User's Manual

16

RL78/F12

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.1.11 Jan 2014

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 users who wish to understand the functions of the RL78/F12 and design application systems using the following devices.

- 20-pin: R5F1096x (x = 8, A, B, C, D, E)
- 30-pin: R5F109Ax (x = A, B, C, D, E)
- 32-pin: R5F109Bx (x = A, B, C, D, E)
- 48-pin: R5F100Gx (x = A, B, C, D, E)
- 64-pin: R5F109Lx (x = A, B, C, D, E)

Purpose

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

Organization

The RL78/F12 manual is divided into two parts: this manual (hardware) and the "RL78 Family User's Manual: Software" (common to the RL78 family).

RL78/F12 User's Manual Hardware	**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

<R>

How to Read This Manual

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

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

<R>

<R> **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 referenced in this manual may include preliminary versions. However, preliminary versions are not marked as such.

<R> **Documents Related to Devices**

Document Name	Document No.
RL78/F12 User's Manual	R01UH0231E
RL78 Family User's Manual: Software	R01US0015E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP5 Flash Memory Programmer User's Manual	R20UT0008E

<R> **Other Documents**

Document Name	Document No.
Renesas MPUs & MCUs RL78 Family	R01CP0003E
Semiconductor Package Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E
Semiconductor Reliability Handbook	R51ZZ0001E

Note See the "Semiconductor Package Mount Manual" website (http://www.renesas.com/products/package/manual/index.jsp).

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

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

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

- 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)
- General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ROM: 8 to 64 KB, RAM: 0.5 to 4 KB, Data flash memory: 4 KB
- High-speed on-chip oscillator
  - Select from 32 MHz (TYP.), 24 MHz (TYP.), 16 MHz (TYP.), 12 MHz (TYP.), 8 MHz (TYP.), 4 MHz (TYP.), and 1 MHz (TYP.)
- On-chip single-power-supply flash memory (with prohibition of block erase/writing function)
- Self-programming (with boot swap function/flash shield window function)
- On-chip debug function
- On-chip power-on-reset (POR) circuit and voltage detector (LVD)
- On-chip watchdog timer (operable with the dedicated internal low-speed on-chip oscillator)
- On-chip multiplier and divider/multiply-accumulator
  - 16 bits × 16 bits = 32 bits (Unsigned or signed)
  - 32 bits ÷ 32 bits = 32 bits (Unsigned)
  - 16 bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)
- On-chip key interrupt function
- On-chip clock output/buzzer output controller
- On-chip BCD adjustment
- I/O ports: 16 to 44 (N-ch open drain: 0 to 4)
- Timer
  - 16-bit timer: 8 channels
  - Watchdog timer: 1 channel
  - Real-time clock: 1 channel
  - Interval timer: 1 channel
  - Wakeup timer: 1 channel
- Serial interface
  - CSI: 0 to 8 channels
  - UART/UART (LIN-bus supported): 1 to 5 channels
  - I^2C/Simplified I^2C communication: 0 to 7 channels
- 8/10-bit resolution A/D converter (V_{DD} = 1.8 to 5.5 V): 4 to 12 channels
- Power supply voltage: V_{DD} = 1.8 to 5.5 V (J version), V_{DD} = 2.7 to 5.5 V (K version)
- Operating ambient temperature: T_A = –40 to +85°C (J version), T_A = –40 to +125°C (K version)

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

○ ROM, RAM capacities

Flash ROM	Data flash	RAM	RL78/F12				
			20 pins	30 pins	32 pins	48 pins	64 pins
64 KB	4 KB	4 KB Note	R5F1096E	R5F109AE	R5F109BE	R5F109GE	R5F109LE
48 KB		3 KB	R5F1096D	R5F109AD	R5F109BD	R5F109GD	R5F109LD
32 KB		2 KB	R5F1096C	R5F109AC	R5F109BC	R5F109GC	R5F109LC
24 KB		1.5 KB	R5F1096B	R5F109AB	R5F109BB	R5F109GB	R5F109LB
16 KB		1 KB	R5F1096A	R5F109AA	R5F109BA	R5F109GA	R5F109LA
8 KB		0.5 KB	R5F10968	–	–	–	–

Note This is 3 KB when the self-programming function is used.

1.2 Ordering Information

- **Flash memory version (lead-free product)**

Pin count	Package	Data flash	Part Number
20 pins	20-pin plastic SSOP (7.62 mm (300))	Mounted	R5F10968JSP, R5F1096AJSP, R5F1096BJSP, R5F1096CJSP, R5F1096DJSP, R5F1096EJSP
30 pins	30-pin plastic SSOP (7.62 mm (300))	Mounted	R5F109AAJSP, R5F109ABJSP, R5F109ACJSP, R5F109ADJSP, R5F109AEJSP
32 pins	32-pin plastic WQFN fine pitch)(5 × 5)	Mounted	R5F109BAJNA, R5F109BBJNA, R5F109BCJNA, 5F109BDJNA, R5F109BEJNA
48 pins	48-pin plastic LQFP (fine pitch) (7 × 7)	Mounted	R5F109GAJFB, R5F109GBJFB, R5F109GCJFB, R5F109GDJFB, R5F109GEJFB
	48-pin plastic WQFN (7 × 7) Note	Mounted	R5F109GAJNA, R5F109GBJNA, R5F109GCJNA, R5F109GDJNA, R5F109GEJNA
64 pins	64-pin plastic LQFP (fine pitch) (10 × 10)	Mounted	R5F109LAJFB, R5F109LBJFB, R5F109LCJFB, R5F109LDJFB, R5F109LEJFB

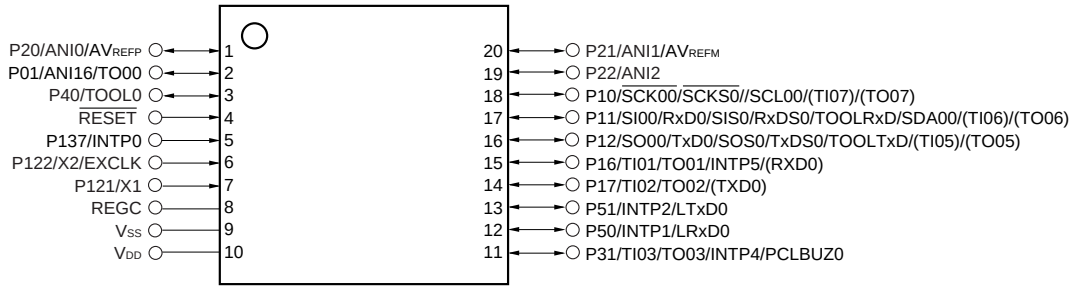
<R> **Note** Contact Renesas local sales office or sales representative for further details on this package.

Caution The RL78/F12 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.

1.3 Pin Configuration (Top View)

1.3.1 20-pin products

- 20-pin plastic SSOP (7.62 mm (300))



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

2. For the following each port, complete the following software processings before performing the operation that reads the port latch Pm having the target port latch Pm.n within 50ms after releasing reset (after staring CPU operation)

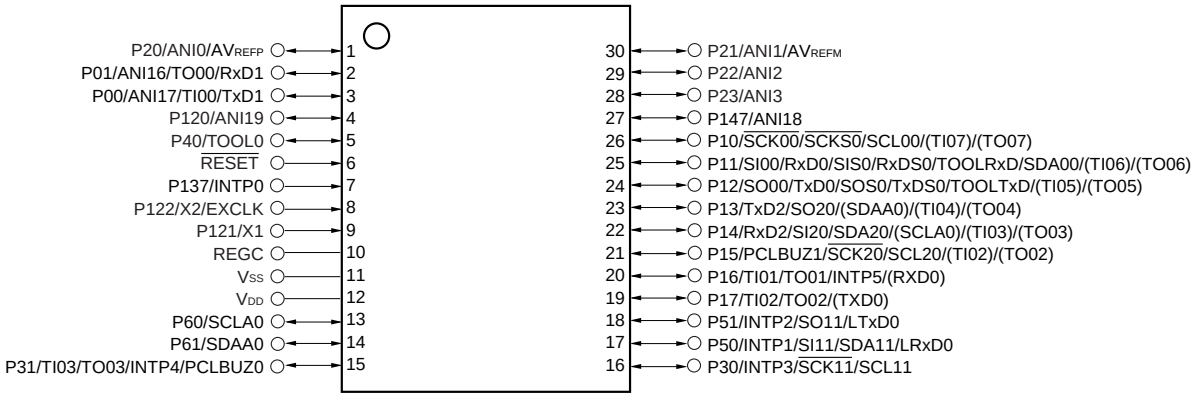
- **Set P00, P13, P14, P15, P30, P60, P61, and P147 to low level output mode by the software (clear the PMm.n and Pm.n bits for the target ports).**
- **Set P23 to digital port and low level output mode by the software (set P23 to digital mode with the ADPC register and clear the PM2.3 and P2.3 bits).**

Remarks 1. For pin identification, see **1.4 Pin Identification**.

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.3.2 30-pin products

- 30-pin plastic SSOP (7.62 mm (300))



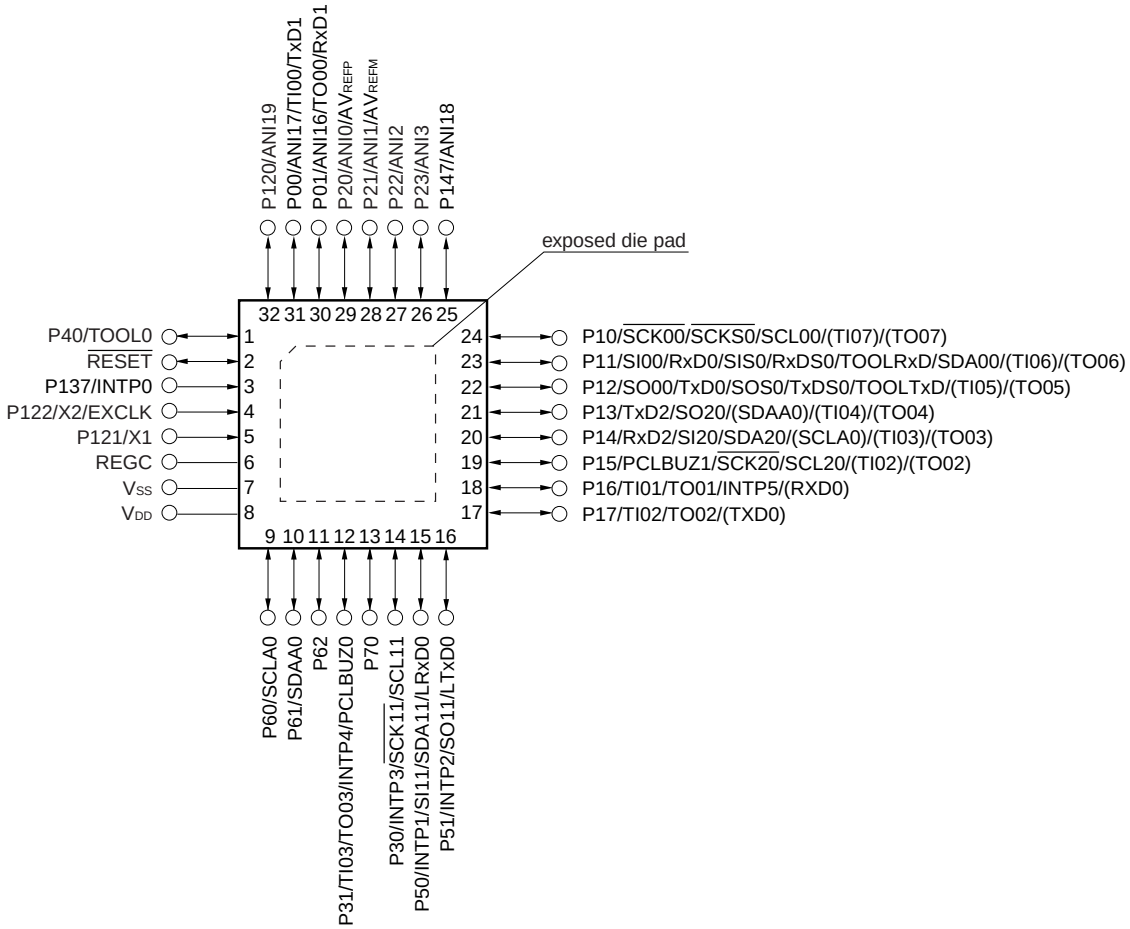
Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

Remarks 1. For pin identification, see **1.4 Pin Identification**.

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.3.3 32-pin products

- 32-pin plastic WQFN (fine pitch) (5 × 5)



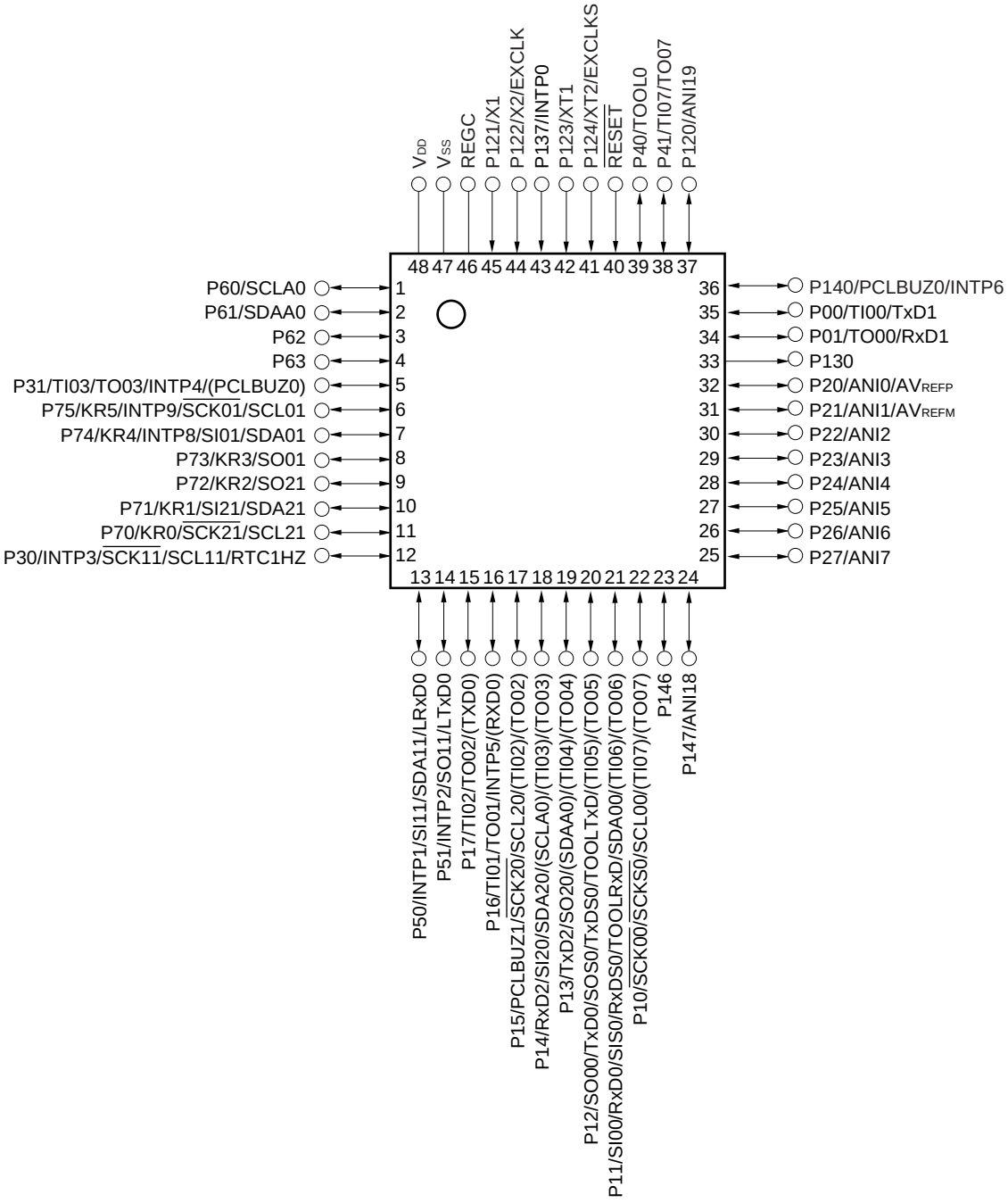
Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

Remarks 1. For pin identification, see **1.4 Pin Identification**.

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.3.4 48-pin products

- 48-pin plastic LQFP (fine pitch) (7 × 7)

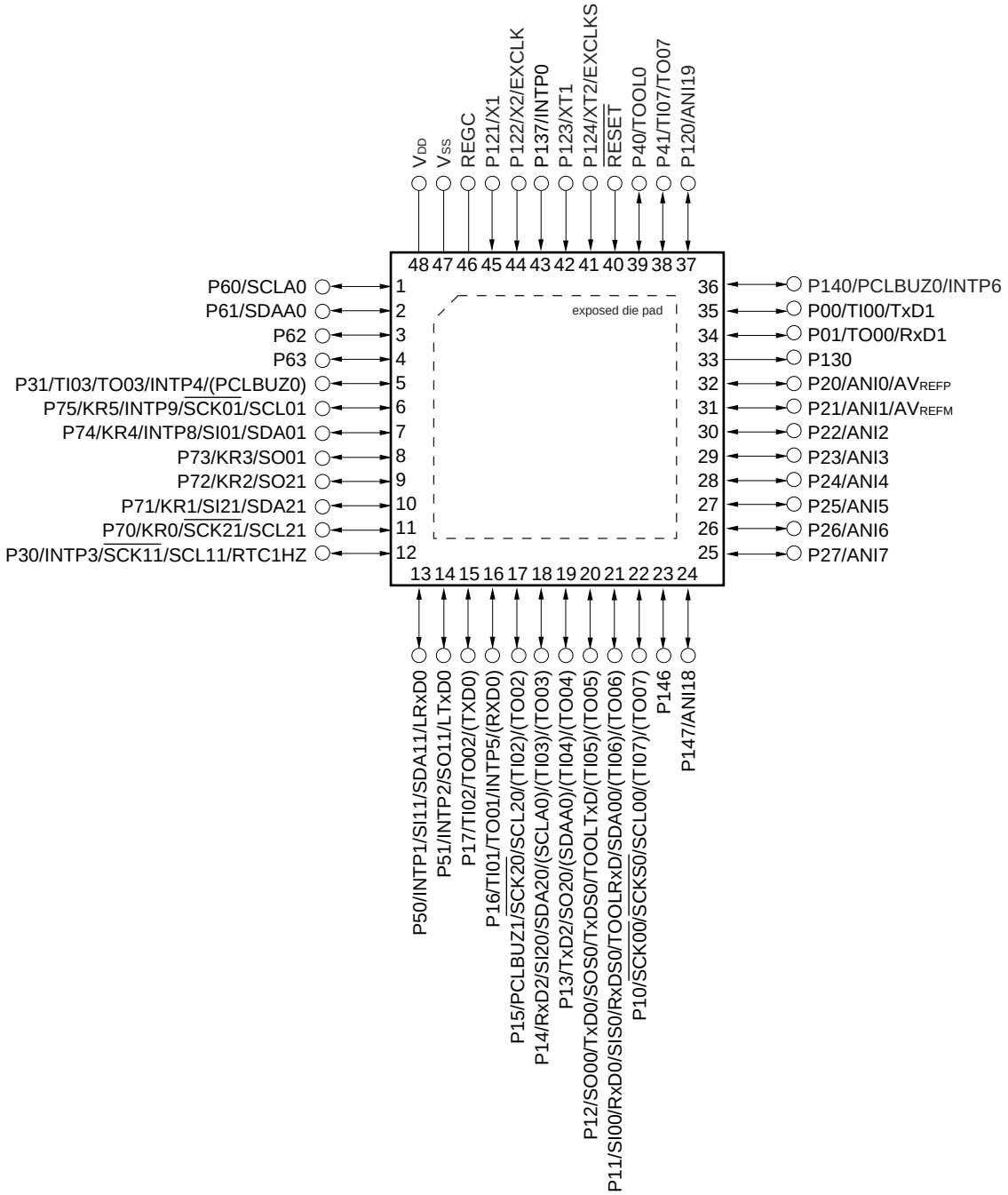


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

Remarks 1. For pin identification, see **1.4 Pin Identification**.

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

- 48-pin plastic WQFN (7 × 7)



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

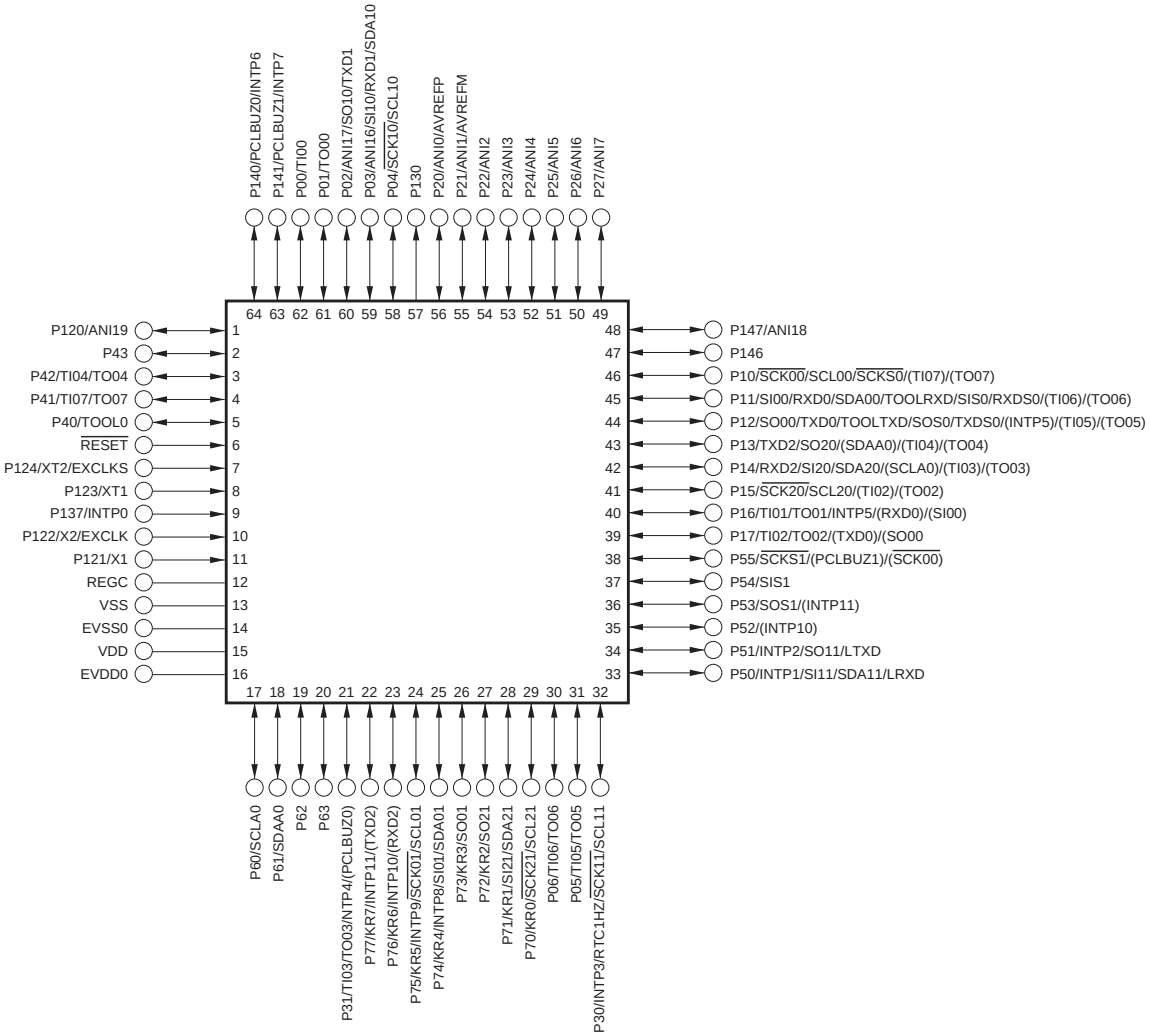
Remarks 1. For pin identification, see **1.4 Pin Identification**.

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

<R> **3.** Contact Renesas local sales office or sales representative for further details on this package.

1.3.5 64-pin products

- 64-pin plastic LQFP



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

Remarks 1. For pin identification, see **1.4 Pin Identification**.

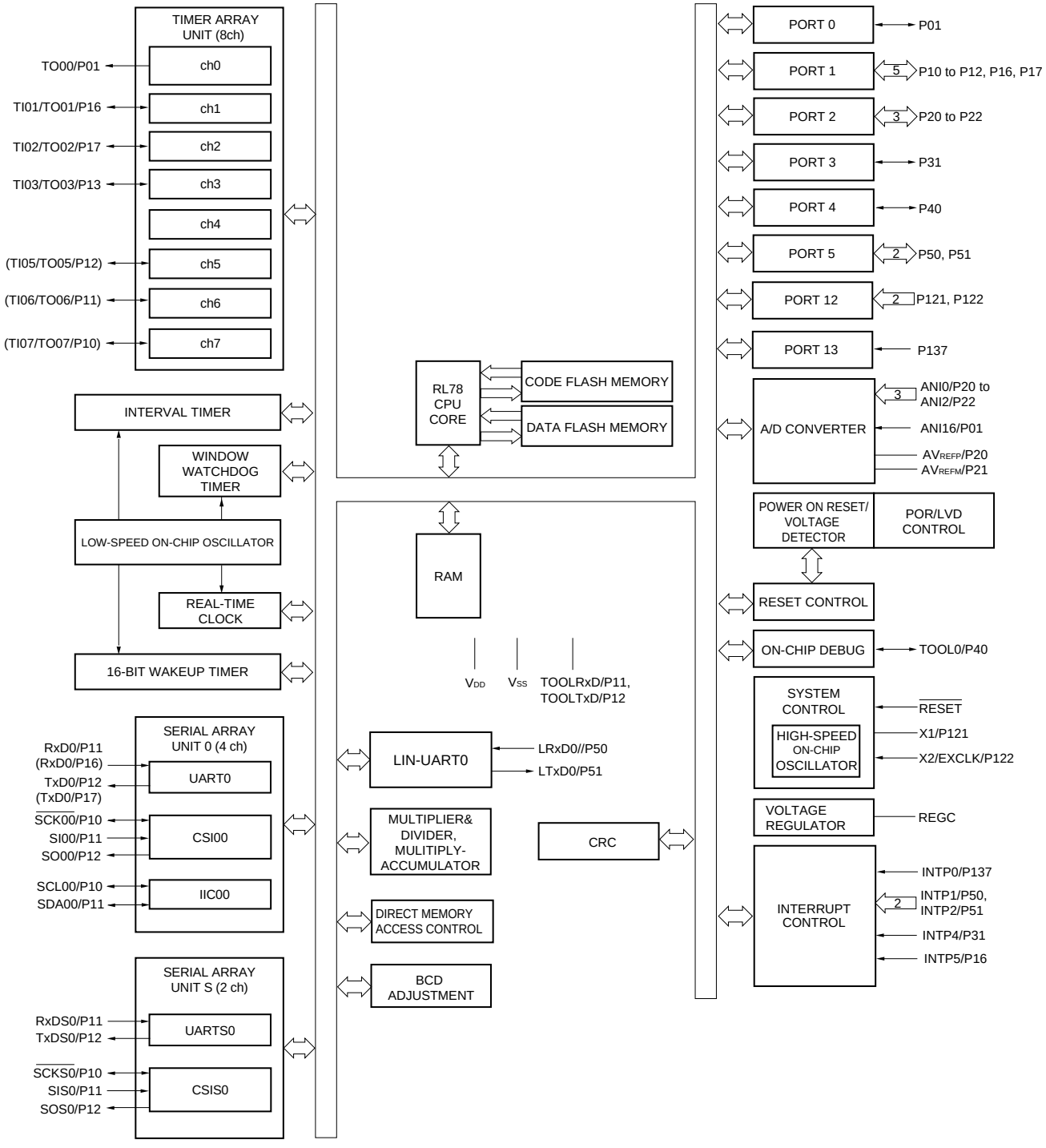
2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.4 Pin Identification

ANI0 to ANI7, ANI16 to ANI19: Analog input
AV_{REFM}: A/D converter reference potential (– side) input
AV_{REFP}: A/D converter reference potential (+ side) input
EXCLK: External clock input (main system clock)
EXCLKS: External clock input (sub system clock)
INTP0 to INTP11: External interrupt input
KR0 to KR7: Key return
LRxD0: Receive Data
LTxD0: Transmit Data
P00, P06: Port 0
P10 to P17: Port 1
P20 to P27: Port 2
P30, P31: Port 3
P40 to P43: Port 4
P50 to P55: Port 5
P60 to P63: Port 6
P70 to P77: Port 7
P120 to P124: Port 12
P130, P137: Port 13
P140, P141, P146, P147: Port 14
PCLBUZ0, PCLBUZ1: Programmable clock output/buzzer output
REGC: Regulator capacitance
$\overline{RESET}$: Reset
RTC1HZ: Real-time clock correction clock (1 Hz) output
RxD0 to RxD2, RxDS0: Receive data
$\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, $\overline{SCK11}$, $\overline{SCK20}$, $\overline{SCK21}$, $\overline{SCKS0}$, $\overline{SCKS1}$: Serial clock input/output
SCL00, SCL01, SCL10, SCL11, SCL20, SCL21, SCLA0: Serial clock input/output
SDA00, SDA01, SDA10, SDA11,SDA20, SDA21, SDAA0: Serial data input/output
SI00, SI01, SI10, SI11, SI20, SI21, SIS0, SIS1: Serial data input
SO00, SO01, SO10, SO11, SO20, SO21, SOS0, SOS1: Serial data output
TI00 to TI07: Timer input
TO00 to TO07: Timer output
TOOL0: Data input/output for tool
TOOLRxD, TOOLTxD: Data input/output for external device
TxD0 to TxD2, TxDS0: Transmit data
V_{DD}: Power supply
V_{SS}: Ground
X1, X2: Crystal oscillator (main system clock)
XT1, XT2: Crystal oscillator (subsystem clock)

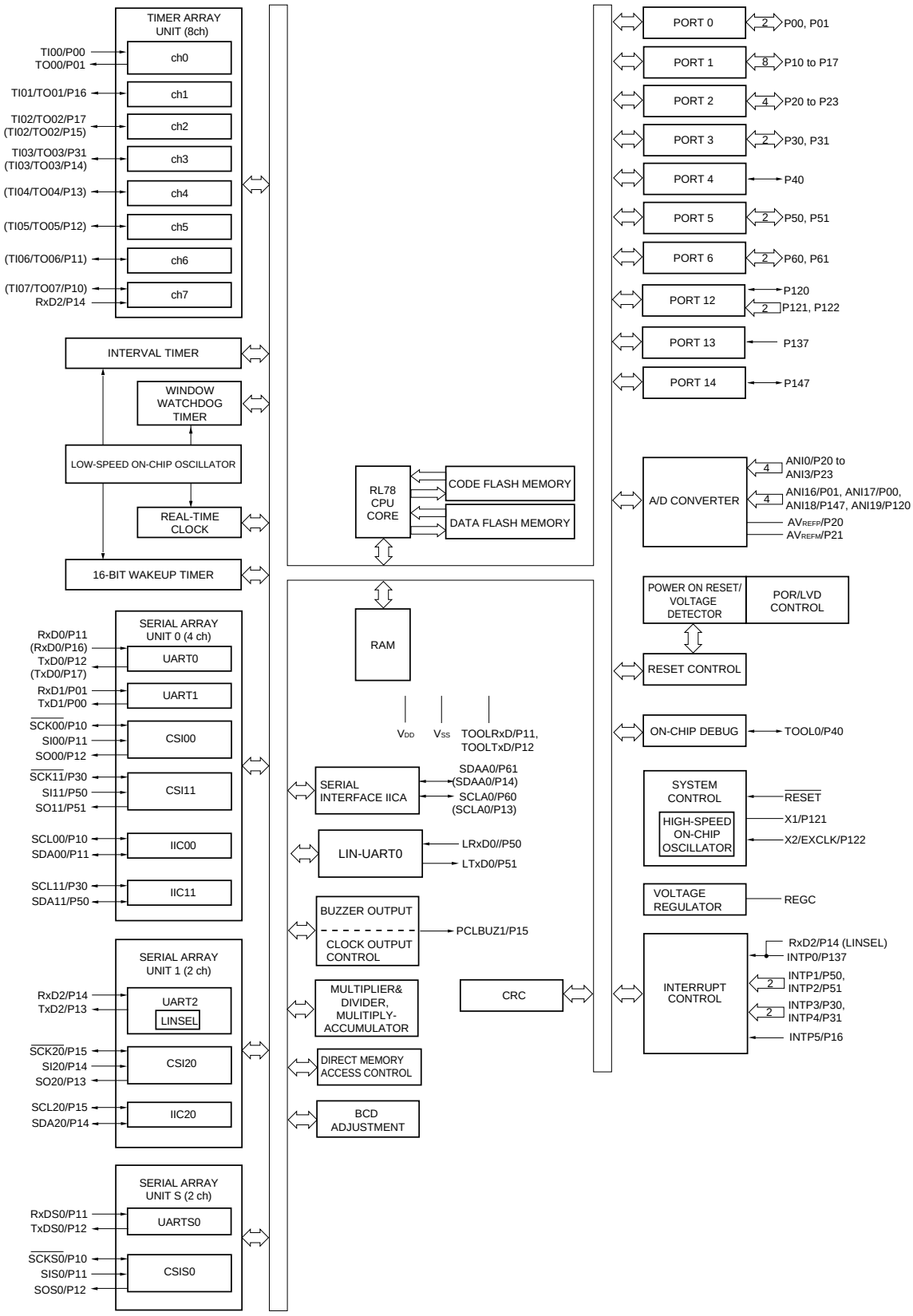
1.5 Block Diagram

1.5.1 20-pin products



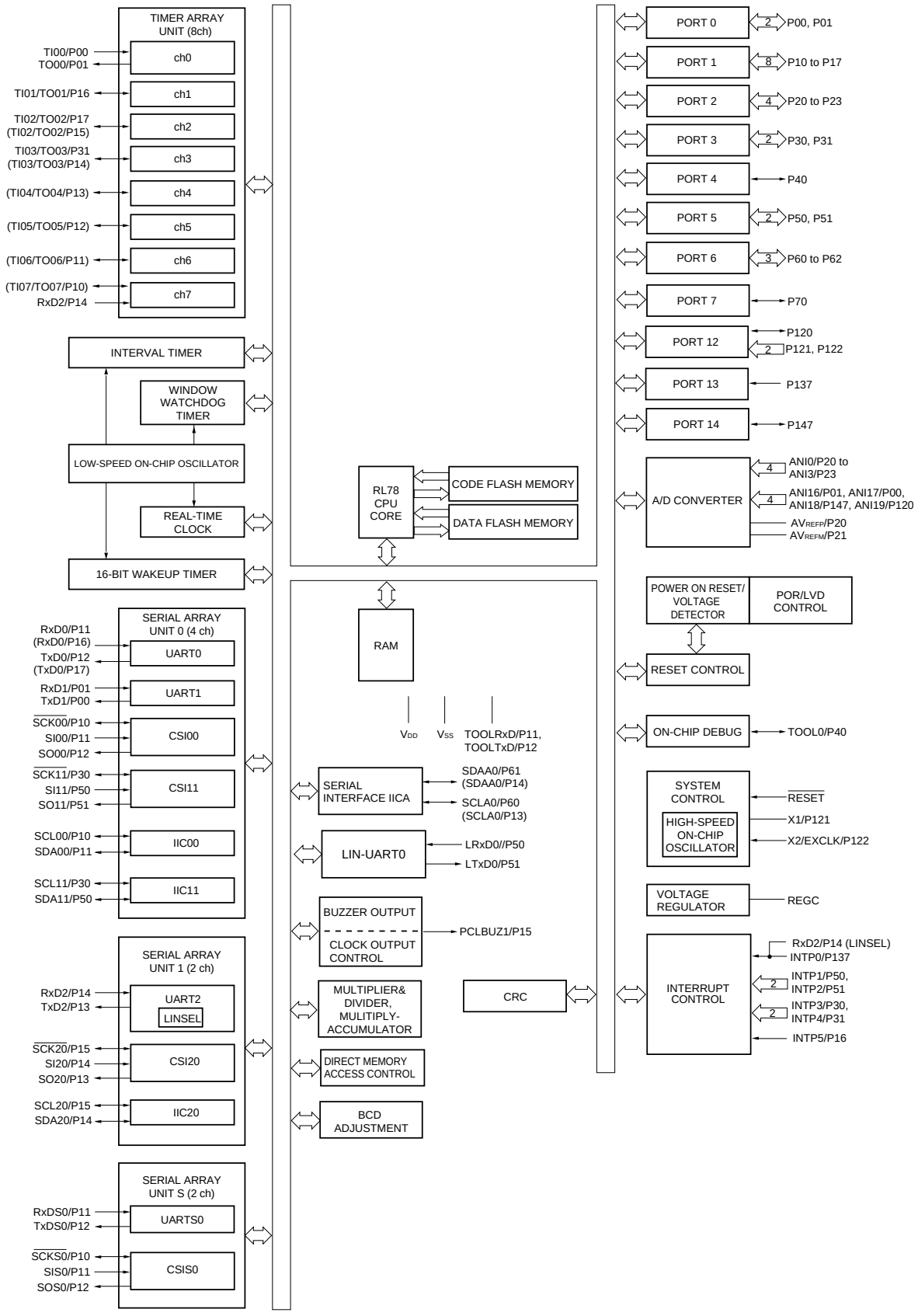
Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.5.2 30-pin products



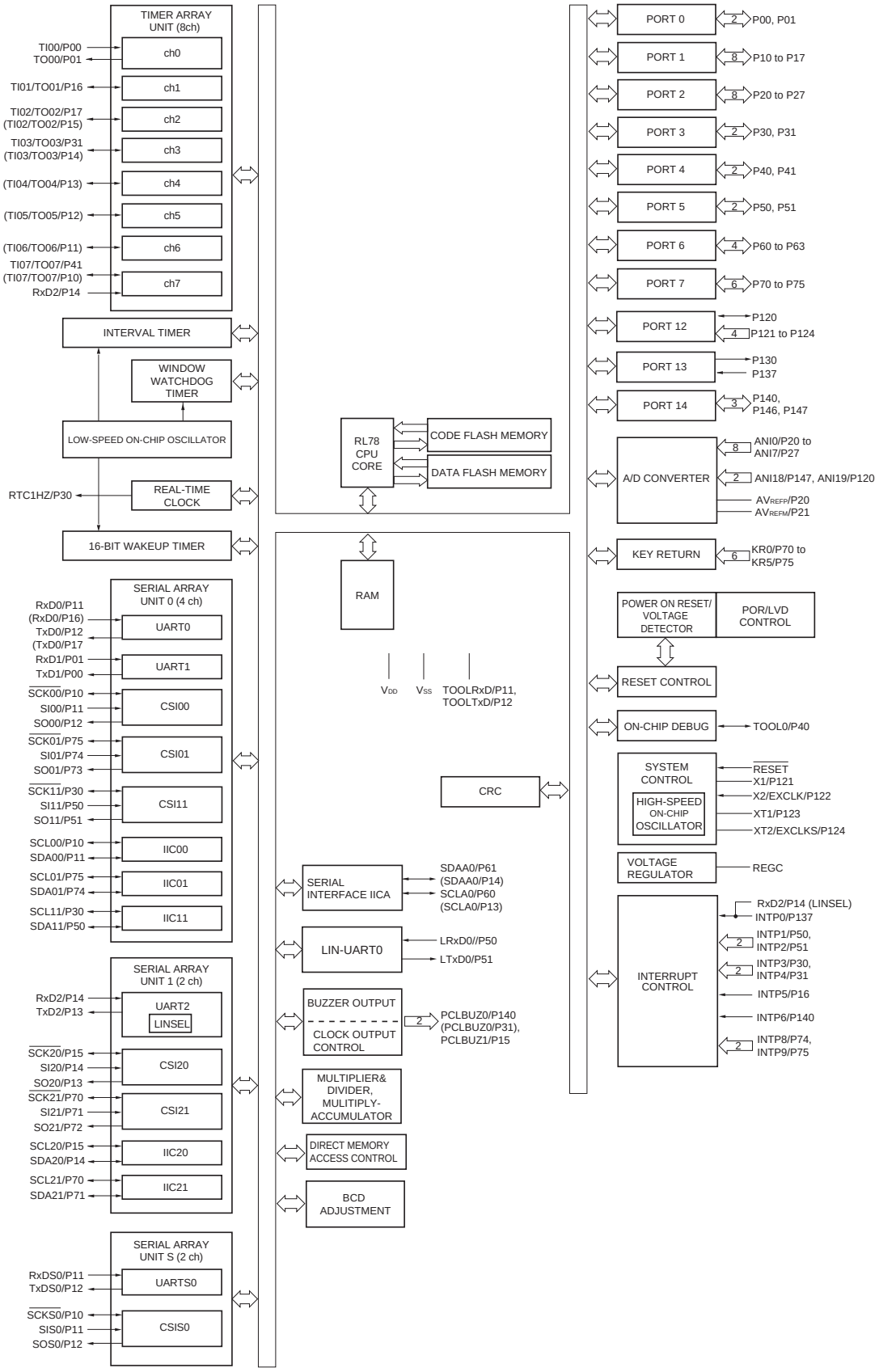
Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.5.3 32-pin products



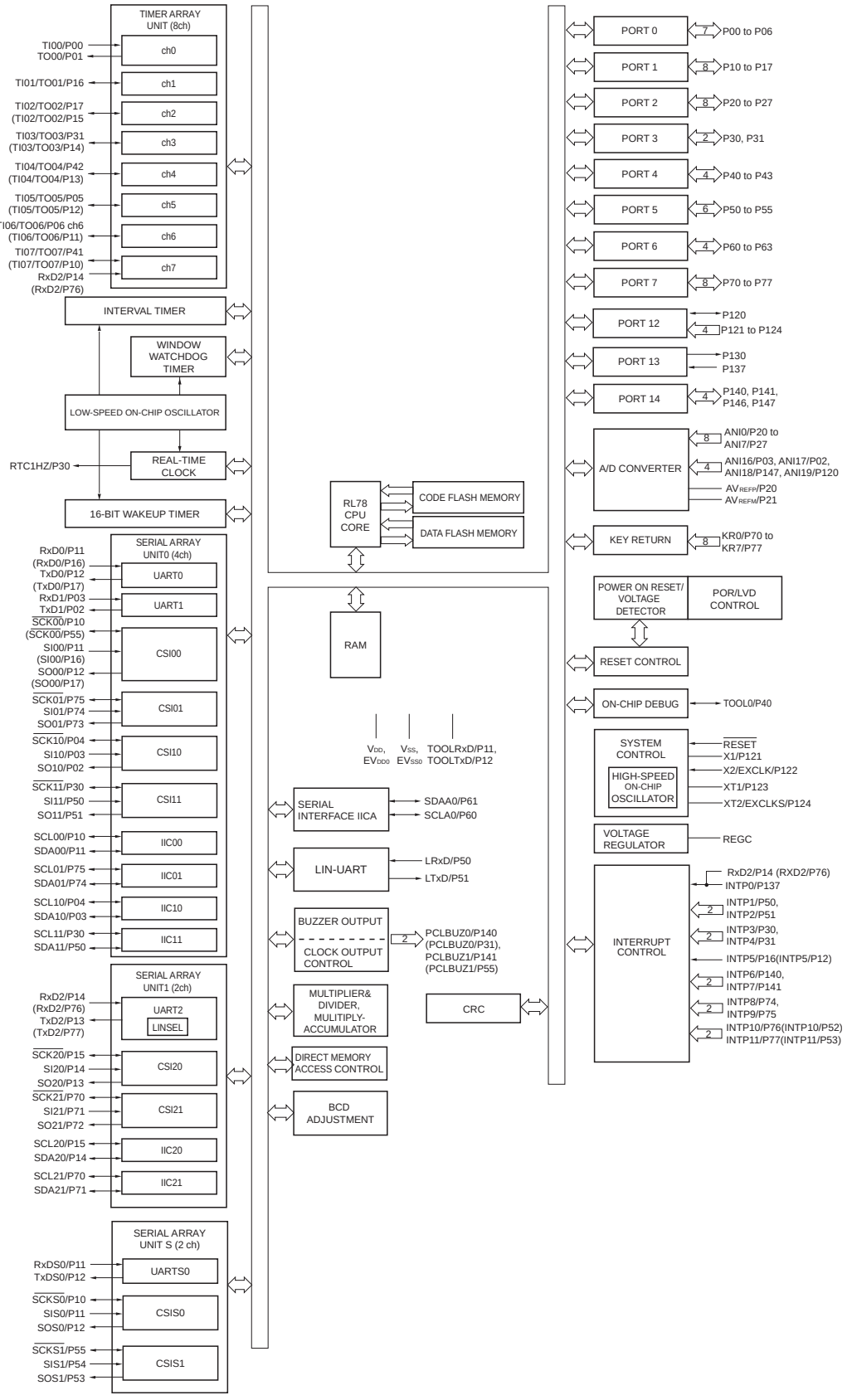
Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.5.4 48-pin products



Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.5.5 64-pin products



Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

1.6 Outline of Functions

Caution This outline describes the functions at the time when Peripheral I/O redirection register (PIOR) is set to 00H.

(1/2)

Item		20-pin	30-pin	32-pin	48-pin	64-pin
		R5F1096x	R5F109Ax	R5F109Bx	R5F109Gx	R5F109Lx
Code flash memory (KB)		8 to 64	16 to 64	16 to 64	16 to 64	16 to 64
Data flash memory (KB)		4	4	4	4	4
RAM (KB)		0.5 to 4 Note1	1 to 4 Note1	1 to 4 Note1	1 to 4 Note1	1 to 4 Note1
Memory space		1 MB				
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 1 to 20 MHz: V_{DD} = 2.7 to 5.5 V, 1 to 8 MHz: V_{DD} = 1.8 to 2.7 V				
	High-speed on-chip oscillator clock	HS (High-speed main) mode: 1 to 32 MHz (V_{DD} = 2.7 to 5.5 V), LS (Low-speed main) mode: 1 to 8 MHz (V_{DD} = 1.8 to 5.5 V)				
Subsystem clock		–			XT1 (crystal) oscillation 32.768 kHz (TYP.): V_{DD} = 1.8 to 5.5 V	
Low-speed on-chip oscillator clock		15 kHz (TYP.): V_{DD} = 1.8 to 5.5 V				
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) Note3				
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits) • Rotate, barrel shift, and bit manipulation (Set, reset, test, and Boolean operation), etc.				
I/O port	Total	16	26	28	44	58
	CMOS I/O	13	21	22	34	48
	CMOS input	3	3	3	5	5
	CMOS output	–	–	–	1	1
	N-ch open-drain I/O (6 V tolerance)	–	2	3	4	4
Timer	16-bit timer	8 channels				
	Watchdog timer	1 channel				
	Real-time clock (RTC)	1 channel				
	Interval timer	1 channel				
	Wakeup timer	1 channel				
	Timer output	4 channels (PWM outputs: 3 Note2)			5 channels (PWM outputs: 4 Note2)	8 channels (PWM outputs: 4 Note2)
	RTC output	–			1 1 Hz (subsystem clock: f_{SUB} = 32.768 kHz)	
Clock output/buzzer output		1	2	2	2	2
		• 2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (Peripheral hardware 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) Note3				

Notes 1. In the case of the 4 KB, this is 3 KB when the self-programming function is used.

2. The number of outputs varies, depending on the setting.

3. Available only in 48- and 64-pin products.

(2/2)

Item		20-pin	30-pin	32-pin	48-pin	64-pin[Note3]
		R5F1096x	R5F109Ax	R5F109Bx	R5F109Gx	R5F109Lx
8/10-bit resolution A/D converter		4 channels (V_{DD}: 3 channels) (EV_{DD}: 1 channels)	8 channels (V_{DD}: 4 channels) (EV_{DD}: 4 channels)	8 channels (V_{DD}: 4 channels) (EV_{DD}: 4 channels)	10 channels (V_{DD}: 8 channels) (EV_{DD}: 2 channels)	12 channels (V_{DD}: 8 channels) (EV_{DD}: 4 channels)
Serial interface		[20-pin, 24-pin, 25-pin products] • CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel • CSI: 1 channel/UART: 1 channel • LIN-UART: 1 channel [30-pin, 32-pin products] • CSI: 2 channels/UART: 2 channels/simplified I^2C: 2 channels • CSI: 1 channel/UART (UART supporting LIN-bus): 1 channel/simplified I^2C: 1 channel • CSI (7 to 16 bits): 1 channel/UART (7 to 9, 16 bits): 1 channel • LIN-UART: 1 channel [48-pin products] • CSI: 3 channels/UART: 2 channels/simplified I^2C: 3 channels • CSI: 2 channels/UART (UART supporting LIN-bus): 1 channel/simplified I^2C: 2 channels • CSI (7 to 16 bits): 1 channel/UART (7 to 9, 16 bits): 1 channel • LIN-UART: 1 channel [64-pin products] • CSI: 4 channels/UART: 2 channels/simplified I^2C: 4 channels • CSI: 2 channels/UART (UART supporting LIN-bus): 1 channel/simplified I^2C: 2 channels • CSI (7 to 16 bits): 2 channels/UART (7 to 9, 16 bits): 1 channel • LIN-UART: 1 channel				
	I^2C bus	–	1 channel	1 channel	1 channel	
Multiplier and divider/multiply-accumulator		• 16 bits × 16 bits = 32 bits (Unsigned or signed) • 32 bits ÷ 32 bits = 32 bits (Unsigned) • 16 bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)				
DMA controller		2 channels				
Vectored interrupt sources	Internal	28	34	34	34[Note1]	34[Note1]
	External	5	6	6	10[Note1]	12[Note1]
Key interrupt		–			6	8
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 [Note2] • Internal reset by RAM parity error • Internal reset by illegal-memory access				
Power-on-reset circuit		• Power-on-reset: 1.51 ±0.03 V • Power-down-reset: 1.50 ±0.03 V				
Voltage detector		• Rising edge : 1.88 V to 4.06 V (12 stages) • Falling edge : 1.84 V to 3.98 V (12 stages)				
On-chip debug function		Provided				
Power supply voltage		V_{DD} = 1.6 to 5.5 V				
Operating ambient temperature		T_A = –40 to +85 °C				

Notes 1. INTP8, INTLR, INTP9, and INTLS are counted as one interrupt source in both an internal and external interrupt, respectively.

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.

3. Available only in 48- and 64-pin products.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

2.1.1 20-pin products

Function Name	I/O	Function	After Reset	Alternate Function
P01	I/O	Port 0. 1-bit I/O port. Input of P01 can be set to TTL input buffer. P01 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI16/TO00
P10	I/O	Port 1. 5-bit I/O port. Input of P16 and P17 can be set to TTL input buffer. Output of P10 to P12, and P17 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK00}$/$\overline{SCKS0}$/ SCL00/(TI07)/(TO07)
P11				SI00/RxD0/ SIS0/RxDS0/ TOOLRxD/SDA00/ (TI06)/(TO06)
P12				SO00/TxD0/SOS0/ TxDS0/TOOLTxD/ (TI05)/(TO05)
P16				TI01/TO01/INTP5/ (RXD0)
P17				TI02/TO02/(TXD0)
P20	I/O	Port 2. 3-bit I/O port. Input/output can be specified in 1-bit units.	Analog input port	ANI0/AV_{REFP}
P21				ANI1/AV_{REFM}
P22				ANI2
P31	I/O	Port 3. 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.	Input port	TI03/TO03/INTP4 PCLBUZ0
P40	I/O	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.	Input port	TOOL0
P50	I/O	Port 5. 2-bit I/O port. Output of P50 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1/LRxD0
P51				INTP2/LTxD0
P121	Input	Port 12. 2-bit input port.	Input port	X1
P122				X2/EXCLK
P137	Input	Port 13. 1-bit input port.	Input port	INTP0

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

2.1.2 30-pin products

(1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 2-bit I/O port. Input of P01 can be set to TTL input buffer. Output of P00 can be set to N-ch open-drain output (V_{DD} tolerance). P00 and P01 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI17/TI00/TxD1
P01				ANI16/TO00/RxD1
P10	I/O	Port 1. 8-bit I/O port. Input of P13 to P17 can be set to TTL input buffer. Output of P10 to P15, and P17 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK00}$/$\overline{SCKS0}$/ SCL00/(TI07)/(TO07)
P11				SI00/RxD0/ SIS0/RxDS0/ TOOLRxD/SDA00/ (TI06)/(TO06)
P12				SO00/TxD0/SOS0/ TxDS0/TOOLTxD/ (TI05)/(TO05)
P13				TxD2/SO20/(SDAA0)/ (TI04)/(TO04)
P14				RxD2/SI20/SDA20/ (SCLA0)/(TI03)/ (TO03)
P15				PCLBUZ1/$\overline{SCK20}$/ SCL20/(TI02)/(TO02)
P16				TI01/TO01/INTP5/ (RXD0)
P17				TI02/TO02/(TXD0)
P20	I/O	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units.	Analog input port	ANI0/AV_{REFP}
P21				ANI1/AV_{REFM}
P22				ANI2
P23				ANI3
P30	I/O	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.	Input port	INTP3/ $\overline{SCK11}$/SCL11
P31				TI03/TO03/INTP4
P40	I/O	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.	Input port	TOOL0
P50	I/O	Port 5. 2-bit I/O port. Output of P50 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1/SI11/SDA11/ LRxD0
P51				INTP2/SO11/LTxD0

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

(2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P60	I/O	Port 6. 2-bit I/O port. Output of P60 and P61 can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCLA0
P61				SDAA0
P120	I/O	Port 12. 1-bit I/O port and 2-bit input port. P120 can be set to analog input. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI19
P121	Input		Input port	X1
P122				X2/EXCLK
P137	Input	Port 13. 1-bit input port.	Input port	INTP0
P147	I/O	Port 14. 1-bit I/O port. P147 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI18

2.1.3 32-pin products

(1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 2-bit I/O port. Input of P01 can be set to TTL input buffer. Output of P00 can be set to N-ch open-drain output (V_{DD} tolerance) P00 and P01 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI17/TI00/TxD1
P01				ANI16/TO00/RxD1
P10	I/O	Port 1. 8-bit I/O port. Input of P13 to P17 can be set to TTL input buffer. Output of P10 to P15, and P17 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK00}$/$\overline{SCKS0}$/ SCL00/(TI07)/(TO07)
P11				SI00/RxD0/ SIS0/RxDS0/ TOOLRxD/SDA00/ (TI06)/(TO06)
P12				SO00/TxD0/SOS0/ TxDS0/TOOLTxD/ (TI05)/(TO05)
P13				TxD2/SO20/(SDAA0)/ (TI04)/(TO04)
P14				RxD2/SI20/SDA20/ (SCLA0)/(TI03)/ (TO03)
P15				PCLBUZ1/$\overline{SCK20}$/ SCL20/(TI02)/(TO02)
P16				TI01/TO01/INTP5/ (RXD0)
P17				TI02/TO02/(TXD0)
P20	I/O	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units.	Analog input port	ANI0/AV_{REFP}
P21				ANI1/AV_{REFM}
P22				ANI2
P23				ANI3
P30	I/O	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.	Input port	INTP3/$\overline{SCK11}$/ SCL11
P31				TI03/TO03/INTP4
P40	I/O	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.	Input port	TOOL0
P50	I/O	Port 5. 2-bit I/O port. Output of P50 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1/SI11/SDA11/ LRxD0
P51				INTP2/SO11/LTxD0

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

(2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P60	I/O	Port 6. 3-bit I/O port. Output of P60 to P62 can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCLA0
P61				SDAA0
P62				–
P70	I/O	Port 7. 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.	Input port	–
P120	I/O	Port 12. 1-bit I/O port and 2-bit input port. P120 can be set to analog input. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI19
P121	Input		Input port	X1
P122				X2/EXCLK
P137	Input	Port 13. 1-bit input port.	Input port	INTP0
P147	I/O	Port 14. 1-bit I/O port. P147 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI18

2.1.4 48-pin products

(1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 2-bit I/O port. Input of P01 can be set to TTL input buffer. Output of P00 can be set to N-ch open-drain output (V_{DD} tolerance) Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI00/TxD1
P01				TO00/RxD1
P10	I/O	Port 1. 8-bit I/O port. Input of P13 to P17 can be set to TTL input buffer. Output of P10 to P15, and P17 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK00}$/$\overline{SCKS0}$/ SCL00/(TI07)/(TO07)
P11				SI00/RxD0/ SIS0/RxDS0/ TOOLRxD/SDA00/ (TI06)/(TO06)
P12				SO00/TxD0/SOS0/ TxDS0/TOOLTxD/ (TI05)/(TO05)
P13				TxD2/SO20/(SDAA0)/ (TI04)/(TO04)
P14				RxD2/SI20/SDA20/ (SCLA0)/(TI03)/ (TO03)
P15				PCLBUZ1/$\overline{SCK20}$/ SCL20/(TI02)/ (TO02)
P16				TI01/TO01/INTP5/ (RXD0)
P17				TI02/TO02/(TXD0)
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Analog input port	ANI0/AV_{REFP}
P21				ANI1/AV_{REFM}
P22				ANI2
P23				ANI3
P24				ANI4
P25				ANI5
P26				ANI6
P27				ANI7
P30	I/O	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.	Input port	INTP3/RTC1HZ/ $\overline{SCK11}$/SCL11
P31				TI03/TO03/INTP4/ (PCLBUZ0)
P40	I/O	Port 4. 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.	Input port	TOOL0
P41				TI07/TO07
				–

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

(2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P50	I/O	Port 5. 2-bit I/O port. Output of P50 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1/SI11/SDA11/ LRxD0
P51				INTP2/SO11/LTxD0
P60	I/O	Port 6. 4-bit I/O port. Output of P60 to P63 can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCLA0
P61				SDAA0
P62				–
P63				–
P70	I/O	Port 7. 6-bit I/O port. Output of P71 and P74 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	KR0/$\overline{\text{SCK21}}$/SCL21
P71				KR1/SI21/SDA21
P72				KR2/SO21
P73				KR3/SO01
P74				KR4/INTP8/SI01/ SDA01
P75				KR5/INTP9/$\overline{\text{SCK01}}$/ SCL01
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. P120 can be set to analog input. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI19
P121	Input		Input port	X1
P122				X2/EXCLK
P123				XT1
P124				XT2/EXCLKS
P130	Output	Port 13. 1-bit output port and 1-bit input port.	Output port	–
P137	Input		Input port	INTP0
P140	I/O	Port 14. 3-bit I/O port. P147 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	PCLBUZ0/INTP6
P146				–
P147			Analog input port	ANI18

2.1.5 64-pin products

(1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 7-bit I/O port. Input of P01, P03 and P04 can be set to TTL input buffer. Output of P00, P02, P03 and P04 can be set to N-ch open-drain output (V_{DD} tolerance) Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI00
P01				TO00
P02				ANI17/SO10/TXD1
P03				ANI16/SI10/RXD1/SDA10
P04				$\overline{SCK10}$/SCL10
P05				TI05/TO05
P06				TI06/TO06
P10	I/O	Port 1. 8-bit I/O port. Input of P13 to P17 can be set to TTL input buffer. Output of P10 to P15, and P17 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK00}$/SCL00/$\overline{SCKS0}$/(TI07)/(TO07)
P11				SI00/RXD0/SDA00/TOOLRXD/SIS0/RXDS0/(TI06)/(TO06)
P12				SO00/TXD0/TOOLTXD/SOS0/TXDS0/(INTP5)/(TI05)/(TO05)
P13				TXD2/SO20/(SDAA0)/(TI04)/(TO04)
P14				RXD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)
P15				$\overline{SCK20}$/SCL20/(TI02)/(TO02)
P16				TI01/TO01/INTP5/(RXD0)/(SI00)
P17				TI02/TO02/(TXD0)/(SO00)
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Analog input port	ANI0/AV_{REFP}
P21				ANI1/AV_{REFM}
P22				ANI2
P23				ANI3
P24				ANI4
P25				ANI5
P26				ANI6
P27				ANI7
P30	I/O	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.	Input port	INTP3/RTC1HZ/$\overline{SCK11}$/SCL11
P31				TI03/TO03/INTP4/(PCLBUZ0)
P40	I/O	Port 4. 4-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.	Input port	TOOL0
P41				TI07/TO07
P42				TI04/TO04
P43				–

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

(2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P50	I/O	Port 5. 6-bit I/O port. Input of P55 can be set to TTL input buffer. Output of P50 and P55 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1/SI11/SDA11/LRXD
P51				INTP2/SO11/LTXD
P52				(INTP10)
P53				SOS1/(INTP11)
P54				SIS1
P55				$\overline{SCKS1}$/(PCLBUZ1)/($\overline{SCK00}$)
P60	I/O	Port 6. 4-bit I/O port. Output of P60 to P63 can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCLA0
P61				SDAA0
P62				–
P63				–
P70	I/O	Port 7. 8-bit I/O port. Output of P71 and P74 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	KR0/$\overline{SCK21}$/SCL21
P71				KR1/SI21/SDA21
P72				KR2/SO21
P73				KR3/SO01
P74				KR4/INTP8/SI01/SDA01
P75				KR5/INTP9/$\overline{SCK01}$/SCL01
P76				KR6/INTP10/(RXD2)
P77				KR7/INTP11/(TXD2)
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. P120 can be set to analog input. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting.	Analog input port	ANI19
P121	Input		Input port	X1
P122				X2/EXCLKS
P123				XT1
P124				XT2/EXCLKS
P130	Output	Port 13. 1-bit output port and 1-bit input port.	Output port	–
P137	Input		Input port	INTP0
P140	I/O	Port 14. 4-bit I/O port. P147 can be set to analog input. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	PCLBUZ0/INTP6
P141				PCLBUZ1/INTP7
P146				–
P147			Analog input port	ANI18

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

2.1.6 Pins for each product (pins other than port pins)

(1/3)

Function Name	I/O	Function	64-pin	48-pin	32-pin	30-pin	20-pin
ANI0	Input	A/D converter analog input	√	√	√	√	√
ANI1			√	√	√	√	√
ANI2			√	√	√	√	√
ANI3			√	√	√	√	–
ANI4			√	√	–	–	–
ANI5			√	√	–	–	–
ANI6			√	√	–	–	–
ANI7			√	√	–	–	–
ANI16			√	–	√	√	√
ANI17			√	–	√	√	–
ANI18			√	√	√	√	–
ANI19			√	√	√	√	–
INTP0	Input	External interrupt request input	√	√	√	√	√
INTP1			√	√	√	√	√
INTP2			√	√	√	√	√
INTP3			√	√	√	√	–
INTP4			√	√	√	√	√
INTP5			√	√	√	√	√
INTP6			√	√	–	–	–
INTP8			√	√	–	–	–
INTP9			√	√	–	–	–
INTP10			√	–	–	–	–
INTP11			√	–	–	–	–
KR0	Input	Key interrupt input	√	√	–	–	–
KR1			√	√	–	–	–
KR2			√	√	–	–	–
KR3			√	√	–	–	–
KR4			√	√	–	–	–
KR5			√	√	–	–	–
KR6			√	–	–	–	–
KR7			√	–	–	–	–
LRxD0	Input	Serial data input to LIN-UART0	√	√	√	√	√
LTxD0	Output	Serial data output from LIN-UART0	√	√	√	√	√
PCLBUZ0	Output	Clock output/buzzer output	√	√	√	√	√
PCLBUZ1			√	√	√	√	–
REGC	–	Connecting regulator output stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	√	√	√	√	√
$\overline{\text{RESET}}$	Input	System reset input	√	√	√	√	√

(2/3)

Function Name	I/O	Function	64-pin	48-pin	32-pin	30-pin	20-pin
RxD0	Input	Serial data input to UART0	√	√	√	√	√
RxD1		Serial data input to UART1	√	√	√	√	–
RxD2		Serial data input to UART2	√	√	√	√	–
RxDS0		Serial data input to UARTS0	√	√	√	√	√
$\overline{SCK00}$	I/O	Clock input/output for CSI00, CSI01, CSI10, CSI11, CSI20, CSI21,CSIS0 and CSIS1	√	√	√	√	√
$\overline{SCK01}$			√	√	–	–	–
$\overline{SCK10}$			√	–	–	–	–
$\overline{SCK11}$			√	√	√	√	–
$\overline{SCK20}$			√	√	√	√	–
$\overline{SCK21}$			√	√	–	–	–
$\overline{SCKS0}$			√	√	√	√	√
$\overline{SCKS1}$			√	–	–	–	–
SCLA0	I/O	Clock input/output for I^2C	√	√	√	√	–
SCL00	I/O	Clock input/output for simplified I^2C	√	√	√	√	√
SCL01			√	√	–	–	–
SCL10			√	–	–	–	–
SCL11			√	√	√	√	–
SCL20			√	√	√	√	–
SCL21			√	√	–	–	–
SDAA0	I/O	Serial data I/O for I^2C	√	√	√	√	–
SDA00	I/O	Serial data I/O for simplified I^2C	√	√	√	√	√
SDA01			√	√	–	–	–
SDA10			√	–	–	–	–
SDA11			√	√	√	√	–
SDA20			√	√	√	√	–
SDA21			√	√	–	–	–
SI00	Input	Serial data input to CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSIS0, and CSIS1	√	√	√	√	√
SI01			√	√	–	–	–
SI10			√	–	–	–	–
SI11			√	√	√	√	–
SI20			√	√	√	√	–
SI21			√	√	–	–	–
SIS0			√	√	√	√	√
SIS1			√	–	–	–	–
SO00	Output	Serial data output from CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSIS0, and CSIS1	√	√	√	√	√
SO01			√	√	–	–	–
SO10			√	–	–	–	–
SO11			√	√	√	√	–
SO20			√	√	√	√	–
SO21			√	√	–	–	–
SOS0			√	√	√	√	√
SOS1			√	–	–	–	–

(3/3)

Function Name	I/O	Function	64-pin	48-pin	32-pin	30-pin	20-pin
TI00	Input	External count clock input to 16-bit timer 00	√	√	√	√	–
TI01		External count clock input to 16-bit timer 01	√	√	√	√	√
TI02		External count clock input to 16-bit timer 02	√	√	√	√	√
TI03		External count clock input to 16-bit timer 03	√	√	√	√	√
TI04		External count clock input to 16-bit timer 04	√	(√)	(√)	(√)	–
TI05		External count clock input to 16-bit timer 05	√	(√)	(√)	(√)	(√)
TI06		External count clock input to 16-bit timer 06	√	(√)	(√)	(√)	(√)
TI07		External count clock input to 16-bit timer 07	√	√	(√)	(√)	(√)
TO00	Output	16-bit timer 00 output	√	√	√	√	√
TO01		16-bit timer 01 output	√	√	√	√	√
TO02		16-bit timer 02 output	√	√	√	√	√
TO03		16-bit timer 03 output	√	√	√	√	√
TO04		16-bit timer 04 output	√	(√)	(√)	(√)	–
TO05		16-bit timer 05 output	√	(√)	(√)	(√)	(√)
TO06		16-bit timer 06 output	√	(√)	(√)	(√)	(√)
TO07		16-bit timer 07 output	√	√	(√)	(√)	(√)
TxD0	Output	Serial data output from UART0	√	√	√	√	√
TxD1		Serial data output from UART1	√	√	√	√	–
TxD2		Serial data output from UART2	√	√	√	√	–
TxDS0		Serial data output from UARTS0	√	√	√	√	√
X1	Input	Resonator connection for main system clock	√	√	√	√	√
X2	Output		√	√	√	√	√
EXCLK	Input	External clock input for main system clock	√	√	√	√	√
EXCLKS	Input	External clock input for subsystem clock	√	√	–	–	–
XT1	Input	Resonator connection for subsystem clock	√	√	–	–	–
XT2	Output		√	√	–	–	–
V_{DD}	–	Positive power supply for all pins	√	√	√	√	√
EV_{DD}	–	Positive power supply for pins other than above-mentioned V_{DD} connected pins	√	–	–	–	–
AV_{REFP}	Input	A/D converter reference potential (+ side) input	√	√	√	√	√
AV_{REFM}	Input	A/D converter reference potential (– side) input	√	√	√	√	√
V_{SS}	–	Ground potential for all pins	√	√	√	√	√
EV_{SS}	–	Ground potential for pins other than above-mentioned V_{SS} connected pins	√	–	–	–	–
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programming	√	√	√	√	√
TOOLTxD	Output	UART transmission pin for the external device connection used during flash memory programming	√	√	√	√	√
TOOL0	I/O	Data I/O for flash memory programmer/debugger	√	√	√	√	√

Remark The checked function is available only when the bit corresponding to the function in the peripheral I/O redirection register (PIOR) is set to 1.

2.2 Description of Pin Functions

Remark The pins mounted depend on the product. See **1.3 Pin Configuration (Top View)** and **2.1 Pin Function List**.

2.2.1 P00 to P06 (port 0)

P00 to P06 function as an I/O port. These pins also function as timer I/O, A/D converter analog input, serial interface data I/O, and clock I/O.

Input to the P01, P03, 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 P00 and P02 to P04 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units, using port output mode register 0 (POM0).

Input to the P00 to P03 pins can be specified as analog input or digital input in 1-bit units, using port mode control register 0 (PMC0).

The following operation modes can be specified in 1-bit units.

(1) Port mode

P00, P01 function as an I/O port. P00, P01 can be set to input or output port in 1-bit units using port mode register 0 (PM0). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

(2) Control mode

P00, P01 function as timer I/O, A/D converter analog input, serial interface data I/O, and clock I/O.

(a) ANI16, ANI17

These are the analog input pins (ANI16, ANI17) of A/D converter.

When using these pins as analog input pins, see **12.10 (5) Analog input (ANIn) pins**.

(b) TI00

This is the pin for inputting an external count clock/capture trigger to 16-bit timer 00.

(c) TO00

This is the timer output pins of 16-bit timer 00.

(d) TxD1

This is a serial data output pin of serial interface UART1.

(e) RxD1

This is a serial data input pin of serial interface UART1.

2.2.2 P10 to P17 (port 1)

P10 to P17 function as an I/O port. These pins also function as serial interface data I/O, clock I/O, programming UART I/O, timer I/O, clock/buzzer output, and external interrupt request input.

Input to the P13 to P17 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units, using port input mode register 1 (PIM1).

Output from the P10 to P15 and P17 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units, using port output mode register 1 (POM1).

The following operation modes can be specified in 1-bit units.

(1) Port mode

P10 to P17 function as an I/O port. P10 to P17 can be set to input or output port in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

(2) Control mode

P10 to P17 function as serial interface data I/O, clock I/O, programming UART I/O, timer I/O, clock/buzzer output, and external interrupt request input.

(a) INTP5

This is an external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) TxD0, TxD2, TxDS0

These are the serial data output pins of serial interface UART0, UART2 and UARTS0.

(c) RxD0, RxD2, RxDS0

These are the serial data input pins of serial interface UART0, UART2 and UARTS0.

(d) $\overline{\text{SCK00}}$, $\overline{\text{SCK20}}$, $\overline{\text{SCKS0}}$

These are the serial clock I/O pins of serial interface CSI00, CSI20 and CSIS0.

(e) SI00, SI20, SIS0

These are the serial data input pins of serial interface CSI00, CSI20 and CSIS0.

(f) SO00, SO11, SO20

These are the serial data output pins of serial interface CSI00, CSI20 and CSIS0.

(g) SDA00, SDA20

These are the serial data I/O pins of serial interface for simplified I^2C.

(h) SCL00, SCL20

These are the serial clock I/O pins of serial interface for simplified I^2C.

(i) TI01, TI02

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 01 and 02.

(j) TO01, TO02

These are the timer output pins of 16-bit timers 01 and 02.

(k) TOOLTxD

This UART serial data output pin for an external device connection is used during flash memory programming.

(l) TOOLRxD

This UART serial data input pin for an external device connection is used during flash memory programming.

2.2.3 P20 to P27 (port 2)

P20 to P27 function as an I/O port. These pins also function as A/D converter analog input and reference voltage input.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P20 to P27 function as an I/O port. P20 to P27 can be set to input or output port in 1-bit units using port mode register 2 (PM2).

(2) Control mode

P20 to P27 function as A/D converter analog input and reference voltage input.

(a) ANI0 to ANI7

These are the analog input pins (ANI0 to ANI7) of A/D converter.
When using these pins as analog input pins, see **12.10 (5) Analog input (ANIn) pins**.

(b) AV_{REFP}

This is a pin that inputs the A/D converter reference potential (+ side).

(c) AV_{REFM}

This is a pin that inputs the A/D converter reference potential (–side).

2.2.4 P30, P31 (port 3)

P30, P31 function as an I/O port. These pins also function as external interrupt request input, real-time clock correction clock output, serial interface clock I/O, and timer I/O.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P30, P31 function as an I/O port. P30 and P31 can be set to input or output port in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

(2) Control mode

P30, P31 function as external interrupt request input, real-time clock correction clock output, serial interface clock I/O, and timer I/O.

(a) INTP3, INTP4

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) $\overline{SCK11}$

This is a serial clock I/O pin of serial interface CSI11.

(c) SCL11

This is a serial clock output pin of serial interface for simplified I^2C.

(e) TI03

This is a pin for inputting an external count clock/capture trigger to 16-bit timer 03.

(f) TO03

This is a timer output pin from 16-bit timer 03.

(g) RTC1HZ

This is a real-time clock correction clock (1 Hz) output pin.

2.2.5 P40 to P43 (port 4)

P40 to P43 function as an I/O port. These pins also function as data I/O for a flash memory programmer/debugger, and timer I/O.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P40 to P43 function as an I/O port. P40 to P43 can be set to input or output port in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

Be sure to connect an external pull-up resistor to P40 when on-chip debugging is enabled (by using an option byte).

(2) Control mode

P40 to P43 function as data I/O for a flash memory programmer/debugger and timer I/O.

(a) TI04, TI07

This is the pin for inputting an external count clock/capture trigger to 16-bit timer 04, 07.

(b) TO04, TO07

This is the timer output pin from 16-bit timer 04, 07.

(c) TOOL0

This is a data I/O pin for a flash memory programmer/debugger.

Be sure to pull up this pin externally when on-chip debugging is enabled (pulling it down is prohibited).

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

Table 2-1. Relationships between P40/TOOL0 and Operation Mode After Reset Release

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

For details, see **27.5, Programming Method**.

2.2.6 P50 to P55 (port 5)

P50 to P55 function as an I/O port. These pins also function as external interrupt request input, and serial interface data I/O.

Output from the P50 and P55 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units, using port output mode register 5 (POM5).

Input of the P55 pin can be specified as normal input buffer or TTL input buffer in 1-bit units, using port input mode register (PIM5).

The following operation modes can be specified in 1-bit units.

(1) Port mode

P50 to P55 function as an I/O port. P50 to P55 can be set to input or output port in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

(2) Control mode

P50 to P55 function as external interrupt request input, and serial interface data I/O.

(a) INTP1, INTP2

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) SI11

This is the serial data input pin of serial interface CSI11.

(c) SO11

This is the serial data output pin of serial interface CSI11.

(d) SDA11

This is the serial data I/O pin of serial interface for simplified I^2C.

(e) LRxD0

This is a serial data input pin of serial interface LIN-UART0.

(f) LTxD0

This is a serial data output pin of serial interface LIN-UART0.

(g) SOS1

This is a serial data output pin of serial interface CSIS1.

(h) SIS1

This is a serial data input pin of serial interface CSIS1.

(i) $\overline{\text{SCKS1}}$

This is a clock I/O pin of serial interface CSIS1.

2.2.7 P60 to P63 (port 6)

P60 to P63 function as an I/O port. These pins also function as serial interface data I/O, and clock I/O.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P60 to P63 function as an I/O port. P60 to P63 can be set to input port or output port in 1-bit units using port mode register 6 (PM6).

Output of P60 to P63 is N-ch open-drain output (6 V tolerance).

(2) Control mode

P60 to P63 function as serial interface data I/O, and clock I/O.

(a) SCLA0

This is the serial clock I/O pin of serial interface IICA.

(b) SDAA0

This is the serial data I/O pin of serial interface IICA.

2.2.8 P70 to P77 (port 7)

P70 to P77 function as an I/O port. These pins also function as key interrupt input, serial interface data I/O, clock I/O, and external interrupt request input.

Output from the P71 and P74 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units, using port output mode register 7 (POM7).

The following operation modes can be specified in 1-bit units.

(1) Port mode

P70 to P77 function as an I/O port. P70 to P77 can be set to input or output port in 1-bit units using port mode register 7 (PM7). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

(2) Control mode

P70 to P77 function as key interrupt input, serial interface data I/O, clock I/O, and external interrupt request input.

(a) INTP8 to INTP11

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) KR0 to KR7

These are the key interrupt input pins.

(c) SI01, SI21

These are the serial data input pins of serial interface CSI01 and CSI21.

(d) SO01, SO21

These are the serial data output pins of serial interface CSI01 and CSI21.

(e) $\overline{\text{SCK01}}$, $\overline{\text{SCK21}}$

These are the serial clock I/O pins of serial interface CSI01 and CSI21.

(f) SCL01, SCL21

These are the serial clock output pins of serial interface for simplified I^2C.

(g) SDA01, SDA21

These are the serial data I/O pins of serial interface for simplified I^2C.

2.2.9 P120 to P124 (port 12)

P120 function as an I/O port. P121 to P124 functions as 4-bit input port. These pins also function as A/D converter analog input, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, and external clock input for subsystem clock.

Input to the P120 pin can be specified as analog input or digital input in 1-bit units, using port mode control register 12 (PMC12).

The following operation modes can be specified in 1-bit units.

(1) Port mode

P120 function as a 1-bit I/O port. P120 can be set to input or output port using port mode register 12 (PM12). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121 to P124 functions as a 4-bit input port.

(2) Control mode

P120 to P124 function as A/D converter analog input, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, and external clock input for subsystem clock.

(a) ANI19

This is an analog input pin of A/D converter.

When using this pin as analog input pin, see **12.10 (5) Analog input (ANIn) pins**.

(b) X1, X2

These are the pins for connecting a resonator for main system clock.

(c) EXCLK

This is an external clock input pin for main system clock.

(d) XT1, XT2

These are the pins for connecting a resonator for subsystem clock.

(e) EXCLKS

This is an external clock input pin for subsystem clock.

2.2.10 P130, P137 (port 13)

P130 functions as a 1-bit output-only port. P137 functions as a 1-bit input-only port. P137 pin also functions as external interrupt request input.

(1) Port mode

P130 functions as a 1-bit output-only port.

P137 functions as a 1-bit input-only port.

(2) Control mode

P137 functions as external interrupt request input.

(a) INTP0

This is an external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

2.2.11 P140, P141, P146, P147 (port 14)

P140, P141, P146, P147 function as an I/O port. These pins also function as clock/buzzer output, external interrupt request input, and A/D converter analog input.

Input to the P147 pin can be specified as analog input or digital input in 1-bit units, using port mode control register 14 (PMC14).

The following operation modes can be specified in 1-bit units.

(1) Port mode

P140, P141, P146, P147 function as an I/O port. P140, P141, P146, P147 can be set to input or output port in 1-bit units using port mode register 14 (PM14). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 14 (PU14).

(2) Control mode

P140, P141, P146, P147 function as clock/buzzer output, external interrupt request input, and A/D converter analog input.

(a) ANI18

This is an analog input pin of A/D converter.

When using this pin as analog input pin, see **12.10 (5) Analog input (ANIn) pins**.

(b) INTP6, INTP7

This is the external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(c) PCLBUZ0, PCLBUZ1

This is the clock/buzzer output pin.

2.2.12 V_{DD}, V_{SS}

(1) V_{DD}

V_{DD} is the positive power supply pin.

(2) V_{SS}

V_{SS} is the ground potential pin.

2.2.13 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

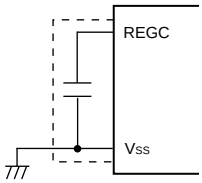
When the external reset pin is not used, connect this pin directly or via a resistor to V_{DD}.

When the external reset pin is used, design the circuit based on V_{DD}.

2.2.14 REGC

This is the pin for connecting regulator output stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF: target).

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.

2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

Table 2-2 shows the types of pin I/O circuits and the recommended connections of unused pins.

Table 2-2. Connection of Unused Pins (64-pin products) (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI00	8-R-1	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01/TO00	5-AN-1		
P02/ANI17/SO10/TXD1	11-U-1		
P03/ANI16/SI10/RXD1/SDA10	11-V-1		
P04/$\overline{\text{SCK10}}$/SCL10	5-AN-1		
P05/TI05/TO05	8-R-1		
P06/TI06/TO06			
P10/$\overline{\text{SCK00}}$/$\overline{\text{SCKS0}}$/SCL00/(TI07)/(TO07)	5-AN-1		
P11/SI00/RxD0/SIS0/RxDS0/TOOLRxD/SDA00/(TI06)/(TO06)			
P12/SO00/TxD0/SOS0/TxDS0/TOOLTxD/(INTP5)/(TI05)/(TO05)	8-R-1		
P13/TxD2/SO20/(SDAA0)/(TI04)/(TO04)	5-AN-1		
P14/RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)			
P15/$\overline{\text{SCK20}}$/SCL20/(TI02)/(TO02)			
P16/TI01/TO01/INTP5/(RXD0)/(SI00)			
P17/TI02/TO02/(TXD0)/(SO00)			
P20/ANI0/AV_{REFP}	11-T		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P21/ANI1/AV_{REFM}			
P22/ANI2	11-G		
P23/ANI3			
P24/ANI4			
P25/ANI5			
P26/ANI6			
P27/ANI7			
P30/INTP3/RTC1HZ/$\overline{\text{SCK11}}$/SCL11	8-R-1		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P31/TI03/TO03/INTP4/(PCLBUZ0)			

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

Caution With products with 48 or less pins, replace EV_{DD} and EV_{SS} described in Recommended Connection of Unused Pins with V_{DD} and V_{SS}, respectively.

Table 2-2. Connection of Unused Pins (64-pin products) (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P40/TOOL0	8-R-1	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P41/TI07/TO07			
P42/TI04/TO04			
P43			
P50/INTP1/SI11/SDA11/ LRxD0			
P51/INTP2/SO11/LTxD0			
P52/(INTP10)			
P53/SOS1/(INTP11)			
P54/SIS1			
P55/$\overline{SCKS1}$/(PCLBUZ1)/ ($\overline{SCK00}$)	5-AN-1		
P60/SCLA0	13-R		
P61/SDAA0			
P62			
P63			
P70/KR0/$\overline{SCK21}$/SCL21	8-R-1		
P71/KR1/SI21/SDA21			
P72/KR2/SO21			
P73/KR3/SO01			
P74/KR4/INTP8/SI01/ SDA01			
P75/KR5/INTP9/$\overline{SCK01}$/ SCL01			
P76/KR6/INTP10/(RXD2)			
P77/KR7/INTP11/(TXD2)			
P120/ANI19	11-U-1		
P121/X1	37-C	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK			
P123/XT1			
P124/XT2/EXCLKS			
P130	3-C	Output	Leave open.
P137/INTP0	2	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P140/PCLBUZ0/INTP6	8-R-1	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P141/PCLBUZ1/INTP7			
P146			
P147/ANI18	11-U-1		
$\overline{RESET}$	2	Input	Connect directly or via a resistor to V_{DD}.
REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

Caution With products with 48 or less pins, replace EV_{DD} and EV_{SS} described in Recommended Connection of Unused Pins with V_{DD} and V_{SS}, respectively.

Figure 2-1. Pin I/O Circuit List (1/2)

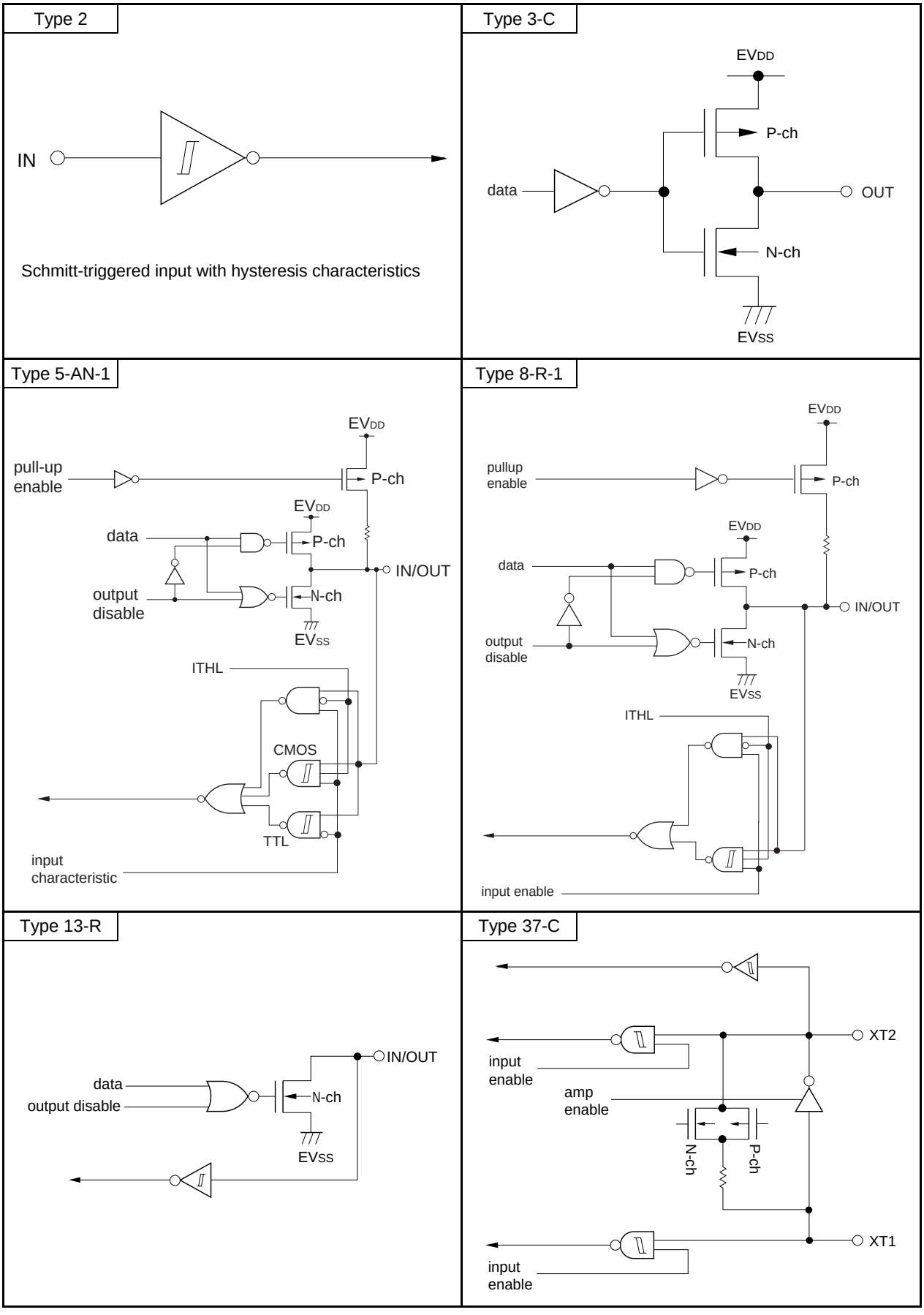
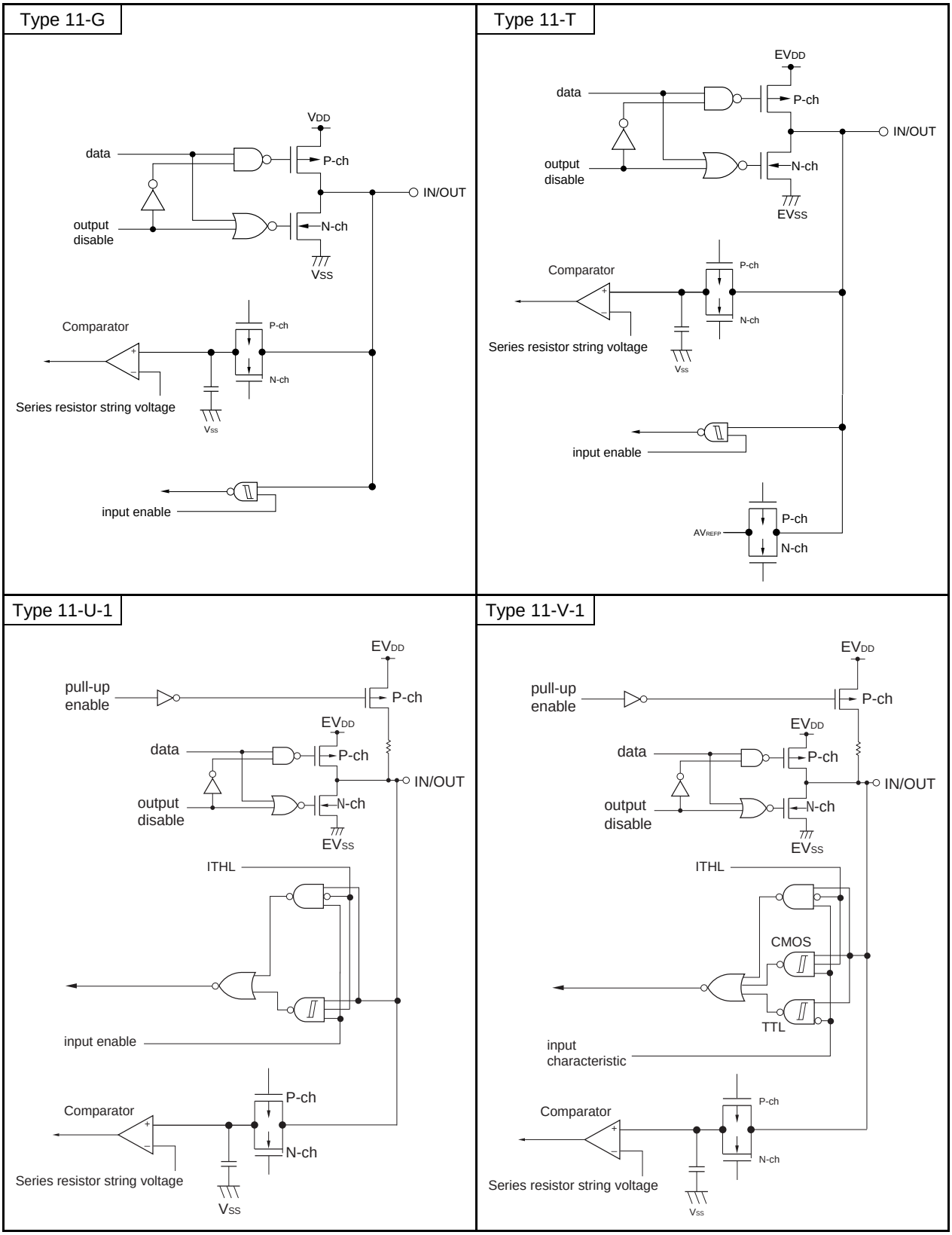


Figure 2-1. Pin I/O Circuit List (2/2)

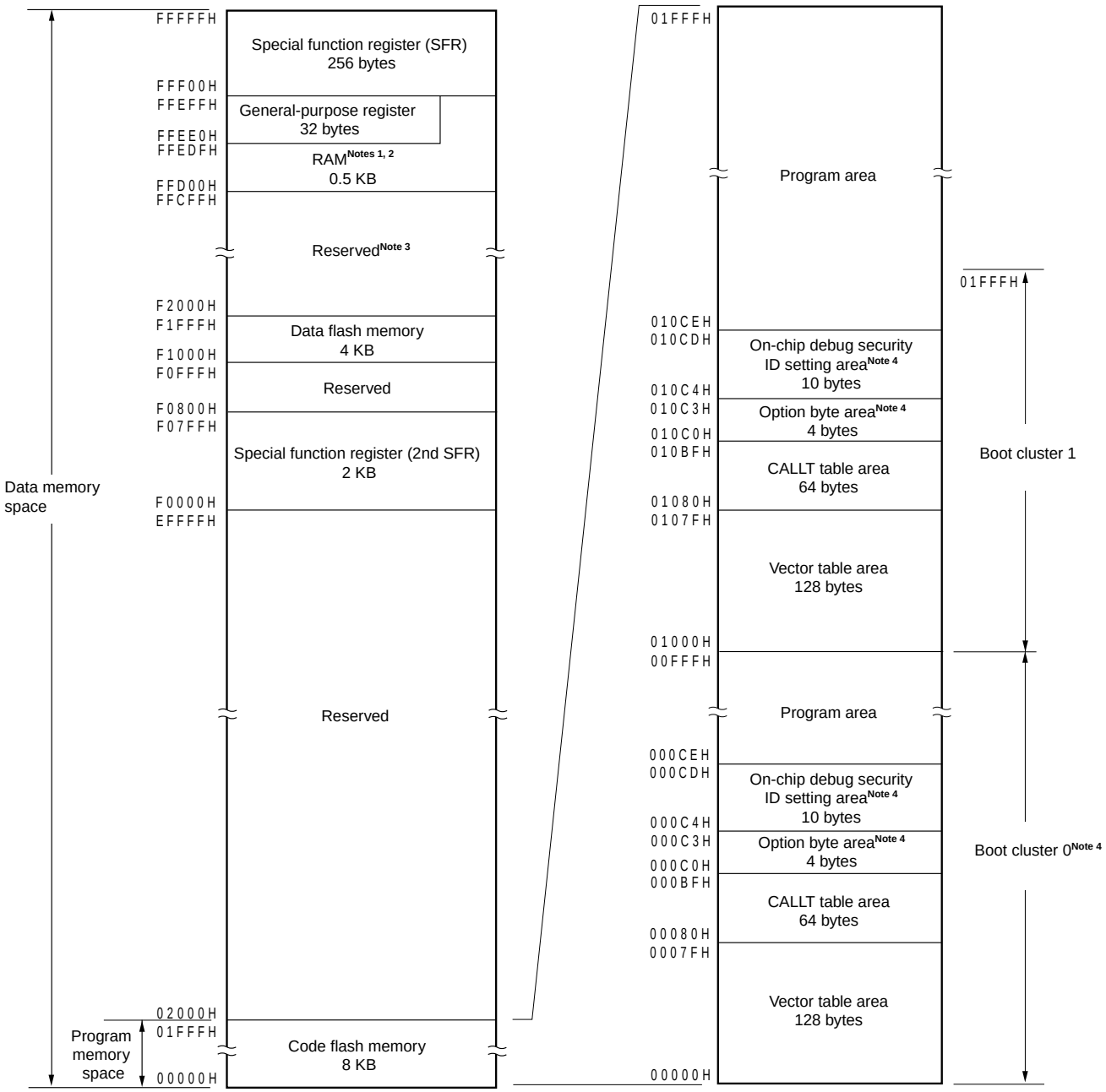


CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the RL78/F12 can access a 1 MB memory space. Figures 3-1 to 3-6 show the memory maps.

Figure 3-1. Memory Map (R5F10968)



<R> **Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffer used by the libraries, branch destination of vector interrupt processing, and DMA destination/source addresses to the area FFE20H to FFEDFH when performing self-programming or rewriting the data flash memory area.

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

3. The mirror area is not provided in the R5F10968.

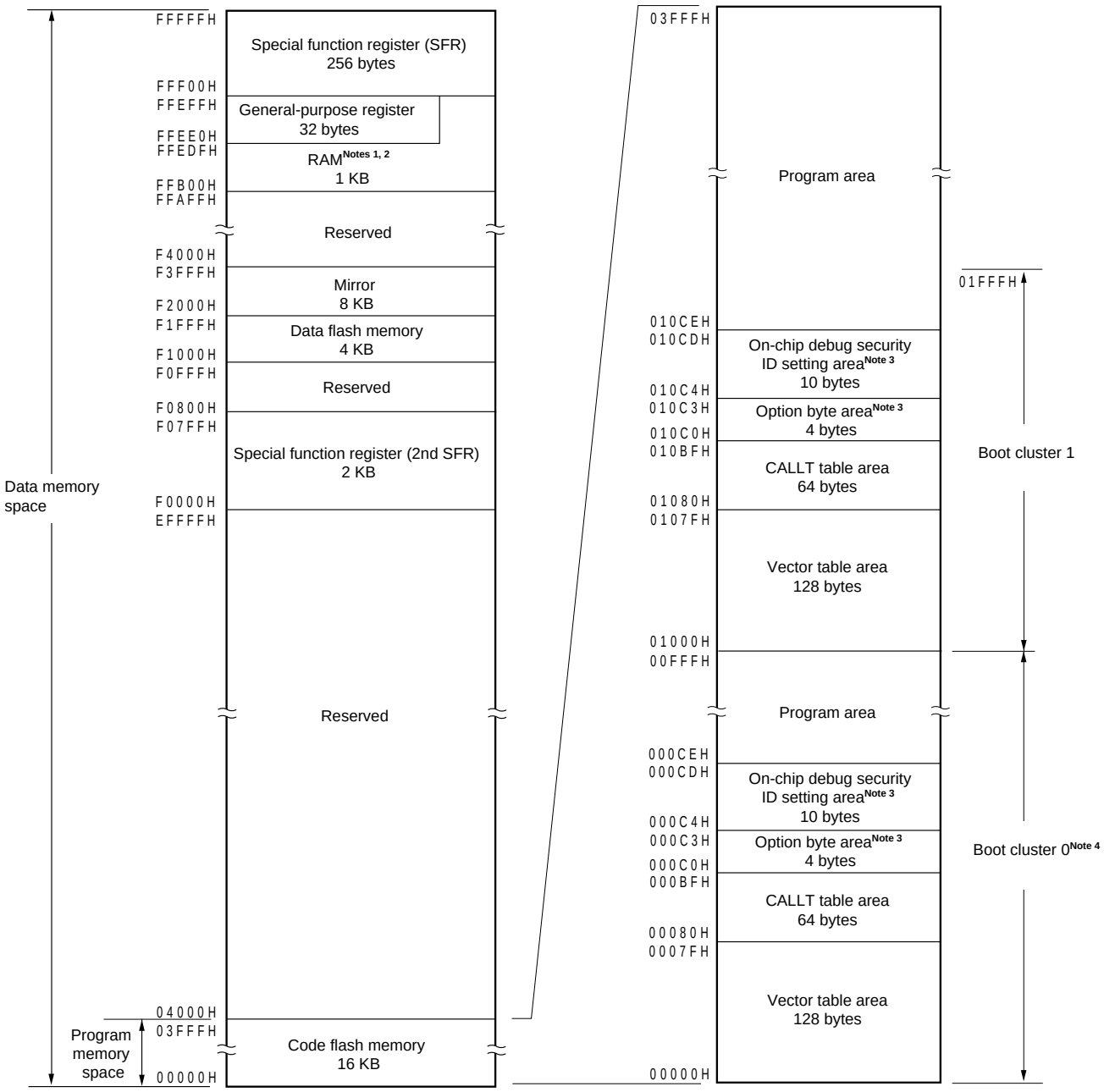
4. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

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

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-2. Memory Map (R5F109xA (x = 6, A, B, G, L))



<R>

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

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

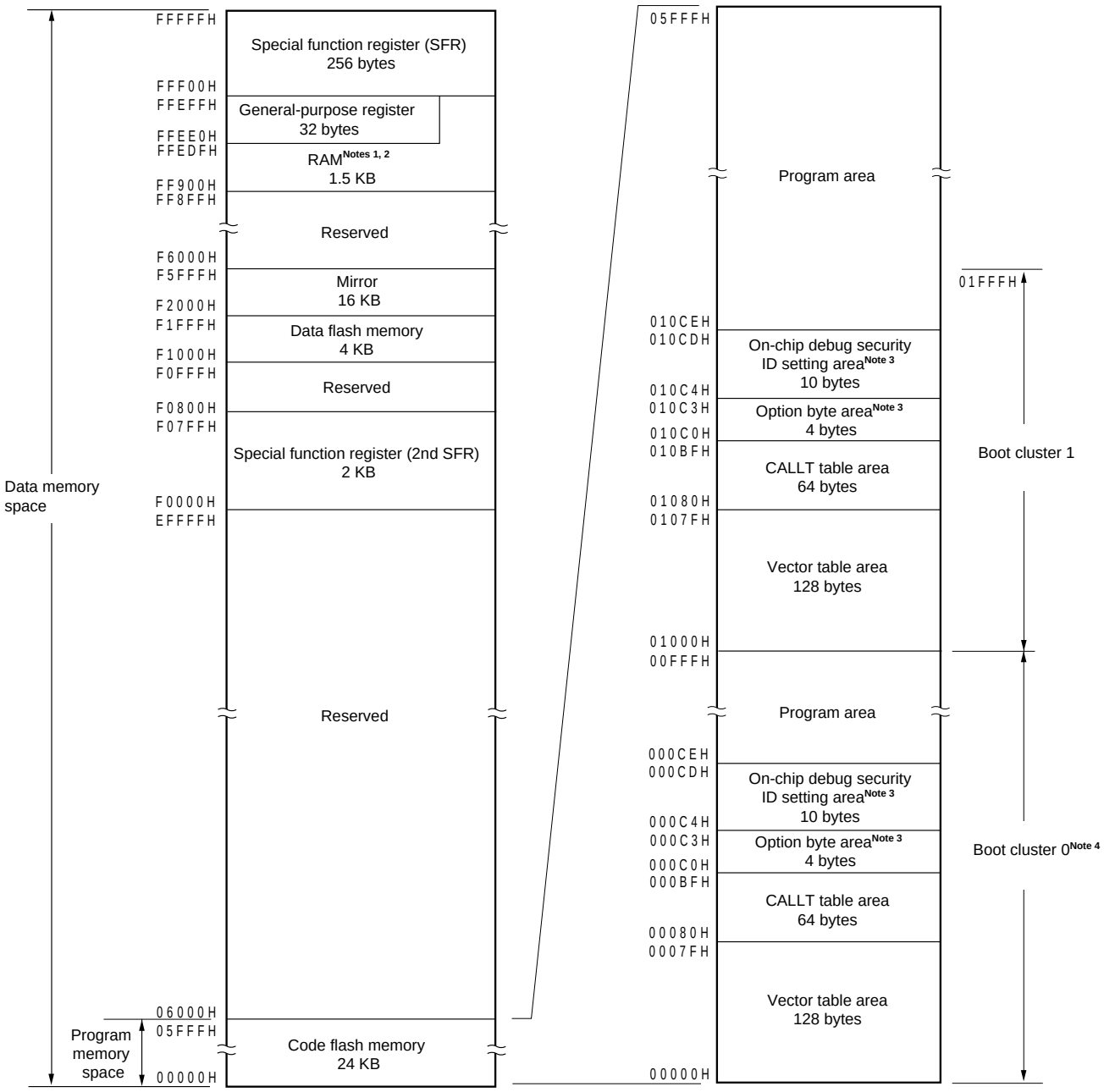
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

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

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-3. Memory Map (R5F109xB (x = 6, A, B, G, L))



<R> **Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffer used by the libraries, branch destination of vector interrupt processing, and DMA destination/source addresses to the area FFE20H to FFEDFH when performing self-programming or rewriting the data flash memory area.

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

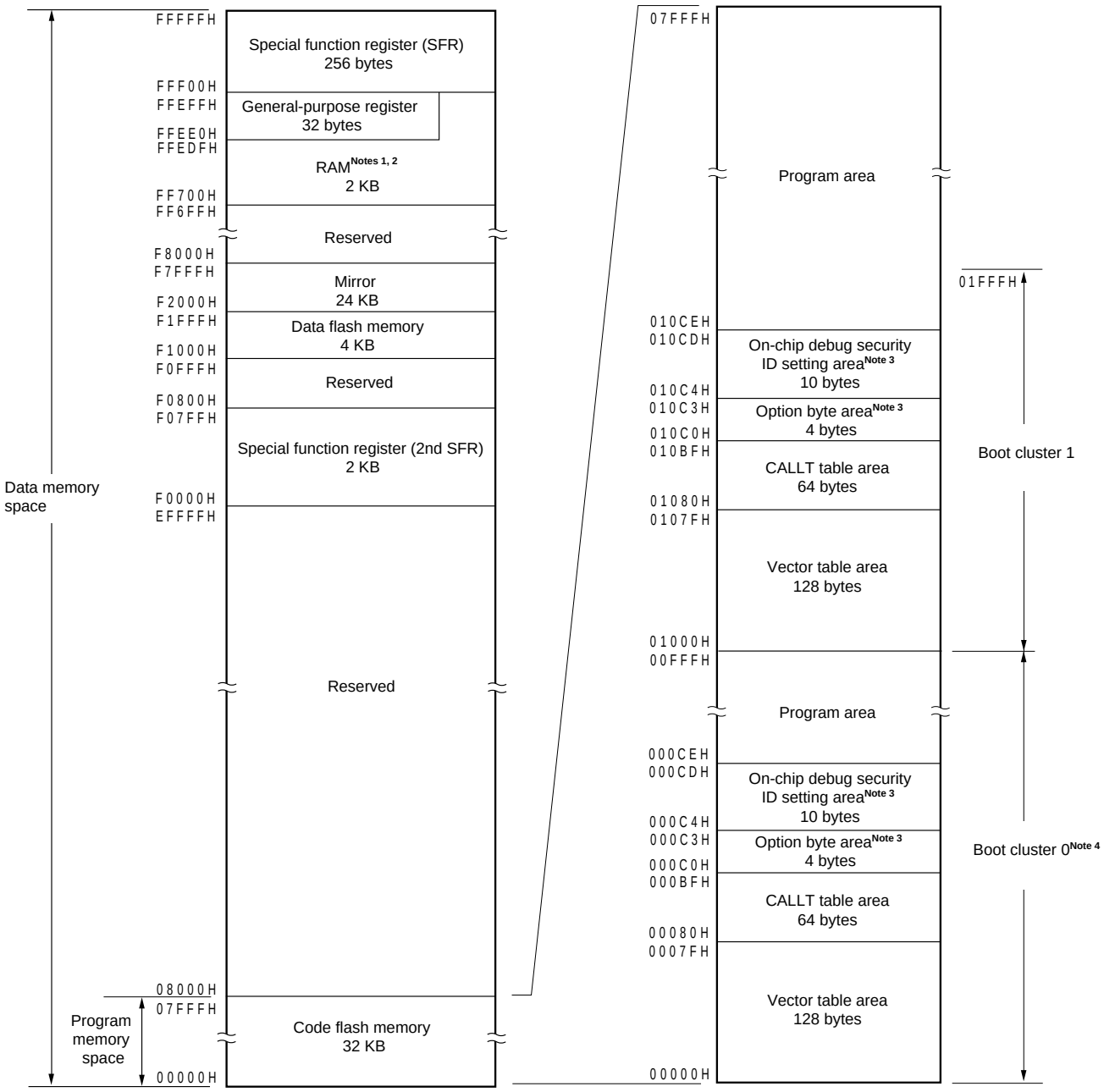
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

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

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-4. Memory Map (R5F109xC (x = 6, A, B, G, L))



<R> **Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffer used by the libraries, branch destination of vector interrupt processing, and DMA destination/source addresses to the area FFE20H to FFEDFH when performing self-programming or rewriting the data flash memory area.

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

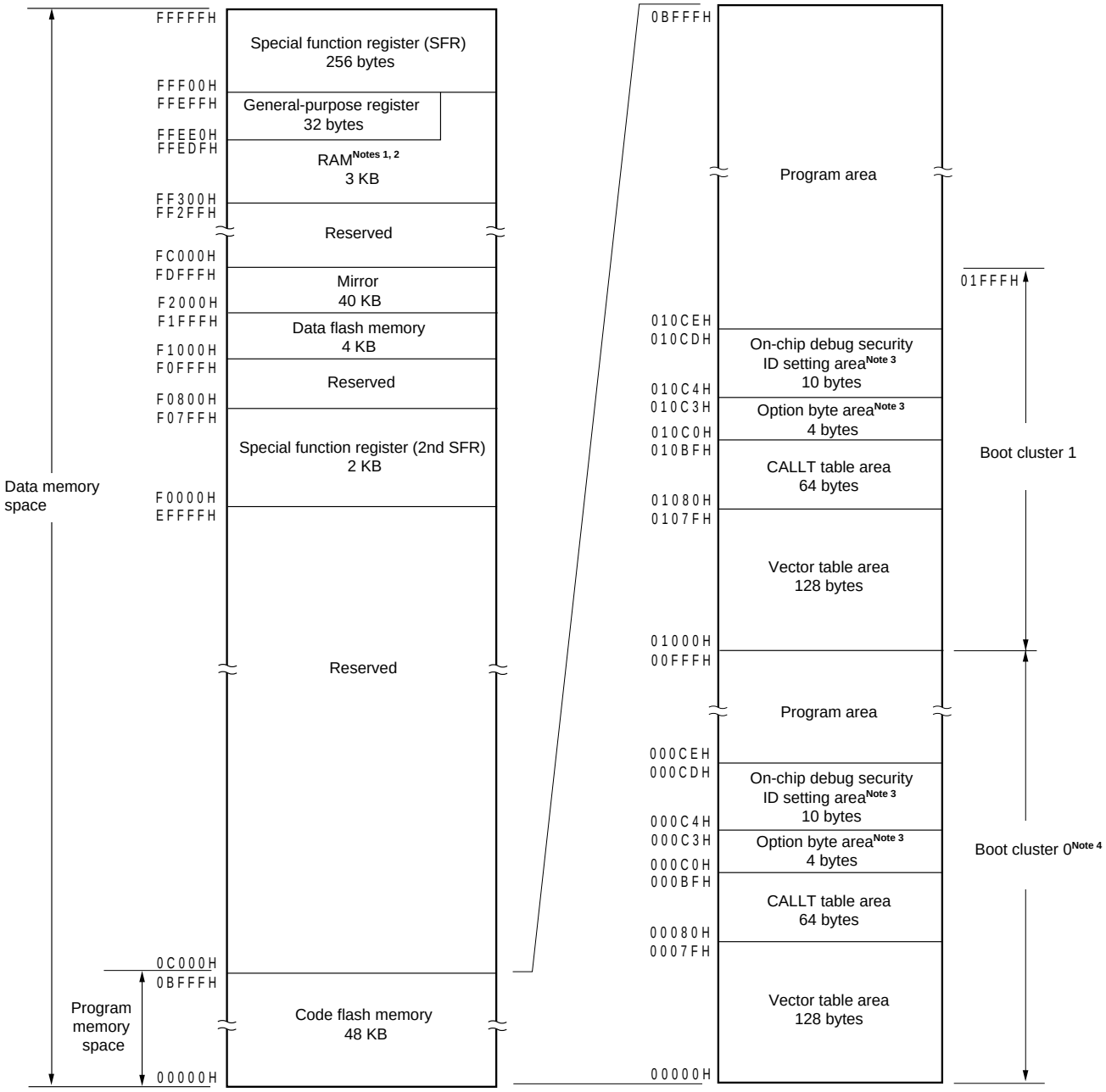
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

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

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-5. Memory Map (R5F109xD (x = 6, A, B, G, L))



<R> **Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffer used by the libraries, branch destination of vector interrupt processing, and DMA destination/source addresses to the area FFE20H to FFEDFH when performing self-programming or rewriting the data flash memory area.

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

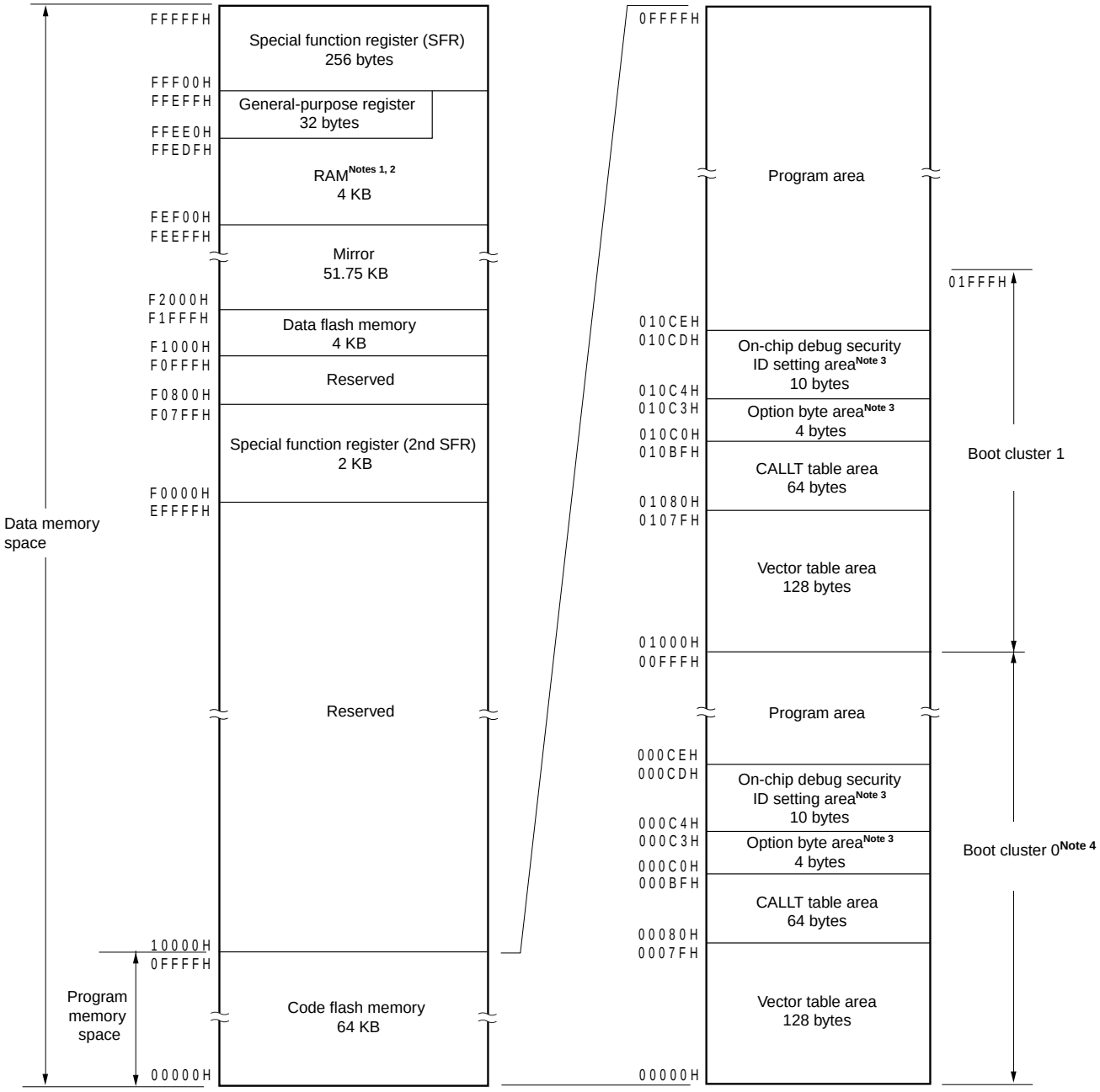
3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

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

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

Figure 3-6. Memory Map (R5F109xE (x = 6, A, B, G, L))



<R>

Notes 1. Do not allocate RAM addresses which are used as stack area, data buffer used by the libraries, branch destination of vector interrupt processing, and DMA destination/source addresses to the area FFE20H to FFEDFH when performing self-programming or rewriting the data flash memory area. In addition, the use of the area FEF00H to FF2FFH is prohibited, because this area might be used by each library. However, the area to which this prohibition applies may vary with the version of the library. For details, please refer to the manual for the individual library.

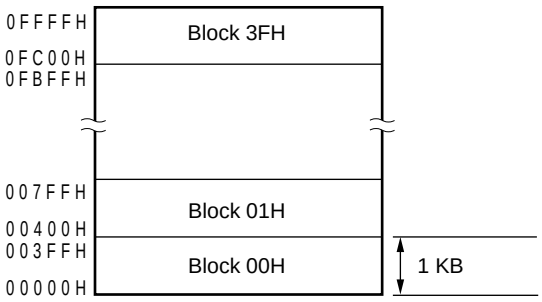
2. Instructions can be executed from the RAM area excluding the general-purpose register area.

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.

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

Caution When executing instructions from the RAM area while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the used RAM area + 10 bytes.

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



(R5F109xE (x = 6, A, B, G, L))

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

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

Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	00H	08000H to 083FFH	20H
00400H to 007FFH	01H	08400H to 087FFH	21H
00800H to 00BFFH	02H	08800H to 08BFFH	22H
00C00H to 00FFFH	03H	08C00H to 08FFFH	23H
01000H to 013FFH	04H	09000H to 093FFH	24H
01400H to 017FFH	05H	09400H to 097FFH	25H
01800H to 01BFFH	06H	09800H to 09BFFH	26H
01C00H to 01FFFH	07H	09C00H to 09FFFH	27H
02000H to 023FFH	08H	0A000H to 0A3FFH	28H
02400H to 027FFH	09H	0A400H to 0A7FFH	29H
02800H to 02BFFH	0AH	0A800H to 0ABFFH	2AH
02C00H to 02FFFH	0BH	0AC00H to 0AFFFH	2BH
03000H to 033FFH	0CH	0B000H to 0B3FFH	2CH
03400H to 037FFH	0DH	0B400H to 0B7FFH	2DH
03800H to 03BFFH	0EH	0B800H to 0BBFFH	2EH
03C00H to 03FFFH	0FH	0BC00H to 0BFFFH	2FH
04000H to 043FFH	10H	0C000H to 0C3FFH	30H
04400H to 047FFH	11H	0C400H to 0C7FFH	31H
04800H to 04BFFH	12H	0C800H to 0CBFFH	32H
04C00H to 04FFFH	13H	0CC00H to 0CFFFH	33H
05000H to 053FFH	14H	0D000H to 0D3FFH	34H
05400H to 057FFH	15H	0D400H to 0D7FFH	35H
05800H to 05BFFH	16H	0D800H to 0DBFFH	36H
05C00H to 05FFFH	17H	0DC00H to 0DFFFH	37H
06000H to 063FFH	18H	0E000H to 0E3FFH	38H
06400H to 067FFH	19H	0E400H to 0E7FFH	39H
06800H to 06BFFH	1AH	0E800H to 0EBFFH	3AH
06C00H to 06FFFH	1BH	0EC00H to 0EFFFH	3BH
07000H to 073FFH	1CH	0F000H to 0F3FFH	3CH
07400H to 077FFH	1DH	0F400H to 0F7FFH	3DH
07800H to 07BFFH	1EH	0F800H to 0FBFFH	3EH
07C00H to 07FFFH	1FH	0FC00H to 0FFFFH	3FH

3.1.1 Internal program memory space

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

The RL78/F12 products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F10968	Flash memory	8192 × 8 bits (00000H to 01FFFH)
R5F109xA (x = 6, A, B, G, L)		16384 × 8 bits (00000H to 03FFFH)
R5F109xB (x = 6, A, B, G, L)		24576 × 8 bits (00000H to 05FFFH)
R5F109xC (x = 6, A, B, G, L)		32768 × 8 bits (00000H to 07FFFH)
R5F109xD (x = 6, A, B, G, L)		49152 × 8 bits (00000H to 0BFFFH)
R5F109xE (x = 6, A, B, G, L)		65536 × 8 bits (00000H to 0FFFFH)

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

(1) Vector table area

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

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

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

Table 3-3. Vector Table (1/2)

Vector Table Address	Interrupt Source	64-pin	48-pin	32-pin	30-pin	20-pin
0000H	RESET, POR, LVD, WDT, TRAP, IAW, RAMTOP	√	√	√	√	√
0004H	INTWDTI	√	√	√	√	√
0006H	INTLVI	√	√	√	√	√
0008H	INTP0	√	√	√	√	√
000AH	INTP1	√	√	√	√	√
000CH	INTP2	√	√	√	√	√
000EH	INTP3	√	√	√	√	–
0010H	INTP4	√	√	√	√	√
0012H	INTP5	√	√	√	√	√
0014H	INTST2/INTCSI20/INTIIC20	√	√	√	√	–
0016H	INTSR2/INTCSI21/INTIIC21	√	√	**Note 1**	**Note 1**	–
0018H	INTSRE2	√	√	√	√	–
001AH	INTDMA0	√	√	√	√	√
001CH	INTDMA1	√	√	√	√	√
001EH	INTST0/INTCSI00/INTIIC00	√	√	√	√	√
0020H	INTSR0/INTCSI01/INTIIC01	√	√	**Note 2**	**Note 2**	**Note 2**
0022H	INTSRE0	√	√	√	√	√
	INTTM01H	√	√	√	√	√
0024H	INTST1/INTCSI10/INTIIC10	√	**Note 3**	**Note 3**	**Note 3**	–
0026H	INTSR1/INTCSI11/INTIIC11	√	√	√	√	–
0028H	INTSRE1/INTTM03H	√	√	√	√	**Note 4**
002AH	INTIICA0	√	√	√	√	–
002CH	INTTM00	√	√	√	√	√
002EH	INTTM01	√	√	√	√	√
0030H	INTTM02	√	√	√	√	√
0032H	INTTM03	√	√	√	√	√
0034H	INTAD	√	√	√	√	√
0036H	INTRTC	√	√	–	–	–
0038H	INTIT	√	√	√	√	√
003AH	INTKR	√	√	–	–	–
003CH	INTCSIS0/INTSTS0	√	√	√	√	√
003EH	INTCSIS1/INTSRS0	√	**Note 5**	**Note 5**	**Note 5**	**Note 5**

Notes 1. INTSR2 only.

2. INTSR0 only.

3. INTST1 only.

4. INTTM03H only.

5. INTCSIS1 only.

Table 3-3. Vector Table (2/2)

Vector Table Address	Interrupt Source	64-pin	48-pin	32-pin	30-pin	20-pin
0040H	INTWUTM	√	√	√	√	√
0042H	INTTM04	√	√	√	√	√
0044H	INTTM05	√	√	√	√	√
0046H	INTTM06	√	√	√	√	√
0048H	INTTM07	√	√	√	√	√
004AH	INTP6	√	√	–	–	–
004CH	INTP7/INTLT	√	**Note 2**	**Note 2**	**Note 2**	**Note 2**
004EH	INTP8/INTLR**Note 1**	√	√	**Note 3**	**Note 3**	**Note 3**
0050H	INTP9/INTLS**Note 1**	√	√	**Note 4**	**Note 4**	**Note 4**
0052H	INTP10/INTSRES0	√	**Note 5**	**Note 5**	**Note 5**	**Note 5**
0054H	INTP11	√	–	–	–	–
005EH	INTMD	√	√	√	√	√
0062H	INTFL	√	√	√	√	√
007EH	BRK	√	√	√	√	√

Notes 1. For 48-pin products, when INTP8 and INTLR interrupts occur at the same time, the interrupt source cannot be distinguished from the vector address. The INTP9 and INTLS interrupts (that occur at the sane time) are the same as the condition above.

2. INTLT only.

3. INTLR only.

4. INTLS only.

5. INTSRES0 only.

(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 of 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 010C4H to 010CDH when the boot swap is used. For details, see **CHAPTER 28 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The RL78/F12 mirrors the code flash area of 00000H to 0FFFFH, to F0000H to FFFFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

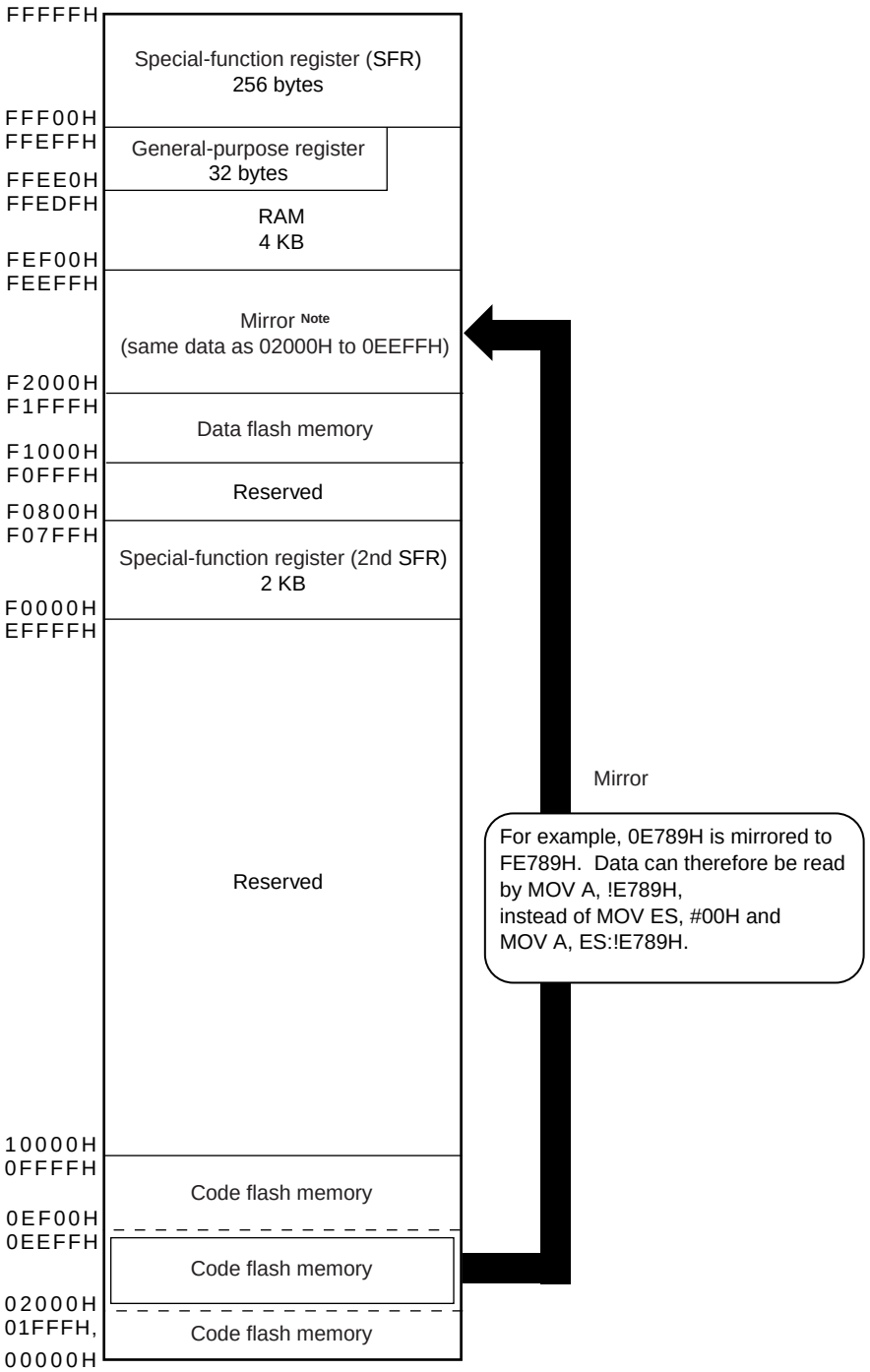
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

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

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

The following show examples.

Example R5F109xE (x = 6, A, B, G) (Flash memory: 64 KB, RAM: 4 KB)



Note The mirror area is not provided in the R5F10968.

The PMC register is described below.

- **Processor mode control register (PMC)**

This register sets the flash memory space for mirroring to area from F0000H to FFFFFH.

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

Reset signal generation sets this register to 00H.

Figure 3-7. Format of Configuration of Processor Mode Control Register (PMC)

Address: FFFFEH After reset: 00H R/W

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

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

Cautions 1. Be sure to clear bit 0 (MAA) of this register to 0 (default value).

2. Set the PMC register only once during the initial settings prior to operating the DMA controller. Rewriting the PMC register other than during the initial settings is prohibited.

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

3.1.3 Internal data memory space

The RL78/F12 products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
R5F10968	512 × 8 bits (FFD00H-FFEFFH)
R5F109xA (x = 6, A, B, G, L)	1024 × 8 bits (FFB00H-FFEFFH)
R5F109xB (x = 6, A, B, G, L)	1536 × 8 bits (FF900H-FFEFFH)
R5F109xC (x = 6, A, B, G, L)	2048 × 8 bits (FF700H to FFEFFH)
R5F109xD (x = 6, A, B, G, L)	3072 × 8 bits (FF300H to FFEFFH)
R5F109xE (x = 6, A, B, G, L)	4096 × 8 bits (FEF00H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are written and executed. 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. However, instructions cannot be executed by using the general-purpose registers.

The internal RAM is used as a stack memory.

Cautions 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

2. While using the self-programming function or data flash function, the area FFE20H to FFEFFH cannot be used as stack memory. Furthermore, the areas of FEF00H to FF309H also cannot be used with the R5F109xE (x = 6, A, B, G), respectively.

3.1.4 Special function register (SFR) area

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

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

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

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

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.

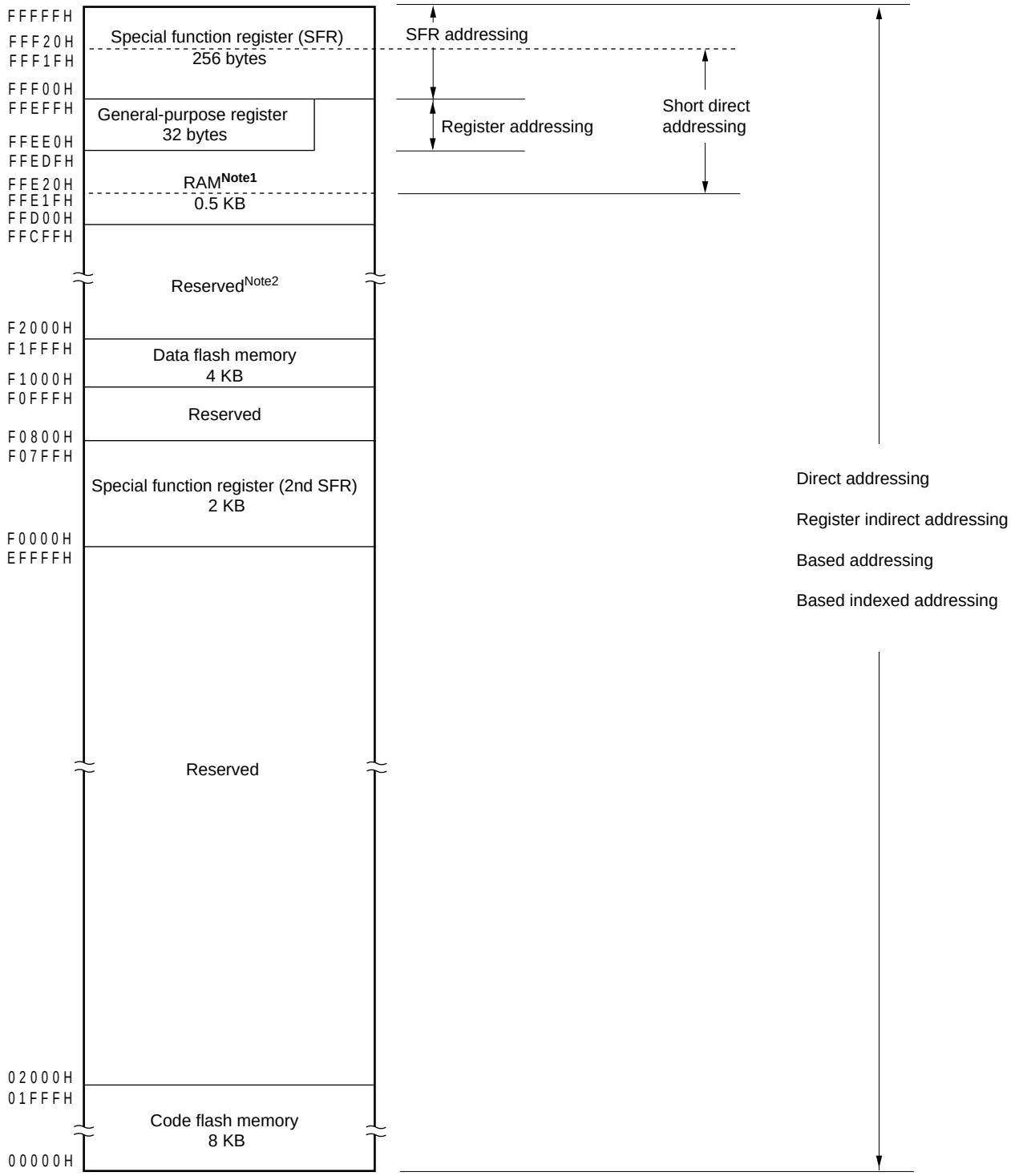
Caution Do not access addresses to which extended SFRs are not assigned.

3.1.6 Data memory addressing

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

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/F12, 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. Figures 3-8 to 3-13 show correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

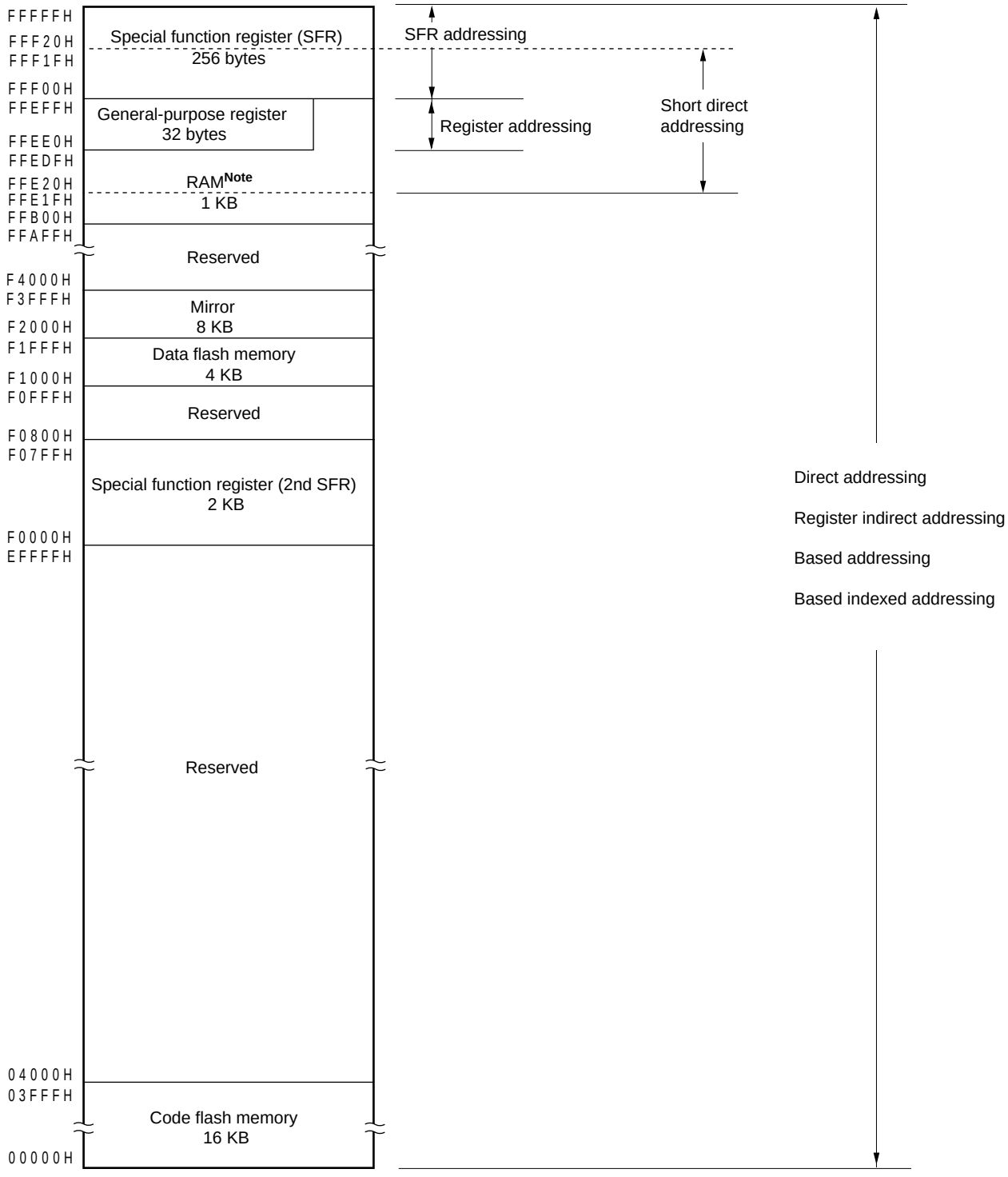
Figure 3-8. Correspondence Between Data Memory and Addressing (R5F10968)



Notes 1. Use of the area FFE20H to FFEDFH is prohibited when using the self-programming function or data flash function, because this area is used for self-programming library.

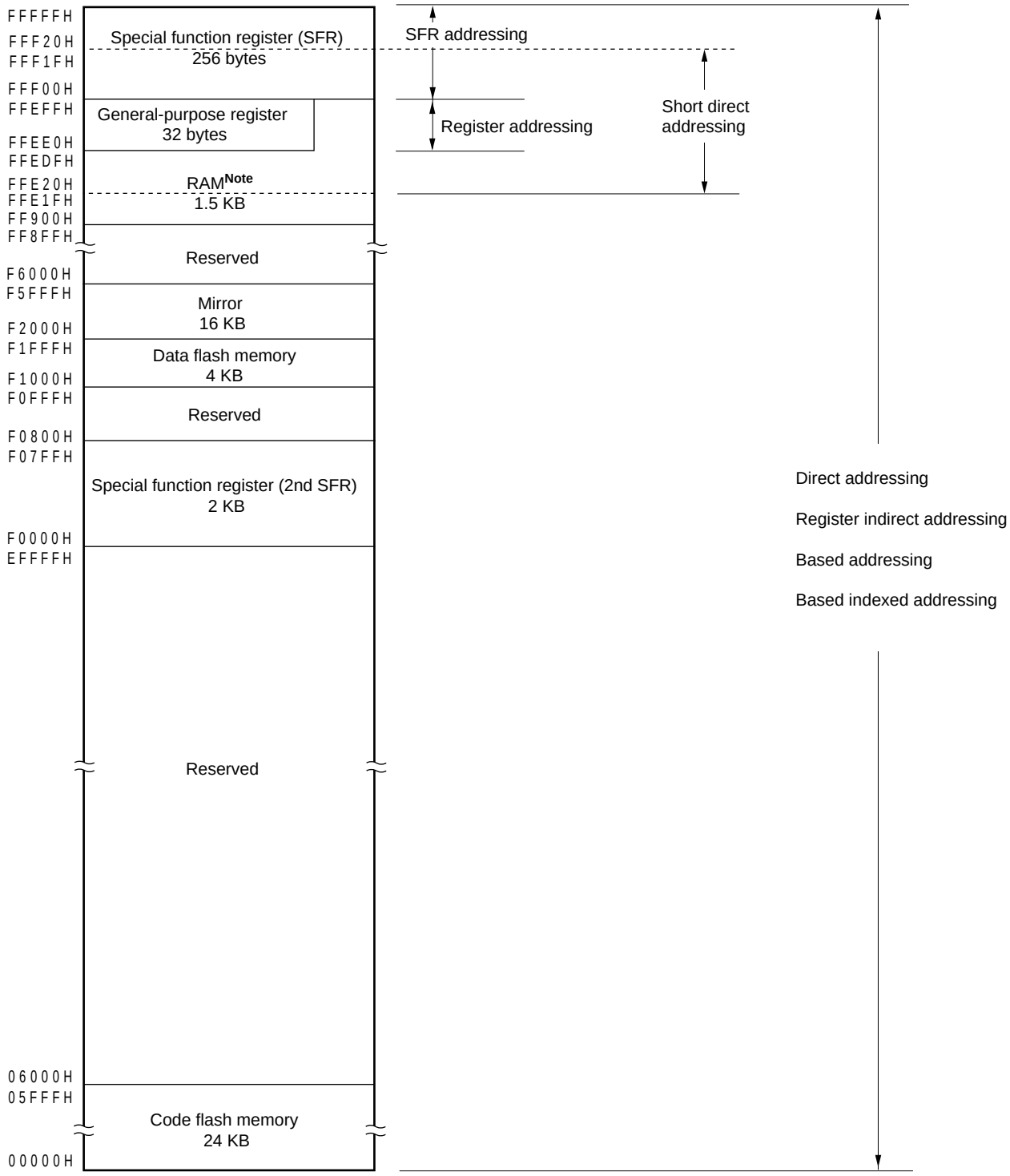
2. The mirror area is not provided in the R5F10968.

Figure 3-9. Correspondence Between Data Memory and Addressing (R5F109xA (x = 6, A, B, G, L))



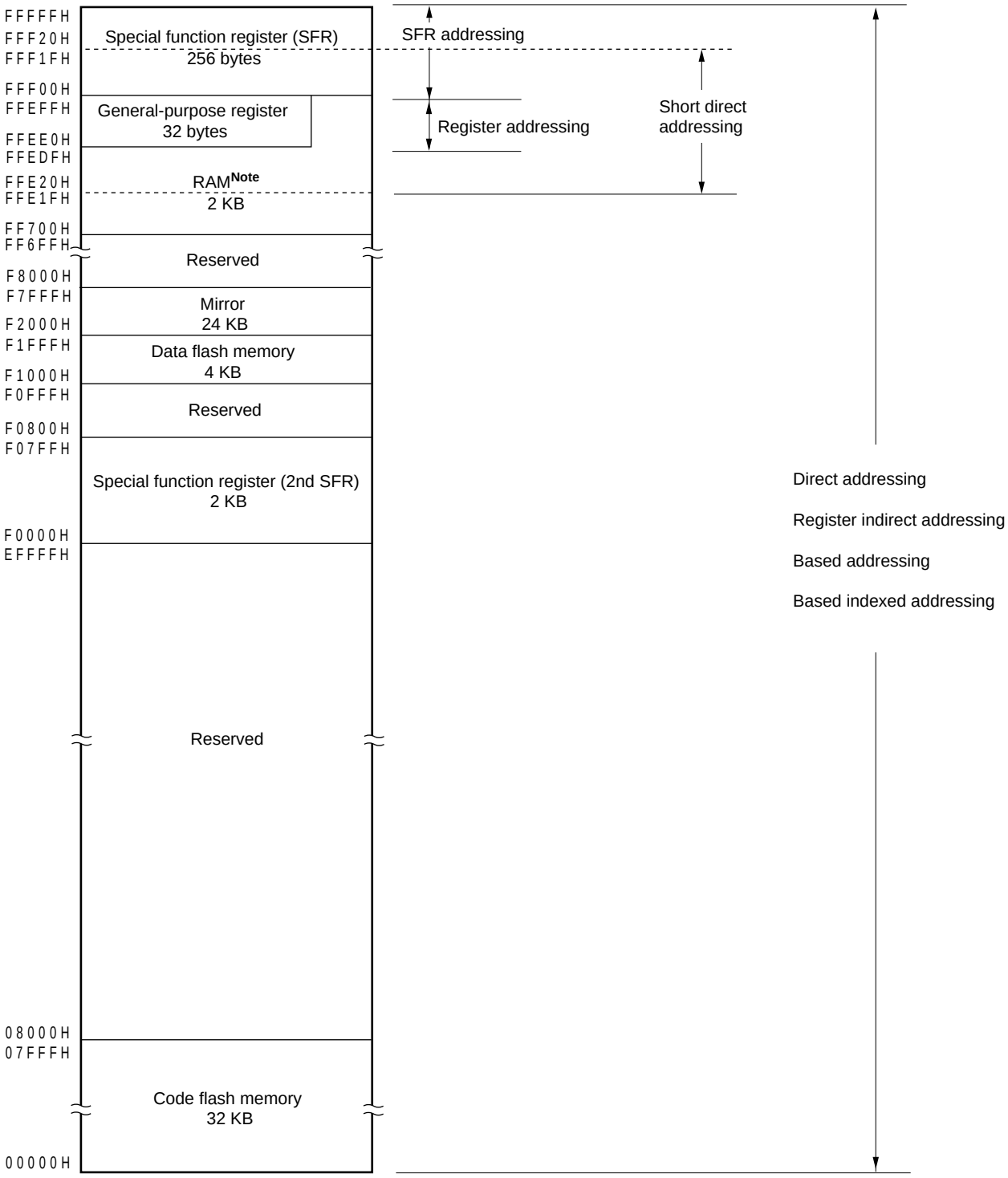
Note Use of the area FFE20H to FFEDFH is prohibited when using the self-programming function or data flash function, because this area is used for self-programming library.

Figure 3-10. Correspondence Between Data Memory and Addressing (R5F109xB (x = 6, A, B, G, L))



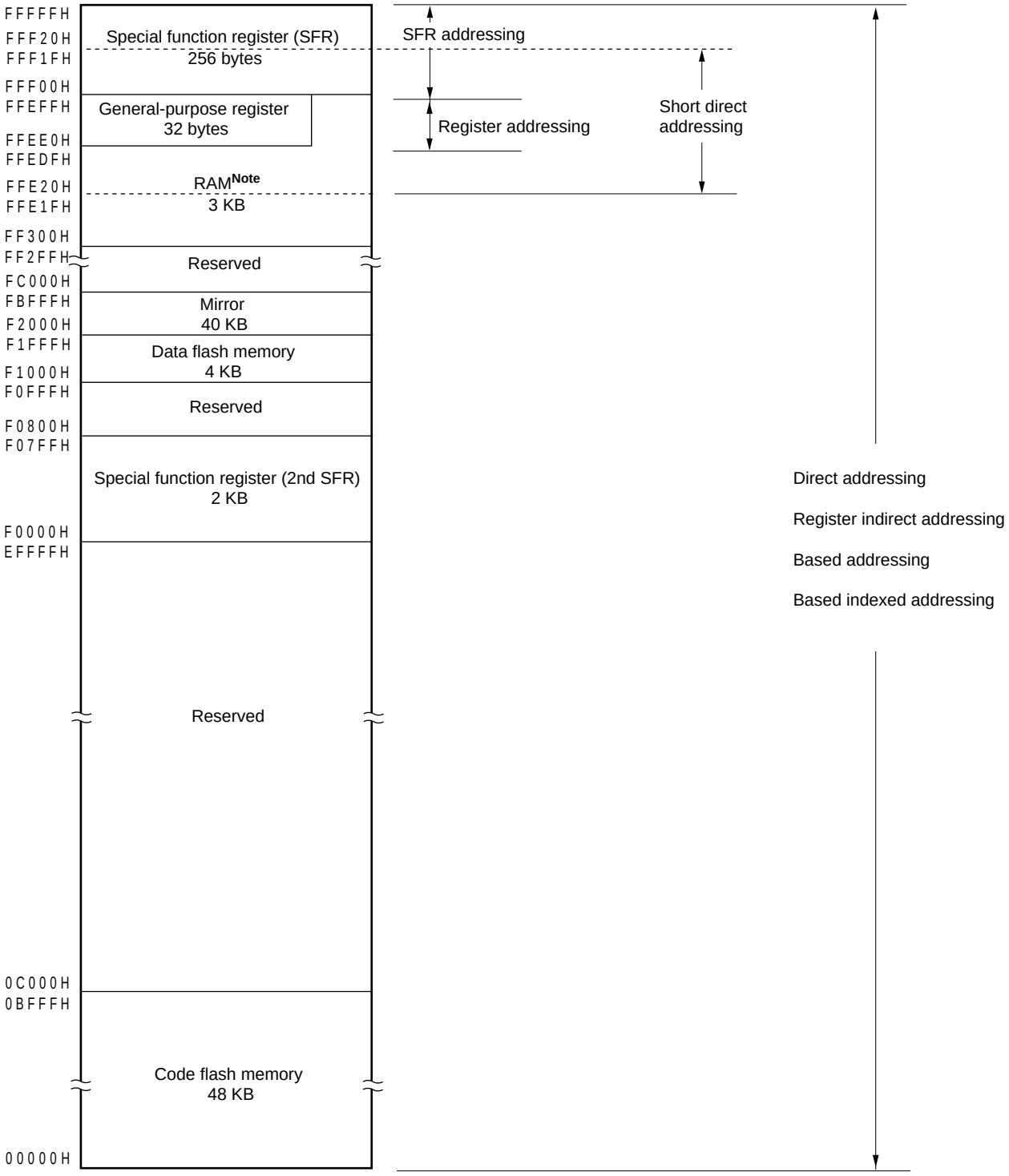
Note Use of the area FFE20H to FFEDFH is prohibited when using the self-programming function or data flash function, because this area is used for self-programming library.

Figure 3-11. Correspondence Between Data Memory and Addressing (R5F109xC (x = 6, A, B, G, L))



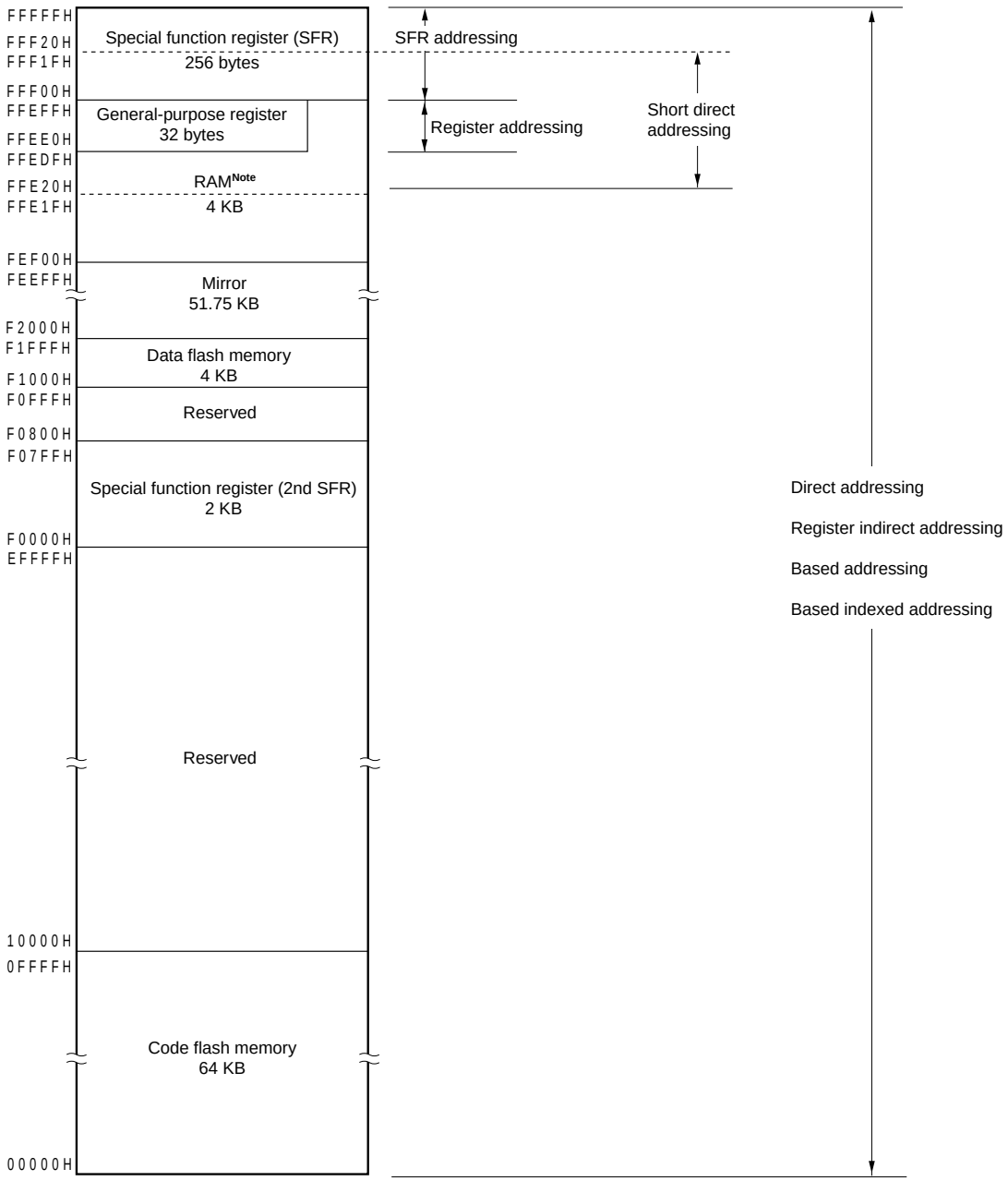
Note Use of the area FFE20H to FFEDFH is prohibited when using the self-programming function or data flash function, because this area is used for self-programming library.

Figure 3-12. Correspondence Between Data Memory and Addressing (R5F109xD (x = 6, A, B, G, L))



Note Use of the area FFE20H to FFEDFH is prohibited when using the self-programming function or data flash function, because this area is used for self-programming library.

Figure 3-13. Correspondence Between Data Memory and Addressing (R5F109xE (x = 6, A, B, G, L))



Note Use of the area FFE20H to FFEDFH and FEF00H to FF309H are prohibited when using the self-programming function or data flash function, because this area is used for self-programming library.

3.2 Processor Registers

The RL78/F12 products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

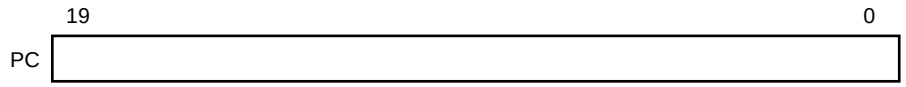
The program counter is a 20-bit register that holds the address information of the next program to be executed.

In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched.

When a branch instruction is executed, immediate data and register contents are set.

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

Figure 3-14. Format of Program Counter



(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags set/reset by instruction execution.

Program status word contents are stored in the stack area upon vectored interrupt request is acknowledged or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions. Reset signal generation sets the PSW register to 06H.

Figure 3-15. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

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

This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **18.3 (3)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-16. Format of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

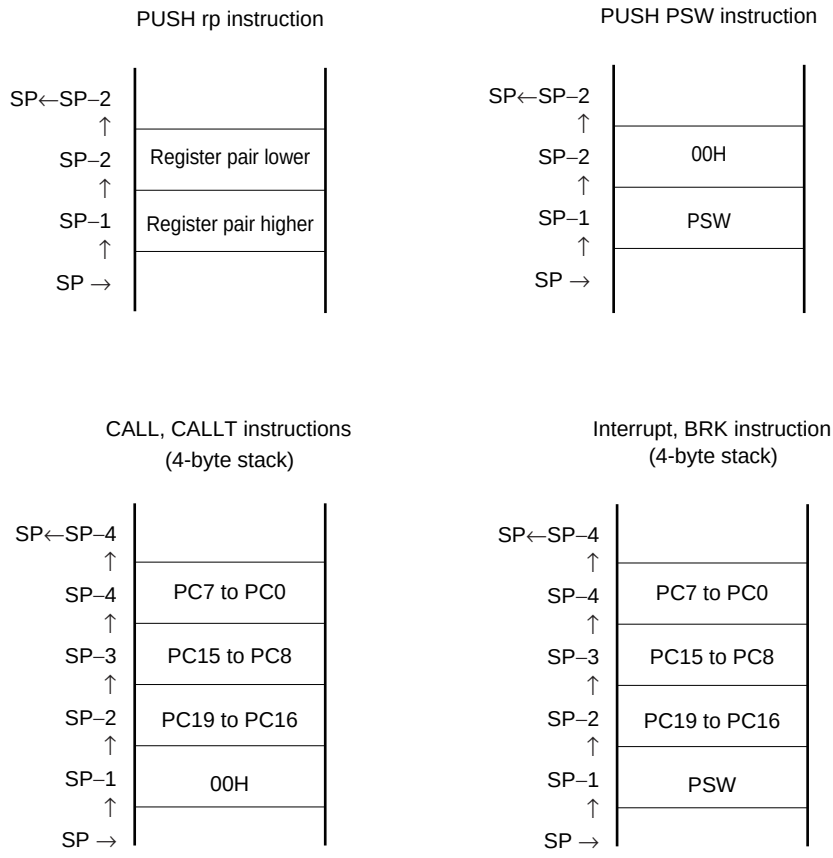
Each stack operation saves data as shown in Figure 3-17.

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

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

3. While using the self-programming function or data flash function, the area FFE20H to FFEFFH cannot be used as stack memory. Furthermore, the areas of FEF00H to FF309H also cannot be used with the R5F109xE (x = 6, A, B, G), respectively.

Figure 3-17. Data to Be Saved to Stack Memory



3.2.2 General-purpose registers

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

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

These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

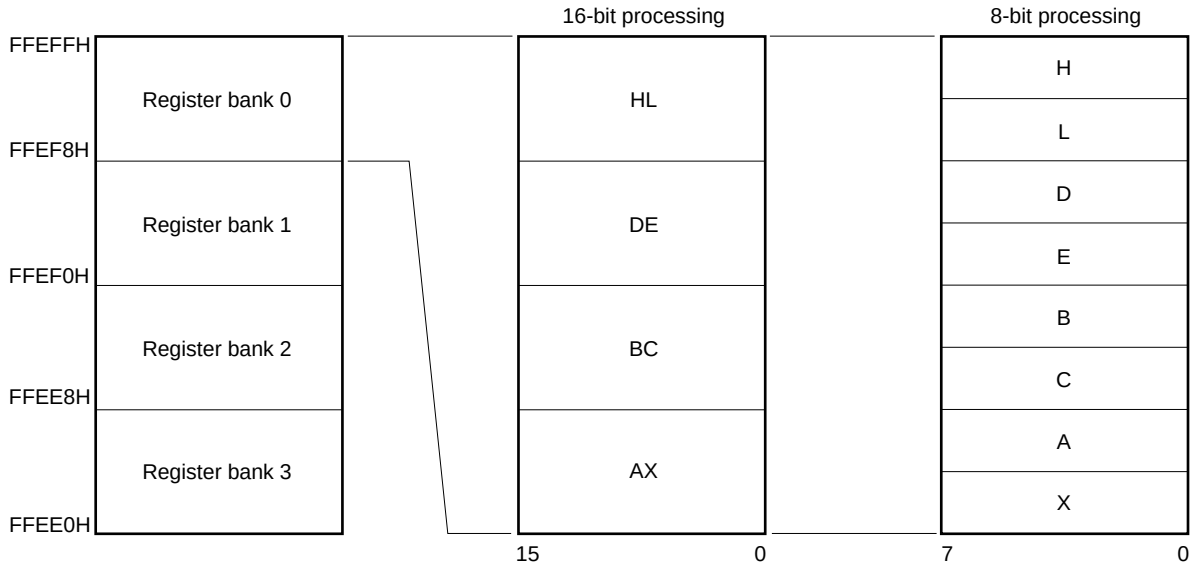
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Cautions 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

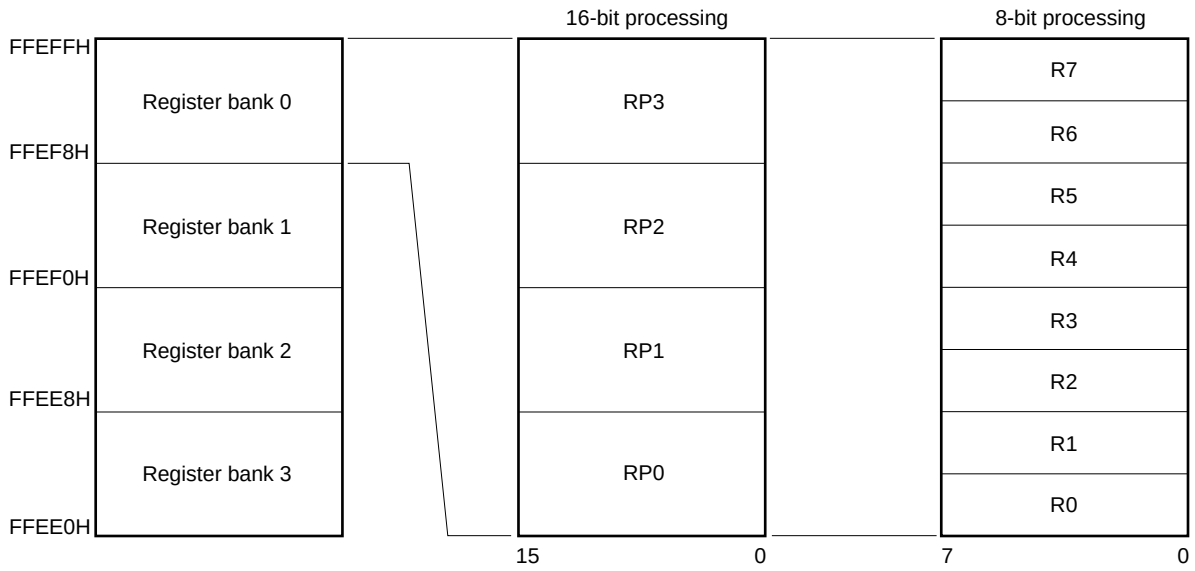
2. While using the self-programming function or data flash function, the area FFE20H to FFEFFH cannot be used as stack memory. Furthermore, the areas of FEF00H to FF309H also cannot be used with the R5F109xE (x = 6, A, B, G), respectively.

Figure 3-18. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



3.2.3 ES and CS registers

The ES register is used for data access and the CS register is used to specify the higher address when a branch instruction is executed.

The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

Figure 3-19. 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	CP2	CP1	CP0

3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each SFR has a special function.

SFRs are allocated to the FFF00H to FFFFFH area.

SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved 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 reserved by the assembler for the 16-bit manipulation instruction operand (sfrp). When specifying an address, describe an even address.

Table 3-5 gives a list of the SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating 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
  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
  Indicates each register status upon reset signal generation.

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

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

Table 3-5. SFR List (1/4)

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
FFF0CH	Port register 12	P12		R/W	√	√	–	Undefined
FFF0DH	Port register 13	P13		R/W	√	√	–	Undefined
FFF0EH	Port register 14	P14		R/W	√	√	–	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	–	√	√	0000H
FFF11H		–			–	–		
FFF12H	Serial data register 01	RXD0	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–		
FFF18H	Timer data register 00	TDR00		R/W	–	–	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	–	√	√	00H
FFF1BH		TDR01H			–	√		00H
FFF1EH	10-bit A/D conversion result register	ADCR		R	–	–	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	–	√	–	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	–	FFH
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
FFF2CH	Port mode register 12	PM12		R/W	√	√	–	FFH
FFF2EH	Port mode register 14	PM14		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
FFF37H	Key return mode register	KRM		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

Table 3-5. SFR List (2/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF44H	Serial data register 02	TXD1	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	Serial data register 03	RXD1/ SIO11	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
FFF48H	Serial data register 10	TXD2/ SIO20	SDR10	R/W	–	√	√	0000H
FFF49H		–			–	–		
FFF4AH	Serial data register 11	RXD2/ 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
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	–	√	√	00H
FFF67H		TDR03H			–	√		00H
FFF68H	Timer data register 04	TDR04		R/W	–	–	√	0000H
FFF69H								
FFF6AH	Timer data register 05	TDR05		R/W	–	–	√	0000H
FFF6BH								
FFF6CH	Timer data register 06	TDR06		R/W	–	–	√	0000H
FFF6DH								
FFF6EH	Timer data register 07	TDR07		R/W	–	–	√	0000H
FFF6FH								
FFF90H	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

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 3-5. SFR List (3/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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
FFFA8H	Reset control flag register	RESF		R	–	√	–	Undefined Note 1
FFFA9H	Voltage detection register	LVIM		R/W	√	√	–	00H Note 2
FFFAAH	Voltage detection level register	LVIS		R/W	√	√	–	00H/01H/81H Note 3
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1A/9A Note 4
FFFACH	CRC input register	CRCIN		R/W	–	√	–	00H
FFFB0H	DMA SFR address register 0	DSA0		R/W	–	√	–	00H
FFFB1H	DMA SFR address register 1	DSA1		R/W	–	√	–	00H
FFFB2H	DMA RAM address register 0L	DRA0L	DRA0	R/W	–	√	√	00H
FFFB3H	DMA RAM address register 0H	DRA0H		R/W	–	√		00H
FFFB4H	DMA RAM address register 1L	DRA1L	DRA1	R/W	–	√	√	00H
FFFB5H	DMA RAM address register 1H	DRA1H		R/W	–	√		00H
FFFB6H	DMA byte count register 0L	DBC0L	DBC0	R/W	–	√	√	00H
FFFB7H	DMA byte count register 0H	DBC0H		R/W	–	√		00H
FFFB8H	DMA byte count register 1L	DBC1L	DBC1	R/W	–	√	√	00H
FFFB9H	DMA byte count register 1H	DBC1H		R/W	–	√		00H
FFFBAH	DMA mode control register 0	DMC0		R/W	√	√	–	00H
FFFBBH	DMA mode control register 1	DMC1		R/W	√	√	–	00H
FFFBCH	DMA operation control register 0	DRC0		R/W	√	√	–	00H
FFFBDH	DMA operation control register 1	DRC1		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

Notes 1. The reset value of the RESF register varies depending on the reset source.

2. The reset value of the LVIM register varies depending on the reset source.

3. The reset value of the LVIS register varies depending on the reset source and the setting of the option byte.

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

Table 3-5. SFR List (4/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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
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 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	Interrupt mask flag register 0H	MK0H		R/W	√	√		FFH
FFFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	Interrupt mask flag register 1H	MK1H		R/W	√	√		FFH
FFFE8H	Priority specification flag register 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	Priority specification flag register 00H	PR00H		R/W	√	√		FFH
FFFEAH	Priority specification flag register 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
FFFF0H	Multiplication/division data register A (L)	MDAL/MULA		R/W	–	–	√	0000H
FFFF1H								
FFFF2H	Multiplication/division data register A (H)	MDAH/MULB		R/W	–	–	√	0000H
FFFF3H								
FFFF4H	Multiplication/division data register B (H)	MDBH/MULOH		R/W	–	–	√	0000H
FFFF5H								
FFFF6H	Multiplication/division data register B (L)	MDBL/MULOL		R/W	–	–	√	0000H
FFFF7H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H

Remark For extended SFRs (2nd SFRs), see **Table 3-6 Extended SFR (2nd SFR) List**.

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

Unlike a general-purpose register, each extended SFR (2nd SFR) has a special function.

Extended SFRs are allocated to the F0000H to F07FFH area. SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Extended SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (!addr16.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved 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 reserved by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Table 3-6 gives a list of the extended SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating 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
  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
  Indicates each register status upon reset signal generation.

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

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

Table 3-6. Extended SFR (2nd SFR) List (1/7)

Address	Special Function Register (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
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H
F0037H	Pull-up resistor option register 7	PU7	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
F0041H	Port input mode register 1	PIM1	R/W	√	√	–	00H
F0045H	Port input mode register 5	PIM5	R/W	√	√	–	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	–	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	–	00H
F0055H	Port output mode register 5	POM5	R/W	√	√	–	00H
F0057H	Port output mode register 7	POM7	R/W	√	√	–	00H
F0060H	Port mode control register 0	PMC0	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
F0073H	Input switch control register	ISC	R/W	√	√	–	00H
F0074H	Timer input select register 0	TIS0	R/W	–	√	–	00H
F0076H	A/D port configuration register	ADPC	R/W	–	√	–	00H
F0077H	Peripheral I/O redirection register	PIOR	R/W	–	√	–	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	–	√	–	00H
F0090H	Data flash control register	DFLCTL	R/W	√	√	–	00H
F00A0H	On-chip high-speed oscillator trimming register	HIOTRM	R/W	–	√	–	**Note**
F00A8H	On-chip high-speed oscillator divider setting register	HOCODIV	R/W	–	√	–	Undefined
F00ACH	Temperature trimming register 0	TEMPCAL0	R	–	√	–	**Note**
F00ADH	Temperature trimming register 1	TEMPCAL1	R	–	√	–	**Note**
F00AEH	Temperature trimming register 2	TEMPCAL2	R	–	√	–	**Note**
F00AFH	Temperature trimming register 3	TEMPCAL3	R	–	√	–	**Note**

Note This value varies depending on the products.

Remark For SFRs in the SFR area, see **Table 3-5 SFR List**.

Table 3-6. Extended SFR (2nd SFR) List (2/7)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F00E0H	Multiplication/division data register C (L)	MDCL		R	–	–	√	0000H
F00E2H	Multiplication/division data register C (H)	MDCH		R	–	–	√	0000H
F00E8H	Multiplication/division control register	MDUC		R/W	√	√	–	00H
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	–	00H
F00F3H	Operation speed mode control register	OSMC		R/W	–	√	–	00H
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	–	00H
F00FEH	BCD adjust result register	BCDADJ		R	–	√	–	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–	–		
F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–	–		
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		–			–	–		
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	–	–	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	–	–	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	–	–	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	–	–	√	0087H
F011FH								

Remark For SFRs in the SFR area, see **Table 3-5 SFR List**.

Table 3-6. Extended SFR (2nd SFR) List (3/7)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		–			–	–		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		–			–	–		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		–			–	–		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	–	√	√	0000H
F0127H		–			–	–		
F0128H	Serial output register 0	SO0		R/W	–	–	√	0303H
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		–			–	–		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	–	√	√	0000H
F0135H		–			–	–		
F0138H	Serial standby control register 0	SSC0L	SSC0	R/W	–	√	√	0000H
		–			–	–		
F0140H	Serial status register 10	SSR10L	SSR10	R	–	√	√	0000H
F0141H		–			–	–		
F0142H	Serial status register 11	SSR11L	SSR11	R	–	√	√	0000H
F0143H		–			–	–		
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		–			–	–		
F0150H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	–	–	√	0020H
F0153H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	–	–	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	–	–	√	0087H
F015BH								
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		–			–	–		

Remark For SFRs in the SFR area, see **Table 3-5 SFR List**.

Table 3-6. Extended SFR (2nd SFR) List (4/7)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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								
F0188H	Timer counter register 04	TCR04		R	–	–	√	FFFFH
F0189H								
F018AH	Timer counter register 05	TCR05		R	–	–	√	FFFFH
F018BH								
F018CH	Timer counter register 06	TCR06		R	–	–	√	FFFFH
F018DH								
F018EH	Timer counter register 07	TCR07		R	–	–	√	FFFFH
F018FH								
F0190H	Timer mode register 00	TMR00		R/W	–	–	√	0000H
F0191H								
F0192H	Timer mode register 01	TMR01		R/W	–	–	√	0000H
F0193H								
F0194H	Timer mode register 02	TMR02		R/W	–	–	√	0000H
F0195H								
F0196H	Timer mode register 03	TMR03		R/W	–	–	√	0000H
F0197H								
F0198H	Timer mode register 04	TMR04		R/W	–	–	√	0000H
F0199H								
F019AH	Timer mode register 05	TMR05		R/W	–	–	√	0000H
F019BH								
F019CH	Timer mode register 06	TMR06		R/W	–	–	√	0000H
F019DH								
F019EH	Timer mode register 07	TMR07		R/W	–	–	√	0000H
F019FH								
F01A0H	Timer status register 00	TSR00L	TSR00	R	–	√	√	0000H
F01A1H		–			–	–		
F01A2H	Timer status register 01	TSR01L	TSR01	R	–	√	√	0000H
F01A3H		–			–	–		

Remark For SFRs in the SFR area, see **Table 3-5 SFR List**.

Table 3-6. Extended SFR (2nd SFR) List (5/7)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01A4H	Timer status register 02	TSR02L	TSR02	R	–	√	√	0000H
F01A5H		–			–	–		
F01A6H	Timer status register 03	TSR03L	TSR03	R	–	√	√	0000H
F01A7H		–			–	–		
F01A8H	Timer status register 04	TSR04L	TSR04	R	–	√	√	0000H
F01A9H		–			–	–		
F01AAH	Timer status register 05	TSR05L	TSR05	R	–	√	√	0000H
F01ABH		–			–	–		
F01ACH	Timer status register 06	TSR06L	TSR06	R	–	√	√	0000H
F01ADH		–			–	–		
F01AEH	Timer status register 07	TSR07L	TSR07	R	–	√	√	0000H
F01AFH		–			–	–		
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		–			–	–		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		–			–	–		
F01B4H	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		–			–	–		
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
F02F0H	Flash memory CRC control register	CRC0CTL		R/W	√	√	–	00H
F02F2H	Flash memory CRC operation result register	PGCRCL		R/W	–	–	√	0000H
F02FAH	CRC data register	CRCD		R/W	–	–	√	0000H

Remark For SFRs in the SFR area, see **Table 3-5 SFR List**.

Table 3-6. Extended SFR (2nd SFR) List (6/7)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0500H	Peripheral enable register	PERX	R/W	√	√	–	00H
F0501H	Peripheral clock select register	PCKSEL	R/W	√	√	–	00H
F0504H	Port mode register X0	PMX0	R/W	√	√	–	01H
F0505H	Port mode register X1	PMX1	R/W	√	√	–	01H
F0506H	Port mode register X2	PMX2	R/W	√	√	–	01H
F0507H	Port mode register X3	PMX3	R/W	√	√	–	01H
F0508H	Port mode register X4	PMX4	R/W	√	√	–	01H
F0509H	Port input enable register X	PIEN	R/W	√	√	–	00H
F050AH	Noise filter enable register X	NFENX	R/W	√	√	–	00H
F0520H	LIN-UART0 control register 0	UF0CTL0	R/W	√	√	–	10H
F0521H	LIN-UART0 option register 0	UF0OPT0	R/W	√	√	–	14H
F0522H	LIN-UART0 control register 1	UF0CTL1	R/W	–	–	√	0FFFH
F0524H	LIN-UART0 option register 1	UF0OPT1	R/W	√	√	–	00H
F0525H	LIN-UART0 option register 2	UF0OPT2	R/W	√	√	–	00H
F0526H	LIN-UART0 status register	UF0STR	R	–	–	√	0000H
F0527H							
F0528H	LIN-UART0 status clear register	UF0STC	R/W	–	–	√	0000H
F0529H							
F052AH	LIN-UART0 8-bit wait transmit data register	UF0WTXB	W	–	√	–	00H
F052BH	LIN-UART0 wait transmit data register	UF0WTX	W	–	–	√	0000H
F052EH	LIN-UART0 ID setting register	UF0ID	R/W	–	√	–	00H
F052FH	LIN-UART0 buffer register 0	UF0BUF0	R/W	–	√	–	00H
F0530H	LIN-UART0 buffer register 1	UF0BUF1	R/W	–	√	–	00H
F0531H	LIN-UART0 buffer register 2	UF0BUF2	R/W	–	√	–	00H
F0532H	LIN-UART0 buffer register 3	UF0BUF3	R/W	–	√	–	00H
F0533H	LIN-UART0 buffer register 4	UF0BUF4	R/W	–	√	–	00H
F0534H	LIN-UART0 buffer register 5	UF0BUF5	R/W	–	√	–	00H
F0535H	LIN-UART0 buffer register 6	UF0BUF6	R/W	–	√	–	00H
F0536H	LIN-UART0 buffer register 7	UF0BUF7	R/W	–	√	–	00H
F0537H	LIN-UART0 buffer register 8	UF0BUF8	R/W	–	√	–	00H
F0538H	LIN-UART0 buffer control register	UF0BUCTL	R/W	–	–	√	0000H
F0539H							
F0540H	Serial data register S0	SDRS0L SDRS0	R/W	–	√	√	0000H
F0541H		–		–	–		
F0542H	Serial data register S1	SDRS1L SDRS1	R/W	–	√	√	0000H
F0543H		–		–	–		
F0548H	LIN-UART0 8-bit transmit data register	UF0TXB UF0TX	R/W	–	√	√	00H
F0549H	LIN-UART0 transmit data register	–		–	–		0000H
F054AH	LIN-UART0 8-bit receive data register	UF0RXB UF0RX	R	–	√	√	00H
F054BH	LIN-UART0 receive data register	–		–	–		0000H

Remark For SFRs in the SFR area, see **Table 3-5 SFR List**.

Table 3-6. Extended SFR (2nd SFR) List (7/7)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0550H	Serial status register S0	SSRS0L	SSRS0	R	–	√	√	0000H
F0551H		–			–	–		
F0552H	Serial status register S1	SSRS1L	SSRS1	R	–	√	√	0000H
F0553H		–			–	–		
F0554H	Serial flag clear trigger register S0	SIRS0L	SIRS0	R/W	–	√	√	0000H
F0555H		–			–	–		
F0556H	Serial flag clear trigger register S1	SIRS1L	SIRS1	R/W	–	√	√	0000H
F0557H		–			–	–		
F0558H	Serial mode register S0	SMRS0		R/W	–	–	√	0020H
F0559H								
F055AH	Serial mode register S1	SMRS1		R/W	–	–	√	0020H
F055BH								
F055CH	Serial communication operation setting register S0	SCRS0		R/W	–	–	√	0087H
F055DH								
F055EH	Serial communication operation setting register S1	SCRS0		R/W	–	–	√	0087H
F055FH								
F0560H	Serial channel enable status register S	SESL	SES	R	√	√	√	0000H
F0561H		–			–	–		
F0562H	Serial channel start register S	SSSL	SSS	R/W	√	√	√	0000H
F0563H		–			–	–		
F0564H	Serial channel stop register S	STSL	STS	R/W	√	√	√	0000H
F0565H		–			–	–		
F0566H	Serial clock select register S	SPSSL	SPSS	R/W	–	√	√	0000H
F0567H		–			–	–		
F0568H	Serial output register S	SOS		R/W	–	–	√	0303H
F0569H								
F056AH	Serial output enable register S	SOESL	SOES	R/W	√	√	√	0000H
F056BH		–			–	–		
F0570H	Serial output level register S	SOLSL	SOLS	R/W	–	√	√	0000H
F0571H		–			–	–		
F0580H	WUTM control register	WUTMCTL		R/W	√	√	–	00H
F0582H	WUTM compare register	WUTMCMP		R/W	–	–	√	0000H
F0583H								

Remark For SFRs in the SFR area, see **Table 3-5 SFR List**.

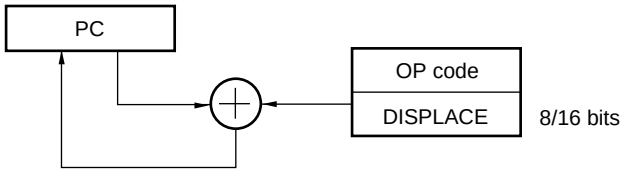
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

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

Figure 3-20. Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

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

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

Figure 3-21. Example of CALL !!addr20/BR !!addr20

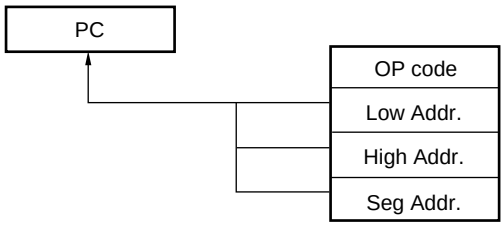
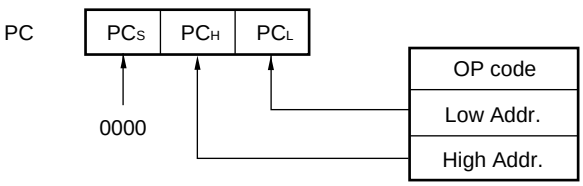


Figure 3-22. Example of CALL !addr16/BR !addr16



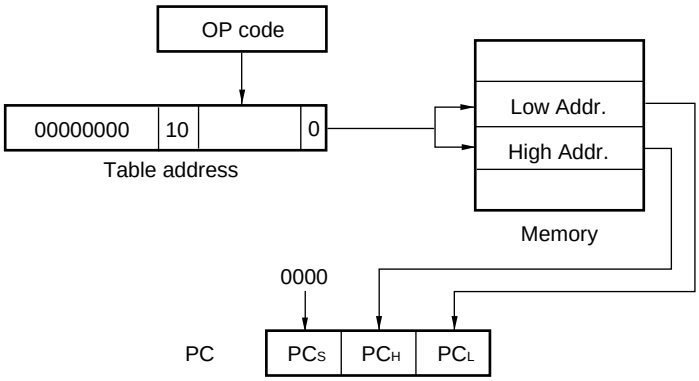
3.3.3 Table indirect addressing

[Function]

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

In the RL78 microcontrollers, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

Figure 3-23. Outline of Table Indirect Addressing

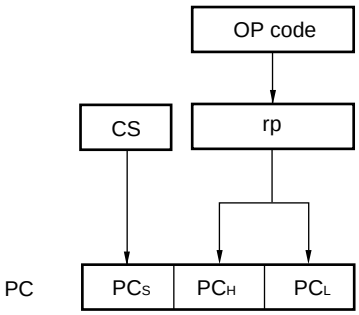


3.3.4 Register direct addressing

[Function]

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

Figure 3-24. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

[Function]

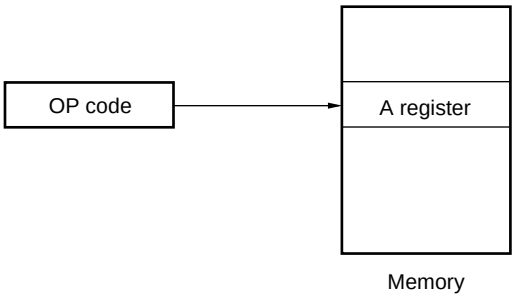
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

Because implied addressing can be automatically employed with an instruction, no particular operand format is necessary.

Implied addressing can be applied only to MULU X.

Figure 3-25. Outline of Implied Addressing



3.4.2 Register addressing

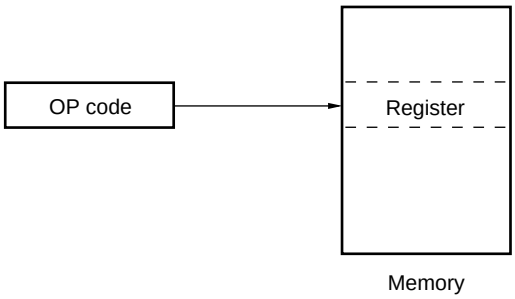
[Function]

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

[Operand format]

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

Figure 3-26. Outline of Register Addressing



3.4.3 Direct addressing

[Function]

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

[Operand format]

Identifier	Description
ADDR16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES: ADDR16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3-27. Example of ADDR16

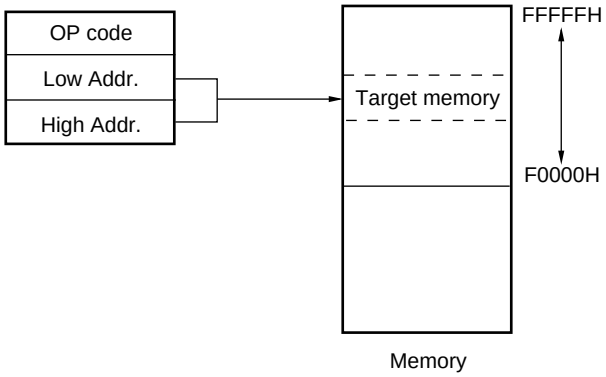
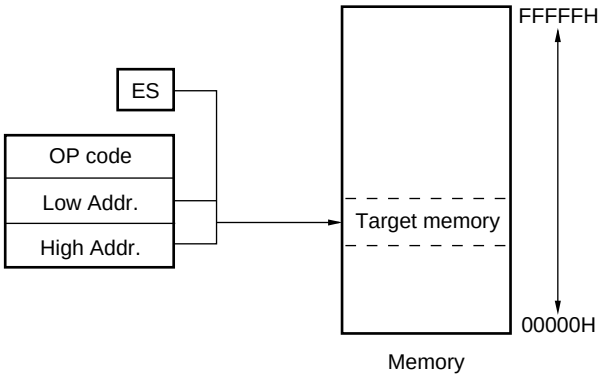


Figure 3-28. Example of ES:ADDR16



3.4.4 Short direct addressing

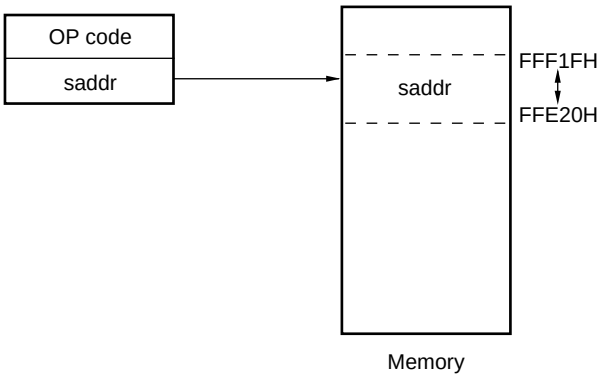
[Function]

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

[Operand format]

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

Figure 3-29. Outline of Short Direct Addressing



Remark SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data.

Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

3.4.5 SFR addressing

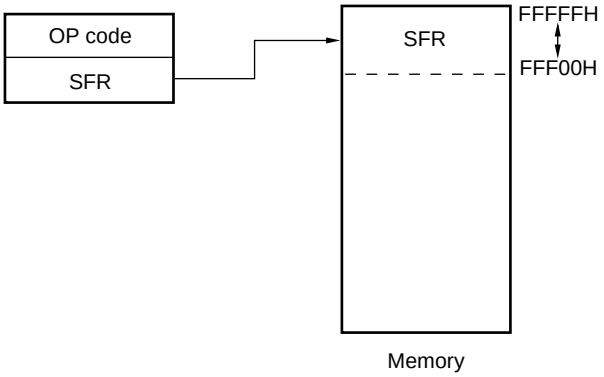
[Function]

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

[Operand format]

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

Figure 3-30. Outline of SFR Addressing



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

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

Figure 3-31. Example of [DE], [HL]

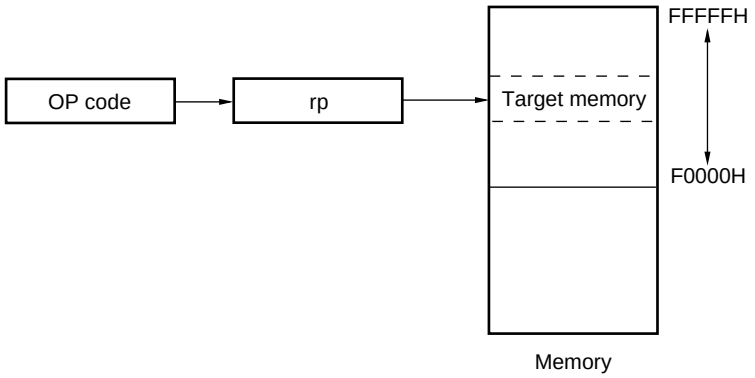
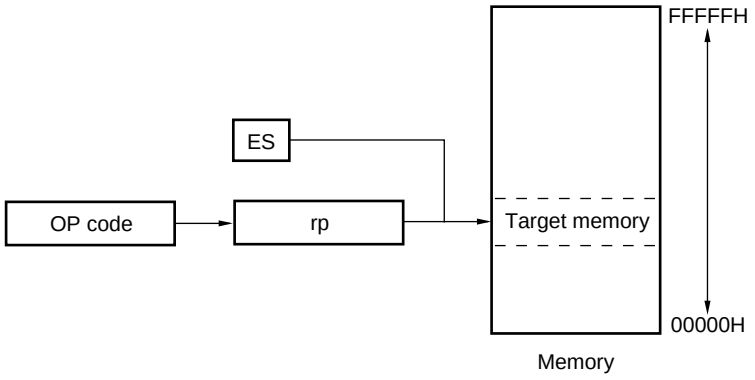


Figure 3-32. Example of ES:[DE], ES:[HL]



3.4.7 Based addressing

[Function]

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

[Operand format]

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

Figure 3-33. Example of [SP+byte]

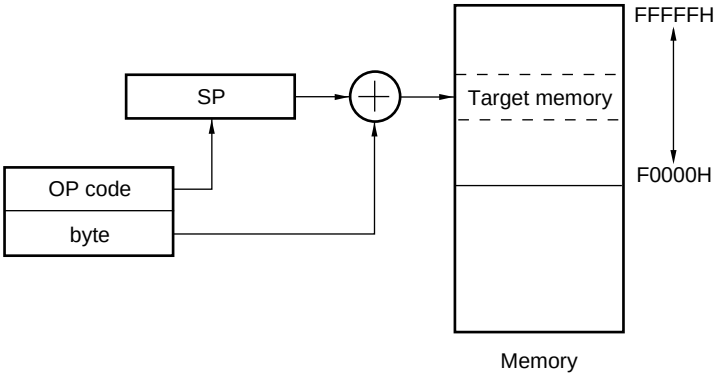


Figure 3-34. Example of [HL + byte], [DE + byte]

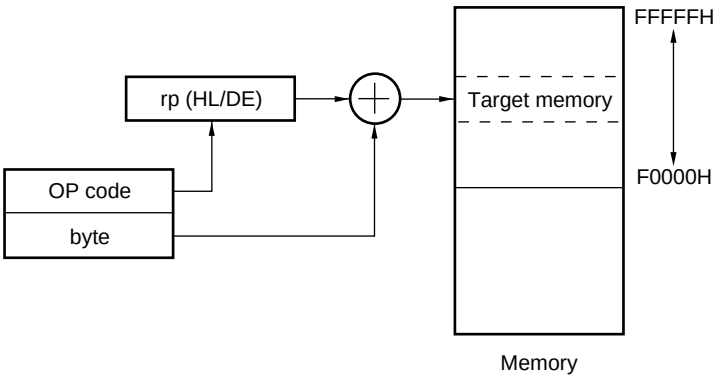


Figure 3-35. Example of word[B], word[C]

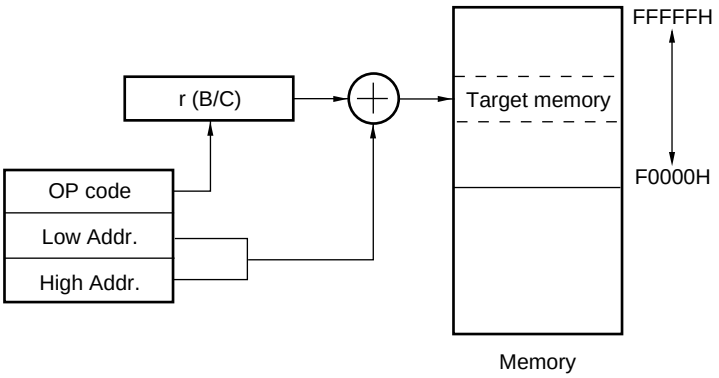


Figure 3-36. Example of word[BC]

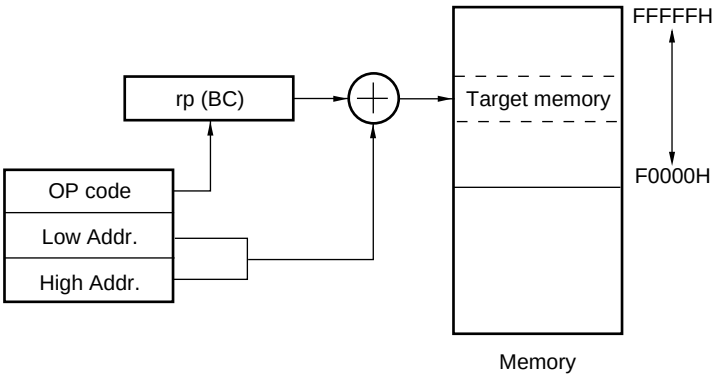


Figure 3-37. Example of ES:[HL + byte], ES:[DE + byte]

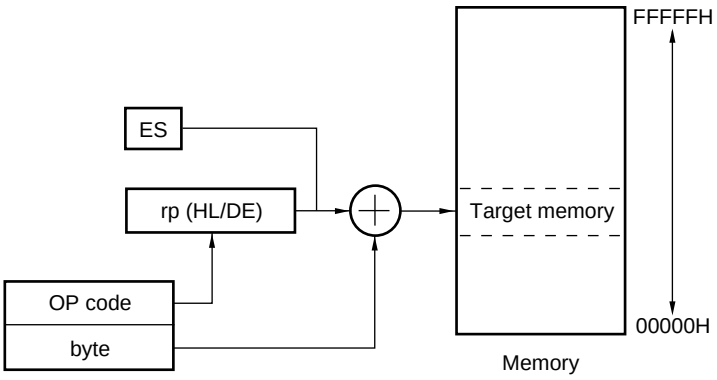


Figure 3-38. Example of ES:word[B], ES:word[C]

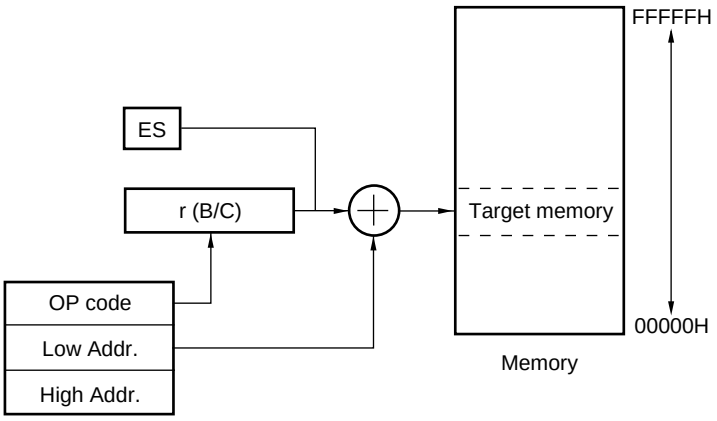
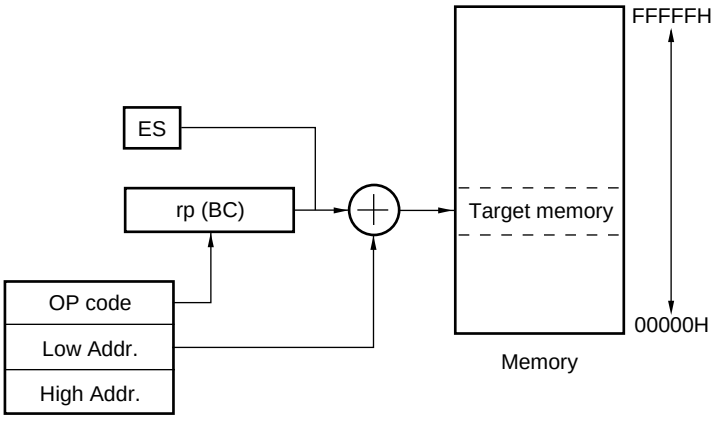


Figure 3-39. Example of ES:word[BC]



3.4.8 Based indexed addressing

[Function]

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

[Operand format]

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

Figure 3-40. Example of [HL+B], [HL+C]

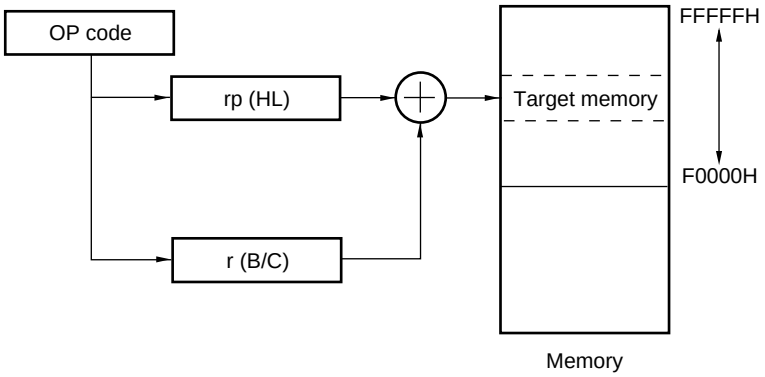
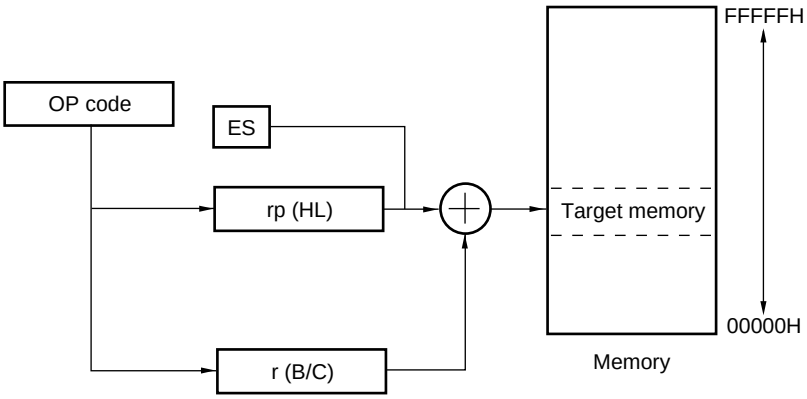


Figure 3-41. Example of ES:[HL+B], ES:[HL+C]



3.4.9 Stack addressing

[Function]

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

Stack addressing is applied only to the internal RAM area.

[Operand format]

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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The RL78/F12 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 2 PIN FUNCTIONS**.

4.2 Port Configuration

Ports include the following hardware.

Table 4-1. Port Configuration

Item	Configuration
Control registers	Port mode registers (PM0 to PM7, PM12, PM14, PMX0 to PMX4) Port registers (P0 to P7, P12-P14) Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU14) Port input mode registers (PIM0, PIM1, PIM5) Port output mode registers (POM0, POM1, POM5, POM7) Port mode control registers (PMC0, PMC12, PMC14) A/D port configuration register (ADPC) Peripheral I/O redirection register (PIOR) Port input enable register (PIEN)
Port	• 20-pin products Total: 16 (CMOS I/O: 13, CMOS input: 3) • 30-pin products Total: 26 (CMOS I/O: 21, CMOS input: 3, N-ch open drain I/O: 2) • 32-pin products Total: 28 (CMOS I/O: 22, CMOS input: 3, N-ch open drain I/O: 3) • 48-pin products Total: 44 (CMOS I/O: 34, CMOS input: 5, CMOS output: 1, N-ch open drain I/O: 4) • 64-pin products Total: 58 (CMOS I/O: 48, CMOS input: 5, CMOS output: 1, N-ch open drain I/O: 4)
Pull-up resistor	• 20-pin products Total: 10 • 30-pin products Total: 17 • 32-pin products Total: 18 • 48-pin products Total: 26 • 64-pin products Total: 40

Caution Most of the following descriptions in this chapter use the 64-pin products with the peripheral I/O redirection register (PIOR) being set to 00H as an example.

4.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P06 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 P01, P03, 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 P00, P02, P03 pins can be specified as N-ch open-drain output (V_{DD} tolerance/EV_{DD} tolerance) in 1-bit units using port output mode register 0 (POM0).

Input to the P00 to P03 pins[Note] can be specified as analog input or digital input in 1-bit units, using port mode control register 0 (PMC0).

This port can also be used for timer I/O, A/D converter analog input, serial interface data I/O, clock I/O.

In 20- to 32-pin products, reset signal generation sets port 0 to analog input.

In 48-pin products, reset signal generation sets port 0 to input mode.

In 64-pin products, reset signal generation sets port 0 to P00, P01, P04 to P06 to analog input and P02/ANI17 and P03/ANI16 to analog input.

For settings of the registers when using port 0, refer to Table 4-2.

Note In 30- and 32-pin products: P00 and P01
In 20-pin products: P00

Table 4-2. Register Settings When Using Port 0

Pins		PM0.x	PIM0.x	POM0.x	PMC0.x	Alternate Function Setting When Using Pin as Port	Remarks
Name	I/O						
P00[Note1]	Input	1	–	×	0	×	
	Output	0		0	0	×	CMOS output
		0		1	0	×	N-ch O.D. output
P01	Input	1	0	–	0	×	CMOS input
		1	1		0	×	TTL input
	Output	0	×		0	TO00 output = 0[Note3]	
P02[Note2]	Input	1	–	×	0	–	
	Output	0		0	0	SO10/TxD1 output = 1 [Note4]	CMOS output
		0		1	0		N-ch O.D. output
P03[Note2]	Input	1	0	×	0	×	CMOS input
		1	1	×	0		TTL input
	Output	0	×	0	0	SDA10 output = 1 [Note4]	CMOS output
		0	×	1	0		N-ch O.D. output
P04[Note2]	Input	1	0	×	–	×	CMOS input
		1	1	×			TTL input
	Output	0	×	0		$\overline{\text{SCK10}}$/SCL10 output = 1 [Note4]	CMOS output
		0	×	1			N-ch O.D. output
P05[Note2]	Input	1	–	–	–	×	
	Output	0				TO05 output = 0[Note3]	
P06[Note2]	Input	1	–	–	–	×	
	Output	0				TO06 output = 0[Note3]	

Important: To use the port 0 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

Notes 1. 30-, 32-, 48-, or 64-pin products only

2. 64-pin products only

3. To use P02/ANI17/SO10/TxD1, P03/ANI16/SI10/RxD1/SDA10, P04/SCK10/SCL10 as a general-purpose port, set serial channel enable status register 0 (SE0), serial output register 0 (SO0) and serial output enable register 0 (SOE0) to the default status. Clear port output mode register 0 (POM0) to 00H.

4. To use P00/TO00, P05/TI05/TO05, P06/TI06/TO06 as a general-purpose port, set bits 0, 5, and 6 (TO0.0, TO0.5, and TO0.6) of timer output register 0 (TO0) and bits 0, 5, and 6 (TOE0.0, TOE0.5, and TOE0.6) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.

Remarks x: don’t care

PM0×: Port mode register 0

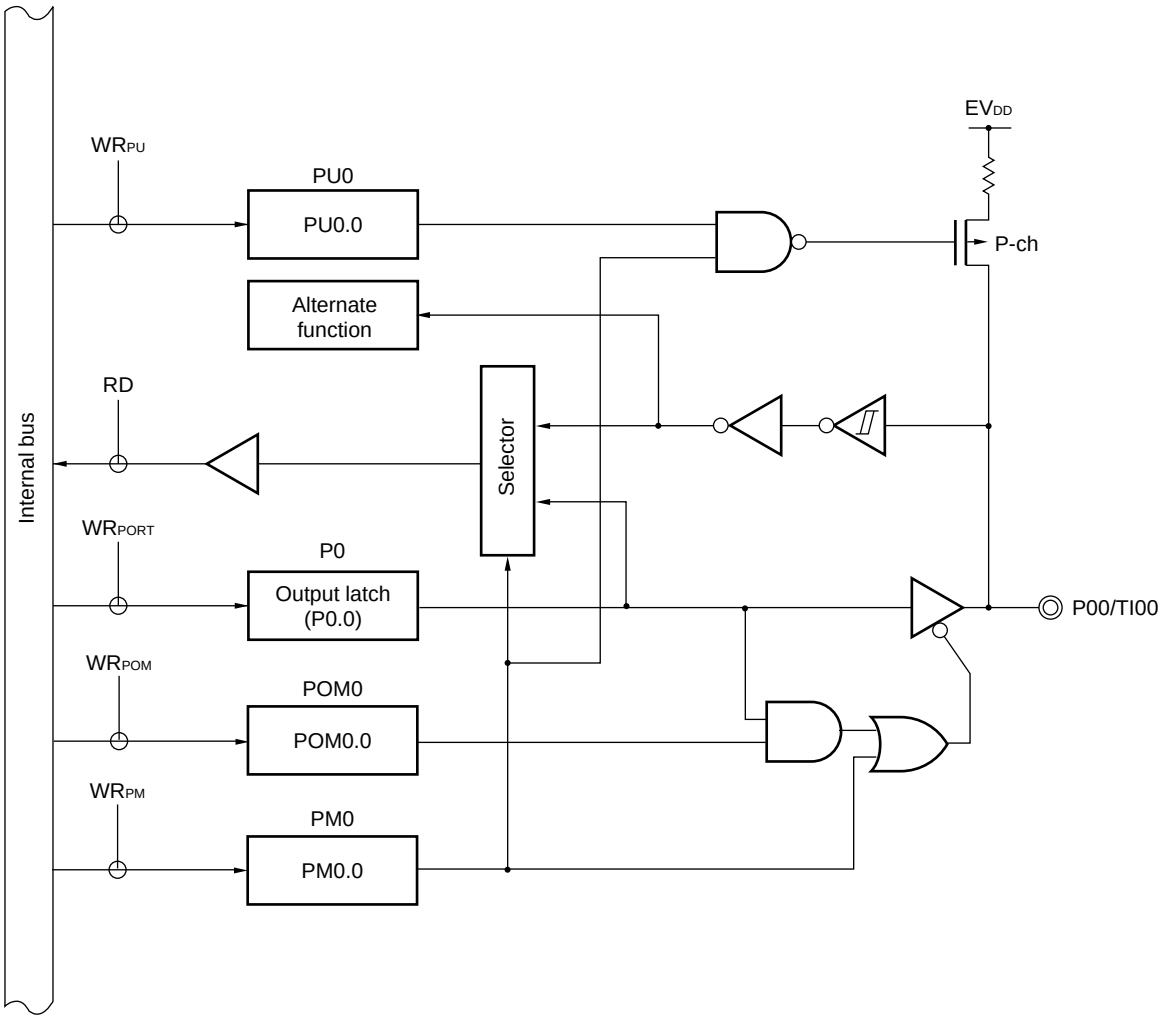
PIM0×: Port input mode register 0

POM0×: Port output mode register 0

PMC0×: Port mode control register 0

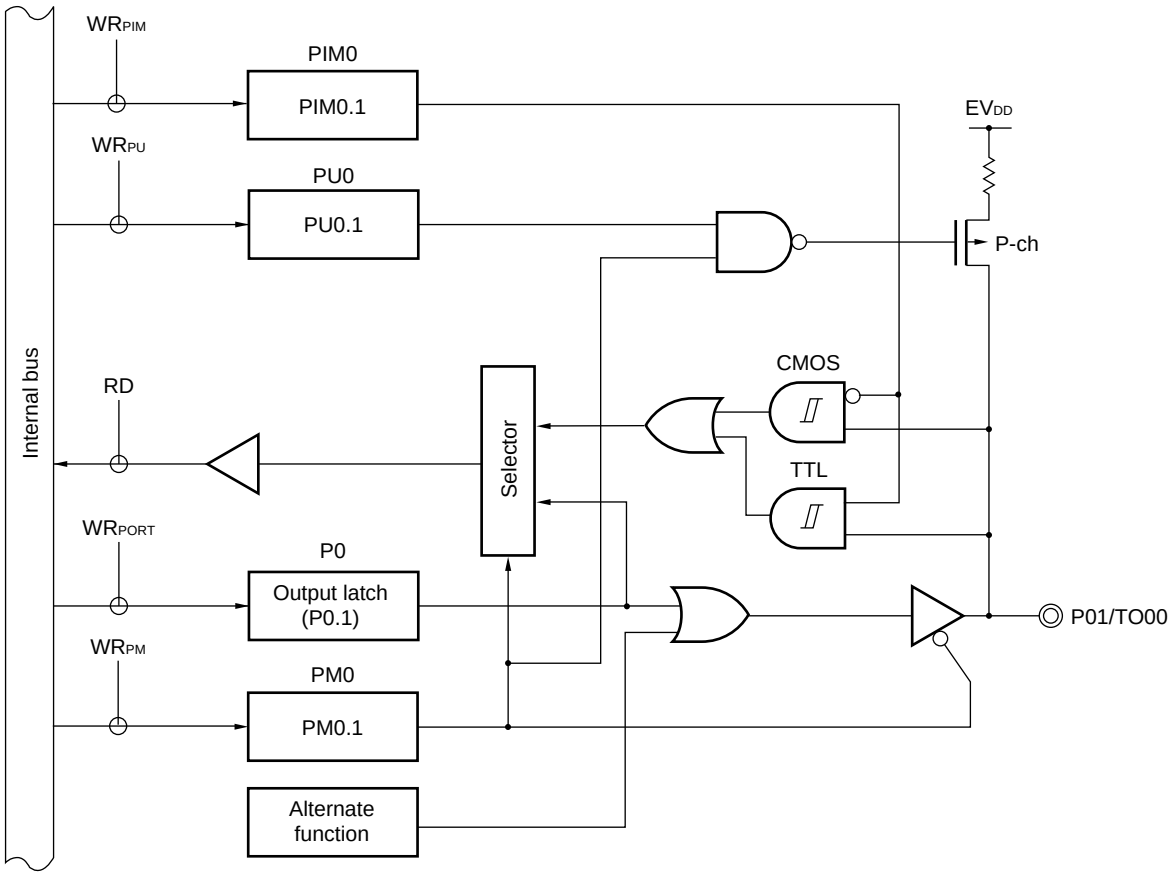
For example, figures 4-1 to 4-6 show block diagrams of port 0 for 64-pin products.

Figure 4-1. Block Diagram of P00



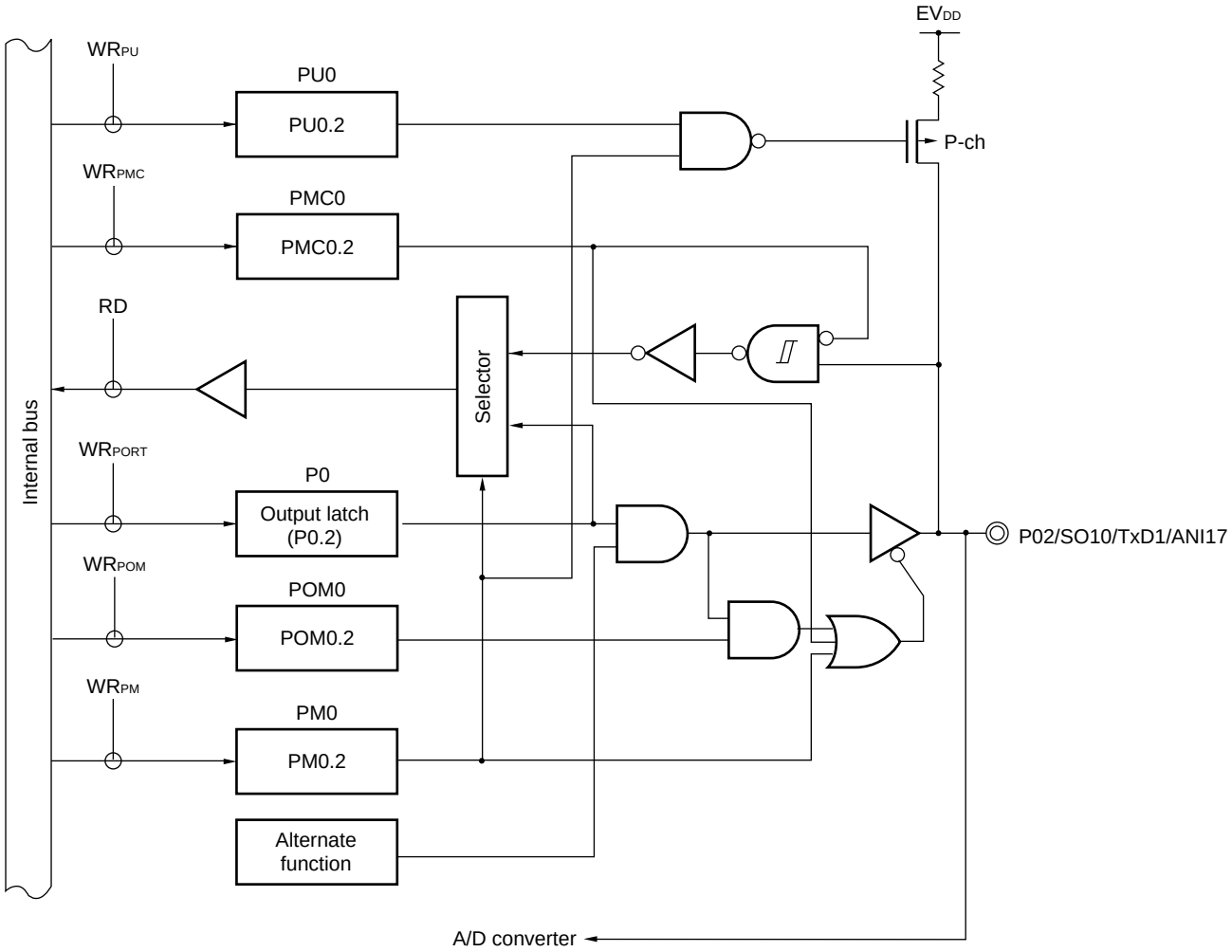
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
POM0: Port output mode register 0
RD: Read signal
WR××: Write signal

Figure 4-2. Block Diagram of P01



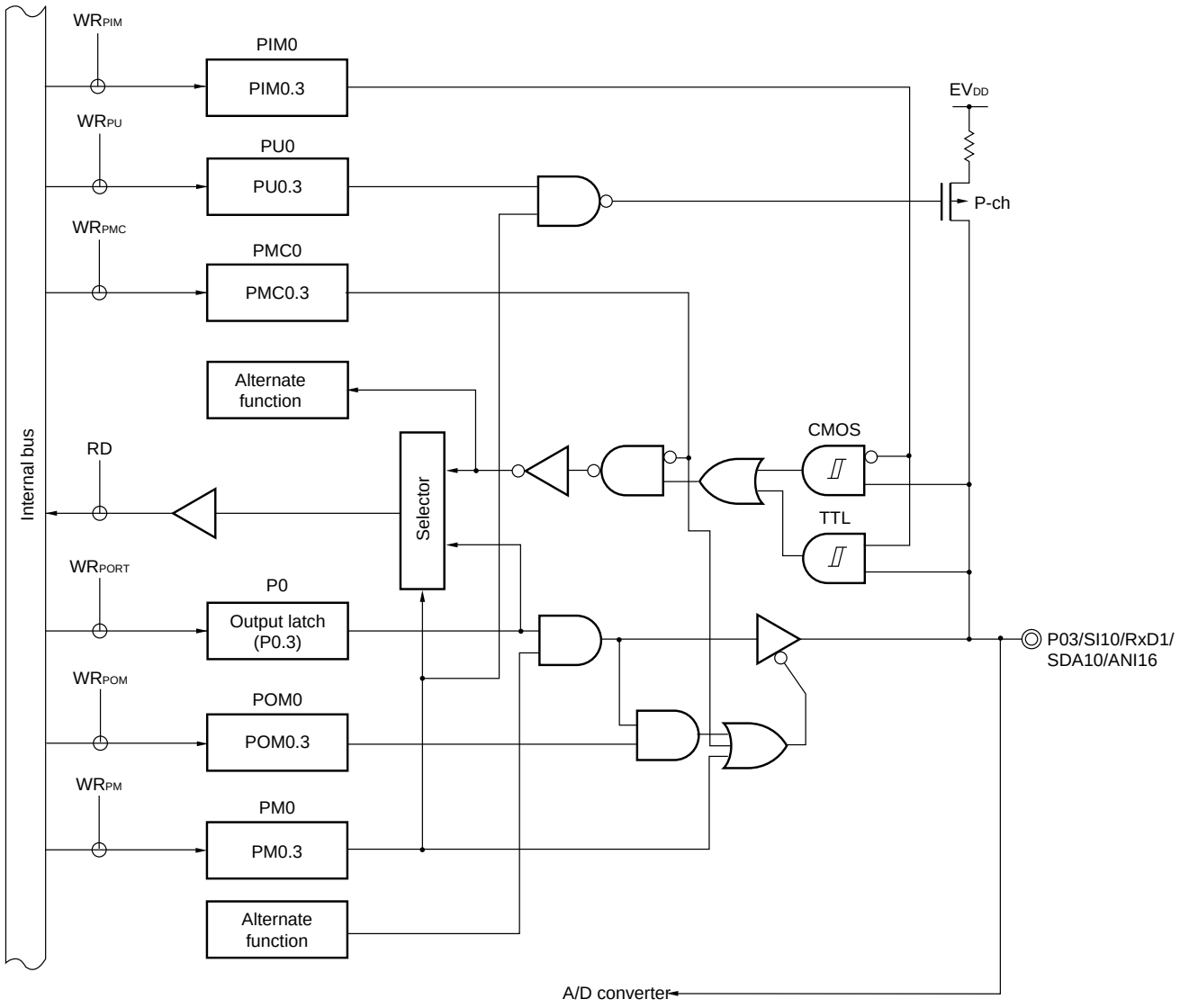
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PIM0: Port input mode register 0
RD: Read signal
WR××: Write signal

Figure 4-3. Block Diagram of P02



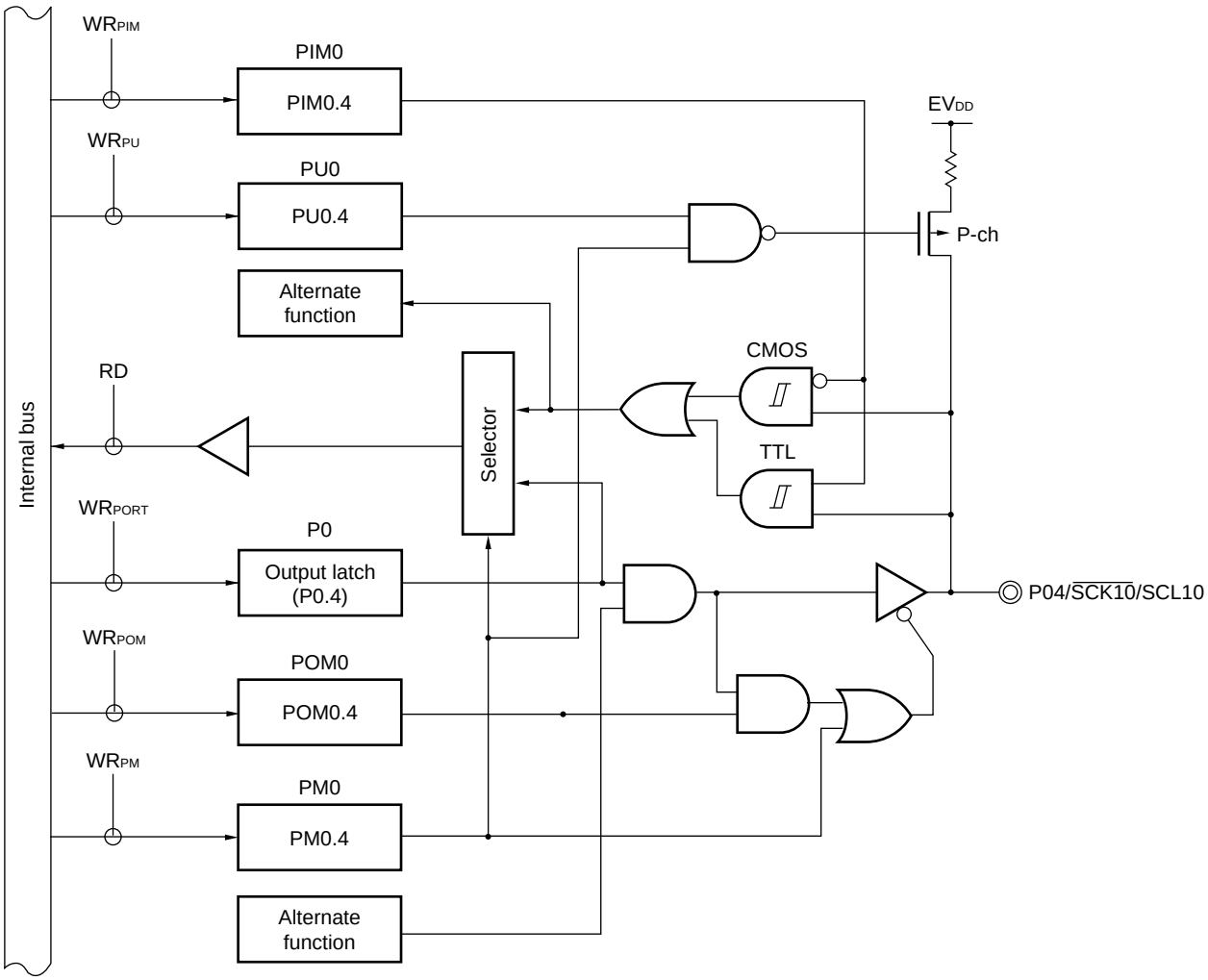
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
POM0: Port output mode register 0
PMC0: Port mode control register 0
RD: Read signal
WR××: Write signal

Figure 4-4. Block Diagram of P03



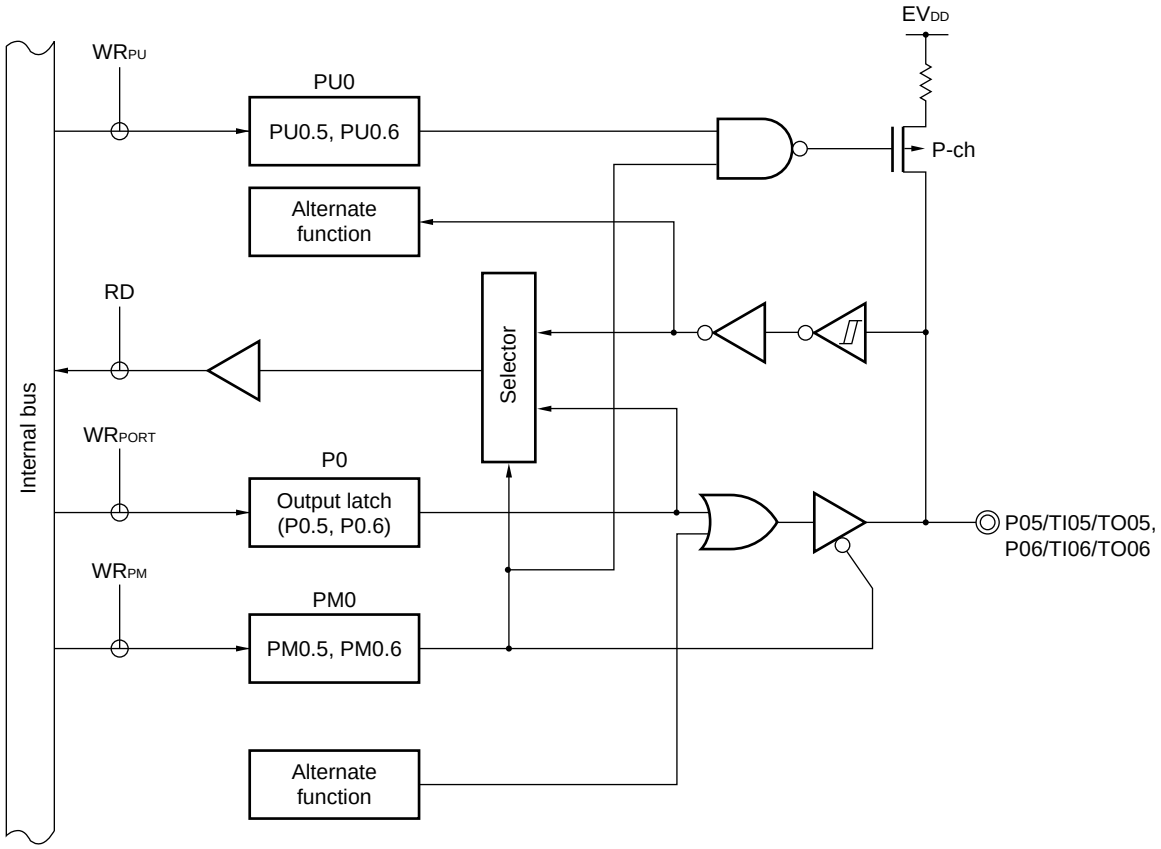
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PIM0: Port input mode register 0
POM0: Port output mode register 0
PMC0: Port mode control register 0
RD: Read signal
WR××: Write signal

Figure 4-5. Block Diagram of P04



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PIM0: Port input mode register 0
POM0: Port output mode register 0
RD: Read signal
WR××: Write signal

Figure 4-6. Block Diagram of P05 and P06



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
RD: Read signal
WR××: Write signal

4.2.2 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 P17 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).

Input to the P13 to P17 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 1 (PIM1).

Output from the P10 to P15 and P17 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 1 (POM1).

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

Reset signal generation sets port 1 to input mode.

For settings of the registers when using port 1, refer to Table 4-3.

Table 4-3. Register Settings When Using Port 1

Pins		PM1.x	PIM1.x	POM1.x	Alternate Function Setting When Using Pin as Port	Remarks
Name	I/O					
P10	Input	1	–	×	×	
	Output	0		0	$\overline{SCK00}$/SCL00 output = 1, ($\overline{TO07}$ output = 0)	CMOS output
		0		1		N-ch O.D. output
P11	Input	1	–	×	×	
	Output	0		0	SDA00 output = 1 (TO06 output = 0)	CMOS output
		0		1		N-ch O.D. output
P12	Input	1	–	×	×	
	Output	0		0	SO00/TxD0 output = 1, (TO05 output = 0)	CMOS output
		0		1		N-ch O.D. output
P13	Input	1	0	×	×	CMOS input
		1	1	×	×	TTL input
	Output	0	×	0	TxD2/SO20 output = 1, (TO04 output = 0, SDAA0 output = 0)	CMOS output
		0	×	1		N-ch O.D. output
P14	Input	1	0	×	×	CMOS input
		1	1	×	×	TTL input
	Output	0	×	0	SDA20 output = 1 (TO03 output = 0, SCLA0 output = 0)	CMOS output
		0	×	1		N-ch O.D. output
P15	Input	1	0	×	×	CMOS input
		1	1	×	×	TTL input
	Output	0	×	0	$\overline{SCK20}$/SCL20 output = 1 (TO02 output = 0)	CMOS output
		0	×	1		N-ch O.D. output
P16	Input	1	0	–	×	CMOS input
		1	1		×	TTL input
	Output	0	×	0	TO01 output = 0	
P17	Input	1	0	×	×	CMOS input
		1	1	×	×	TTL input
	Output	0	×	0	TO02 output = 0 (SO00/TxD0 output = 1)	CMOS output
		0	×	1		N-ch O.D. output

Important To use the port 1 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

Note. To use $\overline{\text{SCKS0}}$ or SOS0/TxDS0 as serial data output or serial clock output, set a bit in the corresponding port mode register (PMxx) for each port to "1". And, set PMX0 and PMX1 registers to "0".

Cautions 1. P10/$\overline{\text{SCK00}}$/$\overline{\text{SCKS0}}$/SCL00, P11/SI00/RxD0/SIS0/RxDS0/SDA00, P12/SO00/TxD0/SOS0/TxDS0, P13/TxD2/SO20, P14/RxD2/SI20/SDA20, or P15/SCK20/SCL20 as a general-purpose port, set serial channel enable status register m (SEm), serial output register m (SOm) and serial output enable register m (SOEm) to the default status (m = 0, 1). Clear port output mode register 1 (POM1) to 00H.

2. To use P16/TI01/TO01 or P17/TI02/TO02 as a general-purpose port, set bits 1 and 2 (TO0.1, TO0.2) of timer output register 0 (TO0) and bits 1 and 2 (TOE0.1, TOE0.2) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

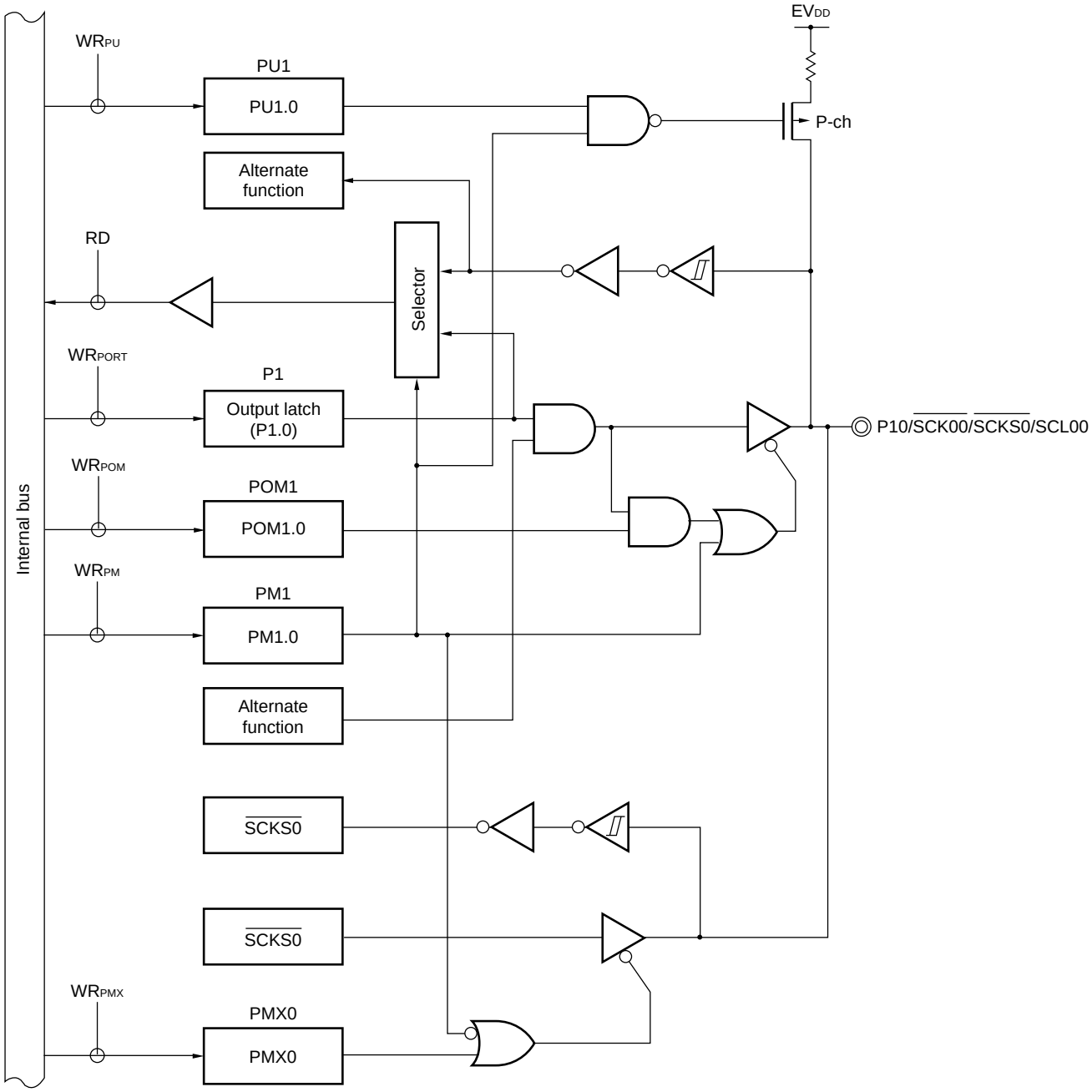
3. If P10 to P15 are used as general-purpose ports and PIOR0 is set to 1, use the corresponding bits in bits 2 to 7 (TO02 to TO07) of timer output register 0 (TO0) and bits 2 to 7 (TOE02 to TOE07) of timer output enable register 0 with "0", which is the same as their initial setting.

4. If P16, P17 are used as general-purpose port and PIOR1 is set to 1, use serial channel enable status register 0 (SE0), serial output register 0 (SO0) and serial output enable register 0 (SOE0) with the same setting as the initial status.

Remark The descriptions in parentheses indicate the case where PIORx = 1.

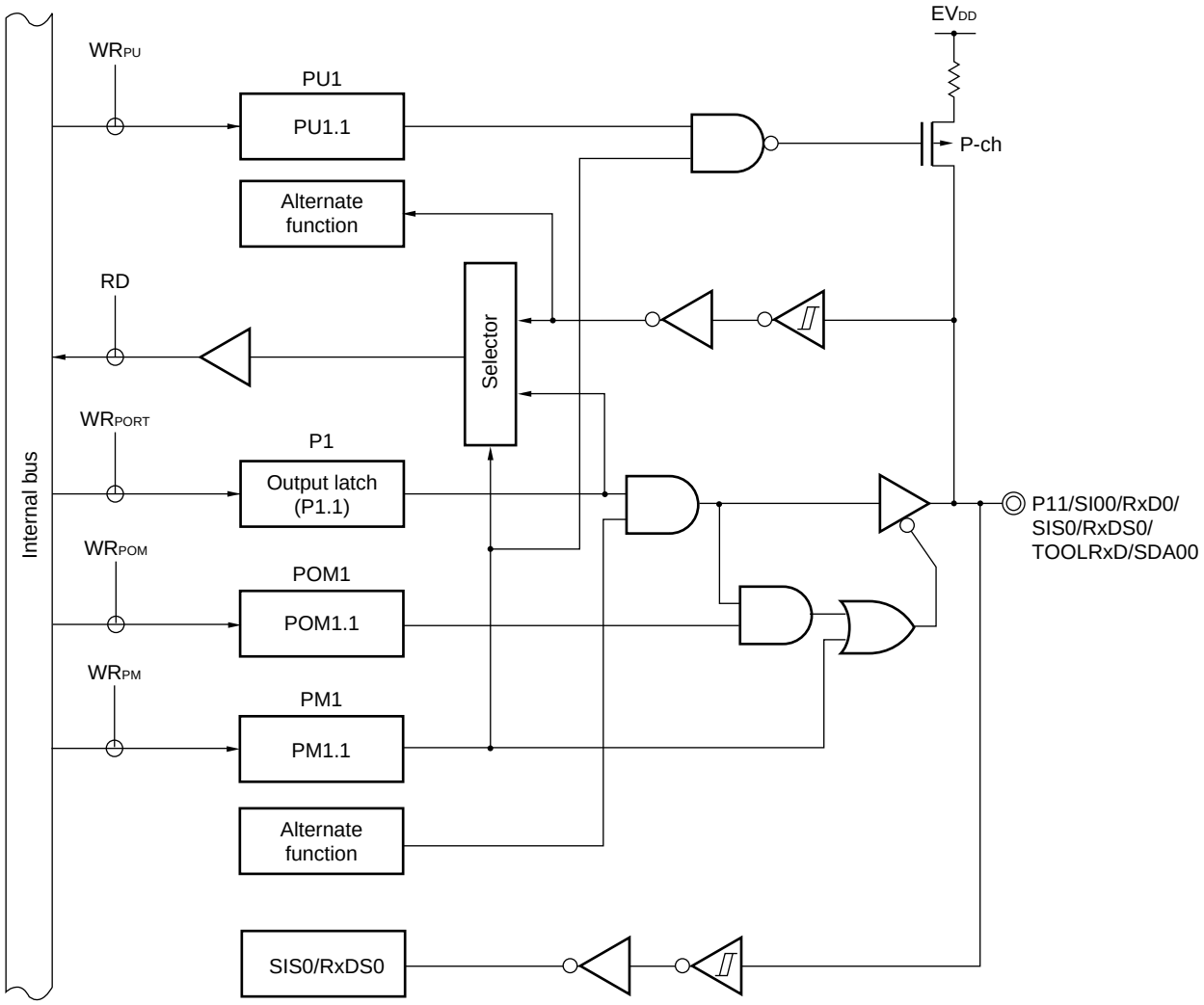
For example, figures 4-7 to 4-14 show block diagrams of port 1 for 64-pin products.

Figure 4-7. Block Diagram of P10



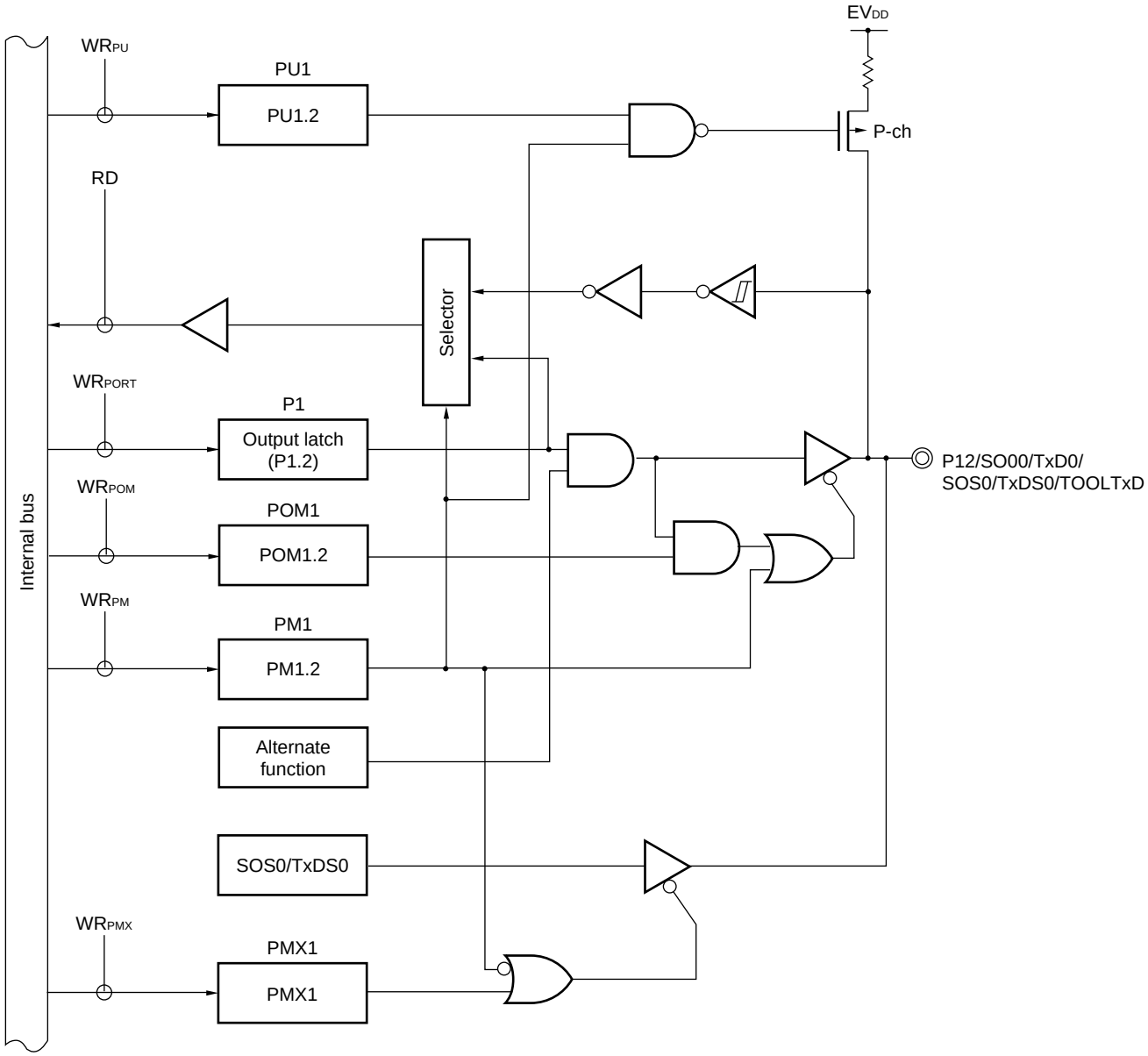
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PMX0: Port mode register X0
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

Figure 4-8. Block Diagram of P11



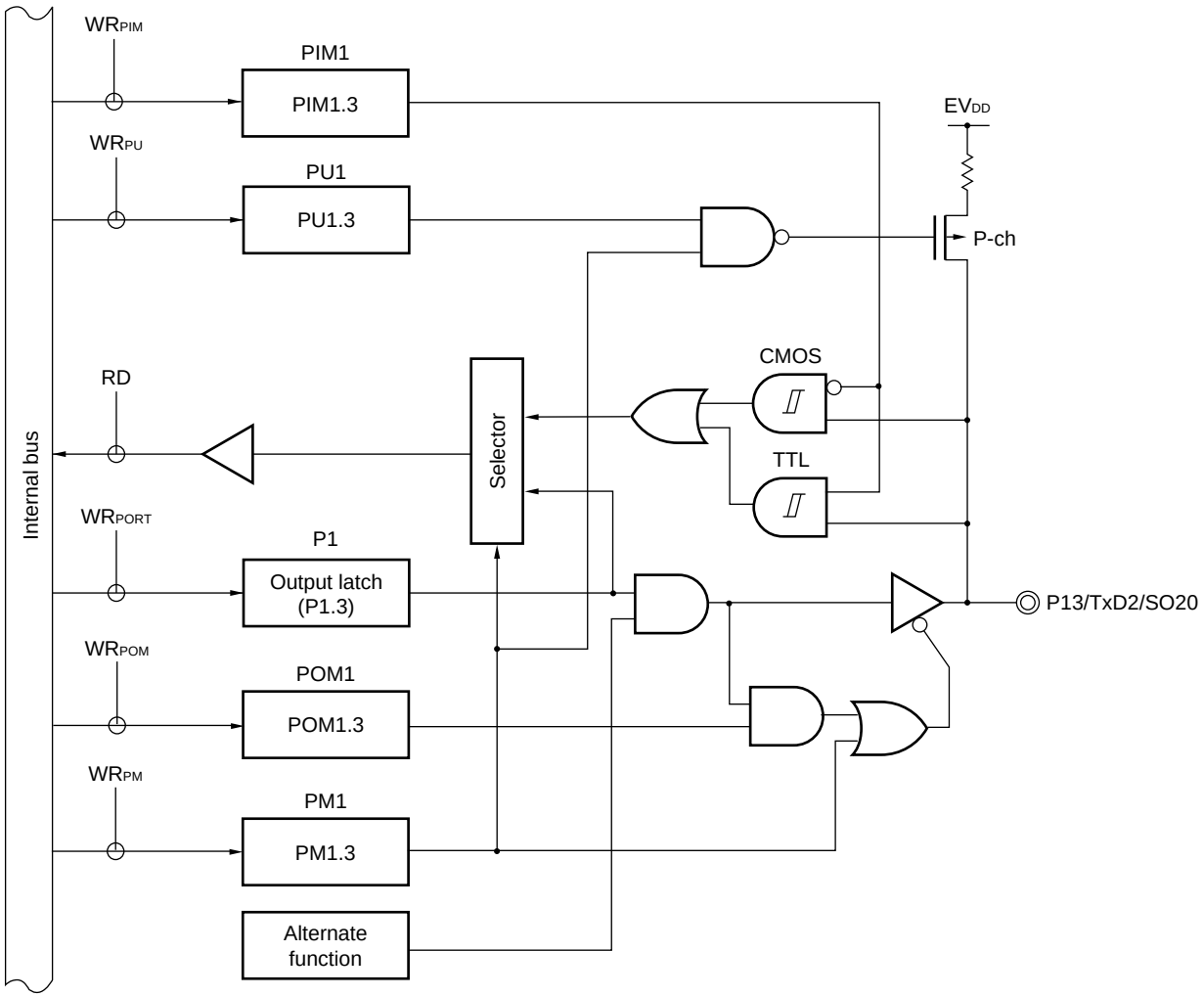
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

Figure 4-9. Block Diagram of P12



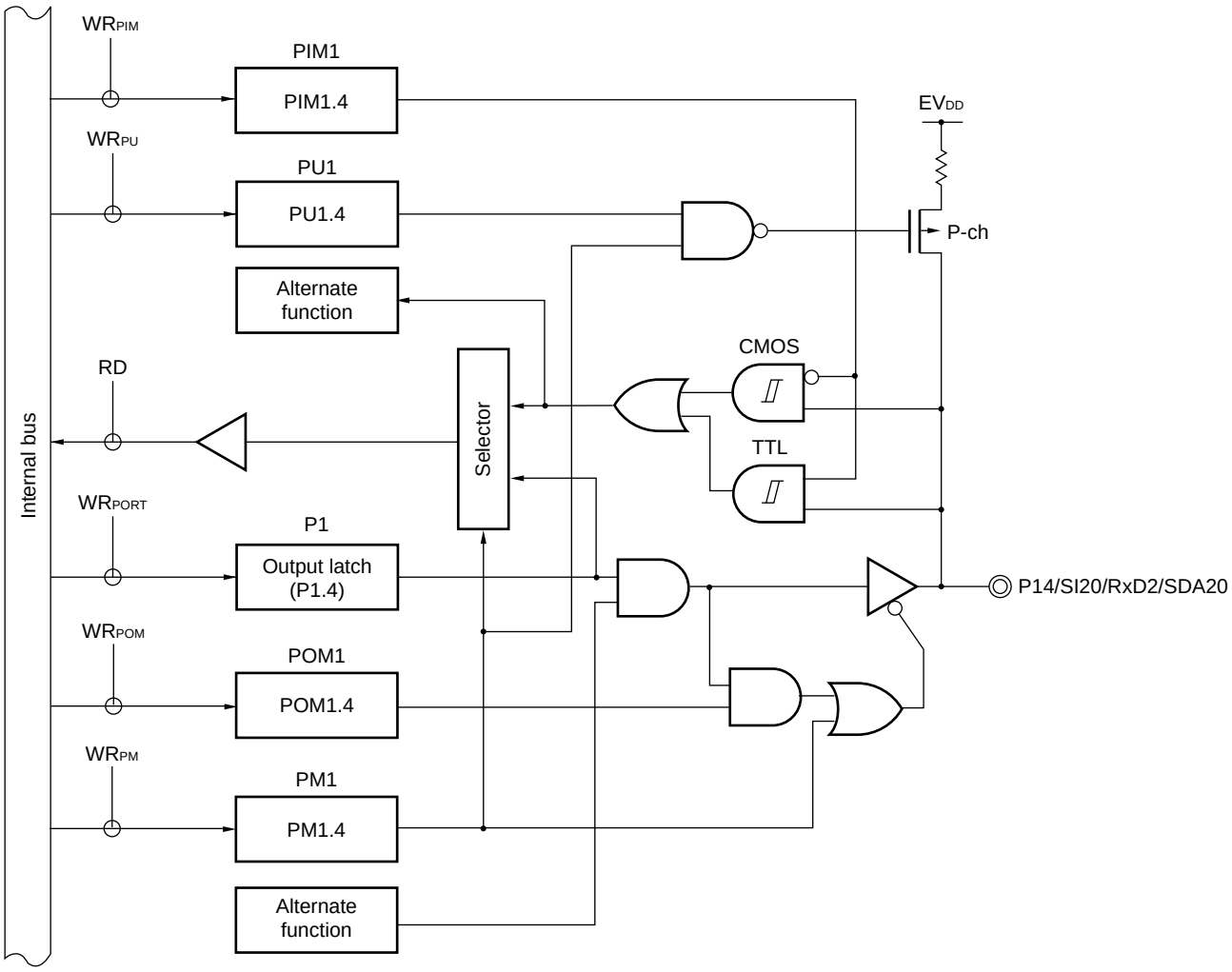
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PMX1: Port mode register X1
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

Figure 4-10. Block Diagram of P13



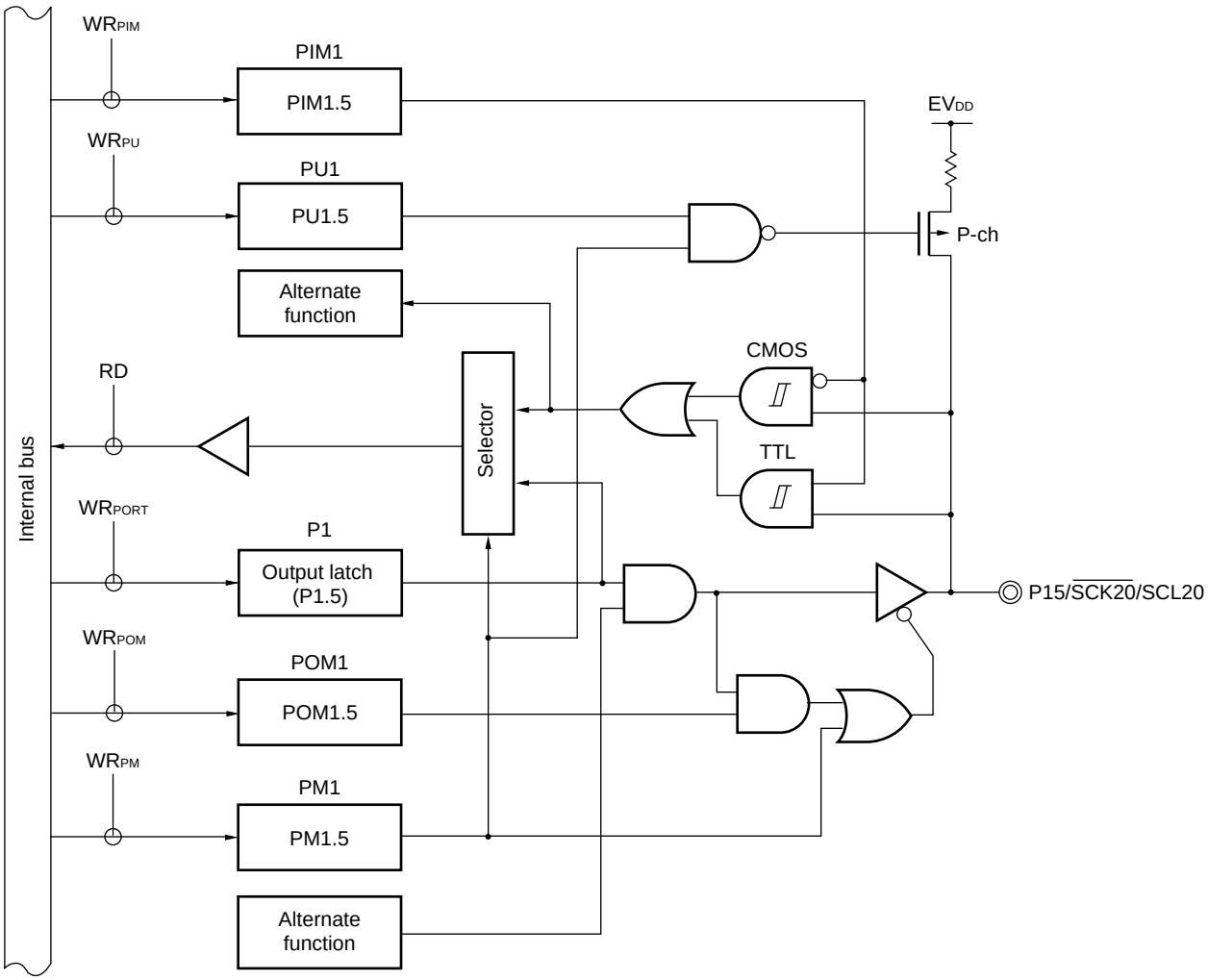
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

Figure 4-11. Block Diagram of P14



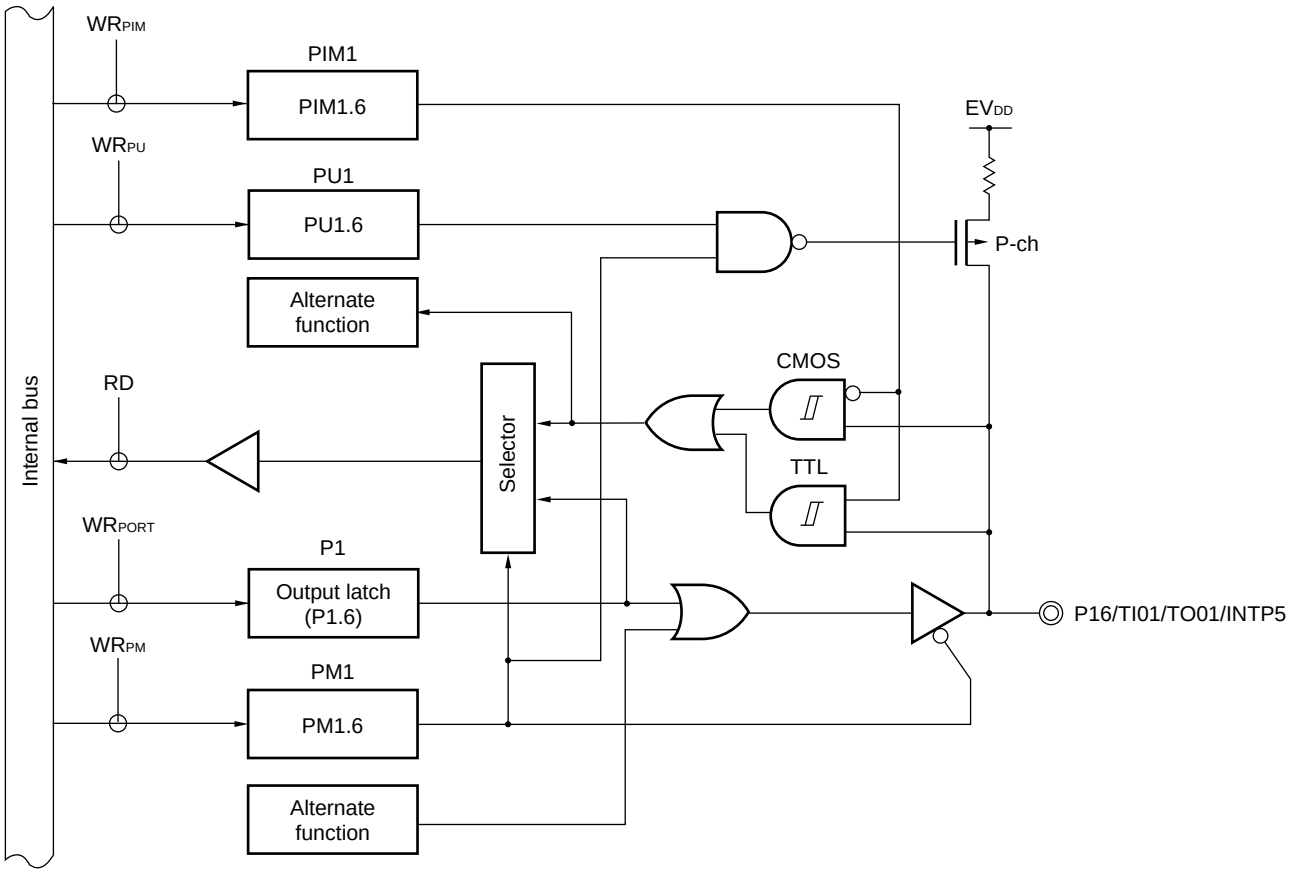
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

Figure 4-12. Block Diagram of P15



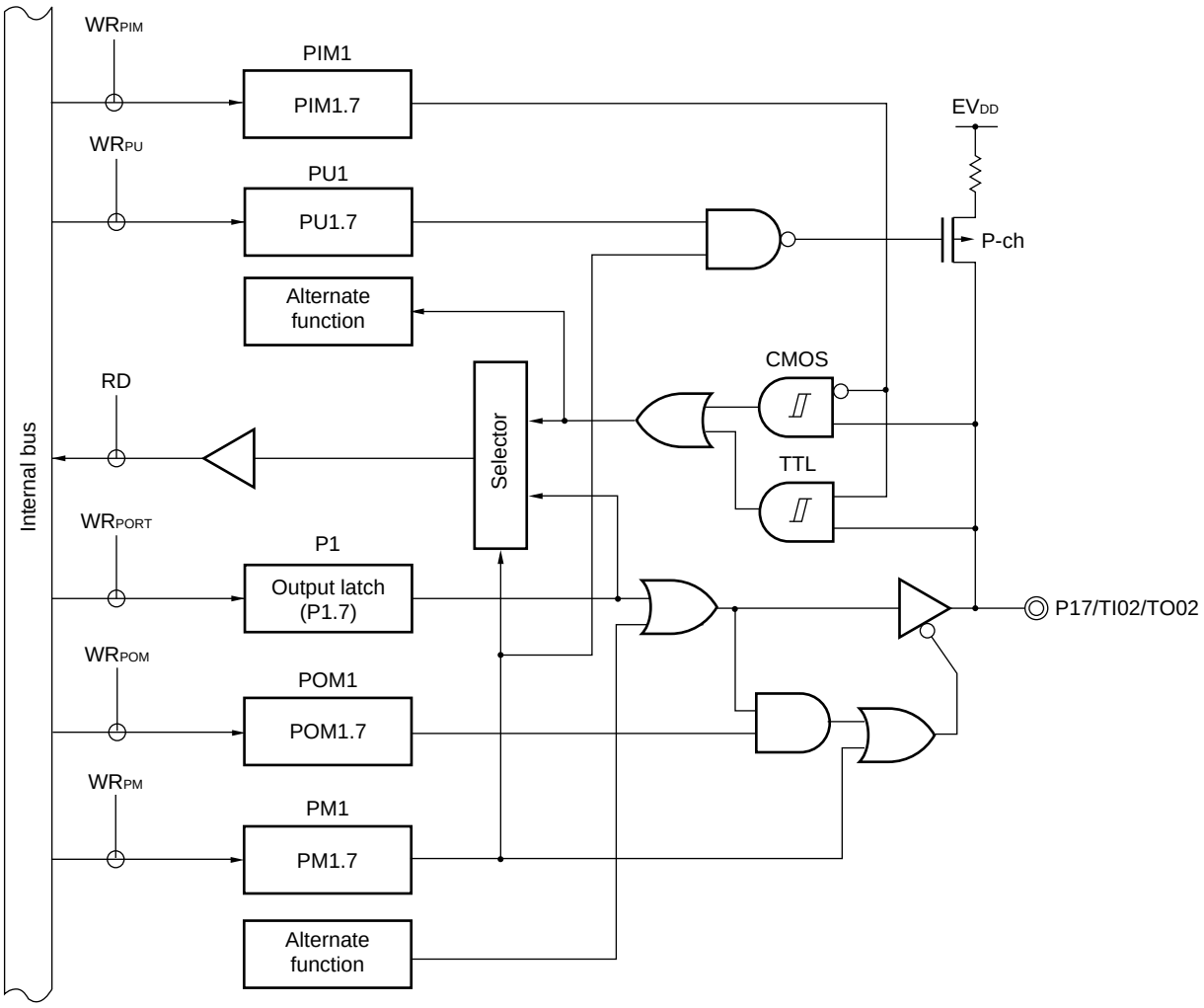
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

Figure 4-13. Block Diagram of P16



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
RD: Read signal
WR××: Write signal

Figure 4-14. Block Diagram of P17



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

4.2.3 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 input.

To use P20/ANI0 to P27/ANI7 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM2 register. Use these pins starting from the upper bit.

To use P20/ANI0 to P27/ANI7 as digital output pins, set them in the digital I/O mode by using the ADPC register and in the output mode by using the PM2 register.

To use P20/ANI0 to P27/ANI7 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM2 register. Use these pins starting from the lower bit.

Table 4-4. Setting Functions of P20/ANI0 to P27/ANI7 Pins

ADPC Register	PM2 Register	ADS Register	P20/ANI0 to P27/ANI7 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input 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 P20/ANI0 to P27/ANI7 are set in the analog input mode when the reset signal is generated.

Figure 4-15 shows a block diagram of port 2.

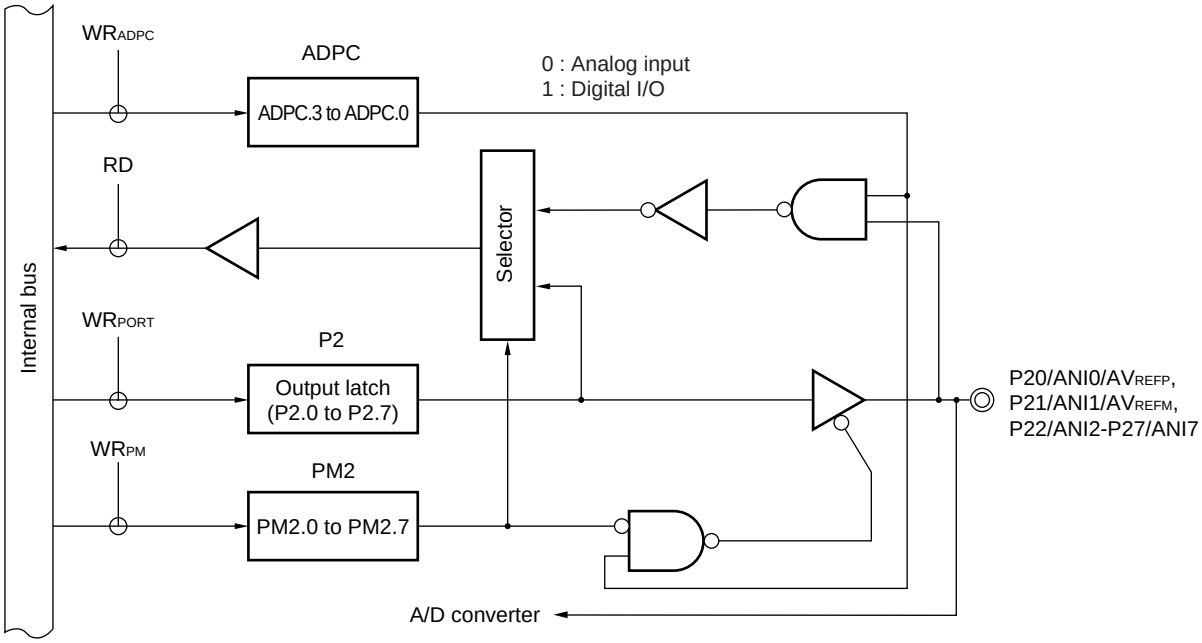
For settings of the registers when using port 2, refer to Table 4-5.

Table 4-5. Register Settings When Using Port 2

Pins		PM2.×	ADPC	Alternate Function Setting When Using Pin as Port	Remarks
Name	I/O				
P2n	Input	1	01 to n+1H	–	Use these pins as a port from the upper bit.
	Output	0	01 to n+1H		

For example, figure 4-15 shows a block diagram of port 2 for 64-pin products.

Figure 4-15. Block Diagram of P20 to P27



P2: Port register 2
PM2: Port mode register 2
ADPC: A/D port configuration register
RD: Read signal
WR××: Write signal

4.2.4 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 an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input, serial interface clock I/O, timer I/O, and real time clock 1 Hz output.

Reset signal generation sets port 3 to input mode.

Figures 4-16 and 4-17 show block diagrams of port 3.

For settings of the registers when using port 3, refer to Table 4-6.

Table 4-6. Register Settings When Using Port 3

Pins		PM3.x	Alternate Function Setting When Using Pin as Port	Remarks
Name	I/O			
P30	Input	1	×	
	Output	0	$\overline{SCK11}$/SCL11 output = 1 Note1 or RTC1HZ output = 0 Note2	
P31	Input	1	×	
	Output	0	TO03 output = 0 Note3 (PCLBUZ0 output = 0) Note3, Note4	

Important To use the port 3 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

Cautions 1. P30/RTC1HZ/INTP3/$\overline{SCK11}$/SCL11 as a general-purpose port, set serial channel enable status register 0 (SE0), serial output register 0 (SO0) and serial output enable register 0 (SOE0) to the default status.

2. To use P31/TI03/TO03/INTP4 as a general-purpose port, set bit 3 (TO0.3) of timer output register 0 (TO0) and bit 3 (TOE0.3) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

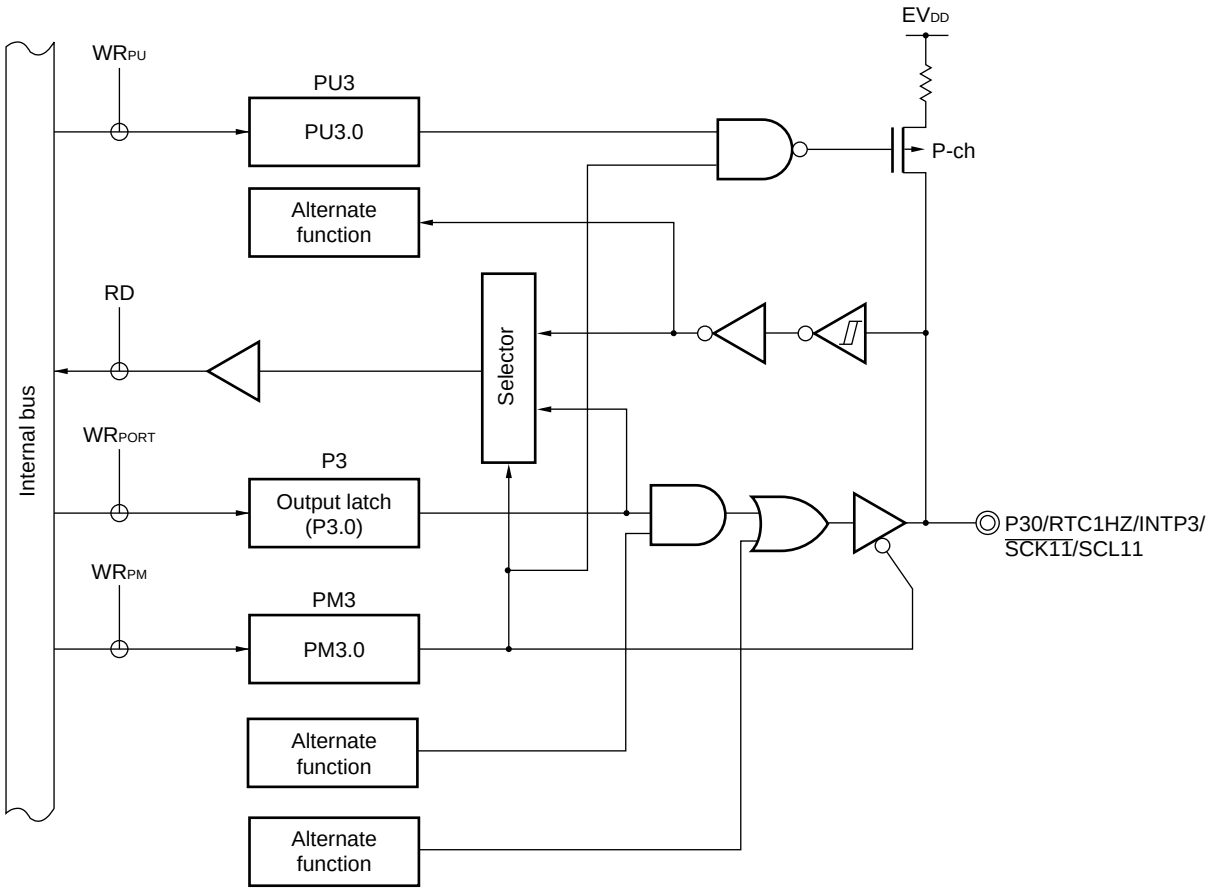
3. To use P31/TI03/TO03/INTP4/PCLBUZ0 as a general-purpose port in 20- to 32-pin products, set bit 7 of the clock output select register 0 (CKS0) to "0", which is the same as their default status setting.

4. To use P31 as a general-purpose port, do not set PIOR3 set to 1.

Remark The descriptions in parentheses indicate the case where PIORx = 1.

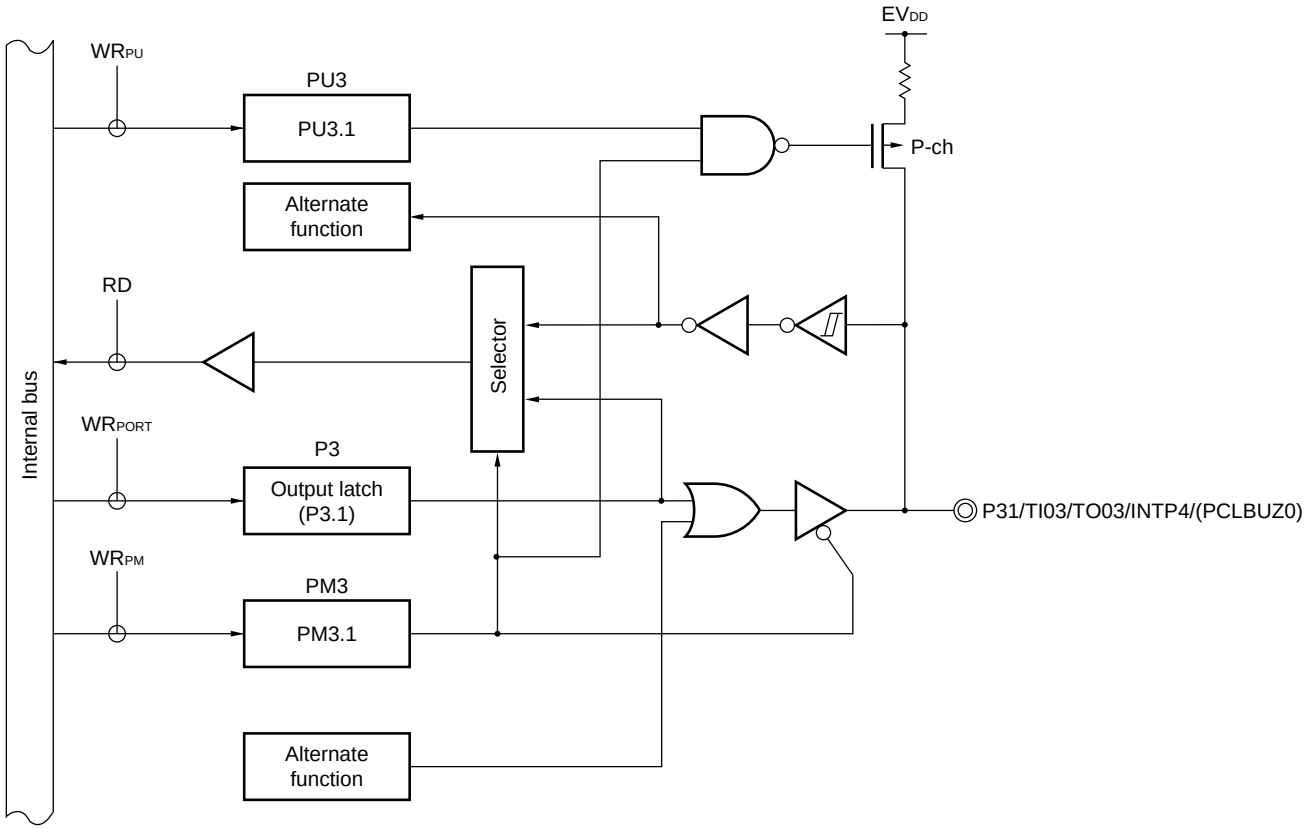
For example, figures 4-16 and 4-17 show block diagrams of port 3 for 64-pin products.

Figure 4-16. Block Diagram of P30



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-17. Block Diagram of P31



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

4.2.5 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 to P43 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 4 (PU4).

This port can also be used for data I/O for a flash memory programmer/debugger, and timer I/O.

Reset signal generation sets port 4 to input mode.

Figures 4-18 to 4-20 show block diagrams of port 4.

For settings of the registers when using port 4, refer to Table 4-7.

Table 4-7. Register Settings When Using Port 4

Pins		PM4.x	Alternate Function Setting When Using Pin as Port	Remarks
Name	I/O			
P40	Input	1	×	
	Output	0	×	
P41	Input	1	×	
	Output	0	TO07 output = 0	
P42	Input	1	×	
	Output	0	TO04 output = 0	
P43	Input	1	×	
	Output	0	×	

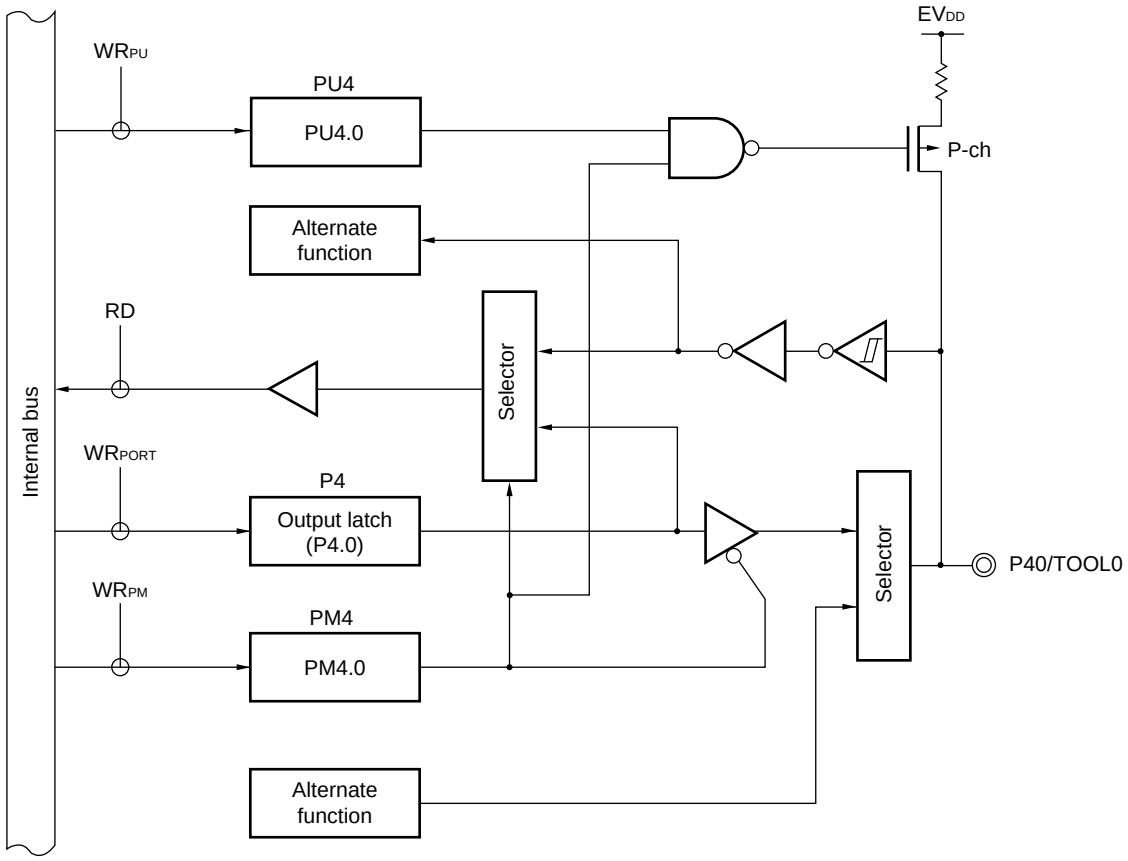
Important To use the port 4 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

Cautions 1. When a tool is connected, the P40 pin cannot be used as a port pin.

2. To use P41/TI07/TO07, P42/TI04/TO04 as a general-purpose port, set bit 4 (TO0.4), bit 7 (TO0.7) of timer output register 0 (TO0) and bit 4 (TOE0.4), bit 7 (TOE0.7) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

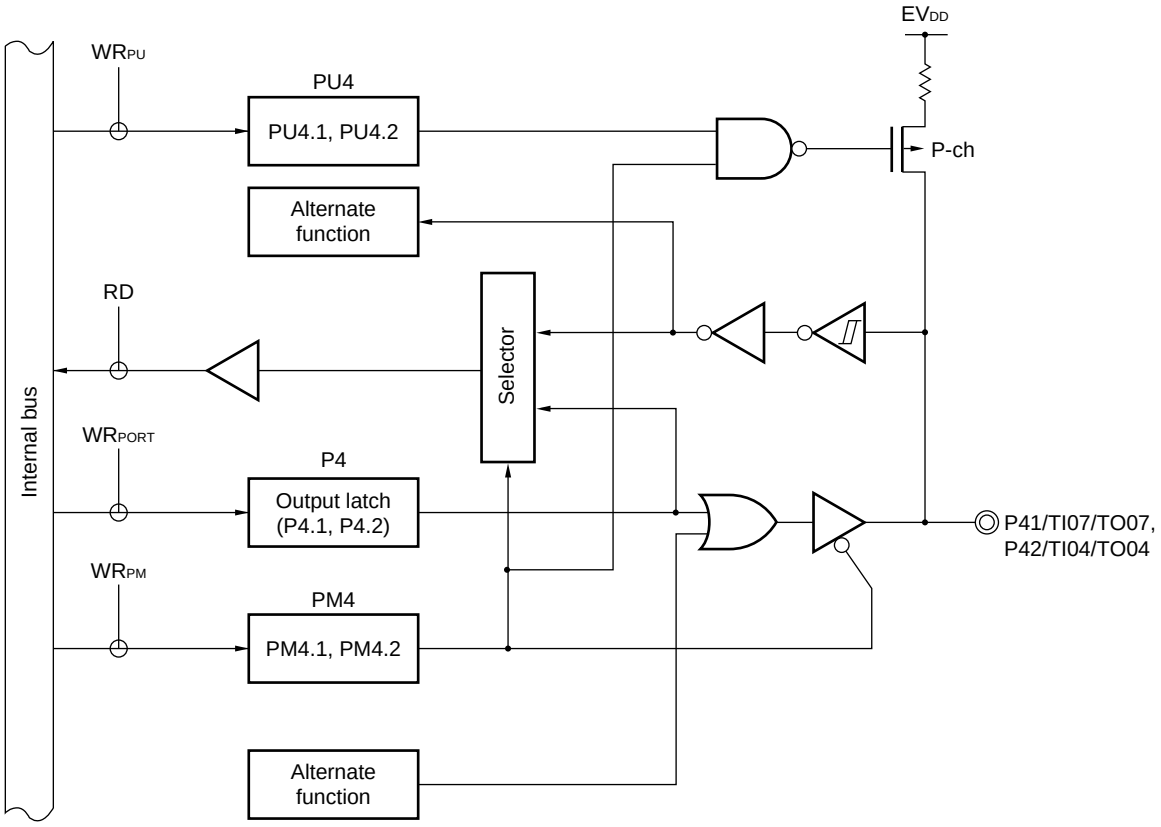
For example, figures 4-18 to 4-20 show block diagrams of port 4 for 64-pin products.

Figure 4-18. Block Diagram of P40



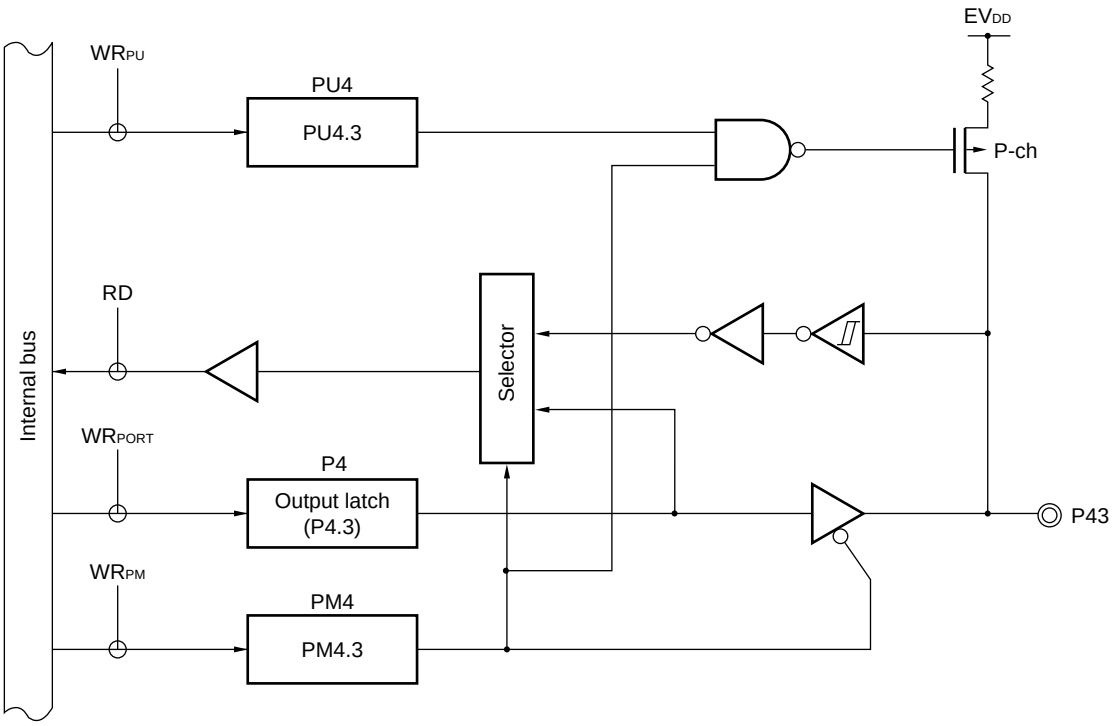
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

Figure 4-19. Block Diagram of P41 and P42



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

Figure 4-20. Block Diagram of P43



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

4.2.6 Port 5

Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P55 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

Output from the P50 to P55 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 5 (POM5).

Input to the P55 pin can be specified as normal input buffer/TTL input buffer in 1-bit units using the port input mode register 5 (PIM5).

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

Reset signal generation sets port 5 to input mode.

For settings of the registers when using port 5, refer to Table 4-8.

Table 4-8. Register Settings When Using Port 5

Pins		PM5.x	PIM5.x	POM5.x	Alternate Function Setting When Using Pin as Port	Remarks
Name	I/O					
P50	Input	1	–	×	SDA11 output = 1	
	Output	0		0		CMOS output
		0		1		N-ch O.D. output
P51	Input	1	–	–	SO output = 1, LTxD0 output = 1 Note 1	
	Output	0				
P52	Input	1	–	–	–	
	Output	0			–	
P53	Input	1	–	–	–	
	Output	0			–	
P54	Input	1	–	–	–	
	Output	0			–	
P55	Input	1	0	×	–	CMOS input
		0	1	×		TTL input
	Output	1	×	0	(PLLBUZ1 output = 1 Note 4 $\overline{\text{SCK00}}$ output = 1 Note 5)	CMOS output
		0	×	1		N-ch O.D. output

Important To use the port 5 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

Notes 1. To use LTxD as a serial data output, set a bit in the corresponding port mode register (PMxx) for each port to “1”. And, set the PMX2 register to “0”.

2. To use SOS1 as a serial data output, set a bit in the corresponding port mode register (PMxx) for each port to “1”. And, set the PMX4 register to “0”.

3. To use $\overline{\text{SCKS1}}$ as a serial data output, set a bit in the corresponding port mode register (PMxx) for each port to “1”. And, set the PMX3 register to “0”.

4. If P55 is used as general-purpose port and PIOR4 is set to 1, set clock output select register 0 (CKS0) to “0”, which is the same as their default status setting.

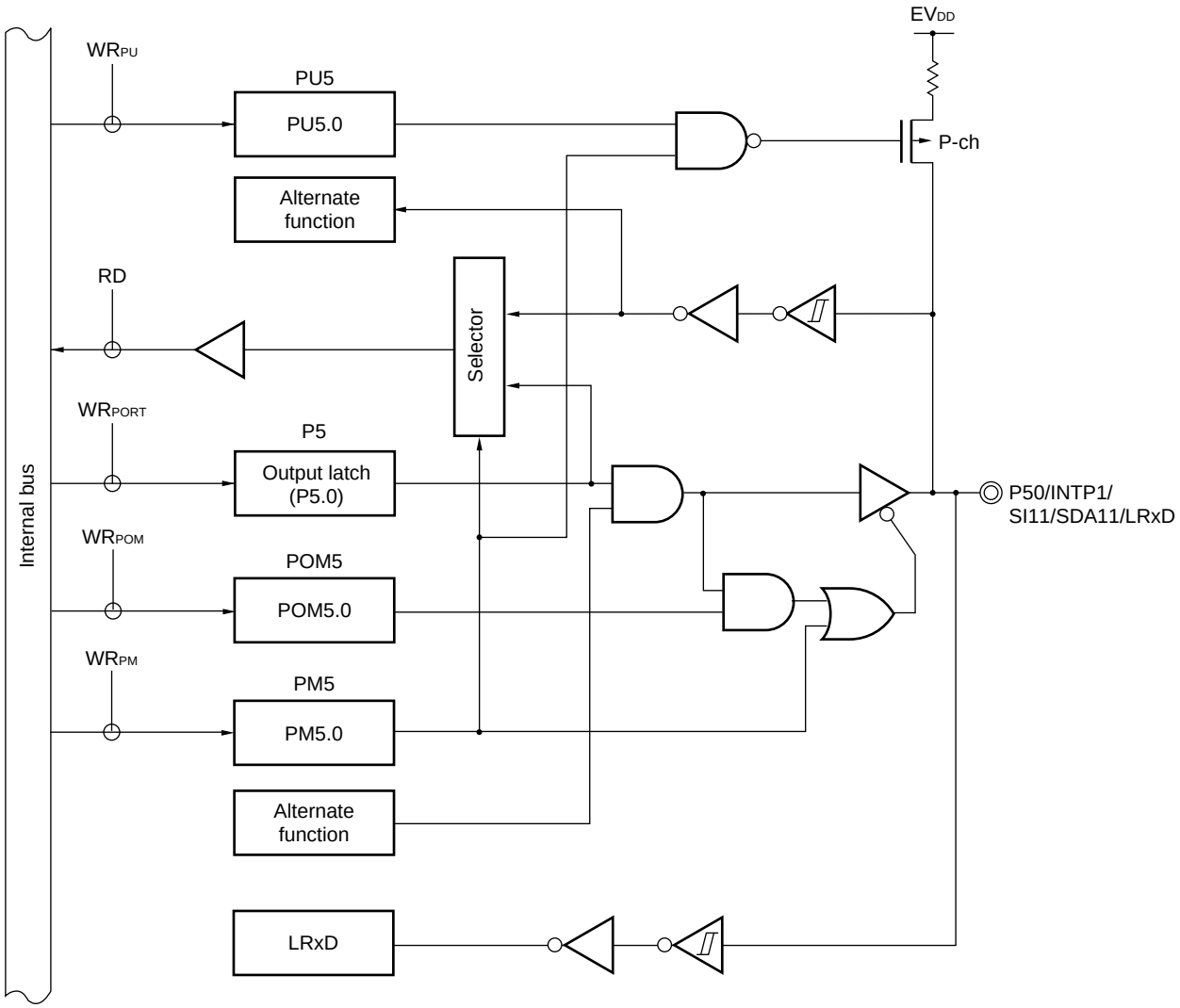
5. If P55 is used as general-purpose port and PIOR1 is set to 1, use serial channel enable status register 0 (SE0), serial output register 0 (SO0) and serial output enable register 0 (SOE0) with the same setting as the initial status.

Caution P50/INTP1/SI11/SDA11/LRxD0, P51/INTP2/SO11/LTxD, P53/SOS1, P55/$\overline{\text{SCKS1}}$ as a general-purpose port, set serial channel enable status register 1 (SE1), serial output register 1 (SO1) and serial output enable register 1 (SOE1) to the default status. Stop the operation of serial interface LIN-UART. Clear port output mode register 5 (POM5) to 00H.

Remark The descriptions in parentheses indicate the case where PIORx = 1.

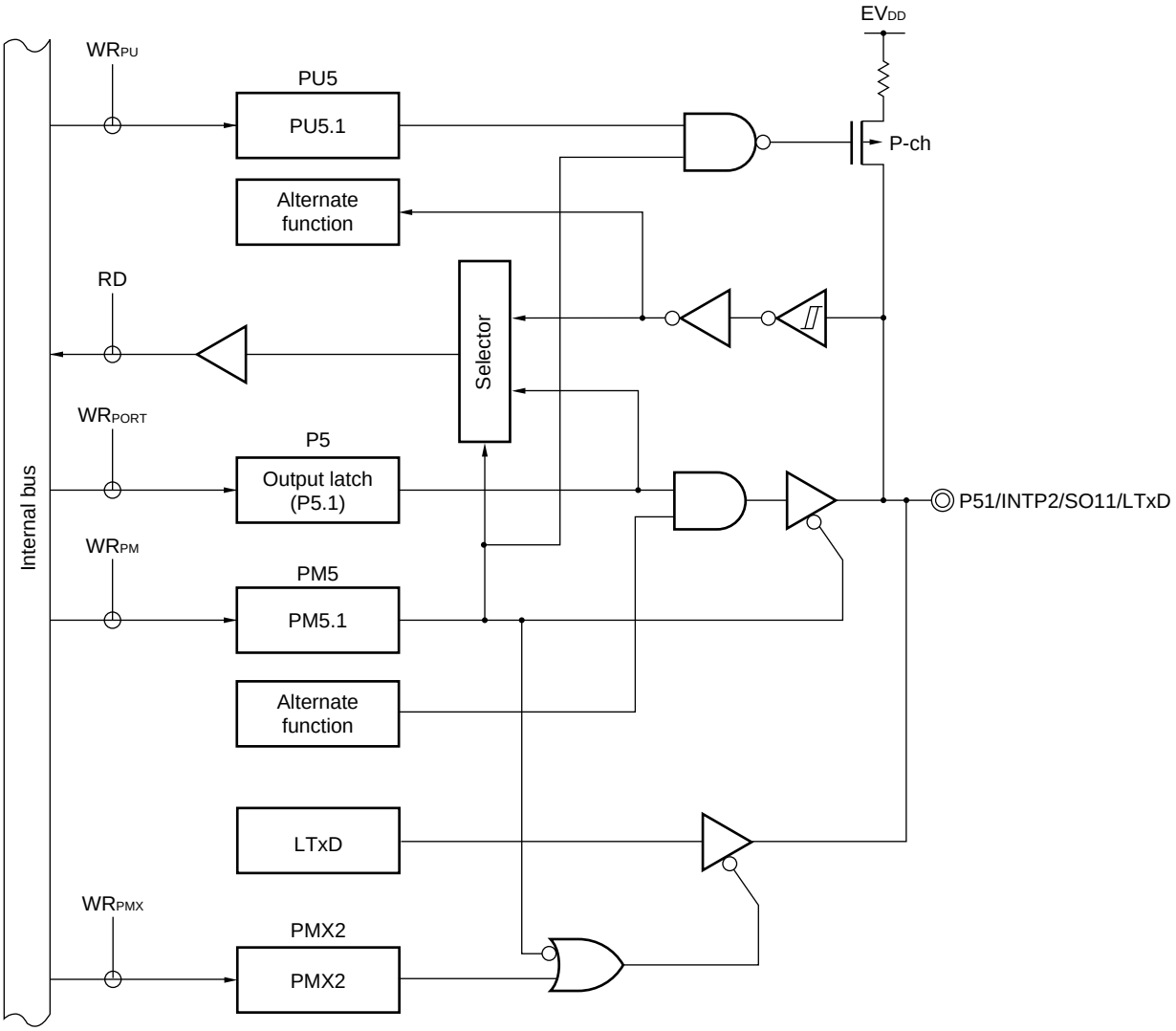
For example, figures 4-21 to 4-26 show block diagrams of port 5 for 64-pin products.

Figure 4-21. Block Diagram of P50



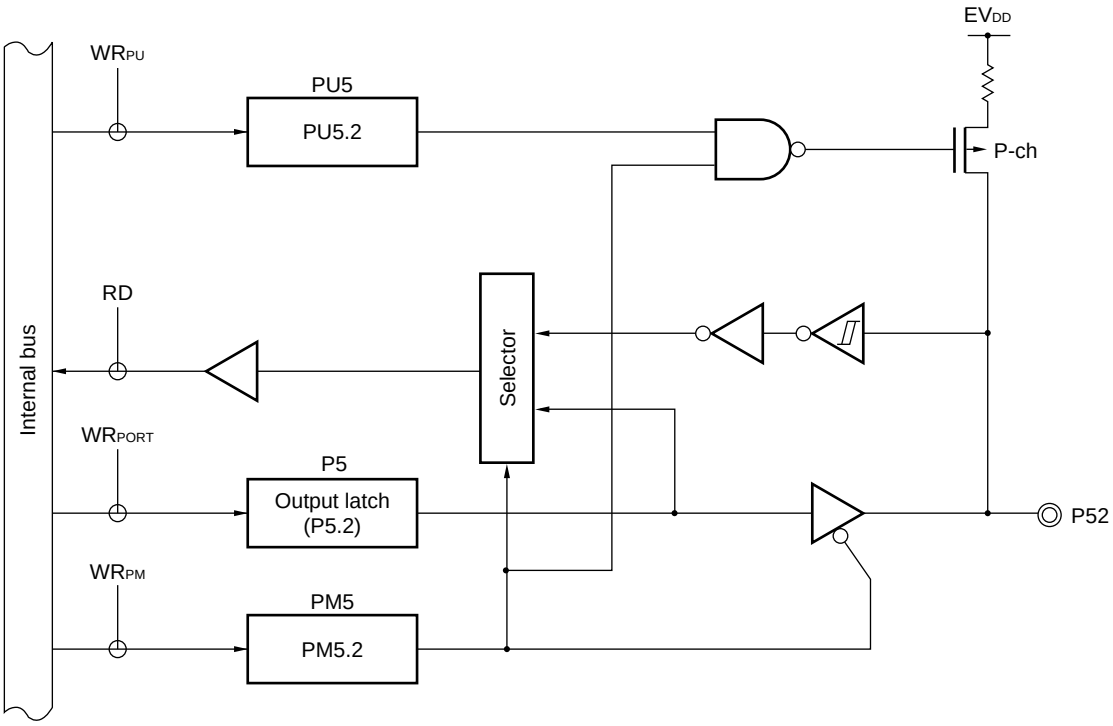
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
POM5: Port output mode register 5
RD: Read signal
WR××: Write signal

Figure 4-22. Block Diagram of P51



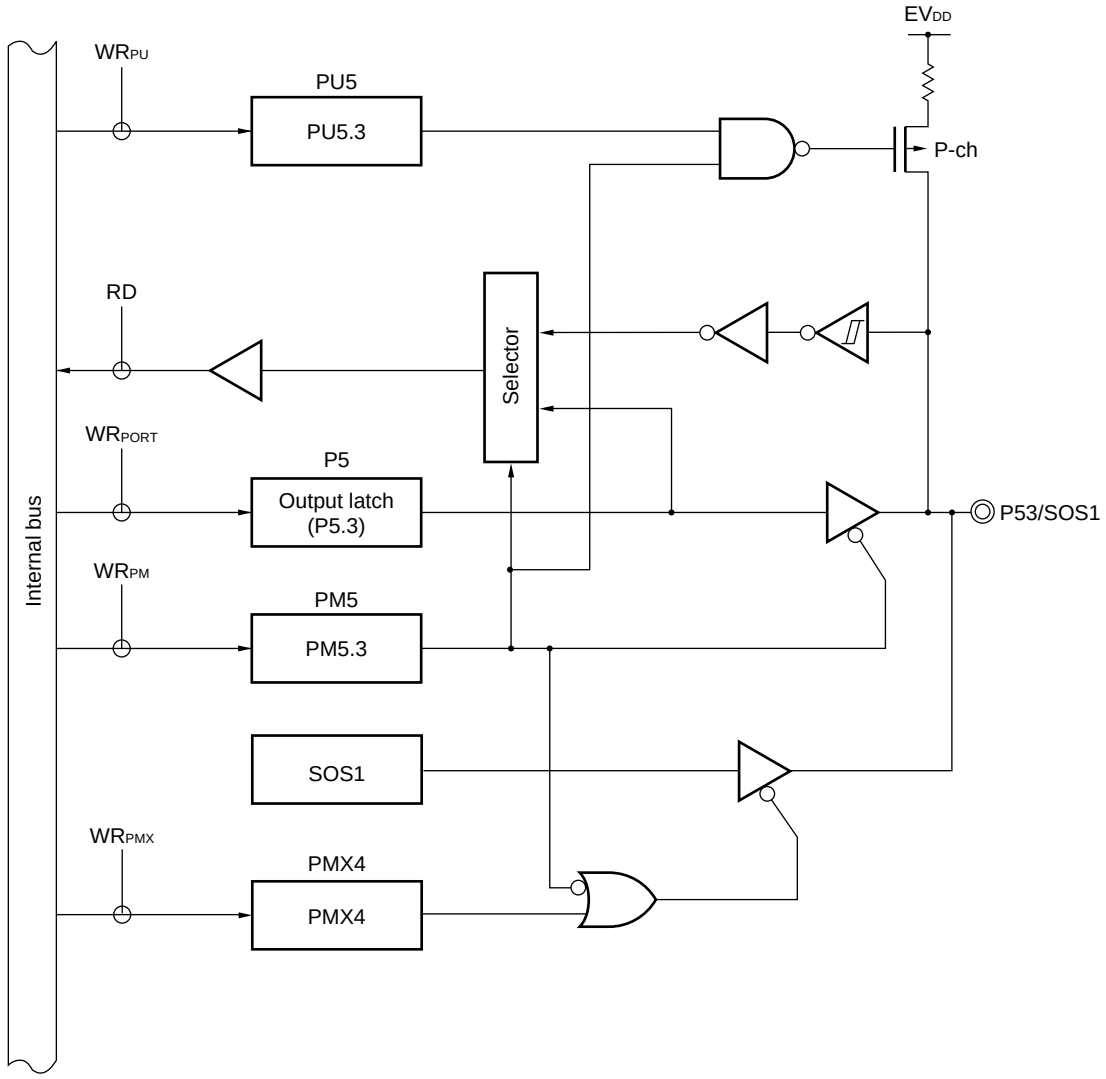
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

Figure 4-23. Block Diagram of P52



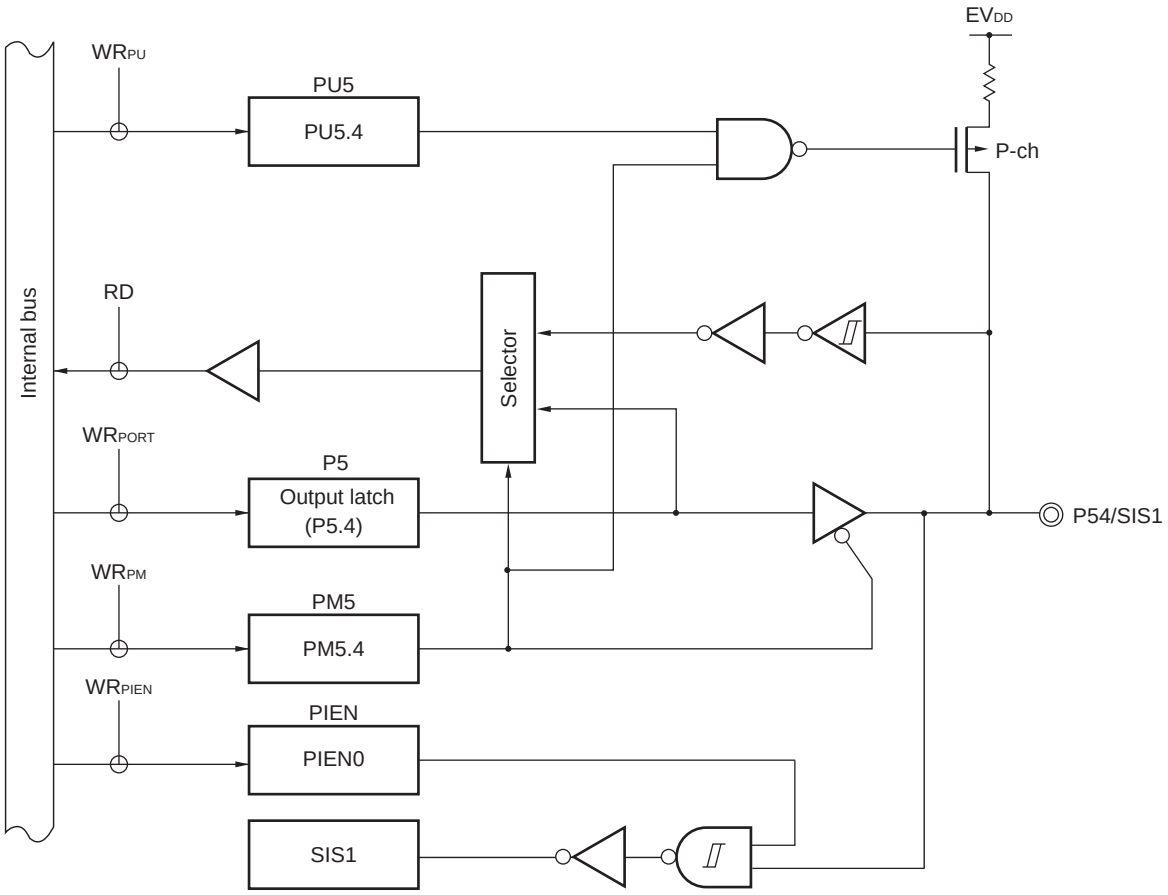
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

Figure 4-24. Block Diagram of P53



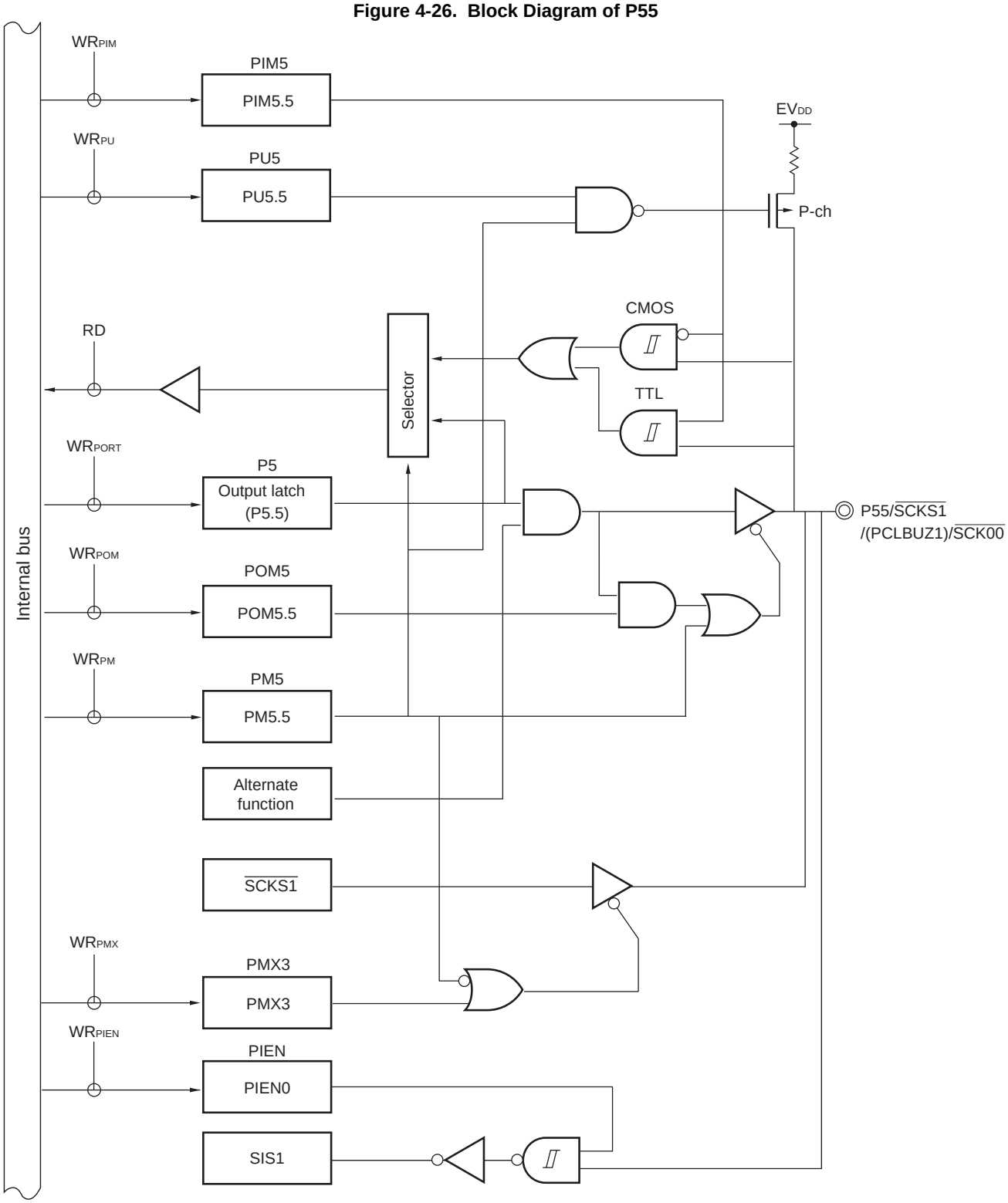
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

Figure 4-25. Block Diagram of P54



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PIEN: Port input enable register
RD: Read signal
WR××: Write signal

Figure 4-26. Block Diagram of P55



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PIEN: Port input enable register
PMX3: Port mode register X3
RD: Read signal
WR××: Write signal

4.2.7 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 mode.

Figures 4-27 and 4-28 show block diagrams of port 6.

For settings of the registers when using port 6, refer to Table 4-9.

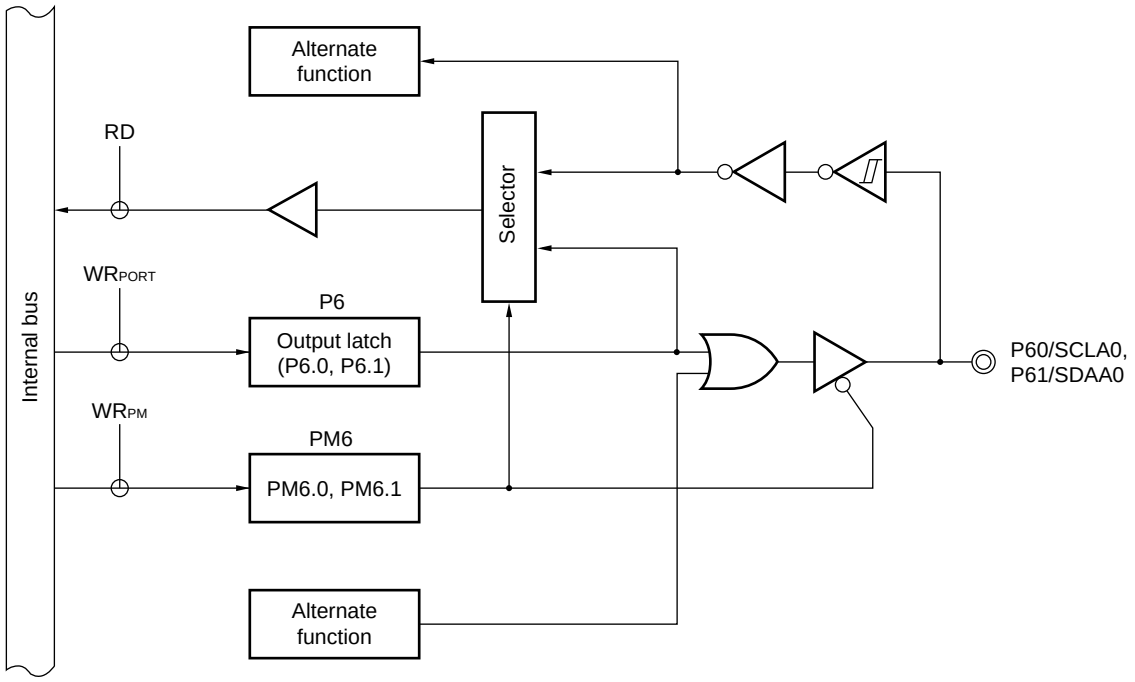
Table 4-9. Register Settings When Using Port 6

Pins		PM6.x	Alternate Function Setting When Using Pin as Port	Remarks
Name	I/O			
P60	Input	1	SCLA0 output = 0	
	Output	0		
P61	Input	1	SDAA0 output = 0	
	Output	0		
P62	Input	1	–	
	Output	0		
P63	Input	1	–	
	Output	0		

Important To use the port 6 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

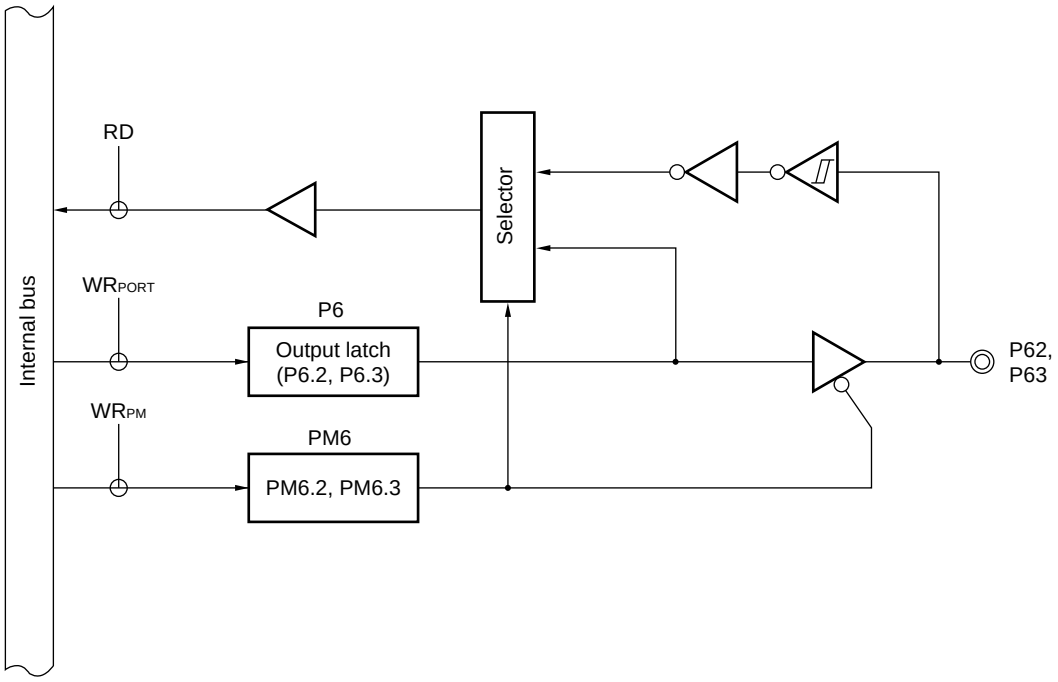
Caution Stop the operation of serial interface IICA when using P60/SCLA0, P61/SDAA0 as general- purpose ports.

Figure 4-27. Block Diagram of P60, P61



P6: Port register 6
PM6: Port mode register 6
RD: Read signal
WR××: Write signal

Figure 4-28. Block Diagram of P62, P63



P6: Port register 6
PM6: Port mode register 6
RD: Read signal
WR××: Write signal

4.2.8 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 and P74 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 7 (POM7).

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

Reset signal generation sets port 7 to input mode.

For settings of the registers when using port 7, refer to Table 4-10.

Table 4-10. Register Settings When Using Port 7

Pins		PM7.x	POM7.x	Alternate Function Setting When Using Pin as Port	Remarks
Name	I/O				
P70	Input	1	–	×	
	Output	0		$\overline{\text{SCK21}}$/SCL21 output = 1	
P71	Input	1	×	×	
	Output	0	0	SDA21 output = 1	CMOS output
		0	1		N-ch O.D. output
P72	Input	1	–	×	
	Output	0		SO21 output = 1	
P73	Input	1	–	×	
	Output	0		SO01 output = 1	
P74	Input	1	×	×	
	Output	0	0	SDA01 output = 1	CMOS output
		0	1		N-ch O.D. output
P75	Input	1	–	×	
	Output	0		$\overline{\text{SCK01}}$/SCL01 output = 1	
P76	Input	1	–	–	
	Output	0			
P77	Input	1	–	×	
	Output	0		(TDX2 output = 1[Note])	

Important To use the port 7 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

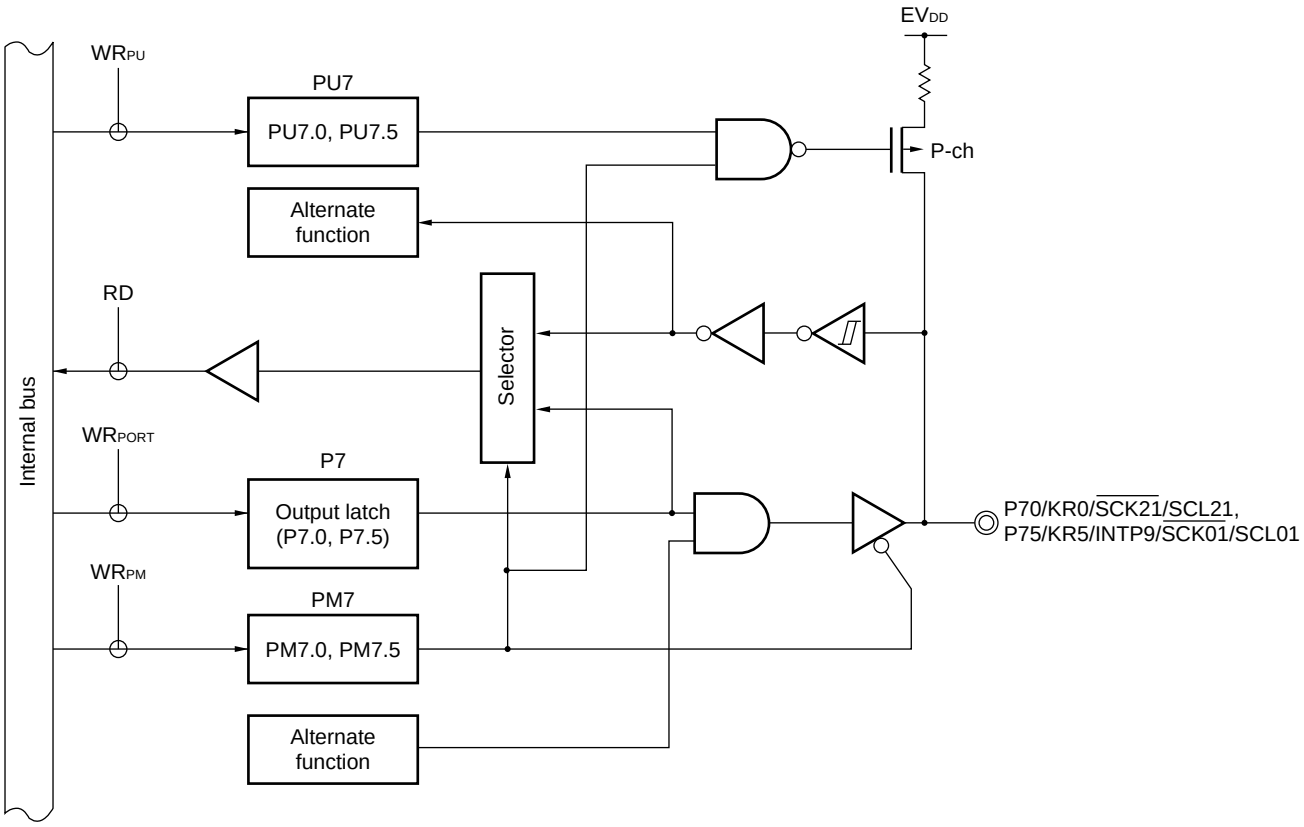
Note If P77 is used as general-purpose port and PIOR1 is set to 1, use serial channel enable status register 1 (SE1), serial output register 1 (SO1) and serial output enable register 1 (SOE1) with the same setting as the initial status.

Caution P70/KR0/$\overline{\text{SCK21}}$/SCL21, P71/KR1/SI21/SDA21 or P72/KR2/SO21, P73/KR3/SO01, P74/KR4/INTP8/SI01/SDA01, P75/INTP9/$\overline{\text{SCK01}}$/SCL01 as a general-purpose port, set serial channel enable status register m (SEm), serial output register m (SOm) and serial output enable register m (SOEm) to the default status (m = 0, 1).

Remark The descriptions in parentheses indicate the case where PIORx = 1.

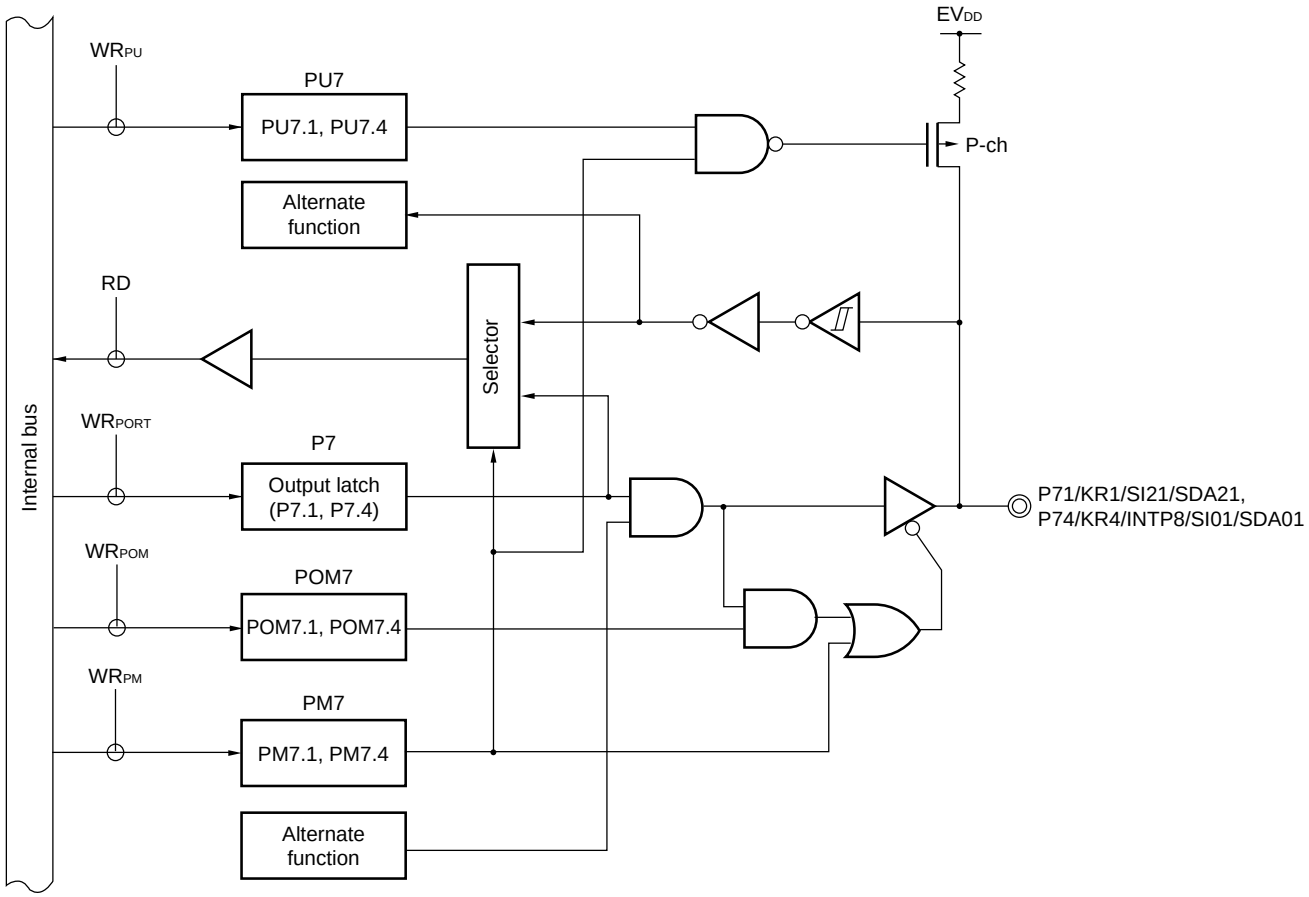
For example, figures 4-29 to 4-32 show block diagrams of port 7 for 64-pin products.

Figure 4-29. Block Diagram of P70, P75



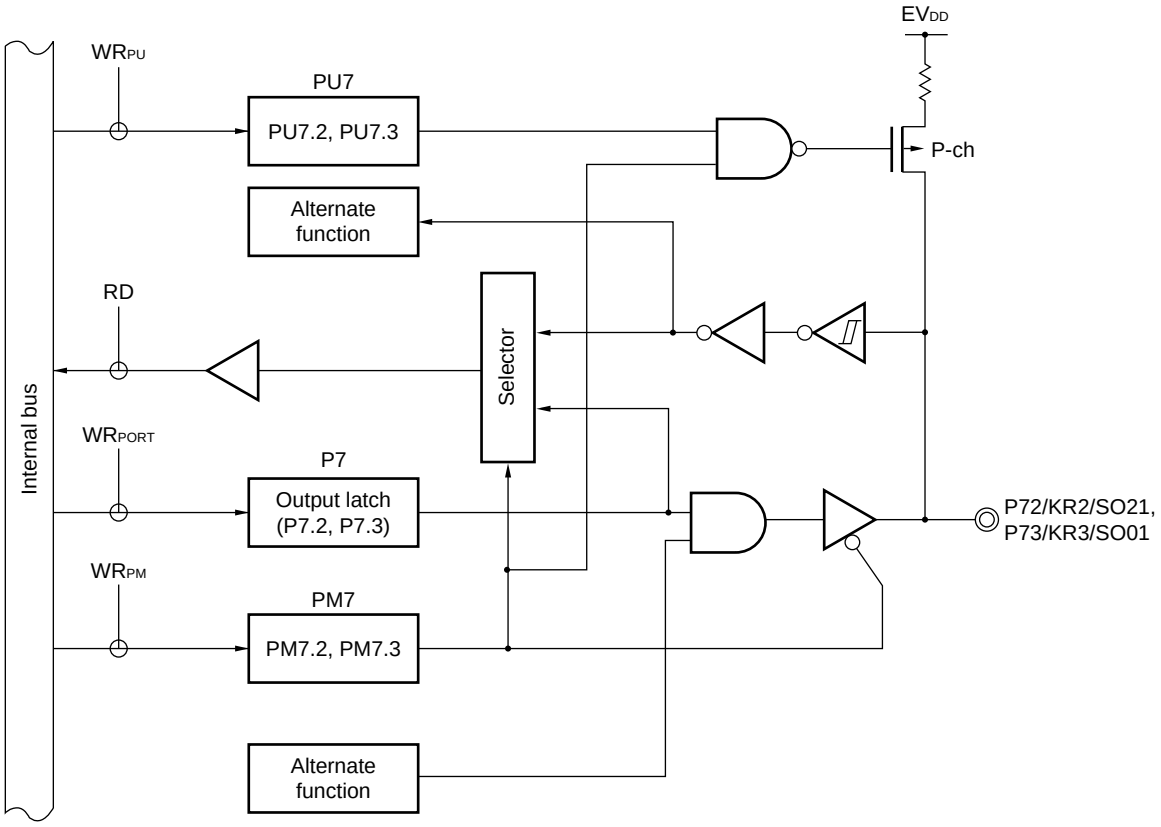
P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

Figure 4-30. Block Diagram of P71, P74



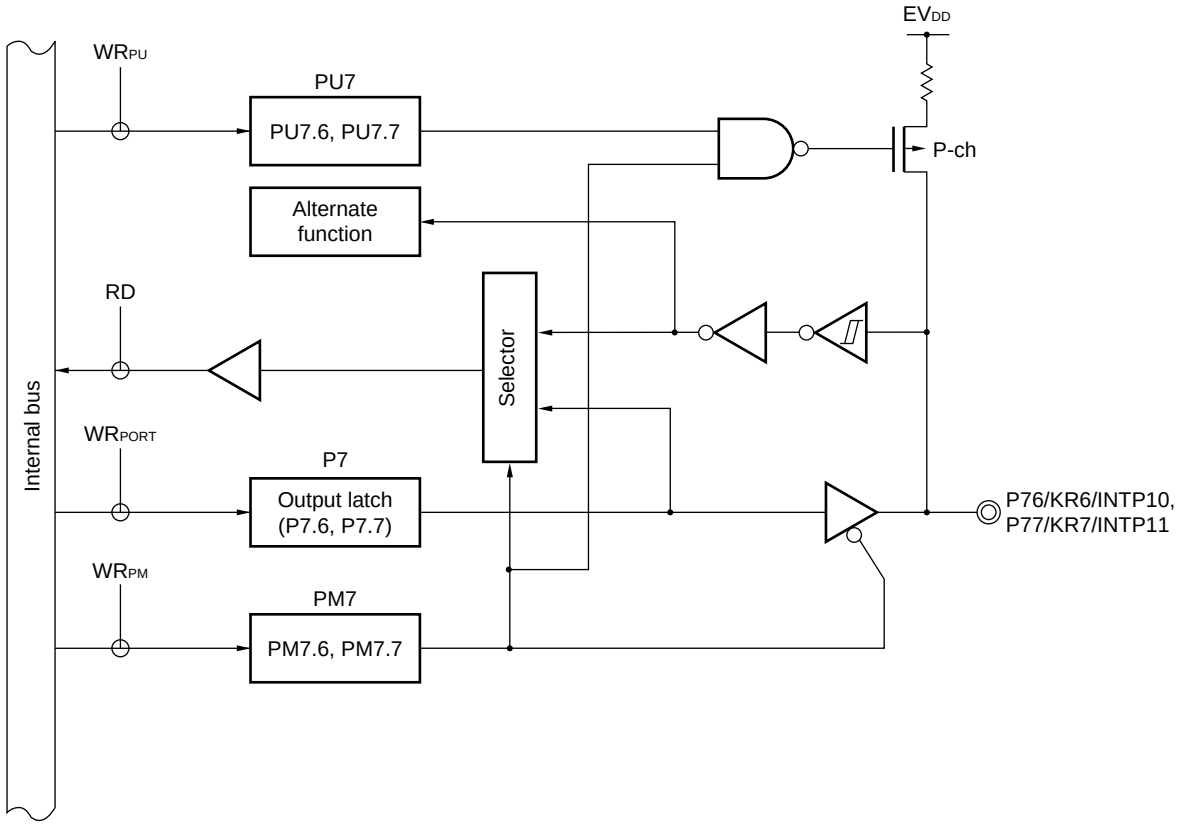
P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
POM7: Port output mode register 7
RD: Read signal
WR××: Write signal

Figure 4-31. Block Diagram of P72, P73



P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

Figure 4-32. Block Diagram of P76, P77



P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

4.2.9 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.

Input to the P120 pin can be specified as analog input or digital input in 1-bit units, using port mode control register 12 (PMC12).

This port can also be used for A/D converter analog input, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, and external clock input for subsystem clock.

Reset signal generation sets P120 to analog input and P121 to P124 to input mode.

Figures 4-33 to 4-35 show block diagrams of port 12.

For settings of the registers when using port 12, refer to Table 4-11.

Table 4-11. Register Settings When Using Port 12

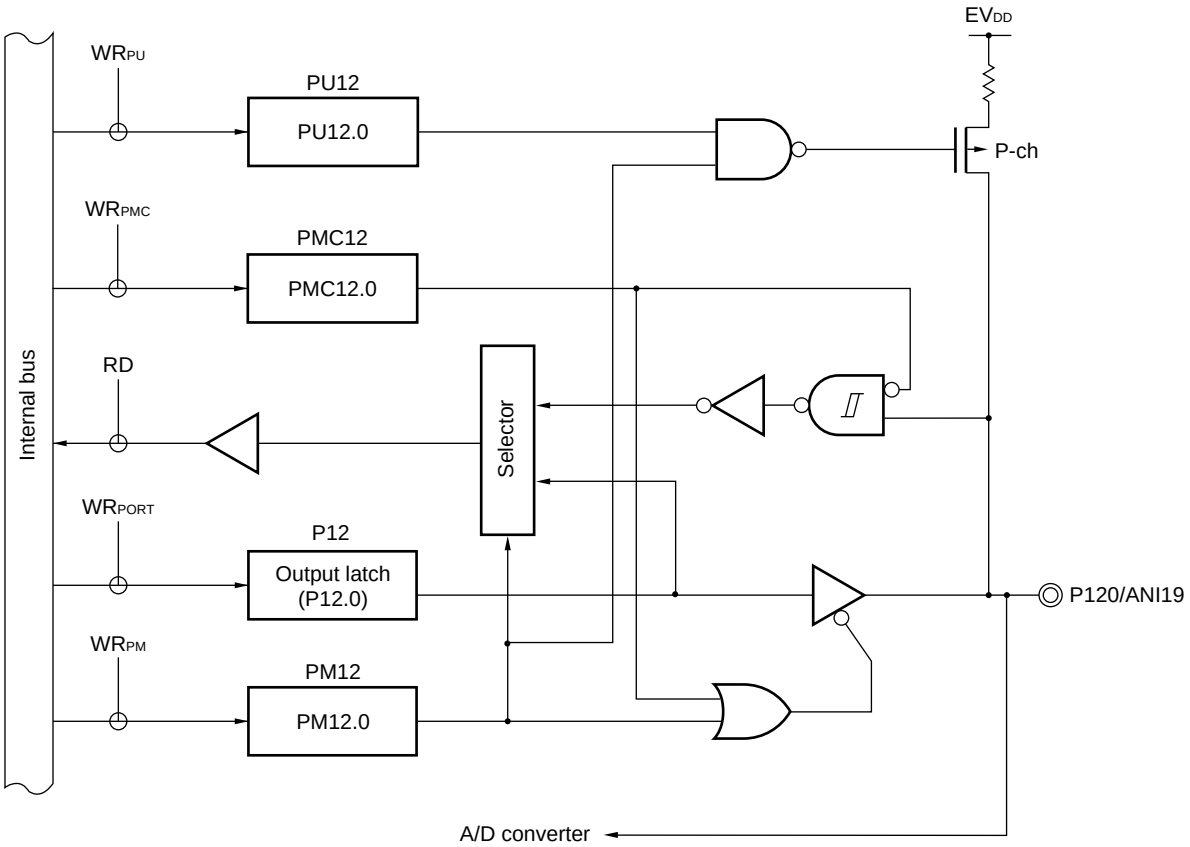
Pins		PM12.x	PMC12.x	Alternate Function Setting When Using Pin as Port	Remarks
Name	I/O				
P120	Input	1	0	×	
	Output	0	0	×	
P121	Input	–	–	Bits OSCSEL = 0 or EXCLK = 1 of the CMC register	
P122	Input	–	–	The OSCSEL bit of the CMC register = 0	
P123	Input	–	–	Bits OSCSELS = 0 or EXCLKS = 1 of the CMC register	
P124	Input	–	–	The OSCSELS bit of the CMC register = 0	

Important To use the port 12 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

Caution The function setting on P121 to P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.

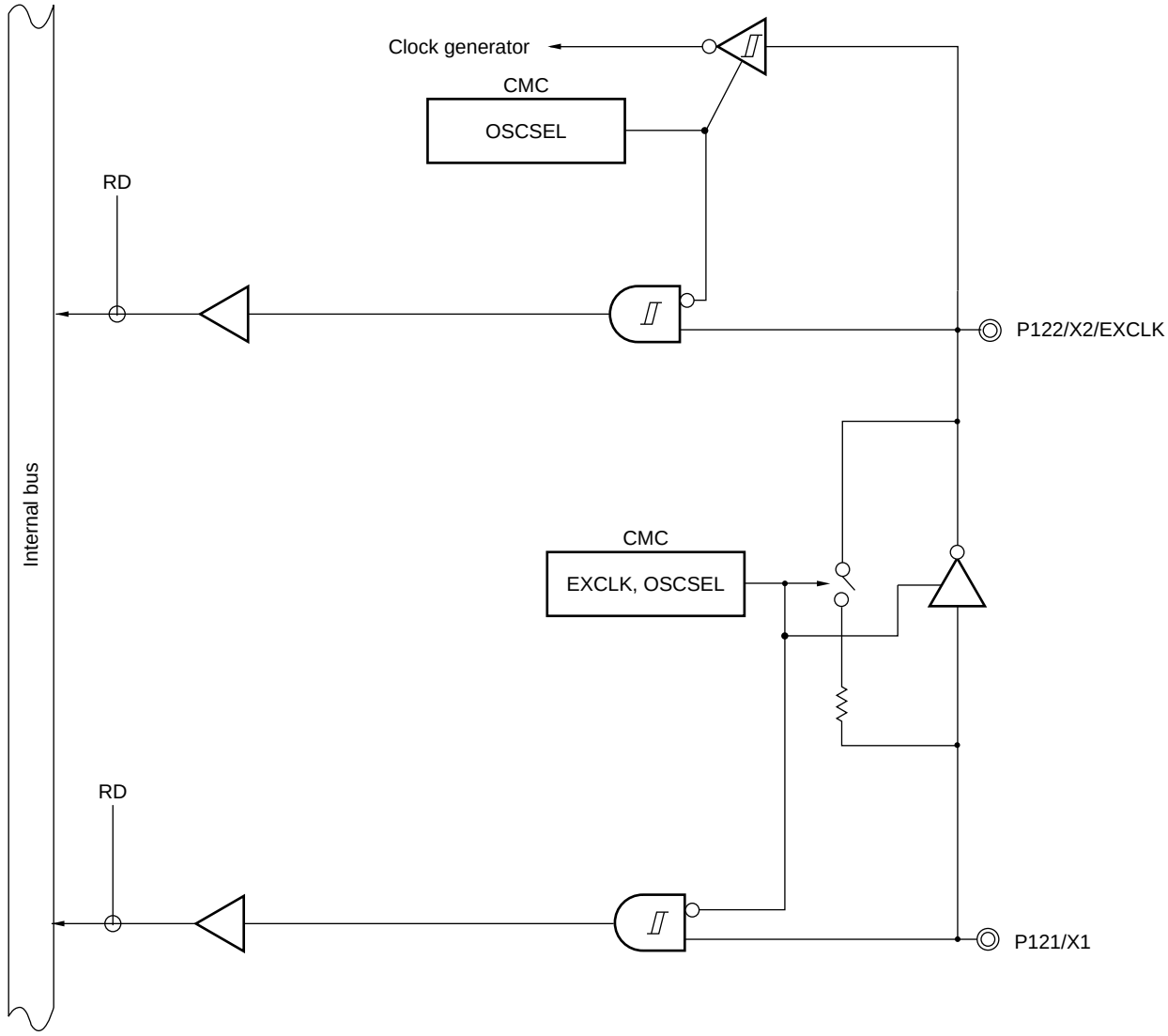
For example, figures 4-33 to 4-35 show block diagrams of port 12 for 64-pin products.

Figure 4-33. Block Diagram of P120



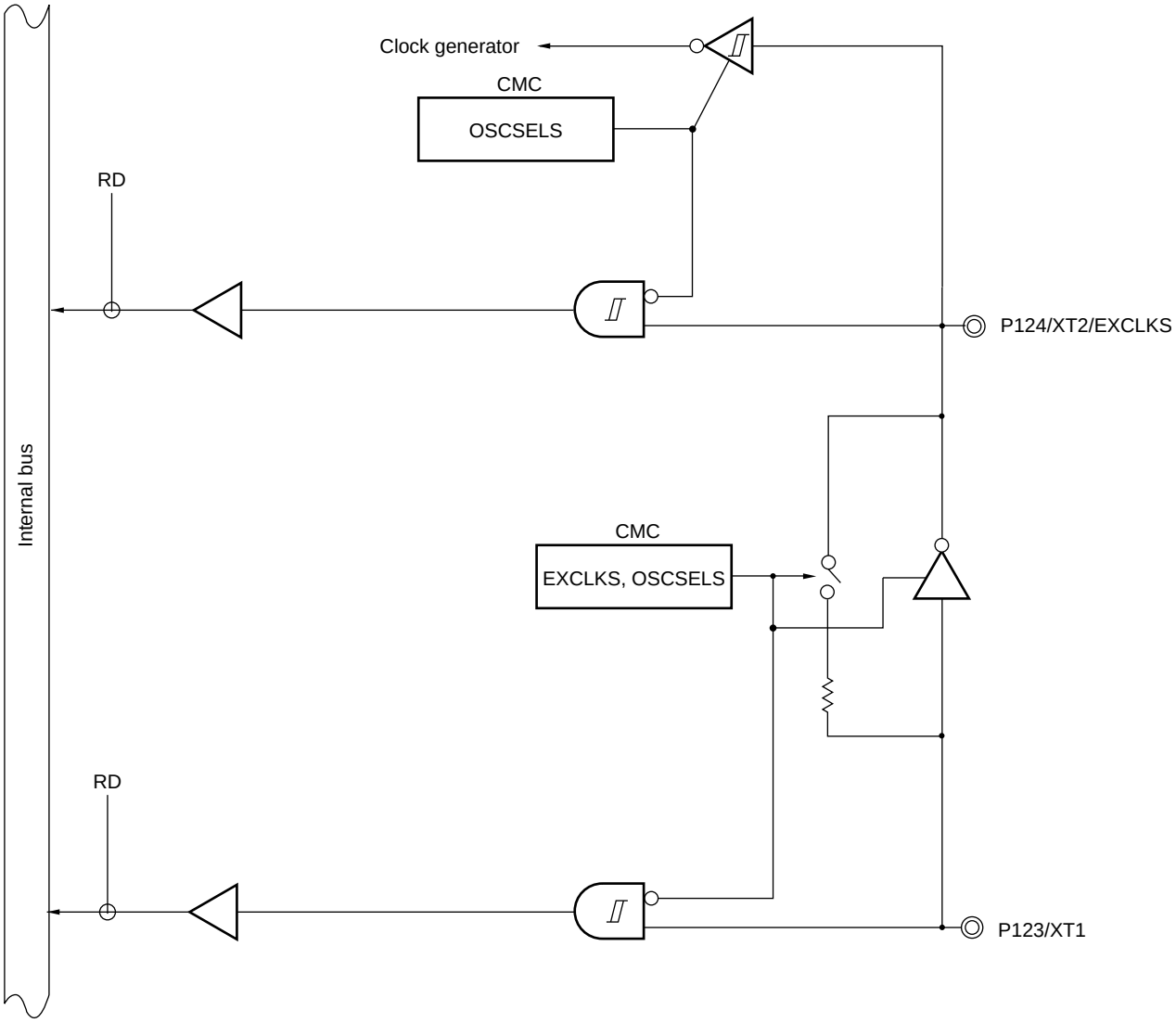
P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
PMC12: Port mode control register 12
RD: Read signal
WR××: Write signal

Figure 4-34. Block Diagram of P121 and P122



CMC: Clock operation mode control register
RD: Read signal

Figure 4-35. Block Diagram of P123 and P124



CMC: Clock operation mode control register
RD: Read signal

4.2.10 Port 13

P130 is a 1-bit output-only port with an output latch.

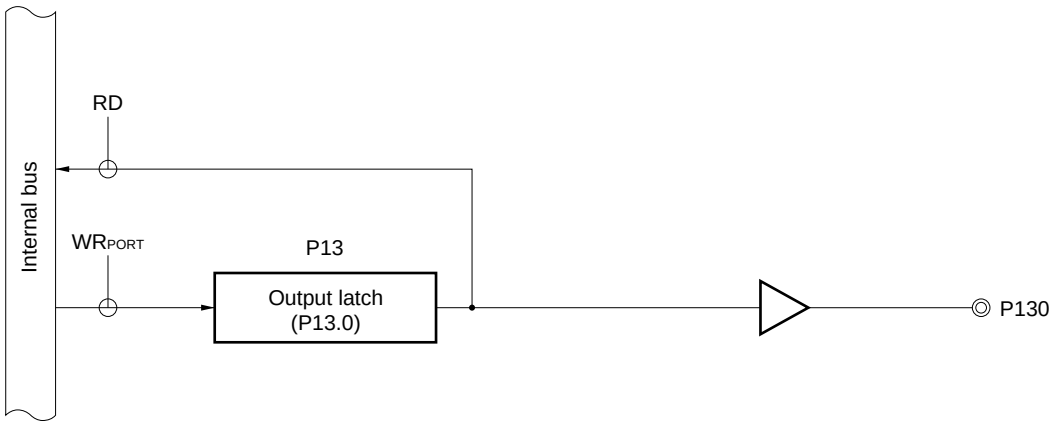
P137 is a 1-bit input-only port.

When the reset signal is generated, P130 is fixed to output mode and P137 to input mode.

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

Figures 4-36 and 4-37 show block diagrams of port 13.

Figure 4-36. Block Diagram of P130



P13: Port register 13

RD: Read signal

WR××: Write signal

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.

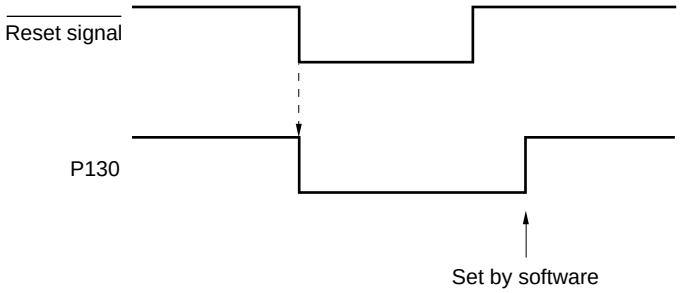
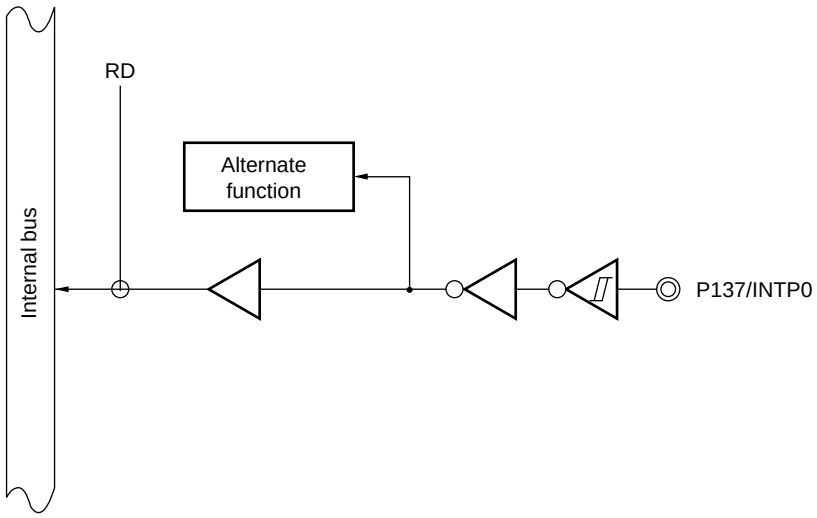


Figure 4-37. Block Diagram of P137



RD: Read signal

4.2.11 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, P141, P146, P147 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 P147 pin can be specified as analog input or digital input in 1-bit units, using port mode control register 14 (PMC14).

This port can also be used for clock/buzzer output, external interrupt request input, and A/D converter analog input.

Reset signal generation sets P140, P141, and P146 to input mode and P147 to analog input mode.

For settings of the registers when using port 14, refer to Table 4-12.

Table 4-12. Register Settings When Using Port 14

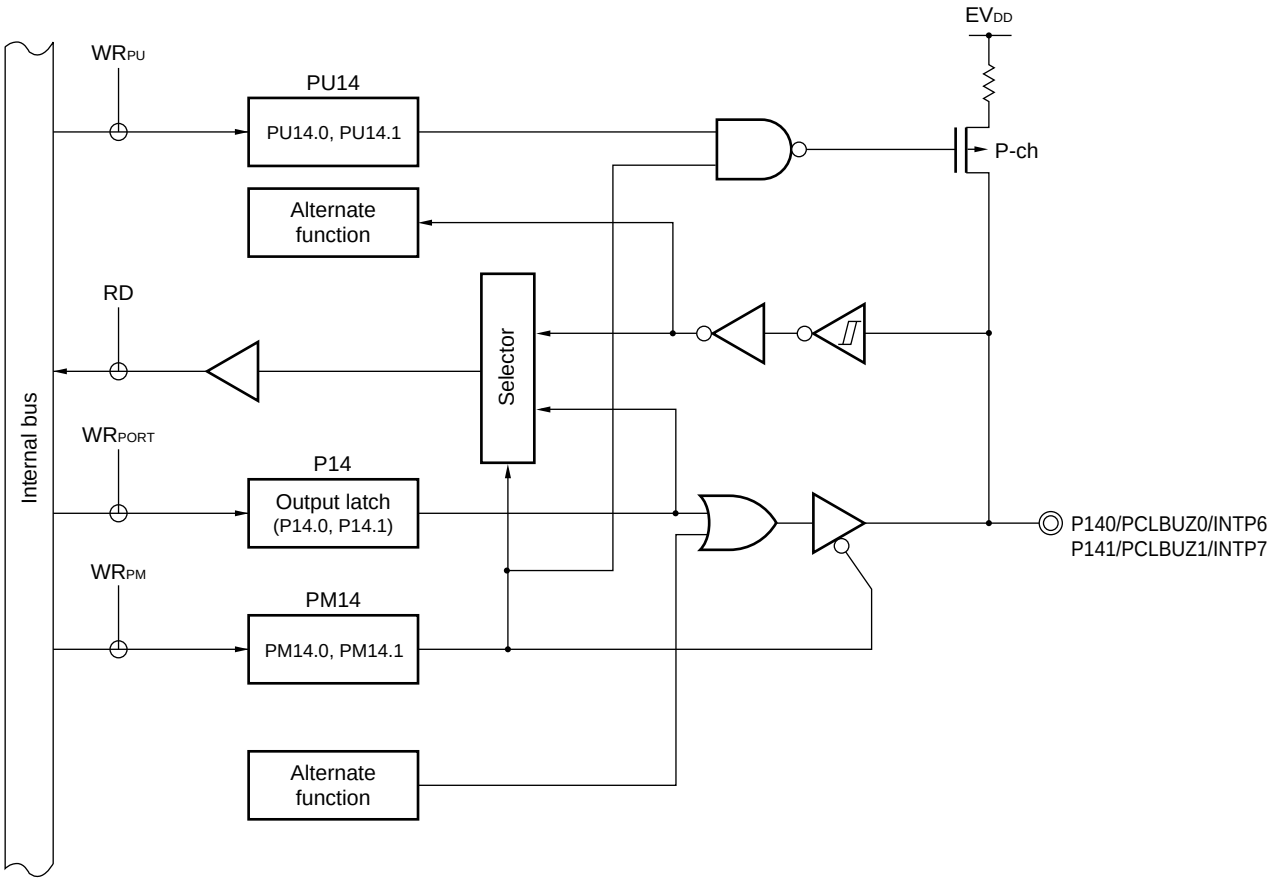
Pins		PM14.x	PMC14.x	Alternate Function Setting When Using Pin as Port	Remarks
Name	I/O				
P140	Input	1	–	×	
	Output	0		PCLBUZ0 output = 0	
P141	Input	1	–	×	
	Output	0		PCLBUZ1 output = 0	
P146	Input	1	–	×	
	Output	0		×	
P147	Input	1	0	×	
	Output	0	0	×	

Important To use the port 14 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

Caution To use P140/PCLBUZ0/INTP6, P141/PCLBUZ1/INTP7 as a general-purpose port, set bit 7 of clock output select register 0, 1 (CKS0, CKS1) to "0", which is the same as their default status settings.

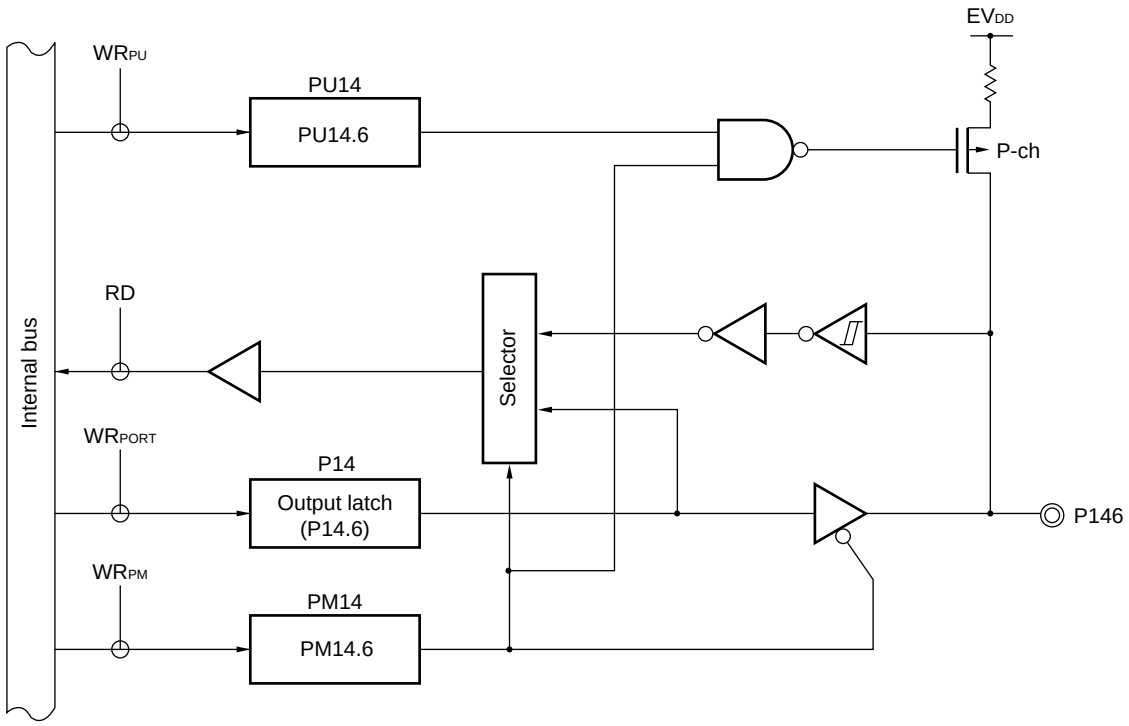
For example, figures 4-38 to 4-40 show block diagrams of port 14 for 64-pin products.

Figure 4-38. Block Diagram of P140 and P141



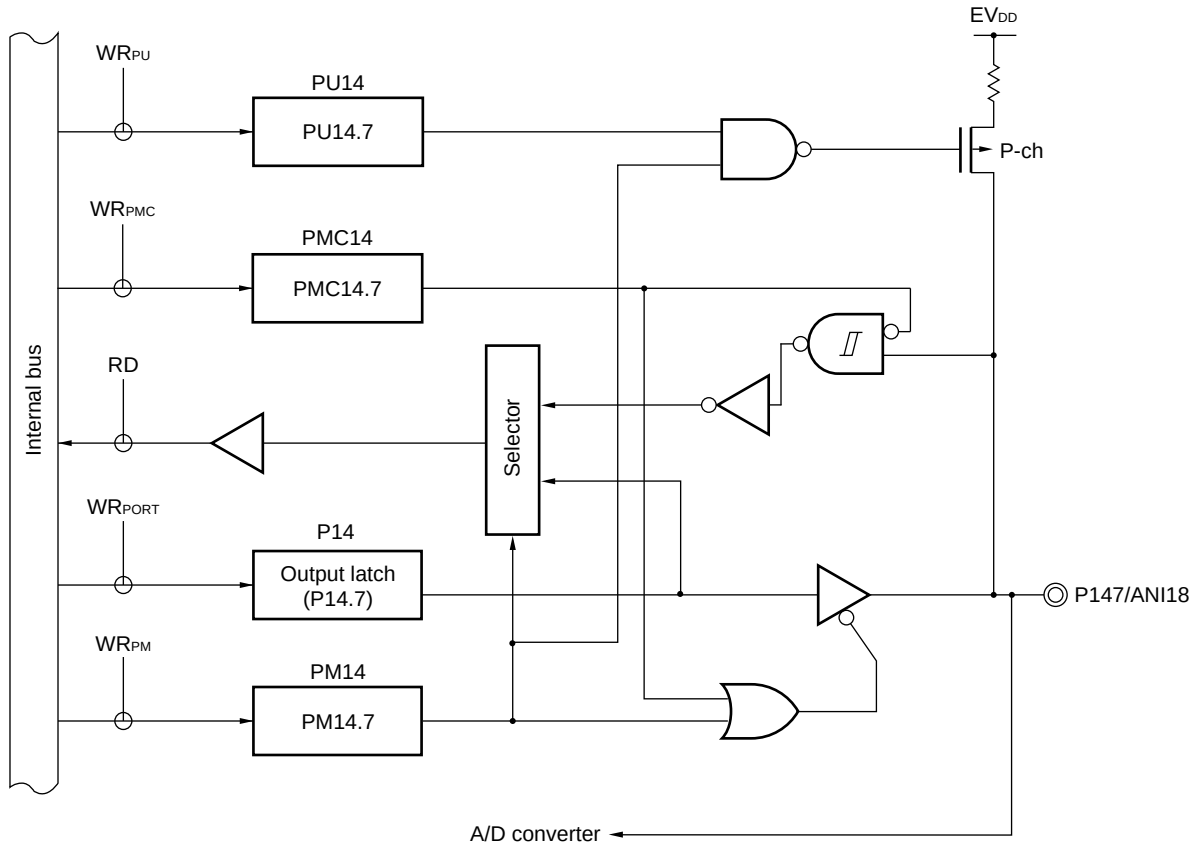
P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
RD: Read signal
WR××: Write signal

Figure 4-39. Block Diagram of P146



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
RD: Read signal
WR××: Write signal

Figure 4-40. Block Diagram of P147



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
PMC14: Port mode control register 14
RD: Read signal
WR××: Write signal

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control registers (PMCxx)
- A/D port configuration register (ADPC)
- Port mode registers X (PMXx)
- Peripheral I/O redirection register (PIOR)
- Port input enable register (PIEN)

Table 4-13. PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (1/2)

Port		Bit name							64 pin	48 pin	32 pin	30 pin	20 pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register	PMXx register					
Port 0	0	PM0.0	P0.0	PU0.0	–	POM0.0	PMC0.0 Note	–	√	√	√	√	–
	1	PM0.1	P0.1	PU0.1	PIM0.1	–	PMC0.1 Note	–	√	√	√	√	√
	2	PM0.2	P0.2	PU0.2	–	POM0.2	PMC0.2	–	√	–	–	–	–
	3	PM0.3	P0.3	PU0.3	PIM0.3	POM0.3	PMC0.3	–	√	–	–	–	–
	4	PM0.4	P0.4	PU0.4	PIM0.4	POM0.4	–	–	√	–	–	–	–
	5	PM0.5	P0.5	PU0.5	–	–	–	–	√	–	–	–	–
	6	PM0.6	P0.6	PU0.6	–	–	–	–	√	–	–	–	–
Port 1	0	PM1.0	P1.0	PU1.0	–	POM1.0	–	PMX0	√	√	√	√	√
	1	PM1.1	P1.1	PU1.1	–	POM1.1	–	–	√	√	√	√	√
	2	PM1.2	P1.2	PU1.2	–	POM1.2	–	PMX1	√	√	√	√	√
	3	PM1.3	P1.3	PU1.3	PIM1.3	POM1.3	–	–	√	√	√	√	–
	4	PM1.4	P1.4	PU1.4	PIM1.4	POM1.4	–	–	√	√	√	√	–
	5	PM1.5	P1.5	PU1.5	PIM1.5	POM1.5	–	–	√	√	√	√	–
	6	PM1.6	P1.6	PU1.6	PIM1.6	–	–	–	√	√	√	√	√
	7	PM1.7	P1.7	PU1.7	PIM1.7	POM1.7	–	–	√	√	√	√	√
Port 2	0	PM2.0	P2.0	–	–	–	–	–	√	√	√	√	√
	1	PM2.1	P2.1	–	–	–	–	–	√	√	√	√	√
	2	PM2.2	P2.2	–	–	–	–	–	√	√	√	√	√
	3	PM2.3	P2.3	–	–	–	–	–	√	√	√	√	–
	4	PM2.4	P2.4	–	–	–	–	–	√	√	–	–	–
	5	PM2.5	P2.5	–	–	–	–	–	√	√	–	–	–
	6	PM2.6	P2.6	–	–	–	–	–	√	√	–	–	–
	7	PM2.7	P2.7	–	–	–	–	–	√	√	–	–	–
Port 3	0	PM3.0	P3.0	PU3.0	–	–	–	–	√	√	√	√	–
	1	PM3.1	P3.1	PU3.1	–	–	–	–	√	√	√	√	√
Port 4	0	PM4.0	P4.0	PU4.0	–	–	–	–	√	√	√	√	√
	1	PM4.1	P4.1	PU4.1	–	–	–	–	√	√	–	–	–
	2	PM4.2	P4.2	PU4.2	–	–	–	–	√	–	–	–	–
	3	PM4.3	P4.3	PU4.3	–	–	–	–	√	–	–	–	–
Port 5	0	PM5.0	P5.0	PU5.0	–	POM5.0	–	–	√	√	√	√	√
	1	PM5.1	P5.1	PU5.1	–	–	–	PMX2	√	√	√	√	√
	2	PM5.2	P5.2	PU5.2	–	–	–	–	√	–	–	–	–
	3	PM5.3	P5.3	PU5.3	–	–	–	PMX4	√	–	–	–	–
	4	PM5.4	P5.4	PU5.4	–	–	–	–	√	–	–	–	–
	5	PM5.5	P5.5	PU5.5	PIM5.5	POM5.5	–	PMX3	√	–	–	–	–
Port 6	0	PM6.0	P6.0	–	–	–	–	–	√	√	√	√	–
	1	PM6.1	P6.1	–	–	–	–	–	√	√	√	√	–
	2	PM6.2	P6.2	–	–	–	–	–	√	√	√	–	–
	3	PM6.3	P6.3	–	–	–	–	–	√	√	–	–	–

Note 20-, 30-, and 32-pin products only.

Table 4-13. PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (2/2)

Port		Bit name							64 pin	48 pin	32 pin	30 pin	20 pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register	PMXx register					
Port 7	0	PM7.0	P7.0	PU7.0	–	–	–	–	√	√	√	–	–
	1	PM7.1	P7.1	PU7.1	–	POM7.1	–	–	√	√	–	–	–
	2	PM7.2	P7.2	PU7.2	–	–	–	–	√	√	–	–	–
	3	PM7.3	P7.3	PU7.3	–	–	–	–	√	√	–	–	–
	4	PM7.4	P7.4	PU7.4	–	POM7.4	–	–	√	√	–	–	–
	5	PM7.5	P7.5	PU7.5	–	–	–	–	√	√	–	–	–
	6	PM7.6	P7.6	PU7.6	–	–	–	–	√	–	–	–	–
	7	PM7.7	P7.7	PU7.7	–	–	–	–	√	–	–	–	–
Port 12	0	PM12.0	P12.0	PU12.0	–	–	PMC12.0	–	√	√	√	√	–
	1	–	PM12.1	–	–	–	–	–	√	√	√	√	√
	2	–	PM12.2	–	–	–	–	–	√	√	√	√	√
	3	–	PM12.3	–	–	–	–	–	√	√	–	–	–
	4	–	PM12.4	–	–	–	–	–	√	√	–	–	–
Port 13	0	–	P13.0	–	–	–	–	–	√	√	–	–	–
	7	–	P13.7	–	–	–	–	–	√	√	√	√	√
Port 14	0	PM14.0	P14.0	PU14.0	–	–	–	–	√	√	–	–	–
	1	PM14.1	P14.1	PU14.1	–	–	–	–	√	–	–	–	–
	6	PM14.6	P14.6	PU14.6	–	–	–	–	√	√	–	–	–
	7	PM14.7	P14.7	PU14.7	–	–	PMC14.7	–	√	√	√	√	–

Note 20-, 30-, and 32-pin products only.

The format of each register is described below. The description here uses the 64-pin products as an example.

(1) Port mode registers (PMxx)

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

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

Reset signal generation sets these registers to FFH.

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

Figure 4-41. Format of Port Mode Register (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM0.6	PM0.5	PM0.4	PM0.3	PM0.2	PM0.1	PM0.0	FFF20H	FFH	R/W
PM1	PM1.7	PM1.6	PM1.5	PM1.4	PM1.3	PM1.2	PM1.1	PM1.0	FFF21H	FFH	R/W
PM2	PM2.7	PM2.6	PM2.5	PM2.4	PM2.3	PM2.2	PM2.1	PM2.0	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM3.1	PM3.0	FFF23H	FFH	R/W
PM4	1	1	1	1	PM4.3	PM4.2	PM4.1	PM4.0	FFF24H	FFH	R/W
PM5	1	1	PM5.5	PM5.4	PM5.3	PM5.2	PM5.1	PM5.0	FFF25H	FFH	R/W
PM6	1	1	1	1	PM6.3	PM6.2	PM6.1	PM6.0	FFF26H	FFH	R/W
PM7	PM7.7	PM7.6	PM7.5	PM7.4	PM7.3	PM7.2	PM7.1	PM7.0	FFF27H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM12.0	FFF2CH	FFH	R/W
PM14	PM14.7	PM14.6	1	1	1	1	PM14.1	PM14.0	FFF2EH	FFH	R/W

PMm.n	Pmn pin I/O mode selection (m = 0 to 7, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Cautions 1. Be sure to set bits 7 of the PM0 registers, bits 2 to7 of the PM3 registers, bits 4 to 7 of the PM4 register, bits 6, 7 of the PM5 register, bits 4 to 7 of the PM6 registers, bits 1 to 7 of the PM12 register, and bits 2 to 5 of the PM14 register to "1".

2. In 20-pin products, complete the following software processing for the following each port before performing the operation that reads the port latch Pm having the target port latch Pm.n within 50 ms after releasing reset (after starting CPU operation).

- **Set P00, P13, P14, P15, P30, P60, P61, P120, and P147 to low level output mode by the software (clear the PMm.n and Pm.n bits for the target ports).**
- **Set P23 to digital port and low level output mode by the software (set P23 to digital mode with the ADPC register and clear the PM2.3 and P2.3 bits).**

(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 If P00 to P03, P20 to P27, P120, and P147 are set up as analog inputs of the A/D converter, when a port is read while in the input mode, 0 is always returned, not the pin level.

Figure 4-42. Format of Port Register (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	FFF00H	00H (output latch)	R/W
P1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	FFF01H	00H (output latch)	R/W
P2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	0	0	P3.1	P3.0	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	P4.3	P4.2	P4.1	P4.0	FFF04H	00H (output latch)	R/W
P5	0	0	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	P6.3	P6.2	P6.1	P6.0	FFF06H	00H (output latch)	R/W
P7	P7.7	P7.6	P7.5	P7.4	P7.3	P7.2	P7.1	P7.0	FFF07H	00H (output latch)	R/W
P12	0	0	0	P12.4	P12.3	P12.2	P12.1	P12.0	FFF0CH	Undefined	R/W[Note]
P13	P13.7	0	0	0	0	0	0	P13.0	FFF0DH	Undefined	R/W[Note]
P14	P14.7	P14.6	0	0	0	0	P14.1	P14.0	FFF0EH	00H (output latch)	R/W

Pm.n	m = 0 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

Notes 1. P121 to P124, and P137 are read-only.

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

Caution For the following each port, complete the following software processing before performing the operation that reads the port latch Pm having the target port latch Pm.n within 50 ms after releasing reset (after staring CPU operation)

- **Set P00, P13, P14, P15, P30, P60, P61, P120, and P147 to low level output mode by the software (clear the PMm.n and Pm.n bits for the target ports).**

- **Set P23 to digital port and low level output mode by the software (set P23 to digital mode with the ADPC register and clear the PM2.3 and P2.3 bits).**

(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) by setting POMmn = 0 of 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, bits used as alternate-function output pins, and bits with analog setting (PMC = 1, ADPC =1), regardless of the settings of these registers.

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).

Caution When a port with the PIMn register is input from different potential device to TTL buffer, pull up to the power supply of the different potential device via a external pull-up resistor by setting PUmn = 0.

Figure 4-43. Format of Pull-up Resistor Option Register (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	PU0.6	PU0.5	PU0.4	PU0.3	PU0.2	PU0.1	PU0.0	F0030H	00H	R/W
PU1	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU3.1	PU3.0	F0033H	00H	R/W
PU4	0	0	0	0	PU4.3	PU4.2	PU4.1	PU4.0	F0034H	01H	R/W
PU5	0	0	PU5.5	PU5.4	PU5.3	PU5.2	PU5.1	PU5.0	F0035H	00H	R/W
PU7	PU7.7	PU7.6	PU7.5	PU7.4	PU7.3	PU7.2	PU7.1	PU7.0	F0037H	00H	R/W
PU12	0	0	0	0	0	0	0	PU12.0	F003CH	00H	R/W
PU14	PU14.7	PU14.6	0	0	0	0	PU14.1	PU14.0	F003EH	00H	R/W

PUm.n	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 5, 7, 12, 14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

(4) Port input mode registers (PIM0, PIM1, PM5)

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

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

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

Reset signal generation clears these registers to 00H.

Figure 4-44. Format of Port Input Mode Register (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	PIM0.4	PIM0.3	0	PIM0.1	0	F0040H	00H	R/W
PIM1	PIM1.7	PIM1.6	PIM1.5	PIM1.4	PIM1.3	0	0	0	F0041H	00H	R/W
PIM5	0	0	PIM5.5	0	0	0	0	0	F0045H	00H	R/W

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

(5) Port output mode registers (POM0, POM1, POM5, POM7)

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

N-ch open drain output (V_{DD} tolerance/EV_{DD} tolerance[Note]) mode can be selected during serial communication with an external device of the different potential, and for the SDA00, SDA01, SDA10, SDA11, SDA20, and SDA21 pins during simplified I^2C communication with an external device of the same potential. In addition, these registers are combined with PUxx registers to specify whether to use an 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.

Figure 4-45. Format of Port Input Mode Register (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	POM0.4	POM0.3	POM0.2	0	POM0.0	FFF50H	00H	R/W
POM1	POM1.7	0	POM1.5	POM1.4	POM1.3	POM1.2	POM1.1	POM1.0	FFF51H	00H	R/W
POM5	0	0	POM5.5	0	0	0	0	POM5.0	FFF55H	00H	R/W
POM7	0	0	0	POM7.4	0	0	POM7.1	0	F007H	00H	R/W

POMm.n	Pmn pin output mode selection (m = 0, 1, 5, 7; n = 0 to 5, 7)
0	Normal output mode PUmn bit is enabled during input.
1	N-ch open-drain output (V_{DD} tolerance[Note1]/EV_{DD} tolerance[Note2]) mode PUmn bit is disabled during input.

Notes 1. For 20- to 48-pin products

2. For 64-pin products

(6) Port mode control registers (PMC0, PMC12, PMC14)

These registers set the P00, P01, P120, and P147 digital I/O/analog input in 1-bit units.

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

Reset signal generation clears these registers to FFH.

Figure 4-46. 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	PMC0.3 Note2	PMC0.2 Note2	PMC0.1 Note1	PMC0.0 Note1	F0060H	FFH	R/W
PMC12	1	1	1	1	1	1	1	PMC12.0 Note3	F006CH	FFH	R/W
PMC14	PMC14.7 Note3	1	1	1	1	1	1	1	F006EH	FFH	R/W

PMCm.n	Pmn pin digital I/O/analog input selection (m = 0, 12, 14; n = 0 to 3, 7)
0	Digital I/O (alternate function other than analog input)
1	Analog input

Notes 1. For 20-, 30-, 32-pin products

2. For 64-pin products

3. For 30-, 32-, 48-, 64-pin products

Cautions 1. Set the channels to be used for A/D conversion to input mode by the port mode registers 0, 12, 14 (PM0, PM12, PM14)

2. Do not set the pins by the analog input channel specification register (ADS) when the pins are to be specified as digital I/O by the PMC register.

(7) A/D port configuration register (ADPC)

This register switches the P20/ANI0 to P27/ANI7, and P150/ANI8 to P156/ANI14 pins to digital I/O of port or analog input 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 4-47. Format of A/D Port Configuration Register (ADPC) (64-pin products)

Address: F0076H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	0	ADPC.3	ADPC.2	ADPC.1	ADPC.0

ADPC.3	ADPC.2	ADPC.1	ADPC.0	Analog input (A)/digital I/O (D) switching							
				ANI7/P27	ANI6/P26	ANI5/P25	ANI4/P24	ANI3/P23	ANI2/P22	ANI1/P21	ANI0/P20
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	D	D	D	D	D	D	D	D
0	0	1	0	D	D	D	D	D	D	D	A
0	0	1	1	D	D	D	D	D	D	A	A
0	1	0	0	D	D	D	D	D	A	A	A
0	1	0	1	D	D	D	D	A	A	A	A
0	1	1	0	D	D	D	A	A	A	A	A
0	1	1	1	D	D	A	A	A	A	A	A
1	0	0	0	D	A	A	A	A	A	A	A
Other than above				Setting prohibited							

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode register 2 (PM2).

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

(8) Port mode registers X (PMX0 to PMX4)

These registers switch modes of some serial communication pins.

These registers are set when the pin SCKS0 (master mode), SOS0, TxDS0, LTxD, or SCKS1 is used.

The PMX0 to PMX4 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 01H.

Figure 4-48. Format of Port Mode Register X (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMX0	0	0	0	0	0	0	0	PMX0	F0504H	01H	R/W

PMX0	Selection of alternate function of P10/$\overline{\text{SCK00}}$/$\overline{\text{SCKS0}}$/SCL00 pin
0	$\overline{\text{SCKS0}}$ output (master mode)
1	$\overline{\text{SCKS0}}$ input (slave mode), or other alternate function (including general-purpose I/O port)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMX1	0	0	0	0	0	0	0	PMX1	F0505H	01H	R/W

PMX1	Selection of alternate function of P12/SO00/TxD0/SOS0/TxDS0/TOOLTxD pin
0	SOS0 or TxDS0 output
1	Other alternate function (including general-purpose I/O port)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMX2	0	0	0	0	0	0	0	PMX2	F0506H	01H	R/W

PMX2	Selection of alternate function of P51/INTP2/SO11/LTxD pin
0	LTxD0 output
1	Other alternate function (including general-purpose I/O port)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMX3	0	0	0	0	0	0	0	PMX3	F0507H	01H	R/W

PMX3	Selection of alternate function of P55/$\overline{\text{SCKS1}}$ pin
0	$\overline{\text{SCKS1}}$ output (master mode)
1	$\overline{\text{SCKS1}}$ input (slave mode) or other alternate function (including general-purpose I/O port)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMX4	0	0	0	0	0	0	0	PMX4	F0508H	01H	R/W

PMX4	Selection of alternate function of P53/SOS1 pin
0	SOS1 output (master mode)
1	Other alternate function (including general-purpose I/O port)

(9) Peripheral I/O redirection register (PIOR)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

This function is used to switch ports to which alternate functions are assigned.

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

Reset signal generation sets this register to 00H.

Figure 4-49. Format of Peripheral I/O redirection register (PIOR)

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIOR	0	0	0	PIOR4	PIOR3	PIOR2	PIOR1	PIOR0

bit	Function	64-pin		48-pin		32-pin		30-pin		20-pin	
		Setting Value		Setting Value		Setting Value		Setting Value		Setting Value	
		0	1	0	1	0	1	0	1	0	1
PIOR4	PCLBUZ1	P141	P55	Setting is unnecessary. This can be used regardless of value.							
	INTP5	P16	P12								
PIOR3	PCLBUZ0	P140	P31	P140	P31						
PIOR2	SCLA0	P60	P14	P60	P14	P60	P14	P60	P14	–	–
	SDAA0	P61	P13	P61	P13	P61	P13	P61	P13	–	–
PIOR1	INTP10	P76	P52	–	–	–	–	–	–	–	–
	INTP11	P77	P53	–	–	–	–	–	–	–	–
	TxD2	P13	P77	P13	–	P13	–	P13	–	–	–
	RxD2	P14	P76	P14	–	P14	–	P14	–	–	–
	SCL20	P15	–	P15	–	P15	–	P15	–	–	–
	SDA20	P14	–	P14	–	P14	–	P14	–	–	–
	SI20	P14	–	P14	–	P14	–	P14	–	–	–
	SO20	P13	–	P13	–	P13	–	P13	–	–	–
	$\overline{\text{SCK20}}$	P15	–	P15	–	P15	–	P15	–	–	–
	TxD0	P12	P17	P12	P17	P12	P17	P12	P17	P12	P17
	RxD0	P11	P16	P11	P16	P11	P16	P11	P16	P11	P16
	SCL00	P10	–	P10	–	P10	–	P10	–	P10	–
	SDA00	P11	–	P11	–	P11	–	P11	–	P11	–
	SI00	P11	P16	P11	–	P11	–	P11	–	P11	–
	SO00	P12	P17	P12	–	P12	–	P12	–	P12	–
	$\overline{\text{SCK00}}$	P10	P55	P10	–	P10	–	P10	–	P10	–
PIOR0	TI02/TO02	P17	P15	P17	P15	P17	P15	P17	P15	P17	–
	TI03/TO03	P31	P14	P31	P14	P31	P14	P31	P14	P31	–
	TI04/TO04	P42	P13	–	P13	–	P13	–	P13	–	–
	TI05/TO05	P05	P12	–	P12	–	P12	–	P12	–	P12
	TI06/TO06	P06	P11	–	P11	–	P11	–	P11	–	P11
	TI07/TO07	P41	P10	P41	P10	–	P10	–	P10	–	P10

(10) Port input enable register (PIEN)

This register specifies whether to use the input function of some serial communication pins.

This register is set when the SIS1 or SCKS1 pin (slave mode) is used.

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

Reset signal generation sets this register to 00H.

Figure 4-50. Format of Port Input Enable Register (PIEN)

Address: F0509H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIOR	0	0	0	0	0	0	0	PIEN0

PIEN0	Selection of alternate function of SIS1 or SCKS1 pin
0	Input function of SIS1 or SCKS1 pin (slave mode) is not used.
1	Input function of SIS1 and/or SCKS1 pin (slave mode) is used.

4.4 Port Function Operations

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

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change. Thus writing in byte units is possible in a port which includes both input and output 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.

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change. Thus writing in byte units is possible in a port which includes both input and output pins.
The data of the output latch is cleared when a reset signal is generated.

4.4.4 Connecting to external device with different potential (2.5 V, 3 V)

When parts of ports 0, 1, 5 operate with V_{DD} = 4.0 V to 5.5 V, I/O connections with an external device that operates on 2.5 V, 3 V power supply voltage are possible.

Regarding inputs, CMOS/TTL switching is possible on a bit-by-bit basis by the port input mode registers (PIM0, PIM1, PIM5).

Moreover, regarding outputs, different potentials can be supported by switching the output buffer to the N-ch open drain (V_{DD} tolerance/EV_{DD} tolerance[Note]) by the port output mode registers (POM0, POM1, POM5, POM7).

Note 64-bit products only

(1) Setting procedure when using I/O pins of UART0, UART1, UART2, CSI00, CSI10, and CSI20 functions

(a) Use as 2.5 V, 3 V input port

<1> After reset release, the port mode is the input mode (Hi-Z).

<2> If pull-up is needed, externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of UART0: (P16)
In case of UART1: P03
In case of UART2: P14
In case of CSI00: (P16, P55)
In case of CSI10: P03, P04
In case of CSI20: P14, P15

Remark Pins in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR).

<3> Set the corresponding bit of the PIM0, PIM1, PIM5 registers to 1 to switch to the TTL input buffer.

<4> V_{IH}/V_{IL} operates on 2.5 V, 3 V operating voltage.

(b) Use as 2.5 V, 3 V output port

<1> After reset release, the port mode changes to the input mode (Hi-Z).

<2> Pull up externally the pin to be used (on-chip pull-up resistor cannot be used).

In case of UART0: P12 (P17)
In case of UART1: P02
In case of UART2: P13
In case of CSI00: P12, P10 (P17, P55)
In case of CSI10: P02, P04
In case of CSI20: P13, P15

Remark Pins in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR).

<3> Set the output latch of the corresponding port to 1.

<4> Set the corresponding bit of the POM0, POM1, POM5 registers to 1 to set the N-ch open drain output (V_{DD} tolerance/EV_{DD} tolerance) mode.

<5> Set the output mode by manipulating the PM0, PM1, and PM5 registers.
At this time, the output data is high level, so the pin is in the Hi-Z state.

<6> Communication is started by setting the serial array unit.

(2) Setting procedure when using I/O pins of simplified IIC20 functions

<1> After reset release, the port mode is the input mode (Hi-Z).

<2> Externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of simplified IIC10: P03, P04
In case of simplified IIC20: P14, P15

<3> Set the output latch of the corresponding port to 1.

<4> Set the corresponding bit of the POM0 or POM1 register to 1 to set the N-ch open drain output (V_{DD} tolerance/EV_{DD} tolerance) mode.

<5> Set the corresponding bit of the POM0 or POM1 register to the output mode (data I/O is possible in the output mode).
At this time, the output data is high level, so the pin is in the Hi-Z state.

<6> Enable the operation of the serial array unit and set the mode to the simplified IIC mode.

4.5 Settings of Port Mode Register, and Output Latch When Using Alternate Function

To use the alternate function of a port pin, set the port mode register, and output latch as shown in Table 4-14.

Table 4-14. Settings of Port Mode Register, and Output Latch When Using Alternate Function (1/5)

Pin Name	Alternate Function		PIOR.×	PMC×.×	PM×.×	P×.×	PMX.×
	Function Name	I/O					
P00	TI00	Input	×	–	1	×	–
P01	TO00	Output	×	–	0	0	–
P02	ANI17	Input	×	1	1	×	–
	SO10	Output	×	0	0	1	–
	TxD1	Output	×	0	0	1	–
P03	ANI16	Input	×	1	1	×	–
	SI10	Input	×	0	1	×	–
	RxD1	Input	×	0	1	×	–
	SDA10	I/O	×	0	0	1	–
P04	$\overline{\text{SCK10}}$	Input	×	–	1	×	–
		Output	×	–	0	1	–
	SCL10	Output	×	–	0	1	–
P05	TI05	Input	0	–	1	×	–
	TO05	Output	0	–	0	0	–
P06	TI06	Input	0	–	1	×	–
	TO06	Output	0	–	0	0	–
P10	$\overline{\text{SCK00}}$	Input	0	–	1	×	1
		Output	0	–	0	1	1
	$\overline{\text{SCKS0}}$	Input	×	–	1	×	1
		Output Note 2	×	–	1	×	0
	SCL00	Output	0	–	0	1	1
	(TI07)	Input	1	–	1	×	1
	(TO07)	Output	1	–	0	0	1

Remarks 1. ×: don't care
PIOR.×: Peripheral I/O redirection register
POM×.×: Port output mode register
PMC×.×: Port mode control register
PM×.×: Port mode register
P×.×: Port output latch
PMX.×: PMX register of the port

2. The relationship between pins and their alternate functions shown in this table indicates the relationship when a 64-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIOR.×, PMC×.×, PM×.×, P×.×, and PMX.× settings remain the same.

3. Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

Table 4-14. Settings of Port Mode Register, and Output Latch When Using Alternate Function (2/5)

Pin Name	Alternate Function		PIOR.×	PMC×.×	PM×.×	P×.×	PMX.×
	Function Name	I/O					
P11	SI00	Input	0	–	1	×	–
	RxD0	Input	0	–	1	×	–
	SIS0	Input	×	–	1	×	–
	RxDS0	Input	×	–	1	×	–
	SDA00 Note 4	I/O	0	–	0	1	–
	(TI06)	Input	1	–	1	×	–
	(TO06)	Output	1	–	0	0	–
P12	SO00	Output	0	–	0	1	1
	TxD0	Output	0	–	0	1	1
	SOS0 Note 2	Output	×	–	1	×	0
	TxDS0 Note 2	Output	×	–	1	×	0
	(TI05)	Input	1	–	1	×	1
	(TO05)	Output	1	–	0	0	1
	(INTP5)	Input	1	–	1	×	1
P13	TxD2	Output	0	–	0	1	–
	SO20	Output	0	–	0	1	–
	(SDAA0)	I/O	1	–	0	0	–
	(TI04)	Input	1	–	1	×	–
	(TO04)	Output	1	–	0	0	–
P14	RxD2	Input	0	–	1	×	–
	SI20	Input	0	–	1	×	–
	SDA20 Note 4	I/O	0	–	0	1	–
	(SCLA0)	I/O	1	–	0	0	–
	(TI03)	Input	1	–	1	×	–
	(TO03)	Output	1	–	0	0	–

Remarks 1. ×: don't care
PIOR.×: Peripheral I/O redirection register
POM×.×: Port output mode register
PMC.× :Port mode control register
PM×.×: Port mode register
P×.×: Port output latch
PMX.×: PMX register of the port

2. The relationship between pins and their alternate functions shown in this table indicates the relationship when a 64-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIOR.×, PMC×.×, PM×.×, and P×.× settings remain the same.

3. Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

Table 4-14. Settings of Port Mode Register, and Output Latch When Using Alternate Function (3/5)

Pin Name	Alternate Function		PIOR.×	PMC×.×	PM×.×	P×.×
	Function Name	I/O				
P15	$\overline{\text{SCK20}}$	Input	0	–	1	×
		Output	0	–	0	1
	SCL20	Output	0	–	0	1
	(TI02)	Input	1	–	1	×
	(TO02)	Output	1	–	0	0
P16	TI01	Input	×	–	1	×
	TO01	Output	×	–	0	0
	INTP5	Input	0	–	1	×
	(RxD0)	Input	1	–	1	×
	(SI00)	Input	1	–	1	×
P17	TI02	Input	0	–	1	×
	TO02	Output	0	–	0	0
	(TxD0)	Output	1	–	0	1
	(SO00)	Output	1	–	0	1
P20	ANI0	Input	×	–	1	×
	AV_{REFP}	Input	×	–	1	×
P21	ANI1	Input	×	–	1	×
	AV_{REFM}	Input	×	–	1	×
P22 to P27 Note 3	ANI2 to ANI7 Note 3	Input	×	–	1	×
P30	INTP3	Input	×	–	1	×
	RTC1HZ	Output	×	–	0	0
	$\overline{\text{SCK11}}$	Input	×	–	1	×
		Output	×	–	0	1
	SCL11	Output	×	–	0	1
P31	TI03	Input	0	–	1	×
	TO03	Output	0	–	0	0
	INTP4	Input	×	–	1	×
	(PCLBUZ0)	Output	1	–	0	0

Remarks 1. ×: don't care
PIOR.×: Peripheral I/O redirection register
PMC×.×: Port mode control register
PM×.×: Port mode register
P×.×: Port output latch

2. The relationship between pins and their alternate functions shown in this table indicates the relationship when a 64-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIOR.×, PMC×.×, PM×.×, and P×.× settings remain the same.

3. Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

Table 4-14. Settings of Port Mode Register, and Output Latch When Using Alternate Function (4/5)

Pin Name	Alternate Function		PIOR.×	PMC×.×	PM×.×	P×.×	PMX.×	PIEN
	Function Name	I/O						
P40	TOOL0	I/O	×	–	×	×	–	–
P41	TI07	Input	0	–	1	×	–	–
	TO07	Output	0	–	0	0	–	–
P42	TI04	Input	0	–	1	×	–	–
	TO04	Output	0	–	0	0	–	–
P50	INTP1	Input	×	–	1	×	–	–
	SI11	Input	×	–	1	×	–	–
	SDA11 Note 4	I/O	×	–	0	1	–	–
	LRxD0	Input	×	–	1	×	–	–
P51	INTP2	Input	×	–	1	×	1	–
	SO11	Output	×	–	0	1	1	–
	LTxD0 Note 2	Output	×	–	1	×	0	–
P52	(INTP10)	Input	1	–	1	×	–	–
P53	SOS1	Output	×	–	1	×	0	–
	(INTP11)	Input	1	–	1	×	1	–
P54	SIS1	Input	×	–	1	×	–	1
P55	$\overline{\text{SCKS1}}$	Input	×	–	1	×	1	1
		Output	×	–	1	×	0	×
	(PCLBUZ1)	Output	1	–	0	0	1	×
	($\overline{\text{SCK00}}$)	Input	1	–	1	×	1	×
		Output	1	–	0	1	1	×
P60	SCLA0	I/O	0	–	0	0	–	–
P61	SDAA0	I/O	0	–	0	0	–	–
P70	KR0	Input	×	–	1	×	–	–
	$\overline{\text{SCK21}}$	Input	×	–	1	×	–	–
		Output	×	–	0	1	–	–
	SCL21	Output	0	–	0	1	–	–

Remarks 1. ×: don't care
PIOR.×: Peripheral I/O redirection register
POM×.×: Port output mode register
PMC×.×: Port mode control register
PM×.×: Port mode register
P×.×: Port output latch
PMX.×: PMX register of the port
PIEN: Port input enable register

2. The relationship between pins and their alternate functions shown in this table indicates the relationship when a 64-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIOR.×, PMC×.×, PM×.×, and P×.× settings remain the same.

3. Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

Table 4-14. Settings of Port Mode Register, and Output Latch When Using Alternate Function (5/5)

Pin Name	Alternate Function		PIOR.×	PMC×.×	PM×.×	P×.×
	Function Name	I/O				
P71	KR1	Input	×	–	1	×
	SI21	Input	×	–	1	×
	SDA21 Note 4	I/O	×	–	0	1
P72	KR2	Input	×	–	1	×
	SO21	Output	×	–	0	1
P73	KR3	Input	×	–	1	×
	SO01	Output	×	–	0	1
P74	KR4	Input	×	–	1	×
	INTP8	Input	×	–	1	×
	SI01	Input	×	–	1	×
	SDA01 Note 4	I/O	×	–	0	1
P75	KR5	Input	×	–	1	×
	INTP9	Input	×	–	1	×
	$\overline{SCK01}$	Input	×	–	1	×
		Output	×	–	0	1
	SCL01	Output	×	–	0	1
P76	KR6	Input	×	–	1	×
	INTP10	Input	0	–	1	×
	(RxD2)	Input	1	–	1	×
P77	KR7	Input	×	–	1	×
	INTP11	Input	0	–	1	×
	(TxD2)	Output	1	–	0	1
P120	ANI19 Note 1	Input	×	1	1	×
P137	INTP0	Input	×	–	–	×
P140	PCLBUZ0	Output	0	–	0	0
	INTP6	Input	×	–	1	×
P141	PCLBUZ1	Output	0	–	0	0
	INTP7	Input	×	–	1	×
P147	ANI18 Note 1	Input	×	1	1	×

Remarks 1. ×: don't care
PIOR.×: Peripheral I/O redirection register
POM×.×: Port output mode register
PMC×.×: Port mode control register
PM×.×: Port mode register
P×.×: Port output latch

2. The relationship between pins and their alternate functions shown in this table indicates the relationship when a 64-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIOR.×, PMC×.×, PM×.×, and P×.× settings remain the same.

3. Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

Notes 1. The functions of the ANI16/P03, ANI17/P02, ANI18/P147, ANI19/P120 pins can be selected by using the port mode control registers 0, 12, 14 (PMC0, PMC12, PMC14), analog input channel specification register (ADS), and port mode registers 0, 3, 10, 11, 12, 14 (PM0, PM3, PM10, PM11, PM12, PM14).

Table 4-15. Settings of Pins ANI16/P03, ANI17/P02, ANI18/P147, and ANI19/P120

PMC0, PMC12, PMC14 Registers	PM0, PM12, PM14 Registers	ADS Register	ANI16/P03, ANI17/P02, ANI18/P147, ANI19/P120 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input 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.	

2. To use P10/$\overline{\text{SCK00}}$/$\overline{\text{SCKS0}}$/SCL00, P12/SO00/TxD0/SOS0/TxDS0/TOOLTxD, P51/INTP2/SO11/LTxD, P53/SOS1, P55/$\overline{\text{SCKS1}}$ as a $\overline{\text{SCKS0}}$ output, SOS0 output, TxDS0 output, LTxD output, SOS1 output, or $\overline{\text{SCKS1}}$ output set the PMX0 to PMX4 registers.
See **4.3 (8) Port mode registers X (PMX0 to PMX4)** for details.

3. The functions of the ANI0/P20 to ANI7/P27 pins can be selected by using the A/D port configuration register (ADPC), analog input channel specification register (ADS), and port mode register 2 (PM2).

Table 4-16. Settings of Pins ANI0/P20 to ANI7/P27

ADPC Register	PM2 Register	ADS Register	ANI0/P20 to ANI7/P27 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input 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.	

4. To use a particular alternate output function of a pin to which multiple alternate output functions are assigned, outputs of unused alternate functions should be set to the same level as the initial state, in addition to the settings in Table 4-15.

4.6 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/F12.

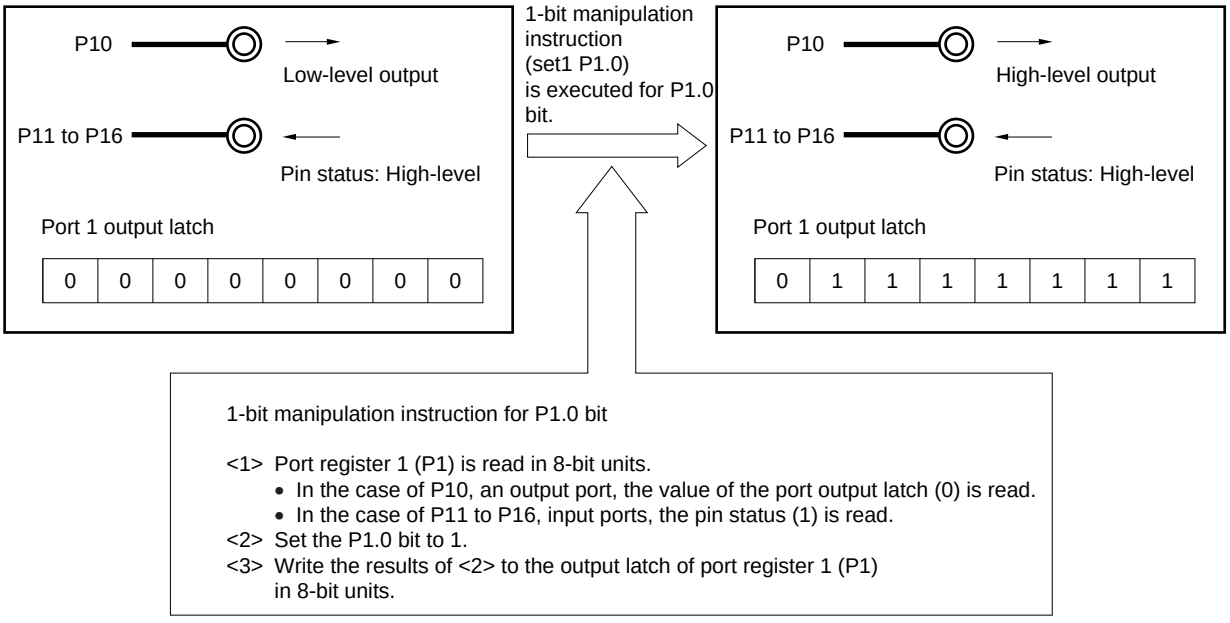
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

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 4-51. Bit Manipulation Instruction (P10)



CHAPTER 5 CLOCK GENERATOR

The presence or absence of connecting resonator pin for main system clock, connecting resonator pin for subsystem clock, external clock input pin for main system clock, and external clock input pin for subsystem clock, depends on the product.

Output pin	20, 30, 32-pin	48, 64-pin
X1, X2 pins	√	√
EXCLK pin	√	√
XT1, XT2 pins	–	√
EXCLKS pin	–	√

Caution The 20, 30, and 32-pin products don't have the subsystem clock.

5.1 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 and X2.
Oscillation can be stopped by executing the STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

<2> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IH} = 32, 24, 16, 12, 8, 4, or 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 the HIOSTOP bit (bit 0 of the CSC register).
The frequency set by the option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For the frequency, see **Figure 5-9 Format of High-Speed On-Chip Oscillator Frequency Select Register (HOCODIV)**.
The table below shows the oscillation frequencies that can be set for the high-speed on-chip oscillator (the variations selectable using the option byte and the high-speed on-chip oscillator frequency select register (HOCODIV).

Power supply voltage	Oscillation frequency (MHz)									
	1	2	3	4	6	8	12	16	24	32
$2.7V \le V_{DD} \le 5.5V$	√	√	√	√	√	√	√	√	√	√
$1.8V \le V_{DD} < 2.7V$	√	√	√	√	√	√	–	–	–	–

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 and XT2. 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 setting the XTSTOP bit.

(3) Low-speed on-chip oscillator clock

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
- Interval timer

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the operation speed mode control register (OSMC), or both are set to 1.
However, when WDTON = 1, WUTMMCK0 = 0, and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, oscillation of the low-speed on-chip oscillator stops if the HALT or STOP instruction is executed.

Caution The low-speed on-chip oscillator clock (f_{IL}) can be selected as the real-time clock operation clock only 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 clock frequency

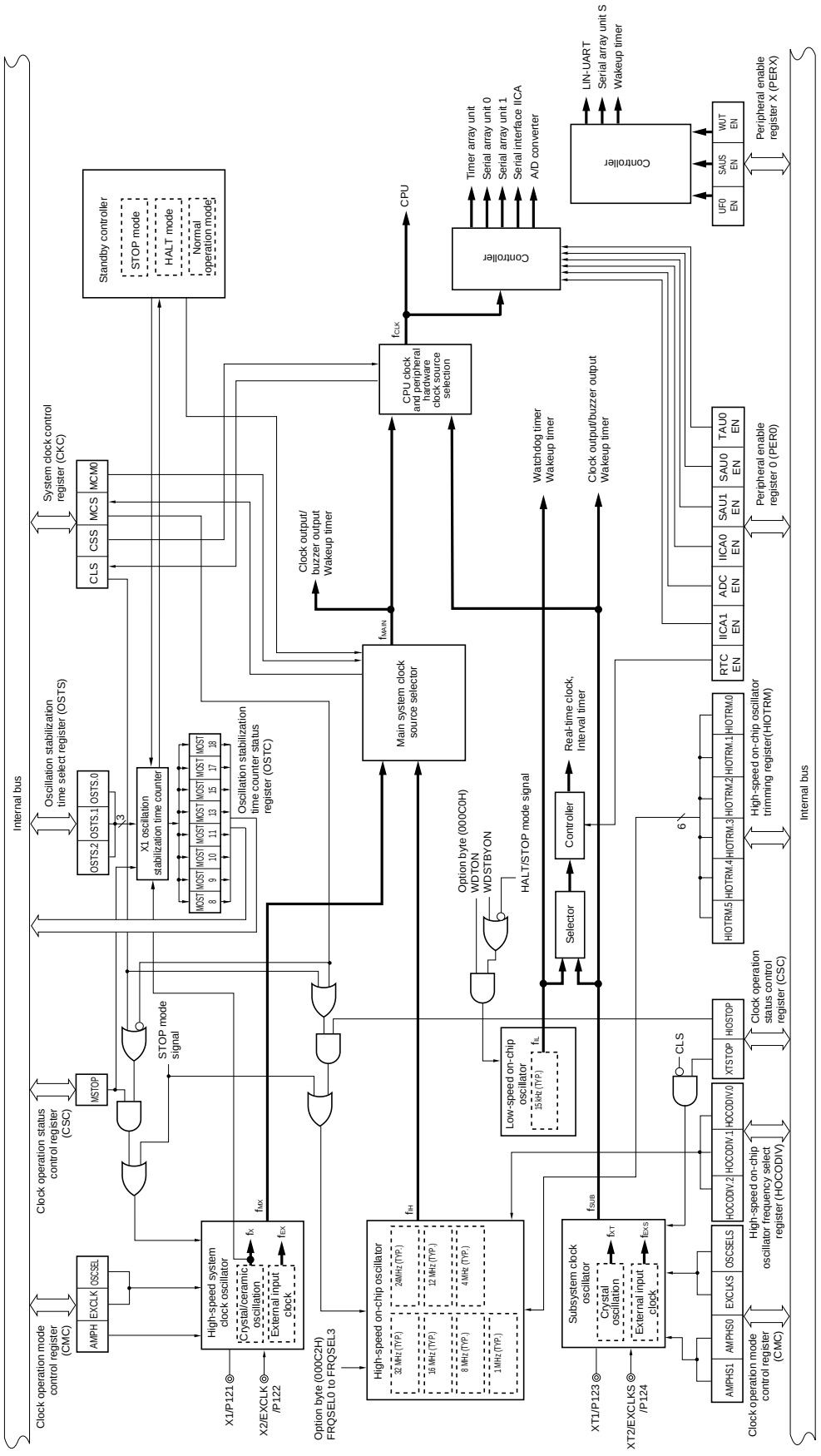
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) System clock control register (CKC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) Peripheral enable register 0 (PER0) Peripheral enable register X (PERX) Operation speed 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 5-1. 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

5.3 Registers Controlling Clock Generator

The following ten 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 register 0 (PER0)
- Peripheral enable register X (PERX)
- Operation speed mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)

5.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121, X2/EXCLK/P122, 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 a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

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

Address: FFFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	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/ceramic 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 ≤ f_X ≤ 10 MHz
<R> 1	1 MHz ≤ f_X ≤ 20 MHz

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.

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).

<R> **3. Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz. Also, if the X1 clock is oscillating at a frequency in the range from 1 to (but not including) 10 MHz, the margin of oscillation can be improved by setting the AMPH bit to 1.**

4. When the CMC register is used at the default value (00H), be sure to set 00H to this register after reset release in order to prevent malfunctioning during a program loop.

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

5. **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, use the recommended resonators described in CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE) and CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE).**
   - **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 V_{SS} 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.**
6. **Set the AMPH, AMPHS1, and AMPHS0 bits while f_{IH} is selected as f_{CLK} (before changing f_{CLK} to F_{MX}) after a reset release.**
7. **Count the oscillation stabilization time of f_{XT} by software.**
8. **Though the maximum frequency of the system clock is 32 MHz, the maximum frequency of the X1 oscillation circuit is 20 MHz.**

Remark f_X: X1 clock frequency

5.3.2 System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and a division ratio.

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

Reset signal generation sets this register to 00H.

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

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

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
CKC	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	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	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})

Notes 1. Bits 7 and 5 are read-only.

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

Remarks 1. 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

(**Cautions** are listed on the next page.)

Cautions 1. Be sure to set 0 in bits 3 to 0.

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, 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.

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 (J GRADE) and CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE).

5.3.3 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 5-4. Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H After reset: C0H R/W

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

Cautions

1. **After reset release, set the clock operation mode control register (CMC) before setting the CSC register.**
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.**
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).**
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.**
5. **Do not stop the clock selected as clock (f_{CLK}) of the CPU or the peripheral hardware with the CSC register.**
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 5-2.**

Table 5-2. Condition Before Stopping Clock Oscillation and Flag Setting

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

5.3.4 Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.

The X1 clock oscillation stabilization time can be checked in the following case,

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

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset signal is generated, the STOP instruction and MSTOP (bit 7 of clock operation status control register (CSC)) = 1 clear the OSTC register to 00H.

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

- When oscillation of the X1 clock starts (EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- When the STOP mode is released

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

Address: FFFA2H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

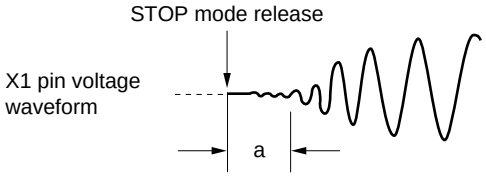
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									f_X = 10 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	$2^8/f_X$ max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	$2^8/f_X$ min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	$2^9/f_X$ min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	$2^{10}/f_X$ min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	$2^{11}/f_X$ min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	$2^{13}/f_X$ min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	$2^{15}/f_X$ min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_X$ min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_X$ min.	26.21 ms min.	13.11 ms min.

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

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 after the oscillation starts.

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

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



Remark f_X: X1 clock oscillation frequency

5.3.5 Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released.

When the X1 clock is selected as the CPU clock, the operation automatically waits for the time set using the OSTS register after the STOP mode is released.

When the high-speed on-chip oscillator clock is selected as the CPU clock, confirm with the oscillation stabilization time counter status register (OSTC) that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using the OSTC register.

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

Reset signal generation sets the OSTS register to 07H.

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

Address: FFFA3H After reset: 07H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS.2	OSTS.1	OSTS.0

OSTS.2	OSTS.1	OSTS.0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	Setting prohibited
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_X$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_X$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.21 ms	13.11 ms

Cautions 1. To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.

2. Setting the oscillation stabilization time to 20 μs or less is prohibited.

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

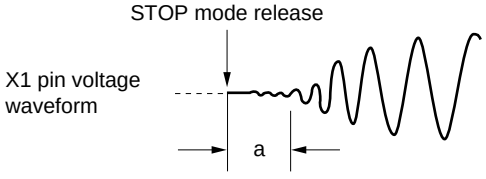
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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

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

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

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



Remark f_X: X1 clock oscillation frequency

5.3.6 Peripheral enable register 0 (PER0)

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 this register, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- Real-time clock, Interval timer
- A/D converter
- Serial interface IICA
- Serial array unit 0
- Serial array unit 1
- Timer array unit

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

Reset signal generation clears these registers to 00H.

Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (1/2)

Address: F00F0H After reset: 00H R/W

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

RTCEN	Control of real-time clock (RTC) and interval timer input clock supply Note
0	Stops input clock supply. • SFR used by the real-time clock (RTC) and interval timer cannot be written. • The real-time clock (RTC) and interval timer are in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) and interval timer 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 IICA input clock supply
0	Stops input clock supply. • SFR used by the serial interface IICA cannot be written. • The serial interface IICA is in the reset status.
1	Enables input clock supply. • SFR used by the serial interface IICA can be read and written.

Note The input clock that can be controlled by the RTCEN bit is used when the register that is used by the real-time clock (RTC) is accessed from the CPU. The RTCEN bit cannot control supply of the operating clock (f_{SUB}) to RTC.

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

Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (2/2)

Address: F00F0H After reset: 00H R/W

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

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.

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

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

5.3.7 Peripheral enable register X (PERX)

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

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

Reset signal generation clears these registers to 00H.

Caution Whether to enable or disable SFR writing only is selected for the 16-bit wakeup timer.
Whether to enable or disable supplying the operating clock is selected using the PCKSEL register.

Figure 5-8. Format of Peripheral Enable Register X (PERX)

Address: F0500H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PERX	0	0	0	0	0	UF0EN	SAUSEN	WUTEN

UF0EN	LIN-UART0 input clock control
0	Stops input clock supply. • Writing to SFR to be used with LIN-UART0 is disabled. • LIN-UART0 is in reset state.
1	Supplies input clock. • Reading from and writing to SFR to be used with LIN-UART0 is enabled.

SAUmEN	Control of serial array unit m input clock supply (m = 0, 1, S)
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.

WUTEN	Control of 16-bit wakeup timer input clock
0	Stops input clock supply for SFR writing. • SFR used by the 16-bit wakeup timer cannot be written.
1	Supplies input clock for SFR writing . • SFR used by the 16-bit wakeup timer can be written.

Caution Be sure to clear the bits 3 to 7 of the PERX register to 0.

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

The frequency of the high-speed on-chip oscillator which is set by an option byte.(000C2H/0102CH) can be changed by using high-speed on-chip oscillator frequency select register (HOCODIV). However, the selectable frequency depends on the FRQSEL3 bit of the option byte (000C2H/0102CH).

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

Reset signal generation clears this register to default value (undefined).

Figure 5-9. Format of High-Speed On-Chip Oscillator Frequency Select Register (HOCODIV)

Address: F00A8H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV.2	HOCODIV.1	HOCODIV.0

HOCODIV.2	HOCODIV.1	HOCODIV.0	Selection of High-Speed On-Chip Oscillator Clock Frequency	
			FRQSEL3 Bit is 0	FRQSEL3 Bit of is 1
0	0	0	24 MHz	32 MHz
0	0	1	12 MHz	16 MHz
0	1	0	6 MHz	8 MHz
0	1	1	3 MHz	4 MHz
1	0	0	Setting prohibited	2 MHz
1	0	1	Setting prohibited	1 MHz
Other than above			Setting prohibited	

Caution 1. Set the HOCODIV register within the operable voltage range both before and after changing the frequency.

2. Use the device within the voltage of the flash operation mode set by the option byte (000C2H/010C2H) even after the frequency has been changed by using the HOCODIV register.

<R>

Option Byte (000C2H/010C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode	1 to 32 MHz	2.7 to 5.5 V

3. The device operates at the old frequency for the duration of 3 clocks after the frequency value has been changed by using the HOCODIV register. Moreover, when the high-speed on-chip oscillator clock is selected as the system clock, the device waits for the oscillation to stabilize for an additional duration of 3 clocks,

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

5.3.9 Operation speed mode control register (OSMC)

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

If the RTCLPC bit is set to 1, current consumption can be reduced, because the circuit that synchronizes the clock to the peripheral functions, except the real-time clock and interval timer, is stopped in STOP mode or HALT mode while subsystem clock is selected as CPU clock. Before setting the RTCLPC bit to 1, set bit 7 (RTCEN) of the peripheral enable register 0 (PER0) to 1.

In addition, the OSMC register can be used to select the operation clock of the real-time clock and interval timer.

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

Reset signal generation clears this register to 00H.

Figure 5-10. Format of Operation Speed 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 **Table 20-1. Operating Statuses in HALT Mode** for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than real-time clock and interval timer.

WUTMMCK0	Selection of operation clock for real-time clock and interval timer.
0	Subsystem clock
1	Low-speed on-chip oscillator clock

Caution The STOP mode current or HALT mode current when the subsystem clock is used can be reduced by setting the RTCLPC bit to 1. However, no clock can be supplied to the peripheral functions other than the real-time clock and interval timer during HALT mode while subsystem clock is selected as CPU clock. Set bit 7 (RTCEN) of peripheral enable registers 0 (PER0), to 1, and bits 0 to 6 of the PER0 register to 0 before setting subsystem clock HALT mode.

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

This register is used to adjust the accuracy of the high-speed on-chip oscillator.

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

Reset signal generation clears this register to default value [Note].

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

Address: F00A0H After reset: See Note. R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	HIOTRM.5	HIOTRM.4	HIOTRM.3	HIOTRM.2	HIOTRM.1	HIOTRM.0

HIOTRM.5	HIOTRM.4	HIOTRM.3	HIOTRM.2	HIOTRM.1	HIOTRM.0	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 reset value depends on the individual chip.

5.4 System Clock Oscillator

5.4.1 X1 oscillator

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

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

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

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

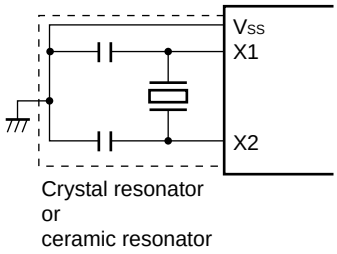
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

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

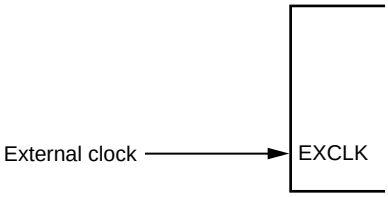
Figure 5-10 shows an example of the external circuit of the X1 oscillator.

Figure 5-12. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Cautions are listed on the next page.

5.4.2 XT1 oscillator

The XT1 oscillator oscillates with a crystal resonator (standard: 32.768 kHz) 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 or ceramic oscillation: EXCLKS, OSCSELS = 0, 1
- External clock input: EXCLKS, OSCSELS = 1, 1

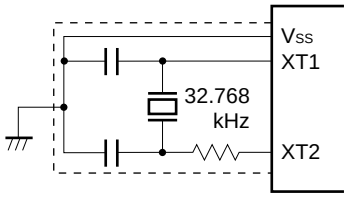
When the XT1 oscillator is not used, set the input port mode (EXCLKS, OSCSELS = 0, 0).

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

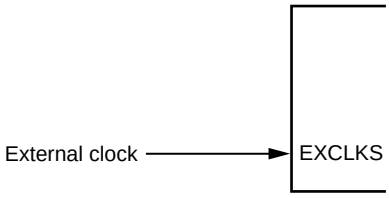
Figure 5-13 shows an example of the external circuit of the XT1 oscillator.

Figure 5-13. Example of External Circuit of XT1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Caution When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-10 and 5-11 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.**

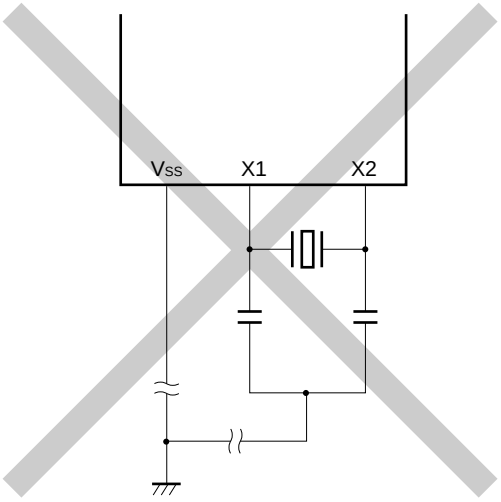
The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **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, use the recommended resonators described in CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE) and CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE).**
- **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.**

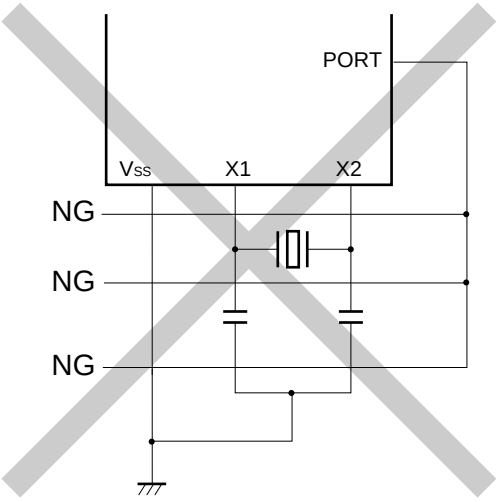
Figure 5-14 shows examples of incorrect resonator connection.

Figure 5-14. Examples of Incorrect Resonator Connection (1/2)

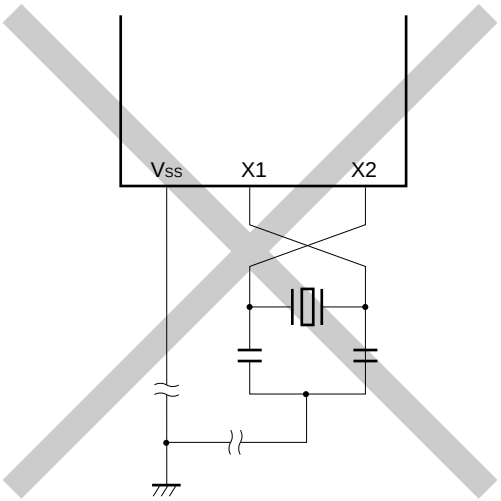
(a) Too long wiring



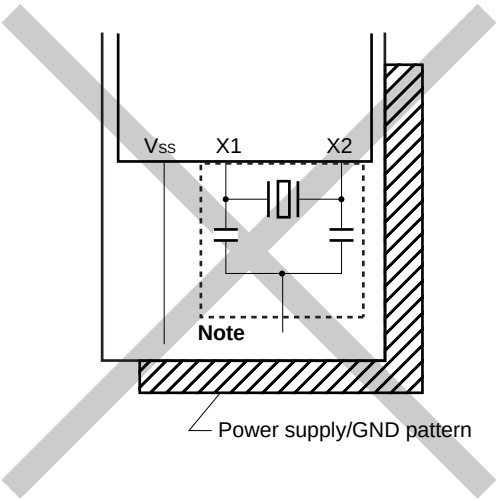
(b) Crossed signal line



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



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

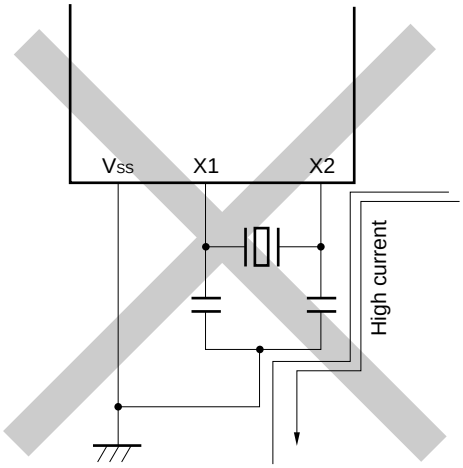


Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

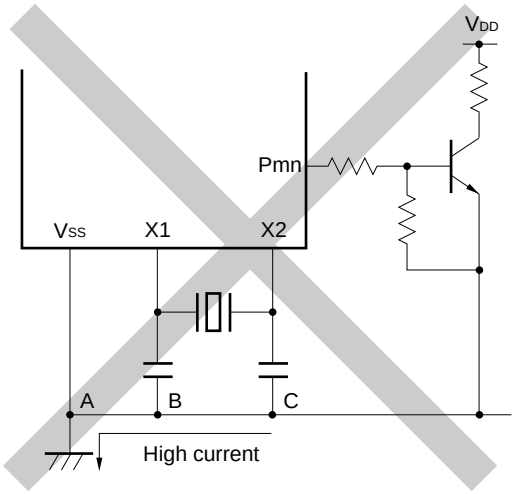
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-14. Examples of Incorrect Resonator Connection (2/2)

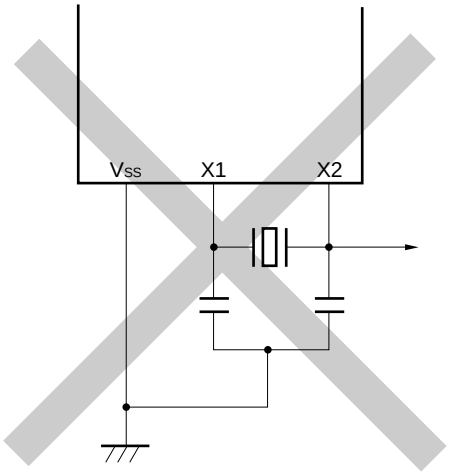
(e) Wiring near high alternating current



(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.

5.4.3 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/F12. The frequency can be selected from among 32, 24, 16, 12, 8, 4, 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.

5.4.4 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/F12.

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

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

Unless the watchdog timer is stopped and WUTMMCK0 is a value other than zero, oscillation of the low-speed on-chip oscillator continues. While the watchdog timer operates, the low-speed on-chip oscillator clock does not stop even if the program freezes.

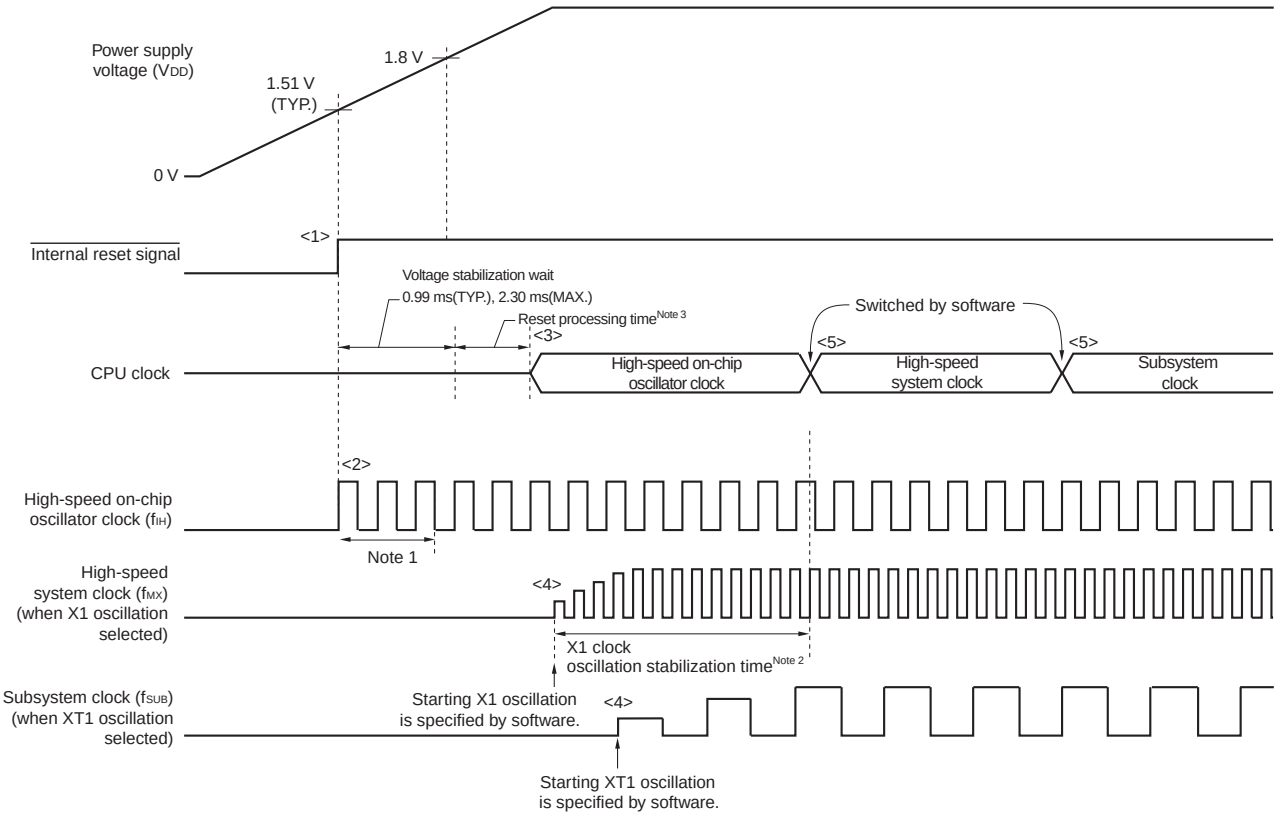
5.5 Clock Generator Operation

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

- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - High-speed on-chip oscillator clock f_{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/F12. When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-15.

Figure 5-15. 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.

<2> When the power supply voltage exceeds 1.51 V (TYP.), the reset is released and the high-speed on-chip oscillator automatically starts oscillation.

<3> The CPU starts operation on the high-speed on-chip oscillator clock after a reset processing such as waiting for the voltage of the power supply or regulator to stabilize has been performed after reset release.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **5.6.2 Example of setting X1 oscillation clock** and **5.6.3 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 **5.6.2 Example of setting X1 oscillation clock** and **5.6.3 Example of setting XT1 oscillation clock**).

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

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

<R> **3.** After the power is turned on and the voltage stabilization wait time has elapsed, the following reset processing time is required release from the reset state (POR), i.e. when the power supply voltage reaches 1.51 V (TYP.) and the $\overline{\text{RESET}}$ signal is driven to the high level.

Following initial release from the POR state:

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)

The following reset processing time is required following the second and subsequent releases from the reset state, i.e. when the $\overline{\text{RESET}}$ signal is asserted (low level) and then de-asserted (high level), as in a reset where a POR is not generated

Following the second and subsequent releases from the reset state:

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)

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

5.6 Controlling Clock

5.6.1 Example of setting high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 32, 24, 16, 12, 8, 4, and 1 MHz by using FRQSEL0 to FRQSEL3 of the option byte (000C2H). The frequency can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

Option byte (000C2H)

7	6	5	4	3	2	1	0
CMODE1 0/1	CMODE0 0/1	1	0	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

CMODE1	CMODE0	Setting of flash operation mode
1	0	LS (low speed main) mode
1	1	HS (high speed main) mode

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
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
1	0	1	1	4 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

[Internal high-speed oscillator frequency select register (HOCODIV) setting]

Address: F00A8H

HOCODIV

7	6	5	4	3	2	1	0
0	0	0	0	0	HOCODIV.2	HOCODIV.1	HOCODIV.0

HOCODIV.2	HOCODIV.1	HOCODIV.0	Selection of High-speed on-chip oscillator clock frequency	
			FRQSEL3 Bit is 0	FRQSEL3 Bit of is 1
0	0	0	24 MHz	32 MHz
0	0	1	12 MHz	16 MHz
0	1	0	6 MHz	8 MHz
0	1	1	3 MHz	4 MHz
1	0	0	Setting prohibited	2 MHz
1	0	1	Setting prohibited	1 MHz
Other than above			Setting prohibited	

5.6.2 Example of setting X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the X1 oscillation clock, set the oscillator and start oscillation by using the 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 <4> below.

<1> Set (1) the OSCSEL bit and the AMPH bit (in the case of fX > 10 MHz) of the CMC register to operate the X1 oscillator.

CMC

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

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

<2> Clear (0) the MSTOP bit of the CSC register to start oscillating the X1 oscillator.

CSC

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

<3> Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.

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

OSTC

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

<4> Use the MCM0 bit of the CKC register to specify the X1 oscillation clock as the CPU/peripheral hardware clock.

CKC

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

5.6.3 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 operation speed 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> To run only the real-time clock and interval timer on the subsystem clock (ultra-low current consumption) when in the STOP mode or sub-HALT mode, set the RTCLPC bit to 1.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC			WUTMMCK0				
	0/1	0	0	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	OSCSEL	EXCLKS	OSCSELS		AMPHS1	AMPHS0	AMPH
	0	0	0	1	0	0/1	0/1	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	XTSTOP						HIOSTOP
	1	0	0	0	0	0	0	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	CSS	MCS	MCM0				
	0	1	0	0	0	0	0	0

5.6.4 CPU clock status transition diagram

Figure 5-16 shows the CPU clock status transition diagram of this product.

Figure 5-16. CPU Clock Status Transition Diagram

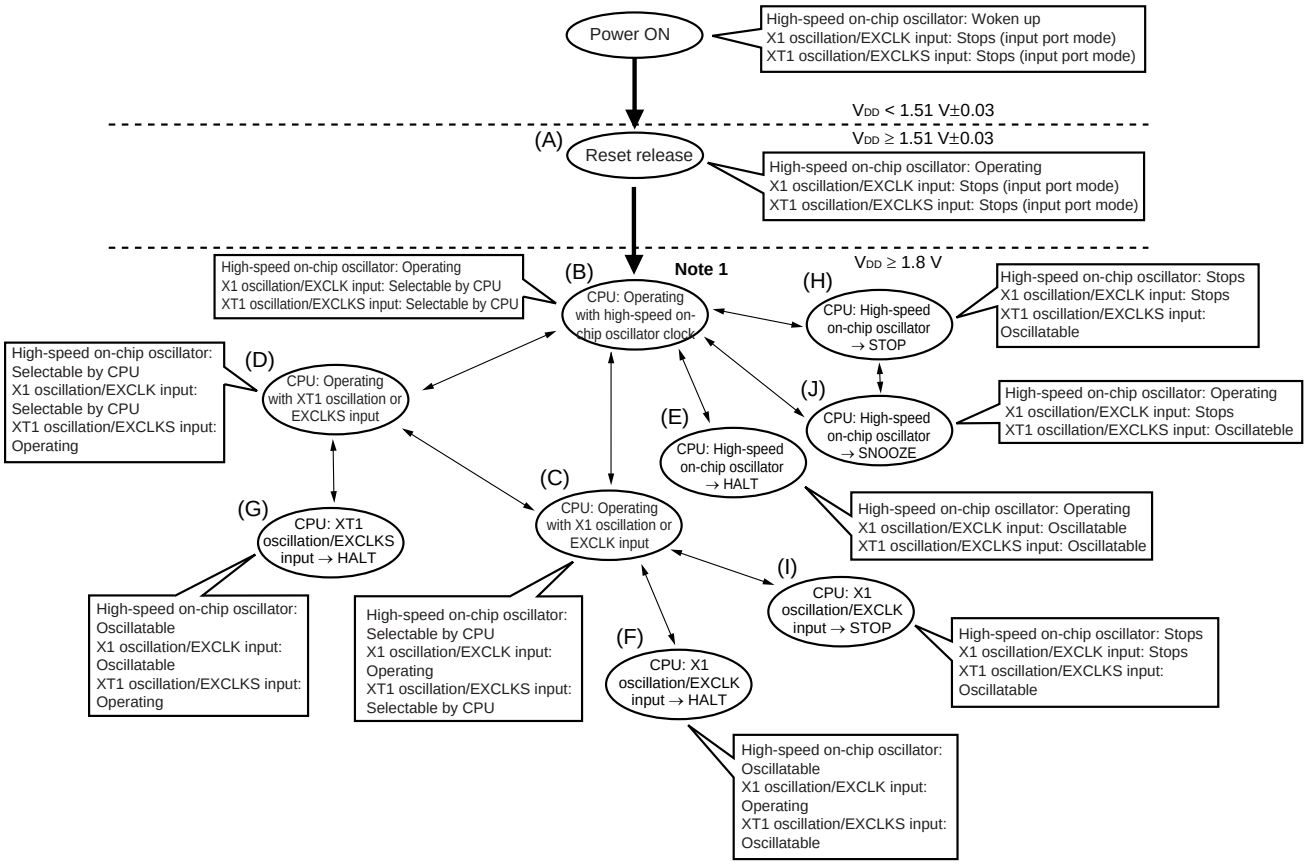


Table 5-3 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-3. 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			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A) → (B) → (C) (X1 clock: 1 MHz ≤ f_X ≤ 10 MHz)	0	1	0	Note 2	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < f_X ≤ 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

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

2. Set the oscillation stabilization time as follows.

- Desired 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 (J GRADE) and CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)).

(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) → (D) (XT1 clock)	0	1	0/1	0/1	0	Necessary	1
(A) → (B) → (D) (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.

Remarks 1. ×: don't care

2. (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-16.

Table 5-3. 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) (X1 clock: 1 MHz ≤ fX ≤ 10 MHz)	0	1	0	**Note 2**	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < fX ≤ 20 MHz)	0	1	1	**Note 2**	0	Must be checked	1
(B) → (C) (external main clock)	1	1	×	**Note 2**	0	Must not be checked	1

Unnecessary if these registers are already set

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

Notes 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.

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 (J GRADE) and CAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)).

(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	XTSTOP		CSS
(B) → (D) (XT1 clock)	0	1	0	Necessary	1
(B) → (D) (external sub clock)	1	1	0	Necessary	1

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.

Remarks 1. ×: don't care

2. (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-16.

Table 5-3. 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	30 μs	0

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

(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	CKC Register	
	HIOSTOP	MCM0	CSS
(D) → (B)	0	0	0

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

Unnecessary if this register is already set

Remark (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-16.

Table 5-3. 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		MCM0	CSS
(D) → (C) (X1 clock: 1 MHz ≤ f_X ≤ 10 MHz)	**Note**	0	Must be checked	1	0
(D) → (C) (X1 clock: 10 MHz < f_X ≤ 20 MHz)	**Note**	0	Must be checked	1	0
(D) → (C) (external main clock)	**Note**	0	Must not be checked	1	0

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

Unnecessary if these registers are already set

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 (J GRADE) and CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)).

(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 Table 5-3 correspond to (A) to (J) in Figure 5-16.

Table 5-3. 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 **12.8 SNOOZE Mode Function**, **13.5.7 SNOOZE mode function (only CSI00)** and **13.6.3 SNOOZE mode function (only UART0 reception)**.

Remark (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-16.

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

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

Table 5-4. 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	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	Oscillation of high-speed on-chip oscillator • HIOSTOP = 0	X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	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	X1 oscillation can be stopped (MSTOP = 1).
External main system clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator • HIOSTOP = 0	External main system clock input can be disabled (MSTOP = 1).
	X1 clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	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	External main system clock input can be disabled (MSTOP = 1).

Table 5-4. 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	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 (To change the clock, set it again after executing reset once.)	–
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	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 (To change the clock, set it again after executing reset once.)	–

5.6.6 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 Table 5-5 to Table 5-7).

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 5-5. Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	◄►	f_{MX}	See Table 5-6
f_{MAIN}	◄►	f_{SUB}	See Table 5-7

Table 5-6. 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 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$		1 + f_{IH}/f_{MX} clock
	$f_{MX} < f_{IH}$		$2f_{IH}/f_{MX}$ clock
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{IH}$	$2f_{MX}/f_{IH}$ clock	
	$f_{MX} < f_{IH}$	1 + f_{MX}/f_{IH} clock	

Table 5-7. 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 + $2f_{MAIN}/f_{SUB}$ clock
1 ($f_{CLK} = f_{SUB}$)	2 + f_{SUB}/f_{MAIN} clock	

(**Remarks 1 and 2** are listed on the next page.)

Remarks 1. The number of clocks listed in Table 5-6 to Table 5-7 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Table 5-6 to Table 5-7 by removing the decimal portion.

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

$1 + f_{IH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 \rightarrow 2$ clocks

5.6.7 Conditions before clock oscillation is stopped

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

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

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
High-speed on-chip oscillator clock	MCS = 1 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

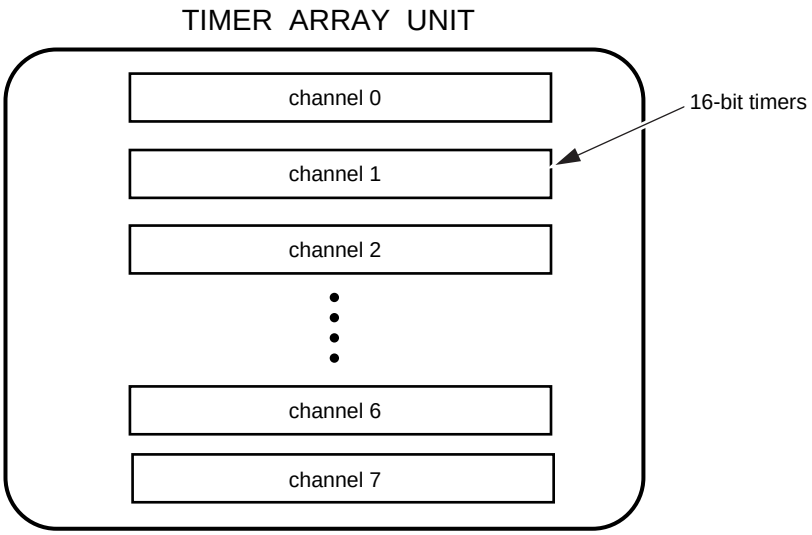
CHAPTER 6 TIMER ARRAY UNIT

Cautions 1. The presence or absence of timer I/O pins depends on the product. See Table 6-2 Timer I/O Pins provided in Each Product for details.

2. Most of the following descriptions in this chapter use the 64-pin products as an example.

The timer array unit has eight 16-bit timers.

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



For details about each function, see the table below.

Independent channel operation function	Simultaneous channel operation function
• Interval timer/square wave output (→ refer to 6.7.1) • Square wave output (→ refer to 6.7.1) • External event counter (→ refer to 6.7.2) • Divider function Note (→ refer to 6.7.3) • Input pulse interval measurement (→ refer to 6.7.4) • Measurement of high-/low-level width of input signal (→ refer to 6.7.5) • Delay counter (→ refer to 6.7.6)	• One-shot pulse output(→ refer to 6.8.1) • PWM output(→ refer to 6.8.2) • Multiple PWM output(→ refer to 6.8.3)

Note Channel 0 only

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

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

Channel 7 can be used to realize LIN-bus reception processing in combination with UART2 of the serial array unit (30, 32, 48, and 64-pin products only).

6.1 Functions of Timer Array Unit

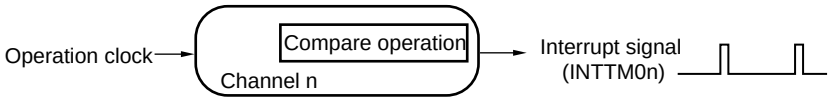
Timer array unit has the following functions.

6.1.1 Independent channel operation function

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

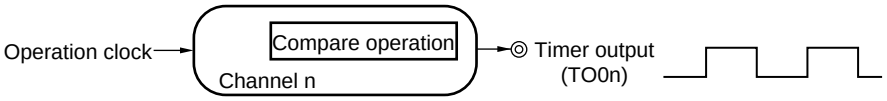
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTM0n) at fixed intervals.



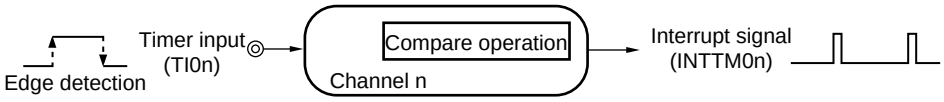
(2) Square wave output

A toggle operation is performed each time INTTM0n interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TO0n).



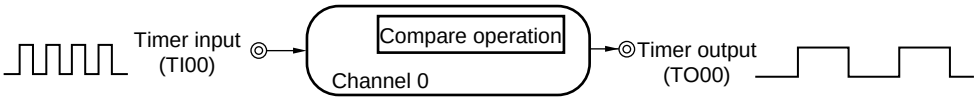
(3) External event counter

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 (TI0n) has reached a specific value.



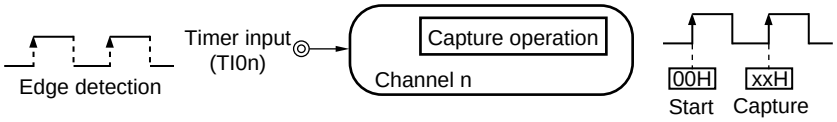
(4) Divider function (channel 0 only)

A clock input from a timer input pin (TI00) is divided and output from an output pin (TO00).



(5) Input pulse interval measurement

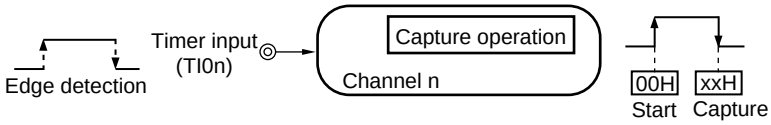
Counting is started by the valid edge of a pulse signal input to a timer input pin (TI0n). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



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

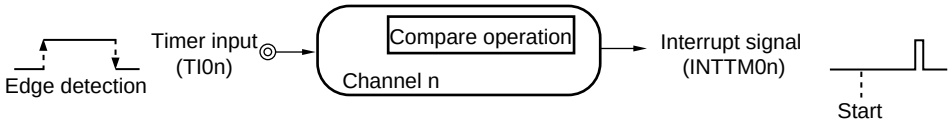
(6) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TI0n), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



(7) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TI0n), and an interrupt is generated after any delay period.



Remarks 1 n: Channel number (n = 0 to 7)

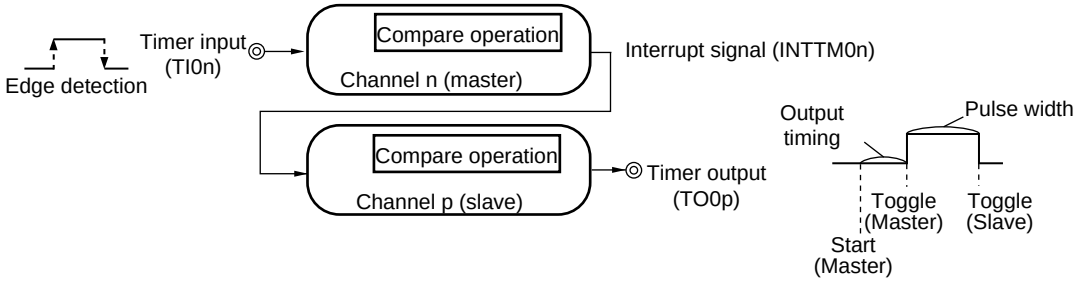
2. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

6.1.2 Simultaneous channel operation function

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

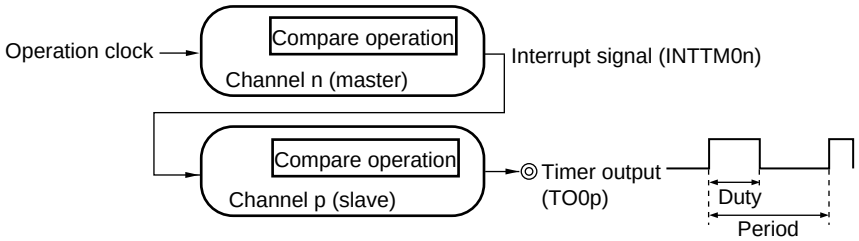
(1) One-shot pulse output

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



(2) PWM (Pulse Width Modulation) output

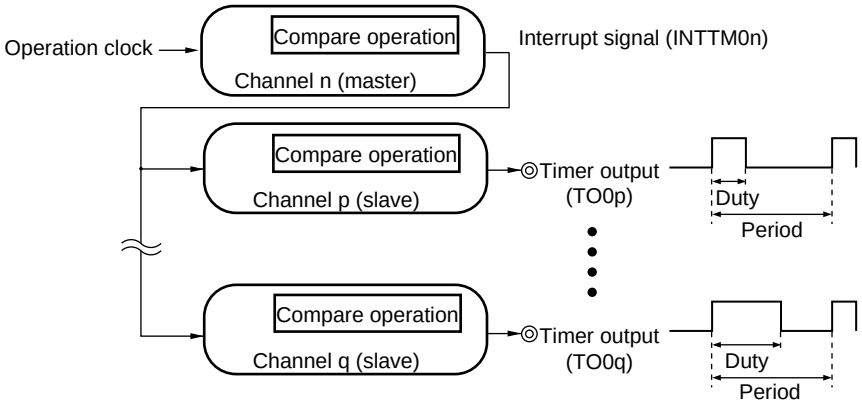
Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



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

(3) Multiple PWM (Pulse Width Modulation) output

By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.



Caution For details about the rules of simultaneous channel operation function, see 6.4.1 Basic Rules of Simultaneous Channel Operation Function.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)
p, q: Slave channel number (n < p < q ≤ 7)

6.1.3 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

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

6.1.4 LIN-bus supporting function (channel 7 only)

Timer array unit is used to check whether signals received in LIN-bus communication match the LIN-bus communication format.

(1) Detection of wakeup signal

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD2) of UART2 and the count value of the timer is captured at the rising edge. In this way, a low-level width can be measured. If the low-level width is greater than a specific value, it is recognized as a wakeup signal.

(2) Detection of break field

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD2) of UART2 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a low-level width is measured. If the low-level width is greater than a specific value, it is recognized as a break field.

(3) Measurement of pulse width of sync field

After a break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD2) of UART2 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

Remark For details about setting up the operations used to implement the LIN-bus, see **6.3 (13) Input switch control register (ISC)** and **6.7.5 Operation as input signal high-/low-level width measurement**.

6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 6-1. Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer/counter register 0n (TCR0n)
Register	Timer data register 0n (TDR0n)
Timer input	TI00 to TI07 Note 1, RxD2 pin (for LIN-bus)
Timer output	TO00 to TO07 pins Note 1, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register 0 (TPS0) • Timer channel enable status register 0 (TE0) • Timer channel start register 0 (TS0) • Timer channel stop register 0 (TT0) • Timer input select register 0 (TIS0) • Timer output enable register 0 (TOE0) • Timer output register 0 (TO0) • Timer output level register 0 (TOL0) • Timer output mode register 0 (TOM0)
	<Registers of each channel> • Timer mode register 0n (TMR0n) • Timer status register 0n (TSR0n) • Input switch control register (ISC) • Noise filter enable register 1 (NFEN1) • Port mode control register (PMCxx) Note 2 • Port mode register (PMxx) Note 2 • Port register (Pxx) Note 2

Notes 1. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

2. The port mode control registers (PMCxx), port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. for details, **see 6.3 (15) Port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4).**

Remark n: Channel number (n = 0 to 7)

The presence or absence of timer I/O pins in each timer array unit channel depends on the product.

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

Timer array unit channels		I/O Pins of Each Product			
		64-pin	48-pin	30, 32-pin	20-pin
Unit 0	Channel 0	P00/TI00, P01/TO00			P01/TO00
	Channel 1	P16/TI01/TO01			
	Channel 2	P17/TI02/TO02 (P15)			P17/TI02/TO00
	Channel 3	P31/TI03/TO03 (P14)			P31/TI03/TO03
	Channel 4	P42/TI04/TO04 (P13)	(P13)		–
	Channel 5	P05/TI05/TO05 (P12)	(P12)		
	Channel 6	P06/TI06/TO06 (P11)	(P11)		
	Channel 7	P41/TI07/TO07 (P10)		(P10)	

Remarks 1. When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

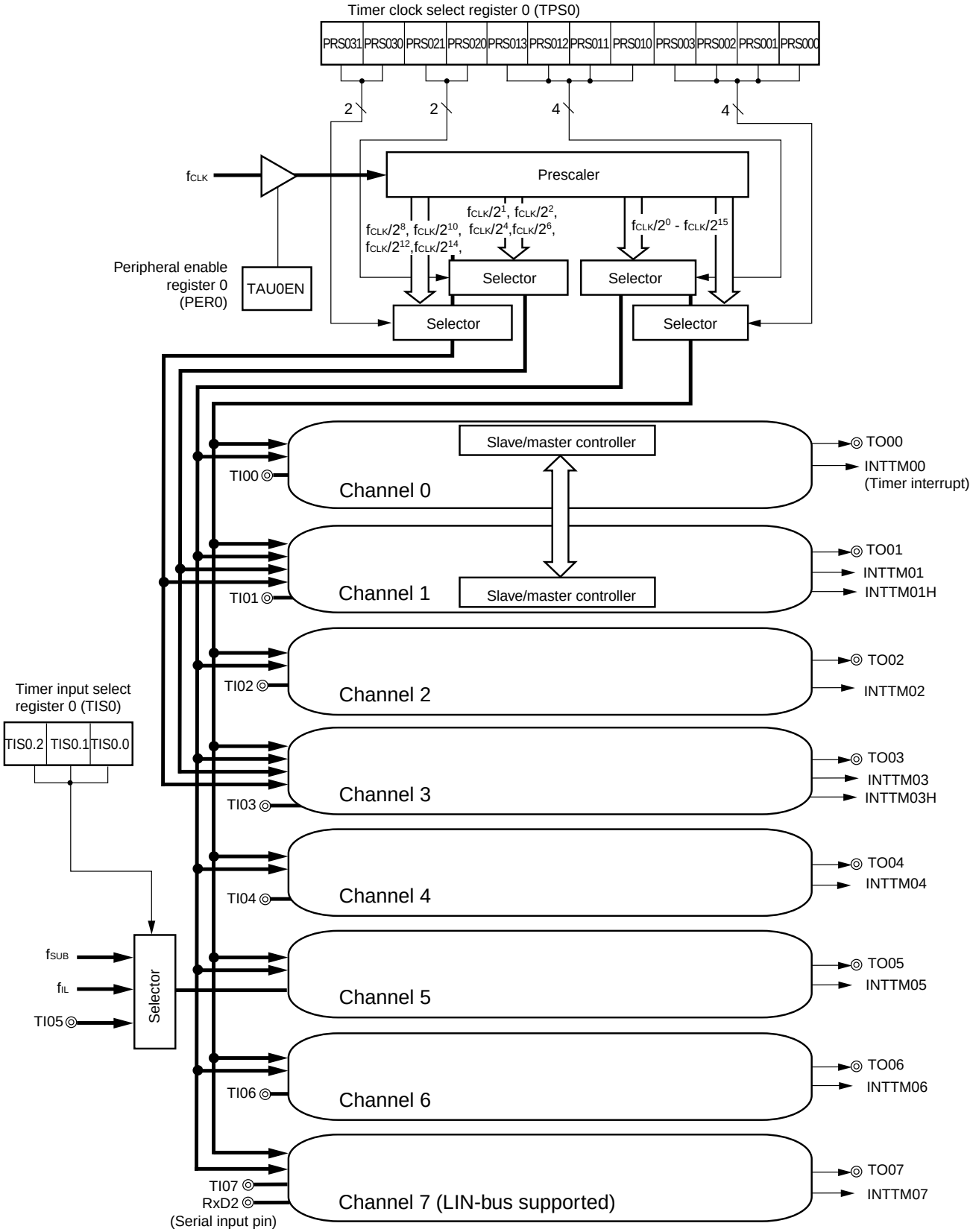
2. –: There is no timer I/O pin, but the channel is available. (However, the channel can only be used as an interval timer.)

×: The channel is not available.

3. "(P1x)" indicates an alternate port when the bit 0 of the peripheral I/O redirection register (PIOR) is set to "1".

Figures 6-1 and 6-2 show the block diagrams of the timer array unit.

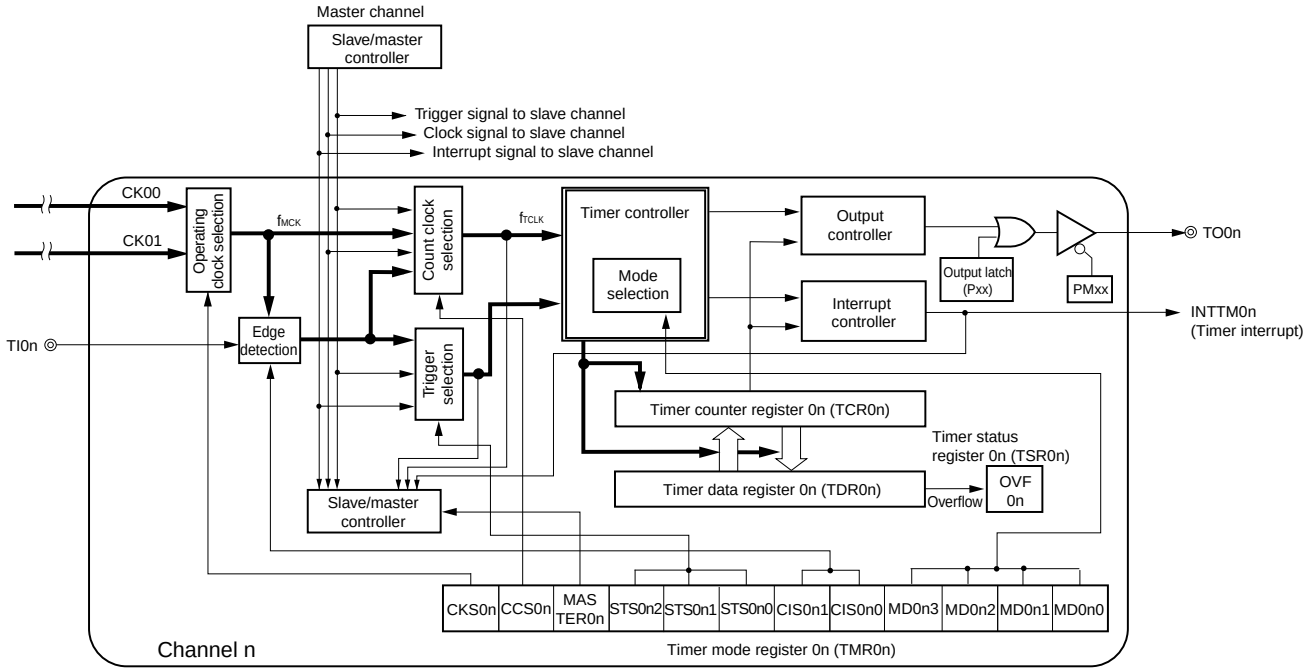
Figure 6-1. Entire Configuration of Timer Array Unit (Example: 64-pin products)



Remark f_{SUB}: Subsystem clock frequency

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

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



Remark n = 0, 2, 4, 6

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

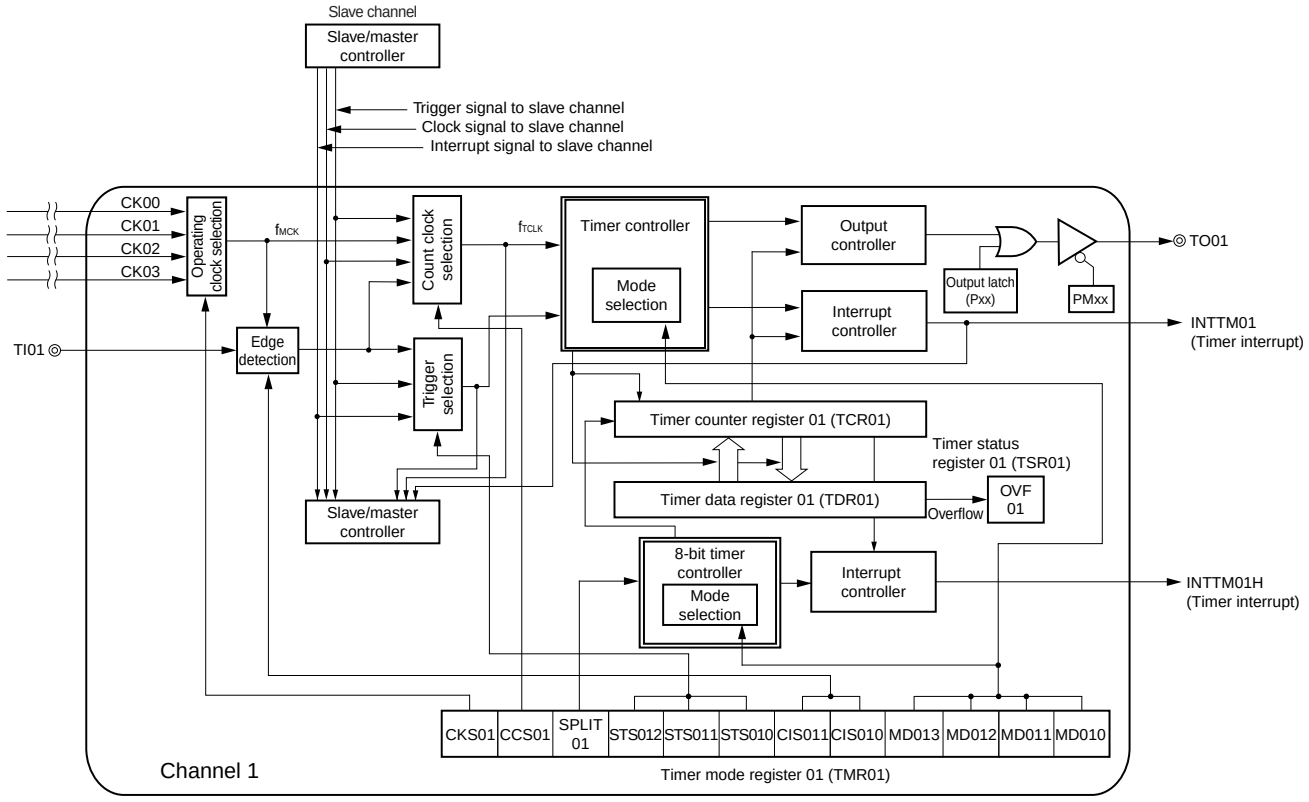


Figure 6-4. Internal Block Diagram of Channel 3 of Timer Array Unit

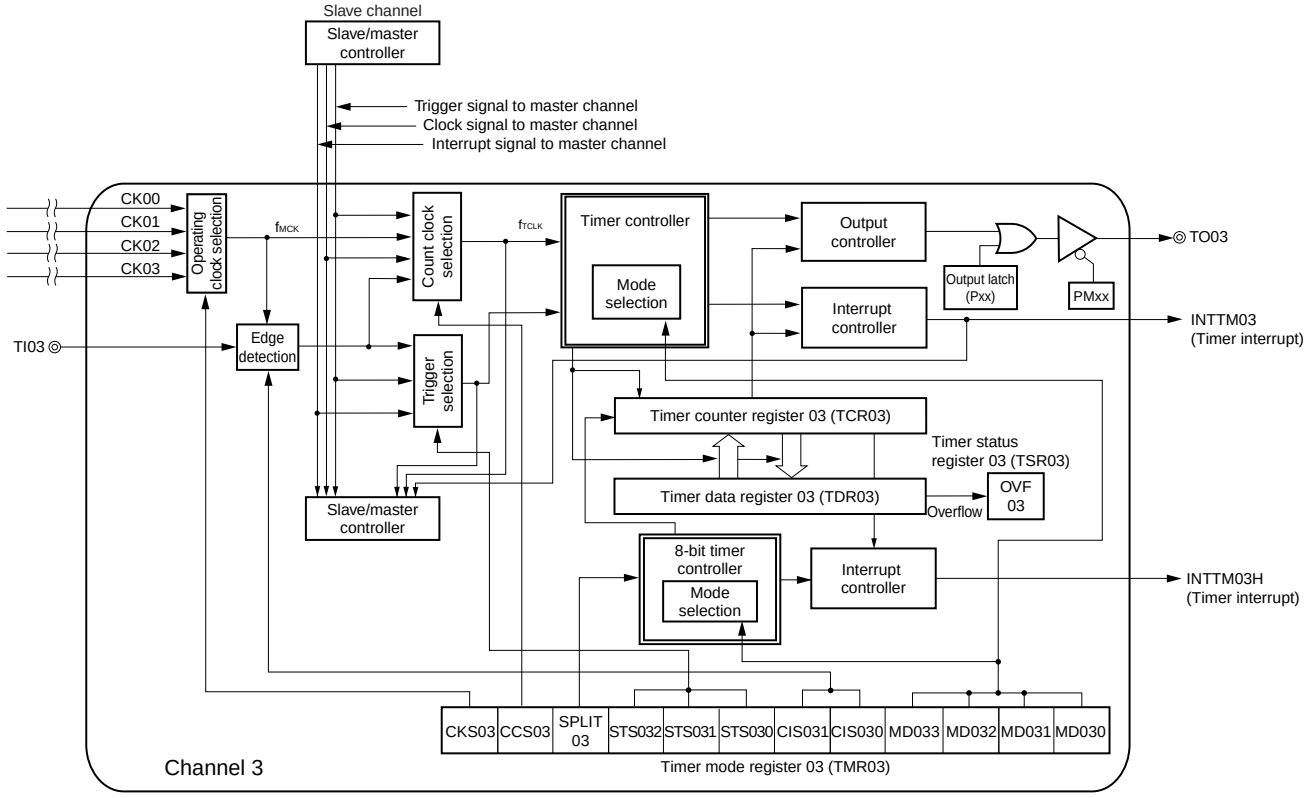


Figure 6-5. Internal Block Diagram of Channel 5 of Timer Array Unit

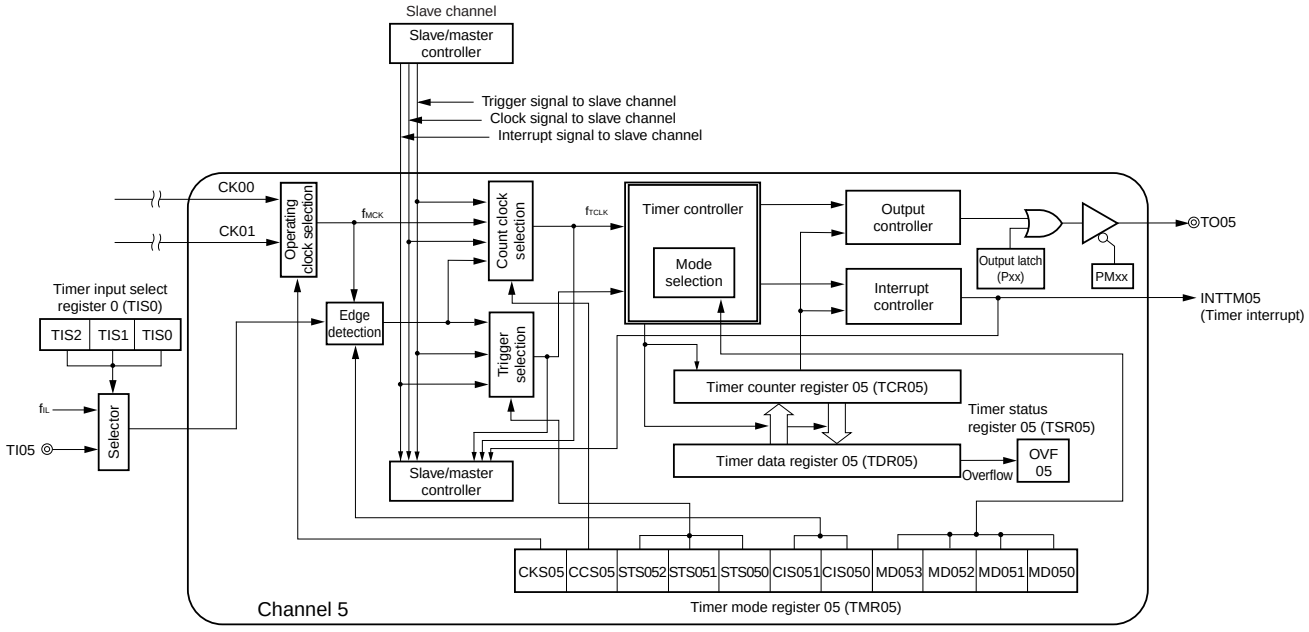
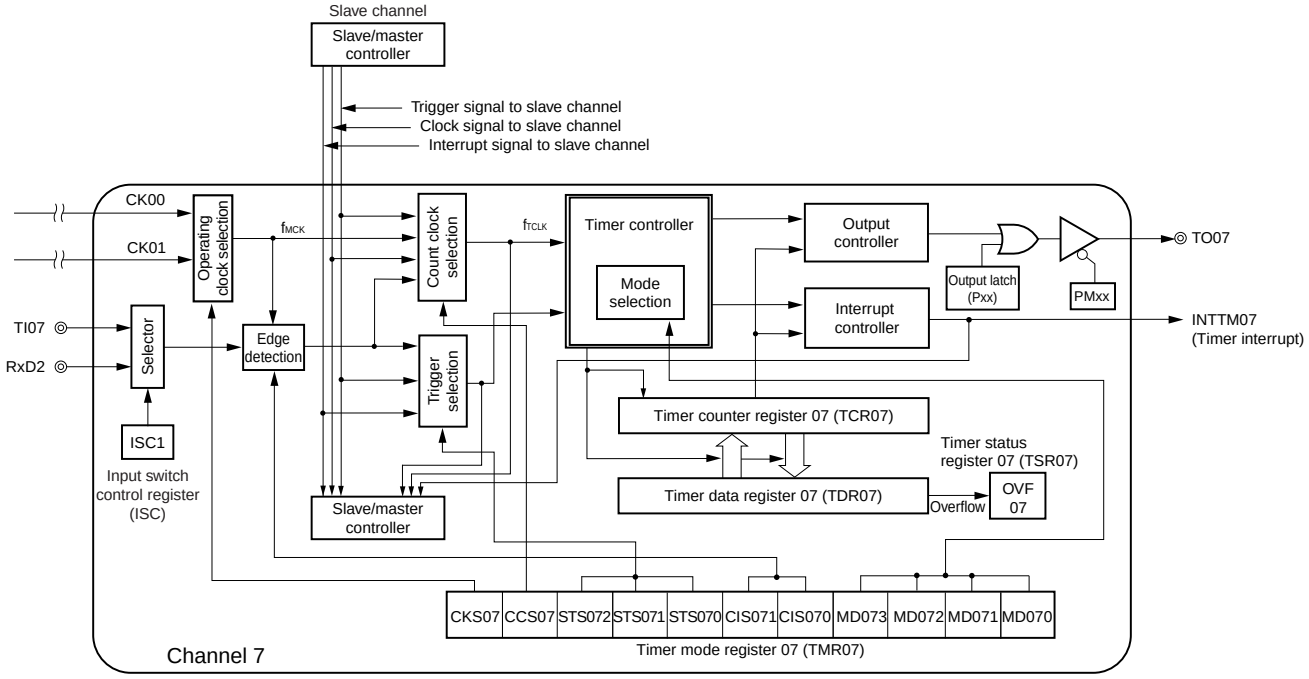


Figure 6-6. Internal Block Diagram of Channel 7 of Timer Array Unit



(1) Timer/counter register 0n (TCR0n)

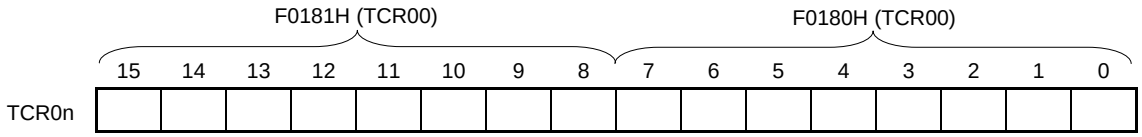
The TCR0n register is a 16-bit read-only register and is used to count clocks.

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 MD0n3 to MD0n0 bits of timer mode register 0n (TMR0n) (refer to **6.3 (3) Timer mode register 0n (TMR0n)**).

Figure 6-7. Format of Timer/Counter Register 0n (TCR0n)

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



Remark n: Channel number (n = 0 to 7)

The count value can be read by reading timer/counter register 0n (TCR0n).

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

- When the reset signal is generated
- When the TAU0EN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register 0n (TDR0n) even when the TCR0n register is read.

The TCR0n register read value differs as follows according to operation mode changes and the operating status.

Table 6-3. Timer/counter Register 0n (TCR0n) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer/counter register 0n (TCR0n) 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 TDR0n register + 1

Note This indicates the value read from the TCR0n register when channel n has stopped operating as a timer (TE0.n = 0) and has been enabled to operate as a counter (TS0.n = 1). The read value is held in the TCR0n register until the count operation starts.

Remark n: Channel number (n = 0 to 7)

(2) Timer data register 0n (TDR0n)

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 MD0n3 to MD0n0 bits of timer mode register 0n (TMR0n).

The value of the TDR0n register can be changed at any time.

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

In addition, for the TDR01 and TDR03 registers, while in the 8-bit timer mode (when the SPLIT bits of timer mode registers 01 and 03 (TMR01, TMR03) are 1), it is possible to rewrite the data in 8-bit units, with TDR01H and TDR03H used as the higher 8 bits, and TDR01L and TDR03L used as the lower 8 bits. However, reading is only possible in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 6-8. Format of Timer Data Register 0n (TDR0n) (n = 0, 2, 4 to 7)

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

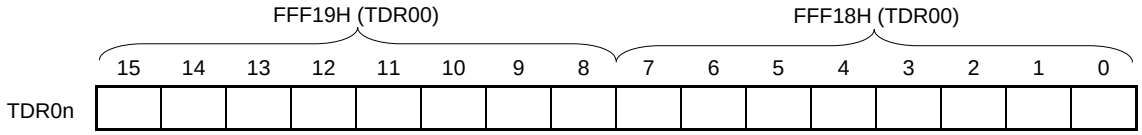
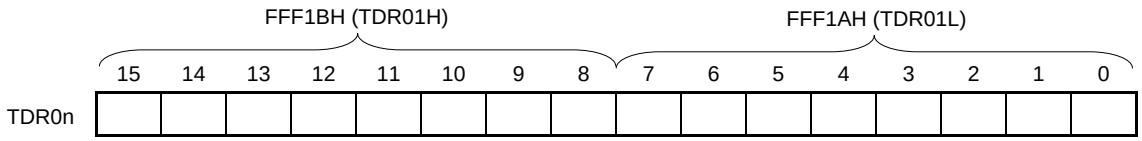


Figure 6-9. Format of Timer Data Register 0n (TDR0n) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF65H, FFF66H (TDR03) After reset: 0000H R/W



(i) When timer data register 0n (TDR0n) is used as compare register

Counting down is started from the value set to the TDR0n register. When the count value reaches 0000H, an interrupt signal (INTTM0n) is generated. The TDR0n register holds its value until it is rewritten.

Caution The TDR0n register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register 0n (TDR0n) is used as capture register

The count value of timer/counter register 0n (TCR0n) is captured to the TDR0n register when the capture trigger is input.

A valid edge of the TI0n pin can be selected as the capture trigger. This selection is made by timer mode register 0n (TMR0n).

Remark n: Channel number (n = 0 to 7)

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register 0 (TPS0)
- Timer mode register 0n (TMR0n)
- Timer status register 0n (TSR0n)
- Timer channel enable status register 0 (TE0)
- Timer channel start register 0 (TS0)
- Timer channel stop register 0 (TT0)
- Timer input select register 0 (TIS0)
- Timer output enable register 0 (TOE0)
- Timer output register 0 (TO0)
- Timer output level register 0 (TOL0)
- Timer output mode register 0 (TOM0)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode contorol register (PMCxx) **Note**
- Port mode register (PMxx) **Note**
- Port register (Pxx) **Note**

Note The port mode control registers (PMCxx), port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. For details, **see 6.3 (15) Port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4).**

Remark n: Channel number (n = 0 to 7)

(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 is used, be sure to set bit 0 (TAU0EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 6-10. 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	0	ADCEN	IICA0EN Note	SAU1EN Note	SAU0EN	0	TAU0EN

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

Note Those are not provided in the 20-pin products.

Cautions 1. When setting the timer array unit, be sure to set the TAU0EN bit to 1 first. If TAU0EN = 0, writing to a control register of timer array unit is ignored, and all read values are default values (except for the timer input select register 0 (TIS0), input switch control register (ISC), noise filter enable register 1 (NFEN1), port mode control register 0 (PMC0), port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4), and port registers 0, 1, 3, 4 (P0, P1, P3, P4)).

2. Be sure to clear the following bits to 0.
20-pin products: bits 1, 3, 4, 6
30, 32-pin products: bits 1, 6
48, 64-pin products: bits 1, 6

(2) Timer clock select register 0 (TPS0)

The TPS0 register is a 16-bit register that is used to select two or four types of operation clocks (CK00, CK01) that are commonly supplied to each channel from external prescaler. CK01 is selected by using bits 7 to 4 of the TPS0 register, and CK00 is selected by using bits 3 to 0.

In addition, for channel 1 and 3, CK02 is selected by using bits 9 and 8 of the TPS0 register, and CK03 is selected by using bits 13 and 12.

Rewriting of the TPS0 register during timer operation is possible only in the following cases.

If the PRS000 to PRS003 bits can be rewritten (n = 0 to 7):

All channels for which CK00 is selected as the operation clock (CKS0n1, CKS0n0 = 0, 0) are stopped (TE0.n = 0).

If the PRS010 to PRS013 bits can be rewritten (n = 0 to 7):

All channels for which CK01 is selected as the operation clock (CKS0n1, CKS0n0 = 0, 1) are stopped (TE0.n = 0).

If the PRS020 and PRS021 bits can be rewritten (n = 1, 3):

All channels for which CK02 is selected as the operation clock (CKS0n1, CKS0n0 = 1, 0) are stopped (TE0.n = 0).

If the PRS030 and PRS031 bits can be rewritten (n = 1, 3):

All channels for which CK03 is selected as the operation clock (CKS0n1, CKS0n0 = 1, 1) are stopped (TE0.n = 0).

The TPS0 register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-11. Format of Timer Clock Select register 0 (TPS0) (1/2)

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

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0	0	0	PRS 031	PRS 030	0	0	PRS 021	PRS 020	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 0k3	PRS 0k2	PRS 0k1	PRS 0k0	Selection of operation clock (CK0k) Note (k = 0, 1)					
					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	312.5 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	39.1 kHz	78.1 kHz	156.2 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	9.76 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz	31.25 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz	15.63 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 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	976 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TT0 = 00FFH).

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

2. If f_{CLK} (undivided) is selected as the operation clock (CK0k) and TDRn0 is set to 0000H (n = 0 or 1), interrupt requests output from timer array units are not detected.

Remarks 1. f_{CLK}: Operation clock frequency

2. Waveform of the clock to be selected in the TPS0 register which becomes high level for one period of f_{CLK} from its rising edge. For details, see **6.5.1 Count clock (f_{TCLK}).**

Figure 6-11. Format of Timer Clock Select register 0 (TPS0) (2/2)

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

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0	0	0	PRS 031	PRS 030	0	0	PRS 021	PRS 020	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 021	PRS 020	Selection of operation clock (CK02) Note					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	1	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
1	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 MHz	1.25 MHz	2 MHz
1	1	$f_{CLK}/2^6$	31.25 kHZ	78.1 kHz	156.2 kHz	312.5 kHz	500 kHz

PRS 031	PRS 030	Selection of operation clock (CK03) Note					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
0	1	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz	31.25 kHz
1	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	$f_{CLK}/2^{14}$	122 HZ	305 Hz	610 Hz	1.22 kHz	1.95 kHz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TT0 = 00FFH).
The timer array unit must also be stopped if the operating clock (f_{MCK}) specified by using the CKS0n0, and CKS0n1 bits or the valid edge of the signal input from the TI0n pin is selected as the count clock (f_{TCLK}).

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

By using channels 1 and 3 in the 8-bit timer mode and specifying CK02 or CK03 as the operation clock, the interval times shown in Table 6-4 can be achieved by using the interval timer function.

Table 6-4. Interval Times Available for Operation Clock CKS02 or CKS03

Clock		Interval time (f_{CLK} = 32 MHz)			
		10 μs	100 μs	1 ms	10 ms
CK02	$f_{CLK}/2$	√	–	–	–
	$f_{CLK}/2^2$	√	–	–	–
	$f_{CLK}/2^4$	√	√	–	–
	$f_{CLK}/2^6$	√	√	–	–
CK03	$f_{CLK}/2^8$	–	√	√	–
	$f_{CLK}/2^{10}$	–	√	√	–
	$f_{CLK}/2^{12}$	–	–	√	√
	$f_{CLK}/2^{14}$	–	–	√	√

Note The margin is within 5 %.

Remarks 1. f_{CLK}: Operation clock frequency
2. For details of asignal of $f_{CLK}/2^i$ selected with the TPS0 register, see **6.5.1 Count clock (f_{TCLK}).**

(3) Timer mode register 0n (TMR0n)

The TMR0n register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master/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 TMR0n register is prohibited when the register is in operation (when TE0.n = 1). However, bits 7 and 6 (CIS0n1, CIS0n0) can be rewritten even while the register is operating with some functions (when TE0.n = 1) (for details, see **6.7 Independent Channel Operation Function of Timer Array Unit** and **6.8 Simultaneous Channel Operation Function of Timer Array Unit**.

The TMR0n 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 TMR0n register.
TMR02, TMR04, TMR06: MASTER0n bit (n = 2, 4, 6)
TMR01, TMR03: SPLIT0n bit (n = 1, 3)
TMR00, TMR05, TMR07: Fixed to 0

Figure 6-12. Format of Timer Mode Register 0n (TMR0n) (1/4)

Address: F0190H, F0191H (TMR00) - F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 2, 4, 6)	CKS 0n1	CKS 0n0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 0, 5, 7)	CKS 0n1	CKS 0n0	0	CCS 0n	0 Note	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

CKS 0n1	CKS 0n0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CK00 set by timer clock select register 0 (TPS0)
0	1	Operation clock CK02 set by timer clock select register 0 (TPS0)
1	0	Operation clock CK01 set by timer clock select register 0 (TPS0)
1	1	Operation clock CK03 set by timer clock select register 0 (TPS0)

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 CCS0n bit.

The operation clocks CK02 and CK03 can only be selected for channels 1 and 3.

CCS 0n	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKS0n0 and CKS0n1 bits
1	Valid edge of input signal input from the TI0n pin

Count clock (f_{TCLK}) is used for the timer/counter, output controller, and interrupt controller.

Note Bit 11 is fixed at 0 of read only, write is ignored.

Cautions 1. Be sure to clear bits 13, 5, and 4 to "0".

2. The timer array unit must be stopped (TT0 = 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 CKS0n0 and CKS0n1 bits (f_{MCK}) or the valid edge of the signal input from the TI0n pin is selected as the count clock (f_{TCLK}).

Remark n: Channel number (n = 0 to 7)

Figure 6-12. Format of Timer Mode Register 0n (TMR0n) (2/4)

Address: F0190H, F0191H (TMR00), F0194H, F0195H (TMR02) After reset: 0000H R/W
F0198H, F0199H (TMR04) to F019EH, F019FH (TMR07)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 0, 2, 4 to 7)	CKS 0n1	CKS 0n0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Address: F0192H, F0193H (TMR01), F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

(Bit 11 of TMR0n (n = 0, 2, 4 to 7))

MASTER 0n	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 even channel can be set as a master channel (MASTER0n = 1). Be sure to use odd-numbered channels as slave channels (MASTER0n = 0). Clear the MASTER0n bit to 0 for a channel that is used with the independent channel operation function.	

(Bit 11 of TMR0n (n = 1, 3))

SPLIT0n	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 0n2	STS 0n1	STS 0n0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TI0n pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of the TI0n pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function).
Other than above			Setting prohibited

Remark n: Channel number (n = 0 to 7)

Figure 6-12. Format of Timer Mode Register 0n (TMR0n) (3/4)

Address: F0190H, F0191H (TMR00), F0194H, F0195H (TMR02) After reset: 0000H R/W
F0198H, F0199H (TMR04) to F019EH, F019FH (TMR07)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 0, 2, 4 to 7)	CKS 0n1	CKS 0n0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Address: F0192H, F0193H (TMR01), F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

CIS 0n1	CIS 0n0	Selection of TI0n pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS0n2 to STS0n0 bits is other than 010B, set the CIS0n1 to CIS0n0 bits to 10B.		

MD 0n3	MD 0n2	MD 0n1	MD 0n0	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	1/0	Interval timer mode	Interval timer/Square wave output/Divider function/PWM output (master)	Counting down
0	1	0	1/0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	0	Event counter mode	External event counter	Counting down
1	0	0	1/0	One-count mode	Delay counter/One-shot pulse output/PWM output (slave)	Counting down
1	1	0	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than above				Setting prohibited		
The operation of the MD0n0 bit varies depending on each operation mode (see table below).						

Remark n: Channel number (n = 0 to 7)

Figure 6-12. Format of Timer Mode Register 0n (TMR0n) (4/4)

Address: F0190H, F0191H (TMR00), F0194H, F0195H (TMR02) After reset: 0000H R/W
F0198H, F0199H (TMR04) to F019EH, F019FH (TMR07)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 0, 2, 4 to 7)	CKS 0n1	CKS 0n0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Address: F0192H, F0193H (TMR01), F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Operation mode (Value set by the MD0n3 to MD0n1 bits (see table above))	MD 0n0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• Capture mode (0, 1, 0)	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 1 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
	1	Start trigger is valid during counting operation Note 2. At that time, interrupt is also generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated, either.
Other than above		Setting prohibited

Notes 1. In one-count mode, interrupt output (INTTM0n) when starting a count operation and TO0n output are not controlled.

2. If the start trigger (TS0.n = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

Remark n: Channel number (n = 0 to 7)

(4) Timer status register 0n (TSR0n)

The TSR0n register indicates the overflow status of the counter of channel n.

The TSR0n register is valid only in the capture mode (MD0n3 to MD0n1 = 010B) and capture & one-count mode (MD0n3 to MD0n1 = 110B). It will not be set in any other mode. See Table 6-5 for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSR0n register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSR0n register can be set with an 8-bit memory manipulation instruction with TSR0nL.

Reset signal generation clears this register to 0000H.

Figure 6-13. Format of Timer Status Register 0n (TSR0n)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSR0n	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 n: Channel number (n = 0 to 7)

Table 6-5. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF bit	Set/clear conditions
• Capture mode	clear	When no overflow has occurred upon capturing
• Capture & one-count mode	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	–
	set	(Use prohibited)

Remark The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

(5) Timer channel enable status register 0 (TE0)

The TE0 register is used to enable or stop the timer operation of each channel.

Each bit of the TE0 register corresponds to each bit of the timer channel start register 0 (TS0) and the timer channel stop register 0 (TT0). When a bit of the TS0 register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TT0 register is set to 1, the corresponding bit of this register is cleared to 0.

The TE0 register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TE0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TE0L.

Reset signal generation clears this register to 0000H.

Figure 6-14. Format of Timer Channel Enable Status register 0 (TE0)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	TEH03	0	TEH01	0	TE0.7	TE0.6	TE0.5	TE0.4	TE0.3	TE0.2	TE0.1	TE0.0

TEH03	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.

TEH01	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.

TE0.n	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 TE0.1 and TE0.3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.	

Remark n: Channel number (n = 0 to 7)

(6) Timer channel start register 0 (TS0)

The TS0 register is a trigger register that is used to clear timer/counter register 0n (TCR0n) and start the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register 0 (TE0) is set to 1. The TS0.n, TSH01, TSH03 bits are immediately cleared when operation is enabled (TE0.n, TEH01, TEH03 = 1), because they are trigger bits.

The TS0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TS0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TS0L.

Reset signal generation clears this register to 0000H.

Figure 6-15. Format of Timer Channel Start register 0 (TS0)

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	TSH03	0	TSH01	0	TS0.7	TS0.6	TS0.5	TS0.4	TS0.3	TS0.2	TS0.1	TS0.0

TSH03	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 TEH03 bit is set to 1 and the count operation becomes enabled. The TCR03 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-6** in **6.5.2 Start timing of counter**).

TSH01	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 TEH01 bit is set to 1 and the count operation becomes enabled. The TCR01 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-6** in **6.5.2 Start timing of counter**).

TS0.n	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TE0.n bit is set to 1 and the count operation becomes enabled. The TCR0n register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6-6** in **6.5.2 Start timing of counter**). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TS0.1 and TS0.3 when channel 1 or 3 is in the 8-bit timer mode.

Cautions 1. Be sure to clear bits 15 to 12, 10, 8 to "0"

2. When switching from a function that does not use TI0n pin input to one that does, the following wait period is required from when timer mode register 0n (TMR0n) is set until the TS0.n (TSH01, TSH03) bit is set to 1.

When the TI0n pin noise filter is enabled (TNFEN = 1): Four cycles of the operation clock (f_{MCK})

When the TI0n pin noise filter is disabled (TNFEN = 0): Two cycles of the operation clock (f_{MCK})

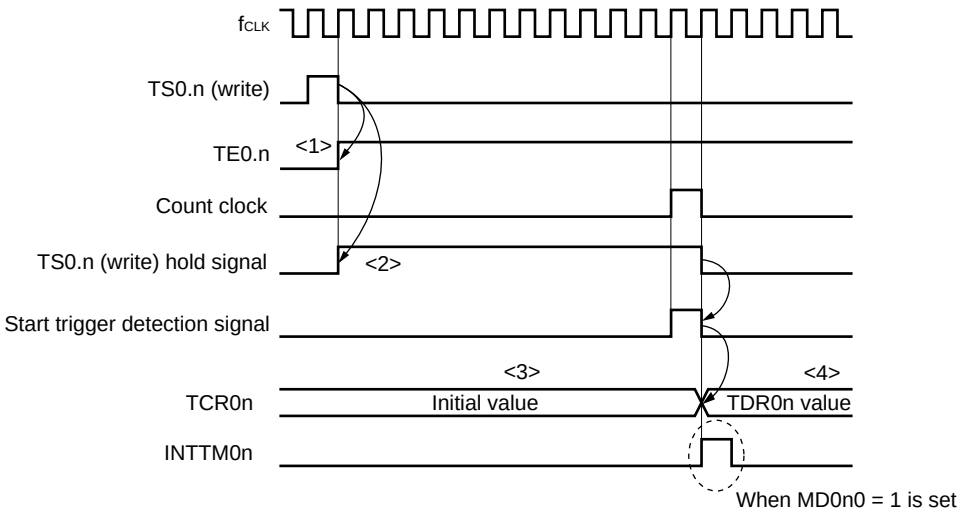
Remarks 1. When the TS0 register is read, 0 is always read.

2. n: Channel number (n = 0 to 7)

(a) Start timing in interval timer mode

<1> Operation is enabled (TE0.n = 1) by writing 1 to the TS0.n bit.
<2> The write data to the TS0.n bit is held until count clock generation.
<3> Timer/counter register 0n (TCR0n) holds the initial value until count clock generation.
<4> On generation of count clock, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and count starts.

Figure 6-16. Start Timing (In Interval Timer Mode)

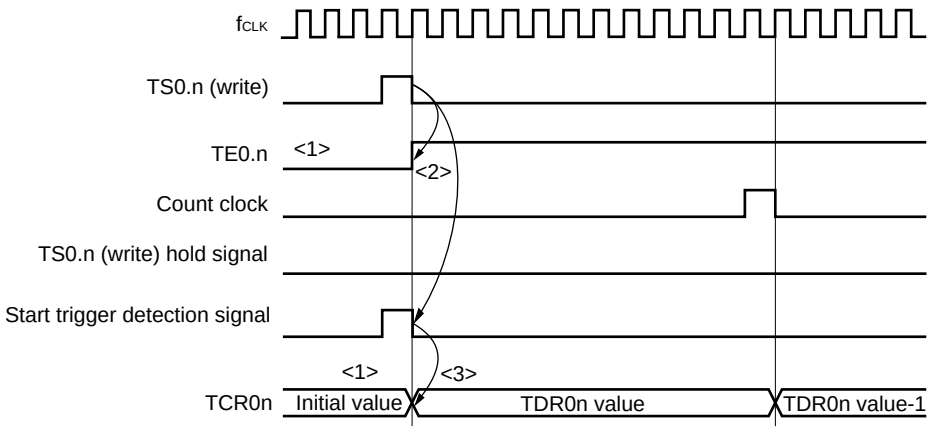


Caution In the first cycle operation of count clock after writing the TS0.n bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0 = 1.

(b) Start timing in event counter mode

<1> Timer/counter register 0n (TCR0n) holds its initial value while operation is stopped (TE0.n = 0).
<2> Operation is enabled (TE0.n = 1) by writing 1 to the TS0.n bit.
<3> As soon as 1 has been written to the TS0.n bit and 1 has been set to the TE0.n bit, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register to start counting.
<4> After that, the TCR0n register value is counted down according to the count clock.

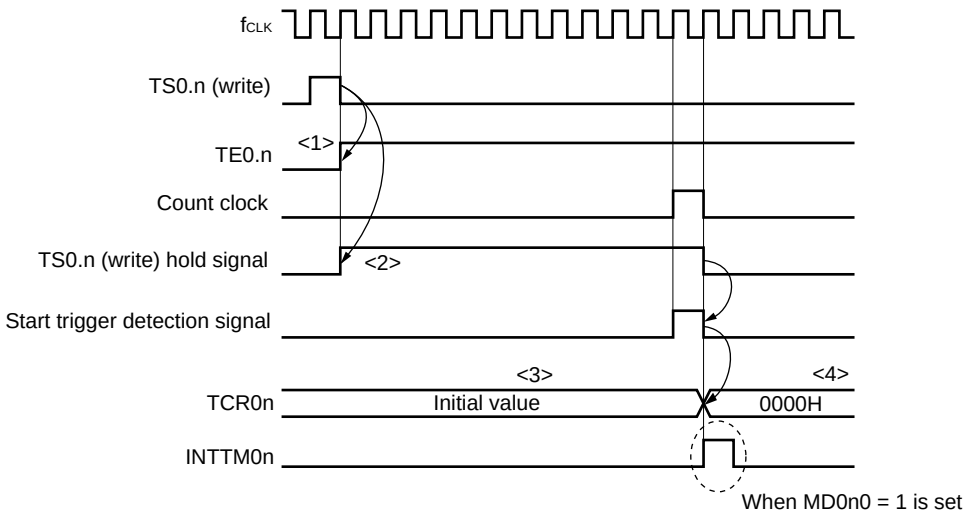
Figure 6-17. Start Timing (In Event Counter Mode)



(c) Start timing in capture mode

<1> Operation is enabled (TE0.n = 1) by writing 1 to the TS0.n bit.
<2> The write data to the TS0.n bit is held until count clock generation.
<3> Timer/counter register 0n (TCR0n) holds the initial value until count clock generation.
<4> On generation of count clock, 0000H is loaded to the TCR0n register and count starts.

Figure 6-18. Start Timing (In Capture Mode)



Caution In the first cycle operation of count clock after writing the TS0.n bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0 = 1.

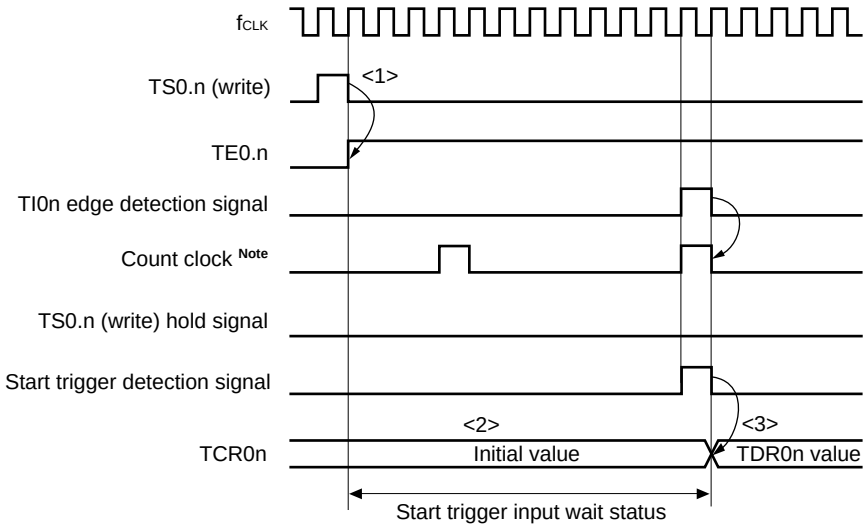
(d) Start timing in one-count mode

<1> Operation is enabled (TE0.n = 1) by writing 1 to the TS0.n bit.

<2> Enters the start trigger input wait status, and timer/counter register 0n (TCR0n) holds the initial value.

<3> On start trigger detection, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and count starts.

Figure 6-19. Start Timing (In One-count Mode)



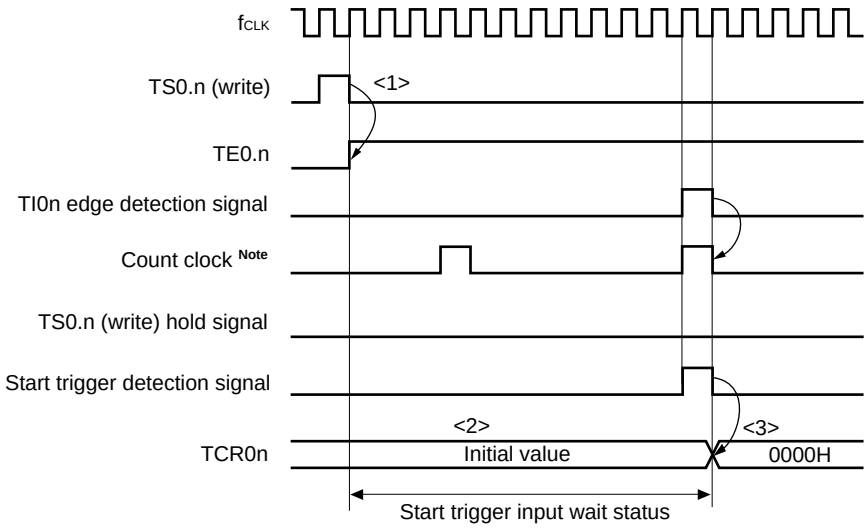
Note When the one-count mode is set, the operation clock (f_{MCK}) is selected as count clock (CCS0n = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (If the TI0n pin input signal is used as a start trigger, an error of one count clock occurs.).

(e) Start timing in capture & one-count mode

<1> Operation is enabled (TE0.n = 1) by writing 1 to the bit n for this register 0 (TS0.n).
<2> Enters the start trigger input wait status, and timer/counter register 0n (TCR0n) holds the initial value.
<3> On start trigger detection, 0000H is loaded to the TCR0n register and count starts.

Figure 6-20. Start Timing (In Capture & One-count Mode)



Note When the capture & one-count mode is set, the operation clock (f_{MCK}) is selected as count clock (CCS0n = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (If the TI0n pin input signal is used as a start trigger, an error of one count clock occurs.)

(7) Timer channel stop register 0 (TT0)

The TT0 register is a trigger register that is used to stop the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register 0 (TE0) is cleared to 0. The TT0.n, TTH01, TTH03 bits are immediately cleared when operation is stopped (TE0.n, TTH01, TTH03 = 0), because they are trigger bits.

The TT0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TT0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TT0L.

Reset signal generation clears this register to 0000H.

Figure 6-21. Format of Timer Channel Stop register 0 (TT0)

Address: F01B4H, F01B5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	TTH03	0	TTH01	0	TT0.7	TT0.6	TT0.5	TT0.4	TT0.3	TT0.2	TT0.1	TT0.0

TTH03	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	Operation is stopped (stop trigger is generated).

TTH01	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	Operation is stopped (stop trigger is generated).

TT0.n	Operation stop trigger of channel n
0	No trigger operation
1	Operation is stopped (stop trigger is generated). This bit is the trigger to stop operation of the lower 8-bit timer for TT0.1 and TT0.3 when channel 1 or 3 is in the 8-bit timer mode.

Caution Be sure to clear bits 15 to 12, 10, 8 of the TT0 register to "0".

Remarks 1. When the TT0 register is read, 0 is always read.

2. n: Channel number (n = 0 to 7)

(8) Timer input select register 0 (TIS0)

The TIS0 register is used to select the channel 5 timer input..

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

Reset signal generation clears this register to 00H.

Figure 6-22. Format of Timer Input Select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS0.2	TIS0.1	TIS0.0

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	
0	1	0	
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

Caution High-level width, low-level width of timer input is selected, will require more than $1/f_{MCK}$ +10 ns. Therefore, when selecting f_{SUB} to f_{CLK} (CSS bit of CKS register = 1), can not TIS02 bit set to 1.

(9) Timer output enable register 0 (TOE0)

The TOE0 register is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TO0.n bit of timer output register 0 (TO0) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TO0n).

The TOE0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOE0 register can be set with a 1-bit or 8-bit memory manipulation instruction with TOE0L.

Reset signal generation clears this register to 0000H.

Figure 6-23. Format of Timer Output Enable register 0 (TOE0)

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
TOE0	0	0	0	0	0	0	0	0	TOE07	TOE06	TOE05	TOE04	TOE03	TOE02	TOE01	TOE00

TOE0n	Timer output enable/disable of channel n
0	Diseble output of timer. Without reflecting on TOmn bit timer operation, to fixed the output. Writing to the TOmn bit is enabled.
1	Enable output of timer. Reflected in the TOmn bit timer operation, to generate the output waveform. Writing to the TOmn bit is disabled (writing is ignored).

Caution Be sure to clear bits 15 to 8 to “0”.

Remark n: Channel number (n = 0 to 7)

(10) Timer output register 0 (TO0)

The TO0 register is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TO0n) of each channel.

The TO0.n bit oh this register can be rewritten by software only when timer output is disabled (TOE0n = 0). When timer output is enabled (TOE0.n = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P01/TO00, P16/TI01/TO01, P17/TI02/TO02 (P15[Note]), P31/TI03/TO03 (P14[Note]), P42/TI04/TO04 (P13[Note]),

P05/TI05/TO05 (P12[Note]), P06/TI06/TO06 (P11[Note]), or P41/TI07/TO07 (P10[Note]) pin as a port function pin, set the corresponding TO0.n bit to “0”.

The TO0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TO0 register can be set with an 8-bit memory manipulation instruction with TO0L.

Reset signal generation clears this register to 0000H.

Figure 6-24. Format of Timer Output register 0 (TO0)

Address: F01B8H, F01B9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	TO0.7	TO0.6	TO0.5	TO0.4	TO0.3	TO0.2	TO0.1	TO0.0

TO0.n	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution Be sure to clear bits 15 to 8 to “0”.

Note “(P1x)” indicates an alternate port when the bit 0 of the peripheral I/O redirection register (PIOR) is set to “1”.

Remark n: Channel number (n = 0 to 7)

(11) Timer output level register 0 (TOL0)

The TOL0 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 (TOE0.n = 1) in the Slave channel output mode (TOM0.n = 1). In the master channel output mode (TOM0.n = 0), this register setting is invalid.

The TOL0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOL0 register can be set with an 8-bit memory manipulation instruction with TOL0L.

Reset signal generation clears this register to 0000H.

Figure 6-25. Format of Timer Output Level register 0 (TOL0)

Address: F01BCH, F01BDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	0	0	TOL 0.7	TOL 0.6	TOL 0.5	TOL 0.4	TOL 0.3	TOL 0.2	TOL 0.1	0

TOL 0.n	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Inverted output (active-low)

Caution Be sure to clear bits 15 to 8, and 0 to "0".

Remarks 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.

2. n: Channel number (n = 0 to 7)

(12) Timer output mode register 0 (TOM0)

The TOM0 register is used to control the timer output mode of each channel.

When a channel is used for the independent channel operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the simultaneous channel operation function (PWM output, one-shot pulse output, or multiple PWM output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOE0.n = 1).

The TOM0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOM0 register can be set with an 8-bit memory manipulation instruction with TOM0L.

Reset signal generation clears this register to 0000H.

Figure 6-26. Format of Timer Output Mode register 0 (TOM0)

Address: F01BEH, F01BFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	TOM 0.7	TOM 0.6	TOM 0.5	TOM 0.4	TOM 0.3	TOM 0.2	TOM 0.1	0

TOM 0.n	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTM0n))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTM0n) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Caution Be sure to clear bits 15 to 8, and 0 to "0".

Remark n: Channel number

n = 0 to 7 (n = 0, 2, 4, 6 for master channel)

p: Slave channel number

$n < p \leq 7$

(For details of the relation between the master channel and slave channel, refer to **6.4.1 Basic Rules of Simultaneous Channel Operation Function**.)

(13) Input switch control register (ISC)

The ISC.1 and ISC.0 bits of the ISC register are used to implement LIN-bus communication operation by using channel 7 in association with the serial array unit. When the ISC.1 bit is set to 1, the input signal of the serial data input pin (RxD2) is selected as a timer input signal.

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

Reset signal generation clears this register to 00H.

Figure 6-27. Format of Input Switch Control Register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC.1	ISC.0

ISC.1	Switching channel 7 input of timer array unit
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of the RxD2 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).

ISC.0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD2 pin as an external interrupt (wakeup signal detection).

Caution Be sure to clear bits 7 to 2 to "0".

Remark When the LIN-bus communication function is used, select the input signal of the RxD2 pin by setting ISC.1 to 1.

(14) 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 before the start of operation by setting the corresponding bits to 1 on the pins in need of noise removal.

When the noise filter is ON, match detection and synchronization of the 2 clocks is performed with the operation clock (f_{MCK}). When the noise filter is OFF, only synchronization is performed with the operation clock (f_{MCK}).

The NFEN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note For details, see **6.5.1 (2) When valid edge of input signal via the TI0n pin is selected (CCS0n = 1)** and **6.5.2 Start timing of counter**.

Figure 6-28. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	Enable/disable using noise filter of TI07/TO07/P41 (P10) pin input signal[Note]
0	Noise filter OFF
1	Noise filter ON

TNFEN06	Enable/disable using noise filter of TI06/TO06/P06 (P11) pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05/TO05/P05 (P12) pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04/TO04/P42 (P13) pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03/TO03/P31 (P14) pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02/TO02/P17 (P15) pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01/P01/P16 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00/P00 pin input signal
0	Noise filter OFF
1	Noise filter ON

Notes 1. The applicable pin can be switched by setting the ISC.1 bit of the ISC register.

ISC.1 = 0: Whether or not to use the noise filter of the TI07 pin can be selected.

ISC.1 = 1: Whether or not to use the noise filter of the RxD2 pin can be selected.

2. “(P1x)” indicates a dedicated port when the bit 0 of the peripheral I/O redirection register (PIOR) is set to “1”.

Remark The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

(15) Port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4)

These registers set input/output of ports 0, 1, 3, 4 in 1-bit units.

The presence or absence of timer I/O pins depends on the product. When using the timer array unit, set the following port mode registers according to the product used.

20, 30, and 32-pin products: PM0, PM1, PM3
48, 64-pin products: PM0, PM1, PM3, PM4

When using the ports (such as P01/TO00 and P17/TO02/TI02) to be shared with the timer output pin for timer output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example: When using P17/TO02/TI02 for timer output
Set the PM1.7 bit of port mode register 1 to 0.
Set the P1.7 bit of port register 1 to 0.

When using the ports (such as P00/TI00 and P17/TO02/TI02) to be shared with the timer output pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P17/TO02/TI02 for timer input
Set the PM1.7 bit of port mode register 1 to 1.
Set the P1.7 bit of port register 1 to 0 or 1.

The PM0, PM1, PM3, PM4 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark In the 20- to 30-pin products, TI00 (P00) and TO00 (P01) pins alternate analog input pins. So, the PMC0 register should be set. See 4.3 (6) Port mode control registers (PMC0, PMC12, PMC14) for details.

Figure 6-29. Format of Port Mode Registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4) (64-pin products)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	PM0.6	PM0.5	PM0.4	PM0.3	PM0.2	PM0.1	PM0.0

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM1.7	PM1.6	PM1.5	PM1.4	PM1.3	PM1.2	PM1.1	PM1.0

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM3.1	PM3.0

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	PM4.3	PM4.2	PM4.1	PM4.0

PMm.n	Pmn pin I/O mode selection (m = 0, 1, 3, 4; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark The figure shown above presents the format of port mode registers 0, 1, 3, and 4 of the 64-pin products. The format of the port mode register of other products, see **4.3 (1) Port mode registers (PMxx)**.

6.4 Basic Rules of Simultaneous Channel Operation Function

6.4.1 Basic Rules of Simultaneous Channel Operation Function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

(1) Only an even channel (channel 0, 2, 4, etc.) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel.
(3) The slave channel must be lower than the master channel.
Example: If channel 2 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example: If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

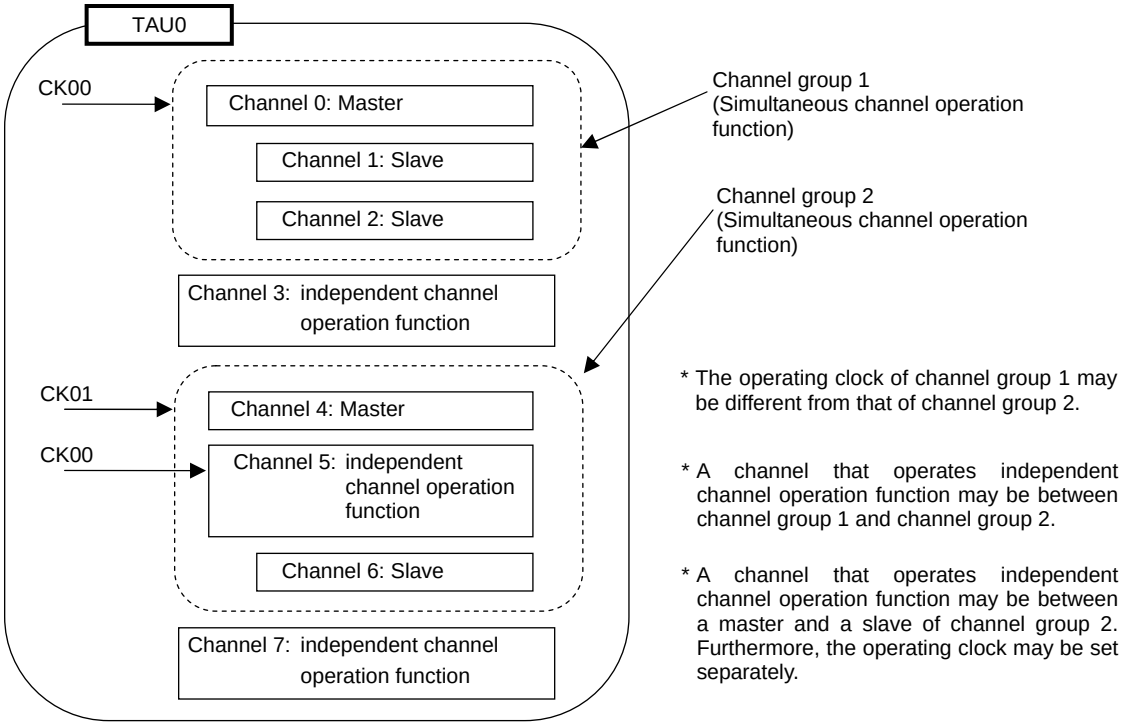
(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS0n0, CKS0n1 bits (bit 15, 14 of timer mode register 0n (TMR0n)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit INTTM0n (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTM0n (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTM0n (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTM0n (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TS0.n) of the channels in combination must be set at the same time.
(11) To stop the channels in combination simultaneously, the channel stop trigger bit (TT0.n) of the channels in combination must be set at the same time.
(12) During the counting operation, a TS0n bit of a master channel or TS0n bits of all channels which are operating simultaneously can be set. It cannot be applied to TS0n bits of slave channels alone.
(13) CK02/CK03 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.
(14) Timer mode register m0 (TMRm0) has no master bit (it is fixed as “0”). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in **6.4.1 Basic Rules of Simultaneous Channel Operation Function** do not apply to the channel groups.

Remark n: Channel number (n = 0 to 7)

Figure 6-30. Basic Rules of Simultaneous Channel Operation Function



* The operating clock of channel group 1 may be different from that of channel group 2.

* A channel that operates independent channel operation function may be between channel group 1 and channel group 2.

* A channel that operates independent channel operation function may be between a master and a slave of channel group 2. Furthermore, the operating clock may be set separately.

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

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

(1) The 8-bit timer operation function applies only to channels 1 and 3.
(2) When using 8-bit timers, set the SPLIT bit of timer mode register 0n (TMR0n) to 1.
(3) The higher 8 bits can be operated as the interval timer function.
(4) At the start of operation, the higher 8 bits output INTTM01H/INTTM03H (an interrupt) (which is the same operation performed when MD0n0 is set to 1).
(5) The operation clock of the higher 8 bits is selected according to the CKS0n1 and CKS0n0 bits of the lower-bit TMR0n register.
(6) For the higher 8 bits, the TSH01/TSH03 bit is manipulated to start channel operation and the TTH01/TTH03 bit is manipulated to stop channel operation. The channel status can be checked using the TEH01/TEH03 bit.
(7) The lower 8 bits operate according to the TMR0n register settings. The following three functions support operation of the lower 8 bits:
- Interval timer function
- External event counter function
- Delay count function

(8) For the lower 8 bits, the TS0.1/TS0.3 bit is manipulated to start channel operation and the TT0.1/TT0.3 bit is manipulated to stop channel operation. The channel status can be checked using the TE0.1/TE0.3 bit.
(9) During 16-bit operation, manipulating the TSH01, TSH03, TTH01, and TTH03 bits is invalid. The TS0.1, TS0.3, TT0.1, and TT0.3 bits are manipulated to operate channels 1 and 3. The TEH03 and TEH01 bits are not changed.
(10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM) cannot be used.

Remark n: Channel number (n = 1, 3)

6.5 Operation of Counter

6.5.1 Count clock (f_{TCLK})

The count clock (f_{TCLK}) of the timer array unit can be selected between following by CCS0n bit of timer mode register 0n (TMR0n).

- Operation clock (f_{MCK}) specified by the CKS0n0 and CKS0n1 bits
- Valid edge of input signal input from the TI0n pin

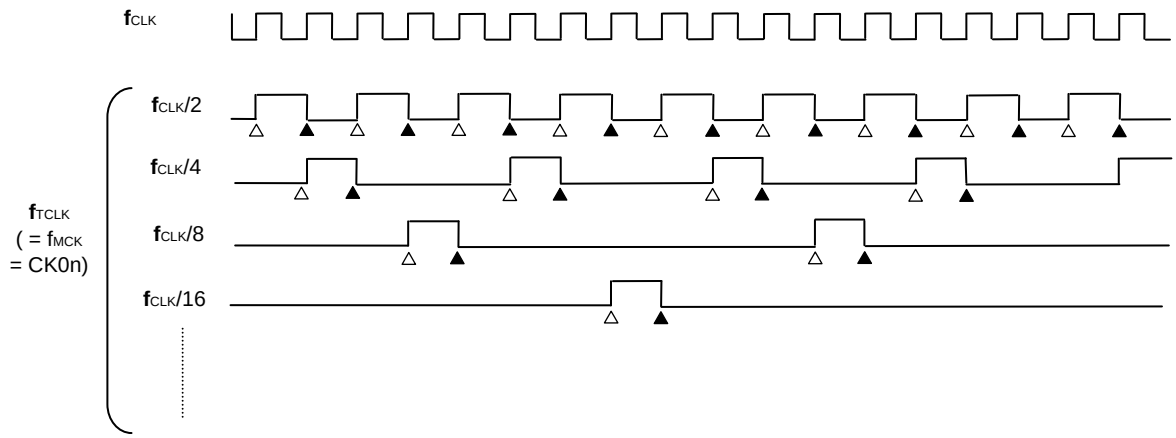
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the CKS0n0 and CKS0n1 bits is selected (CCS0n = 0)

The count clock (f_{TCLK}) is between f_{CLK} to $f_{CLK}/2^{15}$ by setting of timer clock select register 0 (TPS0). When a divided f_{CLK} is selected, however, the clock selected in TPS0n register, but a signal which becomes high level for one period of f_{CLK} from its rising edge. When a f_{CLK} is selected, fixed to high level

Counting of timer count register 0n (TCR0n) 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 rising edge of the count clock”, as a matter of convenience.

Figure 6-31. Timing of f_{CLK} and count clock (f_{TCLK}) (When CCS0n = 0)



Remarks 1. △ : Rising edge of the count clock

▲ : Synchronization, increment/decrement of counter

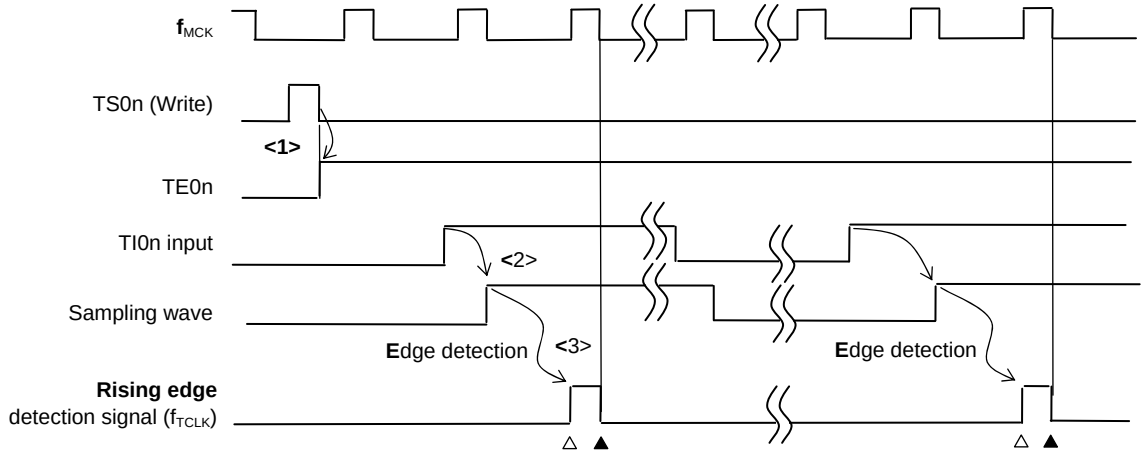
2. f_{CLK}: Operation clock

(2) When valid edge of input signal via the TI0n pin is selected (CCS0n = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TI0n pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TI0n pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer count register 0n (TCR0n) 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 TI0n pin”, as a matter of convenience.

Figure 6-32. Timing of f_{CLK} and count clock (f_{TCLK}) (When CCS0n = 1, noise filter unused)



<1> Setting TS0n bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TI0n pin.

<2> The rise of input signal via the TI0n pin is sampled by f_{MCK}.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remarks 1. △ : Rising edge of the count clock

▲ : Synchronization, increment/decrement of counter

2. f_{CLK}: Operation clock

f_{MCK}: Operation clock of channel n

3. The waveform of the input signal via TI0n pin of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter, the one-shot pulse output are the same as that shown in Figure 6-22.

6.5.2 Start timing of counter

Timer count register 0n (TCR0n) becomes enabled to operation by setting of TS0n bit of timer channel start register 0 (TS0).

Operations from count operation enabled state to timer count Register 0n (TCR0n) count start is shown in Table 6-6.

Table 6-6. Operations from Count Operation Enabled State to Timer/counter Register 0n (TCR0n) Count Start

Timer operation mode	Operation when TS0n = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TS0n = 1) until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see **6.5.3 (1) Operation of interval timer mode**).
• Event counter mode	Writing 1 to the TS0n bit loads the value of the TDR0n register to the TCR0n register. If detect edge of TI0n input. The subsequent count clock performs count down operation (see **6.5.3 (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 TCR0n register and the subsequent count clock performs count up operation (see **6.5.3 (3) Operation of capture mode (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0n bit while the timer is stopped (TE0n = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see **6.5.3 (4) Operation of one-count mode**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0n bit while the timer is stopped (TE0n = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCR0n register and the subsequent count clock performs count up operation (see **6.5.3 (5) Operation of capture & one-count mode (high-level width measurement**).

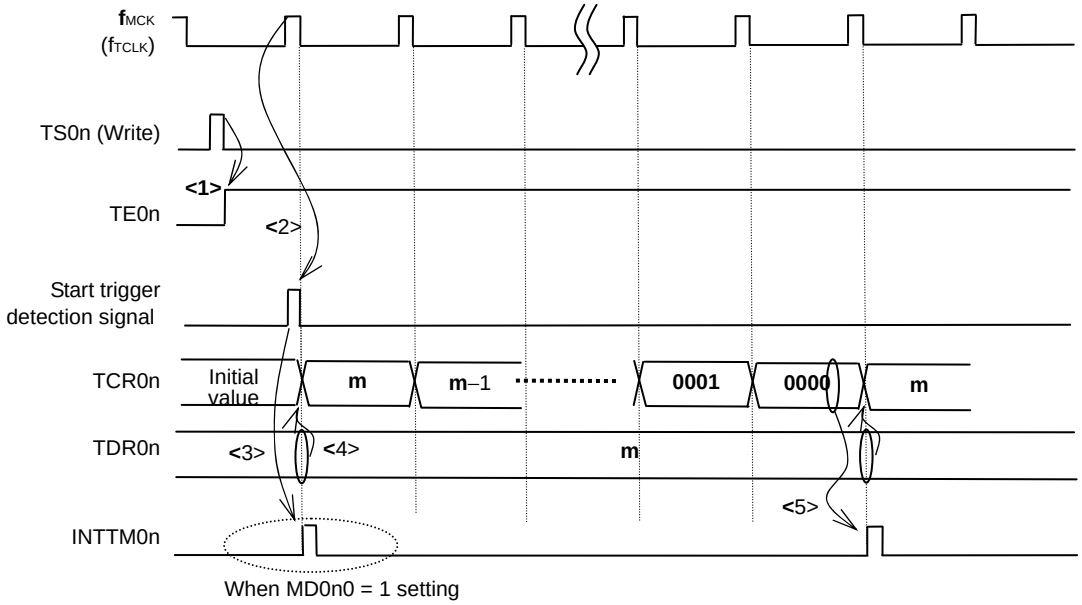
6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit. Timer count register 0n (TCR0n) holds the initial value until count clock generation.
<2> A start trigger is generated at the first count clock after operation is enabled.
<3> When the MD0n0 bit is set to 1, INTTM0n is generated by the start trigger.
<4> By the first count clock after the operation enable, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting starts in the interval timer mode.
<5> When the TCR0n register counts down and its count value is 0000H, INTTM0n is generated and the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting keeps on.

Figure 6-33 Operation Timing (In Interval Timer Mode)



Caution In the first cycle operation of count clock after writing the TS0n 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 MD0n0 = 1.

Remark f_{MCK}, the start trigger detection signal, and INTTM0n become active between one clock in synchronization with f_{CLK}.

(2) Operation of event counter mode

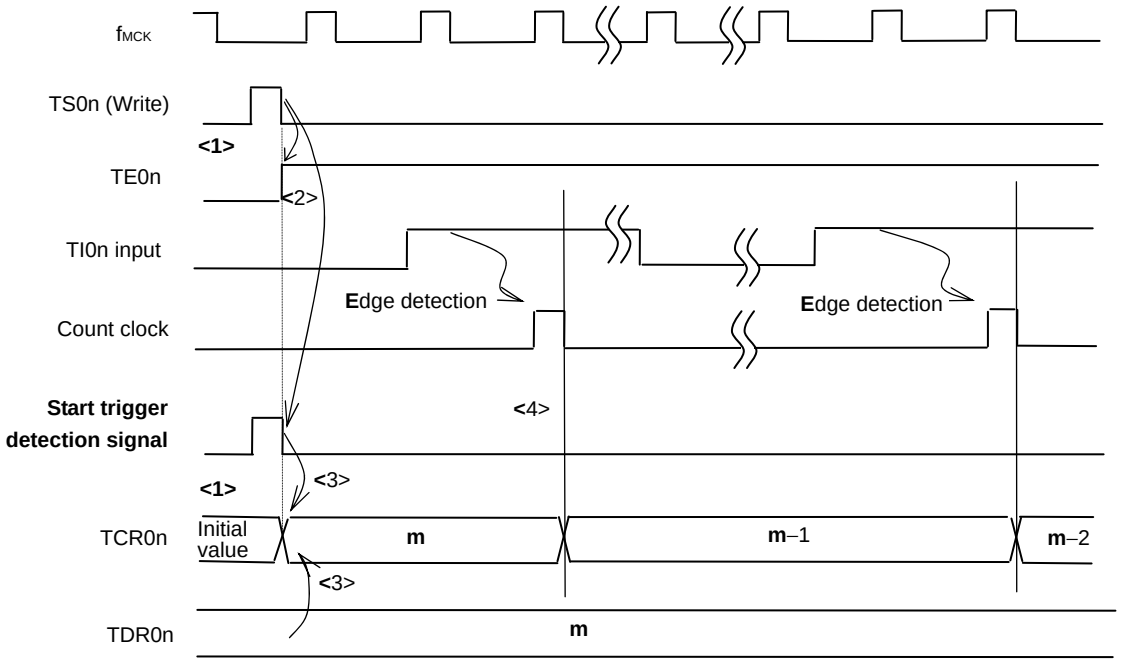
<1> Timer count register 0n (TCR0n) holds its initial value while operation is stopped (TE0n = 0).

<2> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.

<3> As soon as 1 has been written to the TS0n bit and 1 has been set to the TE0n bit, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register to start counting.

<4> After that, the TCR0n register value is counted down according to the count clock of the valid edge of the TI0n input .

Figure 6-34 Operation Timing (In Event Counter Mode)

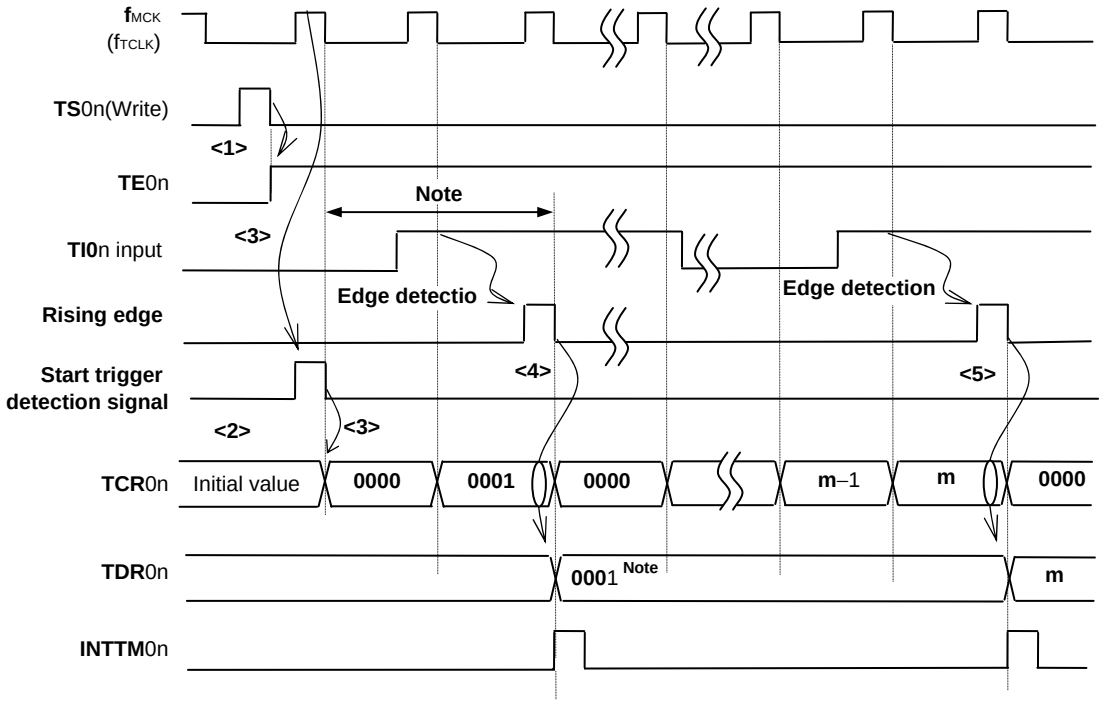


Remark The timing is shown in Figure 6-34 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input.

(3) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<2> Timer count register 0n (TCR0n) holds the initial value until count clock generation.
<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCR0n register and counting starts in the capture mode. (When the MD0n0 bit is set to 1, INTTM0n is generated by the start trigger.)
<4> On detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated. However, this capture value is nomeaning. The TCR0n register keeps on counting from 0000H.
<5> On next detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated.

Figure 6-35 Operation Timing (In Capture Mode : Input Pulse Interval Measurement)



Note If a clock has been input to TI0n (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

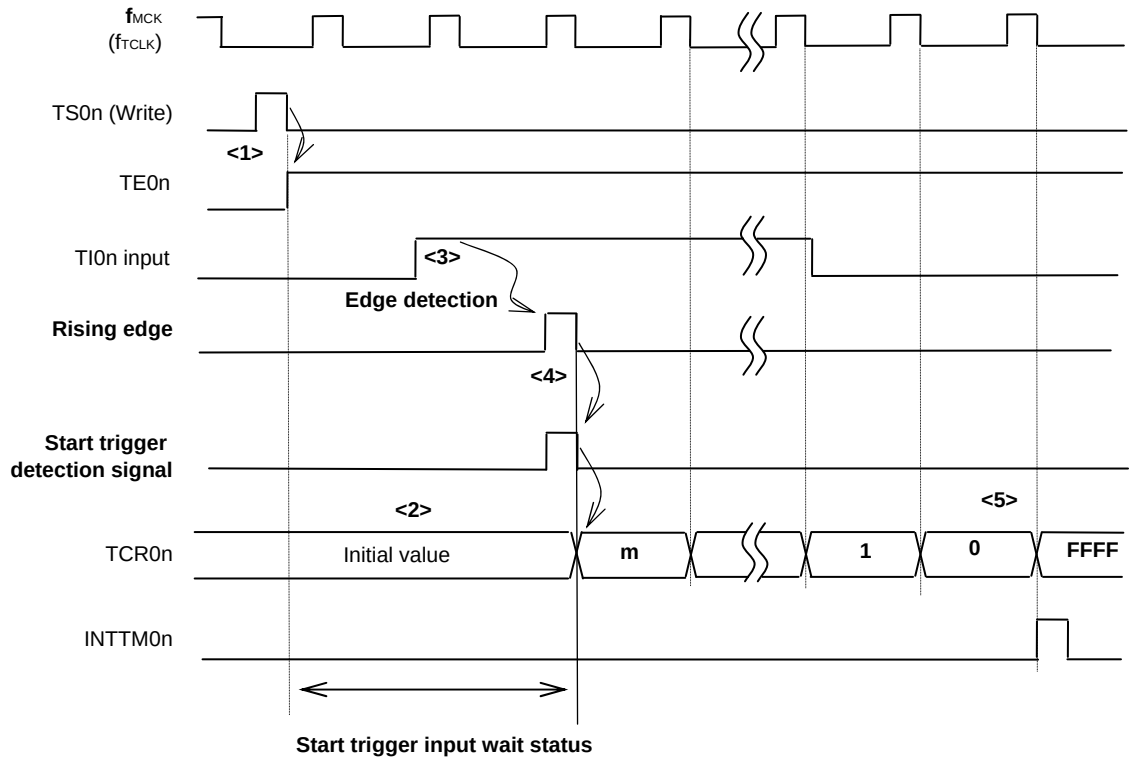
Caution In the first cycle operation of count clock after writing the TS0n bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0 = 1.

Remark The timing is shown in Figure 6-35 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 TI0n input.

(4) Operation of one-count mode

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<2> Timer count register 0n (TCR0n) holds the initial value until start trigger generation.
<3> Rising edge of the TI0n input is detected.
<4> On start trigger detection, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and count starts.
<5> When the TCR0n register counts down and its count value is 0000H, INTTM0n is generated and the value of the TCR0n register becomes FFFFH and counting stops.

Figure 6-36 Operation Timing (In One-count Mode)

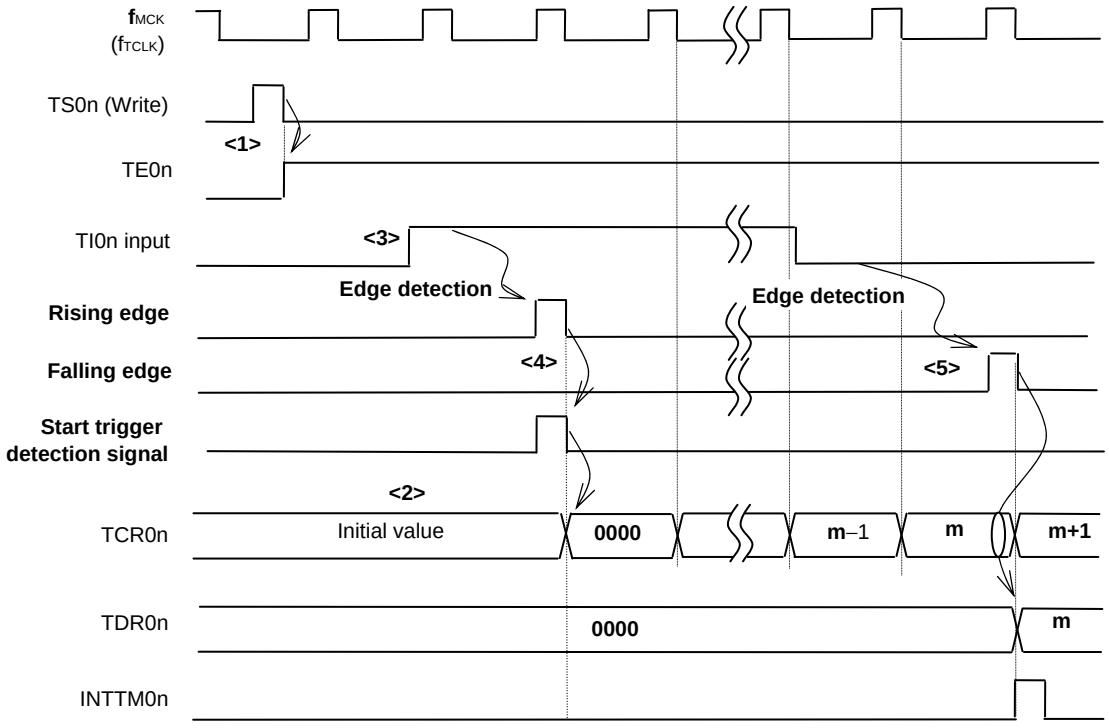


Remark The timing is shown in Figure 6-36 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

(5) Operation of capture & one-count mode (high-level width measurement)

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit of timer channel start register 0 (TS0).
<2> Timer count register 0n (TCR0n) holds the initial value until start trigger generation.
<3> Rising edge of the TI0n input is detected.
<4> On start trigger detection, the value of 0000H is loaded to the TCR0n register and count starts.
<5> On detection of the falling edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated.

Figure 6-37 Operation Timing (In Capture & One-count Mode : High-level Width Measurement)

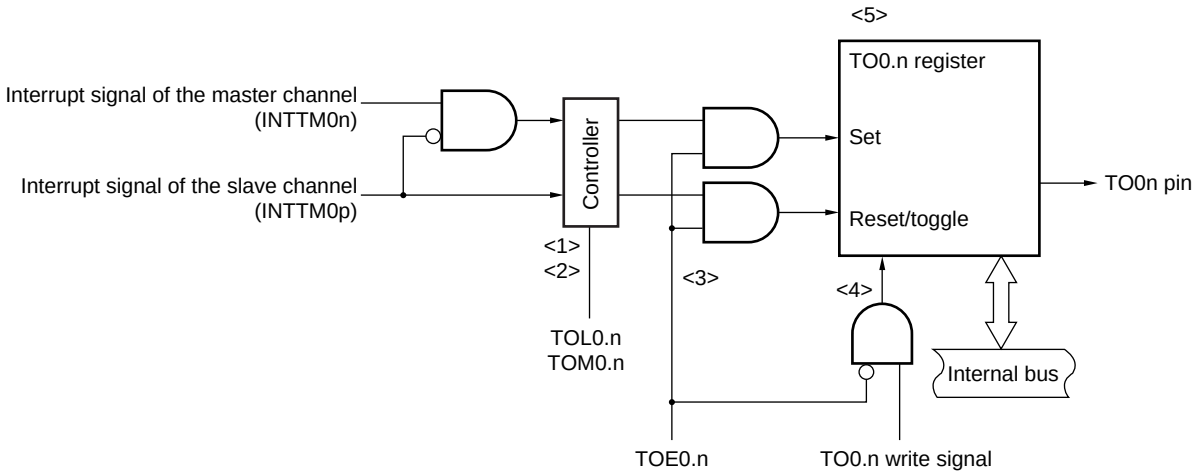


Remark The timing is shown in Figure 6-37 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

6.6 Channel Output (TO0n pin) Control

6.6.1 TO0n pin output circuit configuration

Figure 6-38 Output Circuit Configuration



The following describes the TO0n pin output circuit.

<1> When TOM0.n = 0 (master channel output mode), the set value of timer output level register 0 (TOL0) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register 0 (TO0).

<2> When TOM0.n = 1 (slave channel output mode), both INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0 register.
At this time, the TOL0 register becomes valid and the signals are controlled as follows:

When TOL0.n = 0:	Forward operation (INTTM0n → set, INTTM0p → reset)
When TOL0.n = 1:	Reverse operation (INTTM0n → reset, INTTM0p → set)

When INTTM0n and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTM0n (set signal) is masked.

<3> While timer output is enabled (TOE0.n = 1), INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0 register. Writing to the TO0 register (TO0n write signal) becomes invalid.
When TOE0.n = 1, the TO0n pin output never changes with signals other than interrupt signals.
To initialize the TO0n pin output level, it is necessary to set timer operation is stopeed (TOE0.n = 0) and to write a value to the TO0 register.

<4> While timer output is disabeled (TOE0.n = 0), writing to the TO0.n bit to the target channel (TO0n write signal) becomes valid. When timer output is disabeled (TOE0.n = 0), neither INTTM0n (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TO0 register.

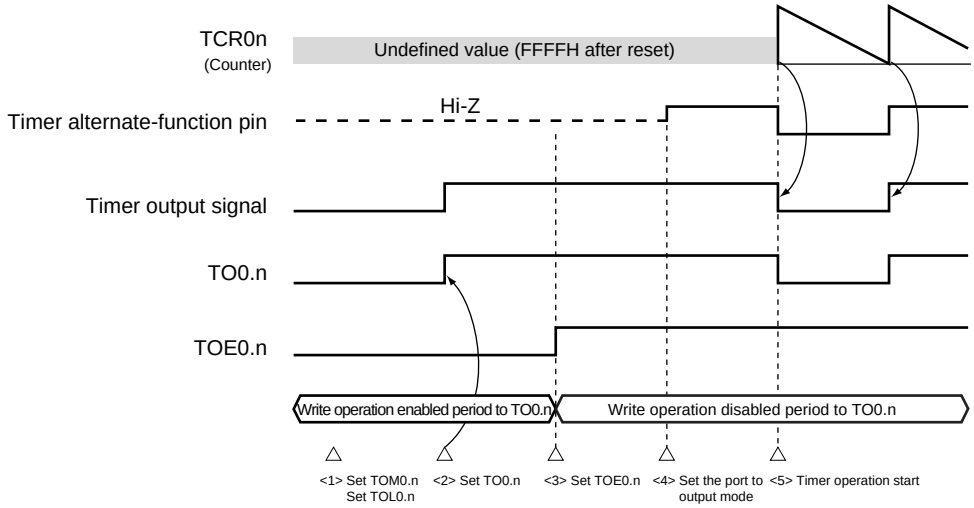
<5> The TO0 register can always be read, and the TO0n pin output level can be checked.

Remark n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

6.6.2 TO0n Pin Output Setting

The following figure shows the procedure and status transition of the TO0n output pin from initial setting to timer operation start.

Figure 6-39 Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOM0.n bit (0: Master channel output mode, 1: Slave channel output mode)
- TOL0.n bit (0: Forward output, 1: Reverse output)

<2> The timer output signal is set to the initial status by setting timer output register 0 (TO0).
<3> The timer output operation is enabled by writing 1 to the TOE0.n bit (writing to the TO0 register is disabled).
<4> The port is set to digital I/O by port mode control register (PMCxx) (see **6.3 (15) Port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4)**).
<5> The port I/O setting is set to output (see **6.3 (15) Port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4)**).
<6> The timer operation is enabled (TS0.n = 1).

Remark n: Channel number (n = 0 to 7)

6.6.3 Cautions on Channel Output Operation

(1) Changing values set in the registers TO0, TOE0, TOL0, and TOM0 during timer operation

Since the timer operations (operations of timer/counter register 0n (TCR0n) and timer data register 0n (TDR0n)) are independent of the TO0n output circuit and changing the values set in timer output register 0 (TO0), timer output enable register 0 (TOE0), timer output level register 0 (TOL0), and timer output mode register 0 (TOM0) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TO0n pin by timer operation, however, set the TO0, TOE0, TOL0, and TOM0 registers to the values stated in the register setting example of each operation.

When the values set to the TOE0, TOL0, and TOM0 registers (but not the TO0 register) are changed close to the occurrence of the timer interrupt (INTTM0n) of each channel, the waveform output to the TO0n pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTM0n) occurs.

Remark n: Channel number (n = 0 to 7)

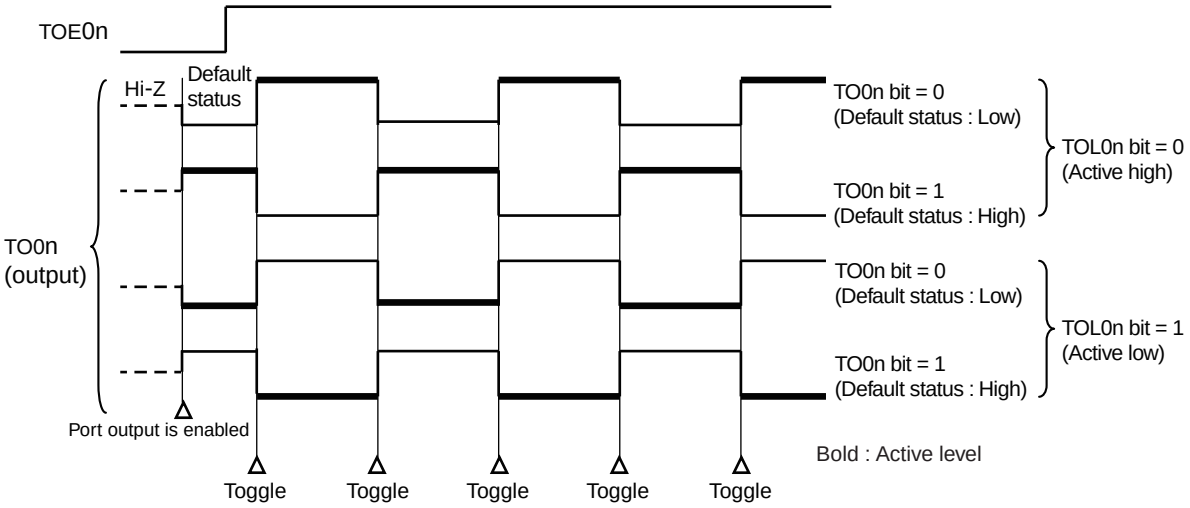
(2) Default level of TO0n pin and output level after timer operation start

The change in the output level of the TO0n pin when timer output register 0 (TO0) is written while timer output is disabled (TOE0.n = 0), the initial level is changed, and then timer output is enabled (TOE0.n = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOM0.n = 0) setting

The setting of timer output level register 0 (TOL0) is invalid when master channel output mode (TOM0.n = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TO0n pin is reversed.

Figure 6-40. TO0n Pin Output Status at Toggle Output (TOM0n = 0)



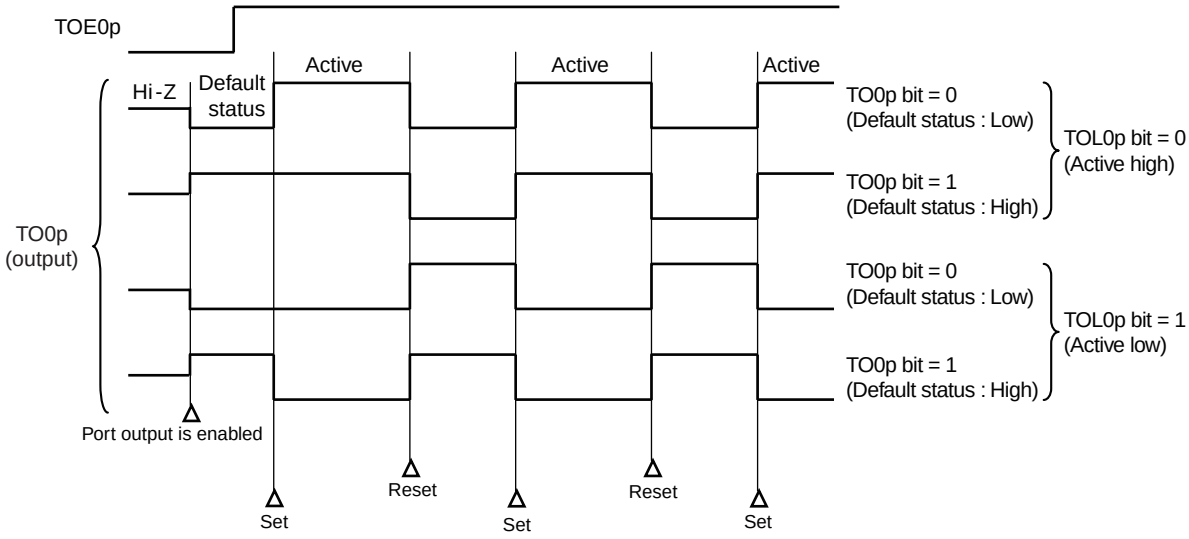
Remarks 1. Toggle: Reverse TO0n pin output status

2. n: Channel number (n = 0 to 7)

(b) When operation starts with slave channel output mode (TOM0.p = 1) setting (PWM output))

When slave channel output mode (TOM0.p = 1), the active level is determined by timer output level register 0 (TOL0) setting.

Figure 6-41. TO0n Pin Output Status at PWM Output (TOM0.p = 1)



Remarks 1. Set: The output signal of the TO0n pin changes from inactive level to active level.
Reset: The output signal of the TO0n pin changes from active level to inactive level.

2. n: Channel number (n = 0 to 7)

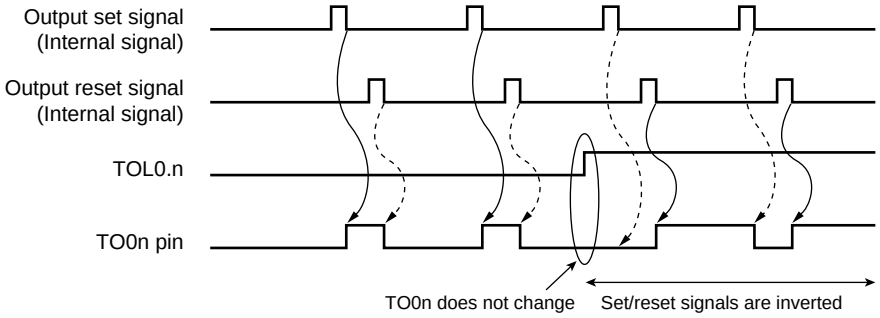
(3) Operation of TO0n pin in slave channel output mode (TOM0.n = 1)

(a) When timer output level register 0 (TOL0) setting has been changed during timer operation

When the TOL0 register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TO0n pin change condition. Rewriting the TOL0 register does not change the output level of the TO0n pin.

The operation when TOM0.n is set to 1 and the value of the TOL0 register is changed while the timer is operating (TE0.n = 1) is shown below.

Figure 6-42. Operation when TOL0 Register Has Been Changed during Timer Operation



Remarks 1. Set: The output signal of the TO0n pin changes from inactive level to active level.
Reset: The output signal of the TO0n pin changes from active level to inactive level.

2. n: Channel number (n = 0 to 7)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TO0n pin/TO0.n bit set timing at master channel timer interrupt (INTTM0n) generation is delayed by 1 count clock by the slave channel.

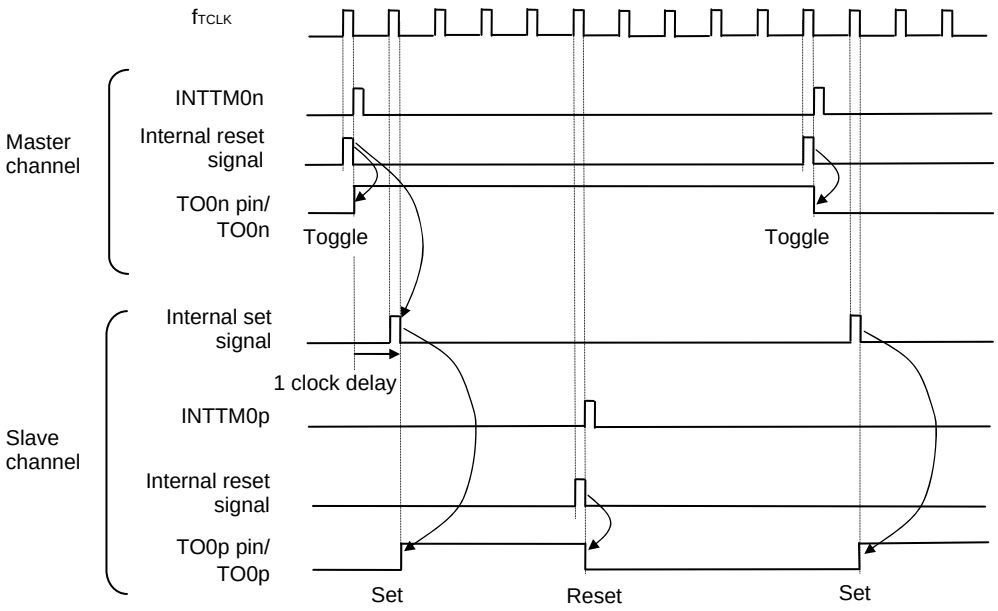
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-43 shows the set/reset operating statuses where the master/slave channels are set as follows.

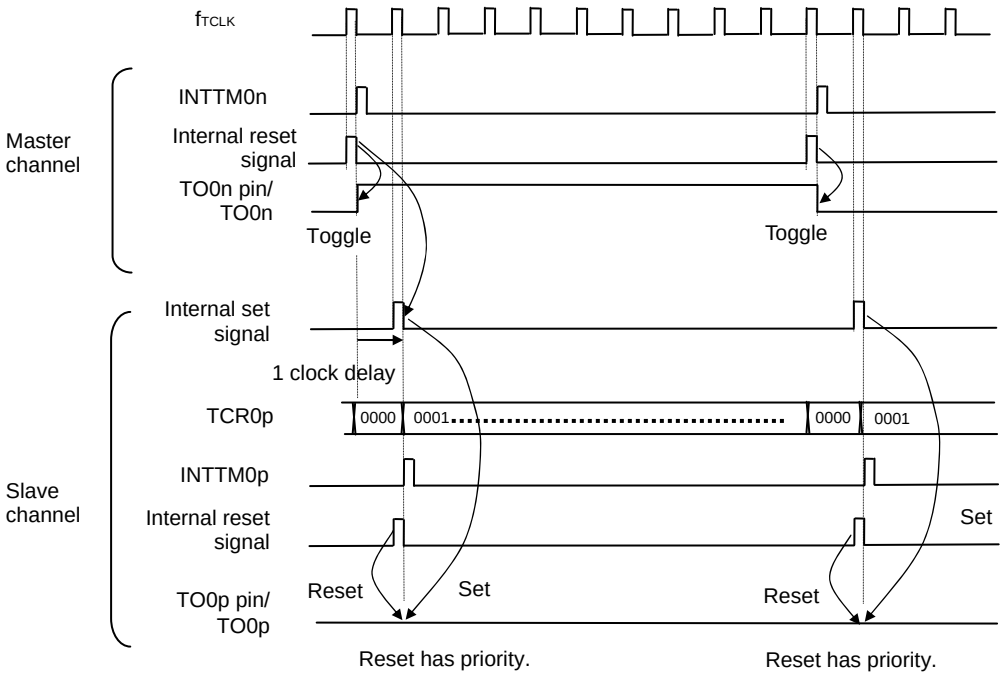
Master channel: TOE0.n = 1, TOM0.n = 0, TOL0.n = 0
Slave channel: TOE0.p = 1, TOM0.p = 1, TOL0.p = 0

Figure 6-43. Set/Reset Timing Operating Statuses

(1) Basic operation timing



(2) Operation timing when 0 % duty



Remarks 1. Internal reset signal: TO0n pin reset/toggle signal
Internal set signal: TO0n pin set signal

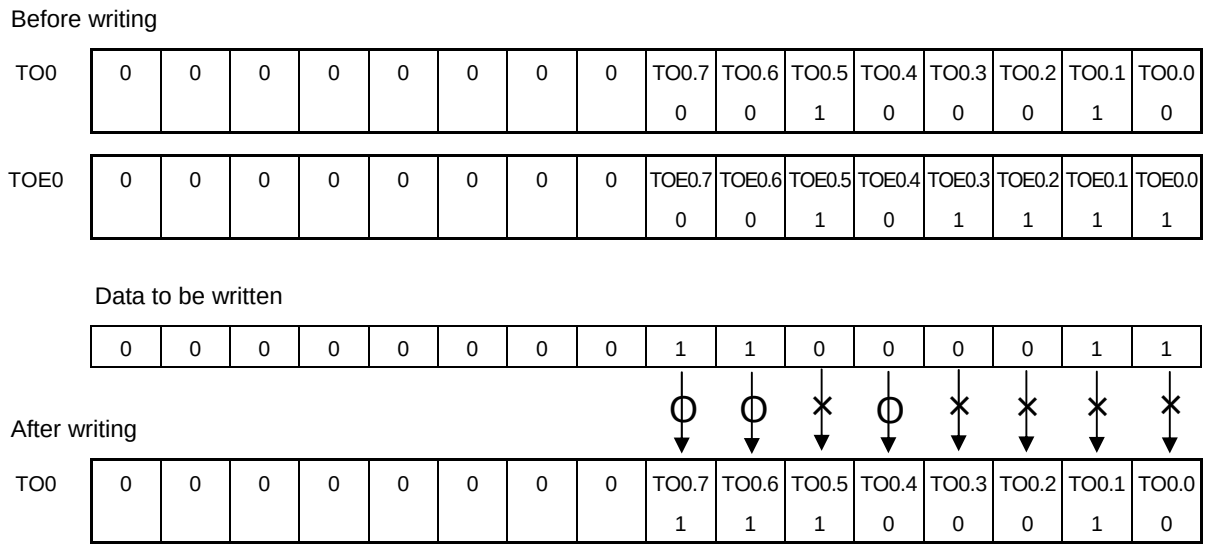
2. n: Channel number (n = 0 to 7)
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

6.6.4 Collective manipulation of TO0.n bit

In timer output register 0 (TO0), the setting bits for all the channels are located in one register in the same way as timer channel start register 0 (TS0). Therefore, the TO0.n bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TO0.n bits (TOE0.n = 0) that correspond to the relevant bits of the channel used to perform output (TO0n).

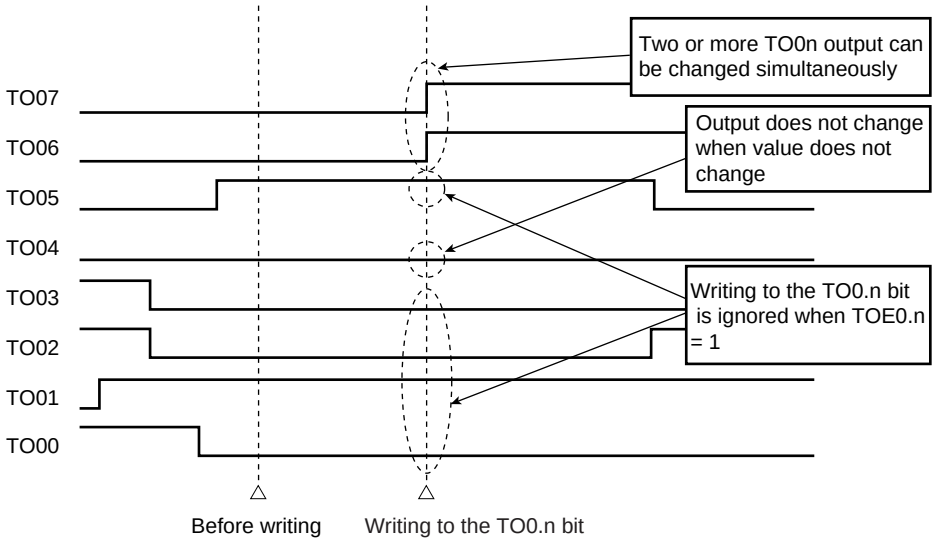
Figure 6-44 Example of TO0n Bit Collective Manipulation



Writing is done only to the TO0.n bit with TOE0.n = 0, and writing to the TO0.n bit with TOE0.n = 1 is ignored.

TO0n (channel output) to which TOE0n = 1 is set is not affected by the write operation. Even if the write operation is done to the TO0.n bit, it is ignored and the output change by timer operation is normally done.

Figure 6-45. TO0n Pin Statuses by Collective Manipulation of TO0.n Bit



Caution While timer output is enabled (TOE0.n = 1), even if the output by timer interrupt of each timer (INTTM0n) contends with writing to the TO0.n bit, output is normally done to the TO0n pin.

Remark n: Channel number (n = 0 to 7)

6.6.5 Timer Interrupt and TO0n Pin Output at Operation Start

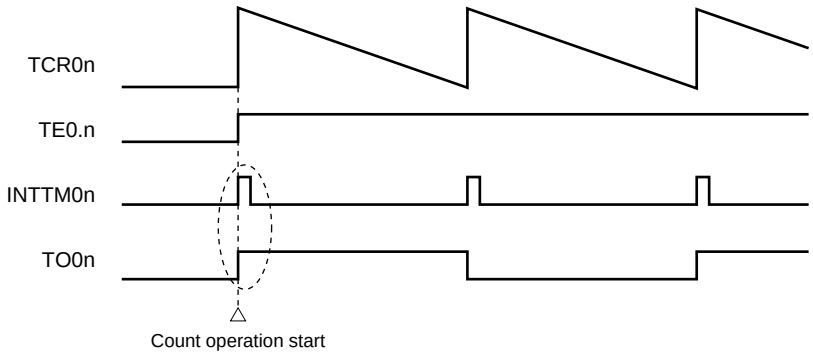
In the interval timer mode or capture mode, the MD0n0 bit in timer mode register 0n (TMR0n) sets whether or not to generate a timer interrupt at count start.

When MD0n0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTM0n) generation.

In the other modes, neither timer interrupt at count operation start nor TO0n output is controlled.

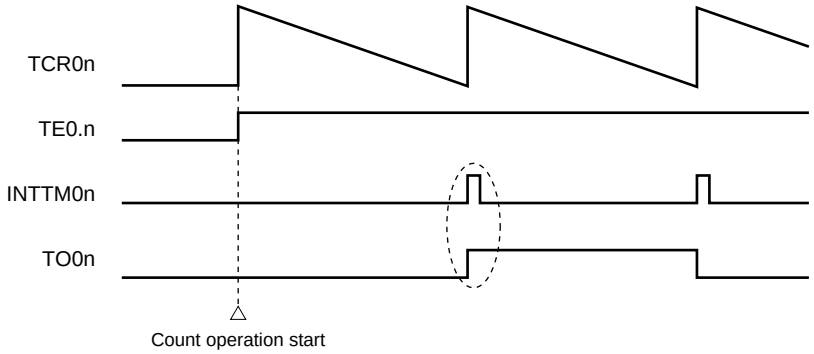
Figures 6-37 and 6-38 show operation examples when the interval timer mode (TOE0.n = 1, TOM0.n = 0) is set.

Figure 6-46. When MD0n0 is set to 1



When MD0n0 is set to 1, a timer interrupt (INTTM0n) is output at count operation start, and TO0n performs a toggle operation.

Figure 6-47. When MD0n0 is set to 0



When MD0n0 is set to 0, a timer interrupt (INTTM0n) is not output at count operation start, and TO0n does not change either. After counting one cycle, INTTM0n is output and TO0n performs a toggle operation.

Remark n: Channel number (n = 0 to 7)

6.7 Independent Channel Operation Function of Timer Array Unit

6.7.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTM0n (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

Generation period of INTTM0n (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

(2) Operation as square wave output

TO0n performs a toggle operation as soon as INTTM0n has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TO0n can be calculated by the following expressions.

- Period of square wave output from TO0n = Period of count clock × (Set value of TDR0n + 1) × 2
- Frequency of square wave output from TO0n = Frequency of count clock/{(Set value of TDR0n + 1) × 2}

Timer/counter register 0n (TCR0n) operates as a down counter in the interval timer mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) at the first count clock after the channel start trigger bit (TS0.n, TSH01, TSH03) of timer channel start register 0 (TS0) is set to 1. If the MD0n0 bit of timer mode register 0n (TMR0n) is 0 at this time, INTTM0n is not output and TO0n is not toggled. If the MD0n0 bit of the TMR0n register is 1, INTTM0n is output and TO0n is toggled.

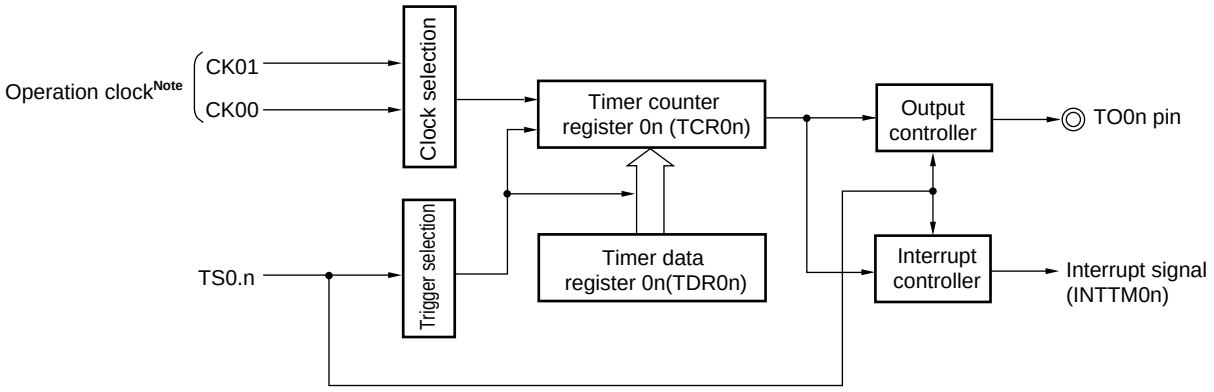
After that, the TCR0n register count down in synchronization with the count clock.

When TCR0n = 0000H, INTTM0n is output and TO0n is toggled at the next count clock. At the same time, the TCR0n register loads the value of the TDR0n register again. After that, the same operation is repeated.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

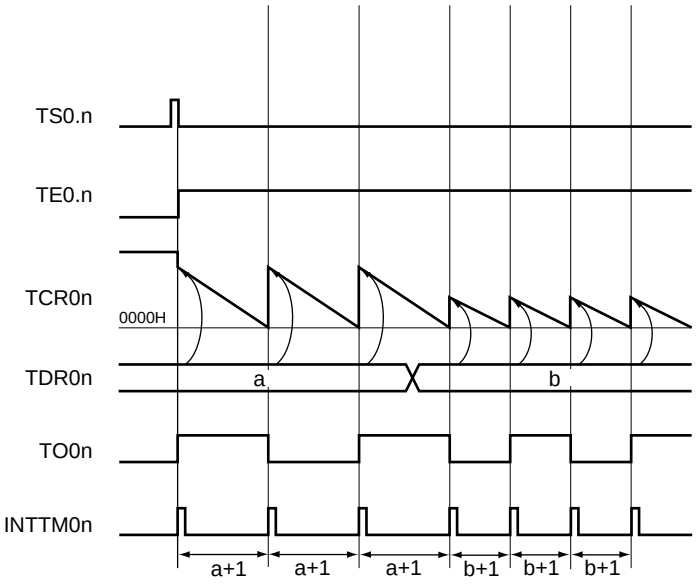
Remark n: Channel number (n = 0 to 7)

Figure 6-48. Block Diagram of Operation as Interval Timer/Square Wave Output



Note When channels 1 and 3, the clock is selected from CK00, CK01, CK02 and CK03.

Figure 6-49. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MD0n0 = 1)

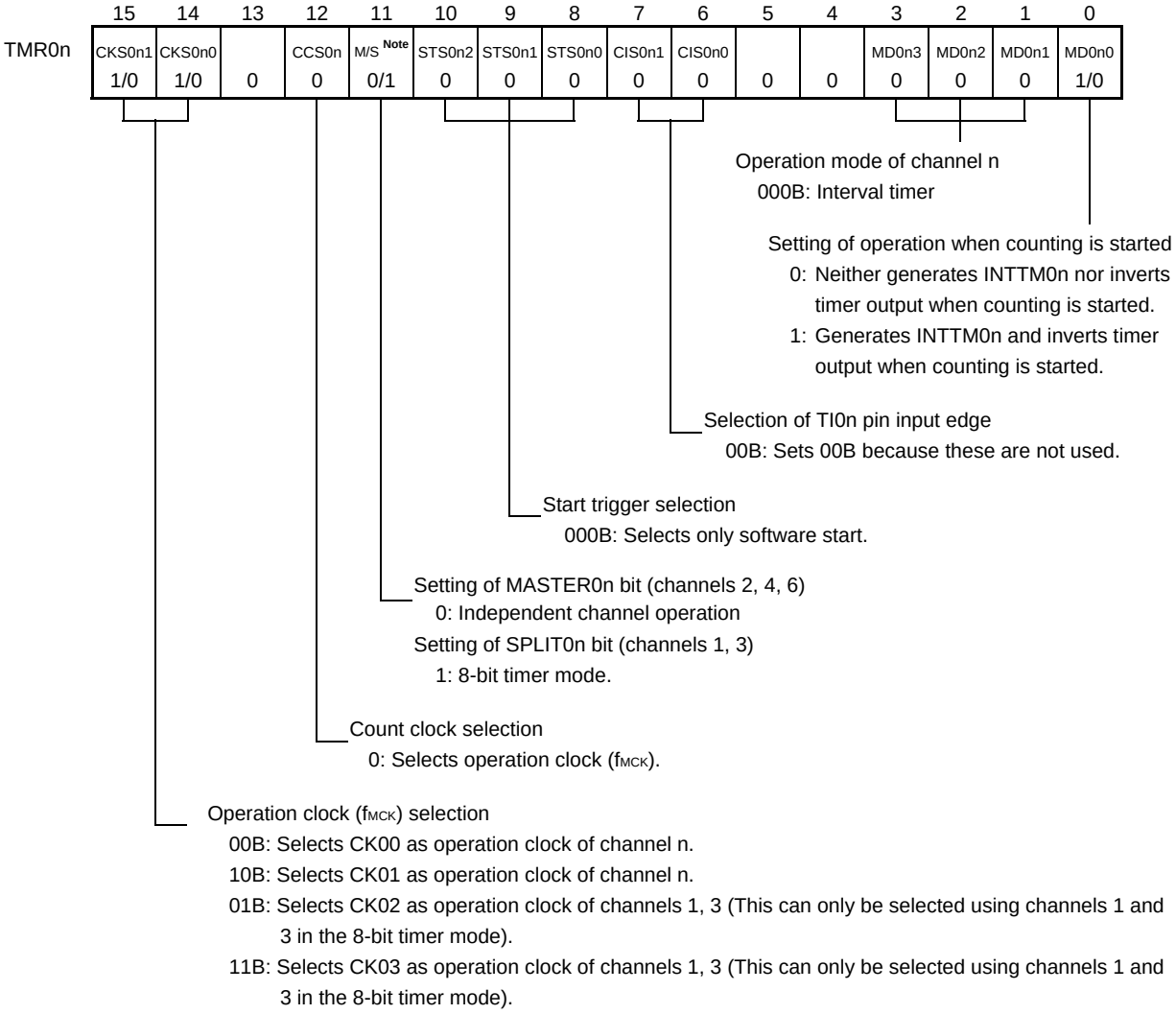


Remarks 1. n: Channel number (n = 0 to 7)

2. TS0.n: Bit n of timer channel start register 0 (TS0)
TE0.n: Bit n of timer channel enable status register 0 (TE0)
TCR0n: Timer/counter register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
TO0n: TO0n pin output signal

Figure 6-50. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0.n 1/0	0: Outputs 0 from TO0n. 1: Outputs 1 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0.n 1/0	0: Stops the TO0n output operation by counting operation. 1: Enables the TO0n output operation by counting operation.

Note TMR00, TMR02, TMR04 to TMR07: MASTER0n bit
TMR01, TMR03: SPLIT0n bit

Remark n: Channel number (n = 0 to 7)

Figure 6-50. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0.n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode)

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0.n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 0 to 7)

Figure 6-51. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01 (or CK02 and CK03 when using the 8-bit timer mode).	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets interval (period) value to timer data register 0n (TDR0n).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TO0n output Clears the TOM0.n bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL0.n bit to 0. Sets the TO0.n bit and determines default level of the TO0n output. →	The TO0n pin goes into Hi-Z output state. The TO0n default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOE0.n bit to 1 and enables operation of TO0n. →	TO0n does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0n pin outputs the TO0n set level.
Operation start	(Sets the TOE0.n bit to 1 only if using TO0n output and resuming operation.). Sets the TS0.n (TSH01, TSH03) bit to 1. → The TS0.n (TSH01, TSH03) bit automatically returns to 0 because it is a trigger bit.	TE0.n (TEH01, TEH03) = 1, and count operation starts. Value of the TDR0n register is loaded to timer/counter register 0n (TCR0n) at the count clock input. INTTM0n is generated and TO0n performs toggle operation if the MD0n0 bit of the TMR0n register is 1.
During operation	Set values of the TMR0n register, TOM0.n, and TOL0.n bits cannot be changed. Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TO0 and TOE0 registers can be changed.	Counter (TCR0n) counts down. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again and the count operation is continued. By detecting TCR0n = 0000H, INTTM0n is generated and TO0n performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT0.n (TTH01, TTH03) bit is set to 1. → The TT0.n (TTH01, TTH03) bit automatically returns to 0 because it is a trigger bit.	TE0.n (TEH01, TEH03), and count operation stops. The TCR0n register holds count value and stops. The TO0n output is not initialized but holds current status.
	The TOE0.n bit is cleared to 0 and value is set to the TO0.n bit. →	The TO0n pin outputs the TO0.n bit set level.

Operation is resumed.

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

Figure 6-51. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TO0n pin output level Clears the TO0.n bit to 0 after the value to be held is set to the port register. →	The TO0n pin output level is held by port function.
	When holding the TO0n pin output level is not necessary Switches the port mode register to input mode. →	The TO0n pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0.n bit is cleared to 0 and the TO0n pin is set to port mode.)

Remark n: Channel number (n = 0 to 7)

6.7.2 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 TI0n pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDR0n + 1

Timer/counter register 0n (TCR0n) operates as a down counter in the event counter mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) by setting any channel start trigger bit (TS0.n, TSH01, TSH03) of timer channel start register 0 (TS0) to 1.

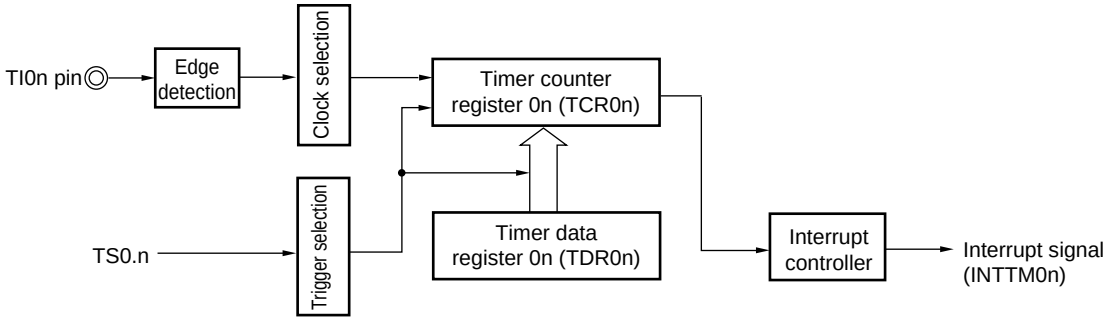
The TCR0n register counts down each time the valid input edge of the TI0n pin has been detected. When TCR0n = 0000H, the TCR0n register loads the value of the TDR0n register again, and outputs INTTM0n.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TO0n pin. Stop the output by setting the TOE0.n bit of timer output enable register 0 (TOE0) to 0.

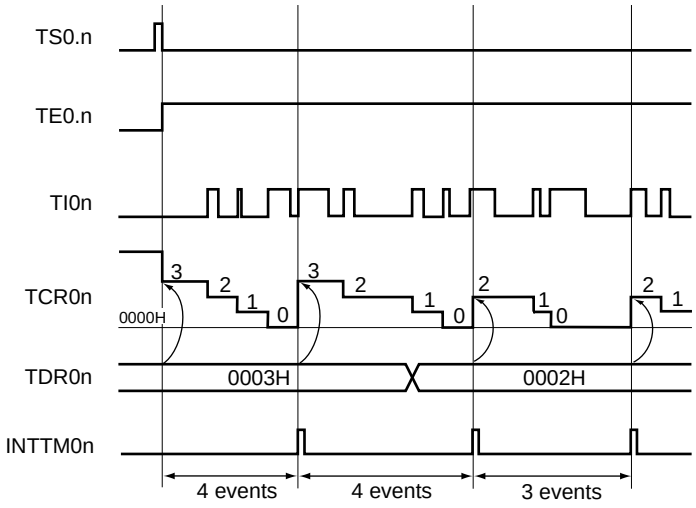
The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid during the next count period.

Figure 6-52. Block Diagram of Operation as External Event Counter



Remark n: Channel number (n = 0 to 7)

Figure 6-53. Example of Basic Timing of Operation as External Event Counter

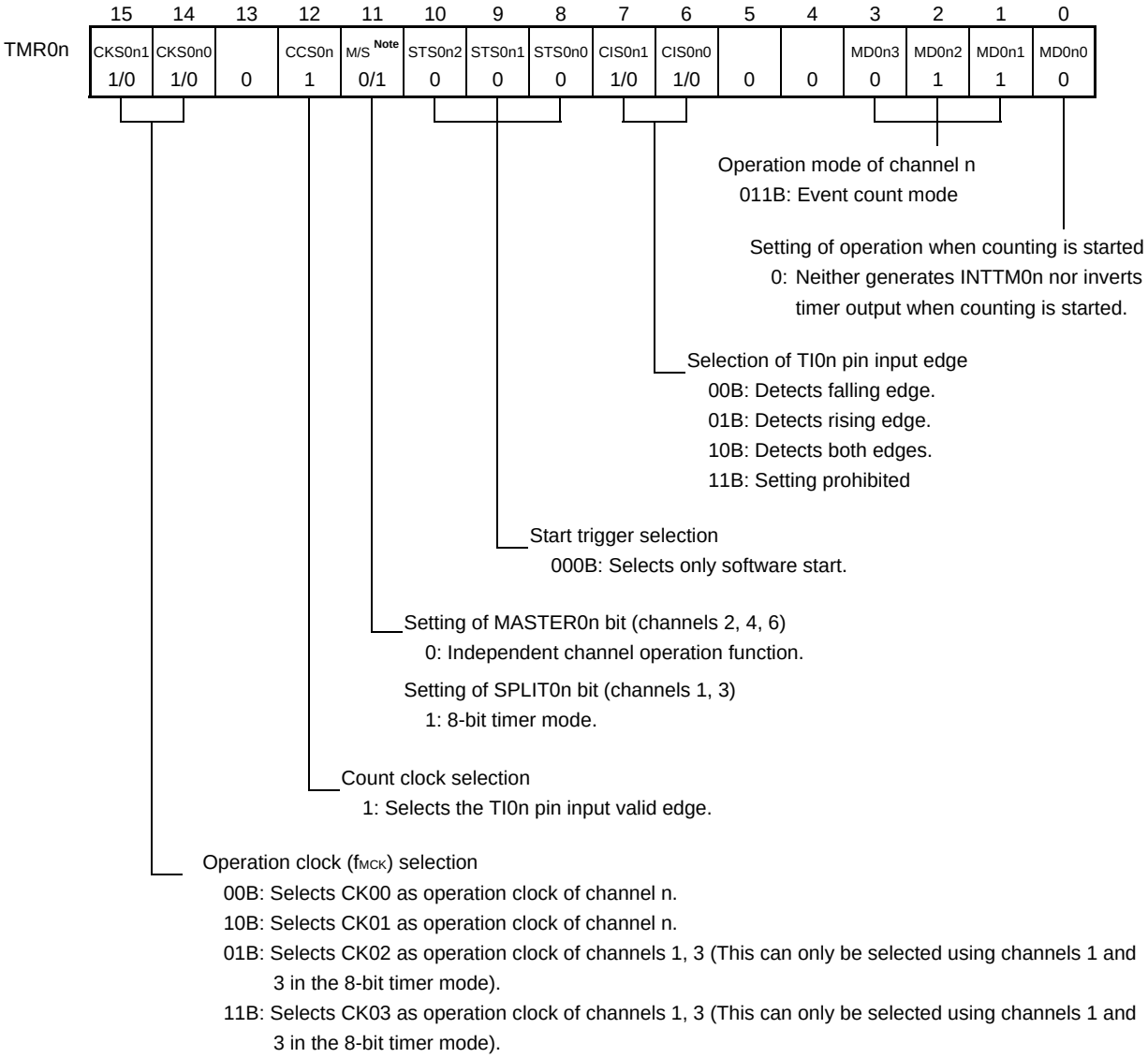


Remarks 1. n: Channel number (n = 0 to 7)

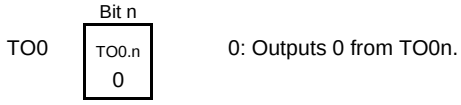
2. TS0.n: Bit n of timer channel start register 0 (TS0)
TE0.n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer/counter register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)

Figure 6-54. Example of Set Contents of Registers in External Event Counter Mode (1/2)

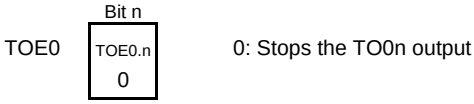
(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)



(c) Timer output enable register 0 (TOE0)



Note TMR00, TMR02, TMR04 to TMR07: MASTER0n bit
TMR01, TMR03: SPLIT0n bit

Remark n: Channel number (n = 0 to 7)

Figure 6-54. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0.n 0	0: Cleared to 0 when TOM0.n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0.n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 0 to 7)

Figure 6-55. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01 (or CK02 and CK03 when using the 8-bit timer mode).	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets number of counts to timer data register 0n (TDR0n). Clears the TOE0.n bit of timer output enable register 0 (TOE0) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0.n (TSH01, TSH03) bit to 1. → The TS0.n (TSH01, TSH03) bit automatically returns to 0 because it is a trigger bit.	TE0.n (TEH01, TEH03) = 1, and count operation starts. Value of the TDR0n register is loaded to timer/counter register 0n (TCR0n) and detection of the TI0n pin input edge is awaited.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TMR0n register, TOM0.n, TOL0.n, TO0.n, and TOE0.n bits cannot be changed.	Counter (TCR0n) counts down each time input edge of the TI0n pin has been detected. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again, and the count operation is continued. By detecting TCR0n = 0000H, the INTTM0n output is generated. After that, the above operation is repeated.
Operation stop	The TT0.n (TTH01, TTH03) bit is set to 1. → The TT0.n (TTH01, TTH03) bit automatically returns to 0 because it is a trigger bit.	TE0.n (TEH01, TEH03) = 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed. ↑

Remark n: Channel number (n = 0 to 7)

6.7.3 Operation as frequency divider (channel 0 only)

The timer array unit can be used as a frequency divider that divides a clock input to the TI00 pin and outputs the result from the TO00 pin.

The divided clock frequency output from TO00 can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR00 + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDR00 + 1)

Timer/counter register 00 (TCR00) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS0.0) of timer channel start register 0 (TS0) is set to 1, the TCR00 register loads the value of timer data register 00 (TDR00) when the TI00 valid edge is detected.

If the MD000 bit of timer mode register 00 (TMR00) is 0 at this time, INTTM00 is not output and TO00 is not toggled. If the MD000 bit of timer mode register 00 (TMR00) is 1, INTTM00 is output and TO00 is toggled.

After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0000H, it toggles TO00. At the same time, the TCR00 register loads the value of the TDR00 register again, and continues counting.

If detection of both the edges of the TI00 pin is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output.

The period of the TO00 output clock includes a sampling error of one period of the operation clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operation clock period (error)

The TDR00 register can be rewritten at any time. The new value of the TDR00 register becomes valid during the next count period.

Figure 6-56. Block Diagram of Operation as Frequency Divider

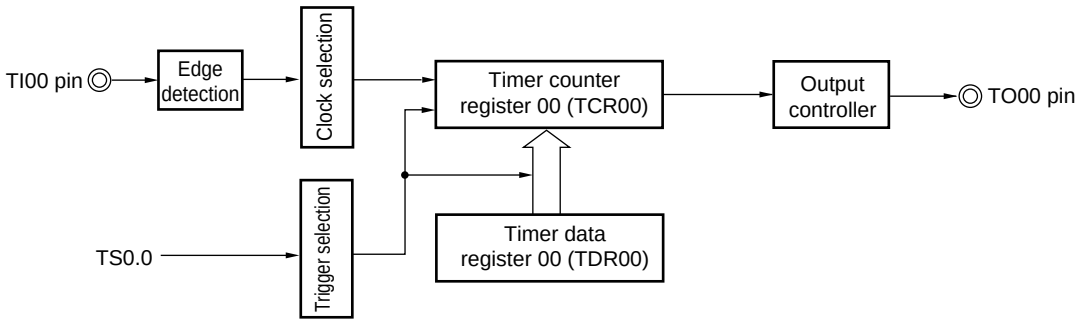
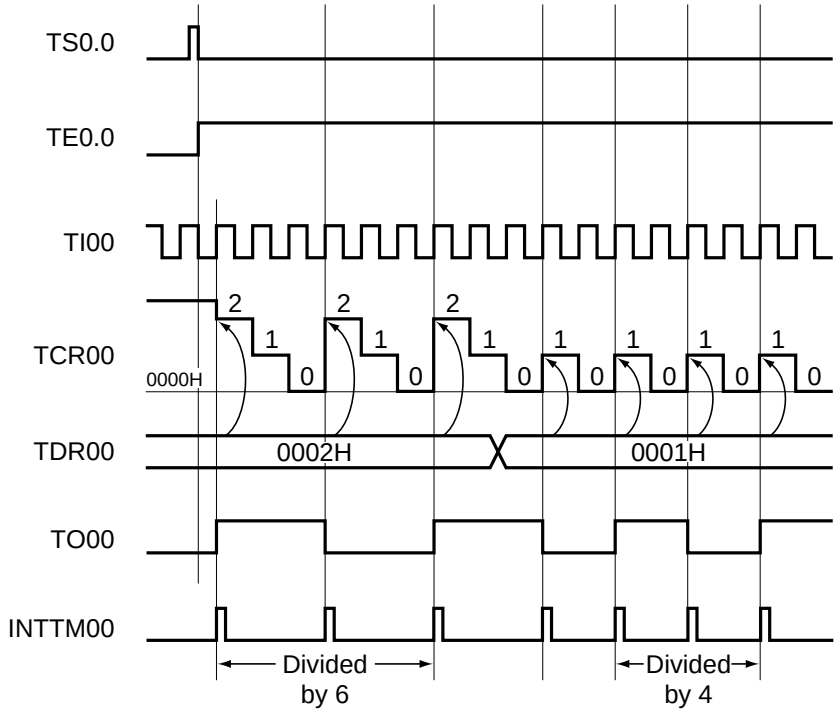


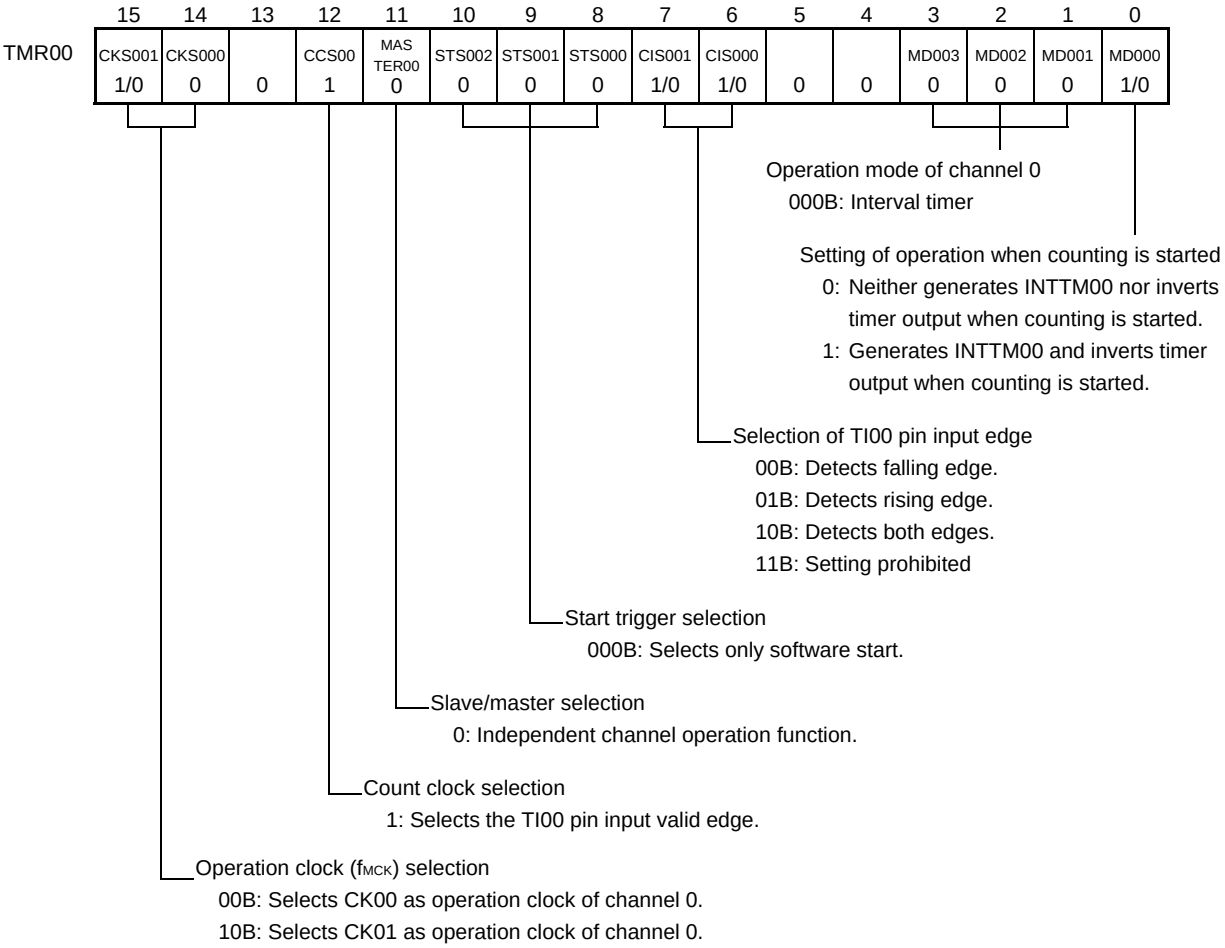
Figure 6-57. Example of Basic Timing of Operation as Frequency Divider (MD000 = 1)



Remark TS0.0: Bit n of timer channel start register 0 (TS0)
TE0.0: Bit n of timer channel enable status register 0 (TE0)
TI00: TI00 pin input signal
TCR00: Timer/counter register 00 (TCR00)
TDR00: Timer data register 00 (TDR00)
TO00: TO00 pin output signal

Figure 6-58. Example of Set Contents of Registers During Operation as Frequency Divider

(a) Timer mode register 00 (TMR00)



(b) Timer output register 0 (TO0)

	Bit 0	
TO0	TO0.0 1/0	0: Outputs 0 from TO00. 1: Outputs 1 from TO00.

(c) Timer output enable register 0 (TOE0)

	Bit 0	
TOE0	TOE0.0 1/0	0: Stops the TO00 output operation by counting operation. 1: Enables the TO00 output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit 0	
TOL0	TOL0.0 0	0: Cleared to 0 when TOM0.0 = 0 (master channel output mode)

(e) Timer output mode register 0 (TOM0)

	Bit 0	
TOM0	TOM0.0 0	0: Sets master channel output mode.

Figure 6-59. Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register 00 (TDR00).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM0.0 bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL0.0 bit to 0. Sets the TO0.0 bit and determines default level of the TO00 output. →	The TO00 pin goes into Hi-Z output state. The TO00 default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0.0 bit to 1 and enables operation of TO00. →	TO00 does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO00 pin outputs the TO00 set level.
Operation start	Sets the TOE0.0 bit to 1 (only when operation is resumed). → Sets the TS0.0 bit to 1. The TS0.0 bit automatically returns to 0 because it is a trigger bit.	TE0.0 = 1, and count operation starts. Value of the TDR00 register is loaded to timer/counter register 00 (TCR00) at the count clock input. INTTM00 is generated and TO00 performs toggle operation if the MD000 bit of the TMR00 register is 1.
During operation	Set value of the TDR00 register can be changed. The TCR00 register can always be read. The TSR00 register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR00 register, TOM00, and TOL00 bits cannot be changed.	Counter (TCR00) counts down. When count value reaches 0000H, the value of the TDR00 register is loaded to the TCR00 register again, and the count operation is continued. By detecting TCR00 = 0000H, INTTM00 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT0.0 bit is set to 1. → The TT0.0 bit automatically returns to 0 because it is a trigger bit.	TE0.0 = 0, and count operation stops. The TCR00 register holds count value and stops. The TO00 output is not initialized but holds current status.
	The TOE0.0 bit is cleared to 0 and value is set to the TO0.0 bit. →	The TO00 pin outputs the TO00 set level.
TAU stop	To hold the TO00 pin output level Clears the TO0.0 bit to 0 after the value to be held is set to the port register. → When holding the TO00 pin output level is not necessary Switches the port mode register to input mode. →	The TO00 pin output level is held by port function. The TO00 pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0.0 bit is cleared to 0 and the TO00 pin is set to port mode).

Operation is resumed.

6.7.4 Operation as input pulse interval measurement

The count value can be captured at the TI0n valid edge and the interval of the pulse input to TI0n can be measured.

The pulse interval can be calculated by the following expression.

TI0n input pulse interval = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0n pin input is sampled using the operating clock selected with the CKS0n bit of timer mode register 0n (TMR0n), so an error of up to one operating clock cycle occurs.

Timer/counter register 0n (TCR0n) operates as an up counter in the capture mode.

When the channel start trigger bit (TS0.n) of timer channel start register 0 (TS0) is set to 1, the TCR0n register counts up from 0000H in synchronization with the count clock.

When the TI0n pin input valid edge is detected, the count value of the TCR0n register is transferred (captured) to timer data register 0n (TDR0n) and, at the same time, the TCR0n register is cleared to 0000H, and the INTTM0n is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

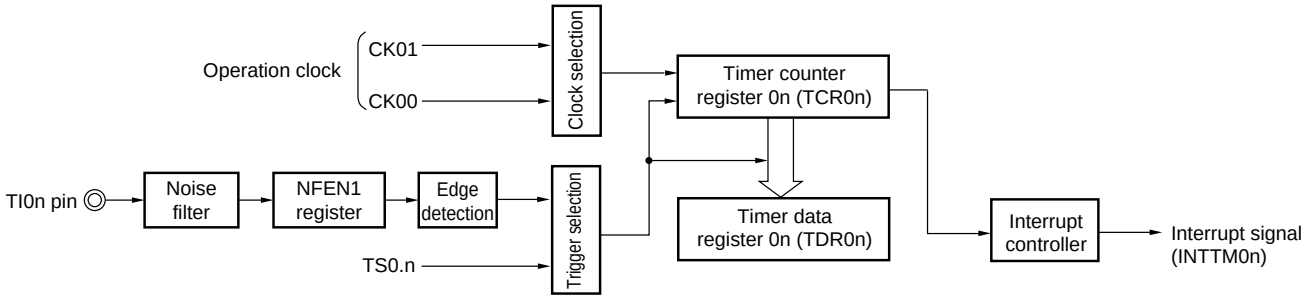
As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

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 TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Set the STS0n2 to STS0n0 bits of the TMR0n register to 001B to use the valid edges of TI0n as a start trigger and a capture trigger.

When TE0.n = 1, a software operation (TS0.n = 1) can be used as a capture trigger, instead of using the TI0n pin input.

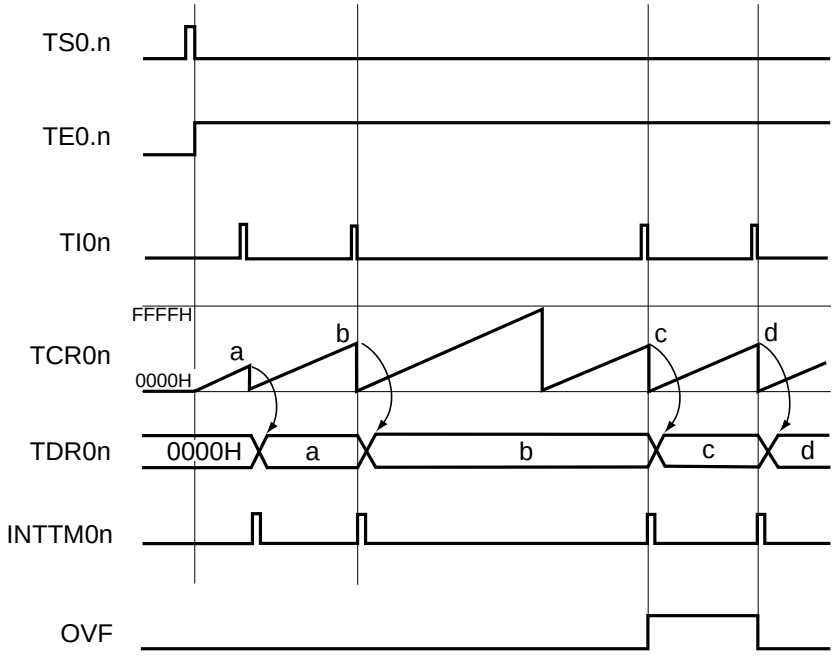
Figure 6-60. Block Diagram of Operation as Input Pulse Interval Measurement



Note For using channels 1 and 3, the clock can be selected from CK00, CK01, CK02, and CK03.

Remark n: Channel number (n = 0 to 7)

Figure 6-61. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MD0n0 = 0)



Remarks 1. n: Channel number (n = 0 to 7)

2. TS0.n: Bit n of timer channel start register 0 (TS0)
TE0.n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer/counter register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
OVF: Bit 0 of timer status register 0n (TSR0n)

Figure 6-62. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register 0n (TMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS0n1	CKS0n0		CCS0n	M/S Note	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0			MD0n3	MD0n2	MD0n1	MD0n0
	1/0	0	0	0	0	0	0	1	1/0	1/0	0	0	0	1	0	1/0

Operation mode of channel n
010B: Capture mode

Setting of operation when counting is started
0: Does not generate INTTM0n when counting is started.
1: Generates INTTM0n when counting is started.

Selection of TI0n pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Capture trigger selection
001B: Selects the TI0n pin input valid edge.

Setting of MASTER0n or SPLIT0n bit
0: Independent channel operation function.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register 0 (TO0)

	Bit n
TO0	TO0.n
	0

0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOE0.n
	0

0: Stops TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOL0.n
	0

0: Cleared to 0 when TOM0.n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOM0.n
	0

0: Sets master channel output mode.

Note TMR00, TMR02, TMR04 to TMR07: MASTER0n bit
TMR01, TMR03: SPLIT0n bit

Remark n: Channel number (n = 0 to 7)

Figure 6-63. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TS0.n bit to 1. → The TS0.n bit automatically returns to 0 because it is a trigger bit.	TE0.n = 1, and count operation starts. Timer/counter register 0n (TCR0n) is cleared to 0000H at the count clock input. When the MD0n0 bit of the TMR0n register is 1, INTTM0n is generated.
During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. The TDR0n register can always be read. The TCR0n register can always be read. The TSR0n register can always be read. Set values of the TOM0.n, TOL0.n, TO0.n, and TOE0.n bits cannot be changed.	Counter (TCR0n) counts up from 0000H. When the TI0n pin input valid edge is detected, the count value is transferred (captured) to timer data register 0n (TDR0n). At the same time, the TCR0n register is cleared to 0000H, and the INTTM0n signal is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TT0.n bit is set to 1. → The TT0.n bit automatically returns to 0 because it is a trigger bit.	TE0.n = 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 0 to 7)

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

Caution When using a channel to implement the LIN-bus, set bit 1 (ISC.1) of the input switch control register (ISC) to 1. In the following descriptions, read TI0n as RxD2.

By starting counting at one edge of the TI0n pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TI0n can be measured. The signal width of TI0n can be calculated by the following expression.

Signal width of TI0n input = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0n pin input is sampled using the operating clock selected with the CKS0n bit of timer mode register 0n (TMR0n), so an error equivalent to one operation clock occurs.

Timer/counter register 0n (TCR0n) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TS0.n) of timer channel start register 0 (TS0) is set to 1, the TE0.n bit is set to 1 and the TI0n pin start edge detection wait status is set.

When the TI0n pin input start edge (rising edge of the TI0n pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TI0n pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register 0n (TDR0n) and, at the same time, INTTM0n is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCR0n register stops at the value “value transferred to the TDR0n register + 1”, and the TI0n pin start edge detection wait status is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow 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 TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Whether the high-level width or low-level width of the TI0n pin is to be measured can be selected by using the CIS0n1 and CIS0n0 bits of the TMR0n register.

Because this function is used to measure the signal width of the TI0n pin input, the TS0.n bit cannot be set to 1 while the TE0.n bit is 1.

CIS0n1, CIS0n0 of TMR0n register = 10B: Low-level width is measured.

CIS0n1, CIS0n0 of TMR0n register = 11B: High-level width is measured.

Figure 6-64. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement

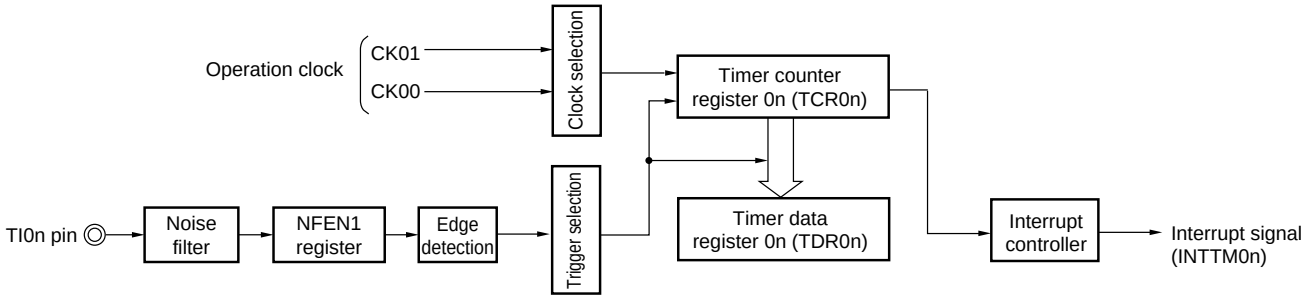
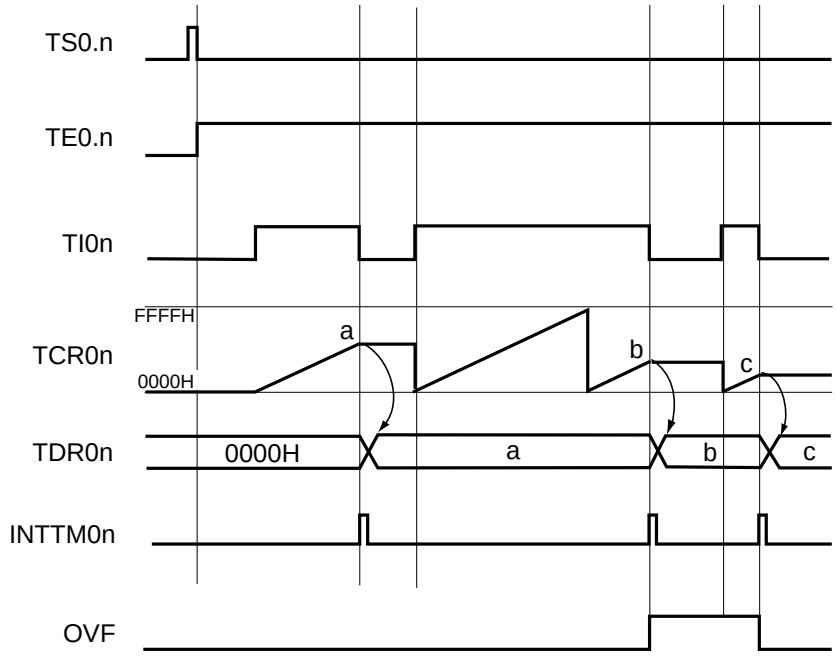


Figure 6-65. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement

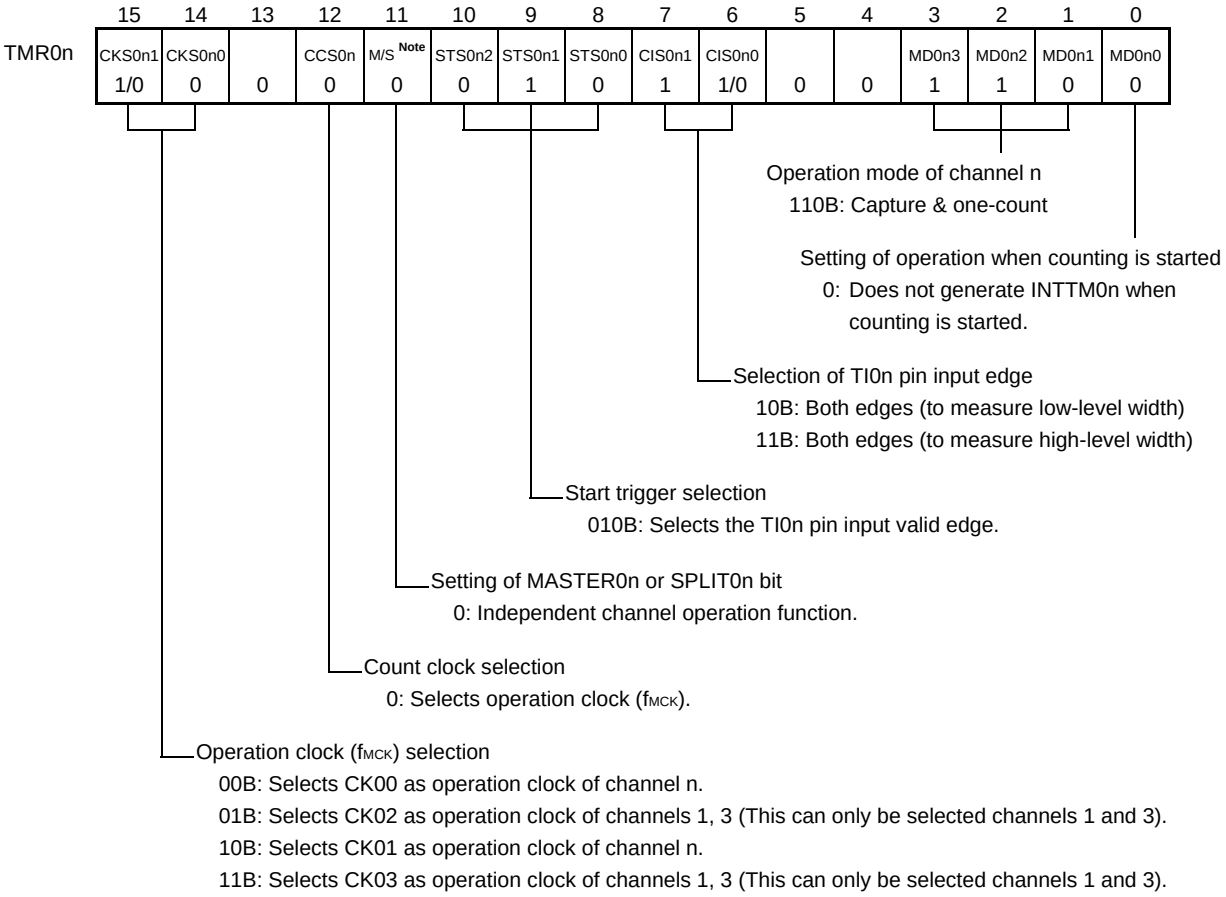


Remarks 1. n: Channel number (n = 0 to 7)

2. TS0.n: Bit n of timer channel start register 0 (TS0)
TE0.n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer/counter register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
OVF: Bit 0 of timer status register 0n (TSR0n)

Figure 6-66. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0.n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0.n	0: Stops the TO0n output operation by counting operation.
	0	

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0.n	0: Cleared to 0 when TOM0.n = 0 (master channel output mode).
	0	

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0.n	0: Sets master channel output mode.
	0	

Note TMR00, TMR02, TMR04 to TMR07: MASTER0n bit
TMR01, TMR03: SPLIT0n bit

Remark n: Channel number (n = 0 to 7)

Figure 6-67. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Clears the TOE0.n bit to 0 and stops operation of TO0n.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0.n bit to 1. → The TS0.n bit automatically returns to 0 because it is a trigger bit.	TE0.n = 1, and the TI0n pin start edge detection wait status is set.
	Detects the TI0n pin input count start valid edge. →	Clears timer/counter register 0n (TCR0n) to 0000H and starts counting up.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TMR0n register, TOM0.n, TOL0.n, TO0.n, and TOE0.n bits cannot be changed.	When the TI0n pin start edge is detected, the counter (TCR0n) counts up from 0000H. If a capture edge of the TI0n pin is detected, the count value is transferred to timer data register 0n (TDR0n) and INTTM0n is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the OVF bit is cleared. The TCR0n register stops the count operation until the next TI0n pin start edge is detected.
Operation stop	The TT0.n bit is set to 1. → The TT0.n bit automatically returns to 0 because it is a trigger bit.	TE0.n = 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 0 to 7)

6.7.6 Operation as delay counter

It is possible to start counting down when the valid edge of the TI0n pin input is detected (an external event), and then generate INTTM0n (a timer interrupt) after any specified interval.

It can also generate INTTM0n (timer interrupt) at any interval by making a software set TS0n = 1 and the count down start during the period of TE0n = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTM0n (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

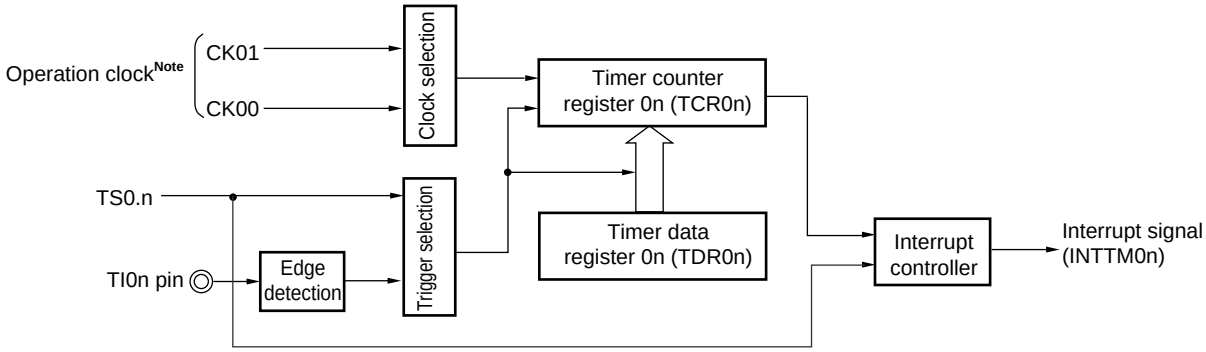
Timer/counter register 0n (TCR0n) operates as a down counter in the one-count mode.

When the channel start trigger bit (TS0.n, TSH01, TSH03) of timer channel start register 0 (TS0) is set to 1, the TE0.n, TEH01, TEH03 bits are set to 1 and the TI0n pin input valid edge detection wait status is set.

Timer/counter register 0n (TCR0n) starts operating upon TI0n pin input valid edge detection and loads the value of timer data register 0n (TDR0n). The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next TI0n pin input valid edge is detected.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

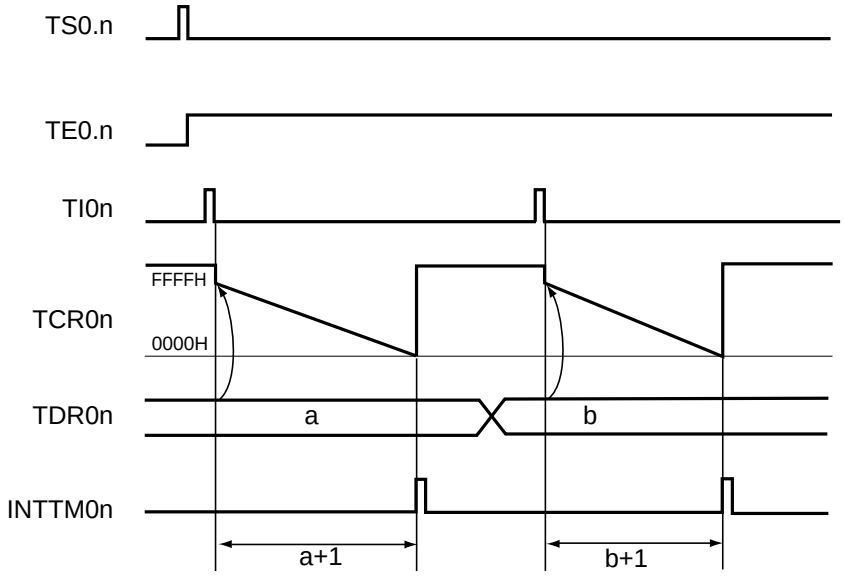
Figure 6-68. Block Diagram of Operation as Delay Counter



Note For using channels 1 and 3, the clock can be selected from CK00, CK01, CK02, and CK03.

Remark n: Channel number (n = 0 to 7)

Figure 6-69. Example of Basic Timing of Operation as Delay Counter

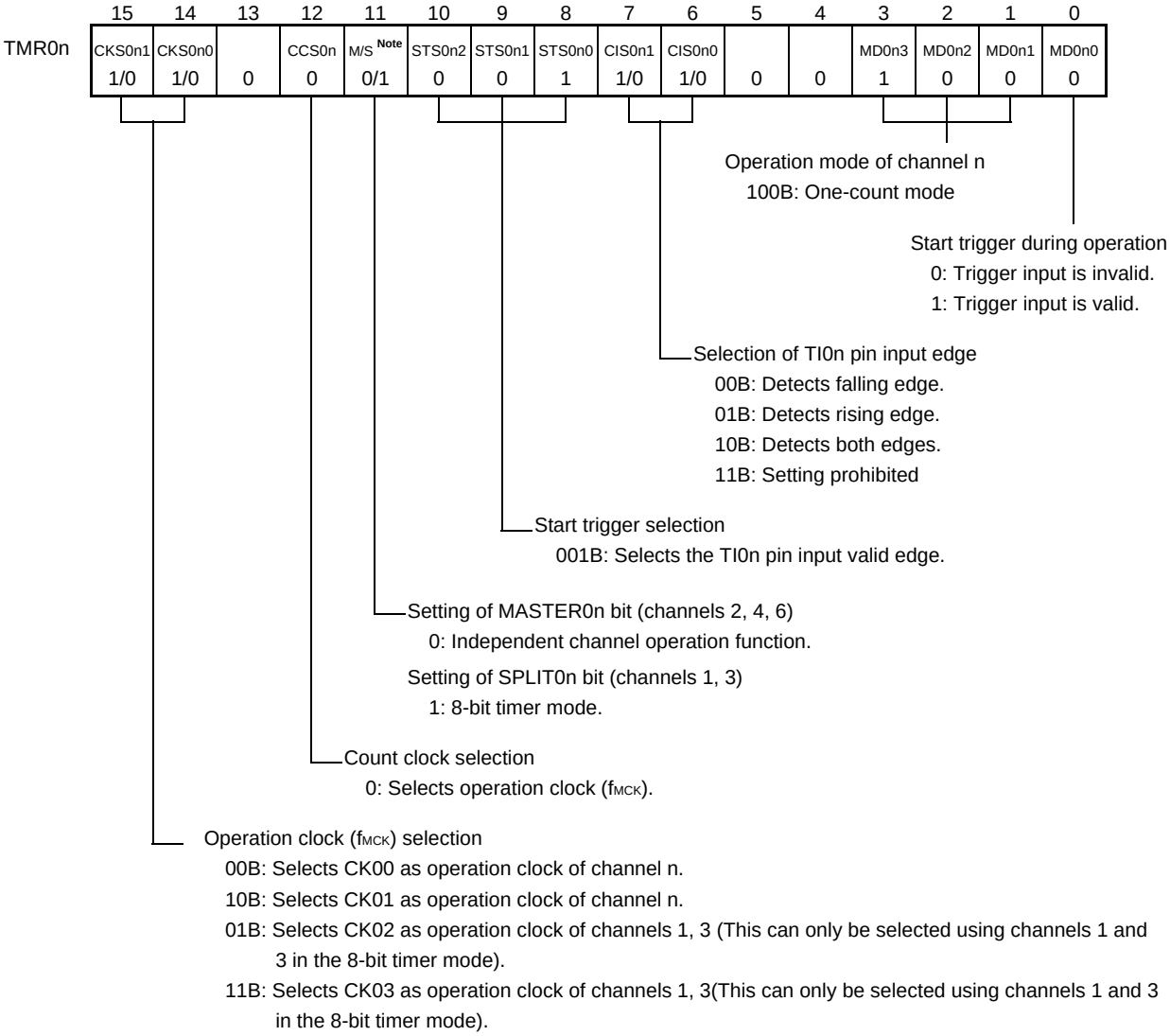


Remarks 1. n: Channel number (n = 0 to 7)

2. TS0.n: Bit n of timer channel start register 0 (TS0)
TE0.n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer/counter register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)

Figure 6-70. Example of Set Contents of Registers to Delay Counter (1/2)

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0.n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0.n 0	0: Stops the TO0n output operation by counting operation.

Note TMR00, TMR02, TMR04 to TMR07: MASTER0n bit
TMR01, TMR03: SPLIT0n bit

Remark n: Channel number (n = 0 to 7)

Figure 6-70. Example of Set Contents of Registers to Delay Counter (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0.n 0	0: Cleared to 0 when TOM0.n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0.n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 0 to 7)

Figure 6-71. Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01 (or CK02 and CK03 when using the 8-bit timer mode).	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). INTTM0n output delay is set to timer data register 0n (TDR0n). Clears the TOE0.n bit to 0 and stops operation of TO0n.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0.n (TSH01, TSH03) bit to 1. → The TS0.n (TSH01, TSH03) bit automatically returns to 0 because it is a trigger bit.	TE0.n (TEH01, TEH03) = 1, and the TI0n pin input valid edge detection wait status is set.
	Detects the TI0n pin input valid edge. →	Value of the TDR0n register is loaded to the timer/counter register 0n (TCR0n).
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used.	The counter (TCR0n) counts down. When TCR0n counts down to 0000H, INTTM0n is output, and counting stops (which leaves TCR0n at 0000H) until the next TI0n pin input.
Operation stop	The TT0.n (TTH01, TTH03) bit is set to 1. → The TT0.n (TTH01, TTH03) bit automatically returns to 0 because it is a trigger bit.	TE0.n (TEH01, TEH03) = 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed. → Operation start

Remark n: Channel number (n = 0 to 7)

6.8 Simultaneous Channel Operation Function of Timer Array Unit

6.8.1 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TI0n pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDR0n (master) + 2} × Count clock period
Pulse width = {Set value of TDR0p (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer/counter register 0n (TCR0n) of the master channel starts operating upon start trigger detection and loads the value of timer data register 0n (TDR0n).

The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next start trigger is detected.

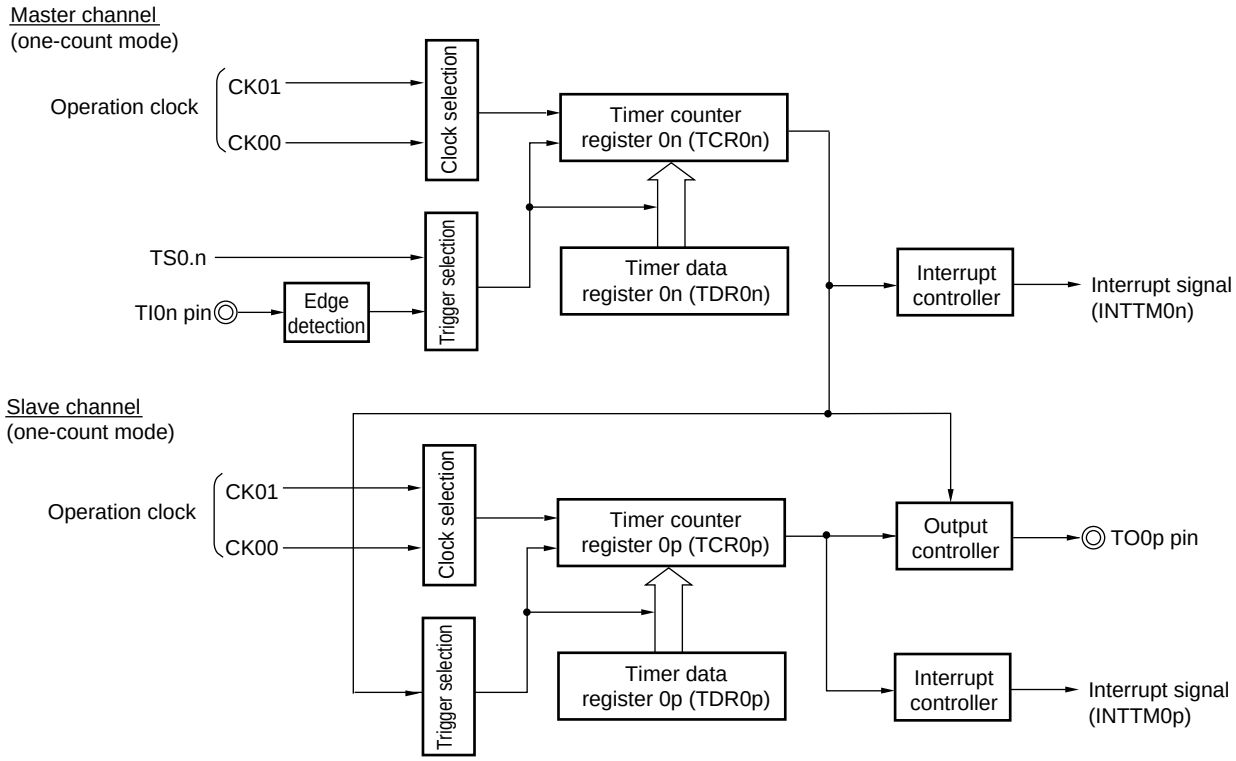
The slave channel operates in the one-count mode and counts the pulse width. The TCR0p register of the slave channel starts operation using INTTM0n of the master channel as a start trigger, and loads the value of the TDR0p register. The TCR0p register counts down from the value of The TDR0p register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTM0p and stops counting until the next start trigger (INTTM0n of the master channel) is detected. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

Instead of using the TI0n pin input, a one-shot pulse can also be output using the software operation (TS0.n = 1) as a start trigger.

Caution The timing of loading of timer data register 0n (TDR0n) of the master channel is different from that of the TDR0p register of the slave channel. f the TDR0n and TDR0p registers are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDR0n register after INTTM0n is generated and the TDR0p register after INTTM0p is generated.

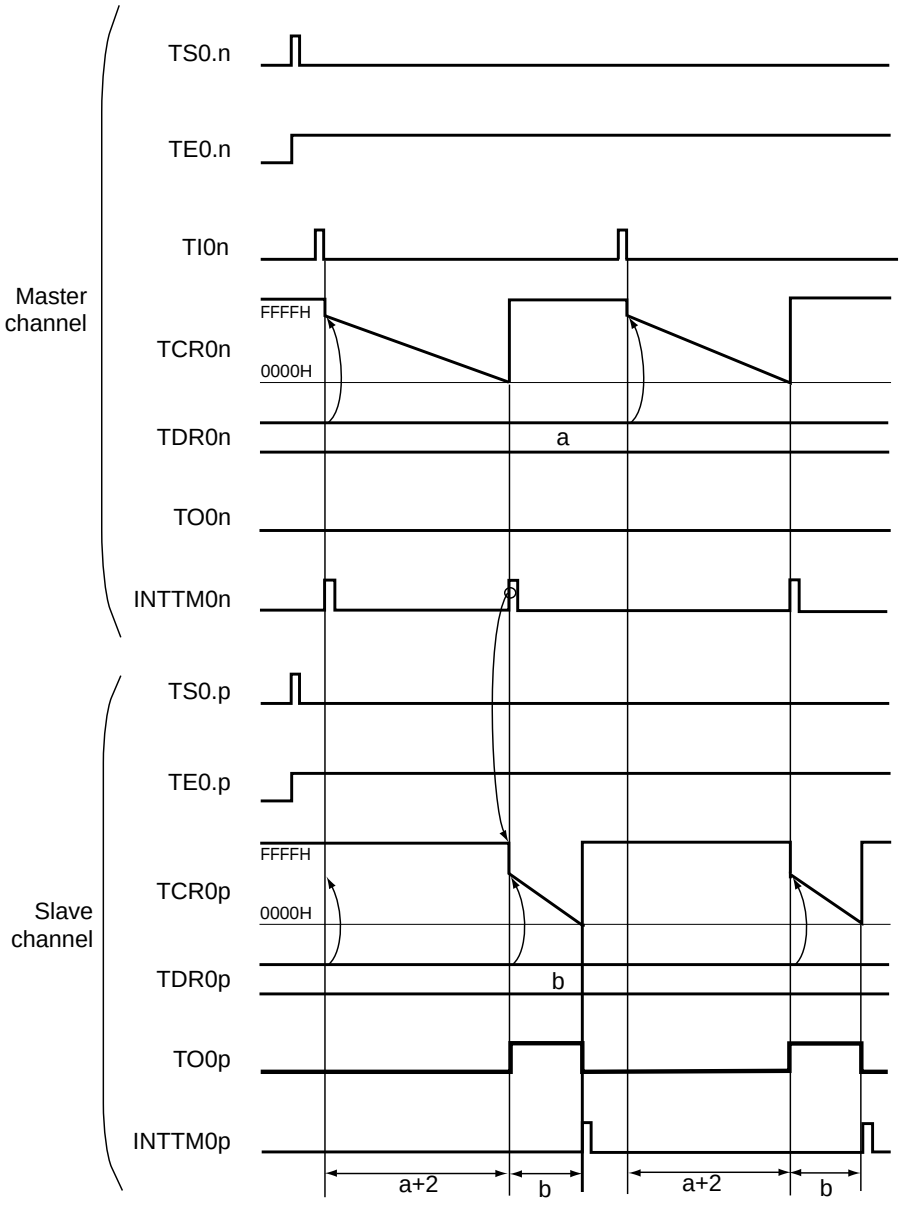
Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-72. Block Diagram of Operation as One-Shot Pulse Output Function



Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-73. Example of Basic Timing of Operation as One-Shot Pulse Output Function

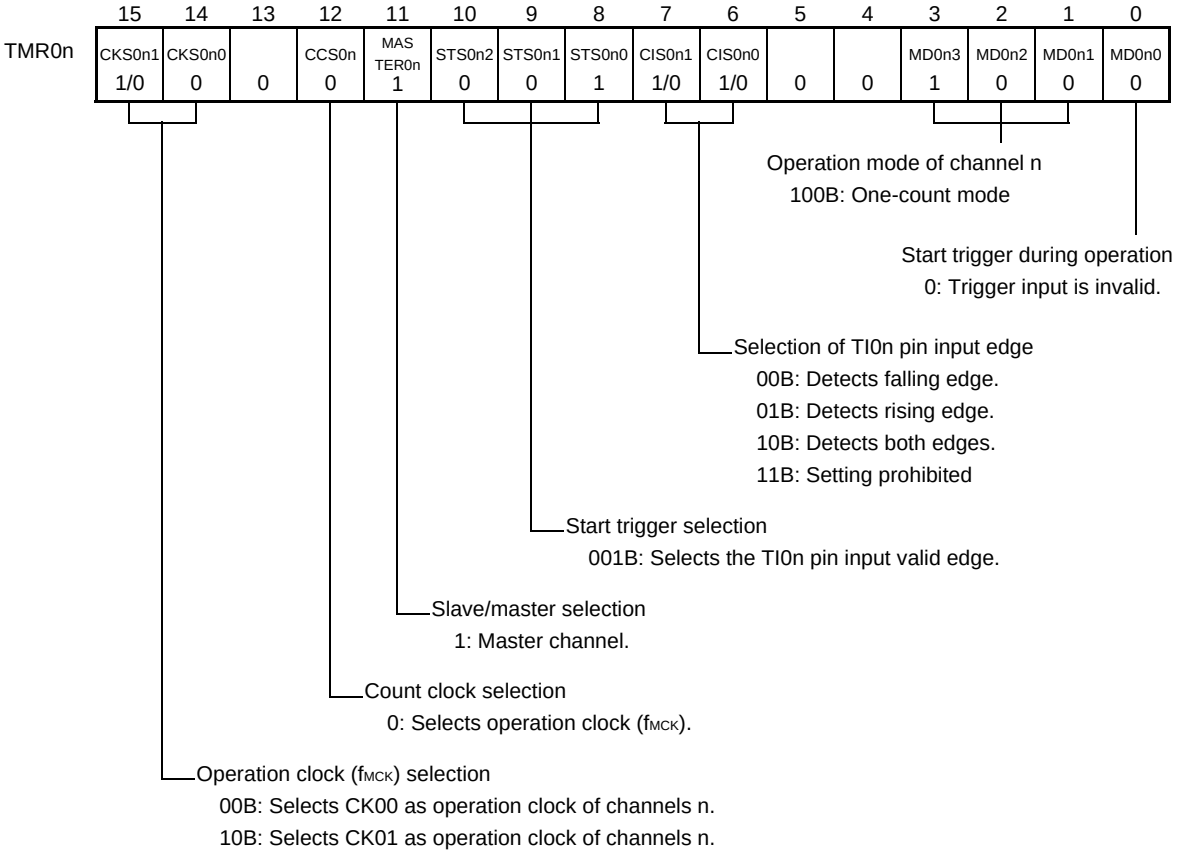


Remarks 1. n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

2. TS0.n, TS0.p: Bit n, p of timer channel start register 0 (TS0)
TE0.n, TE0.p: Bit n, p of timer channel enable status register 0 (TE0)
TI0n, TI0p: TI0n and TI0p pins input signal
TCR0n, TCR0p: Timer/counter registers 0n, 0p (TCR0n, TCR0p)
TDR0n, TDR0p: Timer data registers 0n, 0p (TDR0n, TDR0p)
TO0n, TO0p: TO0n and TO0p pins output signal

Figure 6-74. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0.n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0.n	0: Stops the TO0n output operation by counting operation.
	0	

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0.n	0: Cleared to 0 when TOM0.n = 0 (master channel output mode).
	0	

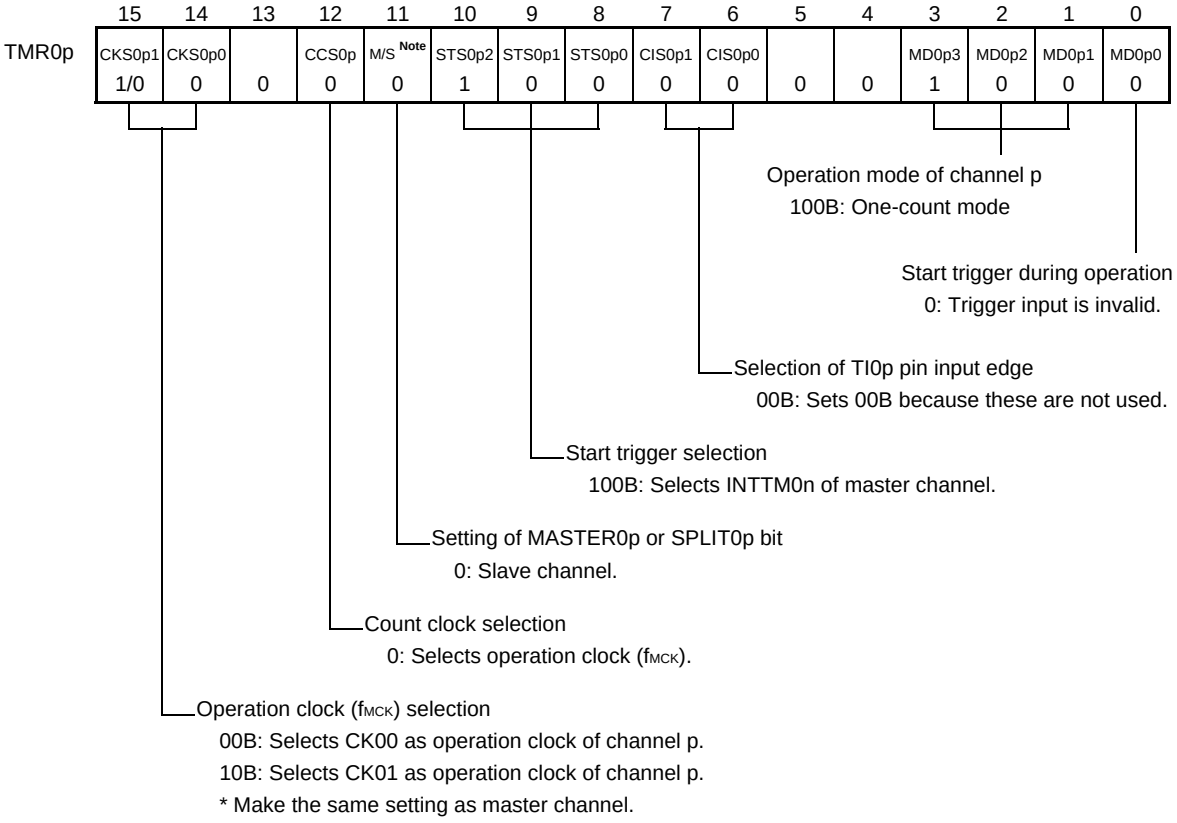
(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0.n	0: Sets master channel output mode.
	0	

Remark n: Channel number (n = 0, 2, 4, 6)

Figure 6-75. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register 0p (TMR0p)



(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0.p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0.p 1/0	0: Stops the TO0p output operation by counting operation. 1: Enables the TO0p output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0.p 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit p	
TOM0	TOM0.p 1	1: Sets the slave channel output mode.

Note TMR05, TMR07: MASTER0p bit
TMR01, TMR03: SPLIT0p bit

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-76. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable registers 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode register 0n, 0p (TMR0n, TMR0p) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register 0n (TDR0n) of the master channel, and a pulse width is set to the TDR0p register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0.p bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0.p bit. Sets the TO0.p bit and determines default level of the TO0p output. →	The TO0p pin goes into Hi-Z output state. The TO0p default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0.p bit to 1 and enables operation of TO0p. →	TO0p does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0p pin outputs the TO0p set level.

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

Figure 6-76. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOE0.p bit (slave) to 1 (only when operation is resumed). The TS0.n (master) and TS0.p (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. →	The TE0.n and TE0.p bits are set to 1 and the master channel enters the TI0n input edge detection wait status. Counter stops operating.
	The TS0.n and TS0.p bits automatically return to 0 because they are trigger bits. Detects the TI0n pin input valid edge of master channel. →	Master channel starts counting.
During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. Sets coresponting bit of noisefilter enable register 1, 2 (NFEN1, NFEN2) to 1. Set values of the TMR0p, TDR0n, TDR0p registers, TOM0.n, TOM0.p, TOL0.n, and TOL0.p bits cannot be changed. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used. Set values of the TO0 and TOE0 registers can be changed.	Master channel loads the value of the TDR0n register to timer/counter register 0n (TCR0n) when the TI0n pin valid input edge is detected, and the counter starts counting down. When the count value reaches TCR0n = 0000H, the INTTM0n output is generated, and the counter stops until the next valid edge is input to the TI0n pin. The slave channel, triggered by INTTM0n of the master channel, loads the value of the TDR0p register to the TCR0p register, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0.n (master) and TT0.p (slave) bits are set to 1 at the same time. →	TE0.n, TE0.p = 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	The TOE0.p bit of slave channel is cleared to 0 and value is set to the TO0.p bit. →	The TO0p pin outputs the TO0p set level.
TAU stop	To hold the TO0p pin output level Clears the TO0.p bit to 0 after the value to be held is set to the port register. →	The TO0p pin output level is held by port function.
	When holding the TO0p pin output level is not necessary Setting not required. →	The TO0p pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0.p bit is cleared to 0 and the TO0p pin is set to port mode.)

Operation is resumed.

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

6.8.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDR0p (slave)}/{Set value of TDR0n (master) + 1} × 100
0% output: Set value of TDR0p (slave) = 0000H
100% output: Set value of TDR0p (slave) ≥ {Set value of TDR0n (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDR0p (slave) > (set value of TDR0n (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TS0.n) of timer channel start register 0 (TS0) is set to 1, an interrupt (INTTM0n) is output, the value set to timer data register 0n (TDR0n) is loaded to timer/counter register 0n (TCR0n), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTM0n is output, the value of the TDR0n register is loaded again to the TCR0n register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TT0.n) of timer channel stop register 0 (TT0) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TO0p) cycle.

The slave channel operates in one-count mode. By using INTTM0n from the master channel as a start trigger, the TCR0p register loads the value of the TDR0p register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTM0p and waits until the next start trigger (INTTM0n from the master channel) is generated.

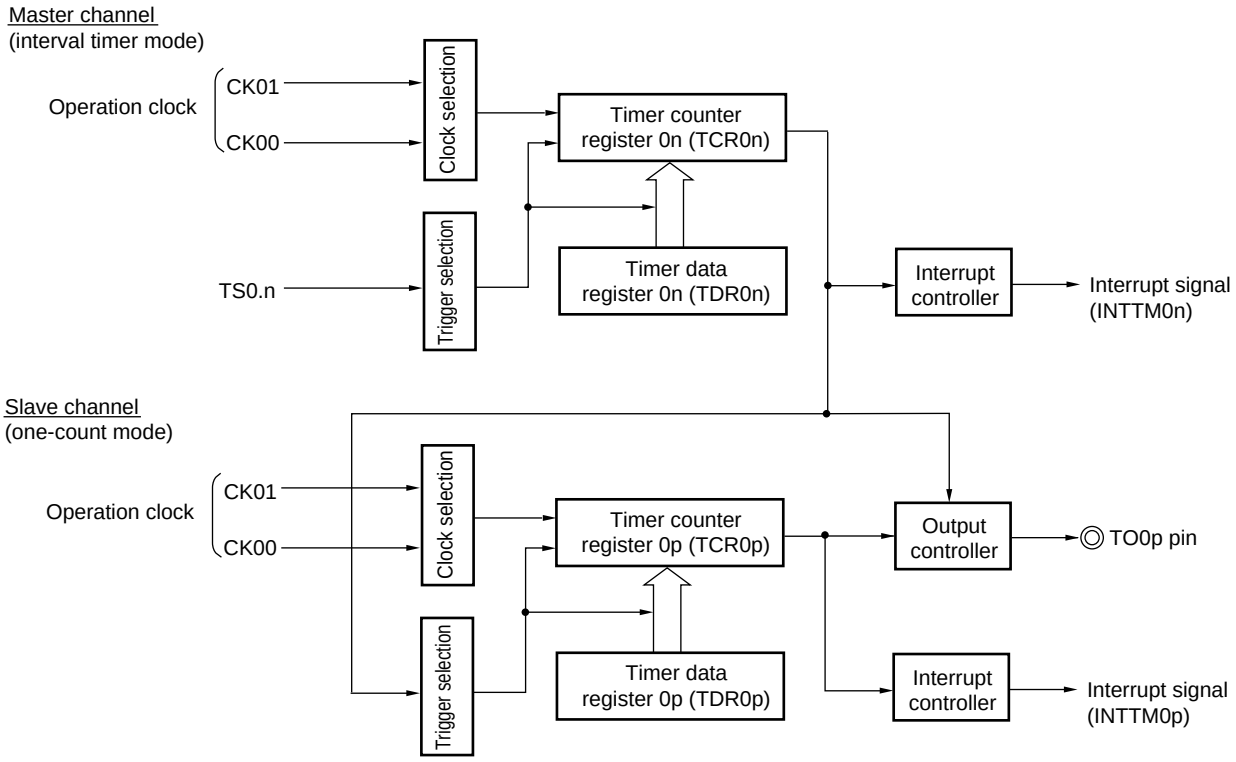
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TO0p) duty.

PWM output (TO0p) goes to the active level one clock after the master channel generates INTTM0n and goes to the inactive level when the TCR0p register of the slave channel becomes 0000H.

Caution To rewrite both timer data register 0n (TDR0n) of the master channel and the TDR0p register of the slave channel, a write access is necessary two times. The timing at which the values of the TDR0n and TDR0p registers are loaded to the TCR0n and TCR0p registers is upon occurrence of INTTM0n of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTM0n of the master channel, the TO0p pin cannot output the expected waveform. To rewrite both the TDR0n register of the master and the TDR0p register of the slave, therefore, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel.

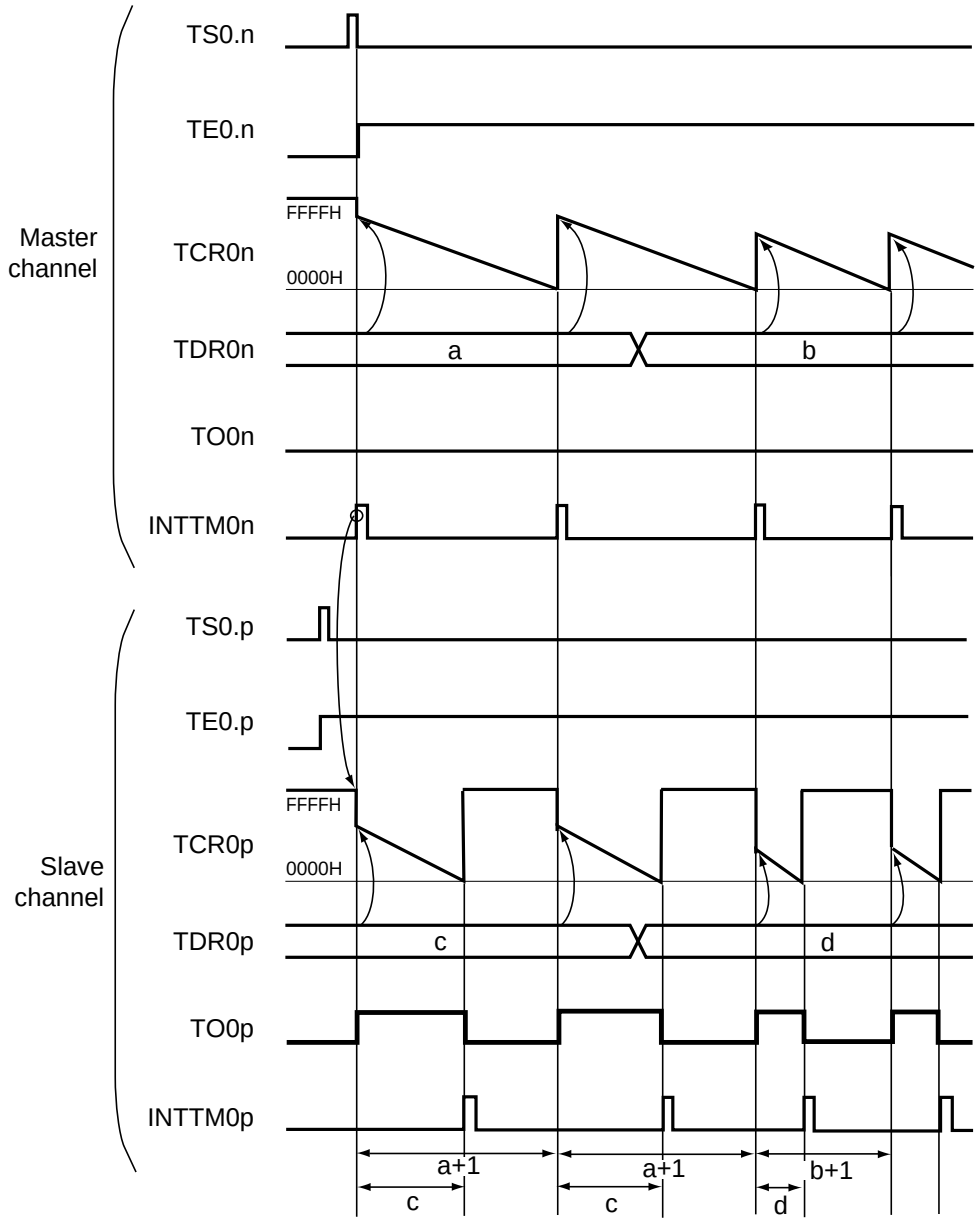
Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-77. Block Diagram of Operation as PWM Function



Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-78. Example of Basic Timing of Operation as PWM Function

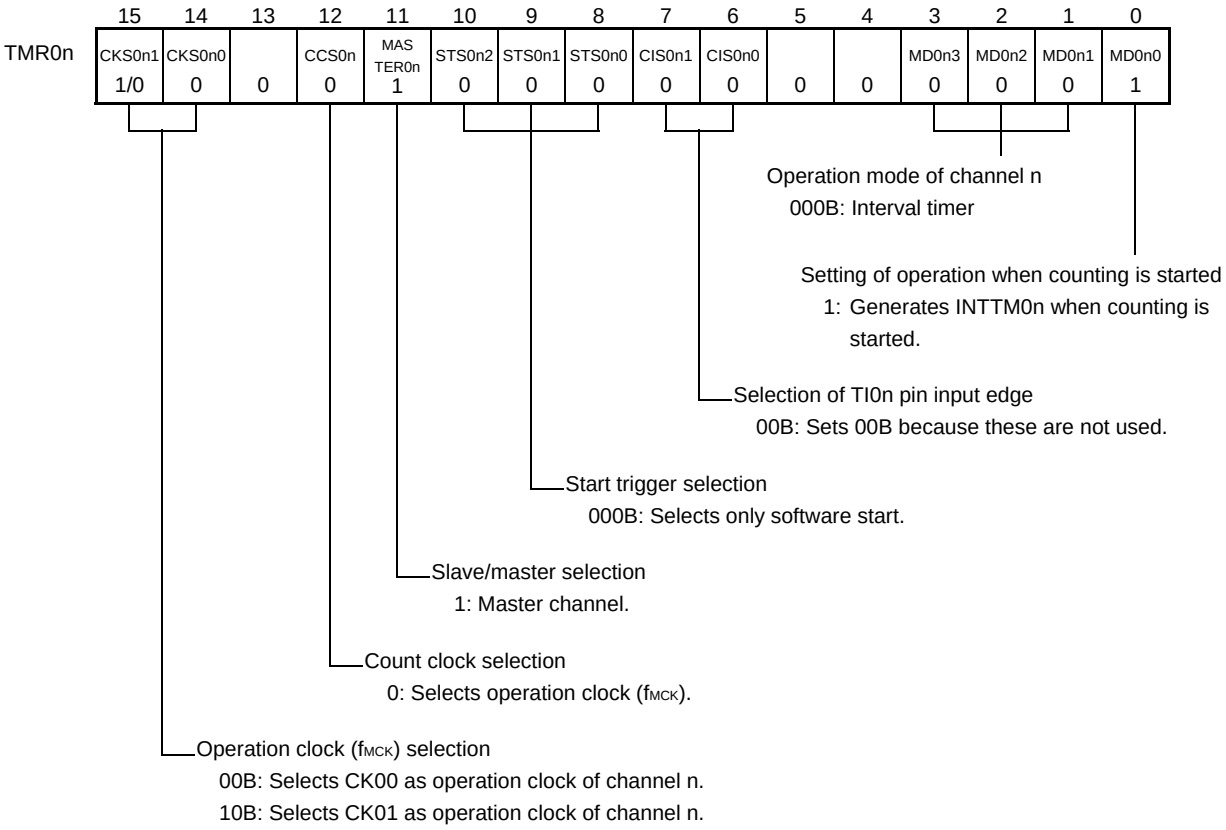


Remark 1. n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

2. TS0.n, TS0.p: Bit n, p of timer channel start register 0 (TS0)
TE0.n, TE0.p: Bit n, p of timer channel enable status register 0 (TE0)
TCR0n, TCR0p: Timer/counter registers 0n, 0p (TCR0n, TCR0p)
TDR0n, TDR0p: Timer data registers 0n, 0p (TDR0n, TDR0p)
TO0n, TO0p: TO0n and TO0p pins output signal

Figure 6-79. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0.n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0.n	0: Stops the TO0n output operation by counting operation.
	0	

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0.n	0: Cleared to 0 when TOM0.n = 0 (master channel output mode).
	0	

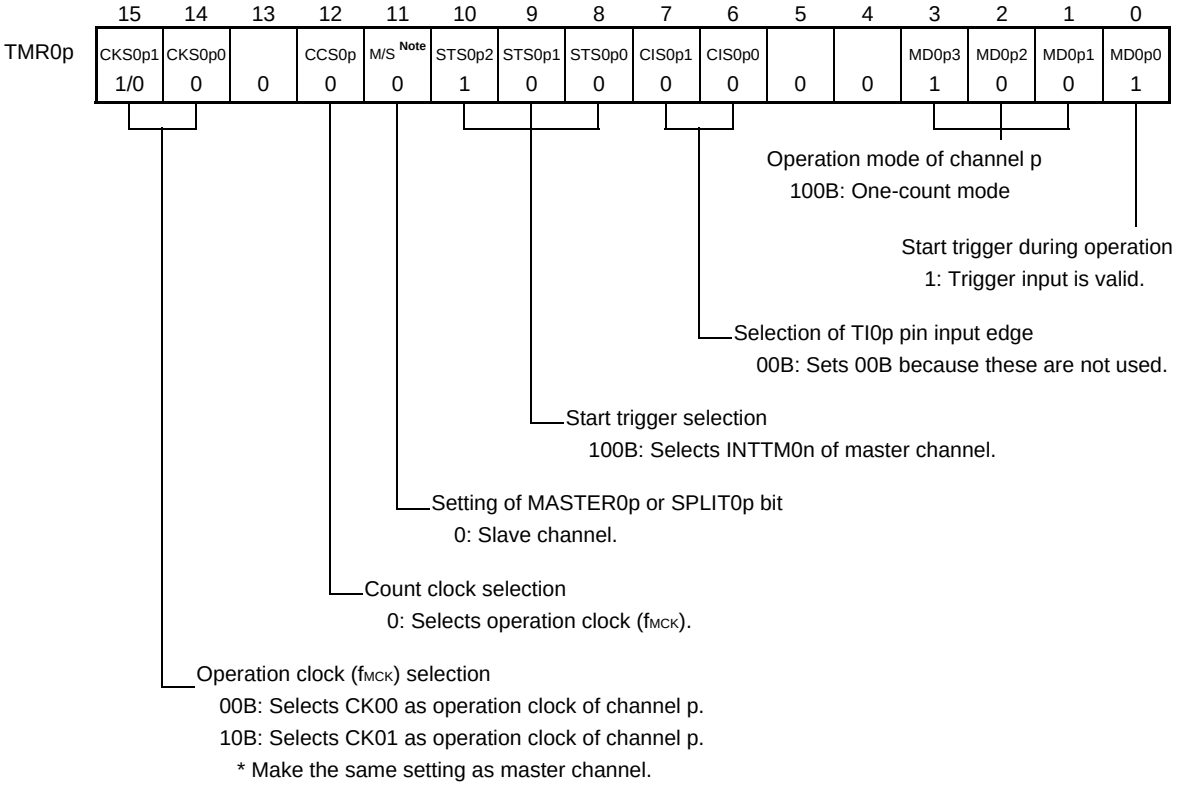
(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0.n	0: Sets master channel output mode.
	0	

Remark n: Channel number (n = 0, 2, 4, 6)

Figure 6-80. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register 0p (TMR0p)



(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0.p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0.p 1/0	0: Stops the TO0p output operation by counting operation. 1: Enables the TO0p output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0.p 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit p	
TOM0	TOM0.p 1	1: Sets the slave channel output mode.

Note TMR05, TMR07: MASTER0p bit
TMR01, TMR03: SPLIT0p bit

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-81. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode registers 0n, 0p (TMR0n, TMR0p) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register 0n (TDR0n) of the master channel, and a duty factor is set to the TDR0p register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0.p bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0.p bit. Sets the TO0.p bit and determines default level of the TO0p output. →	The TO0p pin goes into Hi-Z output state. The TO0p default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0.p bit to 1 and enables operation of TO0p. →	TO0p does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0p pin outputs the TO0p set level.

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

Figure 6-81. Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOE0.p bit (slave) to 1 (only when operation is resumed). The TS0.n (master) and TS0.p (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. The TS0.n and TS0.p bits automatically return to 0 because they are trigger bits.	TE0.n = 1, TE0.p = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMR0n and TMR0p registers, TOM0.n, TOM0.p, TOL0.n, and TOL0.p bits cannot be changed. Set values of the TDR0n and TDR0p registers can be changed after INTTM0n of the master channel is generated. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used. Set values of the TO0 and TOE0 registers can be changed.	The counter of the master channel loads the TDR0n register value to timer/counter register 0n (TCR0n), and counts down. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to the TCR0n register, and the counter starts counting down again. At the slave channel, the value of the TDR0p register is loaded to the TCR0p register, triggered by INTTM0n of the master channel, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0.n (master) and TT0.p (slave) bits are set to 1 at the same time. The TT0.n and TT0.p bits automatically return to 0 because they are trigger bits.	TE0.n, TE0.p = 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	The TOE0.p bit of slave channel is cleared to 0 and value is set to the TO0.p bit.	The TO0p pin outputs the TO0p set level.
TAU stop	To hold the TO0p pin output level Clears the TO0.p bit to 0 after the value to be held is set to the port register. When holding the TO0p pin output level is not necessary Setting not required.	The TO0p pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0.p bit is cleared to 0 and the TO0p pin is set to port mode.)

Remark n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

6.8.3 Operation as multiple PWM output function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDR0p (slave 1)}/{Set value of TDR0n (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDR0q (slave 2)}/{Set value of TDR0n (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDR0p (slave 1) > {set value of TDR0n (master) + 1} or if the {set value of TDR0q (slave 2)} > {set value of TDR0n (master) + 1}, it is summarized into 100% output.

Timer/counter register 0n (TCR0n) of the master channel operates in the interval timer mode and counts the periods.

The TCR0p register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0p pin. The TCR0p register loads the value of timer data register 0p (TDR0p), using INTTM0n of the master channel as a start trigger, and starts counting down. When TCR0p = 0000H, TCR0p outputs INTTM0p and stops counting until the next start trigger (INTTM0n of the master channel) has been input. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

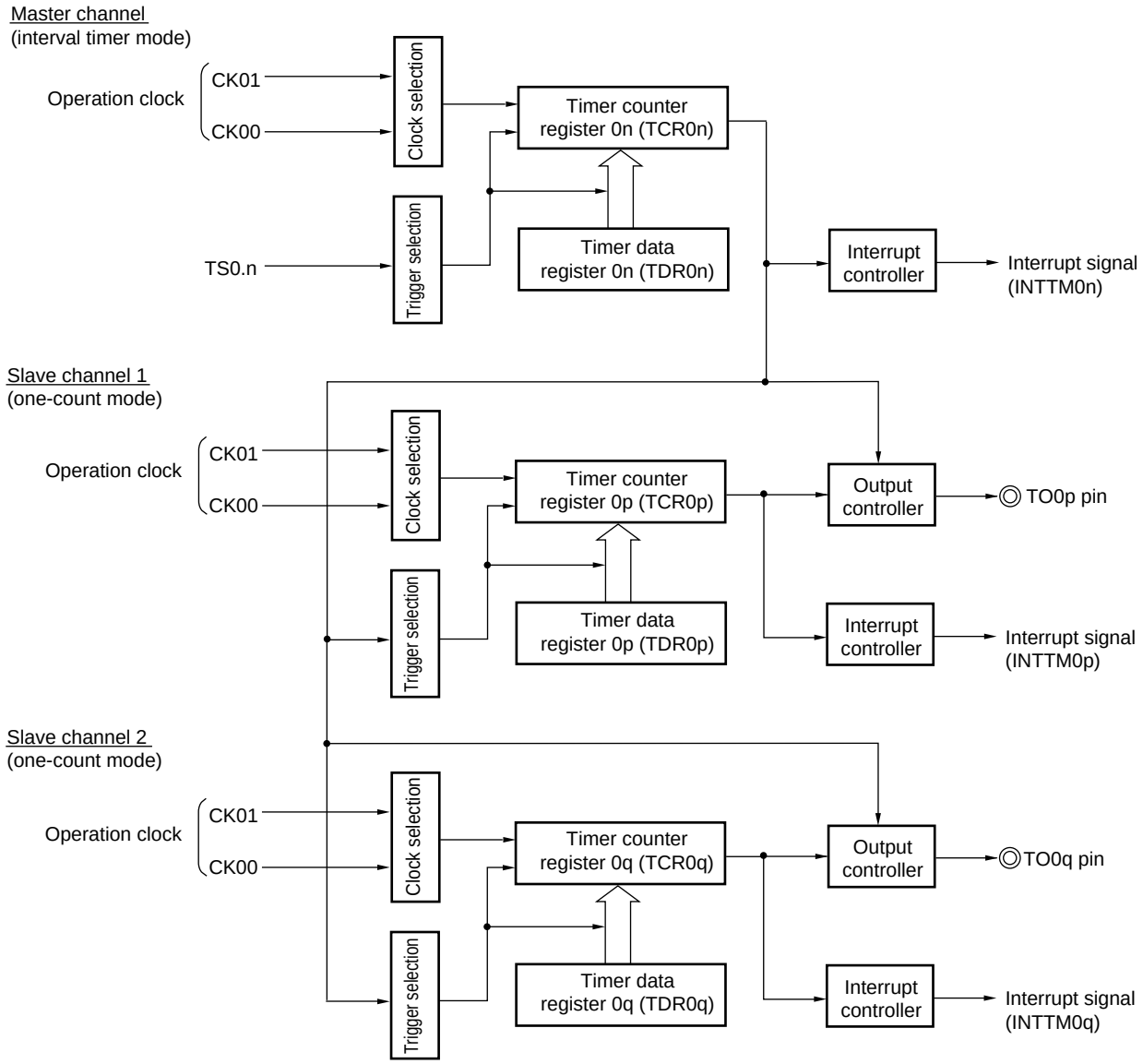
In the same way as the TCR0p register of the slave channel 1, the TCR0q register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0q pin. The TCR0q register loads the value of the TDR0q register, using INTTM0n of the master channel as a start trigger, and starts counting down. When TCR0q = 0000H, the TCR0q register outputs INTTM0q and stops counting until the next start trigger (INTTM0n of the master channel) has been input. The output level of TO0q becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0q = 0000H.

When channel 0 is used as the master channel as above, up to seven types of PWM signals can be output at the same time.

Caution To rewrite both timer data register 0n (TDR0n) of the master channel and the TDR0p register of the slave channel 1, write access is necessary at least twice. Since the values of the TDR0n and TDR0p registers are loaded to the TCR0n and TCR0p registers after INTTM0n is generated from the master channel, if rewriting is performed separately before and after generation of INTTM0n from the master channel, the TO0p pin cannot output the expected waveform. To rewrite both the TDR0n register of the master and the TDR0p register of the slave, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel (This applies also to the TDR0q register of the slave channel 2).

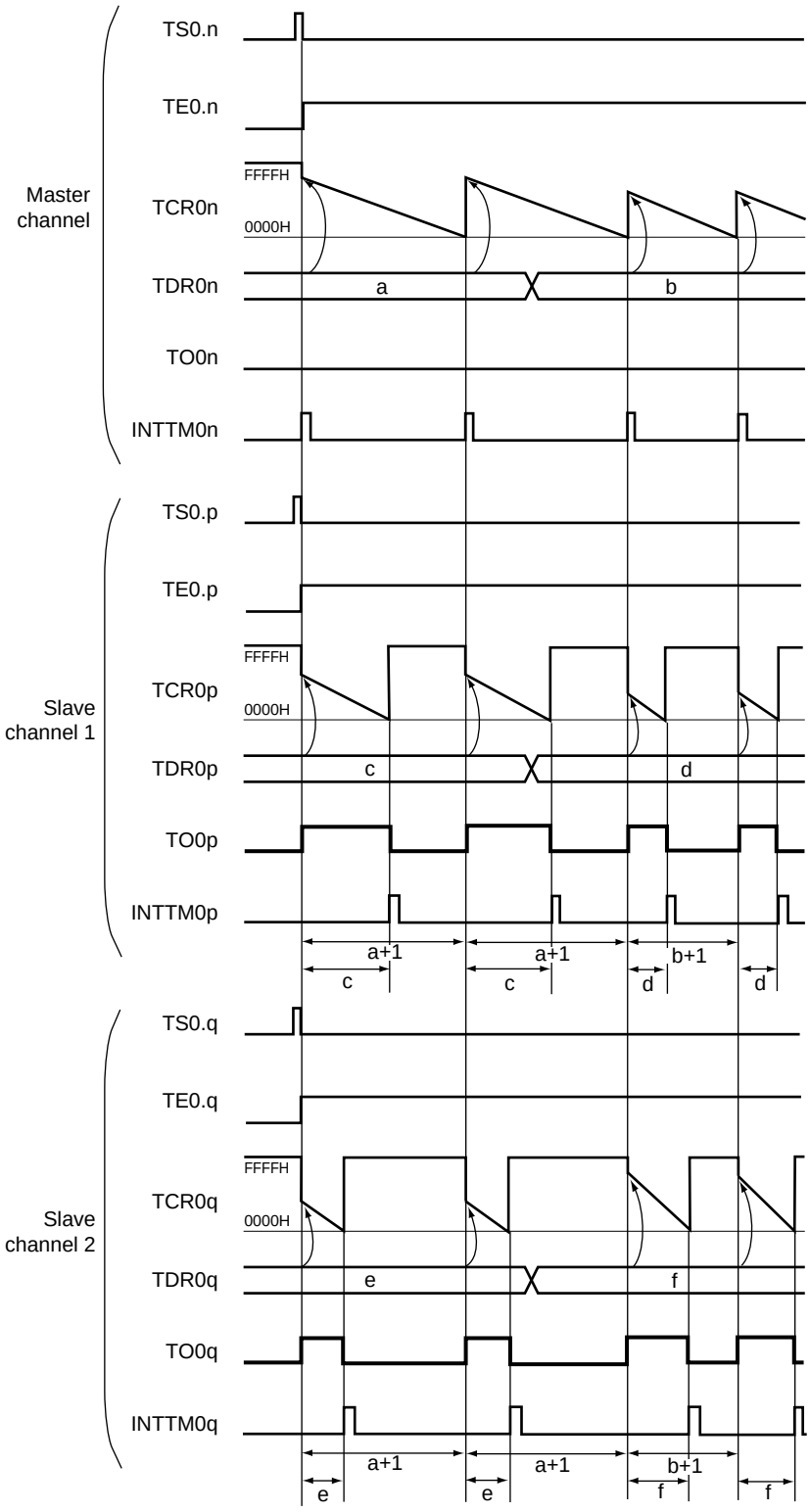
Remark n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6-82. Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)



Remark n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6-83. Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs) (1/2)



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

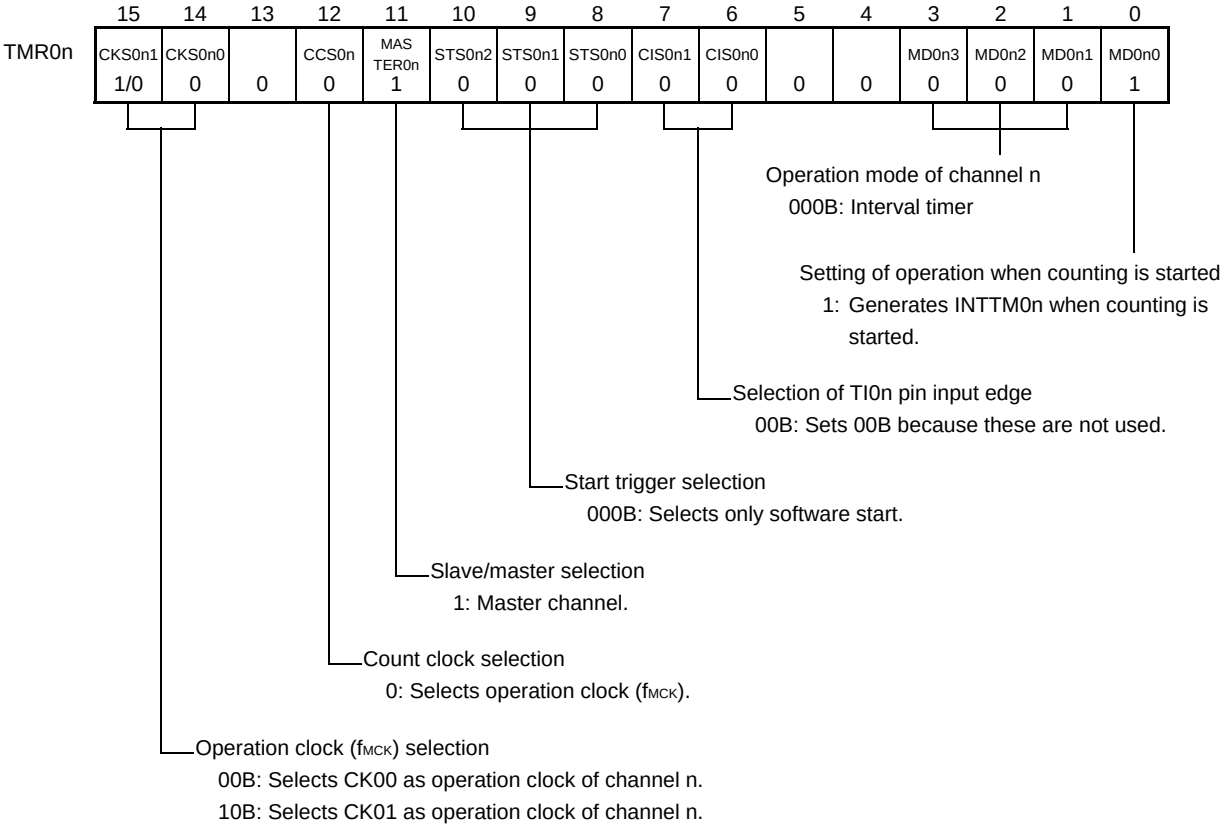
Figure 6-83. Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs) (2/2)

Remark 1. n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

2. TS0.n, TS0.p, TS0.q: Bit n, p, q of timer channel start register 0 (TS0)
TE0.n, TE0.p, TE0.q: Bit n, p, q of timer channel enable status register 0 (TE0)
TCR0n, TCR0p, TCR0q: Timer/counter registers 0n, 0p, 0q (TCR0n, TCR0p, TCR0q)
TDR0n, TDR0p, TDR0q: Timer data registers 0n, 0p, 0q (TDR0n, TDR0p, TDR0q)
TO0n, TO0p, TO0q: TO0n, TO0p, and TO0q pins output signal

Figure 6-84. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0.n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0.n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0.n 0	0: Cleared to 0 when TOM0.n = 0 (master channel output mode).

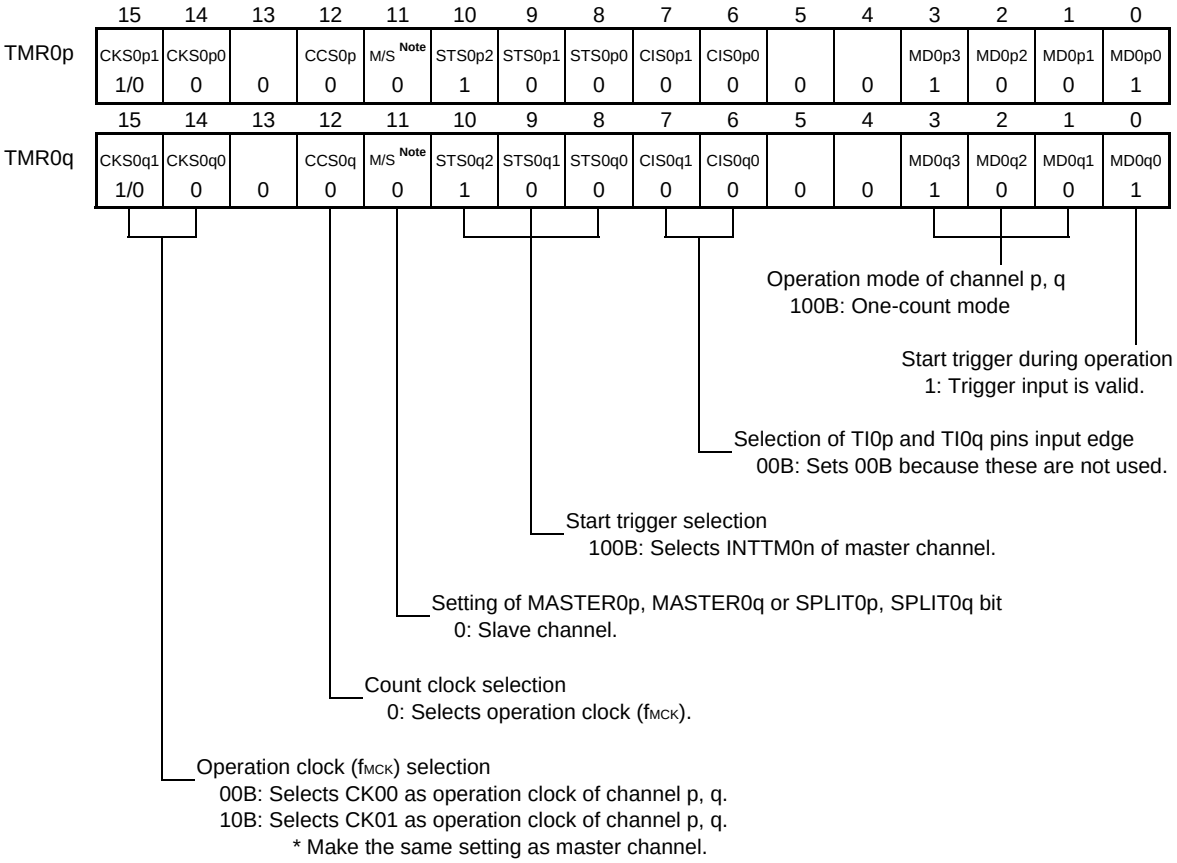
(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0.n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 0, 2, 4)

Figure 6-85. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register 0p, 0q (TMR0p, TMR0q)



(b) Timer output register 0 (TO0)

	Bit q	Bit p	
TO0	TO0.q	TO0.p	0: Outputs 0 from TO0p or TO0q. 1: Outputs 1 from TO0p or TO0q.
	1/0	1/0	

(c) Timer output enable register 0 (TOE0)

	Bit q	Bit p	
TOE0	TOE0.q	TOE0.p	0: Stops the TO0p or TO0q output operation by counting operation. 1: Enables the TO0p or TO0q output operation by counting operation.
	1/0	1/0	

(d) Timer output level register 0 (TOL0)

	Bit q	Bit p	
TOL0	TOL0.q	TOL0.p	0: Positive logic output (active-high) 1: Inverted output (active-low)
	1/0	1/0	

(e) Timer output mode register 0 (TOM0)

	Bit q	Bit p	
TOM0	TOM0.q	TOM0.p	1: Sets the slave channel output mode.
	1	1	

Note TMR05, TMR07: MASTER0p, MASTER0q bit
TMR01, TMR03: SPLIT0p, SPLIT0q bit

Remark n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6-86. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode registers 0n, 0p, 0q (TMR0n, TMR0p, TMR0q) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register 0n (TDR0n) of the master channel, and a duty factor is set to the TDR0p and TDR0q registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOM0.p and TOM0.q bits of timer output mode register 0 (TOM0) are set to 1 (slave channel output mode). Clears the TOL0.p and TOL0.q bits to 0. Sets the TO0.p and TO0.q bits and determines default level of the TO0p and TO0q outputs. →	The TO0p and TO0q pins go into Hi-Z output state. The TO0p and TO0q default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0.p and TOE0.q bits to 1 and enables operation of TO0p and TO0q. →	TO0p and TO0q do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TO0p and TO0q pins output the TO0p and TO0q set levels.

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

Figure 6-86. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	(Sets the TOE0.p and TOE0.q (slave) bits to 1 only when resuming operation.) The TS0.n bit (master), and TS0.p and TS0.q (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. The TS0.n, TS0.p, and TS0.q bits automatically return to 0 because they are trigger bits.	TE0.n = 1, TE0.p, TE0.q = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMR0n, TMR0p, TMR0q registers, TOM0.n, TOM0.p, TOM0.q, TOL0.n, TOL0.p, and TOL0.q bits cannot be changed. Set values of the TDR0n, TDR0p, and TDR0q registers can be changed after INTTM0n of the master channel is generated. The TCR0n, TCR0p, and TCR0q registers can always be read. The TSR0n, TSR0p, and TSR0q registers are not used. Set values of the TO0 and TOE0 registers can be changed.	The counter of the master channel loads the TDR0n register value to timer/counter register 0n (TCR0n) and counts down. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to the TCR0n register, and the counter starts counting down again. At the slave channel 1, the values of the TDR0p register are transferred to the TCR0p register, triggered by INTTM0n of the master channel, and the counter starts counting down. The output levels of TO0p become active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDR0q register are transferred to TCR0q regster, triggered by INTTM0n of the master channel, and the counter starts counting down. The output levels of TO0q become active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0q = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0.n bit (master), TT0.p, and TT0.q (slave) bits are set to 1 at the same time. The TT0.n, TT0.p, and TT0.q bits automatically return to 0 because they are trigger bits.	TE0.n, TE0.p, TE0.q = 0, and count operation stops. The TCR0n, TCR0p, and TCR0q registers hold count value and stop. The TO0p and TO0q output are not initialized but hold current status.
	The TOE0.p and TOE0.q bits of slave channels are cleared to 0 and value is set to the TO0.p and TO0.q bits.	The TO0p and TO0q pins output the TO0p and TO0q set levels.
TAU stop	To hold the TO0p and TO0q pin output levels Clears the TO0.p and TO0.q bits to 0 after the value to be held is set to the port register. When holding the TO0p and TO0q pin output levels are not necessary Setting not required.	The TO0p and TO0q pin output levels are held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0.p and TO0.q bits are cleared to 0 and the TO0p and TO0q pins are set to port mode.)

Operation is resumed.

Remark n: Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

CHAPTER 7 REAL-TIME CLOCK

7.1 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)
- Pin output function of 1 Hz (48, 64-pin products only)

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 on-chip oscillator clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available. The 20- to 32-pin products have the constant-period interrupt function only, because these products have no subsystem clock.
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}.

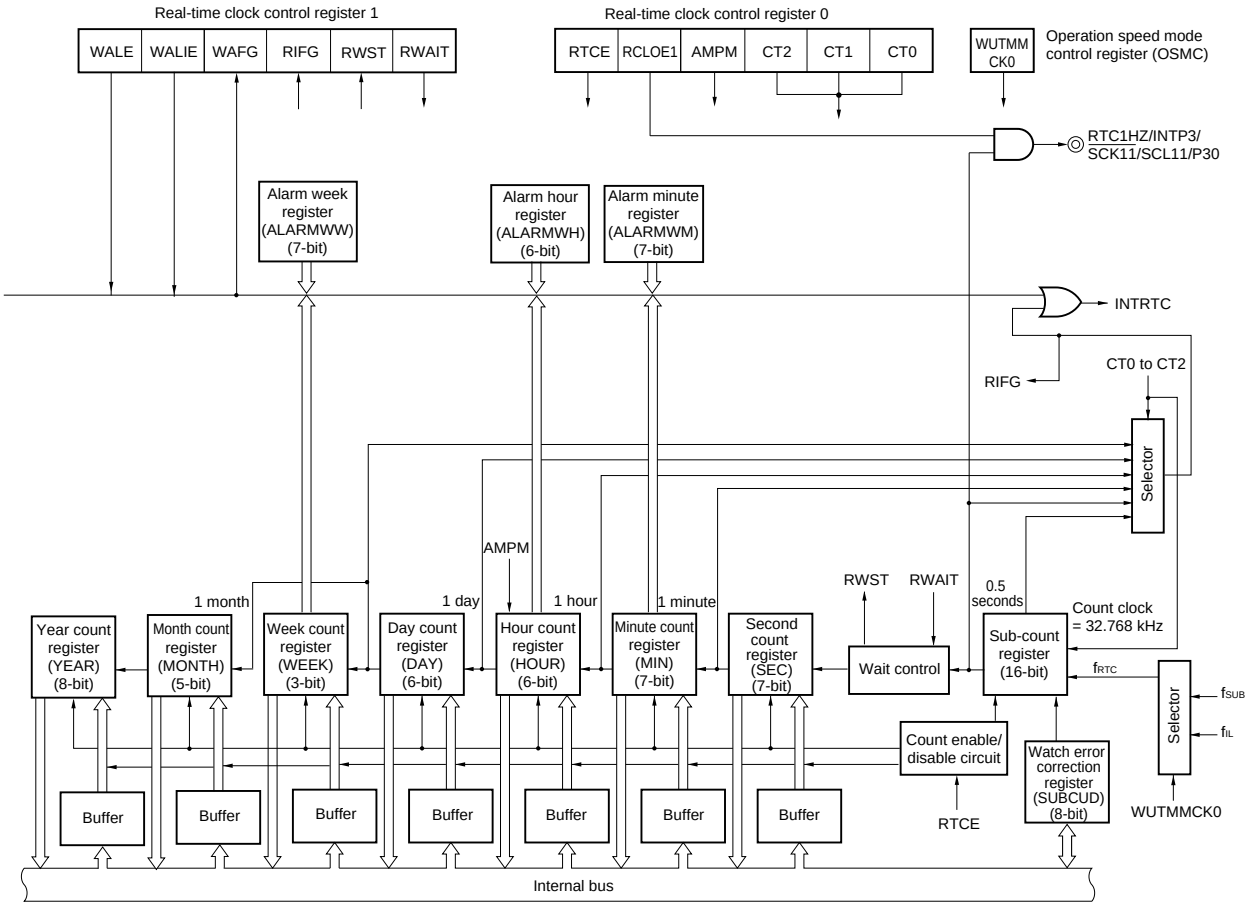
7.2 Configuration of Real-time Clock

The real-time clock includes the following hardware.

Table 7-1. Configuration of Real-time Clock

Item	Configuration
Counter	Sub-count register
Control registers	Peripheral enable register 0 (PER0)
	Operation speed 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 7-1. 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 on-chip oscillator clock (f_{IL}) is selected, only the constant-period interrupt function is available. The 20- to 32-pin products have the constant-period interrupt function only, because these products have no subsystem clock.
However, the constant-period when f_{IL} is selected will be the value of nearly twice selection when f_{SUB} is selected.

7.3 Registers Controlling Real-time Clock

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

- Peripheral enable register 0 (PER0)
- Operation speed 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)

(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 7-2. 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	0	ADCEN	IICA0EN Note 1	SAU1EN Note 1	SAU0EN	0	TAU0EN

RTCEN	Control of real-time clock (RTC) input clock supply Note 2
0	Stops input clock supply. • SFR used by the real-time clock (RTC) cannot be written. • The real-time clock (RTC) is in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) can be read/written.

Notes 1. Those are not provided in the 20-pin products.

2. The RTCEN bit is used to supply or stop the clock used when accessing the real-time clock (RTC) register from the CPU. The RTCEN bit cannot control supply of the operating clock (f_{RTC}) to RTC.

Cautions 1. When using the real-time clock, first set the RTCEN bit to 1, while oscillation of the input clock (f_{RTC}) is stable. If RTCEN = 0, writing to a control register of the real-time clock is ignored, and, even if the register is read, only the default value is read.

2. Clock supply to peripheral functions other than the real-time clock can be stopped in STOP mode or HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the operation speed mode control register (OSMC) to 1. In this case, set the RTCEN bit of the PER0 register to 1 and the other bits (bits 0 to 6) to 0.

3. Be sure to clear the following bits to 0.

20-pin products: bits 1, 3, 4, 6

30, 32-pin products: bits 1, 6

48, 64-pin products: bits 1, 6

(2) Operation speed mode control register (OSMC)

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

In addition, by stopping clock functions that are even a little unnecessary, the RTCLPC bit can be used to reduce power consumption. For details about setting the RTCLPC bit, see **CHAPTER 5 CLOCK GENERATOR**.

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

Reset signal generation clears this register to 00H.

Figure 7-3. Format of Operation Speed 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 and interval timer.
0	Subsystem clock (f_{SUB})
1	Low-speed on-chip oscillator clock (f_{IL})

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 on-chip oscillator clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available. The 20- to 32-pin products have the constant-period interrupt function only, because these products have no subsystem clock.

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}.

(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, control the RTC1HZ pin, 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 7-4. 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	RCLOE1	0	AMPM	CT2	CT1	CT0

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

RCLOE1	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz).
1	Enables output of the RTC1HZ pin (1 Hz).

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 7-2 shows the displayed time digits that are displayed.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period 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.			

Caution Do not change the value of the RTCLOE1 bit when RTCE = 1.

Remark ×: don't care

(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 7-5. 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 clock (32.768 kHz) after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

Figure 7-5. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)

RIFG	Constant-period interrupt status flag
0	Constant-period interrupt is not generated.
1	Constant-period interrupt is generated.
This flag indicates the status of generation of the constant-period interrupt. When the constant-period 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 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 1 clock (f_{RTC}) until the counter value can be read or written (RWST = 1). When the 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 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.

Remark 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.

(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 sub-counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (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 7-6. 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 changing the values of the SEC register while the counter operates (RTCE = 1), rewrite the values of the SEC register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the SEC register.

(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 2 clocks (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 7-7. 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 changing the values of the MIN 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.

(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 2 clocks (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 7-8. 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

Cautions 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

2. If change the value of the AMPM bit, reconfigure to the value of HOUR register.

3. When changing the values of the HOUR register while the counter operates (RTCE = 1), rewrite the values of the HOUR register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the HOUR register.

Table 7-2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 7-2. Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 1)	
Time	HOUR Register	Time	HOUR Register
0	00H	0 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	0 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

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.

(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 2 clocks (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 7-9. 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 changing the values of the DAY register while the counter operates (RTCE = 1), rewrite the values of the DAY register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the DAY register.

(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 2 clocks (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 7-10. 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

Cautions 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	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

2. When changing the values of the WEEK register while the counter operates (RTCE = 1), rewrite the values of the WEEK register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the WEEK register.

(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 2 clocks (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 7-11. 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 changing the values of the MONTH register while the counter operates (RTCE = 1), rewrite the values of the MONTH register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the MONTH register.

(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 2 clocks (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 7-12. 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 changing the values of the YEAR register while the counter operates (RTCE = 1), rewrite the values of the YEAR register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the YEAR register.

(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 sub-count register 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 7-13. 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.

(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 7-14. 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

(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 (bit 3 of the RTCC0 register) 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 7-15. 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).

(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 7-16. 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

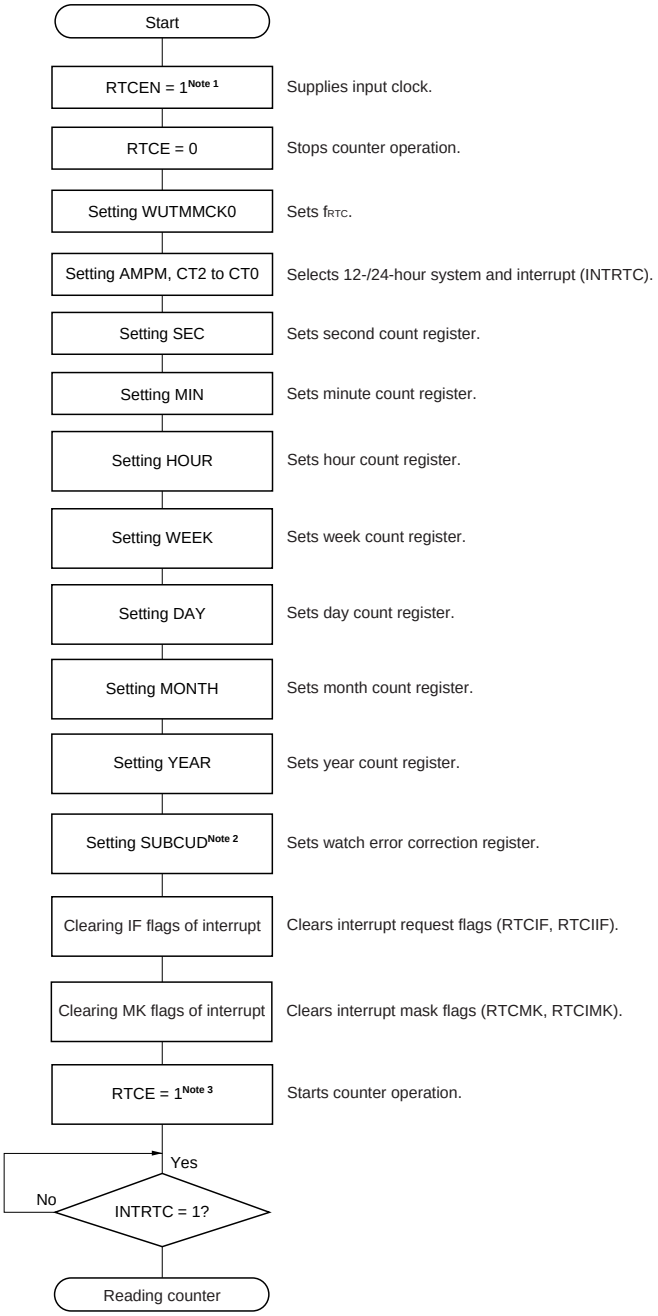
Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 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

7.4 Real-time Clock Operation

7.4.1 Starting operation of real-time clock

Figure 7-17. Procedure for Starting Operation of Real-time Clock



Notes 1. First set the RTCEN bit to 1, while oscillation of the input clock (f_{RTC}) is stable.

2. Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **7.4.6 Example of watch error correction of real-time clock**.

3. Confirm the procedure described in **7.4.2 Shifting to STOP mode after starting operation** when shifting to STOP mode without waiting for INTRTC = 1 after RTCE = 1.

4. Set RWAIT to 0 after completion of reading or writing the counter.

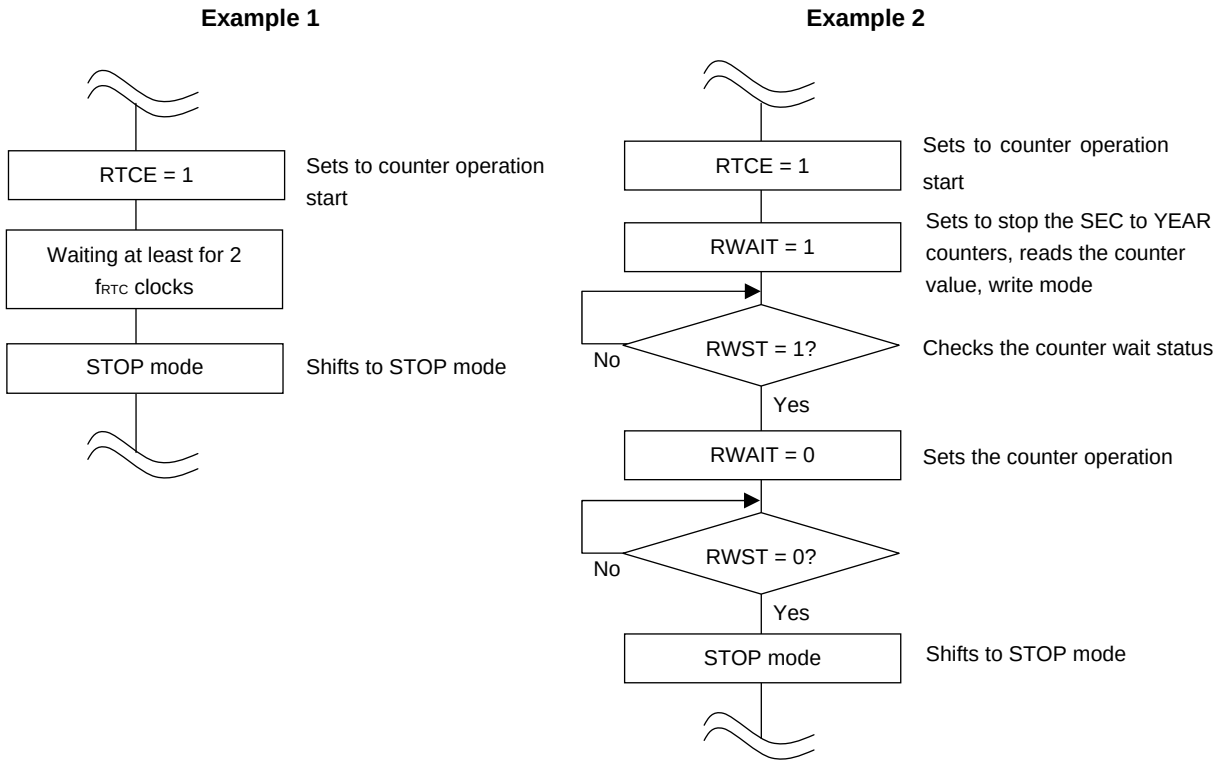
7.4.2 Shifting to STOP mode after starting operation

Perform one of the following processing when shifting to 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 STOP mode after the first INTRTC interrupt has occurred.

- Shifting to STOP mode when at least two input clocks (f_{RTC}) have elapsed after setting the RTCE bit to 1 (see **Figure 7-18**, **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 STOP mode after checking again by polling that the RWST bit has become 0 (see **Figure 7-18**, **Example 2**).

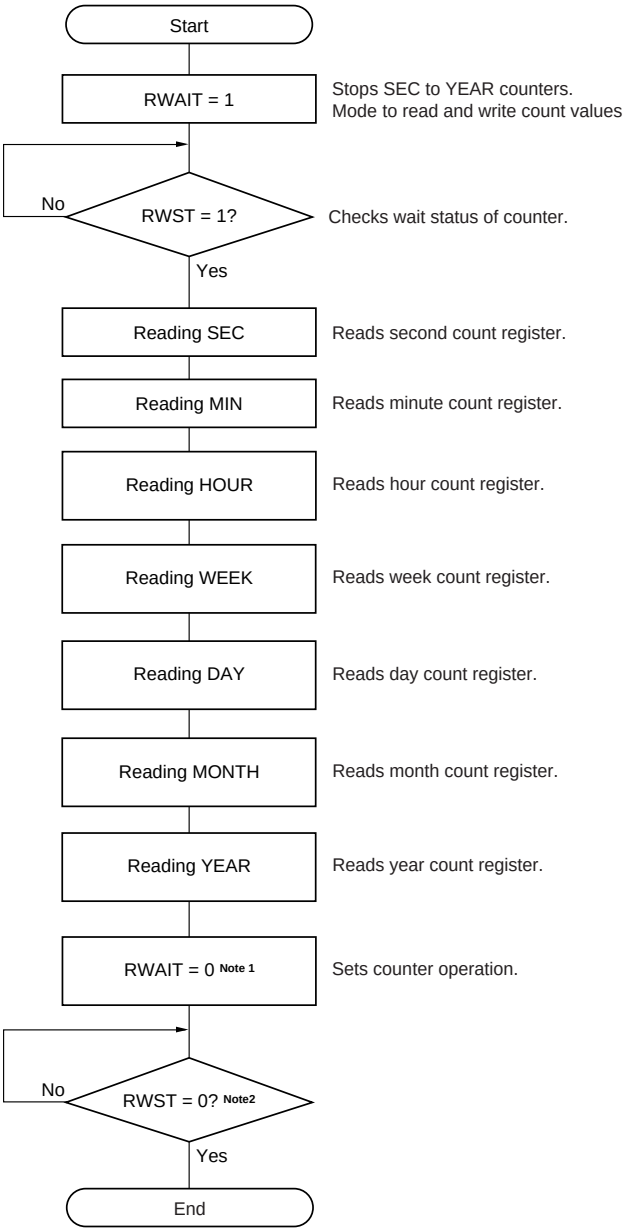
Figure 7-18. Procedure for Shifting to STOP Mode After Setting RTCE bit to 1



7.4.3 Reading/writing real-time clock

Read or write the counter after setting 1 to RWAIT first.

Figure 7-19. Procedure for Reading Real-time Clock



Notes 1. Set the RWAIT bit to 0 after reading/writing count values.

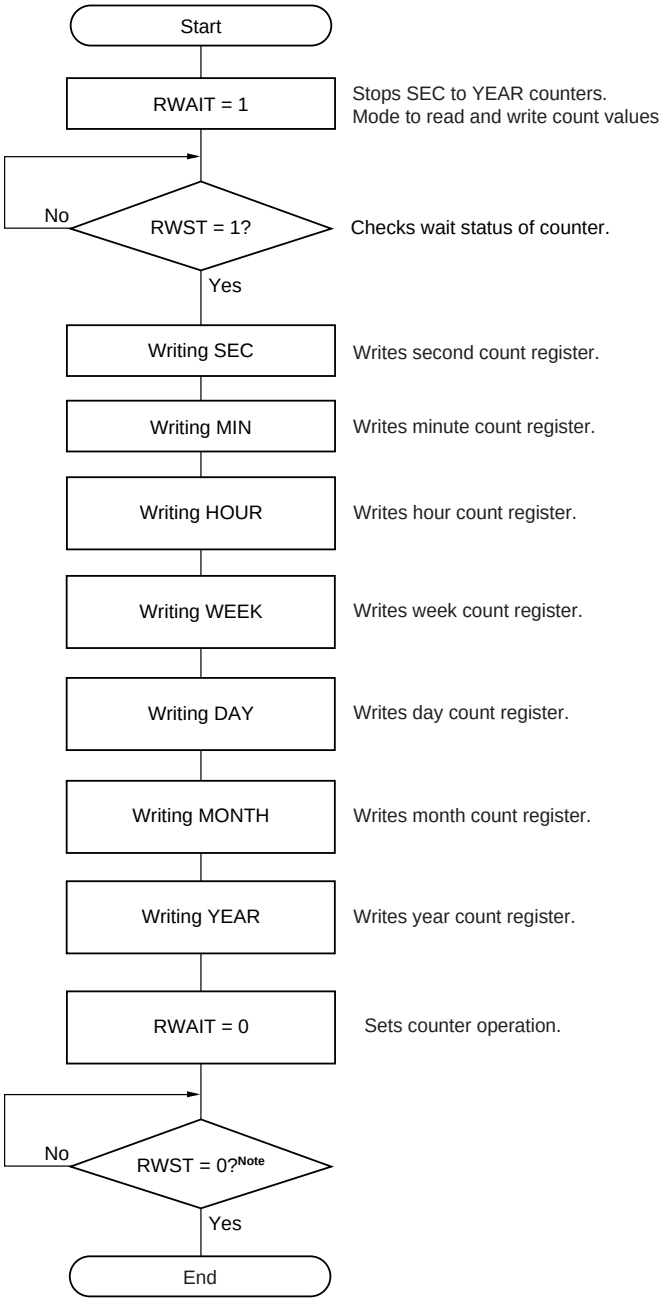
2. Be sure to confirm that RWST = 0 before setting STOP mode.

Caution Complete the series of operations 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 be set and only some registers may be read.

Figure 7-20. Procedure for Writing Real-time Clock



Notes 1. Be sure to confirm that RWST = 0 before setting STOP mode.

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.

Caution Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

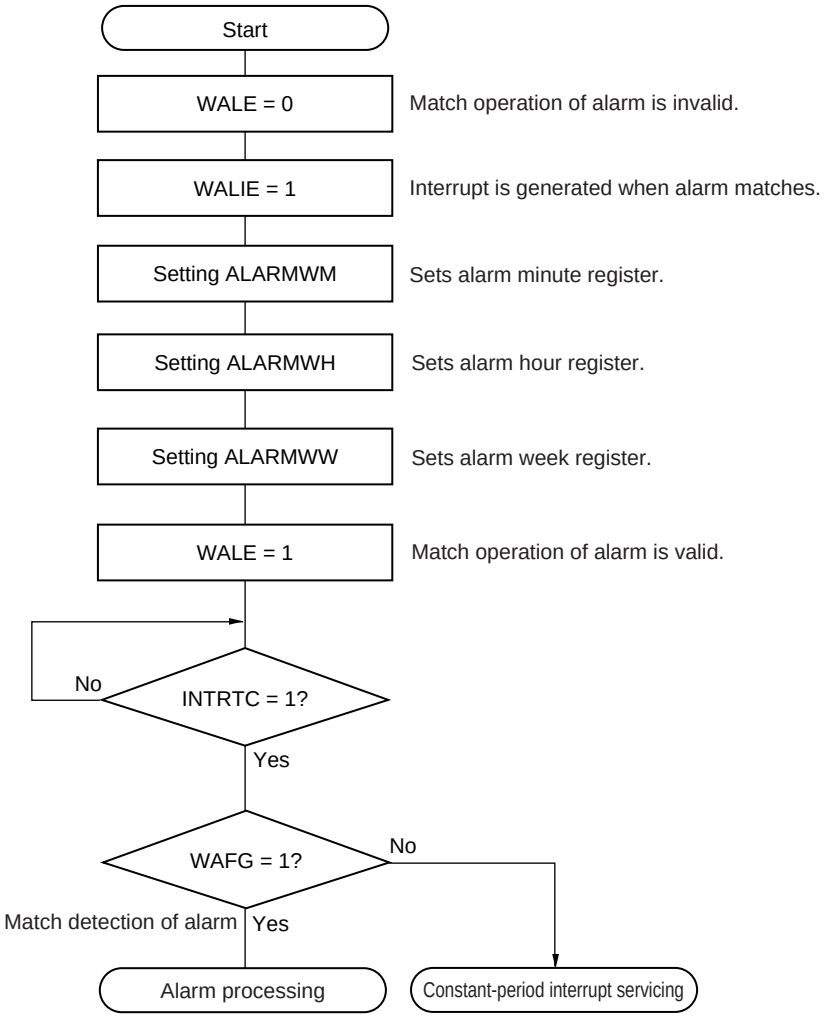
Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence.

All the registers do not have to be set and only some registers may be written.

7.4.4 Setting alarm of real-time clock

Set time of alarm after setting 0 to WALE first.

Figure 7-21. Alarm Setting Procedure

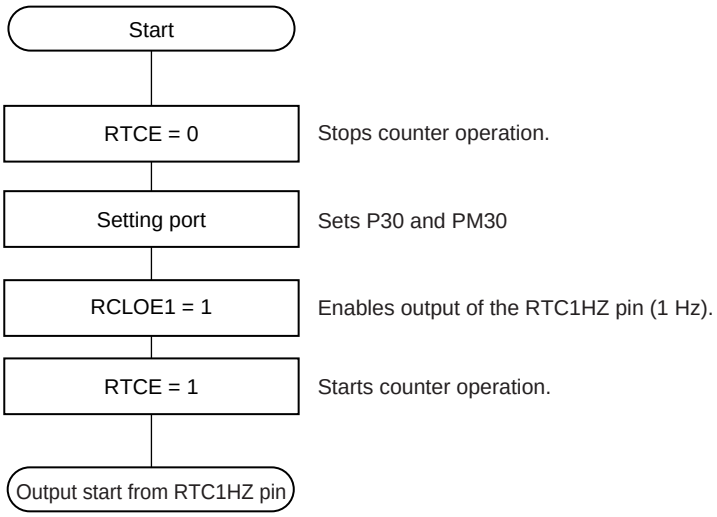


Remarks 1. The alarm week register (ALARMWW), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

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.

7.4.5 1 Hz output of real-time clock

Figure 7-22. 1 Hz Output Setting Procedure



Caution First set the RTCEN bit to 1, while oscillation of the input clock (f_{SUB}) is stable.

7.4.6 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 sub-count register 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).

Remarks 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

2. The oscillation frequency is the input clock (f_{RTC}).
It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 1 Hz from the RTC1HZ pin when the watch error correction register (SUBCUD) is set to its initial value (00H).

Note See **7.4.5 1Hz output of real-time clock** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 1 minute
= (Oscillation frequency ÷ Target frequency - 1) × 32768 × 60
= (32767.4 ÷ 32768 – 1) × 32768 ×60
= -36

[Calculating the values to be set to (F6 to F0)]

(When the correction value is -36)

If the correction value is 0 or less (when quickening), assume F6 to be 1.

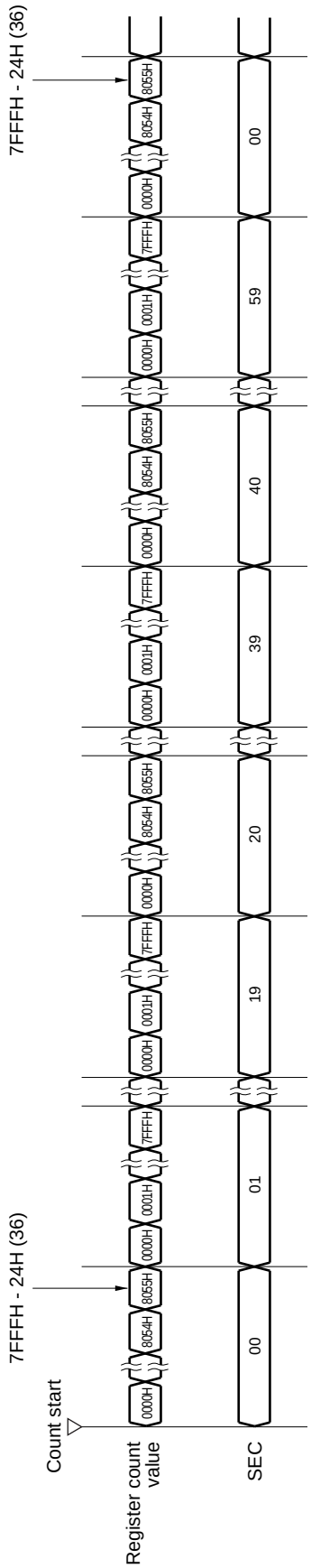
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

-{(/F5, /F4, /F3, /F2, /F1, /F0) - 1} × 2 = -36
(/F5, /F4, /F3, /F2, /F1, /F0) = 17
(/F5, /F4, /F3, /F2, /F1, /F0) = (0, 1, 0, 0, 0, 1)
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is -36 (bits 6 to 0 of the SUBCUD register: 1101110) results in 32768 Hz (0 ppm).

Figure 7-23 chows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 7-23. Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 8 INTERVAL TIMER

8.1 Functions of 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.

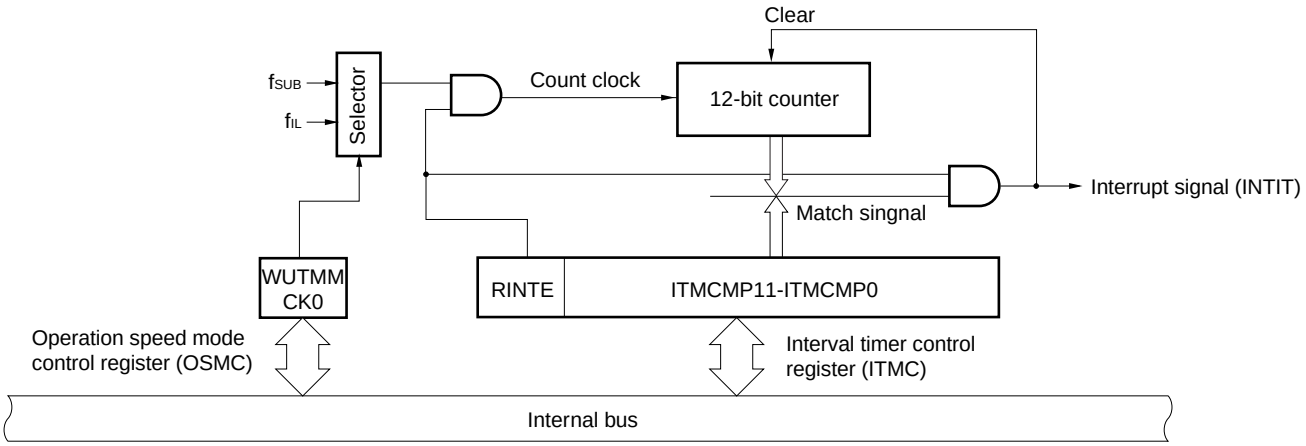
8.2 Configuration of Interval Timer

The interval timer includes the following hardware.

Table 8-1. Configuration of Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 0 (PER0)
	Operation speed mode control register (OSMC)
	Interval timer control register (ITMC)

Figure 8-1. Block Diagram of Interval Timer



8.3 Registers Controlling Interval Timer

The interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Operation speed mode control register (OSMC)
- Interval timer control register (ITMC)

(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 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 8-2. 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	0	ADCEN	IICA0EN Note1	SAU1EN Note1	SAU0EN	0	TAU0EN

RTCEN	Control of real-time clock (RTC) and interval timer input clock supply Note2
0	Stops input clock supply. • SFR used by the real-time clock (RTC) and interval timer cannot be written. • The real-time clock (RTC) and interval timer are in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) and interval timer can be read and written.

Notes 1. This is not provided in the 20-pin products.

2. The input clock that can be controlled by the RTCEN bit is used when the register that is used by the real-time clock (RTC) and interval timer are accessed from the CPU. The RTCEN bit cannot control supply of the operating clock (f_{SUB}) to RTC and interval timer.

Cautions 1. When using the interval timer, first set the RTCEN bit to 1, while oscillation of the input clock (f_{RTC}) is stable. If RTCEN = 0, writing to a control register of the real-time clock or interval timer is ignored, and, even if the register is read, only the default value is read.

2. Clock supply to peripheral functions other than the real-time clock and interval timer can be stopped in STOP mode or HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the operation speed mode control register (OSMC) to 1. In this case, set the RTCEN bit of the PER0 register to 1 and the other bits (bits 0 to 6) to 0.

3. Be sure to clear the following bits to 0.

20-pin products: bits 1, 3, 4, 6

30, 32-pin products: bits1, 6

48, 64-pin products: bits1, 6

(2) Operation speed mode control register (OSMC)

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

In addition, by stopping clock functions that are even a little unnecessary, the RTCLPC bit can be used to reduce power consumption. For details about setting the RTCLPC bit, see **CHAPTER 5 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 Operation Speed 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 interval timer.
0	Subsystem clock (f_{SUB})
1	Low-speed on-chip oscillator clock (f_{IL})

(3) Interval timer control register (ITMC)

This register is used to set up the starting and stopping of the 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 8-4. Format of 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	Interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITCMP11 to ITCMP0	Specification of the interval timer compare value
001H	These bits generate an interrupt at the fixed cycle (count clock cycles x (ITCMP setting + 1)).
• • •	
FFFH1	

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]

Cautions 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. In addition, after rewriting the bit value, clear the ITIF flag, and then enable the interrupt servicing.

2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit. So make the transition to the HALT/STOP state after check the reflection of writing to the RINTE bit.

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.

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.

8.4 Interval Timer Operation

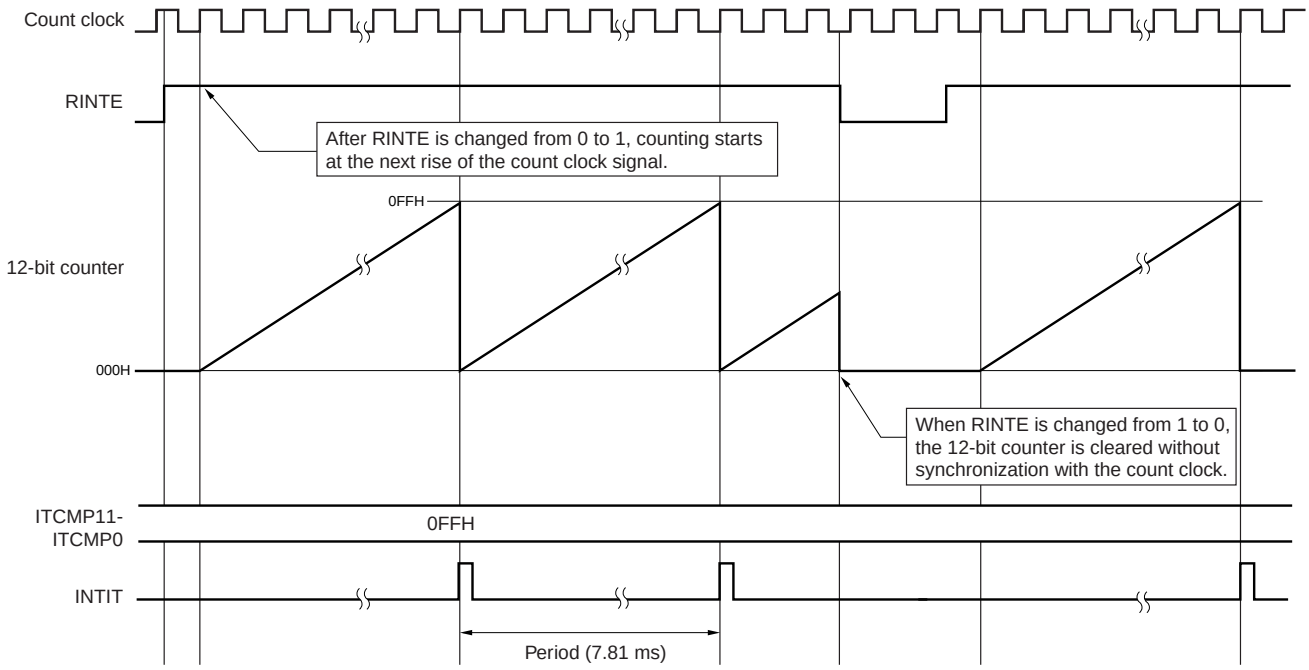
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate an interval timer that repeatedly generates interrupt requests (INTIT).

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 interval timer is as follows.

Figure 8-5. Interval Timer Operation Timing (ITCMP11 to ITCMP0 = 0FFH, count clock: f_{SUB} = 32.768 kHz)



CHAPTER 9 16-BIT WAKEUP TIMER

The RL78/F12 incorporates a 16-bit wakeup timer (WUTM).

9.1 Overview

The 16-bit wakeup timer (WUTM) has the following functions.

- Interval function
  Counter × 1
  Compare × 1
  Compare match interrupt × 1

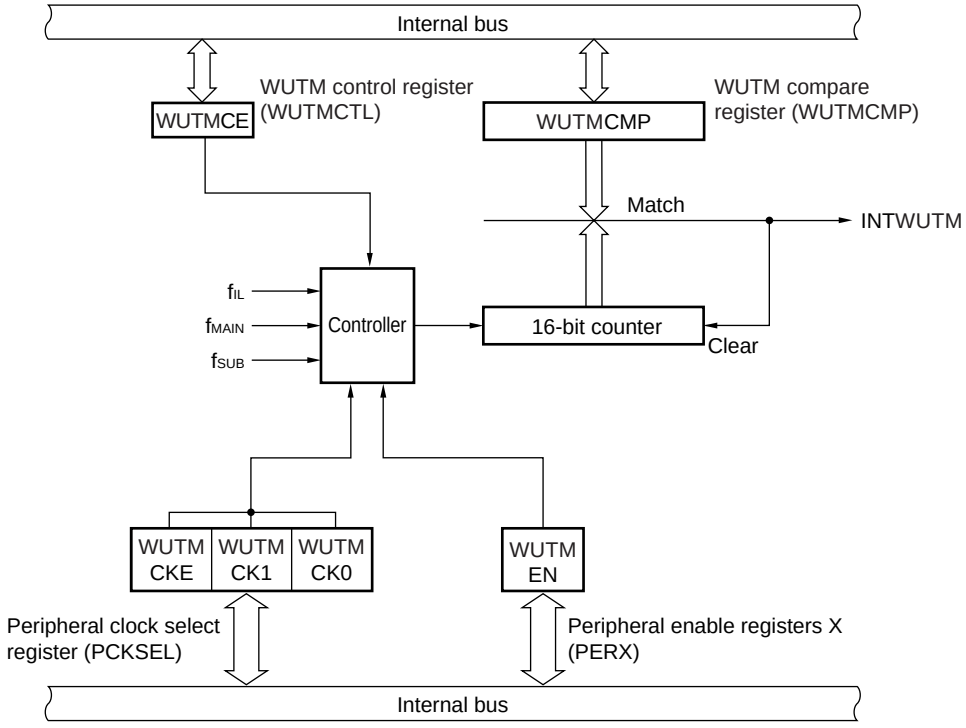
9.2 Configuration

WUTM includes the following hardware.

Table 9-1. Configuration of WUTM

Item	Configuration
Timer register	16-bit counter
Control register	Peripheral enable register X (PERX) Peripheral clock select register (PCKSEL) WUTM control register (WUTMCTL)
Register	WUTM compare register (WUTMCMP)

Figure 9-1. Block Diagram of WUTM



Remark f_{IL}: Low-speed on-chip oscillator clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subclock frequency (48, 64-pin products only)

9.3 Register

(1) Peripheral enable register X (PERX)

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

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

Reset signal generation clears theses registers to 00H.

Caution Whether to enable or disable SFR reading and writing only is selected for the 16-bit wakeup timer.
Whether to enable or disable supplying the operating clock is selected using the PCKSEL register.

Figure 9-2. Format of Peripheral Enable Register X (PERX)

Address: F0500H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PERX	0	0	0	0	0	UF0EN	SAUSEN	WUTEN

WUTEN	Control of 16-bit wakeup timer input clock
0	Stops input clock supply for SFR writing. • SFR used by the 16-bit wakeup timer cannot be read and written.
1	Supplies input clock for SFR writing. • SFR used by the 16-bit wakeup timer can be read and written.

Caution Be sure to clear the bits 3 to 7 of the PERX register to 0.

(2) Peripheral clock select register (PCKSEL)

This register is used to select for and supply to each peripheral hardware device the operating clock.

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

Reset signal generation clears this register to 00H.

Caution Set the PCKSEL register before starting to operate each peripheral hardware device.

Figure 9-3. Format of Peripheral Clock Select Register (PCKSEL)

Address: F0501H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	1	0
PCKSEL	0	0	0	0	0	WUTMCKE	WUTMCK1	WUTMCK0

WUTMCKE	Control of 16-bit wakeup timer operating clock
0	Stops supplying operating clock.
1	Supplies operating clock.

WUTMCK1	WUTMCK0	16-bit wakeup timer operating clock selection
0	0	f_{IL}
0	1	f_{SUB}
1	0	$f_{MAIN}/2^8$
1	1	$f_{MAIN}/2^{12}$

Caution Be sure to clear bits 3 to 7 of the PCKSEL register to 0.

(3) WUTM compare register (WUTMCMP)

The WUTMCMP register is a 16-bit compare register.

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

Reset sets this register to 0000H.

Cautions 1. Rewriting the WUTMCMP register is prohibited while the timer is operating (WUTMCE = 1).

2. When writing the WUTMCMP register, be sure to set bit 0 (WUTEN) of peripheral enable register 1 (PERX) to 1 and supply an input clock.

Figure 9-4. WUTM compare register (WUTMCMP)

Address: F0582H, F0583H After reset: 0000H R/W

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

(4) WUTM control register (WUTMCTL)

The WUTMCTL register is an 8-bit register that controls the WUTM operation.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution When writing the WUTMCTL register, be sure to set bit 0 (WUTEN) of peripheral enable register 1 (PERX) to 1 and supply an input clock.

Figure 9-5. Format of WUTM control register (WUTMCTL)

Address: F0580H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
WUTMCTL	WUTMCE	0	0	0	0	0	0	0

WUTMCE	Control of WUTM operation
0	Operation disabled
1	Operation enabled
WUTM is asynchronously reset by the WUTMCE bit. If the WUTMCE bit is set to “1”, the internal operating clock is enabled within two input clocks after the WUTMCE bit has been set to “1” and WUTM starts counting up.	

Cautions 1. Be sure to clear bits 0 to 6 of the WUTMCTL register to 0.

2. If the WUTMCE bit is cleared to 0, the counter value within WUTM is immediately cleared.

9.4 Operation

9.4.1 Interval timer mode

In the interval timer mode, if the 16-bit counter and WUTM compare register (WUTMCMP) values match, the counter is cleared to 0000H and starts counting up again at the same time a match interrupt signal (INTWUTM) is output.

Figure 9-6. Interval Timer Mode Operation Start Flow

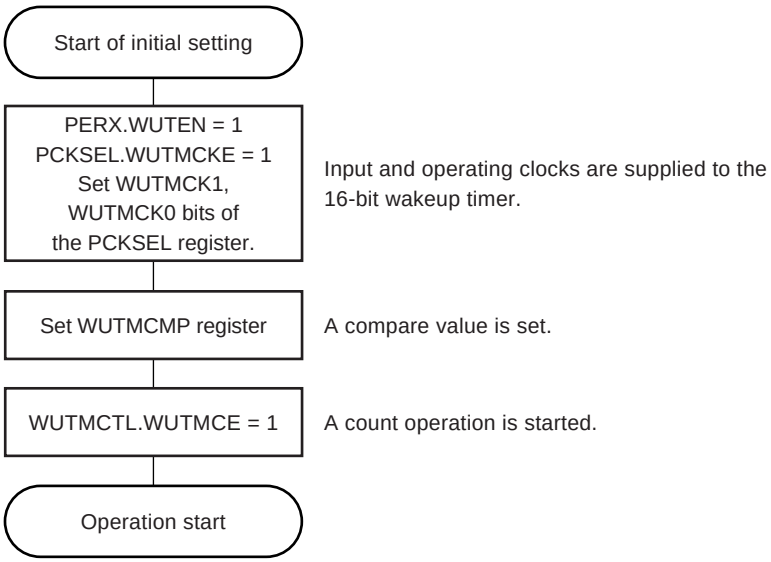
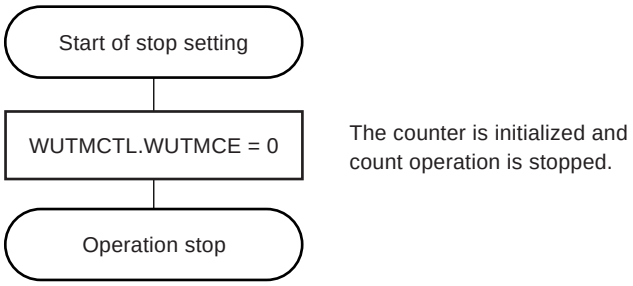
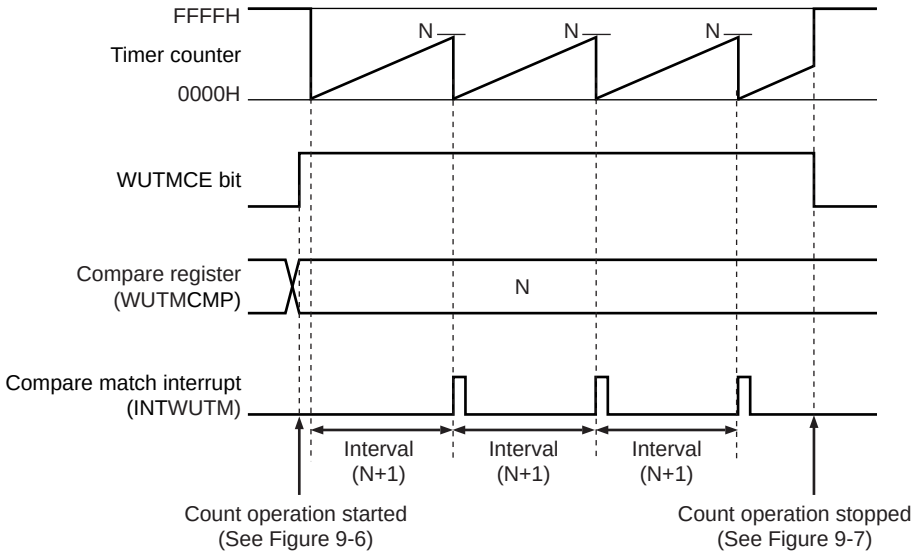


Figure 9-7. Interval Timer Mode Operation Stop Flow



Remark To reduce the power consumption by stopping WUTM, clear (0) also PERX.WUTEN and PCKSEL.WUTMCKE.

Figure 9-8. Operation Timing of Interval Timer Mode



Caution The interrupt cycle can be calculated by using the following expression.

INTWUTM (timer interrupt) generation cycle = Operating clock cycle × (WUTMCTL setting value + 1)

9.4.2 Cautions

(1) Clock generator and clock enable timing

The operation timing of the count clock is shown below.

Figure 9-9. Count Operation Start Timing (Min. Delay)

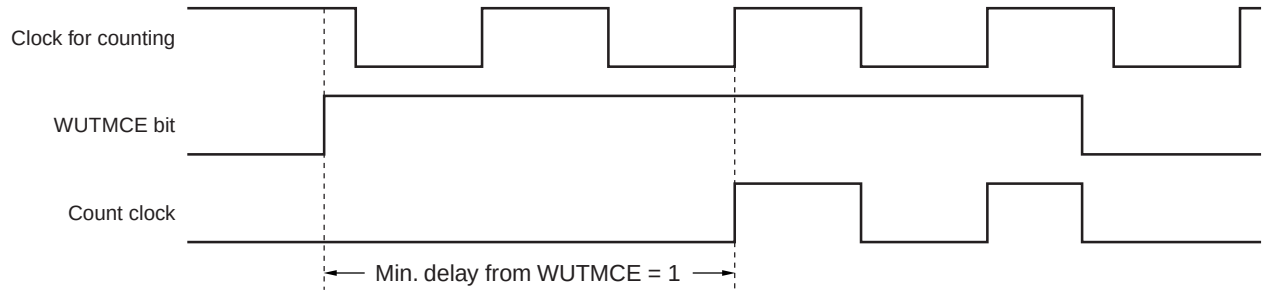
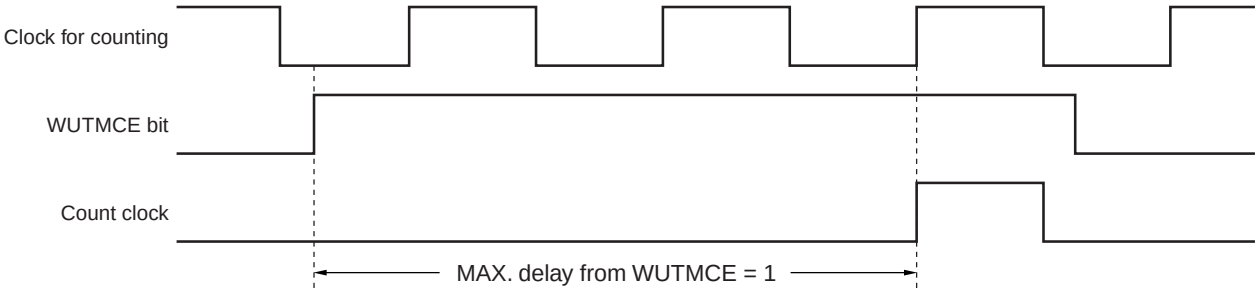


Figure 9-10. Count Operation Start Timing (Max. Delay)



(2) Rewriting register during WUTM operation

Rewriting the WUTMCMP register is prohibited while WUTM is operating.

If the WUTMCMP register is rewritten when the WUTMCE bit is 1, an interrupt may be generated.

(3) WUTM operation in standby mode

Since the operations in standby mode depend on the operation state of the low-speed on-chip oscillator clock (f_{IL}), the operations depend on the operation set by option byte and RTC/interval timer operation clock selection of socket chip.

If WUTM is desired to be operated in standby mode, set WDT so as to continue operating in standby mode or select fIL to the operation clock of RTC/interval timer.

CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

The number of output pins of the clock output and buzzer output controllers differs, depending on the product.

Output pin	20-pin	30, 32-pin	48, 64-pin
PCLBUZ0	√	√	√
PCLBUZ1	–	√	√

Caution Most of the following descriptions in this chapter use the 64-pin as an example.

10.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and 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.

Two output pins, PCLBUZ0 and PCLBUZ1, are available.

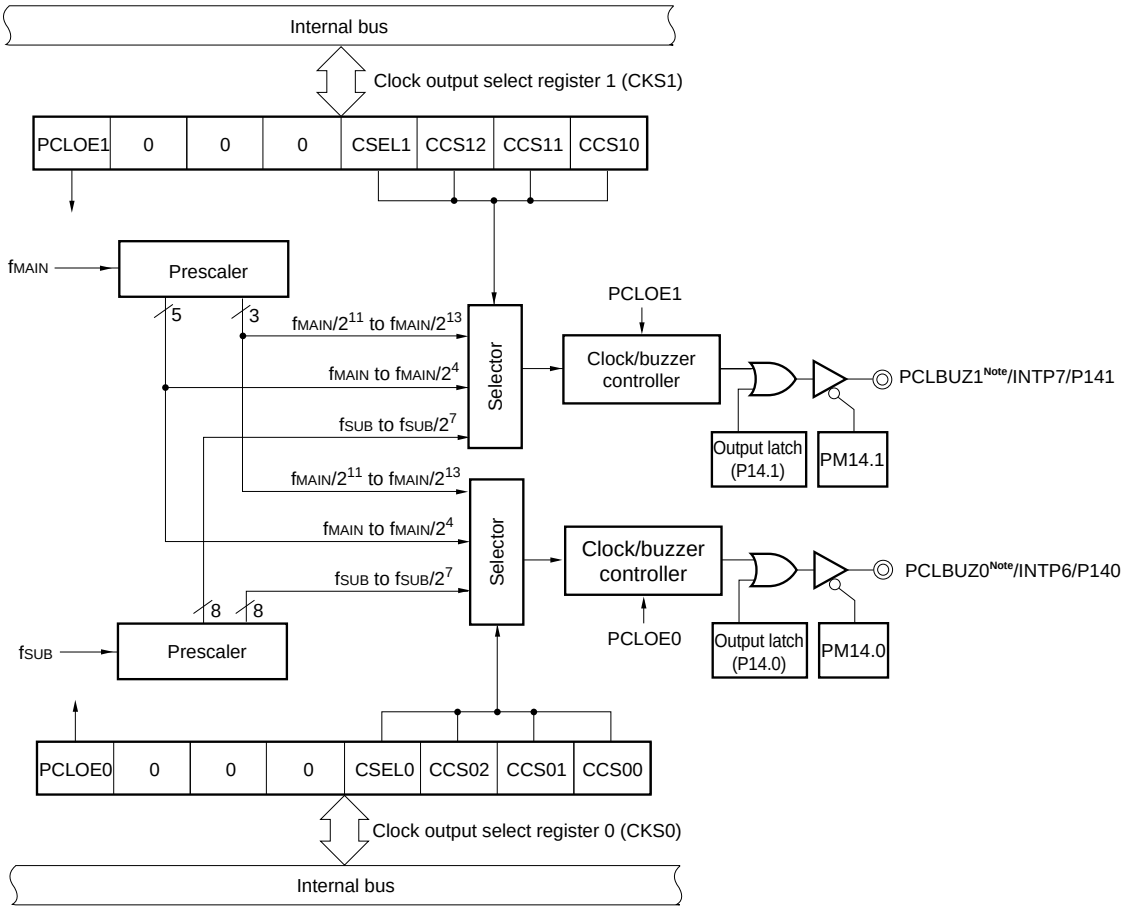
The PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).

Figure 10-1 shows the block diagram of clock output/buzzer output controller.

Caution In the low-consumption RTC mode (when the RTCLPC bit of the operation speed mode control register (OSMC) = 1), it is not possible to output the subsystem clock (f_{SUB}) from the PCLBUZn pin.

Remark n = 0, 1

Figure 10-1. Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **31.5 AC Characteristics** or **32.5 AC Characteristics**.

10.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 10-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Port mode register 14 (PM14) Port register 14 (P14)

10.3 Registers Controlling Clock Output/Buzzer Output Controller

The following two registers are used to control the clock output/buzzer output controller.

- Clock output select registers n (CKSn)
- Port mode register 14 (PM14)

(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.

Figure 10-2. Format of Clock Output Select Register 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	10 MHz[Note]	Setting prohibited[Note]	Setting prohibited[Note]
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz[Note]	16 MHz[Note]
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz	8 MHz
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	9.76 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 16 MHz. Furthermore, the available output frequency depends on the grade. For details, refer to **31.5 AC Characteristics** or **32.5 AC Characteristics**.

Cautions 1. Change the output clock after disabling clock output (PCLOEn = 0).

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 in STOP mode.

3. In the low-consumption RTC mode (when the RTCLPC bit of the operation speed mode control register (OSMC) = 1), it is not possible to output the subsystem clock (f_{SUB}) from the PCLBUZn pin.

Remarks 1. n = 0, 1

2. f_{MAIN}: Main system clock frequency

f_{SUB}: Subsystem clock frequency

(2) Port mode register 14 (PM14)

These registers set input/output of port 1 in 1-bit units.

When using the P141/INTP7/PCLBUZ1 and P140/INTP6/PCLBUZ0 pins for clock output and buzzer output, clear PM14.1, PM14.0 bits and the output latches of P14.1, P14.0 to 0.

The PM14 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 10-3. Format of Port Mode Register 14 (PM14)

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	PM14.7	PM14.6	1	1	1	1	PM14.1	PM14.0

PMmn	Pmn pin I/O mode selection (m = 14 ; n = 0, 1, 6, 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark For details of the port mode register other than 64-pin products, see **4.3 Registers Controlling Port Function**.

10.4 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).

10.4.1 Operation as output pin

The PCLBUZn pin is output as the following procedure.

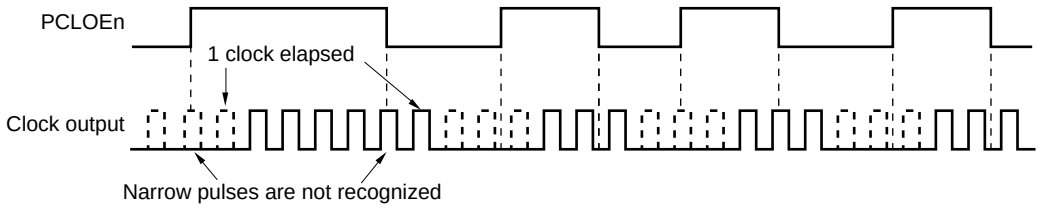
<1> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).

<2> Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Remarks 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 10-4 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.

2. n = 0, 1

Figure 10-4. Remote Control Output Application Example



10.5 Cautions of clock output/buzzer output controller

When the main system clock is selected for the PCLBUZn output (CSEL = 0), if STOP or HALT mode is entered within 1.5 main system clock cycles after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

CHAPTER 11 WATCHDOG TIMER

11.1 Functions of Watchdog Timer

The watchdog timer operates on the low-speed on-chip oscillator clock.

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of the RESF register, see **CHAPTER 21 RESET FUNCTION**.

When 75% + 1/2f_{IL} of the overflow time is reached, an interval interrupt can be generated.

11.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 11-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

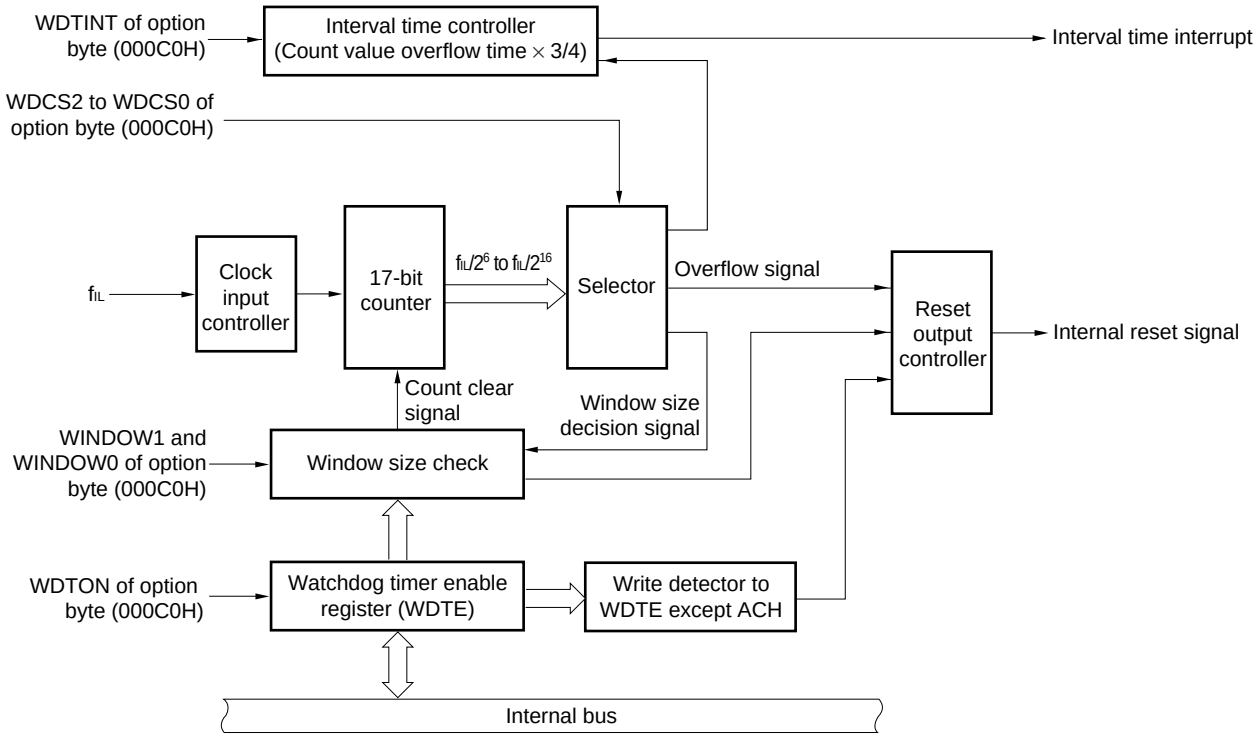
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 11-2. 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 11-1. Block Diagram of Watchdog Timer



11.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

(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 11-2. 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

Cautions 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

3. The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

11.4 Operation of Watchdog Timer

11.4.1 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 **11.4.2** 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 **11.4.3** 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

Cautions 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.

2. If the watchdog timer is cleared by writing “ACH” to the WDTE register, the actual overflow time may be different from the overflow time set by the option byte by up to 2/f_{IL} seconds.

3. The watchdog timer can be cleared immediately before the count value overflows.

Cautions 4. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		

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.

5. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM™ emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

11.4.2 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 11-3. 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	$2^{13}/f_{IL}$ (474.90 ms)
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)

Caution The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

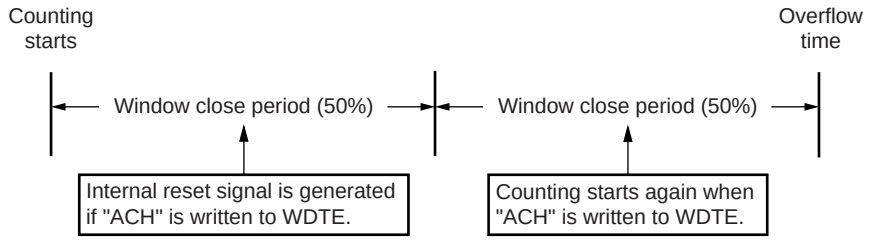
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

11.4.3 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 11-4. 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%
1	1	100%

Cautions 1. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

2. When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.

Remark If the overflow time is set to $2^9/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^9/f_{IL}$ (MAX.) = 2^9/17.25 kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^9/f_{IL}$ (MIN.) × (1 – 0.5) = 0 to 2^9/12.75 kHz (MIN.) × 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 (MIN.) × 0.5 to 2^9/17.25 kHz (MAX.) = 20.08 to 29.68 ms

11.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of an option byte (000C0H), an interval interrupt (INTWDTI) can be generated when 75% + 1/2f_{IL} of the overflow time is reached.

Table 11-5. Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is used.
1	Interval interrupt is generated when 75% + 1/2f_{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 12 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

	20-pin	30, 32-pin	48-pin	64-pin
Analog input channels	4 ch (ANI0 to ANI2, ANI16)	8 ch (ANI0 to ANI3, ANI16 to ANI19)	10 ch (ANI0 to ANI7, ANI18, ANI19)	12 ch (ANI0 to ANI7, ANI16 to ANI19)

Caution Most of the following descriptions in this chapter use the 64-pin as an example.

12.1 Function of A/D Converter

The A/D converter is a 10-bit resolution[Note] converter that converts analog input signals into digital values, and is configured to control analog inputs, including up to twelve channels of A/D converter analog inputs (ANI0 to ANI7 and ANI16 to ANI19).

The A/D converter has the following function.

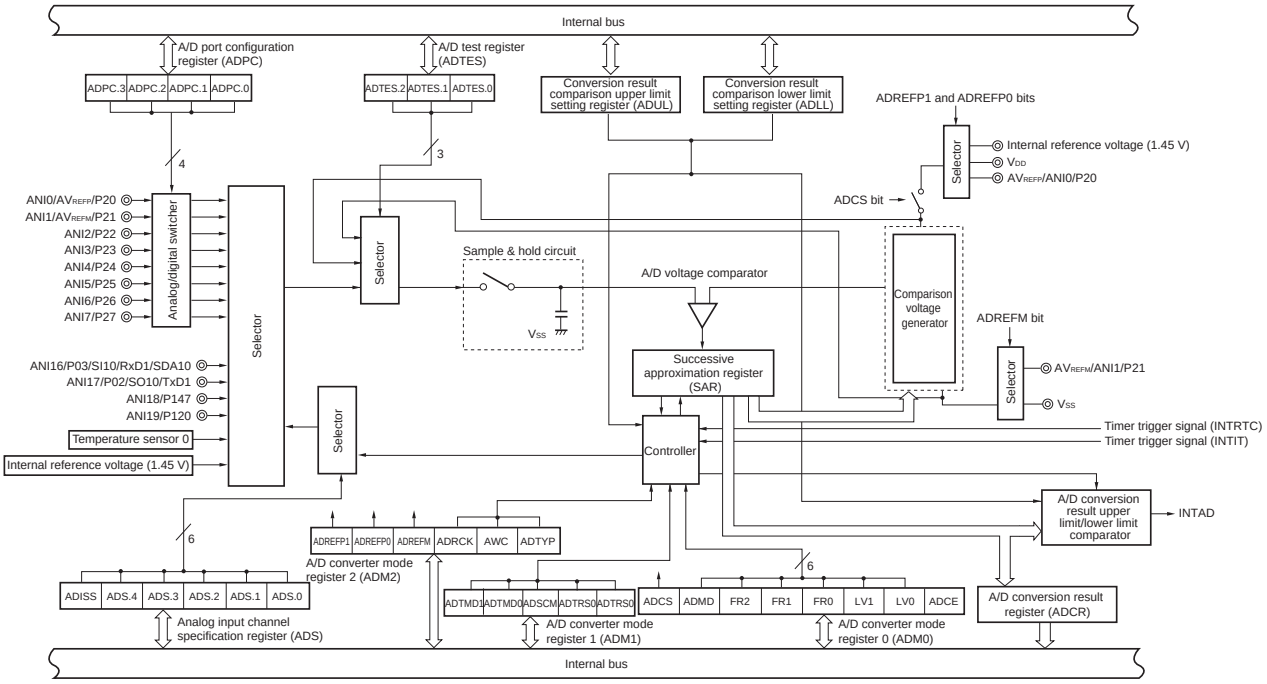
- **10-bit resolution A/D conversion[Note]**
  10-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI7 and ANI16 to ANI19. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

Note 8-bit resolution can also be selected by using the ADTYP bit of A/D converter mode register 2 (ADM2).

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger Mode	Channel Selection Mode	Conversion Operation Mode
• Software trigger Conversion is started by specifying a software trigger. • 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.	• Select mode A/D conversion is performed on the analog input of one channel. • Scan mode A/D conversion is performed on the analog input of four channels in order.	• 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.

Figure 12-1. Block Diagram of A/D Converter



Remark The analog input pins in the figure are provided in the 64-pin products.

12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI7 and ANI16 to ANI19 pins

These are the analog input pins of the up to 12 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}, the internal reference voltage (1.45 V), and V_{DD}.

(4) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR register is a 10-bit 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.

(9) AV_{REFP} pin

This pin inputs an external reference voltage (AV_{REFP}).

If using AV_{REFP} as the + side reference voltage of the A/D converter, set the ADREFP1 and ADREFP0 bits of A/D converter mode register 2 (ADM2) to 0 and 1, respectively.

The analog signals input to ANI0 to ANI7 and ANI16 to ANI19 are converted to digital signals based on the voltage applied between AV_{REFP} and the – side reference voltage (AV_{REFM}/V_{SS}).

In addition to AV_{REFP}, it is possible to select V_{DD} or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

(10) AV_{REFM} pin

This pin inputs an external reference voltage (AV_{REFM}). If using AV_{REFM} as the – side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AV_{REFM}, it is possible to select V_{SS} as the – side reference voltage of the A/D converter.

Caution The A/D conversion accuracy differs depending on the used pins or reference voltage setting. For details, see CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE) and CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE).

12.3 Registers Used in A/D Converter

The A/D converter uses 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 registers 0, 12, and 14 (PMC0, PMC12, PMC14)
- Port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14)

(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 12-2. 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	0	ADCEN	IICA0EN[Note]	SAU1EN[Note]	SAU0EN	0	TAU0EN

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

Note This is not provided in the 20-pin products.

Cautions 1. When setting the A/D converter, be sure to set the ADCEN bit to 1 first. If ADCEN = 0, writing to a control register of the A/D converter is ignored, and, even if the register is read, only the default value is read (except for port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14), port mode control registers 0, 12, and 14 (PMC0, PMC12, PMC14), and A/D port configuration register (ADPC)).

2. Be sure to clear the following bits to 0.

20-pin products: Bits 1, 3, 4, 6

30, 32, 48-pin products: Bits 1, 6

(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 12-3. 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	FR2Note 1	FR1Note 1	FR0Note 1	LV1Note 1	LV0Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation [When read] Conversion stopped/standby status
1	Enables conversion operation [When readNote 2] While in the software trigger mode: Conversion operation status While in the hardware trigger wait mode: Stabilization wait status + conversion operation status

ADMD	Specification of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation controlNote 3
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Notes 1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Table 12-3 A/D Conversion Time Selection**.

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.

Cautions 1. Change the bits ADMD, FR2 to FR0, LV1 and LV0, and ADCE in the conversion stop state or conversion wait state (ADCS = 0).

2. It is prohibited to change the ADCE and ADCS bits from 0 to 1 by an 8-bit manipulation instruction. To change these bits, use the procedure described in 12.7, A/D Converter Setup Flowchart.

Table 12-1. Settings of ADCS and ADCE Bits

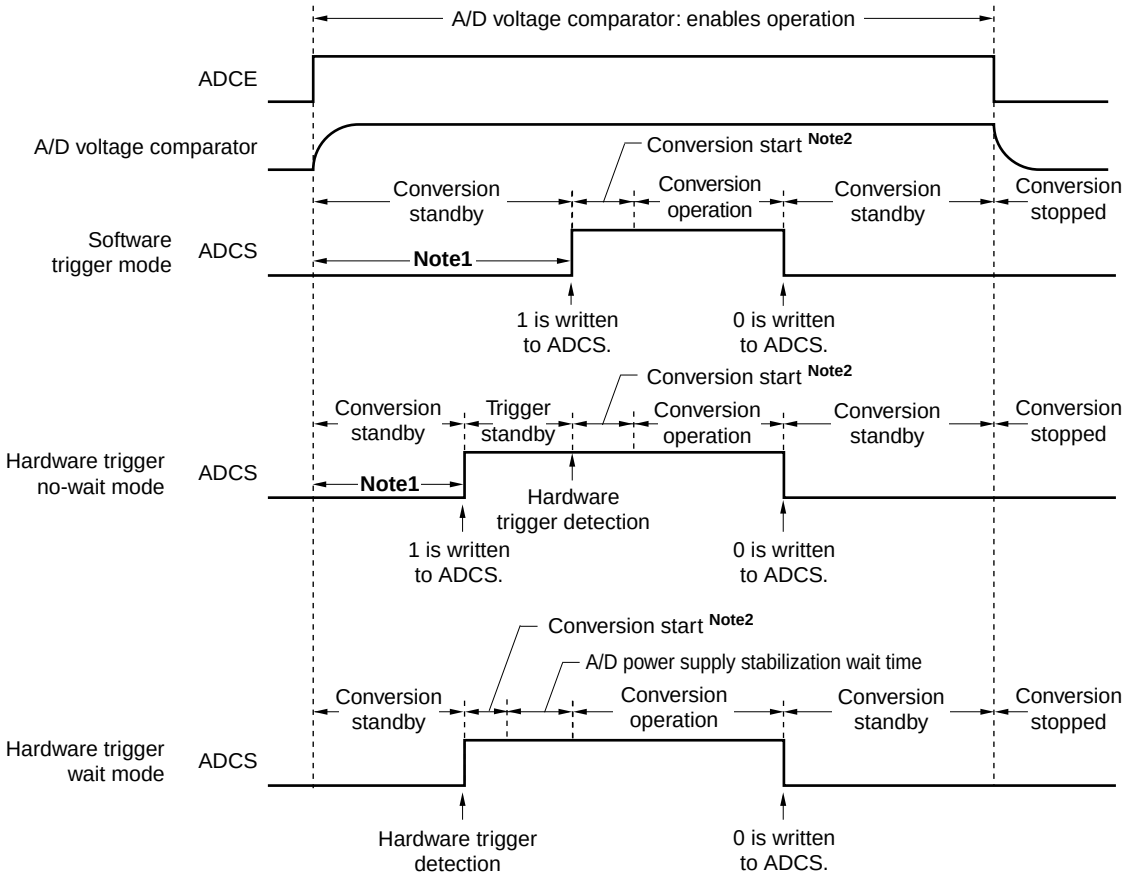
ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion standby mode (only A/D voltage comparator consumes power)
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator: enables operation)

Note In hardware trigger wait mode, the DC power consumption path is not provided even in conversion wait mode.

Table 12-2. Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Select mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.

Figure 12-4. Timing Chart When A/D Voltage Comparator Is Used



Notes 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.

2. In starting conversion, the longer will take up to following time.

ADM0			Conversion clock (f_{AD})	Conversion Operation Time (f_{CLK} clock)	
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	

In the conversion after the second conversion in continuous conversion mode or after the scan 1 in scan mode, the conversion startup time or A/D power supply stabilization wait time is not generated after detection of a hardware trigger.

Cautions 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.

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.

3. **Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).**
4. **To complete A/D conversion, the following hardware trigger interval time is required:**
   In hardware trigger no-wait mode: Two f_{CLK} clock cycles + A/D conversion time
   In hardware trigger wait mode: Two f_{CLK} clock cycles + stabilization wait time + A/D conversion time

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

<R>

Table 12-3. A/D Conversion Time Selection (1/6)

(1) 4.0 V ≤ V_{DD} ≤ 5.5 V

When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 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	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	38 μs
0	0	1				f_{CLK}/32					38 μs	19 μs
0	1	0				f_{CLK}/16				38 μs	19 μs	9.5 μs
0	1	1				f_{CLK}/8			38 μs	19 μs	9.5 μs	4.75 μs
1	0	0				f_{CLK}/6			28.5 μs	14.25 μs	7.125 μs	3.5625 μs
1	0	1				f_{CLK}/5			23.75 μs	11.875 μs	5.938 μs	2.9688 μs
1	1	0				f_{CLK}/4		38 μs	19 μs	9.5 μs	4.75 μs	2.375 μs
1	1	1				f_{CLK}/2	38 μs	19 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	34 μs
0	0	1				f_{CLK}/32					34 μs	17 μs
0	1	0				f_{CLK}/16				34 μs	17 μs	8.5 μs
0	1	1				f_{CLK}/8			34 μs	17 μs	8.5 μs	4.25 μs
1	0	0				f_{CLK}/6			25.5 μs	12.75 μs	6.375 μs	3.1875 μs
1	0	1				f_{CLK}/5			21.25 μs	10.625 μs	5.3125 μs	2.6563 μs
1	1	0				f_{CLK}/4		34 μs	17 μs	8.5 μs	4.25 μs	2.125 μs
1	1	1				f_{CLK}/2	34 μs	17 μs	8.5 μs	4.25 μs	2.125 μs	Setting prohibited
Other than above						Setting prohibited						

Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

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

<R>

Table 12-3. A/D Conversion Time Selection (2/6)

(2) 2.7 V ≤ V_{DD} < 5.5 V

When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 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	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	38 μs
0	0	1				f_{CLK}/32					38 μs	19 μs
0	1	0				f_{CLK}/16				38 μs	19 μs	9.5 μs
0	1	1				f_{CLK}/8			38 μs	19 μs	9.5 μs	4.75 μs
1	0	0				f_{CLK}/6			28.5 μs	14.25 μs	7.125 μs	3.5625 μs
1	0	1				f_{CLK}/5			23.75 μs	11.875 μs	5.9375 μs	Setting prohibited
1	1	0				f_{CLK}/4		38 μs	19 μs	9.5 μs	4.75 μs	
1	1	1				f_{CLK}/2	38 μs	19 μs	9.5 μs	4.75 μs	Setting prohibited	
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	34 μs
0	0	1				f_{CLK}/32					34 μs	17 μs
0	1	0				f_{CLK}/16				34 μs	17 μs	8.5 μs
0	1	1				f_{CLK}/8			34 μs	17 μs	8.5 μs	4.25 μs
1	0	0				f_{CLK}/6			25.5 μs	12.75 μs	6.375 μs	3.1875 μs
1	0	1				f_{CLK}/5			21.25 μs	10.625 μs	5.3125 μs	2.6563 μs
1	1	0				f_{CLK}/4		34 μs	17 μs	8.5 μs	4.25 μs	Setting prohibited
1	1	1				f_{CLK}/2	34 μs	17 μs	8.5 μs	4.25 μs	Setting prohibited	
Other than above						Setting prohibited						

Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

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

<R>

Table 12-3. A/D Conversion Time Selection (3/6)

(3) 1.8 V ≤ V_{DD} < 5.5 V

When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 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	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	38 μs
0	0	1				f_{CLK}/32					38 μs	19 μs
0	1	0				f_{CLK}/16				38 μs	19 μs	Setting prohibited
0	1	1				f_{CLK}/8			38 μs	19 μs	Setting prohibited	
1	0	0				f_{CLK}/6			28.5 μs	Setting prohibited		
1	0	1				f_{CLK}/5			23.75 μs			
1	1	0				f_{CLK}/4		38 μs	19 μs			
1	1	1				f_{CLK}/2	38 μs	19 μs	Setting prohibited			
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	34 μs
0	0	1				f_{CLK}/32					34 μs	17 μs
0	1	0				f_{CLK}/16				34 μs	17 μs	Setting prohibited
0	1	1				f_{CLK}/8			34 μs	17 μs	Setting prohibited	
1	0	0				f_{CLK}/6			25.5 μs	Setting prohibited		
1	0	1				f_{CLK}/5			21.25 μs			
1	1	0				f_{CLK}/4		34 μs	17 μs			
1	1	1				f_{CLK}/2	34 μs	17 μs	Setting prohibited			
Other than above						Setting prohibited						

Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

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

<R>

Table 12-3. A/D Conversion Time Selection (4/6)

(4) 4.0 V ≤ V_{DD} ≤ 5.5 V

When there is stabilization wait time (hardware trigger wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 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	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						27 μs
0	1	0				f_{CLK}/16					27 μs	13.5 μs
0	1	1				f_{CLK}/8				27 μs	13.5 μs	6.75 μs
1	0	0				f_{CLK}/6				20.25 μs	10.125 μs	5.0625 μs
1	0	1				f_{CLK}/5			33.75 μs	16.875 μs	8.4375 μs	4.2188 μs
1	1	0				f_{CLK}/4			27 μs	13.5 μs	6.75 μs	3.375 μs
1	1	1				f_{CLK}/2		27 μs	13.5 μs	6.75 μs	3.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						25 μs
0	1	0				f_{CLK}/16					25 μs	12.5 μs
0	1	1				f_{CLK}/8				25 μs	12.5 μs	6.25 μs
1	0	0				f_{CLK}/6			37.5 μs	18.75 μs	9.375 μs	4.6875 μs
1	0	1				f_{CLK}/5			31.25 μs	15.625 μs	7.8125 μs	3.9063 μs
1	1	0				f_{CLK}/4			25 μs	12.5 μs	6.25 μs	3.125 μs
1	1	1				f_{CLK}/2		25 μs	12.5 μs	6.25 μs	3.125 μs	Setting prohibited
Other than above						Setting prohibited						

Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

3. While in the hardware trigger wait mode, the conversion time includes the time spent waiting for stabilization after the hardware trigger is detected.

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

<R>

Table 12-3. A/D Conversion Time Selection (5/6)

(5) 2.7 V ≤ V_{DD} < 5.5 V

When there is stabilization wait time (hardware trigger wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 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	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						27 μs
0	1	0				f_{CLK}/16					27 μs	13.5 μs
0	1	1				f_{CLK}/8				27 μs	13.5 μs	6.75 μs
1	0	0				f_{CLK}/6				20.25 μs	10.125 μs	5.0625 μs
1	0	1				f_{CLK}/5			33.75 μs	16.875 μs	8.4375 μs	4.2188 μs
1	1	0				f_{CLK}/4			27 μs	13.5 μs	6.75 μs	3.375 μs
1	1	1				f_{CLK}/2		27 μs	13.5 μs	6.75 μs	3.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						25 μs
0	1	0				f_{CLK}/16					25 μs	12.5 μs
0	1	1				f_{CLK}/8				25 μs	12.5 μs	6.25 μs
1	0	0				f_{CLK}/6			37.5 μs	18.75 μs	9.375 μs	4.6875 μs
1	0	1				f_{CLK}/5			31.25 μs	15.625 μs	7.8125 μs	3.9063 μs
1	1	0				f_{CLK}/4			25 μs	12.5 μs	6.25 μs	Setting prohibited
1	1	1				f_{CLK}/2		25 μs	12.5 μs	6.25 μs	Setting prohibited	
Other than above						Setting prohibited						

Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

3. While in the hardware trigger wait mode, the conversion time includes the time spent waiting for stabilization after the hardware trigger is detected.

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

<R>

Table 12-3. A/D Conversion Time Selection (6/6)

(6) 1.8 V ≤ V_{DD} < 5.5 V

When there is stabilization wait time (hardware trigger wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Conversion Time Selection					
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 2 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	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						27 μs
0	1	0				f_{CLK}/16					27 μs	Setting prohibited
0	1	1				f_{CLK}/8				27 μs	Setting prohibited	
1	0	0				f_{CLK}/6				20.25 μs		
1	0	1				f_{CLK}/5			33.75 μs	Setting prohibited		
1	1	0				f_{CLK}/4			27 μs			
1	1	1				f_{CLK}/2		27 μs	Setting prohibited			
0	0	0	0	1	Normal 2	f_{CLK}/64	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32						25 μs
0	1	0				f_{CLK}/16					25 μs	Setting prohibited
0	1	1				f_{CLK}/8				25 μs	Setting prohibited	
1	0	0				f_{CLK}/6			37.5 μs	18.75 μs		
1	0	1				f_{CLK}/5			31.25 μs	Setting prohibited		
1	1	0				f_{CLK}/4			25 μs			
1	1	1				f_{CLK}/2		25 μs	Setting prohibited			
Other than above						Setting prohibited						

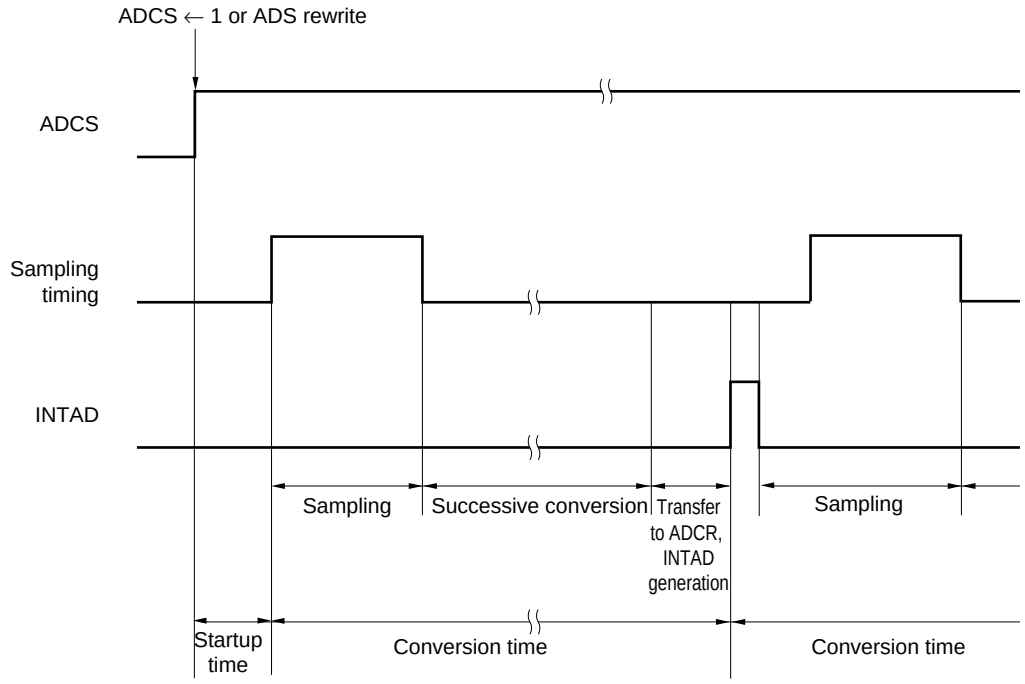
Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. The above conversion time does not include the conversion startup time. Add the conversion startup time for the first conversion. Also, the above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

3. While in the hardware trigger wait mode, the conversion time includes the time spent waiting for stabilization after the hardware trigger is detected.

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

Figure 12-5. A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



(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 12-6. 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	x	Software trigger mode
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	End of timer channel 01 count or capture interrupt signal (INTTM01)
0	1	Setting prohibited
1	0	Real-time clock interrupt signal (INTRTC)
1	1	12-bit interval timer interrupt signal (INTIT)

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

Cautions 1. Only rewrite the value of the ADM1 register while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

2. To complete A/D conversion, the following hardware trigger interval time is required:
In hardware trigger no-wait mode: Two fCLK clock cycles + A/D conversion time
In hardware trigger wait mode: Two fCLK clock cycles + stabilization wait time + A/D conversion time

3. During a mode other than SNOOZE function mode, when INTRTC or INTIT is input, the next INTRTC or INTIT input becomes effective as a trigger after a maximum of four fCLK clock cycles.

Remarks 1. ×: don't care

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

(4) A/D converter mode register 2 (ADM2)

This register is used to select the A/D converter reference voltage, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use 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 12-7. Format of A/D Converter Mode Register 2 (ADM2) (1/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	0	Supplied from V_{DD}
0	1	Supplied from P20/AV_{REFP}/ANI0
1	0	Supplied from the internal reference voltage (1.45 V)[Note]
1	1	Setting prohibited

Rewrite the values of the ADREFP1 and ADREFP0 bits in the following procedure:

1. Set ADCE to 0.
2. Change ADREFP1 and ADREFP0.
3. Count the stabilization wait time (A).
4. Set ADCE to 1.
5. Count the stabilization wait time (B).

To set ADREFP1 to 1 and ADREFP0 to 0: A = 5 μs, B = 1 μs

To set ADREFP1 to 0 and ADREFP0 to 0, or ADREFP1 to 0 and ADREFP0 to 1: A = no wait, B = 1 μs

After step 5, start A/D conversion.

When ADREFP1 = 1 and ADREFP0 = 0, A/D conversion cannot be performed for the temperature sensor output or internal reference voltage output. To perform that, set ADISS = 0.

ADREFM	Selection of the – side reference voltage source of the A/D converter
0	Supplied from V_{SS}
1	Supplied from P21/AV_{REFM}/ANI1

Note Can only be selected in HS (high-speed main) mode.

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 (<1>).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (<2>) or the ADUL register < the ADCR register (<3>).
Figure 12-8 shows the generation range of the interrupt signal (INTAD) for <1> to <3>.	

(**Cautions** are listed on the next page.)

Cautions 1. Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. Also, if the internal reference voltage is selected (ADREFP1, ADREFP0 = 1, 0), the A/D converter reference voltage current (I_{ADREF}) indicated in 32.4.2 Supply current characteristics or 33.4.2 Supply current characteristics will be added to the current consumption.

3. To use AV_{REFP} and AV_{REFM}, set ANI0 and ANI1 to analog inputs and port mode register to input mode.

Figure 12-7. Format of A/D Converter Mode Register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

AWC	Specification of 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).

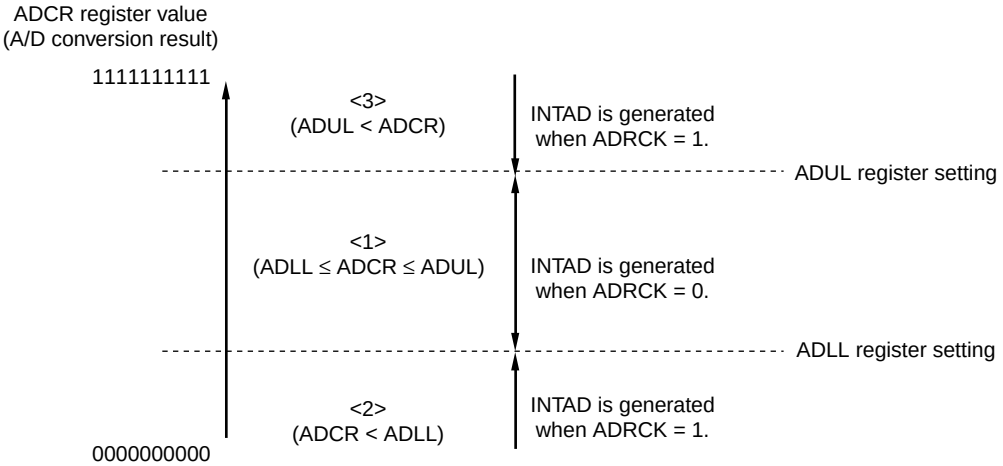
- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE mode function in the software trigger mode or hardware trigger 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 “transition time to SNOOZE mode[Note] + A/D power supply stabilization wait time + A/D conversion time + two f_{CLK} clock cycles”.
- When using the SNOOZE function in normal operation mode, set AWC to 0, and then change it to 1 immediately before a transition to STOP mode.
  Be sure to change AWC to 0 after returning from STOP mode to normal operation mode.
  If AWC remains 1, A/D conversion is not correctly started regardless whether the subsequent mode is SNOOZE mode or normal operation mode.

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note See the descriptions of “From STOP to SNOOZE” in 20.2.3, SNOOZE mode.

Caution Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

Figure 12-8. ADRCK Bit Interrupt Signal Generation Range



Remark: If INTAD is not generated, the A/D conversion results are not stored in the ADCR or ADCRH register.

(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.

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 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 (see figure 12-8)), the value is not stored.

Figure 12-9. Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Cautions 1. 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 ADCR 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.

2. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (ADCR1 and ADCR0).

3. 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.

(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.

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 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 (see figure 12-8)), the value is not stored.

Figure 12-10. 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.

(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 12-11. 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	ADISS	0	0	ADS.4	ADS.3	ADS.2	ADS.1	ADS.0

○ Select mode (ADMD = 0)

ADISS	ADS.4	ADS.3	ADS.2	ADS.1	ADS.0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin
0	0	0	1	1	0	ANI6	P26/ANI6 pin
0	0	0	1	1	1	ANI7	P27/ANI7 pin
0	1	0	0	0	0	ANI16	P03/ANI16 pin[Note1]
0	1	0	0	0	1	ANI17	P02/ANI17 pin[Note2]
0	1	0	0	1	0	ANI18	P147/ANI18 pin
0	1	0	0	1	1	ANI19	P120/ANI19 pin
1	0	0	0	0	0	–	Temperature sensor 0 output[Note3]
1	0	0	0	0	1	–	Internal reference voltage output (1.45 V)[Note3]
Other than the above						Setting prohibited	

Notes 1. P01/ANI16 pin is used in the 20, 30, or 32-pin product.

2. P00/ANI17 pin is used in the 20, 30, or 32-pin product.

3. Can only be selected in HS (high-speed main) mode.

○ Scan mode (ADMD = 1)

ADS.4	ADS.3	ADS.2	ADS.1	ADS.0	Analog input channel			
					Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
Other than the above					Setting prohibited			

(**Cautions** are listed on the next page.)

Cautions 1. Be sure to clear bits 5 and 6 to 0.

2. Set a channel to be used for A/D conversion in the input mode by using port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14).

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.

4. Do not set the pin that is set by port mode control register 0, 12, or 14 (PMC0, PMC12, PMC14) as digital I/O by the ADS register.

5. Only rewrite the value of the ADISS bit while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

6. If using AV_{REFP} as the + side reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.

7. If using AV_{REFM} as the – side reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.

8. If ADISS is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage source.

9. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. Also, if the internal reference voltage is selected (ADREFP1, ADREFP0 = 1, 0), the A/D converter reference voltage current (I_{ADREF}) indicated in 32.4.2 Supply current characteristics or 33.4.2 Supply current characteristics will be added to the current consumption.

10. The corresponding ANI pin does not exist depending on the product. In this case, ignore the conversion result.

Remark ×: don't care

(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 12-8**).

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

Reset signal generation sets this register to FFH.

Caution When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL register.

Figure 12-12. 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	ADUL.7	ADUL.6	ADUL.5	ADUL.4	ADUL.3	ADUL.2	ADUL.1	ADUL.0

(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 12-8**).

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

Reset signal generation clears this register to 00H.

Figure 12-13. 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	ADLL.7	ADLL.6	ADLL.5	ADLL.4	ADLL.3	0	ADLL.1	ADLL.0

Caution When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADLL register.

(10) A/D test register (ADTES)

This register is used to select the + side reference voltage (AV_{REFP}) or - side reference voltage (AV_{REFM}) of the A/D converter, or the analog input channel (ANIxx) as the A/D conversion target for the A/D test function.

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

Reset signal generation clears this register to 00H.

Figure 12-14. 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	ADTES.1	ADTES.0

ADTES.1	ADTES.0	A/D conversion target
0	0	ANIxx (This is specified using the analog input channel specification register (ADS).)
1	0	AV_{REFM}
1	1	AV_{REFP}
Other than above		Setting prohibited

(11) A/D port configuration register (ADPC)

This register switches the ANI0/P20 to ANI7/P27 pins to analog input of A/D converter or digital I/O of port.

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

Reset signal generation clears this register to 00H.

Figure 12-15. 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	ADPC.3	ADPC.2	ADPC.1	ADPC.0

ADPC.3	ADPC.2	ADPC.1	ADPC.0	Analog input (A)/digital I/O (D) switching							
				ANI7/P27	ANI6/P26	ANI5/P25	ANI4/P24	ANI3/P23	ANI2/P22	ANI1/P21	ANI0/P20
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	D	D	D	D	D	D	D	D
0	0	1	0	D	D	D	D	D	D	D	A
0	0	1	1	D	D	D	D	D	D	A	A
0	1	0	0	D	D	D	D	D	A	A	A
0	1	0	1	D	D	D	D	A	A	A	A
0	1	1	0	D	D	D	A	A	A	A	A
0	1	1	1	D	D	A	A	A	A	A	A
1	0	0	0	D	A	A	A	A	A	A	A
Other than above				Setting prohibited							

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode registers 2 (PM2).

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

3. To use AV_{REFP} and AV_{REFM}, set ANI0 and ANI1 to analog inputs and port mode register to input mode.

(12) Port mode control registers 0, 12, and 14 (PMC0, PMC12, PMC14)

These registers are used to switch the ANI16-ANI19 pins between A/D converter analog input and digital I/O.

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

Reset signal generation sets these registers to FFH.

Figure 12-16. Formats of Port Mode Control Registers 0, 12, and 14 (PMC0, PMC12, PMC14)

Address: F0060H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PMC0	1	1	1	1	PMC03 Note2	PMC02 Note2	PMC01 Note1	PMC00 Note1

Address: F006CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PMC12	1	1	1	1	1	1	1	PMC12.0

Address: F006EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PMC14	PMC14.7	1	1	1	1	1	1	1

PMCmn	Pmn pin digital I/O/analog input selection (m = 0, 12, 14; n = 0, 2, 3, 7)
0	Digital I/O (dual-use function other than analog input)
1	Analog input

Notes 1. 20, 30, or 32-pin products only

2. 64-pin products only

3. When the port is set to analog input with the PMC register, the port should be set to input mode with the port mode register (PMx).

(13) Port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14)

When using the pin ANI0 to ANI7 or ANI16 to ANI19 for an analog input port, set the PM20 to PM27, PM03, PM02, PM147, or PM120 bit to 1. The output latches of P20 to P27, P03, P02, P147, and P120 at this time may be 0 or 1.

If the PM20 to PM27, PM03, PM02, PM147, and PM120 bits are set to 0, they cannot be used as analog input port pins.

The PM0, PM2, PM12, and PM14 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Caution If a pin is set as an analog input port, not the pin level but "0" is always read.

Figure 12-17. Formats of Port Mode Registers 0, 2, 12, and 14 (PM0, PM2, PM12, PM14)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	PM0.6	PM0.5	PM0.4	PM0.3	PM0.2	PM0.1	PM0.0

Address: FFF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM2.7	PM2.6	PM2.5	PM2.4	PM2.3	PM2.2	PM2.1	PM2.0

Address: FFF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	1	1	PM12.0

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	PM14.7	PM14.6	1	1	1	1	PM14.1	PM14.0

PMm.n	Pmn pin I/O mode selection (mn = 02, 03, 20 to 27, 120, 140, 141, 147)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution To use AV_{REFP} and AV_{REFM}, set ANI0 and ANI1 to analog inputs and port mode register to input mode.

The ANI0/P20 to ANI7/P27 pins are as shown below depending on the settings of the A/D port configuration register (ADPC), analog input channel specification register (ADS), and PM2 registers.

Table 12-4. Setting Functions of ANI0/P20 to ANI7/P27 Pins

ADPC	PM2	ADS	ANI0/P20 to ANI7/P27 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input 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.	

The ANI16 to ANI19 pins are as shown below depending on the settings of port mode control registers 0, 12, and 14 (PMC0, PMC12, PMC14), analog input channel specification register (ADS), PM0, PM12, and PM14 registers.

Table 12-5. Setting Functions of ANI16/P03, ANI17/P02, ANI18/P147, and ANI19/P120 Pins

PMC0, PMC12, and PMC14	PM0, PM12, and PM14	ADS	ANI16/P03, ANI17/P02, ANI18/P147, and ANI19/P120 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input 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.	

12.4 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[Note1].
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated[Note1].

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0[Note2].
To stop the A/D converter, clear the ADCS bit to 0.

Notes 1. 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. In this case, the result value is not stored in the ADCR **or** ADCRH register.

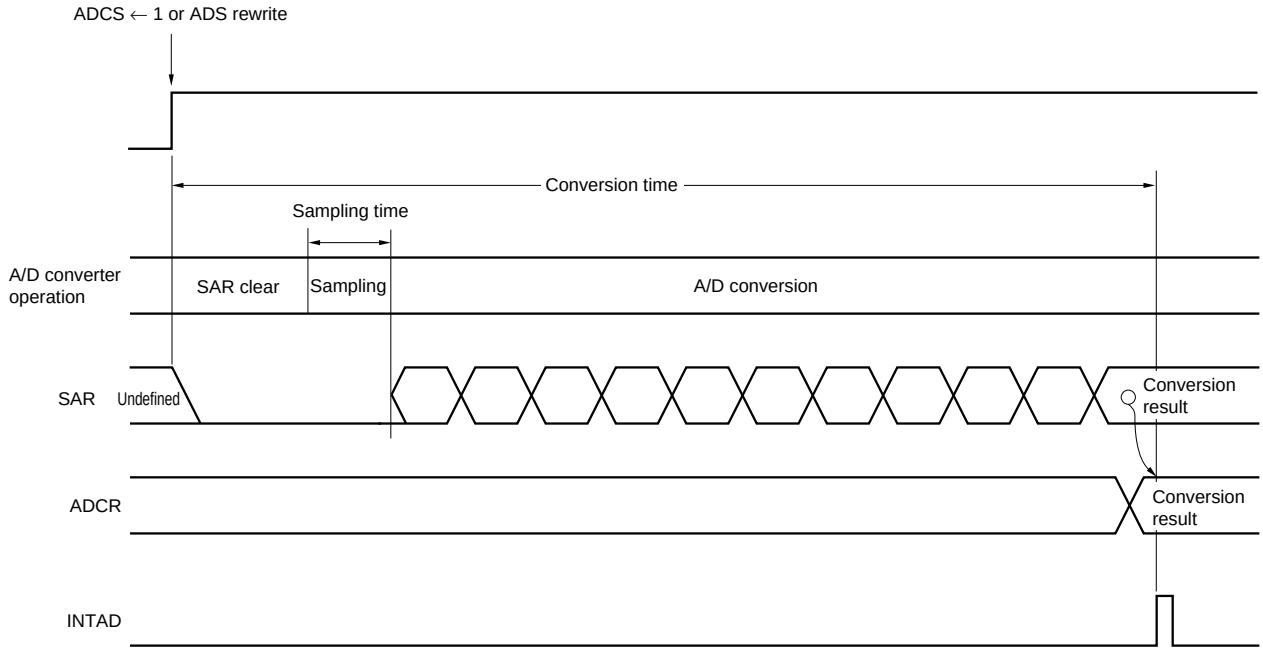
2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remarks 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

2. AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

Figure 12-18. Conversion Operation of A/D Converter (Software Trigger Mode)



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to the analog input channel specification register (ADS) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

12.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7, ANI16 to ANI19) 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

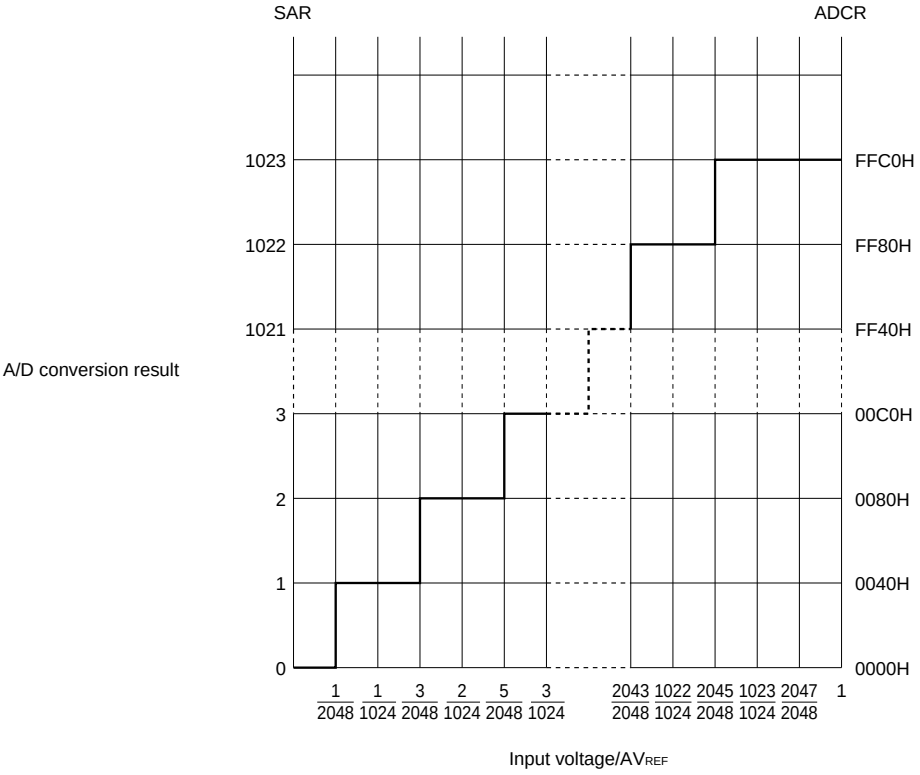
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 12-19 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-19. Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

12.6 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 **12.7 A/D Converter Setup Flowchart**.

12.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the power-down 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.

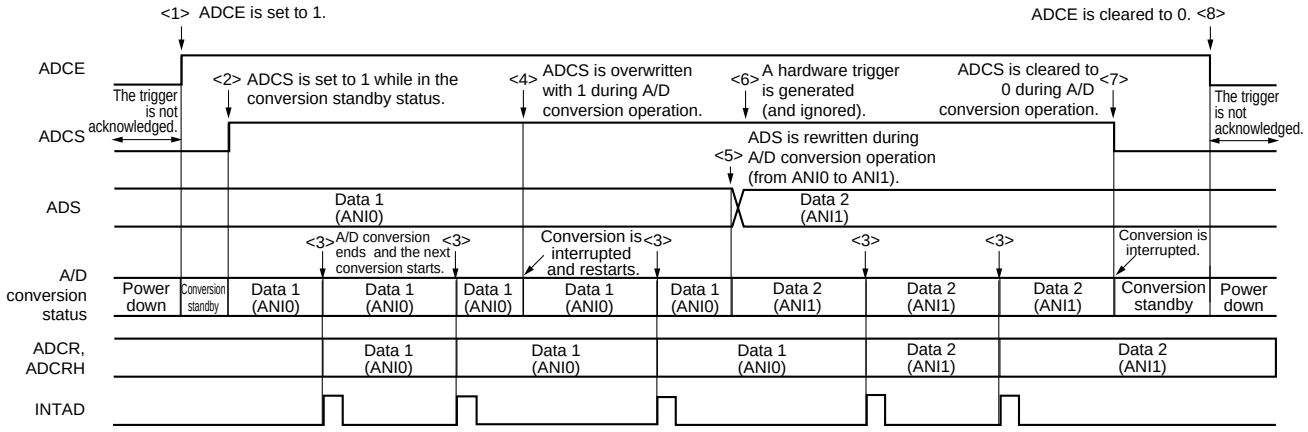
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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 power-down status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 12-20. Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the power-down 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.

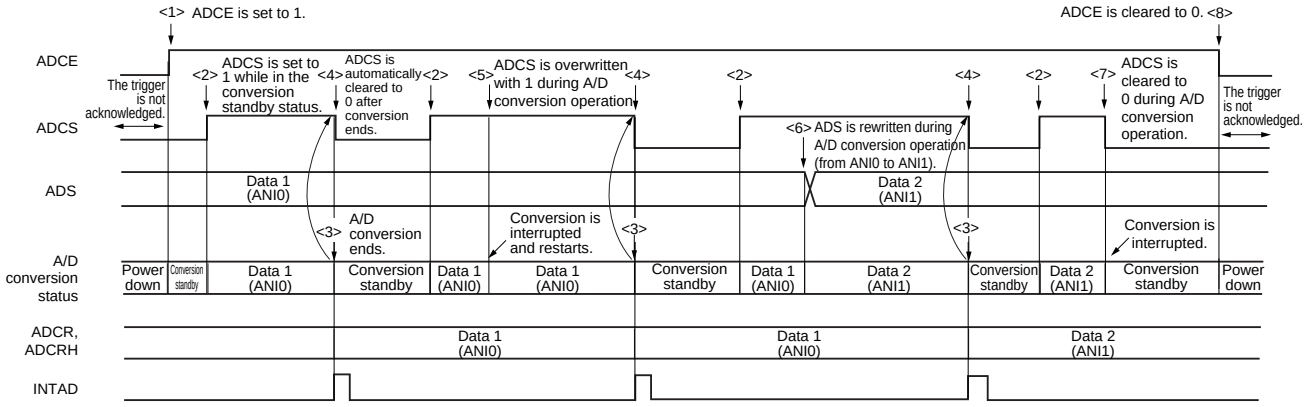
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 power-down 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 12-21. Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



12.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

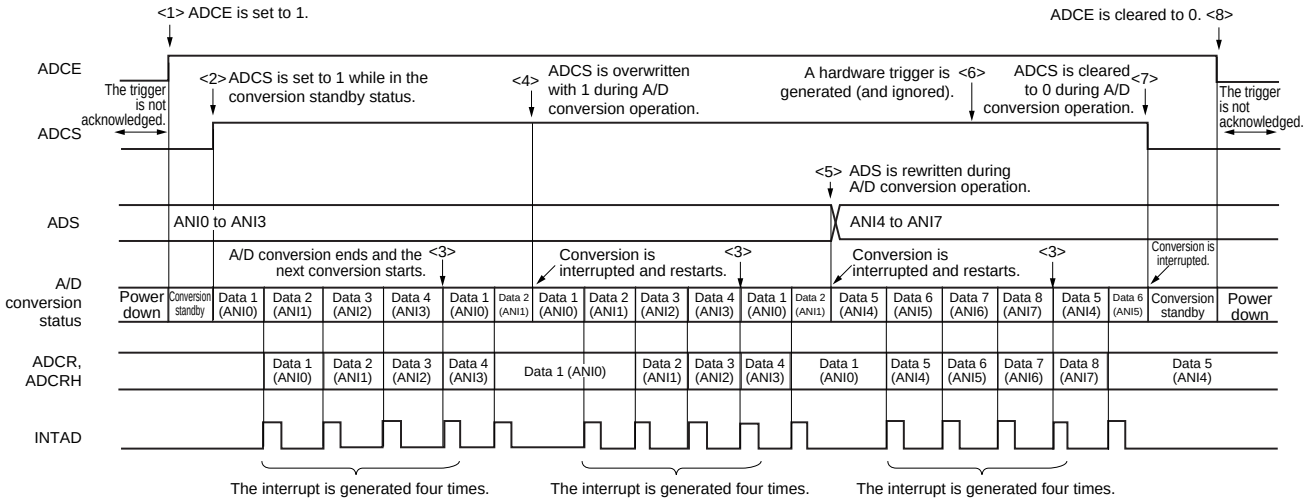
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 12-22. Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

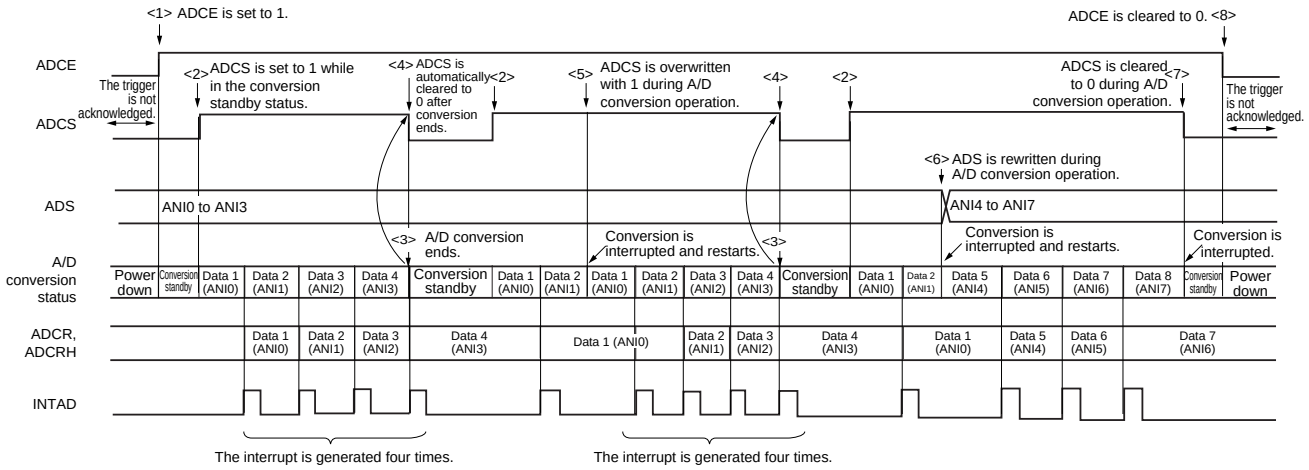
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down 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 12-23. Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



12.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the power-down 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.

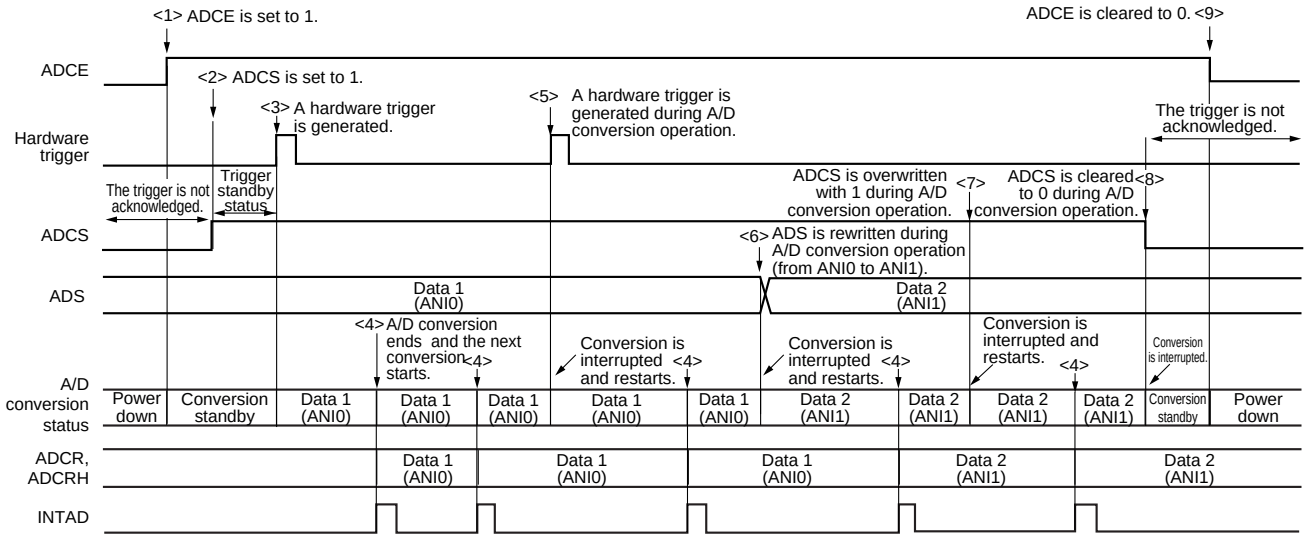
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<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 power down in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-24. Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

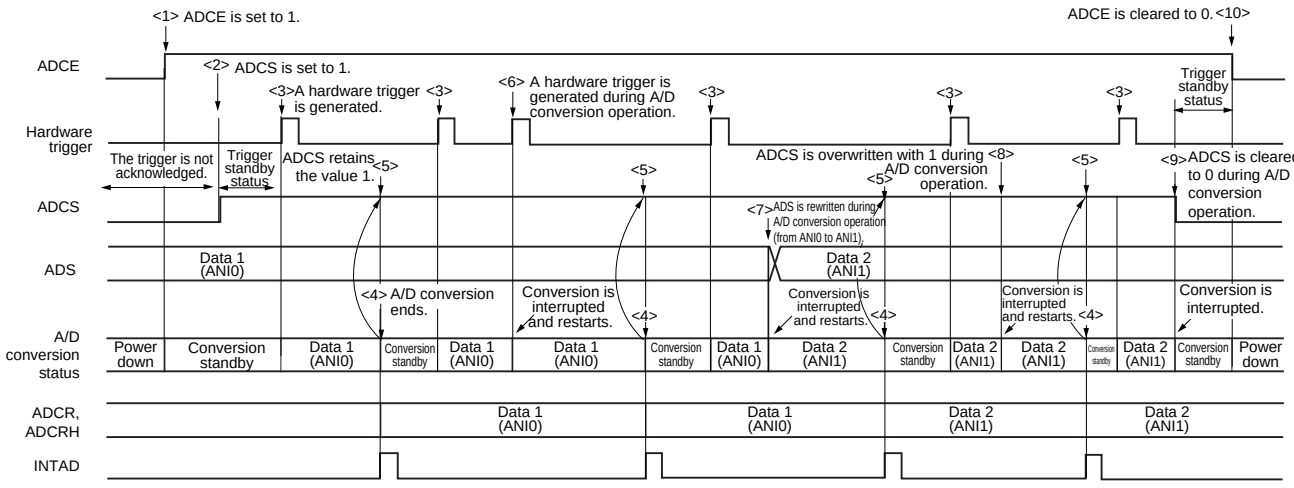
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not power down in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-25. Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



12.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

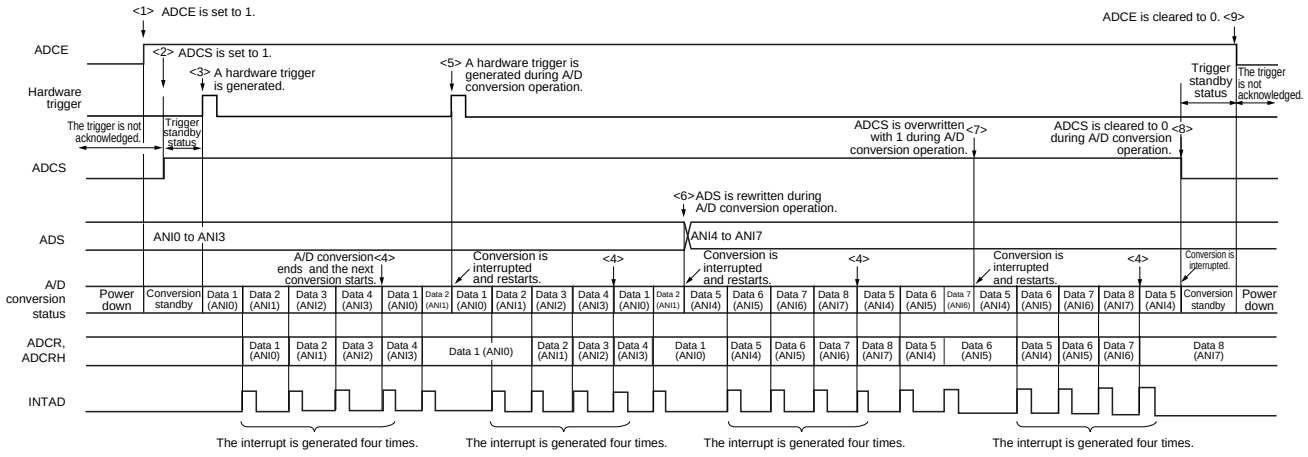
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not power down in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 12-26. Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the power-down 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.

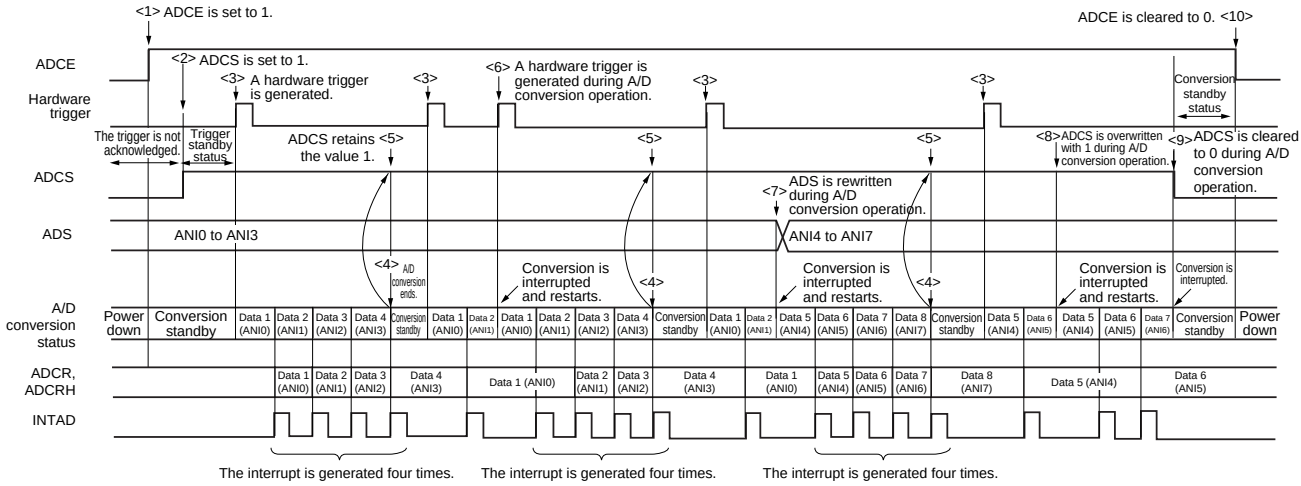
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. 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 power down in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the power-down status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-27. Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



12.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> In the power-down 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.)

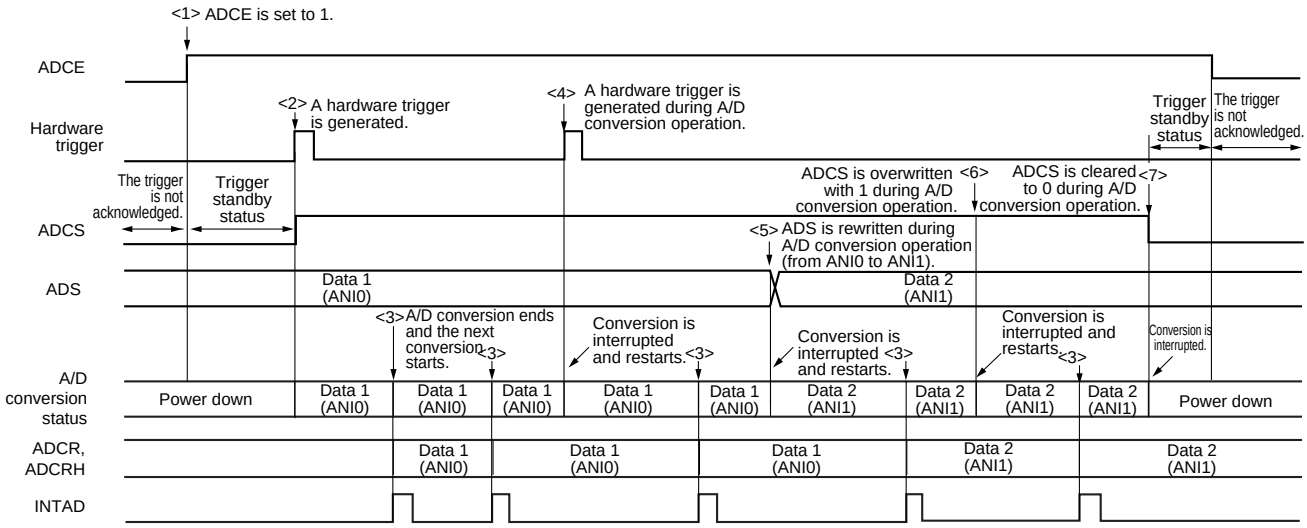
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 power-down status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-28. Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> In the power-down 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 power-down status.

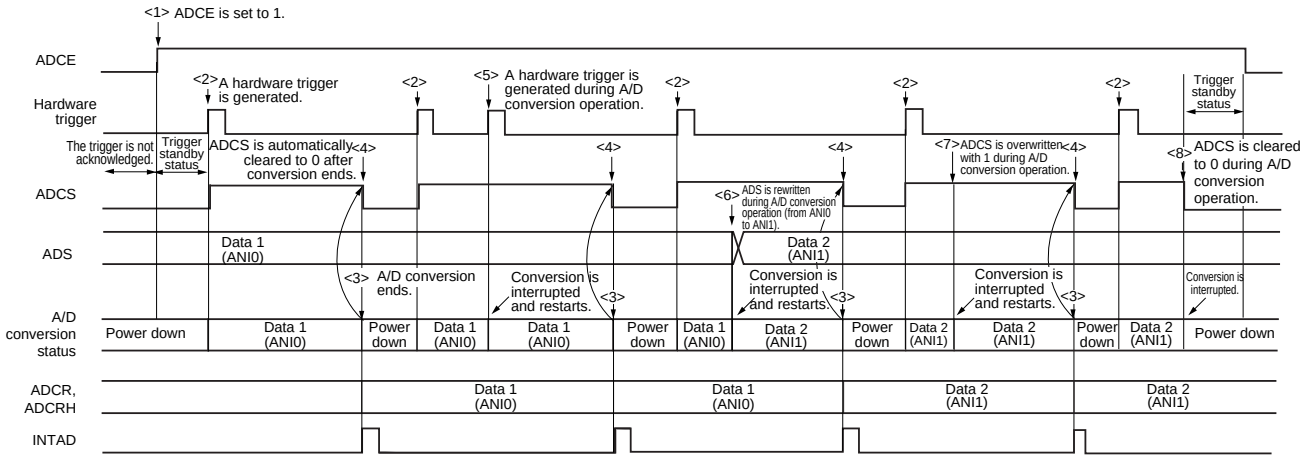
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 power-down status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-29. Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



12.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

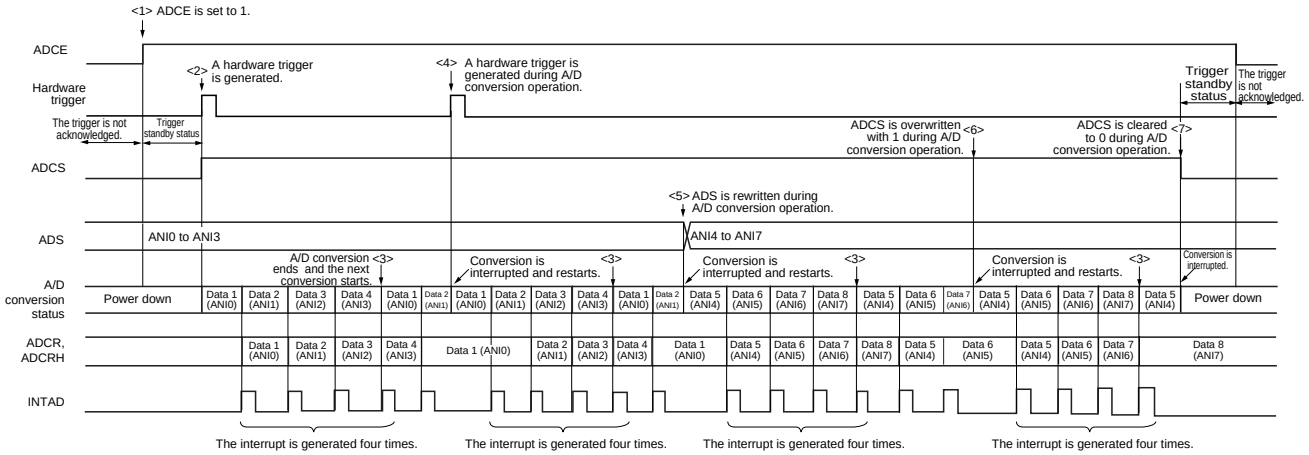
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the power-down status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-30. Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the power-down status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the power-down status.

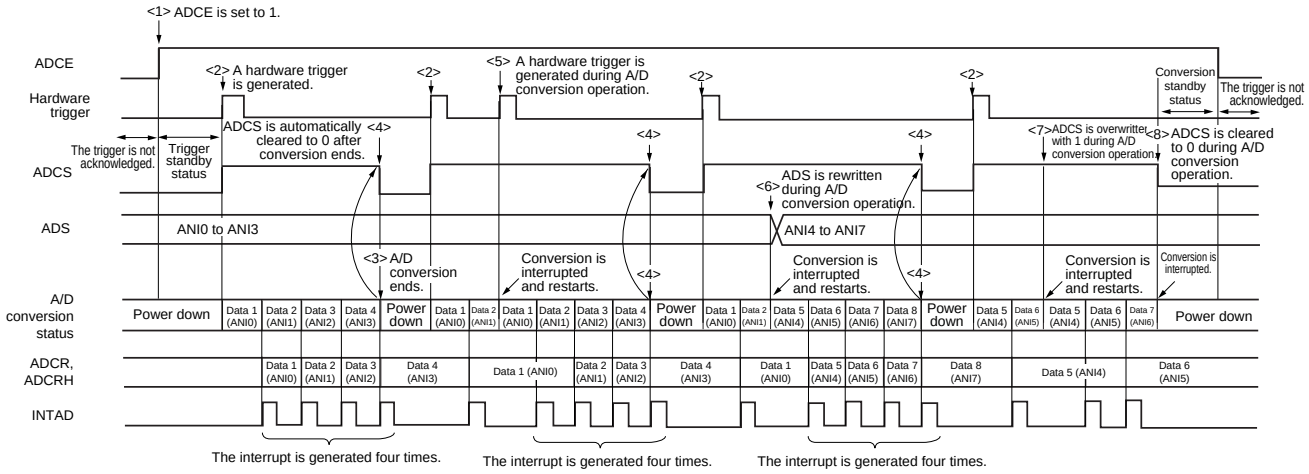
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the power-down status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12-31. Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing

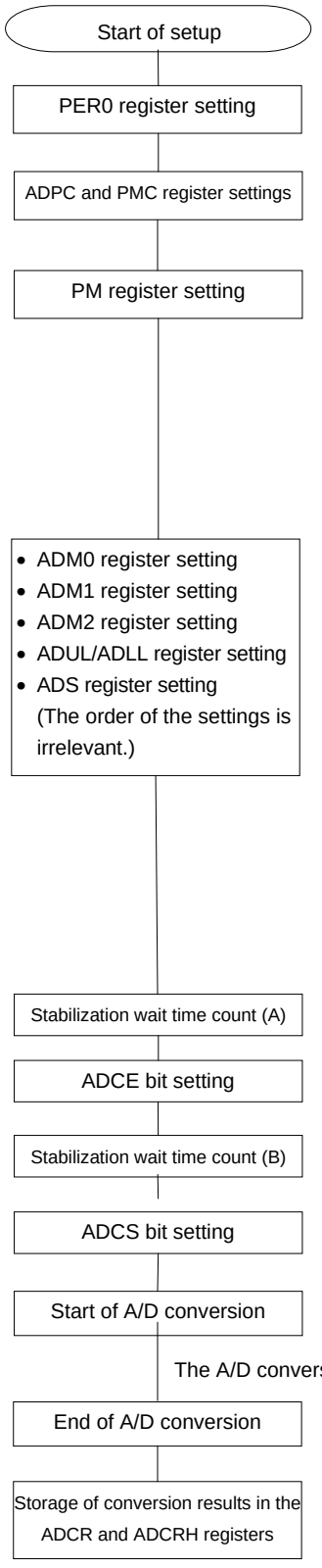


12.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

12.7.1 Setting up software trigger mode

Figure 12-32. Setting up Software Trigger Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI7 pins: Set using the ADPC register
ANI16 to ANI19 pins: Set using the PMC register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: Select mode/scan mode
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage source.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS.4 to ADS.0 bits: These are used to select the analog input channels.

The stabilization wait time A is necessary for changing ADREFP1 and ADREFP0.
A = 5 µs when ADREFP1 and ADREFP0 are set to 1 and 0, respectively.
No wait time is necessary when ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1.

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The software counts up to the stabilization wait time (1 μs).

After counting up to the stabilization wait time ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

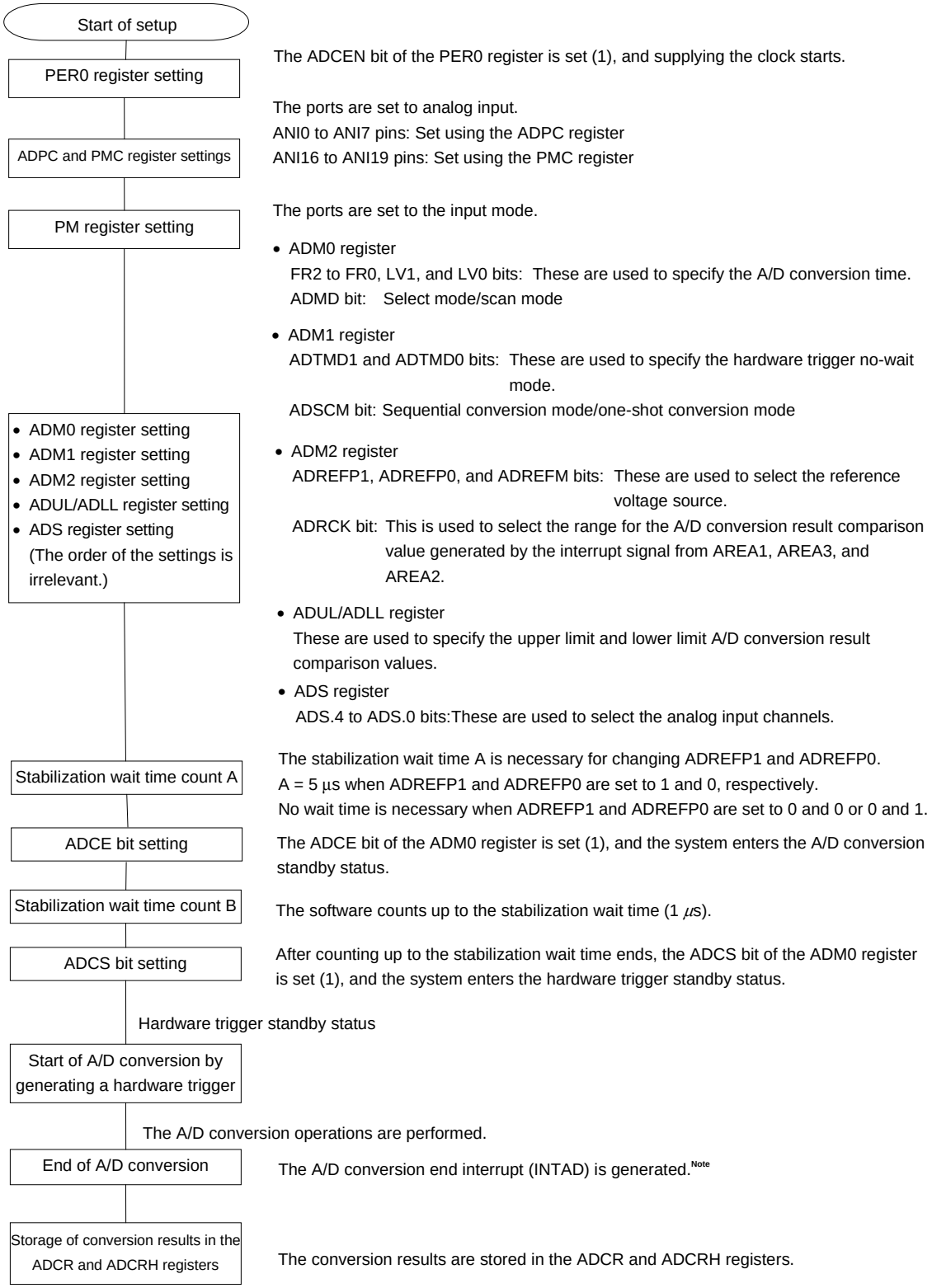
The A/D conversion end interrupt (INTAD) is generated.[Note]

The conversion results are stored in the ADCR and ADCRH registers.

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 and ADCRH registers.

12.7.2 Setting up hardware trigger no-wait mode

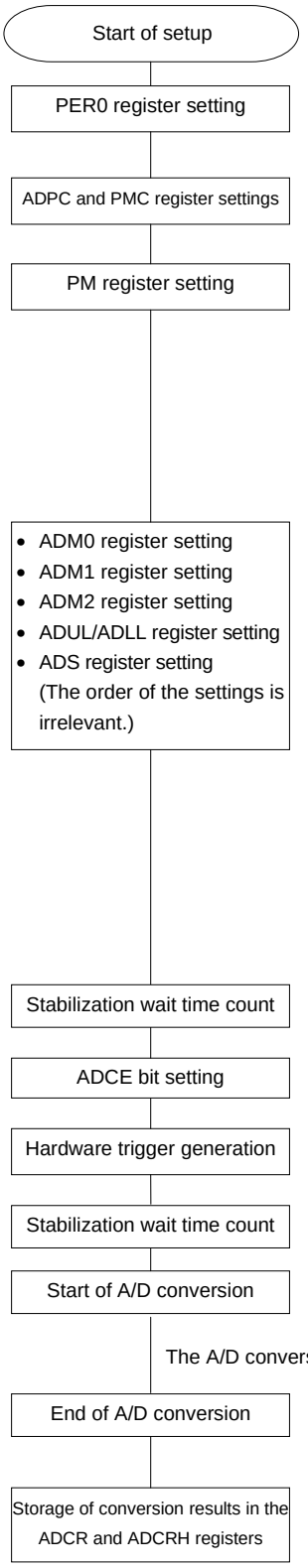
Figure 12-33. Setting up Hardware Trigger No-Wait Mode



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 and ADCRH registers.

12.7.3 Setting up hardware trigger wait mode

Figure 12-34. Setting up Hardware Trigger Wait Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI7 pins: Set using the ADPC register
ANI16 to ANI19 pins: Set using the PMC register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: Select mode/scan mode
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the hardware trigger wait mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
  ADTRS1 and ADTRS0 bits: These are used to select the hardware trigger signal.
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage source.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  AWC bit: This is used to set up the SNOOZE mode function.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS.4 to ADS.0 bits: These are used to select the analog input channels.

The stabilization wait time A is necessary for changing ADREFP1 and ADREFP0.
A = 5 µs when ADREFP1 and ADREFP0 are set to 1 and 0, respectively.
No wait time is necessary when ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1.

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The system automatically counts up to the stabilization wait time.

After counting up to the stabilization wait time ends, A/D conversion starts

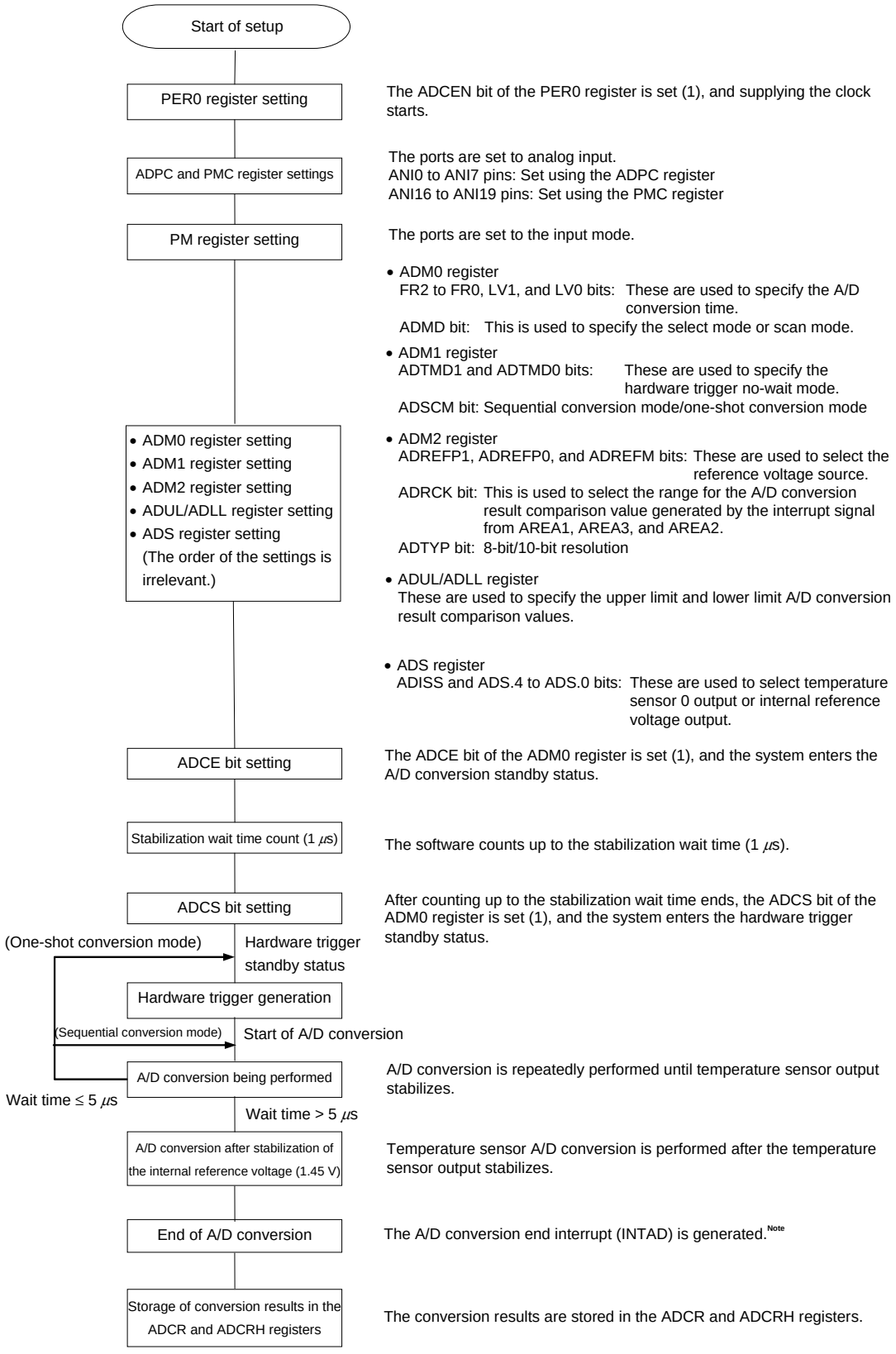
The A/D conversion end interrupt (INTAD) is generated.[Note]

The conversion results are stored in the ADCR and ADCRH registers.

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 and ADCRH registers.

12.7.4 Setup when using temperature sensor (example for hardware trigger no-wait mode)

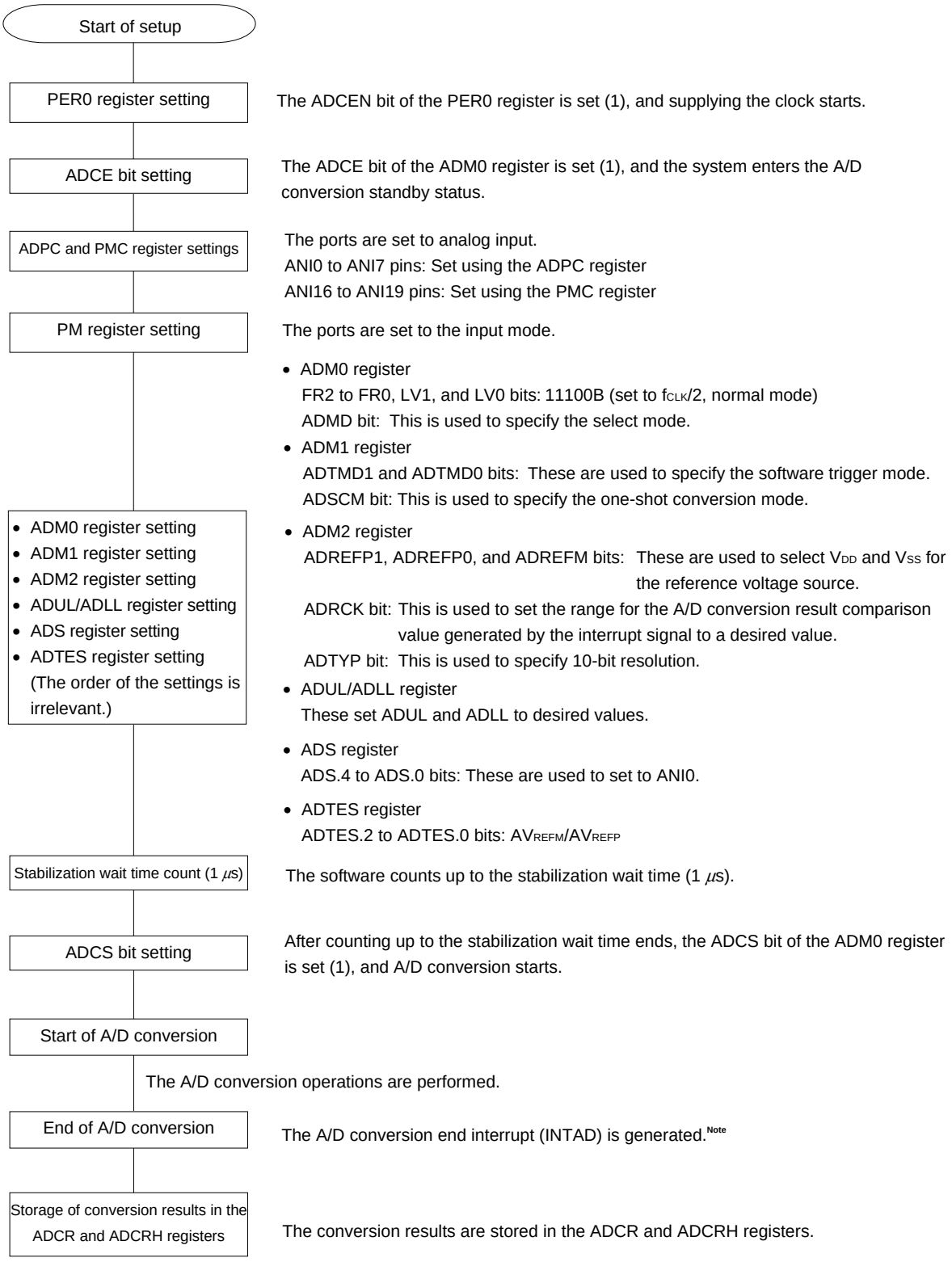
Figure 12-35. Setup When Using Temperature Sensor



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 and ADCRH registers.

12.7.5 Setting up test mode

Figure 12-36. Setting up Test Trigger Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

12.8 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.

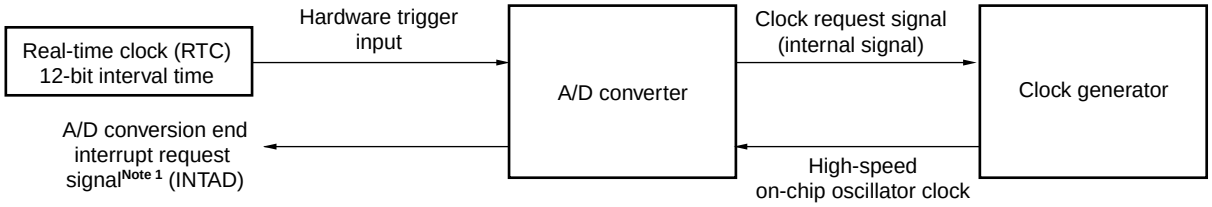
By specifying the conversion result range with ADUL and ADLL in SNOOZE mode, A/D conversion results can be checked at the specified interval, which allows the monitoring of power supply voltage and check of A/D input keys.

In the SNOOZE mode, only the following two conversion modes can be used:

- Hardware trigger wait mode (select mode, one-shot conversion mode)
- Hardware trigger wait mode (scan mode, one-shot conversion mode)

Note that the SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 12-37. 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 **12.7.3 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 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.

Notes 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.

2. Be sure to set the ADM1 register to E2H or E3H.

Remark The hardware trigger is 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 within 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.

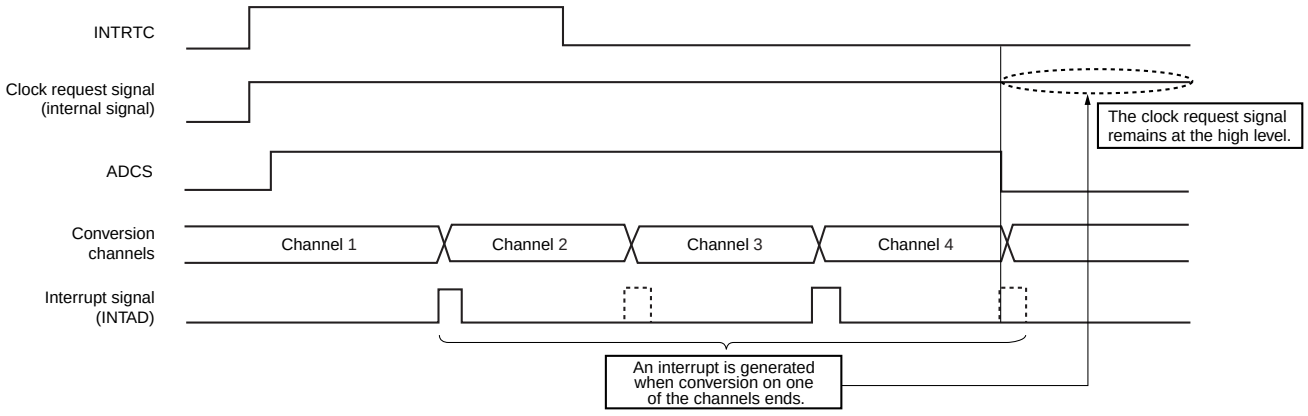
- While in the select mode

  After A/D conversion ends and the A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, clear bit 2 (AWC) of A/D converter mode register 2 (ADM2) to 0 to release SNOOZE mode. If AWC remains 1, A/D conversion is not correctly started regardless whether the subsequent mode is SNOOZE mode or normal operation mode.

- While in the scan mode

  If even one A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, clear bit 2 (AWC) of A/D converter mode register 2 (ADM2) to 0 to release SNOOZE mode. If AWC remains 1, A/D conversion is not correctly started regardless whether the subsequent mode is SNOOZE mode or normal operation mode.

Figure 12-38. Operation Example When Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)



(2) If no interrupt is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

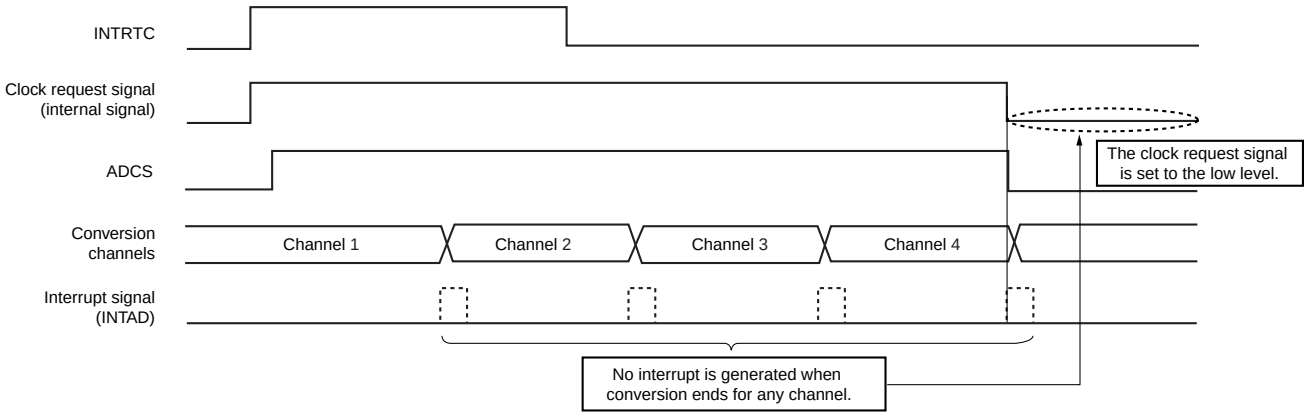
- While in the select mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

- While in the scan mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated even once during A/D conversion of the four channels, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

Figure 12-39. Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)



12.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 12-40. Overall Error

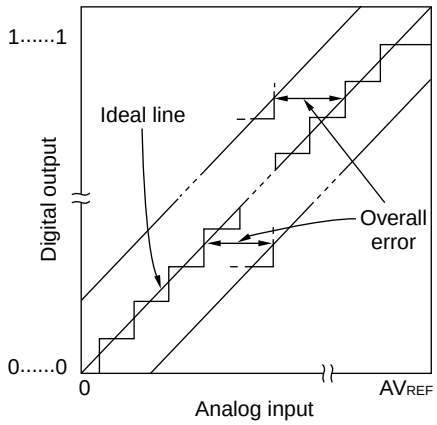
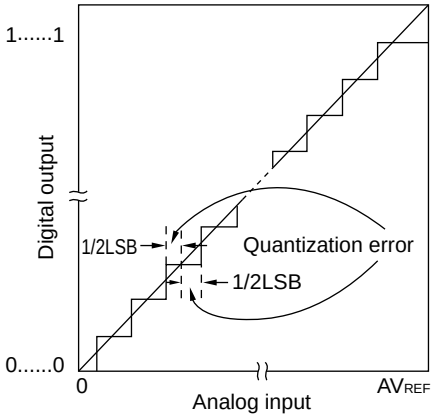


Figure 12-41. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.
If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 12-42. Zero-Scale Error

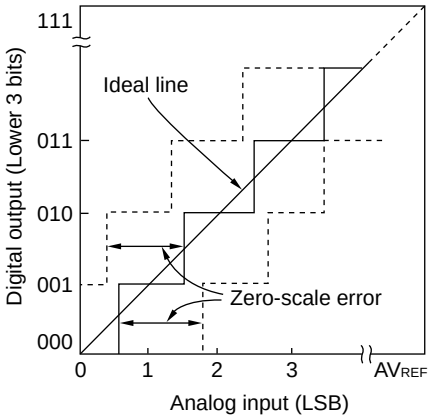


Figure 12-43. Full-Scale Error

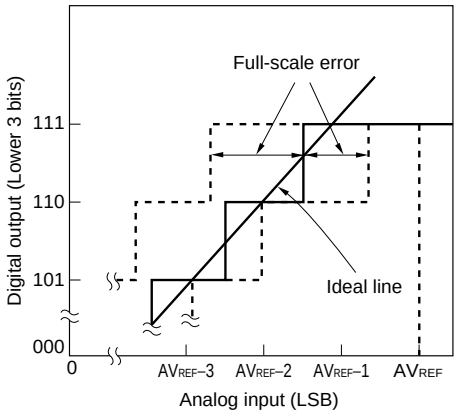


Figure 12-44. Integral Linearity Error

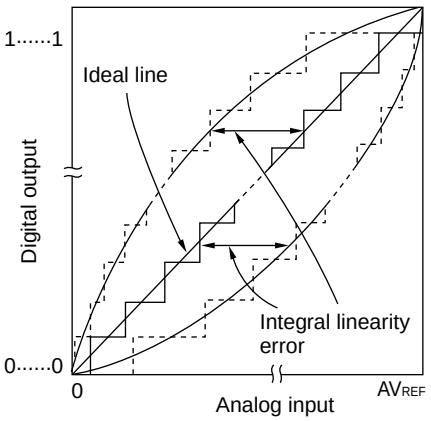
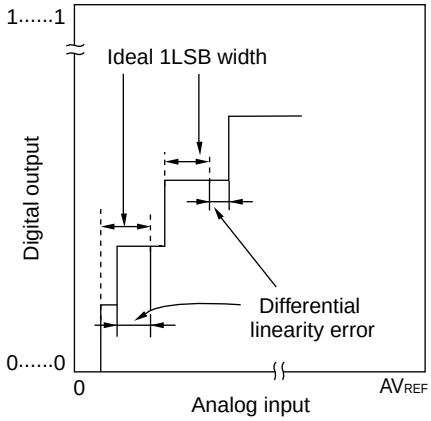


Figure 12-45. Differential Linearity Error



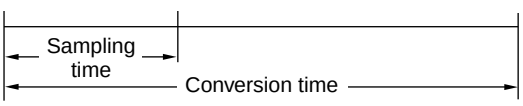
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



12.10 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

(2) Input range of ANI0 to ANI7 and ANI16 to ANI19 pins

Observe the rated range of the ANI0 to ANI7 and ANI16 to ANI19 pins input voltage. If a voltage of V_{DD} and AV_{REFP} or higher and V_{SS} and AV_{REFM} or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

If the internal reference voltage (1.45 V) is selected as the reference voltage on the plus side of the A/D converter, do not apply the voltage of 1.45 V or more to the pin selected by the ADS register. To the other pins, there are no problems if the applied voltage is 1.45 V or more.

Note The internal reference voltage (1.45 V) can only be selected in HS (high-speed main) mode.

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion

The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between the ADCR or ADCRH register write and the A/D converter mode register 0 (ADM0) write, the analog input channel specification register (ADS), 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 ANI7, and ANI16 to ANI19 pins.

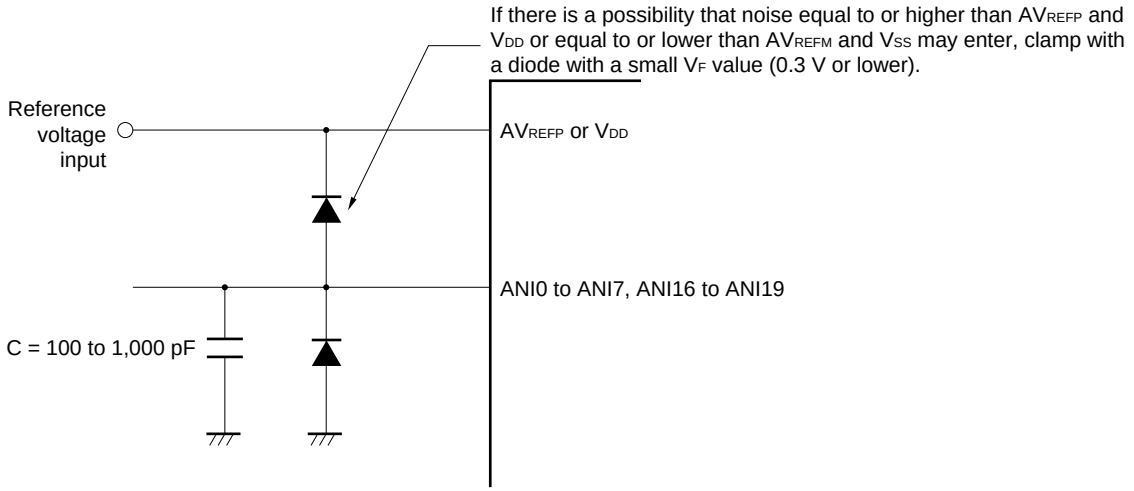
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 12-46 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 12-46. Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> The analog input pins (ANI0 to ANI7, ANI16 to ANI19) are also used as input port pins (P20 to P27, P02, P03, P147, P120).
When A/D conversion is performed with any of the ANI0 to ANI7 and ANI16 to ANI19 pins selected, do not change the output value to P20 to P27, P03, P02, P147 and P120 while conversion is in progress; otherwise the conversion accuracy may be degraded.

<2> If the pins adjacent to the pins currently used for A/D conversion is used as digital I/O ports, the expected value of the A/D conversion may not be obtained due to coupling noise. Take care not to input or output such a pulse to these pins.

(6) Input impedance of analog input (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI7 and ANI16 to ANI19 pins (see **Figure 12-46**).

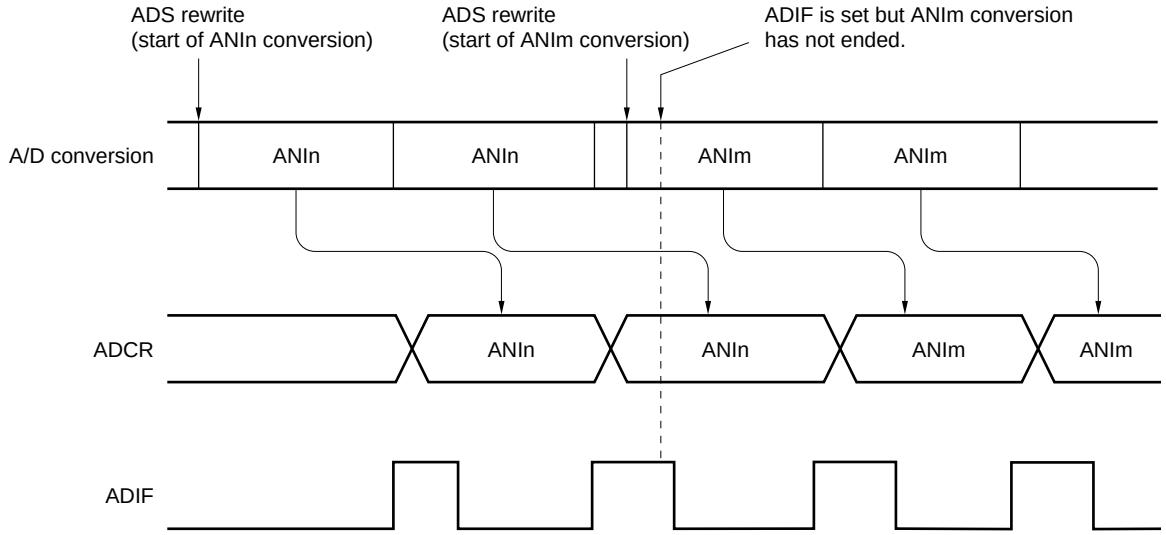
(7) Interrupt request flag (ADIF)

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 12-47. Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), A/D port configuration register (ADPC), and port mode control register (PMC), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, 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 12-48. Internal Equivalent Circuit of ANIn Pin

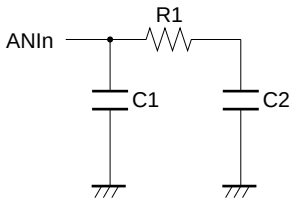


Table 12-6. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REFP}, V_{DD}	ANIn pin	R1 [kΩ]	C1 [pF]	C2 [pF]
$4.0 \leq V_{DD} \leq 5.5$	ANI0 to ANI7	14	8	2.5
	ANI16 to ANI19	18	8	7.0
$2.7 \leq V_{DD} \leq 4.0$	ANI0 to ANI7	39	8	2.5
	ANI16 to ANI19	53	8	7.0
$1.8 \leq V_{DD} \leq 2.7$	ANI0 to ANI7	231	8	2.5
	ANI16 to ANI19	321	8	7.0

Remark The resistance and capacitance values shown in Table 12-6 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 13 SERIAL ARRAY UNIT

Serial array unit 0 has four serial channels, serial array unit 1 has two, and serial array unit S has two. Each channel can achieve 3-wire serial (CSI), UART, and simplified I^2C communication.

Function assignment of each channel supported by the RL78/F12 is as shown below.

- 20-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	–		–
	2	–	–	–
	3	–		–
1	0	–	–	–
	1	–		–
S	0	CSIS0	UARTS0	–
	1	–		–

- 30, 32-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	–		–
	2	–	UART1	–
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	–		–
S	0	CSIS0	UARTS0	–
	1	–		–

- 48-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	–	UART1	–
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
S	0	CSIS0	UARTS0	–
	1	–		–

- 64-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
S	0	CSIS0	UARTS0	–
	1	CSIS1		–

When “UART0” is used for channels 0 and 1 of the unit 0, CSI00 and CSI01 cannot be used, but CSI10, UART1, or IIC10 can be used.

Caution Most of the following descriptions in this chapter use the units and channels of the 48, 64-pin products as an example.

13.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/F12 has the following features.

13.1.1 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSIS0, CSIS1)

Data is transmitted or received in synchronization with the serial clock ($\overline{SCK}$) output from the master channel.
3-wire serial communication is clocked communication performed by using three communication lines: one for the serial clock ($\overline{SCK}$), one for transmitting serial data (SO), one for receiving serial data (SI).

For details about the settings, see **13.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSIS0, CSIS1) Communication.**

[Data transmission/reception]

- Data length of 7 or 8 bits (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)
- Data length of 7 to 16 bits (CSIS0, CSIS1)
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[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
  During master communication (CSI00): Max. $f_{CLK}/2$ **Note**
  During master communication (other than CSI00): Max. $f_{CLK}/4$ **Note**
  During slave communication: Max. $f_{CLK}/6$ **Note**

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 (channel 0 of unit 0) supports the SNOOZE mode. When $\overline{SCK00}$ pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. SNOOZE mode can be specified only for CSI00 that supports asynchronous reception.

Note Use the clocks within a range satisfying the $\overline{SCK}$ cycle time (t_{KCY}) characteristics (see **CHAPTER 31, ELECTRICAL SPECIFICATIONS (J GRADE) and CHAPTER 32, ELECTRICAL SPECIFICATIONS (K GRADE)**).

13.1.2 UART (UART0 to UART2, UARTS0)

This is an asynchronous communication function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using timer array unit with an external interrupt (INTP0).

For details about the settings, see **13.6 Operation of UART (UART0 to UART2, UARTS0) Communication.**

[Data transmission/reception]

- Data length of 7, 8, or 9bits (UART0)
- Data length of 7 or 8 bits (UART1, UART2)
- Data length of 7 to 9, 16 bits (UARTS0)
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART0 reception (channel 1 of unit 0) supports the SNOOZE mode. When RxD0 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. SNOOZE mode can be specified only for UART0 that supports the reception baud rate adjustment function.

The LIN-bus is accepted in UART2 (0 and 1 channels of unit 1) (30, 32, 48, 64-pin products only).

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit

13.1.3 Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **13.8 Operation of Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) Communication**.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function Note and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- Parity error (ACK error)

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

Note When receiving the last data, ACK will not be output if 0 is written to the SOEm.n bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See the processing flow in **13.8.3 (2)** for details.

Remarks 1. To use an I²C bus of full function, see **CHAPTER 15 SERIAL INTERFACE IICA**.

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

13.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 13-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits or 9 bits Note 1 (units 0, 1), 16 bits (unit S)
Buffer register	Serial data register mn (SDRmn) Notes 1, 2
Serial clock I/O	$\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, $\overline{SCK11}$, $\overline{SCK20}$, $\overline{SCK21}$, $\overline{SCKS0}$, $\overline{SCKS1}$ pins (for 3-wire serial I/O), SCL00, SCL01, SCL10, SCL11, SCL20, SCL21 pins (for simplified I^2C)
Serial data input	SI00, SI01, SI10, SI11, SI20, SI21, SIS0, SIS1 pins (for 3-wire serial I/O), RxD0, RxD1, RxDS0 pins (for UART), RxD2 pin (for UART supporting LIN-bus)
Serial data output	SO00, SO01, SO10, SO11, SO20, SO21, SOS0, SOS1 pins (for 3-wire serial I/O), TxD0, TxD1, TxDS0 pins (for UART), TxD2 pin (for UART supporting LIN-bus), output controller
Serial data I/O	SDA00, SDA01, SDA10, SDA11, SDA20, SDA21 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable registers 0, X (PER0, PERX) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Serial standby control register 0 (SSC0) • Input switch control register (ISC) • Noise filter enable registers 0, X (NFEN0, NFENX)
	<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, 1, 5 (PIM0, PIM1, PIM5) • Port output mode registers 0, 1, 5, 7 (POM0, POM1, POM5, POM7) • Port mode registers 0, 1, 3, 5, 7, X0 to X4 (PM0, PM1, PM3, PM5, PM7, PMX0 to PMX4) • Port registers 0, 1, 3, 5, 7 (P0, P1, P3, P5, P7)

Notes 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel. mn = 00, 01: lower 9 bits, mn = 02, 03, 10, 11: lower 8 bits, mn = S0, S1: 16 bits

2. 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)
- IICr communication … SIOr (IICr data register)

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), q: UART number (q = 0 to 2, S0), r: IIC number (r = 00, 01, 10, 11, 20, 21)

Figure 13-1 shows the block diagram of the serial array unit 0.

Figure 13-1. Block Diagram of Serial Array Unit 0

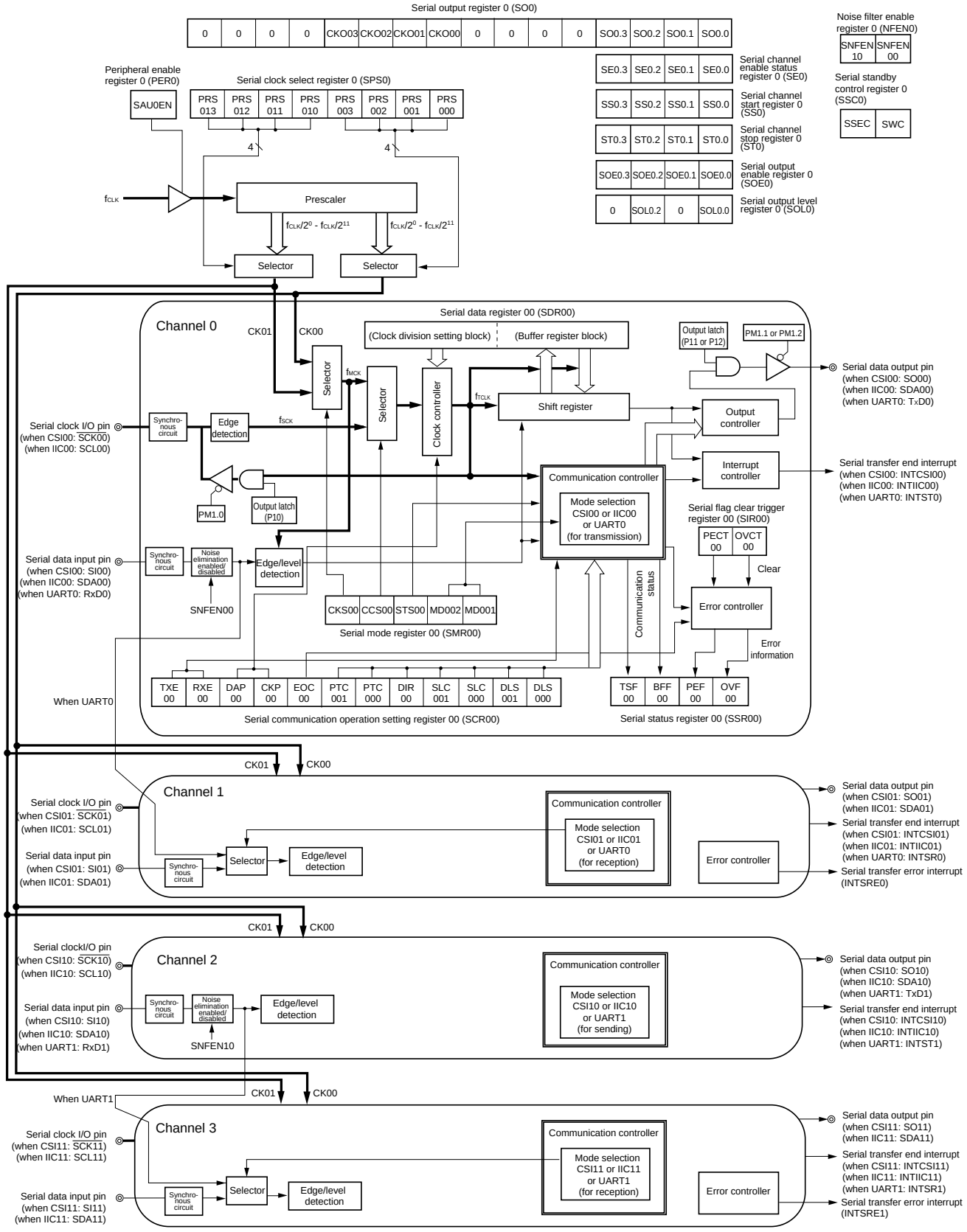


Figure 13-2 shows the block diagram of the serial array unit 1.

Figure 13-2. Block Diagram of Serial Array Unit 1

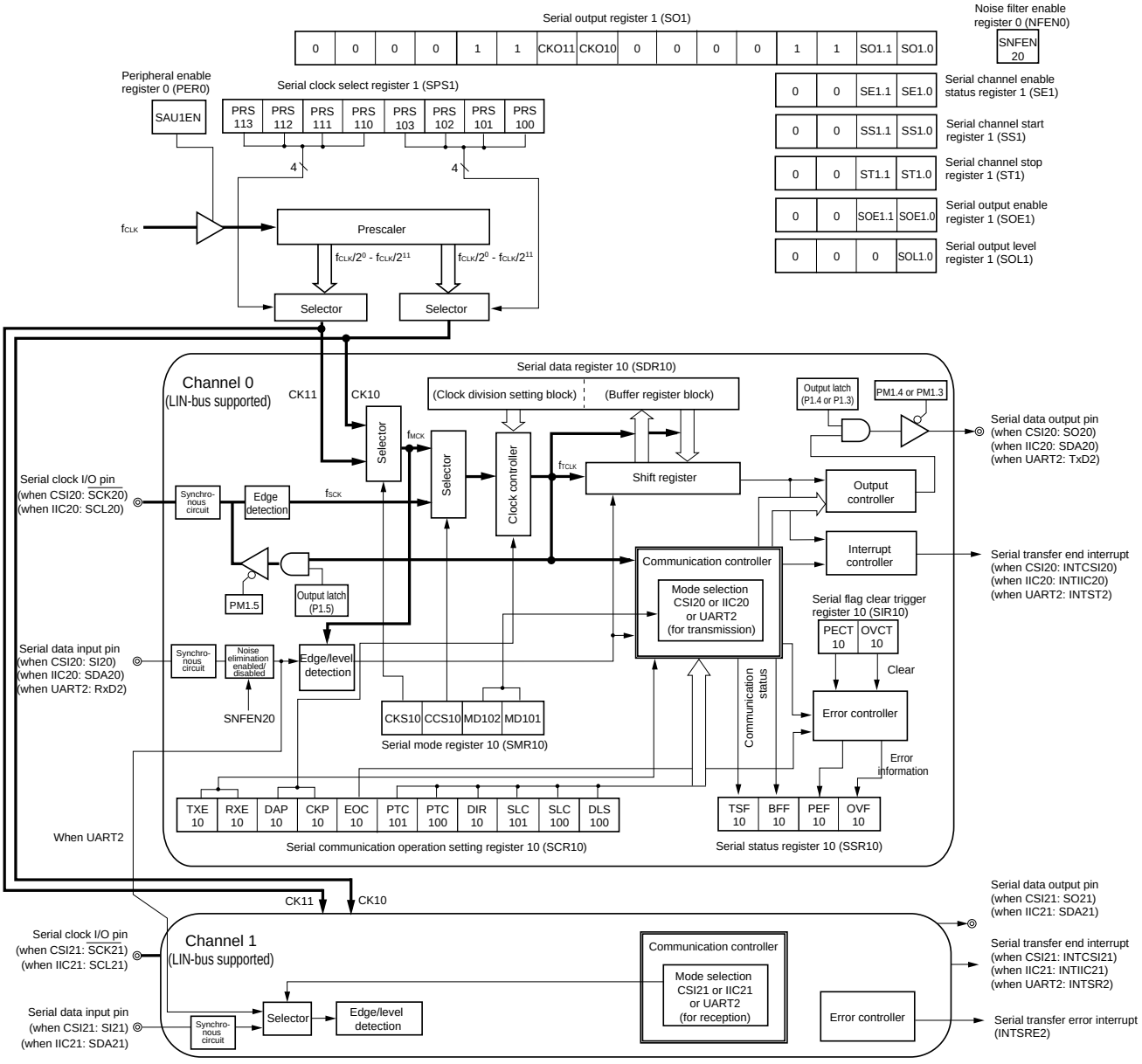
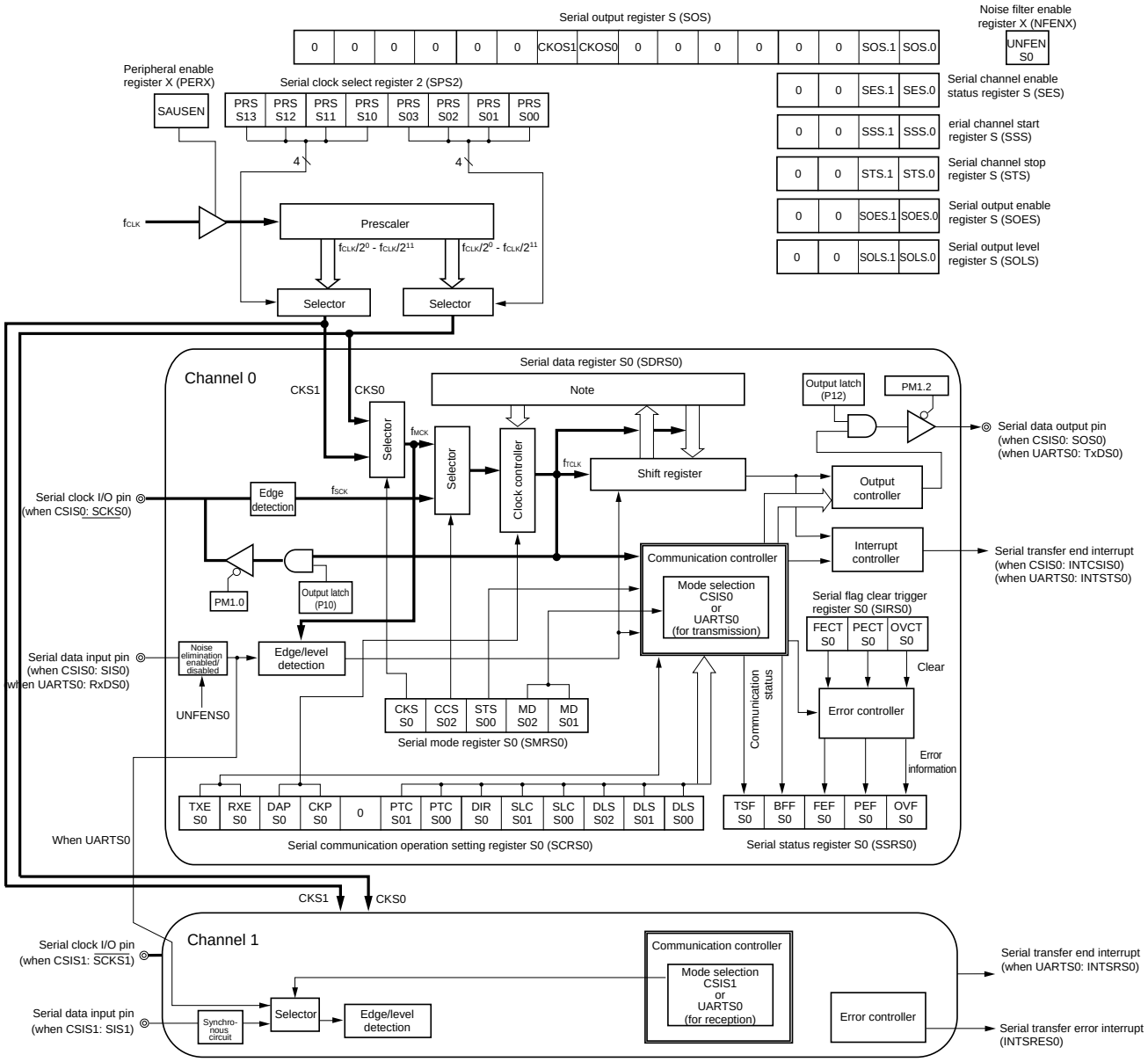


Figure 13-3 shows the block diagram of the serial array unit S.

Figure 13-3. Block Diagram of Serial Array Unit S



Note Only when the operation is stopped (SES.n = 0), the upper 7 bits become the clock division setting part.

(1) Shift register

This is an 8-/16-bit register that converts parallel data into serial data or vice versa.

In case of the UART communication of nine bits of data using UART0, nine bits (bits 0 to 8) are used.

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 serial data register mn (SDRmn) (in units 0 and 1, the lower 8/9 bits only).

- Units 0, 1

	8	7	6	5	4	3	2	1	0
Shift register									

- Unit S

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

(2) Serial data register mn (SDRmn)

(a) Lower 8/9 bits of the serial data register mn (SDRmn) in units 0 and 1

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 or bits 7 to 0 (lower 8 bits) of SDR02, SDR03, SDR10 and SDR11 function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 8/9 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 8/9 bits.

The data stored in the lower 8/9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 9-bit data length (stored in bits 0 to 8 of SDRmn register(mn = 00, 01)) (settable in UART0 mode only)
- 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 8/9 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)
- IICr communication … SIOr (IICr data register)

Reset signal generation clears the SDRmn register to 0000H.

Note Writing in 8-bit units is prohibited when the operation is stopped (SEm.n = 0).

Remarks 1. After data is received, “0” is stored in bits 0 to 8 in bit portions that exceed the data length.

2. m: Unit **number** (m = 0, 1),
n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21),
q: UART number (q = 0 to 2),
r: IIC number (r = 00, 01, 10, 11, 20, 21)

(b) Serial data register mn (SDRSn) in unit S

SDRSn is the transmit/receive data register (16 bits) of channel n. When operation is stopped (SES.n = 0), bits 15 to 9 are used as the division setting register of the operating clock (f_{MCK}). During operation (SES.n = 1), bits 15 to 9 are used as a transmission/reception buffer register.

When data is received, parallel data converted by the shift register is stored. When data is to be transmitted, set transmit to be transferred to the shift register.

The data stored in this register is as follows, depending on the setting of bits 4 to 0 (DLSSn4 to DLSSn0) of the SCRSn register, regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDRSn register)
- 8-bit data length (stored in bits 0 to 7 of SDRSn register)

  :
- 16-bit data length (stored in bits 0 to15 of SDRSn register)

SDRSn can be read or written in 16-bit units.

When SES.n = 1, the lower 8 bits of SDRSn can be read or written[Note] in 8-bit units as SDRSnL. The SDRSnL registers that can be used according to the communication methods are shown below.

- CSISn communication … SIOSnL (CSISn data register)
- UARTS0 reception … RXDS0 (UARTS0 receive data register)
- UARTS0 transmission … TXDS0 (UARTS0 transmit data register)

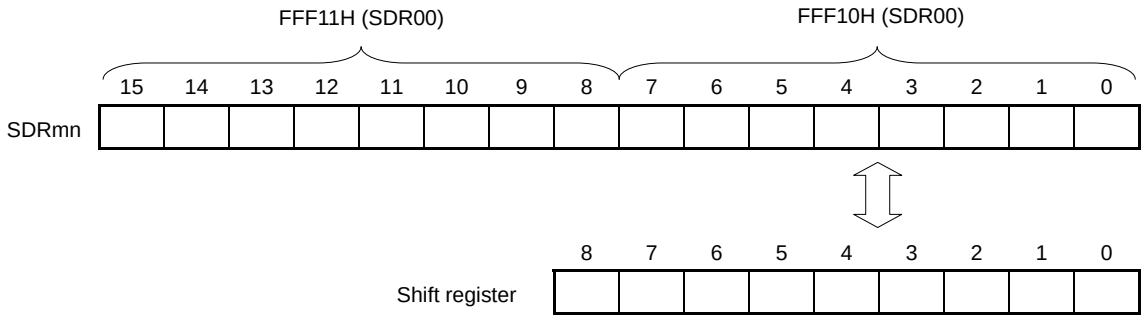
Reset signal generation clears this register to 0000H.

Note Writing in 8-bit units is prohibited when the operation is stopped (SES.n = 0).

Remark n: Channel number (n = 0, 1)

Figure 13-4. Format of Serial Data Register mn (SDRmn) (mn = 00-03, 10, 11)

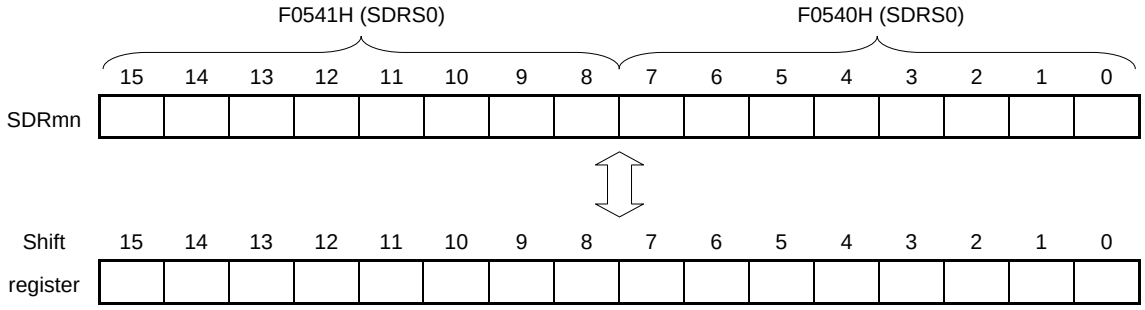
Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), After reset: 0000H R/W
FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)



Remark For the function of the higher 7 bits of the SDRmn register, see **13.3 Registers Controlling Serial Array Unit**.

Figure 13-5. Format of Serial Data Register mn (SDRmn) (mn = S0, S1)

Address: F0540H, F0541H (SDRS0), F0542H, F0543H (SDRS1), After reset: 0000H R/W



Remark For the function of the higher 7 bits of the SDRmn register, see **13.3 Registers Controlling Serial Array Unit**.

13.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable registers 0, X (PER0, PERX)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Serial standby control register 0 (SSC0)
- Input switch control register (ISC)
- Noise filter enable registers 0, X (NFEN0, NFENX)
- Port input mode registers 0, 1, 5 (PIM0, PIM1, PIM5)
- Port output mode registers 0, 1, 5, 7 (POM0, POM1, POM5, POM7)
- Port mode registers 0, 1, 3, 5, 7, X0 to X4 (PM0, PM1, PM3, PM5, PM7, PMX0 to PMX4)
- Port registers 0, 1, 3, 5, 7 (P0, P1, P3, P5, P7)

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

(1) Peripheral enable registers 0, X (PER0, PERX)

PER0, PERX are 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 PER0 register to 1.

When serial array unit 1 is used, be sure to set bit 3 (SAU1EN) of PER0 register to 1.

When serial array unit S is used, be sure to set bit 1 (SAUSEN) of PERX register to 1.

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

Reset signal generation clears the PER0, PERX registers to 00H.

Figure 13-6. 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	0	ADCEN	IICA0EN Note	SAU1EN Note	SAU0EN	0	TAU0EN

Address: F0500H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PERX	0	0	0	0	0	UF0EN	SAUSEN	WUTMEN

SAUmEN	Control of serial array unit m input clock supply (m = 0, 1, S)
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.

Note Those are not provided in the 20-pin products.

Cautions 1. When setting serial array unit m, be sure to set the SAUmEN bit to 1 first. 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 the input switch control register (ISC), noise filter enable registers 0, X (NFEN0, NFENX), port input mode registers 0, 1, 5 (PIM0, PIM1, PIM5), port output mode registers 0, 1, 5, 7 (POM0, POM1, POM5, POM7), port mode registers 0, 1, 3, 5, 7, X0 to X4 (PM0, PM1, PM3, PM5, PM7, PMX0 to PMX4), and port registers 0, 1, 3, 5, 7 (P0, P1, P3, P5, P7)).

2. After setting the SAUmEN bit to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

3. Be sure to clear the following bits to 0.
20-pin products: bits 1, 3, 4, 6 of PER0, and bits 3 to 7 of PERX
30, 32-pin products: bits 1, 6 of PER0, and bits 3 to 7 of PERX
48, 64-pin products: bits 1, 6 of PER0, and bits 3 to 7 of PERX

(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 SEm.n = 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 13-7. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1), F0566H, F0567H (SPSS) 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	Selection of operation clock (CKmk) Note 1					
					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
Other than above				Setting prohibited					

<R> **Note** Stop the operation of the serial array unit (SAU) (by setting bits 3 to 0 of ST0 register and bits 1 and 0 of ST1 and STS register to 1) before changing operation clock (f_{CLK}) selection (by changing the system clock control register (CKC) value).

Cautions 1. Be sure to clear bits 15 to 8 to "0".

2. After setting bit 2 (SAU0EN) and bit 3 (SAU1EN) of the PER0 register and bit 1 (SAUSEN) of PERX register to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency
f_{SUB}: Subsystem clock frequency

2. m: Unit number (m = 0, 1, S)

3. k = 0, 1

(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, UART, or I^2C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEm.n = 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 13-8. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), F0150H, F0151H (SMR10), F0152H, F0153H (SMR11), F0558H, F0559H (SMRS0), F055AH, F055BH (SMRS1) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn[Note]	0	SIS mn0[Note]	1	0	0	MD mn2	MD mn1	MD mn0

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 $\overline{SCKp}$ 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.	

STS mn[Note]	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^2C).
1	Valid edge of the RxDq pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Note The SMR01, SMR03, SMR11, and SMRS1 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 SMR00, SMR02, SMR10, or SMRS0 register) to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), q: UART number (q = 0 to 2, S0, S1), r: IIC number (r = 00, 01, 10, 11, 20, 21)

Figure 13-8. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), After reset: 0020H R/W
F0150H, F0151H (SMR10), F0152H, F0153H (SMR11),
F0558H, F0559H (SMRS0), F055AH, F055BH (SMRS1)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn[Note]	0	SIS mn0[Note]	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0[Note]	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MD mn2	MD mn1	Setting of operation mode of channel n
0	0	CSI mode
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note The SMR01, SMR03, SMR11, and SMRS1 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 SMR00, SMR02, SMR10, or SMRS0 register) to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), q: UART number (q = 0 to 2, S0, S1), r: IIC number (r = 00, 01, 10, 11, 20, 21)

(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 SEm.n = 1).

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

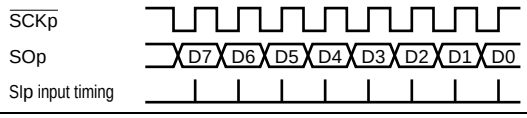
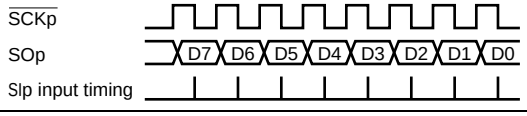
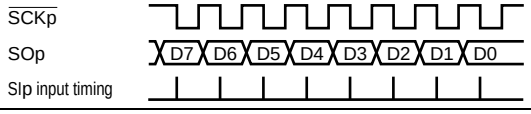
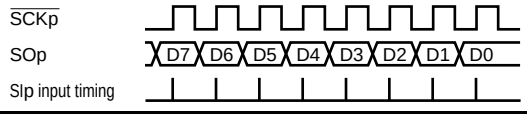
Reset signal generation sets the SCRmn register to 0087H.

Figure 13-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (mn = 00-03, 10, 11) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F0158H, F0159H (SCR10), F015AH, F015BH (SCR11) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1 Note 1	SLCmn0	0	1	DLSmn1 Note 2	DLSmn0

TXEmn	RXEmn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAPmn	CKPmn	Selection of data and clock phase in CSI mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

EOCmn	Selection of masking of error interrupt signal (INTSREx (x = 0 to 3))
0	Masks error interrupt INTSREx (INTSRx is not masked).
1	Enables generation of error interrupt INTSREx (INTSRx is masked if an error occurs).
Set EOCmn = 0 in the CSI mode, simplified I^2C mode, and during UART transmission Note 3.	

Notes 1. The SCR00, SCR02, and SCR10 registers only.

2. The SCR00 and SCR01 registers only. For other registers, the bit is fixed to 1.

3. When using CSI01 not with EOC01 = 0, error interrupt INTSRE0 may be generated.

Caution Be sure to clear bits 3, 6, and 11 to "0". (Also clear bit 5 of the SCR01, SCR03, or SCR11 register to 0, as well as bit 1 of the SCR02, SCR03, SCR10, or SCR11 register.). Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

Figure 13-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (mn = 00-03, 10, 11) (2/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F0158H, F0159H (SCR10), F015AH, F015BH (SCR11) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1 Note 1	SLCmn0	0	1	DLSmn1 Note 2	DLSmn0

PTCmn1	PTCmn0	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 3.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the CSI mode and simplified I^2C mode.			

DIRmn	Selection of data transfer sequence in CSI and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I^2C mode.	

SLCmn1 Note 1	SLCmn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (mn = 00, 02, 10 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode.		

DLSmn1 Note 2	DLSmn0	Setting of data length in CSI and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDRmn register (mn = 00, 01)) (settable in UART0 mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited
Be sure to set DLSmn0 = 1 in the simplified I^2C mode.		

Notes 1. The SCR00, SCR02, and SCR10 registers only.

2. The SCR00 and SCR01 registers only. For other registers, the bit is fixed to 1.

3. 0 is always added regardless of the data contents.

Caution Be sure to clear bits 3, 6, and 11 to "0". (Also clear bit 5 of the SCR01, SCR03, or SCR11 register to 0, and bit 1 of the SCR02, SCR03, SCR10, or SCR11 register to 1.). Be sure to set bit 2 to "1".

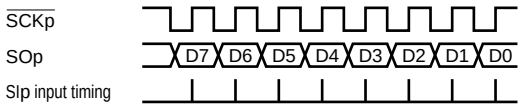
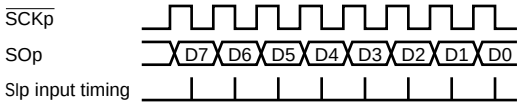
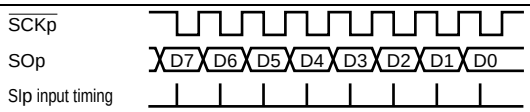
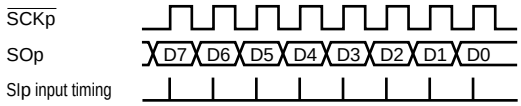
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 13-10. Format of Serial Communication Operation Setting Register mn (SCRmn) (mn = S0, S1) (1/2)

Address: F055CH, F055DH (SCRS0), F055EH, F055FH (SCRS1) 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	0	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	DLS mn3	DLS mn2	DLS mn1	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Does not start 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		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

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].	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the CSI mode and simplified I^2C mode.			

Note 0 is always added regardless of the data contents.

Caution Be sure to clear bits 6, 10, and 11 to "0".

Remark m: Unit number (m = S), n: Channel number (n = 0, 1), p: CSI number (p = S0, S1)

Figure 13-10. Format of Serial Communication Operation Setting Register mn (SCRmn) (mn = S0, S1) (2/2)

Address: F055CH, F055DH (SCRS0), F055EH, F055FH (SCRS1) 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	0	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	DLS mn3	DLS mn2	DLS mn1	DLS mn0

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	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
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.		

DLS mn3	DLS mn2	DLS mn1	DLS mn0	Setting of data length in CSI and UART modes	Serial-function correspondence		
					CSI	UART	IIC
0	1	1	0	7-bit data length (stored in bits 0 to 6 of SDRmn register)	√	√	–
0	1	1	1	8-bit data length (stored in bits 0 to 7 of SDRmn register)	√	√	–
1	0	0	0	9-bit data length (stored in bits 0 to 8 of SDRmn register)	√	√	–
1	0	0	1	10-bit data length (stored in bits 0 to 9 of SDRmn register)	√	–	–
1	0	1	0	11-bit data length (stored in bits 0 to 10 of SDRmn register)	√	–	–
1	0	1	1	12-bit data length (stored in bits 0 to 11 of SDRmn register)	√	–	–
1	1	0	0	13-bit data length (stored in bits 0 to 12 of SDRmn register)	√	–	–
1	1	0	1	14-bit data length (stored in bits 0 to 13 of SDRmn register)	√	–	–
1	1	1	0	15-bit data length (stored in bits 0 to 14 of SDRmn register)	√	–	–
1	1	1	1	16-bit data length (stored in bits 0 to 15 of SDRmn register)	√	√	–
Other than above				Setting prohibited			

Caution Be sure to clear bits 6, 10, and 11 to “0”.

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

(5) Higher 7 bits of the serial data register mn (SDRmn)

(a) Units 0 and 1

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 or bits 7 to 0 (lower 8 bits) of SDR02, SDR03, SDR10 and SDR11 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}).

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.

The lower 8/9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.

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

However, the higher 7 bits can be written or read only when the operation is stopped (SEm.n = 0). During operation (SEm.n = 1), a value is written only to the lower 8/9 bits of the SDRmn register. When the SDRmn register is read during operation, 0 is always read.

Reset signal generation clears the SDRmn register to 0000H.

Figure 13-11. Format of Serial Data Register mn (SDRmn) (mn = 00 to 03, 10, 11)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), After reset: 0000H R/W

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn																

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), After reset: 0000H R/W
FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)

	FFF45H (SDR02)								FFF44H (SDR02)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn																

SDRmn[15:9]							Transfer clock setting by dividing the operating clock (f_{MCK})
0	0	0	0	0	0	0	f_{MCK}/2, f_{SCK} (in CSI slave mode)
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

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

Cautions 1. Be sure to clear bit 8 of the SDR02, SDR03, SDR10, and SDR11 to "0".

2. Setting SDRmn[15:9] = (0000000B, 0000001B, 0000010B) is prohibited when UART is used.

3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.

4. Do not write eight bits to the lower eight bits if operation is stopped (SEm.n = 0). (If these bits are written to, the higher seven bits are cleared to 0.)

Remarks 1. For the function of the lower 8/9 bits of the SDRmn register, see **13.2 Configuration of Serial Array Unit**.

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

(b) Unit S

SDRmn is the transmit/receive data register (16 bits) of channel n. When operation is stopped (SEm.n = 0), bits 15 to 9 are used as the division setting register of the operating clock (f_{MCK}). During operation (SEm.n = 1), bits 15 to 9 are used as a transmission/reception buffer register.

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 SDRmn is used as the transfer clock.

See **13.2 Configuration of Serial Array Unit** for the functions of SDRmn during operation (SEm.n = 1).

SDRmn can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 13-12. Format of Serial Data Register mn (SDRmn) (mn = S0, S1)

Address: F0540H, F0541H (SDRS0), F0542H, F0543H (SDRS1) After reset: 0000H R/W

	F0541H (SDRS0)							F0540H (SDRS0)								
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0	0	0	0	0	0	0	0	0

SDRmn[15:9]							Setting of division ratio of operation clock (f_{MCK})
0	0	0	0	0	0	0	f_{MCK}
0	0	0	0	0	0	1	$f_{MCK}/2$
0	0	0	0	0	1	0	$f_{MCK}/3$
0	0	0	0	0	1	1	$f_{MCK}/4$
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	$f_{MCK}/127$
1	1	1	1	1	1	1	$f_{MCK}/128$

Cautions 1. When operation is stopped (SEm.n = 0), be sure to clear bits 8 to 0 to "0".

2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.

4. Do not write eight bits to the lower eight bits if operation is stopped (SEm.n = 0). (If these bits are written to, the higher seven bits are cleared to 0.)

Remarks 1. For the function of during operation (SEm.n = 1), see **13.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = S), n: Channel number (n = 0, 1)

(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 13-13. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03), F0148H, F0149H (SIR10), F014AH, F014BH (SIR11), F0554H, F0555H (SIRS0), F0556H, F0557H (SIRS1) 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	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Note The SIR01, SIR03, SIR11, and SIRS1 registers only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00, SIR02, SIR10, or SIRS0 register) to "0".

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

2. When the SIRmn register is read, 0000H is always read.

(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 13-14. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), F0140H, F0141H (SSR10), F0142H, F0143H (SSR11), F0550H, F0551H (SSRS0), F0552H, F0553H (SSRS1) 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	TSFmn	BFFmn	0	0	FEFmn[Note]	PEFmn	OVFmn

TSFmn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The STm.n bit of the STm register is set to 1 (communication is stopped) or the SSm.n bit of the SSm register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFFmn	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 STm.n bit of the STm register is set to 1 (communication is stopped) or the SSm.n bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Note The SSR01, SSR03, SSR11, and SSRS1 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.

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

Figure 13-14. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), After reset: 0000H R
F0140H, F0141H (SSR10), F0142H, F0143H (SSR11),
F0550H, F0551H (SSRS0), F0552H, F0553H (SSRS1)

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	TSFmn	BFFmn	0	0	FEFmn[Note]	PEFmn	OVFmn

FEFmn[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.	

PEFmn	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I^2C transmission).
<Clear condition> • 1 is written to the PECTmn bit of the SIRmn register. <Set condition> • The parity of the transmit data and the parity bit do not match when UART reception ends (parity error). • No ACK signal is returned from the slave channel at the ACK reception timing during I^2C transmission (ACK is not detected).	

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> • 1 is written to the OVCTmn bit of the SIRmn register. <Set condition> • Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • Transmit data is not ready for slave transmission or transmission and reception in CSI mode.	

Note The SSR01, SSR03, SSR11, and SSRS1 registers only.

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 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 (SEm.n) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSm.n bit is a trigger bit, it is cleared immediately when SEm.n = 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 13-15. 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	SS0.3	SS0.2	SS0.1	SS0.0

Address: F0162H, F0163H (SS1), F0562H, F0563H (SSS) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm (m = 1, S)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm.1	SSm.0

SSm.n	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEm.n bit to 1 and enters the communication wait status Note.

Note If the SSmn bit is set to 1 during communication, the communication stops and the communication wait state is entered. At this time, the values of the control registers and shift register and the status of the SCKmn and SOmn pins and the FEFmn, PEFmn, and OVFmn flags are held.

Caution Be sure to clear bits 15 to 4 of the SS0 register and bits 15 to 2 of the SS1, SSS registers to "0".

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

2. When the SSm register is read, 0000H is always read.

(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 (STm.n), the corresponding bit (SEm.n) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STm.n bit is a trigger bit, it is cleared immediately when SEm.n = 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 13-16. 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	ST0.3	ST0.2	ST0.1	ST0.0

Address: F0164H, F0165H (ST1), F0564H, F0565H (STS) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm (m = 1, S)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STm.1	STm.0

STm.n	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEm.n bit to 0 and stops the communication operation[Note].

Note Communication stops while holding the value of the control register and shift register, and the status of the serial clock I/O pin, serial data output pin, and each error flag (FEFmn: framing error flag, PEFmn: parity error flag, OVFmn: overrun error flag).

Caution Be sure to clear bits 15 to 4 of the ST0 register and bits 15 to 2 of the ST1, STS registers to "0".

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

2. When the STm register is read, 0000H is always read.

(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 13-17. 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	SE0.3	SE0.2	SE0.1	SE0.0

Address: F0160H, F0161H (SE1), F0560H, F0561H (SES) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm (m = 1, S)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SEm.1	SEm.0

SEm.n	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 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 SOm.n 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 SOm.n 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 13-18. Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH (SOE0) 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	SOE0.3	SOE0.2	SOE0.1	SOE0.0

Address: F016AH, F016BH (SOE1), F056AH, F056BH (SOES) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm (m = 1, S)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm.1	SOEm.0

SOEm.n	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 of the SOE0 register, and bits 15 to2 of the SOE1, SOES registers to "0".

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

(12) Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.

The value of the SOm.n 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 SOm.n bit of this register can be rewritten by software only when serial output is disabled (SOEm.n = 0). When serial output is enabled (SOEm.n = 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 (SEm.n = 0). While channel operation is enabled (SEm.n = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.

To use the serial interface pin as a port function pin, set the corresponding CKOmn and SOm.n 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 13-19. Format of Serial Output Register m (SOm)

Address: F0128H, F0129H (SO0) 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	CKO 01	CKO 00	0	0	0	0	SO 0.3	SO 0.2	SO 0.1	SO 0.0

Address: F0168H, F0169H (SO1), F0568H, F0569H (SOS) After reset: 0303H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm (m = 1, S)	0	0	0	0	0	0	CKO m1	CKO m0	0	0	0	0	0	0	SO m.1	SO m.0

CKO mn	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SO m.n	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution Be sure to set bits 11, 10, 3 and 2 of the SO1, SOS registers to “0”. And be sure to clear bits 15 to 12 and 7 to 4 of the SOm register to “0”.

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

This register can be set only in the UART mode. Be sure to set 0000H in the CSI mode and simplifies I^2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEm.n = 1). When serial output is disabled (SOEm.n = 0), the value of the SOm.n bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEm.n = 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 13-20. 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	SOL 0.2	0	SOL 0.0

Address: F0174H, F0175H (SOL1), F0570H, F0571H (SOLS) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm (m = 1, S)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL m.0

SOL m.n	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, and 1 of the SOL0 register and bits 15 to 1 of the SOL1, SOLS registers to "0".

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0, 2)

(14) Serial standby control register 0 (SSC0)

The SSC0 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data.

The SSC0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSC0 register can be set with an 8-bit memory manipulation instruction with SSC0L.

Reset signal generation clears the SSC0 register to 0000H.

<R> **Caution The maximum transfer rate in the SNOOZE mode is as follows.**

- **When using CSI00 : 1 Mbps**
- **When using UART0 : 4800 bps**

Figure 13-21. Format of Serial Standby Control Register 0 (SSC0)

Address: F0138H, F0139H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSC0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSEC0	SWC0

SSEC0	Selection of whether to enable or stop the generation of transfer end interrupts
0	Enable the generation of error interrupts (INTSRE0). In the following cases, the clock request signal (an internal signal) to the clock generator is also cleared: • When the SWC bit is cleared to 0 • When the UART reception start bit is mistakenly detected
1	Stop the generation of error interrupts (INTSRE0). In the following cases, the clock request signal (an internal signal) to the clock generator is also cleared: • When the SWC bit is cleared to 0 • When the UART reception start bit is mistakenly detected • When the transfer end interrupt generation timing is based on a parity error or framing error

SWC0	SNOOZE mode setting
0	SNOOZE mode function is not used.
1	SNOOZE mode function is used.
• STOP mode is cancelled by the hardware trigger signal generated during STOP mode, and reception operation of the CSI/UART is performed without the CPU operation (SNOOZE mode). • The SNOOZE mode function can be set only when high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). Setting SNOOZE mode function is prohibited when any other clock is selected. • Even when SNOOZE mode is used, the SWC bit must be set to 0 in normal operation mode. Change the bit to 1 immediately before a transition to STOP mode. After a return from STOP mode to normal operation mode, be sure to clear the SWC bit to 0.	

Caution Setting SSEC0, SWC0 = 1, 0 is prohibited.

(15) Input switch control register (ISC)

The ISC.1 and ISC.0 bits of the ISC register are used to realize a LIN-bus communication operation by UART2 in coordination with an external interrupt and the timer array unit.

When bit 0 is set to 1, the input signal of the serial data input (RxD2) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal.

When bit 1 is set to 1, the input signal of the serial data input (RxD2) pin is selected as a timer input, so that wake up signal can be detected, the low width of the sync break field, and the pulse width of the sync field can be measured by the timer.

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

Reset signal generation clears the ISC register to 00H.

Figure 13-22. Format of Input Switch Control Register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC.1	ISC.0

ISC.1	Switching channel 7 input of timer array unit
0	48, 64-pin products: Uses the input signal of the TI07 pin as a timer input (normal operation). 20, 30, and 32-pin products: Do not use a timer input signal for channel 7.
1	Input signal of the RxD2 pin is used as timer input (detects the wakeup signal and measures the low width of the sync break field and the pulse width of the sync field). Setting is prohibited in the 20-pin products.

ISC.0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD2 pin as an external interrupt (wakeup signal detection).

Caution Be sure to clear bits 7 to 2 to “0”.

(16) Noise filter enable register 0, X (NFEN0, NFENX)

The NFEN0 and NFENX registers are used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for CSI or simplified I^2C communication, by clearing the corresponding bit of these registers to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of these registers to 1.

When the noise filter is enabled, CPU/ peripheral hardware clock (f_{CLK}) is synchronized with 2-clock match detection.

The NFEN0 and NFENX registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 and NFENX registers to 00H.

Figure 13-23. Format of Noise Filter Enable Register 0, X (NFEN0, NFENX)

Address: F0070H (NFEN0), F050AH (NFENX) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	SNFEN20	0	SNFEN10	0	SNFEN00

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	0	0	UNFENS0

SNFEN20	Use of noise filter of RxD2 pin (RxD2/SDA20/SI20/P14)
0	Noise filter OFF
1	Noise filter ON
Set SNFEN20 to 1 to use the RxD2 pin. Clear SNFEN20 to 0 to use the other than RxD2 pin.	

SNFEN10	Use of noise filter of RxD1 pin (RxD1/ANI16/SI10/SDA10)
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.	

SNFEN00	Use of noise filter of RxD0 pin (RxD0/SI00/SIS0/RxDS0/TOOLRxD/SDA00/P11)
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear the SNFEN00 bit to 0 to use the other than RxD0 pin.	

UNFENS0	Use of noise filter of RxDS0 pin (RxD0/SI00/SIS0/RxDS0/TOOLRxD/SDA00/P11)
0	Noise filter OFF
1	Noise filter ON
Set the UNFENS0 bit to 1 to use the RxDS0 pin. Clear the UNFENS0 bit to 0 to use the other than RxDS0 pin.	

Caution Be sure to clear bits 7 to 5, 3, and 1 of NFEN0 register, and bits 7 to 1 of NFENX register to "0".

(17) Port input mode registers 0, 1, 5 (PIM0, PIM1, PIM5)

These registers set the input buffer of ports 0, 1, and 5 in 1-bit units.

The PIM0, PIM1, and PIM5 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PIM0, PIM1, and PIM5 registers to 00H.

Figure 13-24. Format of Port Input Mode Registers 0, 1, and 5 (PIM0, PIM1, PIM5) (64-pin products)

Address F0040H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM0	0	0	0	PIM0.4	PIM0.3	0	PIM0.1	0

Address F0041H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM1	PIM1.7	PIM1.6	PIM1.5	PIM1.4	PIM1.3	0	0	0

Address F0045H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM5	0	0	PIM5.5	0	0	0	0	0

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

(18) Port output mode registers 0, 1, 5, 7 (POM0, POM1, POM5, POM7)

These registers set the output mode of ports 0, 1, 5, and 7 in 1-bit units.

The POM0, POM1, POM5, and POM7 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the POM0, POM1, POM5, and POM7 registers to 00H.

Figure 13-25. Format of Port Output Mode Registers 0, 1, 5, and 7 (POM0, POM1, POM5, POM7) (64-pin products)

Address F0050H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM0	0	0	0	POM0.4	POM0.3	POM0.2	0	POM0.0

Address F0051H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM1	POM1.7	0	POM1.5	POM1.4	POM1.3	POM1.2	POM1.1	POM1.0

Address F0055H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM5	0	0	POM5.5	0	0	0	0	POM5.0

Address F0057H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM7	0	0	0	POM7.4	0	0	POM7.1	0

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

(19) Port mode registers 0, 1, 3, 5, 7, X0 to X4 (PM0, PM1, PM3, PM5, PM7, PMX0 to PMX4)

These registers set input/output of ports 0, 1, 3, 5, and 7 in 1-bit units.

When using the ports (such as P02/ANI17/SO10/TXD1) to be shared with the serial data output pin or serial clock output pin for serial data output or serial clock output, set the port mode register (PMxx) bit corresponding to each port to 0. And set the port register (Pxx) bit corresponding to each port to 1.

When using $\overline{\text{SCKS0}}$, SOS0/TXDS0 for serial data output or serial clock output, set the port mode register (PMxx) bit corresponding to each port to 1. And set the Port Mode Registers X0 and X1 (PMX0, PMX1) bit corresponding to each port to 0.

Example: When using P02/ANI17/SO10/TXD1 for serial data output or serial clock output
Set the PM0.2 bit of the port mode register 0 to 0.
Set the P0.2 bit of the port register 0 to 1.

When using the ports (such as P04/$\overline{\text{SCK10}}$/SCL10, P50/INTP1/SI11/SDA11/LRxD) to be shared with the serial data input pin or serial clock input pin for serial data input or serial clock input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

When using $\overline{\text{SCKS0}}$, SIS0/RxDS0 for serial data input or serial clock input, setting the PMX0, PMX1 registers is not necessary.

Example: When using P50/INTP1/SI11/SDA11/LRxD for serial data input
Set the PM5.0 bit of port mode register 5 to 1.
Set the P5.0 bit of port register 5 to 0 or 1.

The PM0, PM1, PM3, PM5, PM7, PMX0 to PMX4 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets the PM0, PM1, PM3, PM5, PM7, PMX0 to PMX4 registers to FFH.

Figure 13-26. Format of Port Mode Registers 0, 1, 3, 5 , and 7 (PM0, PM1, PM3, PM5, and PM7) (64-pin products)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	PM0.6	PM0.5	PM0.4	PM0.3	PM0.2	PM0.1	PM0.0

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM1.7	PM1.6	PM1.5	PM1.4	PM1.3	PM1.2	PM1.1	PM1.0

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM3.1	PM3.0

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	1	1	PM5.5	PM5.4	PM5.3	PM5.2	PM5.1	PM5.0

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM7.7	PM7.6	PM7.5	PM7.4	PM7.3	PM7.2	PM7.1	PM7.0

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 5, 7; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 13-27. Format of Port Mode Registers X0 to X4 (PMX0 to PMX4) (64-pin products)

Address: F0504H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
PMX0	0	0	0	0	0	0	0	PMX0

PMX0	Select the alternate function of P10/SCK00/$\overline{\text{SCKS0}}$/SCL00 pin
0	$\overline{\text{SCKS0}}$ output (master mode)
1	$\overline{\text{SCKS0}}$ input, or other alternate function (including general-purpose I/O port)

Address: F0505H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
PMX1	0	0	0	0	0	0	0	PMX1

PMX1	Select the alternate function of P12/SO00/TxD0/SOS0/TxDS0/TOOLTxD pin
0	SOS0 or TxDS0 output
1	Other alternate function (including general-purpose I/O port)

Address: F0506H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
PMX2	0	0	0	0	0	0	0	PMX2

PMX2	Select the alternate function of P51/INTP2/SO11/LTxD pin
0	LTxD output
1	Other alternate function (including general-purpose I/O port)

Address: F0507H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
PMX3	0	0	0	0	0	0	0	PMX3

PMX3	Select the alternate function of P55/$\overline{\text{SCKS1}}$ pin
0	$\overline{\text{SCKS1}}$ output (master mode)
1	$\overline{\text{SCKS1}}$ input (slave mode), or other alternate function (including general-purpose I/O port)

Address: F0508H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
PMX4	0	0	0	0	0	0	0	PMX4

PMX4	Select the alternate function of P53/SOS1 pin
0	SOS1 output
1	Other alternate function (including general-purpose I/O port)

13.4 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 following pins can be used as port function pins in this mode.

20-pin:

P10/SCK00/SCKS0/SCL00, P11/SI00/RxD0/SIS0/RxDS0/TOOLRxD/SDA00, P12/SO00/TxD0/SOS0/TxDS0/TOOLTxD

30, 32-pin:

P00/ANI17/TI00/TxD1, P01/ANI16/TO00/RxD1, P10/SCK00/SCKS0/SCL00, P11/SI00/RxD0/SIS0/RxDS0/TOOLRxD/SDA00, P12/SO00/TxD0/SOS0/TxDS0/TOOLTxD, P13/TxD2/SO20, P14/RxD2/SI20/SDA20, P15/PCLBUZ1/SCK20/SCL20, P30/INTP3/SCK11/SCL11, P50/INTP1/SI11/SDA11/LRxD0, P51/INTP2/SO11/LTxD0

48-pin:

P00/TI00/TxD1, P01/TO00/RxD1, P10/SCK00/SCKS0/SCL00, P11/SI00/RxD0/SIS0/RxDS0/TOOLRxD/SDA00, P12/SO00/TxD0/SOS0/TxDS0/TOOLTxD, P13/TxD2/SO20, P14/RxD2/SI20/SDA20, P15/PCLBUZ1/SCK20/SCL20, P30/INTP3/SCK11/SCL11, P50/INTP1/SI11/ SDA11/LRxD0, P51/INTP2/SO11/LTxD0, P70/KR0/SCK21/SCL21, P71/KR1/SI21/SDA21, P72/KR2/SO21, P73/KR3/SO01, P74/KR4/INTP8/SI01/SDA01, P75/KR5/INTP9/SCK01/SCL01

64-pin:

P02/ANI17/SO10/TXD1, P03/ANI16/SI10/RXD1/SDA10, P04/SCK10/SCL10, P10/SCK00/SCKS0/SCL00, P11/SI00/RxD0/SIS0/RxDS0/TOOLRxD/SDA00, P12/SO00/TxD0/SOS0/TxDS0/TOOLTxD, P14/RXD2/SI20/SDA20, P15/SCK20/SCL20, P30/INTP3/RTC1HZ/SCK11/SCL11, P50/INTP1/SI11/LRxD0, P51/INTP2/SO11/LTxD0, P53/SOS1, P54/SIS1, P55/SCKS1, P70/KR0/SCK21/SCL21, P71/KR1/SI21/SDA21, P72/KR2/SO21, P73/SO01, P74/INTP8/SI01/SDA01, P75/INTP9/SCK01/SCL01

13.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable registers 0 and X (PER0, PERX).

The PER0 and PERX registers are used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware that is not used is stopped in order to reduce the power consumption and noise.

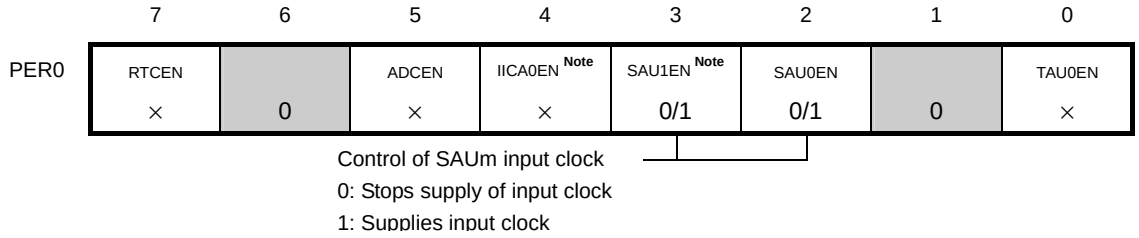
To stop the operation of serial array unit 0, set bit 2 (SAU0EN) of PER0 to 0.

To stop the operation of serial array unit 1, set bit 3 (SAU1EN) of PER0 to 0.

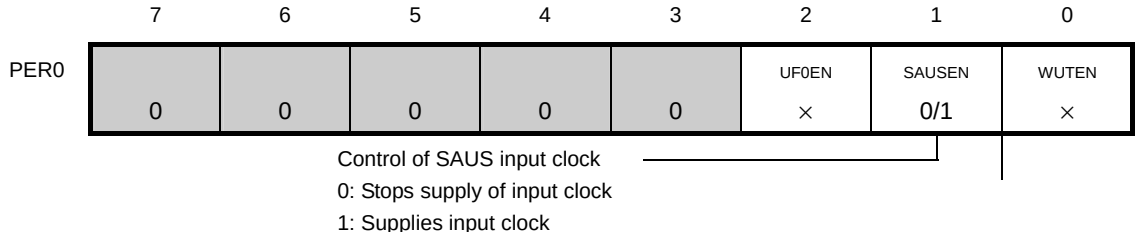
To stop the operation of serial array unit S, set bit 1 (SAUSEN) of PERX to 0.

Figure 13-28. Peripheral Enable Register 0, X (PER0, PERX) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.



(b) Peripheral enable register X (PERX) … Set only the bit of SAUS to be stopped to 0.



Note The bit is not provided in the 20-pin products..

Cautions 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.
Note that this does not apply to the following registers.
- **Input switch control register (ISC)**
- **Noise filter enable registers 0, X (NFEN0, NFENX)**
- **Port input mode registers 0, 1, 5 (PIM0, PIM1, PIM5)**
- **Port output mode registers 0, 1, 5, 7 (POM0, POM1, POM5, POM7)**
- **Port mode registers 0, 1, 3, 5, 7, X0 to X4 (PM0, PM1, PM3, PM5, PM7, PMX0 to PMX4)**
- **Port registers 0, 1, 3, 5, 7 (P0, P1, P3, P5, P7)**

2. Be sure to clear the following bits to 0.
The 20-pin products: bits 1, 3, 4, 6 of PER0 register, bits 3 to 7 of PERX register
The 30, 32-pin products: bits 1, 6 of PER0 register, bits 3 to 7 of PERX register
The 48, 64-pin products: bits 1, 6 of PER0 register, bits 3 to 7 of PERX register

Remark ☐: Setting disabled (set to the initial value)
×: 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

13.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 13-29. 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.

STm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	STm.3 Note 0/1	STm.2 Note 0/1	STm.1 0/1	STm.0 0/1

1: Clears the SEm.n bit to 0 and stops the communication operation

* Because the STm.n bit is a trigger bit, it is cleared immediately when SEm.n = 0.

(b) Serial Channel Enable Status Register m (SEm) … This register indicates whether serial transmission/reception operation of each channel is enabled or stopped.

SEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SEm.3 Note 0/1	SEm.2 Note 0/1	SEm.1 0/1	SEm.0 0/1

0: Operation stops

* 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.

SOEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SOEm.3 Note 0/1	SOEm.2 Note 0/1	SOEm.1 0/1	SOEm.0 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SOm.n 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.

SOm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	CKOm3 Note 0/1	CKOm2 Note 0/1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	SOm.3 Note 0/1	SOm.2 Note 0/1	SOm.1 0/1	SOm.0 0/1

1: Serial clock output value is "1" 1: Serial data output value is "1"

* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOm.n bits to "1".

Note Serial array unit 0 only.

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

2. ☐ : Setting disabled (set to the initial value**), 0/1: Set to 0 or 1 depending on the usage of the user**

13.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSIS0, CSIS1) Communication

This is a clocked communication function that uses three lines: serial clock ($\overline{SCK}$) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)
- Data length of 7 to 16 bits (CSIS0, CSIS1)
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[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

  During master communication (CSI00): Max. $f_{CLK}/2$[Note]

  During master communication (other than CSI00): Max. $f_{CLK}//4$[Note]

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 (channel 0 of unit 0) supports the SNOOZE mode. When $\overline{SCK00}$ pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. SNOOZE mode can be specified only for CSI00.

Note Use the clocks within a range satisfying the $\overline{SCK}$ cycle time (t_{KCY}) characteristics (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE) and CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

The channels supporting 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSIS0, CSIS1) are channels 1 and 3 of SAU0, channels 0 and 1 of SAU1, and channel 0 of SAUS.

- 20-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	–		–
	2	–	–	–
	3	–		–
1	0	–	–	–
	1	–		–
S	0	CSIS0	UARTS0	–
	1	–		–

- 30, 32, and 36-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	–		–
	2	–	UART1	–
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	–		–
S	0	CSIS0	UARTS0	–
	1	–		–

- 48-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	–	UART1	–
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
S	0	CSIS0	UARTS0	–
	1	–		–

- 64-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
S	0	CSIS0	UARTS0	–
	1	CSIS1		–

3-wire serial I/O (CSI00, CSI01, CIS10, CIS11, CIS20, CIS21, CSIS0, CSIS1) performs the following six types of communication operations.

- Master transmission (See **13.5.1**.)
- Master reception (See **13.5.2**.)
- Master transmission/reception (See **13.5.3**.)
- Slave transmission (See **13.5.4**.)
- Slave reception (See **13.5.5**.)
- Slave transmission/reception (See **13.5.6**.)
- SNOOZE mode function (CSI00 only) (See **13.5.7**.)

13.5.1 Master transmission

Master transmission is an operation wherein the RL78/F12 outputs a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSIS0	CSIS1
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 0 of SAUS	Channel 1 of SAUS
Pins used	$\overline{SCK00}$, SO00	$\overline{SCK01}$, SO01	$\overline{SCK10}$, SO10	$\overline{SCK11}$, SO11	$\overline{SCK20}$, SO20	$\overline{SCK21}$, SO21	$\overline{SCKS0}$, SOS0	$\overline{SCKS1}$, SOS1
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSIS0	INTCSIS1
	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						7 to 16 bits	
Transfer rate	Max. $f_{CLK}/2$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] **Note**	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] **Note**						
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the serial clock operation. • 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: Not reversed (Data output at the falling edge of SCK, data input at the rising edge of SCK) • CKPmn = 1: Reversed (Data output at the rising edge of SCK, data input at the falling edge of SCK)							
Data direction	MSB or LSB first							

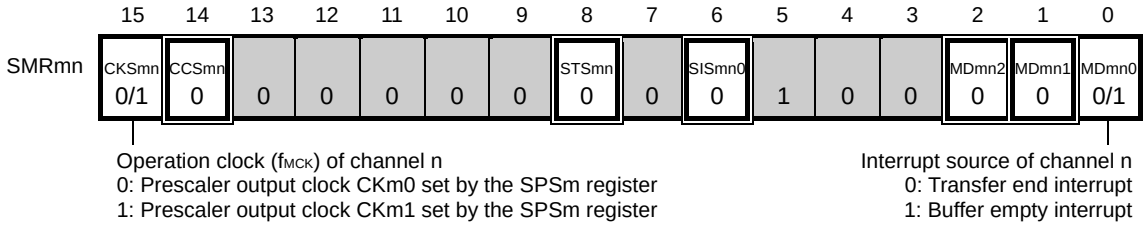
Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1
f_{CLK}: System clock frequency

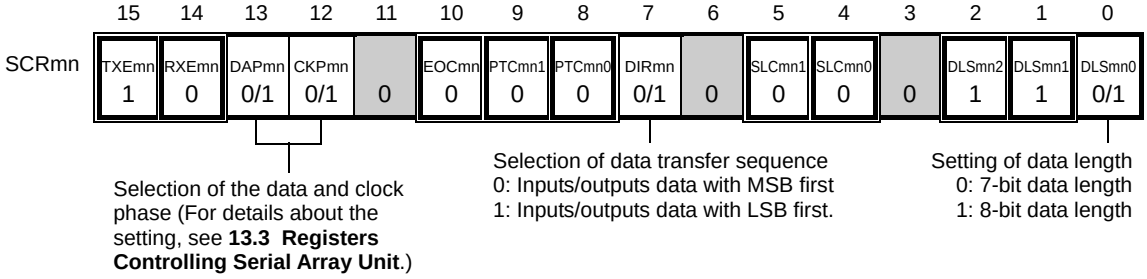
(1) Register setting

Figure 13-30. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (1/2)

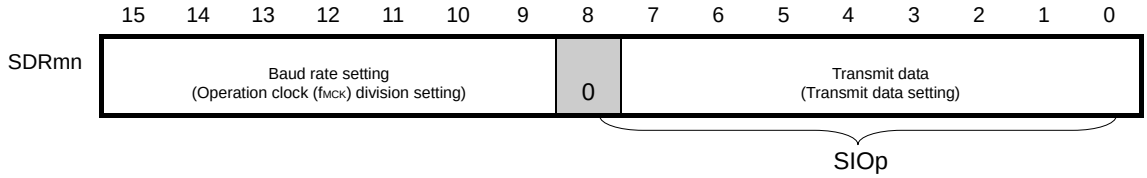
(a) Serial mode register mn (SMRmn)



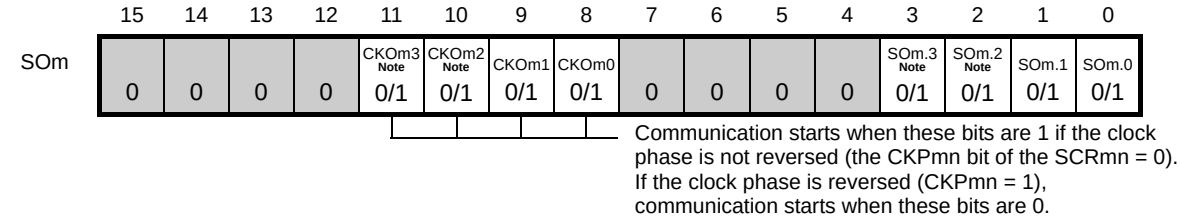
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Sets only the bits of the target channel.



Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

2. ☐: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-30. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (2/2)

(e) 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	SOEm.3 Note 0/1	SOEm.2 Note 0/1	SOEm.1 0/1	SOEm.0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm.3 Note 0/1	SSm.2 Note 0/1	SSm.1 0/1	SSm.0 0/1

Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

2. ☐: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-31. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSISn) (1/2)

(a) Serial output register m (SOS) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOS	0	0	0	0	0	0	CKOS1 0/1	CKOS0 0/1	0	0	0	0	0	0	SOS.1 0/1	SOS.0 0/1

Communication starts when these bits are 1 if the data phase is not reversed (CKPSn = 0). If the phase is reversed (CKPSn = 1), communication starts when these bits are 0.

(b) Serial output enable register S (SOES) … 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
SOES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOES.1 0/1	SOES.0 0/1

(c) Serial channel start register S (SSS) … 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
SSS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSS.1 0/1	SSS.0 0/1

(d) Serial mode register Sn (SMRSn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRSn	CKSSn 0/1	CCSSn 0	0	0	0	0	0	STSSn 0	0	SISAn 0	1	0	0	MDSn2 0	MDSn1 0	MDSn0 0/1

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register Sn (SCRSn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRSn	TXESn 1	RXESn 0	DAPSn 0/1	CKPSn 0/1	0	0	PTCSn1 0	PTCSn0 0	DIRSn 0/1	0	SLCSn1 0	SLCSn0 0	DLSSn3 0/1	DLSSn2 0/1	DLSSn1 0/1	DLSSn0 0/1

(f) Serial data register Sn (SDRSn)

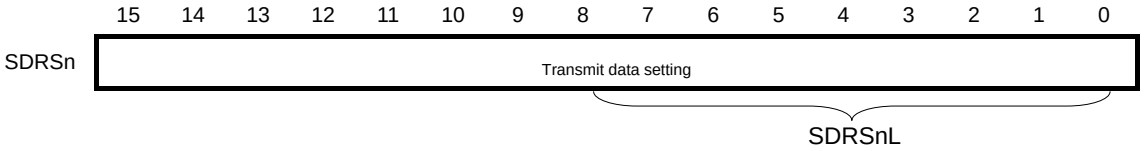
(i) When operation is stopped (SES.n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRSn	Baud rate setting							0	0	0	0	0	0	0	0	0

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
n : Channel number (n = 0, 1)

Figure 13-31. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSISn) (2/2)

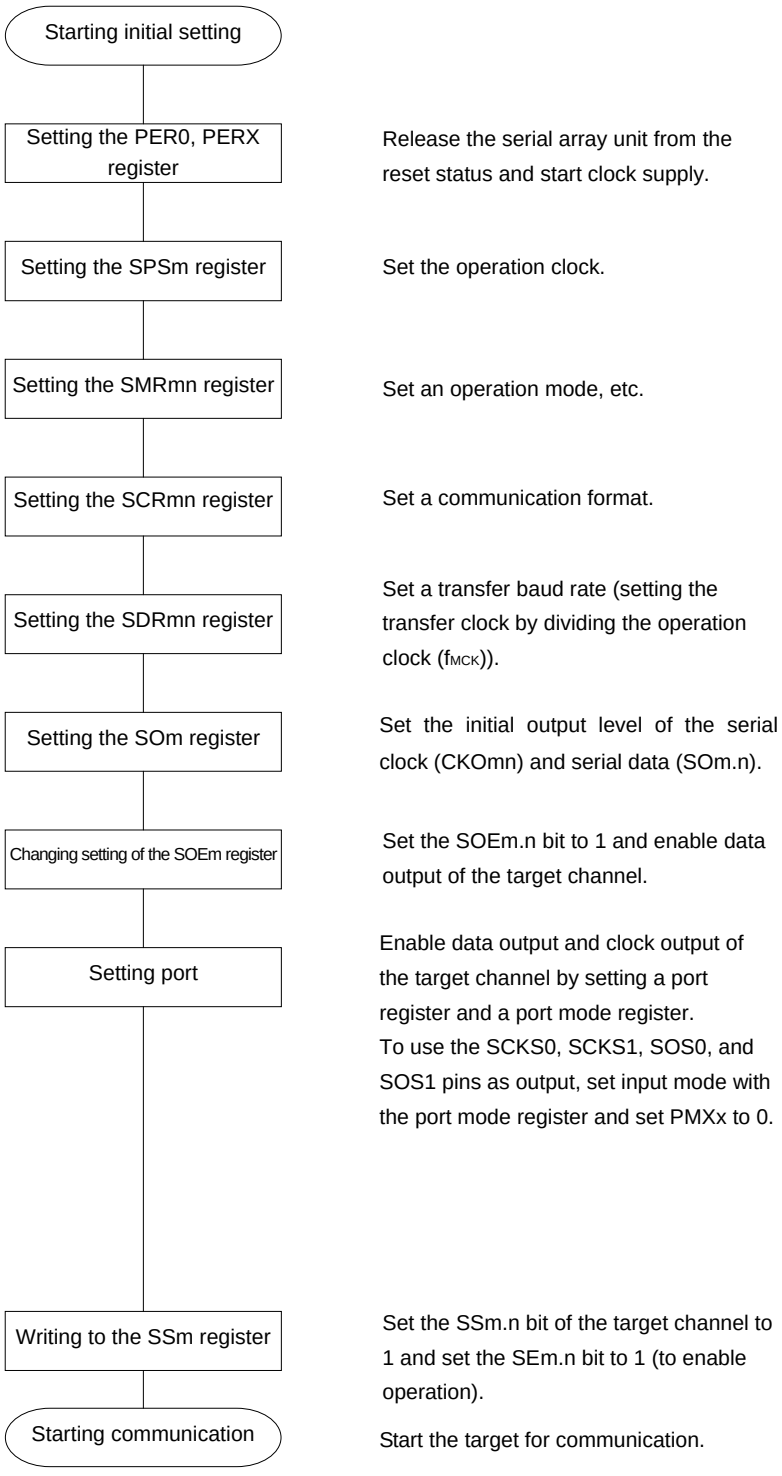
(ii) During operation (SES.n = 1) (lower 8 bits: SDRSnL)



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
n : Channel number (n = 0, 1)

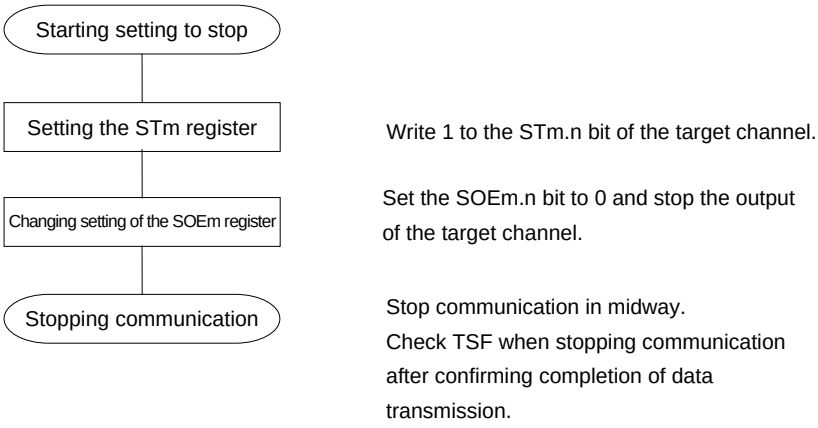
(2) Operation procedure

Figure 13-32. Initial Setting Procedure for Master Transmission



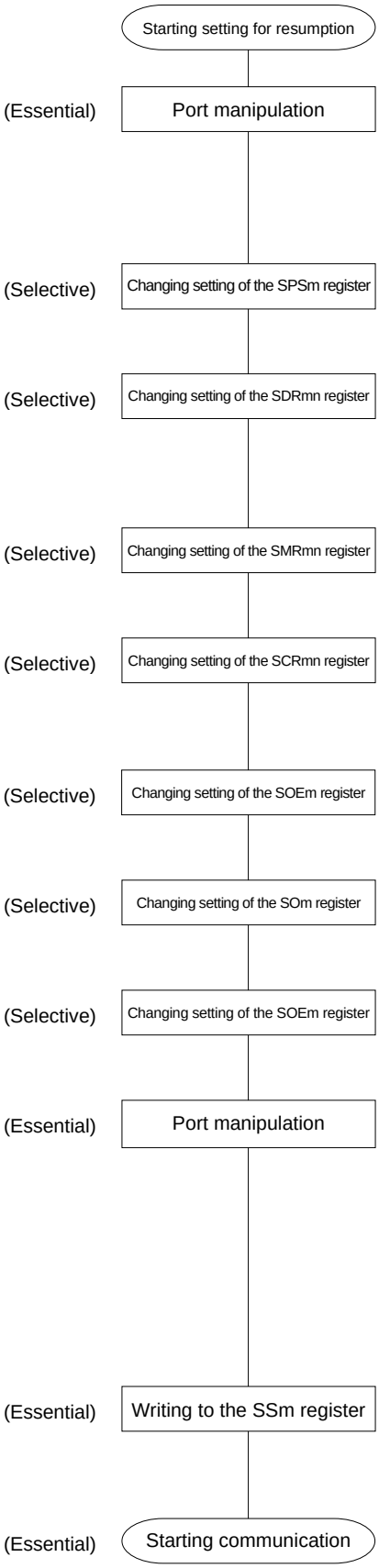
Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 13-33. Procedure for Stopping Master Transmission



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 13-34 Procedure for Resuming Master Transmission**).

Figure 13-34. Procedure for Resuming Master Transmission



Enable data output and clock output of the target channel by setting a port register and a port mode register.
To use the SCKS0, SCKS1, SOS0, and SOS1 pins as output, set input mode with the port mode register and set PMXx to 0.

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.

Set the SOEm.n bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOm.n).

Set the SOEm.n 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.
To use the P10 pin as the $\overline{\text{SCKS0}}$ output, set PMX0 to “0”.
To use the P10 pin as the $\overline{\text{SCKS0}}$ input or $\overline{\text{SCK00}}$/SCL00, set PMX0 to “1”.

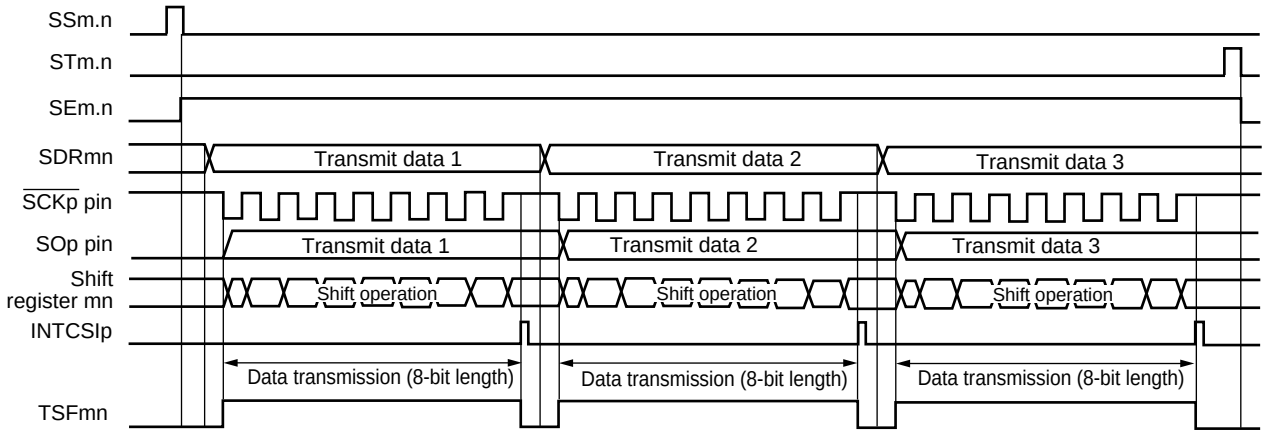
To use the P12 pin as the SOS0/TxDS0 output, set PMX1 to “0”. To use the P12 pin as the SO00/TxD0 output, set PMX1 to “1”.

Set the SSm.n bit of the target channel to 1 and set the SEm.n bit to 1 (to enable operation).

Sets transmit data to the SIOp or SDRmn register and start communication.

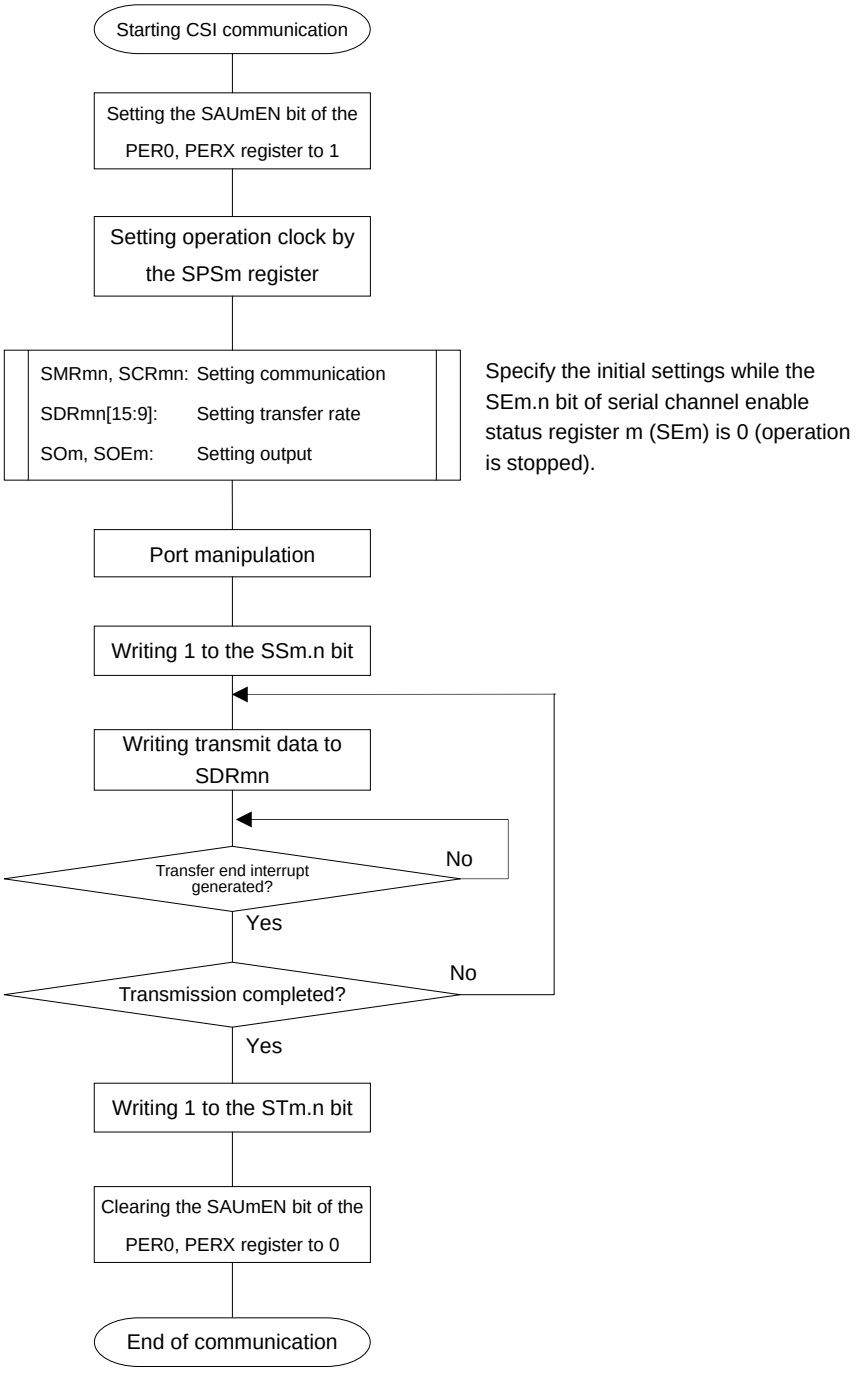
(3) Processing flow (in single-transmission mode)

Figure 13-35. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

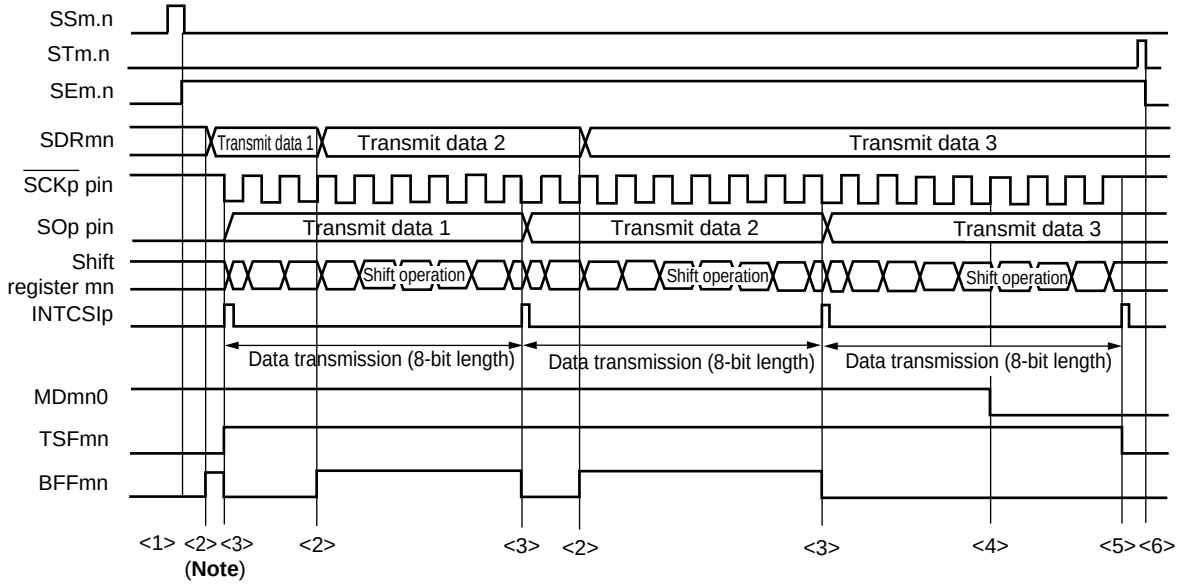
Figure 13-36. Flowchart of Master Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 13-37. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

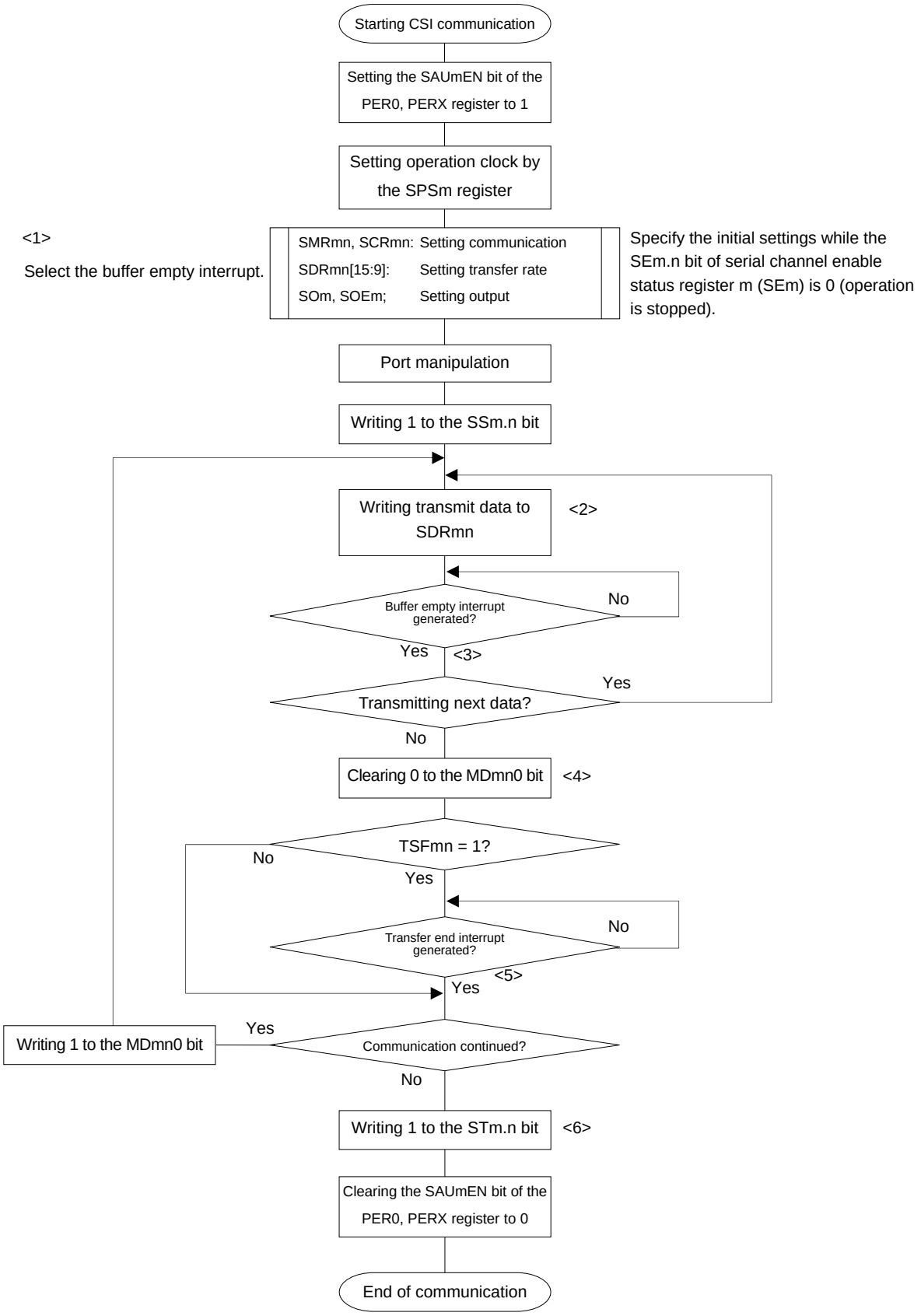


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

Figure 13-38. Flowchart of Master Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 13-37 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

13.5.2 Master reception

Master reception is an operation wherein the RL78/F12 outputs a transfer clock and receives data from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSIS0	CSIS1
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 0 of SAUS	Channel 1 of SAUS
Pins used	$\overline{\text{SCK00}}$, SI00	$\overline{\text{SCK01}}$, SI01	$\overline{\text{SCK10}}$, SI10	$\overline{\text{SCK11}}$, SI11	$\overline{\text{SCK20}}$, SI20	$\overline{\text{SCK21}}$, SI21	$\overline{\text{SCKS0}}$, SIS0	$\overline{\text{SCKS1}}$, SIS1
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSIS0	INTCSIS1
	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						7 to 16 bits	
Transfer rate	Max. $f_{CLK}/2$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]**Note**	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]**Note**						
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the serial clock operation. • 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: Not reversed • CKPmn = 1: Reversed							
Data direction	MSB or LSB first							

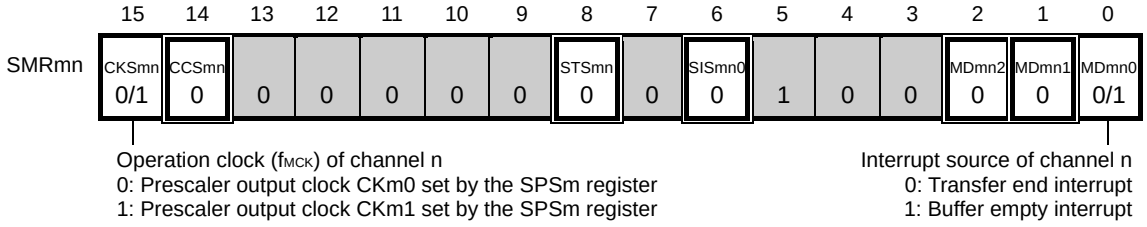
Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1
f_{CLK}: System clock frequency

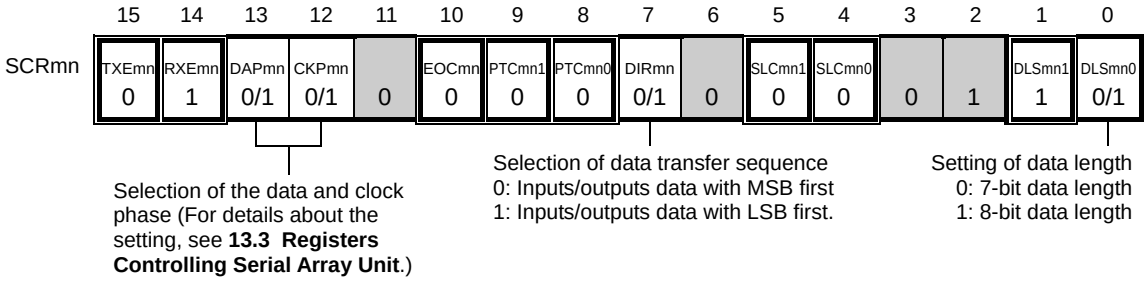
(1) Register setting

Figure 13-39. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (1/2)

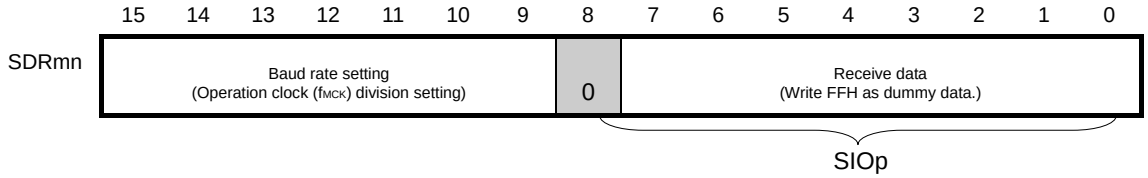
(a) Serial mode register mn (SMRmn)



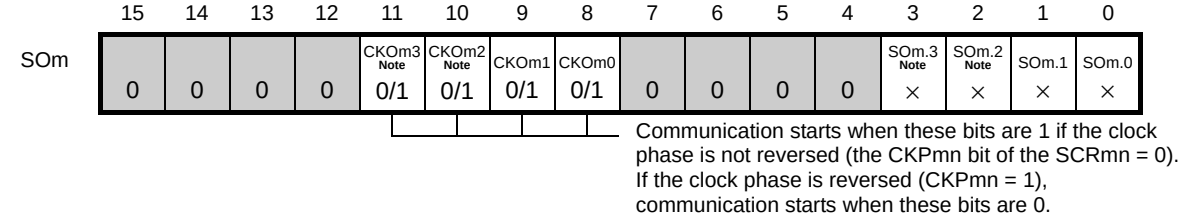
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Sets only the bits of the target channel.



Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00, to 03, 10, 11

2. ☐: Setting is fixed in the CSI master reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-39. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI11, CSI20, CSI21) (2/2)

(e) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm.3 Note ×	SOEm.2 Note ×	SOEm.1 ×	SOEm.0 ×

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm.3 Note 0/1	SSm.2 Note 0/1	SSm.1 0/1	SSm.0 0/1

Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

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

Figure 13-40. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSISn) (1/2)

(a) Serial output register S (SOS) ... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOS	0	0	0	0	0	0	CKOS1 0/1	CKOS0 0/1	0	0	0	0	0	0	SOS.1 ×	SOS.0 ×

Communication starts when these bits are 1 if the data phase is not reversed (CKPSn = 0). If the phase is reversed (CKPSn = 1), communication starts when these bits are 0.

(b) Serial output enable register S (SOES) ...The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOES.1 ×	SOES.0 ×

(c) Serial channel start register S (SSS) ... 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
SSS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSS.1 0/1	SSS.0 0/1

(d) Serial mode register Sn (SMRSn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRSn	CKSSn 0/1	CCSSn 0	0	0	0	0	0	STSSn 0	0	SISSn0 0	1	0	0	MDSn2 0	MDSn1 0	MDSn0 0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register Sn (SCRSn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRSn	TXESn 0	RXESn 1	DAPSn 0/1	CKPSn 0/1	0	0	PTCSn1 0	PTCSn0 0	DIRSn 0/1	0	SLCSn1 0	SLCSn0 0	DLSSn3 0/1	DLSSn2 0/1	DLSSn1 0/1	DLSSn0 0/1

(f) Serial data register Sn (SDRSn)

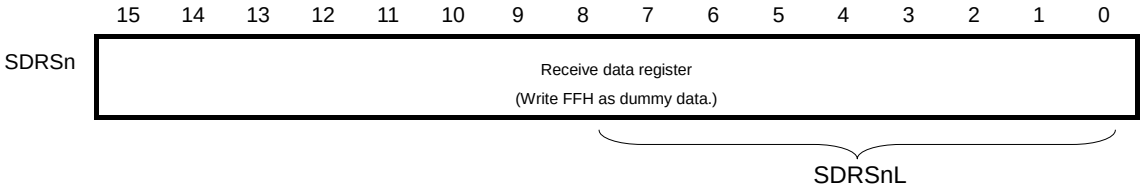
(i) When operation is stopped (SES.n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRSn	Baud rate setting							0	0	0	0	0	0	0	0	0

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
n: Channel number (n = 0, 1)

Figure 13-40. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSISn) (2/2)

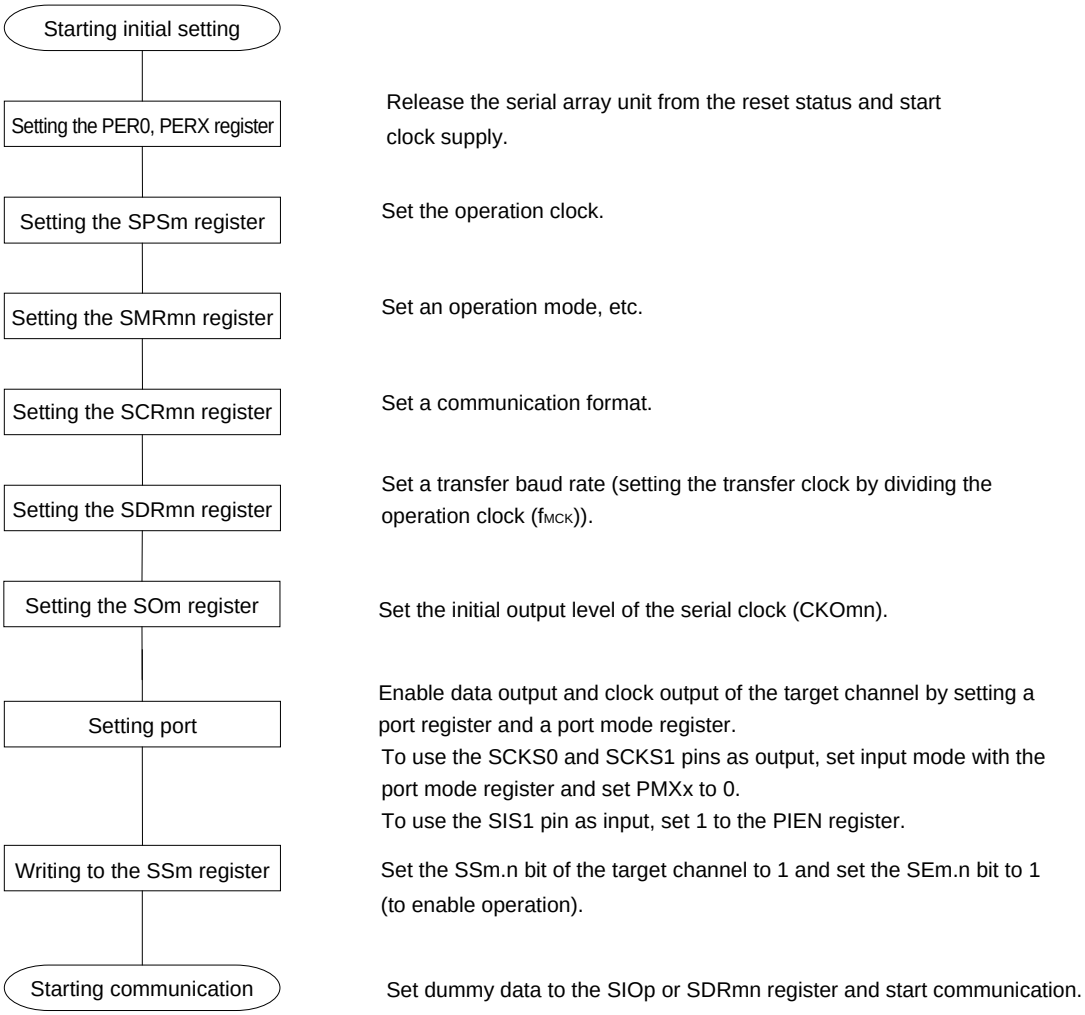
(ii) During operation (SES.n = 1) (lower 8 bits: SDRSnL)



Remark ☐: Setting is fixed in the CSI master reception mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user
n: Channel number (n = 0, 1)

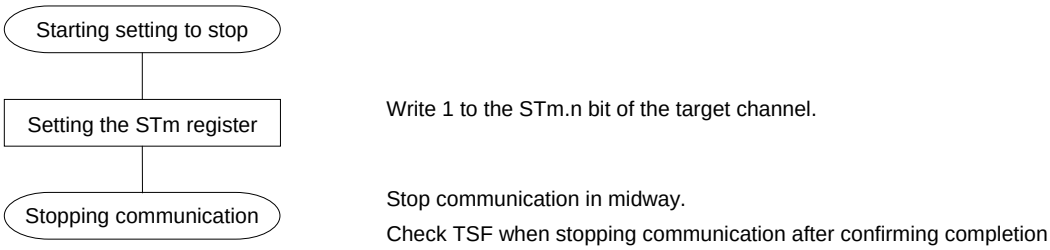
(2) Operation procedure

Figure 13-41. Initial Setting Procedure for Master Reception



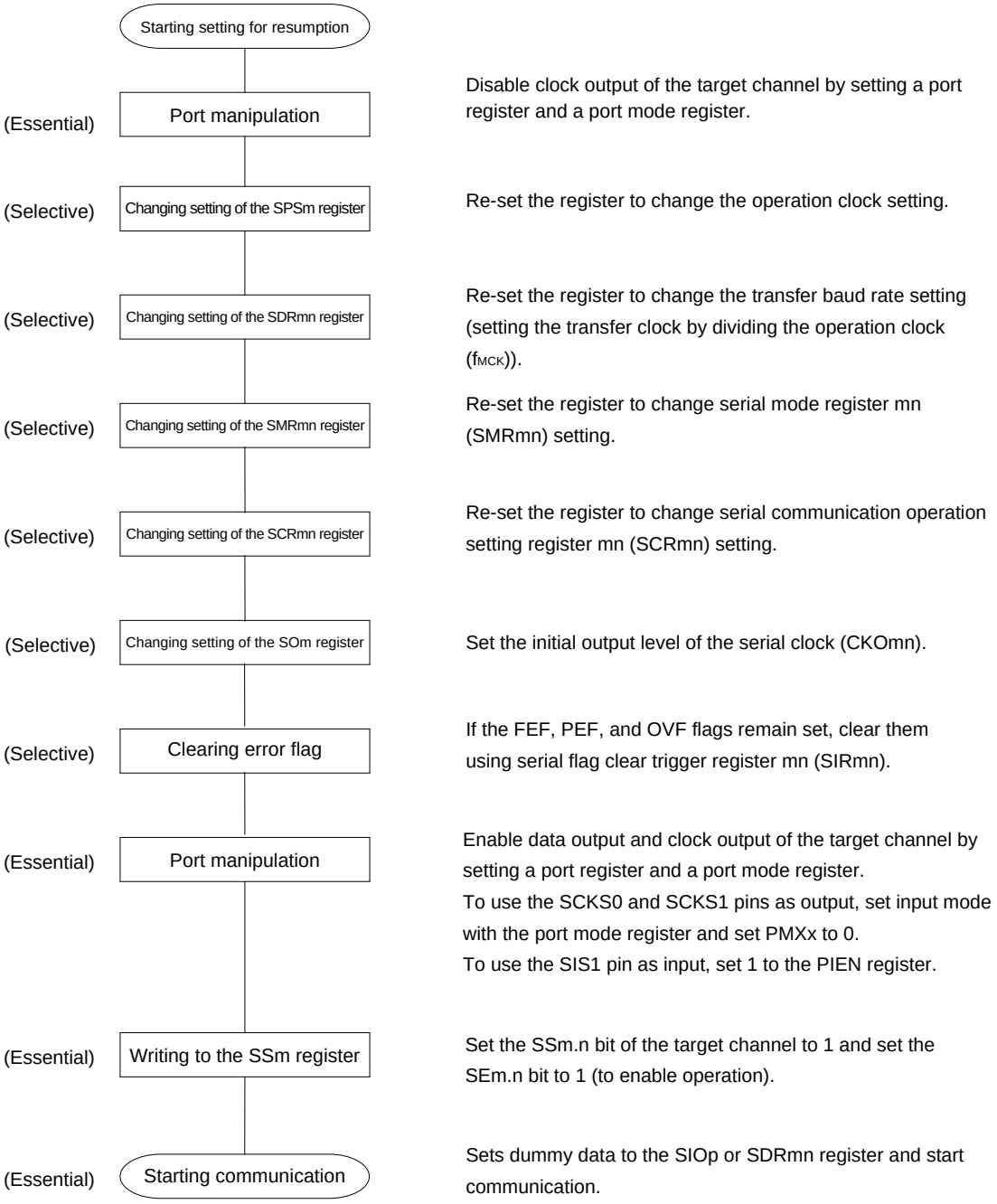
Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 13-42. Procedure for Stopping Master Reception



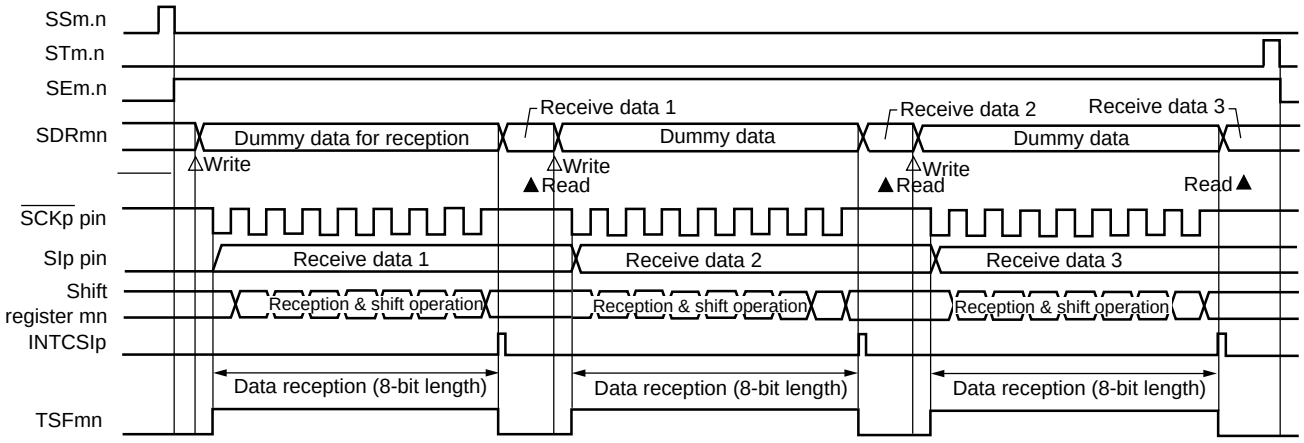
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 13-43 Procedure for Resuming Master Reception**).

Figure 13-43. Procedure for Resuming Master Reception



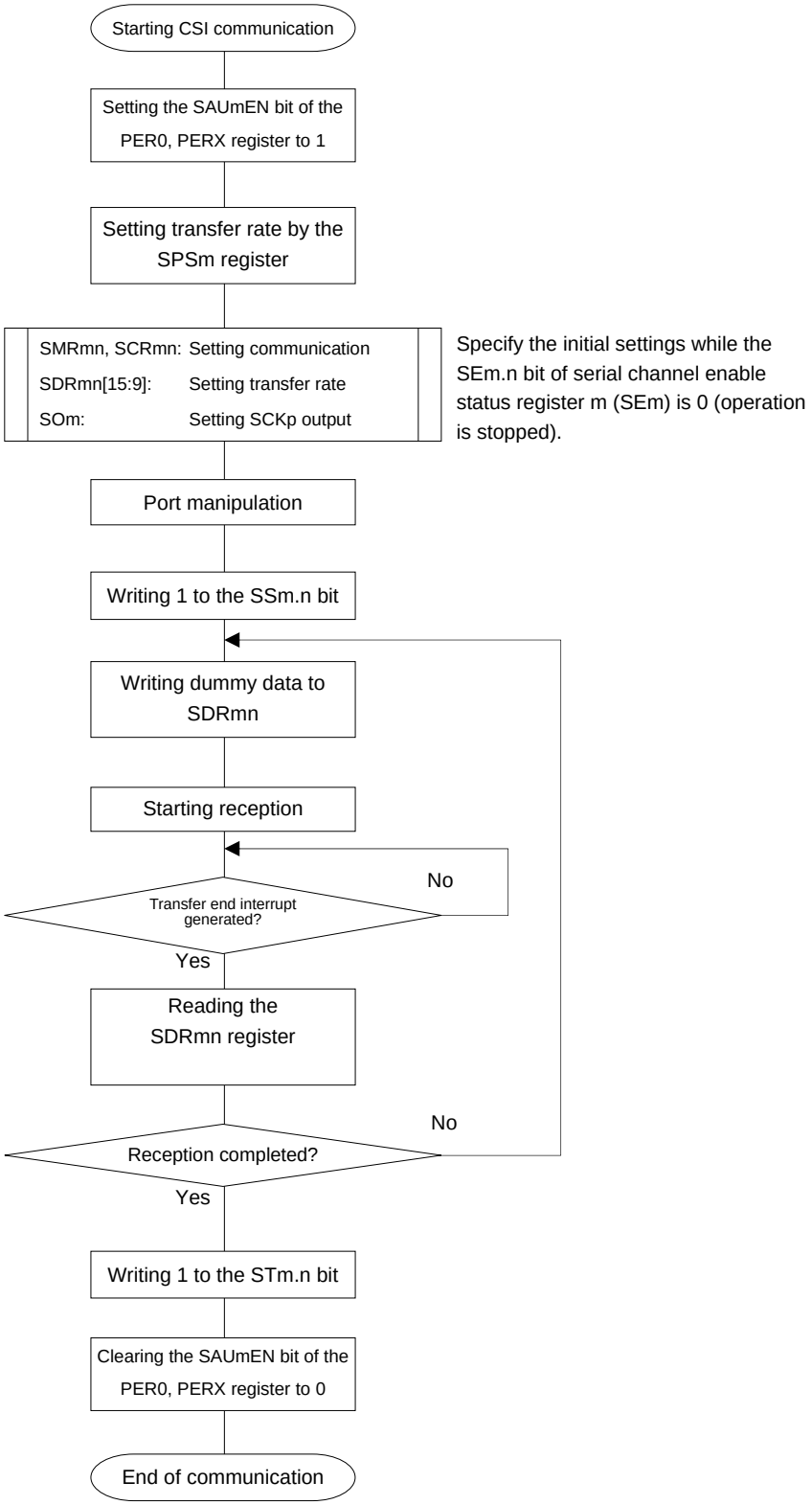
(3) Processing flow (in single-reception mode)

Figure 13-44. Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

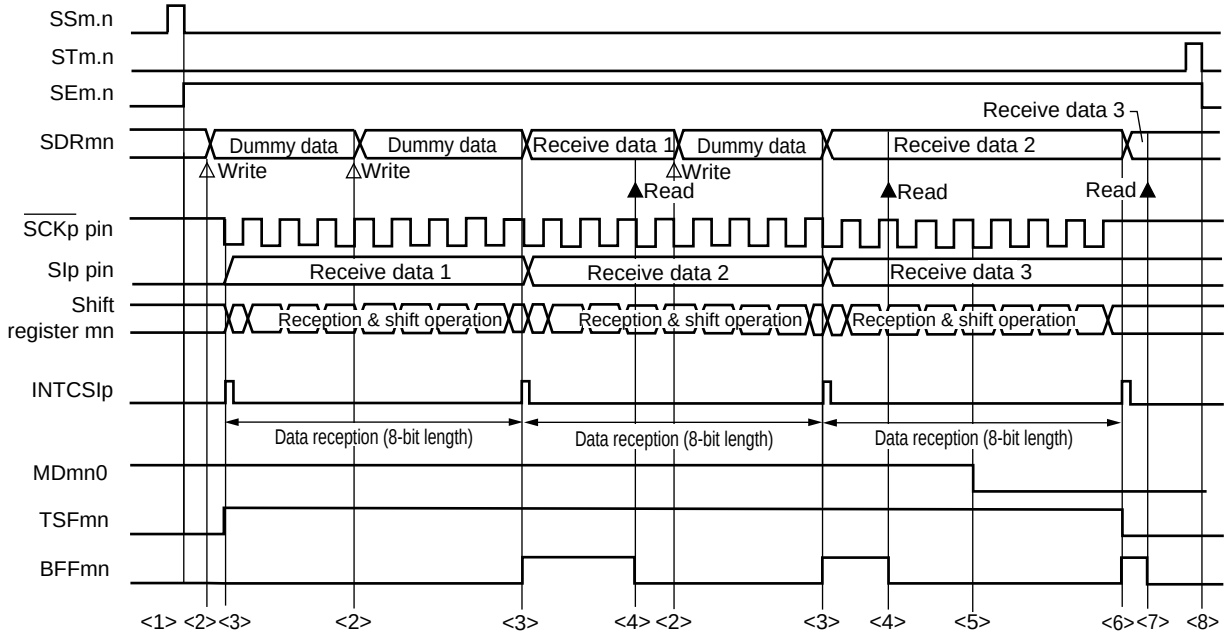
Figure 13-45. Flowchart of Master Reception (in Single-Reception Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous reception mode)

Figure 13-46. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

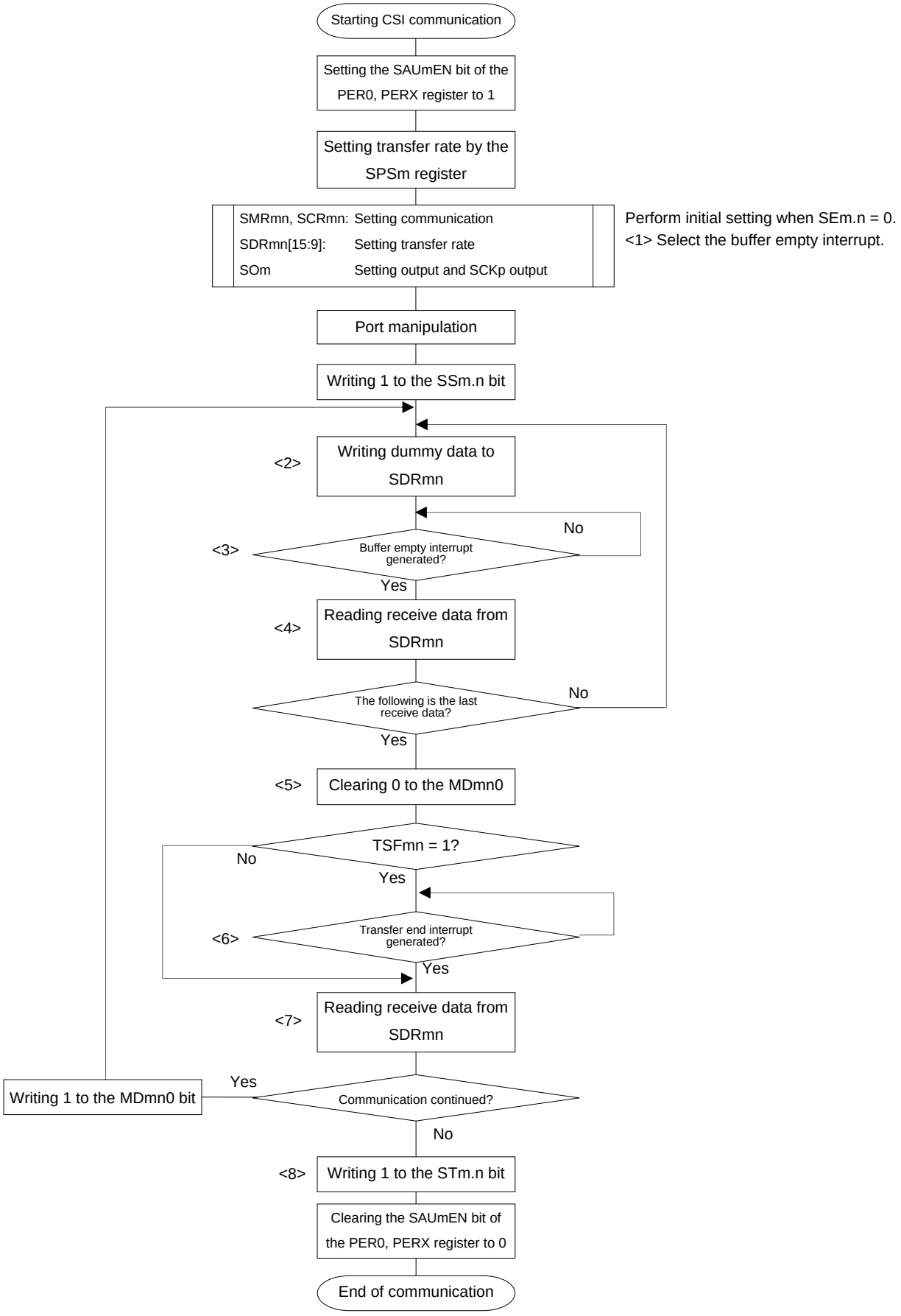


Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 13-47 Flowchart of Master Reception (in Continuous Reception Mode)**.

2. m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

Figure 13-47. Flowchart of Master Reception (in Continuous Reception Mode)



Caution **After setting the PER0, PERX register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.**

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 13-46 Timing Chart of Master Reception (in Continuous Reception Mode)**.

13.5.3 Master transmission/reception

Master transmission/reception is an operation wherein the RL78/F12 outputs a transfer clock and transmits/receives data to/from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSIS0	CSIS1
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 0 of SAUS	Channel 1 of SAUS
Pins used	$\overline{\text{SCK00}}$, SI00, SO00	$\overline{\text{SCK01}}$, SI01, SO01	$\overline{\text{SCK10}}$, SI10, SO10	$\overline{\text{SCK11}}$, SI11, SO11	$\overline{\text{SCK20}}$, SI20, SO20	$\overline{\text{SCK21}}$, SI21, SO21	$\overline{\text{SCKS0}}$, SIS0, SOS0	$\overline{\text{SCKS1}}$, SIS1, SOS1
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSIS0	INTCSIS1
	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						7 to 16 bits	
Transfer rate	Max. $f_{CLK}/2$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]**Note**	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]**Note**						
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the serial clock operation. • 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: Not reversed • CKPmn = 1: Reversed							
Data direction	MSB or LSB first							

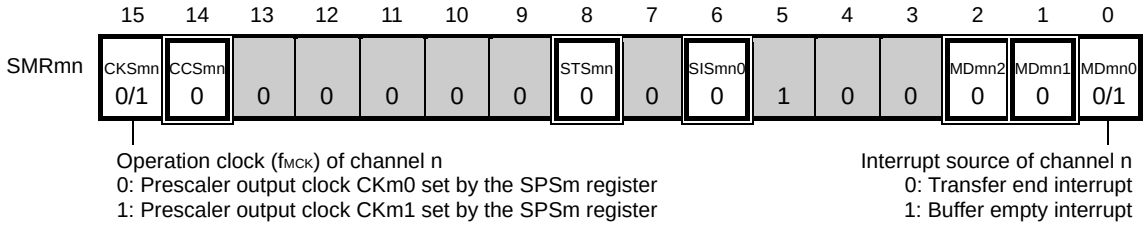
Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1
f_{CLK}: System clock frequency

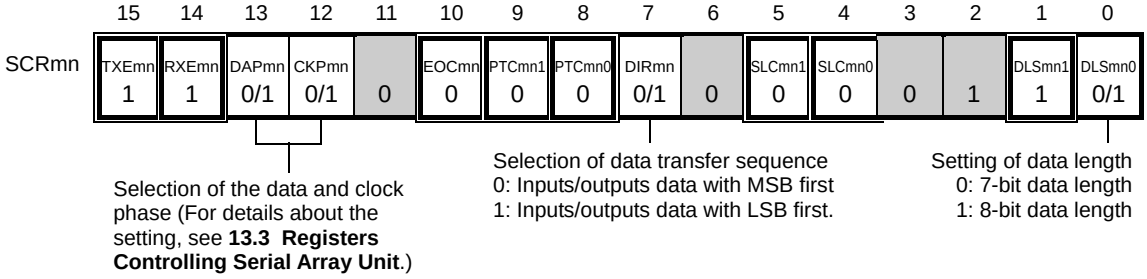
(1) Register setting

Figure 13-48. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (1/2)

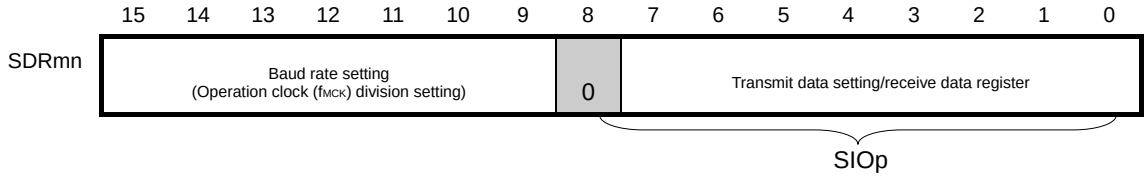
(a) Serial mode register mn (SMRmn)



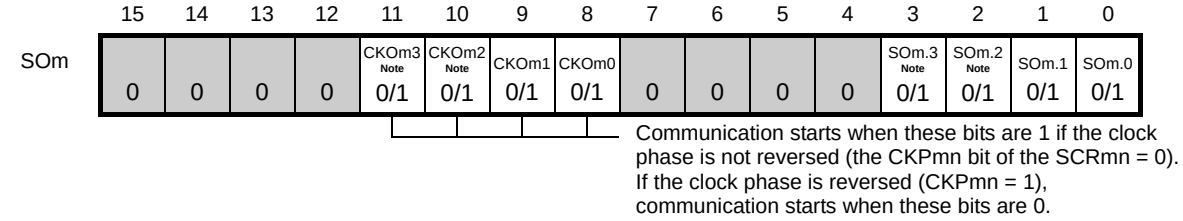
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Sets only the bits of the target channel.



Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

2. ☐: Setting is fixed in the CSI master transmission/reception mode
▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-48. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (2/2)

(e) 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	SOEm.3 Note 0/1	SOEm.2 Note 0/1	SOEm.1 0/1	SOEm.0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm.3 Note 0/1	SSm.2 Note 0/1	SSm.1 0/1	SSm.0 0/1

Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

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

Figure 13-49. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSISn) (1/2)

(a) Serial output register m (SOS) … Sets only the bits of the target channel.

SOS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	CKOS1 0/1	CKOS0 0/1	0	0	0	0	0	0	SOS.1 0/1	SOS.0 0/1

Communication starts when these bits are 1 if the data phase is not reversed (CKPSn = 0). If the phase is reversed (CKPSn = 1), communication starts when these bits are 0.

(b) Serial output enable register S (SOES) … Sets only the bits of the target channel to 1.

SOES

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOES1 0/1	SOES0 0/1

(c) Serial channel start register S (SSS) … Sets only the bits of the target channel to 1.

SSS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSS.1 0/1	SSS.0 0/1

(d) Serial mode register Sn (SMRSn)

SMRSn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKSSn 0/1	CCSSn 0	0	0	0	0	0	STSSn 0	0	SISSn0 0	1	0	0	MDSn2 0	MDSn1 0	MDSn0 0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register Sn (SCRSn)

SCRSn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXESn 1	RXESn 1	DAPSn 0/1	CKPSn 0/1	0	0	PTCSn1 0	PTCSn0 0	DIRSn 0/1	0	SLCSn1 0	SLCSn0 0	DLSSn3 0/1	DLSSn2 0/1	DLSSn1 0/1	DLSSn0 0/1

(f) Serial data register S0 (SDRS0)

(i) When operation is stopped (SES.n = 0)

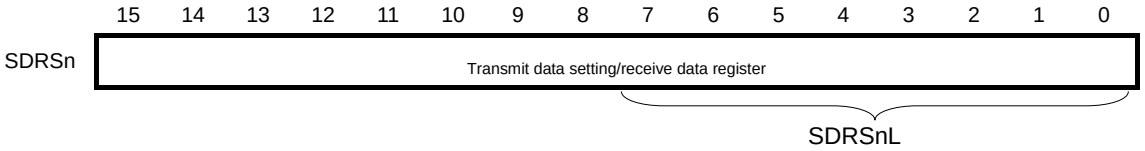
SDRSn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Baud rate setting							0	0	0	0	0	0	0	0	0

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
n: Channel number (n = 0, 1)

Figure 13-49. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSISn) (2/2)

(ii) During operation (SES.n = 1) (lower 8 bits: SDRSnL)



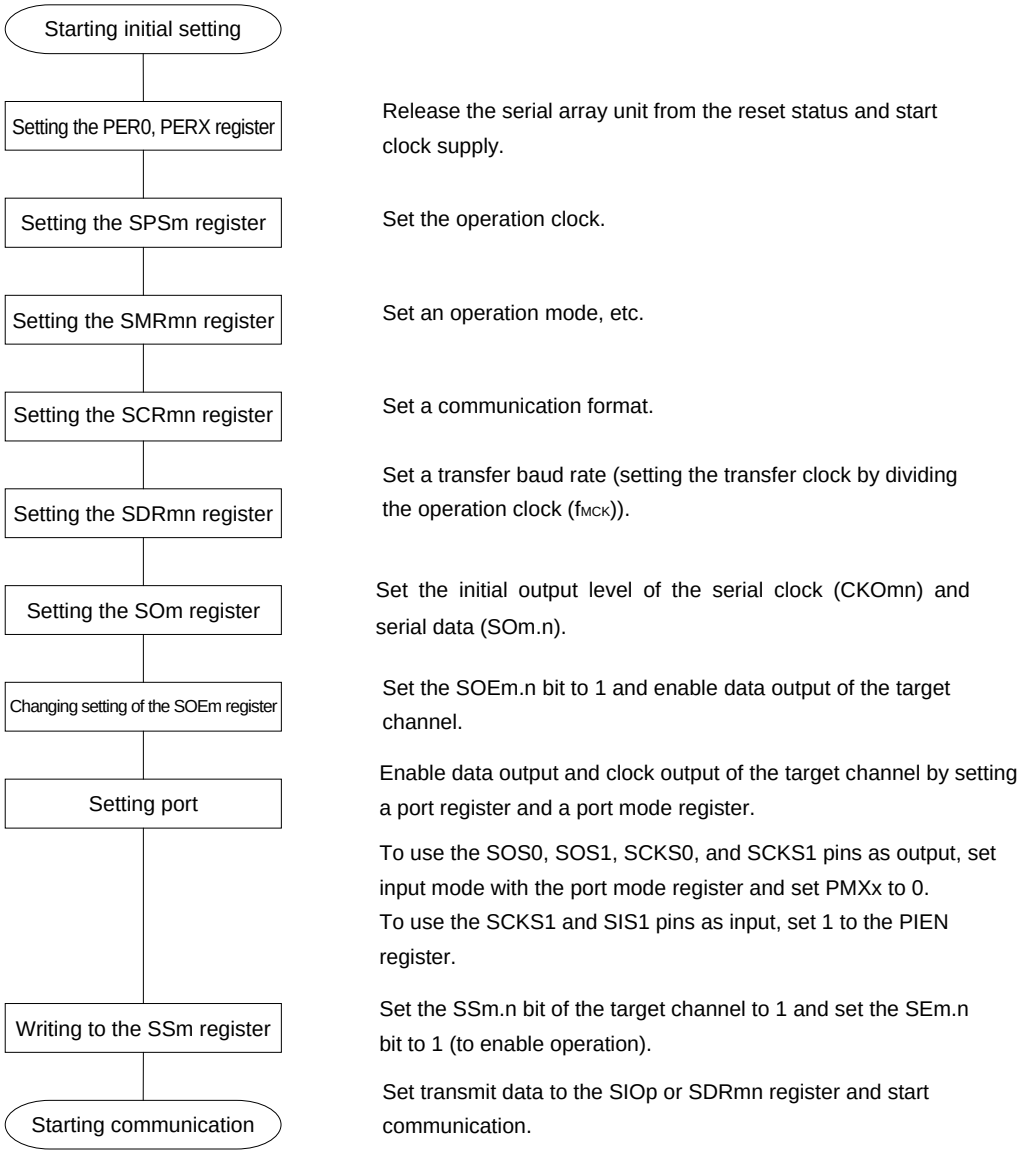
Remark ☐: Setting is fixed in the CSI master transmission/reception mode, ▒ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

n: Channel number (n = 0, 1)

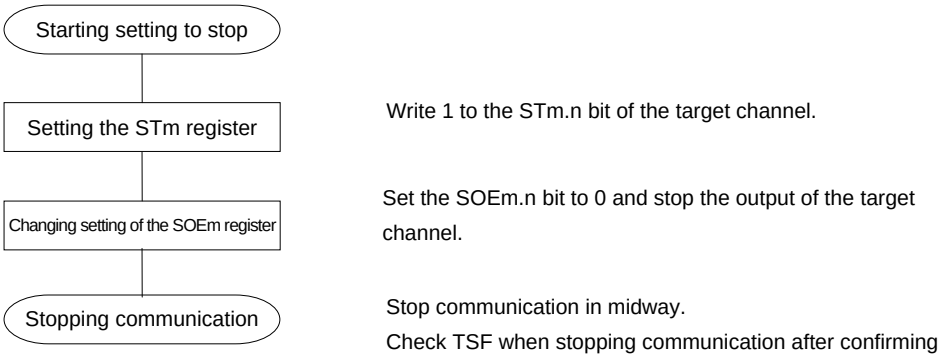
(2) Operation procedure

Figure 13-50. Initial Setting Procedure for Master Transmission/Reception



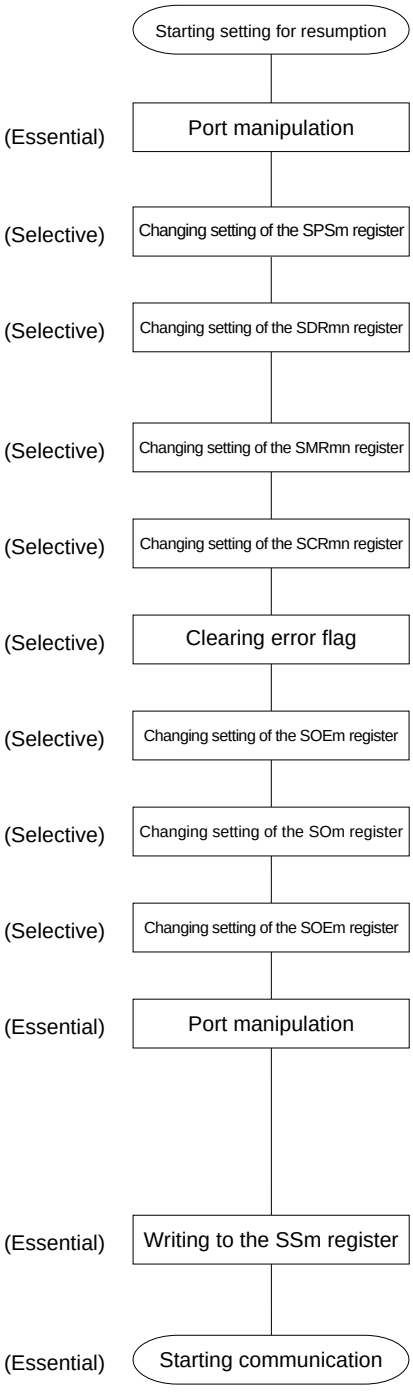
Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 13-51. Procedure for Stopping Master Transmission/Reception



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 13-52 Procedure for Resuming Master Transmission/ Reception**).

Figure 13-52. Procedure for Resuming Master Transmission/Reception



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 FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register mn (SIRmn).

Set the SOEm.n bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOm.n).

Set the SOEm.n bit to 1 and enable output from the target channel.

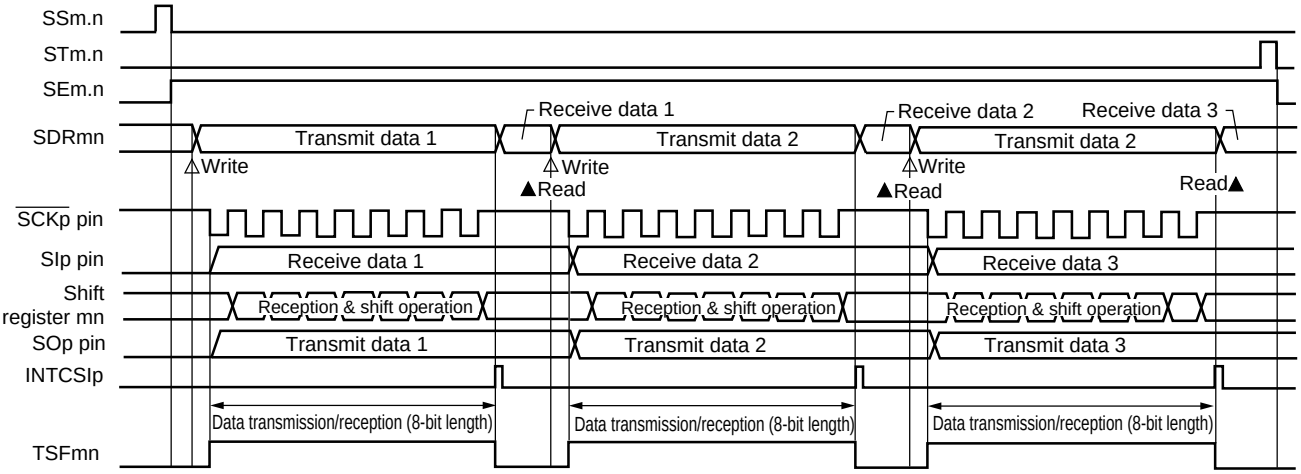
Enable data output and clock output of the target channel by setting a port register and a port mode register.
To use the SOS0, SOS1, SCKS0, and SCKS1 pins as output, set input mode with the port mode register and set PMXx to 0.
To use the SCKS1 and SIS1 pins as input, set 1 to the PIEN register.

Set the SSm.n bit of the target channel to 1 and set the SEm.n bit to 1 (to enable operation).

Sets transmit data to the SIOp or SDRmn register and start communication.

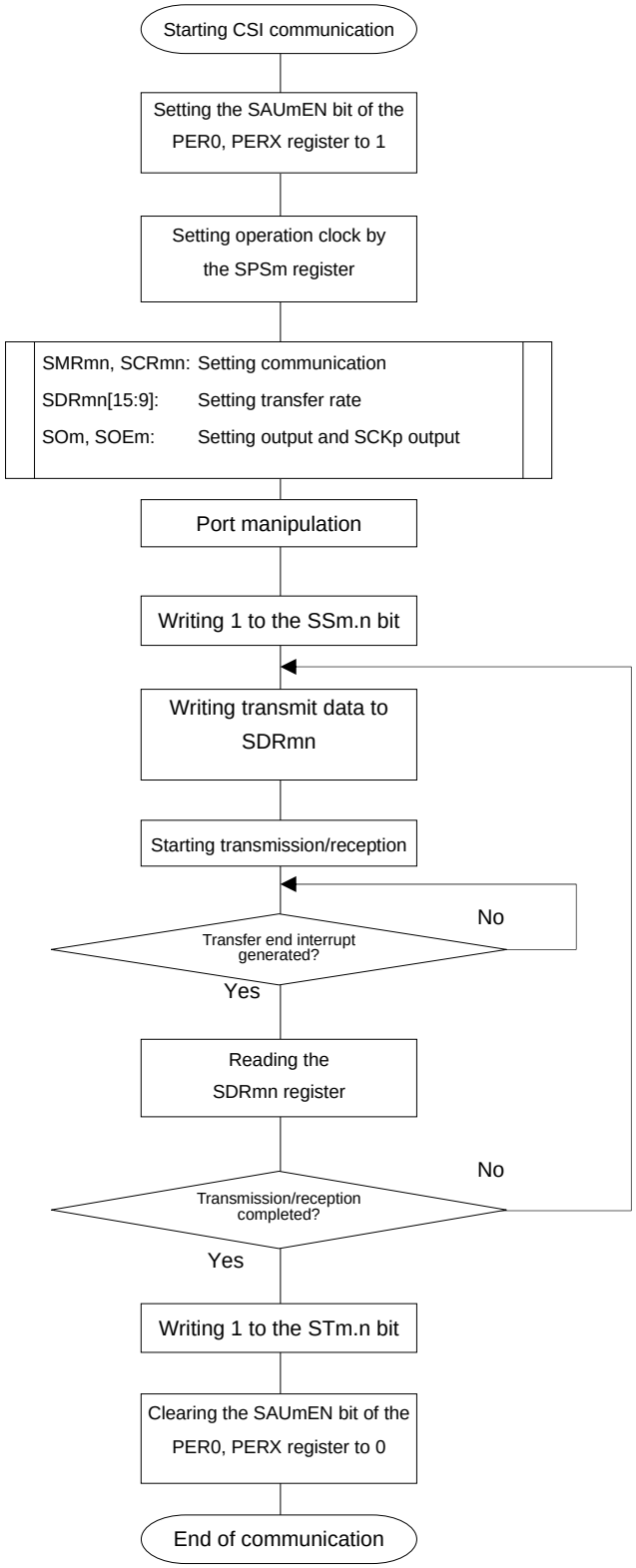
(3) Processing flow (in single-transmission/reception mode)

Figure 13-53. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

Figure 13-54. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)

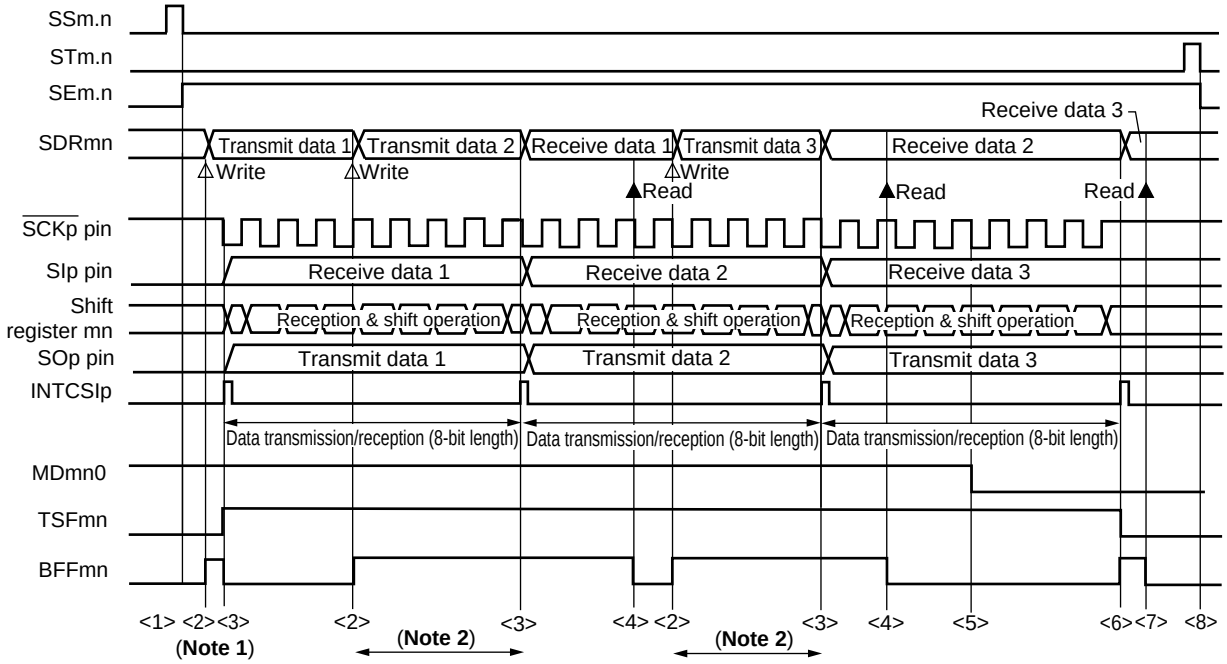


Specify the initial settings while the SEm.n bit of the serial channel enable status register m (SEm) is 0 (operation is stopped).

Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 13-55. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 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.

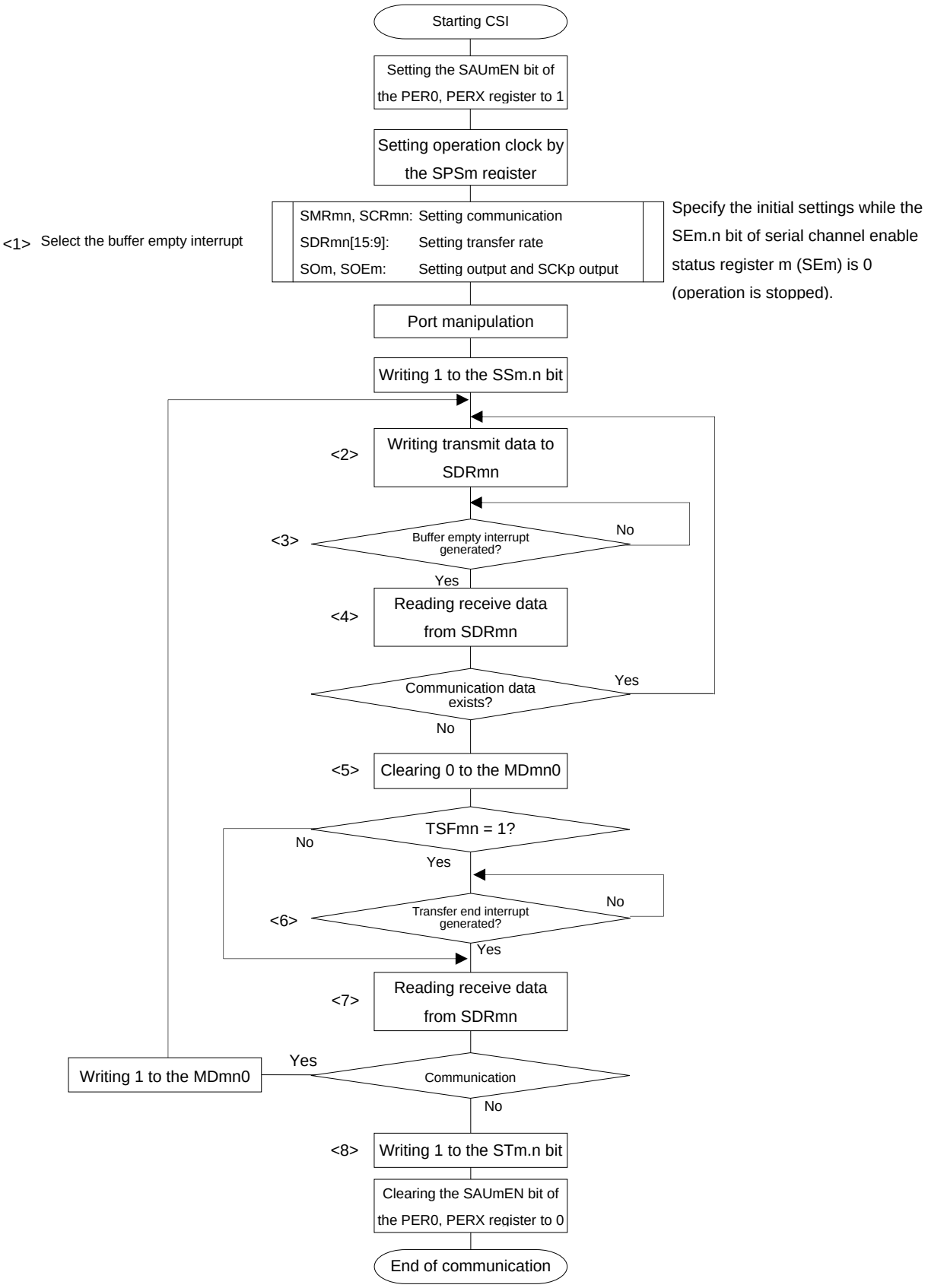
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 13-55 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

Figure 13-56. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 13-55 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

13.5.4 Slave transmission

Slave transmission is that the RL78/F12 transmits data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSIS0	CSIS1
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 0 of SAUS	Channel 1 of SAUS
Pins used	$\overline{SCK00}$, SO00	$\overline{SCK01}$, SO01	$\overline{SCK10}$, SO10	$\overline{SCK11}$, SO11	$\overline{SCK20}$, SO20	$\overline{SCK21}$, SO21	$\overline{SCKS0}$, SOS0	$\overline{SCKS1}$, SOS1
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSIS0	INTCSIS1
	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						7 to 16 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 serial clock operation. • 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: Not reversed • CKPmn = 1: Reversed							
Data direction	MSB or LSB first							

Notes 1. Because the external serial clock input to the $\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, $\overline{SCK11}$, $\overline{SCK20}$, $\overline{SCK21}$, $\overline{SCKS0}$, and $\overline{SCKS1}$ pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz]. Set the SPSm register so that f_{MCK}/6 [Hz] equals f_{SCK}/2 or more that is set with the SDRm register.

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

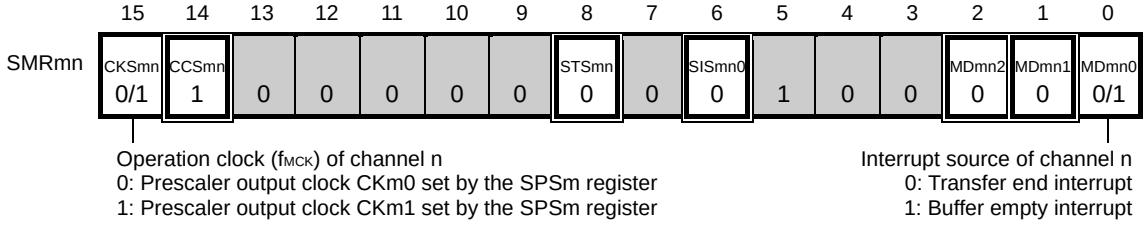
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

2. m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1

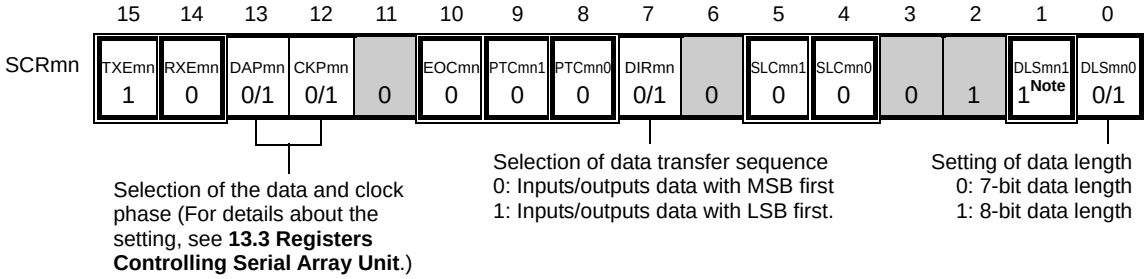
(1) Register setting

Figure 13-57. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (1/2)

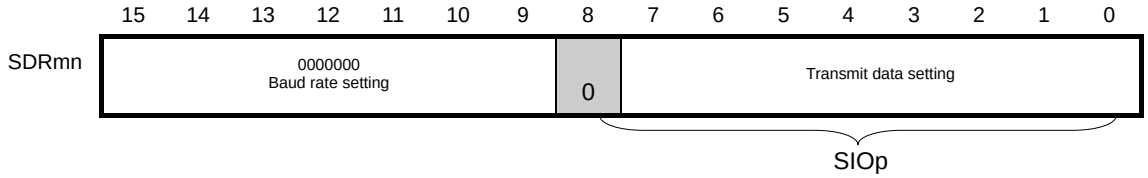
(a) Serial mode register mn (SMRmn)



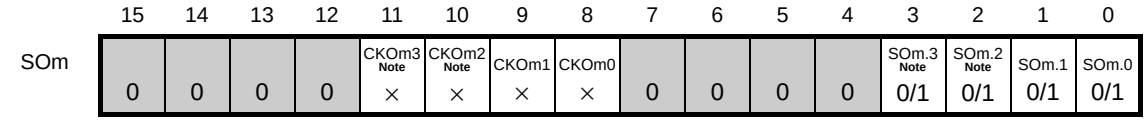
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Sets only the bits of the target channel.



Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

2. ☐: Setting is fixed in the CSI slave transmission mode, ▒ Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-57. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (2/2)

(e) 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	SOEm.3 Note 0/1	SOEm.2 Note 0/1	SOEm.1 0/1	SOEm.0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm.3 Note 0/1	SSm.2 Note 0/1	SSm.1 0/1	SSm.0 0/1

Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

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

Figure 13-58. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSISn) (1/2)

(a) Serial output register S (SOS) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOS	0	0	0	0	0	0	CKOS1 ×	CKOS0 ×	0	0	0	0	0	0	SOS.1 0/1	SOS.0 0/1

(b) Serial output enable register S (SOES) … 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
SOES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOES.1 0/1	SOES.0 0/1

(c) Serial channel start register S (SSS) … 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
SSS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSS.1 0/1	SSS.0 0/1

(d) Serial mode register Sn (SMRSn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRSn	CKSSn 0/1	CCSSn 1	0	0	0	0	0	STSSn 0	0	SISSn0 0	1	0	0	MDSn2 0	MDSn1 0	MDSn0 0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register Sn (SCRSn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRSn	TXESn 1	RXESn 0	DAPSn 0/1	CKPSn 0/1	0	0	PTCSn1 0	PTCSn0 0	DIRSn 0/1	0	SLCSn1 0	SLCSn0 0	DLSSn3 0/1	DLSSn2 0/1	DLSSn1 0/1	DLSSn0 0/1

(f) Serial data register Sn (SDRSn)

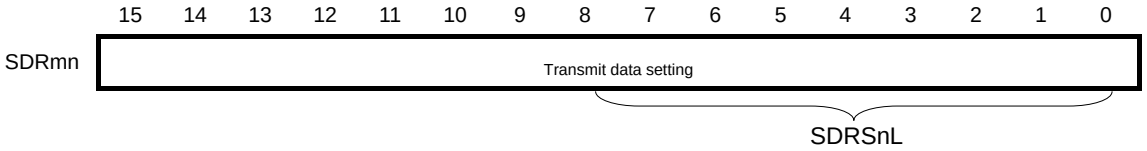
(i) When operation is stopped (SES.n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRSn	0000000 Baud rate setting							0	0	0	0	0	0	0	0	0

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
n: Channel number (n = 0, 1)

Figure 13-58. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSISn) (2/2)

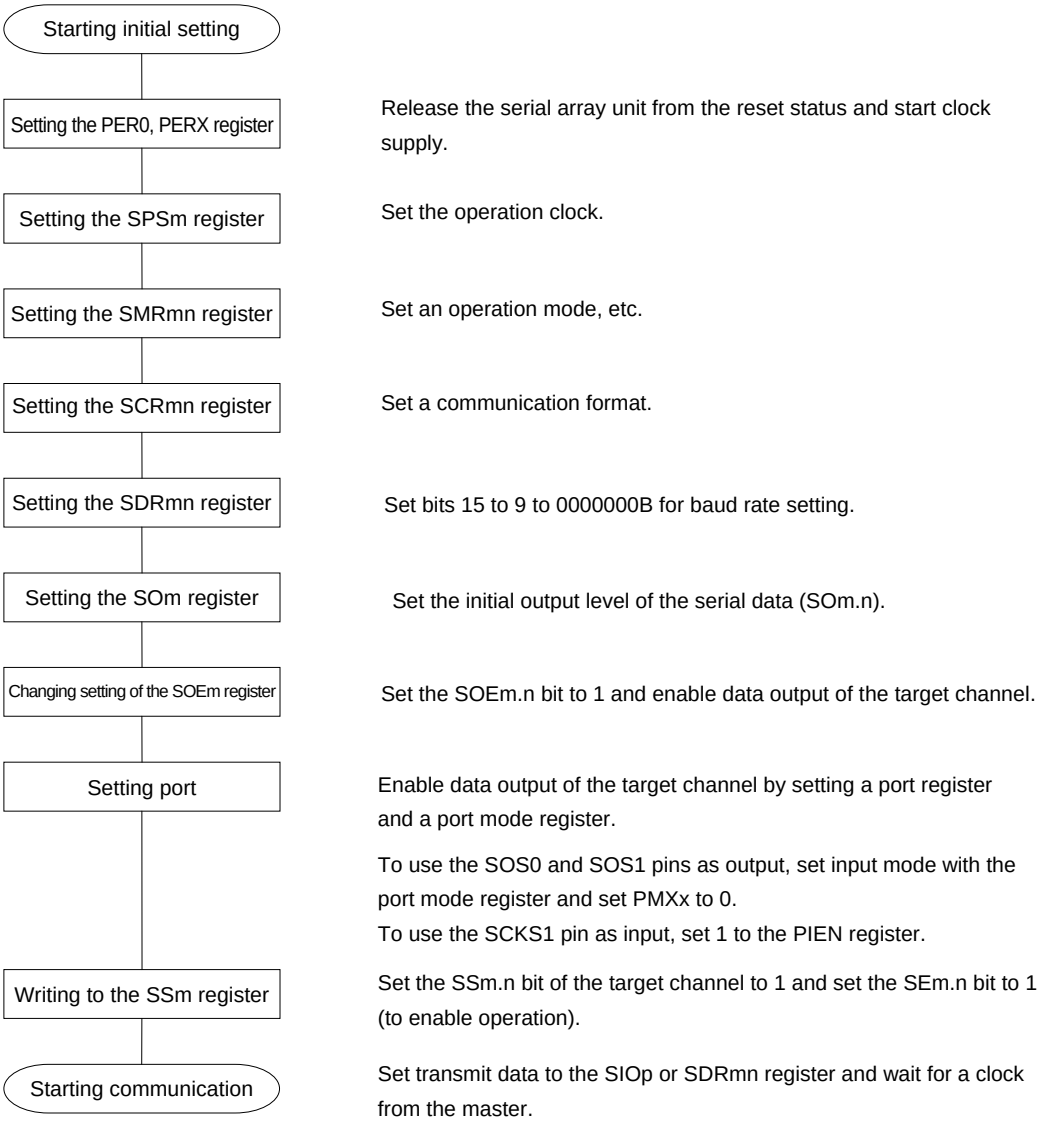
(ii) During operation (SES.n = 1) (lower 8 bits: SDRSnL)



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
n: Channel number (n = 0, 1)

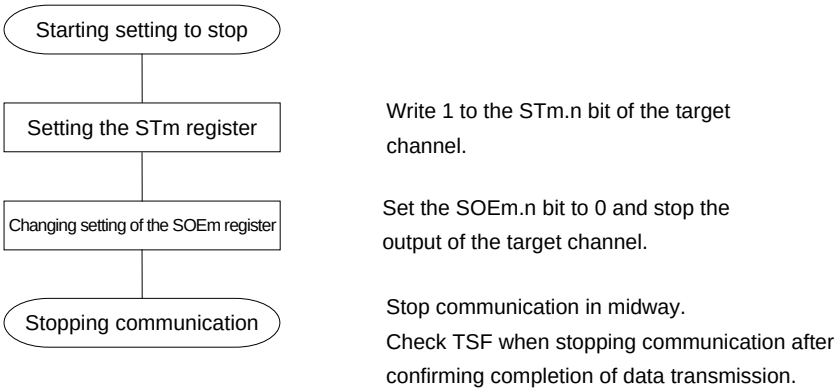
(2) Operation procedure

Figure 13-59. Initial Setting Procedure for Slave Transmission



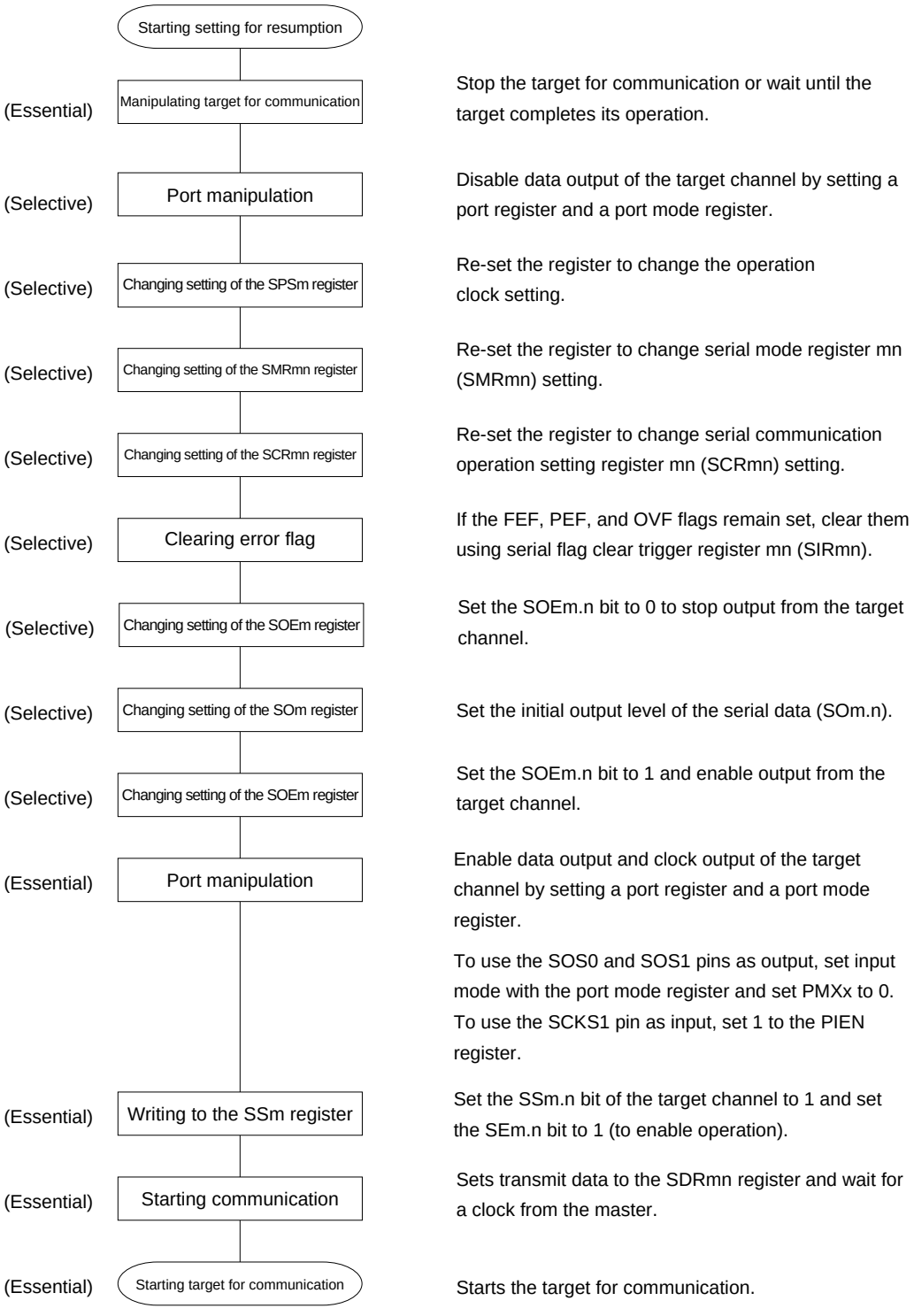
Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 13-60. Procedure for Stopping Slave Transmission



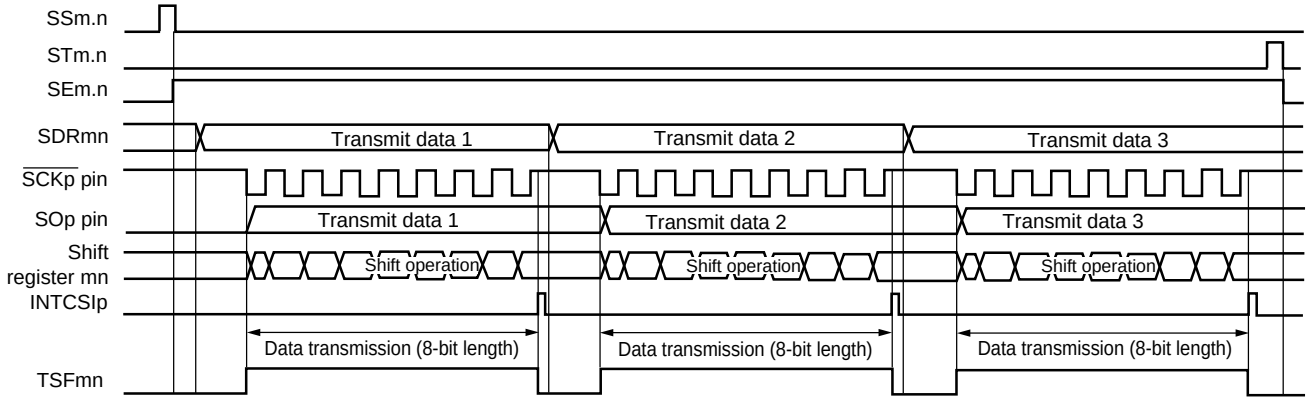
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 13-61 Procedure for Resuming Slave Transmission**).

Figure 13-61. Procedure for Resuming Slave Transmission



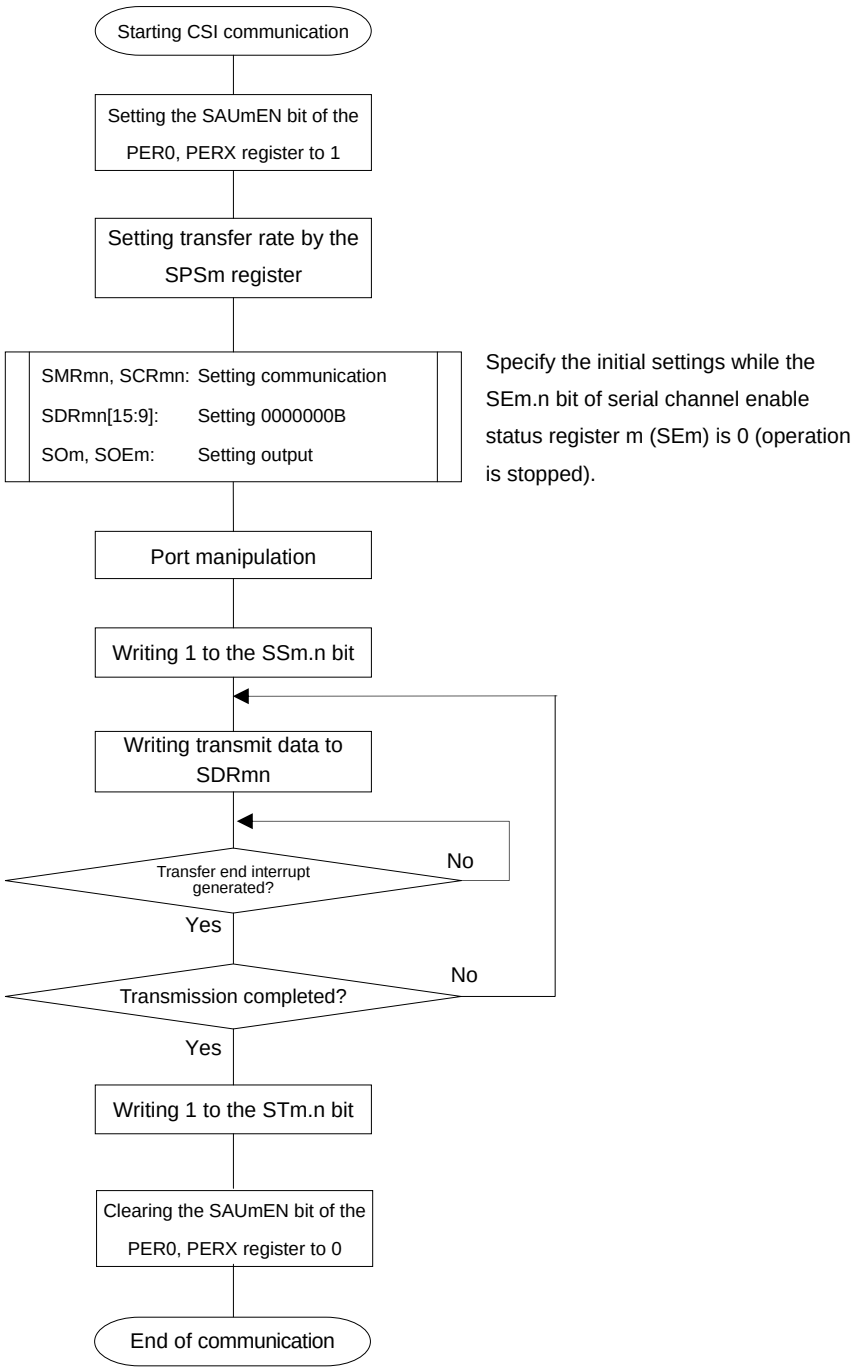
(3) Processing flow (in single-transmission mode)

Figure 13-62. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

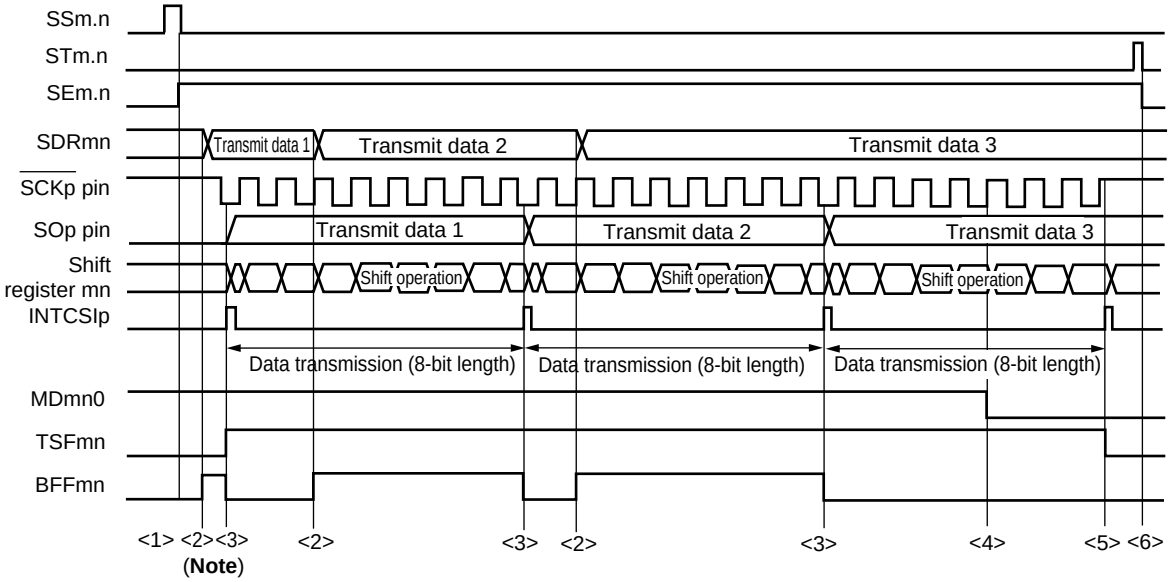
Figure 13-63. Flowchart of Slave Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 13-64. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

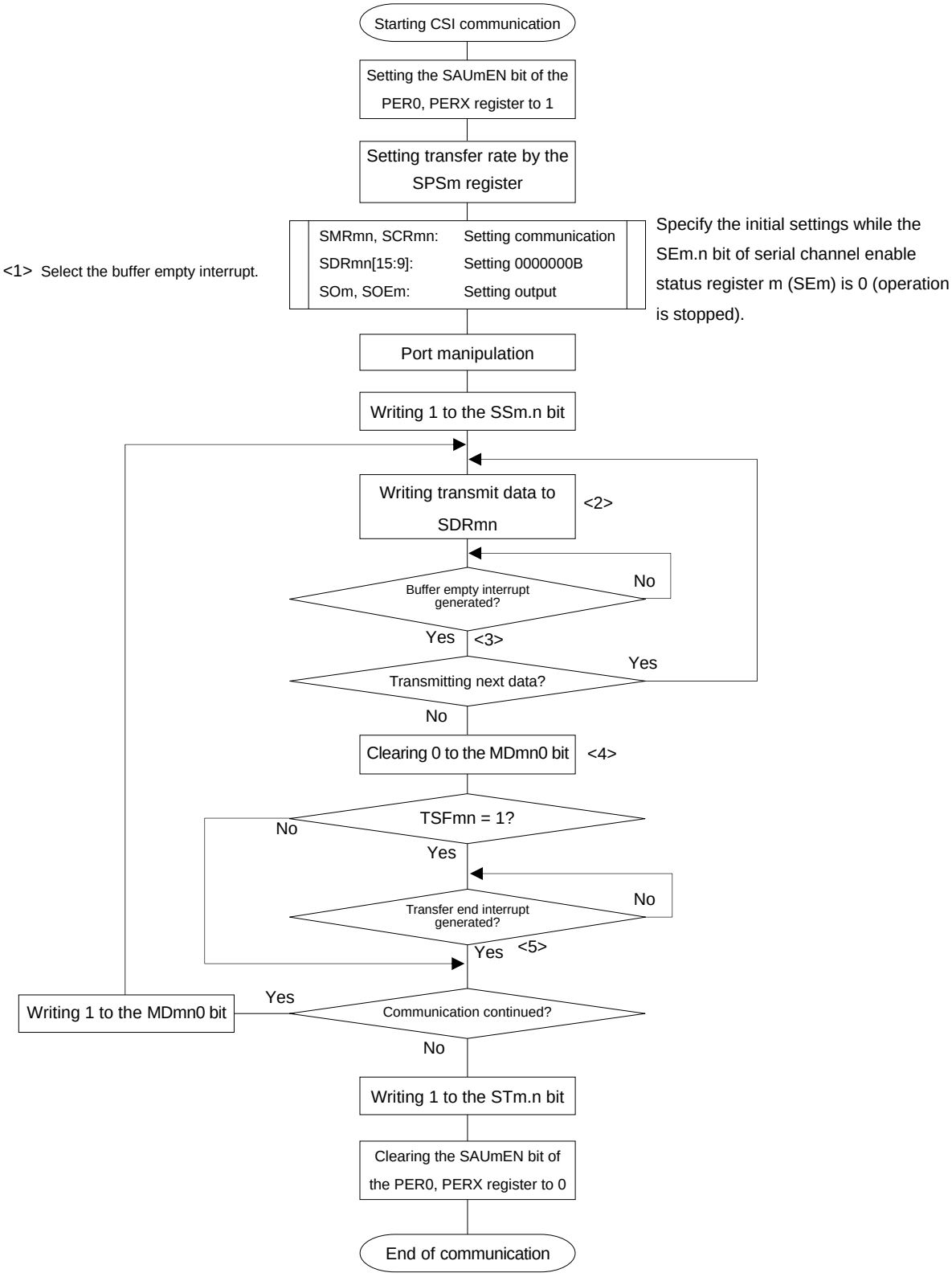


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

Figure 13-65. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 13-64 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

13.5.5 Slave reception

Slave reception is that the RL78/F12 receives data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSIS0	CSIS1
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 0 of SAUS	Channel 1 of SAUS
Pins used	$\overline{SCK00}$, SI00	$\overline{SCK01}$, SI01	$\overline{SCK10}$, SI10	$\overline{SCK11}$, SI11	$\overline{SCK20}$, SI20	$\overline{SCK21}$, SI21	$\overline{SCKS0}$, SIS0	$\overline{SCKS1}$, SIS1
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSIS0	INTCSIS1
	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						7 to 16 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 serial clock operation. • 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: Not reversed • CKPmn = 1: Reversed							
Data direction	MSB or LSB first							

Notes 1. Because the external serial clock input to the $\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, $\overline{SCK11}$, $\overline{SCK20}$, $\overline{SCK21}$, $\overline{SCKS0}$, and $\overline{SCKS1}$ pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz]. Set the SPSm register so that f_{MCK}/6 [Hz] equals f_{SCK} or more that is set with the SDRm register.

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

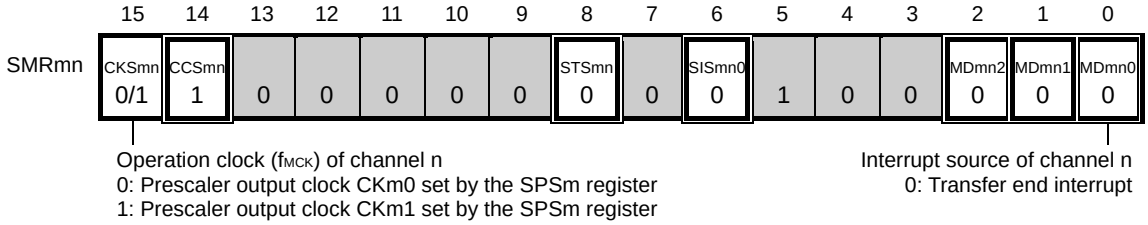
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1

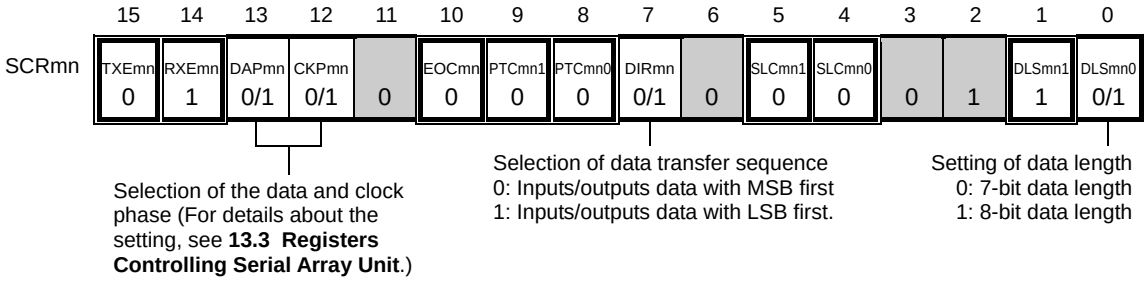
(1) Register setting

Figure 13-66. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

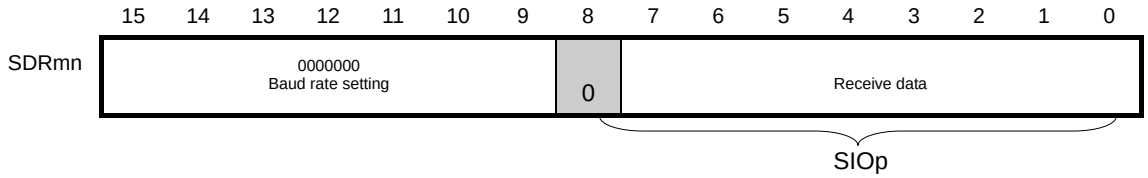
(a) Serial mode register mn (SMRmn)



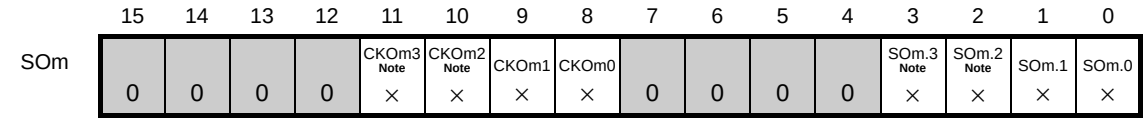
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) …The Register that not used in this mode.



Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

2. □: Setting is fixed in the CSI slave reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-66. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (2/2)

(e) Serial output enable register m (SOEm) …The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm.3 Note ×	SOEm.2 Note ×	SOEm.1 ×	SOEm.0 ×

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm.3 Note 0/1	SSm.2 Note 0/1	SSm.1 0/1	SSm.0 0/1

Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

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

Figure 13-67. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSISn) (1/2)

(a) Serial output register S (SOS) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOS	0	0	0	0	0	0	CKOS1 ×	CKOS0 ×	0	0	0	0	0	0	SOS.1 ×	SOS.0 ×

(b) Serial output enable register S (SOES) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOES.1 ×	SOES.0 ×

(c) Serial channel start register S (SSS) … 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
SSS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSS.1 0/1	SSS.0 0/1

(d) Serial mode register Sn (SMRSn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRSn	CKSSn 0/1	CCSSn 1	0	0	0	0	0	STSSn 0	0	SISSn0 0	1	0	0	MDSn2 0	MDSn1 0	MDSn0 0

Interrupt sources of channel n
0: Transfer end interrupt

(e) Serial communication operation setting register Sn (SCRSn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRSn	TXESn 0	RXESn 1	DAPSn 0/1	CKPSn 0/1	0	0	PTCSn1 0	PTCSn0 0	DIRSn 0/1	0	SLCSn1 0	SLCSn0 0	DLSSn3 0/1	DLSSn2 0/1	DLSSn1 0/1	DLSSn0 0/1

(f) Serial data register Sn (SDRSn)

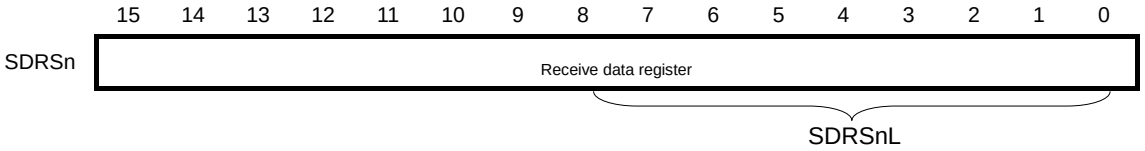
(i) When operation is stopped (SES.n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRSn	0000000 Baud rate setting							0	0	0	0	0	0	0	0	0

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
n: Channel number (n = 0, 1)

Figure 13-67. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSISn) (2/2)

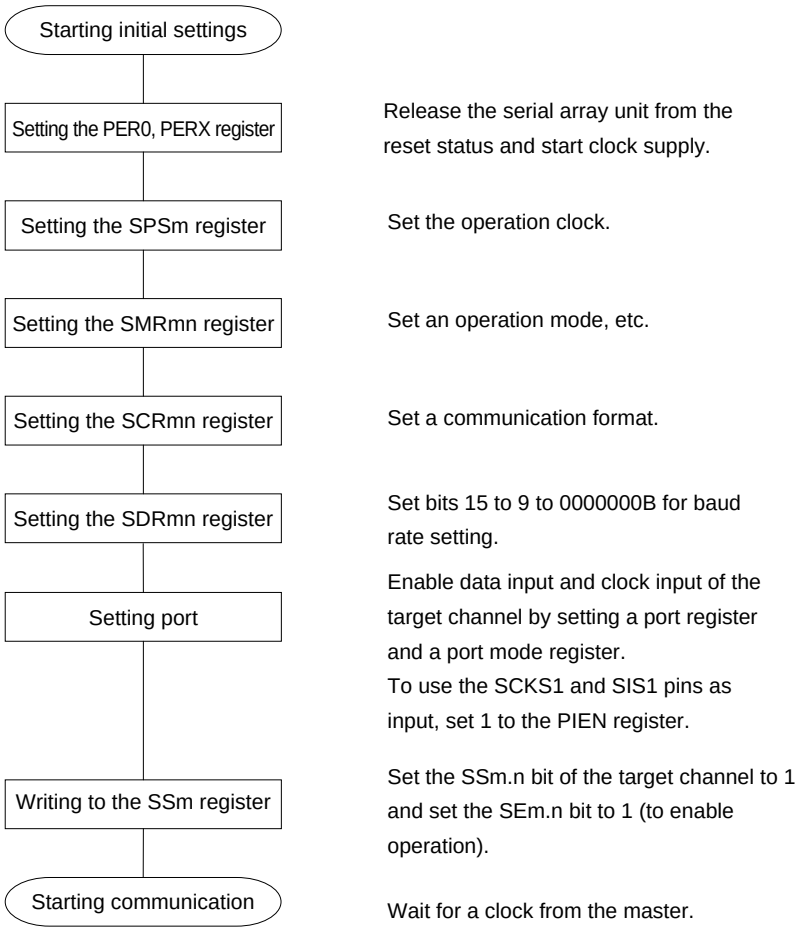
(ii) During operation (SES.n = 1) (lower 8 bits: SDRSnL)



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
n: Channel number (n = 0, 1)

(2) Operation procedure

Figure 13-68. Initial Setting Procedure for Slave Reception



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 13-69. Procedure for Stopping Slave Reception

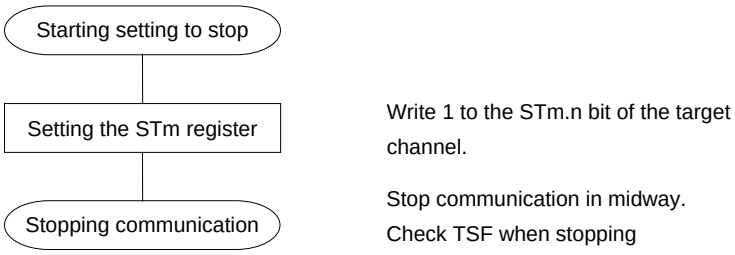
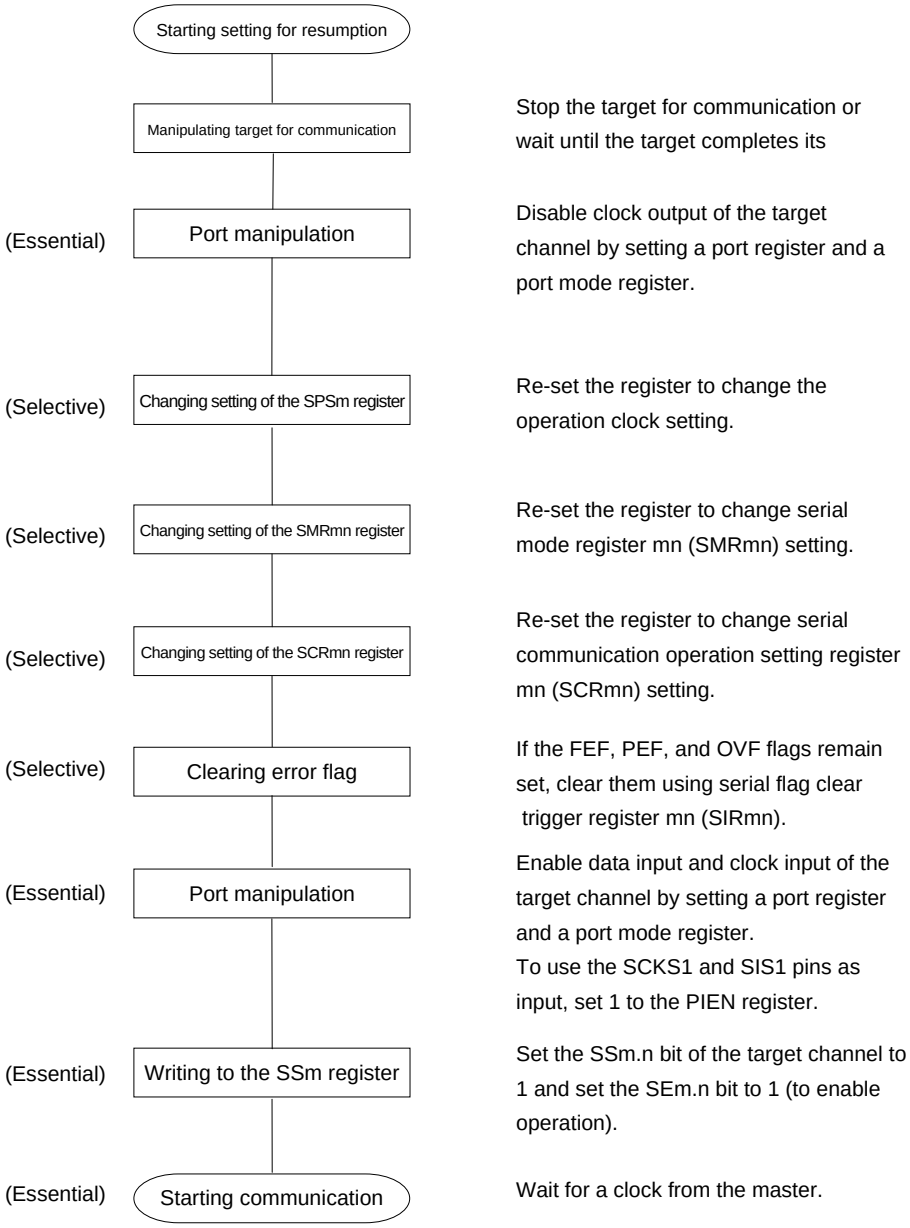
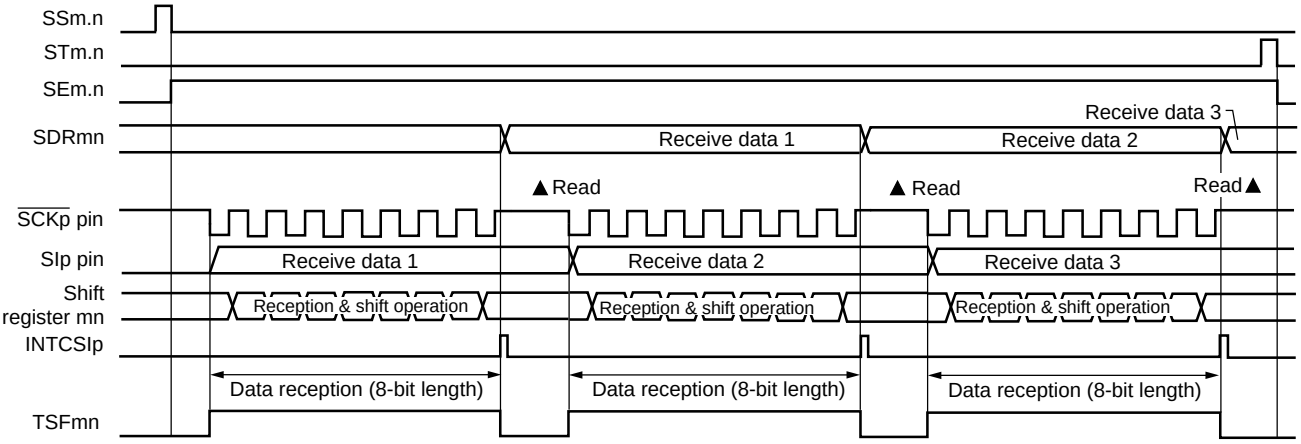


Figure 13-70. Procedure for Resuming Slave Reception



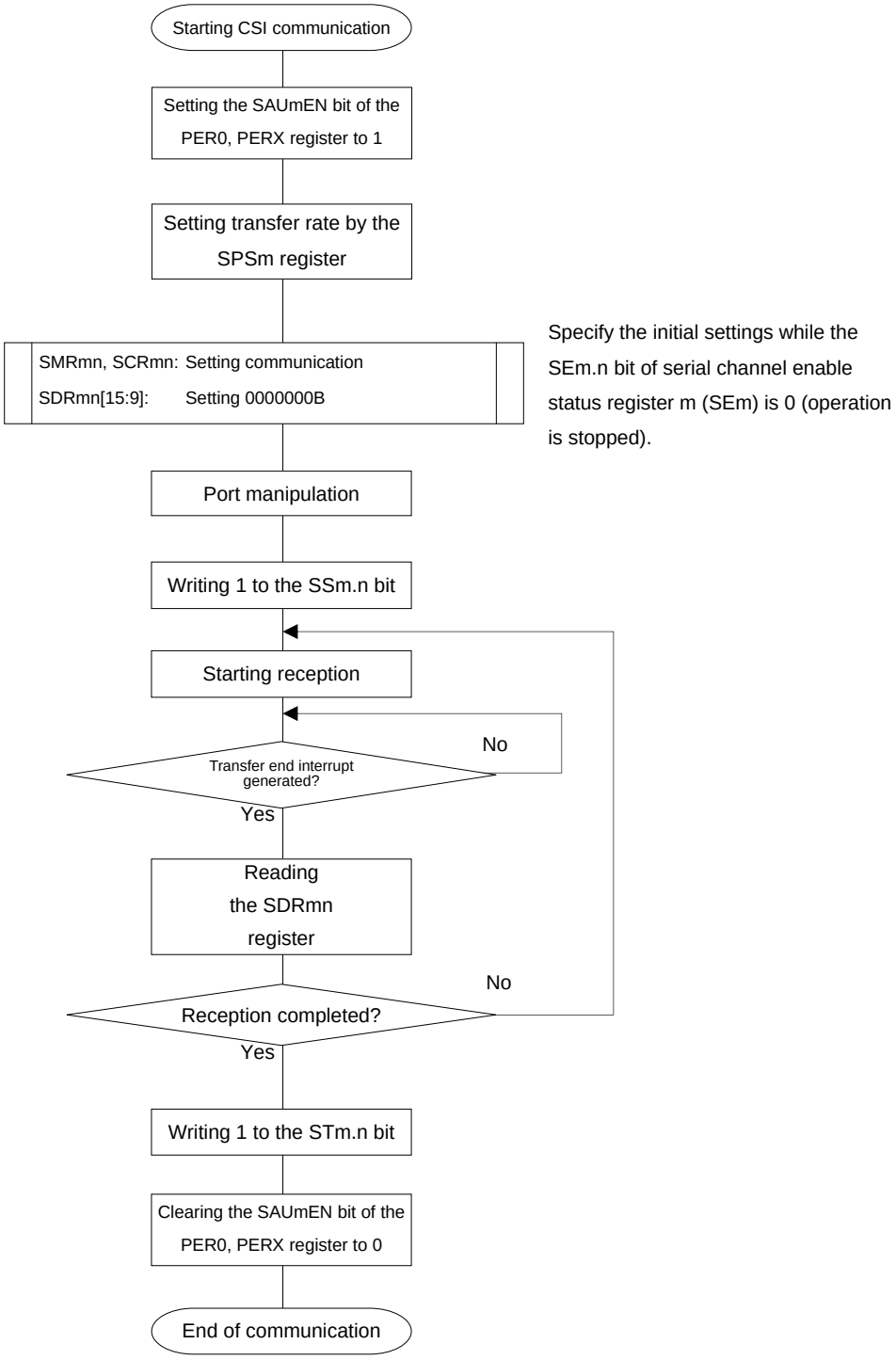
(3) Processing flow (in single-reception mode)

Figure 13-71. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

Figure 13-72. Flowchart of Slave Reception (in Single-Reception Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

13.5.6 Slave transmission/reception

Slave transmission/reception is that the RL78/F12 transmits/receives data to/from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSIS0	CSIS1
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 0 of SAUS	Channel 1 of SAUS
Pins used	$\overline{SCK00}$, SI00, SO00	$\overline{SCK01}$, SI01, SO01	$\overline{SCK10}$, SI10, SO10	$\overline{SCK11}$, SI11, SO11	$\overline{SCK20}$, SI20, SO20	$\overline{SCK21}$, SI21, SO21	$\overline{SCKS0}$, SIS0, SOS0	$\overline{SCKS1}$, SIS1, SOS1
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSIS0	INTCSIS1
	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						7 to 16 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 serial clock operation. • 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: Not reversed • CKPmn = 1: Reversed							
Data direction	MSB or LSB first							

Notes 1. Because the external serial clock input to the $\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, $\overline{SCK11}$, $\overline{SCK20}$, $\overline{SCK21}$, $\overline{SCKS0}$, and $\overline{SCKS1}$ pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz]. Set the SPSm register so that f_{MCK}/6 [Hz] equals f_{SCK} or more that is set with the SDRm register.

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

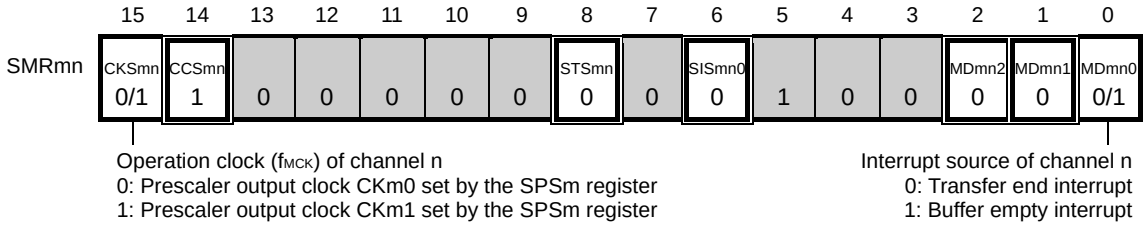
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1

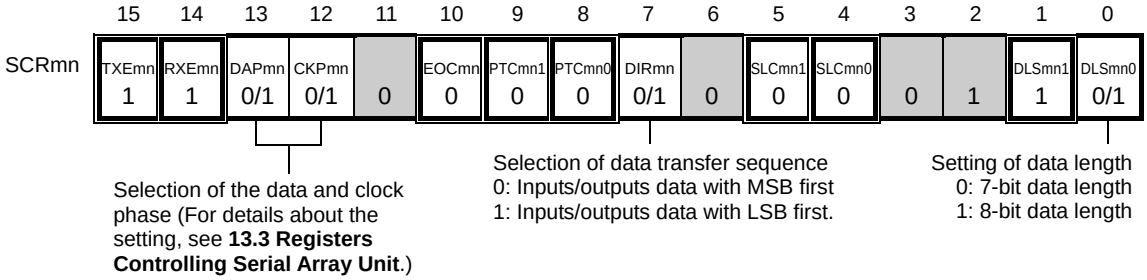
(1) Register setting

Figure 13-73. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (1/2)

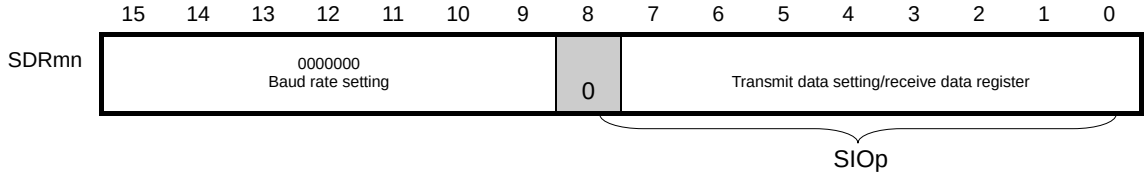
(a) Serial mode register mn (SMRmn)



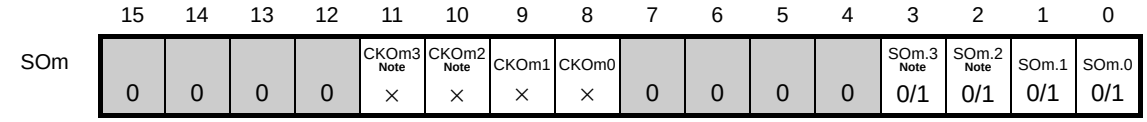
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Sets only the bits of the target channel.



Note Serial array unit 0 only.

Caution Be sure to set transmit data to the SDR register before the clock from the master is started.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

2. ☐: Setting is fixed in the CSI slave 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

Figure 13-73. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (2/2)

(e) 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	SOEm.3 Note 0/1	SOEm.2 Note 0/1	SOEm.1 0/1	SOEm.0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm.3 Note 0/1	SSm.2 Note 0/1	SSm.1 0/1	SSm.0 0/1

Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

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

Figure 13-74. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSISn) (1/2)

(a) Serial output register S (SOS) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOS	0	0	0	0	0	0	CKOS1 ×	CKOS0 ×	0	0	0	0	0	0	SOS.1 0/1	SOS.0 0/1

(b) Serial output enable register S (SOES) … 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
SOES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOES.1 0/1	SOES.0 0/1

(c) Serial channel start register S (SSS) … 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
SSS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSS.1 0/1	SSS.0 0/1

(d) Serial mode register Sn (SMRSn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRSn	CKSSn 0/1	CCSSn 1	0	0	0	0	0	STSSn 0	0	SISS0n 0	1	0	0	MDSn2 0	MDSn1 0	MDSn0 0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register Sn (SCRSn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRSn	TXESn 1	RXESn 1	DAPSn 0/1	CKPSn 0/1	0	0	PTCSn1 0	PTCSn0 0	DIRSn 0/1	0	SLCSn1 0	SLCSn0 0	DLSSn3 0/1	DLSSn2 0/1	DLSSn1 0/1	DLSSn0 0/1

(f) Serial data register Sn (SDRSn)

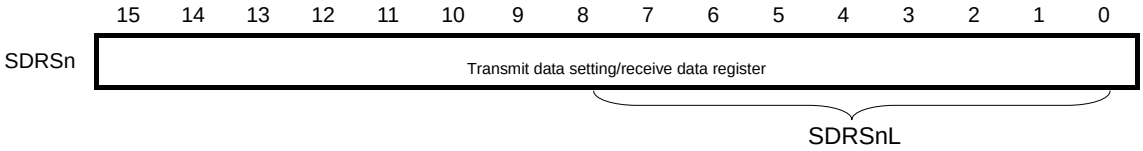
(i) When operation is stopped (SES.n = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRSn	0000000 Baud rate setting							0	0	0	0	0	0	0	0	0

Remark ☐: Setting is fixed in the CSI slave 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
n: Channel number (n = 0, 1)

Figure 13-74. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSISn) (2/2)

(ii) During operation (SES.n = 1) (lower 8 bits: SDRSnL)



Caution Be sure to set transmit data to the SDRSnL register before the clock from the master is started.

Remark ☐: Setting is fixed in the CSI slave transmission/reception mode, ▒: Setting disabled (set to the initial value)

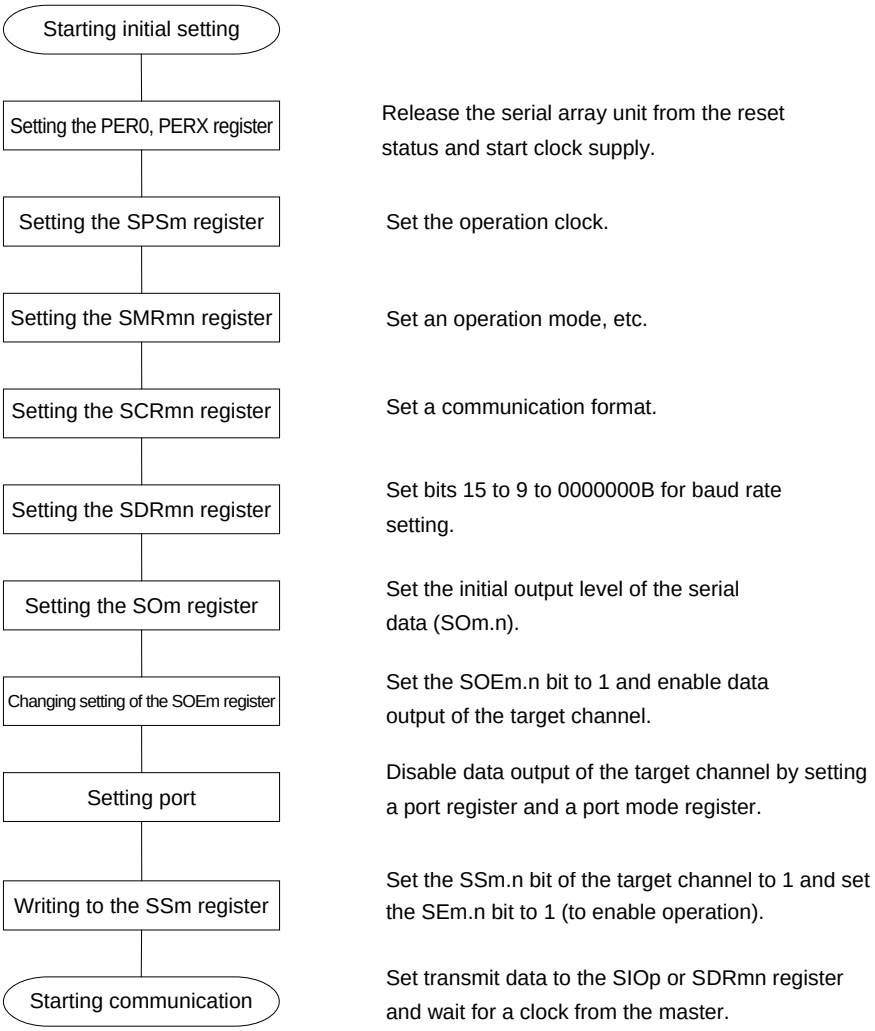
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

n: Channel number (n = 0, 1)

(2) Operation procedure

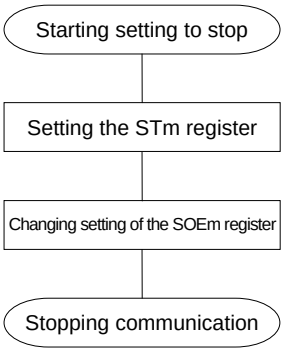
Figure 13-75. Initial Setting Procedure for Slave Transmission/Reception



Cautions 1. After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

2. Be sure to set transmit data to the SDR register before the clock from the master is started.

Figure 13-76. Procedure for Stopping Slave Transmission/Reception



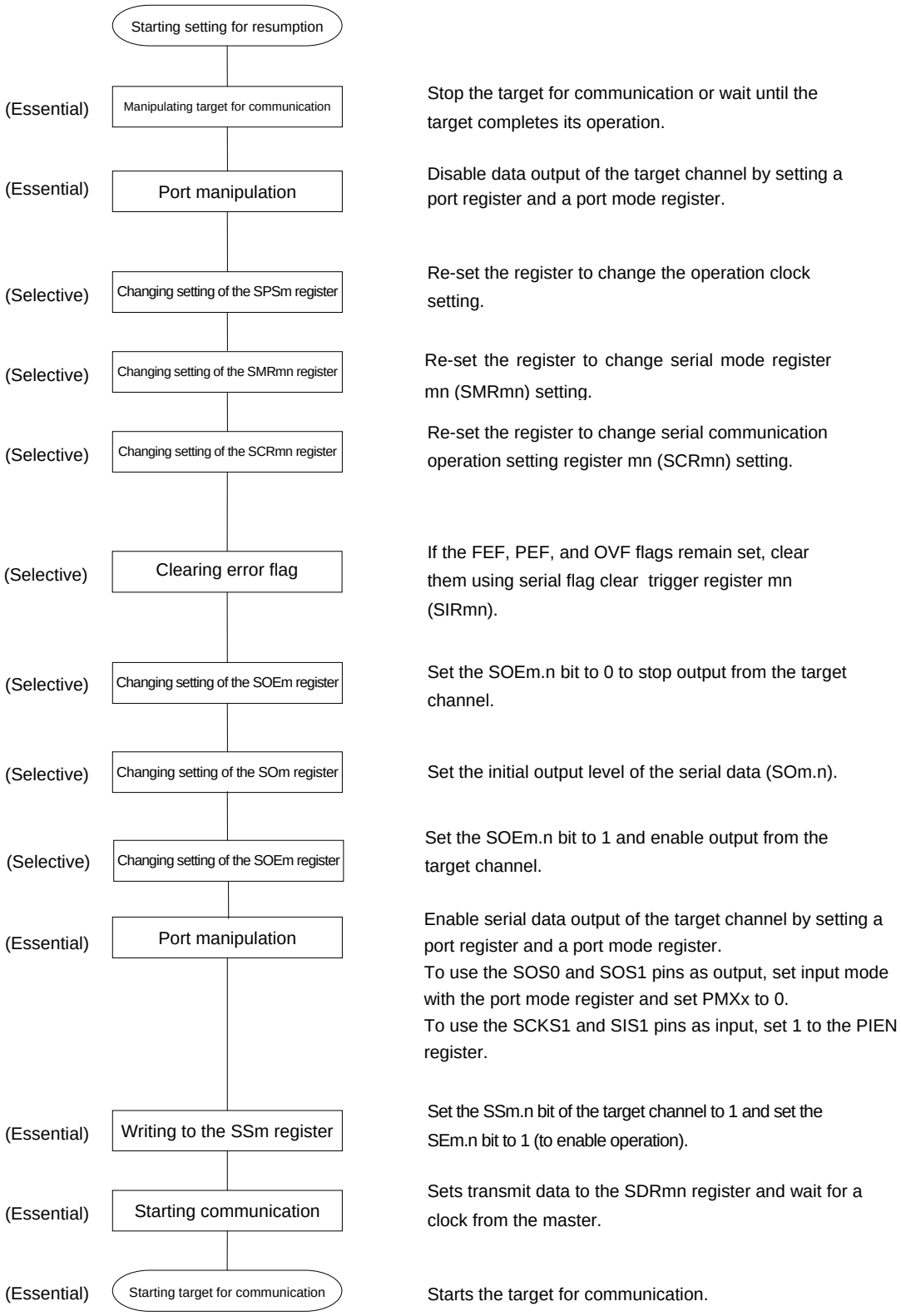
Write 1 to the STm.n bit of the target channel.

Set the SOEm.n bit to 0 and stop the output of the target channel.

Stop communication in midway.
Check TSF when stopping communication after confirming completion of data transmission.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 13-77 Procedure for Resuming Slave Transmission/ Reception**).

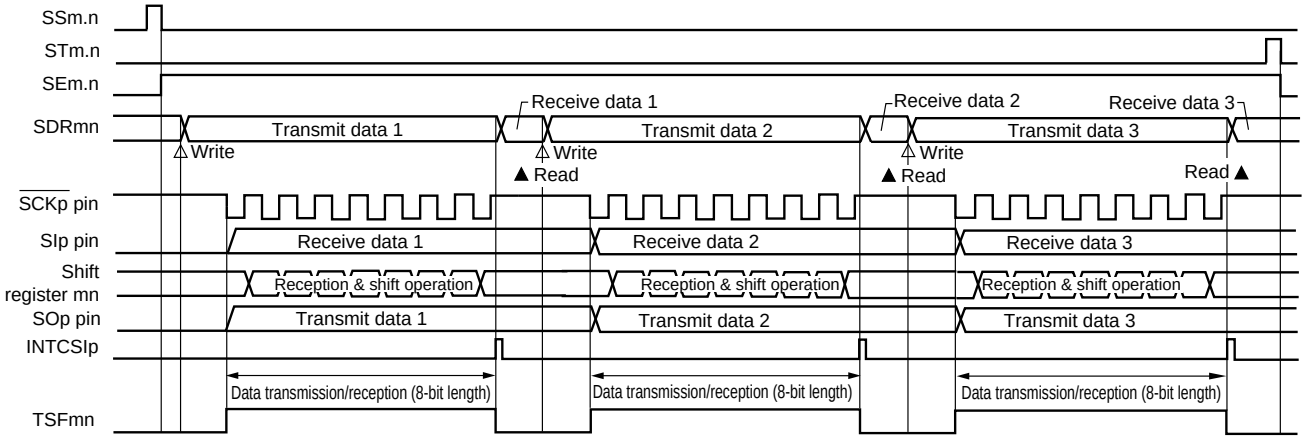
Figure 13-77. Procedure for Resuming Slave Transmission/Reception



Caution Be sure to set transmit data to the SDR register before the clock from the master is started.

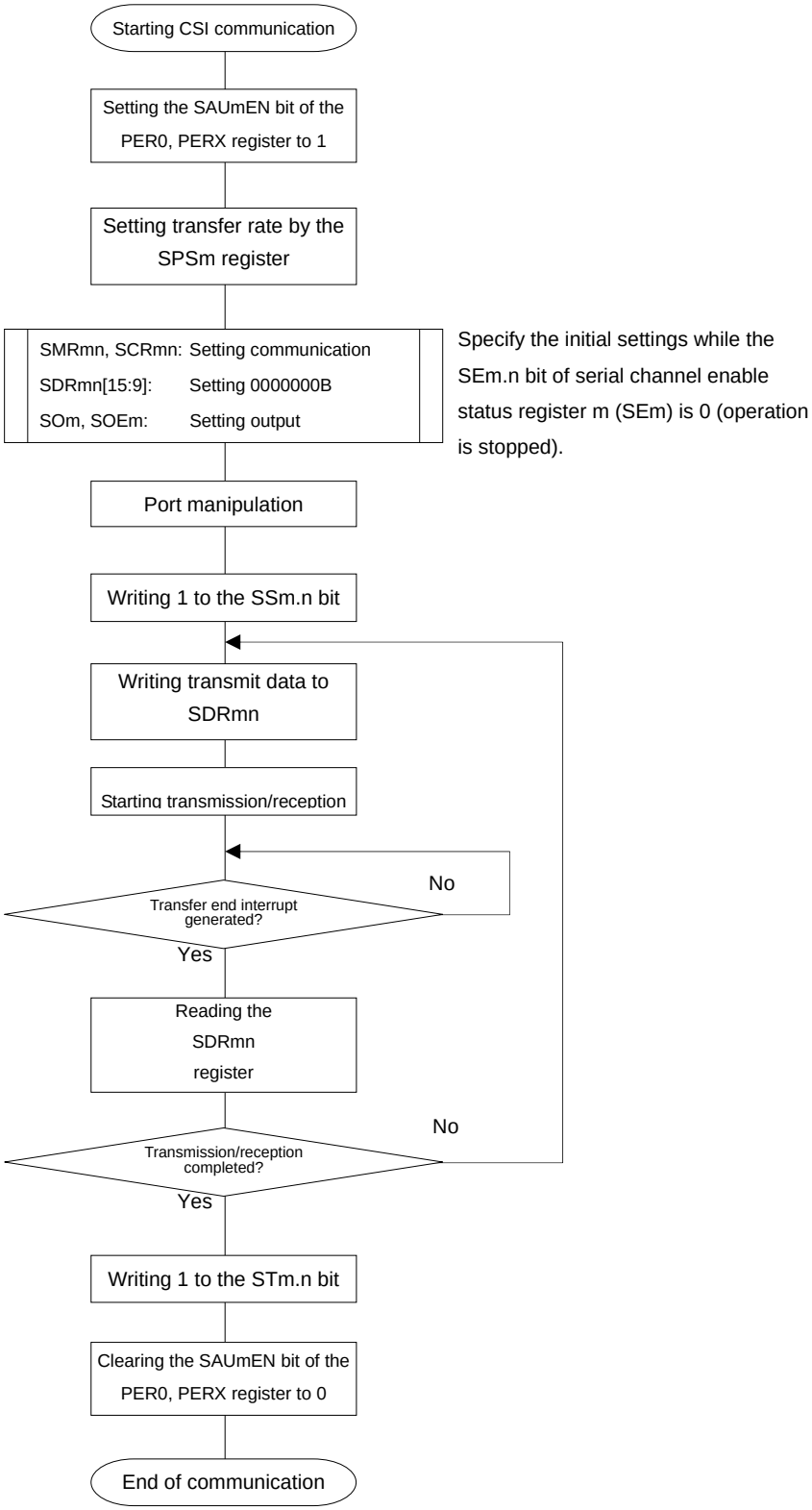
(3) Processing flow (in single-transmission/reception mode)

Figure 13-78. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

Figure 13-79. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

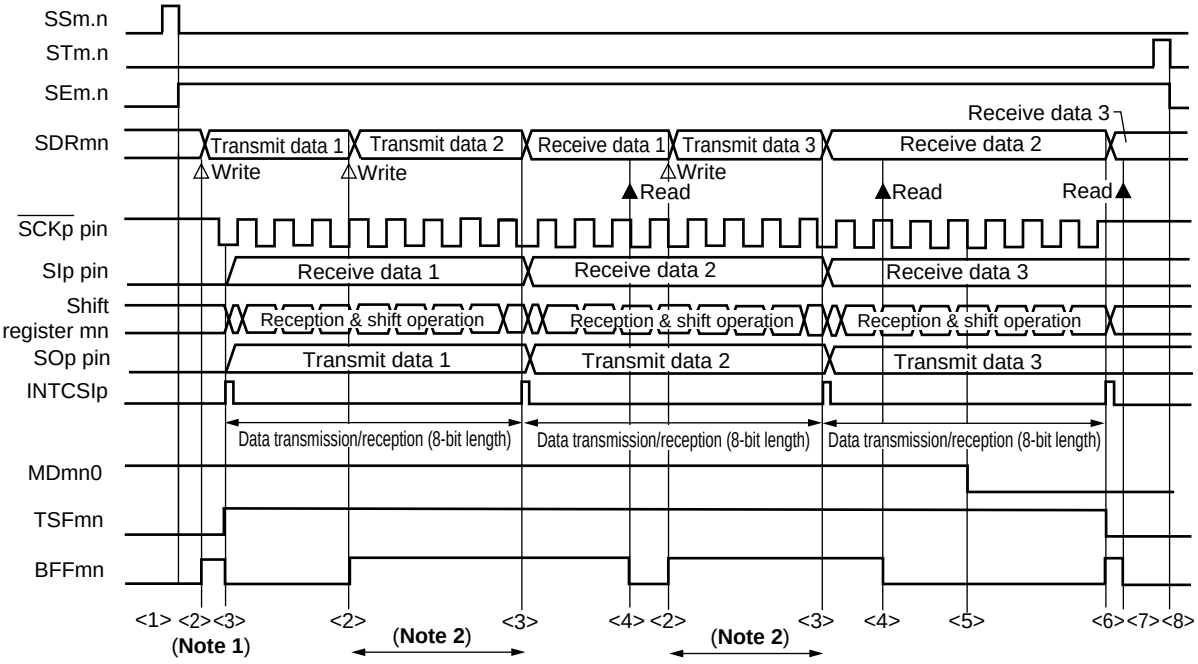


Cautions 1. After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

2. Be sure to set transmit data to the SDR register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 13-80. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 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.

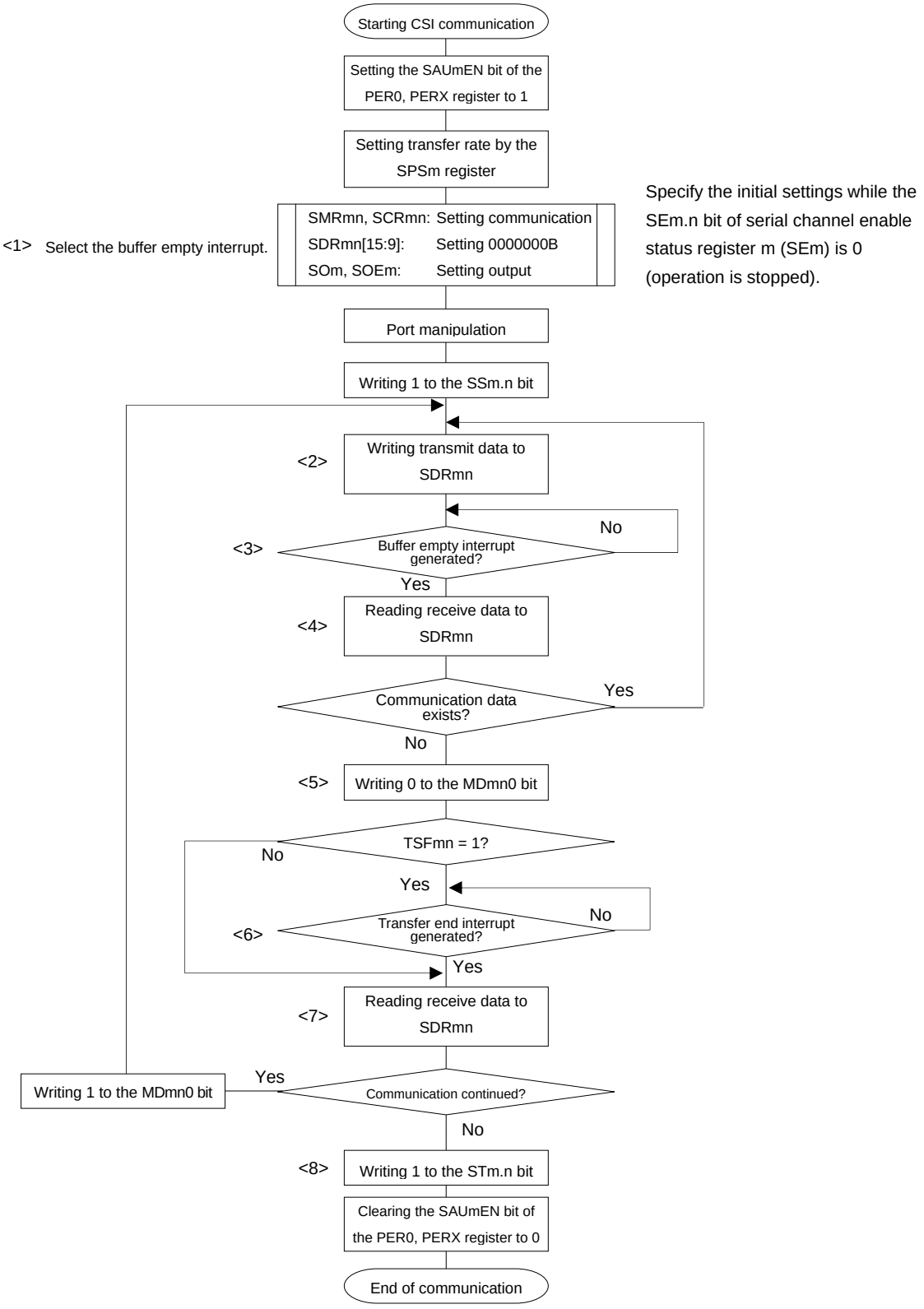
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 13-81 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), mn = 00 to 03, 10, 11, S0, S1

Figure 13-81. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Cautions 1. After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

2. Be sure to set transmit data to the SDR register before the clock from the master is started.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 13-80 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

13.5.7 SNOOZE mode function (only CSI00)

The SNOOZE mode function enables the CSI to perform reception by detection of the $\overline{\text{SCKp}}$ pin input during STOP mode. Usually, the CSI does not operate in STOP mode. However, when using this mode function, the CSI can perform receive operation without CPU operation, by detection of the $\overline{\text{SCKp}}$ pin input. SNOOZE mode can be specified only for CSI00.

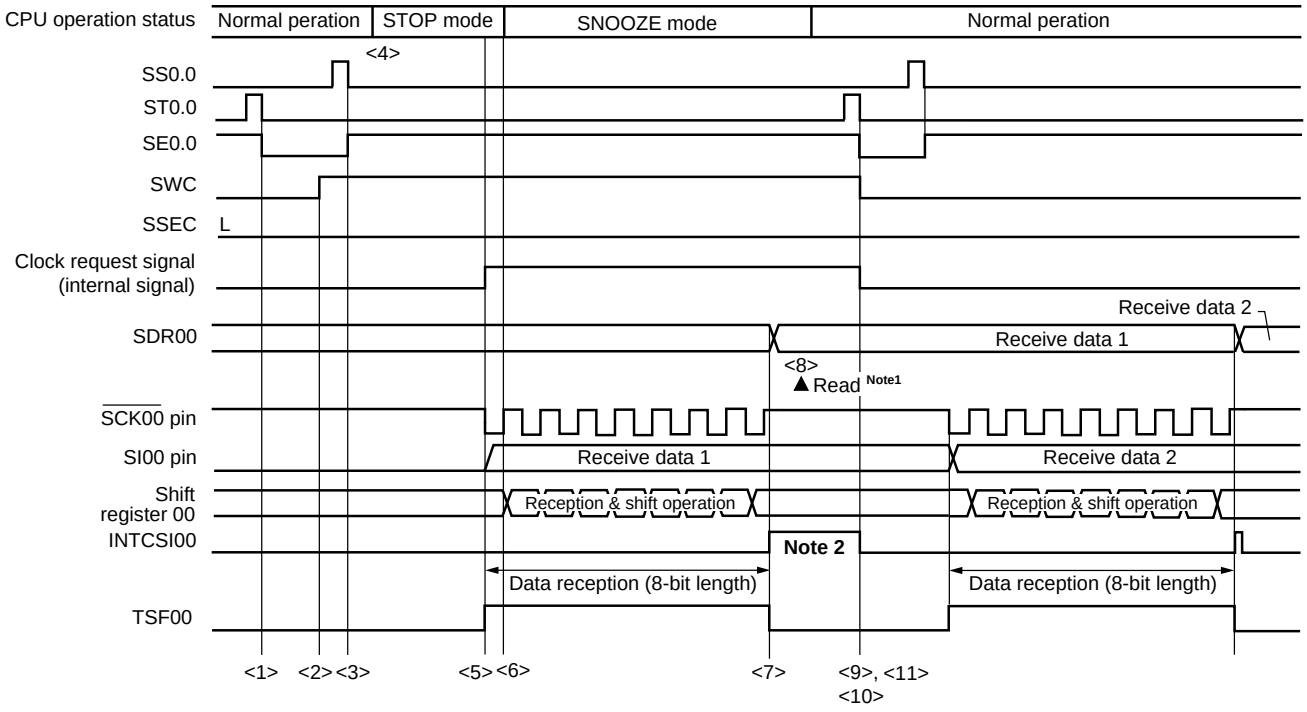
When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 immediately before switching to the STOP mode.

Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

2. The maximum transfer rate when using CSI00 in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 13-82. Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)

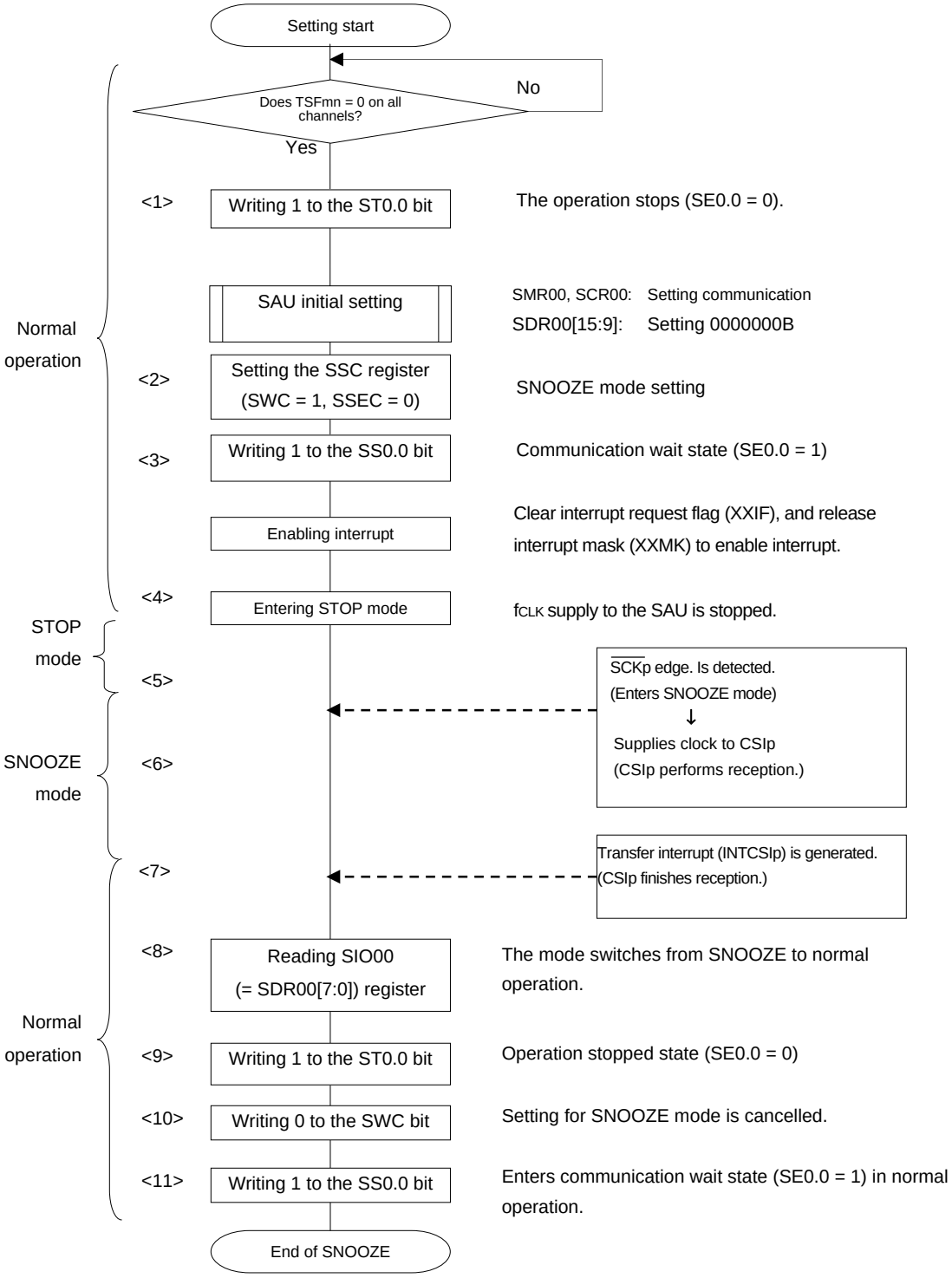


Note Only read received data while SWC = 1 and before the next edge of the SCK00 pin $\overline{\text{input}}$ is detected.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, be sure to set the STm0 bit to 1 (the SEm0 bit is cleared and the operation stops). After completion of reception operation, clear the SWCm bit to cancel the SNOOZE mode.

Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 13-83. Flowchart of SNOOZE Mode Operation (once startup)**.

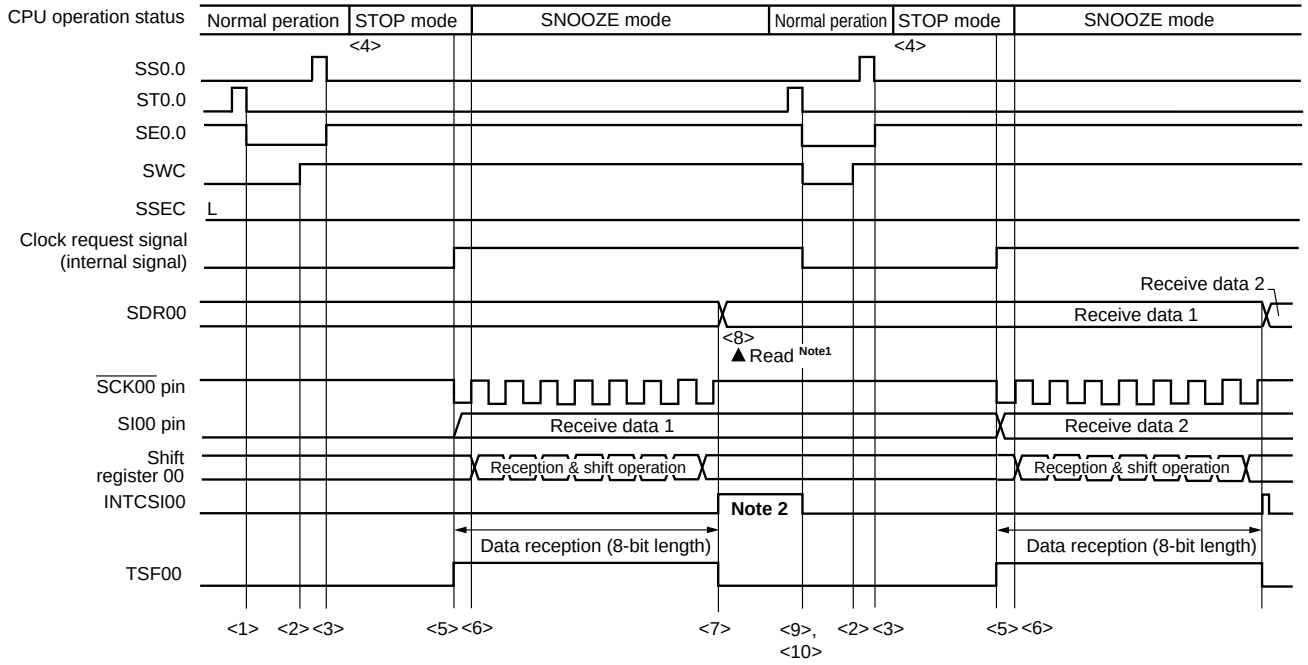
Figure 13-83. Flowchart of SNOOZE Mode Operation (once startup)



Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 13-82. Timing Chart of SNOOZE Mode Operation (once startup)**.

(2) SNOOZE mode operation (continuous startup)

Figure 13-84. Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)

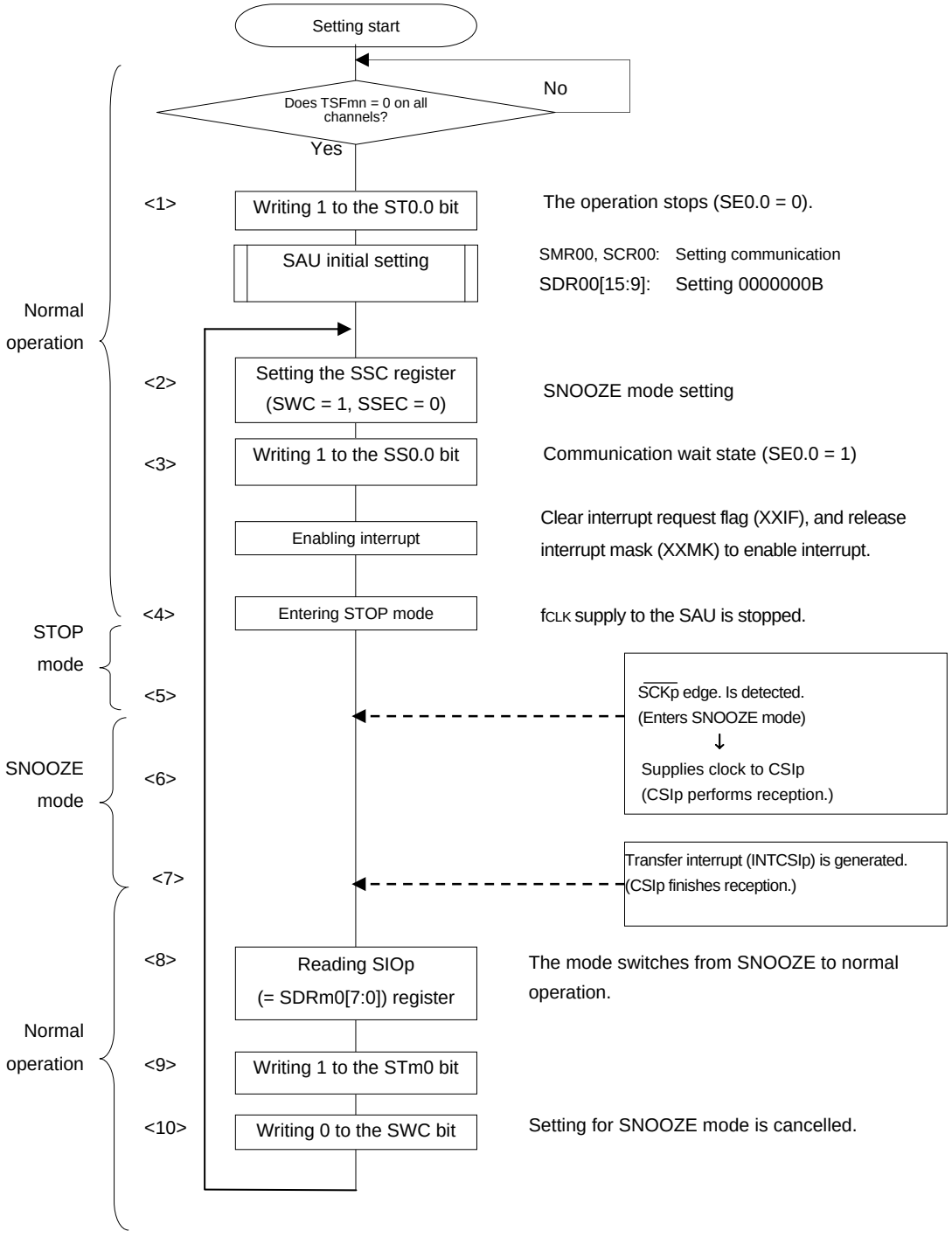


Note Only read received data while SWC = 1 and before the next edge of the $\overline{\text{SCK00}}$ pin input is detected.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, be sure to set the STm0 bit to 1 (the SEm0 bit is cleared and the operation stops). After completion of reception operation, clear the SWCm bit to cancel the SNOOZE mode.

Remark <1> to <10> in the figure correspond to <1> to <10> in **Figure 13-85. Flowchart of SNOOZE Mode Operation (continuous startup)**.

Figure 13-85. Flowchart of SNOOZE Mode Operation (continuous startup)



Remark <1> to <10> in the figure correspond to <1> to <10> in **Figure 13-84. Timing Chart of SNOOZE Mode Operation (continuous startup)**.

13.5.8 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSIS0, CSIS1) 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 ($\overline{SCK}$) 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 13-2. 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	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz
Other than above									Setting prohibited	

<R> **Note** Stop the operation of the serial array unit (SAU) (by setting bits 3 to 0 of ST0 register and bits 1 and 0 of ST1 and STS register to 1) before changing operation clock (f_{CLK}) selection (by changing the system clock control register (CKC) value).

Remarks 1. X: Don't care

2. m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1

13.5.9 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSIS0, CSIS1) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSIS0, CSIS1) communication is described in Figure 13-86.

Figure 13-86. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1

13.6 Operation of UART (UART0 to UART2, UARTS0) 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). The LIN-bus can be implemented by using timer array unit 0 with an external interrupt (INTP0).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits (UART0)
- Data length of 7 or 8 bits (UART1, UART2)
- Data length of 7, 8, or 9 bits (UARTS0)
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART0 reception (channel 1 of unit 0) supports the SNOOZE mode. When RxD0 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only UART0 can be specified for the reception baud rate adjustment function.

The LIN-bus is accepted in UART2 (channels 0 and 1 of unit 1) (30, 32, 48, and 64-pin products only).

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit 0

UART0 uses channels 0 and 1 of SAU0. UART1 uses channels 2 and 3 of SAU0. UART2 uses channels 0 and 1 of SAU1. UARTS0 uses channels 0 and 1 of SAUS.

- 20-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	–		–
	2	–	–	–
	3	–		–
1	0	–	–	–
	1	–		–
S	0	CSIS0	UARTS0	–
	1	–		–

- 30 and 32-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	–		–
	2	–	UART1	–
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	–		–
S	0	CSIS0	UARTS0	–
	1	–		–

- 48-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	–	UART1	–
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
S	0	CSIS0	UARTS0	–
	1	–		–

- 64-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
S	0	CSIS0	UARTS0	–
	1	CSIS1		–

Caution When using serial array unit as UARTs, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UARTs.

UART performs the following four types of communication operations.

- UART transmission (See **13.6.1**.)
- UART reception (See **13.6.2**.)
- LIN transmission (UART2 only) (See **13.7.1**.)
- LIN reception (UART2 only) (See **13.7.2**.)

13.6.1 UART transmission

UART transmission is an operation to transmit data from the RL78/F12 to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0	UART1	UART2	UARTS0
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 0 of SAUS
Pins used	TxD0	TxD1	TxD2	TxDS0
Interrupt	INTST0	INTST1	INTST2	INTSTS0
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	7, 8, or 9 bits	7 or 8 bits		7 to 9, or 16 bits
Transfer rate	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note			Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [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 AC characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

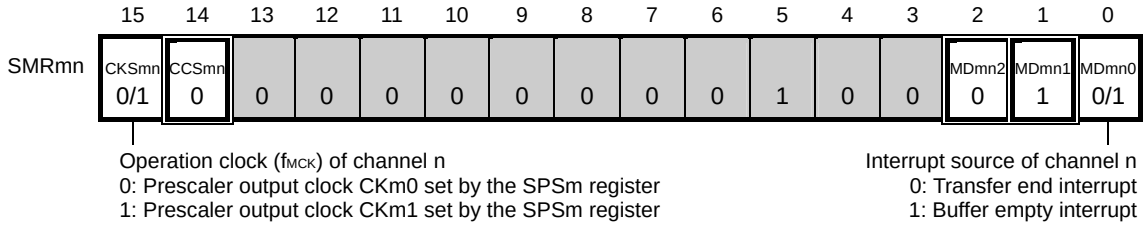
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0, 1, S), n: Channel number (n = 0, 2), mn = 00, 02, 10, S0

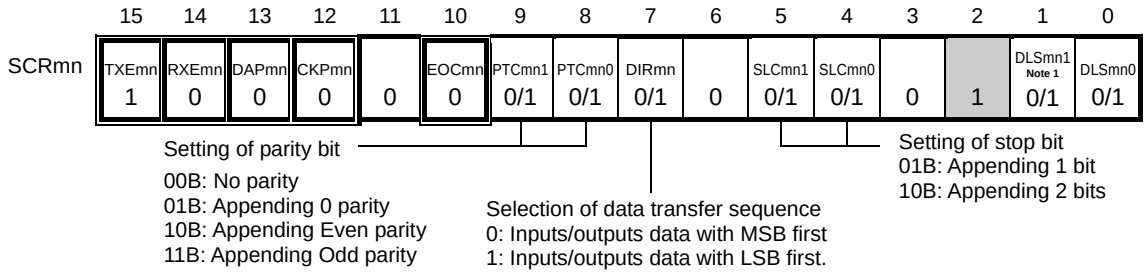
(1) Register setting

Figure 13-87. Example of Contents of Registers for UART Transmission of UART (UART0 to UART2) (1/2)

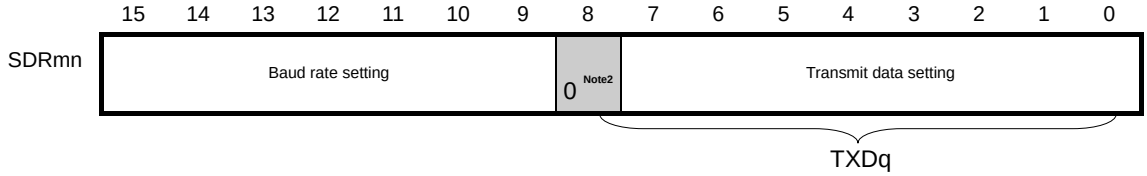
(a) Serial mode register mn (SMRmn)



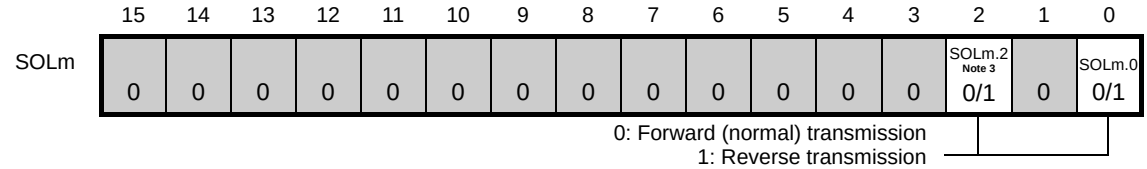
(b) Serial communication operation setting register mn (SCRmn)



(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.



Notes 1. The SCR00 register (UART0) only. This is fixed to 1 for the SCR02 and SCR10 registers.

2. When UART0 performs 9-bit communication (by setting the DLS001 and DLS000 bits of the SCR00 register to 0 and 1, respectively), bits 0 to 8 of the SDR00 register are used as the transmission data specification area. Only UART0 can be used to perform 9-bit communication.

3. Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2), mn = 00, 02, 10

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 13-87. Example of Contents of Registers for UART Transmission of UART (UART0 to UART2) (2/2)

(e) Serial output register m (SOm) … Sets only the bits of the target channel.

SOm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	CKOm3 Note 1 ×	CKOm2 Note 1 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	Som.3 Note 1 ×	SOm.2 Note 1 0/1 Note 2	SOm.1 ×	SOm.0 0/1 Note 2

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.

SOEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SOEm.3 Note 1 ×	SOEm.2 Note 1 0/1	SOEm.1 ×	SOEm.0 0/1

(g) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

SSm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SSm.3 Note 1 ×	SSm.2 Note 1 0/1	SSm.1 ×	SSm.0 0/1

Notes 1. Serial array unit 0 only.

2. Before transmission is started, be sure to set to 1 when the SOLm.n bit of the target channel is set to 0, and set to 0 when the SOLm.n bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2)
mn = 00, 02, 10

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

Figure 13-88. Example of Contents of Registers for UART Transmission of UART (UARTS0) (1/2)

(a) Serial output register S (SOS) … Sets only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOS	0	0	0	0	0	0	CKOS1 ×	CKOS0 ×	0	0	0	0	0	0	SOS.1 ×	SOS.0 0/1Note

(b) Serial output enable register S (SOES) … Sets only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOES.1 ×	SOES.0 0/1

(c) Serial channel start register S (SSS) … 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
SSS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSS.1 ×	SSS.0 0/1

(d) Serial output level register S (SOLS) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOLS.0 0/1

0: Forward (normal) transmission
1: Reverse transmission

(e) Serial mode register S0 (SMRS0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRS0	CKSS0 0/1	CCSS0 0	0	0	0	0	0	STSS0 0	0	SISS00 0	1	0	0	MDS02 0	MDS01 1	MDS00 0/1

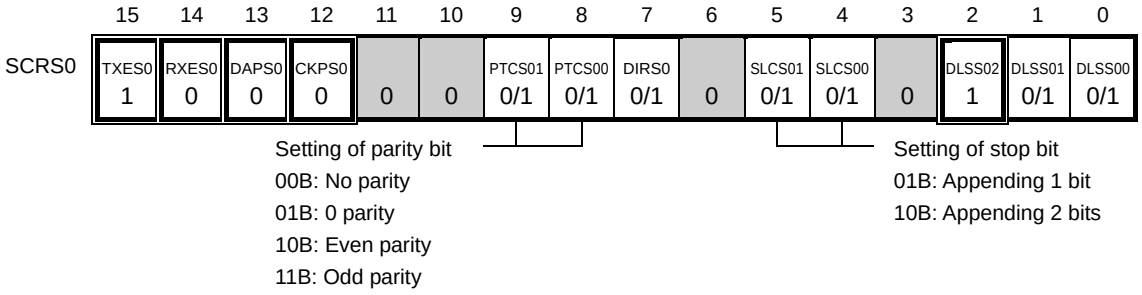
Interrupt sources of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

Note Before transmission is started, be sure to set to 1 when the SOLS.0 bit of the target channel is set to 0, and set to 0 when the SOLS.0 bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remark □: 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

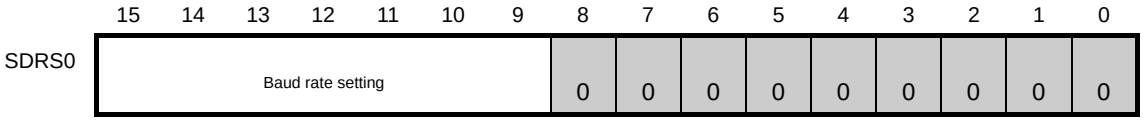
Figure 13-88. Example of Contents of Registers for UART Transmission of UART (UARTS0) (2/2)

(f) Serial communication operation setting register S0 (SCRS0)

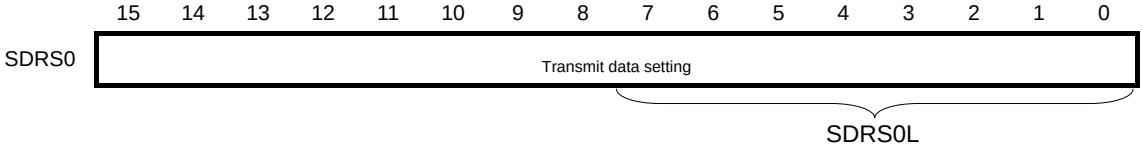


(g) Serial data register S0 (SDRS0)

(i) When operation is stopped (SES0 = 0)



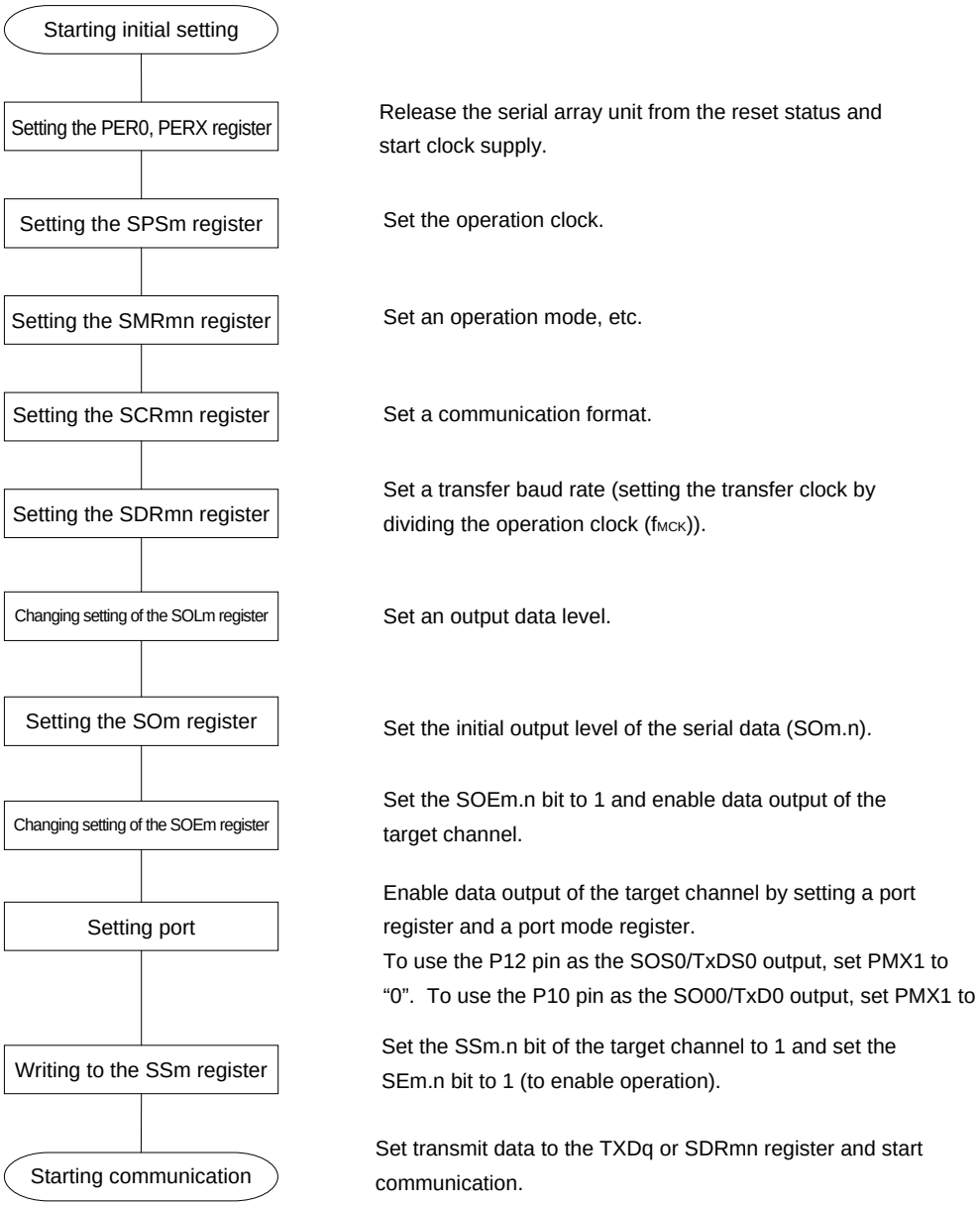
(ii) During operation (SES0 = 1) (lower 8 bits: SDRS0L)



Remark ☐: Setting is fixed in the UART transmission mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

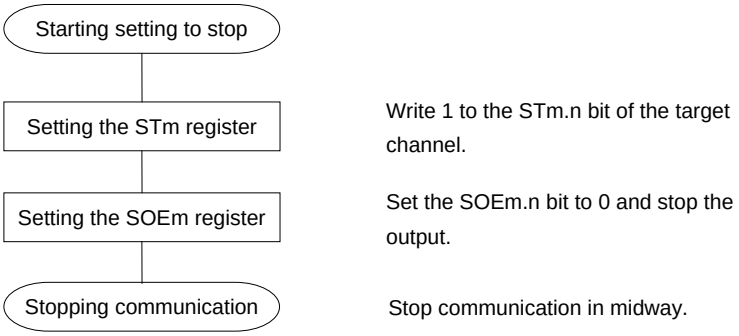
(2) Operation procedure

Figure 13-89. Initial Setting Procedure for UART Transmission



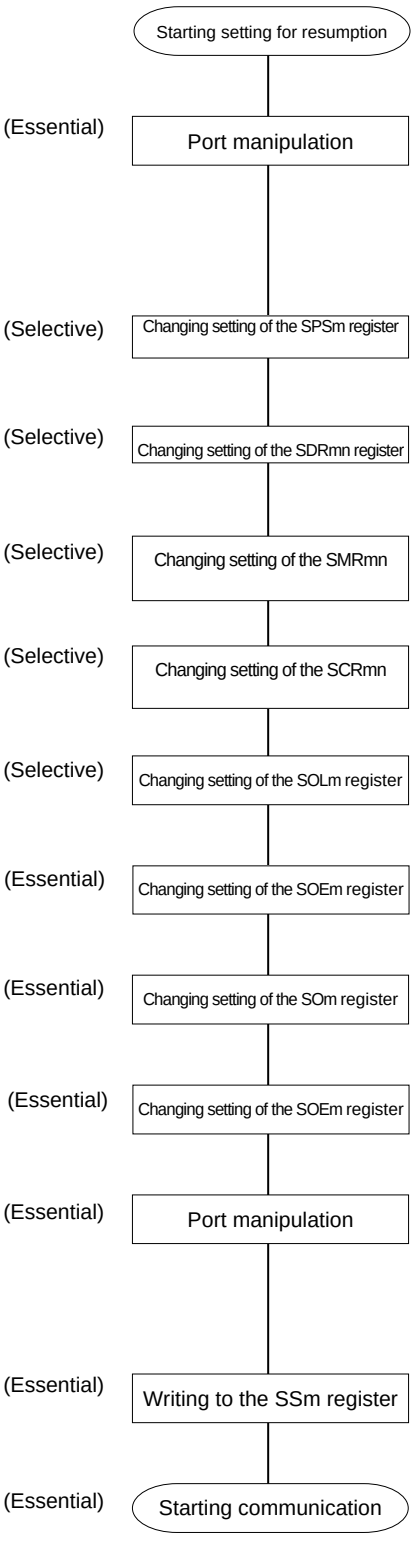
Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 13-90. Procedure for Stopping UART Transmission



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 13-91 Procedure for Resuming UART Transmission**).

Figure 13-91. Procedure for Resuming UART Transmission



Disable data output of the target channel by setting a port register and a port mode register. To use the P12 pin as the SOS0/TxDS0 output, set PMX1 to “0”. To use the P12 pin as the SO00/TxD0 output, set PMX0 to “1”.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change the serial communication operation setting register mn (SCRmn) setting.

Re-set the register to change serial output level register m (SOLm) setting.

Clear the SOEm.n bit to 0 and stop output.

Set the initial output level of the serial data (SOm.n).

Set the SOEm.n bit to 1 and enable output.

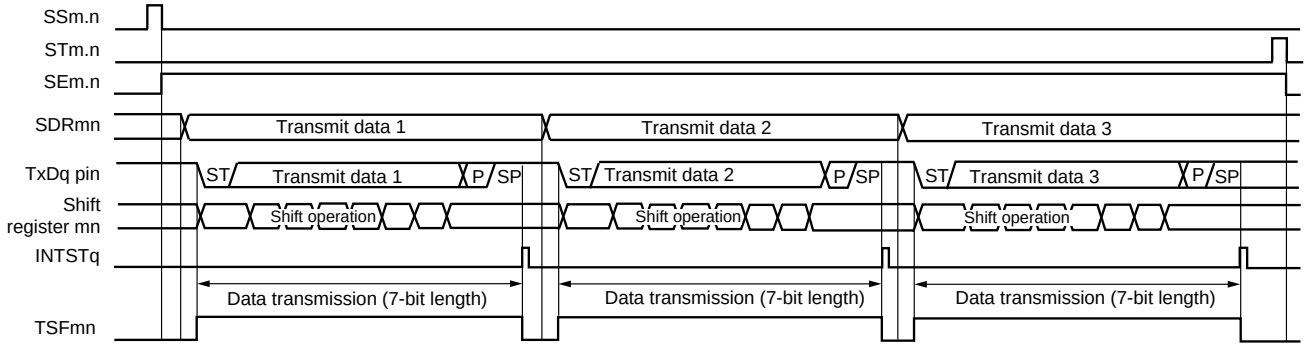
Enable data output of the target channel by setting a port register and a port mode register. To use the P12 pin as the SOS0/TxDS0 output, set PMX1 to “0”. To use the P12 pin as the SO00/TxD0 output, set PMX1 to “1”.

Set the SSm.n bit of the target channel to 1 and set the SEm.n bit to 1 (to enable operation).

Sets transmit data to the TXDq or SDRmn register and start communication.

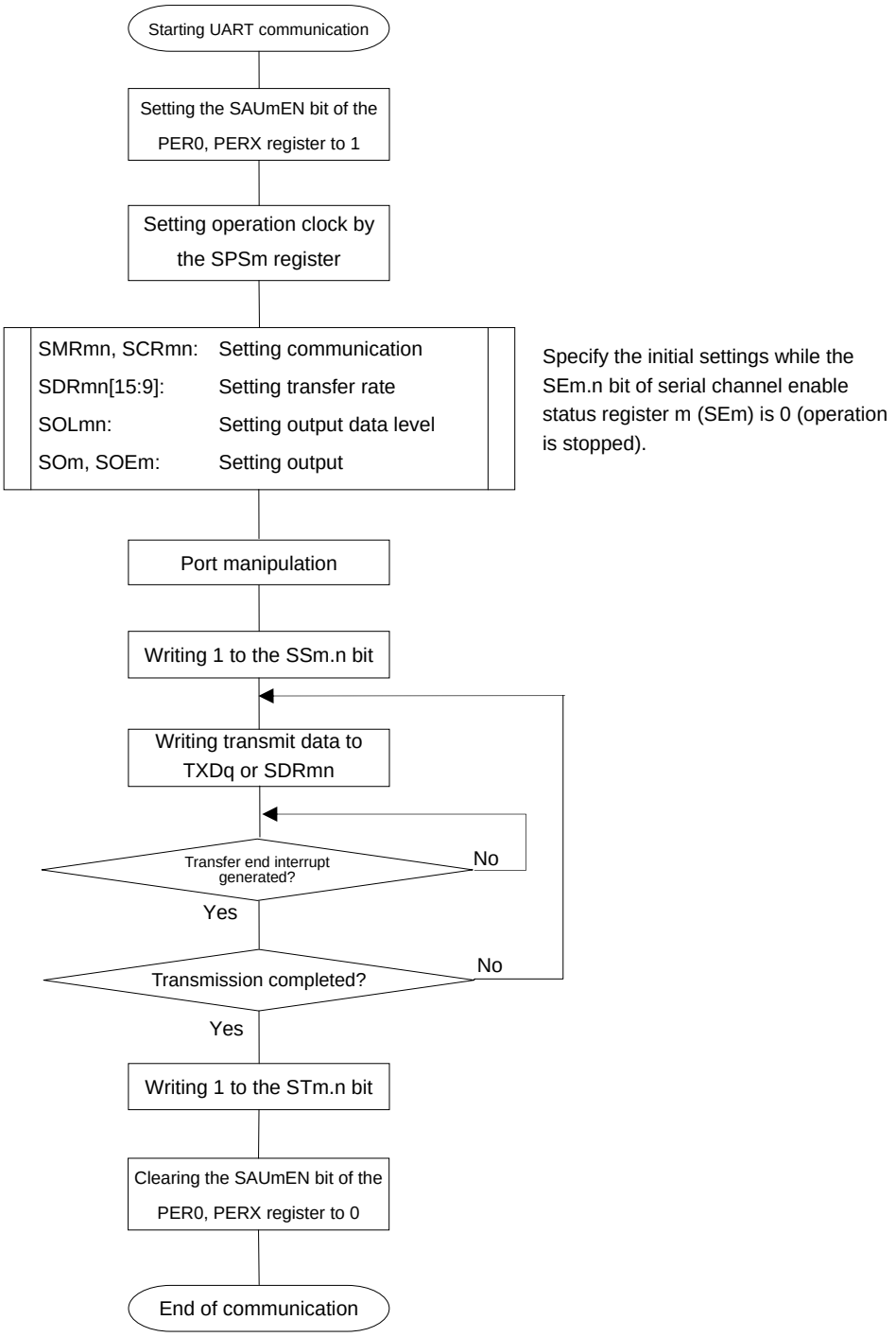
(3) Processing flow (in single-transmission mode)

Figure 13-92. Timing Chart of UART Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2, S0)
mn = 00, 02, 10, S0

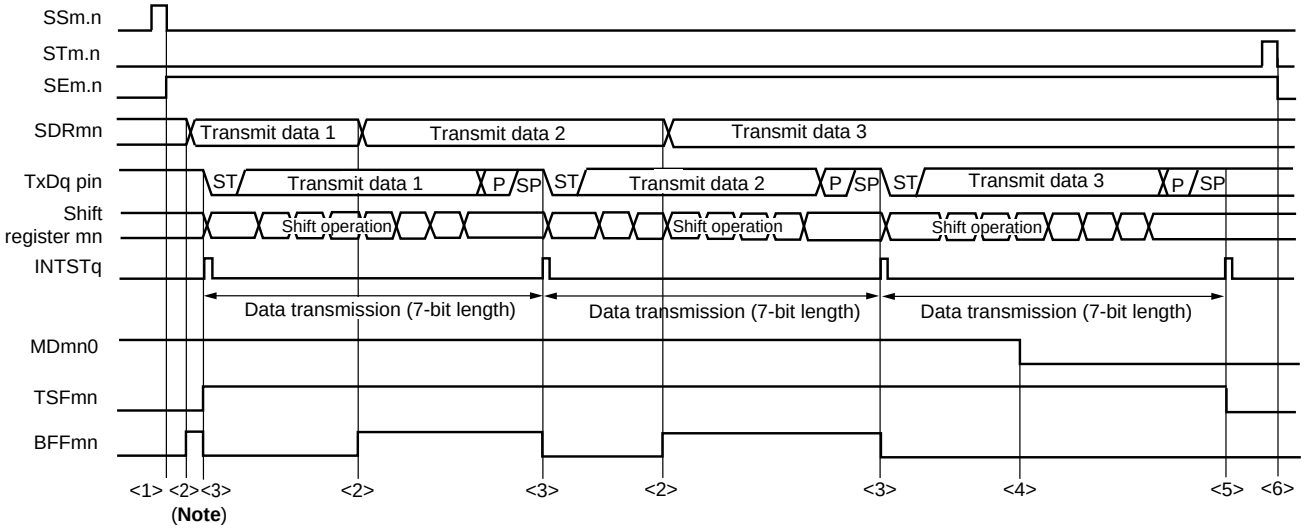
Figure 13-93. Flowchart of UART Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 13-94. Timing Chart of UART Transmission (in Continuous Transmission Mode)

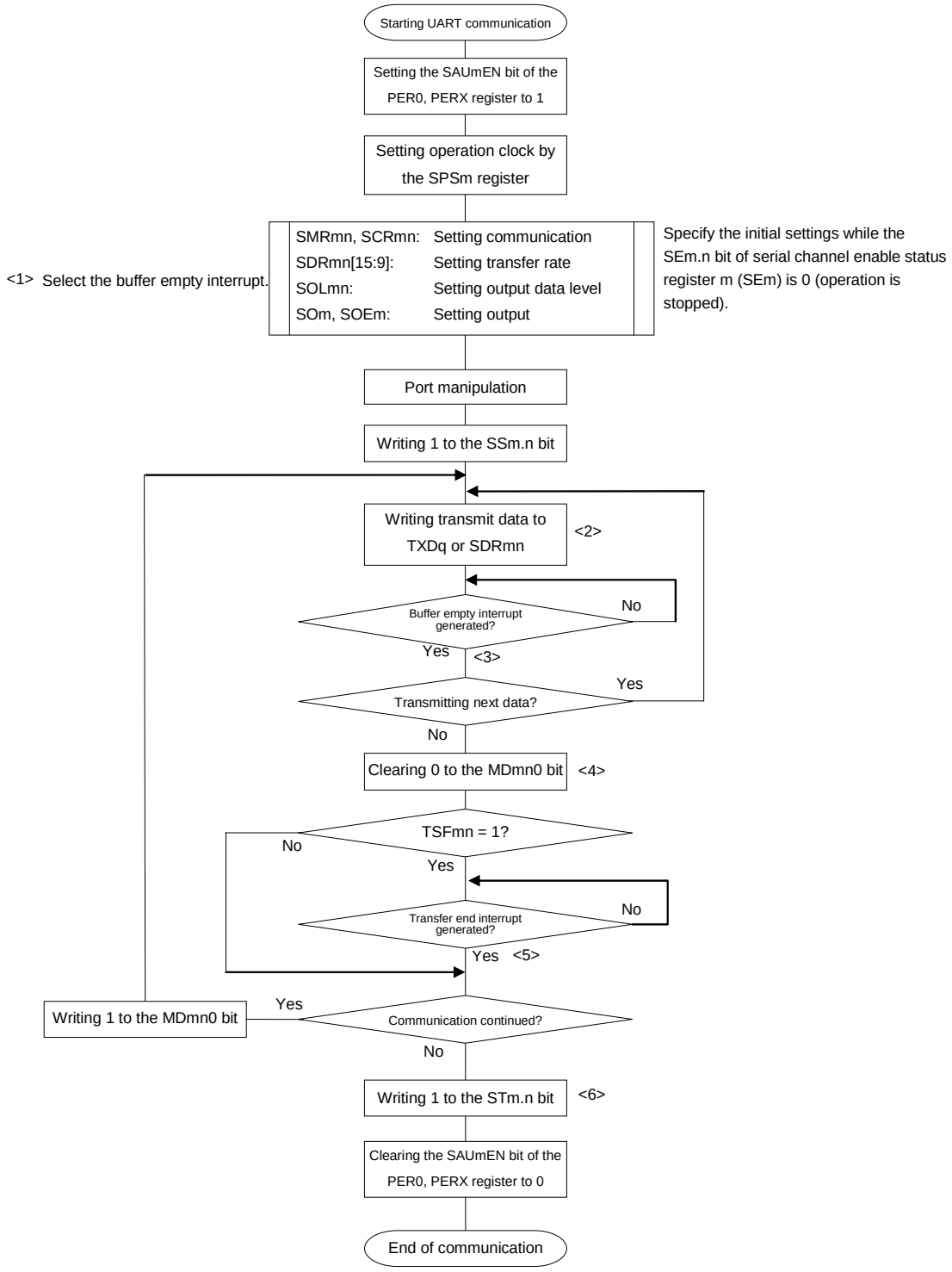


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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, S), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2, S0)
mn = 00, 02, 10, S0

Figure 13-95. Flowchart of UART Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 13-94 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

13.6.2 UART reception

UART reception is an operation wherein the RL78/F12 asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

UART	UART0	UART1	UART2	UARTS0
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1	Channel 0 of SAUS
Pins used	RxD0	RxD1	RxD2	RxDS0
Interrupt	INTSR0	INTSR1	INTSR2	INTSRS0
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	INTSRE0	INTSRE1	INTSRE2	INTSRES0
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)			
Transfer data length	7, 8 or 9 bits	7 or 8 bits		7 to 9, or 16 bits
Transfer rate	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note			Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note
Data phase	Forward 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 AC characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

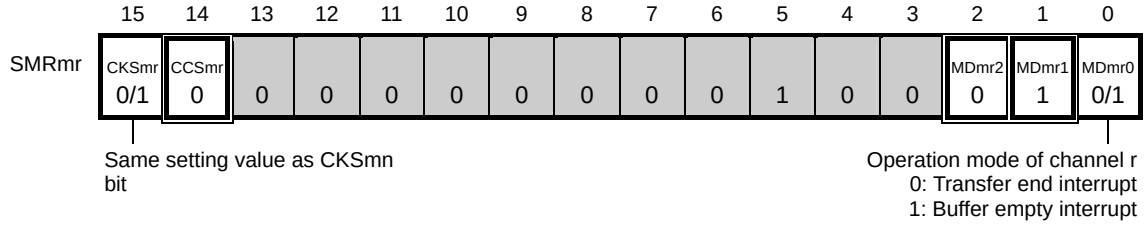
2. m: Unit number (m = 0, 1, S), n: Channel number (n = 1, 3), mn = 01, 03, 11, S1

(1) Register setting

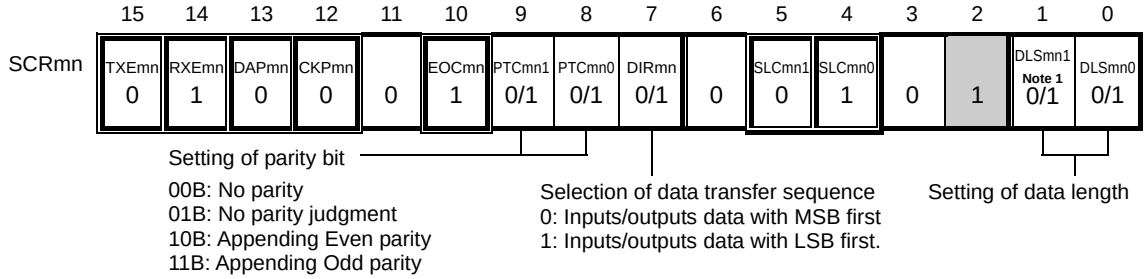
Figure 13-96. Example of Contents of Registers for UART Reception of UART (UART0 to UART2) (1/2)

(a) Serial mode register mn (SMRmn)

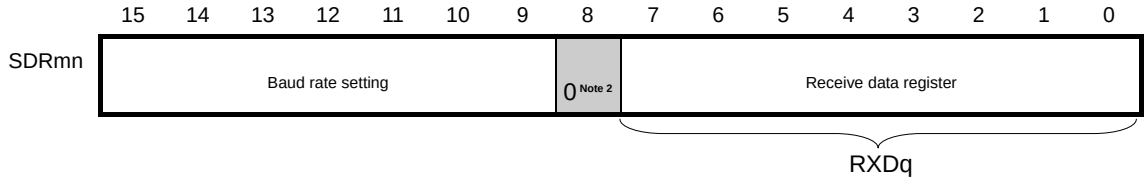
(b) Serial mode register mr (SMRmr)



(c) Serial communication operation setting register mn (SCRmn)



(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



Notes 1. The SCR01 register (UART0) only. This is fixed to 1 for the SCR03 and SCR11 registers.

2. When UART0 performs 9-bit communication (by setting the DLS011 and DLS010 bits of the SMR01 register to 1), bits 0 to 8 of the SDR01 register are used as the transmission data specification area. Only UART0 can be used to perform 9-bit communication.

Caution For the UART reception, be sure to set the SMRmr register of channel r that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11
r: Channel number (r = n – 1), q: UART number (q = 0 to 2)

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 13-96. Example of Contents of Registers for UART Reception of UART (UART0 to UART2) (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 Note ×	CKOm2 Note ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm.3 Note ×	SOm.2 Note ×	SOm.1 ×	SOm.0 ×

(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	SOEm.3 Note ×	SOEm.2 Note ×	SOEm.1 ×	SOEm.0 ×

(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	SSm.3 Note 0/1	SSm.2 Note ×	SSm.1 0/1	SSm.0 ×

Note Serial array unit 0 only.

Caution For the UART reception, be sure to set the SMRmr register of channel r that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11
r: Channel number (r = n – 1), q: UART number (q = 0 to 2)

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 13-97. Example of Contents of Registers for UART Reception of UART (UARTS0) (1/2)

(a) Serial output register S (SOS) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOS	0	0	0	0	0	0	CKOS1 ×	CKOS0 ×	0	0	0	0	0	0	SOS.1 ×	SOS.0 ×

(b) Serial output enable register S (SOES) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOES	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOES.1 ×	SOES.0 ×

(c) Serial channel start register S (SSS) … 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
SSS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSS.1 0/1	SSS.0 ×

(d) Serial mode register S1 (SMRS1)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRS1	CKSS1 0/1	CCSS1 0	0	0	0	0	0	STSS1 1	0	SISS10 0/1	1	0	0	MDS12 0	MDS11 1	MDS10 0

0: Forward (normal) reception
1: Reverse reception

Interrupt sources of channel 1
0: Transfer end interrupt

(e) Serial mode register S0 (SMRS0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRS1	CKSS1 0/1	CCSS1 0	0	0	0	0	0	STSS1 0	0	SISS10 0	1	0	0	MDS12 0	MDS11 1	MDS10 0/1

Same setting value as CKSS1

Interrupt sources of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(f) Serial communication operation setting register S1 (SCRS1)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRS1	TXES1 0	RXES1 1	DAPS1 0	CKPS1 0	0	0	PTCS11 0/1	PTCS10 0/1	DIRS1 0/1	0	SLCS11 0	SLCS10 1	0	DLSS12 1	DLSS11 0/1	DLSS10 0/1

Caution For the UART reception, be sure to set SMRS0 of channel 0 that is to be paired with channel 1.

Remark □: 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 13-97. Example of Contents of Registers for UART Reception of UART (UARTS0) (2/2)

(g) Serial data register S1 (SDRS1)

(i) When operation is stopped (SES.1 = 0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRS1	Baud rate setting							0	0	0	0	0	0	0	0	0

(ii) During operation (SES.1 = 1) (lower 8 bits: SDRS1L)

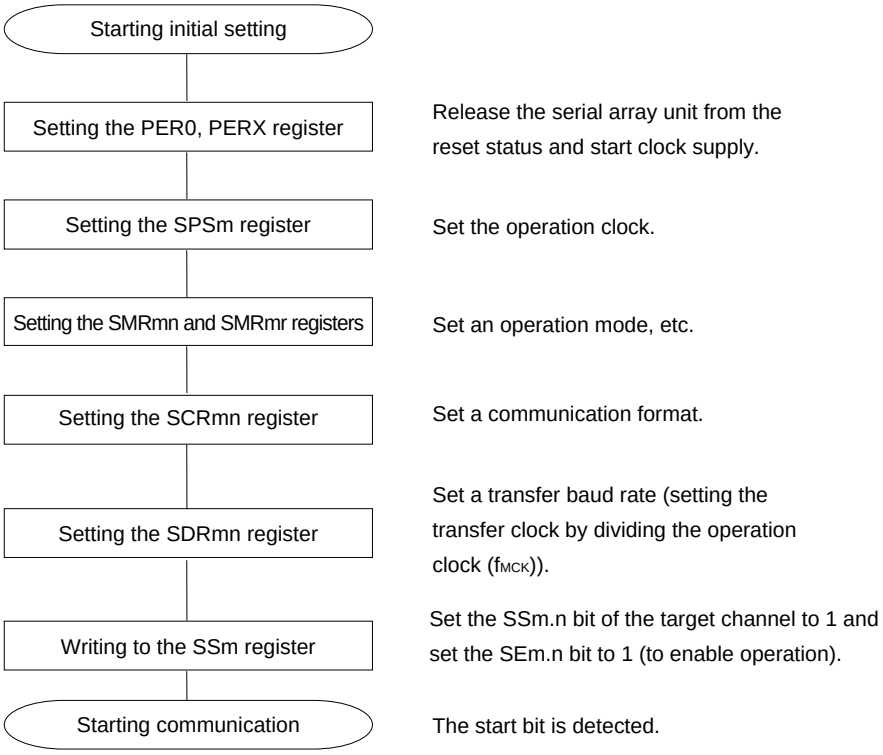
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRS1	Receive data setting															

SDRS1L

Remark □: 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

(2) Operation procedure

Figure 13-98. Initial Setting Procedure for UART Reception



Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 13-99. Procedure for Stopping UART Reception

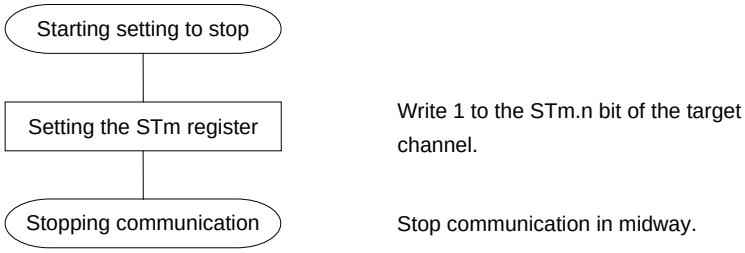
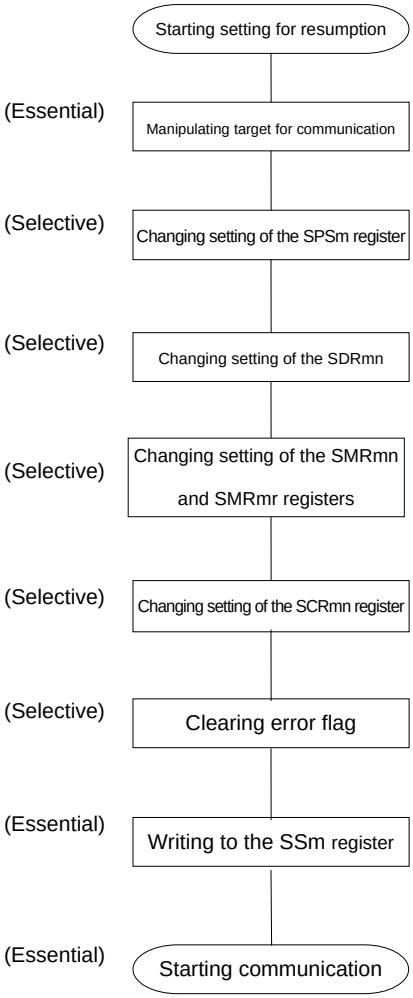


Figure 13-100. Procedure for Resuming UART Reception



Stop the target for communication or wait until the target completes its operation.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the registers to change serial mode registers mn, mr (SMRmn, SMRmr) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

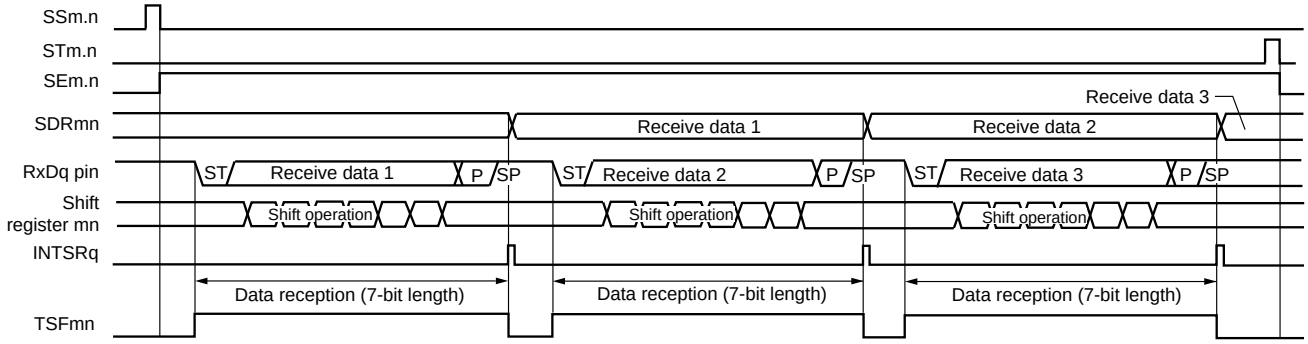
If the FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register mn (SIRmn).

Set the SSm.n bit of the target channel to 1 and set the SEm.n bit to 1 (to enable operation).

The start bit is detected.

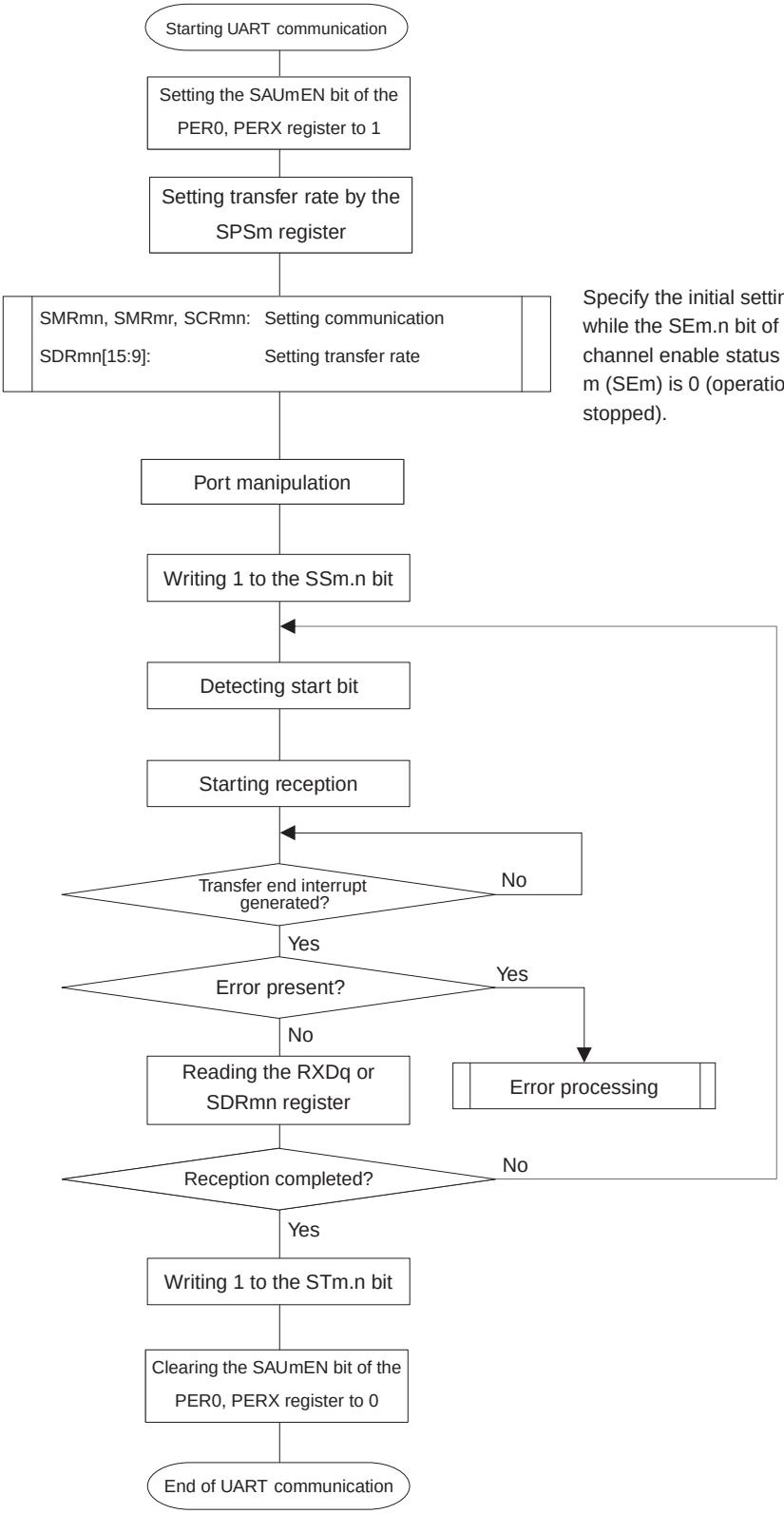
(3) Processing flow

Figure 13-101. Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 1, 3), mn = 01, 03, 11, S1
r: Channel number (r = n – 1), q: UART number (q = 0 to 2, S0)

Figure 13-102. Flowchart of UART Reception



<R>

Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

13.6.3 SNOOZE mode function (only UART0 reception)

UART0 reception (channel 1 of unit 0) supports the SNOOZE mode. When RxD0 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only UART0 can be specified for the reception baud rate adjustment function.

When using the SNOOZE mode function, set the SWC bit of serial standby control register 0 (SSC0) to 1 before switching to the STOP mode.

<R> When using UART0 in the SNOOZE mode, make the following settings before entering the STOP mode.

- In the SNOOZE mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPSm register and bits 15 to 9 of the SDRmn register with reference to Table 13-3.
- Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm1 bit of serial channel start register m (SSm) to 1.

Upon detecting the edge of RxDq (start bit input) after a transition was made to the STOP mode, UART reception is started.

Cautions 1. The SNOOZE mode can only be used when the high-speed on-chip oscillator clock (f_{IH}) is selected for f_{CLK}.

2. The transfer rate in the SNOOZE mode is only 4800 bps.

3. When SWC = 1, UART0 can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.

- **When after the SWC bit has been set to 1, the reception operation is started before the STOP mode is entered**
- **When the reception operation is started while another function is in the SNOOZE mode**
- **After returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWC bit is returned to 0**

4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

<R>

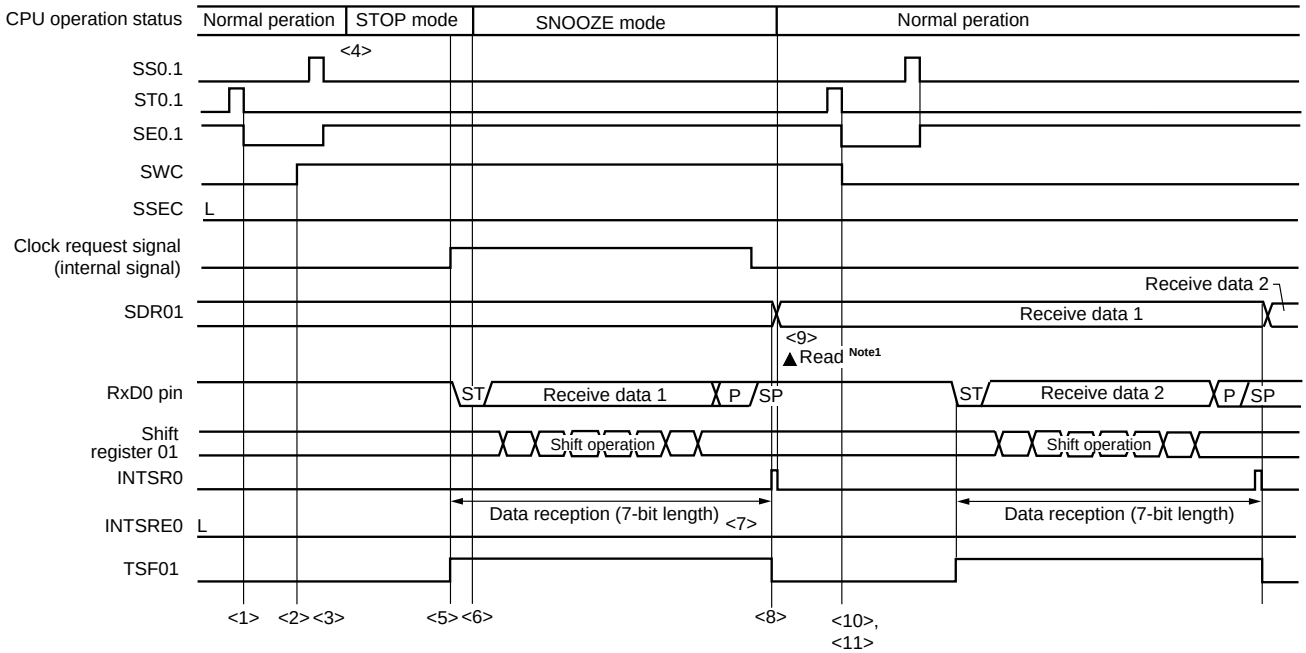
Table 13-3. Baud Rate Setting for UART Reception in SNOOZE Mode

High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Maximum Permissible Value	Minimum Permissible Value
32 MHz ± 2.0%	$f_{CLK}/2^5$	105	1.27%	–0.53%
24 MHz ± 2.0%	$f_{CLK}/2^5$	79	0.60%	–1.18%
16 MHz ± 2.0%	$f_{CLK}/2^4$	105	1.27%	–0.53%
12 MHz ± 2.0%	$f_{CLK}/2^4$	79	0.60%	–1.19%
8 MHz ± 2.0%	$f_{CLK}/2^3$	105	1.27%	–0.53%
4 MHz ± 2.0%	$f_{CLK}/2^2$	105	1.27%	–0.53%
1 MHz ± 2.0%	f_{CLK}	105	1.27%	–0.57%

Remark The maximum and minimum permissible values are values for the baud rate of UART reception. The baud rate of UART transmission should also be set within this range.

(1) SNOOZE mode operation (Normal operation)

Figure 13-103. Timing Chart of SNOOZE Mode Operation (Normal operation mode)



Note Only read received data while SWC = 1 and before the next edge of the RxD0 pin input is detected.

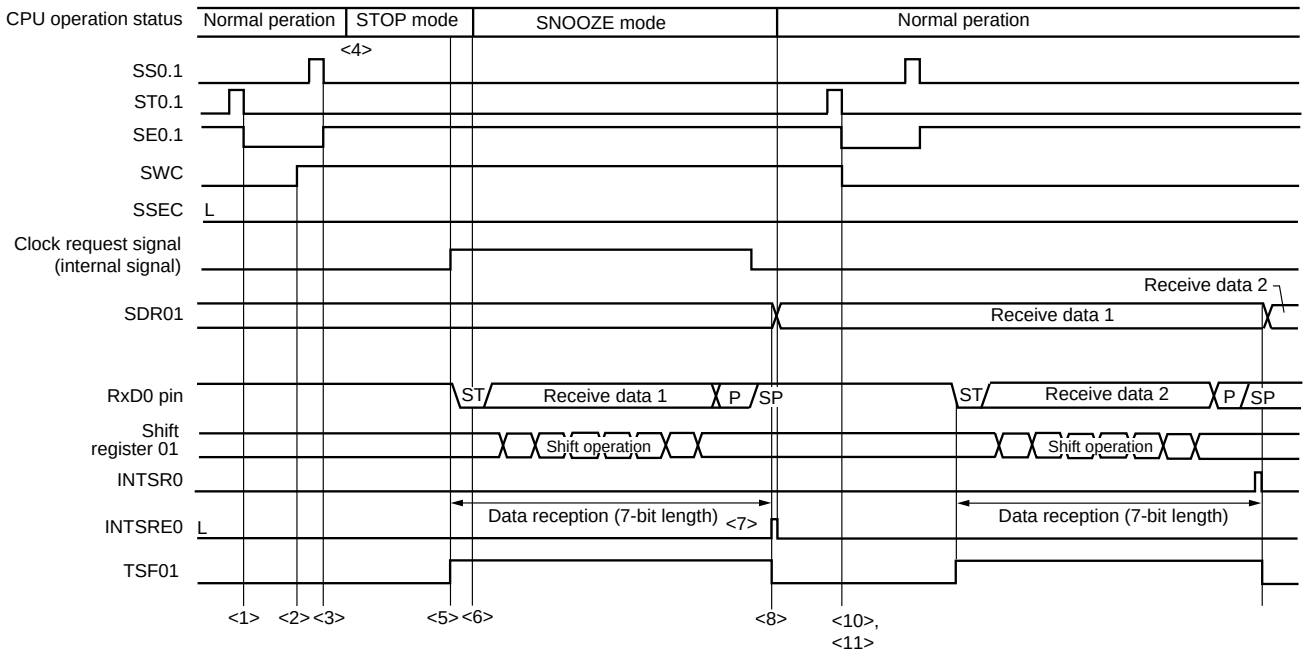
Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, be sure to set the ST0.1 bit to 1 and clear the SE0.1 bit (to stop the operation).

Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 13-105 Flowchart of SNOOZE Mode Operation (Normal Operation/Abnormal Operation <1>)**.

(2) SNOOZE mode operation (Abnormal Operation <1>)

Abnormal operation <1> is the operation performed when a communication error occurs while SSEC = 0. Because SSEC = 0, an error interrupt (INTSRE0) is generated when a communication error occurs.

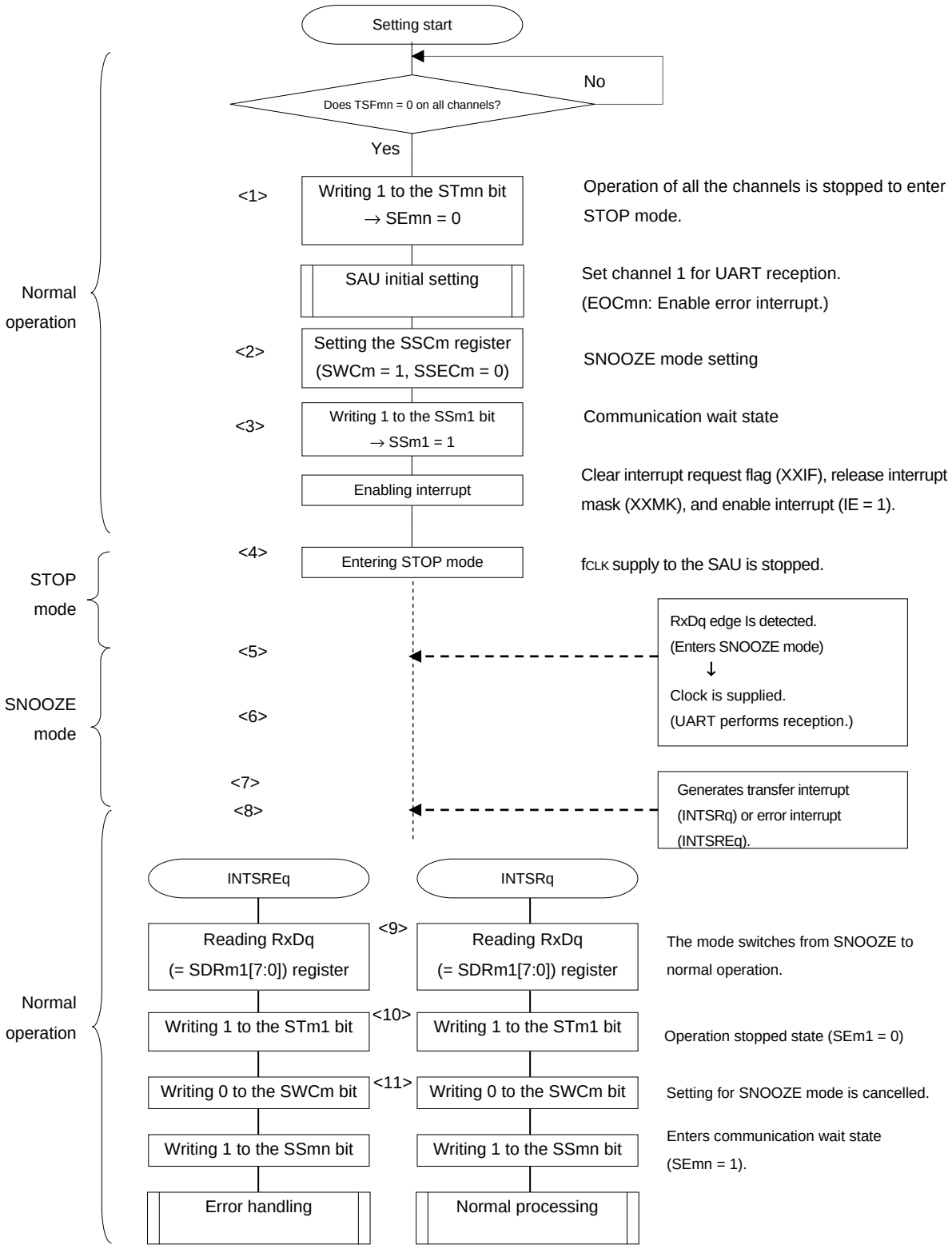
Figure 13-104. Timing Chart of SNOOZE Mode Operation (Abnormal Operation <1>)



Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, be sure to set the ST0.1 bit to 1 and clear the SE0.1 bit (to stop the operation).

Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 13-105 Flowchart of SNOOZE Mode Operation (Normal Operation/Abnormal Operation <1>)**.

Figure 13-105. Flowchart of SNOOZE Mode Operation (Normal Operation/Abnormal Operation <1>)

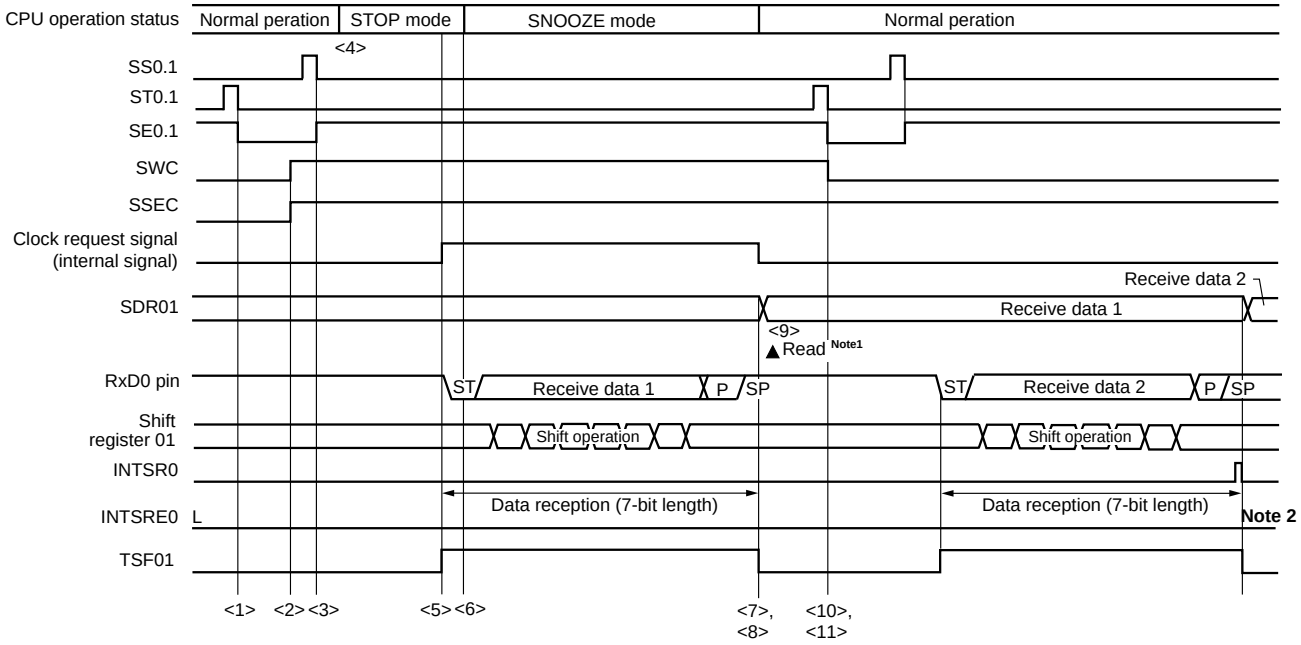


Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 13-103 Timing Chart of SNOOZE Mode Operation (Normal operation mode)** and **Figure 13-104 Timing Chart of SNOOZE Mode Operation (Abnormal Operation <1>)**.

(3) SNOOZE mode operation (Abnormal Operation <2>)

Abnormal operation <2> is the operation performed when a communication error occurs while SSEC = 1. Because SSEC = 1, an error interrupt (INTSRE0) is not generated when a communication error occurs.

Figure 13-106. Timing Chart of SNOOZE Mode Operation (Abnormal Operation <2>)



Notes 1. Only read received data while SWC = 1 and before the next edge of the RxD0 pin input is detected.

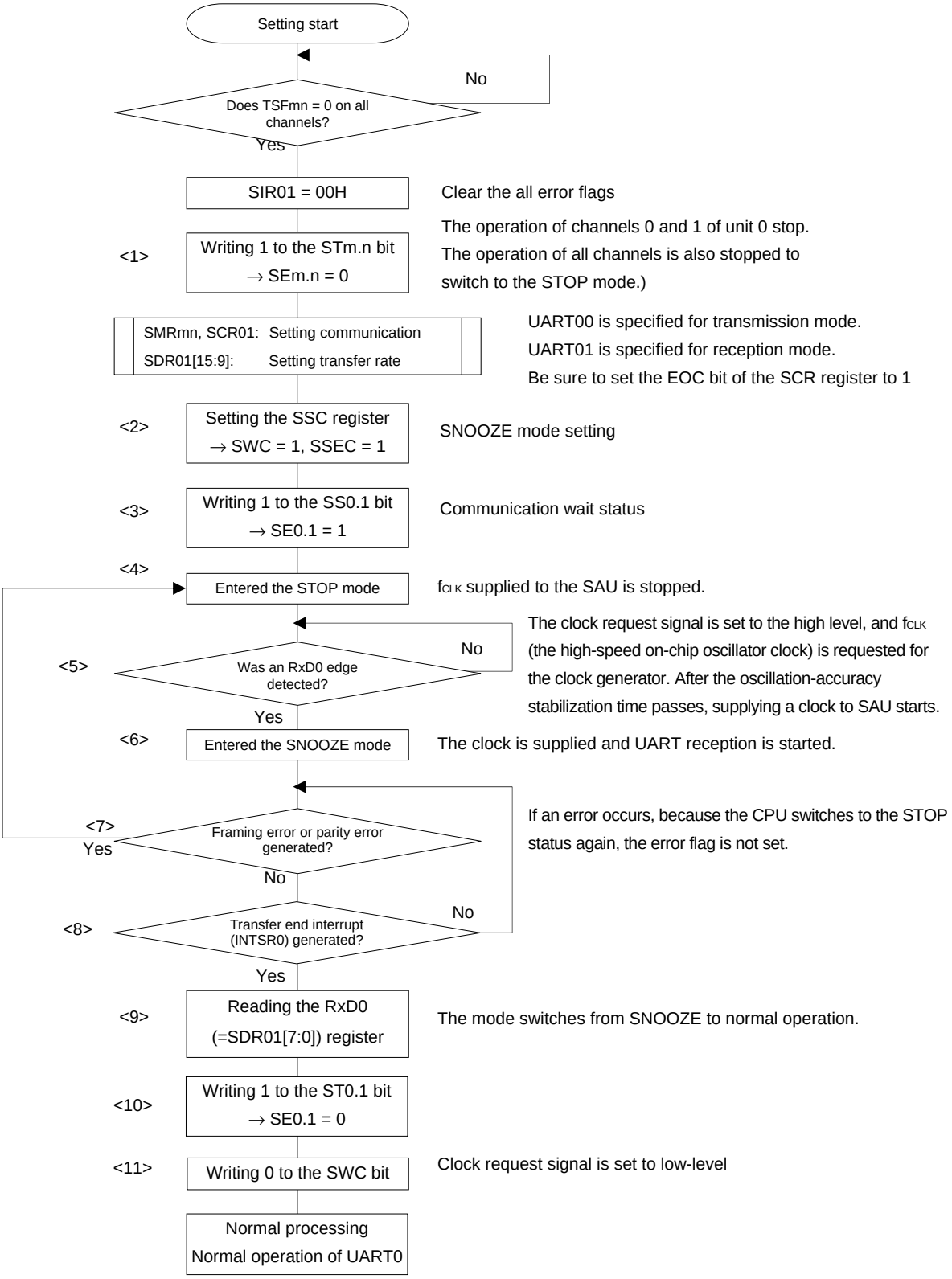
2. After UART0 successfully finishes reception in the SNOOZE mode, it is possible to continue to perform normal reception operations without changing the settings, but, because SSEC = 1, the PEF01 and FEF01 bits are not set even if a framing error or parity error occurs. In addition, no error interrupt (INTSRE0) is generated.

Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, be sure to set the ST0.1 bit to 1 and clear the SE0.1 bit (to stop the operation).

2. When using the SNOOZE mode while SSEC is set to 1, no overrun errors occur. Therefore, when using the SNOOZE mode, read bits 7 to 0 (RxD0) of the SDR01 register before switching to the STOP mode.

Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 13-107 Flowchart of SNOOZE Mode Operation (Abnormal Operation <2>)**.

Figure 13-107. Flowchart of SNOOZE Mode Operation (Abnormal Operation <2>)



Caution When using the SNOOZE mode while SSEC is set to 1, no overrun errors occur. Therefore, when using the SNOOZE mode, read bits 7 to 0 (RxD0) of the SDR01 register before switching to the STOP mode.

Remark <1> to <11> in the figure correspond to <1> to <11> in **Figure 13-106 Timing Chart of SNOOZE Mode Operation (Abnormal Operation <2>)**.

13.6.4 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0 to UART2, UARTS0) 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, 0000010B) is prohibited in UART0 to UART2.
Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited in UARTS0.

Remarks 1. When UART0, UART1, or UART2 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.
When UARTS0 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.

2. m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1

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 13-4. Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 32 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remarks 1. X: Don't care

2. m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1

(2) Baud rate error during transmission

The baud rate error of UART (UART0 to UART2, UARTS0) 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, S), n: Channel number (n = 0, 2), mn = 00, 02, 10, S0

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0 to UART2, UARTS0) 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.

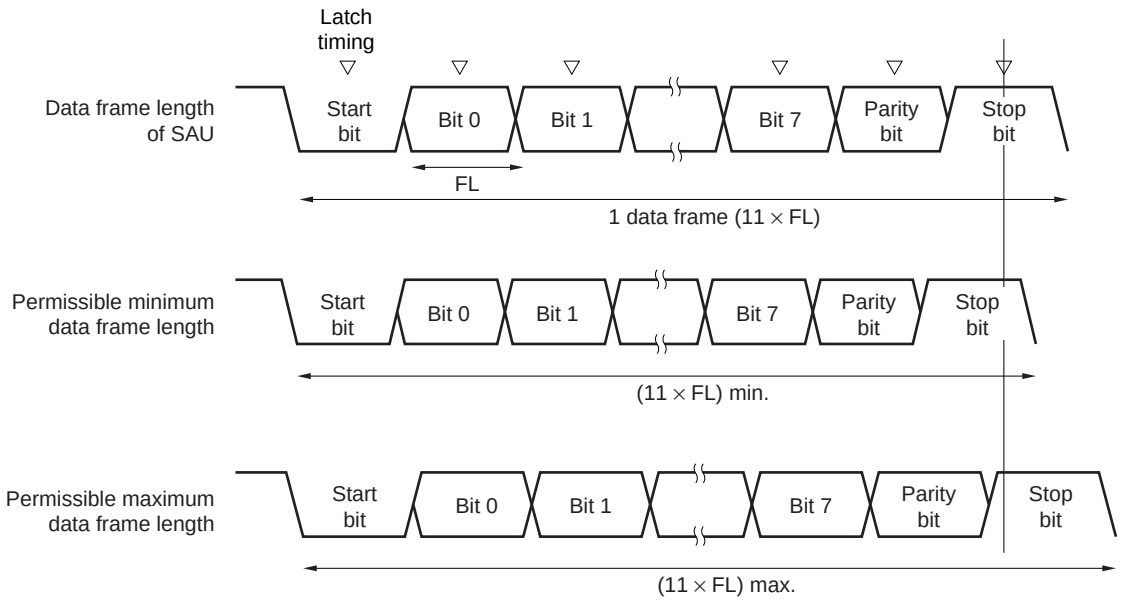
$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **13.6.4 (1) Baud rate calculation expression**.)
k: SDRmn[15:9] + 1
Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0, 1, S), n: Channel number (n = 1, 3), mn = 01, 03, 11, S1

Figure 13-108. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 13-108, 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.

13.6.5 Procedure for processing errors that occurred during UART (UART0 to UART2, UARTS0) communication

The procedure for processing errors that occurred during UART (UART0 to UART2, UARTS0) communication is described in Figures 13-109 and 13-110.

Figure 13-109. 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 13-110. 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 STm.n bit of serial channel stop register m (STm) to 1. →	The SEm.n 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 SSm.n bit of serial channel start register m (SSm) to 1. →	The SEm.n 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, S), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1

13.7 LIN Communication Operation

13.7.1 LIN transmission

Of UART transmission, UART2 of the 30, 32, 48, and 64-pin products support LIN communication.

For LIN transmission, channel 0 of unit 1 is used.

UART	UART0	UART1	UART2	UARTS0
Support of LIN communication	Not supported	Not supported	Supported	Not supported
Target channel	–	–	Channel 0 of SAU1	–
Pins used	–	–	TxD2	–
Interrupt	–	–	INTST2	–
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	8 bits			
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR10 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	The following selectable • No parity bit			
Stop bit	The following selectable • Appending 1 bit			
Data direction	LSB first			

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

Remark f_{MCK}: Operation clock frequency of target channel

f_{CLK}: System clock frequency

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network.

Communication of LIN is single-master communication and up to 15 slaves can be connected to one master.

The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN.

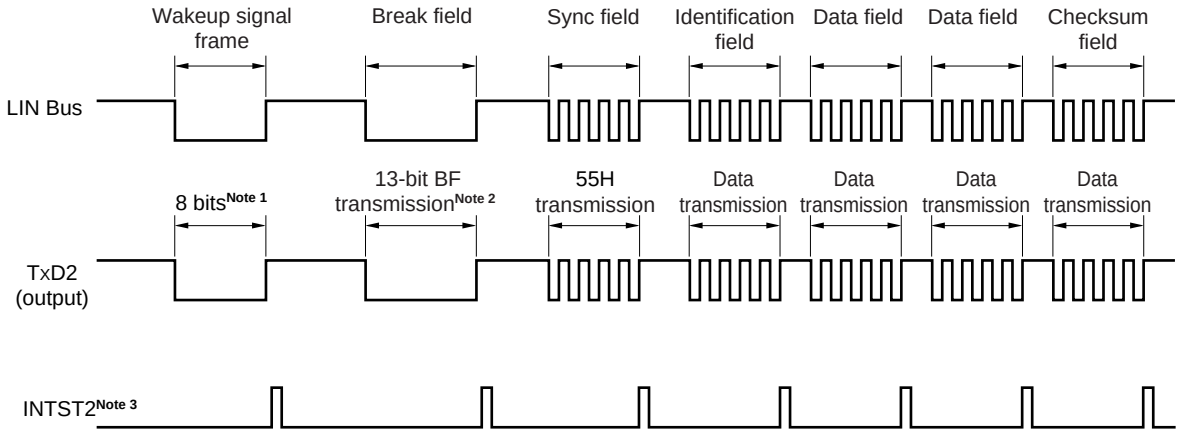
Usually, the master is connected to a network such as CAN (Controller Area Network).

A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.

According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within ±15%, communication can be established.

Figure 13-111 outlines a transmission operation of LIN.

Figure 13-111. Transmission Operation of LIN



Notes 1. The baud rate is set so as to satisfy the standard of the wakeup signal and data of 00H is transmitted.

2. A break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the break field is calculated as follows.

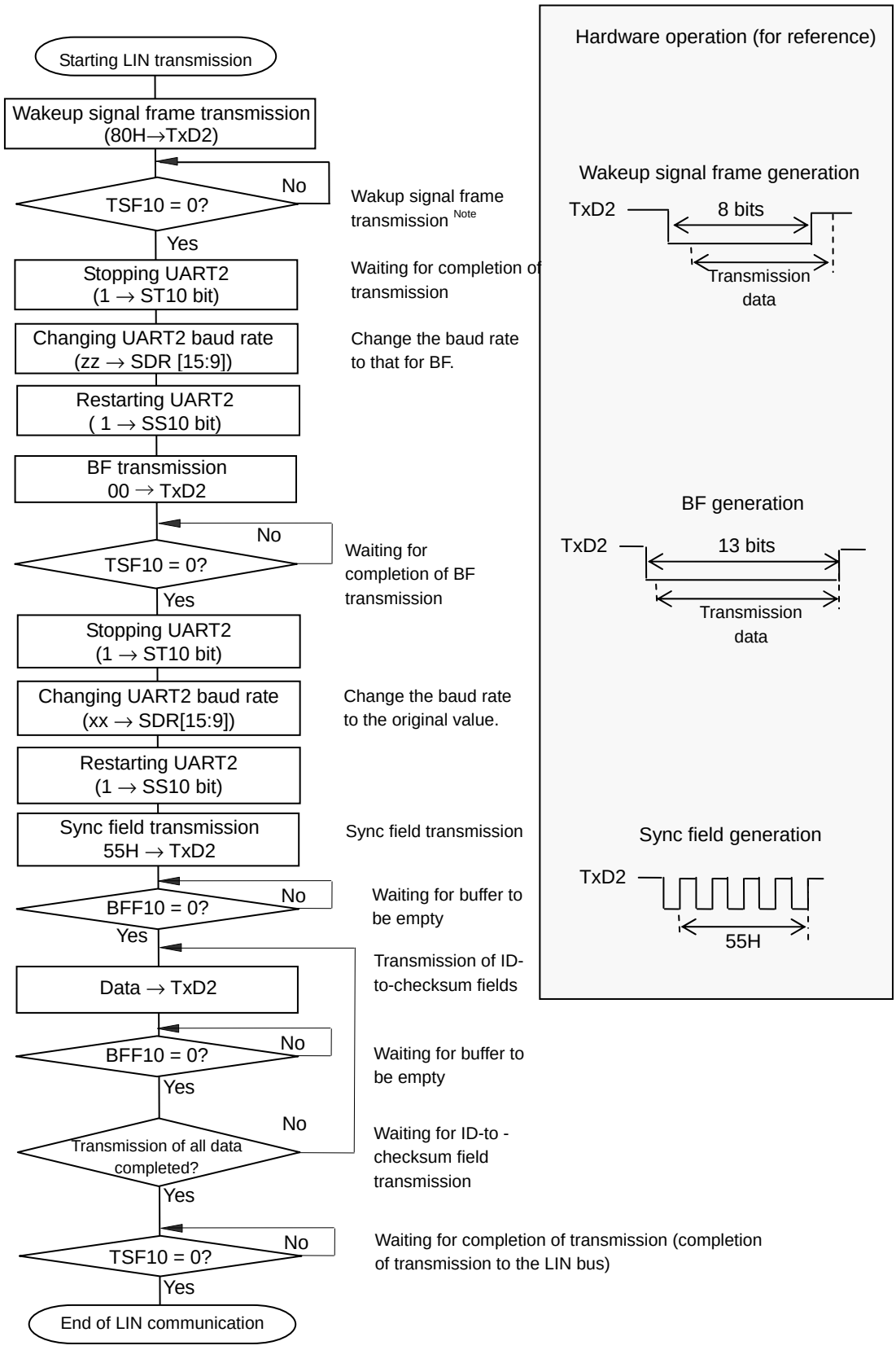
(Baud rate of break field) = 9/13 × N

By transmitting data of 00H at this baud rate, a break field is generated.

3. INTST2 is output upon completion of transmission. INTST2 is also output when BF transmission is executed.

Remark The interval between fields is controlled by software.

Figure 13-112. Flowchart for LIN Transmission



13.7.2 LIN reception

Of UART reception, UART2 of the 30, 32, 48, and 64-pin products support LIN communication.

For LIN reception, channel 1 of unit 1 is used.

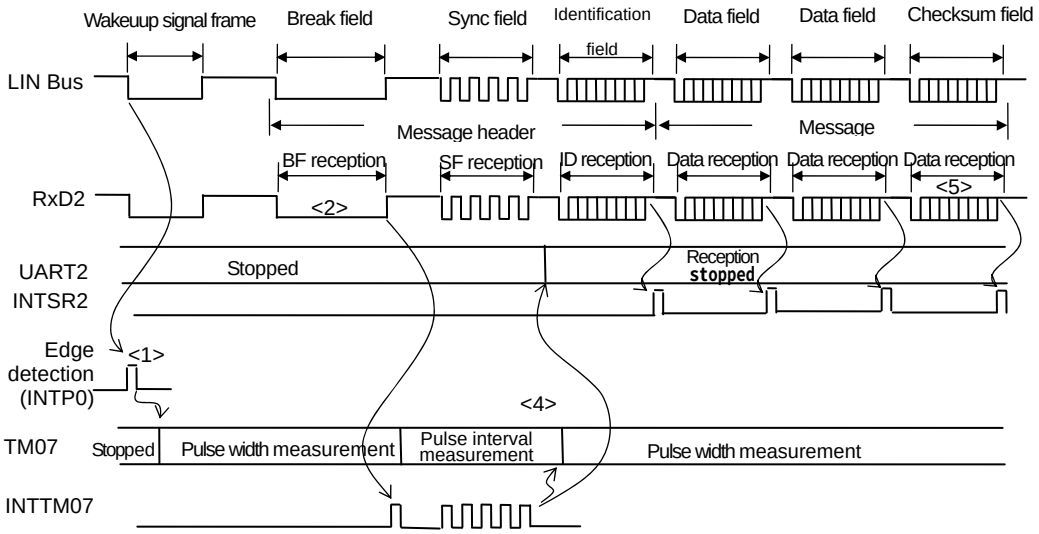
UART	UART0	UART1	UART2	UART0
Support of LIN communication	Not supported	Not supported	Supported	Not supported
Target channel	–	–	Channel 1 of SAU1	–
Pins used	–	–	RxD2	–
Interrupt	–	–	INTSR2	–
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	–	–	INTSRE2	–
Error detection flag	• Framing error detection flag (FEF11) • Overrun error detection flag (OVF11)			
Transfer data length	8 bits			
Transfer rate	Max. f_{MCK}/6 [bps] (SDR11 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	The following selectable • No parity bit (The parity bit is not checked.)			
Stop bit	The following selectable • Appending 1 bit			
Data direction	LSB first			

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)** and **CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Figure 13-113 outlines a reception operation of LIN.

Figure 13-113. Reception Operation of LIN



Here is the flow of signal processing.

<1> The wakeup signal is detected by detecting an interrupt edge (INTP0) on a pin. When the wakeup signal is detected, set TM07 to pulse-width measurement mode to measure low-level width and wait for BF reception.

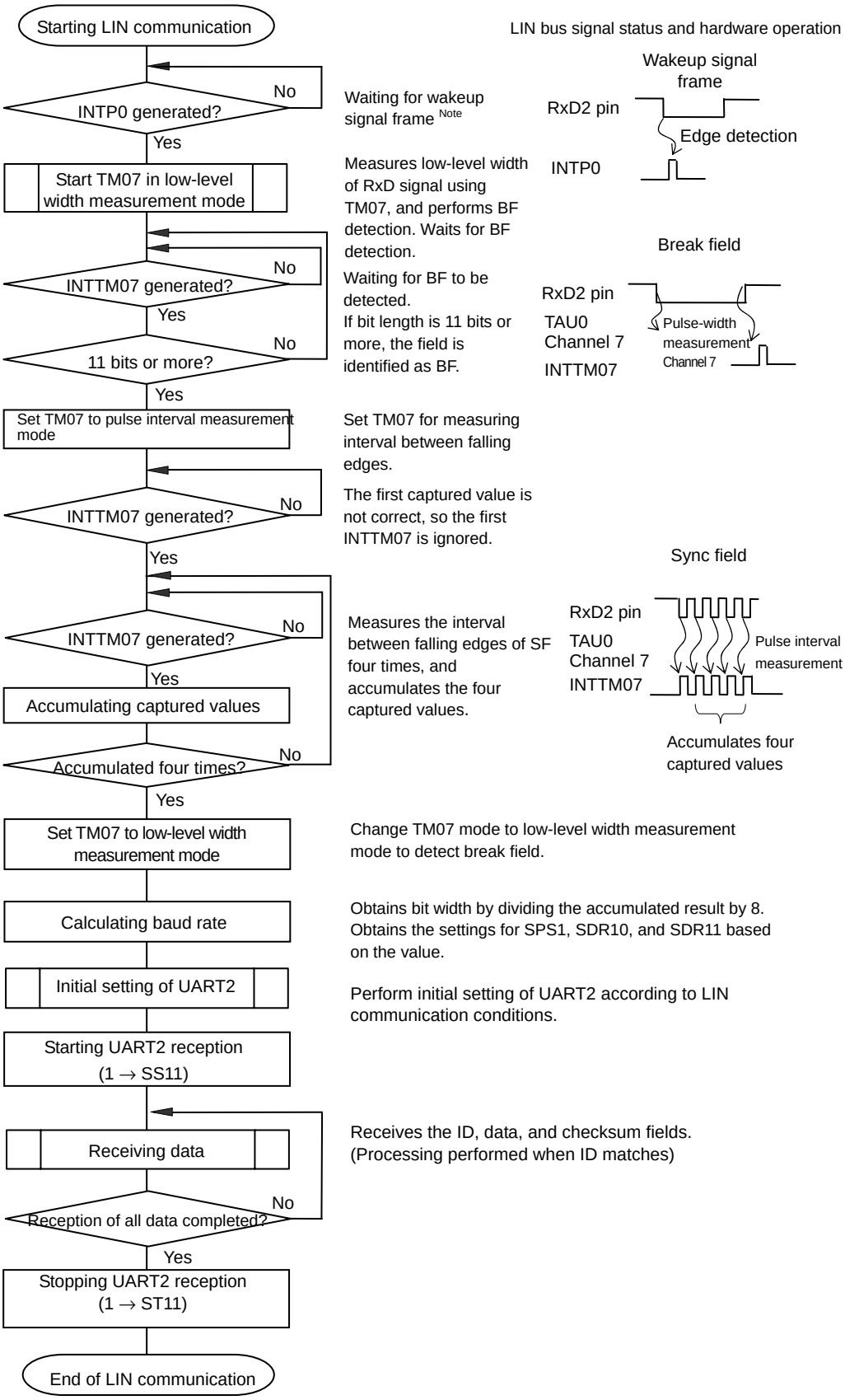
<2> When the falling edge of BF is detected, TM07 starts measuring low-level width and performs capture operation on the rising edge of BF. Based on the captured value, it is determined whether the signal is the BF signal or not.

<3> When BF reception has been correctly completed, set TM07 to pulse interval measurement mode and measure the interval between the falling edges of the RxD2 signal in the sync field four times. (see **6.7.4 Operation as input pulse interval measurement**).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART2 once and adjust (re-set) the baud rate.

<5> The checksum field should be distinguished by software. In addition, processing to initialize UART2 after the checksum field is received and to wait for reception of BF should also be performed by software.

Figure 13-114. Flowchart of LIN Reception



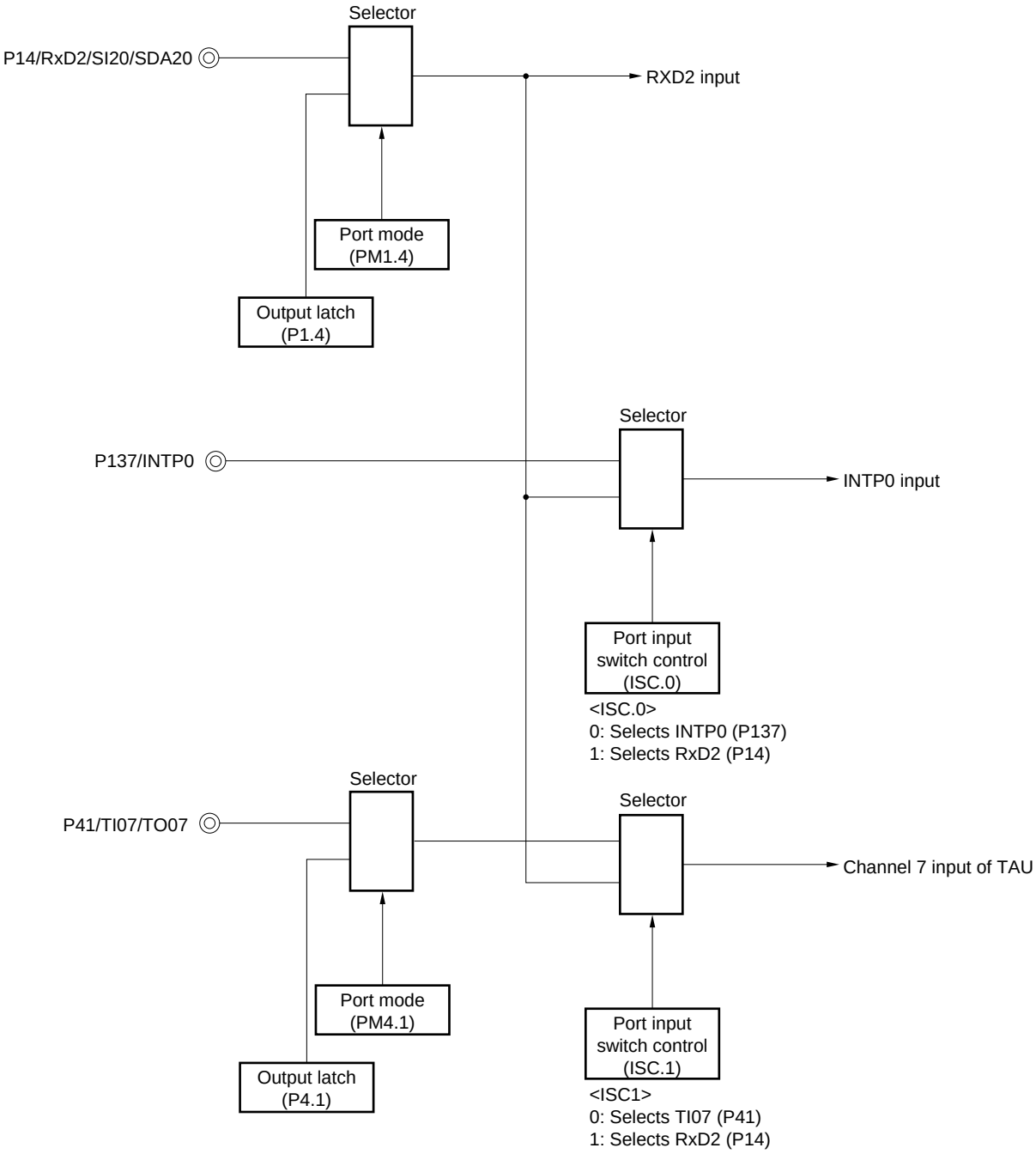
Note Necessary only in sleep mode.

Figures 13-115 and figure 13-116 show the configuration of a port that manipulates reception of LIN.

The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit 0 to calculate a baud-rate error.

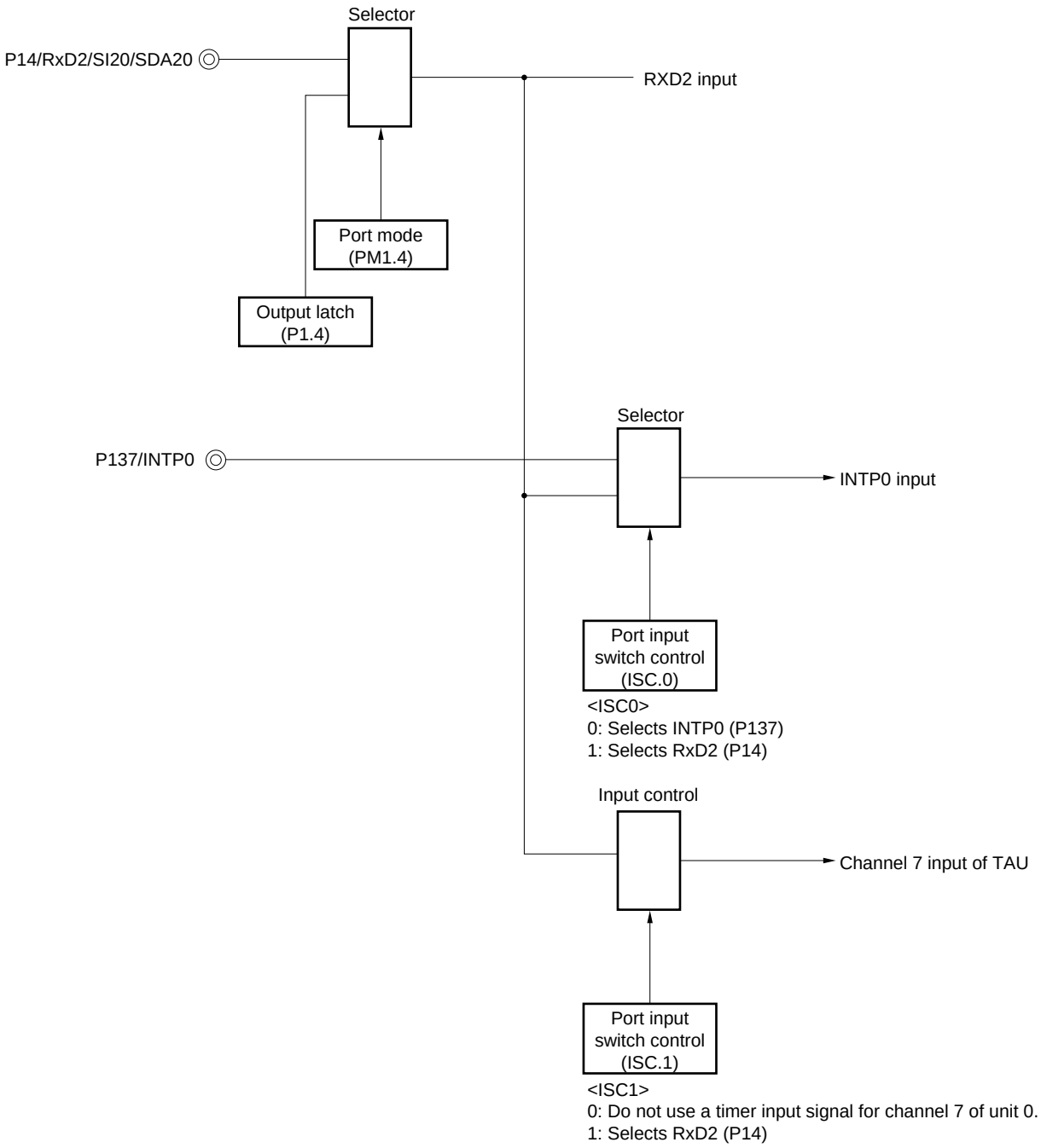
By controlling switch of port input (ISC.0/ISC.1), the input source of port input (RxD2) for reception can be input to the external interrupt pin (INTP0) and timer array unit

Figure 13-115. Port Configuration for Manipulating Reception of LIN (30, 32-pin)



Remark ISC.0, ISC.1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 13-21**.)

Figure 13-116. Port Configuration for Manipulating Reception of LIN (48, 64-pin)



Remark ISC.0, ISC.1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 13-21**.)

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); Wakeup signal detection
  Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 7 of timer array unit; Baud rate error detection, break field (BF) detection
  Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect baud rate error
  (The interval of the edge input to RxD2 is measured in the capture mode.)
  To measure low-level width and determine whether it is break field (BF) or not.
- Channels 0 and 1 (UART2) of serial array unit 1 (SAU1)

13.8 Operation of Simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.

To generate the start and stop conditions, manipulate the control registers through software, considering the IIC bus line characteristics.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note] and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Generation of start condition and stop condition by software

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error detection flag

* [Functions not supported by simplified I^2C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection function

Note When receiving the last data, ACK will not be output if 0 is written to the SOEm.n (SOEm register) bit and serial communication data output is stopped. See the processing flow in **13.8.3 (2)** for details.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

The channel supporting simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) is channels 0 to 3 of SAU0 and channel 0 and 1 of SAU1.

- 20, 24, and 25-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	–		–
	2	–	–	–
	3	–		–
1	0	–	–	–
	1	–		–
S	0	CSIS0	UARTS0	–
	1	–		–

- 30 and 32-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	–		–
	2	–	UART1	–
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	–		–
S	0	CSI20	UARTS0	–
	1	–		–

- 48-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	–	UART1	–
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
S	0	CSIS0	UARTS0	–
	1	–		–

- 64-pin products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
S	0	CSIS0	UARTS0	–
	1	CSIS1		–

Simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) performs the following four types of communication operations.

- Address field transmission (See **13.8.1**.)
- Data transmission (See **13.8.2**.)
- Data reception (See **13.8.3**.)
- Stop condition generation (See **13.8.4**.)

13.8.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^2C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I^2C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00 Note	SCL01, SDA01 Note	SCL10, SDA10 Note	SCL11, SDA11 Note	SCL20, SDA20 Note	SCL21, SDA21 Note
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)					
Error detection flag	ACK error detection flag (PEFmn)					
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)					
Transfer rate	Max. $f_{MCK}/2$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)					
Data level	Non-reverse output (default: high level)					
Parity bit	No parity bit					
Stop bit	Appending 1 bit (for ACK reception timing)					
Data direction	MSB first					

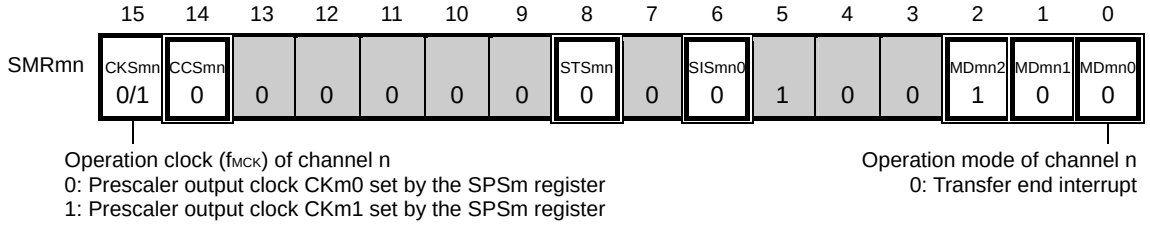
Note To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POM0.3, POM1.1, POM1.3, POM1.4, POM5.0, POM7.1, POM7.4 = 1) for the port output mode registers (POM0, POM1, POM5, POM7) (see **4.3 Registers Controlling Port Function** for details).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

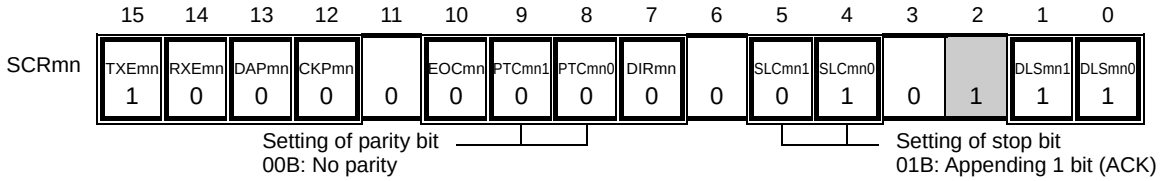
(1) Register setting

Figure 13-117. Example of Contents of Registers for Address Field Transmission of Simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21)(1/2)

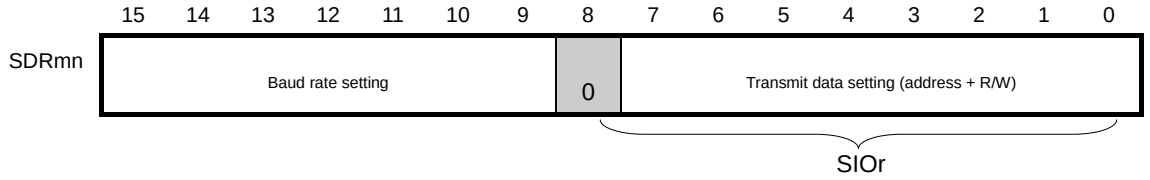
(a) Serial mode register mn (SMRmn)



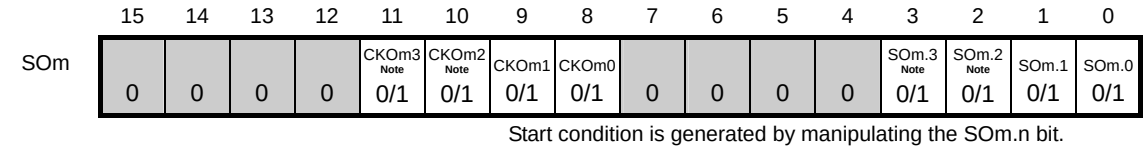
(b) Serial communication operation setting register mn (SCRmn)



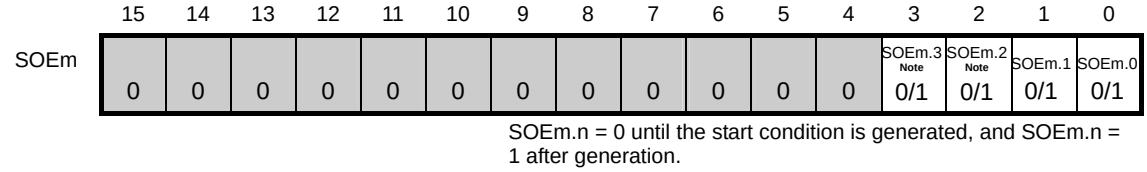
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)



(d) Serial output register m (SOm)



(e) Serial output enable register m (SOEm)



Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21) mn = 00 to 03, 10, 11

2. ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-117. Example of Contents of Registers for Address Field Transmission of Simplified I^2C (IIC00, IIC01, IIC11, IIC20, IIC21)(2/2)

(f) 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	SSm.3 Note 0/1	SSm.2 Note 0/1	SSm.1 0/1	SSm.0 0/1

SSEm.n = 0 until the start condition is generated, and SSEm.n = 1 after generation.

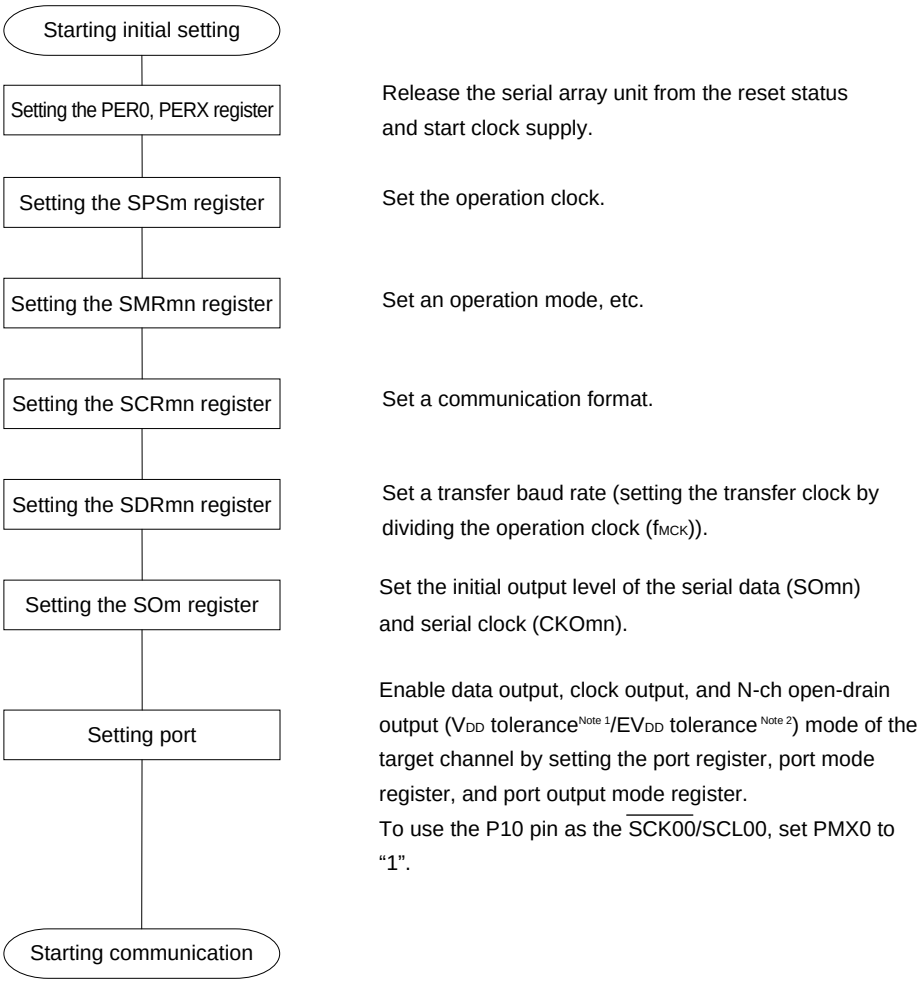
Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

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 13-118. Initial Setting Procedure for Address Field Transmission



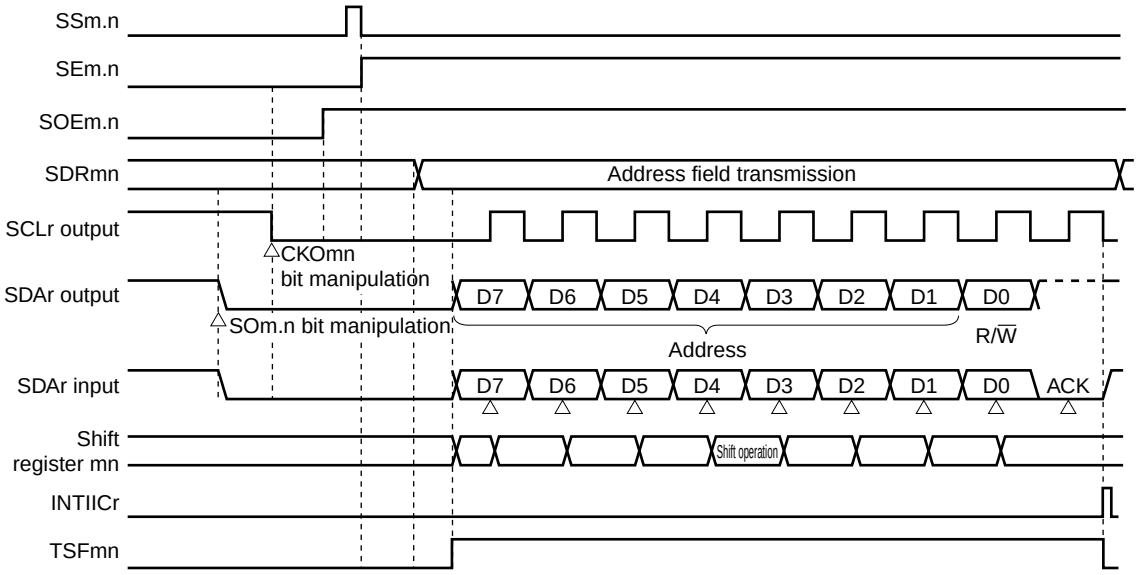
Notes 1. 20- to 48-pin products

2. 64-pin products

Caution After setting the SAUmEN bit of peripheral enable register 0, X (PER0, PERX) to 1, be sure to set serial clock select register m (SPSm) after 4 or more fCLK clocks have elapsed.

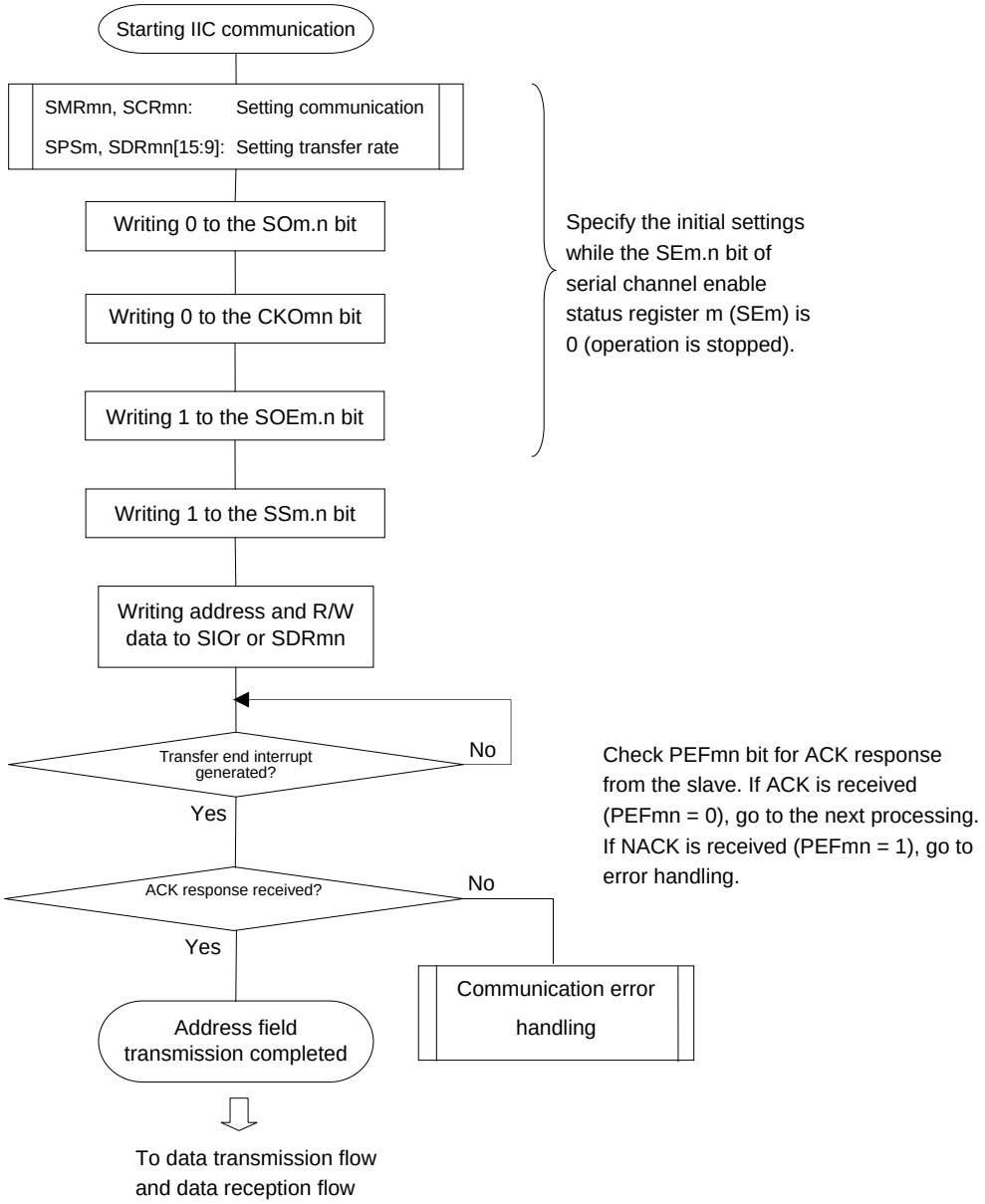
(3) Processing flow

Figure 13-119. Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

Figure 13-120. Flowchart of Address Field Transmission



13.8.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00 Note	SCL01, SDA01 Note	SCL10, SDA10 Note	SCL11, SDA11 Note	SCL20, SDA20 Note	SCL21, SDA21 Note
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)					
Error detection flag	ACK error detection flag (PEFmn)					
Transfer data length	8 bits					
Transfer rate	Max. $f_{MCK}/2$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)					
Data level	Non-reverse output (default: high level)					
Parity bit	No parity bit					
Stop bit	Appending 1 bit (for ACK reception timing)					
Data direction	MSB first					

Note To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POM0.3, POM1.1, POM1.4, POM5.0, POM7.1, POM7.4 = 1) for the port output mode registers (POM0, POM1, POM5, POM7) (see **4.3 Registers Controlling Port Function** for details).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

(1) Register setting

Figure 13-121. Example of Contents of Registers for Data Transmission of Simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) (1/2)

(a) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(b) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting							

SIOr

(d) Serial output register m (SOm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3 Note 1	CKOm2 Note 1	CKOm1	CKOm0					SOm.3 Note 1	SOm.2 Note 1	SOm.1	SOm.0
	0	0	0	0	0/1 Note 2	0/1 Note 2	0/1 Note 2	0/1 Note 2	0	0	0	0	0/1 Note 2	0/1 Note 2	0/1 Note 2	0/1 Note 2

(e) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm.3 Note 1	SOEm.2 Note 1	SOEm.1	SOEm.0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Notes 1. Serial array unit 0 only.

2. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21) mn = 00 to 03, 10, 11

2. ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-121. Example of Contents of Registers for Data Transmission of Simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) (2/2)

(f) Serial channel start register m (SSm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm.3 Note 0/1	SSm.2 Note 0/1	SSm.1 0/1	SSm.0 0/1

Note Serial array unit 0 only.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

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) Processing flow

Figure 13-122. Timing Chart of Data Transmission

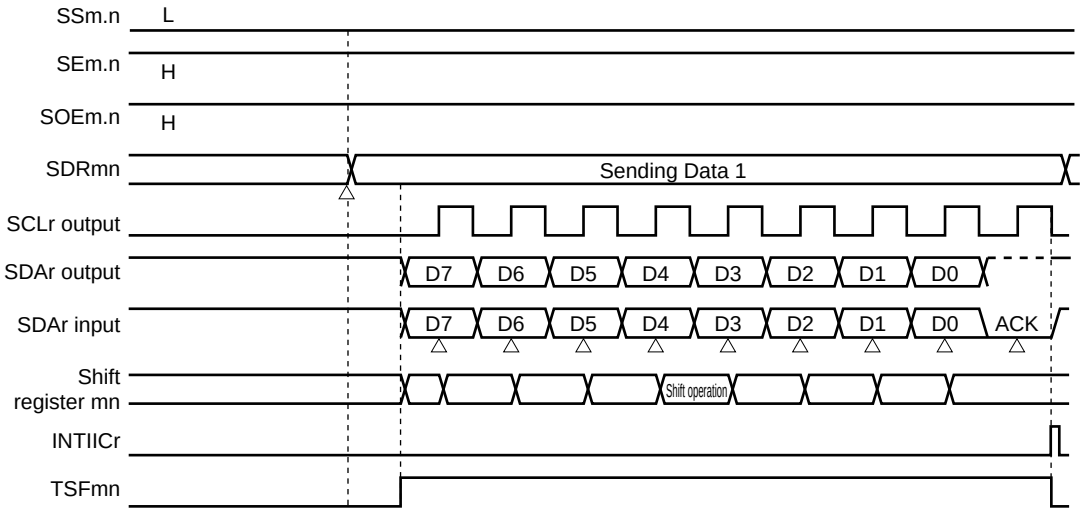
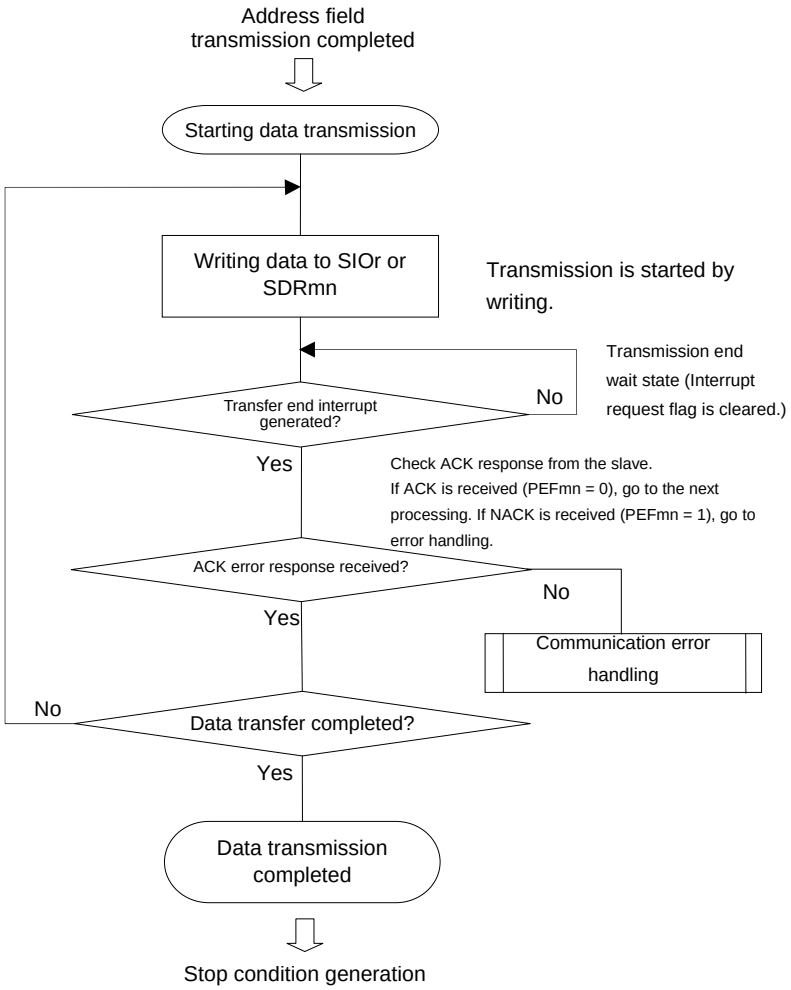


Figure 13-123. Flowchart of Data Transmission



13.8.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00 Note	SCL01, SDA01 Note	SCL10, SDA10 Note	SCL11, SDA11 Note	SCL20, SDA20 Note	SCL21, SDA21 Note
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)					
Error detection flag	Overrun error detection flag (OVFmn) only					
Transfer data length	8 bits					
Transfer rate	Max. f_{MCK}/2 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)					
Data level	Non-reverse output (default: high level)					
Parity bit	No parity bit					
Stop bit	Appending 1 bit (ACK transmission)					
Data direction	MSB first					

Note To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POM0.3, POM1.1, POM1.4, POM5.0, POM7.1, POM7.4 = 1) for the port output mode registers (POM0, POM1, POM5, POM7) (see **4.3 Registers Controlling Port Function** for details).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

(1) Register setting

Figure 13-124. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) (1/2)

(a) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(b) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Dummy transmit data setting (FFH)							

SIOr

(d) Serial output register m (SOm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3 Note 1	CKOm2 Note 1	CKOm1	CKOm0					SOm.3 Note 1	SOm.2 Note 1	SOm.1	SOm.0
	0	0	0	0	0/1 Note 2	0/1 Note 2	0/1 Note 2	0/1 Note 2	0	0	0	0	0/1 Note 2	0/1 Note 2	0/1 Note 2	0/1 Note 2

(e) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm.3 Note 1	SOEm.2 Note 1	SOEm.1	SOEm.0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

Notes 1. Serial array unit 0 only.

2. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21) mn = 00 to 03, 10, 11

2. ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-124. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) (2/2)

(f) Serial channel start register m (SSm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm.3 Note 0/1	SSm.2 Note 0/1	SSm.1 0/1	SSm.0 0/1

Note Serial array unit 0 only.

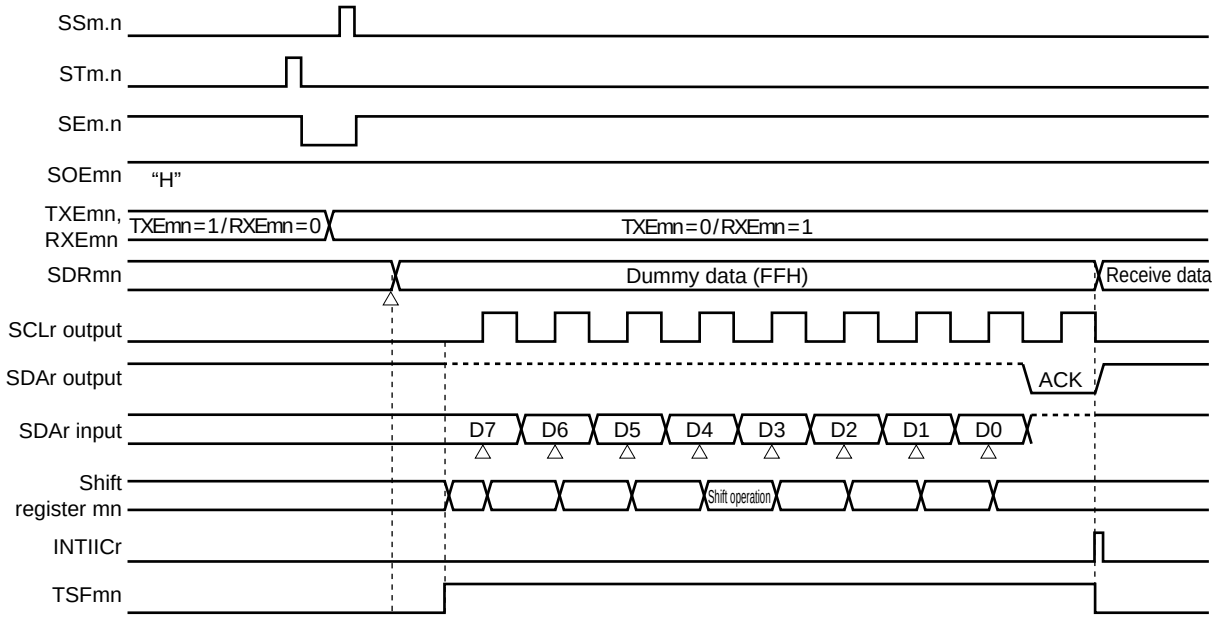
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

2. □: Setting is fixed in the IIC mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

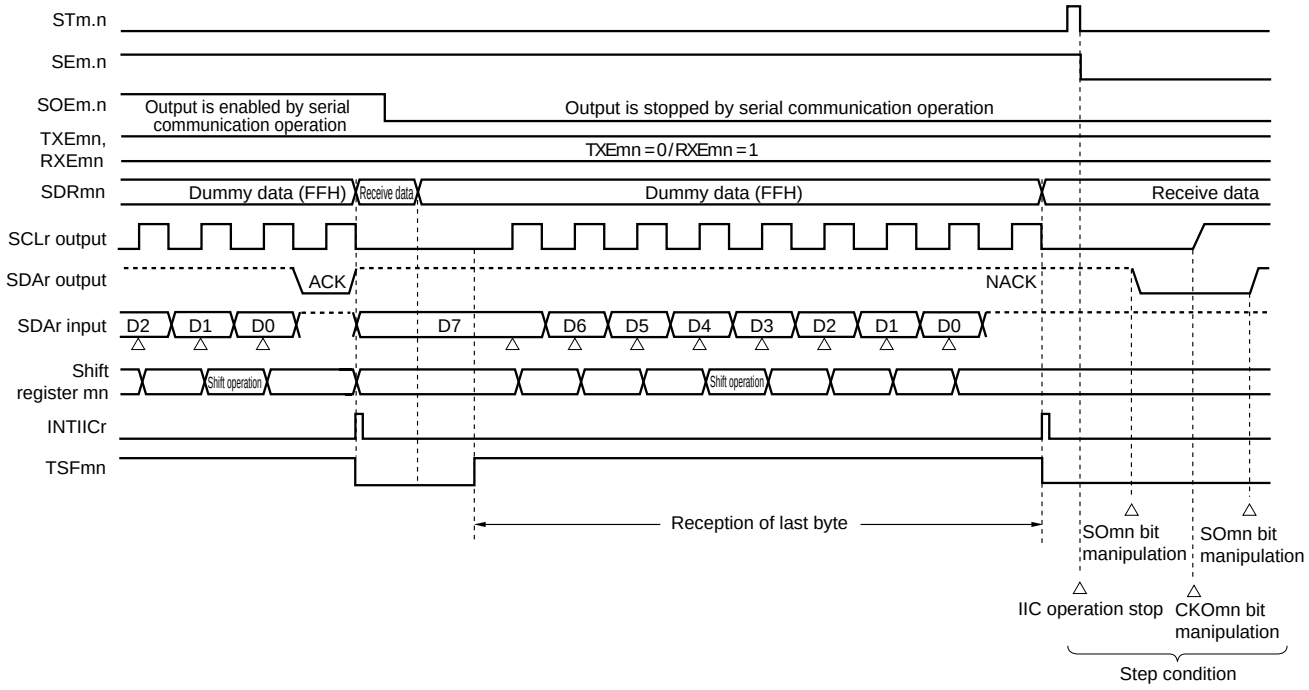
(2) Processing flow

Figure 13-125. Timing Chart of Data Reception

(a) When starting data reception

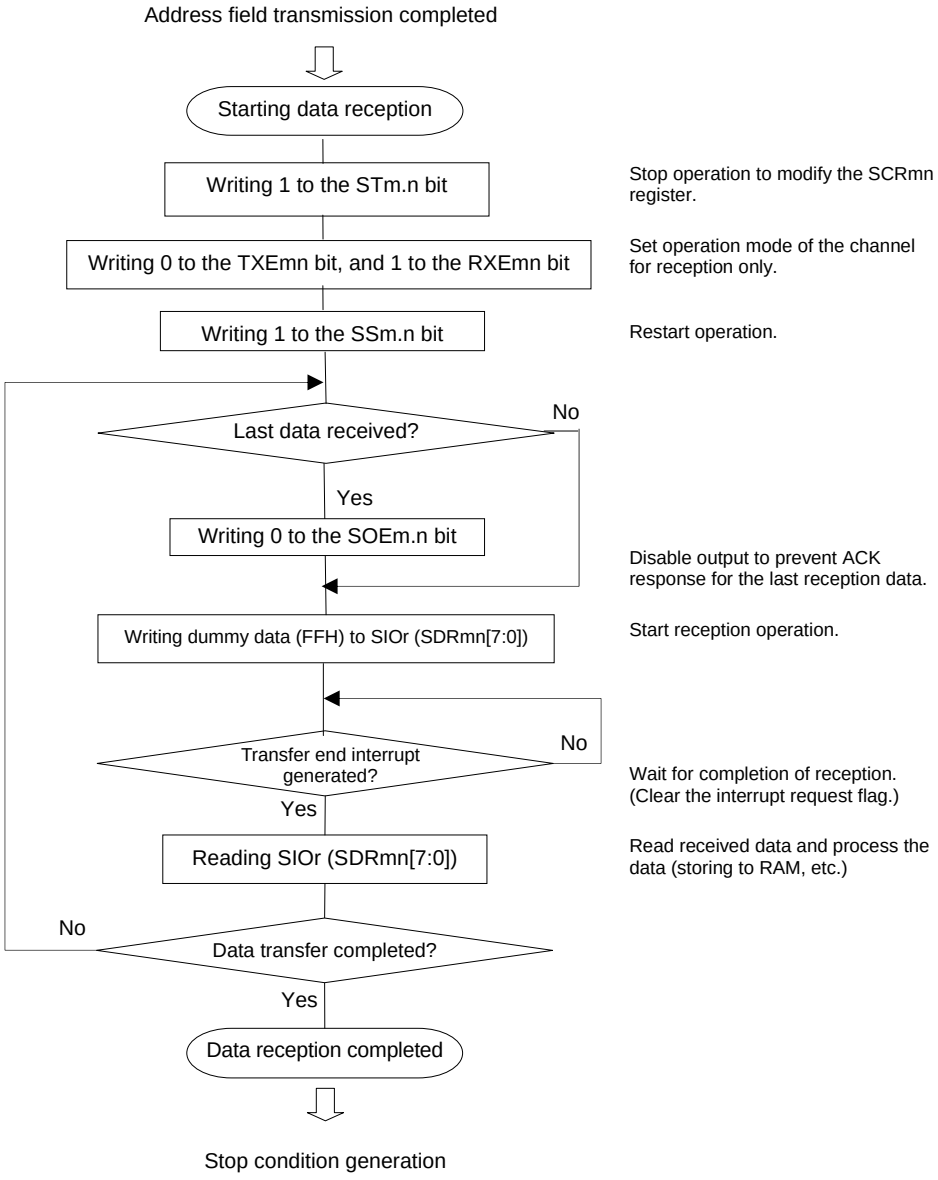


(b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

Figure 13-126. Flowchart of Data Reception



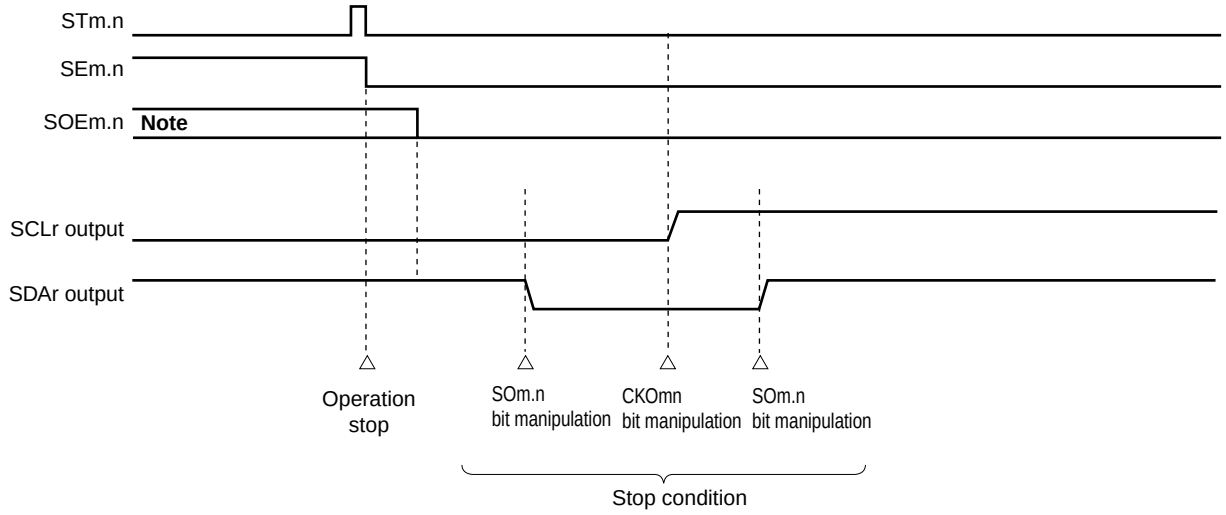
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the STm.n bit of serial channel stop register m (STm) to stop operation and generating a stop condition.

13.8.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

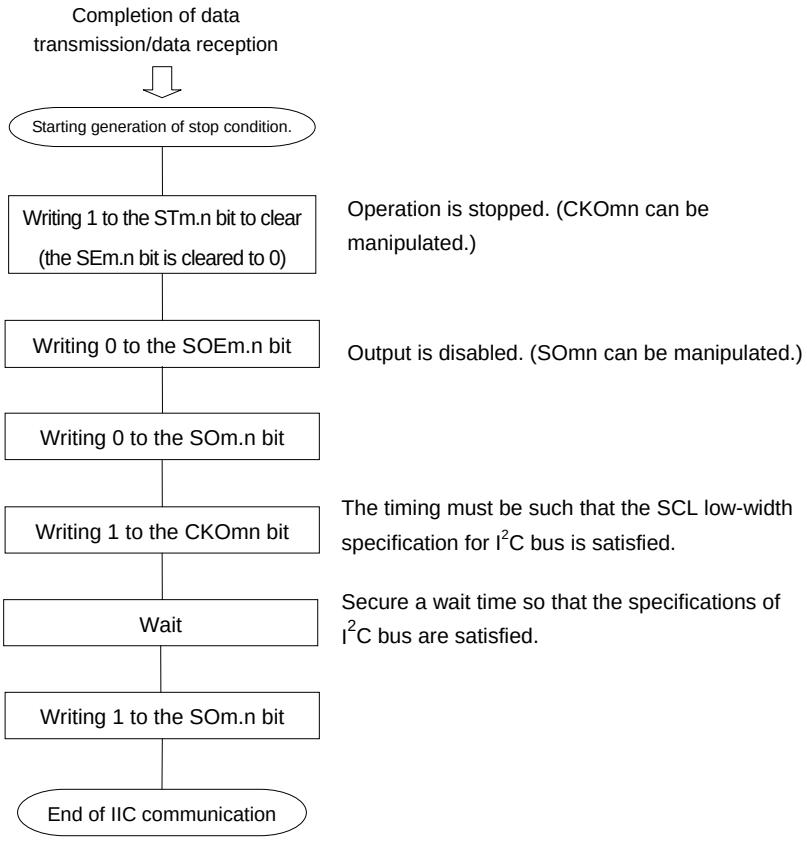
(1) Processing flow

Figure 13-127. Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEm.n bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 13-128. Flowchart of Stop Condition Generation



13.8.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution Setting SDRmn[15:9] = 0000000B is prohibited. Set SDRmn[15:9] to 0000001B or more.

The duty ratio of the SCL signal output from the simplified I^2C is 50%.

Remarks 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 13-5. Selection of Operation Clock For Simplified I^2C

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	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz
Other than above									Setting prohibited	

<R> **Note** Stop the operation of the serial array unit (SAU) (by setting bits 3 to 0 of ST0 register and bits 1 and 0 of ST1 and STS register to 1) before changing operation clock (f_{CLK}) selection (by changing the system clock control register (CKC) value).

Remarks 1. X: Don't care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Here is an example of setting an IIC transfer rate where $f_{MCK} = f_{CLK}$ = 32 MHz.

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 32 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	79	100 kHz	0.0%
400 kHz	f_{CLK}	41	380 kHz	5.0% Note

Note The error cannot be controlled to about 0%, because the duty ratio of the SCL signal is 50%.

13.8.6 Procedure for processing errors that occurred during simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) communication

The procedure for processing errors that occurred during simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21) communication is described in Figure 13-129.

Figure 13-129. Processing Procedure in Case of Parity Error (ACK error) in Simplified I^2C Mode

Software Manipulation	Hardware Status	Remark
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STm.n bit of serial channel stop register m (STm) to 1. →	The SEm.n bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets the SSm.n bit of serial channel start register m (SSm) to 1. →	The SEm.n 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 = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21)
mn = 00 to 03, 10, 11

CHAPTER 14 ASYNCHRONOUS SERIAL INTERFACE LIN-UART (UARTF)

In the RL78/F12, an asynchronous serial interface LIN-UART (UARTF) is provided.

14.1 Features

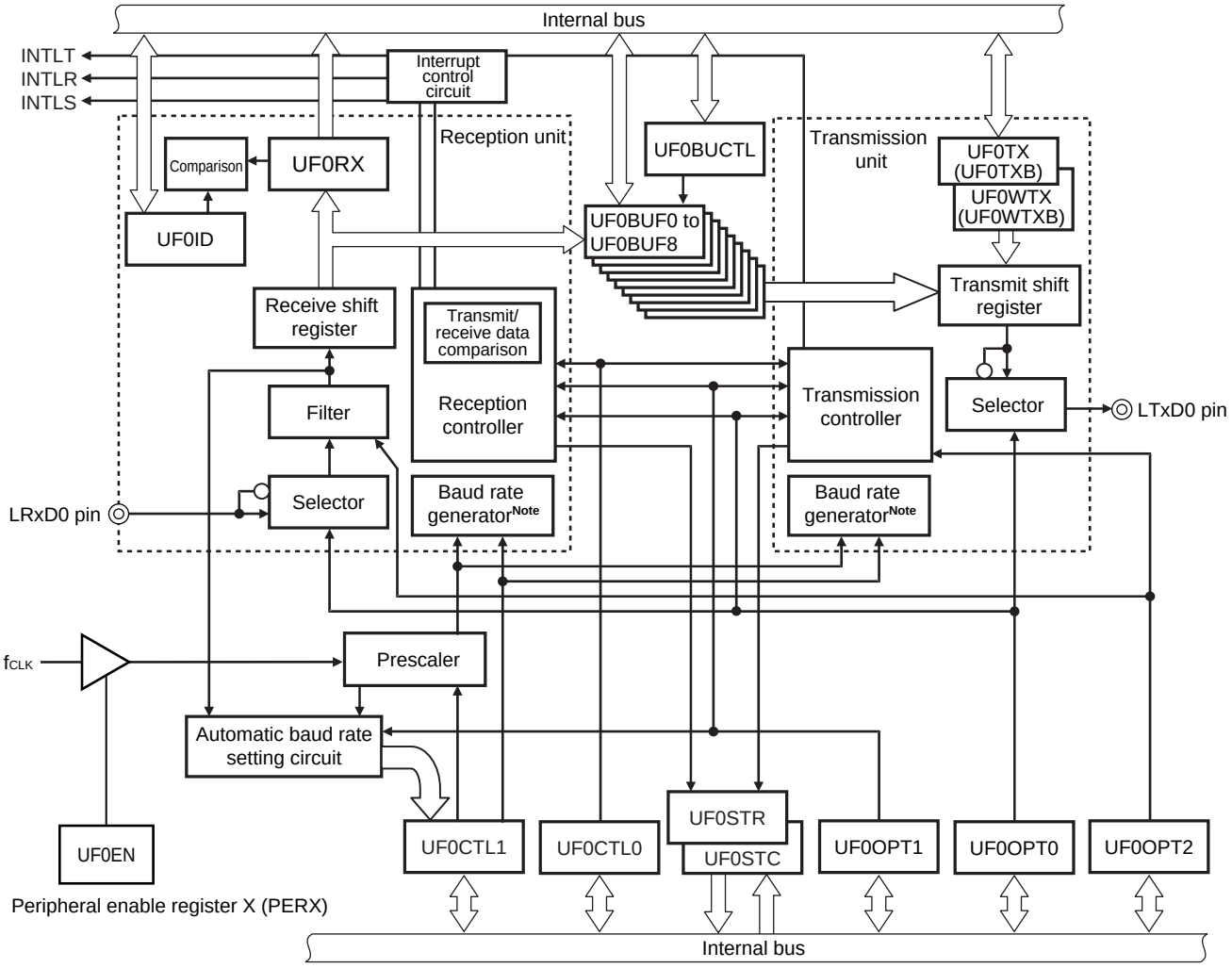
- Maximum transfer rate: 1 Mbps (using dedicated baud rate generator)
- Full-duplex communication: Internal LIN-UART receive data register 0 (UF0RX)
  Internal LIN-UART transmit data register 0 (UF0TX)
- 2-pin configuration: LTxD0: Transmit data output pin
  LRxD0: Receive data input pin
- Reception data/reception error detection function
  - Parity error
  - Framing error
  - Overrun error
  - Function to detect consistency errors in LIN communication data
  - Function to detect successful BF reception
  - ID parity error
  - Checksum error
  - Response preparation error
  - ID match function
  - Expansion bit detection function
- Interrupt sources: 3
  - Reception complete interrupt (INTLR)
  - Transmission interrupt (INTLT)
  - Status interrupt (INTLS)
- Character length: 7, 8 bits
- Communication with 9-bit data length possible by expansion bit setting
- When an expansion bit is at the expected level, the received data can be compared with 8-bit data set in a register in advance
- Internal 3-bit prescaler
- Parity function: Odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- On-chip dedicated baud rate generator
- MSB-/LSB-first transfer selectable
- Transmit/receive data inverted input/output possible
- Guarantee for stop bit of reception (suspension of transmission start during stop bit of reception when starting transmission possible)

○ Transmission/reception function in the LIN (Local Interconnect Network) communication format
- 13 to 20 bits selectable for BF transmission
- Recognition of 11 bits or more in the LIN communication format possible for BF reception
- BF reception flag provided
- Detection of new BF reception possible during data communication
- Function to check consistency of transmit data provided (function to detect mismatches by comparing transmit data and receive data)
- Automatic slave baud rate setting
- Automatic checksum generation function provided (function to automatically calculate the checksum during response transmission or response reception)
- ID parity check function provided (function to automatically check the parity bit of the PID received)

Remark LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.
LIN communication is single-master communication, and up to 15 slaves can be connected to one master.
The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).
In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.
In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±14% or less.

14.2 Configuration

Figure 14-1. Block Diagram of Asynchronous Serial Interface LIN-UART



Note For the configuration of the baud rate generator, see **Figure 14-70 Configuration of Baud Rate Generator**.

LIN-UART consists of the following hardware units.

Table 14-1. Configuration of LIN-UART0

Item	Configuration
Registers	Peripheral enable register X (PERX) LIN-UART0 control registers 0, 1 (UF0CTL0, UF0CTL1) LIN-UART0 option registers 0 to 2 (UF0OPT0 to UF0OPT2) LIN-UART0 status register (UF0STR) LIN-UART0 status clear register (UF0STC) LIN-UART0 receive shift register LIN-UART0 receive data register (UF0RX) LIN-UART0 8-bit receive data register (UF0RXB) LIN-UART0 transmit shift register LIN-UART0 transmit data register (UF0TX) LIN-UART0 8-bit transmit data register (UF0TXB) LIN-UART0 wait transmit data register (UF0WTX) LIN-UART0 8-bit wait transmit data register (UF0WTXB) LIN-UART0 ID setting register (UF0ID) LIN-UART0 buffer registers 0 to 8 (UF0BUF0 to UF0BUF8) LIN-UART0 buffer control register (UF0BUCTL) Serial communication pin select register (STSEL) Port mode register 5, X2 (PM5, PMX2)

14.3 Control Registers

(1) Peripheral enable register X (PERX)

The PERX register is used to set whether to use each peripheral hardware unit. Power consumption and noise can be reduced, because clock supply will be stopped for the hardware not to be used.

When using LIN-UART, be sure to set the bit (bit 2 (UF0EN)) of the LIN-UART to be used to 1.

Set PERX by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 14-2. Format of Peripheral Enable Register 0 (PER0)

Address: F0500H After reset: 00H R/W

	7	6	5	4	3	<2>	<1>	<0>
PERX	0	0	0	0	0	UF0EN	SAUSEN	WUTEN

UF0EN	LIN-UART0 input clock control
0	Stops input clock supply. • Writing to SFR to be used with LIN-UART0 is disabled. • LIN-UART0 is in reset state.
1	Supplies input clock. • Reading from and writing to SFR to be used with LIN-UART0 is enabled.

(2) LIN-UART0 control register 0 (UF0CTL0)

The UF0CTL0 register is an 8-bit register that controls serial communication operation of LIN-UART0.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 10H.

Figure 14-3. Format of LIN-UART0 Control Register 0 (UF0CTL0) (1/2)

Address: F0520H After reset: 10H R/W

	7	<6>	<5>	4	3	2	1	0
UF0CTL0	0	UF0TXE	UF0RXE	UF0DIR	UF0PS1	UF0PS0	UF0CL	UF0SL

UF0TXE	Transmission operation enable
0	Stops transmission operation.
1	Enables transmission operation.
• The setting of the UFnTDL bit in the UFnOPT0 register is reflected in the LTxDn pin level, irrespective of the value of the UFnTXE bit. • When clearing the transmission enable bit (UF0CTL0.UF0TXE) after transmission completion, set (UF0OPT2.UF0ITS = 1) a transmission interrupt upon transmission completion and confirm that the transmission interrupt has been generated, or clear the bit after having confirmed that the transmission status flag (UF0STR.UF0TSF) has been cleared to “0” and communication has been completed.	

UF0RXE	Reception operation enable
0	Stops reception operation. An interrupt is not generated and received data is not stored.
1	Enables reception operation.

UF0DIR	Communication direction mode (MSB/LSB) selection
0	MSB first
1	LSB first
• Rewriting is possible only when UF0TXE = UF0RXE = 0. • To perform transmission and reception in the LIN communication format, set the UF0DIR bit to “1”.	

Figure 14-3. Format of LIN-UART0 Control Register 0 (UF0CTL0) (2/2)

UF0PS1	UF0PS0	Parity selection during transmission	Parity selection during reception
0	0	No parity output	Reception with no parity
0	1	0 parity output	No parity check
1	0	Odd parity output	Odd parity check
1	1	Even parity output	Even parity check

- Rewriting is possible only when UF0TXE = UF0RXE = 0.
- If “Reception with no parity” or “Reception with 0 parity” is selected during reception, a parity check is not performed.
  Consequently, a status interrupt (INTLS) is not generated with parity error, because the UF0PE bit of the UF0STR register is not set.
- To perform transmission and reception in the LIN communication format, set the UF0PS1 and UF0PS0 bits to “00”.

UF0CL	Specification of data character length of 1 frame of transmit/receive data
0	7 bits
1	8 bits

- Rewriting is possible only when UF0TXE = UF0RXE = 0.
- To perform transmission and reception in the LIN communication format, set the UF0CL bit to “1”.

UF0SL	Specification of length of stop bit for transmit data
0	1 bit
1	2 bits

Rewriting is possible only when UF0TXE = UF0RXE = 0.

Caution During receive data framing error detection, only the first bit of the stop bits is checked, regardless of the value of the stop bit length select bit (UF0SL).

Remark For details of parity, see **14.5.7 Parity types and operations**.

(3) LIN-UART0 control register 1 (UF0CTL1)
See **14.10 (2) LIN-UART0 control register 1 (UF0CTL1)** for details.

(4) LIN-UART0 option register 0 (UF0OPT0)

The UF0OPT0 register is an 8-bit register that controls serial communication operation of LIN-UART0.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 14H.

Figure 14-4. Format of LIN-UART0 Option Register 0 (UF0OPT0) (1/3)

Address: F0521H After reset: 14H R/W

	<7>	<6>	<5>	4	3	2	1	0
UF0OPT0	UF0BRF	UF0BRT	UF0BTT	UF0BLS2	UF0BLS1	UF0BLS0	UF0TDL	UF0RDL

UF0BRF	BF reception flag
0	When the UF0CTL0.UF0RXE = 0 is set. Also upon normal end of BF reception.
1	While waiting for successful BF reception (when the UF0BRT bit is set)

- BF (Break Field) reception is judged during LIN communication.
- The UF0BRF bit retains “1” when a BF reception error occurs, and is cleared to “0” when BF reception is started again and ends normally. It cannot be cleared by instruction.
- The UF0BRF bit is read-only.

Caution When the UF0BRF bit is 1, whether BF reception has ended normally can be judged by checking whether the low-level period is at least 11 bits, when a high level, including noise, is input to the receive input data even for a moment. If the low-level period is at least 11 bits, BF reception is judged to be performed successfully.
When in BF reception enable mode during communication (UFnMD1, UFnMD0 = 10B), normal completion of BF reception can be confirmed by checking that the successful BF reception flag (UFnBSF) is set to 1 when a status interrupt is detected.
In BF reception enable mode during communication, a reception complete interrupt is not generated even by setting the BF reception trigger bit. However, the normal completion of BF reception can be confirmed also by checking that the UFnBRF flag is 0 when a status interrupt is detected after setting the bit.

UF0BRT	BF reception trigger
0	–
1	BF reception trigger

- This is the BF reception trigger bit during LIN communication, and when read, “0” is always read. For BF reception, set (1) the UF0BRT bit to enable BF reception.
- Set the UF0BRT bit after having set UF0CTL0.UF0RXE to “1”.
- The status flag will not be updated, an interrupt request signal will not be generated, and data will not be stored.
- This bit can only be set again when the UF0BRF bit is 0.
- When BF reception is enabled during communication, BF reception is detected as the low-level period between when the UF0BRT bit is set and when the rising edge of the reception input data is detected. Therefore, a BF will be detected even if the UF0BRT bit is set during BF reception.

Cautions 1. To release a BF reception enable state without receiving a BF, UF0RXE must be cleared to 0.
2. Transmitting data while UF0DCS and UF0BRF are “1” is prohibited. BF transmission, however, can be performed.
3. Setting the UF0BRT bit in automatic baud rate mode (UF0MD1, UF0MD0 = 11B) is prohibited.

Figure 14-4. Format of LIN-UART0 Option Register 0 (UF0OPT0) (2/3)

UF0BTT	BF transmission trigger
0	–
1	BF transmission trigger

- This is the BF transmission trigger bit during LIN communication, and when read, “0” is always read.
- Set the UF0BTT bit after having set UF0CTL0.UF0TXE to “1”.

Cautions 1. Setting both the next transmit data and the UF0BTT bit during data transmission is prohibited.
Also, even if the UF0BTT bit is set during a BF transmission, it is invalid (a BF transmission is performed once and ends).
2. Completion of a BF transmission can be judged by checking that the UF0TSF bit is “0” after the BF transmission trigger bit has been set. If the next transmit data has been written to the UF0TX register during the BF transmission, however, the UF0TSF bit will not be cleared when transmitting the BF has been completed, but will retain “1”.
When in BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B), completion of a BF transmission can also be judged by checking that the successful BF reception flag (UF0BSF) is “1” after a status interrupt has been detected.
3. Setting the UF0BTT bit is prohibited in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).

UF0BLS2	UF0BLS1	UF0BLS0	BF length selection bit
1	0	1	13-bit output (reset value)
1	1	0	14-bit output
1	1	1	15-bit output
0	0	0	16-bit output
0	0	1	17-bit output
0	1	0	18-bit output
0	1	1	19-bit output
1	0	0	20-bit output

This bit can be set when UF0CTL0.UF0TXE is “0”.

UF0TDL	Transmit data level bit
0	Normal output of transfer data
1	Inverted output of transfer data

- The LTxD0 output value can be inverted by using the UF0TDL bit.
- This bit can be set when UF0CTL0.UF0TXE is “0”.

Cautions 1. The LTxD0 output level is inverted by controlling the UF0TDL bit, regardless of the value of the UF0TXE bit. Consequently, if the UF0TDL bit is set to “1” even when operation is disabled, the LTxD0 output becomes low level.
2. To perform transmission and reception in the LIN communication format, set UF0TDL to “0”.

Figure 14-4. Format of LIN-UART0 Option Register 0 (UF0OPT0) (3/3)

UF0RDL	Receive data level bit
0	Normal input of transfer data
1	Inverted input of transfer data

- The LRxD0 input value can be inverted by using the UF0RDL bit.
- This bit can be set when UF0CTL0.UF0RXE is “0”.

Cautions 1. Be sure to enable reception (UF0RXE = 1) after having changed the UF0RDL bit. When the UF0RDL bit is changed after reception has been enabled, the start bit will be falsely detected, depending on the pin level at that time.

2. To perform transmission and reception in the LIN communication format, set UF0RDL to “0”.

(5) LIN-UART0 option register 1 (UF0OPT1)

The UF0OPT1 register is an 8-bit register that controls serial communication operation of LIN-UART0.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution Set the UF0OPT1 register when UF0TXE and UF0RXE are "0". Only the UF0EBC bit, however, can be changed even if UF0TXE is "1" or UF0RXE is "1". See 14.8.3 Expansion bit mode reception (with data comparison) for details.

Figure 14-5. Format of LIN-UART0 Option Register 1 (UF0OPT1) (1/3)

Address: F0524H After reset: 00H R/W

	7	6	<5>	4	3	2	1	0
UF0OPT1	UF0EBE	UF0EBL	UF0EBC	UF0IPCS	UF0ACE	UF0MD1	UF0MD0	UF0DCS

UF0EBE	Expansion bit enable bit
0	Disables expansion bit operation. (Transmission and reception are performed in the data length (7, 8 bits) set to UF0CTL0.UF0CL.)
1	Enables expansion bit operation. (Transmission and reception are performed in data length (9 bits) when UF0CTL0.UF0CL is "1".)

Cautions
1. **To perform transmission and reception in 9-bit units by setting (1) the UF0EBE bit, the data length must be set to 8 bits (UF0CL = 1). If the data length is set to 7 bits (UF0CL = 0), the setting of the UF0EBE bit will be invalid.**
2. **To perform transmission and reception in the LIN communication format, set UF0EBE to "0".**
3. **The expansion bit is included in parity check.**

UF0EBL	Expansion bit detection level select bit
0	Selects expansion bit value "0" as expansion bit detection level.
1	Selects expansion bit value "1" as expansion bit detection level.

If the level selected by the UF0EBL bit is detected as the expansion bit when the expansion bit has been enabled (UF0CL = UF0EBE = 1), a status interrupt request signal (INTLS) will be generated and an expansion bit detection flag (UE0EBD) will be set.

If the inversion level is detected as the expansion bit, a reception complete interrupt request signal (INTLR) will be generated, but an expansion bit detection flag will not be set.

Remark The UF0EBL bit becomes valid only if UF0CL = UF0EBE = 1. See **14.8.2 Expansion bit mode reception (no data comparison)** and **14.8.3 Expansion bit mode reception (with data comparison)** for details.

Figure 14-5. Format of LIN-UART0 Option Register 1 (UF0OPT1) (2/3)

UF0EBC	Expansion bit data comparison enable bit
0	No comparison (INTLR or INTLS is always generated upon completion of data reception.)
1	Compares UF0RX register and UF0ID register when the level selected for the UF0EBL bit has been detected as the expansion bit. (INTLS is generated only when the UF0RX register and UF0ID register have matched.)

The UF0EBC bit is used to enable comparison between the received data and the UF0ID register when the expansion bit has been enabled (UF0CL = UF0EBE = 1).

Remark The UF0EBC bit becomes valid only if UF0CL = UF0EBE = 1. See **14.8.2 Expansion bit mode reception (no data comparison)** and **14.8.3 Expansion bit mode reception (with data comparison)** for details.

UF0IPCS	ID parity check select bit
0	No automatic ID parity check (Calculating the parity of the PID by using software and checking are required.)
1	Automatic ID parity check

- The UF0IPCS bit is used to select how to handle automatic checking of the parity bit of the received PID, when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).
- If UF0IPCS is “1”, the parity bit is checked when the PID received in LIN communication is stored into the UF0ID register. When an incorrect result has been detected, an ID parity error flag (UF0IPE) will be set and a status interrupt request signal (INTLS) will be generated.

Remark The UF0IPCS bit becomes valid only in the automatic baud rate mode (UF0MD1, UF0MD0 = 11B). See **14.7.3 ID parity check function** for details.

UF0ACE	Automatic checksum enable bit
0	Disables automatic checksum calculation. Response transmission: Checksum must be calculated by using software and set to a buffer. Response reception: Checksum must be calculated from the data stored into the buffer by using software, and compared and checked with the checksum obtained via communication.
1	Enables automatic checksum calculation. Response transmission: Checksum is automatically calculated from the data set to a buffer and is automatically appended at the end of response transmission. Response reception: Checksum is automatically calculated from the data stored into the buffer and is automatically compared and checked with the checksum obtained via communication.

- The UF0ACE bit is used to select how to handle automatic checksum calculation during response transmission and response reception, when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).
- When response reception is performed while UF0ACE is “1”, the checksum received in LIN communication will be checked when it is stored into a receive buffer. When an incorrect result has been detected, a checksum error flag (UF0CSE) will be set and a status interrupt request signal (INTLS) will be generated.

Remark The UF0ACE bit becomes valid only in the automatic baud rate mode (UF0MD1, UF0MD0 = 11B). See **14.7.4 Automatic checksum function** for details.

Figure 14-5. Format of LIN-UART0 Option Register 1 (UF0OPT1) (3/3)

UF0MD1	UF0MD0	LIN-UART operation mode select bit
0	0	Normal UART mode
0	1	Setting prohibited
1	0	LIN communication: BF reception enable mode during communication Detects a new Break Field during data communication. (When a low level has been detected at the stop bit position, a wait is performed until the next high level is detected and a new BF reception is recognized if the low-level period is at least 11 bits.)
1	1	LIN communication: Automatic baud rate mode

Cautions 1. Setting to automatic baud rate mode (UF0MD1, UF0MD0 = 11B) is prohibited for a LIN communication master.
2. Be sure to also set the UF0DCS bit to "1" when in BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B) or in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).

UF0DCS	Data consistency check select bit
0	Does not check data consistency.
1	Checks data consistency.

- The UF0DCS bit is used to select how to handle a data consistency check when transmitting data via LIN communication. For details, see **14.5.8 Data consistency check**.
- When UF0DCS is "1", transmit data and receive data will be compared when transmitting data via LIN communication. When a mismatch is detected, a data consistency error flag (UF0DCE) will be set and a status interrupt request signal (INTLS) will be generated.

Cautions 1. When using LIN communication, the UF0DCS bit can be set. Otherwise, clear the UF0DCS bit to "0".
2. When setting (1) the UF0DCS bit, fix the data bit length to 8 bits. Appending a parity bit is prohibited.
3. Be sure to also set the UF0DCS bit to "1" when in BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B) or in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).

(6) LIN-UART0 option register 2 (UF0OPT2)

The UF0OPT2 register is an 8-bit register that controls serial communication operation of LIN-UART0.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Figure 14-6. Format of LIN-UART0 Option Register 2 (UF0OPT2)

Address: F0525H After reset: 00H R/W

	7	6	5	4	3	2	1	<0>
UF0OPT2	0	0	0	0	0	0	UF0RXFL	UF0ITS

UF0RXFL	Bit to select use of receive data noise filter
0	Uses noise filter.
1	Does not use noise filter.
The UF0RXFL bit is used to select use of the noise filter. See **14.9 Receive Data Noise Filter** for details. **Caution Be sure to set the UF0RXFL bit when UF0CTL0.UF0RXE is "0".**	

UF0ITS	Transmission interrupt (INTLT) generation timing select bit
0	Outputs transmission interrupt request upon transmission start.
1	Outputs transmission interrupt request upon transmission completion.
Caution Be sure to set the UF0ITS bit when UF0CTL0.UF0TXE is "0". **The UF0ITS bit can be changed to 1 after transmission of the last data is started only when completion of transmitting the last data must be known during successive transmission (UF0ITS = 0). However, the change must be completed before the transmission is completed.**	

(7) LIN-UART0 status register (UF0STR)

The UF0STR register is a 16-bit register that displays the LIN-UART0 communication status and reception error contents.

This register is read-only, in 16-bit units.

Reset sets this register to 0000H.

Caution Flags other than the UF0TSF and UF0RSF flags are retained until the target bits of the LIN-UART0 status clear register (UF0STC) are written ("1") and then cleared. To clear a status flag, use a 16-bit manipulation instruction to write ("1") and clear the target bits of the LIN-UART0 status clear register (UF0STC).

Figure 14-7. Format of LIN-UART0 Status Register (UF0STR) (1/6)

Address: F0526H, F0527H After reset: 0000H R

	15	14	13	12	11	10	9	8
UF0STR	0	UF0IPE	UF0CSE	UF0RPE	UF0HDC	UF0BUC	UF0IDM	UF0EBD
	7	6	5	4	3	2	1	0
	UF0TSF	UF0RSF	0	UF0BSF	UF0DCE	UF0PE	UF0FE	UF0OVE

UF0IPE	ID parity error flag
0	No ID parity error has occurred.
1	An ID parity error has occurred. <ID parity error source> Parity of received PID is incorrect

- The UF0IPE bit is a flag indicating the check status by the ID parity check function. It becomes "1", if the parity of the received PID is incorrect when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B). See **14.7.3 ID parity check function** for details.
- The UF0IPE bit will not be cleared until "1" is written to the UF0CLIPE bit of the UF0STC register, because the UF0IPE bit is a cumulative flag. It will not be set if the ID parity check function has been disabled (UF0IPCS = 0).

Figure 14-7. Format of LIN-UART0 Status Register (UF0STR) (2/6)

UF0CSE	Checksum error flag
0	No checksum error has occurred.
1	A checksum error has occurred. <Checksum error source> Result of comparing checksum automatically calculated from data stored into buffer and checksum obtained via communication is incorrect during response reception

- The UF0CSE bit is a flag indicating the check status by the automatic checksum function. It becomes "1" if the received checksum is incorrect when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B) and during response reception. See **14.7.4 Automatic checksum function** for details.
- The UF0CSE bit will not be cleared until "1" is written to the UF0CLCSE bit of the UF0STC register, because the UF0CSE bit is a cumulative flag. It will not be set if the automatic checksum function has been disabled (UF0ACE = 0).

Cautions 1. The check sum error flag will not be set during response transmission. Perform a data consistency check to check for errors.

2. Receive data will be stored in the UF0RX register during response transmission. However, no overrun error will be set, even if the receive data is not read. Consequently, the received check sum can be checked by reading the UF0RX register after the reception completion interrupt has occurred.

UF0RPE	Response preparation error flag
0	No response preparation error has occurred.
1	A response preparation error has occurred. < Response preparation error source> Response could not be prepared before completion of receiving first byte of receive data after header reception

- The UF0RPE bit is a flag indicating the check status by the response preparation detection function. It becomes "1", if a response (setting of UF0NO, UF0RRQ bits) could not be prepared in automatic baud rate mode (UF0MD1, UF0MD0 = 11B). See **14.7.2 Response preparation error detection function** for details.
- The UF0RPE bit will not be cleared until "1" is written to the UF0CLRPE bit of the UF0STC register, because the UF0RPE bit is a cumulative flag. It will not be set when not in automatic baud rate mode (UF0MD1, UF0MD0 = 00B or 10B).

UF0HDC	Header reception completion flag
0	Header reception is not completed.
1	Receiving header has been completed

- The UF0HDC bit is a flag indicating completion of receiving a header. It becomes "1" when receiving the header has been completed when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B). See **14.7.1 Automatic baud rate setting function** for details.
- The UF0HDC bit will not be cleared until "1" is written to the UF0CLHDC bit of the UF0STC register, because the UF0HDC bit is a cumulative flag. It will not be set when not in automatic baud rate mode (UF0MD1, UF0MD0 = 00B or 10B).

Caution This flag will not be set by an error during PID reception.

Figure 14-7. Format of LIN-UART0 Status Register (UF0STR) (3/6)

UF0BUC	Buffer transmission/reception completion flag
0	Buffer transmission/reception is not completed.
1	Buffer transmission/reception is completed <Buffer transmission/reception completion condition> The set number of data is transmitted or received. (only when transmitted when in normal UART mode)

- The UF0BUC bit is a flag indicating the data transmission and reception status of a buffer. It becomes "1" when the set number of data items have been transmitted or received without an error occurring. See **14.6.1 UART buffer mode transmission** and **14.7 LIN Communication Automatic Baud Rate Mode** for details.
- The UF0BUC bit will not be cleared until "1" is written to the UF0CLBUC bit of the UF0STC register, because the UF0BUC bit is a cumulative flag. It will be set only when in normal UART mode (UF0MD1, UF0MD0 = 00B) or automatic baud rate mode (UF0MD1, UF0MD0 = 11B).

UF0IDM	ID match flag
0	The ID does not match.
1	The IDdoes match <ID match condition> When 8 bits of receive data, excluding expansion bit, have matched with UF0ID register value set in advance

- The UF0IDM bit is a flag indicating the result of comparing the 8 bits of receive data, excluding the expansion bit, and the UF0ID register value set in advance when expansion bit data comparison has been enabled (UF0EBC = 1) by enabling the expansion bit (UF0CL = UF0EBE = 1). The comparison will be performed with the data for which the level set by using the expansion bit detection level select bit (UF0EBL) has been detected. The UF0IDM bit becomes "1" when the comparison result has matched. See **14.8.3 Expansion bit mode reception (with data comparison)** for details.
- The UF0IDM bit will not be cleared until "1" is written to the UF0CLIDM bit of the UF0STC register, because the UF0IDM bit is a cumulative flag. It will not be set when the expansion bit has not been enabled and expansion bit data comparison has not been enabled (UF0CL = UF0EBE = UF0EBC = 1).

UF0EBD	Expansion bit detection flag
0	An extension bit is not detected
1	An extension bit is detected <Expansion bit detection condition> When level set by using expansion bit detection level select bit (UF0EBL) has been detected for expansion bit

- The UF0EBD bit is a flag indicating detection of the level set by using the expansion bit detection level select bit (UF0EBL) when the expansion bit has been enabled (UF0CL = UF0EBE = 1). It becomes "1" when the setting level has been detected. See **14.8.2 Expansion bit mode reception (no data comparison)** and **14.8.3 Expansion bit mode reception (with data comparison)** for details.
- The UF0EBD bit will not be cleared until "1" is written to the UF0CLEBD bit of the UF0STC register, because the UF0EBD bit is a cumulative flag. It will not be set when the expansion bit has been disabled (UF0EBE = 0).

Figure 14-7. Format of LIN-UART0 Status Register (UF0STR) (4/6)

UF0TSF	Transmission status flag
0	A transmit operation is not performed. <Transmission stop condition> • When UF0CTL0.UF0TXE has been cleared to “0” • When there was no next transmit data after transmission completion, and at the same time, BF transmit trigger bit (UF0BTT) has not been set • When there was no next transmit data in UF0TX, UF0WTX, UF0BUF0 to UF0BUF8 bit after BF transmission has ended • When the transmission after data consistency error detection is completed
1	A transmit operation is performed. <Transmission start condition> • Writes to UF0TX, UF0WTX register • When BF transmit trigger bit (UF0BTT) has been set[Note] • When the transmission request bit (UF0TRQ) is set
• The UF0TSF bit is always “1” when successive transmission is performed. • To initialize the transmission unit, check that UF0TSF is “0” before performing initialization. If initialization is performed while UF0TSF = 1, the transmission will be aborted midway. • If a BF is detected in BF reception enabled mode during communication and when transmitting data, or if a BF/SF is detected in automatic baud rate mode and when transmitting data, the UF0DCE flag will be set and the UF0TSF bit will be cleared when a status interrupt (INTLS) is issued. **Note** Only during BF period	

UF0RSF	Reception status flag
0	A receive operation is not performed. <Reception stop condition> • When UF0CTL0.UF0RXE has been cleared to “0” • When at sampling point of stop bit (first bit) during reception • When UF0BRT = 1 is set • When a BF is detected in BF reception enabled mode during communication • When a BF/SF is detected in automatic baud rate mode
1	A receive operation is performed. <Reception start condition> When a start bit is detected (when it is detected that the data is 0 at the sampling point of the bit after the LRxD0 falling edge is detected)
To initialize the reception unit, check that UF0RSF is “0” before performing initialization. If initialization is performed while UF0RSF = 1, the reception will be aborted midway.	

Figure 14-7. Format of LIN-UART0 Status Register (UF0STR) (5/6)

UF0BSF	Successful BF reception flag
0	BF reception is not successfully performed.
1	BF reception is successfully performed. <BF reception stop condition> When successive low levels (BF) of at least 11 bits have been received

- The UF0BSF bit is a flag indicating that receiving a BF has been performed successfully. It becomes “1” when successive low levels (BF) of at least 11 bits have been received when in BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B) (This occurs at the same time as the status interrupt (INTLS) is issued upon the detection of the rising edge of the LRxD0 pin.).
- The start of a new frame slot must be checked by reading the UF0BSF bit via status interrupt servicing, because the BF may also be received during data communication when in BF reception enable mode during communication.
- The UF0BSF bit will not be cleared until “1” is written to the UF0CLBSF bit of the UF0STC register, because the UF0BSF bit is a cumulative flag. It will not be set when not in BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B).

UF0DCE	Data consistency error flag
0	No data consistency error has occurred.
1	A data consistency error has occurred. <Data consistency error source> When transmit data and receive data do not match in LIN communication

- When the data consistency check select bit is set (UF0DCS = 1), the transmit data and receive data are compared upon data transmission. The UF0DCE bit becomes “1” at the same time as the status interrupt (INTLS) is issued when a mismatch has been detected.
- The UF0DCE bit will not be cleared until “1” is written to the UF0CLDCE bit of the UF0STC register, because the UF0DCE bit is a cumulative flag. When UF0DCS is “0”, the UF0DCE bit will not be set.

Caution The next transfer will not be performed if a data consistency error is detected. See 14.5.8 Data consistency check for details.

UF0PE	Parity error flag
0	No parity error has occurred.
1	A parity error has occurred. <Parity error source> When parity of data and parity bit do not match during reception

- The operation of the UF0PE bit depends on the settings of the UF0PS1 and UF0PS0 bits.
- The UF0PE bit will not be cleared until “1” is written to the UF0CLPE bit of the UF0STC register or “0” is written to the UF0RXE bit of the UF0CTL0 register, because the UF0PE bit is a cumulative flag. When UF0PS1 and UF0PS0 are “0xB”, the UF0PE bit will not be set. (x: Don’t care)

Figure 14-7. Format of LIN-UART0 Status Register (UF0STR) (6/6)

UF0FE	Framing error flag
0	No framing error has occurred.
1	A framing error has occurred. < Framing error source> When no stop bit is detected during reception

- Only the first bit of the receive data stop bits is checked, regardless of the setting value of the UF0SL bit.
- The UF0FE bit will not be cleared until “1” is written to the UF0CLFE bit of the UF0STC register or “0” is written to the UF0RXE bit of the UF0CTL0 register, because the UF0FE bit is a cumulative flag.

UF0OVE	Overrun error flag
0	No overrun error has occurred.
1	An overrun error has occurred. < Overrun error source> When receive data has been stored into the UF0RX register and the next receive operation is completed before that receive data has been read

- When an overrun error has occurred, the data is discarded without the next receive data being written to the UF0RX register.
- The UF0FE bit will not be cleared until “1” is written to the UF0CLFE bit of the UF0STC register or “0” is written to the UF0RXE bit of the UF0CTL0 register, because the UF0FE bit is a cumulative flag. It will not be set in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).

Caution If no status interrupt due to an ID mismatch is issued while expansion bit data comparison is enabled (UF0EBE = 1 and UF0EBC = 1), as receive data will not be stored in the UF0RX register, the UF0OVE flag will not be set even if the receive data is not read. Furthermore, when transmitting in automatic baud rate mode, the receive data will be always stored in the UF0RX register, but the UF0OVE flag will not be set even if the receive data is not read.

(8) LIN-UART0 status clear register (UF0STC)

The UF0STC register is a 16-bit register that is used to clear an LIN-UART0 status flag.

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

Reset sets this register to 0000H.

Caution An LIN-UART status register (UF0STR) flag can be cleared by writing "1" to a corresponding bit. 0 will be read if the bit is read.

Figure 14-8. Format of LIN-UART0 Status Clear Register (UF0STC) (1/2)

Address: F0528H, F0529H After reset: 0000H R/W

	15	14	13	12	11	10	9	8
UF0STC	0	UF0CLIPE	UF0CLCSE	UF0CLRPE	UF0CLHDC	UF0CLBUC	UF0CLIDM	UF0CLEBD
	7	6	5	4	3	2	1	0
	0	0	0	UF0CLBSF	UF0CLDCE	UF0CLPE	UF0CLFE	UF0CLOVE

UF0CLIPE	Channel n ID parity error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0IPE bit of the UF0STR register.

UF0CLCSE	Channel n checksum error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0CSE bit of the UF0STR register.

UF0CLRPE	Channel n response preparation error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0RPE bit of the UF0STR register.

UF0CLHDC	Channel n header reception completion flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0HDC bit of the UF0STR register.

UF0CLBUC	Channel n buffer transmission/reception completion flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0BUC bit of the UF0STR register.

UF0CLIDM	Channel n ID match flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0IDM bit of the UF0STR register.

UF0CLEBD	Channel n expansion bit detection flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0EBD bit of the UF0STR register.

Figure 14-8. Format of LIN-UART0 Status Clear Register (UF0STC) (2/2)

UF0CLBSF	Channel n successful BF reception flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0BSF bit of the UF0STR register.

UF0CLDCE	Channel n data consistency error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0DCE bit of the UF0STR register.

UF0CLPE	Channel n parity error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0PE bit of the UF0STR register.

UF0CLFE	Channel n framing error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0FE bit of the UF0STR register.

UF0CLOVE	Channel n overrun error flag clear trigger
0	Trigger does not operate.
1	Clears (0) the UF0OVE bit of the UF0STR register.

(9) LIN-UART0 transmit data register (UF0TX)

The UF0TX register is a 16-bit register that is used to set transmit data.

This register can be read or written in 16-bit units. When the UF0TX register is read or written in 8-bit units, it can be accessed as the UF0TXB register.

When no buffer is used and no data consistency error has been detected (UF0DCE = 0) in a transmission enable state (UF0TXE = 1), transmission is started by writing transmit data to the UF0TX register.

When UF0EBE = 0, transmit data of a character length specified by the UF0CL bit will be transmitted.

When UF0EBE = UF0CL = 1, transmit data of 9-bit length will be transmitted. See **14.5.1 Data format** for the transmit data format.

The last data written to the UF0TX register before it is loaded to the transmit shift register is to be transmitted. When UF0ITS is “0”, successive transmission can be performed by writing the next transmit data to the UF0TX register after a transmission interrupt request has been generated. When the next transmit data is written before a transmission interrupt request is generated, the previously written data will be overwritten and only the subsequent data will be transmitted.

Reset input sets this register to 0000H.

Figure 14-9. Format of LIN-UART0 Transmit Data Register (UF0TX)

Address: F0548H, F0549H After reset: 0000H R/W

	15	14	13	12	11	10	9	8
UF0TX	0	0	0	0	0	0	0	UF0TX.8
	7	6	5	4	3	2	1	0
	UF0TX.7	UF0TX.6	UF0TX.5	UF0TX.4	UF0TX.3	UF0TX.2	UF0TX.1	UF0TX.0

When the data length is specified as 7 bits (UF0CL = 0):

- During LSB-first transmission, bits 6 to 0 of the UF0TX register will be transferred as transmit data.
- During MSB-first transmission, bits 7 to 1 of the UF0TX register will be transferred as transmit data.

Cautions 1. If the UF0TX register is written while transmission is disabled (UF0TXE = 0), it will not operate as a transmission start trigger. Consequently, no transmission will be started, even if transmission is enabled after having written to the UF0TX register while transmission was disabled.

2. When the UF0TX register is written in 8-bit units (when the UF0TXB register is written), “0” is written to the UF0TX.8 bit.

3. Writing to the UF0TX register is prohibited when using the UF0BUF0 to UF0BUF8 registers.

4. When using automatic checksum function, set 0000H in the UF0TX register before starting communication.

Remark The UF0TX.8 bit is an expansion bit when expansion bits are enabled (UF0EBE = UF0CL = 1).

(10) LIN-UART0 8-bit transmit data register (UF0TXB)

The UF0TXB register is an 8-bit register that is used to set transmit data.

This register can be read or written in 8-bit units.

When no buffer is used and no data consistency error has been detected (UF0DCE = 0) in a transmission enable state (UF0TXE = 1), transmission is started by writing transmit data to the UF0TXB register.

When UF0EBE = 0, transmit data of a character length specified by the UF0CL bit will be transmitted. For detail of the transmit data format, see **14.5.1 Data format**.

The last data written to the UF0TXB register before it is loaded to the transmit shift register is to be transmitted. When UF0ITS is "0", successive transmission can be performed by writing the next transmit data to the UF0TXB register after a transmission interrupt request has been generated. When the next transmit data is written before a transmission interrupt request is generated, the previously written data will be overwritten and only the subsequent data will be transmitted.

Reset input sets this register to 00H.

Figure 14-10. Format of LIN-UART0 8-bit Transmit Data Register (UF0TXB)

Address: F0548H After reset: 00H R/W

	7	6	5	4	3	2	1	0
UF0TXB	UF0TX.7	UF0TX.6	UF0TX.5	UF0TX.4	UF0TX.3	UF0TX.2	UF0TX.1	UF0TX.0

When the data length is specified as 7 bits (UF0CL = 0):

- During LSB-first transmission, bits 6 to 0 of the UF0TXB register will be transferred as transmit data.
- During MSB-first transmission, bits 7 to 1 of the UF0TXB register will be transferred as transmit data.

Cautions 1. If the UF0TXB register is written while transmission is disabled (UF0TXE = 0), it will not operate as a transmission start trigger. Consequently, no transmission will be started, even if transmission is enabled after having written to the UF0TXB register while transmission was disabled.

2. When the UF0TXB register is written, "0" is written to the UF0TX.8 bit of UF0TX register.

3. Writing to the UF0TXB register is prohibited when using the UF0BUF0 to UF0BUF8 registers.

4. When using automatic checksum function, set 00H in the UF0TXB register before starting communication.

(11) 8-bit transmit data register for LIN-UART0 wait (UF0WTX)

The UF0WTX register is a 16-bit register dedicated to delaying starting transmission until the stop bit of reception is completed during a LIN communication.

This register is write-only, in 16-bit units. When the UF0WTX register is write in 8-bit units, it can be accessed as the UF0WTXB register.

The stop bit length of reception when reception is switched to transmission is guaranteed for the UF0WTX register. See **14.5.11 Transmission start wait function** for details.

The UF0WTX register value will be read when the UF0WTX register has been read.

Reset input sets this register to 0000H.

Figure 14-11. Format of 8-bit transmit data register for LIN-UART0 wait (UF0WTX)

Address: F052AH, F052BH After reset: 0000H W

	15	14	13	12	11	10	9	8
UF0WTX	0	0	0	0	0	0	0	UF0WTX.8

7	6	5	4	3	2	1	0
UF0WTX.7	UF0WTX.6	UF0WTX.5	UF0WTX.4	UF0WTX.3	UF0WTX.2	UF0WTX.1	UF0WTX.0

Cautions 1. Writing to the UF0WTX register is prohibited other than when reception is switched to transmission (such as during transmission).

2. When the UF0WTX register is accessed in 8-bit units (when the UF0WTXB register is accessed), "0" is written to the UF0WTX.8 bit.

3. Writing to the UF0WTX register is prohibited when using the UF0BUF0 to UF0BUF8 registers.

Remark The UF0WTX.8 bit is an expansion bit when expansion bits are enabled (UF0EBE = UF0CL = 1).

(12) LIN-UART0 8-bit wait transmit data register (UF0WTXB)

The UF0WTXB register is an 8-bit register dedicated to delaying starting transmission until the stop bit of reception is completed during a LIN communication.

This register is write-only, in 8-bit units.

The stop bit length of reception when reception is switched to transmission is guaranteed for the UF0WTXB register.

See **14.5.11 Transmission start wait function** for details.

The UF0TXB register value will be read when the UF0WTXB register has been read.

Reset input sets this register to 00H.

Figure 14-12. Format of LIN-UART0 8-bit Wait Transmit Data Register (UF0WTXB)

Address: F052AH After reset: 00H W

	7	6	5	4	3	2	1	0
UF0WTXB	UF0WTX.7	UF0WTX.6	UF0WTX.5	UF0WTX.4	UF0WTX.3	UF0WTX.2	UF0WTX.1	UF0WTX.0

Cautions 1. Writing to the UF0WTXB register is prohibited other than when reception is switched to transmission (such as during transmission).

2. When the UF0WTXB register is accessed in 8-bit units (when the UF0WTXB register is accessed), "0" is written to the UF0WTX.8 bit of UF0WTX register.

3. Writing to the UF0WTXB register is prohibited when using the UF0BUF0 to UF0BUF8 registers.

(13) LIN-UART0 receive data register (UF0RX)

The UF0RX register is a 16-bit register that is used to store receive data.

Receive data of a character length specified by the UF0CL bit after reception completion will be stored into the UF0RX register when not in automatic baud rate mode (UF0MD1, UF0MD0 = 00B/10B) and when UF0EBE is “0”. When UF0EBE = UF0CL = 1, receive data of 9-bit length will be stored.

This register is read-only, in 16-bit units. When the UF0RX register is read in 8-bit units, it can be accessed as the UF0RX register.

Reset input sets this register to 0000H.

Figure 14-13. Format of LIN-UART0 Receive Data Register (UF0RX)

Address: F054AH, F054BH After reset: 0000H R

	15	14	13	12	11	10	9	8
UF0RX	0	0	0	0	0	0	0	UF0RX.8
	7	6	5	4	3	2	1	0
	UF0RX.7	UF0RX.6	UF0RX.5	UF0RX.4	UF0RX.3	UF0RX.2	UF0RX.1	UF0RX.0

When the data length is specified as 7 bits (UF0CL bit = 0):

- During LSB-first reception, receive data is transferred to bits 6 to 0 of the UF0RX register and the MSB always becomes “0”.
- During MSB-first reception, receive data is transferred to bits 7 to 1 of the UF0RX register and the LSB always becomes “0”.
- When an overrun error (UF0OVE = 1) has occurred, the receive data at that time will not be transferred to the UF0RX register.

Remark The UF0RX.8 bit is an expansion bit when expansion bits are enabled (UF0EBE = UF0CL = 1).

(14) LIN-UART0 8-bit receive data register (UF0RXB)

The UF0RXB register is an 8-bit register that is used to store receive data.

Receive data of a character length specified by the UF0CL bit after reception completion will be stored into the UF0RX register when not in automatic baud rate mode (UF0MD1, UF0MD0 = 00B/10B) and when UF0EBE is “0”.

This register is read-only, in 8-bit units.

Reset input sets this register to 00H.

Figure 14-14. Format of LIN-UART0 8-bit Receive Data Register (UF0RXB)

Address: F054AH After reset: 00H R

	7	6	5	4	3	2	1	0
UF0RXB	UF0RX.7	UF0RX.6	UF0RX.5	UF0RX.4	UF0RX.3	UF0RX.2	UF0RX.1	UF0RX.0

When the data length is specified as 7 bits (UF0CL bit = 0):

- During LSB-first reception, receive data is transferred to bits 6 to 0 of the UF0RX register and the MSB always becomes “0”.
- During MSB-first reception, receive data is transferred to bits 7 to 1 of the UF0RX register and the LSB always becomes “0”.
- When an overrun error (UF0OVE = 1) has occurred, the receive data at that time will not be transferred to the UF0RX register.

(15) LIN-UART0 ID setting register (UF0ID)

The UF0ID register is an 8-bit register that stored a PID that has been received when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B) and during a LIN communication. See **14.7 LIN Communication Automatic Baud Rate Mode** for details.

Also, when in normal UART mode (UF0MD1, UF0MD0 = 00B) and expansion bit data comparison is enabled (UF0CL = UF0EBE = UF0EBC = 1), the 8 bits (UF0RX7 to UF0RX0) of the receive data and the UF0ID register are compared upon a match between the received expansion bit and the expansion bit detection level (UF0EBL). See **14.8.3 Expansion bit mode reception (with data comparison)** for details.

Be sure to execute LIN communication by setting the reception enable bit (the UF0RXE bit of the UF0CTL0 register) to 0 when specifying a comparison value, and then setting the bit to 1.

This register can be read or written in 8-bit units.

Reset input sets this register to 00H.

Figure 14-15. Format of LIN-UART0 ID Setting Register (UF0ID)

Address: F052EH After reset: 00H R/W

	7	6	5	4	3	2	1	0
UF0ID	UF0ID.7	UF0ID.6	UF0ID.5	UF0ID.4	UF0ID.3	UF0ID.2	UF0ID.1	UF0ID.0

Caution Set to 00H before starting communication when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B). Writing is prohibited during communication operation in automatic baud rate mode.

(16) LIN-UART0 buffer registers 0 to 8 (UF0BUF0 to UF0BUF8)

The UF0BUF0 to UF0BUF8 registers are 8-bit buffer registers.

These registers can be used when transmitting data in normal UART mode (UF0MD1 and UF0MD0 = 00B) and when transmitting and receiving data in automatic baud rate mode (UF0MD1 and UF0MD0 = 11B).

When in normal UART mode (UF0MD1, UF0MD0 = 00B), data will be sequentially transmitted from the UF0BUF0 register by setting the UF0TRQ bit.

When in automatic baud rate mode (UF0MD1, UF0MD0 = 11B) and during response transmission (UF0TRQ = 1), the transmit data in UF0BUF0 will be transmitted sequentially, but the received data will not be stored.

When in automatic baud rate mode (UF0MD1, UF0MD0 = 11B) and during response reception (UF0RRQ = 1), the received data will be stored sequentially, starting from the UF0BUF0 register.

See **14.6.1 UART buffer mode transmission** and **14.7 LIN Communication Automatic Baud Rate Mode** for details.

These registers can be read or written in 8-bit units.

Reset input sets these registers to 00H.

Figure 14-16. Format of LIN-UART0 Buffer Registers 0 to 8 (UF0BUF0 to UF0BUF8)

Address: F052FH (UF0BUF0), F0520H (UF0BUF1), F0521H (UF0BUF2), F0522H (UF0BUF3), F0523H (UF0BUF4), F0524H (UF0BUF5), F0525H (UF0BUF5), F0526H (UF0BUF7), F0527H (UF0BUF8) After reset: 00H R/W

	7	6	5	4	3	2	1	0
UF0BUFm								

(m = 0 to 8)

Caution These registers cannot be used when expansion bits are enabled (UF0EBE = UF0CL = 1).

(17) LIN-UART0 buffer control register (UF0BUCTL)

The UF0BUCTL register is a 16-bit register that controls a buffer.

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

See **14.6.1 UART buffer mode transmission** and **14.7 LIN Communication Automatic Baud Rate Mode** for details.

Reset input sets this register to 0000H.

Figure 14-17. Format of LIN-UART0 Buffer Control Register (UF0BUCTL) (1/2)

Address: F0528H, F0529H After reset: 0000H R/W

	15	14	13	12	11	10	9	8
UF0BUCTL	0	0	0	0	0	0	UF0TW	UF0CON
	7	6	5	4	3	2	1	0
	UF0ECS	UF0NO	UF0RRQ	UF0TRQ	UF0BUL3	UF0BUL2	UF0BUL1	UF0BUL0

UF0TW	Transmission start wait bit
0	Starts transmission immediately when buffer data transmission is requested.
1	Delays starting of transmission until completion of stop bit of reception when buffer data transmission is requested.

The UF0TW bit is used to delay starting of transmission until completion of the stop bit of reception when transmitting buffer data in LIN communication. It can be set only in automatic baud rate mode (UF0MD1, UF0MD0 = 11B). See **14.5.11 Transmission start wait function** and **14.7 LIN Communication Automatic Baud Rate Mode** for details.

Cautions 1. Setting this bit is prohibited except when switching to response transmission after header reception.

2. The UF0TW bit becomes valid at the same time as the UF0TRQ bit is set (1).

UF0CON	Successive selection bit
0	The data group to be transmitted or received next is the last data group.
1	The data group to be transmitted or received next is not the last data group. (Data transmission or reception is continued without waiting for the next header to be received.)

The UF0CON bit indicates that the data group to be transmitted or received next is not the last data group when the multi-byte response transmission/reception function is used in LIN communication. It can be set only in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).

See **14.7.5 Multi-byte response transmission/reception function** for details.

Cautions 1. Setting this bit is prohibited except when the multi-byte transmission/reception function is used.

2. Set the UF0CON bit at the same time as setting UF0NO, UF0RRQ, and UF0TRQ for 16-bit access.

UF0ECS	Enhanced checksum selection bit
0	Classic checksum (used only for data byte calculation)
1	Enhanced checksum (used for calculating data byte + PID byte)

The UF0ECS bit is used to select how to handle checksum when the automatic checksum function is used in LIN communication. It is valid only when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B) and automatic checksum is enabled (UF0ACE = 1).

See **14.7.4 Automatic checksum function** for details.

Figure 14-17. Format of LIN-UART0 Buffer Control Register (UF0BUCTL) (2/2)

UF0NO	No-response request bit
0	Response for received PID is present.
1	Response for received PID is absent.

The UF0NO bit is used when a PID (PID received by a header) stored into the UF0ID register is excluded in automatic baud rate mode (UF0MD1, UF0MD0 = 11B). After setting the UF0NO bit, the bit will be cleared automatically when the next BF-SF reception is complete. It can be set only in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).

Caution Do not set the UF0TRQ and UF0RRQ bits while the UF0NO bit is "1". Simultaneous rewriting is prohibited.

UF0RRQ	Reception request bit
0	Storing has been started/no reception request
1	Reception start request/during receive operation in automatic baud rate mode

The UF0RRQ bit is used to request starting of storing data into a buffer. It is cleared when a reception completion interrupt for the buffer is generated. It can be set only in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).
See **14.7 LIN Communication Automatic Baud Rate Mode** for details.

Caution Do not set the UF0NO and UF0TRQ bits while the UF0RRQ bit is "1". Simultaneous rewriting is prohibited.

UF0TRQ	Transmission request bit
0	Storing has been started/no transmission request
1	Transmission start request/during transmit operation when using buffer

The UF0TRQ bit is used to request starting of transmitting buffer data. It is cleared when a transmission interrupt for the data prepared in the buffer is generated. It can be set only in normal UART mode (UF0MD1, UF0MD0 = 00B) or automatic baud rate mode (UF0MD1, UF0MD0 = 11B).
See **14.6.1 UART buffer mode transmission** and **14.7 LIN Communication Automatic Baud Rate Mode** for details.

Caution Do not set the UF0NO and UF0RRQ bits while the UF0TRQ bit is "1". Simultaneous rewriting is prohibited.

UF0BUL3 to UF0BUL0	Buffer length bits
0	Transmits or receives 9 bytes.
1 to 9	Transmits or receives number of bytes set.
10 to 15	Transmits or receives 9 bytes.

The UF0BUL3 to UF0BUL0 bits are used to set the number of transmit or receive data in a buffer. The read value is the pointer of the current buffer. The bits are valid only in normal UART mode (UF0MD1, UF0MD0 = 00B) or automatic baud rate mode (UF0MD1, UF0MD0 = 11B). When automatic checksum function is enabled, the checksum bits (one byte) need not be included in buffer length.
See **14.6.1 UART buffer mode transmission** and **14.7 LIN Communication Automatic Baud Rate Mode** for details.

(18) Port mode registers 5, X2 (PM5, PMX2)

The PM5 register is used to set ports 5 to input or output in 1-bit units.

When using the P50/INTP1/SI11/SDA11/LRxD0 pin for serial data input, set the PM5.0 bit to “1”. At this time, the output latches of P5.0 may be “0” or “1”.

When using the P51/INTP2/SO11/LTxD0 pin for serial data output, set the PM5.1 bit to “1”. Then, clear the PMX2 bit of PMX2 register to “0”.

Set the PM5 and PMX2 registers by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Remarks 1. The pins mounted depend on the product. See **1.3 Pin Configuration (Top View)** and **2.1 Pin Function List**.

2. See **CHAPTER 4 PORT FUNCTIONS** for port settings.

Figure 14-18. Format of Port Mode Register 5 (PM5)

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	1	1	PM5.5	PM5.4	PM5.3	PM5.2	PM5.1	PM5.0

PM5.n	P5n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 14-19. Format of Port Mode Register X2 (PMX2)

Address: F0506H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PMX2	1	1	1	1	1	1	1	PMX2

PMX2	P51/INTP2/SO11/LTxD0 pin alternate function selection
0	LTxD0
1	Other alternate function (including general-purpose I/O port)

14.4 Interrupt Request Signals

The following three interrupt request signals are generated from LIN-UART0.

- LIN-UART reception status interrupt (INTLS)
- LIN-UART reception interrupt (INTLR)
- LIN-UART transmission interrupt (INTLT)

Table 14-2 shows the default priority order of these three interrupt request signals.

Table 14-2. Interrupts and Their Default Priorities

Interrupt	Default Priority
Status	Low
Reception complete	|
Transmission start/complete	High

(1) LIN-UART reception status interrupt (INTLS)

LIN-UART reception status interrupt is generated when an error condition is detected during a reception. A UF0STR register flag (UF0PE, UF0FE, UF0OVE, UF0DCE, UF0BSF, UF0IPE, UF0CSE, UF0RPE, UF0IDM, UF0EBD) corresponding to the detected status is set.

See **14.5.10 LIN-UART reception status interrupt generation sources** for details.

(2) LIN-UART reception interrupt (INTLR)

LIN-UART reception interrupt is generated when data is shifted into the receive shift register and transferred to the UF0RX register in the reception enabled status.

When a reception error occurs, LIN-UART reception interrupt is not generated, but LIN-UART reception status interrupt is generated.

LIN-UART reception interrupt is not generated in the reception disabled status.

- If expansion bit operation is enabled (UF0CL = UF0EBE = 1) and expansion bit data comparison is disabled (UF0EBC = 0), LIN-UART reception interrupt is generated when the level of the inverted value set by using the expansion bit detection level select bit (UF0EBL) is detected as an expansion bit.
- When there is no error when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B) and PID reception has been completed (stop bit position), LIN-UART reception itnerrupt is generated.
- When response reception has ended without an error when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B), a reception complete interrupt request signal is generated.

(3) LIN-UART transmission interrupt (INTLT)

When a transmission interrupt request is set to output upon starting a transmission (UF0ITS = 0), a transmission interrupt request signal is generated when transmission from the UF0TX register to the transmit shift register has been completed.

When a transmission interrupt request is set to output upon completion of a transmission (UF0ITS = 1), a transmission interrupt request signal is generated when transmitting a stop bit has been completed.

- When in automatic baud rate mode (UF0MD1, UF0MD0 = 11B), a transmission complete interrupt request signal is generated at the start of transmission of the last byte of a response.

14.5 Operation

14.5.1 Data format

Full-duplex serial data reception and transmission is performed.

As shown in Figure 14-20, one data frame of transmit/receive data consists of a start bit, character bits, an expansion bit, a parity bit, and stop bits.

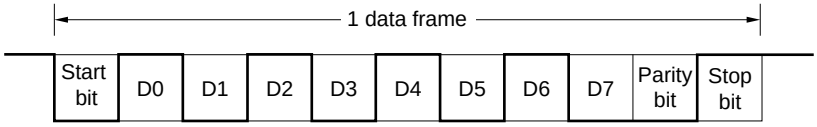
Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB/LSB-first transfer are performed using the UF0CTL0 register.

Moreover, the UF0TDL bit and UF0RDL bit of the UF0OPT0 register are used to control UART output/inverted output for the LTxD0 pin and UART input/inverted input for the LRxD0 pin, respectively.

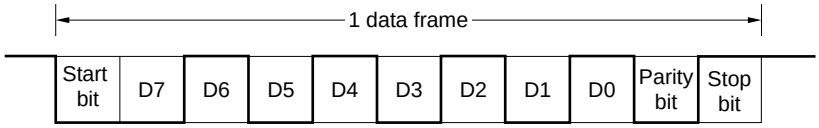
- Start bit.....1 bit
- Character bits.....7 bits/8 bits
- Expansion bit .....1 bit
- Parity bit.....Even parity/odd parity/0 parity/no parity
- Stop bit.....1 bit/2 bits
- Transmission/reception level setting .....Forward/inversion
- Transmission/reception direction setting .....Forward/nversion

Figure 14-20. Format of LIN-UART Transmit/Receive Data (1/2)

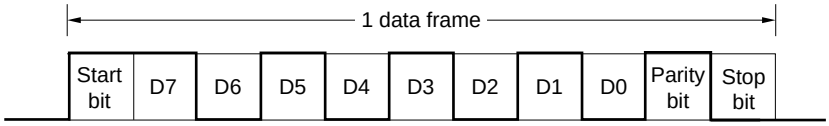
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, LTxD0 inversion



(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H

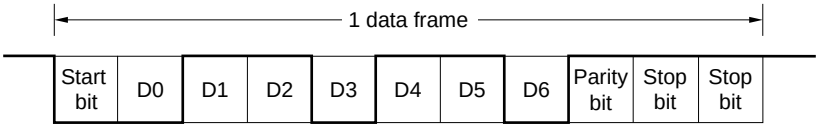
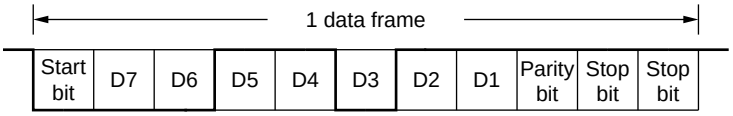
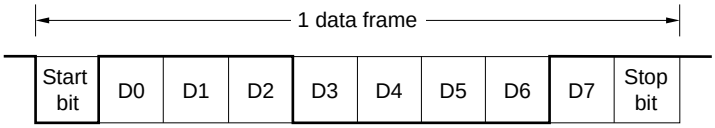


Figure 14-20. Format of LIN-UART Transmit/Receive Data (2/2)

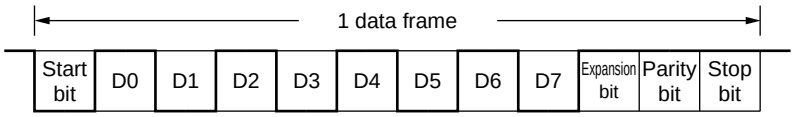
(e) 7-bit data length, MSB first, odd parity, 2 stop bits, transfer data: 36H



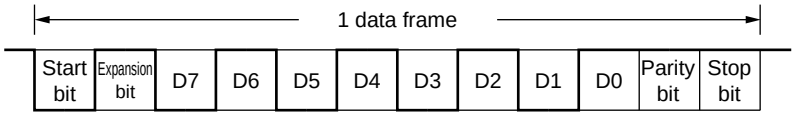
(f) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H



(g) 8-bit data length, LSB first, even parity, expansion bit: enabled, 1 stop bit, transfer data: 155H



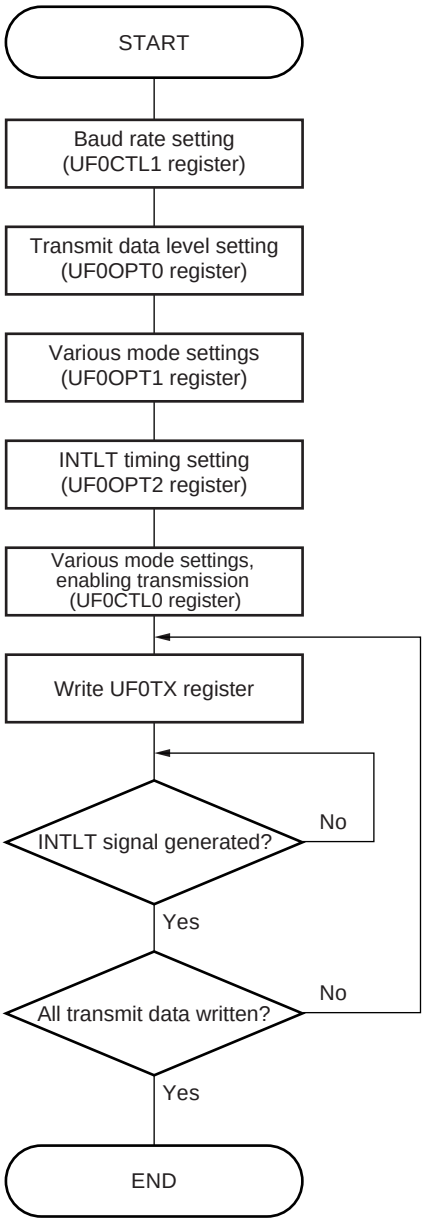
(h) 8-bit data length, MSB first, even parity, expansion bit: enabled, 1 stop bit, transfer data: 155H



14.5.2 Data transmission

Figure 14-21 shows the procedure for transmitting data.

Figure 14-21. Transmission Processing Flow



Cautions 1. When initializing (UF0TXE = 0) the transmission unit, be sure to confirm that the transmission status flag has been reset (UF0TSF = 0). When initialization is performed while UF0TSF is "1", transmission is aborted midway.

2. During LIN communication, confirm that a status interrupt request signal (INTLS) has been generated, because reception is performed simultaneously with transmission.

3. When data consistency error detection has been set (UF0DCS = 1) and a data consistency error has been detected during LIN communication, transmission of the next data frame or BF is stopped at the same as when a status interrupt request signal (INTLS) is generated and a data consistency error flag is set (UF0DCE = 1).

Remark See **(2)** of **14.11 Cautions on Use for details of starting LIN-UART**.

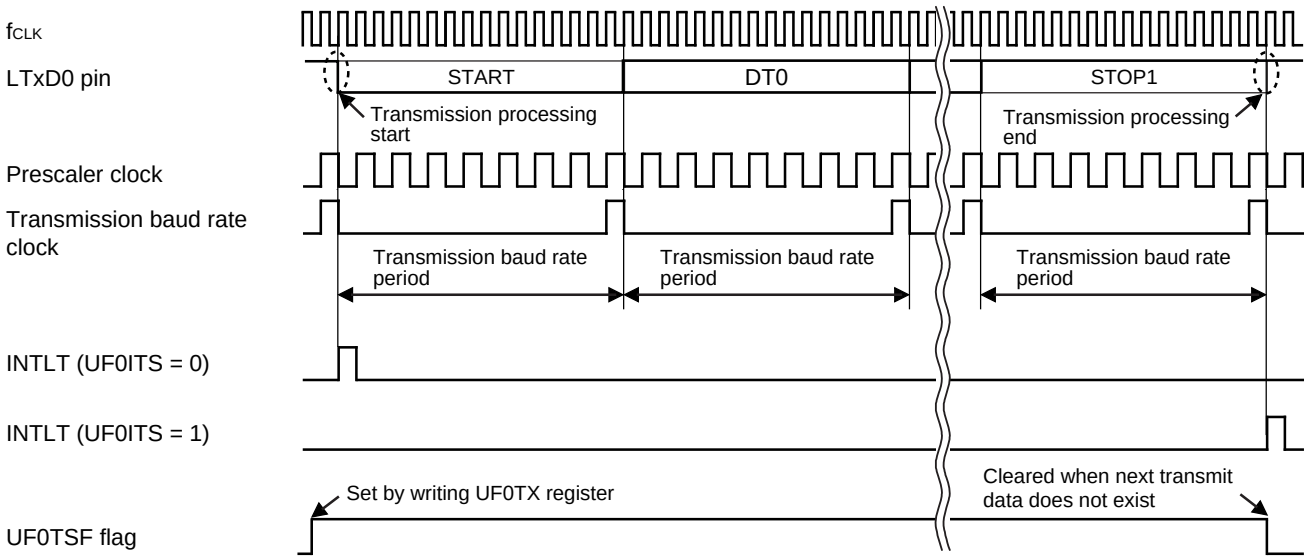
A transmission operation is started by writing transmit data to the transmit data register (UF0TX).

The data stored into the UF0TX register is transferred to the transmit shift register and a start bit, an expansion bit, a parity bit, and stop bits are added to the data, and the data are sequentially output from the LTxD0 pin.

If a transmission interrupt is set upon starting a transmission (UF0ITS = 0), a transmission interrupt request signal (INTLT) is generated when transferring the data stored into the UF0TX register to the transmit shift register has been completed.

If a transmission interrupt is set upon completion of a transmission (UF0ITS = 1), a transmission interrupt request signal (INTLT) is generated when transmitting a stop bit has been completed.

Figure 14-22. Data Transmission Timing Chart



Caution If the stop bit length is set to 2 bits (UF0SL = 1), the transmit completion interrupt (INTLT) will be output after the second stop bit has been transmitted, at which point the transmission status flag (UF0TSF) will be cleared.

When generation of a transmission interrupt is set upon starting a transmission (UF0ITS = 0), successive transmission can be performed by writing the next data to UF0TX during the transmission after INTLT has been generated.

Figure 14-23. Diagram of Timing When Starting Successive Transmission (UF0ITS = 0)

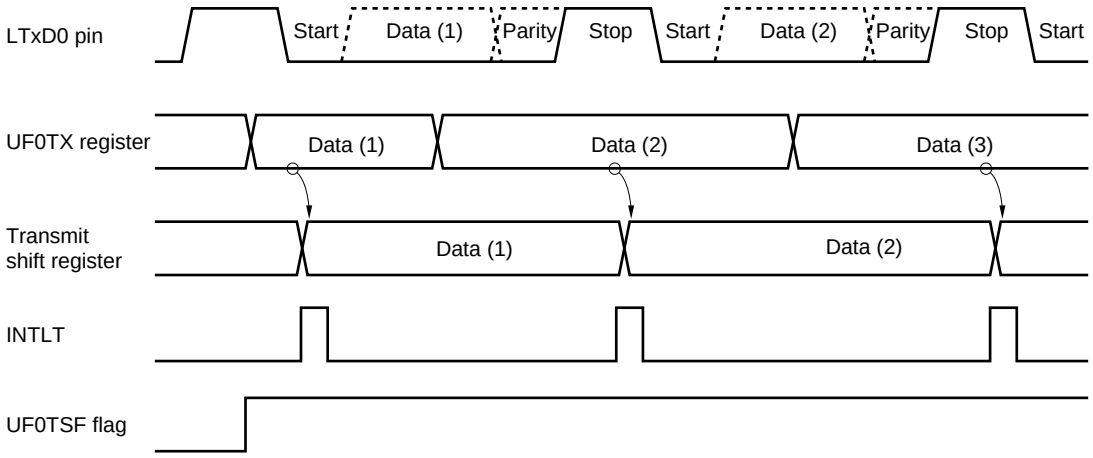
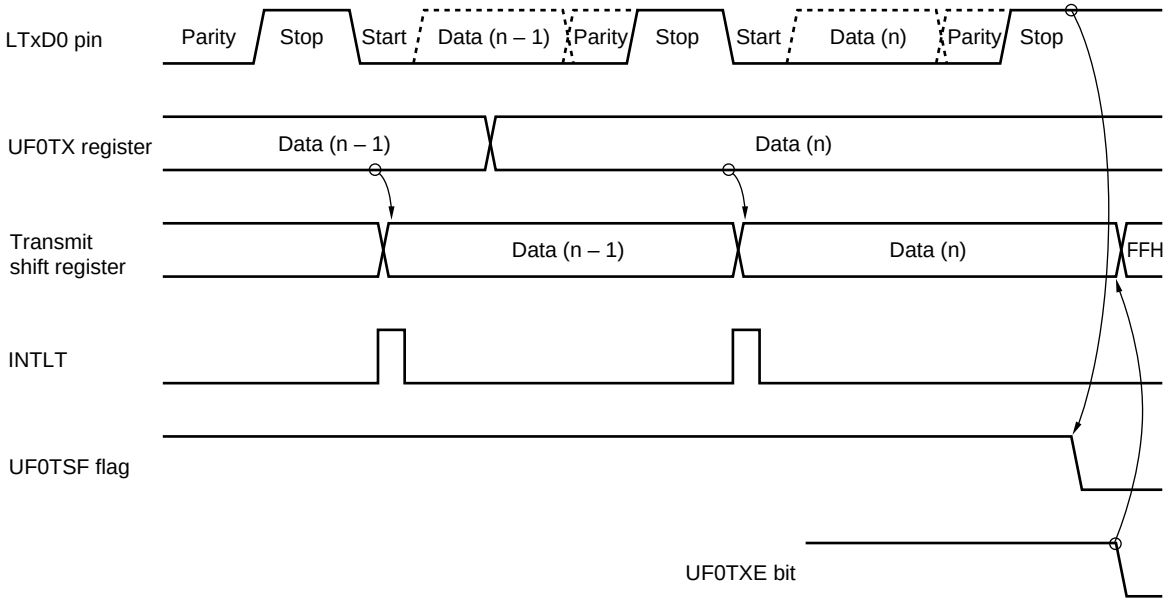


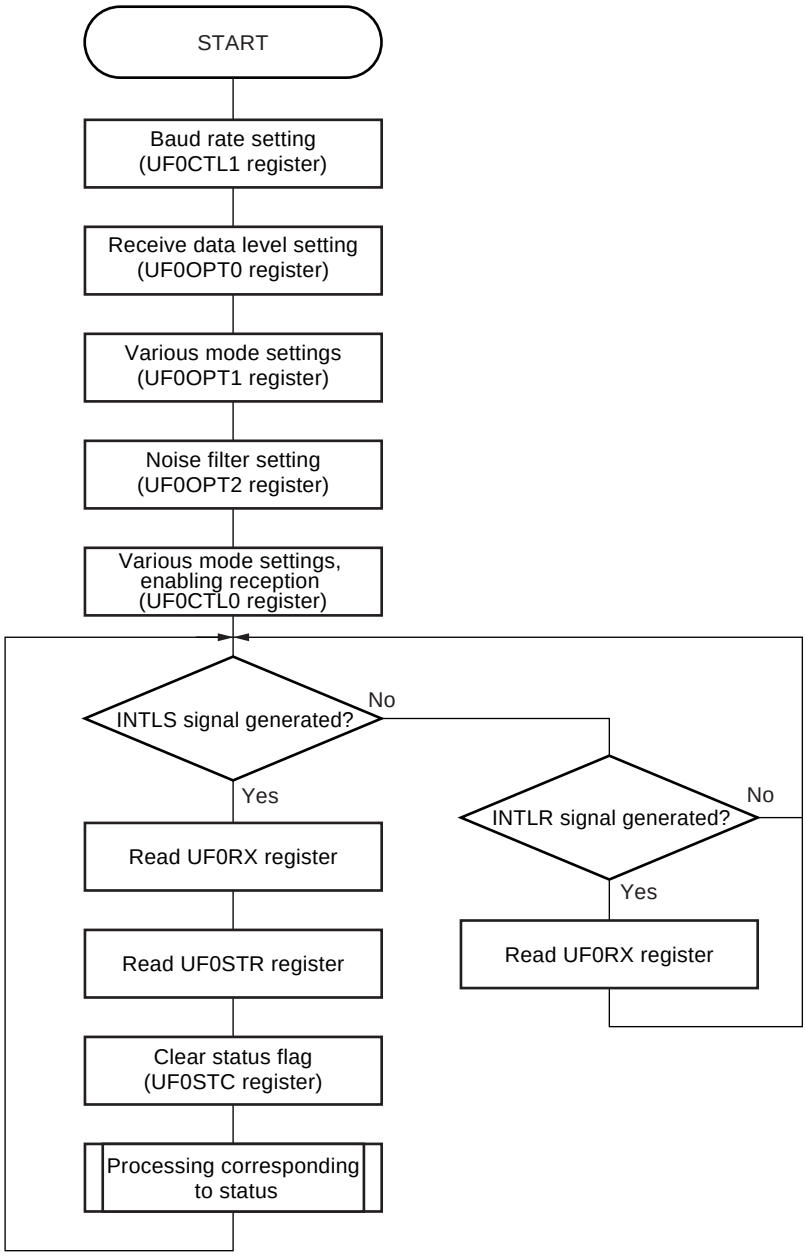
Figure 14-24. Diagram of Timing When Ending Successive Transmission (UF0ITS = 0)



14.5.3 Data reception

Figure 14-25 shows the procedure for receiving data.

Figure 14-25. Reception Processing Flow



Cautions 1. When initializing (UF0RXE = 0) the reception unit, be sure to confirm that the reception status flag has been reset (UF0RSF = 0). When initialization is performed while UF0RSF is "1", reception is aborted midway.

2. Be sure to read the receive data register (UF0RX) when a reception error has occurred. If the UF0RX register is not read, an overrun error occurs upon completion of receiving the next data.

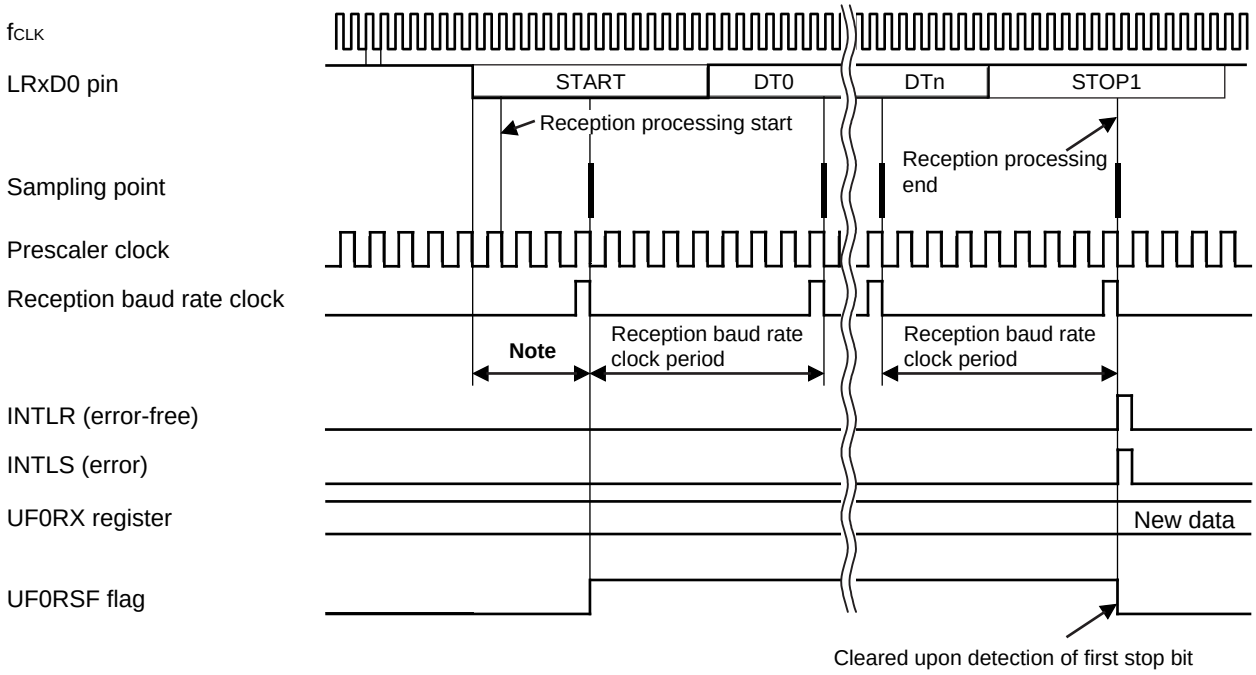
Remark See **(2)** of **14.11 Cautions on Use for details of starting LIN-UART**.

When the LRxD0 pin is sampled by using the operating clock and a falling edge is detected, data sampling of the LRxD0 pin is started and is recognized as a start bit if it is at low level at a timing of half the reception baud rate clock period after the falling edge has been detected. When the start bit has been recognized, a reception operation is started and serial data is sequentially stored into the receive shift register according to the baud rate set. When a stop bit has been received, the data stored into the receive shift register is transferred to the receive data register (UF0RX) at the same time a reception complete interrupt request signal (INTLR) is generated.

When an overrun error has occurred (UF0OVE = 1), however, the receive data is not transferred to the UF0RX register but discarded. When any other error has occurred, the reception is continued up to the reception position of the stop bit and the receive data is transferred to the UF0RX register.

After the occurrence of any reception error, INTLS is generated after completion of the reception and INTLR is not generated.

Figure 14-26. Data Reception Timing Chart



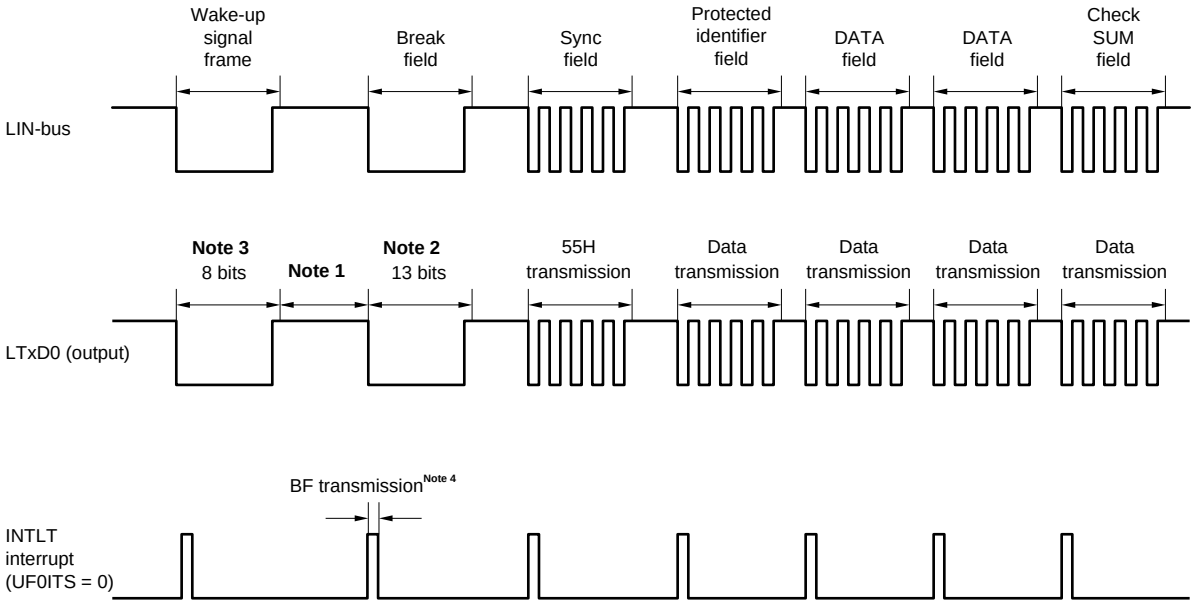
Note One-half the reception baud rate clock period

Cautions
1. **The start bit is not recognized when a high level is detected at a timing of half the reception baud rate clock period after the falling edge of the LRxD0 pin was detected.**
2. **A reception always operates with the number of stop bits as 1. At that time, the second stop bit is ignored.**
3. **When a low level is constantly input to the LRxD0 pin before an operation to enable reception is performed, the receive data is not identified as a start bit.**
4. **For successive reception, the next start bit can be detected immediately after a stop bit of the first receive data has been detected (upon generation of a reception complete interrupt).**
5. **Be sure to enable reception (UF0RXE = 1) after having changed the UF0RDL bit. If the UF0RDL bit is changed after having enabled reception, the start bit may be detected falsely.**

14.5.4 BF transmission/reception format

The RL78/F12 has a BF (Break Field) transmission/reception control function to enable use of the LIN (Local Interconnect Network) function.

Figure 14-27. LIN Transmission Manipulation Outline



Cautions 1. The interval between each field is controlled by software.

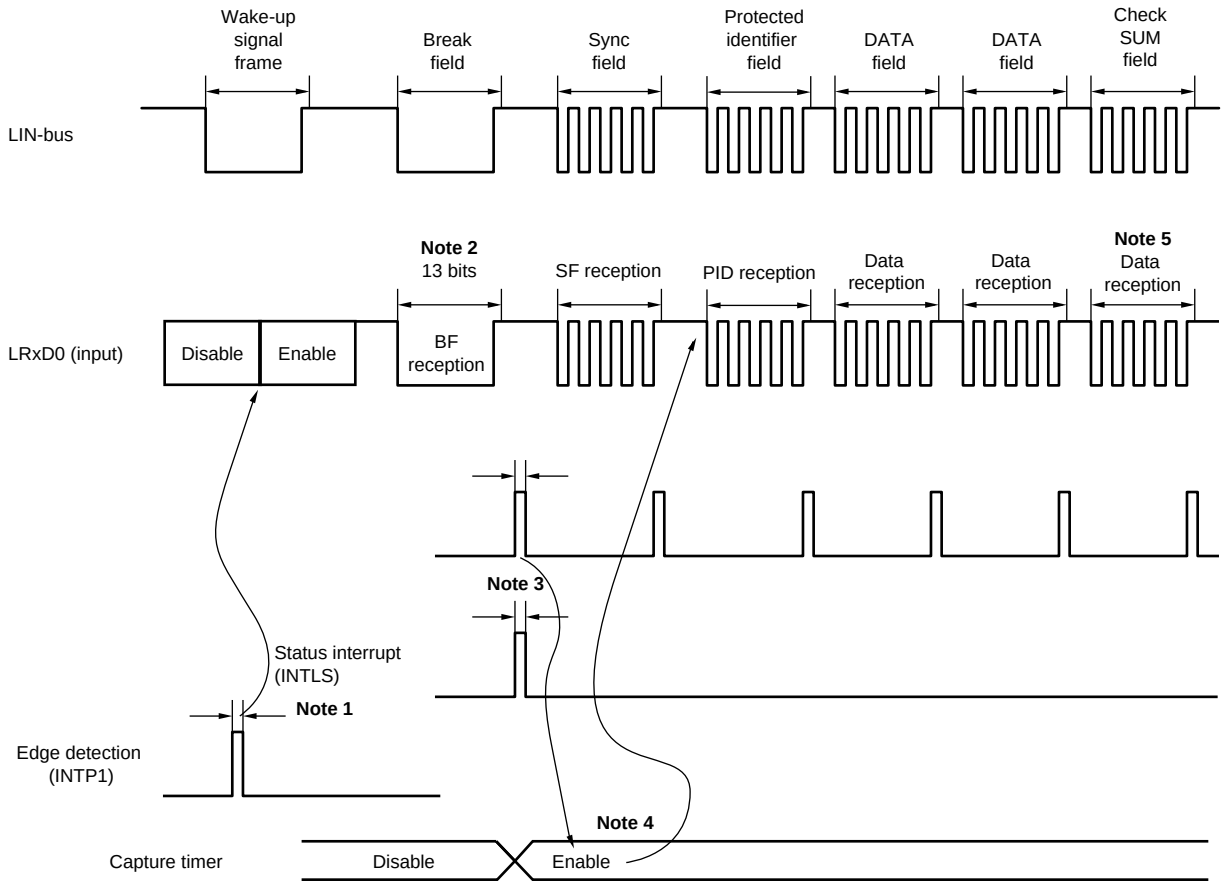
2. BF output is performed by hardware. The output width is the bit length set by the UF0BLS2 to UF0BLS0 bits of the UF0OPT0 register. If even finer output width adjustments are required, such adjustments can be performed using the UF0BRS11 to UF0BRS0 bits of the UF0CTL1 register.

3. 80H transfer in the 8-bit mode or BF transmission is substituted for the wakeup signal.

4. The LIN-UART transmission interrupt (INTLT) is output at the start of each transmission. The INTLT signal is also output at the start of each BF transmission. Be sure to clear UF0OPT2.UF0ITS to "0" when starting a transmission, so that the LIN-UART transmission interrupt is always generated.

Remark Figure 14-27 shows the LIN transmission manipulation outline when in BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B). See **14.7 LIN Communication Automatic Baud Rate Mode** for when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).

Figure 14-28. LIN Reception Manipulation Outline



Cautions 1. A wakeup signal is detected by detecting the interrupt edge of a pin (INTP1). After having received the wakeup signal, enable LIN-UART0, enable reception operation, and set the BF reception trigger bit if needed.

2. If a BF reception of at least 11 bits is detected, the BF reception is judged to be ended normally.

3. When BF reception has ended normally in normal UART mode (UF0MD1, UF0MD0 = 00B), a reception complete interrupt request signal (INTLR) is generated. When BF reception has ended normally in BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B), a status interrupt request signal (INTLS) is generated, and a successful BF reception flag (UF0BSF) is set. When the BF reception flag (UF0BRF) is "1", detection of overrun, parity, and framing errors (UF0OVE, UF0PE, UF0FE) is not performed during BF reception. Moreover, data transfer from the receive shift register to the receive data register (UF0RX) is also not performed. At this time, UF0RX retains the previous value.

4. Connect the LRxD0 pin to the TI (capture input) of the timer array unit. Enable the timer by using a BF reception complete interrupt, measure the baud rate from the SF transfer data, and calculate the baud rate error. Set a reception state by stopping the LIN-UART0 reception operation after SF reception and re-setting the value of LIN-UART0 control register 1 (UF0CTL1) obtained by correcting the baud rate error.

5. Classification of a checksum field is performed by using software. The processing that initializes LIN-UART0 after CSF reception and sets to a successful BF reception wait state (UF0BRF = 1) again is also performed by using software. In BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B), however, BF reception can be automatically performed without setting to a successful BF reception wait state (UF0BRF = 1) again.

(Caution 6 and Remark are given on the next page.)

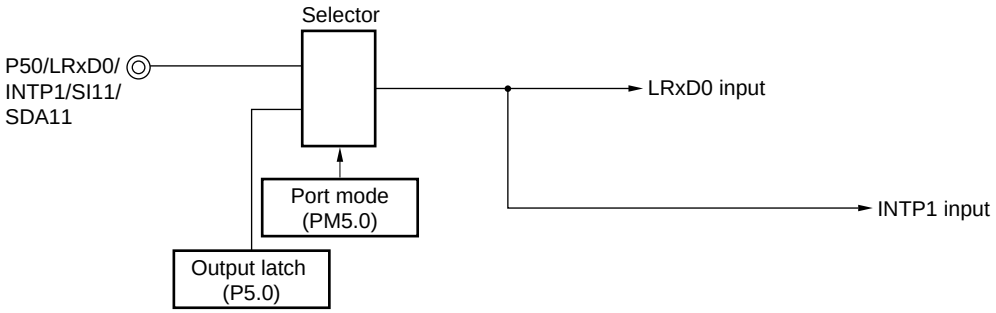
6. With the sync field, the transfer baud rate is calculated using the capture function of the TAU. At this point, stop reception operation to stop generation of a reception interrupt in the LIN-UART0.

Remark See **14.7 LIN Communication Automatic Baud Rate Mode** for when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).

Figure 14-29 shows the port configurations for LIN reception manipulation.

Wakeup signals transmitted from the LIN master are received via INTP1 edge detection. The baud rate error can be calculated by measuring the length of a sync field transmitted from the LIN master via an external event capture operation of the timer array unit (TAU).

Figure 14-29. Port Configuration of LIN Reception Manipulation



A summary of the peripheral functions to be used in LIN communication operation is given below.

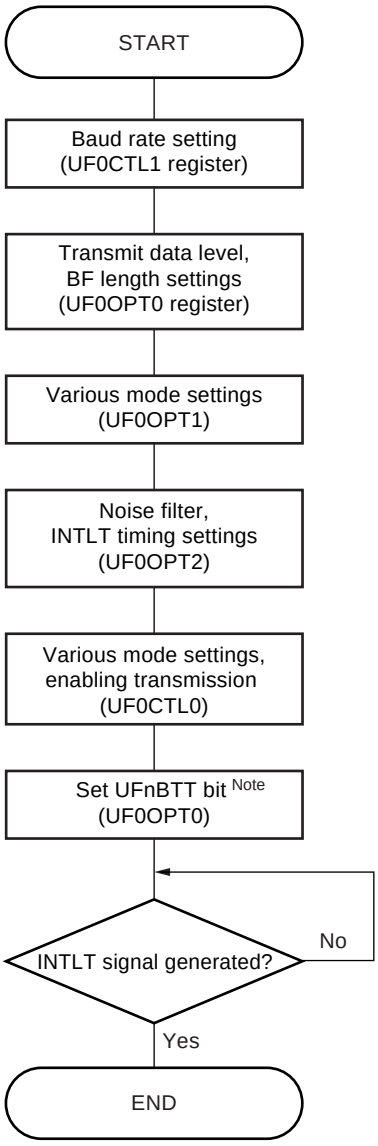
<Peripheral functions to be used>

- LIN-UART0 reception pin interrupt (INTP1); Wakeup signal detection
  Purpose: Detecting wakeup signal edges and detecting the start of communication
- Timer input of the timer array unit (TAU); Baud rate error detection
  Purpose: Detecting the length of a sync field (SF) and detecting the baud rate error by dividing the sync field length by the number of bits (measuring the intervals between the timer input edges in capture mode)
- Asynchronous serial interface LIN-UART0

14.5.5 BF transmission

Figure 14-30 describes the processing of BF transmission in LIN communication.

Figure 14-30. BF Transmission Processing Flow



<R> **Note** **In normal UART mode (UF0MD1, UF0MD0 = 00B), set the UF0BRT bit at the same time as setting the UFnBTT bit.**

Caution **Set the following values when performing BF transmission.**

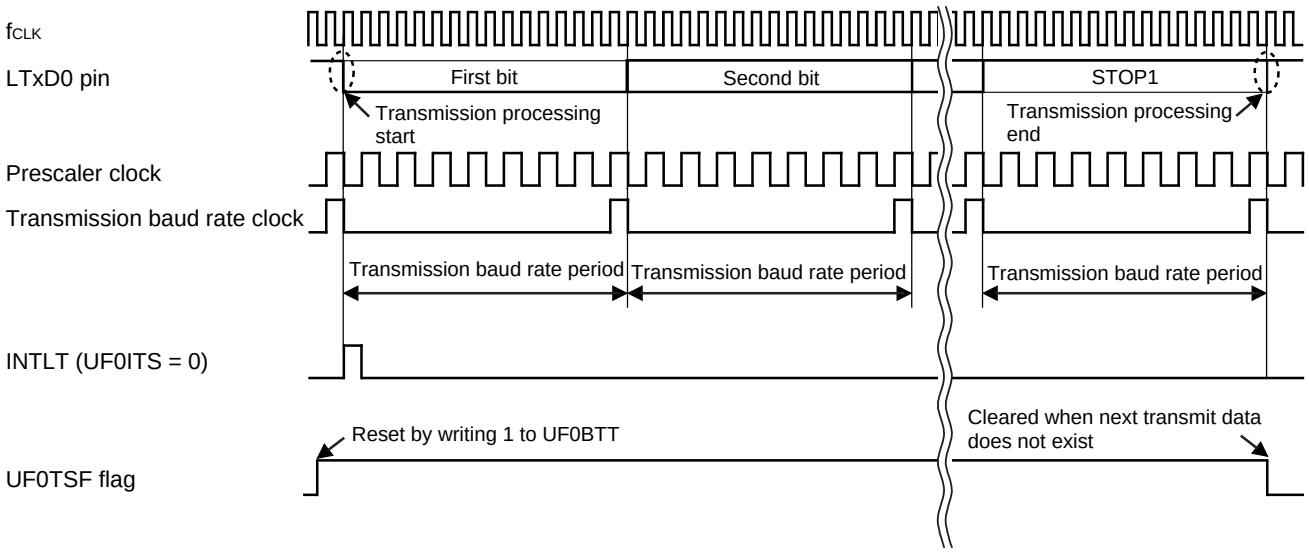
The transmit data level is normal output (UF0TDL = 0).
Communication direction control is LSB first (UF0DIR = 1).
The parity selection bit is no parity bit output (UF0PS1, UF0PS0 = 00B).
The data character length is 8 bits (UF0CL = 1).
The LIN-UART transmission interrupt is generated when starting transmission (UF0ITS = 0).

Remark See **(2)** of **14.11 Cautions on Use for details of starting LIN-UART**.

A BF transmission operation is started when a BF transmission trigger (UF0BTT) is set. 13 to 20 bits of low level (the length specified by the BF length selection bits (UF0BLS2 to UF0BLS0)) is output to the LTxD0 pin. LIN-UART transmission interrupt (INTLT) is generated when the BF transmission is started. After the BF transmission ends, the BF transmission state is automatically released and operation is returned to normal UART transmission mode.

The transmission operation stays in a wait state until the data to be transmitted is written to the UF0TX register or a BF transmission trigger (UF0BTT) is set. Start the next transmission operation after having confirmed that the BF has been received normally according to the LIN-UART reception interrupt (INTLR) during the BF transmission or the LIN-UART reception status interrupt (INTLS).

Figure 14-31. BF Transmission Timing Example

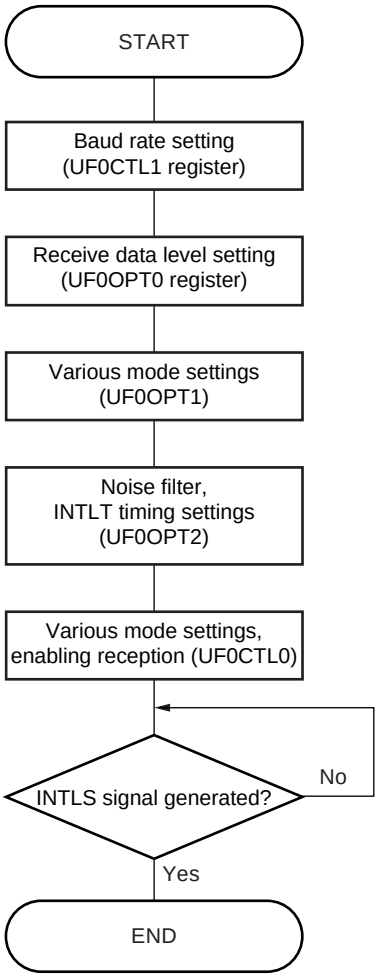


Caution When the stop bit length is set to 2 bits (UF0SL = 1), the transmission status flag (UF0TSF) is cleared when transmission of the second stop bit has been completed.

14.5.6 BF reception

Figure 14-32 describes the processing of BF reception in LIN communication.

Figure 14-32. BF Reception Processing Flow



Caution Set the following values when performing BF transmission.
The input logic level is normal input (UF0RDL = 0).
Communication direction control is LSB first (UF0DIR = 1).
The parity selection bit is no parity bit output (UF0PS1, UF0PS0 = 00B).
The data character length is 8 bits (UF0CL = 1).
Transmission interrupt is generated when starting transmission (UF0ITS = 0).
BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B) as the mode.

Remarks 1. Figure 14-32 shows the reception processing flow of LIN communication in BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B).
See **14.7 LIN Communication Automatic Baud Rate Mode** for when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B).

2. See **(2)** of **14.11 Cautions on Use for details of starting LIN-UART**.

When the BF reception trigger bit (UF0BRT) is set, a successful BF reception wait state (UF0BRF = 1) is entered, the LRxD0 input level is monitored, and start bit detection is performed.

When the falling edge of the LRxD0 input is detected, the BF length is measured by counting up the internal counter until a rising edge is detected. If the BF length is 11 bits or more when a rising edge is detected, BF reception is judged as being normal, and BF reception ends. When ending BF reception, a successful BF reception flag (UF0BSF) is set at the same time as generation of the LIN-UART reception status interrupt (INTLS).

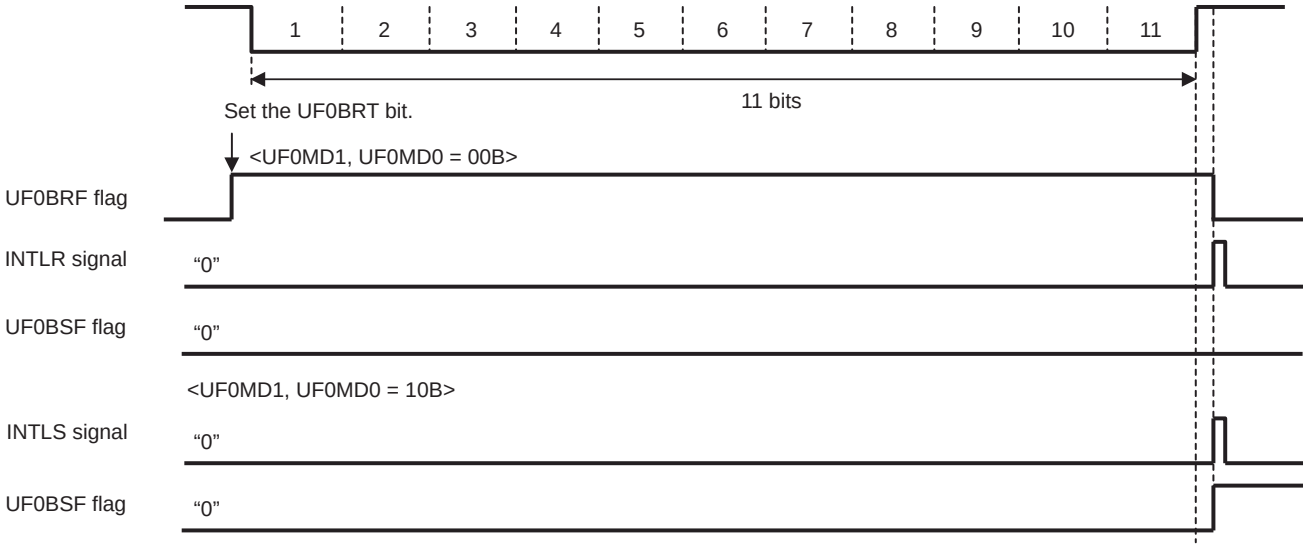
In automatic baud rate mode, detection of overrun, parity, and framing errors (UF0OVE, UF0PE, UF0FE) is limited. Moreover, data transfer from the receive shift register to the receive data register (UF0RX) is not performed. BF reception is judged as being abnormal if the BF width is less than 11 bits. In that case, the error status flag (UFnSTR) is set at the same time as generation of the status interrupt request signal (INTLSn).

When performing a transmission for which a data consistency check is enabled (UF0DCS = 1), a data consistency error flag (UF0DCE) is set and LIN-UART reception status interrupt (INTLS) is output when a mismatch between the transmit data and receive data is detected, regardless of whether BF reception is performed successfully or fails. At that time, INTLR is not output.

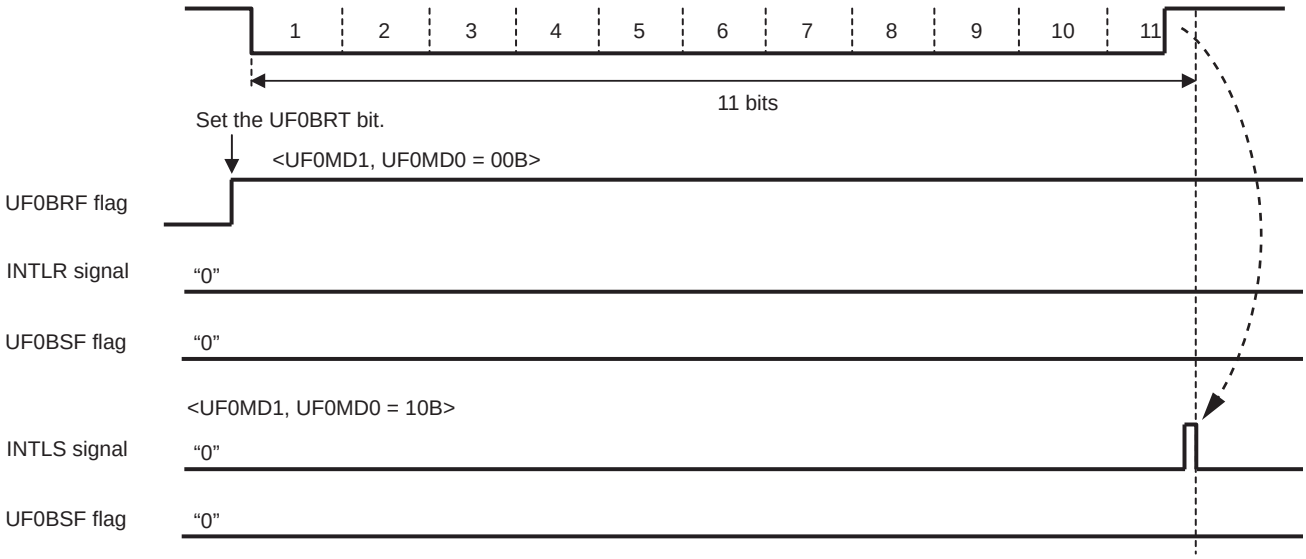
When in BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B), LIN-UART can detect a new BF reception even during data communication or in automatic baud rate mode. See **14.5.9 (2) BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B)** for details.

Figure 14-33. BF Reception Timing Example

- Normal BF reception: A high level is detected after the BF length has exceeded 11 bits.



- BF reception error: A high level is detected when the BF length is less than 11 bits.



Caution The UF0BRF bit is reset by setting the UF0BRT bit to "1" and cleared upon normal BF reception. In BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B), the bit is reset or cleared in the same way as described above.

14.5.7 Parity types and operations

Caution When using the LIN communication, fix the UF0PS1 and UF0PS0 bits of the UF0CTL0 register to 00 (n = 0, 1).

The parity bit is used to detect bit errors in the communication data. Normally the same parity bit is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect 1-bit (odd-count) errors. In the case of 0 parity and no parity, errors cannot be detected.

(1) Even parity

(a) During transmission

The number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 1
- Even number of bits whose value is “1” among transmit data: 0

(b) During reception

The number of bits whose value is “1” among the reception data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(2) Odd parity

(a) During transmission

Opposite to even parity, the number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 0
- Even number of bits whose value is “1” among transmit data: 1

(b) During reception

The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(3) 0 parity

During transmission, the parity bit is always made 0, regardless of the transmit data.

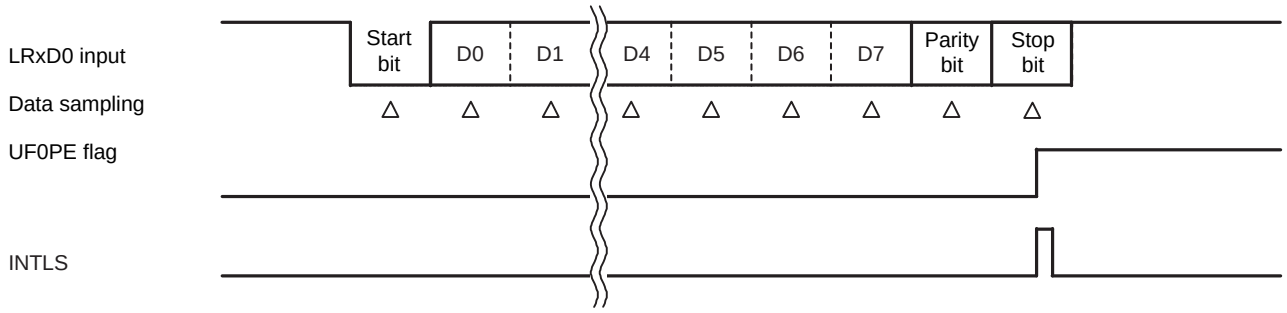
During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(4) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

Figure 14-34. Parity Error Occurrence Timing



14.5.8 Data consistency check

When the data consistency check selection bit (UF0DCS) is set to “1”, transmit data and receive data are compared during transmission operation, even if the reception enable bit is disabled (UF0RXE = 0).

When reception is enabled (UF0RXE = 1), it is also checked that reception processing is not ended early during transmission processing.

When either a mismatch between transmission and reception signals or an early end of reception processing is detected during transmission processing, operation is judged as being abnormal, a status interrupt request signal (INTLS) is output, and a data consistency error flag (UF0DCE) is set. Even if the next transmit data has already been written to the transmit data register (UF0TX), the next transmission is not performed. (The written data within UF0TX is ignored.) When the BF transmission trigger bit (UF0BTT) has been set, a BF is not transmitted.

To restart transmission, transmit data must be written to the transmit data register (UF0TX) or the BF transmission trigger bit (UF0BTT) must be set, after the end of transmission has been confirmed (UF0TSF = 0) and the data consistency error flag (UF0DCE) has been cleared or the UF0EN bit of the PERX register has been cleared and then set. When a buffer is used, communication is stopped even if data not transferred remains in the buffer.

When reception is disabled (UF0RXE = 0), storing receive data and thereby generating LIN-UART reception interrupt (INTLR) as well as setting UF0BSF, UF0FE, and UF0OVE and thereby generating LIN-UART reception status interrupt (INTLS) are not performed since the reception operation itself is not performed. Consequently, receive data is not required to be read.

Caution A store operation of receive data is not affected by whether a data consistency error exists. Storing is performed even if a consistency error occurs.

(1) Mismatch between transmission and reception signals

Serial transmission and reception signals are compared during data (or BF) transmission, a detected mismatch is judged as being abnormal, and the UF0DCE bit is set (1) at the same time a status interrupt (INTLS) is generated.

During data transmission, the comparison is performed from the start bit to the first stop bit.

During BF transmission, the comparison is performed from the first bit of the BF to the first stop bit.

A consistency check is not performed for the second stop bit, even if the stop bit length is specified as two bits by using the stop bit length select bit (UF0SL).

Figure 14-35. Data Consistency Error Occurrence Timing Example 1 (UF0BRF = 0)

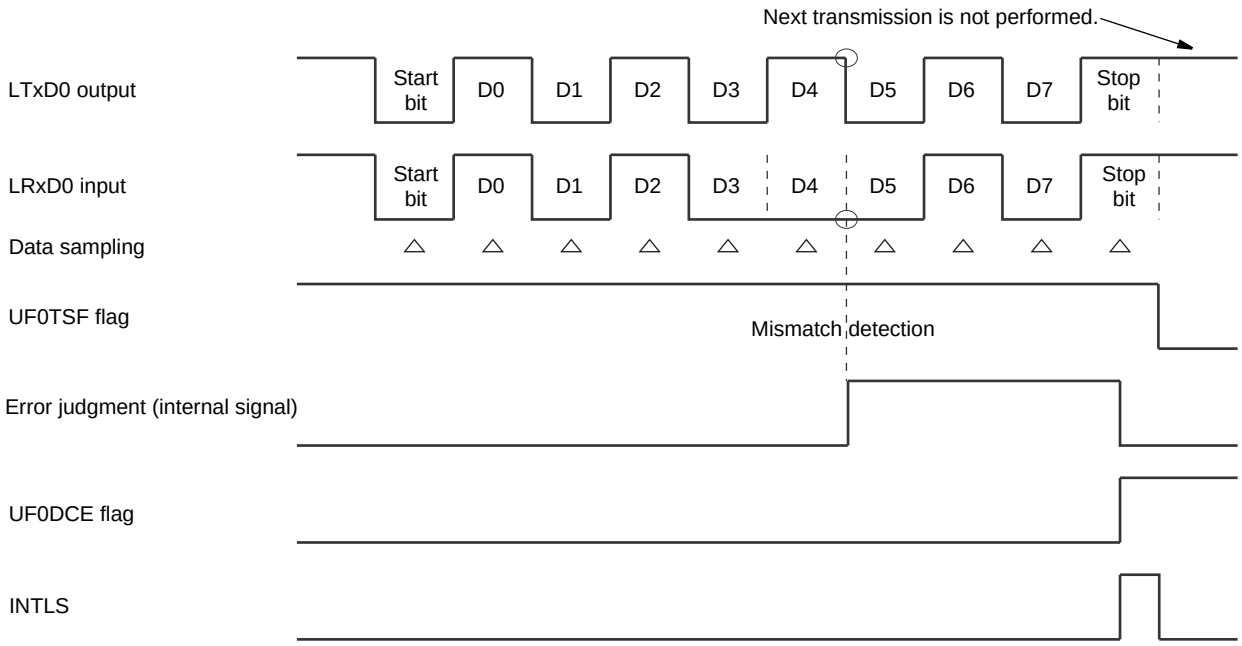
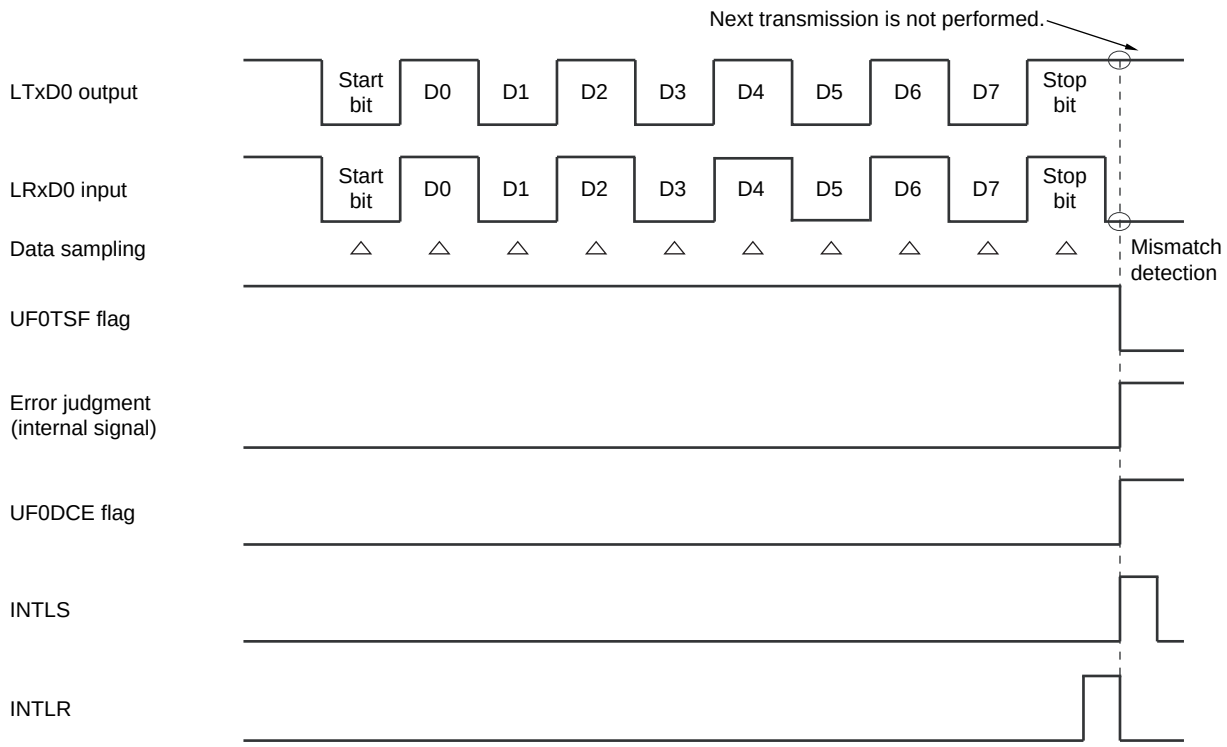


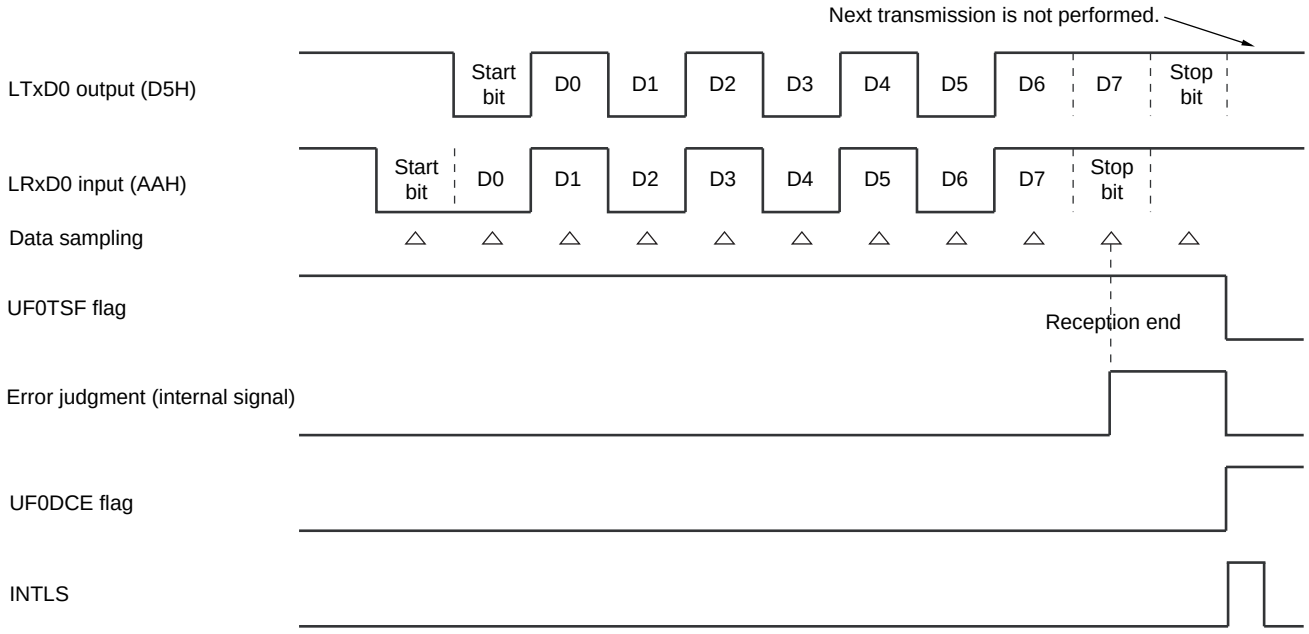
Figure 14-36. Data Consistency Error Occurrence Timing Example 2 (UF0BRF = 0)



(2) Early end of reception processing

When transmission is performed while reception is enabled (UF0TXE = UF0RXE = 1), a stop bit position detected in the reception processing, even though during transmission is judged as being abnormal and the UF0DCE bit is set (1) at the same time a status interrupt (INTLS) is generated.

Figure 14-37. Timing Example of Consistency Error Occurrence due to Early End of Reception Processing



14.5.9 BF reception mode select function

A mode for BF (break field) reception, which can be selected by using the LIN-UART operation mode selection bits (UF0MD1, UF0MD0), is provided.

(1) Normal UART mode (UF0MD1 and UF0MD0 = 00B)

In normal UART mode (UF0MD1 and UF0MD0 = 00B), a new BF is only recognized when the system is waiting for a BF to be successfully received (UF0BRF = 1). When BF reception has been successfully completed, a reception complete interrupt (INTLR) is generated.

If the system is not waiting for a BF to be successfully received (UF0BRF = 0), framing or overrun errors are detected at the data's stop bit position (bit 10) (see Figure 14-38). If an overrun error has not occurred, the received data is stored in the UF0RX register. If the system is waiting for a BF to be successfully received (UF0BRF = 1), framing or overrun errors are not detected and the received data is not stored in the UF0RX register.

If UF0BRF = 0 and reception is stopped when data or the BF stop bit is transmitted, the data consistency error interrupt is issued and the flag is changed when transmission of the bit following the stop bit starts (see 14.5.8 (2)).

If reception is in progress when the stop bit is transmitted, the data consistency error interrupt is issued and the flag is changed when transmission of the stop bit starts (see 14.5.8 (1)). On the other hand, if UF0BRF = 1 and reception is stopped when the stop bit is transmitted, the data consistency error interrupt is issued and the flag is changed when transmission of the bit following the stop bit starts (see Figure 14-39) and if reception is in progress when the stop bit is transmitted, the data consistency error interrupt is issued and the flag is changed when the rising edge of the input data following the stop bit is detected (see Figure 14-40).

Caution The successful BF reception flag (UF0BSF) is not set in normal UART mode.

Figure 14-38. Timing of Judging Framing or Overrun Error in Normal UART Mode

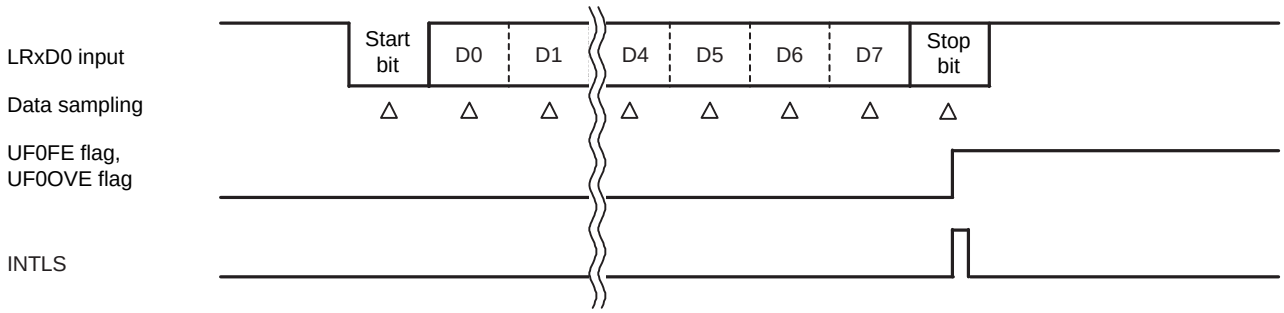


Figure 14-39. Timing of Occurrence of Data Consistency Error When BF Is Transmitted When UF0BRF = 1 (When Reception Is in Progress After Transmission of Stop Bit Has Stopped (Previous Input Data = 1))

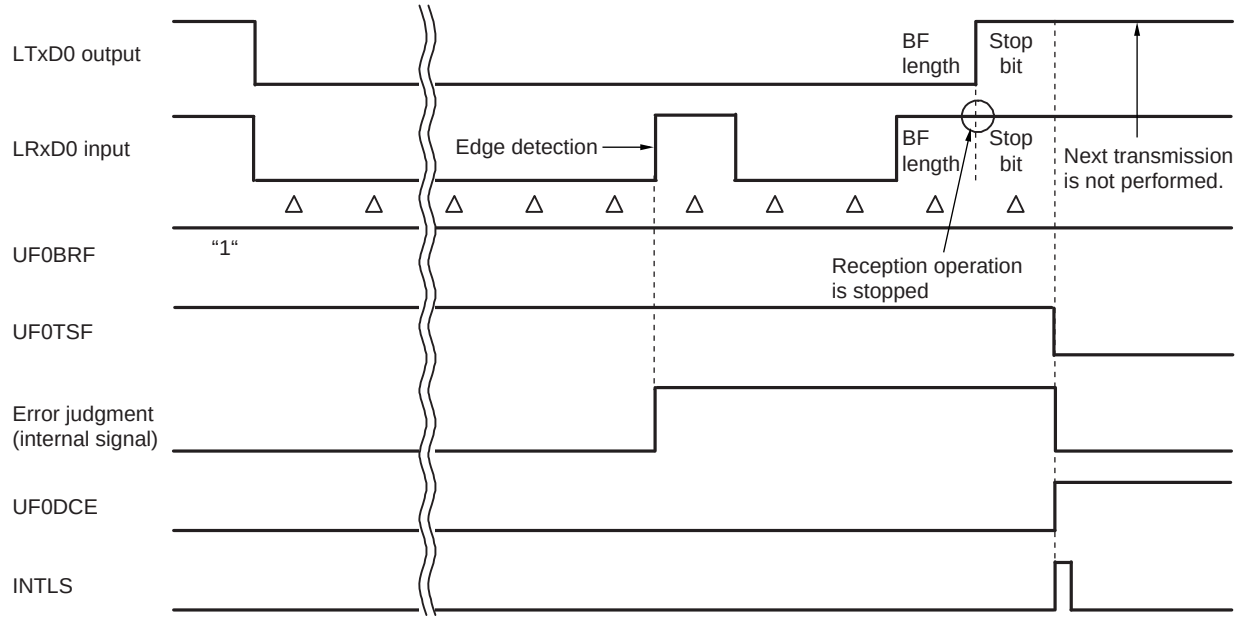
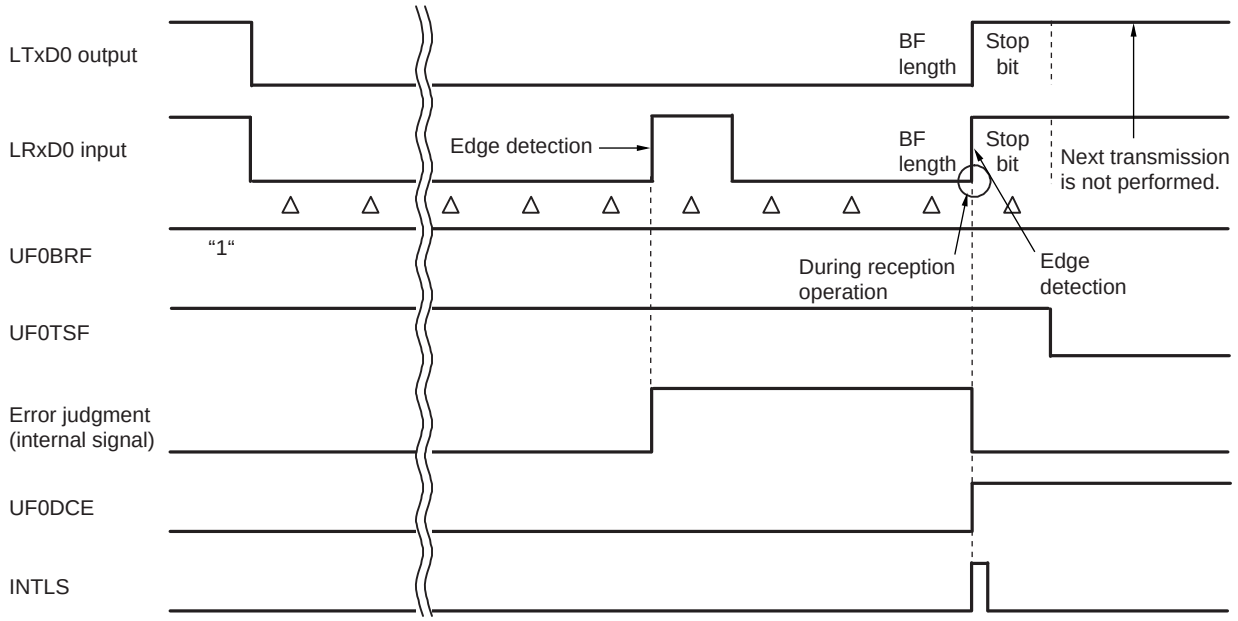


Figure 14-40. Timing of Occurrence of Data Consistency Error When BF Is Transmitted When UF0BRF = 1 (When Reception Is in Progress After Transmission of Stop Bit Has Started (Previous Input Data = 0))



(2) BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B)

If BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B) is set, a mode that recognizes a new BF is entered during data communication in addition to when waiting for successful BF reception (UF0BRF = 1). When not waiting for successful BF reception (UF0BRF = 0) and when a low level has been detected at the data stop bit position (10th bit), judging a framing error or an overrun error is being waited for until input data becomes high level, because a new BF may be undergoing reception. If the successive-low-level period is less than 11 bits, it is judged as error detection (see **Figure 14-41**). If not an overrun error, the first eight bits of receive data are stored into the UF0RX register. At this time, a successful BF reception flag (UF0BSF) is not set. When waiting for successful BF reception (UF0BRF = 1), detecting framing or overrun errors and storing receive data into the UF0RX register are not performed.

On the other hand, if the successive-low-level period is at least 11 bits, receiving of the new BF is judged successful and a successful BF reception flag (UF0BSF) is set (see **Figure 14-42**). Detection of framing or overrun errors is not performed. At this time, receive data is not stored into the UF0RX register.

If a reception operation is stopped when starting to transmit the stop bit of data or a BF while UF0BRF is “0”, the data consistency error interrupt and flag are changed when the bit following the stop bit is started (see **14.5.8 (2)**).

If a reception operation is being performed when starting to transmit the stop bit, it is performed when input data “1” is detected at a position following the stop bit (see **14.5.8 (1)** and **Figure 14-43**).

On the other hand, if input data “1” is detected during BF transmission with UF0BRF set to “1”, it is performed after transmission of the first stop bit has been completed (see **Figure 14-44**). After BF transmission has been completed, it is performed at a bit for which “1” is detected (see **Figure 14-45**).

Caution To set to BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B), be sure to set the UF0DCS bit of the UF0OPT1 register also to “1”.

Figure 14-41. Framing Error/Overrun Error Judgment Timing upon BF Reception Failure (When UF0BRF = 0)

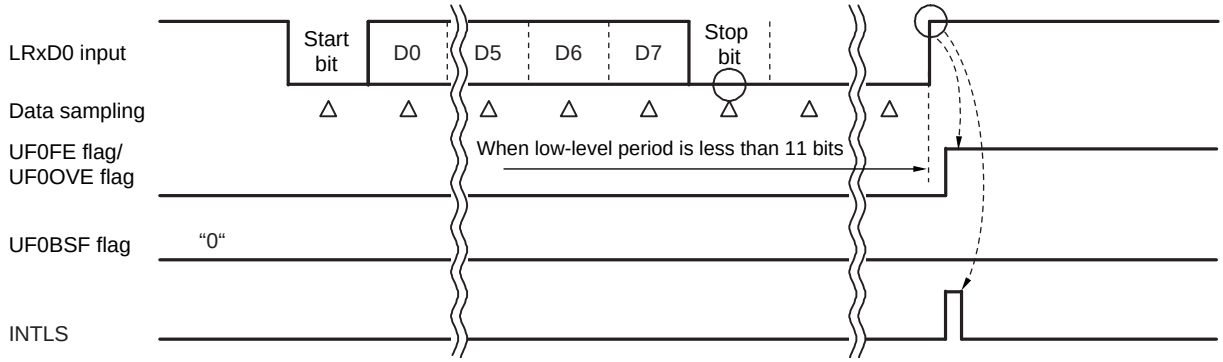


Figure 14-42. Status Interrupt Occurrence Timing upon Successful BF Reception (When UF0BRF = 0)

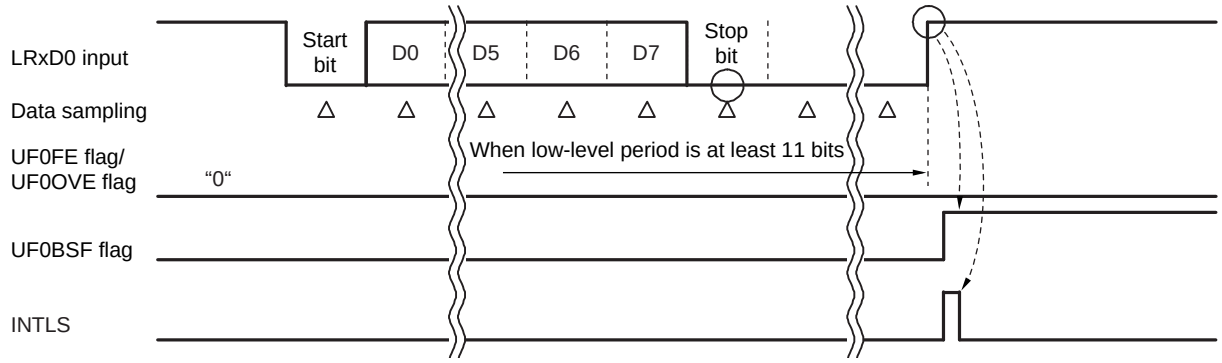


Figure 14-43. Example of Data Consistency Error Occurrence Timing When UF0BRF = 0

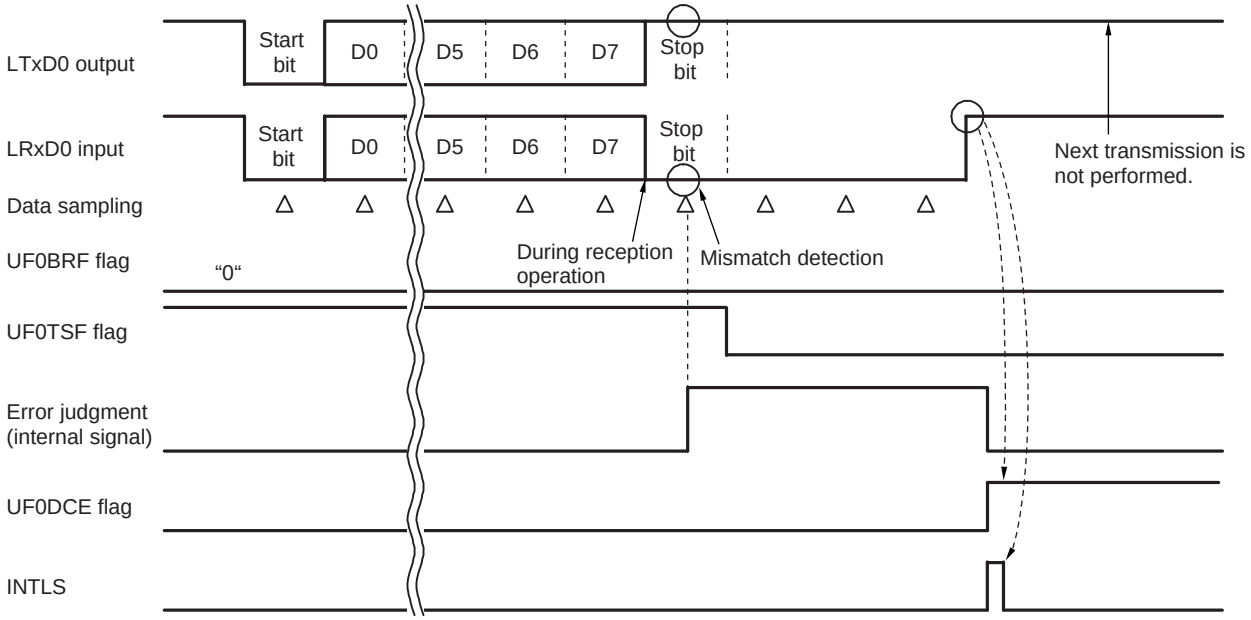


Figure 14-44. Example of Consistency Error Occurrence Timing During BF Transmission When UF0BRF = 1 (If Reception Operation Is Stopped When Input Data "1" Is Detected After Stop Bit (Previous Bit Is "1"))

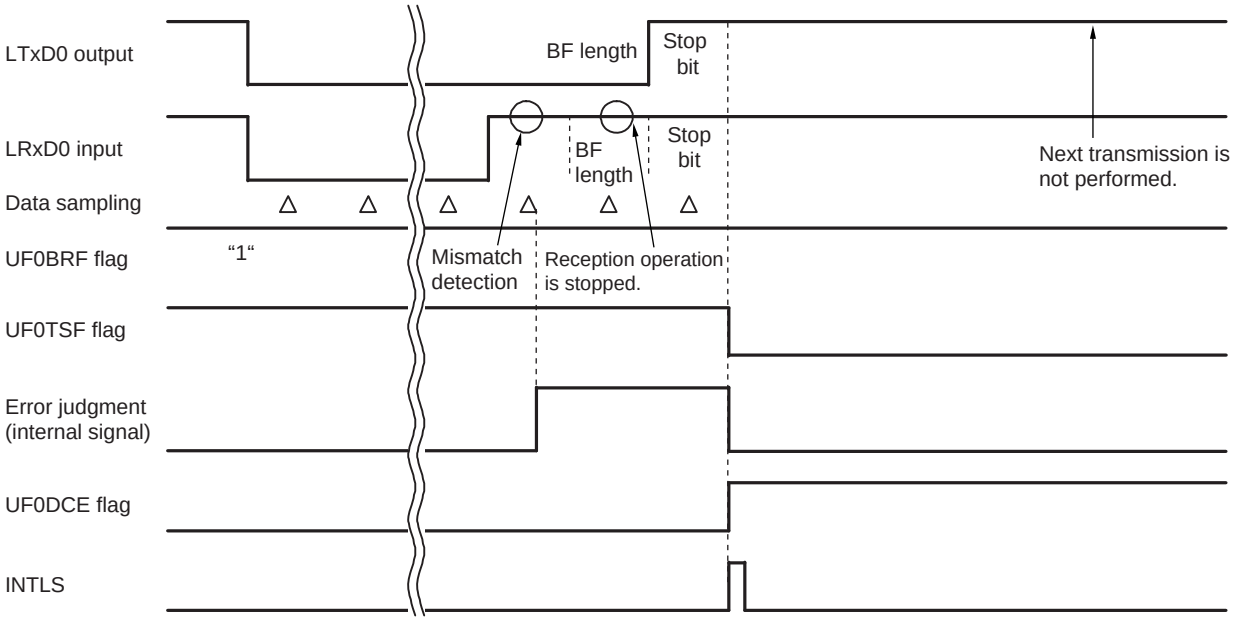
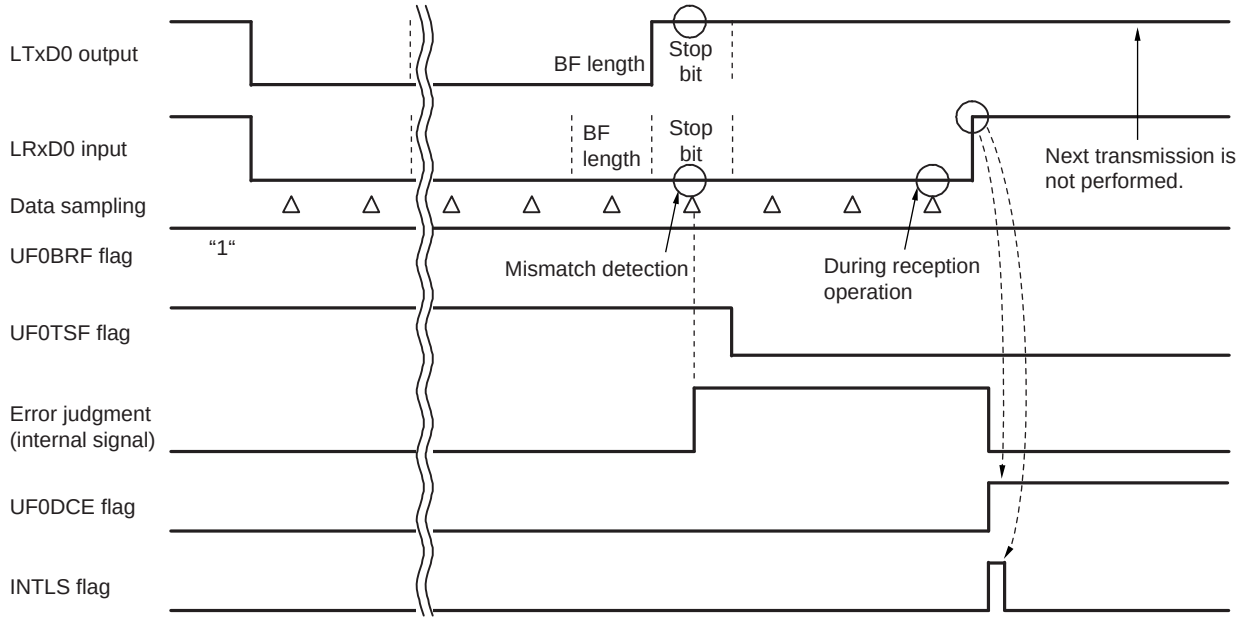


Figure 14-45. Example of Consistency Error Occurrence Timing During BF Transmission When UF0BRF = 1 (If During Reception Operation When Input Data "1" Is Detected After Stop Bit (Previous Bit Is "0"))



14.5.10 LIN-UART reception status interrupt generation sources

LIN-UART reception status interrupt generation sources include parity errors, framing errors, overrun errors, data consistency errors which occur only during LIN communication, successful BF reception, ID parity errors, checksum errors, and response preparation errors which occur only in automatic baud rate mode, and ID matches and expansion bit detections which occur only when expansion bits are enabled. When these sources are detected, LIN-UART reception status interrupt (INTLS) is generated. The type of a generation source can be referenced by using the status register (UF0STR). The content of processing is determined by referencing the UF0STR register in the LIN-UART reception status interrupt servicing routine.

Status flags must be cleared by writing “1” to the corresponding bits (excluding the UF0TSF and UF0RSF bits of the UF0STC register) by using software.

The LIN-UART reception status interrupt generation timing and status flag change timing differ, depending on the mode setting and generation source.

Table 14-3. LIN-UART Reception Status Interrupt Generation Sources

Status Flag	Generation Source	Description
UF0PE	Parity error	The parity calculation result of receive data and the value of the received parity bit do not match.
UF0FE	Framing error	No stop bit is detected. (A low level is detected at a stop bit position.)
UF0OVE	Overrun error	The next data reception is completed before the receive data transferred to the receive data register is read.
UF0DCE	Data consistency error	The data consistency check selection bit (UF0DCS) is set, and the values of transmit data and receive data do not match during data transmission. Transmission operation and reception operation are out of synchronization.
UF0BSF	Successful BF reception	A new BF is successfully received when in BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B). (This occurs also when the master transmits a BF.)
UF0IPE	ID parity error	Either parity bit of the received PID includes an error.
UF0CSE	Checksum error	The result of comparing the checksum received during response reception and the automatically calculated result is illegal.
UF0RPE	Response preparation error	Response preparation could not be performed before reception of the first byte by a response was completed.
UF0IDM	ID match	When the following conditions are satisfied: - Comparison of expansion bit data is enabled (UF0EBC = 1). - The expansion bit is at the level set by using the expansion bit detection level selection bit (UF0EBL). - The received data matches the value of the UF0ID register.
UF0EBD	Expansion bit detection	The level set by using the expansion bit detection level select bit (UF0EBL) is detected at a receive data expansion bit.

The following processing is required depending on the generation source when a status interrupt is generated.

- Parity error, data consistency error
  False data has been received, so read the received data and then discard it. Then perform communication again. If the received data is not read, an overrun error might occur when reception ends next time. For a data consistency error, a data conflict may also be possible.

- Framing error
  The stop bit could not be detected normally, or a bit offset may have occurred due to false detection of the start bit. Furthermore, the baud rate may be offset from that of the transmission side or a BF of insufficient length may have been received in LIN communication.
  When framing errors occur frequently, a bit or the baud rate may be offset, so perform initialize processing on both the transmission side and reception side, and restart communication. Furthermore, to receive the next data after a framing error has occurred, the reception pin must become high level once.

- Overrun error
  Data of one frame that was received immediately before is discarded, because the next reception is completed before receive data is read. Consequently, the data must be retransmitted.

- Successful BF reception
  Preparation for starting a new frame slot must be performed, because a new BF has been received successfully.

- ID parity error
  Set a request bit without a response (UF0NO), because the received PID is illegal. Afterward, do not perform response transmission or reception, wait for the next BF to be received, and ignore that frame.

- Checksum error
  Discard the received response (data field), because it is illegal.

- Response preparation error
  Wait for the next BF to be received and ignore that frame, because response processing cannot be performed normally.

- ID match
  Receive data of the expansion bit of a level set by using the UF0EBL bit has matched with the UF0ID register setting value. Perform, therefore, corresponding processing such as disabling expansion bit data comparison (UF0EBC = 0) to receive subsequent data.

- Expansion bit detection
  Perform corresponding processing such as preparing for starting DMA transfer, because receive data of the expansion bit of a level set by using the UF0EBL bit has been received.

Caution Status flags are an accumulation of all sources that have been generated after the status flag has been cleared, and do not reflect the latest state. Consequently, the above-mentioned processing must be completed before the next reception is completed and the status flag must be cleared.

The following table shows examples of processing corresponding to statuses when performing LIN communication.

Table 14-4. Examples of Processing Corresponding to Statuses During LIN Communication (When in BF Reception Enable Mode During Communication (UF0MD1, UF0MD0 = 10B) and When UF0DCS = 1)

UF0BSF	UF0DCE	UF0FE	UF0OVE	Status	Processing Example
1	1	×	×	A mismatch is detected between transmit and receive data during BF transmission in master operation. Successive low levels of at least 11 bits are received. The transmission is not performed even if the next data transmission has been prepared.	• The next data (Sync field) transmission is not performed and waiting for the next time schedule is performed, because the other party of communication may not have been able to recognize the BF. • The other party of communication may not have been able to recognize the BF, but all status flags are cleared and the next data is written to transmit the next data (Sync field).
1	0	×	×	BF transmission and BF reception are performed successfully in master operation.	Processing to transmit the next data (Sync field) is performed.
				BF reception is performed successfully in slave operation.	Processing to receive the next data (Sync field) is performed.
0	1	×	×	BF transmission or data (including an SF or a PID) transmission has failed in master operation. Even if transmission of the next data or BF has been prepared, the transmission will not be performed.	Subsequent transmit and receive data is discarded, all status registers are cleared, and the system waits for the next time schedule.
				Data transmission has failed in slave operation. Even if transmission of the next data has been prepared, the transmission will not be performed.	Subsequent transmit and receive data is discarded, all status registers are cleared, and the system waits for the next time schedule.
0	0	1	×	A framing error has been detected during data reception.	Processing when a framing error has been detected is performed.
0	0	×	1	An overrun error has been detected during data reception. The single data that was received immediately before has been discarded.	Processing when an overrun error has been detected is performed.

Cautions 1. Clear all status flags that have been set for any processing.

2. When an error is detected in LIN communication (including when BF reception has been performed successfully when BF reception enable mode during communication (UF0MD1, UF0MD0 = 10B) has been set), a status interrupt request signal (INTLS) is generated instead of a reception complete interrupt request signal (INTLR) and a status flag is set according to the communication status.

Remark ×: don't care

14.5.11 Transmission start wait function

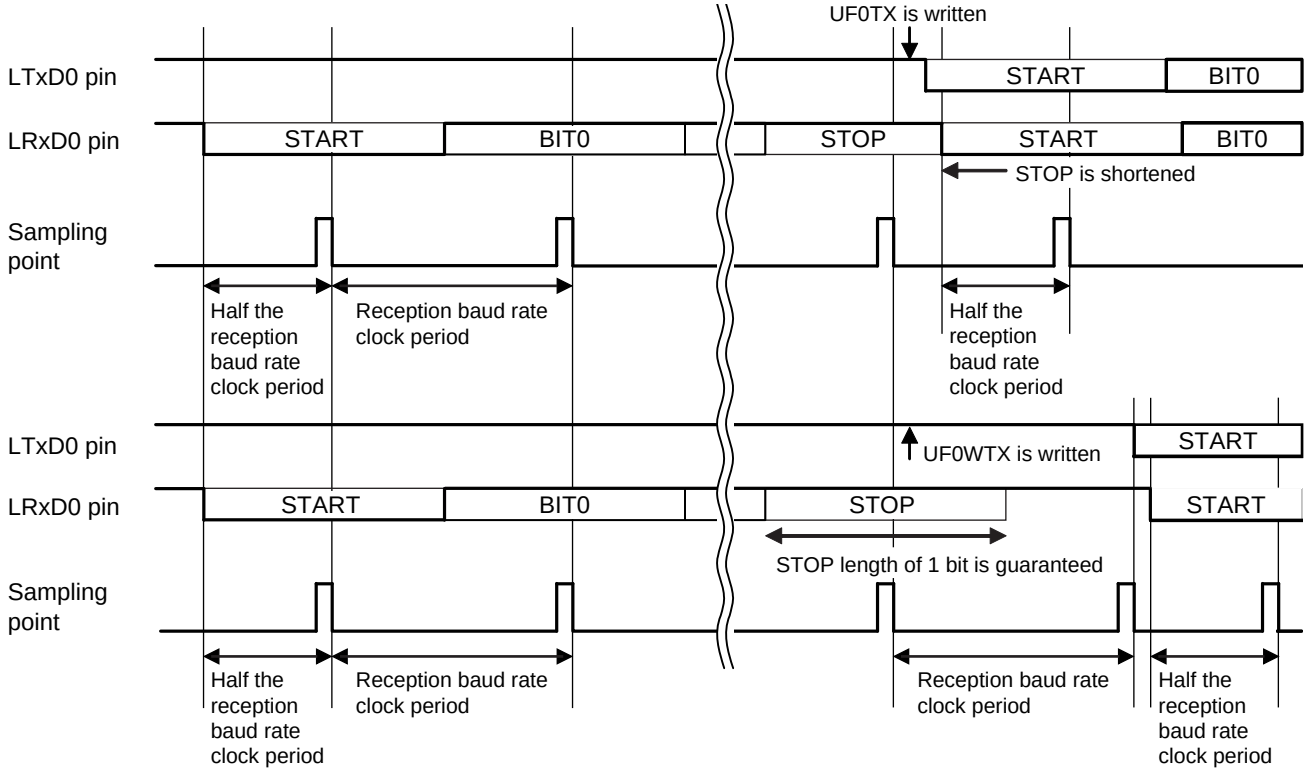
The RL78/F12 is provided with a function to guarantee the stop bit length of reception when reception is switched to transmission to perform LIN communication.

To delay starting of transmission until completion of the stop bit of reception, write data to the UF0WTX register which is a wait-dedicated register, instead of writing transmit data to the UF0TX register as a transmission start request.

In this case, starting transmission is being waited for one bit until the stop bit of receive data has ended for sure.

Note that only a wait of one bit is performed, even if the stop bit length has been set to two bits by using the stop bit length select bit (UF0SL).

Figure 14-46. When Transmit Data Has Been Written During Stop Bit of Receive Data



Cautions 1. When LIN communication is not performed, accessing the UF0WTX register is prohibited.

2. Writing to the UF0WTX register is prohibited except when reception is switched to transmission (such as during transmission).

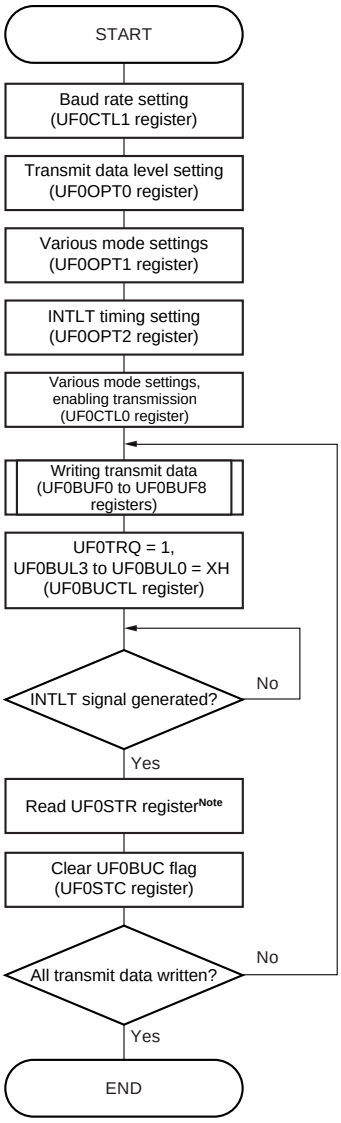
14.6 UART Buffer Mode

The RL78/F12 is provided with a 9-byte transmission buffer that can be used for normal UART communication (UF0MD1, UF0MD0 = 00B).

14.6.1 UART buffer mode transmission

The following figure shows the procedure for transmitting data in UART buffer mode.

Figure 14-47. UART Buffer Mode Transmission Processing Flow



Note This can be omitted.

Cautions 1. Set the following values when performing data transmission in UART buffer transmission mode.

Expansion bits are disabled (UF0EBE = 0).
Normal UART mode (UF0MD1, UF0MD0 = 00B).
Data consistency checking is disabled (UF0DCS = 0).
Waiting for buffer transmission start is disabled (UF0TW = 0).
Continuation of transfer is disabled (UF0CON = 0).
Request bits without responses are present (UF0NO = 0).
Reception requests are disabled (UF0RRQ = 0).

2. UF0PRQ must not be set to 1 before completion of receive data reading.

Remarks 1. See **(2)** of **14.11 Cautions on Use for details of starting LIN-UART**.

2. X: don't care

When transferring the number of bytes (1 to 9) set to the buffer length bit (UF0BUL3 to UF0BUL0) has ended, a transmission interrupt request signal (INTLT) is output. When the buffer length bit is set to “0” or “10 to 15”, transfer of nine bytes is performed.

Writing data to the transmit data register (UF0TX) during transmission in buffer mode is prohibited.

To stop transfer midway, write “0” to the transmission enable bit (UF0TXE). Data transmission processing is stopped and the UF0TRQ bit and UF0TSF flag are cleared.

Figure 14-48. UART Buffer Mode Transmission Example (UF0ITS = 0)

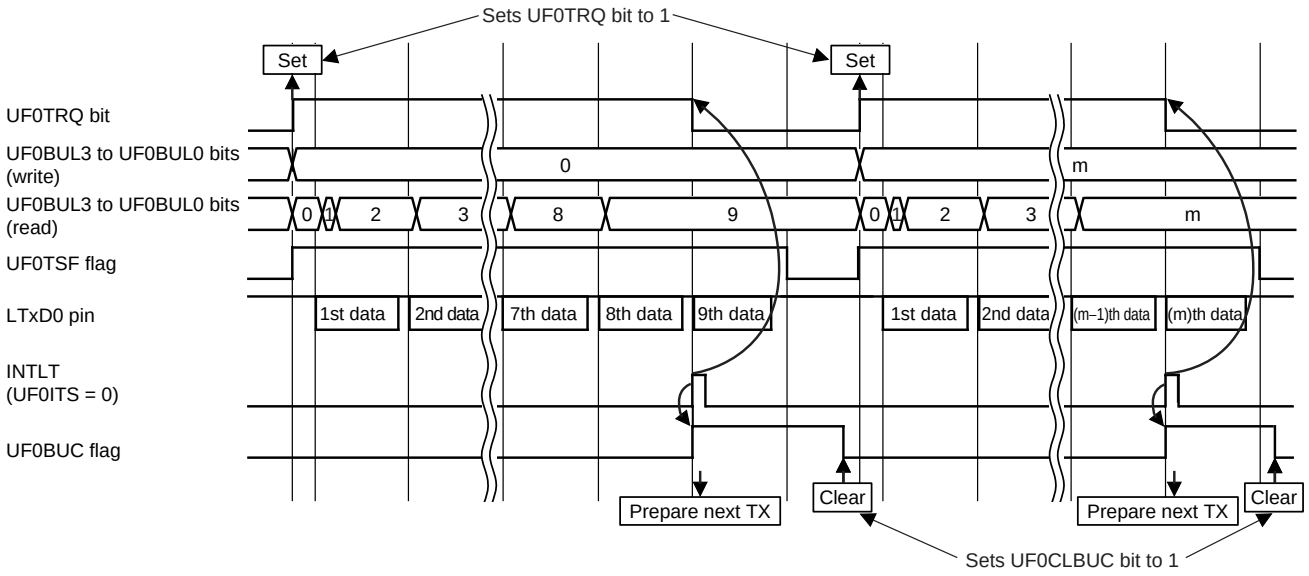
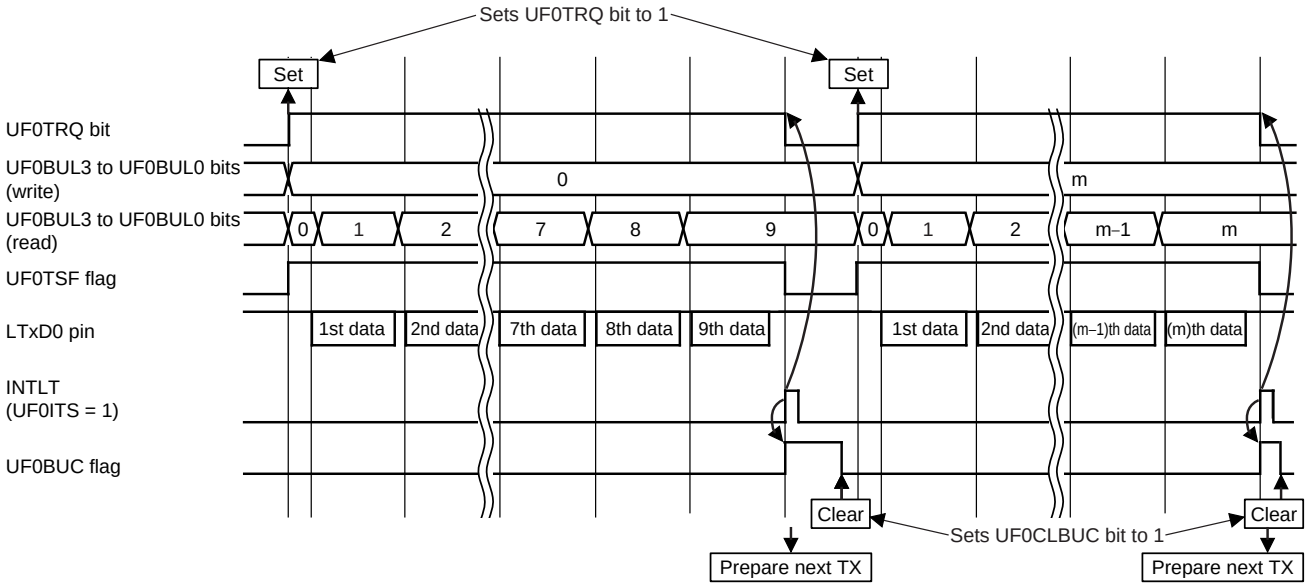


Figure 14-49. UART Buffer Mode Transmission Example (UF0ITS = 1)



Remark m = 1 to 9

14.7 LIN Communication Automatic Baud Rate Mode

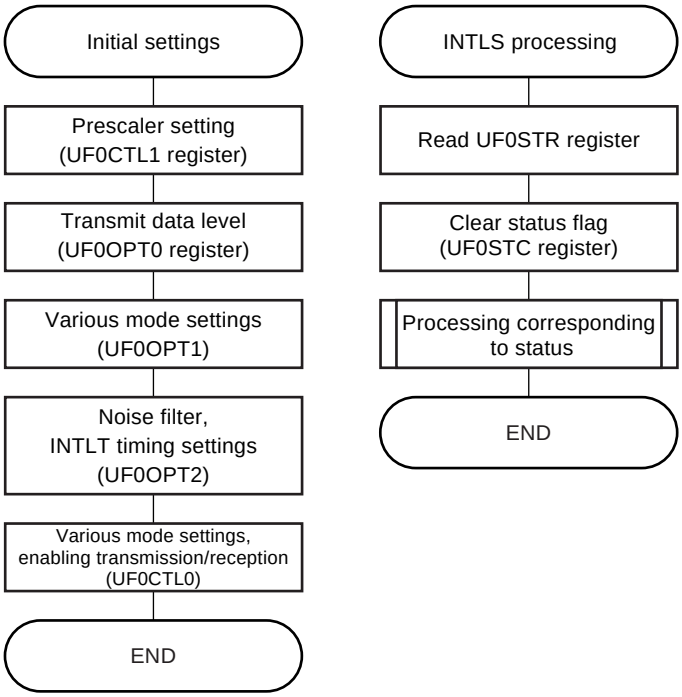
In LIN communication automatic baud rate mode, a BF and an SF are automatically detected and the baud rate is set according to the measurement result of the SF.

When UF0MD1 and UF0MD0 are set to “11B”, operation is performed in automatic baud rate mode.

Operation can be performed with the baud rate at 2,400 bps to 128 kbps. Set to 8 to 12 MHz the clock (prescaler clock) that has been divided by using a prescaler. At that time, the setting values of UF0PRS2 to UF0PRS0 must be calculated from the f_{CLK} frequency and initial settings must be performed.

When using LIN-UART as the master, using automatic baud rate mode (UF0MD1, UF0MD0 = 11B) is prohibited.

Figure 14-50. Basic Processing Flow Example of LIN Communication Automatic Baud Rate Mode (1/2)



Cautions 1. Set the following values when performing LIN communication automatic baud rate mode.

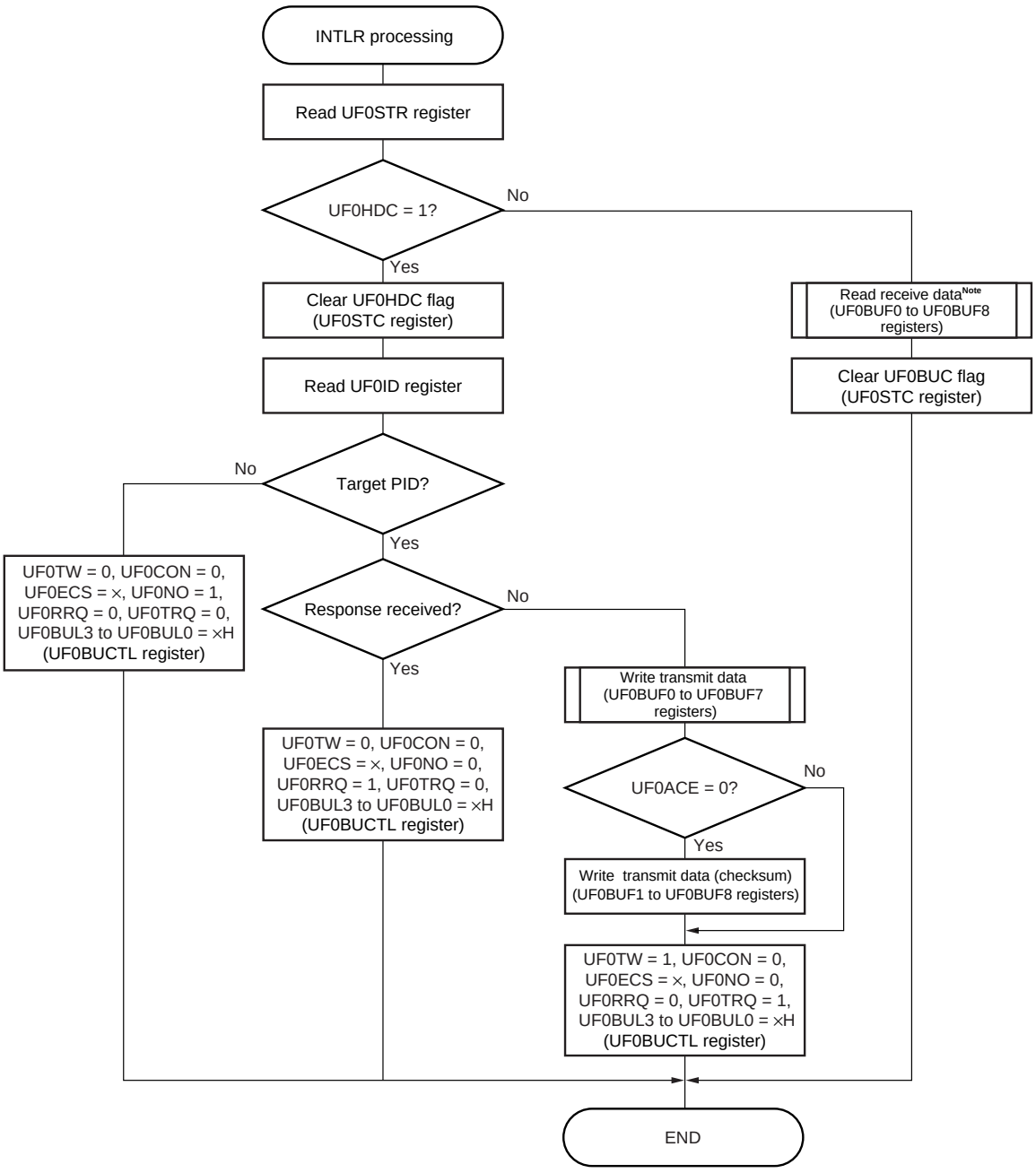
The transmit and receive data levels are normal input (UF0TDL = UF0RDL = 0).
Expansion bits are disabled (UF0EBE = 0).
Automatic baud rate mode (UF0MD1, UF0MD0 = 11B) as the mode.
Consistency check selection (UF0DCS = 1).
Transmission interrupt is transmission start (UF0ITS = 0).
Communication direction control is LSB first (UF0DIR = 1).
The parity selection bit is received without parity (UF0PS1, UF0PS0 = 00B).
The data character length is 8 bits (UF0CL = 1).
Transmit data register is default value (UF0TX = 0000H).

2. Set the UF0PRS2 to UF0PRS0 bits so that the clock that has been divided by using a prescaler is 8 to 12 MHz.

3. The checksum field should be included when UF0ACE = 0.

Remark See (2) of 14.11 Cautions on Use for details of starting LIN-UART.

Figure 14-50. Basic Processing Flow Example of LIN Communication Automatic Baud Rate Mode (2/2)



Note This can be omitted.

Cautions 1. When the buffer length bits (UF0BUL3 to UF0BUL0) have been set to "0" or "10 to 15", reception or transmission of nine bytes is performed. When the buffer length is set to "1 to 8", buffers of the number of bytes set are used in ascending order of the buffer numbers.

Example: When UF0BUL3 to UF0BUL0 are set to "1", data is always stored only into the UF0BUF0 register.

2. Do not set the UF0RRQ bit before completion of receive data reading, because, when the UF0RRQ bit is set, storing (overwriting) into a buffer is performed even if reading receive data has not ended.

3. Setting (1) the UF0TW bit is prohibited, except when operation is switched to response transmission after header reception.

Remark ×: don't care

If a PID stored into the UF0ID register is not a target when header reception is completed (UF0HDC = 1), the UF0NO bit is set and subsequent transmission and reception processing are stopped (responses are ignored).

For a response reception PID, the UF0RRQ bit is set at the same time as the response data length (UF0BUL3 to UF0BUL0), and response reception processing is performed.

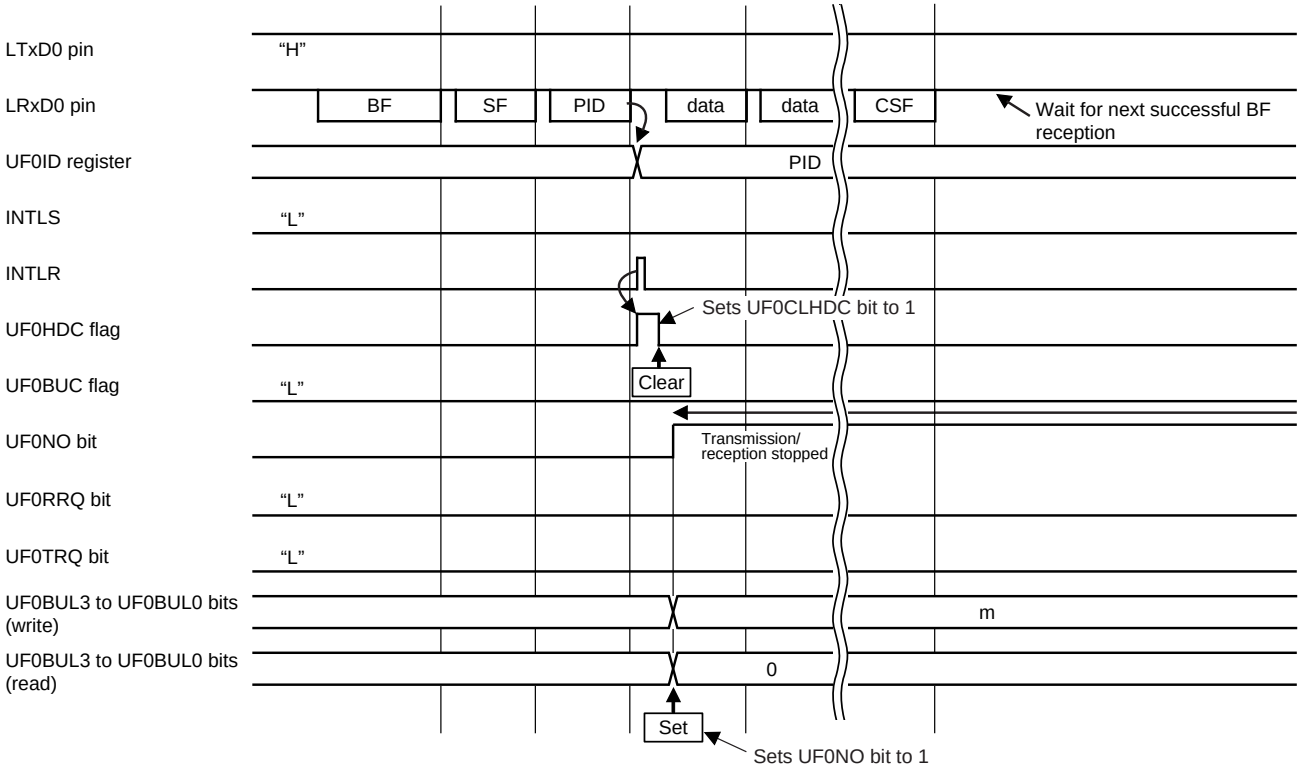
For a response transmission PID, the UF0TRQ bit is set at the same time as the response data length (UF0BUL3 to UF0BUL0), and response transmission processing is performed, after transmit data has been set to a buffer. At that time, the receive data will be stored in the UF0RX register. However, no overrun error will occur even if the receive data is not read.

Perform processing (setting the UF0NO, UF0RRQ, or UF0TRQ bit) for the PID before receiving the first byte of the response is completed. Otherwise, a response preparation error occurs. See **14.7.2 Response preparation error detection function** for details.

During response reception and response transmission also, when a status interrupt request signal (INTLS) has been generated due to an error, transmission and reception operations are stopped and waiting for the next BF reception is performed.

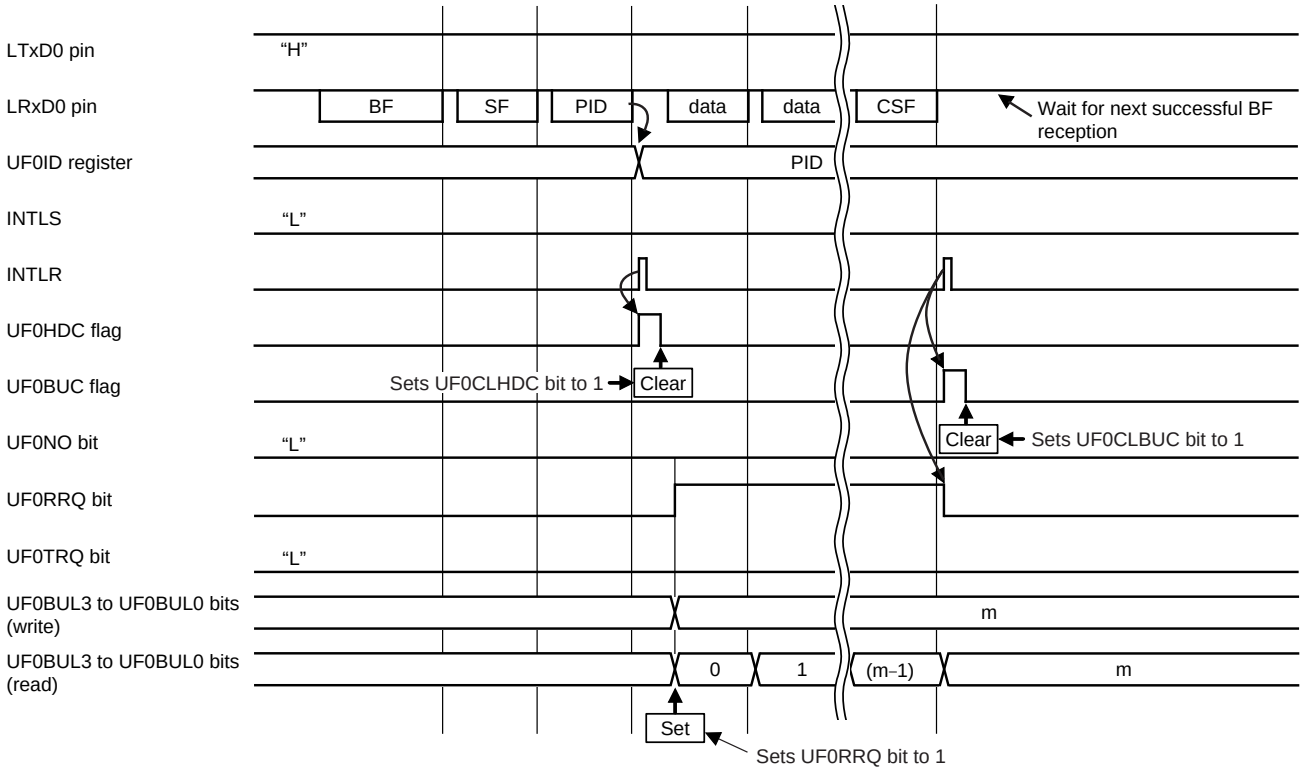
In automatic baud rate mode, no overrun error occurs, because a buffer is used (the UF0RX register is not used).

Figure 14-51. LIN Communication Automatic Baud Rate Mode (Non-Target PID)



Remark m = 1 to 9

Figure 14-52. LIN Communication Automatic Baud Rate Mode (Response Reception)



An example of how reception results are stored into a buffer when 8-byte data is received (UF0BUL3 to UF0BUL0 = 9) and when 3-byte data is received (UF0BUL3 to UF0BUL0 = 3) are shown below.

(1) When 8-byte data is received (UF0BUL3 to UF0BUL0 = 9)

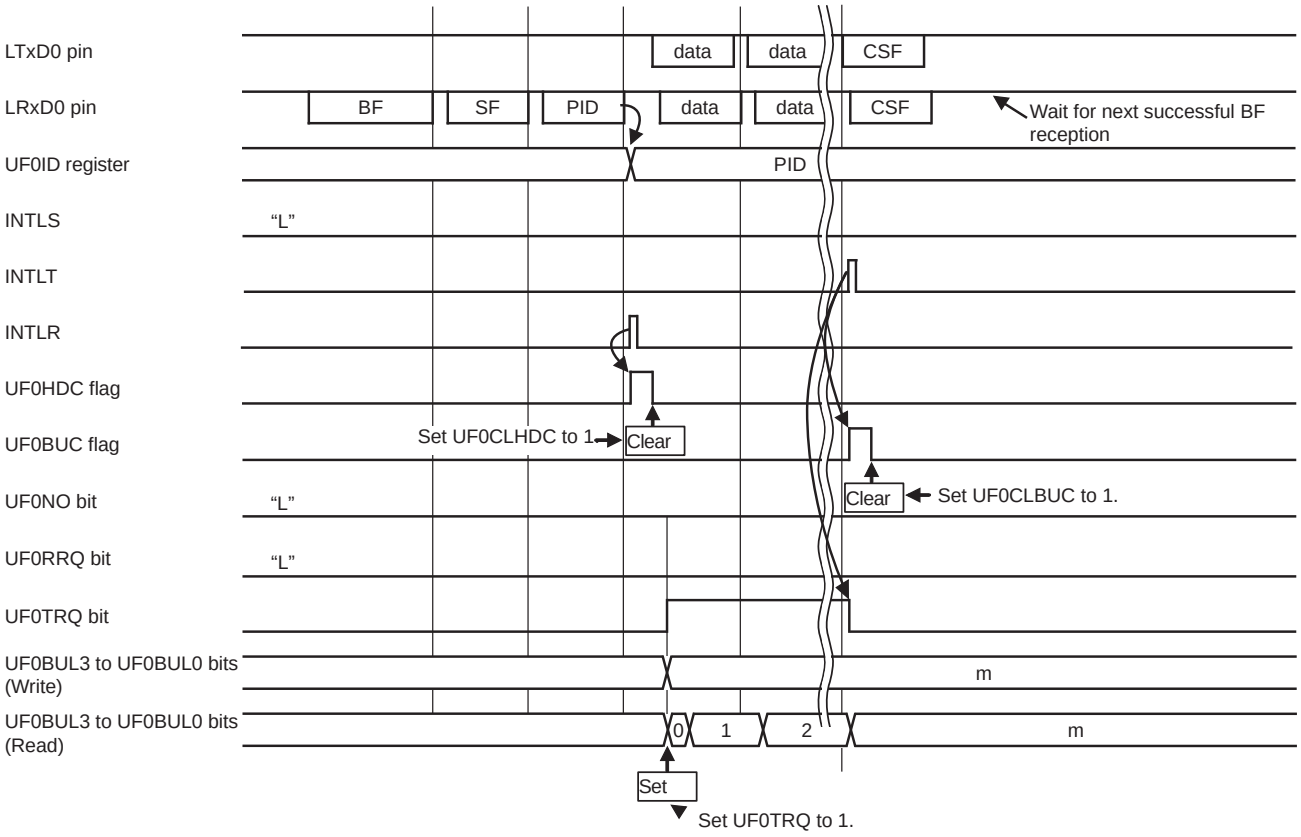
	Reception results
UF0BUF8	Checksum
UF0BUF7	Data7
UF0BUF6	Data6
UF0BUF5	Data5
UF0BUF4	Data4
UF0BUF3	Data3
UF0BUF2	Data2
UF0BUF1	Data1
UF0BUF0	Data0

(2) When 3-byte data is received (UF0BUL3 to UF0BUL0 = 4)

	Reception results
UF0BUF8	–
UF0BUF7	–
UF0BUF6	–
UF0BUF5	–
UF0BUF4	–
UF0BUF3	Checksum
UF0BUF2	Data2
UF0BUF1	Data1
UF0BUF0	Data0

Caution When UARTF is being used with the auto checksum feature enabled (UF0ACE = 1), the checksum data is not stored in a buffer.

Figure 14-53. LIN Communication Automatic Baud Rate Mode (Response Transmission)



Examples of the buffer settings and the status of the buffer after 8 bytes of data have been transmitted (UF0BUL3 to UF0BUL0 = 9) and after 3 bytes of data have been transmitted (UF0BUL3 to UF0BUL0 = 3) are shown below.

(1) When 8-byte data is transmitted (UF0BUL3 to UF0BUL0 = 9)

	Buffer setting	Buffer status
UF0BUF8	TX Checksum	RX Checksum
UF0BUF7	Data7	Data7
UF0BUF6	Data6	Data6
UF0BUF5	Data5	Data5
UF0BUF4	Data4	Data4
UF0BUF3	Data3	Data3
UF0BUF2	Data2	Data2
UF0BUF1	Data1	Data1
UF0BUF0	Data0	Data0

(2) When 3-byte data is received (UF0BUL3 to UF0BUL0 = 4)

	Buffer setting	Buffer status
UF0BUF8	–	–
UF0BUF7	–	–
UF0BUF6	–	–
UF0BUF5	–	–
UF0BUF4	–	–
UF0BUF3	Checksum	Checksum
UF0BUF2	Data2	Data2
UF0BUF1	Data1	Data1
UF0BUF0	Data0	Data0

Caution To enable the automatic checksum function (UF0ACE = 1), checksum is not required to be set to the buffer by using software.

14.7.1 Automatic baud rate setting function

Received low-level widths are always measured when in automatic baud rate mode. BF detection is judged as being performed successfully when the first low-level width is at least 11 times the second low-level width, and it is checked that the data is 55H. If the data is confirmed to be 55H and the SF is judged to have been successfully received, reception is paused, the UF0BRS11 to UF0BRS00 bits are set again, and reception resumes after the start bit is detected.

When it has been confirmed that the data is 55H, successful SF detection is judged and baud rate setting results are automatically set to the UF0BRS11 to UF0BRS00 bits. At that time, the settings of the UF0PRS2 to UF0PRS0 bits are not changed. Afterward, the next data (PID) is received after transmission or reception processing has been enabled. A reception complete interrupt request signal (INTLR) is generated when there are no errors upon PID reception completion (stop bit position), and an error flag is set and a status interrupt request signal (INTLS) is generated when there is an error. In both cases, a header reception completion flag (UF0HDC) is set. On the other hand, when the data is not 55H, SF detection is judged to have failed, the next BF (low level) reception is being waited for with the transmission or reception processing being stopped, and baud rate setting is not performed.

When the stop bit position of reception processing is reached while transmission or reception processing is enabled, errors such as framing errors and consistency errors are detected and a status interrupt request signal (INTLS) may be generated. This is also applicable when a BF has been received during communication.

Figure 14-54. Example of BF/SF Reception Failure

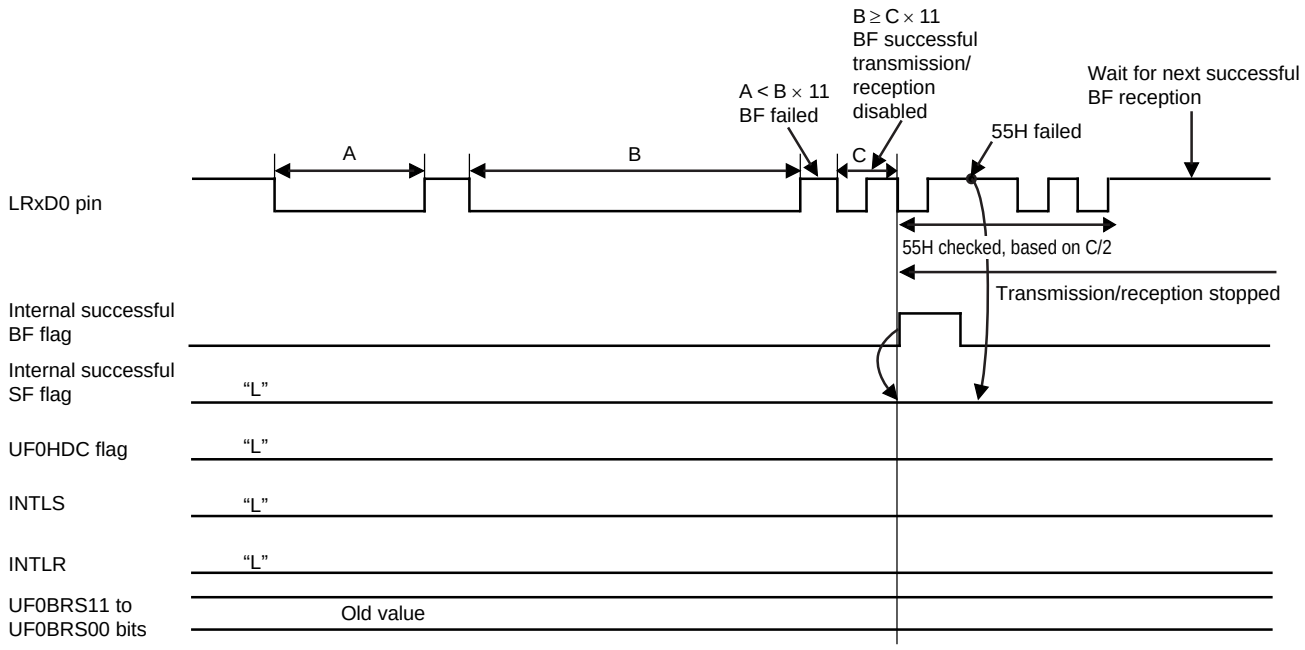
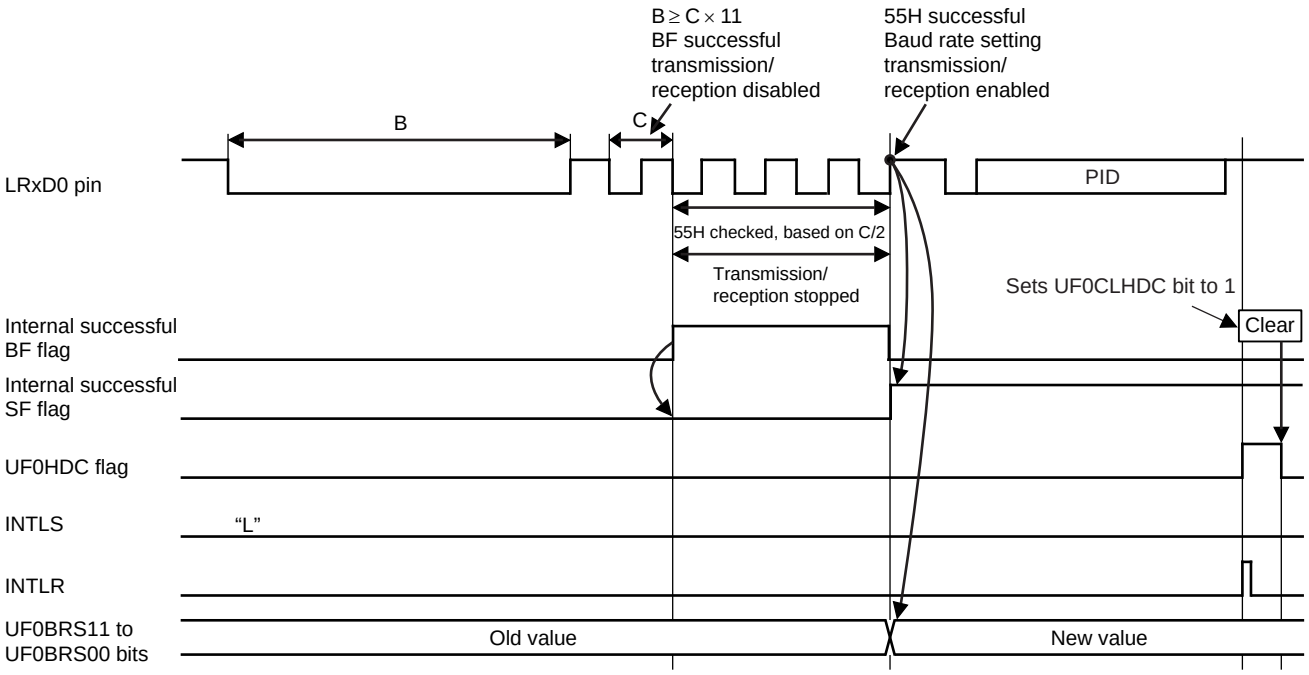
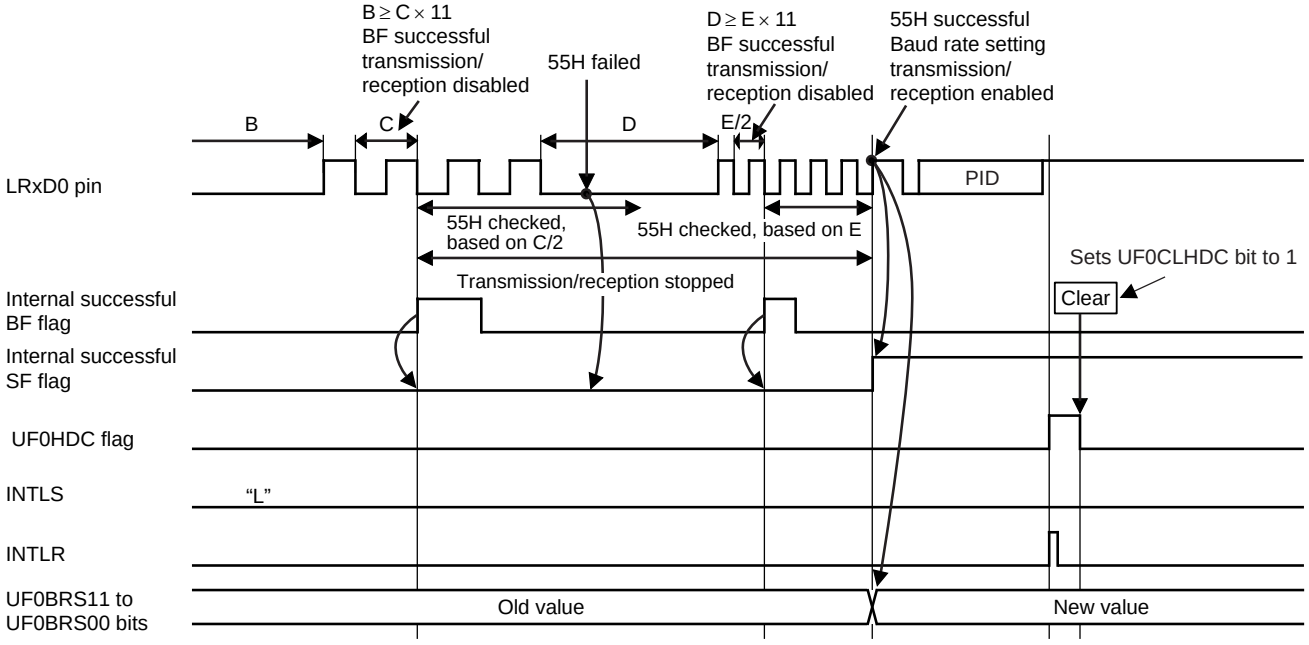


Figure 14-55. Example of Successful BF, SF, and PID Reception



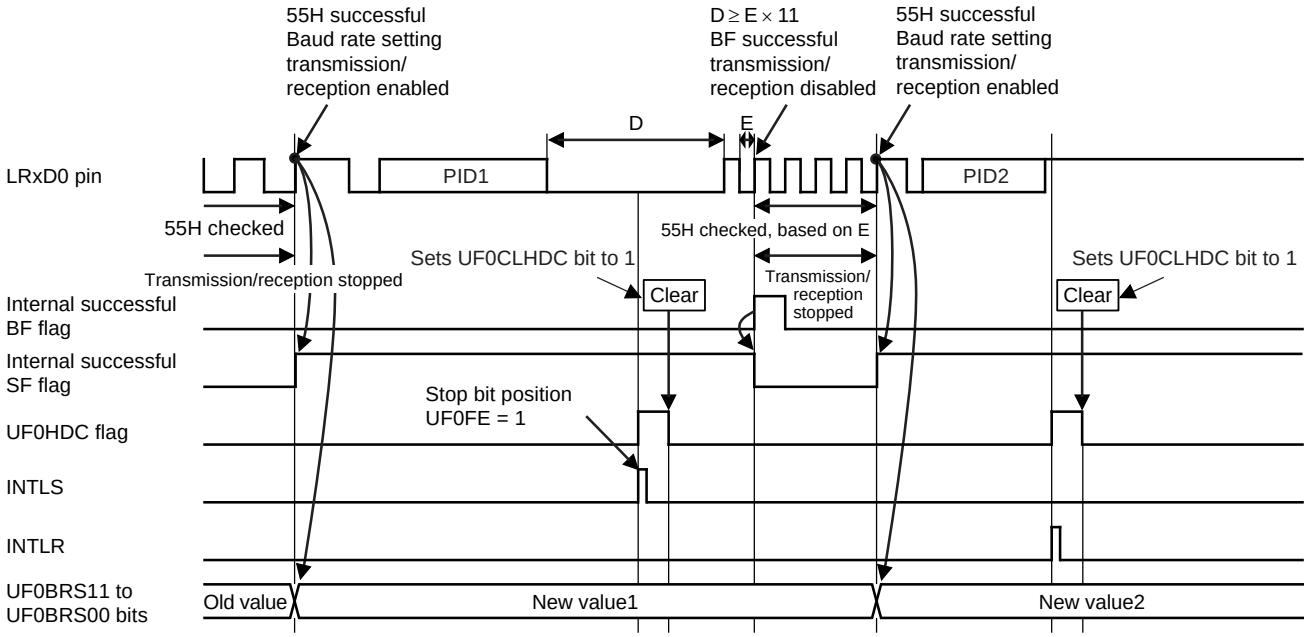
Caution When a PID reception error has occurred, a status interrupt request signal (INTLS) is generated instead of a reception complete interrupt request signal (INTLR) and other error flags (such as UF0FE and UF0IPE) change.

Figure 14-56. Example of Successful BF Reception During SF Reception (No PID Reception Error)



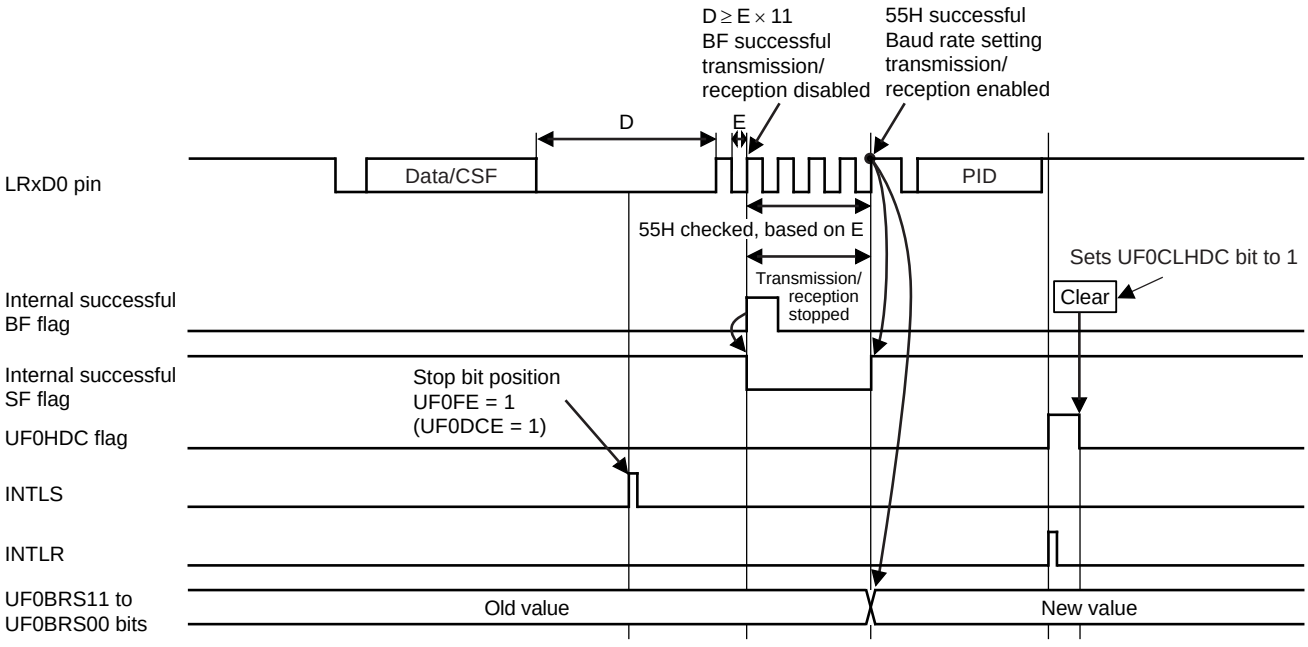
Caution When a PID reception error has occurred, a status interrupt request signal (INTLS) is generated instead of a reception complete interrupt request signal (INTLR) and other error flags (such as UF0FE and UF0IPE) change.

Figure 14-57. Example of Successful BF Reception During PID Reception (No PID2 Reception Error)



Caution If the PID1 stop bit position comes after the point where the internal successful BF flag has been set, the UF0HDC flag and error flags (such as UF0FE and UF0IPE) are not set, and INTLS is also not generated.

Figure 14-58. Example of Successful BF Reception During Data/CSF Reception (No PID Reception Error)



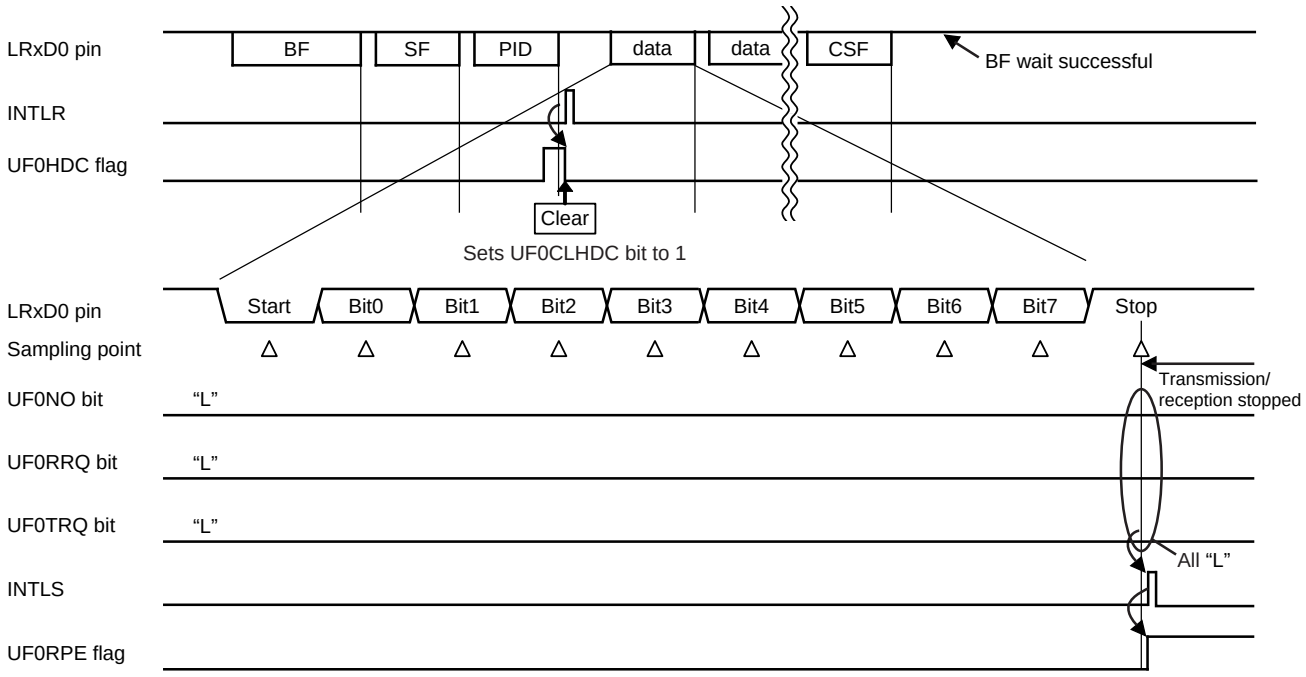
Caution If the Data/CSF stop bit position comes after the point where the internal successful BF flag has been set, the UF0BUC flag and error flags (such as UF0FE, UF0DCE, UF0CSE, and UF0RPE) are not set, and INTLS is also not generated.

14.7.2 Response preparation error detection function

If response preparation (setting of the UF0NO, UF0RRQ, and UF0TRQ bits) is not performed before reception of the first byte by a response is completed (sampling point of the stop bit (first bit)) when in automatic baud rate mode (UF0MD1, UF0MD0 = 11B), a response preparation error flag (UF0RPE) is set, a status interrupt request signal (INTLS) is generated, and subsequent transmission and reception processing are stopped (responses are ignored) without data being stored.

When response transmission is started (UF0TRQ = 1) after reception at the LRxD0 pin has been started, recognition can be performed by the occurrence of consistency errors.

Figure 14-59. Response Preparation Error Occurrence Example

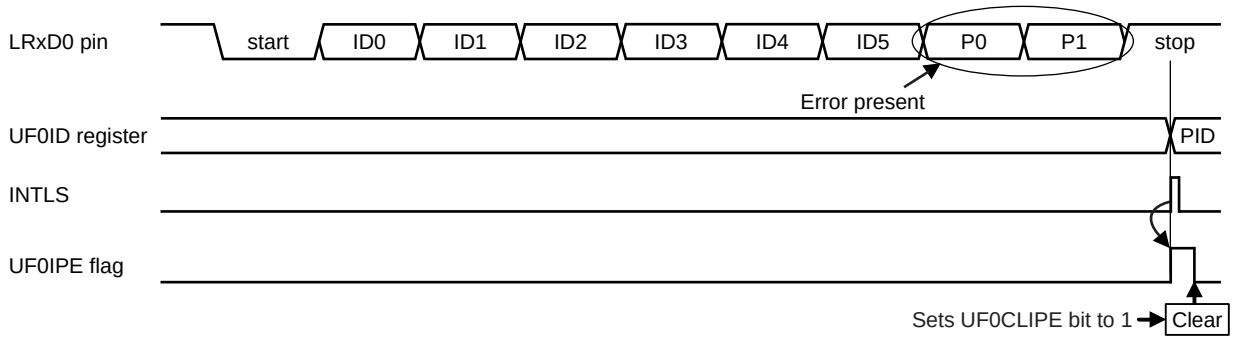


Caution If UF0CON = 0, no response preparation error will occur, because a BF reception wait state is entered after communication of the number of bytes set using the UF0BUL3 to UF0BUL0 bits is completed.
If UF0CON = 1, a response preparation error check state is entered again after communication of the number of bytes set using the UF0BUL3 to UF0BUL0 bits is completed.
A response preparation error will occur if a receive operation is started before setting UF0TRQ the next time after response transmission is completed.

14.7.3 ID parity check function

When the ID parity check select bit is set (UF0IPCS = 1) in automatic baud rate mode (UF0MD1, UF0MD0 = 11B), the PID parity bits (P0, P1) are checked when the received PID is stored into the UF0ID register. At that time, if either parity bit includes an error, an ID parity error flag (UF0IPE) is set, a status interrupt request signal (INTLS) is generated instead of a reception complete interrupt request signal (INTLR), and the PID is stored into the UF0ID register.

Figure 14-60. PID Parity Error Occurrence Example



14.7.4 Automatic checksum function

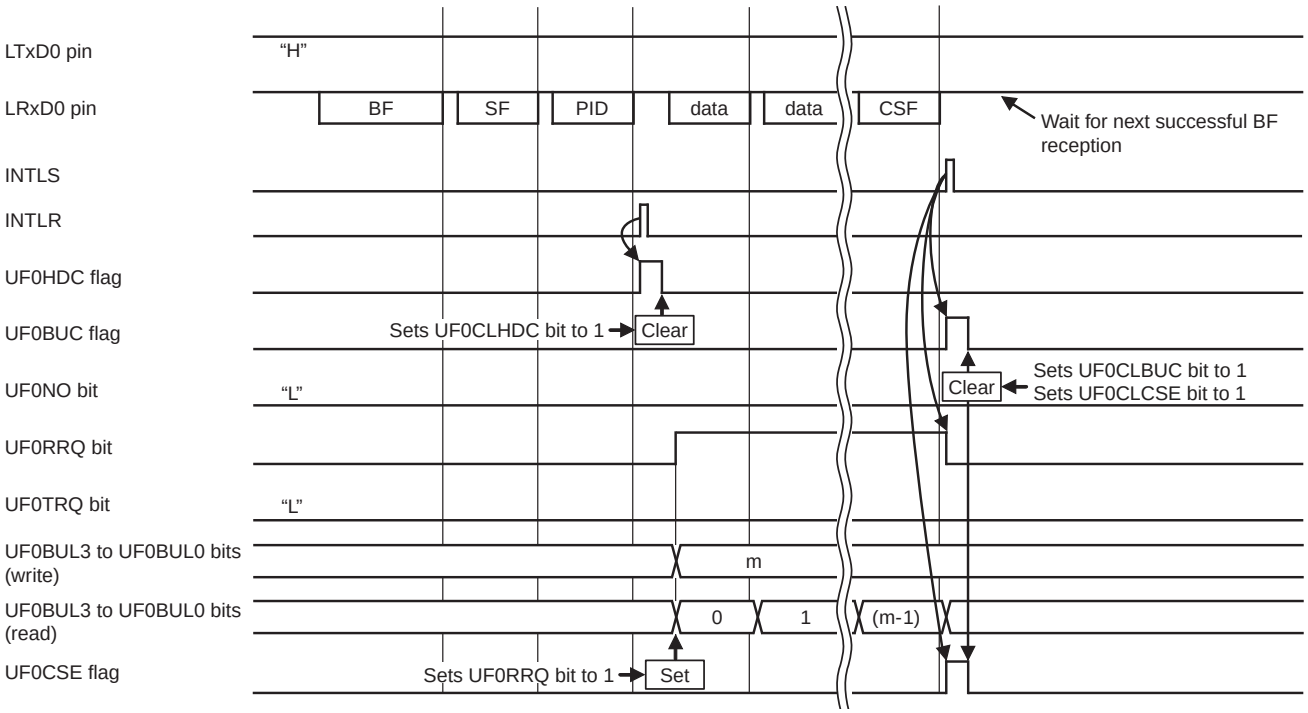
When the automatic checksum enable bit is set (UF0ACE = 1) in automatic baud rate mode (UF0MD1, UF0MD0 = 11B), a checksum is automatically calculated. Enhanced checksum (calculation targets: PID and data) and classic checksum (calculation target: only data) can be selected for each frame by using the enhanced checksum selection bit (UF0ECS).

During response transmission, calculation is performed when data is transferred in 1-byte units from a buffer register to a transmit shift register[Note], and the calculation result is automatically added to the end of response transmission and transmitted. A checksum is not required to be set to a buffer by using software.

During response reception, calculation is performed when data is stored into a buffer register in 1-byte units[Note], and the stored data and calculation result are automatically compared when the received checksum is stored into a buffer. A reception complete interrupt request signal (INTLR) is generated when the comparison result is correct. If the comparison result is illegal, however, a status interrupt request signal (INTLS) is generated instead of a reception complete interrupt request signal (INTLR), a checksum error flag (UF0CSE) is set, and the checksum is stored into the UF0RX register.

<R> **Note** When the enhanced checksum is selected, the value of the UF0ID register is set to its initial value for calculation at the time transfer starts.

Figure 14-61. Automatic Checksum Error Occurrence Example (Response Reception)



<R> **Remark** m = 1 to 8

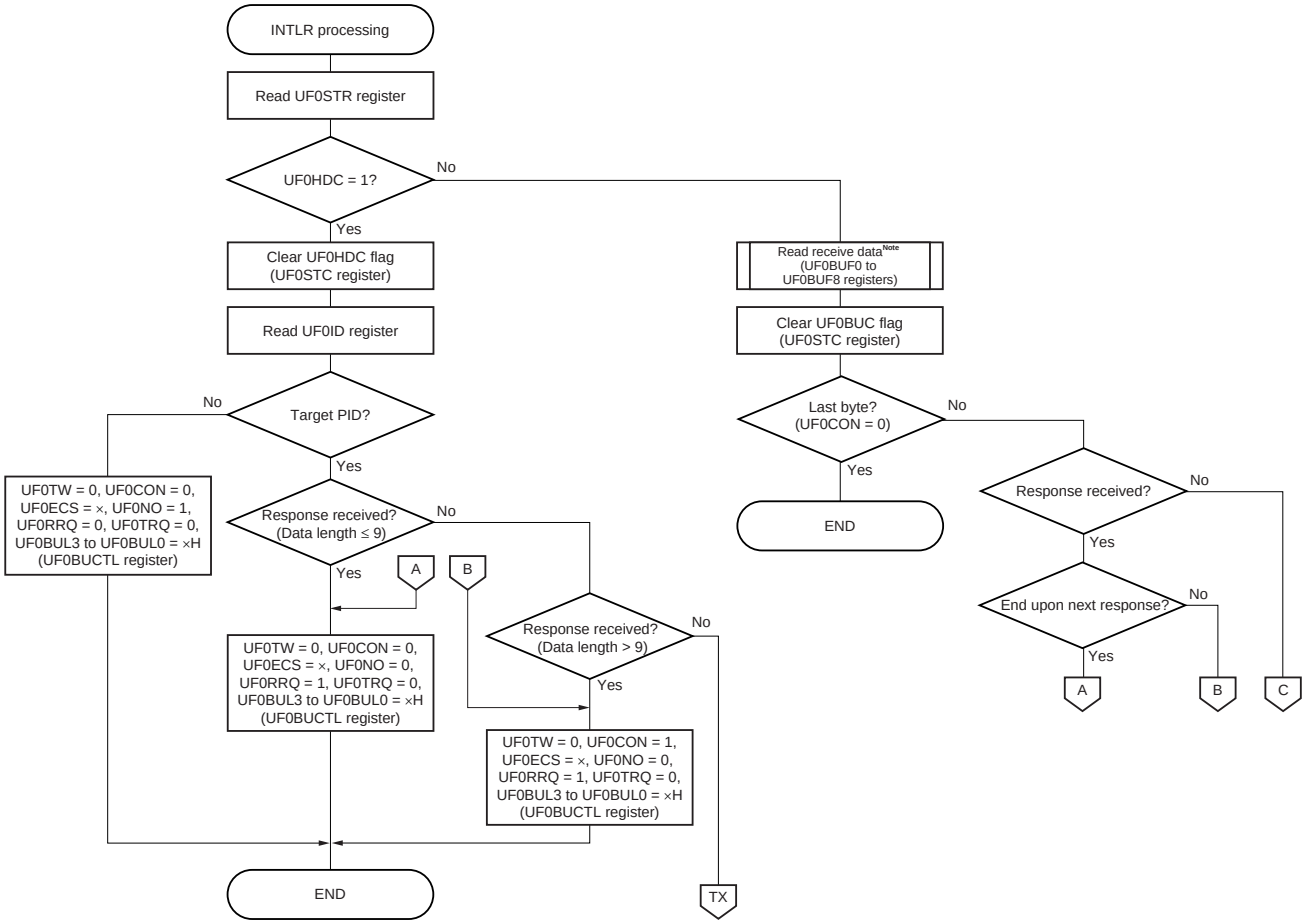
14.7.5 Multi-byte response transmission/reception function

In normal LIN communication, a response is no more than 9 bytes (including the checksum field); but in automatic baud rate mode (UF0MD1, UF0MD0 = 11B), responses of at least 10 bytes can be transmitted and received.

The processing flow of initial settings and INTLS generation is same as the basic processing flow. See **14.7 LIN Communication Automatic Baud Rate Mode**.

The response preparation error detection function, ID parity check function, and automatic checksum function are valid.

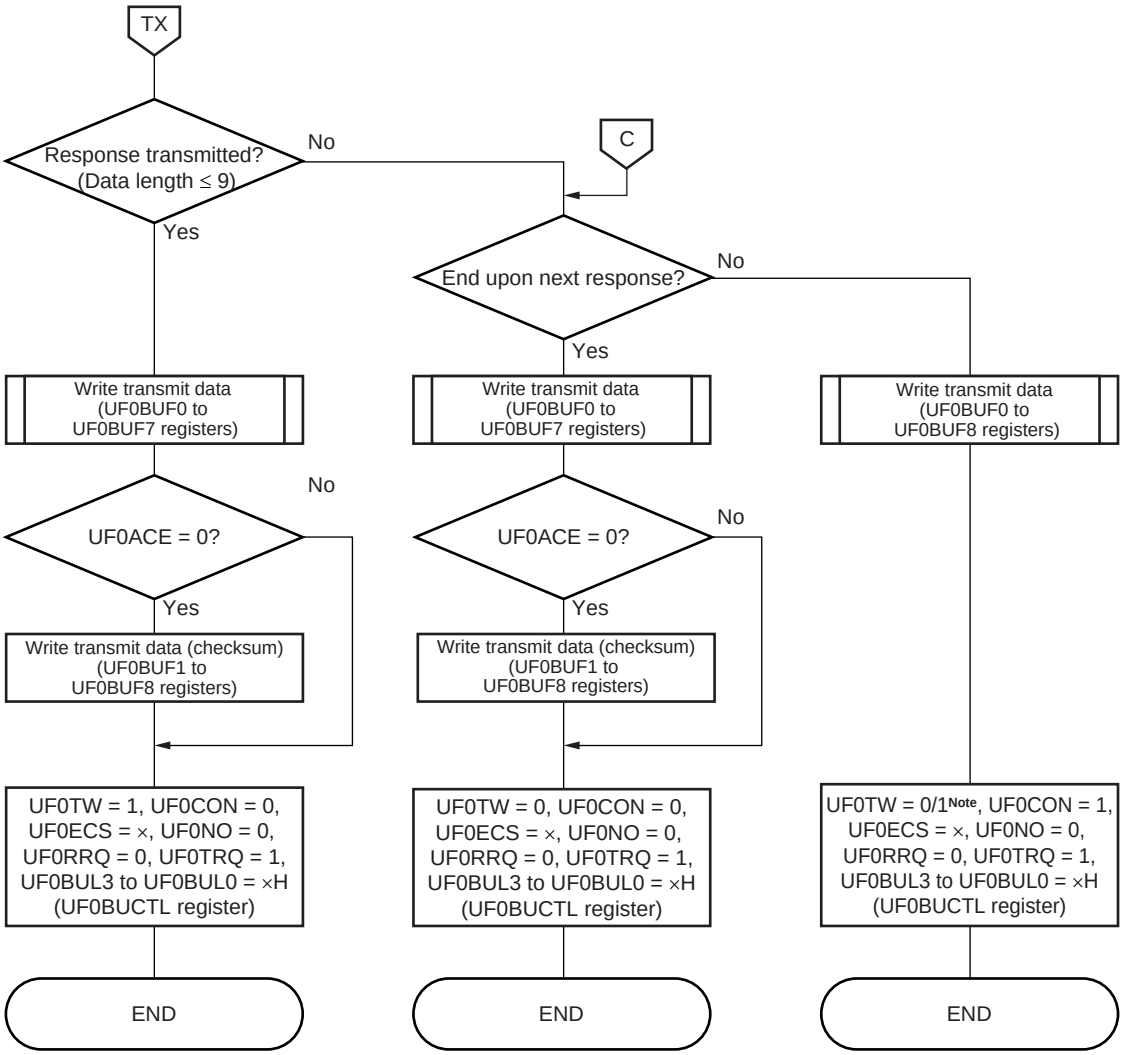
Figure 14-62. Multi-Byte Transmission/Reception Processing Flow Example (1/2)



Note This can be omitted.

Remark ×: don't care

Figure 14-62. Multi-Byte Transmission/Reception Processing Flow Example (2/2)



Note Set UF0TW to 1 during only the first data transmission after PID reception.

Cautions 1. When the buffer length bits (UF0BUL3 to UF0BUL0) have been set to "0" or "10 to 15", reception or transmission of nine bytes is performed. When the buffer length is set to "1 to 8", buffers of the number of bytes set are used in ascending order of the buffer numbers.

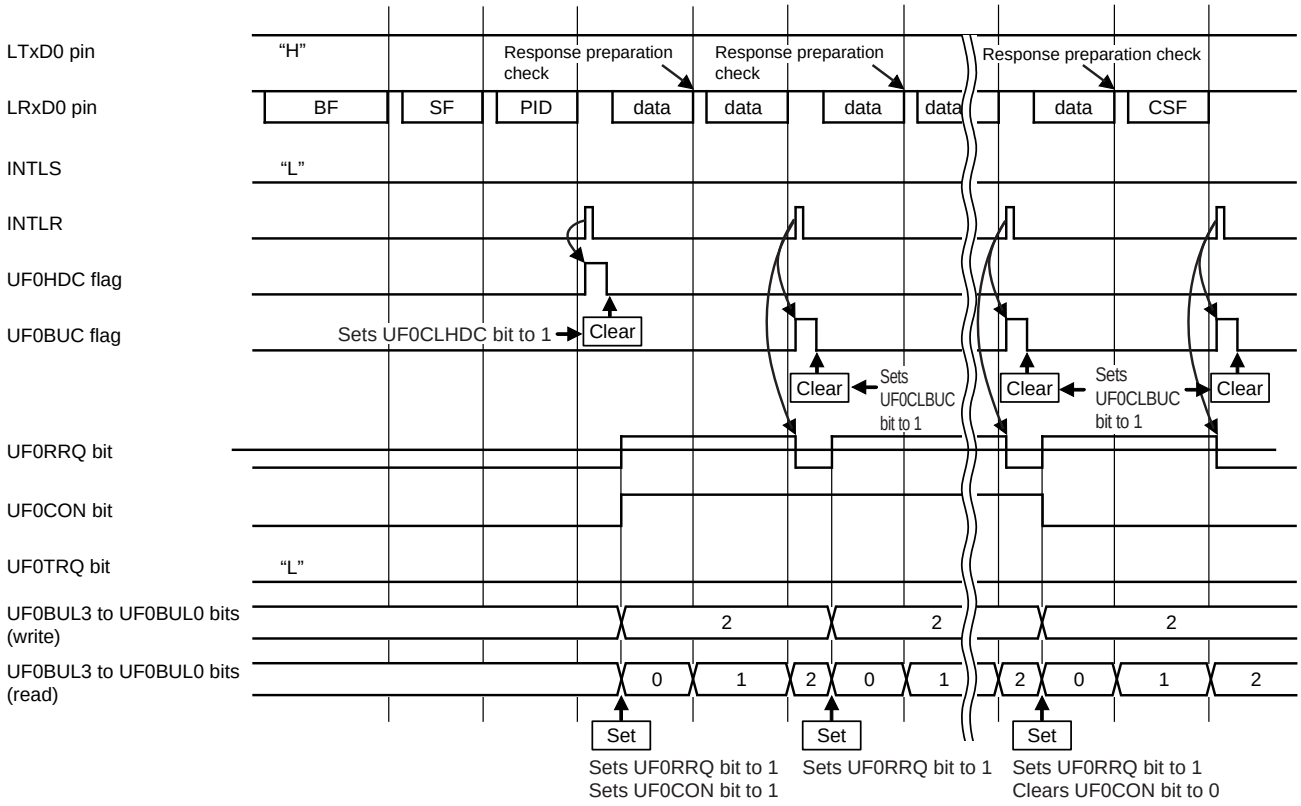
Example: When UF0BUL3 to UF0BUL0 are set to "1", data is always stored only into the UF0BUF0 register.

2. Do not set the UF0RRQ bit before completion of receive data acquisition.

3. Setting the UF0TW bit is prohibited, except when operation is switched to response transmission after header reception.

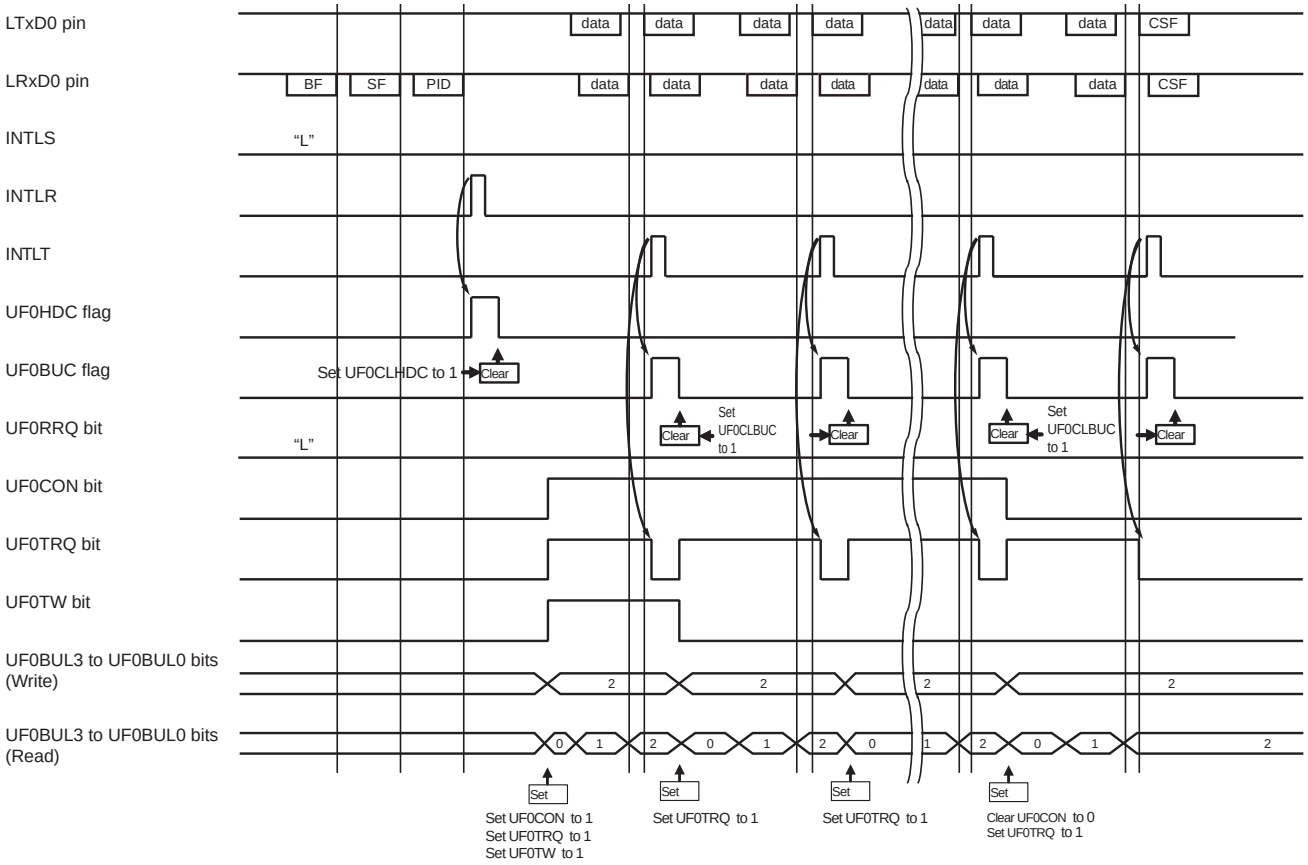
Remark ×: don't care

Figure 14-63. Multi-Byte Reception Implementation Example



Caution When UF0BUL3 to UF0BUL0 are "2", data is always stored into UF0BUF0 and UF0BUF1.
If read processing of the receive data is not performed in time, make adjustments such as setting UF0BUL3 to UF0BUL0 to "1".

Figure 14-64. Multi-Byte Transmission Implementation Example



Caution When UF0BUL3 to UF0BUL0 are "2", data of the UF0BUF0 and UF0BUF1 bits are always transmitted and stored.

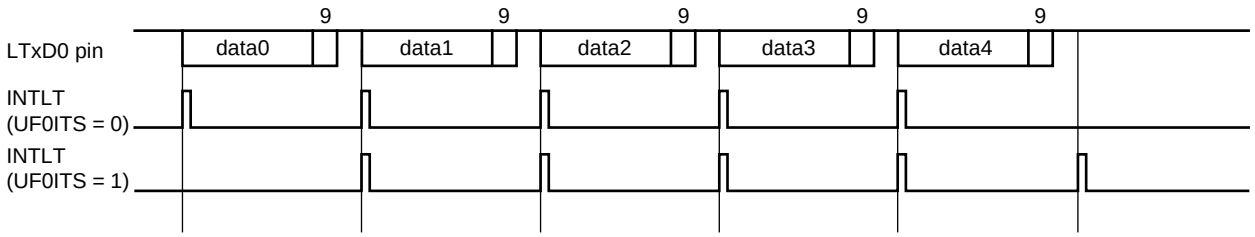
14.8 Expansion Bit Mode

When in normal UART mode (UF0MD1, UF0MD0 = 00B), data of 9-bit lengths can be transmitted or received by setting the expansion bit enable bit (UF0EBE). See **14.5.1 Data format** for the communication data format.

14.8.1 Expansion bit mode transmission

When in expansion bit mode (UF0CL = UF0EBE = 1), transmission in 9-bit lengths is started by writing 9-bit data to the UF0TX register.

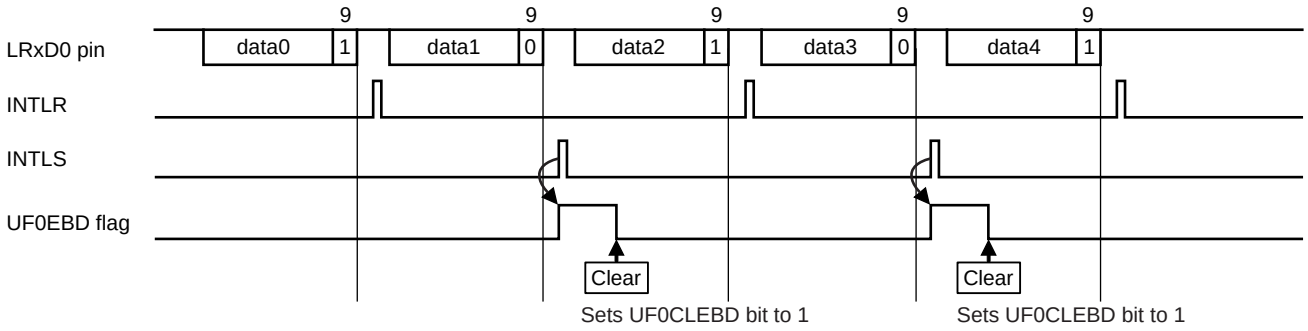
Figure 14-65. Expansion Bit Mode Transmission Example (LSB First)



14.8.2 Expansion bit mode reception (no data comparison)

When in expansion bit mode (UF0CL = UF0EBE = 1) and expansion bit data comparison is disabled (UF0EBC = 0), reception in 9-bit lengths can always be performed without data comparison. When a level set by using the expansion bit detection level select bit (UF0EBL) is detected, a status interrupt request signal (INTLS) is generated upon completion of data reception, and an expansion bit detection flag (UF0EBD) is set. When an inverted value of the expansion bit detection level is detected, a reception complete interrupt request signal (INTLR) is generated. In either case, the receive data is stored into the UF0RX register if no overrun error has occurred.

Figure 14-66. Expansion Bit Mode Reception (No Data Comparison) Example (LSB First, UF0EBL = 0)



Cautions 1. When a reception error (parity error, framing error, or overrun error) occurs at receive data 0, 2, or 4, a status interrupt request signal (INTLS) is generated instead of a reception complete interrupt request signal (INTLR), and the error flag is updated.

2. When a reception error (parity error, framing error, or overrun error) occurs at receive data 1 or 3, the error flag is also updated.

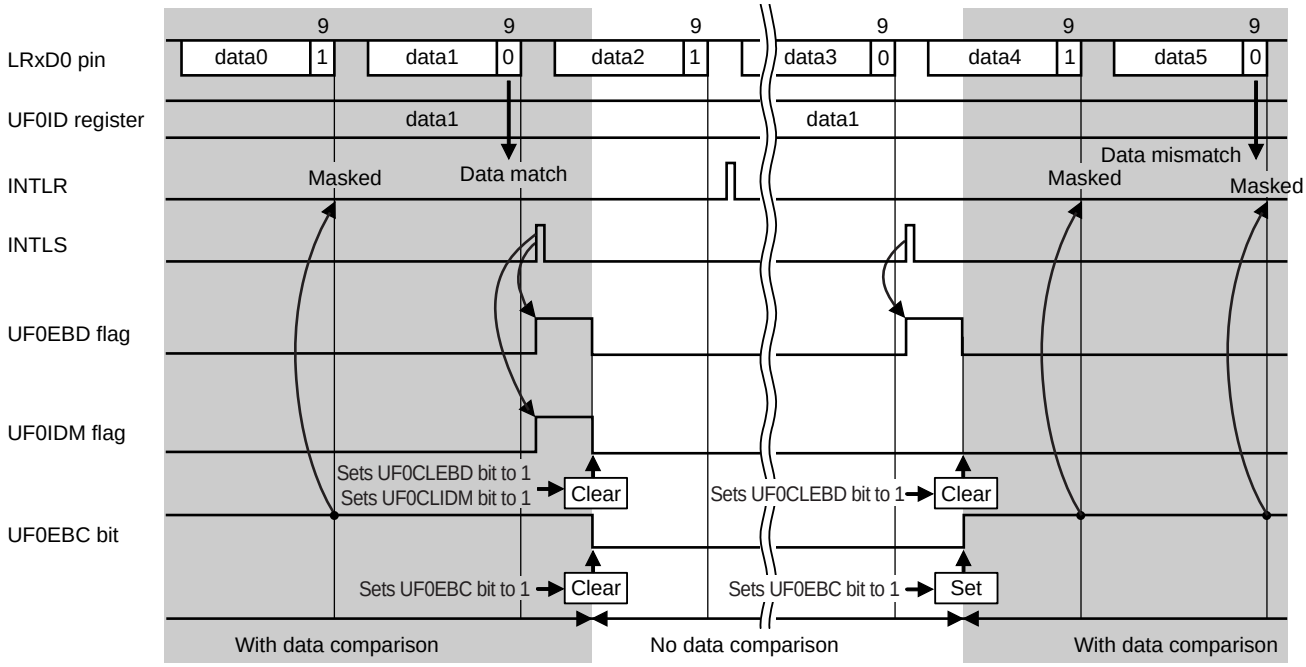
14.8.3 Expansion bit mode reception (with data comparison)

When in expansion bit mode (UF0CL = UF0EBE = 1) and expansion bit data comparison is enabled (UF0EBC = 1), if a level set by using the expansion bit detection level select bit (UF0EBL) is detected, 8 bits excluding the receive data expansion bit are compared with the value of the UF0ID register set in advance.

If the comparison results have matched, a status interrupt request signal (INTLS) is generated, an expansion bit ID match flag (UF0IDM) and an expansion bit detection flag (UF0EBD) are set, and the receive data is stored into UF0RX. If the comparison results do not match, no interrupt is generated, no flag is updated, and the receive data is not stored.

Interrupts (INTLR, INTLS) are generated upon all subsequent completions of data receptions and data can be received by disabling expansion bit data comparison (UF0EBC = 0) via the status interrupt servicing when the comparison results have matched. End the processing before completion of the next data reception, because data will be omitted if the UF0EBC bit is changed after the next data reception has been completed.

Figure 14-67. Expansion Bit Mode Reception (with Data Comparison) Example (LSB First, UF0EBL = 0)



Cautions 1. When a reception error (parity error, framing error, or overrun error) occurs at receive data 2, a status interrupt request signal (INTLS) is generated instead of a reception complete interrupt request signal (INTLR), and the error flag is updated.

2. When a reception error (parity error, framing error, or overrun error) occurs at receive data 1 or 3, the error flag is also updated. When a reception error occurs at receive data 0, 4, or 5, the error flag is not updated.

14.9 Receive Data Noise Filter

The probability of malfunctioning due to noise becomes high with UART reception, because no communication clock exists. The noise filter is used to eliminate noise in a communication bus and reduce false reception of data. The noise filter becomes valid by clearing the receive data noise filter use selection bit (UF0RXFL) to “0”.

A start bit and receive data input from a serial data input pin (LRxD0) are sampled with a clock (prescaler clock) divided by using a prescaler.

When the same sampling value is read twice, the match detector output changes and the receive data is sampled as the input data. Therefore, data not exceeding 2 clock width is judged to be noise and is not delivered to the internal circuit (see **Figure 14-68**). See **14.10 (1) (a) Prescaler clock (f_{UCLK})** regarding the base clock.

Figure 14-68. Noise Filter Circuit Example

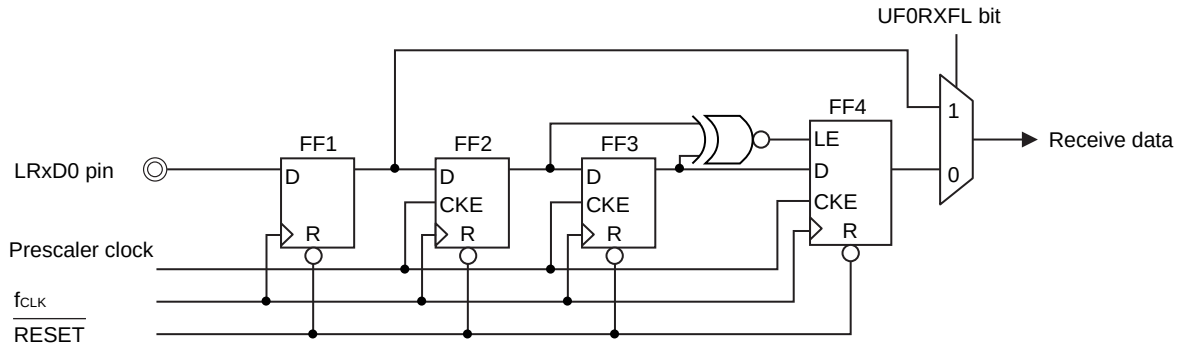
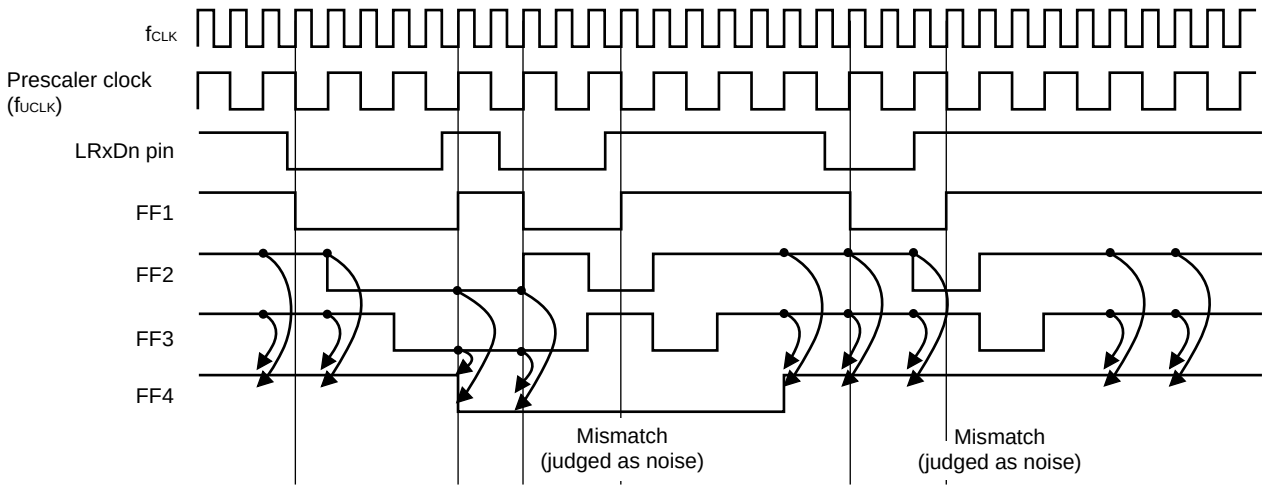


Figure 14-69. Noise Filter Timing Chart Example (UF0PRS = 1)



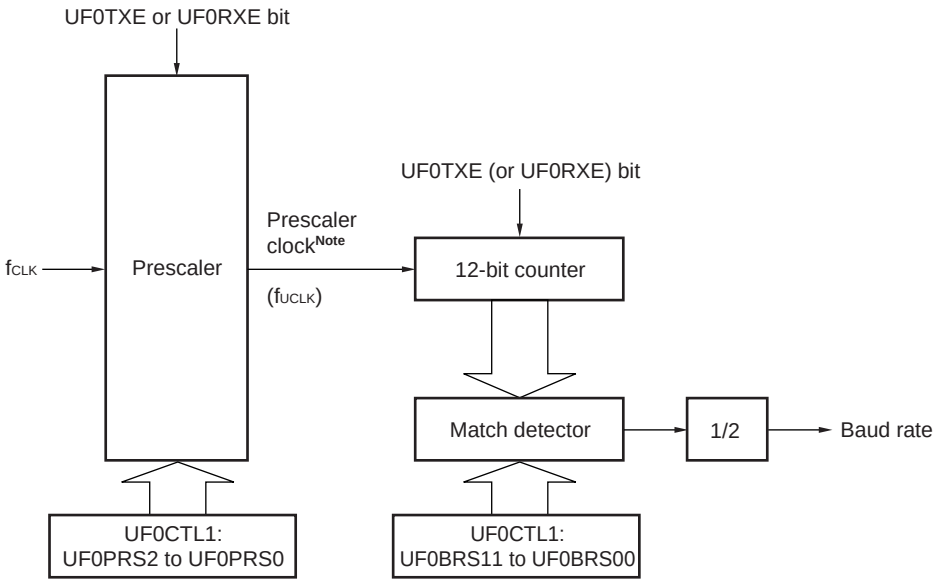
14.10 Dedicated Baud Rate Generator

The dedicated baud rate generator consists of a 3-bit prescaler block and a 12-bit programmable counter, and generates a serial clock during transmission and reception with LIN-UART0. Regarding the serial clock, a dedicated baud rate generator output can be selected for each channel.

There is a 12-bit counter for transmission and another one for reception.

(1) Configuration of baud rate generator

Figure 14-70. Configuration of Baud Rate Generator



Note Clock that divides f_{CLK} by 1, 2, 4, 8, 16, 32, 64, or 128

In automatic baud rate mode, confirm that the receive pin is high before setting the UF0RXE bit to 1.

(a) Prescaler clock (f_{UCLK})

When the UF0EN bit of the PER register is “1”, a clock divided by a frequency division value specified by using the UF0PRS2 to UF0PRS0 bits of the UF0CTL1 register is supplied to the 12-bit counter.

This clock is called the prescaler clock and its frequency is called f_{UCLK}.

(b) Serial clock generation

A serial clock can be generated by setting the UF0CTL1 register.

The frequency division value for the 12-bit counter can be set by using the UF0BRS11 to UF0BRS00 bits of the UF0CTL1 register.

(2) LIN-UART0 control register 1 (UF0CTL1)

The UF0CTL1 register is a 16-bit register that is used to control the baud rate of LIN-UART0.

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

Reset sets this register to 0FFFH.

Figure 14-71. Format of LIN-UART0 Control Register 1 (UF0CTL1)

Address: F0522H, F0523H After reset: 0000H R/W

	15	14	13	12	11	10	9	8
UF0CTL1	UF0PRS2	UF0PRS1	UF0PRS0	0	UF0BRS11	UF0BRS10	UF0BRS9	UF0BRS8
	7	6	5	4	3	2	1	0
	UF0BRS7	UF0BRS6	UF0BRS5	UF0BRS4	UF0BRS3	UF0BRS2	UF0BRS1	UF0BRS0

UF0PRS2	UF0PRS1	UF0PRS0	Prescaler clock frequency division value
0	0	0	No division (prescaler clock = f_{CLK})
0	0	1	Division by 2 (prescaler clock = f_{CLK}/2)
0	1	0	Division by 4 (prescaler clock = f_{CLK}/4)
0	1	1	Division by 8 (prescaler clock = f_{CLK}/8)
1	0	0	Division by 16 (prescaler clock = f_{CLK}/16)
1	0	1	Division by 32 (prescaler clock = f_{CLK}/32)
1	1	0	Division by 64 (prescaler clock = f_{CLK}/64)
1	1	1	Division by 128 (prescaler clock = f_{CLK}/128)

UF0 BRS11	UF0 BRS10	UF0 BRS09	UF0 BRS08	UF0 BRS07	UF0 BRS06	UF0 BRS05	UF0 BRS04	UF0 BRS03	UF0 BRS02	UF0 BRS01	UF0 BRS00	k[Note]	Serial clock
0	0	0	0	0	0	0	0	0	0	×	×	4	f_{UCLK}/4
0	0	0	0	0	0	0	0	0	1	0	0	4	f_{UCLK}/4
0	0	0	0	0	0	0	0	0	1	0	1	5	f_{UCLK}/5
:	:	:		:		:			:	:	:	:	:
1	1	1	1	1	1	1	1	1	1	1	0	4094	f_{UCLK}/4094
1	1	1	1	1	1	1	1	1	1	1	1	4095	f_{UCLK}/4095

Note Specified value

Cautions 1. Rewriting can be performed only when the UF0TXE and UF0RXE bits of the UF0CTL0 register are "0".

2. The baud rate is the value that results by further dividing the serial clock by 2.

3. Writing to UF0BRS11 to UF0BRS00 is invalid when in automatic baud rate mode.

Remarks 1. f_{UCLK} is the frequency division value of the prescaler clock selected by using the UF0PRS2 to UF0PRS0 bits.

2. In automatic baud rate mode (UF0MD1, UF0MD0 = 11B), the value after the baud rate has been set can be checked by reading UF0BRS11 to UF0BRS00 after header reception.

3. ×: don't care

(3) Baud rate

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{UCLK}}{2 \times k} \text{ [bps]}$$

f_{UCLK} = Frequency of prescaler clock selected by the UF0PRS2 to UF0PRS0 bits of the UF0CTL1 register
k = Value set by using the UF0BRS11 to UF0BRS00 bits of the UF0CTL1 register (k = 4, 5, 6, …, 4095)

(4) Baud rate error

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

Cautions 1. The baud rate error during transmission must be within the error tolerance on the receiving side.

2. The baud rate error during reception must satisfy the range indicated in (6) Allowable baud rate range during reception.

Example:
- CPU/peripheral hardware clock frequency = 24 MHz = 24,000,000 Hz
- Setting values
  f_{CLK} = 24 MHz
  Setting values of the UF0PRS2 to UF0PRS0 bits of the UF0CTL1 register = 001B ($f_{UCLK} = f_{CLK}/2$ = 12 MHz)
  Setting values of the UF0BRS11 to UF0BRS00 bits of the UF0CTL1 register = 000000100111B (k = 39)
- Target baud rate = 153,600 bps
- Baud rate = 12,000,000/(2 × 39)
  = 153,846 [bps]
- Error = (153,846/153,600 – 1) × 100
  = 0.160 [%]

(5) Baud rate setting example

Table 14-5. Baud Rate Generator Setting Data
(Normal Operation, f_{CLK} = 24 MHz, UF0PRS2 to UF0PRS0 = 0 to 3)

Target Baud Rate (bps)	UF0PRS2 to UF0PRS0							
	0		1		2		3	
	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)
300	–	–	–	–	–	–	–	–
600	–	–	–	–	–	–	2500	0.00
1200	–	–	–	–	2500	0.00	1250	0.00
2400	–	–	2500	0.00	1250	0.00	625	0.00
4800	2500	0.00	1250	0.00	625	0.00	313	–0.16
9600	1250	0.00	625	0.00	313	–0.16	156	0.16
19200	625	0.00	313	–0.16	156	0.16	78	0.16
31250	384	0.00	192	0.00	96	0.00	48	0.00
38400	313	–0.16	156	0.16	78	0.16	39	0.16
76800	156	0.16	78	0.16	39	0.16	20	–2.34
128000	94	–0.27	47	–0.27	23	1.90	12	–2.34
153600	78	0.16	39	0.16	20	–2.34	10	–2.34
312500	38	1.05	19	1.05	10	–4.00	5	–4.00
1000000	12	0.00	6	0.00	–	–	–	–

Table 14-6. Baud Rate Generator Setting Data
(Normal Operation, f_{CLK} = 24 MHz, UF0PRS2 to UF0PRS0 = 4 to 7)

Target Baud Rate (bps)	UF0PRS2 to UF0PRS0							
	4		5		6		7	
	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)
300	2500	0.00	1250	0.00	625	0.00	313	–0.16
600	1250	0.00	625	0.00	384	–0.16	156	0.16
1200	625	0.00	384	–0.16	313	0.16	78	0.16
2400	313	–0.16	313	0.16	156	0.16	625	0.16
4800	156	0.16	156	0.16	94	0.16	313	–2.34
9600	78	0.16	94	0.16	78	–2.34	156	–2.34
19200	39	0.16	78	–2.34	156	–2.34	78	–2.34
31250	24	0.00	192	0.00	96	0.00	–	–
38400	20	–2.34	156	–2.34	78	–2.34	–	–
76800	10	–2.34	78	–2.34	–	–	–	–
128000	6	–2.34	–	–	–	–	–	–
153600	5	–2.34	–	–	–	–	–	–
312500	–	–	–	–	–	–	–	–
1000000	–	–	–	–	–	–	–	–

Table 14-7. Baud Rate Generator Setting Data
(Normal Operation, f_{CLK} = 12 MHz, UF0PRS2 to UF0PRS0 = 0 to 3)

Target Baud Rate (bps)	UF0PRS2 to UF0PRS0							
	0		1		2		3	
	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)
300	–	–	–	–	–	–	2500	0.00
600	–	–	–	–	2500	0.00	1250	0.00
1200	–	–	2500	0.00	1250	0.00	625	0.00
2400	2500	0.00	1250	0.00	625	0.00	313	–0.16
4800	1250	0.00	625	0.00	313	–0.16	156	0.16
9600	625	0.00	313	–0.16	156	0.16	78	0.16
19200	313	–0.16	156	0.16	78	0.16	39	0.16
31250	192	0.00	96	0.00	48	0.00	24	0.00
38400	156	0.16	78	0.16	39	0.16	20	–2.34
76800	78	0.16	39	0.16	20	–2.34	10	–2.34
128000	47	–0.27	23	1.90	12	–2.34	6	–2.34
153600	39	0.16	20	–2.34	10	–2.34	5	–2.34
312500	19	1.05	10	–4.00	5	–4.00	–	–
1000000	6	0.00	–	–	–	–	–	–

Table 14-8. Baud Rate Generator Setting Data
(Normal Operation, f_{CLK} = 12 MHz, UF0PRS2 to UF0PRS0 = 4 to 7)

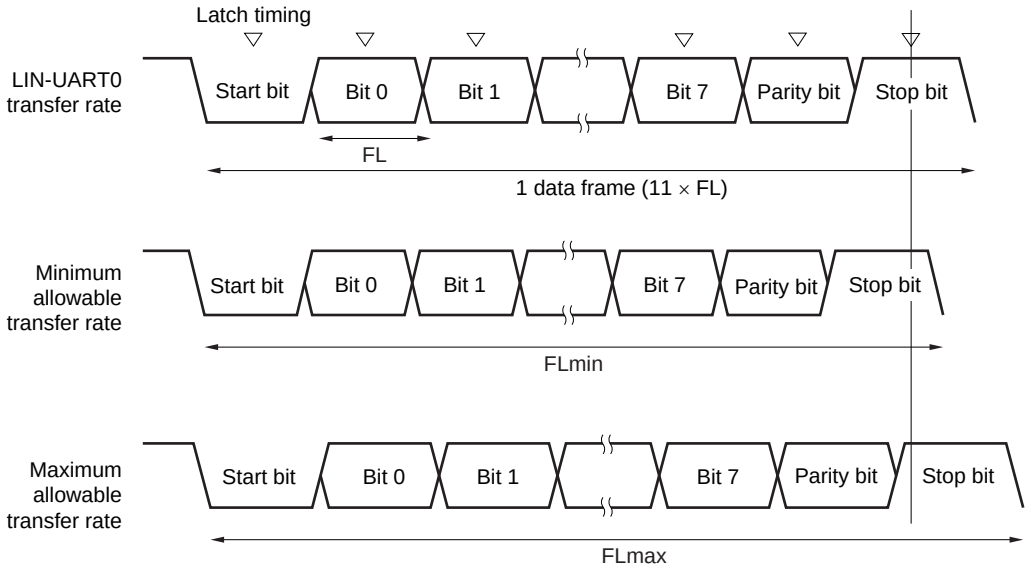
Target Baud Rate (bps)	UF0PRS2 to UF0PRS0							
	4		5		6		7	
	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)	UF0BRS11 to UF0BRS00	ERR (%)
300	1250	0.00	625	0.00	313	–0.16	156	0.16
600	625	0.00	313	–0.16	156	0.16	78	0.16
1200	313	–0.16	156	0.16	78	0.16	39	0.16
2400	156	0.16	78	0.16	39	0.16	20	–2.34
4800	78	0.16	39	0.16	20	–2.34	10	–2.34
9600	39	0.16	20	–2.34	10	–2.34	5	–2.34
19200	20	–2.34	10	–2.34	5	–2.34	–	–
31250	12	0.00	6	0.00	–	–	–	–
38400	10	–2.34	5	–2.34	–	–	–	–
76800	5	–2.34	–	–	–	–	–	–
128000	–	–	–	–	–	–	–	–
153600	–	–	–	–	–	–	–	–
312500	–	–	–	–	–	–	–	–
1000000	–	–	–	–	–	–	–	–

(6) Allowable baud rate range during reception

The baud rate error range at the destination that is allowable during reception is shown below.

Caution The baud rate error during reception must be set within the allowable error range using the following equation.

Figure 14-72 Allowable Baud Rate Range During Reception



In the figure above, the bits from the start bit to the stop bit is 11 bits long.

As shown in Figure 14-72, the receive data latch timing is determined by the counter set using the UF0CTL1 register following start bit detection. The transmit data can be normally received if up to the last data (stop bit) can be received in time for this latch timing.

When this is applied to 11-bit reception while the data bit length is 8 bits, the following is the theoretical result.

$FL = (Brate)^{-1}$

Brate: LIN-UART0 baud rate

k: Setting value of UF0CTL1

FL: 1-bit data length

Latch timing margin: 2 clocks

Minimum allowable transfer rate: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20\ k} FL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

Table 14-9 shows the allowable baud rate error between LIN-UART0 and the transmission source calculated from the above-described equations for obtaining the minimum and maximum baud rate values.

Table 14-9. Maximum/Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error			Minimum Allowable Baud Rate Error		
	BN = 9	BN = 11	BN = 12	BN = 9	BN = 11	BN = 12
4	+2.85%	+2.32%	+2.12%	–3.03%	–2.43%	–2.22%
8	+4.34%	+3.52%	+3.22%	–4.47%	–3.61%	–3.29%
16	+5.10%	+4.14%	+3.78%	–5.18%	–4.19%	–3.82%
64	+5.68%	+4.60%	+4.20%	–5.70%	–4.61%	–4.21%
128	+5.78%	+4.68%	+4.27%	–5.79%	–4.69%	–4.28%
256	+5.83%	+4.72%	+4.31%	–5.83%	–4.72%	–4.31%
512	+5.85%	+4.74%	+4.33%	–5.86%	–4.74%	–4.33%
1024	+5.87%	+4.75%	+4.33%	–5.87%	–4.75%	–4.33%
2048	+5.87%	+4.75%	+4.34%	–5.87%	–4.75%	–4.34%
4095	+3.42%	+4.75%	+4.34%	–3.59%	–4.75%	–4.34%

Remarks 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.

2. BN: Number of bits from the start bit to the stop bit

K: Setting values of UF0CTL1.UF0BRS[11:0]

14.11 Cautions for Use

(1) Execute a STOP instruction during a LIN-UART operation after stopping LIN-UART.

(2) Start up the LIN-UART0 in the following sequence.
<1> Set the ports.
<2> Set PER0.LINnEN to 1.
<3> Set UF0CTL0.UF0TXE to 1, and UF0CTL0.UF0RXE to 1.

(3) Stop the LIN-UART0 in the following sequence.
<1> Set UF0CTL0.UF0TXE to 0, and UF0CTL0.UF0RXE to 0.
<2> Set PER1.LINnEN to 0.
<3> Set the ports. (It is not a problem if port setting is not changed.)

(4) In transmit mode (UF0CTL0.UF0TXE = 1), do not overwrite the same value to the UF0TX register by software because transmission starts by writing to this register. To transmit the same value continuously, overwrite the same value.

CHAPTER 15 SERIAL INTERFACE IICA

The number of channels of the serial Interface IICA differs, depending on the product.

	20-pin	30, 32, 48, 64-pin
channels	–	1 ch

Caution Most of the following descriptions in this chapter use the 64-pin products as an example.

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 (SCLA0) line and a serial data bus (SDAA0) 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 SCLA0 and SDAA0 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 (INTIICA0) 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 WUP0 bit of IICA control register 01 (IICCTL01).

Figure 15-1 shows a block diagram of serial interface IICA.

Figure 15-1. Block Diagram of Serial Interface IICA

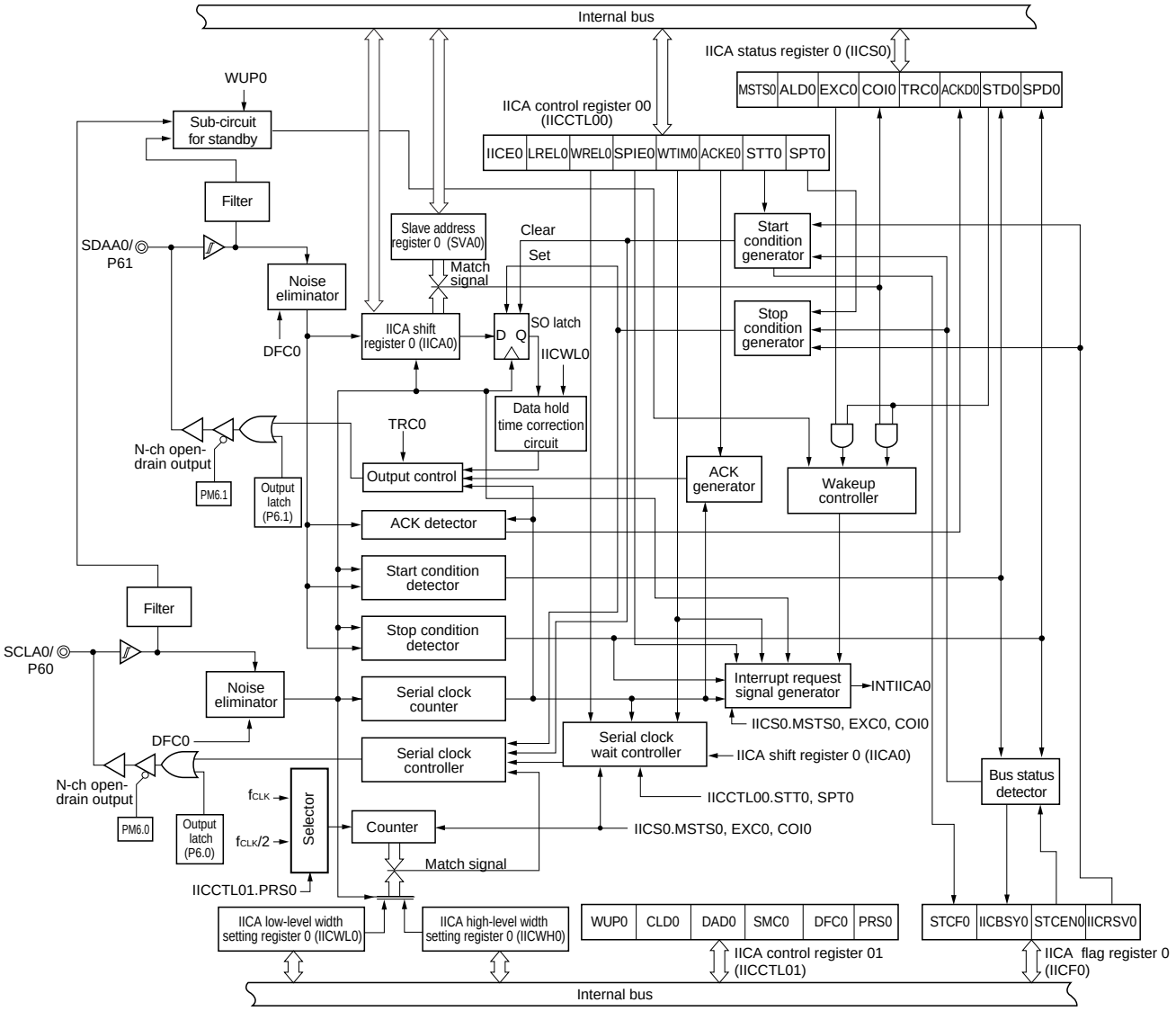
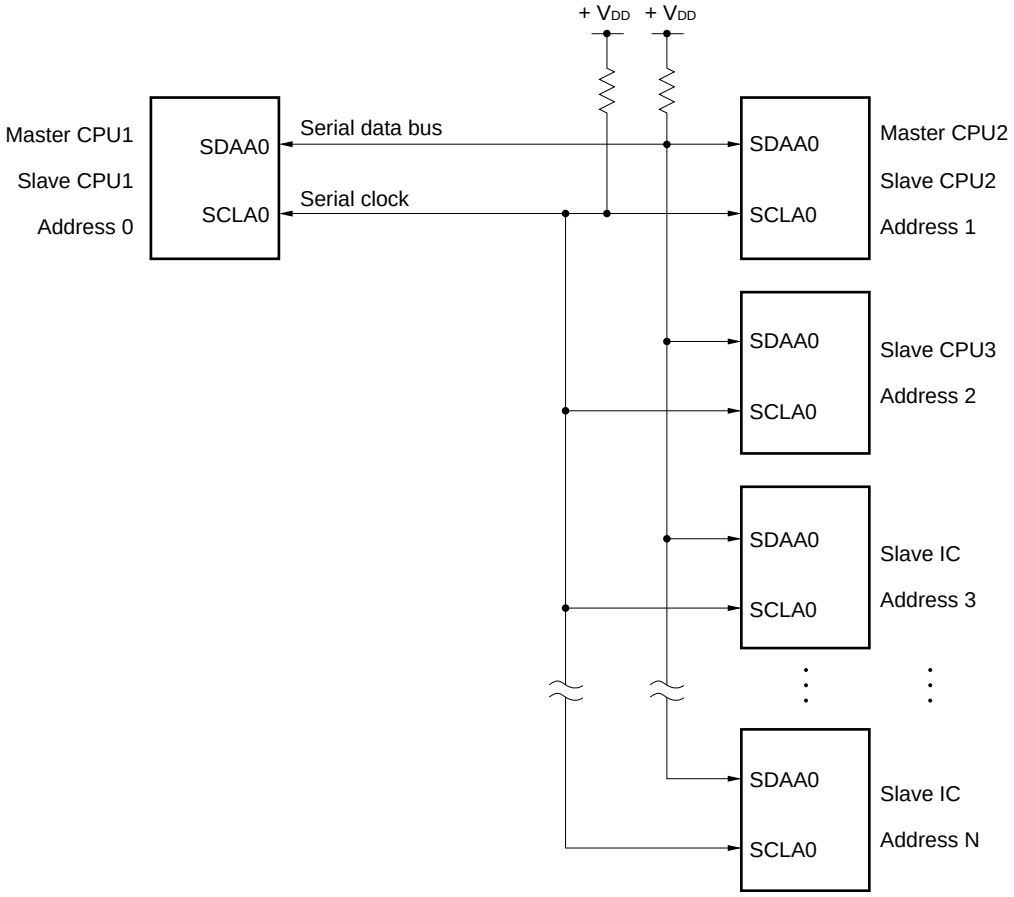


Figure 15-2 shows a serial bus configuration example.

Figure 15-2. Serial Bus Configuration Example Using I²C Bus



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 0 (IICA0) Slave address register 0 (SVA0)
Control registers	Peripheral enable register 0 (PER0) IICA control register 00 (IICCTL00) IICA status register 0 (IICS0) IICA flag register 0 (IICF0) IICA control register 01 (IICCTL01) IICA low-level width setting register 0 (IICWL0) IICA high-level width setting register 0 (IICWH0) Port mode register 6 (PM6)

(1) IICA shift register 0 (IICA0)

The IICA0 register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICA0 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 IICA0 register.

Cancel the wait state and start data transfer by writing data to the IICA0 register during the wait period.

The IICA0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears IICA0 to 00H.

Figure 15-3. Format of IICA Shift Register 0 (IICA0)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA0								

Cautions 1. Do not write data to the IICA0 register during data transfer.

2. Write or read the IICA0 register only during the wait period. Accessing the IICA0 register in a communication state other than during the wait period is prohibited. When the device serves as the master, however, the IICA0 register can be written only once after the communication trigger bit (STT0) is set to 1.

3. When communication is reserved, write data to the IICA0 register after the interrupt triggered by a stop condition is detected.

(2) Slave address register 0 (SVA0)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

The SVA0 register can be set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STD0 = 1 (while the start condition is detected).

Reset signal generation clears the SVA0 register to 00H.

Figure 15-4. Format of Slave Address Register 0 (SVA0)

Address: F0234H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0	A6	A5	A4	A3	A2	A1	A0	0Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDAA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA0) when the address received by this register matches the address value set to the slave address register 0 (SVA0) 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 (INTIICA0).

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 WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by the SPIE0 bit)

Remark WTIM0 bit: Bit 3 of IICA control register 00 (IICCTL00)
SPIE0 bit: Bit 4 of IICA control register 00 (IICCTL00)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLA0 pin from a sampling clock.

(8) Serial clock wait controller

This circuit controls the wait timing.

(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 STT0 bit is set to 1.

However, in the communication reservation disabled status (IICRSV bit = 1), when the bus is not released (IICBSY bit = 1), start condition requests are ignored and the STCF bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPT0 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 STCEN bit.

Remark STT0 bit: Bit 1 of IICA control register 00 (IICCTL00)
SPT0 bit: Bit 0 of IICA control register 00 (IICCTL00)
IICRSV bit: Bit 0 of IICA flag register 0 (IICF0)
IICBSY bit: Bit 6 of IICA flag register 0 (IICF0)
STCF bit: Bit 7 of IICA flag register 0 (IICF0)
STCEN bit: Bit 1 of IICA flag register 0 (IICF0)

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 00 (IICCTL00)
- IICA flag register 0 (IICF0)
- IICA status register 0 (IICS0)
- IICA control register 01 (IICCTL01)
- IICA low-level width setting register 0 (IICWL0)
- IICA high-level width setting register 0 (IICWH0)
- Port mode register 6 (PM6)

(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 IICA is used, be sure to set bit 4 (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	RTCEN	0	ADCEN	IICA0EN Note	SAU1EN Note	SAU0EN	0	TAU0EN

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

Note Those are not provided in the 20-pin products.

Cautions 1. When setting serial interface IICA, be sure to set the IICA0EN bit to 1 first. If IICA0EN = 0, writing to a control register of serial interface IICA is ignored, and, even if the register is read, only the default value is read (except for port mode register 6 (PM6)).

2. Be sure to clear the following bits to 0.

20-pin products: bits 1, 3, 4, 6

30, 32-pin products: bits 1, 6

48, 64-pin products: bits 1, 6

(2) IICA control register 00 (IICCTL00)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

The IICCTL00 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE0, WTIM0, and ACKE0 bits while IICE0 = 0 or during the wait period. These bits can be set at the same time when the IICE0 bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Figure 15-6. Format of IICA Control Register 00 (IICCTL00) (1/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTL00	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	Stop operation. Reset the IICA status register 0 (IICS0)[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLA0 and SDAA0 lines are at high level.	
Condition for clearing (IICE0 = 0)	Condition for setting (IICE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0[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 SCLA0 and SDAA0 lines are set to high impedance. The following flags of IICA control register 00 (IICCTL00) and the IICA status register 0 (IICS0) are cleared to 0. • STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0
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 (LREL0 = 0)	Condition for setting (LREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL0[Notes 2, 3]	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When the WREL0 bit is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC0 = 1), the SDAA0 line goes into the high impedance state (TRC0 = 0).	
Condition for clearing (WREL0 = 0)	Condition for setting (WREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICA status register 0 (IICS0), the STCF and IICBSY bits of the IICA flag register 0 (IICF0), and the CLD0 and DAD0 bits of IICA control register 01 (IICCTL01) are reset.

2. The signal of this bit is invalid while IICE0 is 0.

3. When the LREL0 and WREL0 bits are read, 0 is always read.

Caution If the operation of I^2C is enabled (IICE0 = 1) when the SCLA0 line is high level, the SDAA0 line is low level, and the digital filter is turned on (DFC0 bit of IICCTL01 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LREL0 bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICE0 = 1).

Figure 15-6. Format of IICA Control Register 00 (IICCTL00) (2/4)

SPIE0[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the WUP0 bit of IICA control register 01 (IICCTL01) is 1, no stop condition interrupt will be generated even if SPIE0 = 1.	
Condition for clearing (SPIE0 = 0)	Condition for setting (SPIE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0[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 ($\overline{ACK}$) 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 (WTIM0 = 0)	Condition for setting (WTIM0 = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0[Notes 1, 2]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAA0 line is set to low level.
Condition for clearing (ACKE0 = 0)	Condition for setting (ACKE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0 is 0. Set this bit during that period.

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.

Figure 15-6. Format of IICA Control Register 00 (IICCTL00) (3/4)

STT0[Note]	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSY = 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 (IICRSV = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV = 1) Even if this bit is set (1), the STT0 bit is cleared and the STT0 clear flag (STCF) 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 ACKE0 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 (SPT0).
- Setting the STT0 bit to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (STT0 = 0)	Condition for setting (STT0 = 1)
• Cleared by setting the STT0 bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Note The signal of this bit is invalid while IICE0 is 0.

Remarks 1. Bit 1 (STT0) becomes 0 when it is read after data setting.

2. IICRSV: Bit 0 of IIC flag register 0 (IICF0)
STCF: Bit 7 of IIC flag register 0 (IICF0)

Figure 15-6. Format of IICA Control Register 00 (IICCTL00) (4/4)

SPT0	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 ACKE0 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 (STT0).
- The SPT0 bit can be set to 1 only when in master mode.
- When the WTIM0 bit has been cleared to 0, if the SPT0 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 WTIM0 bit should be changed from 0 to 1 during the wait period following the output of eight clocks, and the SPT0 bit should be set to 1 during the wait period that follows the output of the ninth clock.
- Setting the SPT0 bit to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (SPT0 = 0)	Condition for setting (SPT0 = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Caution When bit 3 (TRC0) of the IICA status register 0 (IICS0) is set to 1 (transmission status), bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the wait performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register 0.

Remark Bit 0 (SPT0) becomes 0 when it is read after data setting.

(3) IICA status register 0 (IICS0)

This register indicates the status of I^2C.

The IICS0 register is read by a 1-bit or 8-bit memory manipulation instruction only when STT0 = 1 and during the wait period.

Reset signal generation clears this register to 00H.

Caution Reading the IICS0 register while the address match wakeup function is enabled (WUP0 = 1) in STOP mode is prohibited. When the WUP0 bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA0 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 (SPIE0 = 1) the interrupt generated by detecting a stop condition and read the IICS0 register after the interrupt has been detected.

Remark STT0: bit 1 of IICA control register 00 (IICCTL00)
WUP0: bit 7 of IICA control register 01 (IICCTL01)

Figure 15-7. Format of IICA Status Register 0 (IICS0) (1/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 (arbitration loss) • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	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 MSTS0 bit is cleared.

Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after the IICS0 register is read[Note] • When the IICE0 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 IICS0 register. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

Figure 15-7. Format of IICA Status Register 0 (IICS0) (2/3)

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 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).

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register 0 (SVA0)) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDAA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDAA0 line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by WREL0 = 1[Note] (wait cancel) • When the ALD0 bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS0, EXC0, COI0 = 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 (TRC0) of the IICA status register 0 (IICS0) is set to 1 (transmission status), bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the wait performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register 0.

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

Figure 15-7. Format of IICA Status Register 0 (IICS0) (3/3)

ACKD0	Detection of acknowledge ($\overline{ACK}$)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD0 = 0)	Condition for setting (ACKD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• After the SDAA0 line is set to low level at the rising edge of SCLA0 line's ninth clock

STD0	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 (STD0 = 0)	Condition for setting (STD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	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 (SPD0 = 0)	Condition for setting (SPD0 = 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 IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

(4) IICA flag register 0 (IICF0)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

The IICF0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STT0 clear flag (STCF) and I^2C bus status flag (IICBSY) bits are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function.

The STCEN bit can be used to set the initial value of the IICBSY bit.

The IICRSV and STCEN bits can be written only when the operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) = 0). When operation is enabled, the IICF0 register can be read.

Reset signal generation clears this register to 00H.

Figure 15-8. Format of IICA Flag Register 0 (IICF0)

Address: FFF52H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	STT0 clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STT0 flag

Condition for clearing (STCF0 = 0)	Condition for setting (STCF0 = 1)
• Cleared by STT0 = 1 • When IICE0 = 0 (operation stop) • Reset	• Generating start condition unsuccessful and the STT0 bit cleared to 0 when communication reservation is disabled (IICRSV0 = 1).

IICBSY0	I^2C bus status flag
0	Bus release status (communication initial status when STCEN0 = 1)
1	Bus communication status (communication initial status when STCEN0 = 0)

Condition for clearing (IICBSY0 = 0)	Condition for setting (IICBSY0 = 1)
• Detection of stop condition • When IICE0 = 0 (operation stop) • Reset	• Detection of start condition • Setting of the IICE0 bit when STCEN0 = 0

STCEN0	Initial start enable trigger
0	After operation is enabled (IICE0 = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE0 = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN0 = 0)	Condition for setting (STCEN0 = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSV0	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV0 = 0)	Condition for setting (IICRSV0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to the STCEN bit only when the operation is stopped (IICE0 = 0).

2. As the bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1, when generating the first start condition (STT0 = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to IICRSV only when the operation is stopped (IICE0 = 0).

Remark STT0: Bit 1 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

(5) IICA control register 01 (IICCTL01)

This register is used to set the operation mode of I^2C and detect the statuses of the SCLA0 and SDAA0 pins.

The IICCTL01 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only.

Set the IICCTL01 register, except the WUP0 bit, while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation clears this register to 00H.

Figure 15-9. Format of IICA Control Register 01 (IICCTL01) (1/2)

Address: F0231H After reset: 00H R/W[Note 1]

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
IICCTL01	WUP0	0	CLD0	DAD0	SMC0	DFC0	0	PRS0

WUP0	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 WUP0 = 1, execute the STOP instruction at least three clocks after setting (1) the WUP0 bit (see **Figure 15-22 Flow When Setting WUP0 = 1**).

Clear (0) the WUP0 bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUP0 bit. (The wait must be released and transmit data must be written after the WUP0 bit has been cleared (0).)

The interrupt timing when the address has matched or when an extension code has been received, while WUP0 = 1, is identical to the interrupt timing when WUP0 = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP0 = 1, a stop condition interrupt is not generated even if the SPIE0 bit is set to 1.

When WUP0 = 0 is set by a source other than an interrupt from serial interface IICA, operation as the master device cannot be performed until the subsequent start condition or stop condition is detected. Do not output a start condition by setting (1) the STT0 bit, without waiting for the detection of the subsequent start condition or stop condition.

Condition for clearing (WUP0 = 0)	Condition for setting (WUP0 = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when the MSTS0, EXC0, and COI0 bits are “0”, and the STD0 bit also “0” (communication not entered))[Note 2]

Notes 1. Bits 4 and 5 are read-only.

2. The status of the IICA status register 0 (IICS0) must be checked and the WUP0 bit must be set during the period shown below.

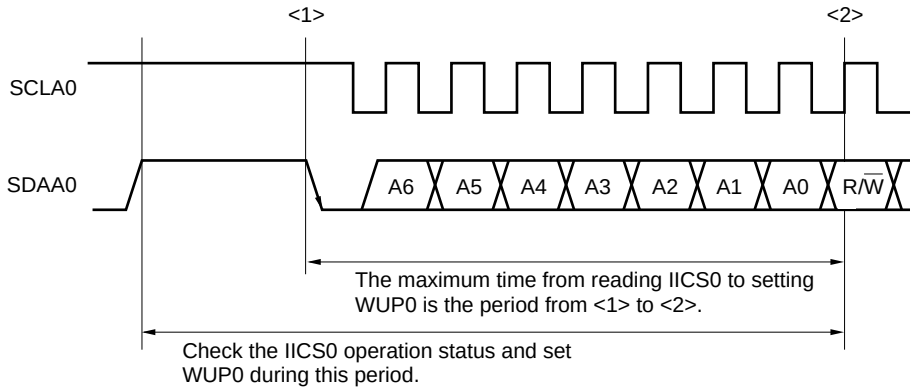


Figure 15-9. Format of IICA Control Register 01 (IICCTL01) (2/2)

CLD0	Detection of SCLA0 pin level (valid only when IICE0 = 1)
0	The SCLA0 pin was detected at low level.
1	The SCLA0 pin was detected at high level.

Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCLA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SCLA0 pin is at high level

DAD0	Detection of SDAA0 pin level (valid only when IICE0 = 1)
0	The SDAA0 pin was detected at low level.
1	The SDAA0 pin was detected at high level.

Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDAA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SDAA0 pin is at high level

SMC0	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).

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in fast mode. In fast mode, the transfer clock does not vary, regardless of the DFC0 bit being set (1) or cleared (0). The digital filter is used for noise elimination in fast mode.	

PRS0	Division of the operation clock
0	Selects f_{CLK} as operation clock.
1	Selects $f_{CLK}/2$ as operation clock.

Remark IICE0: Bit 7 of IICA control register 00 (IICCTL00)

(6) IICA low-level width setting register 0 (IICWL0)

This register is used to set the low-level width of the SCLA0 pin signal that is output by serial interface IICA.

The IICWL0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWL0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

Figure 15-10. Format of IICA Low-Level Width Setting Register 0 (IICWL0)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWL0								

(7) IICA high-level width setting register 0 (IICWH0)

This register is used to set the high-level width of the SCLA0 pin signal that is output by serial interface IICA.

The IICWH0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWH0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

Figure 15-11. Format of IICA High-Level Width Setting Register 0 (IICWH0)

Address: F0233H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWH0								

Remark For how to set the transfer clock by using the IICWL0 and IICWH0 registers, see **15.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers**.

(8) Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCLA0 pin as clock I/O and the P61/SDAA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set the IICE0 bit (bit 7 of IICA control register 00 (IICCTL00)) to 1 before setting the output mode because the P60/SCLA0 and P61/SDAA0 pins output a low level (fixed) when the IICE0 bit is 0.

The PM6 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 15-12. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	PM6.3	PM6.2	PM6.1	PM6.0

PM6.n	P6n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

15.4 I^2C Bus Mode Functions

15.4.1 Pin configuration

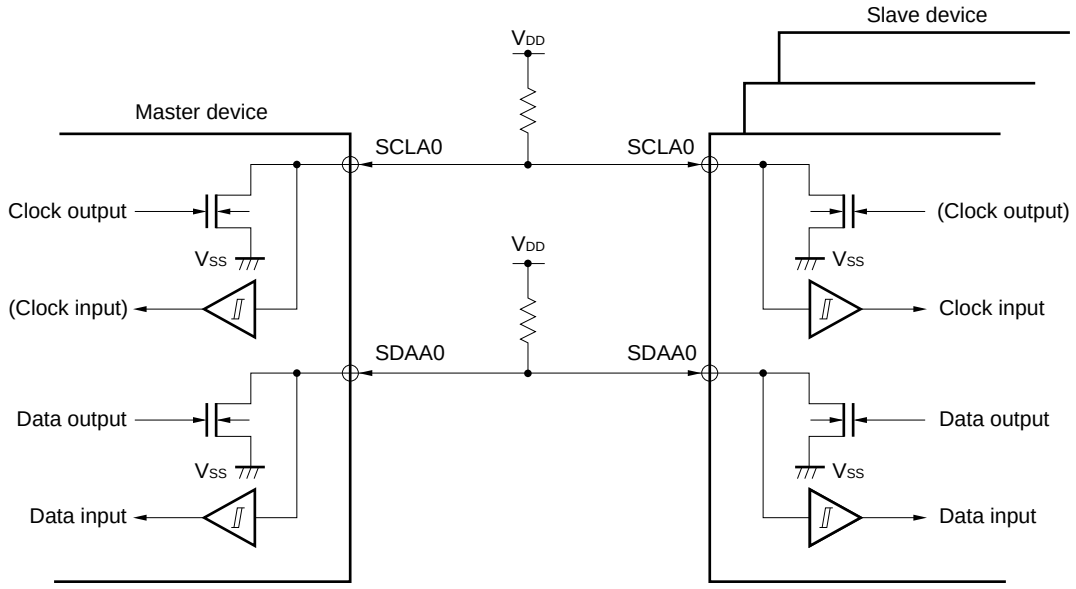
The serial clock pin (SCLA0) and the serial data bus pin (SDAA0) are configured as follows.

(1) SCLA0 .... 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) SDAA0 .... 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 15-13. Pin Configuration Diagram



15.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{CLK}}{IICWL0 + IICWH0 + f_{CLK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWL0 and IICWH0 registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWL0 = \frac{0.52}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH0 = (\frac{0.48}{\text{Transfer clock}} - t_R - t_F) \times f_{CLK}$$

- When the normal mode

$$IICWL0 = \frac{0.47}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH0 = (\frac{0.53}{\text{Transfer clock}} - t_R - t_F) \times f_{CLK}$$

- When the fast mode plus

$$IICWLn = \frac{0.50}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWHn = (\frac{0.50}{\text{Transfer clock}} - t_R - t_F) \times f_{CLK}$$

(2) Setting IICWL0 and IICWH0 registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICWL0 = 1.3\ \mu s \times f_{CLK}$$
$$IICWH0 = (1.2\ \mu s - t_R - t_F) \times f_{CLK}$$

- When the normal mode

$$IICWL0 = 4.7\ \mu s \times f_{CLK}$$
$$IICWH0 = (5.3\ \mu s - t_R - t_F) \times f_{CLK}$$

- When the fast mode plus

$$IICWLn = 0.50\ \mu s \times f_{CLK}$$
$$IICWHn = (0.50\ \mu s - t_R - t_F) \times f_{CLK}$$

Caution Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation 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.)

Note that the maximum operation frequency of the serial interface IICA operation clock is 20 MHz. If f_{CLK} exceeds 20 MHz, select f_{CLK}/2 for the operation clock by setting the PRS0 bit in IICCTL01 to 1.

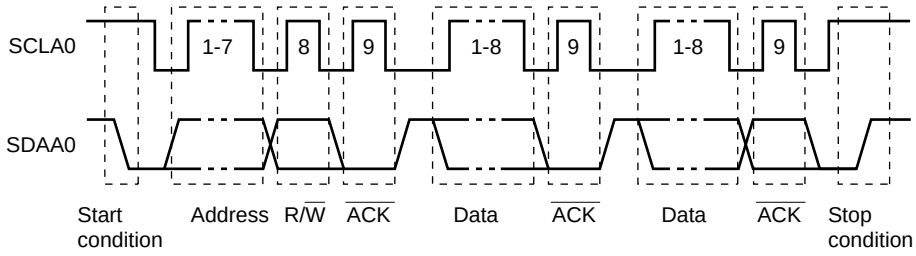
Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAA0 and SCLA0 signals separately, because they differ depending on the pull-up resistance and wire load.

2. IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
t_R: SDAA0 and SCLA0 signal rising times
f_{CLK}: CPU/peripheral hardware clock frequency

15.5 I^2C Bus Definitions and Control Methods

The following section describes the I^2C bus's serial data communication format and the signals used by the I^2C bus. Figure 15-14 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I^2C bus's serial data bus.

Figure 15-14. I^2C Bus Serial Data Transfer Timing



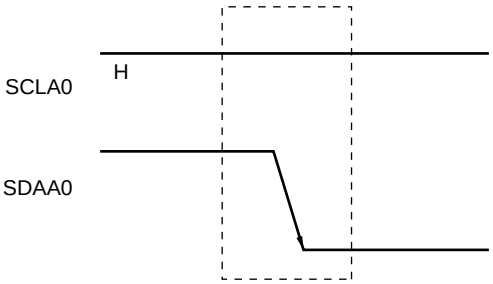
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLA0) is continuously output by the master device. However, in the slave device, the SCLA0 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 SCLA0 pin is at high level and the SDAA0 pin changes from high level to low level. The start conditions for the SCLA0 pin and SDAA0 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 15-15. Start Conditions



A start condition is output when bit 1 (STT0) of IICA control register 00 (IICCTL00) is set (1) after a stop condition has been detected (SPD0: Bit 0 of the IICA status register 0 (IICS0) = 1). When a start condition is detected, bit 1 (STD0) of the IICS0 register is set (1).

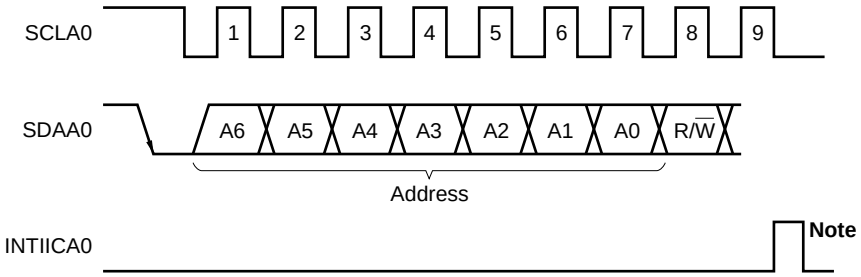
15.5.2 Addresses

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 0 (SVA0). If the address data matches the SVA0 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 15-16. Address



Note INTIICA0 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 0 (IICA0). The received addresses are written to the IICA0 register.

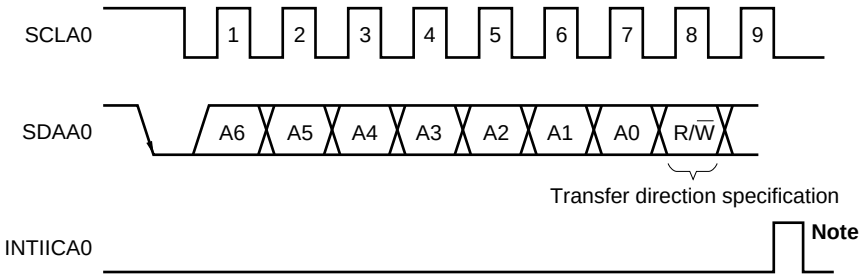
The slave address is assigned to the higher 7 bits of the IICA0 register.

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 15-17. Transfer Direction Specification



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

15.5.4 Acknowledge ($\overline{ACK}$)

$\overline{ACK}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{ACK}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{ACK}$ after transmitting 8-bit data. When $\overline{ACK}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{ACK}$ has been detected can be checked by using bit 2 (ACKD0) of the IICA status register 0 (IICS0).

When the master receives the last data item, it does not return $\overline{ACK}$ and instead generates a stop condition. If a slave does not return $\overline{ACK}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{ACK}$ 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.

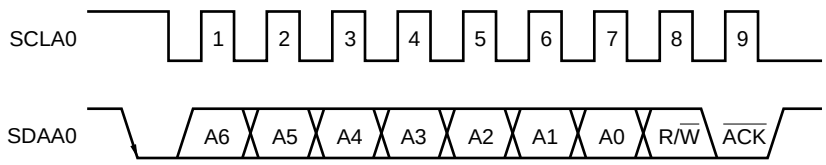
To generate $\overline{ACK}$, the reception side makes the SDAA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{ACK}$ is enabled by setting bit 2 (ACKE0) of IICA control register 00 (IICCTL00) to 1. Bit 3 (TRC0) of the IICS0 register is set by the data of the eighth bit that follows 7-bit address information. Usually, set the ACKE0 bit to 1 for reception (TRC0 = 0).

If a slave can receive no more data during reception (TRC0 = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKE0 bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC0 = 0), it must clear the ACKE0 bit to 0 so that $\overline{ACK}$ 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 15-18. $\overline{ACK}$



When the local address is received, $\overline{ACK}$ is automatically generated, regardless of the value of the ACKE0 bit. When an address other than that of the local address is received, $\overline{ACK}$ is not generated (NACK).

When an extension code is received, $\overline{ACK}$ is generated if the ACKE0 bit is set to 1 in advance.

How $\overline{ACK}$ is generated when data is received differs as follows depending on the setting of the wait timing.

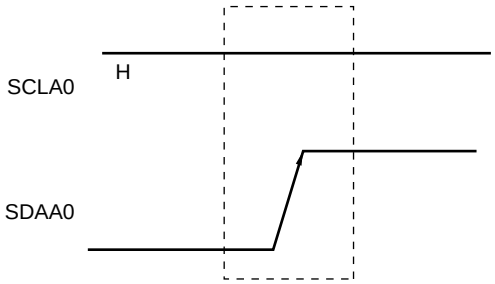
- When 8-clock wait state is selected (bit 3 (WTIM0) of IICCTL00 register = 0):
  By setting the ACKE0 bit to 1 before releasing the wait state, $\overline{ACK}$ is generated at the falling edge of the eighth clock of the SCLA0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM0) of IICCTL00 register = 1):
  $\overline{ACK}$ is generated by setting the ACKE0 bit to 1 in advance.

15.5.5 Stop condition

When the SCLA0 pin is at high level, changing the SDAA0 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 15-19. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IICA control register 00 (IICCTL00) is set to 1. When the stop condition is detected, bit 0 (SPD0) of the IICA status register 0 (IICS0) is set to 1 and INTIICA0 is generated when bit 4 (SPIE0) of the IICCTL00 register is set to 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 SCLA0 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 15-20. 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 ACKE0 = 1)

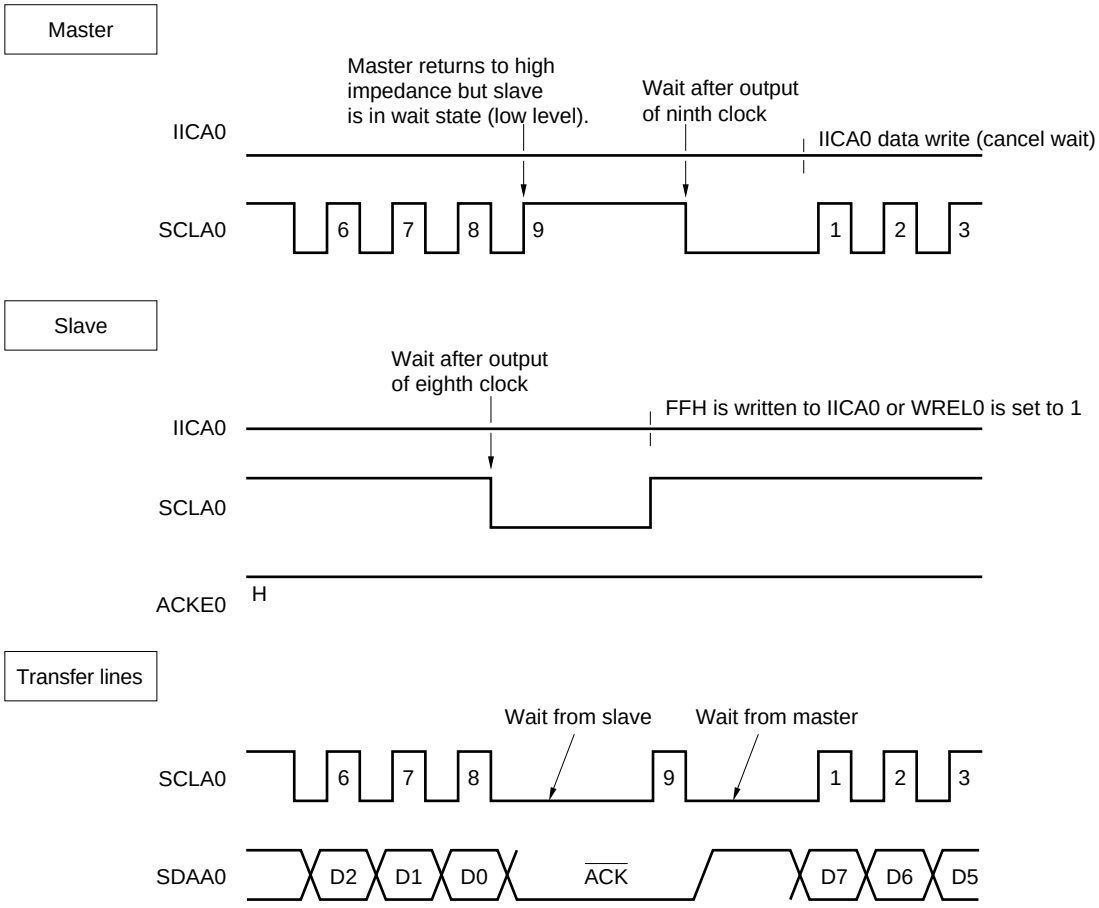
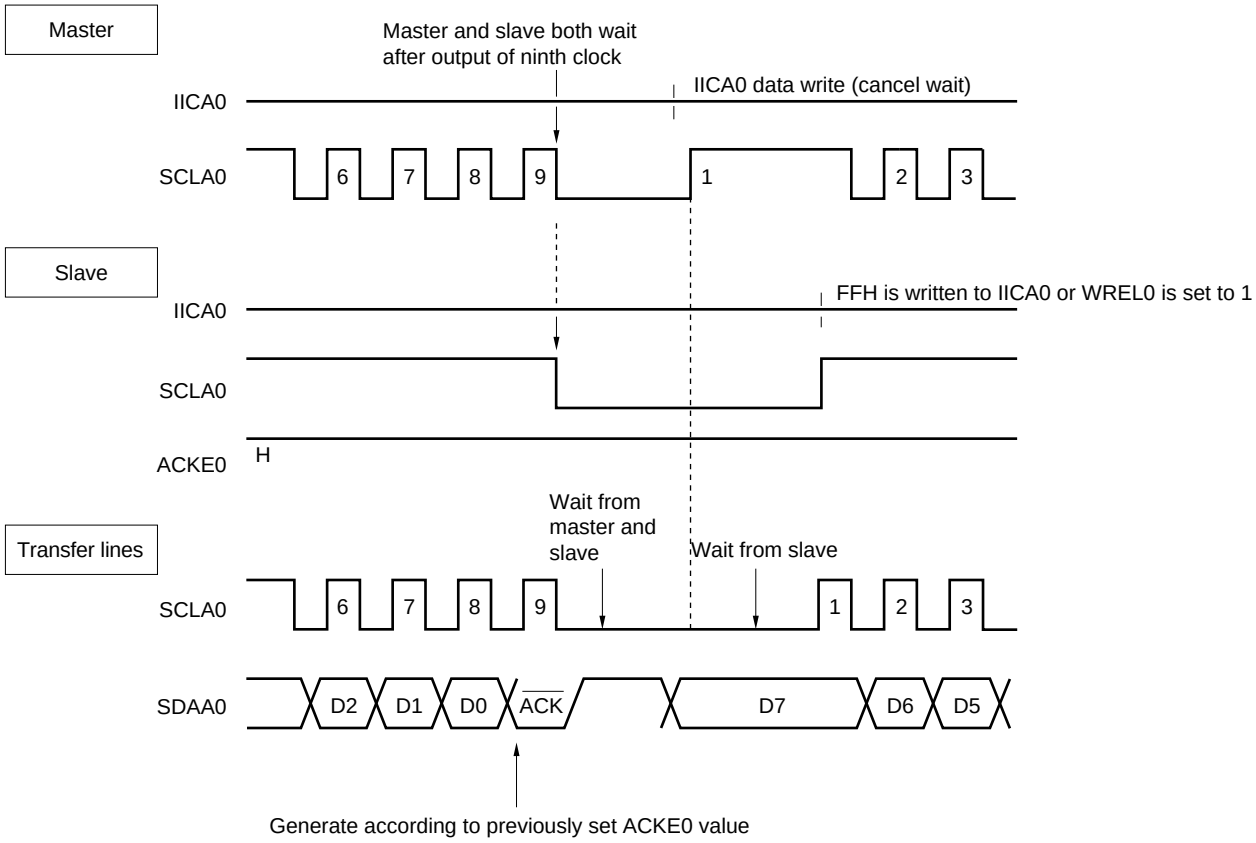


Figure 15-20. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE0 = 1)



Remark ACKE0: Bit 2 of IICA control register 00 (IICCTL00)
WREL0: Bit 5 of IICA control register 00 (IICCTL00)

A wait may be automatically generated depending on the setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00).

Normally, the receiving side cancels the wait state when bit 5 (WREL0) of the IICCTL00 register is set to 1 or when FFH is written to the IICA shift register 0 (IICA0), and the transmitting side cancels the wait state when data is written to the IICA0 register.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT0) of the IICCTL00 register to 1
- By setting bit 0 (SPT0) of the IICCTL00 register to 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 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (canceling wait)
- Setting bit 1 (STT0) of the IICCTL00 register (generating start condition)**Note**
- Setting bit 0 (SPT0) of the IICCTL00 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 IICA0 register.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL0) of the IICCTL00 register to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT0) of the IICCTL00 register to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT0) of the IICCTL00 register to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to the IICA0 register after canceling a wait state by setting the WREL0 bit to 1, an incorrect value may be output to SDAA0 line because the timing for changing the SDAA0 line conflicts with the timing for writing the IICA0 register.

In addition to the above, communication is stopped if the IICE0 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 (LREL0) of the IICCTL00 register, so that the wait state can be canceled.

Caution If a processing to cancel a wait state is executed when WUP0 = 1, the wait state will not be canceled.

15.5.8 Interrupt request (INTIICA0) generation timing and wait control

The setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00) determines the timing by which INTIICA0 is generated and the corresponding wait control, as shown in Table 15-2.

Table 15-2. INTIICA0 Generation Timing and Wait Control

WTIM0	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

Notes 1. The slave device's INTIICA0 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 0 (SVA0).
At this point, $\overline{ACK}$ is generated regardless of the value set to the IICCTL00 register's bit 2 (ACKE0). For a slave device that has received an extension code, INTIICA0 occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICA0 is generated at the falling edge of the 9th clock, but wait does not occur.

2. If the received address does not match the contents of the slave address register 0 (SVA0) and extension code is not received, neither INTIICA0 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 WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (canceling wait)
- Setting bit 1 (STT0) of IICCTL00 register (generating start condition) Note
- Setting bit 0 (SPT0) of IICCTL00 register (generating stop condition) Note

Note Master only.

When an 8-clock wait has been selected (WTIM0 = 0), the presence/absence of $\overline{ACK}$ generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIICA0 is generated when a stop condition is detected (only when SPIE0 = 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 (INTIICA0) occurs when the address set to the slave address register 0 (SVA0) 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 (SDAA0) during data transmission is captured by the IICA shift register 0 (IICA0) 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.

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 (EXC0) is set to 1 for extension code reception and an interrupt request (INTIICA0) is issued at the falling edge of the eighth clock. The local address stored in the slave address register 0 (SVA0) 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 SVA0 register is set to 11110xx0. Note that INTIICA0 occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0 = 1
- Seven bits of data match: COI0 = 1

Remark EXC0: Bit 5 of IICA status register 0 (IICS0)
COI0: Bit 4 of IICA status register 0 (IICS0)

(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 (LREL0) of IICA control register 00 (IICCTL00) 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
0 0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

Remark See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

15.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STT0 bit is set to 1 before the STD0 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 (ALD0) in the IICA status register 0 (IICS0) is set (1) via the timing by which the arbitration loss occurred, and the SCLA0 and SDAA0 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 ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **15.5.8 Interrupt request (INTIICA0) generation timing and wait control**.

Remark STD0: Bit 1 of IICA status register 0 (IICS0)
STT0: Bit 1 of IICA control register 00 (IICCTL00)

Figure 15-21. Arbitration Timing Example

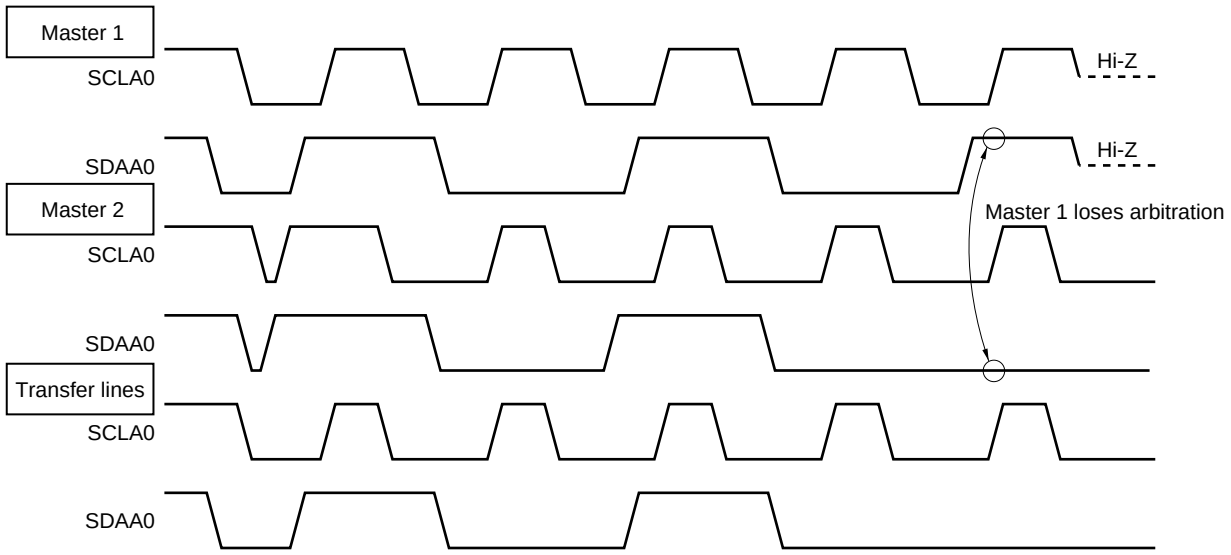


Table 15-4. Status During Arbitration and Interrupt Request Generation Timing

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 $\overline{\text{ACK}}$ 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 SPIE0 = 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 SPIE0 = 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 SCLA0 is at low level while attempting to generate a restart condition	

Notes 1. When the WTIM0 bit (bit 3 of IICA control register 00 (IICCTL00)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

Remark SPIE0: Bit 4 of IICA control register 00 (IICCTL00)

15.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICA0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICA0 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 WUP0 bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA0) 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 WUP0 bit after this interrupt has been generated.

Figure 15-22 shows the flow for setting WUP0 = 1 and Figure 15-23 shows the flow for setting WUP0 = 0 upon an address match.

Figure 15-22. Flow When Setting WUP0 = 1

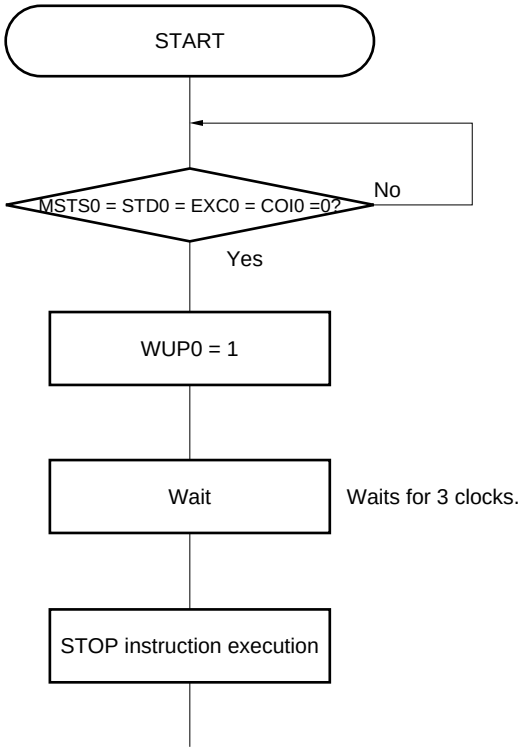
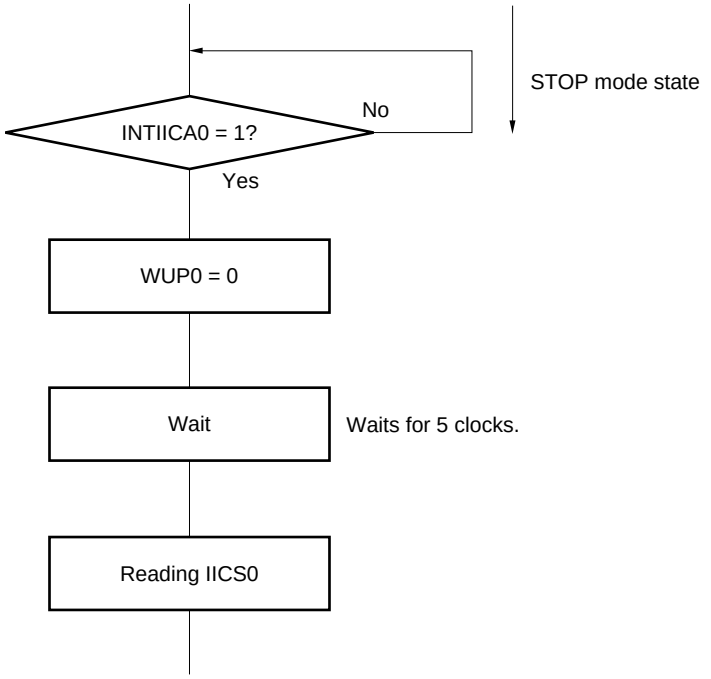


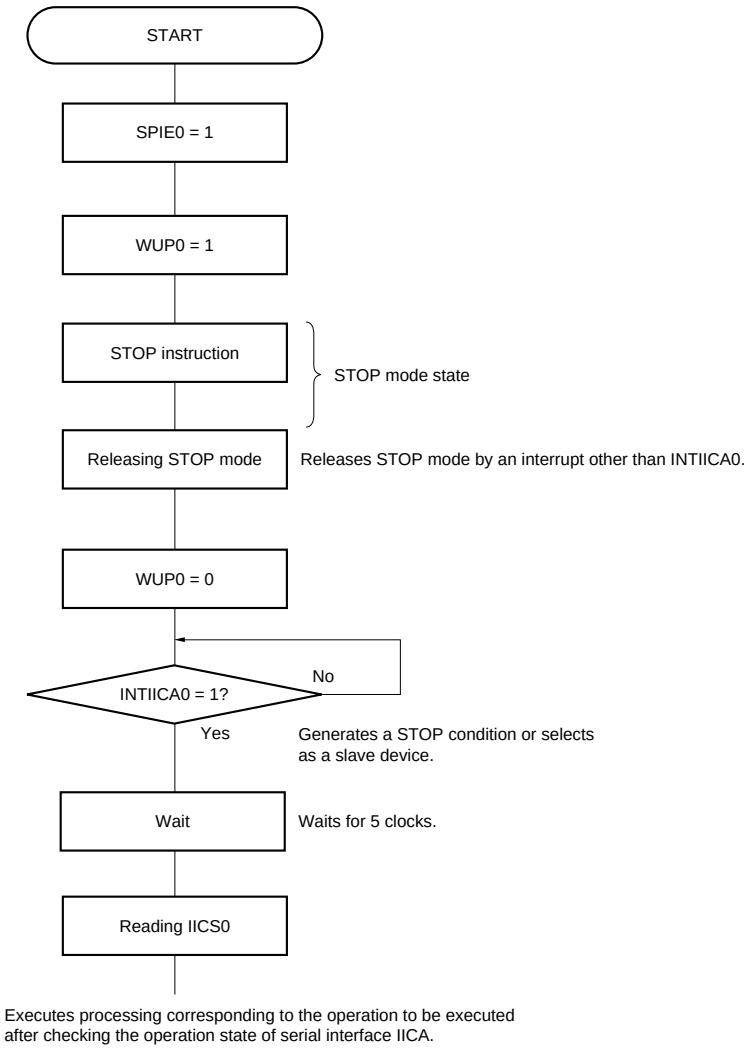
Figure 15-23. Flow When Setting WUP0 = 0 upon Address Match (Including Extension Code Reception)



Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA0) generated from serial interface IICA.

- Master device operation: Flow shown in Figure 15-24
- Slave device operation: Same as the flow in Figure 15-23

Figure 15-24. When Operating as Master Device after Releasing STOP Mode other than by INTIICA0



15.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IICA flag register 0 (IICF0) = 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 ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL0) of IICA control register 00 (IICCTL00) to 1 and saving communication).

If bit 1 (STT0) of the IICCTL00 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 0 (IICA0) after bit 4 (SPIE0) of the IICCTL00 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICA0) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICA0 register before the stop condition is detected is invalid.
When the STT0 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

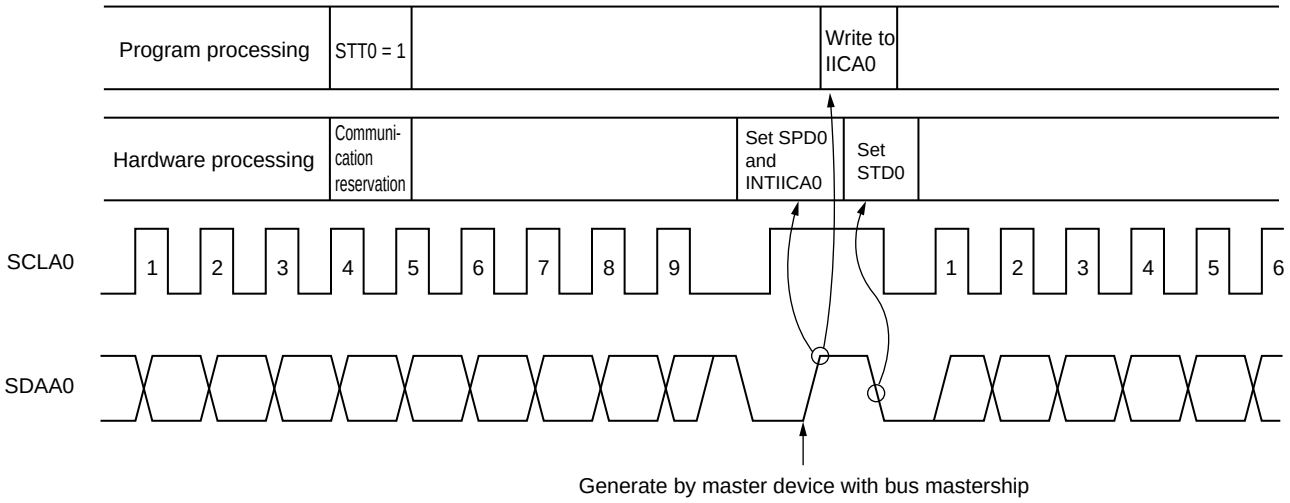
Check whether the communication reservation operates or not by using the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) after the STT0 bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT0 = 1 to checking the MSTS0 flag:
(IICWL0 setting value + IICWH0 setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clocks]

Remark IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 15-25 shows the communication reservation timing.

Figure 15-25. Communication Reservation Timing



Remark IICA0: IICA shift register 0
STT0: Bit 1 of IICA control register 00 (IICCTL00)
STD0: Bit 1 of IICA status register 0 (IICS0)
SPD0: Bit 0 of IICA status register 0 (IICS0)

Communication reservations are accepted via the timing shown in Figure 15-26. After bit 1 (STD0) of the IICA status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IICA control register 00 (IICCTL00) to 1 before a stop condition is detected.

Figure 15-26. Timing for Accepting Communication Reservations

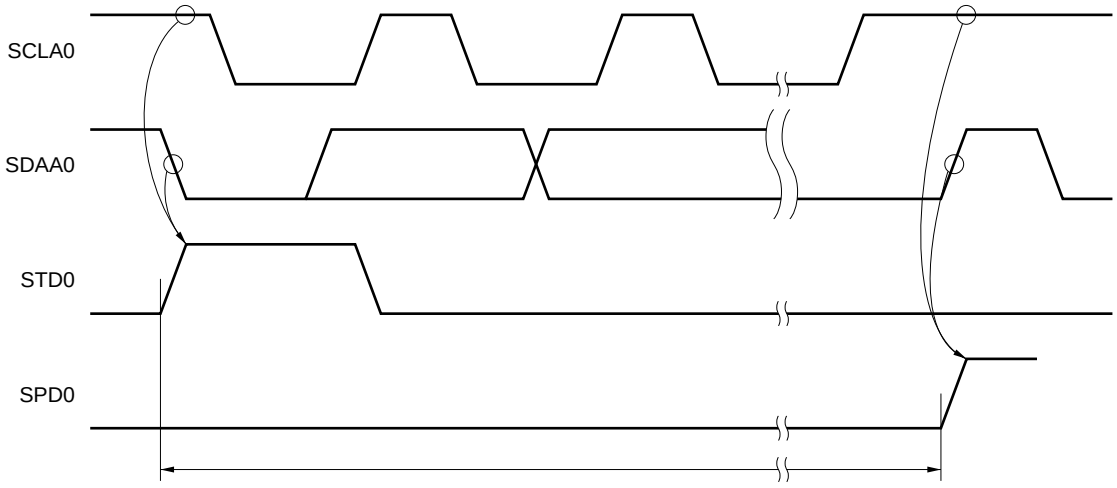
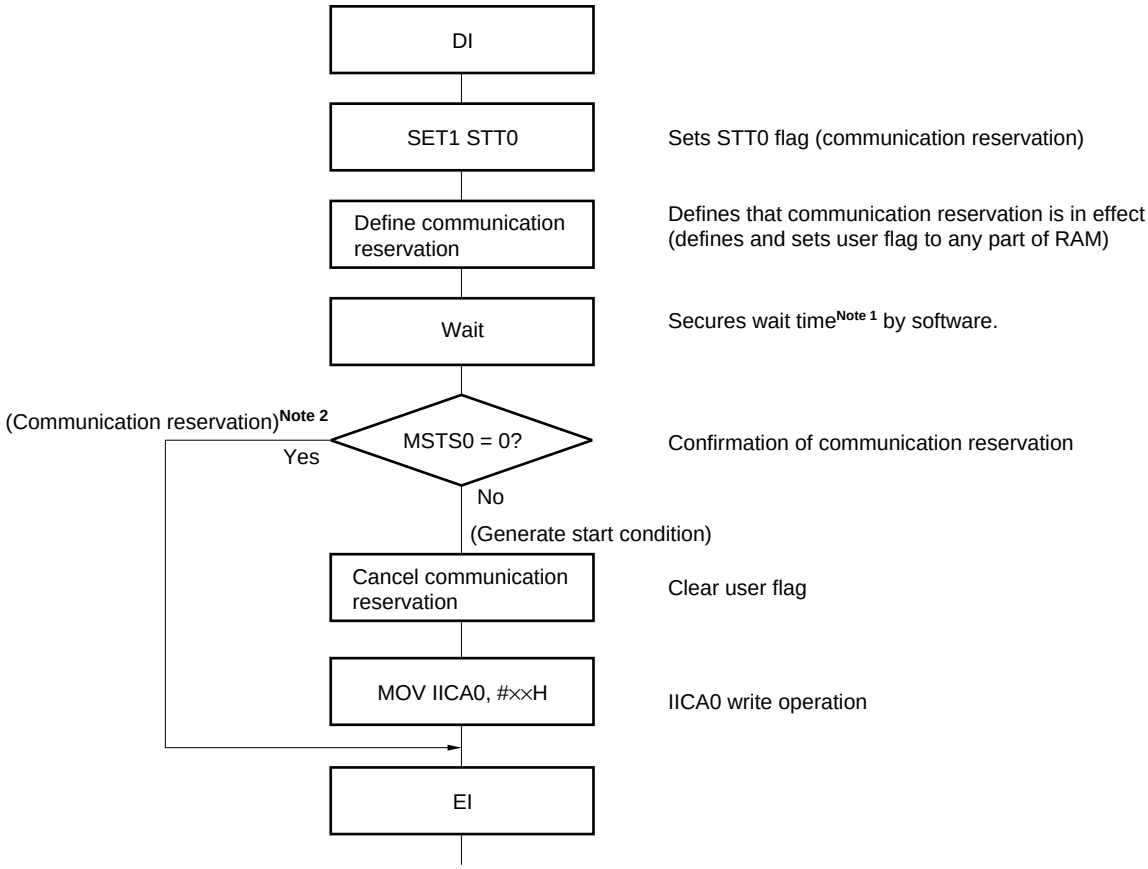


Figure 15-27 shows the communication reservation protocol.

Figure 15-27. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.
(IICWL0 setting value + IICWH0 setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clocks]

2. The communication reservation operation executes a write to the IICA shift register 0 (IICA0) when a stop condition interrupt request occurs.

Remark STT0: Bit 1 of IICA control register 00 (IICCTL00)
MSTS0: Bit 7 of IICA status register 0 (IICS0)
IICA0: IICA shift register 0
IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IICA flag register 0 (IICF0) = 1)

When bit 1 (STT0) of IICA control register 00 (IICCTL00) 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 ($\overline{\text{ACK}}$ is not returned and the bus was released by setting bit 6 (LREL0) of the IICCTL00 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of the IICF0 register). It takes up to 5 clocks until the STCF bit is set to 1 after setting STT0 = 1. Therefore, secure the time by software.

15.5.15 Cautions

(1) When STCEN = 0

Immediately after I^2C operation is enabled (IICE0 = 1), the bus communication status (IICBSY = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IICA control register 01 (IICCTL01).
<2> Set bit 7 (IICE0) of IICA control register 00 (IICCTL00) to 1.
<3> Set bit 0 (SPT0) of the IICCTL00 register to 1.

(2) When STCEN = 1

Immediately after I^2C operation is enabled (IICE0 = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0 = 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 SDAA0 pin is low and the SCLA0 pin is high, the macro of I^2C recognizes that the SDAA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, $\overline{ACK}$ is returned, but this interferes with other I^2C communications. To avoid this, start I^2C in the following sequence.

<1> Clear bit 4 (SPIE0) of the IICCTL00 register to 0 to disable generation of an interrupt request signal (INTIICA0) when the stop condition is detected.
<2> Set bit 7 (IICE0) of the IICCTL00 register to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL0) of the IICCTL00 register to 1 before $\overline{ACK}$ is returned (4 to 80 clocks after setting the IICE0 bit to 1), to forcibly disable detection.

(4) Setting the STT0 and SPT0 bits (bits 1 and 0 of the IICCTL00 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIE0 bit (bit 4 of the IICTL0 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 0 (IICA0) 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 SPIE0 bit to 1 when the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) is detected by software.

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/F12 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/F12 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/F12 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

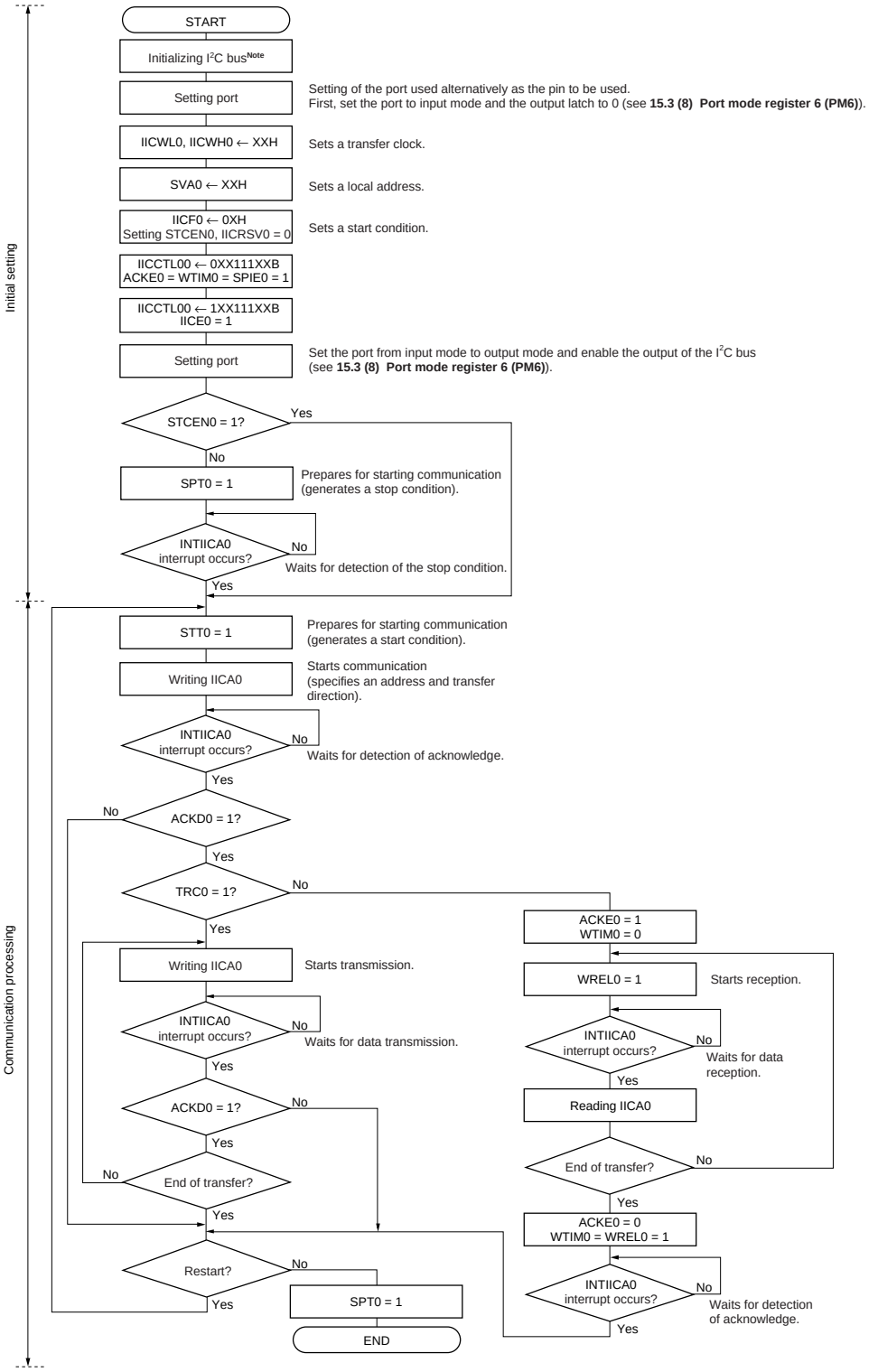
An example of when the RL78/F12 is used as the I^2C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA0 interrupt occurrence (communication waiting). When an INTIICA0 interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

(1) Master operation in single-master system

Figure 15-28. Master Operation in Single-Master System

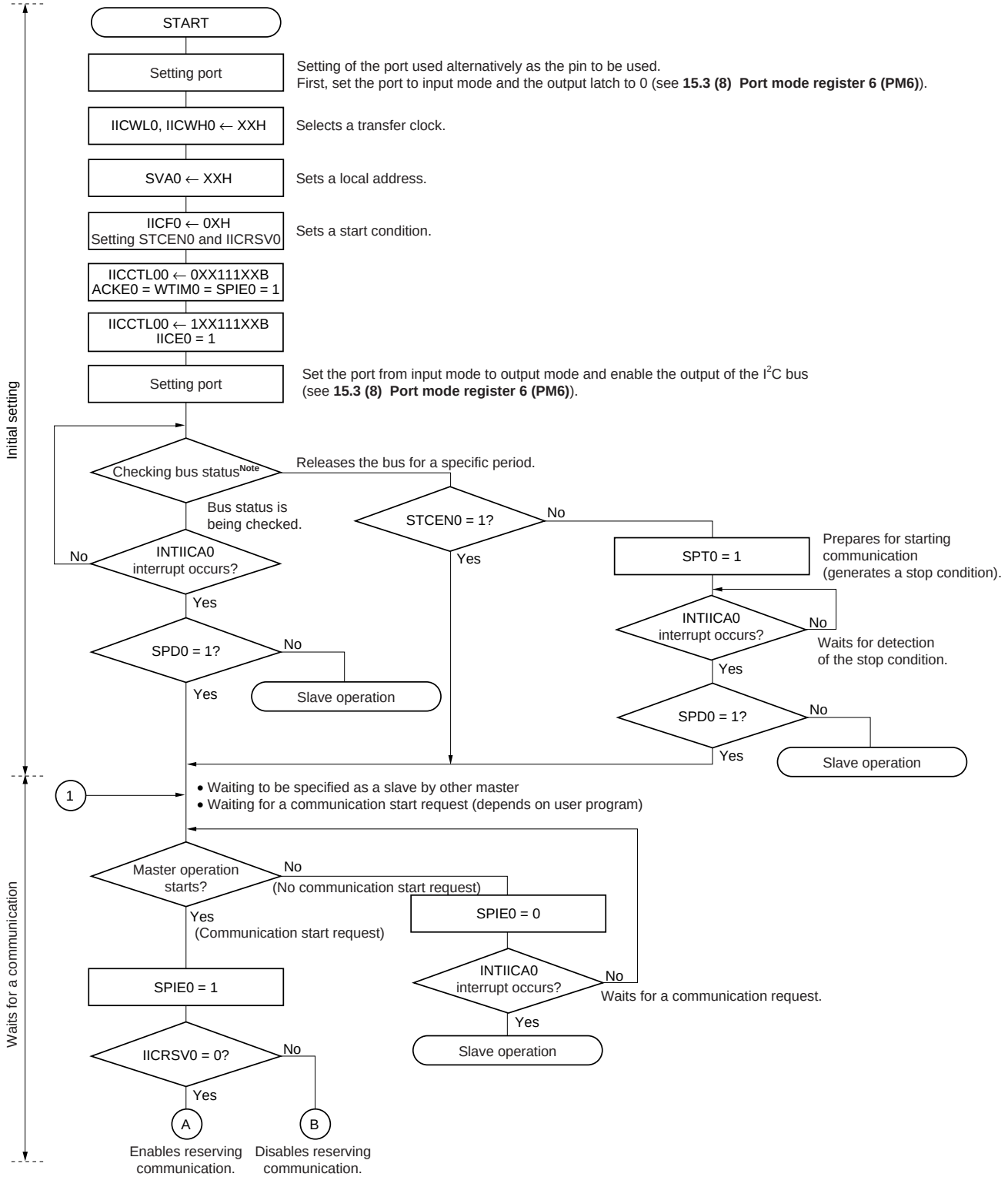


Note Release (SCLA0 and SDAA0 pins = high level) the I^2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAA0 pin, for example, set the SCLA0 pin in the output port mode, and output a clock pulse from the output port until the SDAA0 pin is constantly at high level.

Remark Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

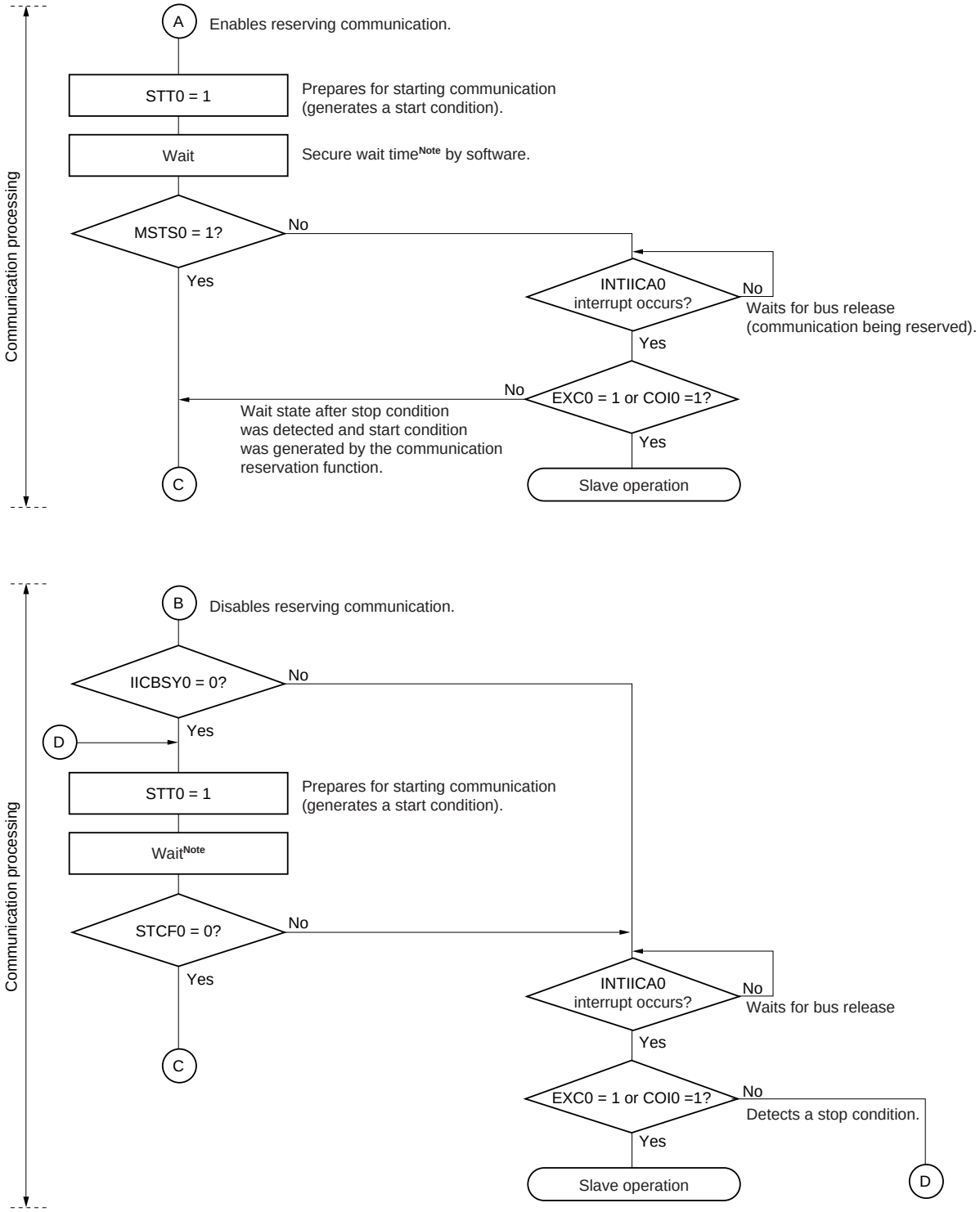
(2) Master operation in multi-master system

Figure 15-29. Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a specific period (for example, for a period of one frame). If the SDAA0 pin is constantly at low level, decide whether to release the I^2C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating.

Figure 15-29. Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.

(IICWL0 setting value + IICWH0 setting value + 4) × f_{CLK} + t_F × 2 [clocks]

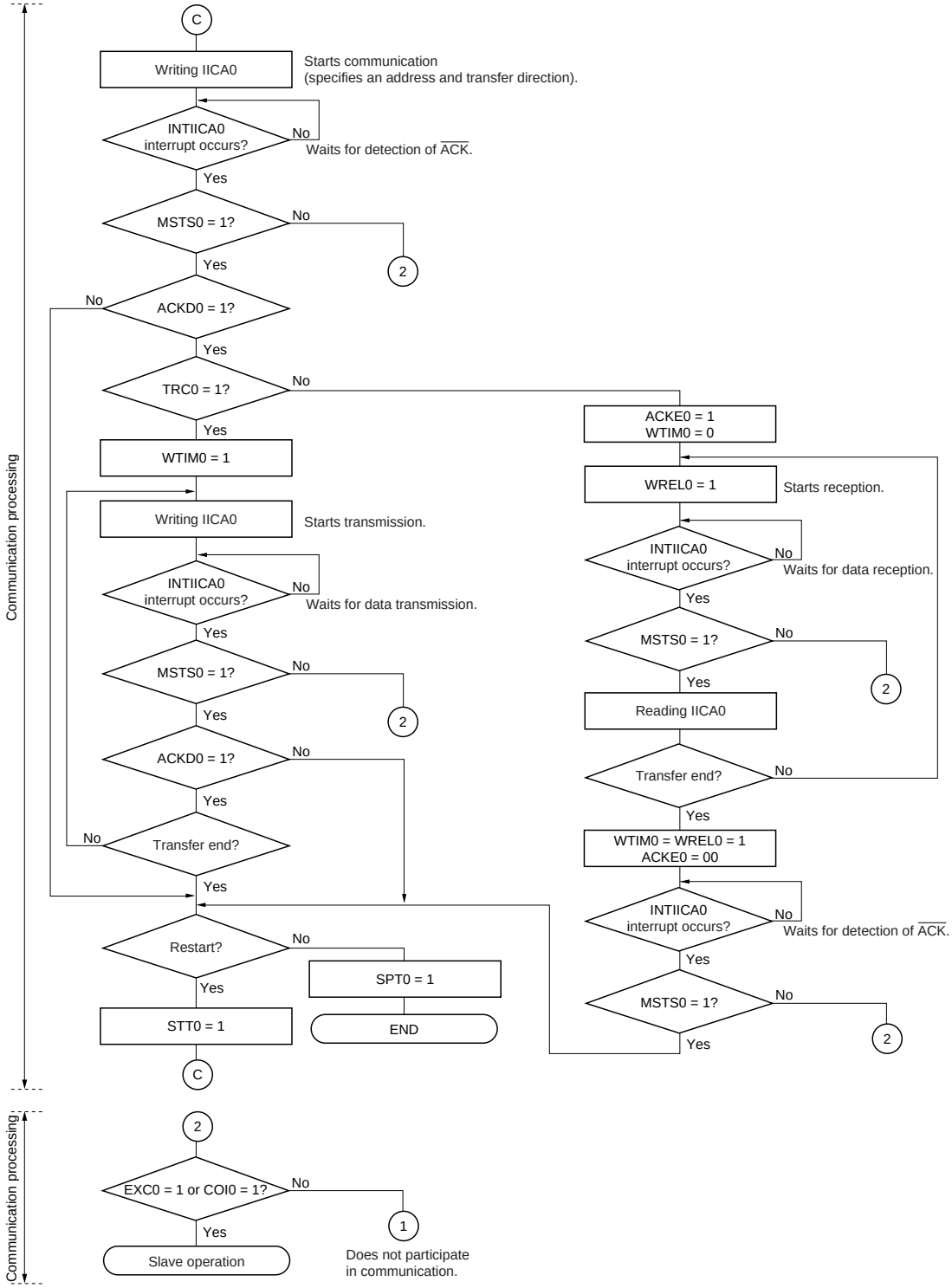
Remark IICWL0: IICA low-level width setting register 0

IICWH0: IICA high-level width setting register 0

t_F: SDAA0 and SCLA0 signal falling times

f_{CLK}: CPU/peripheral hardware clock frequency

Figure 15-29. Master Operation in Multi-Master System (3/3)



Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIICA0 has occurred to check the arbitration result.

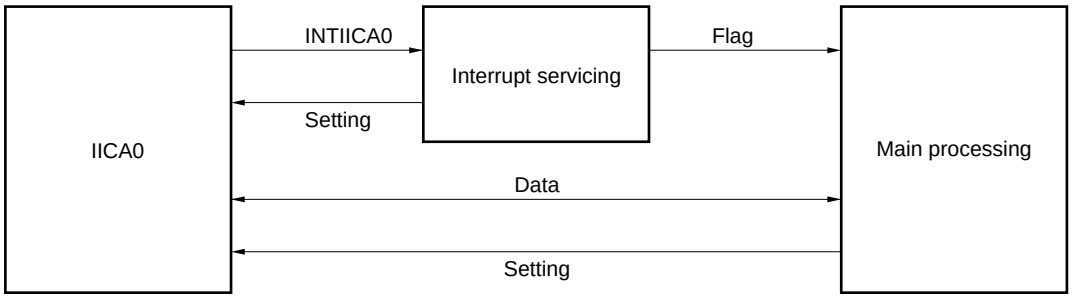
3. To use the device as a slave in a multi-master system, check the status by using the IICA status register 0 (IICS0) and IICA flag register 0 (IICF0) each time interrupt INTIICA0 has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA0 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 INTIICA0 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 INTIICA0.

<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 $\overline{\text{ACK}}$ from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICA0 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 TRC0 bit.

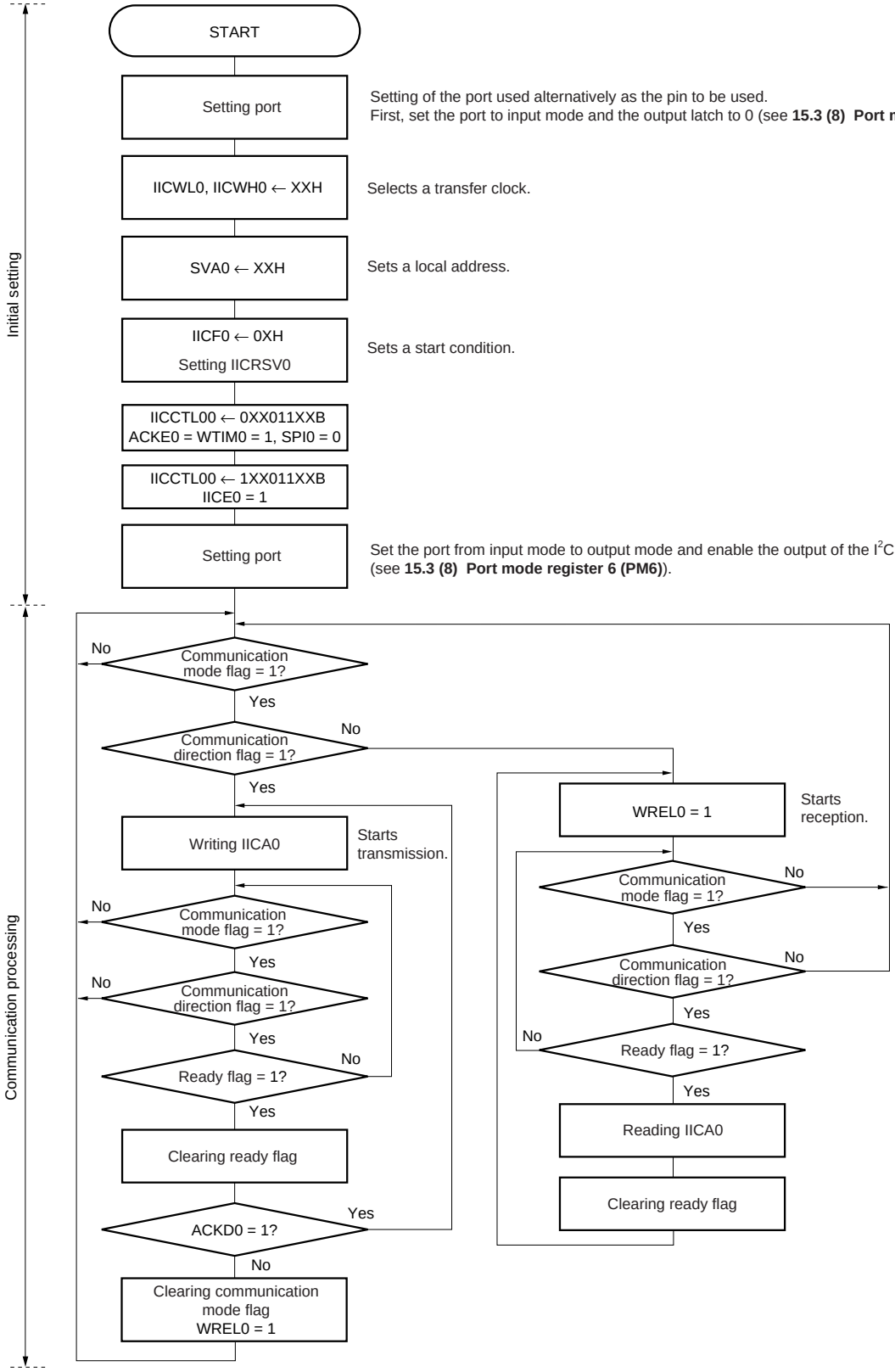
The main processing of the slave operation is explained next.

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 $\overline{\text{ACK}}$. If $\overline{\text{ACK}}$ is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, $\overline{\text{ACK}}$ 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 15-30. Slave Operation Flowchart (1)



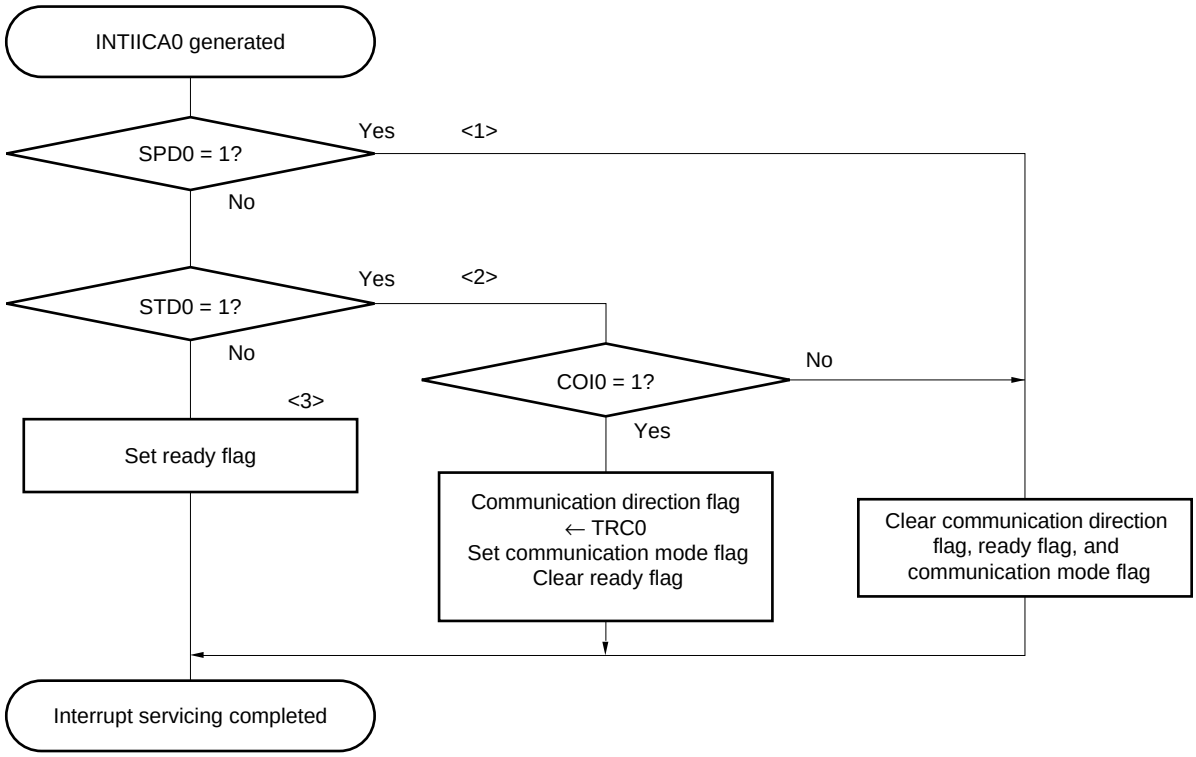
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA0 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 15-31 Slave Operation Flowchart (2).

Figure 15-31. Slave Operation Flowchart (2)



15.5.17 Timing of I^2C interrupt request (INTIICA0) occurrence

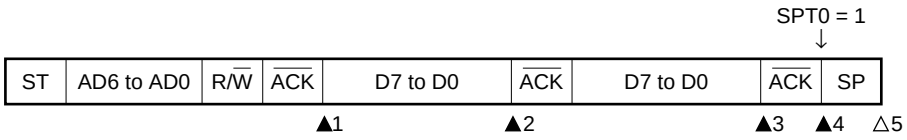
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA0, and the value of the IICA status register 0 (IICS0) when the INTIICA0 signal is generated are shown below.

Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM0 = 0

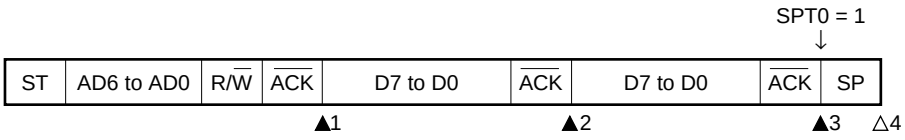


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)[Note]
△5: IICS0 = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

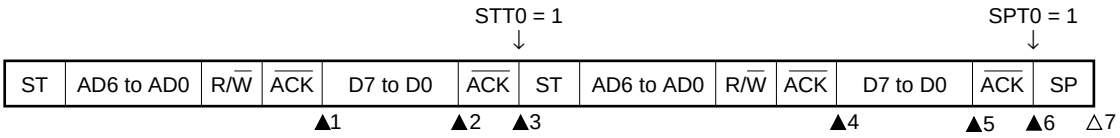


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0

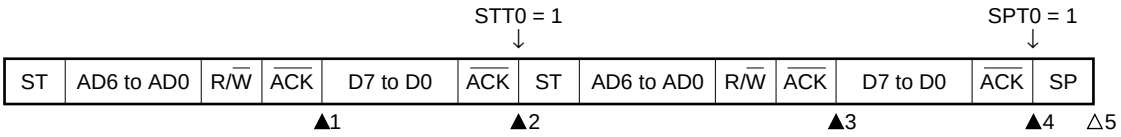


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note 1]
▲3: IICS0 = 1000××00B (Clears the WTIM0 bit to 0[Note 2], sets the STT0 bit to 1)
▲4: IICS0 = 1000×110B
▲5: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note 3]
▲6: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△7: IICS0 = 00000001B

Notes 1. To generate a start condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.
2. Clear the WTIM0 bit to 0 to restore the original setting.
3. To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

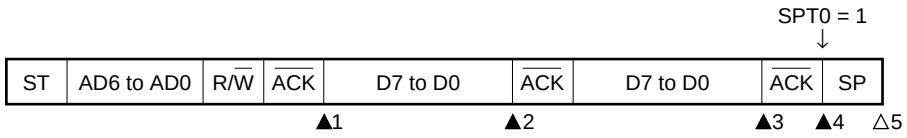


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
▲3: IICS0 = 1000×110B
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0

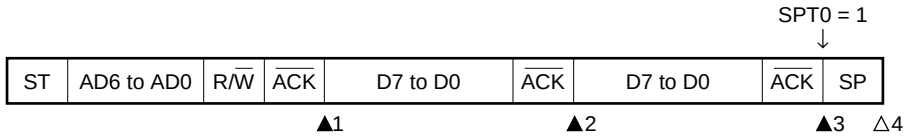


▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×000B
▲3: IICS0 = 1010×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



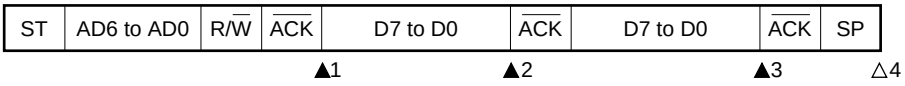
▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×100B
▲3: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

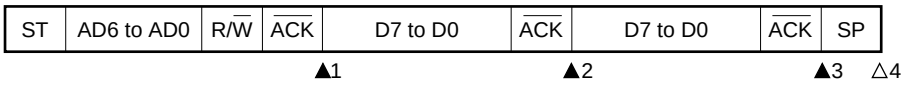
(i) When WTIM0 = 0



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

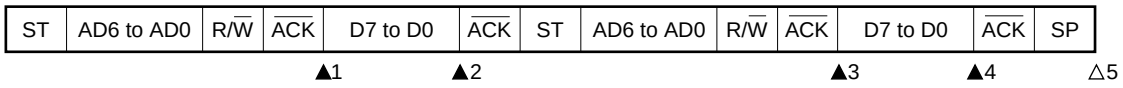


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

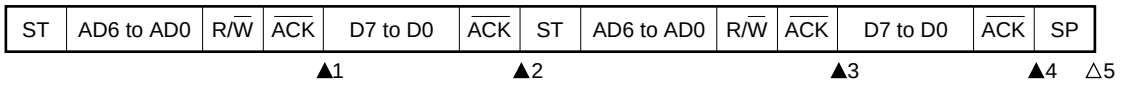
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

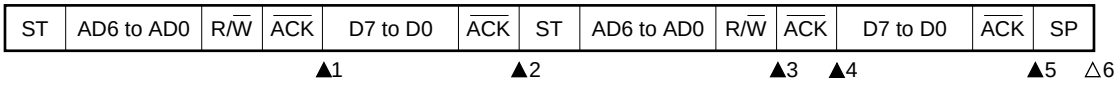
(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= extension code))

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= extension code))



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×110B
▲5: IICS0 = 0010××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

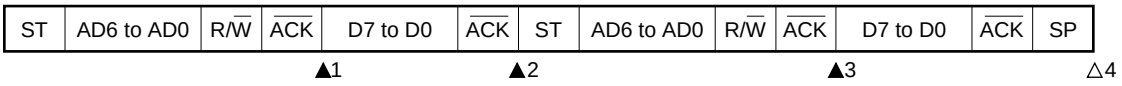
(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

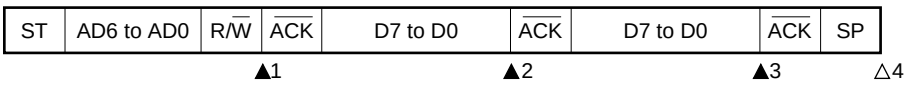
Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(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

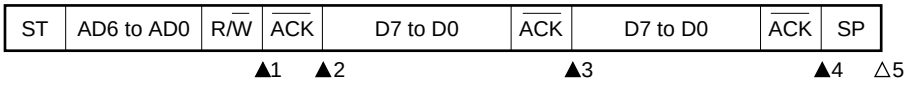
(i) When WTIM0 = 0



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1

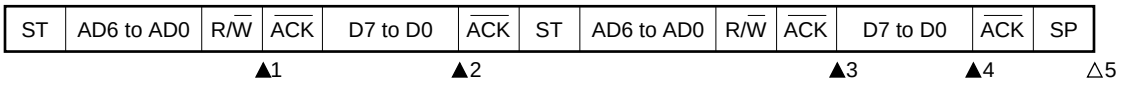


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

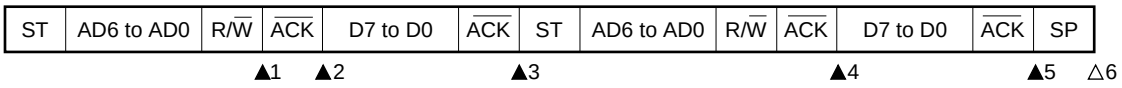
(i) When WTIM0 = 0 (after restart, matches SVA0)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches SVA0)

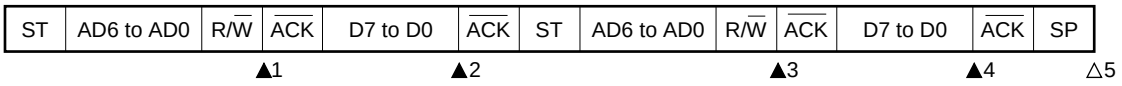


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0001×110B
▲5: IICS0 = 0001××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

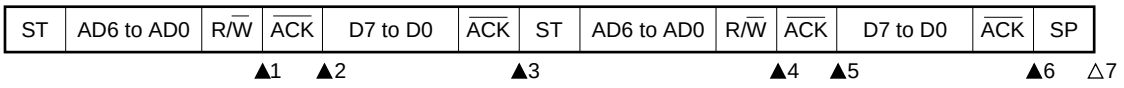
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0010×010B
▲5: IICS0 = 0010×110B
▲6: IICS0 = 0010××00B
△7: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

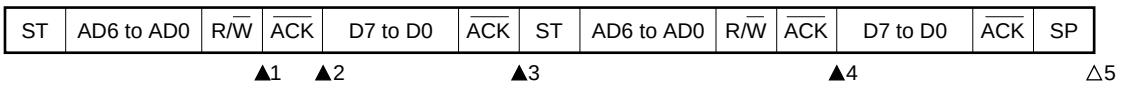
(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))

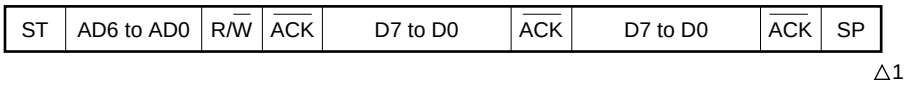


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 00000×10B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICS0 = 00000001B

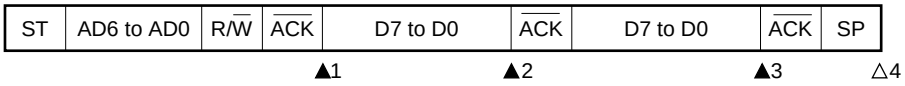
Remark △: Generated only when SPIE0 = 1

(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 MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

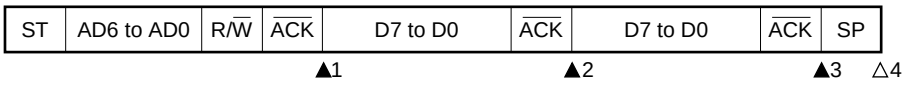
(i) When WTIM0 = 0



▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

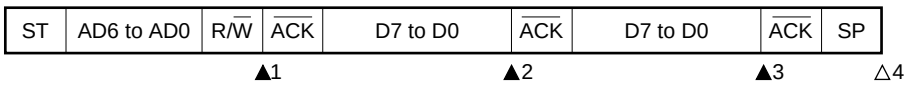


▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

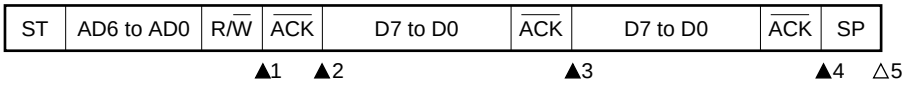
(i) When WTIM0 = 0



▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



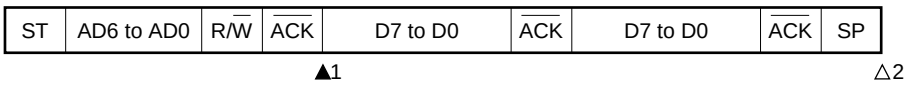
▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 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 MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

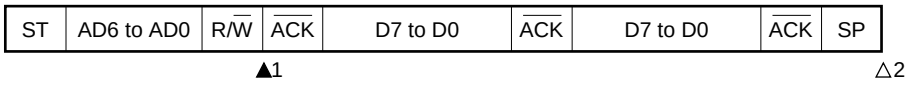
(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)



▲1: IICS0 = 01000110B
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension code

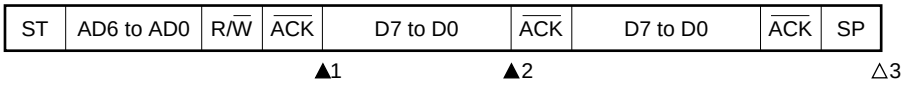


▲1: IICS0 = 0110×010B
Sets LREL0 = 1 by software
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

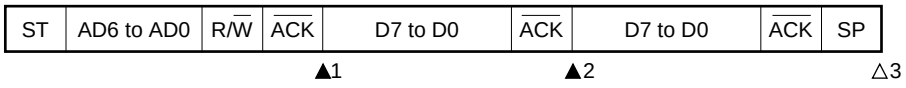
(i) When WTIM0 = 0



▲1: IICS0 = 10001110B
▲2: IICS0 = 01000000B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

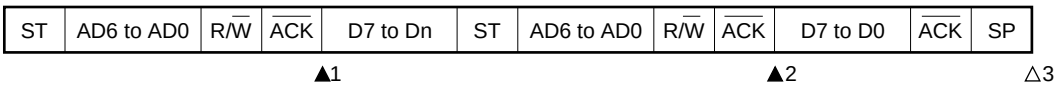


▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

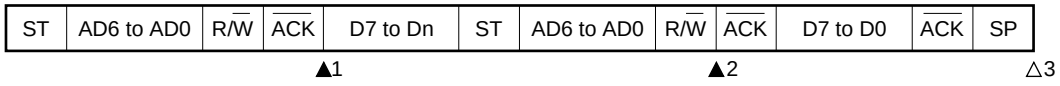
(i) Not extension code (Example: unmatches with SVA0)



▲1: IICS0 = 1000×110B
▲2: IICS0 = 01000110B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(ii) Extension code



▲1: IICS0 = 1000×110B

▲2: IICS0 = 01100010B

Sets LREL0 = 1 by software

△3: IICS0 = 00000001B

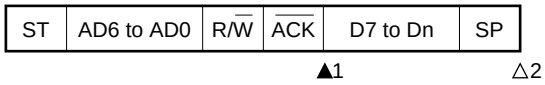
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(e) When loss occurs due to stop condition during data transfer



▲1: IICS0 = 1000×110B

△2: IICS0 = 01000001B

Remark ▲: Always generated

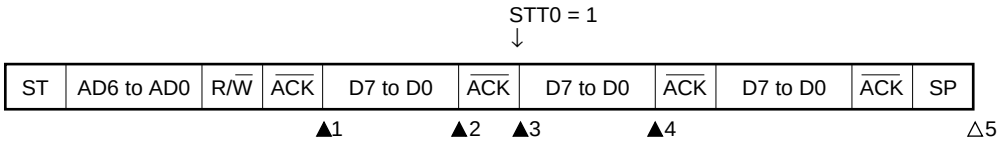
△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

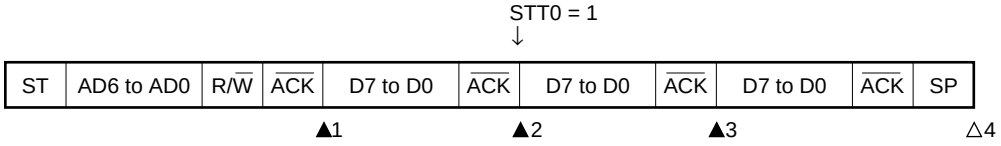
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)
▲4: IICS0 = 01000000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

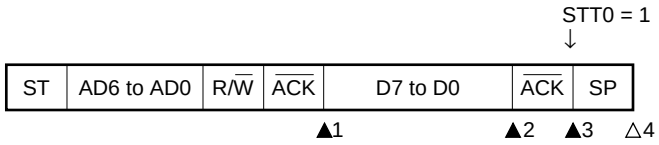


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the STT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

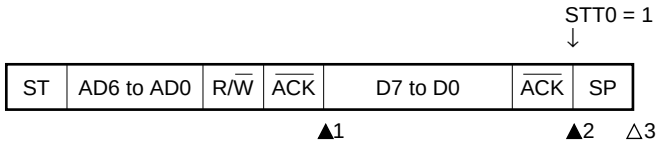
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△4: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

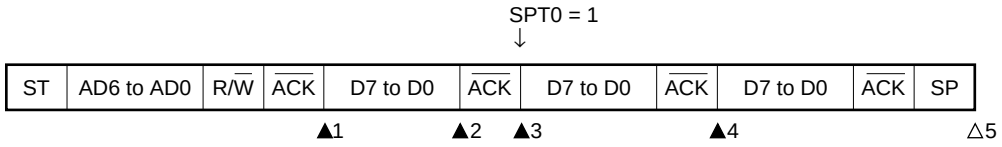


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△3: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

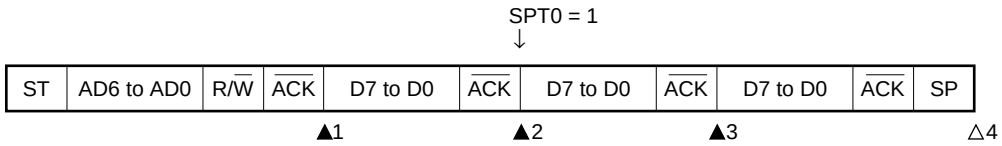
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000××00B (Clears the WTIM0 bit to 0)
▲4: IICS0 = 01000000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the SPT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

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 TRC0 bit (bit 3 of the IICA status register 0 (IICS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

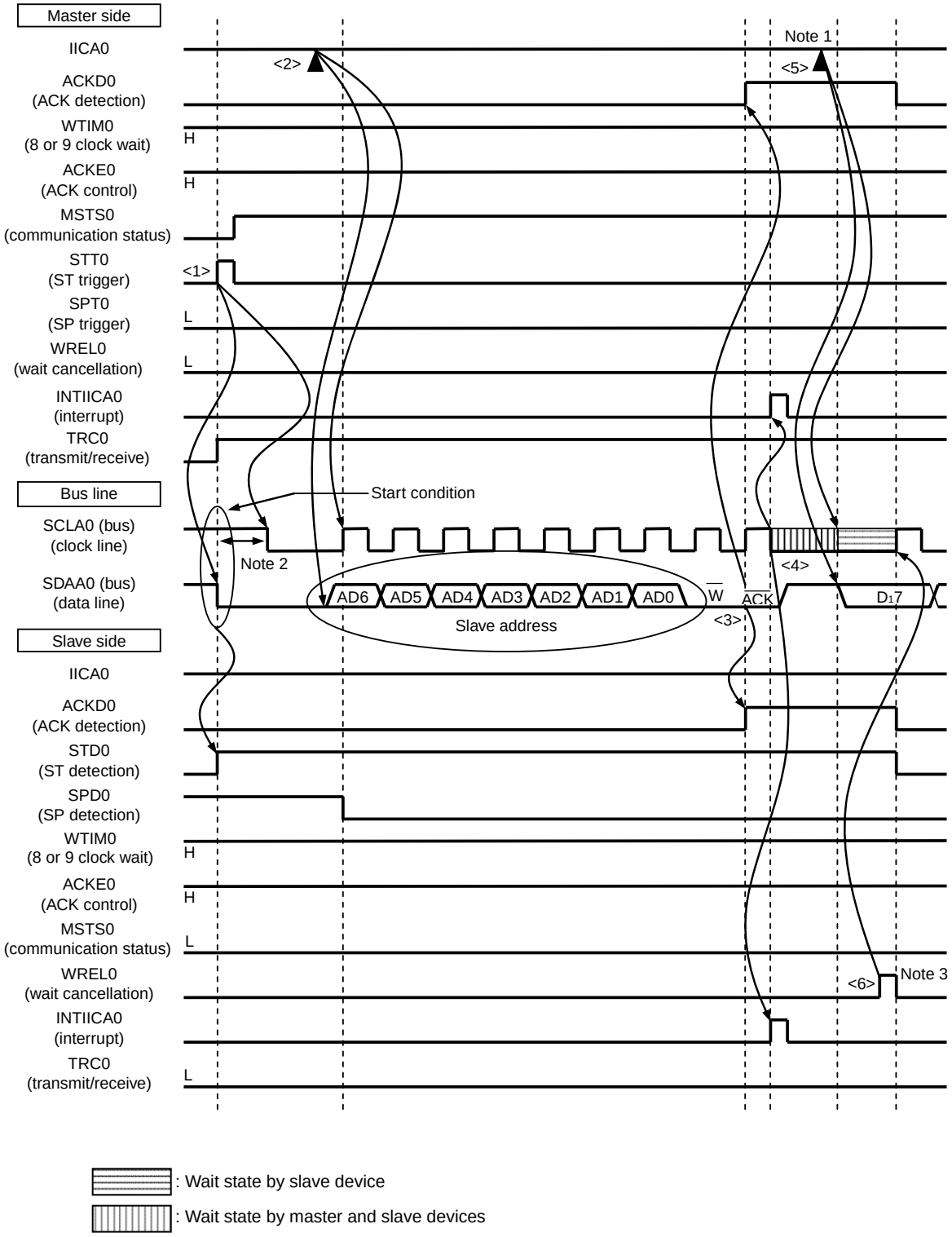
Figures 15-32 and 15-33 show timing charts of the data communication.

The IICA shift register 0 (IICA0)'s shift operation is synchronized with the falling edge of the serial clock (SCLA0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAA0 pin.

Data input via the SDAA0 pin is captured into IICA0 at the rising edge of SCLA0.

Figure 15-32. 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



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission on the master side.

2. Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. To cancel a wait state during reception on the slave side, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 15-32 are explained below.

<1> The start condition trigger is set by the master device (STT0 = 1) and a start condition (SDAA0 = 1 → 0 while SCLA0 = 1) is generated once the bus data line goes low (SDAA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS0 = 1). The master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register 0 (IICA0) and transmits the slave address.

<3> If the address received matches the address 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 (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA0: address match). The master device and slave device also set a wait status (SCLA0 = 0)**Note** when the addresses match.

<5> The master device writes the data to transmit to the IICA0 register and releases the wait status that it set by the master device.

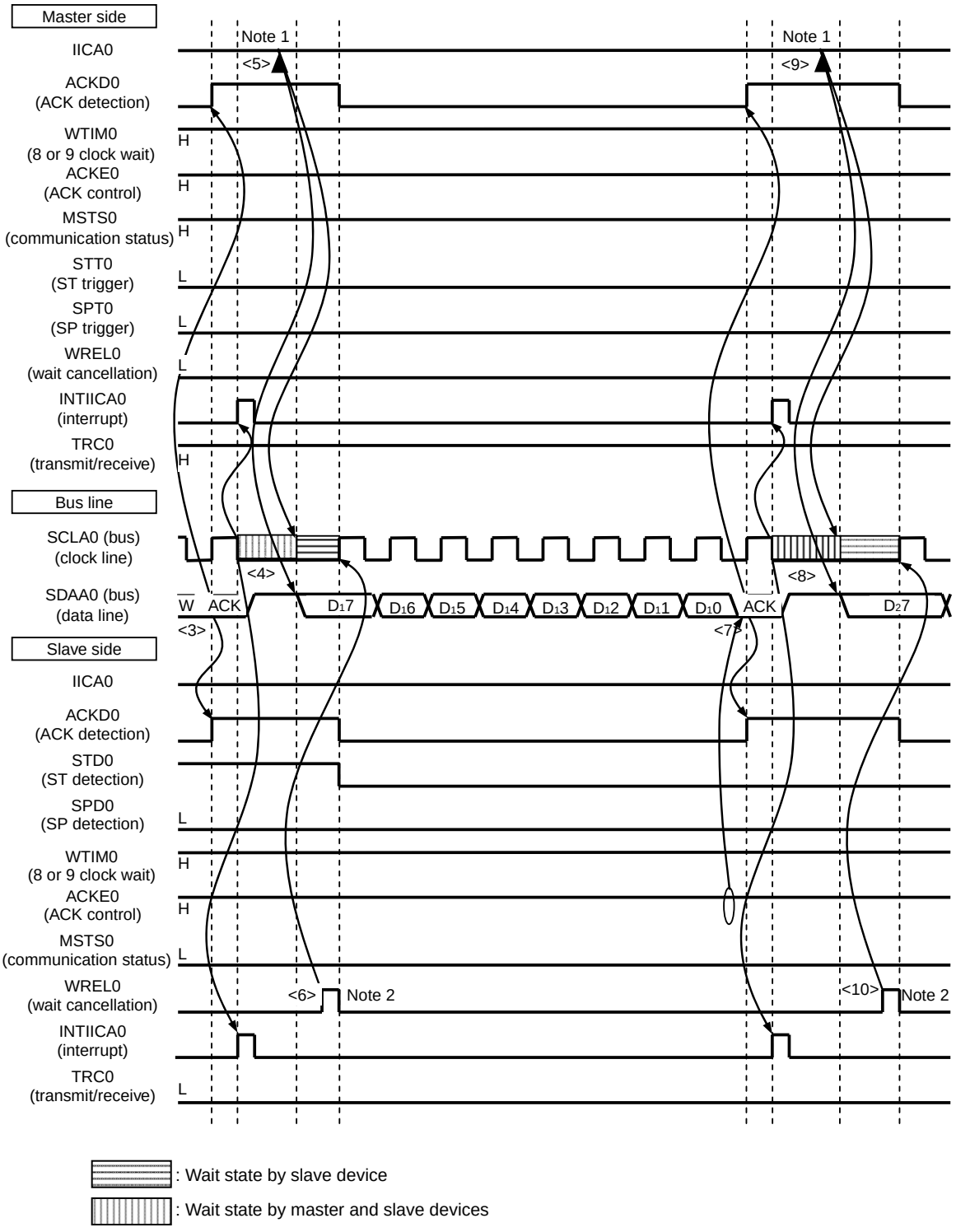
<6> If the slave device releases the wait status (WREL0 = 1), the master device starts transferring data to the slave device.

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: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 15-32 represent the entire procedure for communicating data using the I^2C bus. Figure 15-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 15-32. 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



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission on the master side.

2. To cancel a wait state during reception on the slave side, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 15-32 are explained below.

<3> If the address received matches the address 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 (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA0: address match). The master device and slave device also set a wait status (SCLA0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WREL0 = 1), the master device starts transferring data to the slave device.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<9> The master device writes the data to transmit to the IICA0 register and releases the wait status that it set by the master device.

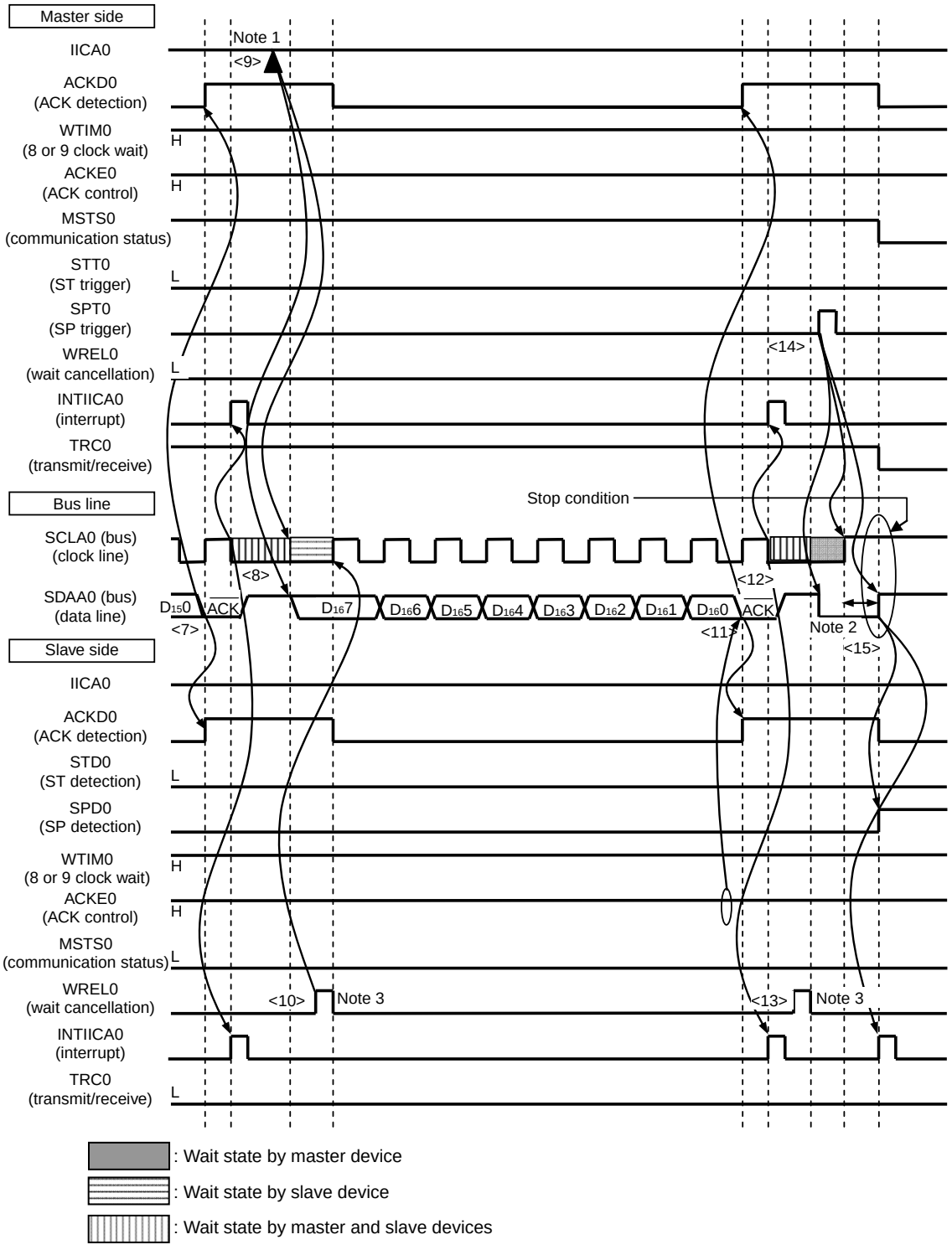
<10> The slave device reads the received data and releases the wait status (WREL0 = 1). The master device then starts transferring data to the slave device.

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: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 15-32 represent the entire procedure for communicating data using the I^2C bus. Figure 15-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 15-32. 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



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission on the master side.

2. Make sure that the time between the rise of the SCLA0 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.

3. To cancel a wait state during reception on the slave side, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 15-32 are explained below.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register 0 (IICA0) 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 (WREL0 = 1). The master device then starts transferring data to the slave device.

<11> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<13> The slave device reads the received data and releases the wait status (WREL0 = 1).

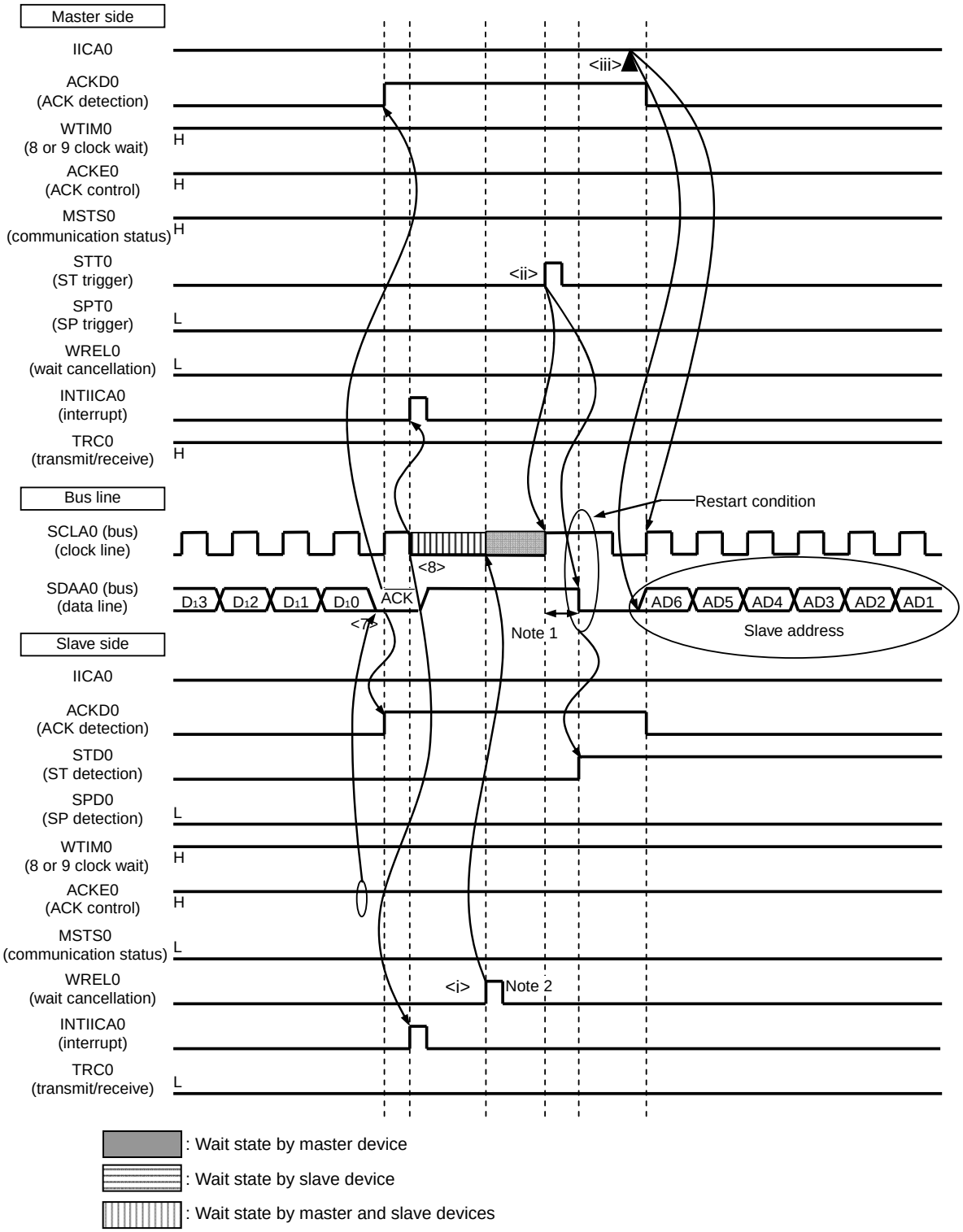
<14> After a stop condition trigger is set, the bus data line is cleared (SDAA0 = 0) and the bus clock line is set (SCLA0 = 1). The stop condition is then generated by setting the bus data line (SDAA0 = 1) after the stop condition setup time has elapsed.

<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICA0: stop condition).

Remark <1> to <15> in Figure 15-32 represent the entire procedure for communicating data using the I^2C bus. Figure 15-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 15-32. 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



Notes 1. Make sure that the time between the rise of the SCLA0 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.

2. To cancel a wait state during reception on the slave side, write “FFH” to IICA0 or set the WREL0 bit.

The following describes the operations in Figure 15-32 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <1> to <3> are performed. These steps return the processing to step <3>, the data transmission step.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

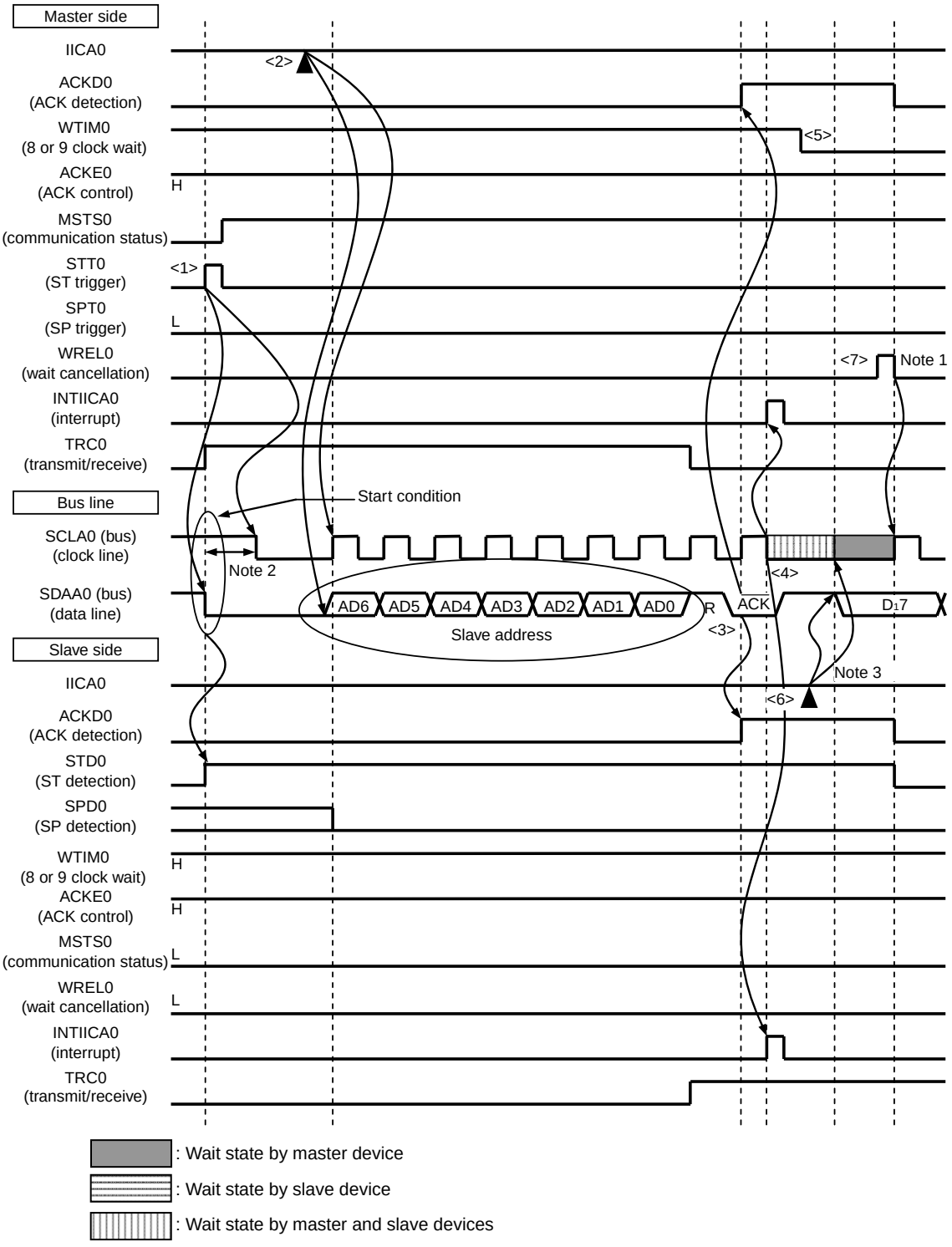
<i> The slave device reads the received data and releases the wait status (WREL0 = 1).

<ii> The start condition trigger is set again by the master device (STT0 = 1) and a start condition (SDAA0 = 1 → 0 while SCLA0 = 1) is generated once the bus clock line goes high (SCLA0 = 1) and the bus data line goes low (SDAA0 = 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 (SCLA0 = 0) after the hold time has elapsed.

<iii> The master device writes the address + R/W (transmission) to the IICA shift register 0 (IICA0) and transmits the slave address.

Figure 15-33. 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



Notes 1. To cancel a wait state during reception on the master side, write “FFH” to IICA0 or set the WREL0 bit.

2. Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission on the slave side.

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 15-33 are explained below.

<1> The start condition trigger is set by the master device (STT0 = 1) and a start condition (SDAA0 = 1 → 0 while SCLA0 = 1) is generated once the bus data line goes low (SDAA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS0 = 1). The master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register 0 (IICA0) and transmits the slave address.

<3> If the address received matches the address 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 (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA0: address match). The master device and slave device also set a wait status (SCLA0 = 0)**Note** when the addresses match.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM0 = 0).

<6> The slave device writes the data to transmit to the IICA0 register and releases the wait status that it set by the slave device.

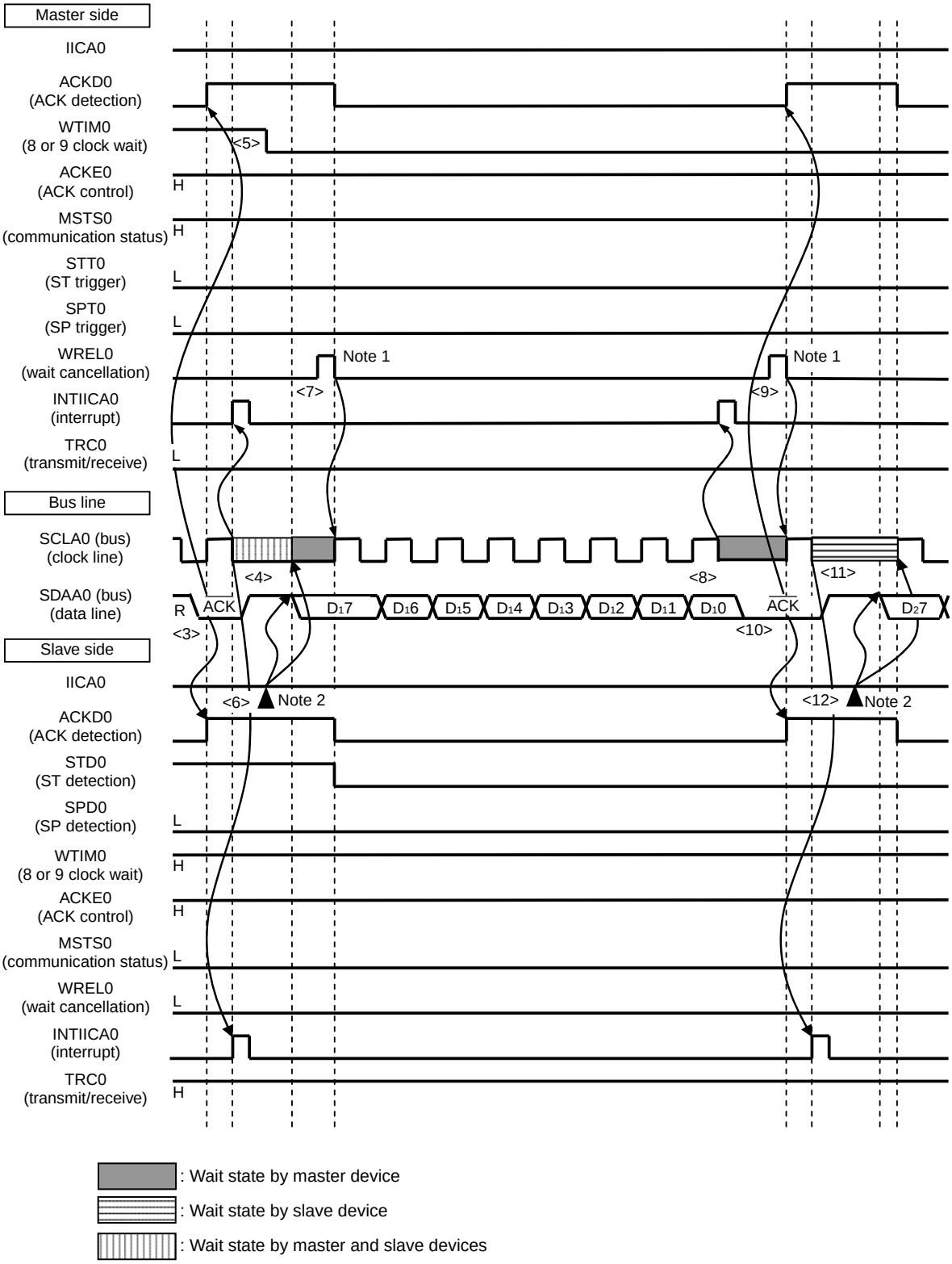
<7> If the master device releases the wait status (WREL0 = 1), the slave device starts transferring data to the master device.

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: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 15-33 represent the entire procedure for communicating data using the I^2C bus. Figure 15-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 15-33. 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



Notes 1. To cancel a wait state during reception on the master side, write “FFH” to IICA0 or set the WREL0 bit.

2. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission on the slave side.

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 15-33 are explained below.

<3> If the address received matches the address 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 (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA0: address match). The master device and slave device also set a wait status (SCLA0 = 0)[Note] when the addresses match.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM0 = 0).

<6> The slave device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the wait status that it set by the slave device.

<7> If the master device releases the wait status (WREL0 = 1), the slave device starts transferring data to the master device.

<8> The master device sets a wait status (SCLA0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA0: end of transfer). 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 (WREL0 = 1).

<10> The ACK is detected by the slave device (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA0: end of transfer).

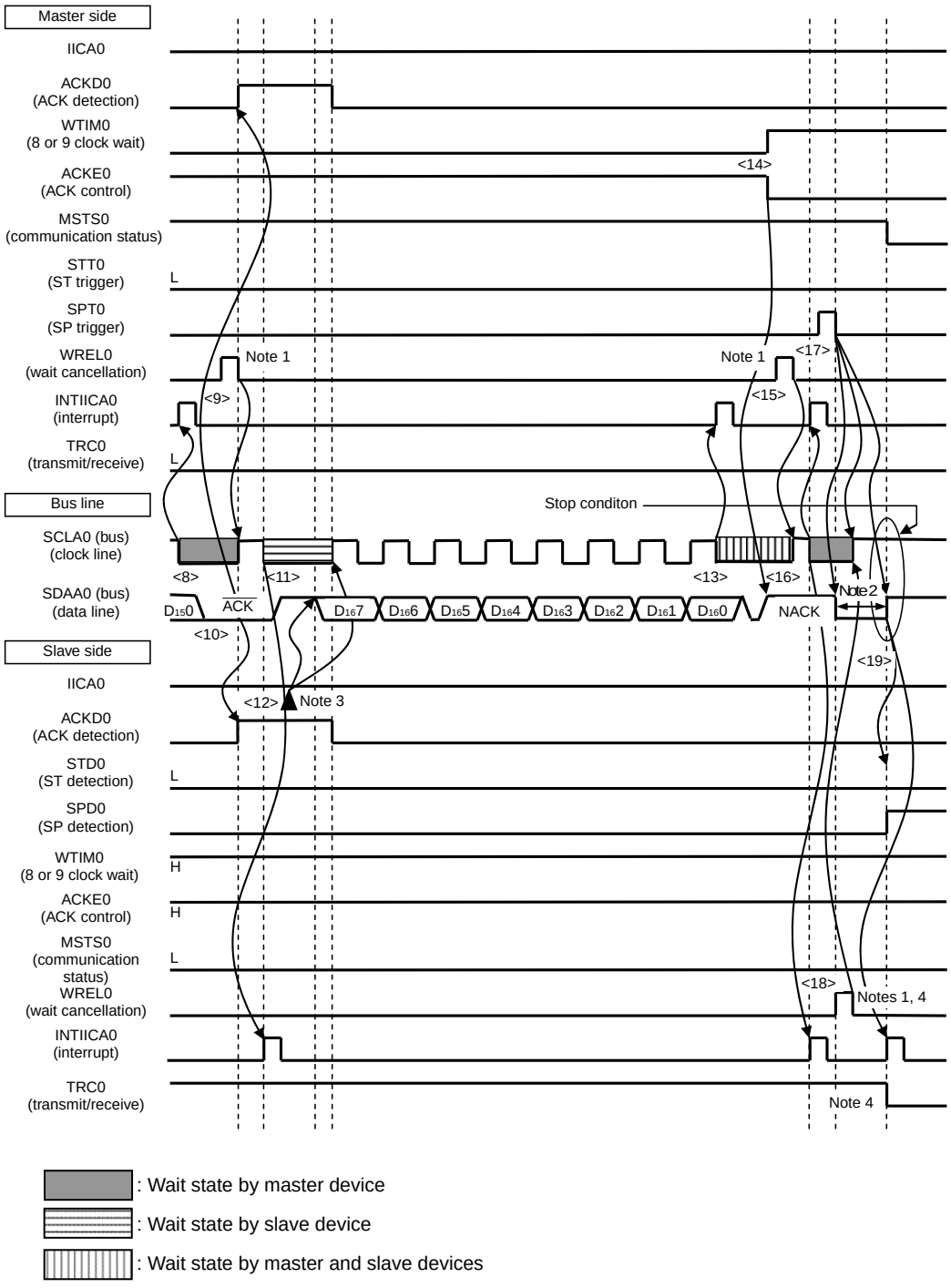
<12> The slave device writes the data to transmit to the IICA0 register and releases the wait status that it set by the slave device. The slave device then starts transferring data to the master device.

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: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 15-33 represent the entire procedure for communicating data using the I^2C bus. Figure 15-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 15-33. 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



: Wait state by master device

: Wait state by slave device

: Wait state by master and slave devices

Notes 1. To cancel a wait state, write “FFH” to IICA0 or set the WREL0 bit.

2. Make sure that the time between the rise of the SCLA0 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.

3. Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission on the slave side.

4. If a wait state during transmission on the slave side is canceled by setting the WREL0 bit, the TRC0 bit will be cleared.

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 15-33 are explained below.

<8> The master device sets a wait status (SCLA0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA0: end of transfer). 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 (WREL0 = 1).

<10> The ACK is detected by the slave device (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA0: end of transfer).

<12> The slave device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the wait status that it set by the slave device. The slave device then starts transferring data to the master device.

<13> The master device issues an interrupt (INTIICA0: end of transfer) at the falling edge of the 8th clock, and sets a wait status (SCLA0 = 0). Because ACK control (ACKE0 = 1) is performed, the bus data line is at the low level (SDAA0 = 0) at this stage.

<14> The master device sets NACK as the response (ACKE0 = 0) and changes the timing at which it sets the wait status to the 9th clock.

<15> If the master device releases the wait status (WREL0 = 1), the slave device detects the NACK (ACK = 0) at the rising edge of the 9th clock.

<16> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<17> When the master device issues a stop condition (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the master device releases the wait status. The master device then waits until the bus clock line is set (SCLA0 = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the wait status (WREL0 = 1) to end communication. Once the slave device releases the wait status, the bus clock line is set (SCLA0 = 1).

<19> Once the master device recognizes that the bus clock line is set (SCLA0 = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAA0 = 1) and issues a stop condition (SDAA0 = 0 → 1 while SCLA0 = 1). The slave device detects the generated stop condition and the slave device issues an interrupt (INTIICA0: stop condition).

Remark <1> to <19> in Figure 15-33 represent the entire procedure for communicating data using the I^2C bus. Figure 15-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

CHAPTER 16 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR

16.1 Functions of Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator has the following functions.

- 16 bits × 16 bits = 32 bits (Unsigned)
- 16 bits × 16 bits = 32 bits (Signed)
- 16 bits × 16 bits + 32 bits = 32 bits (Unsigned)
- 16 bits × 16 bits + 32 bits = 32 bits (Signed)
- 32 bits ÷ 32 bits = 32 bits, 32-bits remainder (Unsigned)

16.2 Configuration of Multiplier and Divider/Multiply-Accumulator

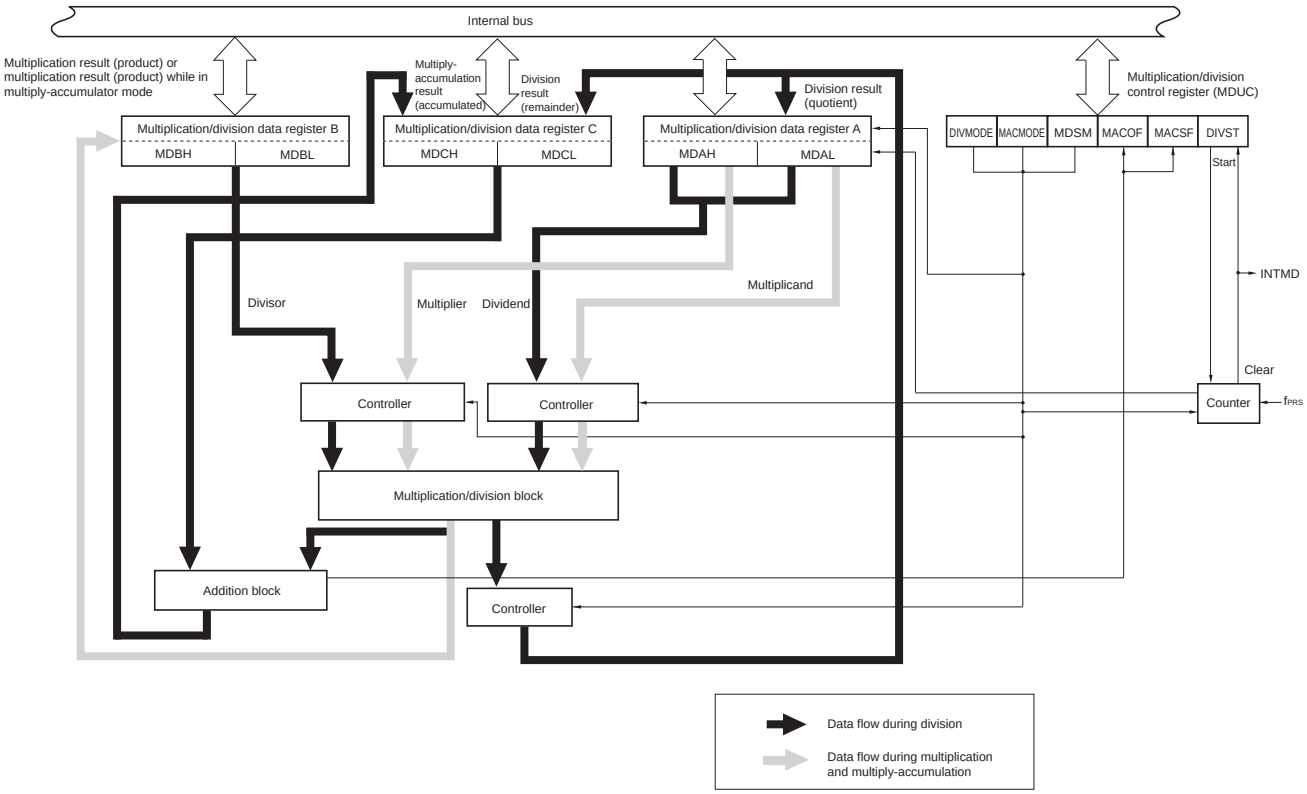
The multiplier and divider/multiply-accumulator consists of the following hardware.

Table 16-1. Configuration of Multiplier and Divider/Multiply-Accumulator

Item	Configuration
Registers	Multiplication/division data register A (L) (MDAL) Multiplication/division data register A (H) (MDAH) Multiplication/division data register B (L) (MDBL) Multiplication/division data register B (H) (MDBH) Multiplication/division data register C (L) (MDCL) Multiplication/division data register C (H) (MDCH)
Control register	Multiplication/division control register (MDUC)

Figure 16-1 shows a block diagram of the multiplier and divider/multiply-accumulator.

Figure 16-1. Block Diagram of Multiplier and Divider/Multiply-Accumulator



(1) Multiplication/division data register A (MDAH, MDAL)

The MDAH and MDAL registers set the values that are used for a multiplication or division operation and store the operation result. They set the multiplier and multiplicand data in the multiplication mode or multiply-accumulator mode, and set the dividend data in the division mode. Furthermore, the operation result (quotient) is stored in the MDAH and MDAL registers in the division mode.

The MDAH and MDAL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 16-2. Format of Multiplication/Division Data Register A (MDAH, MDAL)

Address: FFFF0H, FFFF1H, FFFF2H, FFFF3H After reset: 0000H, 0000H R/W

Symbol	FFFF3H								FFFF2H							
MDAH	MDAH .15	MDAH .14	MDAH .13	MDAH .12	MDAH .11	MDAH .10	MDAH .9	MDAH .8	MDAH .7	MDAH .6	MDAH .5	MDAH .4	MDAH .3	MDAH .2	MDAH .1	MDAH .0

Symbol	FFFF1H								FFFF0H							
MDAL	MDAL .15	MDAL .14	MDAL .13	MDAL .12	MDAL .11	MDAL .10	MDAL .9	MDAL .8	MDAL .7	MDAL .6	MDAL .5	MDAL .4	MDAL 3	MDAL .2	MDAL .1	MDAL .0

Cautions 1. Do not rewrite the MDAH and MDAL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H). The operation will be executed in this case, but the operation result will be an undefined value.

2. The MDAH and MDAL registers values read during division operation processing (when the MDUC register value is 81H or C1H) will not be guaranteed.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDAH and MDAL registers during operation execution.

Table 16-2. Functions of MDAH and MDAL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	MDAH: Multiplier (unsigned) MDAL: Multiplicand (unsigned)	–
Multiplication mode (signed) Multiply-accumulator mode (signed)	MDAH: Multiplier (signed) MDAL: Multiplicand (signed)	–
Division mode	MDAH: Dividend (unsigned) (higher 16 bits) MDAL: Dividend (unsigned) (lower 16 bits)	MDAH: Division result (unsigned) Higher 16 bits MDAL: Division result (unsigned) Lower 16 bits

(2) Multiplication/division data register B (MDBL, MDBH)

The MDBH and MDBL registers set the values that are used for multiplication or division operation and store the operation result. They store the operation result (product) in the multiplication mode and multiply-accumulator mode, and set the divisor data in the division mode.

The MDBH and MDBL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 16-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol FFFF5H FFFF4H

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

Symbol FFFF7H FFFF6H

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

Cautions 1. Do not rewrite the MDBH and MDBL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) or multiply-accumulation operation processing (when the MDUC register value is 41H or 49H). The operation result will be an undefined value.

2. Do not set the MDBH and MDBL registers to 0000H in the division mode. If they are set, the operation result will be an undefined value.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDBH and MDBL registers during operation execution.

Table 16-3. Functions of MDBH and MDBL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	–	MDBH: Multiplication result (product) (unsigned) Higher 16 bits MDBL: Multiplication result (product) (unsigned) Lower 16 bits
Multiplication mode (signed) Multiply-accumulator mode (signed)	–	MDBH: Multiplication result (product) (signed) Higher 16 bits MDBL: Multiplication result (product) (signed) Lower 16 bits
Division mode	MDBH: Divisor (higher 16 bits) MDBL: Divisor (lower 16 bits)	–

(3) Multiplication/division data register C (MDCL, MDCH)

The MDCH and MDCL registers are used to store the accumulated result while in the multiply-accumulator mode or the remainder of the operation result while in the division mode. These registers are not used while in the multiplication mode.

The MDCH and MDCL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 16-4. Format of Multiplication/Division Data Register C (MDCH, MDCL)

Address: F00E0H, F00E1H, F00E2H, F00E3H After reset: 0000H, 0000H R/W

Symbol	F00E3H								F00E2H							
MDCH	MDCH.15	MDCH.14	MDCH.13	MDCH.12	MDCH.11	MDCH.10	MDCH.9	MDCH.8	MDCH.7	MDCH.6	MDCH.5	MDCH.4	MDCH.3	MDCH.2	MDCH.1	MDCH.0

Symbol	F00E1H								F00E0H							
MDCL	MDCL.15	MDCL.14	MDCL.13	MDCL.12	MDCL.11	MDCL.10	MDCL.9	MDCL.8	MDCL.7	MDCL.6	MDCL.5	MDCL.4	MDCL.3	MDCL.2	MDCL.1	MDCL.0

Cautions 1. The MDCH and MDCL registers values read during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) will not be guaranteed.

2. During multiply-accumulator processing (when the MDUC register value is 41H or 49H), do not use software to rewrite the values of the MDCH and MDCL registers. If this is done, the operation result will be undefined.

3. The data is in the two's complement format in the multiply-accumulator mode (signed).

Table 16-4. Functions of MDCH and MDCL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned or signed)	–	–
Multiply-accumulator mode (unsigned)	MDCH: Initial accumulated value (unsigned) (higher 16 bits) MDCL: Initial accumulated value (unsigned) (lower 16 bits)	MDCH: accumulated value (unsigned) (higher 16 bits) MDCL: accumulated value (unsigned) (lower 16 bits)
Multiply-accumulator mode (signed)	MDCH: Initial accumulated value (signed) (higher 16 bits) MDCL: Initial accumulated value (signed) (lower 16 bits)	MDCH: accumulated value (signed) (higher 16 bits) MDCL: accumulated value (signed) (lower 16 bits)
Division mode	–	MDCH: Remainder (higher 16 bits) MDCL: Remainder (lower 16 bits)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

  <Multiplier A> <Multiplier B> <Product>

  MDAL (bits 15 to 0) × MDAH (bits 15 to 0) = [MDBH (bits 15 to 0), MDBL (bits 15 to 0)]

- Register configuration during multiply-accumulation

  <Multiplier A> <Multiplier B> < accumulated value > < accumulated result >

  MDAL (bits 15 to 0) × MDAH (bits 15 to 0) + MDC (bits 31 to 0) = [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

  (The multiplication result is stored in the MDBH (bits 15 to 0) and MDBL (bits 15 to 0).)

- Register configuration during division

  <Dividend> <Divisor>

  [MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ÷ [MDBH (bits 15 to 0), MDBL (bits 15 to 0)] =

  <Quotient> <Remainder>

  [MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ··· [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

16.3 Register Controlling Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator is controlled by using the multiplication/division control register (MDUC).

(1) Multiplication/division control register (MDUC)

The MDUC register is an 8-bit register that controls the operation of the multiplier and divider/multiply-accumulator.

The MDUC 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 Multiplication/Division Control Register (MDUC)

Address: F00E8H After reset: 00H R/W

Symbol	<7>	<6>	5	4	<3>	<2>	<1>	<0>
MDUC	DIVMODE	MACMODE	0	0	MDSM	MACOF	MACSF	DIVST

DIVMODE	MACMODE	MDSM	Operation mode selection
0	0	0	Multiplication mode (unsigned) (default)
0	0	1	Multiplication mode (signed)
0	1	0	Multiply-accumulator mode (unsigned)
0	1	1	Multiply-accumulator mode (signed)
1	0	0	Division mode (unsigned), generation of a division completion interrupt (INTMD occurs/does not occur)
1	1	0	Division mode (unsigned), not generation of a division completion interrupt (INTMD does not occur)
Other than above			Setting prohibited

MACOF[Note1]	Overflow flag of multiply-accumulation result (accumulated value)
0	No overflow
1	With over flow
<Set condition> • For the multiply-accumulator mode (unsigned) The bit is set when the accumulated value goes outside the range from 00000000h to FFFFFFFFh. • For the multiply-accumulator mode (signed) The bit is set when the result of adding a positive product to a positive accumulated value exceeds 7FFFFFFFh and is negative, or when the result of adding a negative product to a negative accumulated value exceeds 80000000h and is positive.	

MACSF	Sign flag of multiply-accumulation result (accumulated value)
0	The accumulated value is positive.
1	The accumulated value is negative.
Multiply-accumulator mode (unsigned):	The bit is always 0.
Multiply-accumulator mode (signed):	The bit indicates the sign bit of the accumulated value.

DIVST[Note 2]	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

Notes 1. The MACOF bit is read only.

2. The DIVST bit can only be set (1) in the division mode. In the division mode, division operation is started by setting (1) the DIVST bit. The DIVST bit is automatically cleared (0) when the operation ends. In the multiplication mode, operation is automatically started by setting the multiplier and multiplicand to multiplication/division data register A (MDAH, MDAL), respectively.

Cautions 1. Do not rewrite the DIVMODE, MDSM bits during operation processing (while the DIVST bit is 1). If it is rewritten, the operation result will be an undefined value.

2. The DIVST bit cannot be cleared (0) by using software during division operation processing (while the DIVST bit is 1).

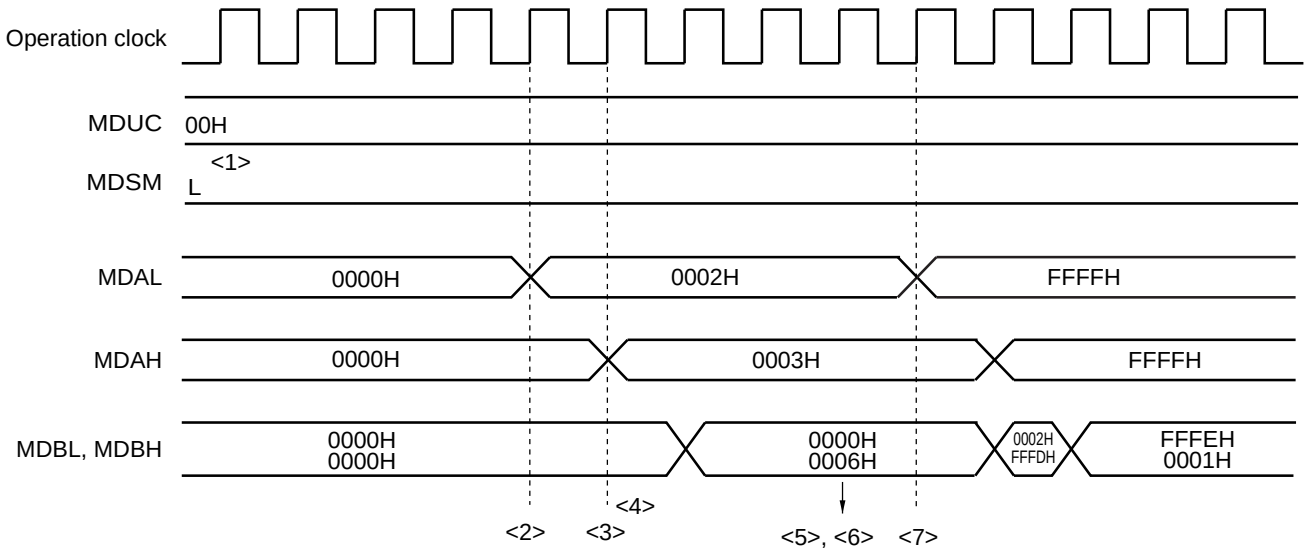
16.4 Operations of Multiplier and Divider/Multiply-Accumulator

16.4.1 Multiplication (unsigned) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 00H.
  - <2> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> To execute multiplication, division, or multiply-accumulation next, start with the initial settings of each step.

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 16-6.

Figure 16-6. Timing Diagram of Multiplication (Unsigned) Operation (2 × 3 = 6)



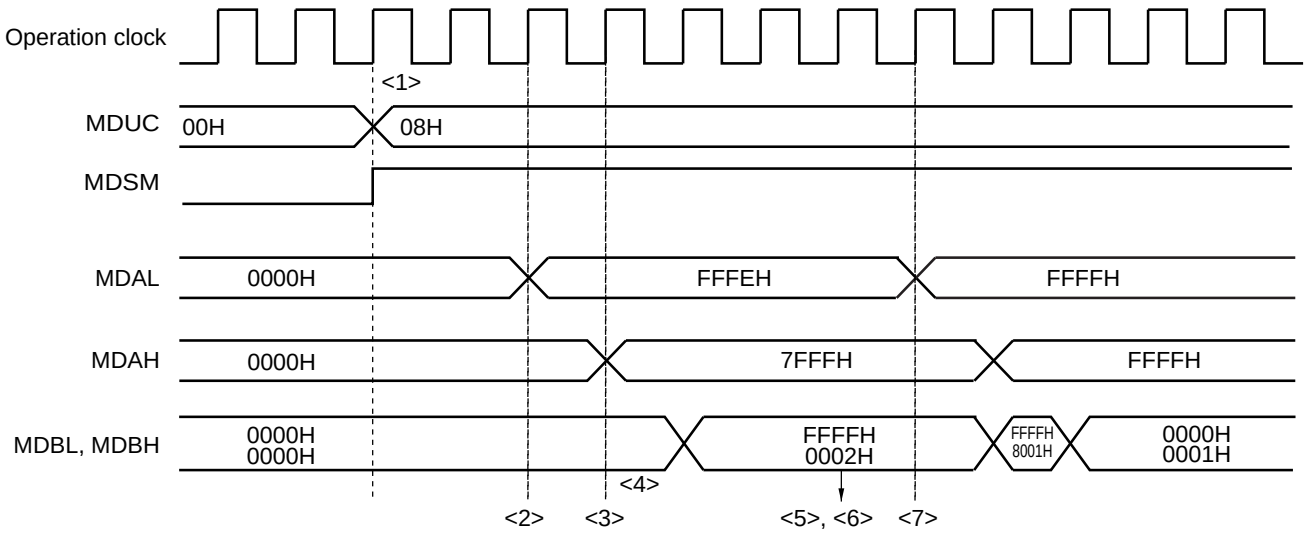
16.4.2 Multiplication (signed) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 08H.
  - <2> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> To execute multiplication (signed) operation next, start from the “Initial setting” for multiplication (signed) operation.
  - <8> The next time multiplication (unsigned), multiply-accumulation (signed or unsigned), or division is performed, start with the initial settings of each step.

Caution The data is in the two's complement format in multiplication mode (signed).

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 16-7.

Figure 16-7. Timing Diagram of Multiplication (Signed) Operation (–2 × 32767 = –65534)

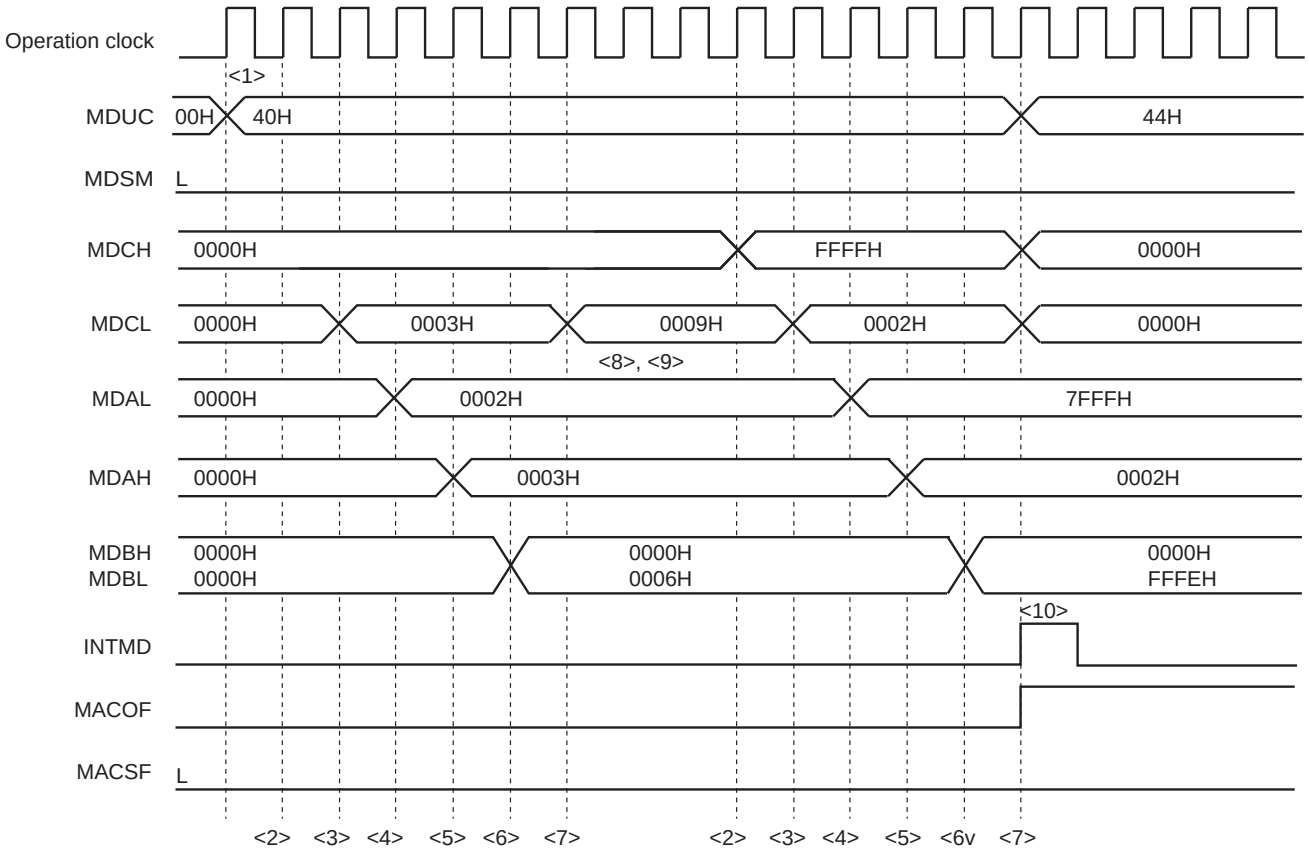


16.4.3 Multiply-accumulation (unsigned) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 40H.
  - <2> Set the initial accumulated value of higher 16 bits to multiplication/division data register C (L) (MDCL).
  - <3> Set the initial accumulated value of lower 16 bits to multiplication/division data register C (H) (MDCH).
  - <4> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <5> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>, or <4> and <5>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <6> The multiplication operation finishes in one clock cycle.
    (The multiplication result is stored in multiplication/division data register B (L) (MDBL) and multiplication/division data register B (H) (MDBH).)
  - <7> After <6>, the multiply-accumulation operation finishes in one additional clock cycle. (There is a wait of at least two clock cycles after specifying the initial settings is finished (<5>).)
- Operation end
  - <8> Read the accumulated value (lower 16 bits) from the MDCL register.
  - <9> Read the accumulated value (higher 16 bits) from the MDCH register.
    (There is no preference in the order of executing steps <8> and <9>.)
  - (<10> If the result of the multiply-accumulation operation causes an overflow, the MACOF bit is set to 1, and 0000H is stored in the MDCH and MDCL registers.)
- Next operation
  - <11> To execute multiply-accumulation (unsigned) operation next, start from the “Initial setting” for multiply-accumulation (unsigned) operation.
  - <12> The next time multiplication (signed or unsigned), multiply-accumulation (signed), or division is performed, start with the initial settings of each step.

Remark Steps <1> to <10> correspond to <1> to <10> in Figure 16-8.

Figure 16-8. Timing Diagram of Multiply-Accumulation (Unsigned) Operation
(2 × 3 + 3 = 9 → 32767 × 2 + 429401762 = 0 (over flow generated))



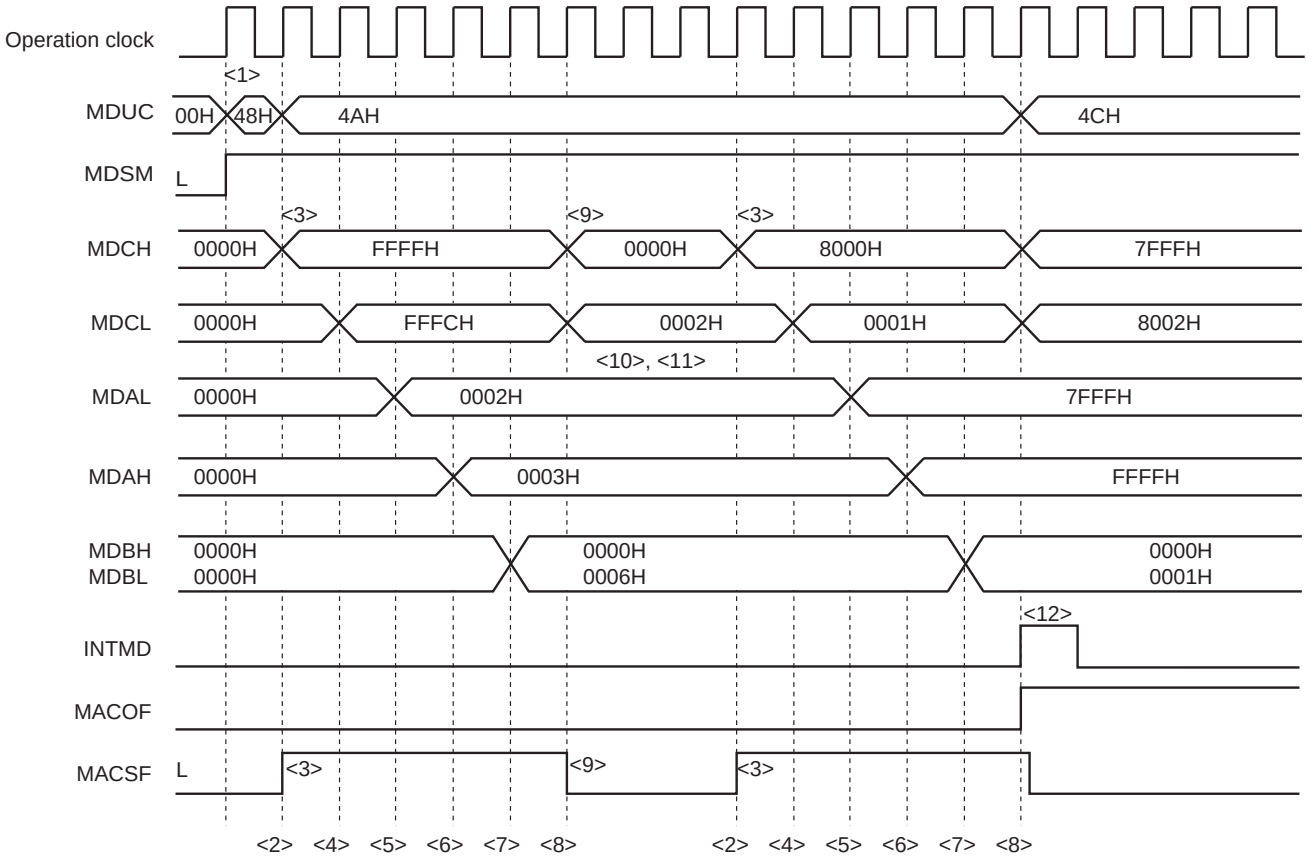
16.4.4 Multiply-accumulation (signed) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 48H.
  - <2> Set the initial accumulated value of higher 16 bits to multiplication/division data register C (H) (MDCH).
    (<3> If the accumulated value in the MDCH register is negative, the MACSF bit is set to 1.)
  - <4> Set the initial accumulated value of lower 16 bits to multiplication/division data register C (L) (MDCL).
  - <5> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <6> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <4>, or <5> and <6>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <7> The multiplication operation finishes in one clock cycle.
    (The multiplication result is stored in multiplication/division data register B (L) (MDBL) and multiplication/division data register B (H) (MDBH).)
  - <8> After <7>, the multiply-accumulation operation finishes in one additional clock cycle. (There is a wait of at least two clock cycles after specifying the initial settings is finished (<6>).)
- Operation end
  - <9> If the accumulated value stored in the MDCL and MDCH registers is positive, the MACSF bit is cleared to 0.
  - <10> Read the accumulated value (lower 16 bits) from the MDCL register.
  - <11> Read the accumulated value (higher 16 bits) from the MDCH register.
    (There is no preference in the order of executing steps <10> and <11>.)
- Next operation
  - <12> To execute multiply-accumulation (signed) operation next, start from the “Initial setting” for multiply-accumulation (signed) operation.
  - <13> The next time multiplication (signed or unsigned), multiply-accumulation (unsigned), or division is performed, start with the initial settings of each step.

Caution The data is in the two's complement format in multiply-accumulation (signed) operation.

Remark Steps <1> to <11> correspond to <1> to <11> in Figure 16-9.

Figure 16-9. Timing Diagram of Multiply-Accumulation (signed) Operation
(2 × 3 + (–4) = 2 → 32767 × (–1) + (–2147483647) = –2147450882 (overflow generated))



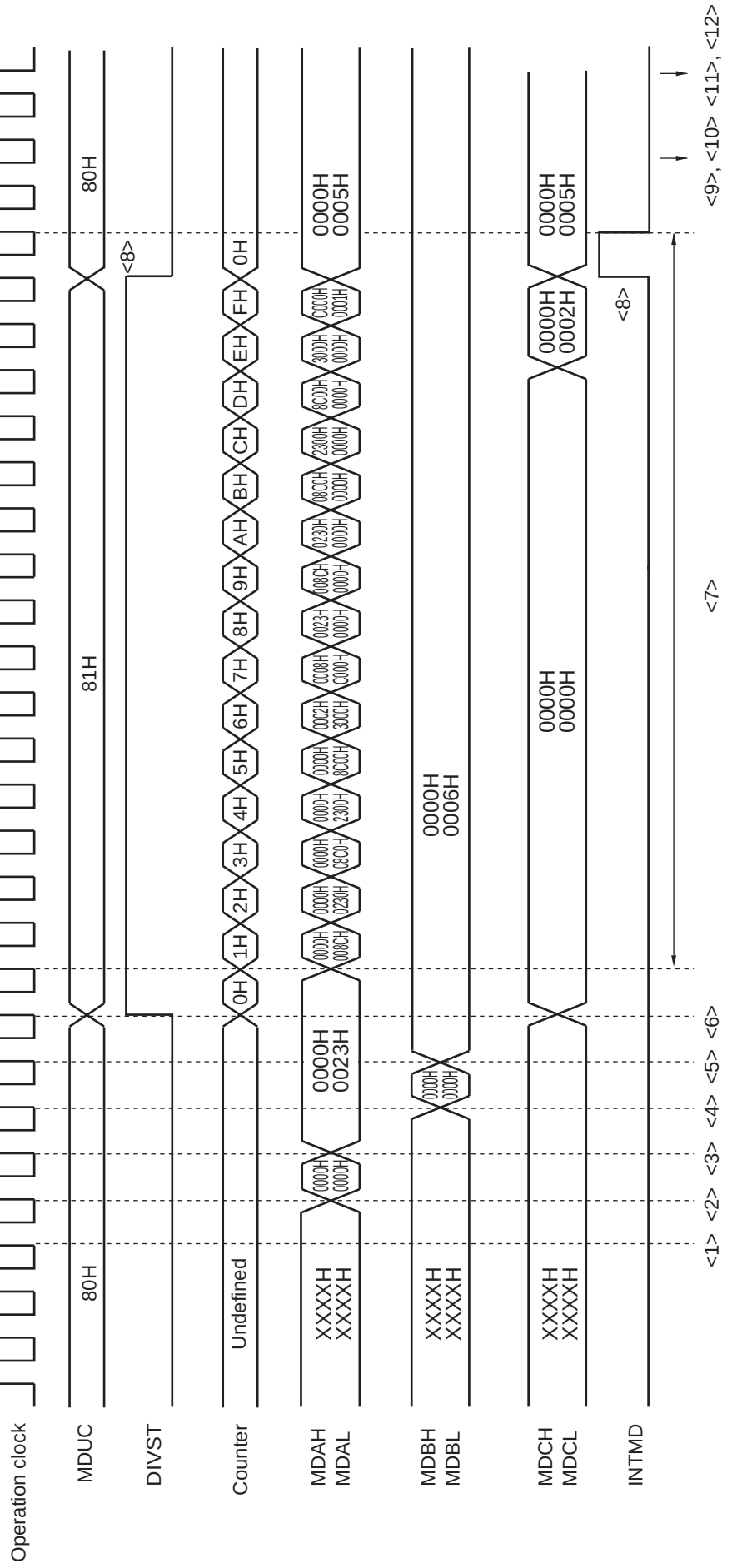
16.4.5 Division operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 80H.
  - <2> Set the dividend (higher 16 bits) to multiplication/division data register A (H) (MDAH).
  - <3> Set the dividend (lower 16 bits) to multiplication/division data register A (L) (MDAL).
  - <4> Set the divisor (higher 16 bits) to multiplication/division data register B (H) (MDBH).
  - <5> Set the divisor (lower 16 bits) to multiplication/division data register B (L) (MDBL).
  - <6> Set bit 0 (DIVST) of the MDUC register to 1.
    (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  - <7> The operation will end when one of the following processing is completed.
    - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
    - A check whether the DIVST bit has been cleared

    (The read values of the MDBL, MDBH, MDCL, and MDCH registers during operation processing are not guaranteed.)
- Operation end
  - <8> The DIVST bit is cleared and the operation ends. At this time, an interrupt request signal (INTMD) is generated if the operation was performed with MACMODE = 0.
  - <9> Read the quotient (lower 16 bits) from the MDAL register.
  - <10> Read the quotient (higher 16 bits) from the MDAH register.
  - <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  - <12> Read the remainder (higher 16 bits) from multiplication/division data register C (H) (MDCH).
    (There is no preference in the order of executing steps <9> to <12>.)
- Next operation
  - <13> To execute multiplication, division, or multiply-accumulation next, start with the initial settings of each step.
  - <14> The next time multiplication (signed or unsigned) or multiply-accumulation (signed or unsigned) is performed, start with the initial settings of each step.

Remark Steps <1> to <12> correspond to <1> to <12> in Figure 16-10.

Figure 16-10. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



CHAPTER 17 DMA CONTROLLER

The RL78/F12 has an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between the peripheral hardware supporting DMA, SFRs, and internal RAM without via CPU.

As a result, the normal internal operation of the CPU and data transfer can be executed in parallel with transfer between the SFR and internal RAM, and therefore, a large capacity of data can be processed. In addition, real-time control using communication, timer, and A/D can also be realized.

17.1 Functions of DMA Controller

- Number of DMA channels: 2 channels
- Transfer unit: 8 or 16 bits
- Maximum transfer unit: 1024 times
- Transfer type: 2-cycle transfer (One transfer is processed in 2 clocks and the CPU stops during that processing.)
- Transfer mode: Single-transfer mode
- Transfer request: Selectable from the following peripheral hardware interrupts
  - A/D converter
  - Serial interface
    (CSIS0, CSI01, CSI11, CSI20, CSI21, UARTS0, UART1, UART2)
  - Timer (channel 0, 1, 2, 3)
- Transfer target: Between SFR and internal RAM

Here are examples of functions using DMA.

- Successive transfer of serial interface
- Batch transfer of analog data
- Capturing A/D conversion result at fixed interval
- Capturing port value at fixed interval

17.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 17-1. Configuration of DMA Controller

Item	Configuration
Address registers	• DMA SFR address registers 0, 1 (DSA0, DSA1) • DMA RAM address registers 0, 1 (DRA0, DRA1)
Count register	• DMA byte count registers 0, 1 (DBC0, DBC1)
Control registers	• DMA mode control registers 0, 1 (DMC0, DMC1) • DMA operation control register 0, 1 (DRC0, DRC1)

(1) DMA SFR address register n (DSAn)

This is an 8-bit register that is used to set an SFR address that is the transfer source or destination of DMA channel n.

Set the lower 8 bits of the SFR addresses FFF00H to FFFFFH.

This register is not automatically incremented but fixed to a specific value.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DSAn register can be read or written in 8-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 00H.

Figure 17-1. Format of DMA SFR Address Register n (DSAn)

Address: FFFB0H (DSA0), FFFB1H (DSA1) After reset: 00H R/W

	7	6	5	4	3	2	1	0
DSAn								

Remark n: DMA channel number (n = 0, 1)

(2) DMA RAM address register n (DRAn)

This is a 16-bit register that is used to set a RAM address that is the transfer source or destination of DMA channel n.

Addresses of the internal RAM area other than the general-purpose registers (FFB00H to FFEDFH in the case of the R5F1096A, R5F109AA, R5F109BA, and R5F109LA) can be set to this register.

Set the lower 16 bits of the RAM address.

This register is automatically incremented when DMA transfer has been started. It is incremented by +1 in the 8-bit transfer mode and by +2 in the 16-bit transfer mode. DMA transfer is started from the address set to this DRAn register. When the data of the last address has been transferred, the DRAn register stops with the value of the last address +1 in the 8-bit transfer mode, and the last address +2 in the 16-bit transfer mode.

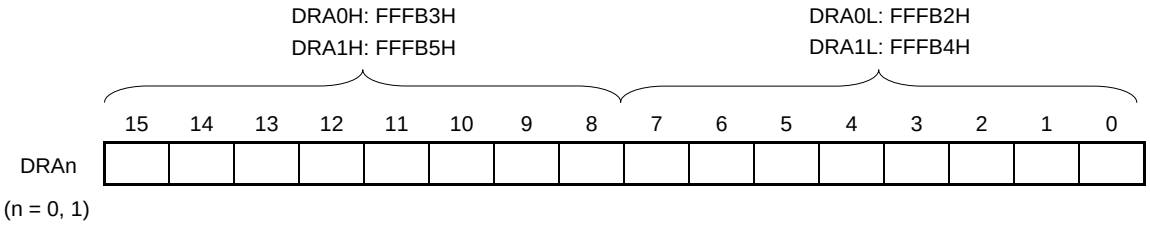
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DRAn register can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 17-2. Format of DMA RAM Address Register n (DRAn)

Address: FFFB2H, FFFB3H (DRA0), FFFB4H, FFFB5H (DRA1) After reset: 0000H R/W



Remark n: DMA channel number (n = 0, 1)

(3) DMA byte count register n (DBCn)

This is a 10-bit register that is used to set the number of times DMA channel n executes transfer. Be sure to set the number of times of transfer to this DBCn register before executing DMA transfer (up to 1024 times).

Each time DMA transfer has been executed, this register is automatically decremented. By reading this DBCn register during DMA transfer, the remaining number of times of transfer can be learned.

The DBCn register can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 17-3. Format of DMA Byte Count Register n (DBCn)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W

DBC0H: FFFB7H
DBC1H: FFFB9H

DBC0L: FFFB6H
DBC1L: FFFB8H

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

(n = 0, 1)

DBCn[9:0]	Number of Times of Transfer (When DBCn is Written)	Remaining Number of Times of Transfer (When DBCn is Read)
000H	1024	Completion of transfer or waiting for 1024 times of DMA transfer
001H	1	Waiting for remaining one time of DMA transfer
002H	2	Waiting for remaining two times of DMA transfer
003H	3	Waiting for remaining three times of DMA transfer
•	•	•
•	•	•
•	•	•
3FEH	1022	Waiting for remaining 1022 times of DMA transfer
3FFH	1023	Waiting for remaining 1023 times of DMA transfer

Cautions 1. Be sure to clear bits 15 to 10 to "0".

2. If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.

Remark n: DMA channel number (n = 0, 1)

17.3 Registers Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)

Remark n: DMA channel number (n = 0, 1)

(1) DMA mode control register n (DMCn)

The DMCn register is a register that is used to set a transfer mode of DMA channel n. It is used to select a transfer direction, data size, setting of pending, and start source. Bit 7 (STGn) is a software trigger that starts DMA.

Rewriting bits 6, 5, and 3 to 0 of the DMCn register is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 17-4. Format of DMA Mode Control Register n (DMCn) (1/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn[Note 1]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is performed once by writing 1 to the STGn bit when DMA operation is enabled (DENn = 1). When this bit is read, 0 is always read.	

DRSn	Selection of DMA transfer direction
0	SFR to internal RAM
1	Internal RAM to SFR

DSn	Specification of transfer data size for DMA transfer
0	8 bits
1	16 bits

DWAITn[Note 2]	Pending of DMA transfer
0	Executes DMA transfer upon DMA start request (not held pending).
1	Holds DMA start request pending if any.
DMA transfer that has been held pending can be started by clearing the value of the DWAITn bit to 0. It takes 2 clocks to actually hold DMA transfer pending when the value of the DWAITn bit is set to 1.	

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

2. When DMA transfer is held pending while using two or more DMA channels, be sure to hold the DMA transfer pending for all channels (by setting the DWAIT0, and DWAIT1 bits to 1).

Remark n: DMA channel number (n = 0, 1)

Figure 17-4. Format of DMA Mode Control Register n (DMCn) (2/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMA start source[Note]	
				Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	0	1	INTAD	A/D conversion end interrupt
0	0	1	0	INTTM00	End of timer channel 0 count or capture end interrupt
0	0	1	1	INTTM01	End of timer channel 1 count or capture end interrupt
0	1	0	0	INTTM02	End of timer channel 2 count or capture end interrupt
0	1	0	1	INTTM03	End of timer channel 3 count or capture end interrupt
0	1	1	0	INTST0/INTCSI00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt
0	1	1	1	INTSR0/INTCSI01	UART0 reception transfer end interrupt/CSI01 transfer end or buffer empty interrupt
1	0	0	0	INTST1	UART1 transmission transfer end or buffer empty interrupt
1	0	0	1	INTSR1/INTCSI11	UART1 reception transfer end interrupt/CSI11 transfer end or buffer empty interrupt
1	0	1	0	INTST2/INTCSI20	UART2 transmission transfer end or buffer empty interrupt/CSI20 transfer end or buffer empty interrupt
1	0	1	1	INTSR2/INTCSI21	UART2 reception transfer end interrupt/CSI21 transfer end or buffer empty interrupt
Other than above				Setting prohibited	

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

Remark n: DMA channel number (n = 0, 1)

(2) DMA operation control register n (DRCn)

The DRCn register is a register that is used to enable or disable transfer of DMA channel n.

Rewriting bit 7 (DENn) of this register is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 17-5. Format of DMA Operation Control Register n (DRCn)

Address: FFFBCH (DRC0), FFFBDH (DRC1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA operation enable flag
0	Disables operation of DMA channel n (stops operating cock of DMA).
1	Enables operation of DMA channel n.
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1).	

DSTn	DMA transfer mode flag
0	DMA transfer of DMA channel n is completed.
1	DMA transfer of DMA channel n is not completed (still under execution).
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by the IFCn3 to IFCn0 bits is input, DMA transfer is started. When DMA transfer is completed after that, this bit is automatically cleared to 0. Write 0 to this bit to forcibly terminate DMA transfer under execution.	

Caution The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set the DSTn bit to 0 and then the DENn bit to 0 (for details, refer to 17.5.5 Forced termination by software).

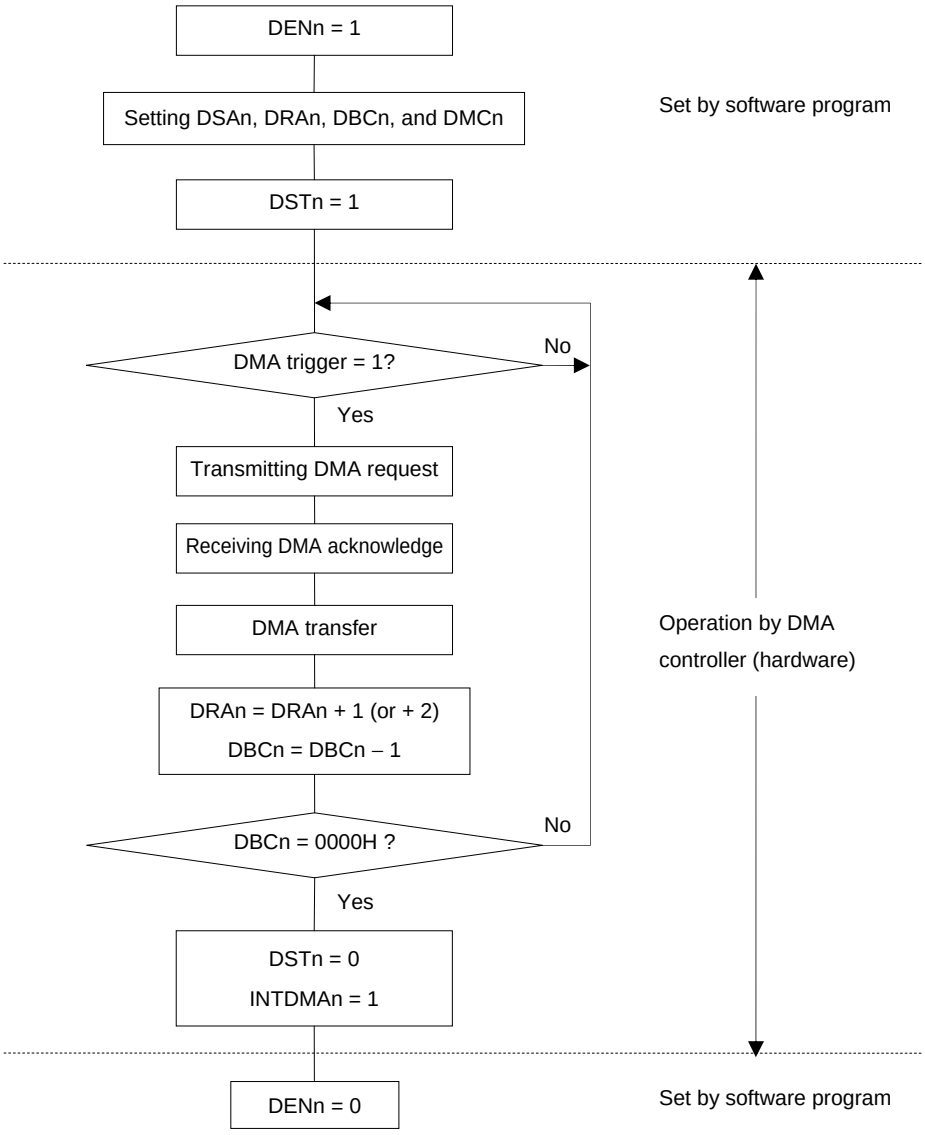
Remark n: DMA channel number (n = 0, 1)

17.4 Operation of DMA Controller

17.4.1 Operation procedure

<1> The DMA controller is enabled to operate when DENn = 1. Before writing the other registers, be sure to set the DENn bit to 1. Use 80H to write with an 8-bit manipulation instruction.
<2> Set an SFR address, a RAM address, the number of times of transfer, and a transfer mode of DMA transfer to DMA SFR address register n (DSAn), DMA RAM address register n (DRAn), DMA byte count register n (DBCn), and DMA mode control register n (DMCn).
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.
<4> When a software trigger (STGn) or a start source trigger specified by the IFCn3 to IFCn0 bits is input, a DMA transfer is started.
<5> Transfer is completed when the number of times of transfer set by the DBCn register reaches 0, and transfer is automatically terminated by occurrence of an interrupt (INTDMAn).
<6> Stop the operation of the DMA controller by clearing the DENn bit to 0 when the DMA controller is not used.

Figure 17-6. Operation Procedure



Remark n: DMA channel number (n = 0, 1)

17.4.2 Transfer mode

The following four modes can be selected for DMA transfer by using bits 6 and 5 (DRSn and DSn) of DMA mode control register n (DMCn).

DRSn	DSn	DMA Transfer Mode
0	0	Transfer from SFR of 1-byte data (fixed address) to RAM (address is incremented by +1)
0	1	Transfer from SFR of 2-byte data (fixed address) to RAM (address is incremented by +2)
1	0	Transfer from RAM of 1-byte data (address is incremented by +1) to SFR (fixed address)
1	1	Transfer from RAM of 2-byte data (address is incremented by +2) to SFR (fixed address)

By using these transfer modes, up to 1024 bytes of data can be consecutively transferred by using the serial interface, data resulting from A/D conversion can be consecutively transferred, and port data can be scanned at fixed time intervals by using a timer.

17.4.3 Termination of DMA transfer

When DBCn = 00H and DMA transfer is completed, the DSTn bit is automatically cleared to 0. An interrupt request (INTDMAn) is generated and transfer is terminated.

When the DSTn bit is cleared to 0 to forcibly terminate DMA transfer, DMA byte count register n (DBCn) and DMA RAM address register n (DRAn) hold the value when transfer is terminated.

The interrupt request (INTDMAn) is not generated if transfer is forcibly terminated.

Remark n: DMA channel number (n = 0, 1)

17.5 Example of Setting of DMA Controller

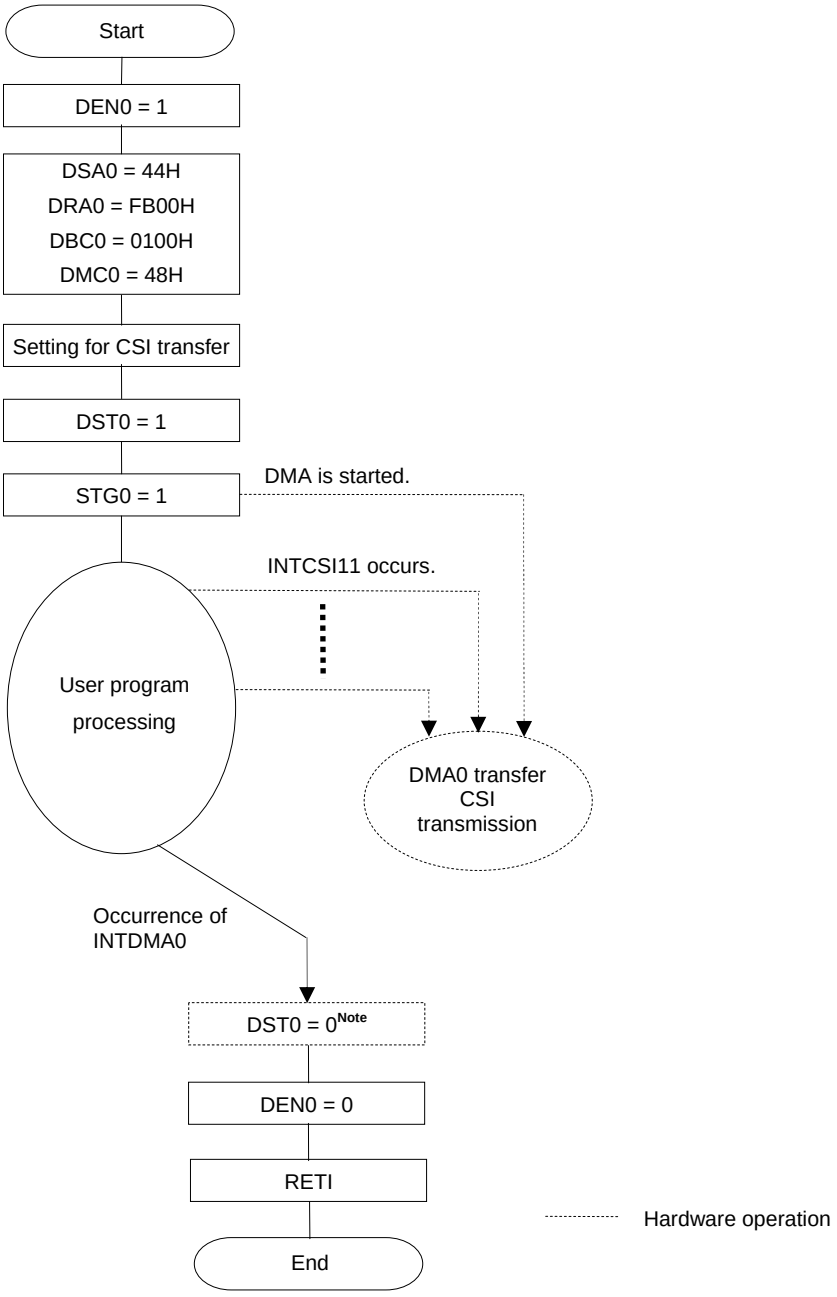
17.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission of CSI11 (256 bytes)
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI11 (software trigger (STG0) only for the first start source)
- Interrupt of CSI11 is specified by IFC03 to IFC00 = 1001B.
- Transfers FFB00H to FFBFFH (256 bytes) of RAM to FFF46H of the data register (SIO11) of CSI.

Remark IFC03 to IFC00: Bits 3 to 0 of DMA mode control registers 0 (DMC0)

Figure 17-7. Example of Setting for CSI Consecutive Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set the DST0 bit to 0 and then the DEN0 bit to 0 (for details, refer to **17.5.5 Forced termination by software**).

The fist trigger for consecutive transmission is not started by the interrupt of CSI. In this example, it start by a software trigger.

CSI transmission of the second time and onward is automatically executed.

A DMA interrupt (INTDMA0) occurs when the last transmit data has been written to the data register.

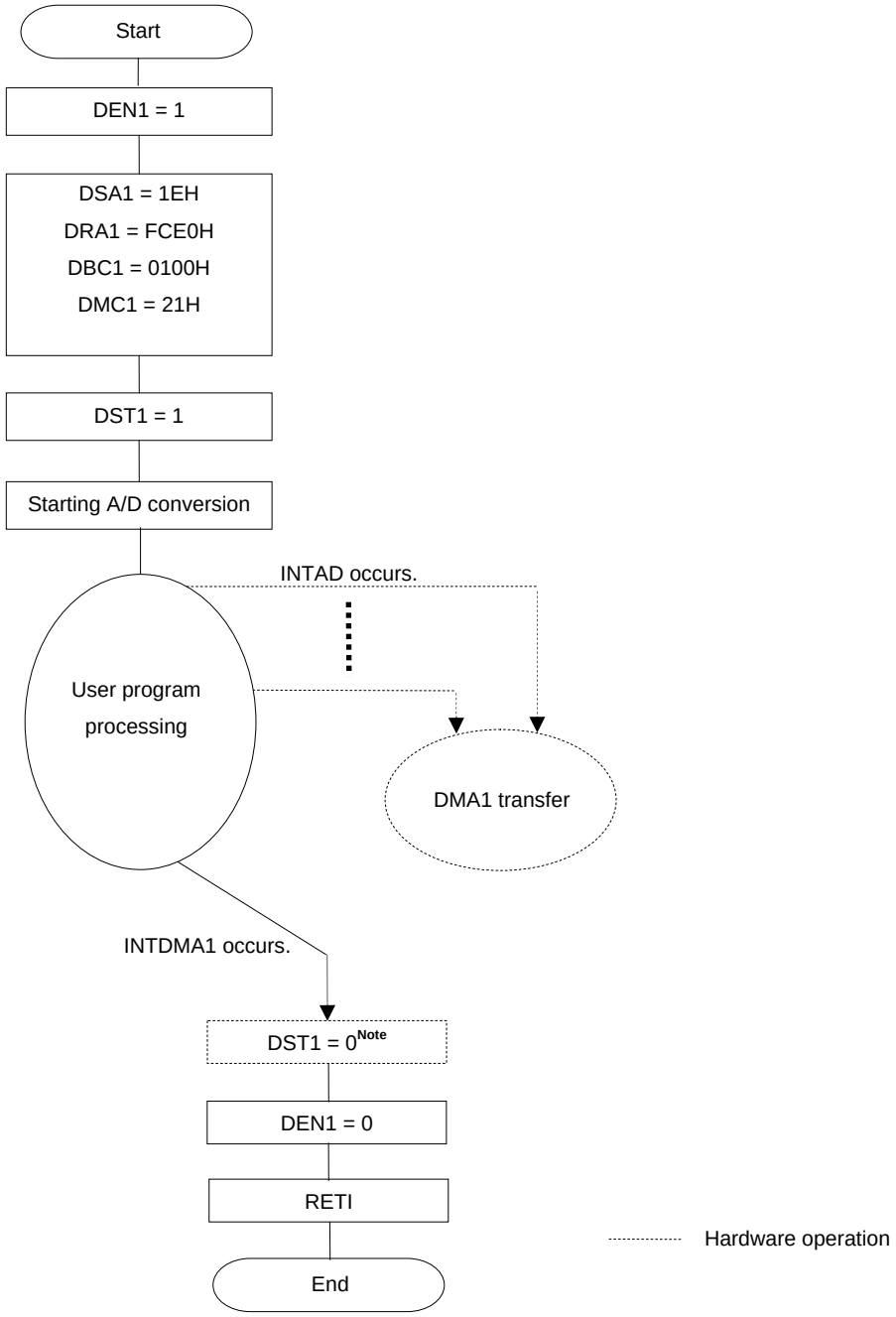
17.5.2 Consecutive capturing of A/D conversion results

A flowchart of an example of setting for consecutively capturing A/D conversion results is shown below.

- Consecutive capturing of A/D conversion results.
- DMA channel 1 is used for DMA transfer.
- DMA start source: INTAD
- Interrupt of A/D is specified by IFC13 to IFC10 = 0001B.
- Transfers FFF1EH and FFF1FH (2 bytes) of the 10-bit A/D conversion result register (ADCR) to 512 bytes of FFCE0H to FFEDFH of RAM.

Remark IFC13 to IFC10: Bits 3 to 0 of DMA mode control registers 1 (DMC1)

Figure 17-8. Example of Setting of Consecutively Capturing A/D Conversion Results



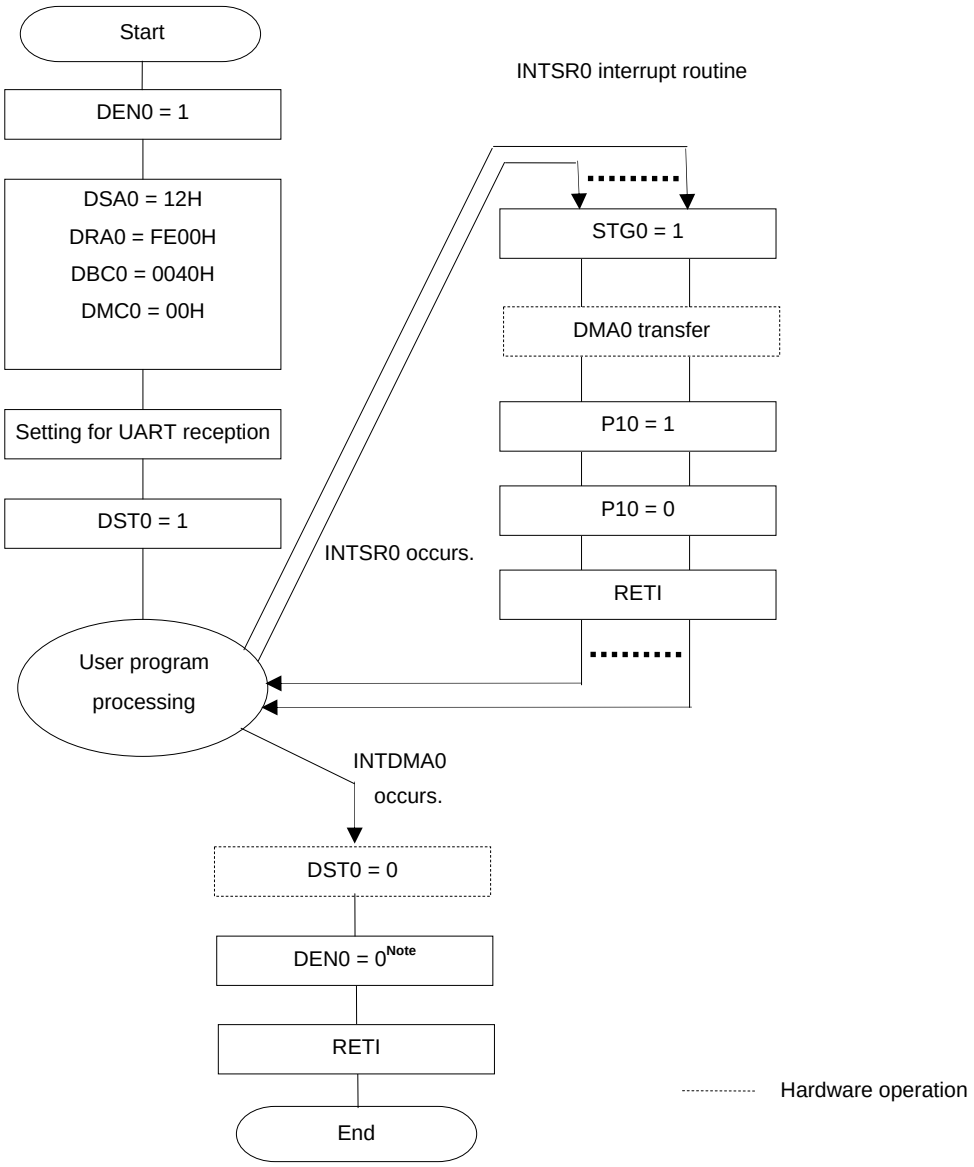
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set the DST1 bit to 0 and then the DEN1 bit to 0 (for details, refer to **17.5.5 Forced termination by software**).

17.5.3 UART consecutive reception + ACK transmission

A flowchart illustrating an example of setting for UART consecutive reception + ACK transmission is shown below.

- Consecutively receives data from UART0 and outputs ACK to P10 on completion of reception.
- DMA channel 0 is used for DMA transfer.
- DMA start source: Software trigger (DMA transfer on occurrence of an interrupt is disabled.)
- Transfers FFF12H of UART receive data register 0 (RXD0) to 64 bytes of FFE00H to FFE3FH of RAM.

Figure 17-9. Example of Setting for UART Consecutive Reception + ACK Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set the DST0 bit to 0 and then the DEN0 bit to 0 (for details, refer to **17.5.5 Forced termination by software**).

Remark This is an example where a software trigger is used as a DMA start source.
If ACK is not transmitted and if only data is consecutively received from UART, the UART reception end interrupt (INTSR0) can be used to start DMA for data reception.

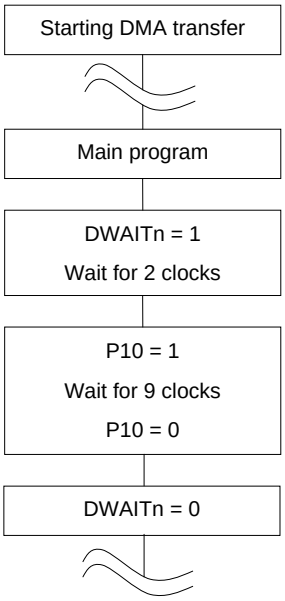
17.5.4 Holding DMA transfer pending by DWAITn bit

When DMA transfer is started, transfer is performed while an instruction is executed. At this time, the operation of the CPU is stopped and delayed for the duration of 2 clocks. If this poses a problem to the operation of the set system, a DMA transfer can be held pending by setting the DWAITn bit to 1. The DMA transfer for a transfer trigger that occurred while DMA transfer was held pending is executed after the pending status is canceled. However, because only one transfer trigger can be held pending for each channel, even if multiple transfer triggers occur for one channel during the pending status, only one DMA transfer is executed after the pending status is canceled.

To output a pulse with a width of 10 clocks of the operating frequency from the P10 pin, for example, the clock width increases to 12 if a DMA transfer is started midway. In this case, the DMA transfer can be held pending by setting the DWAITn bit to 1.

After setting the DWAITn bit to 1, it takes two clocks until a DMA transfer is held pending.

Figure 17-10. Example of Setting for Holding DMA Transfer Pending by DWAITn Bit



Caution When DMA transfer is held pending while using two or more DMA channels, be sure to held the DMA transfer pending for all channels (by setting DWAIT0, DWAIT1, DWAIT2, and DWAIT3 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

17.5.5 Forced termination by software

After the DSTn bit is set to 0 by software, it takes up to 2 clocks until a DMA transfer is actually stopped and the DSTn bit is set to 0. To forcibly terminate a DMA transfer by software without waiting for occurrence of the interrupt (INTDMAn) of DMAn, therefore, perform either of the following processes.

<When using one DMA channel>

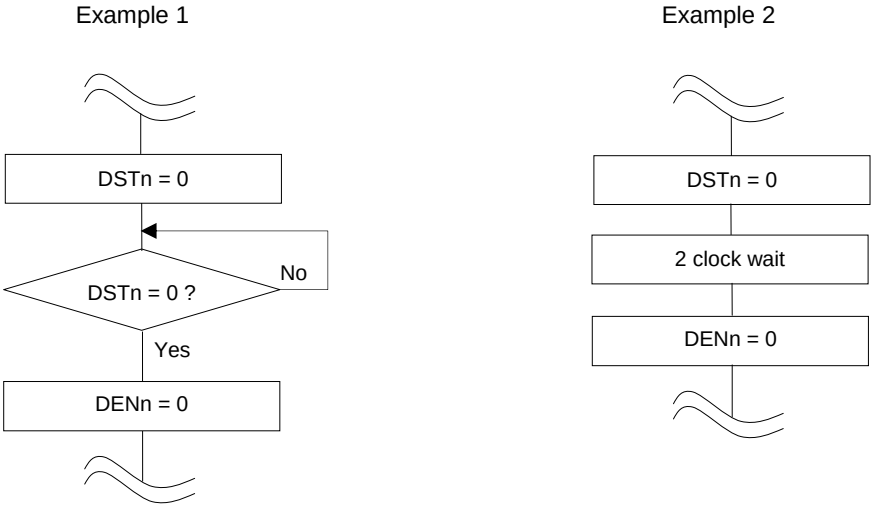
- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that the DSTn bit has actually been cleared to 0, and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

<When using two DMA channels>

- To forcibly terminate DMA transfer by software when using two or more DMA channels (by setting DSTn to 0), clear the DSTn bit to 0 after the DMA transfer is held pending by setting the DWAITn bits of all using channels to 1. Next, clear the DWAITn bits of all using channels to 0 to cancel the pending status, and then clear the DENn bit to 0.

Figure 17-11. Forced Termination of DMA Transfer (1/2)



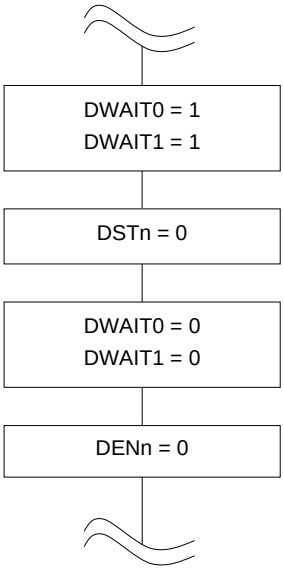
Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

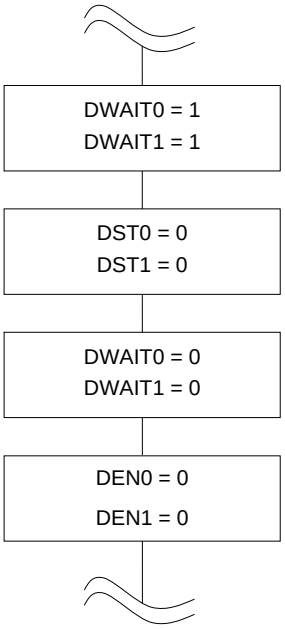
Figure 17-11. Forced Termination of DMA Transfer (2/2)

Example 3

- Procedure for forcibly terminating the DMA transfer for one channel if both channels are used



- Procedure for forcibly terminating the DMA transfer for both channels if both channels are used



Caution In example 3, the system is not required to wait two clock cycles after the DWAITn bit is set to 1. In addition, the system does not have to wait two clock cycles after clearing the DSTn bit to 0, because more than two clock cycles elapse from when the DSTn bit is cleared to 0 to when the DENn bit is cleared to 0.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

17.6 Cautions on Using DMA Controller

(1) Priority of DMA

During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two or more DMA requests are generated at the same time, however, their priority are DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3. If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.

(2) DMA response time

The response time of DMA transfer is as follows.

Table 17-2. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time	3 clocks	10 clocks[Note]

Note The maximum time necessary to execute an instruction from internal RAM is 16 clock cycles.

Cautions 1. The above response time does not include the two clock cycles required for a DMA transfer.

2. When executing a DMA pending instruction (see 17.6 (4)), the maximum response time is extended by the execution time of that instruction to be held pending.

3. Do not specify successive transfer triggers for a channel within a period equal to the maximum response time plus one clock cycle, because they might be ignored.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

(3) Operation in standby mode

The DMA controller operates as follows in the standby mode.

Table 17-3. DMA Operation in Standby Mode

Status	DMA Operation
HALT mode	Normal operation
STOP mode	Stops operation. If DMA transfer and STOP instruction execution contend, DMA transfer may be damaged. Therefore, stop DMA before executing the STOP instruction.
SNOOZE mode	Stops operation

(4) DMA pending instruction

Even if a DMA request is generated, DMA transfer is held pending immediately after the following instructions.

- CALL !addr16
- CALL $!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- Bit manipulation instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PSW each.
- Instruction for accessing the data flash memory

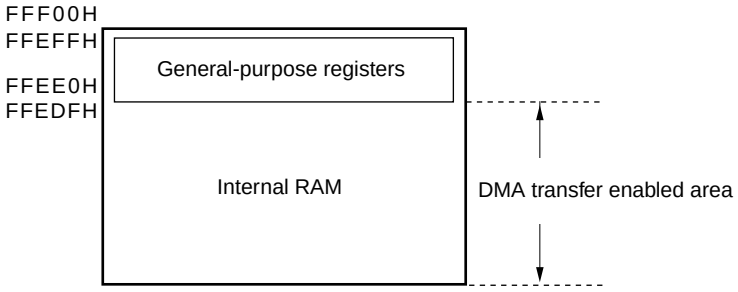
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

The address indicated by DMA RAM address register n (DRAn) is incremented during DMA transfer. If the address is incremented to an address in the general-purpose register area or exceeds the area of the internal RAM, the following operation is performed.

- In mode of transfer from SFR to RAM
  The data of that address is lost.

- In mode of transfer from RAM to SFR
  Undefined data is transferred to SFR.

In either case, malfunctioning may occur or damage may be done to the system. Therefore, make sure that the address is within the internal RAM area other than the general-purpose register area.



(6) Operation if instructions for accessing the data flash area

- Because DMA transfer is suspended to access to the data flash area, be sure to add the DMA pending instruction.
  If the data flash area is accessed after an next instruction execution from start of DMA transfer, a 3-clock wait will be inserted to the next instruction.

```
Instruction 1
DMA transfer
Instruction 2  ◄– The wait of three clock cycles occurs.
MOV A,         ! DataFlash area
```

CHAPTER 18 INTERRUPT FUNCTIONS

The number of interrupt sources differs, depending on the product.

		20-pin	30, 32-pin	48-pin	64-pin
Maskable interrupts	External	5	6	10	13
	Internal	28	33	34	34

18.1 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 priority of vectored interrupt servicing. For the priority order, see **Table 18-1**.

A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

18.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Table 18-1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 18-1. Interrupt Source List (1/3)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	64-pin	48-pin	32-pin	30-pin	20-pin
		Name	Trigger								
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% + 1/2f_{IL} of overflow time)	Internal	0004H	(A)	√	√	√	√	√
	1	INTLVI	Voltage detection Note 4		0006H		√	√	√	√	√
	2	INTP0	Pin input edge detection	External	0008H	(B)	√	√	√	√	√
	3	INTP1			000AH		√	√	√	√	√
	4	INTP2			000CH		√	√	√	√	√
	5	INTP3			000EH		√	√	√	√	–
	6	INTP4			0010H		√	√	√	√	√
	7	INTP5			0012H		√	√	√	√	√
	8	INTST2	UART2 transmission transfer end or buffer empty interrupt	Internal	0014H	(A)	√	√	√	√	–
		INTCSI20	CSI20 transfer end or buffer empty interrupt				√	√	√	√	–
		INTIIC20	IIC20 transfer end				√	√	√	√	–
	9	INTSR2	UART2 reception transfer end		0016H		√	√	√	√	–
		INTCSI21	CSI21 transfer end or buffer empty interrupt				√	√	–	–	–
		INTIIC21	IIC21 transfer end				√	√	–	–	–
	10	INTSRE2	UART2 reception communication error occurrence		0018H		√	√	√	√	–
	11	INTDMA0	End of DMA0 transfer		001AH		√	√	√	√	√
	12	INTDMA1	End of DMA1 transfer		001CH		√	√	√	√	√
	13	INTST0	UART0 transmission transfer end or buffer empty interrupt		001EH		√	√	√	√	√
		INTCSI00	CSI00 transfer end or buffer empty interrupt								
		INTIIC00	IIC00 transfer end								
	14	INTSR0	UART0 reception transfer end		0020H		√	√	√	√	√
		INTCSI01	CSI01 transfer end or buffer empty interrupt				√	√	–	–	–
		INTIIC01	IIC01 transfer end								
	15	INTSRE0	UART0 reception communication error occurrence		0022H		√	√	√	√	√
		INTTM01H	End of timer channel 1 count or capture (at 8-bit timer operation)				√	√	√	√	√
	16	INTST1	UART1 transmission transfer end or buffer empty interrupt		0024H		√	√	√	√	–
		INTCSI10	CSI10 communication end				√	–	–	–	–
		INTIIC10	IIC10 communication end				√	–	–	–	–
	17	INTSR1	UART1 reception transfer end		0026H		√	√	√	√	–
		INTCSI11	CSI11 transfer end or buffer empty interrupt				√	√	√	√	–
		INTIIC11	IIC11 transfer end				√	√	√	√	–

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 42 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 18-1.

3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Table 18-1. Interrupt Source List (2/3)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	64-pin	48-pin	32-pin	30-pin	20-pin
		Name	Trigger								
Maskable	18	INTSRE1	UART1 reception communication error occurrence	Internal	0028H	(A)	√	√	√	√	–
		INTTM03H	End of timer channel 3 count or capture (at 8-bit timer operation)				√	√	√	√	√
	19	INTIICA0	End of IICA0 communication		002AH		√	√	√	√	–
	20	INTTM00	End of timer channel 0 count or capture		002CH		√	√	√	√	√
	21	INTTM01	End of timer channel 1 count or capture		002EH		√	√	√	√	√
	22	INTTM02	End of timer channel 2 count or capture		0030H		√	√	√	√	√
	23	INTTM03	End of timer channel 3 count or capture		0032H		√	√	√	√	√
	24	INTAD	End of A/D conversion		0034H		√	√	√	√	√
	25	INTRTC	Fixed-cycle signal of real-time clock/alarm match detection		0036H		√	√	√	√	√
	26	INTIT	Interval signal detection		0038H		√	√	√	√	√
	27	INTKR	Key return signal detection	External	003AH	(C)	√	√	–	–	–
	28	INTCSIS0	CSIS0 transfer end or buffer empty interrupt	Internal	003CH	(A)	√	√	√	√	√
		INTSTS0	UARTS0 transmission transfer end or buffer empty interrupt				√	√	√	√	√
	29	INTSRS0	UARTS0 reception transfer end		003EH		√	√	√	√	√
		INTCSIS1	CSIS1 transfer end or buffer empty interrupt				√	–	–	–	–
	30	INTWUTM	Wakeup timer compare match		0040H		√	√	√	√	√
	31	INTTM04	End of timer channel 4 count or capture		0042H		√	√	√	√	√
	32	INTTM05	End of timer channel 5 count or capture		0044H		√	√	√	√	√
	33	INTTM06	End of timer channel 6 count or capture		0046H		√	√	√	√	√
	34	INTTM07	End of timer channel 7 count or capture		0048H		√	√	√	√	√
	35	INTP6	Pin input edge detection	External	004AH	(B)	√	√	–	–	–
	36	INTP7	Pin input edge detection	Internal	004CH	(B)	√	–	–	–	–
		INTLT0	LIN-UART reception completion	External		(A)	√	√	√	√	√
	37	INTP8	Pin input edge detection	External	004EH	(B)	√	√	–	–	–
		INTLR0	LIN-UART reception completion	Internal		(A)	√	√	√	√	√
	38	INTP9	Pin input edge detection	External	0050H	(B)	√	√	–	–	–
		INTLS	LIN-UART reception status error	Internal		(A)	√	√	√	√	√
	39	INTP10	Pin input edge detection	External	0052H	(B)	√	–	–	–	–
		INTSRES0	UARTS0 reception communication error occurrence	Internal		(A)	√	√	√	√	√
	40	INTP11	Pin input edge detection	External	0054H	(B)	√	–	–	–	–
	41	INTMD	End of division operation	Internal	005EH	(A)	√	√	√	√	√
	42	INTFL	End of sequencer interrupt		0062H		√	√	√	√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 42 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 18-1.

Table 18-1. Interrupt Source List (3/3)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	48-pin	32-pin	30-pin	20-pin	
		Name	Trigger								
Software	–	BRK	Execution of BRK instruction	–	007EH	(D)	√	√	√	√	√
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–	√	√	√	√	√
		POR	Power-on-reset				√	√	√	√	√
		LVD	Voltage detection Note 3				√	√	√	√	√
		WDT	Overflow of watchdog timer				√	√	√	√	√
		TRAP	Execution of illegal instruction Note 4				√	√	√	√	√
		IAW	Illegal-memory access				√	√	√	√	√
		RAMTOP	RAM parity error				√	√	√	√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 42 indicates the lowest priority.

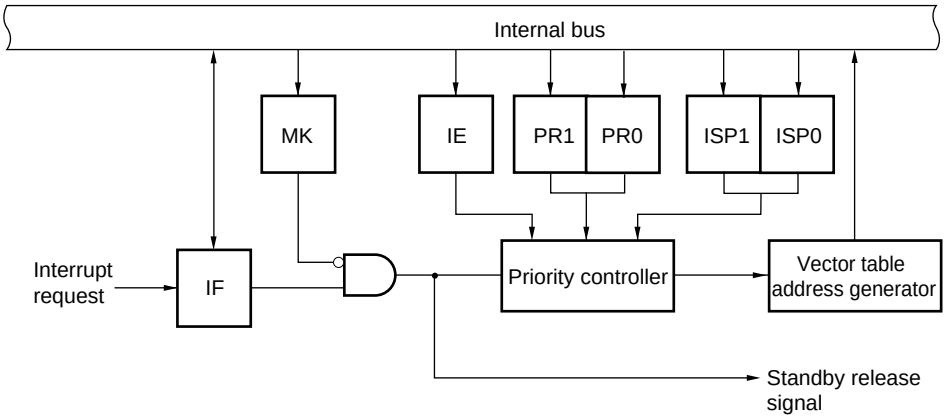
2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 18-1.

3. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

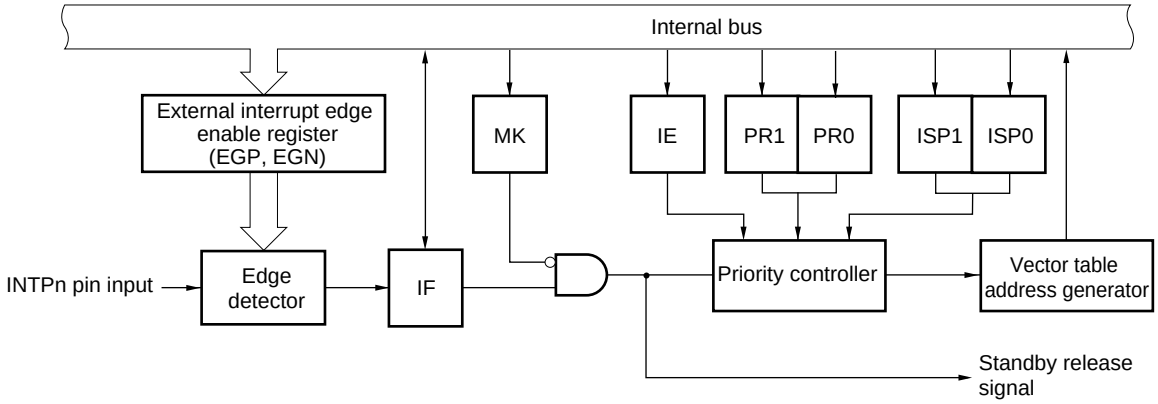
4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 18-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

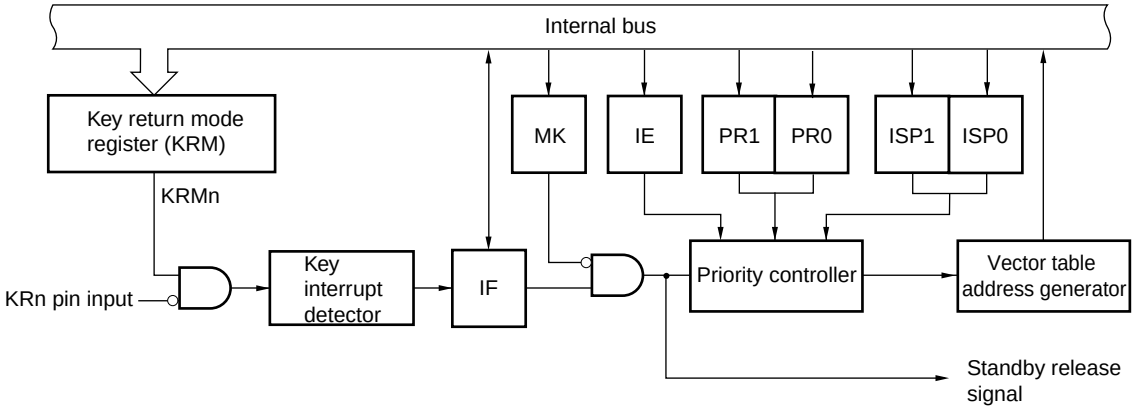


IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

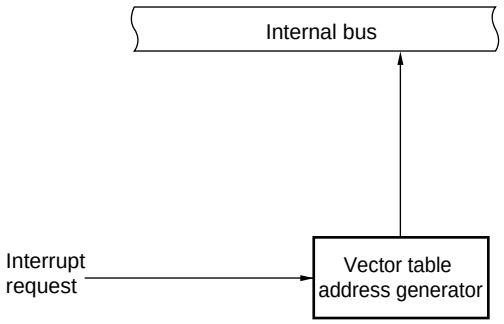
Remark 20-pin: n = 0 to 2, 4, 5
30, 32-pin: n = 0 to 5
48-pin: n = 0 to 6, 8, 9
64-pin: n= 0 to 11

Figure 18-1. Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt (INTKR)



(D) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark 48-pin: n = 0 to 5
64-pin: n= 0 to 7

18.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, 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)

Table 18-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 18-2. Flags Corresponding to Interrupt Request Sources (1/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		64-pin	48-pin	30, 32-pin	20-pin
		Register		Register		Register				
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√	√	√
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		√	√	√	√
INTP0	PIF0		PMK0		PPR00, PPR10		√	√	√	√
INTP1	PIF1		PMK1		PPR01, PPR11		√	√	√	√
INTP2	PIF2		PMK2		PPR02, PPR12		√	√	√	√
INTP3	PIF3		PMK3		PPR03, PPR13		√	√	√	–
INTP4	PIF4		PMK4		PPR04, PPR14		√	√	√	√
INTP5	PIF5		PMK5		PPR05, PPR15		√	√	√	√

Table 18-2. Flags Corresponding to Interrupt Request Sources (2/5)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	64-pin	48-pin	30, 32-pin	20-pin
INTST2Note 1	STIF2Note 1	IF0H	STMK2Note 1	MK0H	STPR02 Note 1, STPR12Note 1	PR00H, PR10H	√	√	√	–
INTCSI20Note 1	CSIIF20Note 1		CSIMK20Note 1		CSIPR020 Note 1, CSIPR120Note 1		√	√	√	–
INTIIC20Note 1	IICIF20Note 1		IICMK20Note 1		IICPR020 Note 1, IICPR120Note 1		√	√	√	–
INTSR2Note2	SRIF2Note2		SRMK2Note2		SRPR02 Note2, SRPR12Note2		√	√	√	–
INTCSI21Note2	CSIIF21Note2		CSIMK21Note2		CSIPR021Note2, CSIPR121Note2		√	√	–	–
INTIIC21Note2	IICIF21Note2		IICMK21Note2		IICPR021Note2, IICPR121Note2		√	√	–	–
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12		√	√	√	–
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10		√	√	√	√
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11		√	√	√	√
INTST0Note 3	STIF0Note 3		STMK0Note 3		STPR00 Note 3, STPR10Note 3		√	√	√	√
INTCSI00Note 3	CSIIF00Note 3		CSIMK00Note 3		CSIPR000 Note 3, CSIPR100Note 3		√	√	√	√
INTIIC00Note 3	IICIF00Note 3		IICMK00Note 3		IICPR000 Note 3, IICPR100Note 3		√	√	√	√
INTSR0 Note 4	SRIF0 Note 4		SRMK0 Note 4		SRPR00 Note 4, SRPR10 Note 4		√	√	√	√
INTCSI01 Note 4	CSIIF01 Note 4		CSIMK01 Note 4		CSIPR001 Note 4, CSIPR101 Note 4		√	√	√	√
INTIIC01 Note 4	IICIF01 Note 4		IICMK01 Note 4		IICPR001 Note 4, IICPR101 Note 4		√	√	√	√
INTSRE0Note 5	SREIF0 Note 5		SREMK0 Note 5		SREPR00 Note 5, SREPR10 Note 5		√	√	√	√
INTTM01H Note 5	TMIF01H Note 5		TMMK01H Note 5		TMPR001H Note 5, TMPR101H Note 5		√	√	√	√

Notes 1. Do not use UART2, CSI20, and IIC20 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST2, INTCSI20, and INTIIC20 is generated, bit 0 of the IF0H register is set to 1. Bit 0 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

2. Do not use UART2, CSI21, and IIC21 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTSR2, INTCSI21, and INTIIC21 is generated, bit 1 of the IF0H register is set to 1. Bit 1 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

3. Do not use UART0, CSI00, and IIC00 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

4. Do not use UART0, CSI01, and IIC01 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST0, INTCSI01, and INTIIC01 is generated, bit 6 of the IF0H register is set to 1. Bit 6 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

5. Do not use UART0 and channel 1 of TAU0 (at 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTSRE0 and INTTM01H is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

Table 18-2. Flags Corresponding to Interrupt Request Sources (3/5)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	64-pin	48-pin	30, 32-pin	20-pin
INTST1 Note 1	STIF1 Note 1	IF1L	STMK1 Note 1	MK1L	STPR01 Note 1, STPR11 Note 1	PR01L, PR11L	√	√	√	–
INTCSI10 Note 1	CSIIF10 Note 1		CSIMK10 Note 1		CSIPR010 Note 1, CSIPR110 Note 1		√	–	–	–
INTIIC10 Note 1	IICIF10 Note 1		IICMK10 Note 1		IICPR010 Note 1, IICPR110 Note 1		√	–	–	–
INTSR1 Note 2	SRIF1 Note 2		SRMK1 Note 2		SRPR01 Note 2, SRPR11 Note 2		√	√	√	–
INTCSI11 Note 2	CSIIF11 Note 2		CSIMK11 Note 2		CSIPR011 Note 2, CSIPR111 Note 2		√	√	√	–
INTIIC11 Note 2	IICIF11 Note 2		IICMK11 Note 2		IICPR011 Note 2, IICPR111 Note 2		√	√	√	–
INTSRE1 Note 3	SREIF1 Note 3		SREMK1 Note 3		SREPR01 Note 3, SREPR11 Note 3		√	√	√	–
INTTM03H Note 3	TMIF03H Note 3		TMMK03H Note 3		TMPR003H Note 3, 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		√	√	√	√

Notes 1. Do not use UART1, CSI10, and IIC10 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTSR1, INTCSI10, and INTIIC10 is generated, bit 0 of the IF1L register is set to 1. Bit 0 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

2. Do not use UART1, CSI11, and IIC11 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTSR1, INTCSI11, and INTIIC11 is generated, bit 1 of the IF1L register is set to 1. Bit 1 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

3. Do not use UART1 and channel 3 of TAU0 (at 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTSRE1 and INTTM03H is generated, bit 2 of the IF1L register is set to 1. Bit 2 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Table 18-2. Flags Corresponding to Interrupt Request Sources (4/5)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	64-pin	48-pin	30, 32-pin	20-pin
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	√	√	√	√
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		√	√	√	√
INTIT	ITIF		ITMK		ITPR0, ITPR1		√	√	√	√
INTKR	KRIF		KRMK		KRPR0, KRPR1		√	√	–	–
INTCSIS0 Note 1	CSIIFS0 Note 1		CSIMKS0 Note 1		CSIPR0S0 Note 1, CSIPR1S0 Note1		√	√	√	√
INTSTS0 Note 1	STIFS0 Note 1		STMKS0 Note 1		STPR0S0 Note 1, STPR1S0 Note 1		√	√	√	√
INTSRS0 Note 2	SRIFS0 Note 2		SRMKS0 Note 2		SRPR0S0 Note 2, SRPR1S0 Note 2		√	√	√	√
INTCSIS1 Note 2	CSIIFS1 Note 2		CSIMKS1 Note 2		CSIPR0S1 Note 2, CSIPR1S1 Note 2		√	–	–	–
INTWUTM	WUTMIF		WUTMMK		WUTMPR0, WUTMPR1		√	√	√	√
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104		√	√	√	√

Notes 1. Do not use UARTS0 and CSIS0 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTSTS0 and INTCSIS0 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.

2. Do not use UARTS0 and CSIS1 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTSRS0 and INTCSIS1 is generated, bit 5 of the IF1H register is set to 1. Bit 5 of the MK1H, PR01H, and PR11H registers supports these two interrupt sources.

Table 18-2. Flags Corresponding to Interrupt Request Sources (5/5)

	Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	64-pin	48-pin	30, 32-pin	20-pin
	INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L	√	√	√	√
	INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		√	√	√	√
	INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		√	√	√	√
	INTP6	PIF6		PMK6		PPR06, PPR16		√	√	–	–
	INTP7 Note 1	PIF7 Note 1		PMK7 Note 1		PPR07 Note 1, PPR17 Note 1		√	–	–	–
<R>	INTLT Note 1	LTIF0 Note 1		LTMK0 Note 1		LTPR00 Note 1, LTPR10 Note 1		√	√	√	√
	INTP8 Note 2	PIF8 Note 2		PMK8 Note 2		PPR08 Note 2, PPR18 Note 2		√	√	–	–
<R>	INTLR Note 2	LRIF0 Note 2		LRMK0 Note 2		LRPR00 Note 2, LRPR10 Note 2		√	√	√	√
	INTP9 Note 3	PIF9 Note 3		PMK9 Note 3		PPR09 Note 3, PPR19 Note 3		√	√	–	–
<R>	INTLS Note 3	LSIF0 Note 3		LSMK0 Note 3		LSPR00 Note 3, LSPR10 Note 3		√	√	√	√
	INTP10 Note 4	PIF10 Note 4		PMK10 Note 4		PPR010 Note 4, PPR110 Note 4		√	–	–	–
	INTSRES0 Note 4	SREIFS0 Note 4		SREMKS0 Note 4		SREPR0S0 Note 4, SREPR1S0 Note 4		√	√	√	√
	INTP11	PIF11	IF2H	PMK11	MK2H	PPR011, PPR111	PR02H, PR12H	√	–	–	–
	INTMD	MDIF		MDMK		MDPR0, MDPR1		√	√	√	√
	INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√	√	√

Notes 1. Do not use LIN-UART0 and INTP7 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTLS0 and INTP7 is generated, bit 4 of the IF2L register is set to 1. Bit 4 of the MK2L, PR02L, and PR12L registers supports these two interrupt sources.

2. Do not use LIN-UART0 and INTP8 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTLR0 and INTP8 is generated, bit 5 of the IF2L register is set to 1. Bit 5 of the MK2L, PR02L, and PR12L registers supports these two interrupt sources.

3. Do not use LIN-UART0 and INTP9 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTLS0 and INTP9 is generated, bit 6 of the IF2L register is set to 1. Bit 6 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

4. Do not use LIN-UART0 and INTP10 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTLS0 and INTP10 is generated, bit 7 of the IF2L register is set to 1. Bit 7 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

(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 the 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 18-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (64-pin) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0 TMIF01H	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	DMAIF1	DMAIF0	SREIF2	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20

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 CSIIF11 IICIF11	STIF1 IICIF10 CSIIF10

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	WUTMIF	SRIFS0 CSIIFS1	STIFS0 CSIIFS0	KRIF	ITIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
<R> IF2L	PIF10 SREIFS0	PIF9 LSIF0	PIF8 LRIF0	PIF7 LTIF0	PIF6	TMIF07	TMIF06	TMIF05

Figure 18-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (64-pin) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
IF2H	FLIF	0	MDIF	0	0	0	0	PIF11

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. The above is the bit layout for the 64-pin. The available bits differ depending on the product. For details about the bits available for each product, see Table 18-2. Be sure to clear bits that are not available to 0.

2. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

3. 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.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

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 18-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (64-pin)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0 TMMK01H	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	DMAMK1	DMAMK0	SREMK2	SRMK2 CSIMK21 IICMK21	STMK2 CSIMK20 IICMK20

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 CSIMK11 IICMK11	STMK1 CSIMK10 IICMK10

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
MK1H	TMMK04	WUTMMK	SRMKS0 CSIMKS1	STMKS0 CSIMKS0	KRMK	ITMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
<R> MK2L	SREMKS0 PMK10	PMK9 LSMK0	PMK8 LRMK0	LTMK0 PMK7	PMK6	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	6	<5>	4	3	2	1	0
MK2H	FLMK	1	MDMK	1	1	1	1	PMK11

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution The above is the bit layout for the 64-pin. The available bits differ depending on the product. For details about the bits available for each product, see Table 18-2. Be sure to set bits that are not available to 1.

(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 18-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (64-pin) (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00 TMPR001H	SRPR00 CSIPR001 IICPR001	STPR00 CSIPR000 IICPR000	DMAPR01	DMAPR00	SREPR02	SRPR02 CSIPR021 IICPR021	STPR02 CSIPR020 IICPR020

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10 TMPR101H	SRPR10 CSIPR101 IICPR101	STPR10 CSIPR100 IICPR100	DMAPR11	DMAPR10	SREPR12	SRPR12 CSIPR121 IICPR121	STPR12 CSIPR120 IICPR120

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 CSIPR011 IICPR011	STPR01 CSIPR010 IICPR010

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 CSIPR111 IICPR111	STPR11 CSIPR110 IICPR110

Figure 18-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (64-pin) (2/2)

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	TMPR004	WUTMPR0	SRPR0S0 CSIPR0S1	STPR0S0 CSIPR0S0	KRPR0	ITPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	TMPR104	WUTMPR1	SRPR1S0 CSIPR1S1	STPR1S0 CSIPR1S0	KRPR1	ITPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
<R> PR02L	SREPR0S0 PPR010	PPR09 LSPR00	PPR08 LRPR00	LTPR00 PPR07	PPR06	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
<R> PR12L	SREPR1S0 PPR110	PPR19 LSPR10	PPR18 LRPR10	LTPR10 PPR17	PPR16	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	6	<5>	4	3	2	1	<0>
PR02H	FLPR0	1	MDPR0	1	1	1	1	PPR011

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	6	<5>	4	3	2	1	<0>
PR12H	FLPR1	1	MDPR1	1	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 above is the bit layout for the 64-pin. The available bits differ depending on the product. For details about the bits available for each product, see Table 18-2. Be sure to set bits that are not available to 1.

(4) External interrupt rising edge enable registers (EGP0, EGP1)

These registers specify the valid edge for INTP0 to INTP11.

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 18-5. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1) (64-pin)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8

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 18-3 shows the ports corresponding to the EGPn and EGNn bits.

Table 18-3. Ports Corresponding to EGPn and EGNn bits

Detection Enable Bit		Edge Detection Port	Interrupt Request Signal	64-pin	48-pin	30, 32-pin	20-pin
EGP0	EGN0	P137	INTP0	√	√	√	√
EGP1	EGN1	P50	INTP1	√	√	√	√
EGP2	EGN2	P51	INTP2	√	√	√	√
EGP3	EGN3	P30	INTP3	√	√	√	–
EGP4	EGN4	P31	INTP4	√	√	√	√
EGP5	EGN5	P16	INTP5	√	√	√	√
EGP6	EGN6	P140	INTP6	√	√	–	–
EGP7	EGN7	P141	INTP7	√	–	–	–
EGP8	EGN8	P74	INTP8	√	√	–	–
EGP9	EGN9	P75	INTP9	√	√	–	–
EGP10	EGN10	P76	INTP10	√	–	–	–
EGP11	EGN11	P77	INTP11	√	–	–	–

Caution Select the port mode by clearing the EGPn and EGNn bits to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark n = 0 to 11

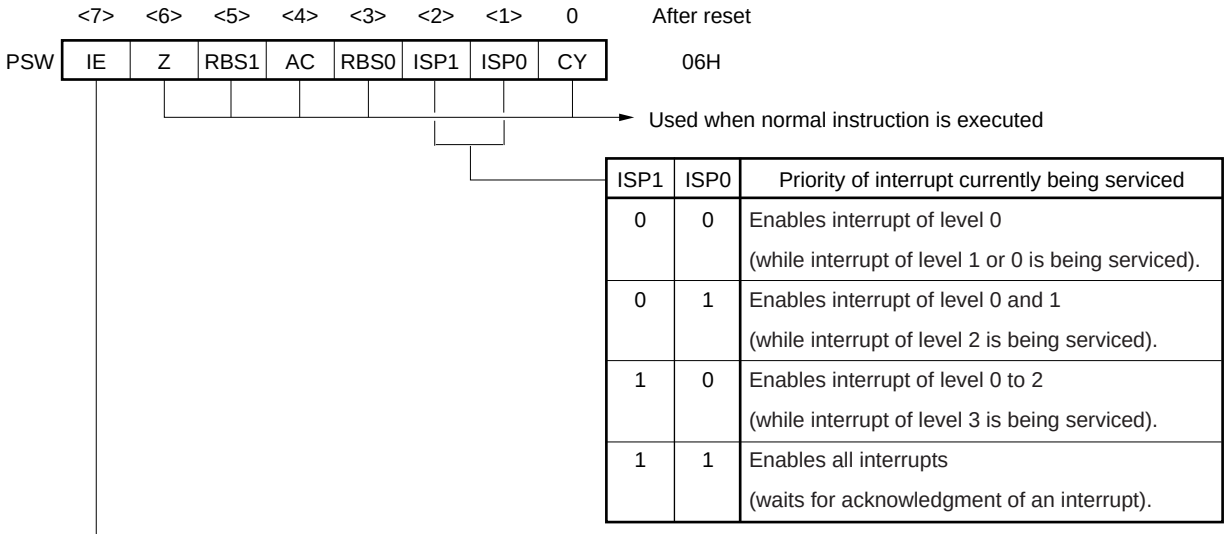
(5) Program status word (PSW)

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. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are 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 18-6. 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

18.4 Interrupt Servicing Operations

18.4.1 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 18-4 below.

For the interrupt request acknowledgment timing, see **Figures 18-8** and **18-9**.

Table 18-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	9 clocks	16 clocks

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: 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.

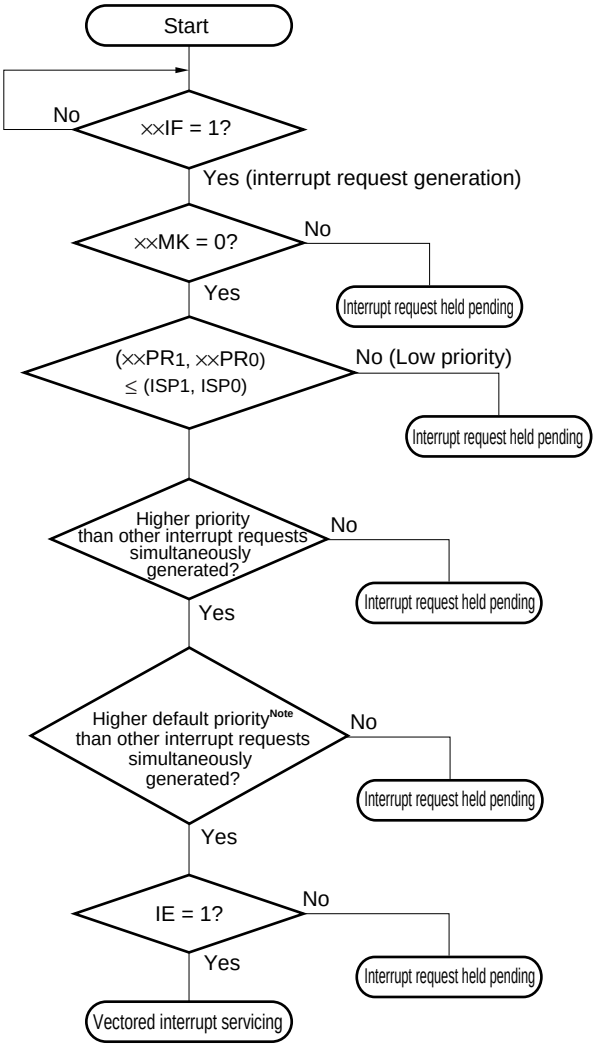
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 18-7 shows the interrupt request acknowledgment 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.

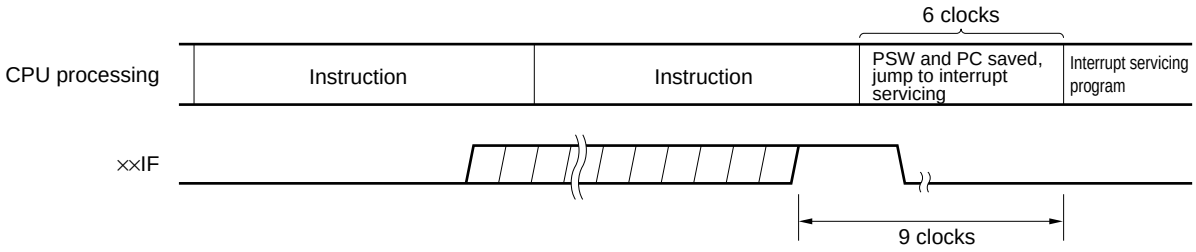
Figure 18-7. Interrupt Request Acknowledgment Processing Algorithm



××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR0: Priority specification flag 0
××PR1: 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 18-6**)

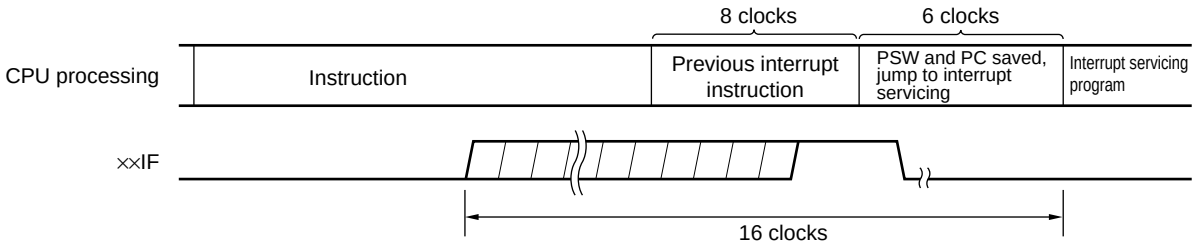
Note For the default priority, refer to **Table 18-1 Interrupt Source List**.

Figure 18-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 18-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

18.4.2 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 Do not use the RETI instruction for restoring from the software interrupt.

18.4.3 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 lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. 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 18-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 18-10 shows multiple interrupt servicing examples.

Table 18-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0, ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0, ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 1, ISP0 = 0	○	×	○	×	○	×	×	×	○
	ISP1 = 1, ISP0 = 1	○	×	○	×	○	×	○	×	○
Software interrupt		○	×	○	×	○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

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.

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

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)

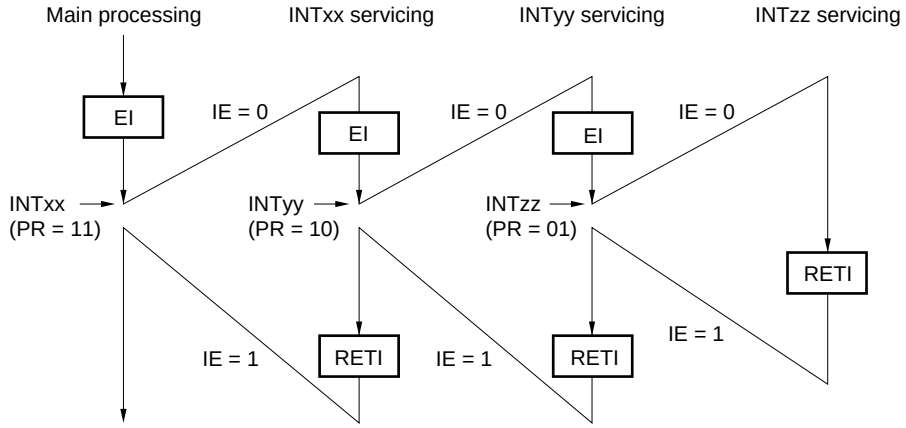
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

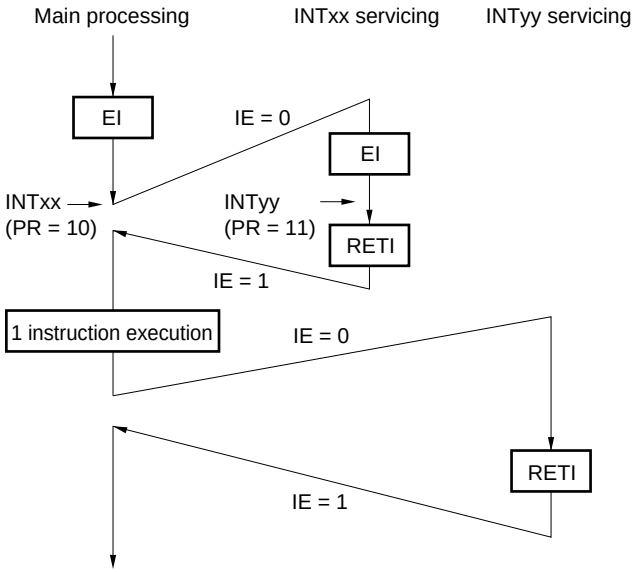
Figure 18-10. Examples of Multiple Interrupt Servicing (1/2)

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

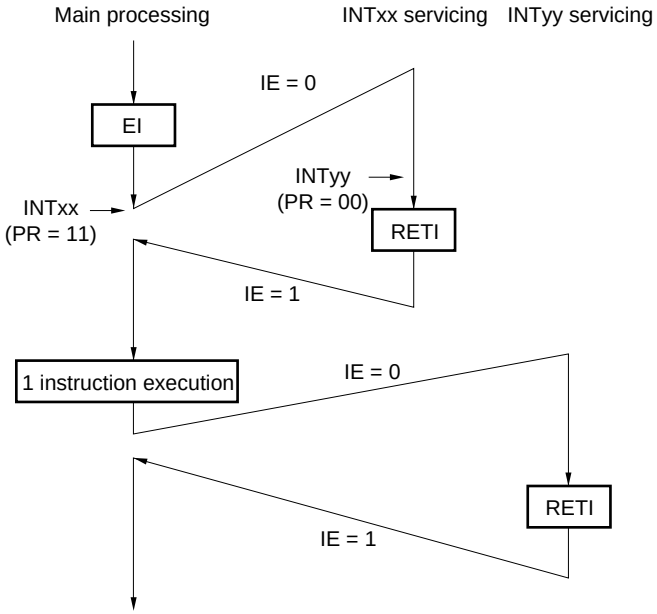


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××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)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 18-10. 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 ××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)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

18.4.4 Interrupt request hold

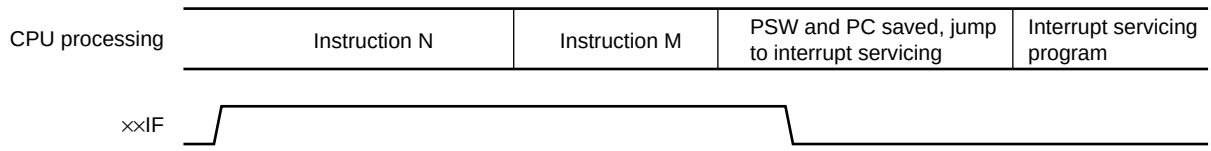
There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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
- Manipulation 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

Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.

Figure 18-11 shows the timing at which interrupt requests are held pending.

Figure 18-11. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 19 KEY INTERRUPT FUNCTION

The number of key interrupt input channels differs, depending on the product.

	20, 30, 32-pin	48-pin	64-pin
Key interrupt input channels	–	6 ch	8 ch

19.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by setting the key return mode register (KRM) and inputting a falling edge to the key interrupt input pins (KR0 to KR7).

Table 19-1. Assignment of Key Interrupt Detection Pins

Flag	Description
KRM0	Controls KR0 signal in 1-bit units.
KRM1	Controls KR1 signal in 1-bit units.
KRM2	Controls KR2 signal in 1-bit units.
KRM3	Controls KR3 signal in 1-bit units.
KRM4	Controls KR4 signal in 1-bit units.
KRM5	Controls KR5 signal in 1-bit units.
KRM6	Controls KR6 signal in 1-bit units.
KRM7	Controls KR7 signal in 1-bit units.

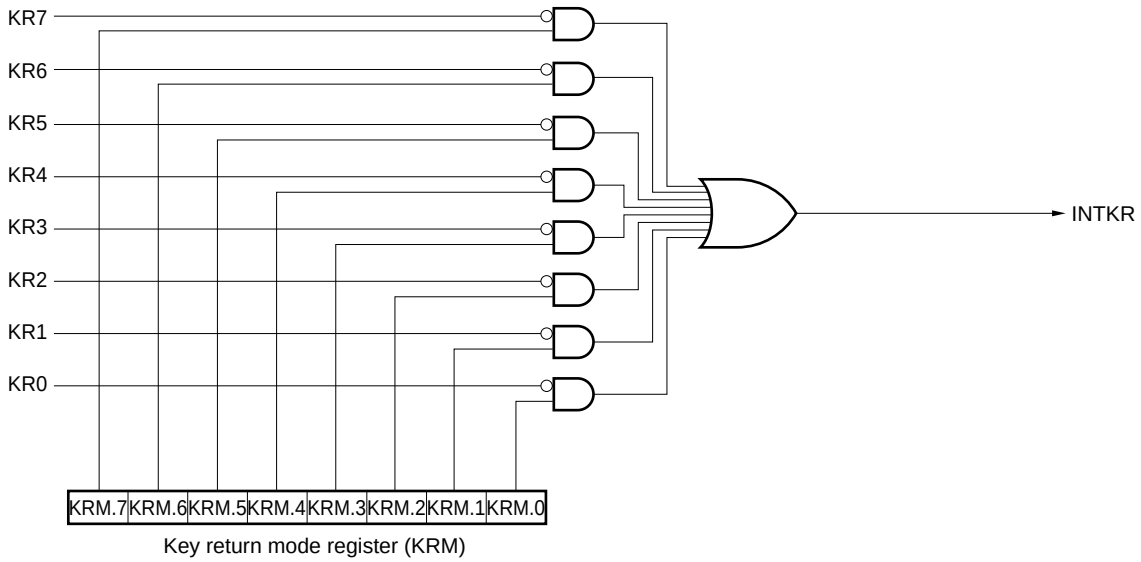
19.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 19-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM)

Figure 19-1. Block Diagram of Key Interrupt



19.3 Register Controlling Key Interrupt

(1) Key return mode register (KRM)

This register controls the KRM.0 to KRM.7 bits using the KR0 to KR7 signals, respectively.

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

Reset signal generation clears this register to 00H.

Figure 19-2. Format of Key Return Mode Register (KRM)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	KRM.7	KRM.6	KRM.5	KRM.4	KRM.3	KRM.2	KRM.1	KRM.0

KRM.n	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Cautions 1. If any of the KRM.0 to KRM.7 bits used is set to 1, set bits 0 to 7 (PU7.0 to PU7.7) of the corresponding pull-up resistor register 7 (PU7) to 1.

2. An interrupt will be generated if the target bit of the KRM register is set while a low level is being input to the key interrupt input pin. To ignore this interrupt, set the KRM register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input low-level width (250 ns or more).

3. The bits not used in the key interrupt mode can be used as normal ports.

Remarks 1. n = 0 to 7

2. KR0 to KR5: 48-pin products

KR0 to KR7: 64-pin products

CHAPTER 20 STANDBY FUNCTION

20.1 Standby Function and Configuration

20.1.1 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 CSI00 or UART0 data reception and an A/D conversion request by the timer trigger signal (the interrupt request signal (INTRTC/INTIT)), the STOP mode is exited, the CSI00 or UART0 data is received without operating the CPU, and A/D conversion is performed. This can only be specified when the high-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set 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.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.

When using CSI00, UART0, or the A/D converter in the SNOOZE mode, set up serial standby control register 0 (SSC0) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 13.3 Registers Controlling Serial Array Unit and 12.3 Registers Used in A/D Converter.

Cautions 3. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

4. It can be selected by the option byte whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 26 OPTION BYTE.

20.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For the registers that start, stop, or select the clock, see **CHAPTER 5 CLOCK GENERATOR**.

(1) Oscillation stabilization time counter status register (OSTC)

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

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

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.
When reset is released (reset by $\overline{\text{RESET}}$ input, POR, LVD, WDT, and executing an illegal instruction), the STOP instruction and MSTOP bit (bit 7 of clock operation status control register (CSC)) = 1 clear this register to 00H.

Figure 20-1. 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	MOST 8	MOST 9	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		
									f_X = 10 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	$2^8/f_X$ max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	$2^8/f_X$ min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	$2^9/f_X$ min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	$2^{10}/f_X$ min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	$2^{11}/f_X$ min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	$2^{13}/f_X$ min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	$2^{15}/f_X$ min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_X$ min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_X$ min.	26.21 ms min.	13.11 ms min.

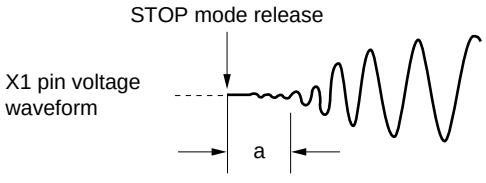
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS). If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC register oscillation stabilization time ≤ Oscillation stabilization time set by OSTS register**

Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after STOP mode is released.

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



Remark f_X: X1 clock oscillation frequency

(2) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released.

When the X1 clock is selected as the CPU clock, the operation waits for the time set using the OSTS register after the STOP mode is released.

When the high-speed on-chip oscillator clock is selected as the CPU clock, confirm with the oscillation stabilization time counter status register (OSTC) that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using the OSTC register.

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

Reset signal generation sets this register to 07H.

Figure 20-2. 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	OSTS.2	OSTS.1	OSTS.0

OSTS.2	OSTS.1	OSTS.0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	Setting prohibited
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_X$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_X$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.21 ms	13.11 ms

Cautions 1. To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.

2. Setting the oscillation stabilization time to 20 μs or less is prohibited.

3. Before changing the setting of the OSTS register, confirm that the count operation of the OSTC register is completed.

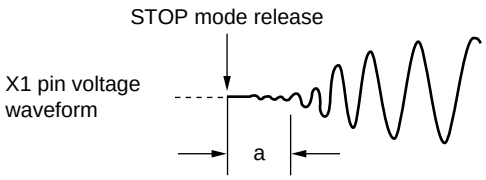
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

5. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register. If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC register oscillation stabilization time ≤ Oscillation stabilization time set by OSTS register**

Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after STOP mode is released.

6. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark f_X: X1 clock oscillation frequency

20.2 Standby Function Operation

20.2.1 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.

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}			
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 1 and WDTON = 0: Stops • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory			Operation stopped		
Data flash memory					
RAM					
Port (latch)			Status before HALT mode was set is retained		
Timer array unit			Operable		
Real-time clock (RTC)					
Interval timer					
Wakeup timer					
Watchdog timer			See CHAPTER 11 WATCHDOG TIMER		
Clock output/buzzer output			Operable		
A/D converter					
Serial array unit (SAU)					
LIN-UART					
Serial interface (IICA)					
Multiplier and divider/multiply-accumulator					
DMA controller					
Power-on-reset function					
Voltage detection function					
External interrupt					
Key interrupt function					
CRC operation function	High-speed CRC				
	General-purpose CRC				
Illegal-memory access detection 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 20-1. 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)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 1 and WDTON = 0: Stops • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 0: Stops	
CPU			Operation stopped	
Code flash memory				
Data flash memory				
RAM				
Port (latch)			Status before HALT mode was set is retained	
Timer array unit			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))	
Real-time clock (RTC)			Operable	
Interval timer				
Wakeup timer			Operable (depends on the operation clock status)	
Watchdog timer			See CHAPTER 11 WATCHDOG TIMER	
Clock output/buzzer output			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))	
A/D converter			Operation disabled	
Serial array unit (SAU)			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))	
LIN-UART			Operable	
Serial interface (IICA)			Operation disabled	
Multiplier and divider/multiply-accumulator			Operable (Operation is disabled while in the low consumption RTC mode (when the RTCLPC bit of the OSMC register is 1))	
DMA controller				
Power-on-reset function			Operable	
Voltage detection function				
External interrupt				
Key interrupt function				
CRC operation function	High-speed CRC		Operation disabled	
	General-purpose CRC		Operable	
Illegal-memory access detection function			Operation stopped	

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

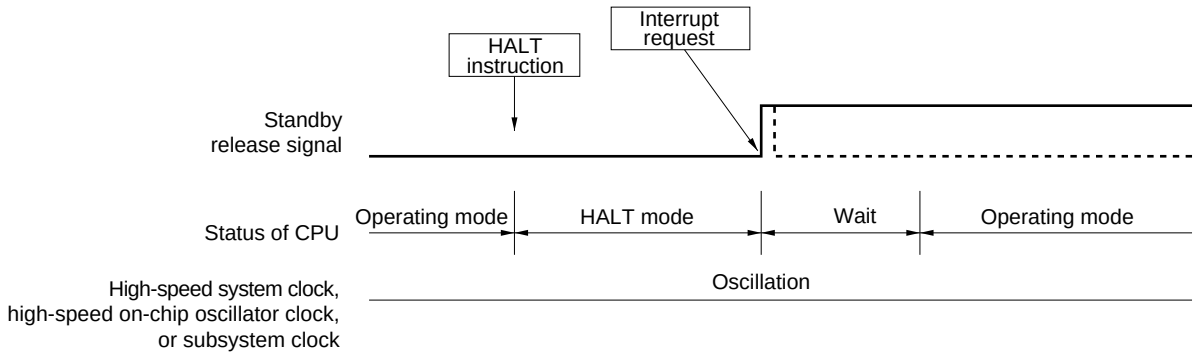
(2) HALT mode release

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 20-3. HALT Mode Release by Interrupt Request Generation



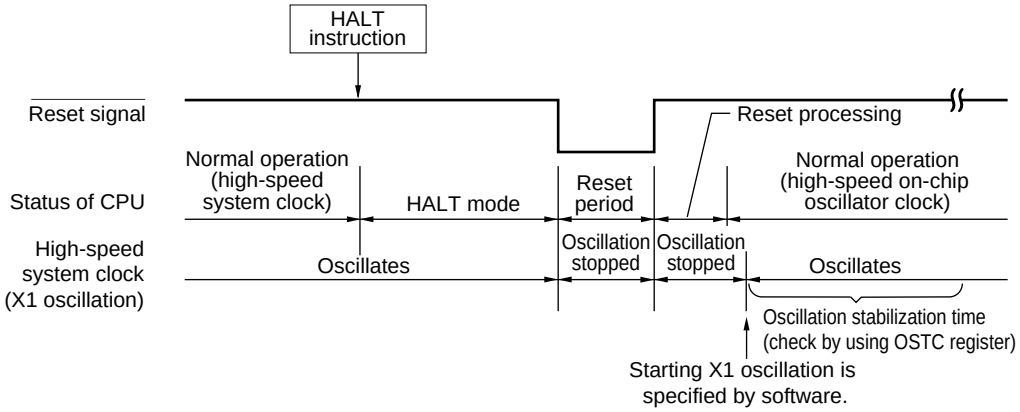
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

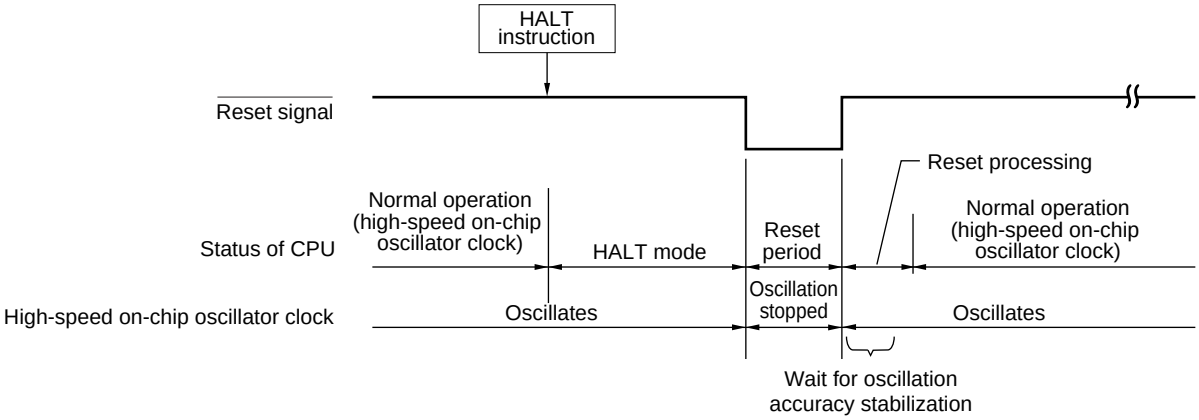
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 20-4. HALT Mode Release by Reset

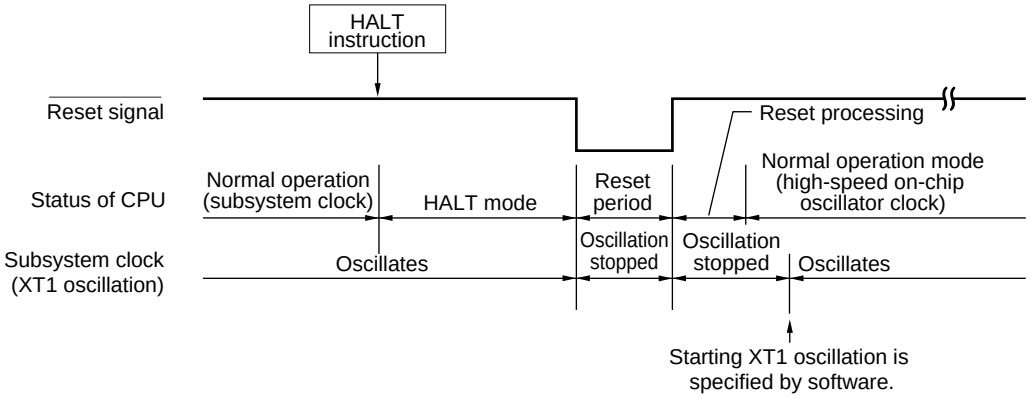
(1) When high-speed system clock is used as CPU clock



When high-speed on-chip oscillator clock is used as CPU clock



(3) When subsystem clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

20.2.2 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.

Cautions 1. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

2. When using CSI00, UART0, or the A/D converter in the SNOOZE mode, set up serial standby control register 0 (SSC0) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 13.3 Registers Controlling Serial Array Unit and 12.3 Registers Used in A/D Converter.

The operating statuses in the STOP mode are shown below.

Table 20-2. Operating Statuses in STOP Mode

Item \ STOP Mode Setting		When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
		When CPU Is Operating on High-Speed On-Chip Oscillator Clock (f_{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 operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 1 and WDTON = 0: Stops • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 0: Stops		
CPU		Operation stopped		
Code flash memory				
Data flash memory		Operation stopped (Executing the STOP instruction is disabled during data flash programming)		
RAM		Operation stopped		
Port (latch)		Status before STOP mode was set is retained		
Timer array unit		Operation disabled		
Real-time clock (RTC)		Operable		
Interval timer				
Wakeup timer		Operable (depends on the operation clock status)		
Watchdog timer		See CHAPTER 11 WATCHDOG TIMER		
Clock output/buzzer output		Operable only when subsystem clock is selected as the count clock		
A/D converter		Wakeup operation is enabled (switching to the SNOOZE mode)		
Serial array unit (SAU)		Wakeup operation is enabled only for CSI00 and UART0 (switching to the SNOOZE mode) Operation is disabled for anything other than CSI00 and UART0		
LIN-UART		Operation disabled		
Serial interface (IICA)		Wakeup by address match operable		
Multiplier and divider/multiply-accumulator		Operation disabled		
DMA controller				
Power-on-reset function		Operable		
Voltage detection function				
External interrupt				
Key interrupt function				
CRC operation function	High-speed CRC	Operation stopped		
	General-purpose CRC			
Illegal-memory access detection 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

Cautions 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

2. To stop the low-speed on-chip oscillator clock in the STOP mode, use an option byte to stop the watchdog timer operation in the HALT/STOP mode (bit 0 (WDSTBYON) of 000C0H = 0), and then execute the STOP instruction.

3. 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. Before changing the CPU clock from the high-speed on-chip oscillator clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

(2) STOP mode release

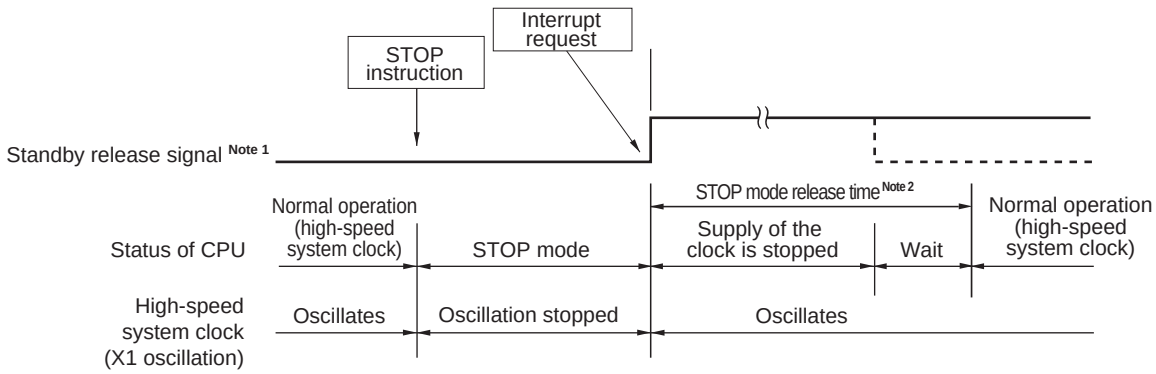
The STOP mode can be released by the following two sources.

(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 20-5. STOP Mode Release by Interrupt Request Generation (1/2)

<R> **(1) When high-speed system clock (X1 oscillation) is used as CPU clock**



Notes 1. For details of the standby release signal, see **Figure 18-1**.

2. STOP mode release time

Supply of the clock is stopped: 18 μs to whichever is longer 65 μs and the oscillation stabilization time (set by OSTS)

Wait

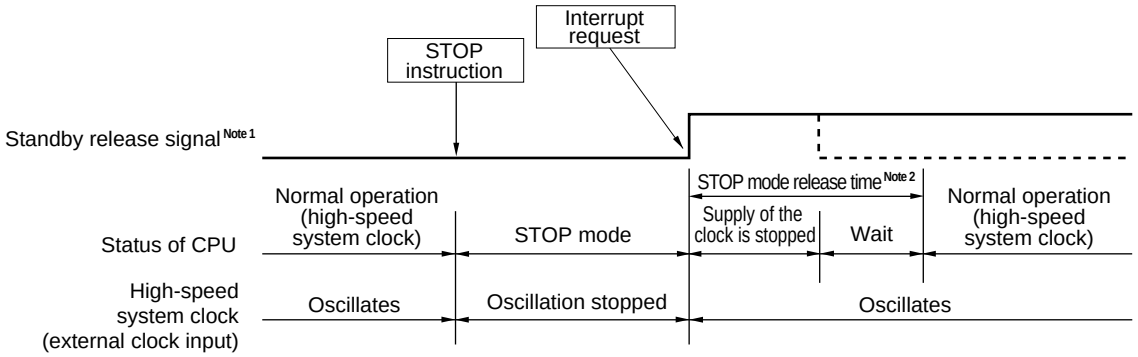
- When vectored interrupt servicing is carried out: 10 to 11 clocks
- When vectored interrupt servicing is not carried out: 4 to 5 clocks

Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

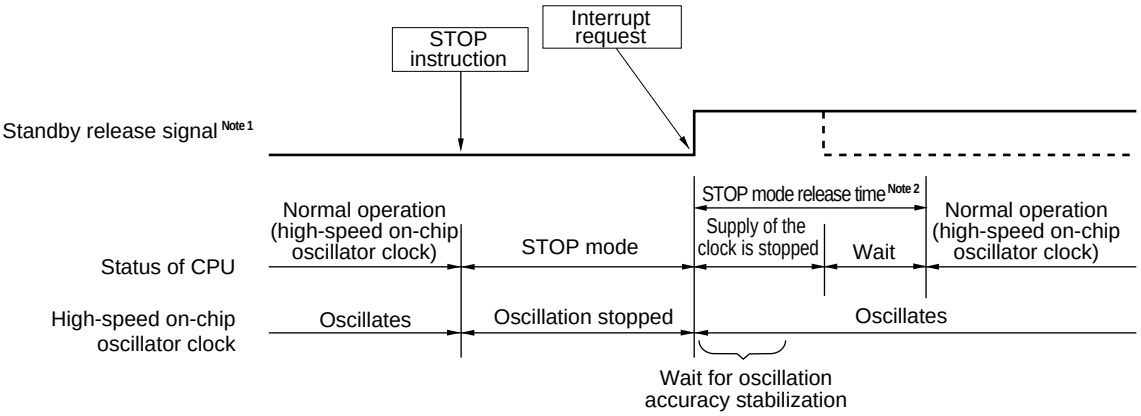
2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 20-5. STOP Mode Release by Interrupt Request Generation (2/2)

<R> **(2) When high-speed system clock (external clock input) is used as CPU clock**



<R> **(3) When high-speed on-chip oscillator clock is used as CPU clock**



Notes 1. For details of the standby release signal, see **Figure 18-1**.

2. STOP mode release time

Supply of the clock is stopped: 18 μs to 65 μs

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

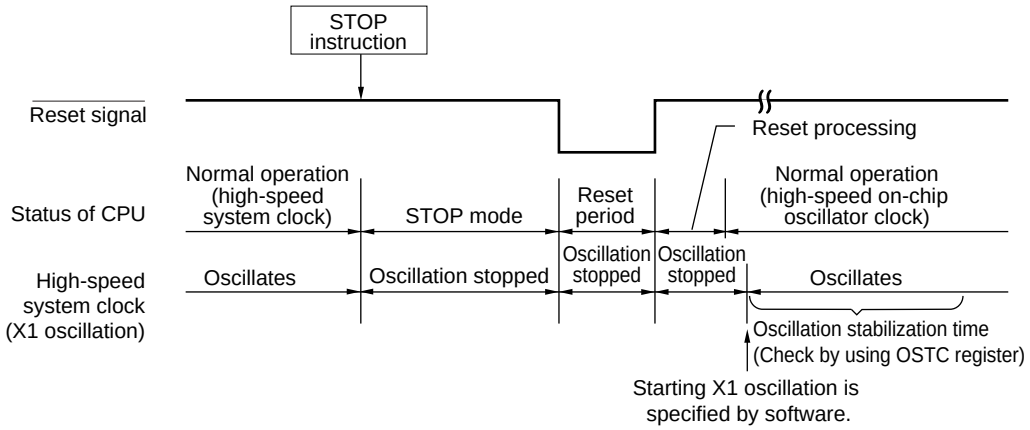
2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

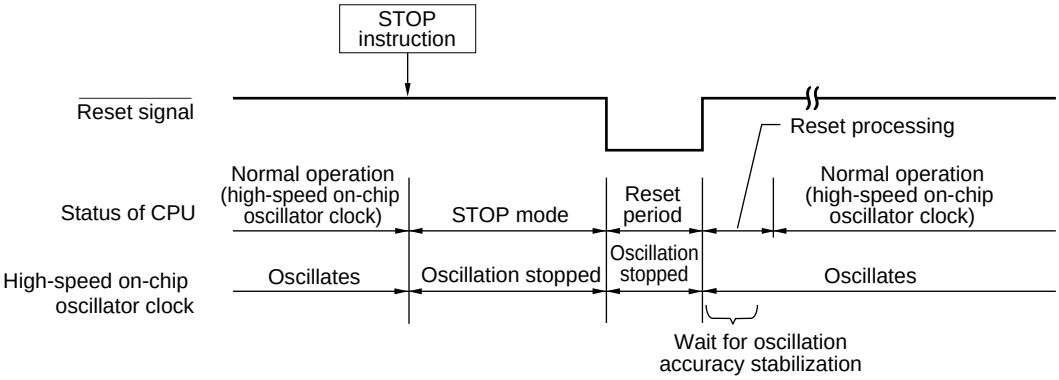
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 20-6. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When high-speed on-chip oscillator clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

20.2.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for CSI00, UART0, or the A/D converter. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.

When using CSI00 or UART0 in the SNOOZE mode, set up serial standby control register 0 (SSC0) before switching to the STOP mode. For details, see **13.3 Registers Controlling Serial Array Unit**.

When using the A/D converter in the SNOOZE mode, set up A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see **12.3 Registers Used in A/D Converter**.

<R> The transition time of going into and getting out from SNOOZE mode is as following.

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:
  4.99 to 9.44 μs + 7 clocks
- When vectored interrupt servicing is not carried out:
  4.99 to 9.44 μs + 1 clock

The operating statuses in the SNOOZE mode are shown below.

Table 20-3. Operating Statuses in SNOOZE Mode

Item \ STOP Mode Setting			When Inputting CSI00/UART0 Data Reception Signal or A/D Converter Timer Trigger Signal While in STOP Mode
			When CPU Is Operating on High-Speed On-Chip Oscillator Clock (f_{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 operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 1 and WDTON = 0: Stops • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 1, WDTON = 1, and WDSTBYON = 0: Stops
CPU			Operation stopped
Code flash memory			
Data flash memory			
RAM			
Port (latch)			Use of the status while in the STOP mode continues
Timer array unit			Operation disabled
Real-time clock (RTC)			Operable
Interval timer			
Wakeup timer			Operable (depends on the operation clock status)
Watchdog timer			See CHAPTER 11 WATCHDOG TIMER
Clock output/buzzer output			Operable only when subsystem clock is selected as the count clock
A/D converter			Operable
Serial array unit (SAU)			Operable only CSI00 and UART0 only. Operation disabled other than CSI00 and UART0.
LIN-UART			Operation disabled
Serial interface (IICA)			
Multiplier and divider/multiply-accumulator			
DMA controller			
Power-on-reset function			Operable
Voltage detection function			
External interrupt			
Key interrupt function			
CRC operation function			Operation disabled
Illegal-memory access detection 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

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 Tables 21-1.

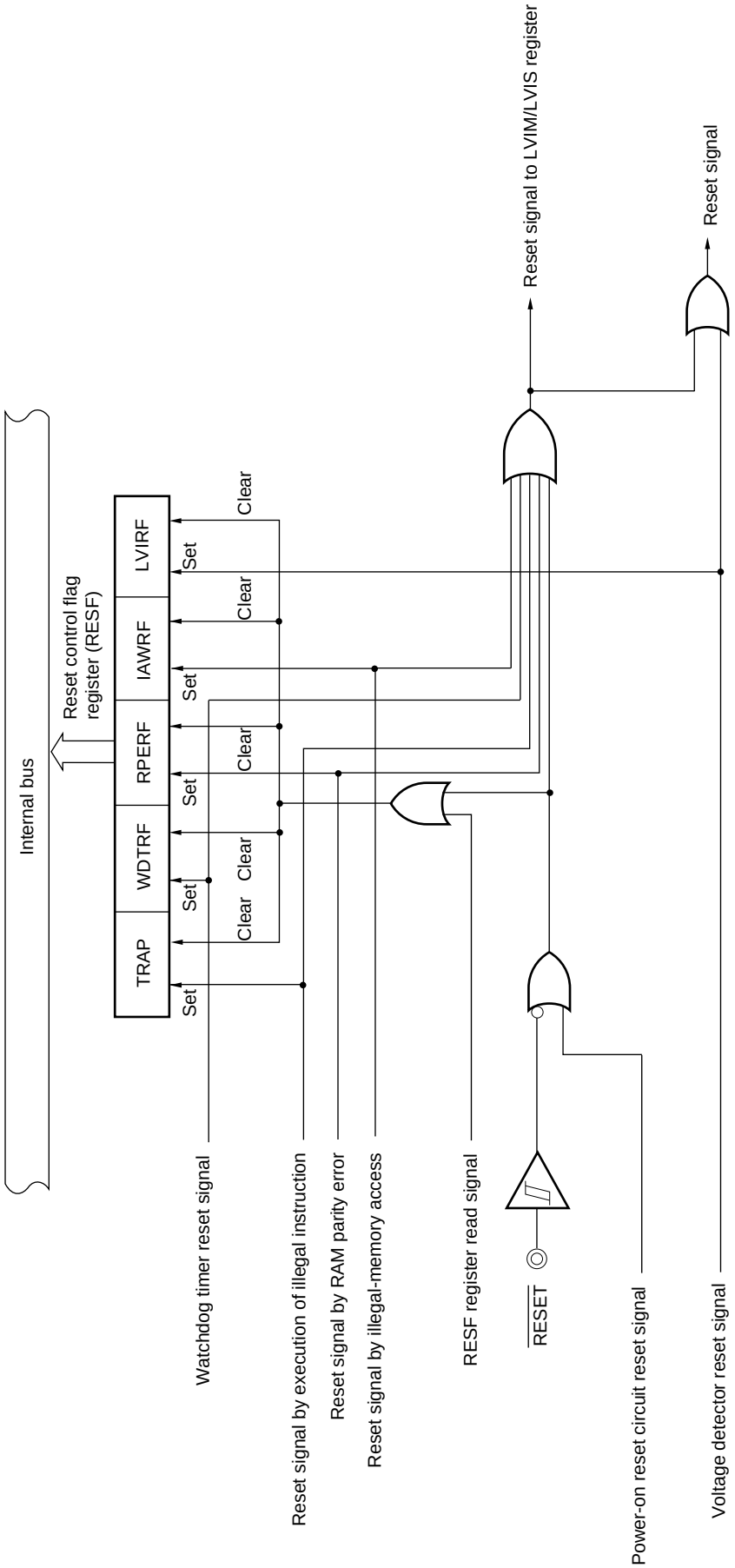
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the high-speed on-chip oscillator clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the high-speed on-chip oscillator clock (see **Figures 21-2** to **21-4**) after reset processing. Reset by POR and LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the high-speed on-chip oscillator clock (see **CHAPTER 22 POWER-ON-RESET CIRCUIT** and **CHAPTER 23 VOLTAGE DETECTOR**) after reset processing.

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.

Cautions 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, a low level of at least 10 μs must be continued during the period in which the supply voltage is within the operating range ($V_{DD} \geq 1.8$ V).)
2. During reset input, the X1 clock, XT1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input and external subsystem clock input become invalid.
3. When reset is effected, port pin P130 is set to low-level output and other port pins become high-impedance, because each SFR and 2nd SFR are initialized.

Remark V_{POR}: POR power supply rise detection voltage

Figure 21-1. Block Diagram of Reset Function



Caution An LVD circuit internal reset does not reset the LVD circuit.

Remarks 1. LVIM: Voltage detection register

2. LVIS: Voltage detection level register

Figure 21-2. Timing of Reset by $\overline{\text{RESET}}$ Input

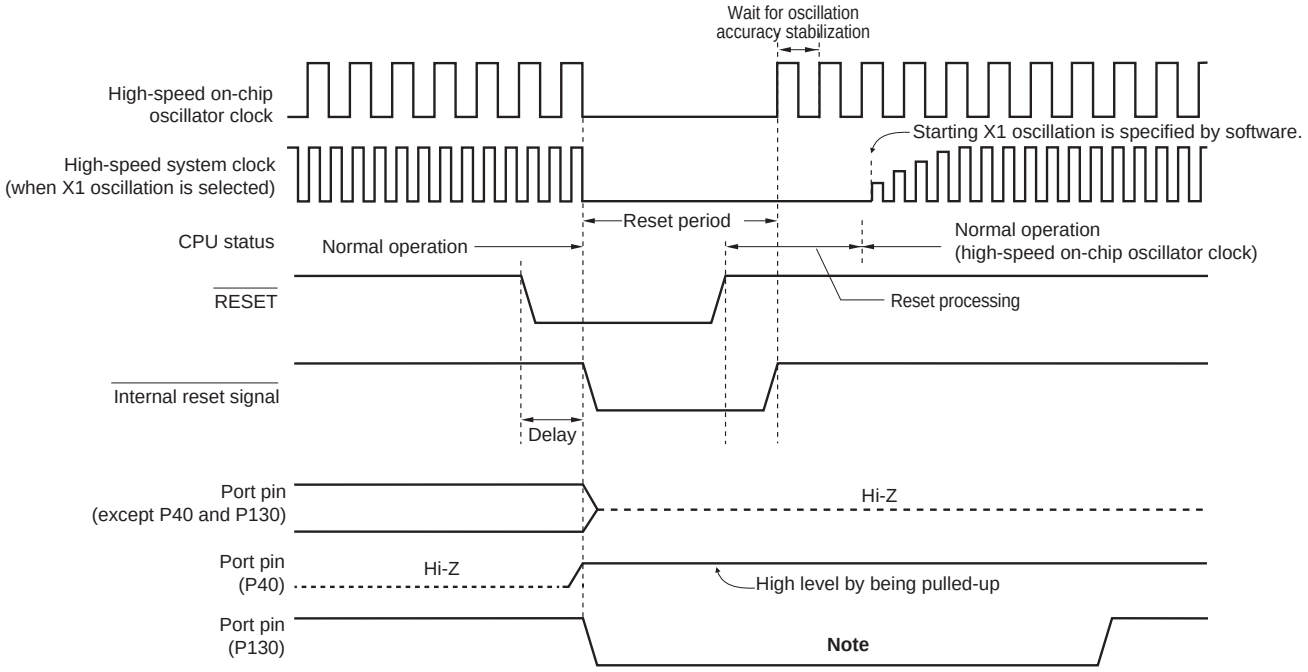
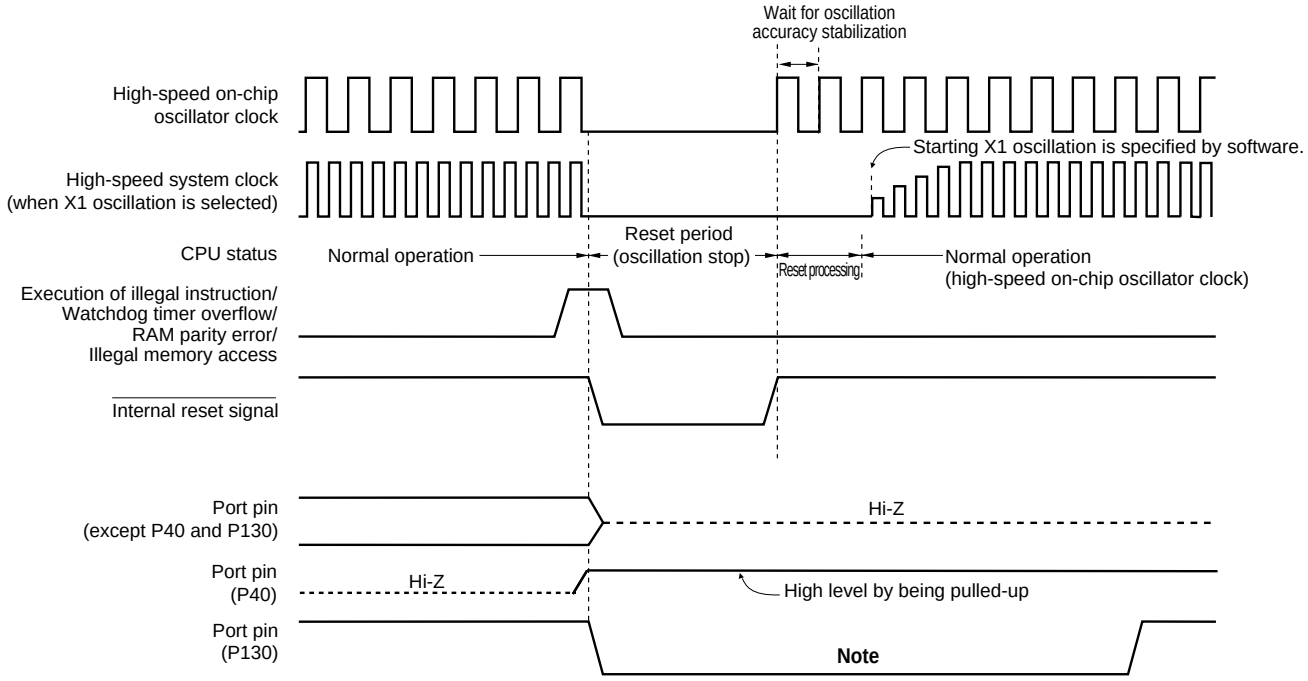


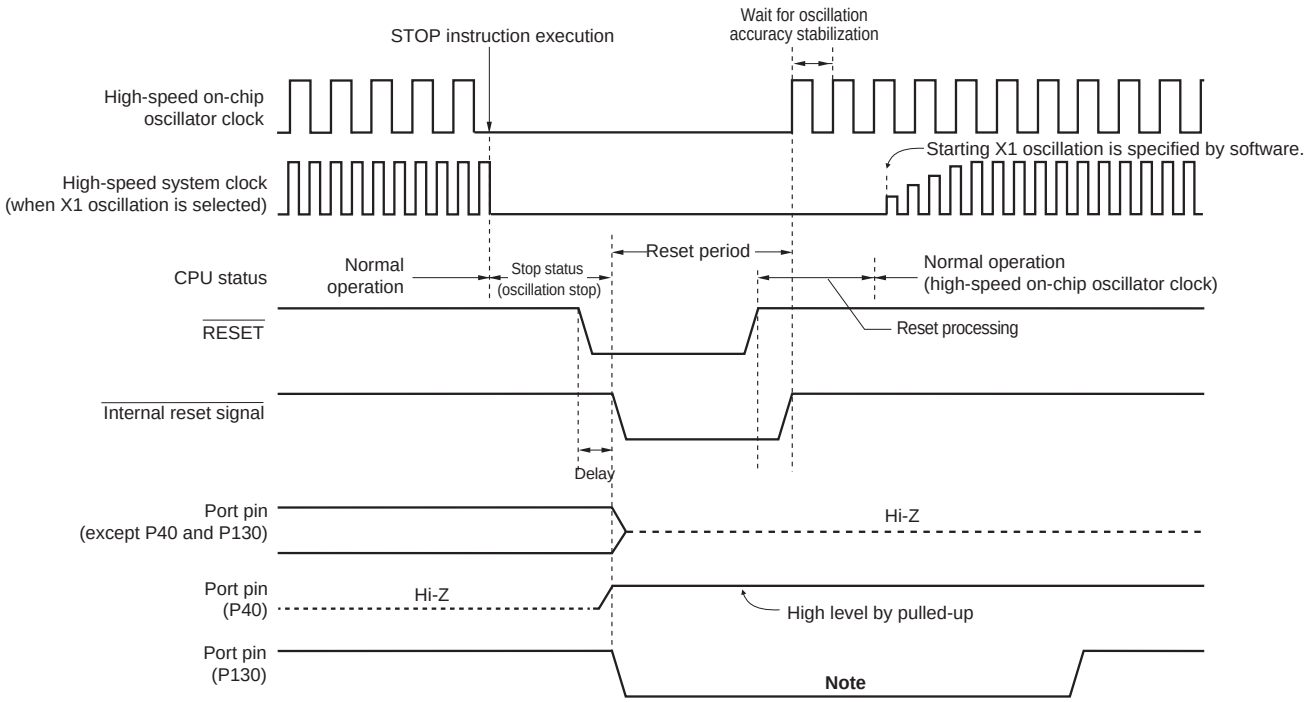
Figure 21-3. Timing of Reset Due to Execution of Illegal Instruction, Watchdog Timer Overflow, RAM Parity Error, or Illegal Memory Access



Note 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.

Caution A watchdog timer internal reset resets the watchdog timer.

Figure 21-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Note 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.

Remark For the reset timing of the power-on-reset circuit and voltage detector, see **CHAPTER 22 POWER-ON-RESET CIRCUIT** and **CHAPTER 23 VOLTAGE DETECTOR**.

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 (The system operates in the LV (low voltage main) mode after reading the option byte)
Data flash memory			Operation stopped
RAM			Operation stopped
Port (latch)			Set P130 to low-level output. The port pins except for P130 become high impedance. P40 is pulled-up (during reset other than the pin reset and the POC reset) or becomes high impedance (during the pin reset or the POC reset).
Timer array unit			Operation stopped
Real-time clock (RTC)			
Interval timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Serial array unit (SAU)			
Serial interface (IICA)			
Multiplier & divider, multiply-accumulator			
DMA controller			
Power-on-reset function			Detection operation possible
Voltage detection function			Operation stopped (LVD detection is possible after reading the option byte)
External interrupt			Operation stopped
Key interrupt function			
CRC operation function	High-speed CRC		
	General-purpose CRC		
Illegal-memory access detection function			

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 (1/5)

Hardware		After Reset Acknowledgment[Note 1]
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
Processor mode control register (PMC)		00H
Port registers (P0 to P7, P12 to P14) (output latches)		00H
Port mode registers	PM0 to PM7, PM12, PM14	FFH
Port mode control registers 0, 12, 14 (PMC0, PMC12, PMC14)		FFH
Port input mode registers 0, 1, 5 (PIM0, PIM1, PIM5)		00H
Port output mode registers 0, 1, 5, 7 (POM0, POM1, POM5, POM7)		00H
Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU14)		00H (PU4 is 01H)
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
System clock control register (CKC)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		07H
Noise filter enable registers 0, 1 (NFEN0, NFEN1)		00H
Peripheral enable register 0 (PER0)		00H
High-speed on-chip oscillator trimming register (HIOTRM)		**Note 2**
Temperature trimming registers 0 to 3 (TEMPCAL0 to TEMPCAL3)		**Note 2**
Operation speed mode control register (OSMC)		00H
Timer array unit	Timer data registers 00 to 07 (TDR00 to TDR07)	0000H
	Timer mode registers 00 to 07 (TMR00 to TMR07)	0000H
	Timer status registers 00 to 07 (TSR00 to TSR07)	0000H
	Timer input select register 0 (TIS0)	00H
	Timer counter registers 00 to 07 (TCR00 to TCR07)	FFFFH
	Timer channel enable status register 0 (TE0)	0000H
	Timer channel start register 0 (TS0)	0000H
	Timer channel stop register 0 (TT0)	0000H
	Timer clock select register 0 (TPS0)	0000H
	Timer output register 0 (TO0)	0000H
	Timer output enable register 0 (TOE0)	0000H
	Timer output level register 0 (TOL0)	0000H
	Timer output mode registers 0 (TOM0)	0000H

Notes 1. 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.

2. The default value differs, depending on the product.

Remark The special function register (SFR) mounted depends on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 21-2. Hardware Statuses After Reset Acknowledgment (2/5)

Hardware		Status After Reset Acknowledgment[Note 1]
Real-time clock	Second count register (SEC)	00H
	Minute count register (MIN)	00H
	Hour count register (HOUR)	12H
	Week count register (WEEK)	00H
	Day count register (DAY)	01H
	Month count register (MONTH)	01H
	Year count register (YEAR)	00H
	Watch error correction register (SUBCUD)	00H
	Alarm minute register (ALARMWM)	00H
	Alarm hour register (ALARMWH)	12H
	Alarm week register ALARMWW)	00H
	Real-time clock control register 0 (RTCC0)	00H
	Real-time clock control register 1 (RTCC1)	00H
Interval timer	Interval timer control register (ITMC)	0FFFH
Clock output/buzzer output controller	Clock output select registers 0, 1 (CKS0, CKS1)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 2]
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	A/D converter mode registers 0 to 2 (ADM0, ADM1, ADM2,)	00H
	Conversion result comparison upper limit setting register (ADUL)	FFH
	Conversion result comparison lower limit setting register (ADLL)	00H
	A/D test register (ADTES)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	00H
Serial array unit (SAU)	Serial data registers 00 to 03, 10, 11 (SDR00 to SDR03, SDR10, SDR11)	0000H
	Serial status registers 00 to 03, 10, 11 (SSR00 to SSR03, SSR10, SSR11)	0000H
	Serial flag clear trigger registers 00 to 03, 10, 11 (SIR00 to SIR03, SIR10, SIR11)	0000H
	Serial mode registers 00 to 03, 10, 11 (SMR00 to SMR03, SMR10, SMR11)	0020H
	Serial communication operation setting registers 00 to 03, 10, 11 (SCR00 to SCR03, SCR10, SCR11)	0087H
	Serial channel enable status registers 0, 1 (SE0, SE1)	0000H
	Serial channel start registers 0, 1 (SS0, SS1)	0000H
	Serial channel stop registers 0, 1 (ST0, ST1)	0000H
	Serial clock select registers 0, 1 (SPS0, SPS1)	0000H
	Serial output registers 0, 1 (SO0, SO1)	0F0FH
	Serial output enable registers 0, 1 (SOE0, SOE1)	0000H
	Serial output logic registers 0, 1 (SOL0, SOL1)	0000H
	Serial standby control register 0 (SSC0)	0000H
	Input switch control register (ISC)	00H
	Serial status register S0 (SSRS0)	0000H
	Serial status register S1 (SSRS1)	0000H

Notes 1. 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.

2. The WDTE reset value depends on the setting of the option byte.

Remark The special function register (SFR) mounted depends on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 21-2. Hardware Statuses After Reset Acknowledgment (3/5)

Hardware		Status After Reset Acknowledgment[Note]
Serial array unit (SAU)	Serial flag clear trigger register S0 (SIRS0)	0000H
	Serial flag clear trigger register S1 (SIRS1)	0000H
	Serial mode register S0 (SMRS0)	0020H
	Serial mode register S1 (SMRS1)	0020H
	Serial communication operation setting register S0 (SCRS0)	0087H
	Serial communication operation setting register S1 (SCRS1)	0087H
	Serial channel enable status register S (SES)	0000H
	Serial channel start register S (SSS)	0000H
	Serial channel stop register S (STS)	0000H
	Serial clock select register S (SPSS)	0000H
	Serial output register S (SOS)	0303H
	Serial output enable register S (SOES)	0000H
	Serial output level register S (SOLS)	0000H
	Serial data register S0 (SDRS0)	0000H
	Serial data register S1 (SDRS1)	0000H
Serial interface IICA	IICA shift register 0 (IICA0)	00H
	IICA status register 0 (IICS0)	00H
	IICA flag register 0 (IICF0)	00H
	IICA control register 00 (IICCTL00)	00H
	IICA control register 01 (IICCTL01)	00H
	IICA low-level width setting register 0 (IICWL0)	FFH
	IICA high-level width setting register 0 (IICWH0)	FFH
	IICA slave address register 0 (SVA0)	00H
Multiplier & divider, multiply-accumulator	Multiplication/division data register A (L) (MDAL/MULA)	0000H
	Multiplication/division data register A (H) (MDAH/MULB)	0000H
	Multiplication/division data register B (L) (MDBL/MULOL)	0000H
	Multiplication/division data register B (H) (MDBH/MULOH)	0000H
	Multiplication/division data register C (L) (MDCL)	0000H
	Multiplication/division data register C (H) (MDCH)	0000H
	Multiplication/division control register (MDUC)	00H
Key interrupt	Key return mode register (KRM)	00H

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 The special function register (SFR) mounted depend on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 21-2. Hardware Statuses After Reset Acknowledgment (4/5)

Hardware		Status After Reset Acknowledgment[Note 1]
Reset function	Reset control flag register (RESF)	Undefined[Note 2]
Voltage detector	Voltage detection register (LVIM)	00H[Note 2]
	Voltage detection level register (LVIS)	00H/01H/81H[Notes 2, 3]
DMA controller	DMA SFR address registers 0, 1 (DSA0, DSA1)	00H
	DMA RAM address registers 0L, 0H, 1L, 1H (DRA0L, DRA0H, DRA1L, DRA1H)	00H
	DMA byte count registers 0L, 0H, 1L, 1H (DBC0L, DBC0H, DBC1L, DBC1H)	00H
	DMA mode control registers 0, 1 (DMC0, DMC1)	00H
	DMA operation control registers 0, 1 (DRC0, DRC1)	00H
Interrupt	Interrupt request flag registers 0L, 0H, 1L, 1H, 2L, 2H (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)	00H
	Interrupt mask flag registers 0L, 0H, 1L, 1H, 2L, 2H (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)	FFH
	Priority specification flag registers 00L, 00H, 01L, 01H, 02L, 02H, 10L, 10H, 11L, 11H, 12L, 12H (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)	FFH
	External interrupt rising edge enable registers 0, 1 (EGP0, EGP1)	00H
	External interrupt falling edge enable registers 0, 1 (EGN0, EGN1)	00H
Safety functions	Flash memory CRC control register (CRC0CTL)	00H
	Flash memory CRC operation result register (PGCRCL)	0000H
	CRC input register (CRCIN)	00H
	CRC data register (CRCD)	0000H
	Invalid memory access detection control register (IAWCTL)	00H
	RAM parity error control register (RPECTL)	00H
Flash memory	Data flash control register (DFLCTL)	00H
BCD correction circuit	BCD correction result register (BCDADJ)	Undefined

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

Notes 1. 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.

2. These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM parity error	Reset by illegal-memory access	Reset by LVD
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held	Held
	WDTRF bit			Held	Set (1)	Held	Held	Held
	RPERF bit			Held	Held	Set (1)	Held	Held
	IAWRF bit			Held	Held	Held	Set (1)	Held
	LVIRF bit			Held	Held	Held	Held	Set (1)
LVIM		Cleared (00H)						Held
LVIS		00H/01H/81H						Held

3. 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

Remark The special function register (SFR) mounted depend on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 21-2. Hardware Statuses After Reset Acknowledgment (5/5)

Hardware		Status After Reset Acknowledgment[Note]
Peripheral I/O redirection register (PIOR)		00H
High-speed on-chip oscillator divider setting register (HOCODIV)		Undefined
Peripheral enable register X (PERX)		00H
Peripheral clock select register (PCKSEL)		00H
Port mode register X0 (PMX0)		01H
Port mode register X1 (PMX1)		01H
Port mode register X2 (PMX2)		01H
Port mode register X3 (PMX3)		01H
Port mode register X4 (PMX4)		01H
Port input enable register (PIEN)		00H
Noise filter enable register X (NFENX)		00H
Asynchronous serial interface LIN-UART (UARTF)	LIN-UART0 control register 0 (UF0CTL0)	10H
	LIN-UART0 option register 0 (UF0OPT0)	14H
	LIN-UART0 control register 1 (UF0CTL1)	0FFFH
	LIN-UART0 option register 1 (UF0OPT1)	00H
	LIN-UART0 option register 2 (UF0OPT2)	00H
	LIN-UART0 status register (UF0STR)	0000H
	LIN-UART0 status clear register (UF0STC)	0000H
	LIN-UART0 wait transmit data register (UF0WTX)	0000H
	LIN-UART0 ID setting register (UF0ID)	00H
	LIN-UART0 buffer register 0 (UF0BUF0)	00H
	LIN-UART0 buffer register 1 (UF0BUF1)	00H
	LIN-UART0 buffer register 2 (UF0BUF2)	00H
	LIN-UART0 buffer register 3 (UF0BUF3)	00H
	LIN-UART0 buffer register 4 (UF0BUF4)	00H
	LIN-UART0 buffer register 5 (UF0BUF5)	00H
	LIN-UART0 buffer register 6 (UF0BUF6)	00H
	LIN-UART0 buffer register 7 (UF0BUF7)	00H
	LIN-UART0 buffer register 8 (UF0BUF8)	00H
	LIN-UART0 buffer control register (UF0BUCTL)	0000H
	LIN-UART0 transmit data register (UF0TX)	0000H
	LIN-UART0 receive data register (UF0RX)	0000H
Wakeup timer control register (WUTMCTL)		00H
Wakeup timer compare register (WUTMCMP)		0000H

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 The special function register (SFR) mounted depend on the product. See **3.1.4 Special function registers (SFRs)** and **3.1.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

21.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the RL78/F12. 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.

Figure 21-5. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: 00H 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.

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

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.

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

2. During RAM fetching, instruction codes are not subject to parity error detection. However, parity error detection is performed on RAM data read during RAM instruction fetching.

3. With the RL78, the CPU performs read-ahead for pipeline operation to read the RAM area not yet initialized that follows the currently used RAM area, which may cause a RAM parity error. Therefore, when RAM parity error reset generation is enabled (RPERDIS = 0), be sure to initialize the RAM area to be used and 10 more bytes.

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

Flag \ 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
TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held	Held
WDTRF bit			Held	Set (1)	Held	Held	Held
RPERF bit			Held	Held	Set (1)	Held	Held
IAWRF bit			Held	Held	Held	Set (1)	Held
LVIRF bit			Held	Held	Held	Held	Set (1)

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 1.51 V ±0.03 V.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.50 V ±0.03 V), generates internal reset signal when $V_{DD} < V_{PDR}$.

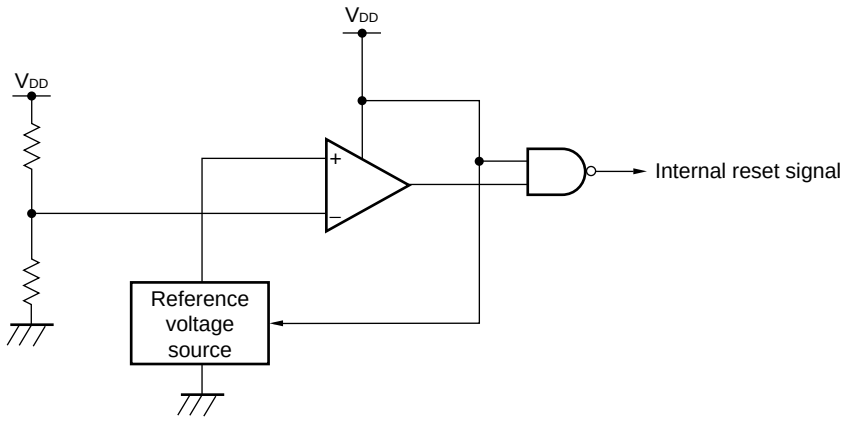
Caution If an internal reset signal is generated in the POR circuit, TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags of the reset control flag register (RESF) is cleared.

Remark This product 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**.

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

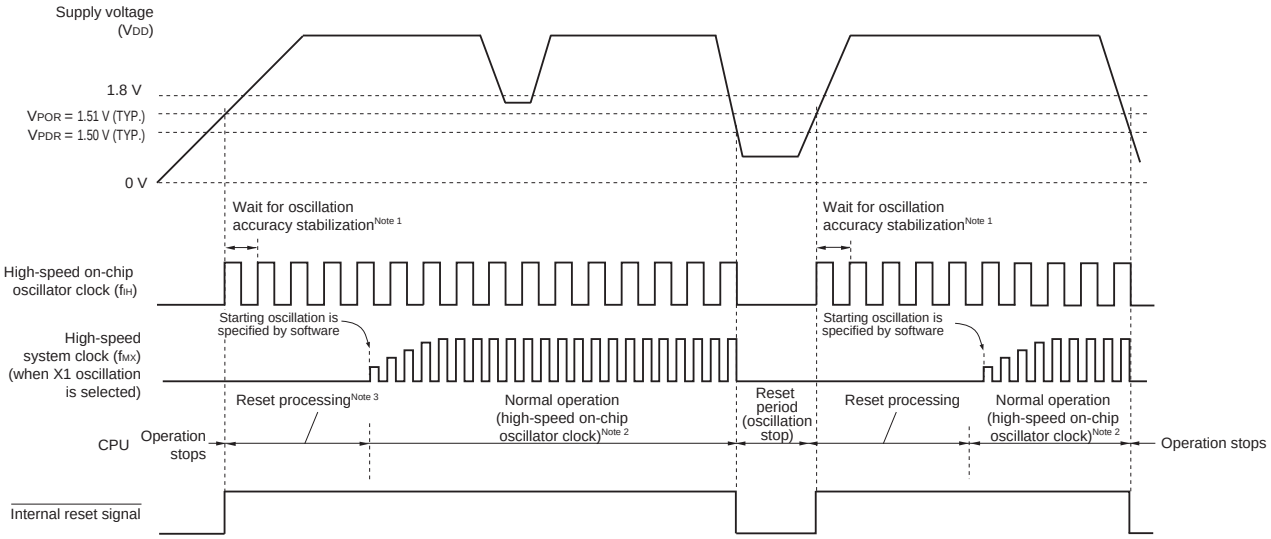
- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{PDR} = 1.51 V ±0.03 V), the reset status is released.

- The supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.50 V ±0.03 V) are compared. When $V_{DD} < V_{PDR}$, the internal reset signal is generated.

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown below.

Figure 22-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/2)

(1) When LVD is OFF (option byte 000C1H/010C1H: VPOC0 to VPOC2 = 111B)



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

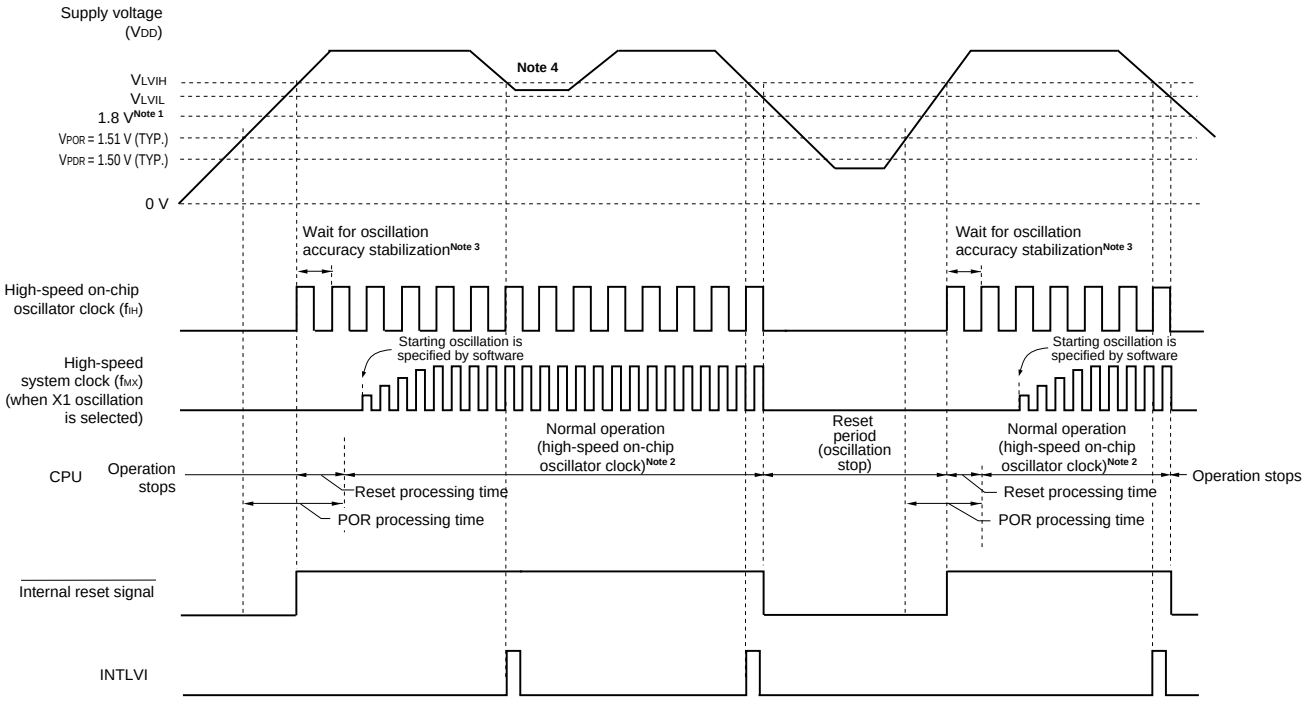
2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

3. For details about the reset processing time, see Figure 5-15.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Figure 22-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/2)

(2) When LVD is interrupt & reset mode (V_{LVIL} < V_{LVIH} (setting by option byte))



Notes 1. 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.

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

3. After the first interrupt request signal (INTLVI) is generated, the LVIL and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. If the operating voltage returns to 1.6 V or higher without falling below the voltage detection level (V_{LVDL}), after INTLVI is generated, perform the required backup processing, and then use software to specify the initial settings in order (see **Figure 23-8. Initial Setting of Interrupt and Reset Mode**).

Remark V_{LVIH}, V_{LVIL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

22.4 Cautions for Power-on-reset Circuit

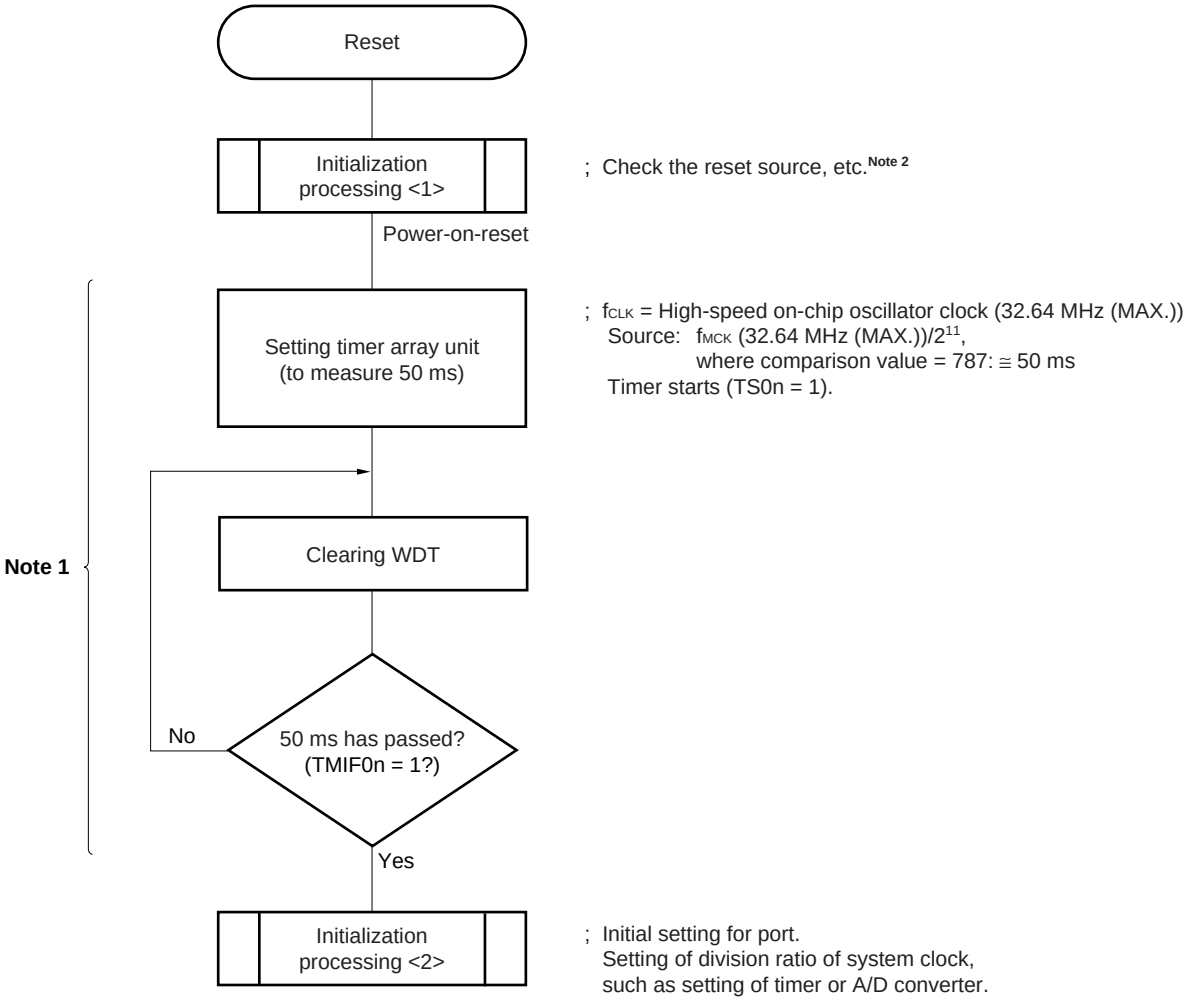
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POR detection voltage (V_{POR}, V_{PDR}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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 22-3. Example of Software Processing After Reset Release (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of POR detection voltage



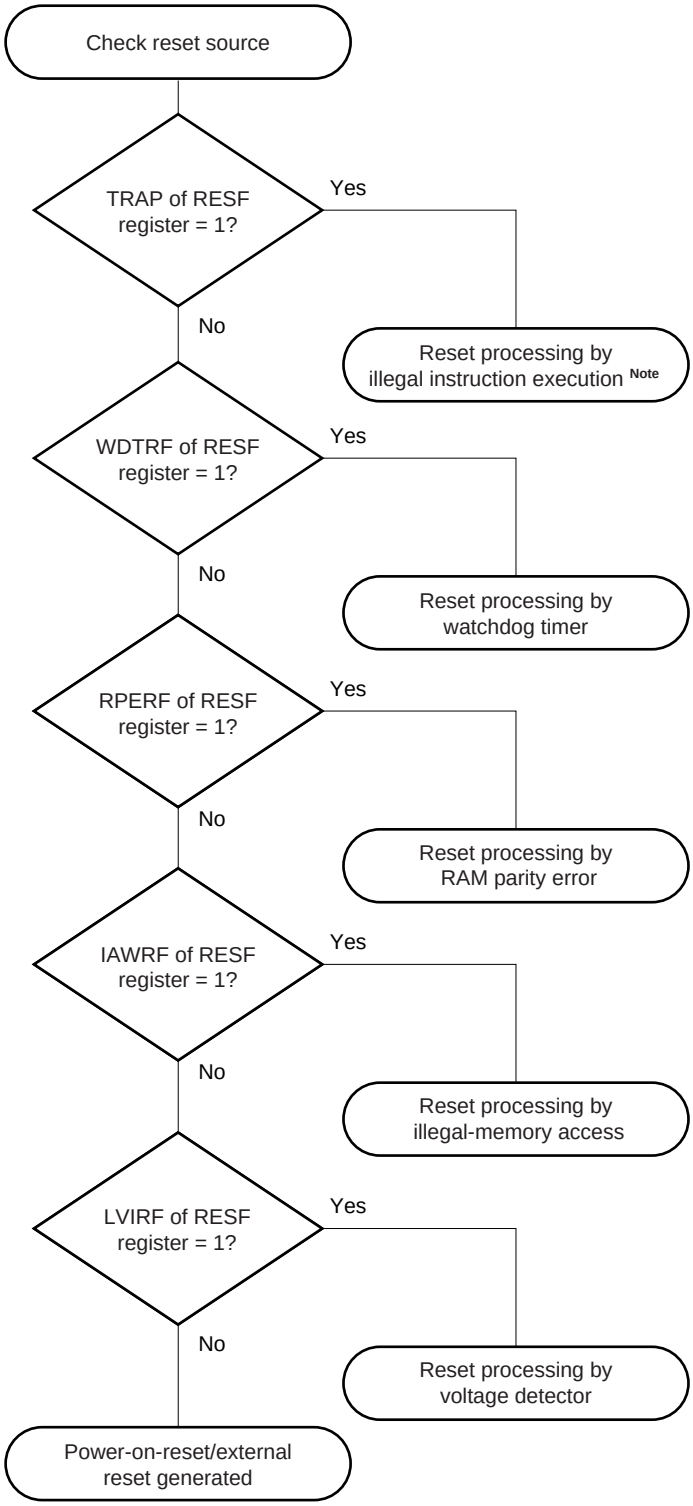
Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Remark n = 0 to 7

Figure 22-3. Example of Software Processing After Reset Release (2/2)

• Checking reset source



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.

CHAPTER 23 VOLTAGE DETECTOR

23.1 Functions of Voltage Detector

The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVIH}, V_{LVIL}), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage (V_{LVIH}, V_{LVIL}) can be selected from a maximum of 12 levels by using the option byte (For details, see **CHAPTER 26 OPTION BYTE**).
- Operable in STOP mode.
- The following three operation modes can be selected by using the option byte.

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)

For the two detection voltages selected by the option byte 000C1H/010C1H, the high-voltage detection level (V_{LVIH}) is used for generating interrupts and ending resets, and the low-voltage detection level (V_{LVIL}) is used for triggering resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)

The detection voltage (V_{LVI}) selected by the option byte 000C1H/010C1H is used for triggering and ending resets.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)

The detection voltage (V_{LVI}) selected by the option byte 000C1H/010C1H is used for generating interrupts.

Two detection voltages (V_{LVIH}, V_{LVIL}) can be specified in the interrupt & reset mode, and one (V_{LVI}) can be specified in the reset mode and interrupt mode.

The reset and interrupt signals are generated as follows according to the option byte (LVIMDS0, LVIMDS1) selection.

Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an internal interrupt signal when $V_{DD} < V_{LVIH}$, and an internal reset when $V_{DD} < V_{LVIL}$. Releases the reset signal when $V_{DD} \geq V_{LVH}$.	Generates an internal reset signal when $V_{DD} < V_{LVI}$ and releases the reset signal when $V_{DD} \geq V_{LVI}$.	Generates an internal interrupt signal when V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$).

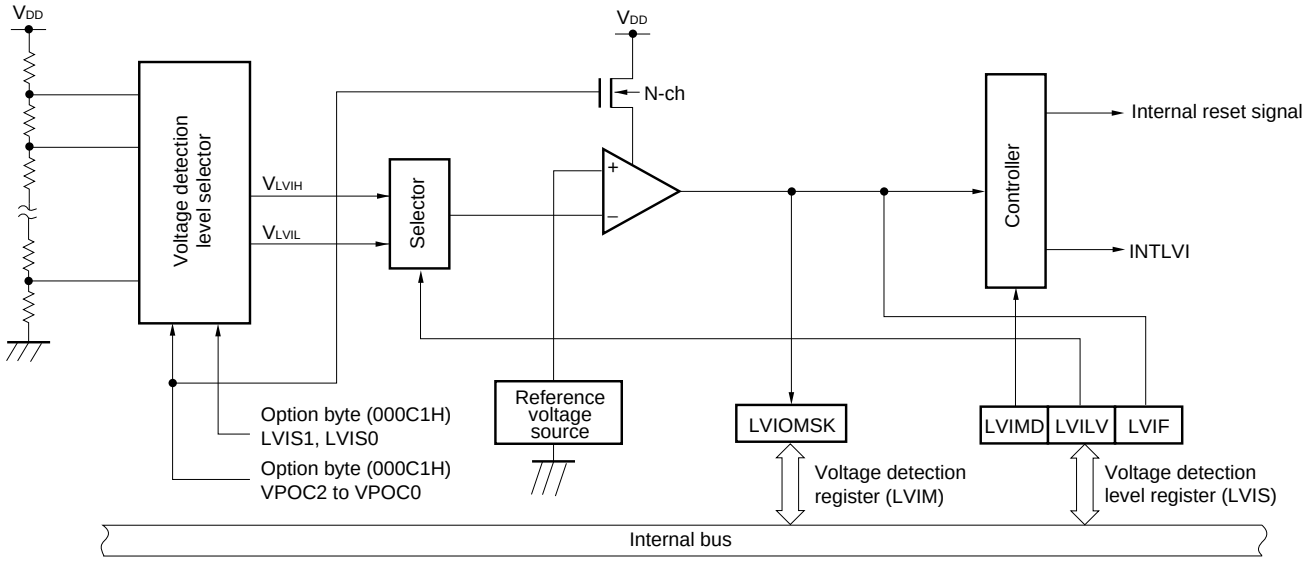
While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).

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)

(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.

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	0	0	0	0	0	LVIOMSK	LVIF

LVISEN	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling rewriting
1	Enabling rewriting Note 3

LVIOMSK	Mask status flag of LVD output
0	Mask is invalid
1	Mask is valid Note 4

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}), or when LVD operation is disabled
1	Supply voltage (V_{DD}) < detection voltage (V_{LVI})

Notes 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value. Only the bit 7 of this register is cleared by “0” if a reset other than by LVD is effected.

2. Bits 0 and 1 are read-only.

3. This can only be set when LVIMDS1 and LVIMDS0 are set to 1 and 0 (interrupt and reset mode) by the option byte.

4. LVIOMSK bit is automatically set to “1” in the following periods and reset or interruption 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.

(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 [Note1].

Figure 23-3. Format of Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 00H/01H/81H [Note 1] R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD	0	0	0	0	0	0	LVILV

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_{LVIH})
1	Low-voltage detection level (V_{LVIL} or V_{LVI})

Notes 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

2. Writing “0” can only be allowed when LVIMDS1 and LVIMDS0 are set to 1 and 0 (interrupt and reset mode) by the option byte. In other cases, writing is not allowed and the value is switched automatically when reset of interrupt is generated.

Cautions 1. Only rewrite the value of the LVIS register after setting the LVISEN bit (bit 7 of the LVIM register) to 1.

2. Specify the LVD operation mode and detection voltage (V_{LVIH}, V_{LVIL}) by using the option byte (000C1H). Table 23-1 shows the option byte (000C1H) settings. For details about the option byte, see CHAPTER 26 OPTION BYTE.

Table 23-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H/010C1H)

- LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	Mode setting						
Rise	Fall	Fall	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
1.98 V	1.94 V	1.84 V	1	0	0	0	1	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
3.75 V	3.67 V							0	0
2.92 V	2.86 V	2.75 V				1	1	1	0
3.02 V	2.96 V							0	1
4.06 V	3.98 V							0	0
Other than above			Setting prohibited						

- LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		Mode setting						
Rise	Fall	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
LVDOFF		1	1	1	1	1	×	×
1.88 V	1.84 V			0	0	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
3.75 V	3.67 V				1	0	0	0
4.06 V	3.98 V				1	1	0	0
Other than above		Setting prohibited						

Remark ×: Don't care.

- LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		Mode setting						
Rise	Fall	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
LVDOFF		0	1	1	1	1	×	×
1.88 V	1.84 V			0	0	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
3.75 V	3.67 V				1	0	0	0
4.06 V	3.98 V				1	1	0	0
Other than above		Setting prohibited						

Remark ×: Don't care.

- LVD setting (LVD off)

Detection voltage		Option byte setting value						
V_{LVD}		Mode setting						
Rise	Fall	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
–	–	0/1	1	1	×	×	×	×
Other than above		Setting prohibited						

Remark ×: Don't care.

23.4 Operation of Voltage Detector

23.4.1 When used as reset mode

- When starting operation

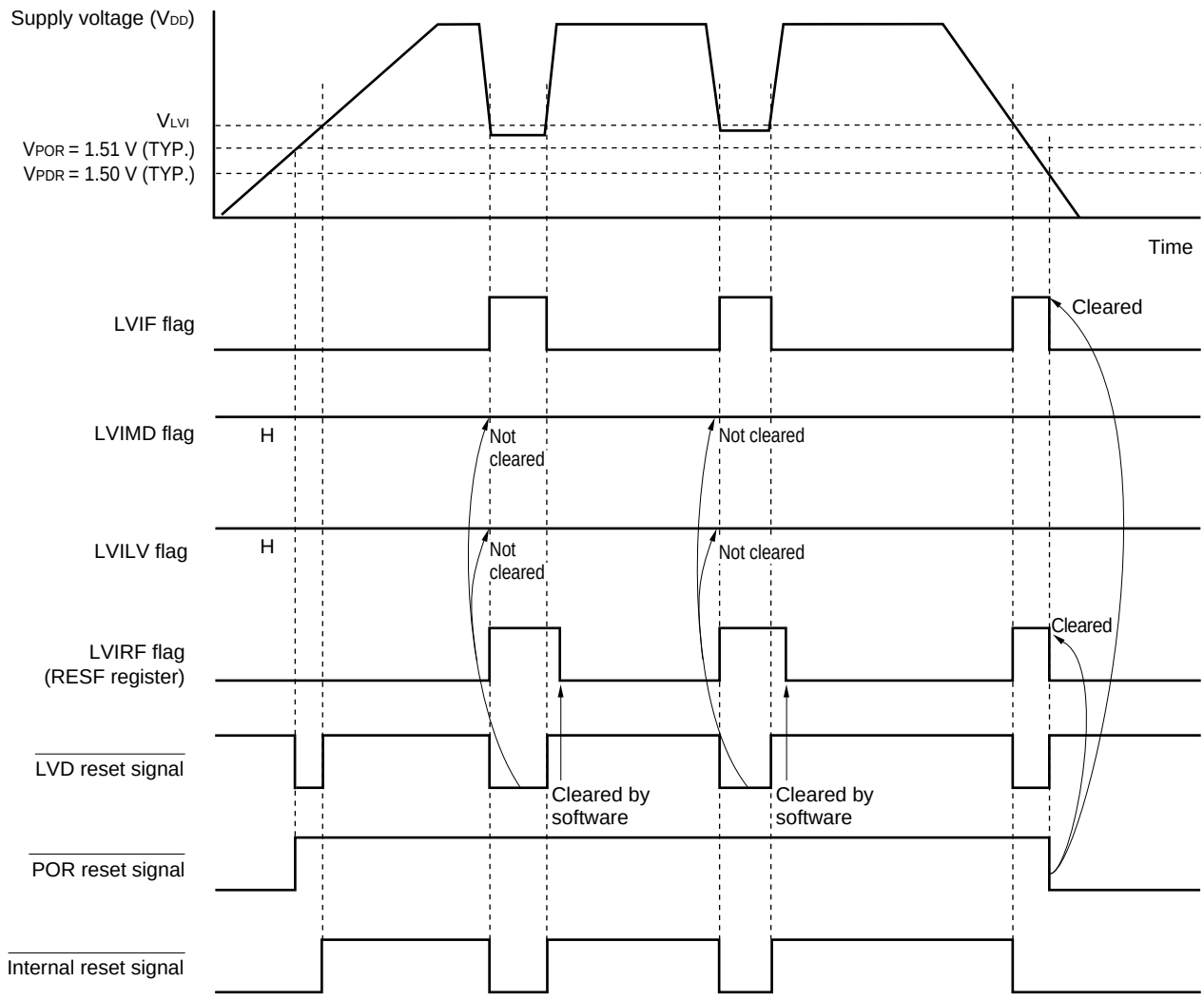
  Start in the following initial setting state.

  Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the detection voltage (V_{LVI}) by using the option byte 000C1H/010C1H.

  - Set bit 7 (LVISEN) of the voltage detection register (LVIM) to 0 (disable rewriting of voltage detection level register (LVIS)).
  - When the option byte LVIMDS1 and LVIMDS0 are set to 1, the initial value of the LVIS register is set to 81H.
    Bit 7 (LVIMD) is 1 (reset mode).
    Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVI}).

  Figure 23-4 shows the timing of the internal reset signal generated by the voltage detector.

Figure 23-4. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

23.4.2 When used as interrupt mode

- When starting operation

  Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the detection voltage (V_{LVI}) by using the option byte 000C1H/010C1H.

  Start in the following initial setting state.

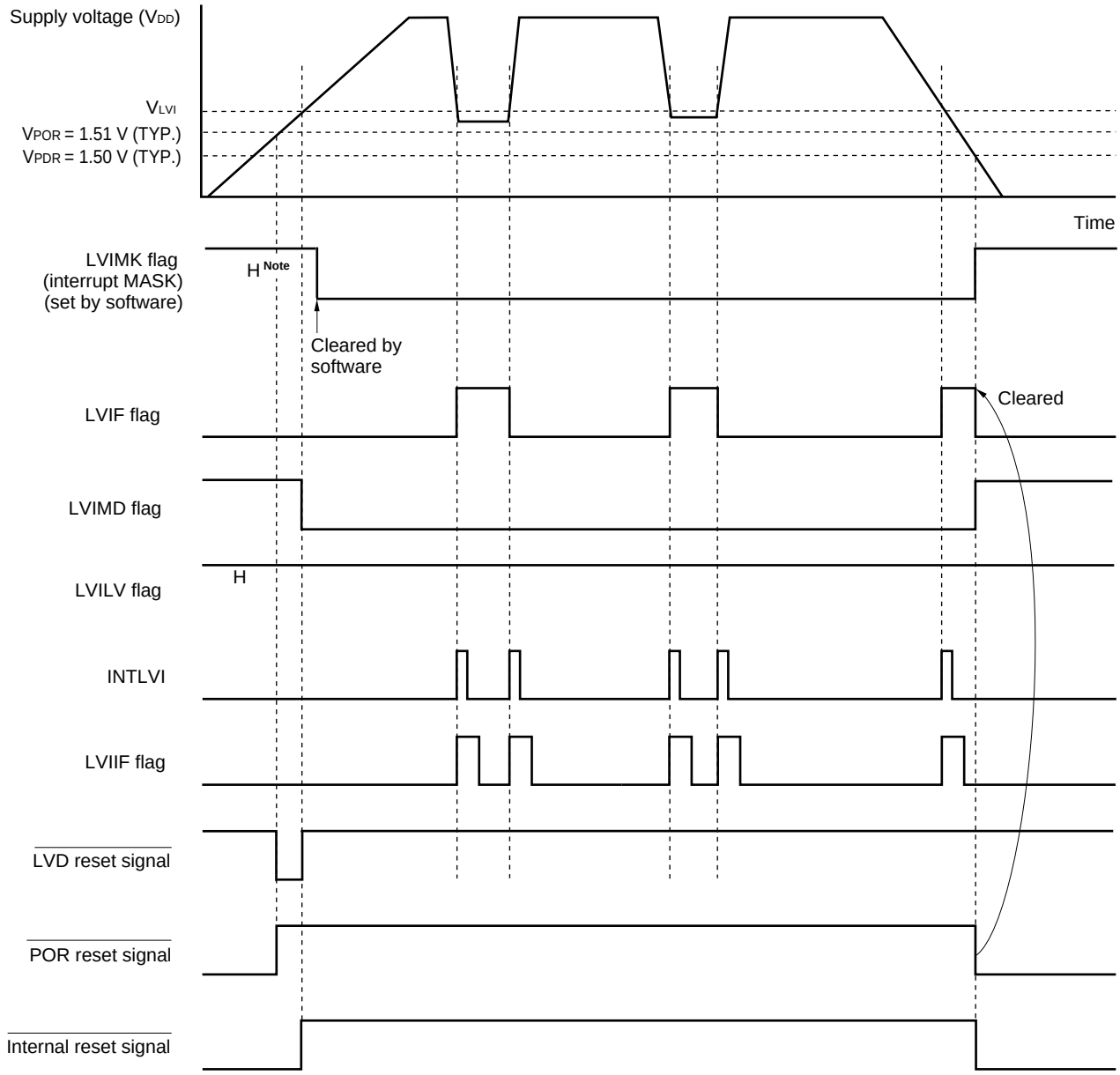
  - Set bit 7 (LVISEN) of the voltage detection register (LVIM) to 0 (disable rewriting of voltage detection level register (LVIS)).
  - When the option byte LVIMDS1 is clear to 0 and LVIMDS0 is set to 1, the initial value of the LVIS register is set to 01H.

    Bit 7 (LVIMD) is 0 (interrupt mode).

    Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVI}).

  Figure 23-5 shows the timing of the internal interrupt signal generated by the voltage detector.

Figure 23-5. Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Note The LVIMK flag is set to “1” by reset signal generation.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

23.4.3 When used as interrupt and reset mode

- When starting operation

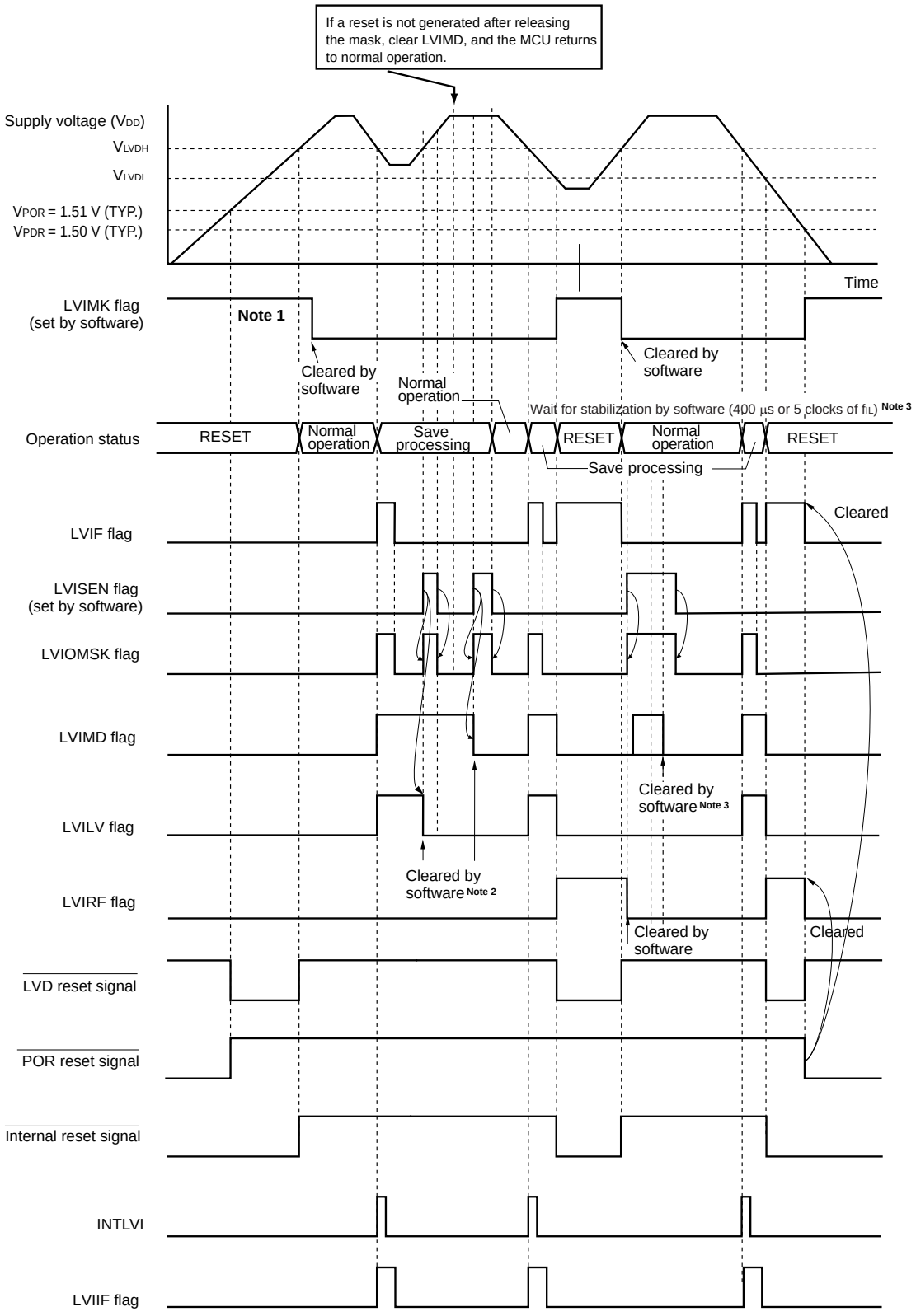
  Specify the operation mode (the interrupt and reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte 000C1H.

  Start in the following initial setting state.

  - Set bit 7 (LVISEN) of the voltage detection register (LVIM) to 0 (disable rewriting of voltage detection level register (LVIS)).
  - When the option byte LVIMDS1 is set to 1 and LVIMDS0 is cleared to 0, the initial value of the LVIS register is set to 00H.
    Bit 7 (LVIMD) is 0 (interrupt mode).
    Bit 0 (LVILV) is 0 (low-voltage detection level: V_{LVDH}).

  Figure 23-6 shows the timing of the internal reset signal and interrupt signal generated by the voltage detector. Perform the processing according to **Figure 23-7 Processing Procedure after an Interrupt is Generated** and **Figure 23-8 Initial Setting of Interrupt and 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.)

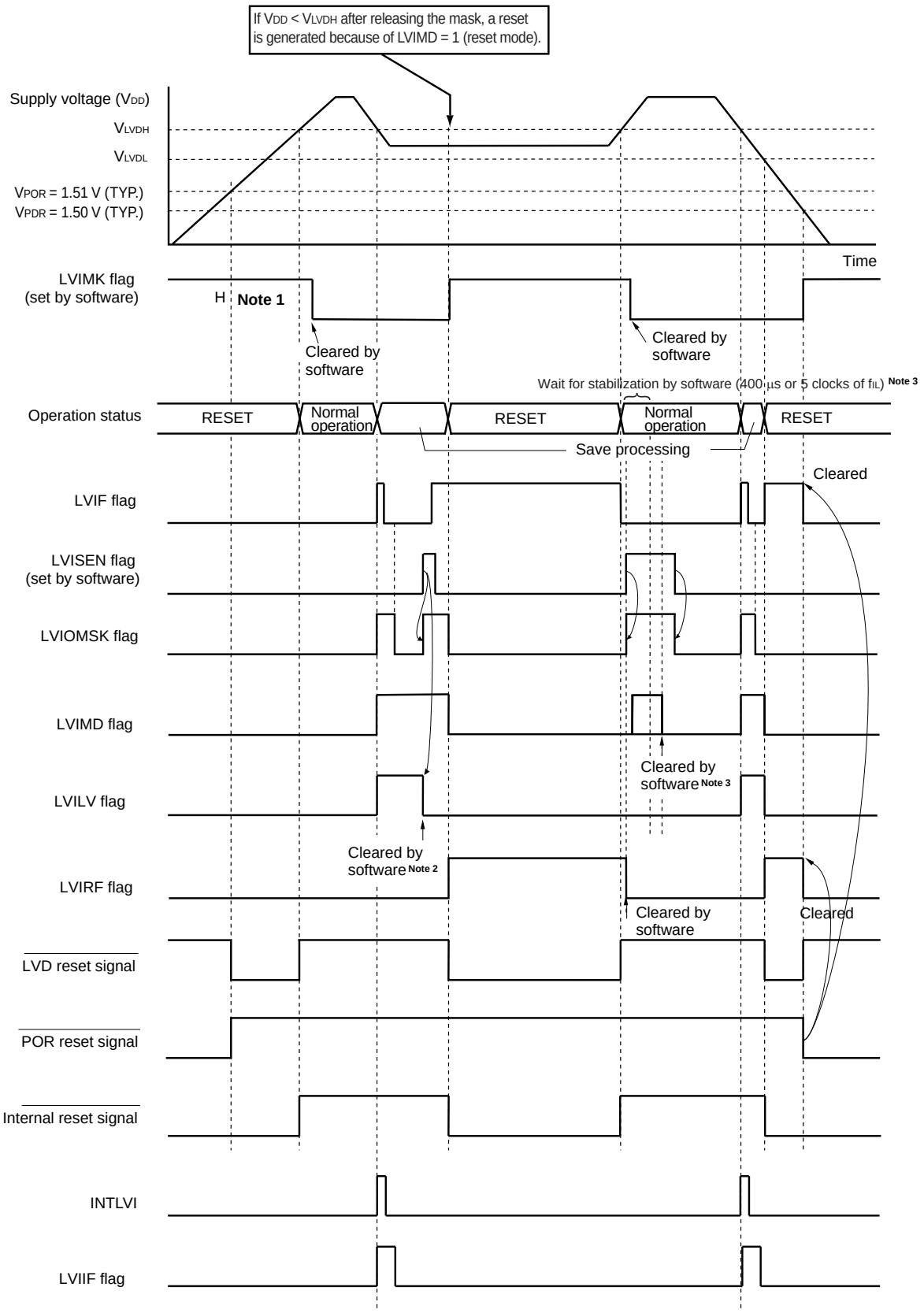
Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 23-7 Processing Procedure after an Interrupt is Generated** in interrupt and reset mode.

3. After a reset is released, perform the processing according to **Figure 23-8 Initial Setting of Interrupt and Reset Mode** in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 23-6. 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.)

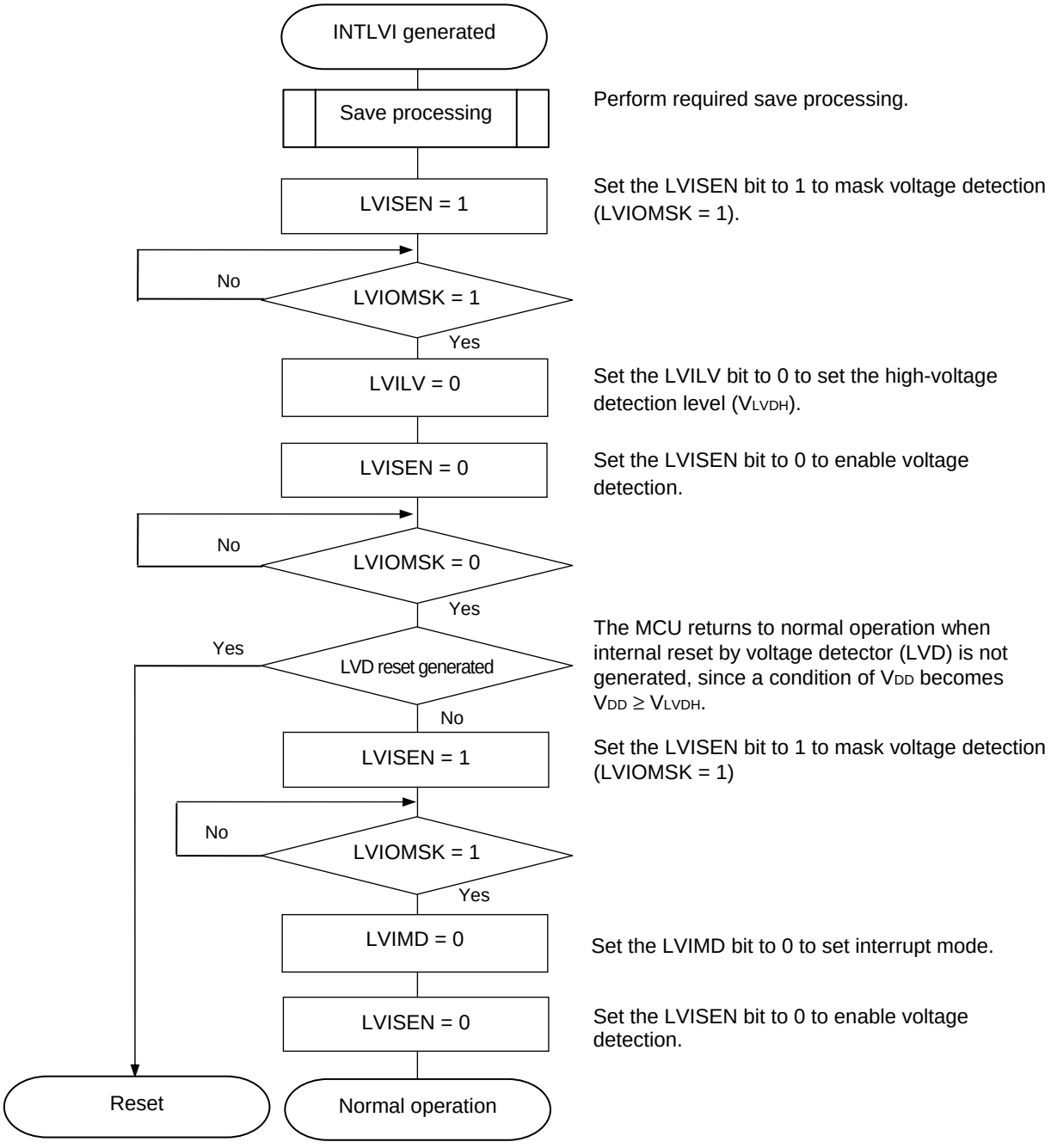
Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 23-7 Processing Procedure after an Interrupt is Generated** in interrupt and reset mode.

3. After a reset is released, perform the processing according to **Figure 23-8 Initial Setting of Interrupt and Reset Mode** in interrupt and reset mode.

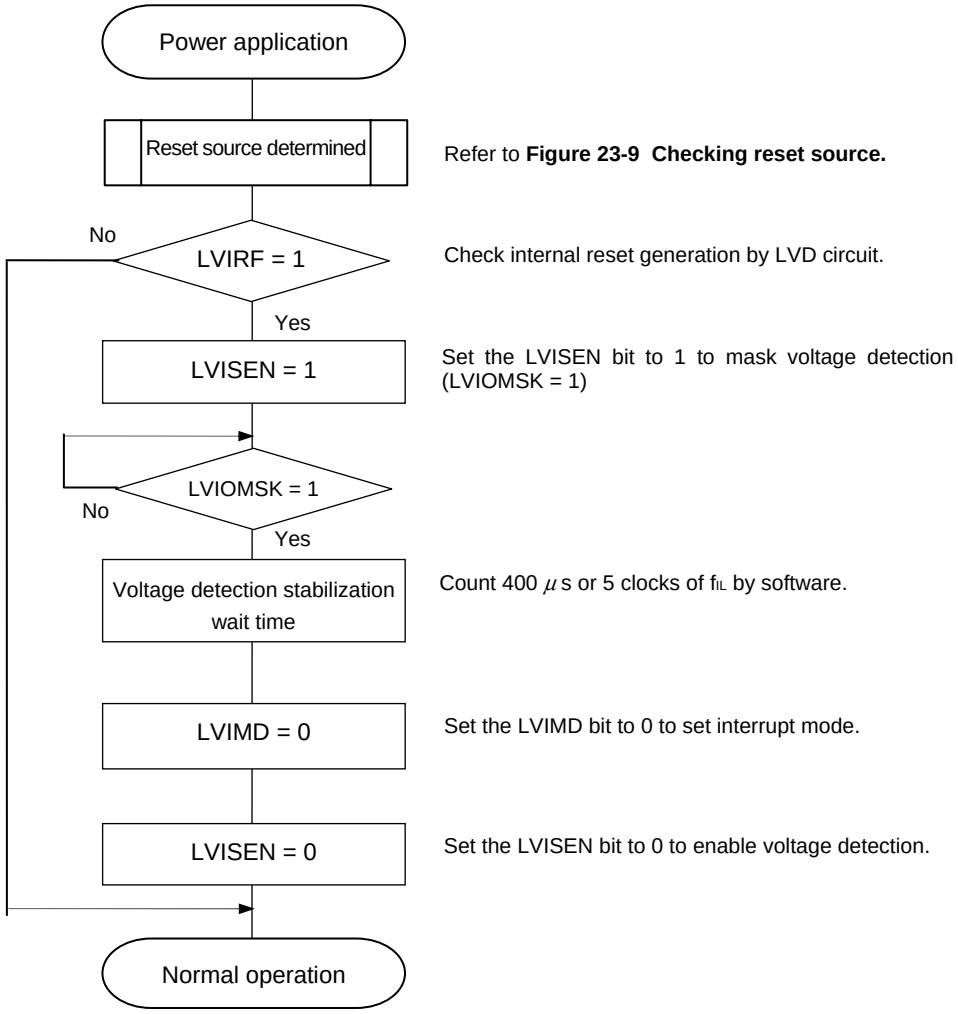
Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 23-7. Processing Procedure after an Interrupt is Generated



When setting an interrupt and reset mode (LVIMDS1, LVIMDS0 = 1, 0), voltage detection stabilization wait time for 400 µs or 5 clocks of f_{IL} is necessary after LVD reset is released (LVIRF = 1). After waiting until voltage detection stabilizes, clear the LVIMD bit to 0 for initialization. While voltage detection stabilization wait time is being counted and when the LVIMD bit is rewritten, set LVISEN to 1 to mask a reset or interrupt generation by LVD.
Figure 23-8 shows the procedure for initial setting of interrupt and reset mode.

Figure 23-8. Initial Setting of Interrupt and Reset Mode



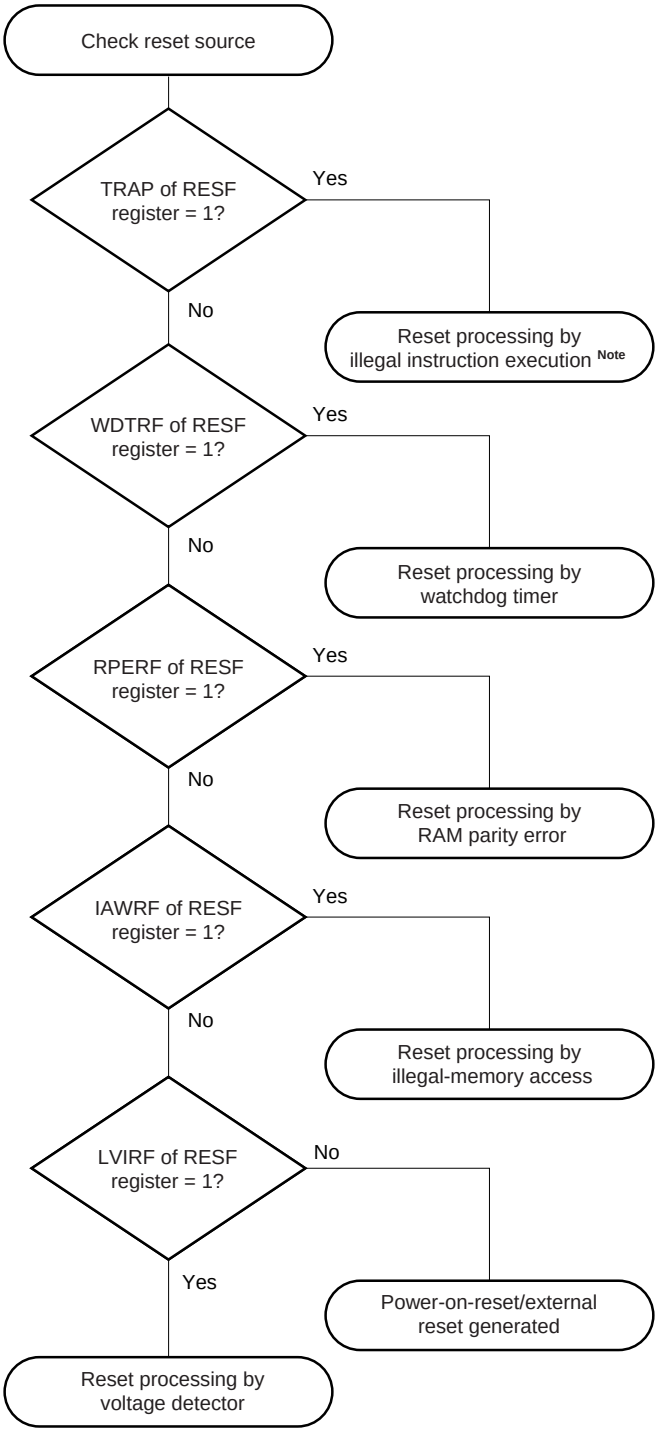
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

23.5 Cautions for Voltage Detector

(1) Checking reset source

When a reset occurs, check the reset source by using the following method.

Figure 23-9. Checking reset source



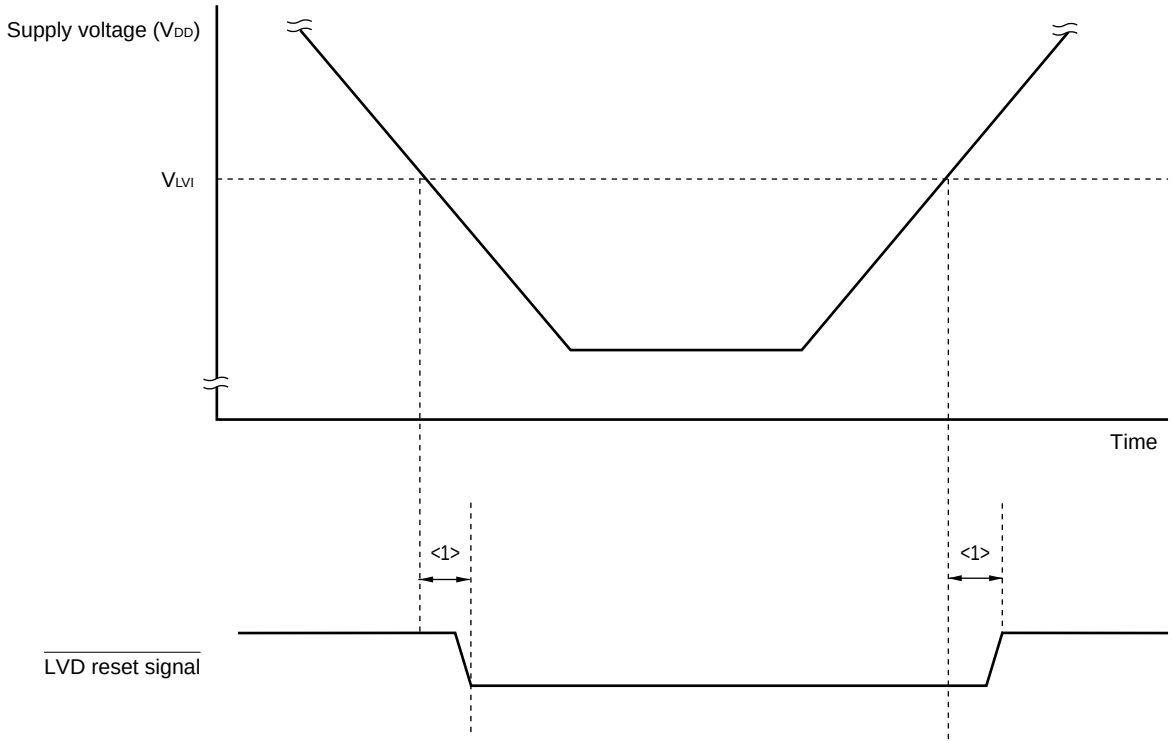
Note When instruction code FFH is executed.
Reset by the illegal instruction execution is not issued by emulation with the in-circuit emulator or on-chip debug emulator.

(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_{LVI}) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage (V_{LVI}) ≤ supply voltage (V_{DD}) until the time LVD reset has been released (see **Figure 23-10**).

Figure 23-10. Delay from the Time LVD Reset Source is Generated until the Time LVD Reset Has been Generated or Released



<1>: Detection delay (300 μs (MAX.))

CHAPTER 24 SAFETY FUNCTIONS

24.1 Overview of Safety Functions

The RL78/F12 is provided with the following safety functions to meet the IEC 60730 and IEC 61508 safety standards. The functions are intended to detect failures through microcomputer's self-diagnosis and to stop the system safely.

(1) Flash memory CRC operation function (high-speed CRC and general-purpose CRC)

This detects errors associated to data in the flash memory by performing CRC operations.
Either of the CRC operations below can be used according to the application and conditions of use.

- High-speed CRC: Can be used for high-speed check of the entire code flash memory area while the CPU is stopped in the initialization routine.
- General-purpose CRC: Can be used for not only check of code flash memory area but versatile check 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 access to invalid memory areas (non-existent areas or access-restricted areas).

(6) Frequency detection function

This uses TAU to detect the oscillation frequency.

(7) A/D test function

This is used to perform a self-check of A/D conversion by performing A/D conversion on the internal reference voltage.

24.2 Registers Used by Safety Functions

Each safety function uses the following registers.

Register Name	Safety Functions
• 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
• Analog input channel specification register (ADS)	Specification of the analog voltage input channel

For details of the registers, refer to **24.3 Operations of Safety Functions**.

24.3 Operations of Safety Functions

24.3.1 Flash Memory CRC Operation Function (High-Speed CRC)

The IEC 60730 standard requires verification of data in flash memory and recommends CRC as the means for verification. With the high-speed CRC operation function, the entire code flash memory area can be checked in the initialization routine. This function is available only in HALT mode of the main system clock through the program on RAM.

With the high-speed CRC operation function, the CPU is stopped and a 32-bit data unit is read from the flash memory in a clock cycle to perform operation on the data. Therefore, check can be completed in a short time (for example, 64-Kbyte flash memory checked in 512 μs at 32 MHz frequency).

The CRC generator polynomial is $X^{16} + X^{12} + X^5 + 1$ of CRC-16-CCITT.

Operation is done with the MSB first, i.e., from bit 31 to bit 0.

Caution Since the monitor program is allocated for on-chip debugging, a different operation result is obtained.

Remark Since the general-purpose CRC uses the LSB-fist method, a different operation result is obtained.

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	0 to 3FFBH (16K – 4 bytes)
0	0	0	0	0	1	0 to 7FFBH (32K – 4 bytes)
0	0	0	0	1	0	0 to BFFBH (48K – 4 bytes)
0	0	0	0	1	1	0 to FFFBH (64K – 4 bytes)
Other than the above						Setting prohibited

Caution With the 8-Kbyte ROM products, the high-speed CRC is not available. With the 24-Kbyte ROM products, the high-speed CRC is only available to the range up to 16 Kbytes.

Remark In the last 4 bytes of the flash memory, store in advance the expected value of the CRC operation result for comparison. The above table thus shows the operation range that is smaller 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

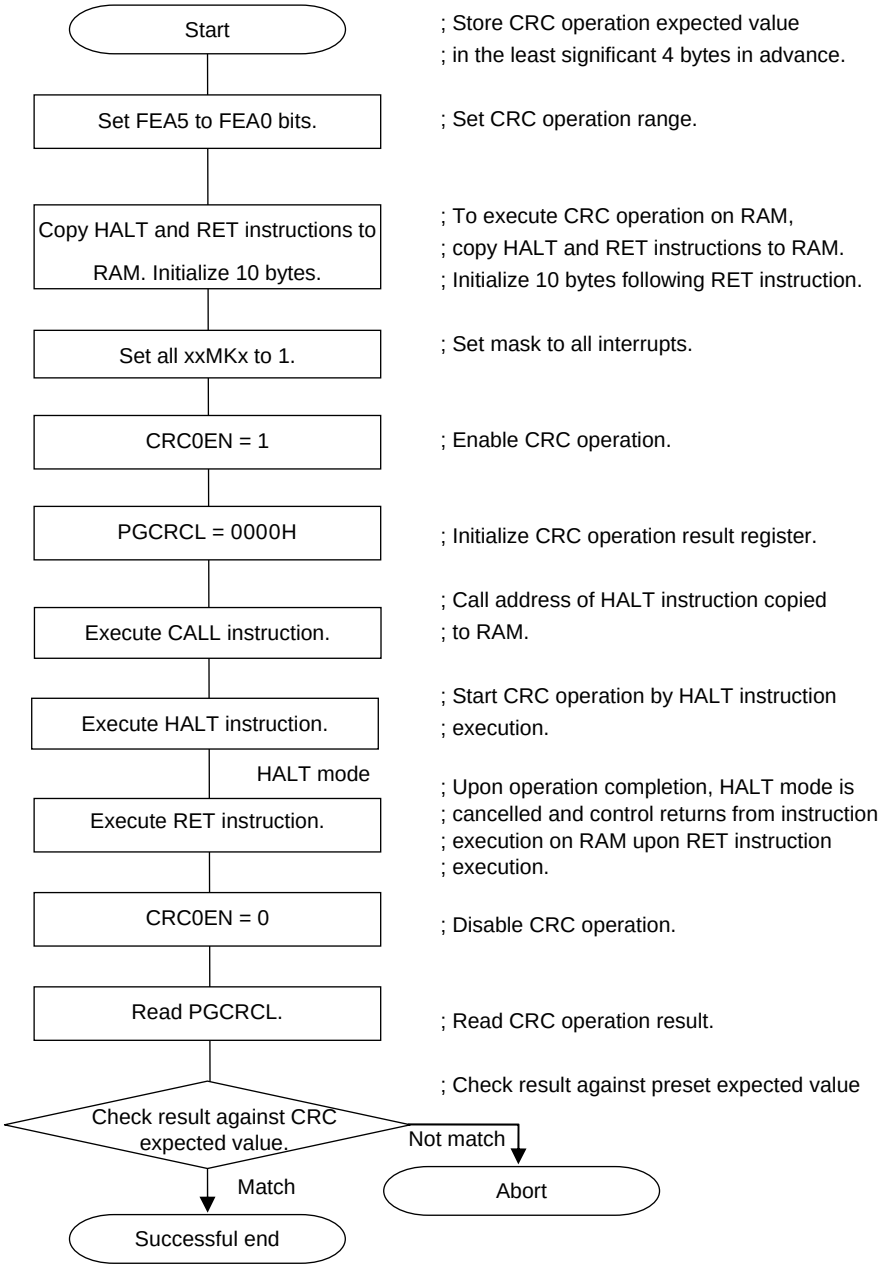
Symbol	7	6	5	4	3	2	1	0
PGCRCL	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

PGCRC15 to 0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written to if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 24-3 shows a flowchart of the flash memory CRC operation function (high-speed CRC).

Figure 24-3. Flowchart of Flash Memory CRC Operation Function (High-Speed CRC)



Cautions 1. Only code flash memory is subject to CRC operation.

2. Store the CRC operation expected value in the area following the operation range of the code flash memory.

3. Boot swapping is not performed during CRC operation.

4. Executing the HALT instruction in the RAM area enables CRC operation; be sure to execute the HALT instruction in the RAM area.

The CRC expected value can be calculated using the development environment CubeSuite+ or the equivalents (refer to the CubeSuite+ user's manual.

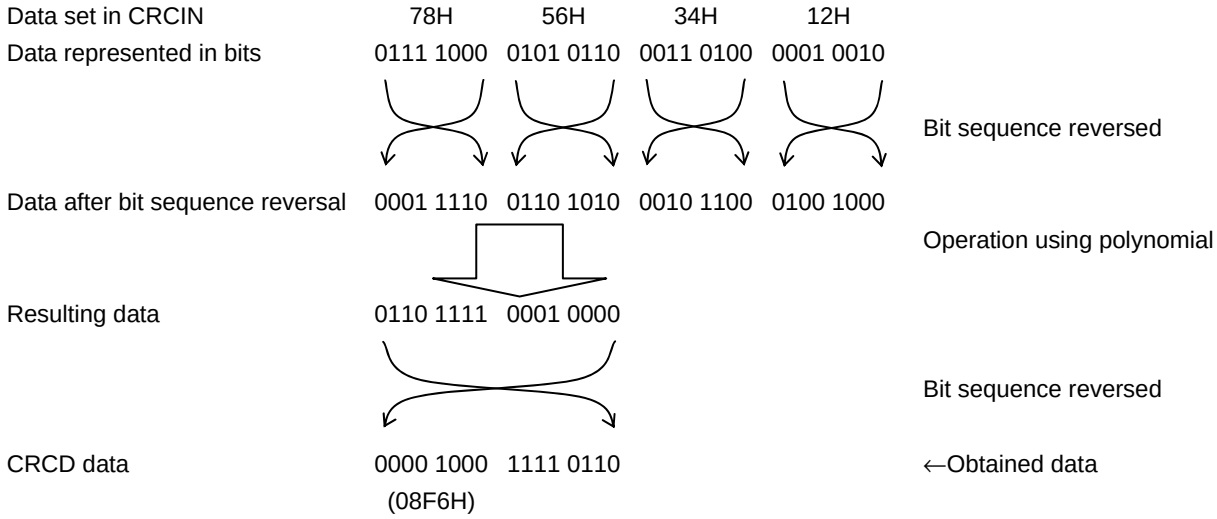
24.3.2 CRC Operation Function (General-Purpose CRC)

The IEC 61508 standard requires safety to be guaranteed during operation and thus the means for data verification during CPU operation is necessary.

With the general-purpose CRC operation function, the CRC operation is possible as a peripheral function during CPU operation. The general-purpose CRC operation function can be used for not only check of code flash memory area but versatile check. Data to be checked is specified with the user program. In HALT mode, the CRC operation function is available only during DMA transfer.

The CRC operation function is available both in main and subsystem clock operation modes.

The CRC generator polynomial is $X^{16} + X^{12} + X^5 + 1$ of CRC-16-CCITT. Operation is done after the input data is reversed in bit sequence to accommodate to the LSB-first communication. For example, when data 12345678H is to be transmitted with the LSB first, writing 78H, 56H, 34H, and 12H in CRCIN register in this order allows value 08E6H to be obtained from the CRCD register. This value is obtained as shown below, where data 12345678H is reversed in bit sequence and then the resulting bit strings are subjected to CRC operation.



Caution During program execution, the debugger replaces the line in which software break is set with the break instruction; therefore, setting a software break in the area subject to CRC operation causes a different operation result to be obtained.

24.3.2.1 CRC input register (CRCIN)

This is an 8-bit register used to set the data for general-purpose CRC operation.

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 to, 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

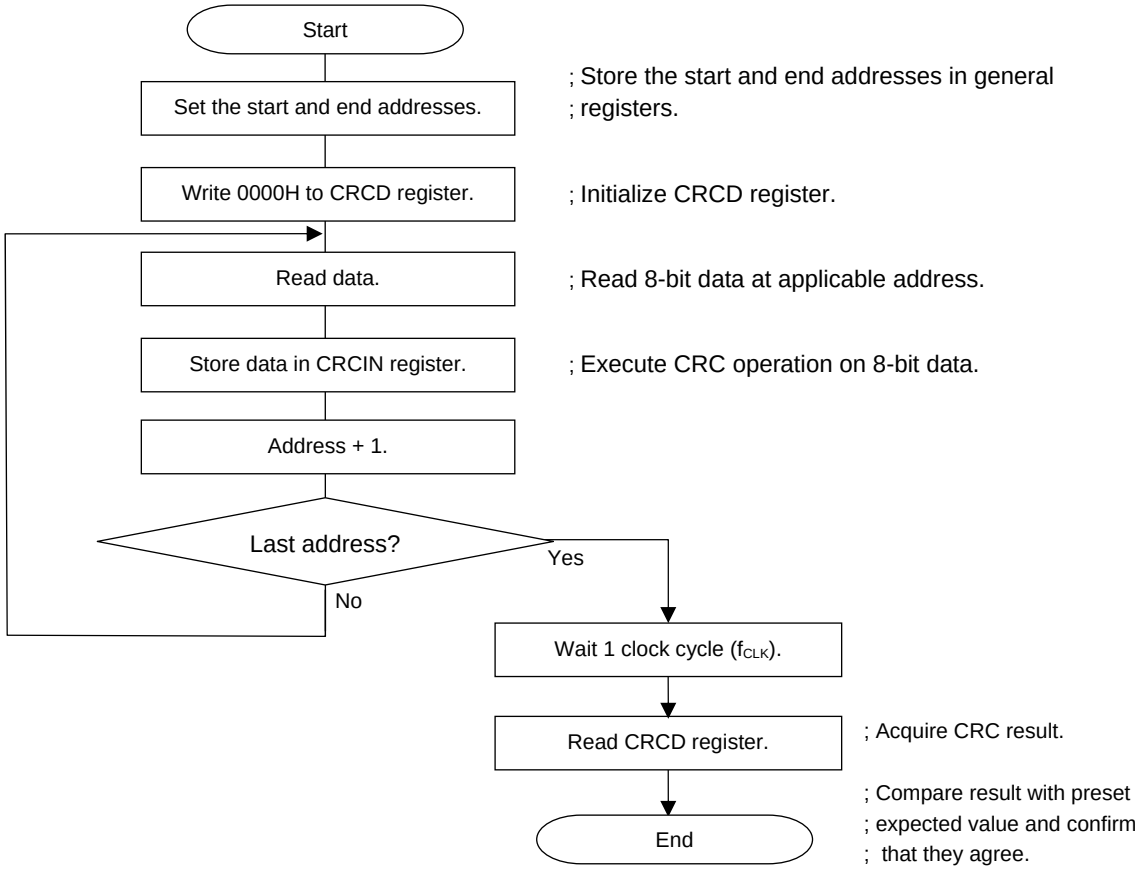
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Cautions 1. Read the value written to CRCD register before writing to CRCIN register.

2. If writing and storing operation result to CRCD register conflict, the writing is ignored.

<Operation flow>

Figure 24-6. Flowchart of CRC Operation Function (General-Purpose CRC)



24.3.3 RAM Parity Error Detection Function

The IEC 60730 standard requires verification of RAM data. To meet the requirement, one parity bit is appended to each 8-bit data in the RL78/F12 RAM. With the RAM parity error detection function, a parity is written when data is written and the parity is checked when data is read out. A reset can be generated upon occurrence of a parity error.

24.3.3.1 RAM parity error control register (RPECTL)

This register is used to check occurrence of a parity error and control resets 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	Enables parity error resets.
1	Disables parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution With the RL78, the CPU performs read-ahead for pipeline operation to read the RAM area not yet initialized that follows the currently used RAM area, which may cause a RAM parity error. Therefore, when RAM parity error reset generation is enabled (RPERDIS = 0), be sure to initialize the RAM area to be used and 10 more bytes. When the self-programming function is used, be sure to initialize the RAM area to be rewritten to and 10 more bytes before rewrite.
Parity error detection is performed on RAM data read during RAM instruction fetching.

Remarks 1. The RAM parity check function is always on and the check results can be read from the RPEF flag.

2. In the initial state, parity error reset generation is enabled (RPERDIS = 0). Even if parity error reset generation is disabled (RPERDIS = 1), the RPEF flag is set (1) when a parity error occurs.

3. The RPEF flag 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.

24.3.4 RAM Guard Function

The IEC 61508 standard requires safety to be guaranteed during operation and thus it is necessary to protect significant data stored in RAM when the CPU freezes.

The RAM guard function protects data in the specified space.

Setting this function disables writing to RAM in the specified space but enables reading normally.

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.

The RAM guard function uses the GRAM1 and GRAM0 bits.

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

Reset signal generation clears this register to 00H.

Figure 24-8. 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 lower RAM address
1	0	The 256 bytes starting at the lower RAM address
1	1	The 512 bytes starting at the lower 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

The IEC 61508 standard requires safety to be guaranteed during operation and thus it is necessary to prevent significant SFRs from being erroneously rewritten when the CPU freezes.

The SFR guard function protects data in the registers used to control the port function, interrupt function, clock control function, voltage detection circuits, and RAM parity error detection function.

Setting this function disables writing to guarded SFRs but enables reading normally.

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.

The SFR guard function uses the GPORT, GINT, and GCSC bits.

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

GPORT	Port function control register guard
0	Disabled. Port function control registers can be read or written to.
1	Enabled. Writing to port function control registers is disabled. Reading is enabled. Guarded SFRs: PMxx, PUxx, PIMxx, POMxx, PMCxx, ADPC, PIOR Note 1

GINT	Interrupt function control register guard
0	Disabled. Interrupt function control registers can be read or written to.
1	Enabled. Writing to interrupt function control registers is disabled. Reading is enabled. Guarded SFRs: IFxx, MKxx, PRxx, EGPx, EGNx

GCSC Note 2	Clock control function, voltage detection circuit, and RAM parity error detection function control register guard
0	Disabled. Registers to control port function, interrupt function, clock control function, voltage detection circuits, and RAM parity error detection function can be read or written to.
1	Enabled. Writing to registers to control port function, interrupt function, clock control function, voltage detection circuits, and RAM parity error detection function is disabled. Reading is enabled. Guarded SFRs: CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, RPECTL

Notes 1. Pxx (port registers) are not guarded.

2. Set GCSC to 0 for self/serial programming.

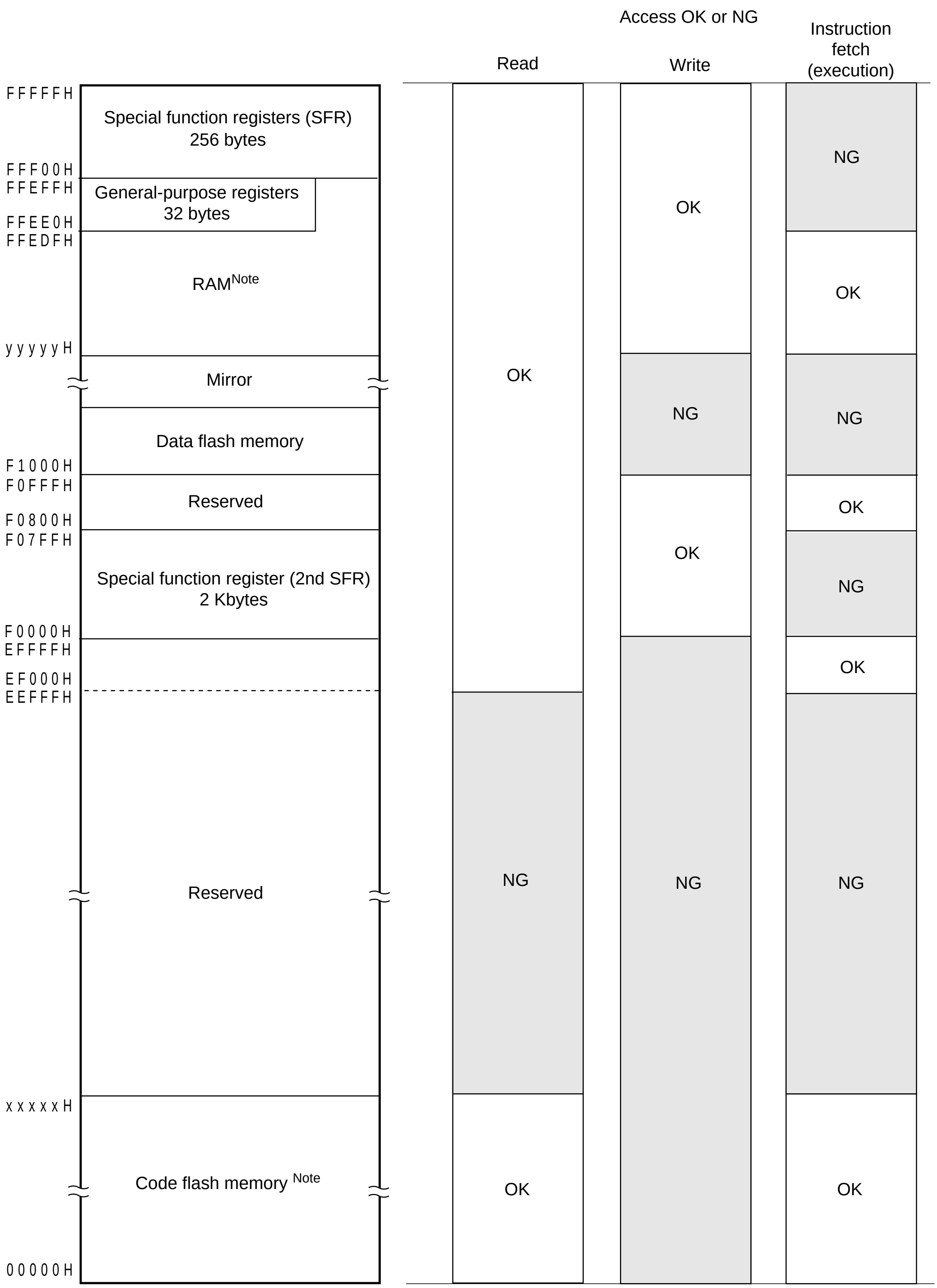
24.3.6 Invalid Memory Access Detection Function

The IEC 60730 standard requires verification of correct operations of the CPU and interrupts.

The invalid memory access detection function allows a reset to be generated when the specified invalid memory access detection space is accessed.

The space in which invalid memory access is to be detected is indicated as NG in figure 24-10.

Figure 24-10. Invalid Memory Access Detection Space



Note The addresses of the RAM and code flash memory are shown below according to the product type.

Product Type	Code Flash Memory (00000H to xxxxxH)	RAM (yyyyyH to FFFFFH)
R5F109xA	8192 × 8 bits (00000H to 01FFFH)	512 × 8 bits
R5F109xA (x = 6, A, B, G, L)	16384 × 8 bits (00000H to 03FFFH)	1024 × 8 bits
R5F109xB (x = 6, A, B, G, L)	24576 × 8 bits (00000H to 05FFFH)	1536 × 8 bits
R5F109xC (x = 6, A, B, G, L)	32768 × 8 bits (00000H to 07FFFH)	2048 × 8 bits (FF700H to FFEFFH)
R5F109xD (x = 6, A, B, G, L)	49152 × 8 bits (00000H to 0BFFFH)	3072 × 8 bits (FF300H to FFEFFH)
R5F109xE (x = 6, A, B, G, L)	65536 × 8 bits (00000H to 0FFFFH)	4096 × 8 bits (FEF00H to FFEFFH)

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.

The invalid memory access detection function uses the IAWEN bit.

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

Reset signal generation clears this register to 00H.

Figure 24-11. 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	Disables the detection of invalid memory access.
1	Enables the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is valid, and writing 0 to it after setting it to 1 is invalid.

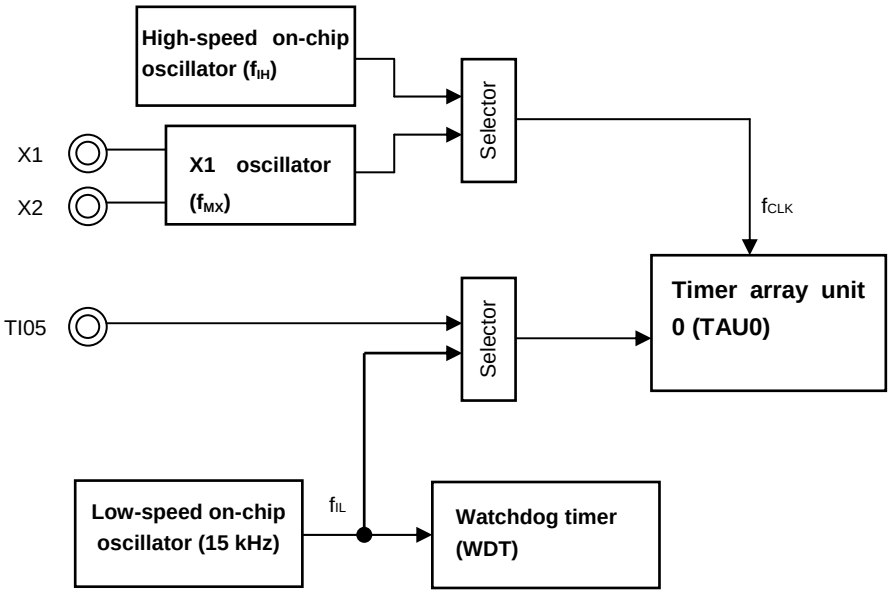
Remark When WDTON = 1 (watchdog timer operation enabled) for the option byte, the invalid memory access detection function is enabled even if IAWEN = 0.

24.3.7 Frequency Detection Function

The IEC 60730 standard requires verification of correct oscillation frequency.

With the frequency detection function, the high-speed on-chip oscillator clock or external X1 oscillator clock is compared with the low-speed on-chip oscillator clock (15 kHz), which allows detection of the clock operating at an abnormal frequency.

Figure 24-12. Configuration of Frequency Detection Function



<Operation summary>

The clock frequency is judged based on the result of pulse interval measurement carried under the following conditions.

- The high-speed on-chip oscillator clock (f_{IH}) or external X1 oscillator clock (f_{MX}) is selected as the CPU/peripheral hardware clock (f_{CLK}).
- The low-speed on-chip oscillator clock (f_{IL}: 15 kHz) is selected as the input to channel 5 of timer array unit 0 (TAU0).

If the pulse interval measurement result is abnormal, the clock frequency is determined to be abnormal. For pulse interval measurement, refer to 6.7.4, Operation as input pulse interval measurement.

24.3.7.1 Timer input select register 0 (TIS0)

This register is used to select the timer input of channel 5.

With the low-speed on-chip oscillator clock being selected for the timer input, measuring the selected clock pulses allows determining whether the ratio of low-speed on-chip oscillator clock to timer operation clock is appropriate.

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

Reset signal generation clears this register to 00H.

Figure 24-13. Format of Timer Input Select Register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Signal input to timer input pin (TI05)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than the above			Setting prohibited

24.3.8 A/D Test Function

The IEC 60730 standard requires an A/D converter to be tested. With the A/D test function, internal 0 V, AV_{REF}, internal reference voltage (1.45 V) are A/D-converted to verify correct A/D converter operation.

Correct operation of the analog multiplexer can be verified using the following procedure.

(1) Perform A/D conversion of ANIx pin (conversion result 1).

(2) Perform A/D conversion with AV_{REFM} being selected with the ADTES register, and adjust the potential difference at both ends of A/D converter sampling capacitor to 0 V.

(3) Perform A/D conversion of ANIx pin (conversion result 2).

(4) Perform A/D conversion with AV_{REFP} being selected with the ADTES register, and adjust the potential difference at both ends of A/D converter sampling capacitor to AV_{REF}.

(5) Perform A/D conversion of ANIx pin (conversion result 3).

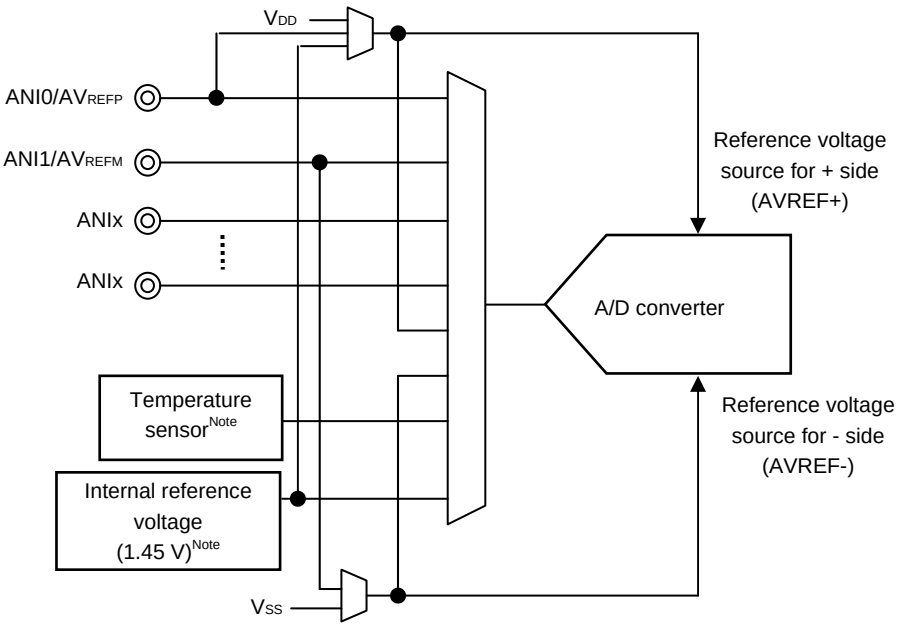
(6) Confirm that conversion results 1, 2, and 3 are identical.

With the above procedure, it can be confirmed that the analog multiplexer is selected and that there is no wire disconnection.

Remarks 1. When the variable analog voltage should be input during conversion in steps 1 through 5, a different method is necessary to check the analog multiplexer.

2. The conversion results include errors; take appropriate errors into consideration when comparing conversion results.

Figure 24-14. A/D Test Function Configuration



Note Can be selected only in HS (high-speed main) mode.

24.3.8.1 A/D test register (ADTES)

This register is used to select the AV_{REFP}, AV_{REFM}, or analog input channel (ANIxx) as the A/D conversion target, where AV_{REFP} and AV_{REFM} are reference voltages for the + and – sides, respectively.

When the A/D test function is used, set this register as follows.

- Select AV_{REFM} as the A/D conversion target to measure internal 0 V.
- Select AV_{REFP} as the A/D conversion target to measure AV_{REF}.

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

Reset signal generation clears this register to 00H.

Figure 24-15. 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/temperature sensor output[Note]/internal reference voltage output (1.45 V)[Note] (This is specified using the analog input channel specification register (ADS).)
1	0	AV_{REFM}
1	1	AV_{REFP}
Other than the above		Setting prohibited

Note Temperature sensor output and internal reference voltage output (1.45 V) can be selected only in HS (high-speed main) mode.

24.3.8.2 Analog input channel specification register (ADS)

This register is used to specify the analog voltage input channel to be A/D-converted. When the A/D test function is used to measure ANIxx, temperature sensor output, or internal reference voltage (1.45 V), set the A/D test register (ADTES) to 00H.

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 24-16. Format of A/D Input Channel Specification Register (ADS)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin
0	0	0	1	1	0	ANI6	P26/ANI6 pin
0	0	0	1	1	1	ANI7	P27/ANI7 pin
0	1	0	0	0	0	ANI16	P03/ANI16 pin
0	1	0	0	0	1	ANI17	P02/ANI17 pin
0	1	0	0	0	0	ANI18	P147/ANI18 pin
0	1	0	0	0	0	ANI19	P120/ANI19 pin
1	0	0	0	0	0	—	Temperature sensor output Note
1	0	0	0	0	1	—	Internal reference voltage output (1.45 V) Note
Other than the above						Setting prohibited	

Note Can be selected only in HS (high-speed main) mode.

Cautions 1. Set 0 in bits 5 and 6.

2. Before rewriting the ADISS bit, sure to stop the A/D conversion comparator (ADCE = 0 in A/D converter mode register 0 (ADM0)).

3. When AV_{REFP} is used as the reference voltage source for the + side (AV_{REF+}) of the A/D converter, do not select ANI0 as the A/D conversion channel.

4. When AV_{REFM} is used as the reference voltage source for the - side (AV_{REF-}) of the A/D converter, do not select ANI1 as the A/D conversion channel.

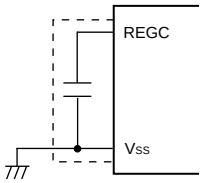
5. With ADISS = 1, the internal reference voltage (1.45 V) cannot be used for the reference voltage source for the + side (AV_{REF+}).

CHAPTER 25 REGULATOR

25.1 Regulator Overview

The RL78/F12 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 V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

The regulator output voltage is normally 2.1 V (typ.), and in the low consumption current mode, 1.8 V (typ.).



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

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 (except during OCD mode)
		When both the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during CPU operation with the subsystem clock (f_{XT})
		When both the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{XT}) has been set
Normal current mode	2.1 V	Other than above (including during on-chip debugging) 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 of the RL78/F12 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.

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.

26.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

(1) 000C0H/010C0H

- ○ Operation of watchdog timer
  - Operation is stopped or enabled in the HALT or STOP mode.
- ○ Setting of interval time of watchdog timer
- ○ Operation of watchdog timer
  - Operation is stopped or enabled.
- ○ Setting of window open period of watchdog timer
- ○ Setting of interval interrupt of watchdog timer
  - 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.
- ○ Setting of LVD detection level (V_{LVIH}, V_{LVIL}, V_{LVI})

Caution 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 threshold voltage (V_{IL})
  - V_{IL} = 0.5 V_{DD}
  - V_{IL} = 0.2 V_{DD}
- Setting of the frequency of the high-speed on-chip oscillator
  - Select from 1 MHz, 4 MHz, 8 MHz, 12 MHz, 16 MHz, 24 MHz, and 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
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2f_{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%
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	$2^{16}/f_{IL}$ (3799.19m s)

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

Notes 1. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

2. The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

Caution The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. The invalid memory access detection function is always enabled when WDTON = 1, regardless of the setting of the IAWEN bit.

Figure 26-2. Format of User Option Byte (000C1H/010C1H) (1/2)

Address: 000C1H/010C1H[Note 1]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- When used as interrupt & reset mode

Detection voltage		Option byte Setting Value						
V_{LVIL}	V_{LVIH}	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
1.84 V	+0.1 V	1	0	0	0	1	1	0
	+0.2 V						0	1
	+1.2 V						0	0
2.45 V	+0.1 V			0	1	0	1	0
	+0.2 V						0	1
	+1.2 V						0	0
2.75 V	+0.1 V			0	1	1	1	0
	+0.2 V						0	1
	+1.2 V						0	0
Other than above		Setting prohibited						

- When used as reset mode

Detection voltage	Option byte Setting Value						
V_{LVI} (= V_{LVIH})	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
LVDOFF	1	1	1	1	1	×	×
1.88 V			0	0	1	1	1
1.98 V			0	0	1	1	0
2.09 V			0	0	1	0	1
2.50 V			0	1	0	1	1
2.61 V			0	1	0	1	0
2.71 V			0	1	0	0	1
2.81 V			0	1	1	1	1
2.92 V			0	1	1	1	0
3.02 V			0	1	1	0	1
3.13 V			0	0	1	0	0
3.75 V			0	1	0	0	0
4.06 V			0	1	1	0	0
Other than above	Setting 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".

Remarks 1. ×: don't care

2. Referring to LVD setting, see **23.1 Functions of Voltage Detector**.

Figure 26-2. Format of User Option Byte (000C1H/010C1H) (2/2)

Address: 000C1H/010C1H[Note 1]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- When used as interrupt mode

Detection voltage	Option byte Setting Value						
V_{LVI} (= V_{LVIH})	LVIMDS1	LVIMDS0	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0
LVDOFF	0	1	1	1	1	×	×
1.88 V			0	0	1	1	1
1.98 V			0	0	1	1	0
2.09 V			0	0	1	0	1
2.50 V			0	1	0	1	1
2.61 V			0	1	0	1	0
2.71 V			0	1	0	0	1
2.81 V			0	1	1	1	1
2.92 V			0	1	1	1	0
3.02 V			0	1	1	0	1
3.13 V			0	0	1	0	0
3.75 V			0	1	0	0	0
4.06 V			0	1	1	0	0
Other than above	Setting 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".

Remarks 1. ×: don't care

2. Referring to LVD setting, see **23.1 Functions of Voltage Detector**.

Figure 26-3. Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H[Note]

7	6	5	4	3	2	1	0
1	CMODE0	ITHL	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE0	Setting of flash operation mode
0	LS (low speed main) mode
1	HS (high speed main) mode
Other than above	Setting prohibited

ITHL	Setting of threshold voltage (V_{IL})
0	V_{IL} = 0.5 V_{DD}
1	V_{IL} = 0.2 V_{DD}

Note The presence or absence of hysteresis characteristics depends on the setting of the ITHL bit.

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
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
1	0	1	1	4 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.

26.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 26-4. Format of On-chip Debug Option Byte (000C3H/010C3H)

Address: 000C3H/010C3H[Note]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Note Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

Caution Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

26.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the assembler or Cube Suite linker option, in addition to describing to the source. When doing so, the contents set by using the linker option take precedence, even if descriptions exist in the source, as mentioned below.

A software description example of the option byte setting is shown below.

```
OPT   CSEG        OPT_BYTE
      DB          36H     ; Does not use interval interrupt of watchdog timer,
                          ; Enables watchdog timer operation,
                          ; Window open period of watchdog timer is 50%,
                          ; Overflow time of watchdog timer is 2^9/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          7AH     ; Select 2.75 V for VLVIL
                          ; Select 2.85 V for VLVIH
                          ; Select the interrupt & reset mode as the LVD operation mode
      DB          8DH     ; Select the LS (low speed main) mode as the flash operation mode,
                          ; threshold voltage (VIL) = 0.5 VDD,
                          ; and 1 MHz as the frequency of the high-speed on-chip oscillator
      DB          85H     ; Enables on-chip debug operation, does not erase flash memory
                          ; data when security ID authorization fails
```

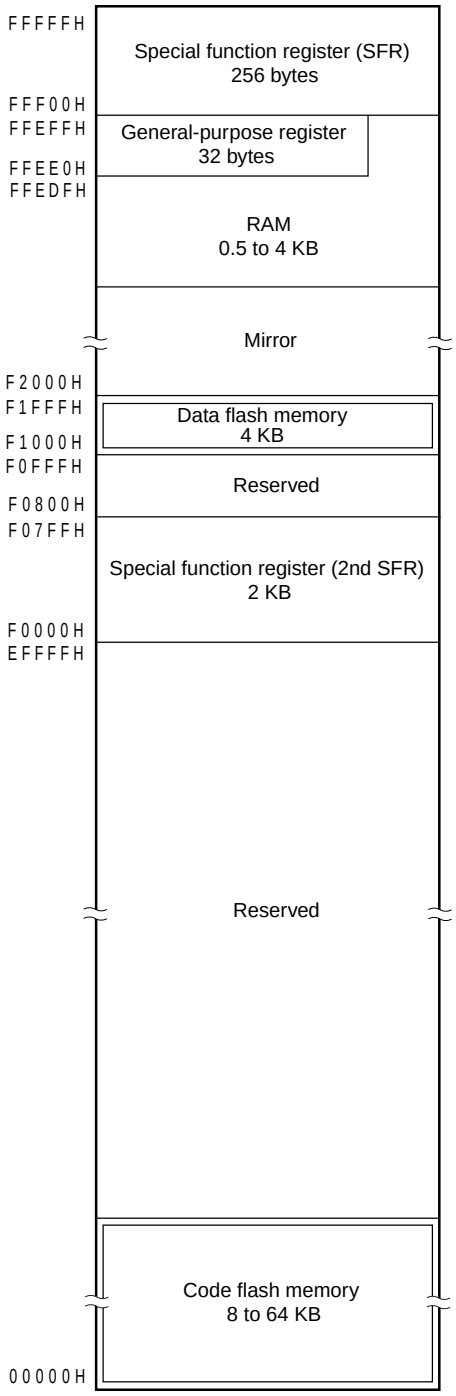
When the boot swap function is used during self programming, 000C0H to 000C3H is switched to 010C0H to 010C3H. Describe to 010C0H to 010C3H, therefore, the same values as 000C0H to 000C3H as follows.

```
OPT2  CSEG   AT   010C0H
      DB          36H     ; Does not use interval interrupt of watchdog timer,
                          ; Enables watchdog timer operation,
                          ; Window open period of watchdog timer is 50%,
                          ; Overflow time of watchdog timer is 2^10/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          7AH     ; Select 2.75 V for VLVIL
                          ; Select 2.85 V for VLVIH
                          ; Select the interrupt & reset mode as the LVD operation mode
      DB          8DH     ; Select the LS (low speed main) mode as the flash operation mode,
                          ; threshold voltage (VIL) = 0.5 VDD,
                          ; and 1 MHz as the frequency of the high-speed on-chip oscillator
      DB          85H     ; Enables on-chip debug operation, does not erase flash memory
                          ; data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 010C0H to 010C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 27 FLASH MEMORY

The RL78/F12 incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board. The flash memory includes the “code flash memory”, in which programs can be executed, and the “data flash memory”, an area for storing data.



The following three methods for programming the flash memory are available:

- Writing to flash memory by using flash memory programmer (see 27.1)
- Writing to flash memory by using external device (that Incorporates UART) (see 27.2)
- Self-programming (see 27.7)

27.1 Writing to Flash Memory by Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78/F12.

- 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/F12 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/F12 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/F12 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.				
Signal Name		I/O	Pin Function		20-pin	30-pin	32-pin	48-pin	64-pin
PG-FP5, FL-PR5	E1 on-chip debugging emulator				SSOP	SSOP	WQFN (5x5)	LQFP (7x7), WQFN (7x7)	LQFP (10x10)
–	TOOL0	I/O	Transmit/receive signal	TOOL0/P40	3	5	1	39	5
SI/RxD	–	I/O	Transmit/receive signal						
SCK	–	Output	–	–	–	–	–	–	–
CLK	–	Output	–	–	–	–	–	–	–
–	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	4	6	2	40	6
/RESET	–	Output							
FLMD0	–	Output	Mode signal	–	–	–	–	–	–
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	10	12	8	48	15
GND		–	Ground	V_{SS}	9	11	7	47	13
				EV_{SS}	–	–	–	–	14
				REGC Note	8	10	6	46	12
EMV_{DD}		–	Driving power for TOOL pin	V_{DD}	10	12	8	48	16
				EV_{DD}	–	–	–	–	16

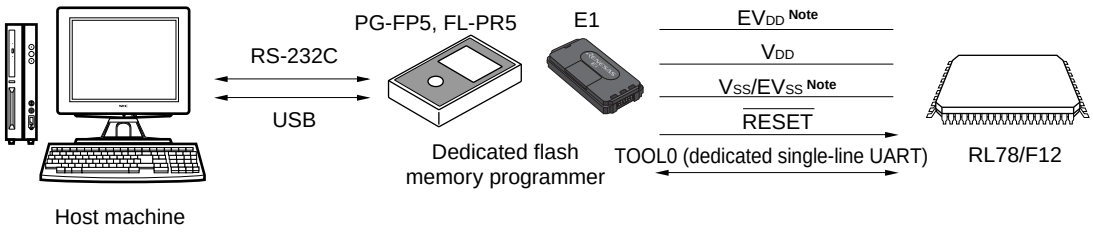
Note Connect REGC pin to ground via a capacitor (default: 0.47 μF).

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/F12 is illustrated below.

Figure 27-1. Environment for Writing Program to Flash Memory



Note 64-pin, 80-pin, 100-pin and 128-pin products only.

A host machine that controls the dedicated flash memory programmer is necessary.

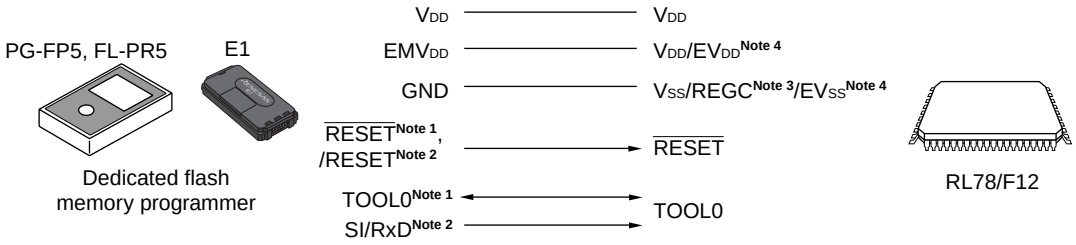
To interface between the dedicated flash memory programmer and the RL78/F12, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

27.1.2 Communication Mode

Communication between the dedicated flash memory programmer and the RL78/F12 is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78/F12.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 27-2. Communication with Dedicated Flash Memory Programmer



Notes 1. When using E1 on-chip debugging emulator.

2. When using PG-FP5 or FL-PR5.

3. Connect REGC pin to ground via a capacitor (default: 0.47 μF).

4. 64-pin products only.

The dedicated flash memory programmer generates the following signals for the RL78/F12. 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/F12	Connection
Signal Name		I/O	Pin Function	Pin Name	
PG-FP5, FL-PR5	E1 on-chip debugging emulator				
FLMD0	–	Output	Mode signal	–	×
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}	◎
GND		–	Ground	V_{SS}, EV_{SS}, REGC Note	◎
EMV_{DD}		–	Driving power for TOOL pin	V_{DD}, EV_{DD}	◎
CLK	–	Output	Clock output	–	×
/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		
SCK	–	Output	Transfer clock	–	×

Note Connect REGC pin to ground via a capacitor (default: 0.47 μF).

Caution Make EV_{DD} the same potential as V_{DD}.

Remark ◎: Be sure to connect the pin.
×: The pin does not have to be connected.

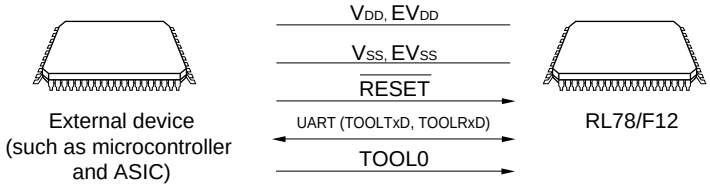
27.2 Writing to Flash Memory by Using External Device (that Incorporates UART)

On-board data writing to the internal flash memory is possible by using the RL78/F12 and an external device (a microcontroller or ASIC) connected to a UART.

27.2.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78/F12 is illustrated below.

Figure 27-3. Environment for Writing Program to Flash Memory



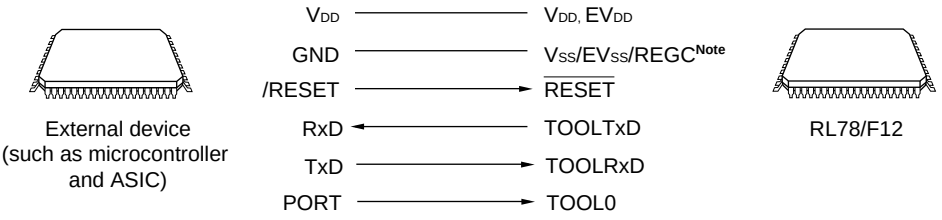
Processing to write data to or delete data from the RL78/F12 by using an external device is performed on-board. Off-board writing is not possible.

27.2.2 Communication Mode

Communication between the external device and the RL78/F12 is established by serial communication using the TOOLTxD and TOOLRxD pins via UART0 of the serial array unit of the RL78/F12.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 27-4. Communication with External Device



Note Connect REGC pin to ground via a capacitor (default: 0.47 μF).

Caution Make EV_{DD} the same potential as V_{DD}.

The external device generates the following signals for the RL78/F12.

Table 27-3. Pin Connection

External Device			RL78/F12	Connection
Signal Name	I/O	Pin Function	Pin Name	
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, EV_{DD}	◎
GND	–	Ground	V_{SS}, EV_{SS}, REGC Note	◎
CLK	Output	Clock output	–	×
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$	◎
RxD	Input	Receive signal	TOOL0TxD	◎
TxD	Output	Transmit signal	TOOL0RxD	◎
PORT	Output	Mode signal	TOOL0	◎
SCK	Output	Transfer clock	–	×

Note Connect REGC pin to ground via a capacitor (default: 0.47 μF).

Caution Make EV_{DD} the same potential as V_{DD}.

Remark ◎: Be sure to connect the pin.
×: The pin does not have to be connected.

27.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

27.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 kΩ pull-up resistor. When this pin is used as the port pin, use that by the following method. When used as an input pin: Input of 1 ms or more width low-level is prohibited after pin reset release. Furthermore, 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.

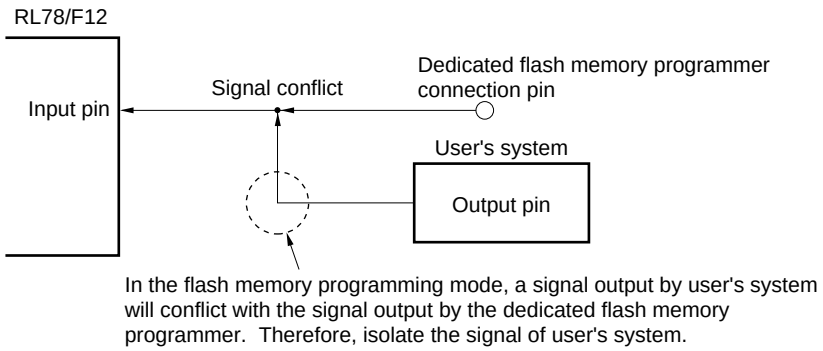
Remark The SAU and IICA pins are not used for communication between the RL78/F12 and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

27.3.2 $\overline{\text{RESET}}$ 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 27-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



27.3.3 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 to V_{DD}, or V_{SS}, via a resistor.

27.3.4 REGC pin

Connect the REGC pin to GND via a capacitor (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.

27.3.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{IH}) is used.

27.3.6 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

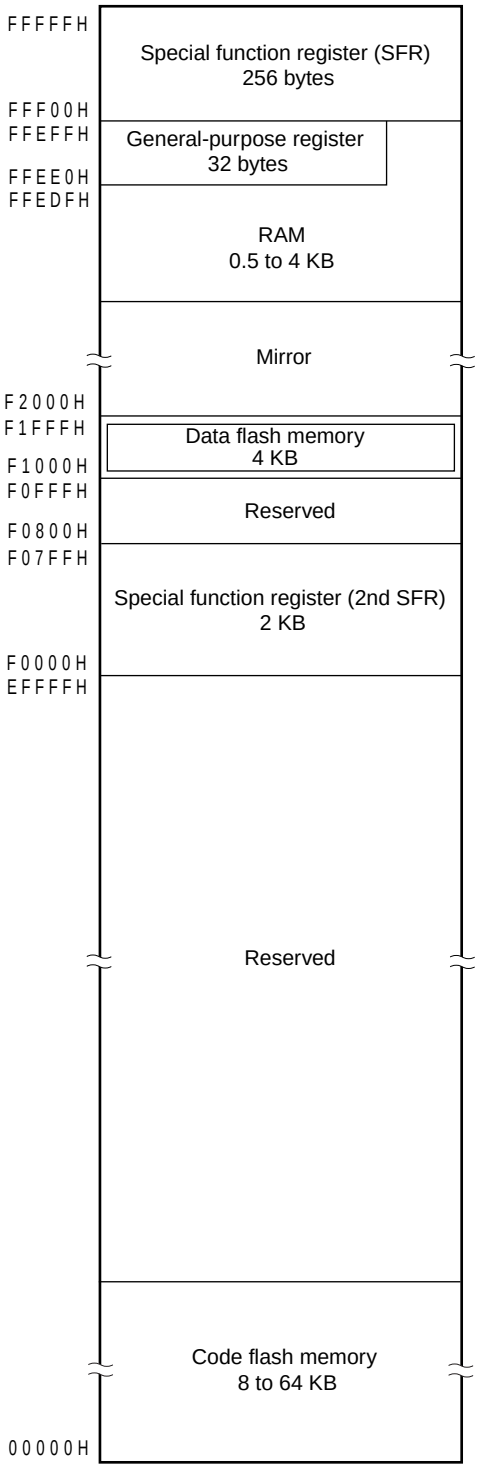
To use the on-board supply voltage, connect in compliance with the normal operation mode.

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.

27.4 Data Flash

27.4.1 Data flash overview

In addition to 8 to 64 KB of code flash memory, the RL78/F12 includes 4 KB of data flash memory for storing data.



An overview of the data flash memory is provided below.

- The data flash memory can be written to by using the flash memory programmer or an external device
- Programming is performed in 8-bit units
- Blocks can be deleted in 1 KB units
- The only access by CPU instructions is byte reading (reading: four clock cycles)
- Because the data flash memory is an area exclusively used for data, it cannot be used to execute instructions (code fetching)
- Instructions can be executed from the code flash memory while rewriting the data flash memory (That is, dual operation is supported)
- Accessing the data flash memory is not possible while rewriting the code flash memory (such as during self programming)
- Because the data flash memory is stopped after a reset ends, the data flash control register (DFLCTL) must be set up in order to use the data flash memory
- Manipulating the DFLCTL register is not possible while rewriting the data flash memory
- Transition to the HALT/STOP status is not possible while rewriting the data flash memory
- Programming of data flash memory is possible while the program is being run by Renesas' library.

27.4.2 Register controlling data flash memory

(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 27-6. 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.

27.4.3 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.
The time setup takes differs for each main clock mode.
<Setup time for each main clock mode>
- HS (high-speed main) mode: 5 μs
- LS (low-speed main) mode: 720 ns

<3> After the wait, the data flash memory can be accessed.

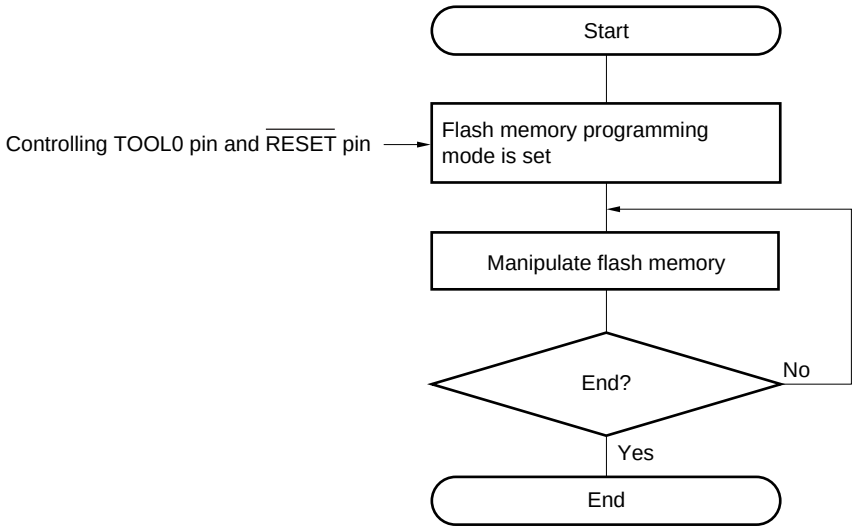
Cautions 1. Accessing the data flash memory is not possible during the setup time.
2. Before executing a STOP instruction during the setup time, temporarily clear DFLEN to 0.
3. Be sure to set the HIOSTOP bit in the CSC register to 0 when the CPU operates with the clock other than the high-speed on-chip oscillator clock.
<R> **4. The data flash should be read in either of following ways.**
- **Use the flash library provided by Renesas (EEL (Pack01) version V1.13 or later).**
- **Stop the DMA transfer before reading.**

27.5 Programming Method

27.5.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 27-7. Flash Memory Manipulation Procedure



27.5.2 Flash memory programming mode

To rewrite the contents of the flash memory, set the RL78/F12 in the flash memory programming mode. To enter the mode, set as follows.

<When programming by using the dedicated flash memory programmer>

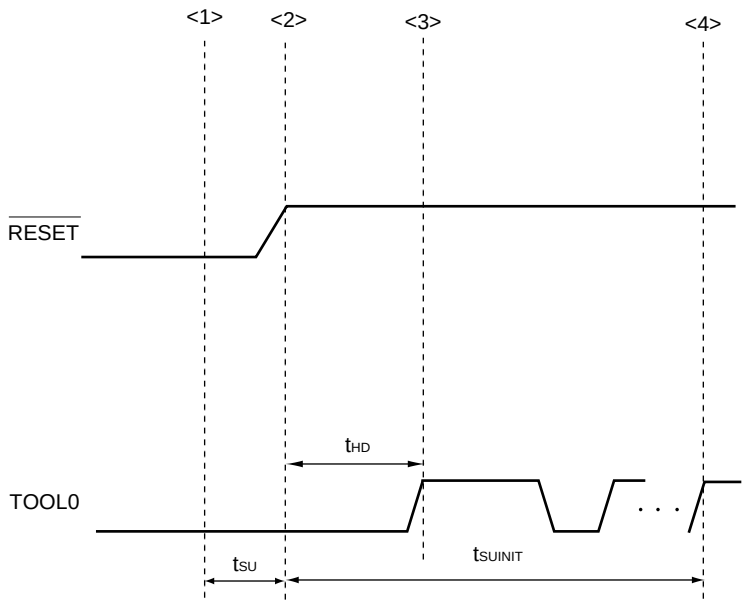
Set the TOOL0 pin to the low level, and then cancel the reset. Next, communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

<When programming by using an external device>

Set the TOOL0 pin to the low level, and then cancel the reset. Keep the TOOL0 pin at the low level for at least 1 ms after the reset ends, and then use UART communication to send the data “00H” from the external device. Complete UART communication within 100 ms after the reset ends.

When performing on-board writing, either switch the mode by using jumpers or perform pin processing in advance so that it will be okay if the flash memory programming mode is switched to (For details, see **27.3 Connection of Pins on Board**).

Figure 27-8. Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The pins reset ends (POR and LVD reset must end before the pin 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 and internal 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 and internal resets end.

Table 27-4. Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0	Flash memory programming mode

There are two flash memory programming modes for which the voltage range in which to write, erase, or verify data differs.

Table 27-5. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Mode	Voltages at which data can be written, erased, or verified
Wide voltage mode	1.8 V to 5.5 V
Full speed mode Note	2.7 V to 5.5 V

Note This can only be specified if the CMODE0 bit is 1.

Specify the mode that corresponds to the voltage range in which to write data. When programming by using the dedicated flash memory programmer, the mode is automatically selected by the voltage setting on GUI.

Remarks 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, deletion, or verification.

2. For details about communication commands, see **27.5.4 Communication commands**.

27.5.3 Selecting communication mode

Communication mode of the RL78/F12 as follows.

Table 27-6. Communication Modes

Communication Mode	Standard Setting Note 1				Pins Used
	Port	Speed Note 2	Frequency	Multiply Rate	
1-line mode (when flash memory programmer is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOL0
UART0 (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOLTxD, TOOLRxD

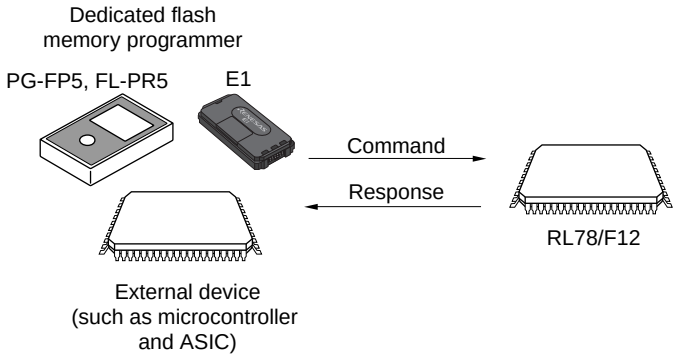
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

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.

27.5.4 Communication commands

The RL78/F12 communicates with the dedicated flash memory programmer or external device by using commands. The signals sent from the flash memory programmer or external device to the RL78/F12 are called commands, and the signals sent from the RL78/F12 to the dedicated flash memory programmer or external device are called response.

Figure 27-9. Communication Commands



The flash memory control commands of the RL78/F12 are listed in the table below. All these commands are issued from the programmer or external device, and the RL78/F12 perform processing corresponding to the respective commands.

Table 27-7. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Write	Programming	Writes data to a specified area in the flash memory.
Getting information	Silicon Signature	Gets the RL78/F12 information (such as the part number and flash memory configuration).
	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.

The RL78/F12 returns a response for the command issued by the dedicated flash memory programmer or external device. The response names sent from the RL78/F12 are listed below.

Table 27-8. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

27.6 Security Settings

The RL78/F12 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. The security setting is valid when the programming mode is set next.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board 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 on-board/off-board programming. However, blocks can be written by means of self programming.

- 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 on-board/off-board programming and self programming. Each security setting can be used in combination.

Table 27-9 shows the relationship between the erase and write commands when the RL78/F12 security function is enabled.

Remark To prohibit writing and erasing during self-programming, use the flash sealed window function (see **27.7.2** for detail).

Table 27-9. Relationship Between Enabling Security Function and Command

(1) During on-board/off-board 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 sealed window function (see **27.7.2** for detail).

Table 27-10. Setting Security in Each Programming Mode

(1) On-board/off-board 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		Execute security release command
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

Caution The security release command can be applied 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) 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		Execute security release command during on-board/off-board programming (cannot be disabled during self programming)
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

27.7 Flash Memory Programming by Self-Programming

The RL78/F12 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/F12 self-programming library, it can be used to upgrade the program in the field.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock. Be sure to set the HIOSTOP bit in the CSC register to 0 when using the self-programming function if the CPU operates with the main system clock.

2. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the 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 self-programming library.

3. Do not transit to standby mode during self programming.

Remarks 1. For details of the self-programming function and the RL78/F12 self-programming library, refer to **RL78 Microcontroller Self Programming Library Type01 User's Manual (R01AN0350E).**

2. For details of the time required to execute self programming, see the notes on use that accompany the flash self programming library tool.

Similar to when writing data by using the flash memory programmer, there are two flash memory programming modes for which the voltage range in which to write, erase, or verify data differs.

Table 27-11. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Mode	Voltages at which data can be written, erased, or verified	Writing Clock Frequency
Wide voltage mode	1.8 V to 5.5 V	8 MHz (MAX.)
Full speed mode Note	2.7 V to 5.5 V	32 MHz (MAX.)

Note This can only be specified if the CMODE0 bits bit is 1.

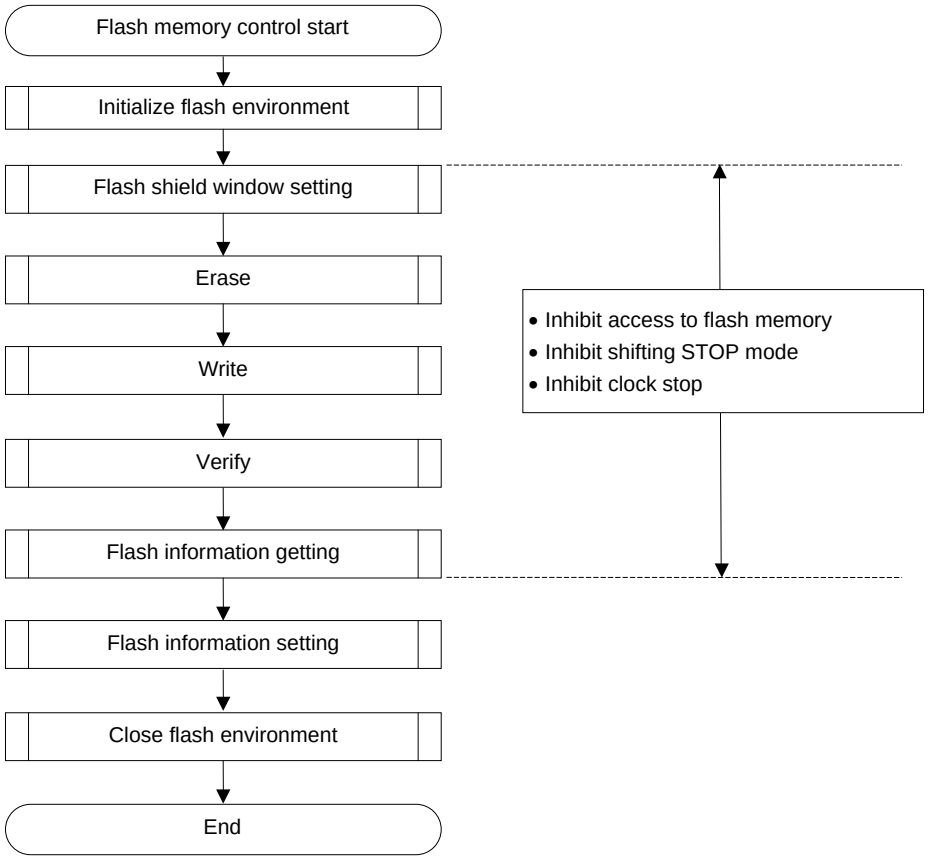
Specify the mode that corresponds to the voltage range in which to write data. If the argument fsl_flash_voltage_u08 is other than 00H when the FSL_Init function of the self programming library provided by Renesas Electronics is executed, wide-voltage mode is specified. If the argument is 00H, full-speed mode is specified.

Remarks 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, deletion, or verification.

2. For details of the self-programming function and the RL78/F12 self-programming library, refer to **RL78 Microcontroller Self Programming Library Type01 User's Manual (R01AN0350E)**.

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

Figure 27-10. Flow of Self Programming (Rewriting Flash Memory)



27.7.1 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.

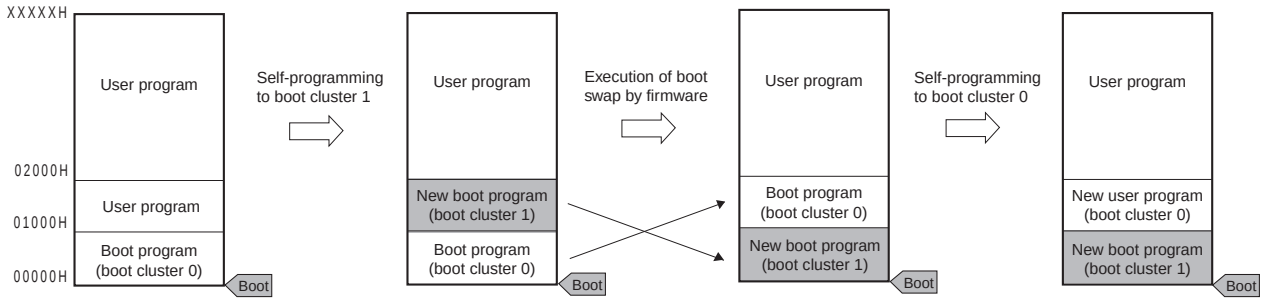
The boot swap function is used to avoid this problem.

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/F12, 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 boot programming 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 27-11. Boot Swap Function

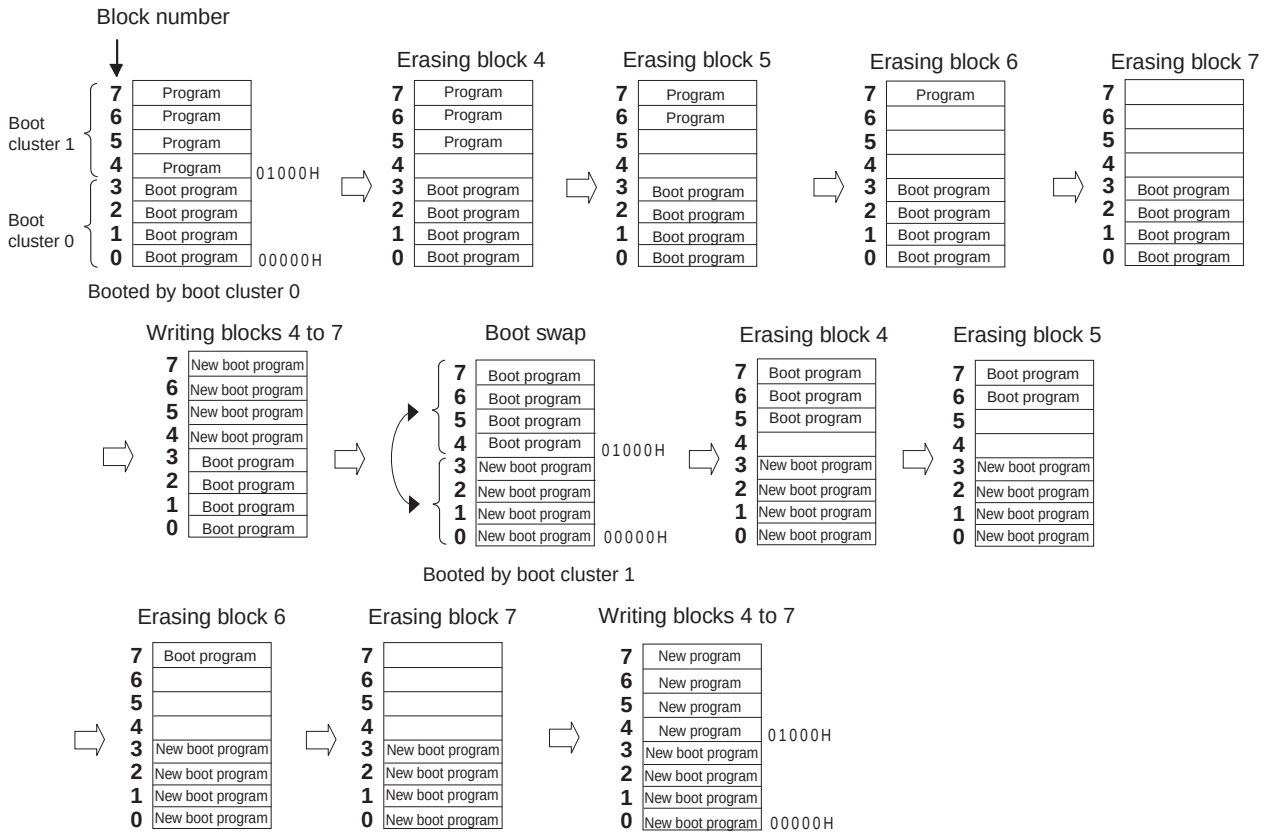


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 27-12. Example of Executing Boot Swapping



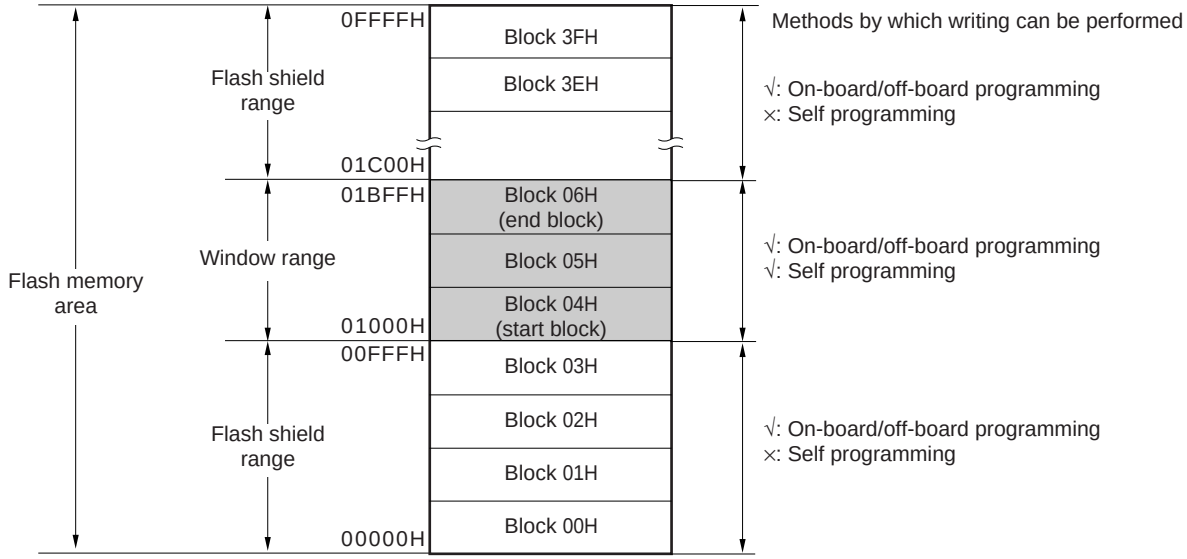
27.7.2 Flash shield window function

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 on-board/off-board programming and self programming.

Writing to and erasing areas outside the window range are disabled during self programming. During on-board/off-board programming, however, areas outside the range specified as a window can be written and erased.

Figure 27-13. Flash Shield Window Setting Example
(Target Devices: R5F109xE (x = 6, A, B, G, L), Start Block: 04H, End Block: 06H)



Cautions 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.

2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 27-12. 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 set information library.	Block erasing is enabled only within the window range.	Writing is enabled only within the range of window range.
On-board/Off-board 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 on-board/off-board programming.

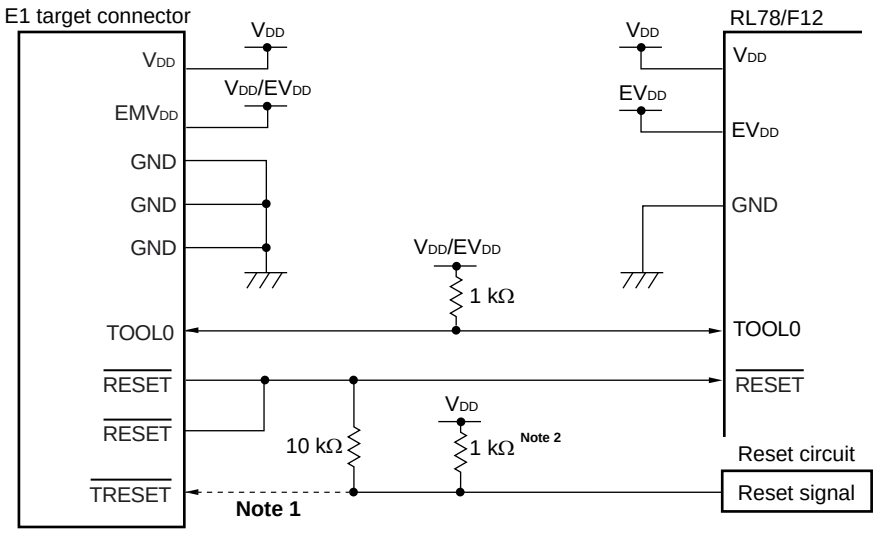
CHAPTER 28 ON-CHIP DEBUG FUNCTION

28.1 Connecting E1 On-chip Debugging Emulator to RL78/F12

The RL78/F12 uses the V_{DD}, $\overline{RESET}$, TOOL0, and V_{SS} 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/F12 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 and RL78/F12



Notes 1. Connecting the dotted line is not necessary during flash programming.

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/F12 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

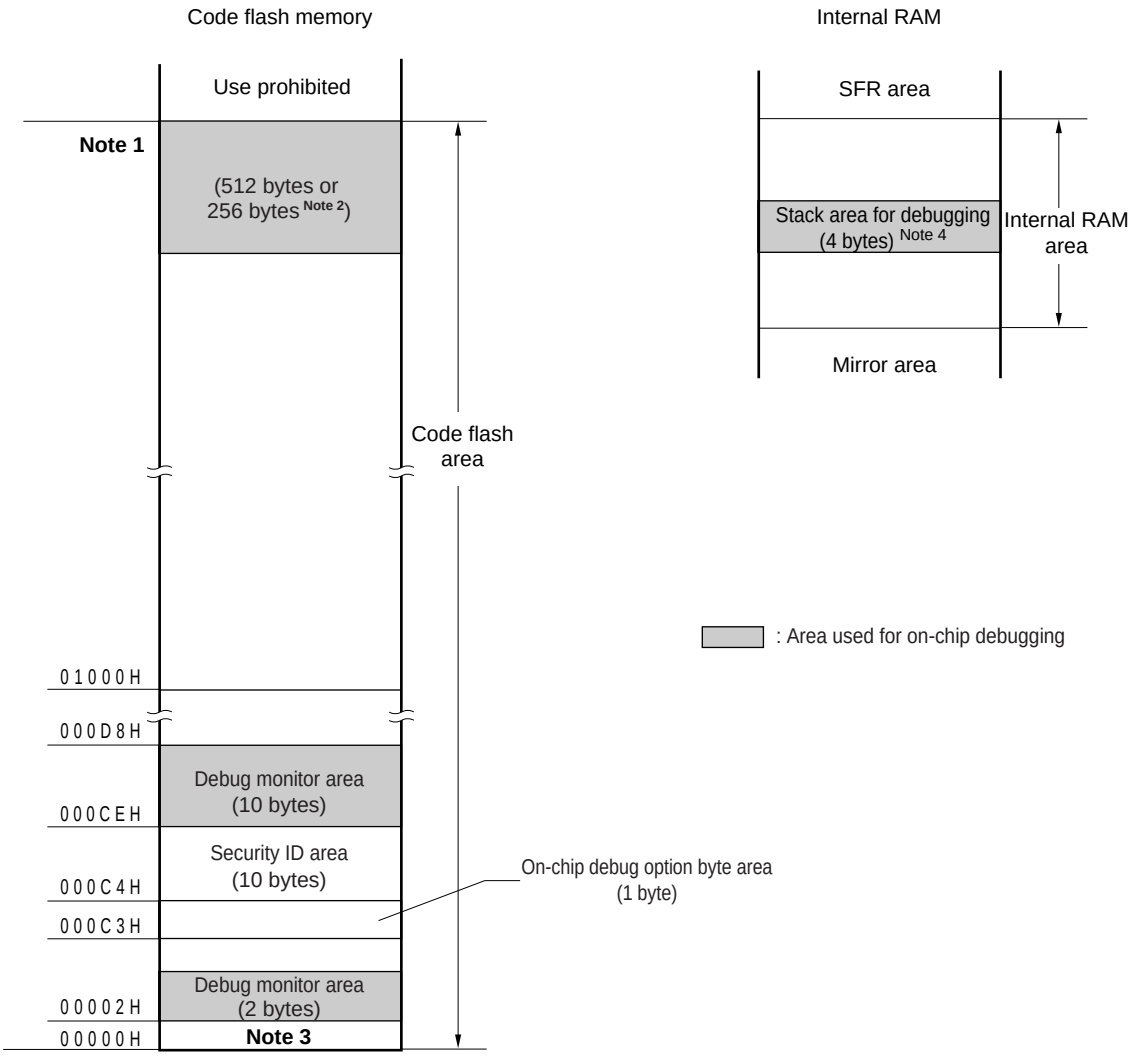
To perform communication between the RL78/F12 and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using linker 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



Notes 1. Address differs depending on products as follows.

Products (code flash memory capacity)	Address of **Note 1**
R5F10968	01FFFH
R5F109xA (x = 6, A, B, G, L)	03FFFH
R5F109xB (x = 6, A, B, G, L)	05FFFH
R5F109xC (x = 6, A, B, G, L)	07FFFH
R5F109xD (x = 6, A, B, G, L)	0BFFFH
R5F109xE (x = 6, A, B, G, L)	0FFFFH

2. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

3. In debugging, reset vector is rewritten to address allocated to a monitor program.

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, 12 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)

(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** (R01US0015E).

Remark The shaded parts of the tables in **Table 30-5 Operation List** indicate the operation or instruction format that is newly added for the RL78 microcontrollers.

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 **Table 3-5 SFR List** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Table 3-6 Extended 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 DMA 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 (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	–	r ← byte			
		saddr, #byte	3	1	–	(saddr) ← byte			
		sfr, #byte	3	1	–	sfr ← byte			
		!addr16, #byte	4	1	–	(addr16) ← byte			
		A, r Note 3	1	1	–	A ← r			
		r, A Note 3	1	1	–	r ← A			
		A, saddr	2	1	–	A ← (saddr)			
		saddr, A	2	1	–	(saddr) ← A			
		A, sfr	2	1	–	A ← sfr			
		sfr, A	2	1	–	sfr ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		!addr16, A	3	1	–	(addr16) ← A			
		PSW, #byte	3	3	–	PSW ← byte	×	×	×
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	×	×	×
		ES, #byte	2	1	–	ES ← byte			
		ES, saddr	3	1	–	ES ← (saddr)			
		A, ES	2	1	–	A ← ES			
		ES, A	2	1	–	ES ← A			
		CS, #byte	3	1	–	CS ← byte			
		A, CS	2	1	–	A ← CS			
		CS, A	2	1	–	CS ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	–	(DE) ← A			
		[DE + byte], #byte	3	1	–	(DE + byte) ← byte			
		A, [DE + byte]	2	1	4	A ← (DE + byte)			
		[DE + byte], A	2	1	–	(DE + byte) ← A			
		A, [HL]	1	1	4	A ← (HL)			
		[HL], A	1	1	–	(HL) ← A			
		[HL + byte], #byte	3	1	–	(HL + byte) ← byte			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, [HL + byte]	2	1	4	A ← (HL + byte)			
		[HL + byte], A	2	1	–	(HL + byte) ← A			
		A, [HL + B]	2	1	4	A ← (HL + B)			
		[HL + B], A	2	1	–	(HL + B) ← A			
		A, [HL + C]	2	1	4	A ← (HL + C)			
		[HL + C], A	2	1	–	(HL + C) ← A			
		word[B], #byte	4	1	–	(B + word) ← byte			
		A, word[B]	3	1	4	A ← (B + word)			
		word[B], A	3	1	–	(B + word) ← A			
		word[C], #byte	4	1	–	(C + word) ← byte			
		A, word[C]	3	1	4	A ← (C + word)			
		word[C], A	3	1	–	(C + word) ← A			
		word[BC], #byte	4	1	–	(BC + word) ← byte			
		A, word[BC]	3	1	4	A ← (BC + word)			
		word[BC], A	3	1	–	(BC + word) ← A			
		[SP + byte], #byte	3	1	–	(SP + byte) ← byte			
		A, [SP + byte]	2	1	–	A ← (SP + byte)			
		[SP + byte], A	2	1	–	(SP + byte) ← A			
		B, saddr	2	1	–	B ← (saddr)			
		B, !addr16	3	1	4	B ← (addr16)			
		C, saddr	2	1	–	C ← (saddr)			
		C, !addr16	3	1	4	C ← (addr16)			
		X, saddr	2	1	–	X ← (saddr)			
		X, !addr16	3	1	4	X ← (addr16)			
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte			
		A, ES:!addr16	4	2	5	A ← (ES, addr16)			
		ES:!addr16, A	4	2	–	(ES, addr16) ← A			
		A, ES:[DE]	2	2	5	A ← (ES, DE)			
		ES:[DE], A	2	2	–	(ES, DE) ← A			
		ES:[DE + byte],#byte	4	2	–	((ES, DE) + byte) ← byte			
		A, ES:[DE + byte]	3	2	5	A ← ((ES, DE) + byte)			
		ES:[DE + byte], A	3	2	–	((ES, DE) + byte) ← A			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, ES:[HL]	2	2	5	A ← (ES, HL)	
		ES:[HL], A	2	2	–	(ES, HL) ← A	
		ES:[HL + byte],#byte	4	2	–	((ES, HL) + byte) ← byte	
		A, ES:[HL + byte]	3	2	5	A ← ((ES, HL) + byte)	
		ES:[HL + byte], A	3	2	–	((ES, HL) + byte) ← A	
		A, ES:[HL + B]	3	2	5	A ← ((ES, HL) + B)	
		ES:[HL + B], A	3	2	–	((ES, HL) + B) ← A	
		A, ES:[HL + C]	3	2	5	A ← ((ES, HL) + C)	
		ES:[HL + C], A	3	2	–	((ES, HL) + C) ← A	
		ES:word[B], #byte	5	2	–	((ES, B) + word) ← byte	
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)	
		ES:word[B], A	4	2	–	((ES, B) + word) ← A	
		ES:word[C], #byte	5	2	–	((ES, C) + word) ← byte	
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)	
		ES:word[C], A	4	2	–	((ES, C) + word) ← A	
		ES:word[BC], #byte	5	2	–	((ES, BC) + word) ← byte	
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)	
		ES:word[BC], A	4	2	–	((ES, BC) + word) ← A	
		B, ES:!addr16	4	2	5	B ← (ES, addr16)	
		C, ES:!addr16	4	2	5	C ← (ES, addr16)	
		X, ES:!addr16	4	2	5	X ← (ES, addr16)	
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	–	A ←→ r	
		A, saddr	3	2	–	A ←→ (saddr)	
		A, sfr	3	2	–	A ←→ sfr	
		A, !addr16	4	2	–	A ←→ (addr16)	
		A, [DE]	2	2	–	A ←→ (DE)	
		A, [DE + byte]	3	2	–	A ←→ (DE + byte)	
		A, [HL]	2	2	–	A ←→ (HL)	
		A, [HL + byte]	3	2	–	A ←→ (HL + byte)	
		A, [HL + B]	2	2	–	A ←→ (HL + B)	
		A, [HL + C]	2	2	–	A ←→ (HL + C)	

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	XCH	A, ES:!addr16	5	3	–	A ←→ (ES, addr16)			
		A, ES:[DE]	3	3	–	A ←→ (ES, DE)			
		A, ES:[DE + byte]	4	3	–	A ←→ ((ES, DE) + byte)			
		A, ES:[HL]	3	3	–	A ←→ (ES, HL)			
		A, ES:[HL + byte]	4	3	–	A ←→ ((ES, HL) + byte)			
		A, ES:[HL + B]	3	3	–	A ←→ ((ES, HL) + B)			
		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			
		saddr	2	1	–	(saddr) ← 01H			
		!addr16	3	1	–	(addr16) ← 01H			
		ES:!addr16	4	2	–	(ES, addr16) ← 01H			
	CLRB	A	1	1	–	A ← 00H			
		X	1	1	–	X ← 00H			
		B	1	1	–	B ← 00H			
		C	1	1	–	C ← 00H			
		saddr	2	1	–	(saddr) ← 00H			
		!addr16	3	1	–	(addr16) ← 00H			
		ES:!addr16	4	2	–	(ES,addr16) ← 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, saddrp	2	1	–	AX ← (saddrp)			
		saddrp, AX	2	1	–	(saddrp) ← AX			
		AX, sfrp	2	1	–	AX ← sfrp			
		sfrp, AX	2	1	–	sfrp ← AX			
		AX, rp Note 3	1	1	–	AX ← rp			
		rp, AX Note 3	1	1	–	rp ← AX			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except rp = AX

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	–	(addr16) ← AX			
		AX, [DE]	1	1	4	AX ← (DE)			
		[DE], AX	1	1	–	(DE) ← AX			
		AX, [DE + byte]	2	1	4	AX ← (DE + byte)			
		[DE + byte], AX	2	1	–	(DE + byte) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	–	(HL) ← AX			
		AX, [HL + byte]	2	1	4	AX ← (HL + byte)			
		[HL + byte], AX	2	1	–	(HL + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	–	(B + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	–	(C + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	–	(BC + word) ← AX			
		AX, [SP + byte]	2	1	–	AX ← (SP + byte)			
		[SP + byte], AX	2	1	–	(SP + byte) ← AX			
		BC, saddrp	2	1	–	BC ← (saddrp)			
		BC, !addr16	3	1	4	BC ← (addr16)			
		DE, saddrp	2	1	–	DE ← (saddrp)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		HL, saddrp	2	1	–	HL ← (saddrp)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	–	(ES, DE) ← AX			
		AX, ES:[DE + byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE + byte], AX	3	2	–	((ES, DE) + byte) ← AX			
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)			
		ES:[HL], AX	2	2	–	(ES, HL) ← AX			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, ES:[HL + byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL + byte], AX	3	2	–	((ES, HL) + byte) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	–	((ES, BC) + word) ← AX			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
	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, saddr	2	1	–	A, CY ← A + (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A,CY ← A + ((ES, HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A,CY ← A + ((ES, HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except rp = AX

4. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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, saddr	2	1	–	A, CY ← A + (saddr) + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C) + CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A + (ES, HL) + CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A + ((ES, HL) + B) + 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, saddr	2	1	–	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A – ((ES:HL) + C)	×	×	×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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, saddr	2	1	–	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16) – CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL) – CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte) – CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B) – 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, saddr	2	1	–	A ← A ∧ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∧ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∧ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ∧ ((ES:HL) + C)	×		

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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, saddr	2	1	–	A ← A ∨ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∨ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∨ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∨ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∨ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∨ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∨ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∨ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∨ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∨ ((ES:HL) + B)	×		
		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, saddr	2	1	–	A ← A ⊻ (saddr)	×		
		A, !addr16	3	1	4	A ← A ⊻ (addr16)	×		
		A, [HL]	1	1	4	A ← A ⊻ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ⊻ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ⊻ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ⊻ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ⊻ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ⊻ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ⊻ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ⊻ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ⊻ ((ES:HL) + C)	×		

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	–	A – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A – (HL + C)	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A – ((ES:HL) + C)	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
	CMP0	A	1	1	–	A – 00H	×	×	×
		X	1	1	–	X – 00H	×	×	×
		B	1	1	–	B – 00H	×	×	×
		C	1	1	–	C – 00H	×	×	×
		saddr	2	1	–	(saddr) – 00H	×	×	×
		!addr16	3	1	4	(addr16) – 00H	×	×	×
		ES:!addr16	4	2	5	(ES:addr16) – 00H	×	×	×
	CMPS	X, [HL + byte]	3	1	4	X – (HL + byte)	×	×	×
		X, ES:[HL + byte]	4	2	5	X – ((ES:HL) + byte)	×	×	×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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, saddrp	2	1	–	AX, CY ← AX + (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX + (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	×	×	×
		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, saddrp	2	1	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	×	×	×
		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, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL) + byte)	×	×	×
Multiply	MULU	X	1	1	–	AX ← A × X			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/ decrement	INC	r	1	1	–	r ← r + 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) + 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) + 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) + 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) + 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) + 1	×	×	
	DEC	r	1	1	–	r ← r – 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) – 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) – 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) – 1	×	×	
	INCW	rp	1	1	–	rp ← rp + 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) + 1			
		!addr16	3	2	–	(addr16) ← (addr16) + 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) + 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) + 1			
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) + 1			
	DECW	rp	1	1	–	rp ← rp – 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) – 1			
		!addr16	3	2	–	(addr16) ← (addr16) – 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 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			×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

3. cnt indicates the bit shift count.

Table 30-5. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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, saddr.bit	3	1	–	CY ← (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← sfr.bit			×
		CY, A.bit	2	1	–	CY ← A.bit			×
		CY, PSW.bit	3	1	–	CY ← PSW.bit			×
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		saddr.bit, CY	3	2	–	(saddr).bit ← CY			
		sfr.bit, CY	3	2	–	sfr.bit ← CY			
		A.bit, CY	2	1	–	A.bit ← CY			
		PSW.bit, CY	3	4	–	PSW.bit ← CY	×	×	
		[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, saddr.bit	3	1	–	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∧ PSW.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, saddr.bit	3	1	–	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, saddr.bit	3	1	–	CY ← CY ⩛ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ⩛ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ⩛ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ⩛ PSW.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⩛ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⩛ (ES, HL).bit			×
	SET1	saddr.bit	3	2	–	(saddr).bit ← 1			
		sfr.bit	3	2	–	sfr.bit ← 1			
		A.bit	2	1	–	A.bit ← 1			
		!addr16.bit	4	2	–	(addr16).bit ← 1			
		PSW.bit	3	4	–	PSW.bit ← 1	×	×	×
		[HL].bit	2	2	–	(HL).bit ← 1			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 1			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 1			
	CLR1	saddr.bit	3	2	–	(saddr.bit) ← 0			
		sfr.bit	3	2	–	sfr.bit ← 0			
		A.bit	2	1	–	A.bit ← 0			
		!addr16.bit	4	2	–	(addr16).bit ← 0			
		PSW.bit	3	4	–	PSW.bit ← 0	×	×	×
		[HL].bit	2	2	–	(HL).bit ← 0			
		ES:!addr16.bit	5	3	–	(ES, addr16).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 ← $\overline{CY}$			×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

Table 30-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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	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			

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum.

3. n indicates the number of register banks (n = 0 to 3)

CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)

<R> **Cautions 1. RL78/F12 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.**

2. Pins mounted are as follows according to product.

31.1 Pins Mounted According to Product

31.1.1 Port functions

Refer to **2.1.1 20-pin products** to **2.1.5 64-pin products**.

31.1.2 Non-port functions

Refer to **2.1.6 Pins for each product (pins other than port pins)**.

Caution The pins mounted depend on the product.

31.2 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +6.5	V
	EV_{DD}		–0.5 to +6.5	V
	V_{SS}		–0.5 to +0.3	V
	EV_{SS}		–0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	–0.3 to +2.8 and –0.3 to V_{DD} +0.3 Note 1	V
Input voltage	V_{I1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147	–0.3 to EV_{DD}+0.3 and –0.3 to V_{DD}+0.3 Note 2	V
	V_{I2}	P60 to P63 (N-ch open-drain)	–0.3 to +6.5	V
	V_{I3}	P20 to P27, P121 to P124, P137, $\overline{\text{RESET}}$	–0.3 to V_{DD} +0.3 Note 2	V
Output voltage	V_{O1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P130, P140, P141, P146, P147	–0.3 to EV_{DD} +0.3 Note 2	V
	V_{O2}	P20 to P27	–0.3 to V_{DD} +0.3	V
Analog input voltage	V_{AI1}	ANI0 to ANI7	–0.3 to V_{DD} +0.3 Note 2	V
	V_{AI2}	ANI16 to ANI19	–0.3 to EV_{DD} +0.3 Note 2	V

Notes 1. Connect the REGC pin to Vss 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.

2. Must be 6.5 V or lower.

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.

Caution The pins mounted depend on the product.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141, P146, P147	–40	mA
		Total of all pins –170 mA	P00 to P04, P40 to P43, P120, P130, P140, P141	–70	mA
			P05, P06, P10 to P17, P30, P31, P50 to P55, P70 to P77, P146, P147	–100	mA
	I_{OH2}	Per pin	P20 to P27	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P130, P140, P141, P146, P147	40	mA
		Total of all pins 170 mA	P00 to P04, P40 to P43, P120, P130, P140, P141	70	mA
			P05, P06, P10 to P17, P30, P31, P50 to P55, P60 to P63, P70 to P77, P146, P147	100	mA
	I_{OL2}	Per pin	P20 to P27	1	mA
		Total of all pins		5	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

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.

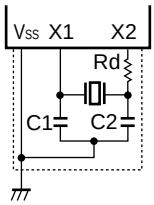
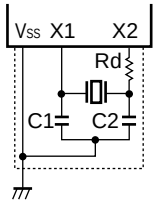
Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Caution The pins mounted depend on the product.

31.3 Oscillator Characteristics

31.3.1 Main system clock oscillator characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (f_X)Note	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	1.0		8.0	MHz
Crystal resonator		X1 clock oscillation frequency (f_X)Note	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	1.0		8.0	MHz

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures 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 V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. 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.

Caution The pins mounted depend on the product.

31.3.2 On-chip oscillator characteristics

(T_A = –20 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Note	f_{IH}	32 MHz selected	31.52	32.0	32.48	MHz
		24 MHz selected	23.64	24.00	24.36	MHz
		16 MHz selected	15.76	16.00	16.24	MHz
		8 MHz selected	7.88	8.00	8.12	MHz
		4 MHz selected	3.94	4.00	4.06	MHz
		1 MHz selected	0.985	1.00	1.015	MHz

Note This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

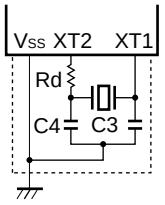
Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Note	f_{IH}	32 MHz selected	31.36	32.00	32.64	MHz
		24 MHz selected	23.52	24.00	24.48	MHz
		16 MHz selected	15.68	16.00	16.32	MHz
		8 MHz selected	7.84	8.00	8.16	MHz
		4 MHz selected	3.92	4.00	4.08	MHz
		1 MHz selected	0.98	1.00	1.02	MHz
Low-speed on-chip oscillator clock frequency	f_{IL}		12.75	15	17.25	kHz

Note This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Caution The pins mounted depend on the product.

31.3.3 Subsystem clock oscillator characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Resonator	Recommended Circuit	Items	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})[Note]		29.0	32.768	35.0	kHz

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Cautions 1. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures 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 V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Caution The pins mounted depend on the product.

31.4 DC Characteristics

31.4.1 Pin characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD} ≤ 5.5 V			–5.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			–3.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			–0.5	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141 (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			–20.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			–10.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			–5.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P55, P70 to P77, P146, P147 (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			–30.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			–19.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			–10.0	mA
		Total of all pins (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			–50.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			–29.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			–15.0	mA
	I_{OH2}	Per pin for P20 to P27				–0.1	mA
		Total of all pins (When duty = 70% Note 2)				–0.8	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from the EV_{DD} pin to an output pin.

2. Specification under conditions where the duty factor is 70%.
The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n% (the duty before change < 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) = –8.75 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.

Caution P00, P10 to P15, P17, P50, P71, P74 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.

Caution The pins mounted depend on the product.

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low Note 1	I_{OL1}	Per pin for P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD} ≤ 5.5 V			8.5	mA
			2.7 V ≤ EV_{DD} < 4.0 V			4.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			0.6	mA
		Per pin for P60 to P63	4.0 V ≤ EV_{DD} ≤ 5.5 V			15.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			4.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			2.0	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141 (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			20.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			15.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			9.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P55, P60 to P63, P70 to P77, P146, P147 (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			45.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			35.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			20.0	mA
		Total of all pins (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			65.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			50.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			29.0	mA
	I_{OL2}	Per pin for P20 to P27				0.4	mA
		Total of all pins (When duty = 70% Note 2)				3.2	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the EV_{SS} and V_{SS} pin.

2. Specification under conditions where the duty factor is 70%.
The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n% (the duty before change < 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) = 8.75 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.

Caution The pins mounted depend on the product.

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer (ITHL = 1)	0.8 EV_{DD}		EV_{DD}	V
	V_{IH2}	P01, P03, P04, P13 to P17, P55	TTL input buffer 4.0 V ≤ EV_{DD} ≤ 5.5 V	2.2		EV_{DD}	V
			TTL input buffer 2.7 V ≤ EV_{DD} < 4.0 V	2.0		EV_{DD}	V
			TTL input buffer 1.8 V ≤ EV_{DD} < 2.7 V	0		0.3EV_{DD}	V
	V_{IH3}	P20 to P27		0.7 V_{DD}		V_{DD}	V
	V_{IH4}	P60 to P63		0.7 EV_{DD}		6.0	V
	V_{IH5}	P121 to P124, P137, $\overline{\text{RESET}}$		0.8 V_{DD}		V_{DD}	V
	V_{IH6}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer (ITHL = 0)	0.8 EV_{DD}		EV_{DD}	V
Input voltage, low	V_{IL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer (ITHL = 1)	0		0.2 EV_{DD}	V
	V_{IL2}	P01, P03, P04, P13 to P17, P55	TTL input buffer 4.0 V ≤ EV_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 2.7 V ≤ EV_{DD} < 4.0 V	0		0.5	V
			TTL input buffer 1.8 V ≤ EV_{DD} < 2.7 V	0		0.3EV_{DD}	V
	V_{IL3}	P20 to P27		0		0.3 V_{DD}	V
	V_{IL4}	P60 to P63		0		0.3 EV_{DD}	V
	V_{IL5}	P121 to P124, P137, $\overline{\text{RESET}}$		0		0.2 V_{DD}	V
	V_{IL6}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer (ITHL = 0) 4.0 V ≤ EV_{DD} ≤ 5.5 V	0		0.5 EV_{DD}	V
			Normal input buffer (ITHL = 0) 2.7 V ≤ EV_{DD} < 4.0 V	0		0.4 EV_{DD}	V
			Normal input buffer 1.8 V ≤ EV_{DD} < 2.7 V	0		0.3 EV_{DD}	V

Cautions The maximum value of V_{IH} of pins P00, P02 to P04, P10 to P15, P17, P50, P55, P71, P74 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. The input pins of alternate-functions: CSIS0, CSIS1, UARTS, and UARTF, do not support TTL inputs.

Caution The pins mounted depend on the product.

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD} ≤ 5.5 V	I_{OH1} = –5.0 mA	EV_{DD} – 0.9			V
				I_{OH1} = –3.0 mA	EV_{DD} – 0.7			
				I_{OH1} = –1.0 mA	EV_{DD} – 0.5			
			2.7 V ≤ EV_{DD} ≤ 5.5 V	I_{OH1} = –3.0 mA	EV_{DD} – 0.7			V
				I_{OH1} = –1.0 mA	EV_{DD} – 0.5			
			1.8 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –0.5 mA		EV_{DD} – 0.5			V
	V_{OH2}	P20 to P27	1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OH2} = –0.1 mA		V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA				0.7	V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 4.0 mA				0.4	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 4.0 mA				0.7	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA				0.4	V
			1.8 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 0.6 mA				0.4	V
	V_{OL2}	P20 to P27	1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 0.4 mA				0.4	V
	V_{OL3}	P60 to P63	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 15.0 mA				2.0	V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 5.0 mA				0.4	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V	I_{OL3} = 4.0 mA			0.5	V
				I_{OL3} = 3.0 mA			0.4	
			1.8 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 2.0 mA				0.4	V

Caution P00, P02 to P04, P10 to P15, P17, P43, P50, P52 to P55, P71, P74 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.

Caution The pins mounted depend on the product.

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P140, P141, P146, P147	$V_I = EV_{DD}$				1	μA
	I_{LIH2}	P20 to P27, P137, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P140, P141, P146, P147	$V_I = EV_{SS}$				–1	μA
	I_{LIL2}	P20 to P27, P137, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{SS}$	In input port or external clock input			–1	μA
				In resonator connection			–10	μA
On-chip pll-up resistance	R_U	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147			10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Caution The pins mounted depend on the product.

31.4.2 Supply current characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1} Note 1	Operating mode	High-speed operation Note 5	f_{IH} = 32 MHz Note 2		5.6	8.2	mA
				f_{IH} = 24 MHz Note 2		4.5	6.5	mA
				f_{IH} = 16 MHz Note 2		3.3	4.8	mA
				f_{MX} = 20 MHz Note 3		4.0	5.5	mA
				f_{MX} = 10 MHz Note 3		2.4	3.1	mA
			Low-speed operation Note 5	f_{IH} = 8 MHz Note 2		1.6	2.3	mA
				f_{MX} = 8 MHz Note 3		1.5	2.3	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 4		4.9	13.0	μA

Notes 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 in the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors (when high-speed on-chip oscillator or subsystem clock, not including the current flowing into the BGO too).

2. When high-speed system clock and subsystem clock are stopped.

3. When high-speed on-chip oscillator and subsystem clock are stopped.

4. When high-speed on-chip oscillator and high-speed system clock are stopped. When watchdog timer is stopped. When AMPHS1 = 1 (Ultra-low power consumption oscillation).

5. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.
High speed operation: V_{DD} = 2.7 to 5.5 V@1 MHz to 32 MHz
Low speed operation: V_{DD} = 1.8 to 5.5 V@1 MHz to 8 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: High-speed on-chip oscillator clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

4. Temperature condition of the TYP. value is T_A = 25°C

Caution The pins mounted depend on the product.

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	High-speed operation Note 7	f_{IH} = 32 MHz Note 3			0.55	3.2	mA
				f_{IH} = 24 MHz Note 3			0.48	2.42	
				f_{IH} = 16 MHz Note 3			0.40	1.75	
				f_{MX} = 20 MHz Note 4			0.43	1.80	
				f_{MX} = 10 MHz Note 4			0.28	0.97	
			Low-speed operation Note 7	f_{IH} = 8 MHz Note 3			0.30	0.84	mA
				f_{MX} = 8 MHz Note 4			0.18	0.60	
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 5	T_A ≤+50°C		0.52	2.15	μA
					T_A ≤+70°C			3.05	
					T_A ≤+85°C			4.24	
	I_{DD3} Note 6	STOP mode	T_A ≤+50°C				0.22	2.05	μA
			T_A ≤+70°C					2.95	
			T_A ≤+85°C					4.16	

Notes 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 in the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. During HALT instruction execution by flash memory.

3. When high-speed system clock and subsystem clock are stopped.

4. When high-speed on-chip oscillator and subsystem clock are stopped.

5. When operating real-time clock (RTC) and setting ultra-low current consumption (AMPHS1 = 1). When high-speed on-chip oscillator and high-speed system clock are stopped. When watchdog timer is stopped.

6. When high-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped. When watchdog timer is stopped.

7. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.
High speed operation: V_{DD} = 2.7 to 5.5 V@1 MHz to 32 MHz
Low speed operation: V_{DD} = 1.8 to 5.5 V@1 MHz to 8 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: High-speed on-chip oscillator clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

4. Temperature condition of the TYP. value is T_A = 25°C

Caution The pins mounted depend on the product.

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
RTC operating current	I_{RTC} Notes 1, 2	f_{SUB} = 32.768 kHz	Real-time clock operation		0.02	0.13	μA
			Interval timer operation		0.02	0.33	μA
WUTM operating current	I_{WUTM}	f_{IL} = 15 kHz			0.25	0.4	μA
Watchdog timer operating current	I_{WDT} Notes 2, 3	f_{IL} = 15 kHz			0.22	0.4	μA
A/D converter operating current	I_{ADC} Note 4	at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		1.30	1.7	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.5	0.7	mA
		Internal reference voltage selected Note 7			75		μA
LVD operating current	I_{LVI} Note 5				0.08	0.20	μA
Temperature sensor operating current	I_{TMPS}				75		μA
BGO operating current	I_{BGO} Note 6				2.50	12.20	mA

Notes 1. Current flowing only to the real-time clock (excluding the operating current of the XT1 oscillator). The TYP. value of the current value of the RL78/F12 is the sum of the TYP. values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock operates in operation mode or HALT mode. The I_{DD1} and I_{DD2} MAX. values also include the real-time clock operating current. When the real-time clock operates during f_{CLK} = f_{SUB}, the TYP. value of I_{DD2} includes the real-time clock operating current.

2. When high-speed on-chip oscillator and high-speed system clock are stopped.

3. Current flowing only to the watchdog timer (including the operating current of the 15 kHz on-chip oscillator). The current value of the RL78/F12 is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when f_{CLK} = f_{SUB} when the watchdog timer operates in STOP mode.

4. Current flowing only to the A/D converter. The current value of the RL78/F12 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.

5. Current flowing only to the LVD circuit. The current value of the RL78/F12 is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVD circuit operates in the Operating, HALT or STOP mode.

6. Current flowing only to the BGO. The current value of the RL78/F12 is the sum of I_{DD1} or I_{DD2} and I_{BGO} when the BGO operates in an operation mode or the HALT mode.

7. This indicates operating current which increases when the internal reference voltage is selected. The Current flows even if the conversion is stopped.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

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

4. Temperature condition of the TYP. value is T_A = 25°C

Caution The pins mounted depend on the product.

31.5 AC Characteristics

31.5.1 Basic operation

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	High-speed main mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.03125		1	μs
			Low-speed main mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.125		1	μs
		Subsystem clock (f_{SUB}) operation SDIV=0			28.5	30.5	31.3	μs
External main system clock frequency	f_{EX}	EXCLK	2.7 V ≤ V_{DD} ≤ 5.5 V		1		20	MHz
			1.8 V ≤ V_{DD} < 2.7 V		1		8	MHz
	f_{EXS}	EXCLKS			29		35	kHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	EXCLK	2.7 V ≤ V_{DD} ≤ 5.5 V		24			ns
			1.8 V ≤ V_{DD} < 2.7 V		60			ns
	t_{EXHS}, t_{EXLS}	EXCLKS			13.7			μs
TI00 to TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}				2/f_{MCK} +10			ns
TO00 to TO07 output frequency	f_{TO}	4.0 V ≤ EV_{DD} ≤ 5.5 V					16	MHz
		2.7 V ≤ EV_{DD} < 4.0 V					8	MHz
		1.8 V ≤ EV_{DD} < 2.7 V					4	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	4.0 V ≤ EV_{DD} ≤ 5.5 V					16	MHz
		2.7 V ≤ EV_{DD} < 4.0 V					8	MHz
		1.8 V ≤ EV_{DD} < 2.7 V					4	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP11			1			μs
Key interrupt input low-level width	t_{KR}	KR0 to KR7			250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock set by the CKS0n bit of Timer mode register 0n (TMR0n). n: Channel number (n = 0 to 7))

Caution The pins mounted depend on the product.

31.6 Peripheral Functions Characteristics

31.6.1 Serial array unit

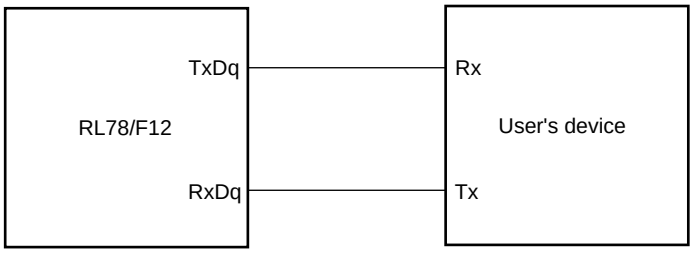
(1) During communication at same potential (UART mode) (dedicated baud rate generator output)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

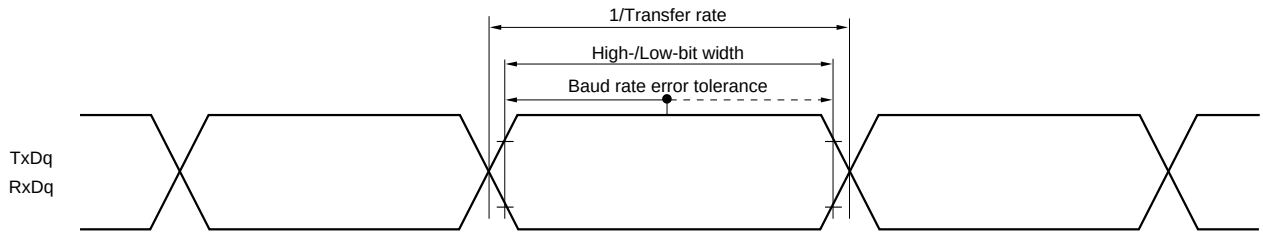
<R>

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		Other than SNOOZE mode	f_{MCK}/256		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate			5.3	Mbps
		Receivable baud rate at SNOOZE mode	4800		4800	bps

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



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 h (POMh).

Remarks 1. q: UART number (q = 0 to 2, S0), g: PIM number (g = 0, 1, 5), h: POM number (h = 0, 1, 5, 7)

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 = 00 to 03, 10, 11, S0, S1))

Caution The pins mounted depend on the product.

(2) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$: internal clock output)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.		TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time Note 1	t_{KCY1}	CSI00	2.7 V ≤ EV_{DD} ≤ 5.5 V	125	Besides $2/f_{MCK}$ ≤			ns
			1.8 V ≤ EV_{DD} ≤ 2.7 V	500				ns
		Other than CSI00	2.7 V ≤ EV_{DD} ≤ 5.5 V	125	Besides $4/f_{MCK}$ ≤			ns
			1.8 V ≤ EV_{DD} ≤ 2.7 V	500				ns
<R> $\overline{SCKp}$ high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V		$t_{KCY1}/2 - 12$				ns
		2.7 V ≤ EV_{DD} ≤ 4.0 V		$t_{KCY1}/2 - 18$				ns
		1.8 V ≤ EV_{DD} ≤ 2.7 V		$t_{KCY1}/2 - 50$				ns
<R> SIp setup time (to $\overline{SCKp}$↑) Note 2	t_{SIK1}	2.7 V ≤ EV_{DD} ≤ 5.5 V		44				ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V		110				ns
SIp hold time (from $\overline{SCKp}$↑) Note 2	t_{KSI1}			19				ns
SOp output delay time (from $\overline{SCKp}$↓) Note 3	t_{KSO1}	C = 30 pF Note 4					25	ns

Notes 1. The value must also be $2/f_{CLK}$ (CS100) or $4/f_{CLK}$ (other than CS100).

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and $\overline{SCKp}$ pin by using port input mode register g (PIMg) and port output mode register h (POMh).

Remark p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), m: Unit number (m = 0, 1, S),
n: Channel number (n = 0 to 3), g: PIM number (g = 0, 1, 5), h: POM number (h = 0, 1, 5, 7)

Caution The pins mounted depend on the product.

(3) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$: external clock input)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY2}	4.0 V ≤ EV_{DD} < 5.5 V	20 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}			ns
		1.8 V ≤ EV_{DD} < 4.0 V	16 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
$\overline{SCKp}$ high-/low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2			ns
SIp setup time (to $\overline{SCKp}$↑) **Note 1**	t_{SIK2}	2.7 V ≤ EV_{DD} ≤ 5.5 V		1/f_{MCK}+20			ns
		1.8 V ≤ EV_{DD} < 2.7 V		1/f_{MCK}+30			
SIp hold time (from $\overline{SCKp}$↑) **Note 1**	t_{KSI2}	1.8 V ≤ EV_{DD} ≤ 5.5 V		1/f_{MCK}+31			ns
SOp output delay time (from $\overline{SCKp}$↓) **Note 2**	t_{KSO2}	C = 30 pF **Note 3**	2.7 V ≤ EV_{DD} ≤ 5.5 V			2/f_{MCK}+44	ns
			1.8 V ≤ EV_{DD} < 2.7 V			2/f_{MCK}+110	ns

Notes 1. This applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time is "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. This applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output is "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. C is the load capacitance of the SOp output lines.

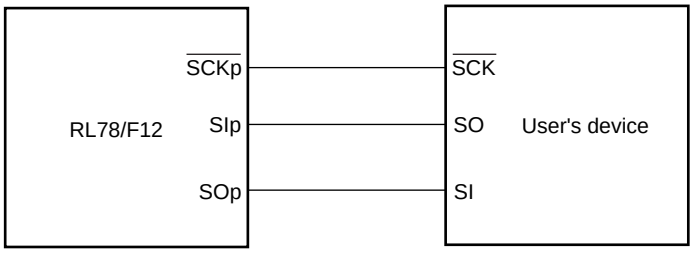
Caution Select the TTL input buffer for the SIp pin and $\overline{SCKp}$ pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register h (POMh).

Remarks 1. p: CSI number (p = 00, 01, 10, 11, 20, 21, S0 ,S1), m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), g: PIM number (g = 0, 1, 5), h: POM number (h = 0, 1, 5, 7)

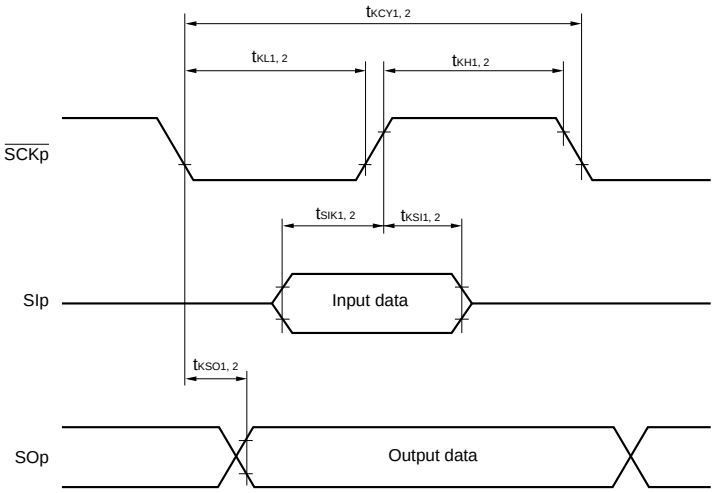
2. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10, 11, S0, S1))

Caution The pins mounted depend on the product.

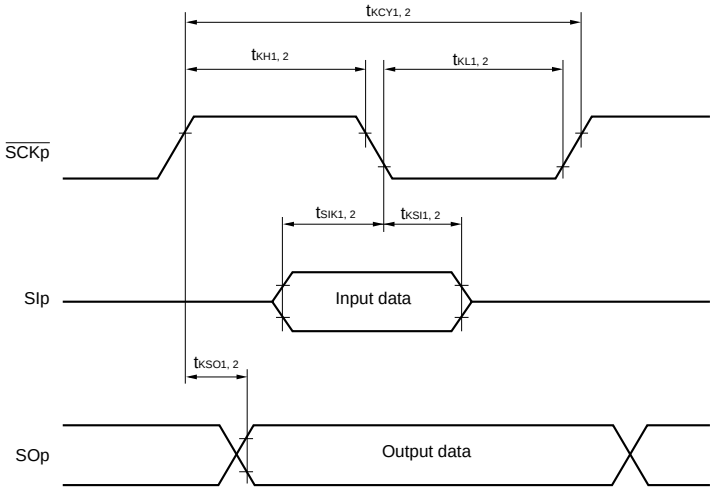
CSI mode connection diagram (during communication at same potential)



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.)



Remarks 1. p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1)

2. m: Unit number, n: Channel number (mn = 00 to 03, 10, 11, S0, S1)

Caution The pins mounted depend on the product.

(4) During communication at same potential (simplified I^2C mode)

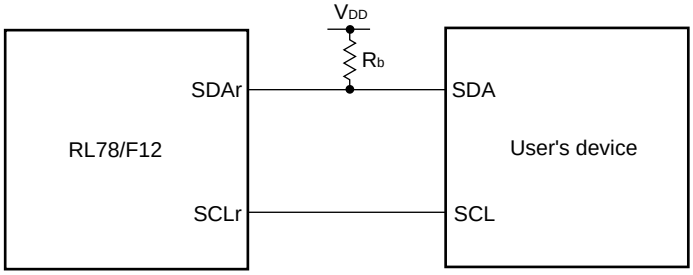
(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400	kHz
Hold time when SCLr = “L”	t_{LOW}	1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		ns
Hold time when SCLr = “H”	t_{HIGH}	1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		ns
Data setup time (reception)	$t_{SU:DAT}$	1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	$1/f_{MCK}$ + 145 **Note**		ns
Data hold time (transmission)	$t_{HD:DAT}$	1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	355	ns

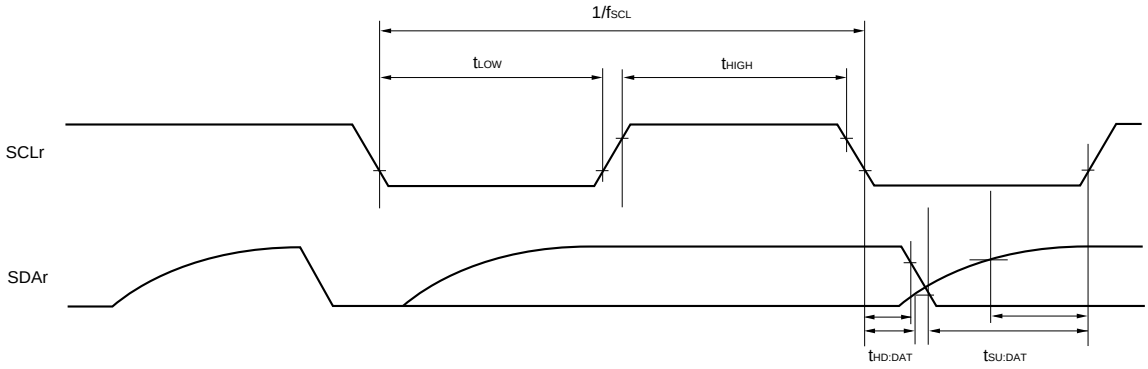
Note The value of f_{MCK} must be such that this does not exceed the hold time for SCLr = L or the hold time for SCLr = H.

Caution The pins mounted depend on the product.

Simplified I²C mode mode connection diagram (during communication at same potential)



Simplified I²C mode serial transfer timing (during communication at same potential)



Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

Remarks 1. R_b[Ω]: Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

2. r: IIC number (r = 00, 01, 10, 11, 20, 21), g: PIM number (g = 0, 1, 5), h: POM number (h = 0, 1, 5, 7)

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 (m = 0, 1, 2), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11, 20, 21)

Caution The pins mounted depend on the product.

31.6.2 Serial interface IICA

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		Standard Mode		Fast Mode		Fast Mode Plus		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode plus: f_{CLK} ≥ 10 MHz	2.7 V ≤ EV_{DD} ≤ 5.5 V					0	1000	kHz
		Fast mode: f_{CLK} ≥ 3.5 MHz	1.8 V ≤ EV_{DD} ≤ 5.5 V			0	400			kHz
		Normal mode: f_{CLK} ≥ 1 MHz	1.8 V ≤ EV_{DD} ≤ 5.5 V	0	100					kHz
Setup time of restart condition Note 1	$t_{SU:STA}$			4.7		0.6		0.26		μs
Hold time	$t_{HD:STA}$			4.0		0.6		0.26		μs
Hold time when SCLA0 = “L”	t_{LOW}			4.7		1.3		0.5		μs
Hold time when SCLA0 = “H”	t_{HIGH}			4.0		0.6		0.26		μs
Data setup time (reception)	$t_{SU:DAT}$			250		100		50		ns
Data hold time (transmission) Note 2	$t_{HD:DAT}$			0	3.45	0	0.9	0		μs
Setup time of stop condition	$t_{SU:STO}$			4.0		0.6		0.26		μs
Bus-free time	t_{BUF}			4.7		1.3		0.5		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

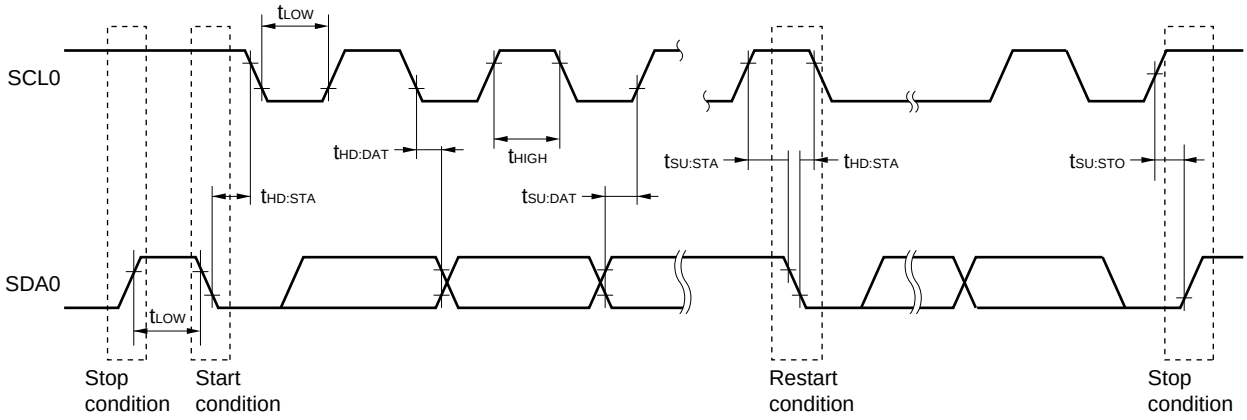
2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the $\overline{ACK}$ (acknowledge) timing.

Remark The maximum value of Cb (communication line capacitance) and the value of Rb (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
Fast mode: C_b = 320 pF, R_b = 1.1 kΩ
Fast mode plus: C_b = 120 pF, R_b = 1.1 kΩ

Caution The pins mounted depend on the product.

IICA serial transfer timing



31.6.3 LIN-UART

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Transfer rate	1/T			1 Note	Mbps

Note However, the upper limit is fCLK/8.

Caution The pins mounted depend on the product.

31.7 Analog Characteristics

31.7.1 A/D converter characteristics

(1) When the setting of AV_{REF} (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1) and AV_{REF} (-) = AV_{REFM}/ANI1 (ADREFM = 1), this applies to the following ANI pins: ANI2 to ANI7 (the ANI pins for which V_{DD} is the power-supply voltage).

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, reference voltage (+) = AV_{REFP}, reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error Note 1	AINL	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V		1.2	±3.0	LSB
			1.8 V ≤ V_{DD} < 4.0 V		1.2	±3.5	LSB
Conversion time	t_{CONV}	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±0.25	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±0.25	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±1.5	LSB
Reference voltage (+)	AV_{REFP}			1.8		V_{DD}	V
Reference voltage (-)	AV_{REFM}			0			
Analog input voltage	V_{AIN}			AV_{REFM}		AV_{REFP}	V
	V_{BGR}	2.7 V ≤ V_{DD} ≤ 5.5 V		1.38	1.45	1.5	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to 2.1.1, 20-pin products to 2.1.5, 64-pin products, and 2.1.6, Pins for each product (pins other than port pins).

(2) When the setting of AV_{REF} (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1) and AV_{REF} (-) = AV_{REFM}/ANI1 (ADREFM = 1), this applies to the following ANI pins: ANI16 to ANI19 (the ANI pins for which EV_{DD} is the power-supply voltage).

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, reference voltage (+) = AV_{REFP}, reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error Note 1	AINL	10-bit resolution AV_{REFP} = V_{DD} AV_{REFM} = V_{SS}	4.0 V ≤ V_{DD} ≤ 5.5 V		1.2	±4.5	LSB
			1.8 V ≤ V_{DD} < 4.0 V		1.2	±5.0	LSB
Conversion time	t_{CONV}	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±0.35	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±0.35	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±3.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Reference voltage (+)	AV_{REFP}			1.8		V_{DD}	V
Reference voltage (-)	AV_{REFM}			0			V
Analog input voltage	V_{AIN}			AV_{REFM}		AV_{REFP}	V
	V_{BGR}	2.7 V ≤ V_{DD} ≤ 5.5 V		1.38	1.45	1.5	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to 2.1.1, 20-pin products to 2.1.5, 64-pin products, and 2.1.6, Pins for each product (pins other than port pins).

(3) When the setting of AV_{REF} (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0) and AV_{REF} (-) = V_{SS} (ADREFM = 0), this applies to the following ANI pins: ANI0 to ANI7.

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, reference voltage (+) = V_{DD}, reference voltage (-) = V_{SS})

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}				8		10	bit
Overall error Note 1	AINL	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V	ANI0-ANI7		1.2	±5.0	LSB
			1.8 V ≤ V_{DD} < 4.0 V	ANI0-ANI7		1.2	±5.5	LSB
Conversion time	t_{CONV}	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V		2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V		3.1875		39	μs
			1.8 V ≤ V_{DD} ≤ 5.5 V		17		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V				±0.50	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V				±0.50	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V				±3.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V				±2.0	LSB
Reference voltage (+)	AV_{REFP}				V_{DD}			V
Reference voltage (-)	AV_{REFM}				V_{SS}			V
Analog input voltage	V_{AIN}	ANI0-ANI7			V_{SS}		V_{DD}	V
	V_{BGR}	2.7 V ≤ V_{DD} ≤ 5.5 V			1.38	1.45	1.5	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to 2.1.1, 20-pin products to 2.1.5, 64-pin products, and 2.1.6, Pins for each product (pins other than port pins).

(4) When the setting of AV_{REF} (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0) and AV_{REF} (-) = V_{SS} (ADREFM = 0), this applies to the following ANI pins: ANI16 to ANI19.

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, reference voltage (+) = V_{DD}, reference voltage (-) = V_{SS})

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}				8		10	Bit
Overall error Note 1	AINL	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V	ANI16-ANI19		1.2	±6.5	LSB
			1.8 V ≤ V_{DD} < 4.0 V	ANI16-ANI19		1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V		2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V		3.1875		39	μs
			1.8 V ≤ V_{DD} ≤ 5.5 V		17		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V				±0.60	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V				±0.60	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V				±4.0	LSB
Differential linearity error Note 1	DLE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V				±2.0	LSB
Reference voltage (+)	AV_{REFP}				V_{DD}			V
Reference voltage (-)	AV_{REFM}				V_{SS}			V
Analog input voltage	V_{AIN}	ANI16-ANI19			V_{SS}		EV_{DD0}	V
	V_{BGR}	2.7 V ≤ V_{DD} ≤ 5.5 V			1.38	1.45	1.5	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to 2.1.1, 20-pin products to 2.1.5, 64-pin products, and 2.1.6, Pins for each product (pins other than port pins).

Caution The pins mounted depend on the product.

31.7.2 Temperature sensor characteristics

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, Vss = EVss0 = EVss1 = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Reference output voltage	V_{CONST}	Setting ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	F_{VTMPS}	Temperature sensor that depends on the temperature		–3.6		mV/C
Operation stabilization wait time	t_{AMP}				5	μs

31.7.3 POR circuit characteristics

(T_A = –40 to +85°C, V_{SS} = FV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.46	1.51	1.59	V
	V_{PDR}	Power supply fall time	1.45	1.50	1.58	V
Minimum pulse width	T_{PW}		300			μs
Detection delay time	T_{PD}				350	μs

Caution The pins mounted depend on the product.

31.7.4 LVD circuit characteristics

(a) Characteristics for LVD Detection at Reset and Interrupt modes

(T_A = –40 to +85°C, $V_{PDR} \leq V_{DD} = EV_{DD} \leq 5.5$ V, $V_{SS} = EV_{SS}$ = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}	Power supply rise time	3.96	4.06	4.25	V
			Power supply fall time	3.89	3.98	4.15	V
		V_{LVI1}	Power supply rise time	3.66	3.75	3.93	V
			Power supply fall time	3.58	3.67	3.83	V
		V_{LVI2}	Power supply rise time	3.06	3.13	3.28	V
			Power supply fall time	2.99	3.06	3.20	V
		V_{LVI3}	Power supply rise time	2.95	3.02	3.17	V
			Power supply fall time	2.89	2.96	3.09	V
		V_{LVI4}	Power supply rise time	2.85	2.92	3.07	V
			Power supply fall time	2.79	2.86	2.99	V
		V_{LVI5}	Power supply rise time	2.74	2.81	2.95	V
			Power supply fall time	2.68	2.75	2.88	V
		V_{LVI6}	Power supply rise time	2.64	2.71	2.85	V
			Power supply fall time	2.59	2.65	2.77	V
		V_{LVI7}	Power supply rise time	2.55	2.61	2.74	V
			Power supply fall time	2.49	2.55	2.67	V
		V_{LVI8}	Power supply rise time	2.44	2.50	2.63	V
			Power supply fall time	2.39	2.45	2.57	V
		V_{LVI9}	Power supply rise time	2.04	2.09	2.21	V
			Power supply fall time	1.99	2.04	2.14	V
		V_{LVI10}	Power supply rise time	1.93	1.98	2.09	V
			Power supply fall time	1.89	1.94	2.04	V
		V_{LVI11}	Power supply rise time	1.83	1.88	1.99	V
			Power supply fall time	1.79[Note]	1.84	1.94	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time						300	μs

Note The minimum value lowers the minimum guaranteed voltage for operation(1.8V). However, LVD detection performs in the same way as in normal mode (operation according to the same specification when VDD is 1.8V) until it is reset at reset mode.

Remark $V_{LVI(n-1)} > V_{LVIn}$: n = 1 to 13

The following relationship is formed under the same temperature conditions: the detection voltage at power supply rise time > the detection voltage at power supply fall time.

Caution The pins mounted depend on the product.

(b) LVD Detection Voltage of Interrupt & Reset Mode

(T_A = –40 to +85°C, $V_{PDR} \leq V_{DD} = EV_{DD} \leq 5.5$ V, $V_{SS} = EV_{SS}$ = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVI11}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 1, falling reset voltage: 1.8 V			1.79[Note]	1.84	1.94	V
	V_{LVI10}		LVIS1, LVIS0 = 1, 0	Rising release reset voltage	1.93	1.98	2.09	V
				Falling interrupt voltage	1.89	1.94	2.04	V
	V_{LVI9}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.04	2.09	2.21	V
				Falling interrupt voltage	1.99	2.04	2.14	V
	V_{LVI2}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.06	3.13	3.28	V
				Falling interrupt voltage	2.99	3.06	3.20	V
	V_{LVI8}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 0, falling reset voltage: 2.4 V			2.39	2.45	2.57	V
	V_{LVI7}		LVIS1, LVIS0 = 1, 0	Rising release reset voltage	2.55	2.61	2.74	V
				Falling interrupt voltage	2.49	2.55	2.67	V
	V_{LVI6}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.64	2.71	2.85	V
				Falling interrupt voltage	2.59	2.65	2.77	V
	V_{LVI1}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.66	3.75	3.93	V
				Falling interrupt voltage	3.58	3.67	3.83	V
	V_{LVI5}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 1, falling reset voltage: 2.7 V			2.68	2.75	2.88	V
	V_{LVI4}		LVIS1, LVIS0 = 1, 0	Rising release reset voltage	2.85	2.92	3.07	V
				Falling interrupt voltage	2.79	2.86	2.99	V
	V_{LVI3}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.95	3.02	3.17	V
				Falling interrupt voltage	2.89	2.96	3.09	V
	V_{LVI0}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.96	4.06	4.25	V
				Falling interrupt voltage	3.89	3.98	4.15	V

Note The minimum value lowers the minimum guaranteed voltage for operation(1.8V). However, LVD detection performs in the same way as in normal mode (operation according to the same specification when VDD is 1.8V) until it is reset at reset mode.

Remark The following relationship is formed under the same temperature conditions: the rising release reset voltage > the falling interrupt voltage > the falling reset voltage

Caution The pins mounted depend on the product.

<R> **31.7.5 Power supply rise time**

(T_A = –40 to +85°C, Vss = EVss = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum slew rate for the supply voltage to rise	S_{vrmax}	0 V → V_{DD} (MIN.) Note 2 (VPOC2 = 0 or 1)			50 Note 1	V/ms
Minimum slew rate for the supply voltage to rise Note 3	S_{vrmin}	0V → 1.8 V (CMODE0 = 0)	3.5 Note 1			V/ms
		0V → 2.7 V (CMODE0 = 1)	6.5 Note 1			V/ms

Notes 1. In case the supply voltage falls to a level of V_{PDR} or below and a power-on reset is generated, the slew rate must not exceed the value S_{vrmax} even if the supply voltage does not go down to 0 V.

2. V_{DD} (MIN.) varies depending on the setting of the flash operation mode in the option byte (CMODE0 bit).
LS (low speed main) mode (CMODE0 = 0): V_{DD} (MIN.) = 1.8 V
HS (high speed main) mode (CMODE0 = 1): V_{DD} (MIN.) = 2.7 V

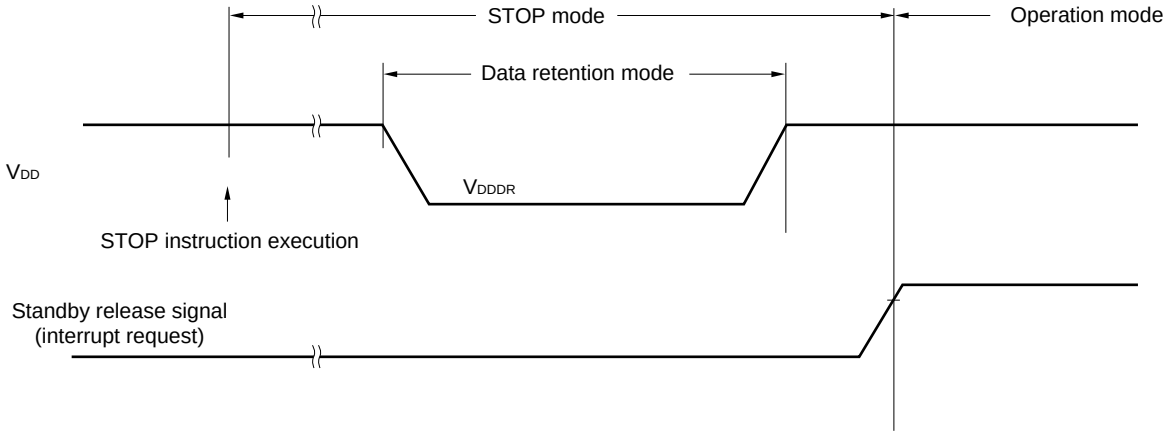
3. The minimum slew rate for the supply voltage (S_{vrmin}) must be met when the voltage detector (LVD) is not used (option byte bit VPOC2 = 1) and an external reset circuit releases before the supply voltage reaches V_{DD} (MIN.) (as specified in Note 2).

31.8 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics

(T_A = –40 to +85°C V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.45 Note		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is effected, but 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} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency		f_{CLK}		1		32	MHz
<R> Number of code flash rewrites Notes 1, 2, 3		C_{erwr}	20 years retention (after rewrite) T_A = +85°C	1000			Times
<R> Number of data flash rewrites Notes 1, 2, 3			20 years retention (after rewrite) T_A = +85°C	10000			
			5 years retention (after rewrite) T_A = +85°C	100000			
Erase time	Block erase	T_{erasa}		5			ms
write time		T_{wrwa}		10			μs

Notes 1. Retention years indicate a period between time for a rewrite and the next.

2. When using flash memory programmer and Renesas Electronics self programming library.

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas.

CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)

<R> **Cautions 1. RL78/F12 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.**

2. Pins mounted are as follows according to product.

32.1 Pins Mounted According to Product

32.1.1 Port functions

Refer to **2.1.1 20-pin products** to **2.1.5 64-pin products**.

32.1.2 Non-port functions

Refer to **2.1.6 Pins for each product (pins other than port pins)**.

Caution The pins mounted depend on the product.

32.2 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +6.5	V
	EV_{DD}		–0.5 to +6.5	V
	V_{SS}		–0.5 to +0.3	V
	EV_{SS}		–0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	–0.3 to +2.8 and –0.3 to V_{DD} +0.3 Note 1	V
Input voltage	V_{I1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147	–0.3 to EV_{DD}+0.3 and –0.3 to V_{DD}+0.3 Note 2	V
	V_{I2}	P60 to P63 (N-ch open-drain)	–0.3 to +6.5	V
	V_{I3}	P20 to P27, P121 to P124, P137, $\overline{\text{RESET}}$	–0.3 to V_{DD} +0.3 Note 2	V
Output voltage	V_{O1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P130, P140, P141, P146, P147	–0.3 to EV_{DD} +0.3 Note 2	V
	V_{O2}	P20 to P27	–0.3 to V_{DD} +0.3	V
Analog input voltage	V_{AI1}	ANI16 to ANI19	–0.3 to EV_{DD} +0.3 Note 2	V
	V_{AI2}	ANI0 to ANI7	–0.3 to V_{DD} +0.3 Note 2	V

Notes 1. Connect the REGC pin to Vss 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.

2. Must be 6.5 V or lower.

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.

Caution The pins mounted depend on the product.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141, P146, P147	–40	mA
		Total of all pins –170 mA	P00 to P04, P40 to P43, P120, P130, P140, P141	–70	mA
			P05, P06, P10 to P17, P30, P31, P50 to P55, P70 to P77, P146, P147	–100	mA
	I_{OH2}	Per pin	P20 to P27	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P130, P140, P141, P146, P147	40	mA
		Total of all pins 170 mA	P00 to P04, P40 to P43, P120, P130, P140, P141	70	mA
			P05, P06, P10 to P17, P30, P31, P50 to P55, P60 to P63, P70 to P77, P146, P147	100	mA
	I_{OL2}	Per pin	P20 to P27	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A	In normal operation mode		–40 to +125	°C
		In flash memory programming mode	Data	–40 to +125	
			Code	–40 to +105	
Storage temperature	T_{stg}			–65 to +150	°C

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.

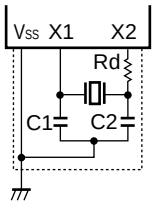
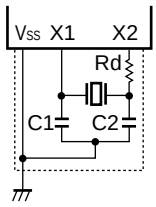
Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Caution The pins mounted depend on the product.

32.3 Oscillator Characteristics

32.3.1 Main system clock oscillator characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (f_X)Note	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz
Crystal resonator		X1 clock oscillation frequency (f_X)Note	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures 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 V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. 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.

Caution The pins mounted depend on the product.

32.3.2 On-chip oscillator characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

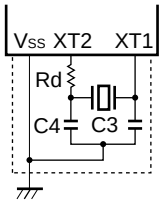
Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Note	f_{IH}	24 MHz selected	23.52	24.00	24.48	MHz
		16 MHz selected	15.68	16.00	16.32	MHz
		8 MHz selected	7.84	8.00	8.16	MHz
		4 MHz selected	3.92	4.00	4.08	MHz
		1 MHz selected	0.98	1.00	1.02	MHz
Low-speed on-chip oscillator clock frequency	f_{IL}		12.75	15	17.25	kHz

Note This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Caution The pins mounted depend on the product.

32.3.3 Subsystem clock oscillator characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Resonator	Recommended Circuit	Items	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})[Note]		29.0	32.768	35.0	kHz

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Cautions 1. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures 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 V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Caution The pins mounted depend on the product.

32.4 DC Characteristics

32.4.1 Pin characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD} ≤ 5.5 V			–5.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			–3.0	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141 (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			–20.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			–10.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P55, P70 to P77, P146, P147 (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			–30.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			–19.0	mA
		Total of all pins (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			–42.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			–29.0	mA
	I_{OH2}	Per pin for P20 to P27				–0.1	mA
		Total of all pins (When duty = 70% Note 2)				–0.8	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from the EV_{DD} pin to an output pin.

2. Specification under conditions where the duty factor is 70%.

The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n% (the duty before change < 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) = –8.75 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.

Caution P00, P10 to P15, P17, P50, P71, P74 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.

Caution The pins mounted depend on the product.

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low Note 1	I_{OL1}	Per pin for P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD} ≤ 5.5 V			8.5	mA
			2.7 V ≤ EV_{DD} < 4.0 V			4.0	mA
		Per pin for P60 to P63	4.0 V ≤ EV_{DD} ≤ 5.5 V			15.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			4.0	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141 (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			20.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			15.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P55, P60 to P63, P70 to P77, P146, P147 (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			45.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			35.0	mA
		Total of all pins (When duty = 70% Note 2)	4.0 V ≤ EV_{DD} ≤ 5.5 V			65.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			50.0	mA
	I_{OL2}	Per pin for P20 to P27				0.4	mA
		Total of all pins (When duty = 70% Note 2)				3.2	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the EV_{SS} and V_{SS} pin.

2. Specification under conditions where the duty factor is 70%.

The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n% (the duty before change < 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) = 8.75 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.

Caution The pins mounted depend on the product.

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer (ITHL = 1)	0.8 EV_{DD}		EV_{DD}	V
	V_{IH2}	P01, P03, P04, P13 to P17, P55	TTL input buffer 4.0 V ≤ EV_{DD} ≤ 5.5 V	2.2		EV_{DD}	V
	V_{IH3}	P20 to P27		0.7 V_{DD}		V_{DD}	V
	V_{IH4}	P60 to P63		0.7 EV_{DD}		6.0	V
	V_{IH5}	P121 to P124, P137, $\overline{\text{RESET}}$		0.8 V_{DD}		V_{DD}	V
	V_{IH6}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer (ITHL = 0)	0.8 EV_{DD}		EV_{DD}	V
Input voltage, low	V_{IL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer (ITHL = 1)	0		0.2 EV_{DD}	V
	V_{IL2}	P01, P03, P04, P13 to P17, P55	TTL input buffer 4.0 V ≤ EV_{DD} ≤ 5.5 V	0		0.8	V
	V_{IL3}	P20 to P27		0		0.3 V_{DD}	V
	V_{IL4}	P60 to P63		0		0.3 EV_{DD}	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0		0.2 V_{DD}	V
	V_{IL6}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147	Normal input buffer (ITHL = 0) 4.0 V ≤ EV_{DD} ≤ 5.5 V	0		0.5 EV_{DD}	V
			Normal input buffer (ITHL = 0) 2.7 V ≤ EV_{DD} < 4.0 V	0		0.4 EV_{DD}	V

Cautions The maximum value of V_{IH} of pins P00, P02 to P04, P10 to P15, P17, P50, P55, P71, P74 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. The input pins of alternate-functions: CSIS0, CSIS1, UARTS, and UARTF, do not support TTL inputs.

Caution The pins mounted depend on the product.

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –5.0 mA	EV_{DD} – 0.9			V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	EV_{DD} – 0.7			V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –1.0 mA	EV_{DD} – 0.5			V
	V_{OH2}	P20 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH2} = –100 μA	EV_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141, P146, P147	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 4.0 mA			0.4	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 4.0 mA			0.7	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
	V_{OL2}	P20 to P27	2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 400 μA			0.4	V
	V_{OL3}	P60 to P63	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 15.0 mA			2.0	V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 5.0 mA			0.4	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} =4.0 mA			0.5	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 3.0 mA			0.4	V

Caution P00, P02 to P04, P10 to P15, P17, P50, P55, P71, P74 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.

Caution The pins mounted depend on the product.

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P140, P141, P146, P147	$V_I = EV_{DD}$				1	μA
	I_{LIH2}	P20 to P27, P137, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P140, P141, P146, P147	$V_I = EV_{SS}$				–1	μA
	I_{LIL2}	P20 to P27, P137, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{SS}$	In input port or external clock input			–1	μA
				In resonator connection			–10	μA
On-chip pll-up resistance	R_U	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, P141, P146, P147			10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Caution The pins mounted depend on the product.

32.4.2 Supply current characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1} Note 1	Operating mode	High-speed operation Note 5	f_{IH} = 24 MHz Note 2			4.5	6.9	mA
				f_{IH} = 16 MHz Note 2			3.3	5.2	mA
				f_{MX} = 20 MHz Note 3			4.0	5.9	mA
				f_{MX} = 10 MHz Note 3			2.4	3.5	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 4	T_A ≤ + 85°C		4.9	13.0	μA
					T_A ≤ + 105°C			25.0	
					T_A ≤ + 125°C			59.0	

Notes 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 in the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors (when high-speed on-chip oscillator or subsystem clock, not including the current flowing into the BGO too).

2. When high-speed system clock and subsystem clock are stopped.

3. When high-speed on-chip oscillator and subsystem clock are stopped.

4. When high-speed on-chip oscillator and high-speed system clock are stopped. When watchdog timer is stopped. When AMPHS1 = 1 (Ultra-low power consumption oscillation).

5. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.
High speed operation: V_{DD} = 2.7 to 5.5 V@1 MHz to 24 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: High-speed on-chip oscillator clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

4. Temperature condition of the TYP. value is T_A = 25°C

Caution The pins mounted depend on the product.

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	High-speed operation Note 7	f_{IH} = 24 MHz Note 3		0.48	5.58	mA
				f_{IH} = 16 MHz Note 3		0.40	3.90	mA
				f_{IH} = 8 MHz Note 3		TBD	TBD	mA
				f_{MX} = 20 MHz Note 4		0.43	1.88	mA
				f_{MX} = 10 MHz Note 4		0.28	1.02	mA
				f_{MX} = 8 MHz Note 4		TBD	TBD	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 5 TA≤ + 50°C		0.52	2.15	μA
				TA ≤ + 70°C			3.05	
				TA ≤ + 85°C			4.24	
				TA ≤ + 105°C			15.0	
				TA ≤ + 125°C			35.0	
	I_{DD3} Note 6	STOP mode	T_A ≤ + 50°C			0.22	2.05	μA
			T_A ≤ + 70°C				3.05	
			T_A ≤ + 85°C				4.24	
			T_A ≤ +105°C				15.0	
			T_A ≤ + 125°C				35.0	

Notes 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 in the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. During HALT instruction execution by flash memory.

3. When high-speed system clock and subsystem clock are stopped.

4. When high-speed on-chip oscillator and subsystem clock are stopped.

5. When operating real-time clock (RTC) and setting ultra-low current consumption (AMPHS1 = 1). When high-speed on-chip oscillator and high-speed system clock are stopped. When watchdog timer is stopped.

6. When high-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped. When watchdog timer is stopped.

7. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.
High speed operation: V_{DD} = 2.7 to 5.5 V@1 MHz to 24 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: High-speed on-chip oscillator clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

4. Temperature condition of the TYP. value is T_A = 25°C

Caution The pins mounted depend on the product.

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
RTC operating current	I_{RTC} Notes 1, 2	f_{SUB} = 32.768 kHz	Real-time clock operation		0.02	0.17	μA
			Interval timer operation		0.02	0.37	μA
WUTM operating current	I_{WUTM}	f_{IL} = 15 kHz			0.25	0.6	μA
Watchdog timer operating current	I_{WDT} Notes 2, 3	f_{IL} = 15 kHz			0.22	0.6	μA
A/D converter operating current	I_{ADC} Note 4	at maximum conversion speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		1.3	1.7	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.5	0.7	mA
		Internal reference voltage selected Note 7			75		μA
LVD operating current	I_{LVI} Note 5				0.08	0.26	μA
Temperature sensor operating current	I_{TMPS}				75		μA
BGO operating current	I_{BGO} Note 6				2.5	12.2	mA

Notes 1. Current flowing only to the real-time clock (excluding the operating current of the XT1 oscillator). The TYP. value of the current value of the RL78/F12 is the sum of the TYP. values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock operates in operation mode or HALT mode. The I_{DD1} and I_{DD2} MAX. values also include the real-time clock operating current. When the real-time clock operates during f_{CLK} = f_{SUB}, the TYP. value of I_{DD2} includes the real-time clock operating current.

2. When high-speed on-chip oscillator and high-speed system clock are stopped.

3. Current flowing only to the watchdog timer (including the operating current of the 15 kHz on-chip oscillator). The current value of the RL78/F12 is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when f_{CLK} = f_{SUB} when the watchdog timer operates in STOP mode.

4. Current flowing only to the A/D converter. The current value of the RL78/F12 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.

5. Current flowing only to the LVD circuit. The current value of the RL78/F12 is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVD circuit operates in the Operating, HALT or STOP mode.

6. Current flowing only to the BGO. The current value of the RL78/F12 is the sum of I_{DD1} or I_{DD2} and I_{BGO} when the BGO operates in an operation mode or the HALT mode.

7. This indicates operating current which increases when the internal reference voltage is selected. The Current flows even if the conversion is stopped.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

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

4. Temperature condition of the TYP. value is T_A = 25°C

Caution The pins mounted depend on the product.

32.5 AC Characteristics

32.5.1 Basic operation

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation High-speed main mode	0.04		1	μs
		Subsystem clock (f_{SUB}) operation	28.5	30.5	34.5	μs
External main system clock frequency	f_{EX}		1		20	MHz
	f_{EXS}		29		35	kHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}		24			ns
	t_{EXHS}, t_{EXLS}		13.7			μs
TI00 to TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}		2/f_{MCK} +10			ns
TO00 to TO07 output frequency	f_{TO}	4.0 V ≤ EV_{DD} ≤ 5.5 V			16	MHz
		2.7 V ≤ EV_{DD} < 4.0 V			8	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	4.0 V ≤ EV_{DD} ≤ 5.5 V			16	MHz
		2.7 V ≤ EV_{DD} < 4.0 V			8	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP11	1			μs
Key interrupt input low-level width	t_{KR}	KR0 to KR7	250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}		10			μs

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock set by the CKS0n bit of Timer mode register 0n (TMR0n). n: Channel number (n = 0 to 7))

Caution The pins mounted depend on the product.

32.6 Peripheral Functions Characteristics

32.6.1 Serial array unit

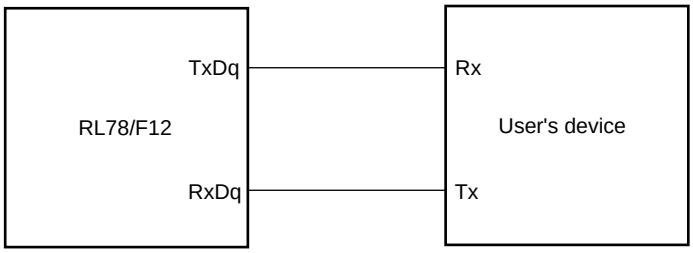
(1) During communication at same potential (UART mode) (dedicated baud rate generator output)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

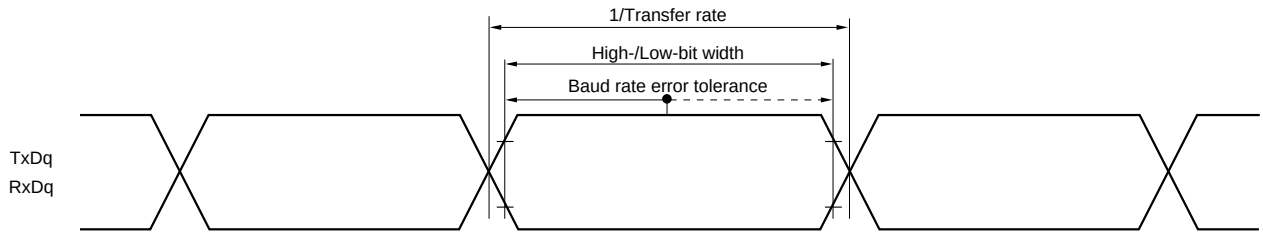
<R>

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		Other than SNOOZE mode	f_{MCK}/256		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate			4.0	Mbps
		Receivable baud rate at SNOOZE mode	4800		4800	bps

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



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 h (POMh).

Remarks 1. q: UART number (q = 0 to 2, S0), g: PIM number (g = 0, 1, 5, 7), h: POM number (h = 0, 1, 5, 7)

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 = 00 to 03, 10 to 11, S0, S1))

Caution The pins mounted depend on the product.

(2) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$: internal clock output)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time Note 1	t_{KCY1}	2.7 V ≤ EV_{DD} ≤ 5.5 V	CSI00 Note 1	125			ns
			Other than CSI00 Note 2	166.6			ns
$\overline{SCKp}$ high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY1}/2 – 12			ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY1}/2 – 18			ns
SIp setup time (to $\overline{SCKp}$↑) Note 3	t_{SIK1}			44			ns
							ns
SIp hold time (from $\overline{SCKp}$↑) Note 3	t_{KSI1}			19			ns
SOp output delay time Note 4 (from $\overline{SCKp}$↓)	t_{KSO1}	C = 30 pF Note 5				25	ns

Notes 1. The value must also be 2/f_{CLK} or more.

2. The value must also be 4/f_{CLK} or more.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

5. C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and $\overline{SCKp}$ pin by using port input mode register g (PIMg) and port output mode register h (POMh).

Remark p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), g: PIM number (g = 0, 1, 5, 7), h: POM number (h = 0, 1, 5, 7)

Caution The pins mounted depend on the product.

(3) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$: external clock input)

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY2}	4.0 V ≤ EV_{DD} ≤ 5.5 V	f_{MCK} > 20 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}			
		2.7 V ≤ EV_{DD} < 4.0 V	f_{MCK} > 16 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
$\overline{SCKp}$ high-/low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2			ns
SIp setup time (to $\overline{SCKp}$↑) **Note 1**	t_{SIK2}	2.7 V ≤ EV_{DD} ≤ 5.5 V		1/f_{MCK}+20			ns
SIp hold time (from $\overline{SCKp}$↑) **Note 1**	t_{KSI2}	2.7 V ≤ EV_{DD} ≤ 5.5 V		1/f_{MCK}+31			ns
SOp output Delay time (from $\overline{SCKp}$↓) **Note 2**	t_{KSO2}	C = 30 pF **Note 3**	2.7 V ≤ EV_{DD} ≤ 5.5 V			2/f_{MCK}+44	ns

Notes 1. This applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time is "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. This applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output is "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. C is the load capacitance of the SOp output lines.

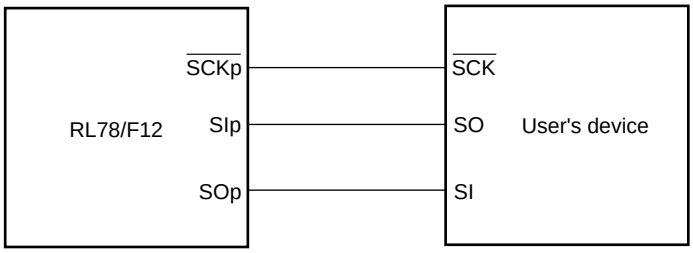
Caution Select the TTL input buffer for the SIp pin and $\overline{SCKp}$ pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register h (POMh).

Remarks 1. p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1), m: Unit number (m = 0, 1, S), n: Channel number (n = 0 to 3), g: PIM number (g = 0, 1, 5), h: POM number (h = 0, 1, 5, 7)

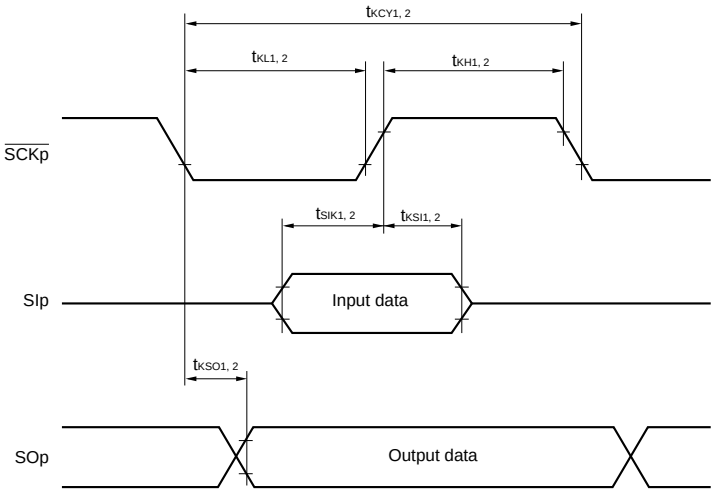
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03, 10, 11, S0, S1))

Caution The pins mounted depend on the product.

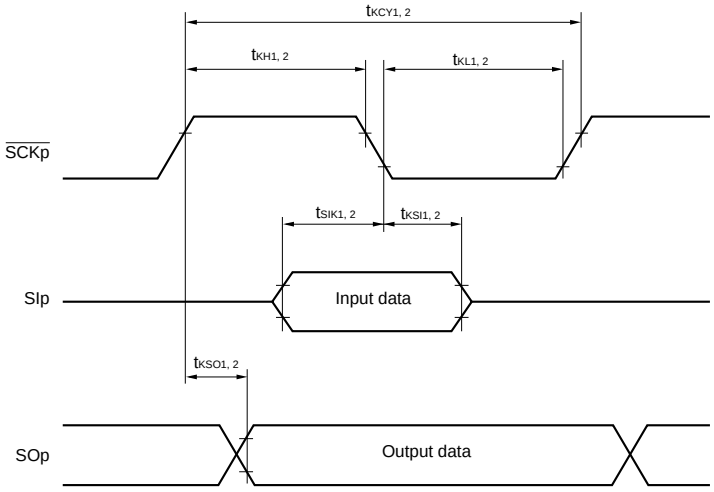
CSI mode connection diagram (during communication at same potential)



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.)



Remarks 1. p: CSI number (p = 00, 01, 10, 11, 20, 21, S0, S1)

2. m: Unit number, n: Channel number (mn = 00 to 03, 10, 11, S0, S1)

Caution The pins mounted depend on the product.

(4) During communication at same potential (simplified I^2C mode)

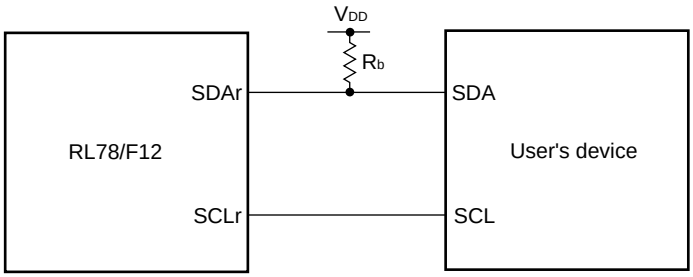
(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ EV_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ		400	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ EV_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	1150		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ EV_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	1150		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	$1/f_{MCK}$ + 145 **Note**		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	0	355	ns

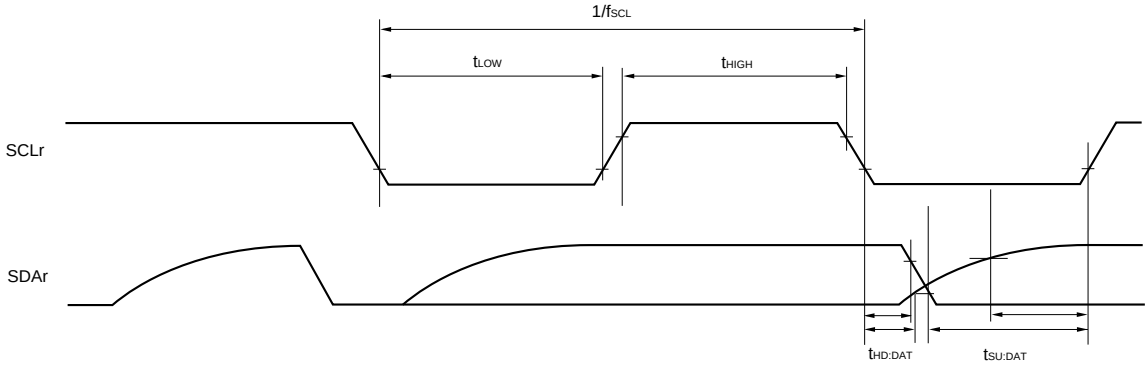
Note The value of f_{MCK} must be such that this does not exceed the hold time for SCLr = L or the hold time for SCLr = H.

Caution The pins mounted depend on the product.

Simplified I²C mode mode connection diagram (during communication at same potential)



Simplified I²C mode serial transfer timing (during communication at same potential)



Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

Remarks 1. R_b[Ω]:Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

2. r: IIC number (r = 00, 01, 11, 20, 21), g: PIM number (g = 0, 1, 5), h: POM number (h = 0, 1, 5, 7)

3. fMCK: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11, S0, S1)

Caution The pins mounted depend on the product.

32.6.2 Serial interface IICA

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	Standard Mode		Fast Mode		Fast Mode Plus		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode plus:f_{CLK} ≥ 10 MHz					0	1000	kHz
		Fast mode:f_{CLK} ≥ 3.5 MHz			0	400			kHz
		Normal mode:f_{CLK} ≥ 1 MHz	0	100					kHz
Setup time of restart condition Note 1	$t_{SU:STA}$		4.7		0.6		0.26		μs
Hold time	$t_{HD:STA}$		4.0		0.6		0.26		μs
Hold time when SCLA0 = “L”	t_{LOW}		4.7		1.3		0.5		μs
Hold time when SCLA0 = “H”	t_{HIGH}		4.0		0.6		0.26		μs
Data setup time (reception)	$t_{SU:DAT}$		250		100		50		ns
Data hold time (transmission) Note 2	$t_{HD:DAT}$		0	3.45	0	0.9	0		μs
Setup time of stop condition	$t_{SU:STO}$		4.0		0.6		0.26		μs
Bus-free time	t_{BUF}		4.7		1.3		0.5		μs

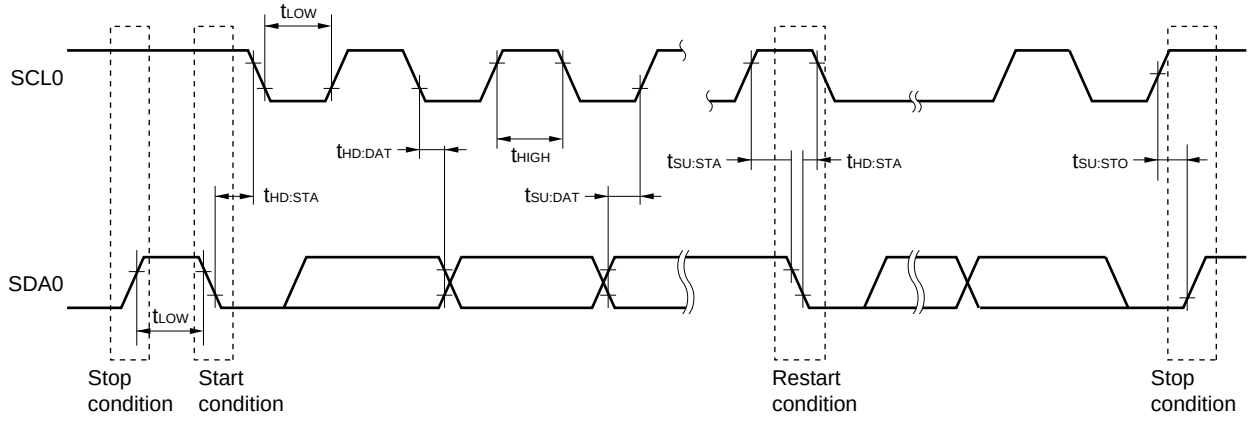
Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the $\overline{ACK}$ (acknowledge) timing.

Remark The maximum value of Cb (communication line capacitance) and the value of Rb (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
Fast mode: C_b = 320 pF, R_b = 1.1 kΩ
Fast mode plus: C_b = 120 pF, R_b = 1.1 kΩ

IICA serial transfer timing



Caution The pins mounted depend on the product.

32.6.3 LIN-UART

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.[Note]	Unit
Transfer rate				1	Mbps

Note However, the upper limit is fCLK/8.

Caution The pins mounted depend on the product.

32.7 Analog Characteristics

32.7.1 A/D converter characteristics

(1) When the setting of AV_{REF} (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1) and AV_{REF} (-) = AV_{REFM}/ANI1 (ADREFM = 1), this applies to the following ANI pins: ANI2 to ANI7 (the ANI pins for which V_{DD} is the power-supply voltage).

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, reference voltage (+) = AV_{REFP}, reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error Note 1	AINL	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V		1.2	±3.0	LSB
			2.7 V ≤ V_{DD} < 4.0 V		1.2	±3.5	LSB
Conversion time	t_{CONV}	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±0.25	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±0.25	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±1.5	LSB
Reference voltage (+)	AV_{REFP}			2.7		V_{DD}	V
Reference voltage (-)	AV_{REFM}			0			V
Analog input voltage	V_{AIN}			AV_{REFM}		AV_{REFP}	V
	V_{BGR}	2.7 V ≤ V_{DD} ≤ 5.5 V		1.38	1.45	1.5	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to 2.1.1, 20-pin products to 2.1.5, 64-pin products, and 2.1.6, Pins for each product (pins other than port pins).

(2) When the setting of AV_{REF} (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1) and AV_{REF} (-) = AV_{REFM}/ANI1 (ADREFM = 1), this applies to the following ANI pins: ANI16 to ANI19 (the ANI pins for which EV_{DD0} is the power-supply voltage).

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, reference voltage (+) = AV_{REFP}, reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error Note 1	AINL	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V		1.2	±4.5	LSB
			2.7 V ≤ V_{DD} < 4.0 V		1.2	±5.0	LSB
Conversion time	t_{CONV}	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±0.35	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±0.35	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±3.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Reference voltage (+)	AV_{REFP}			2.7		V_{DD}	V
Reference voltage (-)	AV_{REFM}			0			V
Analog input voltage	V_{AIN}			AV_{REFM}		AV_{REFP}	V
	V_{BGR}	2.7 V ≤ V_{DD} ≤ 5.5 V		1.38	1.45	1.5	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to 2.1.1, 20-pin products to 2.1.5, 64-pin products, and 2.1.6, Pins for each product (pins other than port pins).

(3) When the setting of AV_{REF} (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0) and AV_{REF} (-) = V_{SS} (ADREFM = 0), this applies to the following ANI pins: ANI0 to ANI7.

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, reference voltage (+) = V_{DD}, reference voltage (-) = V_{SS})

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}				8		10	bit
Overall error Note 1	AINL	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V	ANI0-ANI7		1.2	±5.0	LSB
			2.7 V ≤ V_{DD} < 5.5 V	ANI0-ANI7		1.2	±5.5	LSB
Conversion time	t_{CONV}	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V		2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V		3.1875		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V				±0.5	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V				±0.5	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V				±3.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V				±2.0	LSB
Reference voltage (+)	AV_{REFP}				V_{DD}			V
Reference voltage (-)	AV_{REFM}				V_{SS}			V
Analog input voltage	V_{AIN}	ANI0-ANI7			V_{SS}		V_{DD}	V
	V_{BGR}	2.7 V ≤ V_{DD} ≤ 5.5 V			1.38	1.45	1.5	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to 2.1.1, 20-pin products to 2.1.5, 64-pin products, and 2.1.6, Pins for each product (pins other than port pins).

(4) When the setting of AV_{REF} (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0) and AV_{REF} (-) = V_{SS} (ADREFM = 0), this applies to the following ANI pins: ANI16 to ANI19.

(T_A = –40 to +125°C, 2.7 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, reference voltage (+) = V_{DD}, reference voltage (-) = V_{SS})

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}				8		10	bit
Overall error Note 1	AINL	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V	ANI16-ANI19		1.2	±6.5	LSB
			2.7 V ≤ V_{DD} < 5.5 V	ANI16-ANI19		1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution	4.0 V ≤ V_{DD} ≤ 5.5 V		2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V		3.1875		39	μs
Zero-scale error Notes 1, 2	EZS	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V				±0.60	%FSR
Full-scale error Notes 1, 2	EFS	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V				±0.60	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V				±4.0	LSB
Differential linearity error Note 1	DLE	10-bit resolution	2.7 V ≤ V_{DD} ≤ 5.5 V				±2.0	LSB
Reference voltage (+)	AV_{REFP}				V_{DD}			V
Reference voltage (-)	AV_{REFM}				V_{SS}			V
Analog input voltage	V_{AIN}	ANI16-ANI19			V_{SS}		V_{DD}	V
	V_{BGR}	2.7 V ≤ V_{DD} ≤ 5.5 V			1.38	1.45	1.5	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The pins mounted depend on the product. Refer to 2.1.1, 20-pin products to 2.1.5, 64-pin products, and 2.1.6, Pins for each product (pins other than port pins).

Caution The pins mounted depend on the product.

32.7.2 Temperature sensor characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Reference output voltage	V_{CONST}	Setting ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	F_{VTMPS}	Temperature sensor that depends on the temperature		–3.6		mV/C
Operation stabilization wait time	t_{AMP}				5	μs

32.7.3 POR circuit characteristics

(T_A = –40 to +125°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.46	1.51	1.59	V
	V_{PDR}	Power supply fall time	1.45	1.50	1.58	V
Minimum pulse width	T_{PW}		300			μs
Detection delay time	T_{PD}				350	μs

32.7.4 LVD circuit characteristics

(a) Characteristics for LVD Detection at Reset and Interrupt modes

(T_A = –40 to +125°C, $V_{PDR} \leq V_{DD} = EV_{DD} \leq 5.5$ V, $V_{SS} = EV_{SS}$ = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}	Power supply rise time	3.96	4.06	4.25	V
			Power supply fall time	3.89	3.98	4.15	V
		V_{LVI1}	Power supply rise time	3.66	3.75	3.93	V
			Power supply fall time	3.58	3.67	3.83	V
		V_{LVI2}	Power supply rise time	3.06	3.13	3.28	V
			Power supply fall time	2.99	3.06	3.20	V
		V_{LVI3}	Power supply rise time	2.95	3.02	3.17	V
			Power supply fall time	2.89	2.96	3.09	V
		V_{LVI4}	Power supply rise time	2.85	2.92	3.07	V
			Power supply fall time	2.79	2.86	2.99	V
		V_{LVI5}	Power supply rise time	2.74	2.81	2.95	V
			Power supply fall time	2.68[Note]	2.75	2.88	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time		t_{LD}				300	μs

Note The minimum value lowers the minimum guaranteed voltage for operation(2.7V). However, LVD detection performs in the same way as in normal mode (operation according to the same specification when VDD is 2.7V) until it is reset at reset mode.

Remark $V_{LVI(n-1)} > V_{LVIn}$: n = 1 to 13
The following relationship is formed under the same temperature conditions: the detection voltage at power supply rise time > the detection voltage at power supply fall time.

Caution The pins mounted depend on the product.

(b) LVD Detection Voltage of Interrupt & Reset Mode

(T_A = –40 to +125°C, $V_{PDR} \leq V_{DD} = EV_{DD} \leq 5.5$ V, $V_{SS} = EV_{SS}$ = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVI5}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 1, 1, falling reset voltage: 2.7 V		2.68 Note	2.75	2.88	V
	V_{LVI4}	LVIS1, LVIS0 = 1, 0	Rising release reset voltage	2.85	2.92	3.07	V
			Falling interrupt voltage	2.79	2.86	2.99	V
	V_{LVI3}	LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.95	3.02	3.17	V
			Falling interrupt voltage	2.89	2.96	3.09	V
	V_{LVI0}	LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.96	4.06	4.25	V
			Falling interrupt voltage	3.89	3.98	4.15	V

Note The minimum value lowers the minimum guaranteed voltage for operation(2.7V). However, LVD detection performs in the same way as in normal mode (operation according to the same specification when VDD is 2.7V) until it is reset at reset mode.

Remark The following relationship is formed under the same temperature conditions: the rising release reset voltage > the falling interrupt voltage > the falling reset voltage

<R> **32.7.5 Power supply rise time**

(T_A = –40 to +125°C, $V_{SS} = EV_{SS}$ = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum slew rate for the supply voltage to rise	S_{vrmax}	0 V → 2.7 V (CMODE0 = 1) (VPOC2 = 0 or 1)			50 Note 1	V/ms
Minimum slew rate for the supply voltage to rise Note 2	S_{vrmin}	0 V → 2.7 V (CMODE0 = 1)	6.5 Note 1			V/ms

Notes 1. In case the supply voltage falls to a level of V_{PDR} or below and a power-on reset is generated, the slew rate must not exceed the value S_{vrmax} even if the supply voltage does not go down to 0 V.

2. The minimum slew rate for the supply voltage (S_{vrmin}) must be met when the voltage detector (LVD) is not used (option byte bit VPOC2 = 1) and an external reset circuit releases before the supply voltage reaches V_{DD} (MIN.) (here 2.7 V).

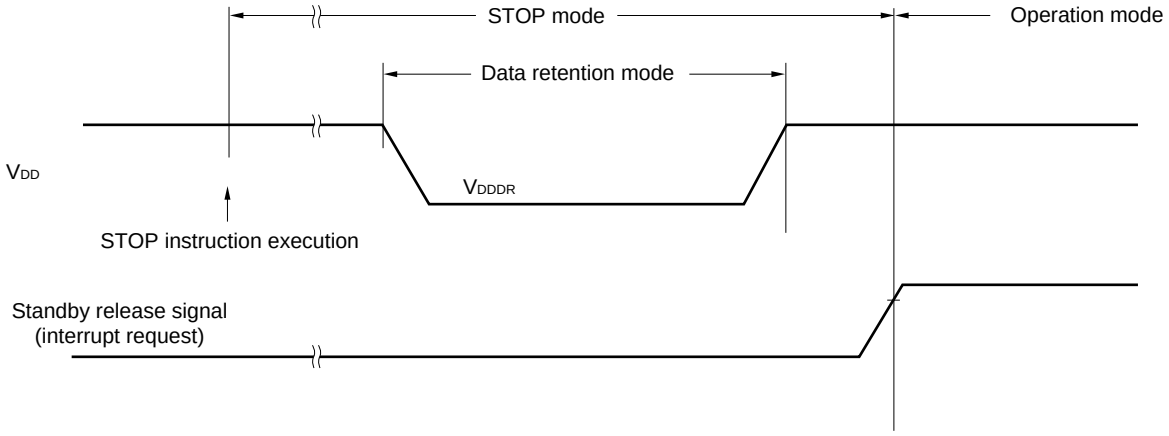
Caution The pins mounted depend on the product.

32.8 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics

(T_A = –40 to +125°C V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}	STOP mode	1.45 Note		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is effected, but data is not retained when a POR reset is effected.



32.9 Flash Memory Programming Characteristics

(T_A = –40 to +125°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency		f_{CLK}		1		24	MHz
<R> Number of code flash rewrites Notes 1, 2, 3		C_{erwr}	20 years retention (after rewrite) T_A = +85°C Note 4	1000			Times
<R> Number of data flash rewrites Notes 1, 2, 3			20 years retention (after rewrite) T_A = +85°C Note 4	10000			
			5 years retention (after rewrite) T_A = +85°C Note 4	100000			
Erase time	Block erase	T_{erasa}		5			ms
write time		T_{wrwa}		10			μs

Notes 1. Retention years indicate a period between time for a rewrite and the next.

2. When using flash memory programmer and Renesas Electronics self programming library.

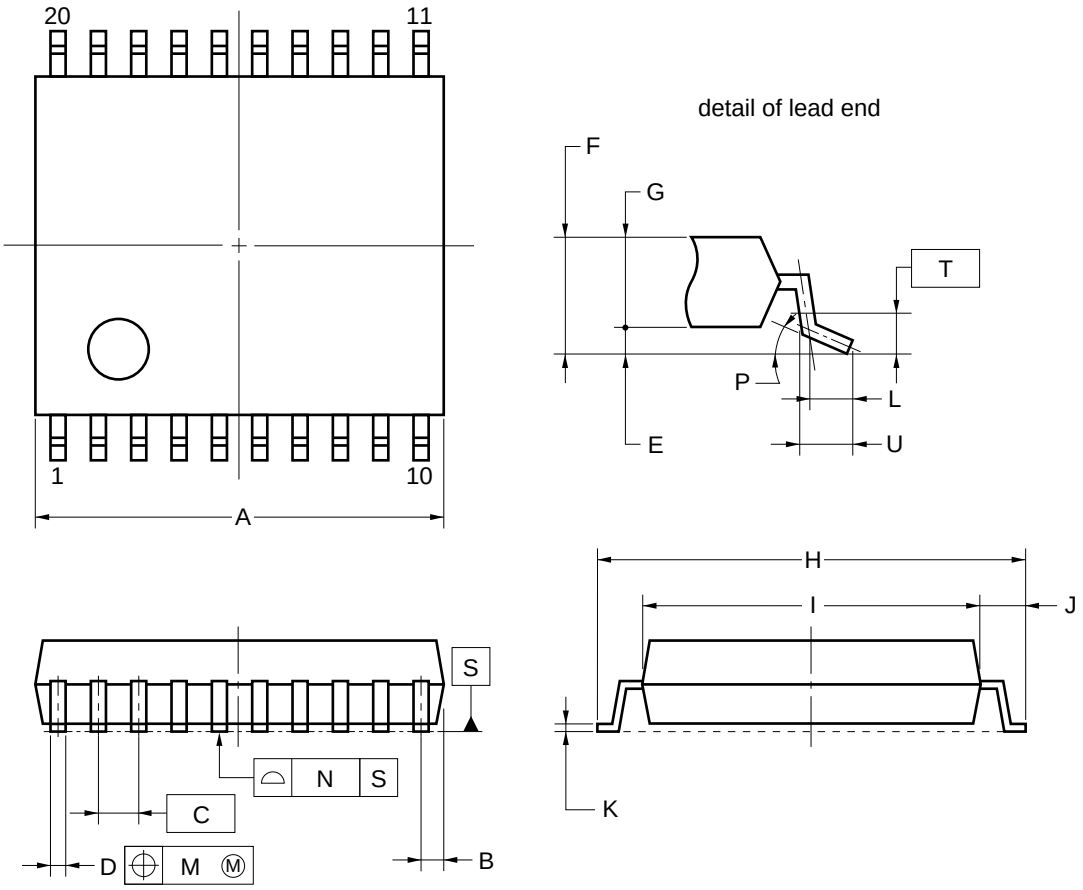
3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas.

<R> **4.** The specified data retention time is given under the condition that the average temperature (T_A) is 85°C or below.

CHAPTER 33 PACKAGE DRAWING

33.1 20-pin products

20-PIN PLASTIC SSOP (7.62 mm (300))



NOTE

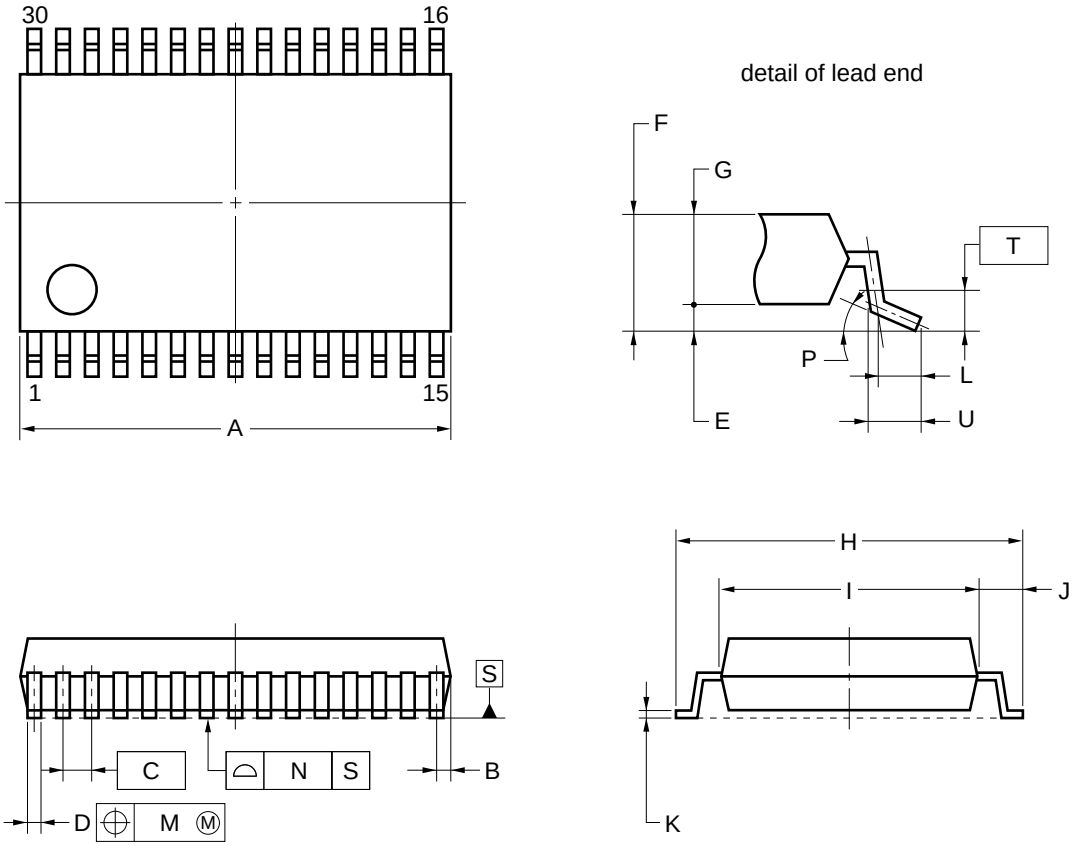
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	6.65±0.15
B	0.475 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

S20MC-65-5A4-2

33.2 30-pin products

30-PIN PLASTIC SSOP (7.62 mm (300))



NOTE

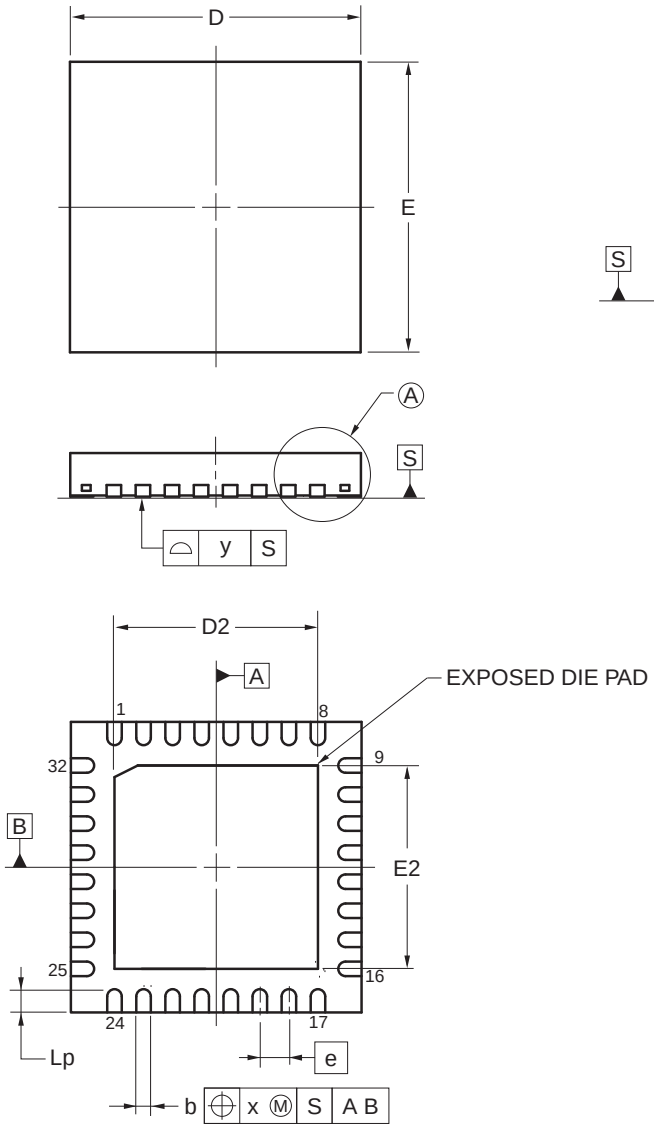
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	9.85±0.15
B	0.45 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

S30MC-65-5A4-2

33.3 32-pin products

32-PIN PLASTIC WQFN(5x5)



(UNIT:mm)

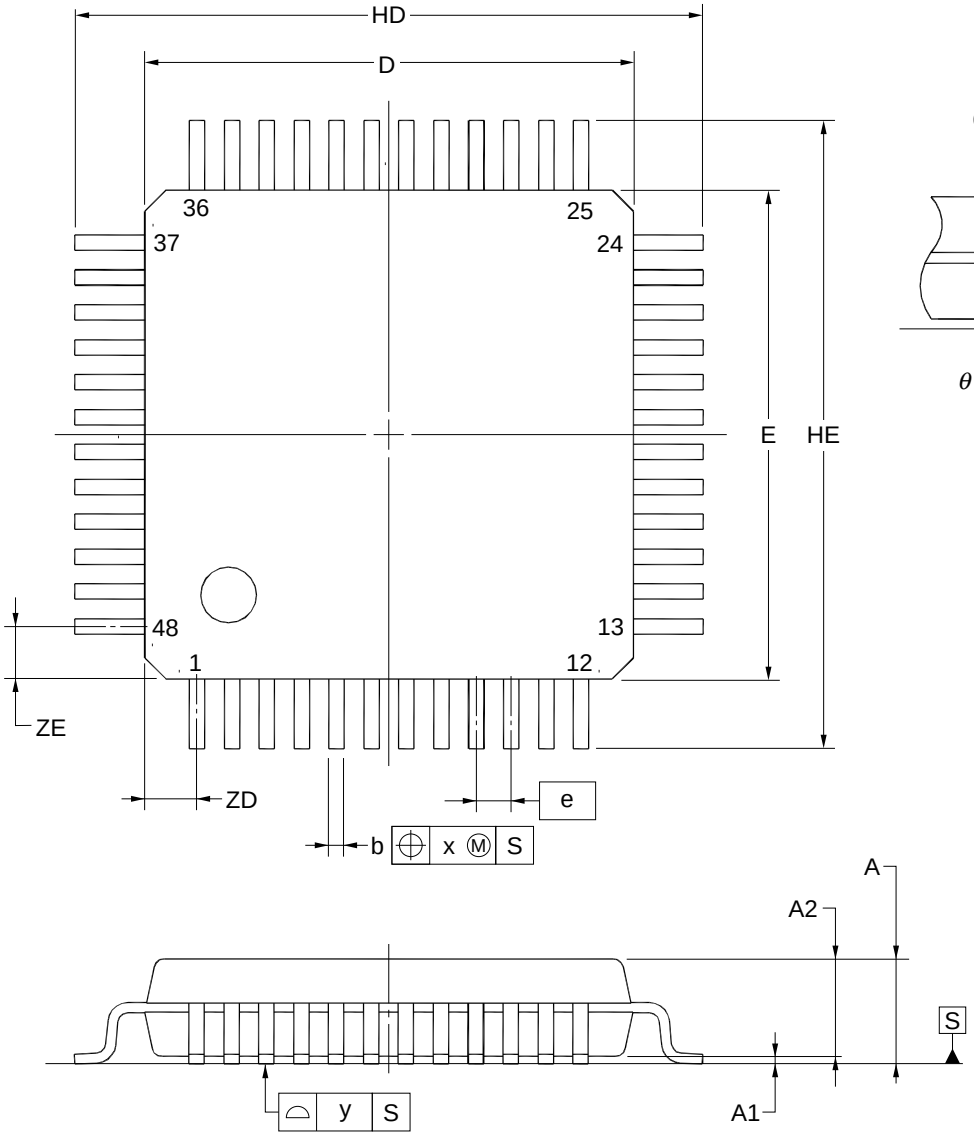
ITEM	DIMENSIONS
D	5.00±0.05
E	5.00±0.05
A	0.75±0.05
b	$0.25^{+0.05}_{-0.07}$
e	0.50
Lp	0.40±0.10
x	0.05
y	0.05

P32K8-50-3B4-2

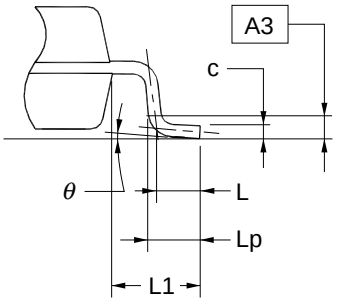
ITEM		D2			E2		
		MIN	NOM	MAX	MIN	NOM	MAX
EXPOSED DIE PAD VARIATIONS	A	3.45	3.50	3.55	3.45	3.50	3.55

33.4 48-pin products

48-PIN PLASTIC LQFP (FINE PITCH)(7x7)



detail of lead end



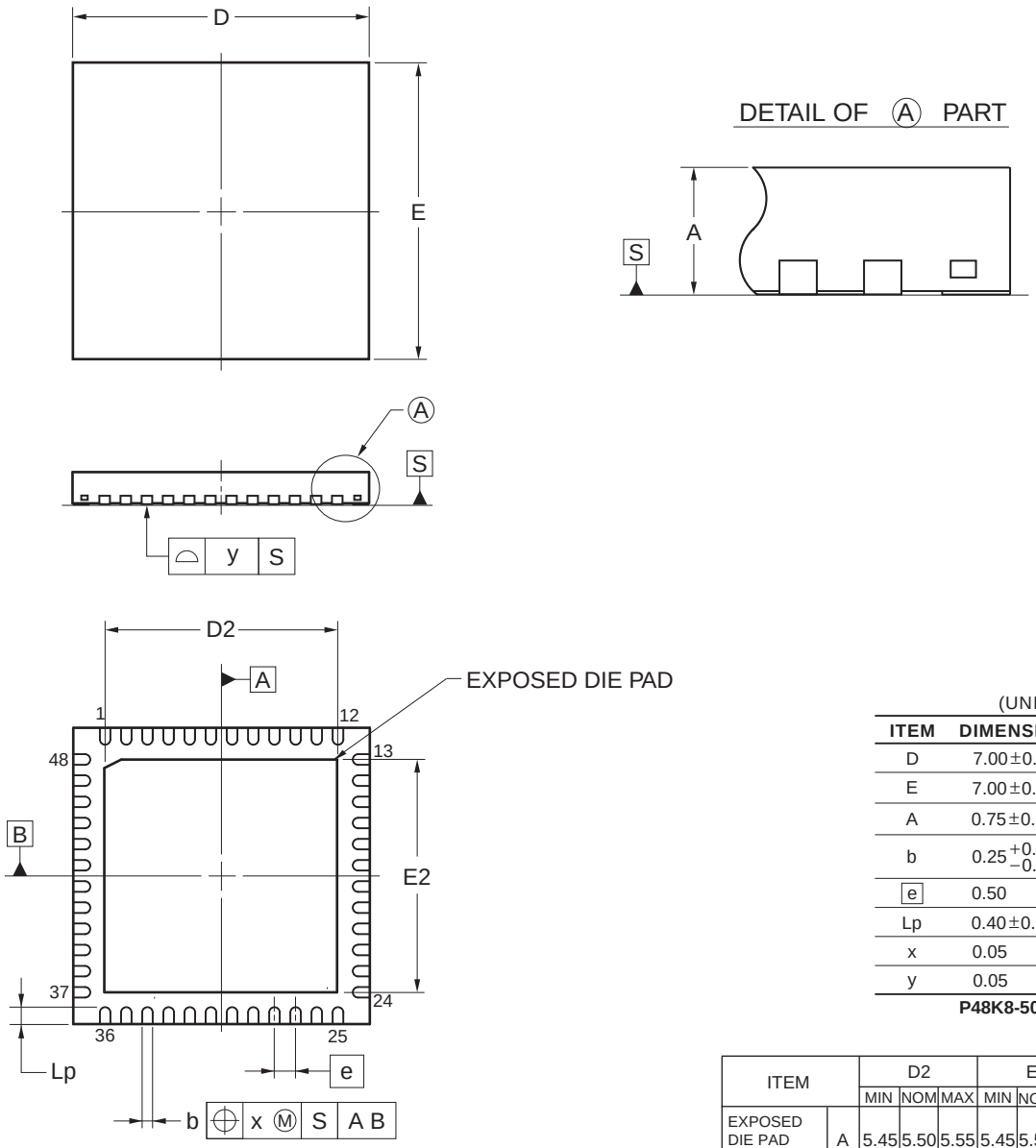
(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.20
E	7.00±0.20
HD	9.00±0.20
HE	9.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75

P48GA-50-8EU

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

48-PIN PLASTIC WQFN(7x7)



(UNIT:mm)

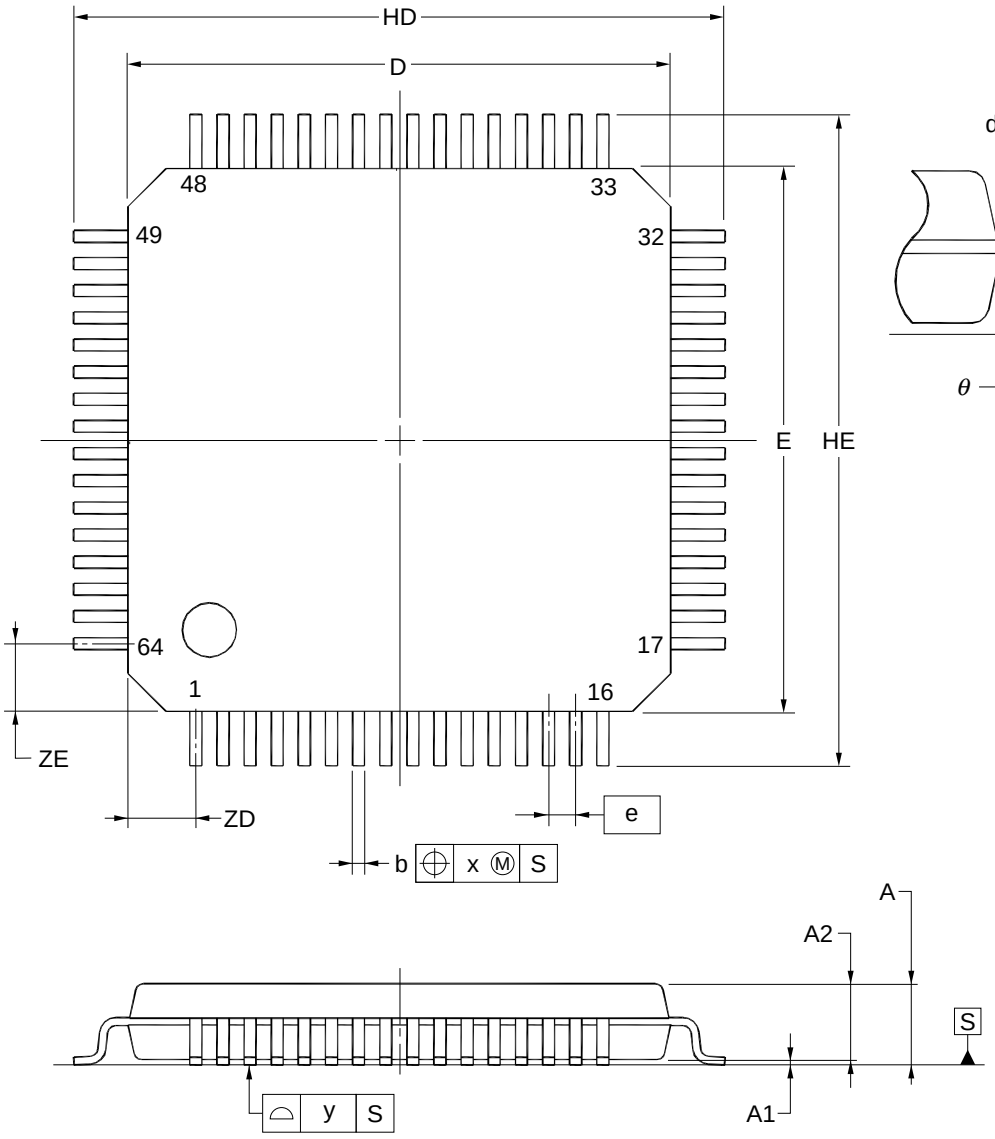
ITEM	DIMENSIONS
D	7.00±0.05
E	7.00±0.05
A	0.75±0.05
b	$0.25^{+0.05}_{-0.07}$
e	0.50
Lp	0.40±0.10
x	0.05
y	0.05

P48K8-50-5B4-3

ITEM		D2			E2		
		MIN	NOM	MAX	MIN	NOM	MAX
EXPOSED DIE PAD VARIATIONS	A	5.45	5.50	5.55	5.45	5.50	5.55

33.5 64-pin products

64-PIN PLASTIC LQFP(FINE PITCH)(10x10)



(UNIT:mm)

ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

P64GB-50-UEU-1

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

(1/2)

Page	Description	Classification
How to Use This Manual		
d, e	The whole descriptions changed.	(c), (e)
CHAPTER 1 OUTLINE		
p.3	1.2 Ordering Information: Note added.	(c)
p.8	48-pin plastic WQFN (7 × 7): Remark 3 added.	(c)
CHAPTER 2 PIN FUNCTIONS		
p.27	2.1.6 Pins for each product (pins other than port pins): Descriptions changed.	(c)
CHAPTER 3 CPU ARCHITECTURE		
p.44 to 49	Figure 3-1. Memory Map (R5F10968) to Figure 3-6. Memory Map (R5F109xE (x = 6, A, B, G, L)): Description of Note 1 modified.	(c)
CHAPTER 5 CLOCK GENERATOR		
p.179	Figure 5-2. Format of Clock Operation Mode Control Register (CMC): Descriptions of the AMPH bit and Caution 3 changed.	(b)
p.191	Figure 5-9. Format of High-Speed On-Chip Oscillator Frequency Select Register (HOCODIV): incorrect descriptions modified.	(a)
p.200	Figure 5-15. Clock Generator Operation When Power Supply Voltage Is Turned On: Descriptions of Figure and Note 3 modified.	(b), (c)
CHAPTER 12 A/D CONVERTER		
p.392 to 397	Table 12-3. A/D Conversion Time Selection (1/6) to Table 12-3. A/D Conversion Time Selection (6/6): Descriptions added.	(c)
CHAPTER 13 SERIAL ARRAY UNIT		
p.459	Figure 13-7. Format of Serial Clock Select Register m (SPSm): Incorrect description of Note modified.	(c)
p.478	13.3 (14) Serial standby control register 0 (SSC0): Caution modified.	(c)
p.568	Table 13-2. Selection of Operation Clock For 3-Wire Serial I/O: Incorrect description of Note modified.	(c)
p.593	Figure 13-102. Flowchart of UART Reception: Incorrect description in flowchart modified.	(a)
p.594, 595	13.6.3 SNOOZE mode function (only UART0 reception): Descriptions added.	(c)
p.635	Table 13-5. Selection of Operation Clock For Simplified I^2C: Incorrect description of Note modified.	(c)
CHAPTER 14 ASYNCHRONOUS SERIAL INTERFACE LIN-UART (UARTF)		
p.682	Figure 14-30. BF Transmission Processing Flow: Incorrect description of Note modified.	(a)
p.714	14.7.4 Automatic checksum function: Incorrect description of Note modified.	(c)
p.715	Figure 14-61. Automatic Checksum Error Occurrence Example (Response Reception): Incorrect description modified.	(a)
CHAPTER 18 INTERRUPT FUNCTIONS		
p.865	Table 18-2. Flags Corresponding to Interrupt Request Sources (5/5): Incorrect descriptions modified.	(a)
p.866	Figure 18-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (64-pin) (1/2): Incorrect descriptions modified.	(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
CHAPTER 18 INTERRUPT FUNCTIONS		
p.868	Figure 18-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (64-pin): Incorrect descriptions modified.	(a)
p.870	Figure 18-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (64-pin) (2/2): Incorrect descriptions modified.	(a)
CHAPTER 20 STANDBY FUNCTION		
p.896, 897	Figure 20-5. STOP Mode Release by Interrupt Request Generation: Descriptions modified.	(c)
p.899	20.2.3 SNOOZE mode: Descriptions modified.	(b)
CHAPTER 22 POWER-ON-RESET CIRCUIT		
p.916	Figure 22-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/2): Descriptions of Notes in Figure modified.	(c)
CHAPTER 27 FLASH MEMORY		
p.977	27.4.3 Procedure for accessing data flash memory: Caution 4 added.	(b)
CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)		
p.1016	Caution 1 deleted.	(c)
p.1031	31.6.1 (1) During communication at same potential (UART mode) (dedicated baud rate generator output): Incorrect descriptions modified.	(b)
p.1032	31.6.1 (2) During communication at same potential (CSI mode) (master mode, $\overline{\text{SCK}}$p: internal clock output): Incorrect descriptions modified.	(a)
p.1046	31.7.5 Power supply rise time: Descriptions modified.	(c)
p.1047	31.9 Flash Memory Programming Characteristics: Descriptions modified.	(c)
CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)		
p.1048	Caution 1 deleted.	(c)
p.1063	32.6.1 (1) During communication at same potential (UART mode) (dedicated baud rate generator output): Incorrect descriptions modified.	(b)
p.1077	32.7.5 Power supply rise time: Descriptions modified.	(c)
p.1078	32.9 Flash Memory Programming Characteristics: Descriptions modified. Note 4 added.	(c)

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 Edition

(1/2)

Edition	Description	Chapter
Rev.1.01	1.6 Outline of Functions (2/2): Description of 8/10-bit resolution A/D converter, number of vectored interrupt sources, power-on-reset circuit and voltage detector modified.	CHAPTER 1 OUTLINE
	2.1.6 Pins for each product (pins other than port pins) (1/3): PCLBUZ0D and REGC modified.	CHAPTER 2 PIN FUNCTIONS
	2.2.14 REGC: Description modified.	
	4.2.6 Port 5: Description, Table 4-8. Register Settings When Using Port 5, and Notes 1 and 3 modified and Important added.	CHAPTER 4 PORT FUNCTIONS
	6.3 (10) Timer output register 0 (TO0): Description modified and Note added.	CHAPTER 6 TIMER ARRAY UNIT
	6.3 (15) Port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4): Description modified and Remark added.	
	Figure 7-17. Procedure for Starting Operation of Real-time Clock: Modified.	CHAPTER 7 REAL-TIME CLOCK
	Figure 7-19. Procedure for Reading Real-time Clock: Note 1 added.	
	11.1 Functions of Watchdog Timer: Description modified.	CHAPTER 11 WATCHDOG TIMER
	11.4.3 Setting window open period of watchdog timer: Remark modified. 11.4.4 Setting watchdog timer interval interrupt: Description modified. Table 11-5. Setting of Watchdog Timer Interval Interrupt: Description modified.	
	12.2 Configuration of A/D Converter: Description and Remark modified, and Caution added.	CHAPTER 12 A/D CONVERTER
	Figure 13-11. Serial Data Register mn (SDRmn) (mn = 00-03, 10, 11): Description modified.	CHAPTER 13 SERIAL ARRAY UNIT
	13.3 (12) Serial output register m (SOm): Description, Figure 13-19. Format of Serial Output Register m (SOm), and Caution modified.	
	Figure 13-27. Format of Port Mode Registers X0 to X4 (PMX0 to PMX4) (64-pin products): Description modified.	
	Figure 13-57. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (1/2): Description modified.	
	13.5.5 Slave reception: Interrupt, and Notes 1 and 2 modified.	
	13.5.6 Slave transmission/reception: Target channel, interrupt, and Notes 1 and 2 modified.	
	Figure 13-77. Procedure for Resuming Slave Transmission/Reception: Supplement modified.	
	13.5.7 SNOOZE mode function (only CSI00): Description and Caution modified.	
	13.8.1 Address field transmission: Error detection flag and Note modified.	
	Table 13-4. Selection of Operation Clock for Simplified I^2C: Description modified.	
	Figure 14-1. Block Diagram of Asynchronous Serial Interface LIN-UART: Description modified.	CHAPTER 14 ASYNCHRONOUS SERIAL INTERFACE LIN-UART (UARTF)
	14.5.6 BF reception: Description modified.	
	Figure 14-54. Example of BF/SF Reception Failure: Description modified.	
	Figure 14-55. Example of Successful BF, SF, and PID reception: Description modified. Figure 14-56. Example of Successful BF Reception During SF Reception (No PID Reception Error): Description modified.	
	Figure 16-5. Format of Multiplication/Division Control Register (MDUC): Description modified.	CHAPTER 16 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR
	Table 18-1. Interrupt Source List (1/3): Description modified.	CHAPTER 18 INTERRUPT FUNCTIONS

(2/2)

Edition	Description	Chapter
Rev.1.01	Figure 23-10. Delay from the Time LVD Reset Source is Generated until the Time LVD Reset Has been Generated or Released: Description modified.	CHAPTER 23 VOLTAGE DETECTOR
	Figure 27-1. Environment for Wiring Program to Flash Memory: Description modified. 27.1.2 Communication Mode: Transfer rate changed. Figure 27-2. Communication with Dedicated Flash Memory Programmer: Description modified, Note 1 deleted.	CHAPTER 27 FLASH MEMORY
	Figure 28-2. Memory Spaces Where Debug Monitor Programs Are Allocated: Description and Notes 1 to 3 modified, and Note 4 added.	CHAPTER 28 ON-CHIP DEBUG FUNCTION
	31.7.2 Temperature sensor characteristics: Description modified. 31.7.3 POR circuit characteristics: Description modified and Notes 1 and 2 added.	CHAPTER 31 ELECTRICAL SPECIFICATIONS (J GRADE)
	31.7.5 Supply Power Rise Time: Description modified.	
	32.7.2 Temperature sensor characteristics: Description modified.	CHAPTER 32 ELECTRICAL SPECIFICATIONS (K GRADE)
	32.7.5 LVD circuit characteristics: Remark modified and section number changed to 32.7.4.	
	32.7.4 Supply Power Rise Time: Description modified and chapter number changed to 32.7.5.	

RL78/F12 User’s Manual: Hardware

Publication Date: Rev.1.11 Jan 31, 2014

Published by: Renesas Electronics Corporation

Renesas Electronics Corporation

http://www.renesas.com

SALES OFFICES

Refer to "http://www.renesas.com/" for the latest and detailed information.

California Eastern Laboratories, Inc.
4590 Patrick Henry Drive, Santa Clara, California 95054, U.S.A.
Tel: +1-408-919-2500, Fax: +1-408-988-0279

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 D üsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, 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 Rd., Putuo District, Shanghai, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, 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 906, 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 Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.0

RL78/F12

Renesas Electronics Corporation

R01UH0231EJ0111

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Renesas Electronics:

R5F109AAJSP#V0