User's Manual

16

RL78/G1H

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev. 1.20 Dec 2016

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended 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; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be 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, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

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

(2012.4)

NOTES FOR CMOS DEVICES

(1) 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 VIL (MAX) and VIH (MIN) due to noise, etc., 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 VIL (MAX) and VIH (MIN).

(2) HANDLING OF UNUSED INPUT PINS: Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) PRECAUTION AGAINST ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that 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 should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) STATUS BEFORE INITIALIZATION: Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) POWER ON/OFF SEQUENCE: In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) INPUT OF SIGNAL DURING POWER OFF STATE : Do not input signals or an I/O pull-up power supply while the device is not powered. 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. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the RL78/G1H and design and develop application systems and programs for these devices.

Purpose

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

Organization

The RL78/G1H manual is separated into two parts: this manual and the software edition (common to the RL78 family).

RL78/G1H User's Manual Hardware (This Manual)	**RL78 Family User's Manual Software**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

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**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.
- 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 RL78/G1H Microcontroller 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)
	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.
RL78/G1H User's Manual Hardware	This manual
RL78 Family User's Manual Software	R01US0015E

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

Document Name		Document No.
PG-FP5 Flash Memory Programmer User's Manual		—
	RL78, 78K, V850, RX100, RX200, RX600 (Except RX64x), R8C, SH	R20UT2923E
	Common	R20UT2922E
	Setup Manual	R20UT0930E

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 MPUs & MCUs 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.

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

CONTENTS

CHAPTER 1 OUTLINE

RL78/G1H is a microcontroller equipped with the low-power-consumption RF transceiver compatible with the SubGHz-band wireless communication. The wireless communication in the SubGHz band is best for the smart meter communication part, HEMS controller, wireless sensor network, etc.

1.1 Features

Ultra-low power consumption technology

- Standby function of MCU: HALT mode, STOP mode, SNOOZE mode
- Standby function of RF unit: DLE mode, SLEEP mode
  - RF operation transmission current: 21 mA (TYP.) (RF; 100 kbps, 2FSK, +10 dBm, 3.0 V/ at STOP mode of MCU)
    : 36 mA (TYP.) (RF; 100 kbps, 2FSK, +13 dBm, 3.0 V/ at STOP mode of MCU)
  - RF operation reception current: 6.9 mA (TYP.) (RF; 100 kbps, 2FSK, 3.0 V/at STOP mode of MCU)
  - RF operation SLEEP mode (POWER_DOWN mode) current: 0.1 μA (TYP.) (3.0 V/at STOP mode of MCU)

RL78 CPU core

- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: Can be changed from high speed (0.03125 μs: @ 32 MHz operation with high-speed on-chip oscillator) to ultra-low speed (30.5 μs: @ 32.768 kHz operation with subsystem clock)
- Address space: 1 MB
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 24 to 48 KB

On-chip RF transceiver

- IEEE802.15.4g standard specification SubGHz-band transceiver
- RF frequency range: 863 to 928 MHz
- Modulation method: 2FSK/GFSK, 4FSK/GFSK
- Data rate: 2FSK/GFSK 10 to 300 kbps, 4FSK/GFSK 200/400 kbps
- Forward error correction (FEC) function

Code flash memory

- Code flash memory: 256 to 512 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: 8 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} = 1.8 to 3.6 V

High-speed on-chip oscillator

- Select from 32 MHz (TYP.), 24 MHz (TYP.), 16 MHz (TYP.), 12 MHz (TYP.), 8 MHz (TYP.), 6 MHz (TYP.), 4 MHz (TYP.), 3 MHz (TYP.), 2 MHz (TYP.), and 1 MHz (TYP.)

Operating ambient temperature

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

Power management and reset function

- On-chip power-on-reset (POR) circuit
- On-chip voltage detector (LVD) (Select interrupt and reset from 10 levels)

Data transfer controller (DTC)

- Transfer modes: Normal transfer mode, repeat transfer mode, block transfer mode
- Activation sources: Activated by interrupt sources.
- Chain transfer function

Event link controller (ELC)

- Event signals of 13 types can be linked to the specified peripheral function.

Serial interface

- CSI: 4 channels (1 channel of 4 channels is used for the internal communication between MCU and RF transceiver.)
- UART: 2 channels
- I^2C: 2 channels

Timer

- 16-bit timer: 9 channels
- 12-bit interval timer: 1 channel
- Real-time clock: 1 channel (calendar for 99 years, alarm function, and clock correction function)
- Watchdog timer: 1 channel (operable with the dedicated low-speed on-chip oscillator)

A/D converter

- 10-bit resolution A/D converter (V_{DD} = 1.8 to 3.6 V)
- Analog input: 6 channels

I/O port

- I/O port: 41
- Can be set to N-ch open drain, TTL input buffer, and on-chip pull-up resistor
- Different potential interface: Can connect to a 1.8/2.5 V device
- On-chip clock output/buzzer output controller

Cipher

- AES cipher processing (128-bit key length)
- Random number generator (true random number, complies with AIS31 standard)

Others

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

○ ROM, RAM capacities

Flash ROM	Data flash	RAM	RL78/G1H
256 KB	8 KB	24 KB	R5F11FLJ
384 KB	8 KB	32 KB	R5F11FLK
512 KB	8 KB	48 KB Note	R5F11FLL

Note This is about 47 KB when the self-programming function is used (For details, see **CHAPTER 4**).

1.2 Ordering Information

Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G1H

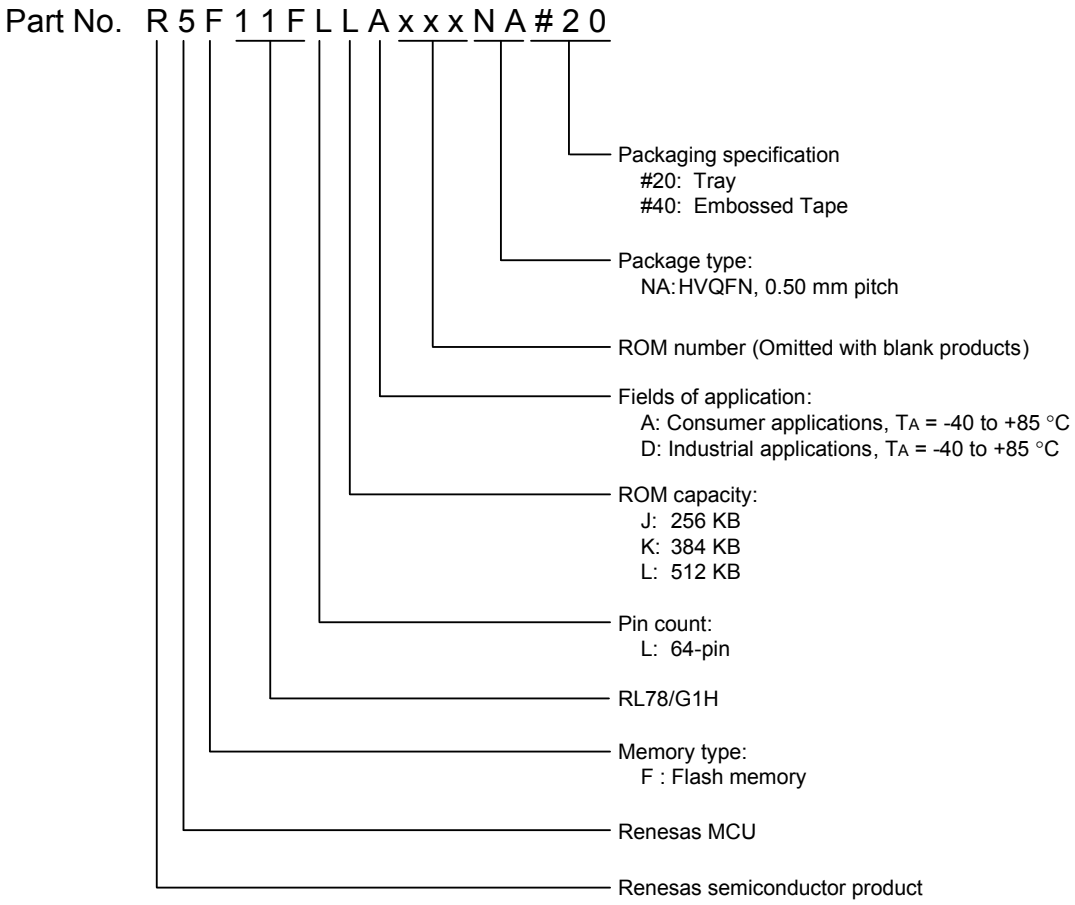


Table 1 - 1 Ordering Part Number List

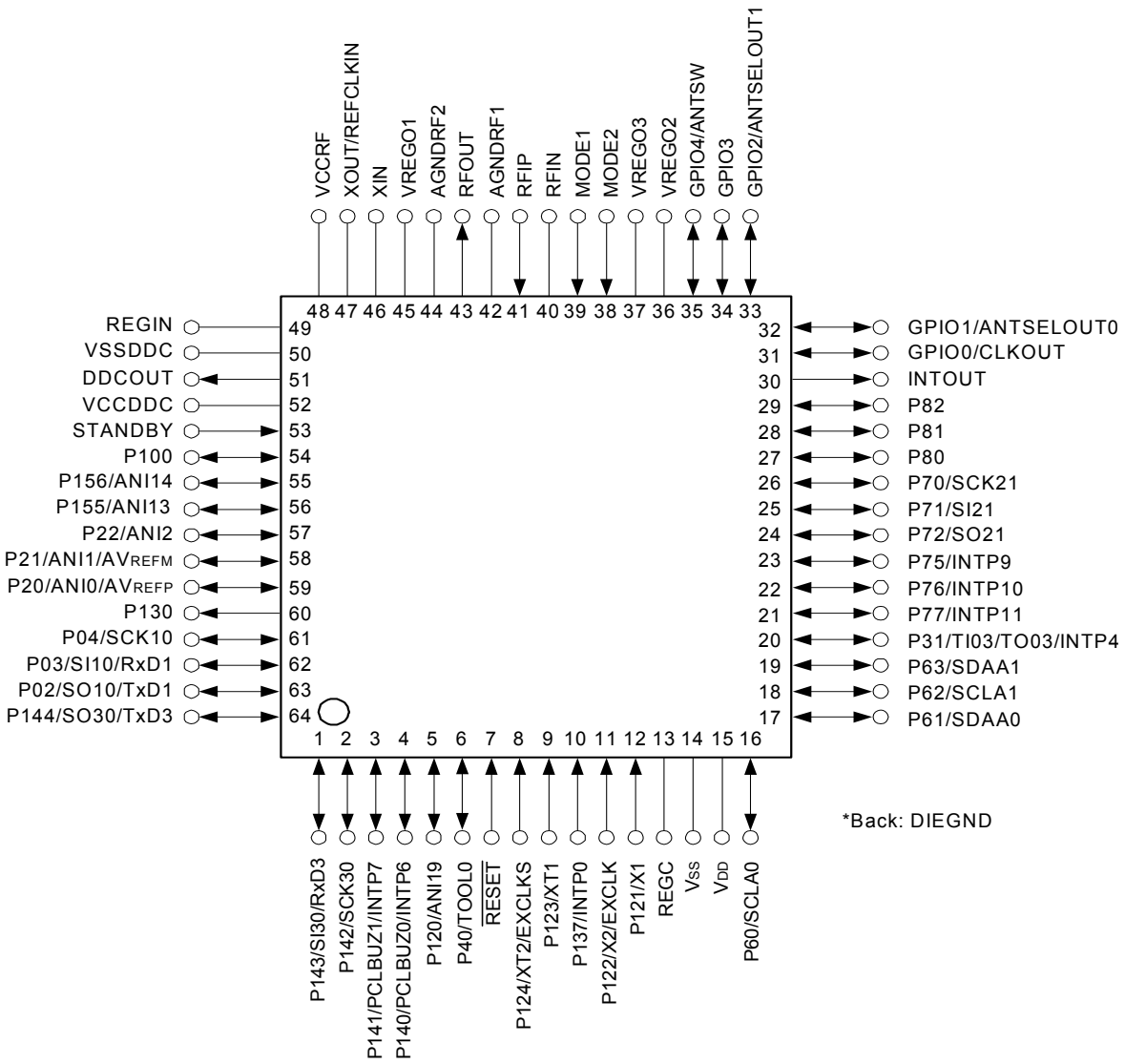
Pin count	Package	Fields of Application Note	Ordering Part Number	Code Flash Memory	Data Flash Memory
64 pins	64-pin plastic HVQFN (9 × 9)	A	R5F11FLJANA#20, R5F11FLJANA#40	256 Kbytes	8 Kbytes
		D	R5F11FLJDNA#20, R5F11FLJDNA#40		
		A	R5F11FLKANA#20, R5F11FLKANA#40	384 Kbytes	
		D	R5F11FLKDNA#20, R5F11FLKDNA#40		
		A	R5F11FLLANA#20, R5F11FLLANA#40	512 Kbytes	
		D	R5F11FLLDNA#20, R5F11FLLDNA#40		

Note For the fields of application, refer to **Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G1H**.

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)

• 64-pin plastic HVQFN (9 × 9)



<R> **Caution** **Connect the metal pad (DIEGND) on the back of the package to GND of the board.**

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

1.4 Pin Identification

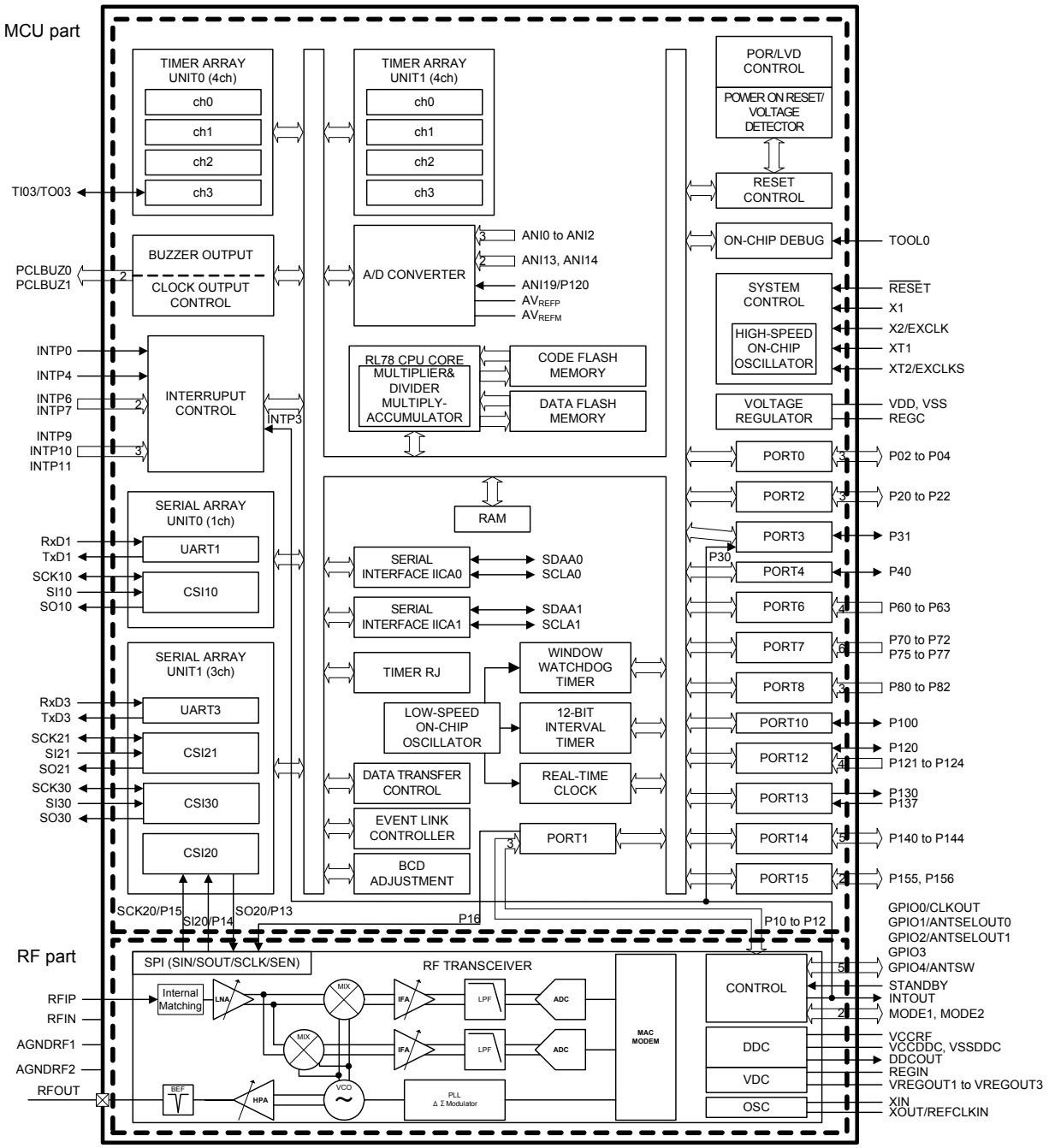
<MCU unit>

ANI0 to ANI2, ANI13:	Analog Input	P130, P137:	Port 13
ANI14, ANI19:	Analog Input	P140 to P144:	Port 14
AV_{REFM}:	Analog Reference	P155, P156:	Port 15
	Voltage Minus	PCLBUZ0, PCLBUZ1:	Programmable Clock Output/
AV_{REFP}:	Analog Reference		Buzzer Output
	Voltage Plus	REGC:	Regulator Capacitance
EXCLK:	External Clock Input	$\overline{\text{RESET}}$:	Reset
	(Main System Clock)	RxD1, RxD3:	Receive Data
EXCLKS:	External Clock Input	SCK10, SCK21,	
	(Subsystem Clock)	SCK30:	Serial Clock Input/Output
INTP0, INTP4:	External Interrupt Input	SCLA0, SCLA1:	Serial Clock Output
INTP6, INTP7:		SDAA0, SDAA1:	Serial Data Input/Output
INTP9 to INTP11:		SI10, SI21, SI30:	Serial Data Input
P02 to P04:	Port 0	SO10, SO21, SO30:	Serial Clock Output
P20 to P22:	Port 2	TI03:	Timer Input
P31:	Port 3	TO03:	Timer Output
P40:	Port 4	TOOL0:	Data Input/Output for Tool
P60 to P63:	Port 6	TxD1, TxD3:	Transmit Data
P70 to P72, P75 to P77:	Port 7	V_{DD}:	Power Supply
P80 to P82:	Port 8	V_{SS}:	Ground
P100:	Port 10	X1, X2:	Crystal Oscillator (Main System Clock)
P120 to P124:	Port 12	XT1, XT2:	Crystal Oscillator (Subsystem Clock)

<RF transceiver unit>

GPIO0-GPIO4	Transceiver I/O port	VREGO1:	Power supply stability
CLKOUT:	Clock output		capacity connection for RF
ANTSELOUT0,	Antenna select	XIN:	Buffer input for the 48 MHz
ANTSELOUT1:			X'tal oscillation
ANTSW:	Antenna switch	XOUT:	48 MHz crystal resonator output
VREGO2:	Power supply stabilization	REFCLKIN:	External clock input
	capacitor connection pin for VCO	REGIN:	Power supply input for the analog, and
VREGO3:	Power supply stability capacity		externally connect with DDCOUT
	connection for PLL	VSSDDC:	DCDC converter GND
MODE1, MODE2:	Mode switch	DDCOUT:	The DCDC converter output, to externally
RFIN:	Transceiver GND		connect with REGIN
RFIP:	RF input	VCCDDC:	DCDC converter power supply
AGNDRF1:	Transceiver GND	STANDBY:	Power down control input of the transceiver,
RFOUT:	RF output		and externally connect with P130
AGNDRF2:	Transceiver GND	INTOUT:	Interrupt output

1.5 Block Diagram



1.6 Outline of Functions

(1/2)

Item		R5F11FLJ	R5F11FLK	R5F11FLL
Code flash memory (KB)		256 KB	384 KB	512 KB
Data flash memory (KB)		8 KB	8 KB	8 KB
RAM (KB)		24 KB	32 KB	48 KB Note 1
Address space		1 MB		
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillator, external main system clock input (EXCLK) HS (high-speed main) mode: 1 to 20 MHz (V_{DD} = 2.7 to 3.6 V) HS (high-speed main) mode: 1 to 16 MHz (V_{DD} = 2.4 to 3.6 V) LS (low-speed main) mode: 1 to 8 MHz (V_{DD} = 1.8 to 3.6 V)		
	High-speed on-chip oscillator clock (f_{IH})	HS (high-speed main) mode: 1 to 32 MHz (V_{DD} = 2.7 to 3.6 V), HS (high-speed main) mode: 1 to 16 MHz (V_{DD} = 2.4 to 3.6 V), LS (low-speed main) mode: 1 to 8 MHz (V_{DD} = 1.8 to 3.6 V),		
Subsystem clock		XT1 (crystal) oscillator, external subsystem clock input (EXCLKS) 32.768 kHz (TYP.)		
Low-speed on-chip oscillator clock		15 kHz (TYP.)		
<R> RF base clock		48 MHz (TYP.)		
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Minimum instruction execution time		0.03125 μs (High-speed on-chip oscillator clock: f_{IH} = 32 MHz operation)		
		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)		
		30.5 μs (Subsystem clock: f_{SUB} = 32.768 kHz 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	41 Note 2		
	CMOS I/O	26		
	CMOS input	5		
	CMOS output	1 Note 2		
	N-ch open-drain I/O (6 V tolerance)	4		
	GPIO (RF unit)	5		
SubGHz RF transceiver		• IEEE802.15.4g standard specification SubGHz-band transceiver • RF frequency range: 863 to 928 MHz • Modulation method: 2FSK/GFSK, 4FSK/GFSK		
Timer	16-bit timer	9 channels		
	Watchdog timer	1 channel		
	Real-time clock (RTC)	1 channel		
	12-bit interval timer	1 channel		
	Timer output	1 channel		

(2/2)

Item		R5F11FLJ	R5F11FLK	R5F11FLL
Clock output/buzzer output		2		
		• 2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz (Main system clock: f_{MAIN} = 20 MHz operation) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz operation)		
10-bit resolution A/D converter		6 channels		
Serial interface		• CSI/UART: 2 channels • CSI: 2 channels (1 channel of 2 channels is used for the internal communication between MCU and RF transceiver.)		
	I^2C bus	2 channels		
Data transfer controller (DTC)		21 sources		
Vectored interrupt sources	Internal	26		
	External	7		
Reset		• Reset by $\overline{\text{RESET}}$ 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 (TYP.) • Power-down-reset: 1.50 (TYP.)		
Voltage detector		• Rising edge: 1.88 V to 3.13 V (10 stages) • Falling edge: 1.84 V to 3.06 V (10 stages)		
On-chip debug function		Provided		
Power supply voltage		V_{DD} = 1.8 to 3.6 V		
Operating ambient temperature		T_A = -40 to +85 °C (A: Consumer applications, D: Industrial applications)		
Package		64-pin HVQFN (9 × 9), (0.5 mm pitch)		

Note 1. This is about 47 KB when the self-programming function is used (For details, see **CHAPTER 4**).

Note 2. When using the RF transceiver, pins which a user uses for external connection between the MCU and RF transceiver on the board are included.

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 CONNECTION BETWEEN MCU AND RF TRANSCEIVER

2.1 Connection Pins of MCU and RF Transceiver

Table 2 - 1 lists the pins connected inside the RL78/G1H. Table 2 - 2 lists the pins which need to be connected on the board by the user.

These pins require initial settings for an appropriate mode/level before starting communication with the RF transceiver.

Table 2 - 1 Internal Pin Connection

Pin Name		Function	Direction
MCU	RF unit		
P10	OSCDRVSEL	This is a switch signal that has the buffer drive capability for 48 MHz crystal oscillator. The buffer size (current) of the oscillator becomes smaller when OSCDRVSEL is High. The buffer size (current) of the oscillator becomes larger when OSCDRVSEL is Low. P10, which is an internal I/O pin of MCU, controls OSCDRVSEL.	MCU to RF unit
P11	DON	This is an enable signal of DCDC converter of RF unit. It becomes active (DCDC converter is operable) when DON is set to High. P11, which is an internal I/O pin of MCU, controls DON.	MCU to RF unit
P12	RFRESETB	Hardware reset signal for the RF unit. RFRESETB=Low resets RF unit. RFRESETB is controlled by P12 which is an internal I/O port of the MCU.	MCU to RF unit
P13/SO20	SIN	Serial interface used for internal communication between blocks. Since it is dedicated to internal communication (MCU: output data, RF unit: input data), it cannot be used for communication with external modules.	MCU to RF unit
P14/SI20	SOUT**Note**	Serial interface used for internal communication between blocks. Since it is dedicated to internal communication (MCU: input data, RF unit: output data), it cannot be used for communication with external modules.	RF unit to MCU
P15/SCK20	SCLK	This is an operation clock of Serial interface for internal communication between blocks. Since it is dedicated to internal communication, it cannot be used for communication with external modules.	MCU to RF unit
P16	SEN	This is a communication enable control signal of Serial interface (RF unit) for internal communication between blocks. It becomes active (Serial communication enabled) when SEN is Low. P16, which is an internal I/O of MCU, controls SEN.	MCU to RF unit
P30/INTP3	INTOUT**Note**	Interrupt request signal from RF transceiver. If an interrupt source is generated in RF unit, the status is output from INTOUT. MCU receives this status at INTP3, and can execute the interrupt processing.	RF unit to MCU

Note These pin outputs might be high impedance causing state of RF transceiver. These pins are required to fix by MCU in this case. For details, see CHAPTER 18 RF TRANSCEIVER.

Table 2 - 2 Pins Externally Connected on User Board

Pin Name		Function	Direction
MCU	RF unit		
P130	STANDBY	This is a power-down control signal of RF unit. RF unit becomes operable when STANDBY is High. RF unit enters the SLEEP state when STANDBY is Low. P130, which is an I/O pin of MCU, controls STANDBY. Connect P130 and STANDBY externally (on the board).	MCU to RF unit

2.2 Communication Interface Between MCU and RF Transceiver

3-wire serial I/O (CSI) is used for the SPI interface for internal communication between the MCU and RF unit. For data transfer between the MCU and RF unit, a transfer clock is output from the MCU to the RF unit and data is transmitted and received. The operation of the 3-wire serial I/O (CSI) is shown in Table 2 - 3.

Table 2 - 3 3-Wire Serial IO Between MCU and RF Transceiver

Item	CSI20 (dedicated for internal communication)
Target channel	Channel 0 of serial array unit 1 (SAU1)
Pins used	SCK20, SI20, SO20 (pins for communication/all on-chip dedicated internal pin), and P16 (pin for control of the SEN pin/on-chip dedicated internal I/O pin)
Operation mode	Only transmission mode or transmission/reception mode is supported.
Master/slave	Only master is supported.
Interrupt	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 (OVF20)
Transfer data length	Only 8-bit length is supported.
Transfer rate	Within the range that satisfies the AC characteristics of the electrical specifications
Data phase	Only Type 1 is supported.
Clock phase	Only Type 1 is supported.
Data direction	Only MSB first is supported.

Caution Use the RL78/G1D so that these conditions and the specifications of the AC characteristics (CHAPTER 31 ELECTRICAL SPECIFICATIONS) are satisfied.

2.3 Initial Settings of Unused Internal Pins of MCU

After reset release, the following internal pins of the MCU need to be set to output mode (set the port registers and port mode registers to 0) by software.

P00, P01, P05, P06, P17, P23 to P27, P41 to P47, P50 to P57, P64 to P67, P73 to P74, P83 to P87
P101, P102, P110, P111, P145 to P147, P150 to P154

2.4 Base Operation Clock of RF Unit

RF unit operation requires a 48 MHz clock. Table 2 - 4 shows clock resonator connection and Figure 2 - 1 shows the clock configuration.

Table 2 - 4 Clock Resonator Connection (Using on-chip low-speed oscillator in the RF unit as a slow clock.)

Pin Name	Funcion
XIN	Base clock of the RF unit. Connect a 48 MHz crystal resonator.
XOUT	

Figure 2 - 1 Clock Configuration

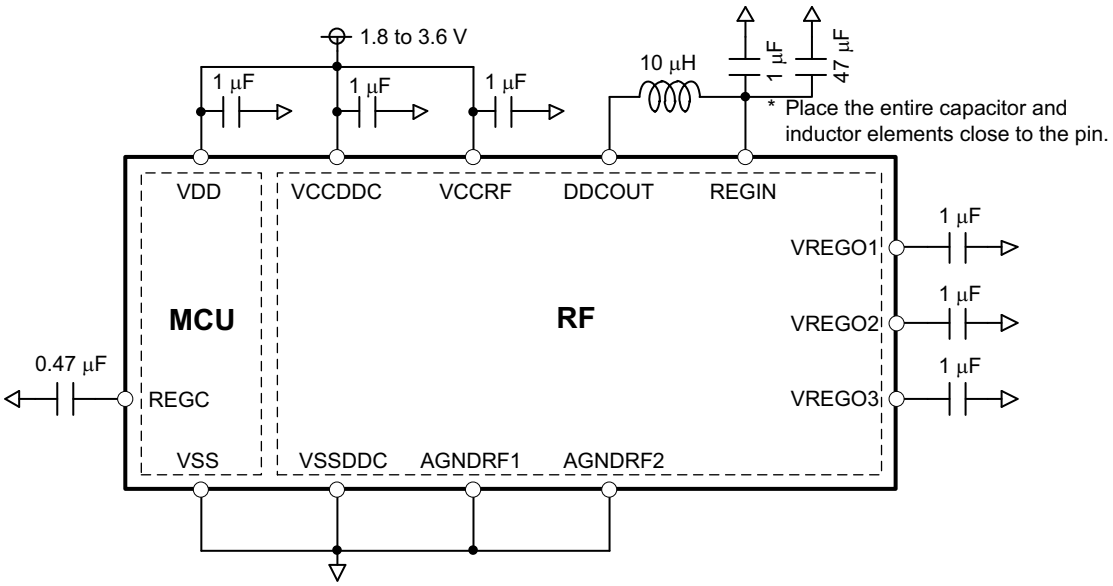
Caution This figure only shows connection between the clock resonator pins and clock line.

2.5 Power Configuration

Power is supplied to both the MCU and RF unit. The RF unit includes a DC-DC converter. The power switched at the DC-DC converter is output to the DDCOUT pin. The power is smoothed by an inductor and capacitor to step down the voltage, and then supplied to the REGIN.

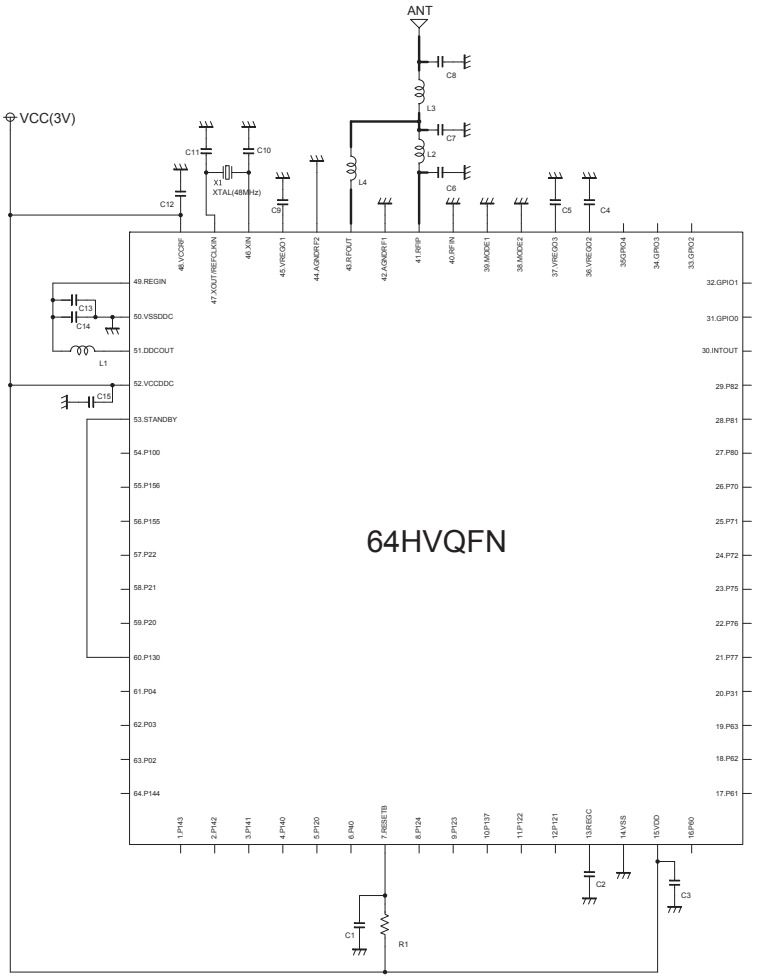
Figure 2 - 2 shows the power configuration of the RL78/G1H.

Figure 2 - 2 Power Configuration



2.6 Peripheral Circuits'Connection Diagram

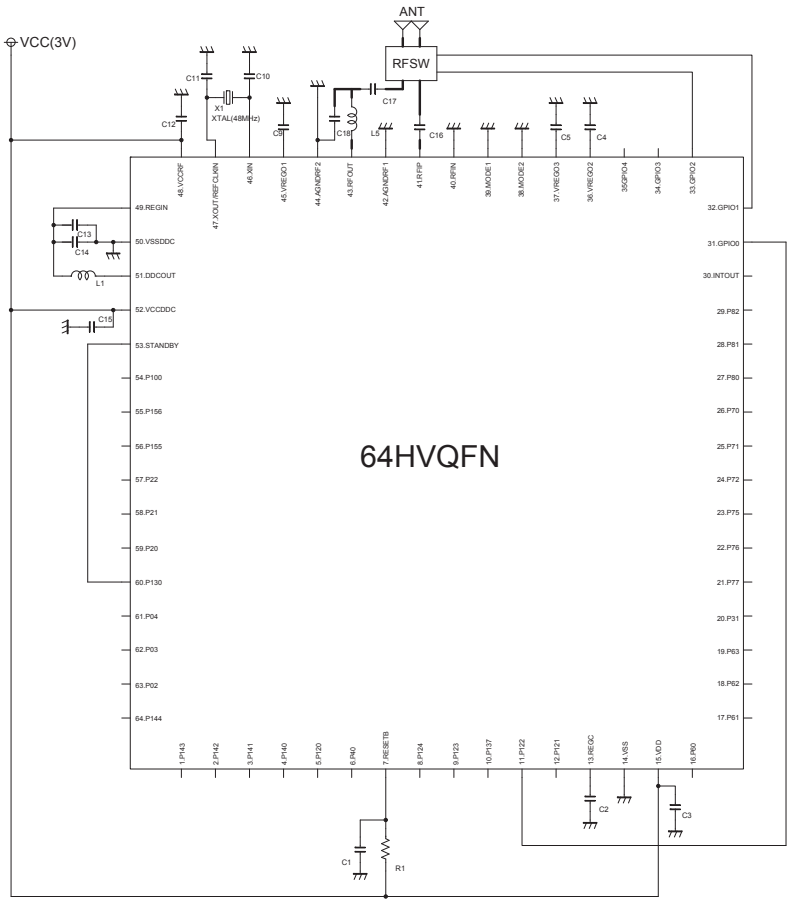
Figure 2 - 3 RL78/G1H (64-pin) Peripheral Circuits'Connection Diagram (without ANTSW)



Part number	Remarks
C6	Acceptable error: use ±0.25 pF accuracy part
C7	Acceptable error: use ±0.25 pF accuracy part
C8	Acceptable error: use ±0.25 pF accuracy part
L1(10 μH)	MLZ1608M100WT(10.0 μH ±20 %) is comfortable. Required inductor with good DC superposition characteristics. Small loss (DC resistance and AC resistance at operational frequency) is comfortable.
L2, L3, L4	Use chip inductance for high frequency. Required high Q at 1 GHz band, and self-resonance is higher than 1 GHz.

Caution The multiple power supply smoothing capacitors may be connected to a pin depending on the routing of the board wiring, noise, and others.

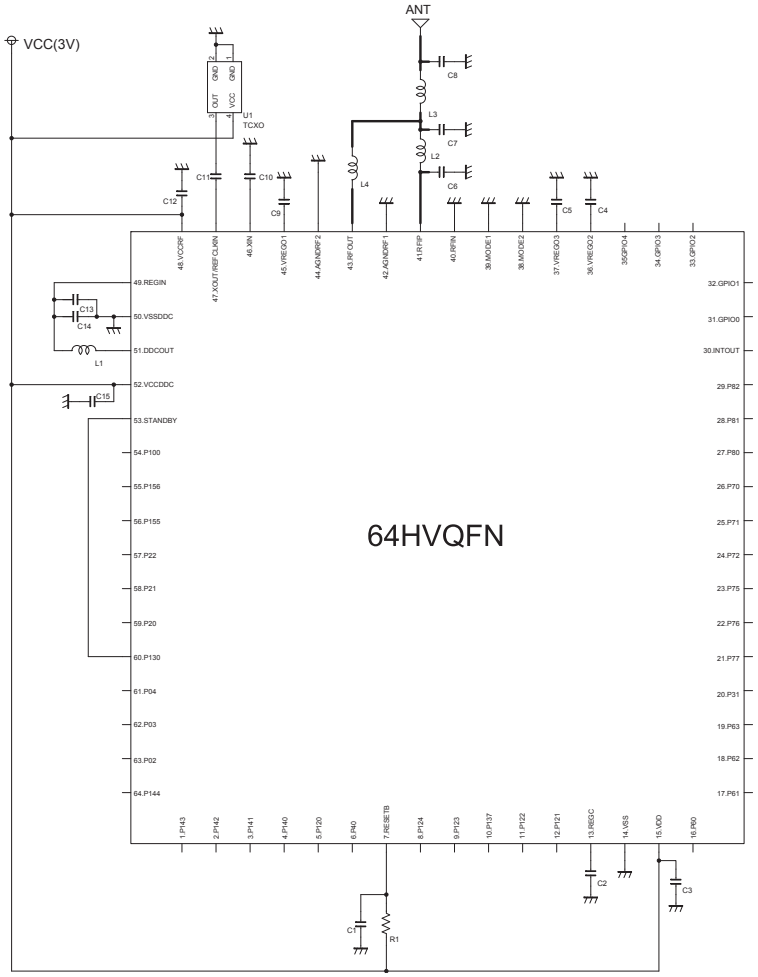
Figure 2 - 4 RL78/G1H (64-pin) Peripheral Circuits'Connection Diagram (with ANTSW)



Part number	Remarks
C18	Acceptable error: use ±0.25 pF accuracy part
L1(10 μH)	MLZ1608M100WT(10.0 μH ±20 %) is comfortable. Required inductor with good DC superposition characteristics. Small loss (DC resistance and AC resistance at operational frequency) is comfortable.
L5	Use chip inductance for high frequency. Required high Q at 1 GHz band, and self-resonance is higher than 1 GHz.

Caution The multiple power supply smoothing capacitors may be connected to a pin depending on the routing of the board wiring, noise, and others.

Figure 2 - 5 RL78/G1H (64-pin) Peripheral Circuit Connection Diagram (Using TCXO)



Part number	Remarks
C10	100 pF
C11	100 pF

Caution The multiple power supply smoothing capacitors may be connected to a pin depending on the routing of the board wiring, noise, and others.

CHAPTER 3 PIN FUNCTIONS

3.1 Port Functions

Pin I/O buffer power supplies depend on the pin. The relationship between these power supplies and the pins is shown below.

Table 3 - 1 Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
V_{DD}	• P02 to P04, P10 to P16, P20 to P22, P30, P31, P40, P60 to P63, P70 to P72, P75 to P77, P80 to P82, P120 to P124, P130, P137, P140 to P144, P155, P156 • $\overline{\text{RESET}}$, REGC
V_{CCDDC}	• GPIO0 to GPIO4, STANDBY, MODE1, MODE2, INTOUT

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

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P02	7-3-4 Note 1	I/O	Prohibit I/O Note 2	SO10/TxD1	Port 0. 3-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 P03 and P04 can be set to TTL input buffer. Output of P02 to P04 can be set to N-ch open-drain output (V_{DD} tolerance). The digital I/O of P02 and P03 is prohibited at the reset release Note 2.
P03	8-3-4 Note 1			SI10/RxD1	
P04	8-1-4		Input port	SCK10	
P10 Note 3	—	I/O	Input port	—	Port 1. 7-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.
P11 Note 3				—	
P12 Note 3				—	
P13 Note 3				SO20	
P14 Note 3				SI20	
P15 Note 3				SCK20	
P16 Note 3				—	
P20	4-3-3	I/O	Analog function	ANI0/AV_{REFP}	Port 2. 3-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input Note 4.
P21				ANI1/AV_{REFM}	
P22				ANI2	
P30 Note 3	8-1-4	I/O	Input port	INTP3	Port 3. 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.
P31	7-1-3			TI03/TO03/INTP4	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-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.
P60	12-1-2	I/O	Input port	SCLA0	Port 6. 4-bit I/O port. Input/output can be specified in 1-bit units. Output of P60 to P63 is N-ch open-drain output (6 V tolerance).
P61				SDAA0	
P62				SCLA1	
P63				SDAA1	
P70	7-1-3	I/O	Input port	SCK21	Port 7. 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. Output of P71 can be set to N-ch open-drain output (V_{DD} tolerance).
P71	7-1-4			SI21	
P72	7-1-3			SO21	
P75				NTP9	
P76				INTP10	
P77				INTP11	

Note 1. Input of A/D converter is not supported.

Note 2. Each pin can be specified as digital I/O port by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Note 3. This pin is used for connection between the MCU and RF transceiver. For details, refer to **CHAPTER 2 CONNECTION BETWEEN MCU AND RF TRANSCEIVER**.

Note 4. Each pin can be specified as either digital or analog by setting the A/D port configuration register (ADPC).

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P80	8-1-4	I/O	Input port	—	Port 8. 3-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.
P81				—	
P82	7-1-4			—	
P100	7-3-3	I/O	Prohibit I/O	—	Port 10. 1-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. The digital I/O of P100 is prohibited at reset release Note 1.
P120	7-3-3	I/O	Analog function	ANI19	Port 12. 1-bit I/O port and 4-bit input-only port. P120 can be set to analog I/O. For only P120, input/output can be specified. For only P120, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P121	2-2-1	Input	Input port	X1	
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P130Note 2	1-1-1	Output	Output port	—	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-2	Input	Input port	INTP0	
P140	7-1-3	I/O	Input port	PCLBUZ0/INTP6	Port 14. 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 P142 and P143 can be set to TTL input buffer. Output of P142 to P144 can be set to N-ch open-drain output (V_{DD} tolerance).
P141				PCLBUZ1/INTP7	
P142	8-1-4			SCK30	
P143				SI30/RxD3	
P144	7-1-4			SO30/TxD3	
P155	4-3-3	I/O	Analog function	ANI13	Port 15. 2-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input Note 3.
P156				ANI14	
$\overline{\text{RESET}}$	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.

Note 1. Each pin can be specified as digital I/O port by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

Note 2. This pin is used for connection between the MCU and RF transceiver. For details, refer to **CHAPTER 2 CONNECTION BETWEEN MCU AND RF TRANSCEIVER**.

Note 3. Each pin can be specified as either digital or analog by setting the A/D port configuration register (ADPC).

3.2 Functions other than port pins

Function Name	I/O	Function
ANI0 to ANI2, ANI13, ANI14, ANI19	Input	A/D converter analog input (see **Figure 13 - 30 Analog Input Pin Connection**)
INTP0, INTP3 Note, INTP4, INTP6, INTP7, INTP9 to INTP11	Input	External interrupt request input Valid edge specification: Rising edge, falling edge, or both rising and falling edges
PCLBUZ0, PCLBUZ1	Output	Clock output/buzzer output
REGC	—	Pin for connecting regulator output stabilization capacitance for internal operation. Connect this 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.
$\overline{\text{RESET}}$	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 VDD.
RxD1, RxD3	Input	Serial data input pins of serial interface UART1 and UART3
TxD1, TxD3	Output	Serial data output pins of serial interface UART1 and UART3
SCK10, SCK20 **Note**, SCK21, SCK30	I/O	Serial clock I/O pins of serial interface CSI10, CSI20, CSI21, and CSI30
SI10, SI20 Note, SI21, SI30	Input	Serial data input pins of serial interface CSI10, CSI20, CSI21, and CSI30
SO10, SO20 Note, SO21, SO30	Output	Serial data output pins of serial interface CSI10, CSI20, CSI21, and CSI30
SCLA0, SCLA1	I/O	Serial clock I/O pins of serial interface IICA0 and IICA1
SDAA0, SDAA1	I/O	Serial data I/O pins of serial interface IICA0 and IICA1
TI03	Input	The pins for inputting an external count clock/capture trigger to 16-bit timer 03
TO03	Output	Timer output pins of 16-bit timer 03
X1, X2	—	Resonator connection for main system clock
EXCLK	Input	External clock input for main system clock
XT1, XT2	—	Resonator connection for subsystem clock
EXCLKS	Input	External clock input for subsystem clock
VDD	—	Positive power supply for all pins
AVREFP	Input	A/D converter reference potential (+ side) input
AVREFM	Input	A/D converter reference potential (- side) input
Vss	—	Ground potential for all pins
TOOL0	I/O	Data I/O for flash memory programmer/debugger

Note This pin is used for connection between the MCU and RF transceiver. For details, refer to **CHAPTER 2 CONNECTION BETWEEN MCU AND RF TRANSCEIVER**.

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

Table 3 - 2 Relationships Between P40/TOOL0 and Operation Mode After Reset Release

P40/TOOL0	Operating mode
VDD	Normal operation mode
0 V	Flash memory programming mode

For details, see 27.3 Programming Method.

Remark A bypass capacitor about 0.1 μF must be connected for measures of noises and latch-up between VDD and Vss lines on the shortest distance and with comparative thick wire.

3.3 Connection of Unused Pins

Table 3 - 3 shows the Connection of Unused Pins.

<R>

Table 3 - 3 Connection of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P02 to P04	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P20 to P22		
P31		
P40/TOOL0		Input: Independently connect to V_{DD} via a resistor, or leave open. Output: Leave open.
P60 to P63		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 P72, P75 to P77		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P80 to P82		
P100		
P120		
P121 to P124	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P130	Output	Leave open.
P137	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P140 to P144	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P155, P156		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
$\overline{\text{RESET}}$	Input	Connect to V_{DD} directly or via a resistor.
REGC	—	Connect to V_{SS} via a capacitor (0.47 to 1 μF).
GPIO0/CLKOUT	I/O	Input: Leave open or independently connect to VCCDDC or VSSDDC via a resistor. Output: Leave open.
GPIO1/ANTSELOUT0		
GPIO2/ANTSELOUT1		
GPIO3		
GPIO4/ANTSW		

Remark For how to handle the ports other than the above, see **CHAPTER 5 PORT FUNCTIONS**.

3.4 Pin Block Diagrams

For the pin types listed in 3.1 Port Functions, pin block diagrams are shown in Figures 3 - 1 to 3 - 13.

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

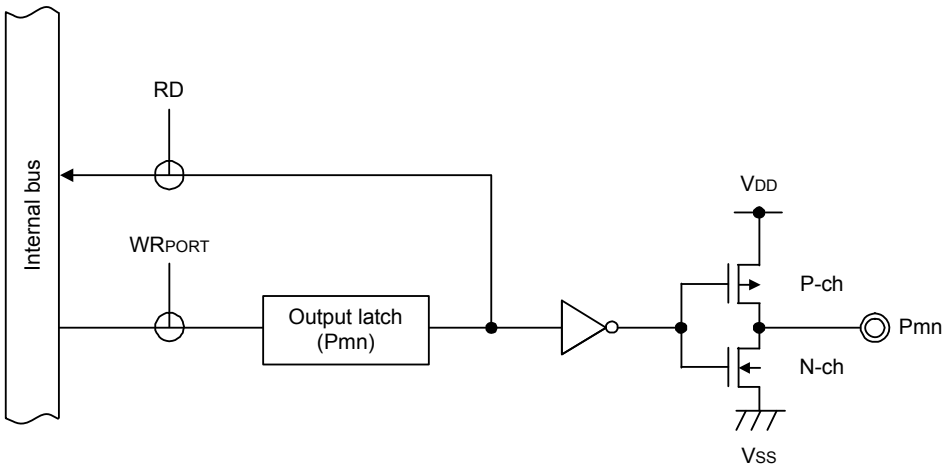


Figure 3 - 2 Pin Block Diagram of Pin Type 2-1-1

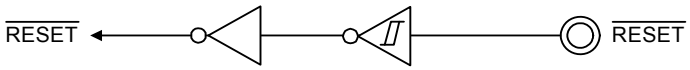
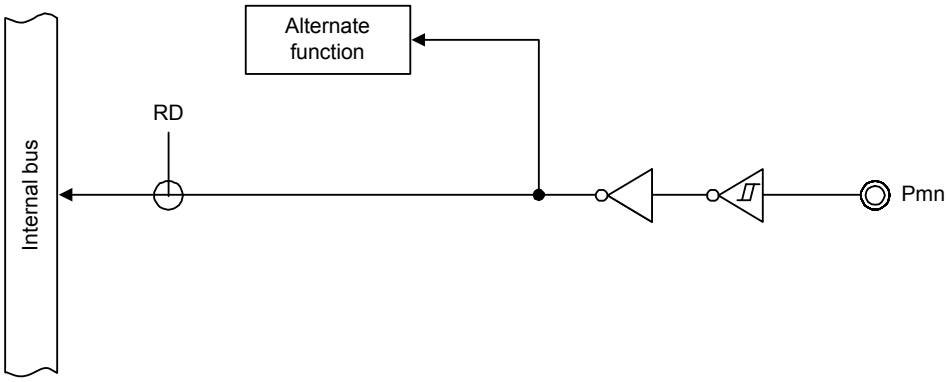
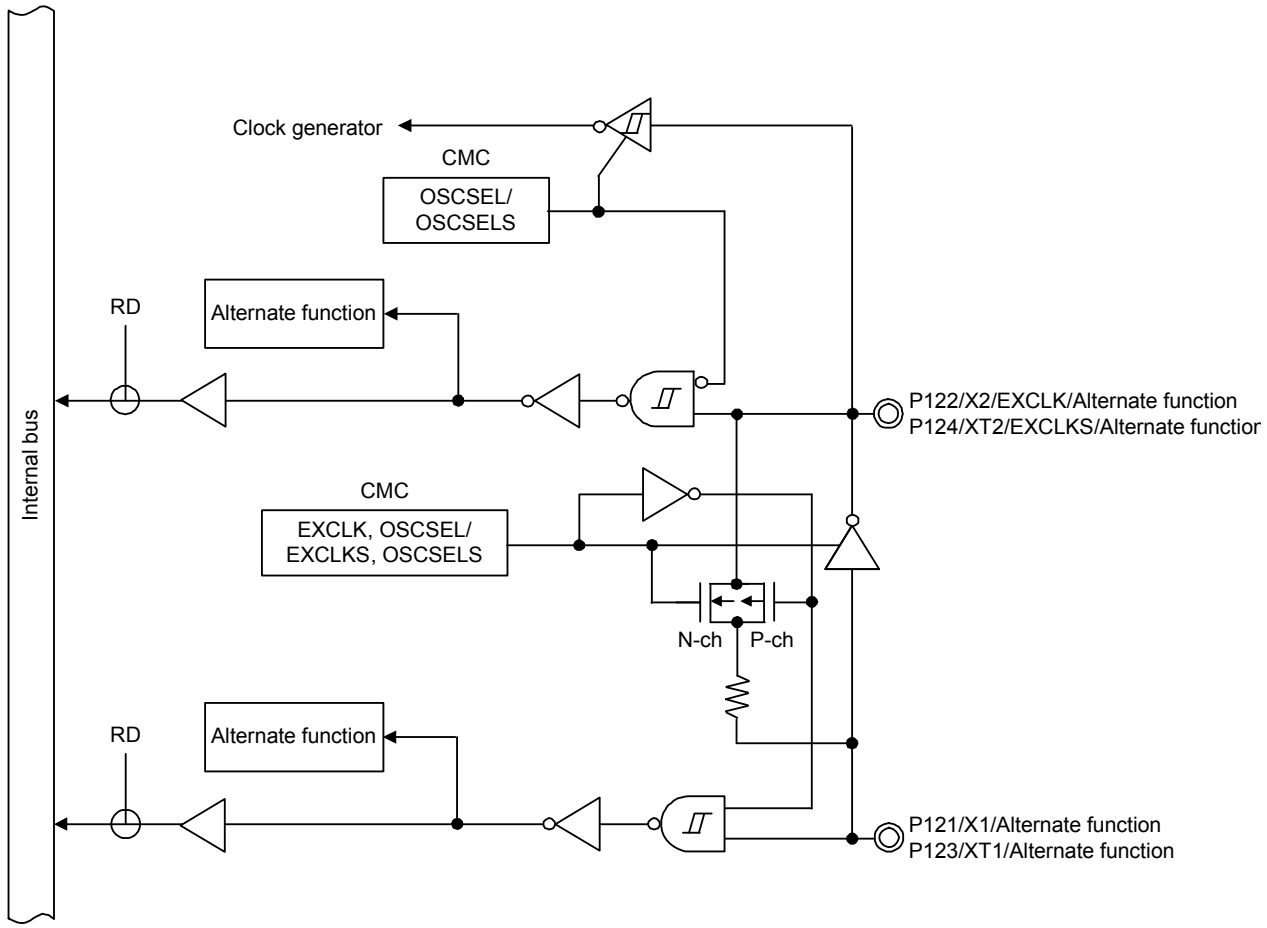


Figure 3 - 3 Pin Block Diagram of Pin Type 2-1-2



Remark Refer to **3.1 Port Functions** for alternate functions.

Figure 3 - 4 Pin Block Diagram of Pin Type 2-2-1



Remark Refer to **3.1 Port Functions** for alternate functions.

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

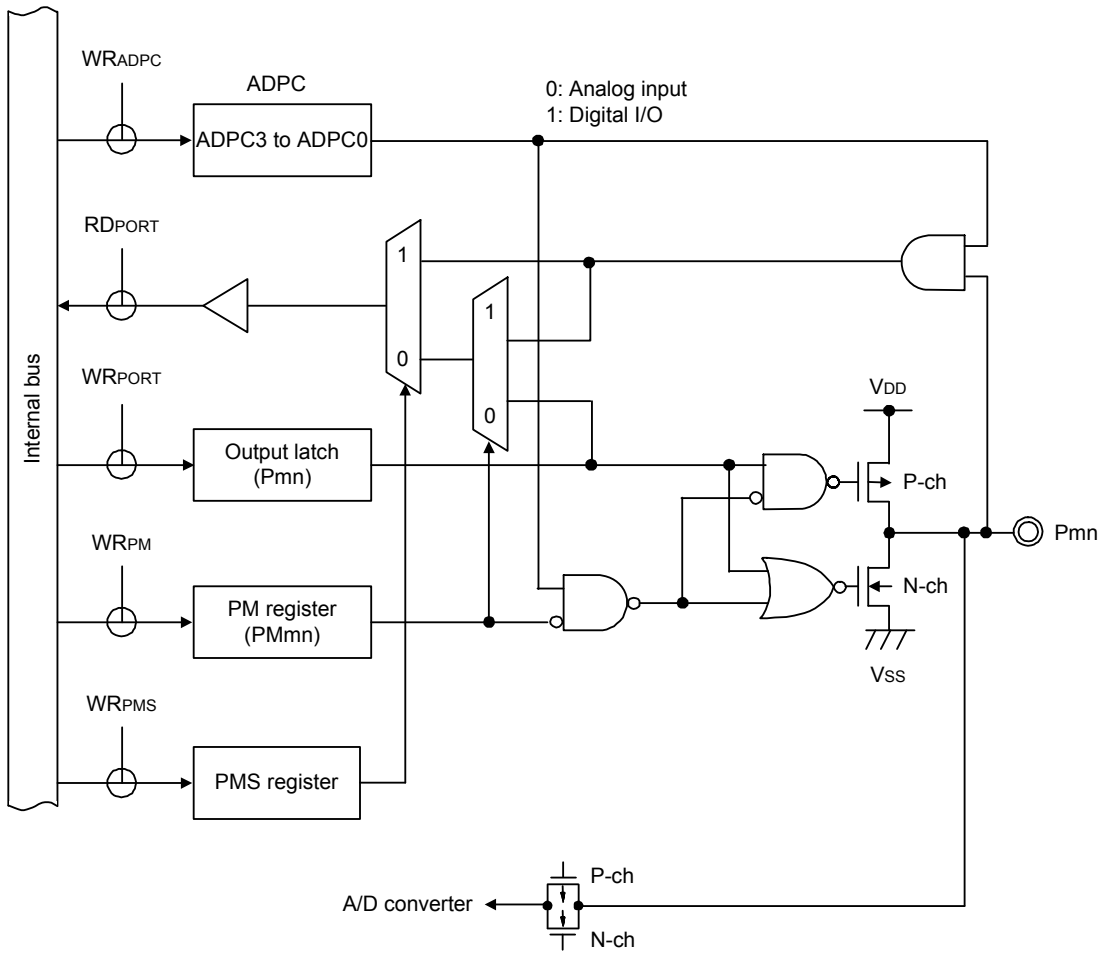
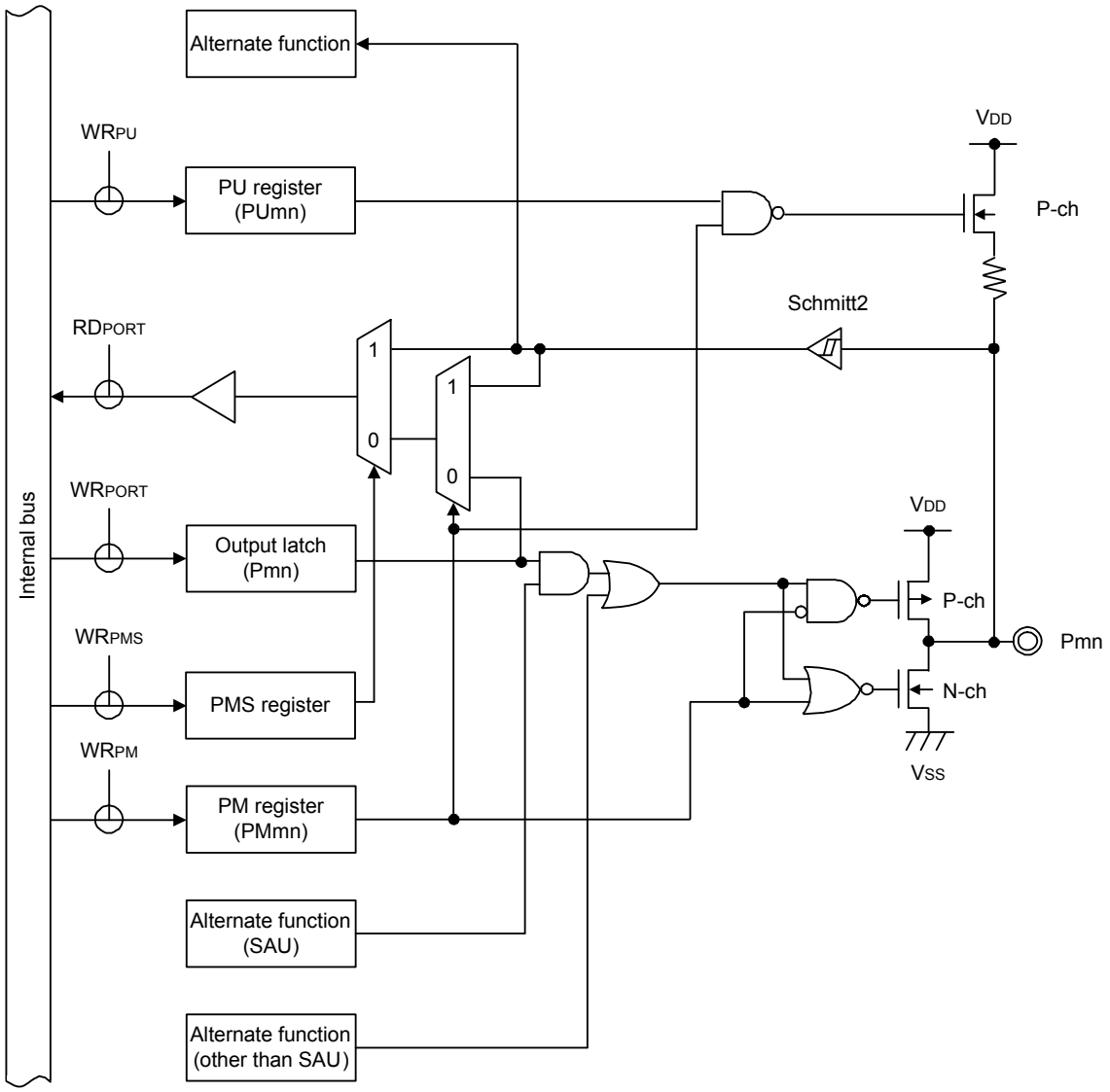
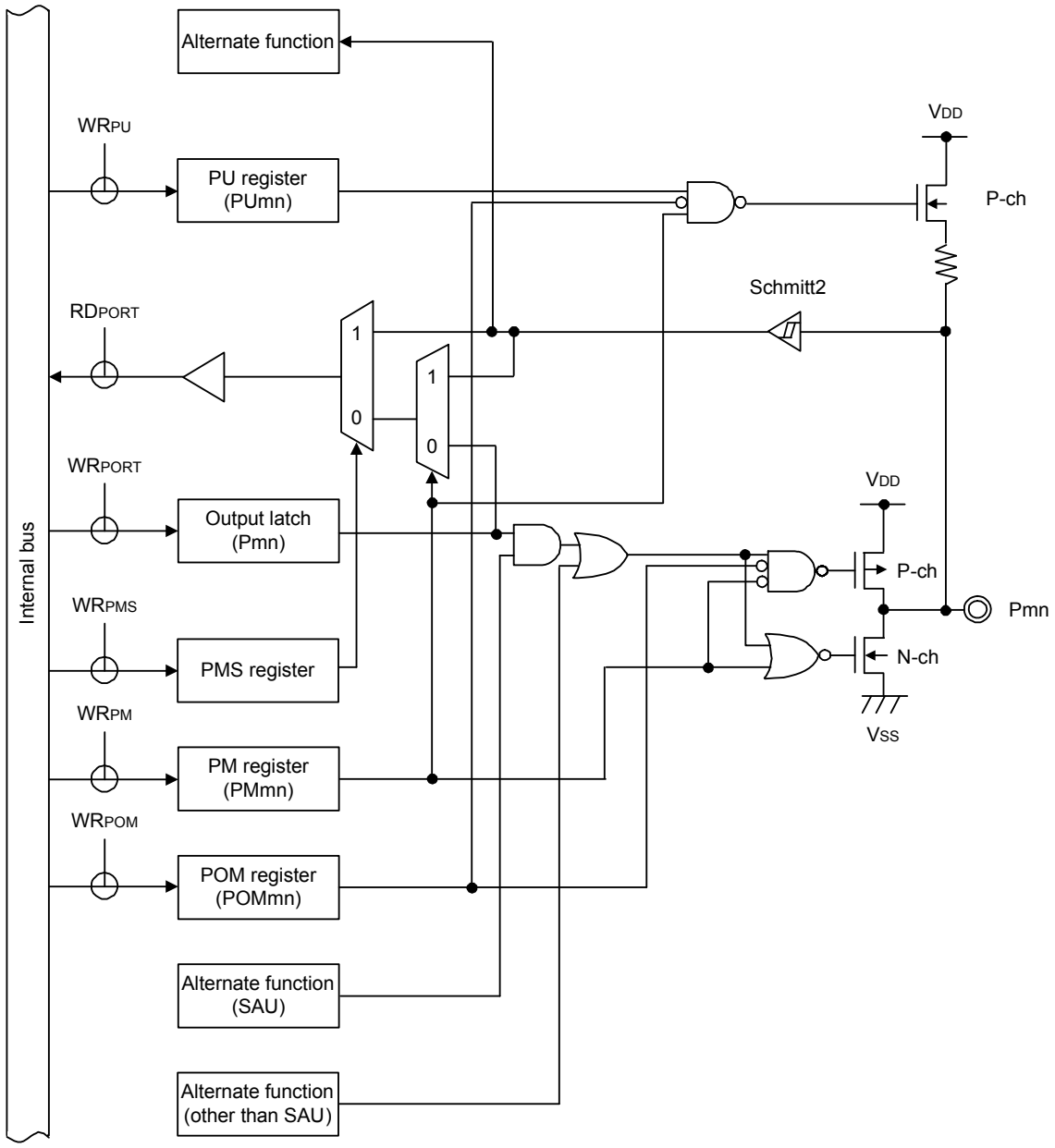


Figure 3 - 6 Pin Block Diagram of Pin Type 7-1-3



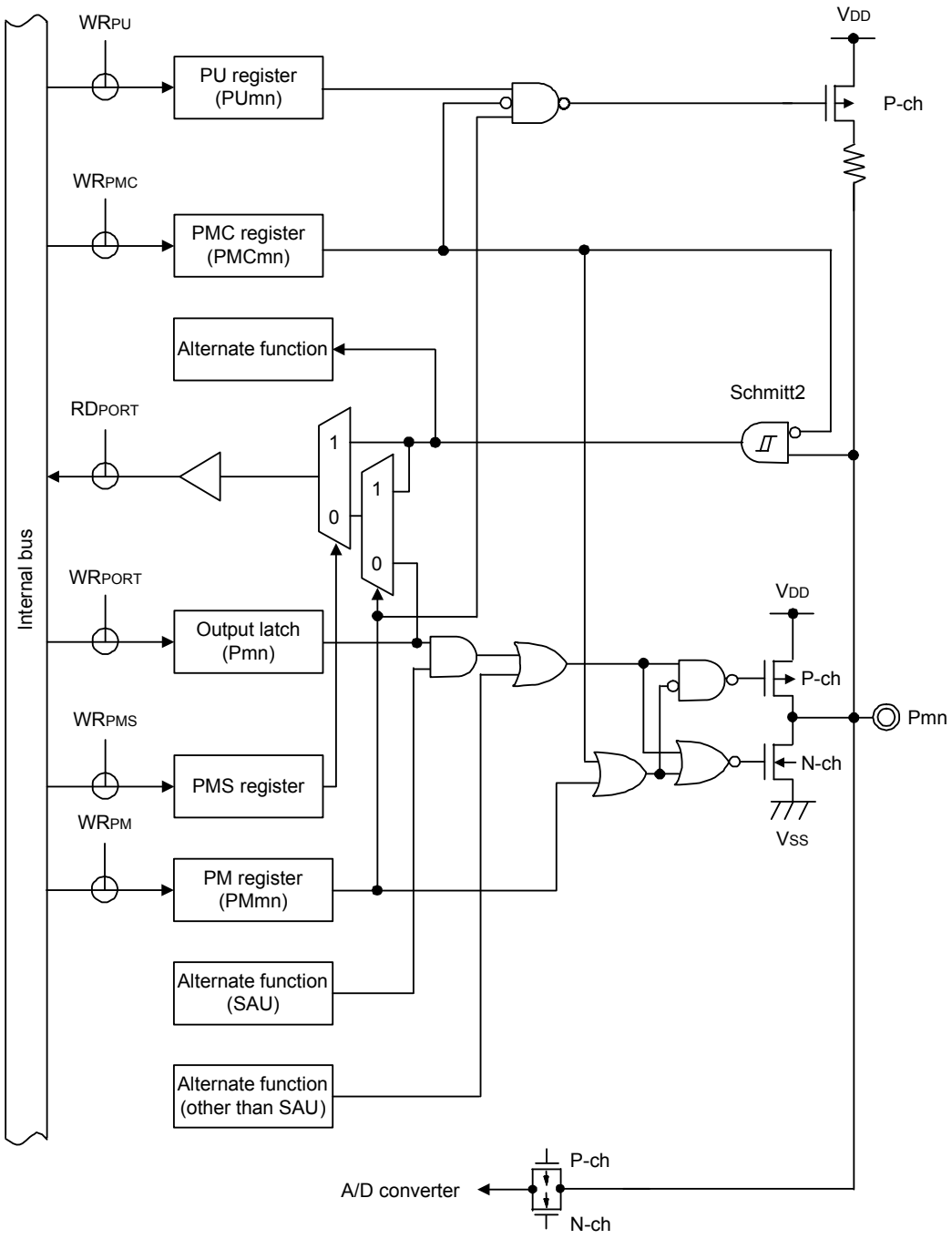
Remark 1. Refer to **3.1 Port Functions** for alternate functions.
Remark 2. SAU: Serial array unit

Figure 3 - 7 Pin Block Diagram of Pin Type 7-1-4



Remark 1. Refer to **3.1 Port Functions** for alternate functions.
Remark 2. SAU: Serial array unit

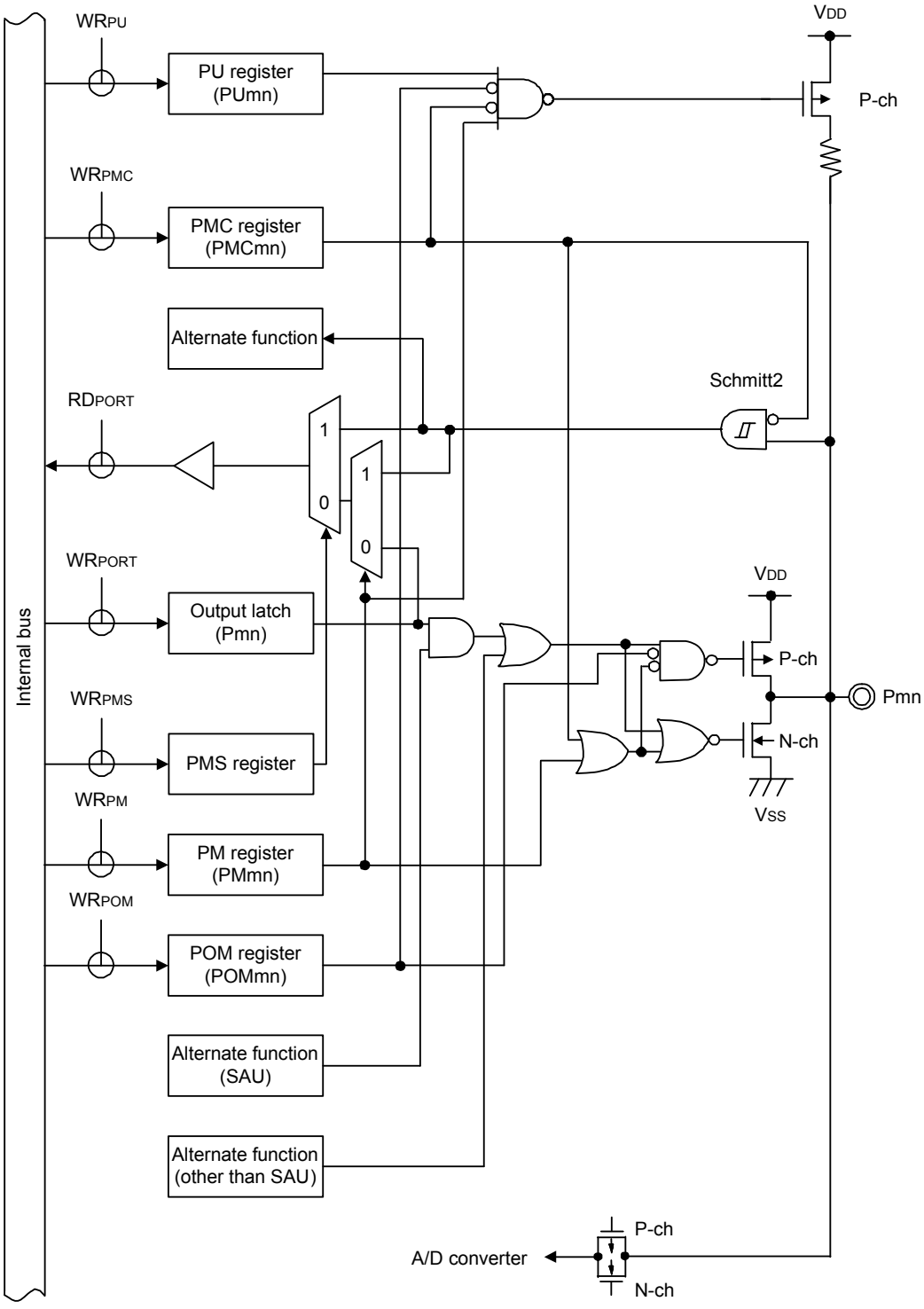
Figure 3 - 8 Pin Block Diagram of Pin Type 7-3-3



Remark 1. Refer to **3.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

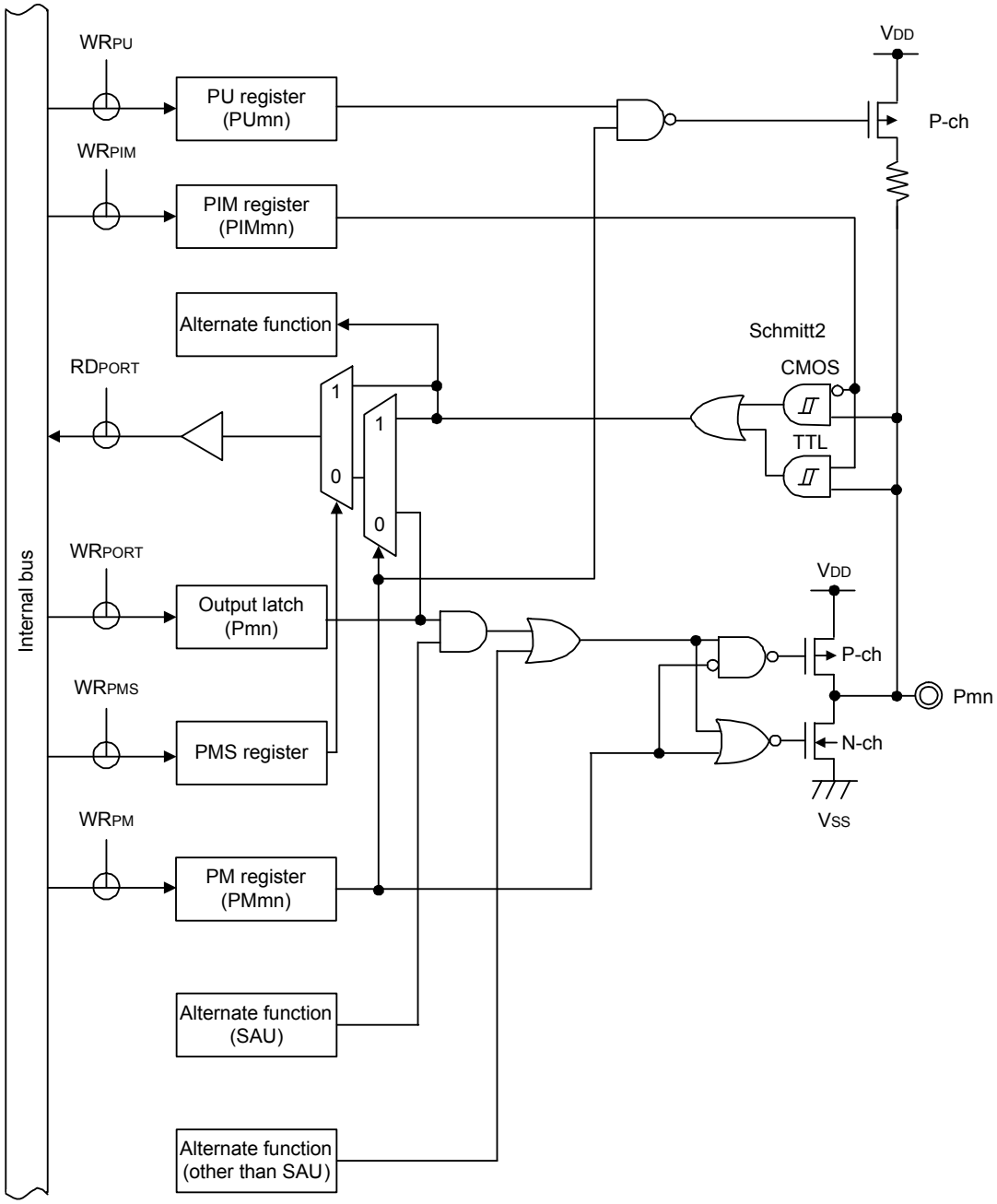
Figure 3 - 9 Pin Block Diagram of Pin Type 7-3-4



Remark 1. Refer to **3.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

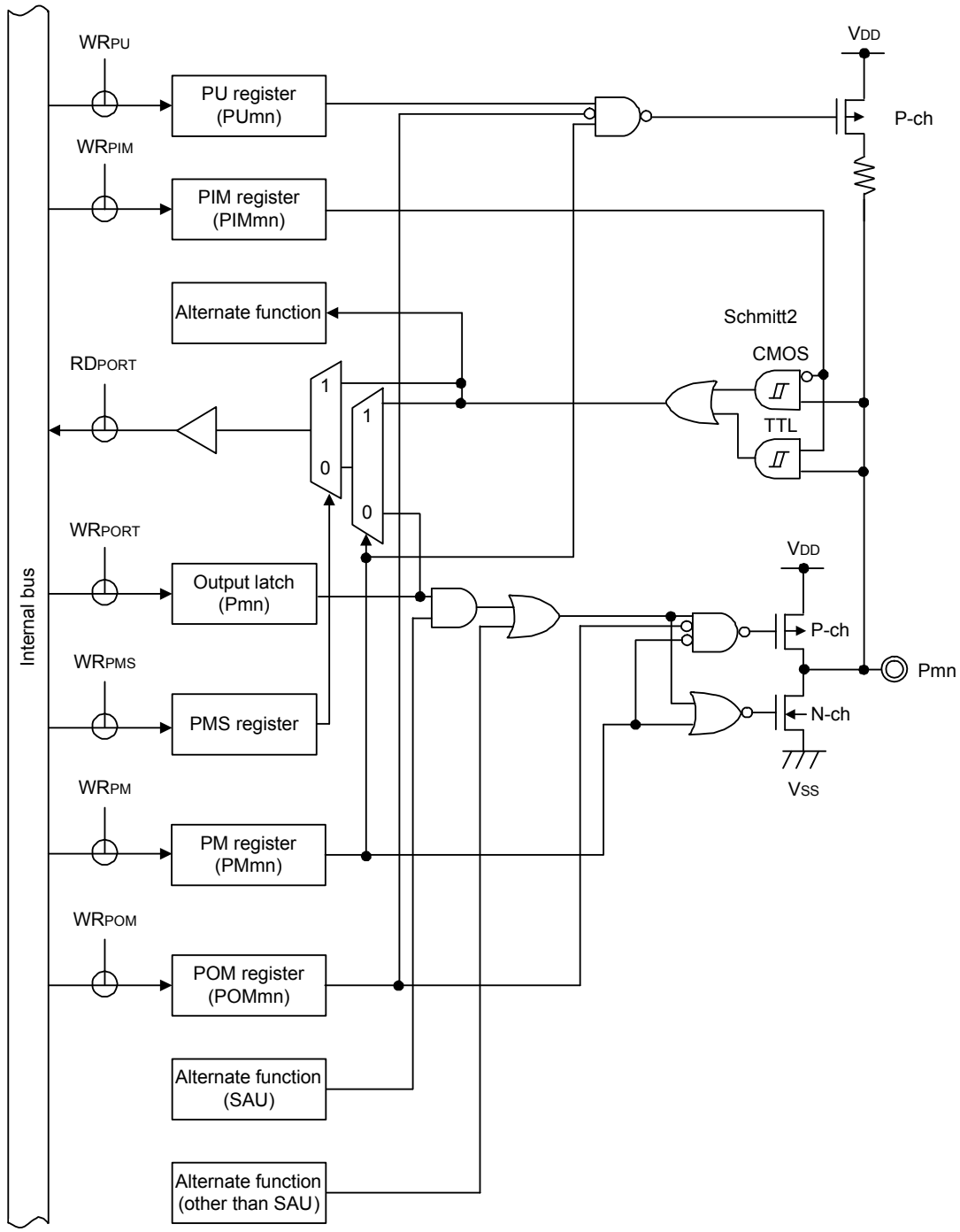
Figure 3 - 10 Pin Block Diagram of Pin Type 8-1-3



Remark 1. Refer to **3.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

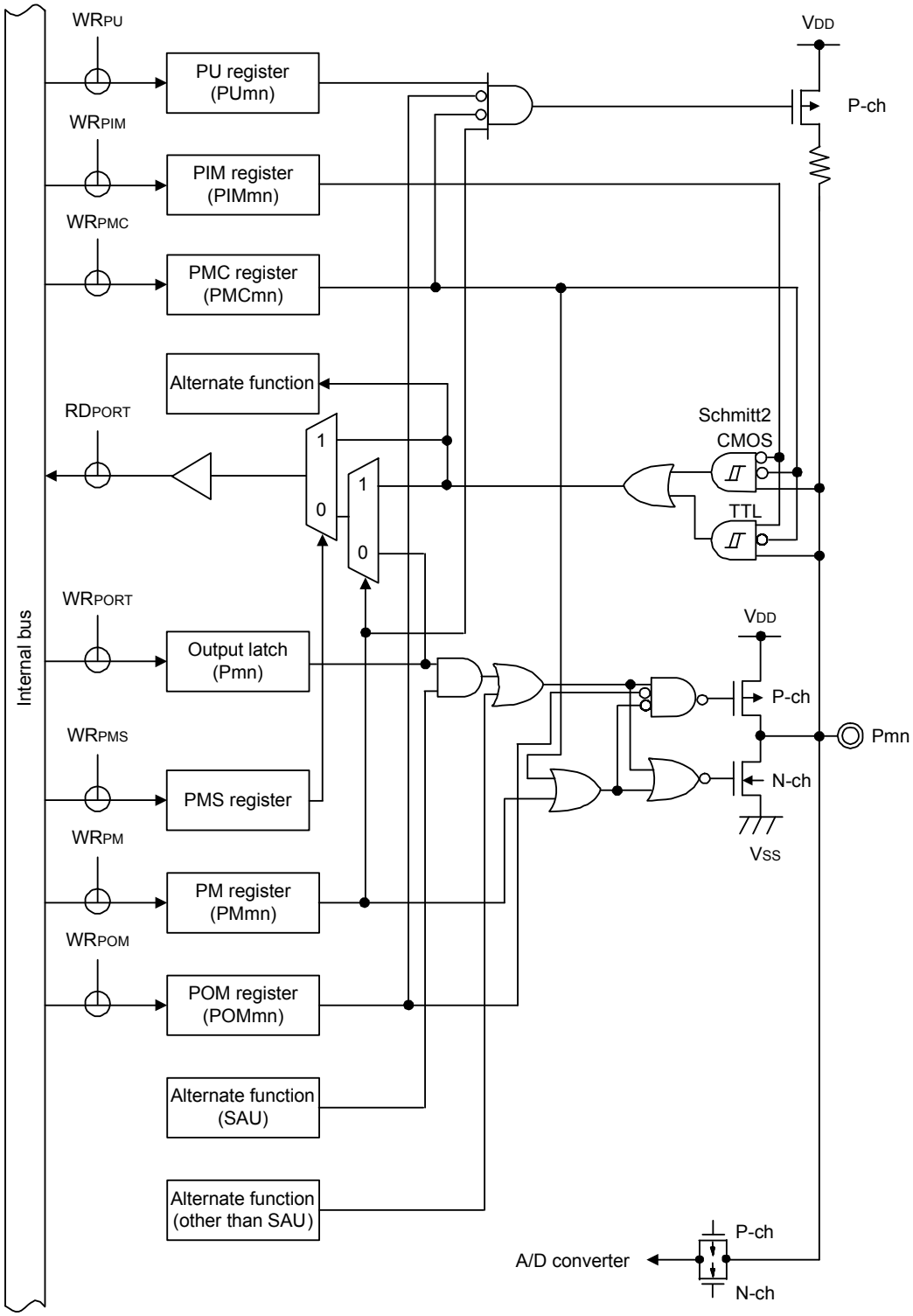
Figure 3 - 11 Pin Block Diagram of Pin Type 8-1-4



Remark 1. Refer to **3.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

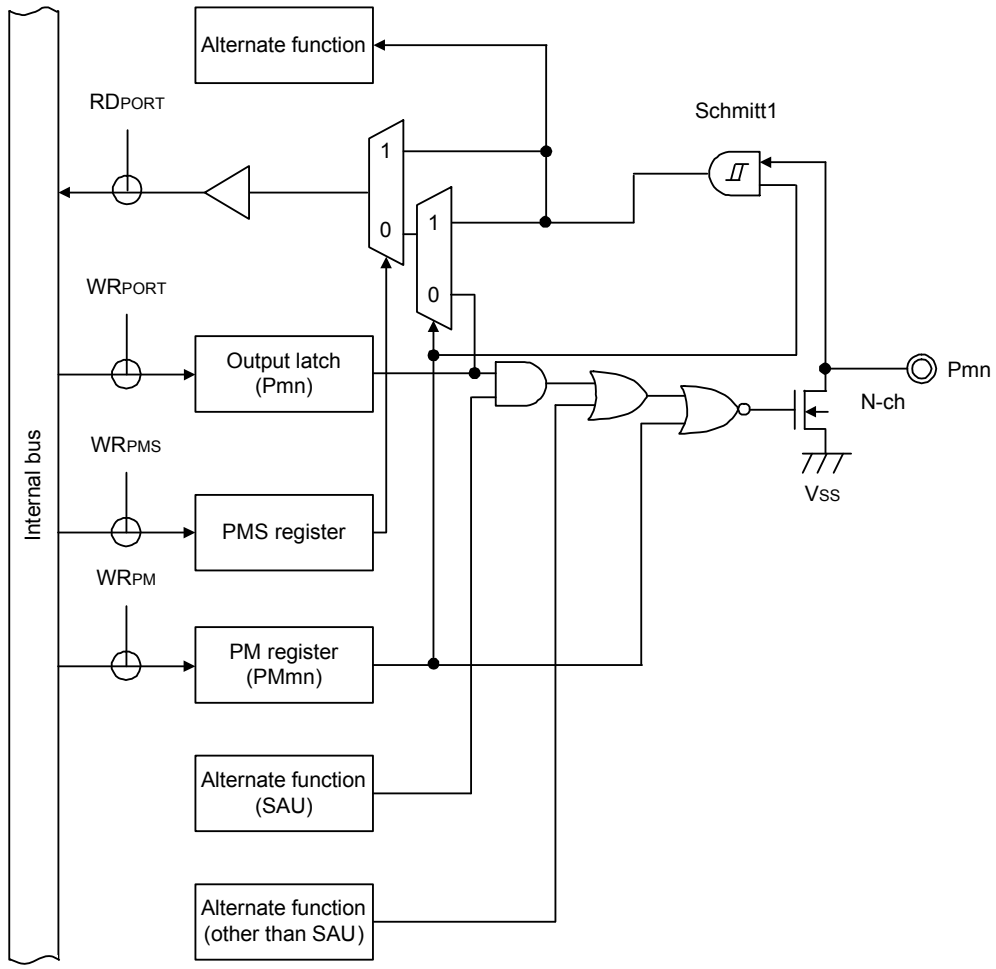
Figure 3 - 12 Pin Block Diagram of Pin Type 8-3-4



Remark 1. Refer to **3.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

Figure 3 - 13 Pin Block Diagram of Pin Type 12-1-2



Remark 1. Refer to **3.1 Port Functions** for alternate functions.
Remark 2. SAU: Serial array unit

Figure 3 - 14 Pin Block Diagram of STANDBY, MODE1, MODE2

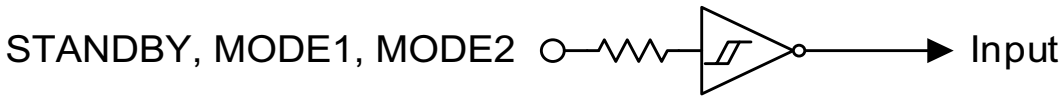


Figure 3 - 15 Pin Block Diagram of Pin GPIO0 to GPIO4

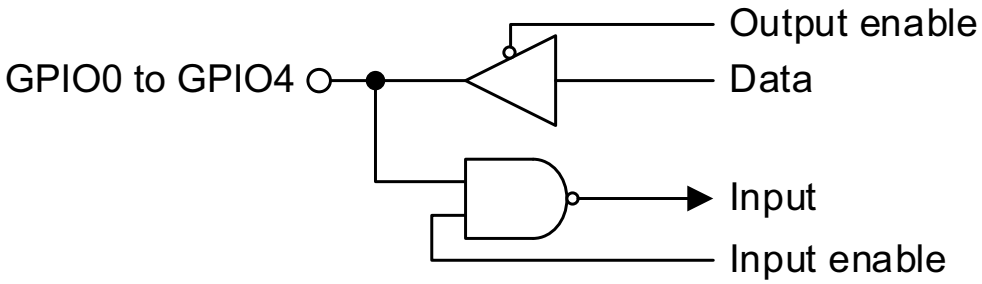
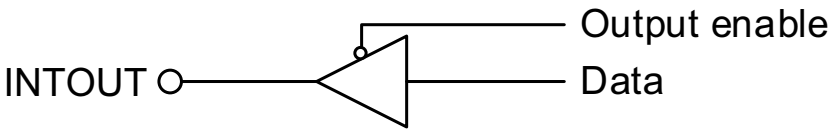


Figure 3 - 16 Pin Block Diagram of Pin INTOUT

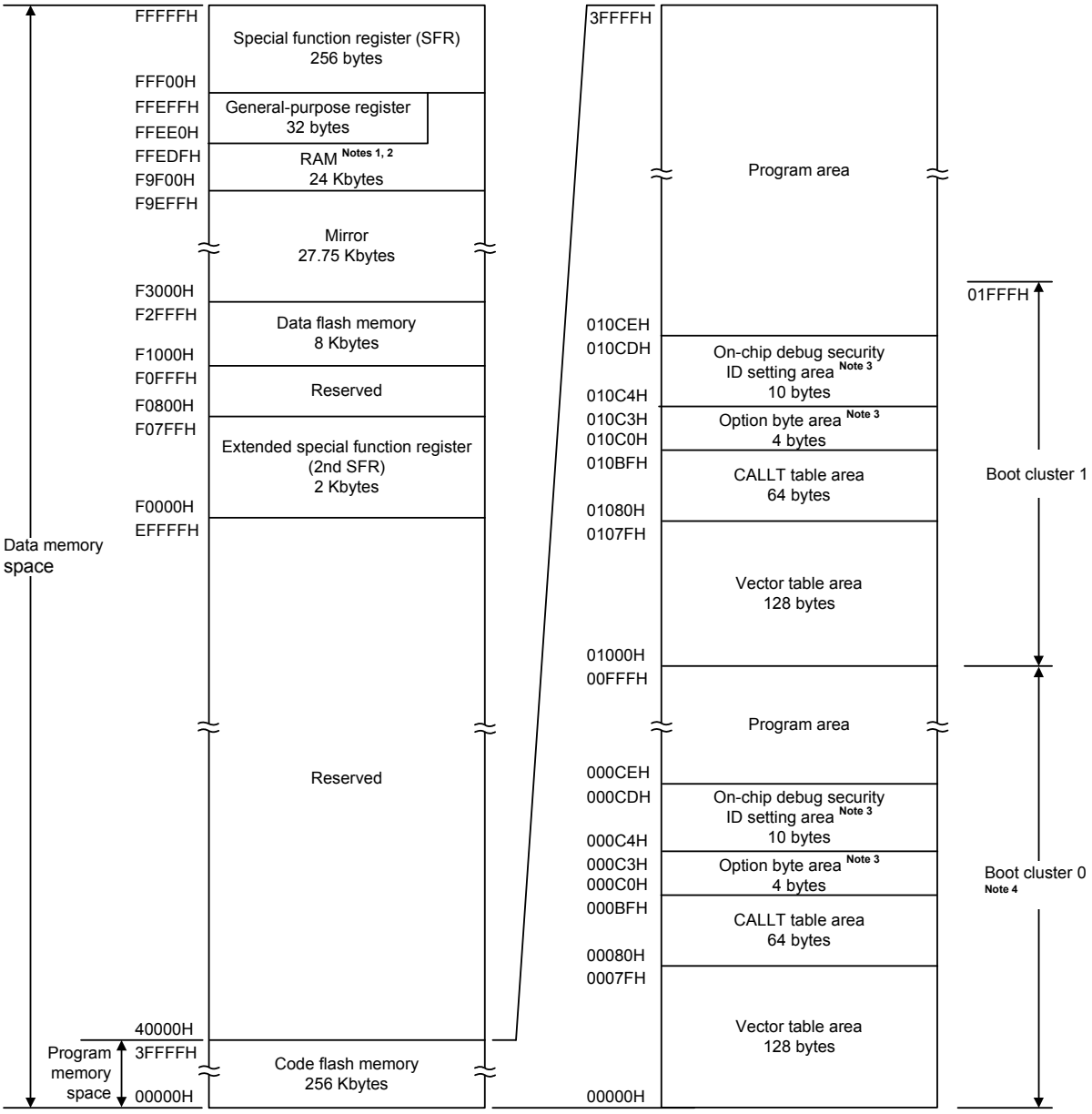


CHAPTER 4 CPU ARCHITECTURE

4.1 Memory Space

Products in the RL78/G1H can access a 1 MB address space. Figures 4 - 1 to 4 - 3 show the memory maps.

Figure 4 - 1 Memory Map (R5F11FLJ)



Note 1. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

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

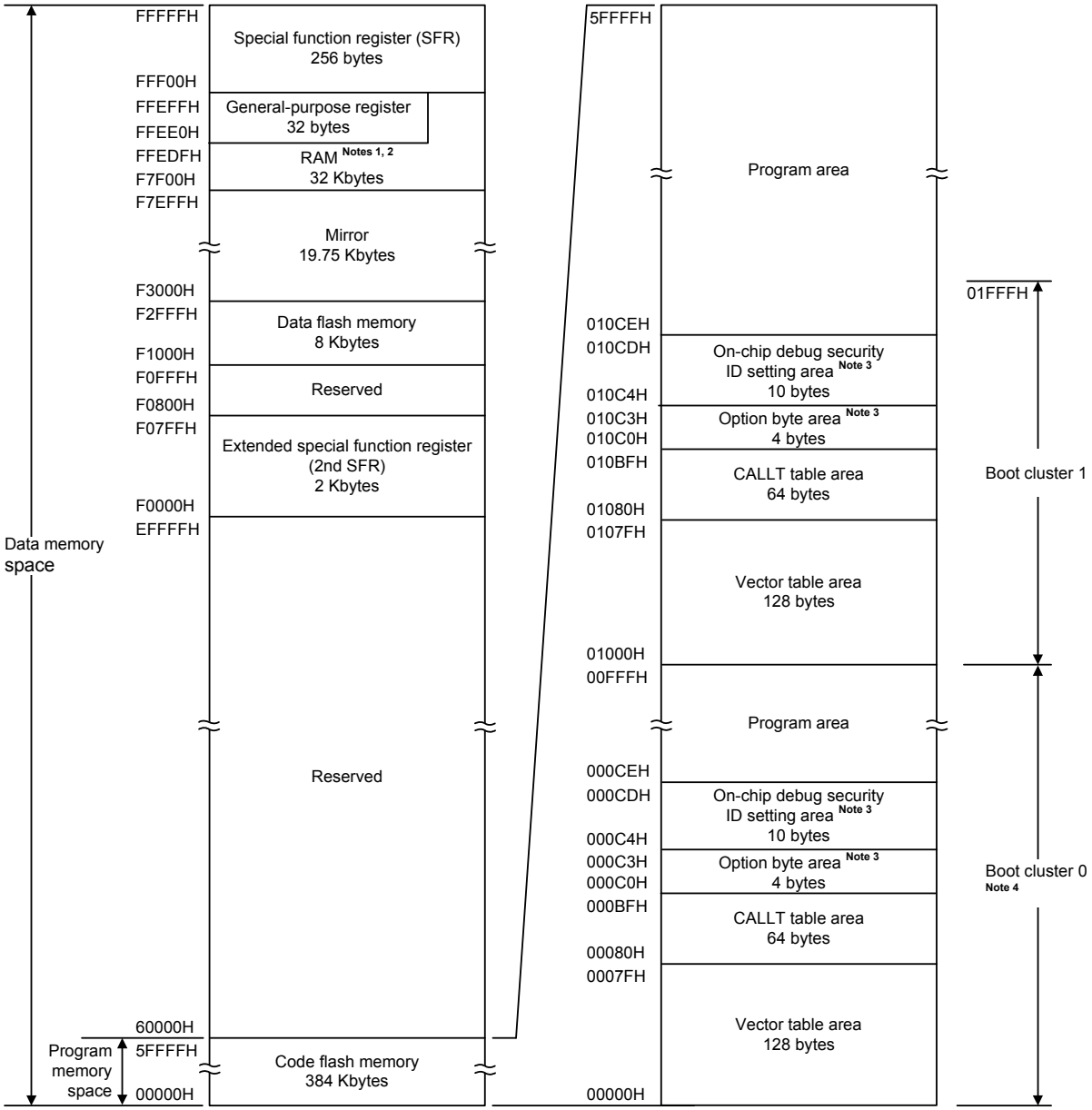
Note 3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

Note 4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **27.6 Security Settings**).

Note 5. Use of the area FA300H to FA6FFH is prohibited when using the on-chip debugging trace function.

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 24.3.3 RAM parity error detection function.

Figure 4 - 2 Memory Map (R5F11FLK)



Note 1. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

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

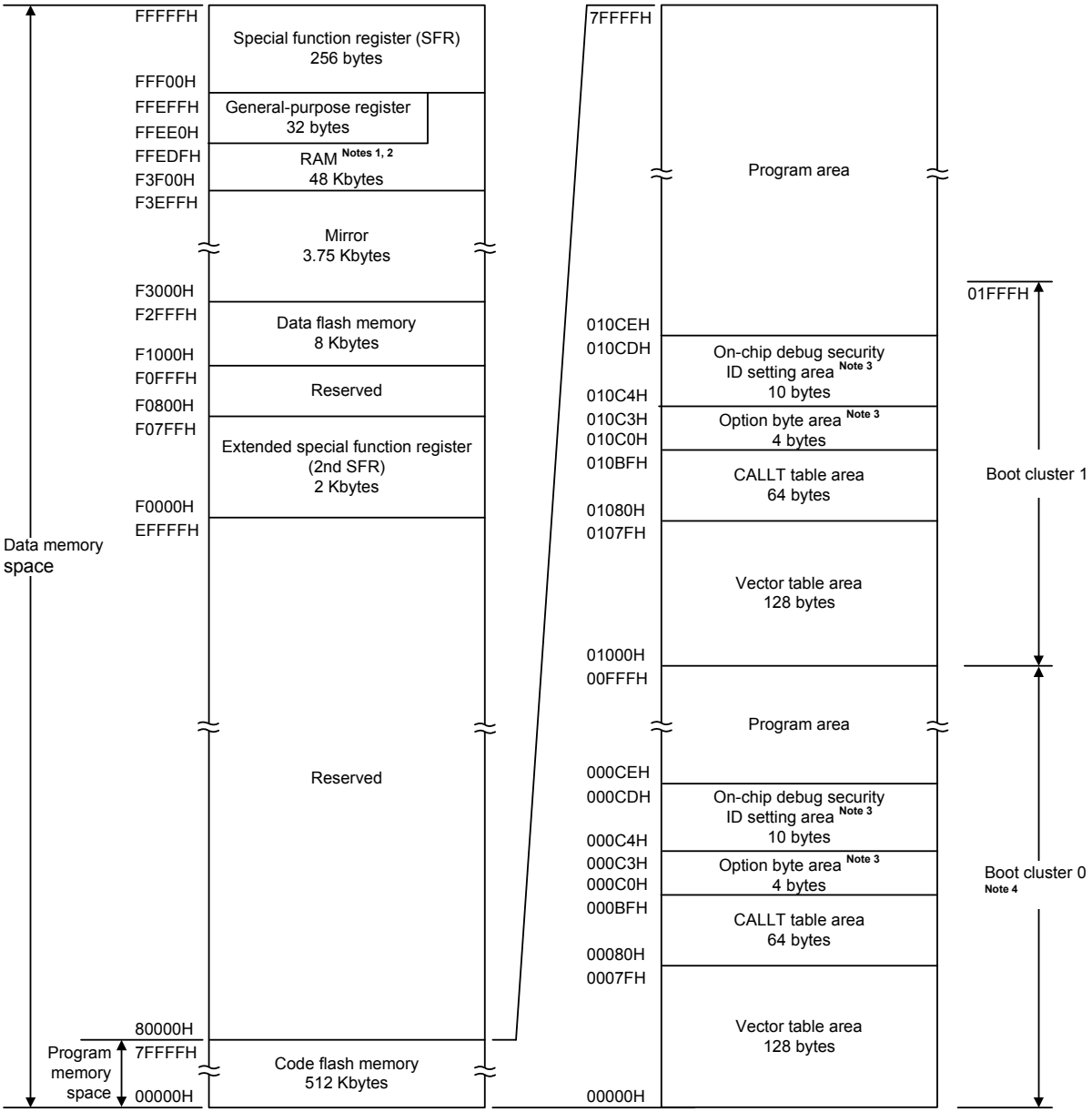
Note 3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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 **27.6 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 24.3.3 RAM parity error detection function.

Figure 4 - 3 Memory Map (R5F11FLL)



Note 1. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory. Also, use of the area F3F00H to F4309H is prohibited, because this area is used for each library.

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 **27.6 Security Settings**).

Note 5. Use of the area F4300H to F46FFH is prohibited when using the on-chip debugging trace function.

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 24.3.3 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 **Tables 4 - 1** to **4 - 4 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 4 - 1 Correspondence Between Address Values and Block Numbers in Flash Memory (1/4)

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 08BFFH	22H	10800H to 10BFFH	42H	18800H to 18BFFH	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

Table 4 - 2 Correspondence Between Address Values and Block Numbers in Flash Memory (2/4)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
20000H to 203FFH	80H	28000H to 283FFH	A0H	30000H to 303FFH	C0H	38000H to 383FFH	E0H
20400H to 207FFH	81H	28400H to 287FFH	A1H	30400H to 307FFH	C1H	38400H to 387FFH	E1H
20800H to 20BFFH	82H	28800H to 28BFFH	A2H	30800H to 30BFFH	C2H	38800H to 38BFFH	E2H
20C00H to 20FFFH	83H	28C00H to 28FFFH	A3H	30C00H to 30FFFH	C3H	38C00H to 38FFFH	E3H
21000H to 213FFH	84H	29000H to 293FFH	A4H	31000H to 313FFH	C4H	39000H to 393FFH	E4H
21400H to 217FFH	85H	29400H to 297FFH	A5H	31400H to 317FFH	C5H	39400H to 397FFH	E5H
21800H to 21BFFH	86H	29800H to 29BFFH	A6H	31800H to 31BFFH	C6H	39800H to 39BFFH	E6H
21C00H to 21FFFH	87H	29C00H to 29FFFH	A7H	31C00H to 31FFFH	C7H	39C00H to 39FFFH	E7H
22000H to 223FFH	88H	2A000H to 2A3FFH	A8H	32000H to 323FFH	C8H	3A000H to 3A3FFH	E8H
22400H to 227FFH	89H	2A400H to 2A7FFH	A9H	32400H to 327FFH	C9H	3A400H to 3A7FFH	E9H
22800H to 22BFFH	8AH	2A800H to 2ABFFH	AAH	32800H to 32BFFH	CAH	3A800H to 3ABFFH	EAH
22C00H to 22FFFH	8BH	2AC00H to 2AFFFH	ABH	32C00H to 32FFFH	CBH	3AC00H to 3AFFFH	EBH
23000H to 233FFH	8CH	2B000H to 2B3FFH	ACH	33000H to 333FFH	CCH	3B000H to 3B3FFH	ECH
23400H to 237FFH	8DH	2B400H to 2B7FFH	ADH	33400H to 337FFH	CDH	3B400H to 3B7FFH	EDH
23800H to 23BFFH	8EH	2B800H to 2BBFFH	AEH	33800H to 33BFFH	CEH	3B800H to 3BBFFH	EEH
23C00H to 23FFFH	8FH	2BC00H to 2BFFFH	AFH	33C00H to 33FFFH	CFH	3BC00H to 3BFFFH	EFH
24000H to 243FFH	90H	2C000H to 2C3FFH	B0H	34000H to 343FFH	D0H	3C000H to 3C3FFH	F0H
24400H to 247FFH	91H	2C400H to 2C7FFH	B1H	34400H to 347FFH	D1H	3C400H to 3C7FFH	F1H
24800H to 24BFFH	92H	2C800H to 2CBFFH	B2H	34800H to 34BFFH	D2H	3C800H to 3CBFFH	F2H
24C00H to 24FFFH	93H	2CC00H to 2CFFFH	B3H	34C00H to 34FFFH	D3H	3CC00H to 3CFFFH	F3H
25000H to 253FFH	94H	2D000H to 2D3FFH	B4H	35000H to 353FFH	D4H	3D000H to 3D3FFH	F4H
25400H to 257FFH	95H	2D400H to 2D7FFH	B5H	35400H to 357FFH	D5H	3D400H to 3D7FFH	F5H
25800H to 25BFFH	96H	2D800H to 2DBFFH	B6H	35800H to 35BFFH	D6H	3D800H to 3DBFFH	F6H
25C00H to 25FFFH	97H	2DC00H to 2DFFFH	B7H	35C00H to 35FFFH	D7H	3DC00H to 3DFFFH	F7H
26000H to 263FFH	98H	2E000H to 2E3FFH	B8H	36000H to 363FFH	D8H	3E000H to 3E3FFH	F8H
26400H to 267FFH	99H	2E400H to 2E7FFH	B9H	36400H to 367FFH	D9H	3E400H to 3E7FFH	F9H
26800H to 26BFFH	9AH	2E800H to 2EBFFH	BAH	36800H to 36BFFH	DAH	3E800H to 3EBFFH	FAH
26C00H to 26FFFH	9BH	2EC00H to 2EFFFH	BBH	36C00H to 36FFFH	DBH	3EC00H to 3EFFFH	FBH
27000H to 273FFH	9CH	2F000H to 2F3FFH	BCH	37000H to 373FFH	DCH	3F000H to 3F3FFH	FCH
27400H to 277FFH	9DH	2F400H to 2F7FFH	BDH	37400H to 377FFH	DDH	3F400H to 3F7FFH	FDH
27800H to 27BFFH	9EH	2F800H to 2FBFFH	BEH	37800H to 37BFFH	DEH	3F800H to 3FBFFH	FEH
27C00H to 27FFFH	9FH	2FC00H to 2FFFFH	BFH	37C00H to 37FFFH	DFH	3FC00H to 3FFFFH	FFH

Remark R5F11FLJ: Block numbers 00H to FFH

Table 4 - 3 Correspondence Between Address Values and Block Numbers in Flash Memory (3/4)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
40000H to 403FFH	100H	48000H to 483FFH	120H	50000H to 503FFH	140H	58000H to 583FFH	160H
40400H to 407FFH	101H	48400H to 487FFH	121H	50400H to 507FFH	141H	58400H to 587FFH	161H
40800H to 40BFFH	102H	48800H to 48BFFH	122H	50800H to 50BFFH	142H	58800H to 58BFFH	162H
40C00H to 40FFFH	103H	48C00H to 48FFFH	123H	50C00H to 50FFFH	143H	58C00H to 58FFFH	163H
41000H to 413FFH	104H	49000H to 493FFH	124H	51000H to 513FFH	144H	59000H to 593FFH	164H
41400H to 417FFH	105H	49400H to 497FFH	125H	51400H to 517FFH	145H	59400H to 597FFH	165H
41800H to 41BFFH	106H	49800H to 49BFFH	126H	51800H to 51BFFH	146H	59800H to 59BFFH	166H
41C00H to 41FFFH	107H	49C00H to 49FFFH	127H	51C00H to 51FFFH	147H	59C00H to 59FFFH	167H
42000H to 423FFH	108H	4A000H to 4A3FFH	128H	52000H to 523FFH	148H	5A000H to 5A3FFH	168H
42400H to 427FFH	109H	4A400H to 4A7FFH	129H	52400H to 527FFH	149H	5A400H to 5A7FFH	169H
42800H to 42BFFH	10AH	4A800H to 4ABFFH	12AH	52800H to 52BFFH	14AH	5A800H to 5ABFFH	16AH
42C00H to 42FFFH	10BH	4AC00H to 4AFFFH	12BH	52C00H to 52FFFH	14BH	5AC00H to 5AFFFH	16BH
43000H to 433FFH	10CH	4B000H to 4B3FFH	12CH	53000H to 533FFH	14CH	5B000H to 5B3FFH	16CH
43400H to 437FFH	10DH	4B400H to 4B7FFH	12DH	53400H to 537FFH	14DH	5B400H to 5B7FFH	16DH
43800H to 43BFFH	10EH	4B800H to 4BBFFH	12EH	53800H to 53BFFH	14EH	5B800H to 5BBFFH	16EH
43C00H to 43FFFH	10FH	4BC00H to 4BFFFH	12FH	53C00H to 53FFFH	14FH	5BC00H to 5BFFFH	16FH
44000H to 443FFH	110H	4C000H to 4C3FFH	130H	54000H to 543FFH	150H	5C000H to 5C3FFH	170H
44400H to 447FFH	111H	4C400H to 4C7FFH	131H	54400H to 547FFH	151H	5C400H to 5C7FFH	171H
44800H to 44BFFH	112H	4C800H to 4CBFFH	132H	54800H to 54BFFH	152H	5C800H to 5CBFFH	172H
44C00H to 44FFFH	113H	4CC00H to 4CFFFH	133H	54C00H to 54FFFH	153H	5CC00H to 5CFFFH	173H
45000H to 453FFH	114H	4D000H to 4D3FFH	134H	55000H to 553FFH	154H	5D000H to 5D3FFH	174H
45400H to 457FFH	115H	4D400H to 4D7FFH	135H	55400H to 557FFH	155H	5D400H to 5D7FFH	175H
45800H to 45BFFH	116H	4D800H to 4DBFFH	136H	55800H to 55BFFH	156H	5D800H to 5DBFFH	176H
45C00H to 45FFFH	117H	4DC00H to 4DFFFH	137H	55C00H to 55FFFH	157H	5DC00H to 5DFFFH	177H
46000H to 463FFH	118H	4E000H to 4E3FFH	138H	56000H to 563FFH	158H	5E000H to 5E3FFH	178H
46400H to 467FFH	119H	4E400H to 4E7FFH	139H	56400H to 567FFH	159H	5E400H to 5E7FFH	179H
46800H to 46BFFH	11AH	4E800H to 4EBFFH	13AH	56800H to 56BFFH	15AH	5E800H to 5EBFFH	17AH
46C00H to 46FFFH	11BH	4EC00H to 4EFFFH	13BH	56C00H to 56FFFH	15BH	5EC00H to 5EFFFH	17BH
47000H to 473FFH	11CH	4F000H to 4F3FFH	13CH	57000H to 573FFH	15CH	5F000H to 5F3FFH	17CH
47400H to 477FFH	11DH	4F400H to 4F7FFH	13DH	57400H to 577FFH	15DH	5F400H to 5F7FFH	17DH
47800H to 47BFFH	11EH	4F800H to 4FBFFH	13EH	57800H to 57BFFH	15EH	5F800H to 5FBFFH	17EH
47C00H to 47FFFH	11FH	4FC00H to 4FFFFH	13FH	57C00H to 57FFFH	15FH	5FC00H to 5FFFFH	17FH

Remark R5F11FLK: Block numbers 00H to 17FH

Table 4 - 4 Correspondence Between Address Values and Block Numbers in Flash Memory (4/4)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
60000H to 603FFH	180H	68000H to 683FFH	1A0H	70000H to 703FFH	1C0H	78000H to 783FFH	1E0H
60400H to 607FFH	181H	68400H to 687FFH	1A1H	70400H to 707FFH	1C1H	78400H to 787FFH	1E1H
60800H to 60BFFH	182H	68800H to 68BFFH	1A2H	70800H to 70BFFH	1C2H	78800H to 78BFFH	1E2H
60C00H to 60FFFH	183H	68C00H to 68FFFH	1A3H	70C00H to 70FFFH	1C3H	78C00H to 78FFFH	1E3H
61000H to 613FFH	184H	69000H to 693FFH	1A4H	71000H to 713FFH	1C4H	79000H to 793FFH	1E4H
61400H to 617FFH	185H	69400H to 697FFH	1A5H	71400H to 717FFH	1C5H	79400H to 797FFH	1E5H
61800H to 61BFFH	186H	69800H to 69BFFH	1A6H	71800H to 71BFFH	1C6H	79800H to 79BFFH	1E6H
61C00H to 61FFFH	187H	69C00H to 69FFFH	1A7H	71C00H to 71FFFH	1C7H	79C00H to 79FFFH	1E7H
62000H to 623FFH	188H	6A000H to 6A3FFH	1A8H	72000H to 723FFH	1C8H	7A000H to 7A3FFH	1E8H
62400H to 627FFH	189H	6A400H to 6A7FFH	1A9H	72400H to 727FFH	1C9H	7A400H to 7A7FFH	1E9H
62800H to 62BFFH	18AH	6A800H to 6ABFFH	1AAH	72800H to 72BFFH	1CAH	7A800H to 7ABFFH	1EAH
62C00H to 62FFFH	18BH	6AC00H to 6AFFFH	1ABH	72C00H to 72FFFH	1CBH	7AC00H to 7AFFFH	1EBH
63000H to 633FFH	18CH	6B000H to 6B3FFH	1ACH	73000H to 733FFH	1CCH	7B000H to 7B3FFH	1ECH
63400H to 637FFH	18DH	6B400H to 6B7FFH	1ADH	73400H to 737FFH	1CDH	7B400H to 7B7FFH	1EDH
63800H to 63BFFH	18EH	6B800H to 6BBFFH	1AEH	73800H to 73BFFH	1CEH	7B800H to 7BBFFH	1EEH
63C00H to 63FFFH	18FH	6BC00H to 6BFFFH	1AFH	73C00H to 73FFFH	1CFH	7BC00H to 7BFFFH	1EFH
64000H to 643FFH	190H	6C000H to 6C3FFH	1B0H	74000H to 743FFH	1D0H	7C000H to 7C3FFH	1F0H
64400H to 647FFH	191H	6C400H to 6C7FFH	1B1H	74400H to 747FFH	1D1H	7C400H to 7C7FFH	1F1H
64800H to 64BFFH	192H	6C800H to 6CBFFH	1B2H	74800H to 74BFFH	1D2H	7C800H to 7CBFFH	1F2H
64C00H to 64FFFH	193H	6CC00H to 6CFFFH	1B3H	74C00H to 74FFFH	1D3H	7CC00H to 7CFFFH	1F3H
65000H to 653FFH	194H	6D000H to 6D3FFH	1B4H	75000H to 753FFH	1D4H	7D000H to 7D3FFH	1F4H
65400H to 657FFH	195H	6D400H to 6D7FFH	1B5H	75400H to 757FFH	1D5H	7D400H to 7D7FFH	1F5H
65800H to 65BFFH	196H	6D800H to 6DBFFH	1B6H	75800H to 75BFFH	1D6H	7D800H to 7DBFFH	1F6H
65C00H to 65FFFH	197H	6DC00H to 6DFFFH	1B7H	75C00H to 75FFFH	1D7H	7DC00H to 7DFFFH	1F7H
66000H to 663FFH	198H	6E000H to 6E3FFH	1B8H	76000H to 763FFH	1D8H	7E000H to 7E3FFH	1F8H
66400H to 667FFH	199H	6E400H to 6E7FFH	1B9H	76400H to 767FFH	1D9H	7E400H to 7E7FFH	1F9H
66800H to 66BFFH	19AH	6E800H to 6EBFFH	1BAH	76800H to 76BFFH	1DAH	7E800H to 7EBFFH	1FAH
66C00H to 66FFFH	19BH	6EC00H to 6EFFFH	1BBH	76C00H to 76FFFH	1DBH	7EC00H to 7EFFFH	1FBH
67000H to 673FFH	19CH	6F000H to 6F3FFH	1BCH	77000H to 773FFH	1DCH	7F000H to 7F3FFH	1FCH
67400H to 677FFH	19DH	6F400H to 6F7FFH	1BDH	77400H to 777FFH	1DDH	7F400H to 7F7FFH	1FDH
67800H to 67BFFH	19EH	6F800H to 6FBFFH	1BEH	77800H to 77BFFH	1DEH	7F800H to 7FBFFH	1FEH
67C00H to 67FFFH	19FH	6FC00H to 6FFFFH	1BFH	77C00H to 77FFFH	1DFH	7FC00H to 7FFFFH	1FFH

Remark R5F11FLL: Block numbers 00H to 1FFH

4.1.1 Internal program memory space

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

For details, see **CHAPTER 19 INTERRUPT FUNCTIONS**.

(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 26 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 28 ON-CHIP DEBUG FUNCTION**.

4.1.2 Mirror area

The code flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, 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.
The mirror area can only be read and no instruction can be fetched from this area.
The following show examples.

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 4 - 4 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	10000H to 1FFFFH is mirrored to F0000H to FFFFFH

Caution After setting the PMC register, wait for at least one instruction and access the mirror area.

4.1.3 Internal data memory space

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 RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

Caution 3. 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.
R5F11FLL: F3F00H to F4309H

Caution 4. The internal RAM area in the following products cannot be used as stack memory when using the on-chip debugging trace function.
R5F11FLL: F4300H to F46FFH

4.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH (see **Tables 4 - 5** to **4 - 9** in **4.2.4 Special function registers (SFRs)**).

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

4.1.5 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 4 - 10** to **4 - 16** in **4.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**).

Caution 1. Do not access addresses to which extended SFRs are not assigned.

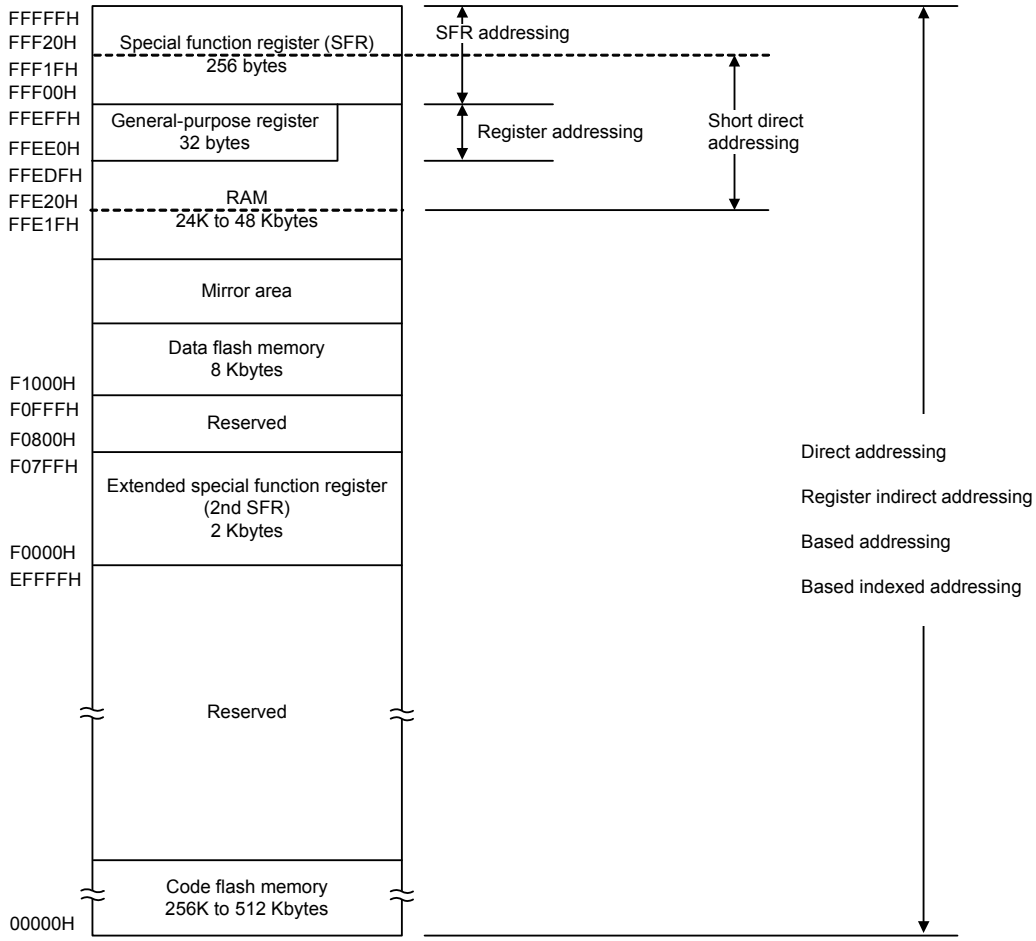
Caution 2. When accessing timer RJ counter register 0 (TRJ0) allocated in F0500H of the extended SFR (2nd SFR), the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to timer RJ counter register 0 (TRJ0) is one clock for both writing and reading.

4.1.6 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 RL78/G1H, 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 4 - 5 shows correspondence between data memory and addressing.

Figure 4 - 5 Correspondence Between Data Memory and Addressing



4.2 Processor Registers

The RL78/G1H products incorporate the following processor registers.

4.2.1 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).

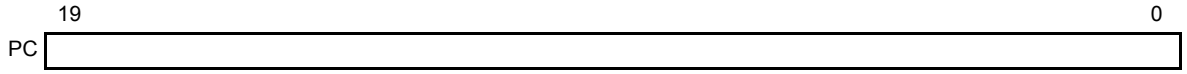
(1) Program counter (PC)

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 4 - 6 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 4 - 7 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 maskable interrupt requests 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 or equal, 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 **19.3.3**) can not be acknowledged. Actual vectored interrupt requests 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 4 - 8 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, and a DTC transfer destination/transfer source 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.
R5F11FLL: F3F00H to F4309H

Caution 5. The internal RAM area in the following products cannot be used as stack memory when using the on-chip debugging trace function.
R5F11FLL: F4300H to F46FFH

4.2.2 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).

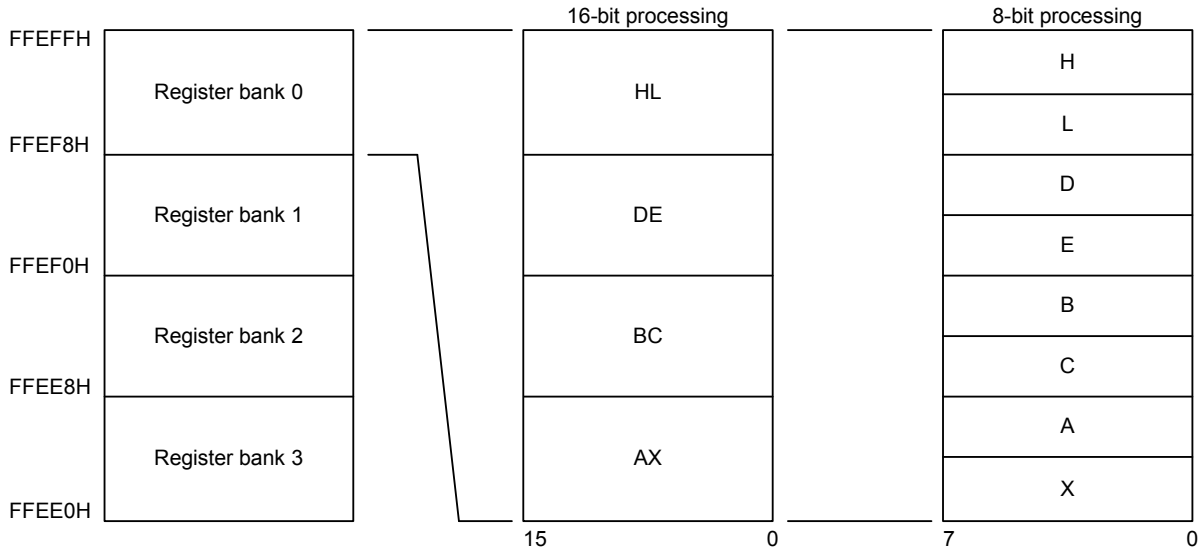
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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 4 - 9 Configuration of General-Purpose Registers

(a) Function name



4.2.3 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.

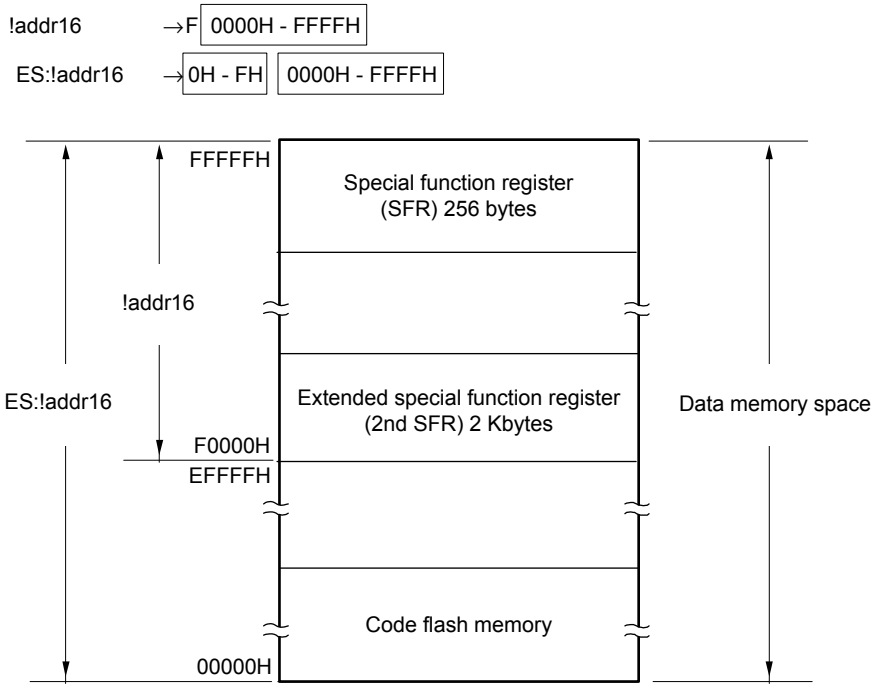
Figure 4 - 10 Configuration of ES and CS Registers

	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 4 - 11 Extension of Data Area Which Can Be Accessed



4.2.4 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 4 - 5 to 4 - 9 give 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 **4.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 4 - 5 Special Function Register (SFR) List (1/5)

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
FFF01H	Port register 1	P1		R/W	√	√	—	00H
FFF02H	Port register 2	P2		R/W	√	√	—	00H
FFF03H	Port register 3	P3		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	P7		R/W	√	√	—	00H
FFF08H	Port register 8	P8		R/W	√	√	—	00H
FFF0AH	Port register 10	P10		R/W	√	√	—	00H
FFF0BH	Port register 11	P11		R/W	√	√	—	00H
FFF0CH	Port register 12	P12		R/W	√	√	—	Undefined
FFF0DH	Port register 13	P13		R/W	√	√	—	Undefined
FFF0EH	Port register 14	P14		R/W	√	√	—	00H
FFF0FH	Port register 15	P15		R/W	√	√	—	00H
FFF14H	Serial data register 12	RXD3/ SIO30	SDR12	R/W	—	√	√	0000H
FFF15H		—			—	—		
FFF16H	Serial data register 13	RXD3	SDR13	R/W	—	√	√	0000H
FFF17H		—			—	—		
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
FFF21H	Port mode register 1	PM1		R/W	√	√	—	FFH
FFF22H	Port mode register 2	PM2		R/W	√	√	—	FFH
FFF23H	Port mode register 3	PM3		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	PM7		R/W	√	√	—	FFH

Table 4 - 6 Special Function Register (SFR) List (2/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF28H	Port mode register 8	PM8		R/W	√	√	—	FFH
FFF2AH	Port mode register 10	PM10		R/W	√	√	—	FFH
FFF2BH	Port mode register 11	PM11		R/W	√	√	—	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	—	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	—	FFH
FFF2FH	Port mode register 15	PM15		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
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	—	√	√	0000H
FFF45H		—			—	—		
FFF46H	Serial data register 03	RXD1	SDR03	R/W	—	√	√	0000H
FFF47H		—			—	—		
FFF48H	Serial data register 10	SIO20	SDR10	R/W	—	√	√	0000H
FFF49H		—			—	—		
FFF4AH	Serial data register 11	SIO21	SDR11	R/W	—	√	√	0000H
FFF4BH		—			—	—		
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
FFF64H	Timer data register 02	TDR02		R/W	—	—	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	—	√	√	00H
FFF67H		TDR03H			—	√		00H
FFF70H	Timer data register 10	TDR10		R/W	—	—	√	0000H
FFF71H								

Table 4 - 7 Special Function Register (SFR) List (3/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF72H	Timer data register 11	TDR11L	TDR11	R/W	—	√	√	00H
FFF73H		TDR11H			—	√		00H
FFF74H	Timer data register 12	TDR12		R/W	—	—	√	0000H
FFF75H								
FFF76H	Timer data register 13	TDR13L	TDR13	R/W	—	√	√	00H
FFF77H		TDR13H			—	√		00H
FFF90H	12-bit interval timer control register	ITMC		R/W	—	—	√	0FFFH
FFF91H								
FFF92H	Second count register	SEC		R/W	—	√	—	00H
FFF93H	Minute count register	MIN		R/W	—	√	—	00H
FFF94H	Hour count register	HOUR		R/W	—	√	—	12H Note
FFF95H	Week count register	WEEK		R/W	—	√	—	00H
FFF96H	Day count register	DAY		R/W	—	√	—	01H
FFF97H	Month count register	MONTH		R/W	—	√	—	01H
FFF98H	Year count register	YEAR		R/W	—	√	—	00H
FFF99H	Watch error correction register	SUBCUD		R/W	—	√	—	00H
FFF9AH	Alarm minute register	ALARMWM		R/W	—	√	—	00H
FFF9BH	Alarm hour register	ALARMWH		R/W	—	√	—	12H
FFF9CH	Alarm week register	ALARMWW		R/W	—	√	—	00H
FFF9DH	Real-time clock control register 0	RTCC0		R/W	√	√	—	00H
FFF9EH	Real-time clock control register 1	RTCC1		R/W	√	√	—	00H
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
FFFA6H	Clock output select register 1	CKS1		R/W	√	√	—	00H

Note The value of this register is 00H if the AMPM bit (bit 3 of real-time clock control register 0 (RTCC0)) is set to 1 after reset.

Table 4 - 8 Special Function Register (SFR) List (4/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFA8H	Reset control flag register	RESF		R	—	√	—	Undefined Note 1
FFFA9H	Voltage detection register	LVIM		R/W	√	√	—	00H Note 1
FFFAAH	Voltage detection level register	LVIS		R/W	√	√	—	00H/01H/81H Note 1
FFFABH	Watchdog timer enable register	WDTE		R/W	—	√	—	9AH/1AH Note 2
FFFACH	CRC input register	CRCIN		R/W	—	√	—	00H
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH	Priority specification flag register 12H	PR12H		R/W	√	√		FFH

Note 1. 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							
LVIS		Cleared (00H/01H/81H)						

Note 2. The reset value of the WDTE register is determined by the setting of the option byte.

Table 4 - 9 Special Function Register (SFR) List (5/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	Interrupt request flag register 1H	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
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		PR11H		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

Remark For extended SFRs (2nd SFRs), see **Tables 4 - 10** to **4 - 16 Extended SFR (2nd SFR) List**.

4.2.5 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 4 - 10 to 4 - 16 give 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 **4.2.4 Special function registers (SFRs)**.

Table 4 - 10 Extended Special Function Register (2nd SFR) List (1/7)

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
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	—	00H
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	—	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	—	01H
F0037H	Pull-up resistor option register 7	PU7	R/W	√	√	—	00H
F0038H	Pull-up resistor option register 8	PU8	R/W	√	√	—	00H
F003AH	Pull-up resistor option register 10	PU10	R/W	√	√	—	00H
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	—	00H
F003EH	Pull-up resistor option register 14	PU14	R/W	√	√	—	00H
F0040H	Port input mode register 0	PIM0	R/W	√	√	—	00H
F0048H	Port input mode register 8	PIM8	R/W	√	√	—	00H
F004EH	Port input mode register 14	PIM14	R/W	√	√	—	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	—	00H
F0057H	Port output mode register 7	POM7	R/W	√	√	—	00H
F0058H	Port output mode register 8	POM8	R/W	√	√	—	00H
F005EH	Port output mode register 14	POM14	R/W	√	√	—	00H
F0060H	Port mode control register 0	PMC0	R/W	√	√	—	FFH
F006AH	Port mode control register 10	PMC10	R/W	√	√	—	FFH
F006CH	Port mode control register 12	PMC12	R/W	√	√	—	FFH
F006EH	Port mode control register 14	PMC14	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
F0076H	A/D port configuration register	ADPC	R/W	—	√	—	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	—	√	—	00H
F007AH	Peripheral enable register 1	PER1	R/W	√	√	—	00H
F007BH	Port mode select register	PMS	R/W	√	√	—	00H
F0090H	Data flash control register	DFLCTL	R/W	√	√	—	00H
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM	R/W	—	√	—	Undefined **Note 1**
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV	R/W	—	√	—	Undefined **Note 2**

Note 1. The value after a reset is adjusted at the time of shipment.

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

Table 4 - 11 Extended Special Function Register (2nd SFR) List (2/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	—	00H
F00F3H	Subsystem clock supply mode control register	OSMC		R/W	—	√	—	00H
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	—	00H
F00FEH	BCD correction result register	BCDADJ		R	—	√	—	Undefined
F0104H	Serial status register 02	SSR02L	SSR02	R	—	√	√	0000H
F0105H		—			—	—		
F0106H	Serial status register 03	SSR03L	SSR03	R	—	√	√	0000H
F0107H		—			—	—		
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		—			—	—		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	—	√	√	0000H
F010DH		—			—	—		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	—	√	√	0000H
F010FH		—			—	—		
F0114H	Serial mode register 02	SMR02		R/W	—	—	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	—	—	√	0020H
F0117H								
F011CH	Serial communication operation setting register 02	SCR02		R/W	—	—	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	—	—	√	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		—			—	—		

Table 4 - 12 Extended Special Function Register (2nd SFR) List (3/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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	—	—	√	0F0FH
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		—			—	—		
F0140H	Serial status register 10	SSR10L	SSR10	R	—	√	√	0000H
F0141H		—			—	—		
F0142H	Serial status register 11	SSR11L	SSR11	R	—	√	√	0000H
F0143H		—			—	—		
F0144H	Serial status register 12	SSR12L	SSR12	R	—	√	√	0000H
F0145H		—			—	—		
F0146H	Serial status register 13	SSR13L	SSR13	R	—	√	√	0000H
F0147H		—			—	—		
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	—	√	√	0000H
F0149H		—			—	—		
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	—	√	√	0000H
F014BH		—			—	—		
F014CH	Serial flag clear trigger register 12	SIR12L	SIR12	R/W	—	√	√	0000H
F014DH		—			—	—		
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	—	√	√	0000H
F014FH		—			—	—		
F0150H	Serial mode register 10	SMR10		R/W	—	—	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	—	—	√	0020H
F0153H								
F0154H	Serial mode register 12	SMR12		R/W	—	—	√	0020H
F0155H								
F0156H	Serial mode register 13	SMR13		R/W	—	—	√	0020H
F0157H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	—	—	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	—	—	√	0087H
F015BH								

Table 4 - 13 Extended Special Function Register (2nd SFR) List (4/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F015CH	Serial communication operation setting register 12	SCR12		R/W	—	—	√	0087H
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W	—	—	√	0087H
F015FH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0161H		—			—	—		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		—			—	—		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		—			—	—		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	—	√	√	0000H
F0167H		—			—	—		
F0168H	Serial output register 1	SO1		R/W	—	—	√	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	√	√	√	0000H
F016BH		—			—	—		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	—	√	√	0000H
F0175H		—			—	—		
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								
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								
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		—			—	—		

Table 4 - 14 Extended Special Function Register (2nd SFR) List (5/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01A6H	Timer status register 03	TSR03L	TSR03	R	—	√	√	0000H
F01A7H		—			—	—		
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	Timer channel stop register 0	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		—			—	—		
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		—			—	—		
F01C0H	Timer counter register 10	TCR10		R	—	—	√	FFFFH
F01C1H								
F01C2H	Timer counter register 11	TCR11		R	—	—	√	FFFFH
F01C3H								
F01C4H	Timer counter register 12	TCR12		R	—	—	√	FFFFH
F01C5H								
F01C6H	Timer counter register 13	TCR13		R	—	—	√	FFFFH
F01C7H								
F01D0H	Timer mode register 10	TMR10		R/W	—	—	√	0000H
F01D1H								
F01D2H	Timer mode register 11	TMR11		R/W	—	—	√	0000H
F01D3H								
F01D4H	Timer mode register 12	TMR12		R/W	—	—	√	0000H
F01D5H								
F01D6H	Timer mode register 13	TMR13		R/W	—	—	√	0000H
F01D7H								
F01E0H	Timer status register 10	TSR10L	TSR10	R/W	—	√	√	0000H
F01E1H		—			—	—		
F01E2H	Timer status register 11	TSR11L	TSR11	R/W	—	√	√	0000H
F01E3H		—			—	—		
F01E4H	Timer status register 12	TSR12L	TSR12	R/W	—	√	√	0000H
F01E5H		—			—	—		

Table 4 - 15 Extended Special Function Register (2nd SFR) List (6/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01E6H	Timer status register 13	TSR13L	TSR13	R/W	—	√	√	0000H
F01E7H		—			—	—		
F01F0H	Timer channel enable status register 1	TE1L	TE1	R/W	—	√	√	0000H
F01F1H		—			—	—		
F01F2H	Timer channel start register 1	TS1L	TS1	R/W	—	√	√	0000H
F01F3H		—			—	—		
F01F4H	Timer channel stop register 1	TT1L	TT1	R/W	—	√	√	0000H
F01F5H		—			—	—		
F01F6H	Timer clock select register 1	TPS1		R/W	—	—	√	0000H
F01F7H								
F01F8H	Timer output register 1	TO1L	TO1	R/W	—	√	√	0000H
F01F9H		—			—	—		
F01FAH	Timer output enable register 1	TOE1L	TOE1	R/W	—	√	√	0000H
F01FBH		—			—	—		
F01FCH	Timer output level register 1	TOL1L	TOL1	R/W	—	√	√	0000H
F01FDH		—			—	—		
F01FEH	Timer output mode register 1	TOM1L	TOM1	R/W	—	√	√	0000H
F01FFH		—			—	—		
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
F0240H	Timer RJ control register 0	TRJCR0		R/W	—	√	—	00H
F0242H	Timer RJ mode register 0	TRJMR0		R/W	√	√	—	00H
F02E0H	DTC base address register	DTCBAR		R/W	√	√	—	FDH
F02E8H	DTC activation enable register 0	DTCEN0		R/W	√	√	—	00H
F02E9H	DTC activation enable register 1	DTCEN1		R/W	√	√	—	00H
F02EAH	DTC activation enable register 2	DTCEN2		R/W	√	√	—	00H
F02EBH	DTC activation enable register 3	DTCEN3		R/W	√	√	—	00H
F02ECH	DTC activation enable register 4	DTCEN4		R/W	√	√	—	00H
F02F0H	Flash memory CRC control register	CRC0CTL		R/W	√	√	—	00H
F02F2H	Flash memory CRC operation result register	PGCRCL		R/W	—	—	√	0000H

Table 4 - 16 Extended Special Function Register (2nd SFR) List (7/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F02FAH	CRC data register	CRCD	R/W	—	—	√	0000H
F0300H	Event output destination select register 00	ELSELR00	R/W	—	√	—	00H
F0303H	Event output destination select register 03	ELSELR03	R/W	—	√	—	00H
F0304H	Event output destination select register 04	ELSELR04	R/W	—	√	—	00H
F0307H	Event output destination select register 07	ELSELR07	R/W	—	√	—	00H
F030DH	Event output destination select register 13	ELSELR13	R/W	—	√	—	00H
F0310H	Event output destination select register 16	ELSELR16	R/W	—	√	—	00H
F0311H	Event output destination select register 17	ELSELR17	R/W	—	√	—	00H
F0312H	Event output destination select register 18	ELSELR18	R/W	—	√	—	00H
F0313H	Event output destination select register 19	ELSELR19	R/W	—	√	—	00H
F0314H	Event output destination select register 20	ELSELR20	R/W	—	√	—	00H
F0315H	Event output destination select register 21	ELSELR21	R/W	—	√	—	00H
F0316H	Event output destination select register 22	ELSELR22	R/W	—	√	—	00H
F0317H	Event output destination select register 23	ELSELR23	R/W	—	√	—	00H
F0500H	Timer RJ counter register 0	TRJ0	R/W	—	—	√	FFFFH
F0501H							

Remark For SFRs in the SFR area, see **Tables 4 - 5** to **4 - 9 SFR List**.

CHAPTER 5 PORT FUNCTIONS

5.1 Port Functions

The RL78/G1H microcontrollers are 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 3 PIN FUNCTIONS**.

The MCU internal pins described below must be set to output mode after reset release by using the software (set the port register and port mode register to 0).

- P00, P01, P05, P06, P17, P23 to P27, P41 to P47, P50 to P57, P64 to P67, P73 to P74, P83 to P87, P101, P102, P110, P111, P145 to P147, P150 to P154

5.2 Port Configuration

Ports include the following hardware.

Table 5 - 1 Port Configuration

Item	Configuration
Control registers	Port mode registers (PM0 to PM8, PM10 to PM12, PM14, PM15) Port registers (P0 to P8, P10 to P15) Pull-up resistor option registers (PU0, PU1, PU3, PU4, PU7, PU8, PU10, PU12, PU14) Port input mode registers (PIM0, PIM8, PIM14) Port output mode registers (POM0, POM7, POM8, POM14) Port mode control registers (PMC0, PMC10, PMC12, PMC14) A/D port configuration register (ADPC)
Port	Total: 41 (CMOS I/O: 31 (N-ch open-drain I/O [V_{DD} tolerance]: 15), CMOS input: 5, CMOS output: 1, N-ch open-drain I/O [6 V tolerance]: 4)
Pull-up resistor	Total: 28

5.2.1 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 P02 to P04 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 0 (PU0).

Input to the P03 and P04 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 0 (PIM0).

Output from the P02 to P04 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 0 (POM0).

To use P02 and P03 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 A/D converter analog input, serial interface data I/O, and clock I/O.

When reset signal is generated, the following configuration will be set.

- P04 pin..... Input port
- P02 and P03 pins..... Digital I/O is prohibited.

5.2.2 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P16 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 1 (PU1).

This port can also be used for serial interface data I/O, and clock output.

Reset signal generation sets port 1 to input mode.

P10 to P16 pins are used for internal connection between the MCU and RF transceiver. For details, see **CHAPTER 2 CONNECTION BETWEEN MCU AND RF TRANSCEIVER**.

5.2.3 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 and reference voltage (+ side and - side) input.

To use P20/ANI0, P21/ANI1, and P22/ANI2 as digital I/O pins, set them to digital I/O using the A/D port configuration register (ADPC). Use these pins starting from the upper bit.

To use P20/ANI0, P21/ANI1, and P22/ANI2 as analog input pins, set them to analog function using the A/D port configuration register (ADPC) and input mode using the PM2 register. Use these pins starting from the lower bit.

All pins are set in the analog function mode when the reset signal is generated.

Table 5 - 2 Setting Functions of P20/ANI0, P21/ANI1, P22/ANI2 Pins

ADPC Register	PM2 Register	ADS Register	P20/ANI0, P21/ANI1, P22/ANI2 Pins
Digital I/O selection	Input mode	—	Digital input
	Output mode	—	Digital output
Analog function selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P2 are set in the analog function mode when the reset signal is generated.

5.2.4 Port 3

Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30, P31 pins are used as input ports, use of on-chip pull-up resistors can be specified by pull-up resistor option register 3 (PU3).

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

Reset signal generation sets port 3 to input port.

P30 pin is used for internal connection between the MCU and RF transceiver. For details, see **CHAPTER 2 CONNECTION BETWEEN MCU AND RF TRANSCEIVER**. .

5.2.5 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 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.

5.2.6 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 P63 pins is N-ch open-drain output (6 V tolerance).

This port can also be used for serial interface data I/O and clock I/O.

Reset signal generation sets port 6 to input port.

5.2.7 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 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).

Output from the P71 pin can be specified as N-ch open-drain output (V_{DD} tolerance) using port output mode register 7 (POM7).

This port can also be used for serial interface data I/O, clock I/O, and external interrupt request input.

Reset signal generation sets port 7 to input port.

5.2.8 Port 8

Port 8 is an I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8). When 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 8 (PU8).

Input to the P80 and P81 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 8 (PIM8).

Output from the P80 to P82 pin can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 8 (POM8).

Reset signal generation sets port 8 to input port.

5.2.9 Port 10

Port 10 is an I/O port with an output latch. Port 10 can be set to the input mode or output mode in 1-bit units using port mode register 10 (PM10). When the P100 to P102 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 10 (PU10).
Set the P100 pin to digital I/O using port mode control register 10 (PMC10).
Reset signal generation sets P100 to prohibit digital I/O.

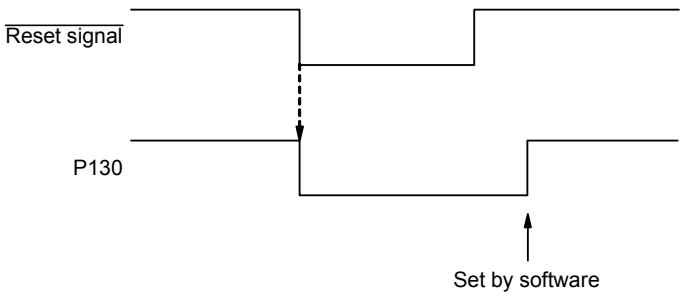
5.2.10 Port 12

P120 is an I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).
P121 to P124 are 4-bit input ports.
To use the P120 pin as a digital I/O port, set it to digital I/O using port mode control register 12 (PMC12).
This port can also be used for A/D converter analog input, resonator connection for the subsystem clock, external clock input for the main system clock, and external clock input for the subsystem clock.
Reset signal generation sets P120 to analog function, and sets P121 to P124 to input port.

5.2.11 Port 13

P130 is a 1-bit output-only port with an output latch. P137 is a 1-bit input-only port.
P130 is fixed an output port, and P137 is fixed an input ports.
This port can also be used for external interrupt request input.
Connect the P130 pin to STANDBY pin.

Remark When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.



5.2.12 Port 14

Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 to P144 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 14 (PU14).

Input to the P142 and P143 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 14 (PIM14).

Output from the P142 to P144 pin can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 14 (POM14).

To use the P147 set it to digital I/O using port mode control register 14 (PMC14).

This port can also be used for clock/buzzer output, external interrupt request input, serial interface data I/O, and clock I/O.

Reset signal generation sets port 14 to input port.

5.2.13 Port 15

Port 15 is an I/O port with an output latch. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15).

This port can also be used for A/D converter analog input.

To use P155/ANI13 and P156/ANI14 as digital I/O pins, set them to digital I/O using the A/D port configuration register (ADPC). Use these pins starting from the upper bit.

To use P155/ANI13 and P156/ANI14 as analog input pins, set them to analog function using the A/D port configuration register (ADPC) and input mode using the PM15 register. Use these pins starting from the lower bit.

Table 5 - 3 Setting Functions of P155/ANI13 and P156/ANI14 Pins

ADPC Register	PM15 Register	ADS Register	P155/ANI13 and P156/ANI14 Pins
Digital I/O selection	Input mode	—	Digital input
	Output mode	—	Digital output
Analog function selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P15 are set in the analog function mode when the reset signal is generated.

5.2.14 GPIO port

GPIO port is an I/O port with an output latch. For GPIO port, SCI20 enables the mode setting and I/O.

For details, see **CHAPTER 18 RF TRANSCEIVER**.

5.3 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)
- A/D port configuration register (ADPC)

Table 5 - 4 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits (1/2)

Port		Bit name					
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register
Port 0	2	PM02	P02	PU02	—	POM02	PMC02
	3	PM03	P03	PU03	PIM03	POM03	PMC03
	4	PM04	P04	PU04	PIM04	POM04	—
Port 1	0	PM10	P10	PU10	—	—	—
	1	PM11	P11	PU11	—	—	—
	2	PM12	P12	PU12	—	—	—
	3	PM13	P13	PU13	—	—	—
	4	PM14	P14	PU14	—	—	—
	5	PM15	P15	PU15	—	—	—
	6	PM16	P16	PU16	—	—	—
Port 2	0	PM20	P20	—	—	—	—
	1	PM21	P21	—	—	—	—
	2	PM22	P22	—	—	—	—
Port 3	0	PM30	P30	PU30	—	—	—
	1	PM31	P31	PU31	—	—	—
Port 4	0	PM40	P40	PU40	—	—	—
Port 6	0	PM60	P60	—	—	—	—
	1	PM61	P61	—	—	—	—
	2	PM62	P62	—	—	—	—
	3	PM63	P63	—	—	—	—
Port 7	0	PM70	P70	PU70	—	—	—
	1	PM71	P71	PU71	—	POM71	—
	2	PM72	P72	PU72	—	—	—
	5	PM75	P75	PU75	—	—	—
	6	PM76	P76	PU76	—	—	—
	7	PM77	P77	PU77	—	—	—
Port 8	0	PM80	P80	PU80	PIM80	POM80	—
	1	PM81	P81	PU81	PIM81	POM81	—
	2	PM82	P82	PU82	—	POM82	—
Port 10	0	PM100	P100	PU100	—	—	PMC100
Port 12	0	PM120	P120	PU120	—	—	PMC120
	1	—	P121	—	—	—	—
	2	—	P122	—	—	—	—
	3	—	P123	—	—	—	—
	4	—	P124	—	—	—	—
Port 13	0	—	P130	—	—	—	—
	7	—	P137	—	—	—	—

Table 5 - 5 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits (2/2)

Port		Bit name					
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register
Port 14	0	PM140	P140	PU140	—	—	—
	1	PM141	P141	PU141	—	—	—
	2	PM142	P142	PU142	PIM142	POM142	—
	3	PM143	P143	PU143	PIM143	POM143	—
	4	PM144	P144	PU144	—	POM144	—
Port 15	5	PM155	P155	—	—	—	—
	6	PM156	P156	—	—	—	—

5.3.1 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 **5.5 Register Settings When Using Alternate Function**.

Figure 5 - 1 Format of Port mode register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06 Note 1	PM05 Note 1	PM04	PM03	PM02	PM01 Note 1	PM00 Note 1	FFF20H	FFH	R/W
PM1	PM17 Note 1	PM16 Note 2	PM15 Note 2	PM14 Note 3	PM13 Note 2	PM12 Note 2	PM11 Note 2	PM10 Note 2	FFF21H	FFH	R/W
PM2	PM27 Note 1	PM26 Note 1	PM25 Note 1	PM24 Note 1	PM23 Note 1	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30 Note 3	FFF23H	FFH	R/W
PM4	PM47 Note 1	PM46 Note 1	PM45 Note 1	PM44 Note 1	PM43 Note 1	PM42 Note 1	PM41 Note 1	PM40	FFF24H	FFH	R/W
PM5	PM57 Note 1	PM56 Note 1	PM55 Note 1	PM54 Note 1	PM53 Note 1	PM52 Note 1	PM51 Note 1	PM50 Note 1	FFF25H	FFH	R/W
PM6	PM67 Note 1	PM66 Note 1	PM65 Note 1	PM64 Note 1	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74 Note 1	PM73 Note 1	PM72	PM71	PM70	FFF27H	FFH	R/W
PM8	PM87 Note 1	PM86 Note 1	PM85 Note 1	PM84 Note 1	PM83 Note 1	PM82	PM81	PM80	FFF28H	FFH	R/W
PM10	1	1	1	1	1	PM102 Note 1	PM101 Note 1	PM100	FFF2AH	FFH	R/W
PM11	1	1	1	1	1	1	PM111 Note 1	PM110 Note 1	FFF2BH	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	PM147 Note 1	PM146 Note 1	PM145 Note 1	PM144	PM143	PM142	PM141	PM140	FFF2EH	FFH	R/W
PM15	1	PM156	PM155	PM154 Note 1	PM153 Note 1	PM152 Note 1	PM151 Note 1	PM150 Note 1	FFF2FH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 8, 10 to 12, 14, 15; n = 0 to 7)
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))

Note 1. Control bits for internal pins. After reset release, be sure to set to output mode by software (set 0 to port mode register).

Note 2. Control bits for internal connecting pins. After reset release, be sure to set to output mode by software (set 0 to port mode register).

Note 3. Control bits for internal connecting pins. After reset release, be sure to set to input mode by software (set 1 to port mode register).

Caution Be sure to set bits that are not mounted to their initial values. For the register that has an instruction to clear the bit to 0, clear it to 0.

5.3.2 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 P20 to P22, P120 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 5 - 2 Format of Port register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	P06 Note 3	P05 Note 3	P04	P03	P02	P01 Note 3	P00 Note 3	FFF00H	00H (output latch)	R/W
P1	P17 Note 3	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	P27 Note 3	P26 Note 3	P25 Note 3	P24 Note 3	P23 Note 3	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	0	0	P31	P30	FFF03H	00H (output latch)	R/W
P4	P47 Note 3	P46 Note 3	P45 Note 3	P44 Note 3	P43 Note 3	P42 Note 3	P41 Note 3	P40	FFF04H	00H (output latch)	R/W
P5	P57 Note 3	P56 Note 3	P55 Note 3	P54 Note 3	P53 Note 3	P52 Note 3	P51 Note 3	P50 Note 3	FFF05H	00H (output latch)	R/W
P6	P67 Note 3	P66 Note 3	P65 Note 3	P64 Note 3	P63	P62	P61	P60	FFF06H	00H (output latch)	R/W
P7	P77	P76	P75	P74 Note 3	P73 Note 3	P72	P71	P70	FFF07H	00H (output latch)	R/W
P8	P87 Note 3	P86 Note 3	P85 Note 3	P84 Note 3	P83 Note 3	P82	P81	P80	FFF08H	00H (output latch)	R/W
P10	0	0	0	0	0	P102 Note 3	P101 Note 3	P100	FFF0AH	00H (output latch)	R/W
P11	0	0	0	0	0	0	P111 Note 3	P110 Note 3	FFF0BH	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W Note 1
P13	P137	0	0	0	0	0	0	P130	FFF0DH	Note 2	R/W Note 1
P14	P147 Note 3	P146 Note 3	P145 Note 3	P144	P143	P142	P141	P140	FFF0EH	00H (output latch)	R/W
P15	0	P156	P155	P154 Note 3	P513 Note 3	P152 Note 3	P151 Note 3	P150 Note 3	FFF0FH	00H (output latch)	R/W

Pmn	m = 0 to 8, 10 to 15; 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 to P124, and P137 are read-only.

Note 2. P137: Undefined
P130: 0 (output latch)

Note 3. Control bits for internal pins. After reset release, be sure to set 0 to port mode register.

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

5.3.3 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 5 - 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	PU04	PU03	PU02	0	0	F0030H	00H	R/W
PU1	0	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU7	PU77	PU76	PU75	0	0	PU72	PU71	PU70	F0037H	00H	R/W
PU8	0	0	0	0	0	PU82	PU81	PU80	F0038H	00H	R/W
PU10	0	0	0	0	0	0	0	PU100	F003AH	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	0	0	0	PU144	PU143	PU142	PU141	PU140	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 8, 10 to 12, 14; n = 0 to 7)
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.

5.3.4 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 5 - 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	PIM04	PIM03	0	0	0	F0040H	00H	R/W
PIM8	0	0	0	0	0	0	PIM81	PIM80	F0048H	00H	R/W
PIM14	0	0	0	0	PIM143	PIM142	0	0	F004EH	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 1, 3 to 5, 8, 14; n = 0 to 7)
0	Normal input buffer
1	TTL input buffer

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

5.3.5 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.

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 (POMmn = 1) is set.

Figure 5 - 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	POM04	POM03	POM02	0	0	F0050H	00H	R/W
POM7	0	0	0	0	0	0	POM71	0	F0057H	00H	R/W
POM8	0	0	0	0	0	POM82	POM81	POM80	F0058H	00H	R/W
POM14	0	0	0	POM144	POM143	POM142	0	0	F005EH	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 1, 3 to 5, 7, 8, 14; n = 0 to 5, 7)
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.

5.3.6 Port mode control registers (PMCxx)

These registers set the P02, P03, P100, P120, and P147 digital I/O/analog input in 1-bit units.

PMC0, PMC10, PMC12, and PMC14 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to FFH.

Figure 5 - 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	PMC03	PMC02	1	1	F0060H	FFH	R/W
PMC10	1	1	1	1	1	1	1	PMC100	F006AH	FFH	R/W
PMC12	1	1	1	1	1	1	1	PMC120	F006CH	FFH	R/W
PMC14	PMC147	1	1	1	1	1	1	1	F006EH	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 0, 10, 12, 14; n = 0, 2, 3, 7)
0	Digital I/O (alternate function other than analog input)
1	PMC120 is an analog input. The digital I/O of PMC02, PMC03,PMC100, and PMC147 is prohibited.

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

5.3.7 A/D port configuration register (ADPC)

This register is used to switch the P20/ANI0, P21/ANI1, P22/ANI2, ANI13/P155, and ANI14/P156 pins to digital I/O of port or analog function of A/D converter.

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

Reset signal generation sets this register to 00H.

Figure 5 - 7 Format of A/D port configuration register (ADPC)

Address: F0076H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	0	ADPC3	ADPC2	ADPC1	ADPC0

ADPC3	ADPC2	ADPC1	ADPC0	Analog function (A)/digital I/O (D) switching				
				ANI14/P156	ANI13/P155	ANI2/P22	ANI1/P21	ANI0/P20
0	0	0	0	A	A	A	A	A
0	0	0	1	D	D	D	D	D
0	0	1	0	D	D	D	D	A
0	0	1	1	D	D	D	A	A
0	1	0	0	D	D	A	A	A
1	1	1	1	D	A	A	A	A

Caution 1. Set the port to analog input by ADPC register to the input mode by using port mode registers 2, 15 (PM2, PM15).

Caution 2. Do not set the pin set by the ADPC register as digital I/O by the analog input channel specification register (ADS).

Caution 3. When using AV_{REFP} and AV_{REFM}, set ANI0 and ANI1 to analog input and set the port mode register to the input mode.

5.4 Port Function Operations

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

5.4.1 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.

5.4.2 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.

5.4.3 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.

5.4.4 Handling different potential (1.8 V, 2.5 V, 3 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, or 3 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, or 3 V), set the port input mode registers 0, 8, and 14 (PIM0, PIM8, and PIM14) 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, or 3 V), set the port output mode registers 0, 7, 8, and 14 (POM0, POM7, POM8, and POM14) 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 UART1, UART3, CSI10, and CSI30 functions for the TTL input buffer

In case of UART1:	P03
In case of UART3:	P143
In case of CSI10:	P03, P04
In case of CSI30:	P142, P143

<1> Using an external resistor, pull up 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 and PIM14 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 UART1, UART3, CSI10, and CSI30 functions in N-ch open-drain output mode

In case of UART1:	P02
In case of UART3:	P144
In case of CSI10:	P02, P04
In case of CSI30:	P142, P144

<1> Using an external resistor, pull up 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 and POM14 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 and PM14 registers to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

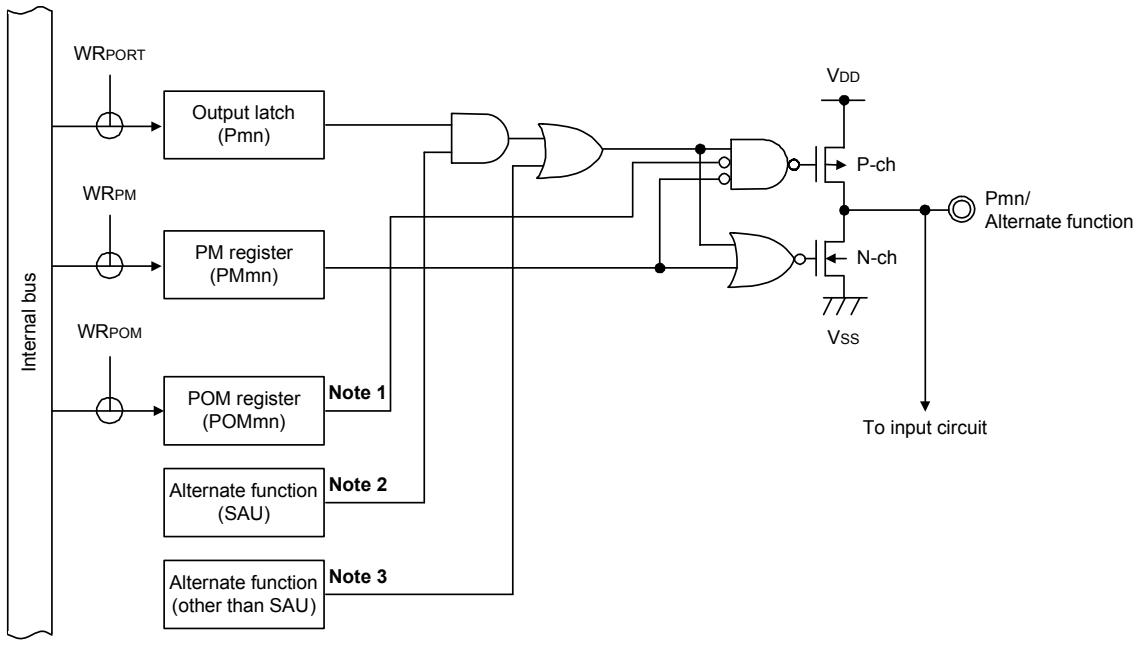
5.5 Register Settings When Using Alternate Function

5.5.1 Basic concept when using alternate function

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

Figure 5 - 8 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, RTC, clock/buzzer output, IICA, 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 5 - 6.

Figure 5 - 8 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 5 - 6 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 **5.5.2 Register settings for alternate function whose output function is not used**.

5.5.2 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) PCLBUZn = 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.

5.5.3 Register setting examples for used port and alternate functions

Register setting examples for used port and alternate functions are shown in Tables 5 - 7 to 5 - 12. The registers used to control the port functions should be set as shown in Tables 5 - 7 to 5 - 12. See the following remark for legends used in Tables 5 - 7 to 5 - 12.

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 5 - 7 Setting Examples of Registers When Using P02 to P16 Pin Function

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O					SAU Output Function	Other than SAU
P02	P02	Input	×	0	1	×	×	—
		Output	0	0	0	0/1	TxD1/SO10 = 1	—
		N-ch OD output	1	0	0	0/1		
	TxD1	Output	0/1	0	0	1	—	—
	SO10	Output	0/1	0	0	1	—	—
P03	P03	Input	×	0	1	×	×	—
		Output	0	0	0	0/1	—	—
		N-ch OD output	1	0	0	0/1		
	SI10	Input	×	0	1	×	×	—
	RxD1	Input	×	0	1	×	×	—
P04	P04	Input	×	—	1	×	—	—
		Output	0	—	0	0/1	SCK10 = 1	—
		N-ch OD output	1	—	0	0/1		
	SCK10	Input	×	—	1	×	—	—
		Output	0/1	—	0	1	—	—
P10	P10	Input	—	—	1	×	—	
		Output	—	—	0	0/1	—	—
P11	P11	Input	—	—	1	×	—	
		Output	—	—	0	0/1	—	—
P12	P12	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	—
P13	P13	Input	×	—	1	×	—	—
		Output	0	—	0	0/1	SO20 = 1	—
	SO20	Output	0/1	—	0	1	—	—
P14	P14	Input	×	—	1	×	—	—
		Output	0	—	0	0/1	—	—
	SI20	Input	×	—	1	×	—	—
P15	P15	Input	×	—	1	×	—	—
		Output	0	—	0	0/1	SCK20 = 1	—
	SCK20	Input	×	—	1	×	—	—
		Output	0/1	—	0	1	—	—
P16	P16	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	—

Table 5 - 8 Setting Examples of Registers When Using P20 to P22 Pin Function

Pin Name	Used Function		ADPC	ADM2	PMxx	Pxx
	Function Name	I/O				
P20	P20	Input	ADPC = 01H	×	1	×
		Output	ADPC = 01H	×	0	0/1
	ANI0	Analog input	ADPC = 00H/02H to 0FH	00x0xx0x, 10x0xx0x	1	×
	AVREFP	Reference voltage input	ADPC = 00H/02H to 0FH	01x0xx0x	1	×
P21	P21	Input	ADPC = 01H/02H	×	1	×
		Output	ADPC = 01H/02H	×	0	0/1
	ANI1	Analog input	ADPC = 00H/03H to 0FH	xx00xx0x	1	×
	AVREFM	Reference voltage input	ADPC = 00H/03H to 0FH	xx10xx0x	1	×
P22	P22	Input	ADPC = 01H to 03H	×	1	×
		Output	ADPC = 01H to 03H	×	0	0/1
	ANI2	Analog input	ADPC = 00H/04H to 0FH	×	1	×

Table 5 - 9 Setting Examples of Registers When Using P30 to P120 Pin Function (1/2)

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O					SAU Output Function	Other than SAU
P30	P30	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	—
	INTP3	Input	—	—	1	×	—	—
P31	P31	Input	—	—	0	0/1	—	—
		Output	—	—	1	×	—	TO03 = 0
	TI03	Input	—	—	1	×	—	—
	TO03	Output	—	—	0	0	—	—
	INTP4	Input	—	—	1	×	—	—
P40	P40	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	—
P60	P60	Input	—	—	1	×	—	—
		N-ch OD output (6 V tolerance)	—	—	0	0/1	—	SCLA0 = 0
	SCLA0	I/O	—	—	0	0	—	—
P61	P61	Input	—	—	1	×	—	—
		N-ch OD output (6 V tolerance)	—	—	0	0/1	—	SDAA0 = 0
	SDAA0	I/O	—	—	0	0	—	—
P62	P62	Input	—	—	1	×	—	—
		N-ch OD output (6 V tolerance)	—	—	0	0/1	—	—
	SCLA1	I/O	—	—	0	0	—	—
P63	P63	Input	—	—	1	×	—	—
		N-ch OD output (6 V tolerance)	—	—	0	0/1	—	SDAA1 = 0
	SDAA1	I/O	—	—	0	0	—	—
P70	P70	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	SCK21 = 1	—
	SCK21	Input	—	—	1	×	—	—
		Output	—	—	0	1	—	—
P71	P71	Input	×	—	1	×		—
		Output	0	—	0	0/1	—	—
		N-ch OD output	1	—	0	0/1		
	SI21	Input	×	—	1	×	—	—
P72	P72	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	SO21 = 1	—
	SO21	Output	—	—	0	1	—	—
P75	P75	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	—
	INTP9	Input	—	—	1	×	—	—
P76	P76	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	—
	INTP10	Input	—	—	1	×	—	—
P77	P77	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	—
	INTP11	Input	—	—	1	×	—	—
P80	P80	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	—
		N-ch OD output	—	—	0	0/1		
P81	P81	Input	0	—	1	×	—	—
		Output	0	—	0	0/1	—	—
		N-ch OD output	1	—	0	0/1		

Table 5 - 9 Setting Examples of Registers When Using P30 to P120 Pin Function (2/2)

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O					SAU Output Function	Other than SAU
P82	P82	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	—
		N-ch OD output	—	—	0	0/1		
P100	P100	Input	—	0	1	×	—	—
		Output	—	0	0	0/1	—	—
P120	P120	Input	—	0	1	×	—	—
		Output	—	0	0	0/1	—	—
	ANI19	Analog input	—	1	1	×	—	—

Table 5 - 10 Setting Examples of Registers When Using P121 to P124 Pin Function

Pin Name	Used Function		CMC	Pxx
	Function Name	I/O	(EXCLK, OSCSEL, EXCLKS, OSCSELS)	
P121	P121	Input	00xx/10xx/11xx	×
	X1	—	01xx	—
P122	P122	Input	00xx/10xx/11xx	×
	X2	—	01xx	—
	EXCLK	Input	11xx	—
P123	P123	Input	xx00/xx10/xx11	×
	XT1	—	xx01	—
P124	P124	Input	xx00/xx10/xx11	×
	XT2	—	xx01	—
	EXCLKS	Input	xx11	—

Table 5 - 11 Setting Examples of Registers When Using P130 to P144 Pin Function

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output	
	Function Name	I/O					SAU Output Function	Other than SAU
P130	P130	Output	—	—	—	0/1	—	—
P137	P137	Input	—	—	—	×	—	—
	INTP0	Input	—	—	—	×	—	—
P140	P140	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	PCLBUZ0 = 0
	PCLBUZ0	Output	—	—	0	0	—	—
	INTP6	Input	—	—	1	×	—	—
P141	P141	Input	—	—	1	×	—	—
		Output	—	—	0	0/1	—	PCLBUZ1 = 0
	PCLBUZ1	Output	—	—	0	0	—	—
	INTP7	Input	—	—	1	×	—	—
P142	P142	Input	×	—	1	×	—	—
		Output	0	—	0	0/1	SCK30 = 1	—
		N-ch OD output	1	—	0	0/1		
	SCK30	Input	×	—	1	×	—	—
<R>		Output	0/1	—	0	1	—	—
P143	P143	Input	×	—	1	×		—
		Output	0	—	0	0/1	—	—
		N-ch OD output	1	—	0	0/1		
	SI30	Input	×	—	1	×	—	—
	RxD3	Input	×	—	1	×	—	—
P144	P144	Input	×	—	1	×	—	—
		Output	0	—	0	0/1	TxD3/ SO30 = 1	—
		N-ch OD output	1	—	0	0/1		
	SO30	Output	0/1	—	0	1	—	—
	TxD3	Output	0/1	—	0	1	—	—

Table 5 - 12 Setting Examples of Registers When Using P155 and P156 Pin Function

Pin Name	Used Function		ADPC	PMxx	Pxx
	Function Name	I/O			
P155	P155	Input	ADPC = 01H to 0EH	1	×
		Output	ADPC = 01H to 0EH	0	0/1
	ANI13	Analog input	ADPC = 00H/0FH	1	×
P156	P156	Input	ADPC = 01H to 0FH	1	×
		Output	ADPC = 01H to 0FH	0	0/1
	ANI14	Analog input	ADPC = 00H	1	×

5.6 Cautions When Using Port Function

5.6.1 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.

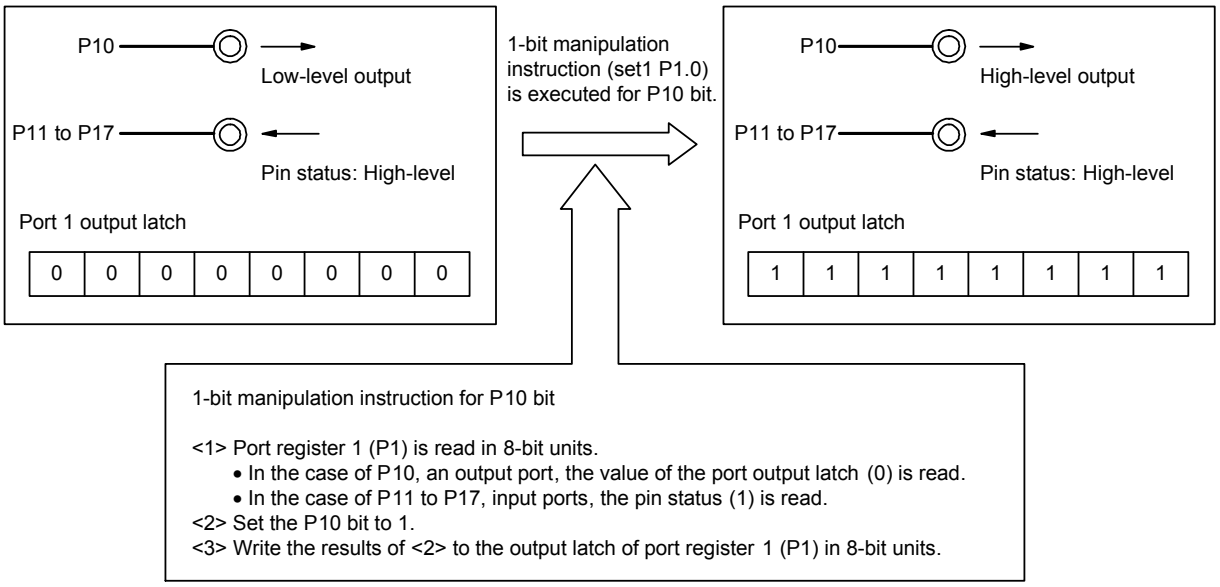
<Example> When P10 is an output port, P11 to P17 are input ports (all pin statuses are high level), and the port latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 is FFH.

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 RL78/G1H.

<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 P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 are high level at this time, the read value is FEH.
The value is changed to FFH by the manipulation in <2>.
FFH is written to the output latch by the manipulation in <3>.

Figure 5 - 9 Bit Manipulation Instruction (P10)



5.6.2 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 5.5 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 6 CLOCK GENERATOR

For details about the clock generator for the RF transceiver, see **2.4 Base Operation Clock of RF Unit**.

6.1 Functions of Clock Generator

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

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of f_X = 1 to 20 MHz by connecting a resonator to X1 pin and X2 pin. 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 (High-speed OCO)

The frequency at which to oscillate can be selected from among f_{IH} = 32 MHz/24 MHz/16 MHz/12 MHz/8 MHz/ 6 MHz/4 MHz/3 MHz/2 MHz/1 MHz (TYP.) by using the option byte (000C2H). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing the STOP instruction or setting of 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 6 - 12 Format of High-speed on-chip oscillator frequency select register (HOCODIV)**.

An external main system clock (f_{EX} = 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 can be selected by setting of the MCM0 bit (bit 4 of the system clock control register (CKC)).

(2) Subsystem clock

• XT1 clock oscillator

This circuit oscillates a clock of f_{XT} = 32.768 kHz by connecting a 32.768 kHz resonator to XT1 pin and XT2 pin. Oscillation can be stopped by setting the XTSTOP bit (bit 6 of the clock operation status control register (CSC)).

An external subsystem clock (f_{EXS} = 32.768 kHz) can also be supplied from the EXCLKS/XT2/P124 pin. An external subsystem clock input can be disabled by the setting of the XTSTOP bit.

(3) Low-speed on-chip oscillator (Low-speed OCO)

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
- Real-time clock
- 12-bit interval timer
- Timer RJ

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

Caution The low-speed on-chip oscillator clock (f_{IL}) can only be selected as the real-time clock count clock when the fixed-cycle interrupt function is used.

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{EX}: External main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{IL}: Low-speed on-chip oscillator frequency

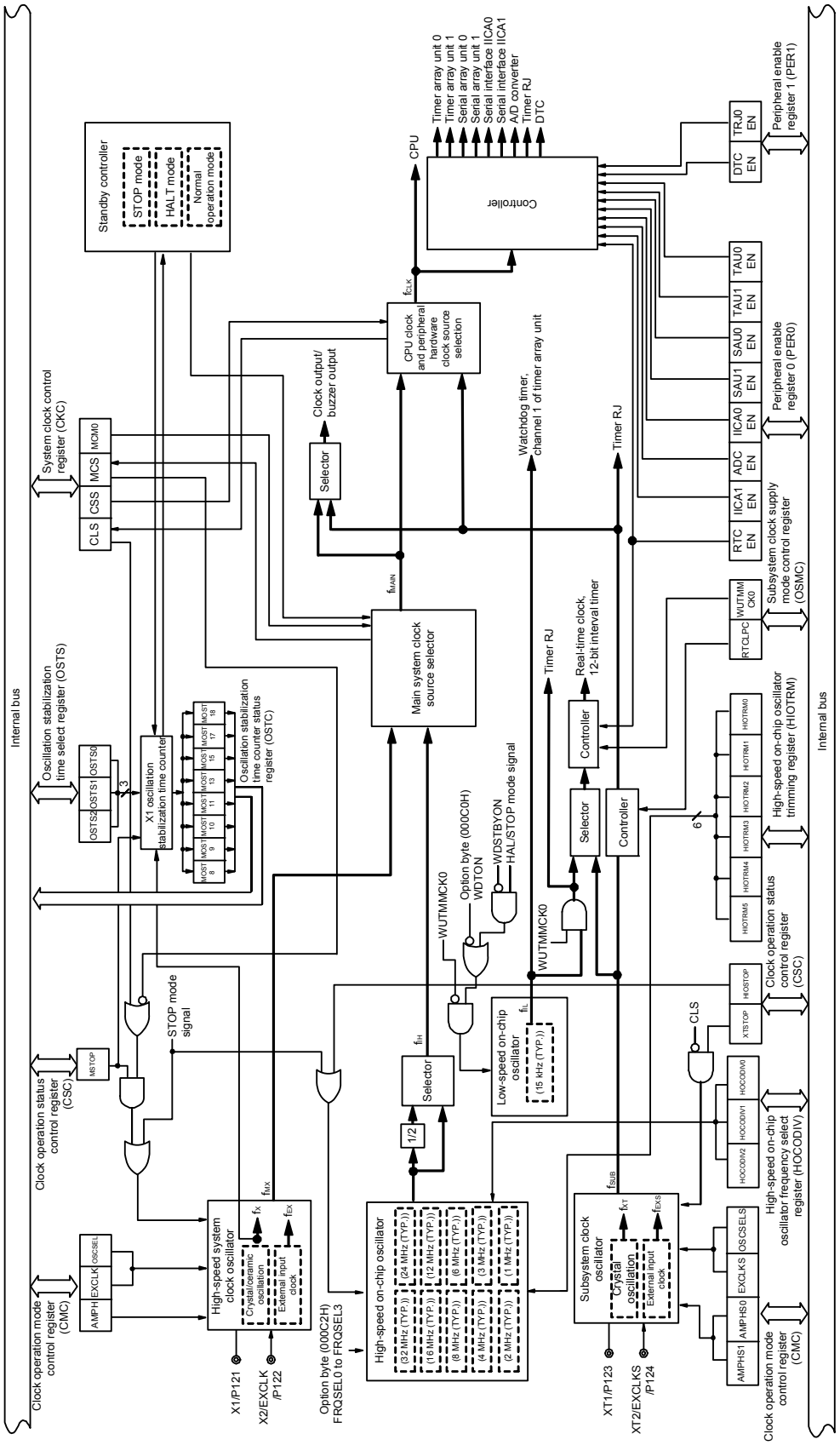
6.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 6 - 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, 1 (PER0, PER1) Subsystem clock supply mode control register (OSMC) High-speed on-chip oscillator frequency select register (HOCODIV) High-speed on-chip oscillator trimming register (HIOTRM)
Oscillators	X1 oscillator XT1 oscillator High-speed on-chip oscillator Low-speed on-chip oscillator

Figure 6 - 1 Block Diagram of Clock Generator



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

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

6.3 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, 1 (PER0, PER1)
- Subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)

6.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121, X2/EXCLK/P122, XT1/P123, and XT2/EXCLKS/P124 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 6 - 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	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	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

EXCLKS	OSCSELS	Subsystem clock pin operation mode	XT1/P123 pin	XT2/EXCLKS/P124 pin
0	0	Input port mode	Input port	
0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

AMPHS1	AMPHS0	XT1 oscillator oscillation mode selection
0	0	Low power consumption oscillation (default)
0	1	Normal oscillation
1	0	Ultra-low power consumption oscillation
1	1	Setting prohibited

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 or XT1 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, AMPHS1, and AMPHS0 bits while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX} or f_{SUB}).

Caution 5. Oscillation stabilization time of f_{XT}, counting on the software.

Caution 6. Although the maximum system clock frequency is 32 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

(Cautions and Remark are given on the next page.)

Caution 7. The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **When using the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little parasitic capacitance and wiring resistance.**
- **Place a ground pattern that has the same potential as Vss as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

Remark fx: X1 clock frequency

6.3.2 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 6 - 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	CLS	CSS	MCS	MCM0	0	0	0	0

CLS	Status of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

CSS[Note 2]	Selection of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

MCS	Status of Main system clock (f_{MAIN})
0	High-speed on-chip oscillator clock (f_{IH})
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_{IH}) 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. Bits 7 and 5 are read-only.

Note 2. Changing the value of the MCM0 bit is prohibited while the CSS bit is set to 1.

Remark f_{IH}:High-speed on-chip oscillator clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency

Caution 1. Be sure to set bits 0 to 3 of the CKC register to 0.

Caution 2. The clock set by the CSS bit is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the real-time clock, 12-bit interval timer, clock output/buzzer output, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.

Caution 3. If the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 31 ELECTRICAL SPECIFICATIONS.

6.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock, high-speed on-chip oscillator clock, and subsystem clock (except the low-speed on-chip oscillator clock).

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 6 - 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	XTSTOP	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	

XTSTOP	Subsystem clock operation control		
	XT1 oscillation mode	External clock input mode	Input port mode
0	XT1 oscillator operating	External clock from EXCLKS pin is valid	Input port
1	XT1 oscillator stopped	External clock from EXCLKS 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. When starting XT1 oscillation by setting the XSTOP bit, wait for oscillation of the subsystem clock to stabilize by setting a wait time using software.

Caution 5. Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the OSC register.

Caution 6. 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 6 - 2. Check the condition before stopping clock before stopping the clock.

Table 6 - 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. (CLS = 0 and MCS = 0, or CLS = 1)	MSTOP = 1
XT1 clock External subsystem clock	CPU and peripheral hardware clocks operate with a clock other than the subsystem clock. (CLS = 0)	XTSTOP = 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. (CLS = 0 and MCS = 1, or CLS = 1)	HIOSTOP = 1

6.3.4 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,

- When the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.
- When 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.
The generation of reset signal, 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 6 - 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

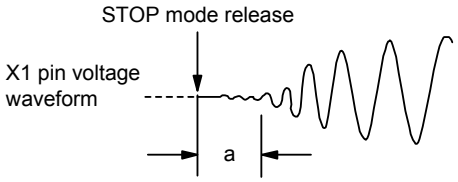
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	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.

- **When the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.**
- **When 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

6.3.5 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 or the subsystem 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 6 - 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	13.1 ms	6.55 ms
1	1	1	2^{18}/fx	26.2 ms	13.1 ms

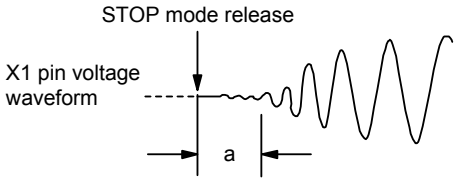
Caution 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

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.

- **When the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.**
- **When 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

6.3.6 Peripheral enable registers 0, 1 (PER0, PER1)

These registers are 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 these registers, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- Real-time clock and 12-bit interval timer
- Serial interface IICA1
- A/D converter
- Serial interface IICA0
- Serial array unit 1
- Serial array unit 0
- Timer array unit 1
- Timer array unit 0
- DTC
- Timer RJ

The PER0 and PER1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to 00H.

Figure 6 - 7 Format of Peripheral enable register 0 (PER0) (1/3)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of supplying input clock for real-time clock (RTC) and 12-bit interval timer
0	Stops input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. • The real-time clock (RTC) and 12-bit interval timer are in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer can be read and written.

Figure 6 - 8 Format of Peripheral enable register 0 (PER0) (2/3)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

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.

SAU1EN	Control of serial array unit 1 input clock supply
0	Stops input clock supply. • SFR used by the serial array unit 1 cannot be written. • The serial array unit 1 is in the reset status.
1	Enables input clock supply. • SFR used by the serial array unit 1 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.

Figure 6 - 9 Format of Peripheral enable register 0 (PER0) (3/3)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

TAU1EN	Control of timer array unit 1 input clock supply
0	Stops input clock supply. • SFR used by timer array unit 1 cannot be written. • Timer array unit 1 is in the reset status.
1	Enables input clock supply. • SFR used by timer array unit 1 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.

Figure 6 - 10 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	4	<3>	2	1	<0>
PER1	0	0	0	0	DTCEN	0	0	TRJ0EN

DTCEN	Control of DTC input clock supply
0	Stops input clock supply. • DTC cannot run.
1	Enables input clock supply. • DTC can run.

TRJ0EN	Control of timer RJ0 input clock supply
0	Stops input clock supply. • SFR used by timer RJ0 cannot be written. • Timer RJ0 is in the reset status.
1	Enables input clock supply. • SFR used by timer RJ0 can be read and written.

6.3.7 Subsystem clock supply mode control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the real-time clock and 12-bit interval timer, is stopped in STOP mode or HALT mode while subsystem clock is selected as CPU clock.

In addition, the OSMC register can be used to select the operation clock of the real-time clock and 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 6 - 11 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	RTCLPC	0	0	WUTMMCK0	0	0	0	0

RTCLPC	Setting in STOP mode or HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions (See **Tables 20 - 1** to **20 - 4** for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than real-time clock and 12-bit interval timer.

WUTMMCK0	Selection of operation clock for real-time clock, 12-bit interval timer, and timer RJ
0	• The subsystem clock is selected as the operation clock for the real-time clock and the 12-bit interval timer. The low-speed on-chip oscillator cannot be selected as the count source for timer RJ.
1	• The low-speed on-chip oscillator clock is selected as the operation clock for the real-time clock and the 12-bit interval timer. • Either the low-speed on-chip oscillator or the subsystem clock can be selected as the count source for timer RJ.

6.3.8 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).

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 6 - 12 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	
			FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	f_{IH} = 32 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz
1	0	0	Setting prohibited	f_{IH} = 2 MHz
1	0	1	Setting prohibited	f_{IH} = 1 MHz
Other than above			Setting prohibited	

Caution 1. 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.

Value of option byte (000C2H) CMODE0	Flash operation mode	Operating frequency range	Operating voltage range
0	LS (low-speed main) mode	1 MHz to 8 MHz	1.8 V to 3.6 V
1	HS (high-speed main) mode	1 MHz to 16 MHz	2.4 V to 3.6 V
		1 MHz to 32 MHz	2.7 V to 3.6 V

Caution 2. Set the HOCODIV register with the high-speed on-chip oscillator clock (f_{IH}) selected as the CPU/peripheral hardware clock (f_{CLK}).

Caution 3. 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**

6.3.9 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, 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 6 - 13 Format of High-speed on-chip oscillator trimming register (HIOTRM)

<R> 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 reset is the value adjusted at shipment.

Remark 1. The HIOTRM register can be used to adjust the high-speed on-chip oscillator clock to an accuracy within about 0.05% on 1 bit per.

Remark 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).

6.4 System Clock Oscillator

6.4.1 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

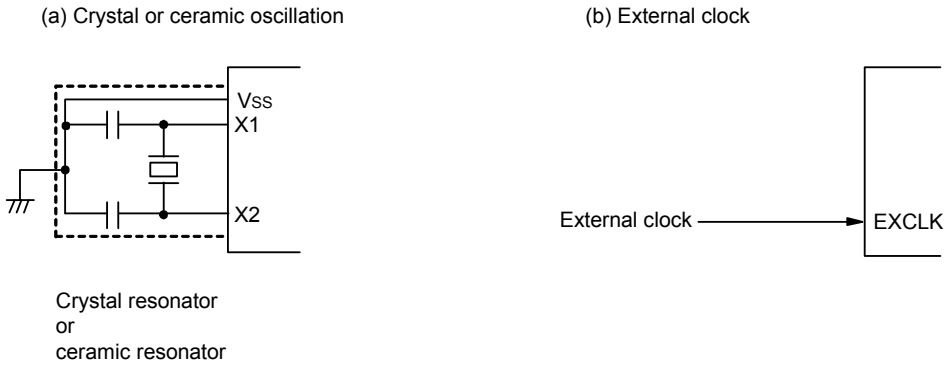
• External clock input: EXCLK, OSCSEL = 1, 1

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 3 - 3 Connection of Unused Pins**.

Figure 6 - 14 shows an example of the external circuit of the X1 oscillator.

Figure 6 - 14 Example of External Circuit of X1 Oscillator



Cautions are listed on the next page.

6.4.2 XT1 oscillator

The XT1 oscillator oscillates with a crystal resonator (32.768 kHz (TYP.)) connected to the XT1 and XT2 pins.

To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

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

To use the XT1 oscillator, set bits 5 and 4 (EXCLKS, OSCSELS) of the clock operation mode control register (CMC) as follows.

• Crystal oscillation:EXCLKS, OSCSELS = 0, 1

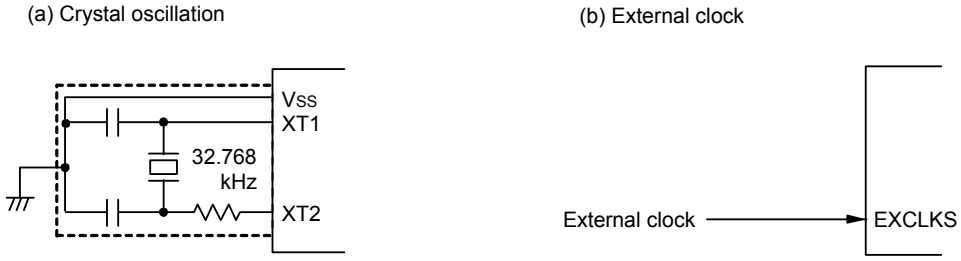
• External clock input:EXCLKS, OSCSELS = 1, 1

When the XT1 oscillator is not used, set the input port mode (EXCLKS, OSCSELS = 0, 0).

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

Figure 6 - 15 shows an example of the external circuit of the XT1 oscillator.

Figure 6 - 15 Example of External Circuit of XT1 Oscillator



Caution 1. When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 6 - 14 and 6 - 15 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.**

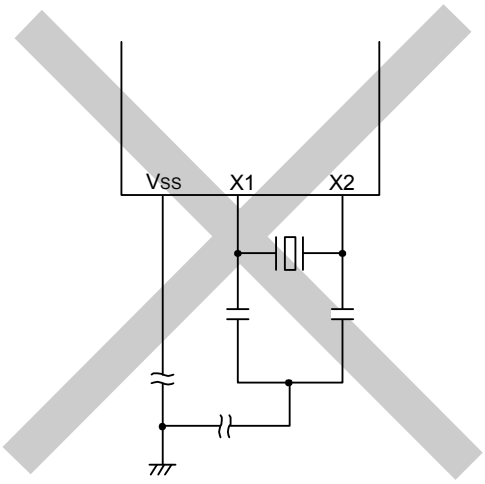
The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **When using the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as Vss as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

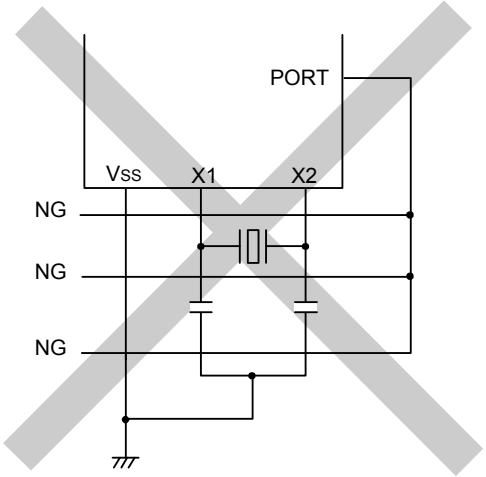
Figure 6 - 16 shows examples of incorrect resonator connection.

Figure 6 - 16 Examples of Incorrect Resonator Connection (1/2)

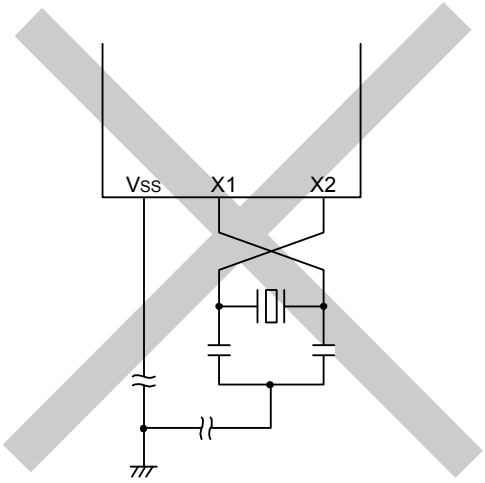
(a) Too long wiring



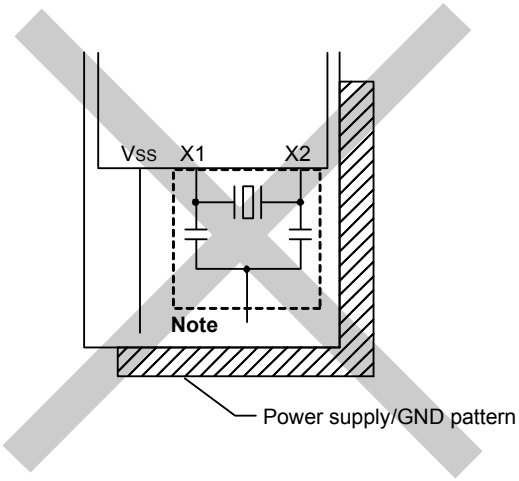
(b) Crossed signal line



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



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

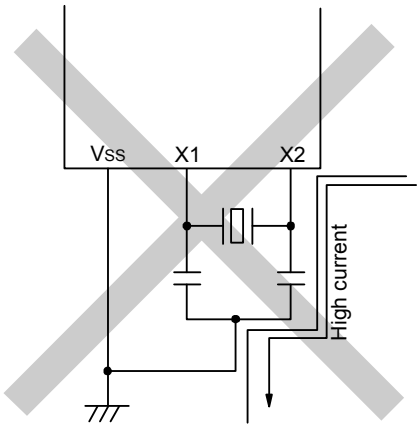


Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

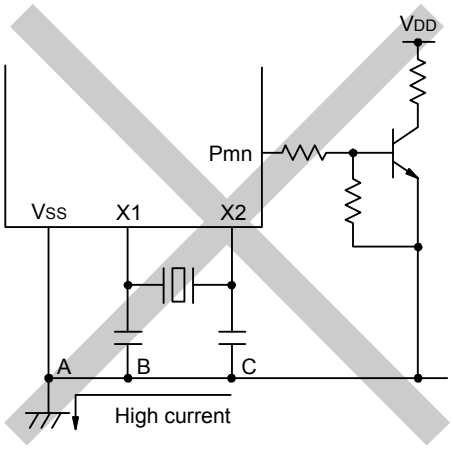
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 6 - 17 Examples of Incorrect Resonator Connection (2/2)

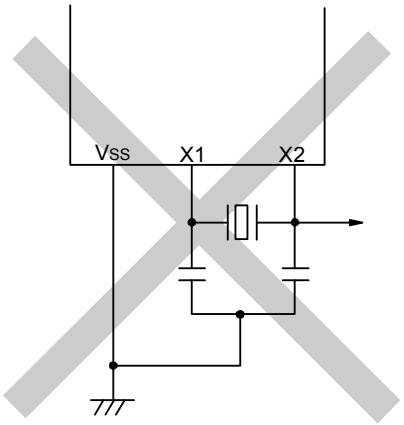
(e) Wiring near high alternating current



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



(g) Signals are fetched



Caution When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.

Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

6.4.3 High-speed on-chip oscillator

The frequency can be selected from among 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using the option byte (000C2H). 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.

6.4.4 Low-speed on-chip oscillator

The low-speed on-chip oscillator clock is used only as the watchdog timer, real-time clock, 12-bit interval timer, and timer RJ clock. The low-speed on-chip oscillator clock cannot be used as the CPU clock.

<R> The low-speed on-chip oscillator runs while the watchdog timer is operating or when bit 4 (WUTMMCK0) in the subsystem clock supply mode control register (OSMC) is set to 1.

The low-speed on-chip oscillator is stopped when the watchdog timer is stopped and WUTMMCK0 is set to 0.

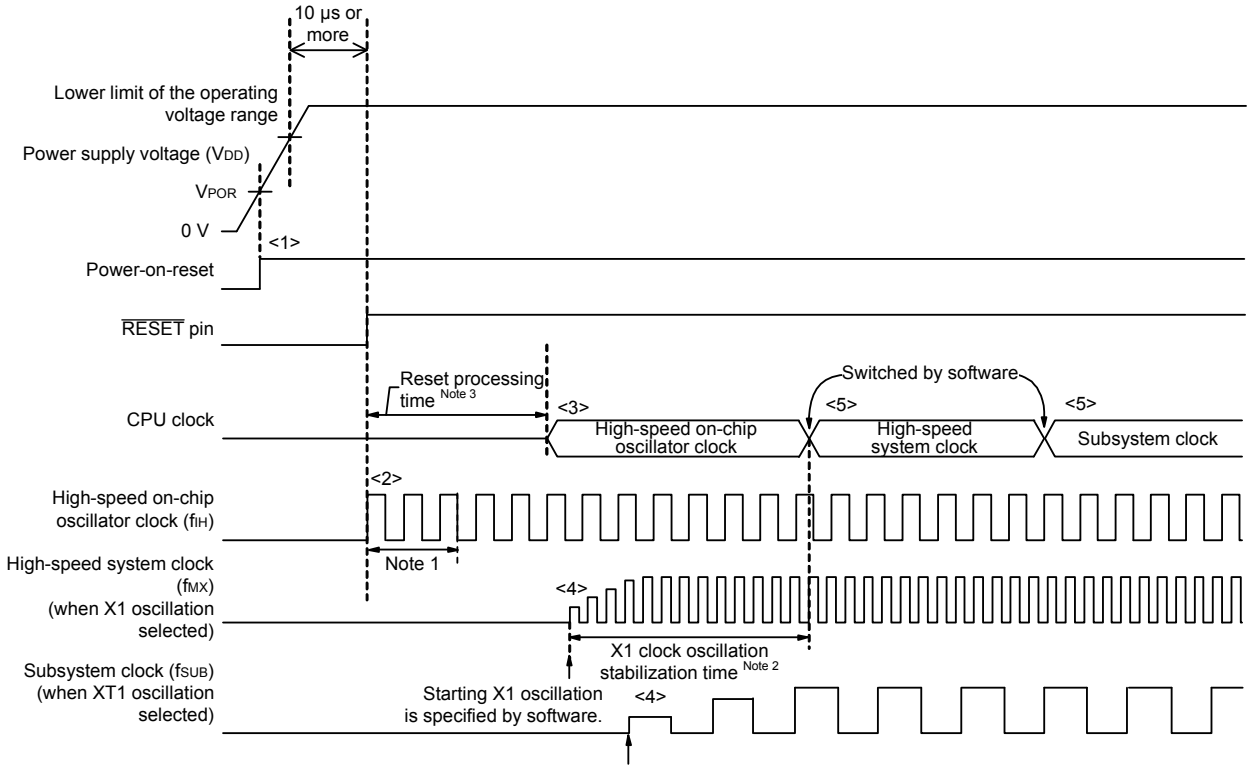
6.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 6 - 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_{IH}
- Subsystem clock f_{SUB}
  - XT1 clock f_{XT}
  - External subsystem clock f_{EXS}
- 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 RL78/G1H.
When the power supply voltage is turned on, the clock generator operation is shown in Figure 6 - 18.

Figure 6 - 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 detection circuit or an external reset until the voltage reaches the range of operating voltage described in **31.4** AC Characteristics (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 or XT1 clock via software (see **6.6.2 Example of setting X1 oscillation clock** and **6.6.3 Example of setting XT1 oscillation clock**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **6.6.2 Example of setting X1 oscillation clock** and **6.6.3 Example of setting XT1 oscillation clock**).

Note 1. The internal 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 21 RESET FUNCTION**.

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

6.6 Controlling Clock

6.6.1 Example of setting high-speed on-chip oscillator

<R> 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 32, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using FRQSEL0 to FRQSEL3 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
	1	CMODE0 0/1	1	0	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

CMODE0	Flash operation mode	Operating frequency range	Operating voltage range
0	LS (low-speed main) mode	1 MHz to 8 MHz	1.8 V to 3.6 V
1	HS (high-speed main) mode	1 MHz to 16 MHz	2.4 V to 3.6 V
		1 MHz to 32 MHz	2.7 V to 3.6 V

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
				f_{IH}
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

[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	
			FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	f_{IH} = 32 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz
1	0	0	Setting prohibited	f_{IH} = 2 MHz
1	0	1	Setting prohibited	f_{IH} = 1 MHz
Other than above			Setting prohibited	

6.6.2 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), 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 the fx is equal to or more than 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	EXCLKS 0	OSCSELS 0	 0	AMPHS1 0	AMPHS0 0	AMPH 0/1

<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	XTSTOP 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 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	CLS 0	CSS 0	MCS 0	MCM0 1	 0	 0	 0	 0

Caution Change the main system clock (f_{MAIN}) by the CKC register within the operable voltage range of the flash operation mode set in the option byte (000C2H) before and after the clock change.

Value of option byte (000C2H) CMODE0	Flash operation mode	Operating frequency range	Operating voltage range
0	LS (low-speed main) mode	1 MHz to 8 MHz	1.8 V to 3.6 V
1	HS (high-speed main) mode	1 MHz to 16 MHz	2.4 V to 3.6 V
		1 MHz to 32 MHz	2.7 V to 3.6 V

6.6.3 Example of setting XT1 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 XT1 oscillation clock, set the oscillator and start oscillation by using the subsystem clock supply mode control register (OSMC), clock operation mode control register (CMC), and clock operation status control register (CSC), set the XT1 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 the RTCLPC bit to 1 to run only the real-time clock, and 12-bit interval timer on the subsystem clock (for ultra-low current consumption) in the STOP mode or HALT mode during CPU operation on the subsystem clock.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC 0/1	 0	 0	WUTMMCK0 0	 0	 0	 0	 0

<2> Set (1) the OSCSELS bit of the CMC register to operate the XT1 oscillator.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	 0	AMPHS1 0/1	AMPHS0 0/1	AMPH 0

AMPHS0 and AMPHS1 bits: These bits are used to specify the oscillation mode of the XT1 oscillator.

<3> Clear (0) the XTSTOP bit of the CSC register to start oscillating the XT1 oscillator.

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

<4> Use the timer function or another function to wait for oscillation of the subsystem clock to stabilize by using software.

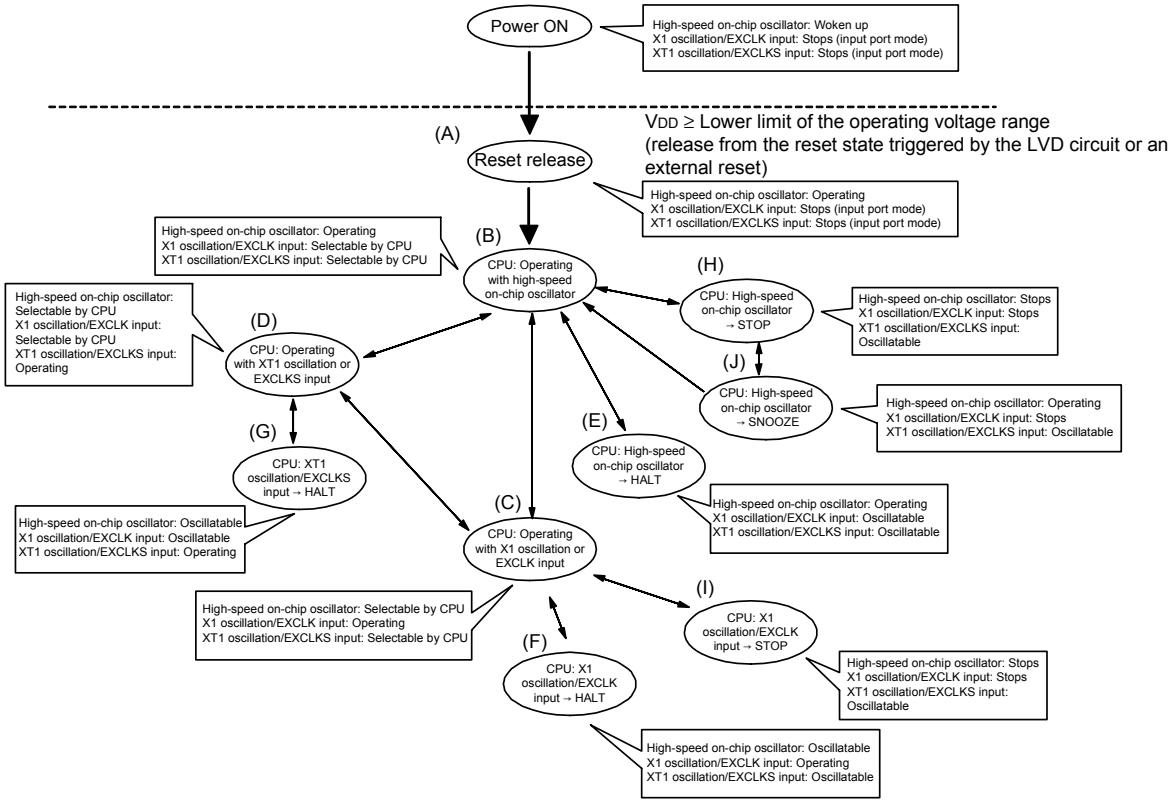
<5> Use the CSS bit of the CKC register to specify the XT1 oscillation clock as the CPU/peripheral hardware clock.

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

6.6.4 CPU clock status transition diagram

Figure 6 - 19 shows the CPU clock status transition diagram of this product.

Figure 6 - 19 CPU Clock Status Transition Diagram



Tables 6 - 3 to 6 - 7 show transition of the CPU clock and examples of setting the SFR registers.

Table 6 - 3 CPU Clock Transition and SFR Register Setting Examples (1/5)

(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 CHAPTER 31 ELECTRICAL SPECIFICATIONS).

(3) CPU operating with subsystem clock (D) 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				CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
(A) → (B) → (C) (XT1 clock)	0	1	0/1	0/1	0	Necessary	1
(A) → (B) → (C) (external sub clock)	1	1	×	×	0	Necessary	1

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

Remark 1. ×: Don't care

Remark 2. (A) to (J) in Tables 6 - 3 to 6 - 7 correspond to (A) to (J) in Figure 6 - 19.

Table 6 - 4 CPU Clock Transition and SFR Register Setting Examples (2/5)

(4) 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) (XT1 clock: 1 MHz ≤ fx ≤ 10 MHz)	0	1	0	**Note 2**	0	Must be checked	1
(B) → (C) (XT1 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 changed only once 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 CHAPTER 31 ELECTRICAL SPECIFICATIONS).

(5) CPU clock changing from high-speed on-chip oscillator clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note			CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS	OSCSELS	AMPHS1,0	XTSTOP		CSS
(B) → (D) (XT1 clock)	0	1	00: Low power consumption oscillation 01: Normal oscillation 10: Ultra-low power consumption oscillation	0	Necessary	1
(B) → (D) (external sub clock)	1	1	×	0	Necessary	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the subsystem clock

Note 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.

Remark 1. ×: Don't care

Remark 2. (A) to (J) in Tables 6 - 3 to 6 - 7 correspond to (A) to (J) in Figure 6 - 19.

Table 6 - 5 CPU Clock Transition and SFR Register Setting Examples (3/5)

(6) 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	**Note**	0

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

<R> **Note** 18 to 65 μs

Remark The oscillation accuracy stabilization time changes according to the temperature conditions and the STOP mode period.

(7) CPU clock changing from high-speed system clock (C) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	XTSTOP		CSS
(C) → (D)	0	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

(8) CPU clock changing from subsystem clock (D) 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	CKC Register
	HIOSTOP		CSS
(D) → (B)	0	**Note**	0

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

<R> **Note** 18 to 65 μs

Remark 1. (A) to (J) in Tables 6 - 3 to 6 - 7 correspond to (A) to (J) in Figure 6 - 19.

Remark 2. The oscillation accuracy stabilization time changes according to the temperature conditions and the STOP mode period.

Table 6 - 6 CPU Clock Transition and SFR Register Setting Examples (4/5)

(9) CPU clock changing from subsystem clock (D) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	OSTS Register	CSC Register	OSTC Register	CKC Register
		MSTOP		CSS
(D) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	**Note**	0	Must be checked	0
(D) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	**Note**	0	Must be checked	0
(D) → (C) (external main clock)	**Note**	0	Need not be checked	0

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

Note 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 CHAPTER 31 ELECTRICAL SPECIFICATIONS).

(10) • 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 (G) set while CPU is operating with subsystem clock (D)

Status Transition	Setting
(B) → (E) (C) → (F) (D) → (G)	Executing HALT instruction

Remark (A) to (J) in Tables 6 - 3 to 6 - 7 correspond to (A) to (J) in Figure 6 - 19.

Table 6 - 7 CPU Clock Transition and SFR Register Setting Examples (5/5)

(11) • STOP mode (H) set while CPU is operating with high-speed on-chip oscillator clock (B)
• 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		—	

(12) 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 **13.8 SNOOZE Mode Function**, **14.5.7 Calculating transfer clock frequency**, and **14.6.3 Calculating baud rate**.

Remark (A) to (J) in Tables 6 - 3 to 6 - 7 correspond to (A) to (J) in Figure 6 - 19.

6.6.5 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 6 - 8 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	
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation 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	—
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	After checking that the CPU clock is switched to the clock after change, X1 oscillation can be stopped (MSTOP = 1).
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	After checking that the CPU clock is switched to the clock after change, external main system clock input can be disabled (MSTOP = 1).
	X1 clock	Transition not possible	—
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	After checking that the CPU clock is switched to the clock after change, external main system clock input can be disabled (MSTOP = 1).
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	

Table 6 - 9 Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
XT1 clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock • HIOSTOP = 0, MCS = 0	After checking that the CPU clock is switched to the clock after change, XT1 oscillation can be stopped (XTSTOP = 1)
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	External subsystem clock	Transition not possible	—
External subsystem clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock • HIOSTOP = 0, MCS = 0	After checking that the CPU clock is switched to the clock after change, external subsystem clock input can be disabled (XTSTOP = 1).
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition not possible	—

6.6.6 Time required for switchover of CPU clock and main system clock

By setting bits 4 and 6 (MCM0, CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), 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 **Tables 6 - 10** to **6 - 12**).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 7 (CLS) of the CKC register. 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 6 - 10 Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	◀⟷▶	f_{MX}	See **Table 6 - 11**
f_{MAIN}	◀⟷▶	f_{SUB}	See **Table 6 - 12**

Table 6 - 11 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 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$	2 f_{MX}/f_{IH} clock	
	$f_{MX} < f_{IH}$	2 clock	

Table 6 - 12 Maximum Number of Clocks Required for $f_{MAIN} \leftrightarrow f_{SUB}$

Set Value Before Switchover	Set Value After Switchover	
CSS	CSS	
	0 ($f_{CLK} = f_{MAIN}$)	1 ($f_{CLK} = f_{SUB}$)
0 ($f_{CLK} = f_{MAIN}$)		1 + 2 f_{MAIN}/f_{SUB} clock
1 ($f_{CLK} = f_{SUB}$)	3 clock	

Remark 1. The number of clocks listed in Tables 6 - 11 and 6 - 12 is the number of CPU clocks before switchover.

Remark 2. Calculate the number of clocks in Tables 6 - 11 and 6 - 12 by rounding up the number after the decimal position.

Example When switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (@ oscillation with f_{IH} = 8 MHz, f_{MX} = 10 MHz)

2 f_{MX}/f_{IH} = 2 (10/8) = 2.5 → 3 clocks

6.6.7 Conditions before clock oscillation is stopped

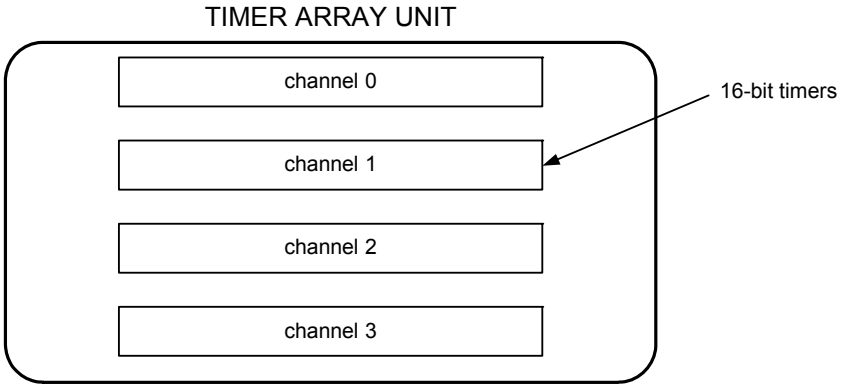
The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped. Check the condition before stopping clock before stopping the clock.

Table 6 - 13 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 or CLS = 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 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
XT1 clock External subsystem clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock.)	XTSTOP = 1

CHAPTER 7 TIMER ARRAY UNIT

Timer array unit incorporates 8 channels of 16-bit timer (4 channels of 16-bit timer × 2 units) in total.
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 **7.8.1**) • Square wave output (→ refer to **7.8.1**) • External event counter (→ refer to **7.8.2**) • Input pulse interval measurement (→ refer to **7.8.3**) • Measurement of high-/low-level width of input signal (→ refer to **7.8.4**) • Delay counter (→ refer to **7.8.5**)	• PWM output (→ refer to **7.9.1**)

It is possible to use the 16-bit timer of channels 1 and 3 of the units 0 and 1 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)

7.1 Functions of Timer Array Unit

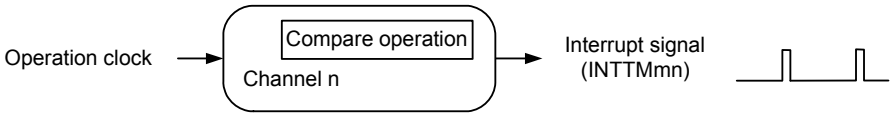
Timer array unit has the following functions.

7.1.1 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.

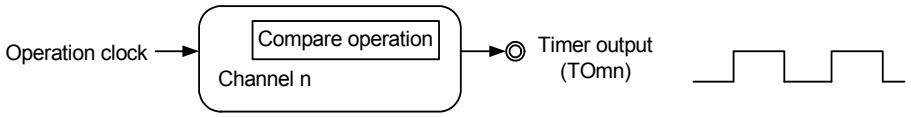
(1) Interval timer

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



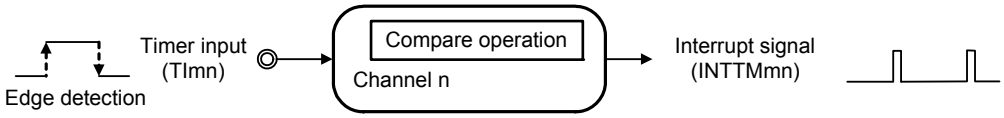
(2) Square wave output

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).



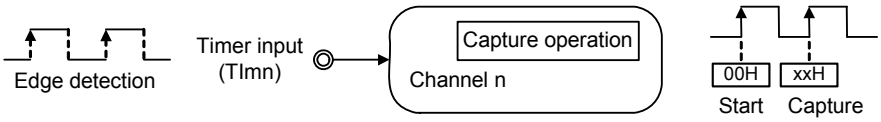
(3) External event counter

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.



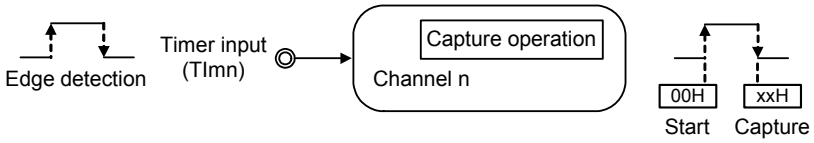
(4) Input pulse interval measurement

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.



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

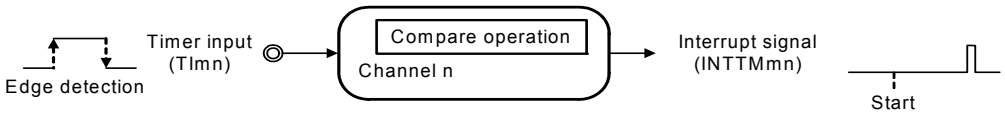
Counting is started by a single edge of the signal input to the timer input pin (TImn), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



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

(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.



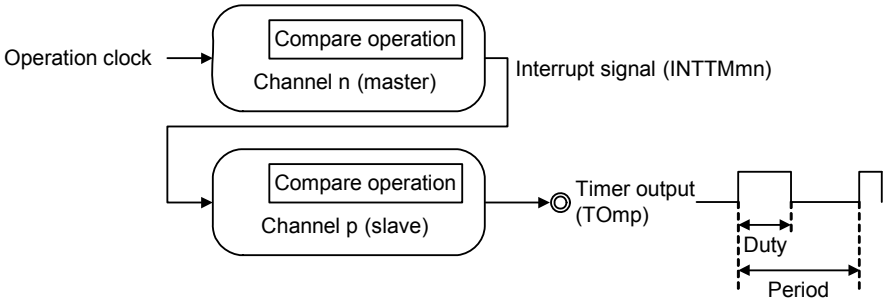
Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

7.1.2 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) 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.



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3),
p, q: Slave channel number ($n < p < q \leq 3$)

7.1.3 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 7.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

7.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 7 - 1 Configuration of Timer Array Unit

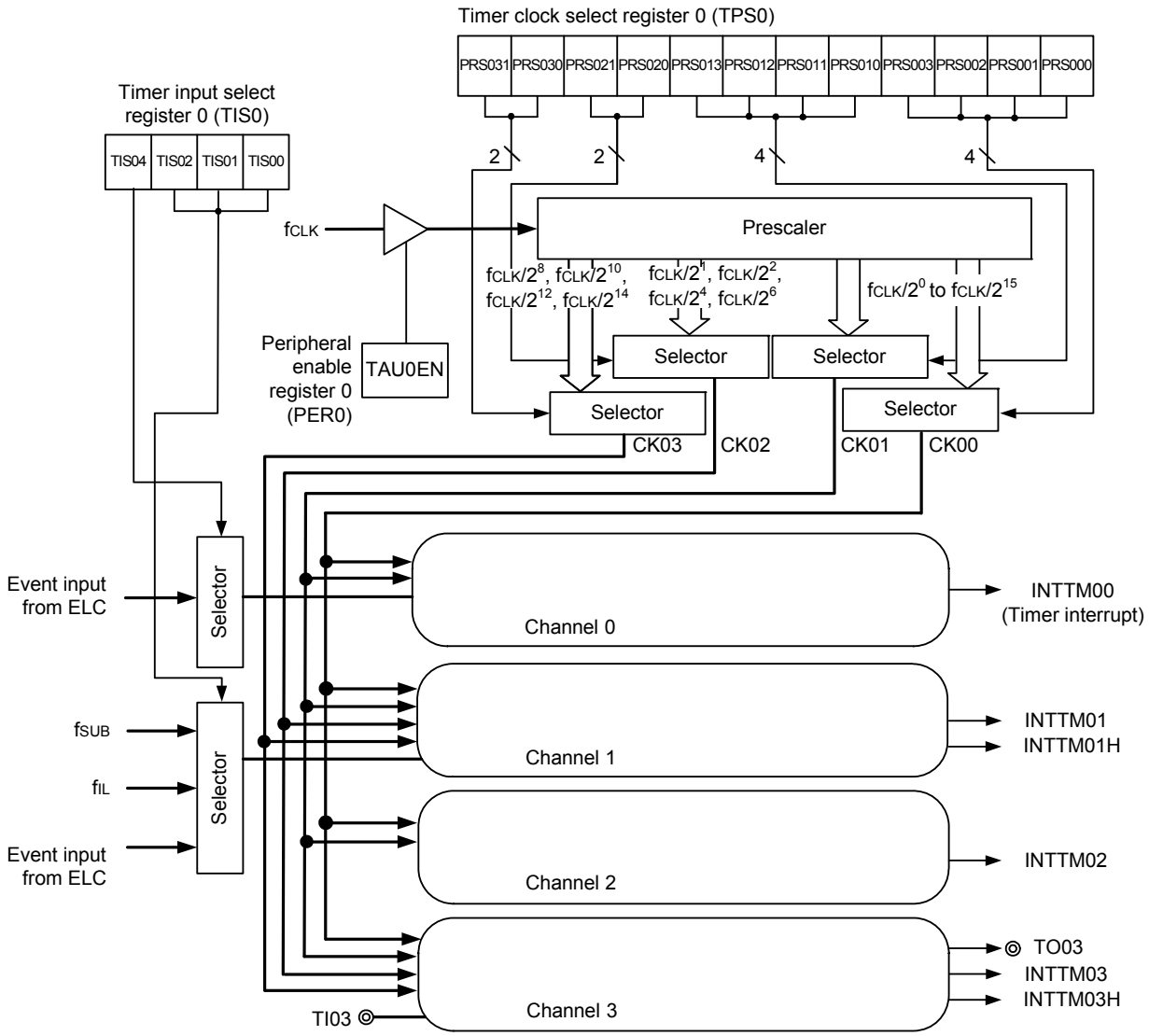
Item	Configuration
Timer/counter	Timer count register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI03
Timer output	TO03
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register 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 register (PMxx) **Note** • Port register (Pxx) **Note**

Note For details, see **5.5 Register Settings When Using Alternate Function**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 7 - 1 and Figure 7 - 2 show the timer array unit block diagram.

Figure 7 - 1 Entire Configuration of Timer Array Unit 0



Remark f_{SUB}: Subsystem clock frequency

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

Figure 7 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit 0

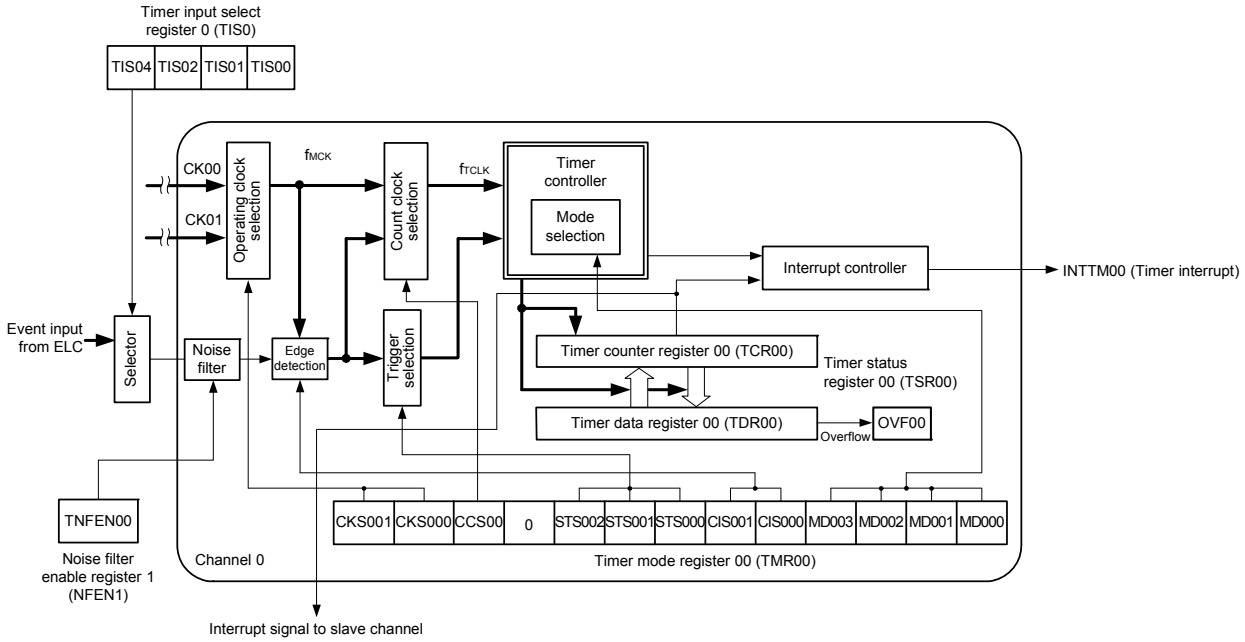


Figure 7 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit 0

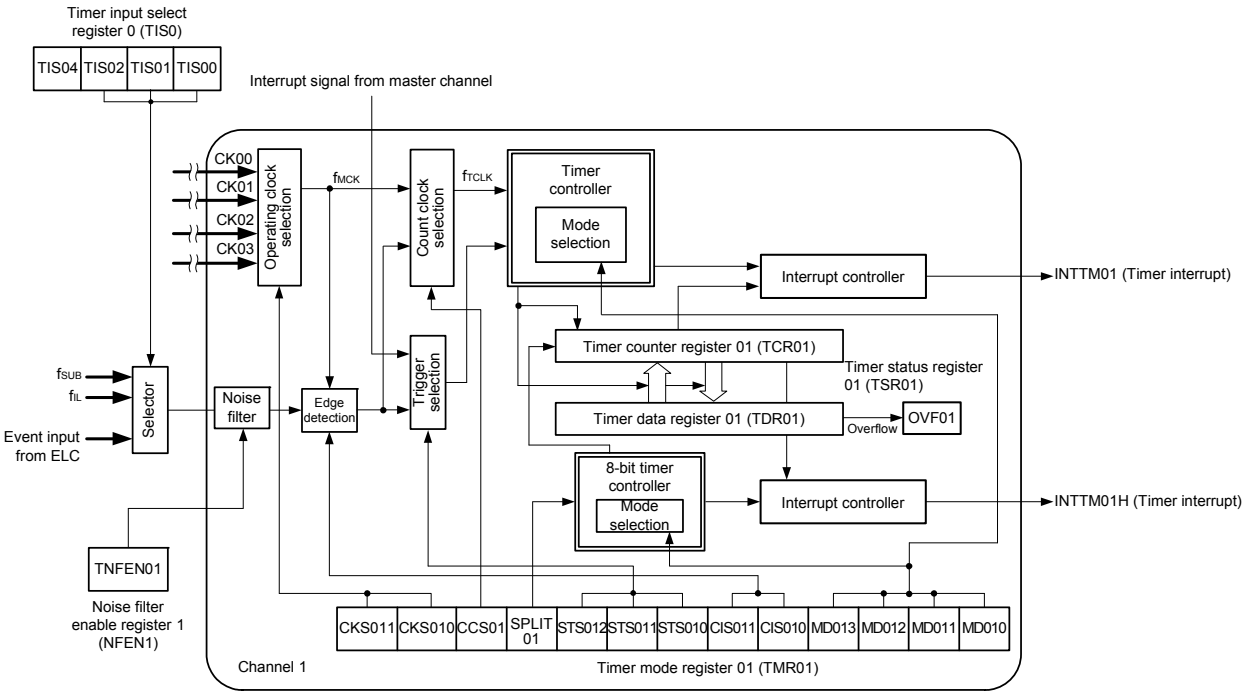


Figure 7 - 4 Internal Block Diagram of Channel 2 of Timer Array Unit 0

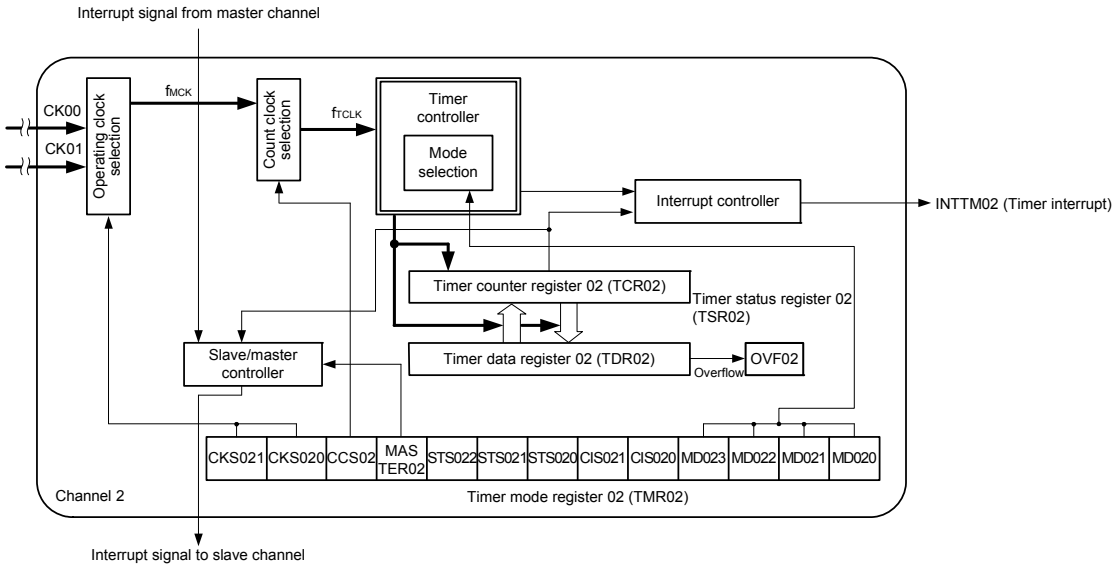
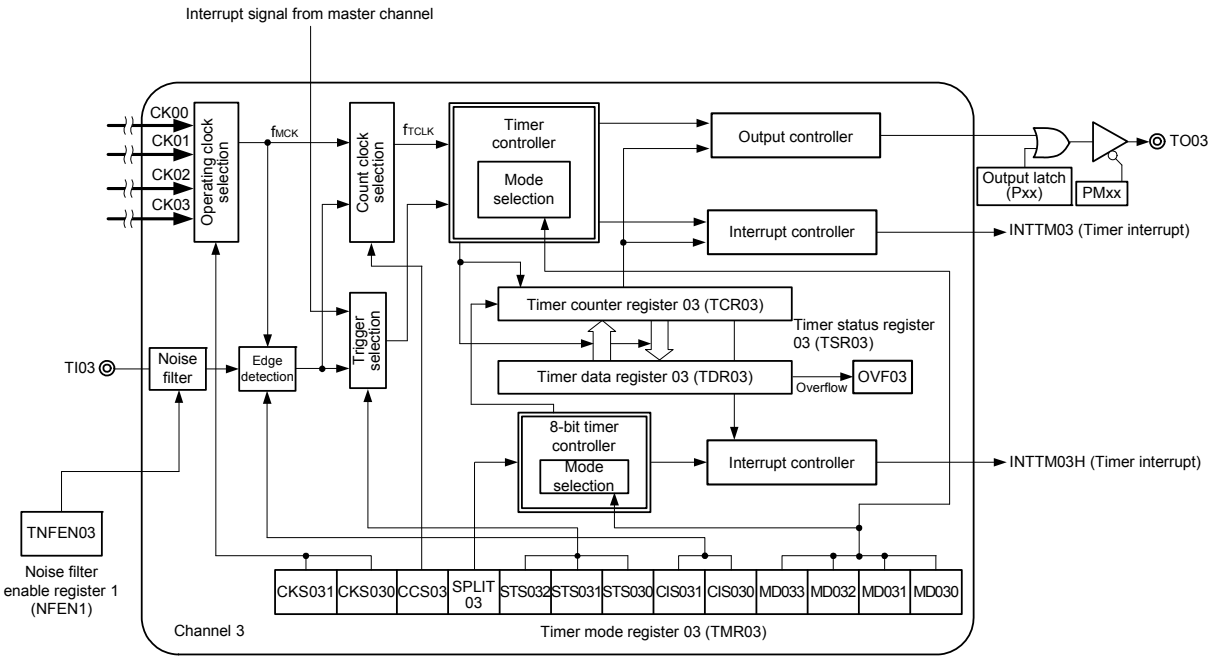


Figure 7 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit 0



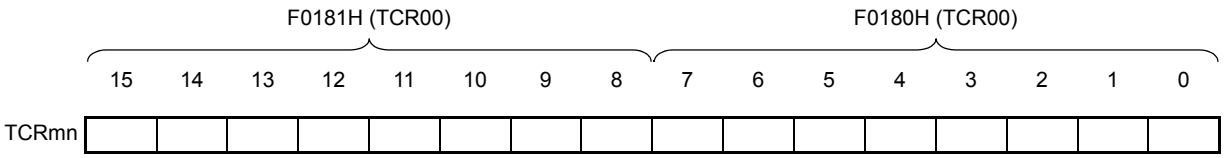
7.2.1 Timer count register mn (TCRmn)

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 **7.3.3 Timer mode register mn (TMRmn)**).

Figure 7 - 6 Format of Timer count register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F0186H, F0187H (TCR03) After reset: FFFFH R
F01C0H, F01C1H (TCR10) to F01C6H, F01C7H (TCR13)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

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

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 7 - 2 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, 1), n: Channel number (n = 0 to 3)

7.2.2 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 m1 and m3 (TMRm1, TMRm3) are 1), it is possible to read and write 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.

Reset signal generation clears this register to 0000H.

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

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02) After reset: 0000H R/W
FFF70H, FFF71H (TDR10), FFF74H, FFF75H (TDR12)

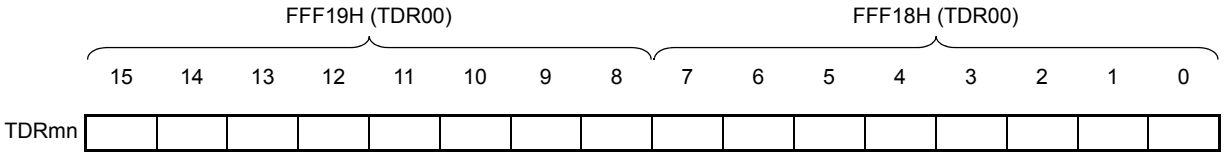
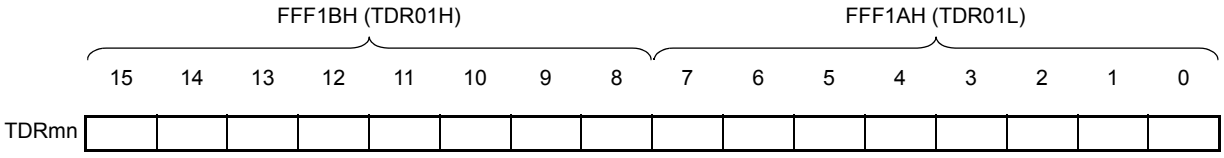


Figure 7 - 8 Format of Timer data register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) After reset: 00H R/W
FFF72H, FFF73H (TDR11), FFF76H, FFF77H (TDR13)



(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, 1), n: Channel number (n = 0 to 3)

7.3 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 register (PMxx)
- Port register (Pxx)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

7.3.1 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.

When the timer array unit 1 is used, be sure to set bit 1 (TAU1EN) 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 7 - 9 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

TAU1EN	Control of timer array unit 1 input clock
0	Stops supply of input clock. • SFR used by the timer array unit 1 cannot be written. • The timer array unit 1 is in the reset status.
1	Supplies input clock. • SFR used by the timer array unit 1 can be read/written.

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

Caution 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 0 (TIS0), noise filter enable register 1 (NFEN1), port mode register 3 (PM3), and port register 3 (P3)).

- **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)**

7.3.2 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 3):
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 3):
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 7 - 10 Format of Timer clock select register m (TPSm) (1/2)

Address: F01B6H, F01B7H (TPS0), F01F6H, F01F7H (TPS1) 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} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	0	0	f_{CLK}	2 MHz	4 MHz	8 MHz	20 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2 MHz	4 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1 MHz	2 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	500 kHz	1 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	250 kHz	500 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	125 kHz	250 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	62.5 kHz	125 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	31.3 kHz	62.5 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	15.6 kHz	31.3 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	7.81 kHz	15.6 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	3.91 kHz	7.81 kHz	19.5 kHz	31.25 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	1.95 kHz	3.91 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	977 Hz	1.95 kHz	4.88 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	488 Hz	977 Hz	2.44 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	244 Hz	488 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61.0 Hz	122 Hz	244 Hz	610 Hz	977 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 000FH).
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 1. Be sure to clear bits 15, 14, 11, 10 to "0".

Caution 2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0 or 1, m = 0 to 3), interrupt requests output from timer array units cannot be used.

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

Remark 2. Waveform of the clock to be selected in the TPSm register which becomes high level for one period of f_{CLK} from its rising edge (m = 1 to 15). For details, see **7.5.1 Count clock (f_{TCLK})**.

Figure 7 - 11 Format of Timer clock select register m (TPSm) (2/2)

Address: F01B6H, F01B7H (TPS0), F01F6H, F01F7H (TPS1) 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 m21	PRS m20	Selection of operation clock (CKm2) **Note**
0	0	$f_{CLK}/2$
0	1	$f_{CLK}/2^2$
1	0	$f_{CLK}/2^4$
1	1	$f_{CLK}/2^6$

PRS m31	PRS m30	Selection of operation clock (CKm3) **Note**
0	0	$f_{CLK}/2^8$
0	1	$f_{CLK}/2^{10}$
1	0	$f_{CLK}/2^{12}$
1	1	$f_{CLK}/2^{14}$

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 000FH).
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, 10 to "0".

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

Remark 2. For details of a signal of $f_{CLK}/2^r$ selected with the TPSm register, see **7.5.1 Count clock (f_{TCLK})**.

7.3.3 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 **7.8 Independent Channel Operation Function of Timer Array Unit** and **7.9 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 the bit 11 of TMRmn register.
TMRm2: MASTERmn bit (n = 2)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0: Fixed to 0

Figure 7 - 12 Format of Timer mode register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), F01D0H, F01D1H (TMR10) to F01D6H, F01D7H (TMR13) 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)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKS mn1	CKS mn0	0	CCS mn	0 **Note**	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKS mn1	CKS mn0	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 • For unit 0 For channel 0, valid edge of input signal selected by TIS0 For channel 1, valid edge of input signal selected by TIS0 For channel 3, valid edge of input signal selected by TI03
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, 1), n: Channel number (n = 0 to 3)

Figure 7 - 13 Format of Timer mode register mn (TMRmn) (2/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), After reset: 0000H R/W
F01D0H, F01D1H (TMR10) to F01D6H, F01D7H (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKS mn1	CKS mn0	0	CCS mn	0 Note**Note**	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

(Bit 11 of TMRmn (n = 2))

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 the channel 2 can be set as a master channel (MASTERmn = 1). Be sure to use channel 0 is 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.

STS mn2	STS mn1	STS mn0	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, 1), n: Channel number (n = 0 to 3)

Figure 7 - 14 Format of Timer mode register mn (TMRmn) (3/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), After reset: 0000H R/W
F01D0H, F01D1H (TMR10) to F01D6H, F01D7H (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKS mn1	CKS mn0	0	CCS mn	0 Note**Note**	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CIS mn1	CIS mn0	Selection of TImn pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
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, 1), n: Channel number (n = 0 to 3)

Figure 7 - 15 Format of Timer mode register mn (TMRmn) (4/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), F01D0H, F01D1H (TMR10) to F01D6H, F01D7H (TMR13)

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)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKS mn1	CKS mn0	0	CCS mn	0 Note1	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

MD mn3	MD mn2	MD mn1	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer / Square wave output / Divider function / 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 / 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 table below).					

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MD mn0	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.
	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.

Note 1. Bit 11 is fixed to 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, and recounting is started (does not occur the interrupt request).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

7.3.4 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 7 - 3** 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 7 - 16 Format of Timer status register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01A6H, F01A7H (TSR03) After reset: 0000H R
F01E0H, F01E1H (TSR10) to F01E6H, F01E7H (TSR13)

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, 1), n: Channel number (n = 0 to 3)

Table 7 - 3 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.

7.3.5 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 7 - 17 Format of Timer channel enable status register m (TEm)

Address: F01B0H, F01B1H (TE0), F01F0H, F01F1H (TE1) 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	0	0	0	0	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, 1), n: Channel number (n = 0 to 3)

7.3.6 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 7 - 18 Format of Timer channel start register m (TSm)

Address: F01B2H, F01B3H (TS0) to F01F2H, F01F3H (TS1) 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	0	0	0	0	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 7 - 4** in **7.5.2 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 7 - 4** in **7.5.2 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 7 - 4** in **7.5.2 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.

(Cautions and Remarks are listed on the next page.)

Caution 1. Be sure to clear bits 15 to 12, 10, 8 to 4 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, 1), n: Channel number (n = 0 to 3)

7.3.7 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 7 - 19 Format of Timer channel stop register m (TTm)

Address: F01B4H, F01B5H (TT0), F01F4H, F01F5H (TT1) 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	0	0	0	0	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	TEmn bit is cleared to 0 and the count operation is stopped.
1	Operation is stopped (stop trigger is generated). 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, 8 to 4 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, 1), n: Channel number (n = 0 to 3)

7.3.8 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channels 0 and 1 of unit 0 timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 20 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	TIS04	0	TIS02	TIS01	TIS00

TIS04	Selection of timer input used with channel 0
0	No timer input
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	No timer input
0	0	1	Event input signal from ELC
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

Caution 1. At least 1/f_{MCK} + 10 ns is necessary as the high-level and low-level widths of the timer input to be selected. Thus, the TIS02 bit cannot be set to 1 when f_{SUB} is selected as f_{CLK} (CSS in CKC register = 1).

Caution 2. When selecting an event input signal from the ELC using timer input select register 0 (TIS0), select f_{CLK} using timer clock select register 0 (TPS0).

7.3.9 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 7 - 21 Format of Timer output enable register m (TOEm)

Address: F01BAH, F01BBH (TOE0), F01FAH, F01FBH (TOE1) 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	0	0	0	0	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 4 , 2 to 0 to "0" in TOE0 register. Set to 0000H register in TOE1 register.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

7.3.10 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 oh 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 TI03/TO03 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 7 - 22 Format of Timer output register m (TOm)

Address: F01B8H, F01B9H (TO0), F01F8H, F01F9H (TO1) 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	0	0	0	0	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 4 , 2 to 0 to “0” in TO0 register. Set to 0000H register in TO1 register.

Remark m: Unit number (m = 0), n: Channel number (n = 3)

7.3.11 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 7 - 23 Format of Timer output level register m (TOLm)

Address: F01BCH, F01BDH (TOL0), F01FCH, F01FDH (TOL1) 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	0	0	0	0	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 4 , 2 to 0 to "0" in TOL0 register. Set to 0000H register in TOL1 register.

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, 1), n: Channel number (n = 0 to 3)

7.3.12 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), 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 7 - 24 Format of Timer output mode register m (TOMm)

Address: F01BEH, F01BFH (TOM0), F01FEH, F01FFH (TOM1) 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	0	0	0	0	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 (INTTMmp) of the slave channel)

Caution Be sure to clear bits 15 to 4 , 2 to 0 to “0” in TOM0 register. Set to 0000H register in TOM1 register.

Remark m: Unit number (m = 0, 1)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
n = 0, p = 1, 2, 3
n = 2, p = 3
(For details of the relation between the master channel and slave channel, refer to **7.4.1 Basic rules of simultaneous channel operation function**)

7.3.13 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 OFF, only synchronization is performed with the operation clock of target channel (f_{MCK}) **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 **7.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)**, **7.5.2 Start timing of counter**, and **7.7 Timer Input (TImn) Control**.

Figure 7 - 25 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	0	0	0	0	TNFEN03	0	TNFEN01	TNFEN00

TNFEN03	Enable/disable using noise filter of TI03 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of input signal selected by TIS0 in channel 1 of unit 0
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of input signal selected by TIS0 in channel 0 of unit 0
0	Noise filter OFF
1	Noise filter ON

7.3.14 Registers controlling port functions of pins to be used for timer I/O

Using port pins for the timer array unit functions requires setting of the registers that control the port functions multiplexed on the target pins (port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx)). For details, see **5.3.1 Port mode registers (PMxx)**, **5.3.2 Port registers (Pxx)**.

For details, see **5.5 Register Settings When Using Alternate Function**.

7.4 Basic Rules of Timer Array Unit

7.4.1 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) 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 0 is set as a master channel, channel 1 or those that follow (channels 1, 2, 3) can be set as a slave channel.

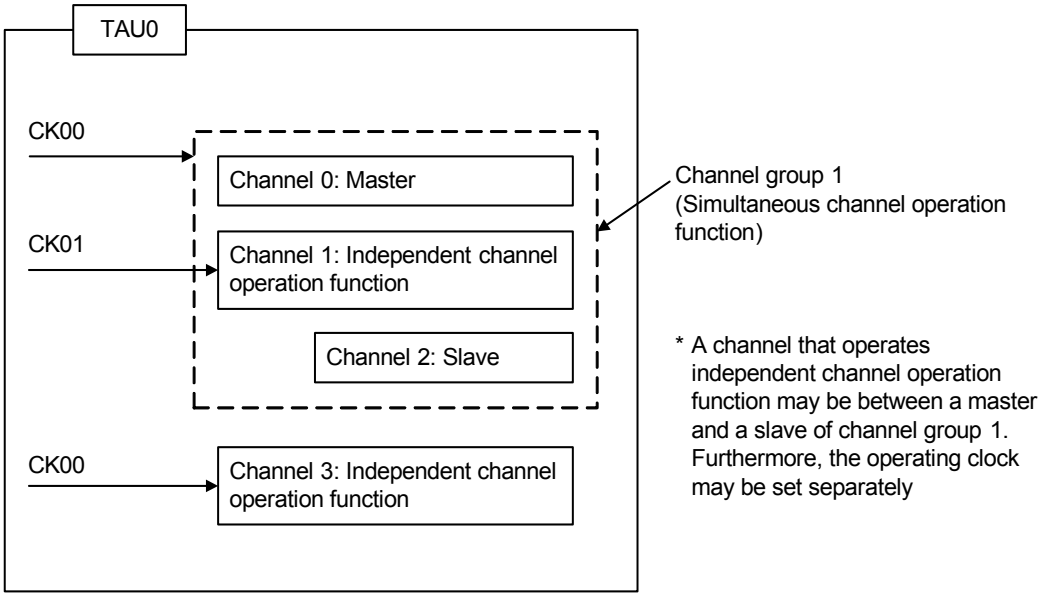
(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example: If channels 0 and 2 are set as master channels, channels 1 can be set as the slave channel of master channel 0. Channel 3 cannot be set as the slave channel 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 **7.4.1 Basic rules of simultaneous channel operation function** do not apply to the channel groups.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Example



7.4.2 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 (PWM) cannot be used.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3)

7.5 Operation of Counter

7.5.1 Count clock (fTCLK)

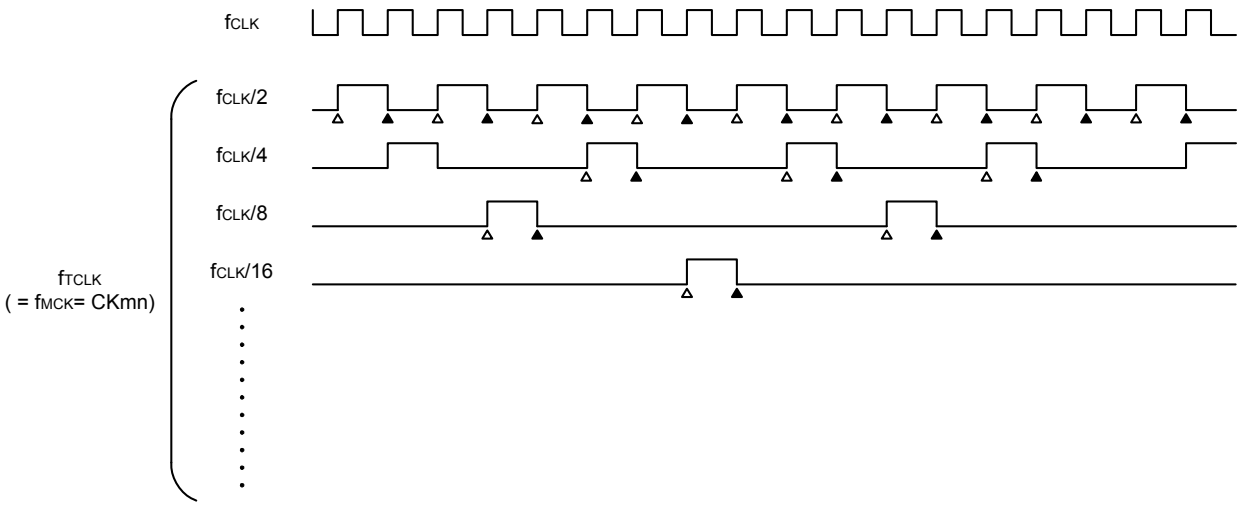
The count clock (fTCLK) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

- 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 7 - 26 Timing of fCLK and count clock (fTCLK) (When CCSmn = 0)



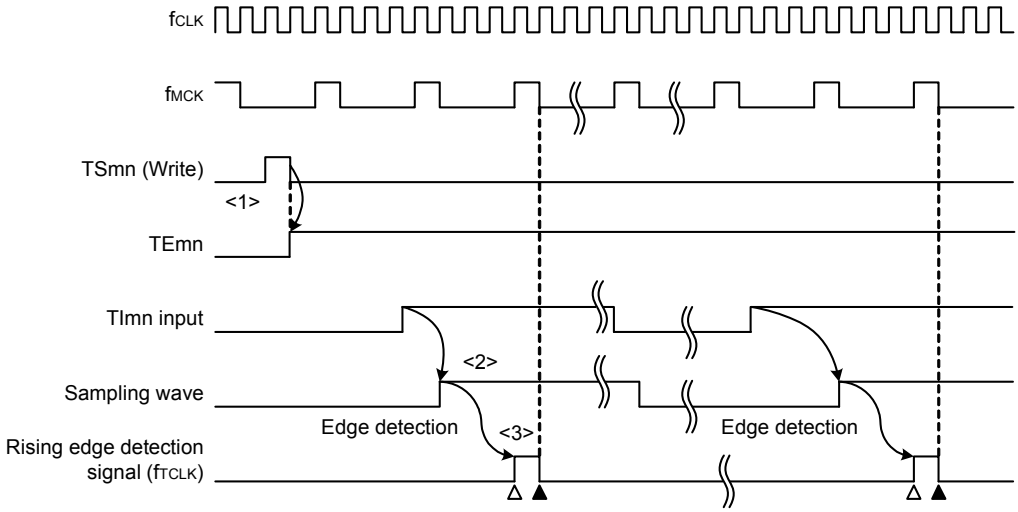
Remark 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter
Remark 2. fCLK: CPU/peripheral hardware clock

(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 period of f_{MCK} from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clock).

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 7 - 27 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 function are the same as that shown in Figure 7 - 27.

7.5.2 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 7 - 4.

Table 7 - 4 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 **7.5.3 (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 **7.5.3 (2) Operation of event counter mode**).
Capture mode	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 **7.5.3 (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 **7.5.3 (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 **7.5.3 (5) Operation of capture & one-count mode (high-level width measurement)**).

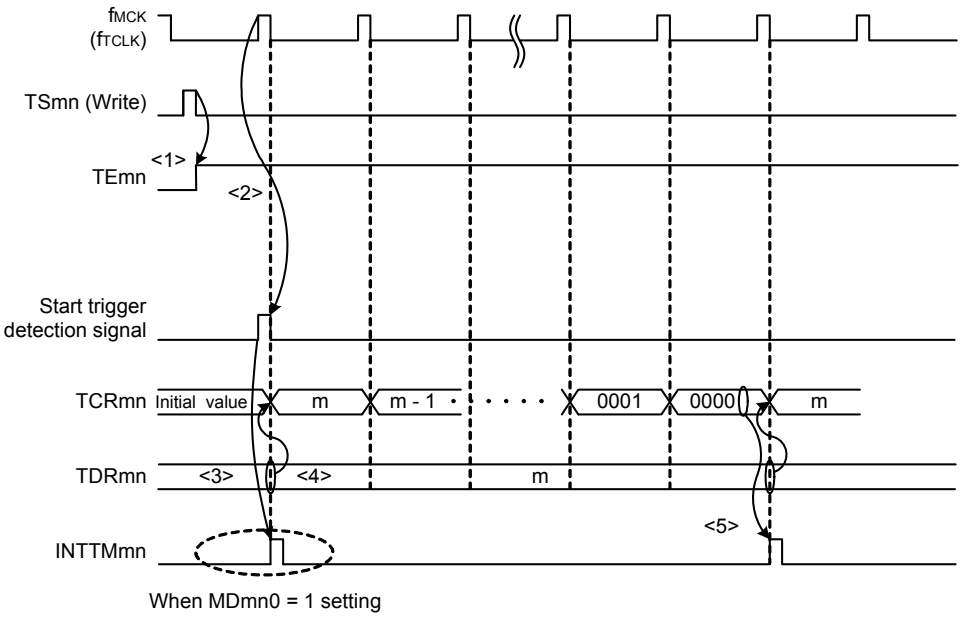
7.5.3 Operation of counter

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 7 - 28 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

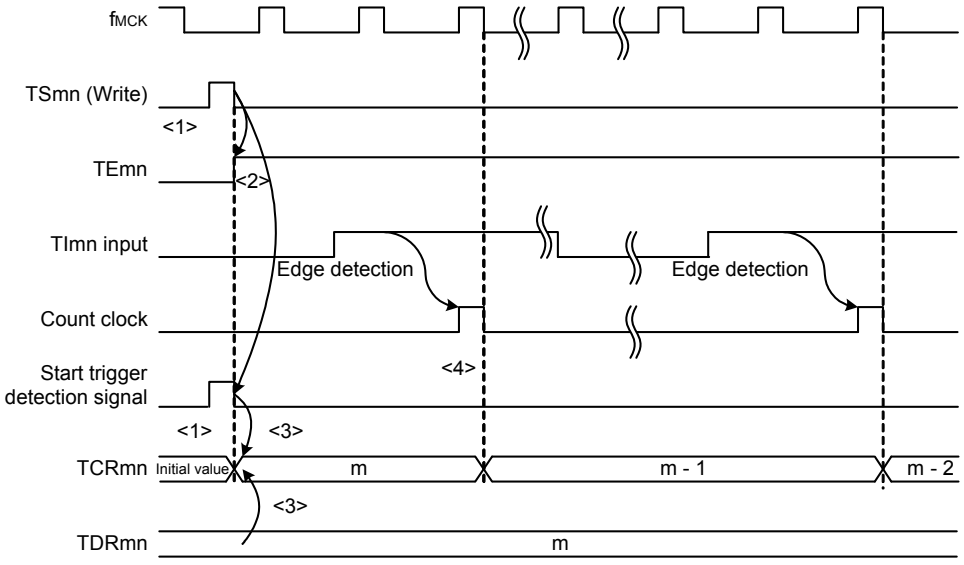
<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).

<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 7 - 29 Operation Timing (In Event Counter Mode)



Remark The timing is shown in Figure 7 - 29 indicates while 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.

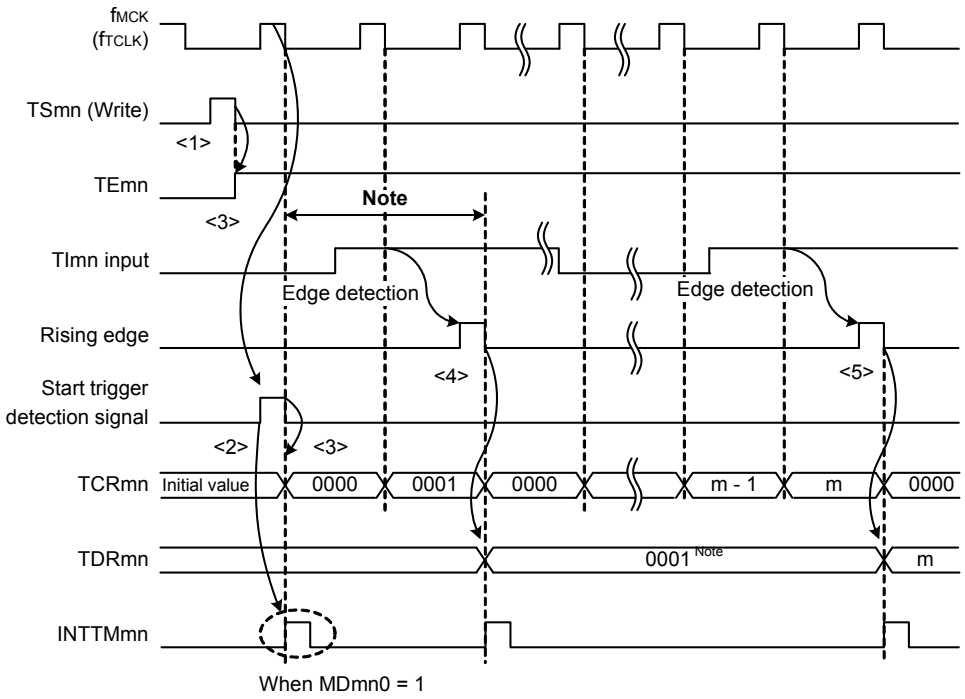
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.

<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 7 - 30 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 timing is shown in Figure 7 - 30 indicates while 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.

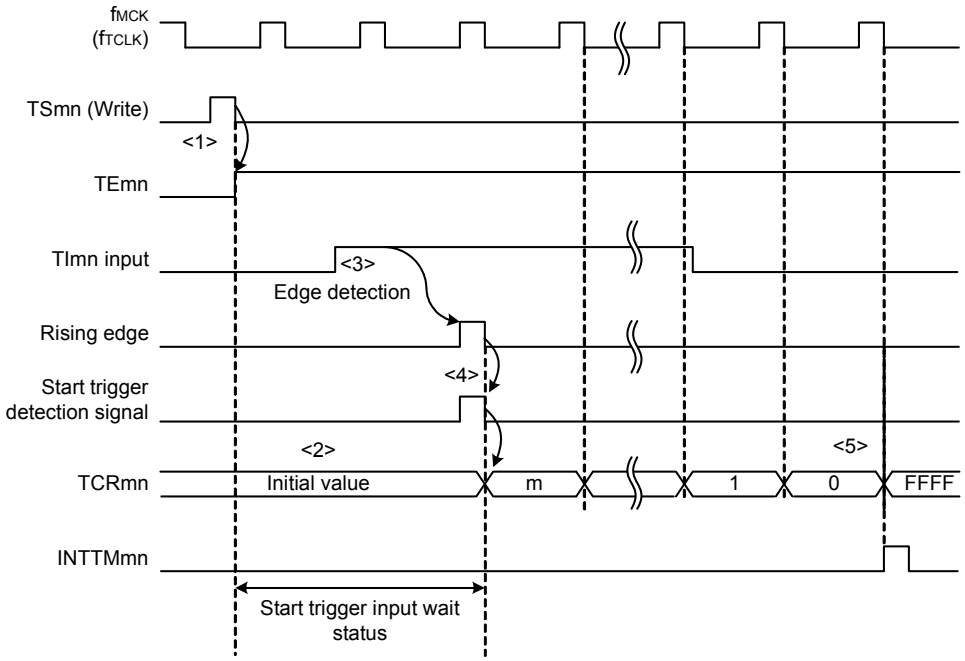
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<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 7 - 31 Operation Timing (In One-count Mode)



Remark The timing is shown in Figure 7 - 31 indicates while 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) Operation of capture & one-count mode (high-level width measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).

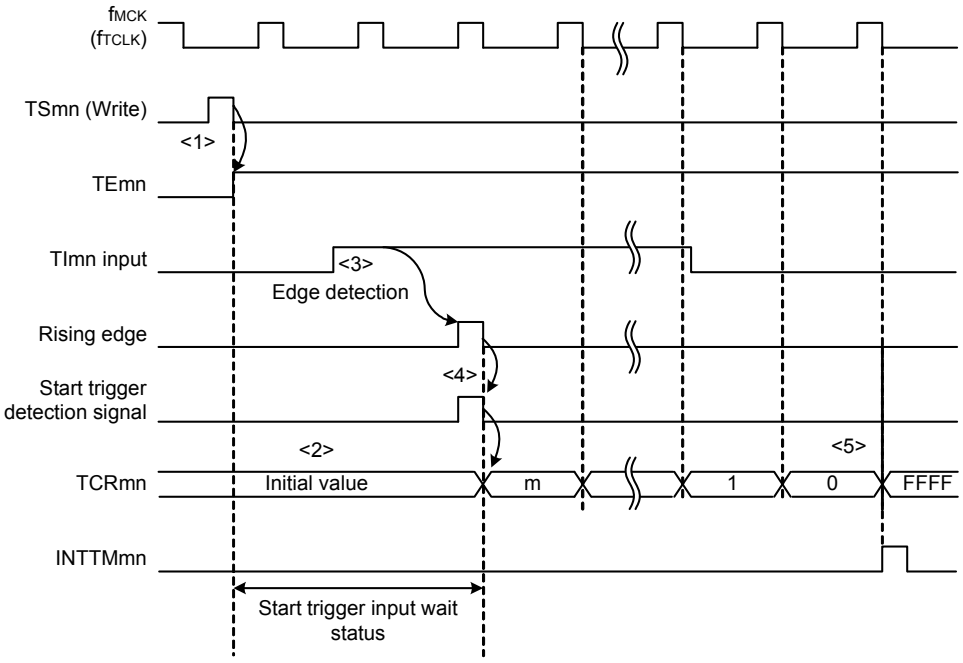
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<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 7 - 32 Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

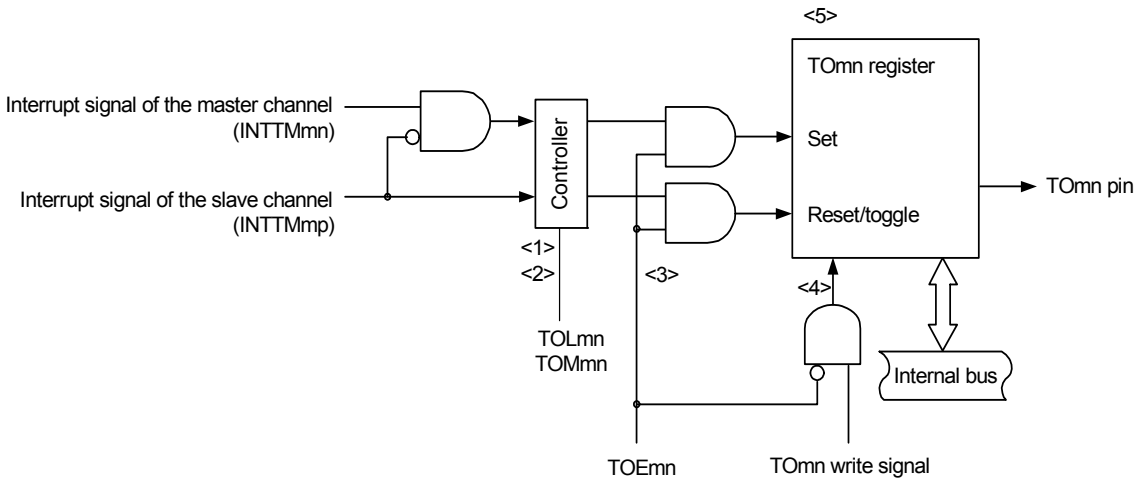


Remark The timing is shown in Figure 7 - 32 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

7.6 Channel Output (TOmn pin) Control

7.6.1 TOmn pin output circuit configuration

Figure 7 - 33 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 INTTMmp (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 INTTMmp (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: Forward operation (INTTMmn → set, INTTMmp → reset)
When TOLmn = 1: Reverse operation (INTTMmn → reset, INTTMmp → set)

When INTTMmn and INTTMmp are simultaneously generated, (0% output of PWM), INTTMmp (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTMmp (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 INTTMmp (slave channel timer interrupt) is transmitted to the TOm register.

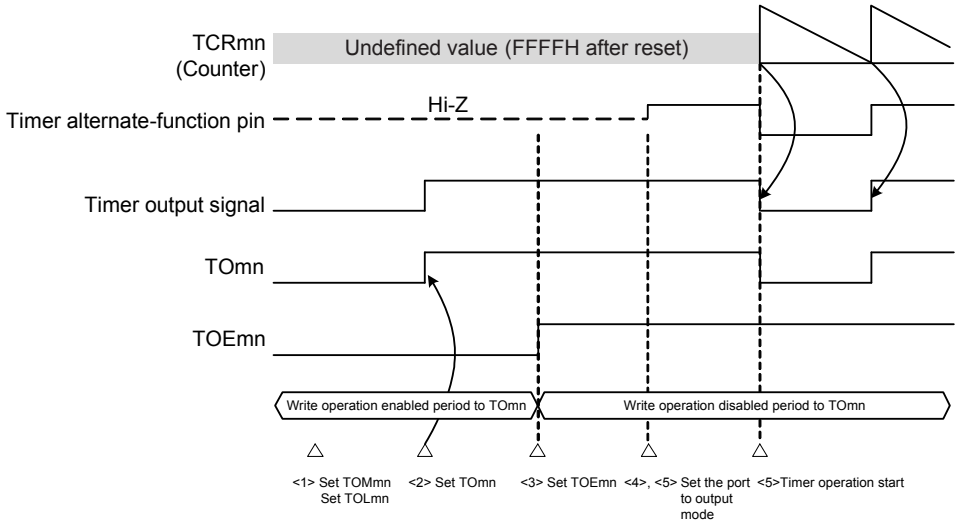
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

Remark m: Unit number (m = 0)
n: Channel number (n = 3 (n = 0, 2 for master channel))
p: Slave channel number (p = 3)

7.6.2 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 7 - 34 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 I/O setting is set to output (see **7.3.14 Registers controlling port functions of pins to be used for timer I/O**).

<5> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 3)

7.6.3 Cautions on Channel Output Operation

(1) Changing values set in the registers TOm, TOEm, TOLm, and TOMm 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 7.8 and 7.9.

When the values set to the TOEm and TOLm 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 = 3)

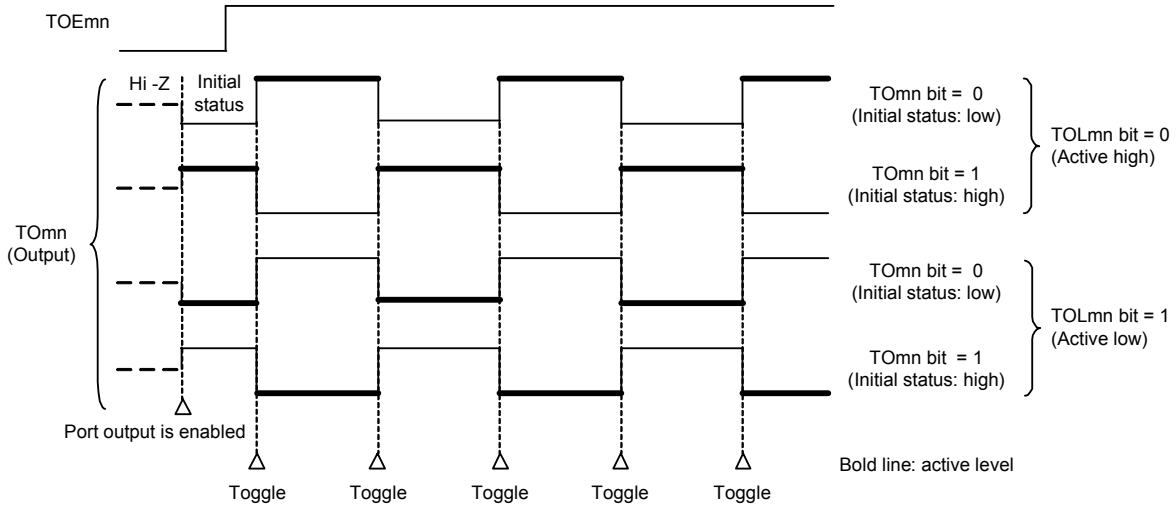
(2) Default level of TOmn pin and output level after timer operation start

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 7 - 35 TOmn Pin Output Status at Toggle Output (TOMmn = 0)



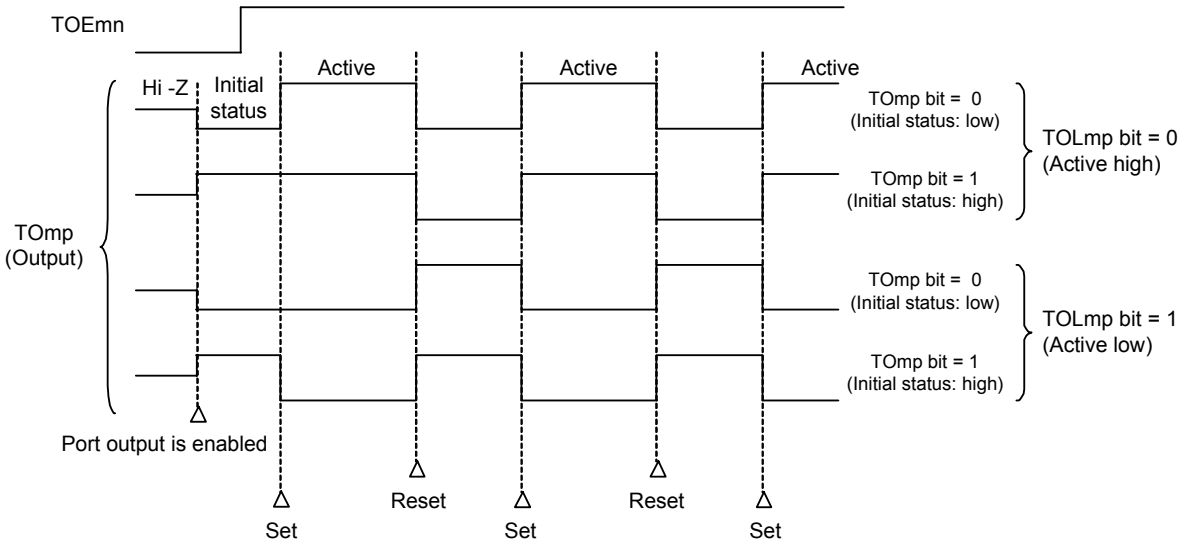
Remark 1. Toggle: Reverse TOmn pin output status

Remark 2. m: Unit number (m = 0), n: Channel number (n = 3)

(b) When operation starts with slave channel output mode (TOMmn = 1) setting (PWM output))

When slave channel output mode (TOMmn = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 7 - 36 TOmn Pin Output Status at PWM Output (TOMmn = 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), n: Channel number (p = 3)

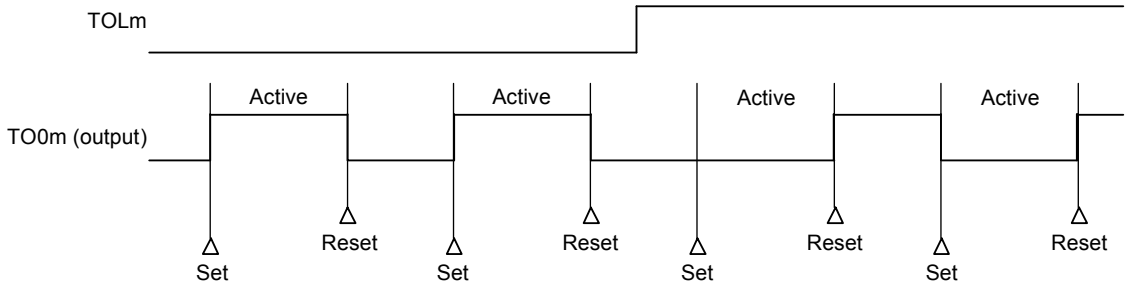
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(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 7 - 37 Operation when TOLm Register Has Been Changed 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 = 3)

(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.

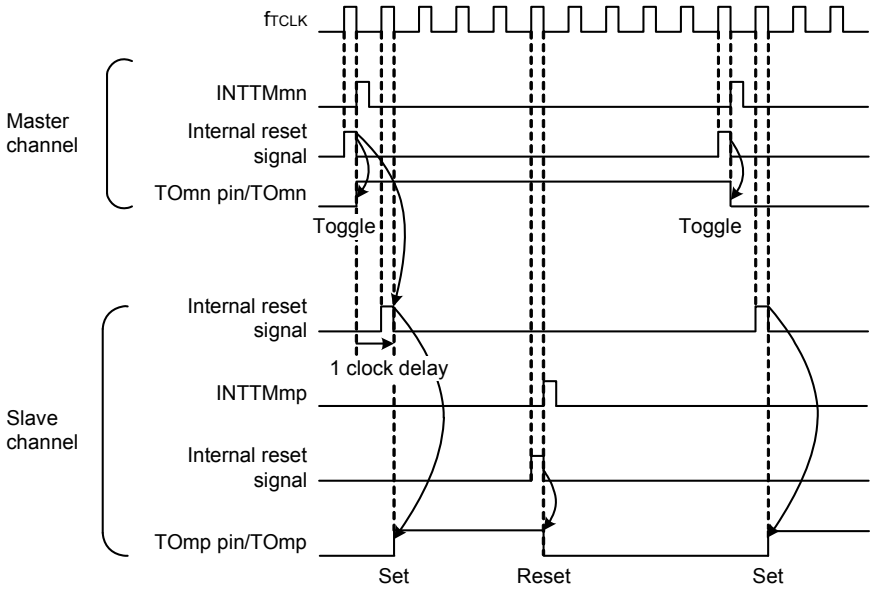
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 7 - 38 shows the set/reset operating statuses where the master/slave channels are set as follows.

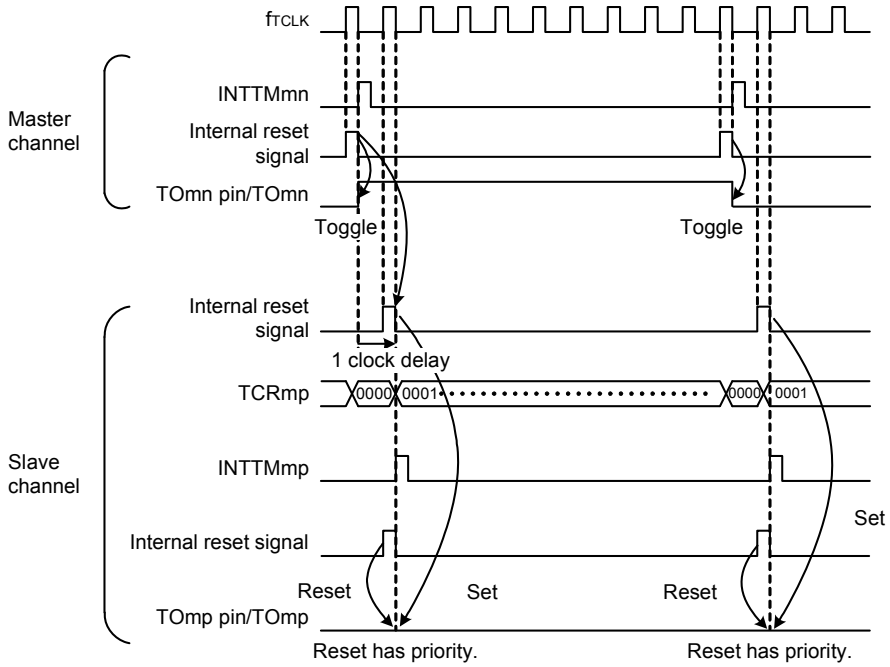
Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 7 - 38 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 = 3 (n = 0, 2 for master channel))
p: Slave channel number (p = 3)

7.6.4 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).

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.

Remark m: Unit number (m = 0), n: Channel number (n = 3)

7.6.5 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.

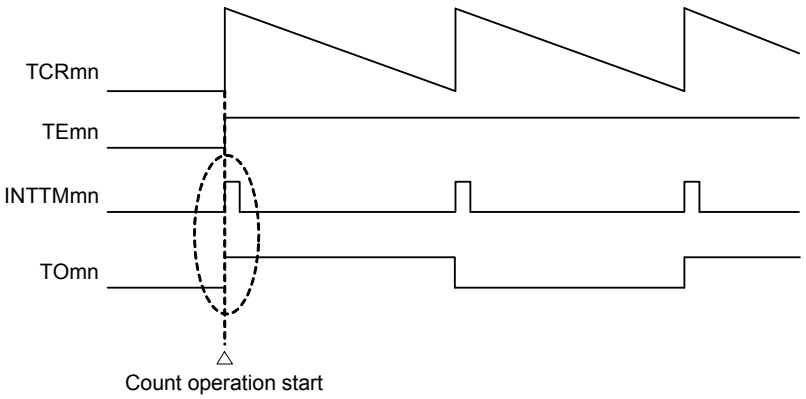
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled.

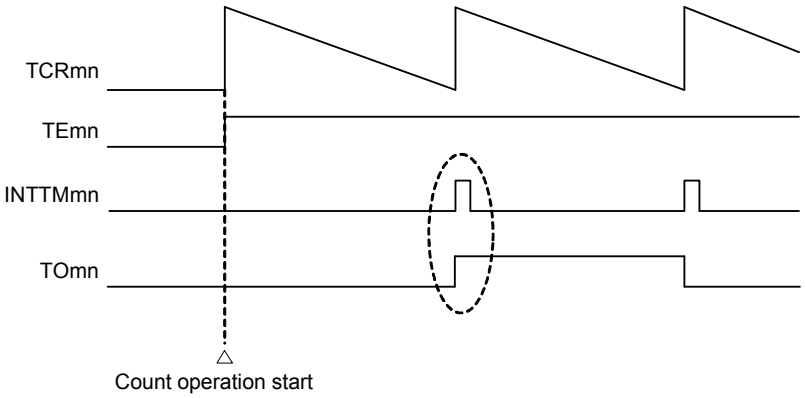
Figure 7 - 39 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 7 - 39 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 = 3)

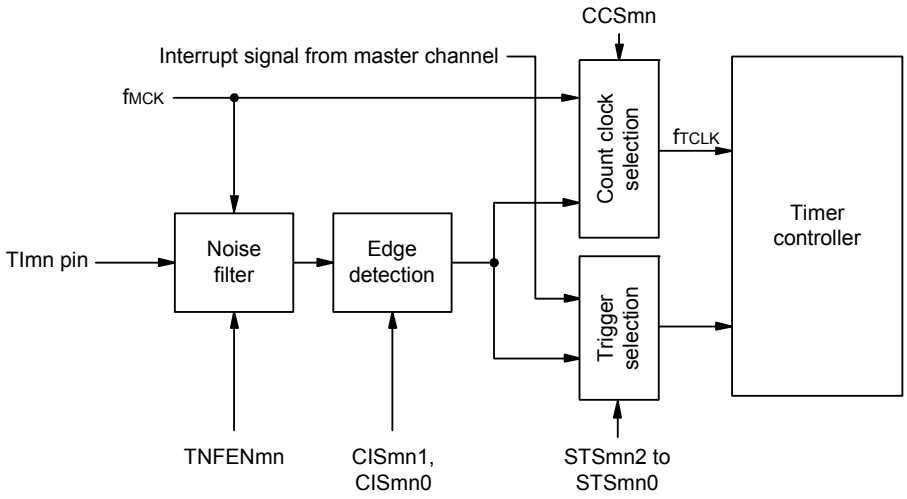
7.7 Timer Input (TImn) Control

7.7.1 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.

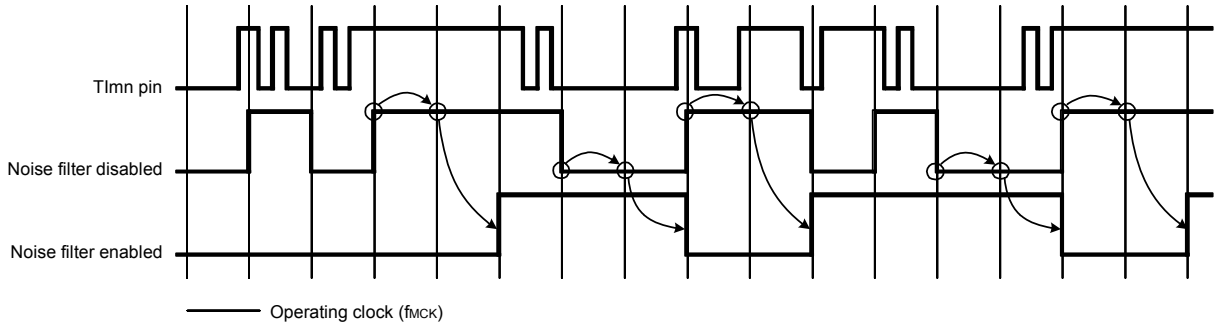
Figure 7 - 40 Input Circuit Configuration



7.7.2 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 7 - 41 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. When actually inputting waveforms, input them according to the TImn input high-level and low-level widths listed in CHAPTER 31 AC Characteristics.

7.7.3 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).

7.8 Independent Channel Operation Function of Timer Array Unit

7.8.1 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.

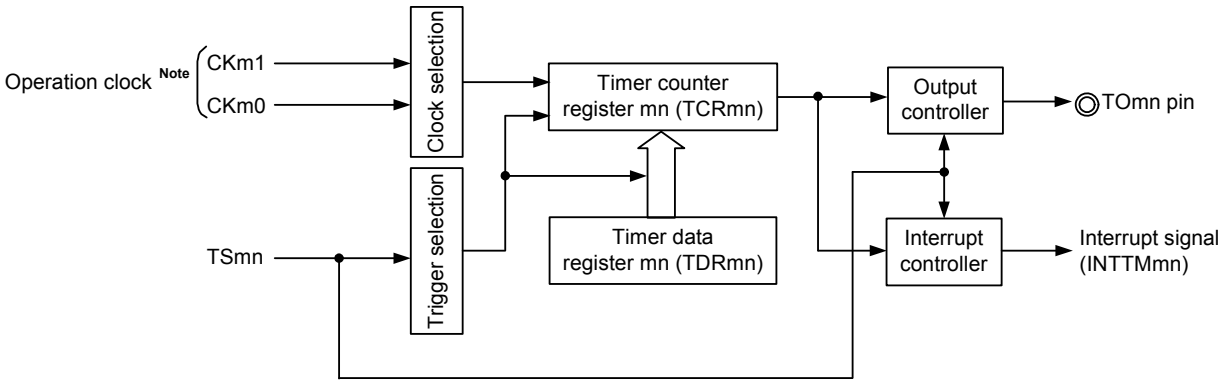
After that, the TCRmn register count down in synchronization with the count clock.

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.

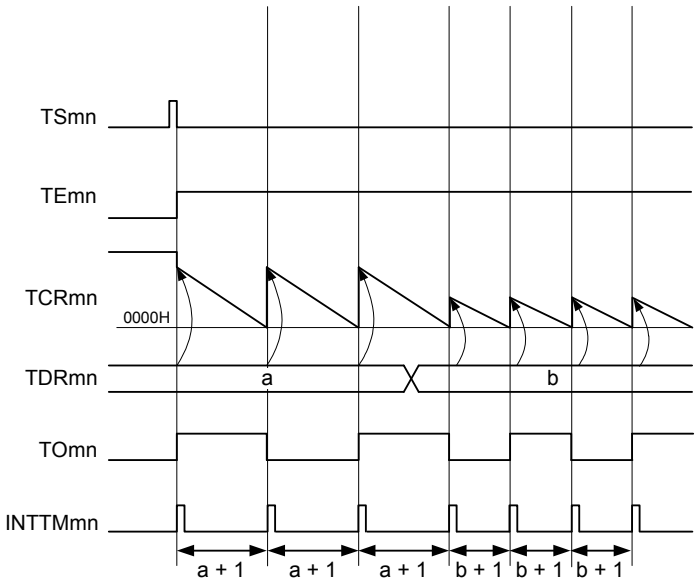
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 7 - 42 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 7 - 43 Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)

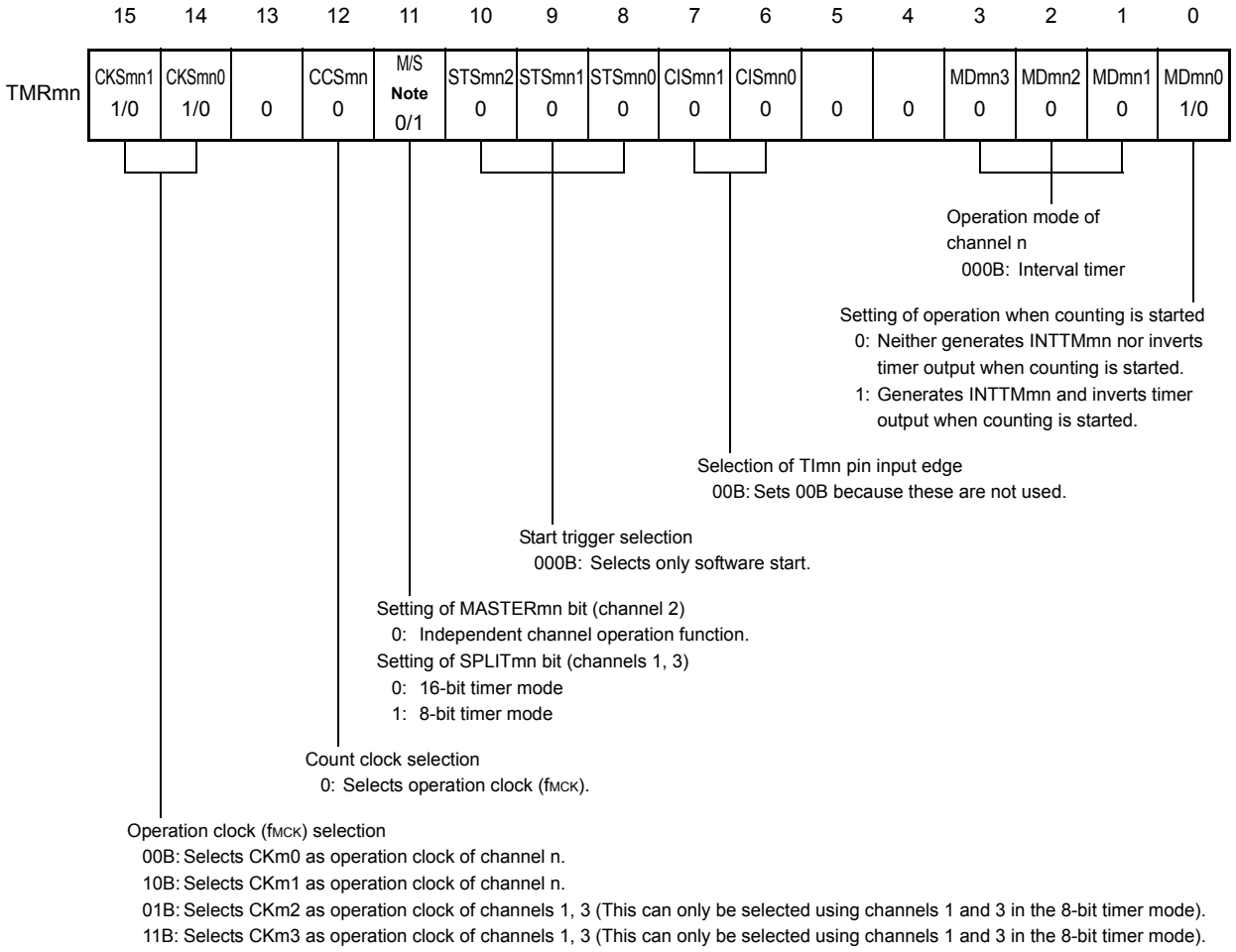


Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

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 7 - 44 Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

Bit n

TOm | TOmn 1/0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

Bit n

TOEm | TOEmn 1/0

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)

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: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 7 - 45 Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied (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 7 - 46 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. →	Input clock supply for timer array unit m is stopped 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, 1), n: Channel number (n = 0 to 3)

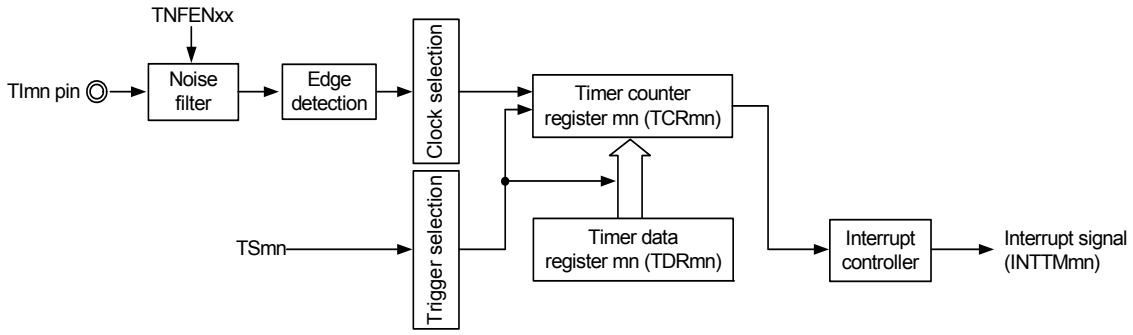
7.8.2 Operation as external event counter

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

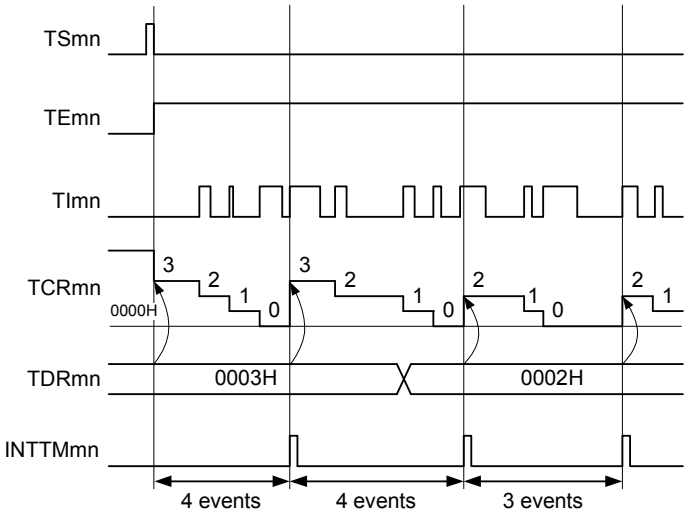
Timer count register mn (TCRmn) operates as a down counter in the event counter mode.
The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) to 1.
The TCRmn register counts down each time 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 7 - 47 Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 3)

Figure 7 - 48 Example of Basic Timing of Operation as External Event Counter

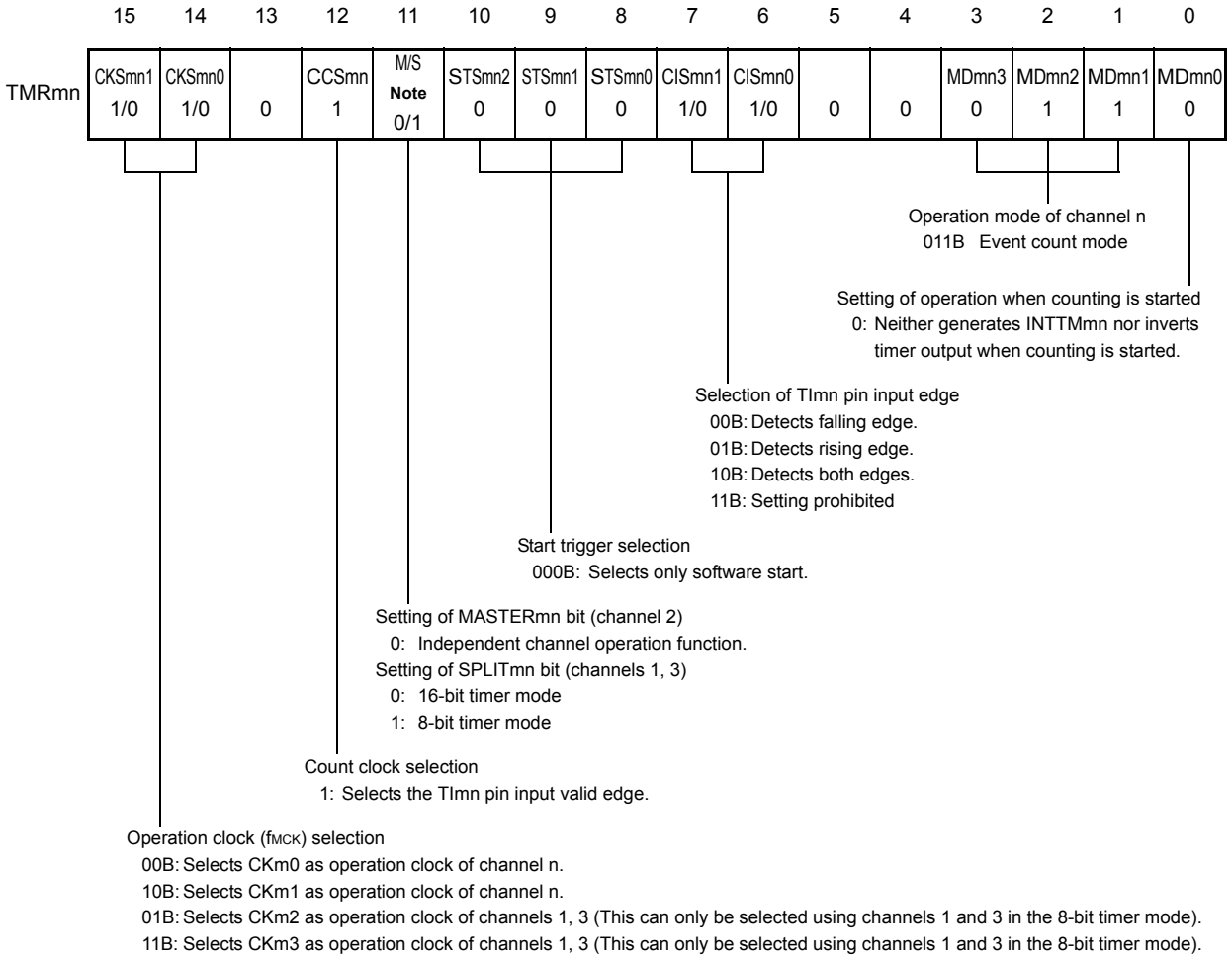


Remark 1. m: Unit number (m = 0), n: Channel number (n = 3)

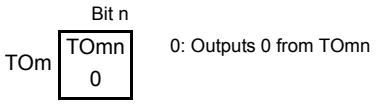
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 7 - 49 Example of Set Contents of Registers in External Event Counter Mode (1/2)

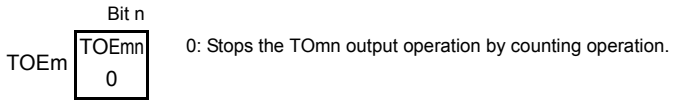
(a) Timer mode register mn (TMRmn)



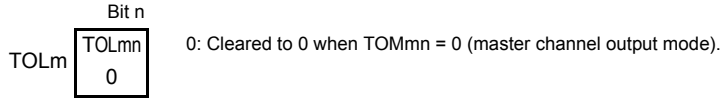
(b) Timer output register m (TOm)



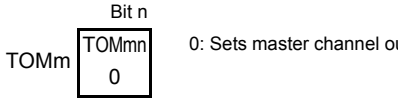
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 3)

Figure 7 - 50 Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. 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. →	Input clock supply for timer array unit m is stopped 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 = 3)

7.8.3 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.

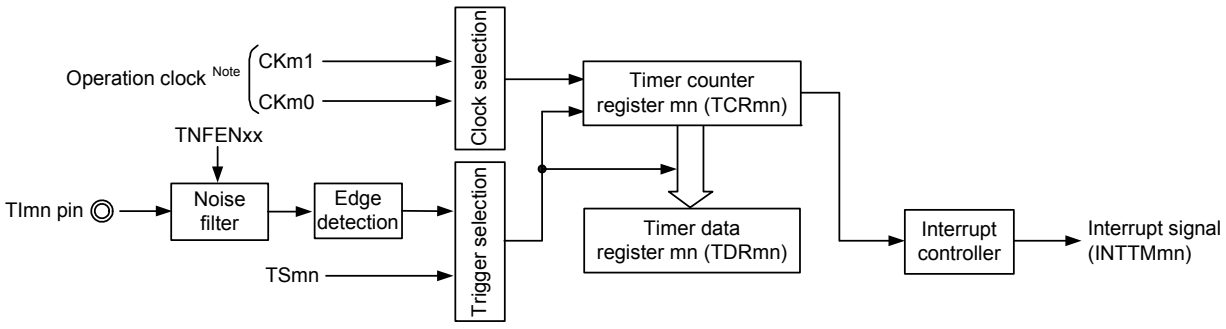
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status 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.

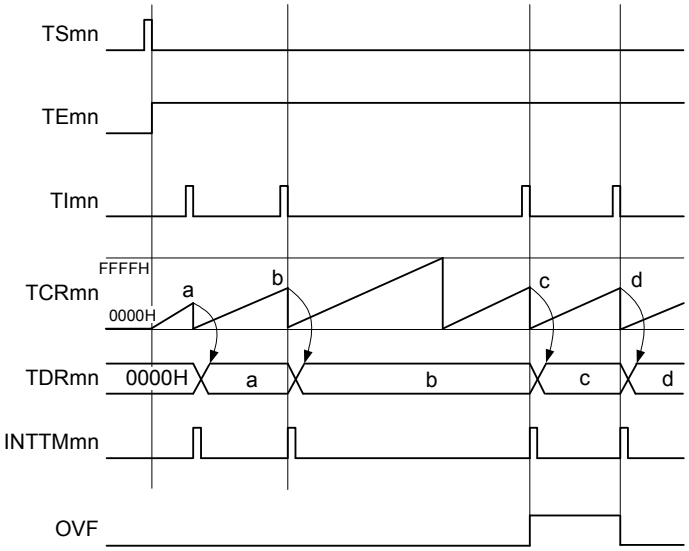
Figure 7 - 51 Block Diagram of Operation as Input Pulse Interval Measurement



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 = 3)

Figure 7 - 52 Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)

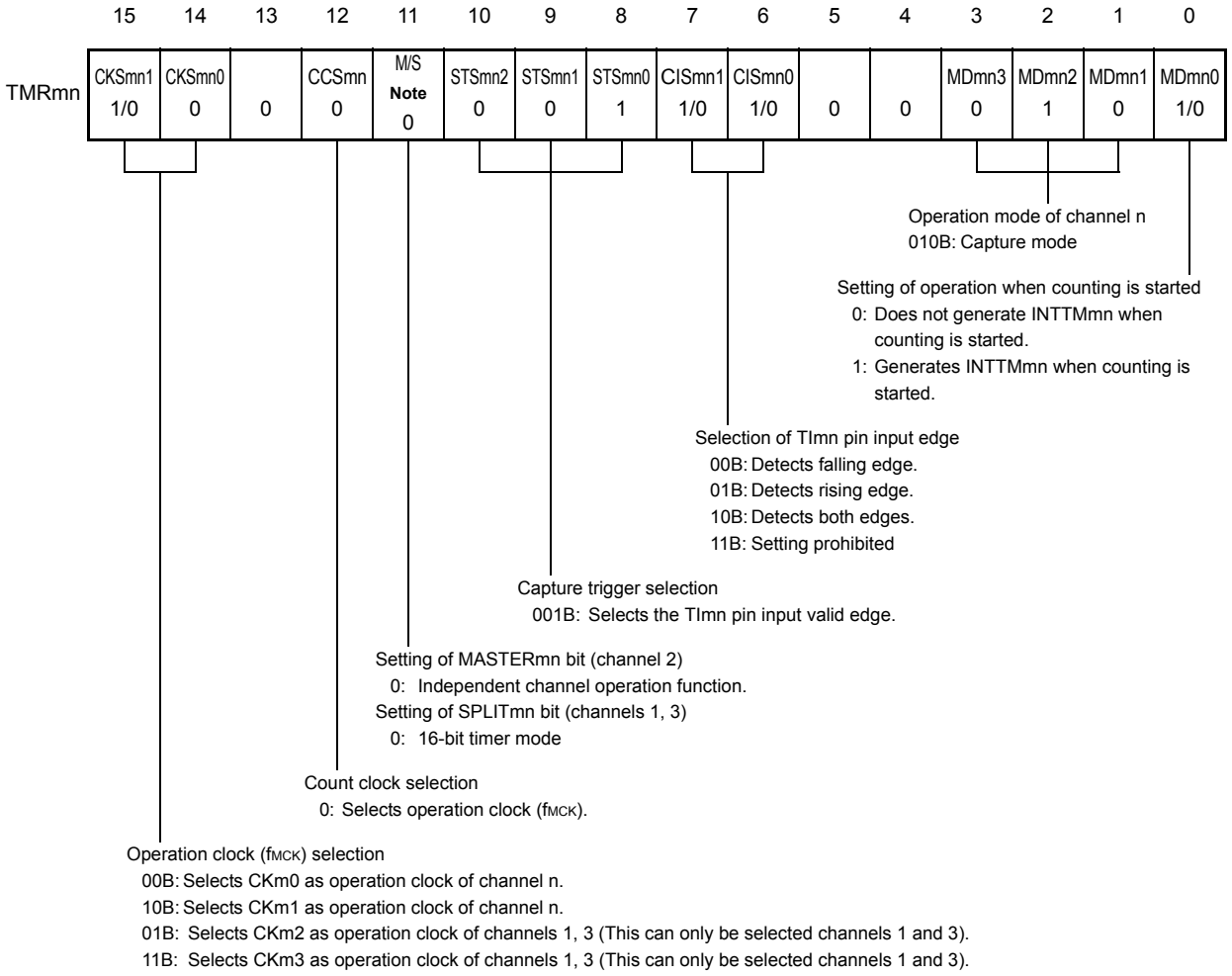


Remark 1. m: Unit number (m = 0), n: Channel number (n = 3)

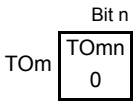
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 7 - 53 Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)

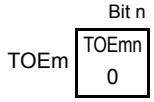


(b) Timer output register m (TOm)



0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

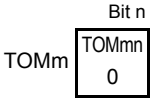


0: Stops TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)



0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 3)

Figure 7 - 54 Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register m (PER0) to 1. →	Input clock supply for timer array unit m is supplied. 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 registers 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. 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. →	Input clock supply for timer array unit m is stopped 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 = 3)

7.8.4 Operation as input signal high-/low-level width measurement

Caution When using a channel to implement the LIN-bus, set bit 1 (ISC1) of the input switch control register (ISC) to 1. In the following descriptions, read TImn as RxD0.

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.

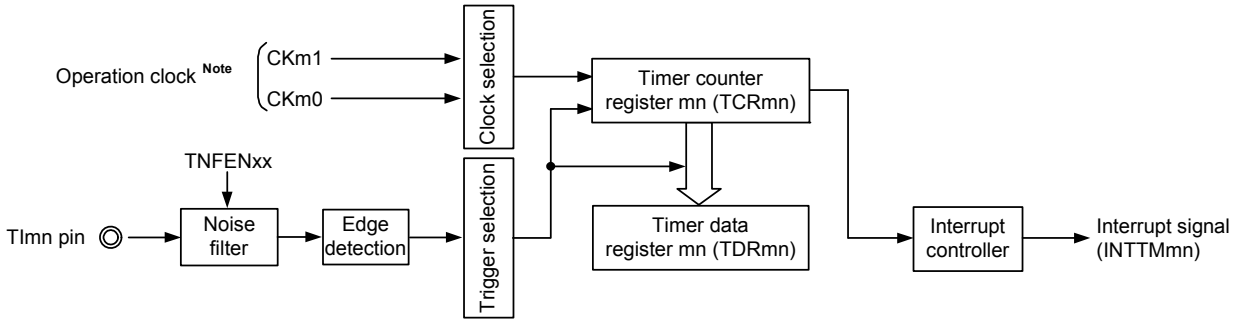
Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

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.

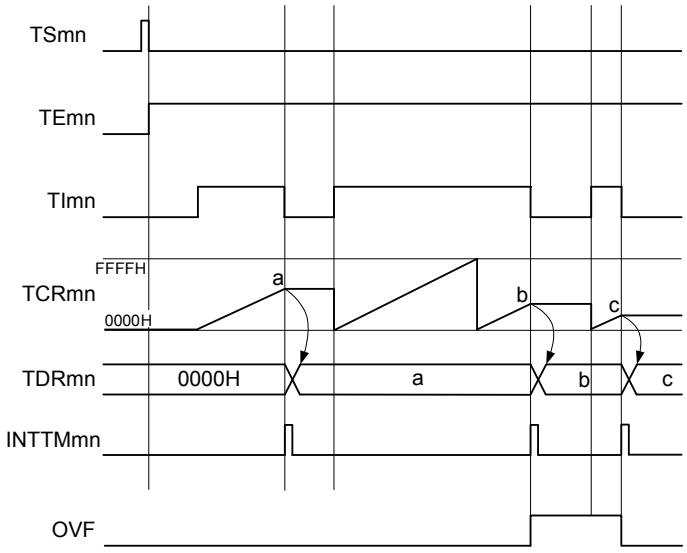
CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 7 - 55 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 7 - 56 Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement

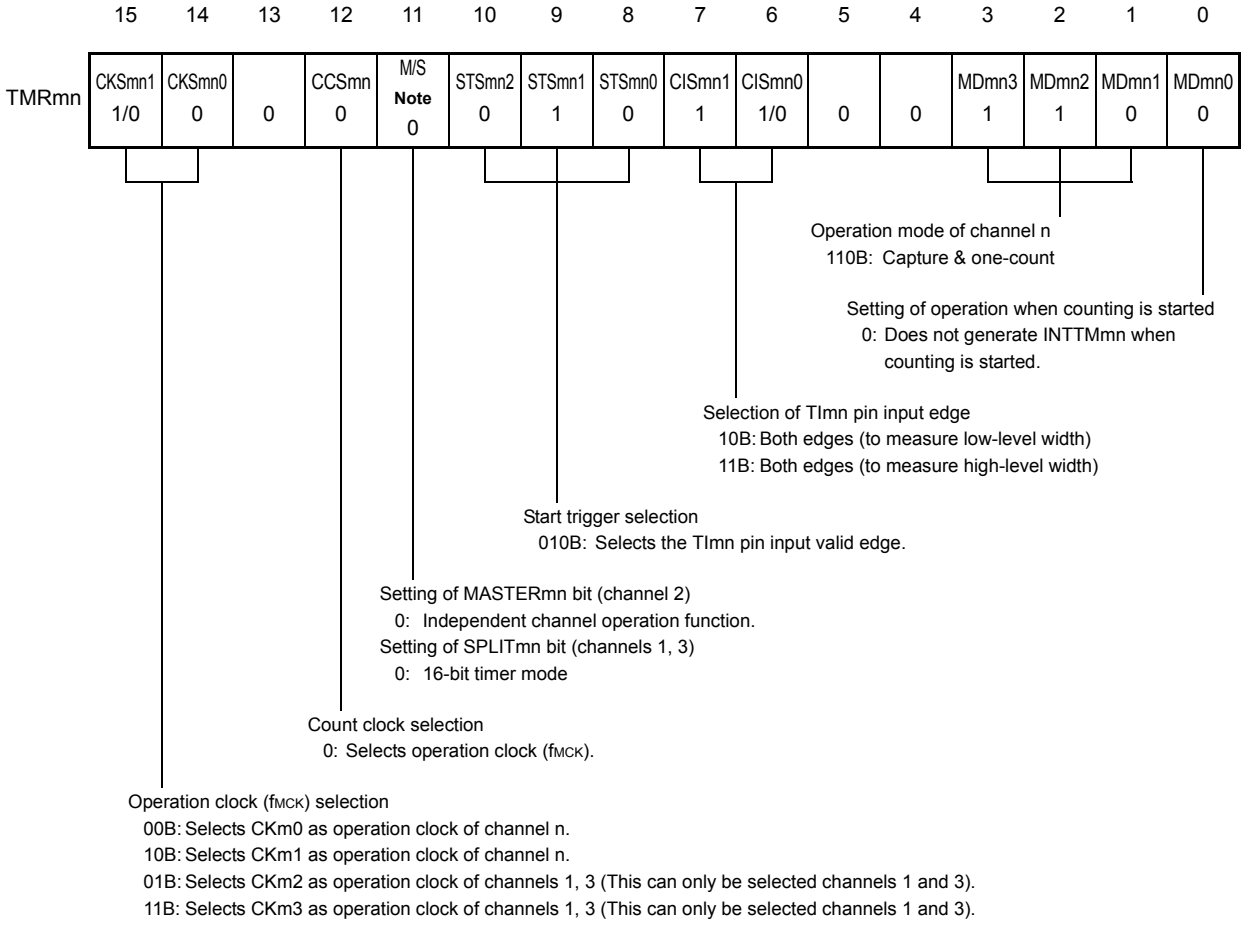


Remark 1. m: Unit number (m = 0), n: Channel number (n = 3)

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 7 - 57 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)

	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 master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 3)

Figure 7 - 58 Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. 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 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.	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. →	Input clock supply for timer array unit m is stopped 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 = 3)

7.8.5 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 can also generate INTTMmn (timer interrupt) at any interval by making a software set TSmn = 1 and the count down start during the period of 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)

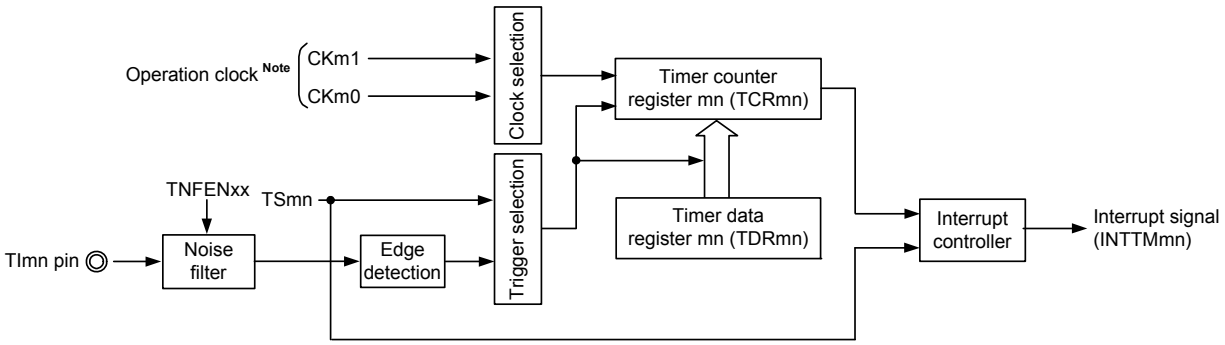
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

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.

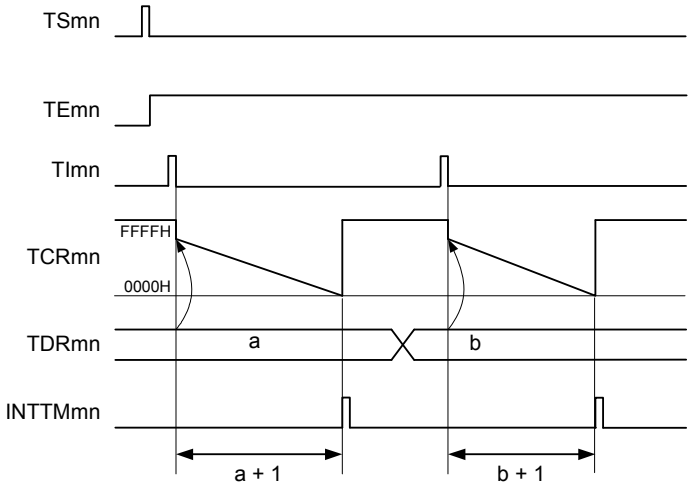
Figure 7 - 59 Block Diagram of Operation as Delay Counter



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 = 3)

Figure 7 - 60 Example of Basic Timing of Operation as Delay Counter

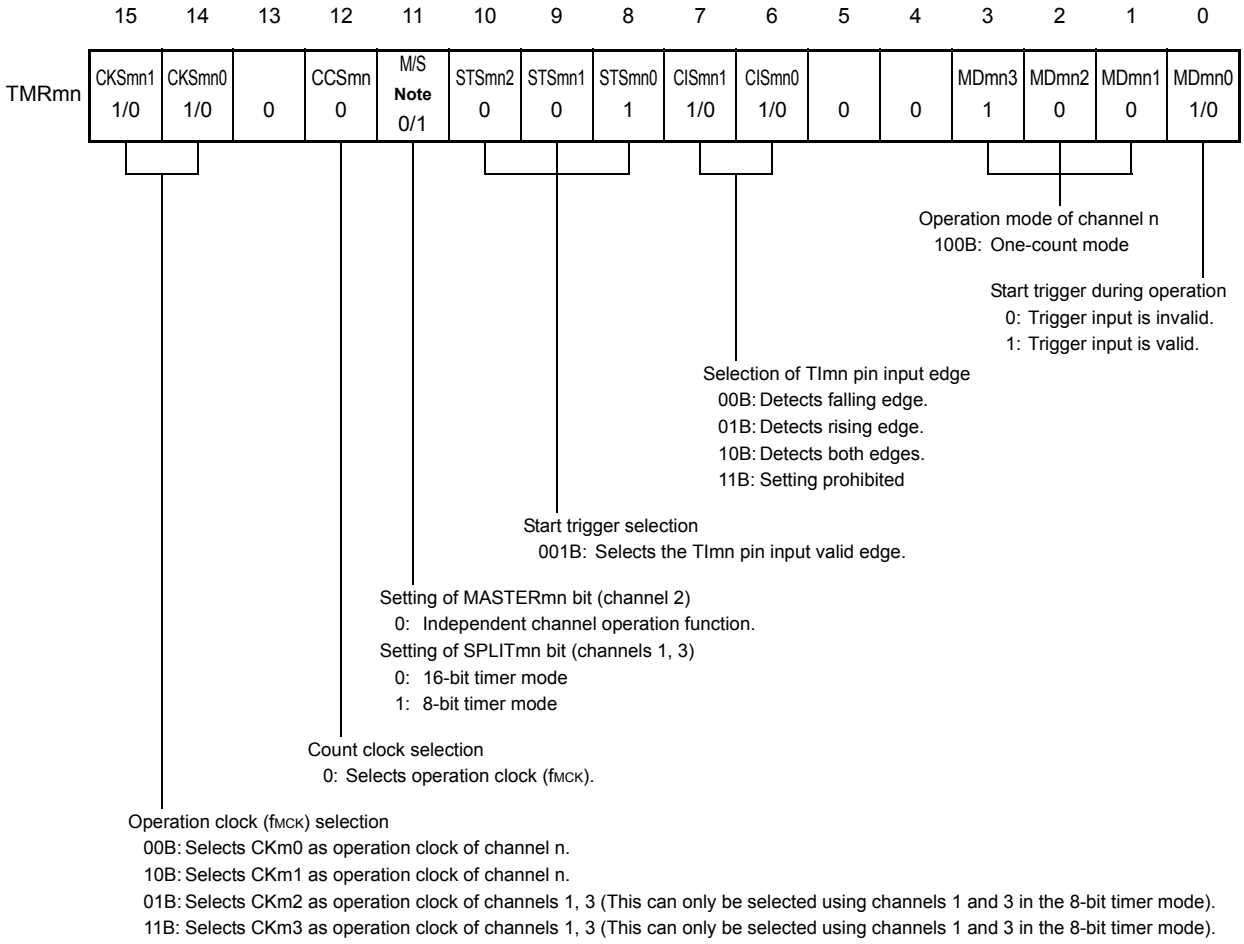


Remark 1. m: Unit number (m = 0), n: Channel number (n = 3)

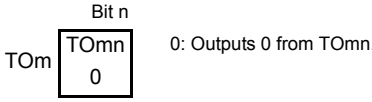
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 7 - 61 Example of Set Contents of Registers to Delay Counter

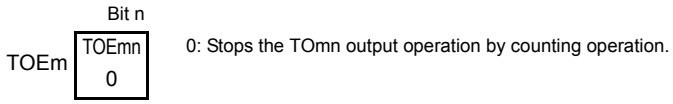
(a) Timer mode register mn (TMRmn)



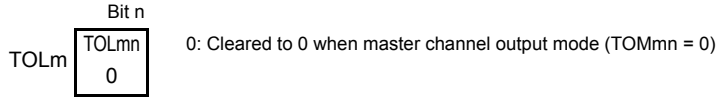
(b) Timer output register m (TOm)



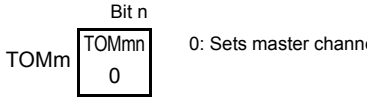
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 3)

Figure 7 - 62 Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. 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 registers 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.
	The counter starts counting down by the next start trigger detection. • Detects the TImn pin input valid edge. • Sets the TSmn bit to 1 by the software. →	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. →	Input clock supply for timer array unit m is stopped 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 = 3)

7.9 Simultaneous Channel Operation Function of Timer Array Unit

7.9.1 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}

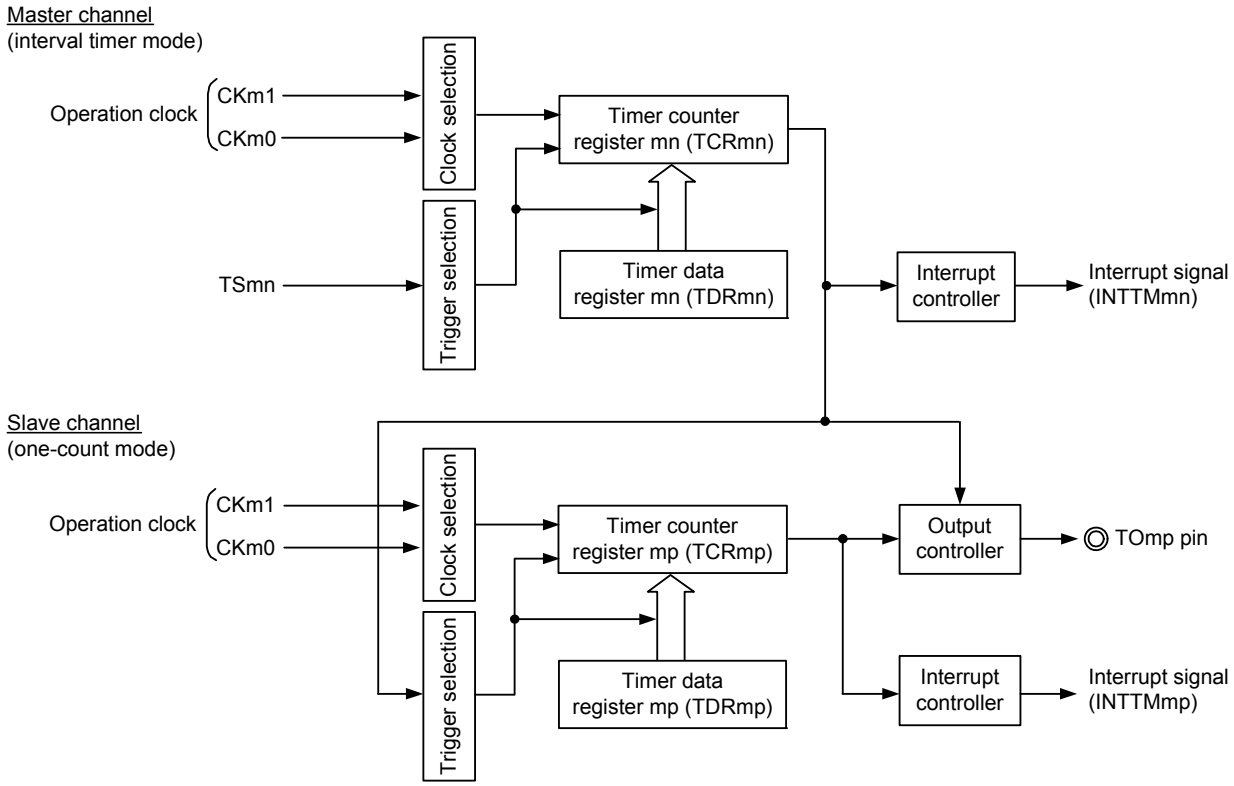
Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

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.**

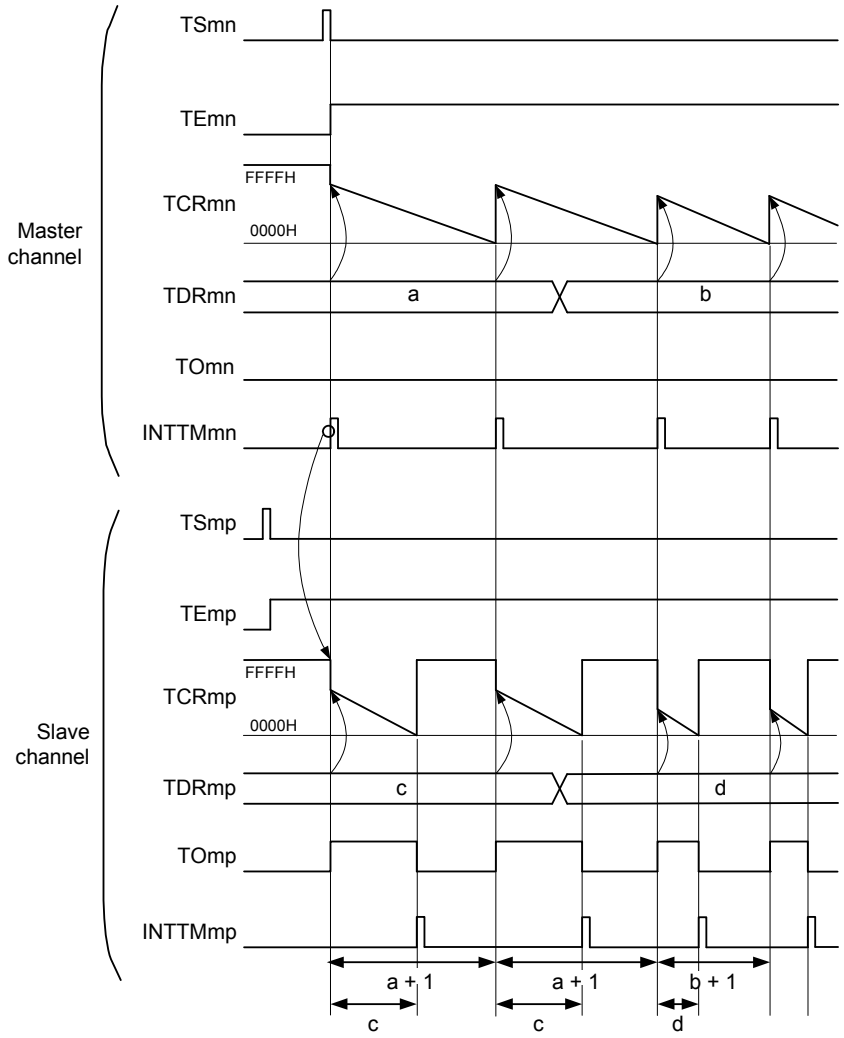
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 7 - 63 Block Diagram of Operation as PWM Function



Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 7 - 64 Example of Basic Timing of Operation as PWM Function

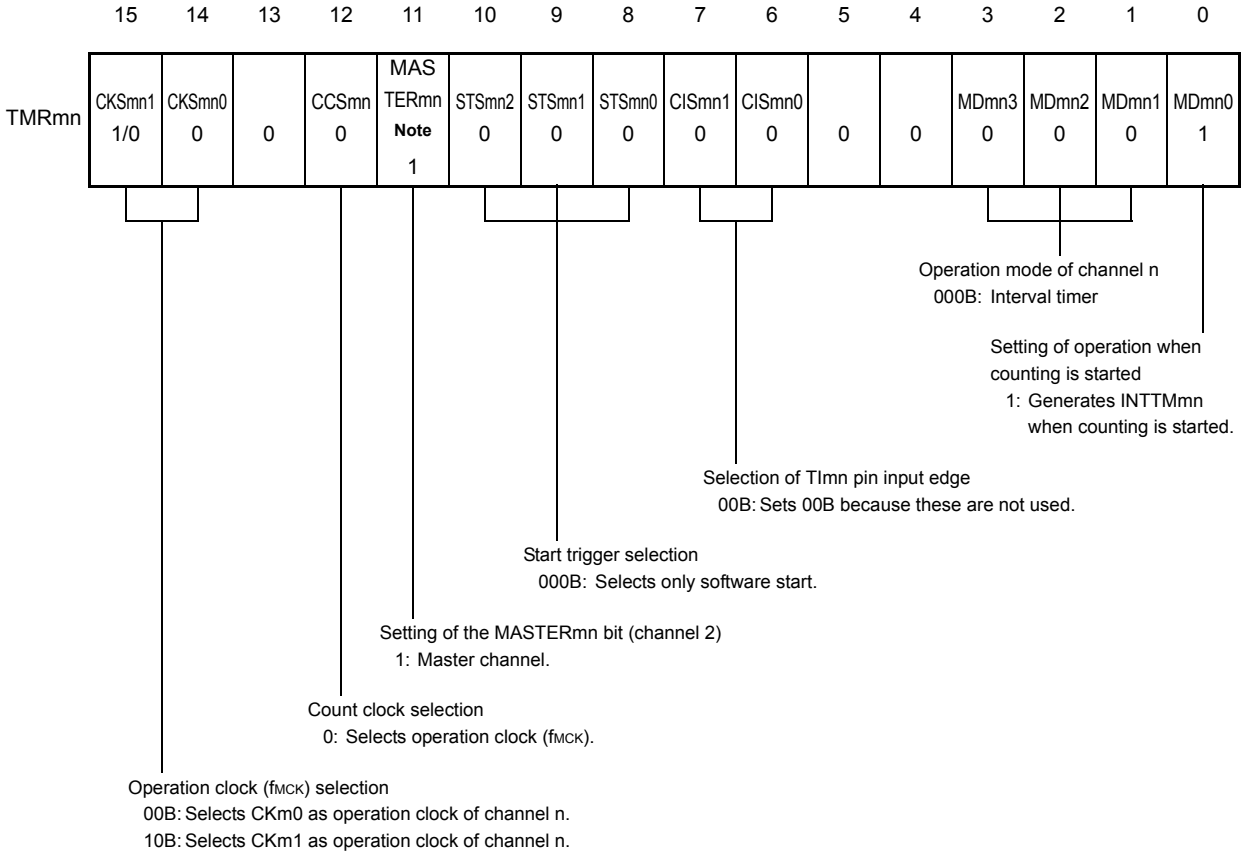


Remark 1. m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

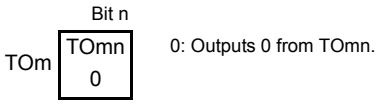
Remark 2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 7 - 65 Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

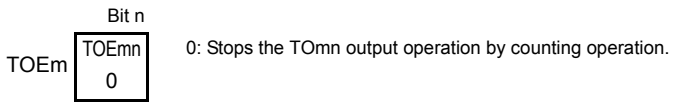
(a) Timer mode register mn (TMRmn)



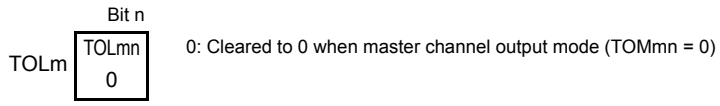
(b) Timer output register m (TOm)



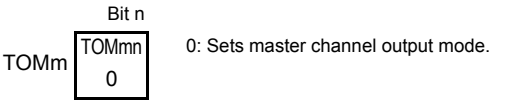
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)

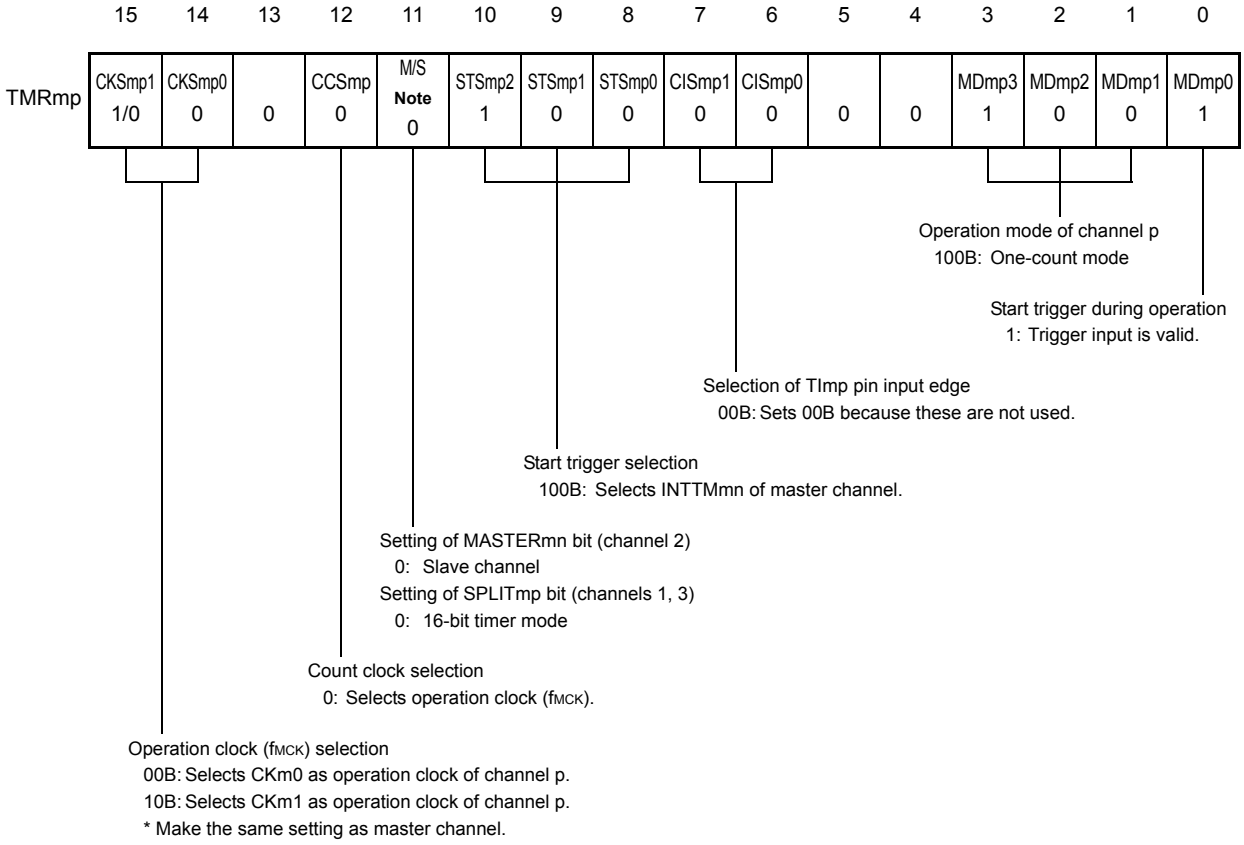


Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

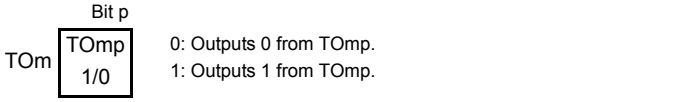
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)

Figure 7 - 66 Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

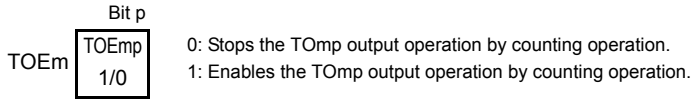
(a) Timer mode register mp (TMRmp)



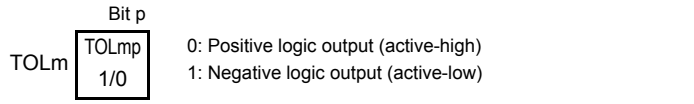
(b) Timer output register m (TOm)



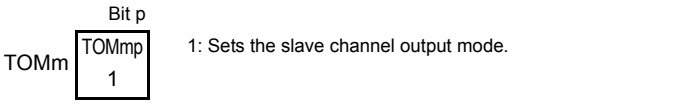
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2: MASTERmp bit
TMRm1, TMRm3: SPLITmp bit

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 7 - 67 Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. 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 7 - 68 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. →	Input clock supply for timer array unit m is stopped 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: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

7.10 Cautions When Using Timer Array Unit

7.10.1 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 **5.5 Register Settings When Using Alternate Function**.

CHAPTER 8 TIMER RJ

8.1 Functions of Timer RJ

Timer RJ is a 16-bit timer.

This 16-bit timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and they can be accessed by accessing the TRJ0 register.

Table 8 - 1 lists the Timer RJ Specifications. Figure 8 - 1 shows the Timer RJ Block Diagram.

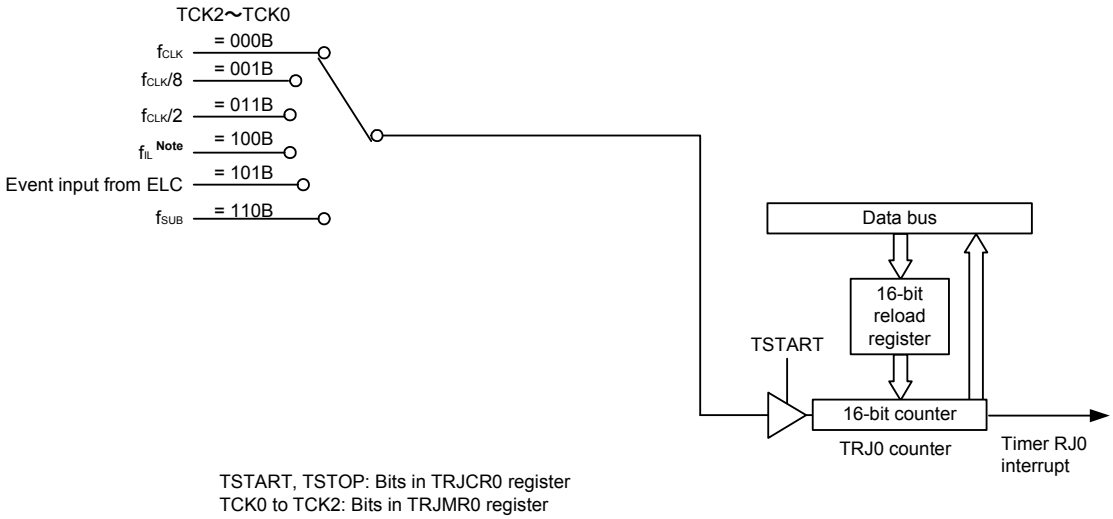
Table 8 - 1 Timer RJ Specifications

Item		Description
Operating modes	Timer mode	The count source is counted.
Count source (Operating clock)		f_{CLK}, f_{CLK}/2, f_{CLK}/8, f_{IL}, f_{SUB}, or event input from the event link controller (ELC) selectable
Interrupt		When the counter underflows.
Selectable functions		• Coordination with the event link controller (ELC). Event input from the ELC is selectable as a count source.

8.2 Configuration of Timer RJ

Figure 8 - 1 shows the Timer RJ Block Diagram.

Figure 8 - 1 Timer RJ Block Diagram



Note When selecting fIL as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1. However, fIL cannot be selected as the count source for timer RJ when fSUB is selected as the count source for the real-time clock or the 12-bit interval timer.

8.3 Registers Controlling Timer RJ

Table 8 - 2 lists the Registers Controlling Timer RJ.

Table 8 - 2 Registers Controlling Timer RJ

Register Name	Symbol
Peripheral enable register 1	PER1
Subsystem clock supply mode control register	OSMC
Timer RJ counter register 0 **Note**	TRJ0
Timer RJ control register 0	TRJCR0
Timer RJ mode register 0	TRJMR0

Note When the TRJ0 register is accessed, the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to the TRJ0 register is one clock for both writing and reading.

8.3.1 Peripheral enable register 1 (PER1)

The PER1 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 Timer RJ, be sure to set bit 0 (TRJ0EN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 8 - 2 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	4	<3>	2	1	<0>
PER1	0	0	0	0	DTCEN	0	0	TRJ0EN

TRJ0EN	Control of timer RJ0 input clock supply
0	Stops input clock supply. • SFR used by timer RJ0 cannot be written. • Timer RJ0 is in the reset status.
1	Enables input clock supply. • SFR used by timer RJ0 can be read and written.

Caution 1. When setting timer RJ, be sure to set the TRJ0EN bit to 1 first. If TRJ0EN = 0, writing to a control register of timer RJ is ignored, and all read values are default values.

Caution 2. Be sure to set the bits 1, 2, 4 to 7 to 0:

8.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the timer RJ operation clock.

In addition, by stopping clock functions that are unnecessary, the RTCLPC bit can be used to reduce power consumption. For details about setting the RTCLPC bit, see **CHAPTER 6 CLOCK GENERATOR**.

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

Reset signal generation clears this register to 00H.

Figure 8 - 3 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	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of operation clock (f_{RTC}) for real-time clock, 12-bit interval timer, and timer RJ
0	Subsystem clock (f_{SUB}) • The subsystem clock is selected as the operation clock for the real-time clock and the 12-bit interval timer. • The low-speed on-chip oscillator cannot be selected as the count source for timer RJ.
1	Low-speed on-chip oscillator clock (f_{IL}) • The low-speed on-chip oscillator clock is selected as the operation clock for the real-time clock and the 12-bit interval timer. • Either the low-speed on-chip oscillator or the subsystem clock can be selected as the count source for timer RJ.

8.3.3 Timer RJ counter register 0 (TRJ0)

TRJ0 is a 16-bit register. The write value is written to the reload register and the read value is read from the counter.

The states of the reload register and the counter are changed depending on the TSTART bit in the TRJCR0 register. For details, see **8.4.1 Reload Register and Counter Rewrite Operation**.

The TRJ0 register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to FFFFH.

Figure 8 - 4 Format of Timer RJ counter register 0 (TRJ0)

Address: F0500H After Reset: FFFFH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRJ0																

—	Function	Setting Range
Bits 15 to 0	16-bit counter Notes 1, 2	0000H to FFFFH

Note 1. When 1 is written to the TSTOP bit in the TRJCR0 register, the 16-bit counter is forcibly stopped and set to FFFFH.

Note 2. When the setting of bits TCK2 to TCK0 in the TRJMR0 register is other than 001B (f_{CLK}/8) or 011B (f_{CLK}/2), if the TRJ0 register is set to 0000H, a request signal to the DTC and the ELC is generated only once immediately after the count starts. starts.
When the TRJ0 register is set to 0000H or a higher value, a request signal is generated each time TRJ underflows.

Caution When the TRJ0 register is accessed, the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to the TRJ0 register is one clock for both writing and reading.

8.3.4 Timer RJ control register 0 (TRJCR0)

The TRJCR0 register starts or stops count operation and indicates the status of timer RJ.

The TRJCR0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8 - 5 Format of Timer RJ control register 0 (TRJCR0)

Address: F0240H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJCR0	0	0	TUNDF	0	0	TSTOP	TCSTF	TSTART

TUNDF	Timer RJ underflow flag
0	No underflow
1	Underflow
[Condition for setting to 0] • When 0 is written to this bit by a program. [Condition for setting to 1] • When the counter underflows.	

TSTOP	Timer RJ count forced stop Note 1
When 1 is written to this bit, the count is forcibly stopped. The read value is 0.	

TCSTF	Timer RJ count status flag Note 2
0	Count stops
1	Count in progress
[Conditions for setting to 0] • When 0 is written to the TSTART bit (the TCSTF bit is set to 0 in synchronization with the count source). • When 1 is written to the TSTOP bit. [Condition for setting to 1] • When 1 is written to the TSTART bit (the TCSTF bit is set to 1 in synchronization with the count source).	

TSTART	Timer RJ count start Note 2
0	Count stops
1	Count starts
Count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When the TSTART bit is set to 1 (count starts), the TCSTF bit is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF bit is set to 0 (count stops) in synchronization with the count source. For details, see **8.5.1 Count Operation Start and Stop Control**.	

Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, bits TSTART and TCSTF are initialized at the same time. The pulse output level is also initialized.

Note 2. For notes on using bits TSTART and TCSTF, see **8.5.1 Count Operation Start and Stop Control**.

8.3.5 Timer RJ mode register 0 (TRJMR0)

The TRJMR0 register sets the operating mode of timer RJ.

The TRJMR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8 - 6 Format of Timer RJ mode register 0 (TRJMR0)

Address: F0242H After Reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJMR0	0	TCK2	TCK1	TCK0	0	TMOD2	TMOD1	TMOD0

TCK2	TCK1	TCK0	Timer RJ count source select Note 1
0	0	0	f_{CLK}
0	0	1	$f_{CLK}/8$
0	1	1	$f_{CLK}/2$
1	0	0	f_{IL} Note 2
1	0	1	Event input from ELC
1	1	0	f_{SUB}
Other than above			Setting prohibited

TMOD2	TMOD1	TMOD0	Timer RJ operating mode select
0	0	0	Timer mode
Other than above			Setting prohibited

Note 1. Do not switch count sources during count operation. Count sources should be switched when both the TSTART and TCSTF bits in the TRJCR0 register are set to 0 (count stops).

Note 2. When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1.
However, f_{IL} cannot be selected as the count source for timer RJ when f_{SUB} is selected as the count source for the real-time clock or the 12-bit interval timer.

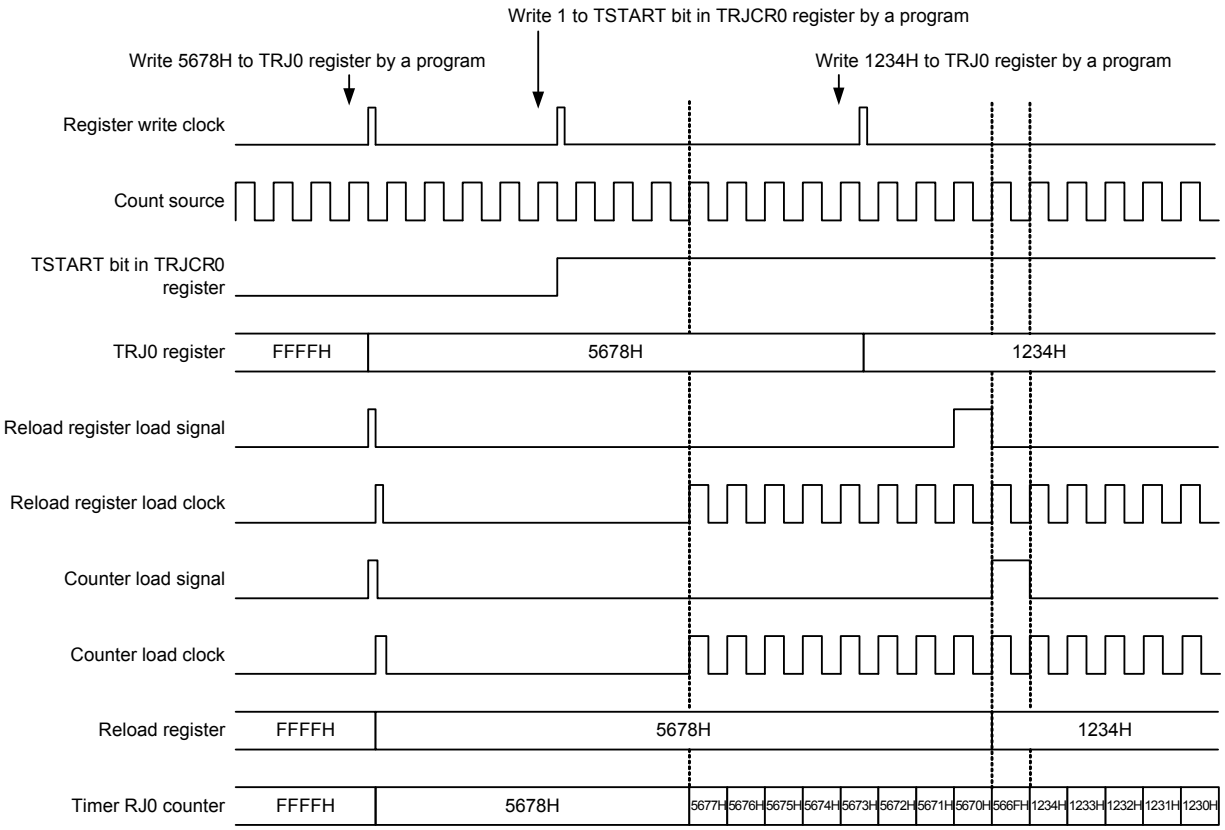
8.4 Timer RJ Operation

8.4.1 Reload Register and Counter Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value in the TSTART bit in the TRJCR0 register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source.

Figure 8 - 7 shows the Timing of Rewrite Operation with TSTART Bit Value.

Figure 8 - 7 Timing of Rewrite Operation with TSTART Bit Value



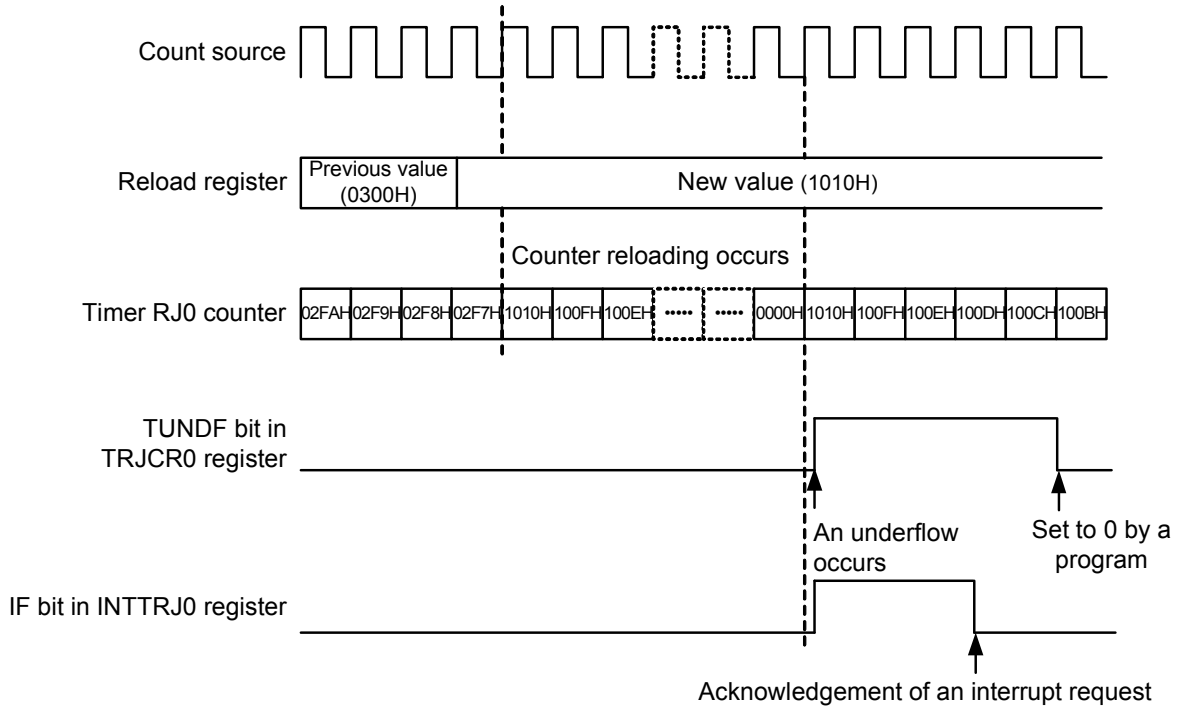
8.4.2 Timer Mode

In this mode, the counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register.

In timer mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 8 - 8 shows the Operation Example in Timer Mode.

Figure 8 - 8 Operation Example in Timer Mode



8.4.3 Coordination with Event Link Controller (ELC)

Through coordination with the ELC, event input from the ELC can be set to be the count source. Bits TCK0 to TCK2 in the TRJMR0 register count at the rising edge of event input from the ELC. However, ELC input does not function in event counter mode.

The ELC setting procedure is shown below:

- Procedure for starting operation
  (1) Set the event output destination select register (ELSELRn) for the ELC.
  (2) Set the operating mode for the event generation source.
  (3) Set the mode for timer RJ.
  (4) Start the count operation of timer RJ.
  (5) Start the operation of the event generation source.
- Procedure for stopping operation
  (1) Stop the operation of the event generation source.
  (2) Stop the count operation of timer RJ.
  (3) Set the event output destination select register (ELSELRn) for the ELC to 0.

8.5 Cautions for Timer RJ

8.5.1 Count Operation Start and Stop Control

• When the count source is set to other than the ELC

After 1 (count starts) is written to the TSTART bit in the TRJCR0 register while the count is stopped, the TCSTF bit in the TRJCR0 register remains 0 (count stops) for three cycles of the count source. Do not access the registers associated with timer RJ **Note** other than the TCSTF bit until this bit is set to 1 (count in progress).

After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for three cycles of the count source. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with timer RJ Note other than the TCSTF bit until this bit is set to 0.

Clear the interrupt register before changing the TATART bit from 0 to 1. Refer to **CHAPTER 19 INTERRUPT FUNCTIONS** for details.

Note Registers associated with timer RJ: TRJ0, TRJCR0, and TRJMR0

• When the count source is set to the ELC

After 1 (count starts) is written to the TSTART bit in the TRJCR0 register while the count is stopped, the TCSTF bit in the TRJCR0 register remains 0 (count stops) for two cycles of the CPU clock. Do not access the registers associated with timer RJ **Note** other than the TCSTF bit until this bit is set to 1 (count in progress).

After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for two cycles of the CPU clock. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with timer RJ Note other than the TCSTF bit until this bit is set to 0.

Clear the interrupt register before changing the TATART bit from 0 to 1. Refer to **CHAPTER 19 INTERRUPT FUNCTIONS** for details.

Note Registers associated with timer RJ: TRJ0, TRJCR0, and TRJMR0

8.5.2 Access to Flags (TUNDF bit in TRJCR0 Register)

Bit TUNDF in the TRJCR0 register are set to 0 by writing 0 by a program, but writing 1 to these bits has no effect. If a read-modify-write instruction is used to set the TRJCR0 register, bit TUNDF may be erroneously set to 0 depending on the timing, even when the TUNDF bit is set to 1 (underflow) during execution of the instruction. Use an 8-bit memory manipulation instruction to access to the TRJCR0 register.

8.5.3 Access to Counter Register

When bits TSTART and TCSTF in the TRJCR0 register are both 1 (count starts), allow at least three cycles of the count source clock between writes when writing to the TRJ0 register successively.

8.5.4 When Changing Mode

The register associated with timer RJ operating mode (TRJMR0) can be changed only

when the count is stopped with both the TSTART and TCSTF bits set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with timer RJ operating mode are changed, the value of TUNDF bit is undefined. Write 0 (no underflow) to the TUNDF bit before starting the count.

8.5.5 When Timer RJ Operating Clock is Stopped

Supplying or stopping the timer RJ clock can be controlled by the TRJ0EN bit in the PER1 register. Note that the following SFRs cannot be accessed while the timer RJ clock is stopped. Make sure the timer RJ clock is supplied before accessing any of these registers.
Registers TRJ0, TRJCR0, and TRJMR0.

8.5.6 When Count is Forcibly Stopped by TSTOP Bit

After the counter is forcibly stopped by the TSTOP bit in the TRJCR0 register, do not access the following SFRs for one cycle of the count source.
Registers TRJ0, TRJCR0, and TRJMR0

8.5.7 When Selecting f_{IL} as Count Source

When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1. However, f_{IL} cannot be selected as the count source for timer RJ when f_{SUB} is selected as the count source for the real-time clock or the 12-bit interval timer.

CHAPTER 9 REAL-TIME CLOCK

9.1 Functions of Real-time Clock

The real-time clock has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: week, hour, minute)

The real-time clock interrupt signal (INTRTC) can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

Caution The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed oscillation clock (f_{IL} = 15kHz) is selected, only the constant-period interrupt function is available.
However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

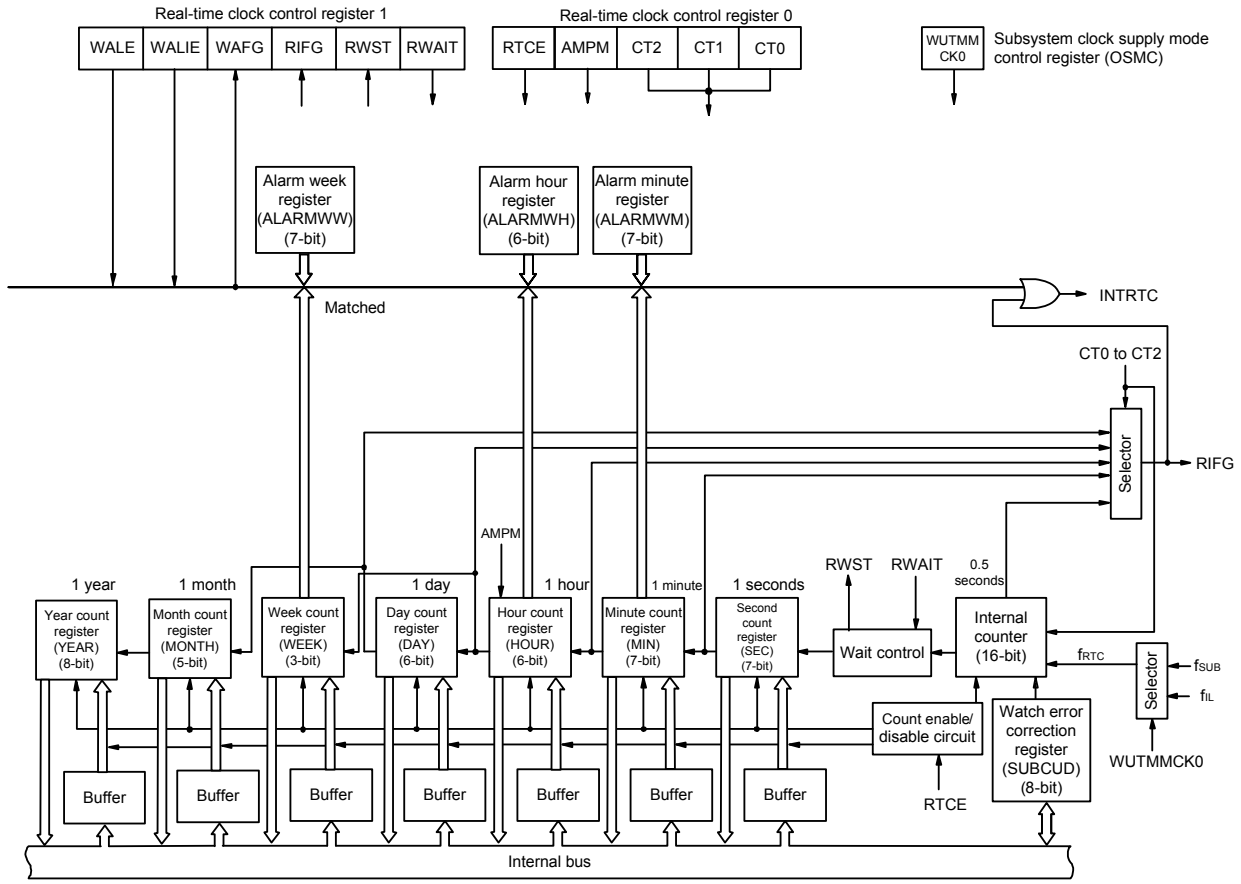
9.2 Configuration of Real-time Clock

The real-time clock includes the following hardware.

Table 9 - 1 Configuration of Real-time Clock

Item	Configuration
Counter	Internal counter (16-bit)
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	Real-time clock control register 0 (RTCC0)
	Real-time clock control register 1 (RTCC1)
	Second count register (SEC)
	Minute count register (MIN)
	Hour count register (HOUR)
	Day count register (DAY)
	Week count register (WEEK)
	Month count register (MONTH)
	Year count register (YEAR)
	Watch error correction register (SUBCUD)
	Alarm minute register (ALARMWM)
	Alarm hour register (ALARMWH)
	Alarm week register (ALARMWW)

Figure 9 - 1 Block Diagram of Real-time Clock



Caution The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock.
When the low-speed oscillation clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available.
The constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

9.3 Registers Controlling Real-time Clock

The real-time clock is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Real-time clock control register 0 (RTCC0)
- Real-time clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Watch error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)

9.3.1 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 real-time clock is used, be sure to set bit 7 (RTCEN) 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 9 - 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	RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of real-time clock (RTC) and 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. • The real-time clock (RTC) and 12-bit interval timer are in the reset status.
1	Input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer can be read/written.

Caution 1. When using the real-time clock, first set the RTCEN bit to 1 and then set the following registers, while oscillation of the count clock (f_{RTC}) is stable. If RTCEN = 0, writing to the control registers of the real-time clock is ignored, and, even if the registers are read, only the default values are read (except for the subsystem clock supply mode control register (OSMC), port mode register 3 (PM3), port register 3 (P3)).

- **Real-time clock control register 0 (RTCC0)**
- **Real-time clock control register 1 (RTCC1)**
- **Second count register (SEC)**
- **Minute count register (MIN)**
- **Hour count register (HOUR)**
- **Day count register (DAY)**
- **Week count register (WEEK)**
- **Month count register (MONTH)**
- **Year count register (YEAR)**
- **Watch error correction register (SUBCUD)**
- **Alarm minute register (ALARMWM)**
- **Alarm hour register (ALARMWH)**
- **Alarm week register (ALARMWW)**

Caution 2. Subsystem clock supply to peripheral functions other than the real-time clock and 12-bit interval timer can be stopped in STOP mode or HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1.

9.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the real-time clock count clock (f_{RTC}).

In addition, by stopping clock functions that are unnecessary, the RTCLPC bit can be used to reduce power consumption. For details about setting the RTCLPC bit, see **CHAPTER 6 CLOCK GENERATOR**.

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

Reset signal generation clears this register to 00H.

Figure 9 - 3 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	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of count clock (f_{RTC}) for real-time clock, 12-bit interval timer, and timer RJ operation clock
0	Subsystem clock (f_{SUB}) • The subsystem clock is selected as the count clock for the real-time clock and the 12-bit interval timer. • The low-speed on-chip oscillator cannot be selected as the count source for timer RJ.
1	Low-speed on-chip oscillator clock (f_{IL}) • The low-speed on-chip oscillator clock is selected as the count clock for the real-time clock and the 12-bit interval timer. • Either the low-speed on-chip oscillator or the subsystem clock can be selected as the count source for timer RJ.

Caution The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed oscillation clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available.

The constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

9.3.3 Real-time clock control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time clock operation, and set a 12- or 24-hour system and the constant-period interrupt function.

The RTCC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9 - 4 Format of Real-time clock control register 0 (RTCC0)

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
RTCC0	RTCE	0	0	0	AMPM	CT2	CT1	CT0

RTCE	Real-time clock operation control
0	Stops counter operation.
1	Starts counter operation.

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• Rewrite the AMPM bit value after setting the RWAIT bit (bit 0 of real-time clock control register 1 (RTCC1)) to 1. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system. • Table 9 - 2 shows the Displayed Time Digits.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use fixed-cycle interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
When changing the values of the CT2 to CT0 bits while the counter operates (RTCE = 1), rewrite the values of the CT2 to CT0 bits after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT2 to CT0 bits, enable interrupt servicing after clearing the RIFG and RTCIF flags.			

Remark ×: Don't care

9.3.4 Real-time clock control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

The RTCC1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9 - 5 Format of Real-time clock control register 1 (RTCC1) (1/2)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of real-time clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm week register (ALARMWW)), set match operation to be invalid (“0”) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one cycle of f_{RTC} after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

Figure 9 - 6 Format of Real-time clock control register 1 (RTCC1) (2/2)

RIFG	Constant-period interrupt status flag
0	Fixed-cycle interrupt is not generated.
1	Fixed-cycle interrupt is generated.
This flag indicates the status of generation of the fixed-cycle interrupt. When the fixed-cycle interrupt is generated, it is set to “1”. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

RWST	Wait status flag of real-time clock
0	Counter is operating.
1	Mode to read or write counter value
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

RWAIT	Wait control of real-time clock
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value
This bit controls the operation of the counter. Be sure to write “1” to it to read or write the counter value. As the internal counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0. When RWAIT = 1, it takes up to one cycle of f_{RTC} until the counter value can be read or written (RWST = 1). When the internal counter (16-bit) overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up. However, when it wrote a value to second count register, it will not keep the overflow event.	

Caution 1. If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG flag and WAFG flag may be cleared. Therefore, to perform writing to the RTCC1 register, be sure to use an 8-bit manipulation instruction. To prevent the RIFG flag and WAFG flag from being cleared during writing, disable writing by setting 1 to the corresponding bit. If the RIFG flag and WAFG flag are not used and the value may be changed, the RTCC1 register may be written by using a 1-bit manipulation instruction.

Caution 2. RWAIT bit functions while RTCE bit is "1". When RTCE is set to 1 and RWAIT is set to 1 within one clock cycle of f_{RTC}, setting RTCE bit to "1" takes one clock cycle of f_{RTC}. Therefore, it takes up to two clock cycles of f_{RTC} until the counter value can be read or written (RWST = 1).

Caution 3. When RTCE is set to 1 and RWAIT is set to 1 within one clock cycle of f_{RTC} after returned from standby (HALT/STOP/SNOOZE) mode, it takes up to two clock cycles of f_{RTC} until the counter value can be read or written (RWST = 1).

Remark 1. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

Remark 2. The internal counter (16 bits) is cleared when the second count register (SEC) is written.

9.3.5 Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the internal counter (16-bit) overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later.

Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 9 - 7 Format of Second count register (SEC)

Address: FFF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 9.4.3 Reading/writing real-time clock.

Remark The internal counter (16 bits) is cleared when the second count register (SEC) is written.

9.3.6 Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later.

Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 9 - 8 Format of Minute count register (MIN)

Address: FFF93H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 9.4.3 Reading/writing real-time clock.

9.3.7 Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later.

Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of real-time clock control register 0 (RTCC0).

If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Figure 9 - 9 Format of Hour count register (HOUR)

Address: FFF94H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

Caution 2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 9.4.3 Reading/writing real-time clock.

Table 9 - 2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 9 - 2 Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 0)	
Time	HOUR Register	Time	HOUR Register
0	00 H	12 a.m.	12 H
1	01 H	1 a.m.	01 H
2	02 H	2 a.m.	02 H
3	03 H	3 a.m.	03 H
4	04 H	4 a.m.	04 H
5	05 H	5 a.m.	05 H
6	06 H	6 a.m.	06 H
7	07 H	7 a.m.	07 H
8	08 H	8 a.m.	08 H
9	09 H	9 a.m.	09 H
10	10 H	10 a.m.	10 H
11	11 H	11 a.m.	11 H
12	12 H	12 p.m.	32 H
13	13 H	1 p.m.	21 H
14	14 H	2 p.m.	22 H
15	15 H	3 p.m.	23 H
16	16 H	4 p.m.	24 H
17	17 H	5 p.m.	25 H
18	18 H	6 p.m.	26 H
19	19 H	7 p.m.	27 H
20	20 H	8 p.m.	28 H
21	21 H	9 p.m.	29 H
22	22 H	10 p.m.	30 H
23	23 H	11 p.m.	31 H

The HOUR register value is set to 12-hour display when the AMPM bit is “0” and to 24-hour display when the AMPM bit is “1”.

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

9.3.8 Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days. It counts up when the hour counter overflows.
This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code.
The DAY register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 01H.

Figure 9 - 10 Format of Day count register (DAY)

Address: FFF96H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 9.4.3 Reading/writing real-time clock.

9.3.9 Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later.

Set a decimal value of 00 to 06 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 9 - 11 Format of Week count register (WEEK)

Address: FFF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00 H
Monday	01 H
Tuesday	02 H
Wednesday	03 H
Thursday	04 H
Friday	05 H
Saturday	06 H

Caution 2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 9.4.3 Reading/writing real-time clock.

9.3.10 Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later.

Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code.

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

Reset signal generation clears this register to 01H.

Figure 9 - 12 Format of Month count register (MONTH)

Address: FFF97H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 9.4.3 Reading/writing real-time clock.

9.3.11 Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month count register (MONTH) overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later.

Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 9 - 13 Format of Year count register (YEAR)

Address: FFF98H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 9.4.3 Reading/writing real-time clock.

9.3.12 Watch error correction register (SUBCUD)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the internal counter (16-bit) to the second count register (SEC) (reference value: 7FFFH).
The SUBCUD register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 9 - 14 Format of Watch error correction register (SUBCUD)

Address: FFF99H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).
Writing to the SUBCUD register at the following timing is prohibited. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • When DEV = 1 is set: For a period of SEC = 00H	

F6	Setting of watch error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.
When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100). Range of correction value: (when F6 = 0) 2, 4, 6, 8, ... , 120, 122, 124 (when F6 = 1) –2, –4, –6, –8, ... , –120, –122, –124	

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Remark If a correctable range is −63.1 ppm or lower and 63.1 ppm or higher, set 0 to DEV.

9.3.13 Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

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

Reset signal generation clears this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 9 - 15 Format of Alarm minute register (ALARMWM)

Address: FFF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

9.3.14 Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 9 - 16 Format of Alarm hour register (ALARMWH)

Address: FFF9BH After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

9.3.15 Alarm week register (ALARMWW)

This register is used to set date of alarm.

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

Reset signal generation clears this register to 00H.

Figure 9 - 17 Format of Alarm week register (ALARMWW)

Address: FFF9CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

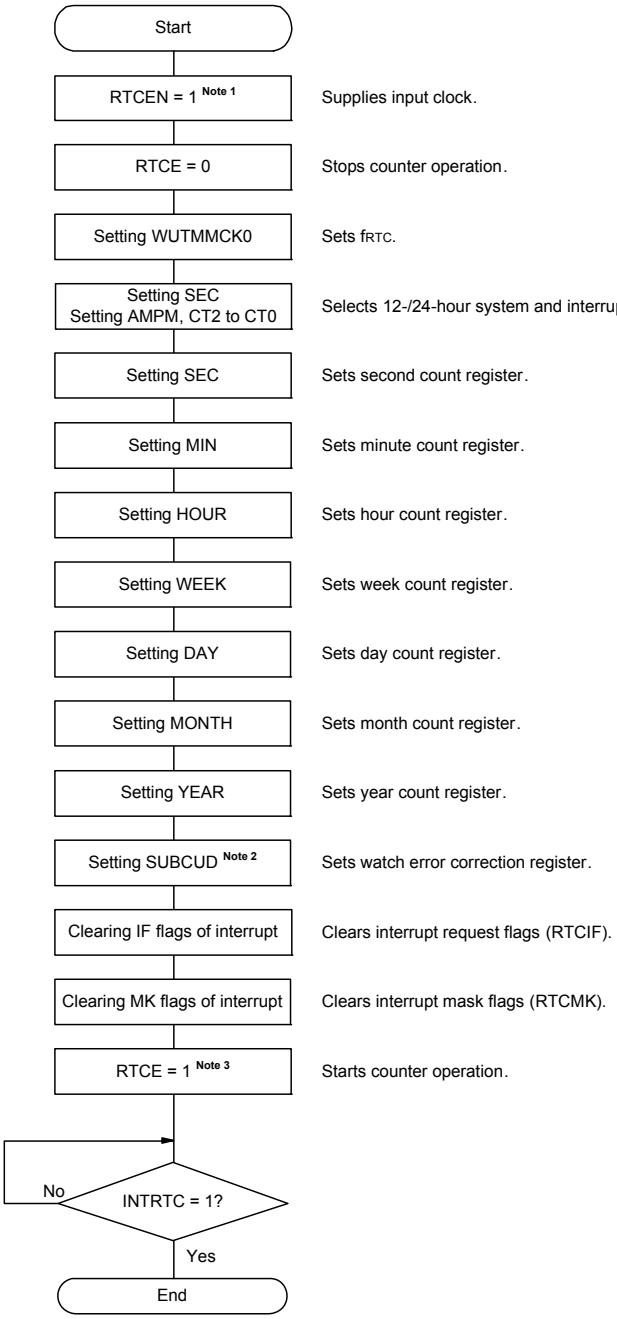
Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
	WW0	WW1	WW2	WW3	WW4	WW5	WW6								
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

9.4 Real-time Clock Operation

9.4.1 Starting operation of real-time clock

Figure 9 - 18 Procedure for Starting Operation of Real-time Clock



Note 1. First set the RTCEN bit to 1, while oscillation of the count clock (fRTC) is stable.

Note 2. Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **9.4.5 Example of watch error correction of real-time clock**.

Note 3. Confirm the procedure described in 9.4.2 Shifting to HALT/STOP mode after starting operation when shifting to HALT/STOP mode without waiting for INTRTC = 1 after RTCE = 1.

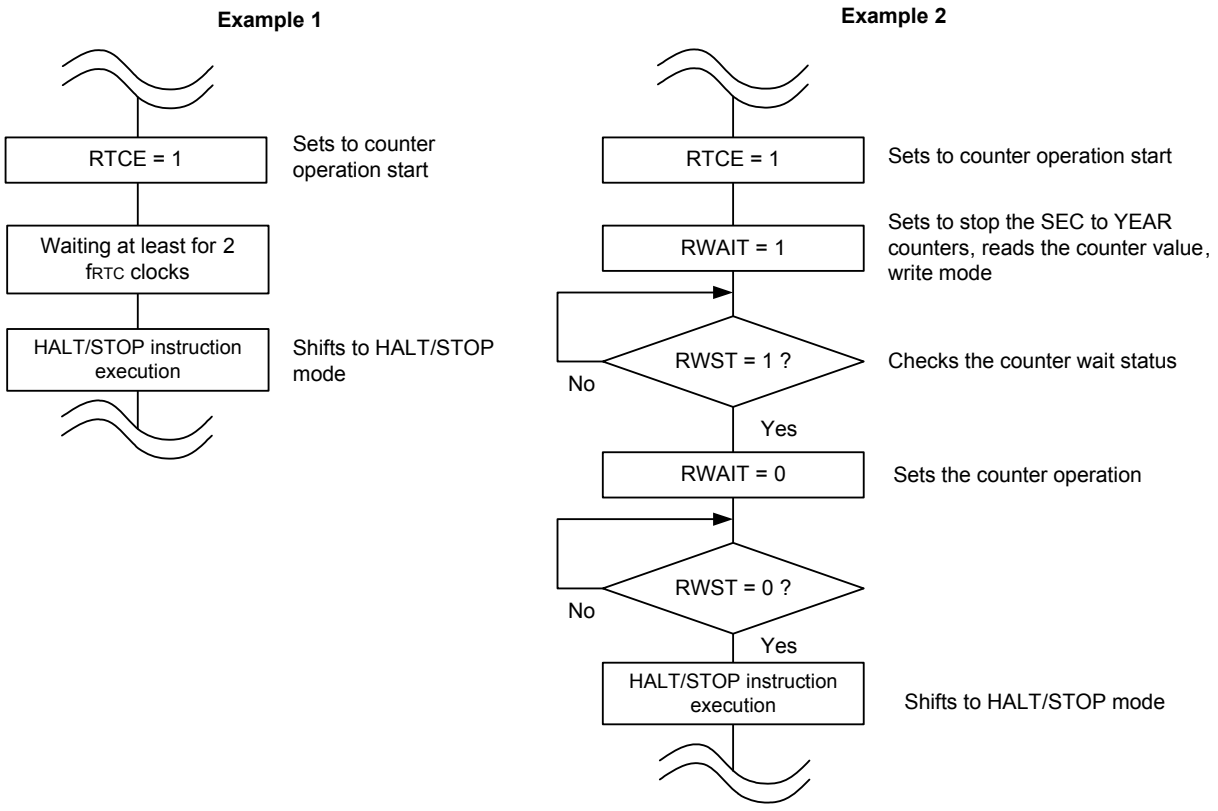
9.4.2 Shifting to HALT/STOP mode after starting operation

Perform one of the following processing when shifting to HALT/STOP mode immediately after setting the RTCE bit to 1.

However, after setting the RTCE bit to 1, this processing is not required when shifting to HALT/STOP mode after INTRTC interrupt has occurred.

- Shifting to HALT/STOP mode when at least two count clock (f_{RTC}) have elapsed after setting the RTCE bit to 1 (see **Figure 9 - 19**, **Example 1**).
- Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to HALT/STOP mode after checking again by polling that the RWST bit has become 0 (see **Figure 9 - 19**, **Example 2**).

Figure 9 - 19 Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1

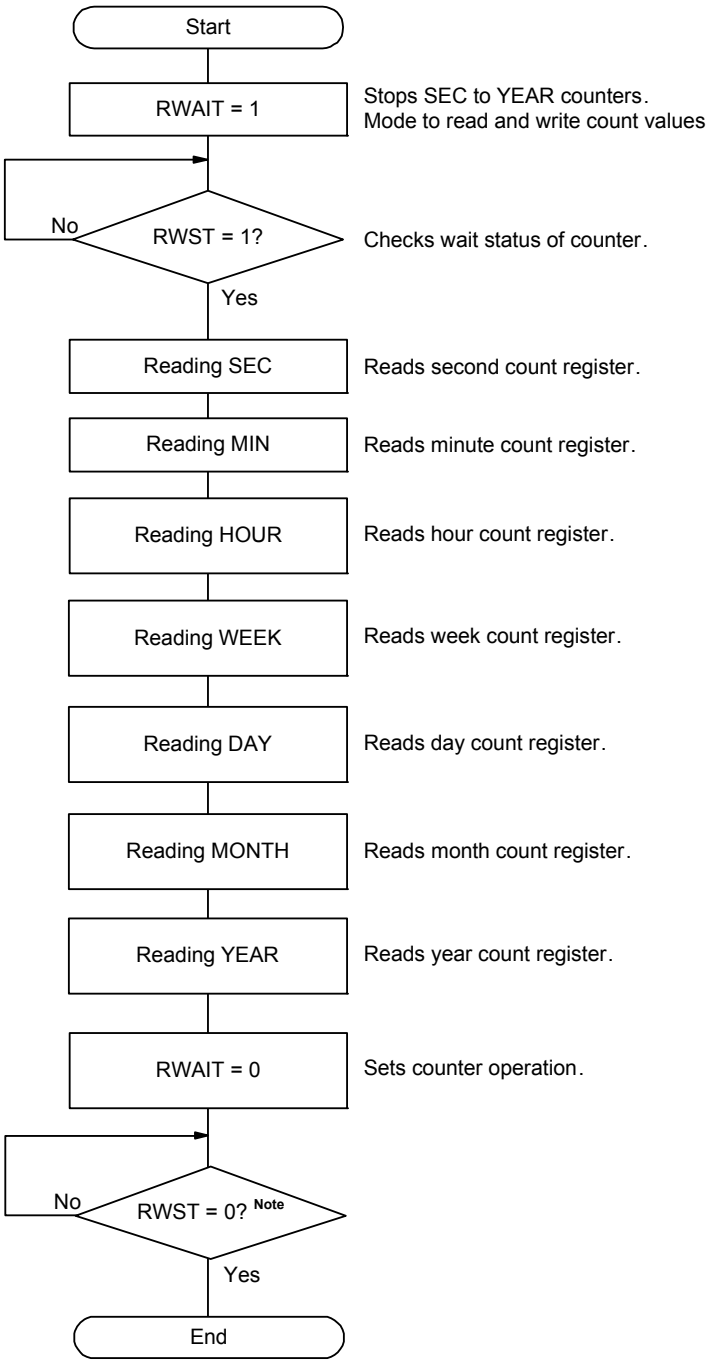


9.4.3 Reading/writing real-time clock

Read or write the counter after setting 1 to RWAIT first.

Set RWAIT to 0 after completion of reading or writing the counter.

Figure 9 - 20 Procedure for Reading Real-time Clock

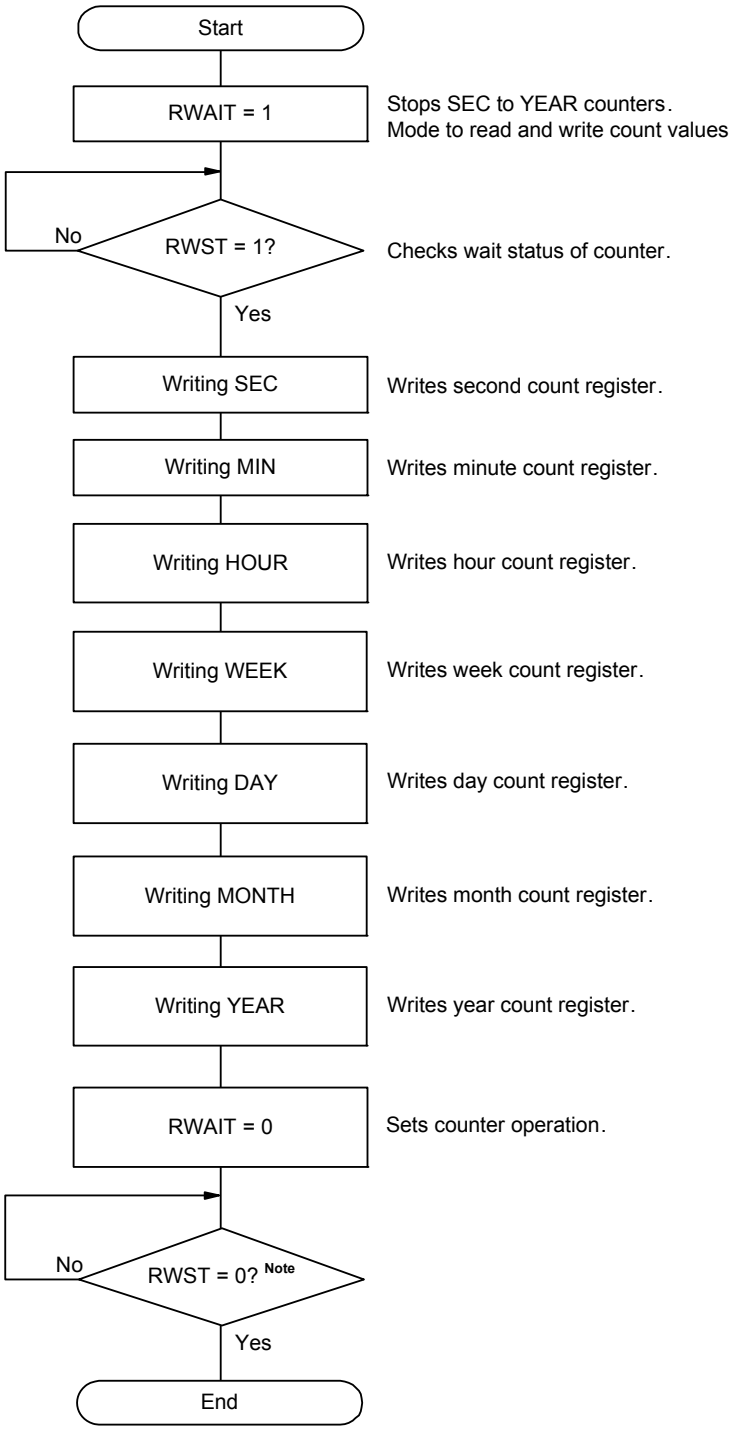


Note Be sure to confirm that RWST = 0 before setting HALT/STOP mode.

Caution **Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.**

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence. All the registers do not have to read and only some registers may be read.

Figure 9 - 21 Procedure for Writing Real-time Clock



Note Be sure to confirm that RWST = 0 before setting HALT/STOP mode.

Caution 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the MIN register.

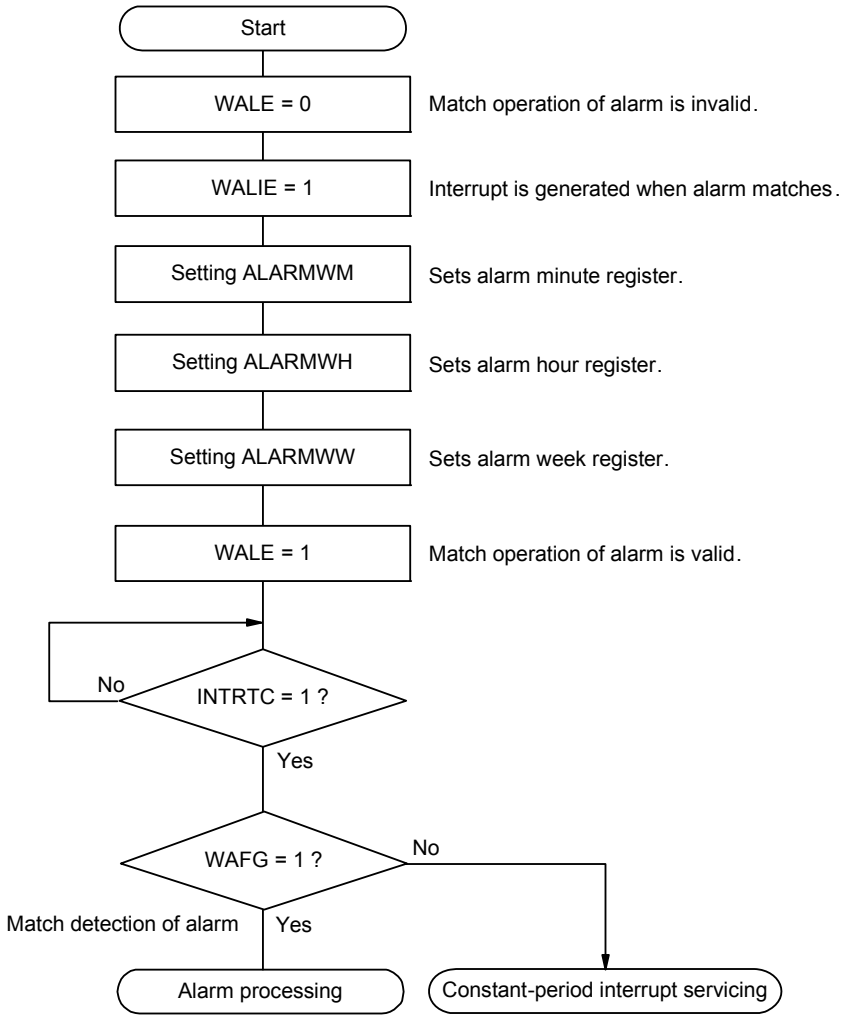
Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence.

All the registers do not have to be set and only some registers may be written.

9.4.4 Setting alarm of real-time clock

Set time of alarm after setting 0 to WALE (alarm operation invalid) first.

Figure 9 - 22 Alarm Setting Procedure



Remark 1. The alarm week register (ALARMWW), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

Remark 2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

9.4.5 Example of watch error correction of real-time clock

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the internal counter (16-bit) is calculated by using the following expression.

Set the DEV bit to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(When DEV = 0)

Correction value **Note** = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value Note = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

Note The correction value is the watch error correction value calculated by using bits 6 to 0 of the watch error correction register (SUBCUD).

(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2

(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), watch error correction is not performed. “*” is 0 or 1.

/F5 to /F0 are bit-inverted values (000011 when 111100).

Remark 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

Remark 2. The oscillation frequency is the count clock (f_{RTC}).

Remark 3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz - 131.2 ppm)

[Measuring the oscillation frequency]

The oscillation frequency **Note** of each product is measured by outputting about 32.768 kHz from the PCLBUZ0 pin.

Note See **11.4 Operations of Clock Output/Buzzer Output Controller** for the setting procedure of outputting about 32 kHz from the PCLBUZ0 pin.

[Calculating the correction value]

(When the output frequency from the PCLBUZ0 pin is 32772.3 Hz)

Assume the target frequency to be 32768 Hz (32772.3 Hz - 131.2 ppm) and DEV to be 0, because the correctable range of -131.2 ppm is -63.1 ppm or lower.

The expression for calculating the correction value when DEV is 0 is applied.

Correction value = Number of correction counts in 1 minute ÷ 3
= (Oscillation frequency ÷ target frequency - 1) × 32768 × 60 ÷ 3
= (32772.3 ÷ 32768 - 1) × 32768 × 60 ÷ 3
= 86

[Calculating the values to be set to (F6 to F0)]

(When the correction value is 86)

If the correction value is 0 or larger (when slowing), assume F6 to be 0.

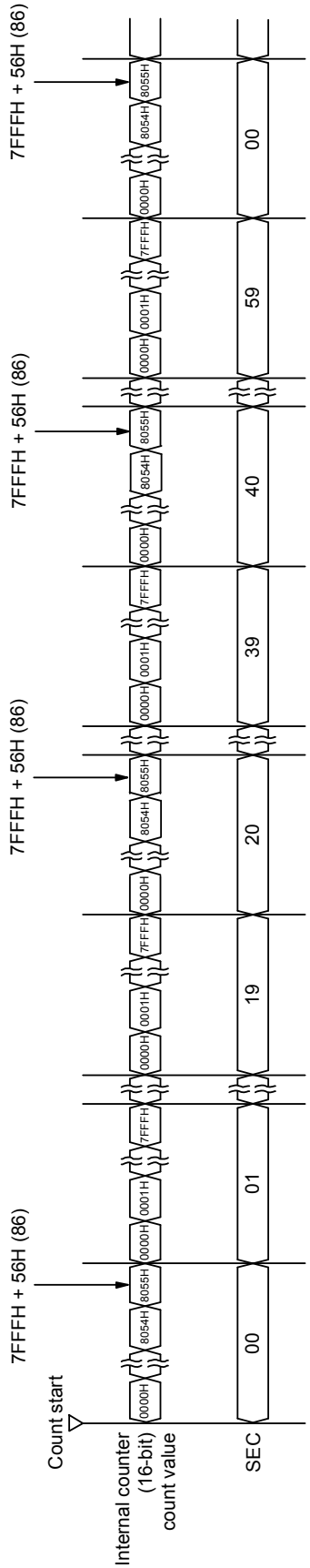
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

{(F5, F4, F3, F2, F1, F0) - 1} × 2 = 86
(F5, F4, F3, F2, F1, F0) = 44
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 0, 0)

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz - 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of the SUBCUD register: 0101100) results in 32768 Hz (0 ppm).

Figure 9 - 23 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 9 - 23 Correction Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



CHAPTER 10 12-BIT INTERVAL TIMER

10.1 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.

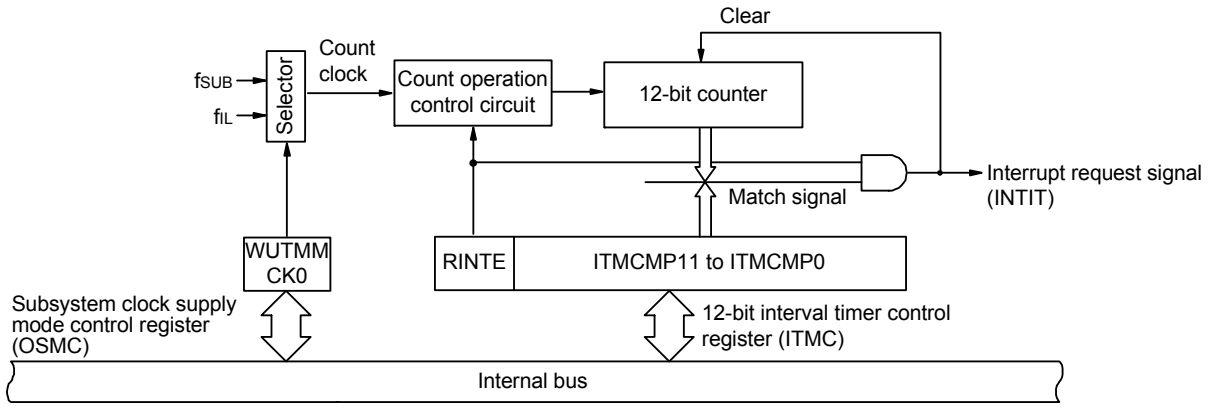
10.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 10 - 1 Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	12-bit interval timer control register (ITMC)

Figure 10 - 1 Block Diagram of 12-bit Interval Timer



10.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- 12-bit interval timer control register (ITMC)

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 12-bit interval timer is used, be sure to set bit 7 (RTCEN) 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	RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of real-time clock (RTC) and 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. • The real-time clock (RTC) and 12-bit interval timer are in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer can be read/written.

Caution 1. When using the 12-bit interval timer, be sure to first set the RTCEN bit to 1 and then set the following register, while oscillation of the count clock is stable. If RTCEN = 0, writing to the control register controlling the 12-bit interval timer is ignored, and, even if the register is read, only the default value is read (except the subsystem clock supply mode control register (OSMC)).
• 12-bit interval timer control register (ITMC)

Caution 2. Clock supply to peripheral functions other than the real-time clock and 12-bit interval timer can be stopped in STOP mode or HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1.

10.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the 12-bit interval timer, real-time clock, and timer RJ operation clock. In addition, by stopping clock functions that are unnecessary, the RTCLPC bit can be used to reduce power consumption. For details about setting the RTCLPC bit, see **CHAPTER 6 CLOCK GENERATOR**.

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

Reset signal generation clears this register to 00H.

Figure 10 - 3 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	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of operation clock for real-time clock, 12-bit interval timer, and timer RJ.
0	Subsystem clock (f_{SUB}) • The subsystem clock is selected as the operation clock for the real-time clock and the 12- bit interval timer. • The low-speed on-chip oscillator cannot be selected as the count source for timer RJ.
1	Low-speed on-chip oscillator clock (f_{IL}) • The low-speed on-chip oscillator clock is selected as the operation clock for the real-time clock and the 12-bit interval timer. • Either the low-speed on-chip oscillator or the subsystem clock can be selected as the count source for timer RJ.

10.3.3 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 10 - 4 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_{SUB} = 32.768 kHz 1/32.768 [kHz] × (1 + 1) = 0.06103515625 [ms] ≅ 61.03 [μs] • ITCMP11 to ITCMP0 = FFFH, count clock: when f_{SUB} = 32.768 kHz 1/32.768 [kHz] × (4095 + 1) = 125 [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.

10.4 12-bit Interval Timer Operation

10.4.1 12-bit interval timer operation timing

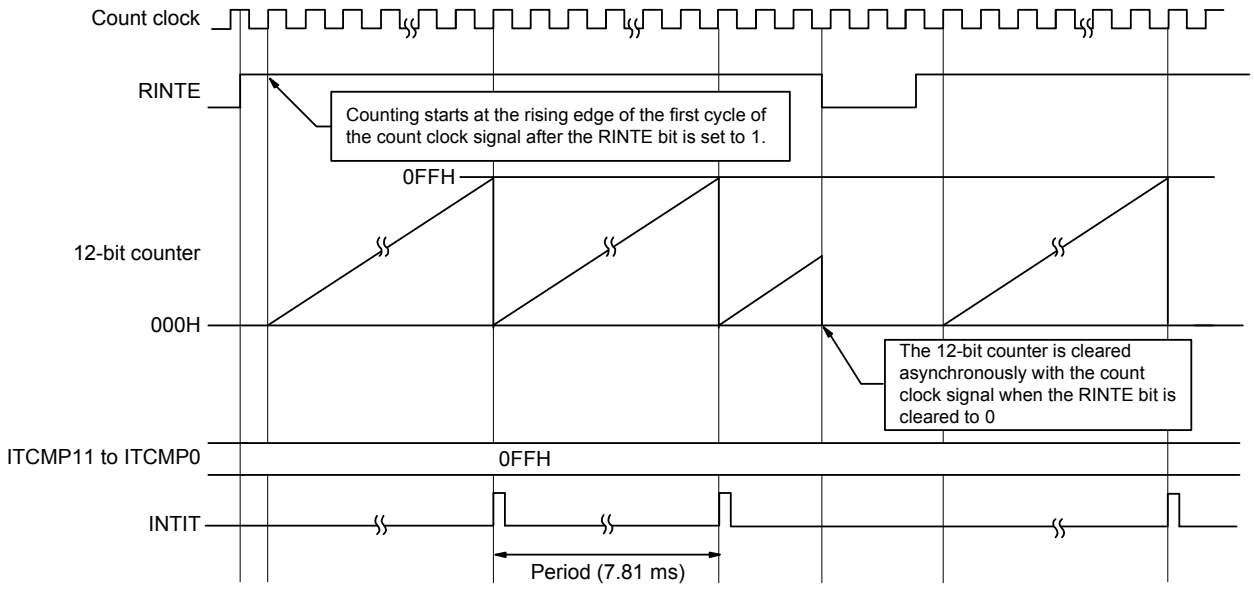
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate an 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

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 10 - 5 12-bit Interval Timer Operation Timing (ITCMP11 to ITCMP0 = 0FFH, count clock: f_{SUB} = 32.768 kHz)

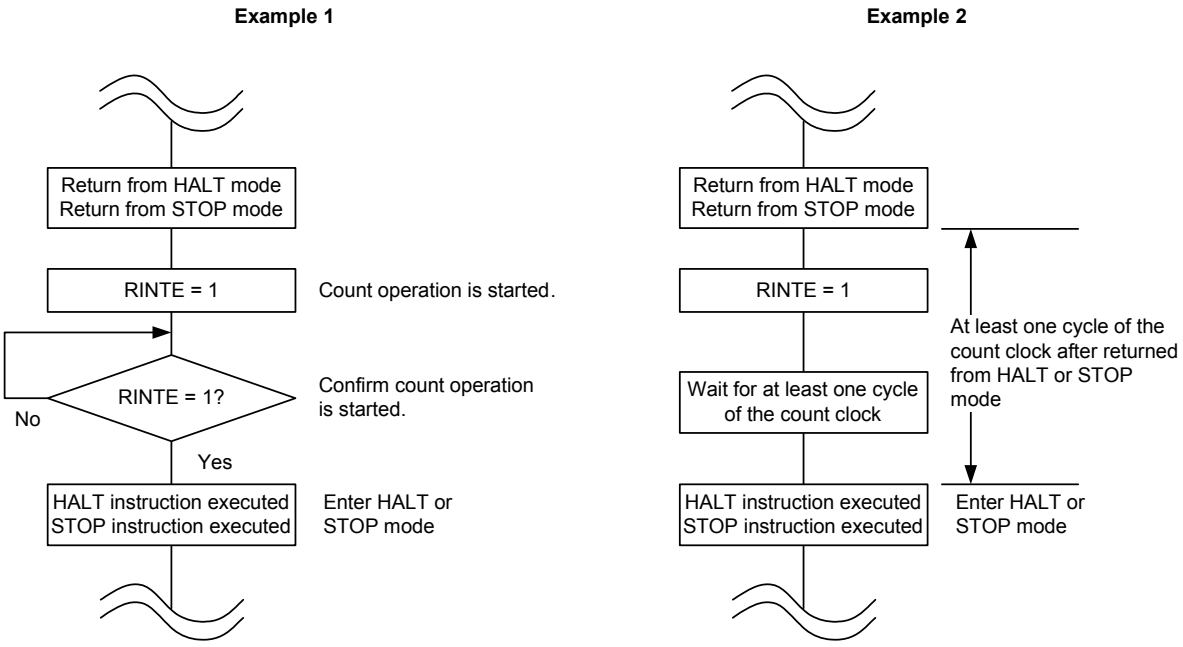


10.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

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 10 - 6**).
- 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 10 - 6**).

Figure 10 - 6 Procedure of entering to HALT or STOP mode after setting RINTE to 1



CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

11.1 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.

One pin can be used to output a clock or buzzer sound.

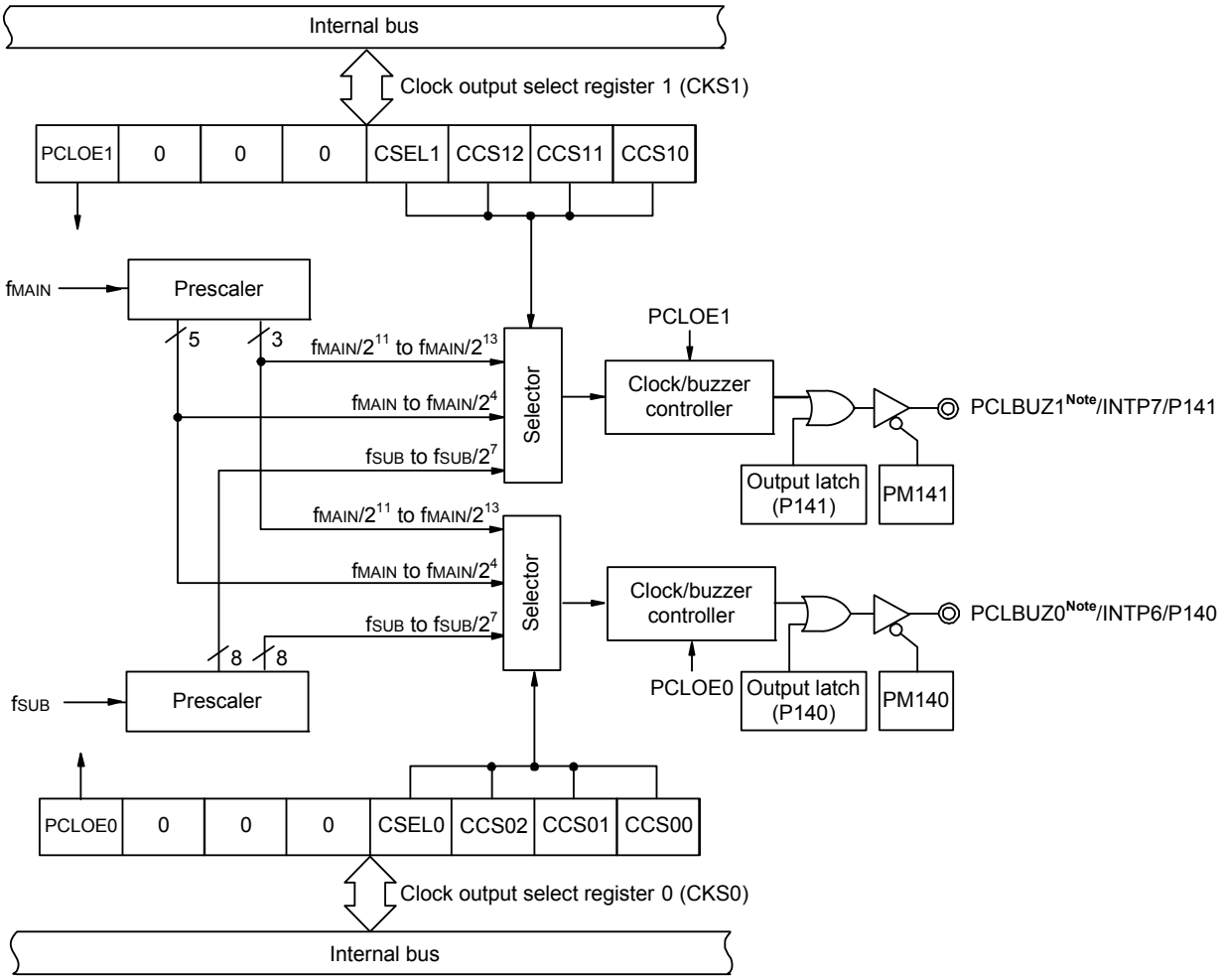
The PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).

Figure 11 - 1 shows the Block Diagram of Clock Output/Buzzer Output Controller.

Caution In HALT mode when RTCLPC in the subsystem clock supply mode control register (OSMC) = 1 and while the subsystem clock (f_{SUB}) is used for CPU operation, it is not possible to output the subsystem clock (f_{SUB}) from the PCLBUZn pin.

Remark n = 0, 1

Figure 11 - 1 Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to 31.4 **AC Characteristics**.

11.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 11 - 1 Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Port mode register 4 (PM14) Port register 14 (P14)

11.3 Registers Controlling Clock Output/Buzzer Output Controller

11.3.1 Clock output select registers n (CKSn)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZn), and set the output clock.

Select the clock to be output from the PCLBUZn 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.

<R>

Figure 11 - 2 Format of Clock output select registers n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn pin output enable/disable specification
0	Output disable (default)
1	Output enable

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn pin output clock selection				
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 32 MHz
0	0	0	0	f_{MAIN}	5 MHz**Note**	Setting prohibited	Setting prohibited	Setting prohibited
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz**Note**	Setting prohibited	Setting prohibited
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHzNote	8 MHzNote
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	15.63 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	0	0	0	f_{SUB}	32.768 kHz			
1	0	0	1	$f_{SUB}/2$	16.384 kHz			
1	0	1	0	$f_{SUB}/2^2$	8.192 kHz			
1	0	1	1	$f_{SUB}/2^3$	4.096 kHz			
1	1	0	0	$f_{SUB}/2^4$	2.048 kHz			
1	1	0	1	$f_{SUB}/2^5$	1.024 kHz			
1	1	1	0	$f_{SUB}/2^6$	512 Hz			
1	1	1	1	$f_{SUB}/2^7$	256 Hz			

Note Use the output clock within a range of 8 MHz. See **31.4 AC Characteristics** for details.

Caution 1. Change the output clock after disabling clock output (PCLOEn = 0).

Caution 2. To shift to STOP mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output while RTCLPC in the subsystem clock supply mode control register (OSMC) = 0 in STOP mode.

Caution 3. In HALT mode when RTCLPC in the subsystem clock supply mode control register (OSMC) = 1 and while the subsystem clock (f_{SUB}) is used for CPU operation, it is not possible to output the subsystem clock (f_{SUB}) from the PCLBUZn pin.

Remark 1. n = 0, 1

Remark 2. f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency

11.3.2 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 **5.3.1 Port mode registers (PMxx)** and **5.3.2 Port registers (Pxx)**.

Specifically, using a port pin with a multiplexed clock or buzzer output function (e.g. P140/INTP6/PCLBUZ0, P141/INTP7/PCLBUZ1) for clock or buzzer output, requires setting the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example: When P140/INTP6/PCLBUZ0 is to be used for clock or buzzer output
Set the PM140 bit of port mode register 14 to 0.
Set the P140 bit of port register 14 to 0.

11.4 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).

The PCLBUZ1 pin outputs a clock/buzzer selected by the clock output select register 1 (CKS1).

11.4.1 Operation as output pin

The PCLBUZn 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.

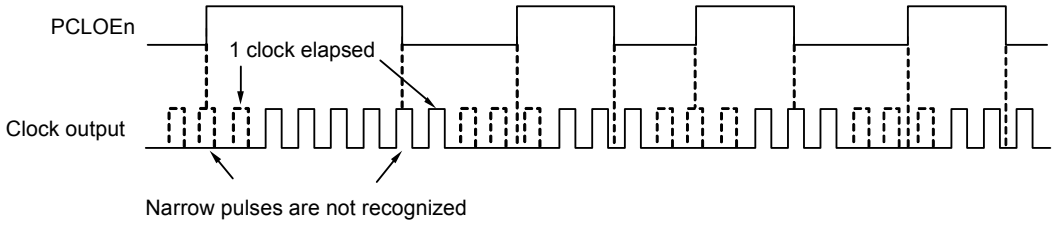
<2> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).

<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 11 - 3 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.

Remark 2. n = 0, 1

Figure 11 - 3 Timing of Outputting Clock from PCLBUZn Pin



11.5 Cautions of clock output/buzzer output controller

When the main system clock is selected for the PCLBUZn output (CSELn = 0), if STOP mode is entered within 1.5 clock cycles output from the PCLBUZn pin after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

CHAPTER 12 WATCHDOG TIMER

12.1 Functions of Watchdog Timer

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.
The watchdog timer operates on the low-speed on-chip oscillator clock (f_{IL}).
The counting operation of the watchdog timer is set by the option byte (000C0H).

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 21 RESET FUNCTION**.
When 75% + 1/2 f_{IL} of the overflow time is reached, an interval interrupt can be generated.

12.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 12 - 1 Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

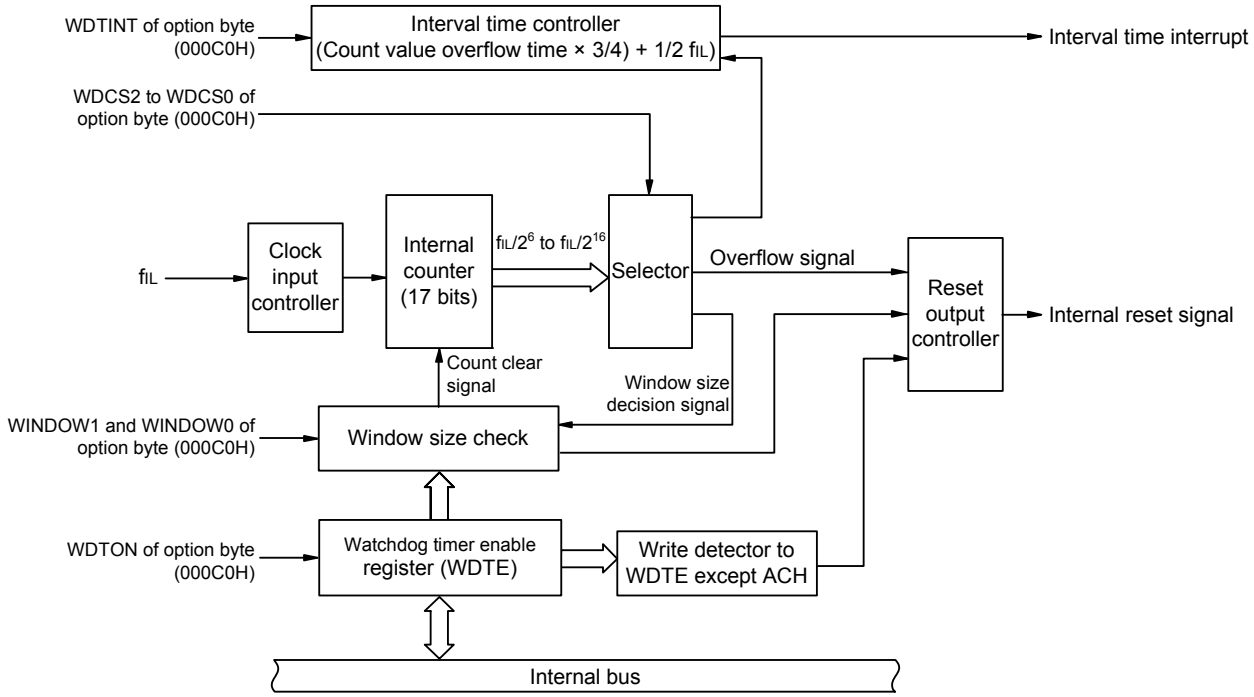
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 12 - 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 26 OPTION BYTE**.

Figure 12 - 1 Block Diagram of Watchdog Timer



Remark fIL: Low-speed on-chip oscillator clock

12.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

12.3.1 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 12 - 2 Format of Watchdog timer enable register (WDTE)

Address: FFFABH After reset: 9AH/1AH 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)).

12.4 Operation of Watchdog Timer

12.4.1 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 26**).

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 **12.4.2** and **CHAPTER 26**).
- Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **12.4.3** and **CHAPTER 26**).

(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).

Status	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.

12.4.2 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 12 - 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)

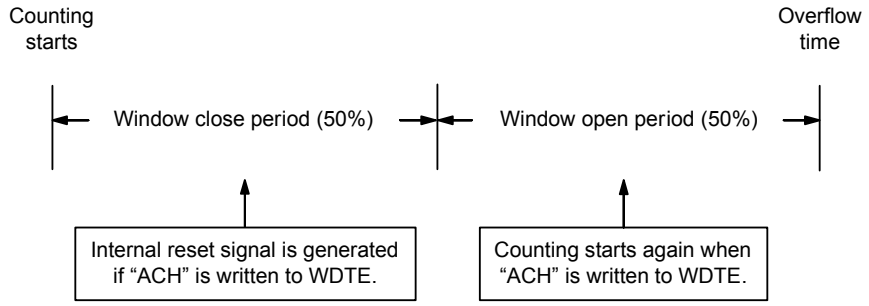
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

12.4.3 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 12 - 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%

<R> **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 to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)	237.44 to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)	474.89 to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)	1899.59 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.

Timing	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

12.4.4 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 12 - 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 13 A/D CONVERTER

13.1 Function of A/D Converter

The A/D converter is a converter that converts analog input signals into digital values, and 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.

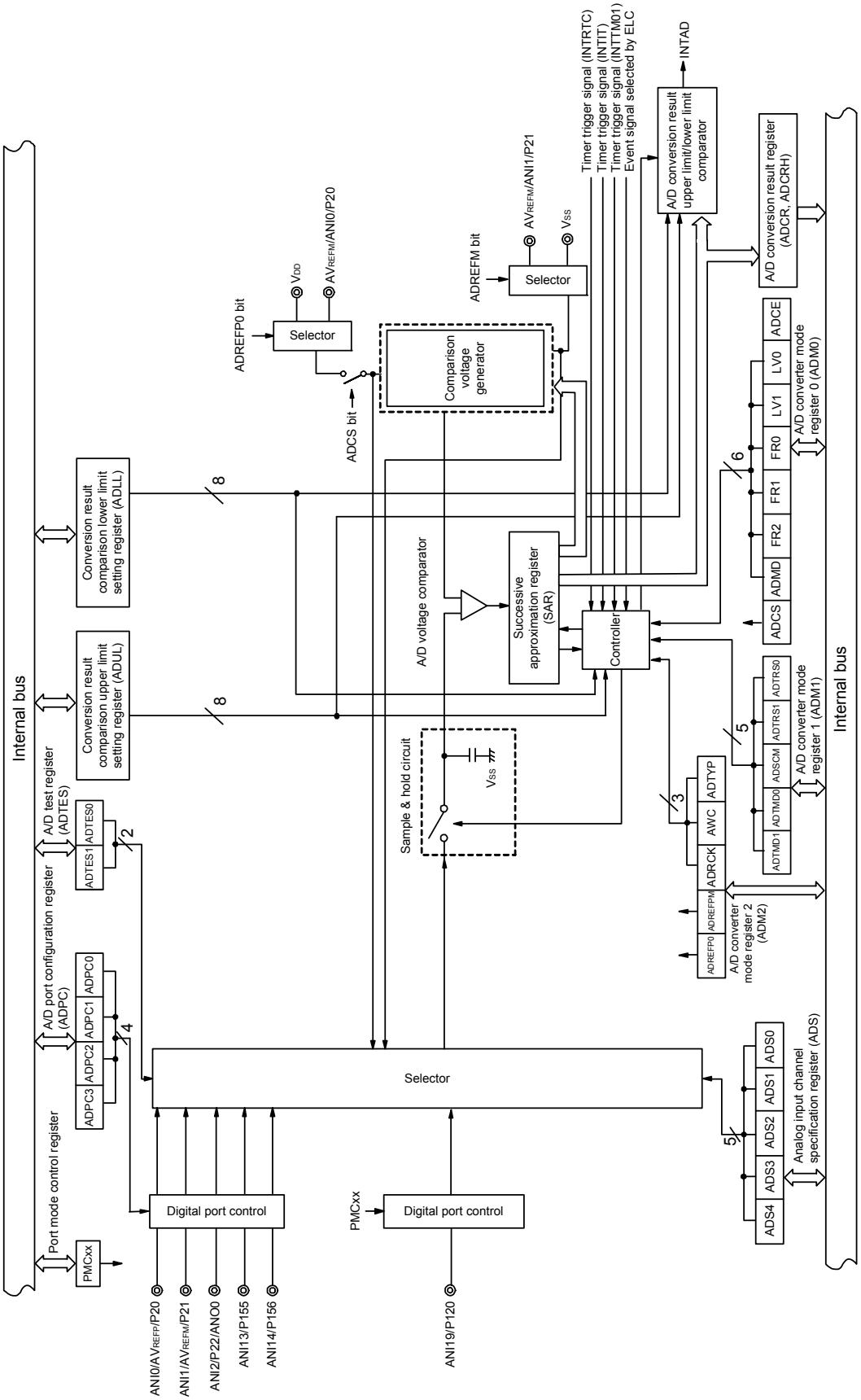
- 10-bit or 8-bit resolution A/D conversion

  10-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI2, ANI13, ANI14, and ANI19. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

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.
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} ≤ 3.6 V.
	Low voltage 1 or low voltage 2 mode	Conversion is done in the operation voltage range of 1.8 V ≤ V_{DD} ≤ 3.6 V. Select this mode for conversion at a low voltage. Because the operation voltage is low, it is internally boosted during conversion.
Sampling time selection	Sampling clock cycles: 7 f_{AD}	The sampling time in standard 1 or low voltage 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 or low voltage 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 13 - 1 Block Diagram of A/D Converter



13.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI2, ANI13, ANI14, and ANI19 pins

These are the analog input pins of the six 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 AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, 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 ADREFP0 bit of A/D converter mode register 2 (ADM2) to 1.

The analog signals input to ANI2, ANI13, ANI14, and ANI19 are converted to digital signals based on the voltage applied between AV_{REFP} and the – side reference voltage (AV_{REFM}/V_{SS}).

(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.

13.3 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)
- A/D port configuration register (ADPC)
- Port mode control register 12 (PMC12)
- Port mode registers 2, 12, and 15 (PM2, PM12, PM15)

13.3.1 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 13 - 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	RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	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.

Caution 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 2, 12, and 15 (PM2, PM12, PM15), port mode control register 12 (PMC12), and A/D port configuration register (ADPC)).

- **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)**

13.3.2 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 13 - 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 read] 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	Setting prohibited

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 **Tables 13 - 3** to **13 - 6 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 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 13.7 A/D Converter Setup Flowchart.

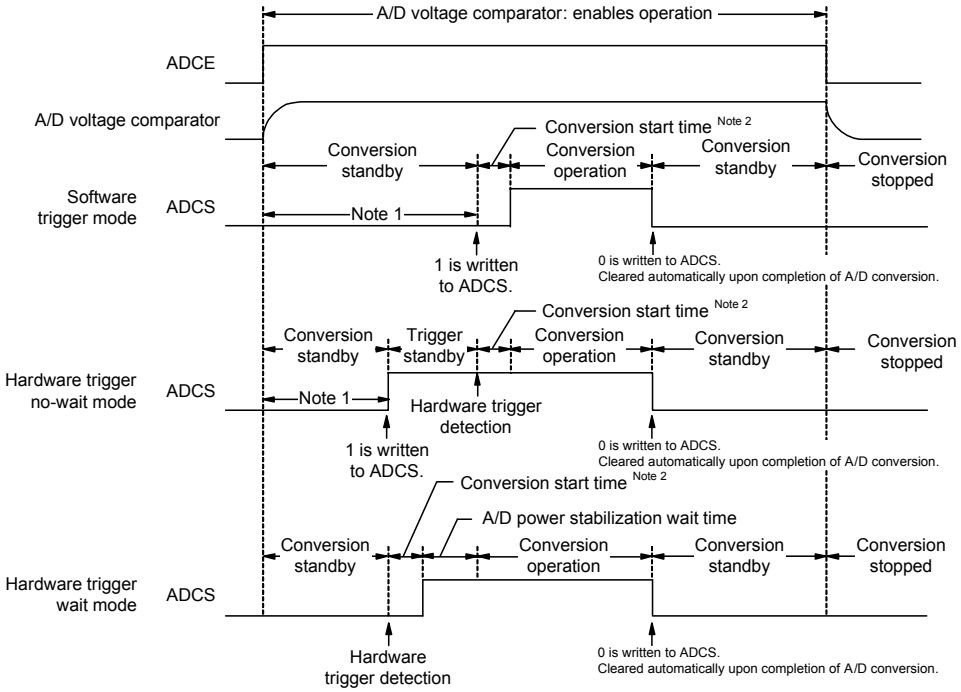
Table 13 - 1 Settings of ADCS and ADCE Bits

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 13 - 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.
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
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.

Figure 13 - 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 f_{CLK} 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	

For the second and subsequent conversion in sequential conversion mode, the stabilization wait time for A/D power supply do not occur in hardware trigger wait mode.

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 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

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

Table 13 - 3 A/D Conversion Time Selection (1/4)

(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 (f_{AD})	Number of Conversion Clock Note	Conversion Time	Conversion Time at 10-Bit Resolution				
									2.7 V ≤ V_{DD} ≤ 3.6 V				
FR2	FR1	FR0	LV1	LV0					f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	19 f_{AD} (number of sampling clock: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	76 µs	38 µs
0	0	1				f_{CLK}/32		608/f_{CLK}			76 µs	38 µs	19 µs
0	1	0				f_{CLK}/16		304/f_{CLK}		76 µs	38 µs	19 µs	9.5 µs
0	1	1				f_{CLK}/8		152/f_{CLK}		38 µs	19 µs	9.5 µs	4.75 µs
1	0	0				f_{CLK}/6		114/f_{CLK}		28.5 µs	14.25 µs	7.125 µs	3.5625 µs
1	0	1				f_{CLK}/5		95/f_{CLK}	95 µs	23.75 µs	11.875 µs	5.938 µs	2.9688 µs
1	1	0				f_{CLK}/4		76/f_{CLK}	76 µs	19 µs	9.5 µs	4.75 µs	2.375 µs
1	1	1				f_{CLK}/2		38/f_{CLK}	38 µs	9.5 µs	4.75 µs	2.375 µs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	17 f_{AD} (number of sampling clock: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	68 µs	34 µs
0	0	1				f_{CLK}/32		544/f_{CLK}			68 µs	34 µs	17 µs
0	1	0				f_{CLK}/16		272/f_{CLK}		68 µs	34 µs	17 µs	8.5 µs
0	1	1				f_{CLK}/8		136/f_{CLK}		34 µs	17 µs	8.5 µs	4.25 µs
1	0	0				f_{CLK}/6		102/f_{CLK}		25.5 µs	12.75 µs	6.375 µs	3.1875 µs
1	0	1				f_{CLK}/5		85/f_{CLK}	85 µs	21.25 µs	10.625 µs	5.3125 µs	2.6563 µs
1	1	0				f_{CLK}/4		68/f_{CLK}	68 µs	17 µs	8.5 µs	4.25 µs	2.125 µs
1	1	1				f_{CLK}/2		34/f_{CLK}	34 µs	8.5 µs	4.25 µs	2.125 µs	Setting prohibited

Note 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 (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 31.6.1 A/D converter characteristics

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 f_{CLK}: CPU/peripheral hardware clock frequency

Table 13 - 4 A/D Conversion Time Selection (2/4)

(2) When there is no A/D power supply stabilization wait time Low-voltage mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Note 4	Conversion Time	Conversion Time at 10-Bit Resolution				
									1.8 V ≤ V_{DD} ≤ 3.6 V		**Note 1**	**Note 2**	**Note 3**
FR2	FR1	FR0	LV1	LV0					f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	1	0	Low-voltage 1	f_{CLK}/64	19 f_{AD} (number of sampling clock: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	76 µs	38 µs
0	0	1				f_{CLK}/32		608/f_{CLK}			76 µs	38 µs	19 µs
0	1	0				f_{CLK}/16		304/f_{CLK}		76 µs	38 µs	19 µs	9.5 µs
0	1	1				f_{CLK}/8		152/f_{CLK}		38 µs	19 µs	9.5 µs	4.75 µs
1	0	0				f_{CLK}/6		114/f_{CLK}		28.5 µs	14.25 µs	7.125 µs	3.5625 µs
1	0	1				f_{CLK}/5		95/f_{CLK}	95 µs	23.75 µs	11.875 µs	5.938 µs	2.9688 µs
1	1	0				f_{CLK}/4		76/f_{CLK}	76 µs	19 µs	9.5 µs	4.75 µs	2.375 µs
1	1	1				f_{CLK}/2		38/f_{CLK}	38 µs	9.5 µs	4.75 µs	2.375 µs	Setting prohibited
0	0	0	1	1	Low-voltage 2	f_{CLK}/64	17 f_{AD} (number of sampling clock: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	68 µs	34 µs
0	0	1				f_{CLK}/32		544/f_{CLK}			68 µs	34 µs	17 µs
0	1	0				f_{CLK}/16		272/f_{CLK}		68 µs	34 µs	17 µs	8.5 µs
0	1	1				f_{CLK}/8		136/f_{CLK}		34 µs	17 µs	8.5 µs	4.25 µs
1	0	0				f_{CLK}/6		102/f_{CLK}		25.5 µs	12.75 µs	6.375 µs	3.1875 µs
1	0	1				f_{CLK}/5		85/f_{CLK}	85 µs	21.25 µs	10.625 µs	5.3125 µs	2.6563 µs
1	1	0				f_{CLK}/4		68/f_{CLK}	68 µs	17 µs	8.5 µs	4.25 µs	2.125 µs
1	1	1				f_{CLK}/2		34/f_{CLK}	34 µs	8.5 µs	4.25 µs	2.125 µs	Setting prohibited

Note 1. 1.8 V ≤ V_{DD} ≤ 3.6 V

Note 2. 2.4 V ≤ V_{DD} ≤ 3.6 V

Note 3. 2.7 V ≤ V_{DD} ≤ 3.6 V

Note 4. 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 (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 31.6.1 A/D converter characteristics.

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 f_{CLK}: CPU/peripheral hardware clock frequency

Table 13 - 5 A/D Conversion Time Selection (3/4)

(3) 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 (f_{AD})	Number of A/D Power Supply Stabilization Wait Clock	Number of Conversion Clock Note 2	A/D Power Supply Stabilization Wait Time + Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time at 10-Bit Resolution				
										2.7 V ≤ V_{DD} ≤ 3.6 V				
FR2	FR1	FR0	LV1	LV0						f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	8 f_{AD}	19 f_{AD} (number of sampling clock: 7 f_{AD})	1728/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	108 µs	54 µs
0	0	1				f_{CLK}/32			864/f_{CLK}			108 µs	54 µs	27 µs
0	1	0				f_{CLK}/16			432/f_{CLK}		108 µs	54 µs	27 µs	13.5 µs
0	1	1				f_{CLK}/8			216/f_{CLK}		54 µs	27 µs	13.5 µs	6.75 µs
1	0	0				f_{CLK}/6			162/f_{CLK}		40.5 µs	20.25 µs	10.125 µs	5.0625 µs
1	0	1				f_{CLK}/5			135/f_{CLK}	135 µs	33.75 µs	16.875 µs	8.4375 µs	4.21875 µs
1	1	0				f_{CLK}/4			108/f_{CLK}	108 µs	27 µs	13.5 µs	6.75 µs	3.375 µs
1	1	1				f_{CLK}/2			54/f_{CLK}	54 µs	13.5 µs	6.75 µs	3.375 µs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	8 f_{AD}	17 f_{AD} (number of sampling clock: 5 f_{AD})	1600/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	100 µs	50 µs
0	0	1				f_{CLK}/32			800/f_{CLK}			100 µs	50 µs	25 µs
0	1	0				f_{CLK}/16			400/f_{CLK}		100 µs	50 µs	25 µs	12.5 µs
0	1	1				f_{CLK}/8			200/f_{CLK}		50 µs	25 µs	12.5 µs	6.25 µs
1	0	0				f_{CLK}/6			150/f_{CLK}		37.5 µs	18.75 µs	9.375 µs	4.6875 µs
1	0	1				f_{CLK}/5			125/f_{CLK}	125 µs	31.25 µs	15.625 µs	7.8125 µs	3.90625 µs
1	1	0				f_{CLK}/4			100/f_{CLK}	100 µs	25 µs	12.5 µs	6.25 µs	3.125 µs
1	1	1				f_{CLK}/2			50/f_{CLK}	50 µs	12.5 µs	6.25 µs	3.125 µs	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion 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 13 - 3**).

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 (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 31.6.1 A/D converter characteristics.
Note that the conversion time (t_{CONV}) 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 f_{CLK}: CPU/peripheral hardware clock frequency

Table 13 - 6 A/D Conversion Time Selection (4/4)

(4) When there is A/D power supply stabilization wait time Low-voltage mode 1, 2

(hardware trigger wait mode Note 1)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of A/D Power Supply Stabilization Wait Clock	Number of Conversion Clock Note 5	A/D Power Supply Stabilization Wait Time + Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time at 10-Bit Resolution				
										1.8 V ≤ V_{DD} ≤ 3.6 V		**Note 2**	**Note 3**	**Note 4**
FR2	FR1	FR0	LV1	LV0						f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	1	0	Low-voltage1	f_{CLK}/64	2 f_{AD}	19 f_{AD} (number of sampling clock: 7 f_{AD})	1344/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	84 µs	42 µs
0	0	1				f_{CLK}/32			672/f_{CLK}			84 µs	42 µs	21 µs
0	1	0				f_{CLK}/16			336/f_{CLK}		84 µs	42 µs	21 µs	10.5 µs
0	1	1				f_{CLK}/8			168/f_{CLK}		42 µs	21 µs	10.5 µs	5.25 µs
1	0	0				f_{CLK}/6			126/f_{CLK}		31.25 µs	15.75 µs	7.875 µs	3.9375 µs
1	0	1				f_{CLK}/5			105/f_{CLK}	105 µs	26.25 µs	13.125 µs	6.5625 µs	3.238125 µs
1	1	0				f_{CLK}/4			84/f_{CLK}	84 µs	21 µs	10.5 µs	5.25 µs	2.625 µs
1	1	1				f_{CLK}/2			42/f_{CLK}	42 µs	10.5 µs	5.25 µs	2.625 µs	Setting prohibited
0	0	0	1	1	Low-voltage2	f_{CLK}/64	2 f_{AD}	17 f_{AD} (number of sampling clock: 5 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	76 µs	38 µs
0	0	1				f_{CLK}/32			608/f_{CLK}			76 µs	38 µs	19 µs
0	1	0				f_{CLK}/16			304/f_{CLK}		76 µs	38 µs	19 µs	9.5 µs
0	1	1				f_{CLK}/8			152/f_{CLK}		38 µs	19 µs	9.5 µs	4.75 µs
1	0	0				f_{CLK}/6			114/f_{CLK}		28.5 µs	14.25 µs	7.125 µs	3.5625 µs
1	0	1				f_{CLK}/5			95/f_{CLK}	96 µs	23.75 µs	11.875 µs	5.938 µs	2.9688 µs
1	1	0				f_{CLK}/4			76/f_{CLK}	76 µs	19 µs	9.5 µs	4.75 µs	2.375 µs
1	1	1				f_{CLK}/2			38/f_{CLK}	38 µs	9.5 µs	4.75 µs	2.375 µs	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion 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 13 - 4**).

Note 2. 1.8 V ≤ V_{DD} ≤ 3.6 V

Note 3. 2.4 V ≤ V_{DD} ≤ 3.6 V

Note 4. 2.7 V ≤ V_{DD} ≤ 3.6 V

Note 5. 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 (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 31.6.1 A/D converter characteristics.
Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.

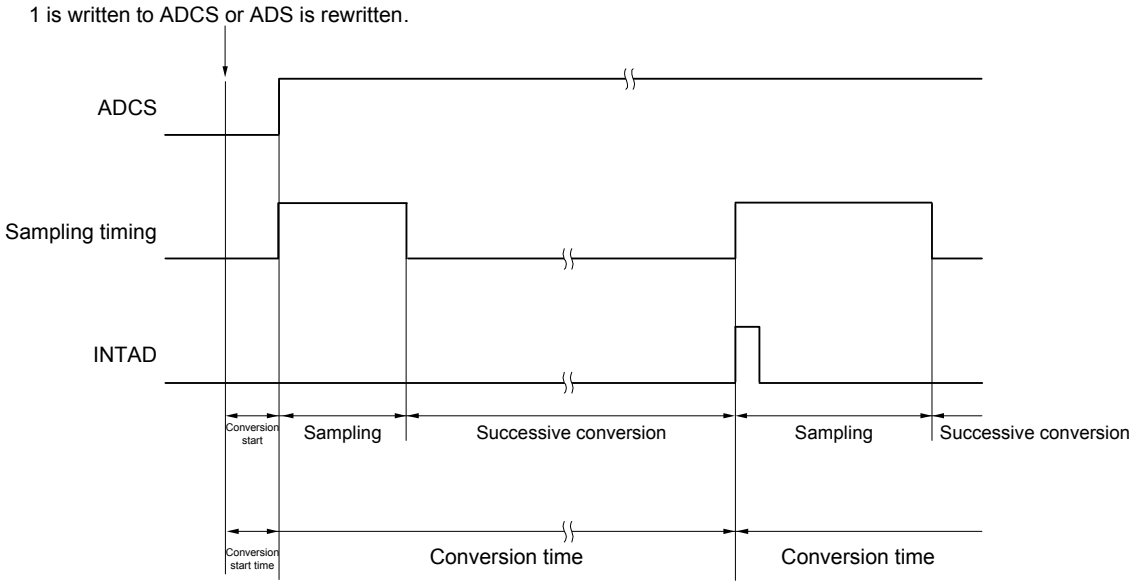
Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

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 f_{CLK}: CPU/peripheral hardware clock frequency

Figure 13 - 5 A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



13.3.3 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 13 - 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	0	Software trigger mode
0	1	
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	Event signal selected by ELC
1	0	Real-time clock interrupt signal (INTRTC)
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 INTRTC or INTIT will not be recognized as a valid hardware trigger for up to four f_{CLK} cycles after the first INTRTC or INTIT is input.

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

13.3.4 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 13 - 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	0	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	Supplied from V_{DD}
1	Supplied from P20/AV_{REFP}/ANI0
• When ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures. (1) Set ADCE = 0 (2) Change the values of ADREFP0 (3) Set ADCE = 1 (4) Reference voltage stabilization wait time (1 μs) Start A/D conversion after the time described in (4) has elapsed.	

ADREFM	Selection of the - side reference voltage of the A/D converter
0	Supplied from V_{SS}
1	Supplied from P21/AV_{REFM}/ANI1

Caution 1. Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. 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 13 - 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	0	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (AREA1).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (AREA2) or the ADUL register < the ADCR register (AREA3).
Figure 13 - 9 shows the generation range of the interrupt signal (INTAD) for AREA1 to AREA3.	

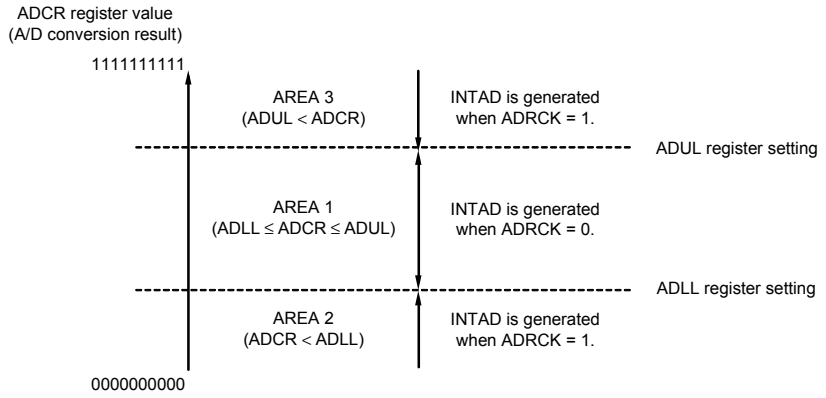
AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode). • The SNOOZE mode function can only 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 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. 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 “Transition time from STOP mode to SNOOZE mode” in **20.3.3 SNOOZE mode**.

Caution **Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).**

Figure 13 - 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.

13.3.5 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result in the select mode. 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 13 - 9**), the result is not stored.

Figure 13 - 10 Format of 10-bit A/D conversion result register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 (bits 7 and 6 of the ADCR register).

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.

13.3.6 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 13 - 9**), the result is not stored.

Figure 13 - 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 When writing to the A/D converter mode register 0 (ADM0), Analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, and ADPC registers. Using timing other than the above may cause an incorrect conversion result to be read.

13.3.7 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 13 - 12 Format of Analog input channel specification register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

• Select mode (ADMD = 0)

ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	ANI0	P20/ANI0/AVREFP pin
0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	1	0	ANI2	P22/ANI2 pin
0	1	1	0	1	ANI13	P155/ANI13 pin
0	1	1	1	0	ANI14	P156/ANI14 pin
1	0	0	1	1	ANI19	P120/ANI19 pin
Other than the above					Setting prohibited	

Caution 1. Be sure to clear bits 5, 6, and 7 to 0.

Caution 2. Set a channel to be set the analog input by ADPC and PMCx registers in the input mode by using port mode registers 2, 12, and 15 (PM2, PM12, PM15).

Caution 3. Do not set the pin that is set by the A/D port configuration register (ADPC) as digital I/O by the ADS register.

Caution 4. Do not set the pin that is set by Port mode control register 12 (PMC12) as digital I/O by the ADS register.

Caution 5. If using AVREFP as the + side reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 6. If using AVREFM as the – side reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.

13.3.8 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 13 - 9**).

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

Reset signal generation sets this register to FFH.

Caution 1.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 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 and ADLL registers must be greater than that of the ADLL register.

Figure 13 - 13 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

13.3.9 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 13 - 9**).

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

Reset signal generation clears this register to 00H.

Figure 13 - 14 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 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 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 and ADLL registers must be greater than that of the ADLL register.

13.3.10 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 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 13 - 15 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 (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 ADREFP0 bit of the ADM2 register)
Other than the above		Setting prohibited

13.3.11 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), and A/D port configuration register (ADPC)). For details, see **5.3.1 Port mode registers (PMxx)**, **5.3.6 Port mode control registers (PMCxx)**, and **5.3.7 A/D port configuration register (ADPC)**.

When using the ANI0 to ANI2, ANI13, and ANI14 pins for analog input of the A/D converter, set the port mode register (PMxx) bit corresponding to each port to 1 and select analog input through the A/D port configuration register (ADPC).
When using the ANI19 pin 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.

13.4 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 **Note 1**.

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 2**.

To stop the A/D converter, clear the ADCS bit to 0.

Note 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see **Figure 13 - 9**), the A/D conversion result interrupt request signal is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.

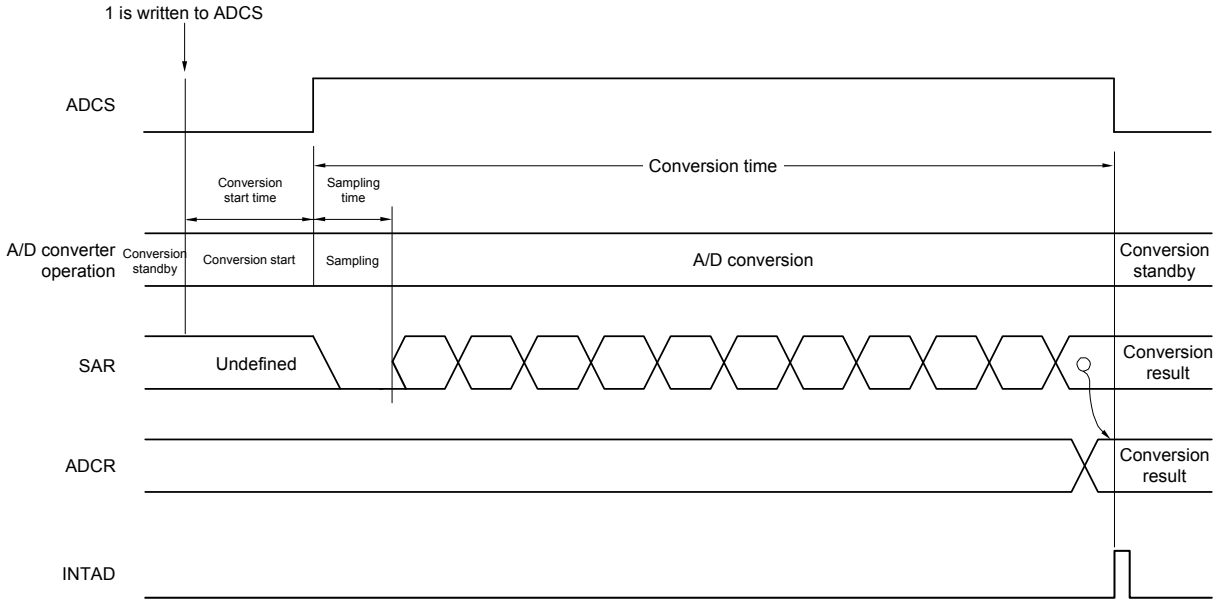
Note 2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

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} and V_{DD}.

Figure 13 - 16 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.

13.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

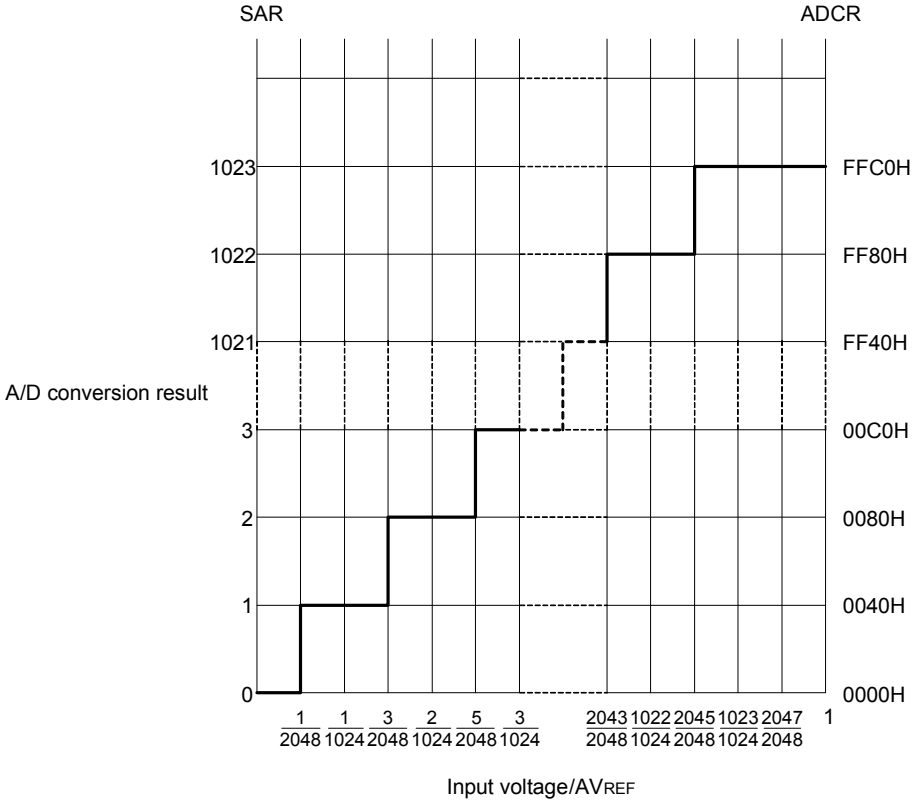
or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

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 13 - 17 shows the Relationship Between Analog Input Voltage and A/D Conversion Result.

Figure 13 - 17 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}, and V_{DD}.

13.6 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 **13.7 A/D Converter Setup Flowchart**.

13.6.1 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.

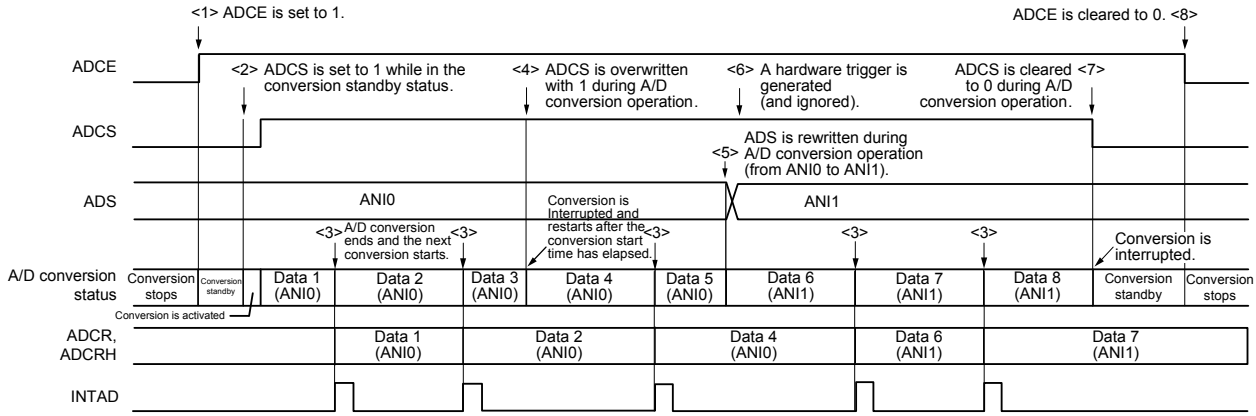
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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 13 - 18 Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



13.6.2 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.

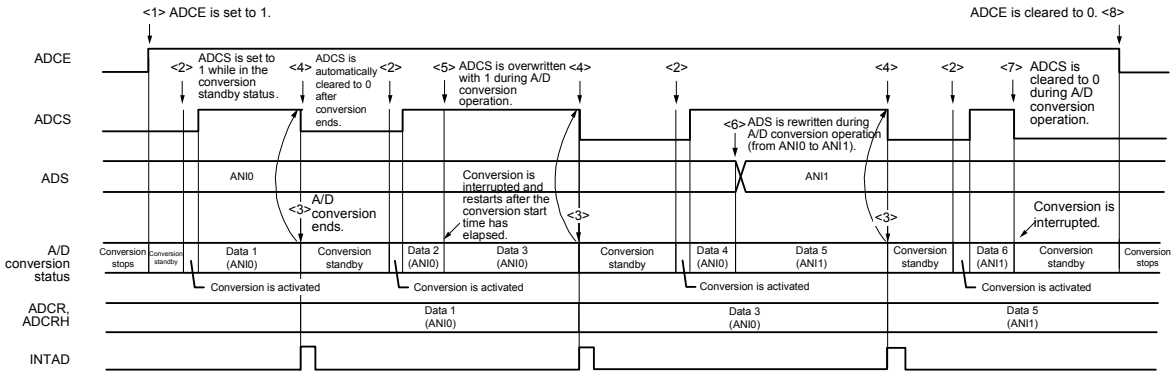
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 13 - 19 Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



13.6.3 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.

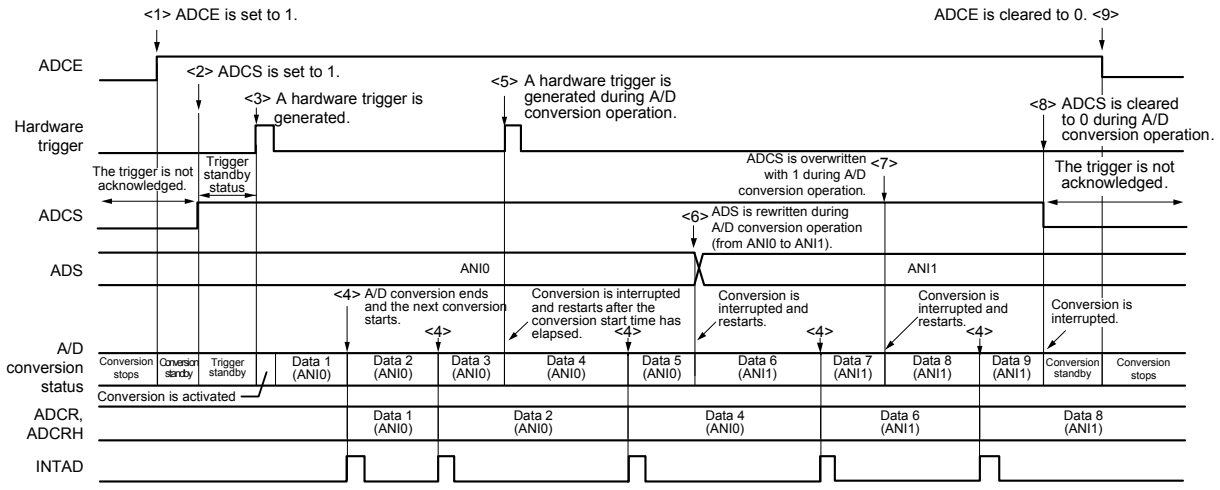
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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 13 - 20 Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



<R> 13.6.4 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 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.

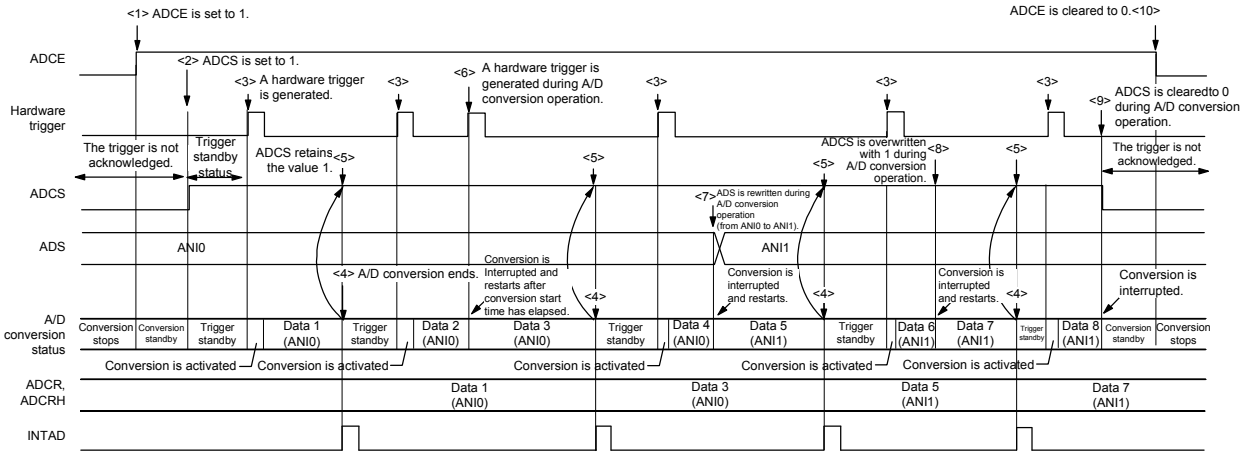
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<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.

<R> Figure 13 - 21 Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



13.6.5 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.)

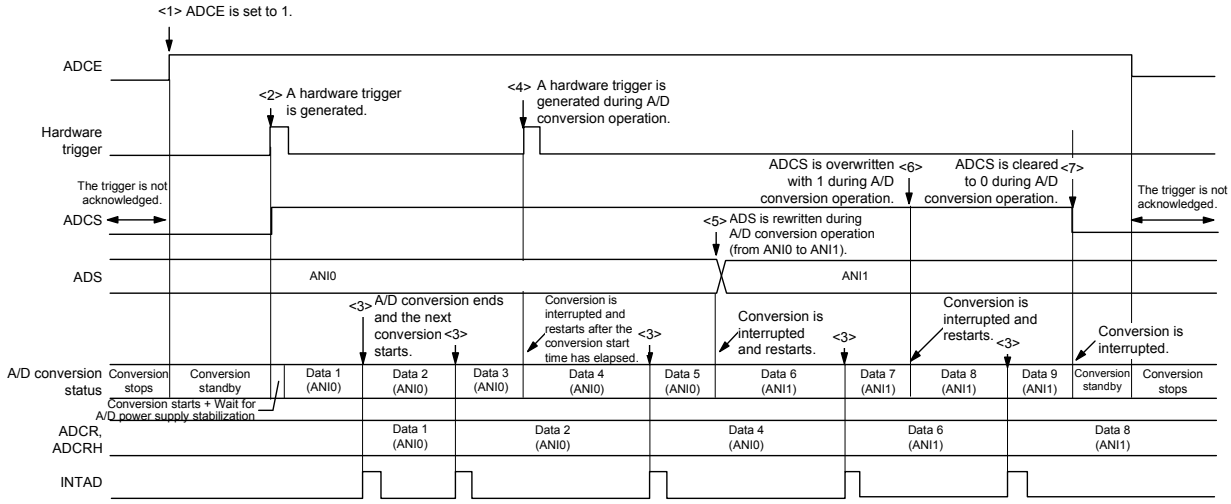
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 13 - 22 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



13.6.6 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.

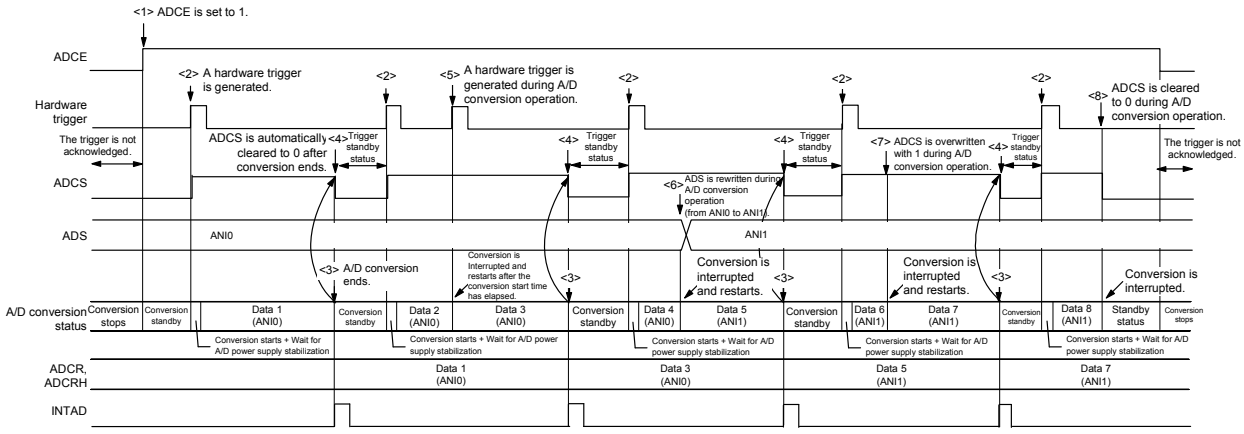
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 13 - 23 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing

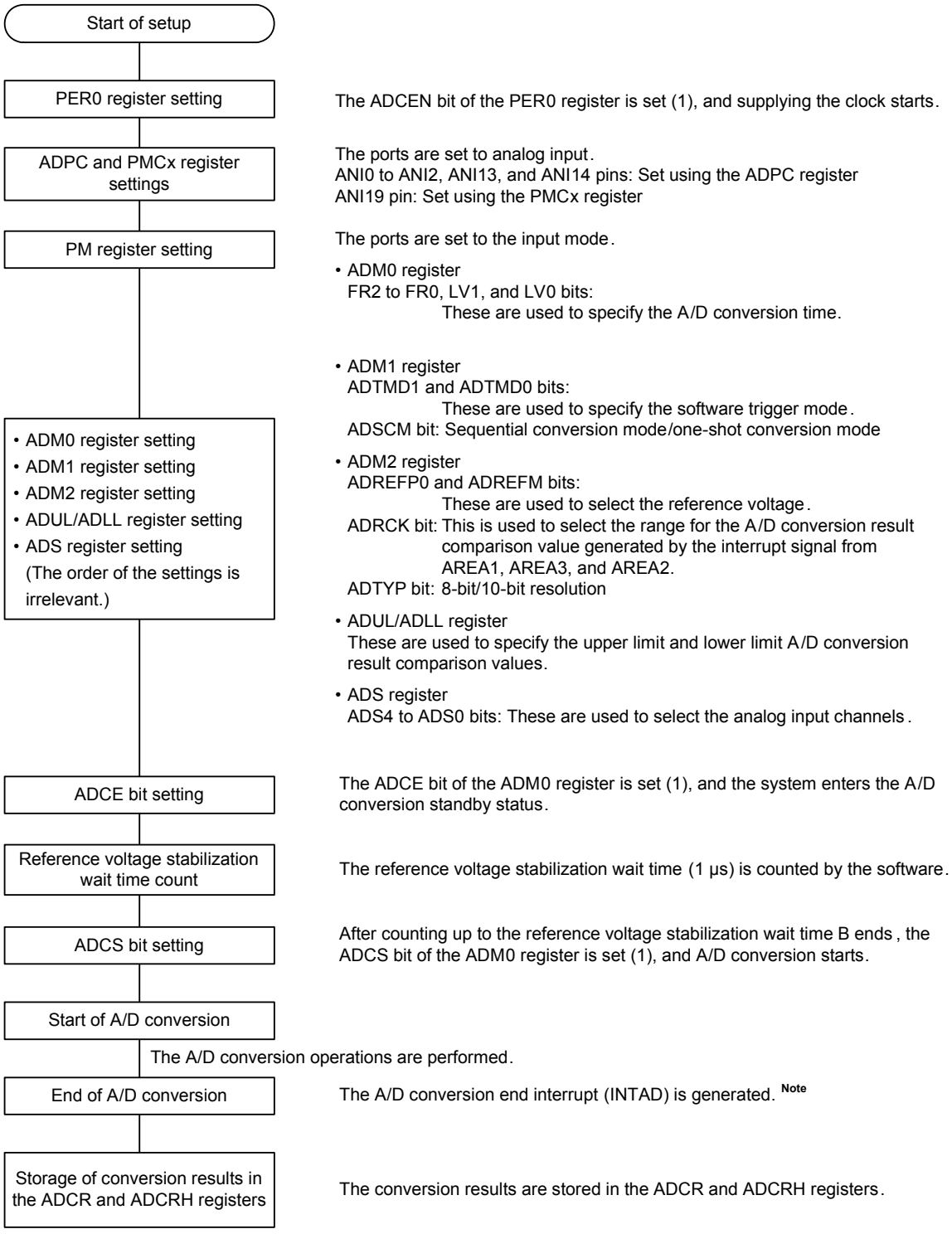


13.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

13.7.1 Setting up software trigger mode

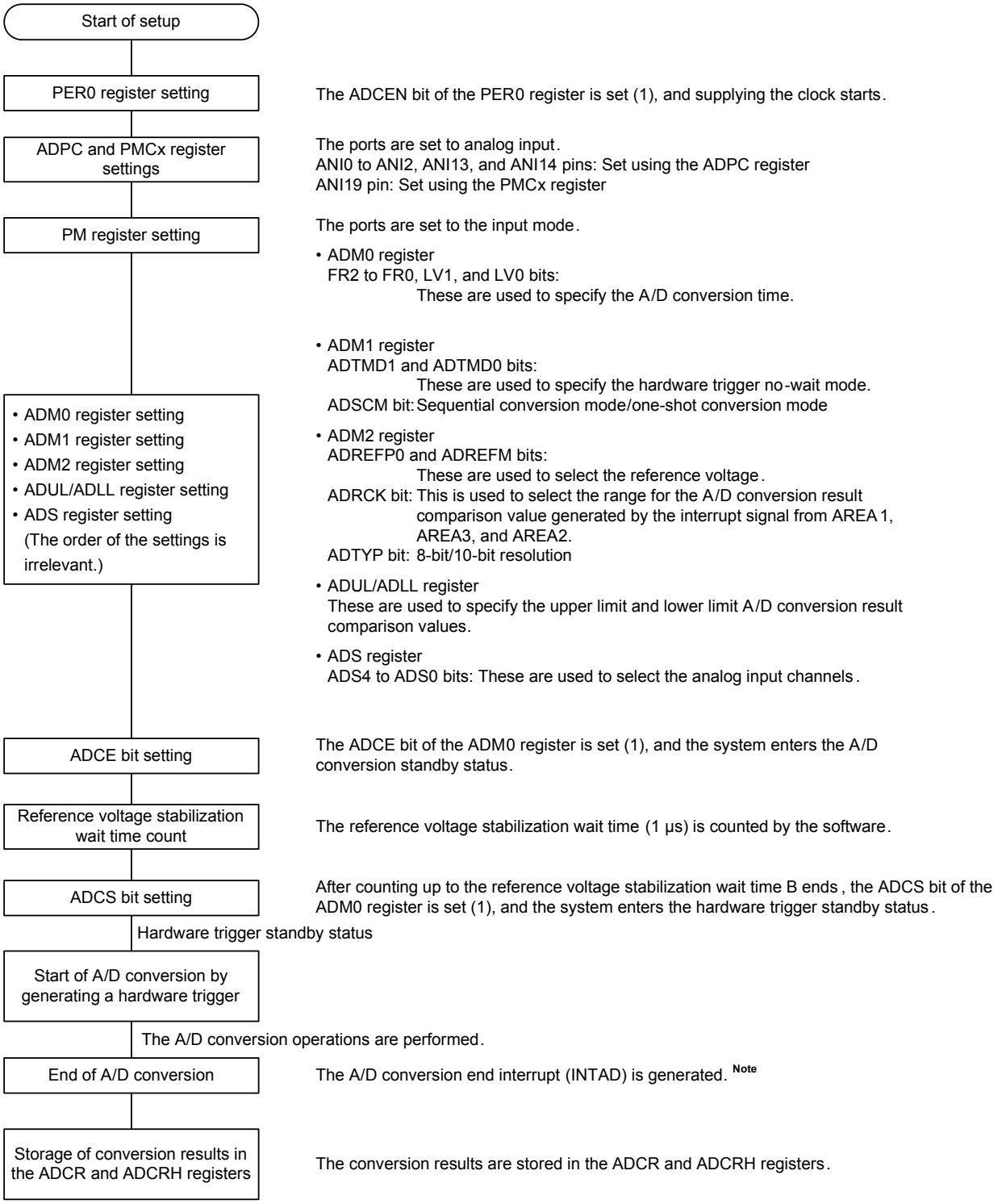
Figure 13 - 24 Setting up Software Trigger Mode



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 register.

13.7.2 Setting up hardware trigger no-wait mode

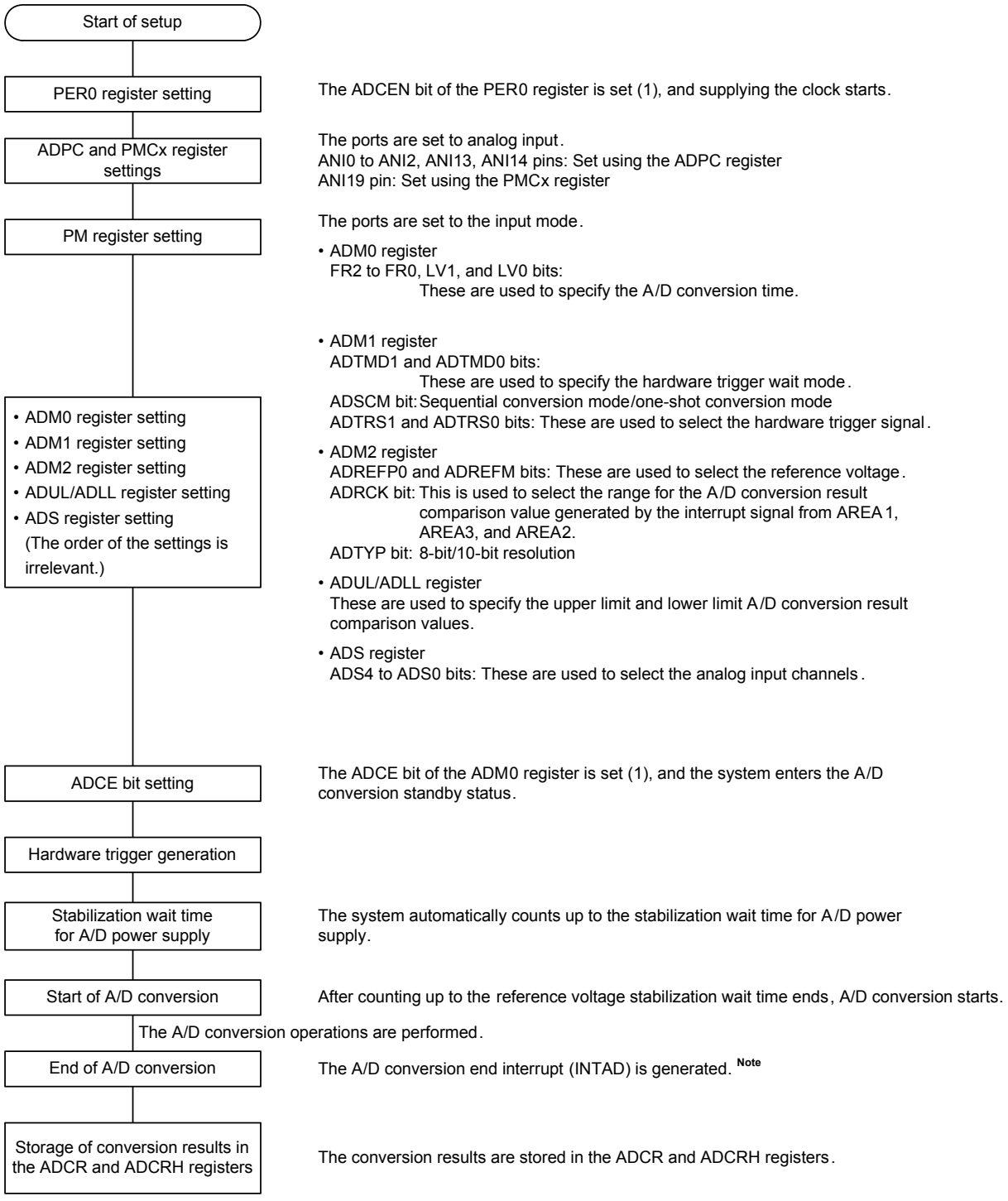
Figure 13 - 25 Setting up Hardware Trigger No-Wait Mode



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 register.

13.7.3 Setting up hardware trigger wait mode

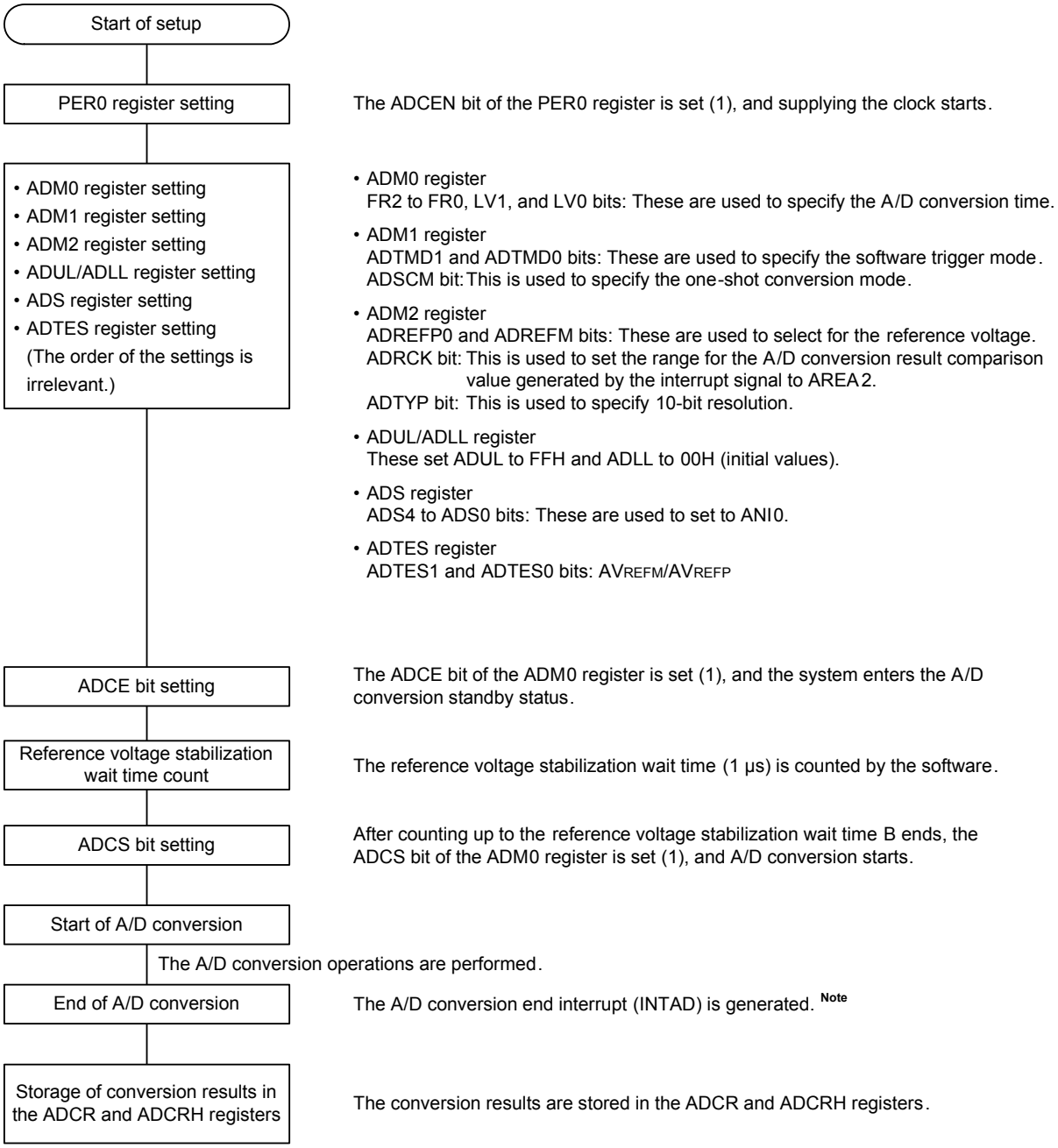
Figure 13 - 26 Setting up Hardware Trigger Wait Mode



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 register.

13.7.4 Setting up test mode

Figure 13 - 27 Setting up Test 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 register.

Caution For the procedure for testing the A/D converter, see 24.3.8 A/D test function.

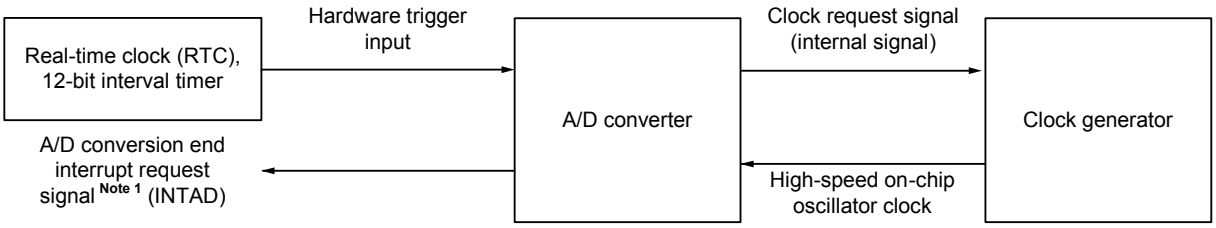
13.8 SNOOZE Mode Function

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 conversion mode can be used:
• Hardware trigger wait mode (select 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 13 - 28 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 **13.7.3 Setting up hardware trigger wait mode** Note 2.) At this time, 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 E1H, E2H or E3H.

Remark The hardware trigger is event selected by ELC, INTRTC or 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.

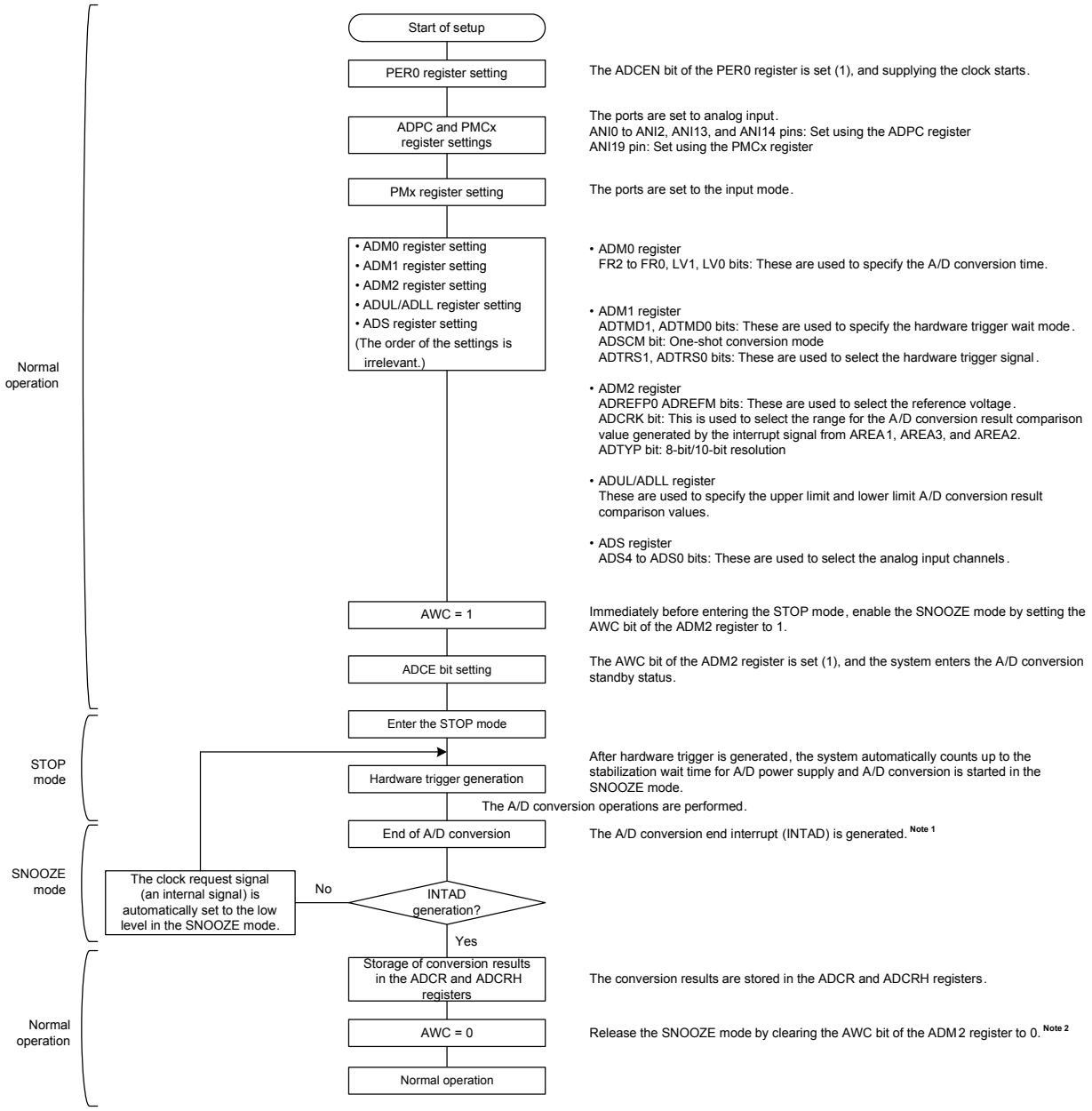
(2) If no interrupt is generated after A/D conversion ends

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.

Figure 13 - 29 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.

13.9 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 ANIn pins

Observe the rated range of the ANIn pins input voltage. If a voltage exceeding V_{DD} and AV_{REFP} or below V_{SS} and AV_{REFM} (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.

(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), or A/D port configuration register (ADPC) write upon the end of conversion

The ADM0, ADS, or ADPC 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 ANI2, ANI13, ANI14, and ANI19 pins.

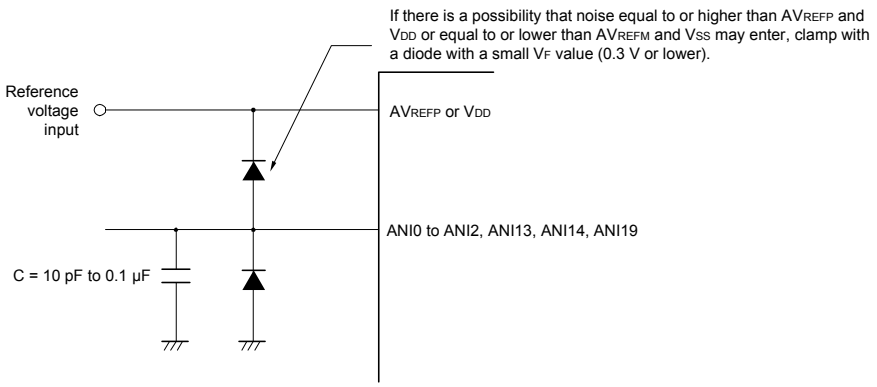
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<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 13 - 30 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 13 - 30 Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> The analog input pins are also used as input port pins.
When A/D conversion is performed with any of the ANI0 to ANI2, ANI13, ANI14, and ANI19 pins selected, do not change to output value P20 to P22, P120, P155, and P156 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to prevent such a pulse from being input or output.

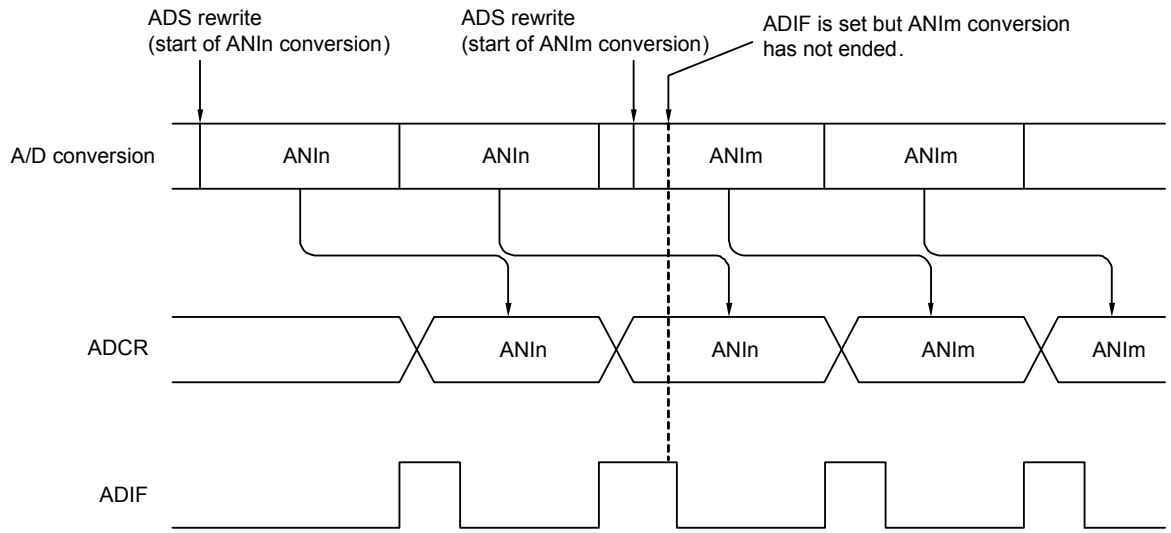
(6) Input impedance of analog input (ANIn) pins

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, we recommend using the converter with analog input sources that have output impedances no greater than 1 kΩ. If a source has a higher output impedance, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1 μF) to the pin from among ANI0 to ANI2, ANI13, ANI14, and ANI19 to which the source is connected (see **Figure 13 - 30**). 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 sampling 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 13 - 31 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), A/D port configuration register (ADPC), and port mode control register (PMCxx), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, ADPC, 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 13 - 32 Internal Equivalent Circuit of ANIn Pin

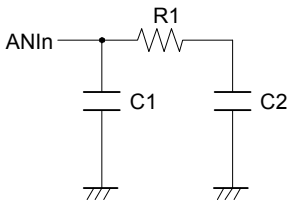


Table 13 - 7 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REFP}, V_{DD}	ANIn Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
2.7 V ≤ V_{DD} < 3.6 V	ANI0 to ANI2, ANI13, ANI14	39	8	2.5
	ANI19	53	8	7.0
1.8 V ≤ V_{DD} < 2.7 V	ANI0 to ANI2, ANI13, ANI14	231	8	2.5
	ANI19	321	8	7.0

Remark The resistance and capacitance values shown in Table 13 - 7 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 14 SERIAL ARRAY UNIT

Serial array unit can achieve communication functions of 3-wire serial (CSI) and UART. Function assignment of each channel is as shown below.

Unit	Channel	Used as CSI	Used as UART
0	0	Cannot be used	Cannot be used
	1		
	2	CSI10	UART1
	3	Cannot be used	
1	0	CSI20 Note	Cannot be used
	1	CSI21	
	2	CSI30	UART3
	3	Cannot be used	

Note CSI20 is used for internal communication between MCU and RF transceiver. For this channel, only the master can be selected.

Remark In this chapter, indexes "m", "n", "p", and "q" are used for the unit number, channel number, CSI number, and UART number respectively.

Each index may be replaced with the following values.

m: Unit number (m = 0, 1)

n: Channel number (n = 0 to 3)

p: CSI number (p = 10, 20, 21, 30)

q: UART number (q = 1, 3)

14.1 Functions of Serial Array Unit

Each serial interface has the following features.

14.1.1 3-wire serial I/O (CSIp)

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 **14.5 Operation of 3-Wire Serial I/O (CSIp) 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}/4
  During slave communication: Max. f_{MCK}/6
[Interrupt function]
• Transfer end interrupt/buffer empty interrupt
[Error detection flag]
• Overrun error

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 31 ELECTRICAL SPECIFICATIONS**.

14.1.2 UART (UARTq)

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 **14.6 Operation of UART (UARTq) Communication**.

[Data transmission/reception]
• Data length of 7, or 8 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

14.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 14 - 1 Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits
Buffer register	Lower 8 bits of serial data register mn (SDRmn) Note 1
Serial clock I/O	SCK10, SCK20, SCK21, SCK30 pins (for 3-wire serial I/O)Note 2
Serial data input	SI10, SI20, SI21, SI30 pins (for 3-wire serial I/O), RxD1, RxD3 pins (for UART)
Serial data output	SO10, SO20, SO21, SO30 pins (for 3-wire serial I/O), TxD1, TxD3 pins (for UART)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • 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) • 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 registers 0, 14 (PIM0, PIM14) • Port output mode registers 0, 14 (POM0, POM14) • Port mode registers 0, 1, 7, 14 (PM0, PM1, PM7, PM14) • Port registers 0, 1, 7, 14 (P0, P1, P7, P14)

Note 1. The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, 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)

Note 2. Output only for SCK20.

Figure 14 - 1 shows the Block Diagram of Serial Array Unit 0.

Figure 14 - 1 Block Diagram of Serial Array Unit 0

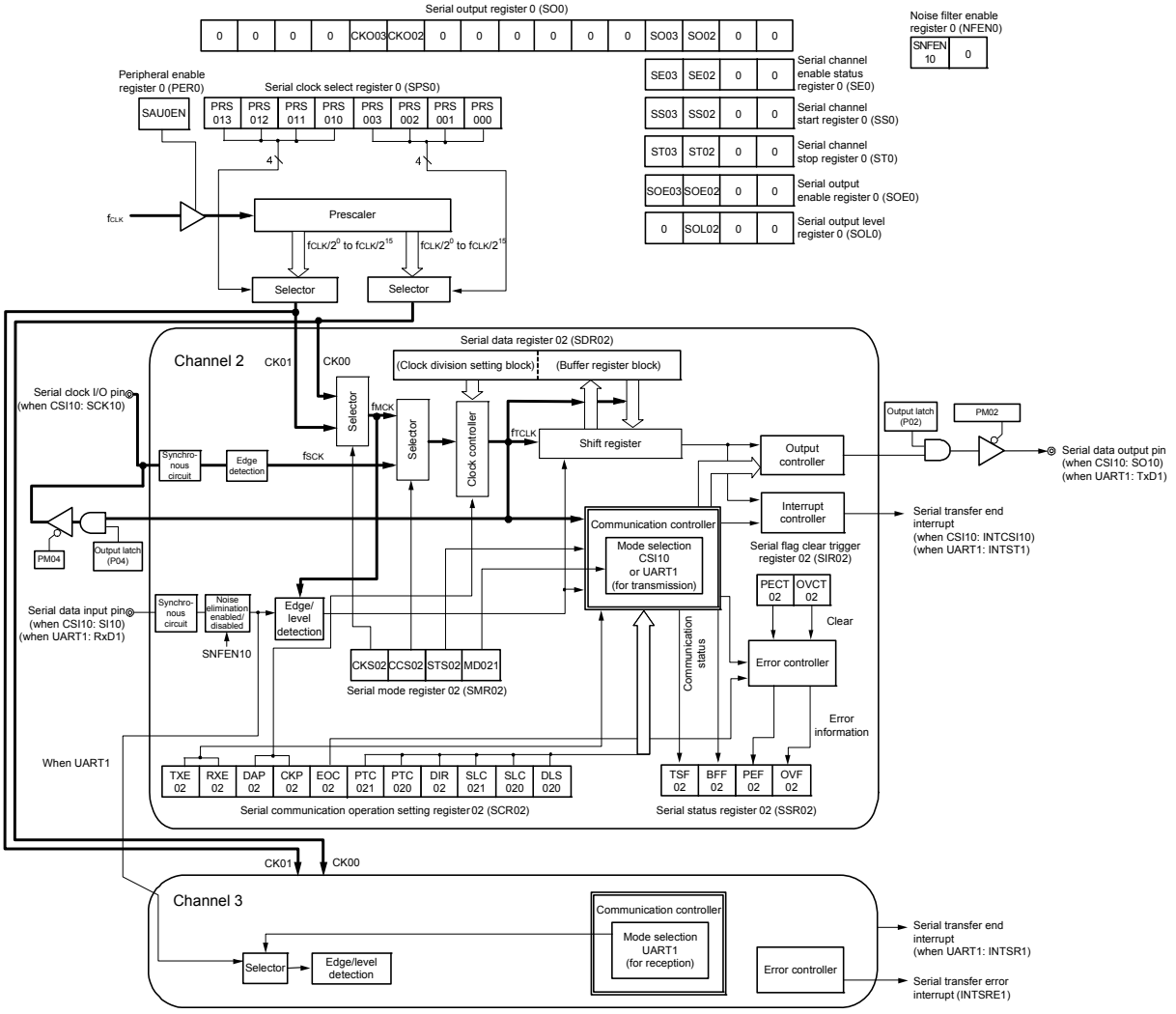
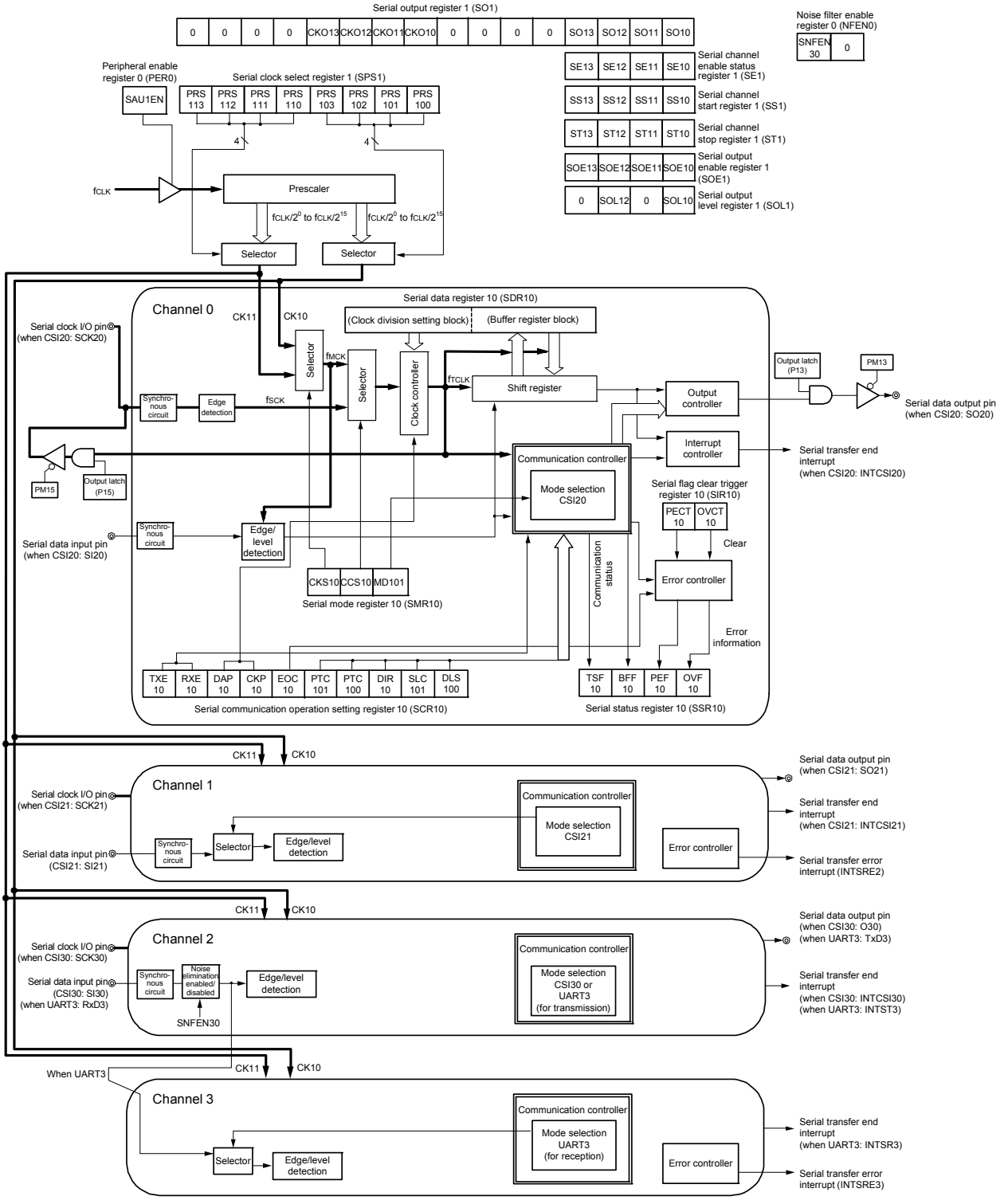


Figure 14 - 2 shows the Block Diagram of Serial Array Unit 1.

Figure 14 - 2 Block Diagram of Serial Array Unit 1



14.2.1 Shift register

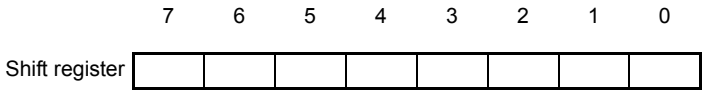
This is an 8-bit register that converts parallel data into serial data or vice versa.

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 bits of serial data register mn (SDRmn).



14.2.2 Lower 8 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n.

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 bits. When data is to be transmitted, set transmit data to be transferred to the shift register to the lower 8 bits.

The data stored in the lower 8 bits of this register is as follows, depending on the setting of bit 0 (DLSmn0) 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)

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

The lower 8 bits of the SDRmn register can be read or written **Note** as the following SFR, 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)

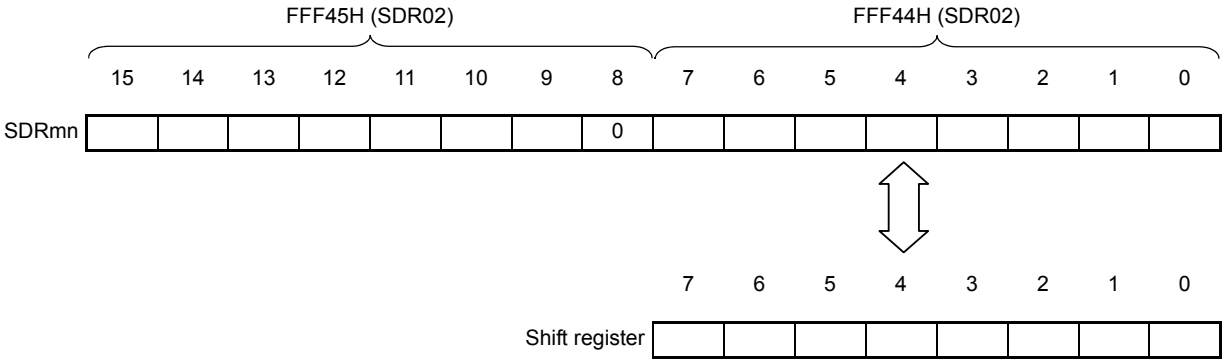
Reset signal generation clears the SDRmn register to 0000H.

Note When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remark After data is received, “0” is stored in bits 0 to 7 in bit portions that exceed the data length.

Figure 14 - 3 Format of Serial data register mn (SDRmn) (mn = 02, 03, 10, 11, 12, 13)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03), FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11) FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13) After reset: 0000H R/W



Caution **Be sure to clear bit 8 to "0".**

Remark For the function of the higher 7 bits of the SDRmn register, see **14.3 Registers Controlling Serial Array Unit**.

14.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

Caution In this chapter, read all the registers of the channels that are not available as the reserved registers.

- Peripheral enable register 0 (PER0)
- 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)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0, 14 (PIM0, PIM14)
- Port output mode registers 0, 14 (POM0, POM14)
- Port mode registers 0, 1, 7, 14 (PM0, PM1, PM7, PM14)
- Port registers 0, 1, 7, 14 (P0, P1, P7, P14)

14.3.1 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.

When serial array unit 1 is used, be sure to set bit 3 (SAU1EN) 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 14 - 4 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	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. • Serial array unit m is in the reset status.
1	Enables input clock supply. • SFR used by serial array unit m can be read/written.

Caution 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, and, even if the register is read, only the default value is read (except for noise filter enable register 0 (NFEN0), port input mode registers, port output mode registers, port mode registers, port mode control registers, and port registers).

- **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)**

14.3.2 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 14 - 5 Format of Serial clock select register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) 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} = 32 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.8 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.9 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	977 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. k = 0, 1

14.3.3 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 or UART), 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 14 - 6 Format of Serial mode register mn (SMRmn) (1/2)

Address: F0114H, F0115H (SMR02), F0116H, F0117H (SMR03), F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13) — 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	0	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 and UART transmission).
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 SMR03 and SMR13 registers only.

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR02, SMR10 to SMR12 register) to "0". Be sure to set bit 5 to "1".

Figure 14 - 7 Format of Serial mode register mn (SMRmn) (2/2)

Address: F0114H, F0115H (SMR02), F0116H, F0117H (SMR03), F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13) 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	0	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 mn1	Setting of operation mode of channel n
0	CSI mode
1	UART mode

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 SMR03 and SMR13 registers only.

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR02, SMR10 to SMR12 register) to "0". Be sure to set bit 5 to "1".

14.3.4 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 14 - 8 Format of Serial communication operation setting register mn (SCRmn) (1/2)

Address: F011CH, F011DH (SCR02), F011EH, F011FH (SCR03), F0158H, F0159H (SCR10)to F015EH, F015FH (SCR13) 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	SLC mn1 Note 1	SLC mn0	0	1	1	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.			

EOC mn	Mask control of error interrupt signal (INTSREx (x = 1, 3))
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, and during UART transmission Note 2.	

Note 1. The SCR02 and SCR12 registers 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 SCR03, SCR10, SCR11, or SCR13 register to 0). Be sure to set bits 2 and 1 to "1".

Figure 14 - 9 Format of Serial communication operation setting register mn (SCRmn) (2/2)

Address: F011CH, F011DH (SCR02), F011EH, F011FH (SCR03), F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13) 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	SLC mn1 Note 1	SLC mn0	0	1	1	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.			

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.

SLC mn1 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 = 02, 10, 12 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. 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.		

DLS mn0	Setting of data length in CSI and UART modes
0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)

Note 1. The SCR02 and SCR12 registers 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 SCR03, SCR10, SCR11, or SCR13 register to 0). Be sure to set bits 2 and 1 to "1".

14.3.5 Serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n.

Bits 7 to 0 (lower 8 bits) function 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) 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 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 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8 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 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 14 - 10 Format of Serial data register mn (SDRmn)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03) After reset: 0000H R/W
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)
FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)

	FFF45H (SDR02)								FFF44H (SDR02)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

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. Be sure to clear bit 8 to "0".

Caution 2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

Caution 3. When operation is stopped (SEmn = 0), do not rewrite SDRmn [7:0] by an 8-bit memory manipulation instruction (SDRmn [15:9] are all cleared to 0).

Remark For the function of the lower 8 bits of the SDRmn register, see **14.2 Configuration of Serial Array Unit**.

14.3.6 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 14 - 11 Format of Serial flag clear trigger register mn (SIRmn)

Address: F010CH, F010DH (SIR02), F010EH, F010FH (SIR03), F0148H, F0149H (SIR10) to F014EH, F014FH (SIR13) — 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 SIR03 and SIR13 registers only.

Caution **Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR02, SIR10 to SIR12 register) to "0".**

Remark When the SIRmn register is read, 0000H is always read.

14.3.7 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 14 - 12 Format of Serial status register mn (SSRmn) (1/2)

Address: F0104H, F0105H (SSR02), F0106H, F0107H (SSR03), F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13) 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 SSR03 and SSR13 registers only.

Figure 14 - 13 Format of Serial status register mn (SSRmn) (2/2)

Address: F0104H, F0105H (SSR02), F0106H, F0107H (SSR03), F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13)

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).
<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).	

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 SSR03 and SSR13 registers only.

Caution 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.

14.3.8 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 14 - 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	SS03	SS02	0	0

Address: F0162H, F0163H (SS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13	SS12	SS11	SS10

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 4, 1, and 0 of the SS0 register, and bits 15 to 4 of the SS1 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 When the SSm register is read, 0000H is always read.

14.3.9 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 14 - 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	ST03	ST02	0	0

Address: F0164H, F0165H (ST1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST1	0	0	0	0	0	0	0	0	0	0	0	0	ST13	ST12	ST11	ST10

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 4, 1, and 0 of the ST0 register, and bits 15 to 4 of the ST1 register to "0".**

Remark When the STm register is read, 0000H is always read.

14.3.10 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 14 - 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	SE03	SE02	0	0

Address: F0160H, F0161H (SE1) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1	0	0	0	0	0	0	0	0	0	0	0	0	SE13	SE12	SE11	SE10

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

14.3.11 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 14 - 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	SOE03	SOE02	0	0

Address: F016AH, F016BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	SOE13	SOE12	SOE11	SOE10

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 4, 1, and 0 of the SOE0 register, and bits 15 to 4 of the SOE1 register to "0".

14.3.12 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 0F0FH.

Figure 14 - 18 Format of Serial output register m (SOm)

Address: F0128H, F0129H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	CKO 03	CKO 02	0	0	0	0	0	0	SO 03	SO 02	0	0

Address: F0168H, F0169H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	CKO 13	CKO 12	CKO 11	CKO 10	0	0	0	0	SO 13	SO 12	SO 11	SO 10

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 12, 9 to 4, 1, and 0 of the SO0 register, and bits 15 to 12 and 7 to 4 of the SO1 register to “0”.

14.3.13 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.

<R> This register can be set only in the UART mode. Be sure to set 0 for corresponding bit in the CSI 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 14 - 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	SOL02	0	0

Address: F0174H, F0175H (SOL1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL12	0	SOL10

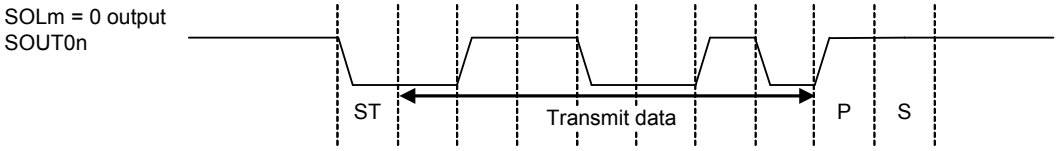
SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 15 to 3, 1, and 0 of the SOL0 register, and bits 15 to 3, and 1 of the SOL1 register to "0".

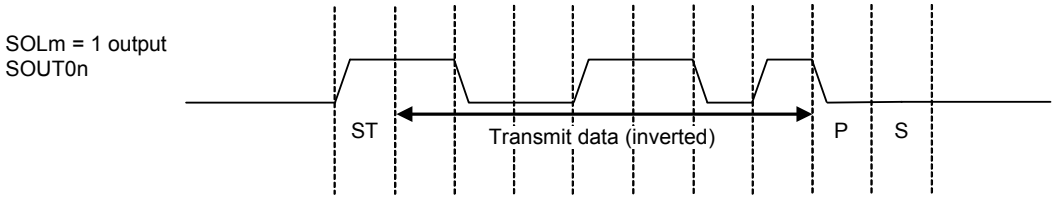
Figure 14 - 20 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 14 - 20 Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



14.3.14 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 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 14 - 21 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	SNFEN30	0	0	0	SNFEN10	0	0

SNFEN30	Use of noise filter of RxD3 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the RxD3 pin. Clear SNFEN30 to 0 to use the other than RxD3 pin.	

SNFEN10	Use of noise filter of RxD1 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN10 bit to 1 to use the RxD1 pin. Clear the SNFEN10 bit to 0 to use the other than RxD1 pin.	

14.3.15 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), port output mode register (POMxx), port mode control register (PMCxx)).

For details, see **5.3.1 Port mode registers (PMxx)**, **5.3.2 Port registers (Pxx)**, **5.3.4 Port input mode registers (PIMxx)**, **5.3.5 Port output mode registers (POMxx)**.

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P02/SO10/TxD1) for serial data or serial clock output, requires setting the corresponding bits in the port mode control register (PMCxx) and 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 **5.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Example When P02/SO10/TxD1 is to be used for serial data output
Set the PMC02 bit of port mode control register 0 to 0.
Set the PM02 bit of port mode register 0 to 0.
Set the P02 bit of port register 0 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P03/SI10/RxD1) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PMxx) to 1, and the corresponding bit in the port mode control register (PMCxx) to 0. 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 **5.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Example When P03/SI10/RxD1 is to be used for serial data input
Set the PMC03 bit of port mode control register 0 to 0.
Set the PM03 bit of port mode register 0 to 1.
Set the P03 bit of port register 0 to 0 or 1.

14.4 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.

14.4.1 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.

To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

Figure 14 - 22 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	RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
	×	×	×	×	0/1	0/1	×	×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Caution **If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read**
Note that this does not apply to the following registers.
- **Noise filter enable register 0 (NFEN0)**
- **Port input mode registers 0, 14 (PIM0, PIM14)**
- **Port output mode registers 0, 14 (POM0, POM14)**
- **Port mode registers 0, 1, 7, 14 (PM0, PM1, PM7, PM14)**
- **Port registers 0, 1, 7, 14 (P0, P1, P7, P14)**

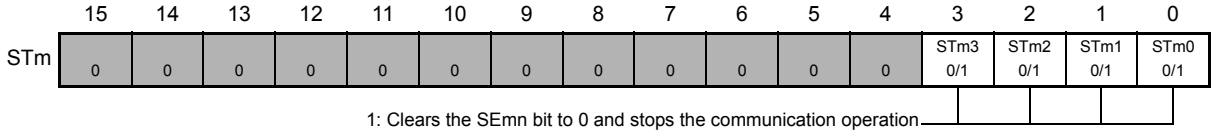
Remark ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

14.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

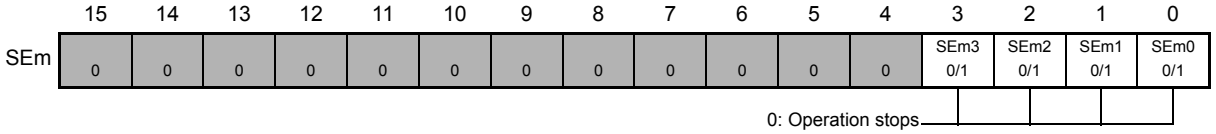
Figure 14 - 23 Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm)... This register is a trigger register that is used to enable stopping communication/count by each channel.



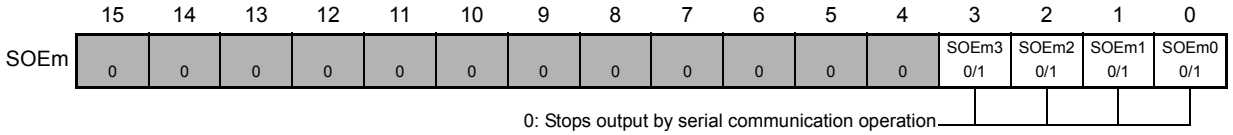
* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm)... This register indicates whether data transmission/reception operation of each channel is enabled or stopped.



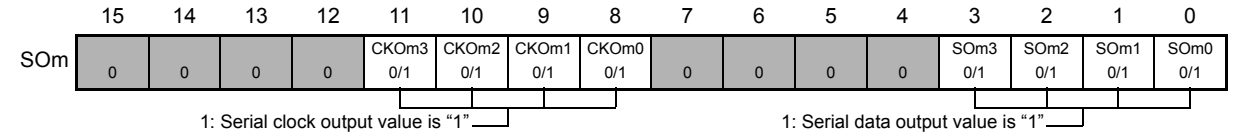
* The SEm register is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(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 : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

14.5 Operation of 3-Wire Serial I/O (CSIp) 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. $f_{CLK}/4$
  - During slave communication: Max. $f_{MCK}/6$

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt

[Error detection flag]
- Overrun error

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 31 ELECTRICAL SPECIFICATIONS**.

The channels supporting 3-wire serial I/O (CSIp) are channel 2 of SAU0 and channels 0 to 2 of SAU1.

Unit	Channel	Used as CSI	Used as UART
0	0	Cannot be used	Cannot be used
	1		
	2	CSI10	UART1
	3	Cannot be used	
1	0	CSI20 Note	Cannot be used
	1	CSI21	
	2	CSI30	UART3
	3	Cannot be used	

Note CSI20 is used for internal communication between MCU and RF transceiver. For this channel, only the master can be selected.

3-wire serial I/O (CSIp) performs the following seven types of communication operations.

- Master transmission (See **14.5.1**.)
- Master reception (See **14.5.2**.)
- Master transmission/reception (See **14.5.3**.)
- Slave transmission (See **14.5.4**.)
- Slave reception (See **14.5.5**.)
- Slave transmission/reception (See **14.5.6**.)

14.5.1 Master transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

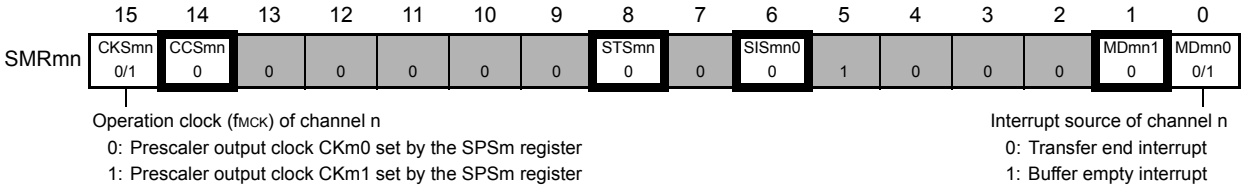
3-Wire Serial I/O	CSI10	CSI20	CSI21	CSI30
Target channel	Channel 2 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1
Pins used	SCK10, SO10	SCK20, SO20	SCK21, SO21	SCK30, SO30
Interrupt	INTCSI10	INTCSI20	INTCSI21	INTCSI30
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	7 or 8 bits			
Transfer rate **Note**	Max. $f_{CLK}/4$ [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 **CHAPTER 31 ELECTRICAL SPECIFICATIONS**).

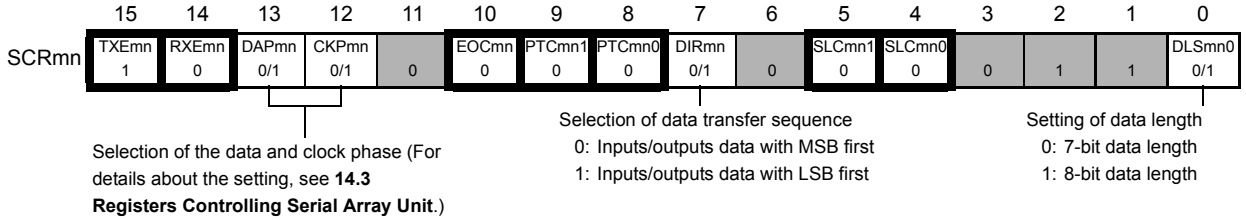
(1) Register setting

Figure 14 - 24 Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSIp)

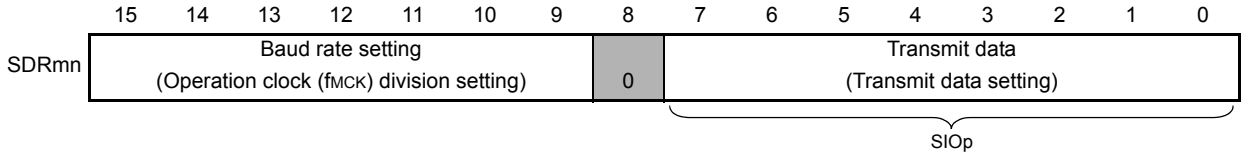
(a) Serial mode register mn (SMRmn)



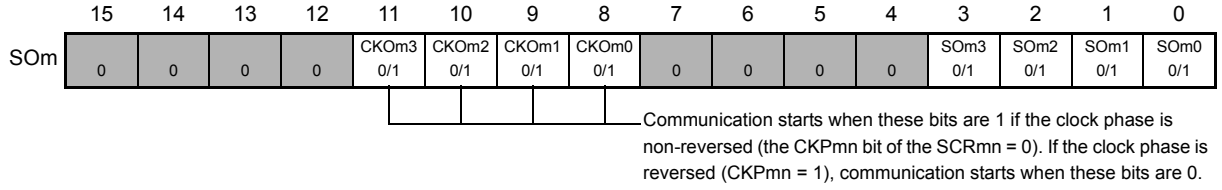
(b) Serial communication operation setting register mn (SCRmn)



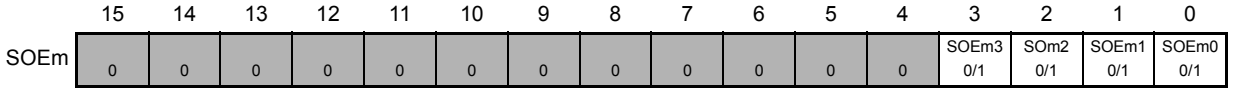
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



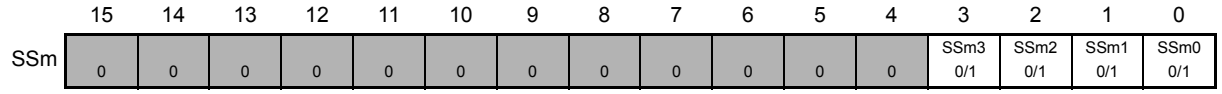
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(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 : 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 14 - 25 Initial Setting Procedure for Master Transmission

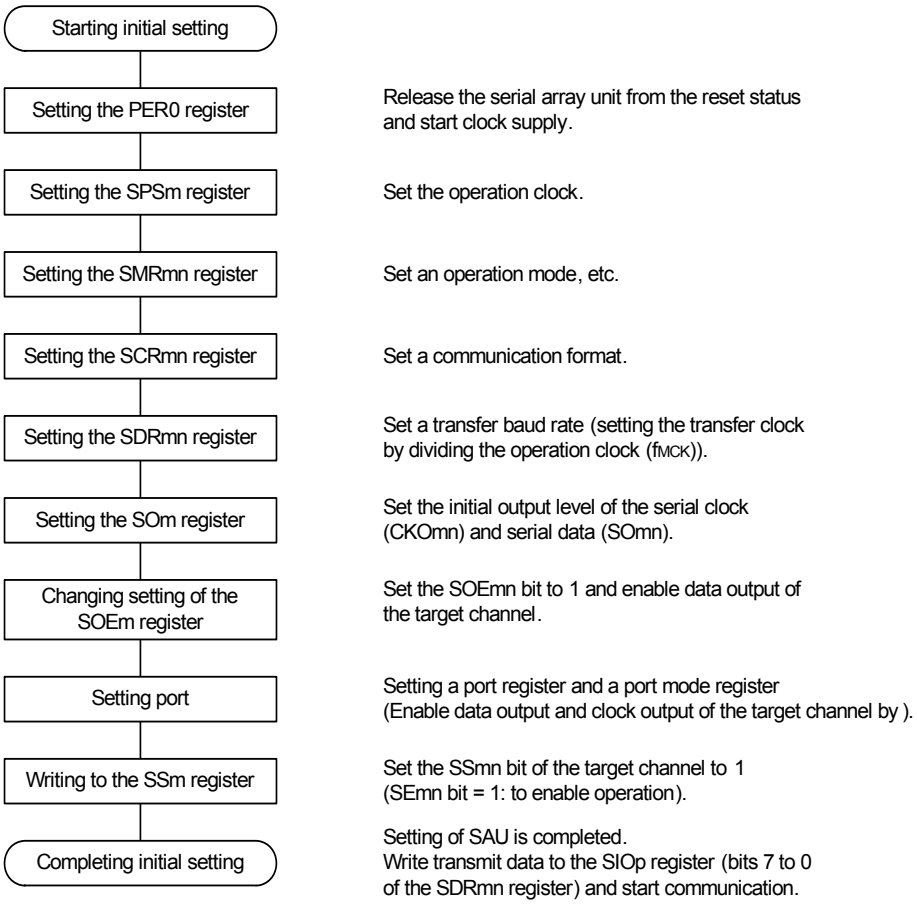


Figure 14 - 26 Procedure for Stopping Master Transmission

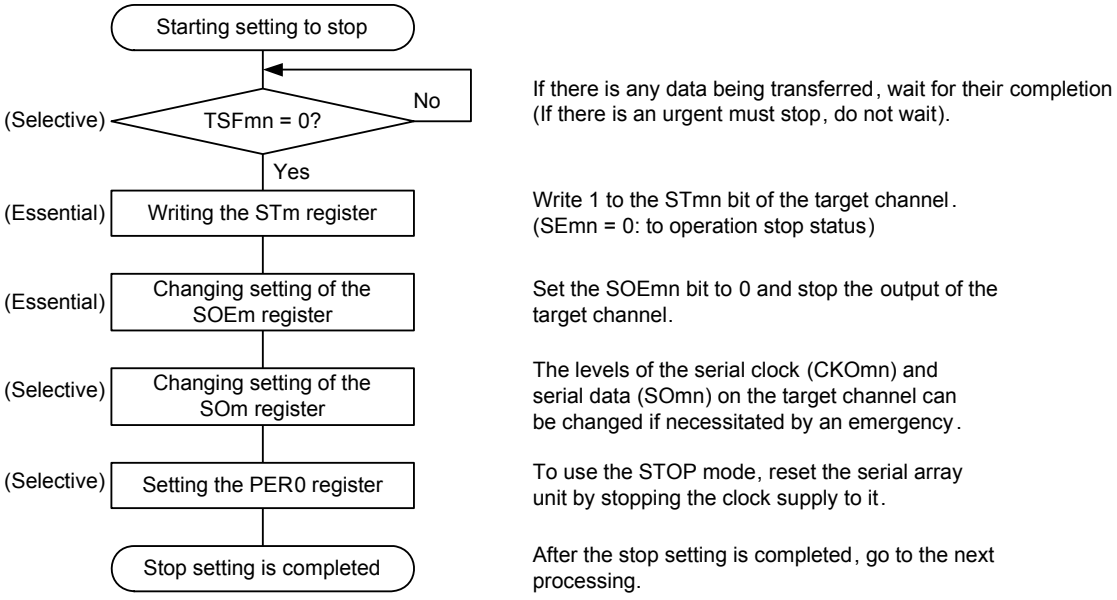
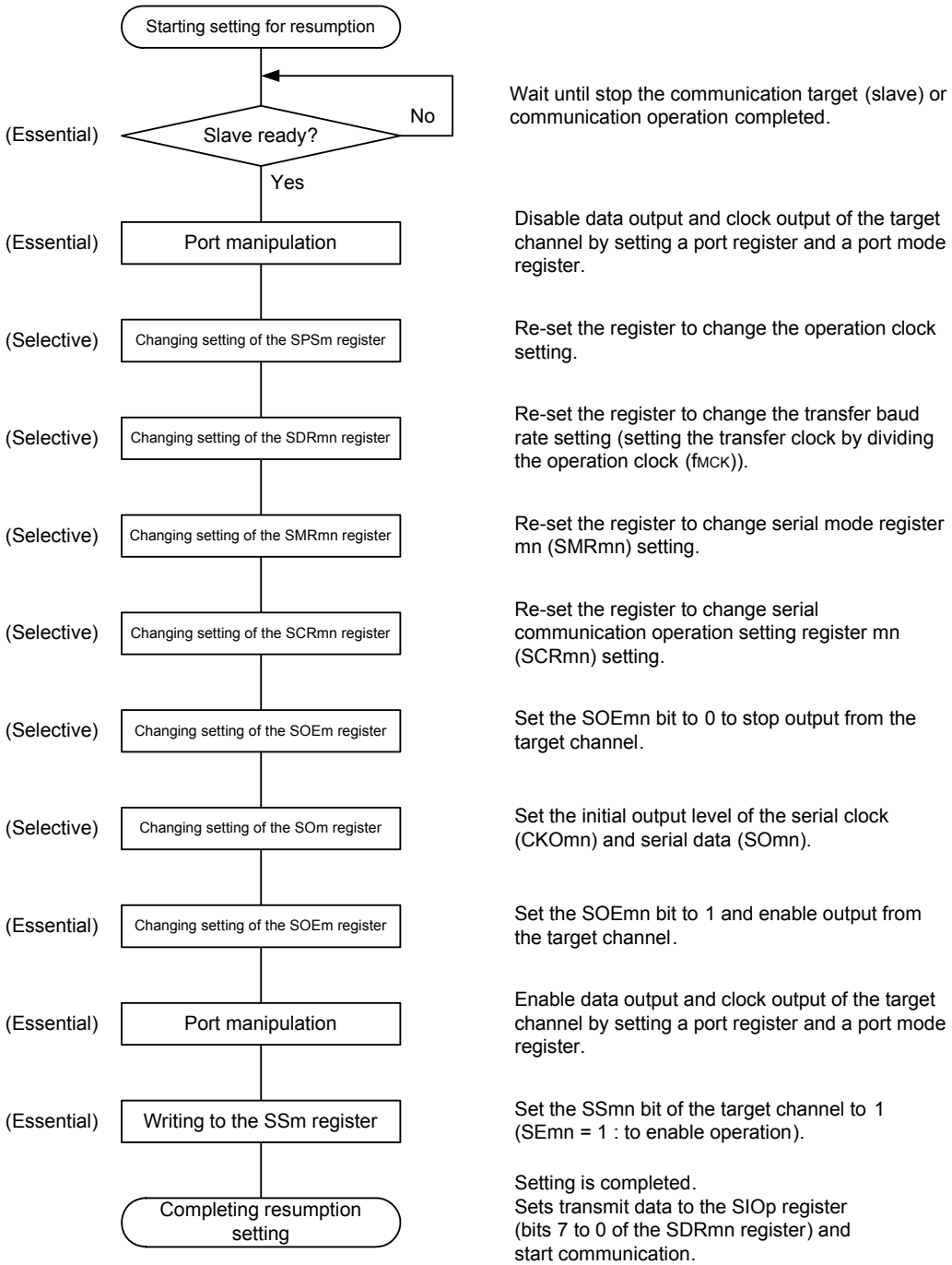


Figure 14 - 27 Procedure for Resuming Master Transmission



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 14 - 28 Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

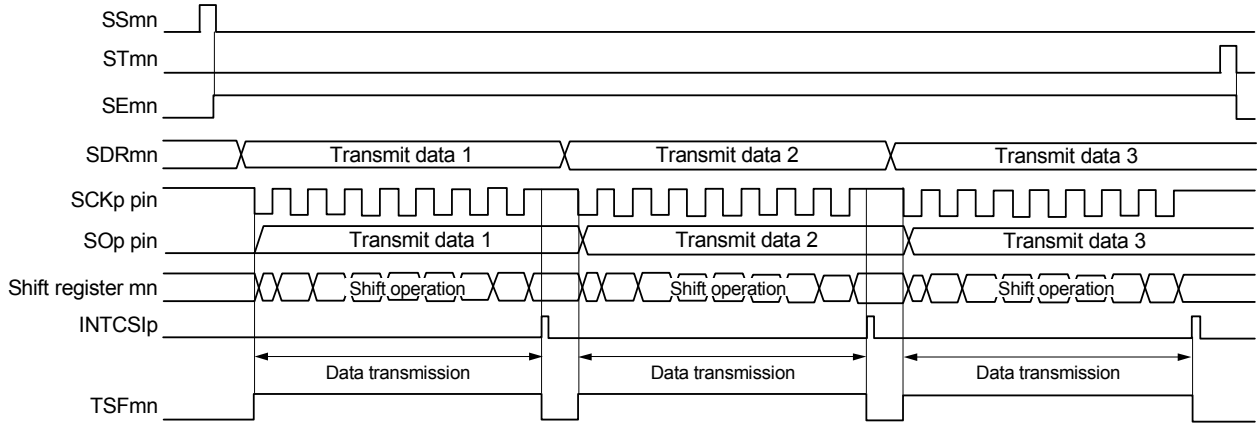
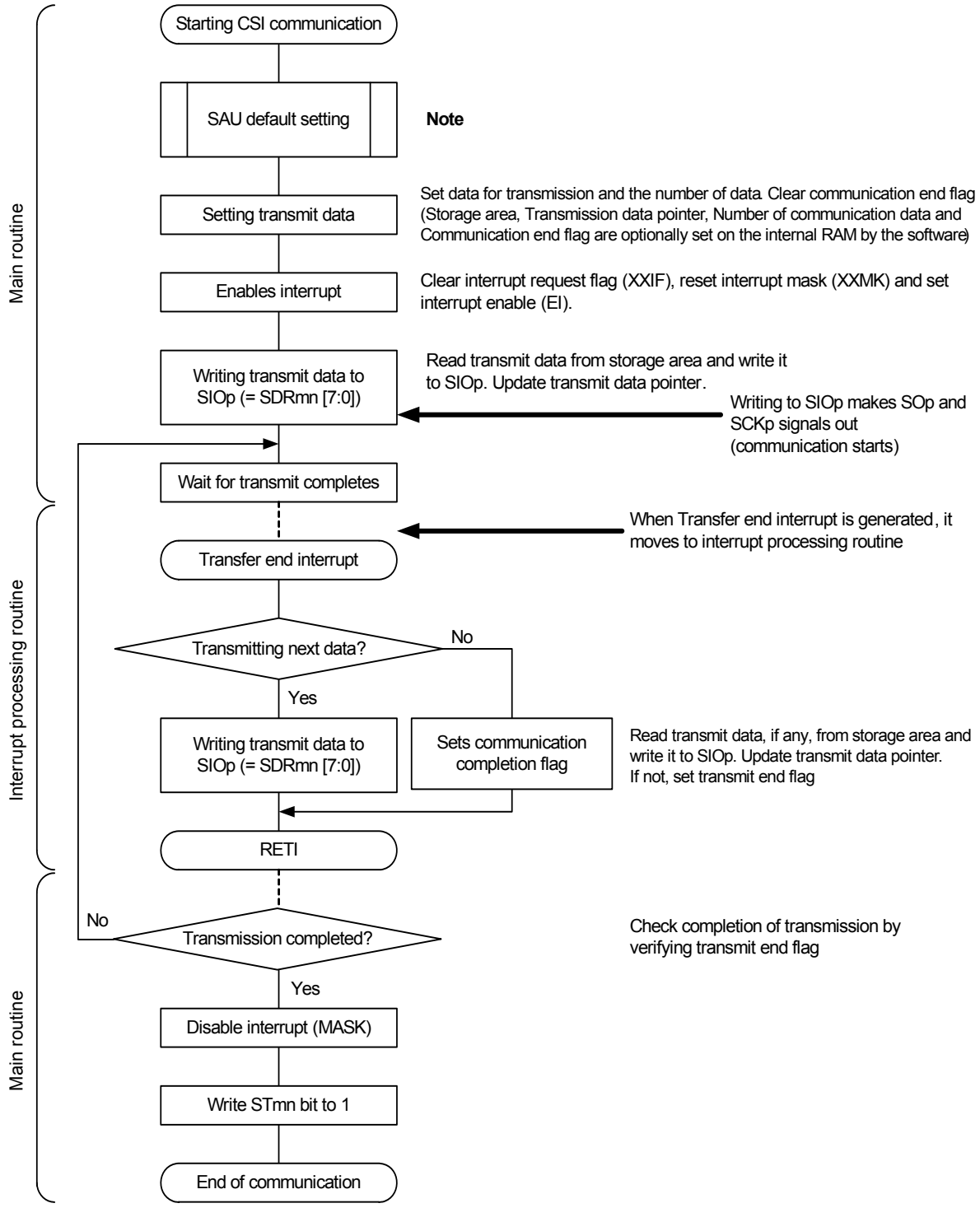


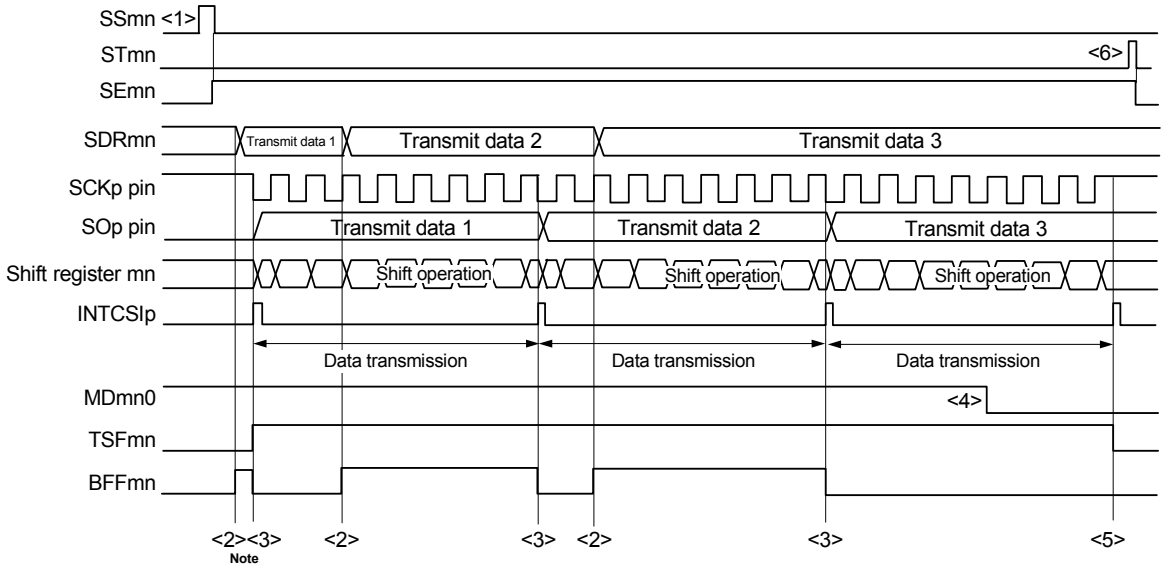
Figure 14 - 29 Flowchart of Master Transmission (in Single-Transmission Mode)



Note For the initial setting, refer to **Figure 14 - 25**.(Select Transfer end interrupt)

(4) Processing flow (in continuous transmission mode)

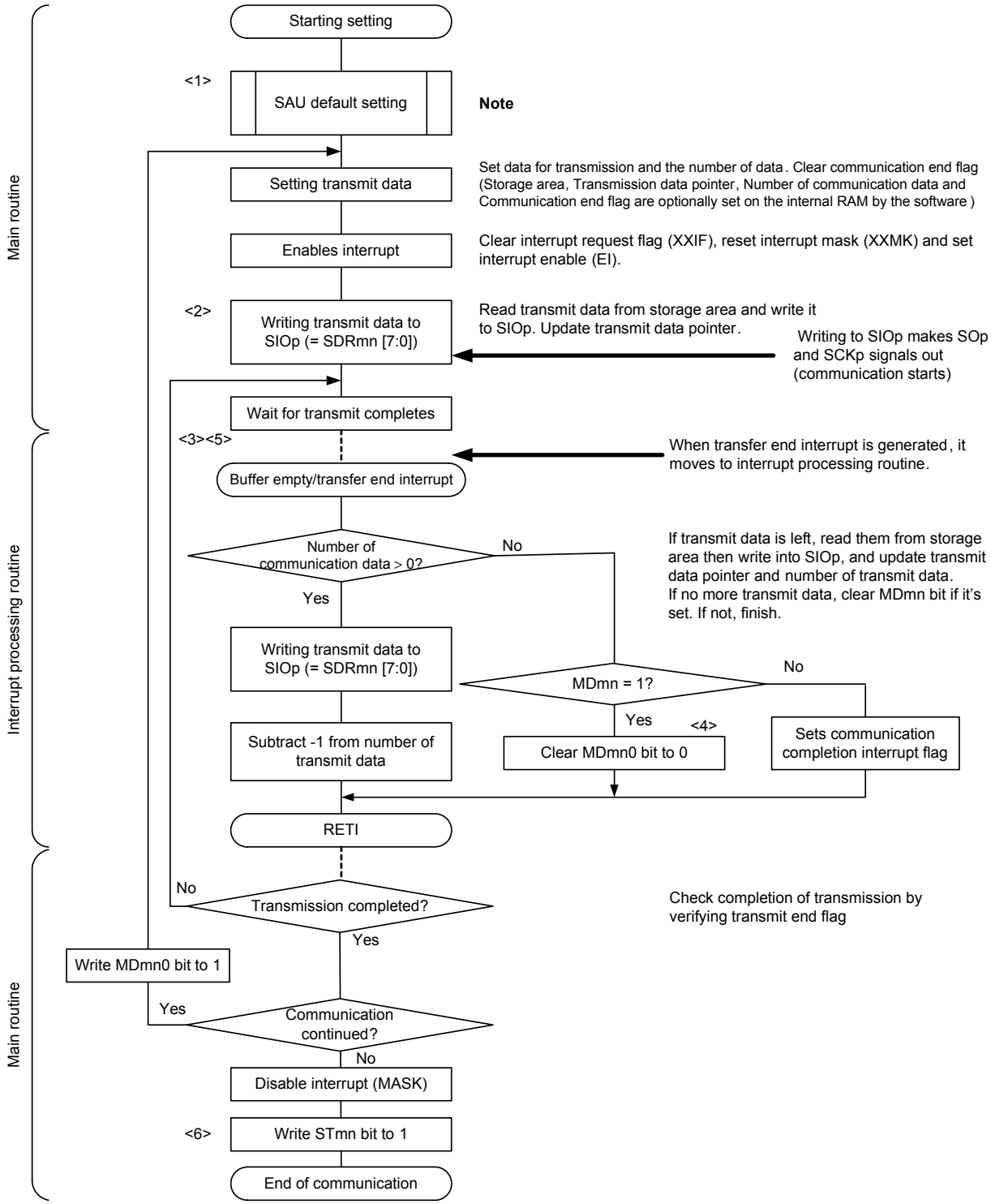
Figure 14 - 30 Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



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.

Figure 14 - 31 Flowchart of Master Transmission (in Continuous Transmission Mode)



Note For the initial setting, refer to **Figure 14 - 25**.(Select buffer empty interrupt)

Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 14 - 30 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

14.5.2 Master reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from other device.

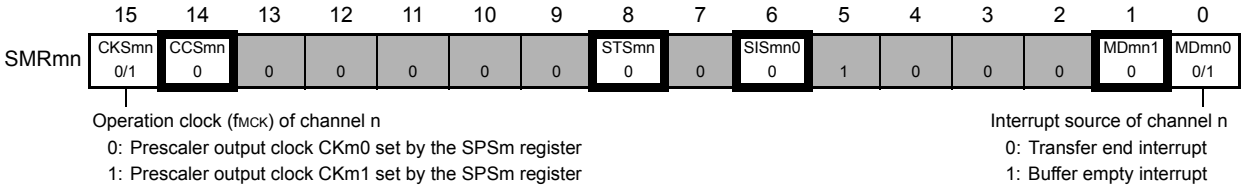
3-Wire Serial I/O	CSI10	CSI20	CSI21	CSI30
Target channel	Channel 2 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1
Pins used	SCK10, SI10	SCK20, SI20	SCK21, SI21	SCK30, SI30
Interrupt	INTCSI10	INTCSI20	INTCSI21	INTCSI30
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (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. $f_{CLK}/4$ [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 CHAPTER 31 **ELECTRICAL SPECIFICATIONS**).

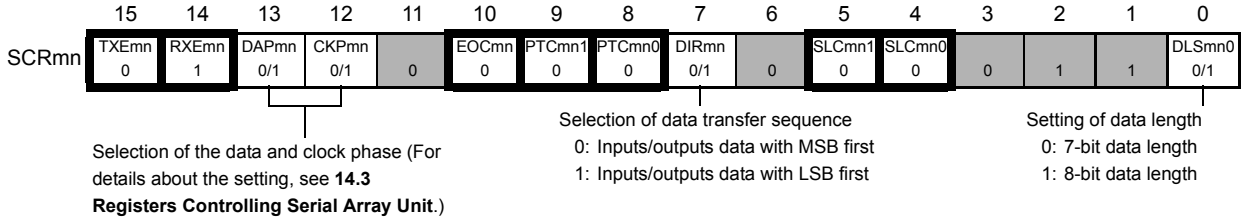
(1) Register setting

Figure 14 - 32 Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSIp)

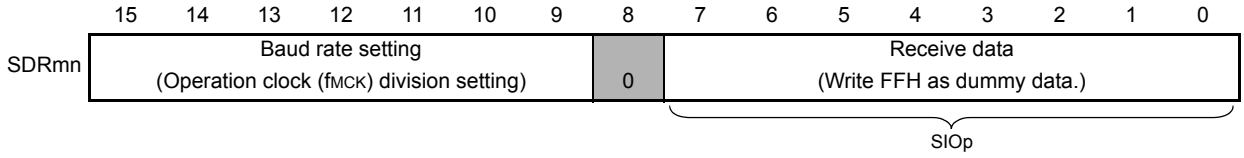
(a) Serial mode register mn (SMRmn)



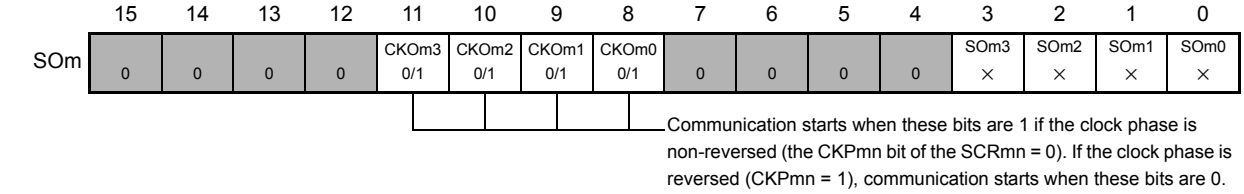
(b) Serial communication operation setting register mn (SCRmn)



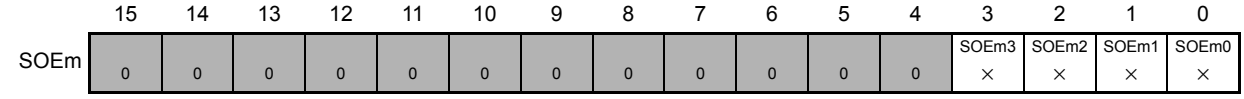
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



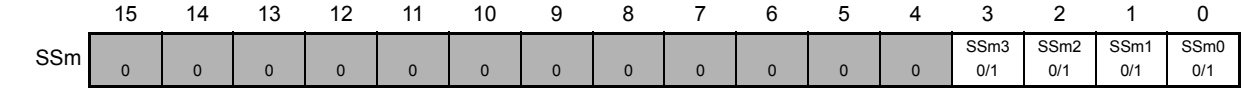
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(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 : 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 14 - 33 Initial Setting Procedure for Master Reception

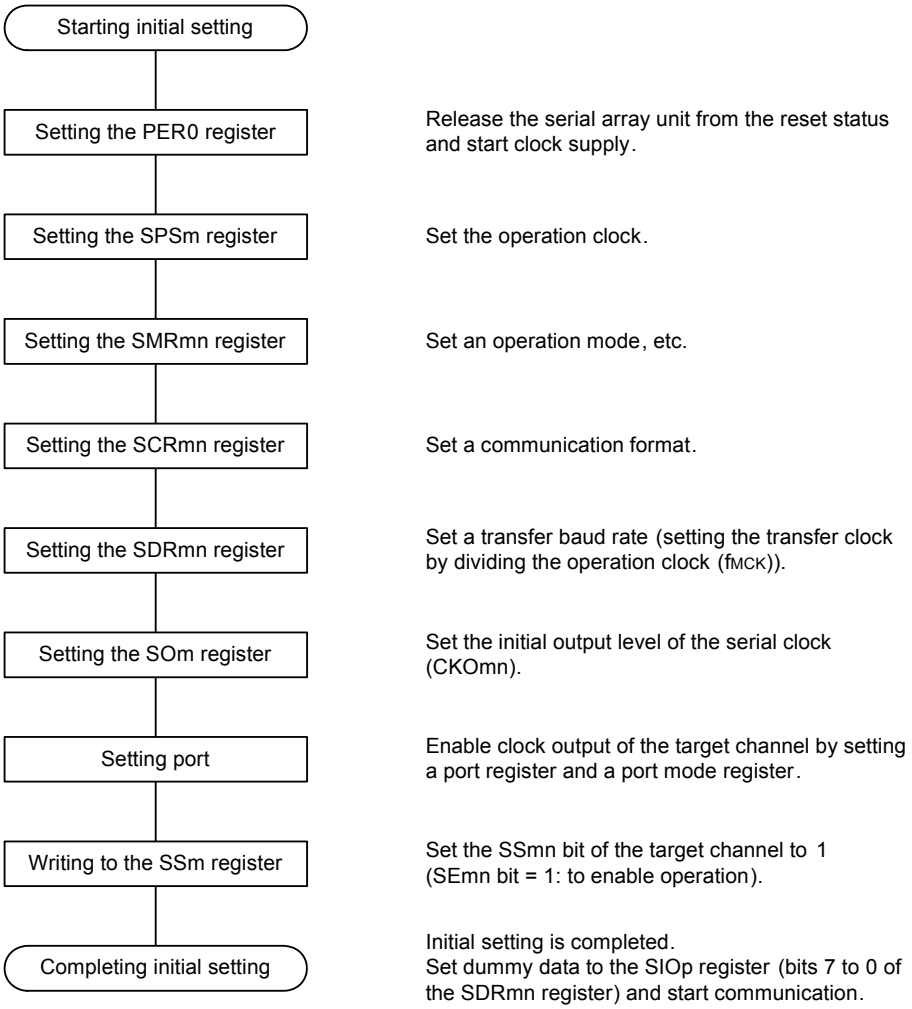


Figure 14 - 34 Procedure for Stopping Master Reception

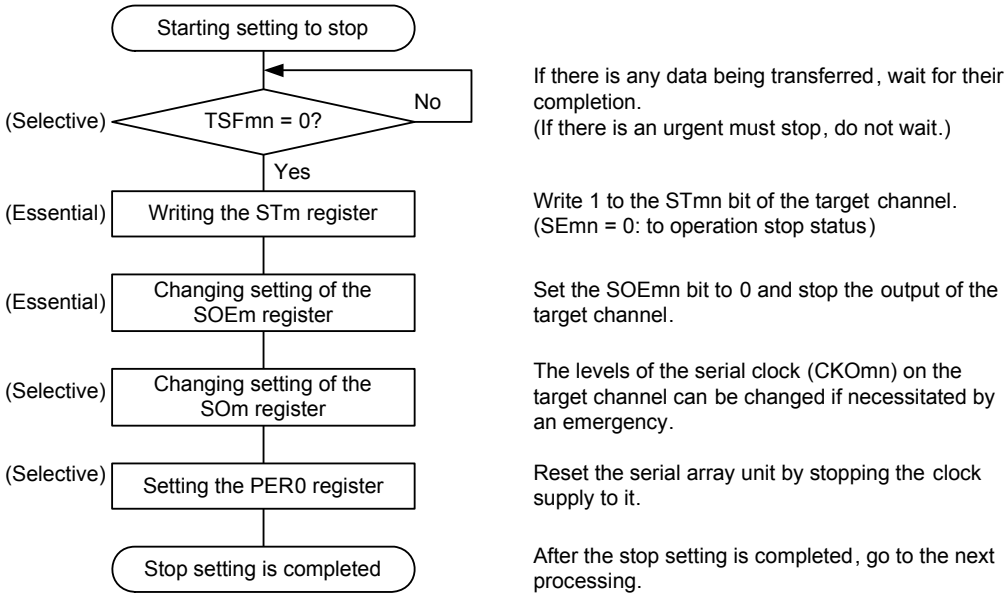
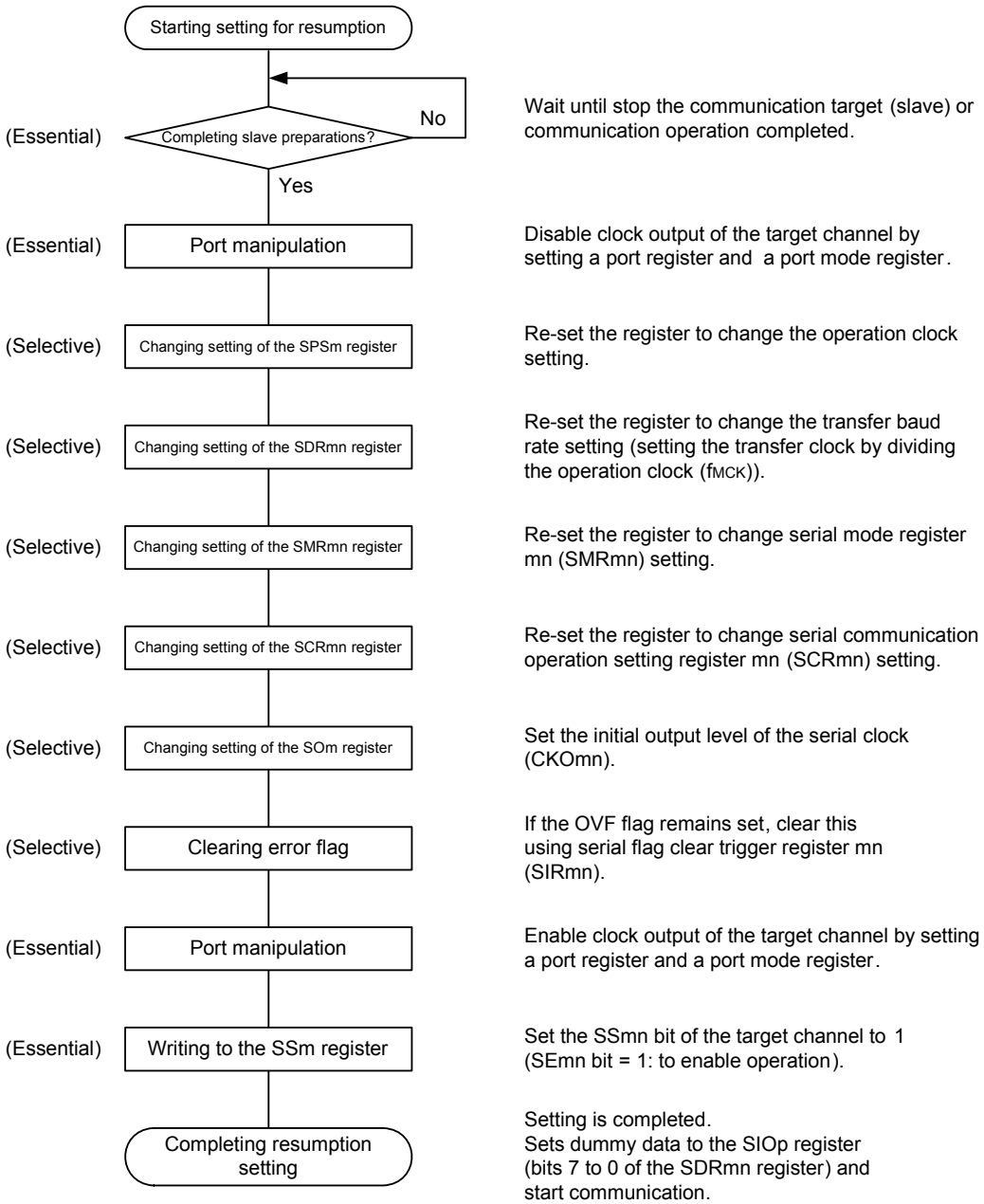


Figure 14 - 35 Procedure for Resuming Master Reception



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 14 - 36 Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

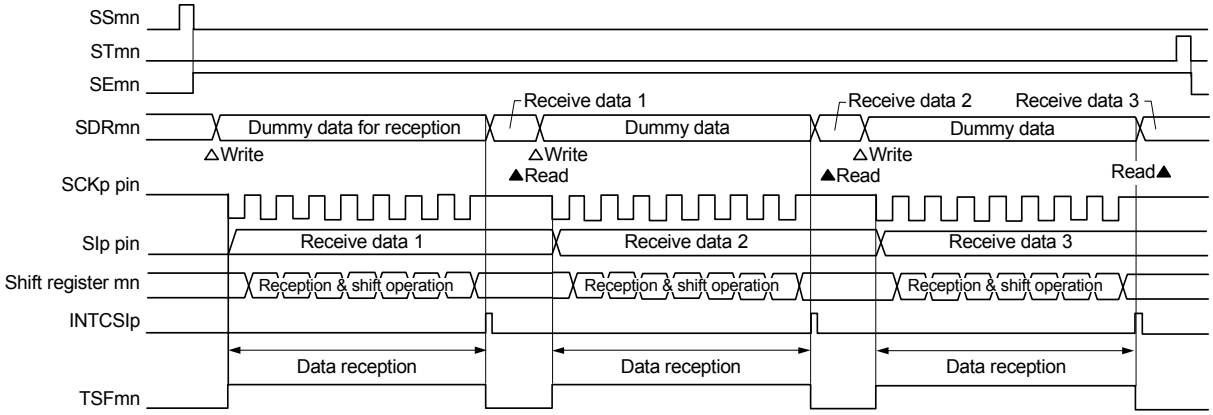
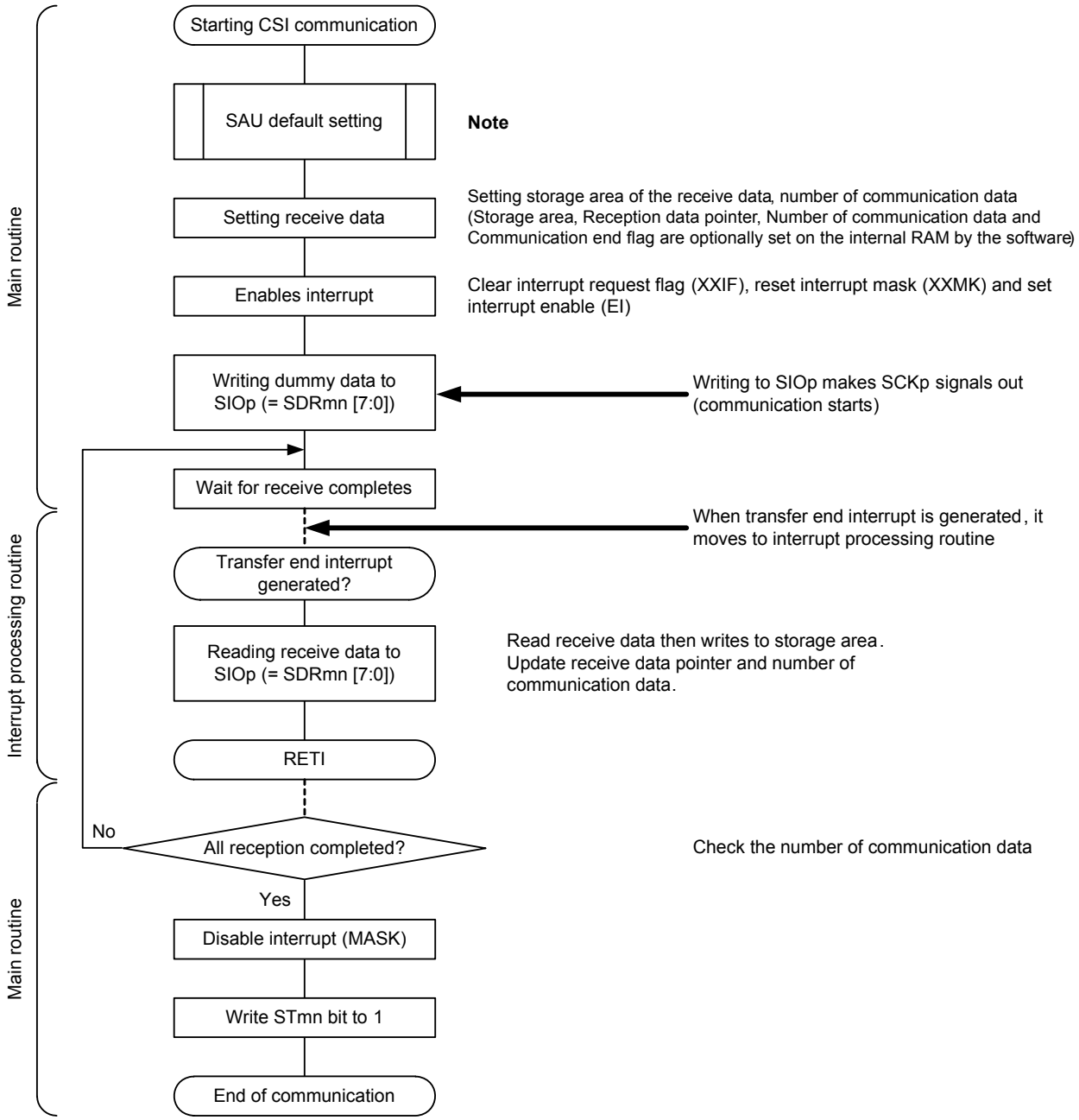


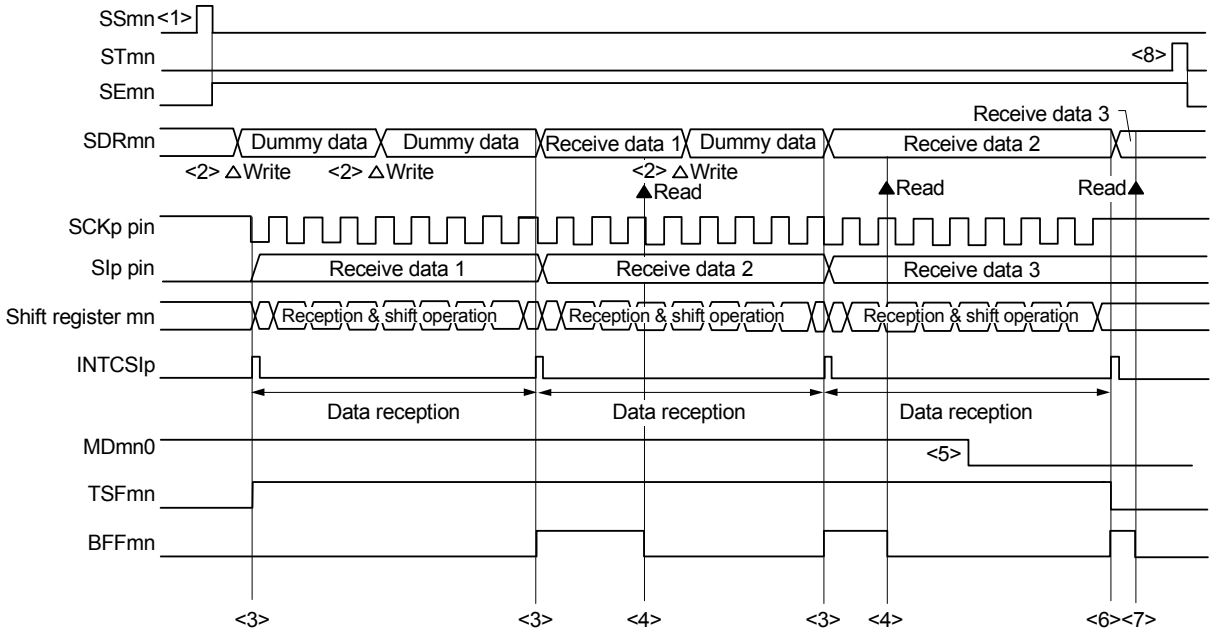
Figure 14 - 37 Flowchart of Master Reception (in Single-Reception Mode)



Note For the initial setting, refer to **Figure 14 - 33**.(Select Transfer end interrupt)

(4) Processing flow (in continuous reception mode)

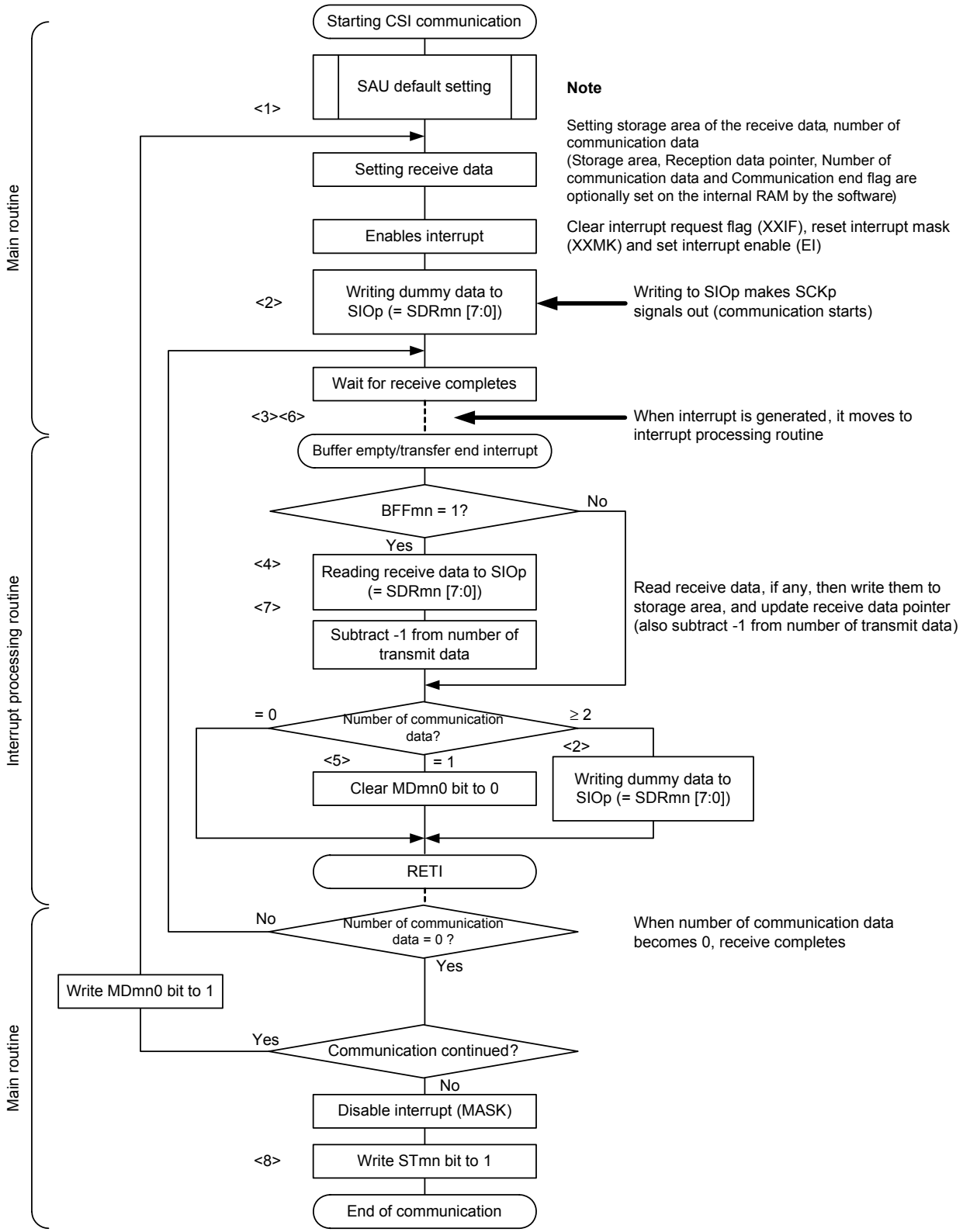
Figure 14 - 38 Timing Chart of Master Reception (in Continuous Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



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> to <8> in the figure correspond to <1> to <8> in Figure 14 - 39 Flowchart of Master Reception (in Continuous Reception Mode).

Figure 14 - 39 Flowchart of Master Reception (in Continuous Reception Mode)



Note For the initial setting, refer to **Figure 14 - 33**.(Select buffer empty interrupt)

Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 14 - 38 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

14.5.3 Master transmission/reception

Master transmission/reception is that the RL78 microcontroller outputs a transfer clock and transmits/receives data to/from other device.

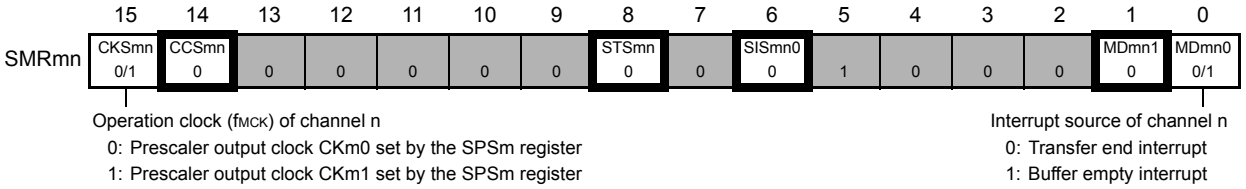
3-Wire Serial I/O	CSI10	CSI20	CSI21	CSI30
Target channel	Channel 2 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1
Pins used	SCK10, SI10, SO10	SCK20, SI20, SO20	SCK21, SI21, SO21	SCK30, SI30, SO30
Interrupt	INTCSI10	INTCSI20	INTCSI21	INTCSI30
	Transfer end interrupt (in single-transfer 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. $f_{CLK}/4$ [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 **CHAPTER 31 ELECTRICAL SPECIFICATIONS**).

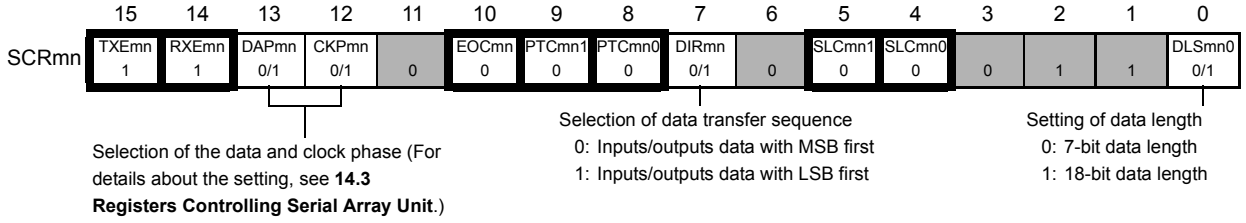
(1) Register setting

Figure 14 - 40 Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSIp)

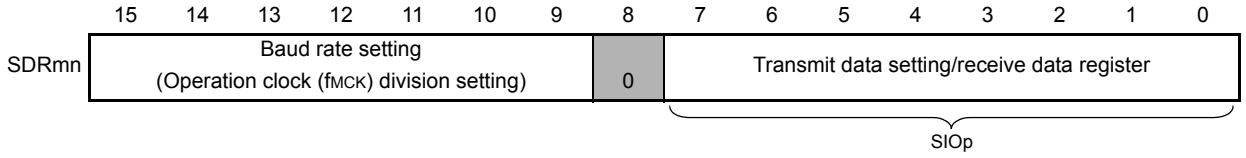
(a) Serial mode register mn (SMRmn)



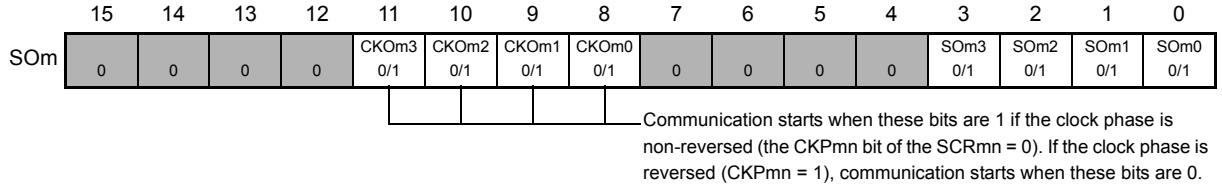
(b) Serial communication operation setting register mn (SCRmn)



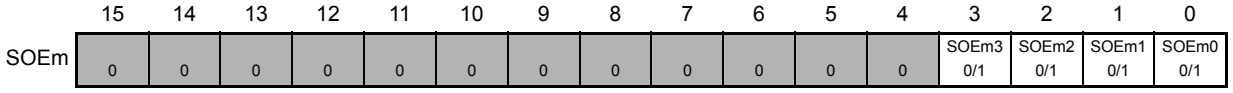
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



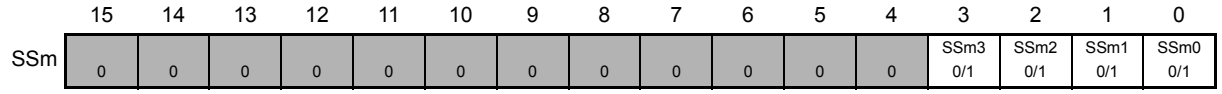
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(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 : 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 14 - 41 Initial Setting Procedure for Master Transmission/Reception

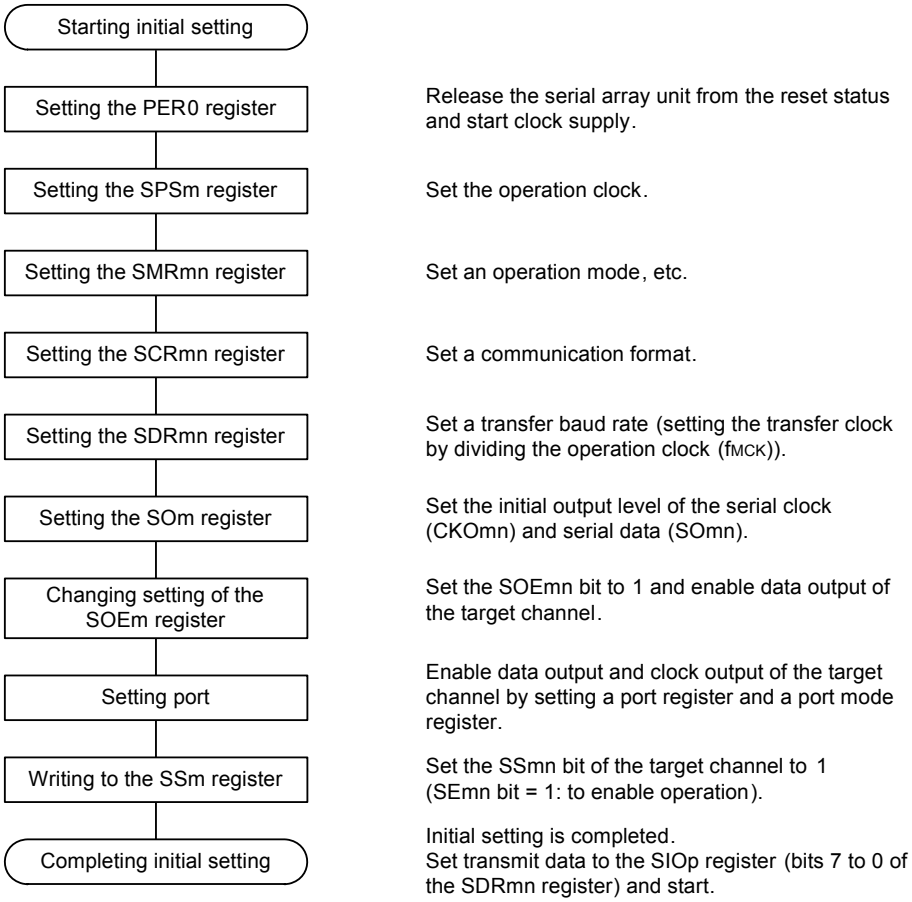


Figure 14 - 42 Procedure for Stopping Master Transmission/Reception

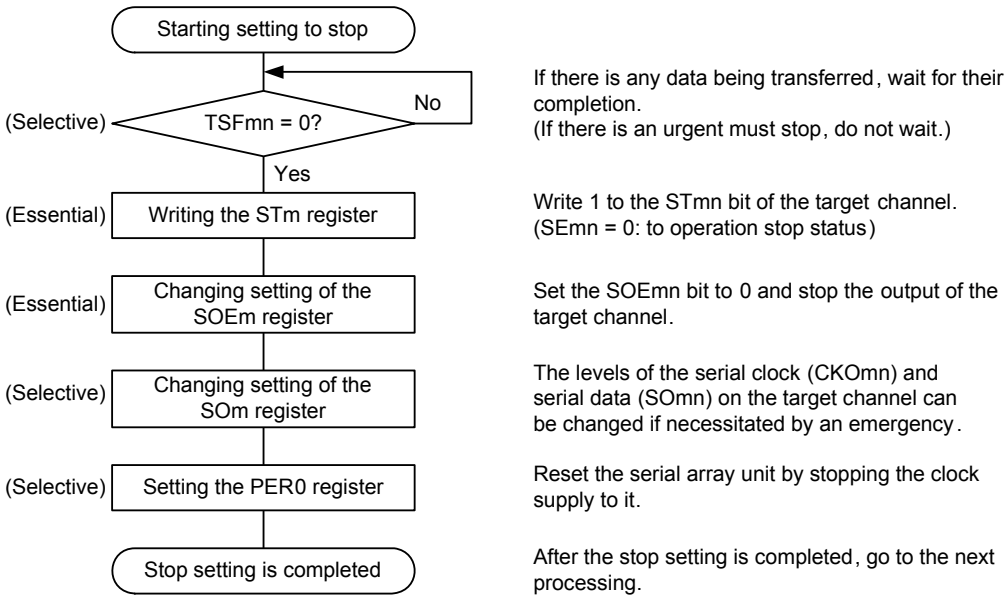
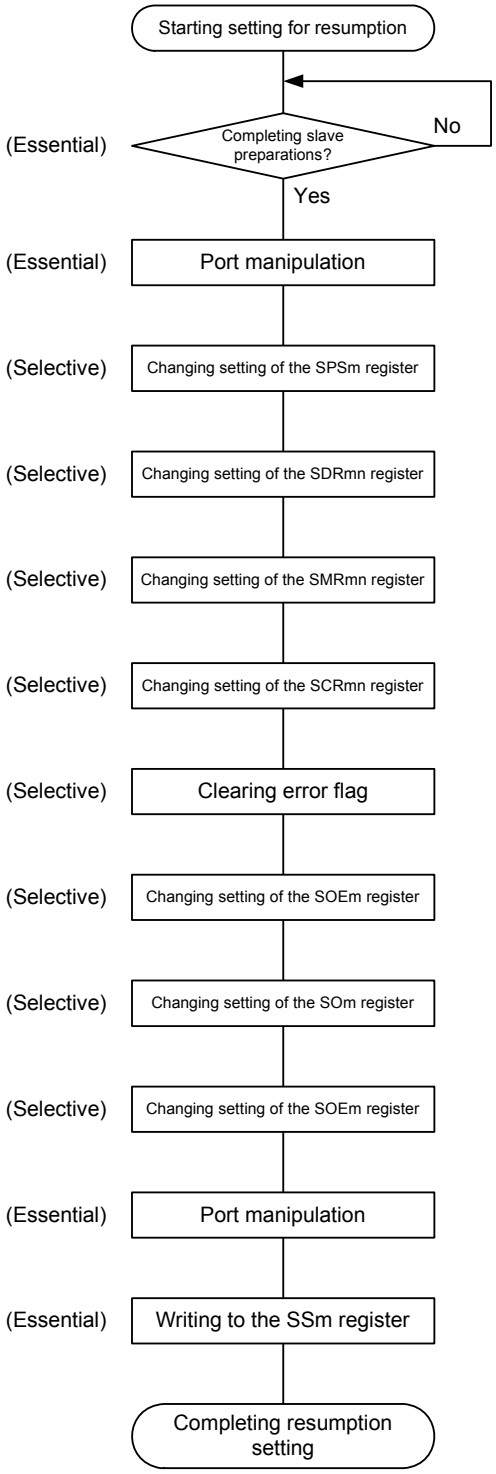


Figure 14 - 43 Procedure for Resuming Master Transmission/Reception



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 14 - 44 Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

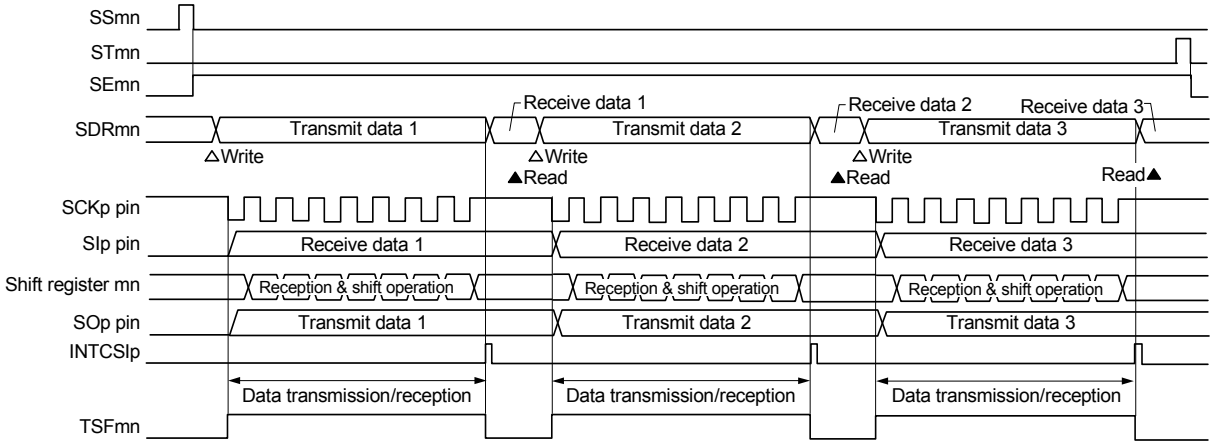
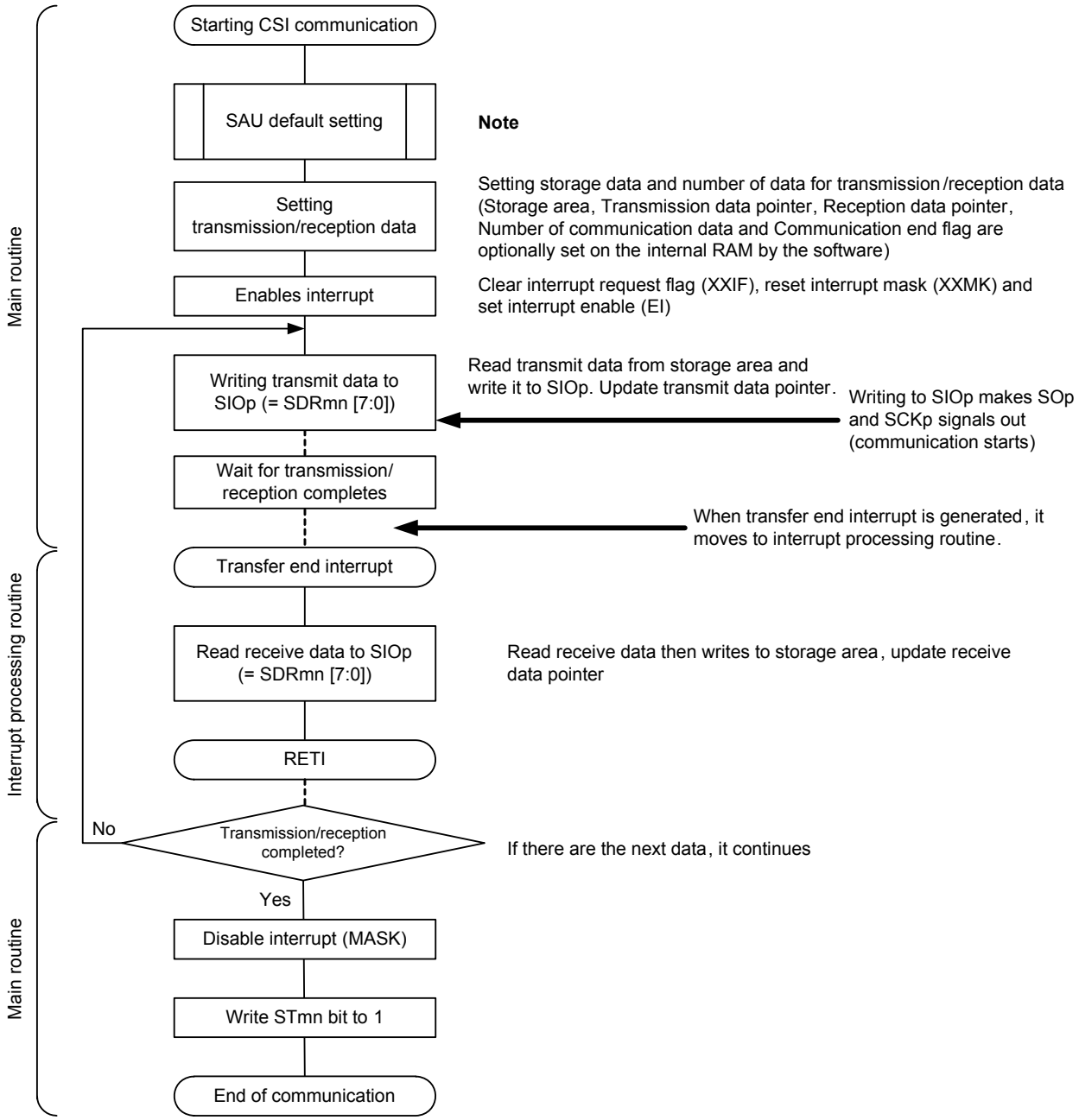


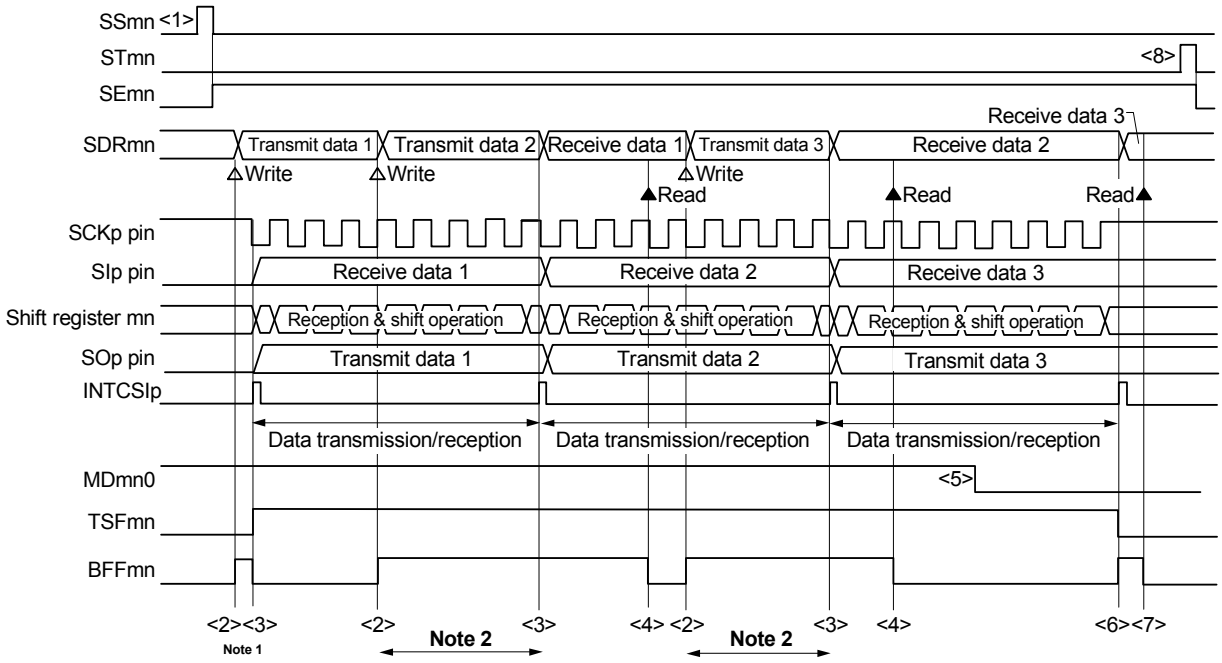
Figure 14 - 45 Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



Note For the initial setting, refer to **Figure 14 - 41**.(Select transfer end interrupt)

(4) Processing flow (in continuous transmission/reception mode)

Figure 14 - 46 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



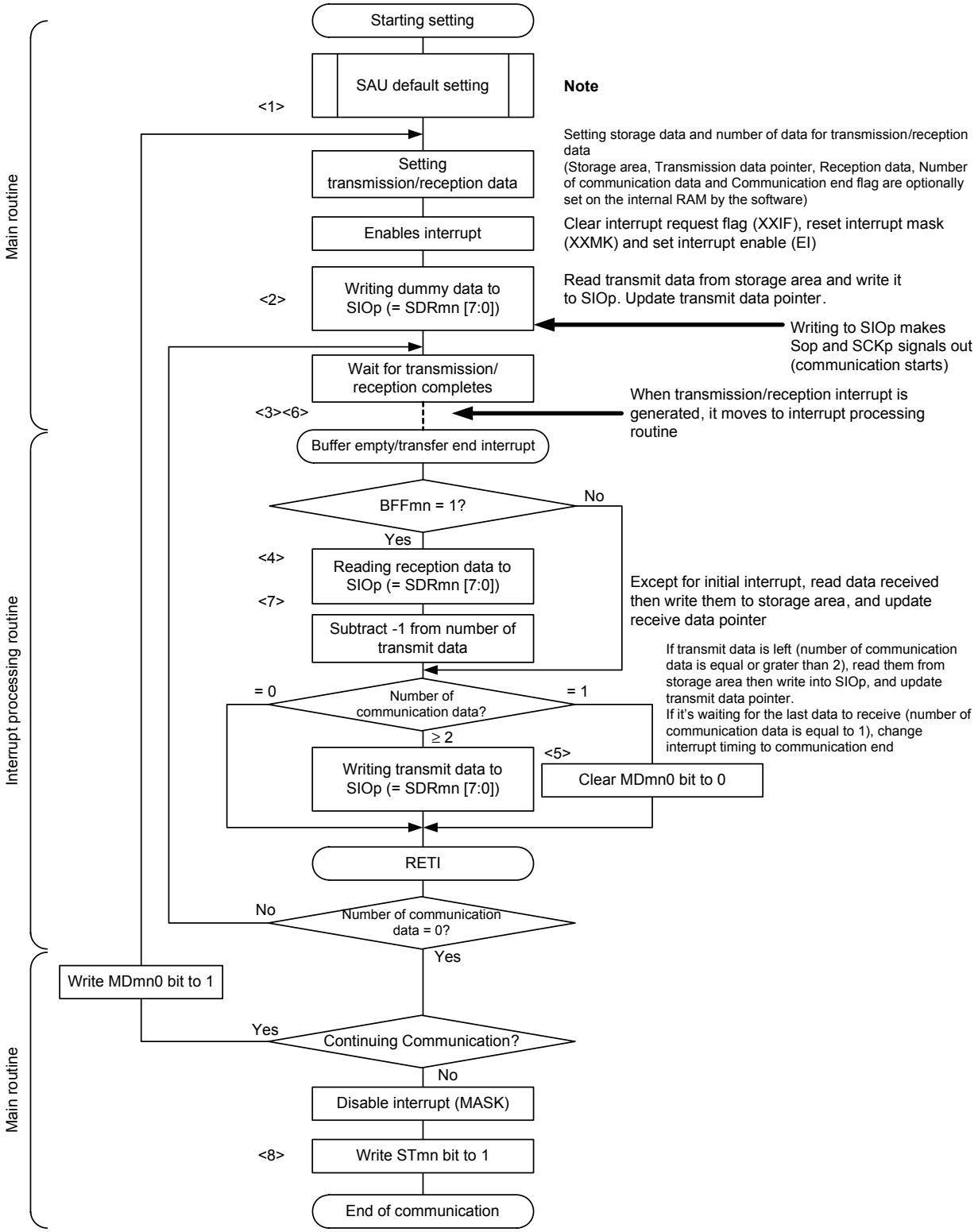
Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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> to <8> in the figure correspond to <1> to <8> in Figure 14 - 47 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

Figure 14 - 47 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Note For the initial setting, refer to **Figure 14 - 41**.(Select buffer empty interrupt)

Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 14 - 46 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

14.5.4 Slave transmission

Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI10	CSI21	CSI30
Target channel	Channel 2 of SAU0	Channel 1 of SAU1	Channel 2 of SAU1
Pins used	SCK10, SO10	SCK21, SO21	SCK30, SO30
Interrupt	INTCSI10	INTCSI21	INTCSI30
	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 SCK10, SCK21, and SCK30 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

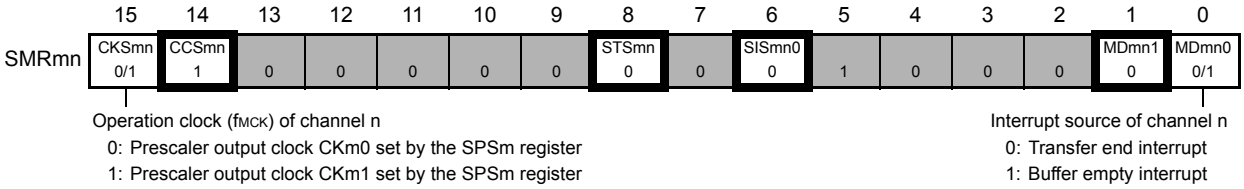
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel

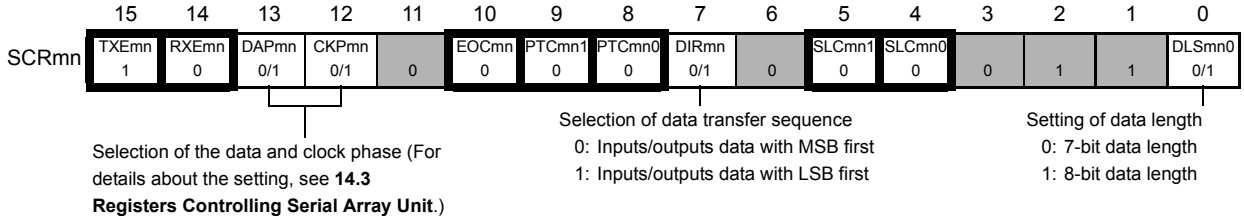
(1) Register setting

Figure 14 - 48 Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSIp)

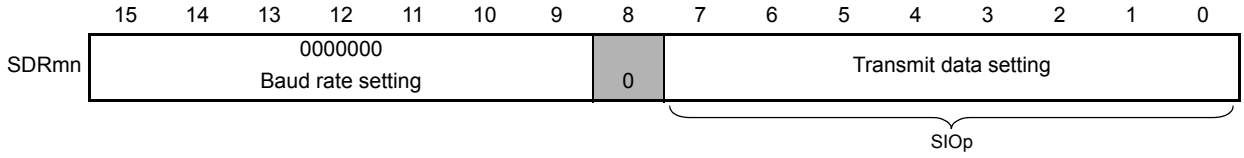
(a) Serial mode register mn (SMRmn)



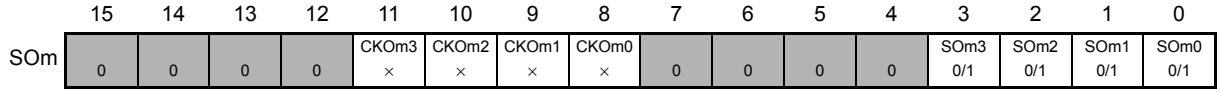
(b) Serial communication operation setting register mn (SCRmn)



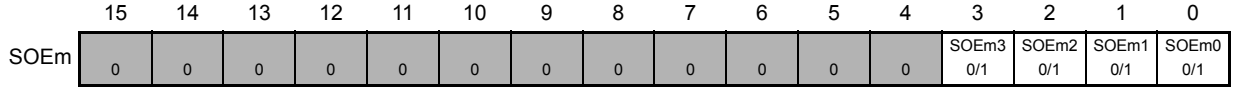
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



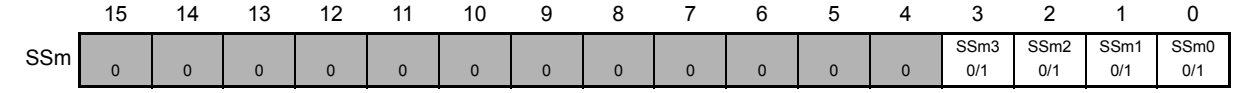
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(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 ■: 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 14 - 49 Initial Setting Procedure for Slave Transmission

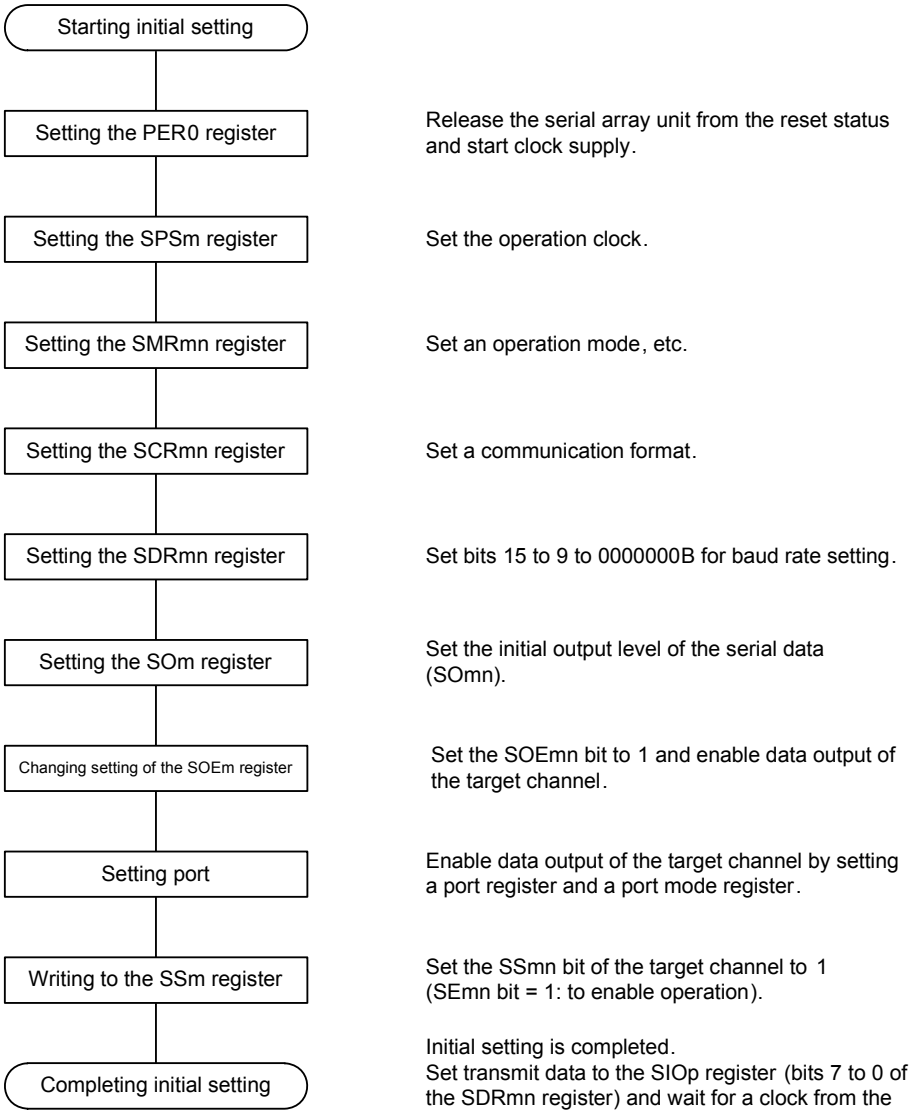


Figure 14 - 50 Procedure for Stopping Slave Transmission

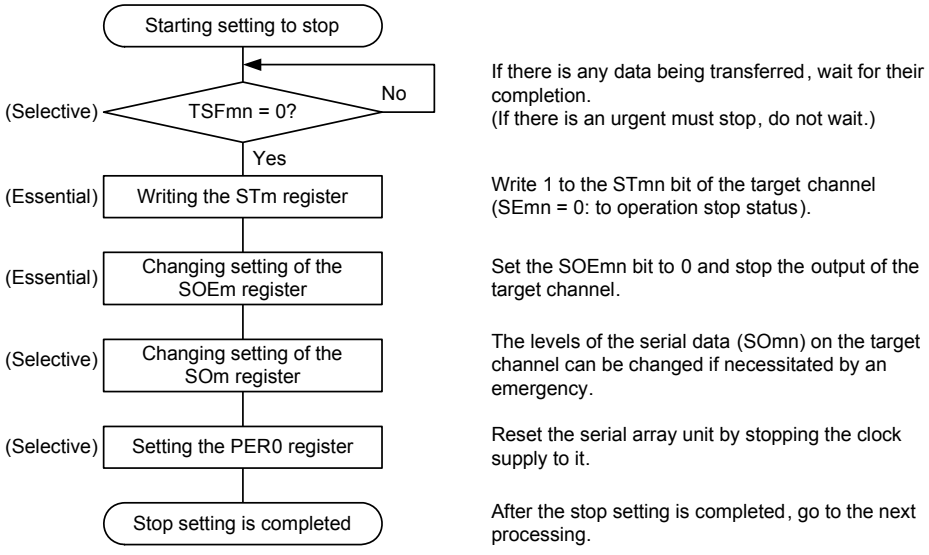
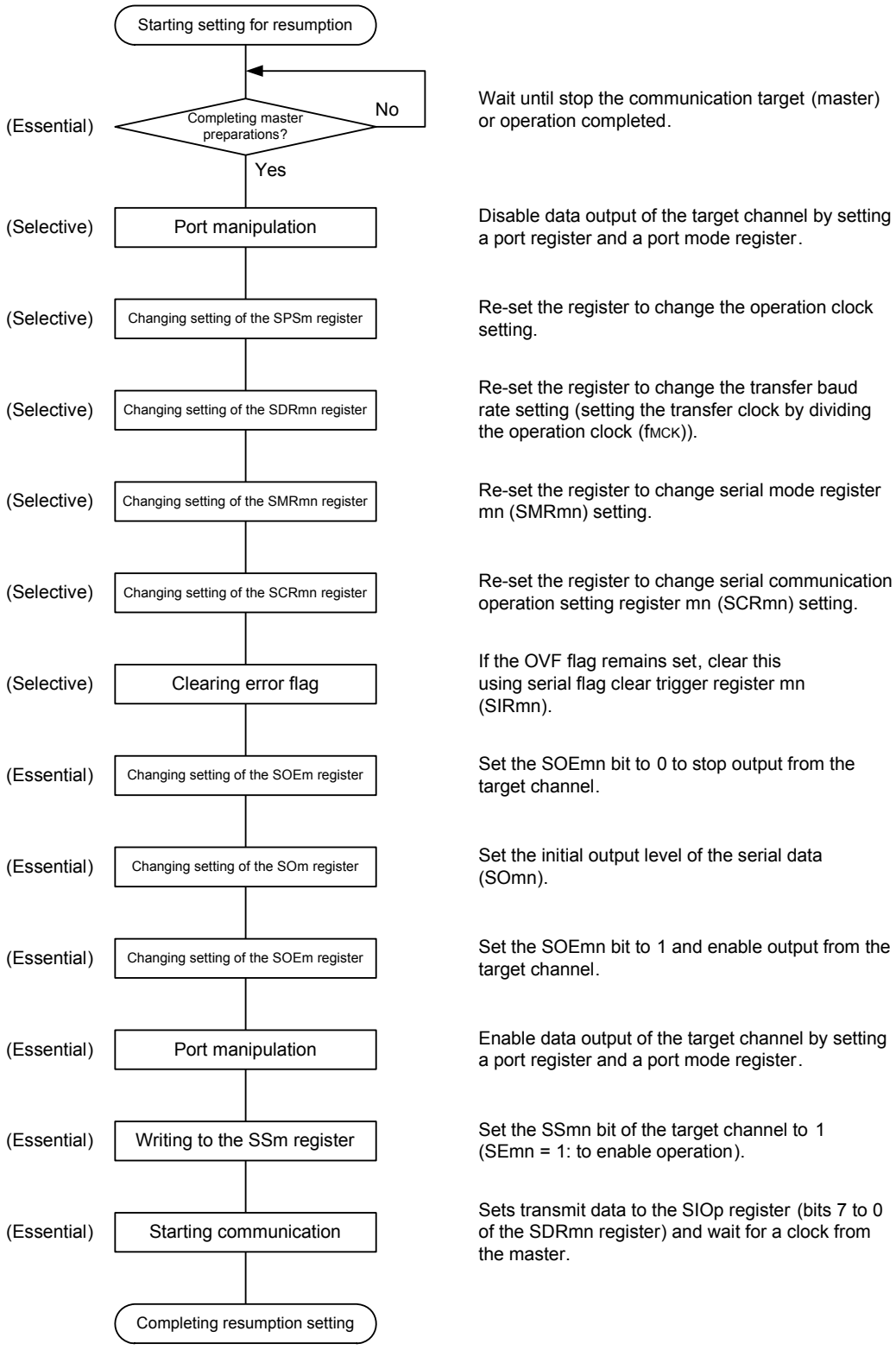


Figure 14 - 51 Procedure for Resuming Slave Transmission



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 14 - 52 Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

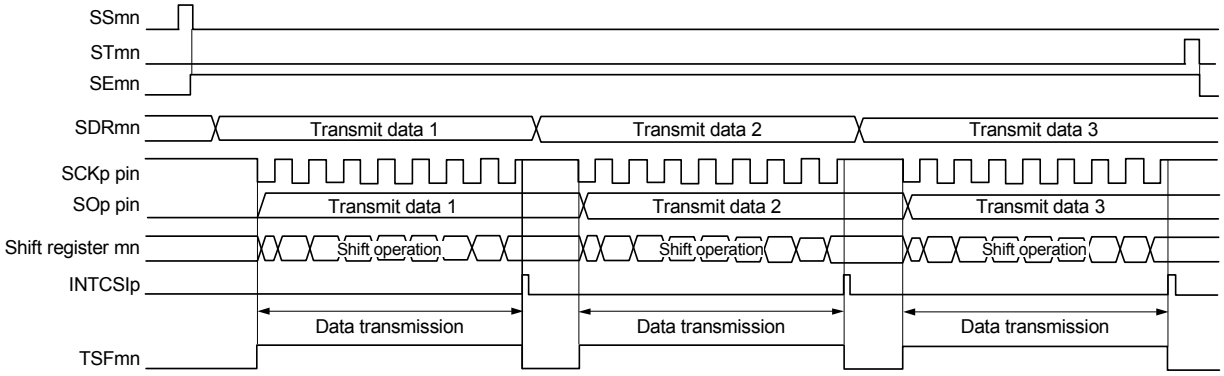
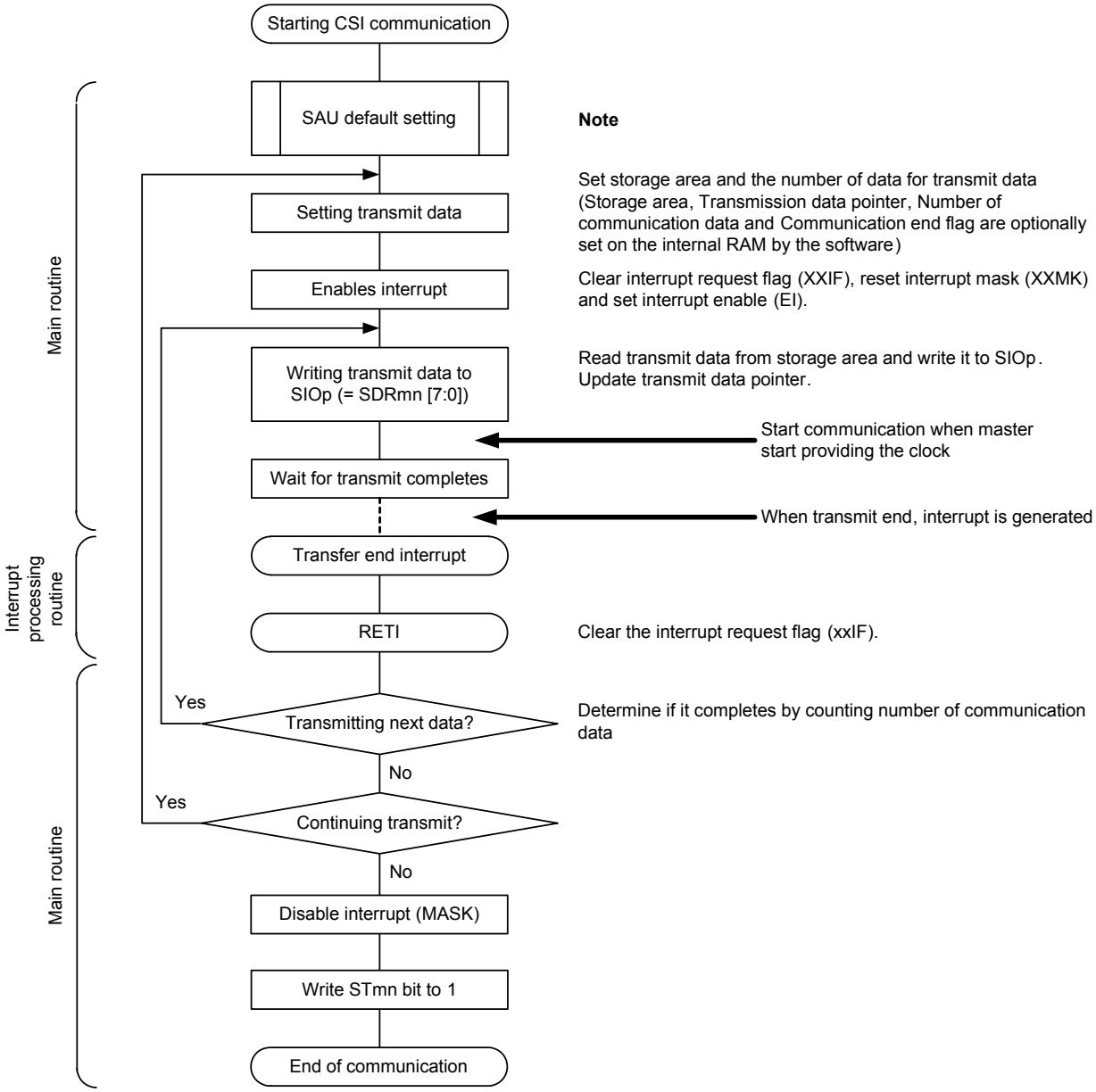


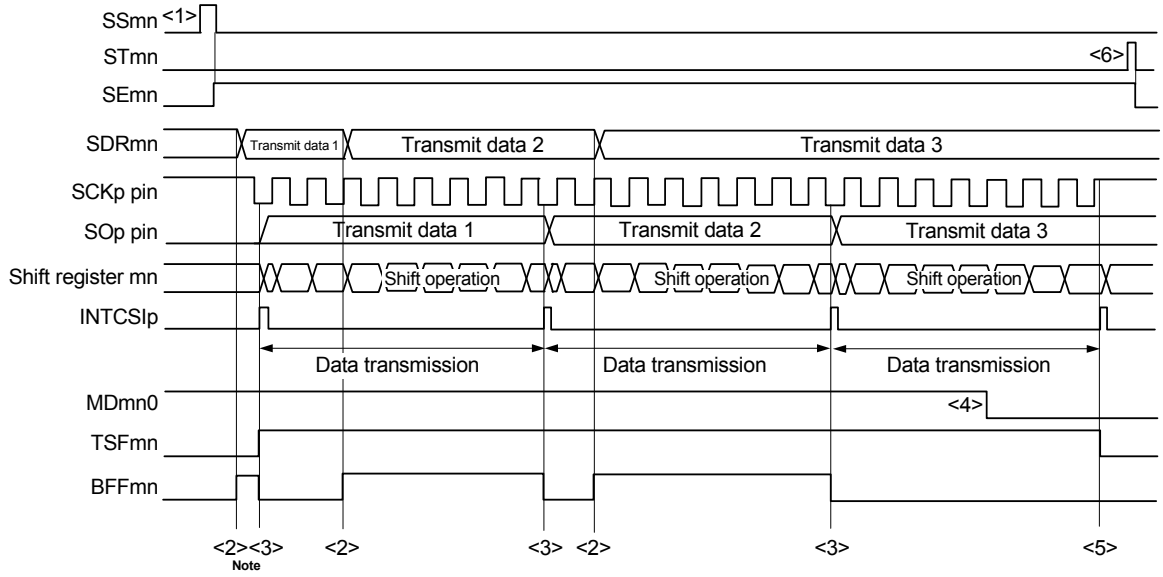
Figure 14 - 53 Flowchart of Slave Transmission (in Single-Transmission Mode)



Note For the initial setting, refer to **Figure 14 - 49**.(Select transfer end interrupt)

(4) Processing flow (in continuous transmission mode)

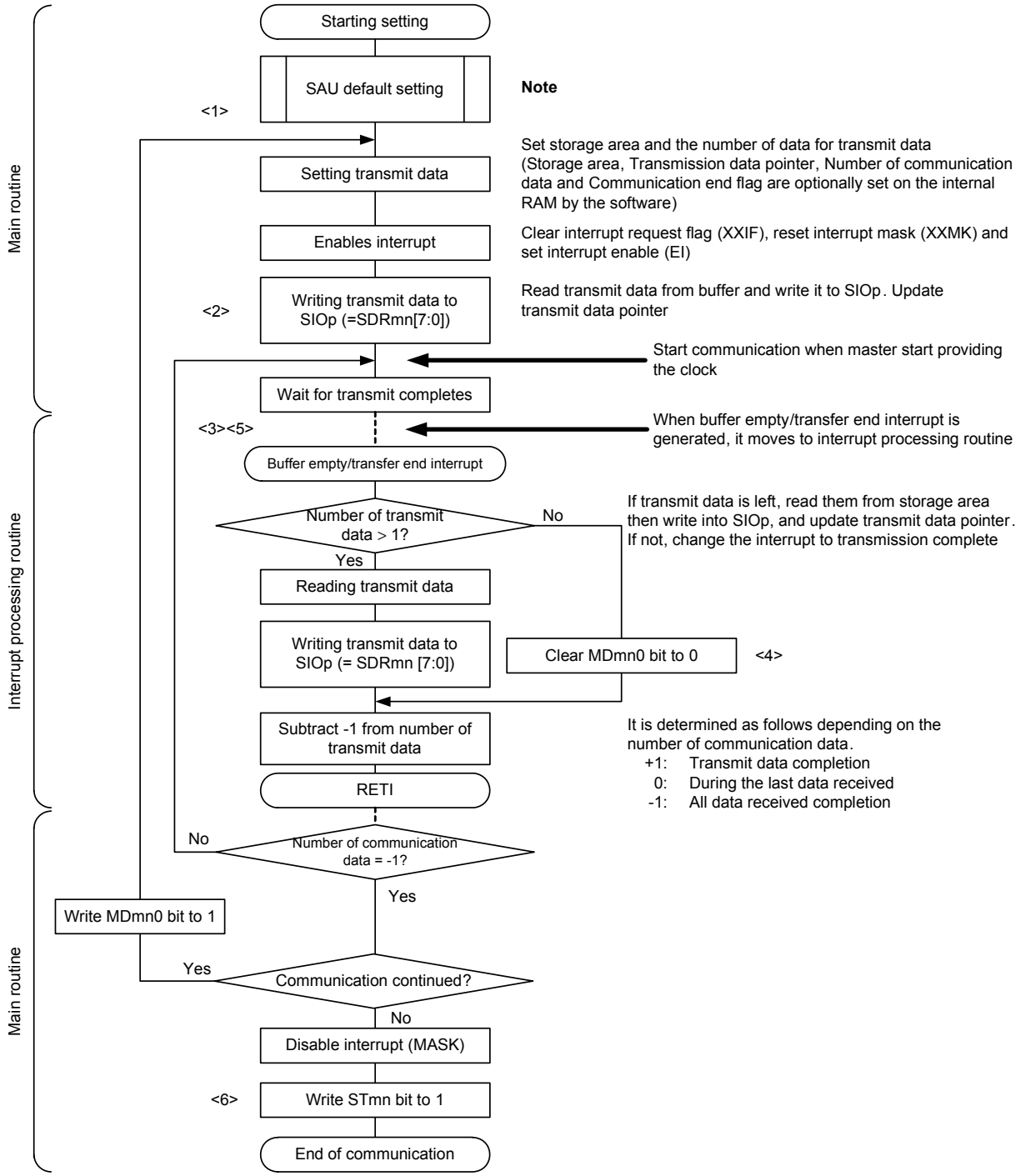
Figure 14 - 54 Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



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.

Figure 14 - 55 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Note For the initial setting, refer to **Figure 14 - 49**.(Select buffer empty interrupt)

Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 14 - 54 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

14.5.5 Slave reception

Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI10	CSI21	CSI30
Target channel	Channel 2 of SAU0	Channel 1 of SAU1	Channel 2 of SAU1
Pins used	SCK10, SI10	SCK21, SI21	SCK30, SI30
Interrupt	INTCSI10	INTCSI21	INTCSI30
	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 SCK10, SCK21, and SCK30 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 **CHAPTER 31 ELECTRICAL SPECIFICATIONS**).

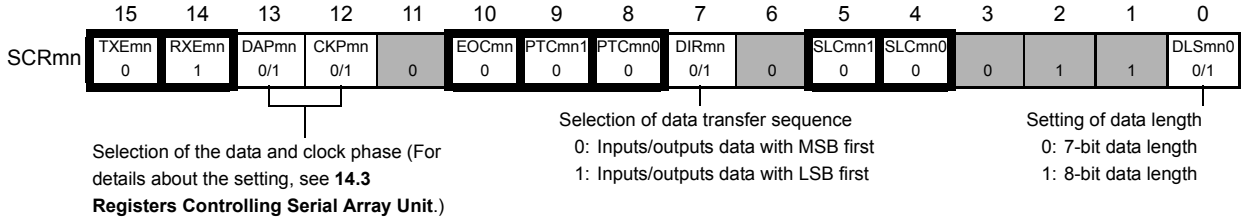
Remark f_{MCK}: Operation clock frequency of target channel

(1) Register setting

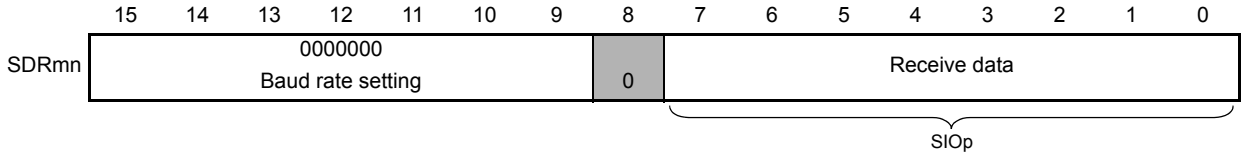
Figure 14 - 56 Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSIp)

(a) Serial mode register mn (SMRmn)

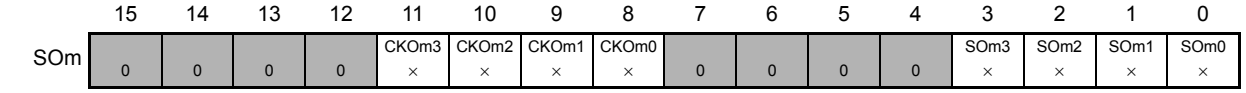
(b) Serial communication operation setting register mn (SCRmn)



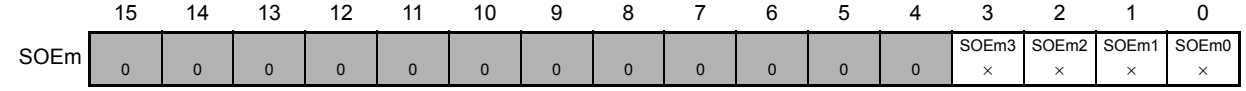
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



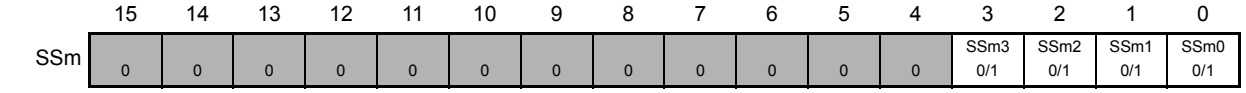
(d) Serial output register m (SOm)... The Register that not used in this mode.



(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 : 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 14 - 57 Initial Setting Procedure for Slave Reception

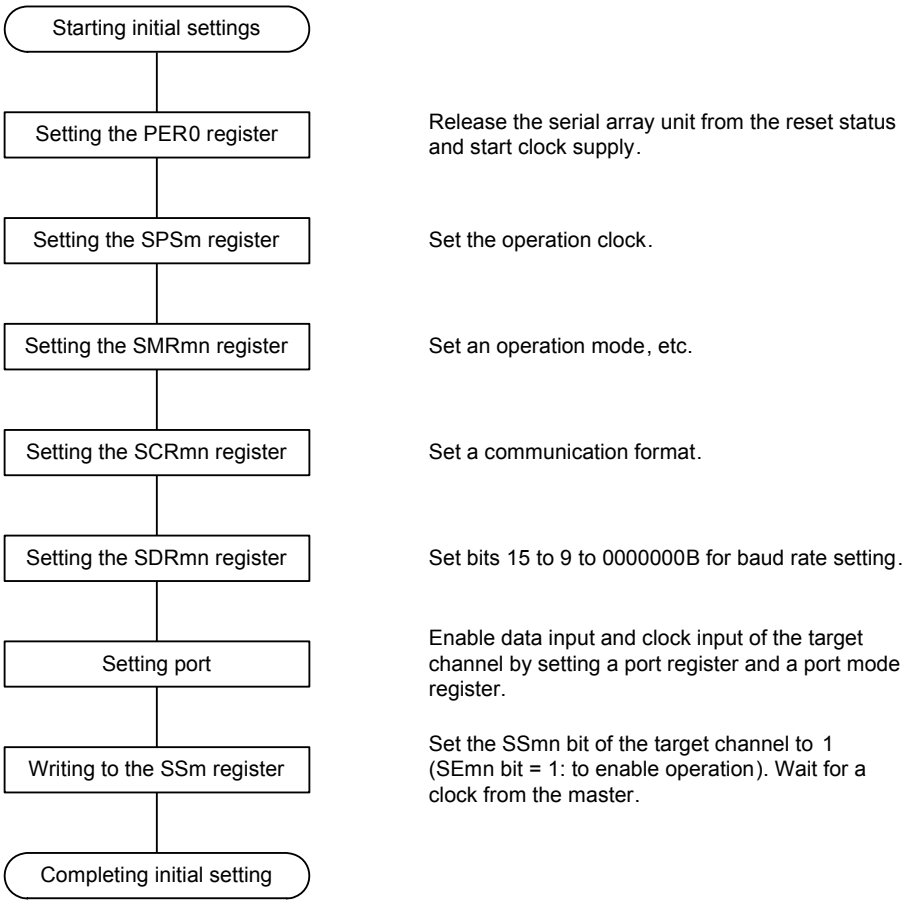


Figure 14 - 58 Procedure for Stopping Slave Reception

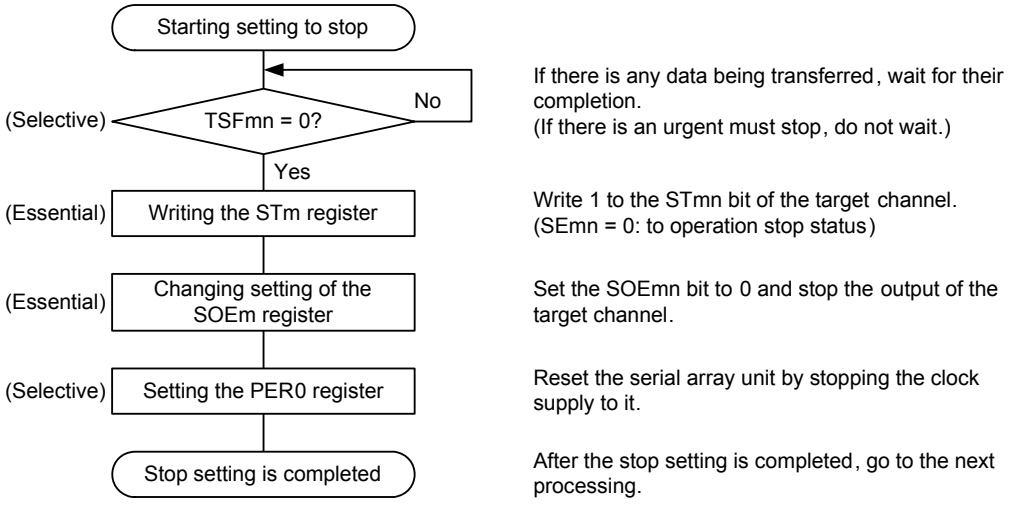
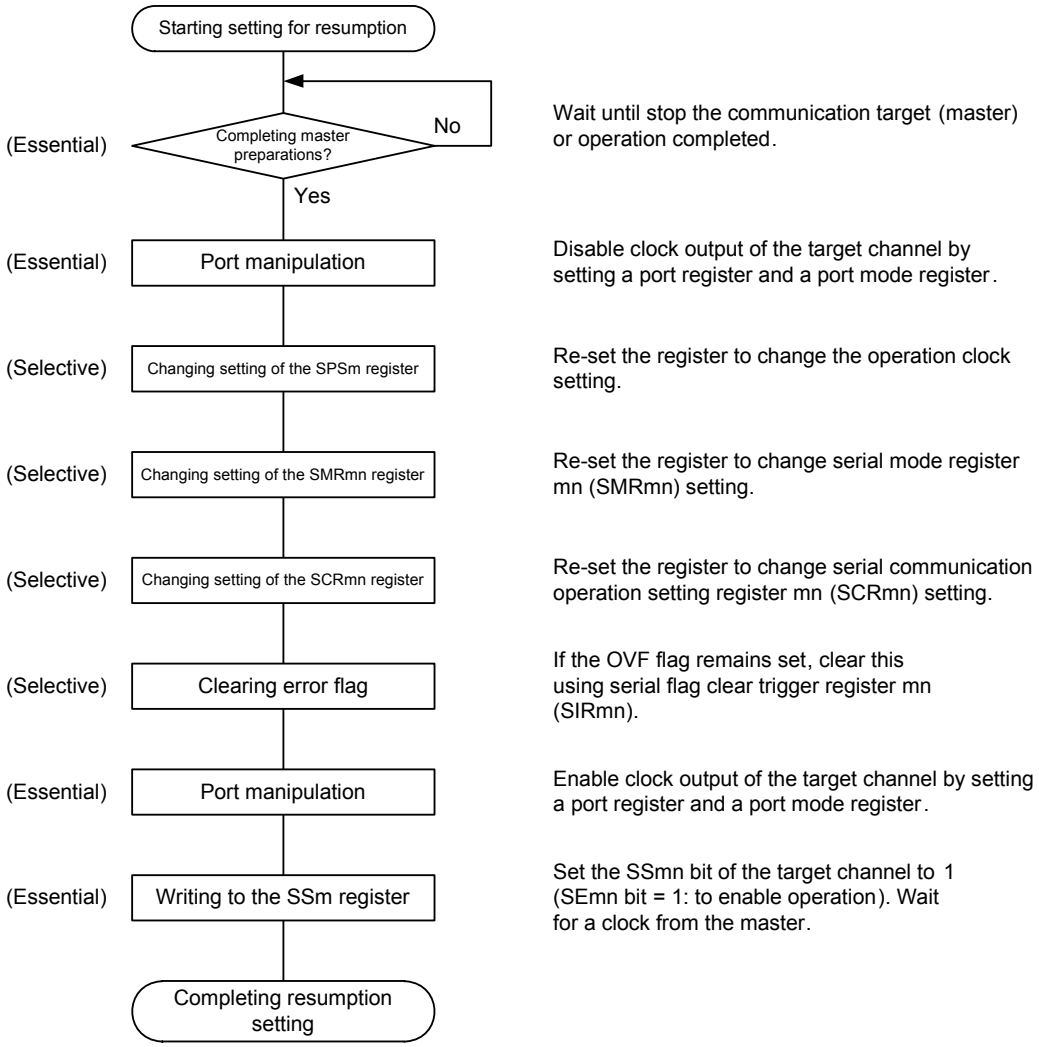


Figure 14 - 59 Procedure for Resuming Slave Reception



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 14 - 60 Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

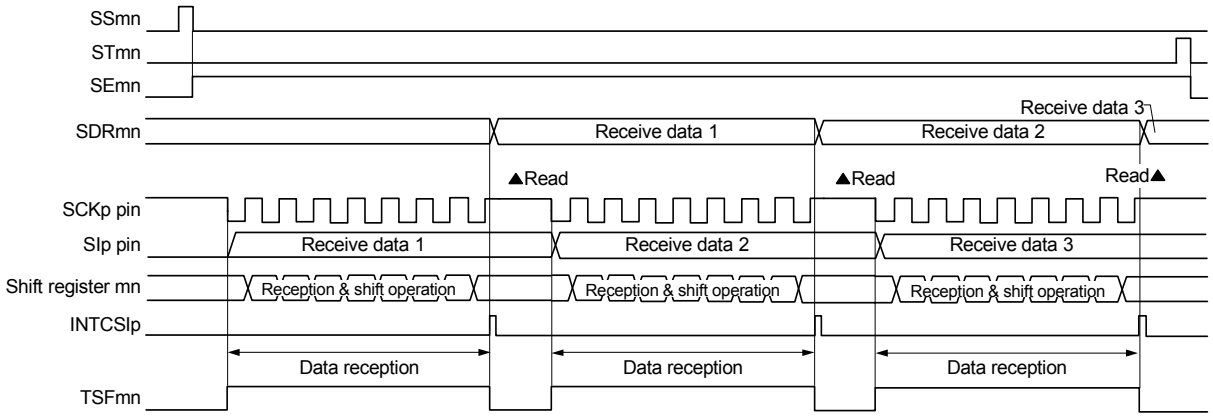
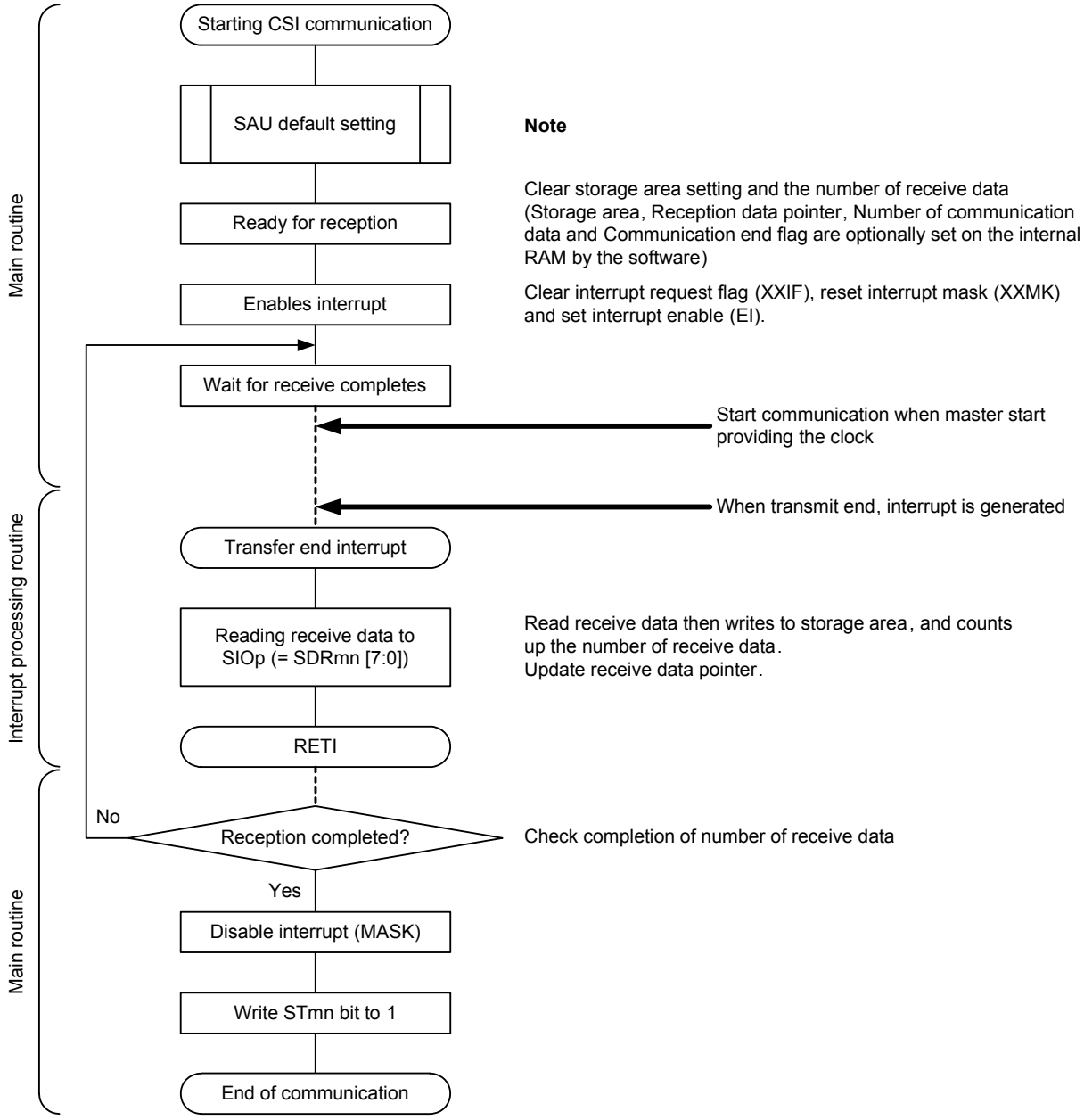


Figure 14 - 61 Flowchart of Slave Reception (in Single-Reception Mode)



Note For the initial setting, refer to **Figure 14 - 57**.(Select transfer end interrupt)

14.5.6 Slave transmission/reception

Slave transmission/reception is that the RL78 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	CSI10	CSI21	CSI30
Target channel	Channel 2 of SAU0	Channel 1 of SAU1	Channel 2 of SAU1
Pins used	SCK10, SI10, SO10	SCK21, SI21, SO21	SCK30, SI30, SO30
Interrupt	INTCSI10	INTCSI21	INTCSI30
	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 SCK10, SCK21, and SCK30 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

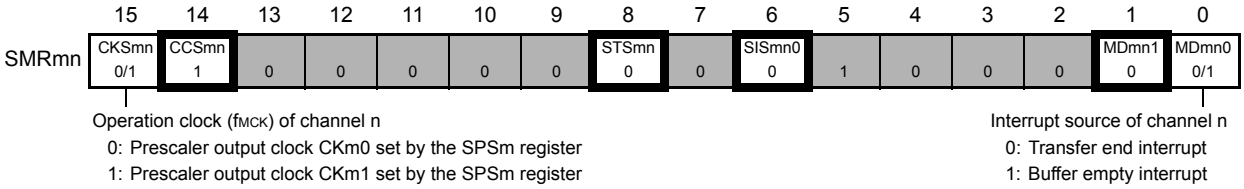
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel

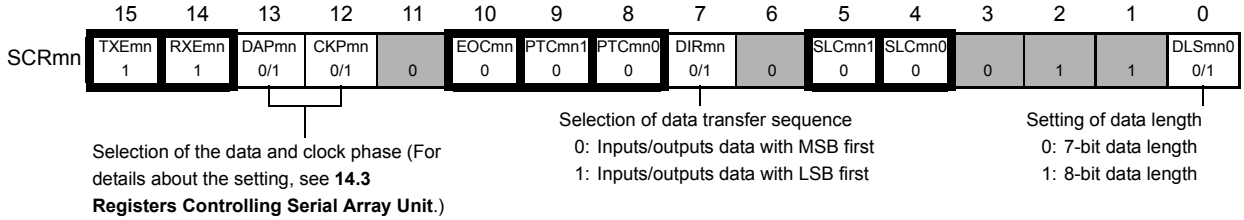
(1) Register setting

Figure 14 - 62 Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSIp)

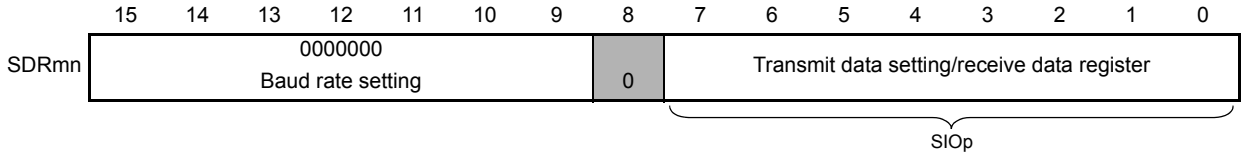
(a) Serial mode register mn (SMRmn)



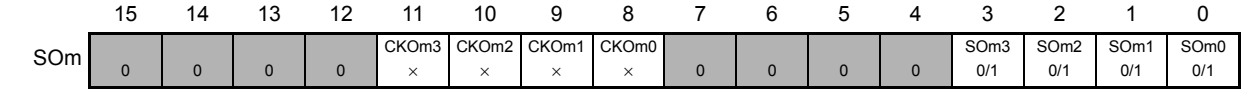
(b) Serial communication operation setting register mn (SCRmn)



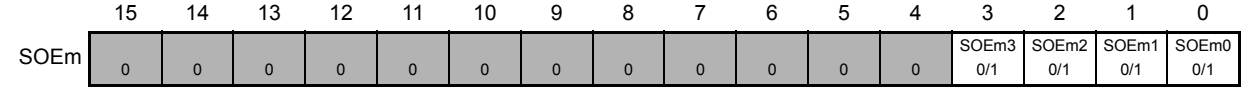
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



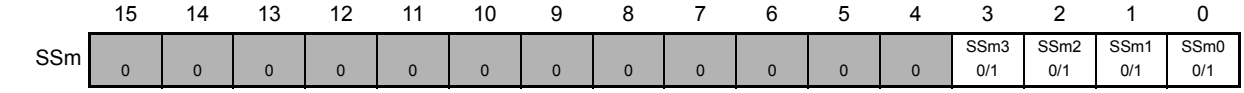
(d) Serial output register m (SOm)... Sets only the bits of the target channel.



(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 : Setting is fixed in the CSI master transmission/reception mode

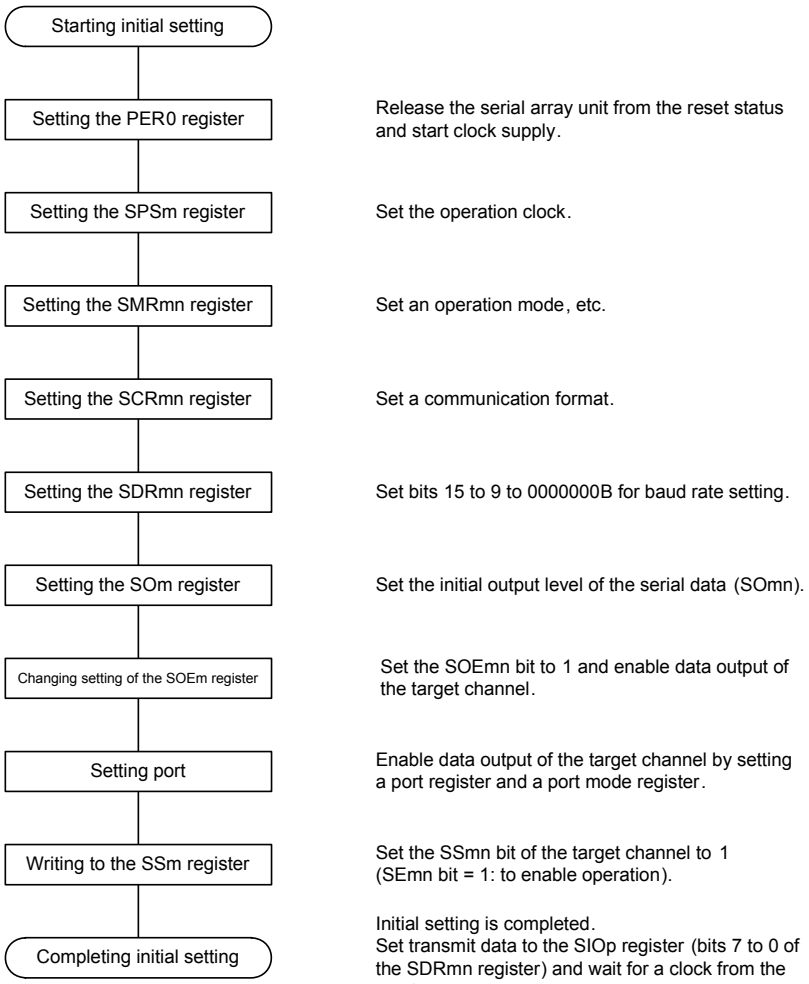
: Setting disabled (set to the initial value)

×: 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 14 - 63 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 14 - 64 Procedure for Stopping Slave Transmission/Reception

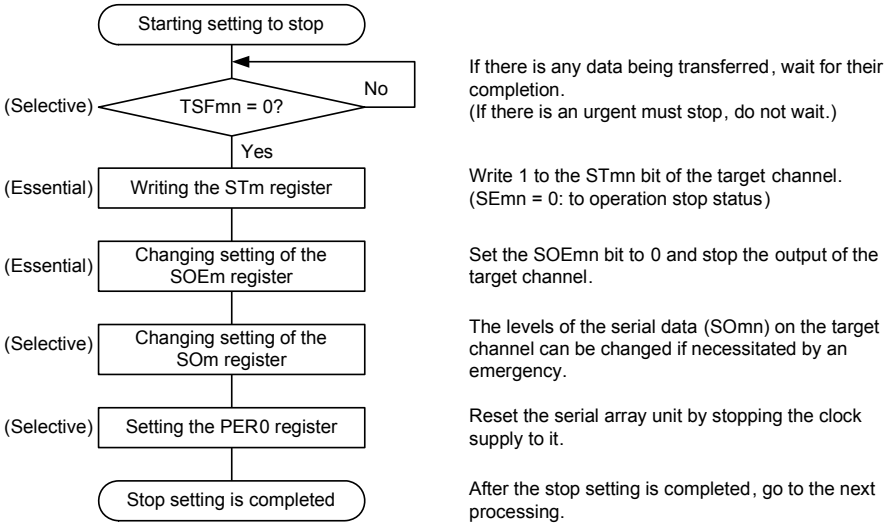
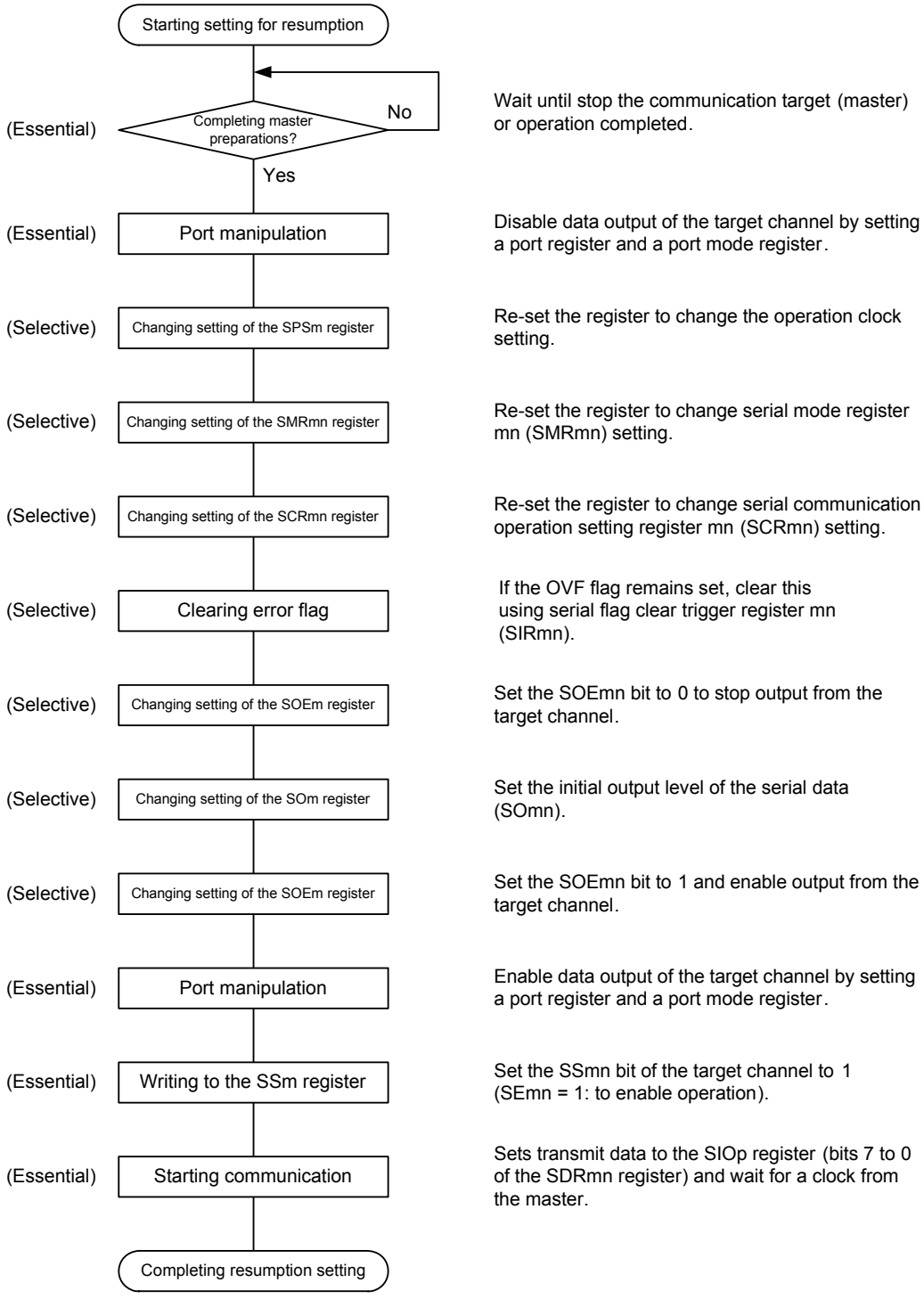


Figure 14 - 65 Procedure for Resuming Slave Transmission/Reception



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 14 - 66 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

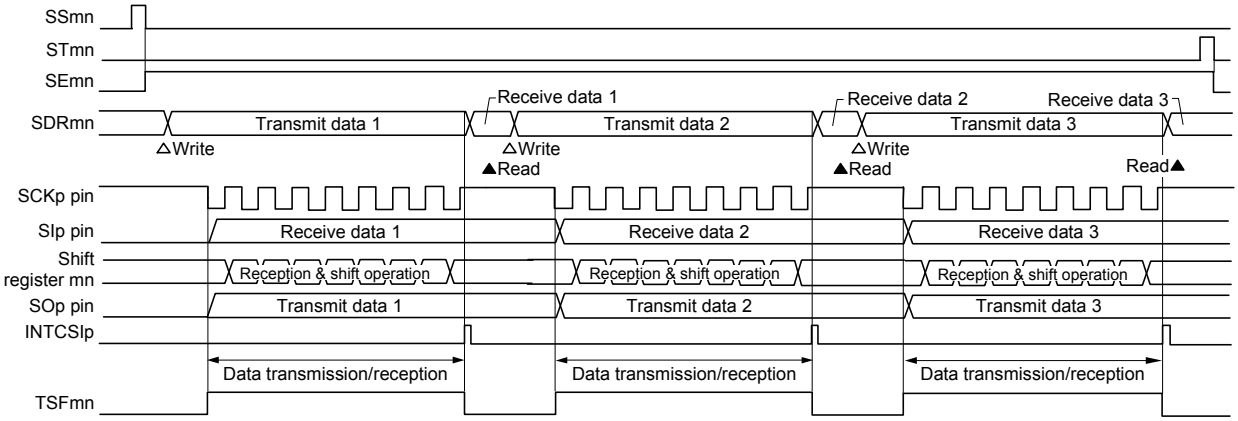
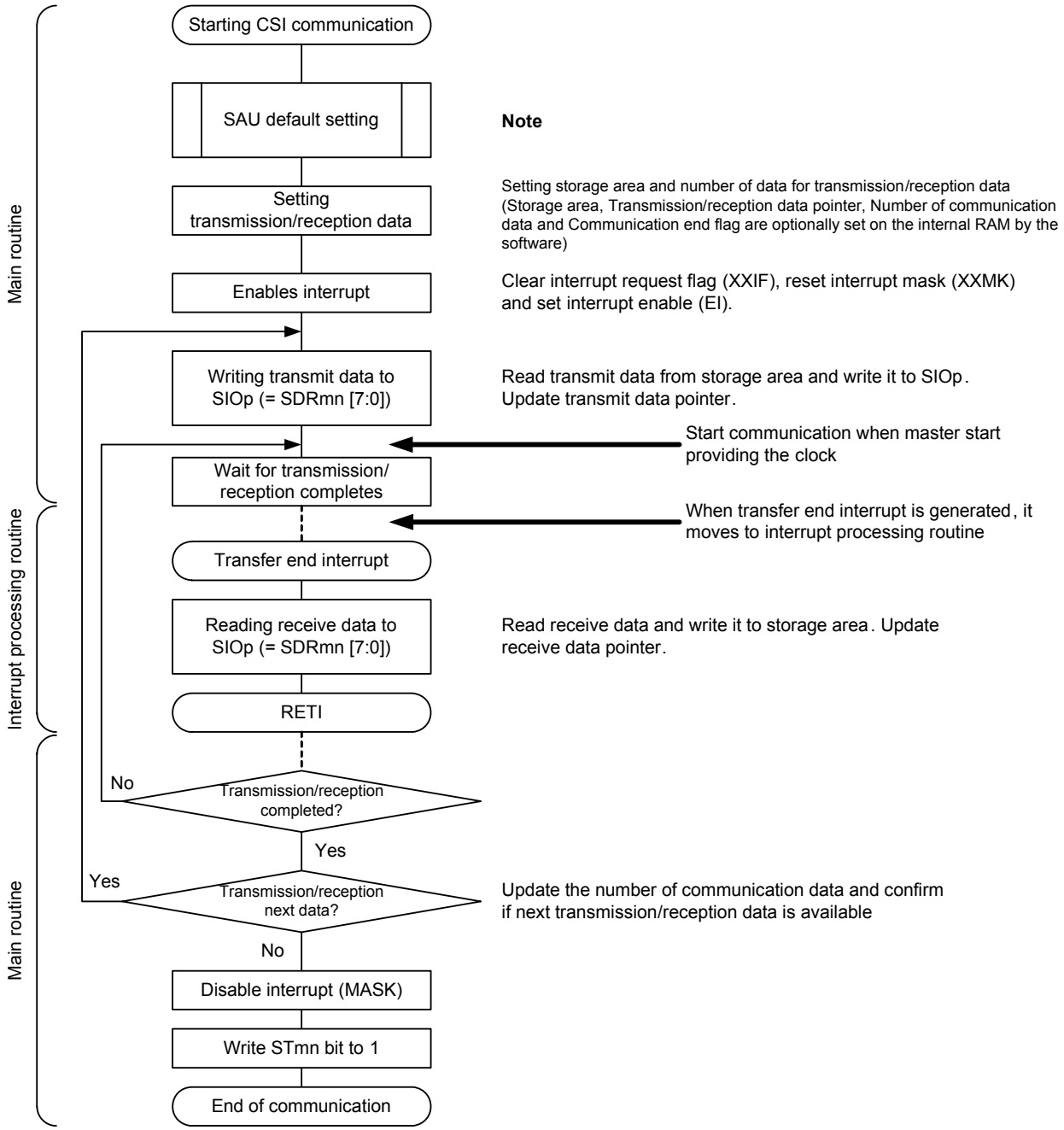


Figure 14 - 67 Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

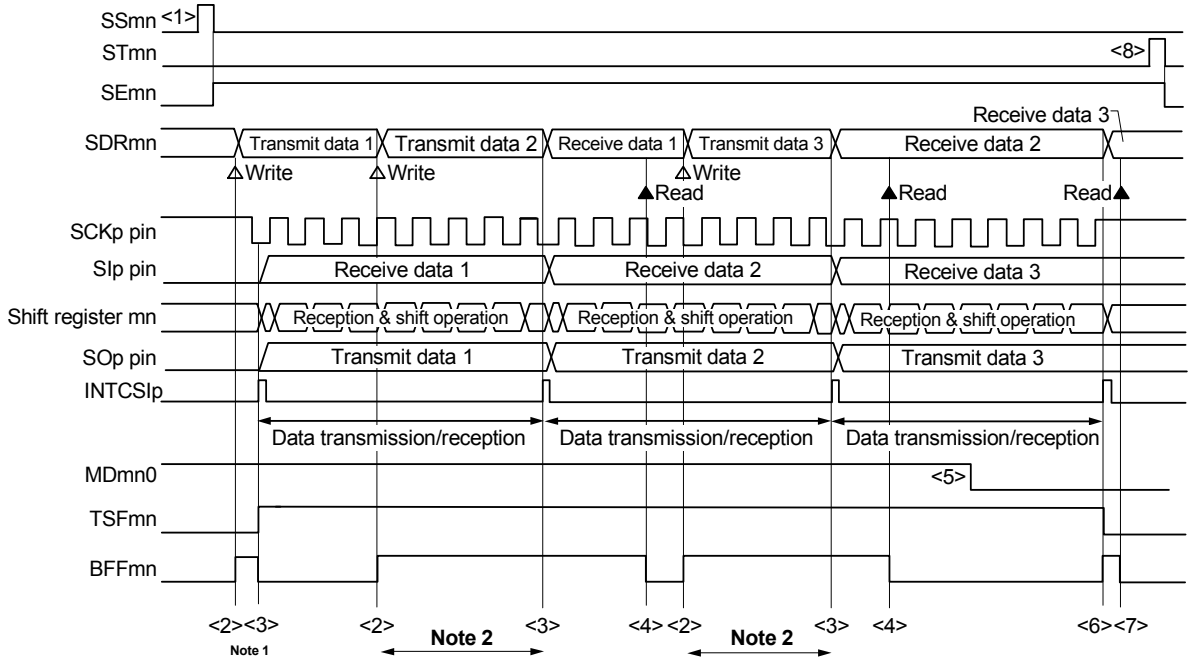


Note For the initial setting, refer to **Figure 14 - 63**.(Select transfer end interrupt)

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 14 - 68 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



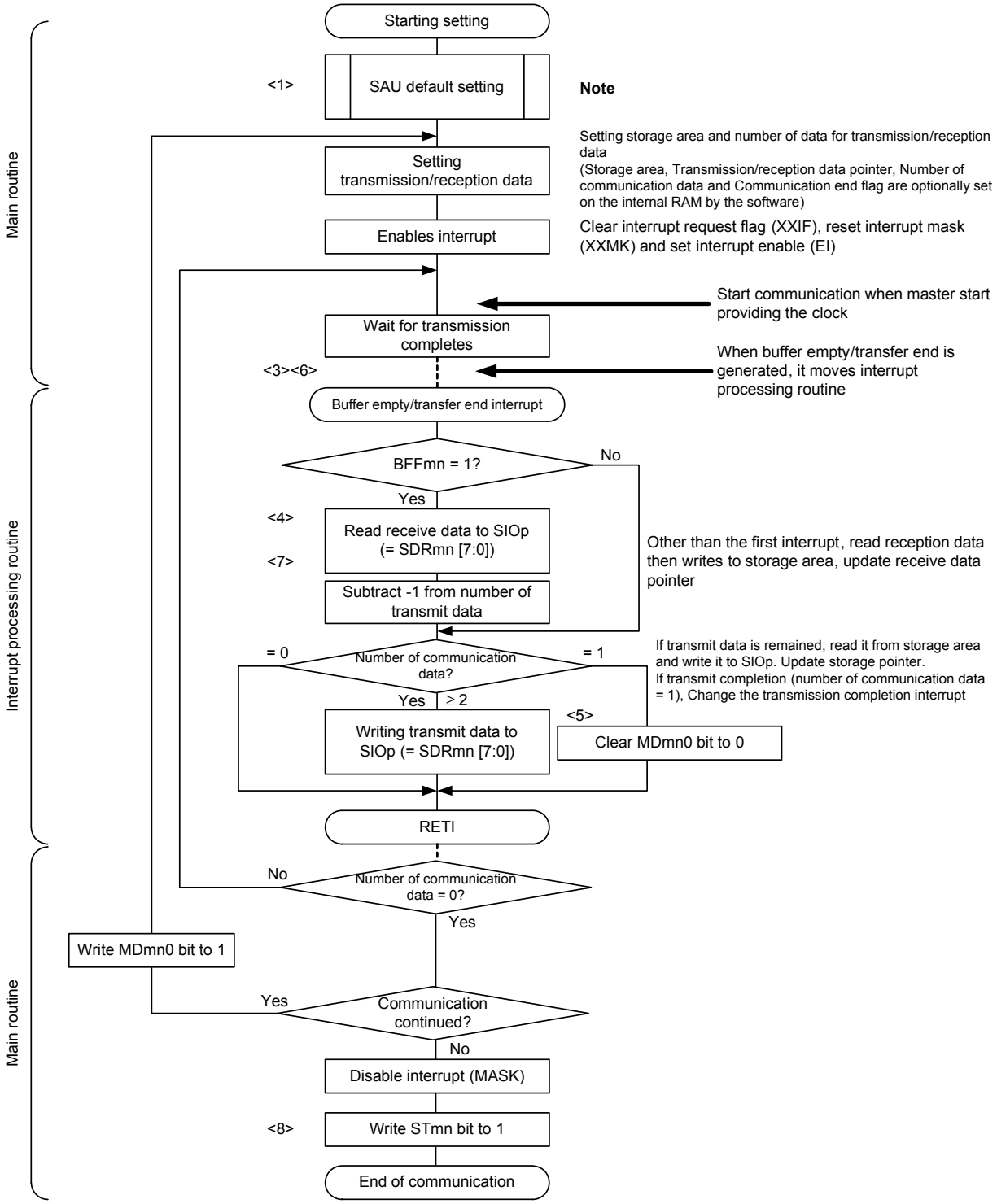
Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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> to <8> in the figure correspond to <1> to <8> in Figure 14 - 69 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Figure 14 - 69 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Note For the initial setting, refer to **Figure 14 - 63**.(Select buffer empty interrupt)

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 14 - 68 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

14.5.7 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSIp) 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 14 - 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} = 32 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	7.81 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	3.91 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.95 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	977 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 ×: Don't care

14.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSIp) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSIp) communication is described in Figure 14 - 70.

Figure 14 - 70 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.

14.6 Operation of UART (UARTq) 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, or 8 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

Unit	Channel	Used as CSI	Used as UART
0	0	Cannot be used	Cannot be used
	1		
	2	CSI10	UART1
	3	Cannot be used	
1	0	CSI20	Cannot be used
	1	CSI21	
	2	CSI30	UART3
	3	Cannot be used	

Select any function for each channel. Only the selected function is possible.

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 two types of communication operations.
- UART transmission (See **14.6.1**.)
- UART reception (See **14.6.2**.)

14.6.1 UART transmission

UART transmission is an operation to transmit data from the RL78 microcontroller to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART1	UART3
Target channel	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	TxD1	TxD3
Interrupt	INTST1	INTST3
	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	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] **Note**	
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	

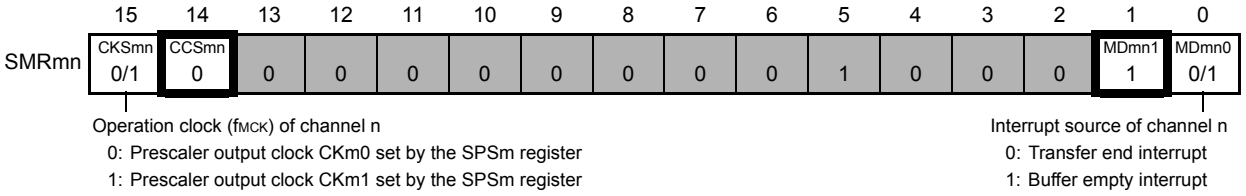
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

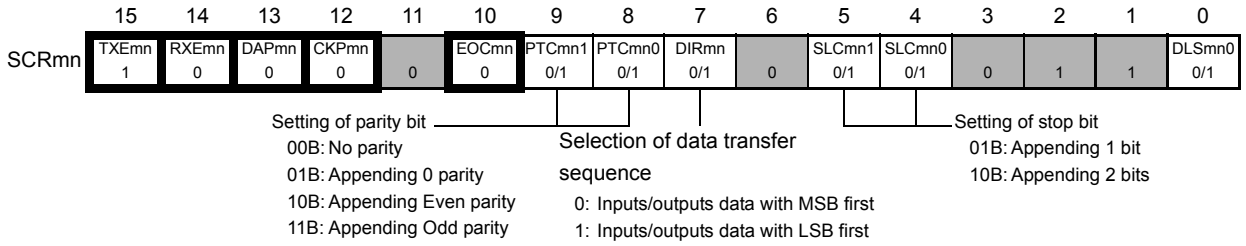
(1) Register setting

Figure 14 - 71 Example of Contents of Registers for UART Transmission of UART (UARTq) (1/2)

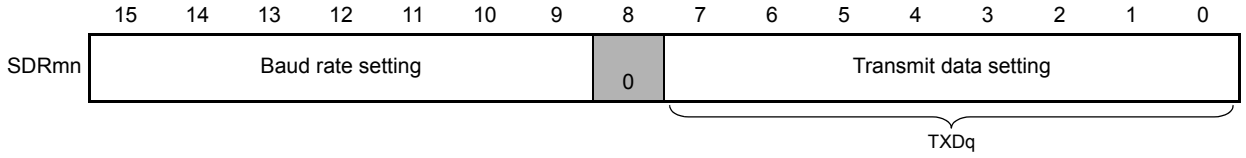
(a) Serial mode register mn (SMRmn)



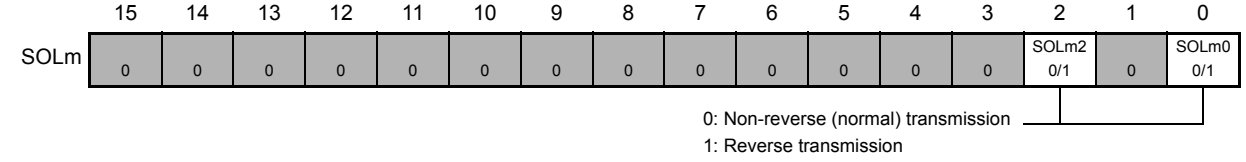
(b) Serial communication operation setting register mn (SCRmn)



(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.



Remark 1. m: Unit number (m = 0, 1) n: Channel number(n = 2) mn = 02, 12 q:UART number(q = 1, 3)

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 14 - 72 Example of Contents of Registers for UART Transmission of UART (UARTq) (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	CKOm3 ×	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm3 ×	SOm2 0/1 Note	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	SOEm3 ×	SOEm2 0/1	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	SSm3 ×	SSm2 0/1	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, 1), n: Channel number (n = 2), mn = 02, 12, q: UART number (q = 1, 3)

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 14 - 73 Initial Setting Procedure for UART Transmission

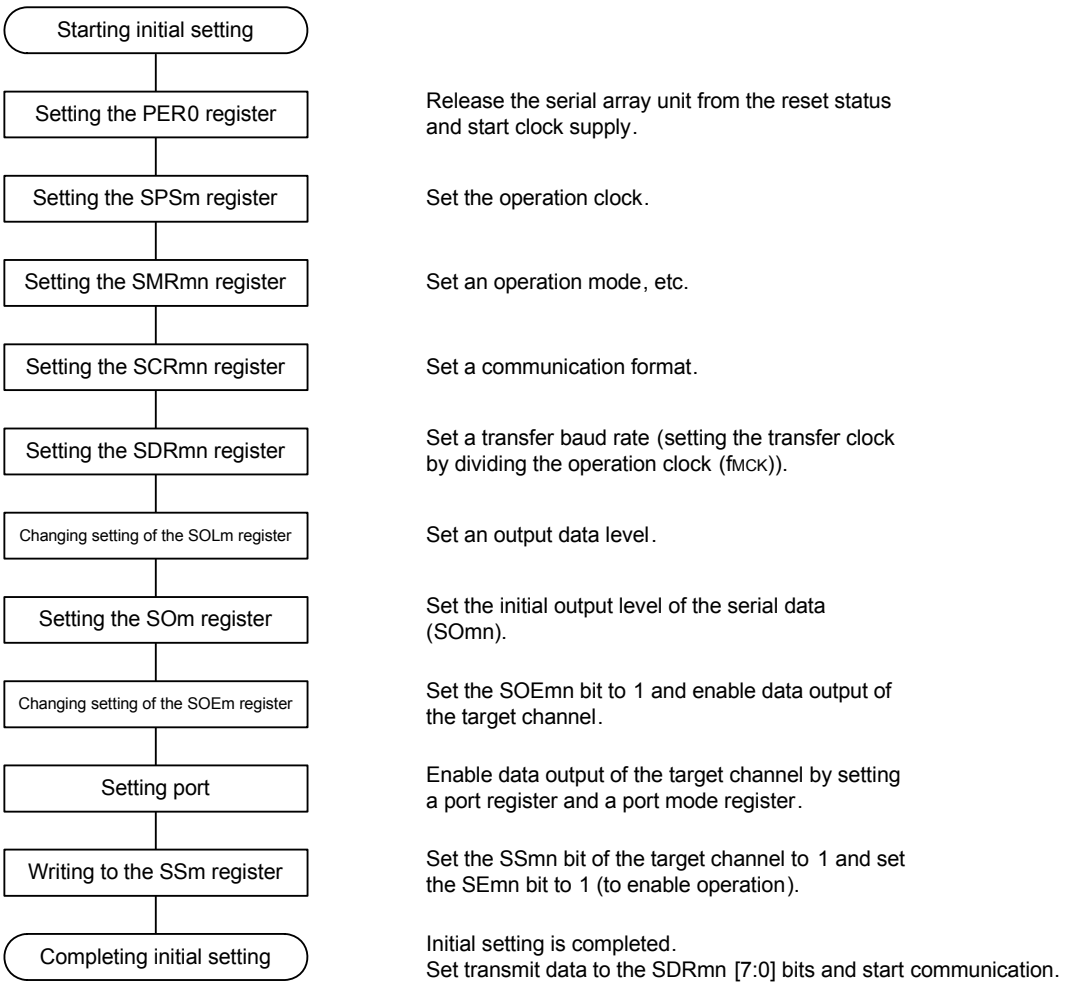


Figure 14 - 74 Procedure for Stopping UART Transmission

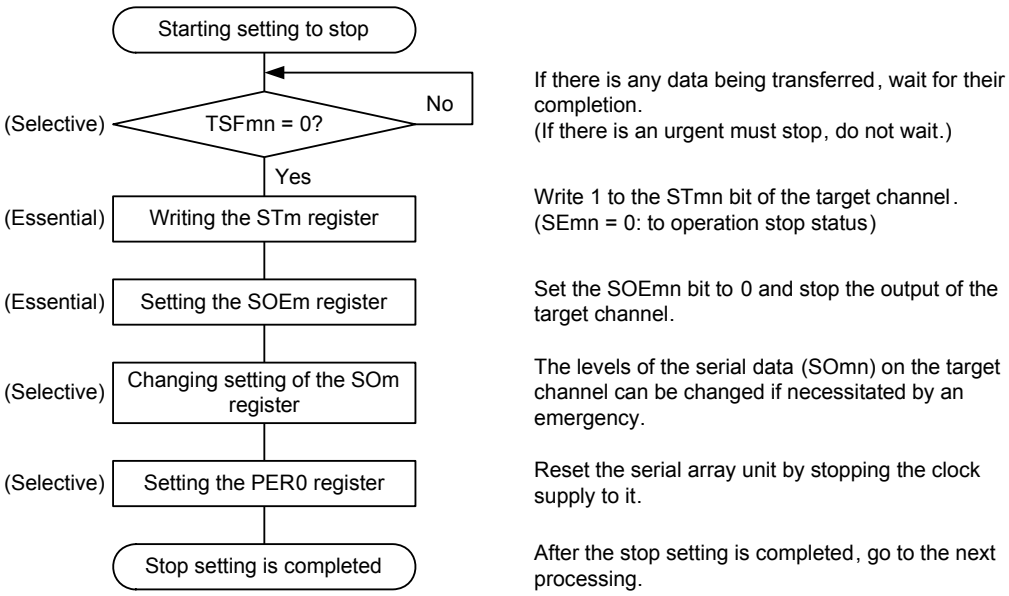
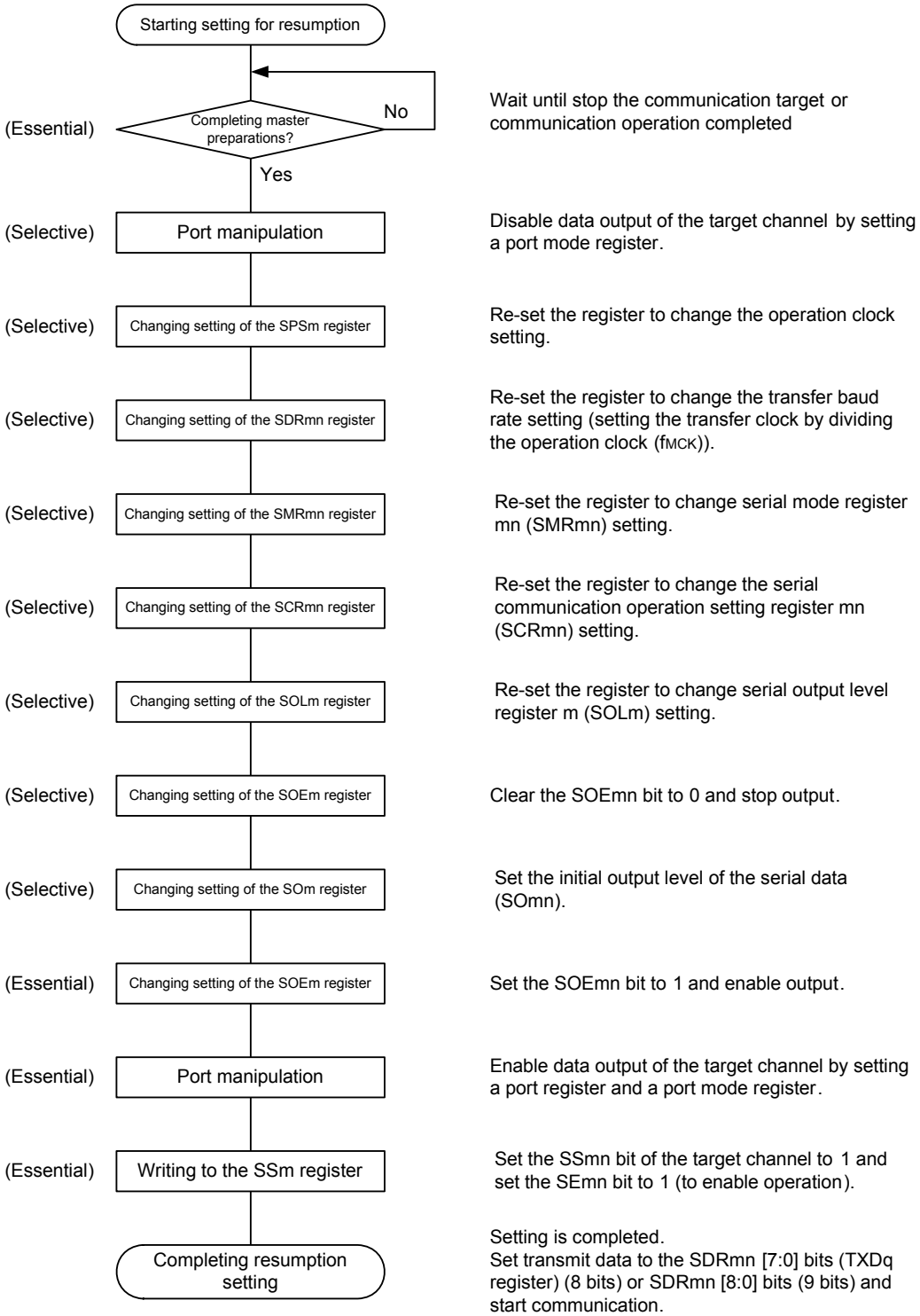


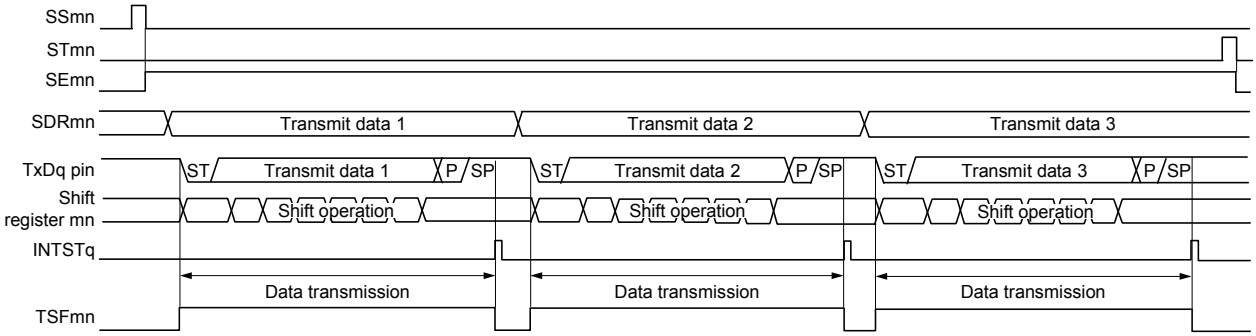
Figure 14 - 75 Procedure for Resuming UART Transmission



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.

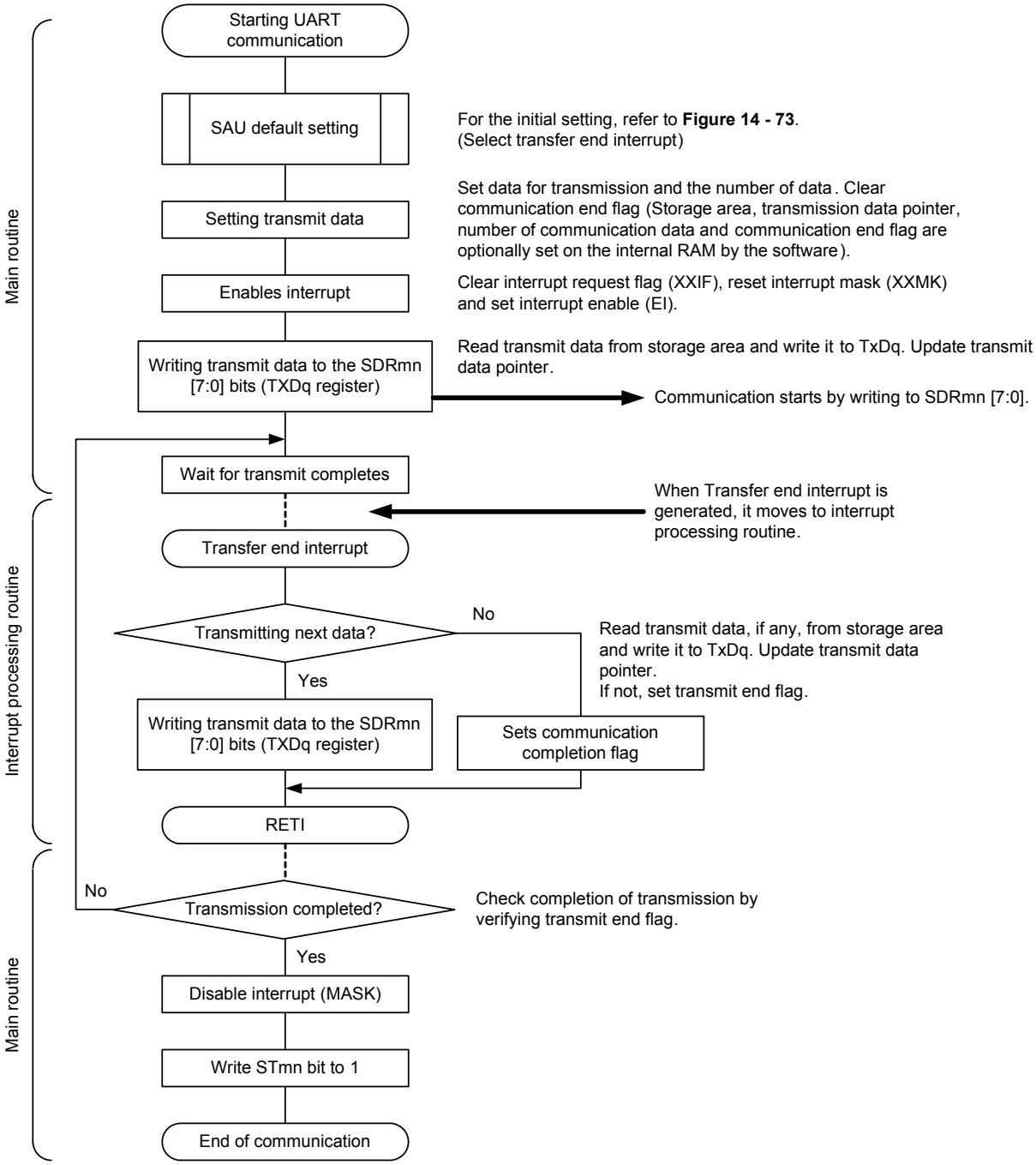
(3) Processing flow (in single-transmission mode)

Figure 14 - 76 Timing Chart of UART Transmission (in Single-Transmission Mode)



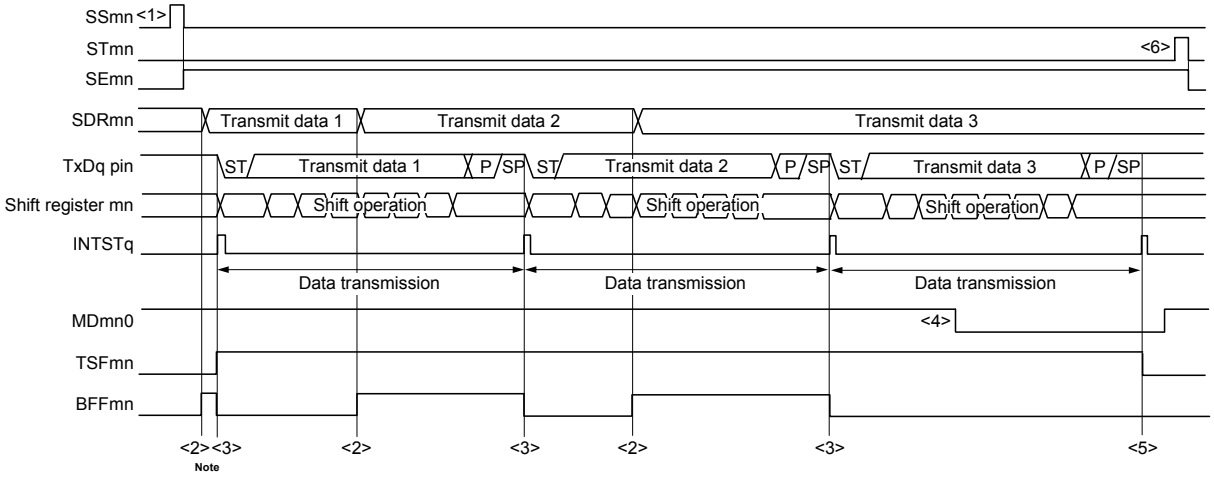
Remark m: Unit number (m = 0, 1), n: Channel number (n = 2), q: UART number (q = 1, 3), mn = 02, 12

Figure 14 - 77 Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 14 - 78 Timing Chart of UART Transmission (in Continuous Transmission Mode)

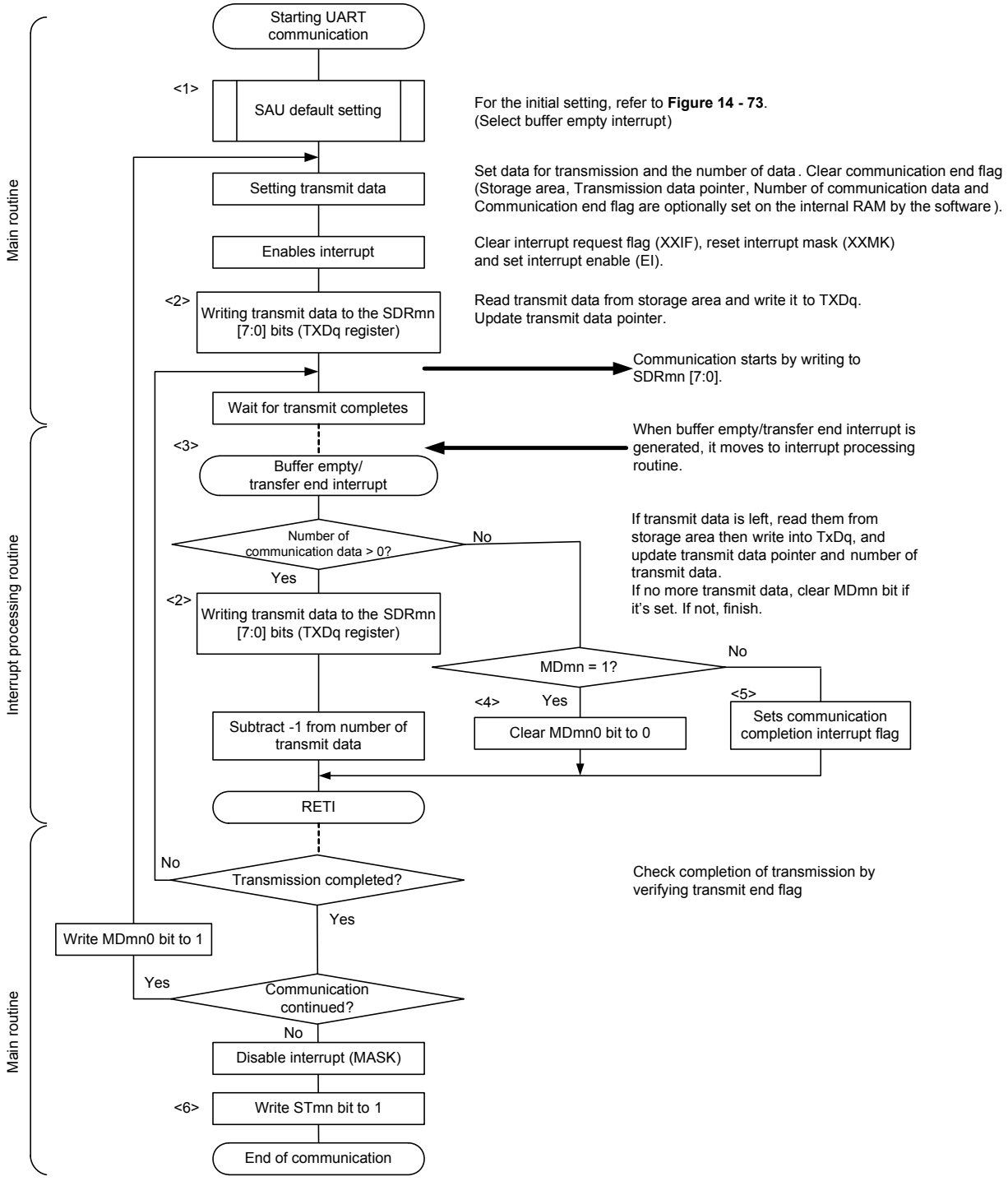


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 m: Unit number (m = 0, 1), n: Channel number (n = 2), q: UART number (q = 1, 3), mn = 02, 12

Figure 14 - 79 Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 14 - 78 Timing Chart of UART Transmission (in Continuous Transmission Mode).

14.6.2 UART reception

UART reception is an operation wherein the RL78 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	UART1	UART3
Target channel	Channel 3 of SAU0	Channel 3 of SAU1
Pins used	RxD1	RxD3
Interrupt	INTST1	INTST3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE1	INTSRE3
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)	
Transfer data length	7, or 8 bits	
Transfer rate **Note**	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 Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS**).

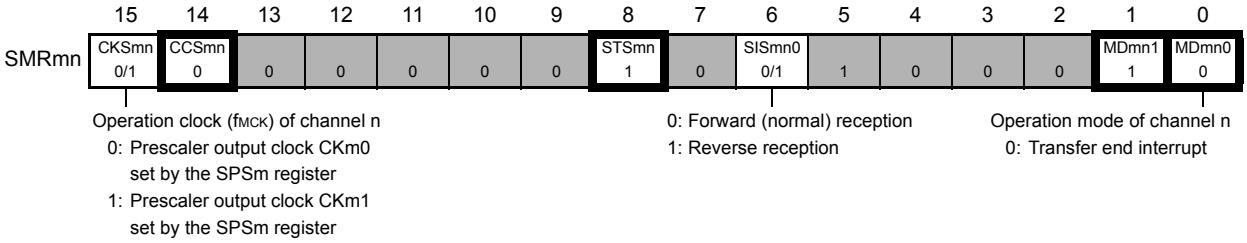
Remark 1. f_{MCK}:Operation clock frequency of target channel
f_{CLK}:System clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 3), mn = 03, 13

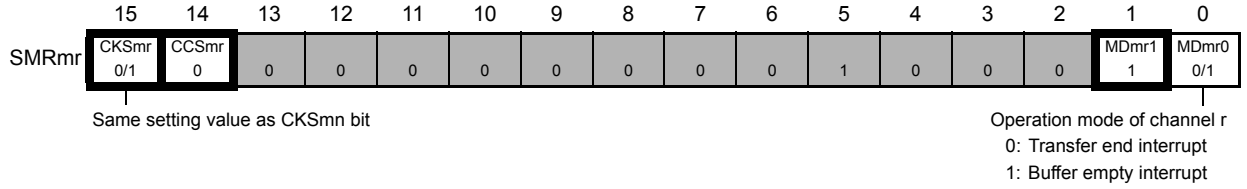
(1) Register setting

Figure 14 - 80 Example of Contents of Registers for UART Reception of UART (UARTq) (1/2)

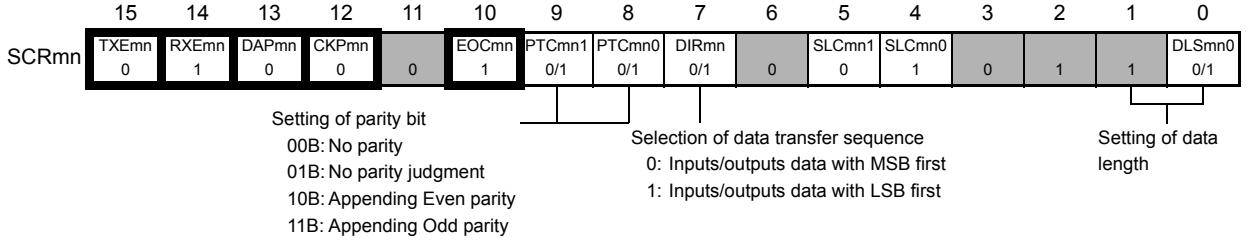
(a) Serial mode register mn (SMRmn)



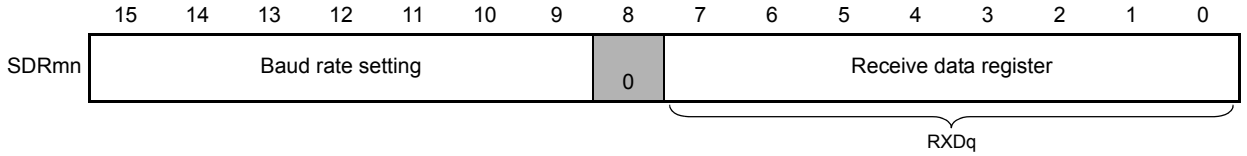
(b) Serial mode register mr (SMRmr)



(c) Serial communication operation setting register mn (SCRmn)



(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



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. q: UART number (q = 1, 3), r: Channel number (r = n – 1)

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 14 - 81 Example of Contents of Registers for UART Reception of UART (UARTq) (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	CKOm3 ×	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm3 ×	SOm2 ×	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	SOEm3 ×	SOEm2 ×	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	SSm3 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

Remark 1. m: Unit number (m = 0, 1), q: UART number (q = 1, 3)

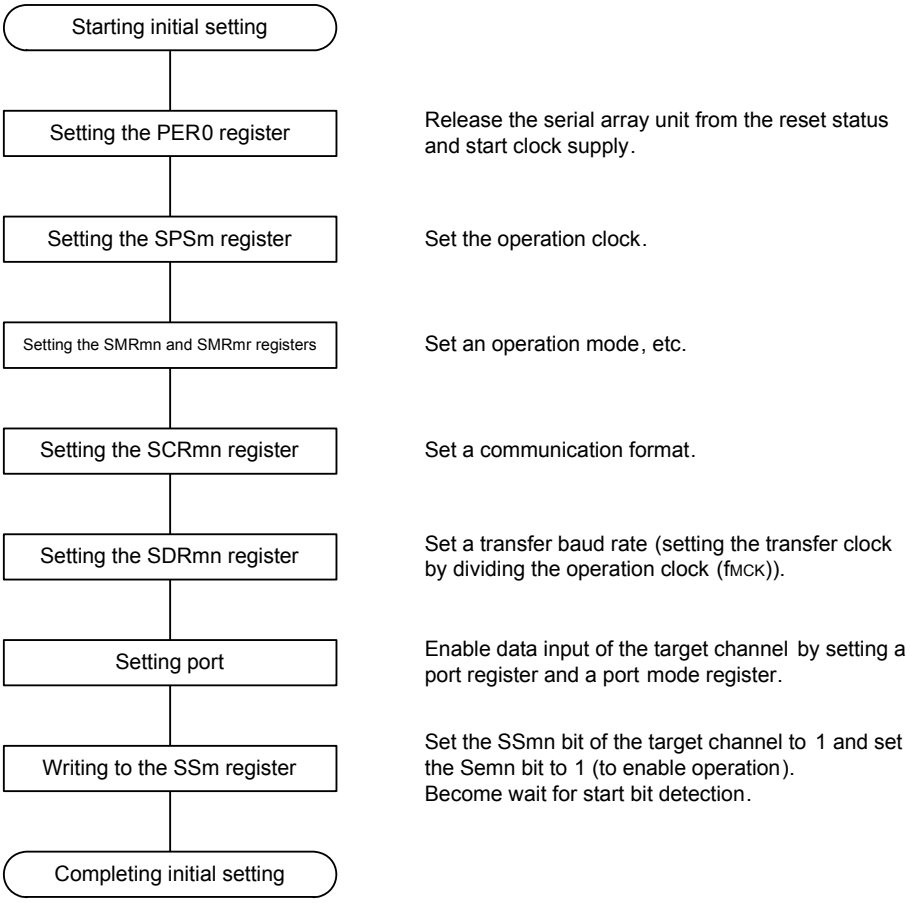
Remark 2. : Setting disabled (set to the initial value)

×: 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 14 - 82 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 14 - 83 Procedure for Stopping UART Reception

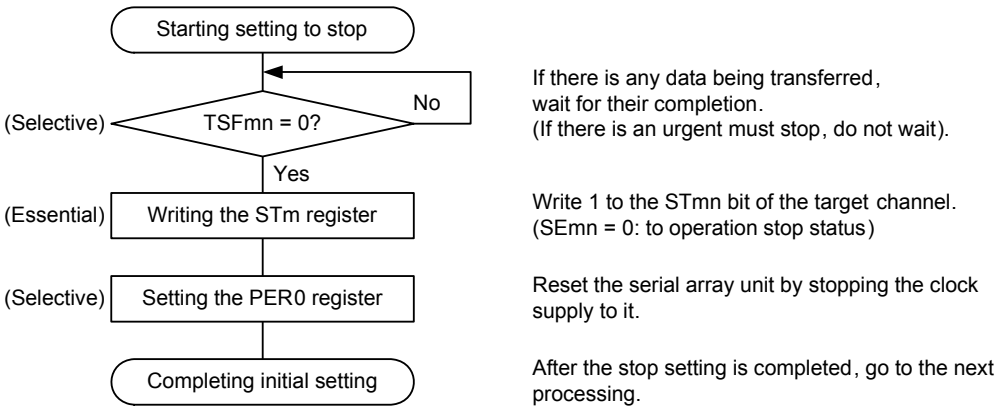
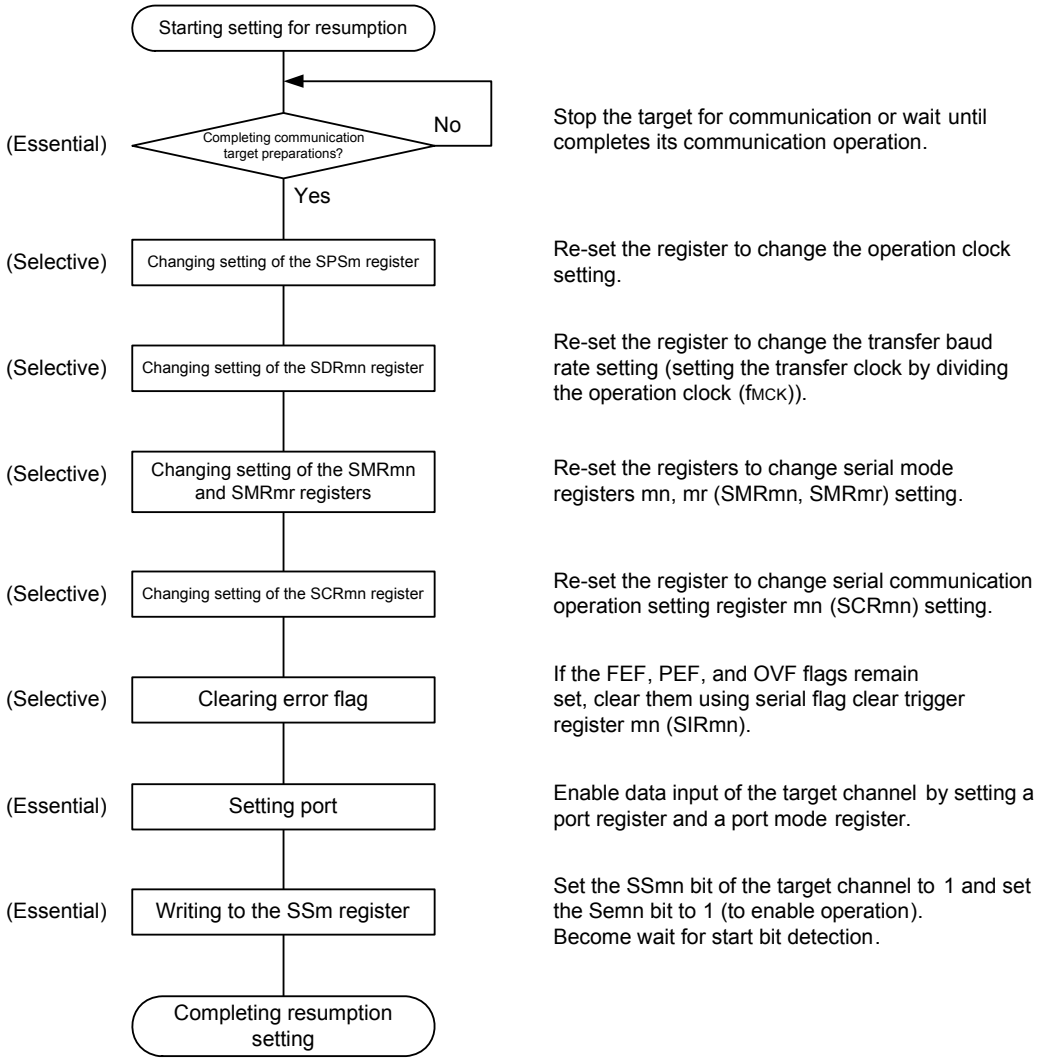


Figure 14 - 84 Procedure for Resuming UART Reception



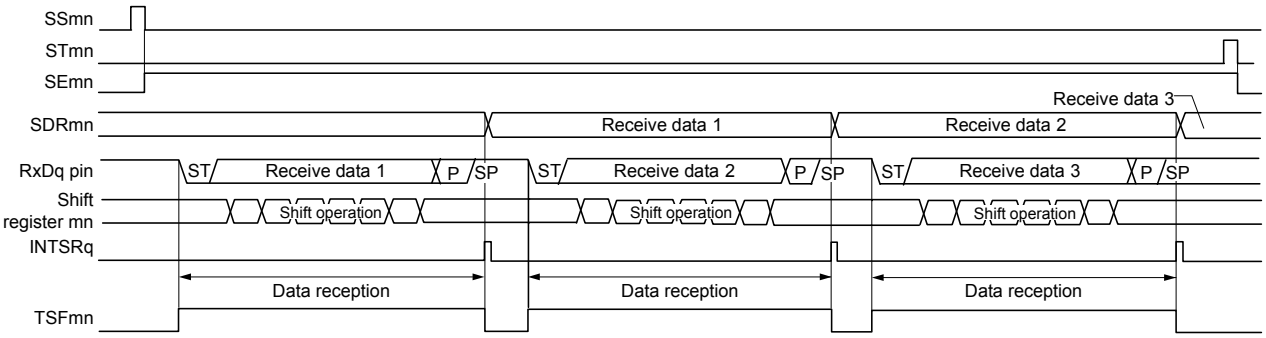
Caution After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of f_{MCK}.

Remark 1. 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.

Remark 2. r: Channel number (r = n – 1)

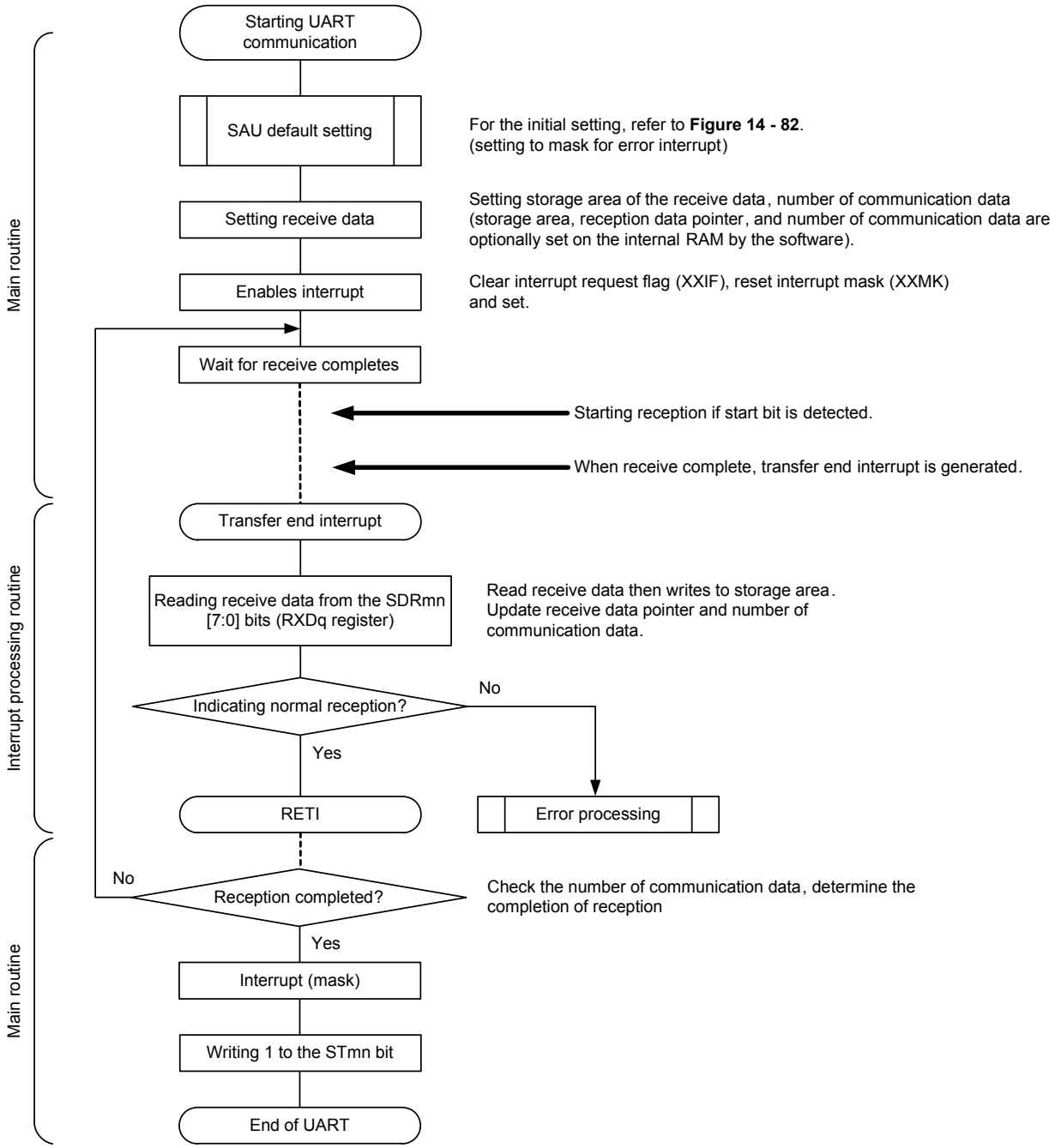
(3) Processing flow

Figure 14 - 85 Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 3), mn = 03, 13, q: UART number (q = 1, 3)

Figure 14 - 86 Flowchart of UART Reception



14.6.3 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UARTq) 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, 1), n: Channel number (n = 2, 3), mn = 02, 03, 12, 13, q: UART number (q = 1, 3)

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 14 - 3 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} = 32 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	7.81 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	3.91 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.95 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	977 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, 1), n: Channel number (n = 2, 3), mn = 02, 03, 12, 13, q: UART number (q = 1, 3)

(2) Baud rate error during transmission

The baud rate error of UART (UARTq) 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} = 32 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 32 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	103	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	103	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	103	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	103	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	103	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	103	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	103	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	63	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	103	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	103	76923.1 bps	+0.16%
153600 bps	f_{CLK}	103	153846 bps	+0.16%
312500 bps	f_{CLK}	50	312500 bps	±0.39%

Remark m: Unit number (m = 0, 1), n: Channel number (n = 2), mn = 02, 12, q: UART number (q = 1, 3)

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UARTq) 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.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

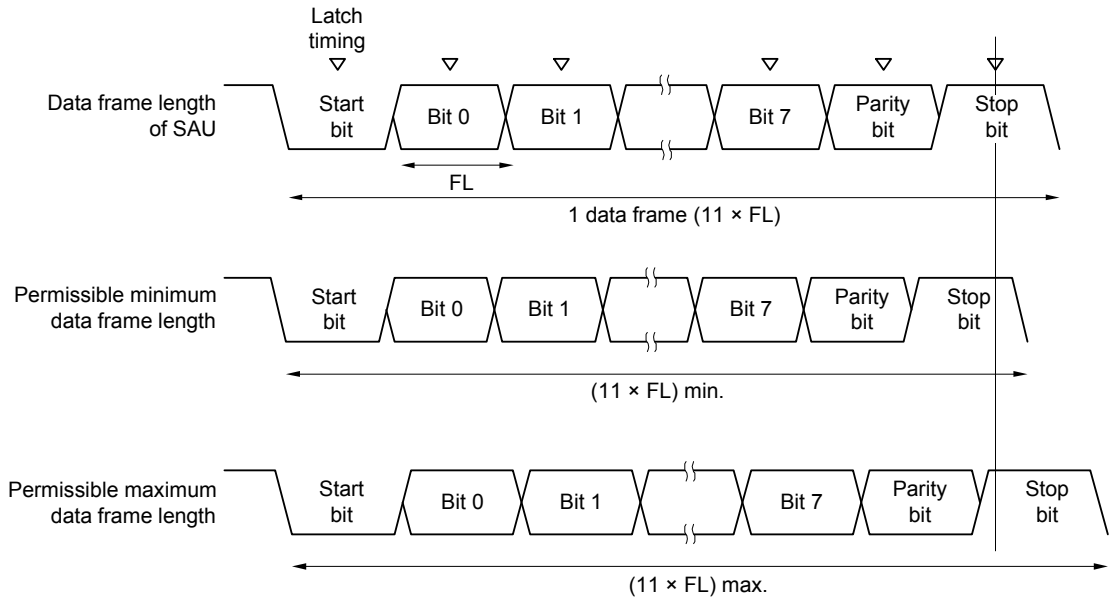
Brate: Calculated baud rate value at the reception side (See **14.6.3 (1) Baud rate calculation expression**.)

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 = 3), mn = 03, 13, q: UART number (q = 1, 3)

Figure 14 - 87 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 14 - 87, 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.

14.6.4 Procedure for processing errors that occurred during UART (UARTq) communication

The procedure for processing errors that occurred during UART (UARTq) communication is described in Figures 14 - 88 and 14 - 89.

Figure 14 - 88 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 14 - 89 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, 1), n: Channel number (n = 2, 3), mn = 02, 03, 12, 13, q: UART number (q = 1, 3)

CHAPTER 15 SERIAL INTERFACE IICA

15.1 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 15 - 1 shows a block diagram of serial interface IICA (channel 0)

Remark n = 0, 1

Figure 15 - 1 Block Diagram of Serial Interface IICA (channel 0)

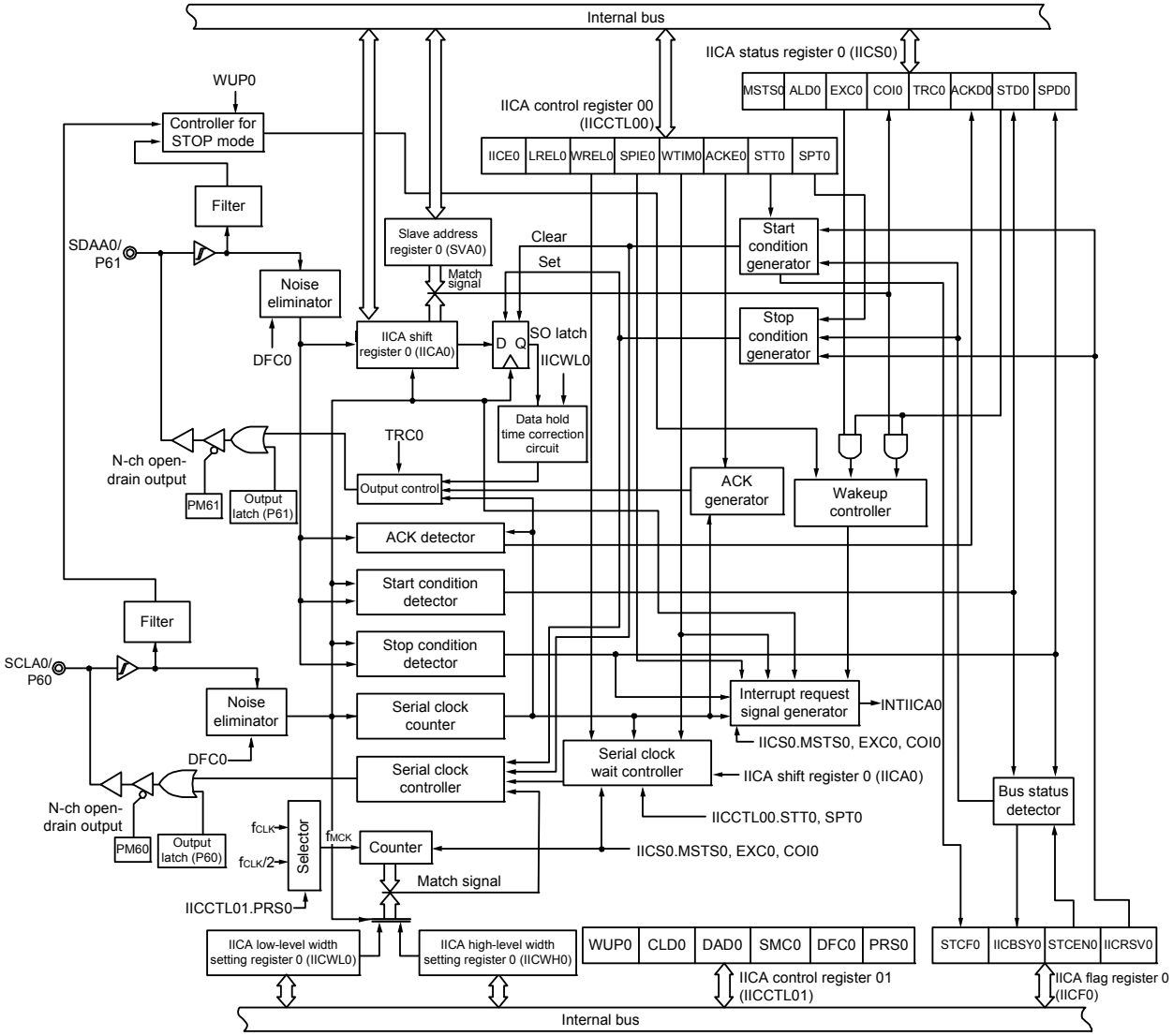
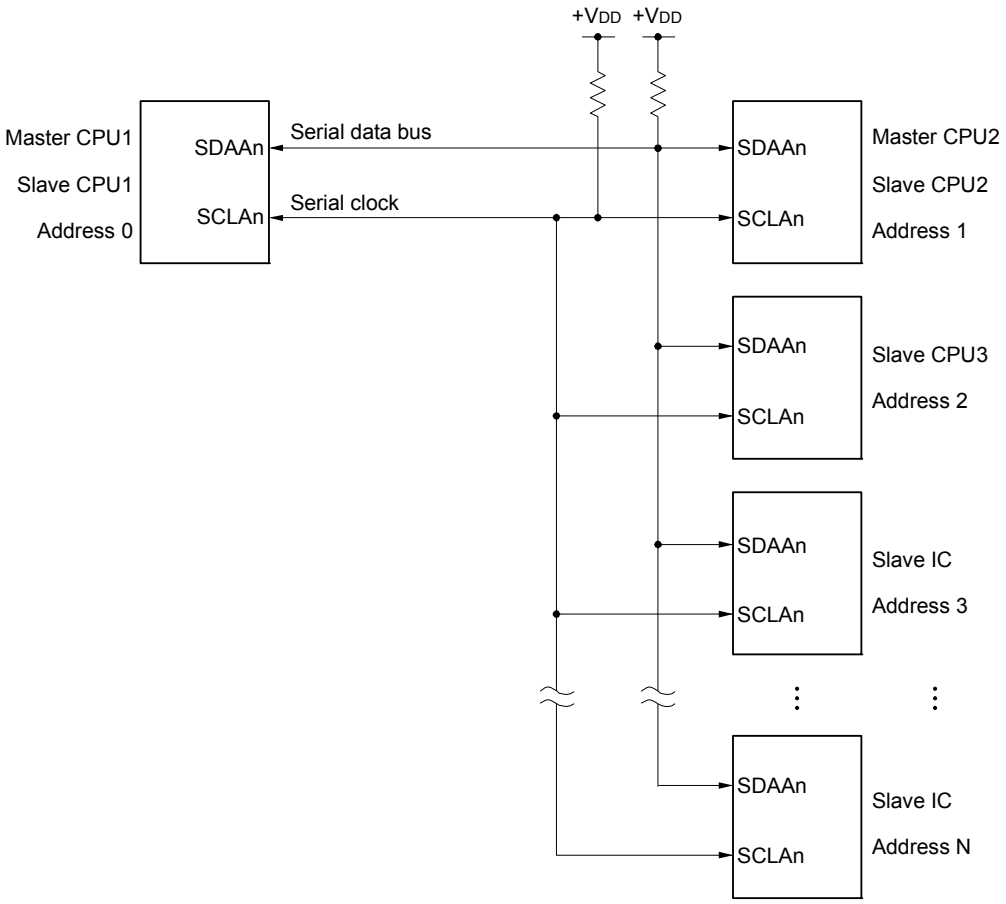


Figure 15 - 2 shows a serial bus configuration example.

Figure 15 - 2 Serial Bus Configuration Example Using I²C Bus



Remark n = 0, 1

15.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 15 - 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 register 6 (PM6) Port register 6 (P6)

Remark n = 0, 1

(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 15 - 3 Format of IICA shift register n (IICAn)

Address: FFF50H (IICA0), FFF54H (IICA1) 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, 1

(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 15 - 4 Format of Slave address register n (SVAn)

Address: F0234H (SVA0), F023CH (SVA1) 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, 1

(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, 1

15.3 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 register 6 (PM6)
- Port register 6 (P6)

Remark n = 0, 1

15.3.1 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 6, 4 (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 15 - 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	RTCWEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	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.

Caution 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 register 6 (PM6) and port register 6 (P6)).
- **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)**

Remark n = 0, 1

15.3.2 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, 1

Figure 15 - 6 Format of IICA control register n0 (IICCTLn0) (1/4)

Address: F0230H (IICCTL00), F0238H (IICCTL10) 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, 1

Figure 15 - 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, 1

Figure 15 - 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, 1

Figure 15 - 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, 1

15.3.3 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 15 - 10 Format of IICA status register n (IICSn) (1/3)

Address: FFF51H (IICS0), FFF55H (IICS1) 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, 1

Figure 15 - 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, 1

Figure 15 - 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, 1

15.3.4 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 15 - 13 Format of IICA flag register n (IICFn)

Address: FFF52H (IICF0), FFF56H (IICF1) 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)

Remark 2. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

15.3.5 IICA control register n1 (IICCTLn1)

This register is used to set the operation mode of I²C 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 I²C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation clears this register to 00H.

Figure 15 - 14 Format of IICA control register n1 (IICCTLn1) (1/2)

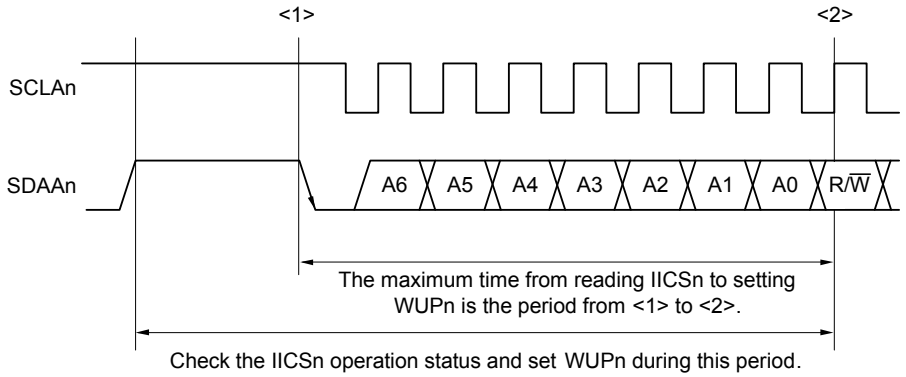
Address: F0231H (IICCTL01), F0239H (IICCTL11) After reset: 00H R/W **Note 1**

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 15 - 28 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, 1

Figure 15 - 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	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.)

Caution 3. The fast mode plus is only available in the products for "A: Consumer applications (T_A = -40 °C to +85 °C)" and "D: Industrial applications (T_A = -40 °C to +85 °C)".

Remark 1. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0, 1

15.3.6 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 data hold time is decided by value the higher 6 bits of IICWL register.

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

Set the IICWLn register while operation of I2C 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 **15.4.2 Setting transfer clock by using IICWLn and IICWHn registers**.

The data hold time is a quarter of the time specified by IICWLn.

Figure 15 - 16 Format of IICA low-level width setting register n (IICWLn)

Address: F0232H (IICWL0), F023AH (IICWL1) After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWLn								

15.3.7 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 I2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

Figure 15 - 17 Format of IICA high-level width setting register n (IICWHn)

Address: F0233H (IICWH0), F023BH (IICWH1) 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 **15.4.2 (1)** and **15.4.2 (2)**, respectively.

Remark 2. n = 0, 1

15.3.8 Register to control port function of serial I/O pin

When serial interface IICA is used, set the registers (port mode register 6 (PM6) and port register 6 (P6)) that controls the port functions for alternately used with SCLAn pin and SDAAn pin.
For details, see 5.3.1 Port mode registers (PMxx), 5.3.2 Port registers (Pxx), and 5.3.6 Port mode control registers (PMCxx).

Set the IICEn bit (bit 7 of IICA control register n0 (IICCTLn0)) to 1 before setting the output mode because the SCLAn and SDAAn pins output a low level (fixed) when the IICEn bit is 0.

15.4 I²C Bus Mode Functions

15.4.1 Pin configuration

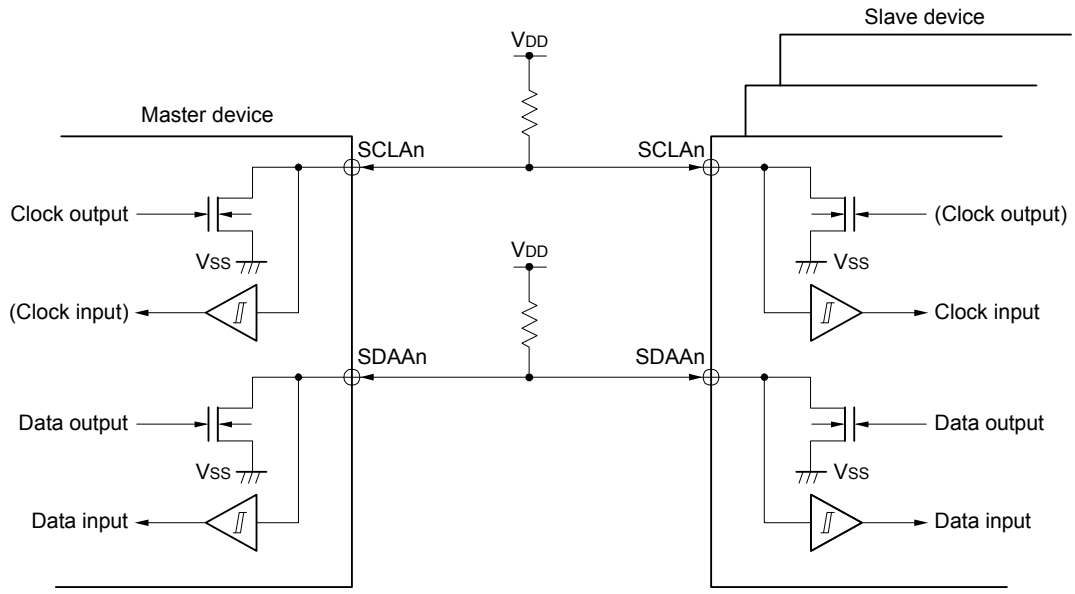
The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are configured as follows.

(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 15 - 18 Pin Configuration Diagram



Remark n = 0, 1

15.4.2 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.

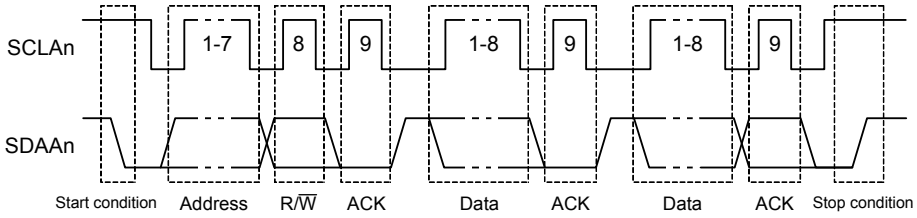
Remark 2. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
t_R: SDAAn and SCLAn signal rising times
f_{MCK}: IICA operating clock frequency

Remark 3. n = 0, 1

15.5 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. Figure 15 - 19 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 15 - 19 I²C Bus Serial Data Transfer Timing

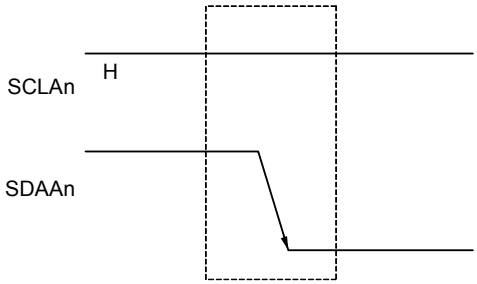


The master device generates the start condition, slave address, and stop condition.
The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).
The serial clock (SCLAn) is continuously output by the master device. However, in the slave device, the SCLAn pin low level period can be extended and a wait can be inserted.

15.5.1 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 15 - 20 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, 1

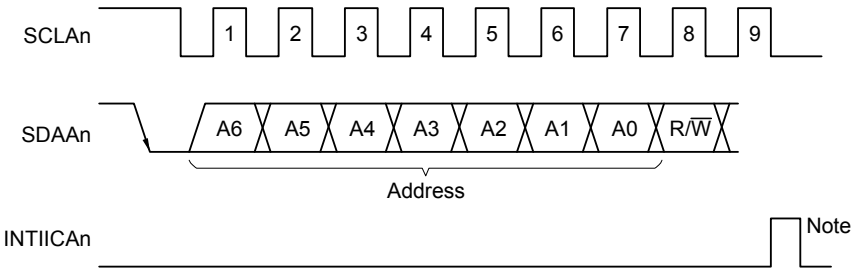
15.5.2 Addresses

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 15 - 21 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 **15.5.3 Transfer direction specification** are written to the IICA shift register n (IICAn). The received addresses are written to the IICAn register.

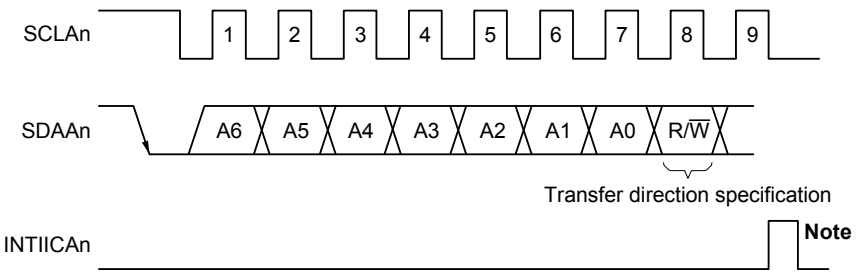
The slave address is assigned to the higher 7 bits of the IICAn register.

15.5.3 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 15 - 22 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, 1

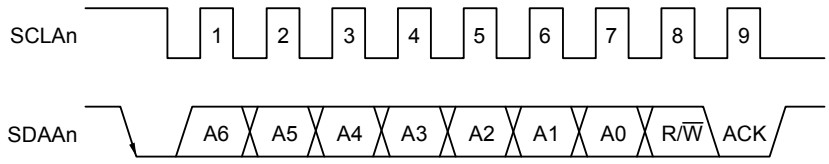
15.5.4Acknowledge (ACK)

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 15 - 23 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.

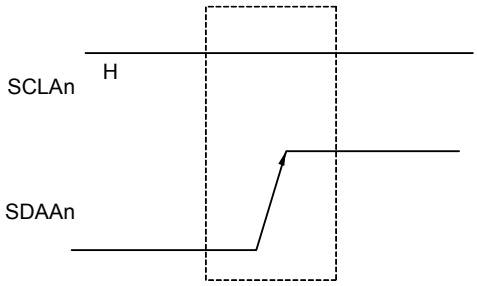
Remark n = 0, 1

15.5.5 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 15 - 24 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, 1

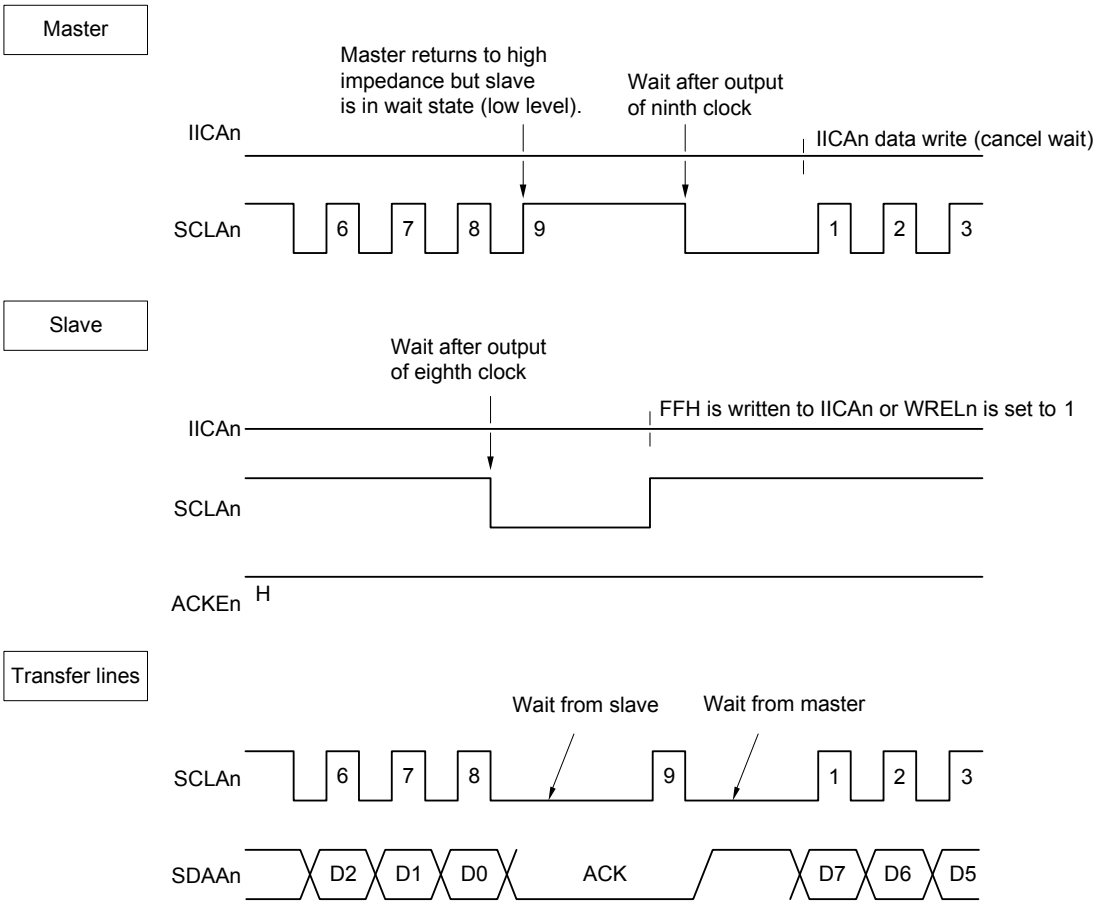
15.5.6 Wait

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 15 - 25 Wait (1/2)

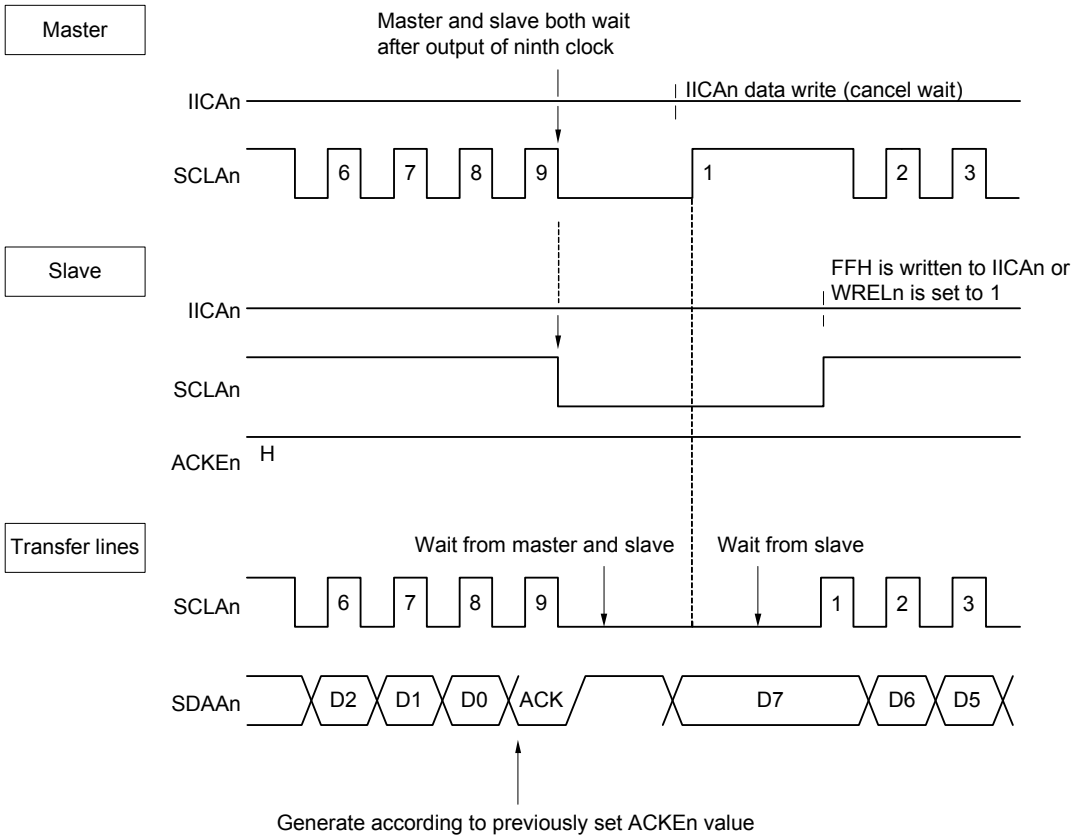
(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKEn = 1)



Remark n = 0, 1

Figure 15 - 26 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, 1

15.5.7 Canceling wait

The I^2C 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 I^2C 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 I^2C 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, 1

15.5.8 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 15 - 2.

Table 15 - 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, 1

(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).

15.5.9 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.

15.5.10 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, 1

15.5.11 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 15 - 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, 1

15.5.12 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.

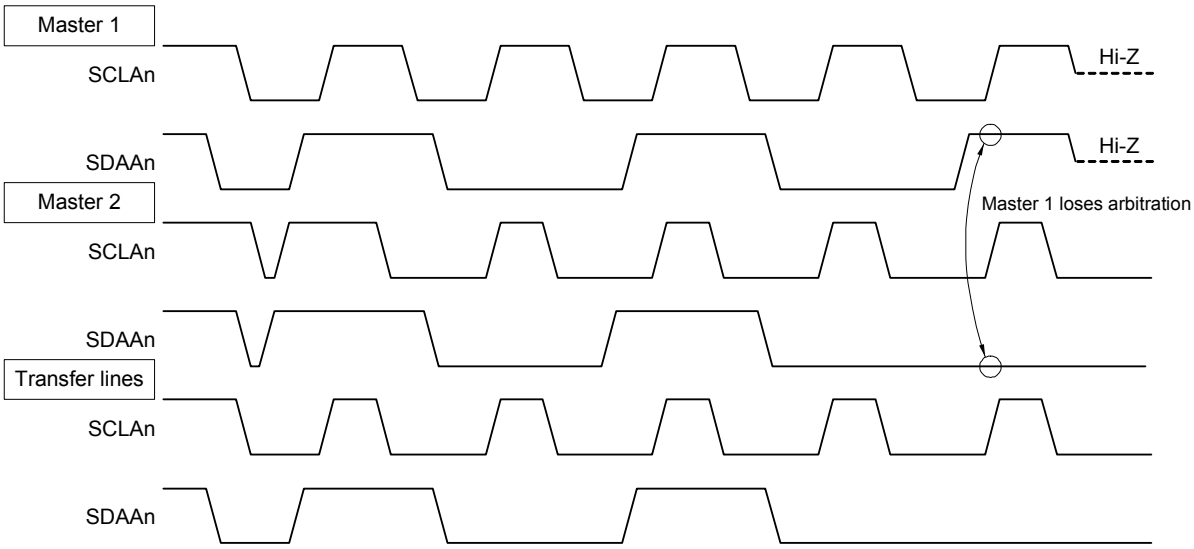
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in the IICA status register n (IICSn) is set (1) via the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus.

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 **15.5.8 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 15 - 27 Arbitration Timing Example



Remark n = 0, 1

Table 15 - 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, 1

15.5.13 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.

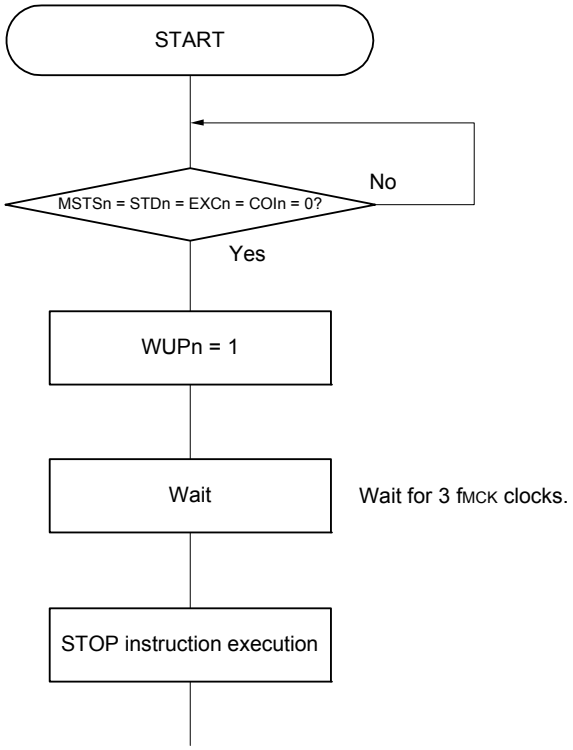
This function makes processing more efficient by preventing unnecessary INTIICAn signal from occurring when addresses do not match.

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.

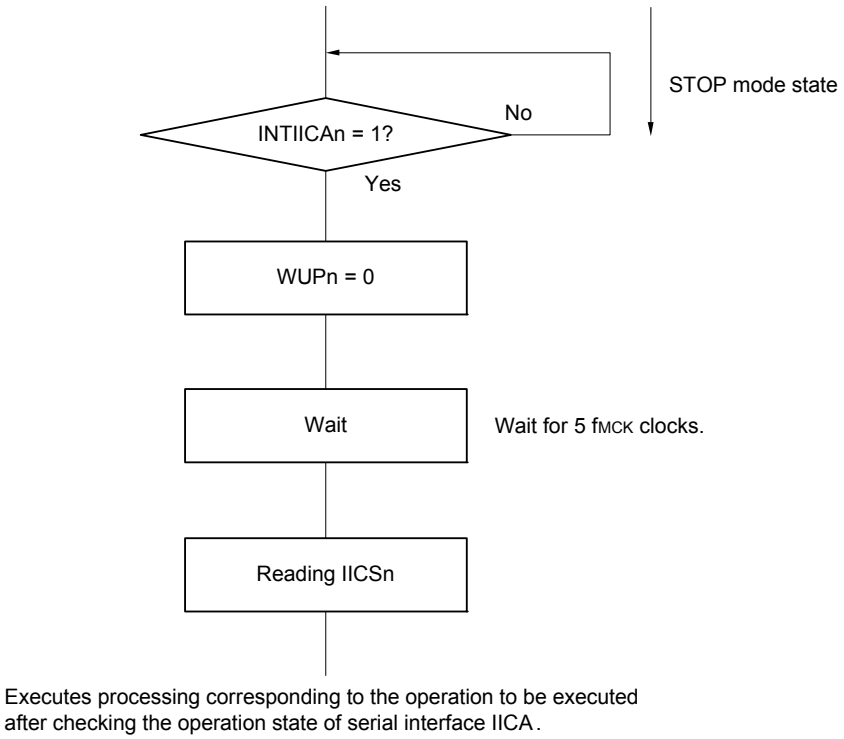
Figure 15 - 28 shows the flow for setting WUPn = 1 and Figure 15 - 29 shows the flow for setting WUPn = 0 upon an address match.

Figure 15 - 28 Flow When Setting WUPn = 1



Remark n = 0, 1

Figure 15 - 29 Flow When Setting WUPn = 0 upon Address Match (Including Extension Code Reception)

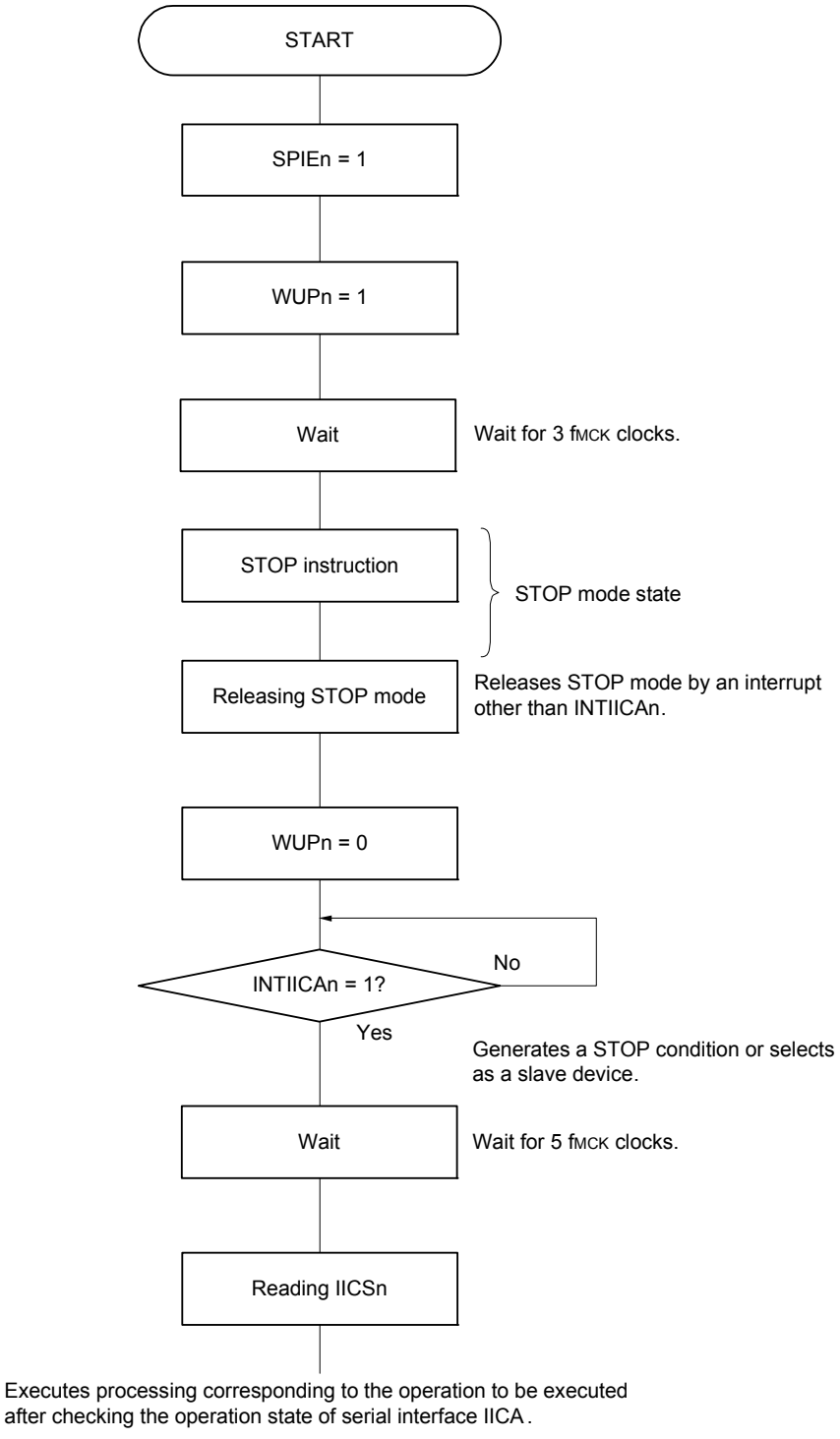


Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICAn) generated from serial interface IICA.

- When operating as the master device for the next IIC communication: Flow shown in Figure 15 - 30
- 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 15 - 29
  - 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, 1

Figure 15 - 30 When Operating as Master Device after Releasing STOP Mode other than by INTIICAn



Remark n = 0, 1

15.5.14 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.

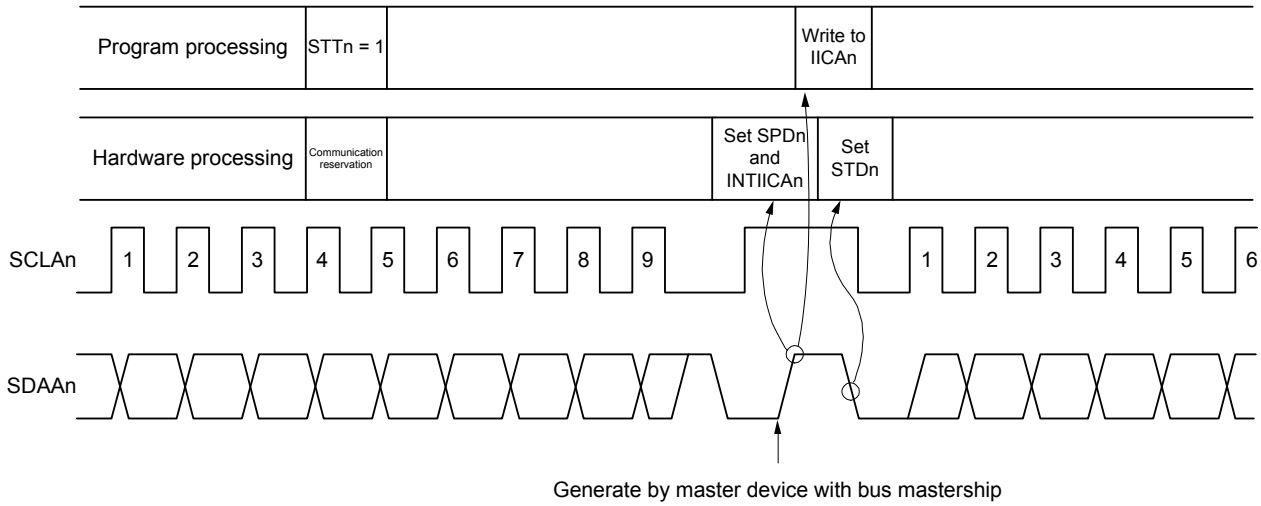
Wait time (number of f_{MCK} clocks) from setting STTn = 1 to checking the MSTSn flag:
(IICWLn setting value + IICWHn setting value + 4) + $t_F \times 2 \times f_{MCK}$ [clocks]

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, 1

Figure 15 - 31 shows the Communication Reservation Timing.

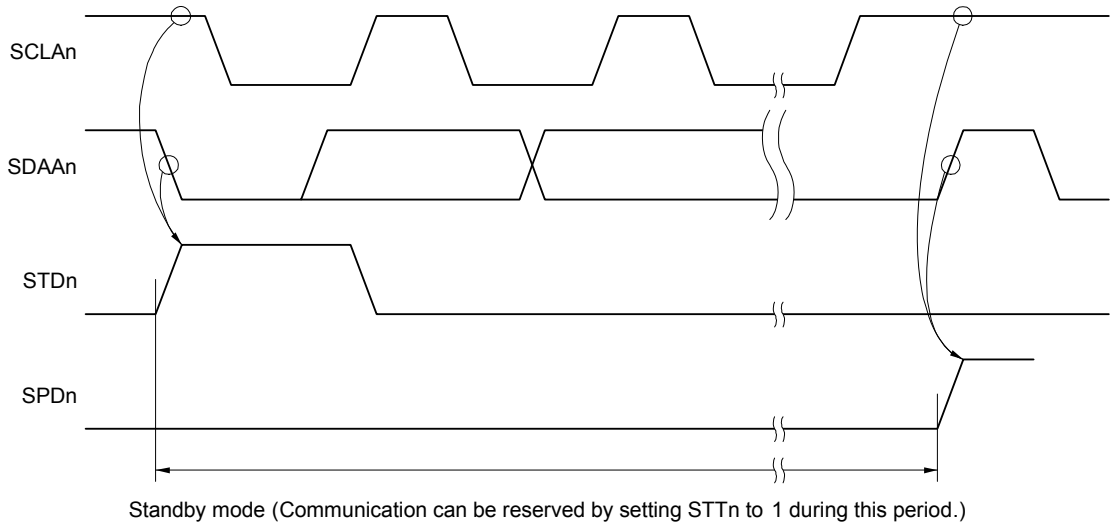
Figure 15 - 31 Communication Reservation Timing



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 15 - 32. 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.

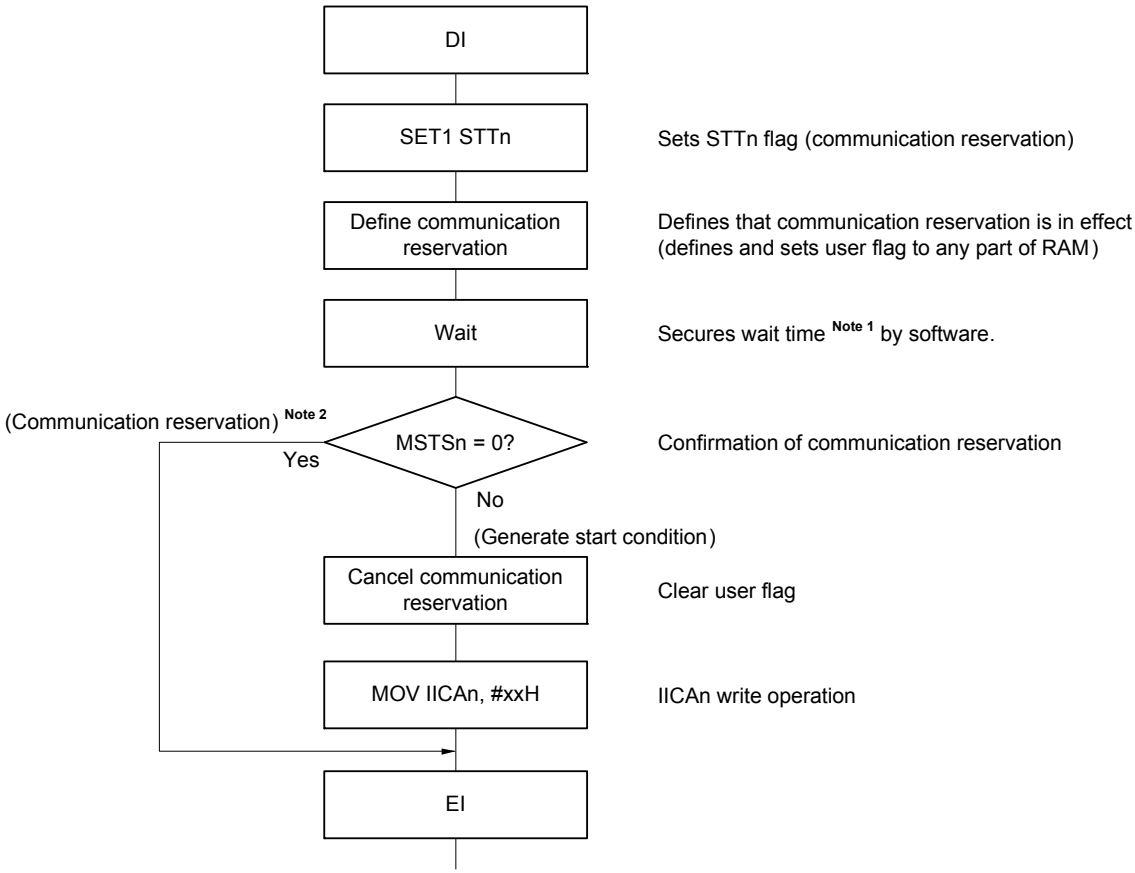
Figure 15 - 32 Timing for Accepting Communication Reservations



Remark n = 0, 1

Figure 15 - 33 shows the Communication Reservation Protocol.

Figure 15 - 33 Communication Reservation Protocol



Note 1. The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4) + $t_F \times 2 \times f_{MCK}$: [clocks]

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, 1

(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, 1

15.5.15 Cautions

(1) When STCENn = 0

Immediately after I^2C 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 I^2C 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 I^2C communications are already in progress

If I^2C operation is enabled and the device participates in communication already in progress when the SDAAn pin is low and the SCLAn pin is high, the macro of I^2C 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 I^2C communications. To avoid this, start I^2C 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 I^2C.
<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, 1

15.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the RL78/G1H 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 RL78/G1H 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 RL78/G1H 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 RL78/G1H is used as the I^2C bus slave is shown below.

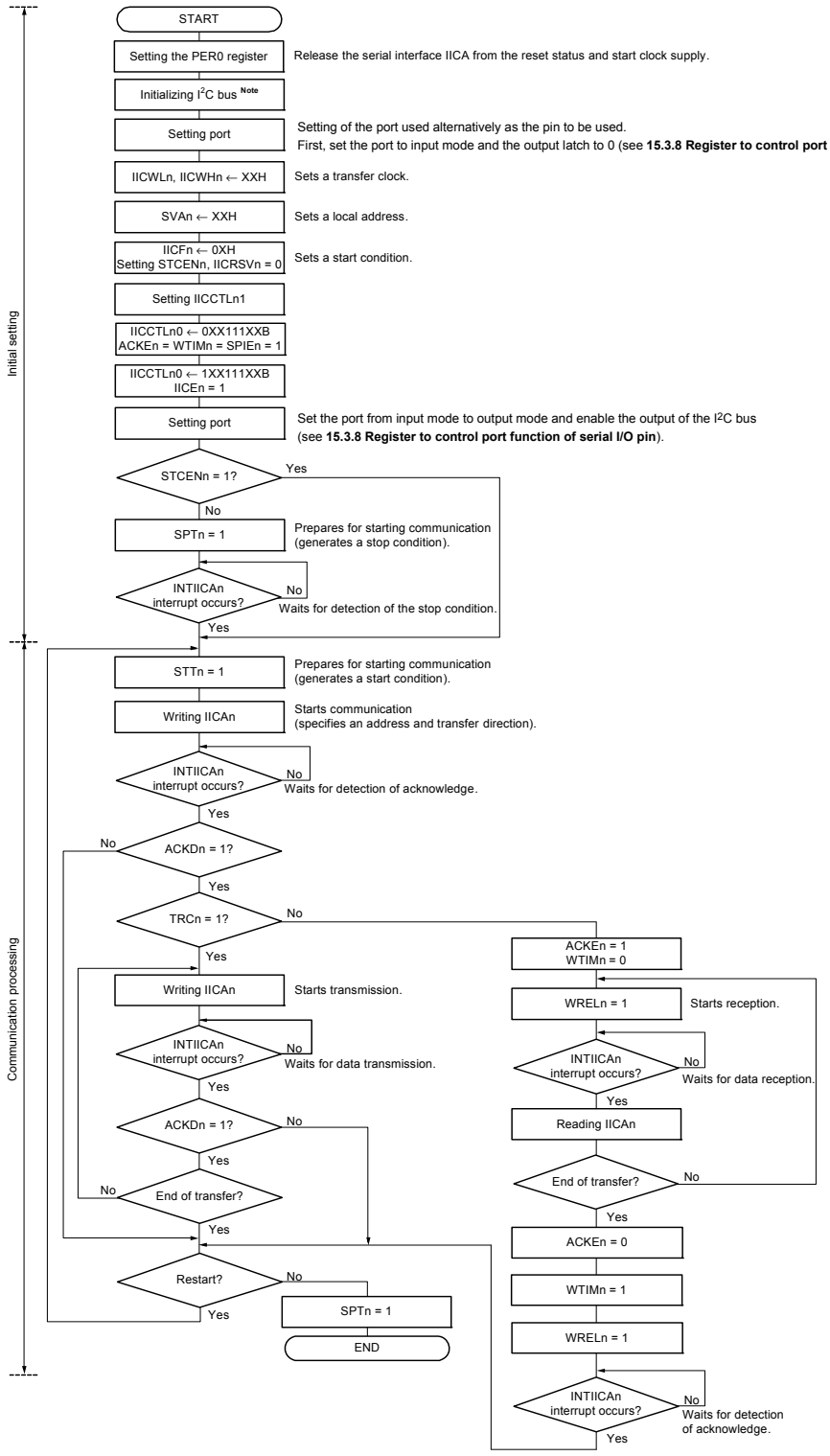
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICAn interrupt occurrence (communication waiting). When an INTIICAn interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

Remark n = 0, 1

(1) Master operation in single master system

Figure 15 - 34 Master Operation in Single-Master System



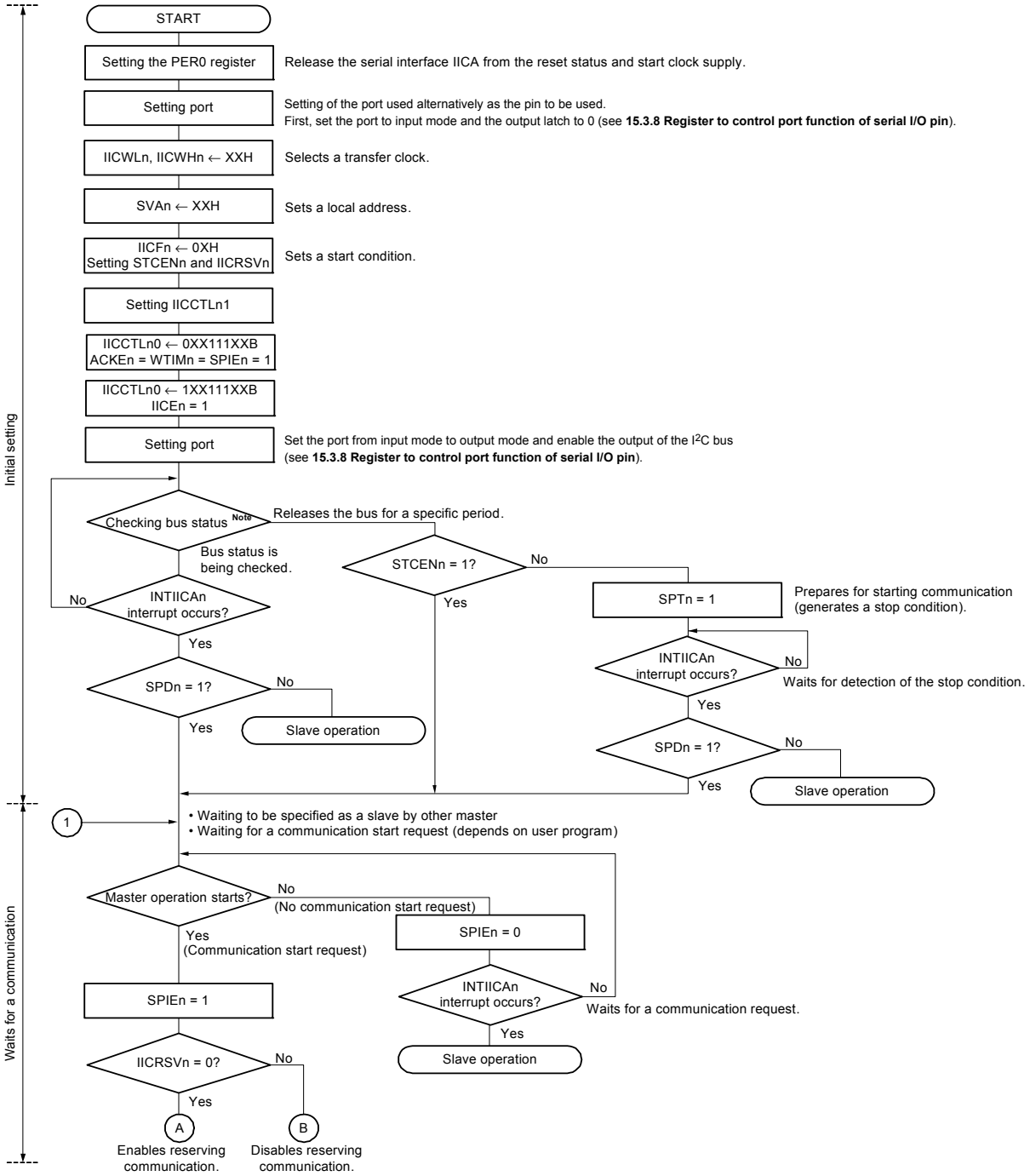
Note Release (SCLAn and SDAAn pins = high level) the I2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the 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, 1

(2) Master operation in multimaster system

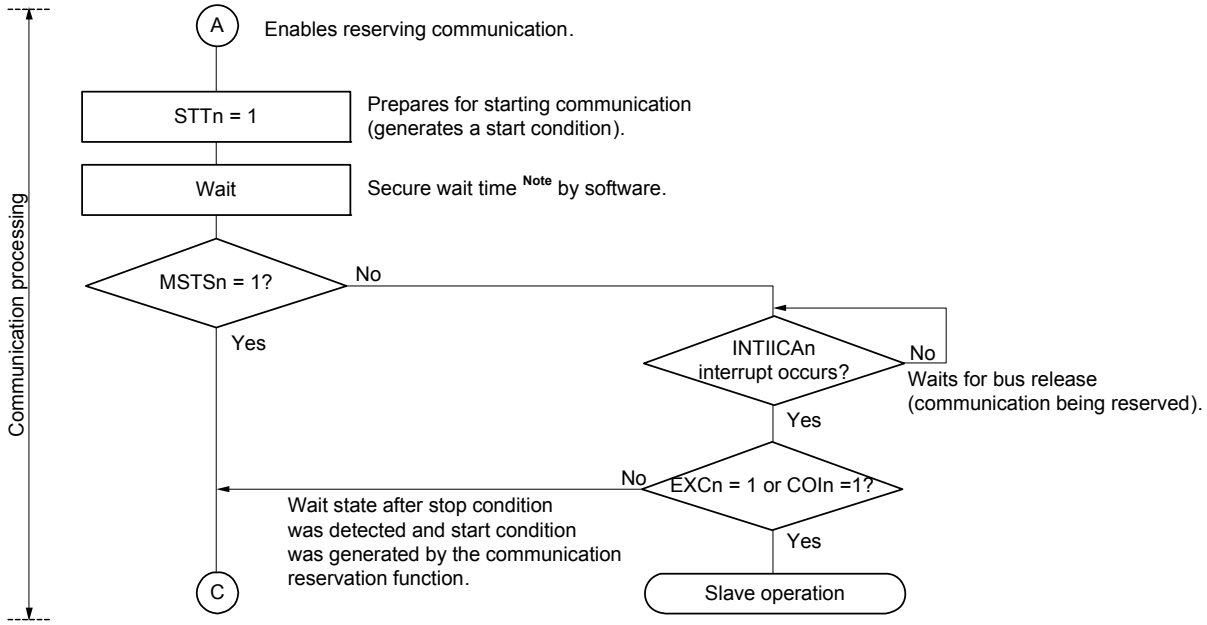
Figure 15 - 35 Master Operation in Multi-Master System (1/3)



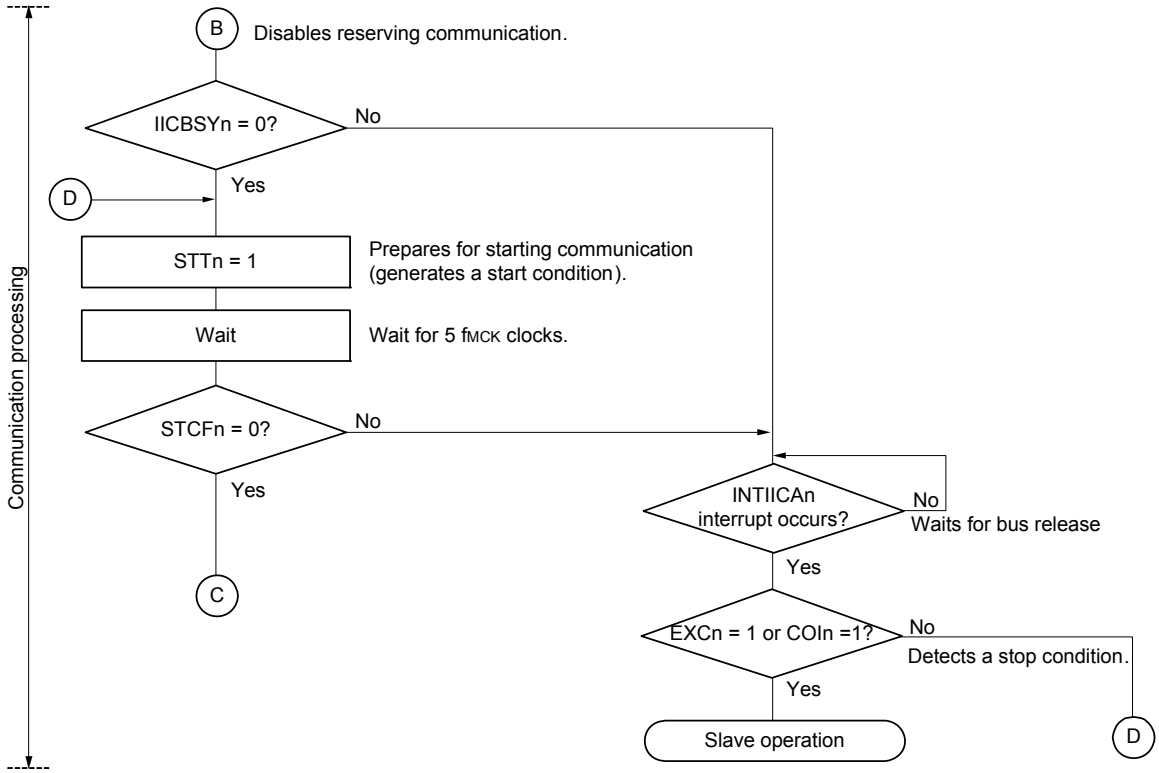
Note Confirm that the bus is released (CLDn bit = 1, DADn bit = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I^2C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

Remark n = 0, 1

Figure 15 - 36 Master Operation in Multi-Master System (2/3)



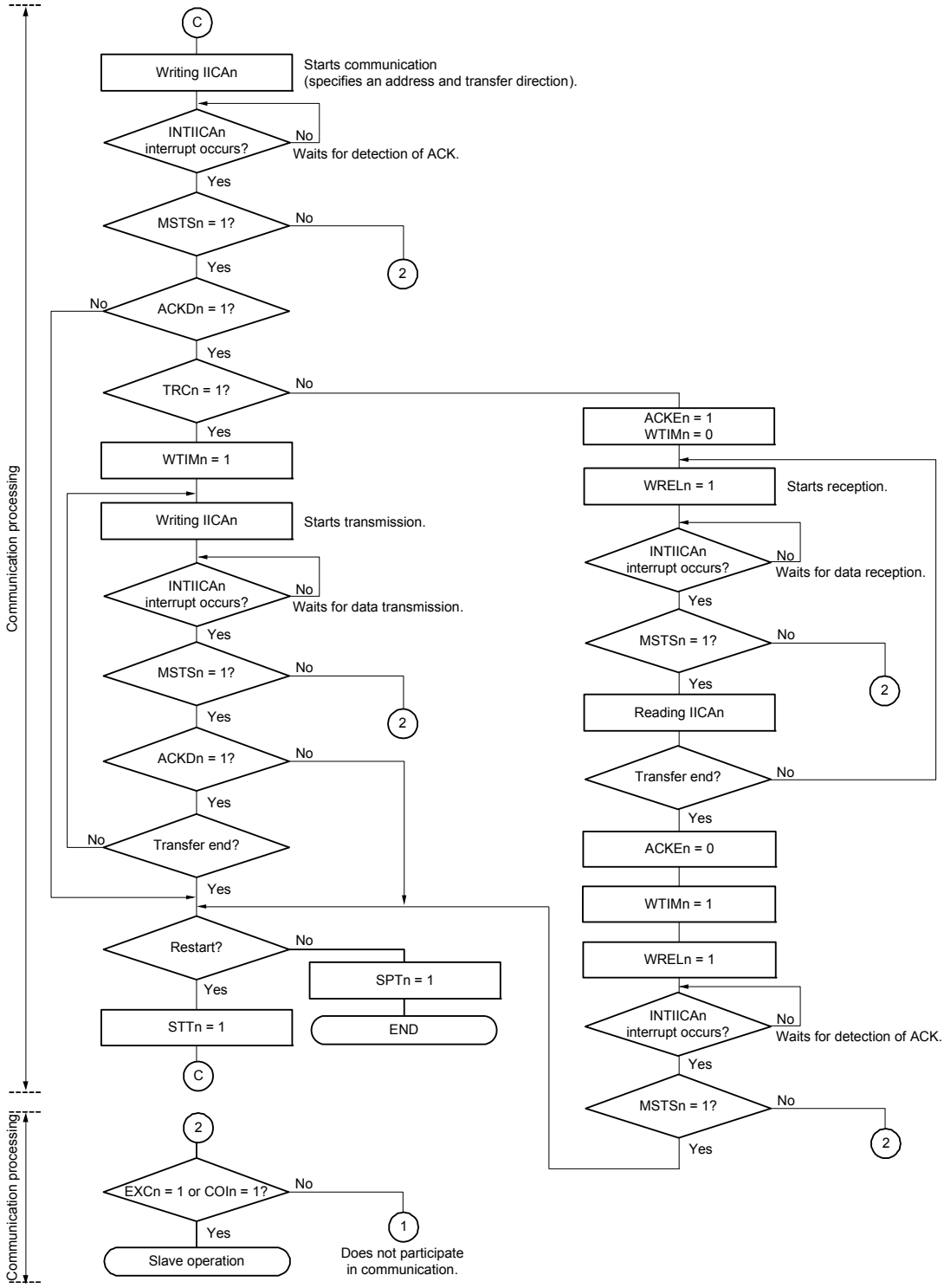
Note The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4) / f_{MCK} + $t_F \times 2$



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, 1

Figure 15 - 37 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.

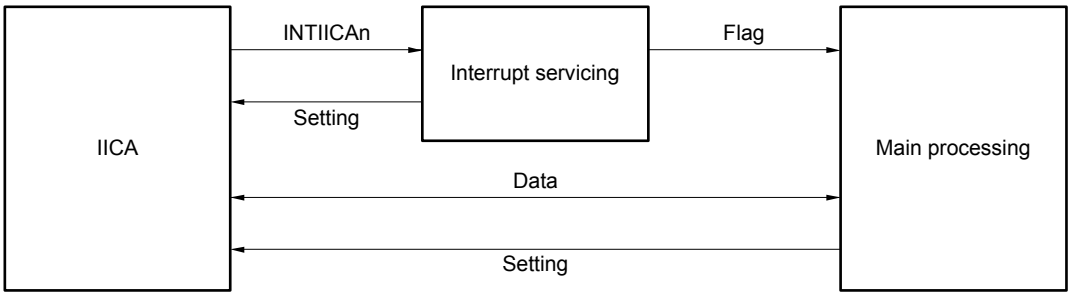
Remark 4. n = 0, 1

(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, 1

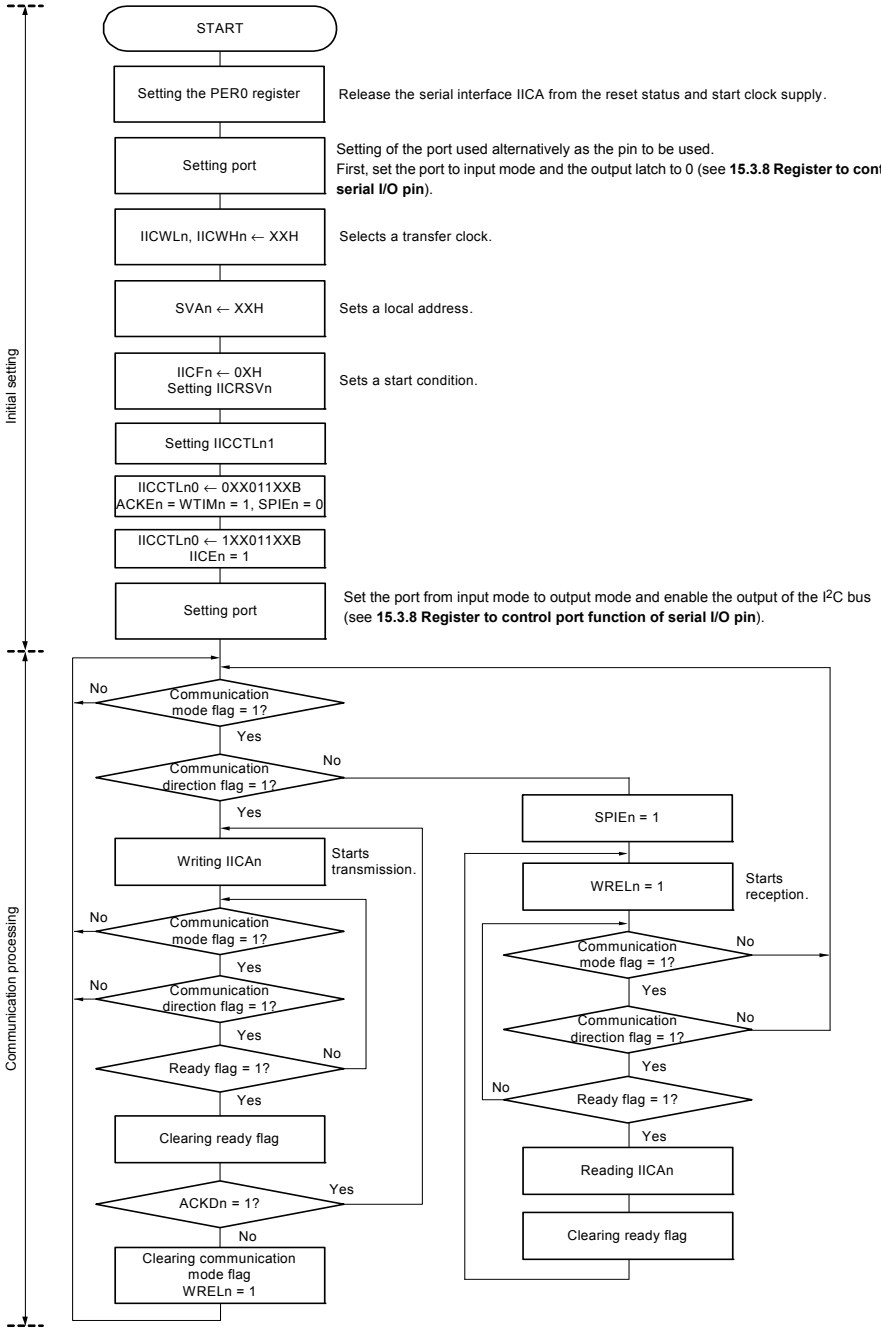
The main processing of the slave operation is explained next.

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 15 - 38 Slave Operation Flowchart (1)



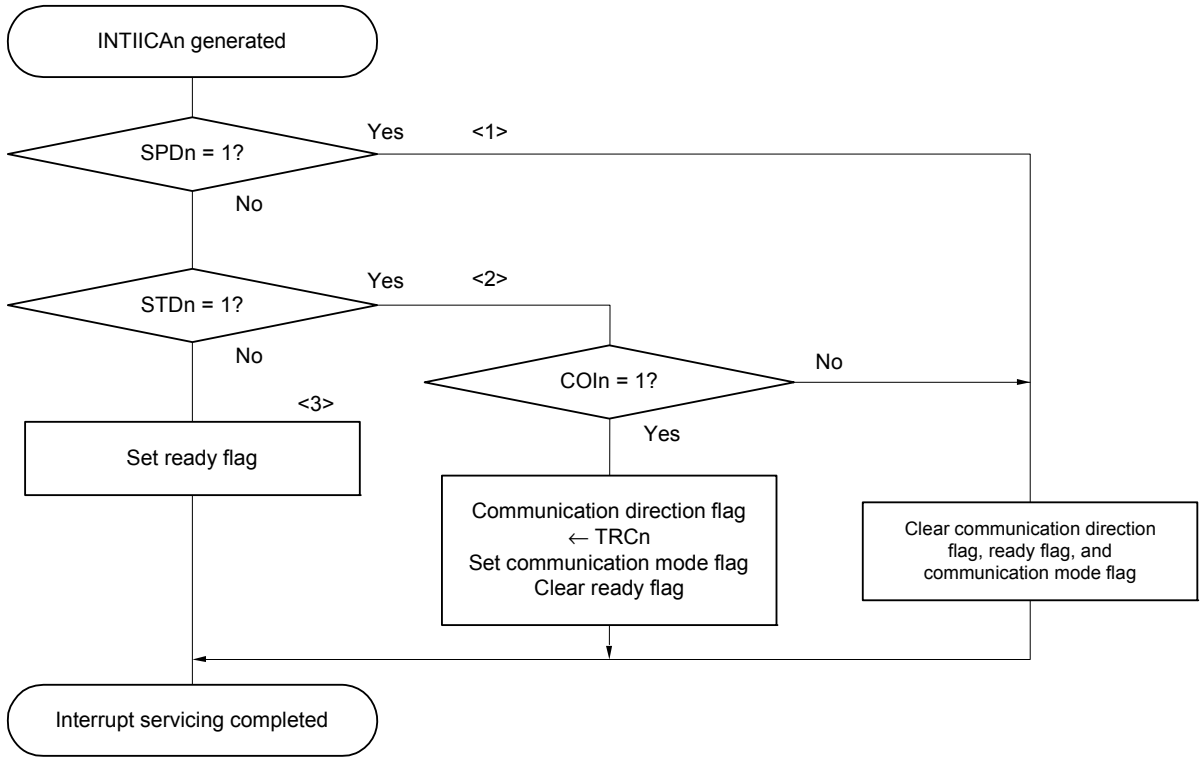
Remark1. Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.
Remark2. n = 0, 1

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 15 - 39 Slave Operation Flowchart (2).

Figure 15 - 39 Slave Operation Flowchart (2)



Remark n = 0, 1

15.5.17 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.

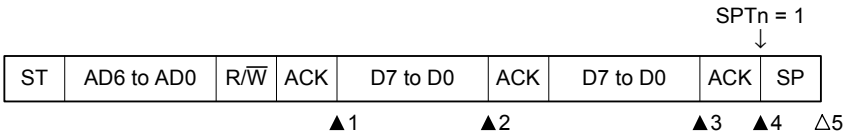
Remark 1. ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
ACK: Acknowledge
D7 to D0: Data
SP: Stop condition

Remark 2. n = 0, 1

(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)

△5: IICSn = 00000001B

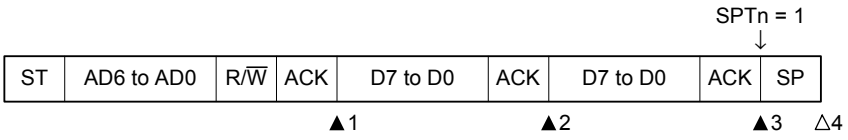
Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

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

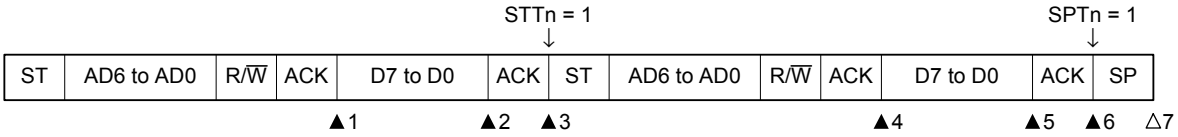
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(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.

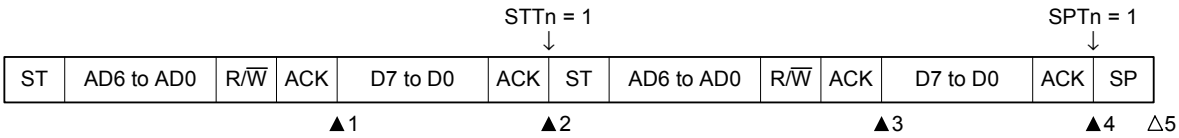
Note 3. To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

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

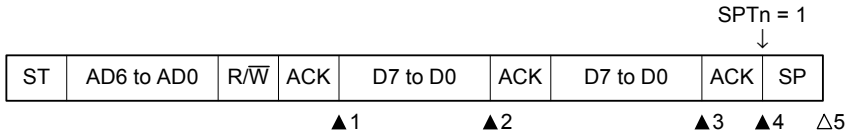
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(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

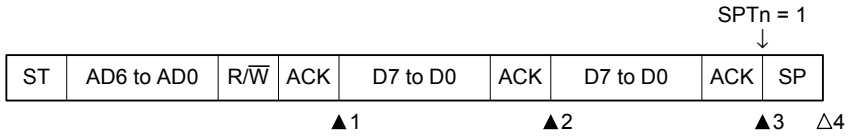
Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

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

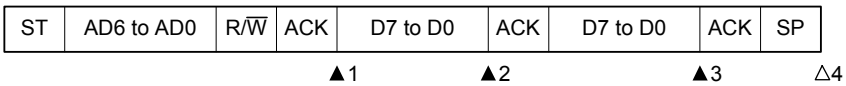
×: Don't care

Remark n = 0, 1

(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

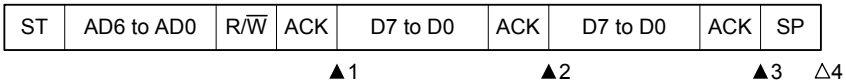
△4: IICSn = 00000001B

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

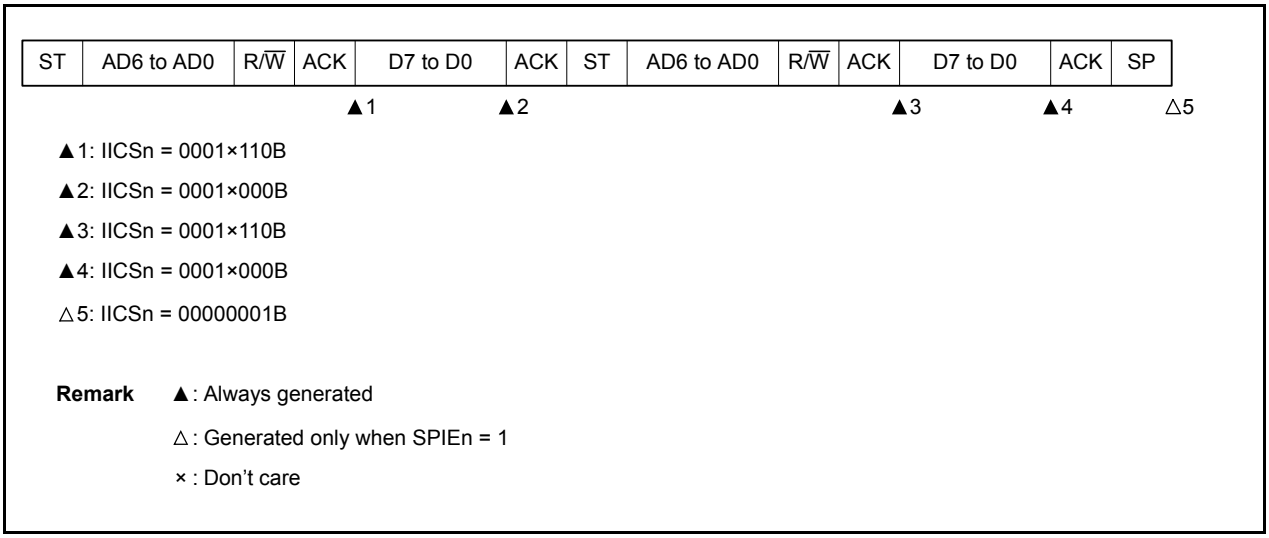
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(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

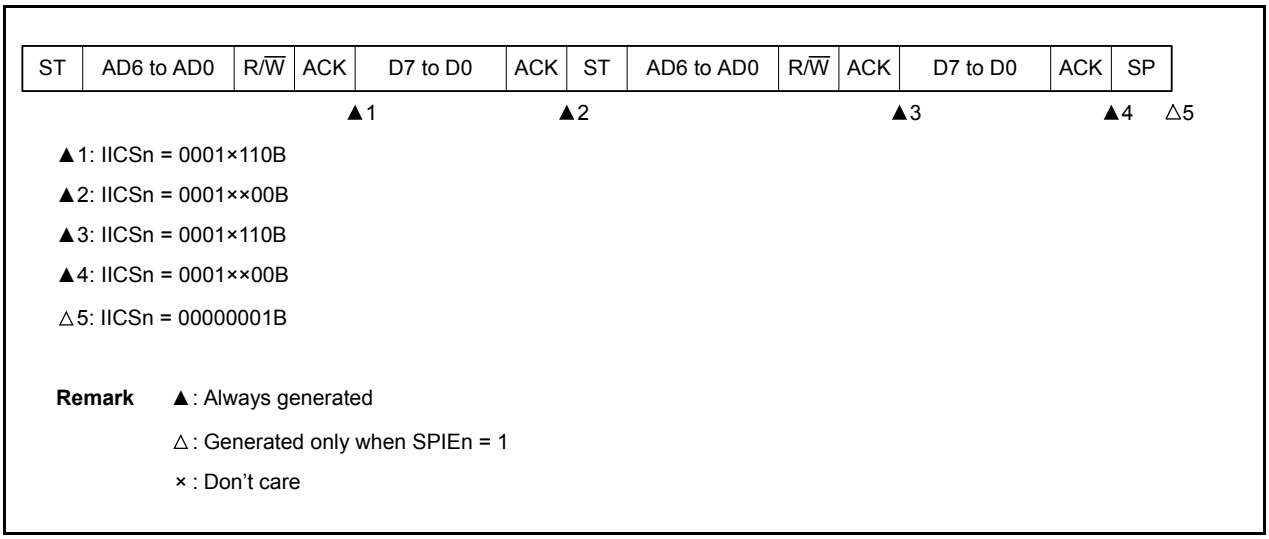
△5: IICSn = 00000001B

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

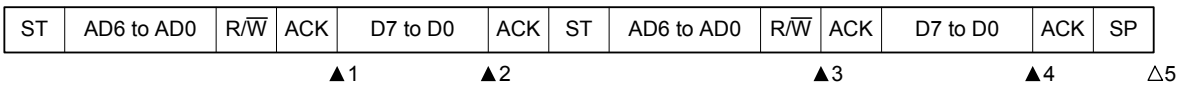
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= extension code))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲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, does not match address (= extension code))

▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×110B

▲5: IICSn = 0010××00B

△6: IICSn = 00000001B

Remark ▲: Always generated

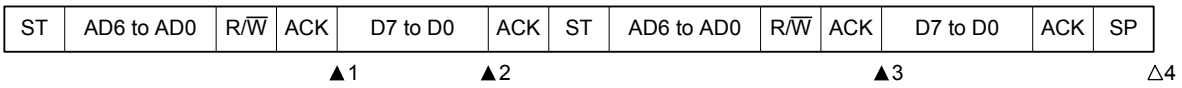
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(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

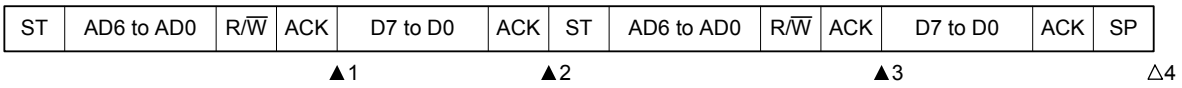
△4: IICSn = 00000001B

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

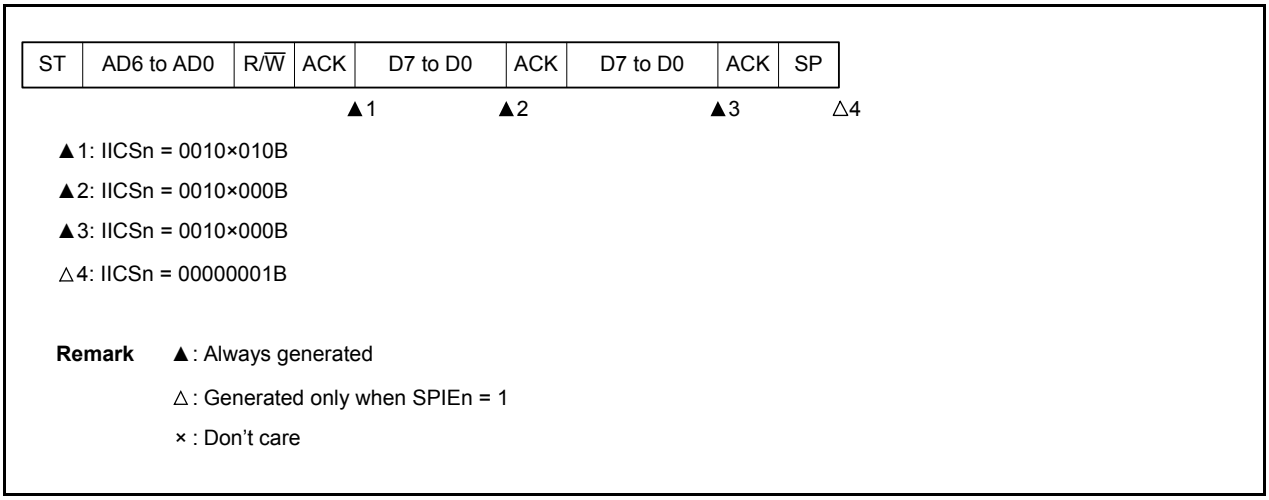
Remark n = 0, 1

(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

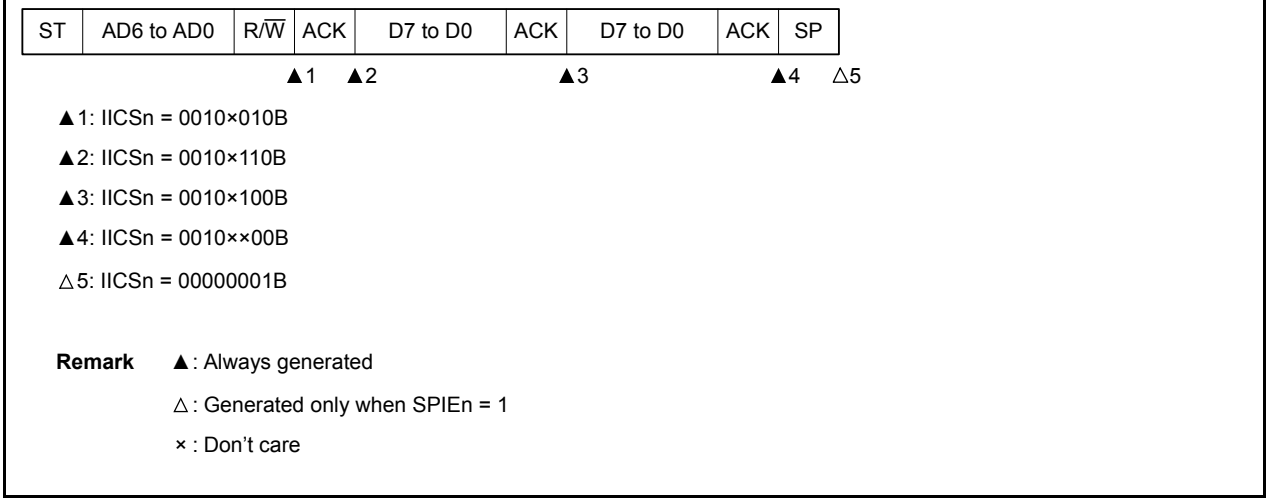
△4: IICSn = 00000001B

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

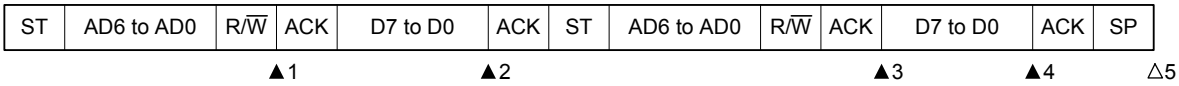
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, matches SVAn)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001×000B

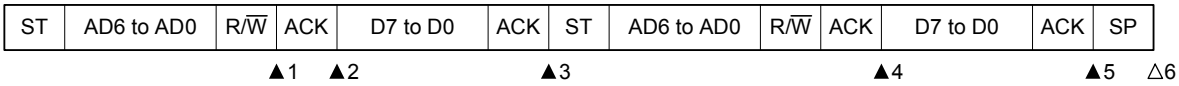
△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, matches SVAn)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010××00B

▲4: IICSn = 0001×110B

▲5: IICSn = 0001××00B

△6: IICSn = 00000001B

Remark ▲: Always generated

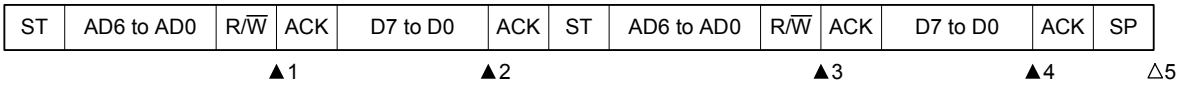
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(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

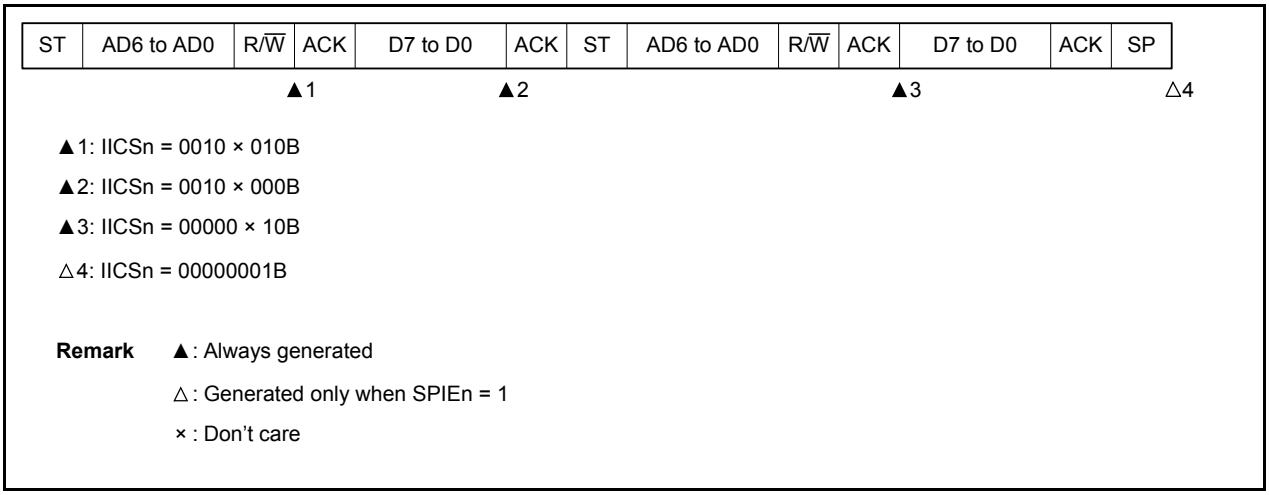
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(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

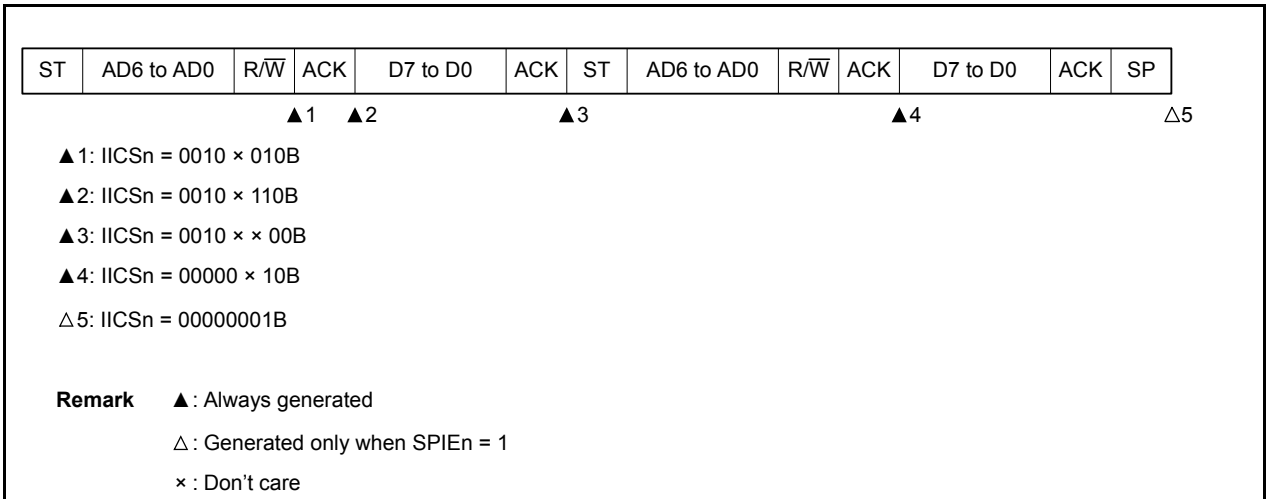
△4: IICSn = 00000001B

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

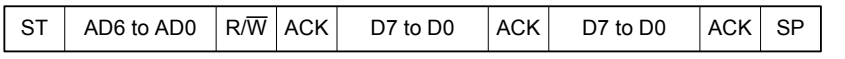
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICSn = 00000001B

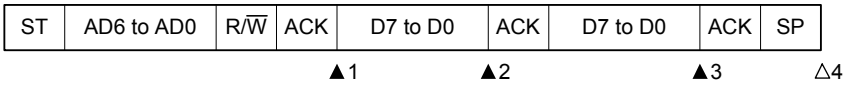
Remark △: Generated only when SPIEn = 1

(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

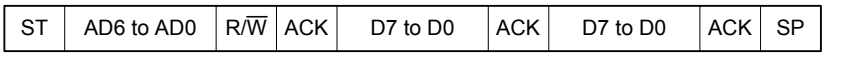
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1 ▲2 ▲3 △4

▲1: IICSn = 0101×110B

▲2: IICSn = 0001×100B

▲3: IICSn = 0001××00B

△4: IICSn = 00000001B

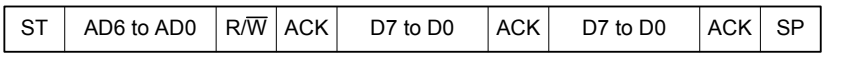
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIMn = 0



▲1 ▲2 ▲3 △4

▲1: IICSn = 0110×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×000B

△4: IICSn = 00000001B

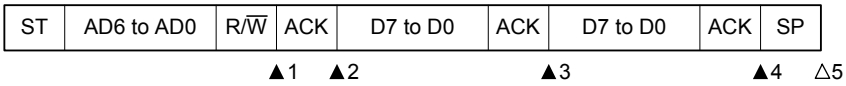
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 0110×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010×100B

▲4: IICSn = 0010××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

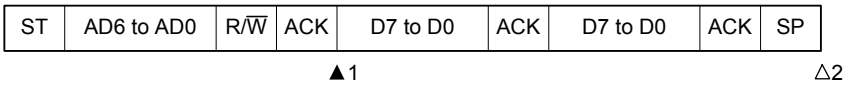
△: Generated only when SPIEn = 1

×: 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: IICSn = 01000110B

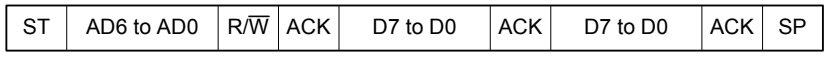
△2: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

Remark n = 0, 1

(b) When arbitration loss occurs during transmission of extension code



▲1 △2

▲1: IICSn = 0110×010B

Sets LRELn = 1 by software

△2: IICSn = 00000001B

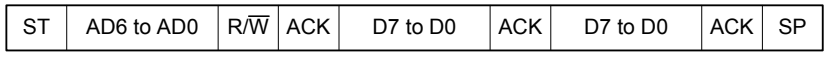
Remark ▲: Always generated

△: 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

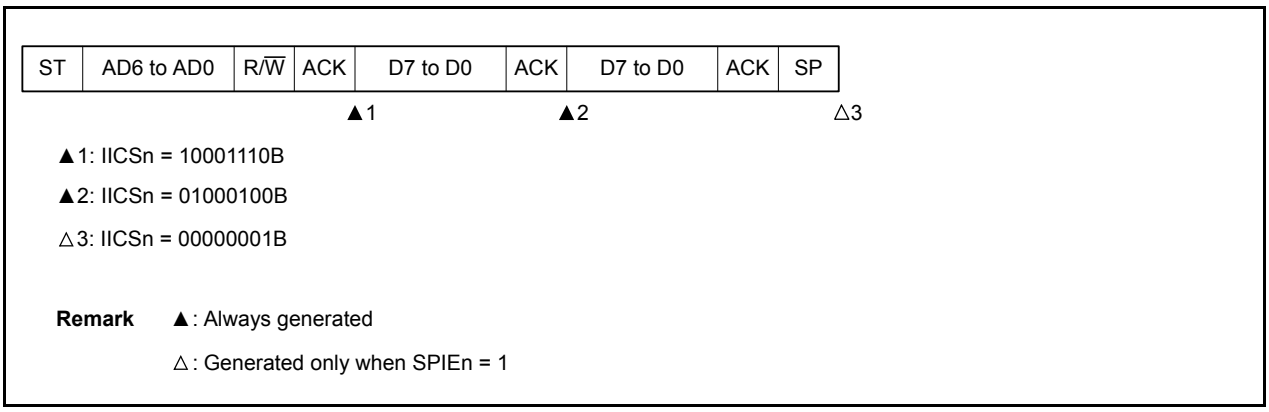
△3: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 10001110B

▲2: IICSn = 01000100B

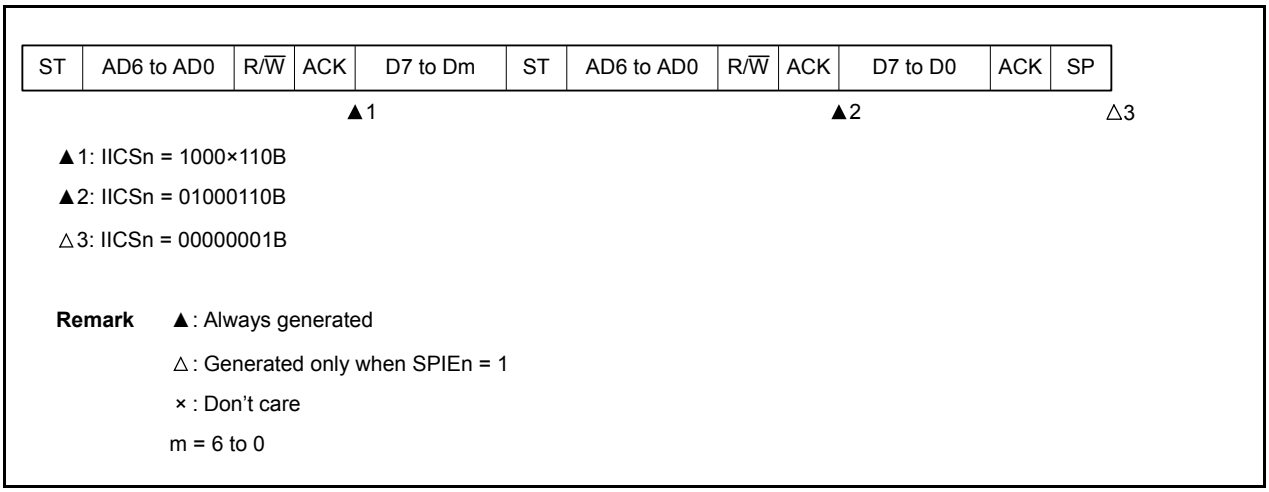
△3: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatches with SVAn)



▲1: IICSn = 1000×110B

▲2: IICSn = 01000110B

△3: IICSn = 00000001B

Remark ▲: Always generated

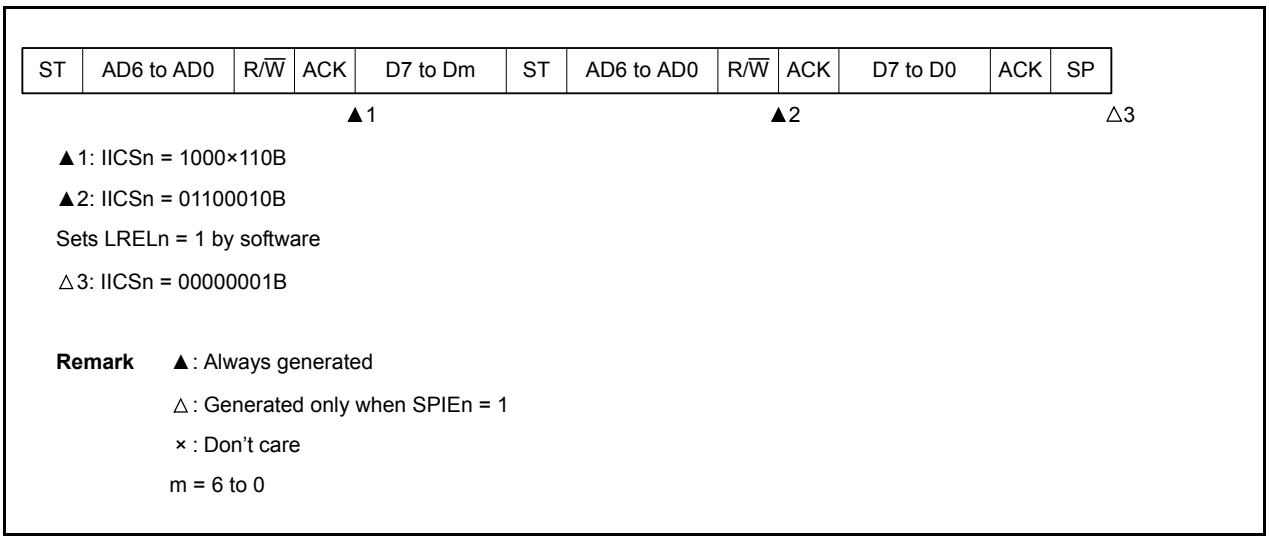
△: Generated only when SPIEn = 1

×: Don't care

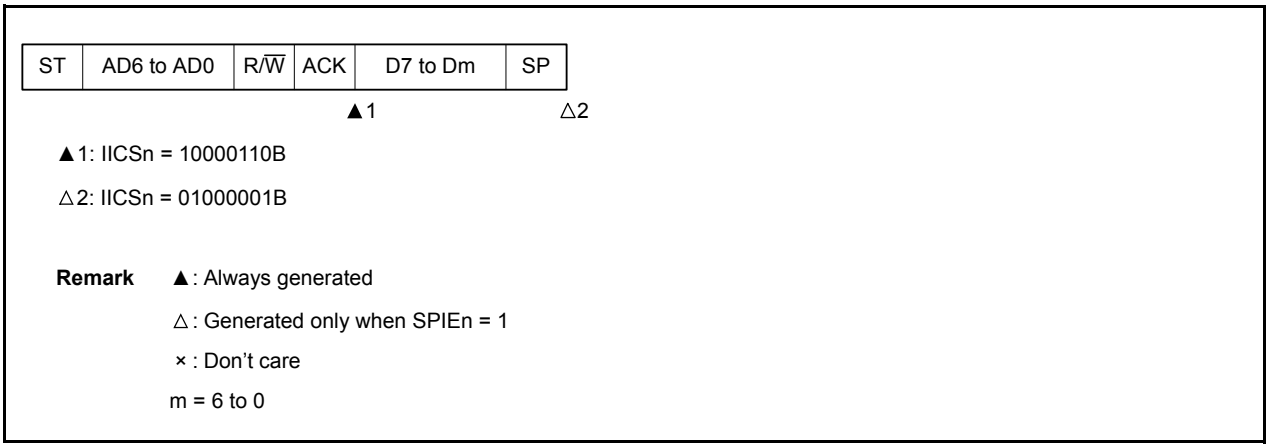
m = 6 to 0

Remark n = 0, 1

(ii) Extension code



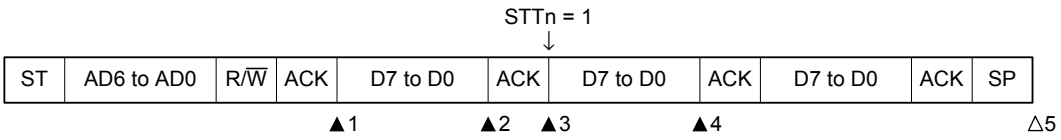
(e) When loss occurs due to stop condition during data transfer



Remark n = 0, 1

(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

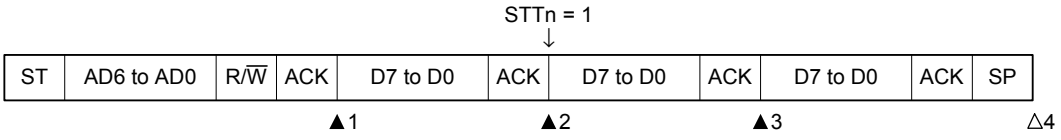
△5: IICSn = 00000001B

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

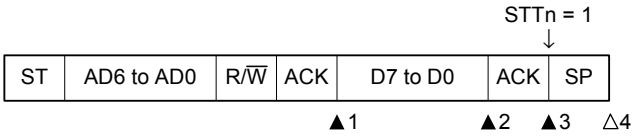
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(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)

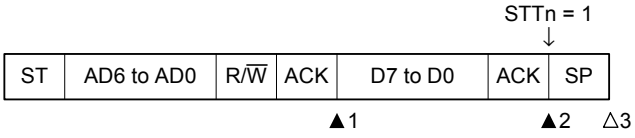
△4: IICSn = 01000001B

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

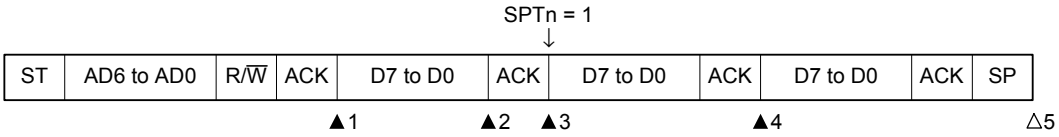
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(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

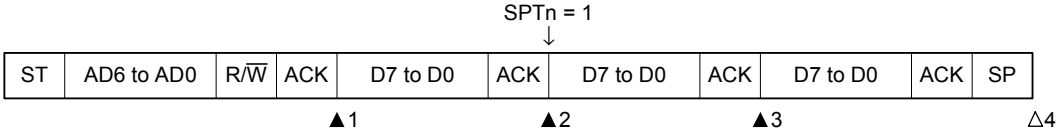
△5: IICSn = 00000001B

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, 1

15.6 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 15 - 40 to 15 - 46 show timing charts of the data communication.

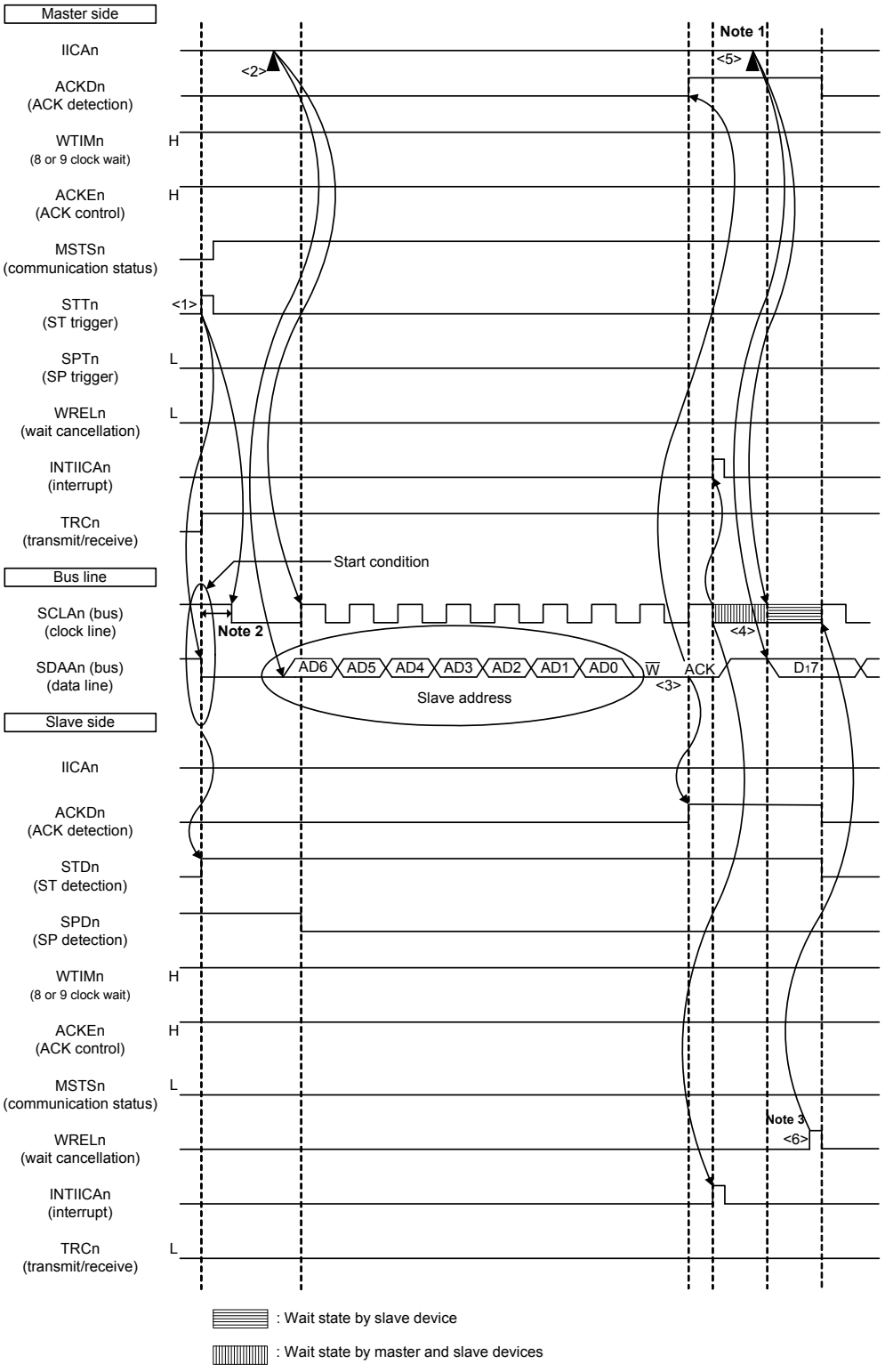
The IICA shift register n (IICAn)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin.

Data input via the SDAAn pin is captured into IICAn at the rising edge of SCLAn.

Remark n = 0, 1

Figure 15 - 40 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, 1

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 15 - 40 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.

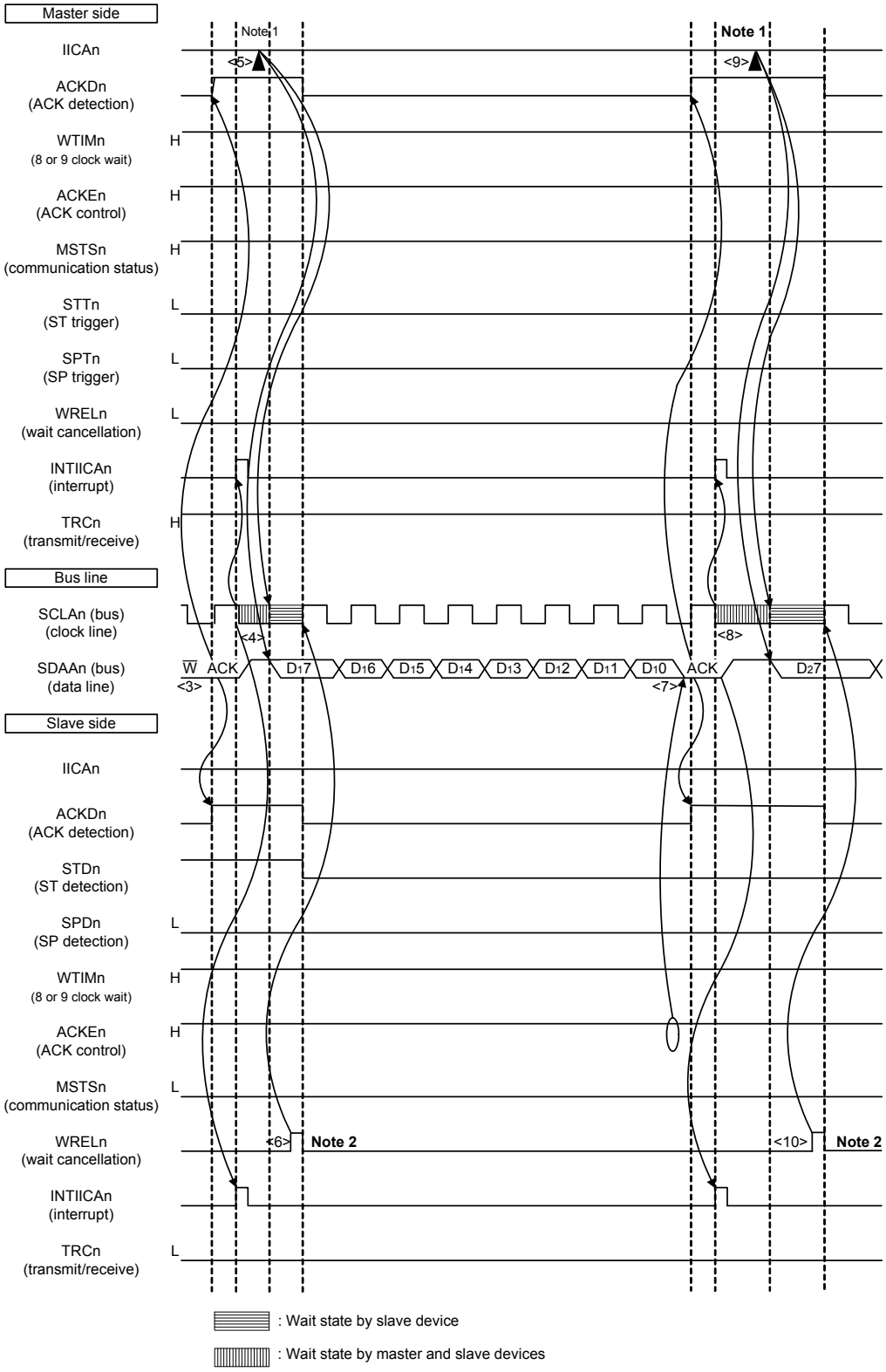
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in Figures 15 - 40 to 15 - 42 represent the entire procedure for communicating data using the I^2C bus. Figure 15 - 40 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15 - 41 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15 - 42 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0, 1

Figure 15 - 41 Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/4)

(2) 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, 1

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 15 - 41 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.

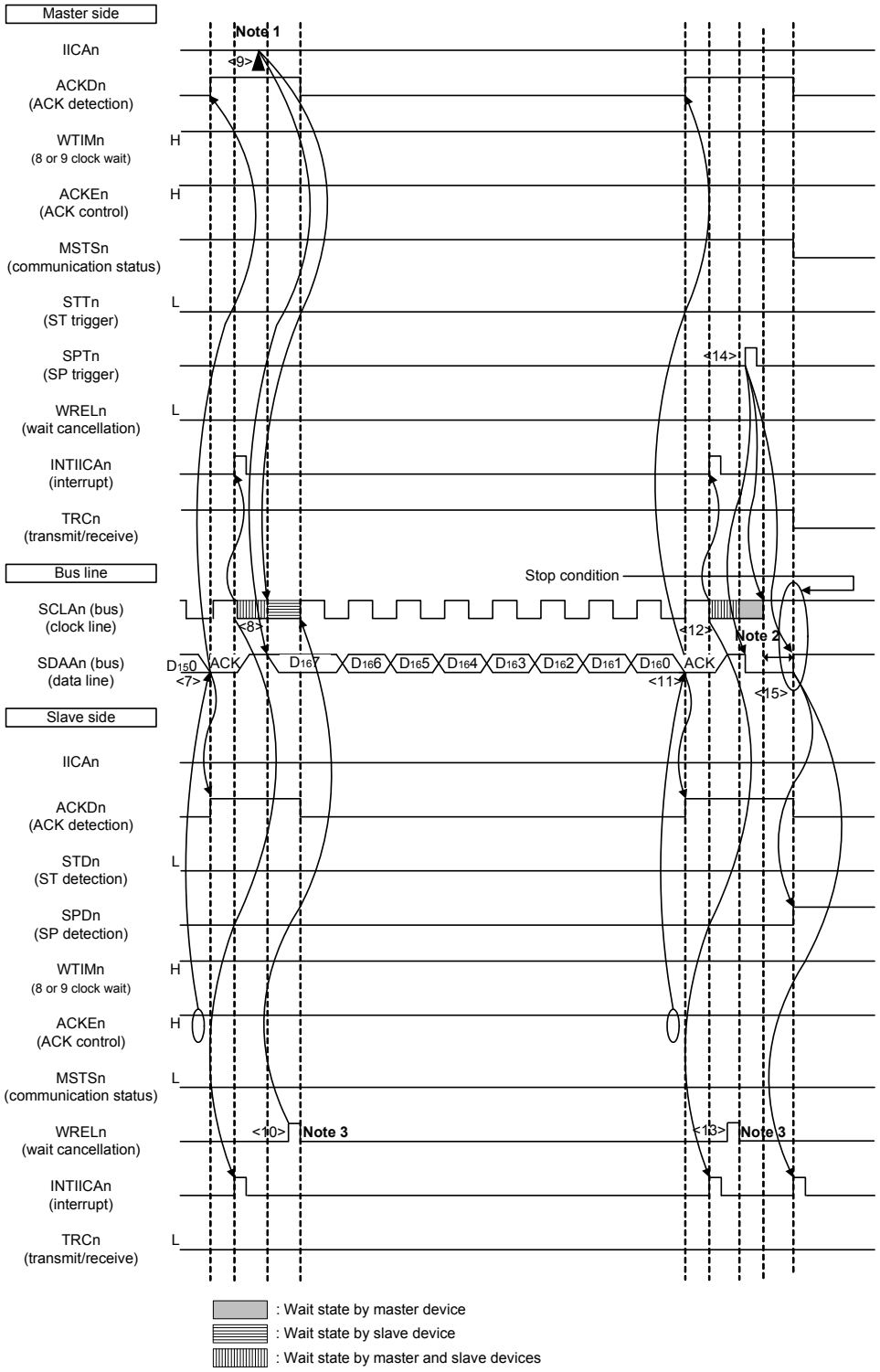
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in Figures 15 - 40 to 15 - 42 represent the entire procedure for communicating data using the I^2C bus. Figure 15 - 40 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15 - 41 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15 - 42 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0, 1

Figure 15 - 42 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, 1

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 15 - 42 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).

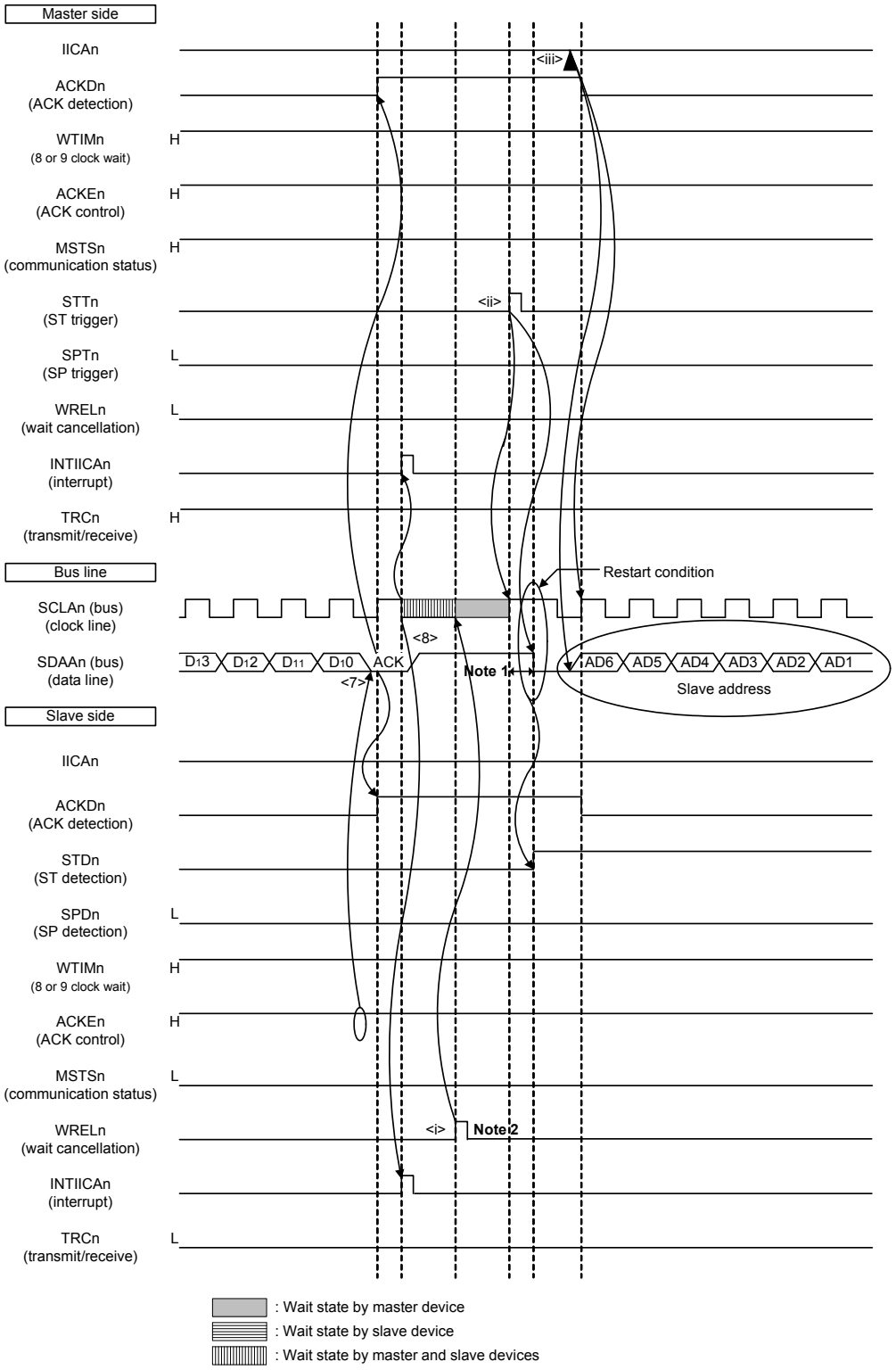
<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <15> in Figures 15 - 40 to 15 - 42 represent the entire procedure for communicating data using the I^2C bus. Figure 15 - 40 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15 - 41 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15 - 42 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0, 1

Figure 15 - 43 Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (4/4)

(4) 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, 1

The following describes the operations in Figure 15 - 43 (4) 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).

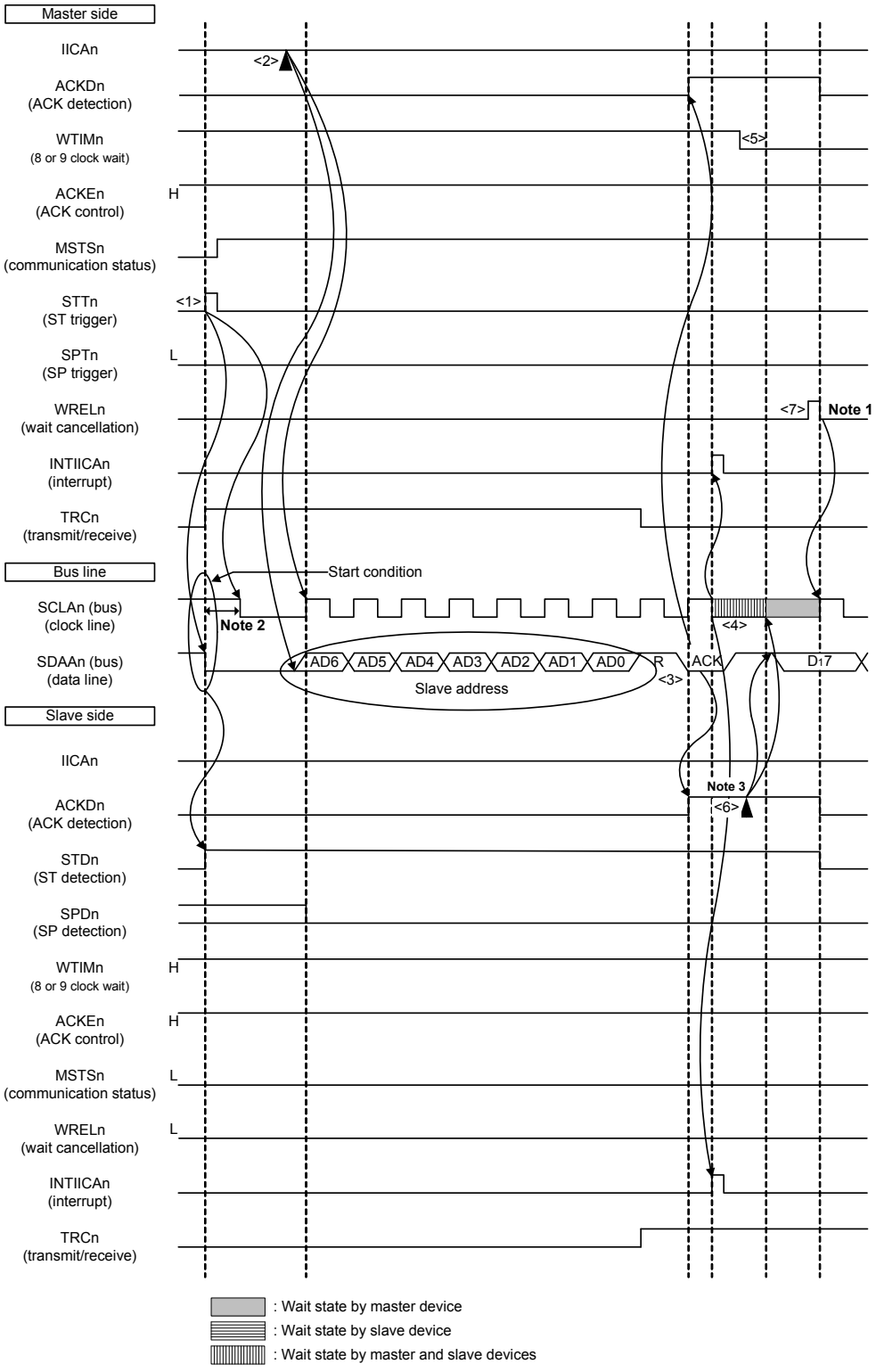
<ii> The start condition trigger is set again by the master device (STTn = 1) and a start condition (i.e. SCLAn =1 changes SDAAn from 1 to 0) is generated once the bus clock line goes high (SCLAn = 1) and the bus data line goes low (SDAAn = 0) after the restart condition setup time has elapsed. When 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, 1

Figure 15 - 44 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, 1

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 15 - 44 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.

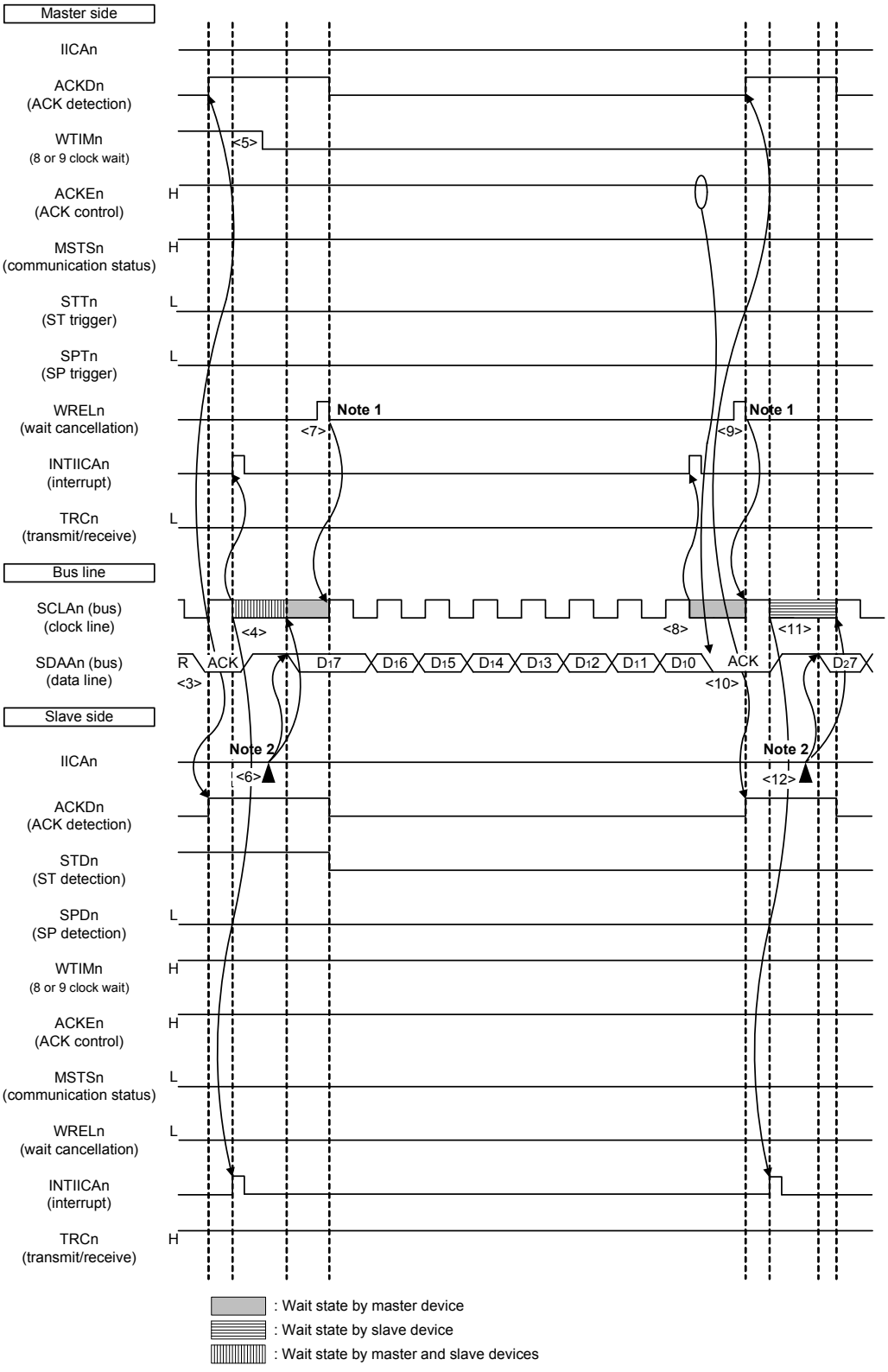
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in Figures 15 - 44 to 15 - 46 represent the entire procedure for communicating data using the I^2C bus. Figure 15 - 44 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15 - 45 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15 - 46 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0, 1

Figure 15 - 45 Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

(2) 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, 1

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 15 - 45 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.

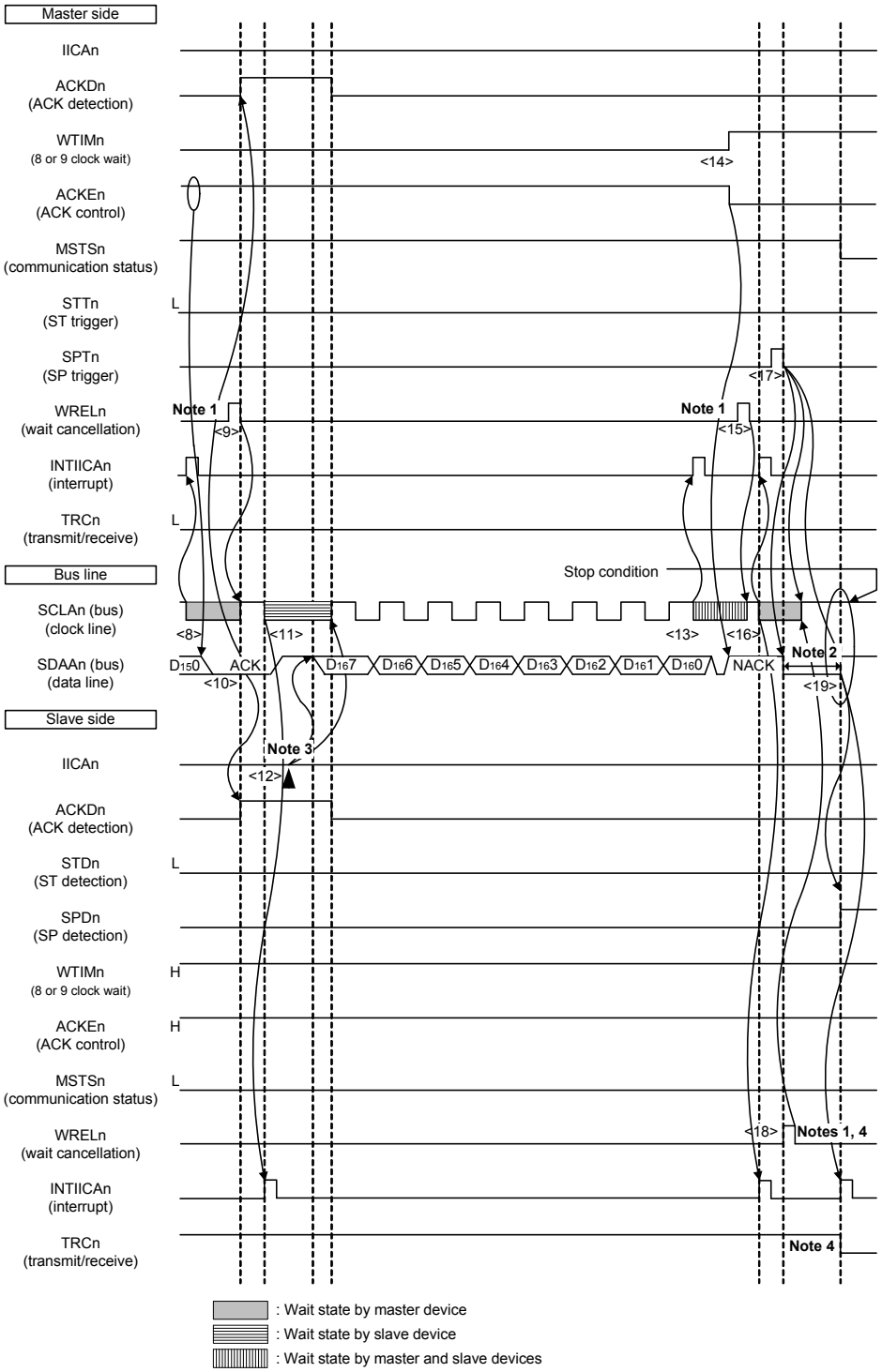
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in Figures 15 - 44 to 15 - 46 represent the entire procedure for communicating data using the I^2C bus. Figure 15 - 44 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15 - 45 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15 - 46 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0, 1

Figure 15 - 46 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, 1

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 15 - 46 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).

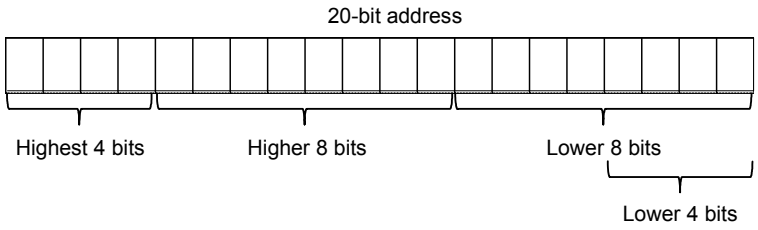
<19> Once the master device recognizes that the bus clock line is set (SCLAn = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAAn = 1) and issues a stop condition (i.e. SCLAn =1 changes SDAAn from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <19> in Figures 15 - 44 to 15 - 46 represent the entire procedure for communicating data using the I^2C bus. Figure 15 - 44 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15 - 45 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15 - 46 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0, 1

CHAPTER 16 DATA TRANSFER CONTROLLER (DTC)

The higher 8 bits for the address described in this chapter are the bits 15 to 8 of 20-bit address shown below.



If not specified especially, all the highest 4 bits of the address are 1 (FxxxxH).

16.1 Functions of DTC

The data transfer controller (DTC) is a function that transfers data between memories without using the CPU. The DTC is activated by a peripheral function interrupt to perform data transfers. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus.

Table 16 - 1 lists the DTC Specifications.

Table 16 - 1 DTC Specifications

Item		Specification
Activation sources		21 sources
Allocatable control data		24 sets
Address space which can be transferred	Address space	64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers
	Sources	Special function register (SFR), RAM area (excluding general-purpose registers), mirror area Note, data flash memory area Note, extended special function register (2nd SFR)
	Destinations	Special function register (SFR), RAM area (excluding general-purpose registers), extended special function register (2nd SFR)
Maximum number of transfers	Normal mode	256 times
	Repeat mode	255 times
Maximum size of block to be transferred	Normal mode (8-bit transfer)	256 bytes
	Normal mode (16-bit transfer)	512 bytes
	Repeat mode	255 bytes
Unit of transfers		8 bits/16 bits
Transfer mode	Normal mode	Transfers end on completion of the transfer causing the DTCCTj register value to change from 1 to 0.
	Repeat mode	On completion of the transfer causing the DTCCTj register value to change from 1 to 0, the repeat area address is initialized and the DTRLDj register value is reloaded to the DTCCTj register to continue transfers.
Address control	Normal mode	Fixed or incremented
	Repeat mode	Addresses of the area not selected as the repeat area are fixed or incremented.
Priority of activation sources		Refer to **Table 16 - 5 DTC Activation Sources and Vector Addresses**.

Table 16 - 1 DTC Specifications

Item		Specification
Interrupt request	Normal mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed, the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer.
	Repeat mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the transfer.
Transfer start		When bits DTCENi0 to DTCENi7 in the DTCENi registers are 1 (activation enabled), data transfer is started each time the corresponding DTC activation sources are generated.
Transfer stop	Normal mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed.
	Repeat mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled).

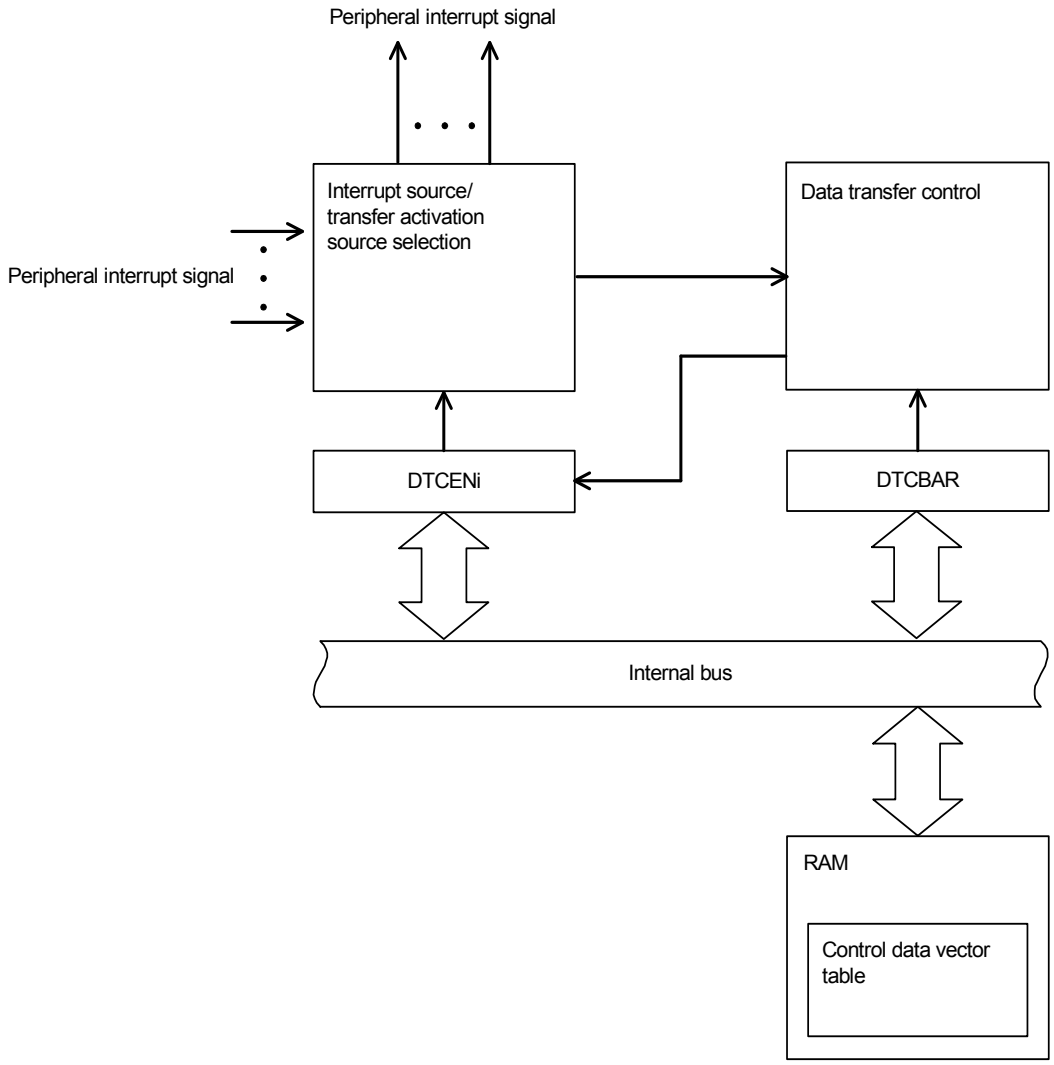
Note In the SNOOZE mode, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

Remark i = 0 to 4, j = 0 to 23

16.2 Configuration of DTC

Figure 16 - 1 shows the DTC Block Diagram.

Figure 16 - 1 DTC Block Diagram



16.3 Registers Controlling DTC

Table 16 - 2 lists the Registers Controlling DTC.

Table 16 - 2 Registers Controlling DTC

Register Name	Symbol
Peripheral enable register 1	PER1
DTC activation enable register 0	DTCEN0
DTC activation enable register 1	DTCEN1
DTC activation enable register 2	DTCEN2
DTC activation enable register 3	DTCEN3
DTC activation enable register 4	DTCEN4
DTC base address register	DTCBAR

Table 16 - 3 lists DTC Control Data.

DTC control data is allocated in the DTC control data area in RAM.

The DTCBAR register is used to set the 256-byte area, including the DTC control data area and the DTC vector table area where the start address for control data is stored.

Table 16 - 3 DTC Control Data

Register Name	Symbol
DTC Control Register j	DTCCRj
DTC Block Size Register j	DTBLSj
DTC Transfer Count Register j	DTCCTj
DTC Transfer Count Reload Register j	DTRLDj
DTC Source Address Register j	DTSARj
DTC Destination Address Register j	DTDARj

Remark j = 0 to 23

16.3.1 Allocation of DTC Control Data Area and DTC Vector Table Area

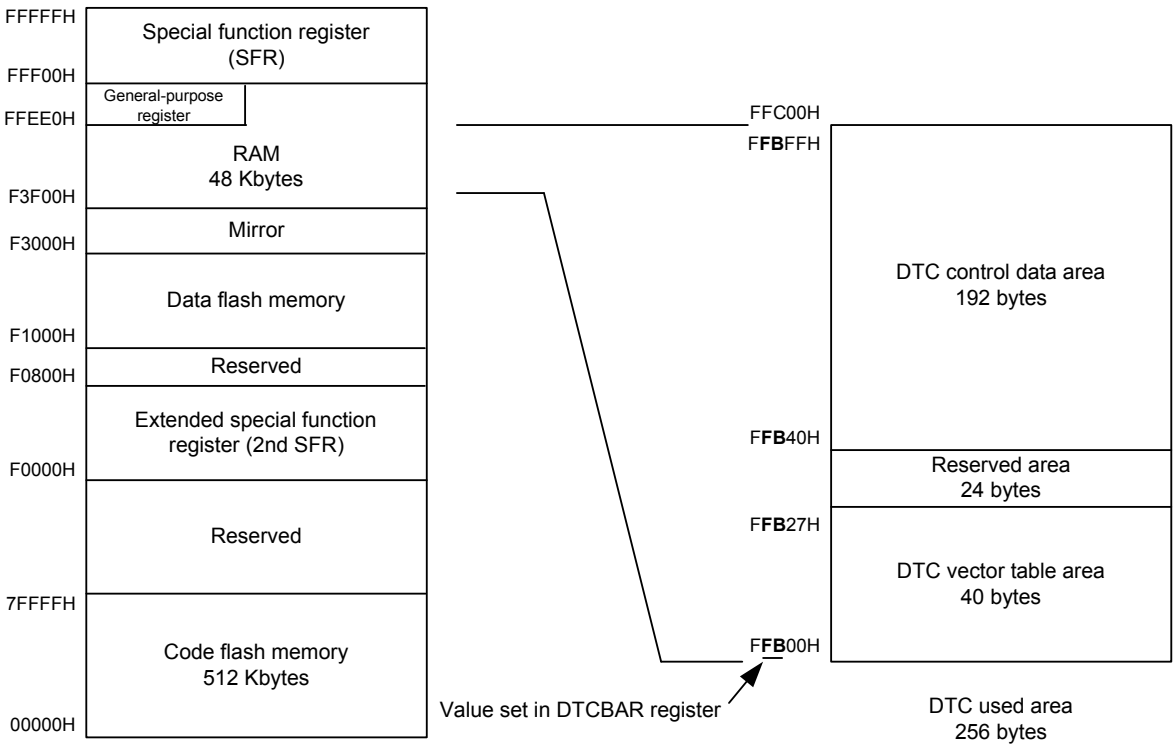
The DTCBAR register is used to set the 256-byte area where DTC control data and the vector table within the RAM area.

Figure 16 - 2 shows a Memory Map Example when DTCBAR Register is Set to FBH.

In the 192-byte DTC control data area, the space not used by the DTC can be used as RAM.

<R>

Figure 16 - 2 Memory Map Example when DTCBAR Register is Set to FBH



The areas where the DTC control data and vector table can be allocated differ depending on the product.

Caution 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.

Caution 2. Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.

Caution 3. The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the self-programming and data-flash functions.
R5F11FLL: F3F00H to F4309H

Caution 4. The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the on-chip debugging trace function.
R5F11FLL: F4300H to F46FFH

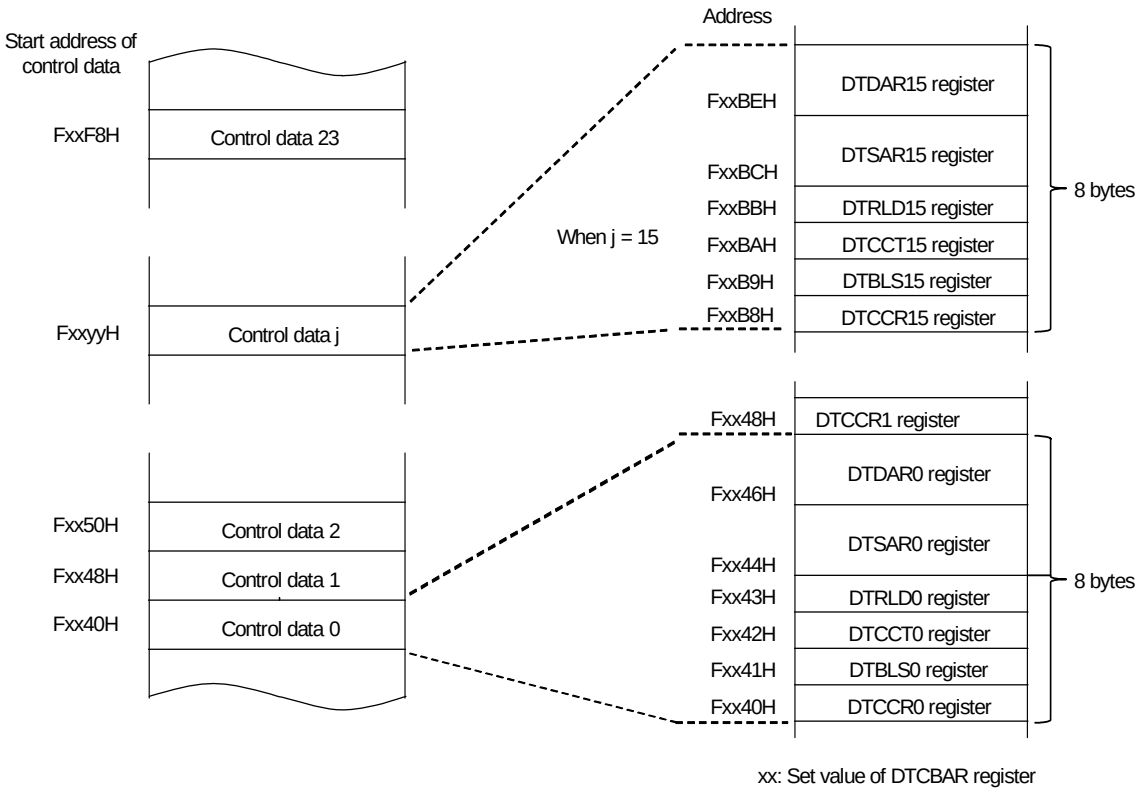
16.3.2 Control Data Allocation

Control data is allocated beginning with each start address in the order: Registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj (j = 0 to 23).
The higher 8 bits for start addresses 0 to 23 are set by the DTCBAR register, and the lower 8 bits are separately set according to the vector table assigned to each activation source.
Figure 16 - 3 shows Control Data Allocation.

Note 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register is set to 0 (activation disabled).

Note 2. Do not access DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj using a DTC transfer.

Figure 16 - 3 Control Data Allocation



<R>

Table 16 - 4 Start Address of Control Data

j	Address	j	Address
11	Fxx98H	23	FxxF8H
10	Fxx90H	22	FxxF0H
9	Fxx88H	21	FxxE8H
8	Fxx80H	20	FxxE0H
7	Fxx78H	19	FxxD8H
6	Fxx70H	18	FxxD0H
5	Fxx68H	17	FxxC8H
4	Fxx60H	16	FxxC0H
3	Fxx58H	15	FxxB8H
2	Fxx50H	14	FxxB0H
1	Fxx48H	13	FxxA8H
0	Fxx40H	12	FxxA0H

16.3.3 Vector Table

When the DTC is activated, one control data is selected according to the data read from the vector table which has been assigned to each activation source, and the selected control data is read from the DTC control data area.

Table 16 - 5 lists the DTC Activation Sources and Vector Addresses. A one byte of the vector table is assigned to each activation source, and data from 40H to F8H is stored in each area to select one of the 24 control data sets. The higher 8 bits for the vector address are set by the DTCBAR register, and 00H to 27H are allocated to the lower 8 bits corresponding to the activation source.

Change the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register is set to 0 (activation disabled).

<R>

Figure 16 - 4 Start Address of Control Data and Vector Table

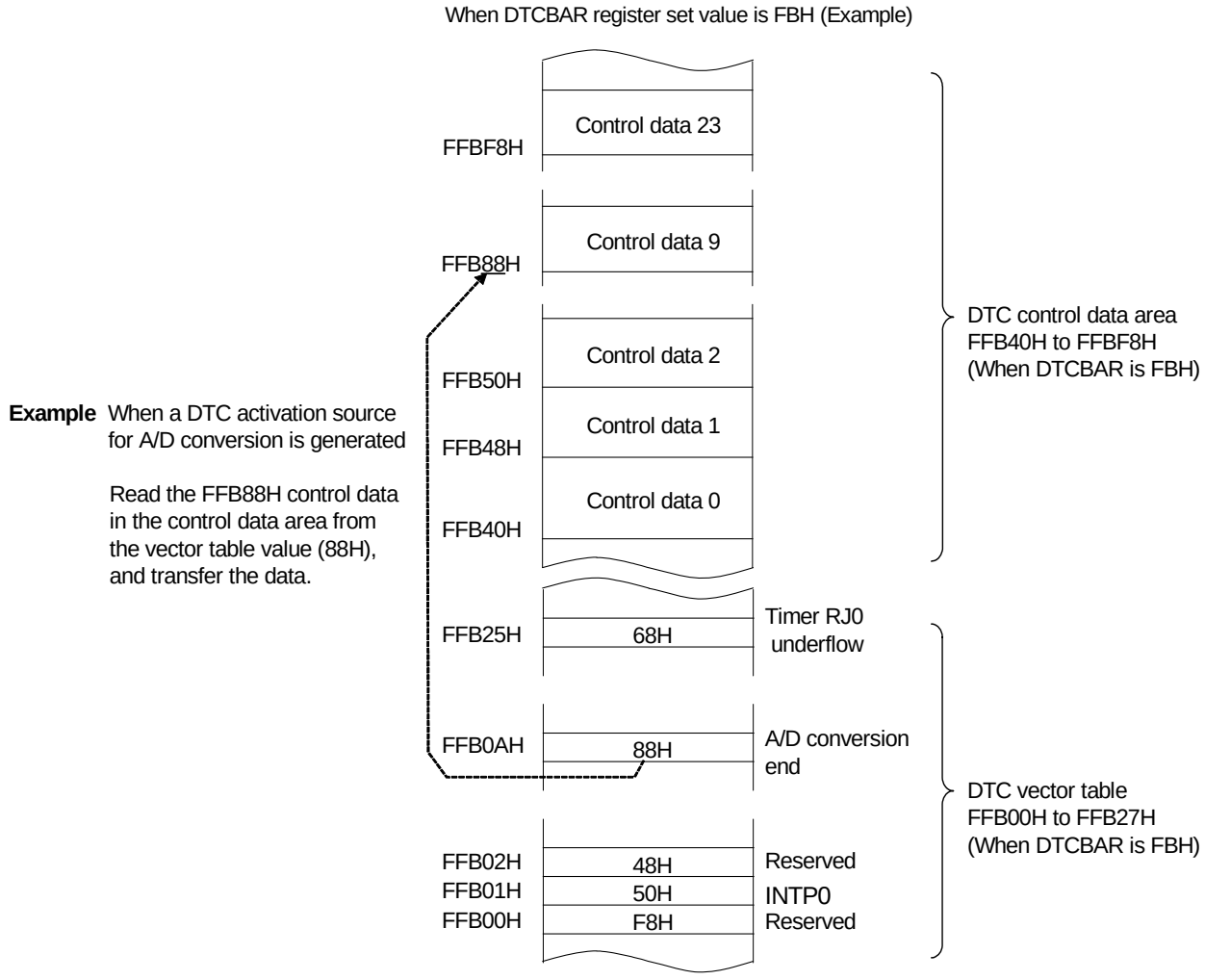


Table 16 - 5 DTC Activation Sources and Vector Addresses

DTC Activation Sources (Interrupt Request Source)	Source No.	Vector Address	Priority
Reserved	0	Address set in DTCBAR register +00H	Highest
INTP0	1	Address set in DTCBAR register +01H	
Reserved	2	Address set in DTCBAR register +02H	
Reserved	3	Address set in DTCBAR register +03H	
INTP3	4	Address set in DTCBAR register +04H	
INTP4	5	Address set in DTCBAR register +05H	
Reserved	6	Address set in DTCBAR register +06H	
INTP6	7	Address set in DTCBAR register +07H	
INTP7	8	Address set in DTCBAR register +08H	
Reserved	9	Address set in DTCBAR register +09H	
A/D conversion end	10	Address set in DTCBAR register +0AH	
Reserved	11	Address set in DTCBAR register +0BH	
Reserved	12	Address set in DTCBAR register +0CH	
UART1 reception transfer end	13	Address set in DTCBAR register +0DH	
UART1 transmission transfer end/CSI10 transfer end or buffer empty	14	Address set in DTCBAR register +0EH	
CSI21 transfer end or buffer empty	15	Address set in DTCBAR register +0FH	
CSI20 transfer end or buffer empty	16	Address set in DTCBAR register +10H	
UART3 reception transfer end	17	Address set in DTCBAR register +11H	
UART3 transmission transfer end/CSI30 transfer end or buffer empty/	18	Address set in DTCBAR register +12H	
End of channel 0 of timer array unit 0 count or capture	19	Address set in DTCBAR register +13H	
End of channel 1 of timer array unit 0 count or capture	20	Address set in DTCBAR register +14H	
End of channel 2 of timer array unit 0 count or capture	21	Address set in DTCBAR register +15H	
End of channel 3 of timer array unit 0 count or capture	22	Address set in DTCBAR register +16H	
End of channel 0 of timer array unit 1 count or capture	23	Address set in DTCBAR register +17H	
End of channel 1 of timer array unit 1 count or capture	24	Address set in DTCBAR register +18H	
End of channel 2 of timer array unit 1 count or capture	25	Address set in DTCBAR register +19H	
End of channel 3 of timer array unit 1 count or capture	26	Address set in DTCBAR register +1AH	
Reserved	27	Address set in DTCBAR register +1BH	
Reserved	28	Address set in DTCBAR register +1CH	
Reserved	29	Address set in DTCBAR register +1DH	
Reserved	30	Address set in DTCBAR register +1EH	
Reserved	31	Address set in DTCBAR register +1FH	
Reserved	32	Address set in DTCBAR register +20H	
Reserved	33	Address set in DTCBAR register +21H	
Reserved	34	Address set in DTCBAR register +22H	
Reserved	35	Address set in DTCBAR register +23H	
Reserved	36	Address set in DTCBAR register +24H	
Timer RJ0 underflow	37	Address set in DTCBAR register +25H	
Reserved	38	Address set in DTCBAR register +26H	
Reserved	39	Address set in DTCBAR register +27H	Lowest

16.3.4 Peripheral enable register 1 (PER1)

The PER1 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.
When using the DTC, be sure to set bit 3 (DTCEN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 16 - 5 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	4	<3>	2	1	<0>
PER1	0	0	0	0	DTCEN	0	0	TRJ0EN

DTCEN	Control of DTC input clock supply
0	Stops input clock supply. • DTC cannot run.
1	Enables input clock supply. • DTC can run.

16.3.5 DTC control register j (DTCCRj) (j = 0 to 23)

The DTCCRj register is used to control the DTC operating mode.

Figure 16 - 6 Format of DTC control register j (DTCCRj)

Address: Refer to **16.3.2 Control Data Allocation**. After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DTCCRj	0	SZ	RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

SZ	Transfer Data size selection
0	8 bits
1	16 bits

RPTINT	Enabling/disabling repeat mode interrupts
0	Interrupt generation disabled
1	Interrupt generation enabled
The setting of the RPTINT bit is invalid when the MODE bit is 0 (normal mode).	

CHNE	Enabling/disabling chain transfers
0	Chain transfers disabled
1	Chain transfers enabled
Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).	

DAMOD	Transfer destination address control
0	Fixed
1	Incremented
The setting of the DAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 0 (transfer destination is the repeat area).	

SAMOD	Transfer source address control
0	Fixed
1	Incremented
The setting of the SAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 1 (transfer source is the repeat area).	

RPTSEL	Repeat area selection
0	Transfer destination is the repeat area
1	Transfer source is the repeat area
The setting of the RPTSEL bit is invalid when the MODE bit is 0 (normal mode).	

MODE	Transfer mode selection
0	Normal mode
1	Repeat mode

Caution Do not access the DTCCRj register using a DTC transfer.

16.3.6 DTC block size register j (DTBLSj) (j = 0 to 23)

This register is used to set the block size of the data to be transferred by one activation.

Figure 16 - 7 Format of DTC block size register j (DTBLSj)

Address: Refer to **16.3.2 Control Data Allocation**. After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DTBLSj	DTBLSj7	DTBLSj6	DTBLSj5	DTBLSj4	DTBLSj3	DTBLSj2	DTBLSj1	DTBLSj0

DTBLSj	Transfer Block Size	
	8-Bit Transfer	16-Bit Transfer
00H	256 bytes	512 bytes
01H	1 byte	2 bytes
02H	2 bytes	4 bytes
03H	3 bytes	6 bytes
• • •	• • •	• • •
FDH	253 bytes	506 bytes
FEH	254 bytes	508 bytes
FFH	255 bytes	510 bytes

Caution Do not access the DTBLSj register using a DTC transfer.

16.3.7 DTC transfer count register j (DTCCTj) (j = 0 to 23)

This register is used to set the number of DTC data transfers. The value is decremented by 1 each time DTC transfer is activated once.

Figure 16 - 8 Format of DTC transfer count register j (DTCCTj)

Address: Refer to **16.3.2 Control Data Allocation**. After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DTCCTj	DTCCTj7	DTCCTj6	DTCCTj5	DTCCTj4	DTCCTj3	DTCCTj2	DTCCTj1	DTCCTj0

DTCCTj	Number of Transfers
00H	256 times
01H	Once
02H	2 times
03H	3 times
• • •	• • •
FDH	253 times
FEH	254 times
FFH	255 times

Caution Do not access the DTCCTj register using a DTC transfer.

16.3.8 DTC transfer count reload register j (DTRLDj) (j = 0 to 23)

This register is used to set the initial value of the transfer count register in repeat mode. Since the value of this register is reloaded to the DTCCT register in repeat mode, set the same value as the initial value of the DTCCT register.

Figure 16 - 9 Format of DTC transfer count reload register j (DTRLDj)

Address: Refer to **16.3.2 Control Data Allocation**. After reset: Undefined R/W

	7	6	5	4	3	2	1	0
DTRLDj	DTRLDj7	DTRLDj6	DTRLDj5	DTRLDj4	DTRLDj3	DTRLDj2	DTRLDj1	DTRLDj0

Caution Do not access the DTRLDj register using a DTC transfer.

16.3.9 DTC source address register j (DTSARj) (j = 0 to 23)

This register is used to specify the transfer source address for data transfer.

When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest bit is ignored and the address is handled as an even address.

Figure 16 - 10 Format of DTC source address register j (DTSARj)

Address: Refer to **16.3.2 Control Data Allocation**. After reset: Undefined R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTSARj	DTS ARj15	DTS ARj14	DTS ARj13	DTS ARj12	DTS ARj11	DTS ARj10	DTS ARj9	DTS ARj8	DTS ARj7	DTS ARj6	DTS ARj5	DTS ARj4	DTS ARj3	DTS ARj2	DTS ARj1	DTS ARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.
Caution 2. Do not access the DTSARj register using a DTC transfer.

16.3.10 DTC destination address register j (DTDARj) (j = 0 to 23)

This register is used to specify the transfer destination address for data transfer.

When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest bit is ignored and the address is handled as an even address.

Figure 16 - 11 Format of DTC destination address register j (DTDARj)

Address: Refer to **16.3.2 Control Data Allocation**. After reset: Undefined R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTDARj	DTD ARj15	DTD ARj14	DTD ARj13	DTD ARj12	DTD ARj11	DTD ARj10	DTD ARj9	DTD ARj8	DTD ARj7	DTD ARj6	DTD ARj5	DTD ARj4	DTD ARj3	DTD ARj2	DTD ARj1	DTD ARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.
Caution 2. Do not access the DTDARj register using a DTC transfer.

16.3.11 DTC activation enable register i (DTCENi) (i = 0 to 4)

This is an 8-bit register which enables or disables DTC activation by interrupt sources. Table 16 - 6 lists the Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7.

The DTCENi register can be set by an 8-bit memory manipulation instruction and a 1-bit memory manipulation instruction.

Caution 1. Modify bits DTCENi0 to DTCENi7 if an activation source corresponding to the bit has not been generated.

Caution 2. Do not access the DTCENi register using a DTC transfer.

Caution 3. The assigned functions differ depending on the product. For the bits to which no function is assigned, be sure to set their values to 0.

Figure 16 - 12 Format of DTC activation enable register i (DTCENi) (i = 0 to 4)

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2), F02EBH (DTCEN3), F02ECH (DTCEN4) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DTCENi	DTCENi7	DTCENi6	DTCENi5	DTCENi4	DTCENi3	DTCENi2	DTCENi1	DTCENi0

DTCENi7	DTC activation enable i7
0	Activation disabled
1	Activation enabled
The DTCENi7 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi6	DTC activation enable i6
0	Activation disabled
1	Activation enabled
The DTCENi6 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi5	DTC activation enable i5
0	Activation disabled
1	Activation enabled
The DTCENi5 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi4	DTC activation enable i4
0	Activation disabled
1	Activation enabled
The DTCENi4 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi3	DTC activation enable i3
0	Activation disabled
1	Activation enabled
The DTCENi3 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi2	DTC activation enable i2
0	Activation disabled
1	Activation enabled
The DTCENi2 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi1	DTC activation enable i1
0	Activation disabled
1	Activation enabled
The DTCENi1 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi0	DTC activation enable i0
0	Activation disabled
1	Activation enabled
The DTCENi0 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

Table 16 - 6 Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7

Register	DTCENi7 Bit	DTCENi6 Bit	DTCENi5 Bit	DTCENi4 Bit	DTCENi3 Bit	DTCENi2 Bit	DTCENi1 Bit	DTCENi0 Bit
DTCEN0	Reserved	INTP0	Reserved	Reserved	INTP3	INTP4	Reserved	INTP6
DTCEN1	INTP7	Reserved	A/D conversion end	Reserved	Reserved	UART1 reception transfer end	UART1 transmission transfer end/CSI10 transfer end or buffer empty	CSI21 transfer end or buffer empty
DTCEN2	CSI20 transfer end or buffer empty	UART3 reception transfer end	UART3 transmission transfer end/CSI30 transfer end or buffer empty	End of channel 0 of timer array unit 0 count or capture	End of channel 1 of timer array unit 0 count or capture	End of channel 2 of timer array unit 0 count or capture	End of channel 3 of timer array unit 0 count or capture	End of channel 0 of timer array unit 1 count or capture
DTCEN3	End of channel 1 of timer array unit 1 count or capture	End of channel 2 of timer array unit 1 count or capture	End of channel 3 of timer array unit 1 count or capture	Reserved	Reserved	Reserved	Reserved	Reserved
DTCEN4	Reserved	Reserved	Reserved	Reserved	Reserved	Timer RJ0 underflow	Reserved	Reserved

Caution For the bits to which no function is assigned, be sure to set their values to 0.

Remark i = 0 to 4

16.3.12 DTC base address register (DTCBAR)

This is an 8-bit register used to set the following addresses: the vector address where the start address of the DTC control data area is stored and the address of the DTC control data area. The value of the DTCBAR register is handled as the higher 8 bits to generate a 16-bit address.

Caution 1. Change the DTCBAR register value with all DTC activation sources set to activation disabled.
Caution 2. Do not rewrite the DTCBAR register more than once.
Caution 3. Do not access the DTCBAR register using a DTC transfer.
Caution 4. For the allocation of the DTC control data area and the DTC vector table area, refer to the notes on 16.3.1 Allocation of DTC Control Data Area and DTC Vector Table Area.

Figure 16 - 13 Format of DTC base address register (DTCBAR)

Address: F02E0H After reset: FDH R/W

Symbol	7	6	5	4	3	2	1	0
DTCBAR	DTCBAR7	DTCBAR6	DTCBAR5	DTCBAR4	DTCBAR3	DTCBAR2	DTCBAR1	DTCBAR0

16.4 DTC Operation

When the DTC is activated, control data is read from the DTC control data area to perform data transfers and control data after data transfer is written back to the DTC control data area. Twenty-four sets of control data can be stored in the DTC control data area, which allows 24 types of data transfers to be performed.
There are two transfer modes (normal mode and repeat mode) and two transfer sizes (8-bit transfer and 16-bit transfer). When the CHNE bit in the DTCCRj (j = 0 to 23) register is set to 1 (chain transfers enabled), multiple control data is read and data transfers are continuously performed by one activation source (chain transfers).
A transfer source address is specified by the 16-bit register DTSARj, and a transfer destination address is specified by the 16-bit register DTDARj.
The values in registers DTSARj and DTDARj are separately incremented or fixed according to the control data after the data transfer.

16.4.1 Activation Sources

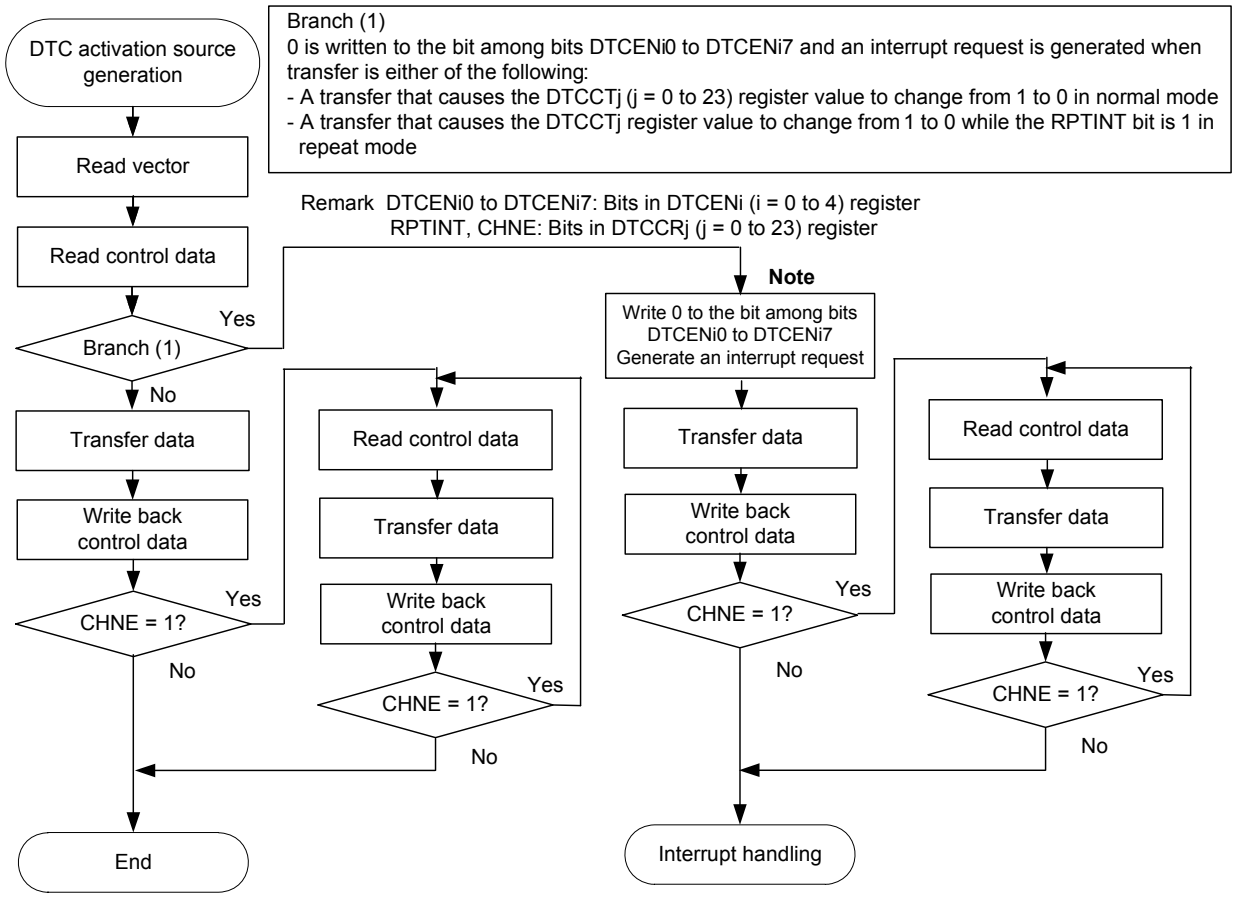
The DTC is activated by an interrupt signal from the peripheral functions. The interrupt signals to activate the DTC are selected with the DTCENi (i = 0 to 4) register.

The DTC sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi register to 0 (activation disabled) during operation when the setting of data transfer (the first transfer in chain transfers) is either of the following:

- A transfer that causes the DTCCTj (j = 0 to 23) register value to change to 0 in normal mode
- A transfer that causes the DTCCTj register value to change to 0 while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled) in repeat mode

Figure 16 - 14 shows the DTC Internal Operation Flowchart.

Figure 16 - 14 DTC Internal Operation Flowchart



Note 0 is not written to the bit among bits DTCENi0 to DTCENi7 for data transfers activated by the setting to enable chain transfers (the CHNE bit is 1). Also, no interrupt request is generated.

16.4.2 Normal Mode

One to 256 bytes of data are transferred by one activation during 8-bit transfer and 2 to 512 bytes during 16-bit transfer. The number of transfers can be 1 to 256 times. When the data transfer causing the DTCCTj (j = 0 to 23) register value to change to 0 is performed, the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register to 0 (activation disabled).

Table 16 - 7 shows Register Functions in Normal Mode. Figure 16 - 15 shows Data Transfers in Normal Mode.

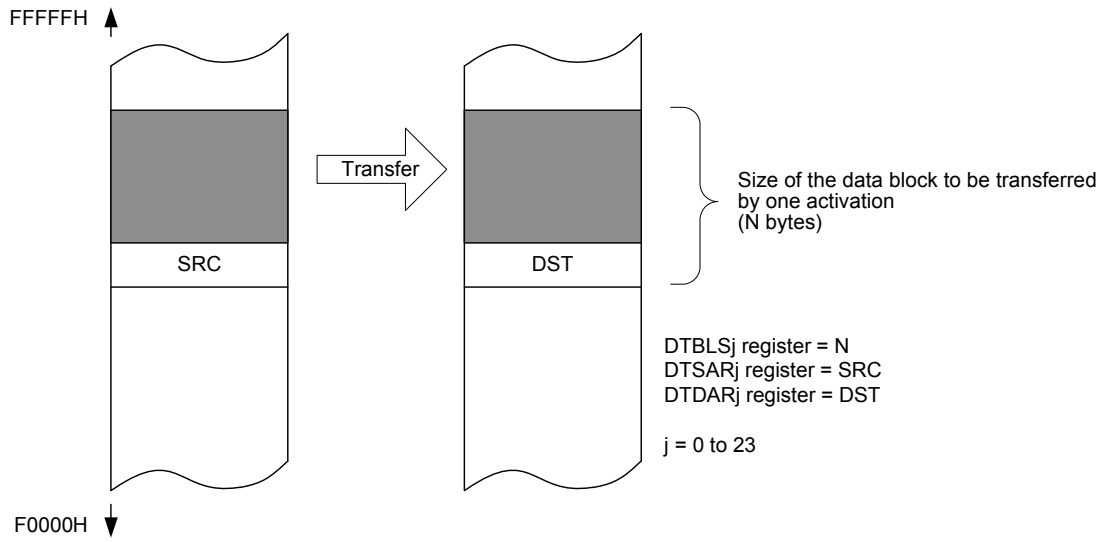
Table 16 - 7 Register Functions in Normal Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	Not used **Note**
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Note Initialize this register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

Remark j = 0 to 23

Figure 16 - 15 Data Transfers in Normal Mode



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	Fixed	Fixed	SRC	DST
0	1	X	0	Incremented	Fixed	SRC + N	DST
1	0	X	0	Fixed	Incremented	SRC	DST + N
1	1	X	0	Incremented	Incremented	SRC + N	DST + N

X: 0 or 1

16.4.3 Repeat Mode

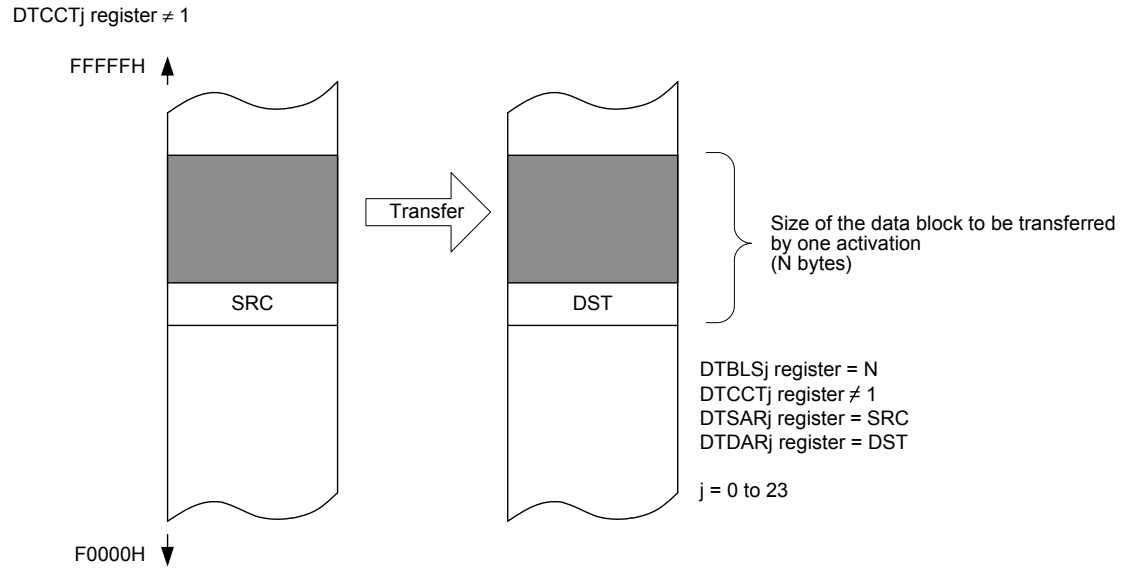
One to 255 bytes of data are transferred by one activation. Either of the transfer source or destination should be specified as the repeat area. The number of transfers can be 1 to 255 times. On completion of the specified number of transfers, the DTCCTj (i = 0 to 23) register and the address specified for the repeat area are initialized to continue transfers. When the data transfer causing the DTCCTj register value to change to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 to 0 (activation disabled). When the RPTINT bit in the DTCCRj register is 0 (interrupt generation disabled), no interrupt request is generated even if the data transfer causing the DTCCTj register value to change to 0 is performed. Also, bits DTCENi0 to DTCENi7 are not set to 0. Table 16 - 8 lists Register Functions in Repeat Mode. Figure 16 - 16 shows Data Transfers in Repeat Mode.

Table 16 - 8 Register Functions in Repeat Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	This register value is reloaded to the DTCCT register (the number of transfers is initialized).
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

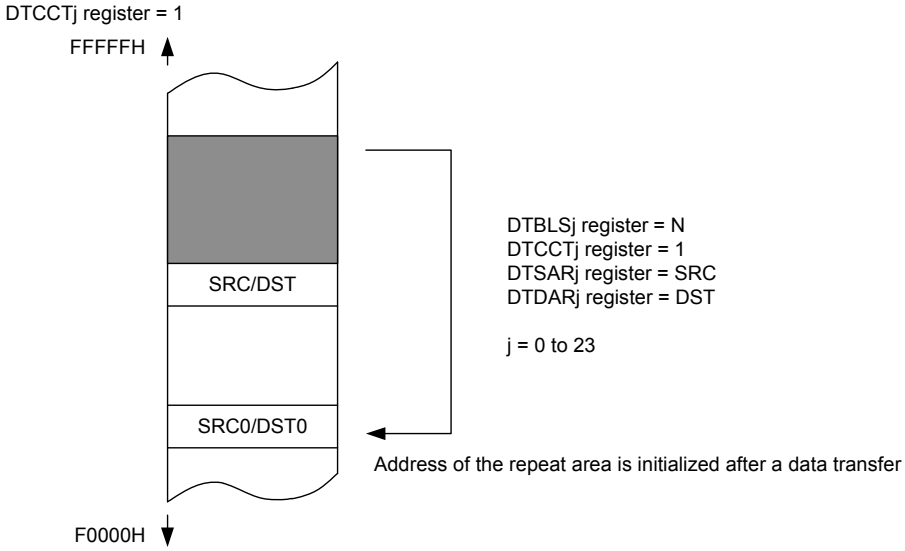
Remark j = 0 to 23

Figure 16 - 16 Data Transfers in Repeat Mode



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	Repeat area	Fixed	SRC + N	DST
1	X	1	1	Repeat area	Incremented	SRC + N	DST + N
X	0	0	1	Fixed	Repeat area	SRC	DST + N
X	1	0	1	Incremented	Repeat area	SRC + N	DST + N

X: 0 or 1



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	Repeat area	Fixed	SRC0	DST
1	X	1	1	Repeat area	Incremented	SRC0	DST + N
X	0	0	1	Fixed	Repeat area	SRC	DST0
X	1	0	1	Incremented	Repeat area	SRC + N	DST0

SRC0: Initial source address value
DST0: Initial destination address value
X: 0 or 1

Caution 1. When repeat mode is used, the lower 8 bits of the initial value for the repeat area address must be 00H.

Caution 2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.

16.4.4 Chain Transfers

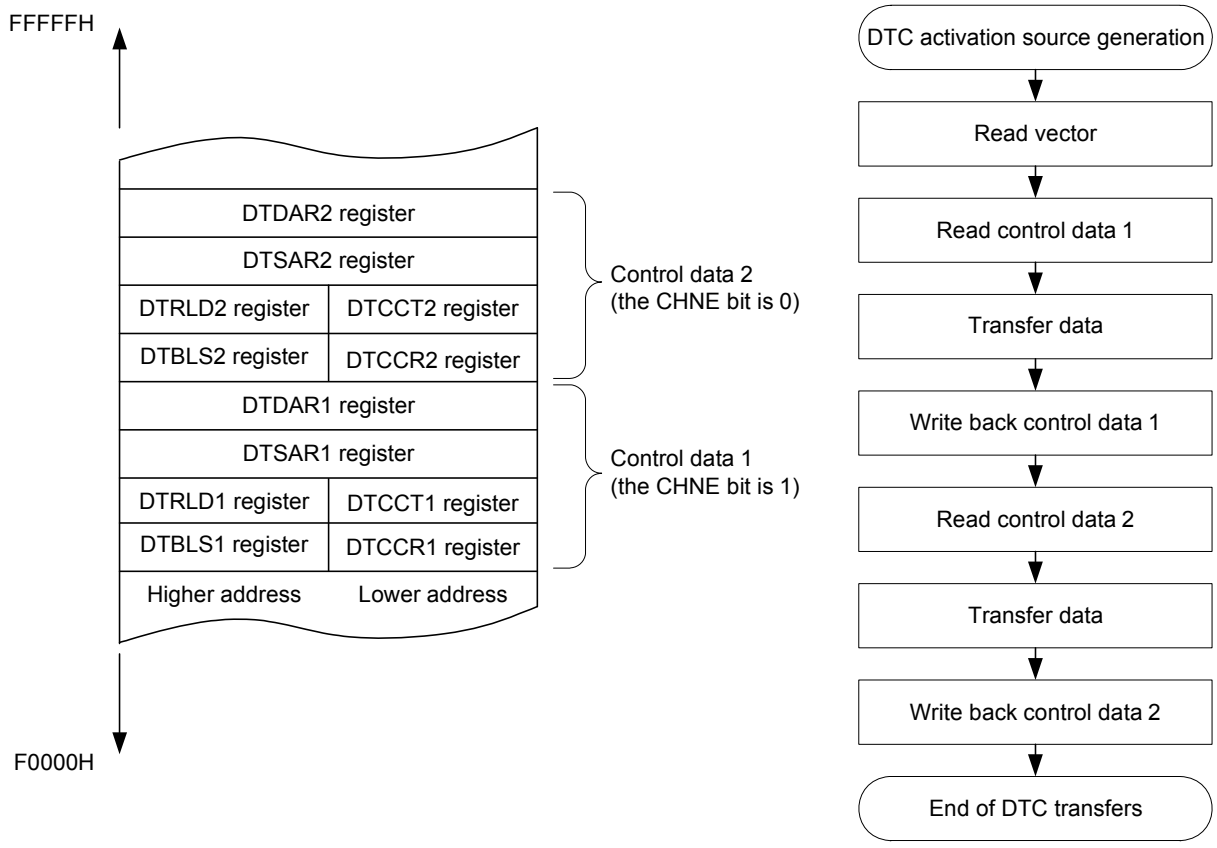
When the CHNE bit in the DTCCRj (j = 0 to 22) register is 1 (chain transfers enabled), multiple data transfers can be continuously performed by one activation source.

When the DTC is activated, one control data is selected according to the data read from the vector address corresponding to the activation source, and the selected control data is read from the DTC control data area. When the CHNE bit for the control data is 1 (chain transfers enabled), the next control data immediately following the current control data is read and transferred after the current transfer is completed. This operation is repeated until the data transfer with the control data for which the CHNE bit is 0 (chain transfers disabled) is completed.

When chain transfers are performed using multiple control data, the number of transfers set for the first control data is enabled, and the number of transfers set for the second and subsequent control data to be processed will be invalid.

Figure 16 - 17 shows Data Transfers during Chain Transfers.

Figure 16 - 17 Data Transfers during Chain Transfers



Caution 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).

Caution 2. During chain transfers, bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register are not set to 0 (activation disabled) for the second and subsequent transfers. Also, no interrupt request is generated.

16.5 Cautions for DTC

16.5.1 Setting DTC Control Data and Vector Table

- Do not access the DTC extended special function register (2nd SFR), the DTC control data area, the DTC vector table area, or the general-register (FFEE0H to FFEFFH) space using a DTC transfer.
- Modify the DTC base address register (DTCBAR) while all DTC activation sources are set to activation disabled.
- Do not rewrite the DTC base address register (DTCBAR) twice or more.
- Modify the data of the DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj register when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 4) register is 0 (activation disabled).
- Modify the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 4) register is 0 (activation disabled).
- Do not allocate RAM addresses which are used as a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

16.5.2 Allocation of DTC Control Data Area and DTC Vector Table Area

The areas where the DTC control data and vector table can be allocated differ, depending on the usage conditions.

- It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.
- Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.
- The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the self-programming and data-flash functions.
  R5F11FLL: F3F00H to F4309H
- The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the on-chip debugging trace function.
  R5F11FLL: F4300H to F46FFH
- Initialize the DTRLD register to 00H even in normal mode when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

16.5.3 DTC Pending Instruction

Even if a DTC transfer request is generated, DTC transfer is held pending immediately after the following instructions. Also, the DTC is not activated between PREFIX instruction code and the instruction immediately after that code.

- Call/return instruction
- Unconditional branch instruction
- Conditional branch instruction
- Read access instruction for code flash memory
- Bit manipulation instructions for IFxx, MKxx, PRxx, and PSW, and an 8-bit manipulation instruction that has the ES register as operand
- Instruction for accessing the data flash memory

Caution 1. When a DTC transfer request is acknowledged, all interrupt requests are held pending until DTC transfer is completed.

Caution 2. While the DTC is held pending by the DTC pending instruction, all interrupt requests are held pending.

16.5.4 Operation when Accessing Data Flash Memory Space

When accessing the data flash space after an instruction execution from the start of DTC data transfer, a wait of three clock cycles will be inserted to the next instruction.

```
Instruction 1
DTC data transfer
Instruction 2 ←The wait of three clock cycles occurs.
MOV A,          ! Data Flash space
```

16.5.5 Number of DTC Execution Clock Cycles

Table 16 - 9 lists the Operations Following DTC Activation and Required Number of Cycles for each operation.

Table 16 - 9 Operations Following DTC Activation and Required Number of Cycles

Vector Read	Control Data		Data Read	Data Write
	Read	Write-back		
1	4	**Note 1**	**Note 2**	**Note 2**

Note 1. For the number of clock cycles required for control data write-back, refer to **Table 16 - 10 Number of Clock Cycles Required for Control Data Write-Back Operation**.

Note 2. For the number of clock cycles required for data read/write, refer to **Table 16 - 11 Number of Clock Cycles Required for One Data Read/Write Operation**.

Table 16 - 10 Number of Clock Cycles Required for Control Data Write-Back Operation

DTCCR Register Setting				Address Setting		Control Register to be Written Back				Number of Clock Cycles
DAMOD	SAMOD	RPTSEL	MODE	Source	Destination	DTCCTj Register	DTRLDj Register	DTSARj Register	DTDARj Register	
0	0	X	0	Fixed	Fixed	Written back	Written back	Not written back	Not written back	1
0	1	X	0	Incremented	Fixed	Written back	Written back	Written back	Not written back	2
1	0	X	0	Fixed	Incremented	Written back	Written back	Not written back	Written back	2
1	1	X	0	Incremented	Incremented	Written back	Written back	Written back	Written back	3
0	X	1	1	Repeat area	Fixed	Written back	Written back	Written back	Not written back	2
1	X	1	1		Incremented	Written back	Written back	Written back	Written back	3
X	0	0	1	Fixed	Repeat area	Written back	Written back	Not written back	Written back	2
X	1	0	1	Incremented		Written back	Written back	Written back	Written back	3

Remark j = 0 to 23; X: 0 or 1

Table 16 - 11 Number of Clock Cycles Required for One Data Read/Write Operation

Operation	RAM	Code Flash Memory	Data Flash Memory	Special function register (SFR)	Extended special function register (2nd SFR)	
					No Wait State	Wait States
Data read	1	2	4	1	1	1 + number of wait states **Note**
Data write	1	—	—	1	1	1 + number of wait states **Note**

Note The number of wait states differs depending on the specifications of the register allocated to the extended special function register (2nd SFR) to be accessed.

16.5.6 DTC Response Time

Table 16 - 12 lists the DTC Response Time. The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts. It does not include the number of DTC execution clocks.

Table 16 - 12 DTC Response Time

	Minimum Time	Maximum Time
Response Time	3 clocks	19 clocks

Note that the response from the DTC may be further delayed under the following cases. The number of delayed clock cycles differs depending on the conditions.

- When executing an instruction from the internal RAM
  Maximum response time: 20 clocks
- When executing a DTC pending instruction (refer to **16.5.3 DTC Pending Instruction**)
- Maximum response time: Maximum response time for each condition + execution clock cycles for the instruction to be held pending under the condition.
- When accessing the TRJ0 register that a wait occurs
  Maximum response time: Maximum response time for each condition + 1 clock

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU/peripheral hardware clock)

16.5.7 DTC Activation Sources

- After inputting a DTC activation source, do not input the same activation source again until DTC transfer is completed.
- While a DTC activation source is generated, do not manipulate the DTC activation enable bit corresponding to the source.
- If DTC activation sources conflict, their priority levels are determined in order to select the source for activation when the CPU acknowledges the DTC transfer. For details on the priority levels of activation sources, refer to **16.3.3 Vector Table**.

16.5.8 Operation in Standby Mode Status

Status	DTC Operation
HALT mode	Operable (Operation is disabled while in the low power consumption RTC mode)
STOP mode	DTC activation sources can be accepted **Note 2**
SNOOZE mode	Operable **Notes 1, 3**

Note 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected as f_{CLK}.

Note 2. In the STOP mode, detecting a DTC activation source enables transition to SNOOZE mode and DTC transfer. After completion of transfer, the system returns to the STOP mode. However, since the code flash memory and the data flash memory are stopped during the SNOOZE mode, the flash memory cannot be set as the transfer source.

Note 3. When an A/D conversion end interrupt is set as a DTC activation source from the A/D converter SNOOZE mode function, release the SNOOZE mode using the A/D conversion end interrupt to start CPU processing after completion of DTC transfer, or use a chain transfer to set the A/D converter SNOOZE mode function again (writing 0 to the AWC bit and then writing 1 to the AWC bit).

CHAPTER 17 EVENT LINK CONTROLLER (ELC)

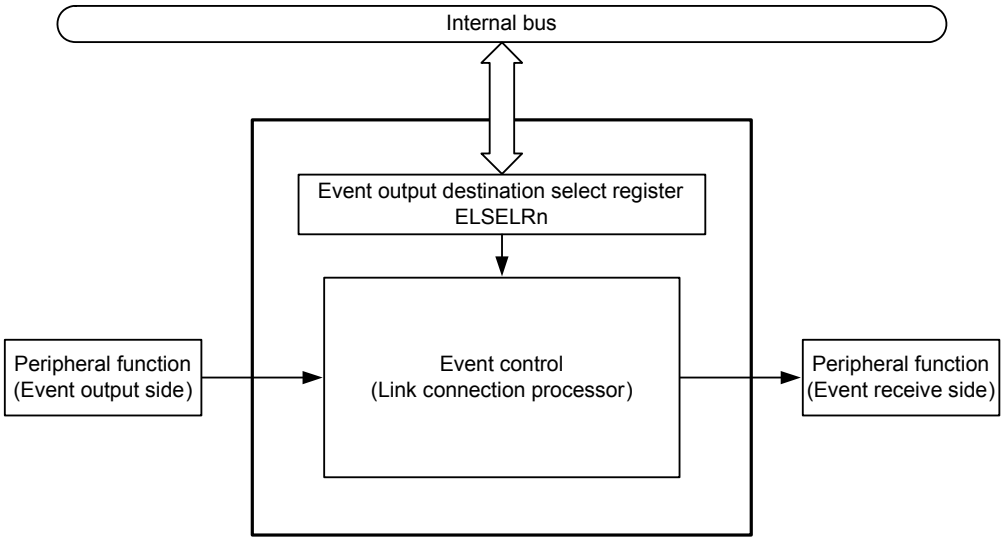
17.1 Functions of ELC

The event link controller (ELC) mutually connects (links) events output from each peripheral function. By linking events, it becomes possible to coordinate operation between peripheral functions directly without going through the CPU.

17.2 Configuration of ELC

Figure 17 - 1 shows the ELC Block Diagram.

Figure 17 - 1 ELC Block Diagram



17.3 Registers Controlling ELC

17.3.1 Event output destination select register n (ELSELRn) (n = 00, 03, 04, 07, 13, 16 to 23)

An ELSELRn register links each event signal to an operation of an event-receiving peripheral function (link destination peripheral function) after reception.

Do not set multiple event inputs to the same event output destination (event receive side). The operation of the event-receiving peripheral function will become undefined, and event signals may not be received correctly. In addition, do not set the event link generation source and the event link output destination to the same function.

Set an ELSELRn register during a period when no event output peripheral functions are generating event signals.

Table 17 - 1 lists the correspondence between ELSELRn registers and peripheral functions.

Figure 17 - 2 Format of Event output destination select register n (ELSELRn)

Address: F0300H (ELSELR00) to F0317H (ELSELR23) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	0	ELSELn2	ELSELn1	ELSELn0

ELSELn2	ELSELn1	ELSELn0	Event Link Selection
0	0	0	Event link disabled
0	0	1	Hardware trigger signal of A/D converter
0	1	0	Timer input of timer array unit 0 channel 0 **Note 1**
0	1	1	Timer input of timer array unit 0 channel 1 **Note 2**
1	0	0	Count source of timer RJ
Other than above			Setting prohibited

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

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

Table 17 - 1 Correspondence Between ELSELRn Registers (n = 00 to 25) and Peripheral Functions

Register Name	Event Generator (Output Origin of Event Input n)	Event Description
ELSELR00	External interrupt edge detection 0	INTP0
ELSELR03	External interrupt edge detection 3	INTP3
ELSELR04	External interrupt edge detection 4	INTP4
ELSELR07	RTC fixed-cycle signal/Alarm match detection	INTRTC
ELSELR13	Timer RJ0 underflow	INTTRJ0
ELSELR16	TAU channel 00 count end	INTTM00
ELSELR17	TAU channel 01 count end	INTTM01
ELSELR18	TAU channel 02 count end	INTTM02
ELSELR19	TAU channel 03 count end/capture end	INTTM03
ELSELR20	TAU channel 10 count end	INTTM10
ELSELR21	TAU channel 11 count end	INTTM11
ELSELR22	TAU channel 12 count end	INTTM12
ELSELR23	TAU channel 13 count end	INTTM13

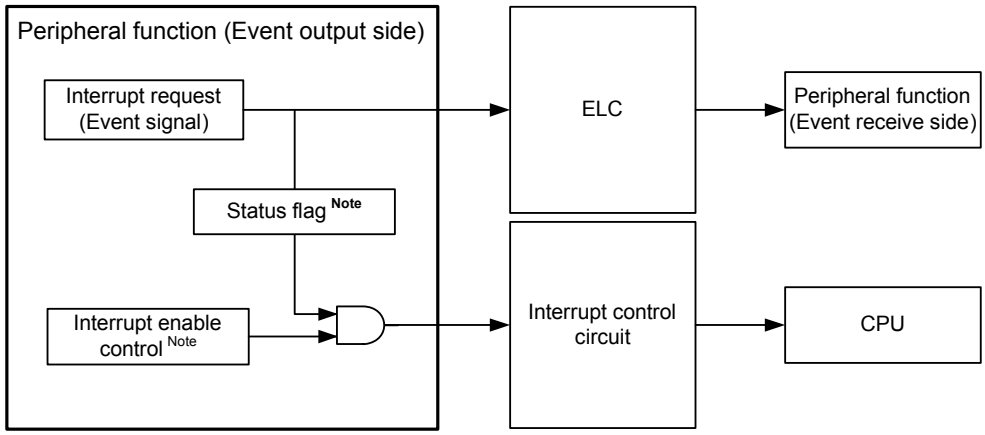
17.4 ELC Operation

The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

Figure 17 - 3 shows the Relationship Between Interrupt Handling and ELC. The figure show an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event-receiving peripheral function after reception of an event.

Figure 17 - 3 Relationship Between Interrupt Handling and ELC



Note Not available depending on the peripheral function.

Table 17 - 2 lists the response of peripheral functions that receive events.

Table 17 - 2 Response of Peripheral Functions That Receive Events

Event Receiver No.	Event Link Destination Function	Operation after Event Reception	Response
1	A/D converter	A/D conversion	An event from the ELC is directly used as a hardware trigger of A/D conversion.
2	Timer array unit 0 Timer input of channel 0	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of fCLK after an ELC event is generated.
3	Timer array unit 0 Timer input of channel 1	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of fCLK after an ELC event is generated.
4	Timer RJ	Count source	An event from the ELC is directly used as the count source of timer RJ.

CHAPTER 18 RF TRANSCEIVER

Precautions for use of RF transceiver
The international standard and the domestic laws and regulations restrict use of the wireless receiver and transmitter. Be sure to observe the standard, laws, regulations of the country where they are used.

18.1 RF Transceiver Overview

RF transceiver supports 2FSK/GFSK and 4FSK/GFSK modulations with lower power consumption, and corresponds to the operation within the frequency band between 863 MHz and 928 MHz.
This transceiver is best suited to the applications used for smart meter (electricity, gas, tap water) supporting products, HEMS controller, wireless sensor network, and others.

The features of the transceiver function are described below.

• Features:

Low power consumption

- RF receiving current: 6.9 mA (TYP) @3.0 V, 100 kbps, 2FSK/at STOP mode of MCU
- RF transmission current: 21 mA (TYP) @3.0 V, 100 kbps, 2FSK, +10 dBm/at STOP mode of MCU

Transceiver:

- RF frequency range: 863 to 928 MHz
- Modulation method: 2FSK/GFSK, 4FSK/GFSK
- Data rate:
  2FSK/GFSK; 10 kbps to 300 kbps
  4FSK/GFSK; 200 kbps/400 kbps
- Forward Error Correction (FEC) function

MAC:

- 32-bit timer function
- Transmission RAM/Receiving RAM: both 256 bytes
- 16-bit/32-bit auto CRC function
- Address filter & auto ACK reply function
- Antenna diversity function
- Auto CSMA-CA function

Caution Registers described in this section are assigned in memory area dedicated to RF transceiver. These registers are controlled by MCU unit through serial interface.

18.2 Pin Functions

18.2.1 Digital pin

This indicates the digital pin of RF unit (RF transceiver).

(1) STANDBY

This is an enable input pin for control of RF unit. Setting to High enables control of RF unit, and Low disables it and turns the power supply to the RF internal circuit off.

Enable the enable control of this pin along with the reset release of MCU. Accordingly, connect to P130 on the user board.

(2) XIN, XOUT/REFCLKIN

These are pins for the reference clock of the RF transceiver.

- When connecting to crystal oscillator
  Connect the 48-MHz crystal oscillator to XIN and XOUT pins.
- When connecting to external clock
  Connect the 48-MHz external clock (square wave) to REFCLKIN pin.
  For this connection, connect XIN pin to GND on the board via the 100-pF decoupling capacitor.

(3) INTOUT

This is an interrupt output pin of RF transceiver.

When the interrupt source is generated in RF unit (RF transceiver), INTOUT pin outputs the status according to the control register setting.

This status signal is connected to INTP3 of MCU in the chip, and MCU can do interrupt processing receiving the status.

(4) GPIO0/CLKOUT, GPIO1/ANTSELOUT0, GPIO2/ANTSELOUT1, GPIO3, GPIO4/ANTSW

These are 5-bit I/O port pins.

GPIO0 pin also provides the clock output (CLKOUT) function, and the default is GPIO0 pin.

<R> Do not use clock output function in transmit or receive state.

GPIO1 and GPIO2 pins are alternative to switch control function for antenna diversity. The defaults are GPIO1 and GPIO2 pins.

GPIO4 pin is alternative to ANTSW signal function. The default is GPIO4.

(5) MODE1, MODE2

Connect these pins to GND on the board so that they are fixed to Low.

18.2.2 Analog pin

The analog pins of the RF transceiver are described below.

(1) RFIP

This is an input pin (RF reception) of the RF transceiver.

(2) RFOUT

This is an output pin (RF transmission) of the RF transceiver.

18.2.3 Description of RF pin functions

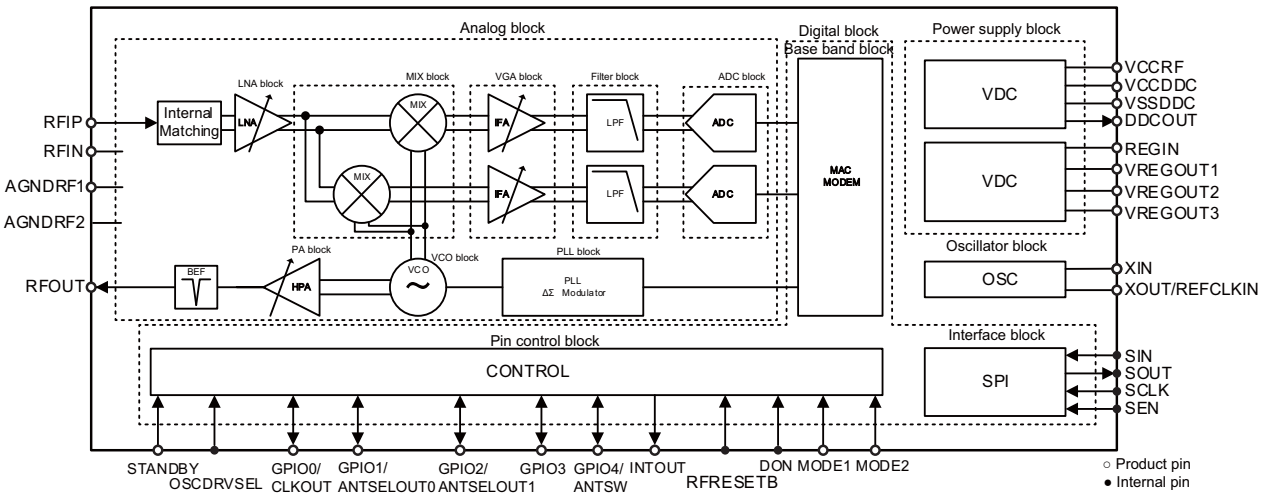
Table 18 - 1 Functions of RF Related Pins

Pin Number	Pin name	I/O	Function
30	INTOUT	Output	Transceiver interrupt output pin
31	GPIO0/CLKOUT	I/O	Transceiver I/O port 0 and CLKOUT pin
32	GPIO1/ANTSELOUT0	I/O	Transceiver I/O port 1 and ANTSELOUT0 pin
33	GPIO2/ANTSELOUT1	I/O	Transceiver I/O port 2 and ANTSELOUT1 pin
34	GPIO3	I/O	Transceiver I/O port 3
35	GPIO4/ANTSW	I/O	Transceiver I/O port 4 and ANTSW pin
36	VREGO2	-	Stabilization capacitor connection pin for VCO (1.1 V)
37	VREGO3	-	Power supply stabilization capacitor connection pin for PLL (1.1 V)
38	MODE2	Input	Mode switch pin 2 (fixed to GND)
39	MODE1	Input	Mode switch pin 1 (fixed to GND)
40	RFIN	-	GND pin
41	RFIP	Input	RF input pin
42	AGNDRF1	-	GND pin
43	RFOUT	Output	RF output pin
44	AGNDRF2	-	GND pin
45	VREGO1	-	Stabilization capacitor connection pin
46	XIN	Input	48 MHz crystal resonator input pin
47	XOUT/REFCLKIN	I/O	48 MHz crystal resonator output and external clock input pin
48	VCCRF	-	Power supply pin
49	REGIN	-	DDCOUT and external connection pin
50	VSSDDC	-	GND pin
51	DDCOUT	-	External connection pin to REGIN
52	VCCDDC	-	Power supply pin
53	STANDBY	Input	Power down control input pin of transceiver
Back side	DIEGND	-	GND

18.3 Configuration of RF Transceiver

RF transceiver consists of the analog block, digital block, power supply circuit block, oscillator block, and interface block.Figure 18 - 1 shows the block diagram. The next section provides description in detail.

Figure 18 - 1 RF Transceiver Block Diagram



18.3.1 Digital block

(1) Baseband block

For details of the baseband block, see **18.4**.

(2) Interface block

This block is equipped with the SPI function for interfacing with MCU.

For details, see **18.5**.

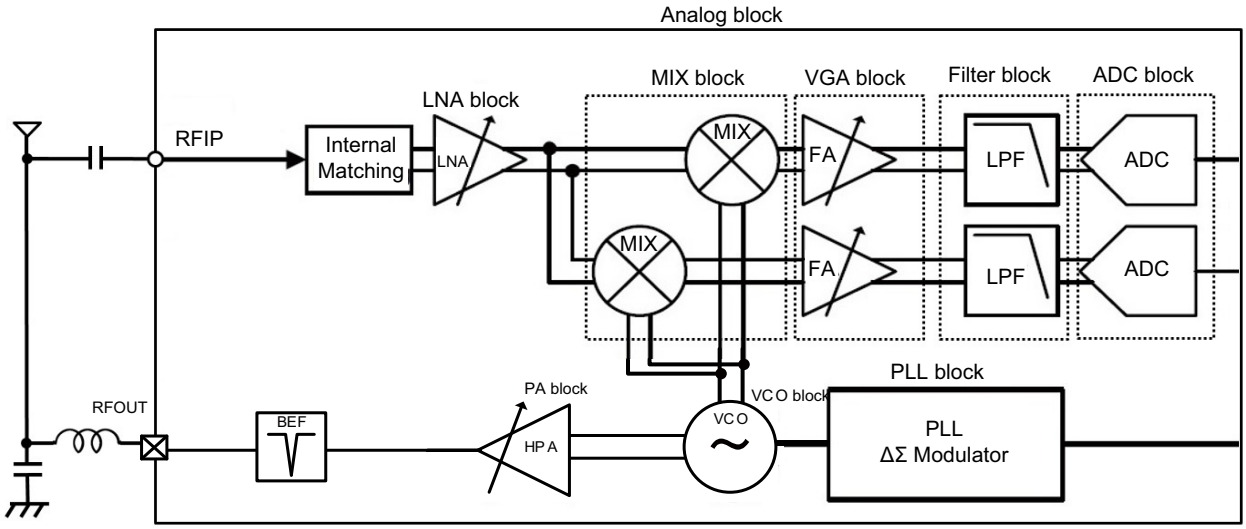
(3) Pin control block

Controls I/O, etc. of pins.

18.3.2 Analog block

The analog block consists of RF transmission/reception circuits. Figure 18 - 2 shows the block diagram of the RF transmission/reception block.

Figure 18 - 2 RF Transmission/Reception Block



(1) LNA block

This is a low noise amplifier which input the high frequency signal captured by antenna.
It switches the gain according to the input signal.

(2) MIX block

This is a mixer circuit which uses the high frequency RF output signal from the LNA as input signal and converts it into low frequency IF (Intermediate Frequency) signal by changing the frequency.

(3) VGA block

This block controls the input amplitudes under the constant state that are transmitted to the AD converter which is connected to the subsequent stage, by changing the gain depending on the input signal level.

(4) Filter block

This removes unnecessary input signals from antenna.

(5) ADC block

This converts analog signal to digital signal.

(6) VCO and PLL blocks

They generate the carrier frequency which is used upon reception/transmission based on the reference clock which is input by the crystal oscillator or TCXO.

(7) PA block

This is a power amplifier which is located at the output stage in the analog circuit in order to output the high frequency signal from antenna.

18.3.3 Oscillator block

The oscillator block generates the reference clock to be supplied to the internal circuit of RF unit. This block also has the function to externally output the generated reference clock.
RF reference clock and RF clock output are classified into the following types.

(1) RF reference clock
RF reference clock is supplied to the digital baseband block entirely.

- XTAL_RF oscillator
  Connection of XIN and XOUT pins to 48-MHz crystal oscillator enables 48-MHz clock to oscillate.
  It takes the oscillation stabilization time from the oscillation stop until the stabilization.
  The oscillation stabilization time requires the stabilization wait time 500 μs + *α* (*α* varies depending on the oscillation circuit).

- REFCLKIN_RF external clock
  This clock can input the 48-MHz square wave to REFCLKIN pin. Use OSCDRVSEL pin that is an internal pin of RF unit with High level fixed in order to use this external input.

(2) RF reference clock output
The divided clock of RF reference clock can be output from CLKOUT pin. The frequencies 12 MHz, 8 MHz, and 4 MHz can be specified for outputting. <R>

18.3.4 Power supply block

The power supply block consists of DC-DC converter and regulator.

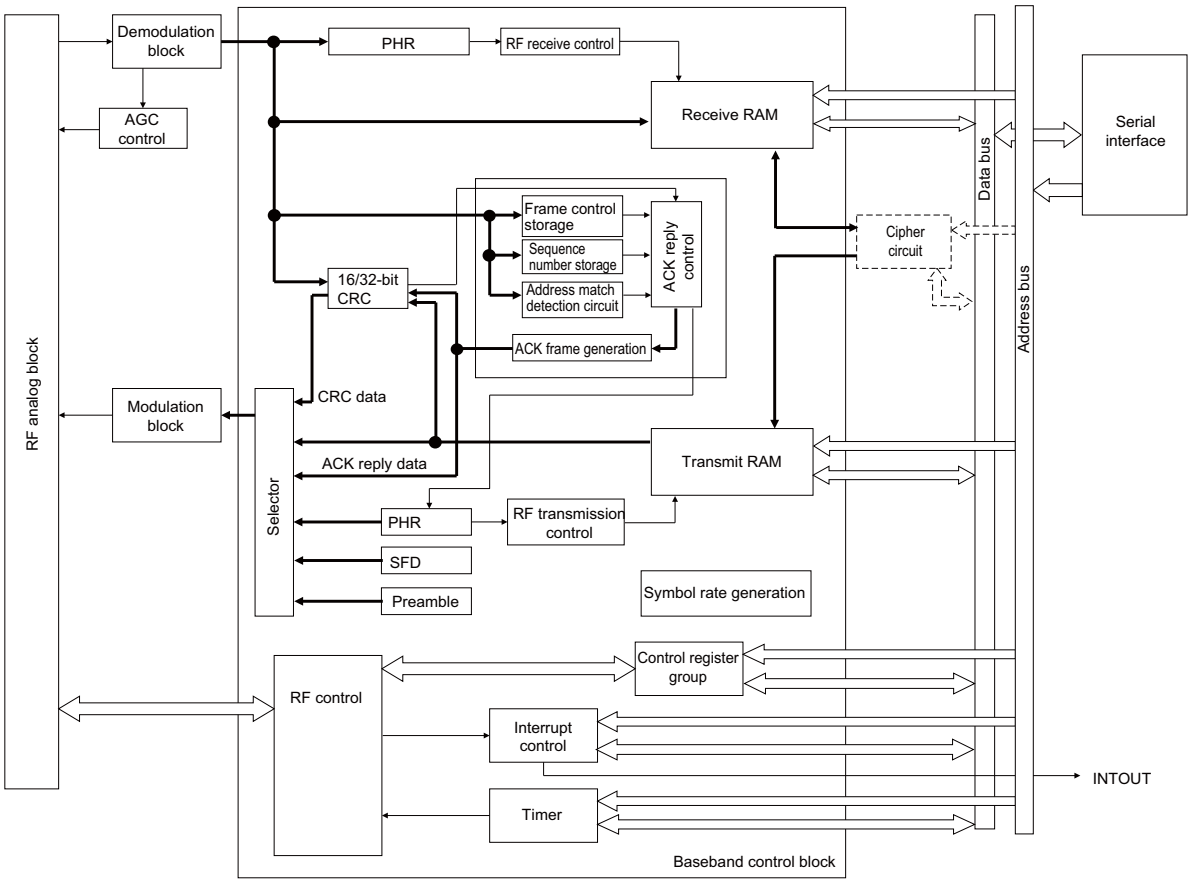
(1) DC-DC converter
The switching regulator is provided to generate a stabilized power from the power supply (1.8 to 3.6 V) for the internal power supply. Control of DON pin (RF internal pin) enables to set ON/OFF.

18.4 Baseband Function

18.4.1 Configuration

Figure 18 - 3 shows the baseband block diagram.

Figure 18 - 3 Baseband Block Diagram

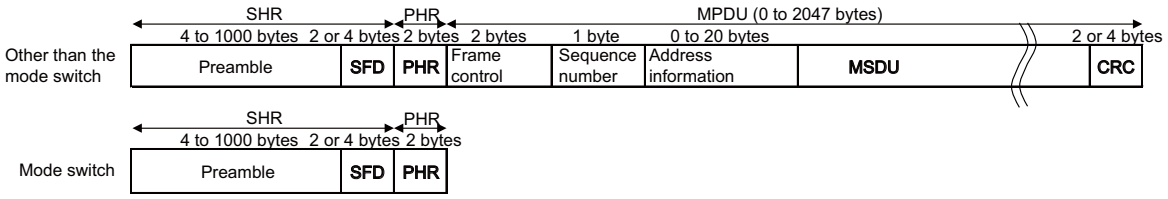


18.4.2 Frame configuration

(1) RF transmission mode

Automatically generates and outputs the transmission frame. Figure 18 - 4 shows the transmission frame configuration. The frame configurations after the PHR other than those of the mode switch frame and the mode switch frame change.

Figure 18 - 4 Transmission Frame Configuration



SHR: Synchronization Header
PHR: PHY Header
MPDU: MAC Protocol Data Unit
SFD: Start of Frame Delimiter
MSDU: MAC Service Data Unit
CRC: Cyclic Redundancy Check

<1> Preamble: 4 to 1000 bytes

<2> SFD: 2 bytes or 4 bytes

<3> PHR: 2 bytes

Value varies when it is transmitted by the frame other than the mode switch frame or the mode switch frame.

- Frames other than the mode switch frame: The value written in the BBTXFLEN register (the length of MPDU), the CRCBIT value (FCSType), and the DWEN value (Data Whitening)
- Mode switch frame: the value written in the BBTXMODESW register

<4> MPDU: data whose length ranged from the minimum 3 bytes to the maximum 2047 bytes. The data written in the transmission RAM is output sequentially.

Note that the CR data which is generated in the CRC calculation circuit is automatically added to the last 2 bytes or 4 bytes when the NOCRC bit in the BBTXRXMODE2 register is “0.”

- Frame control: 2 bytes (4 symbols)
  - Frame type (bit 2 to 0)
    000b: Beacon frame
    001b: the data frame
    010b: the ACK frame
    011b: the MAC command frame
    100b to 111b: reserved
  - With/without security (bit 3). Transmission pending bit (bit 4)
  - ACK request (bit 5), Transmission in the PAN (bit 6)
  - Transmission source address mode (bits 10 and 11), Transmission destination address mode (bits 14 and 15)
  - Frame version (bits 12, 13)
- Sequence number: 1 byte
- Address information: PADNID and address for transmission destination and source
- MSDU (MAC payload): frame payload
- CRC: 2-byte or 4-byte frame CRC inspection array

(2) Preamble value and SFD value

Set preamble length values to preamble length setting register (00A7H, 00A6H addresses) equal or more than the below.

Table 18 - 2 For Normal Receive

<R>

Condition			Preamble length
Way	Symbol rate (ksps)	Modification index	
2FSK/2GFSK	10	0.5	4
2FSK/2GFSK	20	0.5	4
2FSK/2GFSK	40	0.5	4
2FSK/2GFSK	50	1	4
2FSK/2GFSK	100	0.5	4
2FSK/2GFSK	100	1	4
2FSK/2GFSK	150	0.5	5
2FSK/2GFSK	200	0.5	6
2FSK/2GFSK	200	1	6
2FSK/2GFSK	300	0.5	7
4FSK/4GFSK	100	0.33	5
4FSK/4GFSK	200	0.33	6

Table 18 - 3 For Antenna Diversity Receive

<R>

Condition			Preamble length
Way	Symbol rate (ksps)	Modification index	
2FSK/2GFSK	10	0.5	8
2FSK/2GFSK	20	0.5	8
2FSK/2GFSK	40	0.5	8
2FSK/2GFSK	50	1	8
2FSK/2GFSK	100	0.5	15
2FSK/2GFSK	100	1	15
2FSK/2GFSK	150	0.5	18
2FSK/2GFSK	200	0.5	23
2FSK/2GFSK	200	1	23
2FSK/2GFSK	300	0.5	24
4FSK/4GFSK	100	0.33	15
4FSK/4GFSK	200	0.33	23

(3) Configuration of PHR

<1> In the case other than the Mode Switch frame

Figure 18 - 5 shows the PHR bit configuration in the case other than the Mode Switch frame.

The respective PHR bits automatically capture the set value and transmit it to the corresponding register.

Figure 18 - 5 PHR Bit Configuration in the Case Other Than the Mode Switch Frame

Bit array	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit configuration	MS	R1	R0	FCS	DW	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1	L0

MS: Mode Switch

Mode switch enable bit value: Mode switch frame transmission register bit 0

Automatically transmits the following values when this bit value is 0.

R1, R0: Reserved

Fixed to 00b

FCS: FCS Type

The-number-of-CRC-bit switch bit value: SUBG control register bit 3

DW: Data Whitening

DW enable bit value: SUBG control register bit 4

L10 to L0: Frame Length

Transmission frame length register value: Transmit frame length register bits 0 to 10

<2>In the case of the Mode Switch frame

Figure 18 - 6 shows the PHR bit configuration in the case of the Mode Switch frame.

The respective PHR bits automatically capture the set value and transmit it to the corresponding register.

Only the register values are transmitted on the hardware while no Checksum or Parity Check function is included.

Figure 18 - 6 PHR Bit Configuration in the Case of the Mode Switch Frame

Bit array	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit configuration	MS	M1	M0	FEC	New Mode							B3	B2	B1	B0	PC

MS: Mode Switch

Mode switch enable bit value: Mode switch frame transmission register bit 0

Automatically transmits the following values when this bit value is 1.

M1, M0: Mode Switch Parameter Entry

Mode switch frame transmission register bits 1 and 2

FEC: New Mode FEC

Mode switch frame transmission register bit 3

New Mode: New Mode

Mode switch frame transmission register bits 4 to 10

B3 to B0: Checksum

Mode switch frame transmission register bits 11 to 14

PC: Parity Check

Mode switch frame transmission register bit 15

18.4.3 Baseband Interrupt

Table 18 - 4 shows the interrupt sources.

If one or more bits are “1” in interrupt source bits in the table enabled by baseband interrupt registers 0 to 2, INTOUT pin outputs high level. If the all interrupt source bits are “0”, the pin outputs low level.

The polarity of INTOUT output can be changed by INTOUTSEL bit.

Table 18 - 4 Interrupt Sources List

Number	Interrupt Name	Interrupt Occurrence Condition
1	Timer compare 0	Interrupt request occurs when timer value matches timer compare 0 value
2	Timer compare 1	Interrupt request occurs when timer value matches timer compare 1 value
3	Timer compare 2	Interrupt request occurs when timer value matches timer compare 2 value
4	Frame transmit completion	Interrupt request occurs when a frame transmission completes However, interrupt request does not occur at transmit completion when automatic ACK receive mode is enabled, if the transmit frame includes ACK request. The interrupt request occurs at completion of ACK reception or ACK receive timeout
5	Bank 0 transmit completion	Interrupt request occurs when transmit RAM data transmission of bank 0 completes
6	Bank 1 transmit completion	Interrupt request occurs when transmit RAM data transmission of bank 1 completes
7	CCA completion**Note 1**	Interrupt request occurs when CCA sequence completes
8	CSMA-CA completion**Note 1**	Interrupt request occurs when CSMA-CA sequence completes
9	Frame receive completion	Interrupt request occurs when a frame reception completes However, interrupt request does not occur at receive completion when automatic ACK reply mode is enabled, if the receive frame includes ACK request. The interrupt request occurs at completion of ACK reply
10	Bank 0 receive completion	Interrupt request occurs when bank 0 of receive RAM becomes full
11	Bank 1 receive completion	Interrupt request occurs when bank 1 of receive RAM becomes full
12	Start reception	Interrupt request occurs when SFD is detected
<R> 13	Address filter**Note 4**	Interrupt request occurs when address match is recognized
14	Receive overrun	Interrupt request occurs in the following cases - Receive RAM bank 0 status bit is kept 1 and data reception to receive RAM bank 0 is started again - Receive RAM bank 1 status bit is kept 1 and data reception to receive RAM bank 1 is started again
15	Mode switch receive completion	Interrupt request occurs when a mode switch frame is received
16	Receive level filter**Note 2**	Interrupt request occurs when a communication error occurs such as no signal state during frame reception
17	Receive byte counts	Interrupt request occurs when setting count of bytes are received
18	Frame length	Interrupt request occurs when the frame length is received
19	Byte reception**Note 3**	Interrupt request occurs for each receive byte data
20	Automatic receive timeout**Note 3**	Interrupt request occurs when timeout occurs with no receive frame, if the automatic receive timeout function is enabled

Note 1. CCA completion interrupt and CSMA-CA completion interrupt are switched by CCATINTSEL bit.

<R> **Note 2.** See 18.4.4 (89) for setting condition of threshold level occurring receive level filter interrupt causing detection of no communication state.

Note 3. Byte reception interrupt and automatic receive timeout interrupt are switched by TIMEOUTINTSEL bit.

<R> **Note 4.** When the frame length is short and the frame has no address information or incomplete address information, address filter interrupt might not occur and only frame receive completion occurs.

18.4.4 Baseband Function Controlling Register

Table 18 - 5 and Table 18 - 6 show the registers to control the baseband function.

For how to access the control register, see **18.5**.

Table 18 - 5 Baseband Function Controlling Register (1/2)

No.	Register name	Symbol	Address	Reset value
(1)	RF start register	BBRFCON	0000H	01H
(2)	Transmit/receive reset register	BBTXRXRST	0001H	00H
(3)	Transmit/receive mode register 0	BBTXRXMODE0	0002H	00H
(4)	Transmit/receive mode register 1	BBTXRXMODE1	0003H	C0H
(5)	RSSI/CCA result register	BBRSSICCARSLT	0005H, 0004H	0000H
(6)	Enhanced-ACK mode register	BBEACKMODE	0006H	01H
(7)	Transmit/receive status register 0	BBTXRXST0	0007H	80H
(8)	Transmit/receive mode register 2	BBTXRXMODE2	0009H	30H
(9)	Transmit/receive mode register 3	BBTXRXMODE3	000AH	00H
(10)	Transmit/receive status register 1	BBTXRXST1	000BH	02H
(11)	Transmit/receive control register	BBTXRXCON	000CH	00H
(12)	CSMA control register 0	BBCSMACON0	000DH	00H
(13)	CCA level threshold setting register	BBCCAVTH	000FH, 000EH	0100H
(14)	Transmit/receive status register 2	BBTXRXST2	0010H	03H
(15)	Transmit/receive mode register 4	BBTXRXMODE4	0011H	01H
(16)	CSMA control register 1	BBCSMACON1	0012H	20H
(17)	CSMA control register 2	BBCSMACON2	0013H	05H
(18)	PAN identifier register 0	BBPANID0	0015H, 0014H	FFFFH
(19)	Short address register 0	BBSHORTAD0	0017H, 0016H	FFFFH
(20)	Extended address register 0	BBEXTENDAD03	001FH, 001EH	0000H
		BBEXTENDAD02	001DH, 001CH	0000H
		BBEXTENDAD01	001BH, 001AH	0000H
		BBEXTENDAD00	0019H, 0018H	0000H
(21)	Timer read register	BBTIMEREAD0	0021H, 0020H	0000H
		BBTIMEREAD1	0023H, 0022H	0000H
(22)	Timer compare register	BBTCOMP2REG0	002DH, 002CH	0000H
		BBTCOMP1REG0	0029H, 0028H	0000H
		BBTCOMP0REG0	0025H, 0024H	0000H
		BBTCOMP2REG1	002FH, 002EH	0000H
		BBTCOMP0REG1	002BH, 002AH	0000H
		BTCOMP0REG1	0027H, 0026H	0000H
(23)	Time stamp register	BBTSTAMP0	0031H, 0030H	0000H
		BBTSTAMP1	0033H, 0032H	0000H
(24)	Timer control register	BBTIMECON	0034H	00H
(25)	Back off period register	BBBOFFPROD	0035H	00H
(26)	Baseband interrupt source register 0	BBINTREQ0	0036H	00H
(27)	Baseband interrupt source register 1	BBINTREQ1	0037H	00H
(28)	Baseband interrupt source register 2	BBINTREQ2	0038H	00H
(29)	Baseband interrupt enable register 0	BBINTEN0	0039H	00H
(30)	Baseband interrupt enable register 1	BBINTEN1	003AH	00H
(31)	Baseband interrupt enable register 2	BBINTEN2	003BH	00H
(32)	CSMA control register 3	BBCSMACON3	003EH	03H
(33)	ACK counter compare register	ACKCOMP0	0041H, 0040H	0014H
		ACKCOMP1	0043H, 0042H	000EH
(34)	ACK counter initial value register	ACKINI	005FH, 005EH	0000H
(35)	Receive level threshold setting register	BBLVLVTH	0045H, 0044H	0100H
(36)	ACK reply time setting register	ACKRTNTIM	0047H, 0046H	0002H
(37)	Automatic receive switch compare register	AUTORCVCNT	0049H, 0048H	000AH
(38)	Back off cycle register	BOFFPERIOD	004BH, 004AH	0071H
(39)	CSMA-CA end count register	CSMAENDCOUNT	004DH, 004CH	0080H

Table 18 - 6 Baseband Function Controlling Register (2/2)

No.	Register name	Symbol	Address	Reset value
(40)	CSMA-CA start count register	CSMASTACOUNT	004FH, 004EH	000EH
(41)	Communication status register 1	COMSTATE1	0066H	00H
(42)	Communication status register 2	COMSTATE2	0067H	00H
(43)	Estimate control register	BBEVAREG	0068H	00H
(44)	Back off period register 2	BBBOFFPROD2	0069H	00H
(45)	Communication status register 3	COMSTATE3	006FH	00H
(46)	ACK receive wait time setting register	ACKRCVWIT	0071H, 0070H	0300H
(47)	Retransmission start compare register	RETRNWUP	0073H, 0072H	0004H
(48)	ANTSW output timing setting register	BBANTSWTIMG	007BH, 007AH	0072H
(49)	Receive frame length register	BBRXFLEN	00A1H, 00A0H	0000H
(50)	Receive data counter register	BBRXCOUNT	00A3H, 00A2H	0000H
(51)	Transmit frame length register	BBTXFLEN	00A5H, 00A4H	0000H
(52)	Preamble length setting register	BBPAMBL	00A7H, 00A6H	0004H
(53)	Frequency setting register	BBFREQ	00ABH-00A8H	36FC3BA0H
(54)	Symbol rate setting register	BBSYMBLRATE	00AEH-00ACH	0000F0
(55)	SUBG control register	BBSUBGCON	00B0H	0CH
(56)	Modulation method setting register	BBMODSET	00B1H	42H
(57)	CCA time register	CCATIME	00B3H, 00B2H	000DH
(58)	Antenna diversity mode register	BBANTDIV	00B4H	30H
(59)	Mode switch frame transmit register	BBTXMODESW	00B9H, 00B8H	0000H
(60)	Mode switch frame receive register	BBRXMODESW	00BBH, 00BAH	0000H
(61)	Transmit data counter register	BBTXCOUNT	00BDH, 00BCH	0000H
(62)	PHY header receive register	BBPHRRX	00BEH	00H
(63)	Preamble setting register	BBPABL	00C1H, 00C0H	00AAH
(64)	SFD setting register	BBSFD	00C5H-00C2H	00007209H
(65)	SHR control register	BBSHRCON	00C6H	02H
(66)	ANT0 read register	BBANT0RD	00C9H, 00C8H	0100H
(67)	ANT1 read register	BBANT1RD	00CBH, 00CAH	0100H
(68)	Antenna switch time register	BBANTDIVTIM	00CFH, 00CEH	000AH
(69)	Receive start timeout setting register	BBANTTIMOUT	00D1H, 00D0H	0100H
(70)	Number-of-receive-byte interrupt compare register	BBRCVINTCOMP	00D3H, 00D2H	0010H
(71)	Back off period total number register	BBBOPTOTAL	00D5H, 00D4H	0000H
(72)	CCA total number	BBCCATOTAL	00D6H	00H
(73)	RF Initial setting register0	RFINI00-RFINI02	00D8H - 00DAH	XXH
(74)	RF Initial setting register1	RFINI10-RFINI12	00DCH - 00DEH	XXH
(75)	PAN identifier register 1	BBPANID1	00E1H, 00E0H	FFFFH
(76)	Short address register 1	BBSHORTAD1	00E3H, 00E2H	FFFFH
(77)	Extended address register 1	BBEXTENDAD13	00EBH, 00EAH	0000H
		BBEXTENDAD12	00E9H, 00E8H	0000H
		BBEXTENDAD11	00E7H, 00E6H	0000H
		BBEXTENDAD10	00E5H, 00E4H	0000H
(78)	Receive timeout register	BBTIMEOUT	00EDH, 00ECH	07D0H
(79)	ANTSW control register	ANTSWCON	0080H	00H
(80)	Clock output control register	CLKOUTCON	0081H	00H
(81)	Port direction register	GPIODIR	0082H	00H
(82)	Port data register	GPIODATA	0083H	00H
(83)	SFD status register 2	BBSFD2	0103H-0100H	XXXXXXXXH
(84)	SFD status register 3	BBSFD3	0107H-0104H	XXXXXXXXH
(85)	SFD status register 4	BBSFD4	010BH-0108H	XXXXXXXXH
(86)	FEC setting register	BBFECCON	010CH	00H
(87)	Address filter extended address setting register	BBADFCON	010DH	00H
(88)	Antenna diversity mode register2	BBANTDIV2	010EH	00H
(89)	Lower limit threshold setting register after SFD detection	PWRLOWTH2	050FH	00H

(1) RF start register (BBRFCON)

Enables the RF function by setting 1 to the RF function enable bit.

Releases the analog circuit reset by setting 1 to the analog reset release bit.

The BBRFCON register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 01H.

Figure 18 - 7 RF Start Register (BBRFCON) Format

Address: 0000H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
BBRFCON	0	0	0	0	0	0	ANARESETB	RFSTART

ANARESETB	Analog reset release bit
0	Analog circuit reset status
1	Analog circuit reset release

RFSTART	RF function enable bit
0	RF function disabled
1	RF function enabled

Caution Be sure to clear bits 2 to 7 to “0”.

(2) Transmit/receive reset register (BBTXRXRST)

This register stops the RF communication.

Processes of reception, during the CCA and calibration can be stopped by setting 1 to the RF communication stop bit. (It enters into the IDLE state after the stop) In addition, the processes of automatic ACK reply, the automatic receive switch mode function, and others are also canceled. Note that this bit is automatically cleared to 0. However, the values of the respective registers can be retained.

Reset by RFRESETB pin after setting RFSTOP bit to 1 for stopping transmission.

The BBTXRXRST register is set by the serial interface in 8-bit units.

Reset signal generation clears this register to 00H

Figure 18 - 8 Transit/Receive Reset Register (BBTXRXRST) Format

Address: 0001H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBTXRXRST	0	0	0	0	0	0	0	RFSTOP

RFSTOP	RF communication stop trigger bit
0	Nothing is performed. (The RF communications are under the enabled status.)
1	Stops the RF communication. (It is under the IDLE state after the stop while this bit is automatically cleared to 0.)

Caution Be sure to clear bits 1 to 7 to “0”.

(3) Transmit/receive mode register 0 (BBTXRXMODE0)

This register sets the various types of the RF transmit/receive mode.

Set 1 to the CCA type bit when performing the CCA or ED.

Enables the selection whether to perform the automatic ACK reply action after the completion of receive by using the automatic ACK mode enable bit.

Enables the setting to automatically transit to the receive mode after the completion of transmit by using the automatic receive switch mode 0 enable bit. Enables the setting to automatically transit to the receive mode after the completion of receive by using the automatic receive switch mode 1 enable bit. However, note that the ACK reply takes precedence if all of the ACK reply conditions are satisfied when 1 (enable) is set to the automatic ACK enable bit.

The battery life extension mode bit can set the battery life extension mode to enable for the branch condition when performing the CSMA-CA process.

The BEACON mode bit can switch between Non-beacon/Beacon modes.

The intraPAN enable bit can set enable/disable by the intraPAN bit condition of the receive frame when the address filter is enabled.

The BBTXRXMODE0 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears this register to 00H.

Figure 18 - 9 Transmit/Receive Mode Register 0 (BBTXRXMODE0) Format

Address: 0002H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBTXRXMODE0	INTRAPANEN	BEACON	BATLIFEEXT	AUTORCV1	AUTORCV0	AUTOACKEN	0	CCATYPE0

INTRAPANEN	intraPAN enable bit
0	intraPAN disabled
1	intraPAN enabled

BEACON	BEACON mode bit
0	Non-beacon mode
1	BEACON mode

BATLIFEEXT	Battery life extension mode bit
0	Battery life extension mode disabled
1	Battery life extension mode enabled

AUTORCV1	Automatic receive switch mode 1 enable bit (Receive → Receive)
0	Auto receive switch function disabled
1	Auto receive switch function enabled

AUTORCV0	Automatic receive switch mode 0 enable bit (Transmit → Receive)
0	Auto receive switch function disabled
1	Auto receive switch function enabled

AUTOACKEN	Automatic ACK mode enable bit
0	Automatic ACK disabled
1	Automatic ACK enabled

CCATYPE0	CCA type bit
0	Normal
1	CCA/ED

Caution Be sure to clear bit 1 to “0”.

(4) Transmit/receive mode register 1 (BBTXRXMODE1)

This register sets the various types of the RF transmit/receive mode.

The CCA result select bit can be used to select between the CCA/ED value and the RSSI value when reading the RSSI/CCA result register.

The ACK reply frame version setting enable bit can switch the value of the version upon automatic ACK reply between the frame version value of the received frame, and the values of the ACK replay frame version setting bits 0 and 1.

The ACK replay frame version setting bit 0 can be used to set the value of bit 12 for the frame version value upon automatic ACK reply when the ACK reply frame version setting enable bit = 1.

The ACK replay frame version setting bit 1 can be used to set the value of bit 13 for the frame version value upon automatic ACK reply when the ACK reply frame version setting enable bit = 1.

The sequence number suppress bit can be used to set the enable/disable to the sequence number suppress bit of the received frame. There is also no sequence number for the ACK frame upon the automatic ACK reply when there is no sequence number for the received frame when this bit is set to enable.

The ACK receive point setting bit can be used to set the ACK receive recognition point during the ACK receive wait time to the point upon completion of the reception or upon reception of the PHR.

The BBTXRXMODE1 register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to C0H.

Figure 18 - 10 Transmit/Receive Mode Register 1 (BBTXRXMODE1) Format

Address: 0003H After reset: C0H R/W

Symbol	7	6	5	4	3	2	1	0
BBTXRXMODE1	ACKRCVPOINT	SQCNUMSUPEN	ACKFV1	ACKFV0	ACKFVEN	0	0	CCASEL

ACKRCVPOINT	ACK receive point setting bit
0	Upon completion of the receive
1	Upon reception of the PHR

SQCNUMSUPEN	Sequence number suppress bit
0	Sequence number suppress enabled
1	Sequence number suppress disabled

ACKFV1	ACK reply frame version setting bit 1
0	Frame version bit 13 = 0
1	Frame version bit 13 = 1

ACKFV0	ACK reply frame version setting bit 0
0	Frame version bit 12 = 0
1	Frame version bit 12 = 1

ACKFVEN	ACK reply frame version setting enable bit
0	Version value of the receive frame
1	Values of the ACK reply frame version setting bits 0 and 1

CCASEL	CCA result select bit
0	CCA/ED
1	RSSI

Caution Be sure to clear bits 1 and 2 to “0”.

(5) RSSI/CCA result register (BBRSSICCARSLT)

Stores the result data of CCA/ED or RSSI.

The CCA result select bit can be used to switch between the CCA/ED value and the RSSI value.

The result corresponding to the save bank which is specified by the receive data save bank select bit is read out when reading the RSSI value.

The read data is expressed as 2's complement. The unit is dBm (ex. "19EH" is "-98dBm").

The BBRSSICCARSLT register is read by the serial interface in 8-bit units.

Reset signal generation clears this register to 0000H.

Figure 18 - 11 RSSI/CCA Result Register (BBRSSICCARSLT) Format

Address: 0005H, 0004H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
BBRSSICCARSLT	0	0	0	0	0	0	0	CCARSLT8

7	6	5	4	3	2	1	0
CCARSLT7	CCARSLT6	CCARSLT5	CCARSLT4	CCARSLT3	CCARSLT2	CCARSLT1	CCARSLT0

CCARSLT[8:0]	RSSI/CCA result data

Caution "0" is always read for bits 15 to 9.

The loss of antenna input signal revel defers causing LSI board layout and usage of RFSW or SAW filter. Therefore, setting values are required modification to match the input level and value of RSSI/CCA result register (BBRSSICCARSLT). The value of receive level threshold setting register (BBLVLVTH) also required modification.

Set values in Table 18 - 7 Register setting for loss of input signal level from antenna in IDLE state to cancel the loss. The values are deferent for operation enabled/disabled FEC.

The values in Table 18 - 52 to Table 18 - 59 Settings Required for Each Data Rate are reference measured on evaluation board of Renesas Electronics.

Caution Refer to the latest application note for setting data in Table 18 - 7 when to use the product.

Table 18 - 7 Register setting for loss of input signal level from antenna **(1/4)**

Frequency band identifier	PHY	Mode	Data rate (kbps)	1dB loss						
				0044		0483	047C	047D	047E	047F
				FEC: Enabled	FEC: Disable					
4	863MHz (Europe)	001	50	87	8F	3F	1B	0F	03	34
		002	100	8E	96	3F	1A	0E	02	33
		003	200	8E	96	3F	1A	0E	02	34
5	896MHz (US)	004	10	84	8C	0A	1B	0F	03	34
		005	20	88	90	04	1A	0F	03	34
		006	40	8C	94	04	1A	0F	03	34
6	901MHz (US)	007	10	84	8C	0A	1B	0F	03	34
		008	20	88	90	04	1A	0F	03	34
		009	40	8C	94	04	1A	0F	03	34
7	915MHz (US)	010	50	8D	95	3C	1B	0F	03	34
		011	150	92	9A	0A	1B	0F	03	34
		012	200	91	99	0A	1B	0F	03	34
8	917MHz (Korea)	013	50	8D	95	3F	1A	0E	02	33
		014	150	92	9A	0A	1B	0F	03	34
		015	200	91	99	0A	1B	0F	03	34
9	920MHz (Japan)	016	50	8D	95	3D	1B	0F	03	34
		000	100	8E	96	3E	1B	0F	03	34
		017	200	93	9B	3D	1B	0F	03	34
		018	400	8D	95	3D	1B	0F	03	34
other		019	300	95	9D	1A	19	0D	03	33
		020	100	8E	96	3E	1B	0F	03	34
		021	300	95	9D	1A	19	0D	03	33

Table 18 - 7 Register setting for loss of input signal level from antenna **(2/4)**

Frequency band identifier	PHY	Mode	Data rate (kbps)	2dB loss						
				0044		0483	047C	047D	047E	047F
				FEC: Enabled	FEC: Disable					
4	863MHz (Europe)	001	50	88	90	3F	1A	0E	02	33
		002	100	8F	97	3F	19	0D	01	32
		003	200	8F	97	3F	19	0D	01	33
5	896MHz (US)	004	10	85	8D	0B	1B	0F	03	34
		005	20	89	91	05	1A	0F	03	34
		006	40	8D	95	05	1A	0F	03	34
6	901MHz (US)	007	10	85	8D	0B	1B	0F	03	34
		008	20	89	91	05	1A	0F	03	34
		009	40	8D	95	05	1A	0F	03	34
7	915MHz (US)	010	50	8E	96	3D	1B	0F	03	34
		011	150	93	9B	0B	1B	0F	03	34
		012	200	92	9A	0B	1B	0F	03	34
8	917MHz (Korea)	013	50	8E	96	3F	19	0D	01	32
		014	150	93	9B	0B	1B	0F	03	34
		015	200	92	9A	0B	1B	0F	03	34
9	920MHz (Japan)	016	50	8E	96	3E	1B	0F	03	34
		000	100	8F	97	3F	1B	0F	03	34
		017	200	94	9C	3E	1B	0F	03	34
		018	400	8E	96	3E	1B	0F	03	34
other		019	300	96	9E	1B	19	0D	03	33
		020	100	8F	97	3F	1B	0F	03	34
		021	300	96	9E	1B	19	0D	03	33

Table 18 - 7 Register setting for loss of input signal level from antenna **(3/4)**

Frequency band identifier	PHY	Mode	Data rate (kbps)	3dB loss						
				0044		0483	047C	047D	047E	047F
				FEC: Enabled	FEC: Disable					
4	863MHz (Europe)	001	50	89	91	3F	19	0D	01	32
		002	100	90	98	3F	18	0C	00	31
		003	200	90	98	3F	18	0C	00	32
5	896MHz (US)	004	10	86	8E	0C	1B	0F	03	34
		005	20	8A	92	06	1A	0F	03	34
		006	40	8E	96	06	1A	0F	03	34
6	901MHz (US)	007	10	86	8E	0C	1B	0F	03	34
		008	20	8A	92	06	1A	0F	03	34
		009	40	8E	96	06	1A	0F	03	34
7	915MHz (US)	010	50	8F	97	3E	1B	0F	03	34
		011	150	94	9C	0C	1B	0F	03	34
		012	200	93	9B	0C	1B	0F	03	34
8	917MHz (Korea)	013	50	8F	97	3F	18	0C	00	31
		014	150	94	9C	0C	1B	0F	03	34
		015	200	93	9B	0C	1B	0F	03	34
9	920MHz (Japan)	016	50	8F	97	3F	1B	0F	03	34
		000	100	90	98	3F	1A	0E	02	33
		017	200	95	9D	3F	1B	0F	03	34
		018	400	8F	97	3F	1B	0F	03	34
other		019	300	97	9F	1C	19	0D	03	33
		020	100	90	98	3F	1A	0E	02	33
		021	300	97	9F	1C	19	0D	03	33

Table 18 - 7 Register setting for loss of input signal level from antenna **(4/4)**

Frequency band identifier	PHY	Mode	Data rate (kbps)	4dB loss						
				0044		0483	047C	047D	047E	047F
				FEC: Enabled	FEC: Disable					
4	863MHz (Europe)	001	50	8A	92	3F	18	0C	00	31
		002	100	91	99	3F	17	0B	3F	30
		003	200	91	99	3F	17	0B	3F	31
5	896MHz (US)	004	10	87	8F	0D	1B	0F	03	34
		005	20	8B	93	07	1A	0F	03	34
		006	40	8F	97	07	1A	0F	03	34
6	901MHz (US)	007	10	87	8F	0D	1B	0F	03	34
		008	20	8B	93	07	1A	0F	03	34
		009	40	8F	97	07	1A	0F	03	34
7	915MHz (US)	010	50	90	98	3F	1B	0F	03	34
		011	150	95	9D	0D	1B	0F	03	34
		012	200	94	9C	0D	1B	0F	03	34
8	917MHz (Korea)	013	50	90	98	3F	17	0B	3F	30
		014	150	95	9D	0D	1B	0F	03	34
		015	200	94	9C	0D	1B	0F	03	34
9	920MHz (Japan)	016	50	90	98	3F	1A	0E	02	33
		000	100	91	99	3F	19	0D	01	32
		017	200	96	9E	3F	1A	0E	02	33
		018	400	90	98	3F	1A	0E	02	33
other		019	300	98	A0	1D	19	0D	03	33
		020	100	91	99	3F	19	0D	01	32
		021	300	98	A0	1D	19	0D	03	33

(6) Enhanced-ACK mode register (BBEACKMODE)

This register sets the enhanced ACK mode.

The BBEACKMODE register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 01H.

Figure 18 - 12 Enhanced ACK Mode Register (BBEACKMODE) Format

Address: 0006H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
BBEACK MODE	0	0	0	0	0	0	0	ENHACK MODEBIT

ENHACK MODEBIT	Enhanced ACK Mode bit
0	Enhanced ACK Mode function disabled
1	Enhanced ACK Mode

Caution Be sure to clear bits 1 to 7 to “0”.

(7) Transmit/receive status register 0 (BBTXRXST0)

This register stores the RF transmit/receive status.

It stores the CCA judge result in bit 0.

It stores the CRC judge result in bit 1. The CRC result corresponding to the save bank which is specified by the receive data save bank select bit is read out when reading.

It stores the CSMA-CA judge result in bit 2.

The transmit/receive operation completion judge result bit is used to store the result of transmit/receive operation sequence (CSMA-CA → Transmit → ACK receive → Retransmit → ACK receive, and so on) judge upon
completion. The result is NG (no good) when the ACK receive cannot be performed even if the sequences from the transmission of the number of setting times to the ACK receive are repeated.

The receive RAM bank 0 status bit and the receive RAM bank 1 status bit can be used as flags upon data capture for the respective receive RAM banks 0 and 1.

1 is automatically set upon completion of the data receive. Later on, they are cleared to 0 by software after reading the receive RAM data. Set 1 when no data is written because setting 0 enables writing. A receive overrun interrupt occurs when data is received again under the status that 1 is set to these bits resulting in the occurrence of write to each receive RAM.

It stores the pending bit data of the received ACK data in the receive pending bit upon completion of the reception of only the ACK data.

The receive RAM bank pointer bit can be used to check the receive RAM bank after the completion of receive. It indicates 1 after reset or initialization. This bit changes when the receive RAM bank is full or when the frame
receive is complete.

The BBTXRXST0 register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to 80H.

Figure 18 - 13 Transmit/Receive Status Register 0 (BBTXRXST0) Format

Address: 0007H After reset: 80H R/W[Note]

Symbol	7	6	5	4	3	2	1	0
BBTXRXST0	RCVRAMST	RCVPEND	RCVBANK1	RCVBANK0	TRNRCVSQC	CSMACA	CRC	CCA

RCVRAMST	Receive RAM bank pointer bit
0	Receive RAM bank 0
1	Receive RAM bank 1

RCVPEND	Receive pending bit
0	No pending
1	Pending exists

RCVBANK1	Receive RAM bank 1 status bit
0	Receive enabled
1	Receive data exists

RCVBANK0	Receive RAM bank 0 status bit
0	Receive enabled
1	Receive data exists

TRNRCVSQC	Transmit/receive operation completion judge result bit
0	OK
1	NG

CSMACA	CSMA-CA judge result bit
0	OK
1	NG

CRC	CRC judge result bit
0	OK
1	NG

CCA	CCA judge result bit
0	Channel clear
1	Channel busy

Note Bits 0 to 3, 6, and 7 are Read Only.

(8) Transmit/receive mode register 2 (BBTXRXMODE2)

This register sets the various types of the RF transmit/receive mode.

The automatic CRC disable bit can be used to select between to transmit with the CRC calculation result automatically attached upon transmit, and to transmit the data on the RAM.

The frame pending bit can be used to set with/without the frame pending upon ACK reply when the address matches on the first address side. Information on this bit is automatically included in the ACK replay data.

The frame pending status bit is used to display whether the reply is made with frame pending or not upon the automatic ACK reply. The display is updated at the same time of the occurrence of interrupt request upon completion of the transmission. The information on the frame pending upon reply by the save bank which is selected by the receive data save bank select bit is read upon reading because this bit left the result of the automatic ACK reply for every received save bank.

The enhanced ACK enable bit can be used to support the enhanced ACK reply function.

It can be used to set the number of retransmission times by using the retransmission counter when there is no ACK reply even if the automatic ACK receive mode is enabled. The value ranged from 0000b through 1000b can be set.

The BBTXRXMODE2 register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to 30H.

Figure 18 - 14 Transmit/Receive Mode Register 2 (BBTXRXMODE2) Format

Address: 0009H After reset: 30H R/W[Note]

Symbol	7	6	5	4	3	2	1	0
BBTXRXMODE2	RETRN3	RETRN2	RETRN1	RETRN0	ENHACKEN	FLMPENDST	FLMPEND	NOCRC

RETRN[3:0]	Retransmission counter (From 0000B up to 1000B can be set.)

ENHACKEN	Enhanced ACK Enable bit
0	Enhanced ACK disabled
1	Enhanced ACK enabled

FLMPENDST	Frame pending status bit
0	No frame pending
1	Frame pending exists

FLMPEND	Frame pending bit
0	No frame pending
1	Frame pending exists

NOCRC	Automatic CRC enable bit
0	Enable
1	Disable

Note Bit 2 is Read Only.

(9) Transmit/receive mode register 3 (BBTXRXMODE3)

This register sets the various types of the RF transmit/receive mode.

The address filter enable bit can be used to enable the address filter upon receiving.

When this bit is set to “1”, address filter on first address side is enabled.

The PAN coordinator bit can be used to set whether the PAN coordinator is set as one of the address filter conditions on first address side.

Reception is enabled by receive enable bit. The bit must be set to 1 in receive operation.

Receive level filter enable bit can be used to suspend reception and enable receive level filter interrupt when a communication error occurs such as no signal state during frame reception. The state enters to waiting reception again after the suspension through IDLE automatically. Receive completion interrupt does not occur because the suspension is done during frame reception. Destroy the receive frame when receive level filter interrupt occurs. An overwrite enable processing might be required depending on the destroyed timing. Then changing to enable reception of receive bank 0, 1 status bits for corresponding bank, or enabling receive RAM overwrite enable bit is required. Set this bit and receive level filter interrupt enable bit to “1” in initialization.

The receive data save bank select bit specifies the save bank to access the read of the receive-related data (other than the receive RAM).

The receive RAM overwrite enable bit can control the overwrite to the receive RAM. However, the receive data is not overwritten when write access to each receive RAM occurs when the above bit is 0 and receive RAM bank status bit is 1. The receive data is overwritten when write access to each receive RAM occurs when the receive RAM overwrite enable bit is 1 even if the receive RAM bank status bit is 1.

The address filter address extension bit can extend the first address and the second address in two ways.

The address filter general-purpose mode bit can set the address filter operation to the general-purpose mode.

The BBTXRXMODE3 register is set by the serial interface in 8-bit units.

Reset signal generation clears this register to 00H.

Figure 18 - 15 Transmit/Receive Mode Register 3 (BBTXRXMODE3) Format

Address: 000AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBTXRXMODE3	ADFGENMODE	ADFEXTEN	RCVOVERWREN	RCVBANKSEL	LVLFILEN	RXEN	PANCORD	ADRSFILEN

ADFGENMODE	Address filter general-purpose mode bit
0	The WiSUN mode
1	The general purpose mode

ADFEXTEN	Address filter address extension bit
0	PANID&Address expansion disabled
1	PANID&Address expansion enabled

RCVOVERWREN	Receive RAM overwrite enable bit
0	Overwrite disabled
1	Overwrite enabled

RCVBANKSEL	Receive data bank select bit
0	Receive data bank 0
1	Receive data bank 1

LVLFILEN	Receive level filter enable bit
0	Filter disabled
1	Filter enabled

RXEN	Receive enable bit
0	Receive disable
1	Receive enable

PANCORD	PAN coordinator bit
0	Not a PAN coordinator
1	PAN coordinator

ADRSFILEN	Address filter enable bit
0	Address filter disabled
1	Address filter enabled

(10) Transmit/receive status register 1 (BBTXRXST1)

The receive data save bank pointer bit can be used to check the bank which stores the data, for example frame length of the received frame. It indicates 1 after reset or initialization.

The BBTXRXST1 register is read by the serial interface in 8-bit units.

Reset signal generation clears this register to 02H.

Figure 18 - 16 Transmit/Receive Status Register 1 (BBTXRXST1) Format

Address: 000BH After reset: 02H R

Symbol	7	6	5	4	3	2	1	0
BBTXRXST1	0	0	0	0	0	0	RCVSTORET	X

RCVSTORET	Receive data save bank pointer bit
0	Receive data save bank 0
1	Receive data save bank 1

Caution “0“ is always read for bits 2 to 7.
Bit 0 is read as x (undefined).

(11) Transmit/receive control register (BBTXRXCON)

This register controls the RF transmit/receive.

Setting “1” to the receive trigger bit starts warm-up of the RF circuit and enters into the reception available state after 240 µs.

Setting “1” to the transmit trigger bit starts warm-up of the RF circuit and enters transmission after 350 µs.

Setting “1” to the CCA trigger bit starts warm-up of the RF circuit and enters CCA operation after 240 µs. Set value other than 0H to BEMIN bits to perform CCA.

Be sure to set these bits under the state they are IDLE. Note that these bits are automatically cleared to “0” upon completion of transmit/receive or of CCA. Stop the reception by using the RF communication stop bit when you want to stop the operation in the middle.

Do not stop transmission during the operation.

The automatic ACK receive mode bit can select whether to perform the automatic ACK receive operation.

The BBTXRXCON register is set by the serial interface in 8-bit units.

Reset signal generation clears this register to 00H.

Figure 18 - 17 Transmit/Receive Control Register (BBTXRXCON) Format

Address: 000CH After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	2	1	0
BBTXRXCON	0	0	0	0	ACKRCVEN	CCATRG	TRNTRG	RCVTRG

ACKRCVEN	Automatic ACK receive mode bit
0	Auto ACK receive disabled
1	Auto ACK receive enabled

CCATRG	CCA trigger bit
0	Nothing performed
1	CCA start

TRNTRG	Transmission trigger bit
0	Nothing performed
1	Transmission start

RCVTRG	Receive trigger bit
0	Nothing performed
1	Receive start

Note Bits 2 to 0 are Write Only.

Caution Be sure to clear bits 4 to 7 to “0”.

(12) CSMA control register 0 (BBCSMACON0)

This register controls the CSMA-CA operation.

Setting “1” to the automatic CSMA-CA start bit starts the CSMA-CA operation. Be sure to set this bit under the state it is IDLE. Note that these bits are automatically cleared to “0” upon completion of the CSMA-CA operation. In addition, setting “0” also disables writing.

Stop the CSMA-CA operation by using the RF communication stop bit when you want to stop the CSMA-CA operation.

Setting “1” to the transmit after automatic CSMA-CA bit automatically performs the transmission process if it's okay after the completion of the CSMA-CA operation.

Setting “1” to the reception during CSMA-CA function enable bit enables the reception during the wait time for the CCA operation in the CSMA-CA operation.

The back off period count stops during a frame reception.

The unicast frame bit sets the unicast frame value. Setting “1” to the unicast frame enable bit enables the function.

The BBCSMACON0 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears this register to 00H.

Figure 18 - 18 CSMA Control Register 0 (BBCSMACON0) Format

Address: 000DH After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	2	1	0
BBCSMACON0	0	0	0	UNICASTFRM	0	CSMARCVEN	CSMATRNST	CSMAST

UNICASTFRM	Unicast frame bit
0	Unicast frame value = 0
1	Unicast frame value = 1

CSMARCVEN	Reception during CSMA-CA function enable bit
0	Receive function disabled
1	Receive function enabled

CSMATRNST	Transmit after automatic CSMA-CA bit
0	Nothing performed
1	Transmit after CSMA-CA process

CSMAST	Automatic CSMA-CA start bit
0	Nothing performed
1	Automatic CSMA-CA start

Note Bit 0 is Write Only.

Caution Be sure to clear bits 3, 5 to 7 to “0”.

(13) CCA level threshold setting register (BBCCAVTH)

This register is used to set the threshold level to judge the CCA. It judges as busy for the CCA value which is the value set to this register or greater.The value is set as 2's complement. The setting unit is dBm (ex. "19EH" is "- 98dBm").

The BBCCAVTH register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to 0100H.

Figure 18 - 19 CCA Level Threshold Setting Register (BBCCAVTH) Format

Address: 000FH, 000EH After reset: 0100H R/W

Symbol	15	14	13	12	11	10	9	8
BBCCAVTH	0	0	0	0	0	0	0	CCAVTH8
	7	6	5	4	3	2	1	0
	CCAVTH7	CCAVTH6	CCAVTH5	CCAVTH4	CCAVTH3	CCAVTH2	CCAVTH1	CCAVTH0

CCAVTH[8:0]	CCA threshold level value

Caution Be sure to clear bits 9 to 15 to "0".

(14) Transmit/receive status register 2 (BBTXRXST2)

This register stores the information on the various types of the RF transmit/receive status.

The receive RAM bank flag bit is used to indicate the receive RAM bank upon start of the reception.

The receive data storage bank flag bit is used to indicate the reception storage bank upon start of the reception.

The BBTXRXST2 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 03H.

Figure 18 - 20 Transmit/Receive Status Register 2 (BBTXRXST2) Format

Address: 0010H After reset: 03H R

Symbol	7	6	5	4	3	2	1	0
BBTXRXST2	0	0	0	0	0	0	RCVSTOREFLG	RCVBANKFLG

RCVSTOREFLG	Receive data storage bank flag bit
0	Receive data storage bank 0
1	Receive data storage bank 1

RCVBANKFLG	Receive RAM bank flag bit
0	Receive RAM bank 0
1	Receive RAM bank 1

Caution “0” is always read for bits 2 to 7.

(15) Transmit/receive mode register 4 (BBTXRXMODE4)

This register sets the various types of the RF transmit/receive mode.

The CCA interrupt select bit can be used to select the CCA interrupt source between the time upon completion of the CCA sequence and the time upon completion of the CSMA-CS sequence.

The INT output polarity switch bit can be used to select the interrupt polarity of the interrupt output from the INTOUT pin.

Automatic receive with timeout after transmit enable bit enables automatic receive with timeout after transmit by automatic receive switch mode 0 enable bit. Byte receive interrupt and receive timeout interrupt can be switched by receive timeout interrupt switch bit.

The BBTXRXMODE4 register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to 01H.

Figure 18 - 21 Transmit/Receive Mode Register 4 (BBTXRXMODE4) Format

Address: 0011H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
BBTXRXMODE4	TIMEOUTINTSEL	TIMEOUTRCV	0	INTOUTSEL	0	0	0	CCAINTSEL

TIMEOUTINTSEL	Receive timeout interrupt switch bit
0	Byte receive interrupt
1	Receive timeout interrupt

TIMEOUTRCV	Automatic receive with timeout after transmit enable bit
0	Automatic normal reception mode
1	Automatic reception with timeout mode

INTOUTSEL	INTOUT output polarity switch bit
0	When with the interrupt request = INTOUT pin is at high level
1	When with the interrupt request = INTOUT pin is at low level

CCAINTSEL	CCA interrupt select bit
0	Upon completion of the CCA sequence
1	Upon completion of the CSMA-CA sequence

Caution Be sure to clear bit 1 to 3, 5 to “0”.

(16) CSMA control register 1 (BBCSMACON1)

This register controls the CSMA-CA operation.

The NB bit is used to set the macMaxCSMABackoff value. (Initial value is 0H)

The CW bit is used to set the CW value. (Initial value is 2H)

The BBCSMACON1 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 20H.

Figure 18 - 22 CSMA Control Register 1 (BBCSMACON1) Format

Address: 0012H After reset: 20H R/W

Symbol	7	6	5	4	3	2	1	0
BBCSMACON1	0	0	CW1	CW0	0	NB2	NB1	NB0

CW1, CW0	CW value

NB[2:0]	The macMaxCSMABackoff value

Caution Be sure to clear bits 3, 6 and 7 to “0”.

(17) CSMA control register 2 (BBCSMACON2)

This register controls the CSMA-CA operation.

The BEMAX bit is used to set the macMaxBE value. (Initial value is 5H) Set the greater value for the BEMAX bit than the BEMIN bit setting value.

The MacMinBE control bit can be used to set with/without the CCA operation when “000b” is set to macMinBE which is a parameter of automatic CSMA-CA.

The unicast frame enable bit can be used to set the operation by the unicast frame bit.

The BBCSMACON2 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 05H.

Figure 18 - 23 CSMA Control Register 2 (BBCSMACON2) Format

Address: 0013H After reset: 05H R/W

Symbol	7	6	5	4	3	2	1	0
BBCSMACON2	0	0	UNICASTFREN	MACMINBECON	BEMAX3	BEMAX2	BEMAX1	BEMAX0

UNICASTFREN	Unicast frame enable bit
0	Unicast frame disabled
1	Unicast frame enabled

MACMINBECON	MacMinBE control bit
0	Without the CCA operation
1	With the CCA operation

BEMAX[3:0]	MacMaxBE value

Caution Be sure to clear bits 6 and 7 to “0”.

(18) PAN identifier register 0 (BBPANID0)

This register is used to set the PAN identifier of the first address filter. It consists of 16 bits and is used to detect the match with a received PAN identifier.

The BBPANID0 register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to FFFFH.

Figure 18 - 24 PAN Identifier Register 0 (BBPANID0) Format

Address: 0015H, 0014H After reset: FFFFH R/W

Symbol	15	14	13	12	11	10	9	8
BBPANID0								

7	6	5	4	3	2	1	0

BBPANID0	PAN identifier

(19) Short address register 0 (BBSHORTAD0)

This register is used to set the short address of the first address filter. It consists of 16 bits and is used to detect the match with a received PAN identifier.

The BBSHORTAD0 register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to FFFFH.

Figure 18 - 25 Short Address Register 0 (BBSHORTAD0) Format

Address: 0017H, 0016H After reset: FFFFH R/W

Symbol	15	14	13	12	11	10	9	8
BBSHORTAD0								

7	6	5	4	3	2	1	0

BBSHORTAD0	Short address

(20) Extended address register 0 (BBEXTENDAD00 to BBEXTENDAD03)

These registers are used to set the extended address of the first address filter. They consist of 64 bits (16 bit timed by 4) and are used to detect the match with a received extended address.

The BBEXTENDAD00 to BBEXTENDAD03 registers are set by the serial interface in 8-bit units.

Reset signal generation clears these registers to 0000H.

Figure 18 - 26 Extended Address Register 0 (BBEXTENDAD00 to BBEXTENDAD03) Format

Address: 001FH, 001EH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBEXTENDAD03								
	7	6	5	4	3	2	1	0

BBEXTENDAD03	Extended address bits 63 to 48

Address: 001DH, 001CH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBEXTENDAD02								
	7	6	5	4	3	2	1	0

BBEXTENDAD02	Extended address bits 47 to 32

Address: 001BH, 001AH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBEXTENDAD01								
	7	6	5	4	3	2	1	0

BBEXTENDAD01	Extended address bits 31 to 16

Address: 0019H, 0018H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBEXTENDAD00								
	7	6	5	4	3	2	1	0

BBEXTENDAD00	Extended address bits 15 to 0

(21) Timer read registers (BBTIMEREAD0, BBTIMEREAD1)

These registers are used to read the current count value of the 32-bit timer. Read from the lower bytes when reading the timer count value. Then continue to read the upper bytes because the count value is latched when reading the 0020H address of LSB.

The BBTIMEREAD0 and BBTIMEREAD1 registers consist of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears these registers to 0000H.

Figure 18 - 27 Timer Read Register 0 (BBTIMEREAD0) Format

Address: 0021H, 0020H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
BBTIMEREAD0								
	7	6	5	4	3	2	1	0

BBTIMEREAD0	Lower bits of the 32-bit timer (bits 15 to 0)

Figure 18 - 28 Timer Read Register 1 (BBTIMEREAD1) Format

Address: 0023H, 0022H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
BBTIMEREAD1								
	7	6	5	4	3	2	1	0

BBTIMEREAD1	Upper bits of the 32-bit timer (bits 31 to 16)

(22) Timer compare registers 0 and 1
(BBTCOMP0REG0 to BBTCOMP2REG0, BBTCOMP0REG1 to BBTCOMP2REG1)

These registers are used to compare with the 32-bit timer. Each of them includes three channels to compare each of the channels with the 32-bit timer.

The BBTCOMP0REG0 to BBTCOMP2REG0 and BBTCOMP0REG1 to BBTCOMP2REG1 registers consist of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears these registers to 0000H.

Figure 18 - 29 Timer Compare Register 0 (BBTCOMP0REG0 to BBTCOMP2REG0) Format

Address: 002DH, 002CH (BBTCOMP2REG0), 0029H, 0028H (BBTCOMP1REG0), 0025H, 0024H (BBTCOMP0REG0)
After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBTCOMPnREG0 (n = 0 to 2)								
	7	6	5	4	3	2	1	0

BBTCOMPnREG0	Lower bits of the 32-bit compare (bits 15 to 0) (n = 0 to 2)

Figure 18 - 30 Timer Compare Register 1 (BBTCOMP0REG1 to BBTCOMP2REG1) Format

Address: 002FH, 002EH (BBTCOMP2REG1), 002BH, 002AH (BBTCOMP1REG1), 0027H, 0026H (BBTCOMP0REG1)
After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBTCOMPnREG1 (n = 0 to 2)								
	7	6	5	4	3	2	1	0

BBTCOMPnREG1	Upper bits of the 32-bit compare (bits 31 to 16) (n = 0 to 2)

(23) Time stamp registers 0 and 1 (BBTSTAMP0, BBTSTAMP1)

These registers are used to store the timer values upon the packet data receive start, upon the completion of the packet data reception, or upon completion of the packet data transmission. The respective time stamps of receive/transmit are stored. Therefore, the read value can be selected among the values of the receive start stamp, receive completion stamp, and the transmission stamp by using the stamp value read switch bit.

The timer count value upon completion of receive is automatically stored in the time stamp register while the stamp value is retained until the next packet receive completion. The time stamp value corresponding to the save bank which is specified by the receive data save bank select bit is read out when reading.

In addition, the timer count upon completion of the transmission is automatically stored in the time stamp register.

The stamp value is retained until the next packet transmission completion. However, the transmission time stamp value is not updated upon automatic ACK reply.

The BBTSTAMP0 and BBTSTAMP1 registers are read by the serial interface in 8-bit units.

Reset signal generation clears these registers to 0000H.

Figure 18 - 31 Time Stamp Register 0 (BBTSTAMP0) Format

Address: 0031H, 0030H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
BBTSTAMP0								
	7	6	5	4	3	2	1	0

BBTSTAMP0	Lower bits of the 32-bit timer (bits 15 to 0)

Figure 18 - 32 Time Stamp Register 1 (BBTSTAMP1) Format

Address: 0033H, 0032H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
BBTSTAMP1								
	7	6	5	4	3	2	1	0

BBTSTAMP1	Lower bits of the 32-bit timer (bits 31 to 16)

(24) Timer control register (BBTIMECON)

This register is used to control the timer in this transceiver.

The timer count enable bit controls the 32-bit timer count operation. Setting “1” enables the timer count. In addition, setting “0” stops the timer while the count value is initialized to 00000000H.

The RF transmit can be started when the compare 0 value and timer value match by using the COMP0 transmit trigger enable bit. Warm-up starts immediately after the match and starts transmit after 350 μs. Be sure to set these bit under the state they are IDLE.

The stamp timing switch bit can be used to select the stamp timing of the timer value. The value is updated at the timing of receive start regardless of the address filter function when selecting the receive start time.

The COMP0 trigger function select bit can be used to select the CSMA-CA for the start function when COMP0TRG is enabled.

The stamp value read switch bit can be used to select the time stamp register read value among the receive start stamp value, receive completion stamp value, and the transmission completion stamp value.

The count source switch bit can be used to select the count source for the timer count between the prescaler output and the data rate.

The BBTIMECON register is set by the serial interface in 8-bit units.

Reset signal generation clears this register to 00H.

Figure 18 - 33 Timer Control Register (BBTIMECON) Format

Address: 0034H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBTIMECON	0	CNTSRCSEL	STAMPRDSEL1	STAMPRDSEL0	COMP0TRGSEL	STAMPTIMSEL	COMP0TRG	TIMEEN

CNTSRCSEL	Count source switch bit
0	Prescaler output (1 μs)
1	Data rate

STAMPRDSEL1	STAMPRDSEL0	Stamp value read switch bit
0	0	Receive start stamp value
0	1	Receive completion stamp value
1	0	Transmit completion stamp value
1	1	

COMP0TRGSEL	COMP0 trigger function select bit
0	Transmit trigger
1	CSMA-CA trigger

STAMPTIMSEL	Stamp timing switch bit
0	Upon receive start
1	Upon interrupt of receive FL

COMP0TRG	COMP0 transmit trigger enable bit
0	Transmit trigger disabled
1	Transmit trigger enabled

TIMEEN	Timer count enable bit
0	Timer count stop
1	Timer count enable

Caution Be sure to clear bit 7 to “0”.

(25) Back off period register (BBBOFFPROD)

This register is used to control the back off period.

The back off period auto random enable bit is used to automatically generate the random value with setting the value set to the back off period bit as an initial value, and to set the back off period value in the CSMA-CA circuit.

Set the random value to the back-off period register 2, and then set “1” to the back off period auto random enable bit of the back off period register.

The BBBOFFPROD register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears this register to 00H.

Figure 18 - 34 Back off Period Register (BBBOFFPROD) Format

Address: 0035H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBBOFFPROD	0	0	0	0	0	0	0	BOFFPRODEN

BOFFPRODEN	Back-off period automatic random enable bit
0	Disable
1	Automatic random value generation enabled

Caution Be sure to clear bits 1 to 7 to “0”.

(26) Baseband interrupt source register 0 (BBINTREQ0)

This register is used to indicate the baseband interrupt source.

This register indicates that “1” is set to the interrupt source corresponding to each interrupt occurrence timing so that there is an interrupt request. When reading this register, only the bit from which 1 is read is cleared to 0. Note that perform the dummy read when you clear the bit because writing is disabled.

The BBINTREQ0 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.Note

Reset signal generation clears this register to 00H.

Note When reading this register, read 3 bytes of the baseband interrupt source registers 0 to 2 (BBINTREQ0 to BBINTREQ2) continuously (leave SEN internal pin low level).

Figure 18 - 35 Baseband Interrupt Source Register 0 (BBINTREQ0) Format

Address: 0036H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
BBINTREQ0	CCAINTREQ	X	TRN1INTREQ	TRN0INTREQ	TRNFININTREQ	TIM2INTREQ	TIM1INTREQ	TIM0INTREQ

CCAINTREQ	CCA completion interrupt source bit
0	No request
1	Request exists

TRN1INTREQ	Bank 1 transmit completion interrupt source bit
0	No request
1	Request exists

TRN0INTREQ	Bank 0 transmit completion interrupt source bit
0	No request
1	Request exists

TRNFININTREQ	Frame transmit completion interrupt source bit
0	No request
1	Request exists

TIM2INTREQ	Timer compare 2 interrupt source bit
0	No request
1	Request exists

TIM1INTREQ	Timer compare 1 interrupt source bit
0	No request
1	Request exists

TIM0INTREQ	Timer compare 0 interrupt source bit
0	No request
1	Request exists

Caution Bit 6 is X (undefined).

(27) Baseband interrupt source register 1 (BBINTREQ1)

This register is used to indicate the baseband interrupt source.

A bit of this register indicates that “1” is set to the interrupt source corresponding to each interrupt occurrence timing so that there is an interrupt request. When reading this register, only the bit from which 1 is read is cleared to 0. Note that perform the dummy read when you clear the bit because writing is disabled.

The BBINTREQ1 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.Note

Reset signal generation clears this register to 00H.

Note When reading this register, read 3 bytes of the baseband interrupt source registers 0 to 2 (BBINTREQ0 to BBINTREQ2) continuously (leave SEN internal pin low level).

<R>

Figure 18 - 36 Baseband Interrupt Source Register 1 (BBINTREQ1) Format

Address: 0037H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
BBINTREQ1	LVLFILINTREQ	MODESWINTREQ	ROVRINTREQ	ADRSINTREQ	RCVSTINTREQ	RCV1INTREQ	RCV0INTREQ	RCVFININTREQ

LVLFILINTREQ	Receive level filter interrupt source bit
0	No request
1	Request exists

MODESWINTREQ	Mode switch receive completion interrupt source bit
0	No request
1	Request exists

ROVRINTREQ	Receive overrun interrupt source bit
0	No request
1	Request exists

ADRSINTREQ	Address filter interrupt source bit
0	No request
1	Request exists

RCVSTINTREQ	Receive start interrupt source bit
0	No request
1	Request exists

RCV1INTREQ	Bank 1 receive completion interrupt source bit
0	No request
1	Request exists

RCV0INTREQ	Bank 0 receive completion interrupt source bit
0	No request
1	Request exists

RCVFININTREQ	Frame receive completion interrupt source bit
0	No request
1	Request exists

(28) Baseband interrupt source register 2 (BBINTREQ2)

This register is used to indicate the baseband interrupt source.

This register indicates that “1” is set to the interrupt source corresponding to each interrupt occurrence timing so that there is an interrupt request. When reading this register, only the bit from which 1 is read is cleared to 0. Note that perform the dummy read when you clear the bit because writing is disabled.

The BBINTREQ2 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.Note

Reset signal generation clears this register to 00H.

Note When reading this register, read 3 bytes of the baseband interrupt source registers 0 to 2 (BBINTREQ0 to BBINTREQ2) continuously (leave SEN internal pin low level).

<R>

Figure 18 - 37 Baseband Interrupt Source Register 2 (BBINTREQ2) Format

Address: 0038H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
BBINTREQ2	X	BYTERCVINTREQ	FLINTREQ	RCVCUNTINTREQ	X	X	X	X

BYTERCVINTREQ	Byte receive completion interrupt source bit
0	No request
1	Request exists

FLINTREQ	Frame length interrupt source bit
0	No request
1	Request exists

RCVCUNTINTREQ	Number of received byte interrupt source bit
0	No request
1	Request exists

Caution Bit 0 to 3 , 7 are X (undefined).

(29) Baseband interrupt enable register 0 (BBINTEN0)

This register is used to enable the baseband interrupt.

This register enables the interrupt output from the INTOUT pin upon generation of each interrupt. Set “1” to the corresponding interrupt enable bit when you want to enable the interrupt output.

The BBINTEN0 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears this register to 00H.

Figure 18 - 38 Baseband Interrupt Enable Register 0 (BBINTEN0) Format

Address: 0039H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBINTEN0	CCAINTEN	X	TRN1INTEN	TRN0INTEN	TRNFININTEN	TIM2INTEN	TIM1INTEN	TIM0INTEN

CCAINTEN	CCA completion interrupt enable bit
0	Disabled
1	Enabled

TRN1INTEN	Bank 1 transmit completion interrupt enable bit
0	Disabled
1	Enabled

TRN0INTEN	Bank 0 transmit completion interrupt enable bit
0	Disabled
1	Enabled

TRNFININTEN	Frame transmit completion interrupt cause bit
0	Disabled
1	Enabled

TIM2INTEN	Timer compare 2 interrupt enable bit
0	Disabled
1	Enabled

TIM1INTEN	Timer compare 1 interrupt enable bit
0	Disabled
1	Enabled

TIM0INTEN	Timer compare 0 interrupt enable bit
0	Disabled
1	Enabled

Caution Be sure to clear bit 6 to "0". It is read as x (undefined).

(30) Baseband interrupt enable register 1 (BBINTEN1)

This register is used to enable the baseband interrupt.

This register enables the interrupt output from the INTOUT pin upon generation of each interrupt. Set “1” to the corresponding interrupt enable bit when you want to enable the interrupt output.

Set receive level filter interrupt enable bit (LVLFILINTEN) and receive level filter enable bit (LVLFILEN) to <R> "1" in initialization.

The BBINTEN1 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears this register to 00H.

<R> **Figure 18 - 39 Baseband Interrupt Enable Register 1 (BBINTEN1) Format**

Address: 003AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBINTEN1	LVLFILINTEN	MODESWINTEN	ROVRINTEN	ADRSINTEN	RCVSTINTEN	RCV1INTEN	RCV0INTEN	RCVFININTEN

LVLFILINTEN	Receive level filter interrupt enable bit
0	Disabled
1	Enabled

MODESWINTEN	Mode switch receive completion interrupt enable bit
0	Disabled
1	Enabled

ROVRINTEN	Receive Overrun interrupt enable bit
0	Disabled
1	Enabled

ADRSINTEN	Address filter interrupt enable bit
0	Disabled
1	Enabled

RCVSTINTEN	Receive start interrupt enable bit
0	Disabled
1	Enabled

RCV1INTEN	Bank 1 receive completion interrupt enable bit
0	Disabled
1	Enabled

RCV0INTEN	Bank 0 receive completion interrupt enable bit
0	Disabled
1	Enabled

RCVFININTEN	Frame receive completion interrupt enable bit
0	Disabled
1	Enabled

(31) Baseband interrupt enable register 2 (BBINTEN2)

This register is used to enable the baseband interrupt.

This register enables the interrupt output from the INTOUT pin upon generation of each interrupt. Set “1” to the corresponding interrupt enable bit when you want to enable the interrupt output.

<R>

The BBINTEN2 register is set by the serial interface in 8-bit units.

Reset signal generation clears this register to 00H.

<R>

Figure 18 - 40 Baseband Interrupt Enable Register 2 (BBINTEN2) Format

Address: 003BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBINTEN2	X	BYTERCVINTEN	FLINTEN	RCVCUNTINTEN	X	X	X	X

BYTERCVINTEN	Byte receive completion interrupt enable bit
0	Disabled
1	Enabled

FLINTEN	Frame length interrupt enable bit
0	Disabled
1	Enabled

RCVCUNTINTEN	Number of received byte interrupt enable bit
0	Disabled
1	Enabled

Caution Be sure to clear bits 0 to 3 , 7 to “0”. These are read as X (undefined).

(32) CSMA control register 3 (BBCSMACON3)

This register controls the CSMA-CA operation.

The BEMIN bit is used to set the macMinBE value. (Initial value is 3H.)

Set the smaller value for the BEMIN bit than the BEMAX bit setting value.

Set value other than 0H to these bits to perform CCA.

The BBCSMACON3 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 03H.

Figure 18 - 41 CSMA Control Register 3 (BBCSMACON3) Format

Address: 003EH After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
BBCSMACON3	0	0	0	0	BEMIN3	BEMIN2	BEMIN1	BEMIN0

BEMIN[3:0]	BEMIN bit (Sets the macMinBE value.)

Caution Be sure to clear bits 4 to 7 to “0”.

(33) ACK counter compare registers 0 and 1 (ACKCOMP0, ACKCOMP1)

These registers are used to set each of the timings upon automatic ACK reply mode.

The ACKCOMP0 and ACKCOMP1 registers consist of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets ACKCOMP0 to 0014H and sets ACKCOMP1 to 000EH.

Figure 18 - 42 ACK Counter Compare Register 0 (ACKCOMP0) Format

Address: 0041H, 0040H After reset: 0014H R/W

Symbol	15	14	13	12	11	10	9	8
ACKCOMP0								
	7	6	5	4	3	2	1	0

ACKCOMP0	ACK counter compare 0 value

Figure 18 - 43 ACK Counter Compare Register 1 (ACKCOMP1) Format

Address: 0043H, 0042H After reset: 000EH R/W

Symbol	15	14	13	12	11	10	9	8
ACKCOMP1								
	7	6	5	4	3	2	1	0

ACKCOMP1	ACK counter compare 1 value

(34) ACK counter initial value register (ACKINI)

This register is used to set each of the timings upon automatic ACK reply mode.

The ACKINI register consists of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears this register to 0000H.

Figure 18 - 44 ACK Counter Initial Value Register (ACKINI) Format

Address: 005FH, 005EH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
ACKINI								

7	6	5	4	3	2	1	0

ACKINI	ACK counter initial value

(Setting example) In the BEACON mode

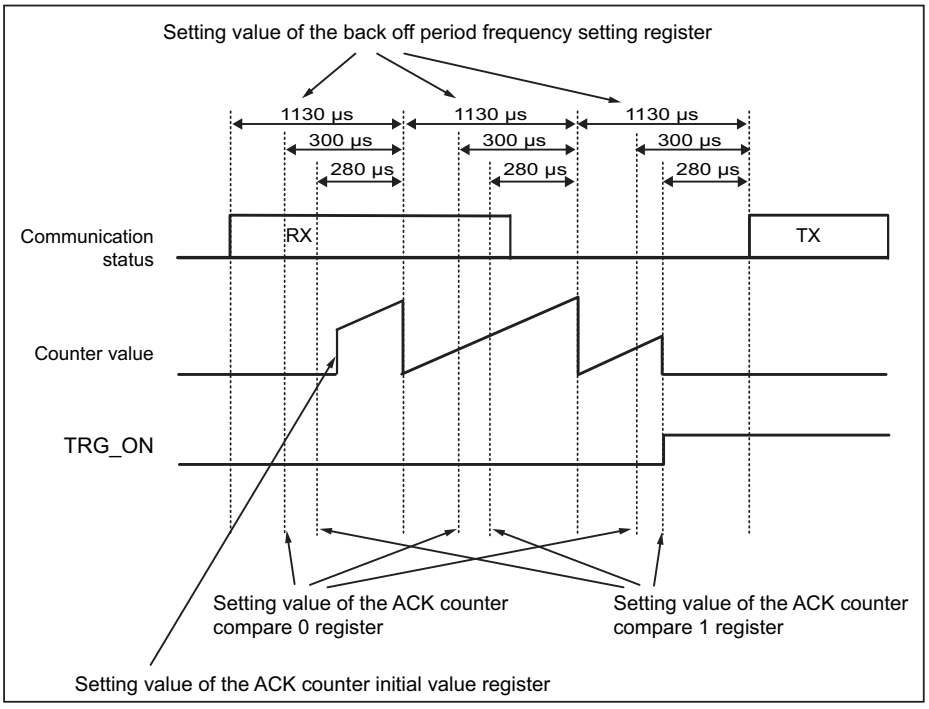
The ACK counter compare 0 register, ACK counter compare 1 register, ACK counter initial value register, and back off cycle setting register are used to control the ACK reply timing in the BEACON mode.

ACK counter compare 0 register: Set the time subtracted 300 μs from the back off cycle.
Initial value: 0014H = 20 symbols (Set value: 1H = 1 symbol)

ACK counter compare 1 register: Set the time subtracted 144 μs from the back off cycle.
Initial value: 000EH = 15 symbols (Set value: 1H = 1 symbol)

ACK counter initial value register: Set the time until the baseband block recognizes the receive start.
Total time of preamble plus SFD plus FL.
Initial value: 0000H = 0 symbols (Set value: 1H = 1 symbol)

Figure 18 - 45 Count Operation in the BEACON Mode



(35) Receive level threshold setting register (BBLVLVTH)

This register is used to set the threshold value of the receive level.

The value is set as 2's complement The setting unit is dBm (ex. ”19EH” is ”-98dBm”).

Reception is done for level beyond the setting value.

The minimum setting value is recommended for the data rate and value of RSSI/CCA result register (BBRSSICCARSLT). For details, see 18.4.4 (5) RSSI/CCA result register (BBRSSICCARSLT).

The BBLVLVTH register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to 0100H.

Figure 18 - 46 Receive Level Threshold Setting Register (BBLVLVTH) Format

Address: 0045H, 0044H After reset: 0100H R/W

Symbol	15	14	13	12	11	10	9	8
BBLVLVTH	0	0	0	0	0	0	0	LVLVTHT8
	7	6	5	4	3	2	1	0
	LVLVTHT7	LVLVTHT6	LVLVTHT5	LVLVTHT4	LVLVTHT3	LVLVTHT2	LVLVTHT1	LVLVTHT0

<R>

LVLVTHT[8:0]	Threshold value of the receive level

Caution Be sure to clear bits 9 to 15 to “0”.

(36) ACK reply time setting register (ACKRTNTIM)

This register is used to set the ACK reply time in the non-BEACON mode.

The ACKRTNTIM register consists of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 0002H.

Figure 18 - 47 ACK Reply Time Setting Register (ACKRTNTIM) Format

Address: 0047H, 0046H After reset: 0002H R/W

Symbol	15	14	13	12	11	10	9	8
ACKRTNTIM								

7	6	5	4	3	2	1	0

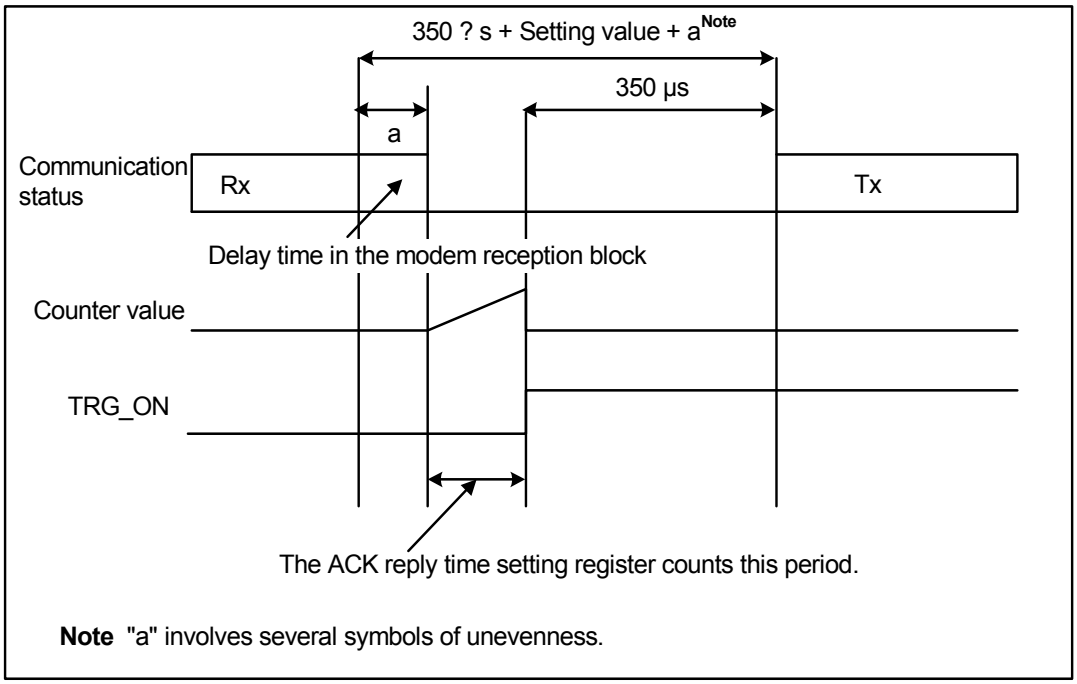
ACKRTNTIM	ACK reply time in the non-BEACON mode

(Setting example) In the non-BEACON mode

Only the ACK counter compare 3 register is used to set the time from the completion of packet receive to the TRG_ON time. Initial value: 0002H = 2 symbols (Set value: 1H = 1 symbol)

Counter stops and H is set to the TRG_ON signal when the 10-bit timer for the ACK reply mode starts from 000H upon completion of receive and it match with the ACK reply time setting register value in the non-BEACON mode.

Figure 18 - 48 Count Operation in the Non-BEACON Mode



Note "a" involves several symbols of unevenness.

(37) Automatic receive switch compare register (AUTORCVCNT)

This register is used to set the time until the TRG_ON signal is set to the high level for the automatic receive switch after the completion of transmit/receive when the mode is set to the automatic receive switch mode.

Initial value: 000AH = 10 symbols (Set value: 1H = 1 symbol)

The AUTORCVCNT register consists of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 000AH.

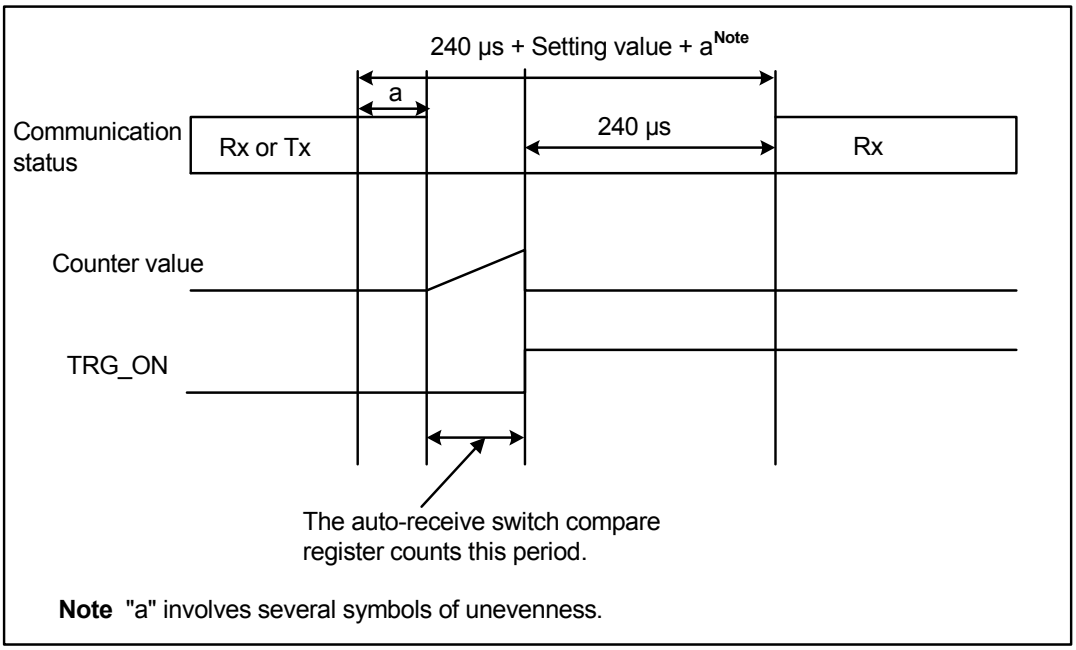
Figure 18 - 49 Automatic Receive Switch Compare Register (AUTORCVCNT) Format

Address: 0049H, 0048H After reset: 000AH R/W

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

AUTORCVCNT	Time until the TRG_ON signal is set to the high level

Figure 18 - 50 Automatic Receive Switch Compare Register Count Operation



Note "a" involves several symbols of unevenness.

(38) Back off cycle register (BOFFPERIOD)

This register is used to set the back off cycle. Initial value: 0071H= 113 symbols (Set value: 1H = 1 symbol)

The BOFFPERIOD register consists of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 0071H.

Figure 18 - 51 Back-off Cycle Register (BOFFPERIOD) Format

Address: 004BH, 004AH After reset: 0071H R/W

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

BOFFPERIOD	Back off cycle setting value

(39) CSMA-CA end count register (CSMAENDCOUNT)

This register is used to set the time until the wait status after the completion of CCA when the automatic CSMA-CA sequence is used.

Initial value is 0080H = 128 μs. (Set value: 1H =1 μs)

The CSMAENDCOUNT register consists of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 0080H.

Figure 18 - 52 CSMA-CA End Count Register (CSMAENDCOUNT) Format

Address: 004DH, 004CH After reset: 0080H R/W

Symbol	15	14	13	12	11	10	9	8
CSMAENDCOUNT								

7	6	5	4	3	2	1	0

CSMAENDCOUNT	Time until the wait status after completion of the CCA

(40) CSMA-CA start count register (CSMASTACOUNT)

This register is used to set the time of warm-up until the start of CCA when the automatic CSMA-CA sequence is used. Set the minimum number of symbols which is greater than 144 μs. Initial value: 000EH = 15 symbols (Set value: 1H = 1 symbol)

The CSMASTACOUNT register consists of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 000EH.

Figure 18 - 53 CSMA-CA Start Count Register (CSMASTACOUNT) Format

Address: 004FH, 004EH After reset: 000EH R/W

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

CSMASTACOUNT	CCA start warm-up time

(41) Communication status register 1 (COMSTATE1)

This register is used to confirm each state in communication.

Transmitting status bit shows transmit state.

CCA status bit shows CCA state.

Receiving frame status bit shows frame receive state.

Figure 18 - 54 Communication status register 1 (COMSTATE1) Format

Address: 0066H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
COMSTATE1	x	x	x	x	FRCVSTATE	CCASTATE	TRNSTATE	x

FRCVSTATE	Receiving frame status bit
0	Unreceived
1	Receiving

CCASTATE	CCA status bit
0	Not CCA
1	CCA

TRNSTATE	Transmitting status bit
0	Untransmitted
1	Transmitting

Caution Bits 0, 4 to 7 are read as x (undefined).

(42) Communication status register 2 (COMSTATE2)

This register is used to confirm each state in communication.

ACK reply status bit shows ACK reply state.

Figure 18 - 55 Communication status register 2 (COMSTATE2) Format

Address: 0067H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
COMSTATE2	x	x	x	x	x	x	x	ACKSTATE

ACKSTATE	ACK reply status bit
0	Unreplied ACK
1	Replying ACK

Caution Bits 1 to 7 are read as x (undefined)..

(43) Estimate control register (BBEVAREG)

This register is used to set the evaluation mode which is required when obtaining the certification of conformance to technical standards.

Setting “1” to both continuous transmission mode and transmit trigger bit enters into the continuous transmission mode. In the continuous transmission mode, the data of the number subtracted CRC byte count from the value

set in the transmission frame length is repetitively transmitted. Note that the transmission data is the value which is written in the transmit RAM.Reset by RFRESETB pin after stopping the continuous transmission mode.

The non-modulation switch bit can switch between the modulation signal and non-modulation signal.

Setting “1” to both continuous receive mode bit and receive trigger bit enters into the continuous receive mode. In the continuous receive mode, the completion of the data reception cannot enter into the IDLE state. The receive state is left as it is.

Setting “1” to both special transmission mode and transmit trigger bit can transmit frames regardless of the IEEE802.15.4 standard.

Setting “1” to both special receive mode bit and receive trigger bit can receive frames regardless of the IEEE802.15.4 standard.

The BBEVAREG register is set by the serial interface in 8-bit units.

Reset signal generation clears this register to 00H.

Figure 18 - 56 Estimate Control Register 1 (BBEVAREG) Format

Address: 0068H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBEVAREG	0	0	0	NOSPECRX	NOSPECTX	CONTRX	NOMOD	CONTTX

NOSPECRX	Special receive mode bit
0	Normal operation
1	Special receive operation

NOSPECTX	Special transmit operation
0	Normal operation
1	Special transmit operation

CONTRX	Continuous receive mode bit
0	Normal operation
1	Continuous receive operation

NOMOD	Non-modulation switch bit
0	Modulation signal
1	Non-modulation signal

CONTTX	Continuous transmit mode bit
0	Normal operation
1	Continuous transmit operation

Caution Be sure to clear bits 7 to 5 to “0”.

Supplemental explanation 1. Special transmit mode

- Preamble part
  - Transmits the pattern which is set in the preamble setting register. (1 byte or 2 bytes)
  - Repetitively transmits the number of bytes which is set by the number-of-preamble-byte setting bit.
  - Transmits the byte length which is set in the preamble setting register.
- SFD part
  - Transmits the pattern which is set in the SFD setting register. (1 byte, 2, or 4 bytes)
  - Transmits the number of bytes which is set by the number-of-SFD-byte setting bit.
- PHR part
  - Transmits the 16-bit data which is set in the mode switch frame transmit register.
    Be sure to set bits 0 and 4 of the mode switch frame transmit register to “0”.
  - Transmits data regardless of the MODESW bit value.
  - Transmits data regardless of the transmit frame length register setting value.
- Transmit data part
  - Transmits data which is set in the transmit RAM from the 0200H address.
  - Transmits data of the 0200H address again after the transmission of the data of the 02FFH address in the transmit RAM.
  - Generates the bank transmit completion interrupt upon every completion of transmitting the transmit RAM bank.
- CRC part
  - The automatic CRC function does not operate.
  - Perform the CRC calculation by using software in order to add the CRC data to the final data part.
- End of transmission
  - Terminate the transmission by using the RF communication stop bit.
    Be sure to reset by RFRESETB pin after stopping.

Supplemental explanation 2. Special receive mode

- Preamble part
  - Operates to receive the pattern which is set in the preamble setting register as a preamble pattern. (1 byte or 2 bytes)
  - The number of bytes which is set by the number-of-preamble-byte setting bit is defined as a repetitive pattern.
- SFD part
  - The pattern which is set in the SFD setting register is defined as the source data for checking the SFD match. (1 byte, 2, or 4 bytes)
  - The number of bytes which is set by the number-of-SFD-byte setting bit is defined as the source data for checking the SFD match.
- PHR part
  - Stores the received PHR part (the 2 bytes of data immediately after the SFD) to the mode switch frame receive register.
    The bit 0 (corresponds to MODESW bit) and bit 4 (corresponds to DW bit) of the first byte of PHR can be received only “0s”.
  - Receives data regardless of the MODESW bit value.
  - Ignores the received frame length.
- Receive data part
  - Stores the receive data part (the 3rd byte and subsequent data immediately after the SFD) to the receive RAM.
  - Generates the bank receive completion interrupt upon every completion of receiving the receive RAM bank.
  - Generates the number-of-received-byte interrupt when receiving the number of bytes which is set in the-number-of-receive-byte interrupt compare register.
  - Enables the generation of the receive byte interrupt for every single bit of the receive data after the SFD.
  - The number of bytes stored in the receive RAM can be checked by the receive data count register.
- CRC part
  - The automatic CRC function does not operate.
  - Perform the CRC calculation by using software in order to check the CRC data when it is added to the final data part.
- End of reception
  - Terminate the reception by using the RF communication stop bit.

(44) Back off period register 2 (BBBOFFPROD2)

This register is used to set the random value of the back off period when performing the CSMA-CA by using the back off period bits 0 to 7. Set the random value to back off period register 2, and then set “1” to the back off period auto-random enable bit of the back off period register.

The BBBOFFPROD2 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears this register to 00H.

Figure 18 - 57 Back off Period Register2 (BBBOFFPROD2) Format

Address: 0069H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBBOFFPROD2	BOFFPROD7	BOFFPROD6	BOFFPROD5	BOFFPROD4	BOFFPROD3	BOFFPROD2	BOFFPROD1	BOFFPROD0

BOFFPROD[7:0]	Back off period bit value

(45) Communication status register 3 (COMSTATE3)

This register is used to display the communication status.

The number of retransmissions can be read when the auto-ACK receive mode is enabled by the number-of-retransmission read bit.

The number of times finally performing the CCA can be read by using the number-of-CCA-time read bit.

The COMSTATE3 register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears this register to 00H.

Figure 18 - 58 Communication Status Register 3 (COMSTATE3) Format

Address: 006FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
COMSTATE3	0	CCARD			RETRNRD			

CCARD	The number of CCA times which is finally performed (3 bits)

RETRNRD	The number of retransmissions (4 bits) when the auto-ACK receive mode is enabled.

Caution “0” is always read for bit 7.

(46) ACK receive wait time setting register (ACKRCVWIT)

This register is used to set the ACK receive wait time after the data transmission. Retransmits data when there is no ACK reply even if it waits for the specified time. Initial value: 0300H = 768 symbols. (Set value: 1H = 1 symbol)

The ACKRCVWIT register consists of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 0300H.

Figure 18 - 59 ACK Receive Wait Time Setting Register (ACKRCVWIT) Format

Address: 0071H, 0070H After reset: 0300H R/W

Symbol	15	14	13	12	11	10	9	8
ACKRCVWIT								

7	6	5	4	3	2	1	0

ACKRCVWIT	ACK receive wait time

(47) Retransmission start compare register (RETRNWUP)

This register is used to set the time to wait until the power supply is turned on again when performing retransmission.

Initial value: 0004H= 4 symbols (Set value: 1H = 1 symbol)

The RETRNWUP register consists of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 0004H.

Figure 18 - 60 Retransmission Start Compare Register (RETRNWUP) Format

Address: 0073H, 0072H After reset: 0004H R/W

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

RETRNWUP	Time to wait until the power-on again

(48) ANTSW output timing setting register (BBANTSWTIMG)

This register is used to set the timing for the ANTSW pin output.

The time to set the ANTSW pin output to high level can be set after the setting of “1” to the transmission trigger bit.

The value ranged from 001H to 154H can be set while the initial value is 072H (The set value is 1H = approx.1 μs).

Do not set the values other than those ranged from 001H (Approx. 1 μs) to 154H (Approx.340 μs).

The BBANTSWTIMG register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to 0072H.

Figure 18 - 61 ANTSW Output Timing Setting Register (BBANTSWTIMG) Format

Address: 007BH, 007AH After reset: 0072H R/W

Symbol	15	14	13	12	11	10	9	8
BBANTSWTIMG	0	0	0	0	0	0		

7	6	5	4	3	2	1	0

BBANTSWTIMG	Time till the ANTSW pin output is set to high level

Caution Be sure to clear bits 10 to 15 to “0”.

(49) Receive frame length register (BBRXFLEN)

This register is used to store the frame length value upon reception. The value is stores at the timing when the packet data reception is started. The value is retained until the start of the next packet data reception. However, the register is updated when the address match is recognized when the address filter is enabled.

The reading value is corresponding save bank specified by receive data save bank select bit.

The BBRXFLEN register is read by the serial interface in 8-bit units.

Reset signal generation clears this register to 0000H.

Figure 18 - 62 Receive Frame Length Register (BBRXFLEN) Format

Address: 00A1H, 00A0H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
BBRXFLEN	0	0	0	0	0	RCVFLEN10	RCVFLEN9	RCVFLEN8

7	6	5	4	3	2	1	0
RCVFLEN7	RCVFLEN6	RCVFLEN5	RCVFLEN4	RCVFLEN3	RCVFLEN2	RCVFLEN1	RCVFLEN0

RCVFLEN[10:0]	The frame length value upon receive

Caution “0” is always read for bits 11 to 15.

(50) Receive data counter register (BBRXCOUNT)

This register is used to indicate the receive data counter value upon reception. It enables the checking of how many bytes of receive data are currently stored to the receive RAM. The value is cleared to “0” when the packet receive ends.

The BBRXCOUNT register consists of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears this register to 0000H.

Figure 18 - 63 Receive Data Counter Register (BBRXCOUNT) Format

Address: 00A3H, 00A2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBRXCOUNT	0	0	0	0	0	RXCOUNT10	RXCOUNT9	RXCOUNT8
	7	6	5	4	3	2	1	0
	RXCOUNT7	RXCOUNT6	RXCOUNT5	RXCOUNT4	RXCOUNT3	RXCOUNT2	RXCOUNT1	RXCOUNT0

RXCOUNT[10:0]	Receive data counter value

Caution Be sure to clear bits 11 to 15 to “0”.

(51) Transmit frame length register (BBTXFLEN)

This register is used to set the frame length value upon transmission. Set the total value of payload data length and the CRC length (in 2 or 4 bytes) for the frame length value. Transmits data regardless of the transmit frame length value when the ACK auto-reply function is enabled to automatically reply with ACK.

The BBTXFLEN register consists of 16 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation clears this register to 0000H.

Figure 18 - 64 Transmit Frame Length Register (BBTXFLEN) Format

Address: 00A5H, 00A4H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBTXFLEN	0	0	0	0	0	TRNFLEN10	TRNFLEN9	TRNFLEN8
	7	6	5	4	3	2	1	0
	TRNFLEN7	TRNFLEN6	TRNFLEN5	TRNFLEN4	TRNFLEN3	TRNFLEN2	TRNFLEN1	TRNFLEN0

TRNFLEN[10:0]	The frame length value upon transmission

Caution Be sure to clear bits 11 to 15 to “0”.

(52) Preamble length setting register (BBPAMBL)

This register is used to set the preamble length upon transmission. Set value over values shown in Table 18 - 2 and Table 18 - 3.

The BBPAMBL register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to 0004H.

Figure 18 - 65 Preamble Length Setting Register (BBPAMBL) Format

Address: 00A7H, 00A6H After reset: 0004H R/W

Symbol	15	14	13	12	11	10	9	8
BBPAMBL	0	0	0	0	0	0	PAMBLLEN9	PAMBLLEN8
	7	6	5	4	3	2	1	0
	PAMBLLEN7	PAMBLLEN6	PAMBLLEN5	PAMBLLEN4	PAMBLLEN3	PAMBLLEN2	PAMBLLEN1	PAMBLLEN0

PAMBLLEN[9:0]	Preamble length setting value

Caution Be sure to clear bits 10 to 15 to “0”.

(53) Frequency setting register (BBFREQ)

This register is used to set frequency. This register consists of 30 bits.

The value ranged from 100 MHz to 1000 MHz can be set in 1 Hz step. Initial value: 36FC3BA0H = 922.5 MHz.

BBFREQ register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 36FC3BA0H.

Figure 18 - 66 Frequency Setting Register (BBFREQ) Format

Address: 00ABH to 00A8H After reset: 36FC3BA0H R/W

Symbol	31	30	29	28	27	26	25	24
BBFREQ	0	0	FREQ29	FREQ28	FREQ27	FREQ26	FREQ25	FREQ24
	23	22	21	20	19	18	17	16
	FREQ23	FREQ22	FREQ21	FREQ20	FREQ19	FREQ18	FREQ17	FREQ16
	15	14	13	12	11	10	9	8
	FREQ15	FREQ14	FREQ13	FREQ12	FREQ11	FREQ10	FREQ9	FREQ8
	7	6	5	4	3	2	1	0
	FREQ7	FREQ6	FREQ5	FREQ4	FREQ3	FREQ2	FREQ1	FREQ0

FREQ[29:0]	Frequency setting value

Caution Be sure to clear bits 30 and 31 to “0”.

For details, see 18.7.2 (2) RF frequency setting for transmission and 18.7.2 (3) RF frequency setting for reception.

(54) Symbol rate setting register (BBSYMBLRATE)

This register is used to set the symbol rate. This register consists of 24 bits.

Table 18 - 8 shows the symbol rate setting examples. The symbol rate is set by using the calculation below.

BBSYMBLRATE register is set via serial interface in 8-bit units.

Symbol rate = (f_{Xin}/2)/(value of symbol rate setting bits)

BBSYMBLRATE register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 0000F0H.

Figure 18 - 67 Symbol Rate Setting Register (BBSYMBLRATE) Format

Address: 00AEH to 00ACH After reset: 0000F0H R/W

Symbol	23	22	21	20	19	18	17	16
BBSYMBLRATE	SYMBLRATE23	SYMBLRATE22	SYMBLRATE21	SYMBLRATE20	SYMBLRATE19	SYMBLRATE18	SYMBLRATE17	SYMBLRATE16
	15	14	13	12	11	10	9	8
	SYMBLRATE15	SYMBLRATE14	SYMBLRATE13	SYMBLRATE12	SYMBLRATE11	SYMBLRATE10	SYMBLRATE9	SYMBLRATE8
	7	6	5	4	3	2	1	0
	SYMBLRATE7	SYMBLRATE6	SYMBLRATE5	SYMBLRATE4	SYMBLRATE3	SYMBLRATE2	SYMBLRATE1	SYMBLRATE0

SYMBLRATE[23:0]	Symbol rate setting bit value

Table 18 - 8 Symbol Setting Comparison Table

Symbol Rate Setting Bit Value	Symbol Rate
000050H	300 ksps
000078H	200 ksps
0000A0H	150 ksps
0000F0H	100 ksps
0001E0H	50 ksps
000258H	40 ksps
0004B0H	20 ksps
000960H	10 ksps

(55) SUBG control register (BBSUBGCON)

This register is used to control the SubGHz transceiver.

The FEC enable bit for reception is used to enable FEC (CODE) of reception.

The FEC mode switch bit is used to switch the FEC encode mode.

The number-of-CRC-bit switch bit is used to switch the number of bits for the CRC calculation process upon transmission. Note that the number of the CRC bits of the receive frame is used for those upon auto-ACK reply.

The DW enable bit is used to enable the data whitening upon transmission.

The PHR transmit bit 1 is used to transmit bit 1 of the PHR upon transmission.

The PHR transmit bit 2 is used to transmit bit 2 of the PHR upon transmission.

The interleaving enable bit is used to enable the interleaving. Set INTERLEAVEEN bit to 1 to use FEC.

The BBSUBGCON register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to 0CH.

Figure 18 - 68 SUBG Control Register (BBSUBGCON) Format

Address: 00B0H After reset: 0CH R/W

Symbol	7	6	5	4	3	2	1	0
BBSUBGCON	0	INTERLEAVEEN	PHRTX2	PHRTX1	DWEN	CRCBIT	FECMODE	FECENRX

INTERLEAVEEN	Interleaving enable bit
0	Interleaving disabled
1	Interleaving enabled

PHRTX2	PHR transmission bit 2 value

PHRTX1	PHR transmission bit 1 value

DWEN	DW enable bit
0	DW disabled
1	DW enabled

CRCBIT	Number-of-CRC-bit switch bit
0	32-bit CRC
1	16-bit CRC

FECMODE	FEC mode switch bit
0	NRNSC encoder
1	RSC encoder

FECENRX	FEC enable bit for reception
0	FEC disabled
1	FEC enabled

Caution Be sure to clear bit 7 to "0".

(56) Modulation method setting register (BBMODSET)

This register is used to set the modulation method.

The modulation method setting bit is used to set the modulation method.

The modulation index setting bit is used to set the modulation index. Table 18 - 9 shows the setting value setting examples for the modulation index setting bit.

The BBMODSET register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 42H.

Figure 18 - 69 Modulation Method Setting Register (BBMODSET) Format

Address: 00B1H After reset: 42H R/W

Symbol	7	6	5	4	3	2	1	0
BBMODSET	MODINDEX1	MODINDEX0	0	0	0	0	MODSET1	MODSET0

MODINDEX1	MODINDEX0	Modulation index setting bit
0	0	0.5
0	1	1.0
1	0	0.33
1	1	Setting prohibited

MODSET1	MODSET0	Modulation method setting bit
0	0	2FSK
0	1	4FSK
1	0	2GFSK (Initial value)
1	1	4GFSK

Caution Be sure to clear bits 2 to 5 to "0".

Table 18 - 9 Modulation Index Setting Bit Setting Examples

MODINDEX1	MODINDEX0	Data rate	Modulation method	Modulation index	Deviation
0	0	50 kbps	2FSK/GFSK	0.5	±12.5 kHz
0	0	100 kbps	2FSK/GFSK	0.5	±25 kHz
0	1	50 kbps	2FSK/GFSK	1.0	±25 kHz
0	1	100 kbps	2FSK/GFSK	1.0	±50 kHz
1	0	200 kbps	4FSK/GFSK	0.33	±25 kHz

(57) CCA time register (CCATIME)

This register is used to set the time taken for the CCA processing. Initial value: 000DH = 13 symbols (Set value: 1H = 1 symbol).

Setting value is up to 2000 symbols.

CCATIME register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 000DH.

Figure 18 - 70 CCA Time Register (CCATIME) Format

Address: 00B3H, 00B2H After reset: 000DH R/W

Symbol	15	14	13	12	11	10	9	8
CCATIME								

7	6	5	4	3	2	1	0

CCATIME	CCA time setting value

Caution The lower limit of setting value is in accordance with data rate. Refer to the latest application note for setting values when to use the product.

(58) Antenna diversity mode register (BBANTDIV)

This register is used to set the antenna diversity mode.

The antenna diversity enable bit is used to enable the antenna diversity mode. Enabling this bit switches the GPIO1 pin and the GPIO2 pin to the ANTSELOUT0 pin and the ANTSELOUT1 pin.

Transmission antenna switch setting bit is used to fix antenna selection for transmission.

The ACK reply antenna switch setting disable bit is used to set the antenna selection upon ACK reply to the antenna setting upon receive instead of the antenna switch setting bit value.

The ACK receive antenna switch setting disable bit is used to set the antenna selection upon ACK receive to the antenna selection operation instead of the antenna switch setting bit value.

The number-of-antenna-selection-judge bit is used to set the number of judges for the antenna input power value when the antenna diversity mode is enabled. Setting 3 to 5 is available (initial value is three times).

The ANTSELOUT pin status bit is used to indicate which ANTSELOUT pin is currently in the high level output state.

The BBANTDIV register is set by the serial interface in 8-bit units.

Reset signal generation sets this register to 30H.

Figure 18 - 71 Antenna Diversity Mode Register (BBANTDIV) Format

Address: 00B4H After reset: 30H R/W[Note]

Symbol	7	6	5	4	3	2	1	0
BBANTDIV	ANTST	ANTCOUNT2	ANTCOUNT1	ANTCOUNT0	ANTACKRCVDIS	ANTACKRTNDIS	ANTSWTRN SET	ANTDIVEN

ANTST	Pins ANTSELOUT0 and ANTSELOUT1 status bit
0	ANTSELOUT0 pin = High level
1	ANTSELOUT1 pin = High level

ANTCOUNT2	ANTCOUNT1	ANTCOUNT0	Number-of-antenna-selection-judge bit
0	1	1	Three times (Initial value)
1	0	0	Four times
1	0	1	Five times
Other than above			Setting prohibited

ANTACKRCVDIS	ACK receive antenna switch setting disable bit
0	ANTSWSET bit enabled
1	ANTSWSET bit disabled

ANTACKRTNDIS	ACK reply antenna switch setting disable bit
0	ANTSWSET bit enabled
1	ANTSWSET bit disabled

ANTSWTRN SET	Transmission antenna switch setting bit
0	ANTSELOUT0 pin = High level
1	ANTSELOUT1 pin = High level

ANTDIVEN	Antenna diversity enable bit
0	Disable
1	Enable

Note Bits 7 is Read Only.

(59) Mode switch frame transmit register (BBTXMODESW)

This register is used to transmit the mode switch frame. The MODESW bit enables the transmission of the mode switch frame. Setting transmission trigger to 1 by setting 1 to this bit transmits the mode switch frame. Set bits 1 to 15 by using the TXMODESW bits 1 to 15 when transmitting the mode switch frame.

BBTXMODESW register is set via serial interface in 8-bit units.

Reset signal generation clears this register to 0000H.

Figure 18 - 72 Mode Switch Frame Transmit Register (BBTXMODESW) Format

Address: 00B9H, 00B8H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBTXMODESW	TXMODESW15	TXMODESW14	TXMODESW13	TXMODESW12	TXMODESW11	TXMODESW10	TXMODESW9	TXMODESW8

7	6	5	4	3	2	1	0
TXMODESW7	TXMODESW6	TXMODESW5	TXMODESW4	TXMODESW3	TXMODESW2	TXMODESW1	MODESW

TXMODESW[15:1]	The values of PHR bits 1 to 15 of the transmitted mode switch frame

MODESW	Mode switch control bit
0	Disable
1	Enable

(60) Mode switch frame receive register (BBRXMODESW)

This register is used to store the PHY header data (PHR) when the mode switch frame is received. The stored data is retained until when the next mode switch frame is received.

BBRXMODESW register is read via serial interface in 8-bit units.

Reset signal generation clears this register to 0000H.

Figure 18 - 73 Mode Switch Frame Receive Register (BBRXMODESW) Format

Address: 00BBH, 00BAH After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
BBRXMODESW	RXMODESW15	RXMODESW14	RXMODESW13	RXMODESW12	RXMODESW11	RXMODESW10	RXMODESW9	RXMODESW8

7	6	5	4	3	2	1	0
RXMODESW7	RXMODESW6	RXMODESW5	RXMODESW4	RXMODESW3	RXMODESW2	RXMODESW1	RXMODESW0

RXMODESW[15:0]	The values of PHR bits 0 to 15 of the received mode switch frame

(61) Transmit data counter register (BBTXCOUNT)

This register is used to indicate the transmit data counter value upon transmission. It enables the checking of how many bytes of transmit data are currently transferred to transmitter from the transmit RAM. The value is cleared to “0” when the packet transmission ends. Be sure to read data in the order from the lower byte to the upper byte because the upper three bits are stored at the timing when the data of lower 8 bits is read.

BBTXCOUNT register is set via serial interface in 8-bit units.

Reset signal generation clears this register to 0000H.

Figure 18 - 74 Transmit Data Counter Register (BBTXCOUNT) Format

Address: 00BDH, 00BCH After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
BBTXCOUNT	0	0	0	0	00	TRNCOUNT10	TRNCOUNT9	TRNCOUNT8
	7	6	5	4	3	2	1	0
	TRNCOUNT7	TRNCOUNT6	TRNCOUNT5	TRNCOUNT4	TRNCOUNT3	TRNCOUNT2	TRNCOUNT1	TRNCOUNT0

TRNCOUNT[10:0]	The transmit data counter value upon transmission

Caution “0” is always read for bits 11 to 15.

(62) PHY header receive register (BBPHRRX)

This register is used to store the PHY header receive data.

The values of bits 0 to 2, FCSTYPE, and Data Whitening of the PHY header data when receiving the non-mode switch frame are stored. The timing of the storage is the same as the timing when the frame length value is stored.

The reading value is corresponding save bank specified by receive data save bank select bit.

The BBPHRRX register is read by the serial interface in 8-bit units.

Reset signal generation clears this register to 00H.

Figure 18 - 75 PHY Header Receive Register (BBPHRRX) Format

Address: 00BEH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
BBPHRRX	0	0	0	DWBIT	FCSTYPE	PHRRX2	PHRRX1	PHRRX0

DWBIT	Data whitening bit
0	Disable
1	Enable

FCSTYPE	FCS type bit
0	32 bits
1	16 bits

PHRRX2	PHR receive bit 2 value

PHRRX1	PHR receive bit 1 value

PHRRX0	PHR receive bit 0 value

(63) Preamble setting register (BBPABL)

This register is used to optionally set the preamble value. This register consists of 16 bits.

Set following values for each modification.

2FSK/2GFSK: 00AAH

4FSK/4GFSK: 00EEH

BBPABL register is read via serial interface in 8-bit units.

Reset signal generation sets this register to 00AAH.

Figure 18 - 76 Preamble Setting Register (BBPABL) Format

Address: 00C1H, 00C0H After reset: 00AAH R/W

Symbol	15	14	13	12	11	10	9	8
BBPABL	PABL15	PABL14	PABL13	PABL12	PABL11	PABL10	PABL9	PABL8
	7	6	5	4	3	2	1	0
	PABL7	PABL6	PABL5	PABL4	PABL3	PABL2	PABL1	PABL0

PABL[15:0]	preamble value

(64) SFD setting register (BBSFD)

This register is used to set the SFD value for FEC disabled frame transmission when MRFSKSFD bit is 0.

This register consists of 32 bits.

Set following values for each modification.

2FSK/2GFSK: 00007209H

4FSK/4GFSK: BFAEAAEBH

BBSFD register is read via serial interface in 8-bit units.

Reset signal generation sets this register to 00007209H.

Figure 18 - 77 SFD Setting Register (BBSFD) Format

Address: 00C5H to 00C2H After reset: 00007209H R/W

Symbol	31	30	29	28	27	26	25	24
BBSFD	SFD31	SFD30	SFD29	SFD28	SFD27	SFD26	SFD25	SFD24
	23	22	21	20	19	18	17	16
	SFD23	SFD22	SFD21	SFD20	SFD19	SFD18	SFD17	SFD16
	15	14	13	12	11	10	9	8
	SFD15	SFD14	SFD13	SFD12	SFD11	SFD10	SFD9	SFD8
	7	6	5	4	3	2	1	0
	SFD7	SFD6	SFD5	SFD4	SFD3	SFD2	SFD1	SFD0

SFD[31:0]	SFD value

(65) SHR control register (BBSHRCON)

This register is used to set the number of bytes in the preamble setting register to be used for the repetitive pattern for preamble by using the number-of-preamble setting bit.

0: 1 byte (preamble setting register bits 7 to 0)

1: 2 byte (preamble setting register bits 15 to 0)

This register is used to set the number of bytes in the SFD setting register to be used for the SFD output pattern by using the number-of-SFD setting bit.

00: 1 byte (SFD setting register bits 7 to 0)

01: 2 bytes (SFD setting register bits 15 to 0)

10: Prohibited

11: 4 bytes (SFD setting register bits 31 to 0)

The BBBSHRCON register consists of 8 bits and can be accessed (serial interface communication) in 8 bit unit.

Reset signal generation sets this register to 02H.

Figure 18 - 78 SHR Control Register (BBSHRCON) Format

Address: 00C6H After reset: 02H R/W

Symbol	7	6	5	4	3	2	1	0
BBSHRCON	0	0	0	0	0	SFDBYTE1	SFDBYTE0	PABLBYTE

SFDBYTE1	SFDBYTE0	Number-of-SFD-byte setting bit
0	0	1 byte
0	1	2 byte
1	0	Setting disabled
1	1	4 bytes

PABLBYTE	Number-of-preamble-byte setting bit
0	1 byte
1	2 byte

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

(66) ANT0 read register (BBANT0RD)

This register is used to store the RSSI value when the ANTSELOUT0 pin output is at the high level.
The reading value is corresponding save bank specified by receive data save bank select bit.
The value is set as 2's complement. The setting unit is dBm (ex. "19EH" is "-98dBm").
BBANT0RD register is set via serial interface in 8-bit units.
Reset signal generation sets this register to 0100H.

Figure 18 - 79 ANT0 Read Register (BBANT0RD) Format

Address: 00C9H, 00C8H After reset: 0100H R

Symbol	15	14	13	12	11	10	9	8
BBANT0RD	0	0	0	0	0	0	0	ANT0RD8

7	6	5	4	3	2	1	0
ANT0RD7	ANT0RD6	ANT0RD5	ANT0RD4	ANT0RD3	ANT0RD2	ANT0RD1	ANT0RD0

ANT0RD[8:0]	RSSI value when the ANTSELOUT0 pin output is at the high level

Caution "0" is always read for bits 9 to 15.

(67) ANT1 read register (BBANT1RD)

This register is used to store the RSSI value when the ANTSELOUT1 pin output is at the high level.
The reading value is corresponding save bank specified by receive data save bank select bit.
The value is set as 2's complement. The setting unit is dBm (ex. "19EH" is "-98dBm").
BBANT1RD register is set via serial interface in 8-bit units.
Reset signal generation sets this register to 0100H.

Figure 18 - 80 ANT1 Read Register (BBANT1RD) Format

Address: 00CBH, 00CAH After reset: 0100H R

Symbol	15	14	13	12	11	10	9	8
BBANT1RD	0	0	0	0	0	0	0	ANT1RD8

7	6	5	4	3	2	1	0
ANT1RD7	ANT1RD6	ANT1RD5	ANT1RD4	ANT1RD3	ANT1RD2	ANT1RD1	ANT1RD0

ANT1RD[8:0]	RSSI value when the ANTSELOUT1 pin output is at the high level

Caution "0" is always read for bits 9 to 15.

(68) Antenna switch time register (BBANTDIVTIM)

This register is used to set the time to switch between high level and low level outputs of the ANTSELOUT0 and ANTSELOUT1 pin outputs when the antenna diversity is enabled. Initial value: 000AH = 10 symbols (Set value: 1H = 1 symbol).

BBANTDIVTIM register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 000AH.

Figure 18 - 81 Antenna Switch Time Register (BBANTDIVTIM) Format

Address: 00CFH, 00CEH After reset: 000AH R/W

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

BBANTDIVTIM	ANTSELOUT0 and ANTSELOUT1 pin output switch time

(69) Receive start timeout setting register (BBANTTIMOUT)

TThis register is used to set the timeout value when there is no receive start operation after when exceeding the threshold value upon antenna diversity in operation. The operation returns to the antenna select operation after the timeout. Initial value: 0100H= 256 symbols (Set value: 1H = 1 symbol).

BBANTTIMOUT register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 0100H.

Figure 18 - 82 Receive Start Timeout Setting Register (BBANTTIMOUT) Format

Address: 00D1H, 00D0H After reset: 0100H R/W

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

BBANTTIMOUT	Timeout value when there is no receive start operation.

(70) Number-of-receive-byte interrupt compare register (BBRCVINTCOMP)

This register is used to generate interrupt depending on the number of received bytes. Generates the interrupt request when receiving the specified number of bytes.

The BBRCVINTCOMP register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 0010H.

Figure 18 - 83 Number-of-Receive-Byte Interrupt Compare Register (BBRCVINTCOMP) Format

Address: 00D3H, 00D2H After reset: 0010H R/W

Symbol	15	14	13	12	11	10	9	8
BBRCVINTCOMP						RCVINTCOMP10	RCVINTCOMP9	RCVINTCOMP8
	7	6	5	4	3	2	1	0
	RCVINTCOMP7	RCVINTCOMP6	RCVINTCOMP5	RCVINTCOMP4	RCVINTCOMP3	RCVINTCOMP2	RCVINTCOMP1	RCVINTCOMP0

RCVINTCOMP[10:0]	Number of bytes which generate the number-of-receive-byte interrupt

(71) Back off period total number register (BBBOPTOTAL)

This register is used to indicate the total number of count values of the back off period upon CSMA-CA. The maximum value of the count value is FFFFH.

BBBOPTOTAL register is set via serial interface in 8-bit units.

Reset signal generation clears this register to 0000H.

Figure 18 - 84 Back off Period Total Number Register (BBBOPTOTAL) Format

Address: 00D5H, 00D4H After reset: 0000H R

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

BBBOPTOTAL	Back off period total number

(72) CCA total number register (BBCCATOTAL)

This register is used to indicate the total number of count vales upon CSMA-CA. The maximum value of the count is FFH.

BBCCATOTAL register is set via serial interface in 8-bit units.

Reset signal generation clears this register to 00H.

Figure 18 - 85 CCA Total Number Register (BBCCATOTAL) Format

Address: 00D6H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
BBCCATOTAL								

BBCCATOTAL	CCA total number count value

(73) RF initial setting register 00 to 02 (RFINI00 to RFINI02)

Data read/write area with sub-address is implemented in RF block. These registers are used to set the area. These registers consist of 24 bits.

-Data write

Write RF initial setting register 00 and RF initial setting register 01 continuously.

RF initial setting register 00 (bits 7 to 0): write data

RF initial setting register 01 (bit 7): 0

RF initial setting register 01 (bits 6 to 0): sub-address in RF

-Data read

Write RF initial setting register 00 and RF initial setting register 01 continuously.

RF initial setting register 00 (bits 7 to 0): 00H

RF initial setting register 01 (bit 7): 1

RF initial setting register 01 (bits 6 to 0): sub-address in RF

Read RF initial setting register 02 after writing RF initial setting register 00 and 01.

Caution Two bytes of RF initial setting register 00 and 01 must be written continuously (SEN signal keeps low level).

RF initial setting register 00 to 02 are set via serial interface in 8-bit units.

RF initial setting register 00 and 01 are write only and the read values are undefined.

RF initial setting register 02 is read only and the initial value is undefined.

Figure 18 - 86 RF initial setting register 00 to 02 (RFINI00 to RFINI02) Format

Address: 00D8H After reset: undefined W

Symbol	7	6	5	4	3	2	1	0
RFINI00	RFINI007	RFINI006	RFINI005	RFINI004	RFINI003	RFINI002	RFINI001	RFINI000
	RFINI00		RFINI00 value					

Address: 00D9H After reset: undefined W

Symbol	7	6	5	4	3	2	1	0
RFINI01	RFINI017	RFINI016	RFINI015	RFINI014	RFINI013	RFINI012	RFINI011	RFINI010
	RFINI01		RFINI01 value					

Address: 00DAH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
RFINI02	RFINI027	RFINI026	RFINI025	RFINI024	RFINI023	RFINI022	RFINI021	RFINI020
	RFINI02		RFINI02 value					

(74) RF initial setting register 10 to 12 (RFINI10 to RFINI12)

Data read/write area with sub-address is implemented in RF block. These registers are used to set the area. These registers consist of 24 bits.

-Data write

Write RF initial setting register 10 and RF initial setting register 11 continuously.

RF initial setting register 10 (bits 7 to 0): write data

RF initial setting register 11 (bit 7): 0

RF initial setting register 11 (bits 6 to 0): sub-address in RF

-Data read

Write RF initial setting register 10 and RF initial setting register 11 continuously.

RF initial setting register 10 (bits 7 to 0): 00H

RF initial setting register 11 (bit 7): 1

RF initial setting register 11 (bits 6 to 0): sub-address in RF

Read RF initial setting register 12 after writing RF initial setting register 10 and 11.

Caution Two bytes of RF initial setting register 10 and 11 must be written continuously (SEN signal keeps low level).

RF initial setting register 10 to 12 are set via serial interface in 8-bit units.

RF initial setting register 10 and 11 are write only and the read values are undefined.

RF initial setting register 12 is read only and the initial value is undefined.

Figure 18 - 87 RF initial setting register 10 to 12 (RFINI10 to RFINI12) Format

Address: 00DCH After reset: undefined W

Symbol	7	6	5	4	3	2	1	0
RFINI10	RFINI107	RFINI106	RFINI105	RFINI104	RFINI103	RFINI102	RFINI101	RFINI100

RFINI10	RFINI10 value

Address: 00DDH After reset: undefined W

Symbol	7	6	5	4	3	2	1	0
RFINI11	RFINI117	RFINI116	RFINI115	RFINI114	RFINI113	RFINI112	RFINI111	RFINI110

RFINI11	RFINI11 value

Address: 00DEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
RFINI12	RFINI127	RFINI126	RFINI125	RFINI124	RFINI123	RFINI122	RFINI121	RFINI120

RFINI12	RFINI12 value

(75) PAN identifier register 1 (BBPANID1)

This register is used to set the PAN identifier of second address filter. It consists of 16 bits and is used to detect the match with a received PAN identifier.

BBPANID1 register is set via serial interface in 8-bit units.

Reset signal generation sets this register to FFFFH.

Figure 18 - 88 PAN Identifier Register 1 (BBPANID1) Format

Address: 00E1H, 00E0H After reset: FFFFH R/W

Symbol	15	14	13	12	11	10	9	8
BBPANID1								

7	6	5	4	3	2	1	0

BBPANID1	PAN identifier

(76) Short address register 1 (BBSHORTAD1)

This register is used to set the short address of second address filter. It consists of 16 bits and is used to detect the match with a received PAN identifier.

BBSHORTAD1 register is set via serial interface in 8-bit units.

Reset signal generation sets this register to FFFFH.

Figure 18 - 89 Short Address Register 1 (BBSHORTAD1) Format

Address: 00E3H, 00E2H After reset: FFFFH R/W

Symbol	15	14	13	12	11	10	9	8
BBSHORTAD1								

7	6	5	4	3	2	1	0

BBSHORTAD1	Short address

(77) Extended address register 1 (BBEXTENDAD10 to BBEXTENDAD13)

This register is used to set the extended address of second address filter. It consists of 64 bits (16 bit timed by 4) and is used to detect the match with a received extended address.

BBEXTENDAD10 to BBEXTENDAD13 registers are set via serial interface in 8-bit units.

Reset signal generation clears these registers to 0000H.

Figure 18 - 90 Extended Address Register 1 (BBEXTENDAD10 to BBEXTENDAD13) Format

Address: 00EBH, 00EAH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBEXTENDAD13								
	7	6	5	4	3	2	1	0

BBEXTENDAD13	Extended address bits 63 to 48

Address: 00E9H, 00E8H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBEXTENDAD12								
	7	6	5	4	3	2	1	0

BBEXTENDAD12	Extended address bits 47 to 32

Address: 00E7H, 00E6H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBEXTENDAD11								
	7	6	5	4	3	2	1	0

BBEXTENDAD11	Extended address bits 31 to 16

Address: 00E5H, 00E4H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
BBEXTENDAD10								
	7	6	5	4	3	2	1	0

BBEXTENDAD10	Extended address bits 15 to 0

(78) Receive timeout register (BBTIMEOUT)

This register is used to set the timeout time when the automatic reception with timeout mode is enabled.

This register consists of 12 bits.

BBTIMEOUT register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 07D0H. (07D0H = 2 ms, 1H = 1 μs)

Figure 18 - 91 Receive Timeout Register (BBTIMEOUT) Format

Address: 00EDH, 00ECH After reset: 07D0H R/W

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

BBTIMEOUT	Timeout time setting value

Caution Be sure to clear bits 12 to 15 to “0”.

(79) ANTSW control register (ANTSWCON)

This register controls the ANTSW signal.

The ANTSW signal can be output from the GPIO4 pin by using the ANTSW output enable bit.

ANTSWCON register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 00H.

Figure 18 - 92 ANTSW Control Register (ANTSWCON) Format

Address: 0080H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ANTSWCON	0	0	0	0	0	0	0	ANTSWEN

ANTSWEN	ANTSW output enable bit
0	Normal port
1	ANTSW output

Caution Be sure to clear bits 1 to 7 to “0”.

(80) Clock output control register (CLKOUTCON)

This register controls the clock output.

The clock can be output from the GPIO0 pin by using the clock output enable bit.

The output clock can be selected between the clock output select bit 0 and the clock output select bit 1.

CLKOUTCON register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 00H.

Figure 18 - 93 Clock Output Control Register (CLKOUTCON) Format

Address: 0081H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CLKOUTCON	0	0	0	0	0	CLKOUTSEL1	CLKOUTSEL0	CLKOUTEN

CLKOUTSEL1	CLKOUTSEL0	Clock output select bit
0	0	8 MHz
0	1	12 MHz
1	0	4 MHz
1	1	Setting prohibited

CLKOUTEN	Clock output enable bit
0	Normal port
1	Clock output enabled

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

(81) Port direction register (GPIODIR)

This register is used to set the I/O of the respective ports of GPIO.

GPIODIR register is set via serial interface in 8-bit units.

Reset signal generation clears this register to 00H.

Figure 18 - 94 Port Direction Register (GPIODIR) Format

Address: 0082H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
GPIODIR	0	0	0	GPIO4DIR	GPIO3DIR	GPIO2DIR	GPIO1DIR	GPIO0DIR

GPIO4DIR	GPIO4 direction bit
0	Input port
1	Output port

GPIO3DIR	GPIO3 direction bit
0	Input port
1	Output port

GPIO2DIR	GPIO2 direction bit
0	Input port
1	Output port

GPIO1DIR	GPIO1 direction bit
0	Input port
1	Output port

GPIO0DIR	GPIO0 direction bit
0	Input port
1	Output port

Caution Be sure to clear bits 5 to 7 to “0”.

(82) Port data register (GPIODATA)

This register is used to set the output value when the respective ports of GPIO are set to output.

When a GPIO port is set to input, the pin state (H/L) can be read.

GPIODATA register is set via serial interface in 8-bit units.

Reset signal generation clears this register to 00H.

Figure 18 - 95 Port Data Register (GPIODATA) Format

Address: 0083H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
GPIODATA	0	0	X	GPIO4DATA	GPIO3DATA	GPIO2DATA	GPIO1DATA	GPIO0DATA

GPIO4DATA	GPIO4 data bit
0	0: Output
1	1: Output

GPIO3DATA	GPIO3 data bit
0	0: Output
1	1: Output

GPIO2DATA	GPIO2 data bit
0	0: Output
1	1: Output

GPIO1DATA	GPIO1 data bit
0	0: Output
1	1: Output

GPIO0DATA	GPIO0 data bit
0	0: Output
1	1: Output

Caution Be sure to clear bits 5 to 7 to “0”. Bit 5 becomes X (undefined) when read out.

(83) SFD setting register 2 (BBSFD2)

This register is used to set the SFD value for FEC enabled frame transmission/reception when MRFSKSFD bit is 0.

Set the following values for each modulation.

2FSK/2GFSK: 000072F6H

4FSK/4GFSK: BFAEFFBEH

The BBSFD2 register consists of 32 bits. Reset signal generation sets this register to undefined.

BBSFD2 register is set via serial interface in 8-bit units.

Figure 18 - 96 SFD Setting Register 2 (BBSFD2) Format

Address: 0103H-0100H After reset: XXXXXXXXH R/W

Symbol	31	30	29	28	27	26	25	24
BBSFD2	SFD231	SFD230	SFD229	SFD228	SFD227	SFD226	SFD225	SFD224
	23	22	24	20	19	18	17	16
	SFD223	SFD222	SFD221	SFD220	SFD219	SFD218	SFD217	SFD216
	15	14	13	12	11	10	9	8
	SFD215	SFD214	SFD213	SFD212	SFD211	SFD210	SFD209	SFD208
	7	6	5	4	3	2	1	0
	SFD207	SFD206	SFD205	SFD204	SFD203	SFD202	SFD201	SFD200

SFD231-SFD200	SFD2 value

(84) SFD setting register 3 (BBSFD3)

This register is used to set the SFD value for FEC disabled frame transmission/reception when MRFSKSFD bit is 1.

Set the following values for each modulation.

2FSK/2GFSK: 0000705EH

4FSK/4GFSK: BFAABBFEH

The BBSFD3 register consists of 32 bits. Reset signal generation sets this register to undefined.

BBSFD3 register is set via serial interface in 8-bit units.

Figure 18 - 97 SFD Setting Register 3 (BBSFD3) Format

Address: 0107H-0104H After reset: XXXXXXXXH R/W

Symbol	31	30	29	28	27	26	25	24
BBSFD3	SFD331	SFD330	SFD329	SFD328	SFD327	SFD326	SFD325	SFD324
	23	22	24	20	19	18	17	16
	SFD323	SFD322	SFD321	SFD320	SFD319	SFD318	SFD317	SFD316
	15	14	13	12	11	10	9	8
	SFD315	SFD314	SFD313	SFD312	SFD311	SFD310	SFD309	SFD308
	7	6	5	4	3	2	1	0
	SFD307	SFD306	SFD305	SFD304	SFD303	SFD302	SFD301	SFD300

SFD331- SFD300	SFD3 value

(85) SFD setting register 4 (BBSFD4)

This register is used to set the SFD value for FEC enabled frame transmission/reception when MRFSKSFD bit is 1.

Set the following values for each modulation.

2FSK/2GFSK: 0000B4C6H

4FSK/4GFSK: EFBAFABEH

The BBSFD4 register consists of 32 bits. Reset signal generation sets this register to undefined.

BBSFD4 register is set via serial interface in 8-bit units.

Figure 18 - 98 SFD Setting Register 4 (BBSFD4) Format

Address: 010BH-0108H After reset: XXXXXXXXH R/W

Symbol	31	30	29	28	27	26	25	24
BBSFD4	SFD431	SFD430	SFD429	SFD428	SFD427	SFD426	SFD425	SFD424
	23	22	24	20	19	18	17	16
	SFD423	SFD422	SFD421	SFD420	SFD419	SFD418	SFD417	SFD416
	15	14	13	12	11	10	9	8
	SFD415	SFD414	SFD413	SFD412	SFD411	SFD410	SFD409	SFD408
	7	6	5	4	3	2	1	0
	SFD407	SFD406	SFD405	SFD404	SFD403	SFD402	SFD401	SFD400

SFD431- SFD400	SFD4 value

(86) FEC control register (BBFECCON)

FEC automatic identification enable bit is used to receive with identification of enabled or disabled FEC.

FEC enable bit for transmission is used to set FEC enable or disable of transmission.

FEC control bit for automatic ACK reply must be set to 1 to use automatic ACK reply function.

FEC enable bit for automatic ACK reply is used to set enable or disable of FEC in automatic ACK reply when FEC control bit for automatic ACK reply is set to 1.

FEC control bit for automatic ACK reception is used to switch enable or disable control of FEC for automatic ACK reception is whether depends on FEC control bit for automatic ACK reply.

FEC enable bit for automatic ACK reception is used to set enable or disable of FEC in automatic ACK reception when FEC control bit for automatic ACK reception is set to 1.

MRFSKSFD select bit is used to set value of phyMRFSKSFD bit.

BBFECCON register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 00H.

Figure 18 - 99 FEC Control Register (BBFECCON) Format

Address: 010CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBFECCON	x	MRFSKSFD	FECEN ACKRCV	FECCON ACKRCV	FECEN ACKRTN	FECCON ACKRTN	FECENTX	FEC AUTOEN

MRFSKSFD	MRFSKSFD select bit
0	phyMRFSKSFD = 0
1	phyMRFSKSFD = 1

FECEN ACKRCV	FEC enable bit for automatic ACK reception
0	Disable
1	Enable

FECCON ACKRCV	FEC control bit for automatic ACK reception
0	Regardless of FEC enable bit for automatic ACK reception
1	Depend on FEC enable bit for automatic ACK reception

FECEN ACKRTN	FEC enable bit for automatic ACK reply
0	Disable
1	Enable

GPIO2DATA FEC bit	FEC control bit for automatic ACK reply
0	Disables automatic ACK reply
1	Enables automatic ACK reply

FECENTX	FEC enable bit for transmission
0	Disable
1	Enable

FECEN AUTOEN	FEC automatic identification enable bit
0	Disable
1	Enable

Caution Be sure to set “0” to bit 7. The bit is read as x (undefined).

(87) Address filter extension address control register (BBADFCON)

PAN coordinator 2 bit is used to set whether PAN coordinator on second address filter side.

Frame pending 2 bit is used to set enable or disable of frame pending for ACK reply on second address filter side.

First address filter match monitor bit is used to monitor whether the first address matches. The reading value is corresponding save bank specified by receive data save bank select bit.

Second address filter match monitor bit is used to monitor whether the second address matches. The reading value is corresponding save bank specified by receive data save bank select bit.

When the address filter extension bit is disabled, values of first and second address filter match monitor bits are invalid.

When the address filter extension bit is enabled, values of first and second address filter match monitor bits are invalid under the following conditions.

- Frame version: 00, 01
- Frame type: Beacon frame
- No destination PANID and destination address
- Values of source PANID and PAN identifier register match, or value of PAN identifier register = FFFFH
- PANCORD bit = 0

BBADFCON register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 00H.

Figure 18 - 100 Address Filter Extension Address Control Register (BBADFCON) Format

Address: 010DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBADFCON	0	0	0	0	ADFMONI2	ADFMONI1	FLMPEND2	PANCORD2

ADFMONI2	Second address filter match monitor bit
0	Not match to second address
1	Match to second address

ADFMONI1	First address filter match monitor bit
0	Not match to first address
1	Match to first address

FLMPEND2	Frame pending 2 bit
0	No frame pending
1	With frame pending

PANCORD2	PAN coordinator 2 bit
0	Not a PAN coordinator
1	PAN coordinator

Caution Be sure to clear bits 4 to 7 to “0”. Bits 2,3 are Read Only.

(88) Antenna diversity mode register 2 (BBANTDIV2)

ACK reply antenna switch setting bit is used to select ACK reply antenna for automatic ACK reply. This bit is enabled when ANT receive ANTSW setting disable bit is 0.

ACK receive antenna switch setting bit is used to select ACK receive antenna for automatic ACK reception. This bit is enable when ANT reply ANTSW setting disable bit is 0.

Receive antenna monitor bit is used to monitor antenna for reception.The reading value is corresponding save bank specified by receive data save bank select bit.

ACK reply antenna switch invert bit is used to invert to receive antenna output as ACK reply antenna for automatic ACK reply.

BBANTDIV2 register is set via serial interface in 8-bit units.

Reset signal generation sets this register to 00H.

Figure 18 - 101 Antenna Diversity Mode Register 2 (BBANTDIV2) Format

Address: 010EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BBANTDIV2	0	0	0	0	ACKRTN RVS	ANTMONI	ANTACK RCVSET	ANTACK RTNSET

ACKRTN RVS	ACK reply antenna switch invert bit
0	Not reversed
1	Reversed

ANTMONI	Receive antenna monitor bit
0	ANTSELOUT0 = H
1	ANTSELOUT1 = H

ANTACK RCVSET	ACK receive antenna switch setting bit
0	ANTSELOUT0 = H
1	ANTSELOUT1 = H

ANTACK RTNSET	ACK reply antenna switch setting bit
0	ANTSELOUT0 = H
1	ANTSELOUT1 = H

Caution Be sure to clear bits 4 to 7 to "0". Bit 2 is Read Only.

(89) Lower limit threshold setting register after SFD detection (PWRLOWTH2)

<R> This register is used to set threshold for detection of no communication state after SFD detection. This register is valid when receive level filter enable bit is 1.

<R> An interrupt request is issued as receive level filter interrupt to return to receive wait state detecting no communication state when the connection is cut during communication or frame length is error detected. When the operation is under noisy environment usually, no communication state is detected correctly adjusting this register. The default is 1 (no communication state is detected when the communication does not formed completely). Set this register to 2 under match noisy environment.

<R> **Figure 18 - 102 Lower limit threshold setting register after SFD detection (PWRLOWTH2) Format**

Address: 050FH After reset: 00H W

Symbol	7	6	5	4	3	2	1	0
PWRLOWTH2								

PWRLOWTH2	Lower limit threshold value after SFD detection (threshold level of receive level filter)

Setting example (at 100 kbps)

Usual use: 01H

Under match noisy environment: 02H

Caution Receive characteristic might be degraded when the setting value is over 02H.

18.5 Serial Interface Only for Internal Communication

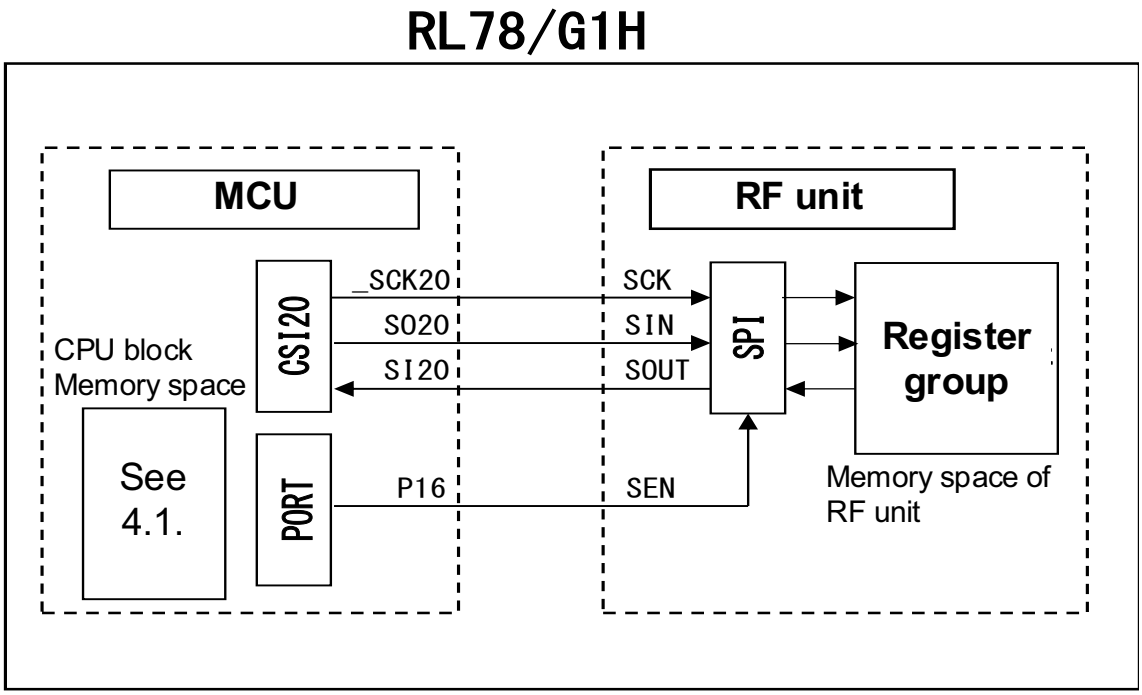
18.5.1 Overview

The control register for each type of RF assigned to the RF memory space accesses from MCU through serial communication.

The serial connection on the user board, etc. is not required since this serial interface is internally connected in RL78/G1H device.

Figure 18 - 103 shows the internal serial interface.

Figure 18 - 103 Serial Interface Only for Internal Communication



18.5.2 Communication specification

Table 18 - 10 shows the internal communication specification.

Table 18 - 10 Internal Communication Specification

Item	MCU	RF Unit
	CSI20	SPI
Target channel	Serial array unit (SAU1) channel 0 (CSI20)	Serial interface (SPI)
Internal pin to be used	• SCK20, SI20, and SO20 inside-only pins 3-wire CSI communication pin • P16 inside-only pin Port pin for SEN control	• SCK, SIN, SOUT, and SEN inside-only pins 4-Wire SPI communication pin
Operating mode	Transmission mode/Transmission/reception mode	Transmission/reception mode
Master/slave	Only master is supported.	Slave
Interrupt	Selectable from the following types. • Transfer end interrupt (in single-transfer mode) • Buffer empty interrupt (in continuous transfer mode)	None
Error detection flag	Overrun/error detection flag	None
Transfer data length	Only 8 bits are supported.	8 bits
Transfer rate	Within a range that satisfies the electrical characteristics (AC specification) **Note 1**	With in the range described in the left
Data phase	Only Type 1 is supported. **Note 2**	Equivalent to the description in the left
Clock phase	Only Type 1 is supported. **Note 2**	Equivalent to the description in the left
Data direction	Only MSB is supported.	MSB
Software setting	As described above	Not especially (Controlling SEN pin to be fixed at the low level allows the communication.)

Note 1. For details, see **CHAPTER 31 ELECTRICAL SPECIFICATIONS**.

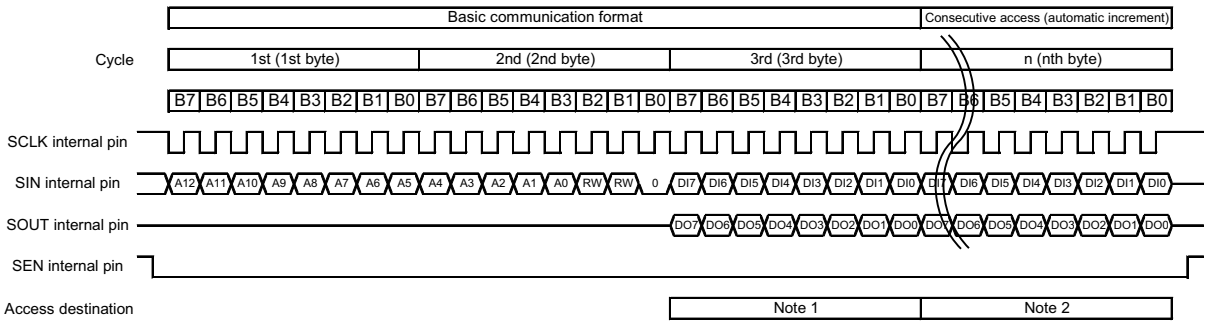
Note 2. For details, see **CHAPTER 14 SERIAL ARRAY UNIT**.

18.5.3 Communication Format

Data communication is performed in the communication format of 8-bit unit serial communication. The basic communication format consists of 2 bytes of address and R/W control and 1 byte of data, 3 bytes in total. When the SEN internal pin remains at the low level after reaching 3 bytes of the basic communication, the data cycle occurs in the 4th byte or later. In this case, the address is automatically incremented, and the data of the 4th byte or later is written in the incremented address or read out.

Figure 18 - 104 shows the serial communication format, and Table 18 - 11 shows the description of the format.

Figure 18 - 104 Internal Serial Communication Format



Note 1. Data is written in the address specified in the 1st or 2nd byte or read.

Note 2. The address is automatically incremented for each byte, and the data is written in the incremented address or read during the consecutive access.

Table 18 - 11 Contents of Communication Format

Format	Contents
A12 to A0	13-bit address at the access destination. Bit 7 to Bit 3 in the 1st byte and 2nd byte.
R/W	2-bit R/W control. Bit 2 and Bit 1 in the 2nd byte. 00b: Writes to the access destination. 11b: Reads from the access destination.
Fixed to 0	Bit 0 in the 2nd byte must be fixed to 0.
DI7 to DI0	8-bit data written to the access destination.
D07 to DO0	8-bit data read from the access destination.

18.6 RF Mode

18.6.1 RF operating mode

RF operating mode is classified into the following three modes.

(1) RF transmission mode
Indicates the transmission state. Outputs the data written in the data RAM to the RFOUT pin in the specified format.

(2) RF reception mode
Indicates the reception state. Receives the signal input from the RFIP pin.

(3) IDLE mode
Power supply circuit (DDC/LDO) and 48 MHz crystal oscillator circuit are activated. The settings and data of each register in the RF transceiver are retained.
-Returns to IDLE after completion of operation in case of transmission.
-Returns to IDLE after completion of operation or setting RFSTOP bit to 1 in case of reception.
-Returns to IDLE after completion of operation or setting RFSTOP bit to 1 in case of CCA..

18.6.2 RF standby mode

RF standby mode has the following mode.

(1) SLEEP mode
Turns all the RF transceiver internal circuits to the OFF state. Each register is reset to default value, and setting is also disabled. To change to the IDLE state, Wake Up (startup sequence) must be executed according to the procedures in 18.6.4.

18.6.3 State transition

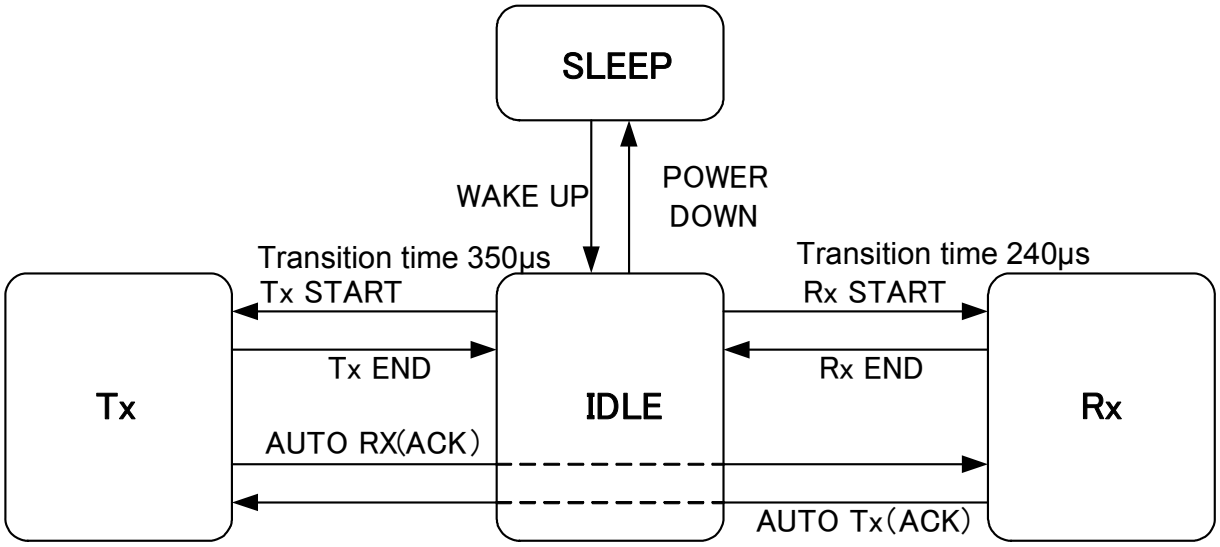
Figure 18 - 105 shows the state transition. The state transition to each state of SLEEP, IDLE, TX, and RX is performed according to the setting.

The state returns to IDLE automatically after transmit completion, receive completion or CCA completion.

The following function is provided between TX and RX.

- AUTO RX (ACK reception) function: Changes to the reception state automatically via the IDLE state after data transmission.
- AUTO TX (ACK transmission) function: Changes to the transmission state automatically via the IDLE state after data reception.

Figure 18 - 105 RF Unit State Transition



18.6.4 Mode transition

(1) Wake Up (transition from SLEEP to IDLE) operation

Wake Up operation means the startup sequence operation (transition) of RF unit.

Figure 18 - 106 shows the Wake Up operation (RF unit startup sequence) for the XTAL_RF oscillator. While the power is applied externally, set to the high level of STANDBY pin → OSCDRVSEL internal pin → DON internal pin → RFRESETB internal pin in this order.

Figure 18 - 107 shows the Wake Up operation (RF unit startup sequence) for the REFCLKIN_RF external clock (TCXO, etc) input. The high level is fixed in the OSCDRVSEL internal pin. Set to the high level in the STANDBY pin and the DON internal pin sequentially. There is no restriction of the interval between setting STANDBY pin to the high level and setting DON internal pin to the high level especially (0 μs or higher must be kept). For each state of <1> <2> and <3> shown in Figure 18 - 106 and Figure 18 - 107, set to secure the described time or longer. The descriptions of each state are listed in Table 18 - 12.

Figure 18 - 106 Wake Up Operation (for XTAL_RF Oscillator)

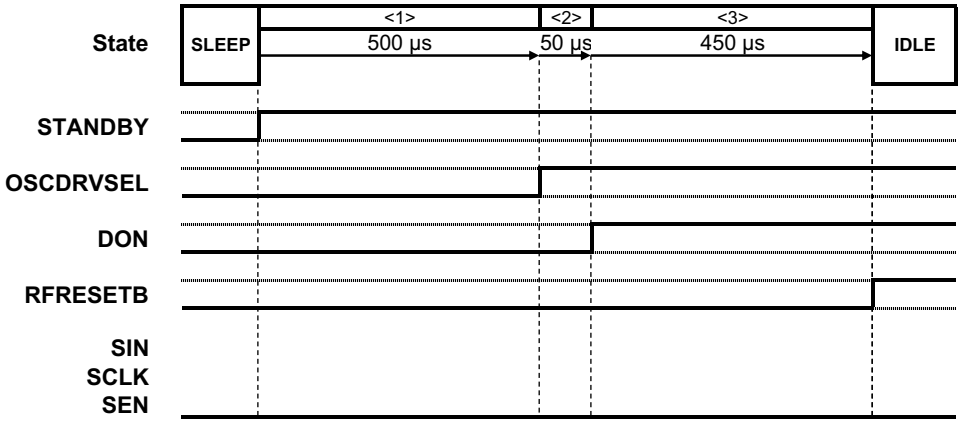


Figure 18 - 107 Wake Up Operation (for REFCLKIN_RF External Clock)

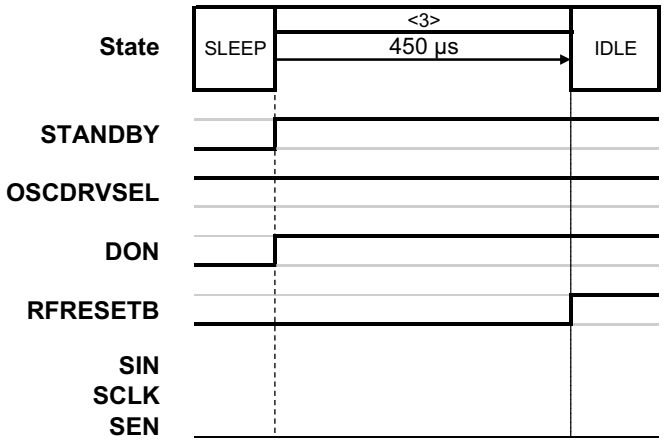


Table 18 - 12 Pin Status

Pin	SLEEP	<1>	<2>	<3>	IDLE
SEN Internal pin	Input	Input	Input	Input	Input
SCLK Internal pin	Input	Input	Input	Input	Input
SIN Internal pin	Input	Input	Input	Input	Input
SOUT Internal pin**Note**	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Output
External pin and internal connection pin**Note**	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Output
GPIO0 to GPIO4	Hi-Z	Hi-Z	Hi-Z	Hi-Z	According to register setting value

Note SOUT and INTOUT internal pins are high impedance state in SLEEP mode and period between statuses <1> and <3>. Then fix the levels with pull-up functions of MCU pins connected internally.

Table 18 - 13 Description of Each State of Wake Up Operation

State	Description
<1>	Time taken from startup of the XTAL_RF oscillator circuit to the oscillation development. When the OSCDRVSEL internal pin is at the low level, the buffer size (current) of the oscillator circuit becomes larger.
<2>	Time taken from switch of the buffer size of the XTAL_RF oscillator circuit to the output stabilization. When the OSCDRVSEL internal pin is at the high level, the buffer size (current) of the oscillator circuit is smaller.
<3>	Time taken from start of the DC-DC converter operation to the output voltage stabilization.

Set initial setting registers in the table below in order from the top in IDLE mode after status <3>.

Setting values for initial setting registers are not depend on “Frequency band identifier” in 31.7.7 Setting values are the same in cases of crystal oscillation and external clock (TCXO, etc.).

Refer to the latest application note for setting data in Table 18 - 14 when to use the product.

Table 18 - 14 Initial Setting Registers

Register address (H)	Setting value (H)
000A	8C
003A	80
0048	04
0052	0E
0053	01
0054	5E
0057	50
0058	E6
005A	F0
005C	F0
0078	E6
007C	50
007E	E6
0086	03
008E	73
0092	7B
0094	26
0096	73
00A6	0F
0402	04
0430	02
043A	94
046F	05
0470	05
0473	F6
0475	32
047A	60
047B	01
047E	03
0481	00
0488	82
04EE	00
04F9	03
0501	10
050D	60
050E	01
050F	02
0510	00
0515	03
0583	7F
0587	7F
05A5	0A
05A6	0A
05AD	40
00D8	04
00D9	44
00DC	8D
00DD	1F
00DC	00
00DD	6E

(2) Power Down (transition from IDLE to SLEEP) operation

Power Down operation stops the internal circuit of RF unit and transits to the low power consumption mode. Set the STANDBY pin to the low level to enter the SLEEP state. When using the XTAL_RF oscillator circuit, set the STANDBY pin to the low level, and then set the OSCDRVSEL internal pin, DON internal pin, and RFRESETB internal pin to the low level respectively. When using the REFCLKIN_RF external clock, fix the OSCDRVSEL internal pin at the high level, set the STANDBY pin to the low level, and then set the DON internal pin and the RFRESETB internal pin to the low level respectively.

(3) RF transmission/reception (transition from IDLE to RF transmission or RF reception)

The state transits from idle to RF transmission or RF reception by the register control.
For the procedure for setting, see **18.7**.

18.6.5 Pin state in each RF mode

Table 18 - 15 shows the pin states in each RF mode.

Table 18 - 15 Function State in Each RF Mode

Pin	SLEEP	IDLE	RF Transmission	RF Reception
STANDBY	Low input	High input	High input	High input
OSCDRVSE internal pin	Low input (During XTAL_RF oscillation) High input (During REFCLKIN_RF external clock)	High input	High input	High input
DON internal pin	Low input	High input	High input	High input
RFRESETB internal pin	Low input	High input	High input	High input
INTOUT	Hi-Z	Operable	Operable	Operable
<R> CLKOUT	Hi-Z	Operable	Operation prohibited	Operation prohibited
GPIO0 GPIO1 GPIO2 GPIO3 GPIO4	Hi-Z	Operable	Operable	Operable

18.6.6 Function state in each RF mode

Table 18 - 16 shows the function states in each RF mode.

Table 18 - 16 Function State in Each RF Mode

Pin	SLEEP	IDLE	RF Transmission	RF Reception
DC-DC converter	Operation stop	Operation	Operation	Operation
Digital block	Operation stop	Operation	Operation	Operation
RF transmission/ reception circuit block	Operation stop	Operation stop	Operation	Operation
Clock circuit 48-MHz oscillation circuit	Operation stop	Operation	Operation	Operation
Clock circuit Clock output circuit	Operation stop	Operable	Operable	Operable

18.7 Example of Procedure for Setting

18.7.1 Example of procedure for each operation

(1) Example of procedure for RF transmission

<1> Set to the IDLE state.

<2> Set the transmission frequency in the frequency setting register (BBFREQ).
For details, see **18.7.2 (2) RF frequency setting for transmission**.

<3> Set the transmission output power.
For details, see **18.7.2**

<4> Set the AUTORCV0 bit of the transmission/reception mode register 0 (BBTXRXMODE0) to ”1” when using automatic reception switch mode.

<5> Write in the transmission RAM: Bank 0 (0200H to 027FH address), Bank 1 (0280H to 02FFH address).

<6> Set the transmission frame length in the transmission frame length register (BBTXFLEN).

<7> Set the TRNTRG bit of the transmission/reception control register (BBTXRXCON) to ”1” (transmission start).

After start of transmission, the transmission completion interrupt occurs when the following events occur.

- Transmission completion
- Transmission together with the ACK request with the ACK reception function enabled to complete the ACK reception
- Failure in the ACK reception for a certain period of time after transmission together with the ACK request with the ACK reception function enabled
- Channel busy as the CCA result after transmission with the automatic CSMA-CA enabled

(2) Example of procedure for RF reception

<1> Set to the IDLE state.

<2> Set the reception frequency in the frequency setting register (BBFREQ).
For details, see **18.7.2 (3) RF frequency setting for reception**.

<3> Set the AUTOACKEN bit, AUTORCV0 bit, and BEACON bit of the transmission/reception mode register 0 (BBTXRXMODE0) to ”1” when using the automatic ACK mode.

<4> Set the PAN identifier in the PAN identifier register 0 (BBPANID0) and PAN identifier register 1 (BBPANID1).

<5> Set the short address register 0 (BBSHORTAD0), short address register 1 (BBSHORTAD1), or extended address register 0 (BBEXTENDAD00 to BBEXTENDAD03), extended address register 1 (BBEXTENDAD10 to BBEXTENDAD13).

<6> Set the RCVTRG bit of the transmission/reception control register (BBTXRXCON) to ”1” (reception start).

<7> Wait for the reception completion interrupt.

<8> Set the bank selection in the RCVBANKSEL bit of the transmission/reception mode register 3 (BBTXRXMODE3).

<9> Read the reception frame length register (BBRXFLEN).

<10> Check the CRC result by the CRC bit of the transmission/reception status register 0 (BBTXRXST0).

<11> Read the reception RAM data: Bank 0 (0300H to 037FH address), Bank 1 (0380H to 03FFH address).

(3) Example of procedure for CCA

<1> Set to the IDLE state.

<2> Set the reception frequency in the frequency setting register (BBFREQ).
For details, see **18.7.2 (3) RF frequency setting for reception**.

<3> Set the CCATRG bit of the transmission/reception control register (BBTXRXCON) to ”1” (CCA start).

<4> Wait for the CCA completion interrupt.

<5> Check the CCA result by the CCA bit of the transmission/reception status register 0 (BBTXRXST0).

(4) Example of procedure for CSMA-CA

<1> Set to the IDLE state.

<2> Set the reception frequency in the frequency setting register (BBFREQ).
For details, see **18.7.2 (3) RF frequency setting for reception**.

<3> For the beacon mode, set the BEACON bit of the transmission/reception mode register 0 (BBTXRXMODE0) to ”1”.

<4> Set the BOFFPROD0-BOFFPROD7 bit of the back-off period register 2 (BBBOFFPROD2) to the default value. And then, set the BOFFPRODEN bit of the back-off period register (BBBOFFPROD) to “1” (back-off period register automatic random permission).

<5> Set the CSMAST bit of the CSMA control register 0 (BBCSMACON0) to ”1” (Automatic CSMA-CA start). To perform transmission processing after CSMA-CA completion simultaneously, set the CSMATRNST bit of the CSMA control register 0 (BBCSMACON0) to ”1” (transmission processing after CSMA-CA).

<6> Wait for the CSMA-CA completion interrupt.

<7> Check the CSMA-CA result by the CSMA-CA bit of the transmission/reception status register 0 (BBTXRXST0).

18.7.2 Example of procedure for function setting

(1) Example of procedure for RF transmission output power setting

<1> See tables according to ”Frequency band identifier” in IEEE802.15.4g frequency/data rate 31.7.7 IEEE802.15.4g frequency/data rate table.
When it is 9, **Table 18 - 17** to **Table 18 - 19**. When it is 4, see **Table 18 - 20** to **Table 18 - 22**. When it is 5 to 7, see **Table 18 - 23** to **Table 18 - 25**. When it is 8, see **Table 18 - 26** to **Table 18 - 28**.

<2> Set the data [1:0] of 0090H address according to the table. Fix the data [7:2] at ”111101b”.
When setting the data [1:0] of the 0090H address to 1H, set V_{DDRF}**Note** to 2.4 V or higher.

<3> Set the data [4:0] of the 0092H address according to **Table 18 - 17** to **Table 18 - 28**.
Fix the data [7:5] at "011b".

<4> Set the data [4:0] of 00DCH address according to the table and 19H to address 00DH. Fix the data [7:5] at ”000b”.

<5> Set the data [4:0] of 00DCH address according to the table and 25H to address 00DH. Fix the data [7:5] at ”000b”.

<6> Set the data [7:4] of 00DCH address according to the table and 1AH to address 00DDH. Fix the data [3:0] at ”0000b”.

Figure 18 - 108 to **Figure 18 - 111** shows the relationship between the transmission output power and gain set. The values may vary depending on the load conditions and samples. The combination of the coarse control and fine control enables the wider range of the transmission output power.

<R> **Note** Voltage of VCCRF and VCCDDC pins.

Caution Refer to the latest application note for setting values of Table 18 - 17 to Table 18 - 28 when to use the product.

Table 18 - 17 Gain Set (Frequency band identifier = 9) (1/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19H)	00DCH address Bit [4:0] (00DDH address is set to 25H)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
0	3	00	0	00	0
1	3	00	0	1F	6
2	3	01	0	1F	6
3	3	02	0	1F	6
4	3	03	0	1F	6
5	3	04	0	1F	6
6	3	05	0	1F	6
7	3	06	0	1F	6
8	3	07	0	1F	6
9	3	08	0	1F	6
10	3	09	0	1F	6
11	3	0A	0	1F	6
12	3	0B	0	1F	6
13	3	0C	0	1F	6
14	3	0D	0	1F	6
15	3	0E	0	1F	6
16	3	0F	0	1F	6
17	3	10	0	1F	6

Table 18 - 17 Gain Set (Frequency band identifier = 9) (1/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19H)	00DCH address Bit [4:0] (00DDH address is set to 25H)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
18	3	11	0	1F	6
19	3	12	0	1F	6

Table 18 - 18 Gain Set (Frequency band identifier = 9) (2/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19H)	00DCH address Bit [4:0] (00DDH address is set to 25H)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
20	3	13	0	1F	6
21	3	14	0	1F	6
22	2	15	0	1F	6
23	2	16	0	1F	6
24	2	17	0	1F	6
25	2	18	0	1F	6
26	2	19	0	1F	6
27	2	1A	0	1F	6
28	2	1B	0	1F	6
29	2	1C	0	1F	6
30	2	1D	0	1F	6
31	2	1E	0	1F	6
32	2	1F	0	1F	6
33	3	05	1	15	6
34	3	06	1	15	6
35	3	07	1	15	6
36	3	08	1	15	6
37	3	09	1	15	6
38	3	0A	1	15	6
39	3	0B	1	15	6
40	3	0C	1	15	6
41	3	0D	1	15	6
42	3	0E	1	15	6
43	3	0F	1	15	6
44	3	10	1	15	6
45	3	11	1	15	6
46	3	12	1	15	6
47	3	13	1	15	6
48	2	14	1	15	6
49	2	15	1	15	6
50	2	16	1	15	6
51	2	17	1	15	6
52	2	18	1	15	6
53	2	19	1	15	6
54	2	1A	1	15	6
55	2	1B	1	15	6

Table 18 - 18 Gain Set (Frequency band identifier = 9) (2/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19H)	00DCH address Bit [4:0] (00DDH address is set to 25H)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
56	2	1C	1	15	6
57	2	1D	1	15	6
58	2	1E	1	15	6
59	2	1F	1	15	6
60	3	07	2	1F	6
61	3	08	2	1F	6

Table 18 - 19 Gain Set (Frequency band identifier = 9) (3/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19H)	00DCH address Bit [4:0] (00DDH address is set to 25H)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
62	3	09	2	1F	6
63	3	0A	2	1F	6
64	3	0B	2	1F	6
65	3	0C	2	1F	6
66	3	0D	2	1F	6
67	3	0E	2	1F	6
68	3	0F	2	1F	6
69	3	10	2	1F	6
70	3	11	2	1F	6
71	3	12	2	1F	6
72	3	13	2	1F	6
73	3	14	2	1F	6
74	2	15	2	1F	6
75	2	16	2	1F	6
76	2	17	2	1F	6
77	2	18	2	1F	6
78	2	19	2	1F	6
79	2	1A	2	1F	6
80	2	1B	2	1F	6
81	2	1C	2	1F	6
82	2	1D	2	1F	6
83	2	1E	2	1F	6
84	2	1F	2	1F	6
85	3	11	4	16	6
86	3	12	4	16	6
87	3	13	4	16	6
88	3	14	4	16	6
89	3	11	6	16	6
90	3	12	6	16	6
91	3	13	6	16	6
92	3	14	6	16	6
93	2	15	7	15	6

Table 18 - 19 Gain Set (Frequency band identifier = 9) (3/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19H)	00DCH address Bit [4:0] (00DDH address is set to 25H)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
94	2	16	7	15	6
95	2	17	7	15	6
96	1	1B	7	15	6
97	1	1C	7	15	6
98	1	1D	7	15	6
99	1	1E	7	15	6
100	1	1E	8	11	6
101	1	1F	8	11	6

Figure 18 - 108 Relationship between Transmission Output Power and Gain Set

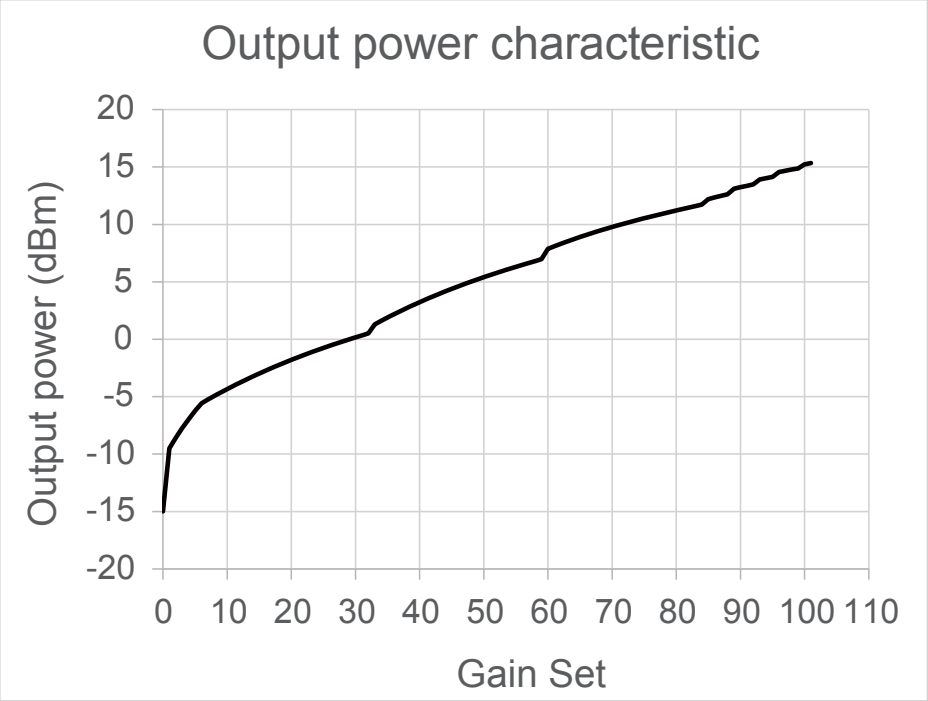


Table 18 - 20 Gain Set (Frequency band identifier = 4) (1/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19H)	00DCH address Bit [4:0] (00DDH address is set to 25H)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
0	3	00	0	00	0
1	3	00	0	1F	7
2	3	01	0	1F	7
3	3	02	0	1F	7
4	3	03	0	1F	7
5	3	04	0	1F	7
6	3	05	0	1F	7
7	3	06	0	1F	7
8	3	07	0	1F	7
9	3	08	0	1F	7
10	3	09	0	1F	7
11	3	0A	0	1F	7
12	3	0B	0	1F	7
13	3	0C	0	1F	7
14	3	0D	0	1F	7
15	3	0E	0	1F	7
16	3	0F	0	1F	7
17	3	10	0	1F	7
18	3	11	0	1F	7
19	3	12	0	1F	7
20	3	13	0	1F	7
21	3	14	0	1F	7
22	2	15	0	1F	7
23	2	16	0	1F	7
24	2	17	0	1F	7
25	2	18	0	1F	7
26	2	19	0	1F	7
27	2	1A	0	1F	7
28	2	1B	0	1F	7
29	2	1C	0	1F	7
30	2	1D	0	1F	7
31	2	1E	0	1F	7
32	2	1F	0	1F	7
33	3	06	1	0F	7
34	3	07	1	0F	7
35	3	08	1	0F	7
36	3	09	1	0F	7
37	3	0A	1	0F	7
38	3	0B	1	0F	7
39	3	0C	1	0F	7

Table 18 - 21 Gain Set (Frequency band identifier = 4) (2/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19H)	00DCH address Bit [4:0] (00DDH address is set to 25H)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
40	3	0D	1	0F	7
41	3	0E	1	0F	7
42	3	0F	1	0F	7
43	3	10	1	0F	7
44	3	11	1	0F	7
45	3	12	1	0F	7
46	3	13	1	0F	7
47	3	14	1	0F	7
48	2	15	1	0F	7
49	2	16	1	0F	7
50	2	17	1	0F	7
51	2	18	1	0F	7
52	2	19	1	0F	7
53	2	1A	1	0F	7
54	2	1B	1	0F	7
55	2	1C	1	0F	7
56	2	1D	1	0F	7
57	2	1E	1	0F	7
58	2	1F	1	0F	7
59	3	05	2	1B	7
60	3	06	2	1B	7
61	3	07	2	1B	7
62	3	08	2	1B	7
63	3	09	2	1B	7
64	3	0A	2	1B	7
65	3	0B	2	1B	7
66	3	0C	2	1B	7
67	3	0D	2	1B	7
68	3	0E	2	1B	7
69	3	0F	2	1B	7
70	3	10	2	1B	7
71	3	11	2	1B	7
72	3	12	2	1B	7
73	3	13	2	1B	7
74	3	14	2	1B	7
75	3	0C	3	1F	7
76	3	0D	3	1F	7
77	3	0E	3	1F	7
78	3	0F	3	1F	7
79	3	10	3	1F	7

Table 18 - 22 Gain Set (Frequency band identifier = 4) (3/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19H)	00DCH address Bit [4:0] (00DDH address is set to 25H)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
80	3	11	3	1F	7
81	3	12	3	1F	7
82	3	13	3	1F	7
83	3	14	3	1F	7
84	3	11	4	1B	7
85	3	12	4	1B	7
86	3	13	4	1B	7
87	3	14	4	1B	7
88	3	11	6	1B	7
89	3	12	6	1B	7
90	3	13	6	1B	7
91	3	14	6	1B	7
92	1	14	7	16	7
93	1	15	7	16	7
94	1	16	7	16	7
95	1	17	7	16	7
96	1	18	7	16	7
97	1	19	7	16	7
98	1	1A	7	16	7
99	1	1B	7	16	7
100	1	1C	7	16	7
101	1	1E	8	15	7
102	1	1F	8	15	7

Figure 18 - 109 Relationship between Transmission Output Power and Gain Set

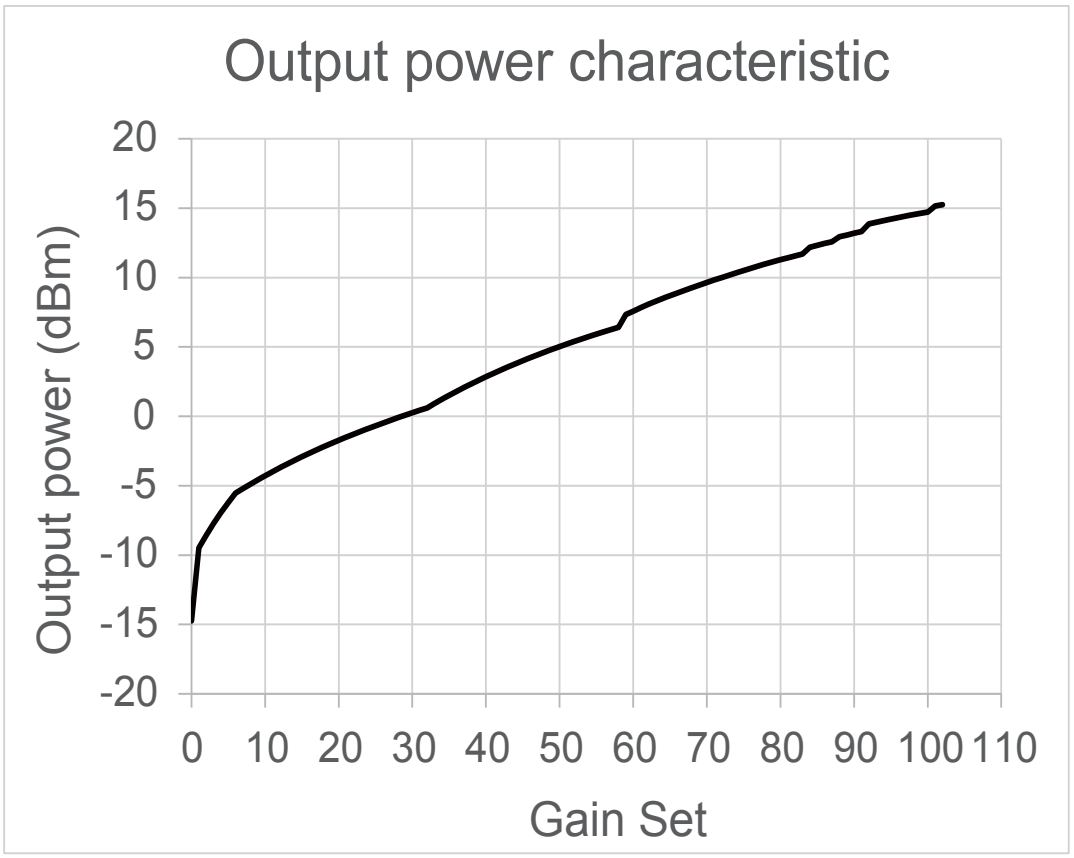


Table 18 - 23 Gain Set (Frequency band identifier = 5 to 7) (1/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19)	00DCH address Bit [4:0] (00DDH address is set to 25)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
0	3	00	0	00	0
1	3	00	0	1F	6
2	3	01	0	1F	6
3	3	02	0	1F	6
4	3	03	0	1F	6
5	3	04	0	1F	6
6	3	05	0	1F	6
7	3	06	0	1F	6
8	3	07	0	1F	6
9	3	08	0	1F	6
10	3	09	0	1F	6
11	3	0A	0	1F	6
12	3	0B	0	1F	6
13	3	0C	0	1F	6
14	3	0D	0	1F	6
15	3	0E	0	1F	6
16	3	0F	0	1F	6
17	3	10	0	1F	6
18	3	11	0	1F	6
19	3	12	0	1F	6
20	3	13	0	1F	6
21	3	14	0	1F	6
22	2	15	0	1F	6
23	2	16	0	1F	6
24	2	17	0	1F	6
25	2	18	0	1F	6
26	2	19	0	1F	6
27	2	1A	0	1F	6
28	2	1B	0	1F	6
29	2	1C	0	1F	6
30	2	1D	0	1F	6
31	2	1E	0	1F	6
32	2	1F	0	1F	6
33	3	05	1	15	6
34	3	06	1	15	6
35	3	07	1	15	6
36	3	08	1	15	6
37	3	09	1	15	6
38	3	0A	1	15	6
39	3	0B	1	15	6
40	3	0C	1	15	6

Table 18 - 24 Gain Set (Frequency band identifier = 5 to 7) (2/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19)	00DCH address Bit [4:0] (00DDH address is set to 25)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
41	3	0D	1	15	6
42	3	0E	1	15	6
43	3	0F	1	15	6
44	3	10	1	15	6
45	3	11	1	15	6
46	3	12	1	15	6
47	3	13	1	15	6
48	3	14	1	15	6
49	2	15	1	15	6
50	2	16	1	15	6
51	2	17	1	15	6
52	2	18	1	15	6
53	2	19	1	15	6
54	2	1A	1	15	6
55	2	1B	1	15	6
56	2	1C	1	15	6
57	2	1D	1	15	6
58	2	1E	1	15	6
59	2	1F	1	15	6
60	3	06	2	1F	6
61	3	07	2	1F	6
62	3	08	2	1F	6
63	3	09	2	1F	6
64	3	0A	2	1F	6
65	3	0B	2	1F	6
66	3	0C	2	1F	6
67	3	0D	2	1F	6
68	3	0E	2	1F	6
69	3	0F	2	1F	6
70	3	10	2	1F	6
71	3	11	2	1F	6
72	3	12	2	1F	6
73	3	13	2	1F	6
74	3	14	2	1F	6
75	2	15	2	1F	6
76	2	16	2	1F	6
77	2	17	2	1F	6
78	2	18	2	1F	6
79	2	19	2	1F	6
80	2	1A	2	1F	6

Table 18 - 25 Gain Set (Frequency band identifier = 5 to 7) (3/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19)	00DCH address Bit [4:0] (00DDH address is set to 25)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
81	2	1B	2	1F	6
82	2	1C	2	1F	6
83	2	1D	2	1F	6
84	2	1E	2	1F	6
85	2	1F	2	1F	6
86	3	11	4	16	6
87	3	12	4	16	6
88	3	13	4	16	6
89	3	14	4	16	6
90	3	11	6	16	6
91	3	12	6	16	6
92	3	13	6	16	6
93	3	14	6	16	6
94	1	14	7	14	6
95	1	15	7	14	6
96	1	16	7	14	6
97	1	17	7	14	6
98	1	18	7	14	6
99	1	19	7	14	6
100	1	1A	7	14	6
101	1	1B	7	14	6
102	1	1C	7	14	6
103	1	1D	8	11	6
104	1	1E	8	11	6

Figure 18 - 110 Relationship between Transmission Output Power and Gain Set

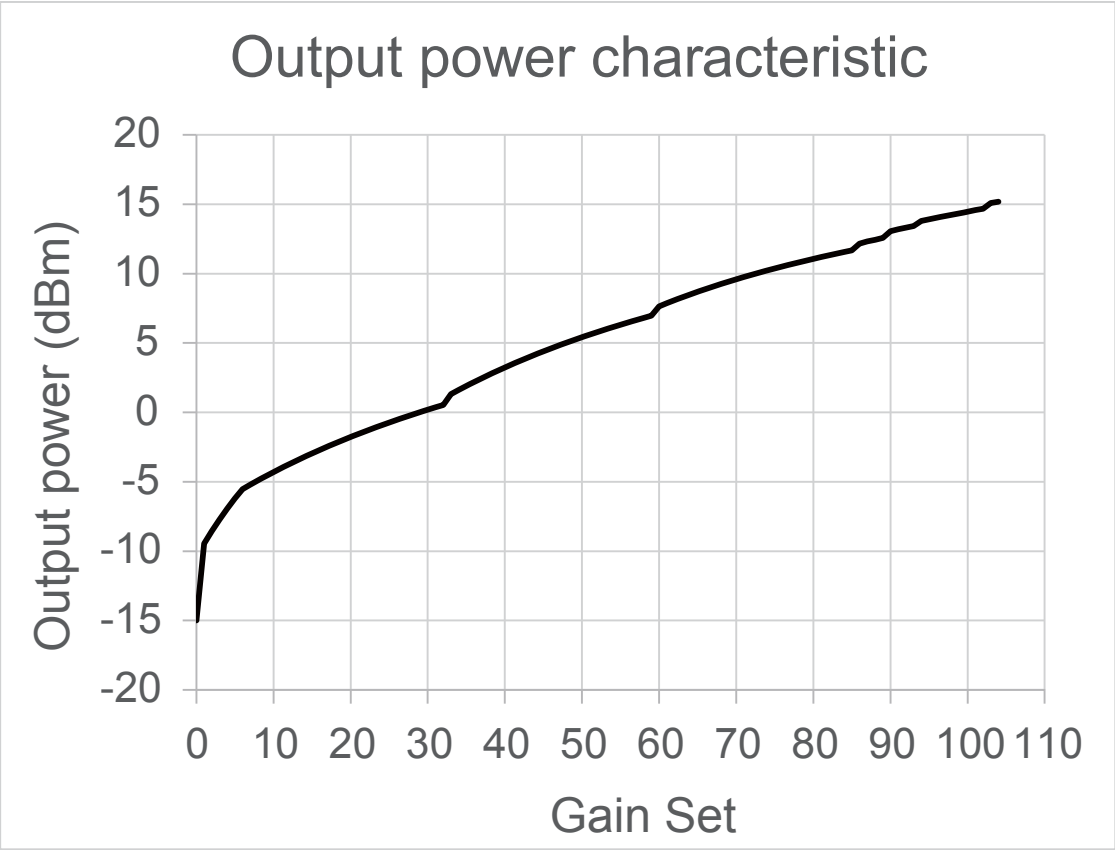


Table 18 - 26 Gain Set (Frequency band identifier = 8) (1/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19)	00DCH address Bit [4:0] (00DDH address is set to 25)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
0	3	00	0	00	0
1	3	00	0	1F	6
2	3	01	0	1F	6
3	3	02	0	1F	6
4	3	03	0	1F	6
5	3	04	0	1F	6
6	3	05	0	1F	6
7	3	06	0	1F	6
8	3	07	0	1F	6
9	3	08	0	1F	6
10	3	09	0	1F	6
11	3	0A	0	1F	6
12	3	0B	0	1F	6
13	3	0C	0	1F	6
14	3	0D	0	1F	6
15	3	0E	0	1F	6
16	3	0F	0	1F	6
17	3	10	0	1F	6
18	3	11	0	1F	6
19	3	12	0	1F	6
20	3	13	0	1F	6
21	3	14	0	1F	6
22	2	15	0	1F	6
23	2	16	0	1F	6
24	2	17	0	1F	6
25	2	18	0	1F	6
26	2	19	0	1F	6
27	2	1A	0	1F	6
28	2	1B	0	1F	6
29	2	1C	0	1F	6
30	2	1D	0	1F	6
31	2	1E	0	1F	6
32	2	1F	0	1F	6
33	3	05	1	15	6
34	3	06	1	15	6
35	3	07	1	15	6
36	3	08	1	15	6
37	3	09	1	15	6
38	3	0A	1	15	6
39	3	0B	1	15	6
40	3	0C	1	15	6

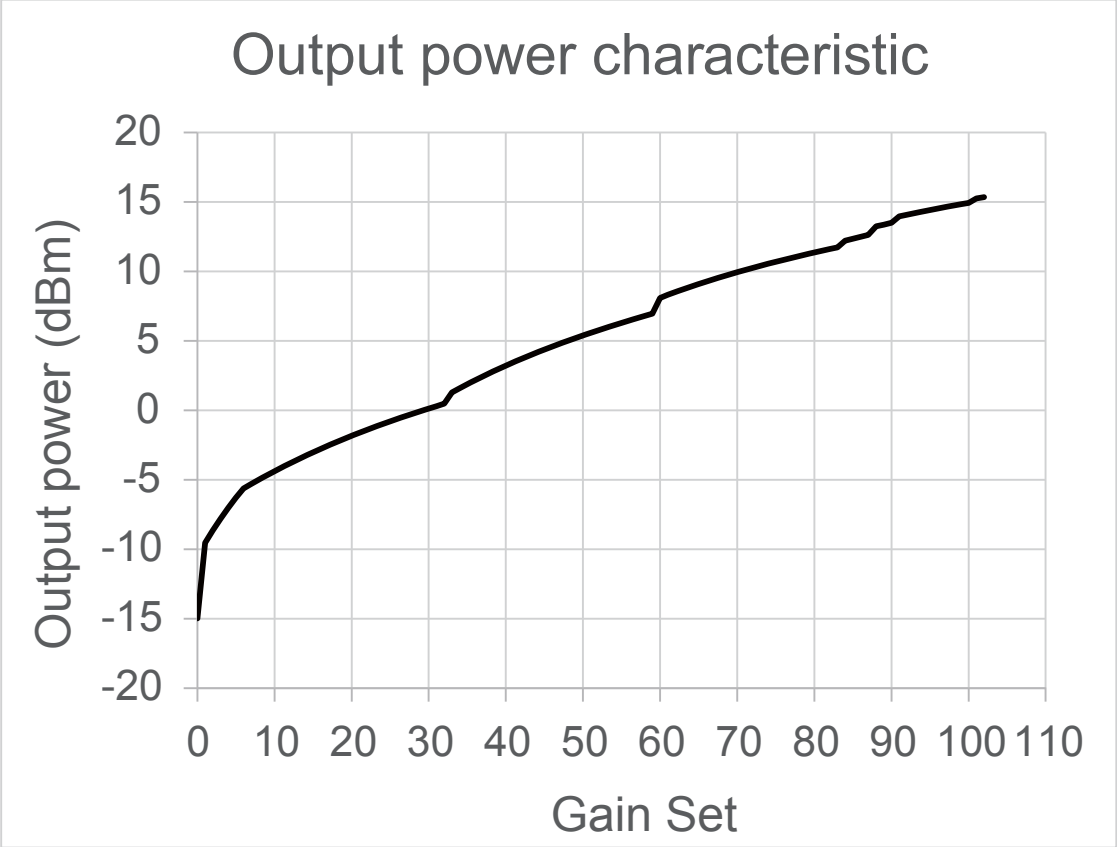
Table 18 - 27 Gain Set (Frequency band identifier = 8) (2/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19)	00DCH address Bit [4:0] (00DDH address is set to 25)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
41	3	0D	1	15	6
42	3	0E	1	15	6
43	3	0F	1	15	6
44	3	10	1	15	6
45	3	11	1	15	6
46	3	12	1	15	6
47	3	13	1	15	6
48	3	14	1	15	6
49	2	15	1	15	6
50	2	16	1	15	6
51	2	17	1	15	6
52	2	18	1	15	6
53	2	19	1	15	6
54	2	1A	1	15	6
55	2	1B	1	15	6
56	2	1C	1	15	6
57	2	1D	1	15	6
58	2	1E	1	15	6
59	2	1F	1	15	6
60	3	08	2	1F	6
61	3	09	2	1F	6
62	3	0A	2	1F	6
63	3	0B	2	1F	6
64	3	0C	2	1F	6
65	3	0D	2	1F	6
66	3	0E	2	1F	6
67	3	0F	2	1F	6
68	3	10	2	1F	6
69	3	11	2	1F	6
70	3	12	2	1F	6
71	3	13	2	1F	6
72	3	14	2	1F	6
73	2	15	2	1F	6
74	2	16	2	1F	6
75	2	17	2	1F	6
76	2	18	2	1F	6
77	2	19	2	1F	6
78	2	1A	2	1F	6
79	2	1B	2	1F	6
80	2	1C	2	1F	6

Table 18 - 28 Gain Set (Frequency band identifier = 8) (3/3)

Gain set	0090H address Bit [1:0]	0092H address Bit [4:0]	00DCH address Bit [4:0] (00DDH address is set to 19)	00DCH address Bit [4:0] (00DDH address is set to 25)	00DCH address Bit [7:4] (00DDH address is set to 1AH)
81	2	1D	2	1F	6
82	2	1E	2	1F	6
83	2	1F	2	1F	6
84	3	11	4	16	6
85	3	12	4	16	6
86	3	13	4	16	6
87	3	14	4	16	6
88	3	12	6	16	6
89	3	13	6	16	6
90	3	14	6	16	6
91	1	15	7	14	6
92	1	16	7	14	6
93	1	17	7	14	6
94	1	18	7	14	6
95	1	19	7	14	6
96	1	1A	7	14	6
97	1	1B	7	14	6
98	1	1C	7	14	6
99	1	1D	7	14	6
100	1	1E	7	14	6
101	1	1E	8	11	6
102	1	1F	8	11	6

Figure 18 - 111 Relationship between Transmission Output Power and Gain Set (Frequency band identifier = 8)



(2) RF frequency setting for transmission

Following descriptions are detail of <2> frequency setting in 18.7.1 (1) Example of procedure for RF transmission.

Set BBFREQ register (54) in 18.4.3 according to RF frequency of transmission.
If the frequency setting is integer times N (864 MHz, 912 MHz) of XIN frequency (48 MHz), the transmission character will degrade. If the frequency matches the following condition, it must be changed to other frequency. N is integer, h is modulation index in 2-level FSK and three times of modulation index in 4-level FSK. R is symbol rate.

48e6 *N - (M2+1.5*R) ≤ Use prohibited transmission frequency [Hz] ≤ 48e6 *N + (M2+1.5*R)
Here, M2 = 3*R*(1+h)

Caution Refer to the latest application note for this setting when to use the product.

(3) RF frequency setting for reception

Following descriptions are detail of <2> frequency setting in 18.7.1 (2) Example of procedure for RF reception.

<1> Set 18.4.3 (53) BBFREQ register according to the RF frequency of reception.

<2> Set the data [7:0] of 0095H address following to the **Table 18 - 29** to **18 - 50** according to the RF frequency of reception.

<3> Set data of 0095H address in IDLE mode.

<4> Frequencies 864 MHz ± 400 kHz and 912 MHz ± 400 kHz are use prohibited.

<5> Set the data [7:0] of 0095H address to 08H for the RF frequency of transmission.
Set also data of 0095H address in IDLE mode for the RF frequency of transmission.

Caution Refer to the latest application note for setting values of Table 18 - 29 to Table 18 - 50 when to use the product.

Table 18 - 29 RF Frequency Set (Frequency band identifier = 4, Mode = 001)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	863.0000	863.5875	8H
2	863.6000	864.4000	Prohibited
3	864.4125	865.8500	8H
4	865.8625	866.1375	28H
5	866.1500	867.8500	8H
6	867.8625	868.1375	28H
7	868.1500	869.8500	8H
8	869.8625	870.0000	28H

Table 18 - 30 RF Frequency Set (Frequency band identifier = 4, Mode = 002)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	863.0000	863.5875	8H
2	863.6000	864.4000	Prohibited
3	864.4125	865.7125	8H
4	865.7250	865.9750	48H
5	865.9875	866.2750	28H
6	866.2875	867.7125	8H
7	867.7250	867.9750	48H
8	867.9875	868.2750	28H
9	868.2875	869.7125	8H
10	869.7250	870.0000	28H

Table 18 - 31 RF Frequency Set (Frequency band identifier = 4, Mode = 003)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	863.0000	863.5875	8H
2	863.6000	864.4000	Prohibited
3	864.4125	865.7500	8H
4	865.7625	865.9375	48H
5	865.9500	866.2375	28H
6	866.2500	867.7500	8H
7	867.7625	867.9375	48H
8	867.9500	868.2375	28H
9	868.2500	869.7500	8H
10	869.7625	870.0000	28H

Table 18 - 32 RF Frequency Set (Frequency band identifier = 5, Mode = 004)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	896.0000	896.0125	28H
2	896.0250	896.1750	8H
3	896.1875	896.2125	28H
4	896.2250	897.9750	8H
5	897.9875	898.0125	28H
6	898.0250	899.7750	8H
7	899.7875	899.8125	28H
8	899.8250	899.9750	8H
9	899.9875	900.0125	28H
10	900.0250	900.1750	8H
11	900.1875	900.2125	28H
12	900.2250	901.0000	8H

Table 18 - 33 RF Frequency Set (Frequency band identifier = 5, Mode = 005)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	896.0000	896.0375	28H
2	896.0500	896.3500	8H
3	896.3625	896.4375	28H
4	896.4500	897.9500	8H
5	897.9625	898.0375	28H
6	898.0500	899.5500	8H
7	899.5625	899.6375	28H
8	899.6500	899.9500	8H
9	899.9625	900.0375	28H
10	900.0500	900.3500	8H
11	900.3625	900.4375	28H
12	900.4500	901.0000	8H

Table 18 - 34 RF Frequency Set (Frequency band identifier = 5, Mode = 006)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	896.0000	896.0625	28H
2	896.0750	896.7250	8H
3	896.7375	896.8625	28H
4	896.8750	897.9250	8H
5	897.9375	897.9500	28H
6	897.9625	898.0625	48H
7	898.0750	899.1250	8H
8	899.1375	899.2625	28H
9	899.2750	899.9250	8H
10	899.9375	900.0375	28H
11	900.0500	900.0625	48H
12	900.0750	900.7250	8H
13	900.7375	900.8375	28H
14	900.8500	901.0000	8H

Table 18 - 35 RF Frequency Set (Frequency band identifier = 6, Mode = 007)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	901.0000	901.9750	8H
2	901.9875	902.0000	28H

Table 18 - 36 RF Frequency Set (Frequency band identifier = 6, Mode = 008)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	901.0000	901.9500	8H
2	901.9625	902.0000	28H

Table 18 - 37 RF Frequency Set (Frequency band identifier = 6, Mode = 009)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	901.0000	901.9250	8H
2	901.9375	902.0000	28H

Table 18 - 38 RF Frequency Set (Frequency band identifier = 7, Mode = 010)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	902.0000	902.1375	28H
2	902.1500	903.8500	8H
3	903.8625	904.1375	28H
4	904.1500	905.8500	8H
5	905.8625	906.1375	28H
6	906.1500	907.8500	8H
7	907.8625	908.1375	28H
8	908.1500	909.8500	8H
9	909.8625	910.1375	28H
10	910.1500	911.5875	8H
11	911.6000	912.4000	Prohibited
12	912.4125	913.8500	8H
13	913.8625	914.1375	28H
14	914.1500	915.8500	8H
15	915.8625	916.1375	28H
16	916.1500	917.8500	8H
17	917.8625	918.1375	28H
18	918.1500	919.8500	8H
19	919.8625	920.1375	28H
20	920.1500	921.8500	8H
21	921.8625	922.1375	28H
22	922.1500	923.8500	8H
23	923.8625	924.1375	28H
24	924.1500	925.8500	8H
25	925.8625	926.1375	28H
26	926.1500	927.8500	8H
27	927.8625	928.0000	28H

Table 18 - 39 RF Frequency Set (Frequency band identifier = 7, Mode = 011)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	902.0000	902.2625	28H
2	902.2750	903.7250	8H
3	903.7375	904.2625	28H
4	904.2750	905.7250	8H
5	905.7375	906.2625	28H
6	906.2750	907.7250	8H
7	907.7375	908.0375	28H
8	908.0500	908.2625	48H
9	908.2750	909.7250	8H
10	909.7375	910.0375	28H
11	910.0500	910.2625	48H
12	910.2750	911.5875	8H
13	911.6000	912.4000	Prohibited
14	912.4125	913.7250	8H
15	913.7375	913.9500	48H
16	913.9625	914.2625	28H
17	914.2750	915.7250	8H
18	915.7375	915.9500	48H
19	915.9625	916.2625	28H
20	916.2750	917.7250	8H
21	917.7375	918.2625	28H
22	918.2750	919.7250	8H
23	919.7375	920.2625	28H
24	920.2750	921.7250	8H
25	921.7375	922.2625	28H
26	922.2750	923.7250	8H
27	923.7375	924.2625	28H
28	924.2750	925.7250	8H
29	925.7375	926.2625	28H
30	926.2750	927.7250	8H
31	927.7375	928.0000	28H

Table 18 - 40 RF Frequency Set (Frequency band identifier = 7, Mode = 012)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	902.0000	902.3500	28H
2	902.3625	903.6375	8H
3	903.6500	904.2625	28H
4	904.2750	904.3500	48H
5	904.3625	905.6375	8H
6	905.6500	906.2625	28H
7	906.2750	906.3500	48H
8	906.3625	907.6375	8H
9	907.6500	907.9500	28H
10	907.9625	908.2125	48H
11	908.2250	908.3500	68H
12	908.3625	909.6375	8H
13	909.6500	909.9500	28H
14	909.9625	910.2125	48H
15	910.2250	911.5875	8H
16	911.6000	912.4000	Prohibited
17	912.4125	913.7750	8H
18	913.7875	914.0375	48H
19	914.0500	914.3500	28H
20	914.3625	915.6375	8H
21	915.6500	915.7750	68H
22	915.7875	916.0375	48H
23	916.0500	916.3500	28H
24	916.3625	917.6375	8H
25	917.6500	917.7250	48H
26	917.7375	918.3500	28H
27	918.3625	919.6375	8H
28	919.6500	919.7250	48H
29	919.7375	920.3500	28H
30	920.3625	921.6375	8H
31	921.6500	922.3500	28H
32	922.3625	923.6375	8H
33	923.6500	924.3500	28H
34	924.3625	925.6375	8H
35	925.6500	926.3500	28H
36	926.3625	927.6375	8H
37	927.6500	928.0000	28H

Table 18 - 41 RF Frequency Set (Frequency band identifier = 8, Mode = 013)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	917.0000	917.8500	8H
2	917.8625	918.1375	28H
3	918.1500	919.8500	8H
4	919.8625	920.1375	28H
5	920.1500	921.8500	8H
6	921.8625	922.1375	28H
7	922.1500	923.5000	8H

Table 18 - 42 RF Frequency Set (Frequency band identifier = 8, Mode = 014)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	917.0000	917.7250	8H
2	917.7375	918.2625	28H
3	918.2750	919.7250	8H
4	919.7375	920.2625	28H
5	920.2750	921.7250	8H
6	921.7375	922.2625	28H
7	922.2750	923.5000	8H

Table 18 - 43 RF Frequency Set (Frequency band identifier = 8, Mode = 015)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	917.0000	917.6375	8H
2	917.6500	917.7250	48H
3	917.7375	918.3500	28H
4	918.3625	919.6375	8H
5	919.6500	919.7250	48H
6	919.7375	920.3500	28H
7	920.3625	921.6375	8H
8	921.6500	922.3500	28H
9	922.3625	923.5000	8H

Table 18 - 44 RF Frequency Set (Frequency band identifier = 8, Mode = 016)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	920.0000	920.1375	28H
2	920.1500	921.8500	8H
3	921.8625	922.1375	28H
4	922.1500	923.8500	8H
5	923.8625	924.1375	28H
6	924.1500	925.8500	8H
7	925.8625	926.1375	28H
8	926.1500	927.8500	8H
9	927.8625	928.0000	28H

Table 18 - 45 RF Frequency Set (Frequency band identifier = 9, Mode = 000)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	920.0000	920.2750	28H
2	920.2875	921.7125	8H
3	921.7250	922.2750	28H
4	922.2875	923.7125	8H
5	923.7250	924.2750	28H
6	924.2875	925.7125	8H
7	925.7250	926.2750	28H
8	926.2875	927.7125	8H
9	927.7250	928.0000	28H

Table 18 - 46 RF Frequency Set (Frequency band identifier = 9, Mode = 017)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	920.0000	920.5375	28H
2	920.5500	921.4500	8H
3	921.4625	921.6125	48H
4	921.6250	922.5250	28H
5	922.5375	923.4500	8H
6	923.4625	923.6125	48H
7	923.6250	924.5250	28H
8	924.5375	924.5375	48H
9	924.5500	925.4500	8H
10	925.4625	926.2125	28H
11	926.2250	926.2500	68H
12	926.2625	926.5375	48H
13	926.5500	927.4500	8H
14	927.4625	928.0000	28H

Table 18 - 47 RF Frequency Set (Frequency band identifier = 9, Mode = 018)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	920.0000	920.4750	28H
2	920.4875	921.5125	8H
3	921.5250	921.5500	48H
4	921.5625	922.4750	28H
5	922.4875	923.5125	8H
6	923.5250	923.5500	48H
7	923.5625	924.4750	28H
8	924.4875	925.5125	8H
9	925.5250	926.2750	28H
10	926.2875	926.4750	48H
11	926.4875	927.5125	8H
12	927.5250	928.0000	28H

Table 18 - 48 RF Frequency Set (Frequency band identifier = Other, Mode = 019)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	920.0000	920.5125	28H
2	920.5250	921.4750	8H
3	921.4875	921.5875	48H
4	921.6000	922.5125	28H
5	922.5250	923.4750	8H
6	923.4875	923.5875	48H
7	923.6000	924.5125	28H
8	924.5250	925.4750	8H
9	925.4875	926.2375	28H
10	926.2500	926.5125	48H
11	926.5250	927.4750	8H
12	927.4875	928.0000	28H

Table 18 - 49 RF Frequency Set (Frequency band identifier = Other, Mode = 020)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	902.0000	902.1750	28H
2	902.1875	903.8125	8H
3	903.8250	904.1750	28H
4	904.1875	905.8125	8H
5	905.8250	906.1750	28H
6	906.1875	907.8125	8H
7	907.8250	908.1250	28H
8	908.1375	908.1750	48H
9	908.1875	909.8125	8H
10	909.8250	910.1250	28H
11	910.1375	910.1750	48H
12	910.1875	911.5875	8H
13	911.6000	912.4000	Prohibited
14	912.4125	913.8125	8H
15	913.8250	913.8625	48H
16	913.8750	914.1750	28H
17	914.1875	915.8125	8H
18	915.8250	915.8625	48H
19	915.8750	916.1750	28H
20	916.1875	917.8125	8H
21	917.8250	918.1750	28H
22	918.1875	919.8125	8H
23	919.8250	920.1750	28H
24	920.1875	921.8125	8H
25	921.8250	922.1750	28H
26	922.1875	923.8125	8H
27	923.8250	924.1750	28H
28	924.1875	925.8125	8H
29	925.8250	926.1750	28H
30	926.1875	927.8125	8H
31	927.8250	928.0000	28H

Table 18 - 50 RF Frequency Set (Frequency band identifier = Other, Mode = 021)

RF frequency set	Lower limit RF frequency [MHz]	Upper limit RF frequency [MHz]	0095H address Bit [7:0]
1	902.0000	902.3875	28H
2	902.4000	902.5250	48H
3	902.5375	903.4625	8H
4	903.4750	904.0875	28H
5	904.1000	904.5250	48H
6	904.5375	905.4625	8H
7	905.4750	906.0875	28H
8	906.1000	906.5250	48H
9	906.5375	907.4625	8H
10	907.4750	907.7750	28H
11	907.7875	908.0375	48H
12	908.0500	908.1500	8H
13	908.1625	908.5250	68H
14	908.5375	909.4625	8H
15	909.4750	909.7750	28H
16	909.7875	910.0375	48H
17	910.0500	911.5875	8H
18	911.6000	912.4000	Prohibited
19	912.4125	913.9500	8H
20	913.9625	914.2125	48H
21	914.2250	914.5250	28H
22	914.5375	915.4625	8H
23	915.4750	915.8375	68H
24	915.8500	915.9500	8H
25	915.9625	916.2125	48H
26	916.2250	916.5250	28H
27	916.5375	917.4625	8H
28	917.4750	917.9000	48H
29	917.9125	918.5250	28H
30	918.5375	919.4625	8H
31	919.4750	919.9000	48H
32	919.9125	920.5250	28H
33	920.5375	921.4625	8H
34	921.4750	921.6000	48H
35	921.6125	922.5250	28H
36	922.5375	923.4625	8H
37	923.4750	923.6000	48H
38	923.6125	924.5250	28H
39	924.5375	925.4625	8H
40	925.4750	926.2375	28H
41	926.2500	926.5250	48H
42	926.5375	927.4625	8H
43	927.4750	928.0000	28H

18.7.3 Setting for each data rate

The list of each data rate is shown in **Table 18 - 51** and the settings required for each data rate is listed in Table 18 - 52 to Table 18 - 59. Set values to addresses listed in Table 18 - 52 to Table 18 - 59 from the left in order according to the desired data rate in IDLE mode.

Caution Refer to the latest application note for setting data in Table 18 – 52 to Table 18 - 59 when to use the product.

Table 18 - 51 Each Data Rate List

Frequency band identifier	PHY	Frequency band (MHz)	Modulation	Mode	Data rate (kbps)	Modulation index	Channel spacing (kHz)
4	863 MHz (Europe)	863 to 870	2FSK/2GFSK	001	50	1	200
				002	100		400
			4FSK/4GFSK	003	200	0.33	
5	896 MHz (US)	896 to 901	2FSK/2GFSK	004	10	0.5	25
				005	20		50
				006	40		100
6	901 MHz (US)	901 to 902	2FSK/2GFSK	007	10	0.5	25
				008	20		50
				009	40		100
7	915 MHz (US)	902 to 928	2FSK/2GFSK	010	50	1	200
				011	150	0.5	400
				012	200		
8	917 MHz (Korea)	917 to 923.5	2FSK/2GFSK	013	50	1	200
				014	150	0.5	400
				015	200		
9	920 MHz (Japan)	920 to 928	2FSK/2GFSK	016	50	1	200
				000	100		400
				017	200		600
			4FSK/4GFSK	018	400	0.33	
Other		920 to 928	2GFSK	019	300	0.5	600
		902 to 928	2GFSK	020	100	0.5	200
		902 to 928	2GFSK	021	300	0.5	600

Table 18 - 52 Settings Required for Each Data Rate (1/9)

Frequency band identifier	PHY	Mode	Data rate (kbps)	Address (H)								
				000E	00B1	00B2		00AC	00AD	00CC	00CE	
						Diversity: No	Diversity: Yes				Diversity: No	Diversity: Yes
4	863 MHz (Europe)	001	50	A3	42	0C	0C	E0	01	01	0C	0C
		002	100	A3	42	20	11	F0	00	00	20	11
		003	200	A1	83	20	11	F0	00	00	20	11
5	896 MHz (US)	004	10	A3	02	0C	0C	60	09	00	0C	0C
		005	20	A3	02	0C	0C	B0	04	01	0C	0C
		006	40	A3	02	0C	0C	58	02	01	0C	0C
6	901 MHz (US)	007	10	A3	02	0C	0C	60	09	00	0C	0C
		008	20	A3	02	0C	0C	B0	04	00	0C	0C
		009	40	A3	02	0C	0C	58	02	01	0C	0C
7	915 MHz (US)	010	50	A3	42	0C	0C	E0	01	01	0C	0C
		011	150	A2	02	20	10	A0	00	00	20	10
		012	200	A1	02	2E	18	78	00	00	2E	18
8	917 MHz (Korea)	013	50	A3	42	0C	0C	E0	01	01	0C	0C
		014	150	A2	02	20	10	A0	00	00	20	10
		015	200	A1	02	2E	18	78	00	00	2E	18
9	920 MHz (Japan)	016	50	A3	42	0C	0C	E0	01	01	0C	0C
		000	100	A3	42	20	11	F0	00	01	20	11
		017	200	A1	42	2E	18	78	00	02	2E	18
		018	400	A1	83	2E	18	78	00	02	2E	18
Other		019	300	A2	02	3E	2A	50	00	02	3E	2A
		020	100	A3	02	20	11	F0	00	01	20	11
		021	300	A2	02	3E	2A	50	00	02	3E	2A

Table 18 - 53 Settings Required for Each Data Rate (2/9)

Frequency band identifier	PHY	Mode	Data rate (kbps)	Address (H)									
				0044		00C0	00C2	00C3	00C4	00C5	0110		00C6
				FEC: Enable	FEC: Disable						FEC: Enable	FEC: Disable	
4	863 MHz (Europe)	001	50	86	8E	AA	09	72	00	00	86	8E	02
		002	100	8D	95	AA	09	72	00	00	90	98	02
		003	200	8D	95	EE	EB	AA	AE	BF	90	98	06
5	896 MHz (US)	004	10	83	8B	AA	09	72	00	00	83	8B	02
		005	20	87	8F	AA	09	72	00	00	87	8F	02
		006	40	8B	93	AA	09	72	00	00	8B	93	02
6	901 MHz (US)	007	10	83	8B	AA	09	72	00	00	83	8B	02
		008	20	87	8F	AA	09	72	00	00	87	8F	02
		009	40	8B	93	AA	09	72	00	00	8B	93	02
7	915 MHz (US)	010	50	8C	94	AA	09	72	00	00	8C	94	02
		011	150	91	99	AA	09	72	00	00	91	99	02
		012	200	90	98	AA	09	72	00	00	91	99	02
8	917 MHz (Korea)	013	50	8C	94	AA	09	72	00	00	8C	94	02
		014	150	91	99	AA	09	72	00	00	91	99	02
		015	200	90	98	AA	09	72	00	00	91	99	02
9	920 MHz (Japan)	016	50	8C	94	AA	09	72	00	00	8C	94	02
		000	100	8D	95	AA	09	72	00	00	90	98	02
		017	200	92	9A	AA	09	72	00	00	91	99	02
		018	400	8C	94	EE	EB	AA	AE	BF	91	99	06
Other		019	300	94	9C	AA	09	72	00	00	90	98	02
		020	100	8D	95	AA	09	72	00	00	90	98	02
		021	300	94	9C	AA	09	72	00	00	90	98	02

Table 18 - 54 Settings Required for Each Data Rate (3/9)

Frequency band identifier	PHY	Mode	Data rate (kbps)	Address (H)									
				043A	042D		0430	0432	0457		0458		0422
					Diversity: No	Diversity: Yes			FEC: Enable	FEC: Disable	FEC: Enable	FEC: Disable	
4	863 MHz (Europe)	001	50	96	0B	09	0C	18	91	D5	00	01	01
		002	100	96	0A	0A	02	14	6C	A4	01	02	01
		003	200	96	0C	0A	02	0C	E6	46	00	03	09
5	896 MHz (US)	004	10	90	06	0A	08	78	73	B7	00	01	09
		005	20	96	3C	3C	0C	3C	91	21	00	02	09
		006	40	96	1E	1E	0C	1E	E6	40	00	03	09
6	901 MHz (US)	007	10	90	06	0A	08	78	73	B7	00	01	09
		008	20	96	3C	3C	0C	3C	91	21	00	02	09
		009	40	96	1E	1E	0C	1E	E6	40	00	03	09
7	915 MHz (US)	010	50	96	0B	09	0C	18	E6	D5	00	03	01
		011	150	96	08	08	02	08	40	F5	02	08	09
		012	200	96	06	06	02	06	91	D5	00	03	09
8	917 MHz (Korea)	013	50	96	0B	09	0C	18	91	D5	00	03	01
		014	150	96	08	08	02	08	40	F5	02	08	09
		015	200	96	06	06	02	06	91	D5	00	03	09
9	920 MHz (Japan)	016	50	96	0B	09	0C	18	91	D5	00	03	01
		000	100	9E	0A	0A	02	14	6C	A4	01	05	01
		017	200	9E	06	06	02	0D	D5	46	02	0C	01
		018	400	96	06	06	02	09	D5	7D	02	0C	09
Other		019	300	96	04	04	02	0D	A7	DD	05	13	09
		020	100	96	0A	0A	02	0C	6C	A7	01	05	09
		021	300	96	04	04	02	0D	47	DD	0D	13	09

Table 18 - 55 Settings Required for Each Data Rate (4/9)

Frequency band identifier	PHY	Mode	Data rate (kbps)	Address (H)										
				0423	0454	0456	0474	047C	047D	047F	0480	0483	0486	0487
4	863 MHz (Europe)	001	50	01	28	02	A6	1B	0F	34	00	3E	55	05
		002	100	01	28	02	A9	1B	0F	34	20	3F	00	00
		003	200	09	28	02	A6	1B	0F	35	00	3F	33	03
5	896 MHz (US)	004	10	09	28	02	A6	1B	0F	34	00	09	02	00
		005	20	09	28	02	A6	1A	0F	34	00	03	32	00
		006	40	09	28	02	A6	1A	0F	34	00	03	32	00
6	901 MHz (US)	007	10	09	28	02	A6	1B	0F	34	00	09	02	00
		008	20	09	28	02	A6	1A	0F	34	00	03	32	00
		009	40	09	28	02	A6	1A	0F	34	00	03	32	00
7	915 MHz (US)	010	50	01	28	02	A6	1B	0F	34	00	3B	55	05
		011	150	09	28	02	A6	1B	0F	34	01	09	00	00
		012	200	09	28	02	A6	1B	0F	34	00	09	00	00
8	917 MHz (Korea)	013	50	01	28	02	A6	1B	0F	34	00	3F	55	05
		014	150	09	28	02	A6	1B	0F	34	01	09	00	00
		015	200	09	28	02	A6	1B	0F	34	00	09	00	00
9	920 MHz (Japan)	016	50	01	28	02	A6	1B	0F	34	00	3C	55	05
		000	100	01	2E	06	A6	1B	0F	34	20	3D	00	00
		017	200	01	28	02	A6	1B	0F	34	01	3C	00	00
		018	400	09	28	02	A7	1B	0F	34	01	3C	00	00
Other		019	300	09	28	02	A7	19	0D	33	02	19	00	00
		020	100	09	28	02	A7	1B	0F	34	00	3D	00	00
		021	300	09	28	02	A7	19	0D	33	06	19	00	00

Table 18 - 56 Settings Required for Each Data Rate (5/9)

Frequency band identifier	PHY	Mode	Data rate (kbps)	Address (H)										
				048D	048F	0491	0492	0493	0494	0581	0582	058F	059F	05A0
4	863 MHz (Europe)	001	50	46	66	20	55	20	55	12	26	FF	FF	3F
		002	100	57	55	20	55	20	55	15	1C	32	56	03
		003	200	46	66	30	55	30	55	10	24	FF	FF	3F
5	896 MHz (US)	004	10	46	66	20	55	20	55	05	30	FF	FF	3F
		005	20	46	55	20	55	20	55	00	30	FF	FF	3F
		006	40	46	55	20	55	20	55	1C	2B	FF	FF	3F
6	901 MHz (US)	007	10	46	66	20	55	20	55	05	30	FF	FF	3F
		008	20	46	55	20	55	20	55	00	30	FF	FF	3F
		009	40	46	55	20	55	20	55	1C	2B	FF	FF	3F
7	915 MHz (US)	010	50	46	66	20	55	20	55	12	26	FF	FF	3F
		011	150	57	55	20	55	20	55	12	26	FF	FF	3F
		012	200	57	55	20	55	20	55	16	2C	FF	FF	3F
8	917 MHz (Korea)	013	50	46	66	20	55	20	55	12	26	FF	FF	3F
		014	150	57	55	20	55	20	55	12	26	FF	FF	3F
		015	200	57	55	20	55	20	55	16	2C	FF	FF	3F
9	920 MHz (Japan)	016	50	46	66	20	55	20	55	12	26	FF	FF	3F
		000	100	57	55	20	55	20	55	15	1C	32	56	03
		017	200	57	55	20	55	20	55	13	18	FF	FF	3F
		018	400	46	66	30	50	30	50	11	25	FF	FF	3F
Other		019	300	46	55	20	54	20	54	15	21	FF	FF	3F
		020	100	46	55	20	54	20	54	17	33	FF	FF	3F
		021	300	46	55	20	54	20	54	15	21	FF	FF	3F

Table 18 - 57 Settings Required for Each Data Rate (6/9)

Frequency band identifier	PHY	Mode	Data rate (kbps)	Address (H)										
				05A1	05A2	0405	04F6	04F8	0505	0403	0415	04D9	0100	0101
4	863 MHz (Europe)	001	50	FF	3F	10	10	00	48	04	00	36	F6	72
		002	100	56	03	10	10	00	48	04	00	36	F6	72
		003	200	FF	3F	04	10	30	48	0C	F1	2F	BE	FF
5	896 MHz (US)	004	10	FF	3F	10	04	38	48	04	F1	36	F6	72
		005	20	FF	3F	10	04	38	48	04	F1	36	F6	72
		006	40	FF	3F	10	04	38	48	04	F1	36	F6	72
6	901 MHz (US)	007	10	FF	3F	10	04	38	48	04	F1	36	F6	72
		008	20	FF	3F	10	04	38	48	04	F1	36	F6	72
		009	40	FF	3F	10	04	38	48	04	F1	36	F6	72
7	915 MHz (US)	010	50	FF	3F	10	10	00	48	04	00	36	F6	72
		011	150	FF	3F	10	04	38	48	04	F1	36	F6	72
		012	200	FF	3F	10	04	38	48	04	F1	36	F6	72
8	917 MHz (Korea)	013	50	FF	3F	10	10	00	48	04	00	36	F6	72
		014	150	FF	3F	10	04	38	48	04	F1	36	F6	72
		015	200	FF	3F	10	04	38	48	04	F1	36	F6	72
9	920 MHz (Japan)	016	50	FF	3F	10	10	00	48	04	00	36	F6	72
		000	100	56	03	10	10	00	48	04	00	36	F6	72
		017	200	FF	3F	10	10	00	48	04	00	36	F6	72
		018	400	FF	3F	04	10	30	48	0C	F1	2F	BE	FF
Other		019	300	FF	3F	10	04	38	48	04	F1	36	F6	72
		020	100	FF	3F	10	04	38	45	04	F1	36	F6	72
		021	300	FF	3F	10	04	38	48	04	F1	36	F6	72

Table 18 - 58 Settings Required for Each Data Rate (7/9)

Frequency band identifier	PHY	Mode	Data rate (kbps)	Address (H)										
				0102	0103	0104	0105	0106	0107	0108	0109	010A	010B	00DC
4	863 MHz (Europe)	001	50	00	00	5E	70	00	00	C6	B4	00	00	11
		002	100	00	00	5E	70	00	00	C6	B4	00	00	10
		003	200	AE	BF	FE	BB	AA	BF	BE	FA	BA	EF	10
5	896 MHz (US)	004	10	00	00	5E	70	00	00	C6	B4	00	00	10
		005	20	00	00	5E	70	00	00	C6	B4	00	00	11
		006	40	00	00	5E	70	00	00	C6	B4	00	00	11
6	901 MHz (US)	007	10	00	00	5E	70	00	00	C6	B4	00	00	10
		008	20	00	00	5E	70	00	00	C6	B4	00	00	10
		009	40	00	00	5E	70	00	00	C6	B4	00	00	11
7	915 MHz (US)	010	50	00	00	5E	70	00	00	C6	B4	00	00	11
		011	150	00	00	5E	70	00	00	C6	B4	00	00	10
		012	200	00	00	5E	70	00	00	C6	B4	00	00	10
8	917 MHz (Korea)	013	50	00	00	5E	70	00	00	C6	B4	00	00	11
		014	150	00	00	5E	70	00	00	C6	B4	00	00	10
		015	200	00	00	5E	70	00	00	C6	B4	00	00	10
9	920 MHz (Japan)	016	50	00	00	5E	70	00	00	C6	B4	00	00	11
		000	100	00	00	5E	70	00	00	C6	B4	00	00	11
		017	200	00	00	5E	70	00	00	C6	B4	00	00	02
		018	400	AE	BF	FE	BB	AA	BF	BE	FA	BA	EF	02
Other		019	300	00	00	5E	70	00	00	C6	B4	00	00	02
		020	100	00	00	5E	70	00	00	C6	B4	00	00	11
		021	300	00	00	5E	70	00	00	C6	B4	00	00	02

Table 18 - 59 Settings Required for Each Data Rate (8/9)

Frequency band identifier	PHY	Mode	Data rate (kbps)	Address (H)										
				00DD	00DC	00DD	00F1	00F2	00F3	00F4	00F5	00FB	00DC	00DD
4	863 MHz (Europe)	001	50	0E	10	23	06	0A	5F	0D	DA	DA	12	63
		002	100	0E	10	23	09	0F	5F	11	DA	DA	12	63
		003	200	0E	10	23	0E	17	5F	0F	DA	DA	12	63
5	896 MHz (US)	004	10	0E	10	23	06	0A	5F	0D	CA	CA	0A	63
		005	20	0E	10	23	06	0A	5F	0D	CA	CA	0A	63
		006	40	0E	10	23	06	0A	5F	0D	CA	CA	0A	63
6	901 MHz (US)	007	10	0E	10	23	06	0A	5F	0D	CA	CA	0A	63
		008	20	0E	10	23	06	0A	5F	0D	CA	CA	0A	63
		009	40	0E	10	23	06	0A	5F	0D	CA	CA	0A	63
7	915 MHz (US)	010	50	0E	10	23	06	0A	5F	0D	BA	BA	0A	63
		011	150	0E	10	23	17	25	5F	0D	BA	BA	0A	63
		012	200	0E	10	23	26	3D	9F	06	BA	BA	0A	63
8	917 MHz (Korea)	013	50	0E	10	23	06	0A	5F	0D	BA	BA	0A	63
		014	150	0E	10	23	17	25	5F	0D	BA	BA	0A	63
		015	200	0E	10	23	26	3D	9F	06	BA	BA	0A	63
9	920 MHz (Japan)	016	50	0E	10	23	06	0A	5F	0D	BA	BA	0A	63
		000	100	0E	10	23	09	0F	5F	11	BA	BA	0A	63
		017	200	0E	40	23	0E	17	5F	0F	BA	BA	0A	63
		018	400	0E	40	23	0E	17	5F	0F	BA	BA	0A	63
Other		019	300	0E	40	23	37	58	5F	04	BA	BA	0A	63
		020	100	0E	10	23	0E	17	5F	0F	BA	BA	0A	63
		021	300	0E	40	23	37	58	5F	04	BA	BA	0A	63

Table 18 - 60 Settings Required for Each Data Rate (9/9)

Frequency band identifier	PHY	Mode	Data rate (kbps)	Address (H)					
				00DC	00DD	00DC	00DD	00DC	00DD
4	863 MHz (Europe)	001	50	Note 1	01H	Note 2	02H	Note 3	0BH
		002	100	Note 1	01H	Note 2	02H	Note 3	0BH
		003	200	Note 1	01H	Note 2	02H	Note 3	0BH
5	896 MHz (US)	004	10	Note 1	01H	Note 2	02H	Note 3	0BH
		005	20	Note 1	01H	Note 2	02H	Note 3	0BH
		006	40	Note 1	01H	Note 2	02H	Note 3	0BH
6	901 MHz (US)	007	10	Note 1	01H	Note 2	02H	Note 3	0BH
		008	20	Note 1	01H	Note 2	02H	Note 3	0BH
		009	40	Note 1	01H	Note 2	02H	Note 3	0BH
7	915 MHz (US)	010	50	Note 1	01H	Note 2	02H	Note 3	0BH
		011	150	Note 1	01H	Note 2	02H	Note 3	0BH
		012	200	Note 1	01H	Note 2	02H	Note 3	0BH
8	917 MHz (Korea)	013	50	Note 1	01H	Note 2	02H	Note 3	0BH
		014	150	Note 1	01H	Note 2	02H	Note 3	0BH
		015	200	Note 1	01H	Note 2	02H	Note 3	0BH
9	920 MHz (Japan)	016	50	Note 1	01H	Note 2	02H	Note 3	0BH
		000	100	Note 1	01H	Note 2	02H	Note 3	0BH
		017	200	Note 1	01H	Note 2	02H	Note 4	0BH
		018	400	Note 1	01H	Note 2	02H	Note 4	0BH
Other		019	300	Note 1	01H	Note 2	02H	Note 4	0BH
		020	100	Note 1	01H	Note 2	02H	Note 3	0BH
		021	300	Note 1	01H	Note 2	02H	Note 4	0BH

Note 1. Value saved to MCU address EFFECH.
Note 2. Value saved to MCU address EFFEDH.
Note 3. Value saved to MCU address EFFEEH.
Note 4. Value saved to MCU address EFFEFH.

18.8 Notice For Using Baseband Function

18.8.1 Notice About Transmission

Do not stop transmit operation.
Reset (initialize) by RFRESETB pin after setting RFSTOP bit to 1 for stopping transmission.
Initialize in same way for stopping in continuous transmission mode and special transmission mode.

18.8.2 Cautions on First and Second Address Filter Match Monitor Bits

(1) When the address filter extension bit is disabled, values of first and second address filter match monitor bits are invalid.

(2) When the address filter extension bit is enabled, values of first and second address filter match monitor bits are invalid under the following conditions.
- Frame version: 00, 01
- Frame type: Beacon frame
- No destination PANID and destination address
- Values of source PANID and PAN identifier register match, or value of PAN identifier register = FFFFH
- PANCORD bit = 0

CHAPTER 19 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.

19.1 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 **Tables 19 - 1** and **19 - 2**.

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.

19.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Tables 19 - 1** and **19 - 2**). 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 19 - 1 Interrupt Source List (1/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2
		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)
	5	INTP3		Internal	000EH	
	6	INTP4		External	0010H	
	8	INTCSI20	CSI20 transfer end or buffer empty interrupt	Internal	0014H	(A)
	9	INTCSI21	CSI21 transfer end or buffer empty interrupt		0016H	
	10	INTTM11H	End of timer channel 11 count or capture (at higher 8-bit timer operation)		0018H	
	13	INTTM01H	End of timer channel 01 count or capture (at higher 8-bit timer operation)		0022H	
	14	INTST1/ INTCSI10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt		0024H	
	15	INTSR1	UART1 reception transfer end		0026H	
	16	INTSRE1	UART1 reception communication error occurrence		0028H	
		INTTM03H	End of timer channel 03 count or capture (at higher 8-bit timer operation)			
	17	INTIICA0	End of IICA0 communication		002AH	
	18	INTTM00	End of timer channel 00 count or capture		002CH	
	19	INTTM01	End of timer channel 01 count or capture		002EH	
	20	INTTM02	End of timer channel 02 count or capture		0030H	
	21	INTTM03	End of timer channel 03 count or capture		0032H	
	22	INTAD	End of A/D conversion		0034H	
	23	INTRTC	Fixed-cycle signal of real-time clock/alarm match detection		0036H	
	24	INTIT	Interval signal detection		0038H	
	26	INTST3/ INTCSI30	UART3 transmission transfer end or buffer empty interrupt/CSI30 transfer end or buffer empty interrupt		003CH	
	27	INTSR3	UART3 reception transfer end		003EH	
	28	INTTRJ0	Timer RJ interrupt		0040H	

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 44 indicates the lowest priority.

Note 2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 19 - 1.

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.

Table 19 - 2 Interrupt Source List (2/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2
		Name	Trigger			
Maskable	29	INTTM10	End of timer channel 10 count or capture	Internal	0042H	(A)
	30	INTTM11	End of timer channel 11 count or capture		0044H	
	31	INTTM12	End of timer channel 12 count or capture		0046H	
	32	INTTM13	End of timer channel 13 count or capture		0048H	
	33	INTP6	Pin input edge detection	External	004AH	(B)
	34	INTP7			004CH	
	35	INTP8			004EH	
	36	INTP9			0050H	
	37	INTP10			0052H	
	38	INTP11			0054H	
	42	INTSRE3	UART3 reception communication error occurrence	Internal	005CH	(A)
		INTTM13H	End of timer channel 13 count or capture (at 8-bit timer operation)			
	43	INTIICA1	End of IICA1 communication		0060H	
	44	INTFL	Reserved Note 3		0062H	
Software	—	BRK	Execution of BRK instruction	—	007EH	(C)
Reset	—	RESET	$\overline{\text{RESET}}$ pin input	—	0000H	—
		POR	Power-on-reset			
		LVD	Voltage detection Note 4			
		WDT	Overflow of watchdog timer			
		TRAP	Execution of illegal instruction Note 5			
		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 44 indicates the lowest priority.

Note 2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 19 - 1.

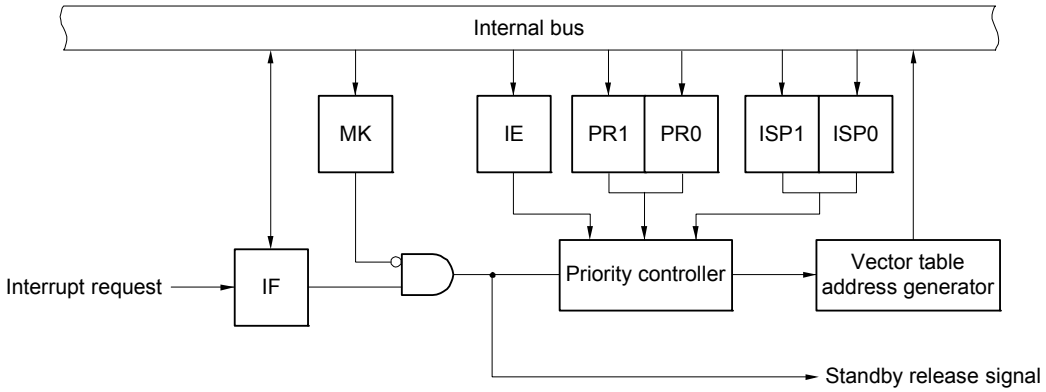
Note 3. Be used at the flash self-programming library or the data flash library.

Note 4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

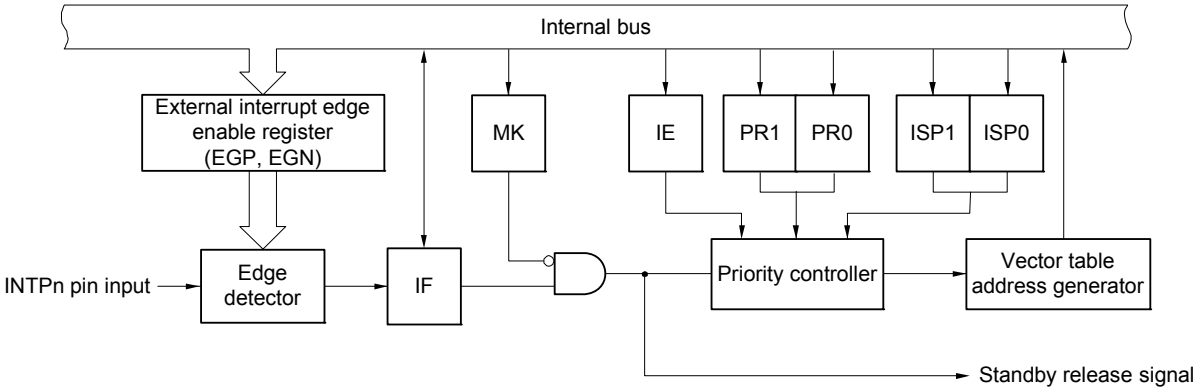
Note 5. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 19 - 1 Basic Configuration of Interrupt Function

(A) Internal maskable interrupt

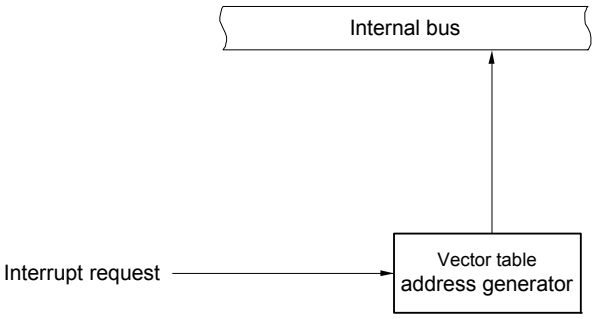


(B) External maskable interrupt (INTPn)



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

(C) Software interrupt



19.3 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)
- IInterrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, 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)

Tables 19 - 3 to 19 - 5 show a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 19 - 3 Flags Corresponding to Interrupt Request Sources (1/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1	
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP3	PIF3		PMK3		PPR03, PPR13	
INTP4	PIF4		PMK4		PPR04, PPR14	
INTCSI20	CSIIF20	IF0H	CSIMK20	MK0H	CSIPR020, CSIPR120	PR00H, PR10H
INTCSI21	CSIIF21		CSIMK21		CSIPR021, CSIPR121	
INTTM11H	TMIF11H		TMMK11H		TMPR011H, TMPR111H	
INTTM01H	TMIF01H		TMMK01H		TMPR001H, TMPR101H	

Table 19 - 4 Flags Corresponding to Interrupt Request Sources (2/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTST1 Note 1	STIF1 Note 1	IF1L	STMK1 Note 1	MK1L	STPR01, STPR11 Note 1	PR01L, PR11L
INTCSI10 Note 1	CSIIF10 Note 1		CSIMK10 Note 1		CSIPR010, CSIPR110 Note 1	
INTSR1 Note 2	SRIF1 Note 2		SRMK1 Note 2		SRPR01, SRPR11 Note 2	
INTSRE1 Note 3	SREIF1 Note 3		SREMK1 Note 3		SREPR01, SREPR11 Note 3	
INTTM03H Note 3	TMIF03H Note 3		TMMK03H Note 3		TMPR003H, TMPR103H Note 3	
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1	
INTIT	ITIF		ITMK		ITPR0, ITPR1	
INTST3 Note 4	STIF3 Note 4		STMK3 Note 4		STPR03, STPR13 Note 4	
INTCSI30 Note 4	CSIIF30 Note 4		CSIMK30 Note 4		CSIPR030, CSIPR130 Note 4	
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13	
INTTRJ0	TRJIF0		TRJMK0		TRJPR00, TRJPR10	
INTTM10	TMIF10		TMMK10		TMPR010, TMPR110	

Note 1. If one of the interrupt sources INTST1 and INTCSI10 is generated, bit 0 of the IF1L register is set to 1. Bit 0 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Note 2. If one of the interrupt sources INTSR1 and INTCSI11 is generated, bit 1 of the IF1L register is set to 1. Bit 1 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Note 3. Do not use a UART1 reception error interrupt and an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. When the UART1 reception error interrupt is not used (EOC03 = 0), UART1 and channel 3 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE1 or INTTM03H is generated, bit 2 of the IF1H register is set to 1. Bit 2 of the MK1L, PR01L, and PR11L registers support these two interrupt sources.

Note 4. If one of the interrupt sources INTST3, and INTCSI30 is generated, bit 4 of the IF1H register is set to 1. Bit 4 of the MK1H, PR01H, and PR11H registers supports these two interrupt sources.

Table 19 - 5 Flags Corresponding to Interrupt Request Sources (3/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTTM11	TMIF11	IF2L	TMMK11	MK2L	TMPR011, TMPR111	PR02L, PR12L
INTTM12	TMIF12		TMMK12		TMPR012, TMPR112	
INTTM13	TMIF13		TMMK13		TMPR013, TMPR113	
INTP6	PIF6		PMK6		PPR06, PPR16	
INTP7	PIF7		PMK7		PPR07, PPR17	
INTP9	PIF9		PMK9		PPR09, PPR19	
INTP10	PIF10		PMK10		PPR010, PPR110	
INTP11	PIF11	IF2H	PMK11	MK2H	PPR011, PPR111	PR02H, PR12H
INTSRE3 **Note**	SREIF3 **Note**		SREMK3 **Note**		SREPR03, SREPR13 **Note**	
INTTM13H **Note**	TMIF13H **Note**		TMMK13H **Note**		TMPR013H, TMPR113H **Note**	
INTIICA1	IICAIF1		IICAMK1		IICAPR01, IICAPR11	
INTFL	FLIF		FLMK		FLPR0, FLPR1	

Note Do not use a UART3 reception error interrupt and an interrupt of channel 3 of TAU1 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. When the UART3 reception error interrupt is not used (EOC03 = 0), UART3 and channel 3 of TAU1 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE3 or INTTM13H is generated, bit 4 of the IF2H register is set to 1. Bit 4 of the MK2H, PR02H, and PR12H registers support these two interrupt sources.

19.3.1 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, 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 19 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
IF0L	0	PIF4	PIF3	0	0	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
IF0H	TMIF01H	0	0	0	0	TMIF11H	CSIIF21	CSIIF20

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICAIF0	SREIF1 TMIF03H	SRIF1	STIF1 CSIIF10

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
IF1H	TMIF10	TRJIF0	SRIF3	STIF3 CSIIF30	0	ITIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
IF2L	PIF10	PIF9	0	PIF7	PIF6	TMIF13	TMIF12	TMIF11

Figure 19 - 3 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	3	2	1	<0>
IF2H	FLIF	IICAIF1	0	SREIF3 TMIF13H	0	0	0	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 Tables 19 - 3 to 19 - 5. 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.

19.3.2 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, 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 19 - 4 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (1/2)

Address: FFFE4H After reset: FFH R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
MK0L	1	PMK4	PMK3	1	1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
MK0H	TMMK01H	1	1	1	1	TMMK11H	CSIMK21	CSIMK20

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICAMK0	SREMK1 TMMK03H	SRMK1	STMK1 CSIMK10

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
MK1H	TMMK10	TRJMK0	SRMK3	STMK3 CSIMK30	1	ITMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
MK2L	PMK10	PMK9	1	PMK7	PMK6	TMMK13	TMMK12	TMMK11

Figure 19 - 5 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (2/2)

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	3	2	1	<0>
MK2H	FLMK	IICAMK1	1	SREMK3 TMMK13H	1	1	1	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 Tables 19 - 3 to 19 - 5. Be sure to set bits that are not available to the initial value.

19.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, 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 the 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 PR02L and PR02H 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, PR02, 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 19 - 6 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, 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	1	PPR04	PPR03	1	1	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	7	<6>	<5>	4	3	<2>	<1>	<0>
PR10L	1	PPR14	PPR13	1	1	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset:FFH R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
PR00H	TMPR001H	1	1	1	1	TMPR011H	CSIPR021	CSIPR020

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
PR10H	TMPR101H	1	1	1	1	TMPR111H	CSIPR121	CSIPR120

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICAPR00	SREPR01 TMPR003H	SRPR01	STPR01 CSIPR010

Figure 19 - 7 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR10	SREPR11 TMPR103H	SRPR11	STPR11 CSIPR110

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR01H	TMPR010	TRJPR00	SRPR03	STPR03 CSIPR030	1	ITPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR11H	TMPR110	TRJPR10	SRPR13	STPR13 CSIPR130	1	ITPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
PR02L	PPR010	PPR09	1	PPR07	PPR06	TMPR013	TMPR012	TMPR011

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
PR12L	PPR110	PPR19	1	PPR17	PPR16	TMPR113	TMPR112	TMPR111

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	3	2	1	<0>
PR02H	FLPR0	IICAPR01	1	SREPR03 TMPR013H	1	1	1	PPR011

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	3	2	1	<0>
PR12H	FLPR1	IICAPR11	1	SREPR13 TMPR113H	1	1	1	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 Tables 19 - 3 to 19 - 5. Be sure to set bits that are not available to the initial value.

19.3.4 External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

These registers specify the valid edge for INTPn pin.

The EGP0, EGP1, EGN0, and EGN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 19 - 8 Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	0	EGP4	EGP3	0	0	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	0	EGN4	EGN3	0	0	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	0

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 11)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 19 - 6 shows the Ports Corresponding to EGPn and EGNn bits.

Table 19 - 6 Ports Corresponding to EGPn and EGNn bits

Detection Enable Bit		Interrupt Request Signal
EGP0	EGN0	INTP0
EGP3	EGN3	INTP3
EGP4	EGN4	INTP4
EGP6	EGN6	INTP6
EGP7	EGN7	INTP7
EGP9	EGN9	INTP9
EGP10	EGN10	INTP10
EGP11	EGN11	INTP11

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 3.1 Port Functions.
Remark 2. n = 0 to 11

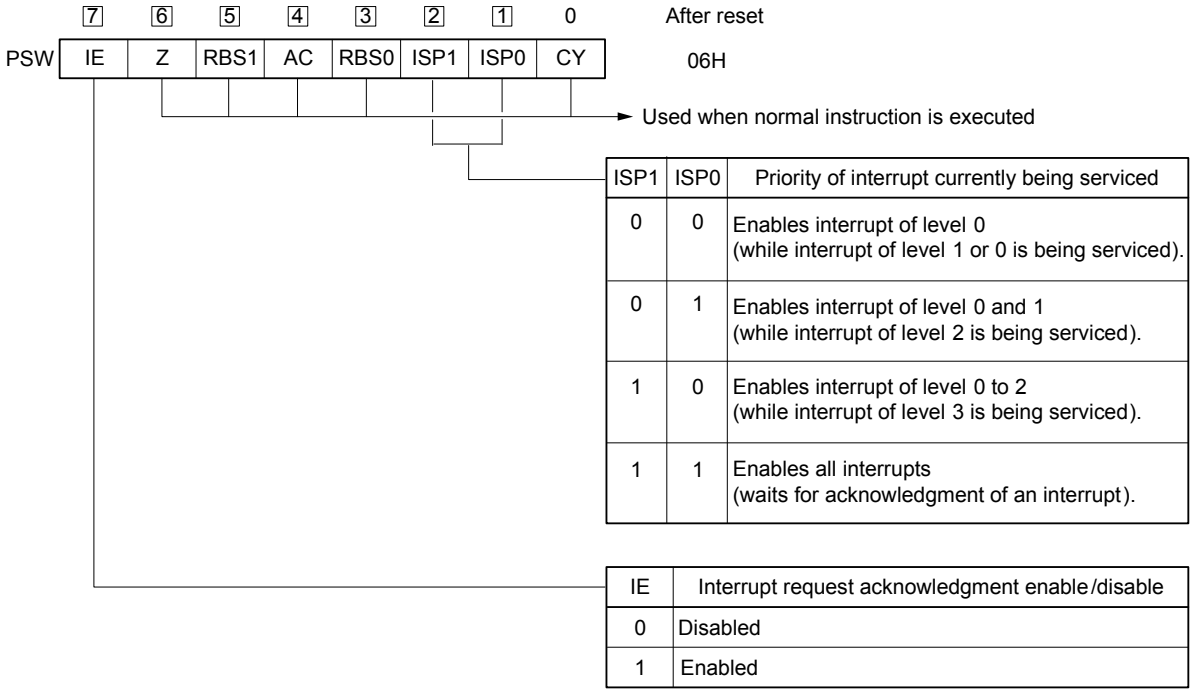
19.3.5 Program status word (PSW)

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 19 - 9 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

19.4 Interrupt Servicing Operations

19.4.1 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 19 - 7 below.

For the interrupt request acknowledgment timing, see **Figures 19 - 11** and **19 - 12**.

Table 19 - 7 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.

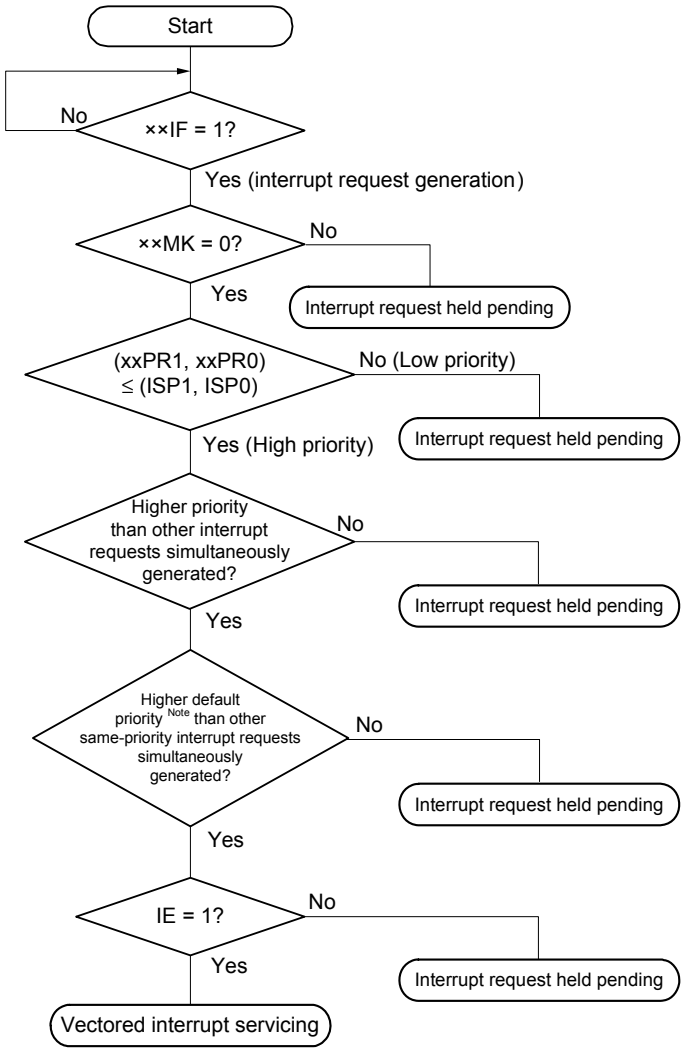
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 19 - 10 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.

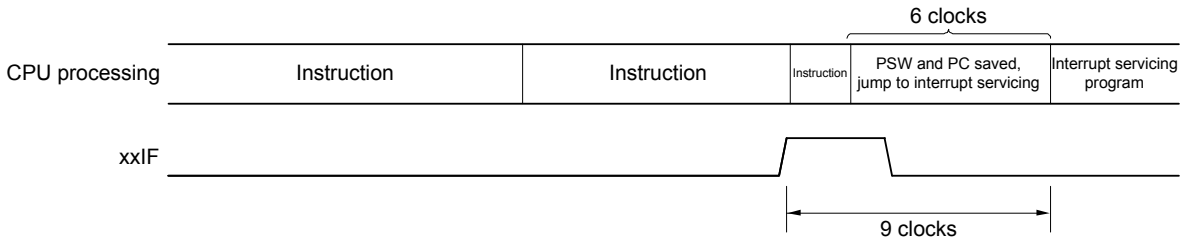
Figure 19 - 10 Interrupt Request Acknowledgment Processing Algorithm



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 19 - 9**)

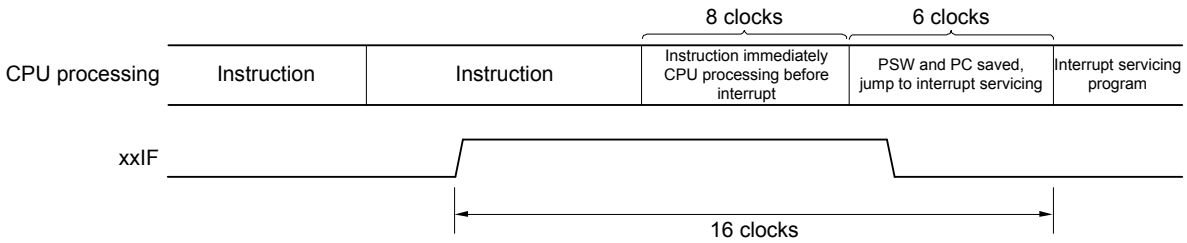
Note For the default priority, refer to **Tables 19 - 1** and **19 - 2 Interrupt Source List**.

Figure 19 - 11 Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

Figure 19 - 12 Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

19.4.2 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.

19.4.3 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 19 - 8 shows Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing and Figures 19 - 13 and 19 - 14 show multiple interrupt servicing examples.

Table 19 - 8 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

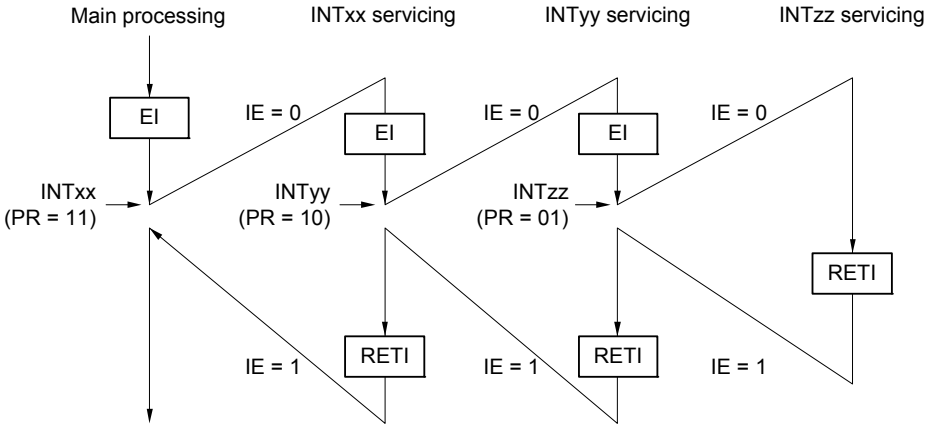
Remark 2. ×: Multiple interrupt servicing disabled

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)

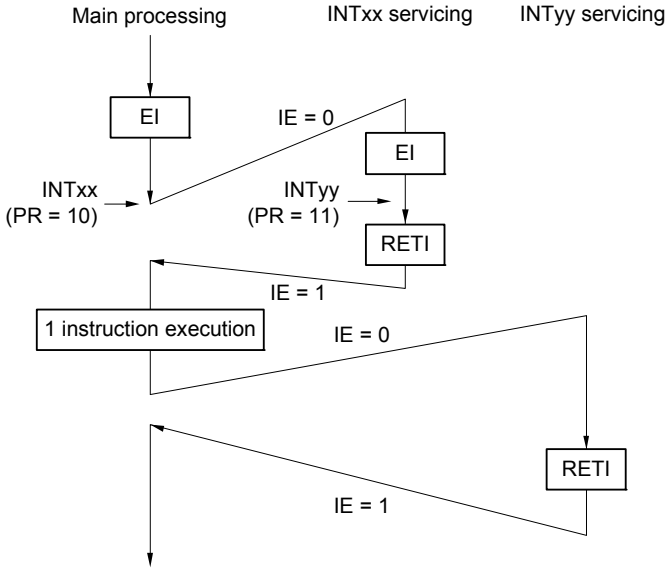
Figure 19 - 13 Examples of Multiple Interrupt Servicing (1/2)

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

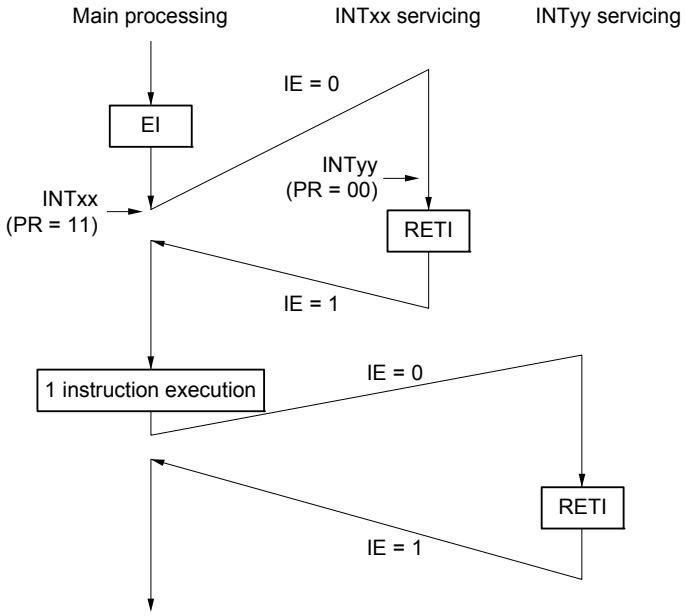


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

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 19 - 14 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.

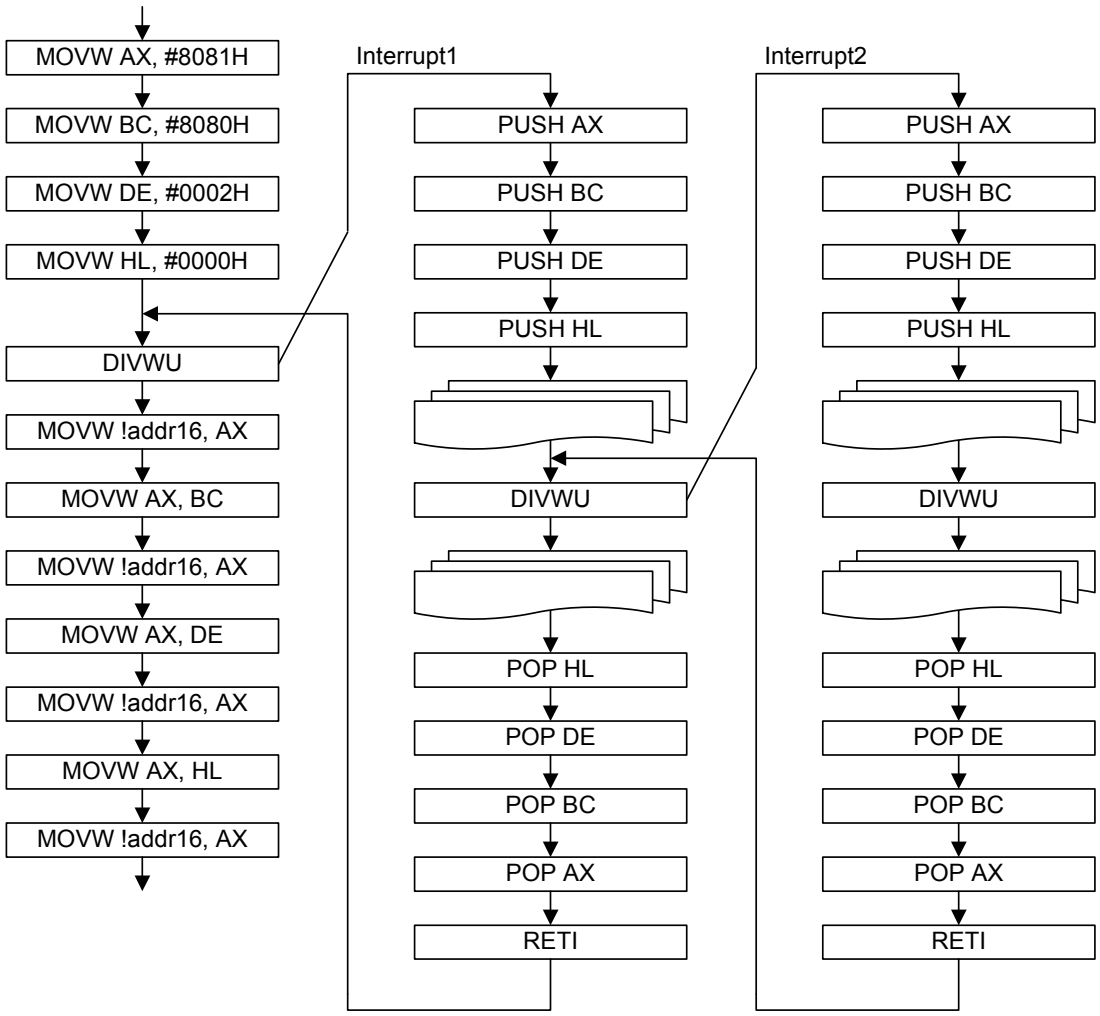
19.4.4 Interrupt servicing during division instruction

The RL78/G1H 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 When executing DIVHU and DIVWU instructions during the interrupt processing, execute them in the interrupt disabled (DI) state.

Excepting the instruction execution in RAM area, when NOP instruction is added right after DIVHU and DIVWU instructions in the assembly language source, DIVHU and DIVWU instructions can be executed even in the interrupt enabled state.

For the following compilers, when DIVHU and DIVWU instructions are output at building, NOP instruction is automatically inserted right after the output.

- **CA78K0R (compiler product manufactured by Renesas Electronics Corporation) V1.71 or later: C language source and assembly language source**
- **EWRL78 (compiler product manufactured by IAR Systems): C language source of Service pack 1.40.6 or later**
- **GNURL78 (compiler manufactured by KPIT Technologies Ltd.): C language source of 1.40.6 or later**

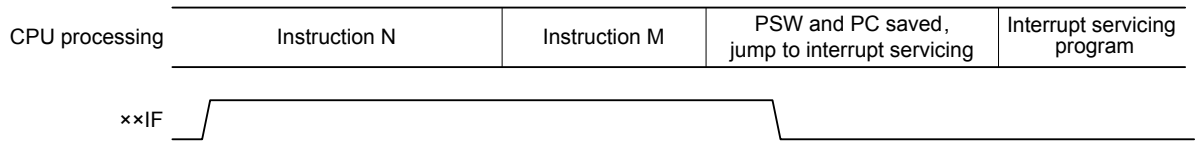
19.4.5 Interrupt request hold

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 19 - 15 shows the timing at which interrupt requests are held pending.

Figure 19 - 15 Interrupt Request Hold



Remark 1. Instruction N: Interrupt request hold instruction
Remark 2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 20 STANDBY FUNCTION

20.1 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, high-speed on-chip oscillator, or subsystem clock 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 an A/D conversion request by the timer trigger signal (the interrupt request signal (INTRTC/INTIT) or ELC event input), and DTC start source, the STOP mode is exited, A/D conversion is performed, and DTC start source without operating the CPU. 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. The STOP mode can be used only when the CPU is operating on the main system clock. Do not set to the STOP mode while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

Caution 2. 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 3. 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 13.3 Registers Controlling A/D Converter.

Caution 4. 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 5. 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 26 OPTION BYTE.

20.2 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 6 CLOCK GENERATOR**. For registers which control the SNOOZE mode, **CHAPTER 13 A/D CONVERTER**.

20.3 Standby Function Operation

20.3.1 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, high-speed on-chip oscillator clock, or subsystem 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 20 - 1 Operating Statuses in HALT Mode (1/2)

HALT Mode Setting / Item			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_{IH})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled	
		f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
		f_{EXS}			
	Low-speed on-chip oscillator clock	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			Operation stopped (Operable while in the DTC is executed)		
Port (latch)			Status before HALT mode was set is retained		
Timer array unit			Operable		
Real-time clock (RTC)					
12-bit Interval timer					
Watchdog timer			See **CHAPTER 12 WATCHDOG TIMER**.		
Timer RJ			Operable		
Clock output/buzzer output					
A/D converter					
Serial array unit (SAU)					
Serial interface (IICA)					
Data transfer controller (DTC)					
Event link controller (ELC)			Operable function blocks can be linked		
Power-on-reset function			Operable		
Voltage detection function					
External interrupt					
CRC operation function		High-speed CRC			
		General-purpose CRC	In the calculation of the RAM area, operable when DTC is executed only		
Illegal-memory access detection function			Operable when DTC is executed only		
RAM parity error detection function					
RAM guard function					
SFR guard function					

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock
f_{XT}: XT1 clock f_{EXS}: External subsystem clock

Table 20 - 2 Operating Statuses in HALT Mode (2/2)

HALT Mode Setting / Item			When HALT Instruction is Executed While CPU is Operating on Subsystem Clock	
			When CPU is Operating on XT1 Clock (f_{XT})	When CPU is Operating on External Subsystem Clock (f_{EXS})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f_{IH}	Operation disabled	
		f_X		
		f_{EX}		
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)	Cannot operate
		f_{EXS}	Cannot operate	Operation continues (cannot be stopped)
	Low-speed on-chip oscillator clock	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			Operation stopped (Operable while in the DTC is executed)	
Port (latch)			Status before HALT mode was set is retained	
Timer array unit			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
Real-time clock (RTC)			Operable	
12-bit Interval timer				
Watchdog timer			See **CHAPTER 12 WATCHDOG TIMER**.	
Timer RJ			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
Clock output/buzzer output				
A/D converter			Operation disabled	
Serial array unit (SAU)			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
Serial interface (IICA)			Operation disabled	
Data transfer controller (DTC)			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
Event link controller (ELC)			Operable function blocks can be linked	
Power-on-reset function			Operable	
Voltage detection function				
External interrupt				
CRC operation function	High-speed CRC		Operation disabled	
	General-purpose CRC		In the calculation of the RAM area, operable when DTC is executed only	
Illegal-memory access detection function			Operable when DTC is executed only	
RAM parity error detection function				
RAM guard function				
SFR guard function				

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock
f_{XT}: XT1 clock f_{EXS}: External subsystem clock

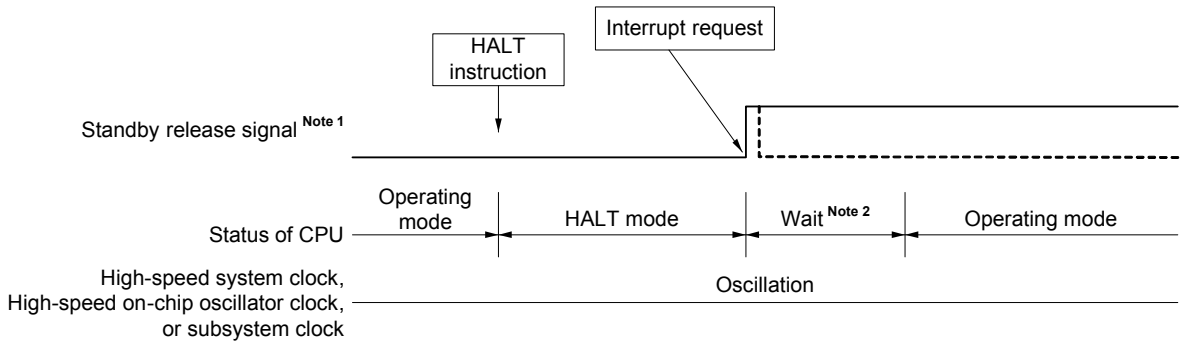
(2) HALT mode release

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 20 - 1 HALT Mode Release by Interrupt Request Generation



Note 1. For details of the standby release signal, see **Figure 19 - 1 Basic Configuration of Interrupt Function**.

Note 2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out
  - Main system clock: 15 to 16 clocks
  - Subsystem clock (RTCLPC = 0): 10 to 11 clocks
  - Subsystem clock (RTCLPC = 1): 11 to 12 clocks
- When vectored interrupt servicing is not carried out
  - Main system clock: 9 to 10 clocks
  - Subsystem clock (RTCLPC = 0): 4 to 5 clocks
  - Subsystem clock (RTCLPC = 1): 5 to 6 clocks

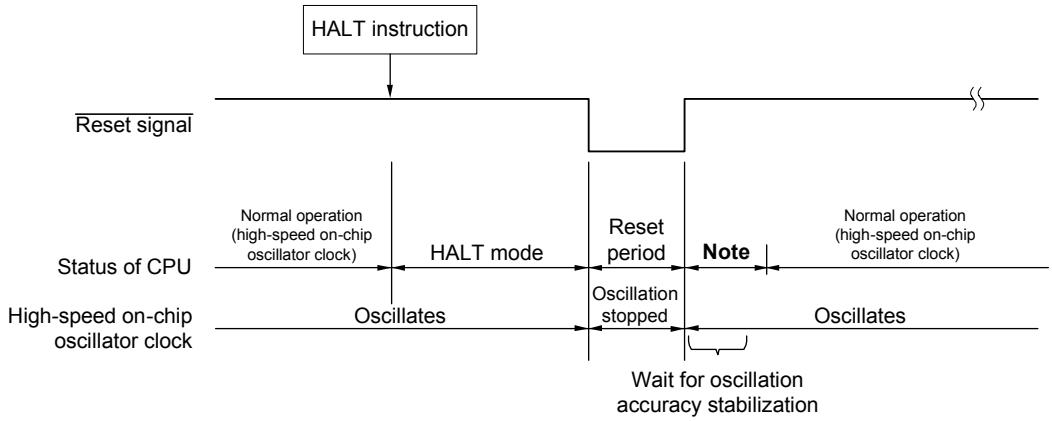
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

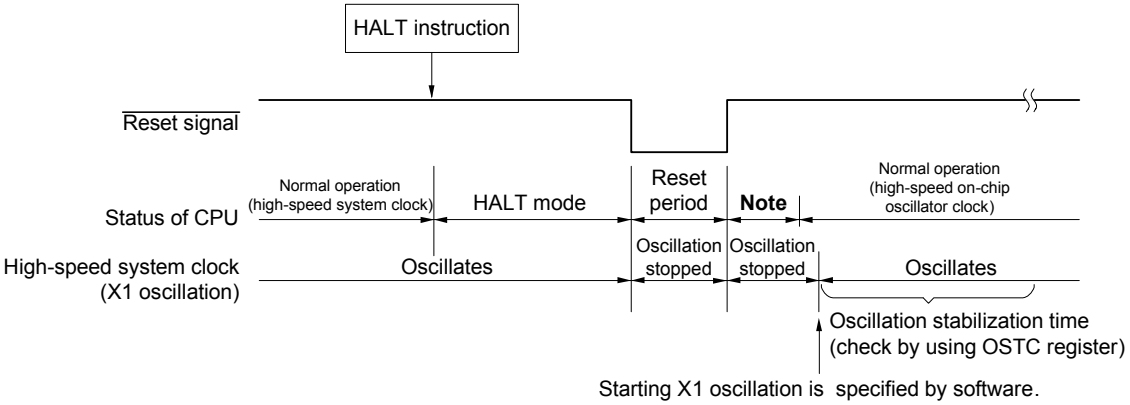
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 20 - 2 HALT Mode Release by Reset (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



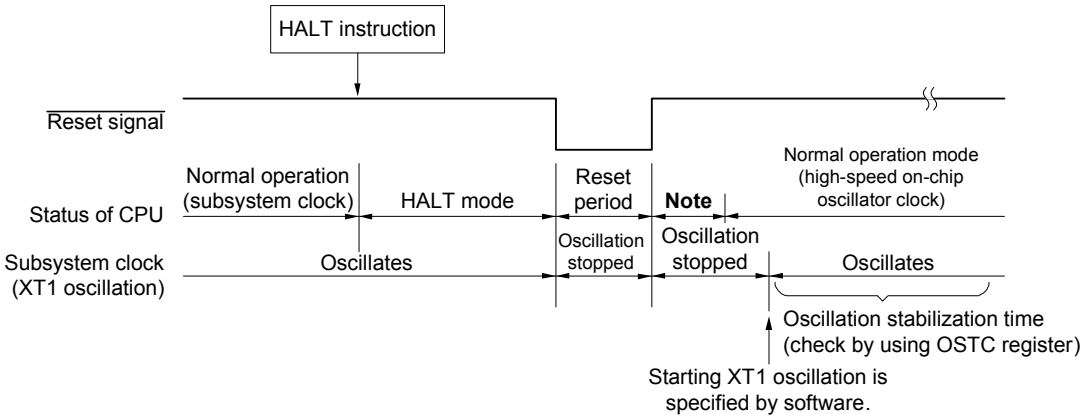
(2) When high-speed system clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 21 RESET FUNCTION**.

Figure 20 - 3 HALT Mode Release by Reset (2/2)

(3) When subsystem clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 21 RESET FUNCTION**.

20.3.2 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 main system clock.

Caution 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.

The operating statuses in the STOP mode are shown below.

Table 20 - 3 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_{IH})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
System clock		Clock supply to the CPU is stopped		
Main system clock	f_{IH}	Stopped		
	f_X			
	f_{EX}			
Subsystem clock	f_{XT}	Status before STOP mode was set is retained		
	f_{EXS}			
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		
Real-time clock (RTC)		Operable		
12-bit Interval timer				
Watchdog timer		See **CHAPTER 12 WATCHDOG TIMER**.		
Timer RJ		• Operable when the subsystem clock is selected as the count source and RTCLPC in the OSMC register = 0 • Operable when the low-speed on-chip oscillator is selected as the count source • Operation is disabled under any conditions other than the above		
Clock output/buzzer output		Operates when the subsystem clock is selected as the clock source for counting and the RTCLPC bit is 0 (operation is disabled when a clock other than the subsystem clock is selected and the RTCLPC bit is not 0).		
A/D converter		Wakeup operation is enabled (switching to SNOOZE mode)		
Serial array unit (SAU)		Operation disabled		
Serial interface (IICA)		Wakeup by address match operable		
Data transfer controller (DTC)		DTC activation source receiving operation enabled (switching to SNOOZE mode)		
Event link controller (ELC)		Operable function blocks can be linked		
Power-on-reset function		Operable		
Voltage detection function				
External interrupt				
CRC operation function	High-speed CRC	Operation stopped		
	General-purpose CRC			
Illegal-memory access detection function				
RAM parity error detection function				
RAM guard function				
SFR guard function				

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock
f_{XT}: XT1 clock f_{EXS}: External subsystem clock

(2) STOP mode release

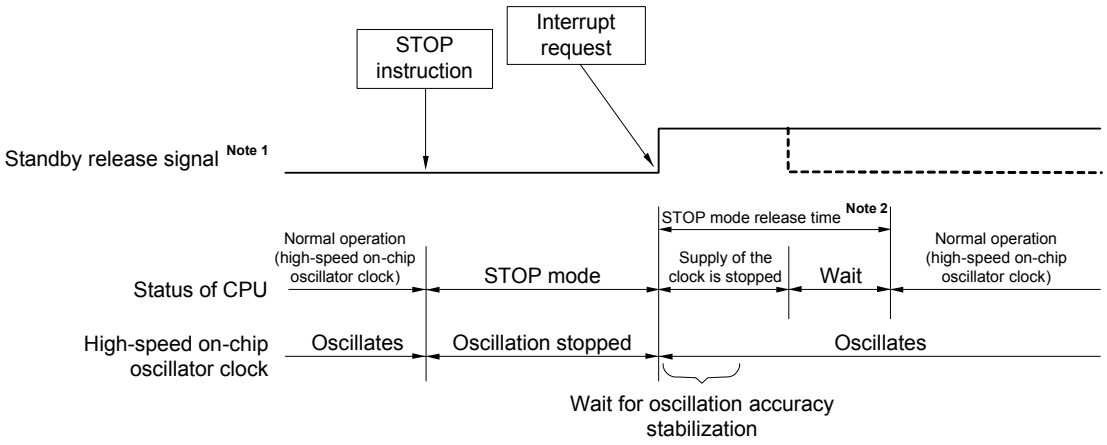
The STOP mode can be released by the following two sources.

(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 20 - 4 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 19 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

<R> Supply of the clock is stopped: 18 to 65 μs

Wait:

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

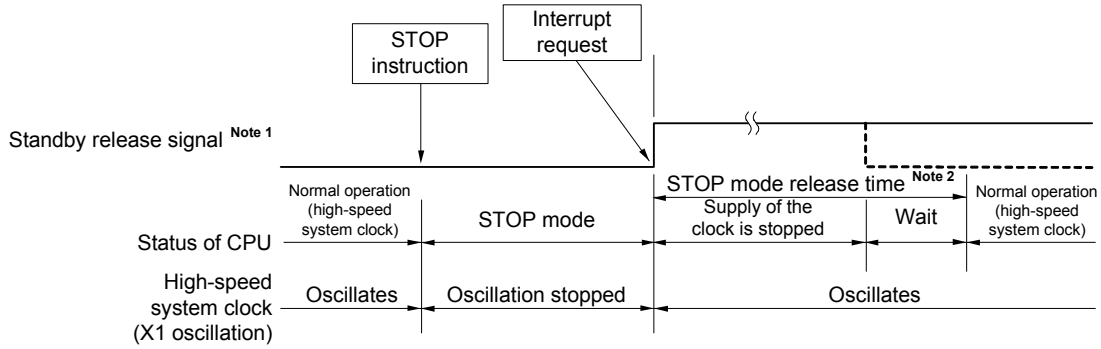
Caution To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction.

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 20 - 5 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 19 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

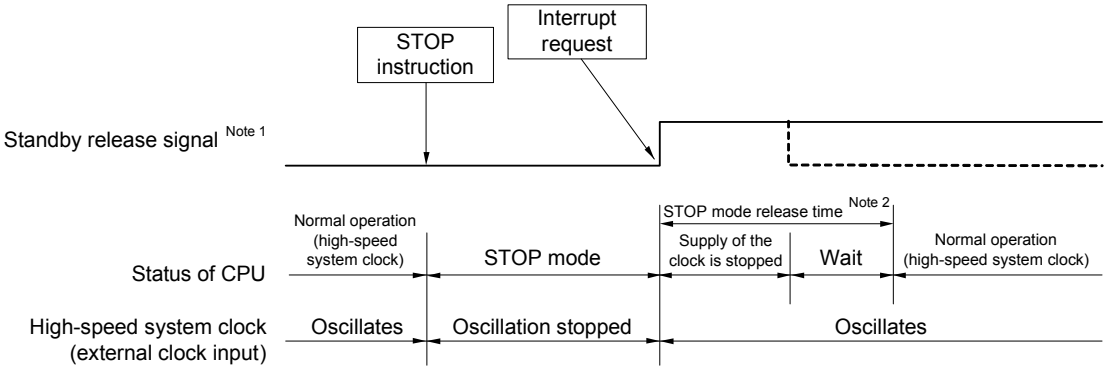
<R> Supply of the clock is stopped: 18 μs to "whichever is longer 65 μs or the oscillation stabilization time (set by OSTS)"

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 19 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

<R> Supply of the clock is stopped: 18 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.

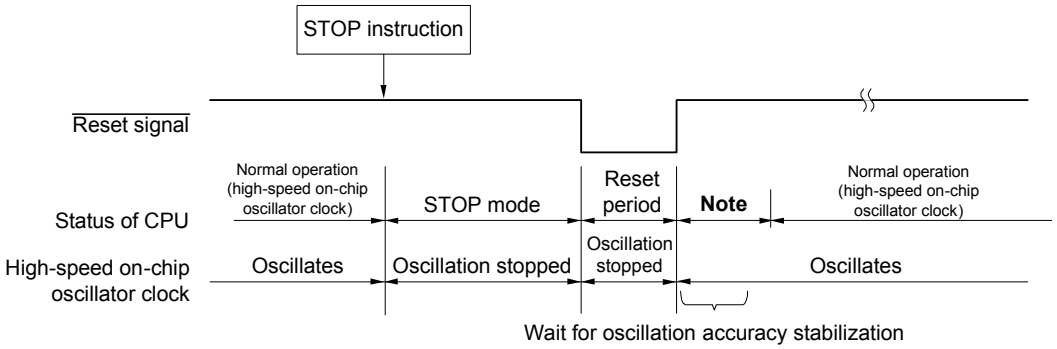
Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

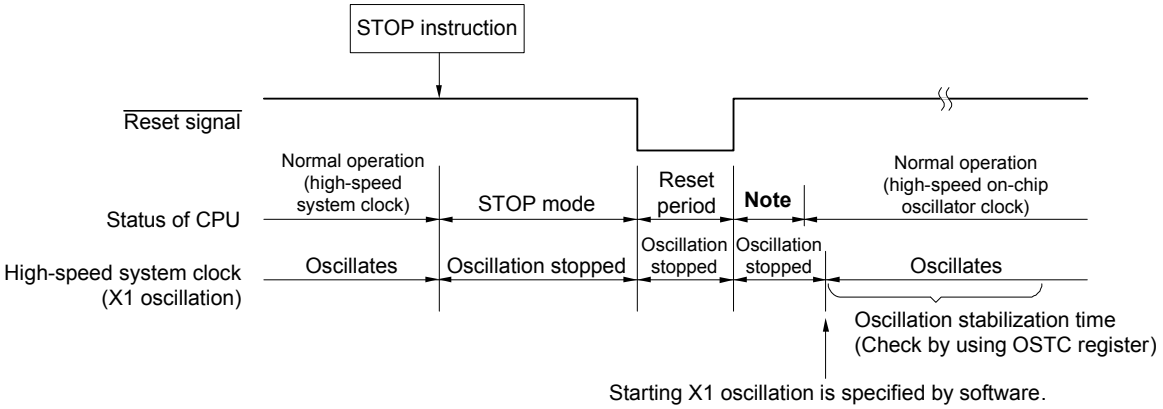
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 20 - 6 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 21 RESET FUNCTION**.

20.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for the A/D converter, or DTC. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.

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 **13.3 Registers Controlling A/D Converter**.

When DTC transfer is used in SNOOZE mode, before switching to the STOP mode, allow DTC activation by interrupt to be used. During STOP mode, detecting DTC activation by interrupt enables DTC transit to SNOOZE mode, automatically. For details, see **16.3 Registers Controlling DTC**.

In SNOOZE mode transition, wait status to be only following time.

<R> Transition time from STOP mode to SNOOZE mode: 18 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 vectored interrupt servicing is carried out:
  HS (High-speed main) mode: “4.99 μs to 9.44 μs” + 7 clocks
  LS (Low-speed main) mode: “1.10 μs to 5.08 μs” + 7 clocks

- When vectored interrupt servicing is not carried out:
  HS (High-speed main) mode: “4.99 μs to 9.44 μs” + 1 clock
  LS (Low-speed main) mode: “1.10 μs to 5.08 μs” + 1 clock

The operating statuses in the SNOOZE mode are shown next.

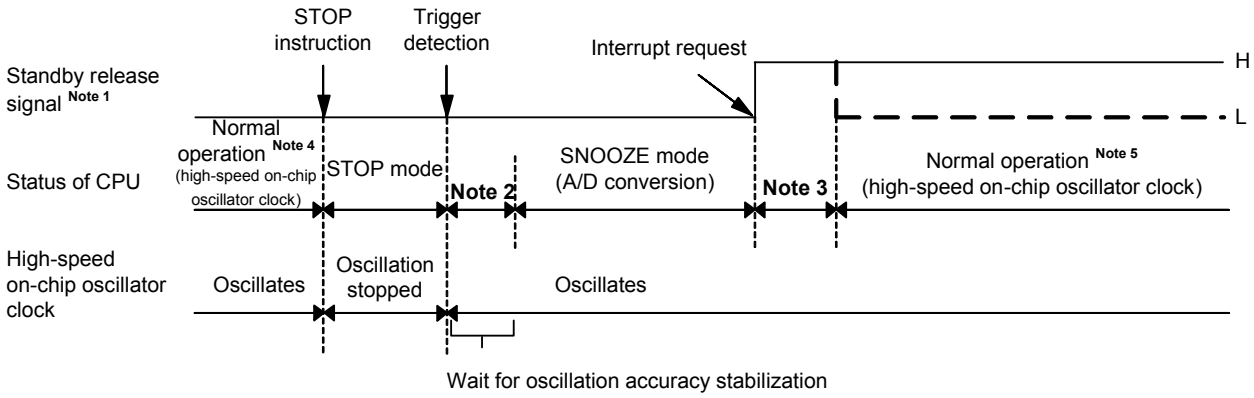
Table 20 - 4 Operating Statuses in SNOOZE Mode

Item \ STOP Mode Setting			During STOP mode, receiving data signal from CSIp and UARTq, inputting timer trigger signal to A/D converter, and generating DTC activation by interrupt
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{IH}	Operation started
		f_X	Stopped
		f_{EX}	
	Subsystem clock	f_{XT}	Use of the status while in the STOP mode continues
		f_{EXS}	
	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			Operation stopped (Operable while in the DTC is executed)
Port (latch)			Use of the status while in the STOP mode continues
Timer array unit			Operation disabled
Real-time clock (RTC)			Operable
12-bit interval timer			
Watchdog timer			See **CHAPTER 12 WATCHDOG TIMER**.
Timer RJ			• Operable when the subsystem clock is selected as the count source and RTCLPC in the OSMC register = 0 • Operable when the low-speed on-chip oscillator is selected as the count source • Operation is disabled under any conditions other than the above
Clock output/buzzer output			Operates when the subsystem clock is selected as the clock source for counting and the RTCLPC bit is 0 (operation is disabled when a clock other than the subsystem clock is selected and the RTCLPC bit is not 0).
A/D converter			Operable
Serial array unit (SAU)			Operation disabled
Serial interface (IICA)			Operation disabled
Data transfer controller (DTC)			Operable
Event link controller (ELC)			Operable function blocks can be linked
Power-on-reset function			Operable
Voltage detection function			
External interrupt			
<R> CRC operation function		High-speed CRC	Operation stopped
		General-purpose CRC	Operation disabled
Illegal-memory access detection function			Operable when executing the DTC
RAM parity error detection function			
RAM guard function			
SFR guard function			

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock
f_{XT}: XT1 clock f_{EXS}: External subsystem clock

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

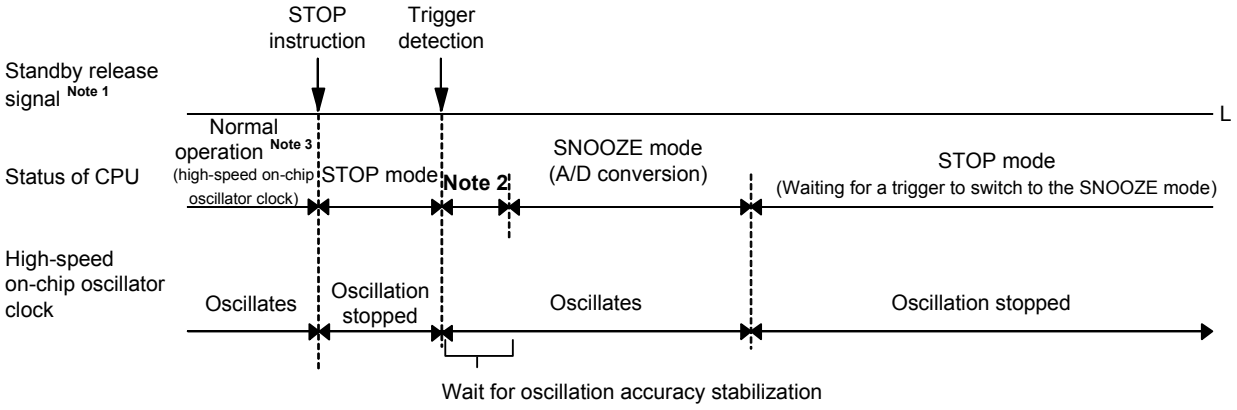
Figure 20 - 7 When the Interrupt Request Signal is Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 19 - 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 20 - 8 When the Interrupt Request Signal is not Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 19 - 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.

Remark For details of the SNOOZE mode function, see **CHAPTER 13 A/D CONVERTER**.

CHAPTER 21 RESET FUNCTION

The following seven operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ 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 21 - 1.

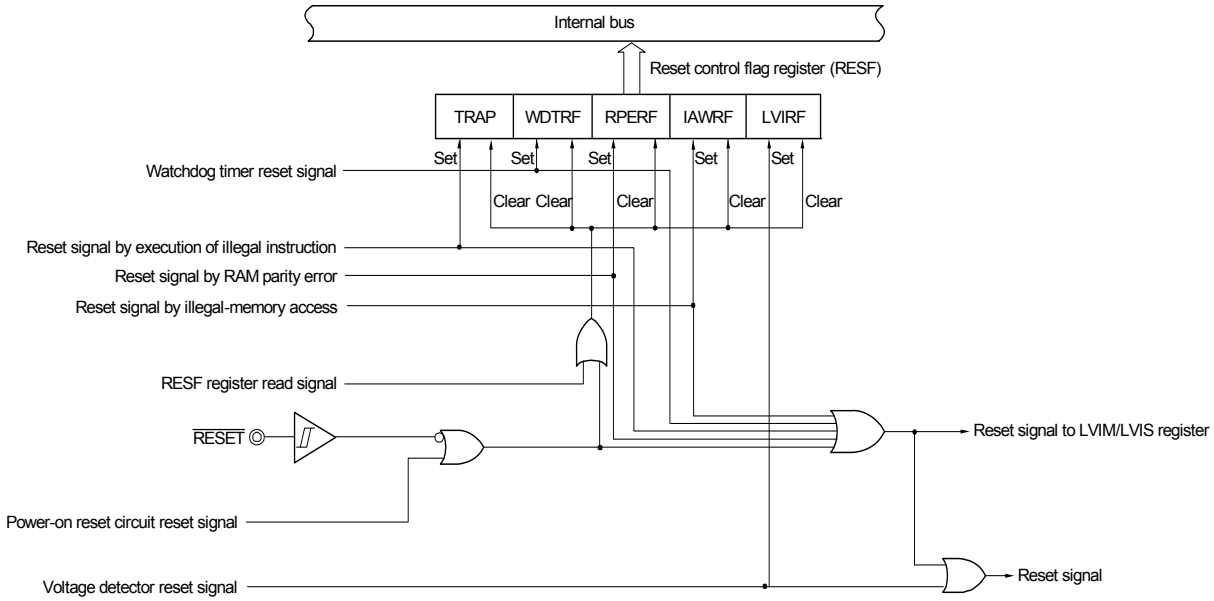
Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Caution 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ 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 31.4 AC Characteristics, and then input a high level to the pin.

Caution 2. During reset input, the X1 clock, XT1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock oscillating. External main system clock input and external subsystem 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).**
- **P130: Low level during the reset period or after receiving a reset signal.**
- **Ports other than P40 and P130: High-impedance during the reset period or after receiving a reset signal.**

Figure 21 - 1 Block Diagram of Reset Function



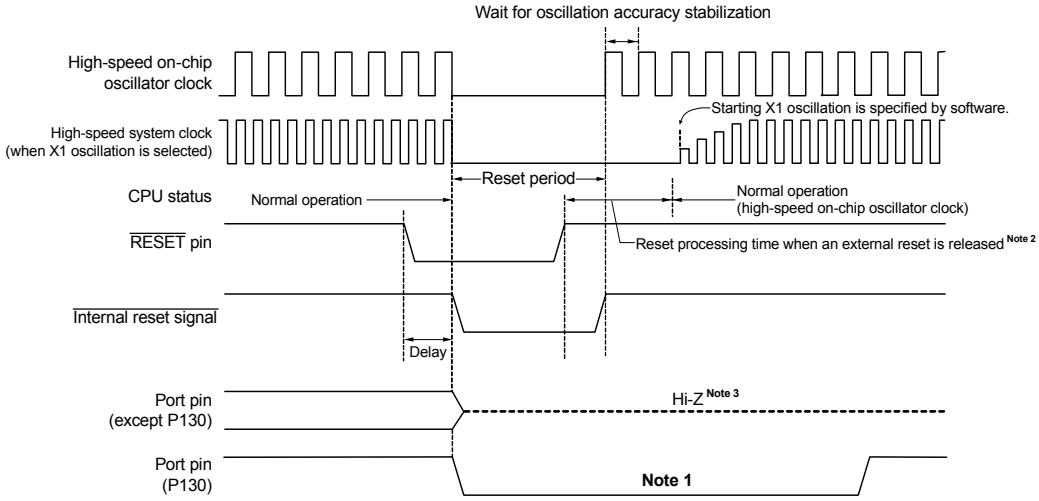
Caution An LVD circuit internal reset does not reset the LVD circuit.

Remark 1. LVIM: Voltage detection register
Remark 2. LVIS: Voltage detection level register

21.1 Timing of Reset Operation

This LSI is reset by input of the low level on the $\overline{\text{RESET}}$ pin and released from the reset state by input of the high level on the $\overline{\text{RESET}}$ pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

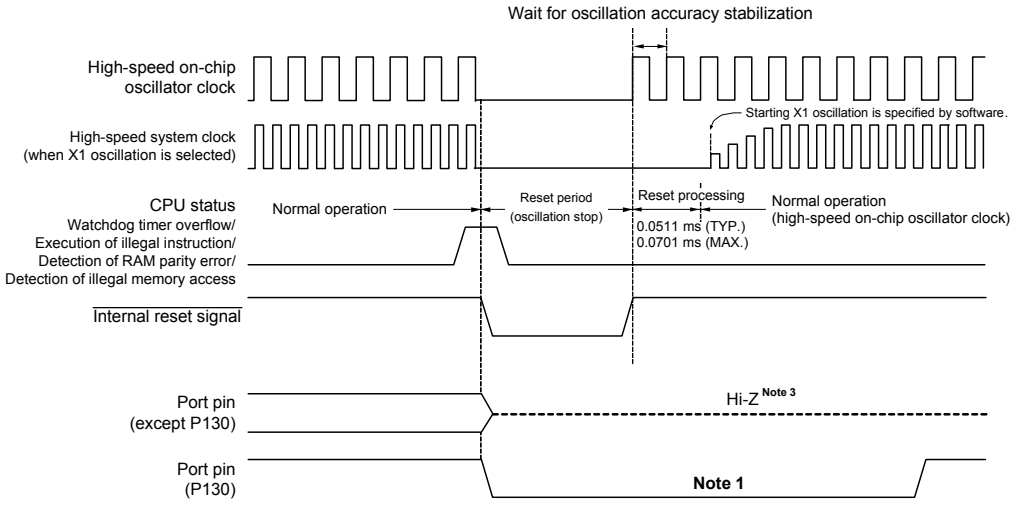
Figure 21 - 2 Timing of Reset by $\overline{\text{RESET}}$ Input



(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 21 - 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. When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

Note 2. 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 3. 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).

Caution The watchdog timer is also reset without exception when an internal reset occurs.

Reset by LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{LVD}$ 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 23 VOLTAGE DETECTOR**.

Remark V_{LVD}: LVD detection voltage

Table 21 - 1 Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	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)
	Subsystem clock	f_{XT}	Operation stopped (the XT1 and XT2 pins are input port mode)
		f_{EXS}	Clock input invalid (the pin is input port mode)
	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
Timer RJ			
Real-time clock (RTC)			
12-bit Interval timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Serial array unit (SAU)			
Serial interface (IICA)			
Data transfer controller (DTC)			
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	
Illegal-memory access detection function			
RAM parity error detection function			
RAM guard function			
SFR guard function			

Note P40 and P130 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 on-chip pull-up resistance).
- P130: Low level during the reset period

Remark f_{IH}: High-speed on-chip oscillator clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock
f_{EXS}: External subsystem clock
f_{IL}: Low-speed on-chip oscillator clock

Table 21 - 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 **4.1.4 Special function register (SFR) area** and **4.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

21.2 Register for Confirming Reset Source

21.2.1 Reset control flag register (RESF)

Many internal reset generation sources exist in the RL78 microcontroller. 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.

$\overline{\text{RESET}}$ input, reset by power-on-reset (POR) circuit, and reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags. To judge the reset source, store RESF register value in the desired RAM, and use the RAM value for judging.

Figure 21 - 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 21 - 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 24.3.3 RAM parity error detection function.

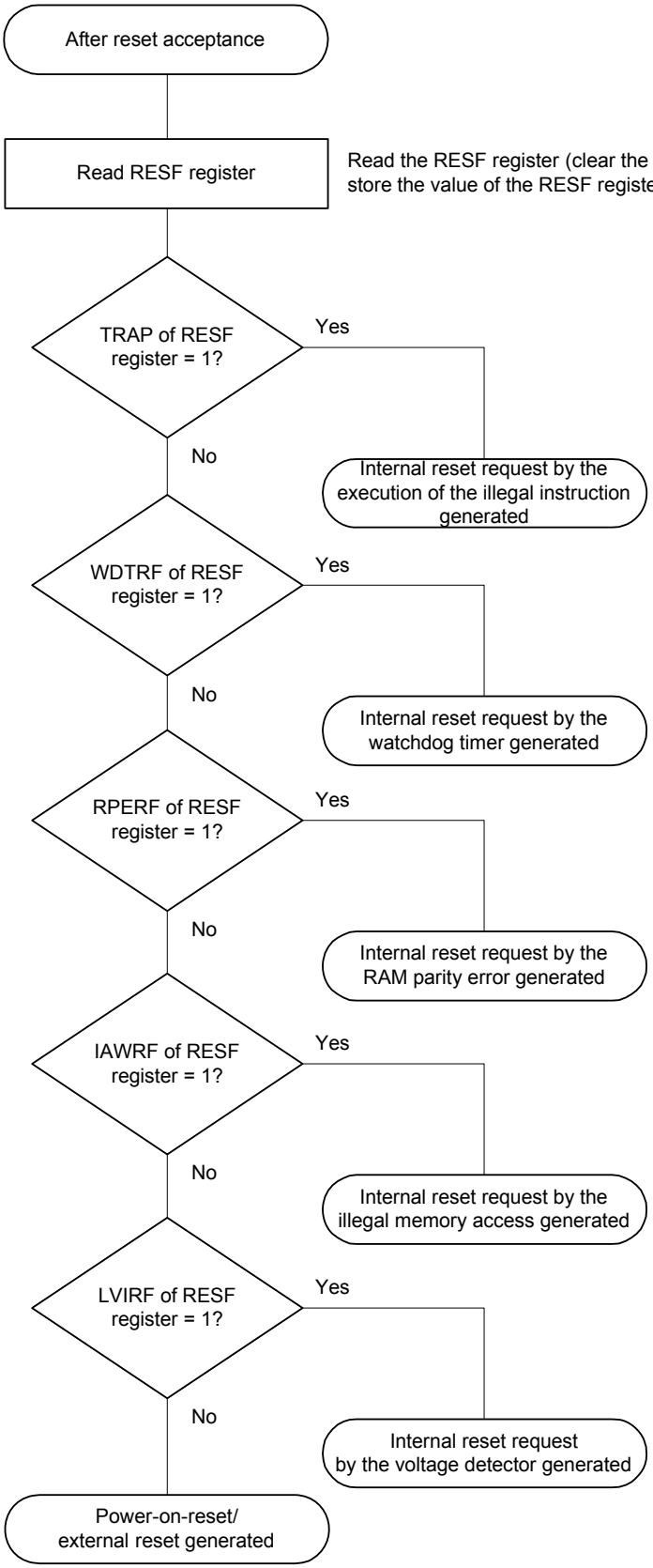
The status of the RESF register when a reset request is generated is shown in Table 21 - 3.

Table 21 - 3 RESF Register Status When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ 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
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)

Figure 21 - 5 shows the procedure for checking a reset source.

Figure 21 - 5 Procedure for Checking Reset Source



Caution The flow shown above is an example of procedure for checking.

CHAPTER 22 POWER-ON-RESET CIRCUIT

22.1 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 31.4 AC Characteristics. 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 31.4 AC Characteristics. 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) is cleared.

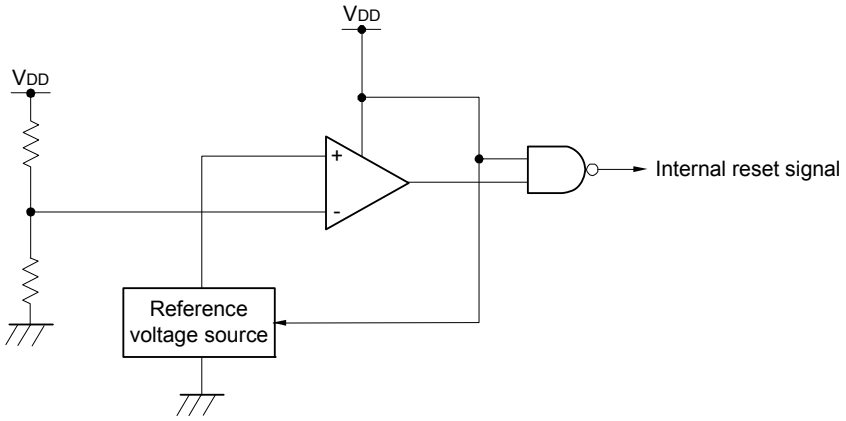
Remark 1. The RL78 microcontroller incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (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 21 RESET FUNCTION**.

Remark 2. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **31.6.2 POR characteristics**.

22.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 22 - 1.

Figure 22 - 1 Block Diagram of Power-on-reset Circuit

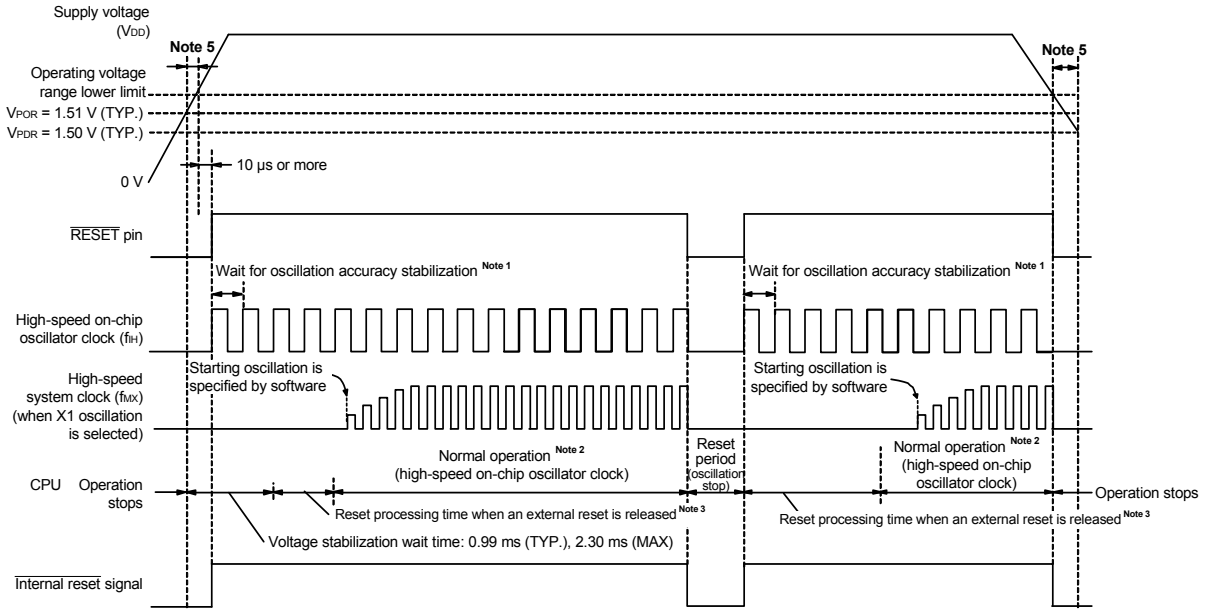


22.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown next.

Figure 22 - 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 $\overline{\text{RESET}}$ 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 or subsystem 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. To use the XT1 clock, use the timer function for confirmation of the lapse of the 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 $\overline{\text{RESET}}$ signal is driven high (1) as well as the voltage stabilization wait time after V_{POR} (1.51 V, TYP.) is reached.

Reset processing time when the external reset is released is shown below.

After the first release of $\overline{\text{RESET}}$ following 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)

Note 4. Reset processing time when the external reset is released after the second release of POR is shown below.

After the second release of $\overline{\text{RESET}}$ following 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 5. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **31.4 AC Characteristics**. This is done by controlling the externally input reset signal.

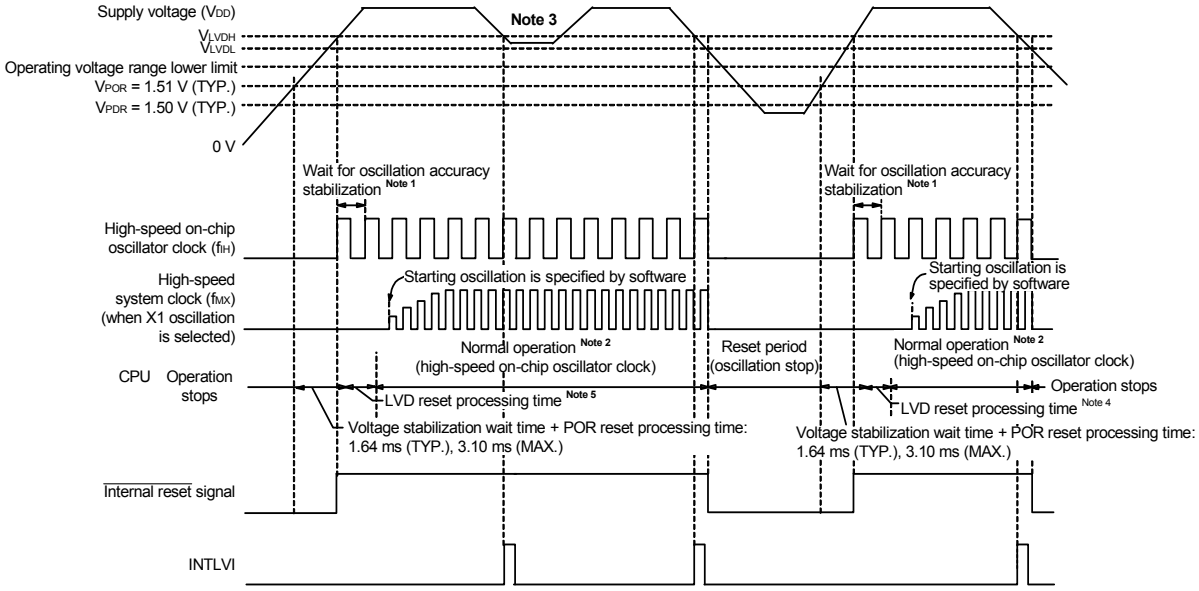
After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Caution For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when the LVD is off. For details, see CHAPTER 23 VOLTAGE DETECTOR.

Figure 22 - 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 or subsystem 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. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

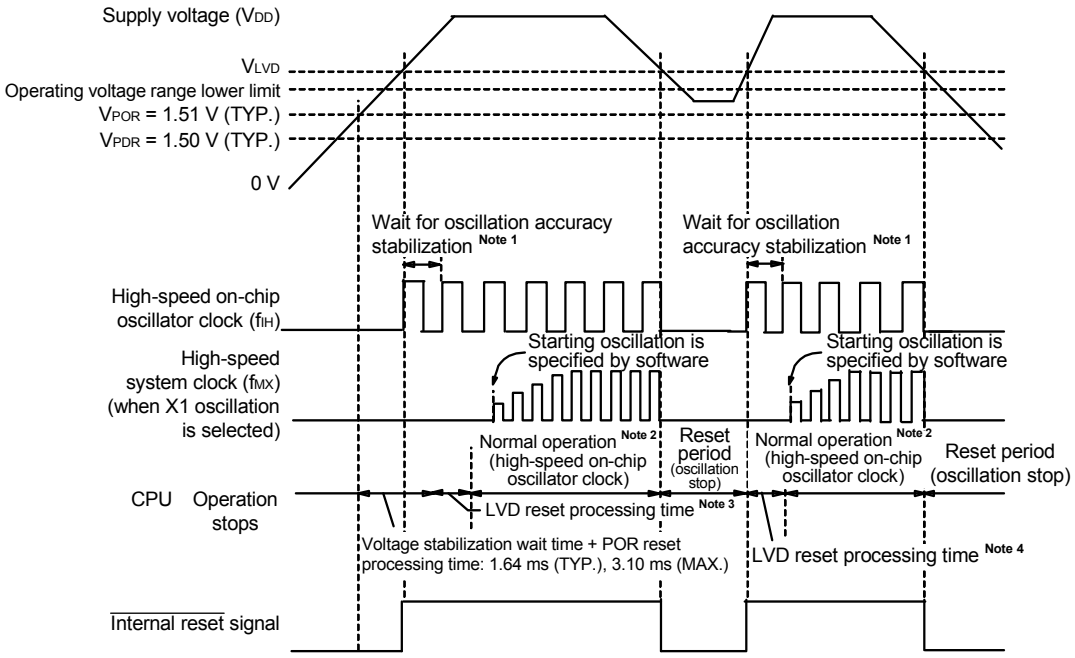
Note 3. After the interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 23 - 8 Setting Procedure for Operating Voltage Check 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 22 - 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 or subsystem 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. To use the XT1 clock, use the timer function for confirmation of the lapse of the 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 22 - 4 (3).

CHAPTER 23 VOLTAGE DETECTOR

23.1 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 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 by using the option byte as one of 14 levels (For details, see **CHAPTER 26 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 **31.4 AC Characteristics**. 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. 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.

(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 reset and internal interrupt signals are generated in each mode as follows.

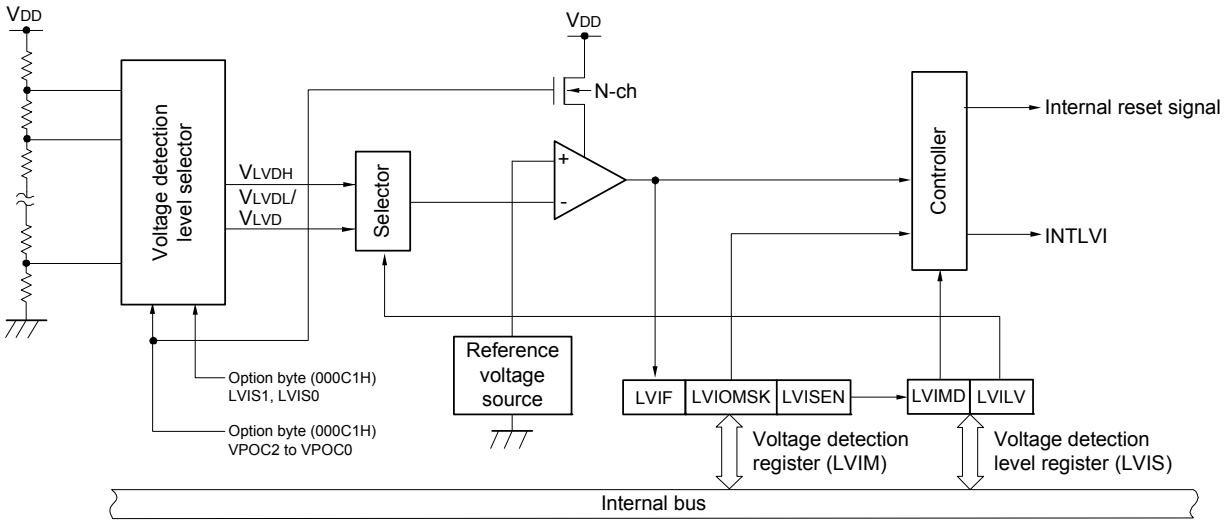
Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal 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 21 RESET FUNCTION**.

23.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 23 - 1.

Figure 23 - 1 Block Diagram of Voltage Detector



23.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

23.3.1 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 23 - 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 **Note 3**	0	0	0	0	0	LVIOMSK	LVIF

LVISEN **Note 3**	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 4**

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 LVIM 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. LVISEN can only be set in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not change the initial value in other modes.

Note 4. LVIOMSK bit is only automatically set to “1” when the interrupt & reset mode is selected (option byte LVIMDS1, LVIMDS0 = 1, 0) and reset or interrupt by LVD is masked.
- Period during LVISEN = 1
- Waiting period from the time when LVD interrupt is generated until LVD detection voltage becomes stable
- Waiting period from the time when the value of LVILV bit changes until LVD detection voltage becomes stable

23.3.2 Voltage detection level register (LVIS)

This register selects the voltage detection level.

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

Reset signal generation input sets this register to 00H/01H/81H Note 1.

Figure 23 - 3 Format of Voltage detection level register (LVIS)

Address: FFFAAH After reset:00H/01H/81H **Note 1** R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD **Note 2**	0	0	0	0	0	0	LVILV **Note 2**

LVIMD **Note 2**	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

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 reset source and the setting of the option byte.
This register is not cleared (00H) by LVD reset.
The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

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.

Caution 1. Rewrite the value of the LVIS register according to Figures 23 - 8.

Caution 2. Specify the LVD operation mode and detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H. For details about the user option byte, see CHAPTER 26 OPTION BYTE.

23.4 Operation of Voltage Detector

23.4.1 When used as reset mode

Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

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.
  Bit 7 (LVIMD) is 1 (reset mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

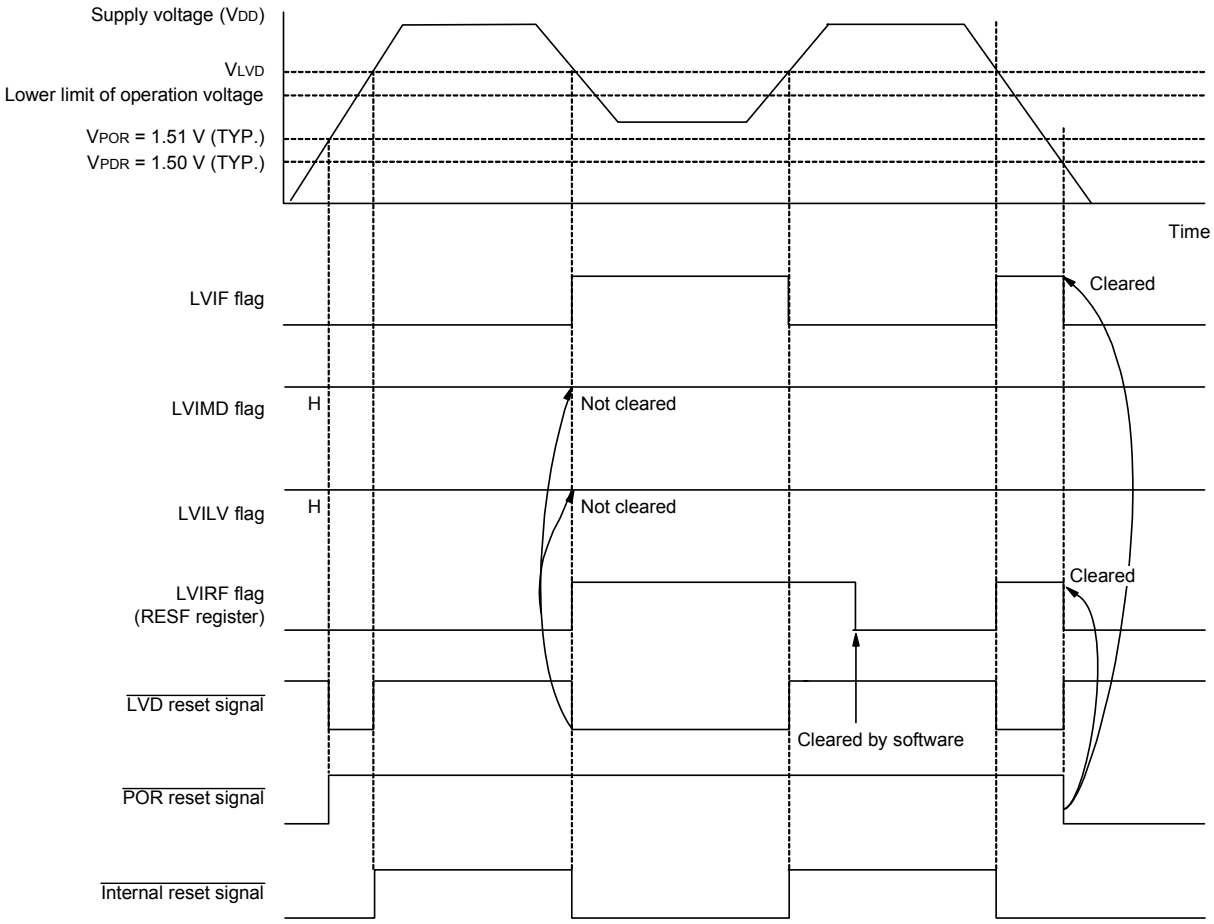
• Operation in LVD reset mode

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}).

Figure 23 - 4 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 23 - 4 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

23.4.2 When used as interrupt mode

Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

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))
- The initial value of the voltage detection level select register (LVIS) is set to 01H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 1 (voltage detection level: V_{LVD}).

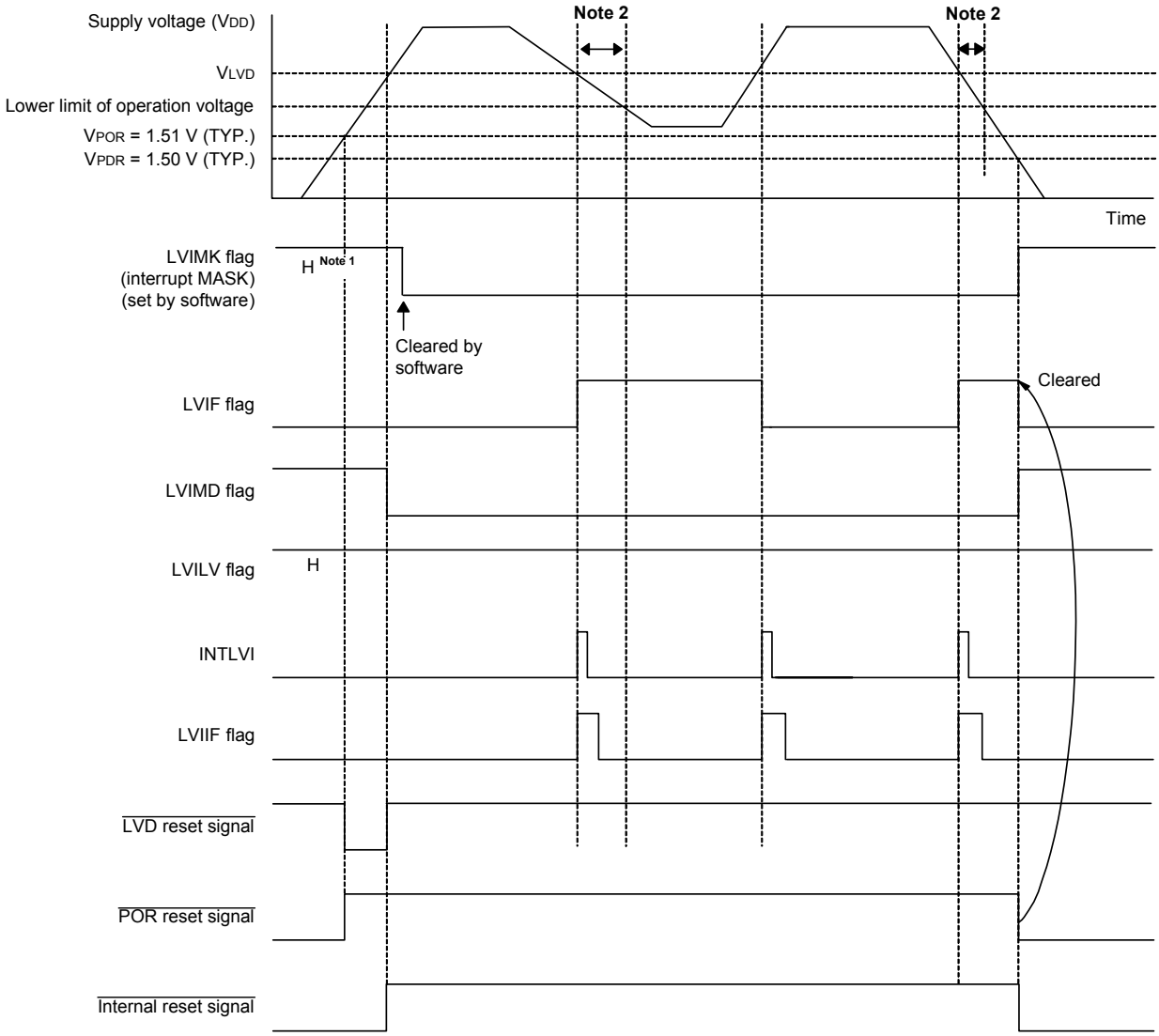
• Operation in LVD interrupt mode

In interrupt mode (LVIMDS1 and LVIMDS0 = 0 and 1 in the option byte), the state of an internal reset by the LVD is retained immediately after a reset until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). 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 31.4 AC Characteristics. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Figure 23 - 5 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 23 - 5 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 **31.4 AC Characteristics**. 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

23.4.3 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.

The operation is started in the following initial setting state when the interrupt & reset mode is set.

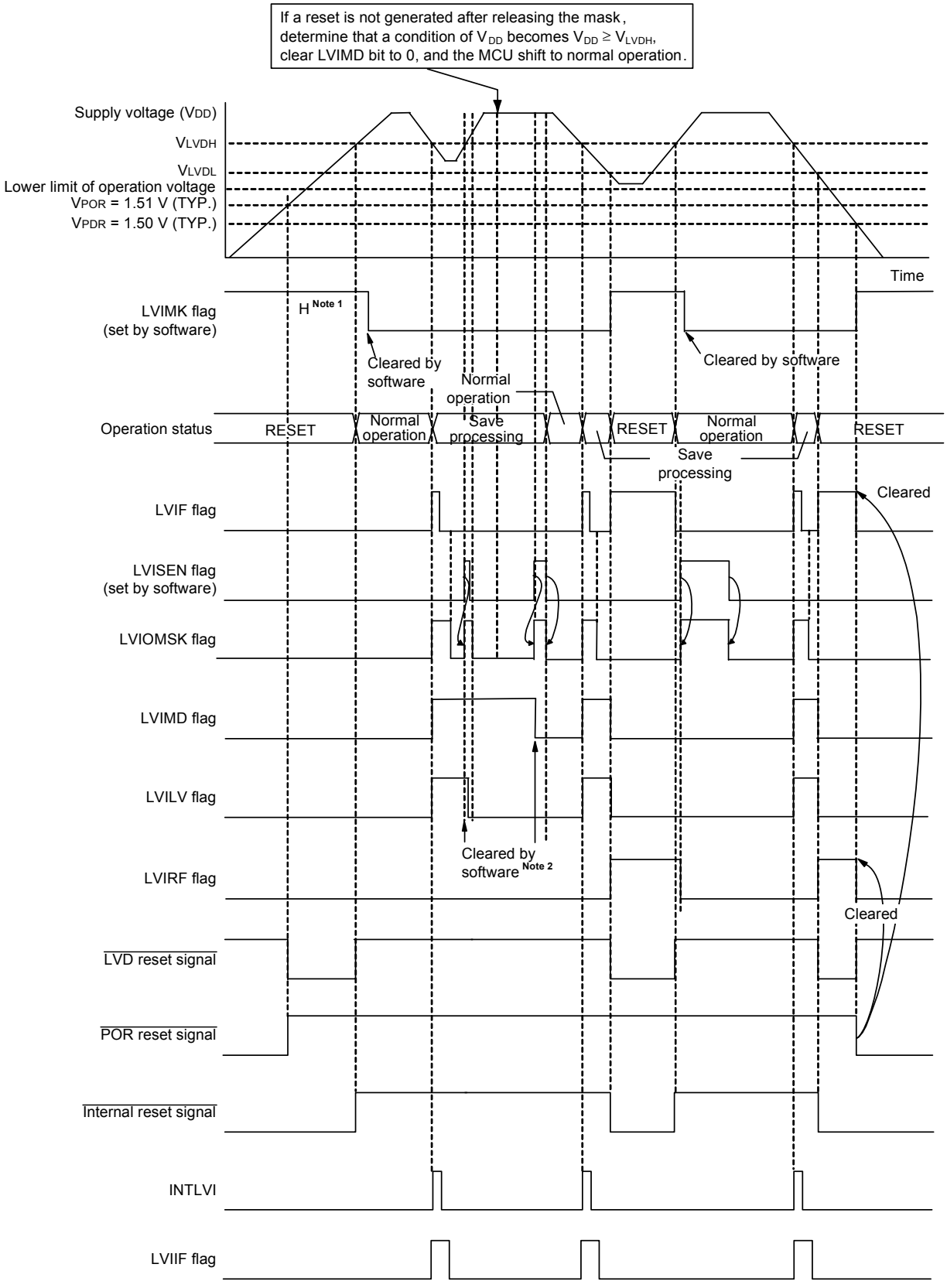
- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 00H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 0 (high-voltage detection 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 23 - 8 Setting Procedure for Operating Voltage Check and Reset**.

Figures 23 - 6 and 23 - 7 show the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 23 - 6 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



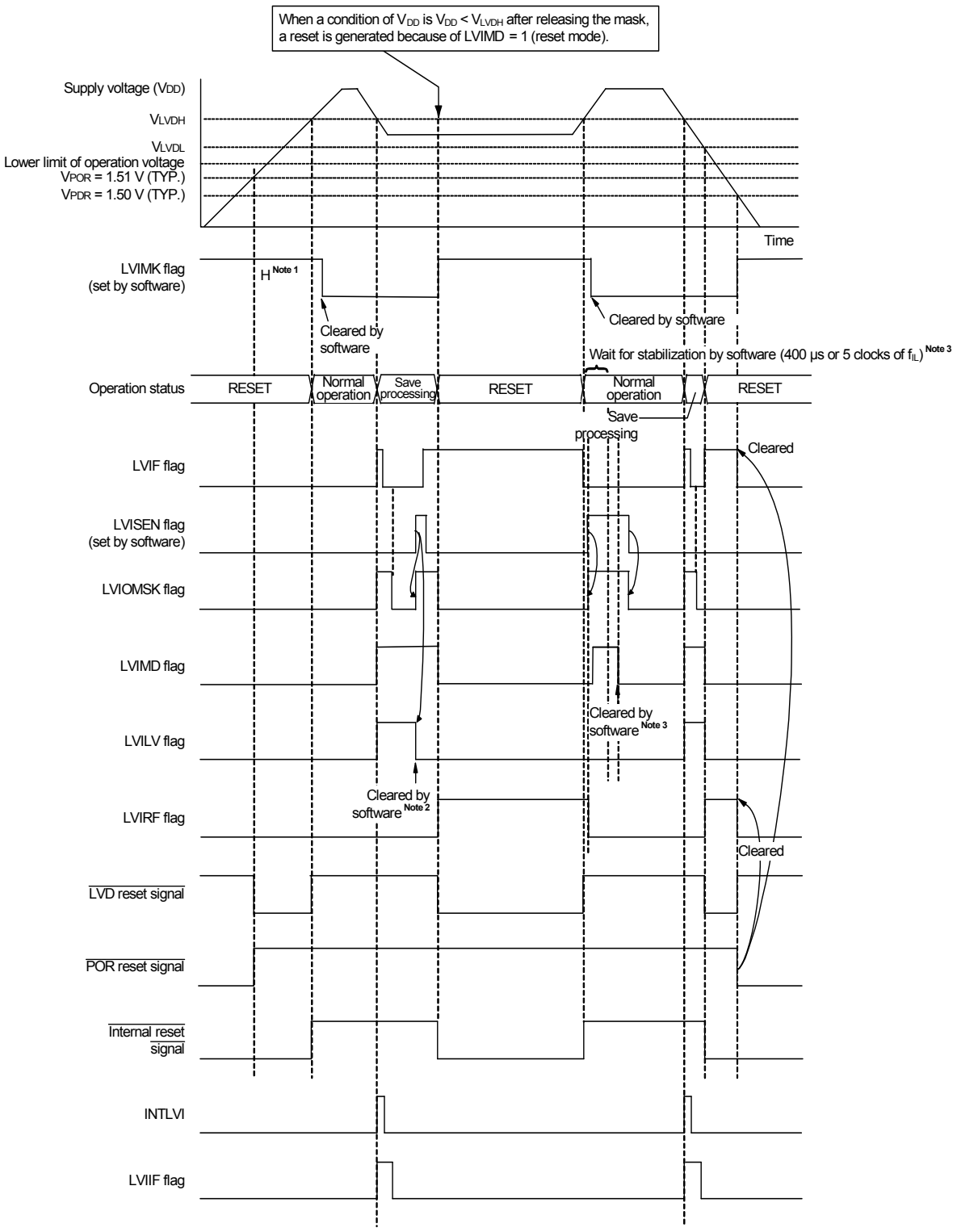
(Notes and Remark are listed on the next page.)

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 23 - 8 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 23 - 7 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.)

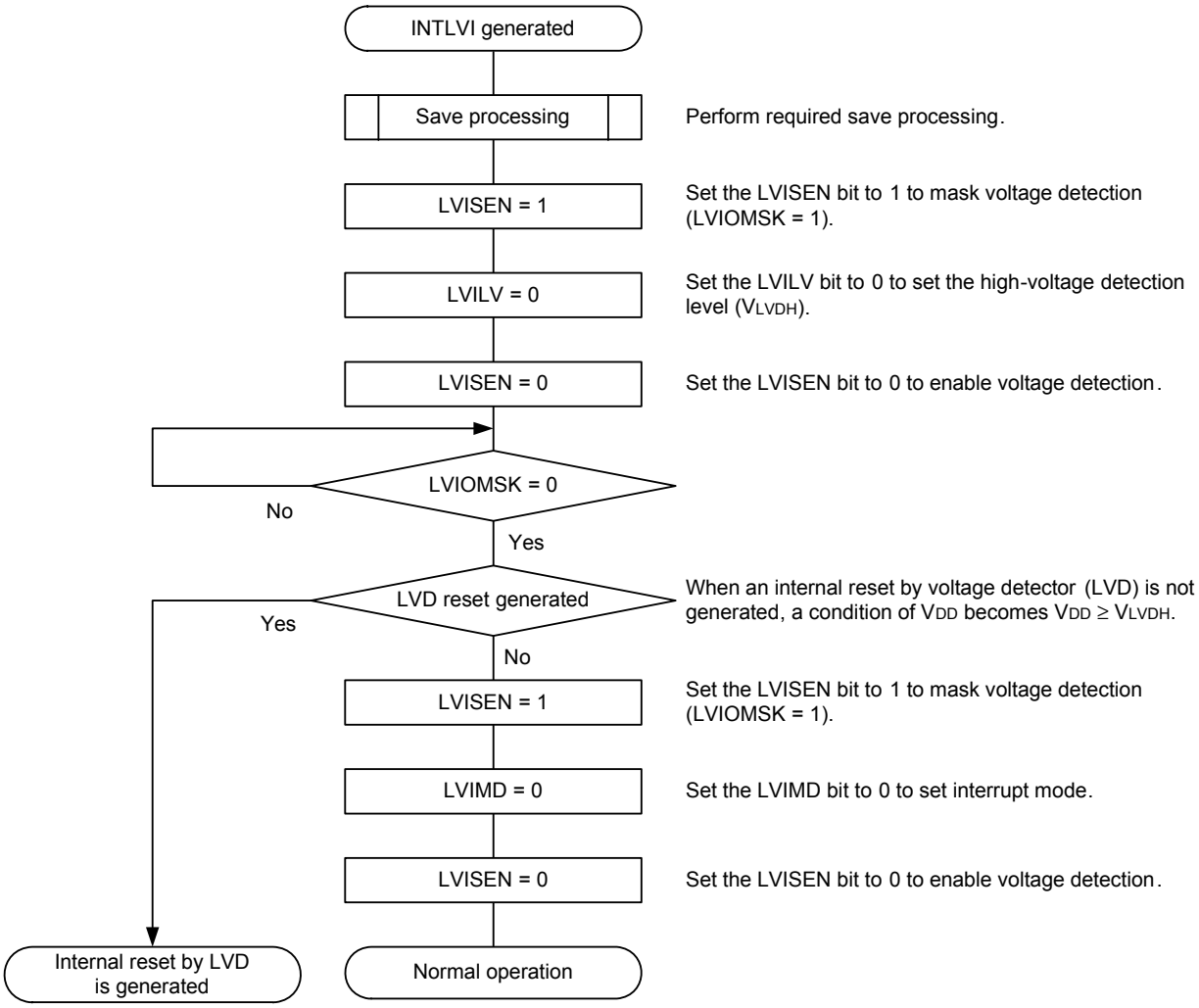
Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. After an interrupt is generated, perform the processing according to Figure 23 - 8 Setting Procedure for Operating Voltage Check and Reset in interrupt and reset mode.

Note 3. After reset is released, perform the processing according to Figure 23 - 8 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 23 - 8 Setting Procedure for Operating Voltage Check and Reset



23.5 Cautions for Voltage Detector

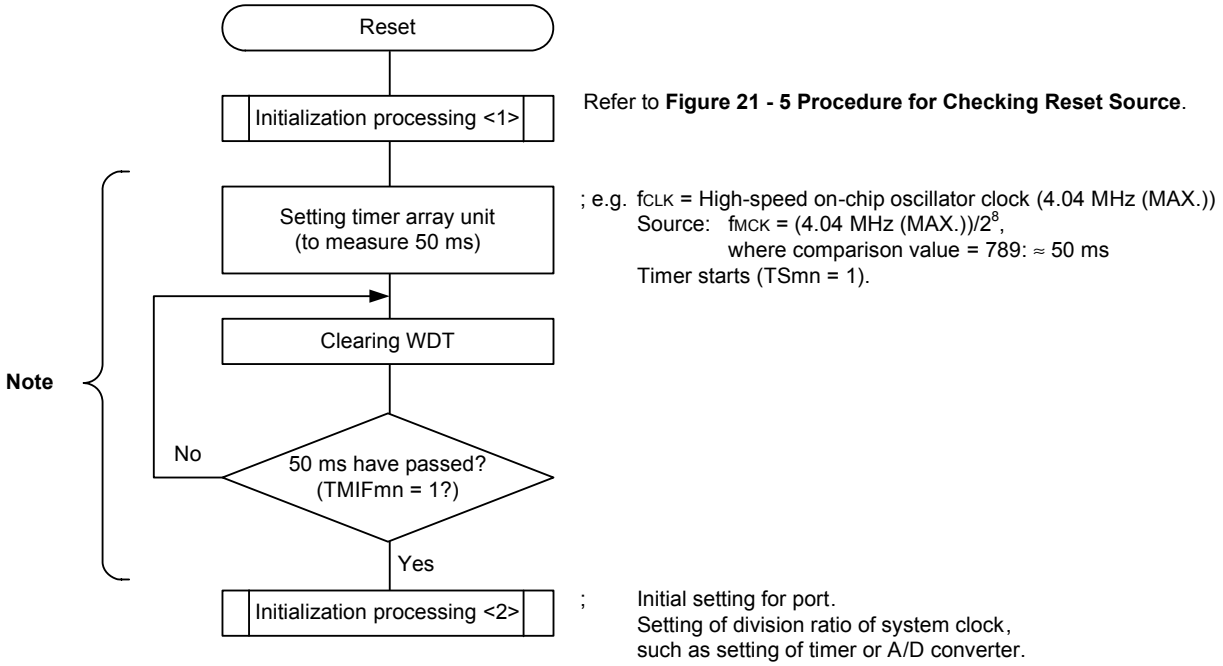
(1) Voltage fluctuation when power is supplied

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 microcontroller 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 23 - 9 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, 1
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. For details, see **31.6.3 LVD characteristics**.

(3) Power on when LVD is off

Use the external reset input via the $\overline{\text{RESET}}$ pin when the LVD is off.

For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ 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 **31.4 AC Characteristics**, 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 **31.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

CHAPTER 24 SAFETY FUNCTIONS

24.1 Overview of Safety Functions

The following safety functions are provided in the RL78/G1H 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 the RL78/G1H 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).

(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.

24.2Registers 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 24.3 Operation of Safety Functions.

24.3 Operation of Safety Functions

24.3.1 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 the RL78/G1H 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.

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.

24.3.1.1 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 24 - 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	00000H to 03FFBH (16 K - 4 bytes)
0	0	0	0	0	1	00000H to 07FFBH (32 K - 4 bytes)
0	0	0	0	1	0	00000H to 0BFFBH (48K - 4 bytes)
0	0	0	0	1	1	00000H to 0FFFBH (64K - 4 bytes)
0	0	0	1	0	0	00000H to 13FFBH (80K - 4 bytes)
0	0	0	1	0	1	00000H to 17FFBH (96K - 4 bytes)
0	0	0	1	1	0	00000H to 1BFFBH (112K - 4 bytes)
0	0	0	1	1	1	00000H to 1FFFBH (128K - 4 bytes)
0	0	1	0	0	0	00000H to 23FFBH (144K - 4 bytes)
0	0	1	0	0	1	00000H to 27FFBH (160K - 4 bytes)
0	0	1	0	1	0	00000H to 2BFFBH (176K - 4 bytes)
0	0	1	0	1	1	00000H to 2FFFBH (192K - 4 bytes)
0	0	1	1	0	0	00000H to 33FFBH (208K - 4 bytes)
0	0	1	1	0	1	00000H to 37FFBH (224K - 4 bytes)
0	0	1	1	1	0	00000H to 3BFFBH (240K - 4 bytes)
0	0	1	1	1	1	00000H to 3FFFBH (256K - 4 bytes)
0	1	0	0	0	0	00000H to 43FFBH (272 K-4 bytes)
0	1	0	0	0	1	00000H to 47FFBH (288 K-4 bytes)
0	1	0	0	1	0	00000H to 4BFFBH (304 K-4 bytes)
0	1	0	0	1	1	00000H to 4FFFBH (320 K-4 bytes)
0	1	0	1	0	0	00000H to 53FFBH (336 K-4 bytes)
0	1	0	1	0	1	00000H to 57FFBH (352 K-4 bytes)
0	1	0	1	1	0	00000H to 5BFFBH (368 K-4 bytes)
0	1	0	1	1	1	00000H to 5FFFBH (384 K-4 bytes)
0	1	1	0	0	0	00000H to 63FFBH (400 K-4 bytes)
0	1	1	0	0	1	00000H to 67FFBH (416 K-4 bytes)
0	1	1	0	1	0	00000H to 6BFFBH (432 K-4 bytes)
0	1	1	0	1	1	00000H to 6FFFBH (448 K-4 bytes)
0	1	1	1	0	0	00000H to 73FFBH (464 K-4 bytes)
0	1	1	1	0	1	00000H to 77FFBH (480 K-4 bytes)
0	1	1	1	1	0	00000H to 7BFFBH (496 K-4 bytes)
0	1	1	1	1	1	00000H to 7FFFBH (512 K-4 bytes)
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.

24.3.1.2 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 24 - 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

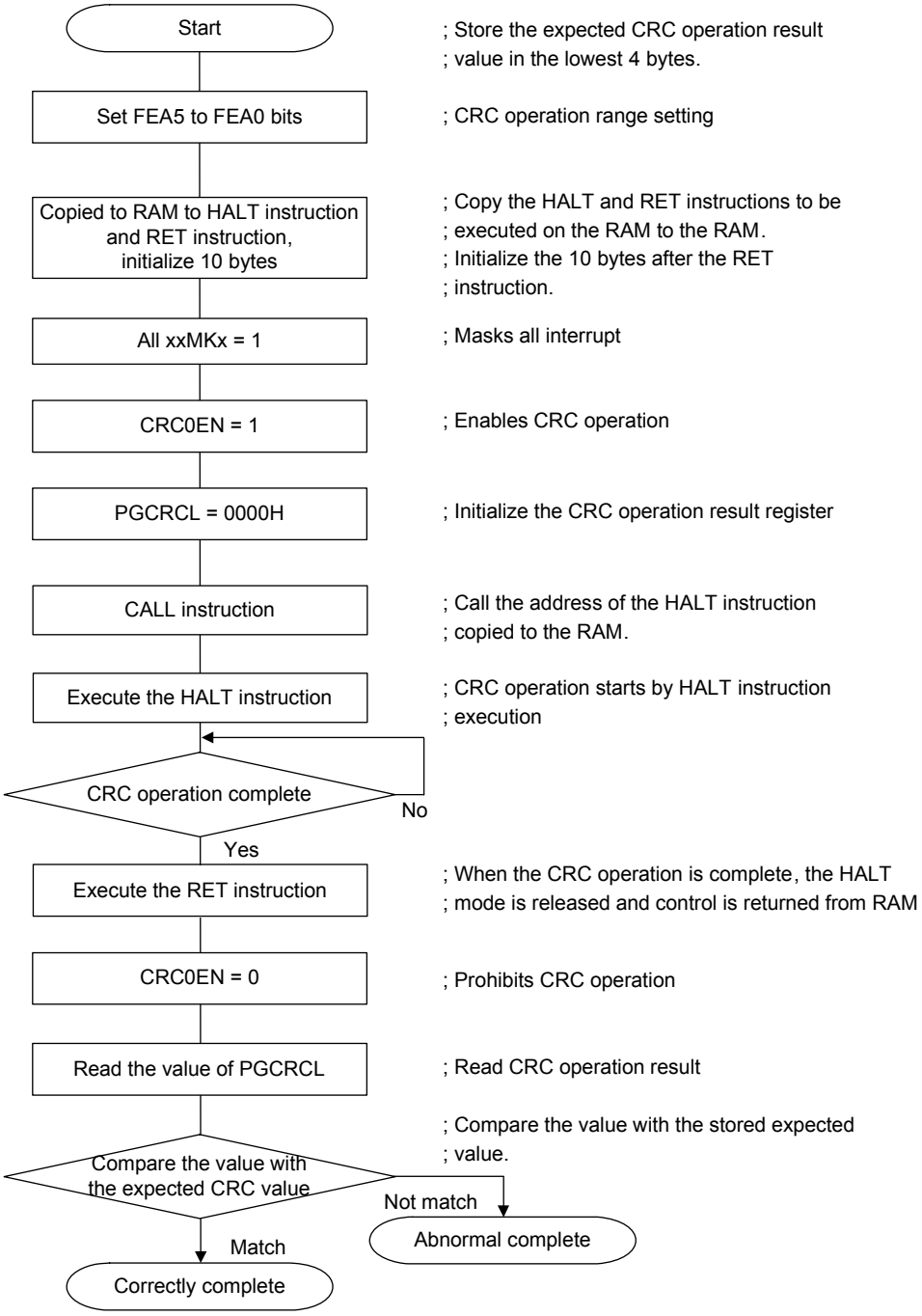
PGCRC15 to PGCRC0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 24 - 3 shows the Flowchart of Flash Memory CRC Operation Function (High-speed CRC).

<Operation flow>

Figure 24 - 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 CS+ development environment. Refer to the CS+ integrated development environment user's manual for details.

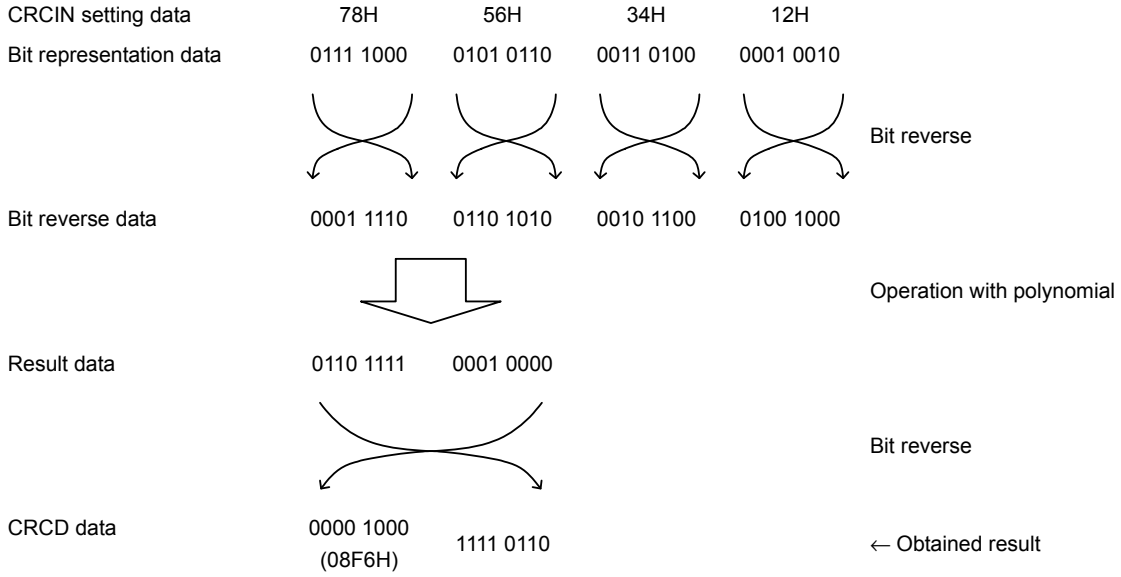
24.3.2 CRC operation function (general-purpose CRC)

In order to guarantee safety during operation, the IEC61508 standard mandates the checking of data even while the CPU is operating.

In the RL78/G1H, 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). In HALT mode, the CRC operation function can be used only during DTC transfer.

The general CRC operation can be executed in the main system clock operation mode as well as the subsystem clock operation mode.

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.

24.3.2.1 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 24 - 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.

24.3.2.2 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 24 - 5 Format of CRC data register (CRCD)

Address: F02FAH After reset: 0000H R/W

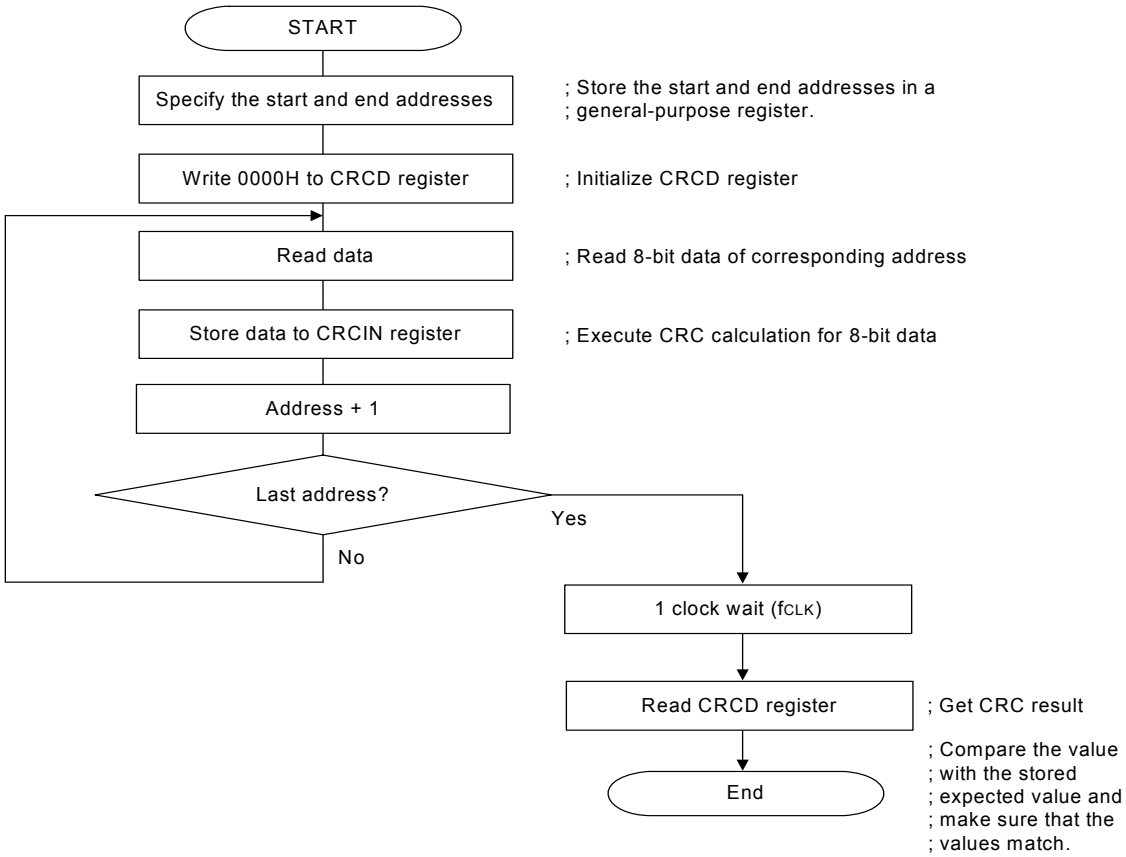
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

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 24 - 6 CRC Operation Function (General-Purpose CRC)



24.3.3 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 the RL78/G1H'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.

24.3.3.1 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 24 - 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

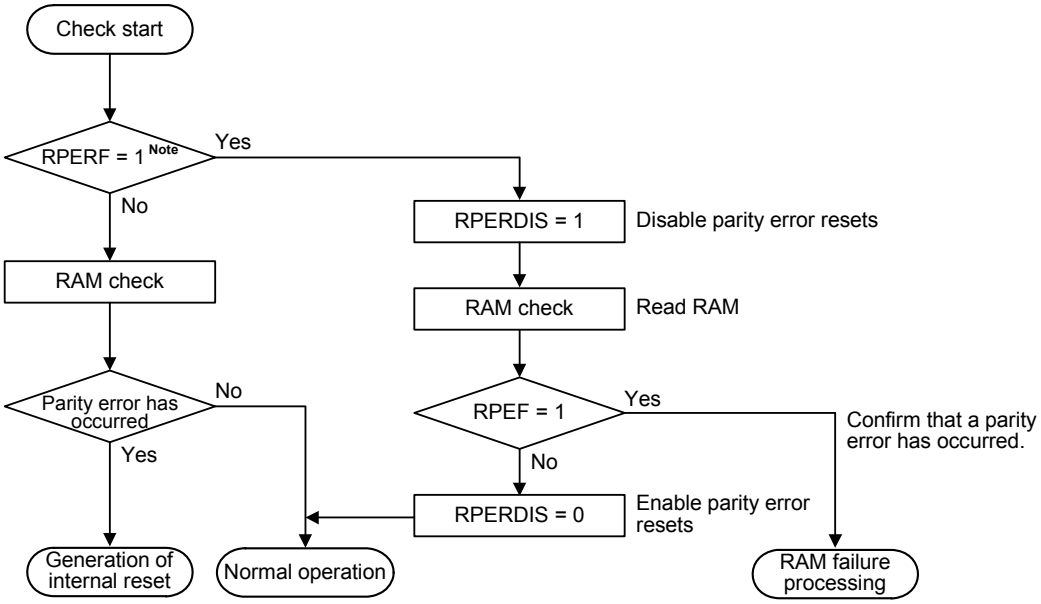
RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

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.
The RL78'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 RPEF flag in the RPECTL 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 24 - 8 RAM Parity Error Check Flow



Note See **CHAPTER 21 RESET FUNCTION** for details on how to confirm internal resets due to RAM parity errors.

24.3.4 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.

24.3.4.1 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 24 - 9 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H 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.

24.3.5 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.

24.3.5.1 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 24 - 10 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H 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, ADPC, PIORx **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, OSMC, LVIM, LVIS, RPECTL

Note Pxx (Port register) is not guarded.

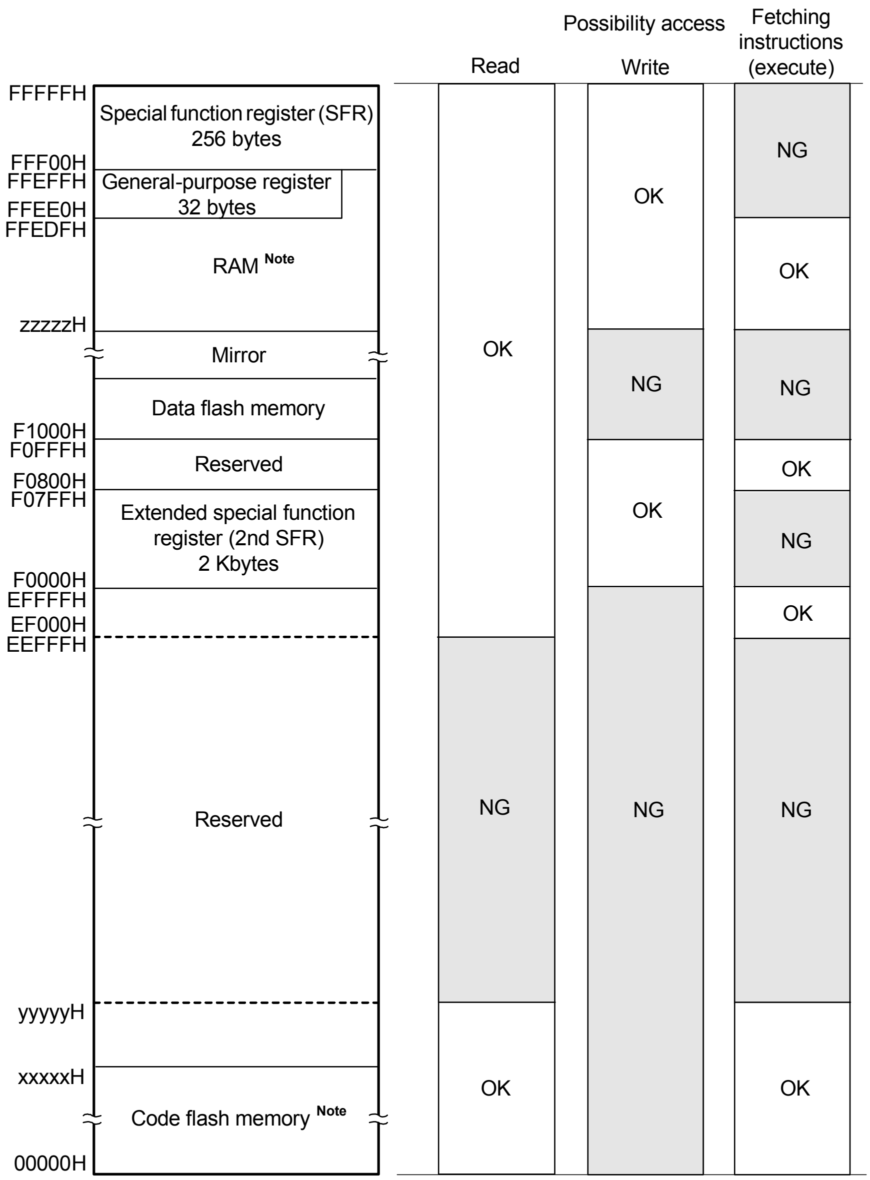
24.3.6 Invalid memory access detection function

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 24 - 11.

Figure 24 - 11 Invalid access detection area



Note The code flash memory, RAM, and lowest detection address of each product are as follows.

Products	Code Flash Memory (00000H to xxxxxH)	RAM (zzzzzH to FFEFFH)	Lowest Detection Address (yyyyyH) when Reading/Fetching (Executing) Instructions
R5F11FLJ	262144 × 8 bits (00000H to 3FFFFH)	24576 × 8 bit (F9F00H to FFEFFH)	40000H
R5F11FLK	393216 × 8 bits (00000H to 5FFFFH)	32768 × 8 bit (F7F00H to FFEFFH)	60000H
R5F11FLL	524688 × 8 bits (00000H to 7FFFFH)	49152 × 8 bit (F3F00H to FFEFFH)	80000H

24.3.6.1 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 24 - 12 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	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.

24.3.7 Frequency detection function

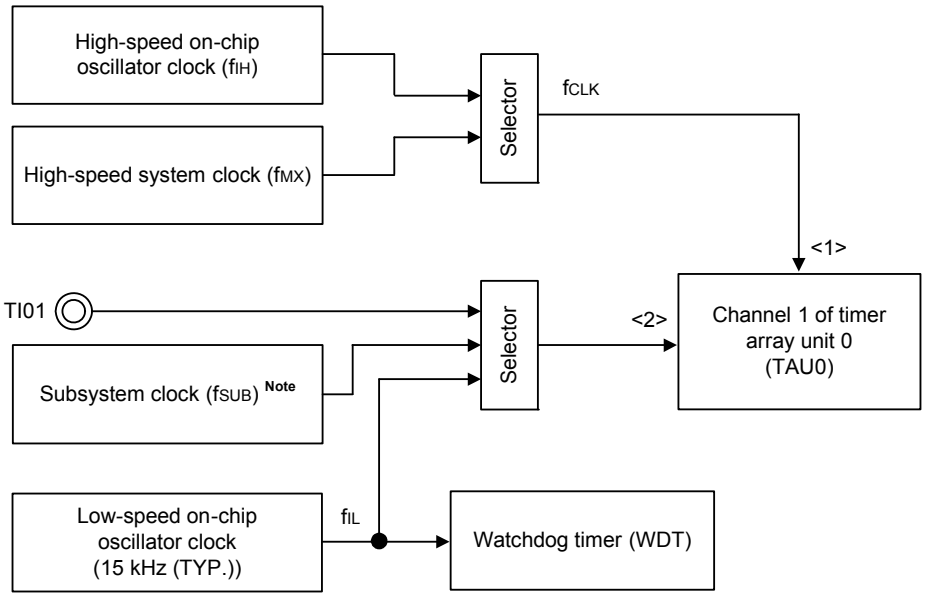
The IEC60730 standard mandates checking that the oscillation frequency is correct.
By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined.
Note that, however, if one or both clock operations are 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.))
- Subsystem clock (f_{SUB}) Note

Figure 24 - 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 **7.8.3 Operation as input pulse interval measurement**.

Note Can only be selected in the products incorporating the subsystem clock.

24.3.7.1 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 24 - 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	TIS04	0	TIS02	TIS01	TIS00

TIS04	Selection of timer input used with channel 0
0	No timer input
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	No timer input
0	0	1	Event input signal from ELC
0	1	0	Setting prohibited
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

24.3.8 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). 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.

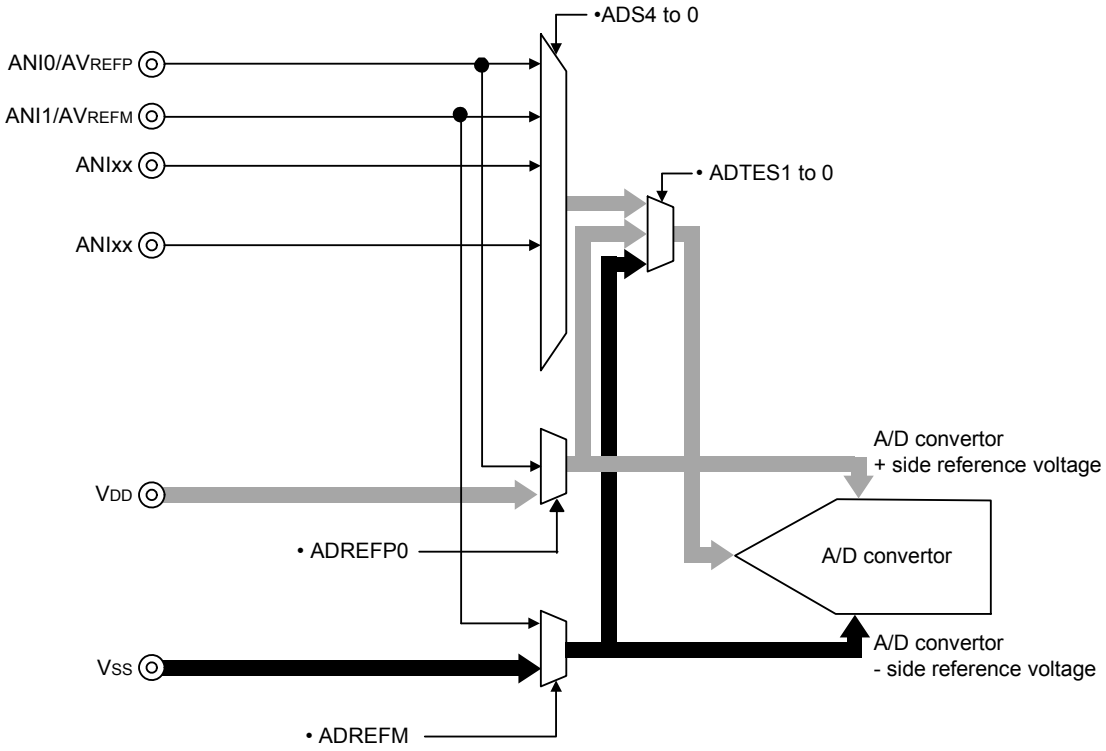
(1) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(2) Perform A/D conversion for the ANIx pin (conversion result 1-1).
(3) Select the negative reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (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.

Figure 24 - 15 Configuration of A/D Test Function



24.3.8.1 A/D test register (ADTES)

This register is used to select the A/D converter positive reference voltage, negative reference voltage, analog input channel (ANIxx) as the target of A/D conversion.

When using the A/D test function, specify the following settings:

- Select the negative reference voltage as the target of A/D conversion when measuring the zero-scale.
- Select the positive reference voltage as the target of A/D conversion when measuring the full-scale.

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

Reset signal generation clears this register to 00H.

Figure 24 - 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 (This is specified using the analog input channel specification register (ADS).)
1	0	Negative reference voltage (selected by the ADREFM bit in the ADM2 register)
1	1	Positive reference voltage (selected by the ADREFP0 bit in the ADM2 register)
Other than the above		Setting prohibited

24.3.8.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

Set A/D test register (ADTES) to 00H when measuring the ANIxx.

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 24 - 17 Format of Analog input channel specification register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	ANI0	P20/ANI0/AVREFP pin
0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	1	0	ANI2	P22/ANI2 pin
0	1	1	0	1	ANI13	P155/ANI13 pin
0	1	1	1	0	ANI14	P156/ANI14 pin
1	0	0	1	1	ANI19	P120/ANI19 pin
Other than the above					Setting prohibited	

Caution 1. Be sure to clear bits 5 to 7 to 0.

Caution 2. For ports that set to analog input using the ADPC and PMC registers, select input mode using port mode register 2, 12, or 15 (PM2, PM12, PM15).

Caution 3. Do not use the ADS register to set ports that to be set as digital I/O using the A/D port configuration register (ADPC).

Caution 4. Do not use the ADS register to set ports that to be set as digital I/O using port mode control register 12 (PMC12).

Caution 5. When using AVREFP as the positive reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 6. When using AVREFM as the negative reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.

24.3.9 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.

24.3.9.1 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 PMmn 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 these registers to 00H.

Figure 24 - 18 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

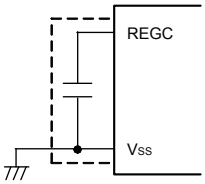
PMS0	Method for selecting output level to be read when pin is output mode
0	Pmn register value is read.
1	Digital output level of the pin is read.

Caution 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.

CHAPTER 25 REGULATOR

25.1 Regulator Overview

The RL78/G1H 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 25 - 1**.

Table 25 - 1 Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
LS (low-speed main) mode	1.8 V	—
HS (high-speed main) mode	1.8 V	In STOP mode
		When both the high-speed system clock (fSUB) and the high-speed on-chip oscillator clock (fIH) are stopped during CPU operation with the subsystem clock (fXT)
		When both the high-speed system clock (fSUB) and the high-speed on-chip oscillator clock (fIH) are stopped during the HALT mode when the CPU operation with the subsystem clock (fXT) has been set
	2.1 V	Other than above (include during OCD mode) **Note**

Note When it shifts to the subsystem clock operation or STOP mode during the on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.8 V).

CHAPTER 26 OPTION BYTE

26.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory 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.

26.1.1 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 $\overline{\text{RESET}}$ 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 31.4 AC Characteristics. 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
- LS (low-speed main) mode
- HS (high-speed main) mode

○ Setting of the frequency of the high-speed on-chip oscillator
- Select from 1 MHz to 32 MHz.

Caution Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

26.1.2 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.

26.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 26 - 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.

<R>

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 to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)	237.44 to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)	474.89 to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)	1899.59 to 2570.04 ms

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

Figure 26 - 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
1.98 V	1.94 V	1.84 V	0	0	1	1	0	1	0
2.09 V	2.04 V					0	1		
3.13 V	3.06 V					0	0		
2.61 V	2.55 V	2.45 V		1	0	1	0		
2.71 V	2.65 V					0	1		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	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 Be sure to set bit 4 to "1".

Remark 1. For details on the LVD circuit, see **CHAPTER 23 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **31.6.3 LVD characteristics**.

Figure 26 - 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
1.88 V	1.84 V	0	0	0	1	1	1	1
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		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		
—		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 Be sure to set bit 4 to "1".

Remark 1. For details on the LVD circuit, see **CHAPTER 23 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see 31.6.3 LVD characteristics.

Figure 26 - 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
1.88 V	1.84 V	0	0	0	1	1	1	0	1
1.98 V	1.94 V		0	0	1	1	0	
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		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		
—		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 Be sure to set bit 4 to "1".

Remark 1. For details on the LVD circuit, see **CHAPTER 23 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **31.6.3 LVD characteristics**.

Figure 26 - 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 $\overline{\text{RESET}}$ 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 31.4 AC Characteristics. 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 23 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a typical value. For details, see **31.6.3 LVD characteristics**.

Figure 26 - 6 Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H **Note**

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode		
			Operating Frequency Range	Operating Voltage Range
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 3.6 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 3.6 V
			1 to 32 MHz	2.7 to 3.6 V
Other than above		Setting prohibited		

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator clock
				f_{IH}
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

Note Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

Caution 1. Be sure to set bit 5 to 1.

Caution 2. The operating frequency range and operating voltage range depend on each operating mode of the flash memory. See 31.4 AC Characteristics for details.

26.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 26 - 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 erase 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.

CHAPTER 27 FLASH MEMORY

The RL78 microcontroller 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. Refer to 4.1 Memory Space.

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 **27.1**)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.
- Self-Programming (see **27.5**)
  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 **27.7 Data Flash**.

27.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP5, FL-PR5
- E1 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 the RL78 microcontroller 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 (FA series) before the RL78 microcontroller is mounted on the target system.

Remark FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 27 - 1 Wiring Between RL78/G1H and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.
Signal Name		I/O	Pin Function		
PG-FP5, FL-PR5	E1 on-chip Debugging Emulator				
SI/RxD	TOOL0	I/O	Transmit/ receive signal	TOOL0/ P40	6
/RESET	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	7
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	15
GND		—	Ground	V_{SS}	14
				REGC **Note**	13
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}	15

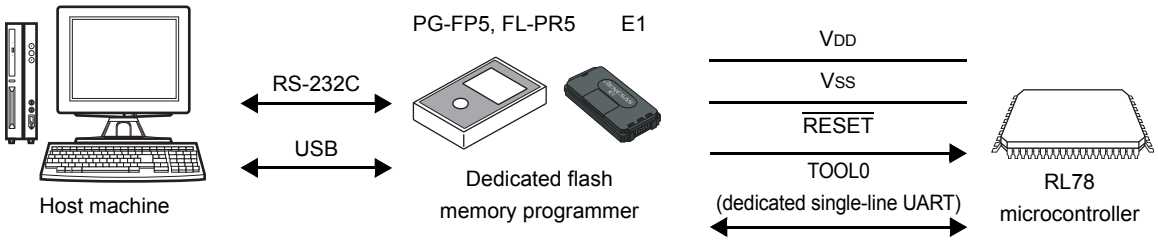
Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

27.1.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 27 - 1 Environment for Writing Program to Flash Memory



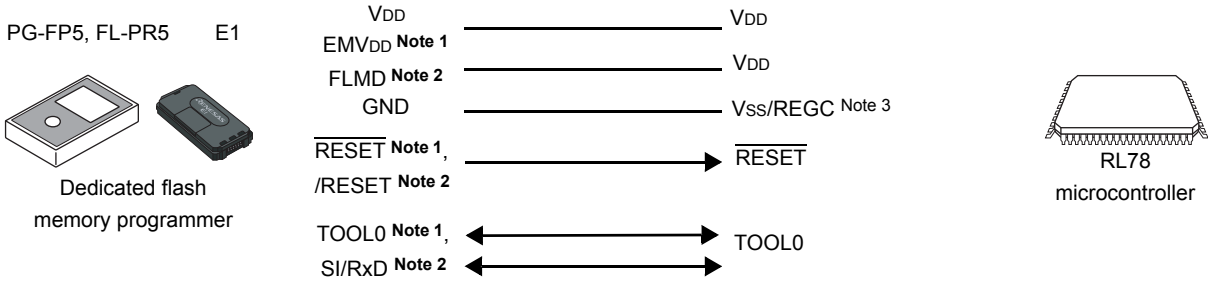
A host machine that controls the dedicated flash memory programmer is necessary.
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

27.1.2 Communication Mode

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 27 - 2 Communication with Dedicated Flash Memory Programmer



Note 1. When using E1 on-chip debugging emulator.
Note 2. When using PG-FP5 or FL-PR5.
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 the RL78 microcontroller. See the manual of PG-FP5, FL-PR5, or E1 on-chip debugging emulator for details.

Table 27 - 2 Pin Connection

Dedicated Flash Memory Programmer				RL78/G1H
Signal Name		I/O	Pin Function	Pin Name
PG-FP5, FL-PR5	E1 on-chip debugging emulator			
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		—	Ground	V_{SS}, REGC **Note**
EMV_{DD}		—	Driving power for TOOL0 pin	V_{DD}
/RESET	—	Output	Reset signal	$\overline{\text{RESET}}$
—	$\overline{\text{RESET}}$	Output		
—	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	—	I/O	Transmit/receive signal	

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

27.2 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 Refer to flash programming mode, see **27.3.2 Flash memory programming mode**.

27.2.1 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 (see 31.11 Timing for Switching Flash Memory Programming Modes).

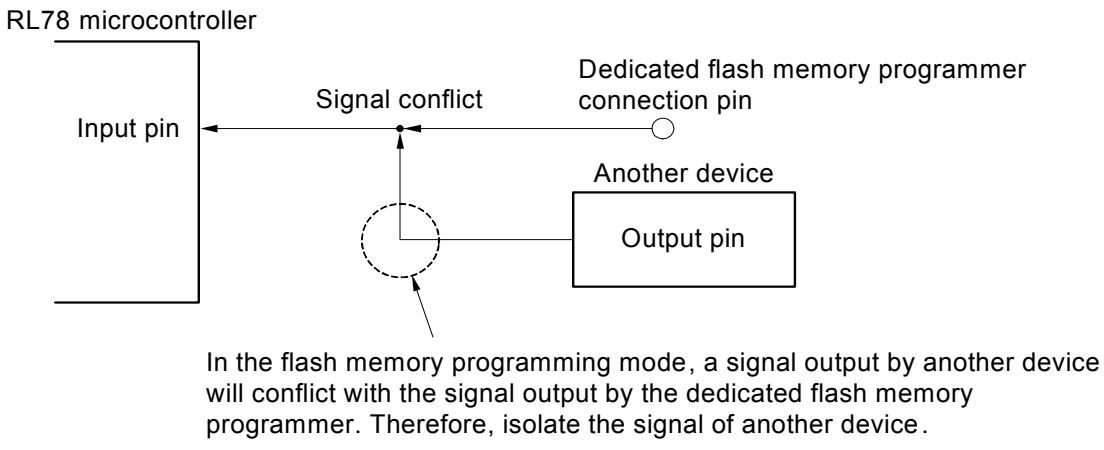
Remark 2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

27.2.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the $\overline{\text{RESET}}$ 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 27 - 3 Signal Conflict ($\overline{\text{RESET}}$ Pin)



27.2.3 Port pins

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 either V_{DD}, or V_{SS}, via a resistor.

27.2.4 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.

27.2.5 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.

27.2.6 Power supply

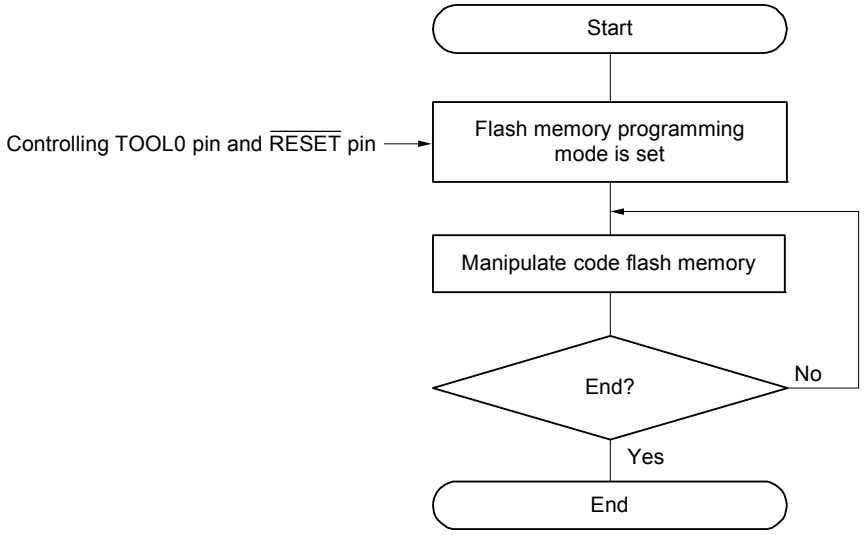
To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.
To use the on-board supply voltage, connect in compliance with the normal operation mode.
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

27.3 Programming Method

27.3.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 27 - 4 Code Flash Memory Manipulation Procedure



27.3.2 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.

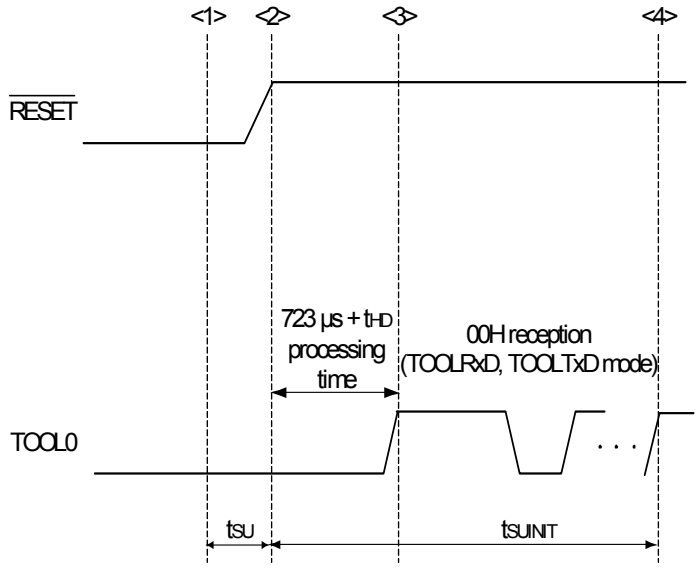
<When serial programming by using the dedicated flash memory programmer>
Connect the RL78 microcontroller to a dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

<When serial programming by using an external device>
Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 27 - 3**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 27 - 5**. For details, refer to the **RL78 microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 27 - 3 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 27 - 5 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, see **31.11 Timing for Switching Flash Memory Programming Modes**.

There are two flash memory programming modes: wide voltage mode and full speed mode. The supply voltage value applied to the microcontroller 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 27 - 4 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} ≤ 3.6 V	Blank state		Full speed mode
	HS (high-speed main) mode	1 MHz to 32 MHz	Full speed mode
	LS (low-speed main) mode	1 MHz to 8 MHz	Wide voltage mode
2.4 V ≤ V_{DD} < 2.7 V	Blank state		Full speed mode
	HS (high-speed main) mode	1 MHz to 16 MHz	Full speed mode
	LS (low-speed main) mode	1 MHz to 8 MHz	Wide voltage mode
1.8 V ≤ V_{DD} < 2.4 V	Blank state		Wide voltage mode
	LS (low-speed main) mode	1 MHz to 8 MHz	Wide voltage mode

Remark 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.
Remark 2. For details about communication commands, see **27.3.4 Communication commands**.

27.3.3 Selecting communication mode

Communication mode of the RL78 microcontroller as follows.

Table 27 - 5 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

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.

27.3.4 Communication commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 27 - 6**.
The signals sent from the dedicated flash memory programmer or external device to the RL78 microcontroller 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 27 - 6 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 the RL78 microcontroller 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 27 - 7 show signature data list and example of signature data list.

Table 27 - 7 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.	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address.	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address.	3 bytes

27.4 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP5 is used as a dedicated flash memory programmer.

Table 27 - 8 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

PG-FP5 Command	Port: TOOL0 (UART)		
	Speed: 1M bps		
	256 Kbytes	384 Kbytes	512 Kbytes
Erasing	2.5 s	3 s	4 s
Writing	6 s	8.5 s	11 s
Verification	5.5 s	8 s	10.5 s
Writing after erasing	8 s	11 s	14.5 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)

27.5 Self-Programming

The RL78 microcontroller 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. The self-programming function cannot be used when the CPU operates with the subsystem clock.

Caution 2. 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 3. The high-speed on-chip oscillator should be kept operating during self-programming. If it is kept stopped, it should be operated (HIOSTOP = 0). The flash self-programming library should be executed after 30 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 (R01AN0350)**.

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. Specify the wide voltage mode when the LS (low-speed main) mode is specified.

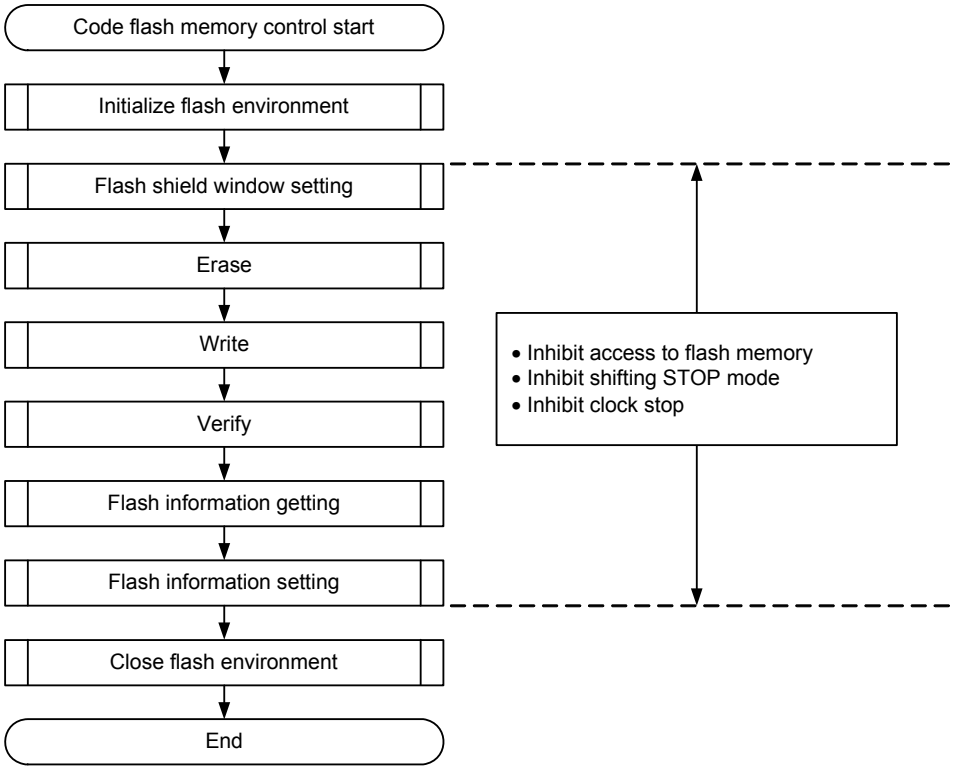
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

Remark Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

27.5.1 Self-programming procedure

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

Figure 27 - 6 Flow of Self-Programming (Rewriting Flash Memory)



27.5.2 Boot swap function

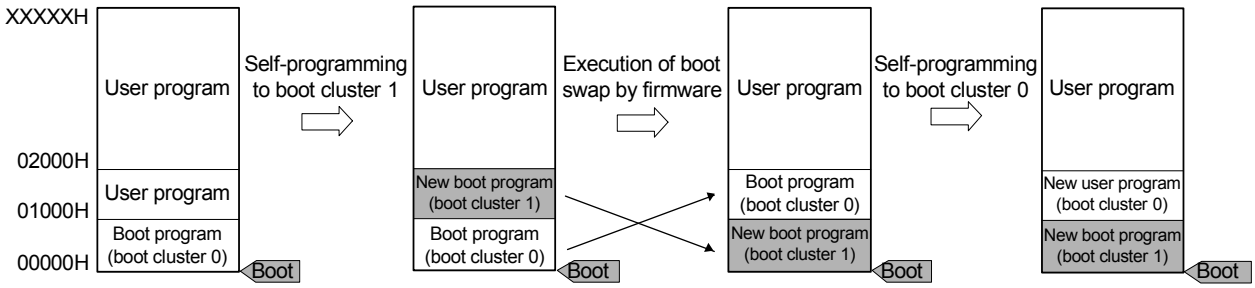
If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

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 the RL78 microcontroller, 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 27 - 7 Boot Swap Function

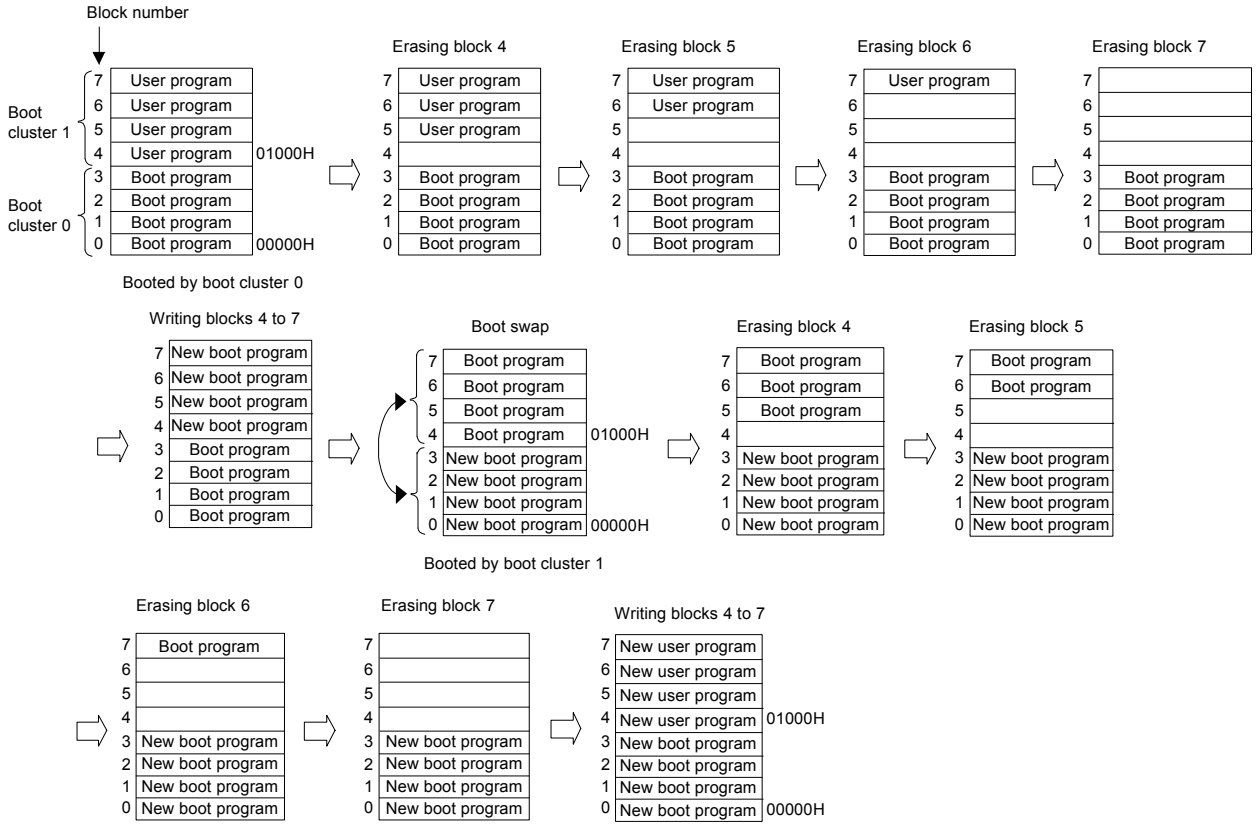


In an example of above figure, it is as follows.

Boot cluster 0: Boot area before boot swap

Boot cluster 1: Boot area after boot swap

Figure 27 - 8 Example of Executing Boot Swapping



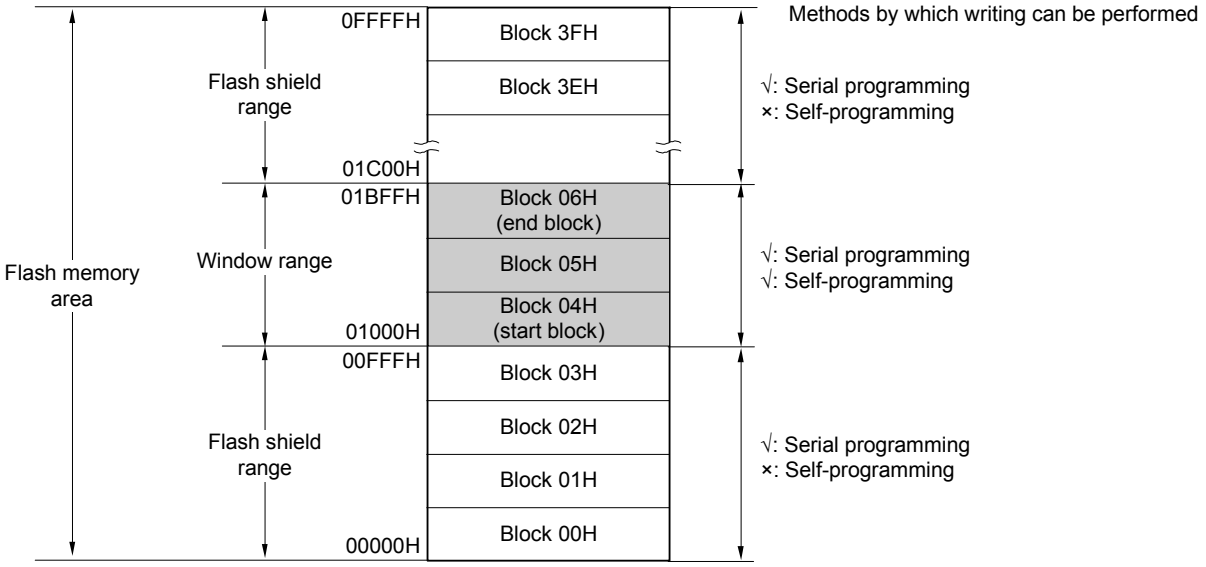
27.5.3 Flash shield window function

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 27 - 9 Flash Shield Window Setting Example
(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 27 - 9 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 **27.6 Security Settings** to prohibit writing/erasing during serial programming.

27.6 Security Settings

The RL78 microcontroller 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 27 - 10 shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.

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 **27.5.3** for detail).

Table 27 - 10 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 **27.5.3** for detail).

Table 27 - 11 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.

27.7 Data Flash

27.7.1 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 Data Flash 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 kept stopped, it should be operated (HIOSTOP = 0). The flash self-programming library should be executed after 30 μs have elapsed.

Remark Refer to flash programming mode, see **27.5 Self-Programming**.

27.7.2 Register controlling data flash memory

27.7.2.1 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 27 - 10 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.

27.7.3 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.
The time setup takes differs for each main clock mode.
<Setup time for each main clock mode>
• HS (High-speed main): 5 μs
• LS (Low-speed main): 720 ns
<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, it should be operated (HIOSTOP = 0). The flash self-programming library should be executed after 30 μs have elapsed.

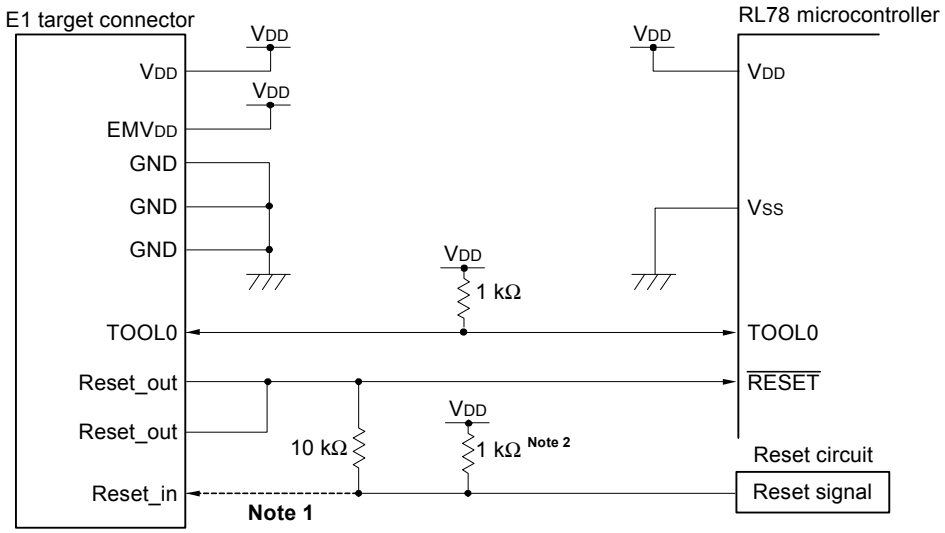
CHAPTER 28 ON-CHIP DEBUG FUNCTION

28.1 Connecting E1 On-chip Debugging Emulator

The RL78 microcontroller uses the VDD, RESET, 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 The RL78 microcontroller 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 28 - 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)

28.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 26 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 28 - 1 On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

28.3 Securing of User Resources

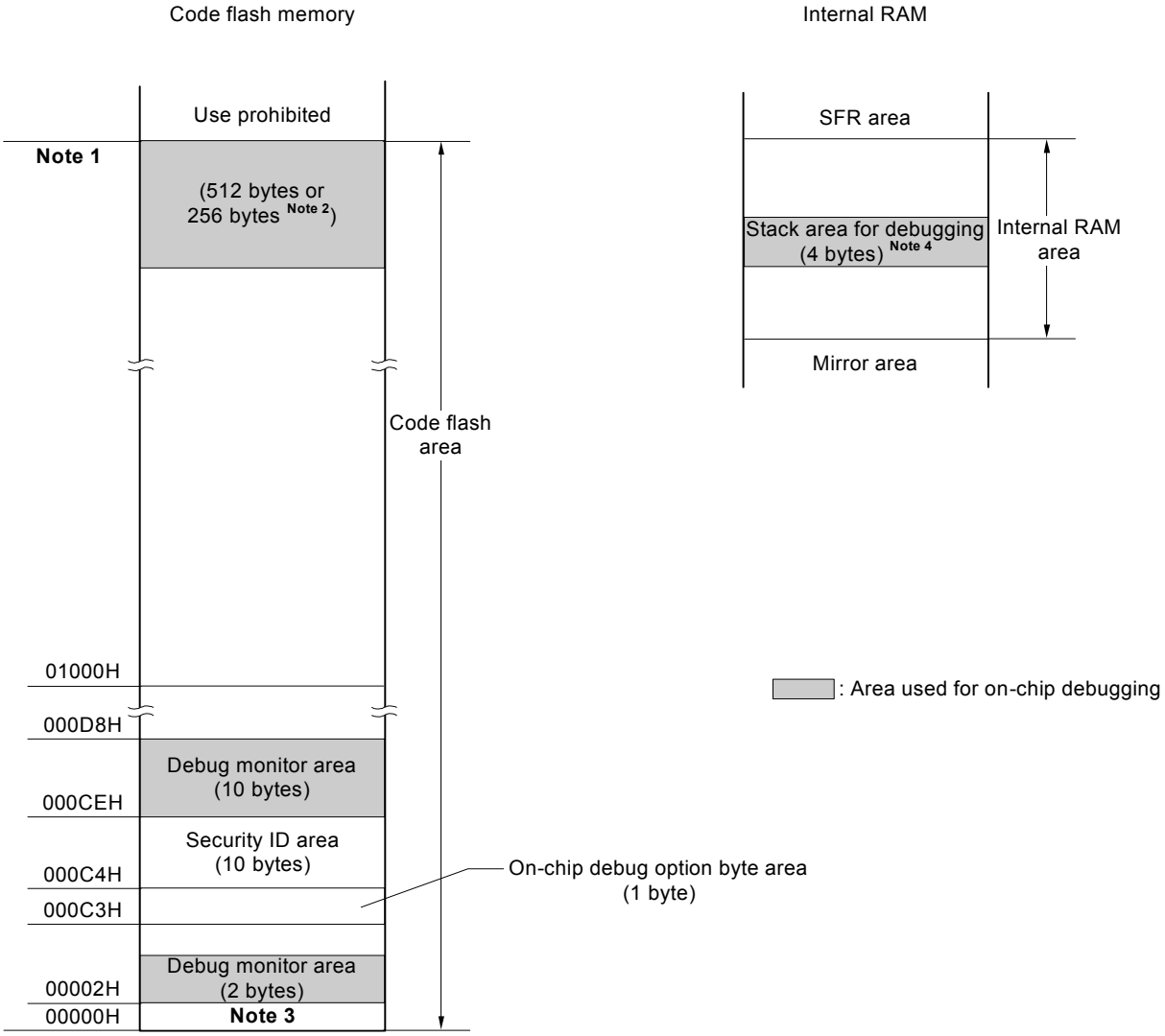
To perform communication between the RL78 microcontroller and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

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 28 - 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 28 - 2 Memory Spaces Where Debug Monitor Programs Are Allocated



Note 1. Address differs depending on products as follows.

Products (code flash memory capacity)	Address of **Note 1.**
R5F11FLJ	3FFFFH
R5F11FLK	5FFFFH
R5F11FLL	7FFFFH

Note 2. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

Note 3. In debugging, reset vector is rewritten to address allocated to a monitor program.

Note 4. 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 29 BCD CORRECTION CIRCUIT

29.1 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).

29.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

29.2.1 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 29 - 1 Format of BCD correction result register (BCDADJ)

Address: F00FEH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

29.3 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 30 INSTRUCTION SET

This chapter lists the instructions in the RL78 microcontroller instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family User's Manual Software (R01US0015)**.

30.1 Conventions Used in Operation List

30.1.1 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 30 - 1 Operand Identifiers and Specification Methods

Identifier	Description Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol (SFR symbol) FFF00H to FFFFFH Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only Note) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FF1FH Immediate data or labels (even addresses only Note)
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions Note) 0080H to 00BFH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 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 **Tables 4 - 5** to **4 - 9 SFR List** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Tables 4 - 10** to **4 - 16 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

30.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 30 - 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)

30.1.3 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 30 - 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

30.1.4 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 and DTC transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 30 - 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.

30.2 Operation List

Table 30 - 5 Operation List (1/18)

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			

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

Remark 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.

Table 30 - 6 Operation List (2/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	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)			
		[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			

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.

Remark 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.

Table 30 - 7 Operation List (3/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	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)			
		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)			

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

Remark 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.

Table 30 - 8 Operation List (4/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	XCH	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 3	1	1	—	AX ← rp			
		rp, AX **Note 3**	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			

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 rp = AX

Remark 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.

Table 30 - 9 Operation List (5/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	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			
		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			

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.

Remark 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.

Table 30 - 10 Operation List (6/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	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 3	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 4	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, 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)	×	×	×

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 rp = AX

Note 4. Except r = A

Remark 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.

Table 30 - 11 Operation List (7/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	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	×	×	×
		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)	×	×	×

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

Remark 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.

Table 30 - 12 Operation List (8/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	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	×	×	×
		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)	×		

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

Remark 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.

Table 30 - 13 Operation List (9/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	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)	×		
		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)	×		

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

Remark 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.

Table 30 - 14 Operation List (10/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	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)	×	×	×

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

Remark 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.

Table 30 - 15 Operation List (11/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
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)	×	×	×
	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)	×	×	×

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.

Remark 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.

Table 30 - 16 Operation List (12/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
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)		×	×

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.

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).

Table 30 - 17 Operation List (13/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
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			
	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			×

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.

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. cnt indicates the bit shift count.

Table 30 - 18 Operation List (14/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	—	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1			×
	ROL	A, 1	2	1	—	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1			×
	RORC	A, 1	2	1	—	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1			×
	ROLC	A, 1	2	1	—	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1			×
	ROLWC	AX,1	2	1	—	(CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) × 1			×
		BC,1	2	1	—	(CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) × 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			×

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.

Remark 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.

Table 30 - 19 Operation List (15/18)

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 ⩛ 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			×
	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 ← $\overline{CY}$			×

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.

Remark 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.

Table 30 - 20 Operation List (16/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	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
	RETB	—	2	6	—	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

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.

Remark 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.

Table 30 - 21 Operation List (17/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	—	(SP - 1) ← PSW, (SP - 2) ← 00H, SP ← SP - 2			
		rp	1	1	—	(SP - 1) ← rp_H, (SP - 2) ← rp_L, SP ← SP - 2			
	POP	PSW	2	3	—	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	—	rp_L ← (SP), rp_H ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	—	SP ← word			
		SP, AX	2	1	—	SP ← AX			
		AX, SP	2	1	—	AX ← SP			
		HL, SP	3	1	—	HL ← SP			
		BC, SP	3	1	—	BC ← SP			
		DE, SP	3	1	—	DE ← SP			
	ADDW	SP, #byte	2	1	—	SP ← SP + byte			
	SUBW	SP, #byte	2	1	—	SP ← SP - byte			
Unconditional branch	BR	AX	2	3	—	PC ← CS, AX			
		$addr20	2	3	—	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	—	PC ← PC + 3 + jdisp16			
		!addr16	3	3	—	PC ← 0000, addr16			
		!!addr20	4	3	—	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4 **Note 3**	—	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4 **Note 3**	—	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4 **Note 3**	—	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4 **Note 3**	—	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4 **Note 3**	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 0			
	BNH	$addr20	3	2/4 **Note 3**	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 1			
	BT	saddr.bit, $addr20	4	3/5 **Note 3**	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5 **Note 3**	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5 **Note 3**	—	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5 **Note 3**	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5 **Note 3**	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6 **Note 3**	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

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. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark 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.

Table 30 - 22 Operation List (18/18)

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 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5 Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6 Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6 Note 3	—	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	RBn Note 4	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. This indicates the number of clocks “when condition is not met/when condition is met”.

Note 4. n indicates the number of register banks (n = 0 to 3)

Remark 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.

<R>

CHAPTER 31 ELECTRICAL SPECIFICATIONS

This chapter describes the electrical specifications for the following products

The target products A: Consumer applications (T_A = –40 to +85 °C)
R5F11FLxANA
D: Industrial applications (T_A = –40 to +85 °C)
R5F11FLxDNA

Caution The RL78 microcontrollers have 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.

31.1 Absolute Maximum Ratings

Absolute Maximum Ratings **(1/2)**

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}	V_{DD}	–0.5 to +3.8 **Note 1**	V
	V_{DDRF}	VCCRF, VCCDDC	–0.3 to +3.8 **Note 1**	V
	V_{SS}	V_{SS}	–0.3 to +0.3	V
	V_{SSRF}	VSSDDC, AGNDRF1, AGNDRF2, RFIN, DIEGND	–0.3 to +0.3	V
Input voltage	V_{I1}	P02 to P04, P20 to P22, P31, P40, P70 to P72, P75 to P77, P80 to P82, P120 to P124, P137, P140 to P144, P155, P156, EXCLK, EXCLKS, $\overline{\text{RESET}}$	–0.3 to V_{DD} + 0.3 **Note 2**	V
	V_{I2}	P100	–0.3 to V_{DD} + 0.3 and –0.3 to V_{DDRF} + 0.3 **Note 2**	V
	V_{I3}	P60 to P63 (N-ch open-drain)	–0.3 to +6.5	V
	V_{IRF1}	STANDBY, GPIO0 to GPIO4, MODE1, MODE2	–0.3 to V_{DDRF} + 0.3 **Note 2**	V
	V_{IRF2}	XIN, REFCLKIN	–0.3 to +1.25	V
	V_{IRF3}	RFIP	–2.0 to +2.0 **Note 3**	V
Output voltage	V_{O1}	P02 to P04, P20 to P22, P31, P40, P60 to P63, P70 to P72, P75 to P77, P80 to P82, P100, P120, P130, P140 to P144, P155, P156	–0.3 to V_{DD} + 0.3 **Note 2**	V
	V_{O2}	P100, INTOUT	–0.3 to V_{DD} + 0.3 and –0.3 to V_{DDRF} + 0.3 **Note 2**	V
	V_{O3}	P60 to P63 (N-ch open-drain)	–0.3 to +6.5	V
	V_{ORF1}	GPIO0 to GPIO4	–0.3 to V_{DDRF} + 0.3 **Note 2**	V
	V_{ORF2}	XOUT	–0.3 to +1.25	V
	V_{ORF3}	RFOUT	–2.0 to +2.0 **Note 3**	V
Analog input voltage	V_{AI}	ANI0 to ANI2, ANI13, ANI14, ANI19	–0.3 to V_{DD} + 0.3 and –0.3 to $AV_{REF}(+)$ + 0.3 **Notes 2, 4**	V

Note 1. Satisfy the relationship of $V_{DD} \geq$ VCCRF and VCCDDC upon power application.

Note 2. Must be 3.8 V or lower.

Note 3. This value is AC rating. Impression of DC voltage is prohibited to RFIP and RFOUT pins.

Note 4. Do not exceed AV_{REF} (+) + 0.3 V in case of A/D conversion target pin.

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Remark 2. AV_{REF} (+): + side reference voltage of the A/D converter.

Remark 3. V_{SS}, V_{SSRF}: Reference voltage

Absolute Maximum Ratings **(2/2)**

Parameter	Symbols	Conditions		Ratings	Unit
REGC pin input voltage	V_{IREGC}	REGC		–0.3 to +2.8 and –0.3 to V_{DD} + 0.3 **Note**	V
RF power supply input	V_{REGIN}	REGIN		–0.3 to +3.8	V
RF power supply output	V_{RFOUT1}	DDCOUT		–0.3 to +3.8	V
	V_{RFOUT2}	VREGO1, VREGO2, VREGO3		–0.3 to +1.25	V
Output current, high	I_{OH1}	Per pin	P02 to P04, P31, P40, P70 to P72, P75 to P77, P80 to P82, P100, P120, P130, P140 to P144	–40	mA
		Total of all pins –170 mA	P02 to P04, P40, P120, P130, P140 to P144	–70	mA
			P31, P70 to P72, P75 to P77, P80 to P82, P100	–100	mA
	I_{OH2}	Per pin	P20 to P22, P155, P156	–0.5	mA
		Total of all pins		–2	mA
	I_{OHRF}	Per pin	GPIO0, GPIO1, GPIO2, GPIO3, GPIO4	–17	mA
Output current, low	I_{OL1}	Per pin	P02 to P04, P31, P40, P60 to P63, P70 to P72, P75 to P77, P80 to P82, P100, P120, P130, P140 to P144	40	mA
		Total of all pins 170 mA	P02 to P04, P40, P120, P130, P140 to P144	70	mA
			P31, P60 to P63, P70 to P72, P75 to P77, P80 to P82, P100	100	mA
	I_{OL2}	Per pin	P20 to P22, P155, P156	1	mA
		Total of all pins		5	mA
	I_{OLRF}	Per pin	GPIO0, GPIO1, GPIO2, GPIO3, GPIO4	17	mA
Operating ambient temperature	T_A	In normal operation mode		–40 to +85	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			–65 to +150	°C

Note Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

31.2 Oscillator Characteristics

31.2.1 X1, XT1 characteristics

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Resonator	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X) **Note**	Ceramic resonator/ crystal resonator	2.7 V ≤ V_{DD} ≤ 3.6 V	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		16.0	
		1.8 V ≤ V_{DD} < 2.4 V	1.0		8.0	
XT1 clock oscillation frequency (f_{XT}) **Note**	Crystal resonator		32	32.768	35	kHz

Note Indicates only permissible oscillator frequency ranges. Refer to **AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator and XT1 oscillator, refer to **6.4 System Clock Oscillator**.

31.2.2 On-chip oscillator characteristics

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency **Notes 1, 2**	f_{IH}			1		32	MHz
High-speed on-chip oscillator clock frequency accuracy		–20 to +85 °C	1.8 V ≤ V_{DD} ≤ 3.6 V	–1.0		+1.0	%
		–40 to –20 °C	1.8 V ≤ V_{DD} ≤ 3.6 V	–1.5		+1.5	%
Low-speed on-chip oscillator clock frequency	f_{IL}				15		kHz
Low-speed on-chip oscillator clock frequency accuracy				–15		+15	%

Note 1. High-speed on-chip oscillator frequency is selected with bits 0 to 4 of the option byte (000C2H/010C2H) and bits 0 to 2 of the HOCODIV register.

Note 2. This only indicates the oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

31.3 DC Characteristics

31.3.1 Pin characteristics

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(1/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P02 to P04, P31, P40, P70 to P72, P75 to P77, P80 to P82, P100, P120, P130, P140 to P144				–10.0 Note 2	mA
		Total of P02 to P04, P40, P120, P130, P140 to P144 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			–10.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			–5.0	mA
		Total of P31, P70 to P72, P75 to P77, P80 to P82, P100 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			–19.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			–10.0	mA
		Total of all pins (When duty ≤ 70% Note 3)				–29.0 Note 4	mA
	I_{OH2}	Per pin for P20 to P22, P155, P156				–0.1 Note 2	mA
		Total of all pins (When duty ≤ 70% Note 3)				–0.5	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

Note 2. Do not exceed the total current value.

Note 3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = –10.0 mA
Total output current of pins = (–10.0 × 0.7)/(80 × 0.01) ≈ –8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor.
A current higher than the absolute maximum rating must not flow into one pin.

Note 4. –100 mA for industrial applications (R5F11FxxDxx).

Caution **P02 to P04, P71, P80 to P82, and P142 to P144 do not output high level in N-ch open-drain mode.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (2/5)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low **Note 1**	I_{OL1}	Per pin for P02 to P04, P31, P40, P70 to P72, P75 to P77, P80 to P82, P100, P120, P130, P140 to P144				20.0 **Note 2**	mA
		Per pin for P60 to P63				15.0 **Note 2**	mA
		Total of P02 to P04, P40, P120, P130, P140 to P144 (When duty ≤ 70% **Note 3**)	2.7 V ≤ V_{DD} ≤ 3.6 V			15.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			9.0	mA
		Total of P31, P60 to P63, P70 to P72, P75 to P77, P80 to P82, P100 (When duty ≤ 70% **Note 3**)	2.7 V ≤ V_{DD} ≤ 3.6 V			35.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			20.0	mA
		Total of all pins (When duty ≤ 70% **Note 3**)				50.0	mA
	I_{OL2}	Per pin for P20 to P22, P155, P156				0.4 **Note 2**	mA
		Total of all pins (When duty ≤ 70% **Note 3**)				2.0	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

Note 2. Do not exceed the total current value.

Note 3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 10.0 mA
Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor.
A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(3/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P02 to P04, P31, P40, P70 to P72, P75 to P77, P80 to P82, P100, P120, P140 to P144	Normal input buffer	0.8 V_{DD}		V_{DD}	V
	V_{IH2}	P03, P04, P80, P81, P142, P143	TTL input buffer 3.3 V ≤ V_{DD} ≤ 3.6 V	2.0		V_{DD}	V
			TTL input buffer 1.8 V ≤ V_{DD} < 3.3 V	1.5		V_{DD}	V
	V_{IH3}	P20 to P22, P155, P156		0.7 V_{DD}		V_{DD}	V
	V_{IH4}	P60 to P63		0.7 V_{DD}		6.0	V
	V_{IH5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P02 to P04, P31, P40, P70 to P72, P75 to P77, P80 to P82, P100, P120, P140 to P144	Normal input buffer	0		0.2 V_{DD}	V
	V_{IL2}	P03, P04, P80, P81, P142, P143	TTL input buffer 3.3 V ≤ V_{DD} ≤ 3.6 V	0		0.5	V
			TTL input buffer 1.8 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P20 to P22, P155, P156		0		0.3 V_{DD}	V
	V_{IL4}	P60 to P63		0		0.3 V_{DD}	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0		0.2 V_{DD}	V

Caution The maximum value of V_{IH} of pins P02 to P04, P71, P80 to P82, and P142 to P144 is V_{DD}, even in the N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(4/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P02 to P04, P31, P40, P70 to P72, P75 to P77, P80 to P82, P100, P120, P130, P140 to P144	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OH1} = –2.0 mA	V_{DD} – 0.6			V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OH1} = –1.5 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P22, P155, P156	1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OH2} = –100 μA	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P02 to P04, P31, P40, P70 to P72, P75 to P77, P80 to P82, P100, P120, P130, P140 to P144	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 3.0 mA			0.6	V
			2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 1.5 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL1} = 0.6 mA			0.4	V
	V_{OL2}	P20 to P22, P155, P156	1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL2} = 400 μA			0.4	V
	V_{OL3}	P60 to P63	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL3} = 3.0 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 3.6 V, I_{OL3} = 2.0 mA			0.4	V

Caution P02 to P04, P71, P80 to P82, and P142 to P144 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(5/5)**

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P02 to P04, P31, P40, P70 to P72, P75 to P77, P80 to P82, P100, P120, P140 to P144	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P20 to P22, P137, P155, P156, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, EXCLK, XT1, XT2, EXCLKS)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P02 to P04, P31, P40, P70 to P72, P75 to P77, P80 to P82, P100, P120, P140 to P144	$V_I = V_{SS}$				–1	μA
	I_{LIL2}	P20 to P22, P137, P155, P156, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL3}	P121 to P124 (X1, X2, EXCLK, XT1, XT2, EXCLKS)	$V_I = V_{SS}$	In input port or external clock input			–1	μA
				In resonator connection			–10	μA
On-chip pull-up resistance	R_U	P02 to P04, P31, P40, P70 to P72, P75 to P77, P80 to P82, P100, P120, P140 to P144	$V_I = V_{SS}$, In input port		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

31.3.2 Supply current characteristics

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	HS (high-speed main) mode **Note 5**	f_{IH} = 32 MHz **Note 3**	Basic operation	V_{DD} = 3.0 V		2.5		mA
			HS (high-speed main) mode **Note 5**	f_{IH} = 32 MHz **Note 3**	Normal operation	V_{DD} = 3.0 V		5.5	10.6	mA
				f_{IH} = 24 MHz **Note 3**	Normal operation	V_{DD} = 3.0 V		4.4	8.2	
				f_{IH} = 16 MHz **Note 3**	Normal operation	V_{DD} = 3.0 V		3.3	5.9	
			LS (low-speed main) mode **Note 5**	f_{IH} = 8 MHz **Note 3**	Normal operation	V_{DD} = 3.0 V		1.5	2.5	mA
						V_{DD} = 2.0 V		1.5	2.5	
			HS (high-speed main) mode **Note 5**	f_{MX} = 20 MHz **Note 2**, V_{DD} = 3.0 V	Normal operation	Square wave input		3.7	6.8	mA
						Resonator connection		3.9	7.0	
				f_{MX} = 8 MHz **Note 2**, V_{DD} = 3.0 V	Normal operation	Square wave input		2.0	4.1	
						Resonator connection		2.0	4.2	
			LS (low-speed main) mode **Note 5**	f_{MX} = 8 MHz **Note 2**, V_{DD} = 3.0 V	Normal operation	Square wave input		1.4	2.4	mA
						Resonator connection		1.4	2.5	
				f_{MX} = 8 MHz **Note 2**, V_{DD} = 2.0 V	Normal operation	Square wave input		1.4	2.4	
						Resonator connection		1.4	2.5	
			Subsystem clock operation	f_{SUB} = 32.768 kHz **Note 4** T_A = –40 °C	Normal operation	Square wave input		5.2		µA
						Resonator connection		5.2		
				f_{SUB} = 32.768 kHz **Note 4** T_A = +25 °C	Normal operation	Square wave input		5.3	7.7	
						Resonator connection		5.3	7.7	
				f_{SUB} = 32.768 kHz **Note 4** T_A = +50 °C	Normal operation	Square wave input		5.5	10.6	
						Resonator connection		5.5	10.6	
				f_{SUB} = 32.768 kHz **Note 4** T_A = +70 °C	Normal operation	Square wave input		5.9	13.2	
						Resonator connection		6.0	13.2	
				f_{SUB} = 32.768 kHz **Note 4** T_A = +85 °C	Normal operation	Square wave input		6.8	17.5	
						Resonator connection		6.9	17.5	

(Notes and Remarks are listed on the next page.)

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the RF transceiver, A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

Note 2. When high-speed on-chip oscillator and subsystem clock are stopped.

Note 3. When high-speed system clock and subsystem clock are stopped.

Note 4. When high-speed on-chip oscillator and high-speed system clock are stopped. When AMPHS1 = 1 (Ultra-low power consumption oscillation). However, not including the current flowing into the RTC, 12-bit interval timer, and watchdog timer.

Note 5. 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} ≤ 3.6 V@1 MHz to 32 MHz
2.4 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 16 MHz

LS (low-speed main) mode: 1.8 V ≤ V_{DD} ≤ 3.6 V@1 MHz to 8 MHz

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{IH}: High-speed on-chip oscillator clock frequency (32 MHz max.)

Remark 3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 4. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current **Note 1**	I_{DD2} **Note 2**	HALT mode	HS (high-speed main) mode **Note 7**	f_{IH} = 32 MHz **Note 4**	V_{DD} = 3.0 V		0.5	2.63	mA
				f_{IH} = 24 MHz **Note 4**	V_{DD} = 3.0 V		0.42	2.03	
				f_{IH} = 16 MHz **Note 4**	V_{DD} = 3.0 V		0.39	1.50	
			LS (low-speed main) mode **Note 7**	f_{IH} = 8 MHz **Note 4**	V_{DD} = 3.0 V		270	800	μA
					V_{DD} = 2.0 V		270	800	
			HS (high-speed main) mode **Note 7**	f_{MX} = 20 MHz **Note 3**, V_{DD} = 3.0 V	Square wave input		0.31	1.69	mA
					Resonator connection		0.41	1.91	
				f_{MX} = 8 MHz **Note 3**, V_{DD} = 3.0 V	Square wave input		0.16	0.94	
					Resonator connection		0.21	1.02	
			LS (low-speed main) mode **Note 7**	f_{MX} = 8 MHz **Note 3**, V_{DD} = 3.0 V	Square wave input		110	610	μA
					Resonator connection		150	660	
				f_{MX} = 8 MHz **Note 3**, V_{DD} = 2.0 V	Square wave input		110	610	
					Resonator connection		150	660	
			Subsystem clock operation	fSUB = 32.768 kHz **Note 5**, T_A = –40 °C	Square wave input		0.31		μA
					Resonator connection		0.50		
				fSUB = 32.768 kHz **Note 5**, T_A = +25 °C	Square wave input		0.38	0.76	
					Resonator connection		0.57	0.95	
				fSUB = 32.768 kHz **Note 5**, T_A = +50 °C	Square wave input		0.47	3.59	
					Resonator connection		0.70	3.78	
				fSUB = 32.768 kHz **Note 5**, T_A = +70 °C	Square wave input		0.80	6.20	
					Resonator connection		1.00	6.39	
				fSUB = 32.768 kHz **Note 5**, T_A = +85 °C	Square wave input		1.65	10.56	
					Resonator connection		1.84	10.75	
	IDD3 **Note 6**	STOP mode **Note 8**	T_A = –40 °C				0.19		μA
			T_A = +25 °C				0.30	0.59	
			T_A = +50 °C				0.41	3.42	
			T_A = +70 °C				0.80	6.03	
			T_A = +85 °C				1.53	10.39	

(Notes and Remarks are listed on the next page.)

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the RF transceiver, A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

Note 2. During HALT instruction execution by flash memory.

Note 3. When high-speed on-chip oscillator and subsystem clock are stopped.

Note 4. When high-speed system clock and subsystem clock are stopped.

Note 5. When high-speed on-chip oscillator and high-speed system clock are stopped. When RTCLPC = 1 and setting ultra-low current consumption (AMPHS1 = 1). The current flowing into the RTC is included. However, not including the current flowing into the 12-bit interval timer and watchdog timer.

Note 6. Not including the current flowing into the RTC, 12-bit interval timer, and watchdog timer.

Note 7. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: $2.7\ V \le V_{DD} \le 3.6\ V$@1 MHz to 32 MHz

$2.4\ V \le V_{DD} \le 3.6\ V$@1 MHz to 16 MHz

LS (low-speed main) mode: $1.8\ V \le V_{DD} \le 3.6\ V$@1 MHz to 8 MHz

Note 8. Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{IH}: High-speed on-chip oscillator clock frequency (32 MHz max.)

Remark 3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 4. Except subsystem clock operation and STOP mode, temperature condition of the TYP. value is T_A = 25 °C

Peripheral Functions (Common to all products)

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.20		μA
RTC operating current	I_{RTC} Notes 1, 2, 3				0.02		μA
12-bit interval timer operating current	I_{IT} Notes 1, 2, 4				0.02		μA
Watchdog timer operating current	I_{WDT} Notes 1, 2, 5	f_{IL} = 15 kHz			0.22		μA
A/D converter operating current	I_{ADC} Notes 1, 6	When conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 3.3 V		1.3	1.7	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.5	0.7	mA
LVD operating current	I_{LVD} Notes 1, 7				0.08		μA
Self-programming operating current	I_{FSP} Notes 1, 9				2.50	12.20	mA
BGO operating current	I_{BGO} Notes 1, 8				2.50	12.20	mA
SNOOZE operating current	I_{SNOZ} Note 1	ADC operation	The mode is performed Note 10		0.50	0.60	mA
			The A/D conversion operations are performed, Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		1.20	1.44	
		DTC operation			3.10		

Note 1. Current flowing to V_{DD}.

Note 2. When high speed on-chip oscillator and high-speed system clock are stopped.

Note 3. Current flowing only to the real-time clock (RTC) (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added. I_{DD2} subsystem clock operation includes the operational current of the real-time clock.

Note 4. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the 12-bit interval timer operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

Note 5. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2}, or I_{DD3} and I_{WDT} when the watchdog timer is in operation.

Note 6. Current flowing only to the A/D converter. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

Note 7. Current flowing only to the LVD circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2}, or I_{DD3} and I_{LVD} when the LVD circuit is in operation.

Note 8. Current flowing during programming of the data flash.

Note 9. Current flowing during self-programming.

Note 10. For shift time to the SNOOZE mode, see **20.3.3 SNOOZE mode**.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 3. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 4. Temperature condition of the TYP. value is T_A = 25 °C

31.4 AC Characteristics

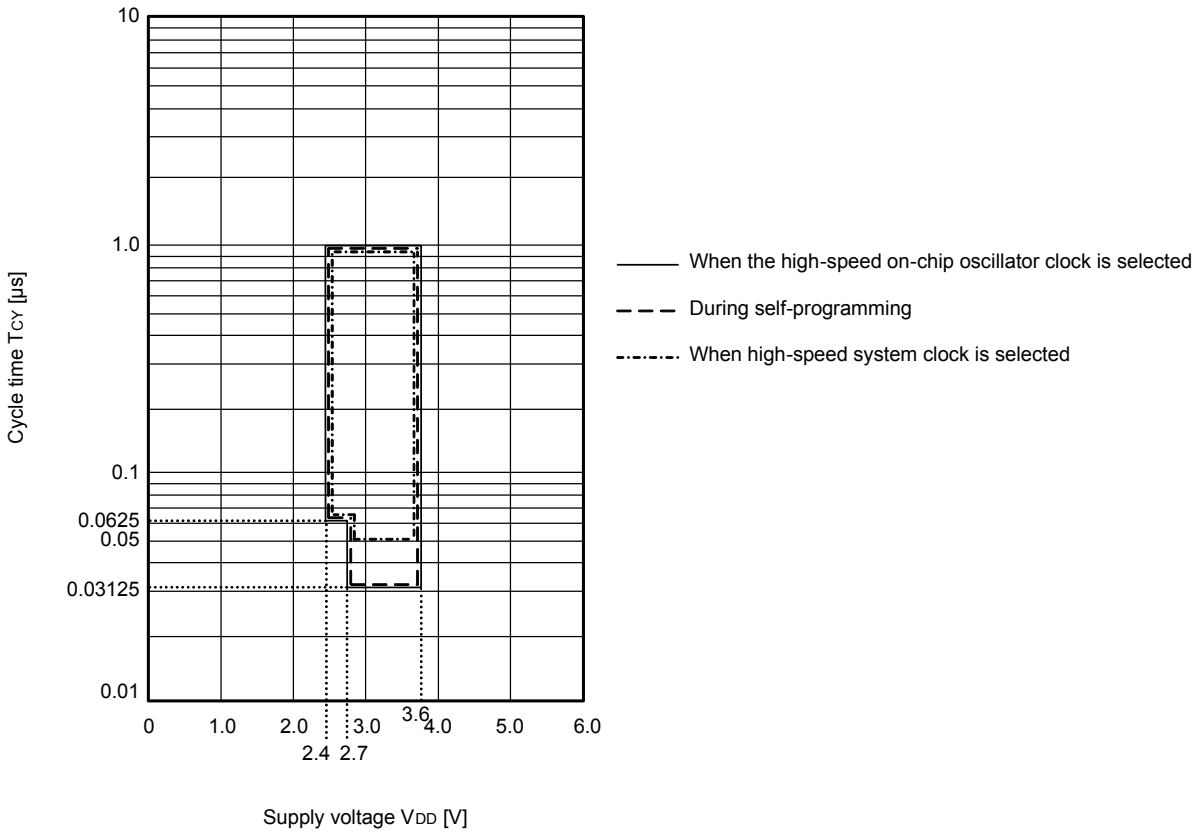
(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V	0.03125		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V	0.125		1	μs
		Subsystem clock (f_{SUB}) operation		1.8 V ≤ V_{DD} ≤ 3.6 V	28.5	30.5	31.3	μs
		In the self-programming mode	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V	0.03125		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V	0.125		1	μs
External system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 3.6 V			1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V			1.0		16.0	MHz
		1.8 V ≤ V_{DD} < 2.4 V			1.0		8.0	MHz
	f_{EXS}				32		35	kHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 3.6 V			24			ns
		2.4 V ≤ V_{DD} < 2.7 V			30			ns
		1.8 V ≤ V_{DD} < 2.4 V			60			ns
	t_{EXHS}, t_{EXLS}				13.7			μs
TI03 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK} + 10			ns
TO03 output frequency	f_{TO}	HS (high-speed main) mode		2.7 V ≤ V_{DD} ≤ 3.6 V			8	MHz
				2.4 V ≤ V_{DD} < 2.7 V			4	MHz
		LS (low-speed main) mode		1.8 V ≤ V_{DD} ≤ 3.6 V			4	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode		2.7 V ≤ V_{DD} ≤ 3.6 V			8	MHz
				2.4 V ≤ V_{DD} < 2.7 V			4	MHz
		LS (low-speed main) mode		1.8 V ≤ V_{DD} ≤ 3.6 V			4	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0, INTP4, INTP6, INTP7, INTP9 to INTP11		1.8 V ≤ V_{DD} ≤ 3.6 V	1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

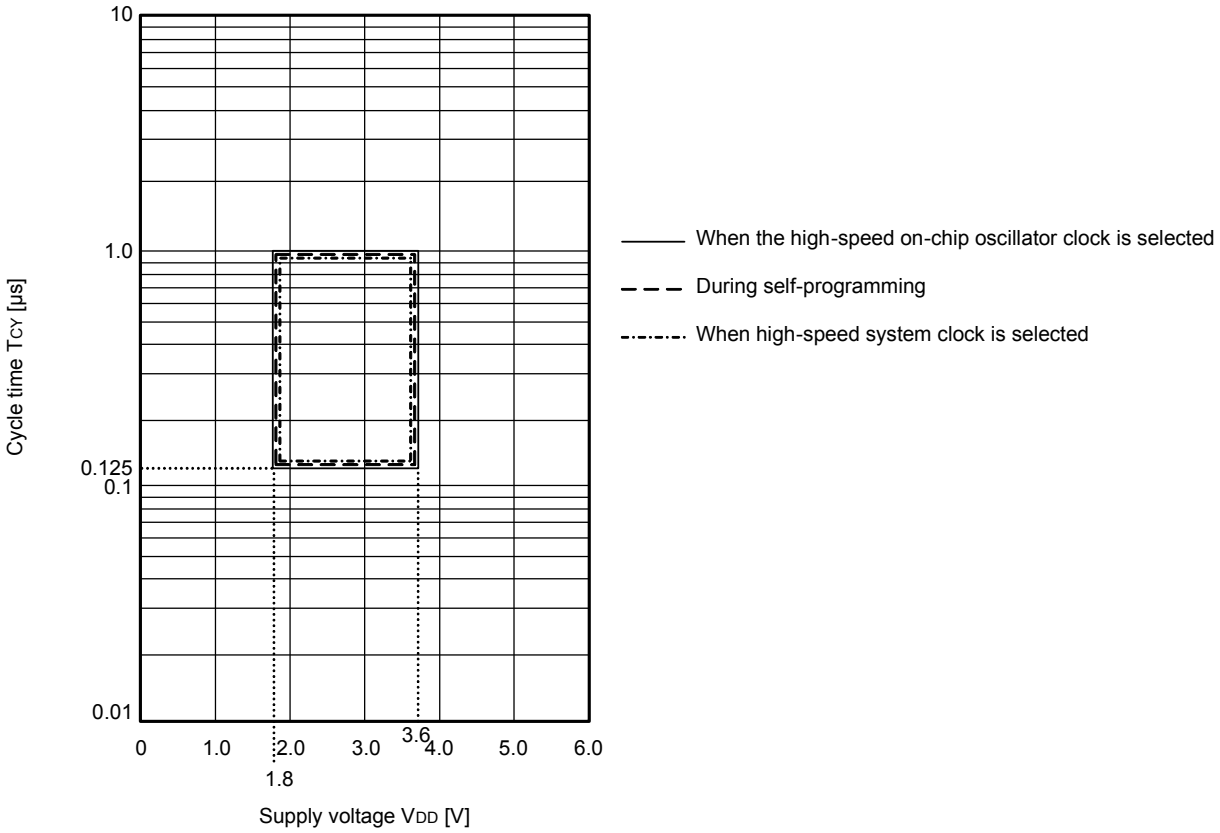
Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of timer mode register mn (TMRmn). m: Unit number (m = 0), n: Channel number (n = 3))

Minimum Instruction Execution Time during Main System Clock Operation

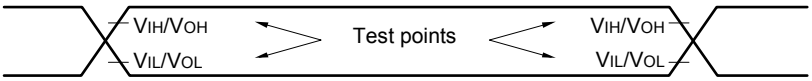
T_{CY} vs V_{DD} (HS (high-speed main) mode)



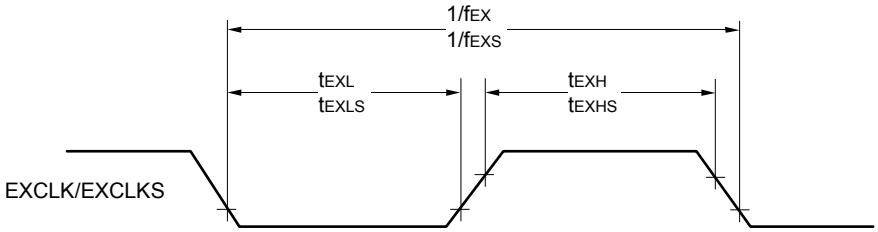
T_{CY} vs V_{DD} (LS (low-speed main) mode)



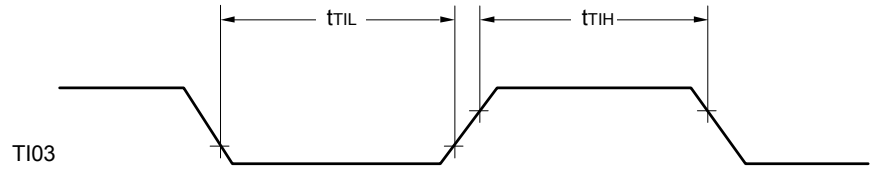
AC Timing Test Points

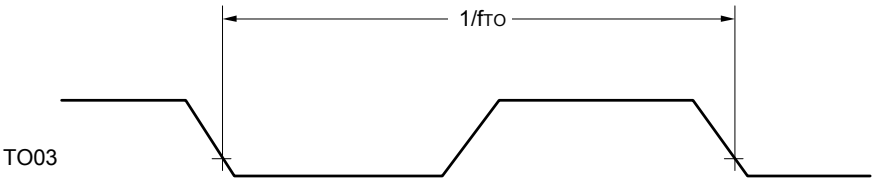


External System Clock Timing

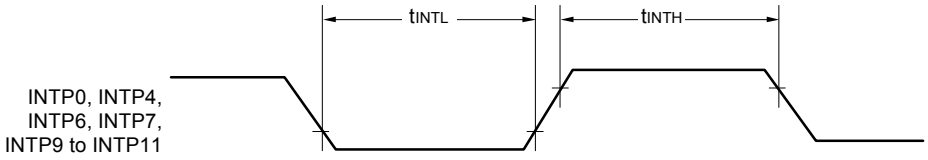


TI/TO Timing

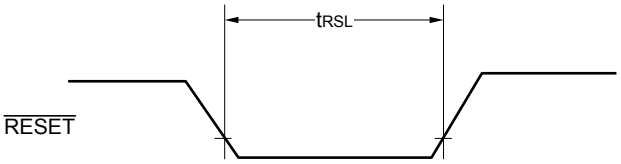




Interrupt Request Input Timing

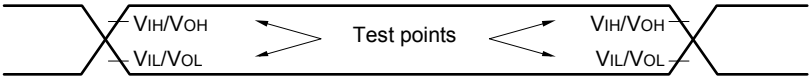


$\overline{\text{RESET}}$ Input Timing



31.5 Peripheral Functions Characteristics

AC Timing Test Points



31.5.1 Serial array unit

(1) During communication at same potential (UART mode)

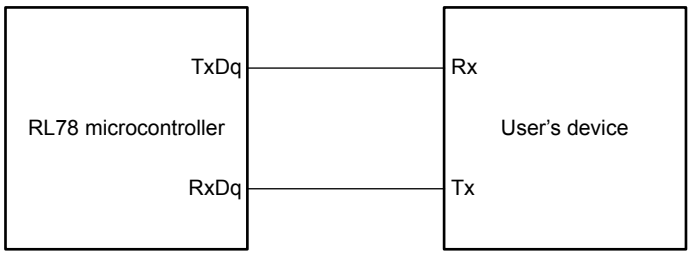
(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
Transfer rate		2.4 V ≤ V_{DD} ≤ 3.6 V		f_{MCK}/6		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note**		5.3		1.3	Mbps
		1.8 V ≤ V_{DD} ≤ 3.6 V		f_{MCK}/6		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note		5.3		1.3	Mbps

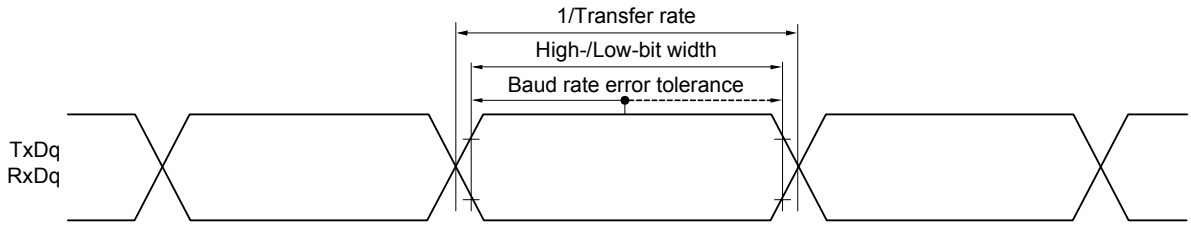
Note The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 32 MHz (2.7 V ≤ V_{DD} ≤ 3.6 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 3.6 V)
LS (low-speed main) mode: 8 MHz (1.8 V ≤ V_{DD} ≤ 3.6 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remark 1. q: UART number (q = 1, 3), g: PIM and POM number (g = 0, 14)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 02, 03, 12, 13))

(2) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	2.7 V ≤ V_{DD} ≤ 3.6 V	125		500		ns
			2.4 V ≤ V_{DD} ≤ 3.6 V	250		500		ns
			1.8 V ≤ V_{DD} ≤ 3.6 V	500		500		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V		t_{KCY1}/2 – 18		t_{KCY1}/2 – 50		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		t_{KCY1}/2 – 38		t_{KCY1}/2 – 50		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
SIp setup time (to SCKp↑) **Note 1**	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V		44		110		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		75		110		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		110		110		ns
SIp hold time (from SCKp↑) **Note 2**	t_{KSI1}	1.8 V ≤ V_{DD} ≤ 3.6 V		19		19		ns
Delay time from SCKp↓ to SOp output **Note 3**	t_{KSO1}	1.8 V ≤ V_{DD} ≤ 3.6 V C = 30 pF **Note 4**			25		25	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark p: CSI number (p = 10, 21, 30), m: Unit number (m = 0, 1), n: Channel number (n = 1, 2),
g: PIM and POM number (g = 0, 14)

(3) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output, supported only for CSI20)

(T_A = -40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	2.4 V ≤ V_{DD} ≤ 3.6 V	250		500		ns
			1.8 V ≤ V_{DD} ≤ 3.6 V	500		500		ns

Remark p: CSI number (p = 20)

(4) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input)

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 3.6 V	16 MHz < f_{MCK}	8/f_{MCK}		—		ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}		6/f_{MCK}		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		6/f_{MCK} and 500		6/f_{MCK} and 500		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		6/f_{MCK} and 750		6/f_{MCK} and 750		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 3.6 V		t_{KCY2}/2 – 8		t_{KCY2}/2 – 8		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		t_{KCY2}/2 – 18		t_{KCY2}/2 – 18		ns
SIp setup time (to SCKp↑) **Note 1**	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 3.6 V		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑) **Note 2**	t_{KSI2}	1.8 V ≤ V_{DD} ≤ 3.6 V		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output **Note 3**	t_{KSO2}	2.7 V ≤ V_{DD} ≤ 3.6 V	C = 30 pF **Note 4**		2/f_{MCK} + 44		2/f_{MCK} + 110	ns
		2.4 V ≤ V_{DD} ≤ 3.6 V			2/f_{MCK} + 75		2/f_{MCK} + 110	ns
		1.8 V ≤ V_{DD} ≤ 3.6 V			2/f_{MCK} + 100		2/f_{MCK} + 110	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

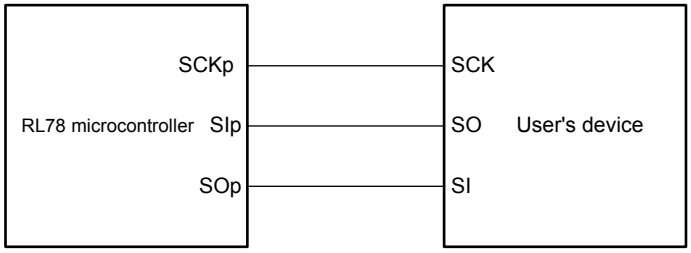
Note 4. C is the load capacitance of the SOp output lines.

Caution Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark 1. p: CSI number (p = 10, 21, 30), m: Unit number (m = 0, 1), n: Channel number (n = 1, 2), g: PIM and POM number (g = 0, 14)

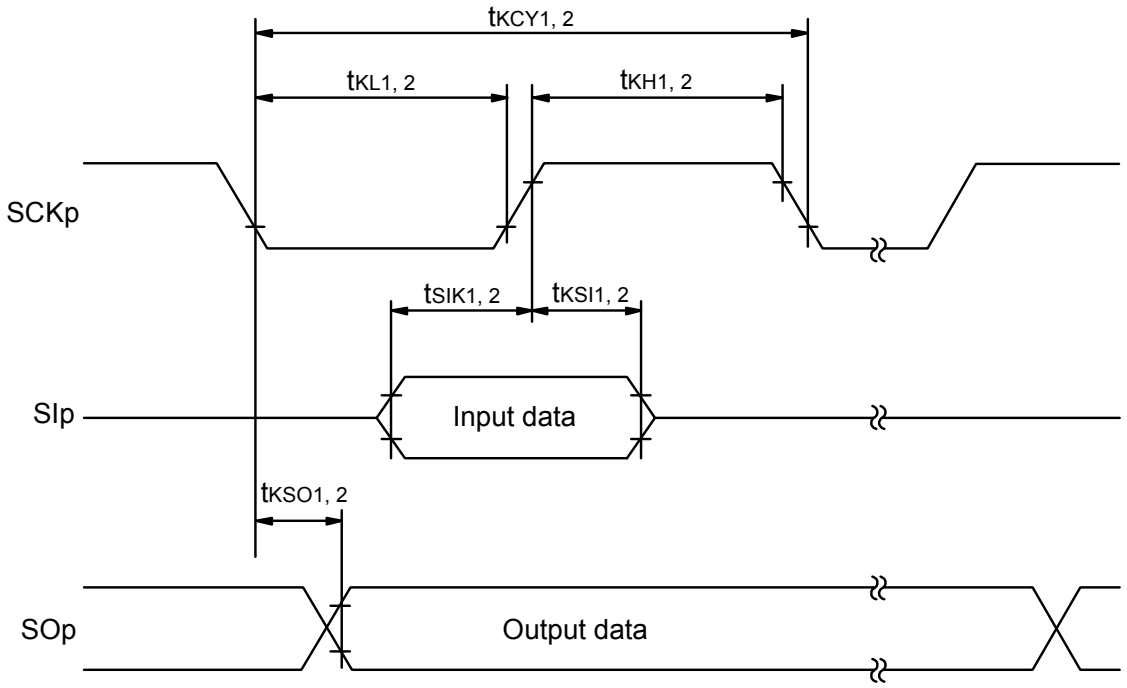
Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 02, 11, 12))

CSI mode connection diagram (during communication at same potential)

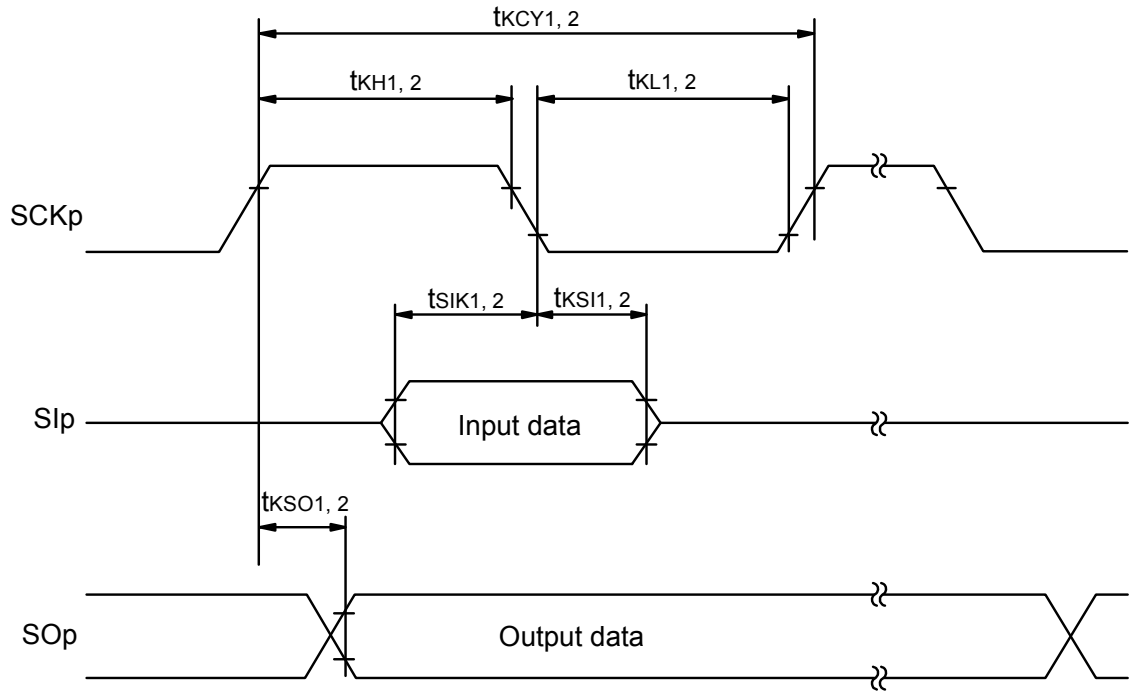


Remark p: CSI number (p = 10, 21, 30)

CSI mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 10, 21, 30)
Remark 2. m: Unit number, n: Channel number (mn = 02, 11, 12)

(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode)

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
Transfer rate		reception	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V		f_{MCK}/6		f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		5.3		1.3	Mbps
			1.8 V ≤ V_{DD} < 3.3 V, 1.8 V ≤ Vb ≤ 2.0 V		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		5.3		1.3	Mbps

Note 1. Use it with V_{DD} ≥ V_b.

Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 32 MHz (2.7 V ≤ V_{DD} ≤ 3.6 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 3.6 V)
LS (low-speed main) mode: 8 MHz (1.8 V ≤ V_{DD} ≤ 3.6 V)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remark 1. V_b [V]: Communication line voltage

Remark 2. q: UART number (q = 1, 3), g: PIM and POM number (g = 0, 14)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 02, 03, 12, 13))

(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode)

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
Transfer rate		transmission	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		**Note 1**		**Note 1**	bps
			Theoretical value of the maximum transfer rate Cb = 50 pF, Rb = 2.7 kΩ, Vb = 2.3 V		1.2 **Note 2**		1.2 **Note 2**	Mbps
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		**Notes 3, 4**		**Notes 3, 4**	bps
			Theoretical value of the maximum transfer rate Cb = 50 pF, Rb = 5.5 kΩ, Vb = 1.6 V		0.43 **Note 5**		0.43 **Note 5**	Mbps

Note 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 3.6 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides .

Note 2. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 1** above to calculate the maximum transfer rate under conditions of the customer.

Note 3. Use it with $V_{DD} \geq V_b$.

Note 4. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 1.8 V ≤ V_{DD} < 3.3 V and 1.8 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

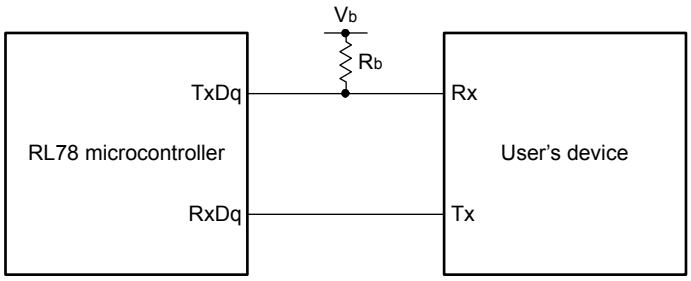
* This value is the theoretical value of the relative difference between the transmission and reception sides .

Note 5. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 4** above to calculate the maximum transfer rate under conditions of the customer.

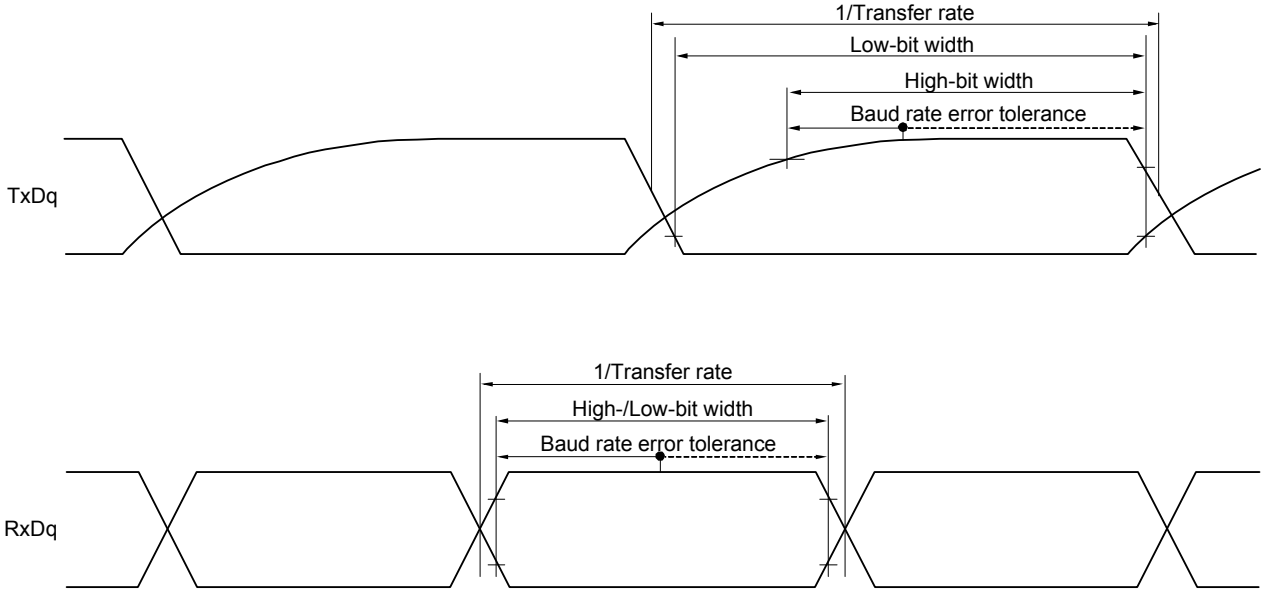
(Caution and Remarks are listed on the next page.)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Remark 1. R_b[Ω]: Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 1, 3), g: PIM and POM number (g = 0, 14)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 02, 03, 12, 13))

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(1/3)**

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	500		1150		ns
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V **Note**, Cb = 30 pF, Rb = 5.5 kΩ	1150		1150		ns
SCKp high-level width	t_{KH1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ		t_{KCY1}/2 – 170		t_{KCY1}/2 – 170		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V **Note**, Cb = 30 pF, Rb = 5.5 kΩ		t_{KCY1}/2 – 458		t_{KCY1}/2 – 458		ns
SCKp low-level width	t_{KL1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ		t_{KCY1}/2 – 18		t_{KCY1}/2 – 50		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V **Note**, Cb = 30 pF, Rb = 5.5 kΩ		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns

Note Use it with $V_{DD} \geq V_b$.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed two pages after the next page.)

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) (2/3)

Parameter	Symbol	Conditions	HS (high-speed main) mode		LS (low-speed main) mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SIp setup time (to SCKp↑) **Note 1**	t_{SIK1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	177		479		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V **Note 2**, Cb = 30 pF, Rb = 5.5 kΩ	479		479		ns
SIp hold time (from SCKp↑) **Note 1**	t_{KSI1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	19		19		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V **Note 2**, Cb = 30 pF, Rb = 5.5 kΩ	19		19		ns
Delay time from SCKp↓ to SOp output **Note 1**	t_{KSO1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ		195		195	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V **Note 2**, Cb = 30 pF, Rb = 5.5 kΩ		483		483	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Note 2. Use it with $V_{DD} \geq V_b$.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the page after the next page.)

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V) **(3/3)**

Parameter	Symbol	Conditions	HS (high-speed main) mode		LS (low-speed main) mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SIp setup time (to SCKp↓) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	44		110		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V Note 2, Cb = 30 pF, Rb = 5.5 kΩ	110		110		ns
SIp hold time (from SCKp↓) Note 1	t_{KSI1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	19		19		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V Note 2, Cb = 30 pF, Rb = 5.5 kΩ	19		19		ns
Delay time from SCKp↑ to SOp output Note 1	t_{KSO1}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ		25		25	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V Note 2, Cb = 30 pF, Rb = 5.5 kΩ		25		25	ns

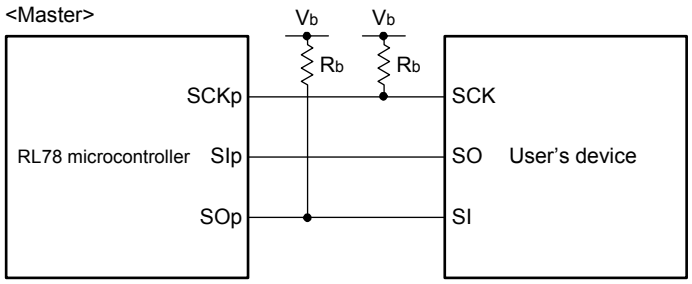
Note 1. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. Use it with V_{DD} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

CSI mode connection diagram (during communication at different potential)

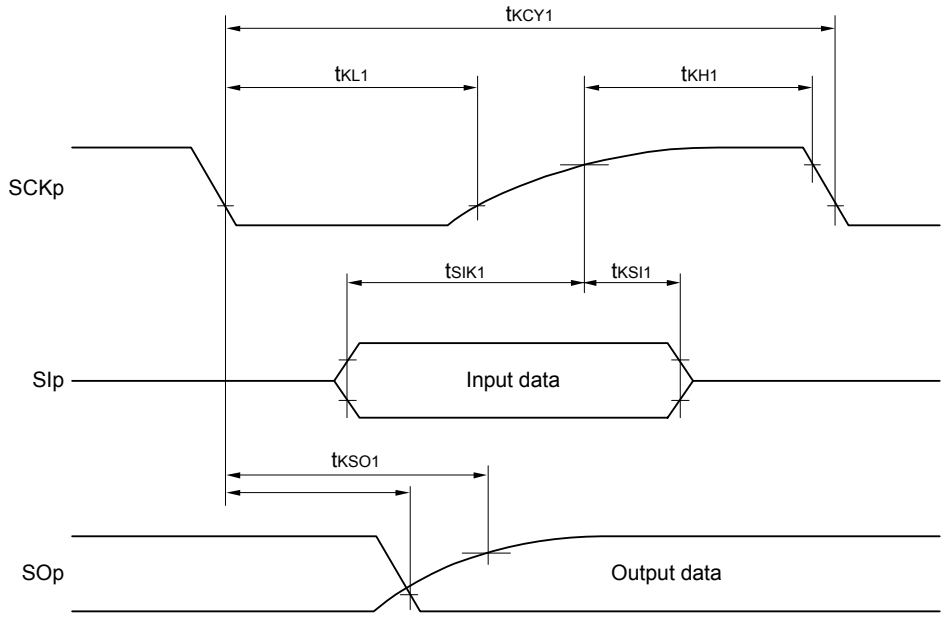


Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

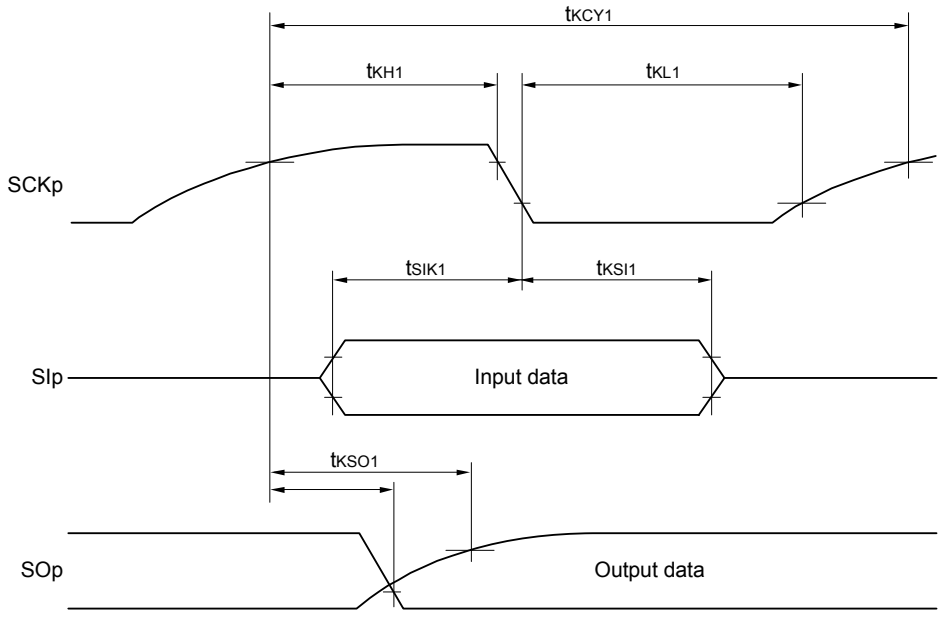
Remark 2. p: CSI number (p = 10, 30), m: Unit number (m = 0, 1), n: Channel number (n = 2), g: PIM and POM number (g = 0, 14)

Remark 3. CSI21 cannot communicate at different potential. Use other CSI for communication at different potential.

CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 10, 30), m: Unit number (m = 0, 1), n: Channel number (n = 2),
g: PIM and POM number (g = 0, 14)

Remark 2. CSI21 cannot communicate at different potential. Use other CSI for communication at different potential.

(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (slave mode, SCKp... external clock input)

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY2}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V	24 MHz < f_{MCK}	20/f_{MCK}		—		ns
			20 MHz < f_{MCK} ≤ 24 MHz	16/f_{MCK}		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	14/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	12/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		10/f_{MCK}		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V **Note 1**	24 MHz < f_{MCK}	48/f_{MCK}		—		ns
			20 MHz < f_{MCK} ≤ 24 MHz	36/f_{MCK}		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	32/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	26/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	10/f_{MCK}		10/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V		t_{KCY2}/2 – 18		t_{KCY2}/2 – 50		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V **Note 1**		t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		ns
SIp setup time (to SCKp↑) **Note 2**	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
		1.8 V ≤ V_{DD} ≤ 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V **Note 1**		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑) **Note 3**	t_{KSI2}			1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output **Note 4**	t_{KSO2}	2.7 V ≤ V_{DD} < 3.6 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ			2/f_{MCK} + 214		2/f_{MCK} + 573	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.8 V ≤ Vb ≤ 2.0 V **Note 1**, Cb = 30 pF, Rv = 5.5 kΩ			2/f_{MCK} + 573		2/f_{MCK} + 573	ns

Note 1. Use it with V_{DD} ≥ Vb.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

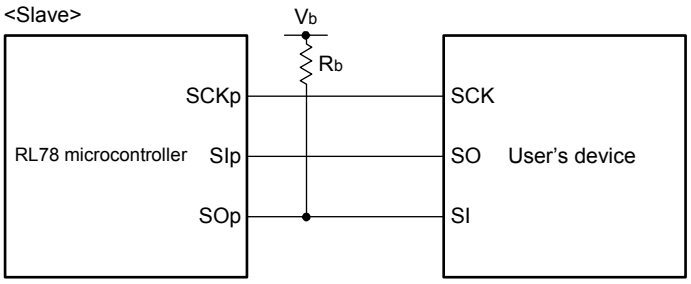
Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the SIp pin and SCKp pin, and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

CSI mode connection diagram (during communication at different potential)



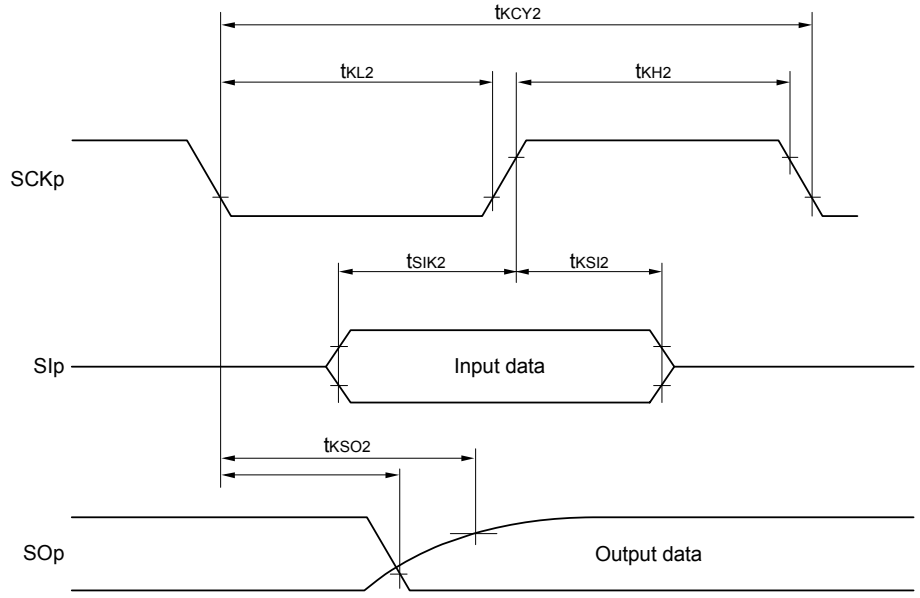
Remark 1. R_b[Ω]: Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance,
V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 10, 30), m: Unit number (m = 0, 1), n: Channel number (n = 2),
g: PIM and POM number (g = 0, 14)

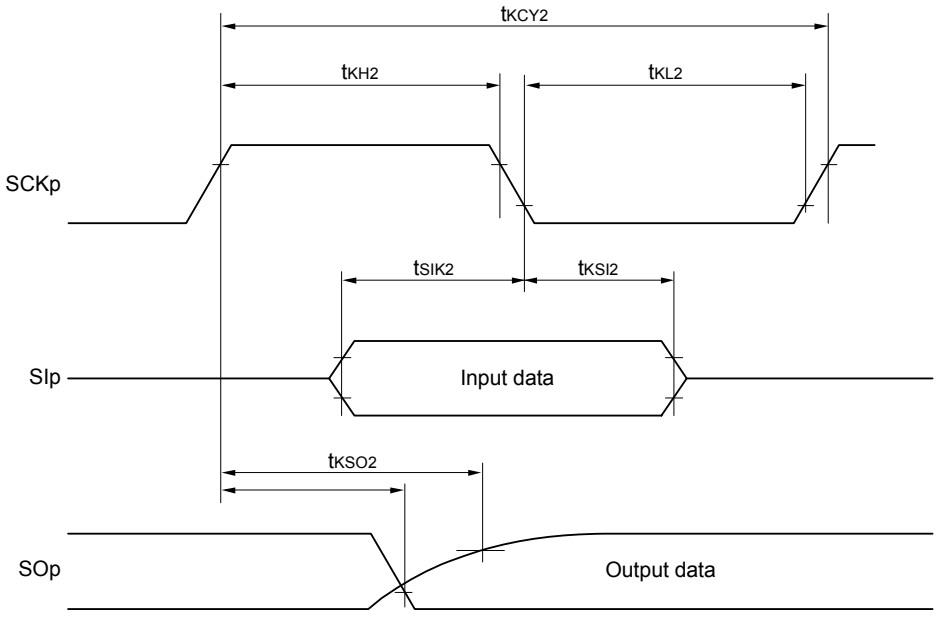
Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 02, 12))

Remark 4. CSI21 cannot communicate at different potential. Use other CSI for communication at different potential.

CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 10, 30), m: Unit number (m = 0, 1), n: Channel number (n = 2),
g: PIM and POM number (g = 0, 14)

Remark 2. CSI21 cannot communicate at different potential. Use other CSI for communication at different potential.

31.5.2 Serial interface IICA

(1) I^2C standard mode

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Standard mode: f_{CLK} ≥ 1 MHz	2.7 V ≤ V_{DD} ≤ 3.6 V	0	100	0	100	kHz
			1.8 V ≤ V_{DD} ≤ 3.6 V	0	100	0	100	kHz
Setup time of restart condition	$t_{SU: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		4.7		4.7		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		4.7		4.7		μs
Hold time Note 1	$t_{HD: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		4.0		4.0		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		4.0		4.0		μs
Hold time when SCLA0 = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V		4.7		4.7		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		4.7		4.7		μs
Hold time when SCLA0 = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V		4.0		4.0		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		4.0		4.0		μs
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		250		250		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		250		250		ns
Data hold time (transmission) Note 2	$t_{HD: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0	3.45	0	3.45	μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0	3.45	0	3.45	μs
Setup time of stop condition	$t_{SU: STO}$	2.7 V ≤ V_{DD} ≤ 3.6 V		4.0		4.0		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		4.0		4.0		μs
Bus-free time	t_{BUF}	2.7 V ≤ V_{DD} ≤ 3.6 V		4.7		4.7		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		4.7		4.7		μs

Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

Note 2. The maximum value (MAX.) of $t_{HD: DAT}$ is during normal transfer and a wait state is inserted in the ACK (acknowledge) timing.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.
Standard mode: C_b = 400 pF, R_b = 2.7 kΩ

(2) I^2C fast mode

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz	2.7 V ≤ V_{DD} ≤ 3.6 V	0	400	0	400	kHz
			1.8 V ≤ V_{DD} ≤ 3.6 V	0	400	0	400	kHz
Setup time of restart condition	$t_{SU: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		μs
Hold time **Note 1**	$t_{HD: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		μs
Hold time when SCLA0 = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V		1.3		1.3		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		1.3		1.3		μs
Hold time when SCLA0 = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		μs
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		100		100		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		100		100		ns
Data hold time (transmission) **Note 2**	$t_{HD: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0	0.9	0	0.9	μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0	0.9	0	0.9	μs
Setup time of stop condition	$t_{SU: STO}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		0.6		0.6		μs
Bus-free time	t_{BUF}	2.7 V ≤ V_{DD} ≤ 3.6 V		1.3		1.3		μs
		1.8 V ≤ V_{DD} ≤ 3.6 V		1.3		1.3		μs

Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

Note 2. The maximum value (MAX.) of $t_{HD: DAT}$ is during normal transfer and a wait state is inserted in the ACK (acknowledge) timing.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.
Fast mode: C_b = 320 pF, R_b = 1.1 kΩ

(3) I²C fast mode plus

(T_A = –40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

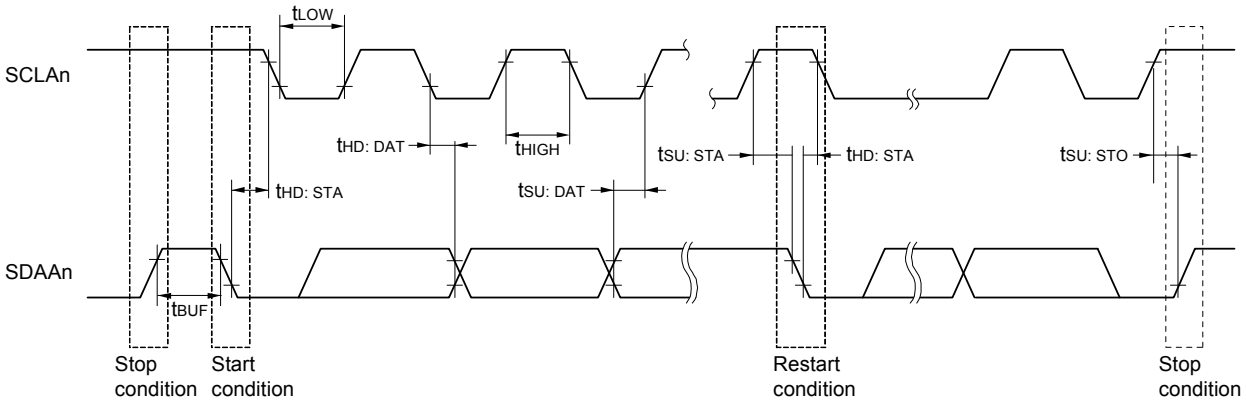
Parameter	Symbol	Conditions		HS (high-speed main) mode		LS (low-speed main) mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode plus: f_{CLK} ≥ 10 MHz	2.7 V ≤ V_{DD} ≤ 3.6 V	0	1000	—		kHz
Setup time of restart condition	$t_{SU: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.26		—		μs
Hold time **Note 1**	$t_{HD: STA}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.26		—		μs
Hold time when SCLA0 = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V		0.5		—		μs
Hold time when SCLA0 = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V		0.26		—		μs
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		50		—		ns
Data hold time (transmission) **Note 2**	$t_{HD: DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0	0.45	—		μs
Setup time of stop condition	$t_{SU: STO}$	2.7 V ≤ V_{DD} ≤ 3.6 V		0.26		—		μs
Bus-free time	t_{BUF}	2.7 V ≤ V_{DD} ≤ 3.6 V		0.5		—		μs

Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

Note 2. The maximum value (MAX.) of $t_{HD: DAT}$ is during normal transfer and a wait state is inserted in the ACK (acknowledge) timing.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.
Fast mode plus: C_b = 120 pF, R_b = 1.1 kΩ

IICA serial transfer timing



Remark n = 0, 1

31.6 Analog Characteristics

31.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Reference Voltage / Input channel	Reference voltage (+) = AVREFP Reference voltage (–) = AVREFM	Reference voltage (+) = VDD Reference voltage (–) = VSS
ANI0 to ANI2, ANI13, ANI14, ANI19	Refer to **31.6.1 (1)**.	Refer to **31.6.1 (2)**.

(1) When reference voltage (+) = AVREFP/ANI0 (ADREFP0 = 1), reference voltage (–) = AVREFM/ANI1 (ADREFM = 1), target pin: ANI0 to ANI2, ANI13, ANI14, ANI19

(TA = –40 to +85 °C, 1.8 V ≤ VDD ≤ 3.6 V, 1.8 V ≤ AVREFP ≤ VDD ≤ 3.6 V, VSS = 0 V, Reference voltage (+) = AVREFP, Reference voltage (–) = AVREFM = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error **Note 1**	AINL	10-bit resolution AVREFP = VDD **Notes 3, 4**	1.8 V ≤ AVREFP ≤ 3.6 V		1.2	±5.0	LSB
Conversion time	tCONV	10-bit resolution Target pin: ANI0 to ANI2, ANI13, ANI14, ANI19	2.7 V ≤ VDD ≤ 3.6 V	3.1875		39	μs
			1.8 V ≤ VDD ≤ 3.6 V	17		39	μs
Zero-scale error **Notes 1, 2**	EZS	10-bit resolution AVREFP = VDD **Notes 3, 4**	1.8 V ≤ AVREFP ≤ 3.6 V			±0.35	%FSR
Full-scale error **Notes 1, 2**	EFS	10-bit resolution AVREFP = VDD **Notes 3, 4**	1.8 V ≤ AVREFP ≤ 3.6 V			±0.35	%FSR
Integral linearity error **Note 1**	ILE	10-bit resolution AVREFP = VDD **Notes 3, 4**	1.8 V ≤ AVREFP ≤ 3.6 V			±3.5	LSB
Differential linearity error **Note 1**	DLE	10-bit resolution AVREFP = VDD **Notes 3, 4**	1.8 V ≤ AVREFP ≤ 3.6 V			±2.0	LSB
Analog input voltage	VAIN	ANI0 to ANI2, ANI13, ANI14, ANI19		0		AVREFP	V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When AVREFP < VDD, the MAX. values are as follows.

Overall error: Add ±1.0 LSB to the MAX. value when AVREFP = VDD.

Zero-scale error/Full-scale error: Add ±0.05%FSR to the MAX. value when AVREFP = VDD.

Integral linearity error/ Differential linearity error: Add ±0.5 LSB to the MAX. value when AVREFP = VDD.

Note 4. When AVREFP < VDD, the MAX. values are as follows.

Overall error: Add ±4.0 LSB to the MAX. value when AVREFP = VDD.

Zero-scale error/Full-scale error: Add ±0.20%FSR to the MAX. value when AVREFP = VDD.

Integral linearity error/ Differential linearity error: Add ±2.0 LSB to the MAX. value when AVREFP = VDD.

(2) When reference voltage (+) = VDD (ADREFP0 = 0), reference voltage (–) = VSS (ADREFM = 0), target pin: ANI0 to ANI2, ANI13, ANI14, ANI19

(TA = –40 to +85 °C, 1.8 V ≤ VDD ≤ 3.6 V, VSS = 0 V, Reference voltage (+) = VDD, Reference voltage (–) = VSS)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution	1.8 V ≤ VDD ≤ 3.6 V		1.2	±7.0	LSB
Conversion time	tCONV	10-bit resolution Target pin: ANI0 to ANI2, ANI13, ANI14, ANI19	2.7 V ≤ VDD ≤ 3.6 V	3.1875		39	μs
			1.8 V ≤ VDD ≤ 3.6 V	17		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution	1.8 V ≤ VDD ≤ 3.6 V			±0.60	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution	1.8 V ≤ VDD ≤ 3.6 V			±0.60	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	1.8 V ≤ VDD ≤ 3.6 V			±4.0	LSB
Differential linearity error Note 1	DLE	10-bit resolution	1.8 V ≤ VDD ≤ 3.6 V			±2.0	LSB
Analog input voltage	VAIN	ANI0 to ANI2, ANI13, ANI14, ANI19		0		VDD	V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (% FSR) to the full-scale value.

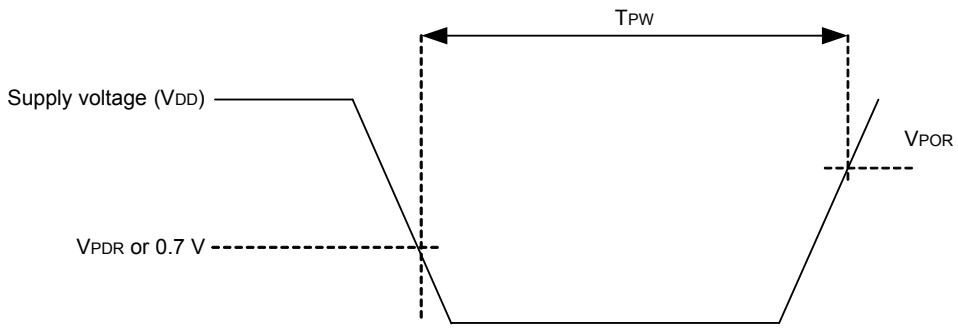
31.6.2 POR characteristics

(TA = –40 to +85 °C, VSS = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Voltage detection threshold	VPOR	Voltage threshold on VDD rising	1.47	1.51	1.55	V
	VPDR	Voltage threshold on VDD falling Note 1	1.46	1.50	1.54	V
Minimum pulse width Note 2	TPW		300			μs

Note 1. However, when the operating voltage falls while the LVD is off, enter STOP mode, or enable the reset status using the voltage detection function or external reset pin before the voltage falls below the operating voltage range shown in **31.4 AC Characteristics**.

Note 2. Minimum time required for a POR reset when VDD exceeds below VPDR. This is also the minimum time required for a POR reset from when VDD exceeds below 0.7 V to when VDD exceeds VPOR while STOP mode is entered or the main system clock (fMAIN) is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



31.6.3 LVD characteristics

(1) Reset Mode and Interrupt Mode

(T_A = –40 to +85 °C, $V_{PDR} \leq V_{DD} \leq$ 3.6 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Voltage detection threshold	Supply voltage level	V_{LVD2}	Rising edge	3.07	3.13	3.19	V
			Falling edge	3.00	3.06	3.12	V
		V_{LVD3}	Rising edge	2.96	3.02	3.08	V
			Falling edge	2.90	2.96	3.02	V
		V_{LVD4}	Rising edge	2.86	2.92	2.97	V
			Falling edge	2.80	2.86	2.91	V
		V_{LVD5}	Rising edge	2.76	2.81	2.87	V
			Falling edge	2.70	2.75	2.81	V
		V_{LVD6}	Rising edge	2.66	2.71	2.76	V
			Falling edge	2.60	2.65	2.70	V
		V_{LVD7}	Rising edge	2.56	2.61	2.66	V
			Falling edge	2.50	2.55	2.60	V
		V_{LVD8}	Rising edge	2.45	2.50	2.55	V
			Falling edge	2.40	2.45	2.50	V
		V_{LVD9}	Rising edge	2.05	2.09	2.13	V
			Falling edge	2.00	2.04	2.08	V
		V_{LVD10}	Rising edge	1.94	1.98	2.02	V
			Falling edge	1.90	1.94	1.98	V
		V_{LVD11}	Rising edge	1.84	1.88	1.91	V
			Falling edge	1.80	1.84	1.87	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time						300	μs

(2) Interrupt & Reset Mode

(T_A = –40 to +85 °C, $V_{PDR} \leq V_{DD} \leq$ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Voltage detection threshold	V_{LVDB0}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 1, falling reset voltage			1.80	1.84	1.87	V
	V_{LVDB1}	LVIS1, LVIS0 = 1, 0		Rising release reset voltage	1.94	1.98	2.02	V
				Falling interrupt voltage	1.90	1.94	1.98	V
	V_{LVDB2}	LVIS1, LVIS0 = 0, 1		Rising release reset voltage	2.05	2.09	2.13	V
				Falling interrupt voltage	2.00	2.04	2.08	V
	V_{LVDB3}	LVIS1, LVIS0 = 0, 0		Rising release reset voltage	3.07	3.13	3.19	V
				Falling interrupt voltage	3.00	3.06	3.12	V
	V_{LVDC0}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 0, falling reset voltage			2.40	2.45	2.50	V
	V_{LVDC1}	LVIS1, LVIS0 = 1, 0		Rising release reset voltage	2.56	2.61	2.66	V
				Falling interrupt voltage	2.50	2.55	2.60	V
	V_{LVDC2}	LVIS1, LVIS0 = 0, 1		Rising release reset voltage	2.66	2.71	2.76	V
				Falling interrupt voltage	2.60	2.65	2.70	V
	V_{LVDD0}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 1, falling reset voltage			2.70	2.75	2.81	V
	V_{LVDD1}	LVIS1, LVIS0 = 1, 0		Rising release reset voltage	2.86	2.92	2.97	V
				Falling interrupt voltage	2.80	2.86	2.91	V
	V_{LVDD2}	LVIS1, LVIS0 = 0, 1		Rising release reset voltage	2.96	3.02	3.08	V
				Falling interrupt voltage	2.90	2.96	3.02	V

31.6.4 Power supply voltage rising slope characteristics

(T_A = –40 to +85 °C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

Caution Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 31.4 AC Characteristics.

31.7 RF Transceiver Characteristics

31.7.1 Recommended operating conditions

Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage		1.8	3.0	3.6	V
Operating ambient temperature		–40		85	°C
XIN frequency			48		MHz
Operating frequency		863		928	MHz
Channel interval		12.5/200/400/600			kHz
Data rate	2FSK/GFSK	10/20/40/50/100/150/200/300			kbps
	4FSK/GFSK	200/400			kbps
Modulation index	2FSK/GFSK	0.5/1.0			—
	4FSK/GFSK		0.33		—

31.7.2 XIN Frequency Deviation

31.7.2.1 Compatible with IEEE802.15.4g

XIN frequency accuracy is required according to the table below to satisfy the IEEE802.15.4g standard.

Frequency band [MHz]	Maximum frequency [MHz]	Symbol rate [ksymbol/s]	Variable index	MIN.	TYP.	MAX.	Unit	IEEE standard
863	870	50	1	–31.6	—	31.6	ppm	31.6
	870	100	1	–50.0	—	50.0	ppm	50.0
	870	100	0.33	–20.9	—	20.9	ppm	20.9
896	901	10	0.5	–3.1	—	3.1	ppm	3.1
	901	20	0.5	–6.1	—	6.1	ppm	6.1
	901	40	0.5	–12.2	—	12.2	ppm	12.2
901	902	10	0.5	–3.1	—	3.1	ppm	3.1
	902	20	0.5	–6.1	—	6.1	ppm	6.1
	902	40	0.5	–12.2	—	12.2	ppm	12.2
915	928	50	1	–29.6	—	29.6	ppm	29.6
	928	150	0.5	–44.4	—	44.4	ppm	44.4
	928	200	0.5	–50.0	—	50.0	ppm	50.0
917	923.5	50	1	–29.8	—	29.8	ppm	29.8
	923.5	150	0.5	–44.6	—	44.6	ppm	44.6
	923.5	200	0.5	–50.0	—	50.0	ppm	50.0
920	928	50	1	–29.6	—	29.6	ppm	29.6
	928	100	1	–50.0	—	50.0	ppm	50.0
	928	200	1	–50.0	—	50.0	ppm	50.0
	928	200	0.33	–39.1	—	39.1	ppm	39.1

31.7.2.2 Compatible with ARIB Standard

XIN frequency accuracy is required according to the table below to satisfy the ARIB standard.

Frequency band [MHz]	Maximum frequency [MHz]	Symbol rate [ksymbols/s]	Variable index	MIN.	TYP.	MAX.	Unit
920	928	50	1	–20	—	+20	ppm
	928	100	1	–20	—	+20	ppm
	928	200	1	–20	—	+20	ppm
	928	200	0.33	–20	—	+20	ppm

31.7.3 DC characteristics

(T_A = 25 °C, V_{DDRF} =3.0 V, V_{SSRF} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
High-level output current	I_{OHRF}	GPIO0 to GPIO4 INTOUT	1.8V ≤ V_{DDRF} ≤ 3.6 V		–2.0		mA
Low-level output current	I_{OLRF}	GPIO0 to GPIO4 INTOUT	1.8V ≤ V_{DDRF} ≤ 3.6 V		2.0		mA
High-level input voltage	V_{IHRF}	STANDBY, GPIO0 to GPIO4		0.85 V_{DDRF}		V_{DDRF}	V
Low-level input voltage	V_{ILRF}	STANDBY, GPIO0 to GPIO4, MODE1, MODE2		0		0.1 V_{DDRF}	V
High-level output voltage	V_{OHRF}	I_{OH} = –2.0 mA	GPIO0 to GPIO4, INTOUT	V_{DDRF} – 0.3			V
Low-level output voltage	V_{OLRF}	I_{OL} = 2.0 mA	GPIO0 to GPIO4, INTOUT			0.3	V
High-level input leak current	I_{LIHRF}	V_I = V_{DDRF}	STANDBY, GPIO0 to GPIO4			10	μA
Low-level input leak current	I_{LILRF}	V_I = V_{SSRF}	STANDBY, GPIO0 to GPIO4, MODE1, MODE2			–10	μA

31.7.4 Power supply current

(T_A = 25 °C, V_{DDRF} =3.0 V, V_{SSRF} = 0 V)

Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Transmission current (100 kbps, 2GFSK)	+14.5 dBm		53		mA
	+13 dBm		36		
	+10 dBm		21		
Reception current (100 kbps, 2GFSK)	During reception operation RFIP –95 dBm, CW		6.9		mA
	Signal reception standby RFIP none		6.3		
SLEEP mode			0.1		μA
IDLE mode			1.3		mA

31.7.5 Transceiver reception characteristics

(T_A = 25 °C, V_{DDRF} =3.0 V, V_{SSRF} = 0 V) (1/2)

Parameter		Conditions	MIN.	TYP.	MAX.	Unit
Reception sensitivity		2GFSK, BT = 0.5, BER < 0.1%, 10 kbps, m = 0.5	—	–114	–109	dBm
		2GFSK, BT = 0.5, BER < 0.1%, 20 kbps, m = 0.5	—	–111	–106	dBm
		2GFSK, BT = 0.5, BER < 0.1%, 40 kbps, m = 0.5	—	–108	–103	dBm
		2GFSK, BT = 0.5, BER < 0.1%, 50 kbps, m = 1	—	–107	–102	dBm
		2GFSK, BT = 0.5, BER < 0.1%, 100 kbps, m = 0.5	—	–104	–99	dBm
		2GFSK, BT = 0.5, BER < 0.1%, 100 kbps, m = 1	—	–105	–100	dBm
		2GFSK, BT = 0.5, BER < 0.1%, 150 kbps, m = 0.5	—	–102	–97	dBm
		2GFSK, BT = 0.5, BER < 0.1%, 200 kbps, m = 0.5	—	–101	–96	dBm
		2GFSK, BT = 0.5, BER < 0.1%, 200 kbps, m = 1	—	–102	–97	dBm
		2GFSK, BT = 0.5, BER < 0.1%, 300 kbps, m = 0.5	—	–97	–92	dBm
		4GFSK, BT = 0.5, BER < 0.1%, 200 kbps, m = 0.33	—	–102	–97	dBm
		4GFSK, BT = 0.5, BER < 0.1%, 400 kbps, m = 0.33	—	–100	–95	dBm
Maximum input level		2GFSK	0	12	—	dBm
Spurious radiation		1 GHz or lower	—	—	–57	dBm
		1 GHz or higher			–47	dBm
ED	Input range	2GFSK, BT = 0.5, 100 kbps, m = 1	–105		–5	dBm
	Total accuracy		–6		6	dB
RSSI	Input range	2GFSK, BT = 0.5, 100 kbps, m = 1	–100	—	–5	dBm
	Total accuracy		–6	—	6	dB
	Relative accuracy		–3	—	3	dB
Adjacent CH suppression ratio	±200 kHz (50 kbps, m = 1)	2GFSK, BT = 0.5, Desired signal 3 dB above the input sensitivity level, CW interferer, BER < 0.1%	—	35	—	dB
	±400 kHz (100 kbps, m = 1)		—	40	—	dB
Next-adjacent CH suppression ratio	±400 kHz (50 kbps, m = 1)	2GFSK, BT = 0.5, Desired signal 3 dB above the input sensitivity level, CW interferer, BER < 0.1%	—	45	—	dB
	±800 kHz (100 kbps, m = 1)		—	50	—	dB

(T_A = 25 °C, V_{DDRF} =3.0 V, V_{SSRF} = 0 V) (2/2)

Parameter		Conditions	MIN.	TYP.	MAX.	Unit
Suppression ratio	±2 MHz	Desired signal 3 dB above the input sensitivity level, CW interferer, BER < 0.1%	—	48	—	dB
	±10 MHz		—	60	—	dB
	±60 MHz		—	60	—	dB
Image suppression ratio		Desired signal 3 dB above the input sensitivity level, CW interferer, -2* if frequency offset	—	25	—	dB

31.7.6 Transceiver transmission characteristics

(T_A = 25 °C, V_{DDRF} =3.0 V, V_{SSRF} = 0 V)

Parameter		Conditions	MIN.	TYP.	MAX.	Unit
Maximum transmission output power		3.6 V ≥ V_{DDRF} > 2.4 V	14.5	15.3	—	dBm
		2.4 V ≥ V_{DDRF} ≥ 1.8 V	—	13.0	—	dBm
Minimum transmission output power			—	–14.0	—	dBm
Variable step size		Within 6 dB from maximum output power (3.6 V ≥ V_{DDRF} > 2.4 V).	—	0.5	—	dB
Transmission output power variation (Power supply voltage variation, temperature variation)		3.6 V ≥ V_{DDRF} ≥ 1.8 V –40 to 85 °C	–1.5	—	1.5	dB
High frequency	2nd order harmonics	At +13 dBm output	—	—	–33.0	dBm
		At +14.5 dBm output	—	—	–31.5	dBm
	3rd order harmonics	At +13 dBm output	—	—	–33.0	dBm
		At +14.5 dBm output	—	—	–31.5	dBm

Caution Variable step size and variable range may vary depending on the maximum transmission output level setting.

31.7.7 IEEE802.15.4g frequency/data rate table

Frequency band identifier	PHY	Frequency band (MHz)	Operating mode	Modulation	Data rate (kbps)	Symbol rate (ksps)	Modulation index	Channel spacing (kHz)	Total number of channels	Channel 0 frequency (MHz)
4	863 MHz (Europe)	863 to 870	#1	2FSK/2GFSK	50	50	1	200	34	863.125
			#2		100	100		400	17	863.225
			#3	4FSK/4GFSK	200	100	0.33			
5	896 MHz (US)	896 to 901	#1	2FSK/2GFSK	10	10	0.5	25	399	896.0125
			#2		20	20		50	397	896.025
			#3		40	40		100	393	896.05
6	901 MHz (US)	901 to 902	#1	2FSK/2GFSK	10	10	0.5	25	79	901.0125
			#2		20	20		50	77	901.025
			#3		40	40		100	73	901.05
7	915 MHz (US)	902 to 928	#1	2FSK/2GFSK	50	50	1	200	129	902.2
			#2		150	150	0.5	400	64	902.4
			#3		200	200				
8	917 MHz (Korea)	917 to 923.5	#1	2FSK/2GFSK	50	50	1	200	32	917.1
			#2		150	150	0.5	400	16	917.3
			#3		200	200				
9	920 MHz (Japan)	920 to 928	#1	2FSK/2GFSK	50	50	1	200	38	920.6
			#2		100	100		400	18	920.9
			#3		200	200		600	12	920.8
			#4	4FSK/4GFSK	400	200	0.33			

31.7.8 AC Characteristics

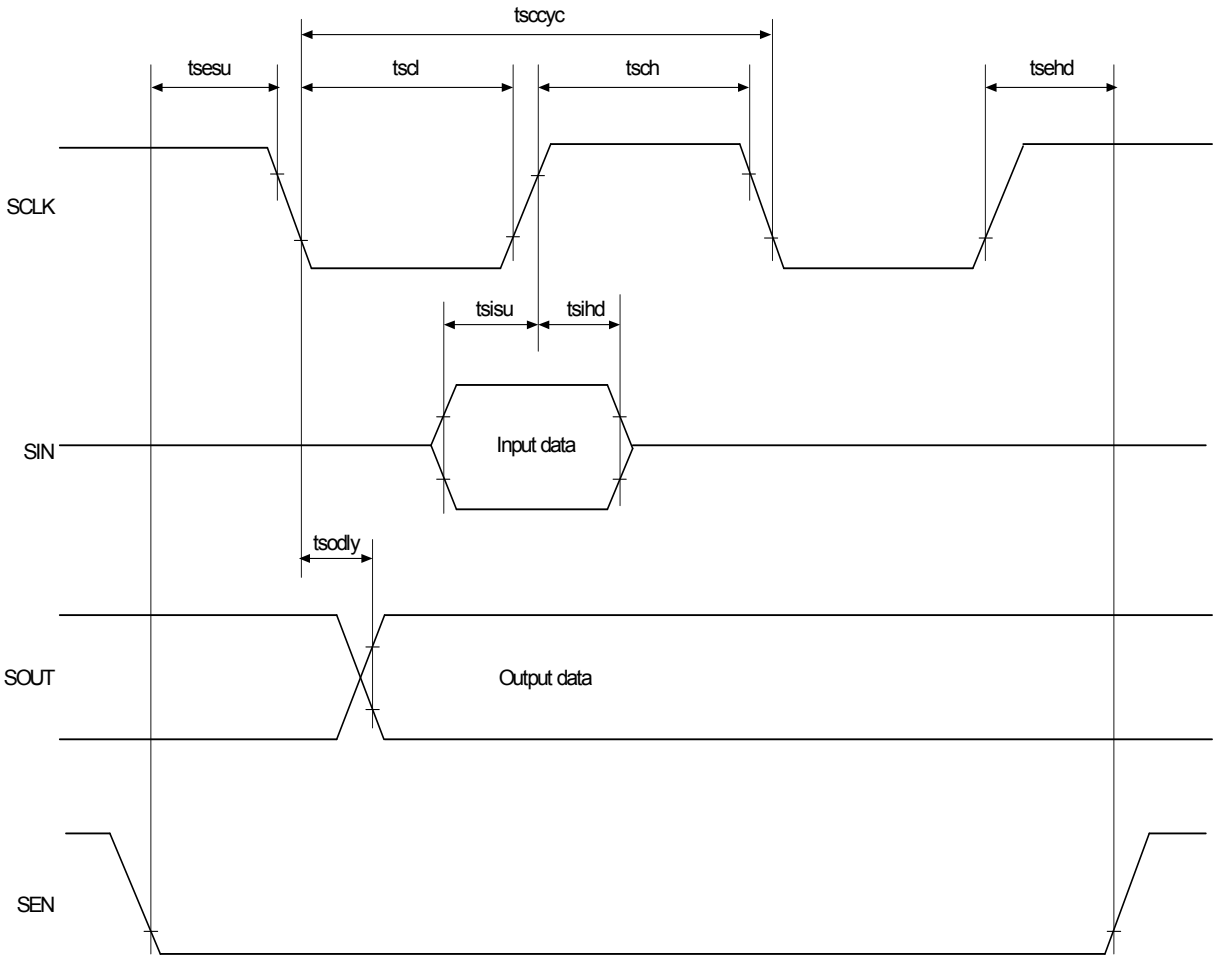
(T_A = −40 °C, to +85 °C, 2.4 V ≤ V_{DDRF} ≤ 3.6 V, V_{SSRF} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SCKL cycle time	tsccyc		250			ns
SEN setup time	tsesu		200			ns
SEN hold time	tsehd		200			ns

(T_A = −40 °C, to +85 °C, 1.8 V ≤ V_{DDRF} ≤ 3.6 V, V_{SSRF} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SCKL cycle time	tsccyc		500			ns
SEN setup time	tsesu		400			ns
SEN hold time	tsehd		400			ns

Figure 31 - 1 Data I/O timing



(T_A = −40 °C, to +85 °C, 1.8 V ≤ V_{DDRF} ≤ 3.6 V, V_{SSRF} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
STANDBYlow-level width	tstbyl		10			μs
OSCDRVSEL setup time (From STANDBY↑)	todssu	Crystal resonator	500			μs
DON setup time (From OSCDRVSEL↑)	tdonsu	Crystal resonator	50			μs
RFRESETB setup time (From DON↑)	trfrstsu	Crystal resonator	450			μs
RFRESETBsetup time (From STANDBY↑)	trfrstsu2	External clock input	450			μs

Figure 31 - 2 Timing using crystal resonator

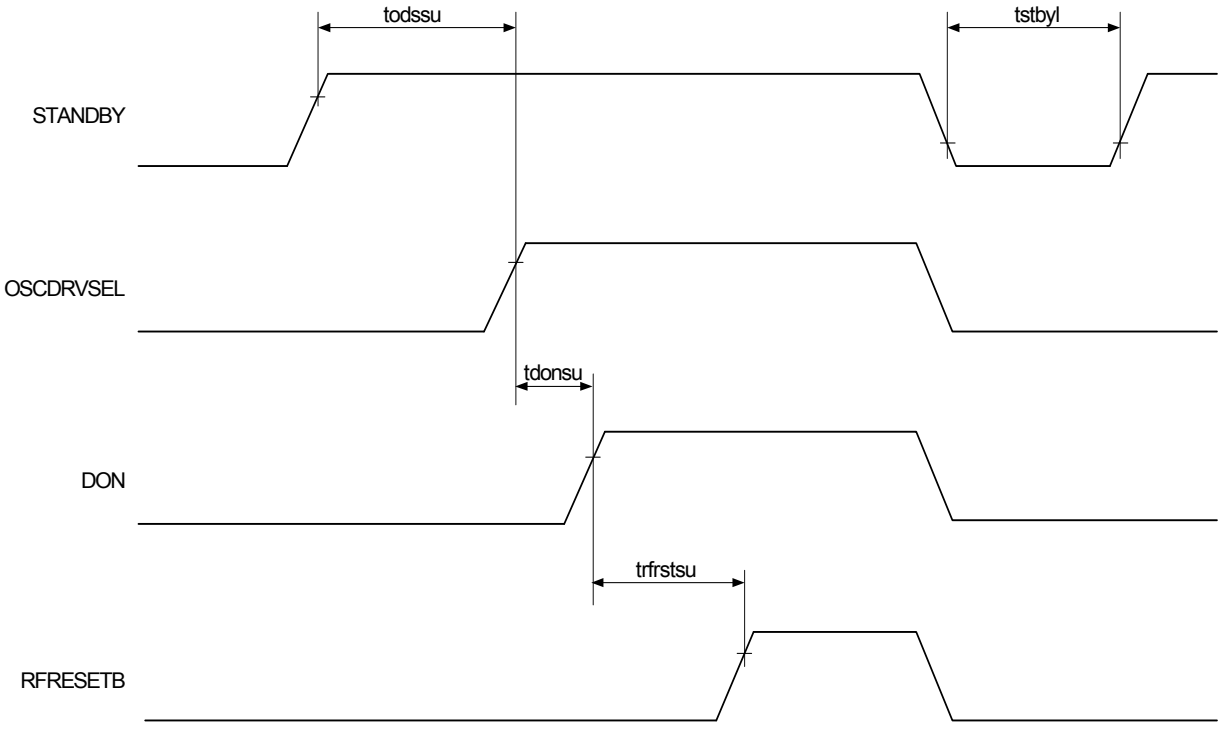
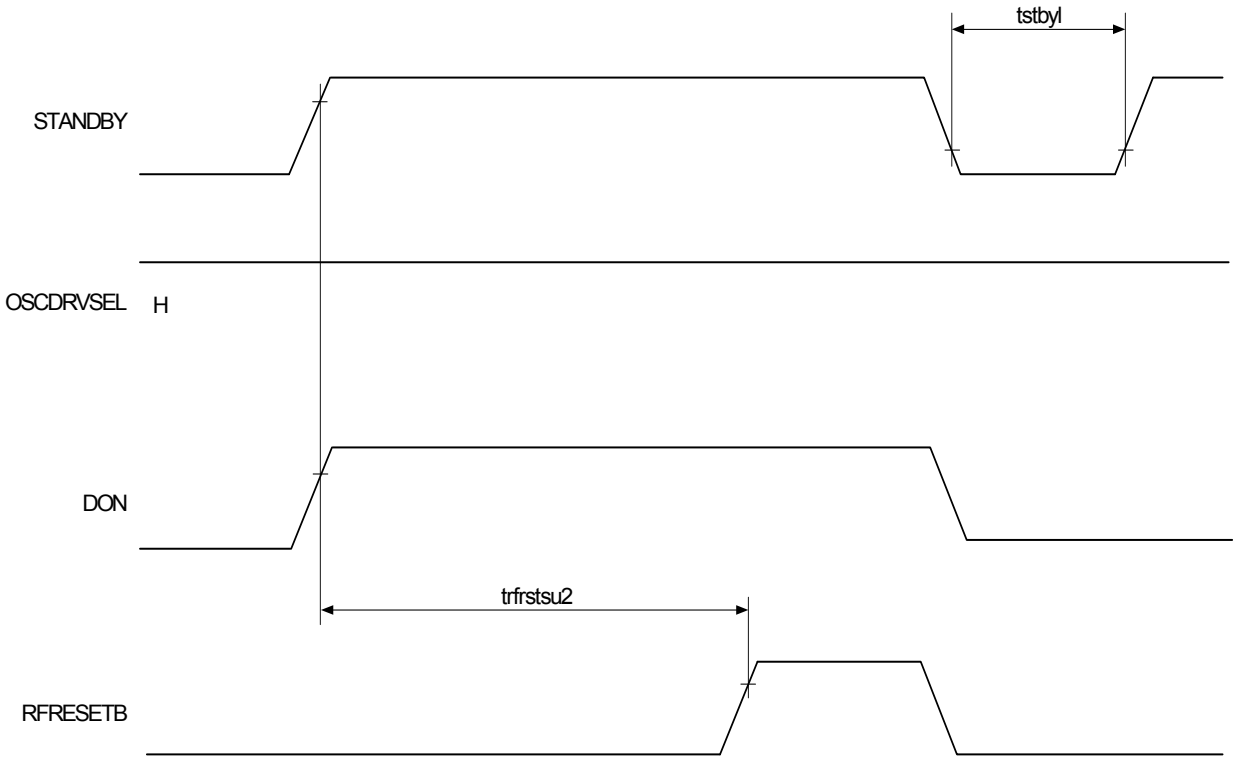


Figure 31 - 3 Timing input external clock

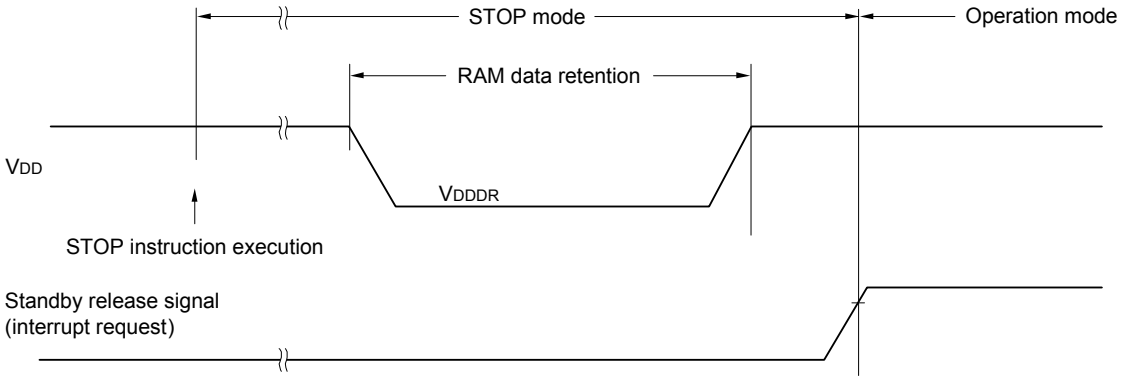


31.8 RAM Data Retention Characteristics

(T_A = −40 to +85 °C, V_{SS} = 0V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.46 **Note**		3.6	V

Note The value depends on the POR detection voltage. When the voltage drops, the RAM data is retained before a POR reset is effected, but RAM data is not retained when a POR reset is effected.



31.9 Flash Memory Programming Characteristics

(T_A = −40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}	1.8 V ≤ V_{DD} ≤ 3.6 V		1		32	MHz
Number of code flash rewrites **Notes 1, 2, 3**	C_{erwr}	Retained for 20 years	T_A = 85 °C	1,000			Times
Number of data flash rewrites **Notes 1, 2, 3**		Retained for 1 year	T_A = 25 °C		1,000,000		
		Retained for 5 years	T_A = 85 °C	100,000			
		Retained for 20 years	T_A = 85 °C	10,000			

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

Note 2. When using flash memory programmer and Renesas Electronics self-programming library.

Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

31.10 Dedicated Flash Memory Programmer Communication (UART)

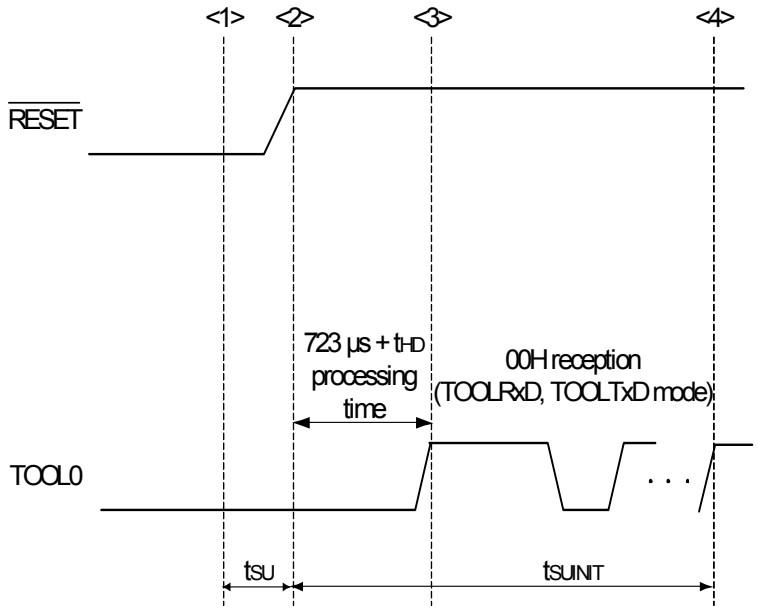
(T_A = -40 to +85 °C, 1.8 V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

31.11 Timing for Switching Flash Memory Programming Modes

(T_A = −40 to +85 °C, 1.8 V ≤ V_{DD} ≤ 3.6 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
How long from when an external reset ends until the initial communication settings are specified	t_{SUINIT}	POR and LVD reset must end before the external reset ends.			100	ms
How long from when the TOOL0 pin is placed at the low level until an external reset ends	t_{SU}	POR and LVD reset must end before the external reset ends.	10			μs
How long the TOOL0 pin must be kept at the low level after an external reset ends (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms



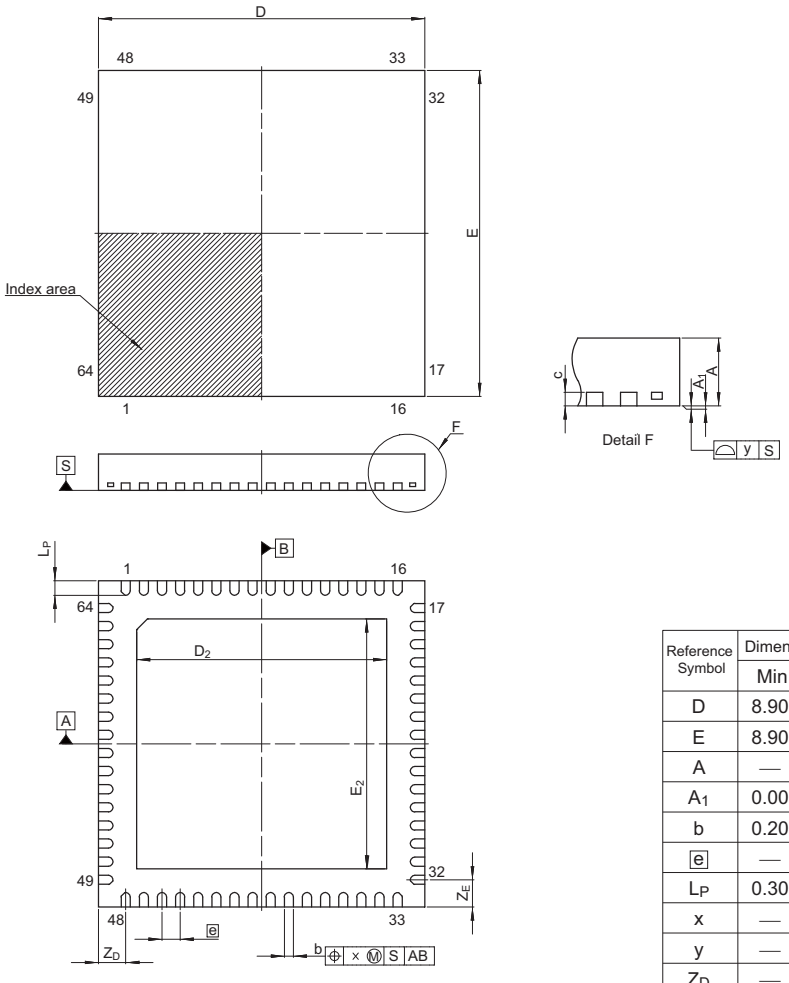
<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends).
<3> The TOOL0 pin is set to the high level.
<4> The baud rate setting by UART reception is completed.

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. (excluding the processing time of the firmware to control the flash memory)

CHAPTER 32 PACKAGE DRAWING

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-HVQFN64-9x9-0.50	PVQN0064KC-A	—	0.21

Unit: mm



Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	8.90	9.00	9.10
E	8.90	9.00	9.10
A	—	—	1.00
A_1	0.00	—	—
b	0.20	0.25	0.30
e	—	0.50	—
L_P	0.30	0.40	0.50
x	—	—	0.05
y	—	—	0.05
Z_D	—	0.75	—
Z_E	—	0.75	—
c	—	0.20	—
D_2	—	6.90	—
E_2	—	6.90	—

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

(1/2)

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.6	Change of Caution in 1.3 Pin Configuration (Top View)	(c)
p.9	Change of 1.6 Outline of Functions	(a)
CHAPTER 3 PIN FUNCTIONS		
p.23	Addition of pins to Table 3 - 3 Connection of Unused Pins	(c)
CHAPTER 5 PORT FUNCTIONS		
p.93	Change of Table 5 - 11 Setting Examples of Registers When Using P130 to P144 Pin Function	(a)
CHAPTER 6 CLOCK GENERATOR		
p.116	Change of Figure 6 - 13 Format of High-speed on-chip oscillator trimming register (HIOTRM)	(a)
p.121	Change of 6.4.4 Low-speed on-chip oscillator	(a)
p.124	Change of 6.6.1 Example of setting high-speed on-chip oscillator	(c)
p.132	Change Notes of Table 6 - 5 CPU Clock Transition and SFR Register Setting Examples (3/5)	(a)
CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER		
p.270	Change of Figure 11 - 2 Format of Clock output select registers n (CKSn)	(a)
CHAPTER 12 WATCHDOG TIMER		
p.278	Addition of Note to Table 12 - 4 Setting Window Open Period of Watchdog Timer	(c)
CHAPTER 13 A/D CONVERTER		
p.309	Change of titles 13.6.4 Hardware trigger no-wait mode (scan mode, one-shot conversion mode) and Figure 13 - 21 Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing	(a)
CHAPTER 14 SERIAL ARRAY UNIT		
p.348	Change of 14.3.13 Serial output level register m (SOLm)	(a)
CHAPTER 16 DATA TRANSFER CONTROLLER (DTC)		
p.518	Change of Figure 16 - 2 Memory Map Example when DTCBAR Register is Set to FBH	(a)
p.519	Change of Table 16 - 4 Start Address of Control Data	(a)
p.520	Change of Figure 16 - 4 Start Address of Control Data and Vector Table	(a)
CHAPTER 18 RF TRANSCEIVER		
p.546	Change of description in 18.2.1 (4) GPIO0/CLKOUT, GPIO1/ANTSELOUT0, GPIO2/ANTSELOUT1, GPIO3, GPIO4/ANTSW	(c)
p.550	Deletion of description in 18.3.3 (2) RF reference clock output	(b)
p.553	Change of title names of Table 18 - 2 For Normal Receive and Table 18 - 3 For Antenna Diversity Receive	(c)
p.556	Change of Table 18 - 4 Interrupt Sources List and Note 2, and addition of Note 4	(c)
p.595	Change of Figure 18 - 36 Baseband Interrupt Source Register 1 (BBINTREQ1) Format	(a)
p.596	Change of Figure 18 - 37 Baseband Interrupt Source Register 2 (BBINTREQ2) Format	(a)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/2)

Page	Description	Classification
p.598	Addition of description to 18.4.4 (30) Baseband interrupt enable register 1 (BBINTEN1) and Figure 18 - 39 Baseband Interrupt Enable Register 1 (BBINTEN1) Format	(c)
p.598	Change of Figure 18 - 39 Baseband Interrupt Enable Register 1 (BBINTEN1) Format	(a)
p.599	Deletion of description in 18.4.4 (31) Baseband interrupt enable register 2 (BBINTEN2) , and change of Figure 18 - 40 Baseband Interrupt Enable Register 2 (BBINTEN2) Format	(a)
p.603	Change of Figure 18 - 46 Receive Level Threshold Setting Register (BBLVLVTH) Format	(c)
p.650	Addition of descriptions to 18.4.4 (89) Lower limit threshold setting register after SFD detection (PWRLOWTH2) and Figure 18 - 102 Lower limit threshold setting register after SFD detection (PWRLOWTH2) Format	(c)
p.662	Change of Table 18 - 15 Function State in Each RF Mode	(a)
p.666	Change of 18.7.2 Example of procedure for function setting	(c)
CHAPTER 20 STANDBY FUNCTION		
p.740, 741	Change of Note 2 in Figure 20 - 4, Figure 20 - 5	(a)
p.743	Change of 20.3.3 SNOOZE mode setting and operating statuses	(a)
p.744	Change of Table 20 - 4 Operating Statuses in SNOOZE Mode	(a)
CHAPTER 26 OPTION BYTE		
p.800	Addition of Note 3 to Figure 26 - 1 Format of User Option Byte (000C0H/010C0H)	(c)
CHAPTER 31 ELECTRICAL SPECIFICATIONS		
Throughout	Change expression of conditions	(c)
p.857	Addition of item and Note 3 to 31.1 Absolute Maximum Ratings	(c)
p.903	Change of 31.7.7 IEEE802.15.4g frequency/data rate table	(a)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

A.2 Revision History of Preceding Editions

(1/9)

Edition	Description	Chapter
Rev1.10	Addition of Caution to 18.4.4 (57) CCA time register (CCATIME)	CHAPTER 18 RF TRANSCEIVER
	Addition of description to 18.4.4 (87) Address filter extension address control register (BBADFCON)	
	Addition of Caution to 18.7.2 (1) Example of procedure for RF transmission output power setting, (2) RF frequency setting for transmission, and (3) RF frequency setting for reception	
	Addition of 18.8.2 Cautions on First and Second Address Filter Match Monitor Bits	
Rev1.00	Deletion of Preliminary	Throughout CHAPTER 1 OUTLINE
	Change of Table 1 - 1 Ordering Part Number List	
	Change of 1.3 Pin Configuration (Top View)	
	Change of description in 1.4 Pin Identification	
	Change of 1.5 Block Diagram	
	Change of description in 1.6 Outline of Functions	
	Addition of Note in Table 2 - 1 Internal Pin Connection	CHAPTER 2 CONNECTION BETWEEN MCU AND RF TRANSCEIVER
	Change of Figure 2 - 3 RL78/G1H (64-pin) Peripheral Circuits'Connection Diagram (without ANTSW)	
	Change of Figure 2 - 4 RL78/G1H (64-pin) Peripheral Circuits'Connection Diagram (with ANTSW)	
	Addition of Figure 2 - 5 RL78/G1H (64-pin) Peripheral Circuit Connection Diagram (Using TCXO)	
	Change of Figure 4 - 1 Memory Map (R5F11FLJ)	CHAPTER 4 CPU ARCHITECTURE
	Deletion of remark in Figure 13 - 1 Block Diagram of A/D Converter	CHAPTER 13 TA/D CONVERTER
	Addition of description to 18.2.1 (4) GPIO0/CLKOUT, GPIO1/ANTSELOUT0, GPIO2/ANTSELOUT1, GPIO3, GPIO4/ANTSW	CHAPTER 18 RF TRANSCEIVER
	Change of Figure 18 - 1 RF Transceiver Block Diagram	
	Change of Table 18 - 2 For Normal Receive	
	Change of Table 18 - 3 For Antenna Diversity Receive	
	Addition of 18.4.3 Baseband Interrupt	
	Change of Table 18 - 5 Baseband Function Controlling Register	
	Change of description in 18.4.4 (2) Transmit/receive reset register (BBTXRXRST)	
	Change of Figure 18 - 8 Transmit/Receive Reset Register (BBTXRXRST) Format	
	Change of description in 18.4.4 (4) Transmit/receive mode register 1 (BBTXRXMODE1)	
	Change of description in 18.4.4 (5) RSSI/CCA result register (BBRSSICCARSLT)	
	Change of Figure 18 - 11 RSSI/CCA Result Register (BBRSSICCARSLT) Format	
	Addition of Table 18 - 7 Register setting for loss of input signal level from antenna	
	Change of description in 18.4.4 (7) Transmit/receive status register 0 (BBTXRXST0)	
	Change of Figure 18 - 13 Transmit/Receive Status Register 0 (BBTXRXST0) Format	
	Change of description in 18.4.4 (8) Transmit/receive mode register 2 (BBTXRXMODE2)	

(2/9)

Edition	Description	Chapter
Rev1.00	Change of description in 18.4.4 (9) Transmit/receive mode register 3 (BBTXRXMODE3)	CHAPTER 18 RF TRANSCEIVER
	Change of Figure 18 - 15 Transmit/Receive Mode Register 3 (BBTXRXMODE3) Format	
	Change of description in 18.4.4 (10) Transmit/receive status register 1 (BBTXRXST1)	
	Change of Figure 18 - 16 Transmit/Receive Status Register 1 (BBTXRXST1) Format	
	Change of description in 18.4.4 (11) Transmit/receive control register (BBTXRXCON)	
	Change of description in 18.4.4 (13) CCA level threshold setting register (BBCCAVTH)	
	Change of description in 18.4.4 (18) PAN identifier register 0 (BBPANID0)	
	Change of description in 18.4.4 (19) Short address register 0 (BBSHORTAD0)	
	Change of description in 18.4.4 (20) Extended address register 0 (BBEXTENDAD00 to BBEXTENDAD03)	
	Change of description in 18.4.4 (23) Time stamp registers 0 and 1 (BBTSTAMP0, BBTSTAMP1)	
	Change of description in 18.4.4 (24) Timer control register (BBTIMECON)	
	Change of Figure 18 - 37 Baseband Interrupt Source Register 0 (BBINTREQ0) Format	
	Change of description in 18.4.4 (31) Baseband interrupt enable register 2 (BBINTEN2)	
	Change of Figure 18 - 40 Baseband Interrupt Enable Register 2 (BBINTEN2) Format	
	Change of description in 18.4.4 (35) Receive level threshold setting register (BBLVLVTH)	
	Change of Figure 18 - 48 Count Operation in the Non-BEACON Mode	
	Change of Figure 18 - 50 Automatic Receive Switch Compare Register Count Operation	
	Deletion of 18.4.4 (41) CCA start time register (CCASTATIME)	
	Addition of 18.4.4 (41) Communication status register 1 (COMSTATE1)	
	Addition of 18.4.4 (42) Communication status register 2 (COMSTATE2)	
	Change of title and description in 18.4.4 (43) Mode transition	
	Change of description in 18.4.4 (48) ANTSW output timing setting register (BBANTSWTIMG)	
	Change of description in 18.4.4 (49) Receive frame length register (BBRXFLEN)	
	Change of description in 18.4.4 (52) Preamble length setting register (BBPAMBL)	
	Change of description in 18.4.4 (53) Frequency setting register (BBFREQ)	
	Change of description in 18.4.4 (54) Symbol rate setting register (BBSYMBLRATE)	
	Change of description in 18.4.4 (55) SUBG control register (BBSUBGCON)	
	Change of description in 18.4.4 (57) CCA time register (CCATIME)	
	Change of description in 18.4.4 (58) Antenna diversity mode register (BBANTDIV)	
	Change of Figure 18 - 71 Antenna Diversity Mode Register (BBANTDIV) Format	
	Change of description in 18.4.4 (59) Mode switch frame transmit register (BBTXMODESW)	
	Change of Figure 18 - 72 Mode Switch Frame Transmit Register (BBTXMODESW) Format	

(3/9)

Edition	Description	Chapter
Rev1.00	Change of description in 18.4.4 (60) Mode switch frame receive register (BBRXMODESW)	CHAPTER 18 RF TRANSCEIVER
	Change of description in 18.4.4 (61) Transmit data counter register (BBTXCOUNT)	
	Change of description in 18.4.4 (62) PHY header receive register (BBPHRRX)	
	Change of description in 18.4.4 (63) Preamble setting register (BBPABL)	
	Change of description in 18.4.4 (64) SFD setting register (BBSFD)	
	Change of description in 18.4.4 (66) ANT0 read register (BBANT0RD)	
	Change of description in 18.4.4 (67) ANT1 read register (BBANT1RD)	
	Change of description in 18.4.4 (68) Antenna switch time register (BBANTDIVTIM)	
	Change of description in 18.4.4 (69) Receive start timeout setting register (BBANTTIMOUT)	
	Change of description in 18.4.4 (71) Back off period total number register (BBBOPTOTAL)	
	Change of description in 18.4.4 (72) CCA total number register (BBCCATOTAL)	
	Deletion of 18.4.4 (72) RF initial setting register (RFINI)	
	Change of description in 18.4.4 (73) RF initial setting register 00 to 02 (RFINI00 to RFINI02)	
	Addition of 18.4.4 (74) RF initial setting register 10 to 12 (RFINI10 to RFINI12)	
	Change of description in 18.4.4 (75) PAN identifier register 1 (BBPANID1)	
	Change of description in 18.4.4 (76) Short address register 1 (BBSHORTAD1)	
	Change of description in 18.4.4 (77) Extended address register 1 (BBEXTENDAD10 to BBEXTENDAD13)	
	Change of description in 18.4.4 (78) Receive timeout register (BBTIMEOUT)	
	Deletion of in 18.4.4 (77) Transmit CP current select register 1 (TXCPISL1) to 18.4.4 (80) Transmit capacitance and resistance select register 1 (TXCR1)	
	Change of description in 18.4.4 (79) ANTSW control register (ANTSWCON)	
	Change of description in 18.4.4 (80) Clock output control register (CLKOUTCON)	
	Change of description in 18.4.4 (81) Port direction register (GPIODIR)	
	Change of description in 18.4.4 (82) Port data register (GPIODATA)	
	Deletion of in 18.4.4 (85) Upon-transmission power supply setting register 0 (TXCON0)	
	Change of description in 18.4.4 (83) SFD setting register 2 (BBSFD2)	
	Change of description in 18.4.4 (84) SFD setting register 3 (BBSFD3)	
	Change of description in 18.4.4 (85) SFD setting register 4 (BBSFD4)	
	Addition of description to 18.4.4 (86) FEC control register (BBFECCON)	
	Addition of Caution to Figure 18 - 99 FEC Control Register (BBFECCON) Format	
	Addition of description to 18.4.3 (87) Address filter extension address control register (BBADFCON)	
	Change of Figure 18 - 100 Address Filter Extension Address Control Register (BBADFCON) Format	
	Addition of description to 18.4.4 (88) Antenna diversity mode register 2 (BBANTDIV2)	
	Change of Figure 18 - 101 Antenna Diversity Mode Register 2 (BBANTDIV2) Format	
	Addition of 18.4.4 (89) Lower limit threshold setting register after SFD detection (PWRLOWTH2)	

(4/9)

Edition	Description	Chapter
Rev1.00	Addition of description to 18.6.1 (3) IDLE mode	CHAPTER 18 RF TRANSCEIVER
	Change of description in 18.6.3 State transition	
	Change of Figure 18 - 105 RF Unit State Transition	
	Change of Figure 18 - 107 Wake Up Operation (for REFCLKIN_RF External Clock)	
	Change of description in 18.6.4 Mode transition	
	Change of Table 18 - 14 Initial Setting Registers	
	Deletion of description in 18.7.1 (2) Example of procedure for RF reception	
	Change of title in 18.7.1 (4) Example of procedure for CSMA-CA	
	Change of description in 18.7.2 Example of procedure for function setting	
	Addition of Table 18 - 18 Gain Set (Frequency band identifier = 9) to Figure 18 - 111 Relationship between Transmission Output Power and Gain Set (Frequency band identifier = 8	
	Change of Table 18 - 30 RF Frequency Set (Frequency band identifier = 4, Mode = 001) to Figure 18 - 51 RF Frequency Set (Frequency band identifier = Other, Mode = 021)	
	Addition of Caution to 18.7.3 Setting for each data rate	
	Change of Table 18 - 52 to Table 18 - 59 Settings Required for Each Data Rate	
	Addition of note in 18.8 Notice For Using Baseband Function	
	Change of 31.2.2 On-chip oscillator characteristics	CHAPTER 31 ELECTRICAL SPECIFICATIONS
	Change of 31.3.2 Supply current characteristics	
	Addition of 31.7.2.2 Compatible with ARB Standard	
	Change of 31.7.3 DC characteristics	
	Change of 31.7.4 Power supply current	
	Change of 31.7.5 Transceiver reception characteristics	
	Change of 31.7.6 Transceiver transmission characteristics	
	Change of 31.7.7 IEEE802.15.4g frequency/data rate table	
Rev.0.60	Change of description in 1.1 Features	CHAPTER 1 OUTLINE
	Change of description in 1.3 Pin Configuration (Top View)	
	Change of description in 1.4 Pin Identification	
	Change of description in 1.5 Block Diagram	
	Change of description in 1.6 Outline of Functions	
	Change of description in 2.1 Connection Pins of MCU and RF Transceiver	CHAPTER 2 CONNECTION BETWEEN MCU AND RF TRANSCEIVER
	Change of Table 2 - 1 Internal Pin Connection	
	Change of Figure 2 - 2 Power Configuration	
	Addition of 2.6 Peripheral Circuits' Connection Diagram	
	Change of Table 3 - 1 Pin I/O Buffer Power Supplies	CHAPTER 3 PIN FUNCTIONS
	Addition of remark in 3.2 Functions other than port pins	
	Change of remark in Table 3 - 3 Connection of Unused Pins	
	Addition of Figure 3 - 14 Pin Block Diagram of STANDBY, MODE1, MODE2 to Figure 3 - 16 Pin Block Diagram of Pin INTOUT	
	Change of Table 4 - 5 Special Function Register (SFR) List (1/5)	CHAPTER 4 CPU ARCHITECTURE
	Change of Table 4 - 6 Special Function Register (SFR) List (2/5)	

(5/9)

Edition	Description	Chapter
Rev.0.60	Deletion of note 1 in Table 4 - 7 Special Function Register (SFR) List (3/5)	CHAPTER 4 CPU ARCHITECTURE
	Change of Table 4 - 11 Extended Special Function Register (2nd SFR) List (2/7)	
	Change of Table 4 - 15 Extended Special Function Register (2nd SFR) List (6/7)	
		CHAPTER 5 PORT FUNCTIONS
	Change of description in 5.1 Port Functions	
	Change of Table 5 - 1 Port Configuration	
	Change of description in 5.2.1 Port 0	
	Change of description in 5.2.2 Port 1	
	Change of description in 5.2.4 Port 3	
	Change of description in 5.2.9 Port 10	
	Change of description in 5.2.12 Port 14	
	Change of Figure 5 - 1 Format of Port mode register, addition and change of notes	
	Change of note in 5.3.2 Port registers (Pxx)	
	Change of Figure 5 - 2 Format of Port register, addition of note3	
	Deletion of note in Figure 5 - 3 Format of Pull-up resistor option register	
	Deletion of note in Figure 5 - 6 Format of Port mode control register	
	Change of remark in Figure 7 - 22 Format of Timer output register m (TOm)	CHAPTER 7 TIMER ARRAY UNIT
	Change of Figure 7 - 54 Operation Procedure When Input Pulse Interval Measurement Function Is Used	
	Change of Figure 7 - 62 Operation Procedure When Delay Counter Function Is Used	
	Change of remark 2 in 13.4 A/D Converter Conversion Operations	CHAPTER 13 TA/D CONVERTER
	Change of remark in CHAPTER 14 SERIAL ARRAY UNIT	CHAPTER 14 SERIAL ARRAY UNIT
	Change of Table 14 - 1 Configuration of Serial Array Unit, addition of note 2	
	Change of description in 14.3 Registers Controlling Serial Array Unit	
	Change of note and caution in Figure 14 - 6 Format of Serial mode register mn (SMRmn) (1/2)	
	Change of note and caution in Figure 14 - 6 Format of Serial mode register mn (SMRmn) (1/2)	
	Change of note and caution in Figure 14 - 7 Format of Serial mode register mn (SMRmn) (2/2)	
	Change of note1 and caution in Figure 14 - 8 Format of Serial communication operation setting register mn (SCRmn) (1/2)	
	Change of note1 and caution in Figure 14 - 9 Format of Serial communication operation setting register mn (SCRmn) (2/2)	
	Change of note and caution in Figure 14 - 11 Format of Serial flag clear trigger register mn (SIRmn)	
	Change of note in Figure 14 - 12 Format of Serial status register mn (SSRmn) (1/2)	
	Change of note in Figure 14 - 13 Format of Serial status register mn (SSRmn) (2/2)	
	Change of caution in Figure 14 - 22 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units	
	Addition of remark1 in Figure 14 - 71 and Figure 14 - 72 Example of Contents of Registers for UART Transmission of UART (UARTq)	

(6/9)

Edition	Description	Chapter
Rev.0.60	Addition of Figure 14 - 73 Initial Setting Procedure for UART Transmission	CHAPTER 14 SERIAL ARRAY UNIT
	Change of Figure 14 - 75 Procedure for Resuming UART Transmission	
	Change of remark in Figure 14 - 76 Timing Chart of UART Transmission (in Single-Transmission Mode)	
	Change of Figure 14 - 77 Flowchart of UART Transmission (in Single-Transmission Mode)	
	Change of remark in Figure 14 - 78 Timing Chart of UART Transmission (in Continuous Transmission Mode)	
	Change of Figure 14 - 79 Flowchart of UART Transmission (in Continuous Transmission Mode)	
	Addition of remark 2 in 14.6.2 UART reception	
	Change of remark1 in Figure 14 - 80 Example of Contents of Registers for UART Reception of UART (UARTq) (1/2)	
	Change of remark1 in Figure 14 - 81 Example of Contents of Registers for UART Reception of UART (UARTq) (2/2)	
	Change of remark in Figure 14 - 85 Timing Chart of UART Reception	
	Change of Figure 14 - 86 Flowchart of UART Reception	
	Change of remark 2 in 14.6.3 (1) Baud rate calculation expression	
	Change of remark 2 in Table 14 - 3 Selection of Operation Clock For UART	
	Change of remark in 14.6.3 (2) Baud rate error during transmission	
	Change of remark in 14.6.3 (3) Permissible baud rate range for reception	
	Addition of remark to Figure 14 - 89 Processing Procedure in Case of Framing Error	
	Deletion of description in 16.5.7 DTC Activation Sources	CHAPTER 16 DATA TRANSFER CONTROLLER (DTC)
	Change of Table 17 - 1 Correspondence Between ELSELRn Registers (n = 00 to 25) and Peripheral Functions	CHAPTER 17 EVENT LINK CONTROLLER (ELC)
	Change of description in 18.1 RF Transceiver Overview, addition of caution	CHAPTER 18 RF TRANSCEIVER
	Change of description in 18.2.1 Digital pin	
	Addition of 18.2.3 Description of RF pin functions	
	Change of Figure 18 - 1 RF Transceiver Block Diagram	
	Change of Figure 18 - 2 RF Transmission/Reception Block	
	Change of description in 18.3.1 (1) LNA block	
	Change of description in 18.3.1 (7) PA block	
	Change of description in 18.3.2 (1) RF reference clock	
	Change of description in 18.3.2 (2) Preamble value and SFD value	
	Change of Table 18 - 4 Baseband Function Controlling Register (1/2)	
	Change of Table 18 - 5 Baseband Function Controlling Register (2/2)	
	Change of description in 18.4.3 (1) RF start register (BBRFCON)	

(7/9)

Edition	Description	Chapter
Rev.0.60	Change of description in 18.4.3 (2) Transmit/receive reset register (BBTXRXRST)	CHAPTER 18 RF TRANSCEIVER
	Change of description in 18.4.3 (9) Transmit/receive mode register 3 (BBTXRXMODE3)	
	Change of Figure 18 - 15 Transmit/Receive Mode Register 3 (BBTXRXMODE3) Format	
	Deletion of description in 18.4.3(10) Transmit/receive status register 1 (BBTXRXST1)	
	Change of Figure 18 - 16 Transmit/Receive Status Register 1 (BBTXRXST1) Format and addition of caution	
	Change of description in 18.4.3 (11) Transmit/receive control register (BBTXRXCON)	
	Change of description in 18.4.3 (12) CSMA control register 0 (BBCSMACON0)	
	Change of Figure 18 - 18 CSMA Control Register 0 (BBCSMACON0) Format	
	Change of description in 18.4.3 (15) Transmit/receive mode register 4 (BBTXRXMODE4)	
	Change of Figure 18 - 21 Transmit/Receive Mode Register 4 (BBTXRXMODE4) Format	
	Change of Figure 18 - 35 BBaseband Interrupt Source Register 0 (BBINTREQ0) Format, addition of caution	
	Change of Figure 18 - 36 Baseband Interrupt Source Register 1 (BBINTREQ1) Format, addition of caution	
	Change of Figure 18 - 37 Baseband Interrupt Source Register 2 (BBINTREQ2) Format, addition of caution	
	Change of Figure 18 - 38 Baseband Interrupt Enable Register 0 (BBINTEN0) Format, addition of caution	
	Change of Figure 18 - 39 Baseband Interrupt Enable Register 1 (BBINTEN1) Forma, addition of caution	
	Change of Figure 18 - 40 Baseband Interrupt Enable Register 2 (BBINTEN2) Format	
	Change of description in 18.4.3 (32) CSMA control register 3 (BBCSMACON3)	
	Deletion of 18.4.3 (33) Calibration register (BBCAL)	
	Change of description in 18.4.3 (35) Receive level threshold setting register (BBLVLVTH)	
	Deletion of 18.4.3 (43) CCA data read register (CCADATARD)	
	Change of description in 18.4.3 (42) Estimate control register 1 (BBEVAREG)	
	Change of description in 18.4.3 (47) ANTSW output timing setting register (BBANTSWTIMG)	
	Change of Figure 18 - 60 ANTSW Output Timing Setting Register (BBANTSWTIMG) Format	
	Change of description in 18.4.3 (54) SUBG control register (BBSUBGCON)	
	Change of Figure 18 - 67 SUBG Control Register (BBSUBGCON) Format, addition of caution	
	Change of Figure 18 - 68 Modulation Method Setting Register (BBMODSET) Format	
	Change of description in 18.4.3 (57) Antenna diversity mode register (BBANTDIV)	

(8/9)

Edition	Description	Chapter
Rev.0.60	Change of Figure 18 - 70 Antenna Diversity Mode Register (BBANTDIV) Format	CHAPTER 18 RF TRANSCEIVER
	Change of description in 18.4.3 (60) Transmit data counter register (BBTXCOUNT)	
	Change of description in 18.4.3 (63) SFD setting register (BBSFD)	
	Change of description in 18.4.3 (68) Receive start timeout setting register (BBANTTIMOUT)	
	Addition of 18.4.3(72) RF initial setting register (RFINI)	
	Deletion of 18.4.3(74) RF serial 2 control registers 0 to 2 (RFSER2REG0 to RFSER2REG2)	
	Change of description in 18.4.3 (81) ANTSW control register (ANTSWCON)	
	Change of description in 18.4.3 (84) Port data register (GPIODATA)	
	Addition of 18.4.3 (86) SFD setting register 2 to 18.4.3 (91) Antenna diversity mode register 2	
	Change of description in 18.5.1 Overview	
	Change of Table 18 - 9 Contents of Communication Format	
	Change of description in 18.6.4 Mode transition	
	Change of Figure 18 - 108 Wake Up Operation (for XTAL_RF Oscillator)	
	Change of Figure 18 - 109 Wake Up Operation (for REFCLKIN_RF External Clock)	
	Addition of Table 18 - 10 Pin Status	
	Change of Table 18 - 11 Description of Each State of Wake Up Operation	
	Addition of Table 18 - 12 Initial setting registers	
	Change of description in 18.6.4 (2) Power Down (transition from IDLE to SLEEP) operation	
	Change of Table 18 - 13 Function State in Each RF Mode	
	Change of description in 18.7.1 Example of procedure for each operation	
	Change of description in 18.7.2 Example of procedure for function setting	
	Change of Table 18 - 15 to Table 18 - 20 Gain set (Frequency band identifier = 9)	
	Change of Figure 18 - 110 Relationship between Transmission Output Power and Gain Set	
	Addition of Table 18 - 21 Gain set (Frequency band identifier = 4) to Figure 18 - 113 Relationship between Transmission Output Power and Gain Set (Frequency band identifier = 8)	
	Addition of 18.7.2 (2) RF frequency setting for transmission	
	Addition of 18.7.2 (3) RF frequency setting for reception	
	Change of description in 18.7.3 Setting for each data rate	
	Addition of Table 18 - 61 To Table 18 - 69 Settings Required for Each Data Rate	
	Change of Table 19 - 1 Interrupt Source List (1/2)	CHAPTER 19 INTERRUPT FUNCTIONS
	Change of Table 19 - 4, 19 - 5 Flags Corresponding to Interrupt Request Sources and change, deletion of note 1 and note 2	
	Change of Figure 19 - 4, 19 - 5 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)	

(9/9)

Edition	Description	Chapter
Rev.0.60	Change of Figure 19 - 6, 19 - 7 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)	CHAPTER 19 INTERRUPT FUNCTIONS
	Addition	CHAPTER 22 POWER-ON-RESET CIRCUIT
	Change of description in 24.1 (7) A/D test function	CHAPTER 24 ISAFETY FUNCTIONS
	Addition	CHAPTER 25 REGULATOR
	Addition	CHAPTER 29 BCD CORRECTION CIRCUIT
	Change of the title and description in CHAPTER 31 ELECTRICAL SPECIFICATIONS	CHAPTER 31 ELECTRICAL SPECIFICATIONS
	Change of description in 31.1 Absolute Maximum Ratings	
	Change of description in 31.3.1 Pin characteristics	
	Change of description in 31.3.2 Supply current characteristics	
	Change of remark in 31.5.1 (2) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output)	
	Deletion of remark 1 in 31.5.1 (3) uring communication at same potential (CSI mode) (master mode, SCKp... internal clock output, supported only for CSI20)	
	Change of remark 1 in 31.5.1 (4) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input)	
	Change of remark 3 and 4 in CSI mode connection diagram (during communication at different potential)	
	Change of remark 2 in Figure CSI mode serial transfer timing (slave mode) (during communication at different potential)(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)	
	Change of description in 31.7.1 Recommended operating conditions	
	Change and addition of description in 31.7.2 XIN frequency deviation (compatible with IEEE802.15.4g)	
	Change of description in 31.7.3 DC characteristics	
	Change of description in 31.7.4 Power supply current	
	Change of description in 31.7.5 Transceiver reception characteristics	
	Change of description in 31.7.6 Transceiver transmission characteristics	
	Change of description in 31.7.7 IEEE802.15.4g frequency/data rate table	
	Addition of 31.7.8 AC Characteristics	
	Change of description in 31.8 RAM Data Retention Characteristics	CHAPTER 31 ELECTRICAL SPECIFICATIONS
	Change of remark 3 and 4 in CSI mode connection diagram (during communication at different potential)	
Rev.0.50	First Edition issued.	Throughout

RL78/G1H User's Manual: Hardware

Publication Date: Rev. 0.50 Sep 18, 2015
Rev. 1.20 Dec 22, 2016

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 America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

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 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.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, P. R. China 200333
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 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RL78/G1H

Renesas Electronics Corporation

R01UH0575EJ0120

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 16-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

M30302FCPFP#U3 MB90F036APMC-GSE1 MB90F428GCPFR-GSE1 MB96F683RBPMC-GSAE1 R5F10MMGDFB#30 R5F111PGGFB#30 R5F117BCGNA#20 DF3026XBL25V DF36014GFTV DF36014GFXV DF36024GFTV DF36034GFPV R5F11B7EANA#U0 R5F21172DSP#U0 MB90092PF-G-BNDE1 MB90F335APMC1-G-SPE1 MB90F345CAPFR-GSE1 MB90F568PMCR-GE1 MB96F395RSAPMC-GSE2 DF36024GFXV MB96F018RBPMC-GSE1 MB90F867ASPFR-GE1 DF2239FA20IV R5F117BCGFP#30 LC88F58B0AU-SQFPH MB90F548GPF-GE1 MB90214PF-GT-310-BND-AE1 MB90F342CESPQC-GSE2 MB90F428GAPF-GSE1 ML620Q504H-NNNTBWBX S912ZVH128F2VLL UPD78F1500AGK-GAK-AX HD64F3337SF16V MB90F428GCPF-GSE1 MB90F342ESPMC-G-JNE1 MB90022PF-GS-358E1 MB96F395RWAPMC-GSE2 MB96395RSAPMC-GS-110E2 MB90F883CSPMC-GE1 S912ZVHY64F1CLL S912ZVHY64F1VLQ ST10F280 MB96F338RSAPMCR-GK5E2 CY90096PF-G-002-BND-ERE1 ML62Q1569-NNNGAZ0AX ML62Q1739-NNNGAZ0AX ML62Q1749-NNNGAZ0AX ML62Q1579-NNNGAZ0AX ML62Q1559-NNNGAZ0AX ML62Q1729-NNNGAZ0AX