VBUS interrupt	When a change in the state of the VBUS input pin has been detected (low to high or high to low)	Host/function Note	VBSTS
RESM	Resume interrupt	When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	[Host controller function is selected] • When an SOF packet with a different frame number has been transmitted [Function controller function is selected] • When an SOF packet with a different frame number has been received	Host/function	—
DVST	Device state transition interrupt	• When a device state transition has been detected A USB bus reset detected Suspend state detected SET_ADDRESS request received SET_CONFIGURATION request received	Function	DVSQ2 to DVSQ0
CTRT	Control transfer stage transition interrupt	• When a stage transition has been detected in control transfer Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	Function	CTSQ2 to CTSQ0
BEMP	Buffer empty interrupt	• When transmission of all data in the buffer memory has been completed and the buffer has become empty • When a packet larger than the maximum packet size has been received	Host/function	PIPEnBEMP in BEMPSTS register
NRDY	Buffer not ready interrupt	[Host controller function is selected] • When STALL has been received from the peripheral device for the issued token • When a response has not been received correctly from the peripheral device for the issued token (no response was returned three consecutive times or a packet reception error occurred three consecutive times) [Function controller function is selected] • When NAK has been returned for an IN or OUT token	Host/function	PIPEnNRDY in NRDYSTS register
BRDY	Buffer ready interrupt	• When the buffer has become ready for read access or write access.	Host/function	PIPEBRDY in BRDYSTS register
OVRCR	Overcurrent input change interrupt	• When a change in the state of the UOVRCUR0 (USB port 0) or UOVRCUR1 (USB port 1) input pin has been detected (low to high or high to low)	Host	OVCMON
BCHG	Bus change interrupt	• When a change of USB bus state has been detected	Host/function	LNST
DTCH	USB disconnection detection interrupt	• When disconnection of the USB device has been detected	Host	RHST
ATTCH	Connection interrupt	• When J-state or K-state is detected on the USB port for 2.5 µs. Used for checking whether a peripheral device is connected.	Host	—
EOFERR	EOF error detection interrupt	• When an EOF error of a peripheral device has been detected	Host	—
SACK	Normal response interrupt for setup transaction	• When the normal response (ACK) for the setup transaction has been received	Host	—

Table 13 - 16 Interrupt Sources (2/2)

Bit to be Set	Name	Interrupt Source	Function That Generates Interrupt	Status Flag
SIGN	Error interrupt for setup transaction	• When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—
PDDETINT	Portable Device detection interrupt	• When connection of the Portable Device has been detected	Host	PDDETSTS

Note Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.

Figure 13 - 56 shows the USB interrupt relationship, Table 13 - 17 lists the USB interrupts.

Figure 13 - 56 USB Interrupts Relationship

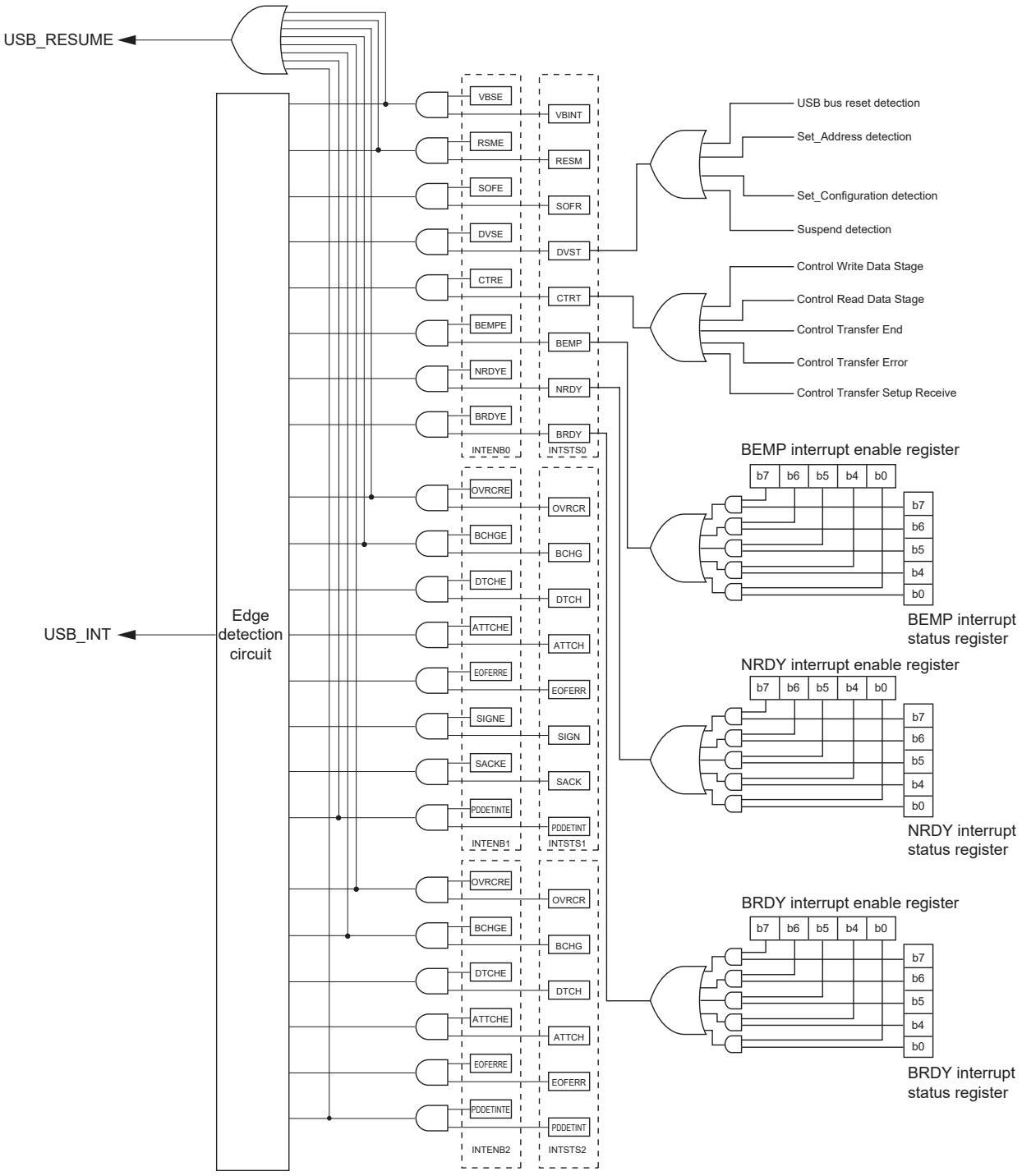


Table 13 - 17 USB Interrupt List

Interrupt Name	Interrupt Flag	Priority
USB_INT	VBUS interrupt Resume interrupt Frame number update interrupt Device state transition interrupt Control transfer stage transition interrupt Buffer empty interrupt Buffer not ready interrupt Buffer ready interrupt Overcurrent input change interrupt Bus change interrupt USB disconnection detection interrupt Connection interrupt EOF error detection interrupt Normal response interrupt for setup transaction Error interrupt for setup transaction Portable Device detection interrupt	High
USB_RESUME	VBUS interrupt Resume interrupt Overcurrent input change interrupt Bus change interrupt Portable Device detection interrupt	Low

13.4.3 Interrupts

13.4.3.1 BRDY interrupt

The BRDY interrupt is generated when either of the host controller function or function controller function is selected. The following shows the conditions under which the USB module sets 1 to a corresponding bit in the BRDYSTS register. Under this condition, the USB module generates a BRDY interrupt if software has set 1 to the PIPEBRDYE bit in the BRDYENB register that corresponds to the pipe and 1 to the BRDYE bit in the INTENB0 register.
The conditions for generating and clearing the BRDY interrupt depend on the settings of the BRDYM bit and BFRE bit for each pipe as described below.

(1) When BRDYM = 0 and BFRE = 0
With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
On any of the following conditions, the USB module generates an internal BRDY interrupt request trigger and sets 1 to the PIPEBRDY bit corresponding to the pertinent pipe.

[For the pipe set to the transmitting direction]
- When software changes the DIR bit from 0 to 1.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When 1 is written to the ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

[For the pipe set to the receiving direction]
- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).
  No request trigger is generated for the transaction in which DATA-PID disagreement has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
  No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

When the function controller function is selected, the BRDY interrupt is not generated in the status stage of control transfers.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY bit in BRDYSTS through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.
Be sure to clear the BRDY status before accessing the FIFO buffer.

(2) When BRDYM bit = 0 and the BFRE bit = 1
With these settings, the USB module generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in BRDYSTS corresponding to the pertinent pipe.

On any of the following conditions, the USB module determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the transaction counter register (TRNCNT bits) is used and the number of packets specified by the TRNCNT bits are completely received.

When the pertinent data is completely read out after any of the above conditions has been satisfied, the USB module determines that all data for a single transfer has been completely read out.
When a zero-length packet is received while the FIFO buffer is empty, the USB module determines that all data for a single transfer has been completely read out upon passing the zero-length packet data to the CPU. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFOCTR register through software.
With these settings, the USB module does not detect a BRDY interrupt for the pipe in the transmitting direction.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY bit through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.
In this mode, the BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the ACLRM bit.

(3) When BRDYM = 1 and BFRE = 0

With these settings, the PIPEBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by the USB module depending on the FIFO buffer status.

[For the pipe set to the transmitting direction]

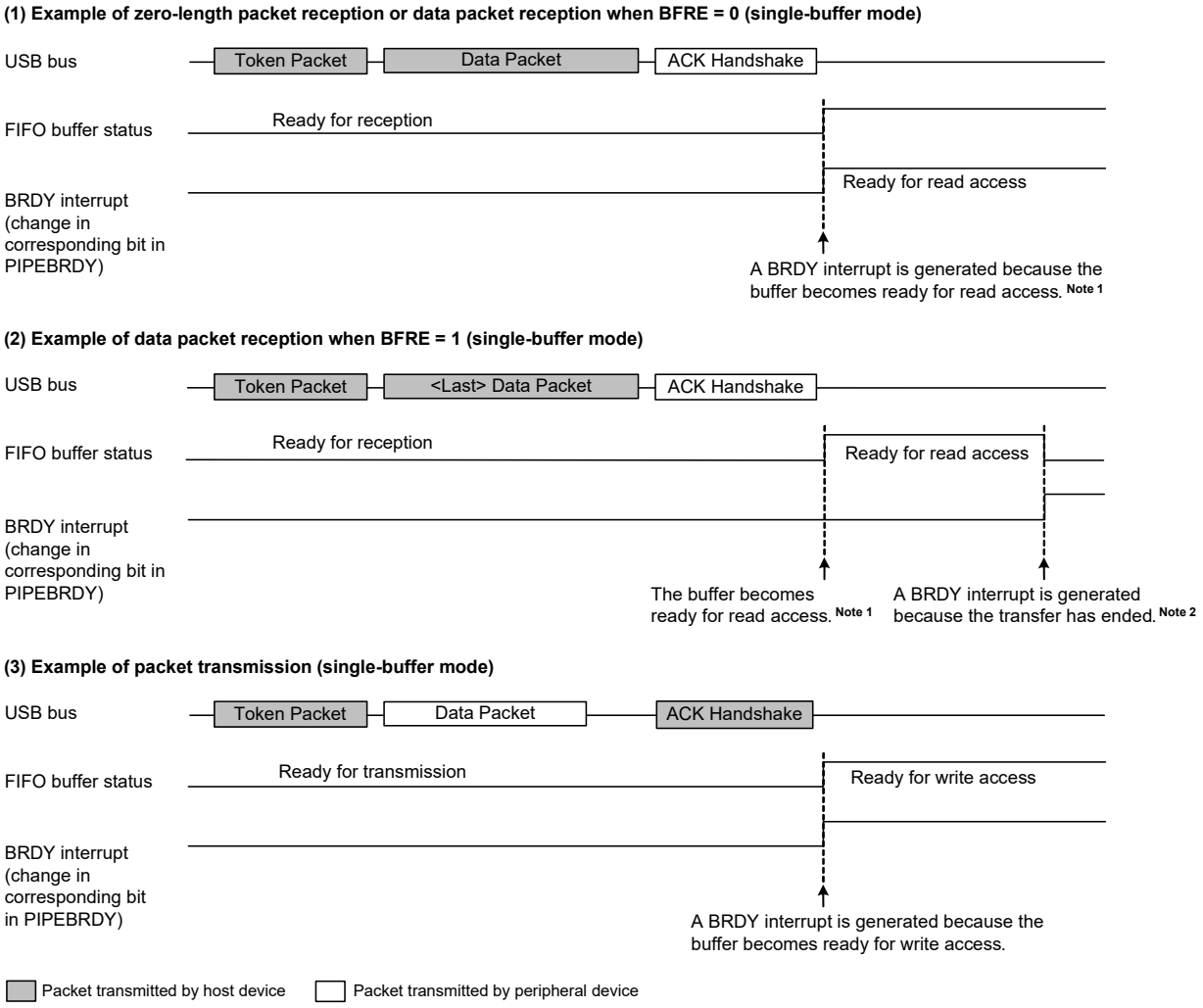
The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.
However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

[For the pipe set to the receiving direction]

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).
When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.
With this setting, the PIPEBRDY bit cannot be cleared to 0 through software.
When the BRDYM bit is set to 1, all BFRE bits (for all pipes) should be cleared to 0.

Figure 13 - 57 shows the Timing of BRDY Interrupt Generation.

Figure 13 - 57 Timing of BRDY Interrupt Generation



Note 1. The FIFO buffer becomes ready for read access under the following condition:
When a packet is received while no data remains unread in the buffer in the CPU.

Note 2. A transfer ends under either of the following conditions:
(1) When a short packet including a zero-length packet is received
(2) When the number of packets specified in the transaction counter are received

The condition that USB module clears the BRDY bit in INTSTS0 depends on the setting of the BRDYM bit in the SOFCFG register. Table 13 - 18 shows the Condition for Clearing BRDY Bit.

Table 13 - 18 Condition for Clearing BRDY Bit

BRDYM	Condition for Clearing BRDY Bit
0	The USB module clears the BRDY bit in the INTSTS0 register when software has cleared all bits in the BRDYSTS register.
1	The USB module clears the BRDY bit in the INTSTS0 register when the BSTS bits for all piles have become 0.

13.4.3.2 NRDY interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USB module sets the corresponding PIPEnNRDY bit in the NRDYSTS register to 1. If the corresponding bit in the NRDYENB register has been set to 1 by software, the USB module sets the NRDY bit in the INTSTS0 register to 1 and generates a USB interrupt.
The following describes the conditions on which the USB module generates the internal NRDY interrupt request for a given pipe.
Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller function is selected. During setup transactions when the host controller function is selected, the SACK or SIGN interrupt is detected.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller function is selected.

(1) When the host controller function is selected

On any of the following conditions, the USB module detects an NRDY interrupt.

[For the pipe set to the transmitting direction]

- During communications other than setup transactions, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB module sets the corresponding PIPEnNRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK.
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.
  In this case, the USB module sets the corresponding PIPEnNRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to STALL (11B).

[For the pipe set to the receiving direction]

- When any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USB module (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB module sets the PIPEnNRDY bit corresponding to the pipe to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK.
- When the STALL handshake is received.
  In this case, the USB module sets the PIPEnNRDY bit corresponding to the pipe to 1 and modifies the setting of the PID bits of the corresponding pipe to STALL (11B).

(2) When the function controller function is selected

On any of the following conditions, the USB module detects an NRDY interrupt.

[For the pipe set to the transmitting direction]

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.

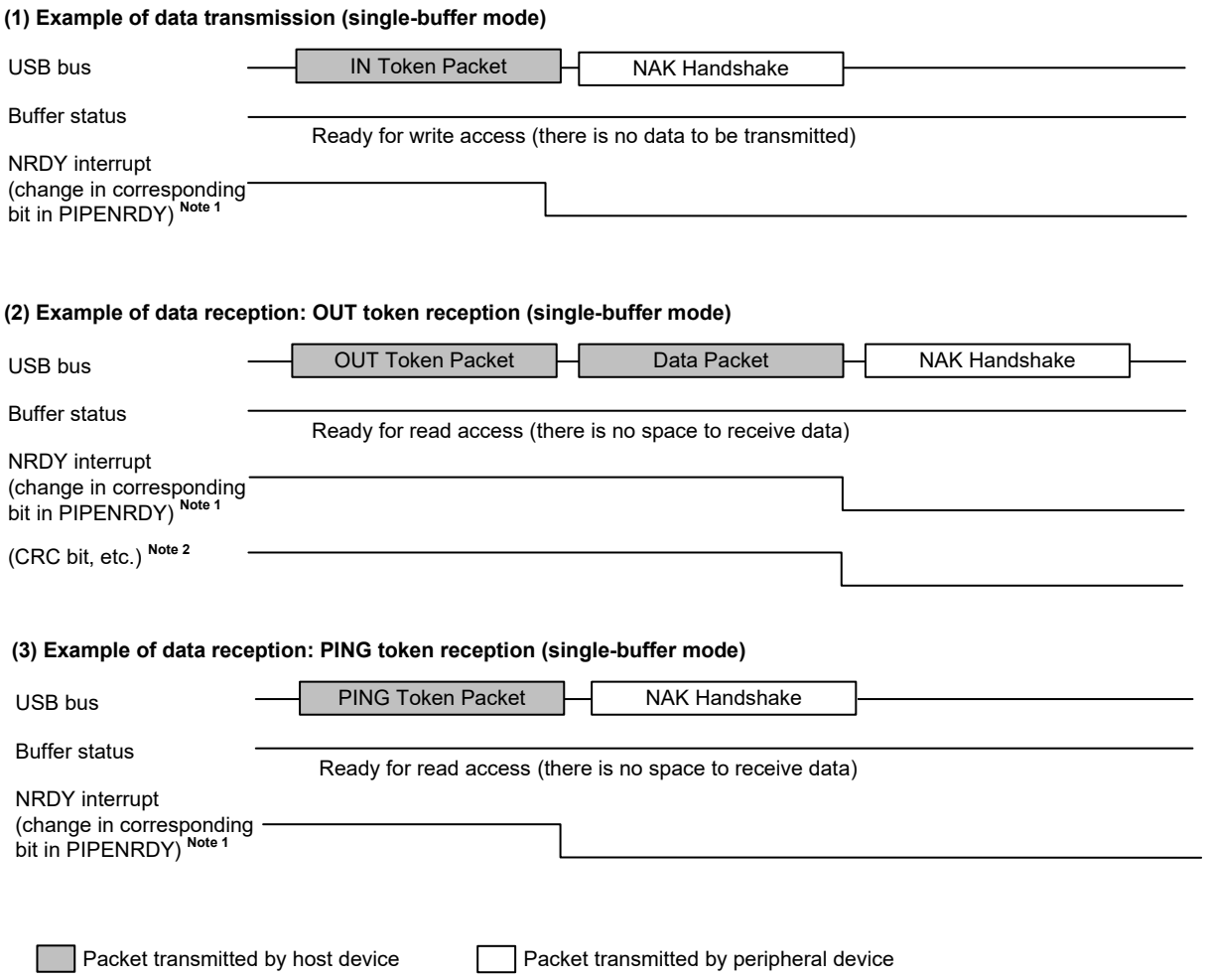
In this case, the USB module generates a NRDY interrupt request at the reception of the IN token and sets the PIPEnNRDY bit to 1.

[For the pipe set to the receiving direction]

- When an OUT token is received while there is no space available in the FIFO buffer.
  The USB module generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY bit to 1.
  However, during re-transmission (due to DATA-PID disagreement), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.

Figure 13 - 58 shows the Timing of NRDY Interrupt Generation When Function Controller Function is Selected.

Figure 13 - 58 Timing of NRDY Interrupt Generation When Function Controller Function is Selected



Note 1. The PIPEnNRDY bit is set to 1 only while the PID bits for the target pipe are set to 1.

Note 2. The CRC and OVRN bits change only while the target pipe is set to isochronous transfers.

13.4.3.3 BEMP interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USB module sets the corresponding PIPEnBEMP bit in the BEMPSTS register to 1. If the corresponding bit in the BEMPENB register has been set to 1 by software, the USB module sets the BEMP bit in the INTSTS0 register to 1 and generates a USB interrupt.

The following describes the conditions on which the USB module generates an internal BEMP interrupt request.

[For the pipe set to the transmitting direction]

- When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
  In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP.
  However, the internal BEMP interrupt request is not generated on any of the following conditions.
- When software has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the ACLRM or BCLR bit to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage while the function controller function is selected.

[For the pipe set to the receiving direction]

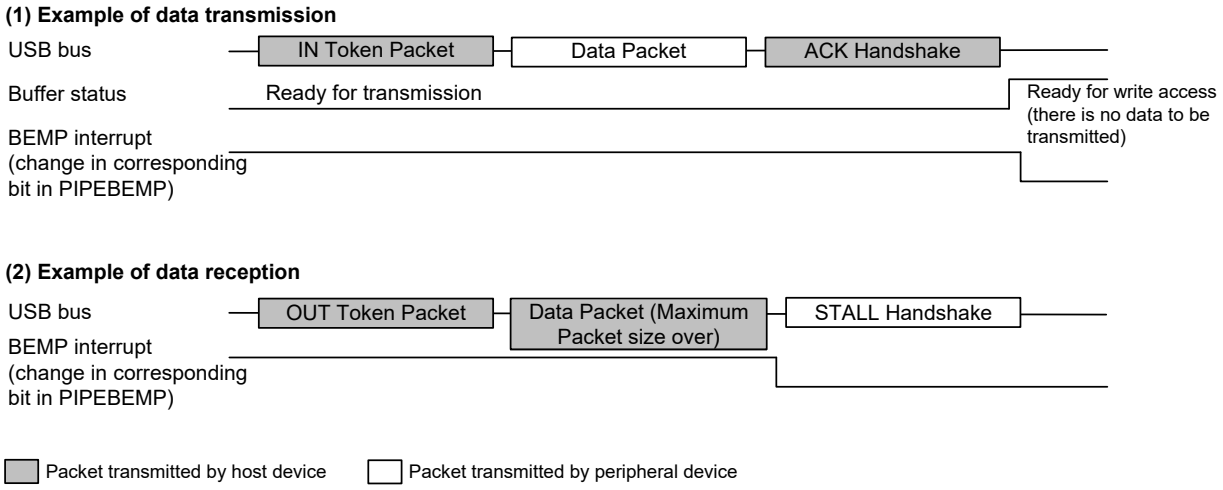
When data whose size is greater than the setting value of MaxPacketSize is successfully received.
In this case, the USB module generates a BEMP interrupt request, sets the corresponding PIPEnBEMP bit in the BEMPSTS register to 1, discards the received data, and modifies the setting of the PID bits of the corresponding pipe to STALL (11B).
Here, the USB module returns no response when used as the host controller, and returns STALL response when used as the function controller.
However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed.

Writing 0 to the PIPEnBEMP bit in the BEMPSTS register clears the status.
Writing 1 to the PIPEnBEMP bit in the BEMPSTS register has no effect.

Figure 13 - 59 shows the Timing of BEMP Interrupt Generation When Function Controller Function is Selected.

Figure 13 - 59 Timing of BEMP Interrupt Generation When Function Controller Function is Selected



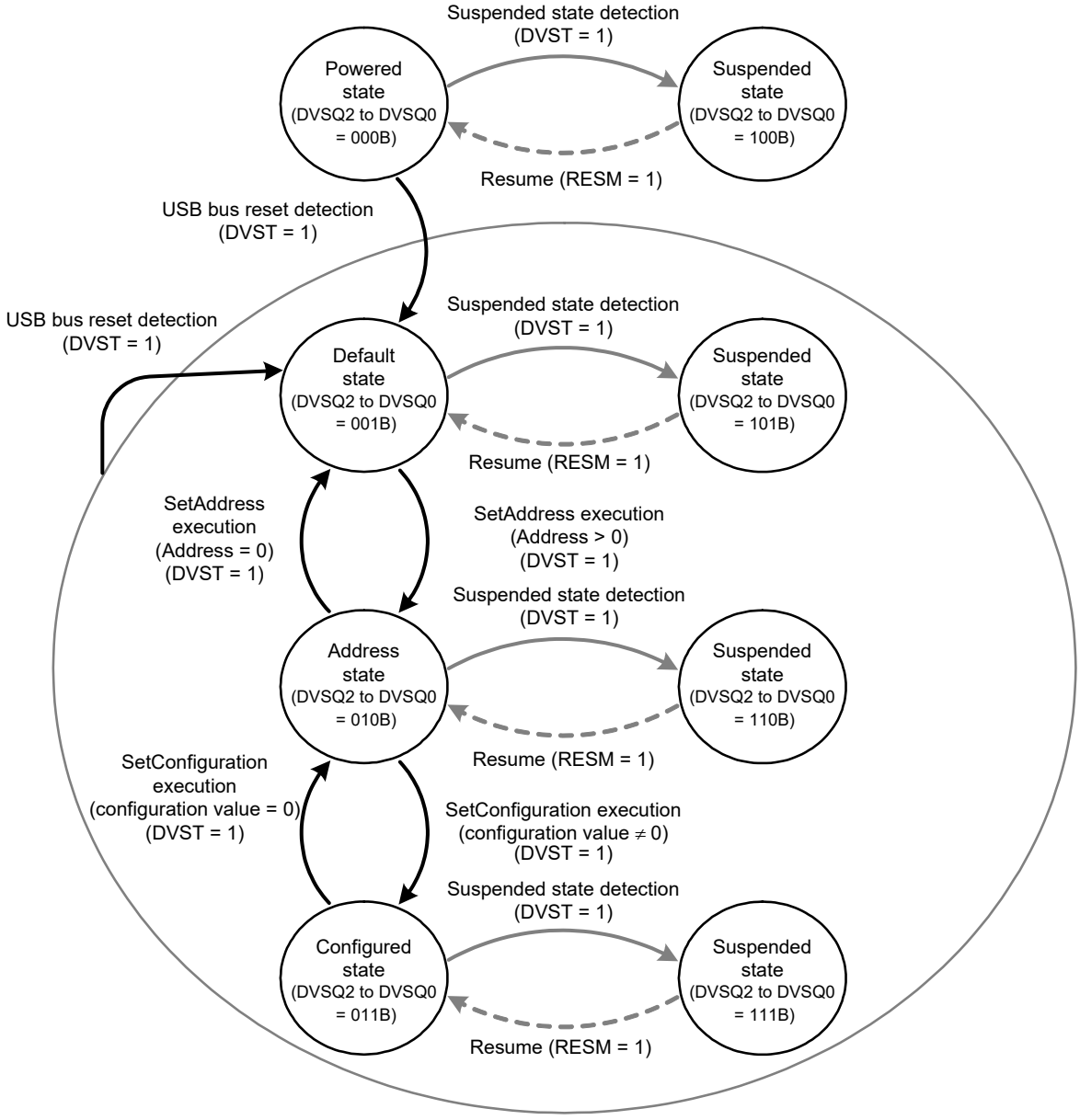
13.4.3.4 Device state transition interrupt

Figure 13 - 60 is a diagram of Device State Transitions in the USB module. The USB module controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using the INTENB0 register. The device state to which a transition was made can be confirmed using the DVSQ2 to DVSQ0 bits in the INTSTS0 register.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller function is selected. The device state transition interrupts can also be generated only when the function controller function is selected.

Figure 13 - 60 Device State Transitions



Caution For the transition indicated in solid line, the DVST bit is set to 1.
For the transition indicated in dashed line, the RESM bit is set to 1.

13.4.3.5 Control transfer stage transition interrupt

Figure 13 - 61 is a diagram of Control Transfer Stage Transitions in the USB module. The USB module controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using the INTENB0 register. The transfer stage to which a transition was made can be confirmed using the CTSQ2 to CTSQ0 bits in the INTSTS0 register. Control transfer stage transition interrupts are generated only when the function controller function is selected. The control transfer sequence errors are listed below. If an error occurs, the PID bits in the DCPCTR register are set to 1xB (STALL response).

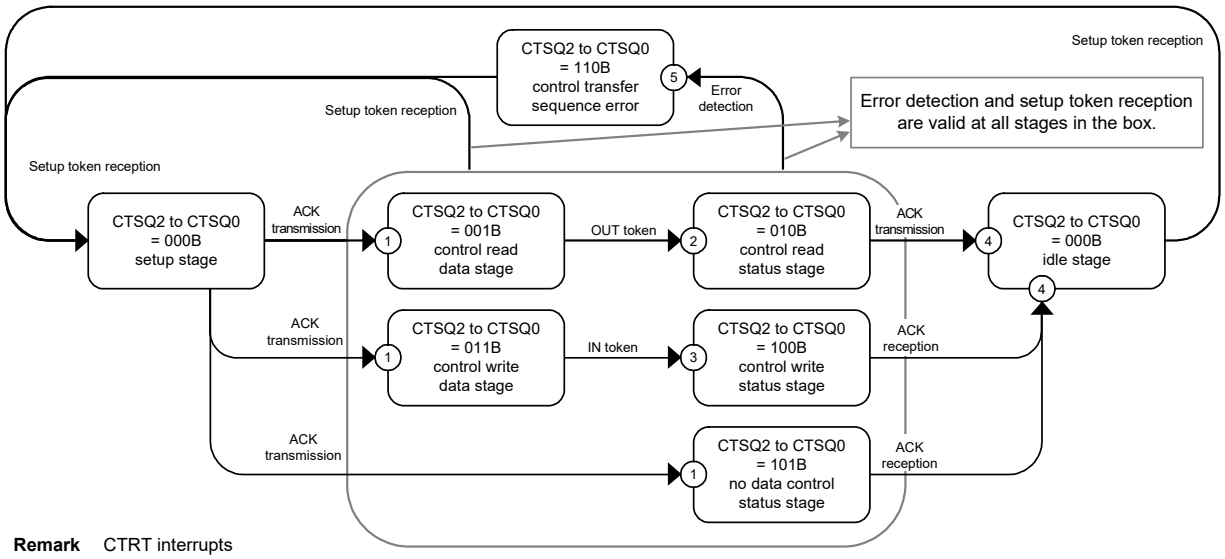
(1) During control read transfer
• An OUT token is received while no data has been transferred for the IN token at the data stage.
• An IN token is received at the status stage.
• A data packet with DATAPID = DATA0 is received at the status stage.

(2) During control write transfer
• An IN token is received while no ACK response has been returned for the OUT token at the data stage.
• A data packet with DATAPID = DATA0 is received for the first data packet at the data stage.
• An OUT token is received at the status stage.

(3) During no-data control transfers
• An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally. When a CTRT interrupt occurs in response to a sequence error (SERR = 1), the CTSQ2 to CTSQ0 bits = 110B value is retained until CTRT = 0 is written from the system (the interrupt status is cleared). Therefore, while the CTSQ2 to CTSQ0 bits = 110B is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB module retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

Figure 13 - 61 Control Transfer Stage Transitions



Remark CTRT interrupts
(1) Setup stage completed
(2) Control read transfer status stage transition
(3) Control write transfer status stage transition
(4) Control transfer completed
(5) Control transfer sequence error

13.4.3.6 Frame update interrupt

With the host controller function selected, an interrupt is generated at the timing when the frame number is updated. With the function controller function selected, an SOFR interrupt is generated when the frame number is updated.

When the function controller function is selected, the USB module updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

13.4.3.7 VBUS interrupt

When the VBUS pin level changes, a VBUS interrupt is generated. The level of the VBUS pin can be checked with the VBSTS bit in the INTSTS0 register. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the VBUS pin level.

13.4.3.8 Resume interrupt

When the function controller function is selected, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.

When the host controller function is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

13.4.3.9 Overcurrent interrupt

For USB port 0, an overcurrent interrupt (OVRCR interrupt) is generated when the UOVRCUR0 pin level has changed. The level of the UOVRCUR0 pin can be checked with the OVCMON1 and OVCMON0 bits in the SYSSTS0 register. The external power-supply IC can check whether overcurrent has been detected using the OVRCR interrupt.

For USB port 1, an OVRCR interrupt is generated when the UOVRCUR1 pin level has changed. The level of the UOVRCUR1 pin can be checked with the OVCMON1 and OVCMON0 bits in the SYSSTS1 register. The external power-supply IC can check whether overcurrent has been detected using the OVRCR interrupt.

13.4.3.10 Bus change interrupt

A bus change interrupt (BCHG interrupt) is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup when the host controller function is selected. The BCHG interrupt is generated regardless of whether the host controller function or function controller function is selected.

13.4.3.11 USB disconnection detection interrupt

A USB disconnection detection interrupt (DTCH interrupt) is generated when disconnection of the USB bus is detected while the host controller function is selected. The USB module detects bus disconnection based on USB Specification 2.0.

After detecting a DTCH interrupt, the USB module controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). Software should terminate all pipes in which communications are currently carried out for the pertinent port and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

13.4.3.12 Setup transaction normal response interrupt

A setup transaction normal response interrupt (SACK interrupt) is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller function selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

13.4.3.13 Setup transaction error interrupt

A setup transaction error interrupt (SIGN interrupt) is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller function selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

13.4.3.14 Connection interrupt

A connection interrupt (ATTCH interrupt) is generated when J-state or K-state of the full-speed/low-speed signal level is detected on the USB port for 2.5 µs with the host controller function selected. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

13.4.3.15 EOF error interrupt

An EOF error interrupt (EOFERR interrupt) is generated when it is detected that communication is not completed at the EOF2 timing prescribed in USB Specification 2.0.

After detecting an EOFERR interrupt, the USB module controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). Software should terminate all pipes in which communications are currently carried out for the pertinent port and perform re-enumeration of the pertinent port.

- Modifies the UACT bit in the DVSTCTR0 register for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

13.4.3.16 Portable device detection interrupt

A Portable Device detection interrupt is generated when the USB module detects a level change (high from low or low from high) in the PDDET output from the USB-PHY. When a Portable Device detection interrupt is generated, use software to repeat reading the PDDETSTS bit until the same value is read three or more times, and eliminate chattering.

13.4.4 Pipe control

Table 13 - 19 lists the Pipe Settings in the USB module. With USB data transfer, data transfer has to be carried out using the logic pipe called the endpoint. The USB module has five pipes that are used for data transfer. Appropriate settings should be made for each of the pipes according to the specifications of the system.

Table 13 - 19 Pipe Settings

Register Name	Bit Name	Setting	Remarks
DCPCFG PIPECFG	TYPE	Specifies the transfer type	PIPE4, PIPE5: Can be set
	BFRE	Selects the BRDY interrupt mode	PIPE4, PIPE5: Can be set
	DBLB	Selects double buffer mode	PIPE4, PIPE5: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM	Endpoint number	PIPE4, PIPE5: Can be set A value other than 0000B should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	PIPE4, PIPE5: Can be set
DCPMAXP PIPEMAXP	DEVSEL	Selects a device	Referenced only when the host controller function is selected.
	MXPS	Maximum packet size	Compliant with the USB standard.
PIPEPERI	IFIS	Buffer flush	PIPE4 to PIPE7: Cannot be set
	IITV	Interval counter	PIPE4, PIPE5: Cannot be set PIPE6, PIPE7: Can be set (only when the host controller function has been selected)
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for PIPE4, PIPE5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	SUREQCLR	SUREQ request clear	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	ATREPM	Auto response mode	PIPE4, PIPE5: Can be set Can be set only when the function controller function has been selected.
	ACLRM	Auto buffer clear	PIPE4 to PIPE7: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit.
	SQSET	Sequence set	Sets the data toggle bit.
	SQMON	Sequence monitor	Monitors the data toggle bit.
	PBUSY	Pipe busy status	Monitors if the pipe is busy.
	PID	Response PID	Refer to **13.4.4.6 Response PID**.
PIPEnTRE	TRENB	Transaction counter enable	PIPE4, PIPE5: Can be set
	TRCLR	Current transaction counter clear	PIPE4, PIPE5: Can be set
PIPEnTRN	TRNCNT	Transaction counter	PIPE4, PIPE5: Can be set

13.4.4.1 Pipe control register switching procedures

The following bits in the pipe control registers can be modified only when USB communication is disabled (PID = NAK).

Registers that Should Not be Set in the USB Communication Enabled (PID = BUF) State:
- Bits in the DCPCFG and DCPMAXP registers
- The SQCLR and SQSET bits in the DCPCTR registers
- Bits in the PIPECFG, PIPEMAXP, and PIPEPERI registers
- The ATREPM, ACLRM, SQCLR, and SQSET bits in the PIPEnCTR register
- Bits in the PIPEnTRE and PIPEnTRN registers

In order to modify the above bits in the USB communication enabled (PID = BUF) state, follow the procedure shown below:
(1) A request to modify bits in the pipe control register occurs.
(2) Modify the PID corresponding to the pipe to NAK.
(3) Wait until the corresponding PBUSY bit is cleared to 0.
(4) Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE bit in the CFIFOSEL, D0FIFOSEL, and D1FIFOSE registers.

Registers that Should Not be Set When CURPIPE in FIFO-PORT is set:
- Bits in the DCPCFG and DCPMAXP registers
- Bits in the PIPECFG, PIPEMAXP and PIPEPERI registers

In order to modify pipe information, the CURPIPE bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using BCLR after the pipe information is modified.

13.4.4.2 Transfer types

The TYPE bits in the PIPEPCFG register are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.
- DCP: No setting is necessary (fixed at control transfer).
- PIPE4, PIPE5: These should be set to bulk transfer.
- PIPE6, PIPE7: These should be set to interrupt transfer.

13.4.4.3 Endpoint number

The EPNUM bits in the PIPEPCFG register are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.
- DCP: No setting is necessary (fixed at end point 0).
- PIPE4 to PIPE7: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the DIR bit and EPNUM bits is unique.

13.4.4.4 Maximum packet size setting

The MXPS bits in the DCPMAXP and PIPEMAXP registers are used to specify the maximum packet size for each pipe. DCP and PIPE4, PIPE5 can be set to any of the maximum pipe sizes defined by the USB

specification. For PIPE6, PIPE7, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

• DCP: Set 8, 16, 32, or 64.
• PIPE4, PIPE5: Set 8, 16, 32, or 64 when using bulk transfer.
• PIPE6, PIPE7: Set a value between 1 and 64.

13.4.4.5 Transaction counter (for PIPE4, PIPE5 in reading direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB module recognizes that the transfer has ended. Two transaction counters are provided: one is the TRNCNT register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the PID of the corresponding PIPE is set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the TRCLR bit. The information read from TRNCNT differs depending on the setting of the TRENB bit.

• TRENB = 0:The specified transaction counter value can be read.
• TRENB = 1:The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

• If the transactions are being counted and PID = BUF, the current counter cannot be cleared.
• If there is any data left in the buffer, the current counter cannot be cleared.

13.4.4.6 Response PID

The PID bits in DCPCTR and PIPEnCTR are used to set the response PID for each pipe.
The following shows the USB module operation with various response PID settings:

(1) Response PID settings when the host controller function is selected

The response PID is used to specify the execution of transactions.

• NAK setting: Using pipes is disabled. No transaction is executed.
• BUF setting: Transactions are executed based on the status of the buffer memory.
For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
• STALL setting: Using pipes is disabled. No transaction is executed.

Caution Setup transactions for the DCP are set with the SUREQ bit.

(2) Response PID settings when the function controller function is selected

The response PID is used to specify the response to transactions from the host.

• NAK setting: The NAK response is always returned in response to the generated transaction.
• BUF setting: Responses are made to transactions according to the status of the buffer memory.
• STALL setting: The STALL response is always returned in response to the generated transaction.

Caution For setup transactions, an ACK response is always returned regardless of the PID setting, and the USB request is stored in the register.

The USB module may write to the PID bits, depending on the results of the transaction as described below.

(3) When the host controller function has been selected and the response PID is set by hardware NAK setting:

In the following cases, PID = NAK is set and issuing of tokens is automatically stopped:

• When an NRDY interrupt is generated.(For details, refer to **13.4.3.2 NRDY interrupt**.)
• If a short packet is received when the SHTNAK bit in the PIPECFG register has been set to 1 for bulk transfer.
• If the transaction counting ends when the SHTNAK bit has been set to 1 for bulk transfer.
BUF setting: There is no BUF writing by the USB module.
STALL setting: In the following cases, PID = STALL is set and issuing of tokens is automatically stopped:
• When STALL is received in response to the transmitted token.
• When the size of the receive data packet exceeds the maximum packet size.

(4) When the function controller function has been selected and the response PID is set by hardware
NAK setting: In the following cases, PID = NAK is set and NAK is always returned in response to transactions:
• When the SETUP token is received normally (DCP only).
• If the transaction counting ends or a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.
BUF setting: There is no BUF writing by the USB module.
STALL setting: In the following cases, PID = STALL is set and STALL is always returned in response to transactions:
• When a maximum packet size exceeded error is detected in the received data packet.
• When a control transfer sequence error has been detected (DCP only).

13.4.4.7 Data PID sequence bit

The USB module automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in the DCPCTR and PIPEnCTR registers. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The SQCLR bit in the DCPCTR register and the SQSET bit in the PIPEnCTR register can be used to change the data PID sequence bit.
When the function controller function has been selected and control transfer is used, the USB module automatically sets the sequence bit when a stage transition is made. DATA0 is returned when the setup stage is ended and DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller function has been selected and control transfer is used, the sequence bit should be set by software at a stage transition.
For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software regardless of whether the host controller function or function controller function is selected.

13.4.4.8 Response PID = NAK function

The USB module has a function that disables pipe operation (PID response = NAK) at the timing at which the final data packet of a transaction is received (the USB module automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the SHTNAK bit in the PIPECFG register to 1. When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (PID response = BUF).
The response PID = NAK function can be used only when bulk transfers are used.

13.4.4.9 Auto response mode

With the pipes for bulk transfer (PIPE4, PIPE5), when the ATREPM bit in the PIPEnCTR register is set to 1, a transition is made to auto response mode. During an OUT transfer (DIR = 0), OUT-NAK mode is entered, and during an IN transfer (DIR = 1), null auto response mode is entered.

13.4.4.10 OUT-NAK mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is retuned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

13.4.4.11 Null auto response mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, the INBUFM bit = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, the buffer should be emptied with the ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10 µs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

13.4.5 FIFO buffer memory

13.4.5.1 FIFO buffer memory

The USB module has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB module. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB module (SIE side).

(1) Buffer Status

Tables 13 - 20 and 13 - 21 show the buffer status in the USB module. The buffer memory status can be confirmed using the BSTS bit in the DCPCTR register and the INBUFM bit in the PIPEnCTR register. The access direction for the buffer memory can be specified using either the DIR bit in the PIPEnCFG register or the ISEL bit in the CFIFOSEL register (when DCP is selected).

The INBUFM bit is valid for PIPE4, PIPE5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS bit to monitor the buffer memory status on the CPU side and the INBUFM bit to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU is slow, software can use the INBUFM bit to confirm the end of transmission.

Table 13 - 20 Buffer Status Indicated by BSTS Bit

ISEL or DIR	BSTS	Buffer Memory Status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission has been completed. CPU write is allowed.

Table 13 - 21 Buffer Status Indicated by INBUFM Bit

DIR	INBUFM	Buffer Memory Status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been completed. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

(2) FIFO Buffer Clearing

Tables 13 - 22 shows the clearing of the FIFO buffer memory by the USB module. The buffer memory can be cleared using the BCLR, DCLRM, and ACLRM bits.

Table 13 - 22 Buffer Clearing Methods

FIFO Buffer Clearing Mode	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(3) Auto Buffer Clear Mode Function

With the USB module, all received data packets are discarded if the ACLRM bit in the PIPEnCTR register is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.

If the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.

However, an access cycle of at least 100 ns is required for the internal hardware sequence processing time between the ACLRM bit = 1 and the ACLRM bit = 0.

(4) Buffer Memory Specifications (Single or Double Setting)

Either a single or double buffer configuration can be selected for PIPE4 and PIPE5, using the DBLB bit in the PIPEnCFG register.

13.4.5.2 FIFO port functions

Tables 13 - 23 shows the FIFO Port Function Settings for the USB module. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the CFIFOCTR or DnFIFOCTR register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.

In read access, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (DTLN = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN bits in the CFIFOCTR or DnFIFOCTR register.

Table 13 - 23 FIFO Port Function Settings

Register Name	Bit Name	Function	Remark
CFIFOSEL DnFIFOSEL	RCNT	Selects DTLN read mode.	
	REW	Buffer memory rewind (re-read, rewrite).	
	DCLRM	Automatically clears receive data for a specified pipe after the data has been read.	Only for DnFIFO.
	MBW	Selects the FIFO port access bit width.	
	BIGEND	Selects FIFO port endian.	
	ISEL	FIFO port access direction.	Only for DCP.
	CURPIPE	Selects the current pipe.	
CFIFOCTR DnFIFOCTR	BVAL	Ends writing to the buffer memory.	
	BCLR	Clears the buffer memory on the CPU side.	
	DTLN	Checks the length of receive data.	

(1) FIFO Port Selection

Table 13 - 24 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE bits in the CFIFOSEL or DnFIFOCTR register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FIFO port can be accessed after FRDY = 1 is checked.

In addition, the bus width to be accessed should be selected using the MBW bit. The buffer memory access direction conforms to the DIR bit in the PIPEnCFG register. Only for the DCP, the ISEL bit determines the direction.

Table 13 - 24 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register (CFIFOM)
PIPE4 to PIPE7	CPU access	CFIFO port register (CFIFOM) D0FIFO/D1FIFO port register (D0FIFOM/D1FIFOM)

(2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the CFIFOSEL or DnFIFOSEL register is used for this processing.

If a pipe is selected through the CURPIPE bits in the CFIFOSEL or DnFIFOSEL register with the REW bit set to 1, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, FRDY = 1 should be checked after selecting a pipe.

13.4.6 Control transfers (DCP)

In the data stage of control transfers, data is transferred using the default control pipe (DCP).
The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

13.4.6.1 Control transfers when host controller function is selected

(1) Setup Stage

USQREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the SUREQ and bit in the DCPCTR register transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is cleared to 0. The above USB request registers should not be modified while SUREQ = 1.
After the attached state of the connected function device is detected, the first setup transaction for the device should be issued by using the sequence described above with the DEVSEL bits in the DCPMAXP register set to 0 and the USBSPD and RTPORT bits in the DEVADD0 register set appropriately.
After the connected function device is shifted to the Address state, setup transactions should be issued by using the sequence described above with the assigned USB address set in the DEVSEL bits and the bits in the DEVADDn register corresponding to the specified USB address set appropriately. For example, when the DEVSEL bit in the PIPEMAXP register = 2H, make appropriate settings in the DEVADD2 register; when the DEVSEL bit = 5H, make appropriate settings in the DEVADD5 register.
When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN1 or SACK bit in the INTSTS1 register), by means of which the result of the setup transactions can be confirmed.
A data packet of DATA0 (USB request) is transmitted as the data packet for a setup transaction regardless of the setting of the SQMON bit in the DCPCTR register.

(2) Data Stage

Data is transferred using the DCP buffer memory.
The access direction of the DCP buffer memory should be specified using the ISEL bit in the CFIFOSEL register. The transfer direction should be specified using the DIR bit in the DCPCFG register.
For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the SQSET bit and the PID bits = BUF in the DCPCFG register. Completion of data transfer is detected using the BRDY or BEMP interrupt.
For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, software should control so as to send a zero-length packet at the end.

(3) Status Stage

Zero-length packet data is transferred in the direction opposite to that in the data stage. As in the data stage, data is transferred using the DCP buffer memory. Transactions are done in the same manner as the data stage.
For the data packets of the status stage, the data PID should be set to DATA1 using the SQSET bit in the DCPCFG register.
For reception of a zero-length packet, the received data length should be confirmed using the DTLN bits in the CFIFOCTR register after a BRDY interrupt is generated, and the buffer memory should then be cleared using the BCLR bit.

13.4.6.2 Control transfers when function controller function is selected

(1) Setup Stage

The USB module always sends an ACK response for a correct setup packet targeted to the USB module. The operation of the USB module in the setup stage is described below.

(a) When receiving a new setup packet, the USB module sets the following bits.
- Set the VALID bit in the INTSTS0 register to 1.
- Set the PID bits in the DCPCTR register to NAK.
- Set the CCPL bit in the DCPCTR register to 0.

(b) When receiving a data packet right after the setup packet, the USB module stores the USB request parameters in the USBREQ, USBVAL, USBINDX, and USBLENG registers.

Response processing with respect to the control transfer should always be carried out after setting VALID = 0. In VALID = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.
Using the function of the VALID bit, the USB module can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.
In addition, the USB module automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For control on the stages of the USB module, refer to Figure 13 - 61.

(2) Data Stage

Data transfers corresponding to received USB requests should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the ISEL bit in the CFIFOSEL register. If the transfer data is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status Stage

Control transfers are terminated by setting the CCPL bit to 1 while the PID bits in the DCPCTR register are set to BUF.
After the above settings have been made, the USB module automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

[For control read transfers]
The USB module receives a zero-length packet and transmits an ACK response.

[For control write transfers and no-data control transfers]
The USB module transmits a zero-length packet and receives an ACK response from the USB host.

(4) Control Transfer Auto Response Function

The USB module automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.
- Any transfer other than a control read transfer: bmRequestType ≠ 00H
- Request error: wIndex ≠ 00H
- Any transfer other than a no-data control transfer: wLength ≠ 00H
- Request error: wValue > 7FH
- Control transfer of a device state error: DVSQ2 to DVSQ2 bits = 011B (Configured)

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

13.4.7 Bulk transfers (PIPE4, PIPE5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (BFRE bit: refer to **13.4.3.1 BRDY interrupt**.)
- Transaction count function
  (TRENB, TRCLR, and TRNCNT bits: refer to **13.4.4.5 Transaction counter (for PIPE4, PIPE5 in reading direction)**
- Response PID = NAK function (SHTNAK bit: refer to **13.4.4.8 Response PID = NAK function**)
- Auto response mode (ATREPM bit: refer to **13.4.4.9 Auto response mode**)

13.4.8 Interrupt transfers (PIPE6, PIPE7)

When the function controller function is selected, the USB module carries out interrupt transfers in accordance with the timing controlled by the host controller.
When the host controller function is selected, the timing of issuing a token can be specified using the interval counter.

13.4.8.1 Interval counter during interrupt transfers when host controller function is selected

For interrupt transfers, intervals between transactions are set in the IITV bits in PIPEPERI. The USB controller issues interrupt transfer tokens based on the specified intervals.

(1) Counter Initialization

The USB controller initializes the interval counter under the following conditions.

- Power-on reset:

The IITV bits are initialized.

- Buffer memory initialization using the ACLRM bit:

The IITV bits are not initialized but the count value is initialized. Setting the ACLRM bit in the PIPEnCTR register to 0 starts counting from the value set in the IITV bits.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended

The IITV bits are not initialized. Setting 1 to the UACT bit in the DVSTCTR0 register starts counting from the value before entering the USB bus reset state or USB suspended state.

(2) Operation when Transmission/Reception is Impossible at Token Issuance Timing

The USB module cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB module attempts transactions at the subsequent interval.

- When the PID is set to NAK or STALL.
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the transmitting (OUT) direction.

13.4.9 SOF interpolation function

When the function controller function is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB module interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in the SYSCFG register have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB module supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing

If an SOF packet is missing, the FRNM bit in the FRMNUM register is not updated.

13.4.10 Pipe schedule

13.4.10.1 Conditions for generating a transaction

When the host controller function is selected and the UACT bit has been set to 1, the USB module generates a transaction under the conditions shown in Table 13 - 25.

Table 13 - 25 Conditions for Generating a Transaction

Transaction	Conditions for Generation				
	DIR	PID	IITV0	Buffer State	SUREQ
Setup	- Note	- Note	- Note	- Note	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	- Note
	OUT	BUF	Invalid	Transmit data exists	- Note
Interrupt transfer	IN	BUF	Valid	Receive area exists	- Note
	OUT	BUF	Valid	Transmit data exists	- Note

Note Symbols (-) in the table indicate that the condition is unrelated to the generating of tokens. “Valid” indicates that, for interrupt transfers, a transaction is generated only in transfer frames that are based on the interval counter. “Invalid” indicates that a transaction is generated regardless of the interval counter.

13.4.10.2 Transfer schedule

This section describes the transfer scheduling within a frame of the USB module. After the USB module sends an SOF, the transfer is carried out in the sequence described below.

(1) Execution of periodic transfers

A pipe is searched in the order of PIPE6→PIPE7, and then, if there is a pipe for interrupt transfer transaction can be generated, the transaction is generated.

(2) Setup transactions for control transfers

The DCP is checked, and if a setup transaction is possible, it is sent.

(3) Execution of bulk transfers, control transfer data stages, and control transfer status stages

A pipe is searched in the order of DCP→PIPE4→PIPE5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.

When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

13.4.10.3 Enabling USB communication

Setting the UACT bit in the DVSTCTR register to 1 initiates SOF transmission and transaction generation is enabled.

Setting the UACT bit to 0 stops SOF transmission and a suspend state is entered. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF is sent.

13.4.11 Controlling battery charging detection

It is possible to control the processing for data contact detection (D+ line contact check), primary detection (charger detection), and secondary detection (charger verification) in compliance with Battery Charging Specification Revision 1.2.
By executing these processes according to charger detection algorithms, it is possible to determine whether the connected device is a standard downstream port, charging downstream port, or dedicated charging port (function BC connection detection function; USB ports 0 and 1) as a portable device. It is also possible to operate as a charging downstream port or dedicated charging port for a portable device (host BC connection detection function; USB ports 0 and 1).

The above data contact detection, primary detection, and secondary detection are each detected by the interface detection circuit for BC connection detection that is provided along with the USB transceiver. This circuit has a function to connect necessary voltage sources (V_{DP_SRC}, V_{DM_SRC}), current source (I_{DP_SRC}), voltage (V_{DAT_REF}) detection function, and UDP and UDM pins via a resistor (R_{DCP_DAT}) to perform connection detection compliant to Battery Charging Specification Revision 1.2. These can be controlled and monitored by setting the bits in BC control register n (USBBCCTRLn) (n = 0, 1).
When using the BC connection detection function, specify PXXCON = 0 and VDDUSBE = 1 to externally apply power to the UV_{DD} pin, or specify PXXCON = 1 and VDDUSBE = 1 (internal USB power supply is used) to internally generate 3.3 V.
The temperature sensor cannot be used and A/D conversion that uses the internal reference voltage cannot be performed when using the BC connection detection function.
Figures 13 - 62 and Figures 13 - 63 show the interface circuit for BC connection detection.

Figure 13 - 62 BC Connection Detection Interface Circuit (USB Port 0) with Host/Function BC Connection Detection Function

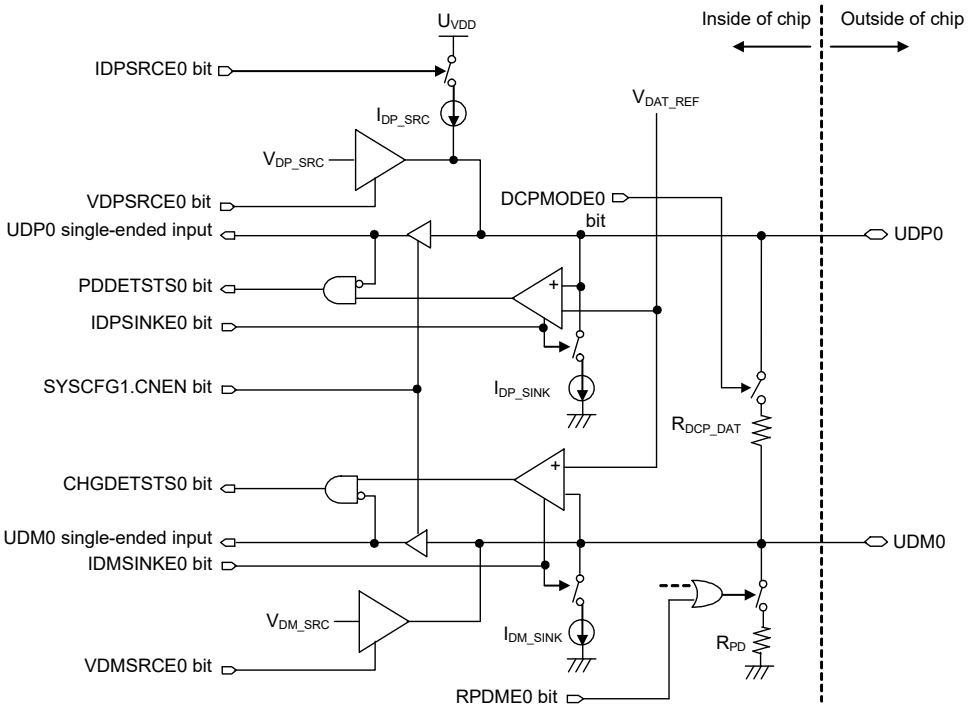
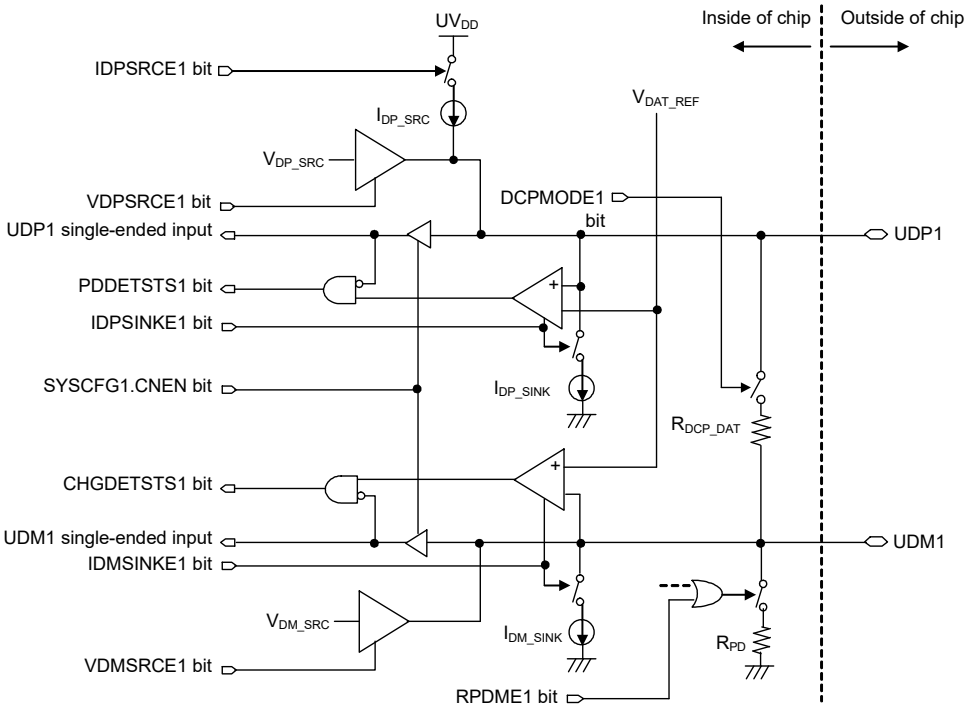


Figure 13 - 63 BC Connection Detection Interface Circuit (USB Port1) with Host/Function BC Connection Detection Function



13.4.12 Battery charging connection detection optional functions

For extensibility of the battery charging specifications, the following optional functions are added to control connection detection.

• USB port voltage output function (four patterns)
As an optional function of the host BC connection detection function, this function can divide 5 V applied to the UV_{BUS} pin and output to the USB port. Furthermore, this function can detect whether the voltage of the USB port has risen and dropped while the divided voltage is output. It can also detect whether the signal has conflicted with the connected USB port output of the device.

• USB port voltage detection function (16 stages)
As an optional function of the function BC connection detection function, this function can detect the voltage level to be input to the USB port, using the 16-stage reference voltage that is obtained by dividing 5 V to be applied to the UV_{BUS} pin.

After power is supplied to the UV_{DD} pin (using an external supply or internal power supply for the USB) and the voltage is applied to the UV_{BUS} pin, these functions can control various functions and detection results by setting BC option control register n (USBBCOPTn) (n = 0, 1).

Figure 13 - 64 and Figure 13 - 65 show the Interface Circuit for BC Connection Detection Optional Functions.

Figure 13 - 64 BC Connection Detection Optional Function Interface Circuit (USB Port 0) with Host/Function BC Connection Detection Function

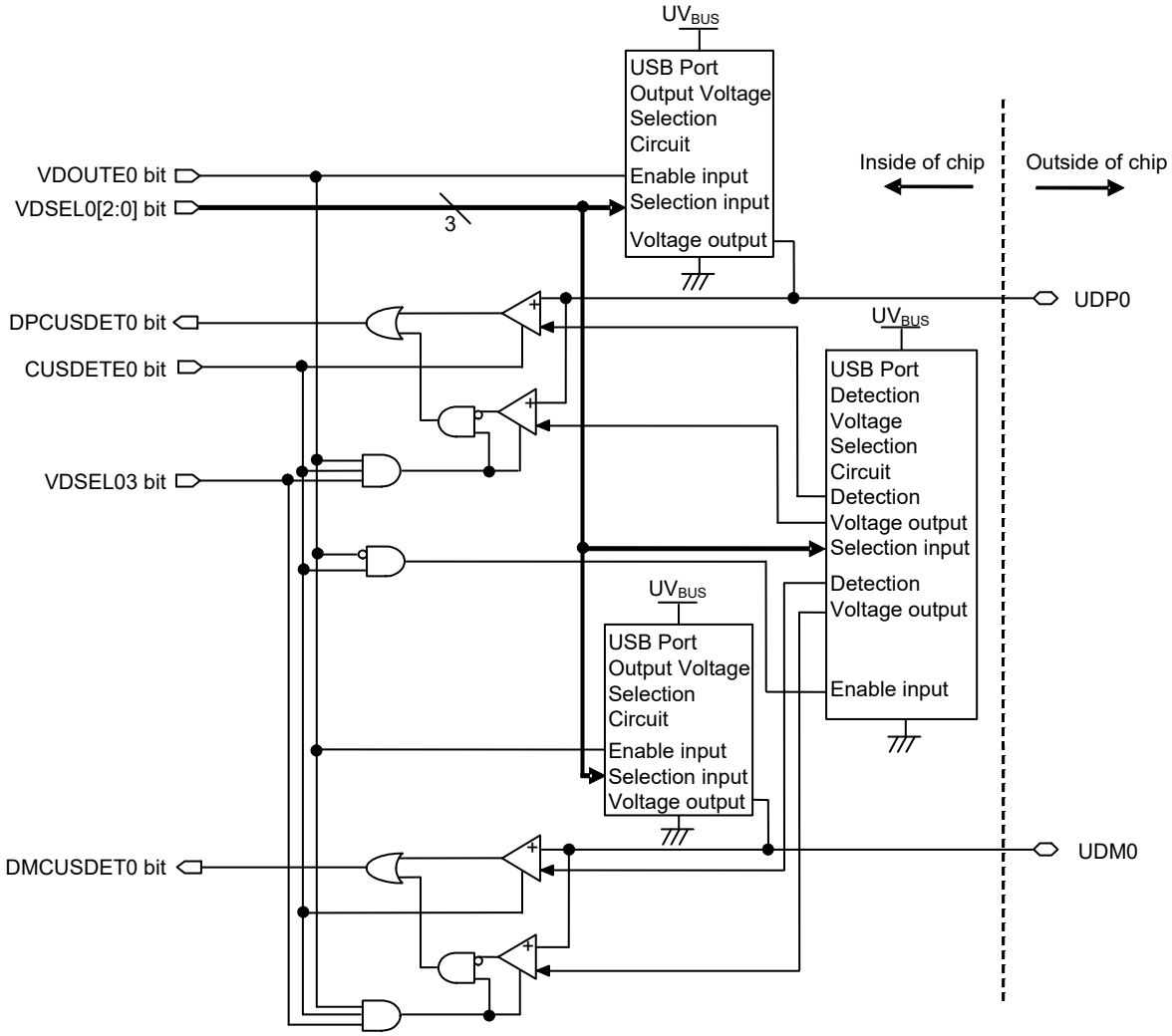
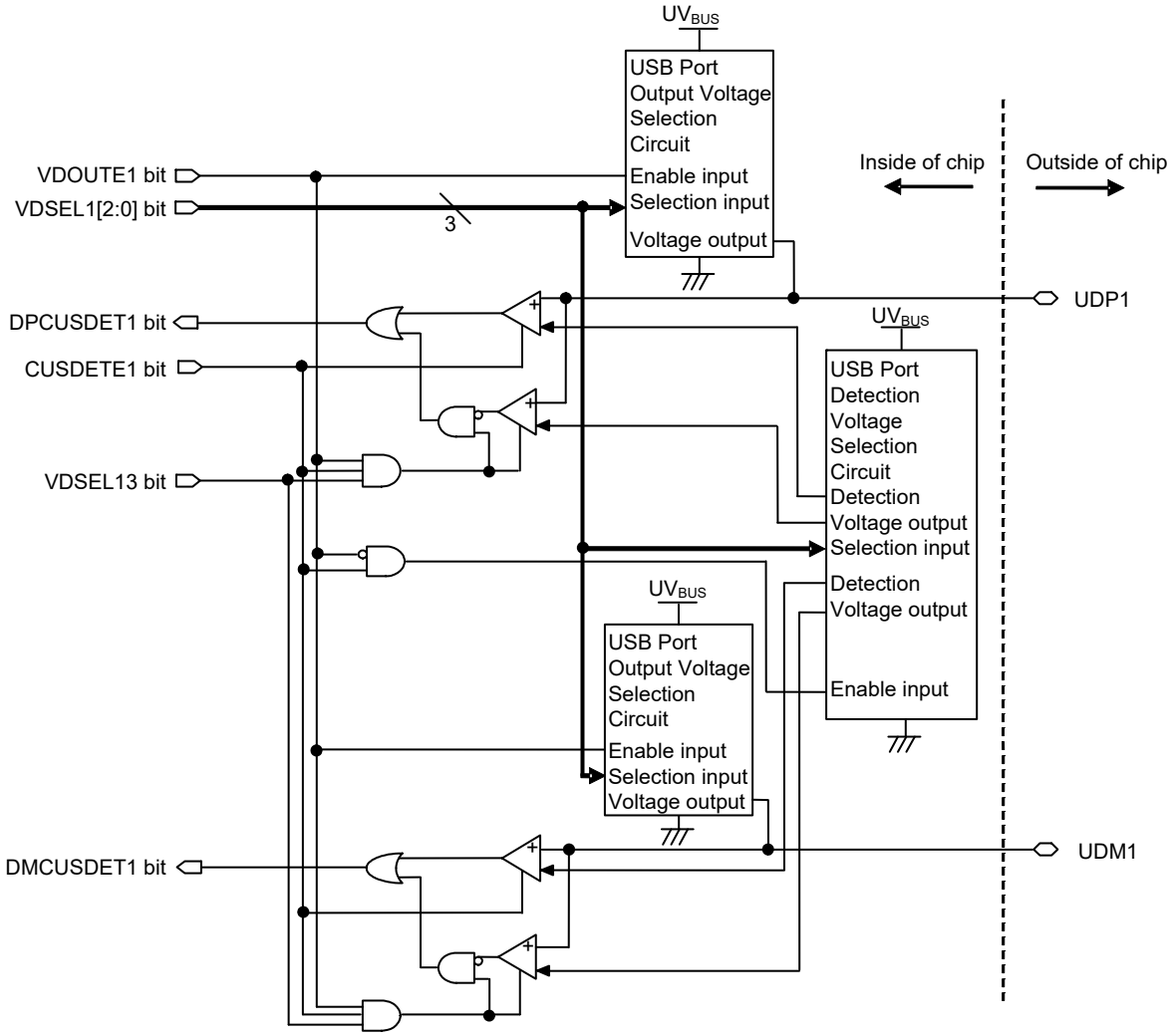


Figure 13 - 65 BC Connection Detection Optional Function Interface Circuit (USB Port 1) with Host/Function BC Connection Detection Function



13.4.13 Battery charging detection processing

It is possible to control to the processing for Data Contact Detection (D+ line contact check), Primary Detection (Charger detection), and Secondary Detection (Charger verification), which are defined by the Battery Charging Specification.
The following shows required operations for a Peripheral Device and a Host Device, individually.

13.4.13.1 Processing when function controller is selected

The following processing is required when operating the USB module as a Portable Device for Battery Charging.

(1) Detect when the data lines (D+/D-) have made contact and start the processing for Primary Detection.
(2) After Primary Detection starts, wait 40 ms for masking, and then check the D- voltage level to confirm the Primary Detection result.
(3) If the Charger is detected during Primary Detection, also start Secondary Detection.
(4) After Secondary Detection starts, wait 40 ms for masking, and then check the D+ voltage level to confirm the Secondary Detection result.

For the above step (1), after VBUS is detected using the VBINT interrupt and the VBSTS bit, wait for 300 ms to 900 ms by software, and then set the VDPSRCE and IDMSINKE bits in the USBBCCTRL register. Or set the IDPSRCE bit, and after a change from high to low on the D+ line is detected using the LNST bits, clear the IDPSRCE bit and set the VDPSRCE and IDMSINKE bits. The VDPSRCE and IDMSINKE bits must be set at the same time. Note 1

For the above step (2), set the VDPSRCE and IDMSINKE bits and wait for 40 ms by software, and then use the CHGDETSTS bit to verify the Primary Detection result. Note 2

For the above step (3), verify that the Charger is detected if the CHGDETSTS bit is set in the above step (2), and then clear the VDPSRCE and IDMSINKE bits and set the VDMSRCE and IDPSINKE bits.

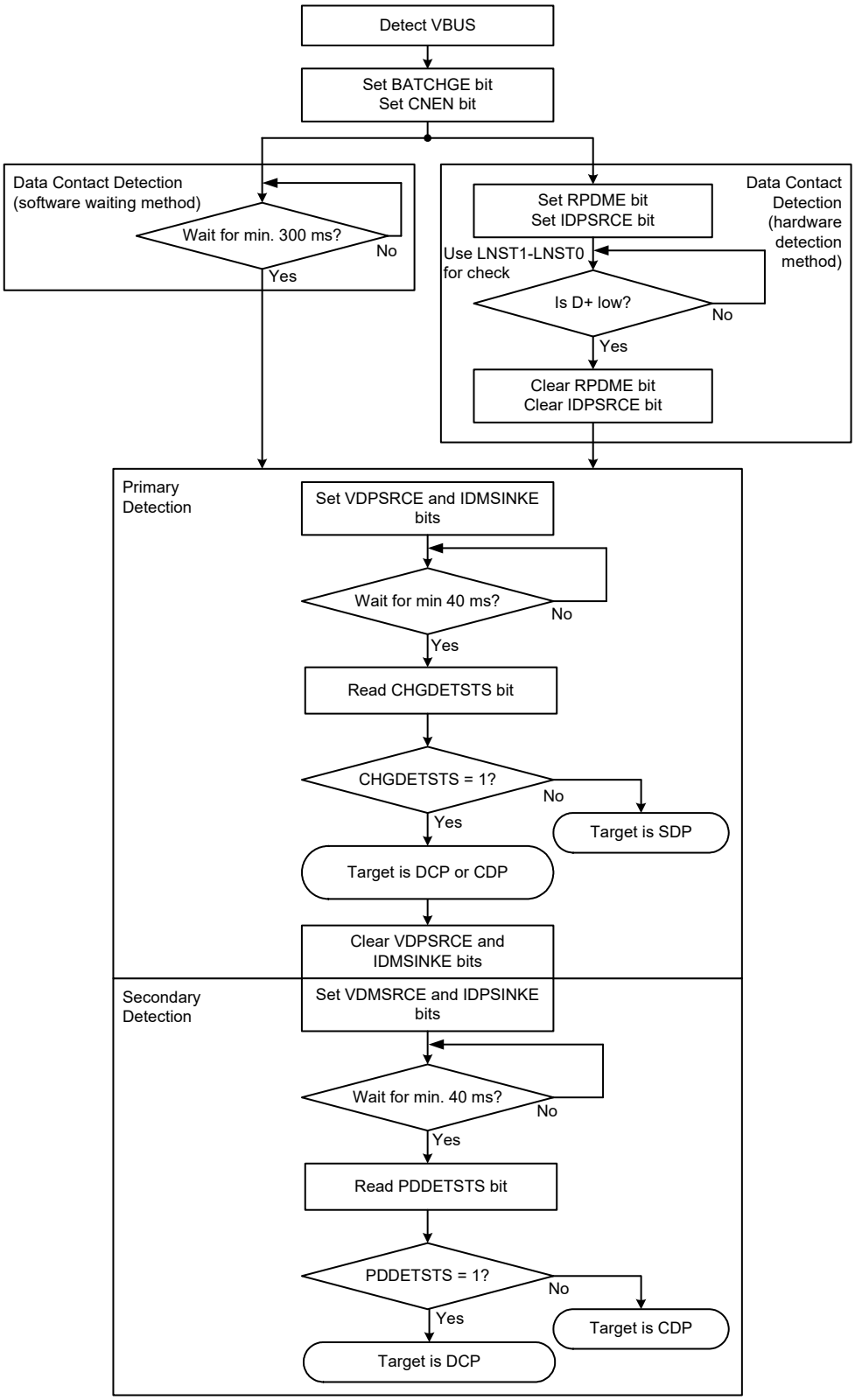
For the above step (4), set the VDMSRCE and IDPSINKE bits and wait for 40 ms by software, and then use the PDDETSTS bit to verify the Secondary Detection result.

Figure 13 - 66 shows the Process Flow for Operating as Portable Device.

Note 1. The Battery Charging Specification describes two implementation methods of the process flow for Data Contact Detection (D+/D- line contact check). One of the methods is to detect a change to logic low due to the pull-down resistor of the Host Device when the D+/D- lines have made contact with the target while the D+ line is held at logic high by applying a current of 7 to 13 uA on the D+ line.
The other method is to wait for 300 ms to 900 ms after VBUS is detected.

Note 2. During Primary Detection, when the voltage on the D- line is detected to be 0.25 V to 0.4 V or above and 0.8 V to 2.0 V or below, the target device is recognized as the Host Device for Battery Charging (Charging Downstream Port).
When using a PHY in which the 0CHGDETSTS bit only indicates that the voltage on the D- line is 0.25 V to 0.4 V or above, add the processing to check that the voltage on D- line is 0.8 V to 2.0 V or below using the LNST bits, as necessary.

Figure 13 - 66 Process Flow for Operating as Portable Device



13.4.13.2 Processing when host controller is selected

The following processing is required when operating the USB module as a Charging Downstream Port for Battery Charging.

(1) Start driving the VBUS.
(2) Enable the Portable Device detection circuit.
(3) Monitor the Portable Device detection signal, and start driving the D- line if the detection signal is high.
(4) Detect when the Portable Device detection signal is low level and stop driving the D- line.

Or perform the following processing.

(A) After disconnection is detected, start driving the D- line within 200 ms.
(B) After connection is detected, stop driving the D- line within 10 ms.

The D- line must be driven to allow the Portable Device to detect the Primary Detection described in section 13.4.13.1. The above steps (1) to (4) apply when the Portable Device detection function is provided by hardware. This method is to drive the D- line when the Portable Device is detected. The above steps (A) and (B) apply when the Portable Device function is not provided or used by hardware. Regardless of detection of the Portable Device, the D- line is driven in the Dis-Connect state and the line is not driven in the Connect state. In the Battery Charging Specification, either of these methods can be used.

For the above steps (3) and (4), after a change in the Portable Device detection signal is detected using the PDDETINT interrupt, the current signal state can be confirmed by reading the PDDETSTS bit.

The above steps (A) and (B) can be performed only in a software timer.

Figure 13 - 67 shows the above process flow for the above steps (1) to (4) and Figure 13 - 68 shows the process flow for the above steps (A) to (B).

Figure 13 - 67 Process Flow for Operating as Charging Downstream Port (Steps (1) to (4))

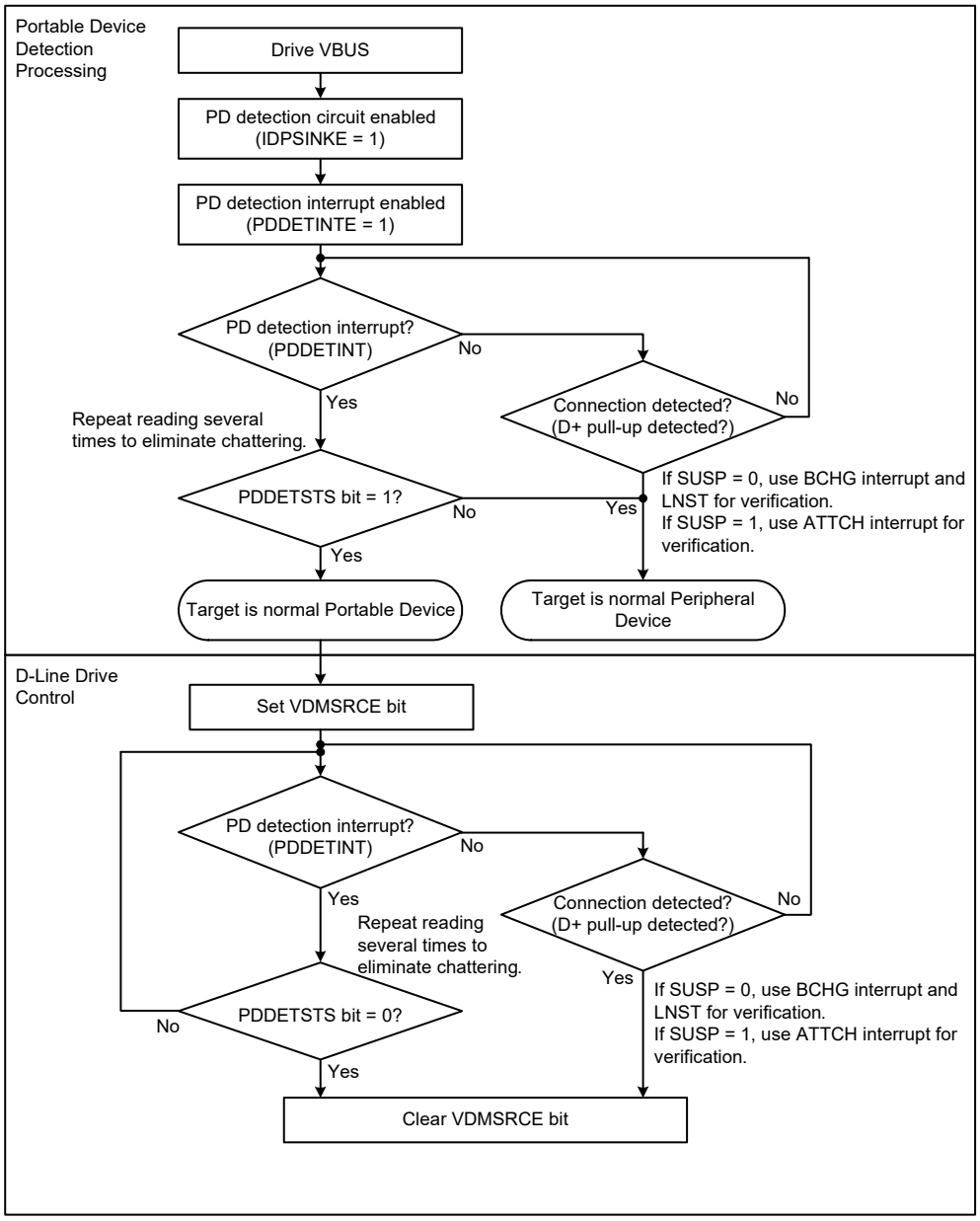
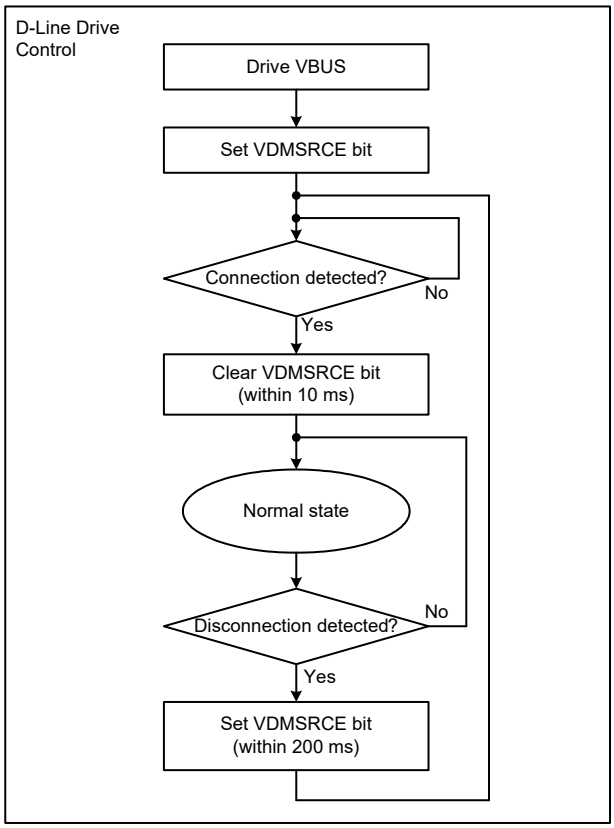


Figure 13 - 68 Process Flow for Operating as Charging Downstream Port (Steps (A) to (B))



CHAPTER 14 INTERRUPT FUNCTIONS

The interrupt function switches the program execution to other processing. When the branch processing is finished, the program returns to the interrupted processing.
The number of interrupt sources differs, depending on the product.

		with USB	without USB
		32-pin	32-pin
Maskable interrupts	External	14	16
	Internal	22	21

14.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H).
Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the default priority of vectored interrupt servicing. Default priority, see **Table 14 - 1**.
A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.
External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

14.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 14 - 1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 14 - 1 Interrupt Source List (1/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	with USB	without USB
		Name	Trigger					
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% of overflow time + 1/2 fIL)	Internal	0004H	(A)	✓	✓
	1	INTLVI	Voltage detection Note 4		0006H		✓	✓
	2	INTP0	Pin input edge detection	External	0008H	(B)	✓	✓
	3	INTP1			000AH		✓	✓
	4	INTP2			000CH		✓	✓
	5	INTP3			000EH		✓	✓
	6	INTP4			0010H		✓	✓
	7	INTP5			0012H		✓	✓
	8	INTP6			0014H		✓	✓
	9	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/ CSI00 transfer end or buffer empty interrupt/ IIC00 transfer end	Internal	0016H	(A)	✓	✓
	10	INTSR0/ INTCSI01/ INTIIC01	UART0 reception transfer end/ CSI01 transfer end or buffer empty interrupt/ IIC01 transfer end		0018H		✓	✓
	11	INTSRE0	UART0 reception communication error occurrence		001EH		✓	✓
	12	INTTM00	End of TAU channel 00 count or capture (at 16-bit operation or lower 8-bit operation)		0020H		✓	✓
	13	INTTM01H	End of TAU channel 01 count or capture (at higher 8-bit operation)		0026H		✓	✓
	14	INTTM03H	End of TAU channel 03 count or capture (at higher 8-bit operation)		0028H		✓	✓
	15	INTTM01	End of TAU channel 01 count or capture (at 16-bit operation or lower 8-bit operation)		002AH		✓	✓
	16	INTTM02	End of TAU channel 02 count or capture (at 16-bit operation or lower 8-bit operation)		002CH		✓	✓
	17	INTTM03	End of TAU channel 03 count or capture (at 16-bit operation or lower 8-bit operation)		002EH		✓	✓
	18	INTIICA0	End of IICA0 communication		0030H		✓	✓
	19	INTIICA1	End of IICA1 communication		0032H		✓	✓
	20	INTAD	End of A/D conversion		0034H		✓	✓
	21	INTIT	12-bit interval timer interval signal detection		0038H		✓	✓
	22	INTUSB	USB INT interrupt		003CH		✓	—
	23	INTRSUM	USB RESUME interrupt		003EH		✓	—
	24	INTIICA2	End of IICA2 communication		0040H		—	✓
	25	INTTM04	End of timer channel 04 count or capture		0044H		✓	✓
	26	INTTM05	End of timer channel 05 count or capture		0046H		✓	✓
	27	INTTM06	End of timer channel 06 count or capture		0048H		✓	✓
	28	INTTM07	End of timer channel 07 count or capture		004AH		✓	✓

Table 14 - 1 Interrupt Source List (2/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	with USB	without USB
		Name	Trigger					
Maskable	29	INTP7	Pin input edge detection	External	004CH	(B)	✓	✓
	30	INTP8			004EH		✓	✓
	31	INTP9			0050H		✓	✓
	32	INTP10			0052H		✓	✓
	33	INTP11			0054H		✓	✓
	34	INTP12			0056H		✓	✓
	35	INTP13			0058H		✓	✓
	36	INTP14			005AH		—	✓
	37	INTP15			005CH		—	✓
	38	INTFL	End of Sequencer Note 5	Internal	0062H	(A)	✓	✓
Software	—	BRK	Execution of BRK instruction	—	007EH	(C)	✓	✓
Reset	—	RESET	RESET pin input	—	0000H	—	✓	✓
		POR	Power-on-reset				✓	✓
		LVD	Voltage detection Note 6				✓	✓
		WDT	Overflow of watchdog timer				✓	✓
		TRAP	Execution of illegal instruction Note 7				✓	✓
		IAW	Illegal-memory access				✓	✓
		RPE	RAM parity error				✓	✓

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 38 indicates the lowest priority.

Note 2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figures 14 - 1 and 14 - 2.

Note 3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

Note 4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

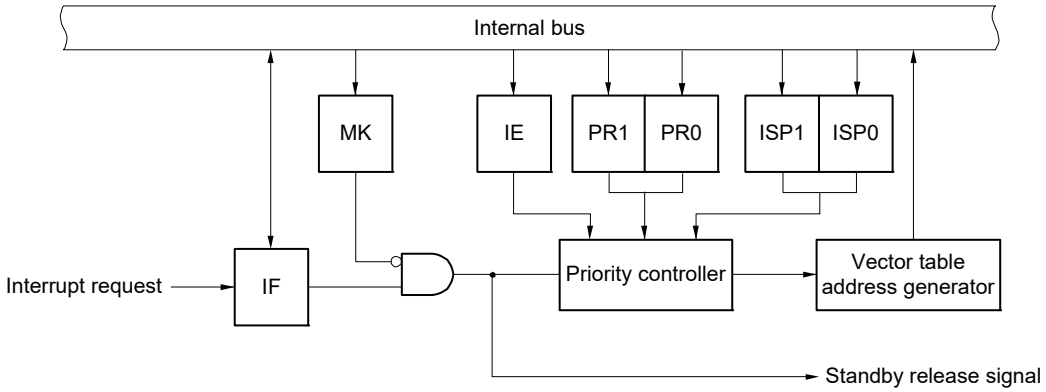
Note 5. Be used at the flash self programming library or the data flash library.

Note 6. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

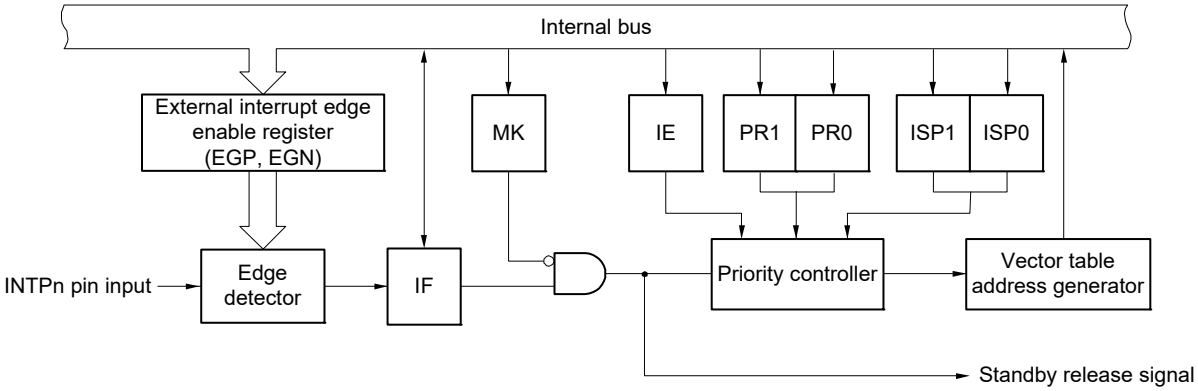
Note 7. When the instruction code in FFH is executed. Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 14 - 1 Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

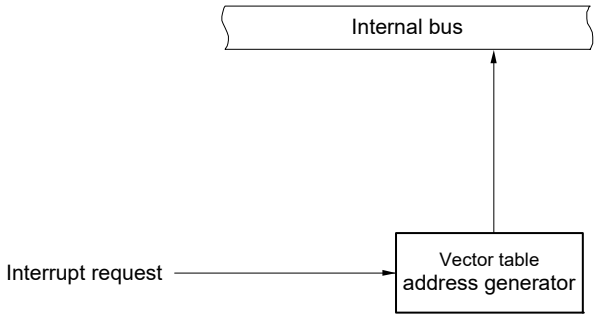


IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark with USB: n = 0 to 13
without USB: n = 0 to 15

Figure 14 - 2 Basic Configuration of Interrupt Function (2/2)

(C) Software interrupt



14.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- External interrupt rising edge enable registers (EGP0, EGP1)
- External interrupt falling edge enable registers (EGN0, EGN1)
- Program status word (PSW)

Table 14 - 2 show a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 14 - 2 Flags Corresponding to Interrupt Request Sources (1/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		with USB	without USB
		Register		Register		Register		
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	✓	✓
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		✓	✓
INTP0	PIF0		PMK0		PPR00, PPR10		✓	✓
INTP1	PIF1		PMK1		PPR01, PPR11		✓	✓
INTP2	PIF2		PMK2		PPR02, PPR12		✓	✓
INTP3	PIF3		PMK3		PPR03, PPR13		✓	✓
INTP4	PIF4		PMK4		PPR04, PPR14		✓	✓
INTP5	PIF5		PMK5		PPR05, PPR15		✓	✓
INTP6	PIF6	IF0H	PMK6	MK0H	PPR06, PPR16	PR00H, PR10H	✓	✓
INTST0 Note 1	STIF0		STMK0		STPR00, STPR10		✓	✓
INTCSI00 Note 1	CSIIF00		CSIMK00		CSIPR000, CSIPR100		✓	✓
INTIIC00 Note 1	IICIF00		IICMK00		IICPR000, IICPR100		✓	✓
INTSR0 Note 2	SRIF0		SRMK0		SRPR00, SRPR10		✓	✓
INTCSI01 Note 2	CSIIF01		CSIMK01		CSIPR001, CSIPR101		✓	✓
INTIIC01 Note 2	IICIF01		IICMK01		IICPR001, IICPR101		✓	✓
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10		✓	✓
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		✓	✓
INTTM01H	TMIF01H	IF1L	TMMK01H	MK1L	TMPR001H, TMPR101H	PR01L, PR11L	✓	✓
INTTM03H	TMIF03H		TMMK03H		TMPR003H, TMPR103H		✓	✓
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		✓	✓
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		✓	✓
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		✓	✓
INTIICA0	IICAIF0		IICAMK0		IICAPR00, ICAPR10		✓	✓
INTIICA1	IICAIF1		IICAMK1		IICAPR01, ICAPR11		✓	✓
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	✓	✓
INTIT	TMKAIF		TMKAMK		TMKAPR0, TMKAPR1		✓	✓
INTUSB	USBIF		USBMK		USBPR0, USBPR1		✓	—
INTRSUM	RSUIF		RSUMK		RSUPR0, RSUPR1		✓	—
INTIICA2	IICAIF2		IICAMK2		IICAPR02, IICAPR12		—	✓
INTTM04	TMIF04	IF2L	TMMK04	MK2L	TMPR004, TMPR104	PR02L, PR12L	✓	✓
INTTM05	TMIF05		TMMK05		TMPR005, TMPR105		✓	✓
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		✓	✓
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		✓	✓
INTP7	PIF7		PMK7		PPR07, PPR17		✓	✓
INTP8	PIF8		PMK8		PPR08, PPR18		✓	✓
INTP9	PIF9		PMK9		PPR09, PPR19		✓	✓
INTP10	PIF10		PMK10		PPR010, PPR110		✓	✓
INTP11	PIF11	IF2H	PMK11	MK2H	PPR011, PPR111	PR02H, PR12H	✓	✓
INTP12	PIF12		PMK12		PPR012, PPR112		✓	✓
INTP13	PIF13		PMK13		PPR013, PPR113		✓	✓
INTP14	PIF14		PMK14		PPR014, PPR114		—	✓
INTP15	PIF15		PMK15		PPR015, PPR115		—	✓
INTFL	FLIF		FLMK		FLPR0, FLPR1		✓	✓

Note 1. If one of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 1 of the IF0H register is set to 1. Bit 1 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 2. If one of the interrupt sources INTSR0, INTCSI01, and INTIIC01 is generated, bit 2 of the IF0H register is set to 1. Bit 2 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

14.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, IF2L and IF2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers and the IF1L and IF1H registers and the IF2L and IF2H registers are combined to form 16-bit registers IF0, IF1 and IF2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 14 - 3 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
IF0H	0	TMIF00	SREIF0	0	0	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	PIF6

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	0
IF1L	IICAIF1	IICAIF0	TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	0

Address: FFFE3H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	<2>	1	<0>
IF1H	0	IICAIF2	RSUIF	USBIF	0	TMKAIF	0	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	PIF10	PIF9	PIF8	PIF7	TMIF07	TMIF06	TMIF05	TMIF04

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>
IF2H	INTFL	0	0	PIF15	PIF14	PIF13	PIF12	PIF11

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution 1. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 14 - 2. Be sure to set bits that are not available to the initial value.

Caution 2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm ("clr1 IF0L.0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).
If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.
mov a, IF0L
and a, #0FEH
mov IF0L, a

In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

14.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt.
The MK0L, MK0H, MK1L, MK1H, MK2L and MK2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers and the MK1L and MK1H registers and the MK2L and MK2H registers are combined to form 16-bit registers MK0, MK1 and MK2 they can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 14 - 4 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
MK0H	0	TMMK00	SREMK0	0	0	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	PMK6

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	0
MK1L	IICAMK1	IICAMK0	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	0

Address: FFFE7H After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	3	<2>	1	<0>
MK1H	0	IICAMK2	RSUMK	USBMK	0	TMKAMK	0	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	PMK10	PMK9	PMK8	PMK7	TMMK07	TMMK06	TMMK05	TMMK04

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>
MK2H	FLMK	0	0	PMK15	PMK14	PMK13	PMK12	PMK11

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 14 - 2. Be sure to set bits that are not available to the initial value.

14.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.
A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L or 2H).
The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L and PR12H registers can be set by a 1-bit or 8-bit memory manipulation instruction. If the PR00L and PR00H registers, the PR01L and PR01H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, and the PR12L and PR12H registers are combined to form 16-bit registers PR00, PR01, PR10, PR11 and PR12, they can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 14 - 5 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset:FFH R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
PR00H	0	TMPR000	SREPR00	0	0	SRPR00 CSIPR001 IICPR001	STPR00 CSIPR000 IICPR000	PPR06

Address: FFFEDH After reset: FFH R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
PR10H	0	TMPR100	SREPR10	0	0	SRPR10 CSIPR101 IICPR101	STPR10 CSIPR100 IICPR100	PPR16

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	0
PR01L	IICAPR01	IICAPR00	TMPR003	TMPR002	TMPR001	TMPR003H	TMPR001H	0

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	0
PR11L	IICAPR11	IICAPR10	TMPR103	TMPR102	TMPR101	TMPR103H	TMPR101H	0

Figure 14 - 6 Format of Priority Specification Flag Registers
(PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)

Address: FFFEBH After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	3	<2>	1	<0>
PR01H	0	IICAPR02	RSUPR0	USBPR0	0	TMKAPR0	0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	3	<2>	1	<0>
PR11H	0	IICAPR12	RSUPR1	USBPR1	0	TMKAPR1	0	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	PPR010	PPR09	PPR08	PPR07	TMPR007	TMPR006	TMPR005	TMPR004

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	PPR110	PPR19	PPR18	PPR17	TMPR107	TMPR106	TMPR105	TMPR104

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>
PR02H	FLPR0	0	0	PPR015	PPR014	PPR013	PPR012	PPR011

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>
PR12H	FLPR1	0	0	PPR115	PPR114	PPR113	PPR112	PPR111

XXPR1X	XXPR0X	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	1	Specify level 3 (low priority level)

Caution The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 14 - 2. Be sure to set bits that are not available to the initial value.

14.3.4 External interrupt rising edge enable register (EGP0, EGP1), external interrupt falling edge enable register (EGN0, EGN1)

These registers specify the valid edge for INTP0 to INT15.

The EGP0, EGP1, EGN0, EGN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 14 - 7 Format of External Interrupt Rising Edge Enable Register (EGP0, EGP1) and External Interrupt Falling Edge Enable Register (EGN0, EGN1)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	EGP15	EGP14	EGP13	EGP12	EGP11	EGP10	EGP9	EGP8

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	EGN15	EGN14	EGN13	EGN12	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 15)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 14 - 3 shows the Ports Corresponding to EGPn and EGNn Bits.

Table 14 - 3 Ports Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Interrupt Request Signal	with USB	without USB
			32-pin	32-pin
EGP0	EGN0	INTP0	✓	✓
EGP1	EGN1	INTP1	✓	✓
EGP2	EGN2	INTP2	✓	✓
EGP3	EGN3	INTP3	✓	✓
EGP4	EGN4	INTP4	✓	✓
EGP5	EGN5	INTP5	✓	✓
EGP6	EGN6	INTP6	✓	✓
EGP7	EGN7	INTP7	✓	✓
EGP8	EGN8	INTP8	✓	✓
EGP9	EGN9	INTP9	✓	✓
EGP10	EGN10	INTP10	✓	✓
EGP11	EGN11	INTP11	✓	✓
EGP12	EGN12	INTP12	✓	✓
EGP13	EGN13	INTP13	✓	✓
EGP14	EGN14	INTP14	—	✓
EGP15	EGN15	INTP15	—	✓

Caution When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge.
When switching the input port pins to the output mode, set the port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).

Remark 1. For edge detection port, see **2.1 Port Functions**.
Remark 2. n = 0 to 15

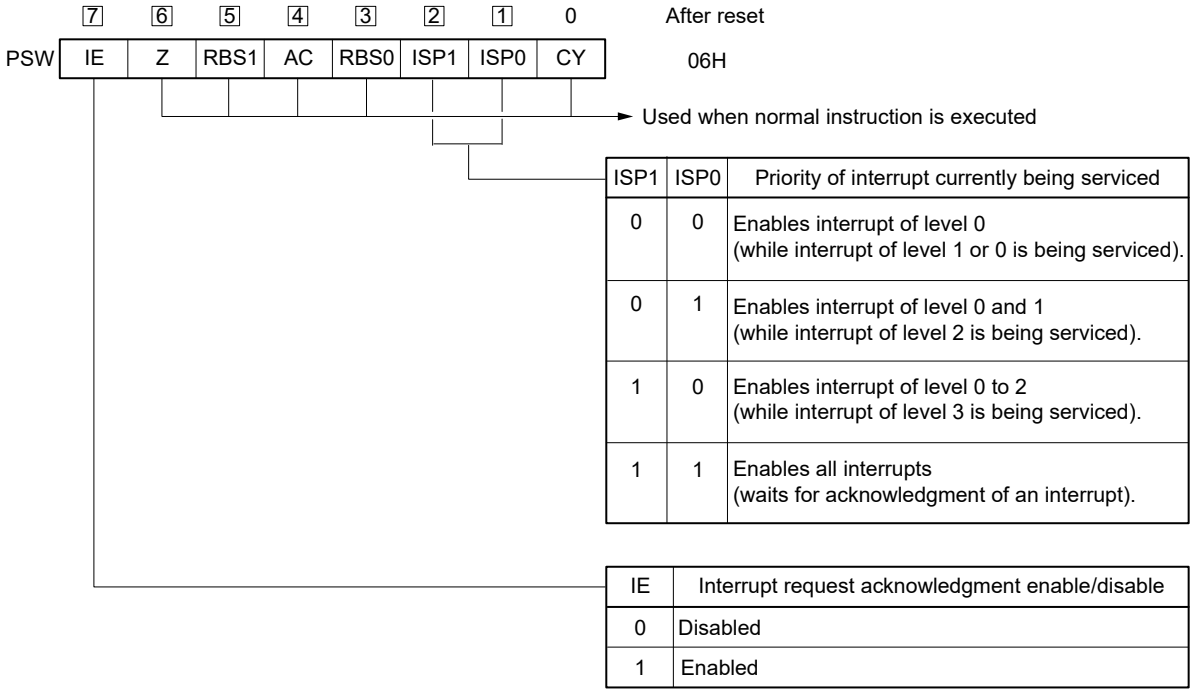
14.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. Upon acknowledgment of a maskable interrupt request, if the value of the priority specification flag register of the acknowledged interrupt is not 00, its value minus 1 is transferred to the ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 14 - 8 Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

14.4 Interrupt Servicing Operations

14.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 14 - 4 below.

For the interrupt request acknowledgment timing, see **Figures 14 - 10** and **14 - 11**.

Table 14 - 4 Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time Note
Servicing time	9 clocks	16 clocks

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: 1/fCLK (fCLK: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

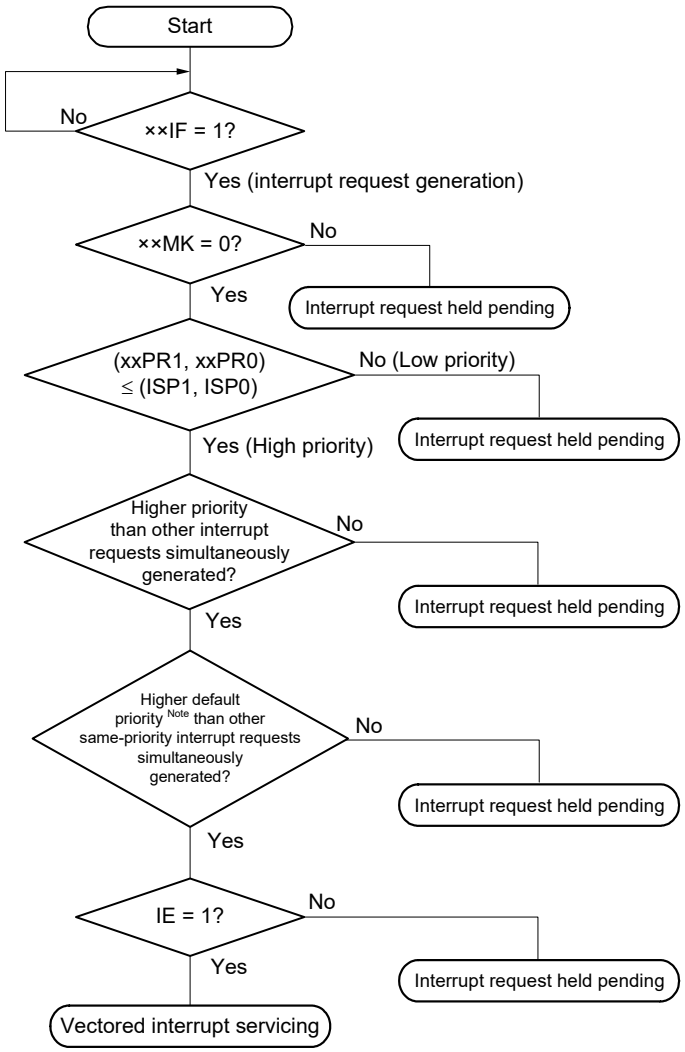
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 14 - 9 shows the Interrupt Request Acknowledgment Processing Algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

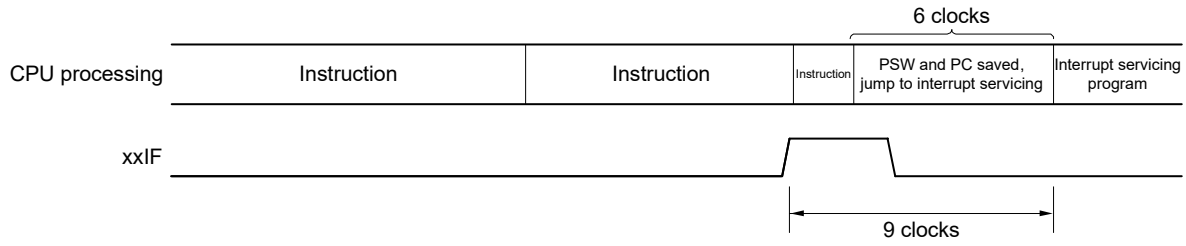
Figure 14 - 9 Interrupt Request Acknowledgment Processing Algorithm



xxIF: Interrupt request flag
xxMK: Interrupt mask flag
xxPR0: Priority specification flag 0
xxPR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 14 - 8**)

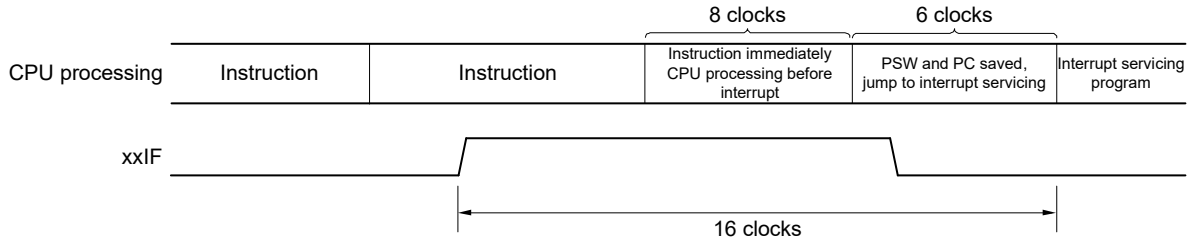
Note For the default priority, refer to **Table 14 - 1 Interrupt Source List**.

Figure 14 - 10 Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

Figure 14 - 11 Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

14.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Can not use the RETI instruction for restoring from the software interrupt.

14.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority equal to or lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. However, when setting the IE flag to 1 during the interruption at level 0, other level 0 interruptions can be allowed.

Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 14 - 5 shows Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing and Figures 14 - 12 and 14 - 13 show multiple interrupt servicing examples.

Table 14 - 5 Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0 ISP0 = 0	✓	×	×	×	×	×	×	×	✓
	ISP1 = 0 ISP0 = 1	✓	×	✓	×	×	×	×	×	✓
	ISP1 = 1 ISP0 = 0	✓	×	✓	×	✓	×	×	×	✓
	ISP1 = 1 ISP0 = 1	✓	×	✓	×	✓	×	✓	×	✓
Software interrupt		✓	×	✓	×	✓	×	✓	×	✓

Remark 1. ✓: Multiple interrupt servicing enabled

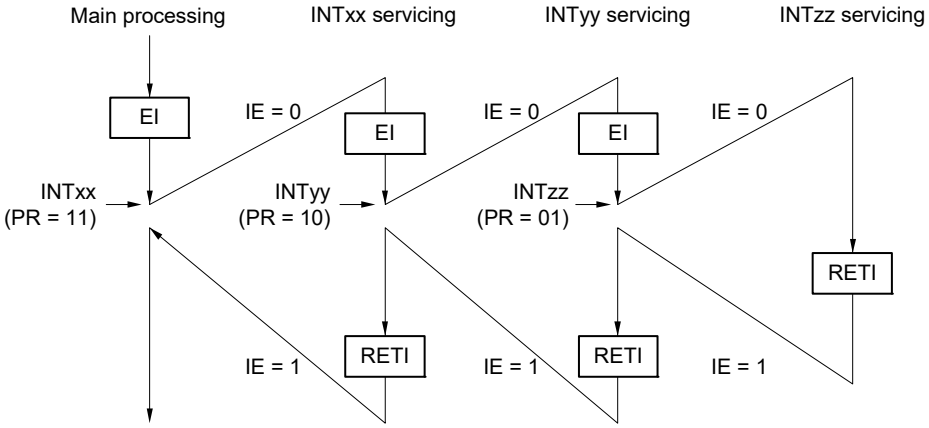
Remark 2. ×: Multiple interrupt servicing disabled

Remark 3. ISP0, ISP1, and IE are flags contained in the PSW.
ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.
ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.
ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.
ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment (all interrupts enabled).
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

Remark 4. PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers.
PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

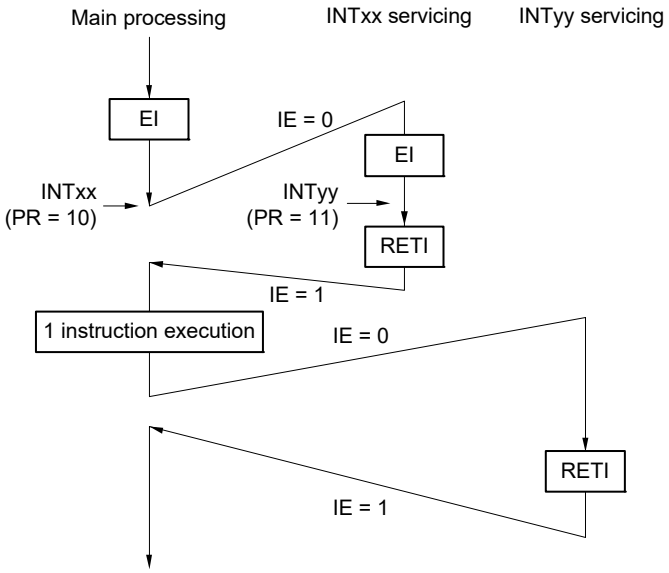
Figure 14 - 12 Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

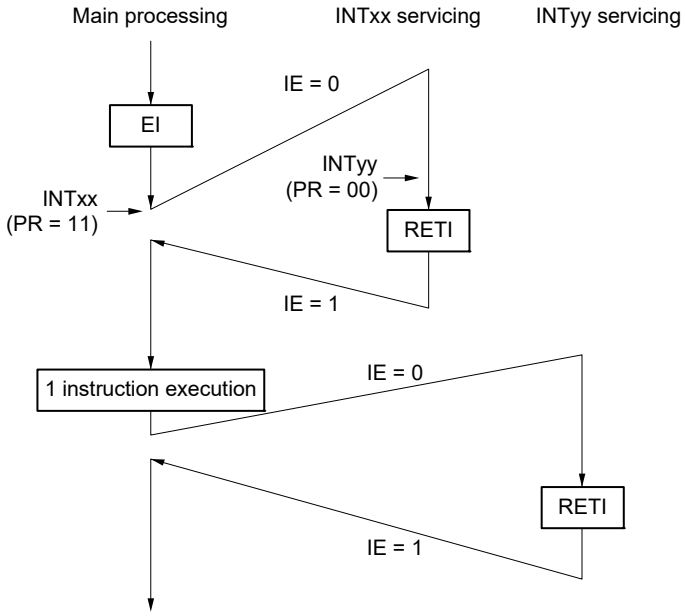


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 14 - 13 Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

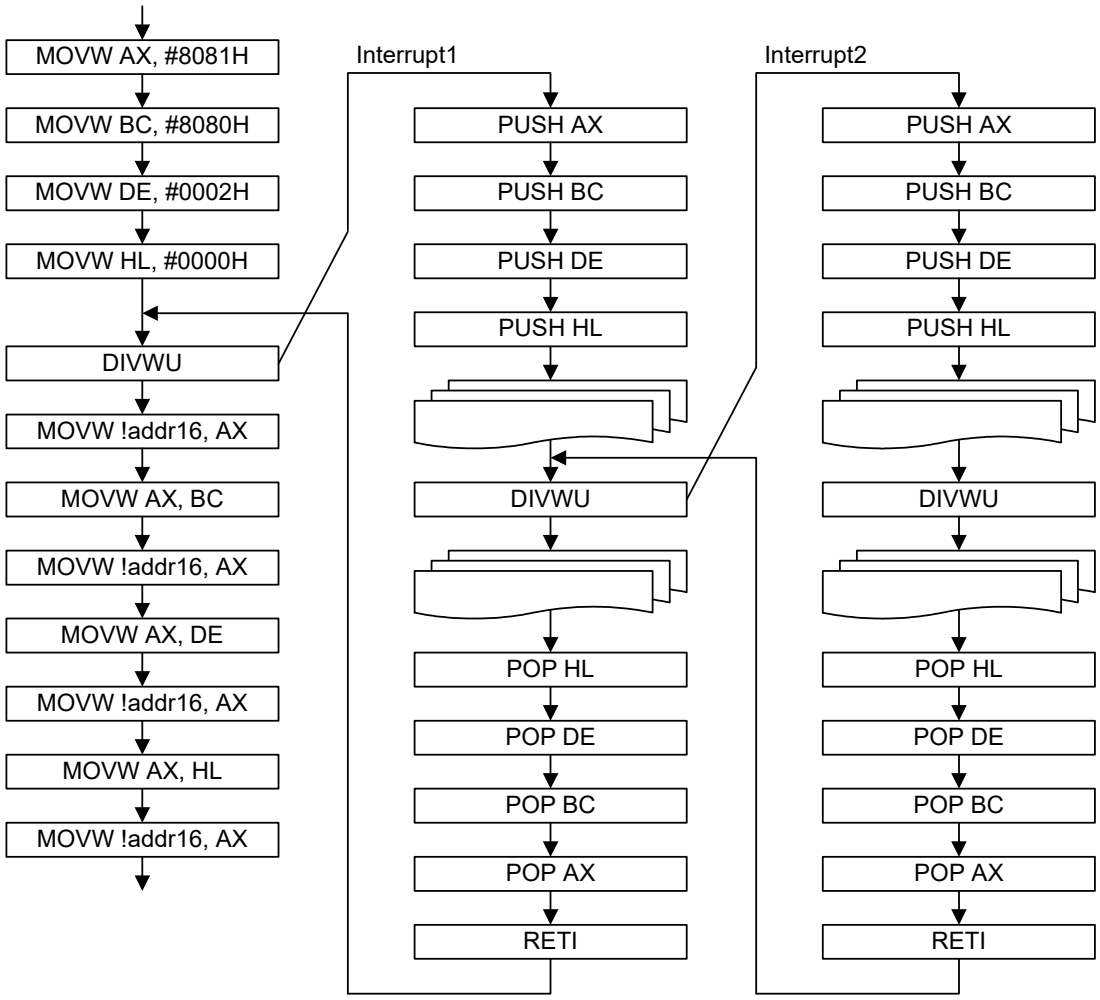
14.4.4 Interrupt servicing during division instruction

The R9A02G015 handles interrupts during the DIVHU/DIVWU instruction in order to enhance the interrupt response when a division instruction is executed.

- When an interrupt is generated while the DIVHU/DIVWU instruction is executed, the instruction is suspended
- After the instruction is suspended, the PC indicates the next instruction after DIVHU/DIVWU
- An interrupt is generated by the next instruction
- PC-3 is stacked to execute the DIVHU/DIVWU instruction again

Normal interrupt	Interrupts while Executing DIVHU/DIVWU Instruction
(SP-1) ← PSW	(SP-1) ← PSW
(SP-2) ← $(PC)_S$	(SP-2) ← $(PC-3)_S$
(SP-3) ← $(PC)_H$	(SP-3) ← $(PC-3)_H$
(SP-4) ← $(PC)_L$	(SP-4) ← $(PC-3)_L$
PC_S ← 0000	PC_S ← 0000
PC_H ← (Vector)	PC_H ← (Vector)
PC_L ← (Vector)	PC_L ← (Vector)
SP ← SP-4	SP ← SP-4
IE ← 0	IE ← 0

The AX, BC, DE, and HL registers are used for DIVHU/DIVWU. Use these registers by stacking them for interrupt servicing.



Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine.

Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.3 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

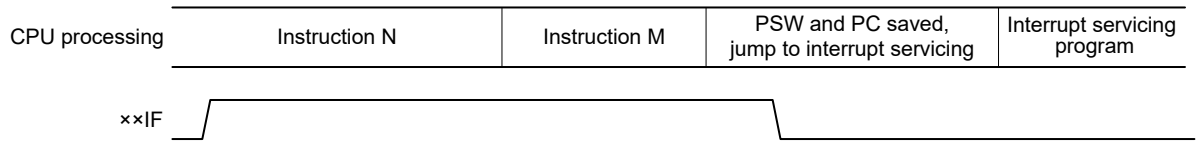
14.4.5 Interrupt request hold

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- MULHU
- MULH
- MACHU
- MACH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers

Figure 14 - 14 shows the timing at which interrupt requests are held pending.

Figure 14 - 14 Interrupt Request Hold



Remark 1. Instruction N: Interrupt request hold instruction
Remark 2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 15 STANDBY FUNCTION

15.1 Standby Function

The standby function reduces the operating current of the system, and the following three modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, or high-speed on-chip oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and high-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

(3) SNOOZE mode

In the case of CSI00 or UART0 data reception, an A/D conversion request by the timer trigger signal (the interrupt request signal (INTIT) the STOP mode is exited, the CSI00 or UART0 data is received without operating the CPU, and A/D conversion is performed. This can only be specified when the high-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Caution 1. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (except SNOOZE mode setting unit).

Caution 2. When using CSI00, UART0, or the A/D converter in the SNOOZE mode, set up serial standby control register m (SSCm) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 11.3 Registers Controlling Serial Array Unit and 10.3 Registers Controlling A/D Converter.

Caution 3. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

Caution 4. It can be selected by the option byte whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 21 OPTION BYTE.

15.2 Registers controlling standby function

The registers which control the standby function are described below.

- Subsystem clock supply mode control register (OSMC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For details of registers described above, see **CHAPTER 5 CLOCK GENERATOR**. For registers which control the SNOOZE mode, **CHAPTER 10 A/D CONVERTER** and **CHAPTER 11 SERIAL ARRAY UNIT**.

15.3 Standby Function Operation

15.3.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, or high-speed on-chip oscillator clock.

The operating statuses in the HALT mode are shown below.

Caution Because the interrupt request signal is used to clear the HALT mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the HALT mode is not entered even if the HALT instruction is executed in such a situation.

Table 15 - 1 Operating Statuses in HALT Mode

Item \ HALT Mode Setting		When HALT Instruction Is Executed While CPU Is Operating on Main System Clock			
		When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{HOCO})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})	When CPU Is Operating on PLL Clock (f_{PLL})
System clock		Clock supply to the CPU is stopped			
Main system clock	f_{HOCO}	Operation continues (cannot be stopped)	Operation disabled		
	f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate	Cannot be stopped while the clock is supplied to the PLL
	f_{EX}		Cannot operate	Operation continues (cannot be stopped)	Cannot be stopped while the clock is supplied to the PLL
	f_{PLL}		Operation disabled	Operation disabled	Operation continues (cannot be stopped)
f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops			
CPU		Operation stopped			
Code flash memory					
Data flash memory					
RAM					
Port (latch)		Status before HALT mode was set is retained			
Timer array unit		Operable			
12-bit Interval timer					
Watchdog timer		See **CHAPTER 9 WATCHDOG TIMER**			
Clock output/buzzer output		Operable			
A/D converter					
Serial array unit (SAU)					
Serial interface (IICA)					
USB		Operable (low speed mode transmission only)	Operation disabled		Operable
Power-on-reset function		Operable			
Voltage detection function					
External interrupt					
CRC operation function	High-speed CRC				
	General-purpose CRC	Operation stopped			
RAM parity error detection function		Operation stopped			
RAM guard function					
SFR guard function					
Illegal-memory access detection function					

Remark Operation stopped: Operation is automatically stopped before switching to HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
f_{HOCO}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock

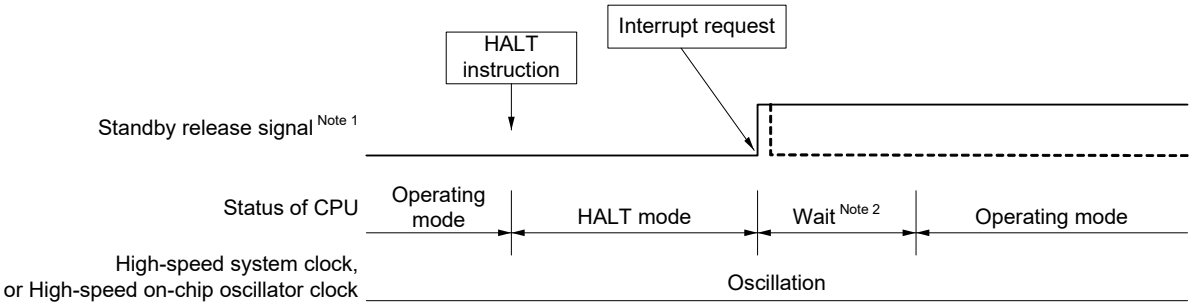
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 15 - 1 HALT Mode Release by Interrupt Request Generation



Note 1. For details of the standby release signal, see **Figure 14 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out
  Main system clock: 15 to 16 clocks
- When vectored interrupt servicing is not carried out
  Main system clock: 9 to 10 clocks

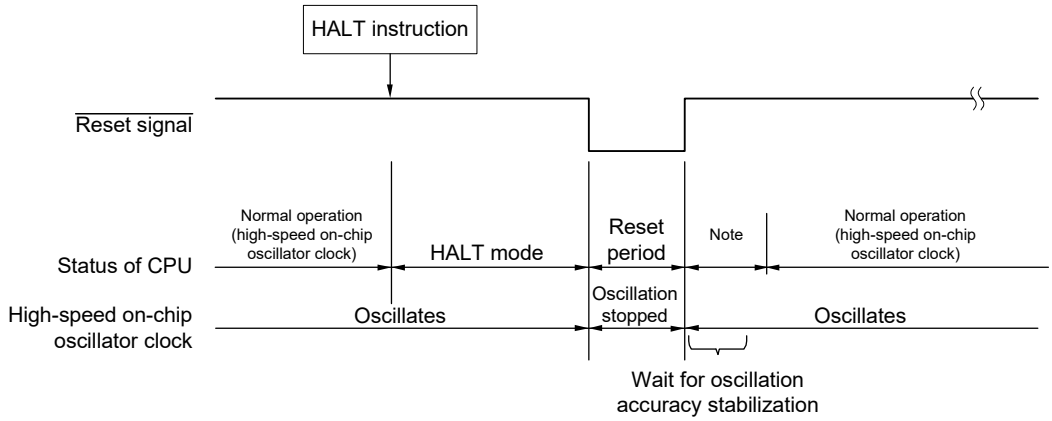
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

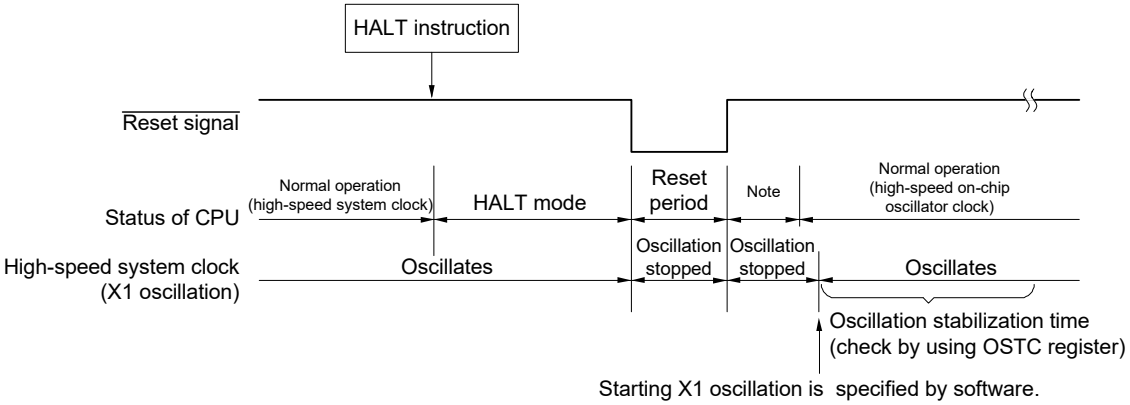
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 15 - 2 HALT Mode Release by Reset

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 16 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 17 POWER-ON-RESET CIRCUIT**.

15.3.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the high-speed on-chip oscillator clock, X1 clock, or external main system clock.

Caution 1. Because the interrupt request signal is used to clear the STOP mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the STOP mode is immediately cleared if set when the STOP instruction is executed in such a situation.
Accordingly, once the STOP instruction is executed, the system returns to its normal operating mode after the elapse of release time from the STOP mode.

Caution 2. When shifting to STOP mode, be sure to stop the PLL operation by setting the DSCON bit (bit 0 in the DSCCTL register) before executing the STOP instruction.

The operating statuses in the STOP mode are shown below.

Table 15 - 2 Operating Statuses in STOP Mode

Item \ STOP Mode Setting		When STOP Instruction Is Executed While CPU Is Operating on Main System Clock			
		When CPU Is Operating on High-speed on-chip oscillator clock (f_{HOCO})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})	When CPU Is Operating on PLL Clock (f_{PLL})
System clock		Clock supply to the CPU is stopped			
Main system clock	f_{HOCO}	Stopped			
	f_X				
	f_{EX}				
	f_{PLL}	Operation disabled			
f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops			
CPU		Operation stopped			
Code flash memory					
Data flash memory					
RAM					
Port (latch)		Status before STOP mode was set is retained			
Timer array unit		Operation disabled			
12-bit Interval timer		Operable			
Watchdog timer		See **CHAPTER 9 WATCHDOG TIMER**			
Clock output/buzzer output		Operation stopped			
A/D converter		Wakeup operation is enabled (switching to the SNOOZE mode)			
Serial array unit (SAU)		Wakeup operation is enabled only for CSI00 (switching to the SNOOZE mode) Operation is disabled for anything other than CSI00			
Serial interface (IICA)		Wakeup by address match operable			
USB		Operation disabled			
Power-on-reset function		Operable			
Voltage detection function					
External interrupt					
CRC operation function	High-speed CRC	Operation stopped			
	General-purpose CRC				
RAM parity error detection function					
RAM guard function					
SFR guard function					
Illegal-memory access detection function					

Remark Operation stopped: Operation is automatically stopped before switching to STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{HOCO}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock

(2) STOP mode release

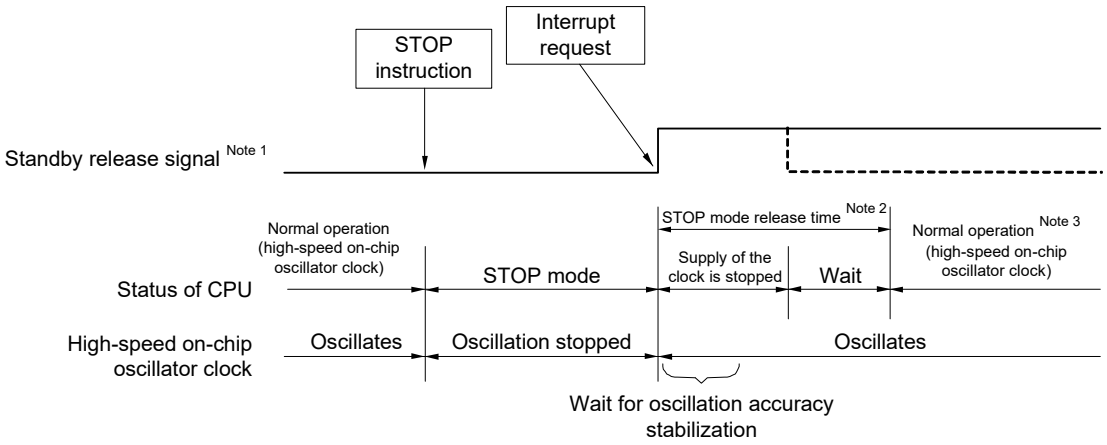
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 15 - 3 STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 14 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. STOP mode release time

Supply of the clock is stopped:

When high-speed on-chip oscillator clock: 18 μs to 65 μs

Wait:

(common to the high-speed on-chip oscillator clock)

•When vectored interrupt servicing is carried out: 7 clocks

•When vectored interrupt servicing is not carried out: 1 clock

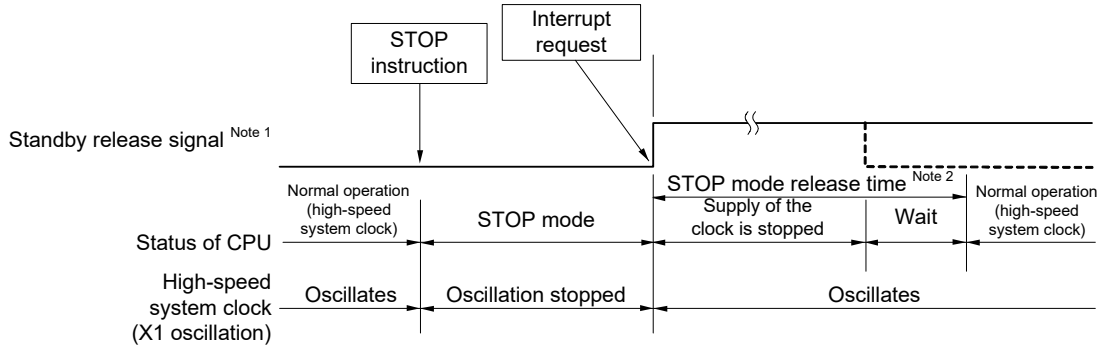
Caution To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction.

Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 15 - 4 STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 14 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. STOP mode release time

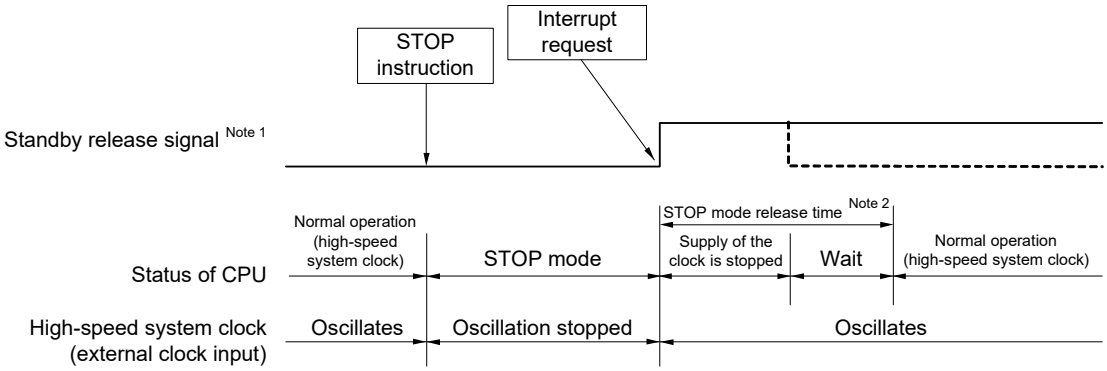
Supply of the clock is stopped:

18 μs to "whichever is longer 65 μs or the oscillation stabilization time (set by OSTS)"

Wait:

- When vectored interrupt servicing is carried out: 10 to 11 clocks
- When vectored interrupt servicing is not carried out: 4 to 5 clocks

(3) When high-speed system clock (external clock input) is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 14 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. STOP mode release time

Supply of the clock is stopped:

18 μs to 65 μs

Wait:

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Caution To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.

Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

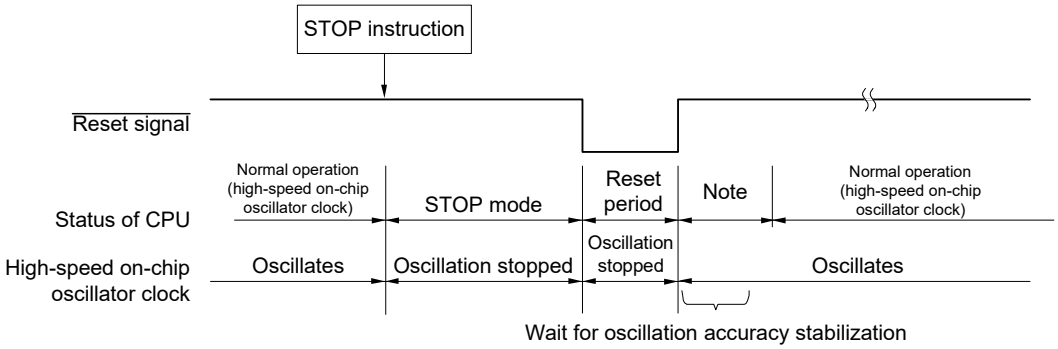
Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

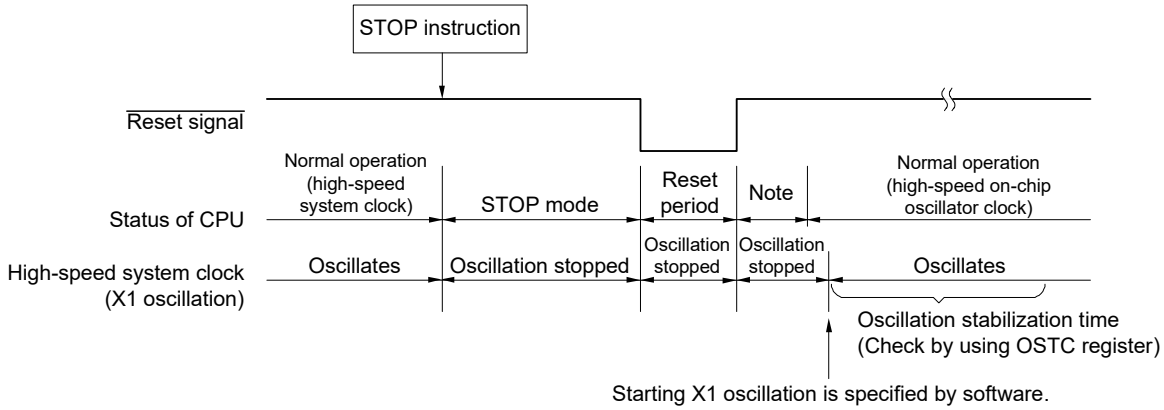
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 15 - 5 STOP Mode Release by Reset

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 16 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 17 POWER-ON-RESET CIRCUIT**.

15.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can be set by the CSI00, UART0, and A/D converter. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.

When using CSI00 or UART0 in the SNOOZE mode, set up serial standby control register m (SSCm) before switching to the STOP mode. For details, see **11.3 Registers Controlling Serial Array Unit**.

When using the A/D converter in the SNOOZE mode, set up A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see **10.3 Registers Controlling A/D Converter**.

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode:

When high-speed on-chip oscillator clock: 18 μs to 65 μs

Remark Transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

Transition time from SNOOZE mode to normal operation:

When high-speed on-chip oscillator clock:

- When vectored interrupt servicing is carried out:
  HS (High-speed main) mode: “5.2 μs to 9.4 μs” + 7 clocks
- When vectored interrupt servicing is not carried out:
  HS (High-speed main) mode: “5.2 μs to 9.4 μs” + 1 clock

The operating statuses in the SNOOZE mode are shown next.

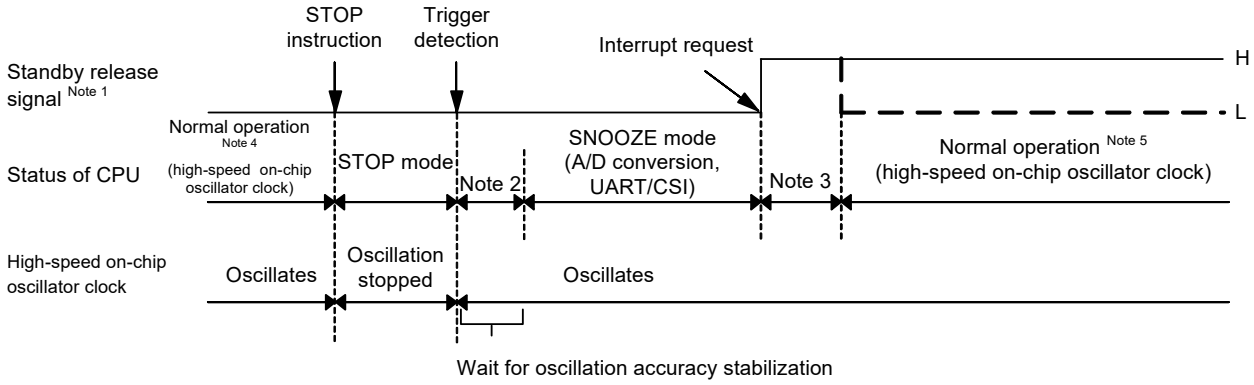
Table 15 - 3 Operating Statuses in SNOOZE Mode

Item \ SNOOZE Mode Setting		When Inputting CSI00 Data Reception Signal or A/D Converter Timer Trigger Signal While in STOP Mode
		When CPU Is Operating on High-speed on-chip oscillator clock (f_{HOCO})
System clock		Clock supply to the CPU is stopped
Main system clock	f_{HOCO}	Operation started
	f_X	Stopped
	f_{EX}	
	f_{PLL}	
f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops
CPU		Operation stopped
Code flash memory		
Data flash memory		
RAM		
Port (latch)		Use of the status while in the STOP mode continues
Timer array unit		Operation disabled
12-bit Interval timer		Operable
Watchdog timer		See **CHAPTER 9 WATCHDOG TIMER**
Clock output/buzzer output		Operation disabled
A/D converter		Operable
Serial array unit (SAU)		Operable only CSI00 only. Operation disabled other than CSI00.
Serial interface (IICA)		Operation disabled
USB		
Power-on-reset function		Operable
Voltage detection function		
External interrupt		
CRC operation function	High-speed CRC	Operation disabled
	General-purpose CRC	
RAM parity error detection function		
RAM guard function		
SFR guard function		
Illegal-memory access detection function		

Remark Operation stopped: Operation is automatically stopped before switching to the SNOOZE mode.
Operation disabled: Operation is stopped before switching to the SNOOZE mode.
f_{HOCO}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

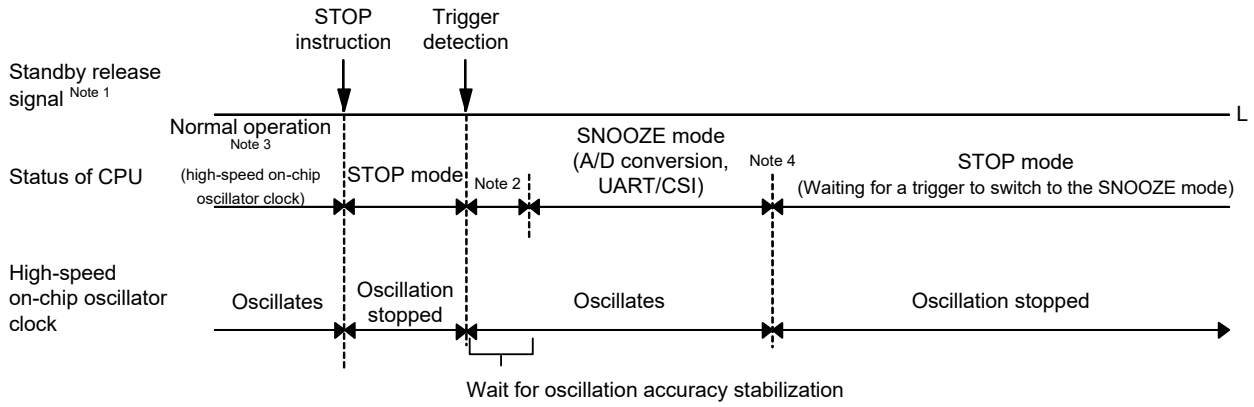
Figure 15 - 6 When the Interrupt Request Signal is Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 14 - 1**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Transition time from SNOOZE mode to normal operation
Note 4. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.
Note 5. Be sure to release the SNOOZE mode (AWC = 0 or SWC = 0) immediately after return to the normal operation.

(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 15 - 7 When the Interrupt Request Signal is not Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 14 - 1**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.
Note 4. If a standby release signal is generated in response to an interrupt from a module which is not set to operate in the SNOOZE mode during a transition of the chip from SNOOZE mode to STOP mode, the high-speed on-chip oscillator clock may run slowly for up to 15 μs from when the CPU starts to operate. If the clock frequency accuracy specified in the electrical characteristics is required immediately after release from standby, wait for the number of cycles at the actual CPU clock frequency that is equivalent to 15 μs.

Remark For details of the SNOOZE mode function, see **CHAPTER 10 A/D CONVERTER** and **CHAPTER 11 SERIAL ARRAY UNIT**.

CHAPTER 16 RESET FUNCTION

The following seven operations are available to generate a reset signal.

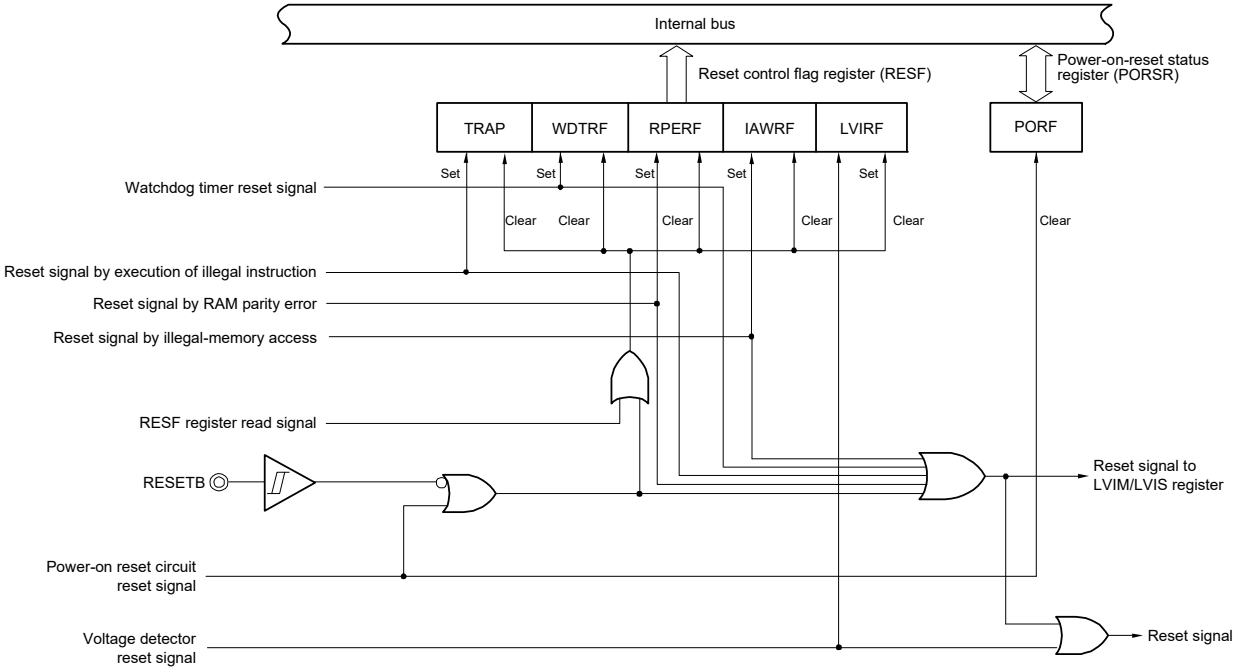
(1) External reset input via RESETB pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-reset (POR) circuit
(4) Internal reset by comparison of supply voltage of the voltage detector (LVD) and detection voltage
(5) Internal reset by execution of illegal instruction Note
(6) Internal reset by RAM parity error
(7) Internal reset by illegal-memory access

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.
A reset is effected when a low level is input to the RESETB pin, the watchdog timer overflows, or by POR and LVD circuit voltage detection, execution of illegal instruction Note, RAM parity error or illegal-memory access, and each item of hardware is set to the status shown in Table 16 - 1.

Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Caution 1. For an external reset, input a low level for 10 µs or more to the RESETB pin.
To perform an external reset upon power application, input a low level to the RESETB pin, turn power on, continue to input a low level to the pin for 10 µs or more within the operating voltage range shown in 2.4 AC Characteristics of the R9A02G015 Data Sheet (R19DS0101E), and then input a high level to the pin.
Caution 2. During generation of a reset signal, the high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input become invalid.
Caution 3. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.
- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the on-chip pull-up resistance).**
- **Ports other than P40: High-impedance during the reset period or after receiving a reset signal.**

Figure 16 - 1 Block Diagram of Reset Function



Caution An LVD circuit internal reset does not reset the LVD circuit.

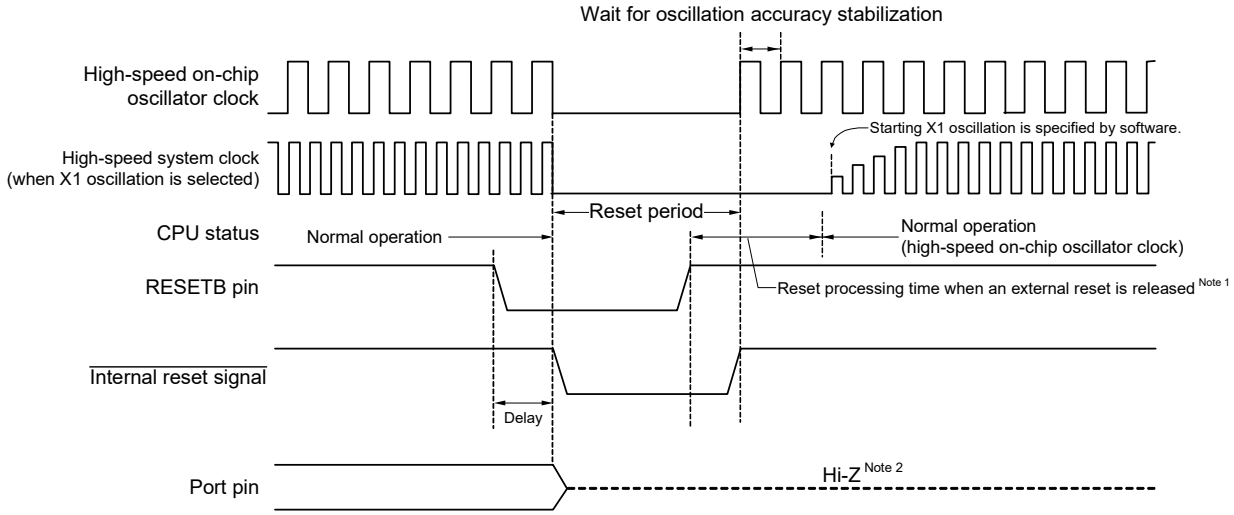
Remark 1. LVIM: Voltage detection register

Remark 2. LVIS: Voltage detection level register

16.1 Timing of Reset Operation

The R9A02G015 is reset by input of the low level on the RESETB pin and released from the reset state by input of the high level on the RESETB pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

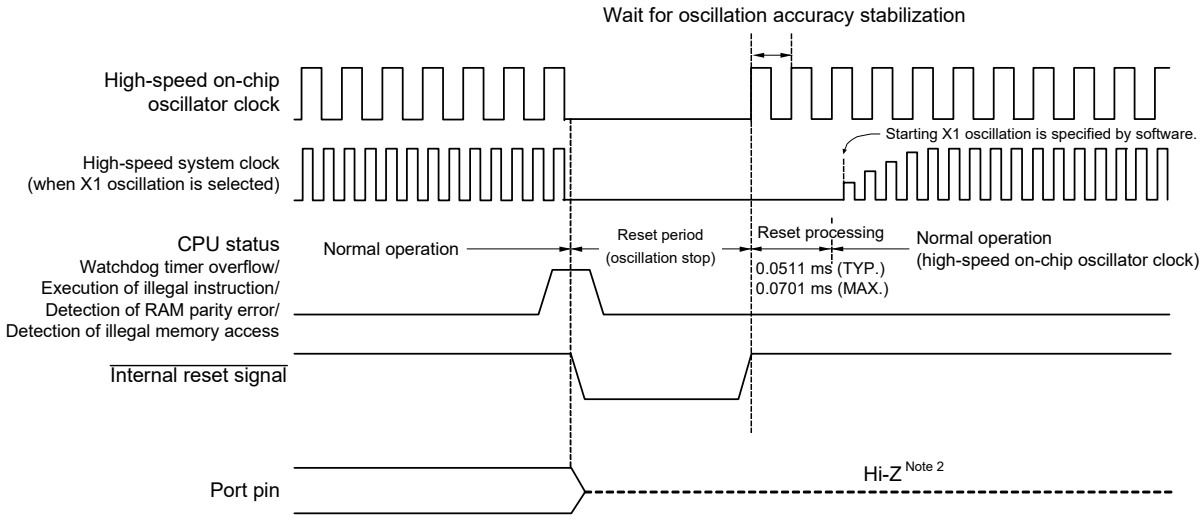
Figure 16 - 2 Timing of Reset by RESETB Input



(**Notes** and **Caution** are listed on the next page.)

Release from the reset state is automatic in the case of a reset due to a watchdog timer overflow, execution of an illegal instruction, detection of a RAM parity error, or detection of illegal memory access. After reset processing, program execution starts with the high-speed on-chip oscillator clock as the operating clock.

Figure 16 - 3 Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory Access



(**Notes** and **Caution** are listed on the next page.)

Note 1. Reset times (times for release from the external reset state)

After the first release of the POR: 0.672 ms (TYP.), 0.832 ms (MAX.) when the LVD is in use.
0.399 ms (TYP.), 0.519 ms (MAX.) when the LVD is off.

After the second release of the POR: 0.531 ms (TYP.), 0.675 ms (MAX.) when the LVD is in use.
0.259 ms (TYP.), 0.362 ms (MAX.) when the LVD is off.

After power is supplied, a voltage stabilization waiting time of about 0.99 ms (TYP.) and up to 2.30 ms (MAX.) is required before reset processing starts after release of the external reset.

Note 2. The state of P40 is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset or after receiving a reset signal (connected to the on-chip pull-up resistance).

Reset by POR and LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ after the reset. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts. For details, see **CHAPTER 17 POWER-ON-RESET CIRCUIT** or **CHAPTER 18 VOLTAGE DETECTOR**.

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

Table 16 - 1 Operation Statuses During Reset Period

Item		During Reset Period
System clock		Clock supply to the CPU is stopped.
Main system clock	f_{HOCO}	Operation stopped
	f_X	Operation stopped (the X1 and X2 pins are input port mode)
	f_{EX}	Clock input invalid (the pin is input port mode)
	f_{PLL}	Operation stopped
f_{IL}		Operation stopped
CPU		
Code flash memory		Operation stopped
Data flash memory		Operation stopped
RAM		Operation stopped
Port (latch)		High impedance Note
Timer array unit		Operation stopped
12-bit Interval timer		
Watchdog timer		
Clock output/buzzer output		
A/D converter		
Serial array unit (SAU)		
Serial interface (IICA)		
USB		
Power-on-reset function		Detection operation possible
Voltage detection function		Operation is possible in the case of an LVD reset and stopped in the case of other types of reset.
External interrupt		Operation stopped
CRC operation function	High-speed CRC	
	General-purpose CRC	
RAM parity error detection function		
RAM guard function		
SFR guard function		
Illegal-memory access detection function		

Note P40 become the following state.

- P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset (connected to the internal pull-up resistor).

Remark f_{HOCO}: High-speed on-chip oscillator clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock

Table 16 - 2 Hardware Statuses After Reset Acknowledgment

Hardware		After Reset Acknowledgment Note
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Remark For the state of the special function register (SFR) after receiving a reset signal, see **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

16.2 Register for Confirming Reset Source

16.2.1 Reset control flag register (RESF)

Many internal reset generation sources exist in the R9A02G015. The reset control flag register (RESF) is used to store which source has generated the reset request.

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

RESETB input, reset by power-on-reset (POR) circuit, and accessing SFR other than the RESF register after reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags.

Figure 16 - 4 Format of Reset control flag register (RESF)

Address: FFFA8H After reset: Undefined Note 1 R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

TRAP	Internal reset request by execution of illegal instruction Note 2
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

RPERF	Internal reset request t by RAM parity
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

IAWRF	Internal reset request t by illegal-memory access
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by voltage detector (LVD)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Note 1. The value after reset varies depending on the reset source. See **Table 16 - 3**.

Note 2. The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Caution 1. Do not read data by a 1-bit memory manipulation instruction.

Caution 2. When enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area at data access or the used RAM area + 10 bytes at execution of instruction from the RAM area.
Reset generation enables RAM parity error resets (RPERDIS = 0). For details, see 19.3.3 RAM parity error detection function.

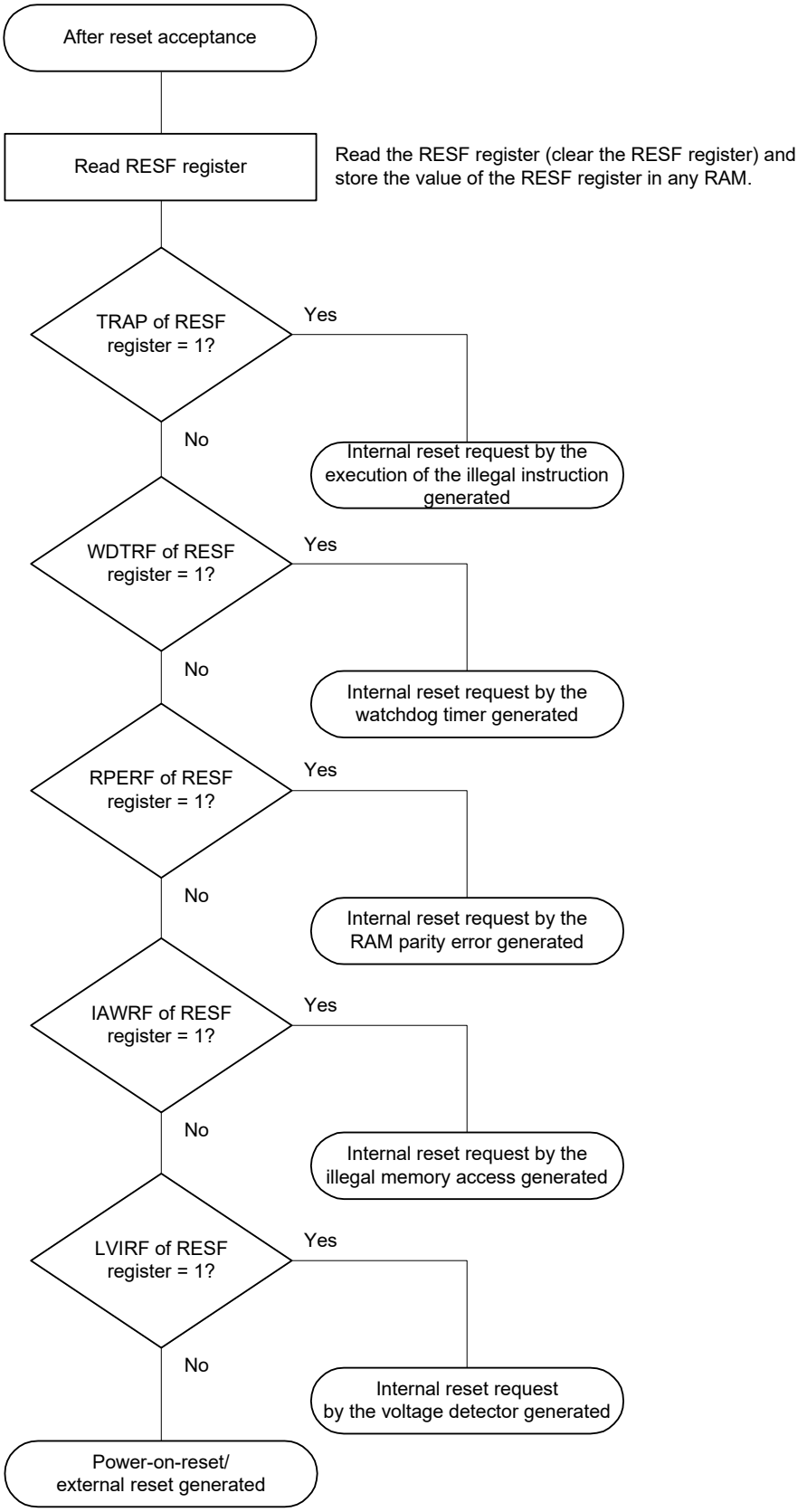
The status of the RESF register when a reset request is generated is shown in Table 16 - 3.

Table 16 - 3 RESF Register Status When Reset Request Is Generated

Reset Source / Flag	RESETB Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM parity error	Reset by illegal-memory access	Reset by LVD
TRAP	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held	Held
WDTRF			Held	Set (1)			
RPERF				Held	Set (1)		
IAWRF					Held	Set (1)	
LVIRF						Held	Set (1)

Accessing SFR other than the RESF register after reading the RESF register by an 8-bit memory manipulation instruction clears the RESF register automatically. Figure 16 - 5 shows the procedure for checking a reset source.

Figure 16 - 5 Example of Procedure for Checking Reset Source



16.2.2 Peripheral reset control register 0 (PRR0)

This register is used for individual reset control of each peripheral hardware.

The R9A02G015 controls reset and reset release of each peripheral hardware supported by the PRR0 register.

Figure 16 - 6 Format of Peripheral reset control register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PRR0	IICA2RES Note	IICA1RES	ADCRES	IICA0RES	0	SAU0RES	0	TAU0RES

PRR0n	Peripheral reset control on each peripheral hardware
0	Peripheral reset release
1	Peripherals reset state

Note This bit is incorporated with R9A02G0151, but is not incorporated with R9A02G0150.

Remark n = 0, 2, 4, 5, 6 or 7

The controlled hardware by each bit are as follows.

Table 16 - 4 Controlled Hardware by Each Bit in PRR0

Bit	Bit Name	Controlled Hardware
0	TAU0RES	Timer array unit (Unit 0)
2	SAU0RES	Serial array unit (Unit 0)
4	IICA0RES	Serial interface IICA0
5	ADCRES	A/D Converter
6	IICA1RES	Serial interface IICA1
7	IICA2RES	Serial interface IICA2

16.2.3 Peripheral reset control register 2 (PRR2)

This register is used for individual reset control of each peripheral hardware.

The R9A02G015 controls reset and reset release of each peripheral hardware supported by the PRR2 register.

Figure 16 - 7 Format of Peripheral reset control register 2 (PRR2)

Address: F00FDH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
PRR2	TMKARES	0	0	0	0	0	0	0

PRR2n	Peripheral reset control on each peripheral hardware macro
0	Peripheral reset release
1	Peripherals reset state

Remark n = 7

The controlled hardware by each bit are as follows.

Table 16 - 5 Macro Controlled by Each Bit in PRR2

Bit	Bit Name	Controlled Hardware
7	TMKARES	12-bit interval timer

CHAPTER 17 POWER-ON-RESET CIRCUIT

17.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR}). Note that the reset state must be retained until the operating voltage becomes in the range defined in **2.4 AC Characteristics** of the R9A02G015 Data Sheet (R19DS0101E).
  This is done by utilizing the voltage detection circuit or controlling the externally input reset signal.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR}), generates internal reset signal when $V_{DD} < V_{PDR}$.
  Note that, after power is supplied, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the operation voltage falls below the range defined in **2.4 AC Characteristics** of the R9A02G015 Data Sheet (R19DS0101E). When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Caution **If an internal reset signal is generated in the power-on-reset circuit, the reset control flag register (RESF) and the power-on-reset status register (PORSR) are cleared (00H).**

Remark 1. The R9A02G015 incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access.
For details of the RESF register, see **CHAPTER 16 RESET FUNCTION**.

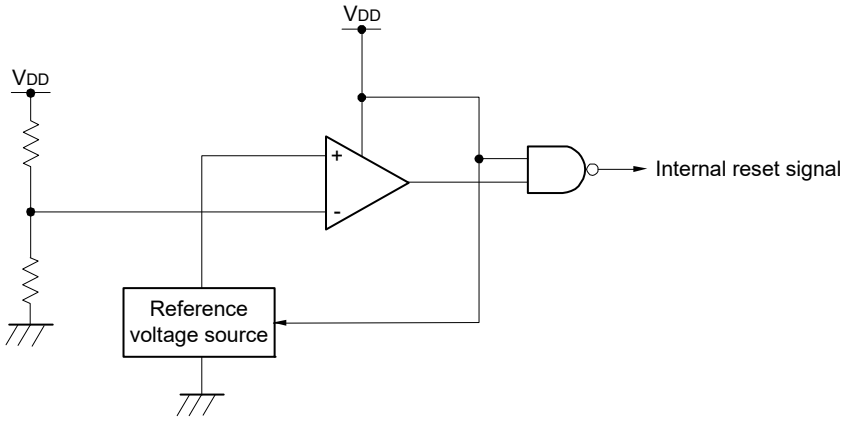
Remark 2. The occurrence of an internal reset in the power-on-reset circuit can be checked by the power-on reset status register (PORSR). For details on the PORSR register, refer to **CHAPTER 16 RESET FUNCTION**.

Remark 3. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **2.6.3 POR circuit characteristics** in the R9A02G015 Data Sheet (R19DS0101E).

17.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 17 - 1.

Figure 17 - 1 Block Diagram of Power-on-reset Circuit

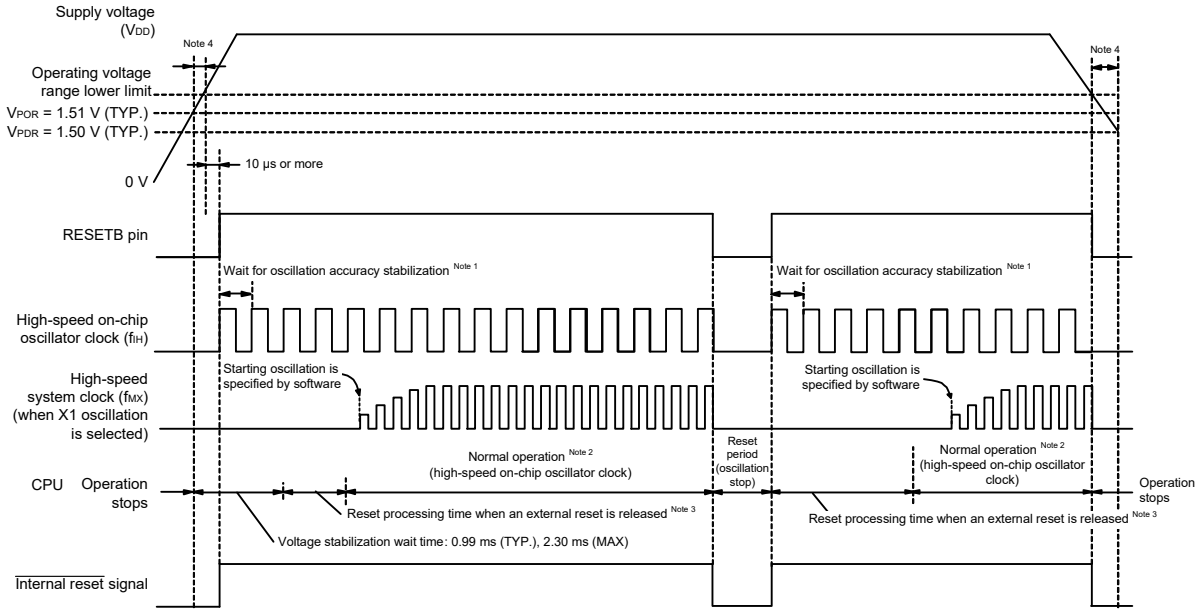


17.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown next.

Figure 17 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)

(1) When using an external reset by the RESETB pin



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.

Note 3. The time until normal operation starts includes the following reset processing time when the external reset is released (after the first release of POR) after the RESETB signal is driven high (1) as well as the voltage stabilization wait time after VPOR (1.51 V, TYP.) is reached.

Reset processing time when the external reset is released is shown below.

After the first release of POR: 0.672 ms (TYP.), 0.832 ms (MAX.) (when the LVD is in use)
0.399 ms (TYP.), 0.519 ms (MAX.) (when the LVD is off)

Reset processing time when the external reset is released after the second release of POR is shown below.

After the second release of POR: 0.531 ms (TYP.), 0.675 ms (MAX.) (when the LVD is in use)
0.259 ms (TYP.), 0.362 ms (MAX.) (when the LVD is off)

Note 4. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **2.4 AC Characteristics** of the R9A02G015 Data Sheet (R19DS0101E). This is done by controlling the externally input reset signal.

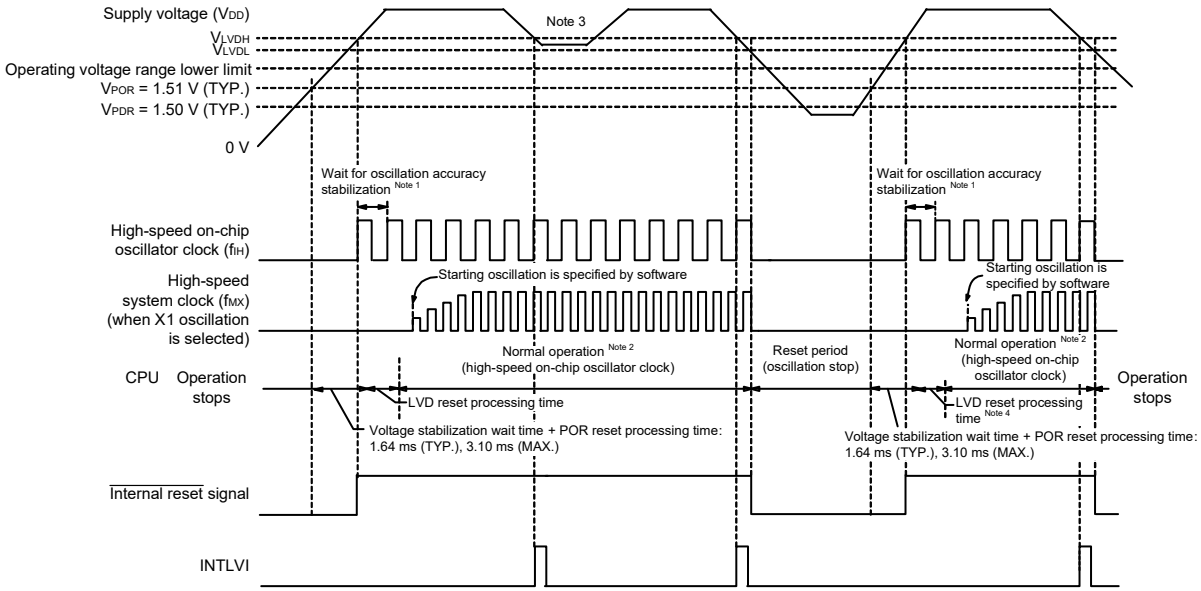
After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Caution **For power-on reset, be sure to use the externally input reset signal on the RESETB pin when the LVD is off. For details, see CHAPTER 18 VOLTAGE DETECTOR.**

Figure 17 - 3 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)

(2) LVD is interrupt & reset mode (option byte 000C1: LVIMDS1, LVIMDS0 = 1, 0)



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.

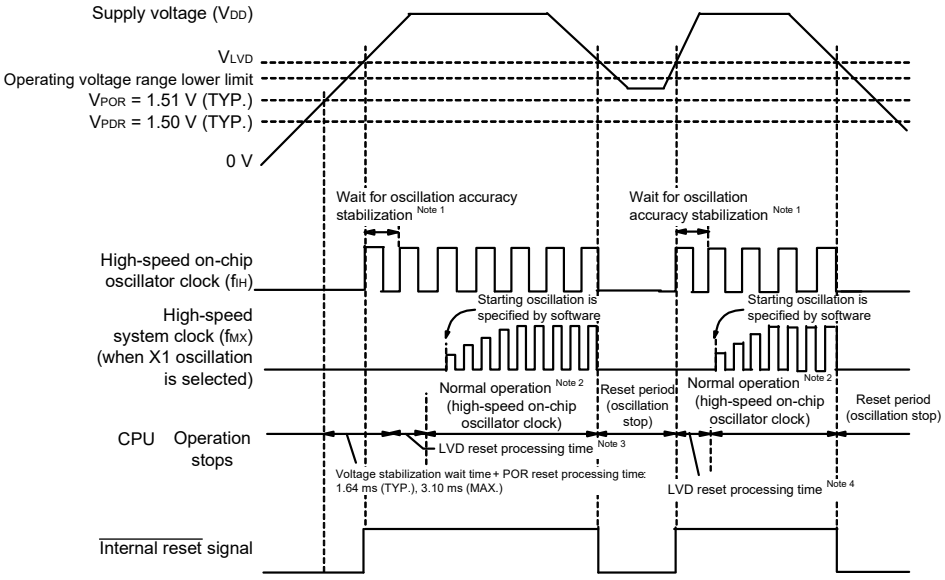
Note 3. After the interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 18 - 9 Setting Procedure for Operating Voltage Check and Reset**, taking into consideration that the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

Note 4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, TYP.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (MAX.)

Remark V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 17 - 4 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (3/3)

(3) LVD reset mode (option byte 000C1H: LVIMDS1, LVIMDS0 = 1, 1)



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.

Note 3. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVD}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, TYP.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (MAX.)

Note 4. When the power supply voltage is below the lower limit for operation and the power supply voltage is then restored after an internal reset is generated only by the voltage detector (LVD), the following LVD reset processing time is required after the LVD detection level (V_{LVD}) is reached.
LVD reset processing time: 0.0511 ms (TYP.), 0.0701 ms (MAX.)

Remark 1. V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Remark 2. When the LVD interrupt mode is selected (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1), the time until normal operation starts after power is turned on is the same as the time specified in Note 3 of Figure 17 - 4 (3).

CHAPTER 18 VOLTAGE DETECTOR

18.1 Functions of Voltage Detector

The operation mode and detection voltages (V_{LVDH}, V_{LVDL}, V_{LVD}) for the voltage detector is set by using the option byte (000C1H). The detection voltages can be reset using the LVIS register. The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage (V_{LVDH}, V_{LVDL}) can be selected as one of 14 levels (For details, see **18.3.2 Voltage detection level register (LVIS)** and **CHAPTER 21 OPTION BYTE**).
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **2.4 AC Characteristics** of the R9A02G015 Data Sheet (R19DS0101E). This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)
The two detection voltages (V_{LVDH}, V_{LVDL}) are selected by the option byte 000C1H. The high-voltage detection level (V_{LVDH}) is used for releasing resets and generating interrupts. This level is also used for generating resets. The low-voltage detection level (V_{LVDL}) is used for generating resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for triggering and ending resets. The detection voltages can be reset using the LVIS register.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for generating interrupts/reset release. The detection voltages can be reset using the LVIS register.

The reset and internal interrupt signals are generated in each mode as follows.

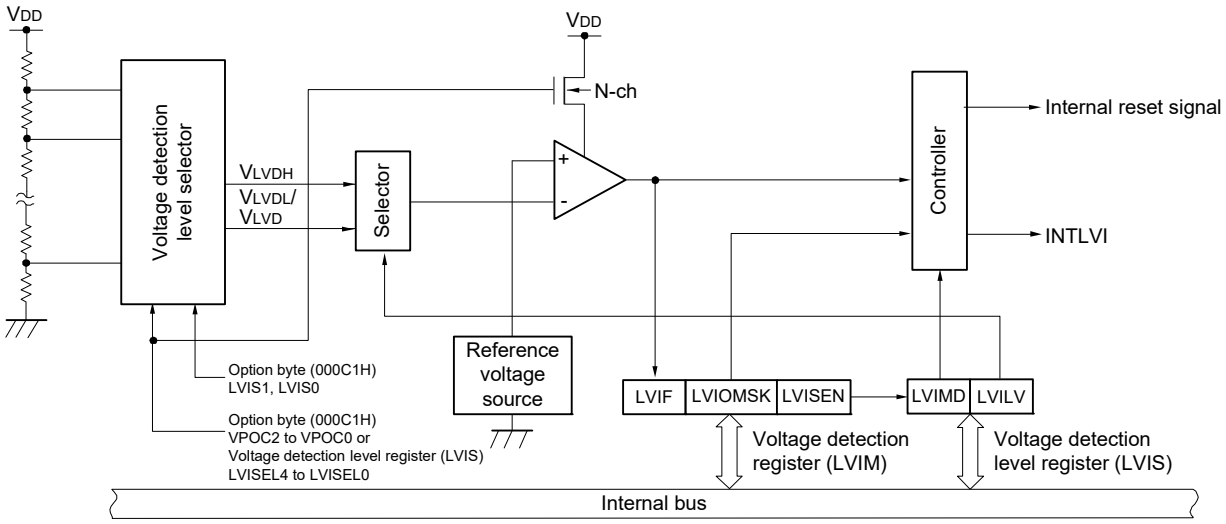
Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an internal reset by detecting $V_{DD} < V_{LVD}$.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq V_{LVD}$. Releases the LVD internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ after the LVD internal reset is released.

While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).
Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 16 RESET FUNCTION**.

18.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 18 - 1.

Figure 18 - 1 Block Diagram of Voltage Detector



18.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

18.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the LVD output mask status.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 18 - 2 Format of Voltage detection register (LVIM)

Address: FFFA9H After reset: 00H Note 1 R/W Note 2

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN	0	0	0	0	0	LVIOMSK	LVIF

LVISEN	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling of rewriting the LVIS register (LVIOMSK = 0 (Mask of LVD output is invalid)
1	Enabling of rewriting the LVIS register Note 3 (LVIOMSK = 1 (Mask of LVD output is valid)

LVIOMSK	Mask status flag of LVD output
0	Mask of LVD output is invalid
1	Mask of LVD output is valid Note 3

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVD}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVD})

Note 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value. In other reset, LVISEN is cleared to 0.

Note 2. Bits 0 and 1 are read-only.

Note 3. The LVIOMSK bit is automatically set to 1 for the following periods and generation of an LVD reset or interrupt is masked.

- Period when LVISEN = 1
  In either of the following cases, generation of an LVD reset or interrupt is masked only in interrupt & reset mode.
- Wait time until the LVD detection voltage stabilizes after an LVD interrupt is generated
- Wait time until the LVD detection voltage stabilizes after the value of the LVILV bit is changed

18.3.2 Voltage detection level register (LVIS)

This register selects the voltage detection level. The minimum supply voltage (LVD detection voltage) and LVD detection level settings that are set by the user option byte can be changed by software.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation input sets this register to Note 1.

Caution Do not change the detection voltage in interrupt & reset mode.

Figure 18 - 3 Format of Voltage detection level register (LVIS)

Address: FFFAAH After reset: Note 1 R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD Note 2	0	LVISEL4 Note 6	LVISEL3	LVISEL2	LVISEL1	LVISEL0	LVILV Note 2

LVIMD Note 2	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVISEL4 Note 6	LVISEL3	LVISEL2	Minimum operating voltage (typical falling value) Note 5
0	0	1	1.84 V
0	1	0	2.45 V
0	1	1	2.75 V
1	1	1	1.53 V (LVD OFF)
Other than above			Setting prohibited

LVISEL1	LVISEL0	LVD detection level setting Note 5
0	0	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 + 1.2 V Note 3
0	1	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 + 0.2 V Note 3
1	0	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 + 0.1 V Note 3
1	1	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 Note 4

LVILV Note 2	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVD})

Note 1. The reset value changes depending on the setting of the option byte.
After a reset is released, the values of VPOC2 to VPOC0 and LVIS1 and LVIS0 in the user option byte are reflected in LVISEL4 to LVISEL2, LVISEL1, and LVISEL0, respectively.
The reset values of LVIMD and LVIVL are set as follows.
When LVIMDS1, LVIMDS0 in the option byte = 1, 0: LVIMD = 0, LVILV = 0
When LVIMDS1, LVIMDS0 in the option byte = 1, 1: LVIMD = 1, LVILV = 1
When LVIMDS1, LVIMDS0 in the option byte = 0, 1: LVIMD = 0, LVILV = 1

Note 2. Writing “0” can only be allowed in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set LVIMD and LVILV in other cases. The value is switched automatically when reset or interrupt is generated in the interrupt & reset mode.

Note 3. Indicates an approximate detection value. For details on the actual detection voltage, refer to **2.6.4 LVD circuit characteristics** in the R9A02G015 Data Sheet (R19DS0101E).

Note 4. Cannot be selected when LVIMDS1 and LVIMDS0 = 1 and 0.

Note 5. When changing LVISEL4 to LVISEL0 to use two or more LVD detection voltages, the setting value that indicates the highest voltage value among the LVD detection voltages to be used should be set in the VPOC2 to VPOC0 bits and LVIS1 and LVIS0 bits before using the voltages.

Note 6. Rewriting LVISEL4 is prohibited. Keep the initial value unchanged.

Caution 1. When rewriting the LVIMD and LVILV bits, use the procedure shown in Figure 18 - 9.

Caution 2. Specify the LVD operation mode and initial detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H. Figure 18 - 4 shows the format of the user option byte (000C1H/010C1H). For details about the option byte, see CHAPTER 21 OPTION BYTE.

Figure 18 - 4 Format of User Option Byte (000C1H/010C1H) (1/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt & reset mode)

Detection Voltage			Option Byte Setting Value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
2.92 V	2.86 V	2.75 V	0	1	1	1	0	1	0
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
—			Setting of values other than above is prohibited.						

• LVD setting (reset mode)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.81 V	2.75 V	0	1	1	1	1	1	1
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Remark 1. For details on the LVD circuit, see **CHAPTER 18 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a TYP. value. For details, see **2.6.4 LVD circuit characteristics** in the R9A02G015 Data Sheet (R19DS0101E).

(**Cautions** are listed on the next page.)

Figure 18 - 4 Format of User Option Byte (000C1H/010C1H) (2/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt mode)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.81 V	2.75 V	0	1	1	1	1	0	1
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Setting of values other than above is prohibited.						

• LVD off setting (external reset input from the RESETB pin is used)

Detection voltage		Option byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution 1. Set bit 4 to 1.

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 2.4 AC Characteristics of the R9A02G015 Data Sheet (R19DS0101E). This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

Remark 1. ×: Don't care

Remark 2. For details on the LVD circuit, see **CHAPTER 18 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a TYP. value. For details, see **2.6.4 LVD circuit characteristics** in the R9A02G015 Data Sheet (R19DS0101E).

18.4 Operation of Voltage Detector

18.4.1 When used as reset mode

Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the initial detection voltage (V_{LVD}) by using the option byte 000C1H. The detection voltages can be reset using the LVIS register.

The operation is started in the following initial setting state when the reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 81H.
  See **18.3.2 Voltage detection level register (LVIS)** for details on the initial value of the voltage detection level register (LVIS).
  Bit 7 (LVIMD) is 1 (reset mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

• Operation in LVD reset mode

In the reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).

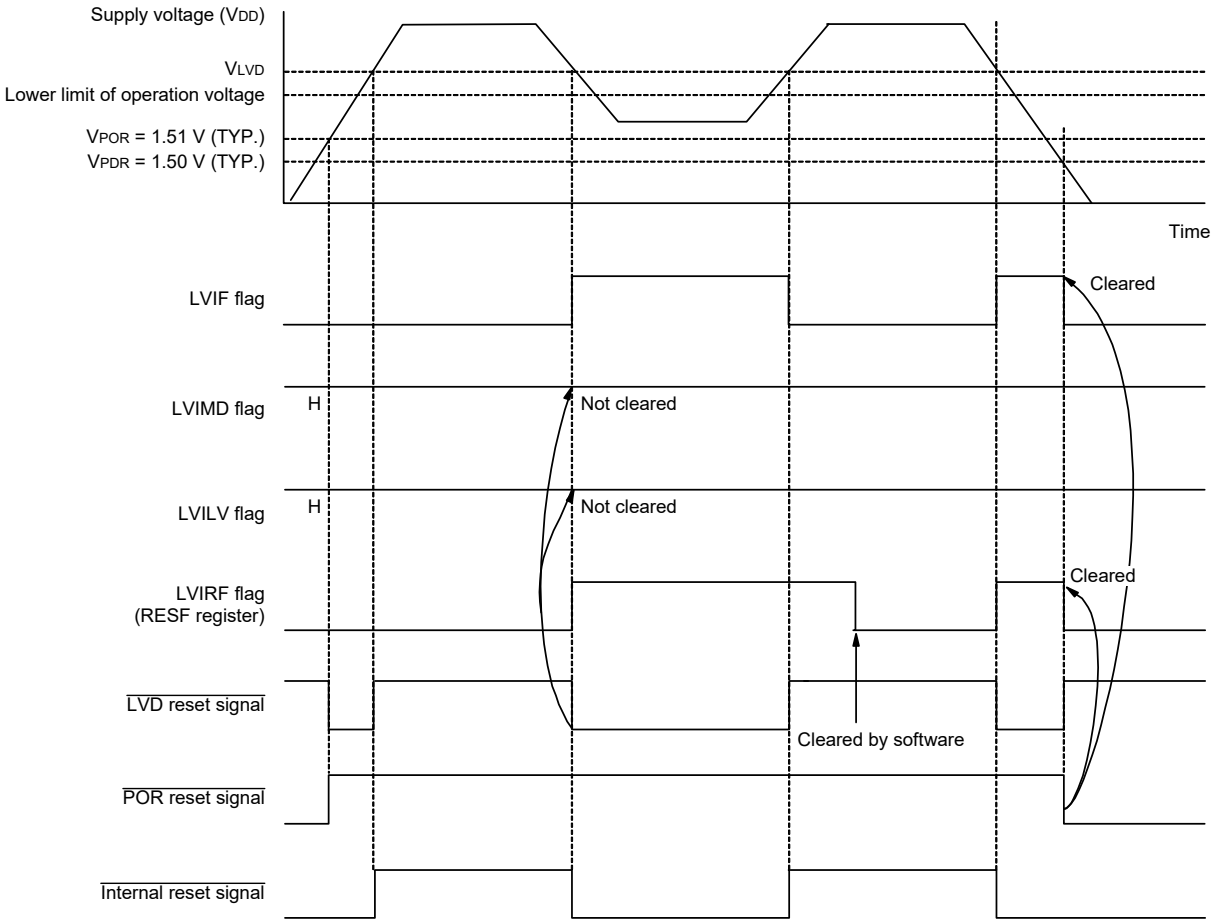
At the fall of the operating voltage, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}).

The reset release voltage when an LVD reset is generated is the detection voltage set by the option byte or detection voltage set by the LVIS register, whichever is higher. The state of an internal reset by the LVD is retained until the supply voltage exceeds the voltage detection level.

The reset release voltage used for resets other than an LVD reset is the same voltage detection level set by the option byte.

Figure 18 - 5 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 18 - 5 Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

18.4.2 When used as interrupt mode

Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the initial detection voltage (V_{LVD}) by using the option byte 000C1H. The detection voltages can be reset using the LVIS register.

The operation is started in the following initial setting state when the interrupt mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- See **18.3.2 Voltage detection level register (LVIS)** for details on the initial value of the voltage detection level register (LVIS).
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

• Operation in LVD interrupt mode

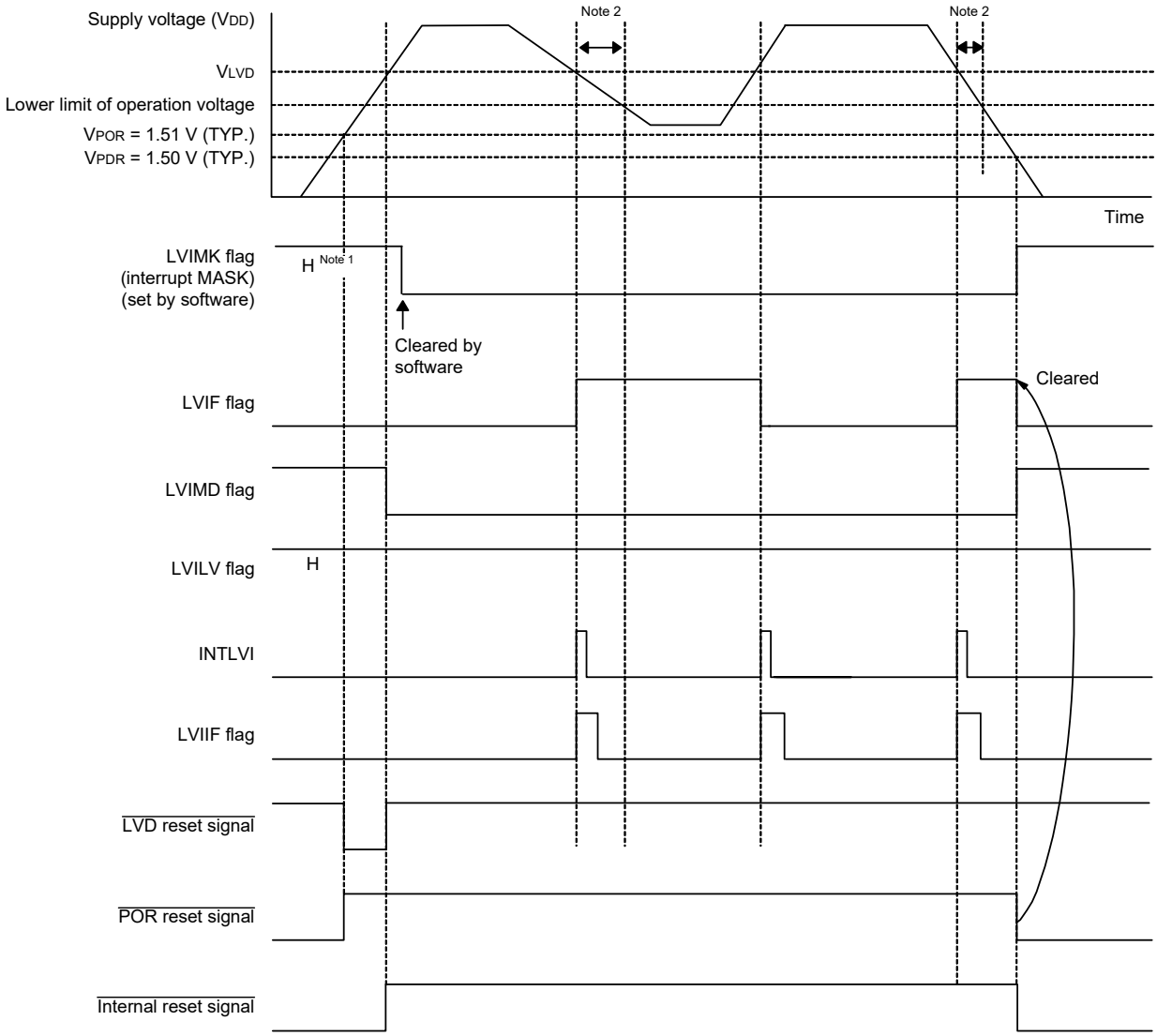
In the interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1), the state of an internal reset by LVD is retained immediately after a reset until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied (after the first release of the POR). The LVD internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).

After the LVD internal reset is released, an interrupt request signal (INTLVI) by the LVD is generated when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **2.4 AC Characteristics** of the R9A02G015 Data Sheet (R19DS0101E). When restarting the operation, make sure that the operation voltage has returned within the range of operation.

After the LVISEN bit is set to 1 (LVD is masked) by changing the detection level, if the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}) when LVISEN is set to 0, an interrupt request signal (INTLVI) by the LVD is generated.

Figure 18 - 6 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 18 - 6 Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **2.4 AC Characteristics** of the R9A02G015 Data Sheet (R19DS0101E). When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

18.4.3 When used as interrupt and reset mode

Specify the operation mode (the interrupt & reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte 000C1H. Do not manipulate the detection voltage using the LVIS register.

The operation is started in the following initial setting state when the interrupt & reset mode is set.

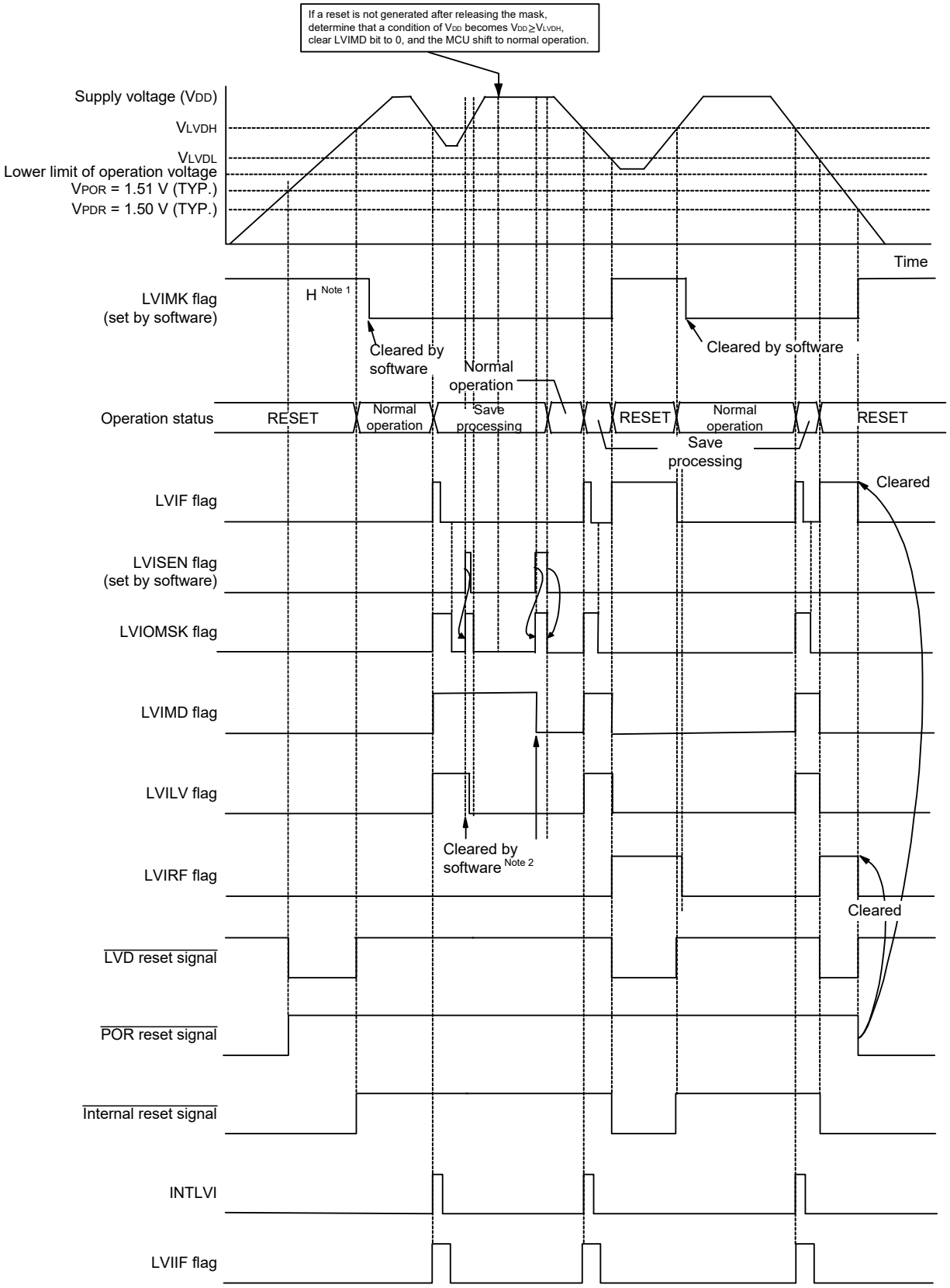
- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- See **18.3.2 Voltage detection level register (LVIS)** for details on the initial value of the voltage detection level register (LVIS).
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 0 (high-voltage detection level: V_{LVDH}).

• Operation in LVD interrupt & reset mode

In the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}). An interrupt request signal by LVD (INTLVI) is generated and arbitrary save processing is performed when the supply voltage (V_{DD}) falls below the high-voltage detection level (V_{LVDH}). After that, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the low-voltage detection level (V_{LVDL}). After INTLVI is generated, an interrupt request signal is not generated even if the supply voltage becomes equal to or higher than the high-voltage detection voltage (V_{LVDH}) without falling below the low-voltage detection voltage (V_{LVDL}). To use the LVD reset & interrupt mode, perform the processing according to **Figure 18 - 9 Setting Procedure for Operating Voltage Check and Reset**.

Figures 18 - 7 and 18 - 8 show the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 18 - 7 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



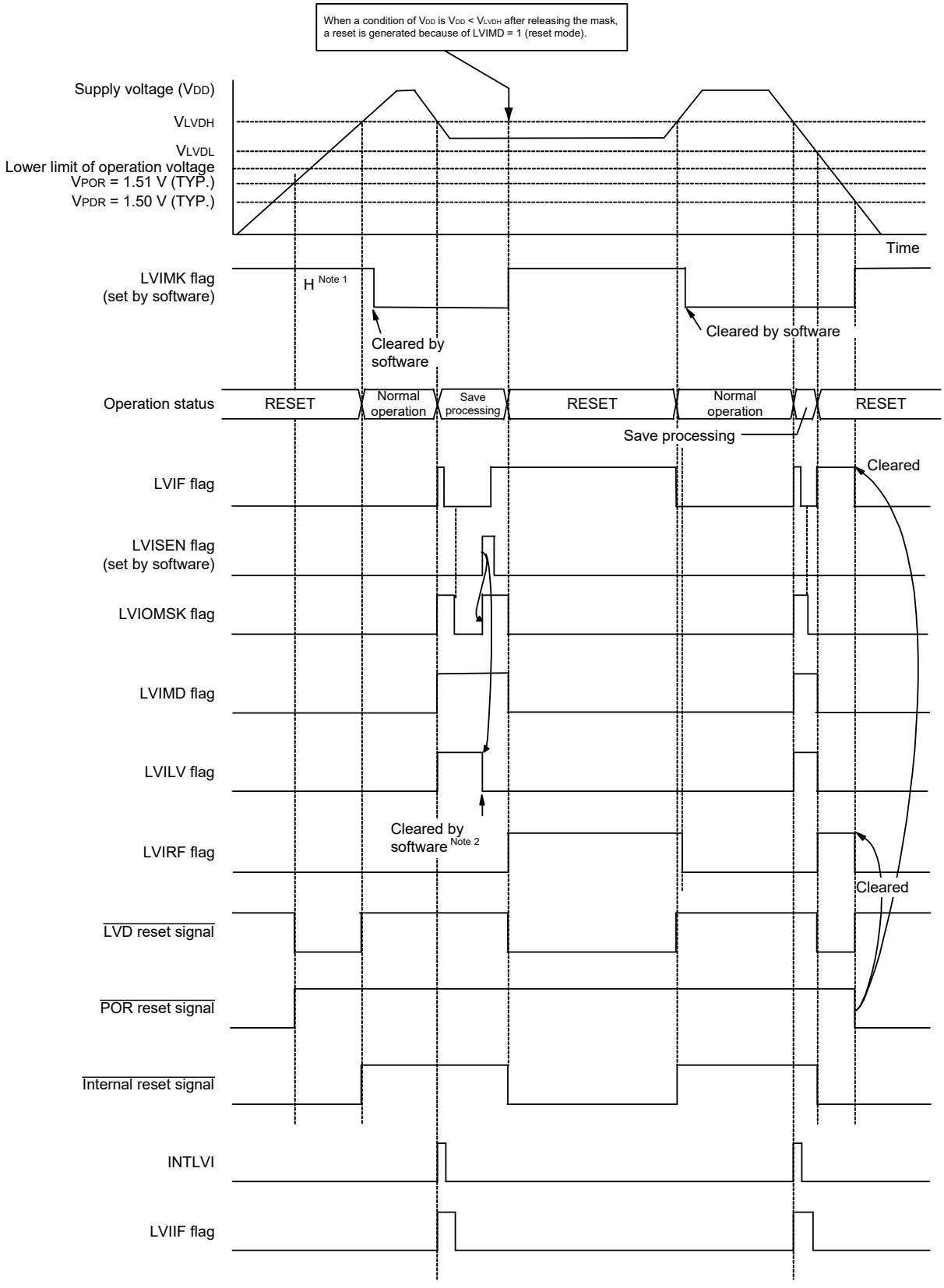
(**Notes** and **Remark** are listed on the next page.)

Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. After an interrupt is generated, perform the processing according to Figure 18 - 9 Setting Procedure for Operating Voltage Check and Reset in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{POR}: POR power supply fall detection voltage

Figure 18 - 8 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



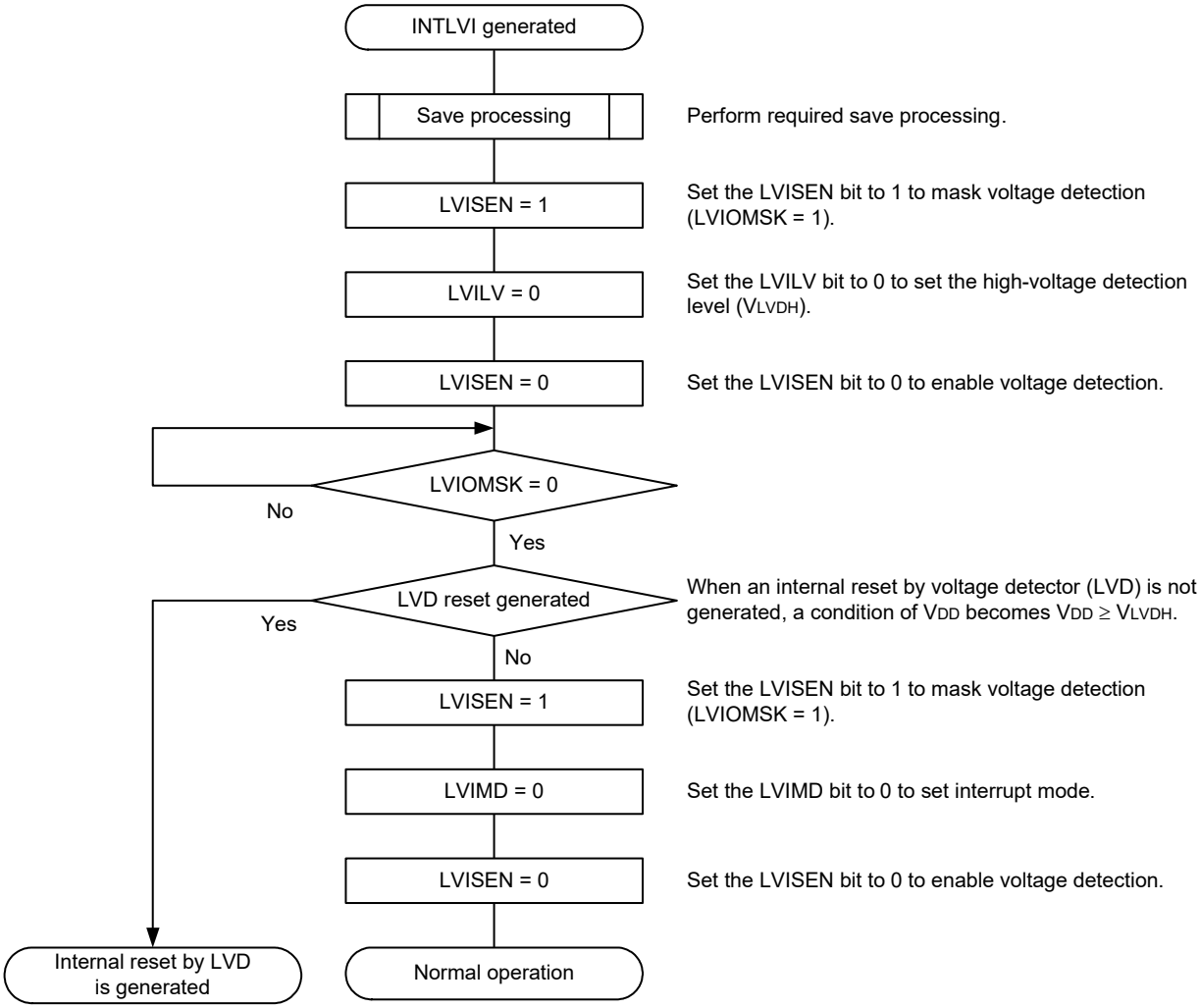
(**Notes** and **Remark** are listed on the next page.)

Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. After an interrupt is generated, perform the processing according to Figure 18 - 9 Setting Procedure for Operating Voltage Check and Reset in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 18 - 9 Setting Procedure for Operating Voltage Check and Reset



18.5 Changing of LVD Detection Voltage Setting

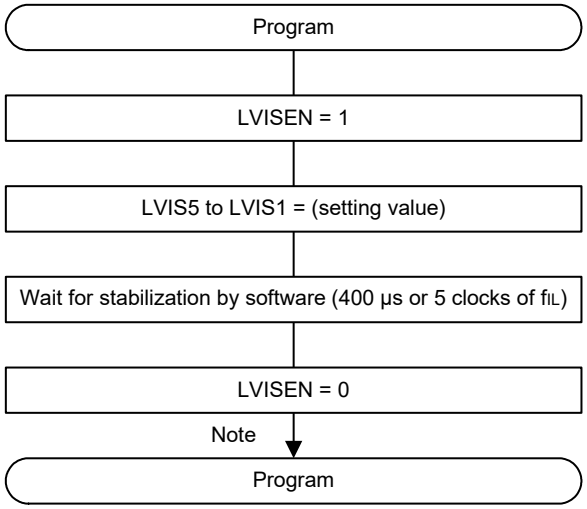
To change the LVD detection voltage by software, use the following procedure.

The LVD detection voltage can be changed in interrupt mode and reset mode.

In interrupt & reset mode, the value of the LVD detection voltage cannot be changed. Keep the initial value (set value in the option byte) unchanged.

To use two or more LVD detection voltages by changing LVISEL4 to LVISEL0 in the LVIS register by software, the highest voltage value of the used LVD detection voltages must be specified in the VPOC2 to VPOC0, LVIS1, and LVIS0 bits in the option byte (000C1H).

Figure 18 - 10 Changing of LVD Detection Voltage Setting

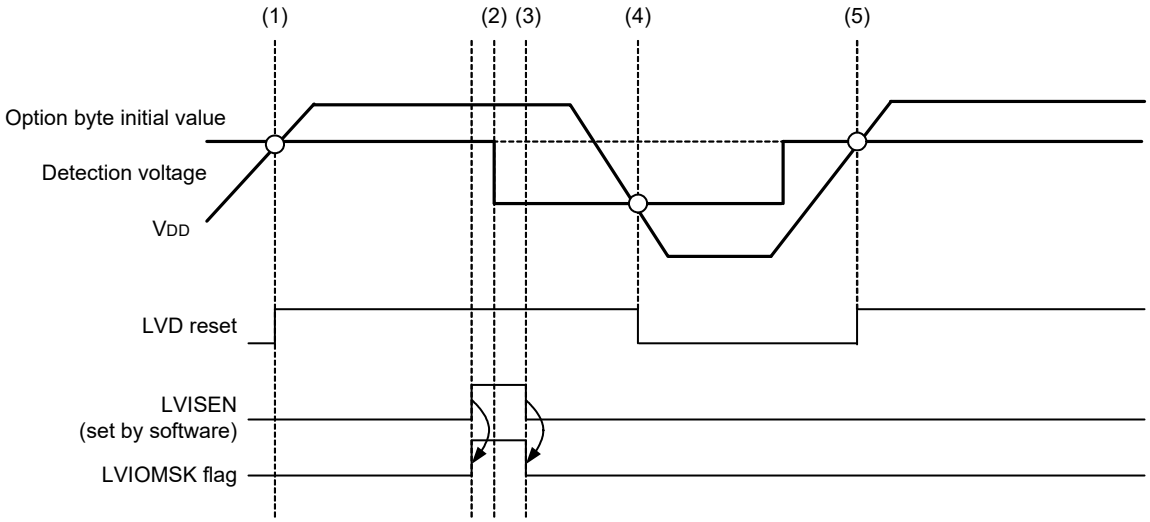


Note After LVISEN is set to 0, LVD is detected if $V_{LVD} > V_{DD}$, and a reset/interrupt is generated.

18.5.1 Changing of LVD detection voltage setting in LVD reset mode

Figure 18 - 11 shows an Example of Timing for Changing LVD Detection Voltage Setting in LVD Reset Mode.

Figure 18 - 11 Example of Timing for Changing LVD Detection Voltage Setting in LVD Reset Mode



Operation

(1) When the supply voltage rises, the detection voltage set by the option byte is used to release the reset.
(2) The value of the LVIS register is changed.
(3) Waiting for stabilization by software is completed (400 μs or five fIL clock cycles after (2))
(4) At LVD detection (falling), the detection voltage set by the LVIS register
(5) At the LVD reset release (rising), the detection voltage set by the option byte

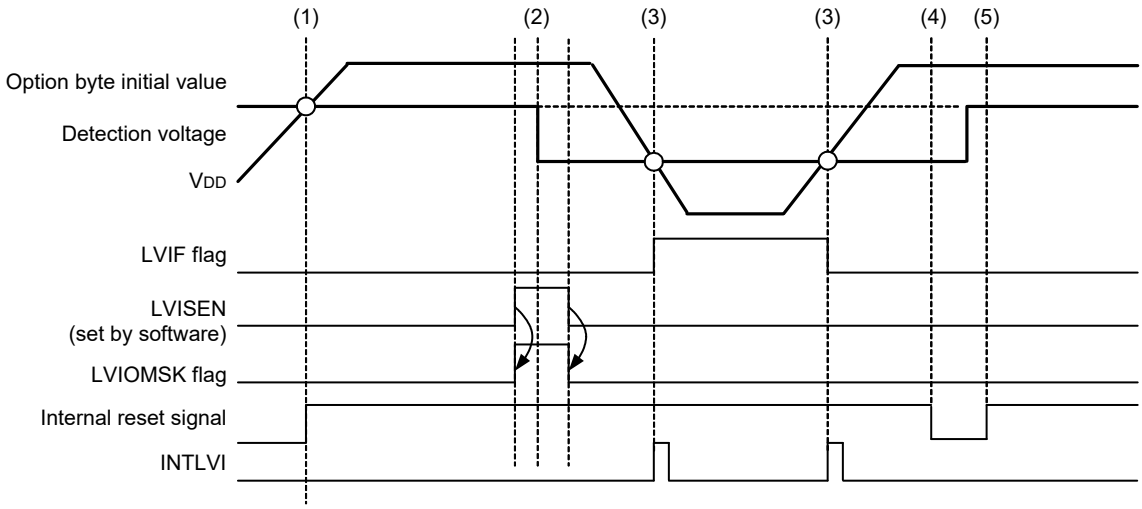
When changing the LVD detection voltage setting, note the following.

Caution The value of the reset release voltage in LVD reset mode is set to the set value in the option byte.

18.5.2 Changing of LVD detection voltage setting in LVD interrupt mode

Figure 18 - 12 shows an Example of Timing for Changing LVD Detection Voltage Setting in LVD Interrupt Mode.

Figure 18 - 12 Example of Timing for Changing LVD Detection Voltage Setting in LVD Interrupt Mode



Operation

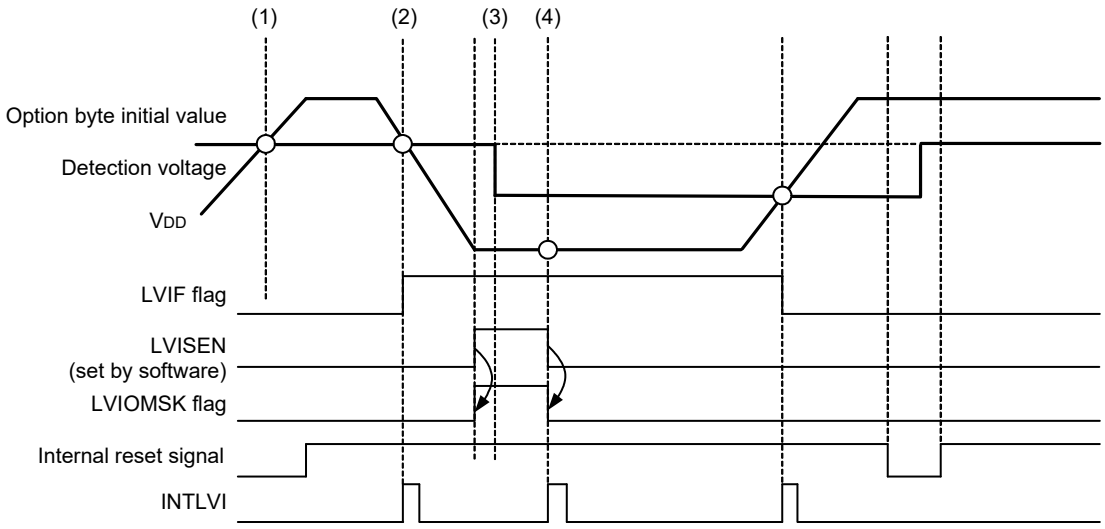
(1) When the supply voltage rises, the detection voltage set by the option byte is used to release the reset.
(2) The value of the LVIS register is changed.
(3) At LVD detection (falling and rising), the detection voltage set by the LVIS register
(4) An internal reset is generated.
(5) The voltage value is changed to the set value in the option byte again when the internal reset is released.

When changing the LVD detection voltage setting, note the following.

Caution 1. Immediately after all resets are generated, the LVD internal reset retains its reset state until $V_{DD} \geq V_{LVD}$ (set value in the option byte). The LVD internal reset is released when $V_{DD} \geq V_{LVD}$ is detected (set value in the option byte).
After that, an interrupt request signal (INTLVI) is generated when $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ is detected.

Caution 2. If the LVD set voltage is changed by setting LVISEL4 to LVISEL0 in the LVIS register while $V_{DD} < V_{LVD}$, an LVD interrupt is generated when the masking is released (LVISEN = 0). See Figure 18 - 13.

Figure 18 - 13 Example of Timing for Changing LVD Detection Voltage Using LVDIS When $V_{DD} < V_{LVD}$



Operation

(1) When the supply voltage rises, the detection voltage set by the option byte is used to release the reset.

(2) At LVD detection (falling), the detection voltage set by the option byte

(3) The value of the LVIS register is changed.

(4) If $V_{DD} < V_{LVD}$ at the same time the masking is released, an interrupt is generated.

18.6 Cautions for Voltage Detector

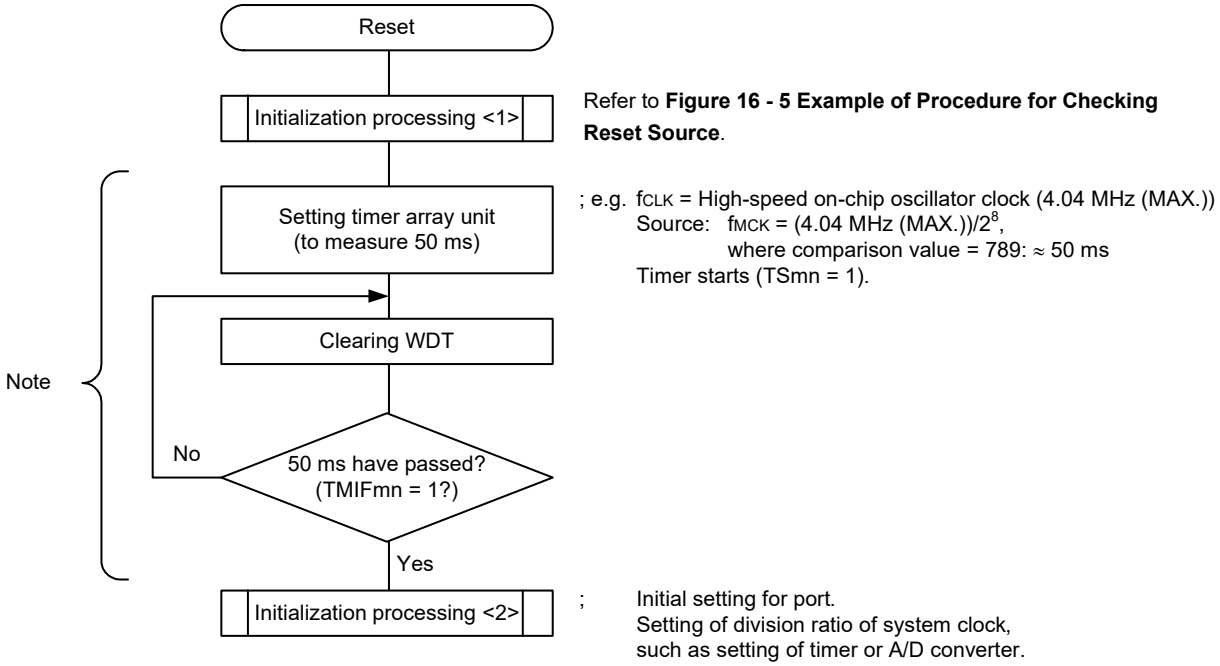
(1) Voltage fluctuation when power is supplied

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVD detection voltage, the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the R9A02G015 can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 18 - 14 Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage



Note If reset is generated again during this period, initialization processing <2> is not started.

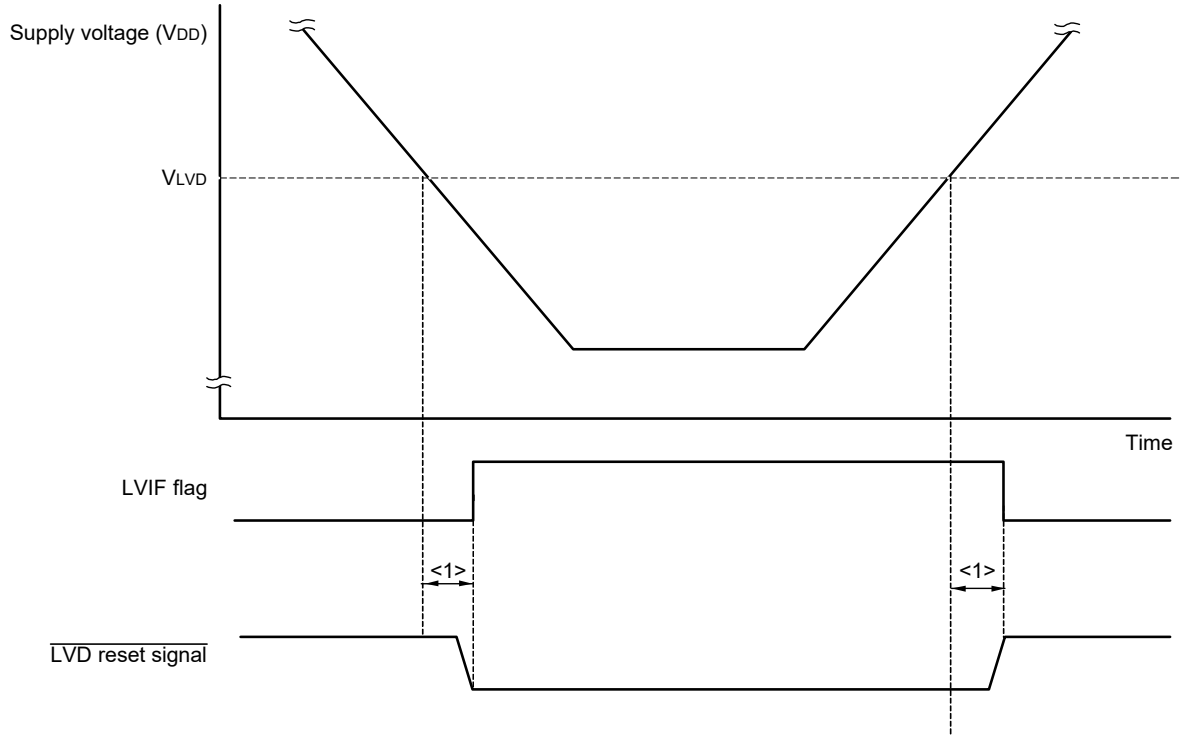
Remark m = 0
n = 0 to 3

(2) Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVD detection voltage (V_{LVD}) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage (V_{LVD}) ≤ supply voltage (V_{DD}) until the time LVD reset has been released (see **Figure 18 - 15**).

Figure 18 - 15 Delay from the time LVD reset source is generated until the time LVD reset has been generated or released



<1>: Detection delay (300 µs (MAX.))

(3) Power on when LVD is off

Use the external rest input via the RESETB pin when the LVD is off.

For an external reset, input a low level for 10 µs or more to the RESETB pin. To perform an external reset upon power application, input a low level to the RESETB pin, turn power on, continue to input a low level to the pin for 10 µs or more within the operating voltage range shown in **2.4 AC Characteristics** of the R9A02G015 Data Sheet (R19DS0101E), and then input a high level to the pin.

(4) Operating voltage fall when LVD is off or LVD interrupt mode is selected

When the operating voltage falls with the LVD is off or with the LVD interrupt mode is selected, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **2.4 AC Characteristics** of the R9A02G015 Data Sheet (R19DS0101E). When restarting the operation, make sure that the operation voltage has returned within the range of operation.

CHAPTER 19 SAFETY FUNCTIONS

19.1 Overview of Safety Functions

The following safety functions are provided in R9A02G015 to comply with the IEC60730 and IEC61508 safety standards.
These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

(1) Flash memory CRC operation function (high-speed CRC, general-purpose CRC)
This detects data errors in the flash memory by performing CRC operations.
Two CRC functions are provided in R9A02G015 that can be used according to the application or purpose of use.
- High-speed CRC: The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
- General CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) RAM parity error detection function
This detects parity errors when the RAM is read as data.

(3) RAM guard function
This prevents RAM data from being rewritten when the CPU freezes.

(4) SFR guard function
This prevents SFRs from being rewritten when the CPU freezes.

(5) Invalid memory access detection function
This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

(6) Frequency detection function
This uses the timer array unit to perform a self-check of the CPU/peripheral hardware clock frequency.

(7) A/D test function
This is used to perform a self-check of A/D converter by performing A/D conversion on the positive internal reference voltage, negative reference voltage, analog input channel (ANI), temperature sensor output, and internal reference voltage output.

(8) Digital output signal level detection function for I/O pins
When the I/O pins are output mode, the output level of the pin can be read.

Remark Refer to the IEC60730/60335 self-test library application notes (R01AN1062, R01AN1296) for the RL78 MCU Series, for more information on usage examples of the safety functions required to comply with the IEC60730 and IEC61508 safety standards.

19.2 Registers Used by Safety Functions

The safety functions use the following registers:

Register	Each Function of Safety Function
• Flash memory CRC control register (CRC0CTL) • Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (high-speed CRC)
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC operation function (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection function
• Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection function
• Timer input select register 0 (TIS0)	Frequency detection function
• A/D test register (ADTES)	A/D test function
• Port mode select register (PMS)	Digital output signal level detection function for I/O pins

The content of each register is described in 19.3 Operation of Safety Functions.

19.3 Operation of Safety Functions

19.3.1 Flash memory CRC operation function (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using CRC to do it. The high-speed CRC provided in R9A02G015 can be used to check the entire code flash memory area during the initialization routine. The high-speed CRC can be executed only when the program is allocated on the RAM and in the HALT mode of the main system clock.

The high-speed CRC performs an operation by reading 32-bit data per clock from the flash memory while stopping the CPU. This function therefore can finish a check in a shorter time (for example, 341 µs@24 MHz with 32-KB flash memory).

The CRC generator polynomial used complies with "$X^{16} + X^{12} + X^5 + 1$" of CRC-16-CCITT.

The high-speed CRC operates in MSB first order from bit 31 to bit 0.

Caution **The CRC operation result might differ during on-chip debugging because the monitor program is allocated.**

Remark The operation result is different between the high-speed CRC and the general CRC, because the general CRC operates in LSB first order.

19.3.1.1 Flash memory CRC control register (CRC0CTL)

This register is used to control the operation of the high-speed CRC ALU, as well as to specify the operation range.

The CRC0CTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 19 - 1 Format of Flash memory CRC control register (CRC0CTL)

Address: F02F0H After reset:00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of high-speed CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	0	0	0	0	0	0 to 3FFBH (16 K - 4 byte)
0	0	0	0	0	1	0 to 7FFBH (32 K - 4 byte)
0	0	0	0	1	0	0 to BFFBH (48 K - 4 byte)
0	0	0	0	1	1	0 to FFFBH (64 K - 4 byte)
0	0	0	1	0	0	0 to 13FFBH (80 K - 4 byte)
0	0	0	1	0	1	0 to 17FFBH (96 K - 4 byte)
0	0	0	1	1	0	0 to 1BFFBH (112 K - 4 byte)
0	0	0	1	1	1	0 to 1FFFBH (128 K - 4 byte)
Other than the above						Setting prohibited

Remark Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes.

19.3.1.2 Flash memory CRC operation result register (PGCRCL)

This register is used to store the high-speed CRC operation results.

The PGCRCL register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 19 - 2 Format of Flash memory CRC operation result register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8

7	6	5	4	3	2	1	0
PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

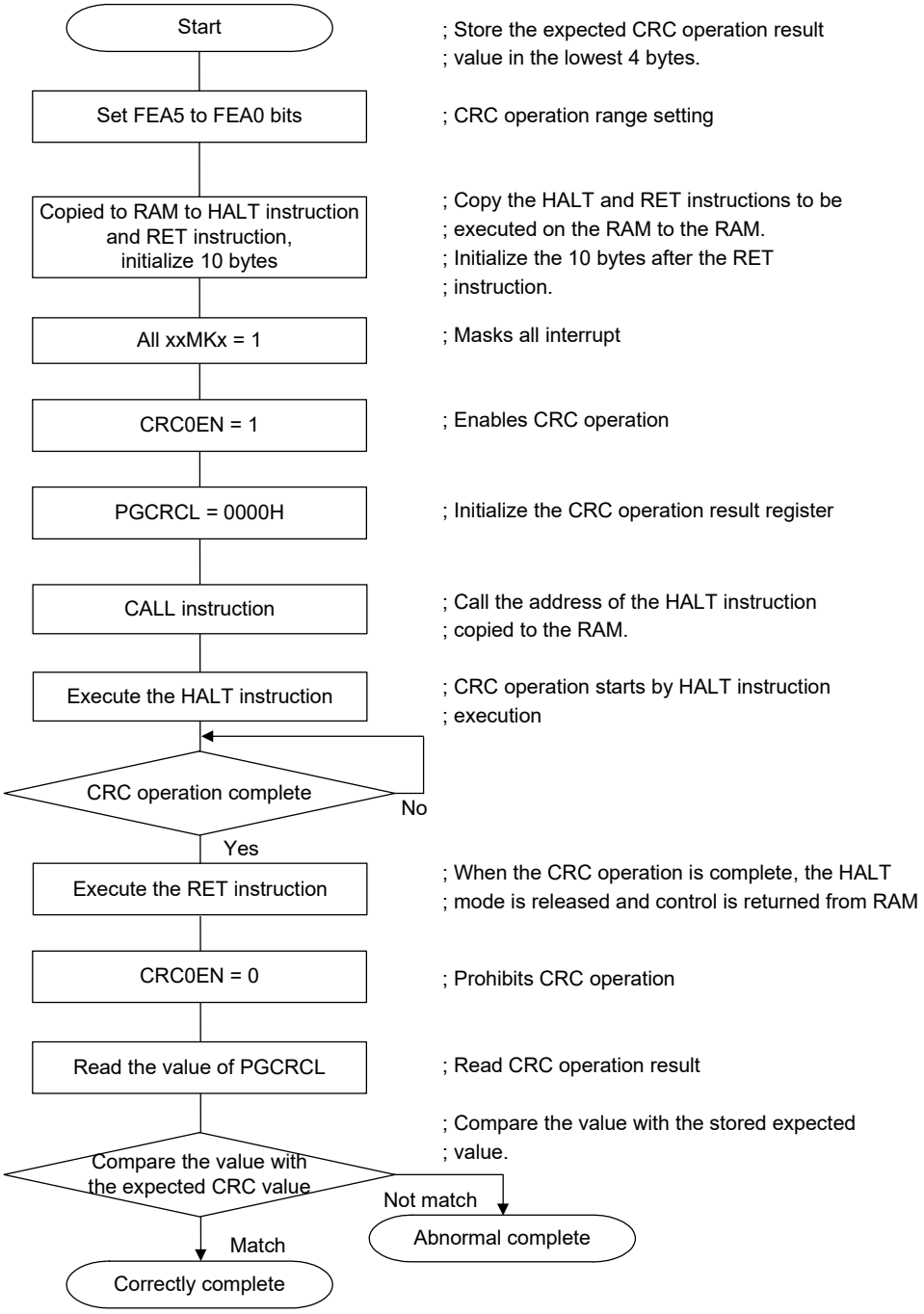
PGCRC15 to 0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 19 - 3 shows the Flowchart of Flash Memory CRC Operation Function (High-speed CRC).

<Operation flow>

Figure 19 - 3 Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Caution 1. The CRC operation is executed only on the code flash.
Caution 2. Store the expected CRC operation value in the area below the operation range in the code flash.
Caution 3. The CRC operation is enabled by executing the HALT instruction in the RAM area.
Be sure to execute the HALT instruction in RAM area.

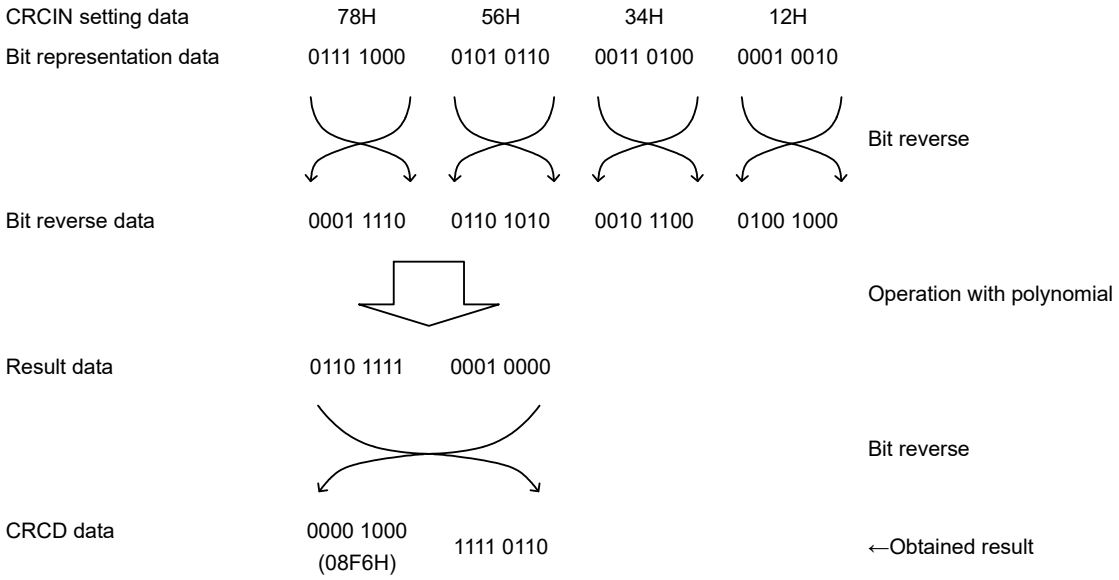
The expected CRC operation value can be calculated by using the integrated development environment CubeSuite+ development environment. Refer to the CubeSuite+ integrated development environment user's manual for details.

19.3.2 CRC operation function (general-purpose CRC)

In order to guarantee safety during operation, the IEC61508 standard mandates the checking of data even while the CPU is operating.

In R9A02G015, a general CRC operation can be executed as a peripheral function while the CPU is operating. The general CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program).

The CRC generator polynomial used is "$X^{16} + X^{12} + X^5 + 1$" of CRC-16-CCITT. The data to be input is inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H is sent from the LSB, values are written to the CRCIN register in the order of 78H, 56H, 34H, and 12H, enabling a value of 08F6H to be obtained from the CRCD register. This is the result obtained by executing a CRC operation on the bit rows shown below, which consist of the data 12345678H inverted in bit order.



Caution Because the debugger rewrites the software break setting line to a break instruction during program execution, the CRC operation result differs if a software break is set in the CRC operation target area.

19.3.2.1 CRC input register (CRCIN)

CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC.

The possible setting range is 00H to FFH.

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

Reset signal generation clears this register to 00H.

Figure 19 - 4 Format of CRC input register (CRCIN)

Address:FFFACH After reset:00H R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input.

19.3.2.2 CRC data register (CRCD)

This register is used to store the general-purpose CRC operation result.

The possible setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 19 - 5 Format of CRC data register (CRCD)

Address: F02FAH After reset: 0000H R/W

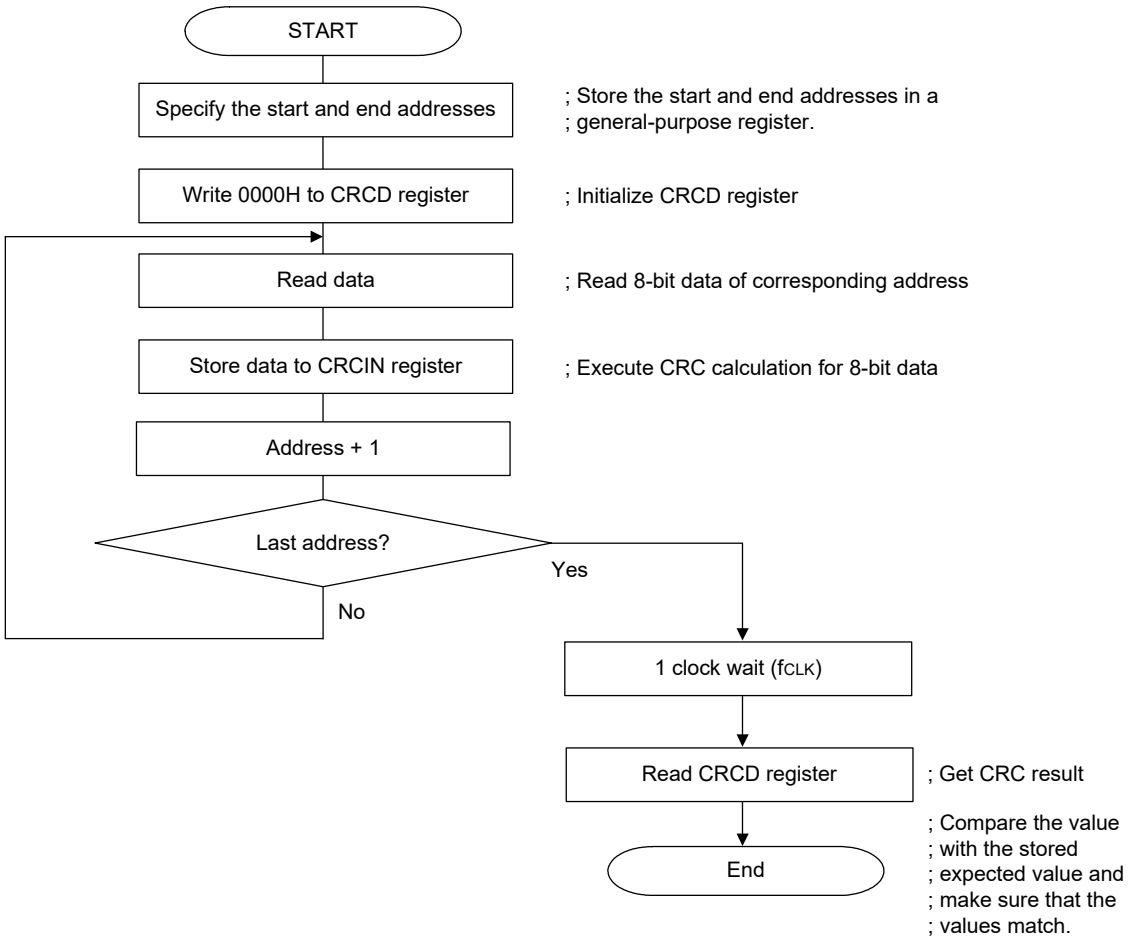
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Caution 1. Read the value written to CRCD register before writing to CRCIN register.

Caution 2. If writing and storing operation result to CRCD register conflict, the writing is ignored.

<Operation flow>

Figure 19 - 6 CRC Operation Function (General-Purpose CRC)



19.3.3 RAM parity error detection function

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in R9A02G015's RAM. By using this RAM parity error detection function, the parity bit is appended when data is written, and the parity is checked when the data is read. This function can also be used to trigger a reset when a parity error occurs.

19.3.3.1 RAM parity error control register (RPECTL)

This register is used to control parity error generation check bit and reset generation due to parity errors.
The RPECTL register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 19 - 7 Format of RAM parity error control register (RPECTL)

Address: F00F5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

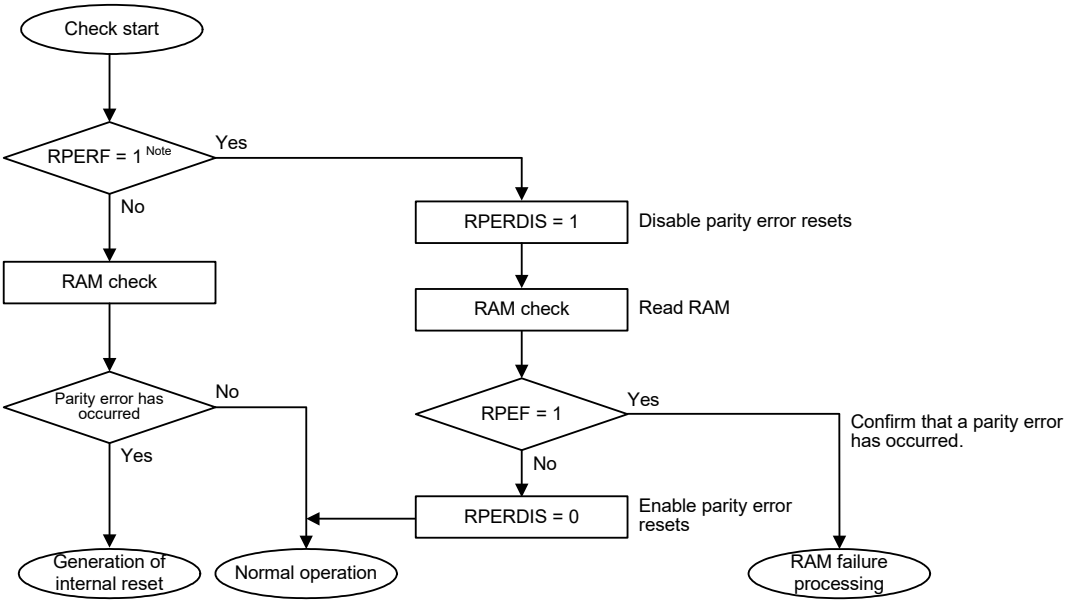
RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution **The parity bit is appended when data is written, and the parity is checked when the data is read. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed before reading data.**
R9A02G015's CPU executes look-ahead due to the pipeline operation, the CPU might read an uninitialized RAM area that is allocated beyond the RAM used, which causes a RAM parity error. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area + 10 bytes when instructions are fetched from RAM areas.

Remark 1. The parity error reset is enabled by default (RPERDIS = 0).
Remark 2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs. If the parity error reset is enabled (RPERDIS = 0) while RPEF = 1, a parity error reset occurs when RPERDIS is cleared (0).
Remark 3. The RPECTL flag in the RESF register is set (1) by RAM parity errors and cleared (0) by writing 0 to it or by any reset source. When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.
Remark 4. General-purpose registers are not included in the range of RAM parity error detection.

Figure 19 - 8 RAM Parity Error Check Flow



Note See **CHAPTER 16 RESET FUNCTION** for details on how to confirm internal resets due to RAM parity errors.

19.3.4 RAM guard function

In order to guarantee safety during operation, the IEC61508 standard mandates that important data stored in the RAM be protected, even if the CPU freezes.

This RAM guard function is used to protect data in the specified memory space.

If the RAM guard function is specified, writing to the specified RAM space is disabled, but reading from the space can be carried out as usual.

19.3.4.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GRAM1 and GRAM0 bits are used in RAM guard function.

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

Reset signal generation clears this register to 00H.

Figure 19 - 9 Format of Invalid memory access detection control register (IAWCTL)

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space Note
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes starting at the start RAM address
1	0	The 256 bytes starting at the start RAM address
1	1	The 512 bytes starting at the start RAM address

Note The RAM start address differs depending on the size of the RAM provided with the product.

19.3.5 SFR guard function

In order to guarantee safety during operation, the IEC61508 standard mandates that important SFRs be protected from being overwritten, even if the CPU freezes.

This SFR guard function is used to protect data in the control registers used by the port function, interrupt function, clock control function, voltage detection function, and RAM parity error detection function.

If the SFR guard function is specified, writing to the specified SFRs is disabled, but reading from the SFRs can be carried out as usual.

19.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GPORT, GINT and GCSC bits are used in SFR guard function.

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

Reset signal generation clears this register to 00H.

Figure 19 - 10 Format of Invalid memory access detection control register (IAWCTL)

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Control registers of port function guard
0	Disabled. Control registers of port function can be read or written to.
1	Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR] PMxx, PUxx, PIMxx, POMxx, PMCxx Note

GINT	Registers of interrupt function guard
0	Disabled. Registers of interrupt function can be read or written to.
1	Enabled. Writing to registers of interrupt function is disabled. Reading is enabled. [Guarded SFR] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Control registers of clock control function, voltage detector, and RAM parity error detection function guard
0	Disabled. Control registers of clock control function, voltage detector and RAM parity error detection function can be read or written to.
1	Enabled. Writing to control registers of clock control function, voltage detector and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR] CMC, CSC, OSTS, CKC, PERx, PRRx, OSMC, LVIM, LVIS, RPECTL, DSCCTL, MCKC

Note Pxx (Port register) is not guarded.

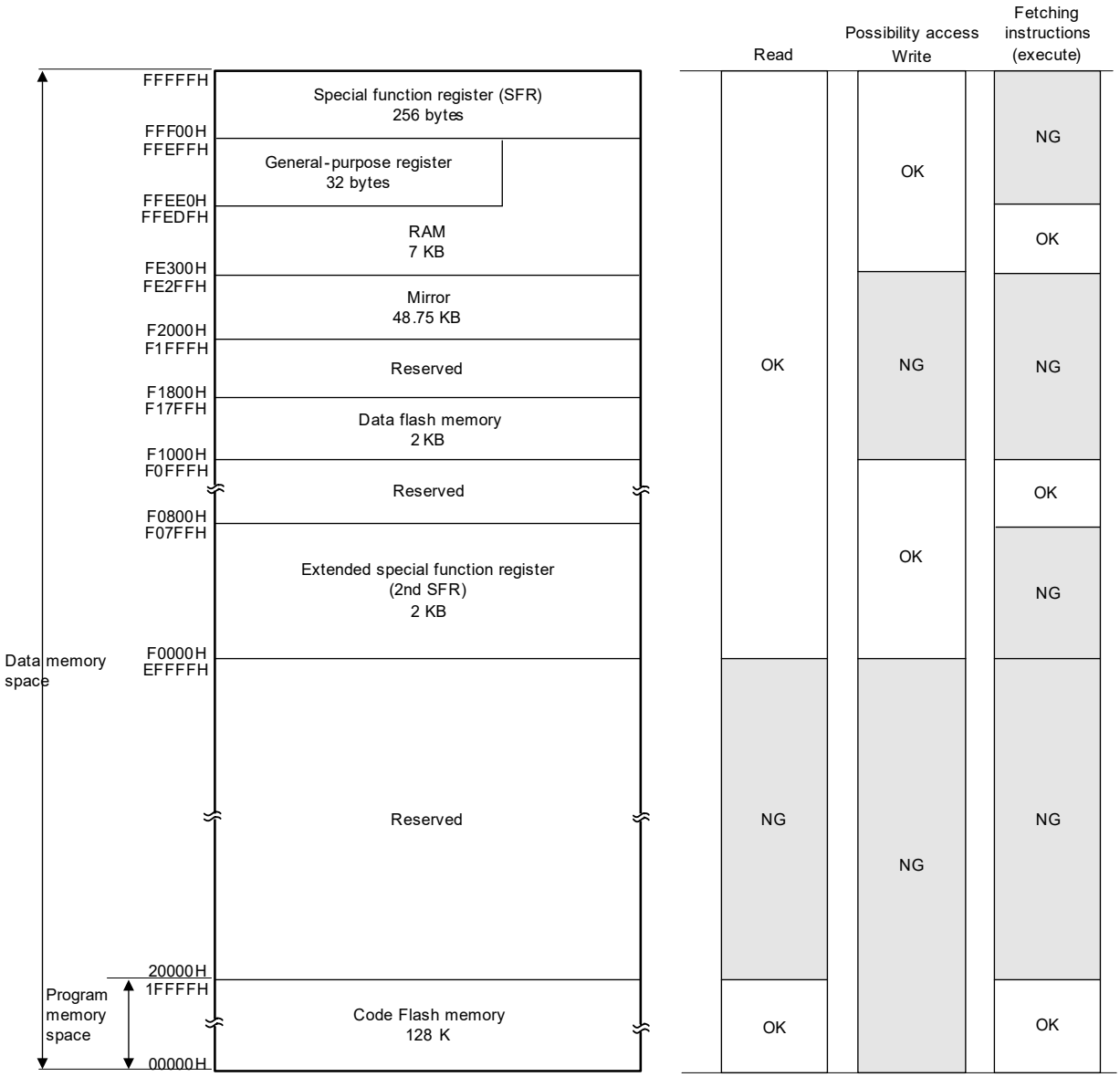
19.3.6 Invalid memory access detection function

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.

The illegal memory access detection function triggers a reset if a memory space specified as access-prohibited is accessed.

The illegal memory access detection function applies to the areas indicated by NG in Figure 19 - 11.

Figure 19 - 11 Invalid access detection area



19.3.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

IAWEN bit is used in invalid memory access detection function.

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

Reset signal generation clears this register to 00H.

Figure 19 - 12 Format of Invalid memory access detection control register (IAWCTL)

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN Note	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN Note	Control of invalid memory access detection
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is enabled, not writing 0 to it after setting it to 1.

Remark By specifying WDTON = 1 for the option byte (watchdog timer operation enable), the invalid memory access detection function is enabled even if IAWEN = 0.

19.3.7 Frequency detection function

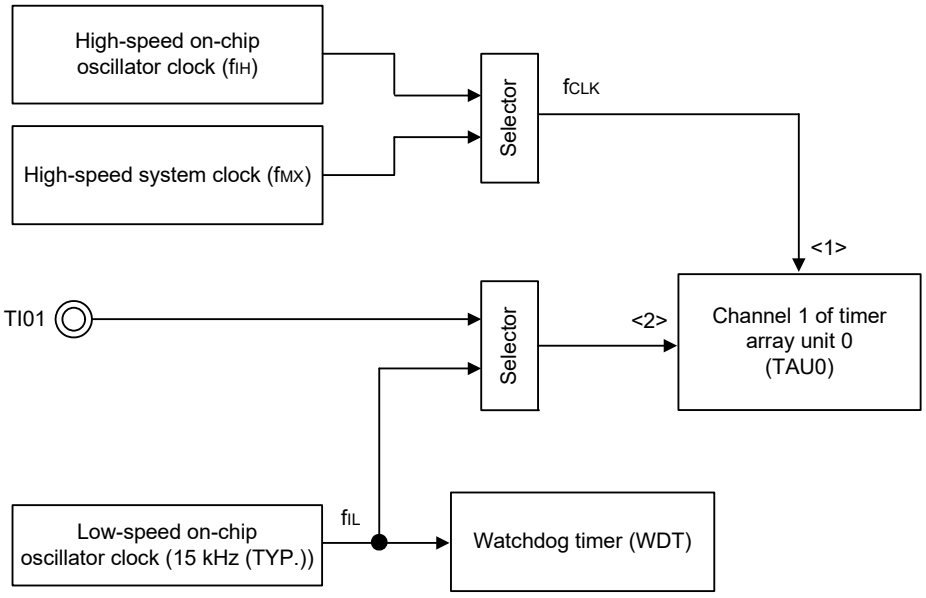
The IEC60730 standard mandates checking that the oscillation frequency is correct.
By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clocks to be compared>

<1> CPU/peripheral hardware clock frequency (f_{CLK}):
- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 1 of the timer array unit 0
- Timer input to channel 1 (TI01)
- Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))

Figure 19 - 13 Configuration of Frequency Detection Function



If pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.
For how to execute pulse interval measurement, see **6.8.3 Operation as input pulse interval measurement**.

19.3.7.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channels 0 and 1 of the timer array unit 0 (TAU0).

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 19 - 14 Format of Timer input select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
Other than above			Setting prohibited

Caution At least 1/f_{MCK} + 10 ns is necessary as the high-level and low-level widths of the timer input to be selected.

19.3.8 A/D test function

The IEC60730 standard mandates testing the A/D converter. The A/D test function is used to check whether the A/D converter is operating normally by executing A/D conversions of the positive reference voltage and negative reference voltage of the A/D converter, analog input channel (ANI), temperature sensor output voltage, and internal reference voltage. For details on the checking method, refer to the safety function (A/D test) application note (R01AN0955).

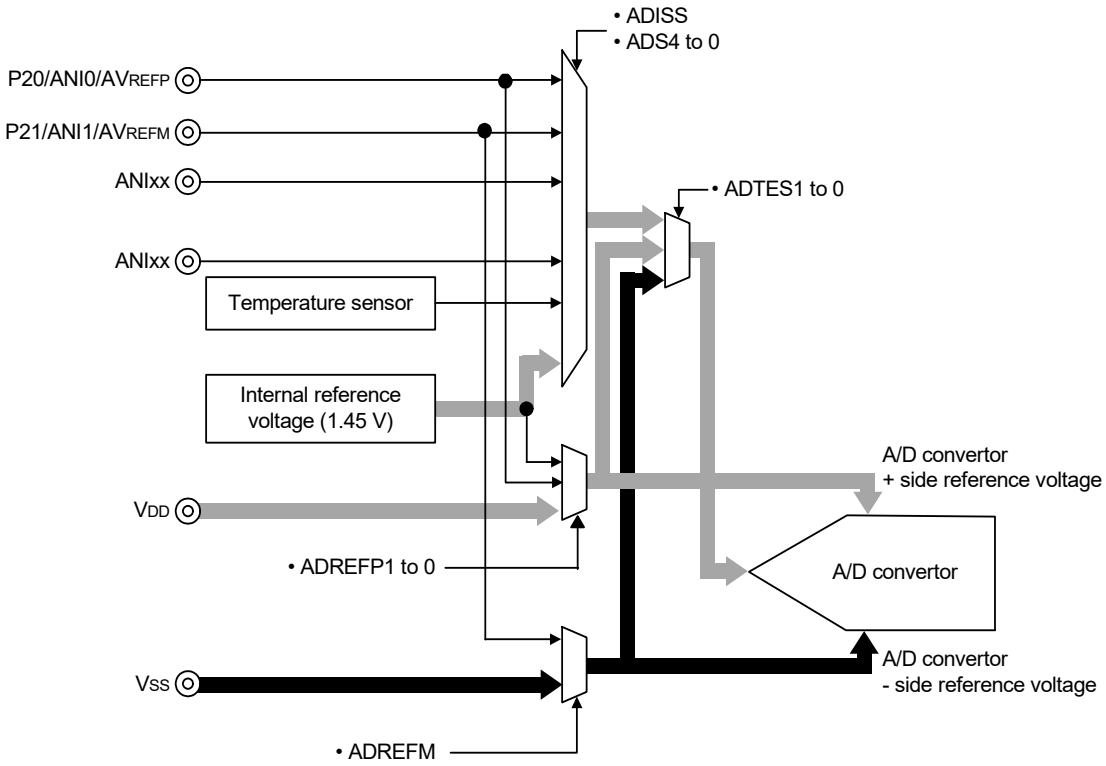
The analog multiplexer can be checked using the following procedure.

(1) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(2) Perform A/D conversion for the ANIx pin (conversion result 1-1).
(3) Select the negative reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 1, 0).
(4) Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
(5) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(6) Perform A/D conversion for the ANIx pin (conversion result 1-2).
(7) Select the positive reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 1, 1).
(8) Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
(9) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(10) Perform A/D conversion for the ANIx pin (conversion result 1-3).
(11) Make sure that “conversion result 1-1” = “conversion result 1-2” = “conversion result 1-3”.
(12) Make sure that the A/D conversion results of “conversion result 2-1” are all 0 and those of “conversion result 2-2” are all 1.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remark 1. If the analog input voltage is variable during conversion in steps (1) to (10) above, use another method to check the analog multiplexer.
Remark 2. The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.
Remark 3. Set A/D test register (ADTES) to 00H when measuring the ANIxx/temperature sensor output /internal reference voltage (1.45 V).
Remark 4. For details on Analog input channel specification register, refer to **Figure 10 - 12 Format of Analog Input Channel Specification Register (ADS)**.
For details on the A/D test register, refer to **Figure 10 - 16 Format of A/D Test Register (ADTES)**.

Figure 19 - 15 Configuration of A/D Test Function



19.3.9 Digital output signal level detection function for I/O pins

In the IEC60730, it is required to check that the I/O function correctly operates.

By using the digital output signal level detection function for I/O pins, the digital output level of the pin can be read when the port is set to output mode.

19.3.9.1 Port mode select register (PMS)

This register is used to select the output level from output latch level or pin output level when the pin is output mode in which PMm bit of port mode register (PMm) is 0.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 19 - 16 Format of Port mode select register (PMS)

Address: F007BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	Method for selecting output level to be read when pin is output mode
0	Pmn register value is read.
1	Digital output level of the pin is read.

Caution 1. While the PMS0 bit in the PMS register is set to 1, do not change the value of the port register (Pxx) using a bit manipulation instruction. To change the value of the port register (Pxx), use an 8-bit data manipulation instruction.

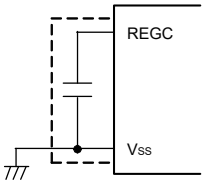
Caution 2. When P60 to P64 are used as general-purpose ports, the pin output level cannot read using PMS0. The read value is 0.

Remark m = 0, 2, 4 to 7
n = 0 to 5

CHAPTER 20 REGULATOR

20.1 Regulator Overview

R9A02G015 contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

The regulator output voltage, see **Table 20 - 1**.

Table 20 - 1 Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
HS (high-speed main) mode	1.8 V	In STOP mode
	2.1 V	Other than above (include during OCD mode) Note

Note When it shifts to STOP mode during the on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.8 V).

CHAPTER 21 OPTION BYTE

21.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the R9A02G015 form an option byte area.
Option bytes consist of user option byte (000C0H to 000C2H) and on-chip debug option byte (000C3H).
Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes.
For the bits to which no function is allocated, do not change their initial values.
To use the boot swap operation during self-programming, 000C0H to 000C3H are replaced by 010C0H to 010C3H. Therefore, set the same values as 000C0H to 000C3H to 010C0H to 010C3H.

Remark The option bytes should always be set regardless of whether each function is used.

21.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

(1) 000C0H/010C0H

- ○ Setting of watchdog timer operation
  - • Enabling or disabling of counter operation
  - • Enabling or disabling of counter operation in the HALT or STOP mode
- ○ Setting of interval time of watchdog timer
- ○ Setting of window open period of watchdog timer
- ○ Setting of interval interrupt of watchdog timer
  - • Interval interrupt is used or not used

Caution Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

(2) 000C1H/010C1H

- ○ Setting of LVD operation mode
  - • Interrupt & reset mode
  - • Reset mode
  - • Interrupt mode
  - • LVD off (external reset input from the RESETB pin is used)
- ○ Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})

Caution 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 2.4 AC Characteristics of the R9A02G015 Data Sheet (R19DS0101E). This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

Caution 2. Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

(3) 000C2H/010C2H

○ Setting of flash operation mode

Make the setting depending on the main system clock frequency (f_{MAIN}) and power supply voltage (V_{DD}) to be used.

• HS (high-speed main) mode

○ Setting of the frequency of the high-speed on-chip oscillator

• Select from 48MHz, 24MHz, 16MHz, 12MHz, 8MHz, 6MHz, 4MHz, 3MHz, 2MHz or 1MHz(TYP).

Caution Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

21.1.2 On-chip debug option byte (000C3H/010C3H)

○ Control of on-chip debug operation

• On-chip debug operation is disabled or enabled.

○ Handling of data of flash memory in case of failure in on-chip debug security ID authentication

• Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

Caution Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

21.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 21 - 1 Format of User Option Byte (000C0H/010C0H)

Address: 000C0H/010C0H Note 1

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2 f_{IL} of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period Note 2
0	0	Setting prohibited
0	1	50%
1	0	75% Note 3
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode Note 2
1	Counter operation enabled in HALT/STOP mode

Note 1. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

Note 2. The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

(**Note** continues on the next page.)

Note 3. When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)	1899.59 ms to 2570.04 ms

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 21 - 2 Format of User Option Byte (000C1H/010C1H) (1/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt & reset mode)

Detection Voltage			Option Byte Setting Value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
2.92 V	2.86 V	2.75 V	0	1	1	1	0	1	0
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
—			Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution Be sure to set bit 4 to "1".

Remark 1. For details on the LVD circuit, see **CHAPTER 18 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **2.6.4 LVD circuit characteristics** in the R9A02G015 Data Sheet (R19DS0101E).

Figure 21 - 3 Format of User Option Byte (000C1H/010C1H) (2/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (reset mode)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.81 V	2.75 V	0	1	1	1	1	1	1
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution Be sure to set bit 4 to "1".

Remark 1. For details on the LVD circuit, see **CHAPTER 18 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **2.6.4 LVD circuit characteristics** in the R9A02G015 Data Sheet (R19DS0101E).

Figure 21 - 4 Format of User Option Byte (000C1H/010C1H) (3/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt mode)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.81 V	2.75 V	0	1	1	1	1	0	1
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution Be sure to set bit 4 to "1".

Remark 1. For details on the LVD circuit, see **CHAPTER 18 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **2.6.4 LVD circuit characteristics** in the R9A02G015 Data Sheet (R19DS0101E).

Figure 21 - 5 Format of User Option Byte (000C1H/010C1H) (4/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD off setting (external reset input from the RESETB pin is used)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution 1. Be sure to set bit 4 to "1".

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 2.4 AC Characteristics of the R9A02G015 Data Sheet (R19DS0101E). This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

Remark 1. ×: Don't care

Remark 2. For details on the LVD circuit, see **CHAPTER 18 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a typical value. For details, see **2.6.4 LVD circuit characteristics** in the R9A02G015 Data Sheet (R19DS0101E).

Figure 21 - 6 Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H Note 1

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode		
			Operating Frequency Range (f_{MAIN})	Operating Voltage Range (V_{DD})
1	1	HS (high-speed main) mode	1 to 24 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

FRQSEL4 Note 4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator clock	
					f_{HOCO}	f_{IH}
1	0	0	0	0	48 MHz Note 3	24 MHz, 12 MHz, 6 MHz Note 2
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than above					Setting prohibited	

Note 1. Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

Note 2. See the MCKC register for f_{IH} division ratio settings.

Note 3. When using the high-speed on-chip oscillator clock (f_{HOCO}) to operate the USB controller, be sure to set f_{HOCO} = 48MHz.

Note 4. When using R9A02G0151 (without USB), be sure to set FRQSEL4 to 0.

21.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 21 - 7 Format of On-chip Debug Option Byte (000C3H/010C3H)

Address: 000C3H/010C3H Note

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Note Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

Caution **Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.**
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

21.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the assembler linker option, in addition to describing in the source. When doing so, the contents set by using the link option take precedence, even if descriptions exist in the source, as mentioned below.

A software description example of the option byte setting is shown below.

```
OPT     CSEG    OPT_BYTE
        DB      36H     ; Does not use interval interrupt of watchdog timer,
                        ; Enables watchdog timer operation,
                        ; Window open period of watchdog timer is 50%,
                        ; Overflow time of watchdog timer is 2^9/fIL,
                        ; Stops watchdog timer operation during HALT/STOP mode
        DB      7AH     ; Select 2.75 V for VLVDL
                        ; Select rising edge 2.92 V, falling edge 2.86 V for VLVDH
                        ; Select the interrupt & reset mode as the LVD operation mode
        DB      F0H     ; Select the HS (high speed main) mode as the flash operation mode
                          and 48 MHz as the frequency of the high-speed on-chip oscillator clock
        DB      85H     ; Enables on-chip debug operation, does not erase flash memory
                          data when security ID authorization fails
```

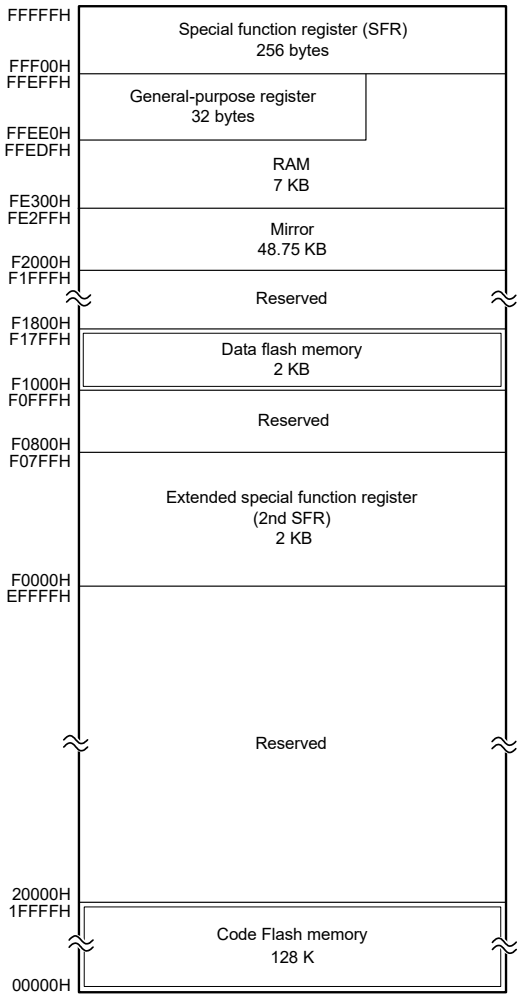
When the boot swap function is used during self-programming, 000C0H to 000C3H is switched to 010C0H to 010C3H. Describe to 010C0H to 010C3H, therefore, the same values as 000C0H to 000C3H as follows.

```
OPT2    CSEG    AT      010C0H
        DB      36H     ; Does not use interval interrupt of watchdog timer,
                        ; Enables watchdog timer operation,
                        ; Window open period of watchdog timer is 50%,
                        ; Overflow time of watchdog timer is 2^9/fIL,
                        ; Stops watchdog timer operation during HALT/STOP mode
        DB      7AH     ; Select 2.75 V for VLVDL
                        ; Select rising edge 2.92 V, falling edge 2.86 V for VLVDH
                        ; Select the interrupt & reset mode as the LVD operation mode
        DB      F0H     ; Select the HS (high speed main) mode as the flash operation mode
                          and 48 MHz as the frequency of the high-speed on-chip oscillator clock
        DB      85H     ; Enables on-chip debug operation, does not erase flash memory
                          data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 010C0H to 010C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 22 FLASH MEMORY

R9A02G015 incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board. The flash memory includes the “code flash memory”, in which programs can be executed, and the “data flash memory”, an area for storing data.



The following methods for programming the flash memory are available.

The code flash memory can be rewritten to through serial programming using a flash memory programmer or an external device (UART communication), or through self-programming.

- Serial Programming Using Flash Memory Programmer (see **22.1**)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.
- Serial Programming Using External Device (that Incorporates UART) (see **22.2**)
  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).
- Self-Programming (see **22.6**)
  The user application can execute self-programming of the code flash memory by using the flash self-programming library.

The data flash memory can be rewritten to by using the data flash library during user program execution (background operation). For access and writing to the data flash memory, see **22.8 Data Flash**.

22.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of R9A02G015.

- PG-FP6
- E1, E2, E2 Lite, E20 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming
The contents of the flash memory can be rewritten after R9A02G015 has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming
Data can be written to the flash memory with a dedicated program adapter before the RL78 microcontroller is mounted on the target system.

Table 22 - 1 Wiring Between R9A02G015 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.
					32-pin
Signal Name		I/O	Pin Function		HVQFN (4 x 4)
PG-FP6	E1, E2, E2 Lite, E20 on-chip debugging emulator				
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/ P40	1
SI/RxD	—	I/O	Transmit/ receive signal		
—	$\overline{\text{RESET}}$	Output	Reset signal	RESETB	2
/RESET	—	Output			
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	7
GND		—	Ground	V_{SS}	exposed die pad (V_{SS})
				REGC Note	6
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}	7

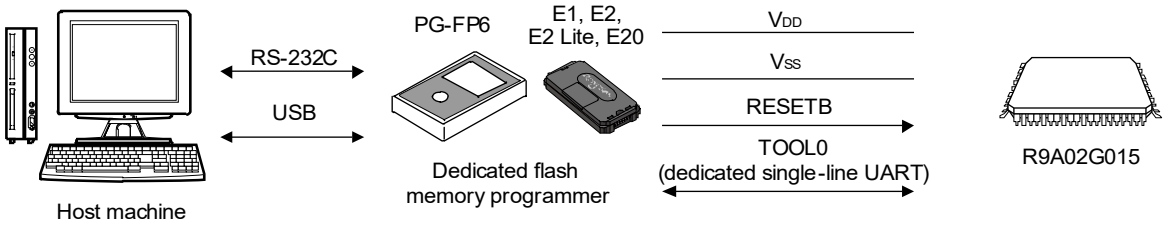
Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

22.1.1 Programming Environment

The environment required for writing a program to the flash memory of R9A02G015 is illustrated below.

Figure 22 - 1 Environment for Writing Program to Flash Memory



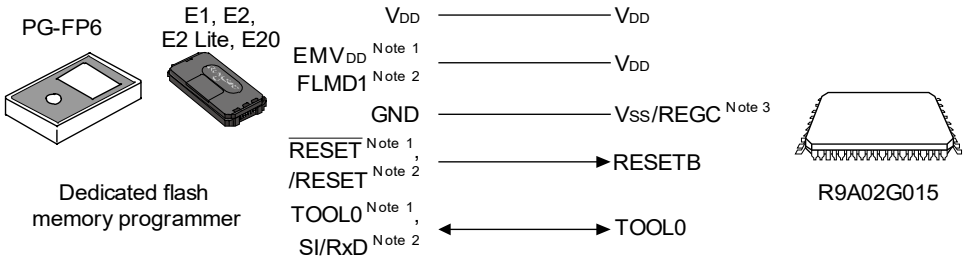
A host machine that controls the dedicated flash memory programmer is necessary.
To interface between the dedicated flash memory programmer and R9A02G015, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

22.1.2 Communication Mode

Communication between the dedicated flash memory programmer and R9A02G015 is established by serial communication using the TOOL0 pin via a dedicated single-line UART of R9A02G015.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 22 - 2 Communication with Dedicated Flash Memory Programmer



Note 1. When using E1, E2, E2 Lite, E20 on-chip debugging emulator.
Note 2. When using PG-FP6.
Note 3. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The dedicated flash memory programmer generates the following signals for R9A02G015. See the manual of PG-FP6 or E1, E2, E2 Lite, E20 on-chip debugging emulator for details.

Table 22 - 2 Pin Connection

Dedicated Flash Memory Programmer				R9A02G015
Signal Name		I/O	Pin Function	Pin Name Note 2
PG-FP6	E1, E2, E2 Lite, E20 on-chip debugging emulator			
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		—	Ground	V_{SS}, REGC Note 1
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}
/RESET	—	Output	Reset signal	RESETB
—	$\overline{RESET}$	Output		
—	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	—	I/O	Transmit/receive signal	

Note 1. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Note 2. Pins to be connected differ with the product. For details, see **Table 22 - 1**.

22.2 Serial Programming Using External Device (that Incorporates UART)

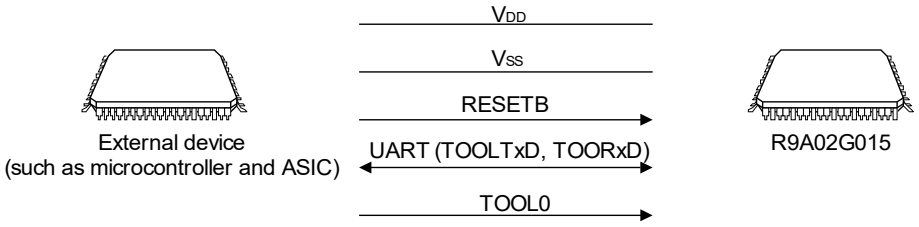
On-board data writing to the internal flash memory is possible by using R9A02G015 and an external device (a microcontroller or ASIC) connected to a UART.

On the development of flash memory programmer by user, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

22.2.1 Programming Environment

The environment required for writing a program to the flash memory of R9A02G015 is illustrated below.

Figure 22 - 3 Environment for Writing Program to Flash Memory



Processing to write data to or delete data from R9A02G015 by using an external device is performed on-board. Off-board writing is not possible.

22.2.2 Communication Mode

Communication between the external device R9A02G015 is established by serial communication using the TOOLTxD and TOOLRxD pins via the dedicated UART of R9A02G015.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 22 - 4 Communication with External Device

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The external device generates the following signals for R9A02G015.

Table 22 - 3 Pin Connection

External Device			R9A02G015
Signal Name	I/O	Pin Function	Pin Name
VDD	I/O	VDD voltage generation/power monitoring	VDD
GND	—	Ground	VSS, REGC Note
RESETOUT	Output	Reset signal output	RESETB
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

22.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

Remark For details on flash memory programming mode, refer to **22.4.2 Flash memory programming mode**.

22.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 kΩ pull-up resistor.

When this pin is used as the port pin, use that by the following method.

When used as an input pin: Input of low-level is prohibited for t_{HD} period after external reset release. However, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

When used as an output pin: When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

Remark 1. t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode. For details, refer to **R9A02G015 Data Sheet (R19DS0101E)**.

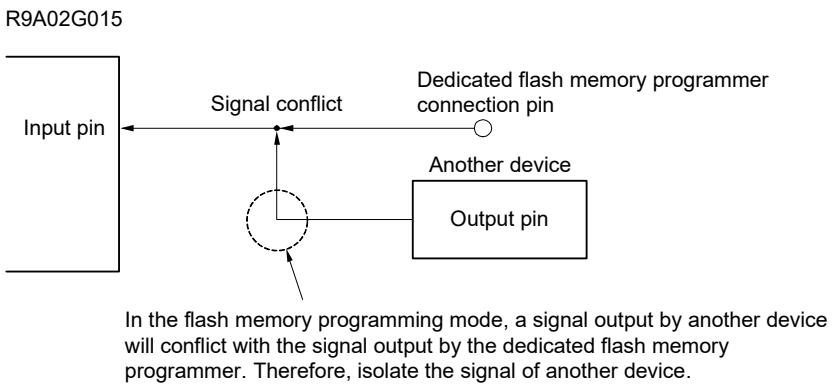
Remark 2. The SAU and IICA pins are not used for communication between R9A02G015 and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

22.3.2 RESETB pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the RESETB pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

Figure 22 - 5 Signal Conflict (RESETB Pin)



22.3.3 Port pins

Example When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD}, or V_{SS} via a resistor

22.3.4 REGC pin

Connect the REGC pin to GND via a capacitor having excellent characteristics (0.47 to 1 μF) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

22.3.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{IH}) is used.

22.3.6 Power supply

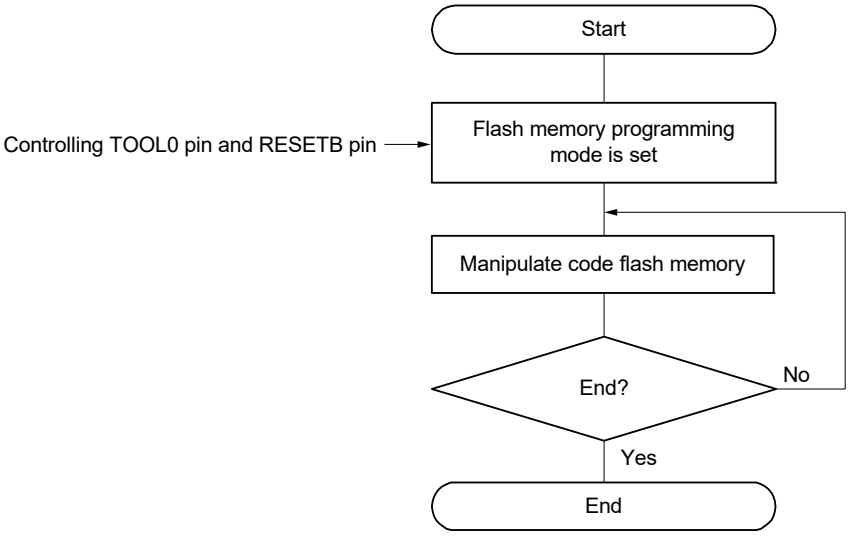
To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the exposed die pad (V_{SS}) to GND of the flash memory programmer.
To use the on-board supply voltage, connect in compliance with the normal operation mode.
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and exposed die pad (V_{SS}) to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

22.4 Programming Method

22.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 22 - 6 Code Flash Memory Manipulation Procedure



22.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify the flash memory programming mode. To enter the mode, set as follows.

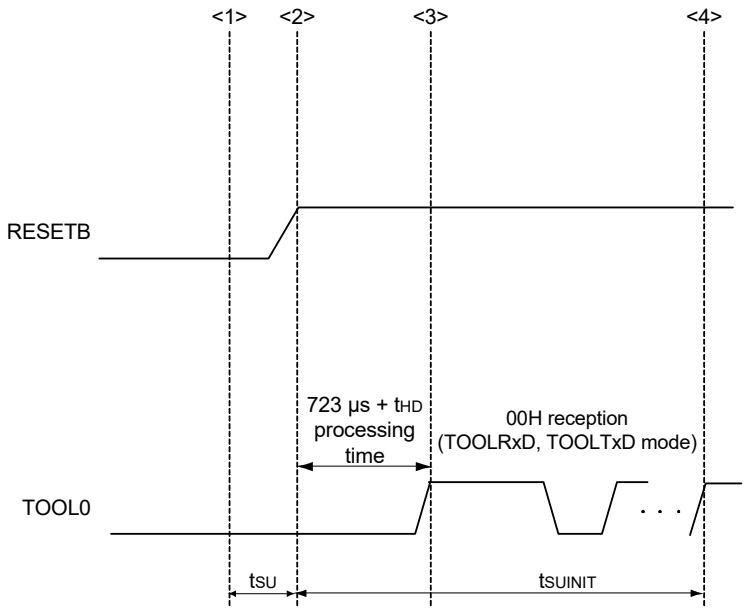
<When serial programming by using the dedicated flash memory programmer>
Connect R9A02G015 to a dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

<When serial programming by using an external device>
Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 22 - 4**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 22 - 7**. For details, refer to the **RL78 microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 22 - 4 Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

Figure 22 - 7 Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the external resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends.
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external resets end (the flash firmware processing time is excluded).

For details, refer to **R9A02G015 Data Sheet (R19DS0101E)**.

The supply voltage value applied to R9A02G015 during write operations and the setting information of the user option byte for setting of the flash memory programming mode determine which mode is selected.
When a dedicated flash memory programmer is used for serial programming, setting the voltage on GUI selects the mode automatically.

Table 22 - 5 Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Power Supply Voltage (V_{DD})	User Option Byte Setting for Switching to Flash Memory Programming Mode		Flash Programming Mode
	Flash Operation Mode	Operating Frequency (f_{CLK})	
2.7 V ≤ V_{DD} ≤ 5.5 V	Blank state		Full speed mode
	HS (high-speed main) mode	1 MHz to 24 MHz	Full speed mode

Remark For details about communication commands, see **22.4.4 Communication commands**.

22.4.3 Selecting communication mode

Communication mode of R9A02G015 as follows.

Table 22 - 6 Communication Modes

Communication Mode	Standard Setting Note 1				Pins Used
	Port	Speed Note 2	Frequency	Multiply Rate	
1-line mode (when flash memory programmer is used, or when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOL0
Dedicated UART (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOLTxD, TOOLRxD

Note 1. Selection items for Standard settings on GUI of the flash memory programmer.

Note 2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

22.4.4 Communication commands

R9A02G015 executes serial programming through the commands listed in **Table 22 - 7**.

The signals sent from the dedicated flash memory programmer or external device to R9A02G015 are called commands, and programming functions corresponding to the commands are executed. For details, refer to the **RL78 microcontroller (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 22 - 7 Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased
Write	Programming	Writes data to a specified area in the flash memory Note.
Getting information	Silicon Signature	Gets R9A02G015 information (such as the part number, flash memory configuration, and programming firmware version).
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
	Security Get	Gets security information.
	Security Release	Release setting of prohibition of writing.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

Product information (such as product name and firmware version) can be obtained by executing the “Silicon Signature” command.

Tables 22 - 8 and 22 - 9 show signature data list and example of signature data list.

Table 22 - 8 Signature Data List

Field name	Description	Number of transmit data
Device code	The serial number assigned to the device	3 bytes
Device name	Device name (ASCII code)	10 bytes
Code flash memory area last address	Last address of code flash memory area (Sent from lower address. Example. 00000H to 07FFFH (32 KB) → FFH, 7FH, 00H)	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address. Example. F1000H to F17FFH (2 KB) → FFH, 17H, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address. Example. From Ver. 1.23 → 01H, 02H, 03H)	3 bytes

Table 22 - 9 Signature Data List

Field name	Description	Number of transmit data	Data (hexadecimal)
Device code	RL78 protocol A	3 bytes	10 00 06
Device name	R9A02G0150	10 bytes	52 = “R” 39 = “9” 41 = “A” 30 = “0” 32 = “2” 47 = “G” 30 = “0” 31 = “1” 35 = “5” 30 = “0”
Code flash memory area last address	Code flash memory area 00000H to 1FFFFH (128 KB)	3 bytes	FFH FFH 01H
Data flash memory area last address	Data flash memory area F1000H to F17FFH (2 KB)	3 bytes	FFH 17H 0FH
Firmware version	Ver.1.23	3 bytes	01 02 03

22.5 Processing Time for Each Command When PG-FP6 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP6 is used as a dedicated flash memory programmer.

Table 22 - 10 Processing Time for Each Command When PG-FP6 Is in Use (Reference Value)

PG-FP6 Command	Port: TOOL0 (UART) Speed: 1M bps 128 Kbytes
Erasing	2 s
Writing	3.2 s
Verification	3.5 s
Writing after erasing	4.5 s

Remark The command processing times (reference values) shown in the table are typical values under the following conditions.
Port: TOOL0 (single-line UART)
Speed: 1,000,000 bps
Mode: Full speed mode (flash operation mode: HS (high speed main) mode)

22.6 Self-Programming

R9A02G015 supports a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the RL78 microcontroller self-programming library, it can be used to upgrade the program in the field.

Caution 1. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the flash self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the flash self-programming library.

Caution 2. The high-speed on-chip oscillator should be kept operating during self-programming. If it is stopped, its clock should be operated (HIOSTOP = 0), and the flash self-programming library should be executed after 65 μs have elapsed.

Remark 1. For details of the self-programming function, refer to the **RL78 microcontroller Flash Self-Programming Library Type01 User's Manual (R01US0050)**.

Remark 2. For details of the time required to execute self-programming, see the notes on use that accompany the flash self-programming library tool.

The self-programming function has two flash memory programming modes; wide voltage mode and full speed mode. Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H.

Specify the full speed mode when the HS (high-speed main) mode is specified.

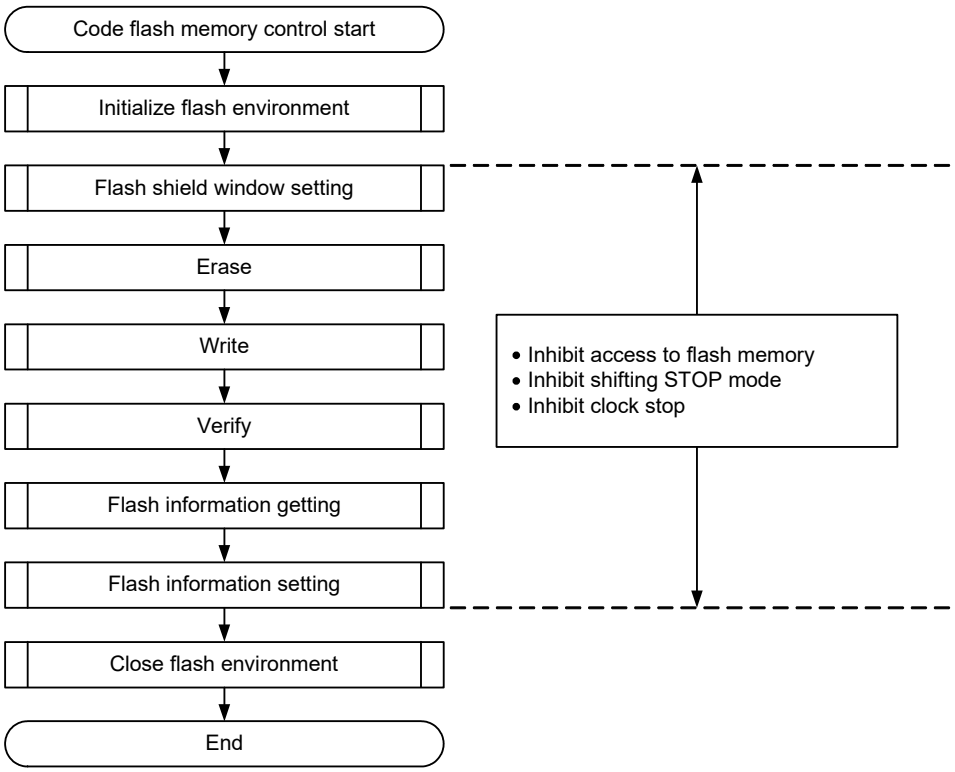
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

Remark The only Full speed mode can be used for R9A02G015.

22.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

Figure 22 - 8 Flow of Self-Programming (Rewriting Flash Memory)



22.6.2 Boot swap function

If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

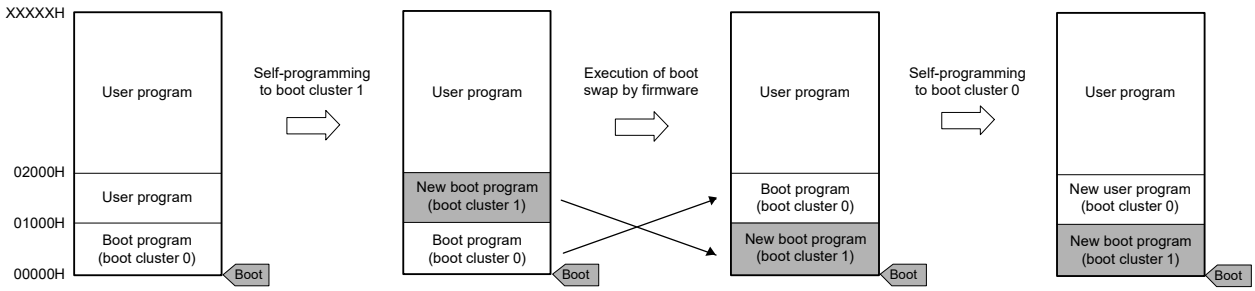
The boot swap function is used to avoid this problem.

Before erasing boot cluster 0 Note, which is a boot program area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of R9A02G015, so that boot cluster 1 is used as a boot area. After that, erase or write the original boot program area, boot cluster 0.

As a result, even if a power failure occurs while the area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Figure 22 - 9 Boot Swap Function

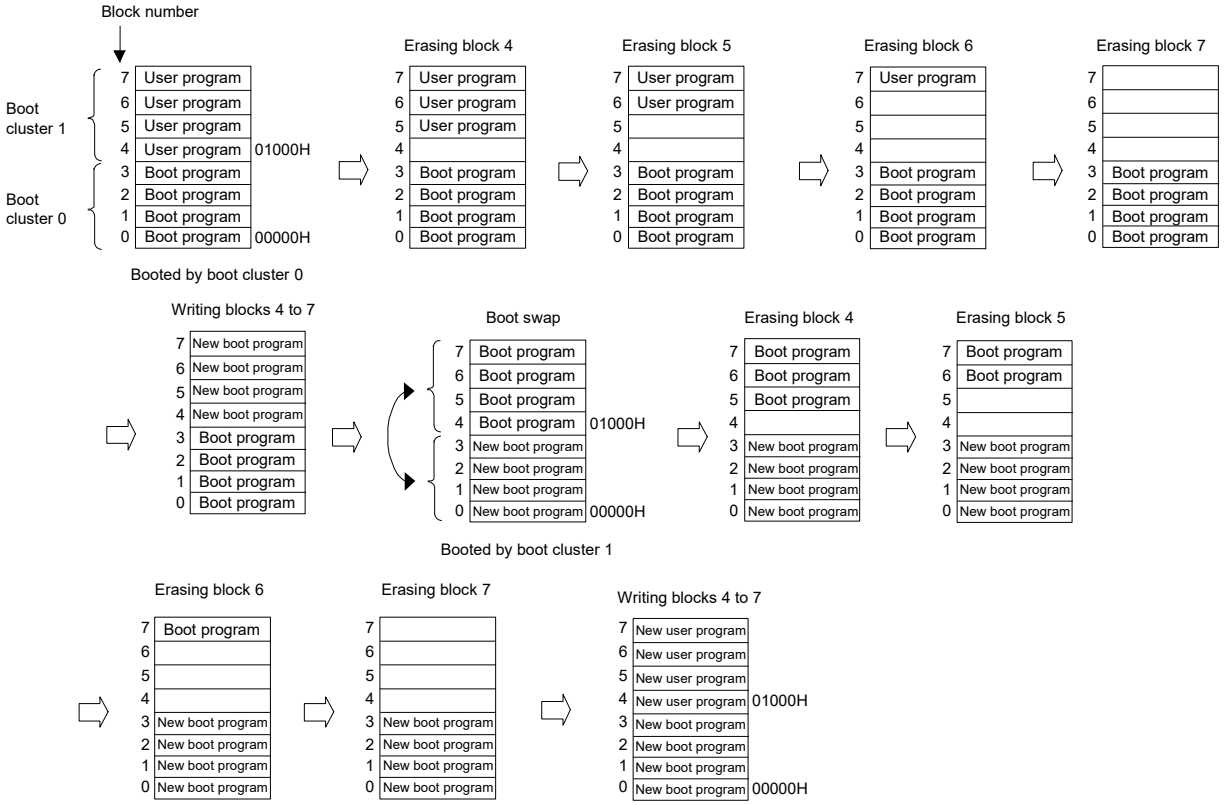


In an example of above figure, it is as follows.

Boot cluster 0: Boot area before boot swap

Boot cluster 1: Boot area after boot swap

Figure 22 - 10 Example of Executing Boot Swapping



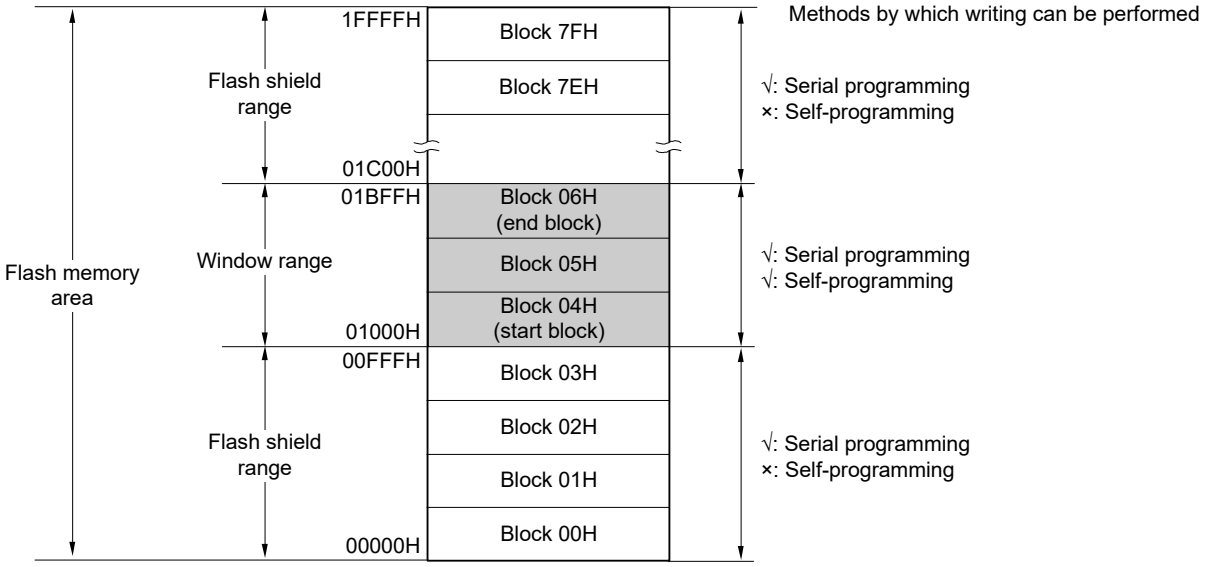
22.6.3 Flash shield window function

The flash shield window function is provided as one of the security functions for self-programming. It disables writing to and erasing areas outside the range specified as a window only during self-programming.

The window range can be set by specifying the start and end blocks. The window range can be set or changed during both serial programming and self-programming.

Writing to and erasing areas outside the window range are disabled during self-programming. During serial programming, however, areas outside the range specified as a window can be written and erased.

Figure 22 - 11 Flash Shield Window Setting Example
(Target Devices: R9A02G015, Start Block: 04H, End Block: 06H)



Caution 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

Caution 2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 22 - 11 Relationship between Flash Shield Window Function Setting/Change Methods and Commands

Programming conditions	Window Range Setting/ Change Methods	Execution Commands	
		Block erase	Write
Self-programming	Specify the starting and ending blocks by the flash self-programming library.	Block erasing is enabled only within the window range.	Writing is enabled only within the range of window range.
Serial programming	Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc.	Block erasing is enabled also outside the window range.	Writing is enabled also outside the window range.

Remark See **22.7 Security Settings** to prohibit writing/erasing during serial programming.

22.7 Security Settings

R9A02G015 supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.
The operations shown below can be performed using the Security Set command.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during serial programming. However, blocks can be erased by means of self-programming.

- Disabling write
  Execution of the write command for entire blocks in the flash memory is prohibited during serial programming. However, blocks can be written by means of self-programming.
  After the setting of prohibition of writing is specified, releasing the setting by the Security Release command is enabled by a reset.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting.

The block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by serial programming and self-programming. Each security setting can be used in combination.
Table 22 - 12 shows the relationship between the erase and write commands when R9A02G015 security function is enabled.
After the security settings are specified, releasing the security settings by the Security Release command is enabled by a reset.

Caution **The security function of the flash programmer does not support self-programming.**

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **22.6.3** for detail).

Table 22 - 12 Relationship Between Enabling Security Function and Command

(1) During serial programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks cannot be erased.	Can be performed. Note
Prohibition of writing	Blocks can be erased.	Cannot be performed.
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

(2) During self-programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks can be erased.	Can be performed.
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **22.6.3** for detail).

Table 22 - 13 Setting Security in Each Programming Mode

(1) During serial programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of writing		Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

Caution Releasing the setting of prohibition of writing is enabled only when the security is not set as the block erase prohibition and the boot cluster 0 rewrite prohibition with code flash memory area and data flash memory area being blanks.

(2) During self-programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set by using flash self-programming library.	Cannot be disabled after set.
Prohibition of writing		Cannot be disabled during self-programming (set via GUI of dedicated flash memory programmer, etc. during serial programming).
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

22.8 Data Flash

22.8.1 Data flash overview

An overview of the data flash memory is provided below.

- The user program can rewrite the data flash memory by using the flash data library. For details, refer to RL78 Family Flash Data Library User's Manual.
- The data flash memory can also be rewritten to through serial programming using the dedicated flash memory programmer or an external device.
- The data flash can be erased in 1-block (1 KB) units.
- The data flash can be accessed only in 8-bit units.
- The data flash can be directly read by CPU instructions.
- Instructions can be executed from the code flash memory while rewriting the data flash memory (that is, background operation (BGO) is supported).
- Because the data flash memory is an area exclusively used for data, it cannot be used to execute instructions.
- Accessing the data flash memory is not possible while rewriting the code flash memory (during self-programming).
- Manipulating the DFLCTL register is not possible while rewriting the data flash memory.
- Transition to the STOP mode is not possible while rewriting the data flash memory.

Caution 1. The data flash memory is stopped after a reset is canceled. The data flash control register (DFLCTL) must be set up in order to use the data flash memory.

Caution 2. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is stopped, its clock should be operated (HIOSTOP = 0), and the data flash library should be executed after 65 μs have elapsed.

Remark For the flash programming mode, see **22.6 Self-Programming**.

22.8.2 Register controlling data flash memory

22.8.2.1 Data flash control register (DFLCTL)

This register is used to enable or disable accessing to the data flash.

The DFLCTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 22 - 12 Format of Data flash control register (DFLCTL)

Address: F0090H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash access control
0	Disables data flash access
1	Enables data flash access

Caution Manipulating the DFLCTL register is not possible while rewriting the data flash memory.

22.8.3 Procedure for accessing data flash memory

The data flash memory is initially stopped after a reset ends and cannot be accessed (read or programmed). To access the memory, perform the following procedure:

<1> Write 1 to bit 0 (DFLEN) of the data flash control register (DFLCTL).
<2> Wait for the setup to finish for software timer, etc.
<Setup time for main clock mode>
• HS (high-speed main) mode: 5 μs
<3> After the wait, the data flash memory can be accessed.

Caution 1. Accessing the data flash memory is not possible during the setup time.
Caution 2. Transition to the STOP mode is not possible during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.
Caution 3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopped, its clock should be operated (HIOSTOP = 0), and the data flash library should be executed after 65 μs have elapsed.

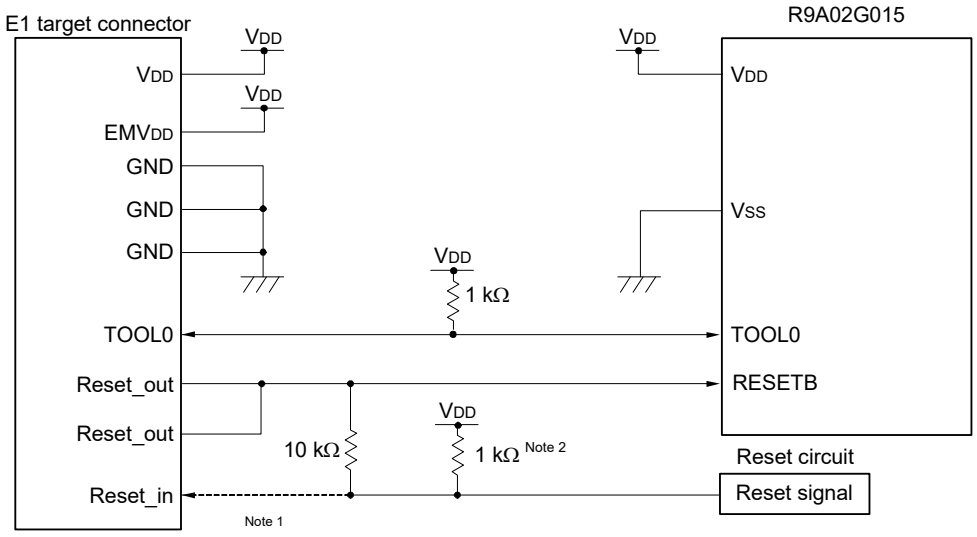
CHAPTER 23 ON-CHIP DEBUG FUNCTION

23.1 Connecting E1 On-chip Debugging Emulator

R9A02G015 uses the VDD, RESETB, TOOL0, and VSS pins to communicate with the host machine via an E1 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

Caution R9A02G015 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 23 - 1 Connection Example of E1 On-chip Debugging Emulator



Note 1. Connecting the dotted line is not necessary during serial programming.

Note 2. If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

Caution This circuit diagram is assumed that the reset signal outputs from an N-ch O.D. buffer (output resistor: 100 Ω or less)

23.2 On-Chip Debug Security ID

R9A02G015 has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 21 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 010C3H and 010C4H to 010CDH in advance, because 000C3H, 000C4H to 000CDH and 010C3H, and 010C4H to 010CDH are switched.

Table 23 - 1 On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes (except All FFH)
010C4H to 010CDH	

23.3 Securing of User Resources

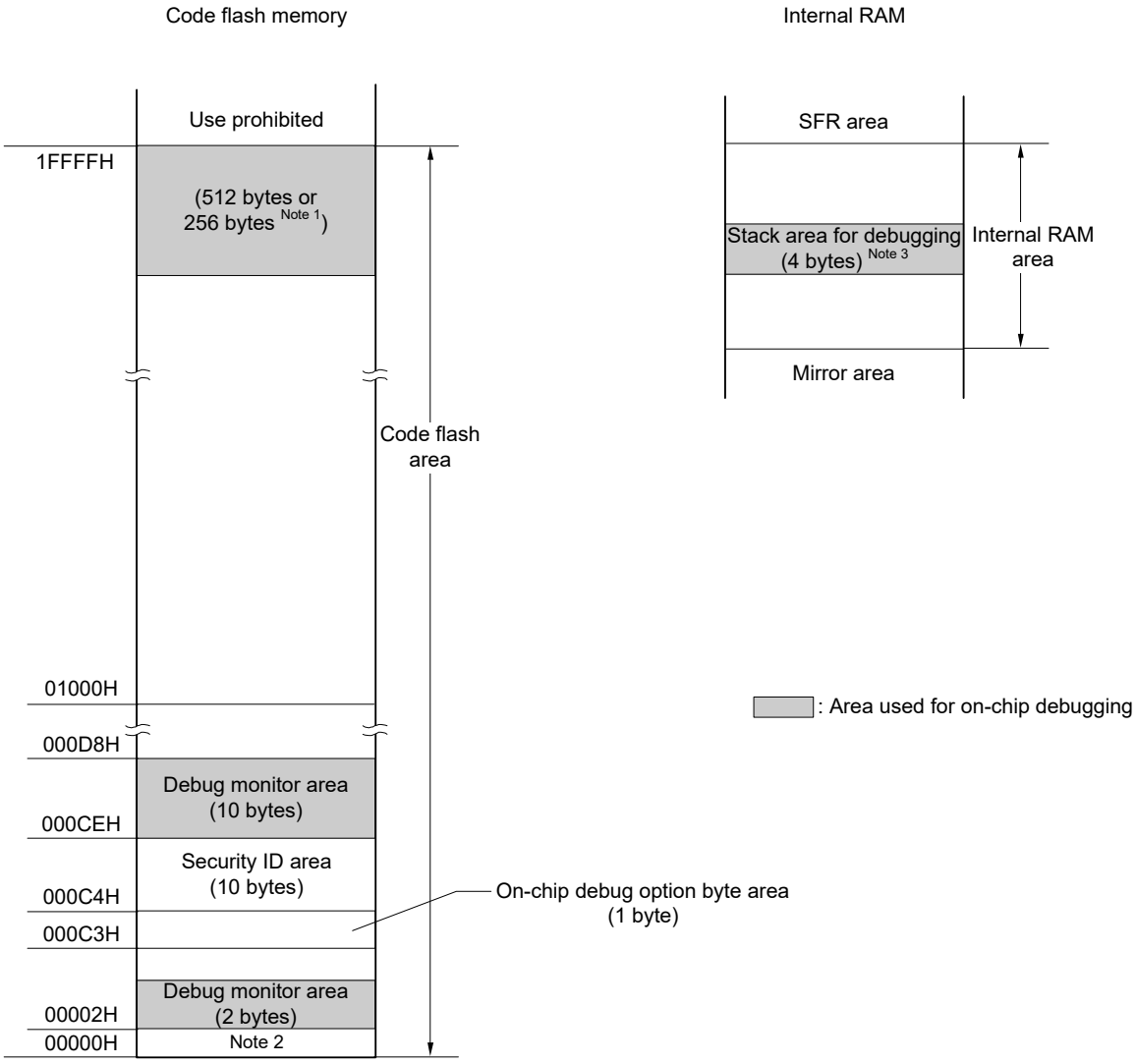
To perform communication between R9A02G015 and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using link options.

(1) Securement of memory space

The shaded portions in Figure 23 - 2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 23 - 2 Memory Spaces Where Debug Monitor Programs Are Allocated



Note 1. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

Note 2. In debugging, reset vector is rewritten to address allocated to a monitor program.

Note 3. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used.
When using self-programming, 12 extra bytes are consumed for the stack area used.

CHAPTER 24 BCD CORRECTION CIRCUIT

24.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCD correction result register (BCDADJ).

24.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

24.2.1 BCD correction result register (BCDADJ)

The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.

The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.

The BCDADJ register is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

Figure 24 - 1 Format of BCD correction result register (BCDADJ)

Address: F00FEH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

24.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Examples 1: 99 + 89 = 188

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #99H	; <1>	99H	—	—	—
ADD	A, #89H	; <2>	22H	1	1	66H
ADD	A, !BCDADJ	; <3>	88H	1	0	—

Examples 2: 85 + 15 = 100

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #85H	; <1>	85H	—	—	—
ADD	A, #15H	; <2>	9AH	0	0	66H
ADD	A, !BCDADJ	; <3>	00H	1	1	—

Examples 3: 80 + 80 = 160

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #80H	; <1>	80H	—	—	—
ADD	A, #80H	; <2>	00H	1	0	60H
ADD	A, !BCDADJ	; <3>	60H	1	0	—

(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 - 52 = 39

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #91H	; <1>	91H	—	—	—
SUB	A, #52H	; <2>	3FH	0	1	06H
SUB	A, !BCDADJ	; <3>	39H	0	0	—

CHAPTER 25 INSTRUCTION SET

This chapter lists the instructions in the R9A02G015 instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family User's Manual Software (R01US0015)**.

25.1 Conventions Used in Operation List

25.1.1 Operand identifiers and specification methods

Operands are described in the "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, $, $!, [ ], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- $: 8-bit relative address specification
- $!: 16-bit relative address specification
- [ ]: Indirect address specification
- ES:: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, $, $!, [ ], and ES: symbols.
For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 25 - 1 Operand Identifiers and Specification Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol (SFR symbol) FFF00H to FFFFFH
sfrp	Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only Note) FFF00H to FFFFFH
saddr	FFE20H to FFF1FH Immediate data or labels
saddrp	FFE20H to FF1FH Immediate data or labels (even addresses only Note)
addr20	00000H to FFFFFH Immediate data or labels
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions Note)
addr5	0080H to 00BFH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Table 3 - 5 Special Function Register (SFR) List** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Table 3 - 6 Extended Special Function Register (2nd SFR) List** for the symbols of the extended special function registers.

25.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 25 - 2 Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
∨	Exclusive logical sum (exclusive OR)
—	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

25.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

Table 25 - 3 Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
×	Set/cleared according to the result
R	Previously saved value is restored

25.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.
A interrupt is not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 25 - 4 Use Example of PREFIX Operation Code

Instruction	Opcode				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	—
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	—	—	—	—
MOV A, ES: [HL]	11H	8BH	—	—	—

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

25.2 Operation List

Table 25 - 5 Operation List (1/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	—	r ← byte			
		PSW, #byte	3	3	—	PSW ← byte	×	×	×
		CS, #byte	3	1	—	CS ← byte			
		ES, #byte	2	1	—	ES ← byte			
		!addr16, #byte	4	1	—	(addr16) ← byte			
		ES:!addr16, #byte	5	2	—	(ES, addr16) ← byte			
		saddr, #byte	3	1	—	(saddr) ← byte			
		sfr, #byte	3	1	—	sfr ← byte			
		[DE+byte], #byte	3	1	—	(DE + byte) ← byte			
		ES:[DE+byte], #byte	4	2	—	((ES, DE) + byte) ← byte			
		[HL+byte], #byte	3	1	—	(HL + byte) ← byte			
		ES:[HL+byte], #byte	4	2	—	((ES, HL) + byte) ← byte			
		[SP+byte], #byte	3	1	—	(SP + byte) ← byte			
		word[B], #byte	4	1	—	(B + word) ← byte			
		ES:word[B], #byte	5	2	—	((ES, B) + word) ← byte			
		word[C], #byte	4	1	—	(C+word) ← byte			
		ES:word[C], #byte	5	2	—	((ES, C) + word) ← byte			
		word[BC], #byte	4	1	—	(BC+word) ← byte			
		ES:word[BC], #byte	5	2	—	((ES, BC) + word) ← byte			
		A, r Note 3	1	1	—	A ← r			
		r, A Note 3	1	1	—	r ← A			
		A, PSW	2	1	—	A ← PSW			
		PSW, A	2	3	—	PSW ← A	×	×	×
		A, CS	2	1	—	A ← CS			
		CS, A	2	1	—	CS ← A			
		A, ES	2	1	—	A ← ES			
		ES, A	2	1	—	ES ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		A, ES:!addr16	4	2	5	A ← (ES, addr16)			
		!addr16, A	3	1	—	(addr16) ← A			
		ES:!addr16, A	4	2	—	(ES, addr16) ← A			
		A, saddr	2	1	—	A ← (saddr)			
		saddr, A	2	1	—	(saddr) ← A			
		A, sfr	2	1	—	A ← sfr			
		sfr, A	2	1	—	sfr ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	—	(DE) ← A			
		A, ES:[DE]	2	2	5	A ← (ES, DE)			
		ES:[DE], A	2	2	—	(ES, DE) ← A			
		A, [HL]	1	1	4	A ← (HL)			
		[HL], A	1	1	—	(HL) ← A			
		A, ES:[HL]	2	2	5	A ← (ES, HL)			
		ES:[HL], A	2	2	—	(ES, HL) ← A			
		A, [DE+byte]	2	1	4	A ← (DE + byte)			

Table 25 - 5 Operation List (2/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	[DE+byte], A	2	1	—	(DE + byte) ← A			
		A, ES:[DE+byte]	3	2	5	A ← ((ES, DE) + byte)			
		ES:[DE+byte], A	3	2	—	((ES, DE) + byte ← A			
		A, [HL+byte]	2	1	4	A ← (HL + byte)			
		[HL+byte], A	2	1	—	(HL + byte) ← A			
		A, ES:[HL+byte]	3	2	5	A ← ((ES, HL) + byte)			
		ES:[HL+byte], A	3	2	—	((ES, HL) + byte) ← A			
		A, [SP+byte]	2	1	—	A ← (SP + byte)			
		[SP+byte], A	2	1	—	(SP + byte) ← A			
		A, word[B]	3	1	4	A ← (B + word)			
		word[B], A	3	1	—	(B + word) ← A			
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)			
		ES:word[B], A	4	2	—	((ES, B) + word) ← A			
		A, word[C]	3	1	4	A ← (C + word)			
		word[C], A	3	1	—	(C + word) ← A			
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)			
		ES:word[C], A	4	2	—	((ES, C) + word) ← A			
		A, word[BC]	3	1	4	A ← (BC + word)			
		word[BC], A	3	1	—	(BC + word) ← A			
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)			
		ES:word[BC], A	4	2	—	((ES, BC) + word) ← A			
		A, [HL+B]	2	1	4	A ← (HL + B)			
		[HL+B], A	2	1	—	(HL + B) ← A			
		A, ES:[HL+B]	3	2	5	A ← ((ES, HL) + B)			
		ES:[HL+B], A	3	2	—	((ES, HL) + B) ← A			
		A, [HL+C]	2	1	4	A ← (HL + C)			
		[HL+C], A	2	1	—	(HL + C) ← A			
		A, ES:[HL+C]	3	2	5	A ← ((ES, HL) + C)			
		ES:[HL+C], A	3	2	—	((ES, HL) + C) ← A			
		X, !addr16	3	1	4	X ← (addr16)			
		X, ES:!addr16	4	2	5	X ← (ES, addr16)			
		X, saddr	2	1	—	X ← (saddr)			
		B, !addr16	3	1	4	B ← (addr16)			
		B, ES:!addr16	4	2	5	B ← (ES, addr16)			
		B, saddr	2	1	—	B ← (saddr)			
		C, !addr16	3	1	4	C ← (addr16)			
		C, ES:!addr16	4	2	5	C ← (ES, addr16)			
		C, saddr	2	1	—	C ← (saddr)			
		ES, saddr	3	1	—	ES ← (saddr)			
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	—	A ←→ r			
		A, !addr16	4	2	—	A ←→ (addr16)			
		A, ES:!addr16	5	3	—	A ←→ (ES, addr16)			
		A, saddr	3	2	—	A ←→ (saddr)			
		A, sfr	3	2	—	A ←→ sfr			
		A, [DE]	2	2	—	A ←→ (DE)			

Table 25 - 5 Operation List (3/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	XCH	A, ES:[DE]	3	3	—	A ←→ (ES, DE)			
		A, [HL]	2	2	—	A ←→ (HL)			
		A, ES:[HL]	3	3	—	A ←→ (ES, HL)			
		A, [DE+byte]	3	2	—	A ←→ (DE + byte)			
		A, ES:[DE+byte]	4	3	—	A ←→ ((ES, DE) + byte)			
		A, [HL+byte]	3	2	—	A ←→ (HL + byte)			
		A, ES:[HL+byte]	4	3	—	A ←→ ((ES, HL) + byte)			
		A, [HL+B]	2	2	—	A ←→ (HL + B)			
		A, ES:[HL+B]	3	3	—	A ←→ ((ES, HL) + B)			
		A, [HL+C]	2	2	—	A ←→ (HL + C)			
		A, ES:[HL+C]	3	3	—	A ←→ ((ES, HL) + C)			
	ONEB	A	1	1	—	A ← 01H			
		X	1	1	—	X ← 01H			
		B	1	1	—	B ← 01H			
		C	1	1	—	C ← 01H			
		!addr16	3	1	—	(addr16) ← 01H			
		ES:!addr16	4	2	—	(ES, addr16) ← 01H			
		saddr	2	1	—	(saddr) ← 01H			
	CLRB	A	1	1	—	A ← 00H			
		X	1	1	—	X ← 00H			
		B	1	1	—	B ← 00H			
		C	1	1	—	C ← 00H			
		!addr16	3	1	—	(addr16) ← 00H			
		ES:!addr16	4	2	—	(ES,addr16) ← 00H			
		saddr	2	1	—	(saddr) ← 00H			
	MOVS	[HL+byte], X	3	1	—	(HL + byte) ← X	×		×
		ES:[HL+byte], X	4	2	—	(ES, HL + byte) ← X	×		×
16-bit data transfer	MOVW	rp, #word	3	1	—	rp ← word			
		saddrp, #word	4	1	—	(saddrp) ← word			
		sfrp, #word	4	1	—	sfrp ← word			
		AX, rp Note 4	1	1	—	AX ← rp			
		rp, AX Note 4	1	1	—	rp ← AX			
		AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	—	(addr16) ← AX			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	—	(ES, addr16) ← AX			
		AX, saddrp	2	1	—	AX ← (saddrp)			
		saddrp, AX	2	1	—	(saddrp) ← AX			
		AX, sfrp	2	1	—	AX ← sfrp			
		sfrp, AX	2	1	—	sfrp ← AX			
		AX, [DE]	1	1	4	AX ← (DE)			
		[DE], AX	1	1	—	(DE) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	—	(ES, DE) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	—	(HL) ← AX			

Table 25 - 5 Operation List (4/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, ES:[HL]	2	2	5	AX ← (ES, HL)			
		ES:[HL], AX	2	2	—	(ES, HL) ← AX			
		AX, [DE+byte]	2	1	4	AX ← (DE + byte)			
		[DE+byte], AX	2	1	—	(DE + byte) ← AX			
		AX, ES:[DE+byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE+byte], AX	3	2	—	((ES, DE) + byte) ← AX			
		AX, [HL+byte]	2	1	4	AX ← (HL + byte)			
		[HL+byte], AX	2	1	—	(HL + byte) ← AX			
		AX, ES:[HL+byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL+byte], AX	3	2	—	((ES, HL) + byte) ← AX			
		AX, [SP+byte]	2	1	—	AX ← (SP + byte)			
		[SP+byte], AX	2	1	—	(SP + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	—	(B + word) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	—	((ES, B) + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	—	(C + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	—	((ES, C) + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	—	(BC + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	—	((ES, BC) + word) ← AX			
		BC, !addr16	3	1	4	BC ← (addr16)			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
		BC, saddrp	2	1	—	BC ← (saddrp)			
		DE, saddrp	2	1	—	DE ← (saddrp)			
		HL, saddrp	2	1	—	HL ← (saddrp)			
	XCHW	AX, rp Note 4	1	1	—	AX ←→ rp			
	ONEW	AX	1	1	—	AX ← 0001H			
		BC	1	1	—	BC ← 0001H			
	CLRW	AX	1	1	—	AX ← 0000H			
		BC	1	1	—	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	—	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 3	2	1	—	A, CY ← A + r	×	×	×
		r, A	2	1	—	r, CY ← r + A	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×

Table 25 - 5 Operation List (5/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADD	A, saddr	2	1	—	A, C ← A + (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A + ((ES, HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×
	ADDC	A, #byte	2	1	—	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, rv Note 3	2	1	—	A, CY ← A + r + CY	×	×	×
		r, A	2	1	—	r, CY ← r + A + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, saddr	2	1	—	A, CY ← A + (saddr) + CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL) + CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL + byte) + CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL + B) + CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A + ((ES, HL) + B) + CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL + C) + CY	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C) + CY	×	×	×
	SUB	A, #byte	2	1	—	A, CY ← A - byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) - byte	×	×	×
		A, r Note 3	2	1	—	A, CY ← A - r	×	×	×
		r, A	2	1	—	r, CY ← r - A	×	×	×
		A, !addr16	3	1	4	A, CY ← A - (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A - (ES, addr16)	×	×	×
		A, saddr	2	1	—	A, CY ← A - (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A - (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A - (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A - (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A - ((ES, HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A - (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A - ((ES, HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A - (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A - ((ES, HL) + C)	×	×	×
	SUBC	A, #byte	2	1	—	A, CY ← A - byte - CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) - byte - CY	×	×	×
		A, r Note 3	2	1	—	A, CY ← A - r - CY	×	×	×
		r, A	2	1	—	r, CY ← r - A - CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A - (addr16) - CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A - (ES, addr16) - CY	×	×	×
		A, saddr	2	1	—	A, CY ← A - (saddr) - CY	×	×	×

Table 25 - 5 Operation List (6/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, [HL]	1	1	4	A, CY ← A - (HL) - CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A - (ES, HL) - CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A - (HL + byte) - CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A - ((ES, HL) + byte) - CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A - (HL + B) - CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A - ((ES, HL) + B) - CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A - (HL + C) - CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY ← A - ((ES:HL) + C) - CY	×	×	×
	AND	A, #byte	2	1	—	A ← A ∧ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	—	A ← A ∧ r	×		
		r, A	2	1	—	R ← r ∧ A	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ∧ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∧ (HL + byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∧ ((ES:HL) + byte)	×		
		A, [HL+B]	2	1	4	A ← A ∧ (HL + B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∧ ((ES:HL) + B)	×		
		A, [HL+C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL) + C)	×		
	OR	A, #byte	2	1	—	A ← A ∨ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	1	—	A ← A ∨ r	×		
		r, A	2	1	—	r ← r ∨ A	×		
		A, !addr16	3	1	4	A ← A ∨ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∨ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ∨ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∨ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∨ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∨ (HL + byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∨ ((ES:HL) + byte)	×		
		A, [HL+B]	2	1	4	A ← A ∨ (HL + B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∨ ((ES:HL) + B)	×		
		A, [HL+C]	2	1	4	A ← A ∨ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∨ ((ES:HL) + C)	×		
	XOR	A, #byte	2	1	—	A ← A ⩒ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ⩒ byte	×		
		A, r Note 3	2	1	—	A ← A ⩒ r	×		
		r, A	2	1	—	r ← r ⩒ A	×		
		A, !addr16	3	1	4	A ← A ⩒ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ⩒ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ⩒ (saddr)	×		
		A, [HL]	1	1	4	A ← A ⩒ (HL)	×		

Table 25 - 5 Operation List (7/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	XOR	A, ES:[HL]	2	2	5	A ← A ⩛ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ⩛ (HL + byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ⩛ ((ES:HL) + byte)	×		
		A, [HL+B]	2	1	4	A ← A ⩛ (HL + B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ⩛ ((ES:HL) + B)	×		
		A, [HL+C]	2	1	4	A ← A ⩛ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ⩛ ((ES:HL) + C)	×		
	CMP	A, #byte	2	1	—	A - byte	×	×	×
		!addr16, #byte	4	1	4	(addr16) - byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) - byte	×	×	×
		saddr, #byte	3	1	—	(saddr) - byte	×	×	×
		A, r Note 3	2	1	—	A - r	×	×	×
		r, A	2	1	—	r - A	×	×	×
		A, !addr16	3	1	4	A - (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A - (ES:addr16)	×	×	×
		A, saddr	2	1	—	A - (saddr)	×	×	×
		A, [HL]	1	1	4	A - (HL)	×	×	×
		A, ES:[HL]	2	2	5	A - (ES:HL)	×	×	×
		A, [HL+byte]	2	1	4	A - (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A - ((ES:HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A - (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A - ((ES:HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A - (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A - ((ES:HL) + C)	×	×	×
	CMP0	A	1	1	—	A - 00H	×	0	0
		X	1	1	—	X - 00H	×	0	0
		B	1	1	—	B - 00H	×	0	0
		C	1	1	—	C - 00H	×	0	0
		!addr16	3	1	4	(addr16) - 00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16) - 00H	×	0	0
		saddr	2	1	—	(saddr) - 00H	×	0	0
	CMPS	X, [HL+byte]	3	1	4	X - (HL + byte)	×	×	×
		X, ES:[HL+byte]	4	2	5	X - ((ES:HL) + byte)	×	×	×
16-bit operation	ADDW	AX, #word	3	1	—	AX, CY ← AX + word	×	×	×
		AX, AX	1	1	—	AX, CY ← AX + AX	×	×	×
		AX, BC	1	1	—	AX, CY ← AX + BC	×	×	×
		AX, DE	1	1	—	AX, CY ← AX + DE	×	×	×
		AX, HL	1	1	—	AX, CY ← AX + HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX + (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY ← AX + (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX + ((ES:HL) + byte)	×	×	×

Table 25 - 5 Operation List (8/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	SUBW	AX, #word	3	1	—	AX, CY ← AX - word	×	×	×
		AX, BC	1	1	—	AX, CY ← AX - BC	×	×	×
		AX, DE	1	1	—	AX, CY ← AX - DE	×	×	×
		AX, HL	1	1	—	AX, CY ← AX - HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX - (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX - (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY ← AX - (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX - (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX - ((ES:HL) + byte)	×	×	×
	CMPW	AX, #word	3	1	—	AX - word	×	×	×
		AX, BC	1	1	—	AX - BC	×	×	×
		AX, DE	1	1	—	AX - DE	×	×	×
		AX, HL	1	1	—	AX - HL	×	×	×
		AX, !addr16	3	1	4	AX - (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX - (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX - (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX - (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX - ((ES:HL) + byte)	×	×	×
Multiply, Divide, Multiply & accumulate	MULU	X	1	1	—	AX ← A × X			
	MULHU		3	2	—	BCAX ← AX × BC (unsigned)			
	MULH		3	2	—	BCAX ← AX × BC (signed)			
	DIVHU		3	9	—	AX (quotient), DE (remainder) ← AX ÷ DE (unsigned)			
	DIVWU		3	17	—	BCAX (quotient), HLDE (remainder) ← BCAX ÷ HLDE (unsigned)			
	MACHU		3	3	—	MACR ← MACR + AX × BC (unsigned)		×	×
	MACH		3	3	—	MACR ← MACR + AX × BC(signed)		×	×
Increment/ decrement	INC	r	1	1	—	r ← r + 1	×	×	
		!addr16	3	2	—	(addr16) ← (addr16) + 1	×	×	
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) + 1	×	×	
		saddr	2	2	—	(saddr) ← (saddr) + 1	×	×	
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) + 1	×	×	
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) + 1	×	×	
	DEC	r	1	1	—	r ← r - 1	×	×	
		!addr16	3	2	—	(addr16) ← (addr16) - 1	×	×	
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) - 1	×	×	
		saddr	2	2	—	(saddr) ← (saddr) - 1	×	×	
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) - 1	×	×	
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) - 1	×	×	
	INCW	rp	1	1	—	rp ← rp + 1			
		!addr16	3	2	—	(addr16) ← (addr16) + 1			
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) + 1			
		saddrp	2	2	—	(saddrp) ← (saddrp) + 1			
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) + 1			
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) + 1			

Table 25 - 5 Operation List (9/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/ decrement	DECW	rp	1	1	—	rp ← rp - 1			
		!addr16	3	2	—	(addr16) ← (addr16) - 1			
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) - 1			
		saddrp	2	2	—	(saddrp) ← (saddrp) - 1			
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) - 1			
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) - 1			
Shift	SHR	A, cnt	2	1	—	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) × cnt			×
	SHRW	AX, cnt	2	1	—	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) × cnt			×
	SHL	A, cnt	2	1	—	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) × cnt			×
		B, cnt	2	1	—	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) × cnt			×
		C, cnt	2	1	—	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) × cnt			×
	SHLW	AX, cnt	2	1	—	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) × cnt			×
		BC, cnt	2	1	—	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) × cnt			×
	SAR	A, cnt	2	1	—	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) × cnt			×
	SARW	AX, cnt	2	1	—	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) × cnt			×
Rotate	ROR	A, 1	2	1	—	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1			×
	ROL	A, 1	2	1	—	(CY, A_0 ← A7, A_{m+1} ← A_m) × 1			×
	RORC	A, 1	2	1	—	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
	ROLC	A, 1	2	1	—	(CY ← A7, A_0 ← CY, A_{m+1} ← A_m) × 1			×
	ROLWC	AX,1	2	1	—	(CY ← AX_{15}, AX_0 ← CY, AX_{m+1} ← AX_m) × 1			×
		BC,1	2	1	—	(CY ← BC_{15}, BC_0 ← CY, BC_{m+1} ← BCm) × 1			×
Bit manipulate	MOV1	CY, A.bit	2	1	—	CY ← A.bit			×
		A.bit, CY	2	1	—	A.bit ← CY			
		CY, PSW.bit	3	1	—	CY ← PSW.bit			×
		PSW.bit, CY	3	4	—	PSW.bit ← CY	×	×	
		CY, saddr.bit	3	1	—	CY ← (saddr).bit			×
		saddr.bit, CY	3	2	—	(saddr).bit ← CY			
		CY, sfr.bit	3	1	—	CY ← sfr.bit			×
		sfr.bit, CY	3	2	—	sfr.bit ← CY			
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		[HL].bit, CY	2	2	—	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	—	(ES, HL).bit ← CY			
	AND1	CY, A.bit	2	1	—	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∧ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∧ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, A.bit	2	1	—	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∨ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∨ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Table 25 - 5 Operation List (10/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, A.bit	2	1	—	CY ← CY ⩛ bit			×
		CY, PSW.bit	3	1	—	CY ← CY ⩛ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ⩛ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ⩛ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⩛ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⩛ (ES, HL).bit			×
	SET1	A.bit	2	1	—	A.bit ← 1			
		PSW.bit	3	4	—	PSW.bit ← 1	×	×	×
		!addr16.bit	4	2	—	(addr16).bit ← 1			
		ES:!addr16.bit	5	3	—	(ES, addr16).bit ← 1			
		saddr.bit	3	2	—	(saddr).bit ← 1			
		sfr.bit	3	2	—	sfr.bit ← 1			
		[HL].bit	2	2	—	(HL).bit ← 1			
		ES:[HL].bit	3	3	—	(ES, HL).bit ← 1			
	CLR1	A.bit	2	1	—	A.bit ← 0			
		PSW.bit	3	4	—	PSW.bit ← 0	×	×	×
		!addr16.bit	4	2	—	(addr16).bit ← 0			
		ES:!addr16.bit	5	3	—	(ES, addr16).bit ← 0			
		saddr.bit	3	2	—	(saddr.bit) ← 0			
		sfr.bit	3	2	—	sfr.bit ← 0			
		[HL].bit	2	2	—	(HL).bit ← 0			
		ES:[HL].bit	3	3	—	(ES, HL).bit ← 0			
	SET1	CY	2	1	—	CY ← 1			1
	CLR1	CY	2	1	—	CY ← 0			0
	NOT1	CY	2	1	—	CY ← $\overline{CY}$			×
Call/return	CALL	rp	2	3	—	(SP - 2) ← (PC + 2)S, (SP - 3) ← (PC + 2)H, (SP - 4) ← (PC + 2)L, PC ← CS, rp, SP ← SP - 4			
		$!addr20	3	3	—	(SP - 2) ← (PC + 3)S, (SP - 3) ← (PC + 3)H, (SP - 4) ← (PC + 3)L, PC ← PC + 3 + jdisp16, SP ← SP - 4			
		!addr16	3	3	—	(SP - 2) ← (PC + 3)S, (SP - 3) ← (PC + 3)H, (SP - 4) ← (PC + 3)L, PC ← 0000, addr16, SP ← SP - 4			
		!!addr20	4	3	—	(SP - 2) ← (PC + 4)S, (SP - 3) ← (PC + 4)H, (SP - 4) ← (PC + 4)L, PC ← addr20, SP ← SP - 4			
	CALLT	[addr5]	2	5	—	(SP - 2) ← (PC + 2)S, (SP - 3) ← (PC + 2)H, (SP - 4) ← (PC + 2)L , PCS ← 0000, PCH ← (0000, addr5 + 1), PCL ← (0000, addr5), SP ← SP - 4			
	BRK	—	2	5	—	(SP - 1) ← PSW, (SP - 2) ← (PC + 2)S, (SP - 3) ← (PC + 2)H, (SP - 4) ← (PC + 2)L, PCS ← 0000, PCH ← (0007FH), PCL ← (0007EH), SP ← SP - 4, IE ← 0			
	RET	—	1	6	—	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), SP ← SP + 4			
	RETI	—	2	6	—	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

Table 25 - 5 Operation List (11/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	RETB	—	2	6	—	$PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP + 1)$, $PC_S \leftarrow (SP + 2)$, $PSW \leftarrow (SP + 3)$, $SP \leftarrow SP + 4$	R	R	R
Stack manipulate	PUSH	PSW	2	1	—	$(SP - 1) \leftarrow PSW$, $(SP - 2) \leftarrow 00H$, $SP \leftarrow SP - 2$			
		rp	1	1	—	$(SP - 1) \leftarrow rp_H$, $(SP - 2) \leftarrow rp_L$, $SP \leftarrow SP - 2$			
	POP	PSW	2	3	—	$PSW \leftarrow (SP + 1)$, $SP \leftarrow SP + 2$	R	R	R
		rp	1	1	—	$rp_L \leftarrow (SP)$, $rp_H \leftarrow (SP + 1)$, $SP \leftarrow SP + 2$			
	MOVW	SP, #word	4	1	—	$SP \leftarrow$ word			
		SP, AX	2	1	—	$SP \leftarrow AX$			
		AX, SP	2	1	—	$AX \leftarrow SP$			
		HL, SP	3	1	—	$HL \leftarrow SP$			
		BC, SP	3	1	—	$BC \leftarrow SP$			
		DE, SP	3	1	—	$DE \leftarrow SP$			
	ADDW	SP, #byte	2	1	—	$SP \leftarrow SP +$ byte			
	SUBW	SP, #byte	2	1	—	$SP \leftarrow SP -$ byte			
Unconditional branch	BR	AX	2	3	—	$PC \leftarrow CS, AX$			
		$addr20	2	3	—	$PC \leftarrow PC + 2 +$ jdisp8			
		$!addr20	3	3	—	$PC \leftarrow PC + 3 +$ jdisp16			
		!addr16	3	3	—	$PC \leftarrow 0000$, addr16			
		!!addr20	4	3	—	$PC \leftarrow$ addr20			
Conditional branch	BC	$addr20	2	2/4 Note 5	—	$PC \leftarrow PC + 2 +$ jdisp8 if CY = 1			
	BNC	$addr20	2	2/4 Note 5	—	$PC \leftarrow PC + 2 +$ jdisp8 if CY = 0			
	BZ	$addr20	2	2/4 Note 5	—	$PC \leftarrow PC + 2 +$ jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4 Note 5	—	$PC \leftarrow PC + 2 +$ jdisp8 if Z = 0			
	BH	$addr20	3	2/4 Note 5	—	$PC \leftarrow PC + 3 +$ jdisp8 if $(Z \vee CY) = 0$			
	BNH	$addr20	3	2/4 Note 5	—	$PC \leftarrow PC + 3 +$ jdisp8 if $(Z \vee CY) = 1$			
	BT	saddr.bit, $addr20	4	3/5 Note 5	—	$PC \leftarrow PC + 4 +$ jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5 Note 5	—	$PC \leftarrow PC + 4 +$ jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5 Note 5	—	$PC \leftarrow PC + 3 +$ jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5 Note 5	—	$PC \leftarrow PC + 4 +$ jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5 Note 5	6/7	$PC \leftarrow PC + 3 +$ jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6 Note 5	7/8	$PC \leftarrow PC + 4 +$ jdisp8 if (ES, HL).bit = 1			

Table 25 - 5 Operation List (12/12)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5 Note 5	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5 Note 5	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5 Note 5	—	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5 Note 5	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5 Note 5	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6 Note 5	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5 Note 5	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5 Note 5	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5 Note 5	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5 Note 5	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5 Note 5	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6 Note 5	—	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit			
Conditional skip	SKC	—	2	1	—	Next instruction skip if CY = 1			
	SKNC	—	2	1	—	Next instruction skip if CY = 0			
	SKZ	—	2	1	—	Next instruction skip if Z = 1			
	SKNZ	—	2	1	—	Next instruction skip if Z = 0			
	SKH	—	2	1	—	Next instruction skip if (Z ∨ CY) = 0			
	SKNH	—	2	1	—	Next instruction skip if (Z ∨ CY) = 1			
CPU control	SEL Note 6	RBn	2	1	—	RBS[1:0] ← n			
	NOP	—	1	1	—	No Operation			
	EI	—	3	4	—	IE ← 1 (Enable Interrupt)			
	DI	—	3	4	—	IE ← 0 (Disable Interrupt)			
	HALT	—	2	3	—	Set HALT Mode			
	STOP	—	2	3	—	Set STOP Mode			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Note 4. Except rp = AX

Note 5. This indicates the number of clocks “when condition is not met/when condition is met”.

Note 6. n indicates the number of register banks (n = 0 to 3)

Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine. Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.3 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

Remark 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Remark 2. MACR indicates the multiplication and accumulation register (MACRH, MACRL).

Remark 3. cnt indicates the bit shift count.

CHAPTER 26 ELECTRICAL SPECIFICATIONS

For details, refer to **R9A02G015 Data Sheet (R19DS0101E)**.

CHAPTER 27 PACKAGE DRAWINGS

For details, refer to **R9A02G015 Data Sheet (R19DS0101E)**.

REVISION HISTORY	R9A02G015 User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
0.10	Sep 28, 2018	—	First Draft of the Preliminary User's Manual
0.20	Dec 26, 2018	—	The following chapters were newly added. • CHAPTER 5 CLOCK GENERATOR • CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER • CHAPTER 11 SERIAL ARRAY UNIT • CHAPTER 12 SERIAL INTERFACE IICA • CHAPTER 13 USB HOST/FUNCTION MODULE (USB) In addition, minor modifications were made.
0.90	Feb 15, 2019	—	The following chapters were newly added. • CHAPTER 15 STANDBY FUNCTION • CHAPTER 16 RESET FUNCTION • CHAPTER 19 SAFETY FUNCTIONS • CHAPTER 20 REGULATOR • CHAPTER 21 OPTION BYTE • CHAPTER 22 FLASH MEMORY • CHAPTER 23 ON-CHIP DEBUG FUNCTION
1.00	Mar 29, 2019	—	First edition issued

R9A02G015 User's Manual: Hardware

Publication Date: Rev.0.10 Sep 28, 2018
Rev.1.00 Mar 29, 2019

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai 200333, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia
Tel: +60-3-5022-1288, Fax: +60-3-5022-1290

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

© 2019 Renesas Electronics Corporation. All rights reserved.

Colophon 6.0

R9A02G015

Renesas Electronics Corporation

R19UH0112EJ0100