User's Manual

16

PD78F8040, 78F8041, 78F8042, 78F8043

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.3.00 Sep 2010

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "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.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. 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.
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.

NOTES FOR CMOS DEVICES

(1) VOLTAGE APPLICATION WAVEFORM AT INPUT PIN: Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).

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

(3) PRECAUTION AGAINST ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

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

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

(6) INPUT OF SIGNAL DURING POWER OFF STATE : Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the μPD78F8040, 78F8041, 78F8042, 78F8043 microcontrollers and design and develop application systems and programs for these devices.
The target products are as follows.

- μPD78F8040, 78F8041, 78F8042, 78F8043

Purpose

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

Organization

The manual for the μPD78F8040, 78F8041, 78F8042, 78F8043 microcontrollers is separated into two parts: this manual and the instructions edition (common to the 78K0R microcontrollers).

μPD78F8040, 78F8041, 78F8042, 78F8043 User's Manual (This Manual)	**78K0R Microcontroller User's Manual Instructions**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R.
- To know details of the 78K0R microcontroller instructions:
  → Refer to the separate document **78K0R Microcontroller Instructions User's Manual (U17792E)**.

Conventions

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

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

Documents Related to Devices

Document Name	Document No.
μPD78F8040, 78F8041, 78F8042, 78F8043 User's Manual	This manual
78K0R Microcontroller Instructions User's Manual	U17792E
78K0R Microcontroller Self Programming Library Type02 User's Manual Note	U19193E

Note This document is under engineering management. For details, consult an Renesas Electronics sales representative.

Documents Related to Development Tools (Software) (User's Manuals)

Document Name		Document No.
CC78K0R Ver. 2.00 C Compiler	Operation	U18549E
	Language	U18548E
RA78K0R Ver. 1.20 Assembler Package	Operation	U18547E
	Language	U18546E
SM+ System Simulator	Operation	U18010E
PM+ Ver. 6.30		U18416E
ID78K0R-QB Ver. 3.20 Integrated Debugger	Operation	U17839E

Documents Related to Development Tools (Hardware) (User's Manuals)

Document Name	Document No.
QB-MINI2 On-Chip Debug Emulator with Programming Function	U18371E
QB-78K0RKX3C In-Circuit Emulator	U19324E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP5 Flash Memory Programmer User's Manual	U18865E

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

Other Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE – Products and Packages –	X13769X
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

<R> **Note** See the “Semiconductor Device Mount Manual” website (http://www2.renesas.com/pkg/en/mount/index.html).

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

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

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

- Minimum instruction execution time can be changed from high speed (0.05 μs: @ 20 MHz operation with high-speed system clock) to low-speed (8 μs: @ internal high-speed oscillation divided clock operation)
- General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ROM, RAM capacities

Item / Part Number	Program Memory (ROM)		Data Memory (RAM)
μPD78F8040	Flash memory	32 KB	4 KB
μPD78F8041		64 KB	4 KB
μPD78F8042		96 KB	6 KB
μPD78F8043		128 KB	7 KB

- Built-in IO-Link transceiver
  - C/Q reverse polarity protection
  - Integrated 5 V voltage regulator
  - A baud rate of up to 230400 [bps] can be selected.
  - Wakeup detection function
  - 3.3 V / 5 V compatible digital interface
  - Overcurrent detection and shutoff threshold level selection
- On-chip internal high-speed oscillation clocks
  - 20 MHz Internal high-speed oscillation clock: 20 MHz (TYP.)
  - 8 MHz Internal high-speed oscillation clock: 8 MHz (TYP.)
  - 1 MHz Internal high-speed oscillation clock: 1 MHz (TYP.)
- 1 MHz/8 MHz/20 MHz On-chip internal high-speed oscillation clocks
- On-chip single-power-supply flash memory (with prohibition of chip erase/block erase/writing function)
- Self-programming (with boot swap function/flash shield window function)
- On-chip debug function
- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- On-chip watchdog timer (operable with the dedicated internal low-speed oscillation clock)
- On-chip multiplier/divider (16 bits × 16 bits, 32 bits ÷ 32 bits)
- On-chip BCD adjustment
- I/O ports: 26[Note] (N-ch open drain: 2)
- Timer: 13 channels
  - 16-bit timer: 12 channels
  - Watchdog timer: 1 channel
- Serial interface
  - IO-Link (use UART0): 1 channel
  - CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel
  - UART (LIN-bus supported): 1 channel
  - I^2C: 1 channel
- 10-bit resolution A/D converter (AV_{REF} = 1.8 to 5.5 V): 6 channels
- Power supply voltage: $V_{DD}/EV_{DD} = V_{DD\_IO}$ = 3.0 to 5.5 V, IV_{DDH} = 4.75 to 5.25 V, V_{DDH} = 8.0 to 36.0 V
- Operating ambient temperature: T_A = −40 to +85°C

Note Three of these pins (P11/RxD0, P50/INTP1, P51/INTP2) are used for IO-Link communication. They must be connected to RXD, ILIM, WAKE respectively on the PCB by users.

1.2 Applications

○ Industrial sensor and actuator

1.3 Ordering Information

- **Flash memory version**

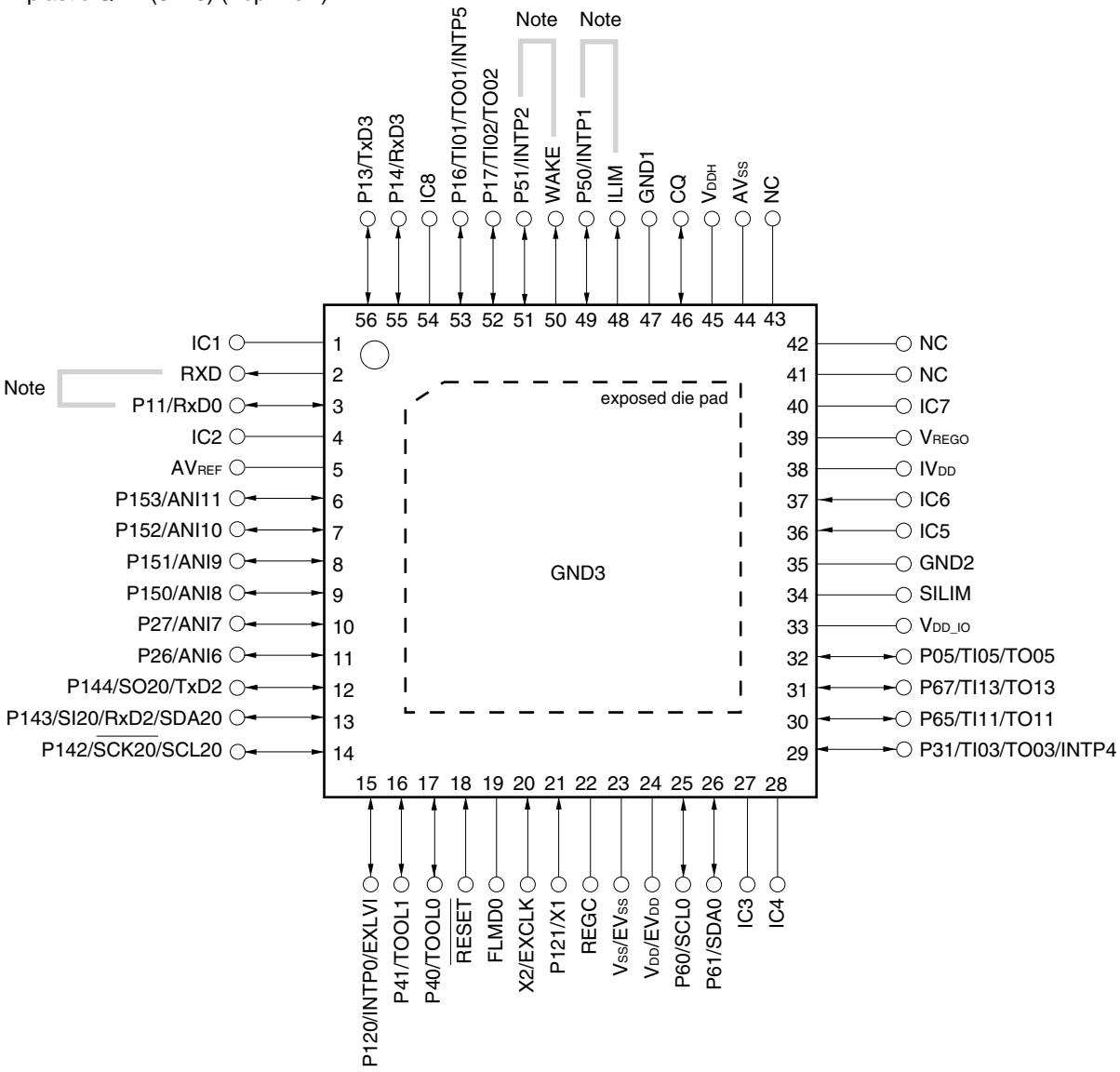
	Part Number	Package
	μPD78F8040K8-9B4-AX	56-pin plastic QFN (8 × 8)
	μPD78F8041K8-9B4-AX	56-pin plastic QFN (8 × 8)
	μPD78F8042K8-9B4-AX	56-pin plastic QFN (8 × 8)
	μPD78F8043K8-9B4-AX	56-pin plastic QFN (8 × 8)
<R>	μPD78F8040F1-AD1-AX Note	56-pin plastic FBGA (4 × 7)
<R>	μPD78F8041F1-AD1-AX Note	56-pin plastic FBGA (4 × 7)
<R>	μPD78F8042F1-AD1-AX Note	56-pin plastic FBGA (4 × 7)
<R>	μPD78F8043F1-AD1-AX Note	56-pin plastic FBGA (4 × 7)

Note Under development

Caution The μPD78F8040, 78F8041, 78F8042, 78F8043 have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

1.4 Pin Configuration

- 56-pin plastic QFN (8 × 8) (Top View)



Note Connect P11/RxD0, P50/INTP1, and P51/INTP2 to RXD, ILIM, and WAKE respectively on PCB by users.

Cautions 1. Make GND1, GND2, and AVSS the same potential as VSS/EVSS.

2. Make package exposed die pad (GND3) the same potential as VSS/EVSS.

3. Make VDD_IO the same potential as VDD/EVDD.

4. For details about how to handle IC1 to IC8, see Table 3-3 Connection of Unused Pins.

5. Connect the REGC pin to Vss/EVss via a capacitor (0.47 to 1 μF).

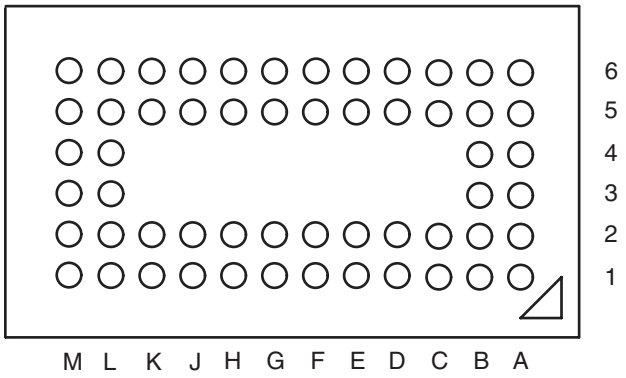
6. Connect the IVDD pin to Vss/EVss via a capacitor (330 to 2000 nF).

7. P26/ANI6, P27/ANI7, and P150/ANI8 to P153/ANI11 are set as analog inputs in the order of P153/ANI11, …, P150/ANI8, P27/ANI7, P26/ANI6 by the A/D port configuration register (ADPC). When using P26/ANI6, P27/ANI7, and P150/ANI8 to P153/ANI11 as analog inputs, start designing from P153/ANI11 (see 10.3 (6) A/D port configuration register (ADPC) for details).

8. Connect CQ pin and VDDH pin via a diode (required specification: breakthrough voltage: 70 V or more, forward current: 100 mA or more, reverse recovery time: 10 ns or less)

<R> • 56-pin plastic FBGA (4 × 7) (Bottom View)

Bottom View



P120/ INTP0/ EXLVI	P41/ TOOL1	FLMD0	X2/EXCLK	REGC	P60/SCL0	P61/SDA0	P31/TI03/ TO03/ INTP4	V_{DD_IO}	SILIM	GND2	IC5	6
P142/ SCK20/ SCL20	P40/ TOOL0	$\overline{RESET}$	P121/X1	V_{SS}/EV_{SS}	V_{SS}/EV_{SS}	V_{DD}/EV_{DD}	V_{DD}/EV_{DD}	V_{DD_IO}	P65/TI11/ TO11	P67/TI13/ TO13	IC6	5
P144/ SO20/ TXD2	P143/SI20/ RXD2/ SDA20	Note				Note		Note		IV_{DD}	IV_{DD}	4
P26/ANI6	P27/ANI7									P05/TI05/ TO05	V_{REGO}	3
P152/ ANI10	P153/ ANI11	P11/RXD0	RXD	IC2	IC8	P51/INTP2	WAKE	P50/INTP1	ILIM	IC1	GND3	2
AV_{SS}	AV_{REF}	P13/TXD3	P14/RXD3	P16/TI01/ TO01/ INTP5	P17/TI02/ TO02	P151/ANI9	P150/ANI8	GND1	CQ	V_{DDH}	GND3	1
M	L	K	J	H	G	F	E	D	C	B	A	

Note Connect P11/RxD0, P50/INTP1, and P51/INTP2 to RXD, ILIM, and WAKE respectively on PCB by users.

Cautions 1. Make GND1, GND2, and AV_{SS} the same potential as V_{SS}/EV_{SS}.

2. Make package exposed die pad (GND3) the same potential as V_{SS}/EV_{SS}.

3. Make V_{DD_IO} the same potential as V_{DD}/EV_{DD}.

4. For details about how to handle IC1 to IC8, see Table 3-3 Connection of Unused Pins.

5. Connect the REGC pin to Vss/EVss via a capacitor (0.47 to 1 μF).

6. Connect the IV_{DD} pin to Vss/EVss via a capacitor (330 to 2000 nF).

7. P26/ANI6, P27/ANI7, and P150/ANI8 to P153/ANI11 are set as analog inputs in the order of P153/ANI11, …, P150/ANI8, P27/ANI7, P26/ANI6 by the A/D port configuration register (ADPC). When using P26/ANI6, P27/ANI7, and P150/ANI8 to P153/ANI11 as analog inputs, start designing from P153/ANI11 (see 10.3 (6) A/D port configuration register (ADPC) for details).

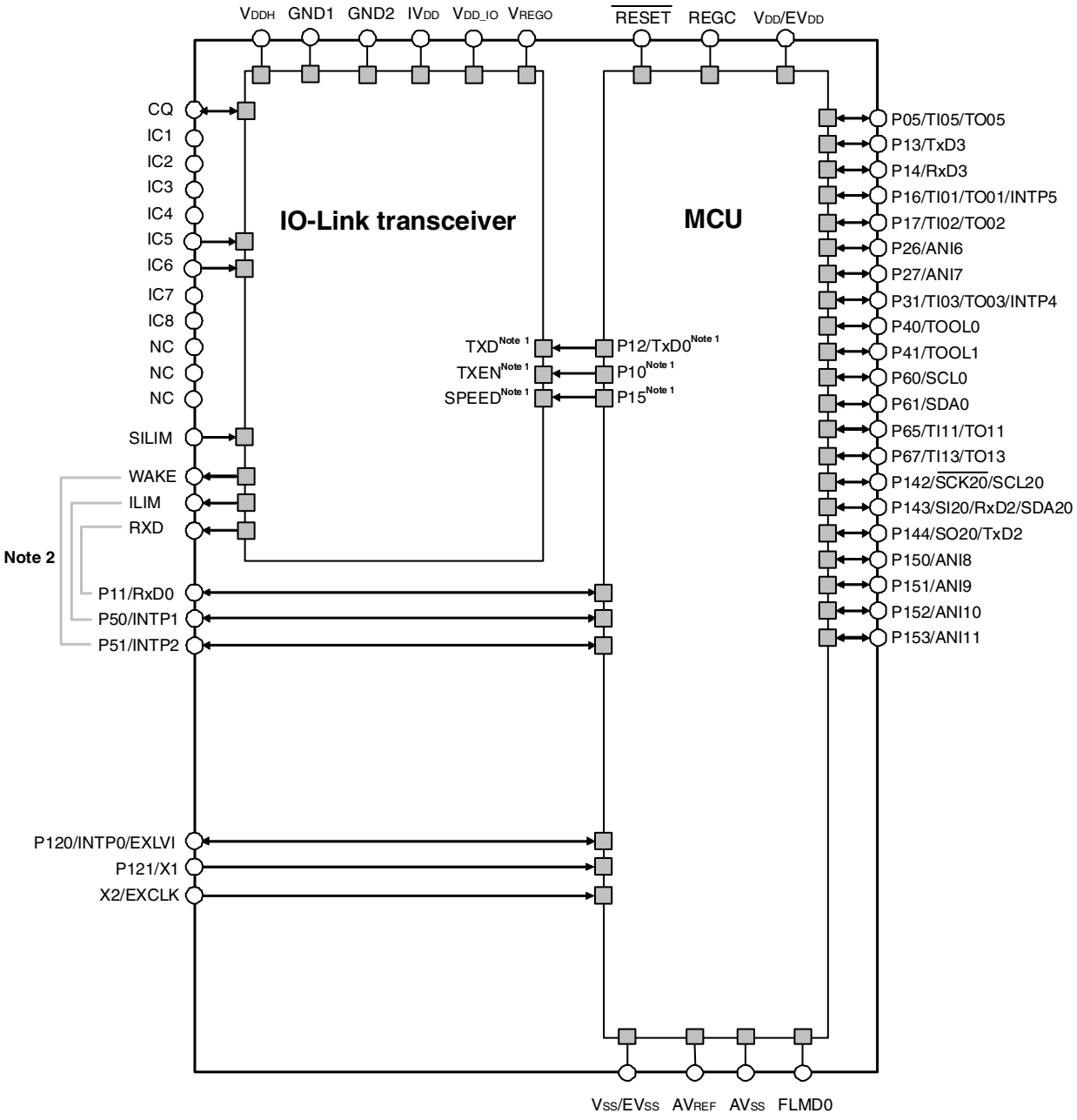
8. Connect CQ pin and V_{DDH} pin via a diode (required specification: breakthrough voltage: 70 V or more, forward current: 100 mA or more, reverse recovery time: 10 ns or less)

Pin Identification

ANI6 to ANI11:	Analog input	P150 to P153:	Port 15
AV_{REF}:	Analog reference voltage	REGC:	Regulator capacitance
AV_{SS}:	Analog ground	RESET:	Reset
CQ:	C/Q, connection for communication (C) or switching (Q) signal of IO-Link	RXD:	Receive signal of IO-Link transceiver
		RxD0, RxD2, RxD3:	Receive data
		SCK20:	Serial clock input/output
EXCLK:	External clock input (main system clock)	SCL0, SCL20:	Serial clock input/output
		SDA0, SDA20:	Serial data input/output
EXLVI:	External potential input for low-voltage detector	SI20:	Serial data input
		SILIM:	Overcurrent shutoff threshold level selection
FLMD0:	Flash programming mode		
GND1:	L-, ground connection of IO-Link	SO20:	Serial data output
GND2:	Ground of IO-Link transceiver	TI01 to TI03, TI05, TI11, TI13:	Timer input
GND3:	Package exposed die pad		
IC1 to IC8:	Internal connect	TO01 to TO03, TO05, TO11, TO13:	Timer output
ILIM:	Overcurrent signal		
INTP0 to INTP2, INTP4, INTP5:	External interrupt input	TOOL0:	Data input/output for tool
		TOOL1:	Clock output for tool
IVDD:	Power supply for IO-Link transceiver	TxD2, TxD3:	Transmit data
		V_{DD}/EV_{DD}:	Power supply of MCU
NC:	Non connect	V_{DDH}:	L+, power supply connection of IO-Link
P05:	Port 0		
P11, P13, P14, P16, P17:	Port 1	V_{DD_IO}:	Power supply for the interface between MCU and IO-Link transceiver
P26, P27:	Port 2		
P31:	Port 3	V_{REGO}:	Regulator control for IO-Link transceiver
P40, P41:	Port 4		
P50, P51:	Port 5	V_{SS}/EV_{SS}:	Ground of MCU
P60, P61, P65, P67:	Port 6	WAKE:	Wakeup request
P120, P121:	Port 12	X1, X2:	Crystal oscillator (main system clock)
P142 to P144:	Port 14		

1.5 Configuration

μPD78F8040, 78F8041, 78F8042, 78F8043 are the products configured by the MCU and IO-Link transceiver.



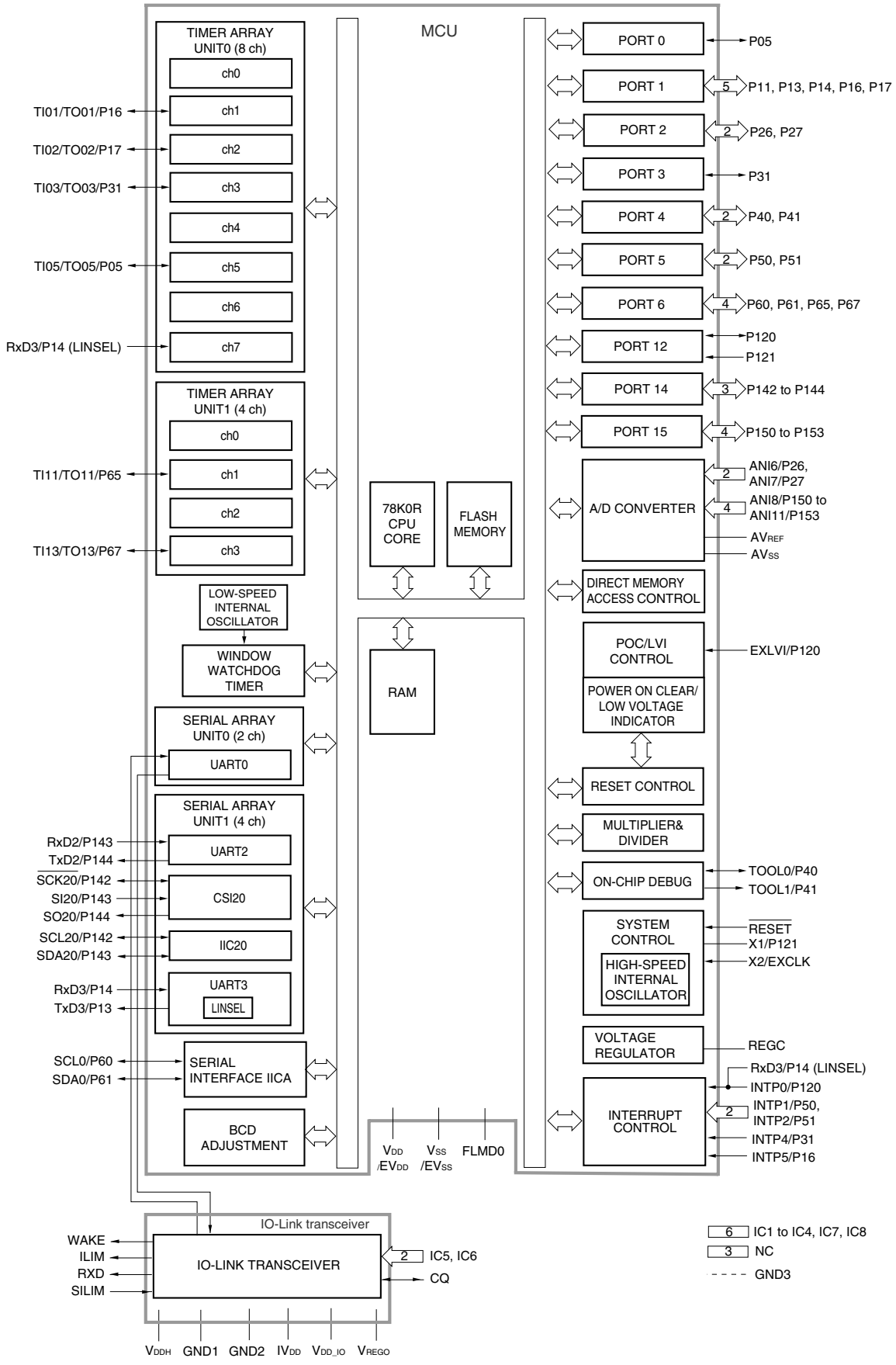
Notes 1. Internal connection pins between MCU and IO-Link transceiver.

2. Connect P11/RxD0, P50/INTP1, and P51/INTP2 to RXD, ILIM, and WAKE respectively on PCB by users.

3. For details about how to handle IC1 to IC8, see **Table 3-3 Connection of Unused Pins**.

Caution Make package exposed die pad (GND3) the same potential as VSS/EVSS.

1.6 Block Diagram



1.7 Outline of Functions

(1/2)

Item		μPD78F8040	μPD78F8041	μPD78F8042	μPD78F8043
Internal memory	Flash memory (self-programming supported)	32 KB	64 KB	96 KB	128 KB
	RAM	4 KB	4 KB	6 KB	7 KB
Memory space		1 MB			
Main system clock (Oscillation frequency)	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 2 to 20 MHz: V_{DD} = 3.0 to 5.5 V			
	Internal high-speed oscillation clock	Internal oscillation 1 MHz (Typ.), 8 MHz (Typ.): V_{DD} = 3.0 to 5.5 V			
	20 MHz internal high-speed oscillation clock	Internal oscillation 20 MHz (Typ.): V_{DD} = 3.0 to 5.5 V			
Internal low-speed oscillation clock (dedicated to WDT)		Internal oscillation 30 kHz (Typ.): V_{DD} = 3.0 to 5.5 V			
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)			
Minimum instruction execution time		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)			
		0.125 μs (Internal high-speed oscillation clock clock: f_{IH} = 8 MHz operation)			
Instruction set		• 8-bit operation, 16-bit operation • Multiplication (8 bits × 8 bits) • Bit manipulation (Set, reset, test, and Boolean operation), etc.			
I/O port (MCU)		Total: 26 Note 1 CMOS I/O: 23 Note 1 CMOS input: 1 N-ch open-drain I/O (6 V tolerance): 2			
I/O port (IO-Link transceiver)		IO-Link I/O: 1 (CQ)			
Timer		• 16-bit timer: 12 channels (Timer input: 6 channels, Timer output: 6 channels) • Watchdog timer: 1 channel			
	Timer output	6 (PWM outputs: timer array unit 0: 4 Note 2, timer array unit 1: 2 Note 2)			
A/D converter		10-bit resolution × 6 channels (AV_{REF} = 1.8 to 5.5 V)			

Notes 1. Three of these pins (P11/RxD0, P50/INTP1, and P51/INTP2) are used for IO-Link communication. They must be connected to the IO-Link transceiver. Connect P11/RxD0, P50/INTP1, and P51/INTP2 to RXD, ILIM, and WAKE respectively on PCB by users.

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

(2/2)

Item		μPD78F8040	μPD78F8041	μPD78F8042	μPD78F8043
Serial interface		• IO-Link (use UART0): 1 channel • CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel • UART supporting LIN-bus: 1 channel • I^2C bus: 1 channel			
Multiplier/divider		• 16 bits × 16 bits = 32 bits (multiplication) • 32 bits ÷ 32 bits = 32 bits (division)			
DMA controller		2 channels			
Vectored interrupt sources	Internal	28			
	External	5 Note 1			
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector • Internal reset by illegal instruction execution Note 2 • Internal reset by a reset processing check error			
Power-on-clear circuit		• Power-on-reset: 1.61 ±0.09 V • Power-down-reset: 1.59 ±0.09 V			
Low-voltage detector		3.15 V to 4.22 V (8 stages)			
On-chip debug function		Provided			
Power supply voltage		V_{DD} = 3.0 to 5.5 V			
Operating ambient temperature		T_A = –40 to +85 °C			
<R> Package		• 56-pin plastic QFN (8 × 8) (0.5 mm pitch) • 56-pin plastic FBGA (4 × 7) Note 3			

Notes 1. Connect P11/RxD0, P50/INTP1, and P51/INTP2 to RXD, ILIM, and WAKE respectively on PCB by users.

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. Under development

CHAPTER 2 CONNECTION BETWEEN MCU AND IO-LINK TRANSCEIVER

2.1 Pins for Connecting MCU to IO-Link Transceiver

The MCU pins, which are shown below, are used to connect the MCU to the IO-Link transceiver so the two can communicate.

These pins must be initially set to appropriate modes or levels before communicating with the IO-Link transceiver. Tables 2-1 and 2-2 show the description of the pins. (For details, see **11.7.3** to **11.7.5**.)

Table 2-1. Internal Connect Pins

Pin Name		Function (IO-Link transceiver)	Direction
IO-Link Transceiver	MCU		
TXEN	P10	Transmitter enable Low level: Turn the transmitter off (default) High level: Enable the transmitter	MCU → IO-Link transceiver
SPEED	P15	Baud rate selection Low level: 38.4k Baud (default) High level: 230.4k Baud	MCU → IO-Link transceiver
TXD	P12/TxD0	Transmit signal TxD determines the output level of the active transmitter. Low level: Set the output level on CQ pin to high High level: Set the output level on CQ pin to low (default)	MCU → IO-Link transceiver

Table 2-2. Connection Externally on the PCB by users

Pin Name		Initial Setting	Direction
IO-Link Transceiver	MCU		
RXD	P11/RxD0	Receive signal RxD provides the receiver output information from reading the bus level on CQ pin Low level: If CQ level is high High level: If CQ level is low	IO-Link transceiver → MCU
WAKE	P51/INTP2	Wakeup request A high level indicates the detection of a wakeup event	IO-Link transceiver → MCU
ILIM	P50/INTP1	Overcurrent signal A high level signals an over current condition for the transmitter on CQ pin	IO-Link transceiver → MCU

2.2 Initial Setting of Unused Internal Pins in MCU

The following internal MCU pins must be set to output mode by using software after a reset release (by clearing the port mode register to 0).

- P03, P04, P06, P30, P42 to P44, P46, P47, P53 to P55, P64, P66, P70 to P77, P110, P140

For setting of the port mode register, see **Figure 6-19. Format of Port Mode Register**.

2.3 Power Supply Configurations of MCU and IO-Link Transceiver

The following two power supply configurations are available for the μPD78F8040, 78F8041, 78F8042, and 78F8043.

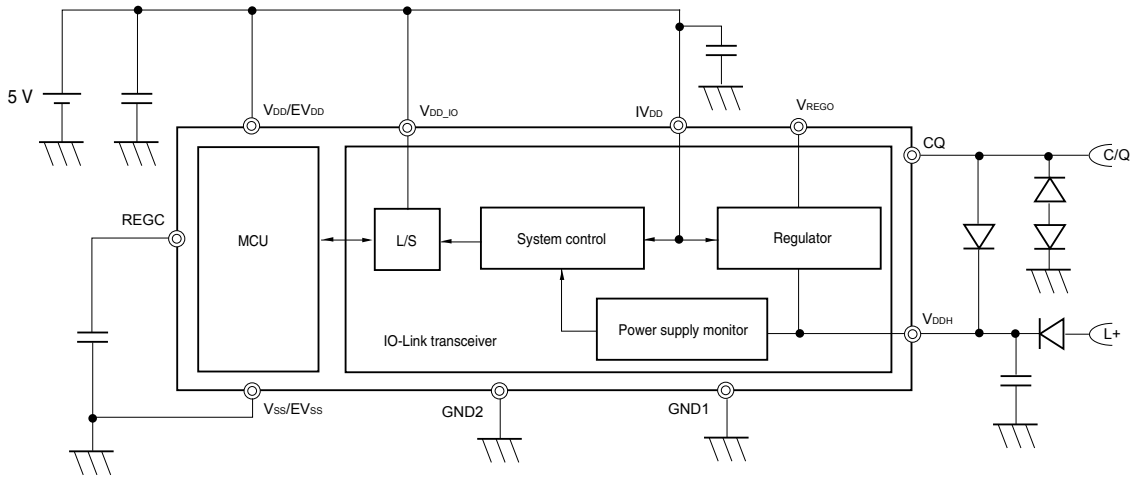
2.3.1 Externally supplying power

The internal 5 V regulator of the IO-Link transceiver is not used because the V_{REGO} pin, which is a regulator output, is left open.

A 5 V power supply is required because the IO-Link transceiver operates at 5 V.

Figure 2-1 shows the power supply configuration when using the MCU with a 5 V interface.

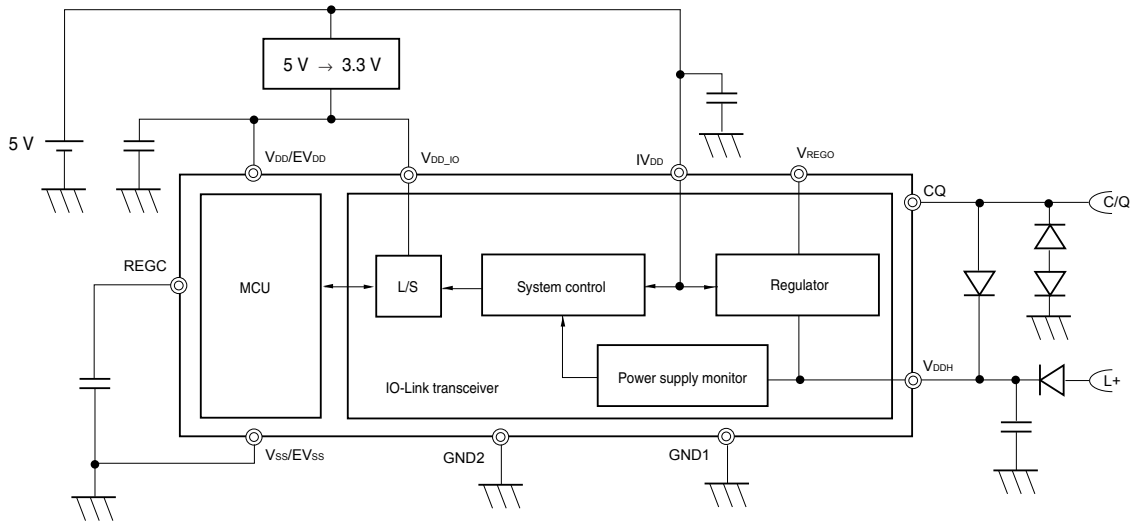
Figure 2-1. Power Supply Configuration When Externally Supplying 5 V to IV_{DD} Pin (5 V Interface)



To use the MCU with a 3.3 V interface, the MCU must be used at 3.3 V such as by connecting an external 3.3 V regulator to the MCU.

Figure 2-2 shows the power supply configuration when using the MCU with a 3.3 V interface.

Figure 2-2. Power Supply Configuration When Externally Supplying 5 V to IV_{DD} Pin (3.3 V Interface)



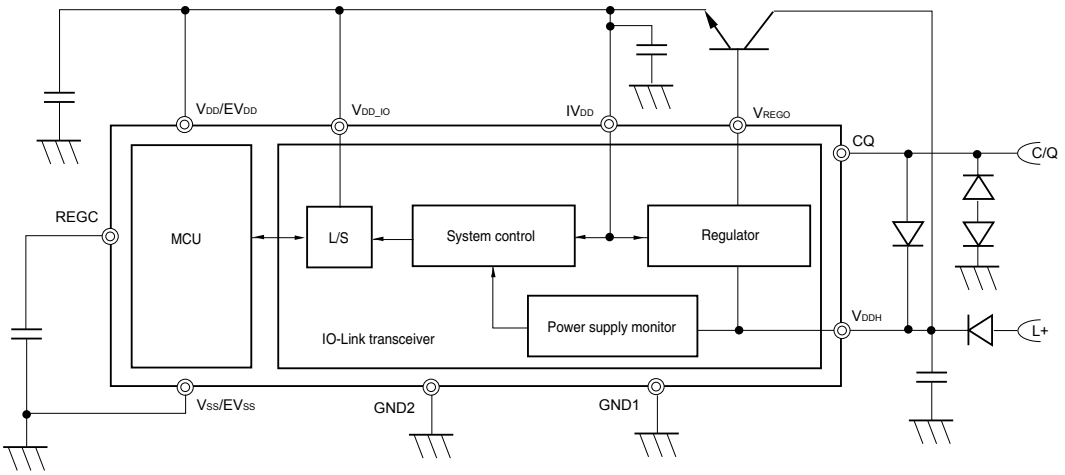
Caution Make QFN package exposed die pad (GND3) the same potential as V_{SS}/EV_{SS}.

2.3.2 Using internal regulator of IO-Link transceiver

When a current larger than 20 mA, including the I/O current, is supplied to the MCU, an external NPN transistor is required because the internal 5 V regulator can only supply a current of up to 20 mA. If required, connect an NPN transistor as shown in Figure 2-3.

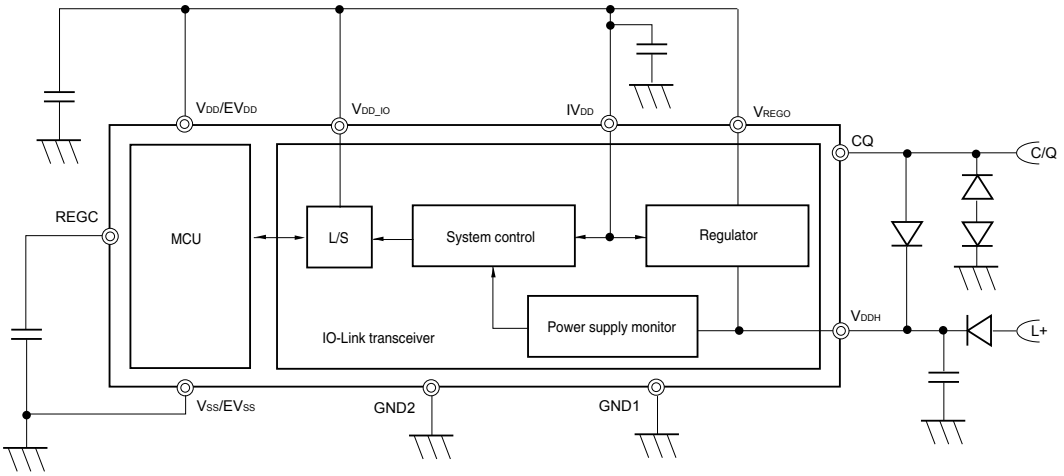
The V_{REGO} pin voltage is controlled by connecting an NPN transistor. As a result, the V_{REGO} pin voltage becomes 0.7 V higher than the IV_{DD} pin voltage.

Figure 2-3. Power Supply Configuration When Supplying Current Larger than 20 mA to MCU and External Components by Using Internal 5 V Regulator of IO-Link Transceiver



A maximum current of 20 mA can be supplied to the MCU by using the internal regulator of the IO-Link transceiver. The current can be supplied from the V_{REGO} pin to the MCU by connecting the V_{REGO} and IV_{DD} pins as shown in Figure 2-4.

Figure 2-4. Power Supply Configuration When Supplying Current of 20 mA or Less to MCU by Using Internal 5 V Regulator of IO-Link Transceiver



Cautions 1. Make QFN package exposed die pad (GND3) the same potential as V_{SS}/EV_{SS}.

<R> **2. Be sure to connect an NPN transistor regardless of a supply current when the internal regulator is used in the FBGA package poducts (refer to figure 2-3). The power supply configuration of figure 2-4 is prohibited in the FBGA package poducts.**

CHAPTER 3 PIN FUNCTIONS

3.1 Pin Function List

There are two types of pin I/O buffer power supplies: AV_{REF} and V_{DD}/EV_{DD}. The relationship between these power supplies and the pins is shown below.

Table 3-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
V_{DD}/EV_{DD}	• Port pins other than P26, P27, and P150 to P153 • Pins other than port pins
AV_{REF}	P26, P27, P150 to P153

(1) Port functions (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P05	I/O	Port 0. 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	TI05/TO05
P11 Note 1	I/O	Port 1. 5-bit I/O port. Input of P11 can be set to TTL input buffer. 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	RxD0
P13				TxD3
P14				RxD3
P16				TI01/TO01/INTP5
P17				TI02/TO02
P26	I/O	Port 2. 2-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI6
P27				ANI7
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
P40 Note 2	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				TOOL1
P50 Note 1	I/O	Port 5. 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	INTP1
P51 Note 1				INTP2

Notes 1. Connect P11/RxD0, P50/INTP1, and P51/INTP2 to RXD, ILIM, and WAKE respectively on PCB by users.

2. If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally (see **Caution** in **3.2.5 P40, P41 (port 4)**).

(1) Port functions (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P60	I/O	Port 6. 4-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. For only P65 and P67, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCL0
P61				SDA0
P65				TI11/TO11
P67				TI13/TO13
P120	I/O	Port 12. 1-bit I/O port and 1-bit input port. 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.	Input port	INTP0/EXLVI
P121	Input			X1
P142	I/O	Port 14. 3-bit I/O port. Input of P142 and P143 can be set to TTL input buffer. Output of P142 to P144 can be set to the 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	$\overline{SCK20}$/SCL20
P143				SI20/RxD2/SDA20
P144				SO20/TxD2
P150 to P153	I/O	Port 15. 4-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8 to ANI11

(2) Non-port functions (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
ANI6, ANI7	Input	A/D converter analog input	Digital input port	P26, P27
ANI8 to ANI11	Input	A/D converter analog input	Digital input port	P150 to P153
CQ	High voltage I/O	C/Q, connection pin for communication (C) or switching (Q) signal of IO-Link	–	–
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
IC1	–	Internal connection pin	–	–
IC2	–		–	–
IC3	–		–	–
IC4	–		–	–
IC5	Input		Input port	–
IC6	Input		Input port	–
IC7	–		–	–
IC8	–		–	–
ILIM	Output	Overcurrent signal	–	–
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P120/EXLVI
INTP1[Note]				P50
INTP2[Note]				P51
INTP4				P31/TI03/TO03
INTP5				P16/TI01/TO01
REGC	–	Connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect to V_{SS}/EV_{SS} via a capacitor (0.47 to 1 μF).	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RXD	Output	Receive Signal of IO-Link transceiver	–	–
RxD0[Note]	Input	Serial data input to UART0	Input port	P11
RxD2	Input	Serial data input to UART2	Input port	P143/SI20/SDA20
RxD3	Input	Serial data input to UART3	Input port	P14
$\overline{\text{SCK20}}$	I/O	Clock input/output for CSI20	Input port	P142/SCL20
SCL0	I/O	Clock input/output for I^2C	Input port	P60
SCL20	I/O	Clock input/output for simplified I^2C	Input port	P142/$\overline{\text{SCK20}}$
SDA0	I/O	Serial data I/O for I^2C	Input port	P61
SDA20	I/O	Serial data I/O for simplified I^2C	Input port	P143/SI20/RxD2
SI20	Input	Serial data input to CSI20	Input port	P143/RxD2/SDA20
SILIM	Input	Overcurrent shutoff threshold level selection signal	–	–
SO20	Output	Serial data output from CSI20	Input port	P144/TxD2
TI01	Input	External count clock input to 16-bit timer 01	Input port	P16/TO01/INTP5
TI02		External count clock input to 16-bit timer 02		P17/TO02
TI03		External count clock input to 16-bit timer 03		P31/TO03/INTP4
TI05		External count clock input to 16-bit timer 05		P05/TO05

Note Connect P11/RxD0, P50/INTP1, and P51/INTP2 to RXD, ILIM, and WAKE respectively on PCB by users.

(2) Non-port functions (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
TI11	Input	External count clock input to 16-bit timer 11	Input port	P65/TO11
TI13		External count clock input to 16-bit timer 13		P67/TO13
TO01	Output	16-bit timer 01 output	Input port	P16/TI01/INTP5
TO02		16-bit timer 02 output		P17/TI02
TO03		16-bit timer 03 output		P31/TI03/INTP4
TO05		16-bit timer 05 output		P05/TI05
TO11		16-bit timer 11 output		P65/TI11
TO13		16-bit timer 13 output		P67/TI13
TxD2	Output	Serial data output from UART2	Input port	P144/SO20
TxD3		Serial data output from UART3		P13
NC	–	Non connection pin	–	–
WAKE	Output	Wakeup request	–	–
EXCLK	Input	External clock input for main system clock	Input port	X2
X1	–	Resonator connection for main system clock	Input port	P121
X2	–		Input port	EXCLK
AV_{REF}	–	• A/D converter reference voltage input • Positive power supply for P26, P27, P150 to P153, and A/D converter	–	–
AV_{SS}	–	Ground potential for A/D converter, P26, P27, and P150 to P153. Use this pin with the same potential as V_{SS}/EV_{SS}.	–	–
GND1	–	L-, ground connection of IO-Link transceiver.	–	–
GND2	–	Ground connection of IO-Link transceiver.	–	–
GND3	–	Package exposed die pad.	–	–
IV_{DD}	–	Power supply for IO-Link transceiver. Connect to V_{SS}/EV_{SS} via a capacitor (330 to 2000 nF).	–	–
V_{DD}/EV_{DD}	–	Positive power supply (P26, P27, P150 to P153 and other than ports).	–	–
V_{DDH}	–	L+, power supply connection of IO-Link transceiver	–	–
V_{DD_IO}	–	Power supply for the interface between MCU and IO-Link transceiver	–	–
V_{REGO}	–	Regulator control for IO-Link transceiver	–	–
V_{SS}/EV_{SS}	–	Ground potential (port part (P26, P27, P150 to P153) and other than ports)	–	–
FLMD0	–	Flash memory programming mode setting	–	–
TOOL0[Note]	I/O	Data I/O for flash memory programmer/debugger	Input port	P40
TOOL1	Output	Clock output for debugger	Input port	P41

Note If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally (see **Caution** in **3.2.5 P40, P41 (port 4)**).

3.2 Description of Pin Functions

3.2.1 P05 (port 0)

P05 functions as a 1-bit I/O port. These pins also function as timer I/O.

The following operation modes can be specified.

(1) Port mode

P05 functions as a 1-bit I/O port. P05 can be set to input or output port 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

P05 functions as timer I/O.

(a) TI05

This is a pin for inputting an external count clock/capture trigger to 16-bit timer 05.

(b) TO05

This is a timer output pins of 16-bit timer 05.

3.2.2 P11, P13, P14, P16, P17 (port 1)

P11, P13, P14, P16, and P17 function as a 5-bit I/O port. These pins also function as external interrupt request input, serial interface data I/O, and timer I/O.

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

(1) Port mode

P11, P13, P14, P16, and P17 function as a 5-bit I/O port. P11, P13, P14, P16, and 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

P11, P13, P14, P16, and P17 function as external interrupt request input, serial interface data I/O, and timer I/O.

(a) RxD3

This is a serial data input pin of serial interface UART3.

(b) TxD3

This is a serial data output pin of serial interface UART3.

(c) TI01, TI02

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

(d) TO01, TO02

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

(e) 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.

3.2.3 P26, P27 (port 2)

P26 and P27 function as a 2-bit I/O port. These pins also function as A/D converter analog input.

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

(1) Port mode

P26 and P27 function as a 2-bit I/O port. P26 and P27 can be set to input or output port in 1-bit units using port mode register 2 (PM2).

(2) Control mode

P26 and P27 function as A/D converter analog input pins (ANI6 and ANI7). When using these pins as analog input pins, see **10.6 (5) ANI6/P26, ANI7/P27, and ANI8/P150 to ANI11/P153**.

Caution ANI6/P26 and ANI7/P27 are set in the digital input (general-purpose port) mode after release of reset.

3.2.4 P31 (port 3)

P31 functions as a 1-bit I/O port. These pins also function as external interrupt request input and timer I/O.

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

(1) Port mode

P31 functions as a 1-bit I/O port. P31 can be set to input or output port 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

P31 functions as external interrupt request input and timer I/O.

(a) INTP4

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) TI03

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

(c) TO03

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

3.2.5 P40, P41 (port 4)

P40 and P41 function as a 2-bit I/O port. These pins also function as data I/O for a flash memory programmer/debugger and clock output.

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

(1) Port mode

P40 and P41 function as a 2-bit I/O port. P40 and P41 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 and P41 function as data I/O for a flash memory programmer/debugger and clock output.

(a) 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).

(b) TOOL1

This is a clock output pin for a debugger.

When the on-chip debug function is used, P41/TOOL1 pin can be used as follows by the mode setting on the debugger.

1-line mode: can be used as a port (P41).

2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).

Caution The function of the P40/TOOL0 pin varies as described in (a) to (c) below.
In the case of (b) or (c), make the specified connection.

(a) In normal operation mode and when on-chip debugging is disabled (OCDENSET = 0) by an option byte (000C3H)
=> Use this pin as a port pin (P40).

(b) In normal operation mode and when on-chip debugging is enabled (OCDENSET = 1) by an option byte (000C3H)
=> Connect this pin to EV_{DD} via an external resistor, and always input a high level to the pin before reset release.

(c) When on-chip debug function is used, or in write mode of flash memory programmer
=> Use this pin as TOOL0.
Directly connect this pin to the on-chip debug emulator or a flash memory programmer, or pull it up by connecting it to EV_{DD} via an external resistor.

3.2.6 P50, P51 (port 5)

P50 and P51 function as a 2-bit I/O port. These pins also function as external interrupt request input.

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

(1) Port mode

P50 and P51 function as a 2-bit I/O port. P50 and P51 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 and P51 function as external interrupt request input.

(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.

3.2.7 P60, P61, P65, P67 (port 6)

P60, P61, P65, P67 function as a 4-bit I/O port. These pins also function as serial interface data I/O, clock I/O, and timer I/O.

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

(1) Port mode

P60, P61, P65, P67 function as a 4-bit I/O port. P60 to P67 can be set to input port or output port in 1-bit units using port mode register 6 (PM6). Only for P65, P67, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 6 (PU6).

Output of P60, P61 is N-ch open-drain output (6 V tolerance).

(2) Control mode

P60, P61, P65, and P67 function as serial interface data I/O, clock I/O, and timer I/O.

(a) SDA0

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

(b) SCL0

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

(c) TI11, TI13

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 11 and 13.

(d) TO11, TO13

These are the timer output pins of 16-bit timers 11 and 13.

3.2.8 P120, P121 (port 12)

P120 functions as a 1-bit I/O port. P121 functions as a 1-bit input port. These pins also function as external interrupt request input, potential input for external low-voltage detection, and connecting resonator for main system clock.

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

(1) Port mode

P120 functions 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 functions as a 1-bit input port.

(2) Control mode

P120, P121 function as external interrupt request input, potential input for external low-voltage detection, and connecting resonator for main system clock.

(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.

(b) EXLVI

This is a potential input pin for external low-voltage detection.

(c) X1

This is a pin for connecting a resonator for main system clock.

3.2.9 P142 to P144 (port 14)

P142 to P144 function as a 3-bit I/O port. These pins also function as serial interface data I/O and clock I/O.

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

Output from the P142 to P144 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 14 (POM14).

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

(1) Port mode

P142 to P144 function as a 3-bit I/O port. P142 to P144 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

P142 to P144 function as serial interface data I/O and clock I/O.

(a) SI20

This is a serial data input pin of serial interface CSI20.

(b) SO20

This is a serial data output pin of serial interface CSI20.

(c) $\overline{\text{SCK20}}$

This is a serial clock I/O pin of serial interface CSI20.

(d) TxD2

This is a serial data output pin of serial interface UART2.

(e) RxD2

This is a serial data input pin of serial interface UART2.

(f) SDA20

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

(g) SCL20

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

3.2.10 P150 to P153 (port 15)

P150 to P153 function as a 4-bit I/O port. These pins also function as A/D converter analog input.

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

(1) Port mode

P150 to P153 function as a 4-bit I/O port. P150 to P153 can be set to input or output port in 1-bit units using port mode register 15 (PM15).

(2) Control mode

P150 to P153 function as A/D converter analog input pins (ANI8 to ANI11). When using these pins as analog input pins, see **10.6 (5) ANI6/P26, ANI7/P27, and ANI8/P150 to ANI11/P153**.

Caution ANI8/P150 to ANI11/P153 are set in the digital input (general-purpose port) mode after release of reset.

3.2.11 CQ

This is the I/O pin of the C/Q (communication (C)/switching (Q)) line.

3.2.12 IC1 to IC8

These are internal connect pins.

3.2.13 ILIM

This pin outputs overcurrent detection signals from the IO-Link transceiver.

Connect this pin to P50/INTP1 on PCB by users.

3.2.14 NC

These are non connect connect pins.

3.2.15 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

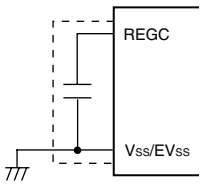
When the external reset pin is not used, connect this pin directly to EV_{DD} or via a resistor.

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

3.2.16 REGC

This is the pin for connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect this pin to V_{SS}/EV_{SS} via a capacitor (0.47 to 1 μF).

Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

3.2.17 RXD

This pin outputs the signals received from the IO-Link transceiver.

Connect this pin to P11/RxD0 on PCB by users.

3.2.18 SILIM

The signal for selecting the overcurrent shutoff level is input to this pin.

Table 3-2. Overcurrent Shutoff Threshold Level Selection

SILIM	Overcurrent shutoff threshold level
0	220 to 480 [mA]
1	110 to 240 [mA]

3.2.19 WAKE

This pin outputs wakeup request signals from the IO-Link transceiver.

Connect this pin to P51/INTP2 on PCB by users.

3.2.20 EXCLK

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

3.2.21 X1, X2

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

3.2.22 AV_{REF}

This is the A/D converter reference voltage input pin and the positive power supply pin of P26, P27, P150 to P153, and A/D converter.

When all pins of ports 2 and 15 are used as the analog port pins, make the potential of AV_{REF} be such that 1.8 V ≤ AV_{REF} ≤ V_{DD}. When one or more of the pins of ports 2 and 15 are used as the digital port pins or when the A/D converter is not used, make AV_{REF} the same potential as V_{DD}/EV_{DD}.

3.2.23 AV_{SS}

This is the ground potential pin of A/D converter, P26, P27, and P150 to P153. Even when the A/D converter is not used, always use this pin with the same potential as V_{SS}/EV_{SS}.

3.2.24 GND1

This is a ground connection pin of IO-Link transceiver.

This pin is connected to the L- line of IO-Link.

3.2.25 GND2

This is a ground connection pin of IO-Link transceiver.

3.2.26 GND3

This metal pad is provided on the back of the package.

3.2.27 IV_{DD}

IV_{DD} is the power supply of IO-Link transceiver. It can be applied externally or by the internal 5 V regulator. An external NPN transistor can be connected to pins V_{REGO} (Base) and IV_{DD} (Emitter) in order to supply more current.

If the voltage regulator is used without external transistor, both pins have to be connected.

If IV_{DD} is under IV_{DD_UV} the transmitter is disabled (see **CHAPTER 26 ELECTRICAL SPECIFICATIONS**).

3.2.28 V_{DD}/EV_{DD}

V_{DD}/EV_{DD} are a positive power supply pin for ports other than P26, P27, and P150 to P153 pins and pins other than ports.

3.2.29 V_{DDH}

V_{DDH} is a positive power supply pin for IO-Link transceiver.

This pin is connected to the L+ line of IO-Link.

3.2.30 V_{DD_IO}

The voltage for converting the voltage level at the interface between the MCU and IO-Link transceiver is input to this pin.

If V_{DD_IO} is under $V_{DD\_IO\_UV}$ the transmitter is disabled (see **CHAPTER 26 ELECTRICAL SPECIFICATIONS**).

3.2.31 V_{REGO}

This pin is used to control the regulator of the IO-Link transceiver.

3.2.32 V_{SS}/EV_{SS}

V_{SS}/EV_{SS} is the ground potential pin for ports other than P26, P27, and P150 to P153 and pins other than ports..

3.2.33 FLMD0

This is a pin for setting flash memory programming mode.

Perform either of the following processing.

(a) In normal operation mode

It is recommended to leave this pin open during normal operation.

The FLMD0 pin must always be kept at the V_{SS} level before reset release but does not have to be pulled down externally because it is internally pulled down by reset. However, pulling it down must be kept selected (i.e., FLMDPUP = “0”, default value) by using bit 7 (FLMDPUP) of the background event control register (BECTL) (see **22.5 (1) Back ground event control register**). To pull it down externally, use a resistor of 200 kΩ or smaller.

Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS}/EV_{SS} pin.

(b) In self programming mode

It is recommended to leave this pin open when using the self programming function. To pull it down externally, use a resistor of 100 kΩ to 200 kΩ.

In the self programming mode, the setting is switched to pull up in the self programming library.

(c) In flash memory programming mode

Directly connect this pin to a flash memory programmer when data is written by the flash memory programmer. This supplies a writing voltage of the V_{DD} level to the FLMD0 pin.

The FLMD0 pin does not have to be pulled down externally because it is internally pulled down by reset. To pull it down externally, use a resistor of 1 kΩ to 200 kΩ.

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

Table 3-3 shows the types of pin I/O circuits and the recommended connections of unused pins.

Table 3-3. Connection of Unused Pins (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P05/TI05/TO05	8-R	I/O	Input: Independently connect to V_{DD}/EV_{DD} or V_{SS}/EV_{SS} via a resistor. Output: Leave open.
P13/TxD3	5-AG		
P14/RxD3	8-R		
P16/TI01/TO01/INTP5	8-R		
P17/TI02/TO02			
P26/ANI6, P27/ANI7 Note 1	11-G		Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.
P31/TI03/TO03/INTP4	8-R		Input: Independently connect to V_{DD}/EV_{DD} or V_{SS}/EV_{SS} via a resistor. Output: Leave open.
P40/TOOL0			<When on-chip debugging is enabled> Pull this pin up (pulling it down is prohibited). <When on-chip debugging is disabled> Input: Independently connect to V_{DD}/EV_{DD} or V_{SS}/EV_{SS} via a resistor. Output: Leave open.
P41/TOOL1	5-AG		Input: Independently connect to V_{DD}/EV_{DD} or V_{SS}/EV_{SS} via a resistor. Output: Leave open.
P60/SCL0	13-R		Input: Connect to V_{SS}/EV_{SS}. Output: Set the port output latch to 0 and leave these pins open via low-level output.
P61/SDA0			
P65/TI11/TO11	8-R		Input: Independently connect to V_{DD}/EV_{DD} or V_{SS}/EV_{SS} via a resistor. Output: Leave open.
P67/TI13/TO13			
P120/INTP0/EXLVI	8-R		
P121/X1 Note 2	37-B	Input	Independently connect to V_{DD}/EV_{DD} or V_{SS}/EV_{SS} via a resistor.
P142/$\overline{SCK20}$/SCL20	5-AN	I/O	Input: Independently connect to V_{DD}/EV_{DD} or V_{SS}/EV_{SS} via a resistor. Output: Leave open.
P143/SI20/RxD2/SDA20			
P144/SO20/TxD2	5-AG		
P150/ANI8 to P153/ANI11 Note 1	11-G		Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.

Notes 1. P26/ANI6, P27/ANI7, and P150/ANI8 to P153/ANI11 are set in the digital input port mode after release of reset.

2. Use recommended connection above in input port mode (see **Figure 7-2 Format of Clock Operation Mode Control Register (CMC)**) when these pins are not used.

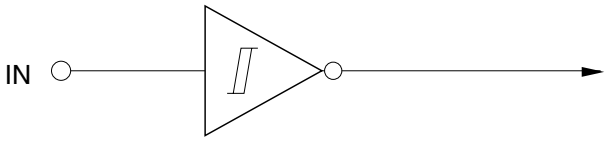
Table 3-3. Connection of Unused Pins (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
CQ	–	High voltage I/O	Independently connect to V_{SS}/EV_{SS} via a resistor.
IC1, IC2, IC3, IC4, IC7, IC8	–	–	Leave open.
IC5, IC6	–	Input	Make these pins the same potential as V_{SS}/EV_{SS}.
NC	–	–	Leave open.
SILIM	–	–	Leave open.
X2/EXCLK	37-C	Input	Independently connect to V_{DD}/EV_{DD} or V_{SS}/EV_{SS} via a resistor.
AV_{REF}	–	–	<When one or more of P26, P27, and P150 to P153 are set as a digital port> Make this pin the same potential as V_{DD}/EV_{DD}. <When all of P26, P27 and P150 to P153 are set as analog ports> Make this pin to have a potential where 1.8 V ≤ AV_{REF} ≤ V_{DD}.
AV_{SS}	–	–	Make this pin the same potential as V_{SS}.
GND1	–	–	Make this pins the same potential as V_{SS}/EV_{SS}.
GND2	–	–	Make this pins the same potential as V_{SS}/EV_{SS}.
GND3	–	–	Make this pins the same potential as V_{SS}/EV_{SS}.
IV_{DD}	–	–	Make this pins the same potential as V_{DD}/EV_{DD}.
V_{DDH}	–	–	Make this pins the same potential as V_{DD}/EV_{DD}.
V_{DD_IO}	–	–	Make this pins the same potential as V_{DD}/EV_{DD}.
V_{REGO}	–	–	Leave open.
FLMD0	2-W	–	Leave open or connect to V_{SS}/EV_{SS} via a resistor of 100 kΩ or more.
$\overline{\text{RESET}}$	2	Input	Connect directly or via a resistor to EV_{DD}.
REGC	–	–	Connect to V_{SS}/EV_{SS} via capacitor (0.47 to 1 μF).

Remark IC3, IC4, IC7, and NC pins are not mounted onto the FBGA package products.

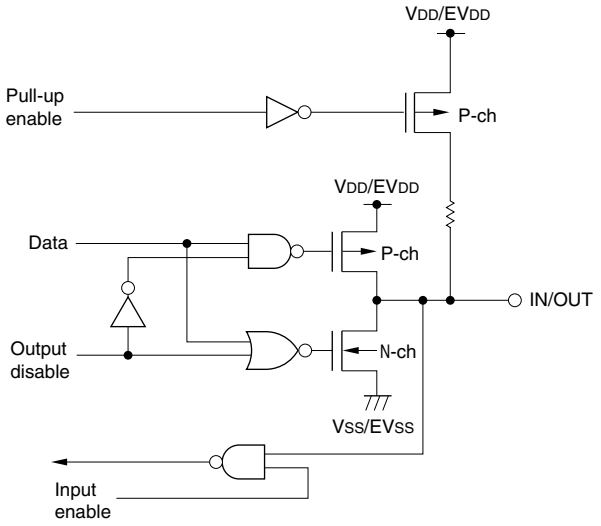
Figure 3-1. Pin I/O Circuit List (1/2)

Type 2

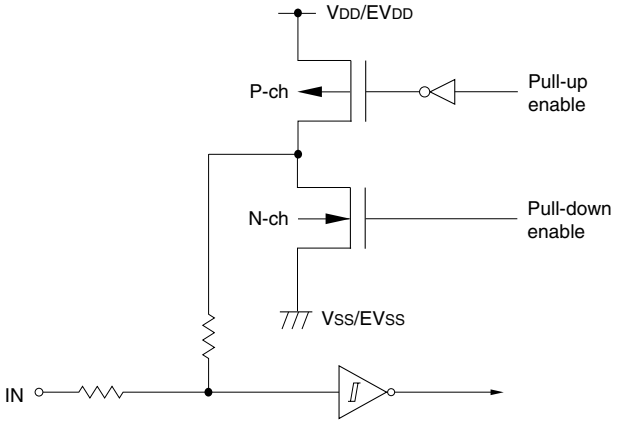


Schmitt-triggered input with hysteresis characteristics

Type 5-AG

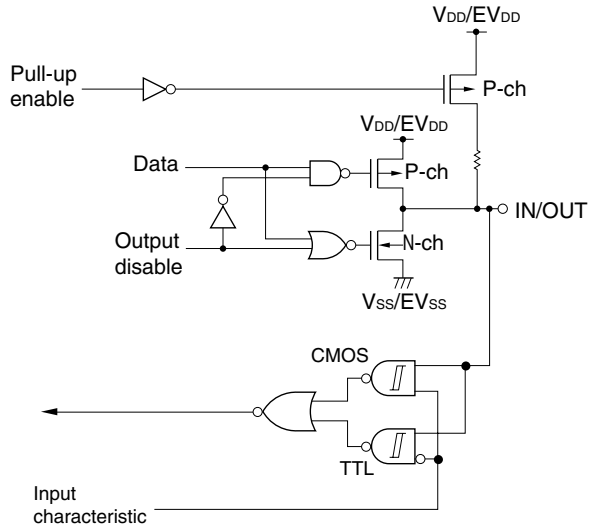


Type 2-W

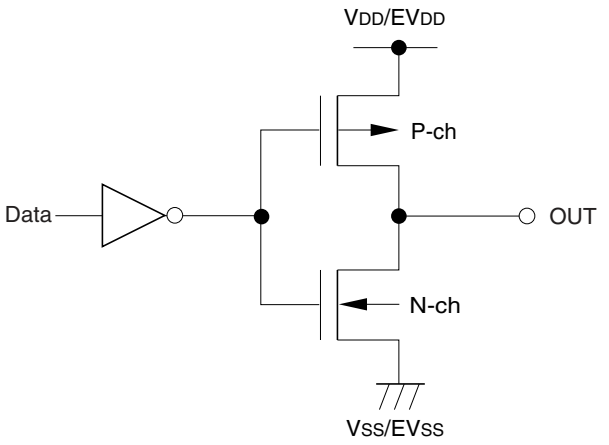


Schmitt-triggered input with hysteresis characteristics

Type 5-AN



Type 3-C



Type 8-R

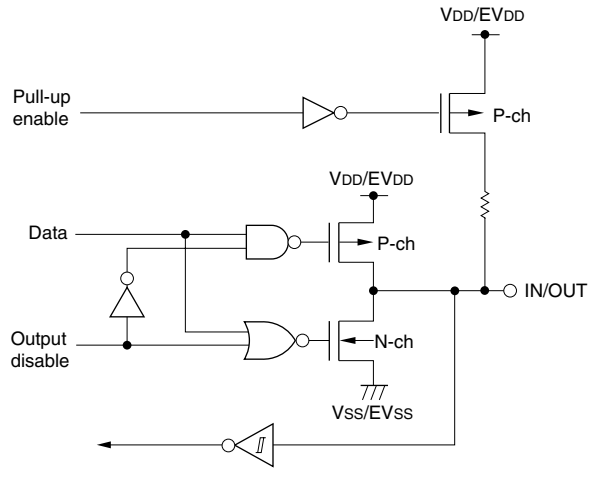
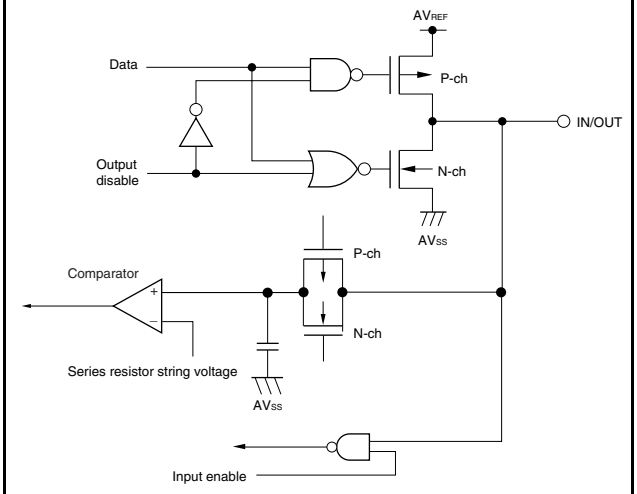
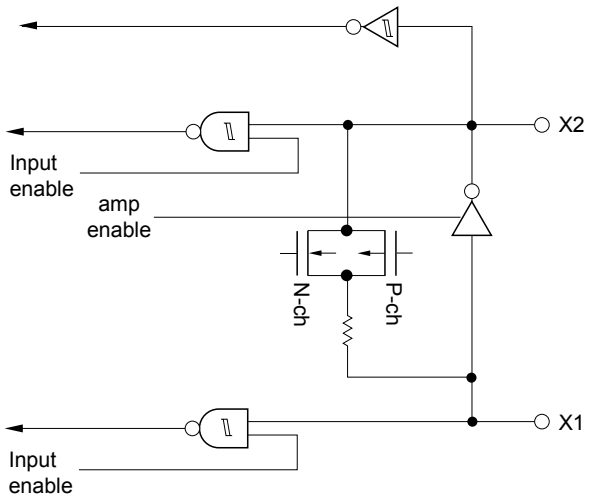


Figure 3-1. Pin I/O Circuit List (2/2)

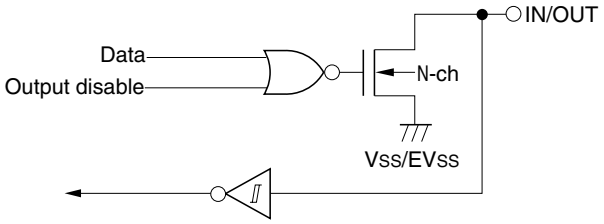
Type 11-G



Type 37-C



Type 13-R



CHAPTER 4 IO-LINK FUNCTIONS

For the μPD78F8040, 78F8041, 78F8042, and 78F8043, communication with the IO-Link transceiver is performed by using UART0 of the MCU (see **11.7.3** to **11.7.5**).

4.1 Outline of IO-Link

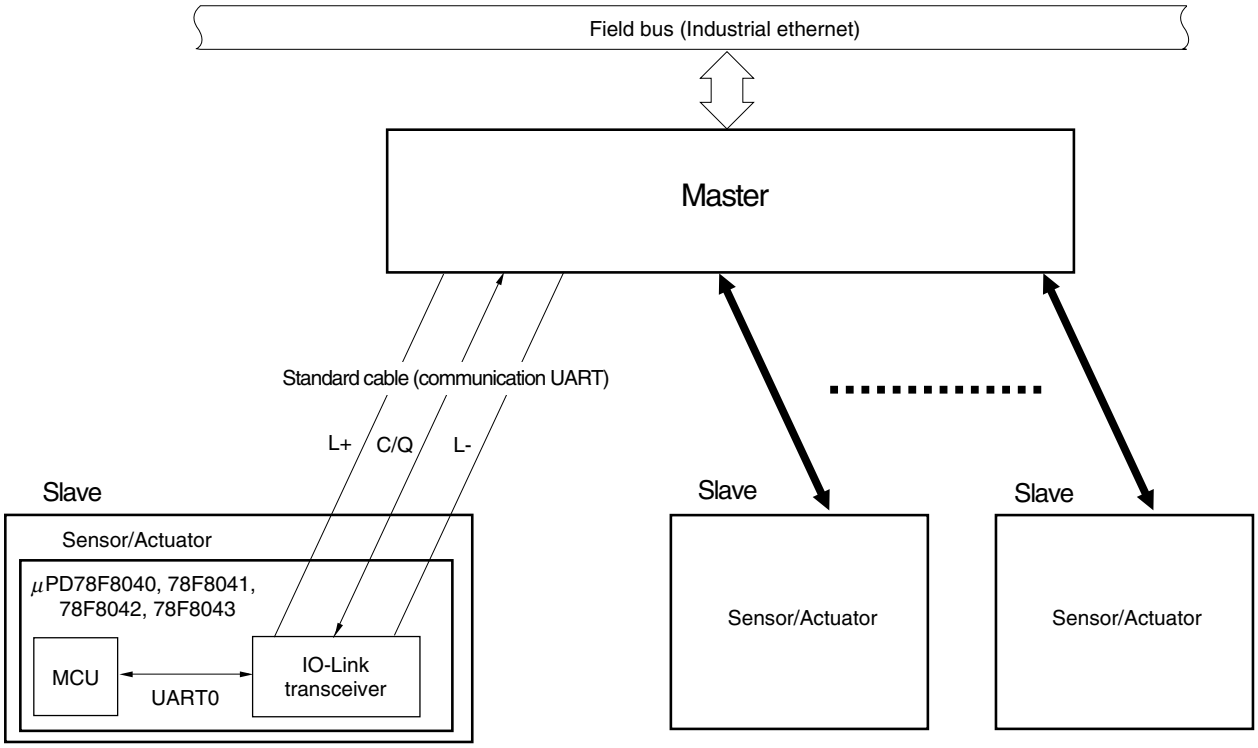
IO-Link has the following features:

- **Standard protocol for sensors and actuators (Point-to-Point Protocol).**
- **Easily maintained due to the extensive diagnostic functions**
- **Easily installed**
- **Existing connectors can be used.**
- **Multiple slaves can be connected to one master.**

4.2 Setup for the IO-Link connections

The following figure shows a setup for the IO-Link connections between the μPD78F8040, 78F8041, 78F8042, or 78F8043 and the master.

Figure 4-1. Figure of Setup for the IO-Link Connections



4.3 Features of IO-Link Transceiver Mounted on This Product

For the μPD78F8040, 78F8041, 78F8042, and 78F8043, the IO-Link function is implemented by IO-Link transceiver inside. The IO-Link transceiver is used as a slave incorporated in a sensor or actuator. The communication and the power supply run via 3 separate lines to IO-Link Master.

This IO-Link transceiver has the following features:

- **Supply voltage range: 8 to 36 V**
- **Integrated 5 V voltage regulator**
- **A baud rate of up to 230400 [bps] can be selected.**
- **Wakeup detection function**
- **3.3 V / 5 V compatible digital interface**
- **Overcurrent detection and shutoff threshold level selection**

4.3.1 Transmitter ON/OFF control

A high level at TXEN enables the transmitter. A low level turns the transmitter off.

TXEN can be set using P10 (an internal connection pin of MCU).

Caution If IV_{DD} is under IV_{DD_UV} the transmitter is disabled.

If V_{DD_IO} is under $V_{DD\_IO\_UV}$ the transmitter is disabled (see CHAPTER 26 ELECTRICAL SPECIFICATIONS).

Table 4-1. Transmitter ON/OFF Control

TXEN	Operation
0	Turns the transmitter off (default)
1	Enable the transmitter

4.3.2 Transfer baud rate selection

The communication baud rate of the transceiver can be selected using P15 (an internal connection pin of the MCU) to SPEED (an internal connection pin of the transceiver).

Table 4-2. Transfer Baud Rate Selection

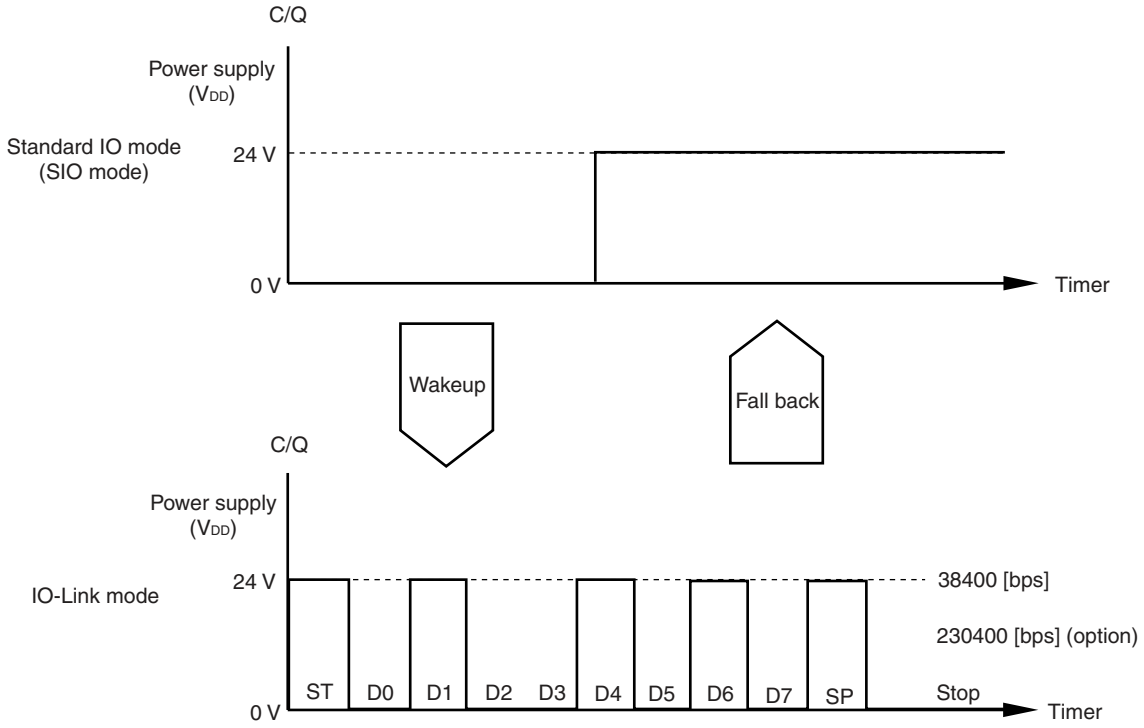
SPEED	Transfer baud rate
0	38400 [bps]
1	230400 [bps]

4.3.3 IO-Link mode and standard I/O (SIO) mode

This transceiver supports IO-Link mode and SIO-mode applications.

- IO-Link mode: P12/TxD0 (internal pin) and P11/RxD0 pins are used as serial interface data I/O
- Standard IO mode: P12/TxD0 (internal pin) and P11/RxD0 pins are used as a general-purpose port

Figure 4-2. Standard I/O mode and IO-Link mode



4.3.4 Wakeup detection function

The receiver monitors CQ pin in order to detect wakeup events. If the information read from the bus changes state, while TXD remains constant, a wakeup event is assumed. After wakeup debounce time, the wakeup event is signaled and the transmitter is turned off.

The transceiver detects wakeup signals by using the signal input from WAKE (an internal connection pin of the transceiver) to P51/INTP2 (an internal connection pin of the MCU).

4.3.5 Overcurrent detection and shutoff threshold level selection

The transceiver detects overcurrent by using the signal input from ILIM (an internal connection pin of the transceiver) to P50/INTP1 (an internal connection pin of the MCU).

Overcurrent shutoff threshold level can be selected by input signal to SILIM pin.

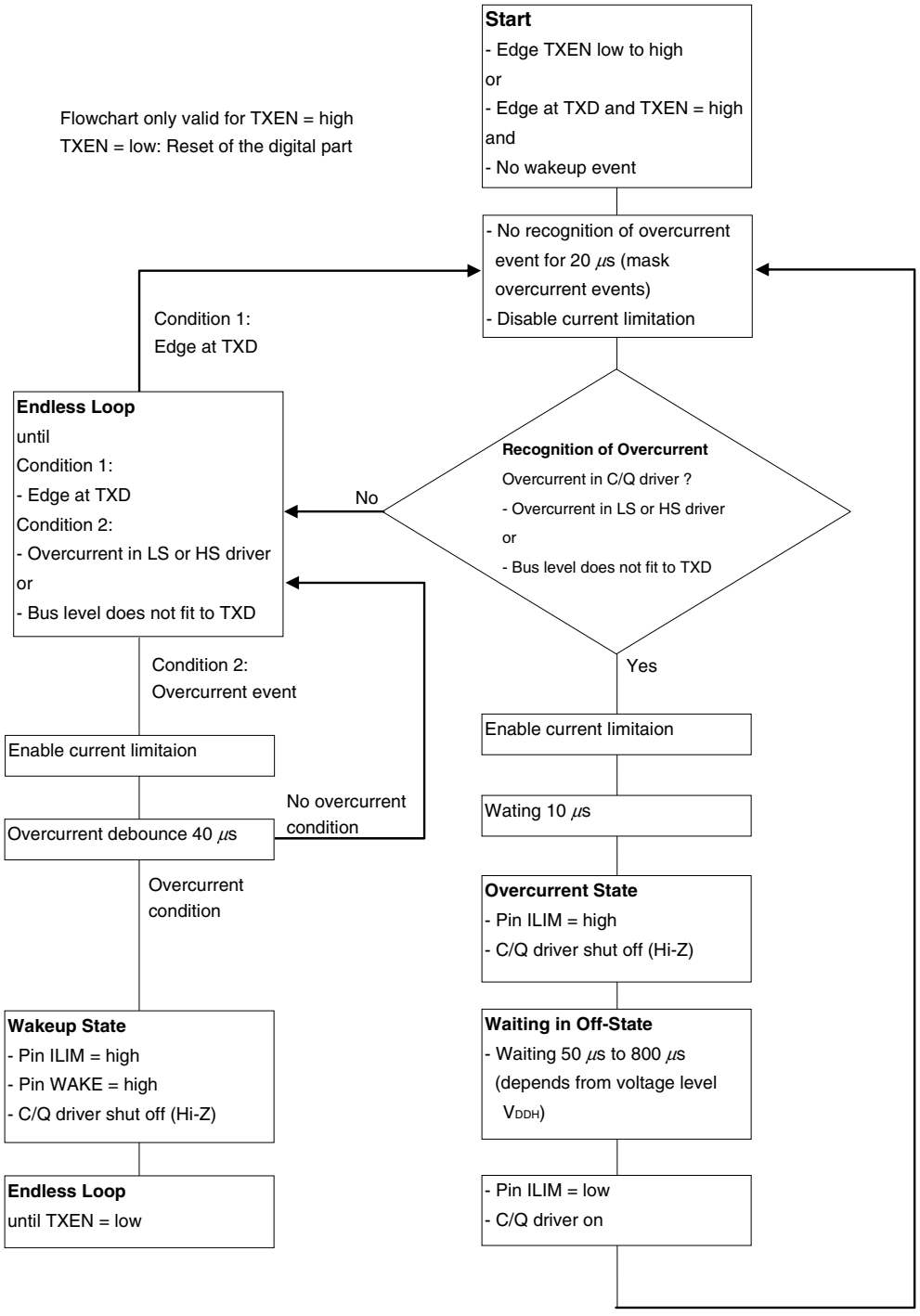
Table 4-3. Overcurrent Shutoff Threshold Level Selection

SILIM	Overcurrent shutoff threshold level
0	220 to 480 [mA]
1	110 to 240 [mA]

4.4 Wakeup Detection and Overcurrent Detection Functionality

Figure 4-3 shows the flowchart of wakeup detection and overcurrent detection functionality.

Figure 4-3. Overcurrent and wakeup functionality



4.5 Performing IO-Link Communication

4.5.1 Overview of performing IO-Link communication

The “Non Return to Zero” (NRZ) code is used for bit-by-bit coding.

Here, logic value “1” corresponds to a voltage value of 24 V between the L+ and C/Q lines.

Logic value “0” corresponds to a voltage value of 0 volts between the L+ and C/Q lines.

The open-circuit level on the C/Q line is 0 V with reference to L- and -24 V with reference to L+.

The UART format is used for character-by-character coding.

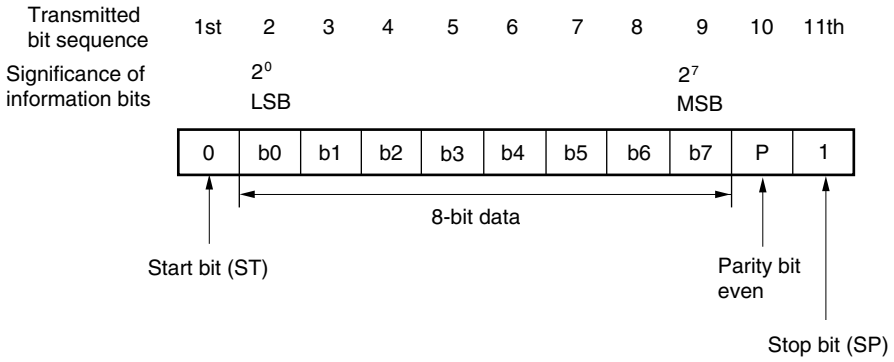
UART uses asynchronous communication.

A UART character is a character limited by a start signal and a stop signal and used for the asynchronous transmission of a telegram.

A start bit has logic value “0”.

Figure 4-4 shows the format of one UART code.

Figure 4-4. Structure of a one UART code

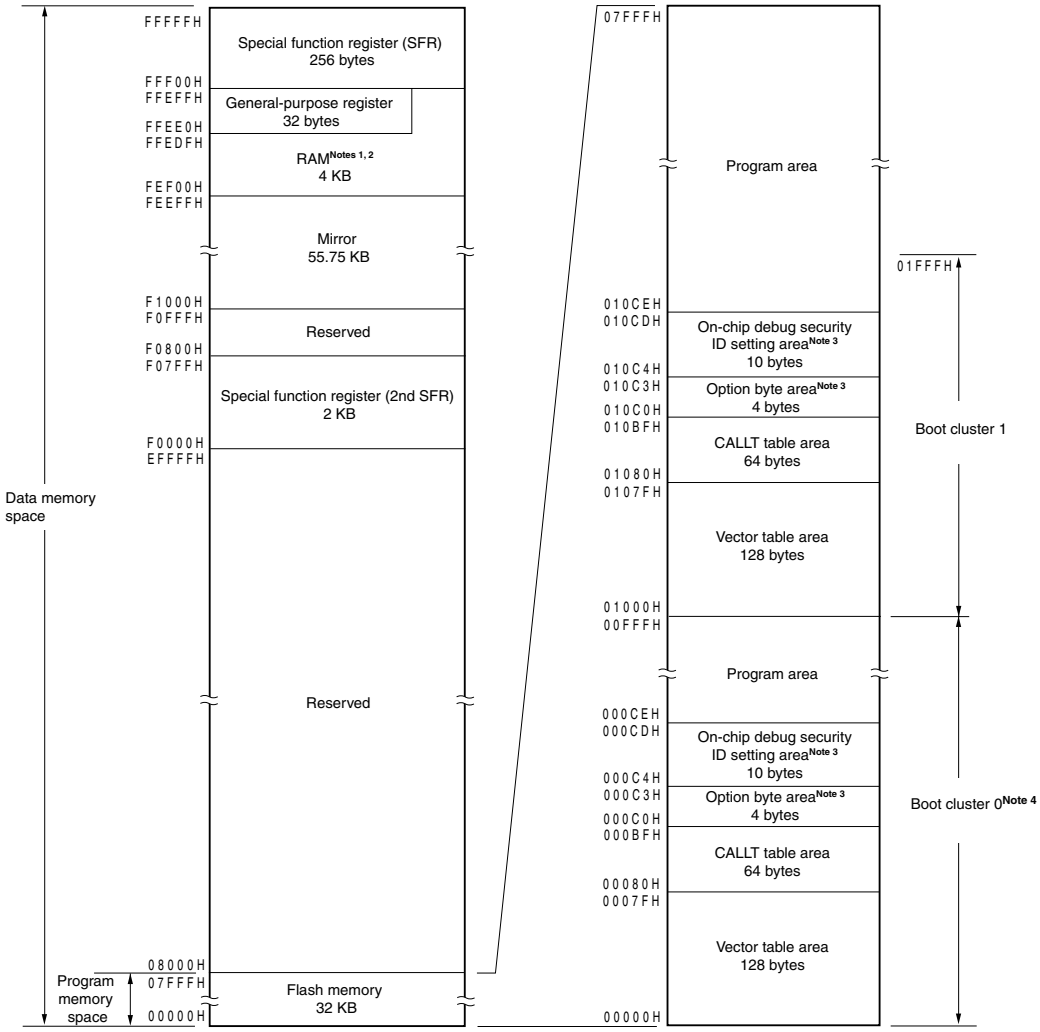


CHAPTER 5 CPU ARCHITECTURE

5.1 Memory Space

Products in the μPD78F8040, 78F8041, 78F8042, 78F8043 can access a 1 MB memory space. Figures 5-1 to 5-4 show the memory maps.

Figure 5-1. Memory Map (μPD78F8040)



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

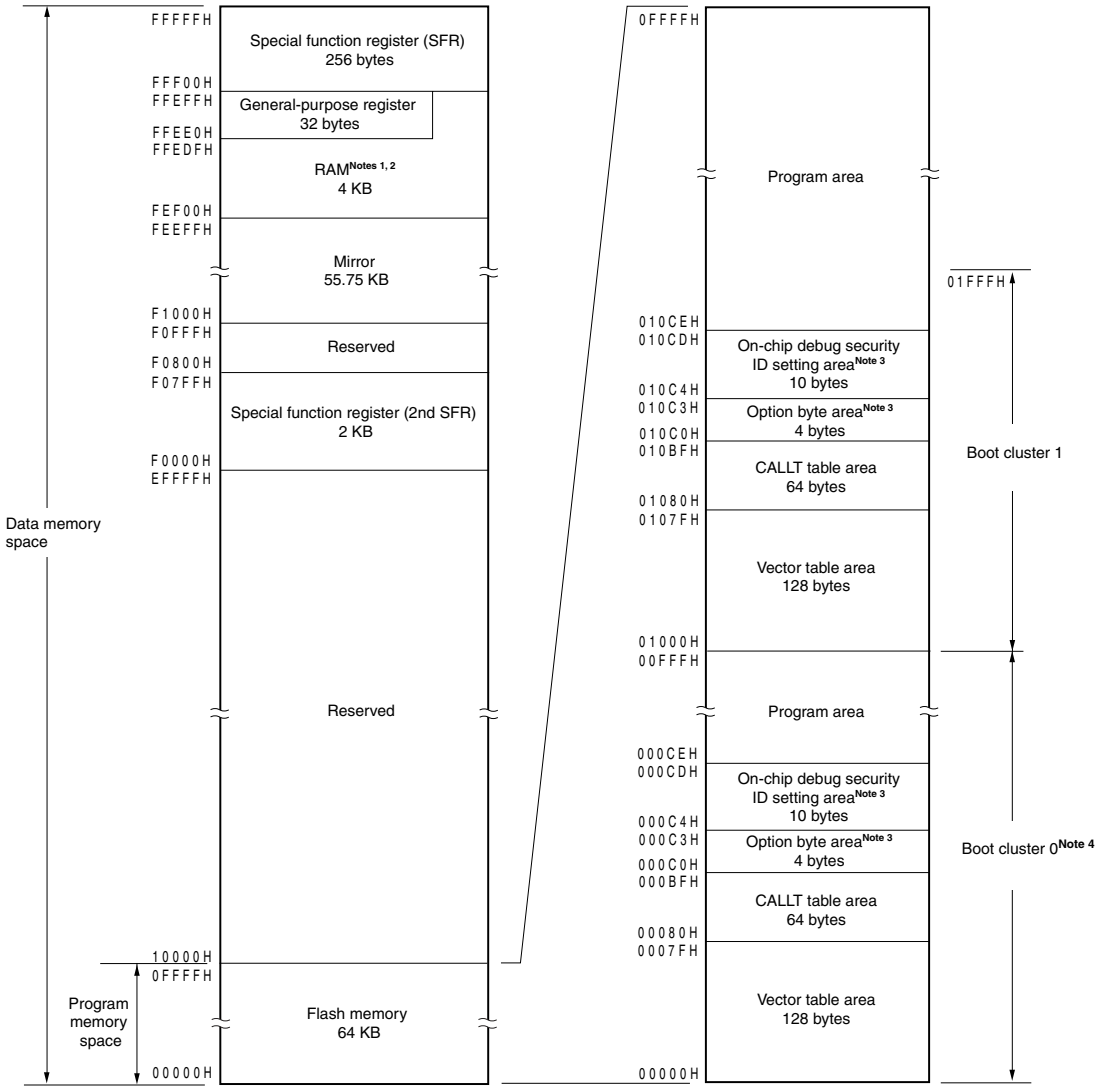
2. Instructions can be executed from the RAM area excluding the general-purpose register area.

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 **22.7 Security Setting**).

Figure 5-2. Memory Map (μPD78F8041)



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

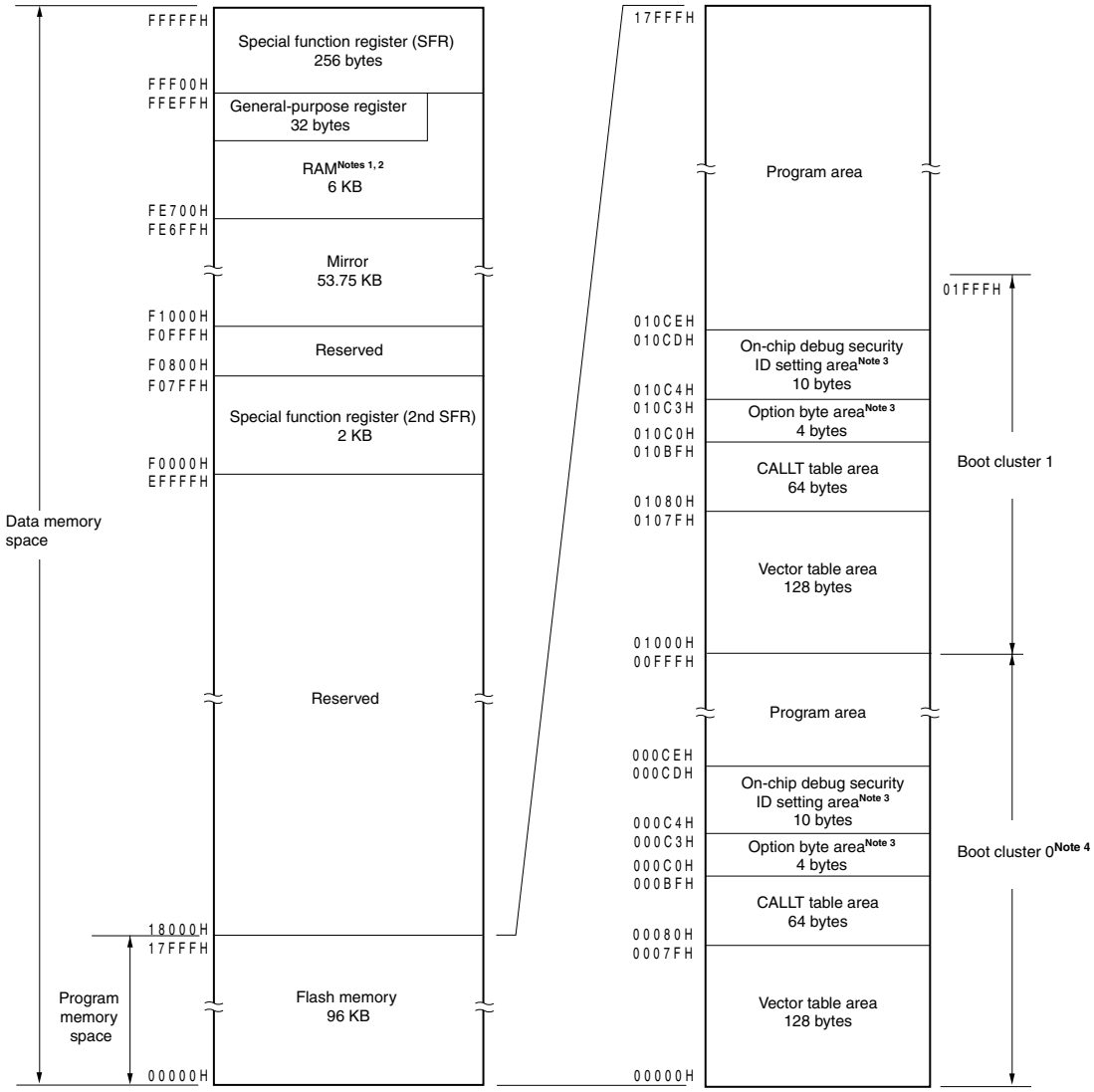
2. Instructions can be executed from the RAM area excluding the general-purpose register area.

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 **22.7 Security Setting**).

Figure 5-3. Memory Map (μPD78F8042)



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

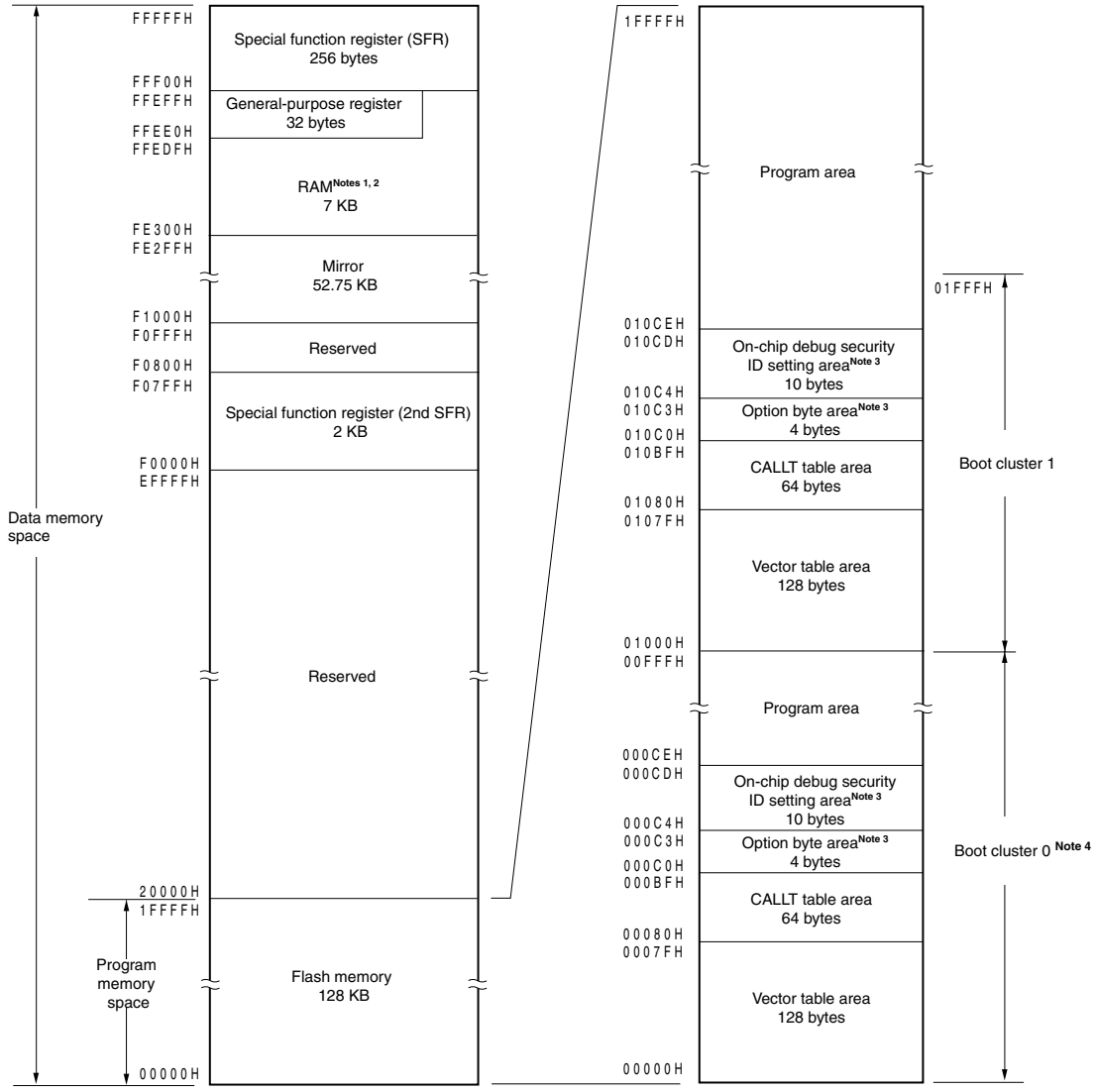
2. Instructions can be executed from the RAM area excluding the general-purpose register area.

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 **22.7 Security Setting**).

Figure 5-4. Memory Map (μPD78F8043)



Notes 1. Use of the area FFE20H to FFEDFH and FDF00H to FE2FFH are prohibited when using the self-programming function, since this area is used for self-programming library.

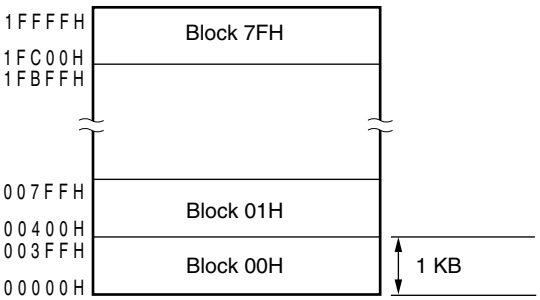
2. Instructions can be executed from the RAM area excluding the general-purpose register area.

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 **22.7 Security Setting**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 5-1 Correspondence Between Address Values and Block Numbers in Flash Memory**.



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

Table 5-1. Correspondence Between Address Values and Block Numbers in Flash Memory

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

Remark μPD78F8040: Block numbers 00H to 1FH
μPD78F8041: Block numbers 00H to 3FH
μPD78F8042: Block numbers 00H to 5FH
μPD78F8043: Block numbers 00H to 7FH

5.1.1 Internal program memory space

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

μPD78F8040, 78F8041, 78F8042, 78F8043 products incorporate internal ROM (flash memory), as shown below.

Table 5-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F8040	Flash memory	32768 × 8 bits (00000H to 07FFFH)
μPD78F8041		65536 × 8 bits (00000H to 0FFFFH)
μPD78F8042		98304 × 8 bits (00000H to 17FFFH)
μPD78F8043		131072 × 8 bits (00000H to 1FFFFH)

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.

Table 5-3. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
00000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT, TRAP	0002CH	INTTM00
		0002EH	INTTM01
00004H	INTWDTI	00030H	INTTM02
00006H	INTLVI	00032H	INTTM03
00008H	INTP0	00034H	INTAD
0000AH	INTP1	0003CH	INTST2/INTCSI20/INTIIC20
0000CH	INTP2	00040H	INTTM13
00010H	INTP4	00042H	INTTM04
00012H	INTP5	00044H	INTTM05
00014H	INTST3	00046H	INTTM06
00016H	INTSR3	00048H	INTTM07
00018H	INTSRE3	0004AH	INTSR2
0001AH	INTDMA0	00056H	INTTM10
0001CH	INTDMA1	00058H	INTTM11
0001EH	INTST0	0005AH	INTTM12
00020H	INTSR0	0005CH	INTSRE2
00022H	INTSRE0	0005EH	INTMD
0002AH	INTIICA	0007EH	BRK

(2) CALLT instruction table area

The 64-byte area 00080H to 000BFH can store the subroutine entry address of a 2-byte call instruction (CALLT). Set the subroutine entry address to a value in a range of 00000H to 0FFFFH (because an address code is 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 21 OPTION BYTE**.

(4) On-chip debug security ID setting area

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

5.1.2 Mirror area

The μPD78F8040, 78F8041, 78F8042, 78F8043 mirror the data flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH (the data flash area to be mirrored is set by the processor mode control register (PMC)).

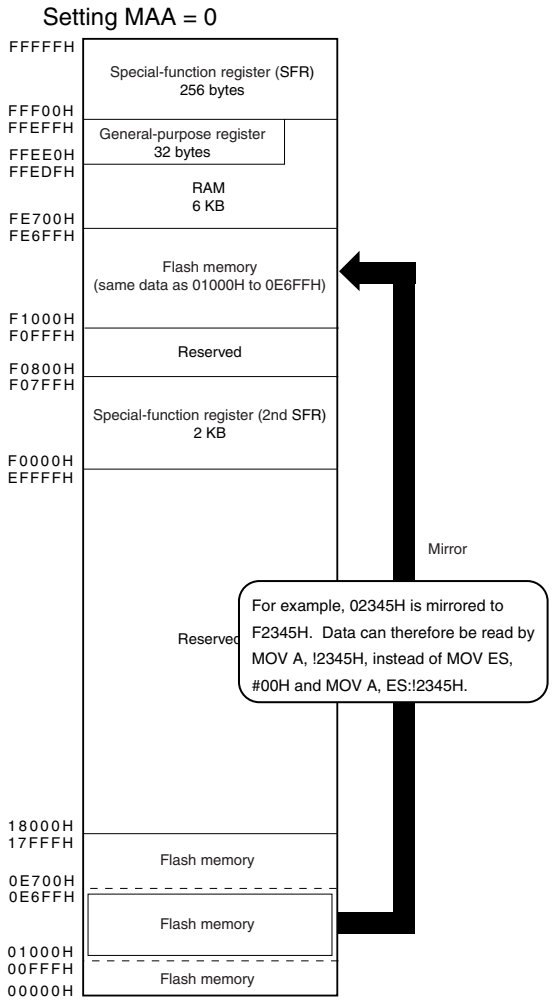
By reading data from F0000H to FFFFFH, an instruction that does not have the ES registers as an operand can be used, and thus the contents of the data flash can be read with the shorter code. However, the data flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

See **5.1 Memory Space** for the mirror area of each product.

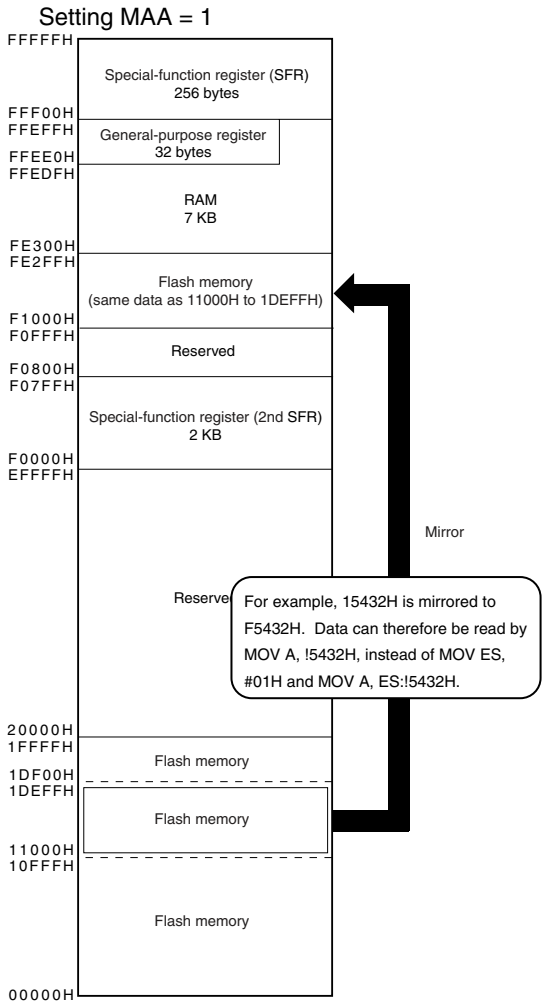
The mirror area can only be read and no instruction can be fetched from this area.

The following show examples.

Example 1 μPD78F8042
(Flash memory: 96 KB, RAM: 6 KB)



Example 2 μPD78F8043
(Flash memory: 128 KB, RAM: 7 KB)



Remark MAA: Bit 0 of the processor mode control register (PMC)

PMC register is described below.

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

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

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

Reset signal generation sets this register to 00H.

Figure 5-5. Format of Processor Mode Control Register (PMC)

Address: FFFFEH After reset: 00H R/W

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

MAA	Selection of flash memory space for mirroring to area from F0000H to FFFFFH
0	00000H to 0FFFFH is mirrored to F0000H to FFFFFH
1	10000H to 1FFFFH is mirrored to F0000H to FFFFFH

Note This setting is prohibited in the products of μPD78F8040 and 78F8041.

Cautions 1. In the products of μPD78F8040 and 78F8041, be sure to set bit 0 (MAA) to 0 (default value).

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

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

5.1.3 Internal data memory space

μPD78F8040, 78F8041, 78F8042, 78F8043 products incorporate the following RAMs.

Table 5-4. Internal RAM Capacity

Part Number	Internal RAM
μPD78F8040	4096 × 8 bits (FEF00H to FFEFFH)
μPD78F8041	4096 × 8 bits (FEF00H to FFEFFH)
μPD78F8042	6144 × 8 bits (FE700H to FFEFFH)
μPD78F8043	7168 × 8 bits (FE300H 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 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, the area of FFE20H to FFEFFH cannot be used as a stack memory.

5.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH (see **Table 5-5** in **5.2.4 Special function registers (SFRs)**).

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

5.1.5 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 5-6** in **5.2.5 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.

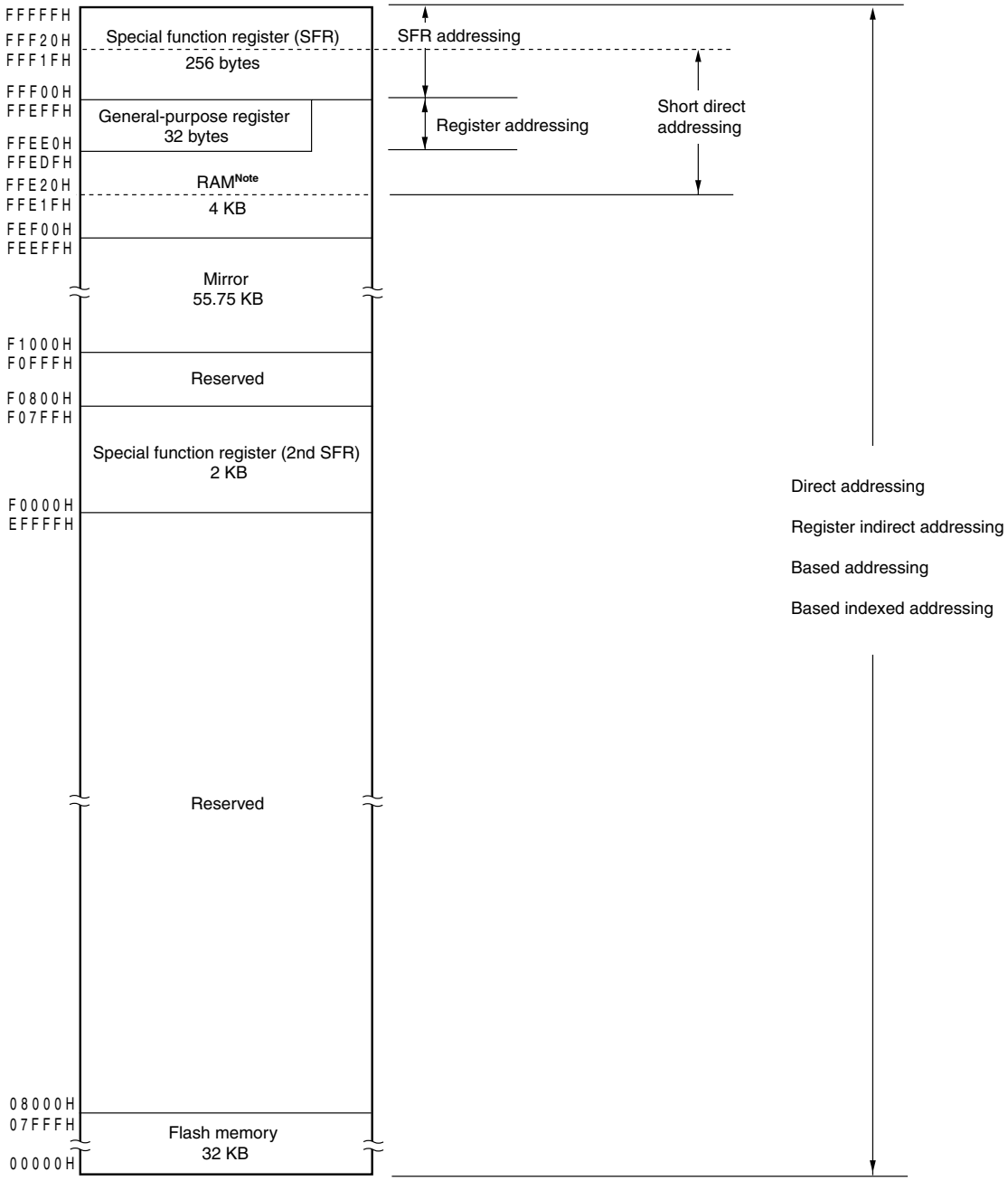
Caution Do not access addresses to which the 2nd SFR is not assigned.

5.1.6 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.

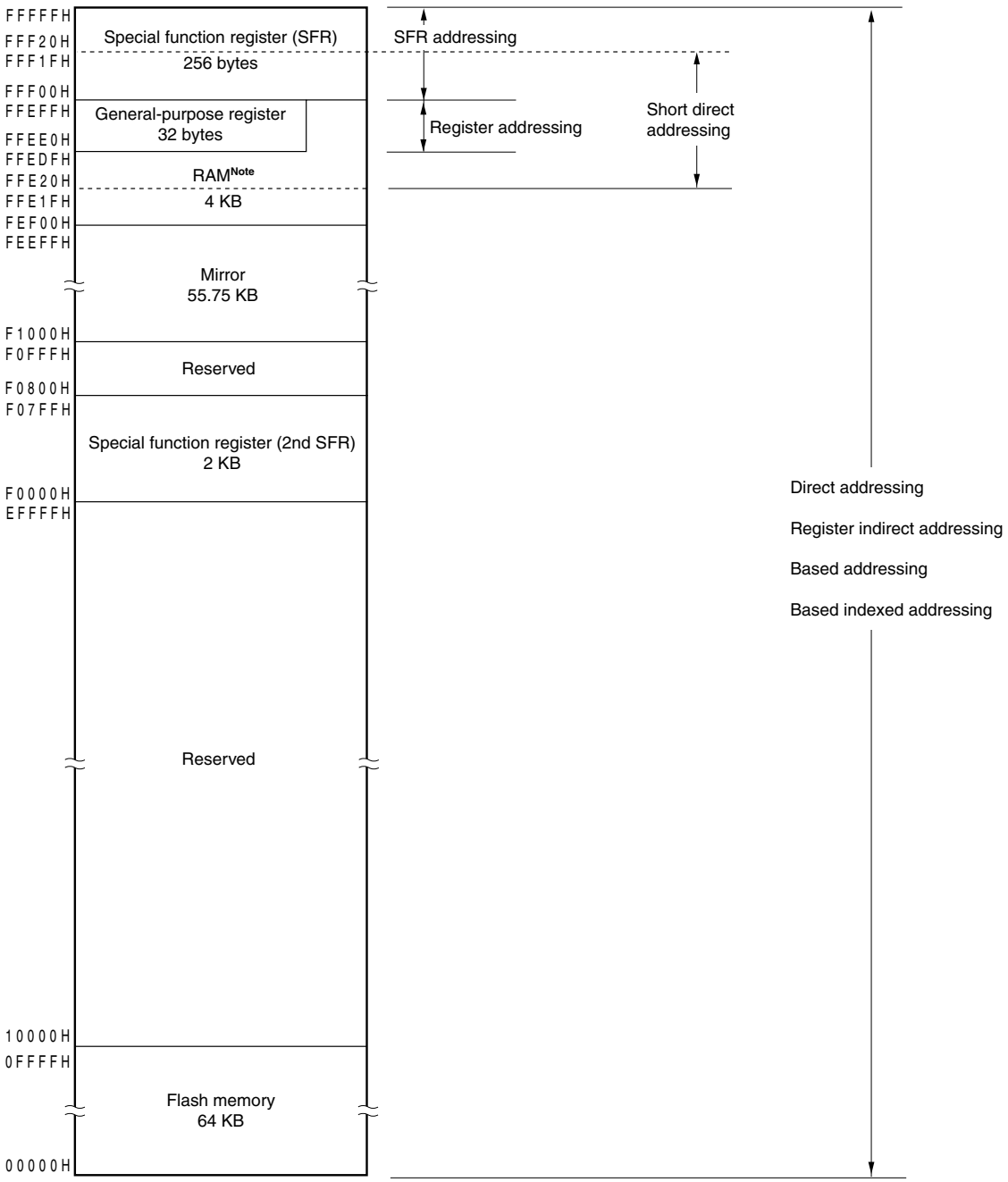
Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD78F8040, 78F8041, 78F8042, 78F8043, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figures 5-6 to 5-9 show correspondence between data memory and addressing.

Figure 5-6. Correspondence Between Data Memory and Addressing (μPD78F8040)



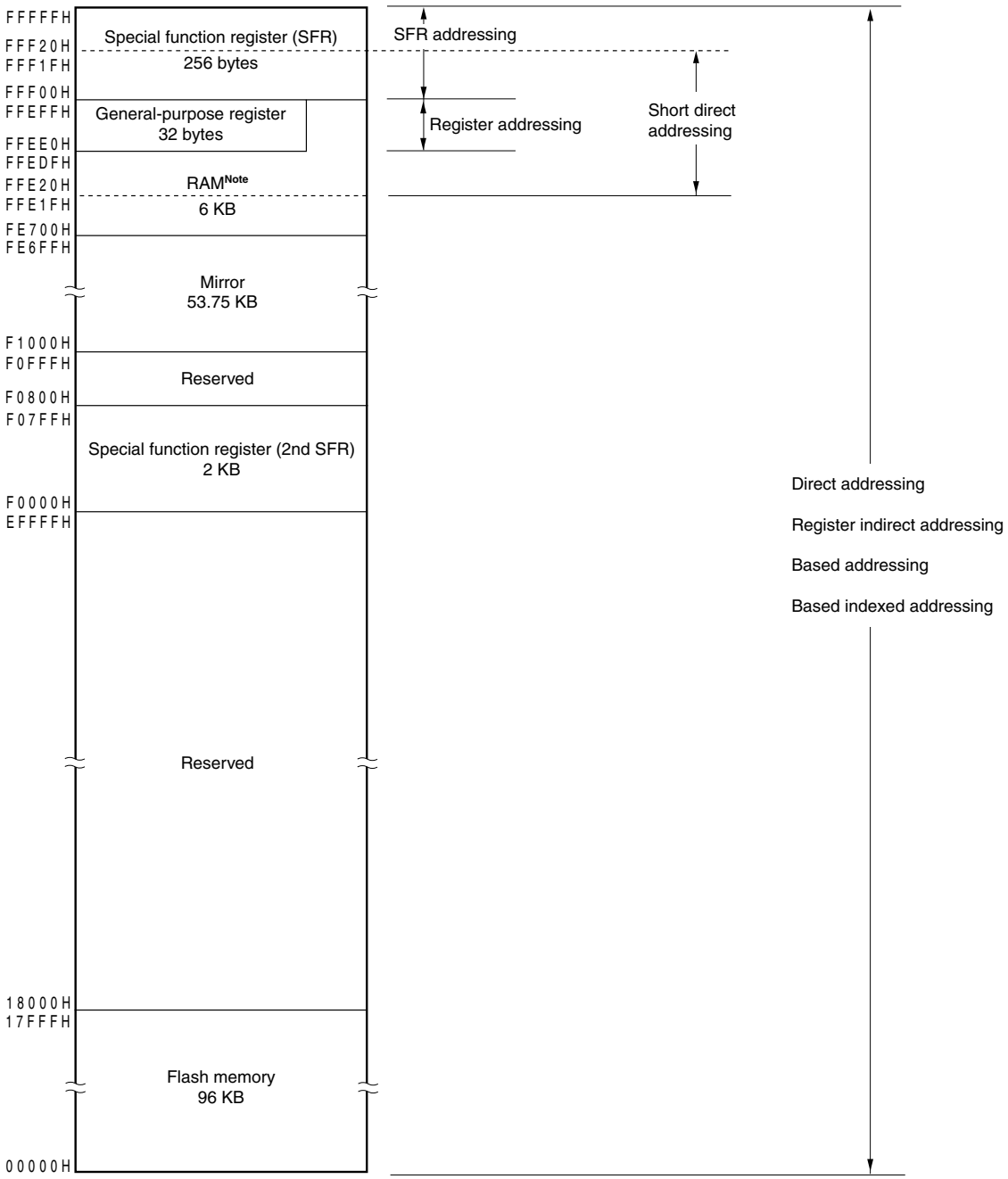
Note Use of the area FFE20H to FFEDFH is prohibited when using the self-programming function, since this area is used for self-programming library.

Figure 5-7. Correspondence Between Data Memory and Addressing (μPD78F8041)



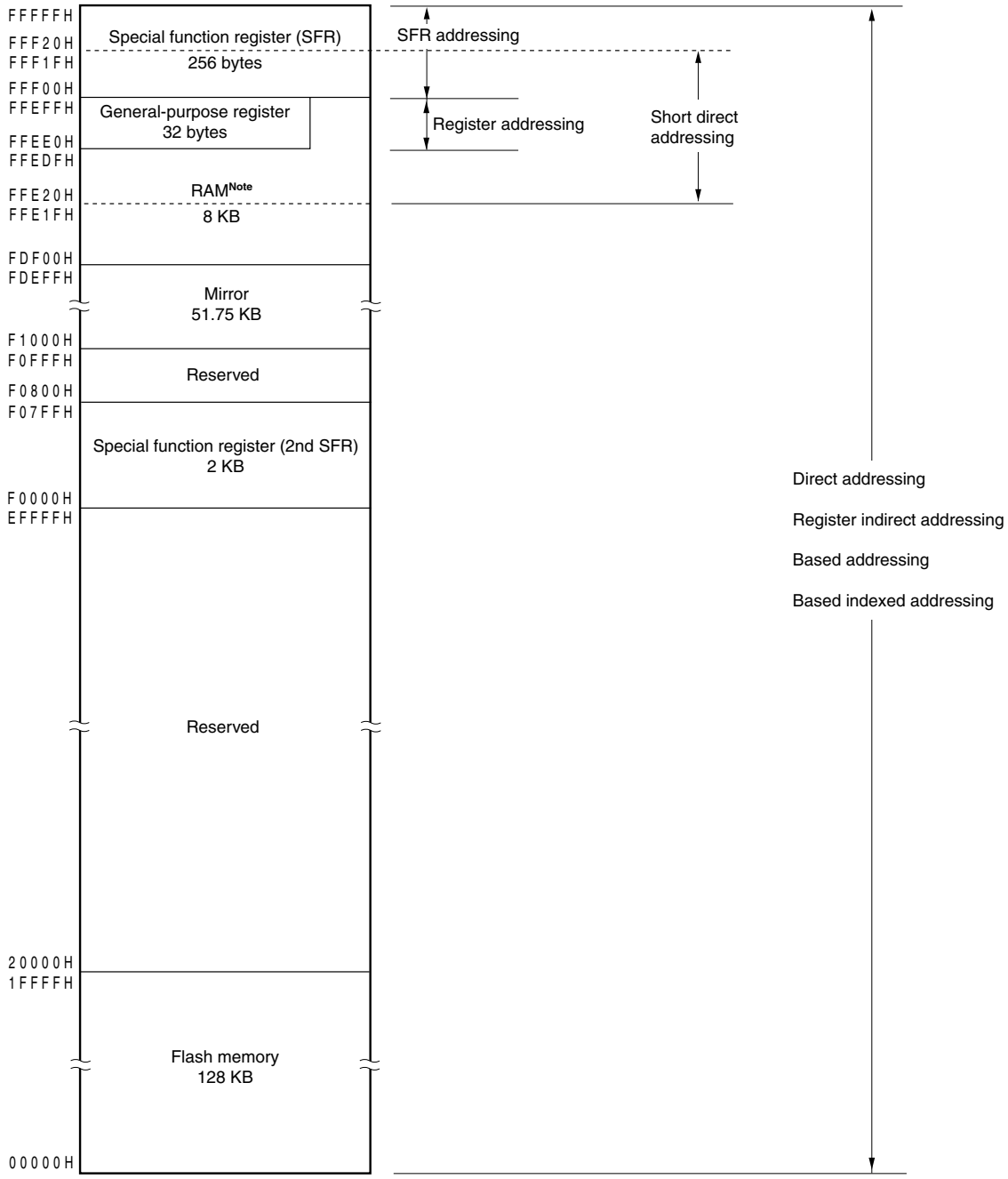
Note Use of the area FFE20H to FFEDFH is prohibited when using the self-programming function, since this area is used for self-programming library.

Figure 5-8. Correspondence Between Data Memory and Addressing (μPD78F8042)



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

Figure 5-9. Correspondence Between Data Memory and Addressing (μPD78F8043)



Note Use of the area FFE20H to FFEDFH and FDF00H to FE2FFH are prohibited when using the self-programming function, since this area is used for self-programming library.

5.2 Processor Registers

The μPD78F8040, 78F8041, 78F8042, 78F8043 products incorporate the following processor registers.

5.2.1 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)

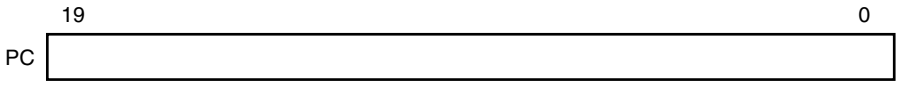
The program counter is a 20-bit register that holds the address information of the next program to be executed.

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 5-10. 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 vector interrupt request acknowledgment or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 5-11. 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 by a priority specification flag register (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **15.3 (3)**) cannot 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 5-12. 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 5-13.

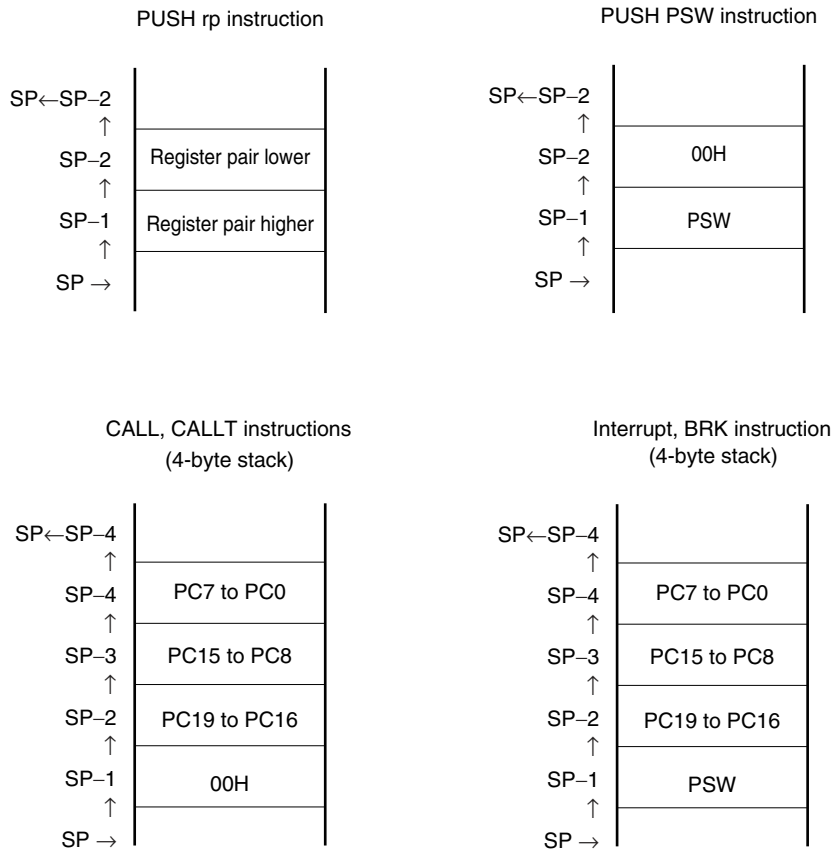
Cautions 1. Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

2. Be sure to specify an even number for the stack pointer. If an odd number is specified, the least significant bit is automatically set to 0.

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

4. While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory.

Figure 5-13. Data to Be Saved to Stack Memory



5.2.2 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).

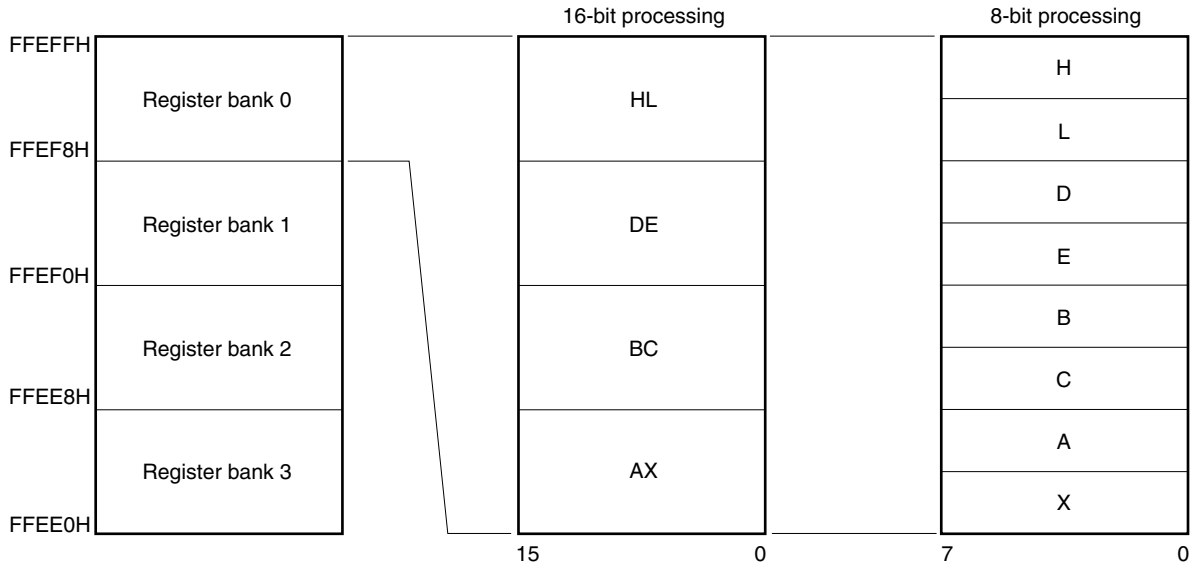
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

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

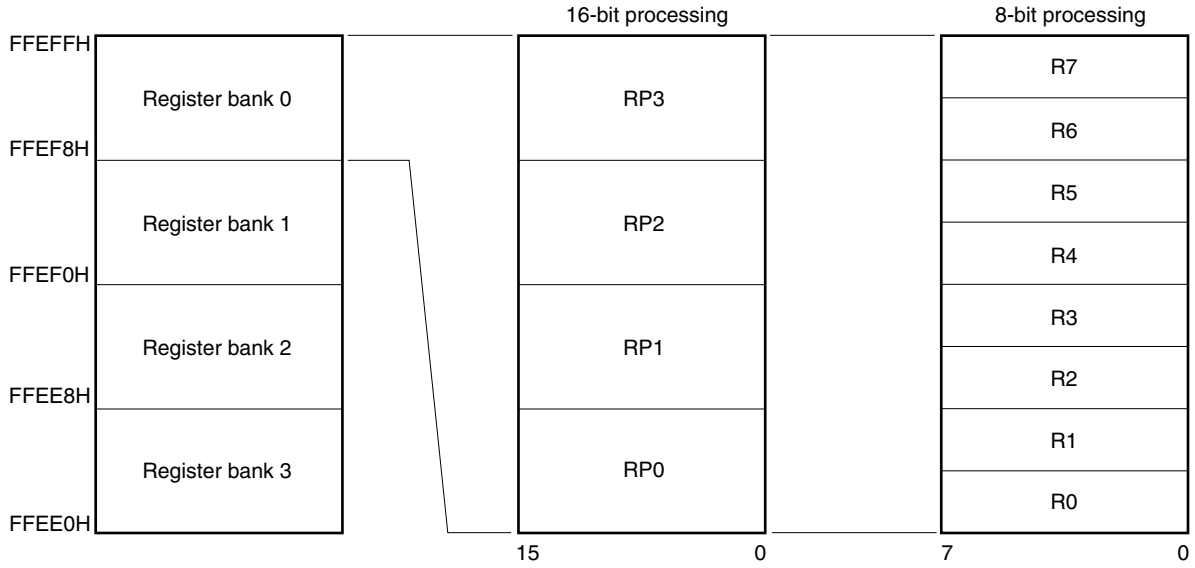
2. While using the self-programming function, the area FFE20H to FFEFFH cannot be used as stack memory.

Figure 5-14. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



5.2.3 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 5-15. 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

5.2.4 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 5-5 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 RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R. When using the RA78K0R, ID78K0R-QB, and SM+ for 78K0R, 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 SFRs are not assigned.

Remark For extended SFRs (2nd SFRs), see **5.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 5-5. 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
FFF0CH	Port register 12	P12		R/W	√	√	–	Undefined
FFF0EH	Port register 14	P14		R/W	√	√	–	00H
FFF0FH	Port register 15	P15		R/W	√	√	–	00H
FFF10H	Serial data register 00	TXD0	SDR00	R/W	–	√	√	0000H
FFF11H		–			–	–		
FFF12H	Serial data register 01	RXD0	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–		
FFF14H	Serial data register 12	TXD3	SDR12	R/W	–	√	√	0000H
FFF15H		–			–	–		
FFF16H	Serial data register 13	RXD3	SDR13	R/W	–	√	√	0000H
FFF17H		–			–	–		
FFF18H	Timer data register 00	TDR00		R/W	–	–	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01		R/W	–	–	√	0000H
FFF1BH								
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	PM7Note		R/W	√	√	–	FFH
FFF2BH	Port mode register 11	PM11Note		R/W	√	√	–	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	–	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	–	FFH
FFF2FH	Port mode register 15	PM15		R/W	√	√	–	FFH
FFF30H	A/D converter mode register	ADM		R/W	√	√	–	00H
FFF31H	Analog input channel specification register	ADS		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
FFF3CH	Input switch control register	ISC		R/W	√	√	–	00H

Note Port mode register of unused internal pins. See **2.2 Initial Setting of Unused internal pins in MCU**.

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF48H	Serial data register 10	TXD2/ SIO20	SDR10	R/W	–	√	√	0000H
FFF49H		–			–	–		
FFF4AH	Serial data register 11	RXD2	SDR11	R/W	–	√	√	0000H
FFF4BH		–			–	–		
FFF50H	IICA shift register	IICA		R/W	–	√	–	00H
FFF51H	IICA status register	IICS		R	√	√	–	00H
FFF52H	IICA flag register	IICF		R/W	√	√	–	00H
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03		R/W	–	–	√	0000H
FFF67H								
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								
FFF70H	Timer data register 10	TDR10		R/W	–	–	√	0000H
FFF71H								
FFF72H	Timer data register 11	TDR11		R/W	–	–	√	0000H
FFF73H								
FFF74H	Timer data register 12	TDR12		R/W	–	–	√	0000H
FFF75H								
FFF76H	Timer data register 13	TDR13		R/W	–	–	√	0000H
FFF77H								
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	√	√	–	09H

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFA8H	Reset control flag register	RESF		R	–	√	–	Undefined Note 1
FFFA9H	Low-voltage detection register	LVIM		R/W	√	√	–	00H Note 2
FFFAAH	Low-voltage detection level select register	LVIS		R/W	√	√	–	0EH Note 3
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1A/9A Note 4
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
FFFBEH	Back ground event control register	BECTL		R/W	√	√	–	00H
FFFC0H	–	PFCMD Note 5		–	–	–	–	Undefined
FFFC2H	–	PFS Note 5		–	–	–	–	Undefined
FFFC4H	–	FLPMC Note 5		–	–	–	–	Undefined
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH	Priority specification flag register 12H	PR12H		R/W	√	√		FFH
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

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

2. The reset value of LVIM varies depending on the reset source and the setting of the option byte.

3. The reset value of LVIS varies depending on the reset source.

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

5. Do not directly operate this SFR, because it is to be used in the self programming library.

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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 5-6 Extended SFR (2nd SFR) List**.

5.2.5 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 5-6 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 RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R. When using the RA78K0R, ID78K0R-QB, and SM+ for 78K0R, 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 the 2nd SFR is not assigned.

Remark For SFRs in the SFR area, see **5.2.4 Special function registers (SFRs)**.

Table 5-6. Extended SFR (2nd SFR) List (1/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0017H	A/D port configuration register	ADPC	R/W	–	√	–	10H
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	√	√	–	00H
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H
F0036H	Pull-up resistor option register 6	PU6	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
F004EH	Port input mode register 14	PIM14	R/W	√	√	–	00H
F005EH	Port output mode register 14	POM14	R/W	√	√	–	00H
F0060H	Noise filter enable register 0	NFEN0	R/W	√	√	–	00H
F0061H	Noise filter enable register 1	NFEN1	R/W	√	√	–	00H
F0062H	Noise filter enable register 2	NFEN2	R/W	√	√	–	00H
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
F00F4H	Regulator mode control register	RMC	R/W	–	√	–	00H
F00F6H	20 MHz internal high-speed oscillation control register	DSCCTL	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		–		–	–		
F0108H	Serial flag clear trigger register 00	SIR00L SIR00	R/W	–	√	√	0000H
F0109H		–		–	–		
F010AH	Serial flag clear trigger register 01	SIR01L SIR01	R/W	–	√	√	0000H
F010BH		–		–	–		
F0110H	Serial mode register 00	SMR00	R/W	–	–	√	0020H
F0111H							
F0112H	Serial mode register 01	SMR01	R/W	–	–	√	0020H
F0113H							

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		–			–	–		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		–			–	–		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		–			–	–		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	–	√	√	0000H
F0127H		–			–	–		
F0128H	Serial output register 0	SO0		R/W	–	–	√	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		–			–	–		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	–	√	√	0000H
F0135H		–			–	–		
F0140H	Serial status register 10	SSR10L	SSR10	R	–	√	√	0000H
F0141H		–			–	–		
F0142H	Serial status register 11	SSR11L	SSR11	R	–	√	√	0000H
F0143H		–			–	–		
F0144H	Serial status register 12	SSR12L	SSR12	R	–	√	√	0000H
F0145H		–			–	–		
F0146H	Serial status register 13	SSR13L	SSR13	R	–	√	√	0000H
F0147H		–			–	–		
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	–	√	√	0000H
F0149H		–			–	–		
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	–	√	√	0000H
F014BH		–			–	–		
F014CH	Serial flag clear trigger register 12	SIR12L	SIR12	R/W	–	√	√	0000H
F014DH		–			–	–		
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	–	√	√	0000H
F014FH		–			–	–		
F0150H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	–	–	√	0020H
F0153H								
F0154H	Serial mode register 12	SMR12		R/W	–	–	√	0020H
F0155H								
F0156H	Serial mode register 13	SMR13		R/W	–	–	√	0020H
F0157H								

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0158H	Serial communication operation setting register 10	SCR10		R/W	–	–	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	–	–	√	0087H
F015BH								
F015CH	Serial communication operation setting register 12	SCR12		R/W	–	–	√	0087H
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W	–	–	√	0087H
F015FH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0161H		–			–	–		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		–			–	–		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		–			–	–		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	–	√	√	0000H
F0167H		–			–	–		
F0168H	Serial output register 1	SO1		R/W	–	–	√	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	√	√	√	0000H
F016BH		–			–	–		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	–	√	√	0000H
F0175H		–			–	–		
F0180H	Timer counter register 00	TCR00		R	–	–	√	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	–	–	√	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	–	–	√	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	–	–	√	FFFFH
F0187H								
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								

Table 5-6. Extended SFR (2nd SFR) List (4/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		–			–	–		
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	TPS0L	TPS0	R/W	–	√	√	0000H
F01B7H		–			–	–		

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01B8H	Timer output register 0	TO0L	TO0	R/W	–	√	√	0000H
F01B9H		–			–	–		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		–			–	–		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	–	√	√	0000H
F01BDH		–			–	–		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	–	√	√	0000H
F01BFH		–			–	–		
F01C0H	Timer counter register 10	TCR10		R	–	–	√	FFFFH
F01C1H								
F01C2H	Timer counter register 11	TCR11		R	–	–	√	FFFFH
F01C3H								
F01C4H	Timer counter register 12	TCR12		R	–	–	√	FFFFH
F01C5H								
F01C6H	Timer counter register 13	TCR13		R	–	–	√	FFFFH
F01C7H								
F01C8H	Timer mode register 10	TMR10		R/W	–	–	√	0000H
F01C9H								
F01CAH	Timer mode register 11	TMR11		R/W	–	–	√	0000H
F01CBH								
F01CCH	Timer mode register 12	TMR12		R/W	–	–	√	0000H
F01CDH								
F01CEH	Timer mode register 13	TMR13		R/W	–	–	√	0000H
F01CFH								
F01D0H	Timer status register 10	TSR10L	TSR10	R	–	√	√	0000H
F01D1H		–			–	–		
F01D2H	Timer status register 11	TSR11L	TSR11	R	–	√	√	0000H
F01D3H		–			–	–		
F01D4H	Timer status register 12	TSR12L	TSR12	R	–	√	√	0000H
F01D5H		–			–	–		
F01D6H	Timer status register 13	TSR13L	TSR13	R	–	√	√	0000H
F01D7H		–			–	–		
F01D8H	Timer channel enable status register 1	TE1L	TE1	R	√	√	√	0000H
F01D9H		–			–	–		
F01DAH	Timer channel start register 1	TS1L	TS1	R/W	√	√	√	0000H
F01DBH		–			–	–		
F01DCH	Timer channel stop register 1	TT1L	TT1	R/W	√	√	√	0000H
F01DDH		–			–	–		
F01DEH	Timer clock select register 1	TPS1L	TPS1	R/W	–	√	√	0000H
F01DFH		–			–	–		
F01E0H	Timer output register 1	TO1L	TO1	R/W	–	√	√	0000H
F01E1H		–			–	–		

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01E2H	Timer output enable register 1	TOE1L	TOE1	R/W	√	√	√	0000H
F01E3H		–			–	–		
F01E4H	Timer output level register 1	TOL1L	TOL1	R/W	–	√	√	0000H
F01E5H		–			–	–		
F01E6H	Timer output mode register 1	TOM1L	TOM1	R/W	–	√	√	0000H
F01B7H		–			–	–		
F0230H	IICA control register 0	IICCTL0		R/W	√	√	–	00H
F0231H	IICA control register 1	IICCTL1		R/W	√	√	–	00H
F0232H	IICA low-level width setting register	IICWL		R/W	–	√	–	FFH
F0233H	IICA high-level width setting register	IICWH		R/W	–	√	–	FFH
F0234H	Slave address register	SVA		R/W	–	√	–	00H

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

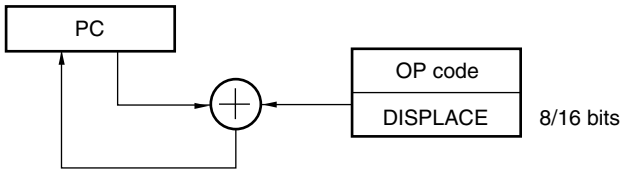
5.3 Instruction Address Addressing

5.3.1 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 5-16. Outline of Relative Addressing



5.3.2 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 5-17. Example of CALL !!addr20/BR !!addr20

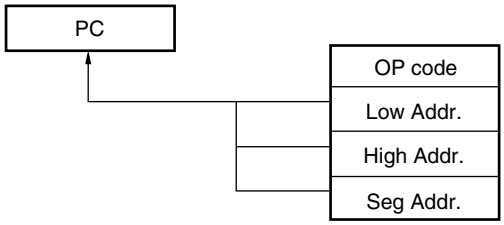
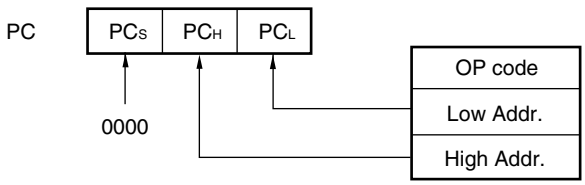


Figure 5-18. Example of CALL !addr16/BR !addr16



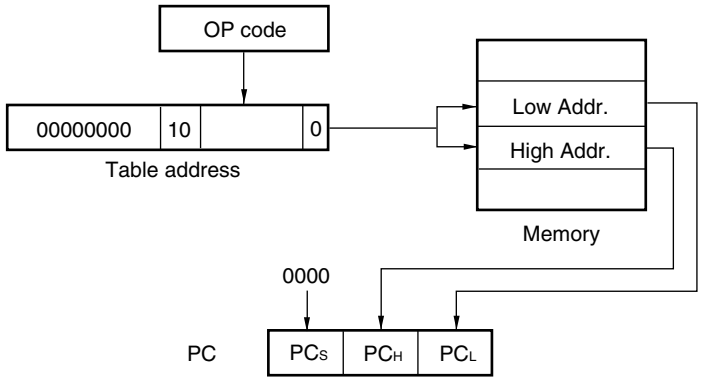
5.3.3 Table indirect addressing

[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 78K0R microcontrollers, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

Figure 5-19. Outline of Table Indirect Addressing

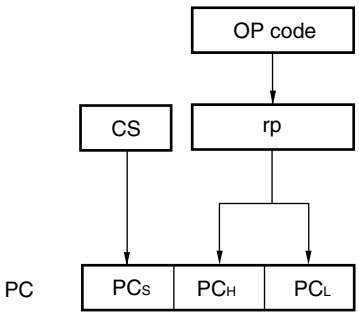


5.3.4 Register direct addressing

[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 5-20. Outline of Register Direct Addressing



5.4 Addressing for Processing Data Addresses

5.4.1 Implied addressing

[Function]

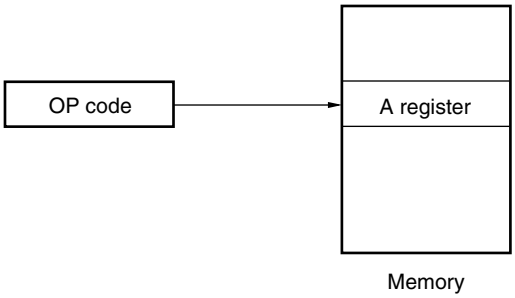
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[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 5-21. Outline of Implied Addressing



5.4.2 Register addressing

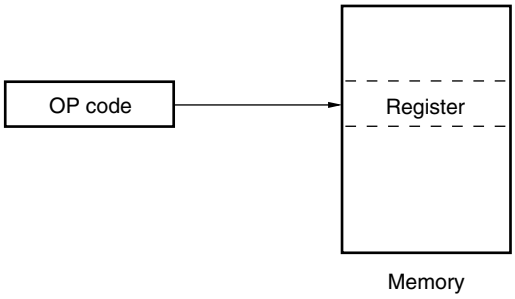
[Function]

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 5-22. Outline of Register Addressing



5.4.3 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 5-23. Example of ADDR16

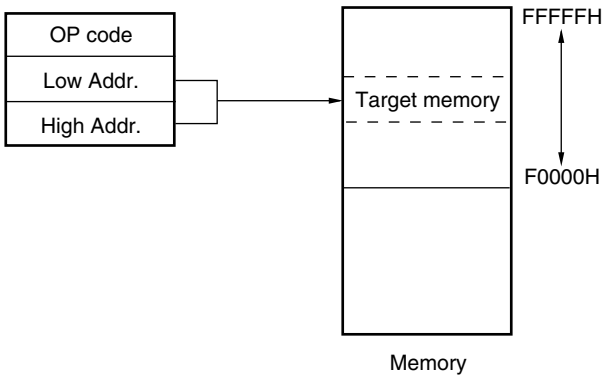
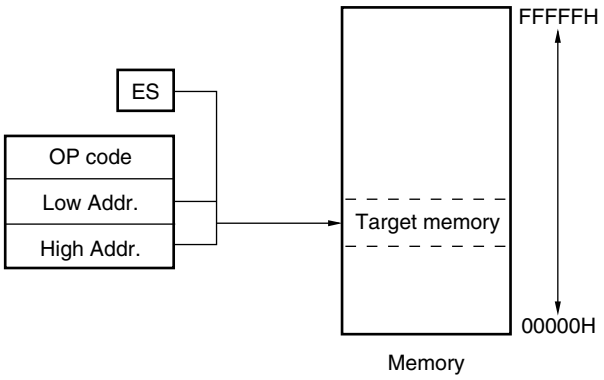


Figure 5-24. Example of ES:ADDR16



5.4.4 Short direct addressing

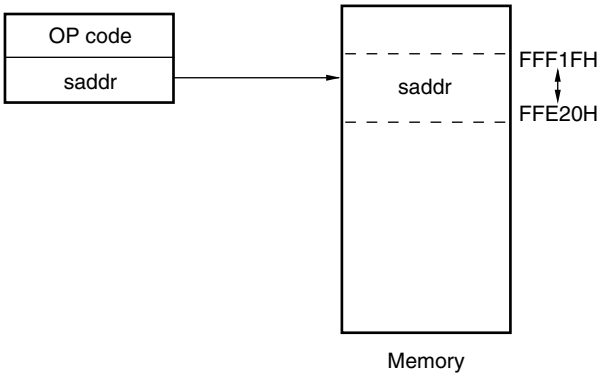
[Function]

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 5-25. 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 16-bit or 20-bit immediate data is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

5.4.5 SFR addressing

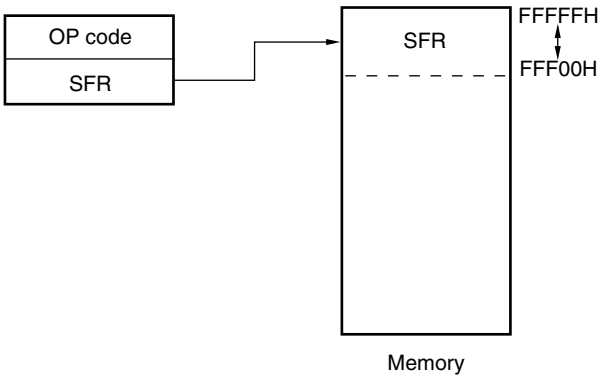
[Function]

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-manipulable SFR name (even address only)

Figure 5-26. Outline of SFR Addressing



5.4.6 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 5-27. Example of [DE], [HL]

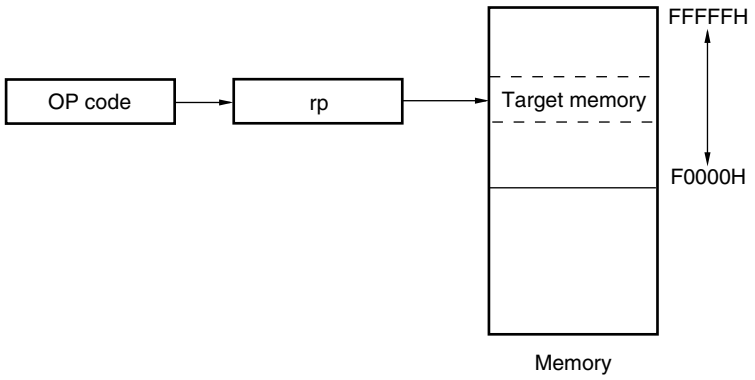
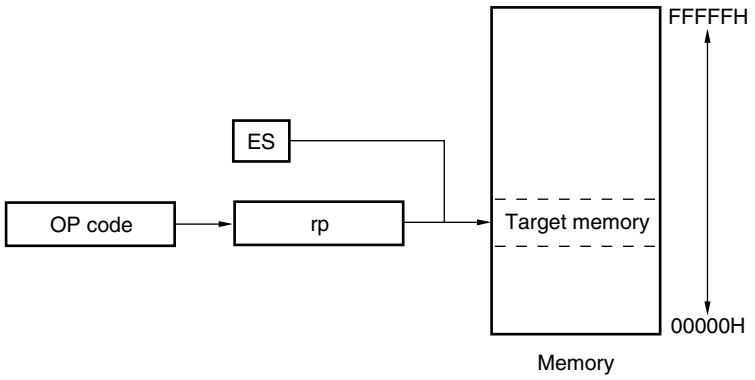


Figure 5-28. Example of ES:[DE], ES:[HL]



5.4.7 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 5-29. Example of [SP+byte]

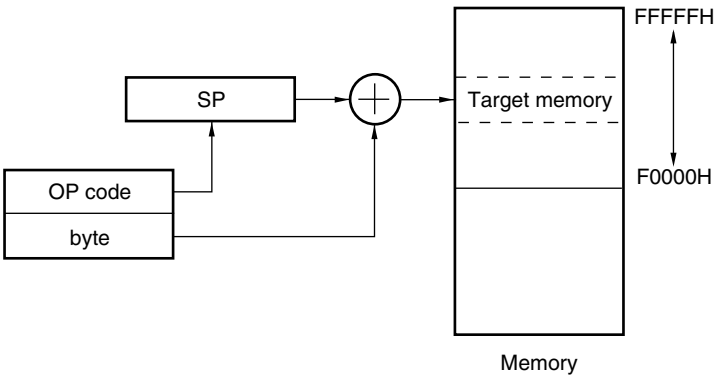


Figure 5-30. Example of [HL + byte], [DE + byte]

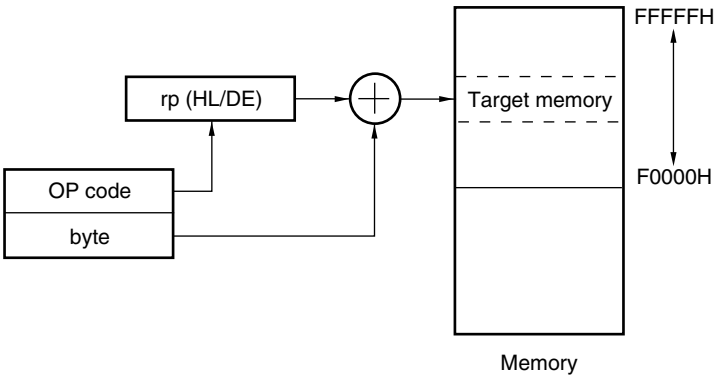


Figure 5-31. Example of word[B], word[C]

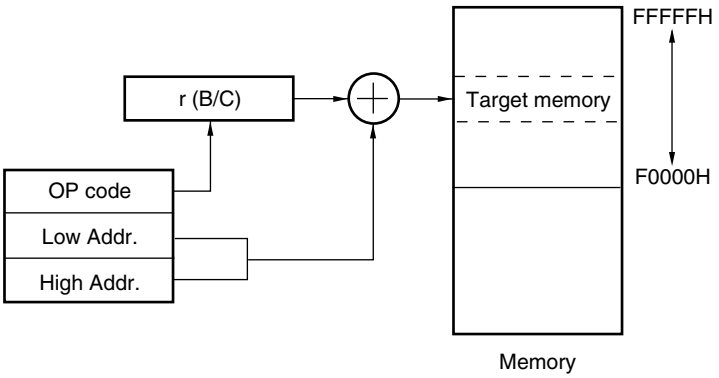


Figure 5-32. Example of word[BC]

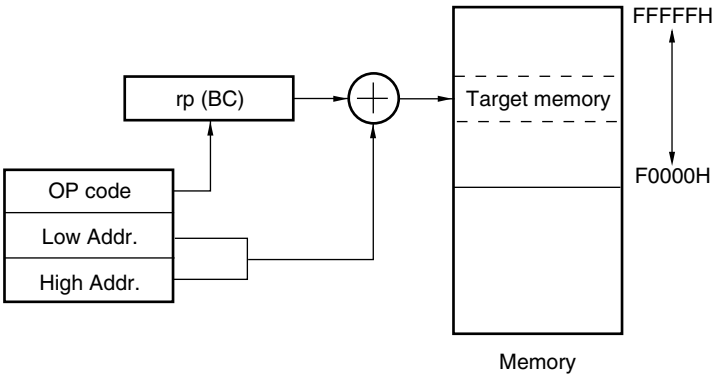


Figure 5-33. Example of ES:[HL + byte], ES:[DE + byte]

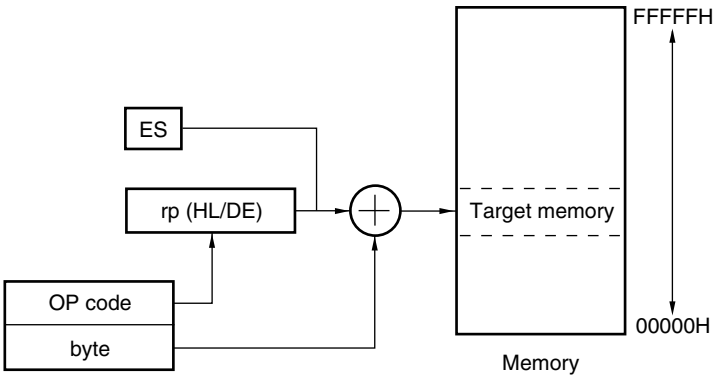


Figure 5-34. Example of ES:word[B], ES:word[C]

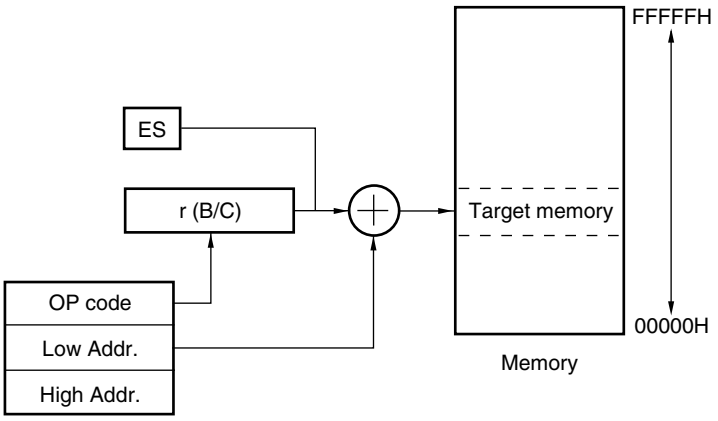
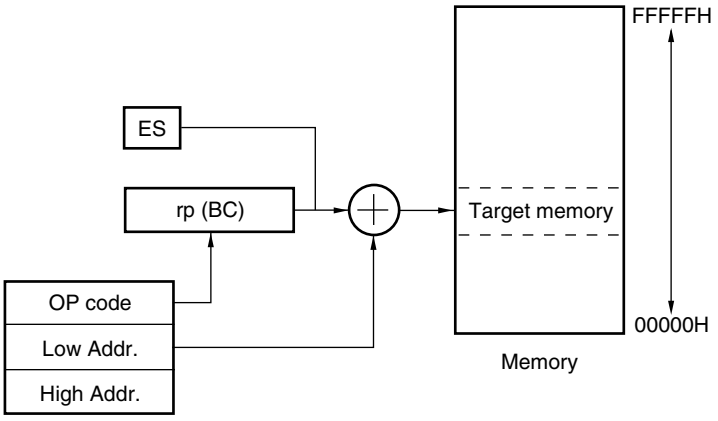


Figure 5-35. Example of ES:word[BC]



5.4.8 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 5-36. Example of [HL+B], [HL+C]

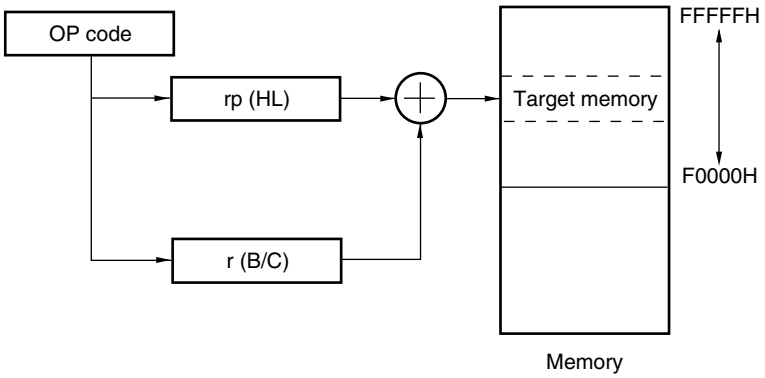
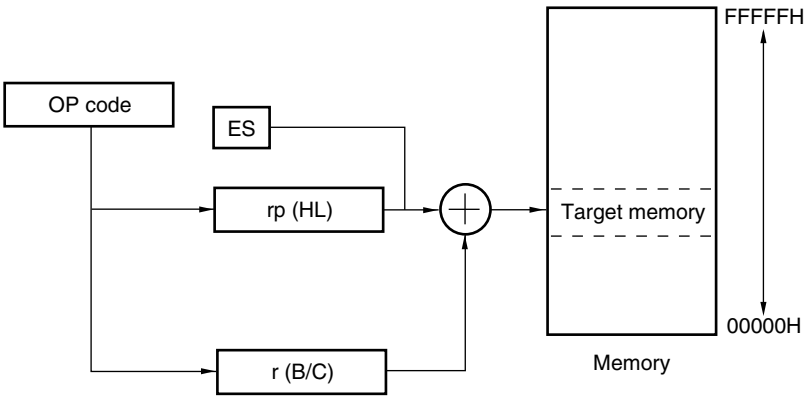


Figure 5-37. Example of ES:[HL+B], ES:[HL+C]



5.4.9 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 6 PORT FUNCTIONS

6.1 Port Functions

There are two types of pin I/O buffer power supplies: AV_{REF} and V_{DD} /EV_{DD}. The relationship between these power supplies and the pins is shown below.

Table 6-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
V_{DD}/EV_{DD}	• Port pins other than P26 to P27, and P150 to P153 • Non-port pins
AV_{REF}	P26, P27, P150 to P153

μPD78F8040, 78F8041, 78F8042, 78F8043 products are provided with the digital I/O ports shown in Figure 6-1, which enable variety of control operations. The functions of each port are shown in Table 6-2.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 3 PIN FUNCTIONS**.

Figure 6-1. Port Types

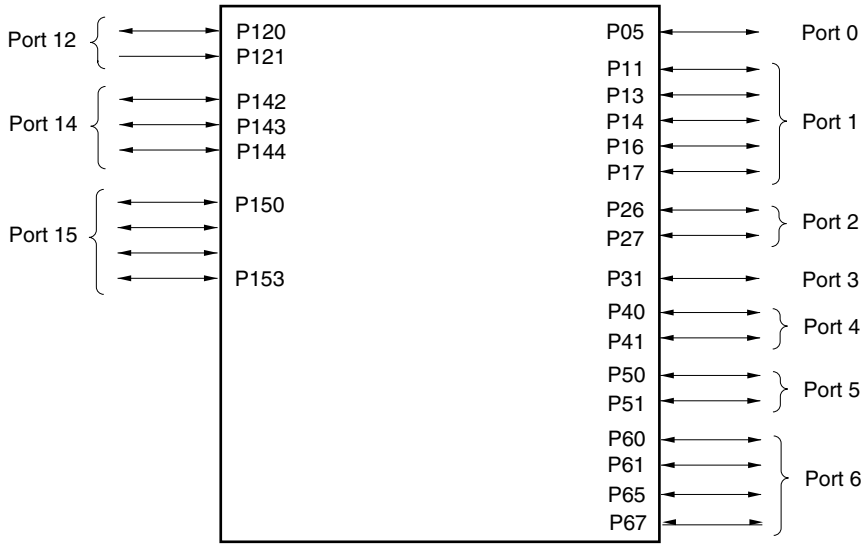


Table 6-2. Port Functions (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P05	I/O	Port 0. 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	TI05/TO05
P11 Note 1	I/O	Port 1. 5-bit I/O port. Input of P11 can be set to TTL input buffer. 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	RxD0
P13				TxD3
P14				RxD3
P16				TI01/TO01/INTP5
P17				TI02/TO02
P26	I/O	Port 2. 2-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI6
P27				ANI7
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
P40 Note 2	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				TOOL1
P50 Note 1	I/O	Port 5. 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	INTP1
P51 Note 1				INTP2
P60	I/O	Port 6. 4-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. For only P65 and P67, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCL0
P61				SDA0
P65				TI11/TO11
P67				TI13/TO13
P120	I/O	Port 12. 1-bit I/O port and 1-bit input port. 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.	Input port	INTP0/EXLVI
P121	Input			X1

Notes 1. Connect P11/RxD0, P50/INTP1, and P51/INTP2 to RXD, ILIM, and WAKE respectively on PCB by users.

2. If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally (see **Caution** in **3.2.5 P40, P41 (port 4)**).

Table 6-2. Port Functions (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P142	I/O	Port 14. 3-bit I/O port. Input of P142 and P143 can be set to TTL input buffer. Output of P142 to P144 can be set to the 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	$\overline{SCK20}$/SCL20
P143				SI20/RxD2/SDA20
P144				SO20/TxD2
P150 to P153	I/O	Port 15. 4-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8 to ANI11

6.2 Port Configuration

Ports include the following hardware.

Table 6-3. Port Configuration

Item	Configuration
Control registers	Port mode registers (PM0 to PM7, PM11, PM12, PM14, PM15) Port registers (P0 to P7, P11, P12, P14, P15) Pull-up resistor option registers (PU0, PU1, PU3 to PU6, PU12, PU14) Port input mode register 14 (PIM14) Port output mode register 14 (POM14) A/D port configuration register (ADPC)
Port	Total: 26 (CMOS I/O: 23, CMOS input: 1, N-ch open drain I/O: 2)
Pull-up resistor	Total: 18

6.2.1 Port 0

Port 0 is a 1-bit 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 P05 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

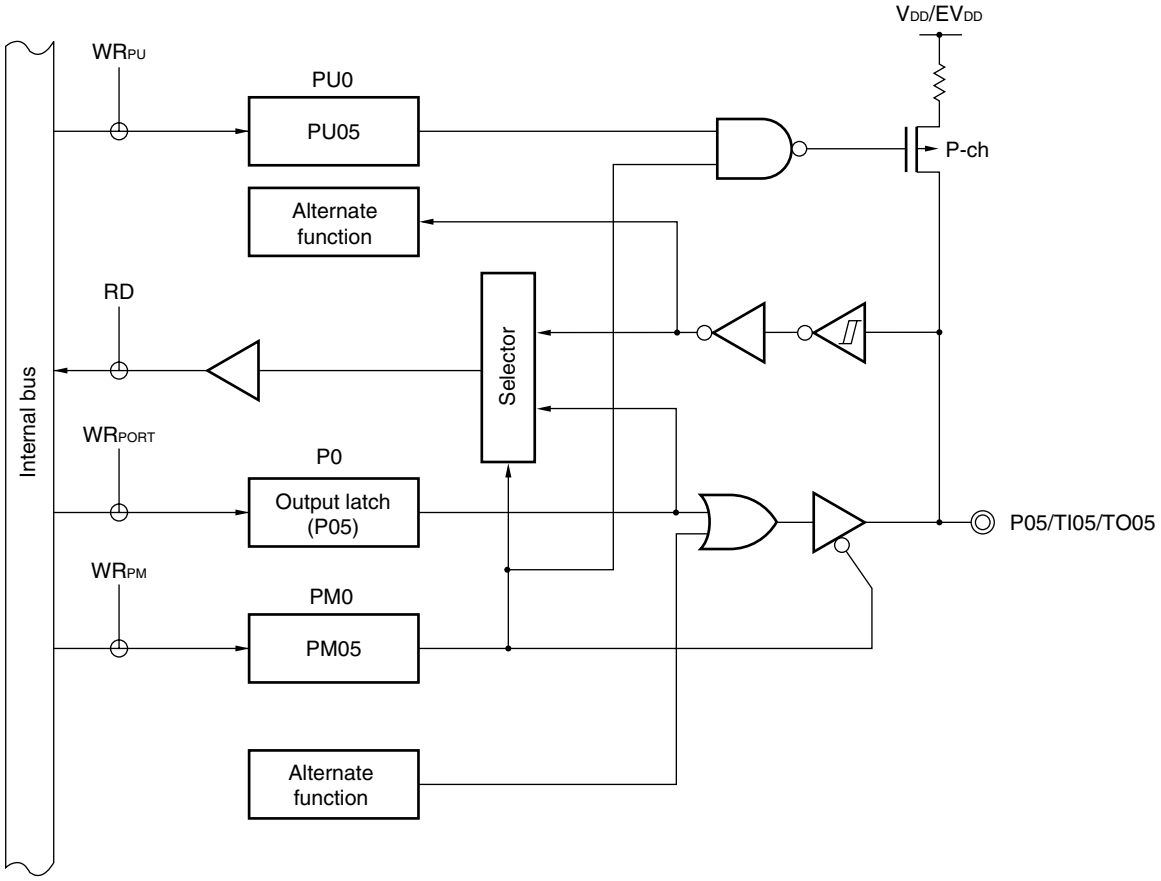
This port can also be used for timer I/O.

Reset signal generation sets port 0 to input mode.

Figure 6-2 shows block diagram of port 0.

Caution 1. To use P05/TI05/TO05 as a general-purpose port, set bit 5 (TO05) of timer output register 0 (TO0) and bit 5 (TOE05) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

Figure 6-2. Block Diagram of P05



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

6.2.2 Port 1

Port 1 is a 5-bit 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 P11, P13, P14, P16, 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).

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

Reset signal generation sets port 1 to input mode.

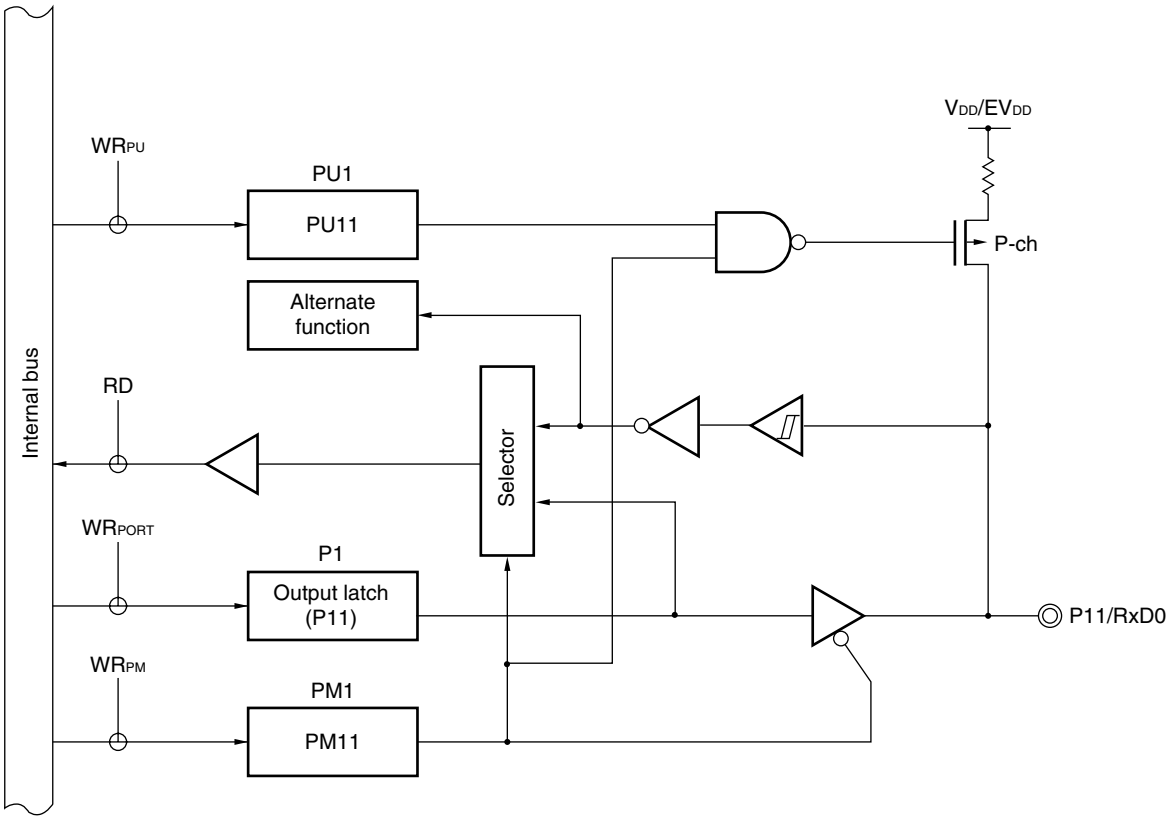
Figures 6-3 to 6-6 show block diagrams of port 1.

Cautions 1. To use P13/TxD3, P14/RxD3 as a general-purpose port, note the serial array unit setting. For details, refer to the following tables.

- **Table 11-7 Relationship Between Register Settings and Pins (Channel 2 of Unit 1: UART3 Transmission)**
- **Table 11-8 Relationship Between Register Settings and Pins (Channel 3 of Unit 1: UART3 Reception)**

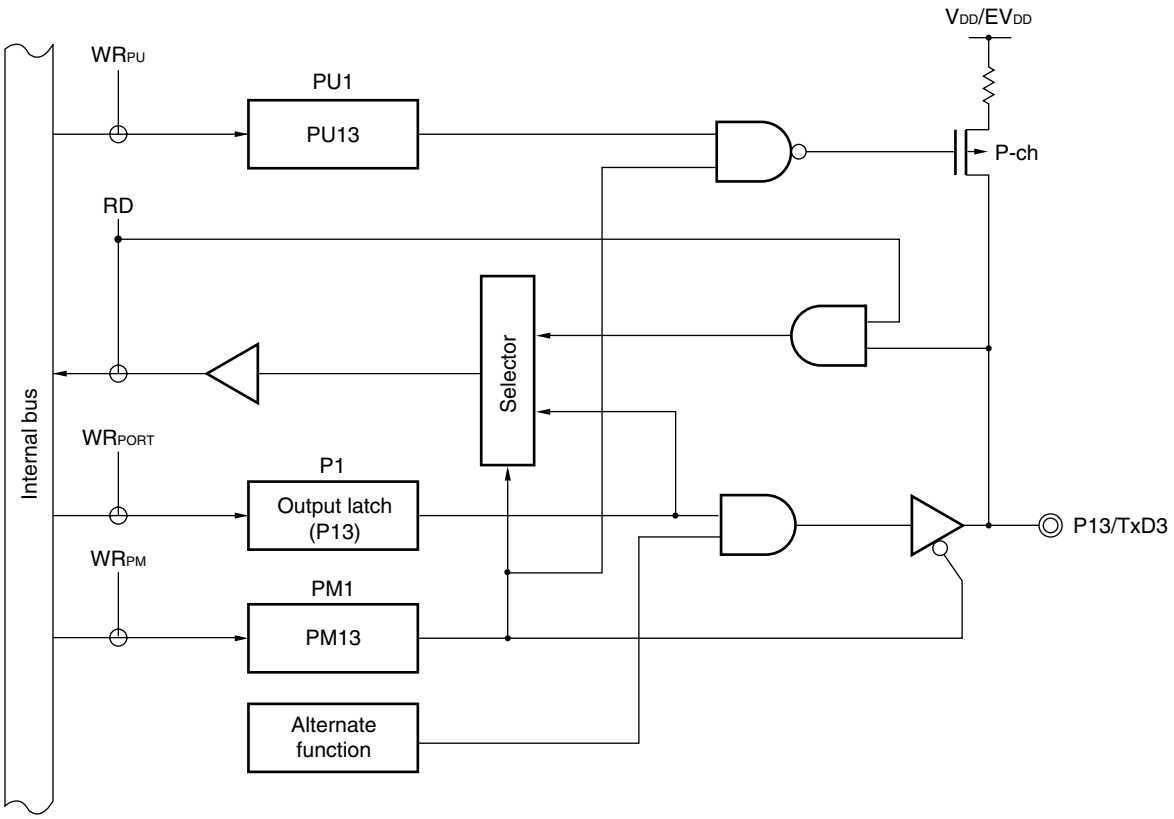
2. To use P16/TI01/TO01/INTP5 or P17/TI02/TO02 as a general-purpose port, set bits 1 and 2 (TO01, TO02) of timer output register 0 (TO0) and bits 1 and 2 (TOE01, TOE02) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.

Figure 6-3. Block Diagram of P11



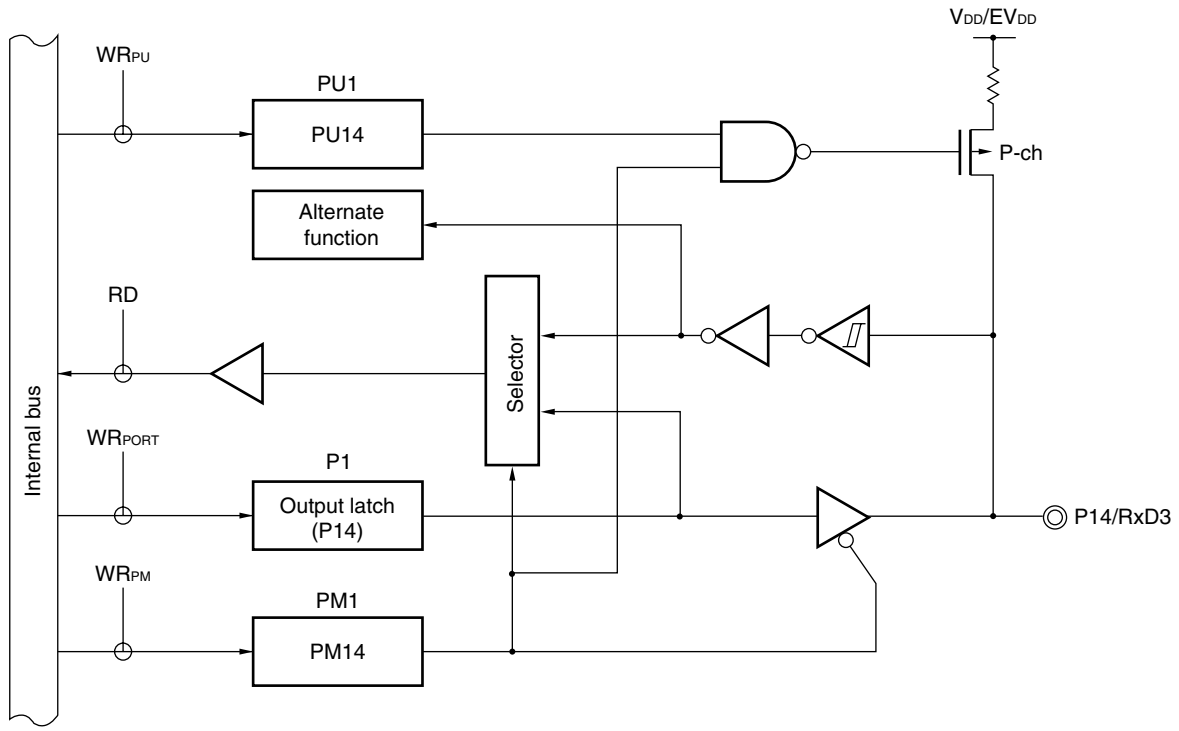
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 6-4. Block Diagram of P13



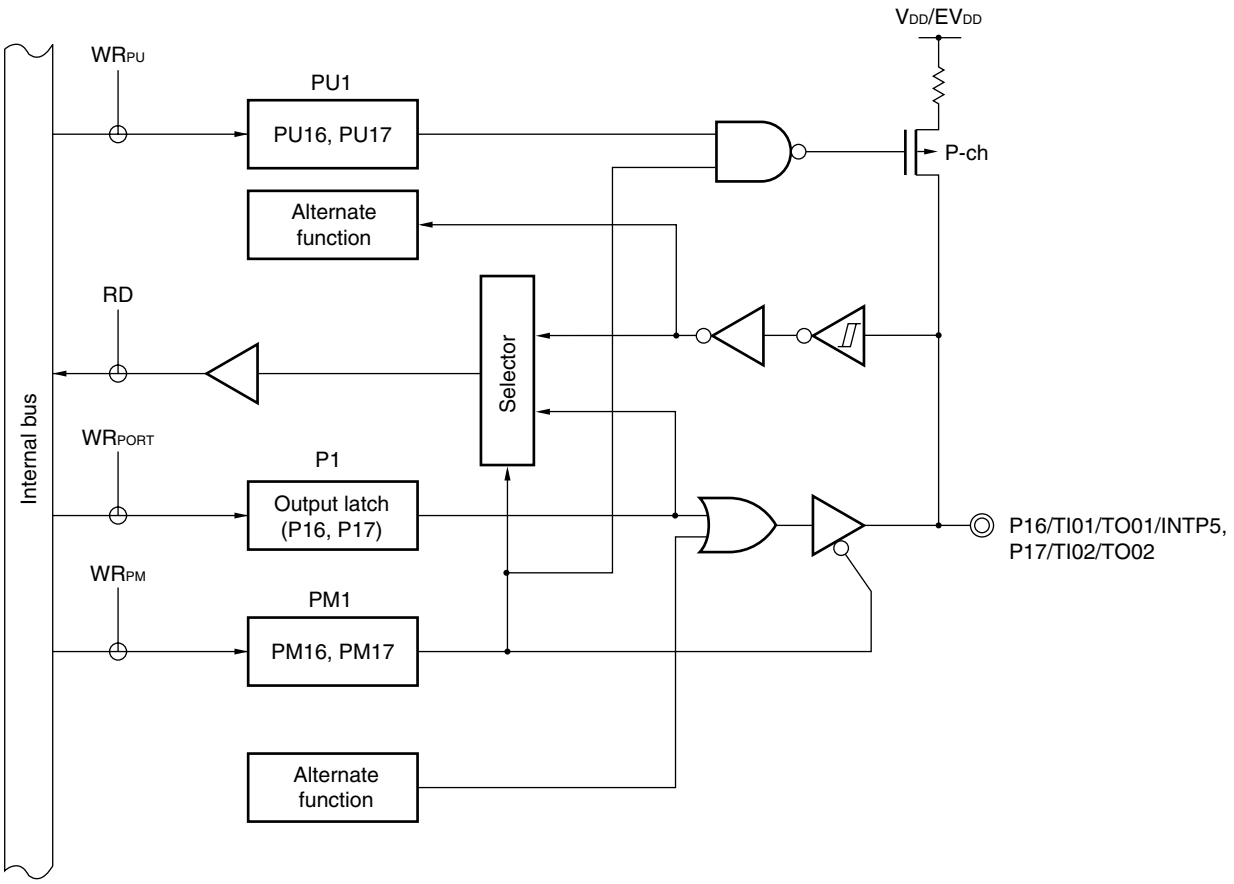
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 6-5. Block Diagram of P14



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

Figure 6-6. Block Diagram of P16 and P17



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

6.2.3 Port 2

Port 2 is a 2-bit I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units using port mode register 2 (PM2).

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

To use P26/ANI6 and 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 PM2. Use these pins starting from the lower bit.

To use P20/ANI6 and P27/ANI7 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM2.

To use P26/ANI6 and P27/ANI7 as analog input pins, set them in the analog input mode by using the ADPC and in the input mode by using PM2. Use these pins starting from the upper bit.

Table 6-4. Setting Functions of P26/ANI6, P27/ANI7 Pins

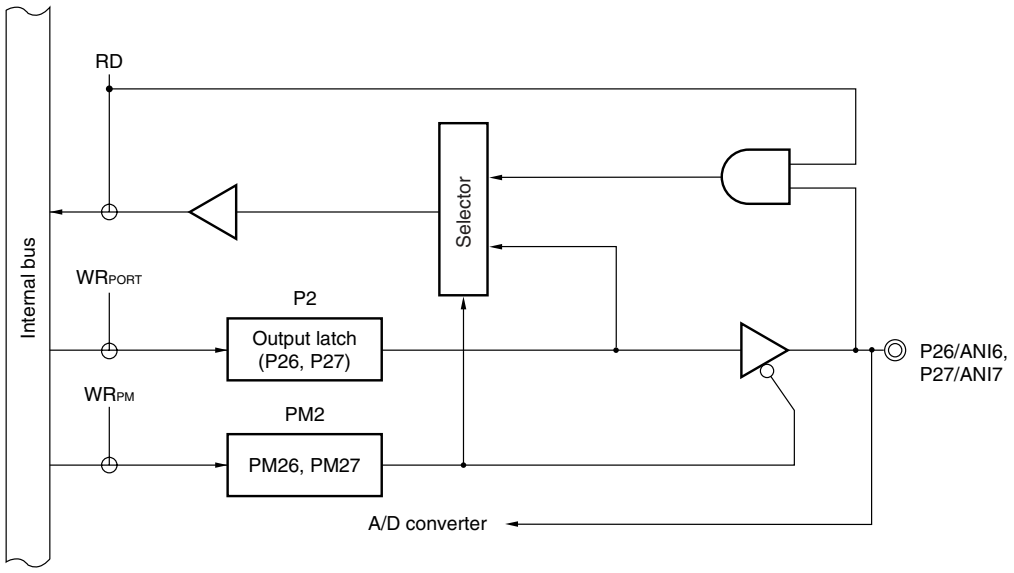
ADPC	PM2	ADS	P26/ANI6, P27/ANI7 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P26/ANI6 and P27/ANI7 are set in the digital input mode when the reset signal is generated.

Figure 6-7 shows a block diagram of port 2.

Caution See 3.2.22 AV_{REF} for the voltage to be applied to the AV_{REF} pin when using port 2 as a digital I/O.

Figure 6-7. Block Diagram of P26, P27



P2: Port register 2
PM2: Port mode register 2
RD: Read signal
WR××: Write signal

6.2.4 Port 3

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

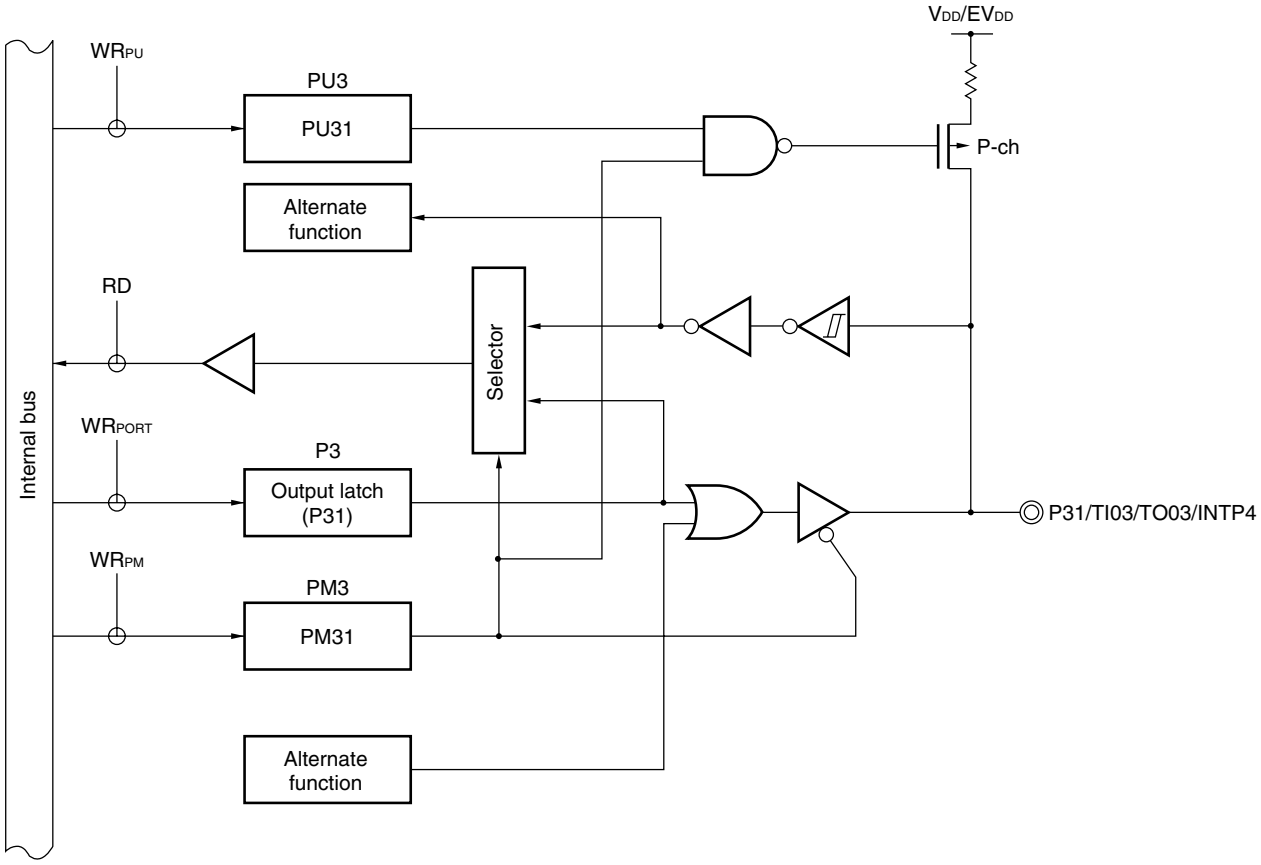
This port can also be used for external interrupt request input and timer I/O.

Reset signal generation sets port 3 to input mode.

Figure 6-8 shows block a diagram of port 3.

Cautions 1. To use P31/TI03/TO03/INTP4 as a general-purpose port, set bit 3 (TO03) of timer output register 0 (TO0) and bit 3 (TOE03) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

Figure 6-8. 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

6.2.5 Port 4

Port 4 is an 2-bit 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, P41 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) [Note].

This port can also be used for flash memory programmer/debugger data I/O and clock output.

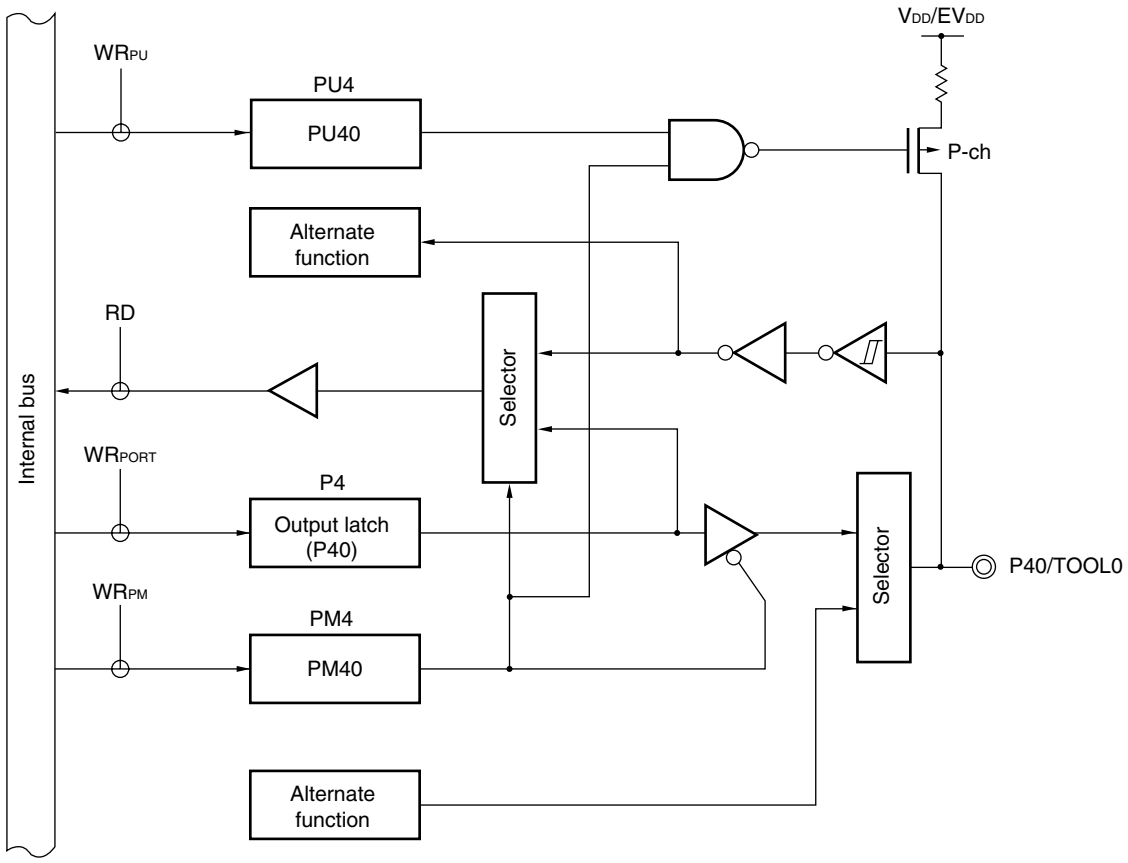
Reset signal generation sets port 4 to input mode.

Figures 6-9 and 6-10 show block diagrams of port 4.

Note When a tool is connected, the P40 and P41 pins cannot be connected to a pull-up resistor.

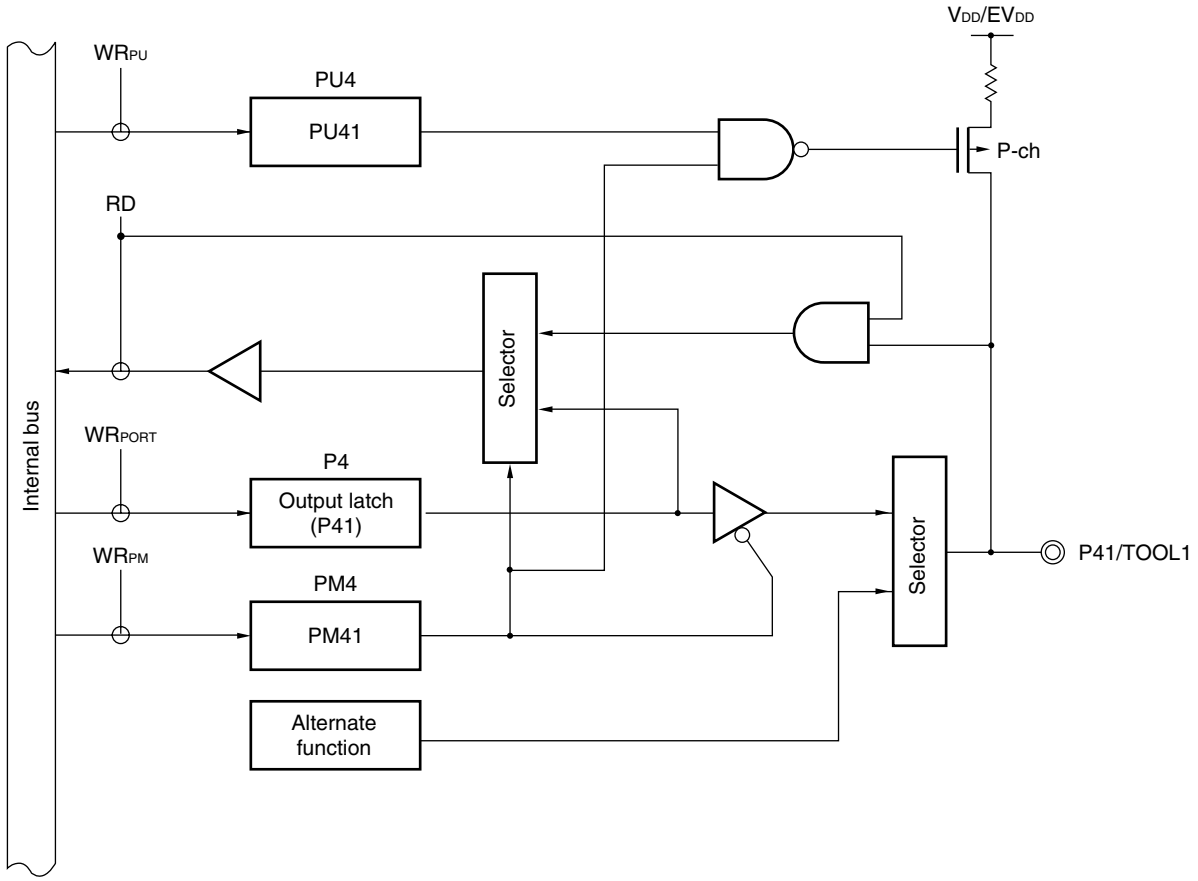
Caution When a tool is connected, the P40 pin cannot be used as a port pin.
When the on-chip debug function is used, P41 pin can be used as follows by the mode setting on the debugger.
1-line mode: can be used as a port (P41).
2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).

Figure 6-9. 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 6-10. Block Diagram of P41



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

6.2.6 Port 5

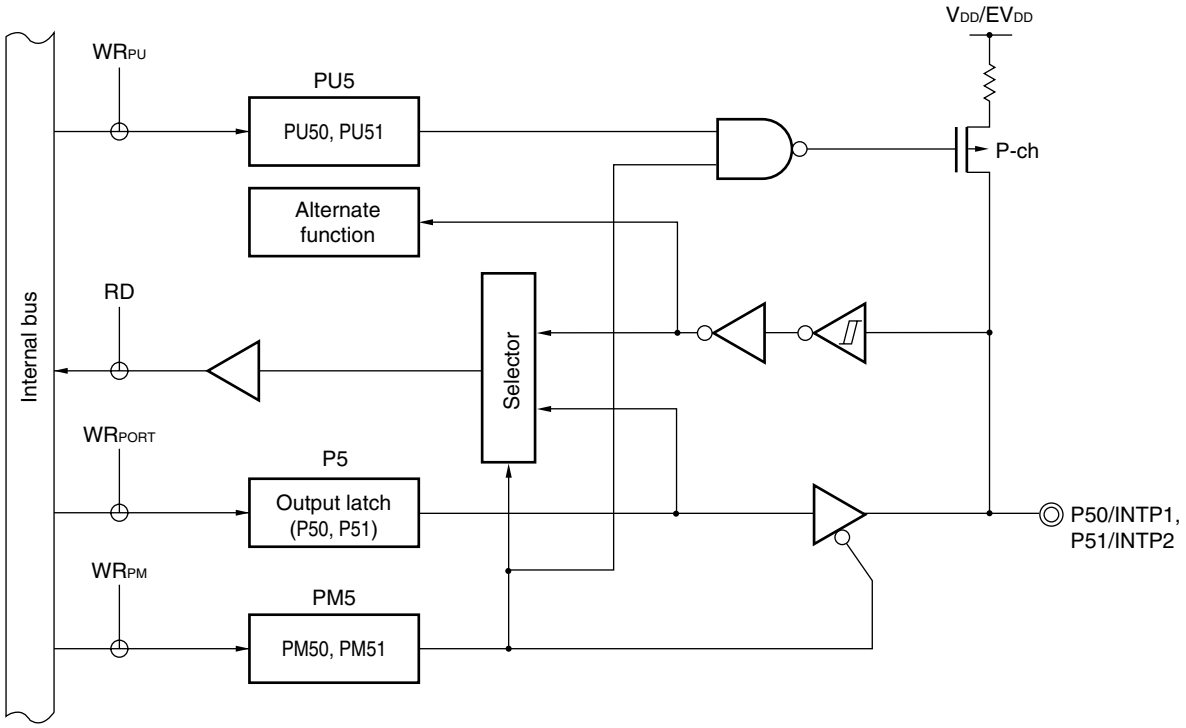
Port 5 is a 2-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode using port mode register 5 (PM5). When the P50 pin is used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

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

Reset signal generation sets port 5 to input mode.

Figure 6-11 show block diagram of port 5.

Figure 6-11. Block Diagram of P50, P51



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

6.2.7 Port 6

Port 6 is an 4-bit 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). When the P65, P67 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 6 (PU6).

The output of the P60 and P61 pins is N-ch open-drain output (6 V tolerance).

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

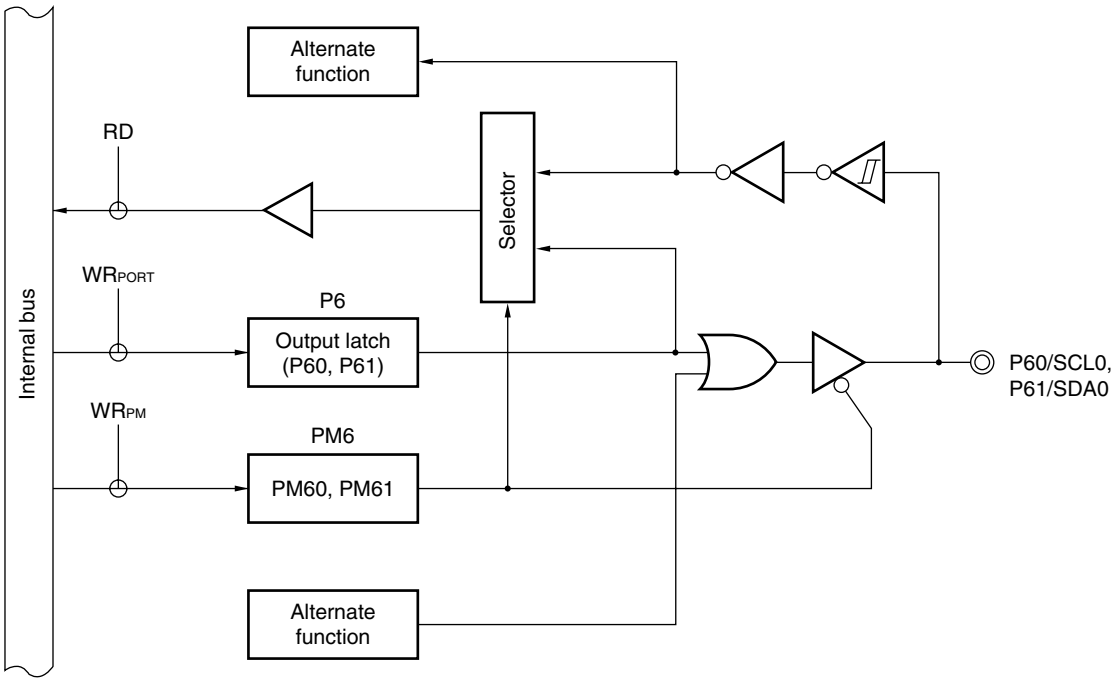
Reset signal generation sets port 6 to input mode.

Figures 6-12 and 6-13 show block diagrams of port 6.

Cautions 1. Stop the operation of serial interface IICA when using P60/SCL0 and P61/SDA0 as general-purpose ports.

2. To use P65/TI11/TO11 to P67/TI13/TO13 as a general-purpose port, set bits 1 and 3 (TO11 and TO13) of timer output register 1 (TO1) and bits 1 and 3 (TOE11 and TOE13) of timer output enable register 1 (TOE1) to "0", which is the same as their default status setting.

Figure 6-12. Block Diagram of P60 and P61



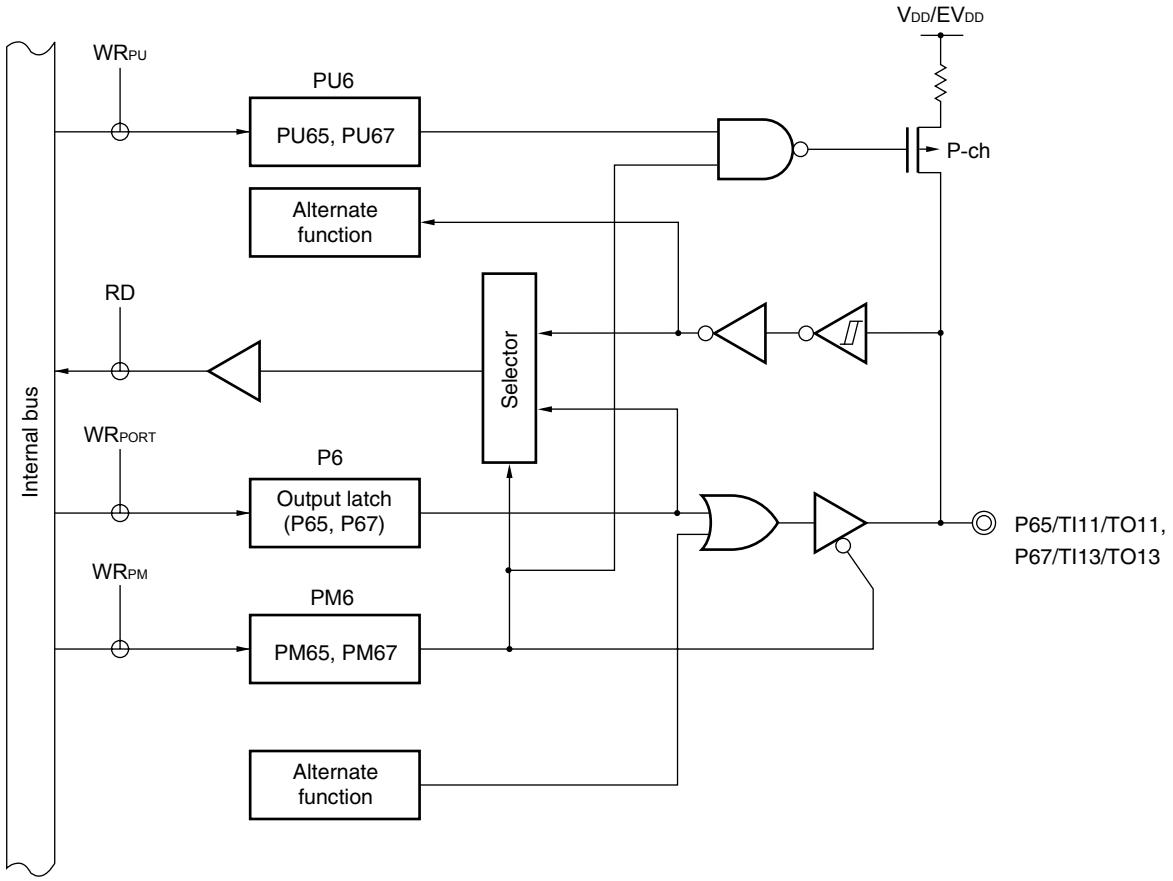
P6: Port register 6

PM6: Port mode register 6

RD: Read signal

WR××: Write signal

Figure 6-13. Block Diagram of P65, P67



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
RD: Read signal
WR××: Write signal

6.2.8 Port 12

P120 is a 1-bit 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 is 1-bit input ports.

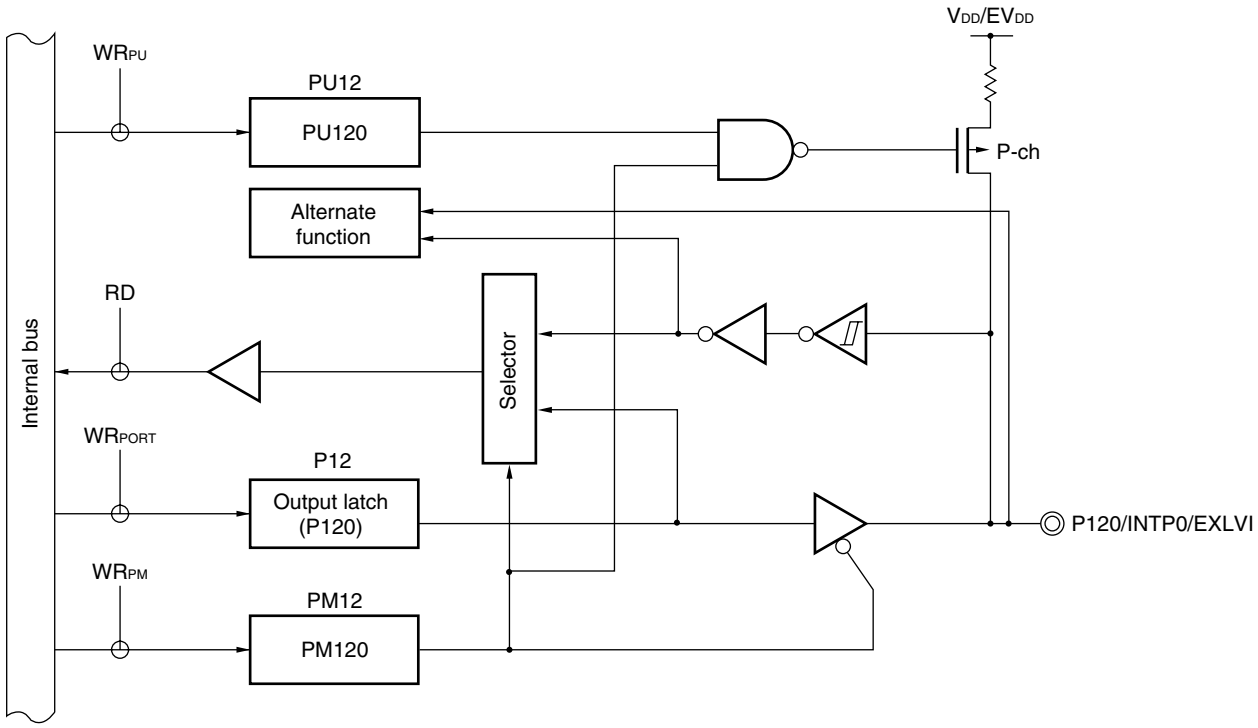
This port can also be used for external interrupt request input, potential input for external low-voltage detection, and connecting resonator for main system clock.

Reset signal generation sets port 12 to input mode.

Figures 6-14 and 6-15 show block diagrams of port 12.

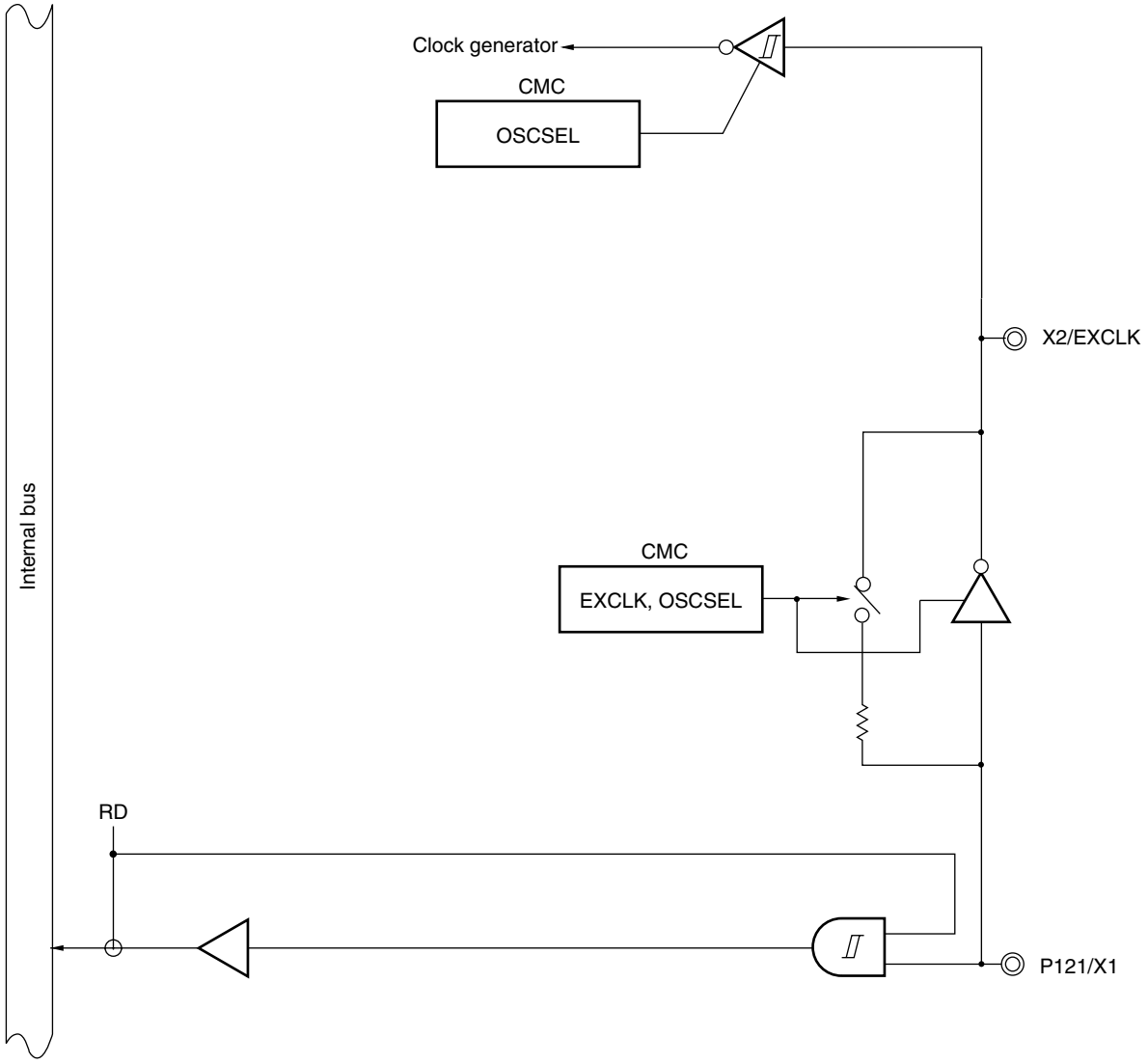
Caution The function setting on P121 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.

Figure 6-14. Block Diagram of P120



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

Figure 6-15. Block Diagram of P121



CMC: Clock operation mode control register
RD: Read signal

6.2.9 Port 14

Port 14 is a 3-bit 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 and P142 to P144 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

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

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

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

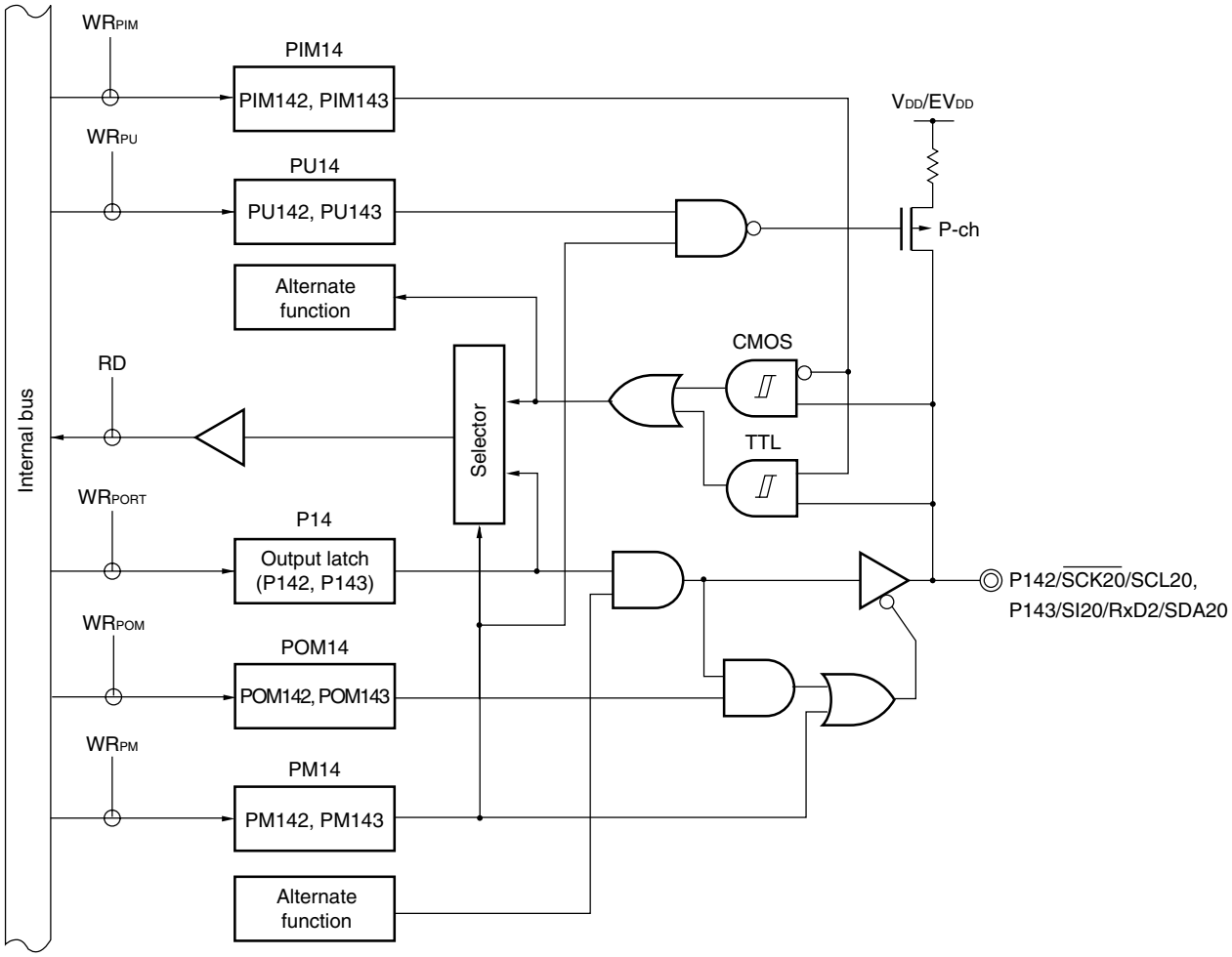
Reset signal generation sets port 14 to input mode.

Figures 6-16 and 6-17 show block diagrams of port 14.

Caution To use P142/$\overline{\text{SCK20}}$/SCL20, P143/SI20/RxD2/SDA20, or P144/SO20/TxD2 as a general-purpose port, note the serial array unit 1 setting. For details, refer to the following tables.

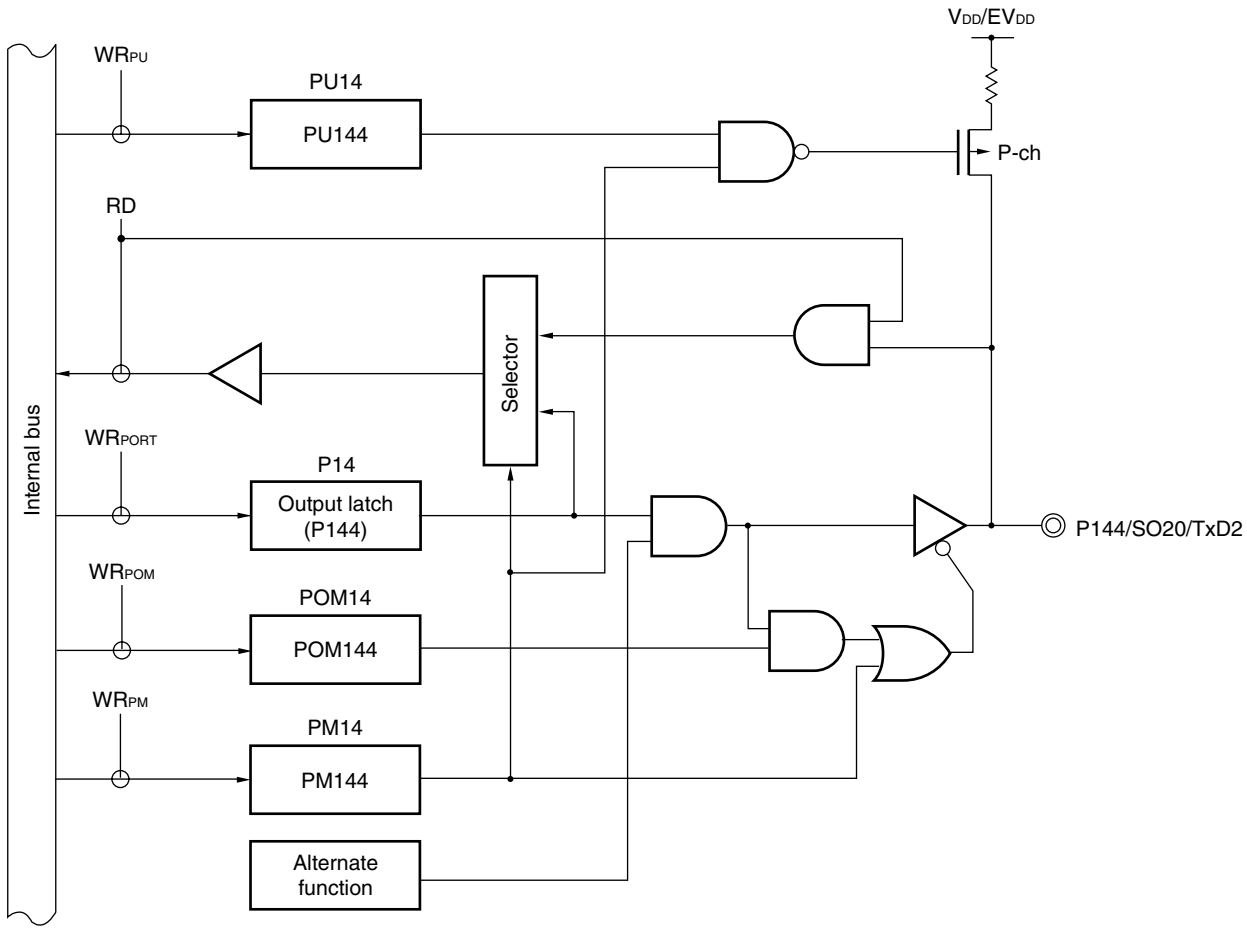
- **Table 11-5 Relationship Between Register Settings and Pins (Channel 0 of Unit 1: CSI20, UART2 Transmission, IIC20)**
- **Table 11-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 1: UART2 Reception)**

Figure 6-16. Block Diagram of P142 and P143



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
PIM14: Port input mode register 14
POM14: Port output mode register 14
RD: Read signal
WR××: Write signal

Figure 6-17. Block Diagram of P144



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
POM14: Port output mode register 14
RD: Read signal
WR××: Write signal

6.2.10 Port 15

Port 15 is a 4-bit I/O port with an output latch. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15).

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

To use P150/ANI8 to P153/ANI11 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 PM15. Use these pins starting from the lower bit.

To use P150/ANI8 to P153/ANI11 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM15.

Table 6-5. Setting Functions of P150/ANI8 to P153/ANI11 Pins

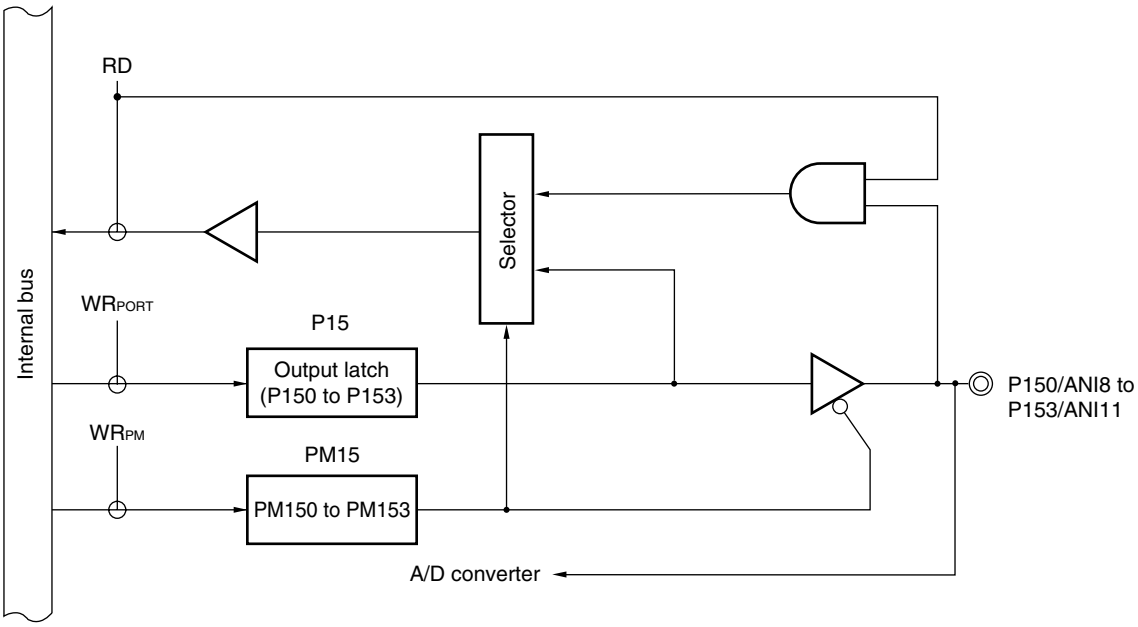
ADPC	PM15	ADS	P150/ANI8 to P153/ANI11 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P150/ANI8 to P153/ANI11 are set in the digital input mode when the reset signal is generated.

Figure 6-18 shows a block diagram of port 15.

Caution See 3.2.22 AV_{REF} for the voltage to be applied to the AV_{REF} pin when using port 15 as a digital I/O.

Figure 6-18. Block Diagram of P150 to P153



P15: Port register 15
PM15: Port mode register 15
RD: Read signal
WR××: Write signal

6.3 Registers Controlling Port Function

Port functions are controlled by the following six types of registers.

- Port mode registers (PM0 to PM7, PM11, PM12, PM14, PM15)
- Port registers (P0 to P7, P11, P12, P14, P15)
- Pull-up resistor option registers (PU0, PU1, PU3 to PU6, PU12, PU14)
- Port input mode register 14 (PIM14)
- Port output mode register 14 (POM14)
- A/D port configuration register (ADPC)

(1) Port mode registers (PM0 to PM7, PM11, PM12, PM14, PM15)

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 **6.5 Settings of Port Mode Register, and Output Latch When Using Alternate Function**.

Figure 6-19. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06	PM05	PM04	PM03	0	1	1	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15Note	PM14	PM13	PM12Note	PM11	PM10Note	FFF21H	FFH	R/W
PM2	PM27	PM26	0	0	0	0	0	0	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30Note	FFF23H	FFH	R/W
PM4	PM47	PM46	0	PM44	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	PM55	PM54	PM53	0	PM51	PM50	FFF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	0	0	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM11	1	1	1	1	1	1	1	PM110	FFF2BH	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	1	1	1	PM144	PM143	PM142	1	PM140	FFF2EH	FFH	R/W
PM15	1	1	1	1	PM153	PM152	PM151	PM150	FFF2FH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 7, 11, 12, 14, 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note Port mode register of an internal connection pin between the MCU and IO-Link transceiver (For the pin settings, see **Table 2-1 Settings of Internal Connection Pins**).

Cautions 1. Be sure to set bits 0, 1, and 7 of PM0, bits 2 to 7 of PM3, bits 6 and 7 of PM5, bits 1 to 7 of PM11, bits 1 to 7 of PM12, bits 1 and 5 to 7 of PM14, and bits 4 to 7 of PM15 to "1".

2. Be sure to clear bit 2 of PM0, bits 0 to 5 of PM2, bit 5 of PM4, bit 2 of PM5, and bits 2 and 3 of PM6 to "0".

3. Be sure to clear PM03, PM04, PM06, PM42 to PM44, PM46, PM47, PM53 to PM55, PM64, PM66, PM70 to PM77, PM110, and PM140 to 0 after a reset release (see 2.2 Initial Setting of Unused Internal Pins in MCU).

(2) Port registers (P0 to P6, P12, P14, P15)

These registers write the data that is output from the chip when data is output from 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 It is always 0 and never a pin level that is read out if a port is read during the input mode when P2 and P15 are set to function as an analog input for a A/D converter.

Figure 6-20. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	P05	0	0	0	0	0	FFF00H	00H (output latch)	R/W
P1	P17	P16	P15 Note 1	P14	P13	P12 Note 1	P11	P10 Note 1	FFF01H	00H (output latch)	R/W
P2	P27	P26	0	0	0	0	0	0	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	0	0	P31	0	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (output latch)	R/W
P5	0	0	0	0	0	0	P51	P50	FFF05H	00H (output latch)	R/W
P6	P67	0	P65	0	0	0	P61	P60	FFF06H	00H (output latch)	R/W
P12	0	0	0	Undefined	Undefined	Undefined	P121	P120	FFF0CH	Undefined	R/W Note 2
P14	0	0	0	P144	P143	P142	0	0	FFF0EH	00H (output latch)	R/W
P15	0	0	0	0	P153	P152	P151	P150	FFF0FH	00H (output latch)	R/W

Pmn	m = 0 to 6, 12, 14, 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. Port register of an internal connection pin between the MCU and IO-Link transceiver (For the pin settings, see **Table 2-1 Settings of Internal Connection Pins**).

2. P121 is read-only.

(3) Pull-up resistor option registers (PU0, PU1, PU3 to PU6, PU12, PU14)

These registers specify whether the on-chip pull-up resistors of P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P65, P67, P120, or P142 to P144 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 of the pins to which the use of an on-chip pull-up resistor has been specified in PU0, PU1, PU3 to PU6, PU12 and PU14. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of PU0, PU1, PU3 to PU6, PU12 and PU14.

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

Reset signal generation clears these registers to 00H.

Figure 6-21. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	PU05	0	0	0	0	0	F0030H	00H	R/W
PU1	PU17	PU16	PU15 Note	PU14	PU13	0	PU11	0	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	0	F0033H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	00H	R/W
PU5	0	0	0	0	0	0	PU51	PU50	F0035H	00H	R/W
PU6	PU67	0	PU65	0	0	0	0	0	F0036H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	0	0	0	PU144	PU143	PU142	0	0	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 6, 12, 14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Note Port output mode register of an internal connection pin between the MCU and IO-Link transceiver (For the pin settings, see **Table 2-1 Settings of Internal Connection Pins**).

(4) Port input mode register 14 (PIM14)

This register set the input buffer of P142 or P143 in 1-bit units.

TTL input buffer can be selected during serial communication with an external device of the different potential by PIM142 and PIM143 bit.

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

Reset signal generation clears this register to 00H.

Figure 6-22. Format of Port Input Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM14	0	0	0	0	PIM143	PIM142	0	0	F004EH	00H	R/W

PIM14n	P14n pin input buffer selection (m = 14; n = 2, 3)
0	Normal input buffer
1	TTL input buffer

(5) Port output mode register 14 (POM14)

This register set the output mode of P142 to P144 in 1-bit units.

N-ch open drain output (V_{DD} tolerance) mode can be selected during serial communication with an external device of the different potential.

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

Reset signal generation clears this register to 00H.

Figure 6-23. Format of Port Output Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM14	0	0	0	POM144	POM143	POM142	0	0	F005EH	00H	R/W

POM14n	P14n pin output mode selection (n = 2 to 4)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

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

This register switches the P26/ANI6, P27/ANI7, and P150/ANI8 to P153/ANI11 pins to digital I/O of port or analog input of A/D converter.

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

Reset signal generation sets this register to 10H.

Figure 6-24. Format of A/D Port Configuration Register (ADPC)

Address: F0017H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching					
					Port 15				Port 2	
					ANI11/ P153	ANI10/ P152	ANI9/ P151	ANI8/ P150	ANI7/ P27	ANI6/ P26
0	0	1	1	0	A	A	A	A	A	A
0	0	1	1	1	A	A	A	A	A	D
0	1	0	0	0	A	A	A	A	D	D
0	1	0	0	1	A	A	A	D	D	D
0	1	0	1	0	A	A	D	D	D	D
0	1	0	1	1	A	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D
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 and 15 (PM2, PM15).

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

3. Be sure to first set the ADCEN bit of peripheral enable register 0 (PER0) to 1 when setting up the ADPC register. If ADCEN = 0, writing to the ADPC register is ignored and specified values are returned to the initial values.

4. P26/ANI6, P27/ANI7, and P150/ANI8 to P153/ANI11 are set as analog inputs in the order of P153/ANI11, …, P150/ANI8, P27/ANI7, …, P20/ANI0 by the A/D port configuration register (ADPC). When using P26/ANI6, P27/ANI7, and P150/ANI8 to P153/ANI11 as analog inputs, start designing from P153/ANI11.

6.4 Port Function Operations

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

6.4.1 Writing to I/O port

(1) Output mode

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

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

6.4.2 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.

6.4.3 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.
The data of the output latch is cleared when a reset signal is generated.

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

When parts of port 14 operate with V_{DD} = 4.0 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 port input mode register (PIM14).

Moreover, regarding outputs, different potentials can be supported by switching the output buffer to the N-ch open drain (V_{DD} withstand voltage) by the port output mode register (POM14).

(1) Setting procedure when using I/O pins of UART2 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 UART2: P143
In case of CSI20: P142, P143

<3> Set the corresponding bit of the PIM14 register 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 UART2: P144
In case of CSI20: P142, P144

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

<4> Set the corresponding bit of the POM14 register to 1 to set the N-ch open drain output (V_{DD} withstand voltage) mode.

<5> Set the output mode by manipulating the PM14 register.
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 IIC20: P142, P143

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

<4> Set the corresponding bit of the POM14 register to 1 to set the N-ch open drain output (V_{DD} withstand voltage) mode.

<5> Set the corresponding bit of the PM14 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.

6.5 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 6-6.

Table 6-6. Settings of Port Mode Register, and Output Latch When Using Alternate Function

Pin Name	Alternate Function		PM××	P××	Pin Name	Alternate Function		PM××	P××
	Function Name	I/O				Function Name	I/O		
P05	TI05	Input	1	×	P60	SCL0	I/O	0	0
	TO05	Output	0	0	P61	SDA0	I/O	0	0
P11	RxD0	Input	1	×	P65	TI11	Input	1	×
P13	TxD3	Output	0	1		TO11	Output	0	0
P14	RxD3	Input	1	×	P67	TI13	Input	1	×
P16	TI01	Input	1	×		TO13	Output	0	0
	TO01	Output	0	0	P120	INTP0	Input	1	×
	INTP5	Input	1	×		EXLVI	Input	1	×
P17	TI02	Input	1	×	P142	$\overline{\text{SCK20}}$	Input	1	×
	TO02	Output	0	0			Output	0	1
P26, P27 Note	ANI6, ANI7 Note	Input	1	×		SCL20	I/O	0	1
P31	TI03	Input	1	×	P143	SI20	Input	1	×
	TO03	Output	0	0		RxD2	Input	1	×
	INTP4	Input	1	×		SDA20	I/O	0	1
P40	TOOL0	I/O	×	×	P144	SO20	Output	0	1
P41	TOOL1	Output	×	×		TxD2	Output	0	1
P50	INTP1	Input	1	×	P150 to P153 Note	ANI8 to ANI11 Note	Input	1	×
P51	INTP2	Input	1	×					

Note The functions of the ANI6/P26, ANI7/P27, and ANI8/P150 to ANI11/P153 pins can be selected by using the A/D port configuration register (ADPC), the analog input channel specification register (ADS), and port mode registers 2, 15 (PM2, PM15).

Table 6-7. Setting Functions of ANI6/P26, ANI7/P27, and ANI8/P150 to ANI11/P153 Pins

ADPC	PM2, PM15	ADS	ANI6/P26, ANI7/P27, ANI8/P150 to ANI11/P153 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.	

Remark ×: don't care
PM××: Port mode register
P××: Port output latch

6.6 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 μPD78F8040, 78F8041, 78F8042, 78F8043.

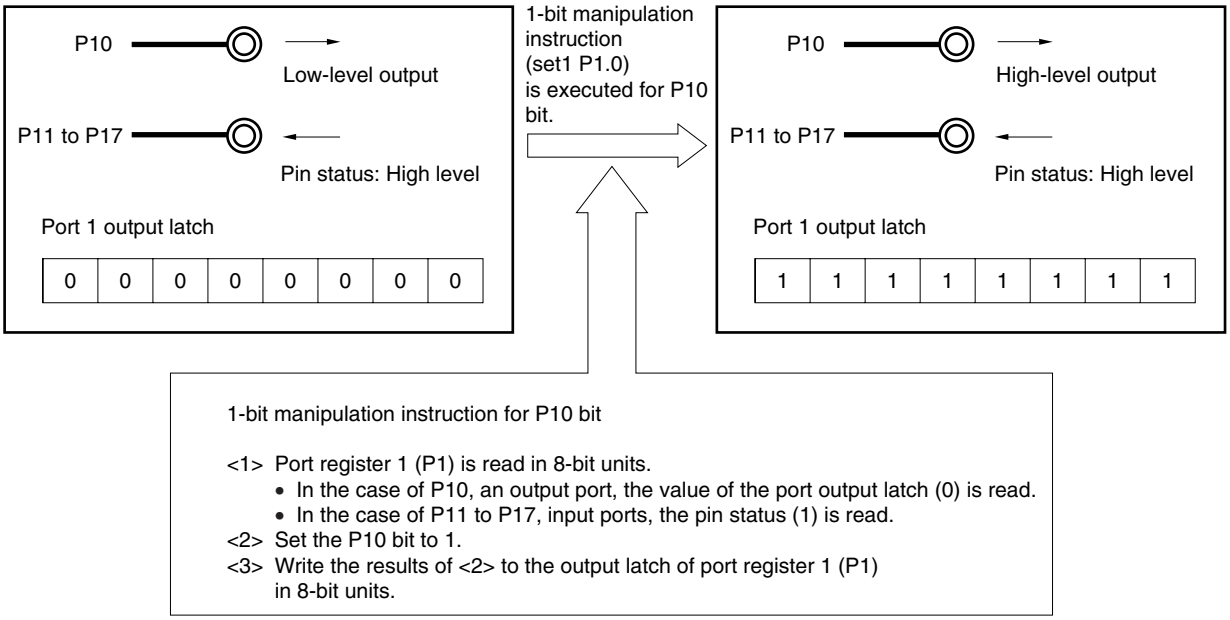
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

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 6-25. Bit Manipulation Instruction (P10)



CHAPTER 7 CLOCK GENERATOR

7.1 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 = 2 to 20 MHz by connecting a resonator to X1 and X2.
Oscillation can be stopped by executing the STOP instruction or setting of MSTOP (bit 7 of the clock operation status control register (CSC)).

<2> Internal high-speed oscillator[Note]

This circuit oscillates clocks of f_{IH} = 1, 8 MHz (TYP.). After a reset release, the CPU always starts operating with this internal high-speed oscillation clock. Oscillation can be stopped by executing the STOP instruction or setting HIOSTOP (bit 0 of CSC).

<3> 20 MHz internal high-speed oscillation clock oscillator[Note]

This circuit oscillates a clock of f_{IH20} = 20 MHz (TYP.). Oscillation can be started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1. Oscillation can be stopped by setting DSCON to 0.

Note To use the 1, 8 or 20 MHz internal high-speed oscillation clock, use the option byte to set the frequency in advance (for details, see **CHAPTER 21 OPTION BYTE**). Also, the internal high-speed oscillator automatically starts oscillating after reset release. To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1.

An external main system clock (f_{EX} = 2 to 20 MHz) can also be supplied from the EXCLK/X2 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of MSTOP.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or internal high-speed oscillation clock can be selected by setting of MCM0 (bit 4 of the system clock control register (CKC)).

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency

(2) Internal low-speed oscillation clock (clock dedicated to watchdog timer)

- **Internal low-speed oscillator**

  This circuit oscillates a clock of f_{IL} = 30 kHz (TYP.).

  The internal low-speed oscillation clock cannot be used as the CPU clock. The only hardware that operates with the internal low-speed oscillation clock is the watchdog timer.

  Oscillation is stopped when the watchdog timer stops.

Remarks 1. f_{IL}: Internal low-speed oscillation clock frequency

2. The watchdog timer stops in the following cases.

- When bit 4 (WDTON) of an option byte (000C0H) = 0
- If the HALT or STOP instruction is executed when bit 4 (WDTON) of an option byte (000C0H) = 1 and bit 0 (WDSTBYON) = 0

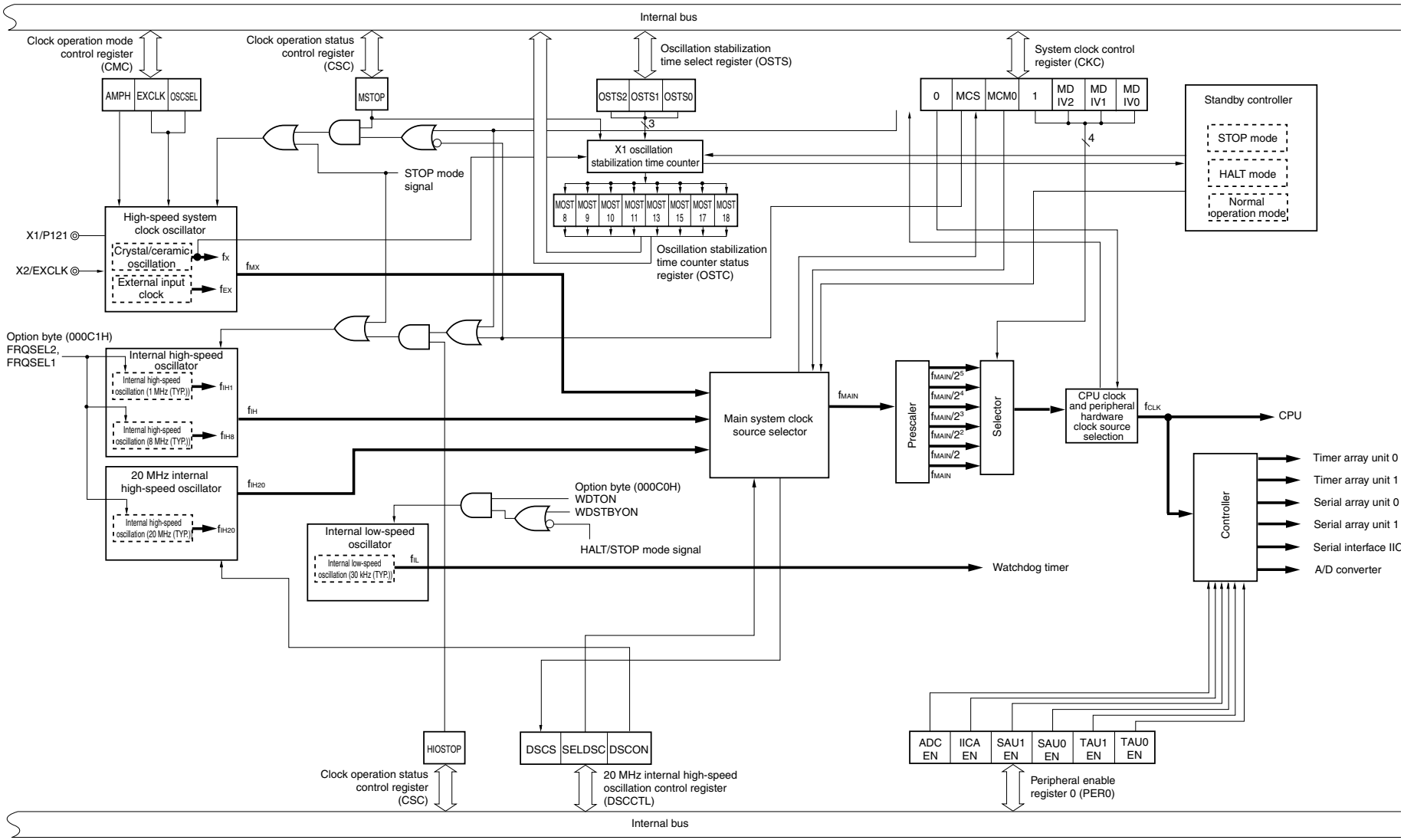
7.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 7-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) System clock control register (CKC) 20 MHz internal high-speed oscillation control register (DSCCTL) Peripheral enable register 0 (PER0) Operation speed mode control register (OSMC)
Oscillators	X1 oscillator Internal high-speed oscillator Internal low-speed oscillator

Figure 7-1. Block Diagram of Clock Generator



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

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Internal low-speed oscillation clock frequency

7.3 Registers Controlling Clock Generator

The following eight registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- System clock control register (CKC)
- 20 MHz internal high-speed oscillation control register (DSCCTL)
- Peripheral enable register 0 (PER0)
- Operation speed mode control register (OSMC)

(1) Clock operation mode control register (CMC)

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

CMC 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 7-2. Format of Clock Operation Mode Control Register (CMC)

Address: FFFA0H After reset: 00H R/W

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

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

AMPH	Control of X1 clock oscillation frequency
0	2 MHz ≤ fx ≤ 10 MHz
1	10 MHz < fx ≤ 20 MHz

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

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

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

4. When CMC 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.

Remark fx: X1 clock oscillation frequency

(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 09H.

Figure 7-3. Format of System Clock Control Register (CKC)

Address: FFFA4H After reset: 09H R/W[Note 1]

Symbol	7	6	<5>	<4>	3	2	1	0
CKC	0	0	MCS	MCM0	1	MDIV2	MDIV1	MDIV0

MCS	Status of Main system clock (f_{MAIN})
0	Internal high-speed oscillation clock (f_{IH}) or 20 MHz internal high-speed oscillation clock (f_{IH20})
1	High-speed system clock (f_{MX})

MCM0	Main system clock (f_{MAIN}) operation control
0	Selects the internal high-speed oscillation clock (f_{IH}) or 20 MHz internal high-speed oscillation clock (f_{IH20}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

MDIV2	MDIV1	MDIV0	Selection of CPU/peripheral hardware clock (f_{CLK})
0	0	0	f_{MAIN}
0	0	1	$f_{MAIN}/2$ (This is the default setting if MCM0 = 0.)
0	1	0	$f_{MAIN}/2^2$
0	1	1	$f_{MAIN}/2^3$ **Note 2**
1	0	0	$f_{MAIN}/2^4$ **Note 2**
1	0	1	$f_{MAIN}/2^5$ **Notes 2, 3**
Other than above			Setting prohibited

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

2. Setting is prohibited if the 1 MHz Internal high-speed oscillation clock frequency (f_{IH1}) is selected as the main system clock (f_{MAIN}).

3. Setting is prohibited if the high-speed system clock (f_{MX}) is selected as the main system clock (f_{MAIN}) and if $f_{MX} < 4$ MHz.

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

Remarks 1. f_{IH}: Internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz Internal high-speed oscillation clock frequency
f_{IH1}: 1 MHz Internal high-speed oscillation clock frequency
f_{MX}: High-speed system clock frequency
2. ×: don't care

Cautions 1. Be sure to set bit 3 to 1.
2. The clock set by the MCM0 and MDIV2 to MDIV0 bits is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.

The fastest instruction can be executed in 1 clock of the CPU clock in the μPD78F8040, 78F8041, 78F8042, 78F8043. Therefore, the relationship between the CPU clock (f_{CLK}) and the minimum instruction execution time is as shown in Table 7-2.

Table 7-2. Relationship Between CPU Clock and Minimum Instruction Execution Time

CPU Clock (Value set by the MDIV2 to MDIV0 bits)	Minimum Instruction Execution Time: $1/f_{CLK}$			
	Main System Clock (CSS = 0)			
	High-Speed System Clock (MCM0 = 1)		Internal High-Speed Oscillation Clock (MCM0 = 0)	
	At 10 MHz Operation	At 20 MHz Operation	At 8 MHz (TYP.) Operation	At 20 MHz (TYP.) Operation
f_{MAIN}	0.1 μs	0.05 μs	0.125 μs (TYP.)	0.05 μs (TYP.)
$f_{MAIN}/2$	0.2 μs	0.1 μs	0.25 μs (TYP.) (default)	0.1 μs (TYP.)
$f_{MAIN}/2^2$	0.4 μs	0.2 μs	0.5 μs (TYP.)	0.2 μs (TYP.)
$f_{MAIN}/2^3$	0.8 μs	0.4 μs	1.0 μs (TYP.)	0.4 μs (TYP.)
$f_{MAIN}/2^4$	1.6 μs	0.8 μs	2.0 μs (TYP.)	0.8 μs (TYP.)
$f_{MAIN}/2^5$	3.2 μs	1.6 μs	4.0 μs (TYP.)	1.6 μs (TYP.)

Remark f_{MAIN}: Main system clock frequency (f_{IH}, f_{IH20}, or f_{MX})

(3) Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock and internal high-speed oscillation clock (except the 20 MHz internal high-speed oscillation clock and internal low-speed oscillation clock).

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

Reset signal generation sets this register to C0H.

Figure 7-4. 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	0	0	0	0	0	0	HIOSTOP

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

HIOSTOP	Internal high-speed oscillation clock operation control
0	Internal high-speed oscillator operating
1	Internal high-speed oscillator stopped[Note]

Note The 1 MHz or 8 MHz (TYP.) internal high-speed oscillation clock stops. Stopping the internal high-speed oscillator (HIOSTOP = 1) is prohibited while the 20 MHz internal high-speed oscillation clock is operating (DSCON = 1). Stop the 20 MHz internal high-speed oscillation clock by using the 20 MHz internal high-speed oscillation control register (DSCCTL) and not the HIOSTOP bit.

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

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

3. Do not stop the clock selected for the CPU/peripheral hardware clock (f_{CLK}) with the CSC register.

4. 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 7-3.

5. 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.

Table 7-3. 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. (MCS = 0)	MSTOP = 1
Internal high-speed oscillation clock	CPU and peripheral hardware clocks operate with a clock other than the internal high-speed oscillator clock. (MCS = 1)	HIOSTOP = 1

(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 internal high-speed oscillation clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock with the X1 clock oscillating.

The OSTC 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 7-5. 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

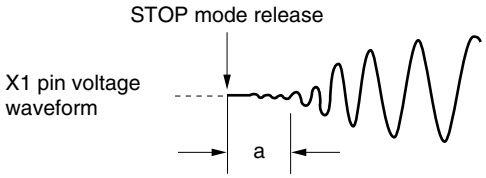
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx 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 MOST8 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 internal high-speed oscillation clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the internal high-speed oscillation 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 OSTS is set to OSTC 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 fx: X1 clock oscillation frequency

(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 OSTS after the STOP mode is released.

When the internal high-speed oscillation 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.

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

Reset signal generation sets OSTS to 07H.

Figure 7-6. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				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 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 before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

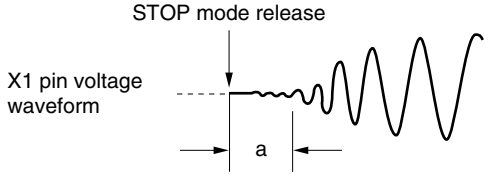
4. Do not change the value of the OSTS during the X1 clock oscillation stabilization time.

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

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

- **If the X1 clock starts oscillation while the internal high-speed oscillation clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the internal high-speed oscillation 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 OSTS is set to OSTC 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

(6) 20 MHz internal high-speed oscillation control register (DSCCTL)

This register controls the 20 MHz internal high-speed oscillation clock (DSC) function.

This register can be used to control oscillation of the 20 MHz internal high-speed oscillation clock (f_{IH20}) and select the 20 MHz internal high-speed oscillation clock (f_{IH20}) as the CPU/peripheral hardware clock.

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

Reset signal generation clears this register to 00H.

Figure 7-7. Format of 20 MHz Internal High-Speed Oscillation Control Register (DSCCTL)

Address: F00F6H After reset: 00H R/W[Note]

Symbol	7	6	5	4	<3>	<2>	1	<0>
DSCCTL	0	0	0	0	DSCS	SELDSC	0	DSCON

DSCS	20 MHz internal high-speed oscillation supply status flag
0	Not supplied
1	Supplied (The CPU/peripheral hardware clock (f_{CLK}) operates on the 20 MHz internal high-speed oscillation clock.)

SELDSC	Selection of 20 MHz internal high-speed oscillation for CPU/peripheral hardware clock (f_{CLK})
0	Does not select 20 MHz internal high-speed oscillation (clock selected by the system clock control register (CKC) is supplied to f_{CLK})
1	Selects 20 MHz internal high-speed oscillation (20 MHz internal high-speed oscillation is supplied to f_{CLK})

DSCON	20 MHz internal high-speed oscillation clock (f_{IH20}) operation enable/disable
0	Stopped
1	Operated

Note Bit 3 is read-only.

Cautions 1. Set SELDSC when 100 μs have elapsed after having set DSCON.

2. The internal high-speed oscillator must be operated (HIOSTOP = 0) when DSCON = 1.

3. If 1 MHz internal oscillation is selected by using the option byte, 20 MHz internal high-speed oscillation cannot be used. Do not set (1) the DSCON bit.

(7) 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.

- A/D converter
- Serial interface IICA
- Serial array unit SAU
- Timer array unit TAUS

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

Reset signal generation clears this register to 00H.

Figure 7-8. 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	0	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

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

IICAEN	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	Supplies input clock. • SFR used by the serial interface IICA can be read and written.

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

Caution Be sure to clear bits 6 and 7 to 0.

Figure 7-8. 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	0	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

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

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

Caution Be sure to clear bits 6 and 7 to 0.

(8) Operation speed mode control register (OSMC)

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

The FLPC and FSEL bits can be used to control the step-up circuit of the flash memory for high-speed operation.

If the microcontroller operates on a system clock of 10 MHz or more, set this register to 01B.

If the microcontroller operates at low speed on a system clock of 10 MHz or less, power consumption can be reduced, because the voltage booster can be stopped by setting this register to its initial value, 00B. Furthermore, when CPU operates with the system clock of 1 MHz, the power consumption can be further reduced by setting the FLPC bit to 1.

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

Reset signal generation clears this register to 00H.

Figure 7-9. 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	0	0	0	0	0	0	FLPC	FSEL

FLPC	FSEL	f_{CLK} frequency selection
0	0	Operates at a frequency of 10 MHz or less (default).
0	1	Operates at a frequency higher than 10 MHz.
1	0	Operates at a frequency of 1 MHz.
1	1	Setting prohibited

Cautions 1. Write "1" to FSEL before the following two operations.

- **Changing the clock prior to dividing f_{CLK} to a clock other than f_{IH}.**
- **Operating the DMA controller.**

2. The CPU waits (140.5 clock (f_{CLK})) when "1" is written to the FSEL.
Interrupt requests issued during a wait will be suspended.
However, counting the oscillation stabilization time of f_X can continue even while the CPU is waiting.

3. To increase f_{CLK} to 10 MHz or higher, set the FSEL bit to "1", then change f_{CLK} after three or more clocks have elapsed.

4. To set FSEL = 0 when the clock is operating at 10 MHz or less.

5. Set FSEL = 0 to shift to STOP mode.

6. If the FLPC bit is set to a frequency of 1 MHz or less and then set (1), it cannot be cleared (0) or set to a frequency of more than 1 MHz.

7.4 System Clock Oscillator

7.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (2 to 20 MHz) connected to the X1 and X2 pins.

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

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

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

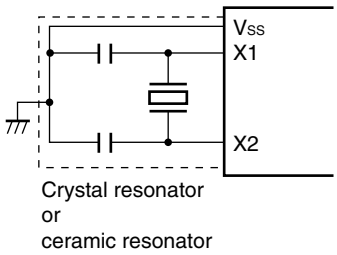
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

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

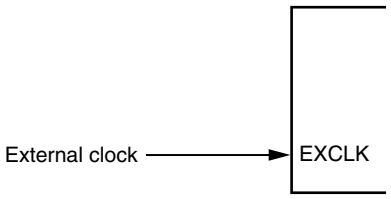
Figure 7-10 shows an example of the external circuit of the X1 oscillator.

Figure 7-10. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



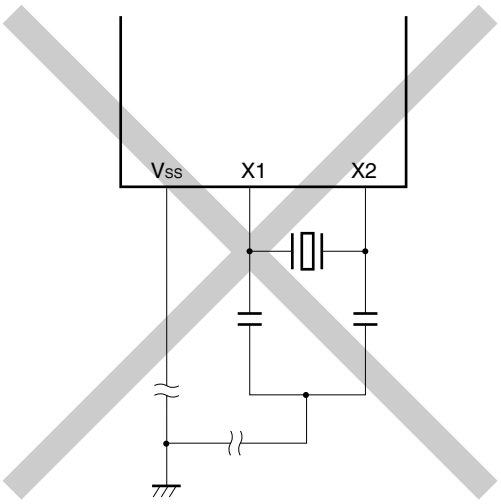
Caution When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the Figure 7-10 to avoid an adverse effect from wiring capacitance.

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

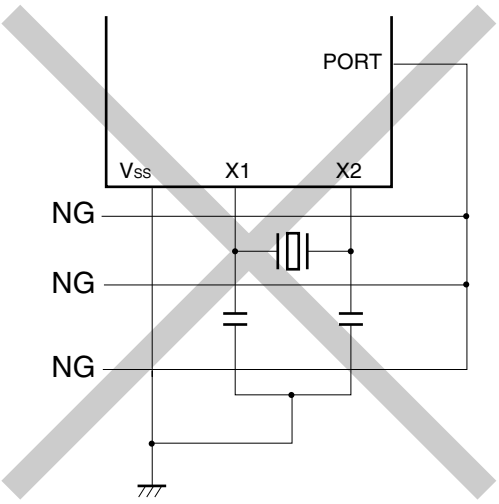
Figure 7-11 shows examples of incorrect resonator connection.

Figure 7-11. Examples of Incorrect Resonator Connection (1/2)

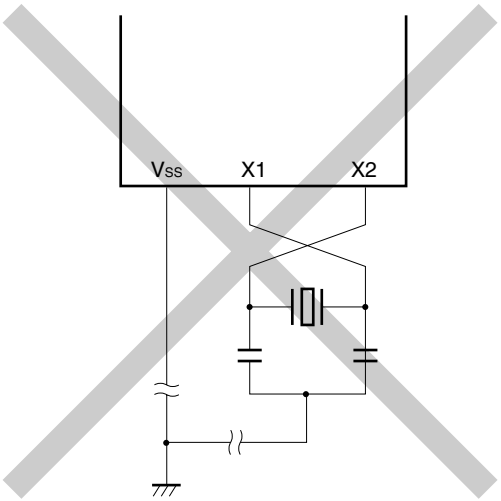
(a) Too long wiring



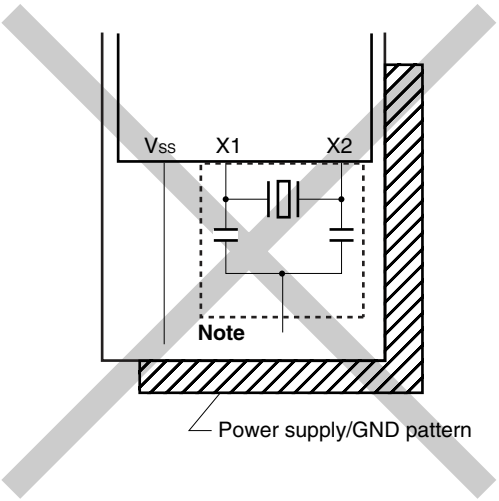
(b) Crossed signal line



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



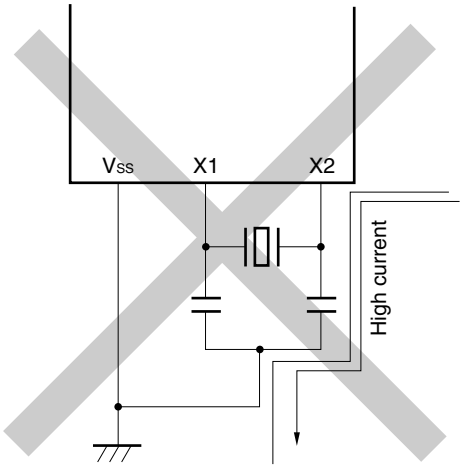
(d) A power supply/GND pattern exists under the X1 and X2 wires.



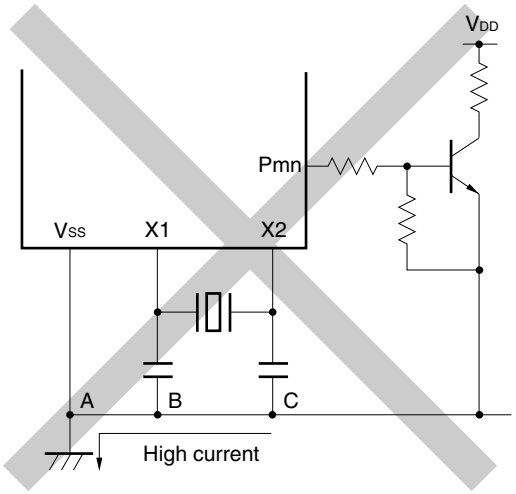
Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

Figure 7-11. Examples of Incorrect Resonator Connection (2/2)

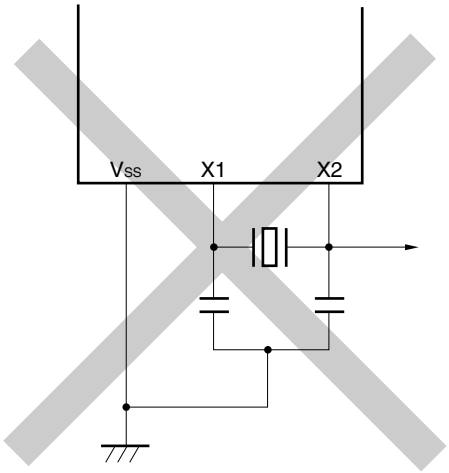
(e) Wiring near high alternating current



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



(g) Signals are fetched



7.4.2 Internal high-speed oscillator

The 20 MHz internal high-speed oscillator is incorporated in the μPD78F8040, 78F8041, 78F8042, 78F8043 (1, 8, and 20 MHz (TYP.)). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC) and bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL).

Caution To use the 1, 8, or 20 MHz internal high-speed oscillation clock, use the option byte to set the frequency in advance (for details, see CHAPTER 21 OPTION BYTE). Also, the internal high-speed oscillator automatically starts oscillating after reset release (If 8 MHz or 20 MHz is selected by using the option byte, the microcontroller operates using the 8 MHz internal high-speed oscillator). To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the DSCCTL register to 1.

7.4.3 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the μPD78F8040, 78F8041, 78F8042, 78F8043.

The internal low-speed oscillation clock is used only as the watchdog timer clock. The internal low-speed oscillation clock cannot be used as the CPU clock.

After a reset release, the internal low-speed oscillator automatically starts oscillation, and the watchdog timer is driven (30 kHz (TYP.)) if the watchdog timer operation is enabled by the option byte.

The internal low-speed oscillator continues oscillation except when the watchdog timer stops. When the watchdog timer operates, the internal low-speed oscillation clock does not stop, even in case of a program loop.

7.4.4 Prescaler

The prescaler generates a CPU/peripheral hardware clock by dividing the main system clock.

7.5 Clock Generator Operation

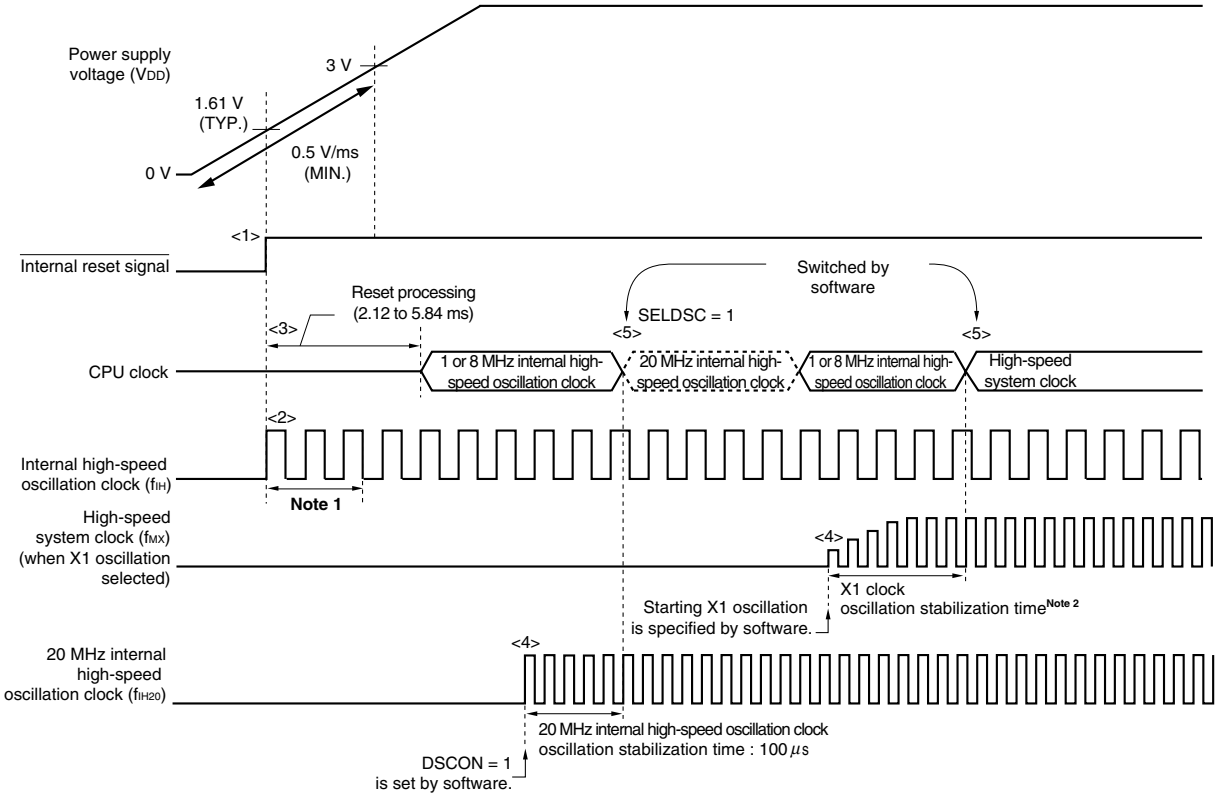
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 7-1**).

- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - Internal high-speed oscillation clock f_{IH}
  - 20 MHz internal high-speed oscillation clock f_{IH20}
- Internal low-speed oscillation clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

The CPU starts operation when the internal high-speed oscillator starts outputting after a reset release in the μPD78F8040, 78F8041, 78F8042, 78F8043.

When the power supply voltage is turned on, the clock generator operation is shown in Figures 7-12 to 7-15.

Figure 7-12. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))



<1> When the power is turned on, an internal reset signal is generated by the power-on-clear (POC) circuit.

<2> When the power supply voltage exceeds 1.61 V (TYP.), the reset is released and the internal high-speed oscillator[Note 3] automatically starts oscillation.

<3> The CPU starts operation on the internal high-speed oscillation clock[Note 3] 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 clock via software (see **(1)** in **7.6.4 Example of setting X1 oscillation clock**).

<5> When switching the CPU clock to the X1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **7.6.4 Example of setting X1 oscillation clock**).

Switch to the 20 MHz internal high-speed oscillation clock by setting the DSCON bit (bit 0 of the 20 MHz internal high-speed oscillation control register (DSCCTL)), waiting for 100 μs, and then setting the SELDSC bit to 1 by using software[Note 4].

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

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

3. The microcontroller operates on the 8 MHz internal high-speed oscillation clock if 8 MHz or 20 MHz is selected for the internal high-speed oscillator by using the option byte or on the 1 MHz internal high-speed oscillation clock if 1 MHz is selected.

4. If the internal high-speed oscillator is set to 1 MHz by using the option byte, the 20 MHz internal high-speed oscillation clock cannot be used.

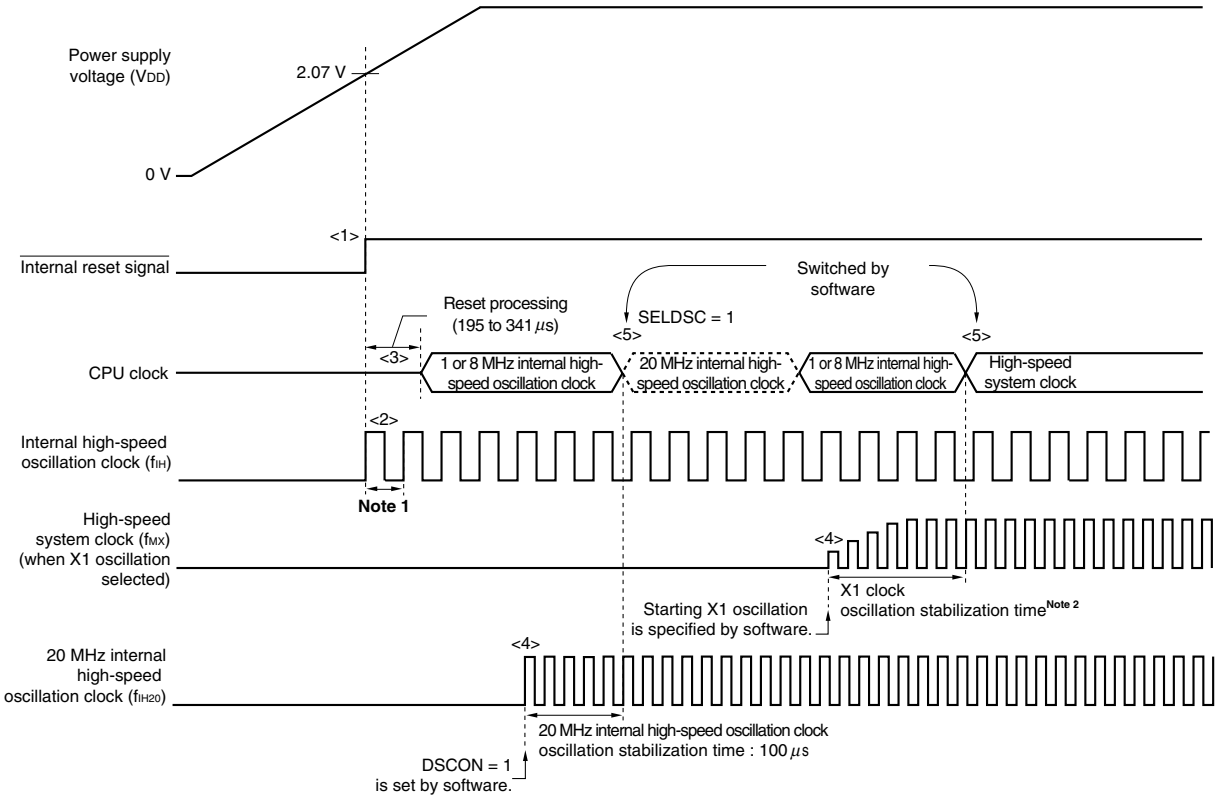
(**Cautions** are listed on the next page.)

Cautions 1. If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 3.0 V, input a low level to the RESET pin from power application until the voltage reaches 3.0 V, or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 7-14). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 7-13 after reset release by the RESET pin.

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

3. Some operations can also be executed while V_{DD} < 3.0 V (For details, figures of CHAPTER26 ELECTRICAL SPECIFICATIONS AC Characteristics (1) Basic operation.

Figure 7-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0))



<1> When the power is turned on, an internal reset signal is generated by the low-voltage detector (LVI) circuit.

<2> When the power supply voltage exceeds 2.07 V (TYP.), the reset is released and the internal high-speed oscillator Note 3 automatically starts oscillation.

<3> After the reset is released and reset processing is performed, the CPU starts operation on the internal high-speed oscillation clock Note 3.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **7.6.4 Example of setting X1 oscillation clock**).

Switch to oscillation using the 20 MHz internal high-speed oscillation clock after setting the DSCON bit to 1 by using software.

<5> When switching the CPU clock to the X1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **7.6.4 Example of setting X1 oscillation clock**).

Switch to the 20 MHz internal high-speed oscillation clock after setting the DSCON bit (bit 0 of the 20 MHz internal high-speed oscillation control register (DSCCTL)), waiting for 100 μs, and then setting the SELDSC bit to 1 by using software Note 4.

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

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

3. The microcontroller operates on the 8 MHz internal high-speed oscillation clock if 8 MHz or 20 MHz is selected for the internal high-speed oscillator by using the option byte or on the 1 MHz internal high-speed oscillation clock if 1 MHz is selected.

4. If the internal high-speed oscillator is set to 1 MHz by using the option byte, the 20 MHz internal high-speed oscillation clock cannot be used.

Cautions 1. A voltage stabilization time (about 2.12 to 5.84 ms) is required after the supply voltage reaches 1.61 V (TYP.). If the time for the supply voltage to rise from 1.61 V (TYP.) to 2.07 V (TYP.) is shorter than the voltage stabilization time, reset processing is entered after the voltage stabilization time elapses.

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

7.6 Controlling Clock

7.6.1 Example of setting 8 MHz internal high-speed oscillator

To use the 8 MHz internal high-speed oscillation clock as the CPU/peripheral hardware clock (f_{CLK}), set 000C1H of the option byte to FBH. Use the system clock control register (CKC) to specify the division ratio for the clock to be supplied to the CPU/peripheral hardware clock after releasing reset. When using the default division setting (f_{IH}/2 = 4 MHz), the CKC register is not required to be set.

[Option byte setting]

Set address 000C1H to FBH.

Option byte (000C1H)	7	6	5	4	3	2	1	0
	1	1	1	1	1	FRQSEL2 0	FRQSEL1 1	LVIOFF 1

LVIOFF bit: Set this bit to 0 to turn on the LVI by default when releasing the power-on-reset.

[Register settings]

<1> Use the MDIV2 to MDIV0 bits of the CKC register to specify the division ratio for the CPU/peripheral hardware clock.

CKC	7	6	5	4	3	2	1	0
	0	0	MCS 0	MCM0 0	1	MDIV2 0/1	MDIV1 0/1	MDIV0 0/1

7.6.2 Example of setting 1 MHz internal high-speed oscillator

To use the 1 MHz internal high-speed oscillation clock as the CPU/peripheral hardware clock (f_{CLK}), set 000C1H of the option byte to FDH. Use the system clock control register (CKC) to specify the division ratio for the clock to be supplied to the CPU/peripheral hardware clock after releasing reset. When using the default division setting ($f_{IH}/2$ = 0.5 MHz), the CKC register is not required to be set.

[Option byte setting]

Set address 000C1H to FDH.

Option byte (000C1H)	7	6	5	4	3	2	1	0
	1	1	1	1	1	FRQSEL2 1	FRQSEL1 0	LVIOFF 1

LVIOFF bit: Set this bit to 0 to turn on the LVI by default when releasing the power-on-reset.

[Register settings]

<1> Use the MDIV2 to MDIV0 bits of the CKC register to specify the division ratio for the CPU/peripheral hardware clock.

CKC	7	6	5	4	3	2	1	0
	0	0	MCS 0	MCM0 0	1	MDIV2 0/1	MDIV1 0/1	MDIV0 0/1

7.6.3 Example of setting 20 MHz internal high-speed oscillator

To use the 20 MHz internal high-speed oscillation clock as the CPU/peripheral hardware clock (f_{CLK}), set 000C1H of the option byte to FBH. After releasing reset, set the operation speed mode control register (OSMC) and then the 20 MHz internal high-speed oscillation control register (DSCCTL).

[Option byte setting]

Set address 000C1H to FBH.

Option byte (000C1H)	7	6	5	4	3	2	1	0
	1	1	1	1	1	FRQSEL2 0	FRQSEL1 1	LVIOFF 1

LVIOFF bit: Set this bit to 0 to turn on the LVI by default when releasing the power-on-reset.

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

<1> Set the OSMC register so that the microcontroller operates at a frequency exceeding 10 MHz.

OSMC	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	FLPC 0	FSEL 1

RTCLPC bit: Set this bit to 1 to operate only the watch in sub-HALT mode (ultra-low current consumption).

<2> Set (1) the DSCON bit of the DSCCTL register to operate the 20 MHz internal high-speed oscillator.

DSCCTL	7	6	5	4	3	2	1	0
	0	0	0	0	DSCS 0	SELDSC 0	0	DSCON 1

<3> Set (1) the DSCON bit and then wait for 100 μs.

<4> Set (1) the SELDSC bit of the DSCCTL register to switch the internal high-speed oscillation clock from 8 MHz to 20 MHz.

DSCCTL	7	6	5	4	3	2	1	0
	0	0	0	0	DSCS 0	SELDSC 1	0	DSCON 1

<5> Use the MDIV2 to MDIV0 bits of the CKC register to specify the division ratio for the CPU/peripheral hardware clock.

CKC	7	6	5	4	3	2	1	0
	0	0	MCS 0	MCM0 0	1	MDIV2 0/1	MDIV1 0/1	MDIV0 0/1

7.6.4 Example of setting X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the internal high-speed oscillation clock. To subsequently change the clock to the X1 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) 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).

Set the frequency of the internal oscillation clock to be supplied immediately after releasing reset by using the option byte.

[Option byte setting]

Set address 000C1H to FBH.

Option byte (000C1H)

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2 0	FRQSEL1 1	LVIOFF 1

LVIOFF bit: Set this bit to 0 to turn on the LVI by default when releasing the power-on-reset.

FRQSEL2 and FRQSEL1 bits: Set the FRQSEL2 and FRQSEL1 bits to 1 and 0, respectively, to set the internal oscillation clock frequency to 1 MHz.

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

<1> Use the OSMC register to set the frequency of the CPU/peripheral hardware.

OSMC

7	6	5	4	3	2	1	0
0	0	0	0	0	0	FLPC 0	FSEL 1

FSEL bit: Set this bit to 0 if the CPU/peripheral hardware clock is 10 MHz or less.

<2> Set (1) the OSCSEL bit of the CMC register to operate the X1 oscillator.

CMC

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

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

<3> 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 0	0	0	0	0	0	0	HIOSTOP 0

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

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

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

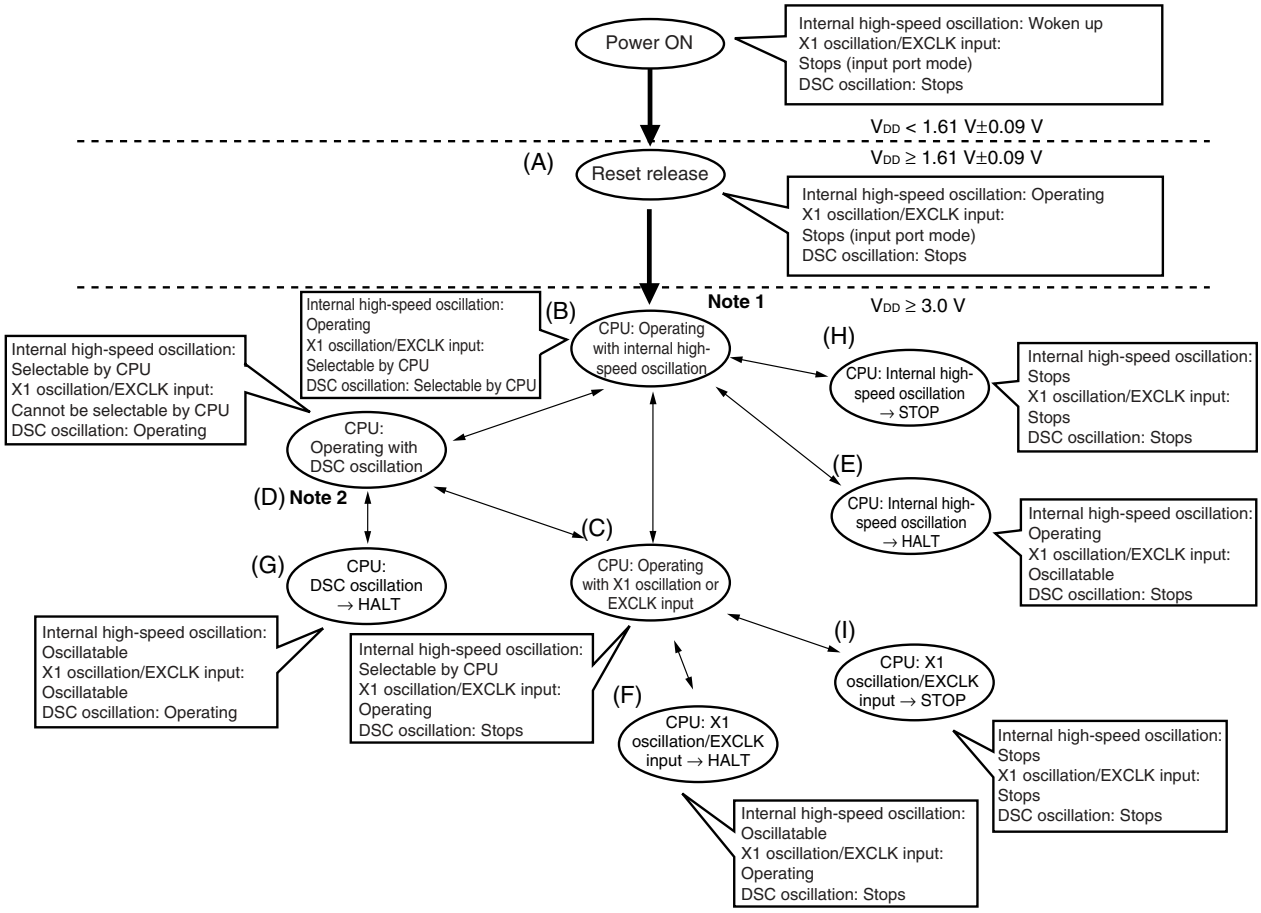
<5> Use the MCM0 bit of the CKC register to specify the X1 oscillation clock as the CPU/peripheral hardware clock. Use the MDIV2 to MDIV0 bits to specify the division ratio.

	7	6	5	4	3	2	1	0
CKC			MCS	MCM0		MDIV2	MDIV1	MDIV0
	0	0	0	1	1	0/1	0/1	0/1

7.6.5 CPU clock status transition diagram

Figure 7-14 shows the CPU clock status transition diagram of this product.

Figure 7-14. CPU Clock Status Transition Diagram



Notes 1. After reset release, following operation is started, because $f_{CLK} = f_{IH}/2$ has been selected by setting the system clock control register (CKC) to 09H.

- When 1 MHz has been selected by using the option byte: 500 kHz (1 MHz/2)
- When 8 MHz or 20 MHz has been selected by using the option byte: 4 MHz (8 MHz/2)

2. 20 MHz internal oscillation cannot be used if 1 MHz internal oscillation is selected by using the option byte.

Remarks 1. If the low-power-supply detector (LVI) is set to ON by default by the option bytes, the reset will not be released until the power supply voltage (V_{DD}) exceeds 2.07 V±0.2 V.
After the reset operation, the status will shift to (B) in the above figure.

2. DSC: 20 MHz internal high-speed oscillation clock

Table 7-4 shows transition of the CPU clock and examples of setting the SFR registers.

Table 7-4. CPU Clock Transition and SFR Register Setting Examples (1/3)

(1) CPU operating with internal high-speed oscillation 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 internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note 1			CSC Register	OSMC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH	MSTOP	FSEL		MCM0
(A) → (B) → (C) (X1 clock: 2 MHz ≤ fx ≤ 10 MHz)	0	1	0	0	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	0	1 Note 2	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	×	0	0/1 Note 2	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. FSEL = 1 when f_{CLK} > 10 MHz
If a divided clock is selected and f_{CLK} ≤ 10 MHz, use with FSEL = 0 is possible even if fx > 10 MHz.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 26 ELECTRICAL SPECIFICATIONS).

Remark ×: don't care

(3) CPU operating with 20 MHz internal high-speed oscillation clock (D) after reset release (A)

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register	Waiting for Oscillation Stabilization	DSCCTL Register
	DSCON		SELDSC
(A) → (B) → (D)	1	Necessary (100 μs)	1

Remark (A) to (i) in Table 7-4 correspond to (A) to (i) in Figure 7-14.

Table 7-4. CPU Clock Transition and SFR Register Setting Examples (2/3)

(4) CPU clock changing from internal high-speed oscillation 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	OSMC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP	FSEL		MCM0
(B) → (C) (X1 clock: 2 MHz ≤ fx ≤ 10 MHz)	0	1	0	**Note 2**	0	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	**Note 2**	0	1[Note 3]	Must be checked	1
(B) → (C) (external main clock)	1	1	×	**Note 2**	0	0/1	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 OSTS

3. FSEL = 1 when f_{CLK} > 10 MHz

If a divided clock is selected and f_{CLK} ≤ 10 MHz, use with FSEL = 0 is possible even if fx > 10 MHz.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 26 ELECTRICAL SPECIFICATIONS).

(5) CPU clock changing from internal high-speed oscillation clock (B) to 20 MHz internal high-speed oscillation clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register	Waiting for Oscillation Stabilization	DSCCTL Register
	DSCON		SELDSC
(B) → (D)	1	Necessary (100 μs)	1

Unnecessary if the CPU is operating with the 20 MHz internal high-speed oscillation clock

Remarks 1. ×: don't care

2. (A) to (i) in Table 7-4 correspond to (A) to (i) in Figure 7-14.

Table 7-4. CPU Clock Transition and SFR Register Setting Examples (3/3)

(6) CPU clock changing from high-speed system clock (C) to internal high-speed oscillation 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	10 μs	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

(7) CPU clock changing from 20 MHz internal high-speed oscillation clock (D) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register	
	SELDSC	DSCON
(D) → (B)	0	0

(8)
- **HALT mode (E) set while CPU is operating with internal high-speed oscillation 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 20 MHz internal high-speed oscillation clock (D)**

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

(9)
- **STOP mode (H) set while CPU is operating with internal high-speed oscillation 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 stop		Sets the OSTS register	
	In X1 oscillation		–	

Remark (A) to (i) in Table 7-4 correspond to (A) to (i) in Figure 7-14.

7.6.6 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 7-5. Changing CPU Clock

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Internal high-speed oscillation 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 internal high-speed oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	20 MHz internal high-speed oscillation clock	Stabilization of DSC oscillation with 20 MHz set by using the option byte • After elapse of oscillation stabilization time (100 μs) after setting to DSCON = 1 • SELDSC = 1	–
X1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed 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.)	–
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
External main system clock	Internal high-speed oscillation clock	Oscillation of internal high-speed 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.)	–
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
20 MHz internal high-speed oscillation clock	Internal high-speed oscillation clock	• SELDSC = 0 (Set when changing the clock.)	20 MHz internal high-speed oscillation clock can be stopped (DSCON = 0)
	X1 clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
	External main system clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–

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

By setting bits 0 to 2 and 4 (MDIV0 to MDIV2, MCM0) of the system clock control register (CKC), main system clock can be switched (between the internal high-speed oscillation clock and the high-speed system clock) and the division ratio of the main system clock can be changed.

The actual switchover operation is not performed immediately after rewriting to CKC; operation continues on the pre-switchover clock for several clocks (see Table 7-6 to Table 7-8).

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

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

Table 7-6. Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{MAIN}	←→ (changing the division ratio)	f_{MAIN}	See Table 7-7
f_{IH}	←→	f_{MX}	See Table 7-8

Table 7-7. Maximum Number of Clocks Required for f_{MAIN} ↔ f_{MAIN} (Changing the Division Ratio)

Set Value Before Switchover	Set Value After Switchover	
	Clock A	Clock B
Clock A		1 + f_A/f_B clock
Clock B	1 + f_B/f_A clock	

Table 7-8. Maximum Number of Clocks Required for f_{IH} ↔ 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	

Remarks 1. The number of clocks listed in Table 7-6 to Table 7-8 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Table 7-6 to Table 7-8 by removing the decimal portion.

Example When switching the main system clock from the internal high-speed oscillation clock 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

7.6.8 Conditions before clock oscillation is stopped

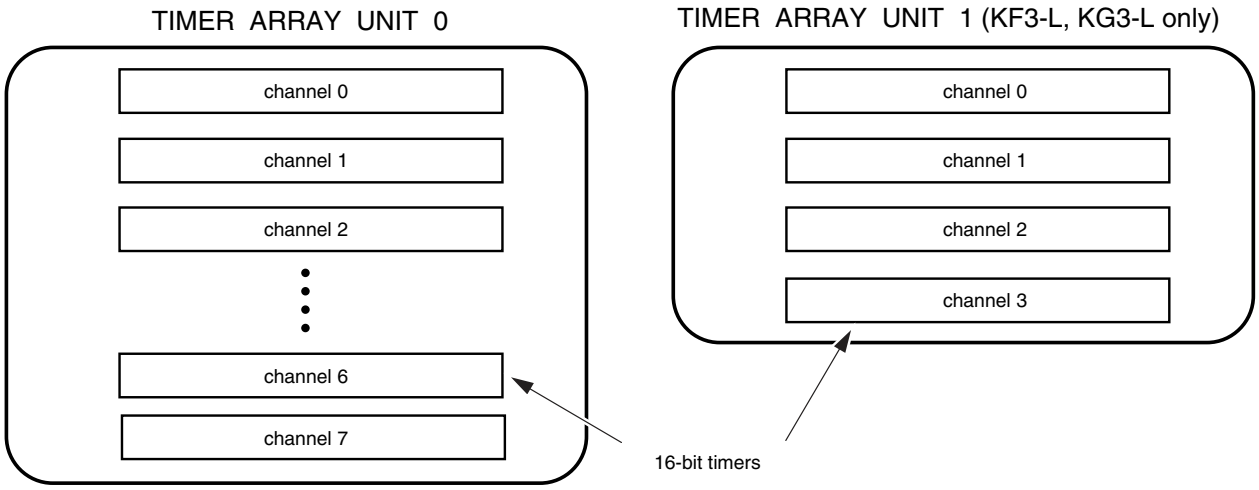
The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

Table 7-9. 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
Internal high-speed oscillation clock	MCS = 1 (The CPU is operating on a clock other than the internal high-speed oscillation clock.)	HIOSTOP = 1
X1 clock	MCS = 0 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		
20 MHz internal high-speed oscillation clock	SELDSC = 0 (The main system clock is operating on a clock other than the 20 MHz internal high-speed oscillation clock.)	DSCON = 0

CHAPTER 8 TIMER ARRAY UNIT

The timer array unit has two units. The timer array unit 0 has eight 16-bit timers and the timer array unit 1 has four 16-bit timers. Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more "channels" can be used to create a high-accuracy timer.



For details about each function, see the table below.

Independent channel operation function	Simultaneous channel operation function
• **Interval timer (→ refer to 8.7.1)** • **Square wave output (→ refer to 8.7.1)** • **External event counter (→ refer to 8.7.2)** • **Input pulse interval measurement (→ refer to 8.7.3)** • **Measurement of high-/low-level width of input signal (→ refer to 8.7.4)**	• **One-shot pulse output(→ refer to 8.8.1)** • **PWM output(→ refer to 8.8.2)** • **Multiple PWM output(→ refer to 8.8.3)**

Channel 7 of the unit 0 can be used to realize LIN-bus reception processing in combination with UART of the serial array unit.

8.1 Functions of Timer Array Unit

The timer array unit has the following functions.

8.1.1 Functions of each channel when it operates independently

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

(1) Interval timer

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

(2) Square wave output

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

(3) External event counter

Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TImn) has reached a specific value.

(4) Input pulse interval measurement

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

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

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 13
In case of timer input pin (TImn): mn = 01 to 03, 05, 11, 13
In case of timer output pin (TOmn): mn = 01 to 03, 05, 11, 13

8.1.2 Functions of each channel when it operates with another channel

Combination-operation functions are those functions that are attained by using the master channel (mostly the reference timer that controls cycles) and the slave channels (timers that operate following the master channel) in combination.

(1) One-shot pulse output

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

(2) PWM (Pulse Width Modulation) output

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

(3) Multiple PWM (Pulse Width Modulation) output

By extending the PWM function and using one master channel and two or more slave channels, up to three types of PWM signals that have a specific period and a specified duty factor can be generated.

Caution The following rules apply when using multiple channels simultaneously.

- **Only an even-numbered channel (channel 0, 2, 4, …) can be specified as the master channel.**
- **Only channels with lower channel numbers than the master channel can be specified as slave channels (multiple slave channels can be set).**

For details about the rules of simultaneous channel operation function, see 8.4 Basic Rules of Simultaneous Channel Operation Function.

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number , q: Slave channel number
When m = 0: $n < p < q \leq 5$
When m = 1: $n < p < q \leq 3$
(Where p and q are a consecutive integer greater than n)

8.1.3 LIN-bus supporting function (channel 7 only)

Timer array unit 0 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 (RxD3) of UART3 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 sync break field

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 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 sync break field.

(3) Measurement of pulse width of sync field

After a sync break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD3) of UART3 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 **8.3 (12) Input switch control register (ISC)** and **8.7.4 Operation as input signal high-/low-level width measurement**.

8.2 Configuration of Timer Array Unit

The timer array unit includes the following hardware.

Table 8-1. Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer counter register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI01 to TI03, TI05, TI11, TI13 pins, RxD3 pin (for LIN-bus)
Timer output	TO01 to TO03, TO05, TO11, TO13 pins, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer output enable register m (TOEm) • Timer output register m (TOm) • Timer output level register m (TOLm) • Timer output mode register m (TOMm)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Input switch control register (ISC) • Noise filter enable registers 1, 2 (NFEN1, NFEN2) • Port mode registers 0, 1, 3, 6 (PM0, PM1, PM3, PM6) • Port registers 0, 1, 3, 6 (P0, P1, P3, P6)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 13

Figure 8-1 to Figure 8-3 show the block diagrams.

Figure 8-1. Entire Configuration of Timer Array Unit 0

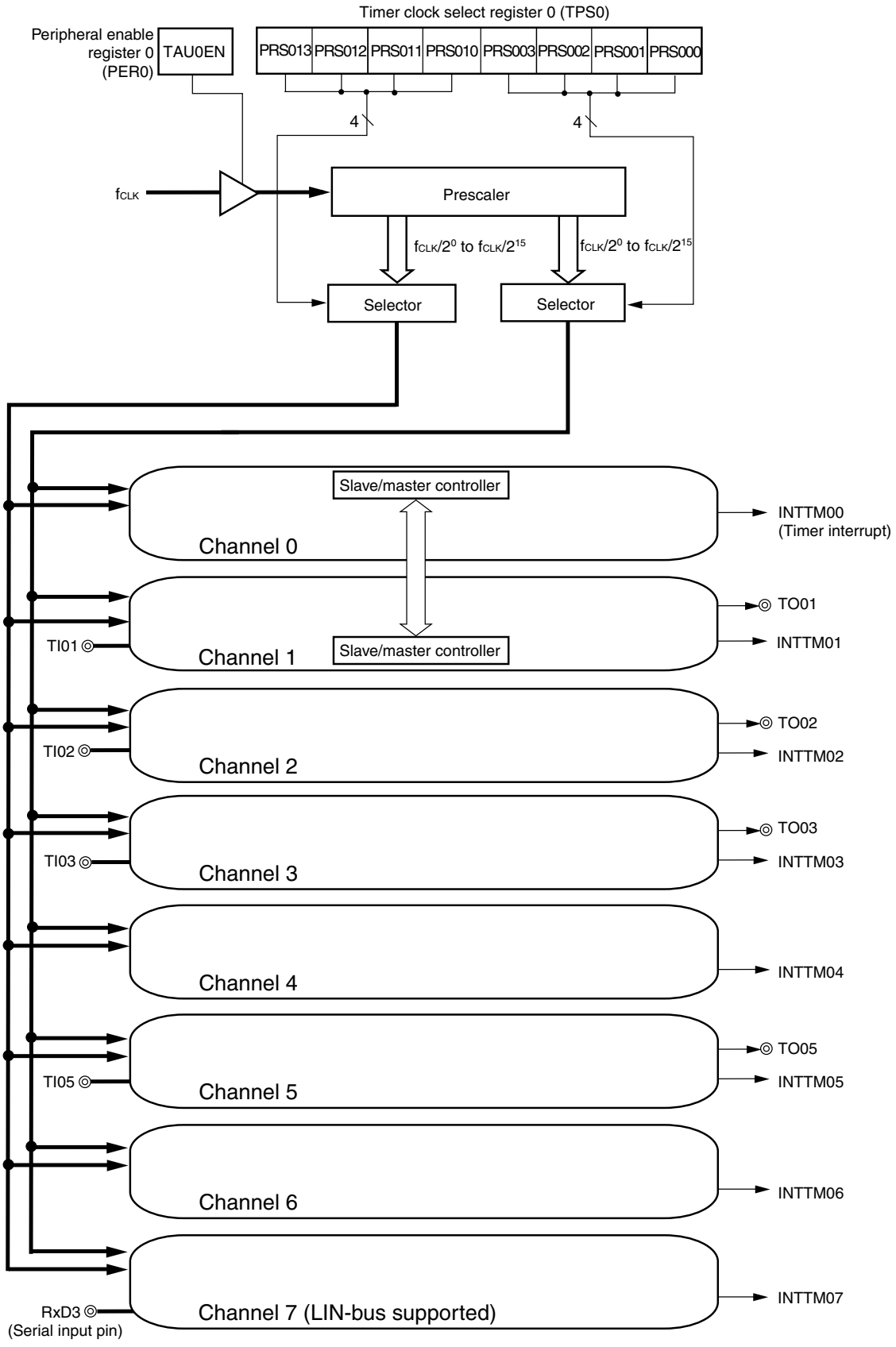


Figure 8-2. Entire Configuration of Timer Array Unit 1

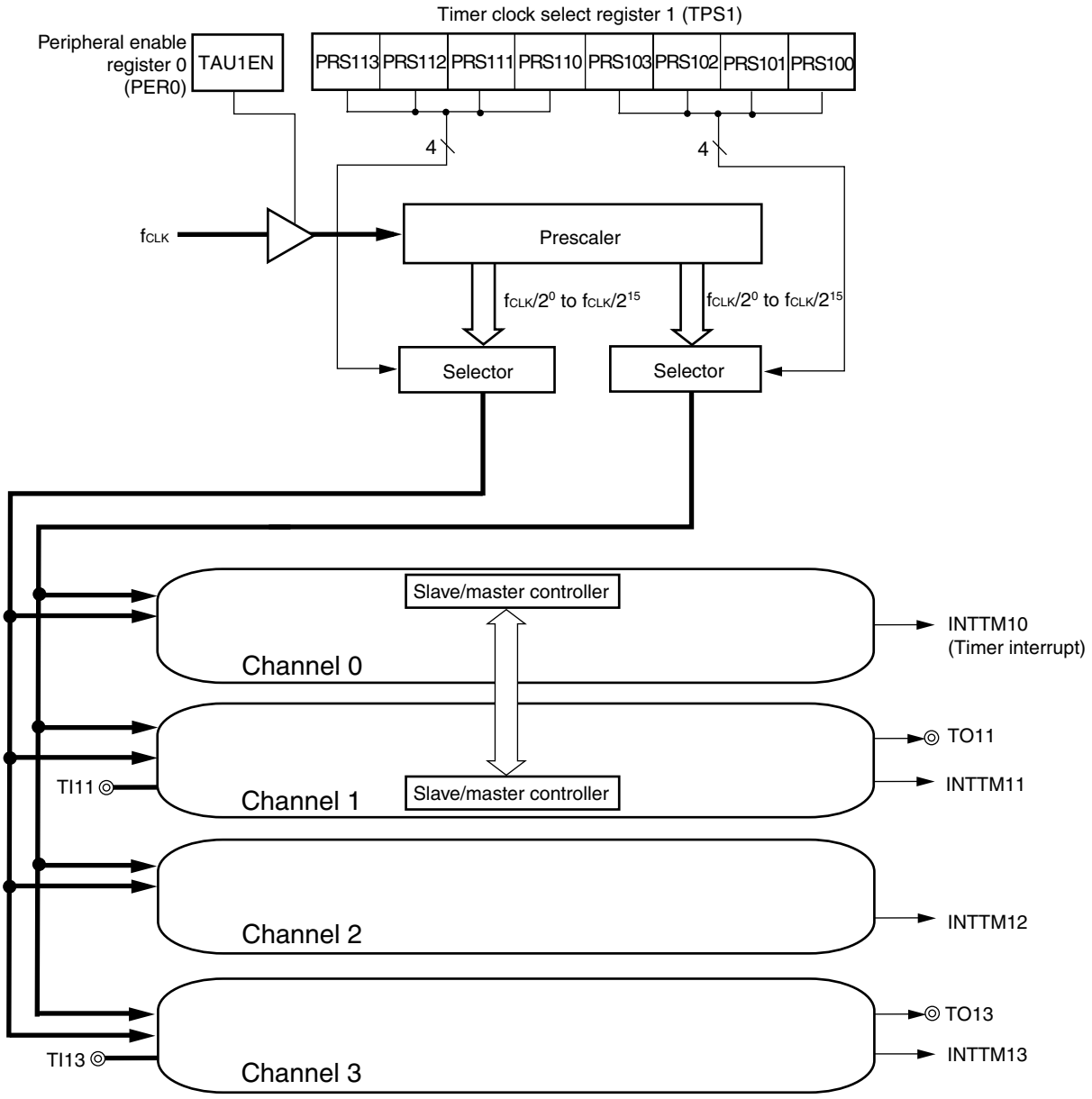
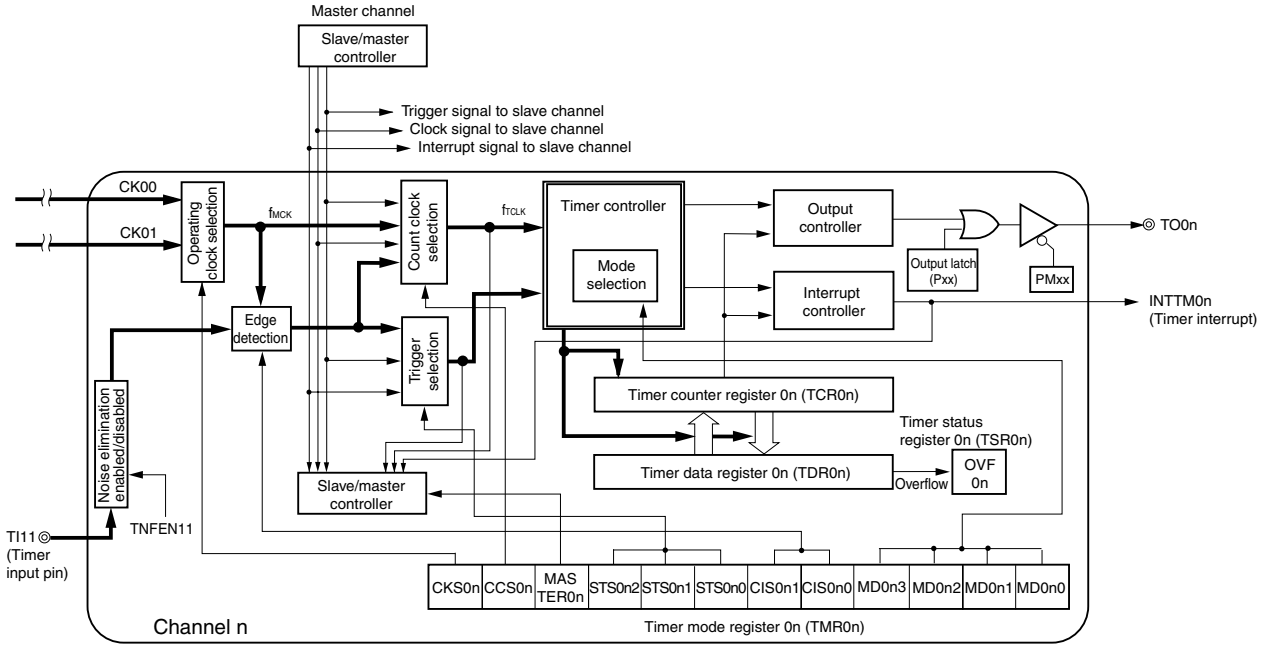


Figure 8-3. Internal Block Diagram of Channel of Timer Array Unit 0



Remark n = 0 to 7

(1) Timer counter register mn (TCRmn)

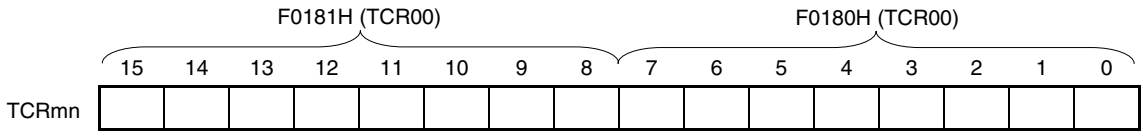
The TCRmn register is a 16-bit read-only register and is used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn) (refer to **8.3 (3) Timer mode register mn (TMRmn)**).

Figure 8-4. Format of Timer Counter Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07) After reset: FFFFH R
F01C0H, F01C1H (TCR10) to F01C6H, F01C7H (TCR13)



The count value can be read by reading TCRmn.

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

- When the reset signal is generated
- When the TAU0EN bit (in case of TAU0) or TAU1EN bit (in case of TAU1) 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 master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13

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

Table 8-2. Timer/counter Register mn (TCRmn) Read Value in Various Operation Modes

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

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13

(2) Timer data register mn (TDRmn)

This is a 16-bit register from which a capture function and a compare function can be selected.

The capture or compare function can be switched by selecting an operation mode by using the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn).

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

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

Reset signal generation clears this register to 0000H.

Figure 8-5. Format of Timer Data Register mn (TDRmn)

Address: FFF18H, FFF19H (TDR00), FFF1AH, FFF1BH (TDR01), FFF64H, FFF65H (TDR02) to FFF6EH, FFF6FH (TDR07) FFF70H, FFF71H (TDR10) to FFF76H, FFF77H (TDR13) After reset: 0000H R/W

	FFF19H (TDR00)								FFF18H (TDR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRmn																

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

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

Caution TDRmn does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register mn (TDRmn) is used as capture register

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

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

8.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Input switch control register (ISC)
- Noise filter enable registers 1, 2 (NFEN1, NFEN2)
- Port mode registers 0, 1, 3, 6 (PM0, PM1, PM3, PM6)
- Port registers 0, 1, 3, 6 (P0, P1, P3, P6)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13

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

When the timer array unit 1 is used, be sure to set bit 1 (TAU1EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 8-6. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

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

Cautions 1. When setting the timer array unit, be sure to set TAUmEN to 1 first. If TAUmEN = 0, writing to a control register of the timer array unit is ignored, and all read values are default values (except for noise filter enable registers 1, 2 (NFEN1, NFEN2), port mode registers 0, 1, 3, 6 (PM0, PM1, PM3, PM6), and port registers 0, 1, 3, 6 (P0, P1, P3, P6)).

2. Be sure to clear bits 6 and 7 to 0.

Remark m = 0, 1

(2) Timer clock select register m (TPSm)

TPSm 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 TPSm, and CKm0 is selected by bits 3 to 0.

Rewriting of TPSm register during timer operation is possible only in the following cases.

If the PRSm00 to PRSm03 bits can be rewritten:

All channels for which CKm0 is selected as the operation clock (CKSmn = 0) are stopped (TEmn = 0).

If the PRSm10 to PRSm13 bits can be rewritten:

All channels for which CKm1 is selected as the operation clock (CKSmn = 1) are stopped (TEmn = 0).

TPSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TPSm can be set with an 8-bit memory manipulation instruction with TPSmL.

Reset signal generation clears this register to 0000H.

Figure 8-7. Format of Timer Clock Select Register m (TPSm)

Address: F01B6H, F01B7H (TPS0) After reset: 0000H R/W
F01DEH, F01DFH (TPS1)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	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				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	39.1 kHz	78.1 kHz	156.2 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz

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

Caution Be sure to clear bits 15 to 8 to "0".

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency
2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), k = 0, 1, mn = 00 to 07, 10 to 13

(3) Timer mode register mn (TMRmn)

TMRmn sets an operation mode of channel n. It is used to select an operation clock (f_{MCK}), a count clock, whether the timer operates as the master or a slave, a start trigger and a capture trigger, the valid edge of the timer input, and an operation mode (interval, capture, event counter, one-count, or capture & one-count).

Rewriting TMRmn is prohibited when the register is in operation (when TEm = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEm = 1) (for details, see **8.7 Operation of Timer Array Unit as Independent Channel** and **8.8 Operation of Plural Channels of Timer Array Unit**).

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

Reset signal generation clears this register to 0000H.

Figure 8-8. Format of Timer Mode Register mn (TMRmn) (1/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn	0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by timer clock select register m (TPSm)
1	Operation clock CKm1 set by timer clock select register m (TPSm)
Operation clock f_{MCK} is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit.	

CCSmn	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock f_{MCK} specified by CKSmn bit
1	Valid edge of input signal input from TImn pin
Count clock f_{TCLK} is used for the timer/counter, output controller, and interrupt controller.	

Cautions 1. Be sure to clear bits 14, 13, 5, and 4 to "0".

2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn bit (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, n = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

Figure 8-8. Format of Timer Mode Register mn (TMRmn) (2/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn	0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

MASTERmn	Selection between using channel n independently or simultaneously with another channel (as a slave or master)
0	Operates in independent channel operation function or as slave channel in simultaneous channel operation function.
1	Operates as master channel in simultaneous channel operation function.
Only the even channel can be set as a master channel (MASTERmn = 1). Be sure to use odd-numbered channels as slave channels (MASTERmn = 0). Clear the MASTERmn bit to 0 for a channel that is used with the independent channel operation function.	

STSmn2	STSmn1	STSmn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of TImn pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of TImn pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function).
Other than above			Setting prohibited

CISmn1	CISmn0	Selection of TImn pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B.		

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, n = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

Figure 8-8. Format of Timer Mode Register mn (TMRmn) (3/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn	0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

MDmn3	MDmn2	MDmn1	MDmn0	Operation mode of channel n	Count operation of TCR	Independent operation
0	0	0	1/0	Interval timer mode	Counting down	Possible
0	1	0	1/0	Capture mode	Counting up	Possible
0	1	1	0	Event counter mode	Counting down	Possible
1	0	0	1/0	One-count mode	Counting down	Impossible
1	1	0	0	Capture & one-count mode	Counting up	Possible
Other than above				Setting prohibited		
The operation of MDmn0 bit varies depending on each operation mode (see table below).						

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MDmn0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0) • Capture mode (0, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode Note 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 (INTTMmn) when starting a count operation and TOmn output are not controlled.

2. If the start trigger (TSmn = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, n = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

(4) Timer status register mn (TSRmn)

TSRmn indicates the overflow status of the counter of channel n.

TSRmn is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). It will not be set in any other mode. See Table 8-3 for the operation of the OVF bit in each operation mode and set/clear conditions.

TSRmn can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TSRmn can be set with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

Figure 8-9. Format of Timer Status Register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) After reset: 0000H R
F01D0H, F01D1H (TSR10) to F01D6H, F01D7H (TSR13)

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

OVF	Counter overflow status of channel n
0	Overflow does not occur.
1	Overflow occurs.
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13

Table 8-3. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF bit	Set/clear conditions
• Capture mode • Capture & one-count mode	clear	When no overflow has occurred upon capturing
	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	–
	set	(Use prohibited, not set and not cleared)

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 m (TEm)

TEm is used to enable or stop the timer operation of each channel.

When a bit of timer channel start register m (TSm) is set to 1, the corresponding bit of this register is set to 1.

When a bit of timer channel stop register m (TTm) is set to 1, the corresponding bit of this register is cleared to 0.

TEm can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TEm can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

Figure 8-10. Format of Timer Channel Enable Status Register m (TEm)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	0	0	0	0	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

Address: F01D8H, F01D9H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE1	0	0	0	0	0	0	0	0	0	0	0	0	TE13	TE12	TE11	TE10

TE mn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13

(6) Timer channel start register m (TSm)

The TSm register is a trigger register that is used to clear timer/counter register mn (TCRmn) and start the counting operation of each channel.

When a bit (TSmn) of this register is set to 1, the corresponding bit (TEmn) of timer channel enable status register m (TEm) is set to 1. The TSmn bit is immediately cleared when operation is enabled (TEmn = 1), because it is a trigger bit.

TSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TSm can be set with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

Figure 8-11. Format of Timer Channel Start Register m (TSm)

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	0	0	0	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

Address: F01DAH, F01DBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1	0	0	0	0	0	0	0	0	0	0	0	0	TS13	TS12	TS11	TS10

TSmn	Operation enable (start) trigger of channel n
0	No trigger operation
1	TEmn is set to 1 and the count operation becomes enabled. The TCRmn count operation start in the count operation enabled state varies depending on each operation mode (see Table 8-4).

Caution Be sure to clear bits 15 to 8 of TS0 and bits 15 to 4 of TS1 to "0"

Remarks 1. When the TSm register is read, 0 is always read.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13

Table 8-4. Operations from Count Operation Enabled State to Timer/counter Register mn (TCRmn) Count Start (1/2)

Timer operation mode	Operation when TSmn = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until count clock generation. The first count clock loads the value of TDRmn to TCRmn and the subsequent count clock performs count down operation (see **8.3 (6) (a) Start timing in interval timer mode**).
• Event counter mode	Writing 1 to TSmn bit loads the value of TDRmn to TCRmn. The subsequent count clock performs count down operation. The external trigger detection selected by STSmn2 to STSmn0 bits in the TMRmn register does not start count operation (see **8.3 (6) (b) Start timing in 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 TCRmn and the subsequent count clock performs count up operation (see **8.3 (6) (c) Start timing in capture mode**).

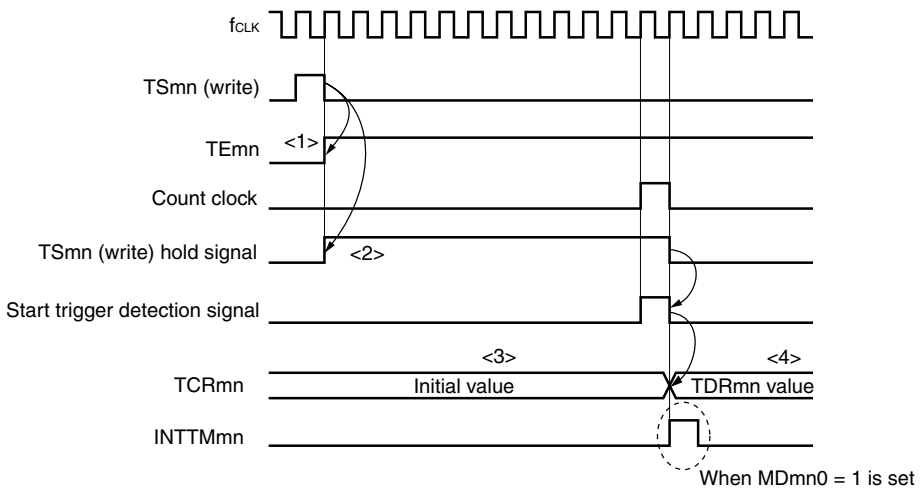
Table 8-4. Operations from Count Operation Enabled State to Timer/counter Register mn (TCRmn) Count Start (2/2)

Timer operation mode	Operation when TSmn = 1 is set
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of TDRmn to TCRmn and the subsequent count clock performs count down operation (see **8.3 (6) (d) Start timing in one-count mode**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to TCRmn and the subsequent count clock performs count up operation (see **8.3 (6) (e) Start timing in capture & one-count mode**).

(a) Start timing in interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn.
<2> The write data to the TSmn is held until count clock generation.
<3> Timer/counter register mn (TCRmn) holds the initial value until count clock generation.
<4> On generation of count clock, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and count starts.

Figure 8-12. Start Timing (In Interval Timer Mode)

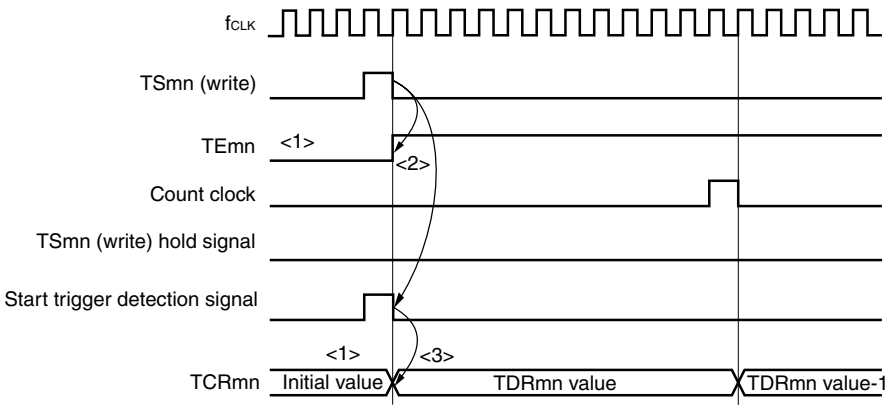


Caution In the first cycle operation of count clock after writing TSmn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

(b) Start timing in event counter mode

<1> Timer/counter register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).
<2> Operation is enabled (TEmn = 1) by writing 1 to the TSmn.
<3> As soon as 1 has been written to the TSmn bit and 1 has been set to the TEmn bit, the value of timer data register mn (TDRmn) is loaded to the TCRmn to start counting.
<4> After that, the TCRmn register value is counted down according to the count clock.

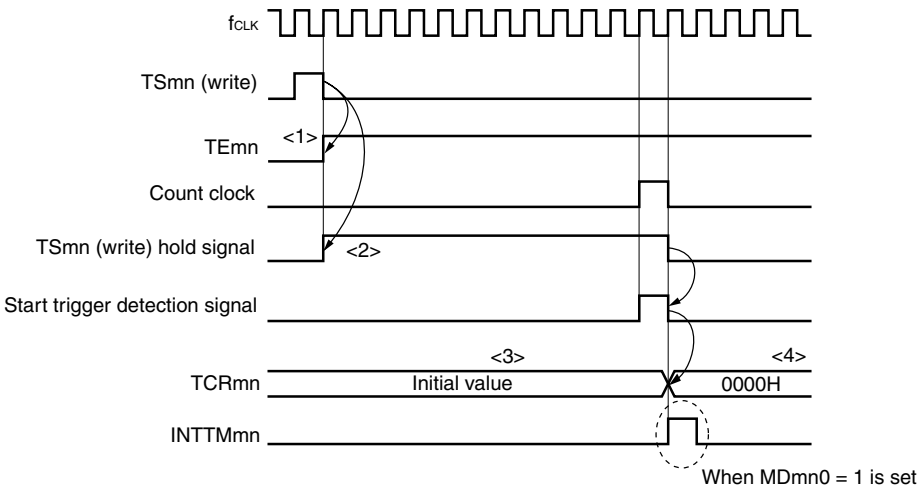
Figure 8-13. Start Timing (In Event Counter Mode)



(c) Start timing in capture mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn.
<2> The write data to the TSmn is held until count clock generation.
<3> Timer/counter register mn (TCRmn) holds the initial value until count clock generation.
<4> On generation of count clock, 0000H is loaded to the TCRmn register and count starts.

Figure 8-14. Start Timing (In Capture Mode)



Caution In the first cycle operation of count clock after writing TSmn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

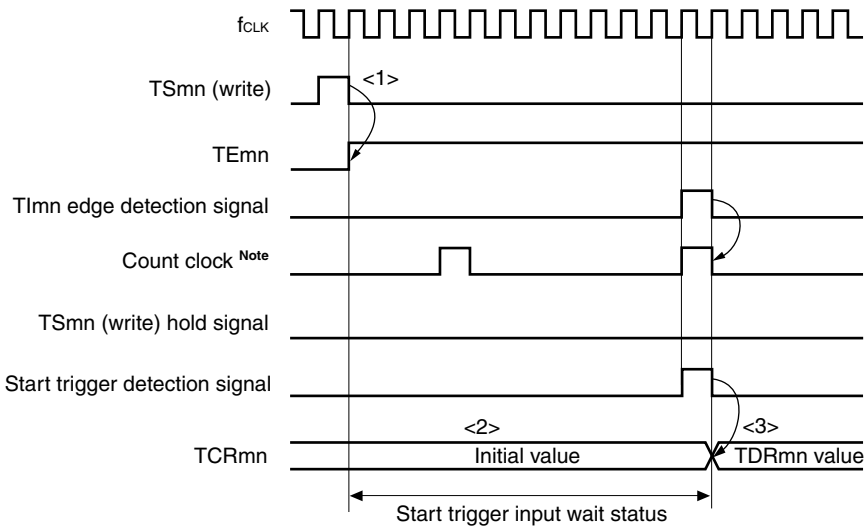
(d) Start timing in one-count mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn.

<2> Enters the start trigger input wait status, and timer/counter register mn (TCRmn) holds the initial value.

<3> On start trigger detection, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and count starts.

Figure 8-15. Start Timing (In One-count Mode)



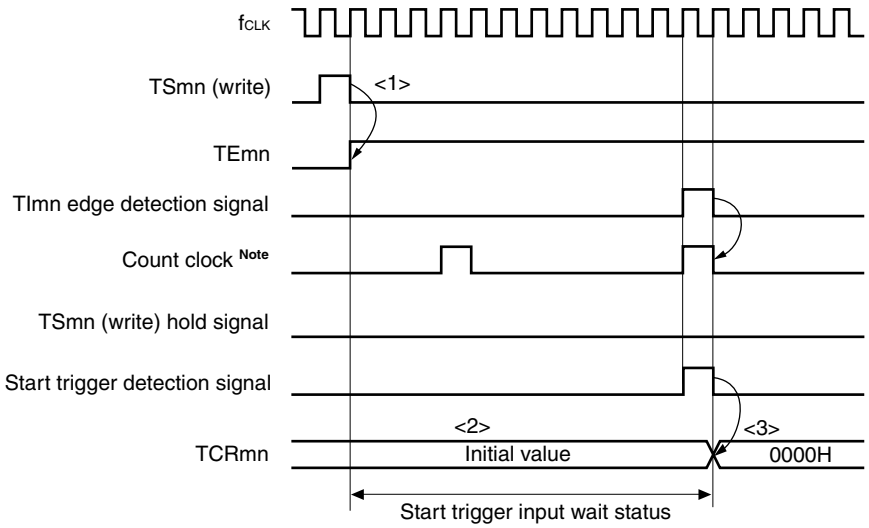
Note When the one-count mode is set, the operation clock (f_{MCK}) is selected as count clock (CCSmn = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (If the TImn 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 (TEmn = 1) by writing 1 to the Timer channel start register mn (TSmn) bit.
<2> Enters the start trigger input wait status, and timer/counter register mn (TCRmn) holds the initial value.
<3> On start trigger detection, 0000H is loaded to the TCRmn register and count starts.

Figure 8-16. 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 (CCSmn = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (If the TImn pin input signal is used as a start trigger, an error of one count clock occurs.)

(7) Timer channel stop register m (TTm)

The TTm register is a trigger register that is used to clear timer/counter register mn (TCRmn) and start the counting operation of each channel.

When a bit (TTmn) of this register is set to 1, the corresponding bit (TEmn) of timer channel enable status register m (TEm) is cleared to 0. The TTmn bit is immediately cleared when operation is stopped (TEmn = 0), because it is a trigger bit.

TTm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TTm can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

Figure 8-17. Format of Timer Channel Stop Register m (TTm)

Address: F01B4H, F01B5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	0	0	0	0	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

Address: F01DCH, F01DDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT1	0	0	0	0	0	0	0	0	0	0	0	0	TT13	TT12	TT11	TT10

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	Operation is stopped (stop trigger is generated).

Caution Be sure to clear bits 15 to 8 of TT0 and bits 15 to 4 of TT1 to "0".

Remarks 1. When the TTm register is read, 0 is always read.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13

(8) Timer output enable register m (TOEm)

TOEm is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TOmn bit of the timer output register (TOm) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TOmn).

TOEm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOEm can be set with a 1-bit or 8-bit memory manipulation instruction with TOEmL.

Reset signal generation clears this register to 0000H.

Figure 8-18. Format of Timer Output Enable Register m (TOEm)

Address: F01BAH, F01BBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	0	0	0	0	TOE07	TOE06	TOE05	TOE04	TOE03	TOE02	TOE01	TOE00

Address: F01E2H, F01E3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE1	0	0	0	0	0	0	0	0	0	0	0	0	TOE13	TOE12	TOE11	TOE10

TOEmn	Timer output enable/disable of channel n
0	The TOmn operation stopped by count operation (timer channel output bit). Writing to the TOmn bit is enabled. The TOmn pin functions as data output, and it outputs the level set to the TOmn bit. The output level of the TOmn pin can be manipulated by software.
1	The TOmn operation enabled by count operation (timer channel output bit). Writing to the TOmn bit is disabled (writing is ignored). The TOmn pin functions as timer output, and the TOEmn bit is set or reset depending on the timer operation. The TOmn pin outputs the square-wave or PWM depending on the timer operation.

Caution Be sure to clear bits 15 to 8 of TOE0 and bits 15 to 4 of TOE1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, n = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

(9) Timer output register m (TOm)

TOm is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TOmn) of each channel.

This register can be rewritten by software only when timer output is disabled (TOEmn = 0). When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P16/TO01, P17/TO02, P31/TO03, or P67/TO13 pin as a port function pin, set the corresponding TOmn bit to "0".

TOm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOm can be set with an 8-bit memory manipulation instruction with TOmL.

Reset signal generation clears this register to 0000H.

Figure 8-19. Format of Timer Output Register m (TOm)

Address: F01B8H, F01B9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	TO07	TO06	TO05	TO04	TO03	TO02	TO01	TO00

Address: F01E0H, F01E1H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO1	0	0	0	0	0	0	0	0	0	0	0	0	TO13	TO12	TO11	TO10

TOmn	Timer output of channel n
0	Timer output value is "0".
1	Timer output value is "1".

Caution Be sure to clear bits 15 to 8 of TO0 and bits 15 to 4 of TO1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, n = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

(10) Timer output level register m (TOLm)

The TOLm register is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOEmn = 1) in the simultaneous channel output mode (TOMmn = 1). In the master channel output mode (TOMmn = 0), this register setting is invalid.

TOLm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOLm can be set with an 8-bit memory manipulation instruction with TOLmL.

Reset signal generation clears this register to 0000H.

Figure 8-20. Format of Timer Output Level Register m (TOLm)

Address: F01BCH, F01BDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	0	0	TOL07	TOL06	TOL05	TOL04	TOL03	TOL02	TOL01	TOL00

Address: F01E4H, F01E5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL1	0	0	0	0	0	0	0	0	0	0	0	0	TOL13	TOL12	TOL11	TOL10

TOLmn	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 of TOL0 and bits 15 to 4 of TOL1 to "0".

Remarks 1. If the value of this register is rewritten during timer operation, the timer output is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13

(11) Timer output mode register m (TOMm)

TOMm is used to control the timer output mode of each channel.

When a channel is used for the independent channel operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the simultaneous channel operation function (PWM output, one-shot pulse output, or multiple PWM output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOEmn = 1).

TOMm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOMm can be set with an 8-bit memory manipulation instruction with TOMmL.

Reset signal generation clears this register to 0000H.

Figure 8-21. Format of Timer Output Mode Register m (TOMm)

Address: F01BEH, F01BFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	TOM07	TOM06	TOM05	TOM04	TOM03	TOM02	TOM01	TOM00

Address: F01E6, F01E7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM1	0	0	0	0	0	0	0	0	0	0	0	0	TOM13	TOM12	TOM11	TOM10

TOMmn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTMmn))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTMmp) of the slave channel)

Caution Be sure to clear bits 15 to 8 of TOM0 and bits 15 to 4 of TOM1 to "0".

Remark m: Unit number, n: Channel number, p: Slave channel number

When m = 0

n = 0 to 7 (n = 0, 2, 4 for master channel)

n < p ≤ 5 (where p is a consecutive integer greater than n)

When m = 1

n = 0 to 3 (n = 0 for master channel)

n < p ≤ 3 (where p is a consecutive integer greater than n)

(12) Input switch control register (ISC)

ISC0 bit of the ISC register is used to implement LIN-bus communication operation with channel 7 in association with serial array unit.

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

Reset signal generation clears this register to 00H.

Figure 8-22. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	1	ISC0

ISC0	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 RxD3 pin as an external interrupt (wakeup signal detection).

Caution Be sure to clear bits 7 to 2 to "0". And be sure to set bit 1 to "1".

Remark When the LIN-bus communication function is used, select the input signal of the RxD0 pin by setting ISC1 to 1.

(13) Noise filter enable registers 1, 2 (NFEN1, NFEN2)

The NFEN1 and NFEN2 registers are used to set whether the noise filter can be used for the timer input signal to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.

When the noise filter is ON, match detection of the 2 clocks and synchronization are performed with the CPU/peripheral hardware clock (f_{MCK}). When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f_{MCK}).

NFEN1 and NFEN2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-23. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0061H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	0	TNFEN05	0	TNFEN03	TNFEN02	TNFEN01	0

TNFEN07	Enable/disable using noise filter of RxD3/P14 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05/TO05/P05 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03/TO03/INTP4/P31 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02/TO02/P17 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01/TO01/INTP5/P16 pin input signal
0	Noise filter OFF
1	Noise filter ON

Figure 8-24. Format of Noise Filter Enable Register 2 (NFEN2)

Address: F0062H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN2	0	0	0	0	TNFEN13	0	TNFEN11	0

TNFEN13	Enable/disable using noise filter of TI13/TO13/P67 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN11	Enable/disable using noise filter of TI11/TO11/P65 pin input signal
0	Noise filter OFF
1	Noise filter ON

(14) Port mode registers 0, 1, 3, 6 (PM0, PM1, PM3, PM6)

These registers set input/output of ports 0, 1, and 3, 6 in 1-bit units.

When using the P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, P05/TO05/TI05, P65/TO11/TI11, and P67/TI13/TO13 pins for timer output, set PM16, PM17, PM31, PM05, PM65, and PM67 and the output latches of P16, P17, P31, P05, P65, and 67 to 0.

When using the P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, P05/TO05/TI05, P65/TI11/TO11, and P67/TI13/TO13 pins for timer input, set PM16, PM17, PM31, PM05, PM65, and PM67 to 1. At this time, the output latches of P16, P17, P31, P05, P65, and P67 may be 0 or 1.

PM0, PM1, PM3, and PM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 8-25. Format of Port Mode Registers 0, 1, 3, 6 (PM0, PM1, PM3, PM6)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	0	1	1

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15Note	PM14	PM13	PM12Note	PM11	PM10Note

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	1

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	0	0	PM61	PM60

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 6; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note Port register of an internal connection pin between the MCU and IO-Link transceiver (For the pin settings, see **Table 2-1 Settings of Internal Connection Pins**).

Cautions 1. Be sure to set bits 0-2 and 7 of PM0, bits 0 to 5 of PM2, bits 2 to 7 of PM3, bit 5 of PM4, bits 2, 6, and 7 of PM5, bits 2 to 3 of PM6, bits 1 to 7 of PM11, bits 1 to 7 of PM12, bits 1 and 5 to 7 of PM14, and bits 4 to 7 of PM15 to "1".

2. Be sure to clear PM03, PM04, PM06, PM64, and PM66 to 0 after a reset release (see 2.2 Initial Setting of Unused Internal Pins in MCU).

8.4 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 of the TAU0 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.
If channel 0 of the TAU1 is set as a master channel, channel 1 or those that follow (channels 1, 2, 3) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example: If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

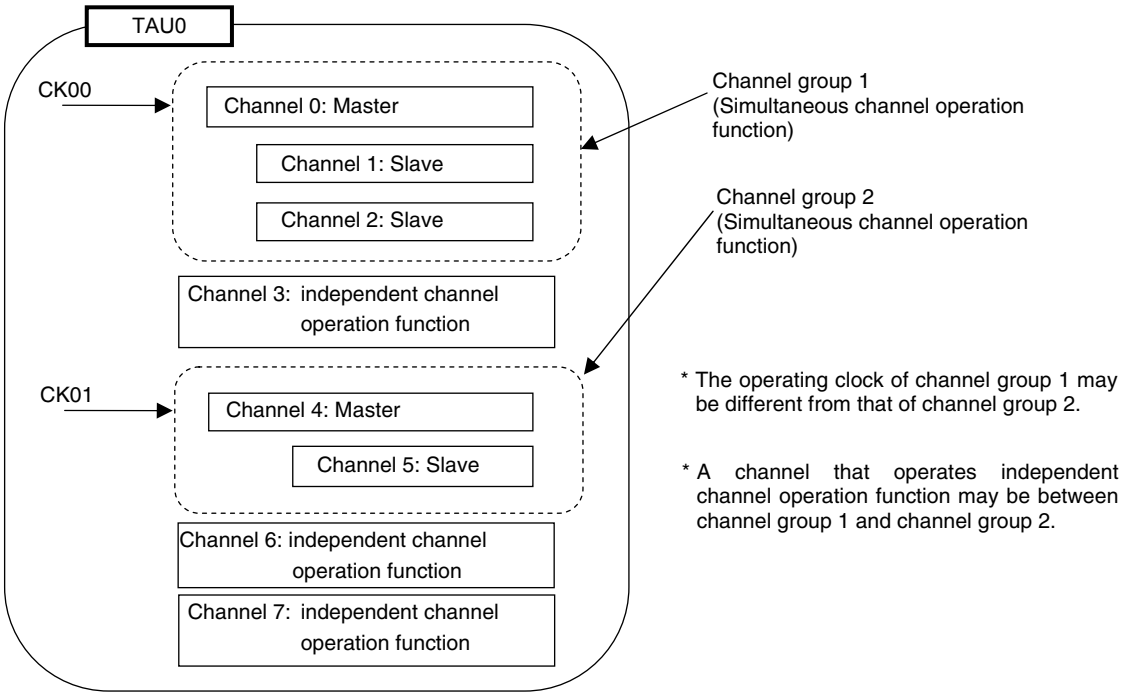
(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS bit (bit 15 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTMmn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TSmn) of the channels in combination must be set at the same time.
(11) During a counting operation, the TSmn bit of all channels that operate in combination or only the master channel can be set. TSmn of only a slave channel cannot be set.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 13

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 **8.4 Basic Rules of Simultaneous Channel Operation Function** do not apply to the channel groups.

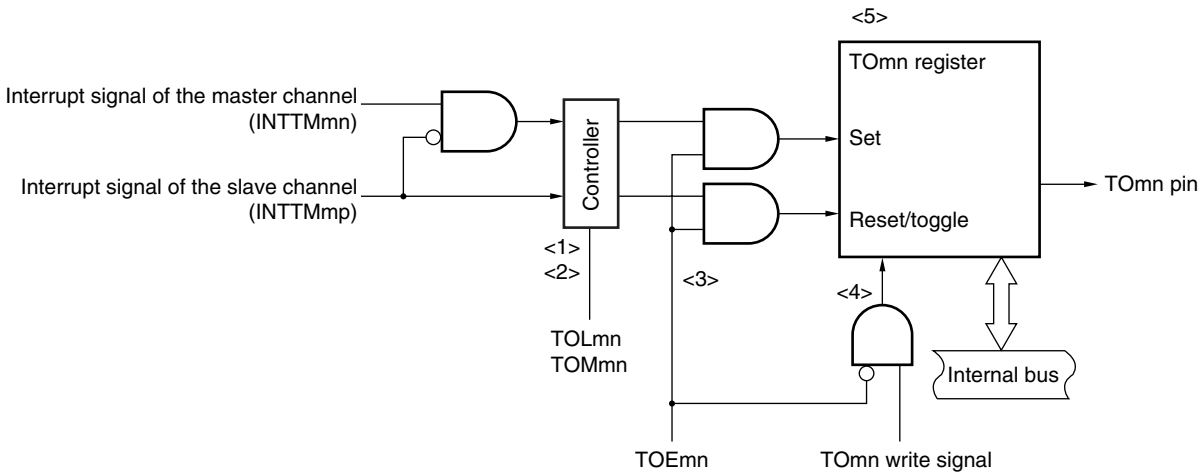
Example



8.5 Channel Output (TOmn pin) Control

8.5.1 TOmn pin output circuit configuration

Figure 8-26. Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTMmp (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:

When TOLmn = 0: Forward operation (INTTMmn → set, INTTMmp → reset)
When TOLmn = 1: Reverse operation (INTTMmn → reset, INTTMmp → set)

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

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation is stopeed (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabeled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabeled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTMmp (slave channel timer interrupt) is transmitted to the TOm register.

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

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

Remark m: Unit number, n: Channel number, p: Slave channel number

When m = 0

n = 0 to 7 (n = 1 to 3 and 5 in the case of the timer input pin (TI0n), n = 1 to 3 and 5 in the case of the timer output pin (TO0n); however, n = 0, 2, 4 in the case of the master channel)

n < p ≤ 5 (where p is a consecutive integer greater than n)

When m = 1

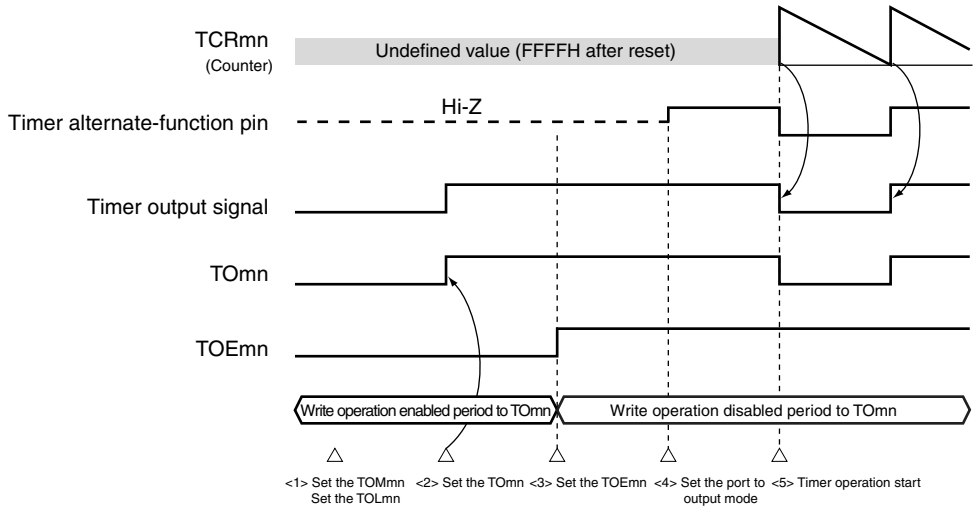
n = 0 to 3 (n = 1 and 3 in the case of the timer input pin (TI1n), n = 1 and 3 in the case of the timer output pin (TO1n); however, n = 0, 2 in the case of the master channel)

n < p ≤ 3 (where p is a consecutive integer greater than n)

8.5.2 TOmn Pin Output Setting

The following figure shows the procedure and status transition of TOmn out put pin from initial setting to timer operation start.

Figure 8-27. Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn bit (0: Forward output, 1: Reverse output)

<2> The timer output signal is set to the initial status by setting timer output register m (TOm).

<3> The timer output operation is enabled by writing 1 to the TOEmn bit (writing to the TOm register is disabled).

<4> The port I/O setting is set to output (see **8.3 (14) Port mode registers 0, 1, 3, 6 (PM0, PM1, PM3, PM6)**).

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

8.5.3 Cautions on Channel Output Operation

(1) Changing values set in registers TOm,TOEm,TOLm, and TOMm during timer operation

Since the timer operations (operations of timer/counter register mn (TCRmn) and timer data register mn (TDRmn)) are independent of the TOmn output circuit and changing the values set in timer output register m (TOm), timer output enable register m (TOEm), timer output level register m (TOLm), and timer output mode register m (TOMm) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOmn pin by timer operation, however, set the TOm, TOEm, TOLm, and TOMm registers to the values stated in the register setting example of each operation.

When the values set to the TOEm, TOLm, and TOMm registers (but not the TOm register) are changed close to the occurrence of the timer interrupt (INTTMmn) of each channel, the waveform output to the TOmn pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) occurs.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

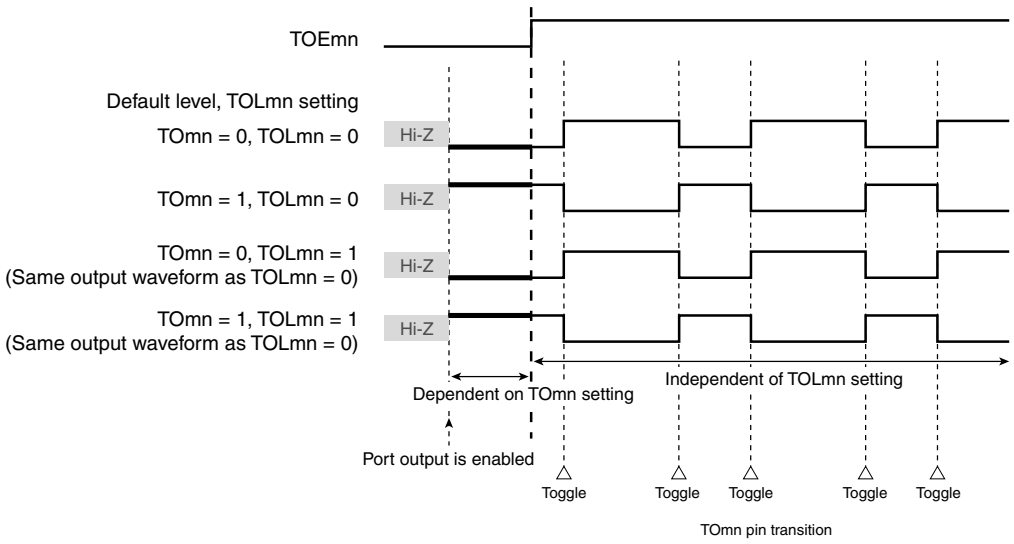
(2) Default level of TOmn pin and output level after timer operation start

The change in the output level of the TOmn pin when timer output register m (TOm) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOMmn = 0) setting

The setting of timer output level register m (TOLm) is invalid when master channel output mode (TOMmn = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TOmn pin is reversed.

Figure 8-28. TOmn Pin Output Status at Toggle Output (TOMmn = 0)



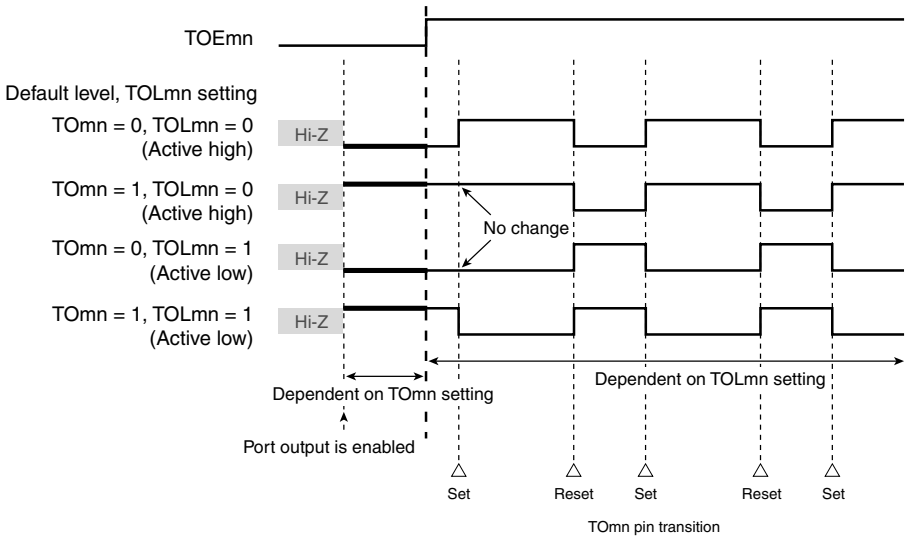
Remarks 1. Toggle: Reverse TOmn pin output status

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

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

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

Figure 8-29. TOmn Pin Output Status at PWM Output (TOMmn = 1)



Remarks 1. Set: The output signal of TOmn pin changes from inactive level to active level.
Reset: The output signal of TOmn pin changes from active level to inactive level.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

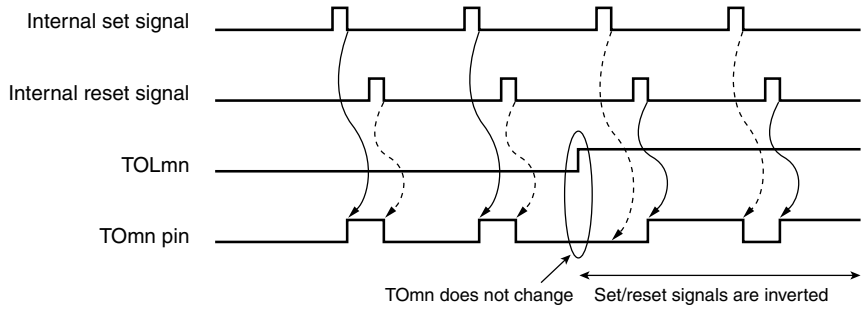
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(a) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOmn pin change condition. Rewriting the TOLm register does not change the output level of the TOmn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

Figure 8-30. Operation when TOLm Register Has Been Changed during Timer Operation



Remarks 1. Set: The output signal of TOmn pin changes from inactive level to active level.
Reset: The output signal of TOmn pin changes from active level to inactive level.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

(b) Set/reset timing

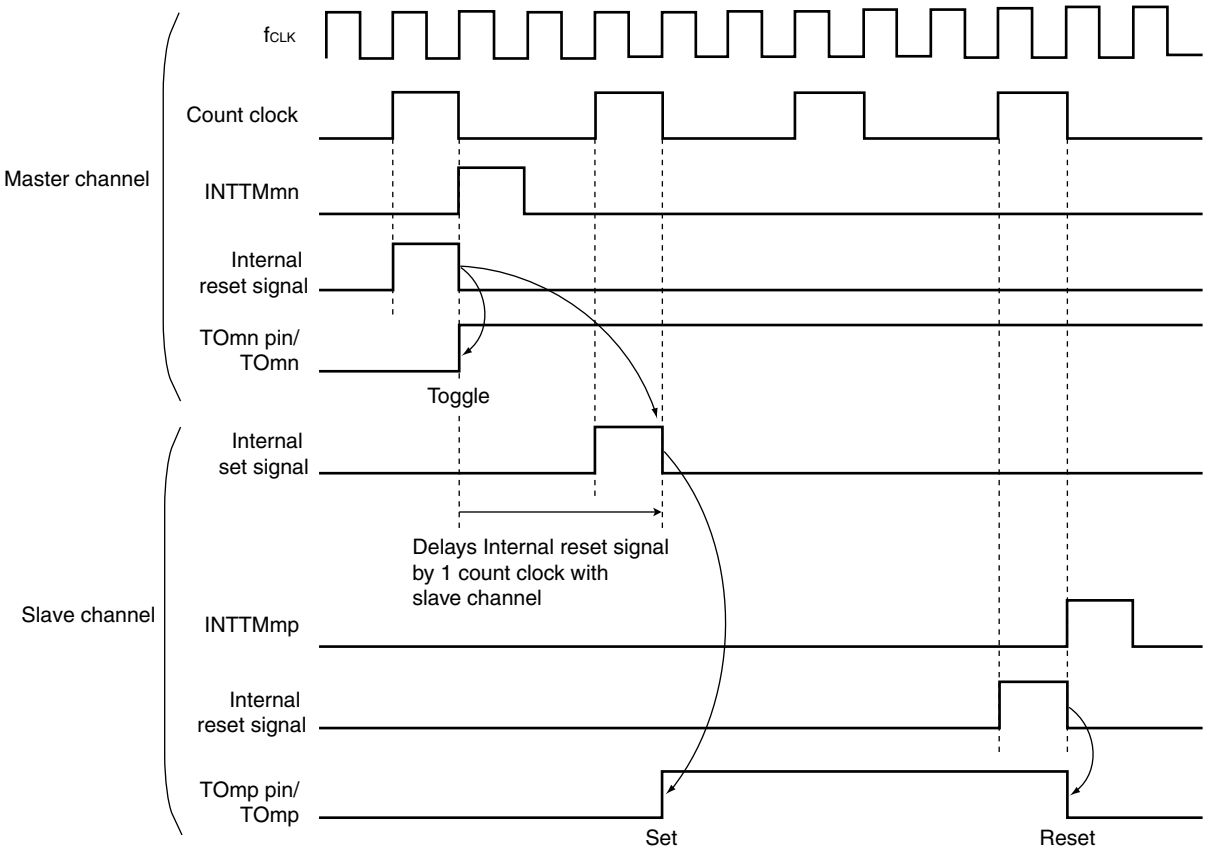
To realize 0%/100% output at PWM output, the TOmn pin/TOmn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

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

Figure 8-31 shows the set/reset operating statuses where the master/slave channels are set as follows.

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

Figure 8-31. Set/Reset Timing Operating Statuses



Remarks 1. Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

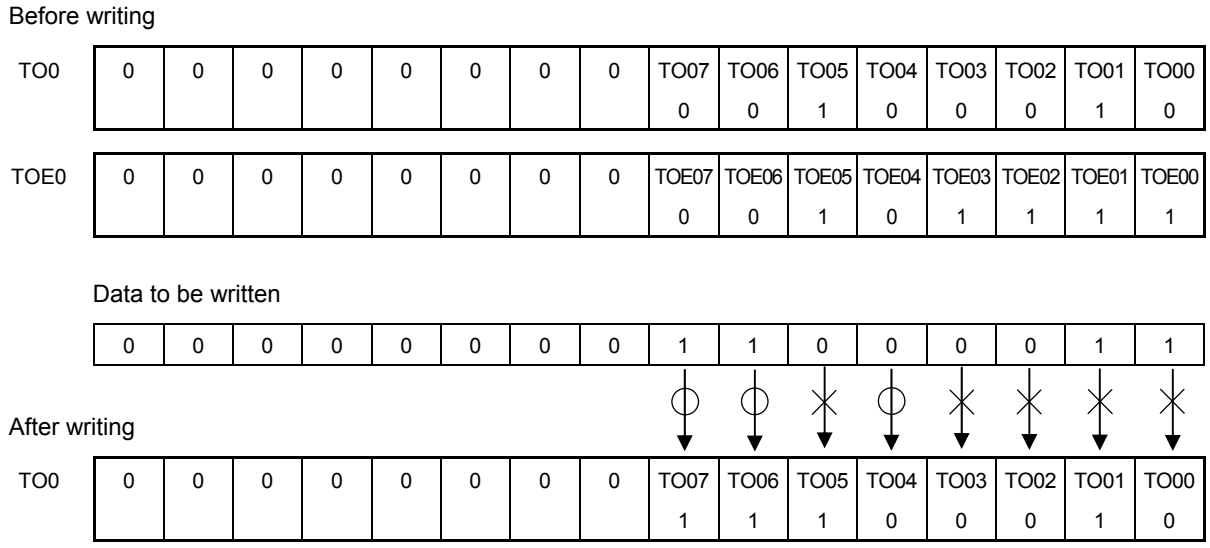
2. m: Unit number, n: Channel number, p: Slave channel number
When m = 0
n = 0 to 7 (n = 1 to 3 and 5 in the case of the timer output pin (TO0n); however, n = 0, 2, 4 in the case of the master channel)
n < p ≤ 7 (where p is a consecutive integer greater than n)
When m = 1
n = 0 to 3 (n = 1 to 3 in the case of the timer output pin (TO1n); however, n = 0, 2 in the case of the master channel)
n < p ≤ 3 (where p is a consecutive integer greater than n)

8.5.4 Collective manipulation of TOmn bits

In timer output register m (TOm), the setting bits for all the channels are located in one register in the same way as timer channel start register m (TSm). Therefore, the TOmn bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TOmn bits (TOEmn = 0) that correspond to the relevant bits of the channel used to perform output (TOmn).

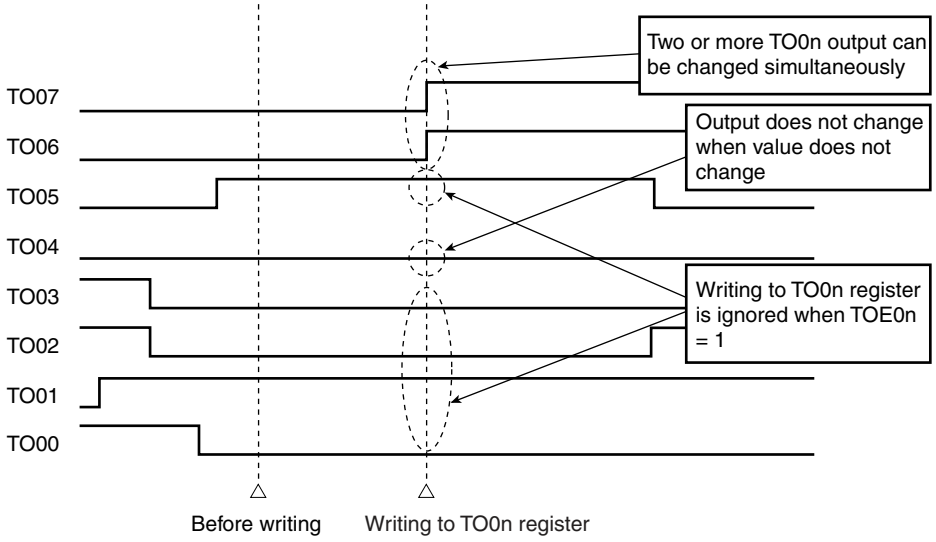
Figure 8-32. Example of TOmn Bit Collective Manipulation



Writing is done only to TOmn bits with TOEmn = 0, and writing to TOmn bits with TOEmn = 1 is ignored.

TOmn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to TOmn, it is ignored and the output change by timer operation is normally done.

Figure 8-33. TOmn Pin Statuses by Collective Manipulation of TOmn Bits



(**Caution** and **Remark** are given on the next page.)

Caution While timer output is enabled (TOEmn = 1), even if the output by timer interrupt of each timer (INTTMmn) contends with writing to the TOmn, output is normally done to the TOmn pin.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

8.5.5 Timer Interrupt and TOmn Pin Output at Operation Start

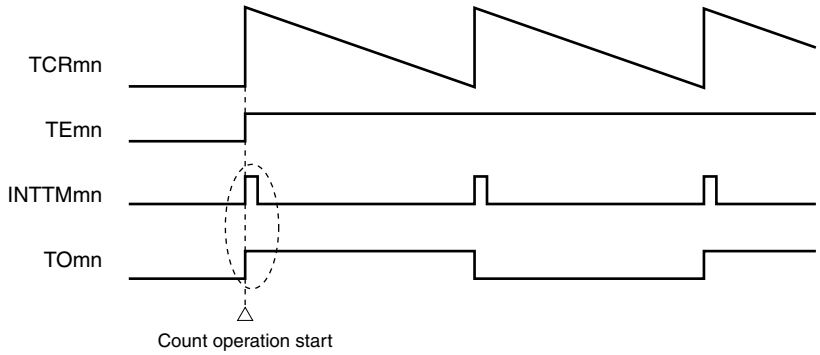
In the interval timer mode or capture mode, the MDmn0 bit in the timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

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

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

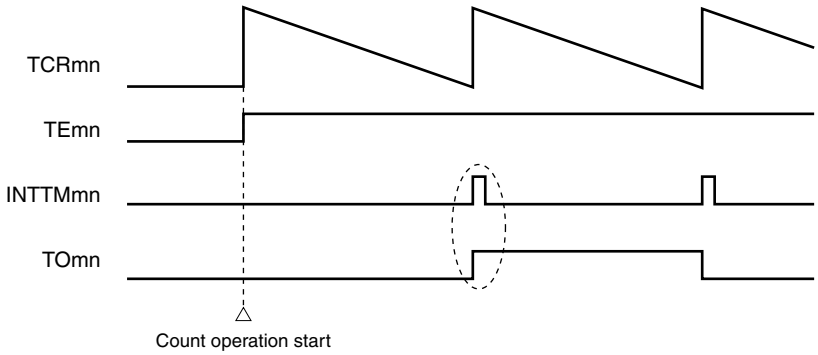
Figures 8-33 and 8-34 show operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 8-34. When MDmn0 is set to 1



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

Figure 8-35. When MDmn0 is set to 0



When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOmn does not change either. After counting one cycle, INTTMmn is output and TOmn performs a toggle operation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

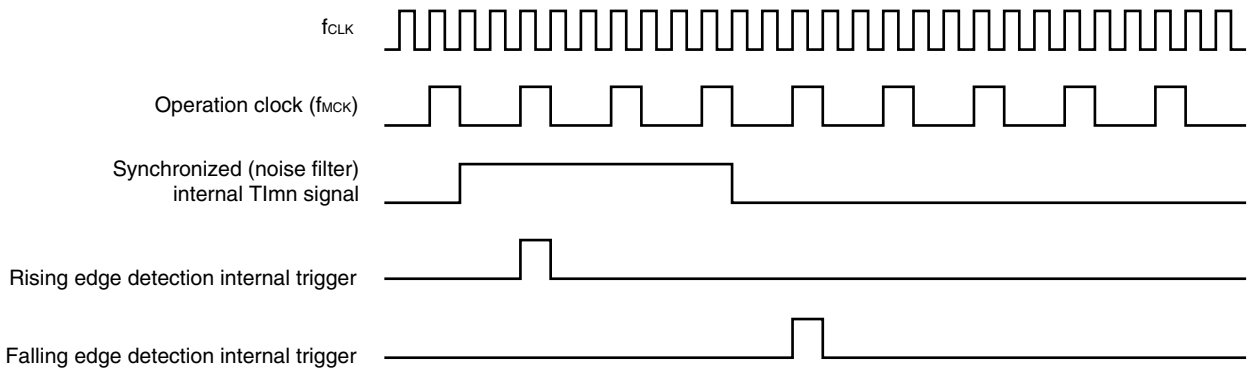
8.6 Channel Input (TImn Pin) Control

8.6.1 TImn edge detection circuit

(1) Edge detection basic operation timing

Edge detection circuit sampling is done in accordance with the operation clock (f_{MCK}).

Figure 8-36. Edge Detection Basic Operation Timing



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

8.7 Operation of Timer Array Unit as Independent Channel

8.7.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × Set value of (TDRmn) + 1

(2) Operation as square wave output

TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

• Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
• Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

Timer/counter register mn (TCRmn) operates as a down counter in the interval timer mode.

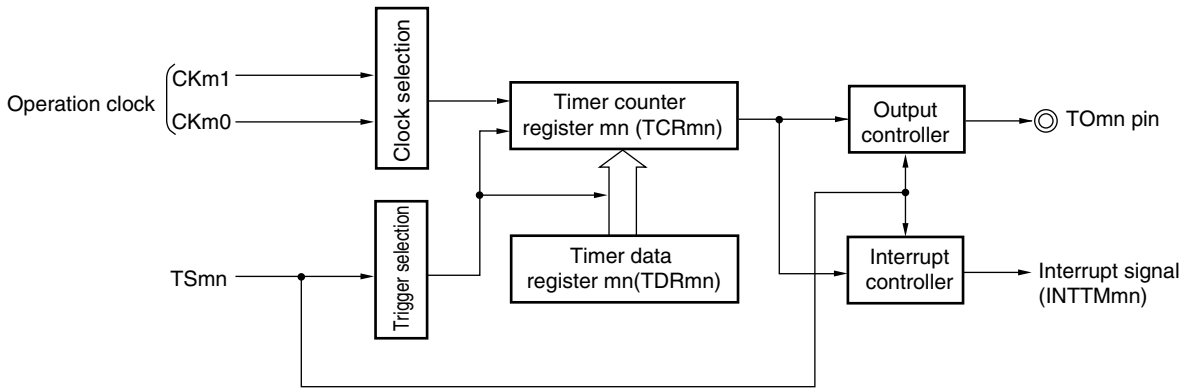
The TCRmn loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1. If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and TOmn is toggled.

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

When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, TCRmn loads the value of TDRmn again. After that, the same operation is repeated.

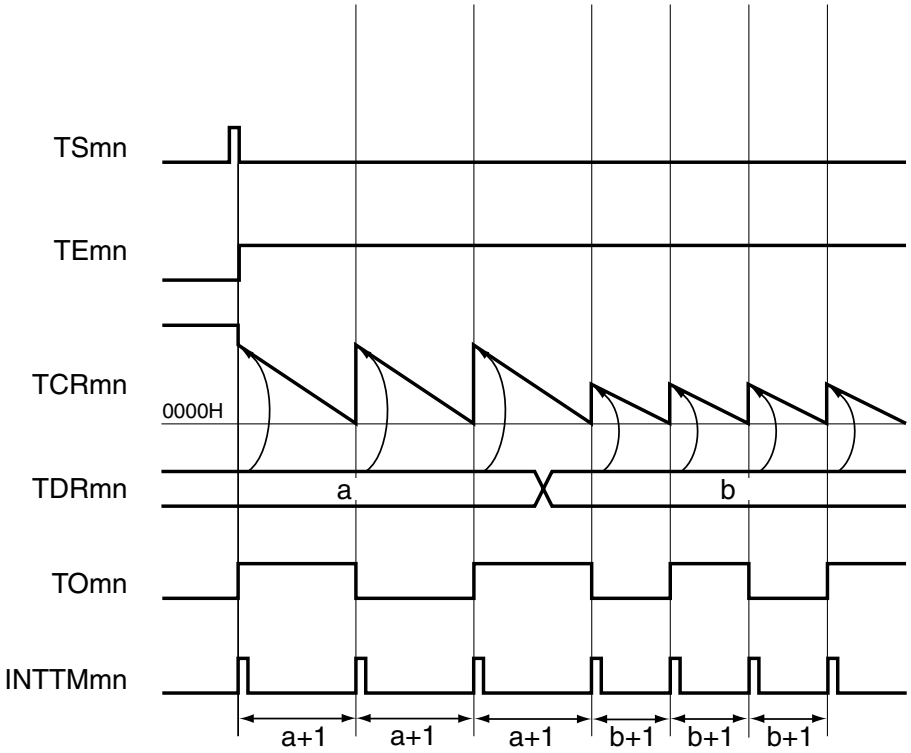
TDRmn can be rewritten at any time. The new value of TDRmn becomes valid from the next period.

Figure 8-37. Block Diagram of Operation as Interval Timer/Square Wave Output



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

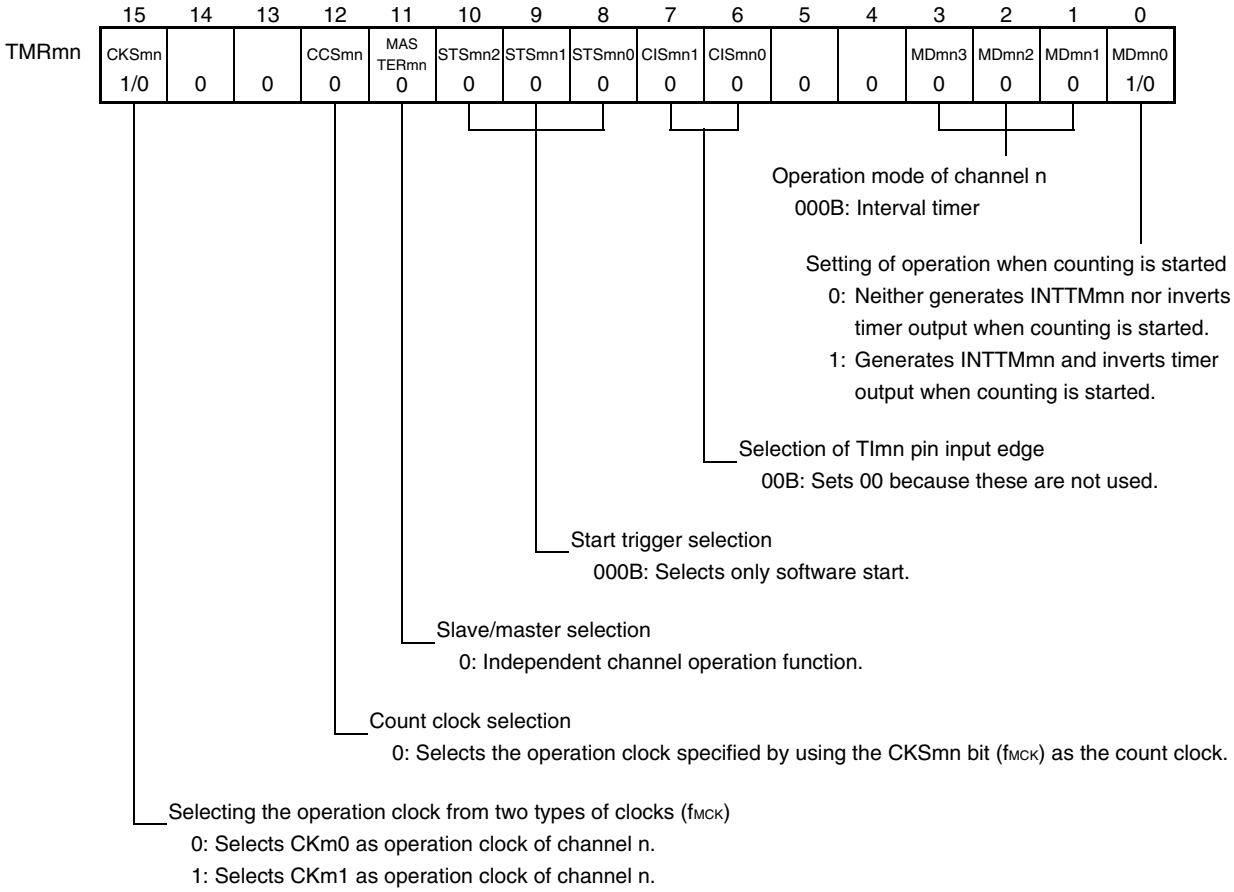
Figure 8-38. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

Figure 8-39. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 1/0	0: Outputs 0 from TOmn. 1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 1/0	0: Stops the TOmn output operation by counting operation. 1: Enables the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

Figure 8-40. 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 and TAU1EN bits of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOmn output Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. →	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets TOEmn bit to 1 and enables operation of TOmn. →	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmn pin outputs the TOmn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.). Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of the TDRmn register is loaded to timer/counter register mn (TCRmn) at the count clock input. INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set values of the TMRmn register, TOMmn and TOLmn bits cannot be changed. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn register holds count value and stops. The TOmn output is not initialized but holds current status.
	TOEmn bit is cleared to 0 and value is set to TOmn bit. →	The TOmn pin outputs the TOmn bit set level.

Operation is resumed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

Figure 8-40. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmn pin output level Clears TOmn bit to 0 after the value to be held is set to the port register. →	The TOmn pin output level is held by port function.
	When holding the TOmn pin output level is not necessary Switches the port mode register to input mode. →	The TOmn pin output level goes into Hi-Z output state.
	The TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode.)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmn).)

8.7.2 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 TImn pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDRmn + 1

Timer/counter register mn (TCRmn) operates as a down counter in the event counter mode.

The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn) of timer channel start register m (TSm) to 1.

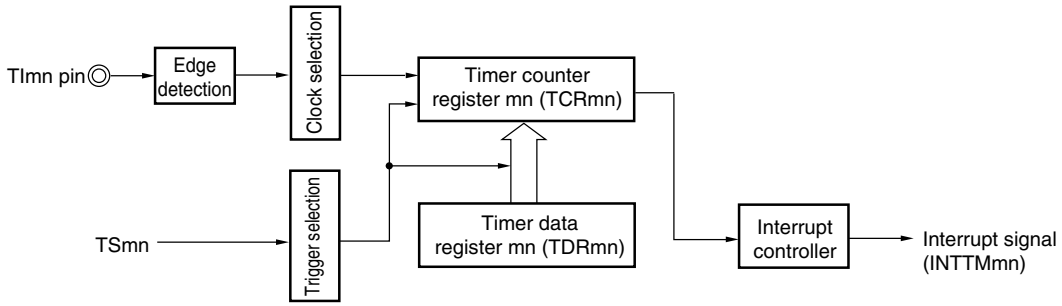
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TOmn pin. Stop the output by setting the TOEmn bit of timer output enable register m (TOEm) to 0.

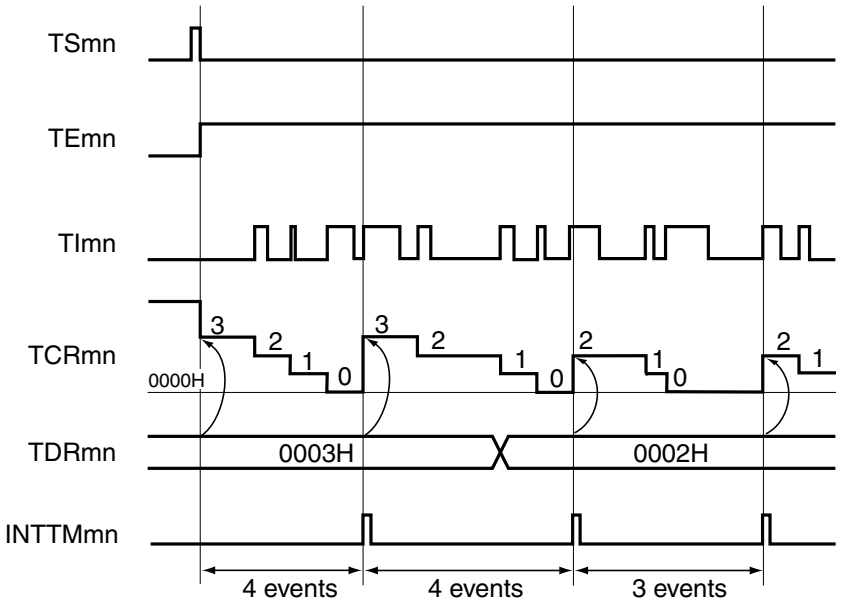
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 8-41. Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

Figure 8-42. Example of Basic Timing of Operation as External Event Counter

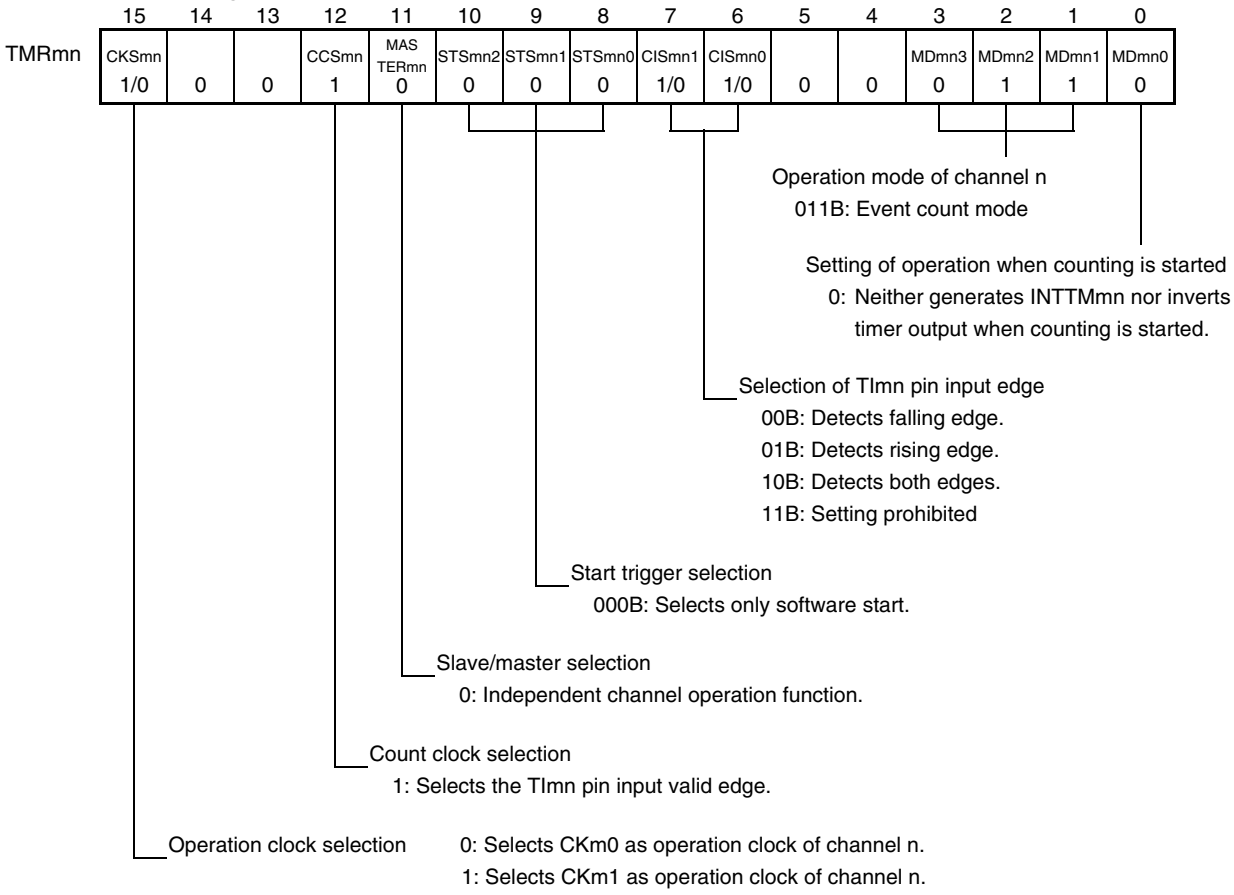


Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

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

Figure 8-43. Example of Set Contents of Registers in External Event Counter Mode

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops the TOmn output operation by counting operation.
	0	

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn	0: Cleared to 0 when TOMmn = 0 (master channel output mode).
	0	

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

Figure 8-44. 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 and TAU1EN bits of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets number of counts to timer data register mn (TDRmn). Clears the TOEmn bit of timer output enable register m (TOEm) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start ← Operation is resumed.	Sets the TSmn bit to 1. →	TEmn = 1, and count operation starts. Value of timer/counter register mn (TCRmn) is loaded to TCRmn register and detection of the TImn pin input edge is awaited.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of TMRmn register, TOMm, TOLm, TOm, and TOEm bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of TDRmn register is loaded to TCRmn register again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. →	TEmn = 0, and count operation stops. TCRmn register holds count value and stops.
TAU stop	The TAU0EN and TAU1EN bits of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

8.7.3 Operation as input pulse interval measurement

The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured.

The pulse interval can be calculated by the following expression.

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

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of the timer mode register mn (TMRmn), so an error equal to the number of operating clocks occurs.

The timer/counter register mn (TCRmn) operates as an up counter in the capture mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, TCRmn register counts up from 0000H in synchronization with the count clock.

When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

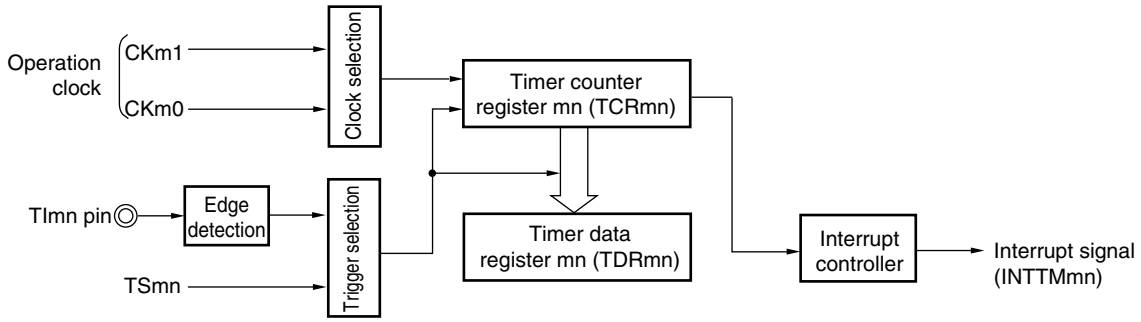
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, the OVF bit is configured as a cumulative flag, the correct interval value cannot be measured if an overflow occurs more than once.

Set STSmn2 to STSmn0 of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

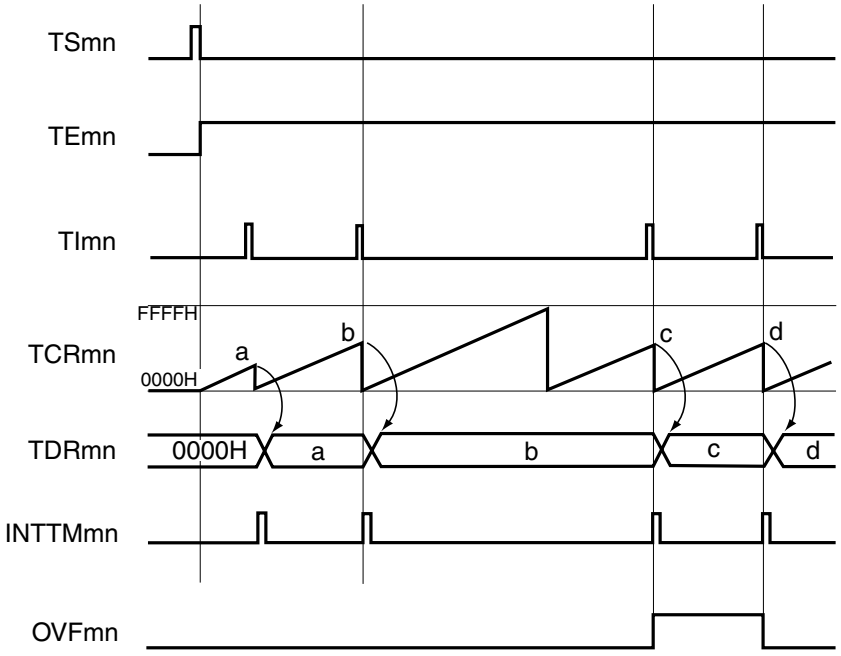
When TEmn = 1, instead of the TImn pin input, a software operation (TSmn = 1) can be used as a capture trigger.

Figure 8-45. Block Diagram of Operation as Input Pulse Interval Measurement



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

Figure 8-46. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)

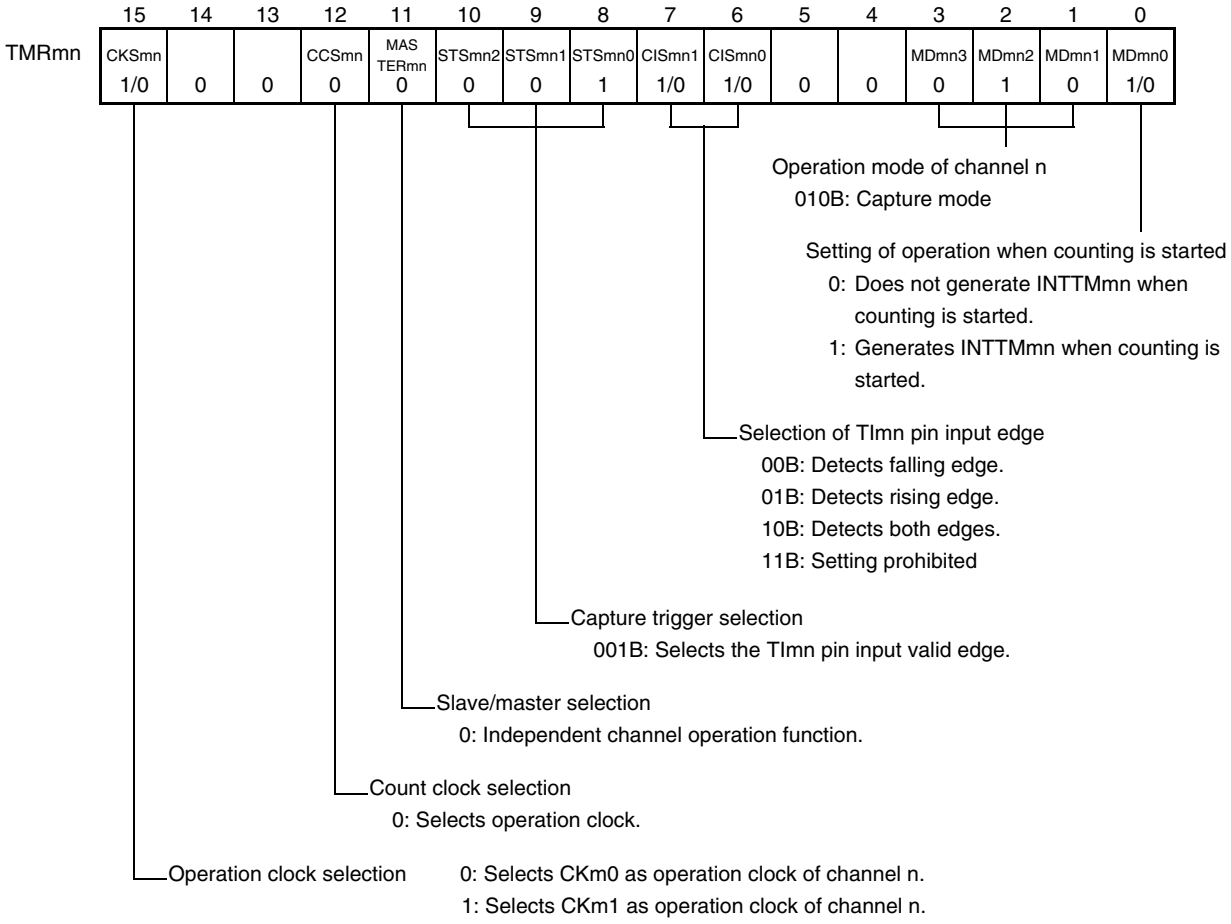


Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

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

Figure 8-47. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops TOmn output operation by counting operation.
	0	

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn	0: Cleared to 0 when TOMmn = 0 (master channel output mode).
	0	

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

Figure 8-48. 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, TAU1EN bit of the peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Timer/counter register mn (TCRmn) is cleared to 0000H at the count clock input. When the MDmn0 bit of the TMRmn register is 1, INTTMmn is generated.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of TOMm, TOLm, TOm, and TOEm bits cannot be changed.	Counter (TCRmn) counts up from 0000H. When the TImn pin input valid edge is detected, the count value is transferred (captured) to timer data register mn (TDRmn). At the same time, the TCRmn register is cleared to 0000H, and the INTTMmn signal is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAU0EN and TAU1EN bits of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

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

Caution When using a channel to implement the LIN-bus, set bit 1 (ISC1) of the input switch control register (ISC) to 1. In the following descriptions, read TImn as RxD3.

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

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

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of the TMRmn register, so an error equal equivalent to one operation clock occurs.

Timer/counter register mn (TCRmn) operates as an up counter in the capture & one-count mode.

When the channel start trigger (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.

When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value “value transferred to the TDRmn register + 1”, and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, the OVF bit is configured as an integral flag, and the correct interval value cannot be measured if an overflow occurs more than once.

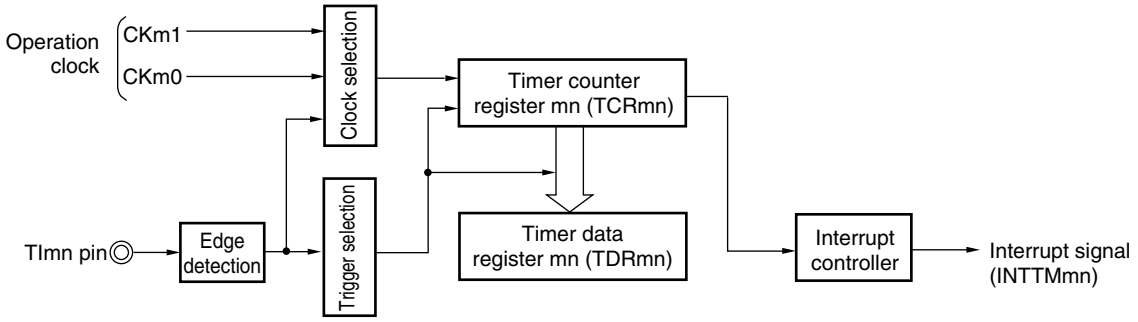
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

Because this function is used to measure the signal width of the TImn pin input, TSmn cannot be set to 1 while TEmn is 1.

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

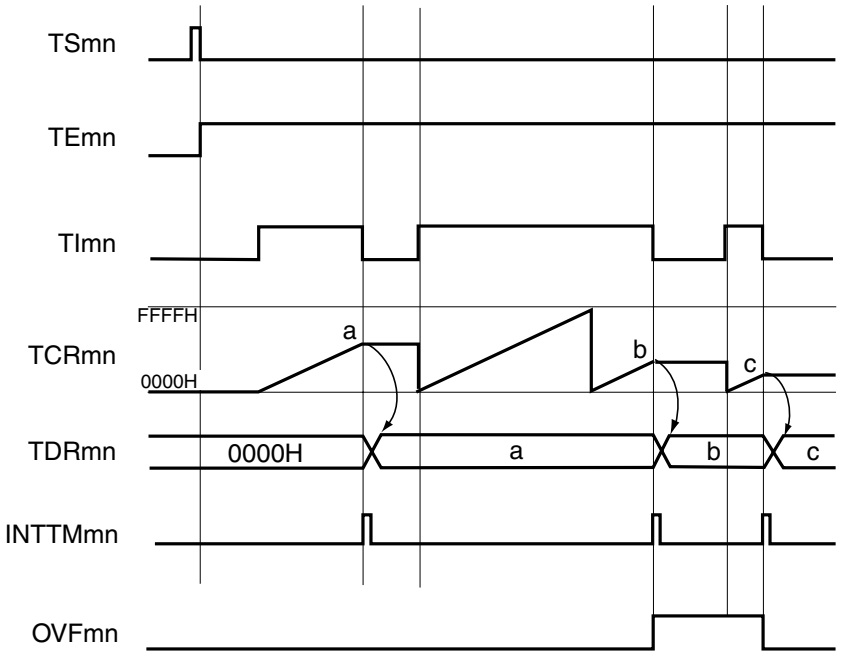
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

Figure 8-49. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

Figure 8-50. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement

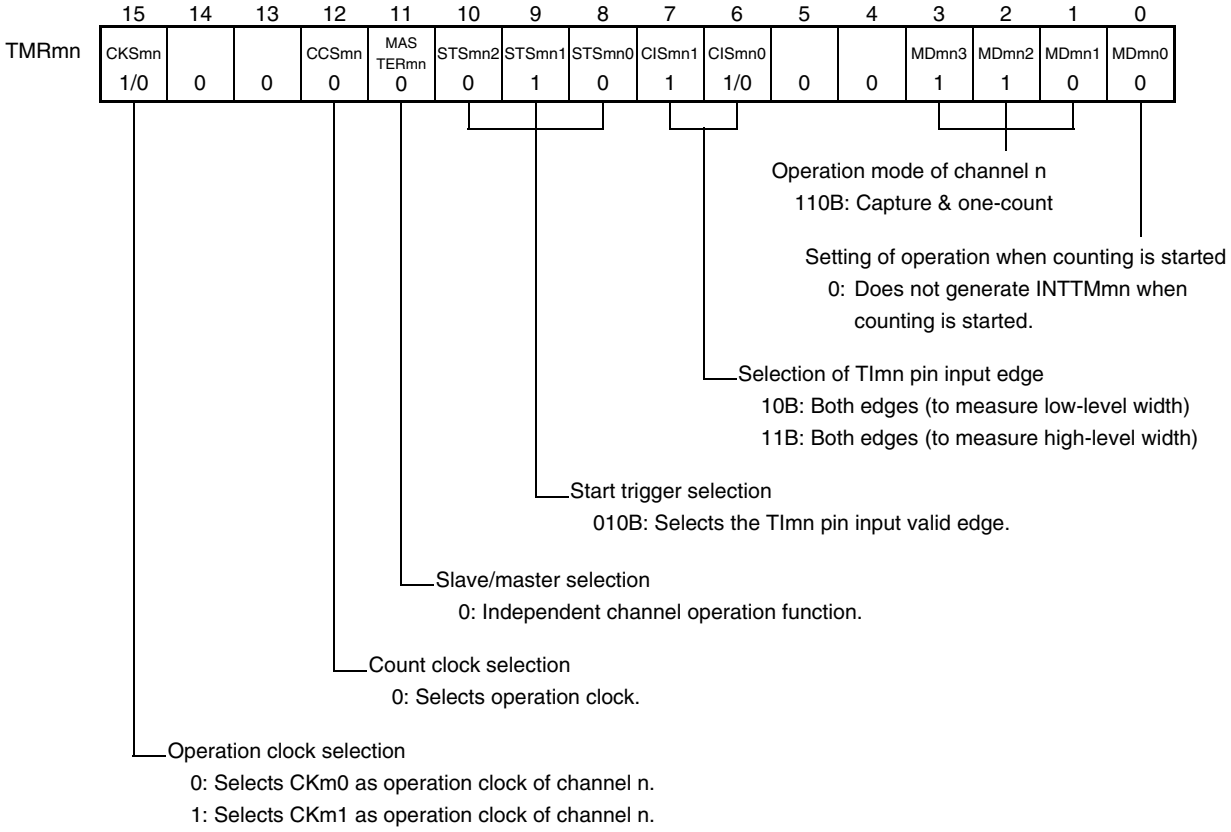


Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

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

Figure 8-51. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

Figure 8-52. 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 and TAU1EN bits of peripheral enable registers 0, 2 (PER0, PER2) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects TImn pin input count start valid edge. →	Clears timer/counter register mn (TCRmn) to 0000H and starts counting up.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected.
Operation stop	The TTmn bit is set to 1. → TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAU0EN and TAU1EN bits of PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 13 (This is, however, mn = 01 to 03, 05, 11, and 13 in the case of the timer input pin (TImn).)

8.8 Simultaneous Channel Operation Function of Timer Array Unit

8.8.1 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 TImn pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDRmn (master) + 2} × Count clock period
Pulse width = {Set value of TDRmp (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer/counter register mn (TCRmn) of the master channel starts operating upon start trigger detection and loads the value of timer data register mn (TDRmn).

The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

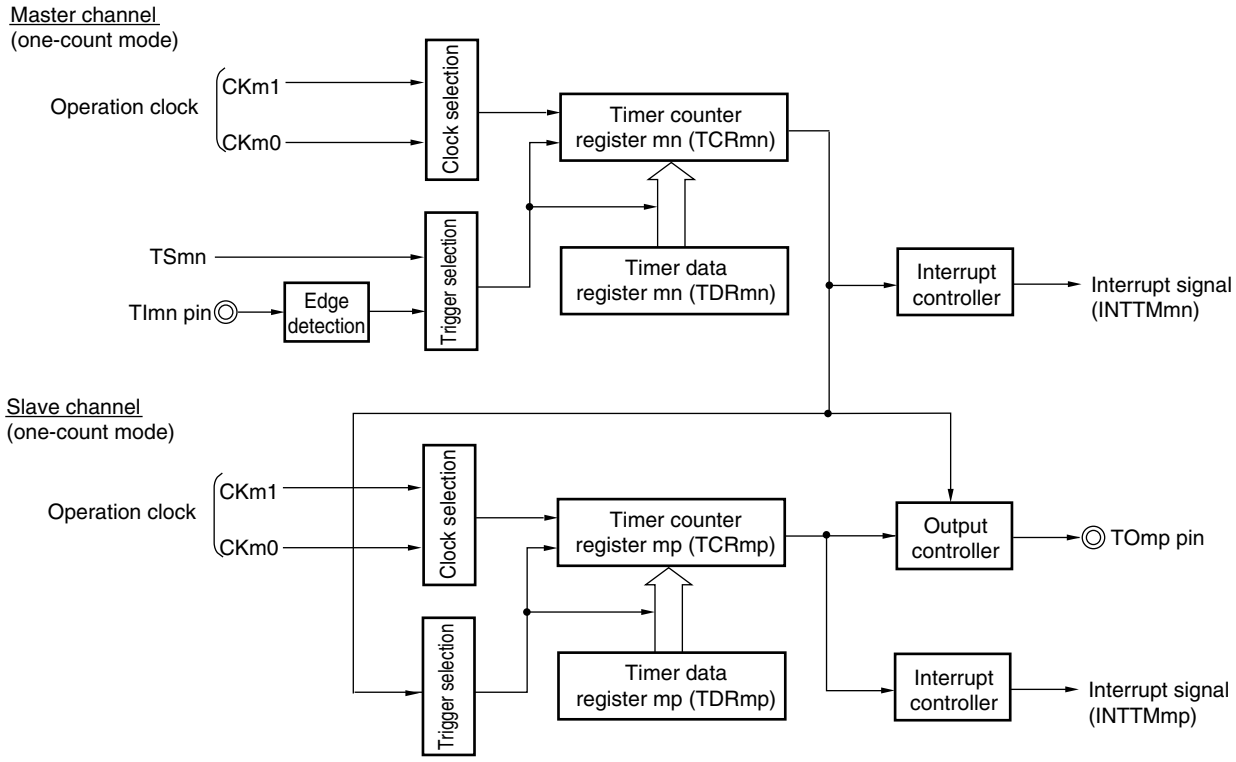
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution The timing of loading of timer data register mn (TDRmn) of the master channel is different from that of the TDRmp register of the slave channel. If the TDRmn and TDRmp registers are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDRmn register after INTTMmn is generated and the TDRmp register after INTTMmp is generated.

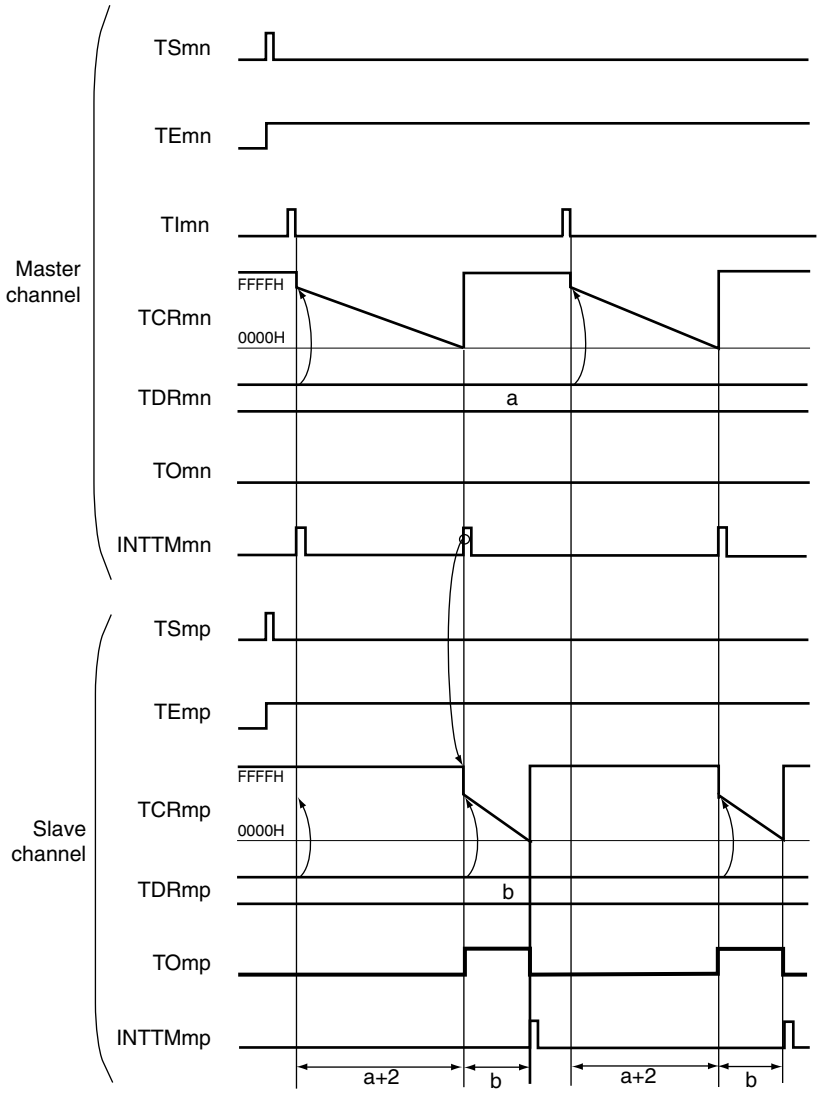
Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1)
When m = 0: n = 0, 2, 4
When m = 1: n = 0, 2
However, mn = 02 in the case of the timer input pin (TImp), mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

Figure 8-53. Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1)
When m = 0: n = 0, 2, 4
When m = 1: n = 0, 2
However, mn = 02 in the case of the timer input pin (TImp), mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

Figure 8-54. Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remarks 1. m: Unit number, n: Channel number, p: Slave channel number (p = n+1)

When m = 0: n = 0, 2, 4

When m = 1: n = 0, 2

However, mn = 02 in the case of the timer input pin (TImp), mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

2. TSmn, TSmp: Bit n, m of timer channel start register m (TSm)

TEmn, TEmp: Bit n, m of timer channel enable status register m (TEm)

TImn, TImp: TImn and TImp pins input signal

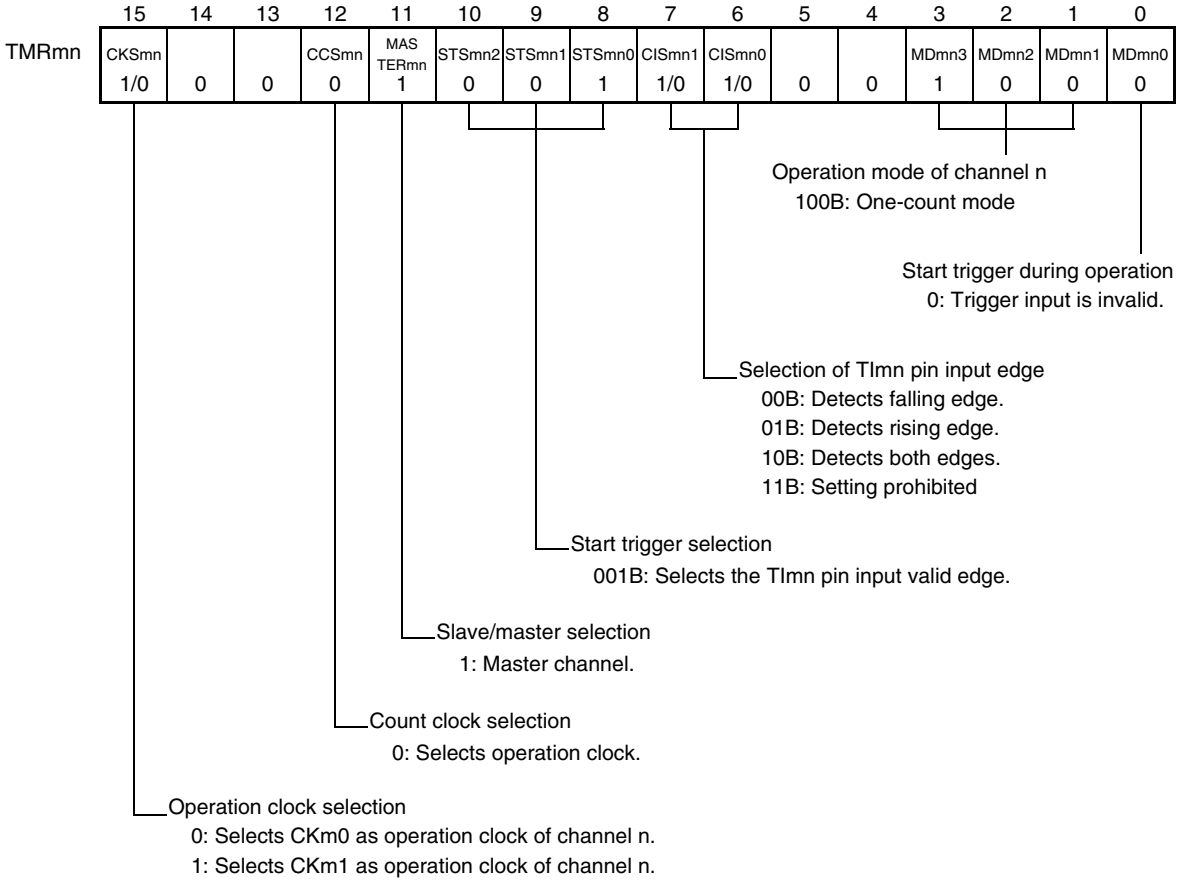
TCRmn, TCRmp: Timer/counter registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)

TOmn, TOmp: TOmn and TOmp pins output signal

Figure 8-55. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number, n: Channel number
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0, 2
However, mn = 02 in the case of the timer input pin (TImp)

Figure 8-56. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)

TMRmp

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKSmp			CCSmp	MAS TERmp	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Operation mode of channel p
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Slave/master selection
0: Slave channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CKm0 as operation clock of channel p.
1: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

TOm

Bit p
TOmp
1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

TOEm

Bit p
TOEmp
1/0

0: Stops the TOmp output operation by counting operation.
1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

TOLm

Bit p
TOLmp
1/0

0: Positive logic output (active-high)
1: Inverted output (active-low)

(e) Timer output mode register m (TOMm)

TOMm

Bit p
TOMmp
1

1: Sets the slave channel output mode.

Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1)
When m = 0: n = 0, 2, 4
When m = 1: n = 0, 2
However, mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

Figure 8-57. 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 and TAU1EN bits of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register mn (TDRmn) of the master channel, and a pulse width is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmn pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEmp to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

Remark m: Unit number, n: Muster Channel number, p: Slave channel number (p = n+1)
When m = 0: n = 0, 2, 4
When m = 1: n = 0, 2
However, mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

Figure 8-57. Operation Procedure of One-Shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
	Detects the TImn pin input valid edge of master channel.	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers can be changed.	Master channel loads the value of the TDRmn register to timer/counter register mn (TCRmn) when the TImn pin valid input edge is detected, and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next valid edge is input to the TImn pin. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. TCRmn and TCRmp hold count value and stops. The TOmp output is not initialized but holds current status.
	TOEmp of slave channel is cleared to 0 and value is set to the TOmp bit.	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output levels Clears TOmp bit to 0 after the value to be held is set to the port register.	The TOmp pin output levels is held by port function.
	When holding the TOmp pin output levels is not necessary Switches the port mode register to input mode.	The TOmp pin output levels go are into Hi-Z output state.
	The TAU0EN and TAU1EN bits of the PER0 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1)
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0, 2
However, mn = 02 in the case of the timer input pin (TImp), mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

8.8.2 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 TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

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

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer/counter register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTm) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.

The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn from the master channel) is generated.

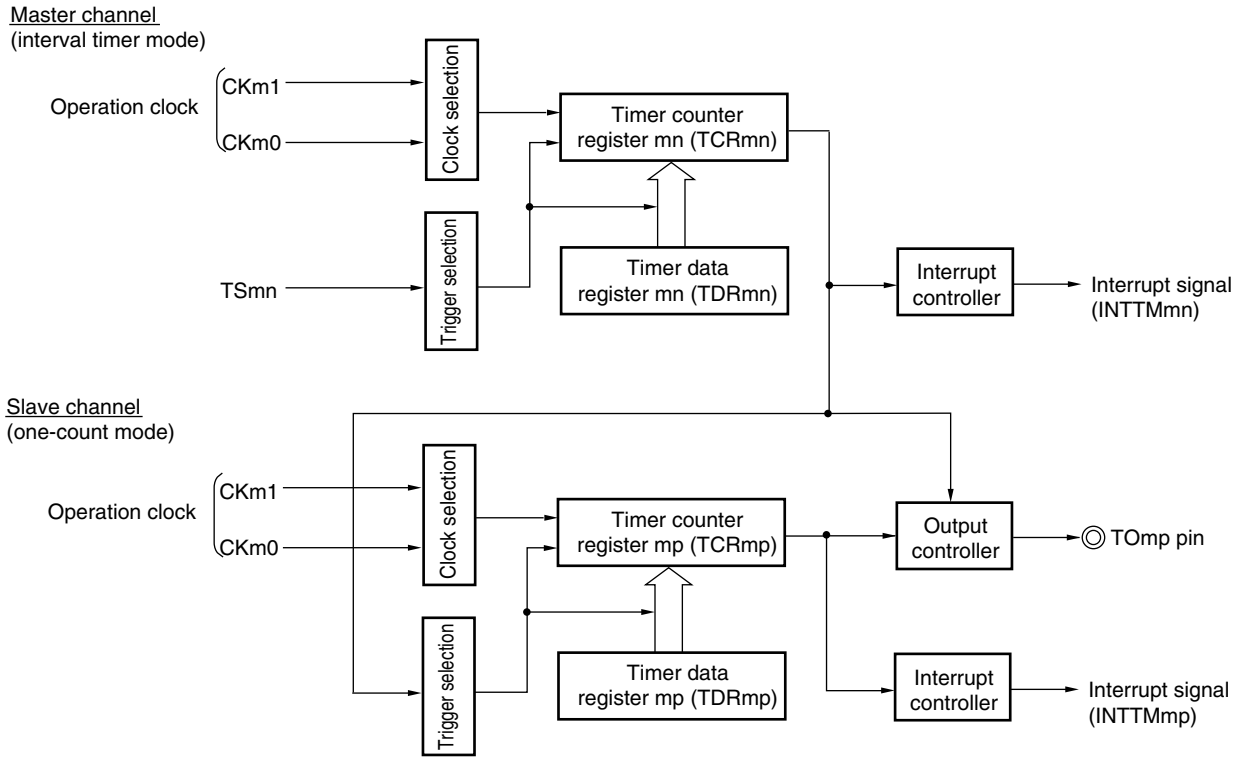
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.

PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

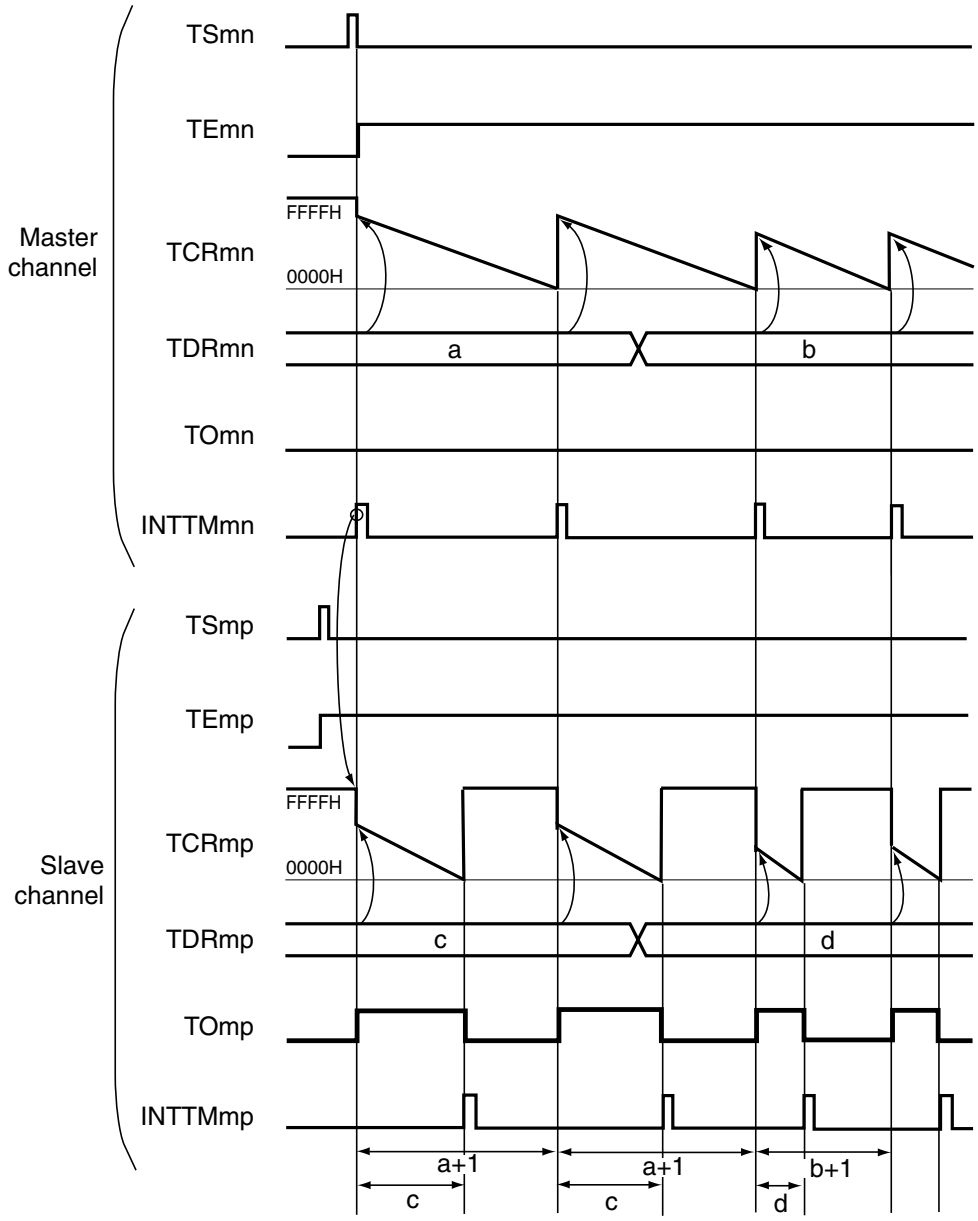
Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1)
When m = 0: n = 0, 2, 4
When m = 1: n = 0, 2
However, mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

Figure 8-58. Block Diagram of Operation as PWM Function



Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1)
When m = 0: n = 0, 2, 4
When m = 1: n = 0, 2
However, mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

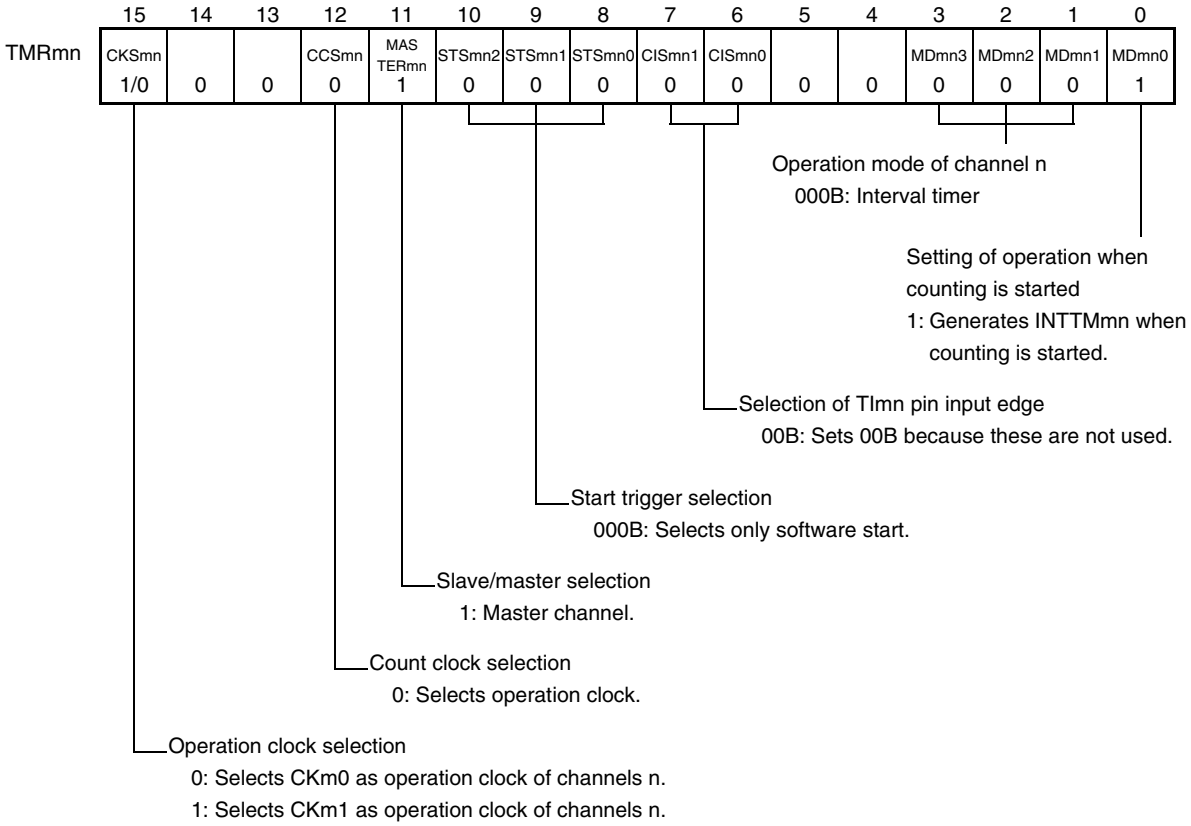
Figure 8-59. Example of Basic Timing of Operation as PWM Function



Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1)
When m = 0: n = 0, 2, 4
When m = 1: n = 0, 2
However, mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

Figure 8-60. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops the TOmn output operation by counting operation.
	0	

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn	0: Cleared to 0 when TOMmn = 0 (master channel output mode).
	0	

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1),
When m = 0: n = 0, 2, 4
When m = 1: n = 0, 2

Figure 8-61. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp			CCSmp	MAS TERmp	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Slave/master selection
0: Slave channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CKm0 as operation clock of channel p.
1: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p
TOm	TOmp
	1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p
TOEm	TOEmp
	1/0

0: Stops the TOmp output operation by counting operation.
1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p
TOLm	TOLmp
	1/0

0: Positive logic output (active-high)
1: Inverted output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p
TOMm	TOMmp
	1

1: Sets the slave channel output mode.

Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1),
When m = 0: n = 0, 2, 4, When m = 1: n = 0, 2
However, mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

Figure 8-62. 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.)
	Set the TAU0EN and TAU1EN bits of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmn pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

Remark m: Unit number, n: Muster Channel number, p: Slave channel number (p = n+1),
When m = 0: n = 0, 2, 4
When m = 1: n = 0, 2
However, mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

Figure 8-62. Operation Procedure of One-Shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. →	TEmn = 1, TEmp = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
	The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	
During operation	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers can be changed.	The counter of the master channel loads the TDRmn register value to timer/counter register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. →	TEmn, TEmp = 0, and count operation stops. TCRmn and TCRmp register hold count value and stops. The TOmp output is not initialized but holds current status.
	TOEmp of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output levels Clears TOmp bit to 0 after the value to be held is set to the port register. →	The TOmp pin output levels is held by port function.
	When holding the TOmp pin output levels is not necessary Switches the port mode register to input mode. →	The TOmp pin output levels go are into Hi-Z output state.
	The TAU0EN and TAU1EN bits of the PER0 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1),
When m = 0: n = 0, 2, 4
When m = 1: n = 0, 2
However, mp = 01 to 03, 05, 11, and 13 in the case of the timer output pin (TOmp).

8.8.3 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 TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer/counter register mn (TCRmn) of the master channel operates in the interval timer mode and counts the periods.

The TCRmp register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.

When channel 0 is used as the master channel as above, as for the timer array unit 0, up to seven types of PWM signals can be generated, while as for the timer array unit 1, up to three types of PWM signals can be generated.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).

Remark m: Unit number, n: Master channel number, p: Slave channel number 1 (n+1), q: Slave channel number 2 (n+2)

When m = 0

n = 0, 2, 4 (This is, however, p = 1 to 3 and 5 in the case of the timer output pin (TO0p), q = 1 to 3 and 5 in the case of the timer output pin (TO0q).)

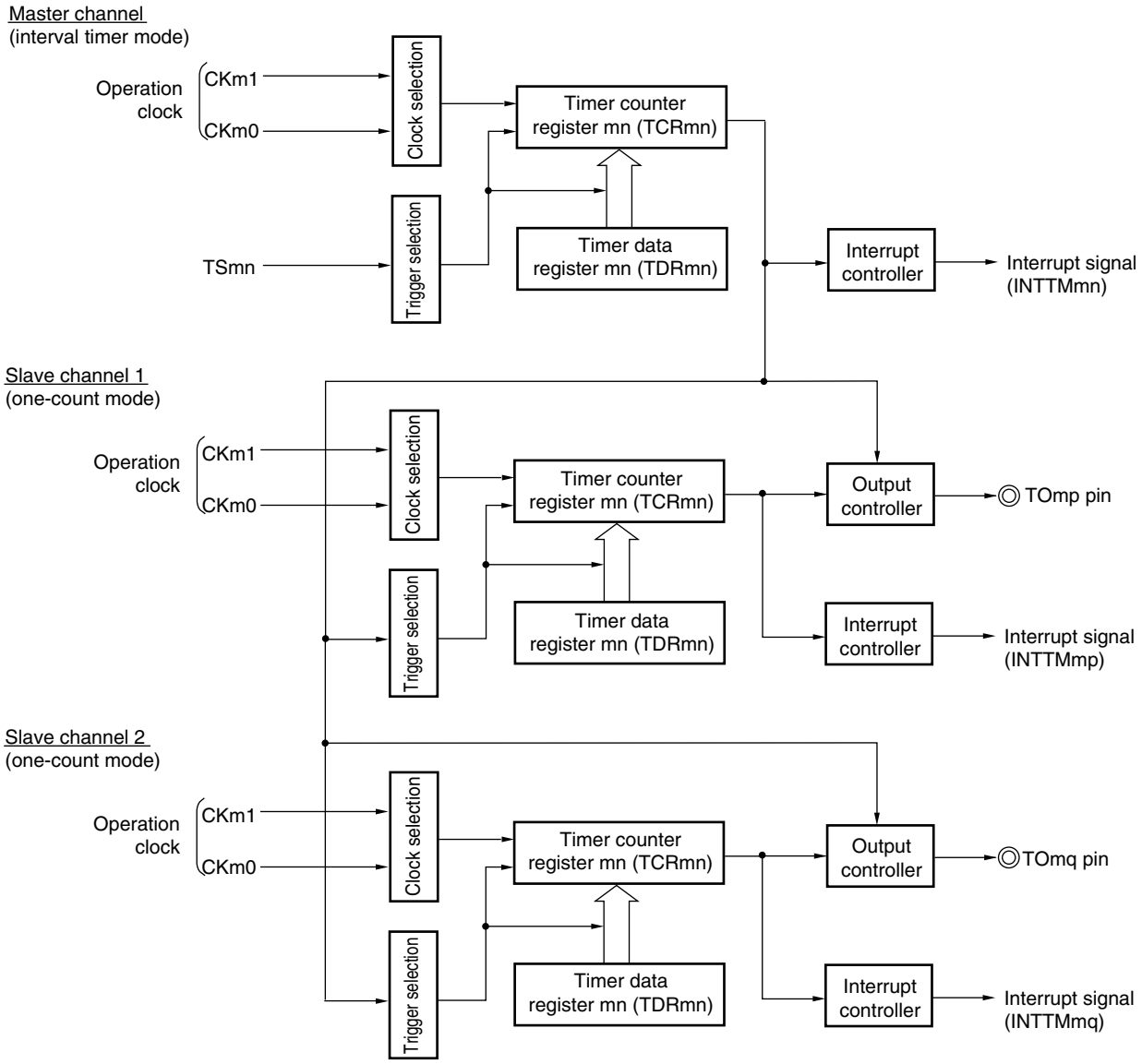
n < p < q ≤ 7 (where p and q are a consecutive integer greater than n)

When m = 1

n = 0

n < p < q ≤ 3 (where p and q are a consecutive integer greater than n)

Figure 8-63. Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)



Remark m: Unit number, n: Channel number, p: Slave channel number 1 (n+1), q: Slave channel number 2 (n+2)

When m = 0

n = 0, 2, 4 (This is, however, p = 1 to 3 and 5 in the case of the timer output pin (TO0p), q = 1 to 3 and 5 in the case of the timer output pin (TO0q).)

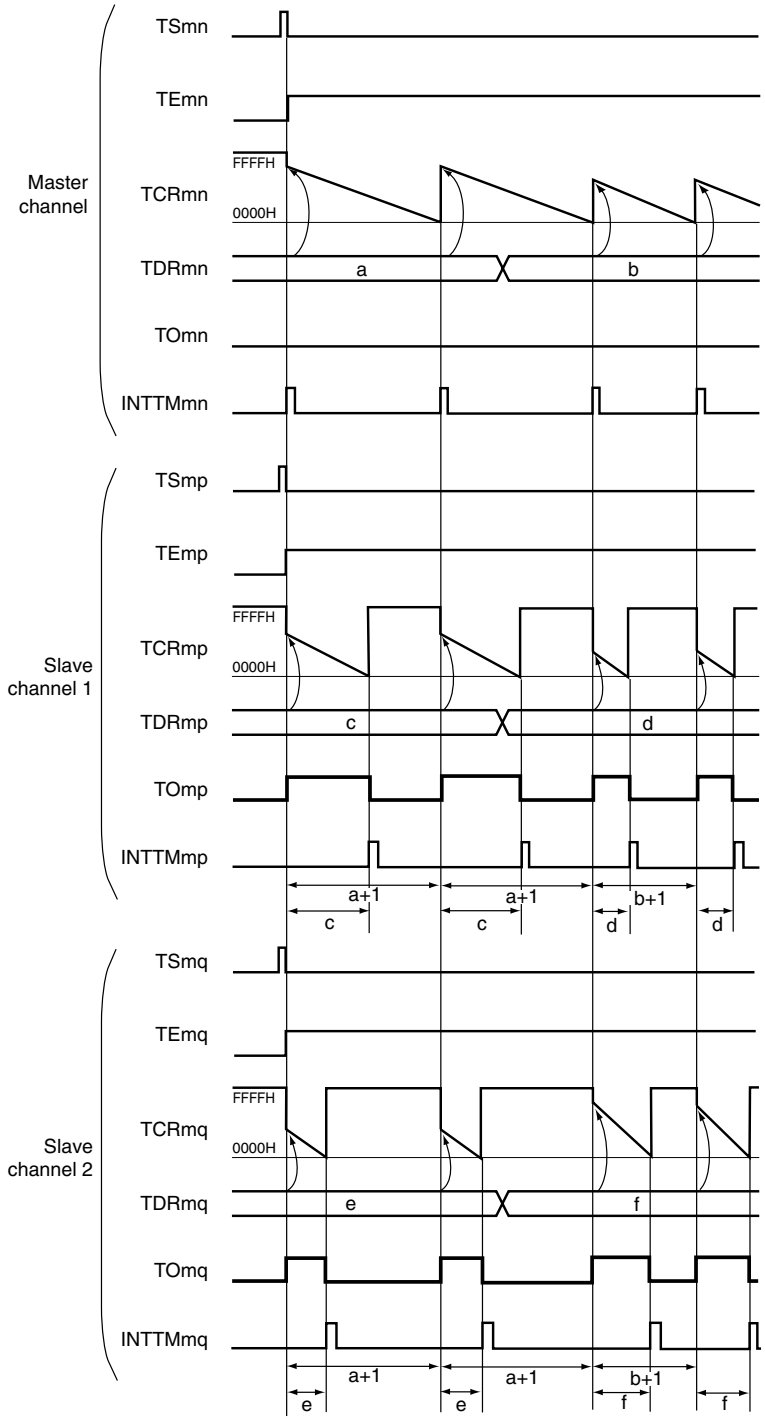
n < p < q ≤ 7 (where p and q are a consecutive integer greater than n)

When m = 1

n = 0

n < p < q ≤ 3 (where p and q are a consecutive integer greater than n)

Figure 8-64. Example of Basic Timing of Operation as Multiple PWM Output Function (output two types of PWMs) (1/2)



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

Figure 8-64. Example of Basic Timing of Operation as Multiple PWM Output Function (output two types of PWMs) (2/2)

Remarks 1. m: Unit number, n: Channel number, p: Slave channel number 1 (n+1), q: Slave channel number 2 (n+2)

When m = 0

n = 0, 2, 4 (This is, however, p = 1 to 3 and 5 in the case of the timer output pin (TO0p), q = 1 to 3 and 5 in the case of the timer output pin (TO0q).)

n < p < q ≤ 7 (where p and q are a consecutive integer greater than n)

When m = 1

n = 0

n < p < q ≤ 3 (where p and q are a consecutive integer greater than n)

2. TSmn, TSmp, TSmq: Bit n, p, q of timer channel start register m (TSm)

TEmn, TEmp, TEmq: Bit n, p, q of timer channel enable status register m (TEm)

TCRmn, TCRmp, TCRmq: Timer/counter registers mn, mp, mq (TCRmn, TCRmp, TCRmq)

TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)

TOmn, TOmp, TOmq: TOmn, TOmp, and TOmq pins output signal

Figure 8-65. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn			CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Master channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CKm0 as operation clock of channel n.
1: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (toggle mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets toggle mode.

Remark m: Unit number (m = 0, 1), n: Channel number
When m = 0: n = 0, 2, 4
When m = 1: n = 0

Figure 8-66. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs) (1/2)

(a) Timer mode register mp, mq (TMRmp, TMRmq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp			CCSmp	MAS TERmp	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1
TMRmq	CKSmq			CCSmq	MAS TERmq	STSmq2	STSmq1	STSmq0	CISmq1	CISmq0			MDmq3	MDmq2	MDmq1	MDmq0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp and TImq pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Slave/master selection
0: Slave channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CKm0 as operation clock of channel p, q.
1: Selects CKm1 as operation clock of channel p, q.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit q	Bit p	
TOm	TOmq	TOmp	0: Outputs 0 from TOmp or TOmq. 1: Outputs 1 from TOmp or TOmq.
	1/0	1/0	

(c) Timer output enable register m (TOEm)

	Bit q	Bit p	
TOEm	TOEmq	TOEmp	0: Stops the TOmp or TOmq output operation by counting operation. 1: Enables the TOmp or TOmq output operation by counting operation.
	1/0	1/0	

(d) Timer output level register m (TOLm)

	Bit q	Bit p	
TOLm	TOLmq	TOLmp	0: Positive logic output (active-high) 1: Inverted output (active-low)
	1/0	1/0	

(e) Timer output mode register m (TOMm)

	Bit q	Bit p	
TOMm	TOMmq	TOMmp	1: Sets the combination-operation mode.
	1	1	

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

Figure 8-66. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs) (2/2)

Remark m: Unit number, n: Channel number, p: Slave channel number 1 (n+1), q: Slave channel number 2 (n+2)

When m = 0

n = 0, 2, 4 (This is, however, p = 1 to 3 and 5 in the case of the timer output pin (TO0p), q = 1 to 3 and 5 in the case of the timer output pin (TO0q).)

n < p < q ≤ 7 (where p and q are a consecutive integer greater than n)

When m = 1

n = 0

n < p < q ≤ 3 (where p and q are a consecutive integer greater than n)

Figure 8-67. 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 and TAU1EN bits of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp, mq (TMRmn, TMRmp, TMRmq) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Clears the TOLmp and TOLmq bits to 0. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmn pin goes into Hi-Z output state. The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. →	TOmp or TOmq does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.
Operation start	(Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp, TEmq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.

Operation is resumed (on the next page).

Figure 8-67. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

Operation is resumed (on the before page).

	Software Operation	Hardware Status
During operation	Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSRmq registers are not used. Set values of the TOm, and TOEm registers can be changed.	The counter of the master channel loads the TDRmn register value to timer/counter register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to TCRmn register, and the counter starts counting down again. At the slave channel 1, the values of TDRmp are transferred to TCRmp, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of TDRmq are transferred to TCRmq register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	TEmn, TEmp and TEmq = 0, and count operation stops. TCRmn, TCRmp and TCRmq hold count value and stops. The TOmp and TOmq output is not initialized but holds current status.
	TOEmp or TOEmq bits of slave channel is cleared to 0 and value is set to the TOmp and TOmq bits.	The TOmp and TOmq pins output the TOmp and TOmq set levels.
TAU stop	To hold the TOmp and TOmq pin output levels Clears TOmp and TOmq bits to 0 after the value to be held is set to the port register.	The TOmp and TOmq pin output levels are held by port function.
	When holding the TOmp and TOmq pin output levels is not necessary Switches the port mode register to input mode.	The TOmp and TOmq pin output levels go into Hi-Z output state.
	The TAU0EN and TAU1EN bits of the PER0 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

Remark m: Unit number, n: Channel number, p: Slave channel number 1 (n+1), q: Slave channel number 2 (n+2)
When m = 0
n = 0, 2, 4 (This is, however, p = 1 to 3 and 5 in the case of the timer output pin (TO0p), q = 1 to 3 and 5 in the case of the timer output pin (TO0q).)
$n < p < q \leq 7$ (where p and q are a consecutive integer greater than n)
When m = 1
n = 0
$n < p < q \leq 3$ (where p and q are a consecutive integer greater than n)

CHAPTER 9 WATCHDOG TIMER

9.1 Functions of Watchdog Timer

The watchdog timer operates on the internal low-speed oscillation 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 WDTE
- If data is written to WDTE during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDRF) of the reset control flag register (RESF) is set to 1. For details of RESF, see **CHAPTER 17 RESET FUNCTION**.

When 75% of the overflow time is reached, an interval interrupt can be generated.

9.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

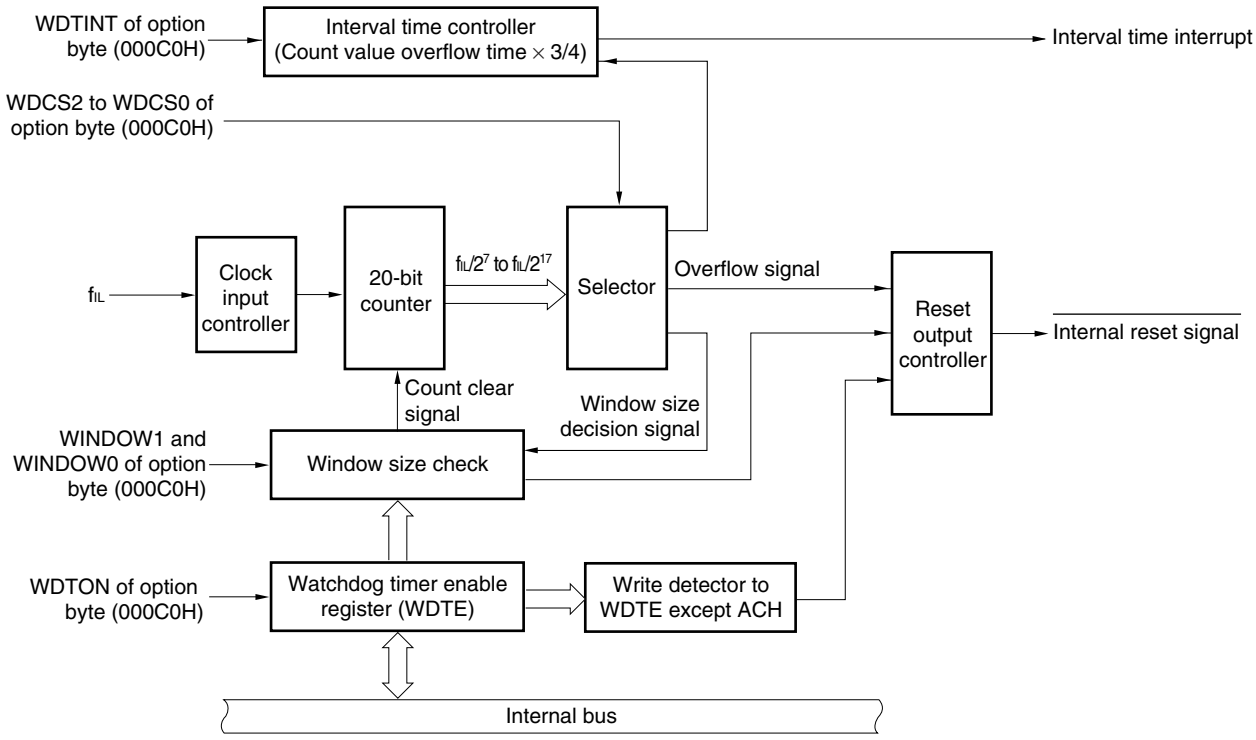
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 9-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the option byte, see **CHAPTER 21 OPTION BYTE**.

Figure 9-1. Block Diagram of Watchdog Timer



9.3 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 WDTE clears the watchdog timer counter and starts counting again.

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

Reset signal generation sets this register to 9AH or 1AH[Note].

Figure 9-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 9AH/1AH[Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE reset value differs depending on the WDTON setting value of the option byte (000C0H). To operate watchdog timer, set WDTON to 1.

WDTON Setting Value	WDTE 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 WDTE, an internal reset signal is generated.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated.

3. The value read from WDTE is 9AH/1AH (this differs from the written value (ACH)).

9.4 Operation of Watchdog Timer

9.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 21**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **9.4.2 Setting overflow time of watchdog timer** and **CHAPTER 21**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **9.4.3 Setting window open period of watchdog timer** and **CHAPTER 21**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to 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 WDTE the second time or later after a reset release during the window open period. If WDTE is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to WDTE, 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 watchdog timer enable register (WDTE)
   - If data other than “ACH” is written to WDTE

Cautions 1. When data is written to 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 WDTE, 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.

9.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to WDTE during the window open period before the overflow time.

The following overflow times can be set.

Table 9-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 34.5 kHz (MAX.))
0	0	0	$2^7/f_{IL}$ (3.71 ms)
0	0	1	$2^8/f_{IL}$ (7.42 ms)
0	1	0	$2^9/f_{IL}$ (14.84 ms)
0	1	1	$2^{10}/f_{IL}$ (29.68 ms)
1	0	0	$2^{12}/f_{IL}$ (118.72 ms)
1	0	1	$2^{14}/f_{IL}$ (474.90 ms)
1	1	0	$2^{15}/f_{IL}$ (949.80 ms)
1	1	1	$2^{17}/f_{IL}$ (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.

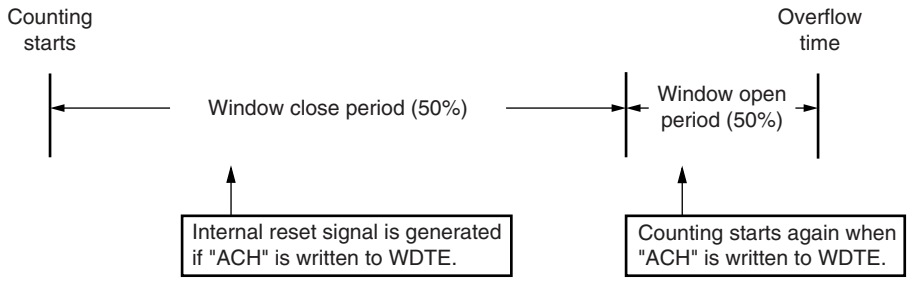
Remark f_{IL}: Internal low-speed oscillation clock frequency

9.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to WDTE during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to WDTE 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 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.

The window open period can be set is as follows.

Table 9-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	Setting prohibited
0	1	50%
1	0	75%
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 WINDOW1 and WINDOW0.

Remark If the overflow time is set to $2^{10}/f_{IL}$, 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:
  $2^{10}/f_{IL}$ (MAX.) = 2^{10}/34.5 kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^{10}/f_{IL}$ (MIN.) × (1 − 0.5) = 0 to 2^{10}/25.5 kHz (MIN.) × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^{10}/f_{IL}$ (MIN.) × (1 − 0.5) to $2^{10}/f_{IL}$ (MAX.) = 2^{10}/25.5 kHz (MIN.) × 0.5 to 2^{10}/34.5 kHz (MAX.) = 20.08 to 29.68 ms

9.4.4 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% of the overflow time is reached.

Table 9-5. Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is used.
1	Interval interrupt is generated when 75% 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 WDTE register). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

CHAPTER 10 A/D CONVERTER

10.1 Function of A/D Converter

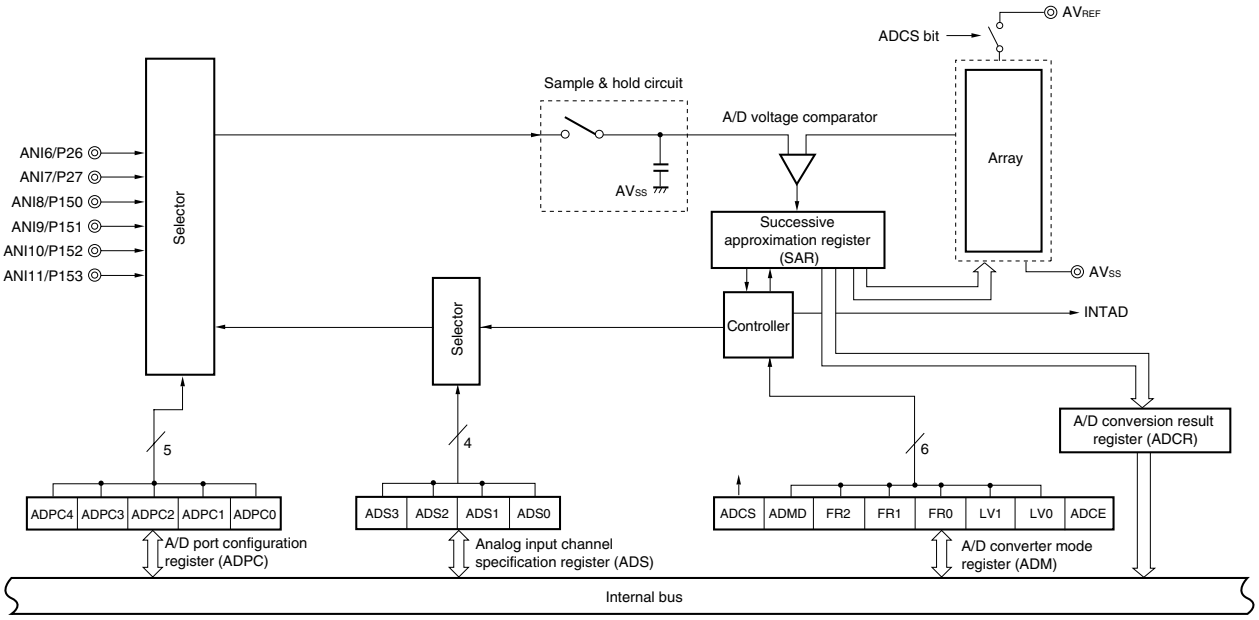
The A/D converter converts an analog input signal into a digital value, and consists of up to 6 channels (ANI6 to ANI11) with a resolution of 10 bits.

The A/D converter has the following function.

- **10-bit resolution A/D conversion**

  10-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI6 to ANI11. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 10-1. Block Diagram of A/D Converter



10.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI6 to ANI11 pins

These are the analog input pins of the 6 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 array with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB of the SAR is reset.

After that, bit 10 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the array is selected by the value of bit 11, to which the result has been already set.

Bit 11 = 0: (1/4 AV_{REF})
Bit 11 = 1: (3/4 AV_{REF})

The voltage tap of the array and the analog input voltage are compared and bit 10 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of array: Bit 10 = 1
Analog input voltage ≤ Voltage tap of array: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR register.

(4) Array

The array generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR register is a 12-bit register that sets voltage tap data whose values from the array 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_{REF} pin

This pin inputs the reference voltage of the A/D converter, the power supply pins and A/D converter of the comparator. When all pins of ports 2 and 15 are used as the analog port pins, make the potential of AV_{REF} be such that 1.8 V ≤ AV_{REF} ≤ V_{DD}. When one or more of the pins of ports 2 and 15 are used as the digital port pins, make AV_{REF} the same potential as V_{DD}.

The analog signal input to ANI6 to ANI11 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

(10) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the V_{SS} pin even when the A/D converter is not used.

10.3 Registers Used in A/D Converter

The A/D converter uses the following seven registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register (ADM)
- A/D port configuration register (ADPC)
- Analog input channel specification register (ADS)
- Port mode registers 2, 15 (PM2, PM15)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)

(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.

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

Reset signal generation clears this register to 00H.

Figure 10-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

ADCEN	Control of A/D converter input clock
0	Stops supply of input clock. • 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.

Cautions 1. When setting the A/D converter, be sure to set ADCEN 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 2, 15 (PM2, PM15)).

2. Be sure to clear bits 6 and 7 to 0.

3. Be sure to first set the ADCEN bit of peripheral enable register 0 (PER0) to 1 when setting up the ADPC register. If ADCEN = 0, writing to the ADPC register is ignored and specified values are returned to the initial values.

(2) A/D converter mode register (ADM)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

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

Reset signal generation clears this register to 00H.

Figure 10-3. Format of A/D Converter Mode Register (ADM)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

ADCE	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

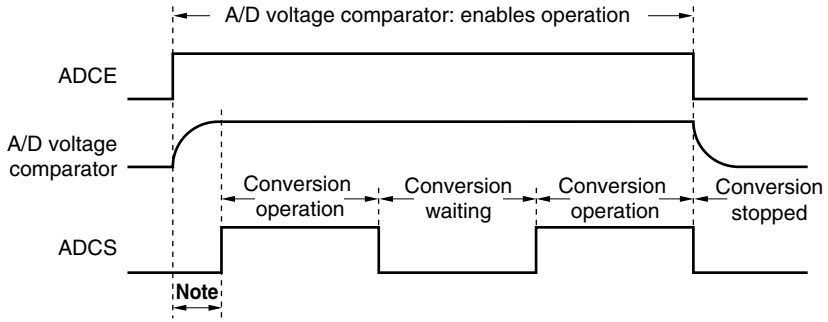
Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 10-2 A/D Conversion Time Selection**.

2. The operation of the A/D voltage comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, by waiting for at least 1 μs to elapse before setting ADCS to 1 after ADCE has been set to 1, the conversion results are valid from the first result. Otherwise, ignore data of the first conversion.

Table 10-1. Settings of ADCS and ADCE

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (only A/D voltage comparator consumes power)
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator: enables operation)

Figure 10-4. Timing Chart When A/D Voltage Comparator Is Used



Note To stabilize the internal circuit, the time from the rising of the ADCE bit to the rising of the ADCS bit must be 1 μs or longer.

Caution A/D conversion must be stopped before rewriting bits FR0 to FR2, LV1, and LV0 to values other than the identical data.

Table 10-2. A/D Conversion Time Selection (1/3)

(1) 4.0 V ≤ AV_{REF} ≤ 5.5 V

A/D Converter Mode Register (ADM)					Mode	Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	
0	0	0	0	0	Normal	Setting prohibited	Setting prohibited	34.2 μs	17.1 μs	f_{CLK}/20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f_{CLK}/10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f_{CLK}/8
0	1	1				52.0 μs	20.8 μs	10.4 μs	5.2 μs	f_{CLK}/6
1	0	0				35.0 μs	14.0 μs	7.0 μs	Setting prohibited	f_{CLK}/4
1	0	1				26.5 μs	10.6 μs	5.3 μs		f_{CLK}/3
1	1	0				18.0 μs	7.2 μs	Setting prohibited		f_{CLK}/2
1	1	1				9.5 μs	Setting prohibited			f_{CLK}
×	×	×	0	1	Low-voltage	Setting prohibited				–
0	0	0	1	0	High speed 1	Setting prohibited	64.4 μs	32.2 μs	16.1 μs	f_{CLK}/20
0	0	1					32.4 μs	16.2 μs	8.1 μs	f_{CLK}/10
0	1	0				65.0 μs	26.0 μs	13.0 μs	6.5 μs	f_{CLK}/8
0	1	1				49.0 μs	19.6 μs	9.8 μs	4.9 μs	f_{CLK}/6
1	0	0				33.0 μs	13.2 μs	6.6 μs	3.3 μs	f_{CLK}/4
1	0	1				25.0 μs	10.0 μs	5.0 μs	2.5 μs	f_{CLK}/3
1	1	0				17.0 μs	6.8 μs	3.4 μs	Setting prohibited	f_{CLK}/2
1	1	1				9.0 μs	3.6 μs	Setting prohibited		f_{CLK}
0	0	0	1	1	High speed 2	Setting prohibited	Setting prohibited	34.2 μs	17.1 μs	f_{CLK}/20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f_{CLK}/10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f_{CLK}/8
0	1	1				52.0 μs	20.8 μs	10.4 μs	5.2 μs	f_{CLK}/6
1	0	0				35.0 μs	14.0 μs	7.0 μs	3.5 μs	f_{CLK}/4
1	0	1				26.5 μs	10.6 μs	5.3 μs	Setting prohibited	f_{CLK}/3
1	1	0				18.0 μs	7.2 μs	3.6 μs		f_{CLK}/2
1	1	1				9.5 μs	3.8 μs	Setting prohibited		f_{CLK}

Cautions 1. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. 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

Table 10-2. A/D Conversion Time Selection (2/3)

(2) 2.7 V ≤ AV_{REF} ≤ 5.5 V

A/D Converter Mode Register (ADM)					Mode	Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	
0	0	0	0	0	Normal	Setting prohibited	Setting prohibited	34.2 μs	17.1 μs	f_{CLK}/20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f_{CLK}/10
0	1	0					27.6 μs	13.8 μs	Setting prohibited	f_{CLK}/8
0	1	1				52.0 μs	20.8 μs	10.4 μs		f_{CLK}/6
1	0	0				35.0 μs	14.0 μs	Setting prohibited		f_{CLK}/4
1	0	1				26.5 μs	10.6 μs			f_{CLK}/3
1	1	0				18.0 μs	Setting prohibited			f_{CLK}/2
1	1	1				9.5 μs				f_{CLK}
×	×	×	0	1	Low-voltage	Setting prohibited				–
×	×	×	1	0	High speed 1	Setting prohibited				–
0	0	0	1	1	High speed 2	Setting prohibited	Setting prohibited	34.2 μs	17.1 μs	f_{CLK}/20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f_{CLK}/10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f_{CLK}/8
0	1	1				52.0 μs	20.8 μs	10.4 μs	5.2 μs	f_{CLK}/6
1	0	0				35.0 μs	14.0 μs	7.0 μs	3.5 μs	f_{CLK}/4
1	0	1				26.5 μs	10.6 μs	5.3 μs	Setting prohibited	f_{CLK}/3
1	1	0				18.0 μs	7.2 μs	3.6 μs		f_{CLK}/2
1	1	1				9.5 μs	3.8 μs	Setting prohibited		f_{CLK}

Cautions 1. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. 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

Table 10-2. A/D Conversion Time Selection (3/3)

(3) 1.8 V ≤ AV_{REF} ≤ 5.5 V

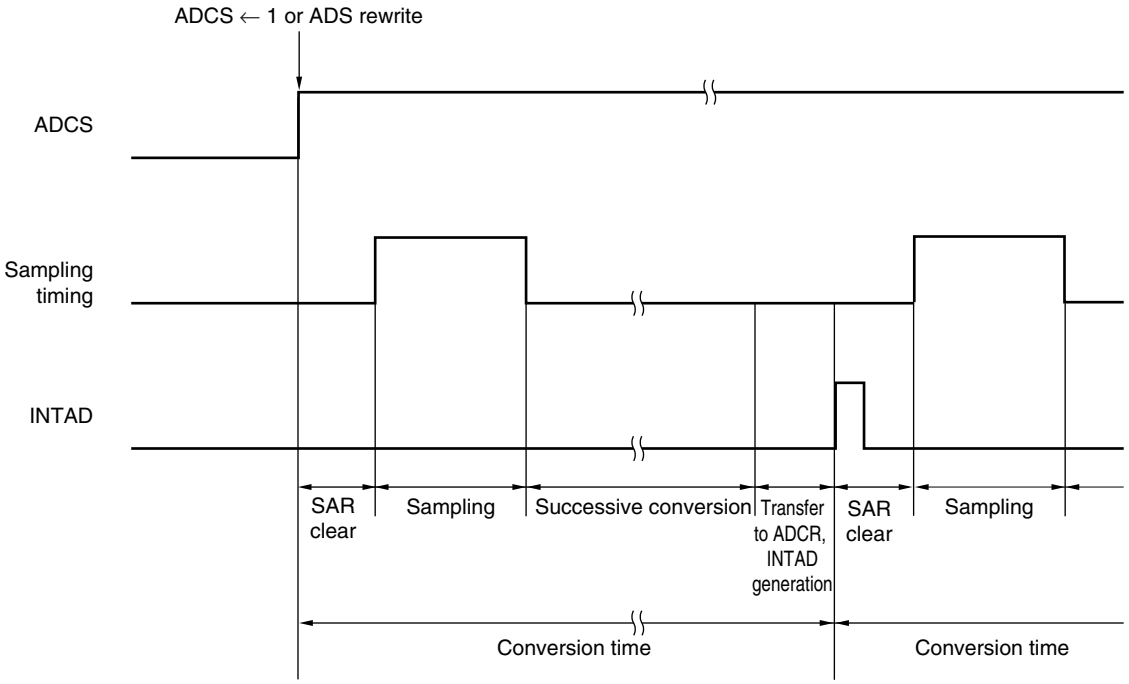
A/D Converter Mode Register (ADM)					Mode	Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	
×	×	×	0	0	Normal	Setting prohibited				
0	0	0	1	1	Low-voltage	Setting prohibited	Setting prohibited	48.2 μs	24.1 μs	f_{CLK}/20
0	0	1					48.4 μs	24.2 μs	Setting prohibited	f_{CLK}/10
0	1	0					38.8 μs	Setting prohibited		f_{CLK}/8
0	1	1					29.2 μs			f_{CLK}/6
1	0	0				49.0 μs	Setting prohibited			f_{CLK}/4
1	0	1				37.0 μs				f_{CLK}/3
1	1	0				25.0 μs				f_{CLK}/2
1	1	1				Setting prohibited				f_{CLK}
×	×	×	1	0	High speed 1	Setting prohibited				–
×	×	×	1	1	High speed 2	Setting prohibited				–
Other than above					Setting prohibited					

Cautions 1. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. 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

Figure 10-5. A/D Converter Sampling and A/D Conversion Timing



(3) 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.

ADCR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10-6. Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Caution When writing to A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

(4) 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.

ADCRH can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-7. 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 A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

(5) Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

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

Reset signal generation clears this register to 00H.

Figure 10-8. Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	1	1	0	ANI6	P26/ANI6 pin
0	1	1	1	ANI7	P27/ANI7 pin
1	0	0	0	ANI8	P150/ANI8 pin
1	0	0	1	ANI9	P151/ANI9 pin
1	0	1	0	ANI10	P152/ANI10 pin
1	0	1	1	ANI11	P153/ANI11 pin
Other than above				Setting prohibited	

Cautions 1. Be sure to clear bits 4 to 7 to "0".

2 Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 15 (PM2 and PM15).

3. Do not set the pin that is set by ADPC as digital I/O by ADS.

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

This register switches the ANI6/P26, ANI7/P27, and ANI8/P150 to ANI11/P153 pins to analog input of A/D converter or digital I/O of port.

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

Reset signal generation sets this register to 10H.

Figure 10-9. Format of A/D Port Configuration Register (ADPC)

Address: F0017H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching					
					Port 15				Port 2	
					ANI11/ P153	ANI10/ P152	ANI9/ P151	ANI8/ P150	ANI7/ P27	ANI6/ P26
0	0	1	1	0	A	A	A	A	A	A
0	0	1	1	1	A	A	A	A	A	D
0	1	0	0	0	A	A	A	A	D	D
0	1	0	0	1	A	A	A	D	D	D
0	1	0	1	0	A	A	D	D	D	D
0	1	0	1	1	A	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D
Other than above					Setting prohibited					

Cautions 1. Set a channel to be used for A/D conversion in the input mode by using port mode register 2 and 15 (PM2, PM15).

2. Do not set the pin that is set by ADPC as digital I/O by ADS.

(7) Port mode registers 2 and 15 (PM2, PM15)

When using the ANI6/P26 to ANI7/P27 and ANI8/P150 to ANI11/P153 pins for analog input port, set PM26, PM27, and PM150 to PM153 to 1. The output latches of P26, P27, and P150 to P153 at this time may be 0 or 1.

If PM26, PM27, and PM150 to PM153 are set to 0, they cannot be used as analog input port pins.

PM2 and PM15 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 10-10. Formats of Port Mode Registers 2 and 15 (PM2, PM15)

Address: FFF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	0	0	0	0	0	0

Address: FFF2FH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM15	1	1	1	1	PM153	PM152	PM151	PM150

PMmn	Pmn pin I/O mode selection (mn = 26, 27, 150 to 153)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

The ANI6/P26, ANI7/P27 and ANI8/P150 to ANI11/P153 pins are as shown below depending on the settings of ADPC, ADS, PM2, and PM15.

Table 10-3. Setting Functions of ANI6/P26, ANI7/P27, and ANI8/P150 to ANI11/P153 Pins

ADPC	PM2 and PM15	ADS	ANI6/P26, ANI7/P27, and ANI8/P150 to ANI11/P153 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.	

10.4 A/D Converter Operations

10.4.1 Basic operations of A/D converter

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the A/D converter.

<2> Set the A/D conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM, and set the operation mode by using bit 6 (ADMD) of ADM.

<3> Set bit 0 (ADCE) of A/D converter mode register (ADM) to 1 to start the operation of the A/D voltage comparator.

<4> Set the channels for A/D conversion to analog input by using the A/D port configuration register (ADPC) and set to input mode by using port mode registers (PM2 and PM15).

<5> Select one channel for A/D conversion using the analog input channel specification register (ADS).

<6> Start the conversion operation by setting bit 7 (ADCS) of ADM to 1.
A timer trigger wait state is entered if the timer trigger mode is set in step <7>.
(<7> to <13> are operations performed by hardware.)

<7> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<8> 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.

<9> 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.

<10> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset to 0.

<11> Next, bit 8 of SAR 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 SAR is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<12> Comparison is continued in this way up to bit 0 of SAR.

<13> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

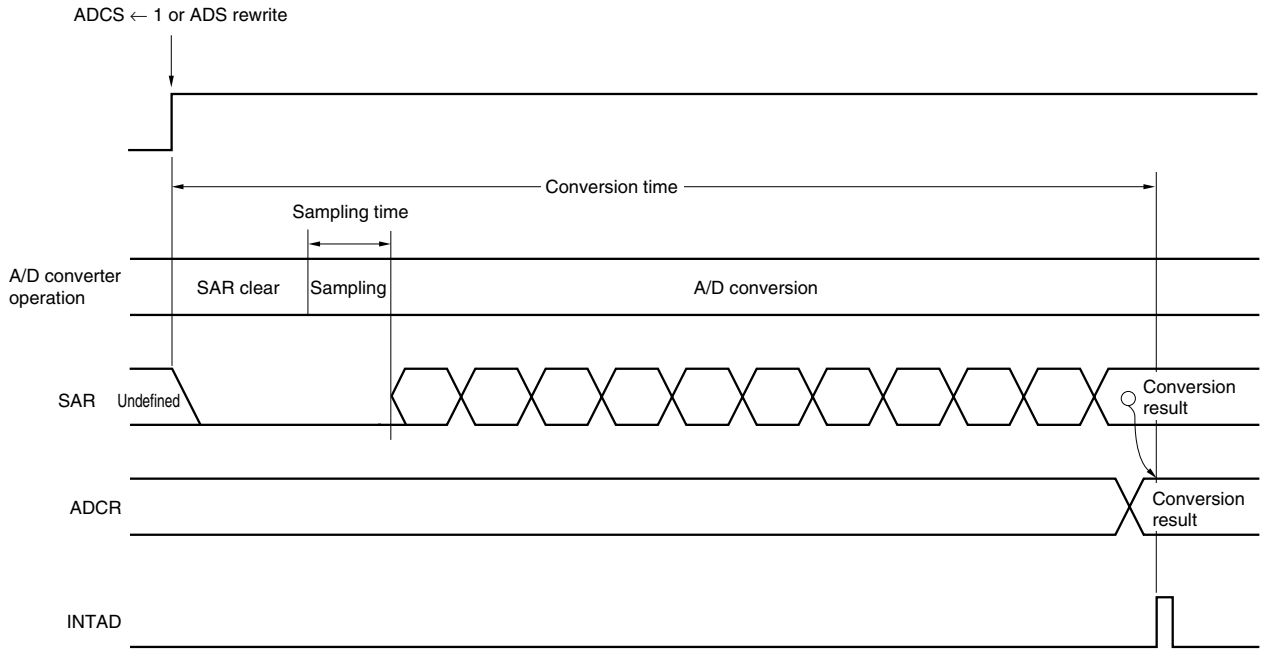
<14> Repeat steps <7> to <13>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <6>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <6>. To change a channel of A/D conversion, start from <5>.

Caution Make sure the period of <3> to <6> is 1 μs or more.

Remark Two types of A/D conversion result registers are available.

- ADCR (16 bits): Store 10-bit A/D conversion value
- ADCRH (8 bits): Store 8-bit A/D conversion value

Figure 10-11. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS) of 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.

10.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI6 to ANI11) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

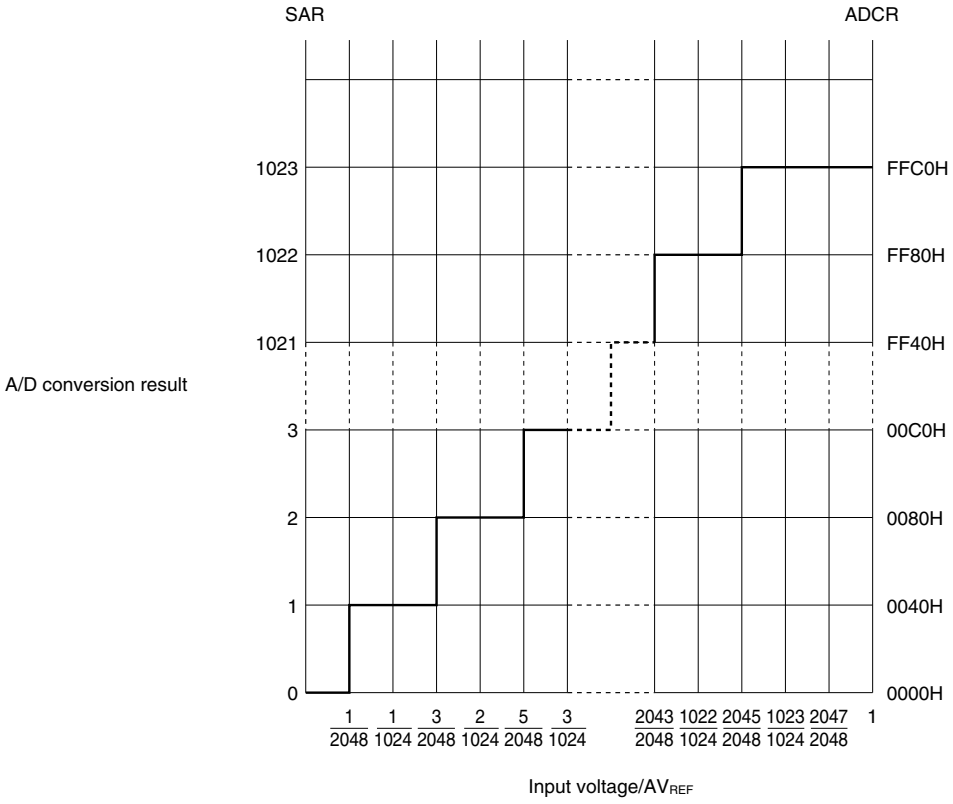
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 10-12 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-12. Relationship Between Analog Input Voltage and A/D Conversion Result



10.4.3 A/D converter operation modes

The select mode is provided as the A/D converter operation mode.

(1) Select mode

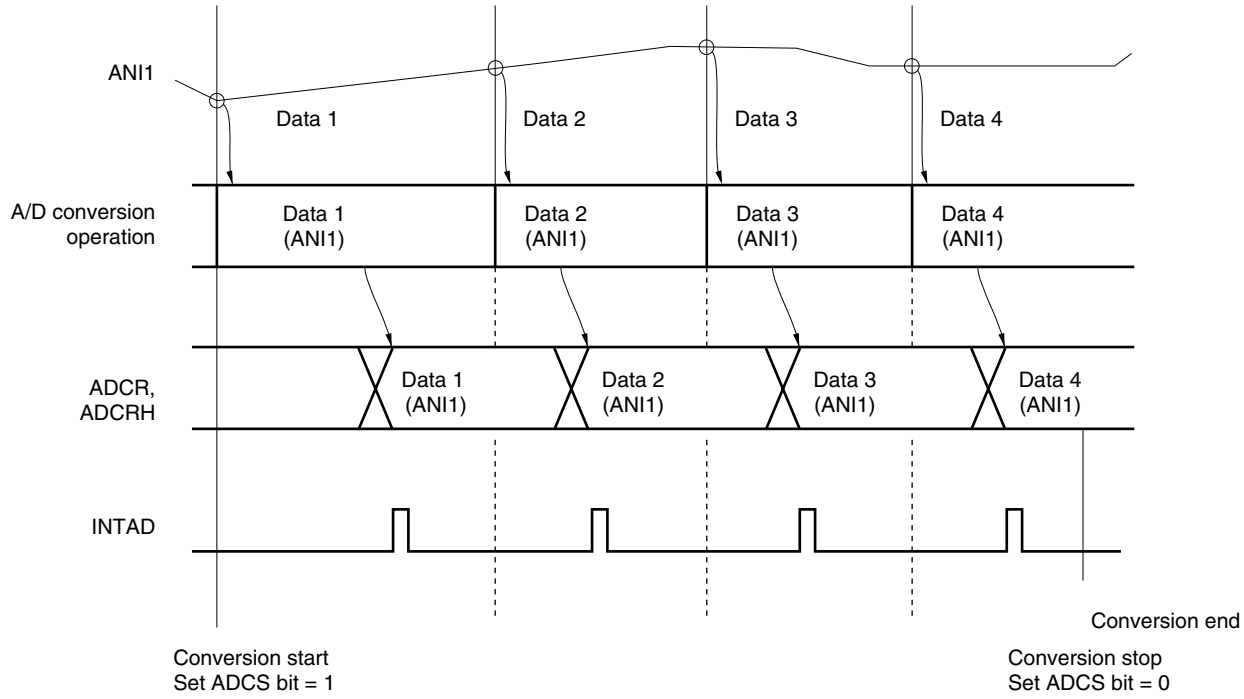
One analog input specified by the analog input channel specification register (ADS), while the ADMD bit of A/D converter mode register (ADM) is 0, is A/D converted.

When A/D conversion is complete, the conversion result is stored in the A/D conversion result register (ADCR) and the A/D conversion end interrupt request signal (INTAD) is generated.

After A/D conversion has been completed, A/D conversion is repeated successively, unless the ADCS bit is set to 0.

If anything is written to ADM or ADS during conversion, A/D conversion is aborted. In this case, A/D conversion is started again from the beginning.

Figure 10-13. Example of Select Mode Operation Timing



The setting methods are described below.

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1.
<2> Select the conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM, and select the operation mode by using bit 6 (ADMD) of ADM.
<3> Set bit 0 (ADCE) of A/D converter mode register (ADM) to 1.
<4> Set the channel to be used in the analog input mode by using bits 4 to 0 (ADPC4 to ADPC0) of the A/D port configuration register (ADPC), bits 7 and 6 (PM27 and PM26) of port mode register 2 (PM2), and bits 3 to 0 (PM153 to PM150) of port mode register 15 (PM15).
<5> Select a channel to be used by using bits 3 to 0 (ADS3 to ADS0) of the analog input channel specification register (ADS).
<6> Set bit 7 (ADCS) of ADM to 1 to start A/D conversion.
<7> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<8> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Change the channel>

<9> Change the channel using bits 3 to 0 (ADS3 to ADS0) of ADS to start A/D conversion.
<10> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<11> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>

<12> Clear ADCS to 0.
<13> Clear ADCE to 0.
<14> Clear bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 0.

Cautions 1. Make sure the period of <3> to <6> is 1 μs or more.
2. <3> may be done between <4> and <5>.
3. <3> can be omitted. However, ignore data of the first conversion after <7> in this case.
4. The period from <7> to <10> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <9> to <10> is the conversion time set using FR2 to FR0, LV1, and LV0.

10.5 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 10-14. Overall Error

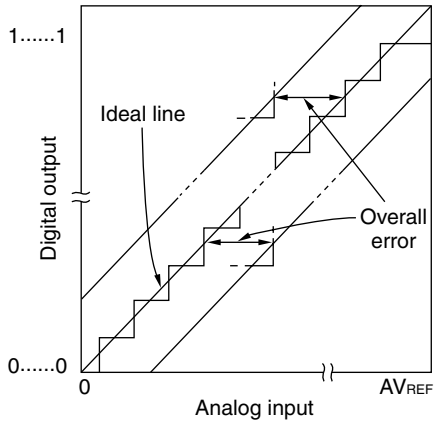
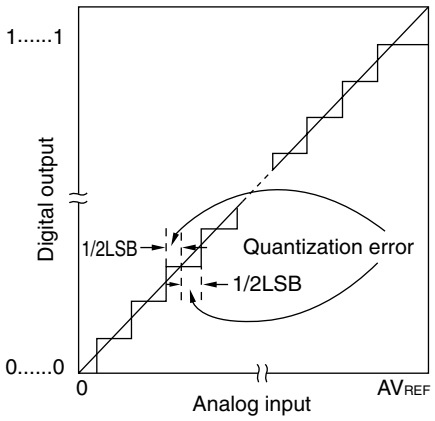


Figure 10-15. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.
If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 10-16. Zero-Scale Error

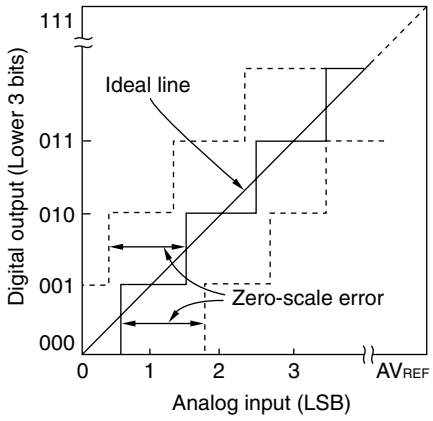


Figure 10-17. Full-Scale Error

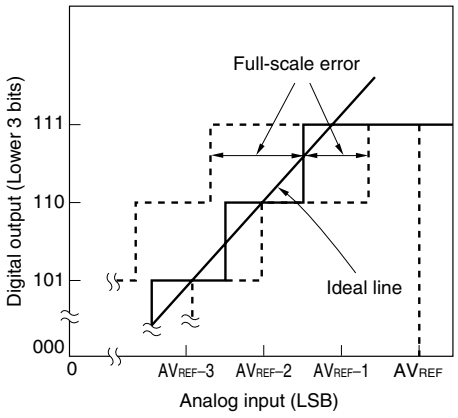


Figure 10-18. Integral Linearity Error

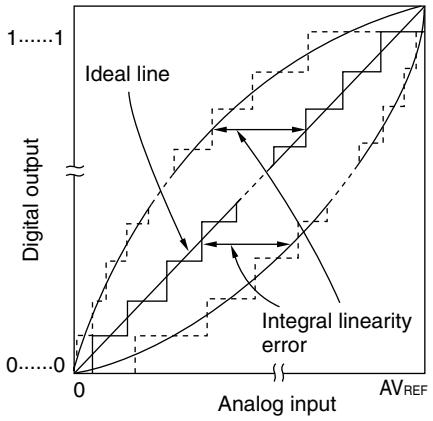
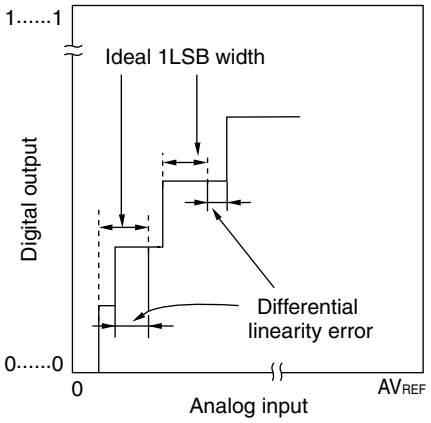


Figure 10-19. Differential Linearity Error



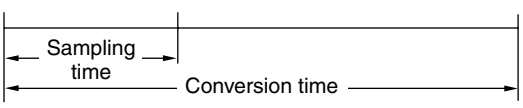
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



10.6 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 the A/D converter mode register (ADM) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.

(2) Input range of ANI6 to ANI11

Observe the rated range of the ANI6 to ANI11 input voltage. If a voltage of AV_{REF} or higher and AV_{SS} 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.

(3) Conflicting operations

<1> Conflict between A/D conversion result register (ADCR, ADCRH) write and ADCR or ADCRH read by instruction upon the end of conversion

ADCR or ADCRH read has priority. After the read operation, the new conversion result is written to ADCR or ADCRH.

<2> Conflict between ADCR or ADCRH write and A/D converter mode register (ADM) write, analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

ADM, ADS, or ADPC write has priority. ADCR or ADCRH 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_{REF} pin and pins ANI6 to ANI11.

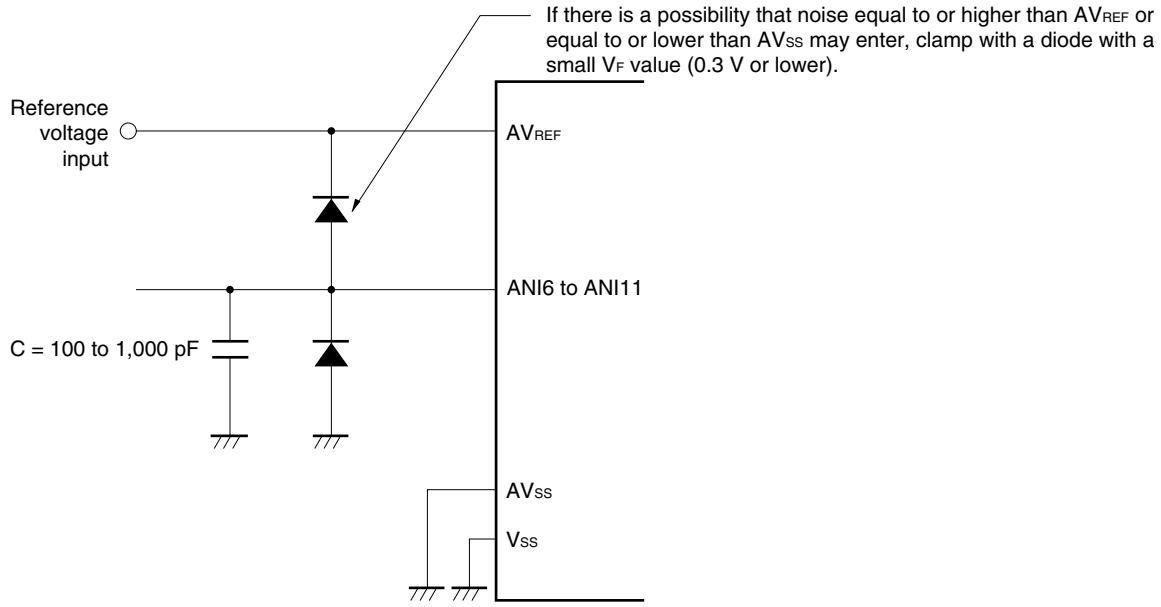
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<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 10-20 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 10-20. Analog Input Pin Connection



(5) ANI6/P26 to ANI7/P27 and ANI8/P150 to ANI11/P153

<1> The analog input pins (ANI6 and ANI7) are also used as input port pins (P26 and P27).
The analog input pins (ANI8 to ANI11) are also used as input port pins (P150 to P153).
When A/D conversion is performed with any of ANI6 to ANI11 selected, do not access P26, P27, and P150 to P153 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P26, P27, and P150 to P153 starting with the ANI6/P26 that is the furthest from AVREF.

<2> If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

(6) Input impedance of ANI6 to ANI11 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 ANI6 to ANI11 pins (see **Figure 10-20**).

(7) AVREF pin input impedance

A series resistor string of several tens of kΩ is connected between the AVREF and AVSS pins.
Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AVREF and AVSS pins, resulting in a large reference voltage error.

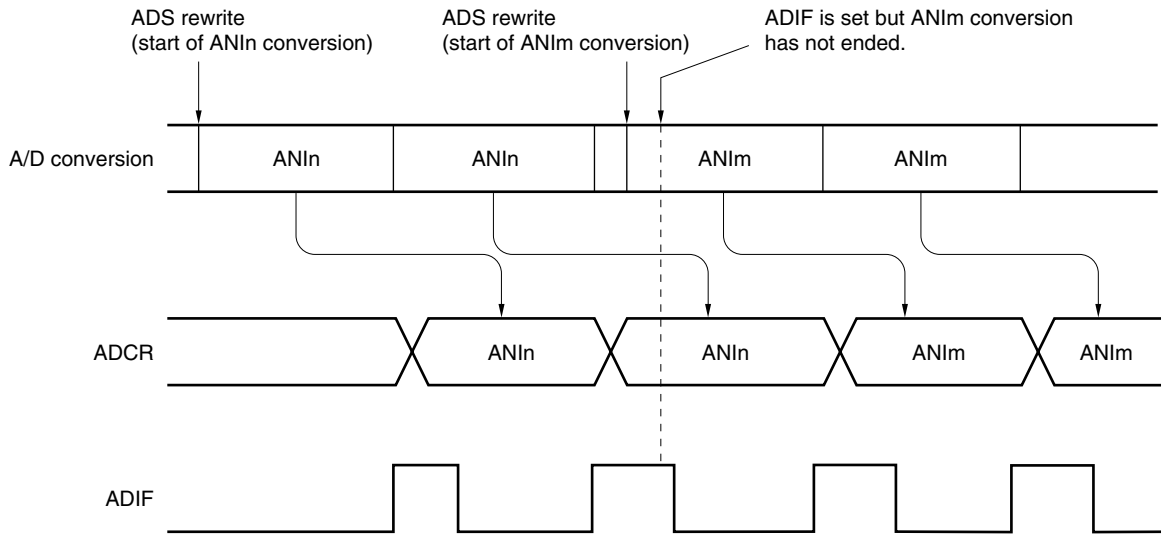
(8) Interrupt request flag (ADIF)

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 for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF 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 before the A/D conversion operation is resumed.

Figure 10-21. Timing of A/D Conversion End Interrupt Request Generation



Remark n = 6 to 11
m = 6 to 11

(9) Conversion results just after A/D conversion start

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.

(10) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, or ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 10-22. Internal Equivalent Circuit of ANIn Pin

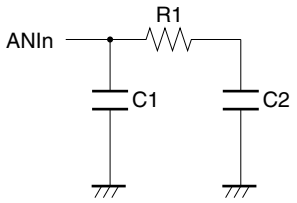


Table 10-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	Mode	R1	C1	C2
4.0 V ≤ AV_{REF} ≤ 5.5 V	Normal	5.2 kΩ	8 pF	6.3 pF
	High speed 1	5.2 kΩ		
	High speed 2	7.8 kΩ		
2.7 V ≤ AV_{REF} < 4.0 V	Normal	18.6 kΩ		
	High speed 2	7.8 kΩ		
1.8 V ≤ AV_{REF} < 4.0 V	Low-voltage	169.8 kΩ		

Remarks 1. The resistance and capacitance values shown in Table 10-4 are not guaranteed values.

2. n = 6 to 11

(12) Starting the A/D converter

Start the A/D converter after the AV_{REF} voltage stabilize.

CHAPTER 11 SERIAL ARRAY UNIT

Each serial array unit has four serial channels, each of which can be used for 3-wire serial (CSI), UART, and simplified I^2C communication.

Function assignment of each channel supported by the μPD78F8040, 78F8041, 78F8042, 78F8043 is as shown below (channels 0 and 1 of unit 0 are dedicated to IO-Link communication, channels 2 and 3 of unit 1 are dedicated to UART3 (supporting LIN-bus)).

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	–	UART0 (dedicated to IO-Link communication)	–
	1	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

When “UART2” is used for channels 0 and 1 of the unit 1, CSI20 and IIC20 cannot be used, but UART3 can be used for channels 2 and 3.

11.1 Functions of Serial Array Unit

Each serial interface supported by the μPD78F8040, 78F8041, 78F8042, 78F8043 have the following features.

11.1.1 3-wire serial I/O (CSI20)

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 **11.5 Operation of 3-Wire Serial I/O (CSI20) Communication.**

[Data transmission/reception]

- Data length of 7 or 8 bits
- 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: Max. $f_{CLK}/4$, during slave communication: Max. $f_{MCK}/6$ Note

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note Use the clocks within a range satisfying the $\overline{SCK}$ cycle time (t_{KCY}) characteristics (see **CHAPTER 26 ELECTRICAL SPECIFICATIONS**).

11.1.2 UART (UART0, UART2, UART3)

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).

For details about the settings, see **11.6 Operation of UART (UART0, UART2, UART3) Communication.**

[Data transmission/reception]

- Data length of 5, 7, or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

The LIN-bus is accepted in UART3 (2 and 3 channels of unit 1)

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

} External interrupt (INTP0) or timer array unit 0 is used.

Caution UART0 (0 and 1 channels of unit 0) is dedicated to IO-Link communication (see 11.7.3 to 11.7.5).

11.1.3 Simplified I²C (IIC20)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **11.8 Operation of Simplified I²C (IIC20) 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]

- Parity error (ACK error)

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

Note An ACK is not output when the last data is being received by writing 0 to the SOE10 (SOE1 register) bit and stopping the output of serial communication data. See **11.8.3 (2) Processing flow** for details.

Remark To use the full-function I²C bus, see **CHAPTER 12 SERIAL INTERFACE IICA**.

11.2 Configuration of Serial Array Unit

Serial array unit includes the following hardware.

Table 11-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits
Buffer register	Lower 8 bits of serial data register mn (SDRmn)[Note 1]
Serial clock I/O	$\overline{SCK20}$ pins (for 3-wire serial I/O), SCL20 pins (for simplified I^2C)
Serial data input	SI20 pins (for 3-wire serial I/O), RxD0 pin (for UART dedicating IO-Link), RxD2 pin (for UART), RxD3 pin (for UART supporting LIN-bus)
Serial data output	SO20 pins (for 3-wire serial I/O), TxD0[Note 2] (for UART dedicating IO-Link), TxD2 pin (for UART), TxD3 pin (for UART supporting LIN-bus), output controller
Serial data I/O	SDA20 pin (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn) • Port input mode register 14 (PIM14) • Port output mode register 14 (POM14) • Port mode registers 1, 14 (PM1, PM14) • Port registers 1, 14 (P1, P14)

Notes 1. The lower 8 bits of the serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.
- CSI20 communication … SIO20 (CSI20 data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)
- IIC20 communication … SIO20 (IIC20 data register)

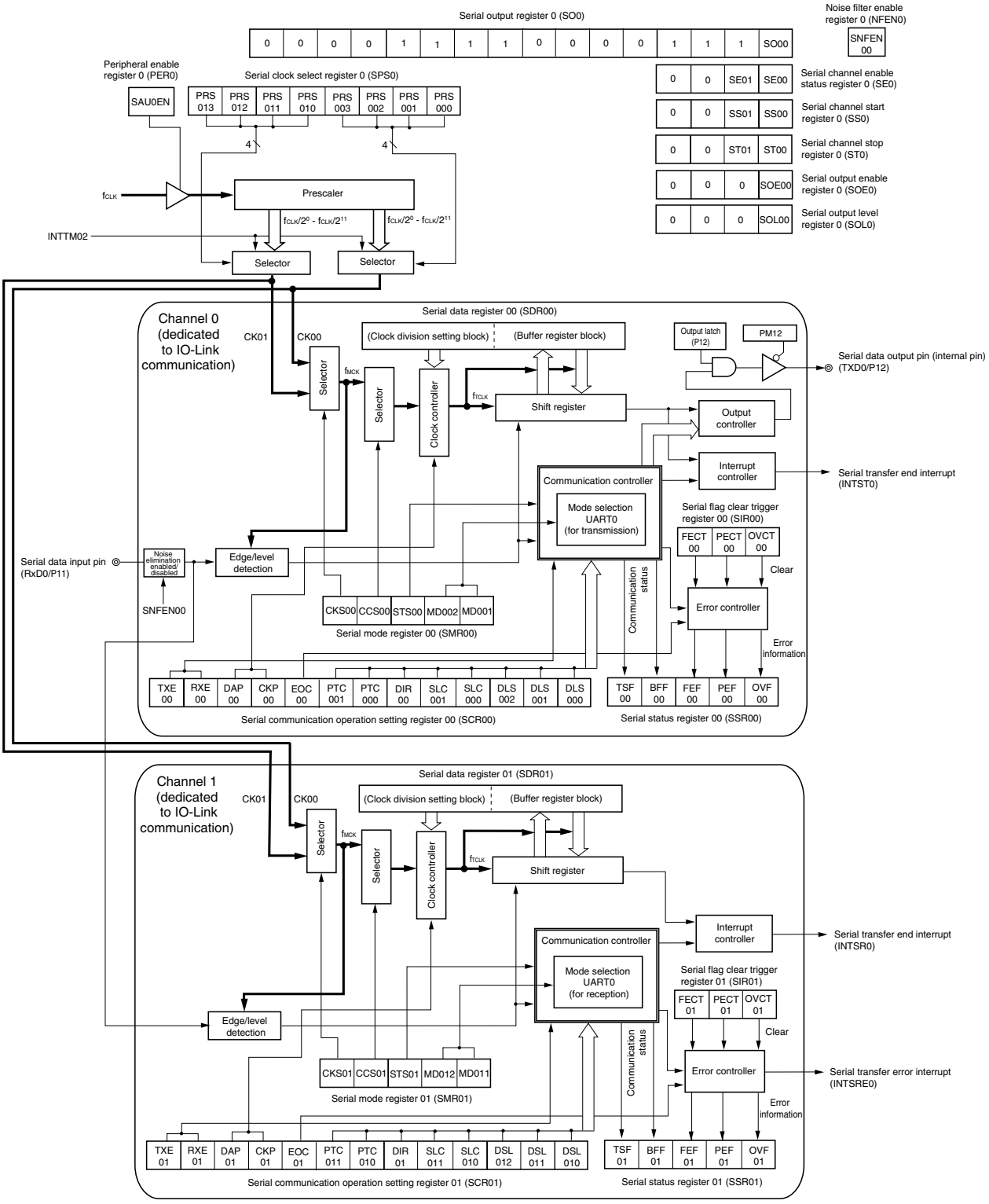
2. Internal connection pins between MCU and IO-Link transceiver.

Caution UART0 (0 and 1 channels of unit 0) is dedicated to IO-Link communication (see 11.7.3 to 11.7.5).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13
q: UART number (q = 0, 2, 3)

Figure 11-1 shows the block diagram of serial array unit 0.

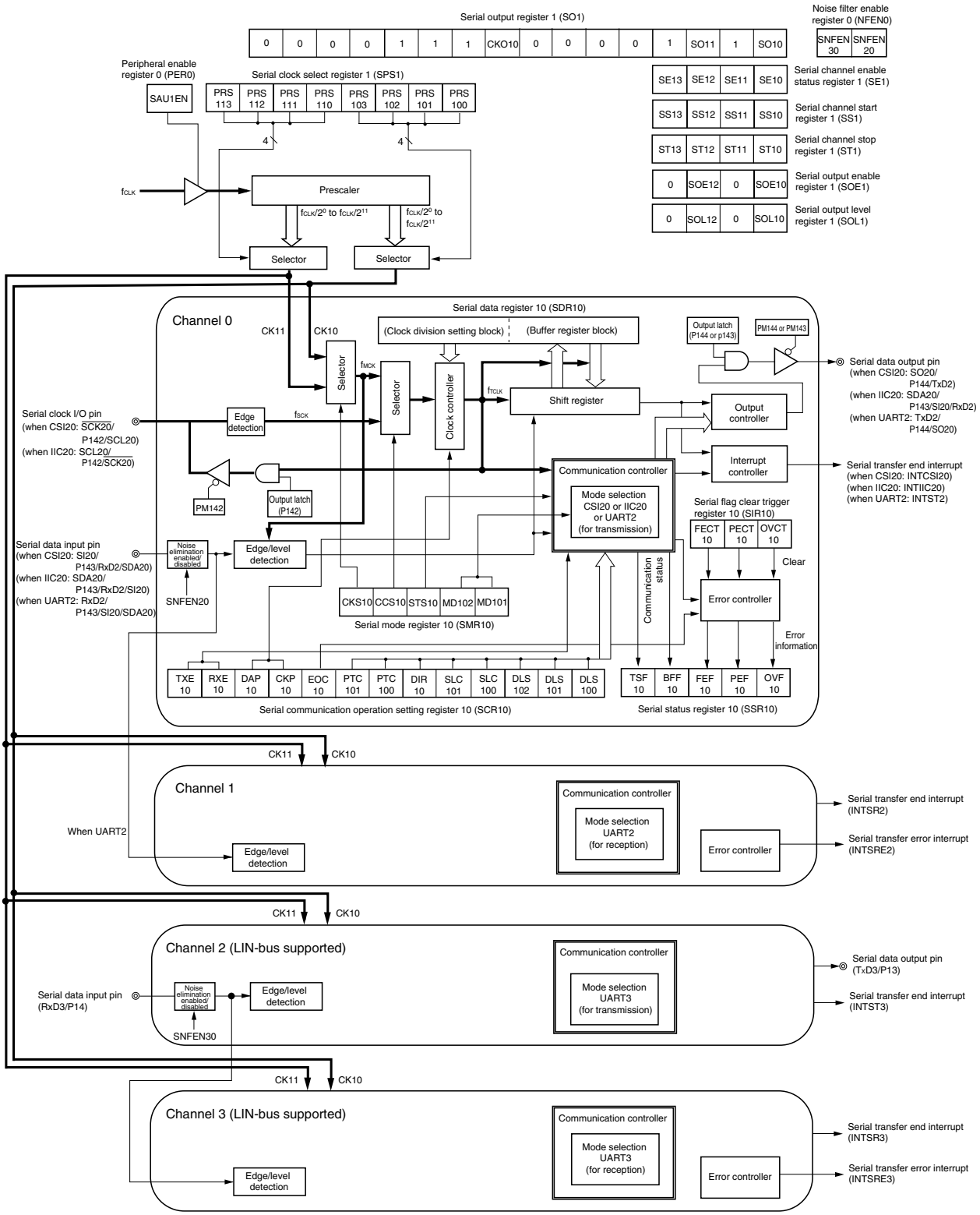
Figure 11-1. Block Diagram of Serial Array Unit 0



Caution UART0 (0 and 1 channels of unit 0) is dedicated to IO-Link communication (see 11.7.3 to 11.7.5).

Figure 11-2 shows the block diagram of serial array unit 1.

Figure 11-2. Block Diagram of Serial Array Unit 1



(1) Shift register

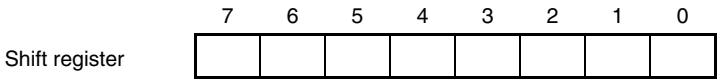
This is an 8-bit register that converts parallel data into serial data or vice versa.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8 bits of serial data register mn (SDRmn).



(2) Lower 8 bits of the serial data register mn (SDRmn)

SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 8 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 8 bits.

The data stored in the lower 8 bits of this register is as follows, depending on the setting of bits 0 to 2 (DLSmn0 to DLSmn2) of the SCRmn register, regardless of the output sequence of the data.

- 5-bit data length (stored in bits 0 to 4 of SDRmn register) (settable in UART 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)

SDRmn can be read or written in 16-bit units.

The lower 8 bits of SDRmn of SDRmn can be read or written[Note] as the following SFR, depending on the communication mode.

- CSI20 communication … SIO20 (CSI20 data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)
- IIC20 communication … SIO20 (IIC20 data register)

Reset signal generation clears this register to 0000H.

Note Writing in 8-bit units is prohibited when the operation is stopped (SEmn = 0).

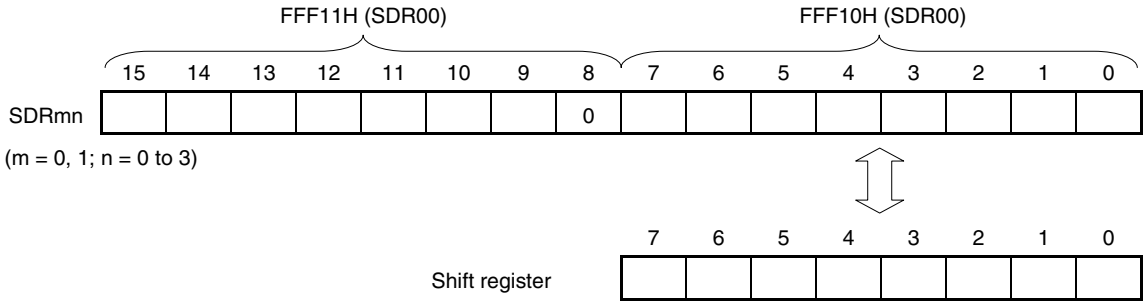
Remarks 1. After data is received, "0" is stored in bits 0 to 7 in bit portions that exceed the data length.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13
q: UART number (q = 0, 2, 3)

Caution UART0 (0 and 1 channels of unit 0) is dedicated to IO-Link communication (see 11.7.3 to 11.7.5).

Figure 11-3. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11), FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13) After reset: 0000H R/W



Caution Be sure to clear bit 8 to "0".

Remarks 1. For the function of the higher 7 bits of SDRmn, see **11.3 Registers Controlling Serial Array Unit**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

11.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial status register mn (SSRmn)
- Serial flag clear trigger register mn (SIRmn)
- 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 level register m (SOLm)
- Serial output register m (SOm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode register 14 (PIM14)
- Port output mode register 14 (POM14)
- Port mode registers 1, 14 (PM1, PM14)
- Port registers 1, 14 (P1, P14)

Remark m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 3)
mn = 00, 01, 10 to 13

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

When serial array unit 1 is used, be sure to set bit 3 (SAU1EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 11-4. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

SAUmEN	Control of serial array unit m input clock
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.

Cautions 1. When setting serial array unit m, be sure to set SAUmEN 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 noise filter enable register (NFEN0), port input mode register (PIM14), port output mode register (POM14), port mode registers (PM1, PM14), and port registers (P1, P14)).

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 bits 6 and 7 to 0.

Remark m: Unit number (m = 0, 1)

(2) Serial clock select register m (SPSm)

SPSm is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of SPSm, and CKm0 is selected by bits 3 to 0.

Rewriting SPSm is prohibited when the register is in operation (when SEmn = 1).

SPSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SPSm can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears this register to 0000H.

Figure 11-5. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mp3	PRS mp2	PRS mp1	PRS mp0	Section of operation clock (CKmp) Note 1				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	1	1	INTTM02 if m = 0, setting prohibited if m = 1				
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 (STm = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

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 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

2. m: Unit number (m = 0, 1), p = 0, 1

(3) Serial mode register mn (SMRmn)

SMRmn 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 SMRmn is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

SMRmn can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets this register to 0020H.

Figure 11-6. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00), F0112H, F0113H (SMR01), F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13), After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

CKS mn	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.	

CCS mn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by 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	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^2C).
1	Valid edge of RxD pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

Figure 11-6. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00), F0112H, F0113H (SMR01), F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13), After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0	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 SDR0n register to the shift register.)
For successive transmission, the next transmit data is written by setting MDmn0 to 1 when SDRmn data has run out.	

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

(4) Serial communication operation setting register mn (SCRmn)

SCRmn 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 SCRmn is prohibited when the register is in operation (when SEmn = 1).

SCRmn can be set by a 16-bit memory manipulation instruction.

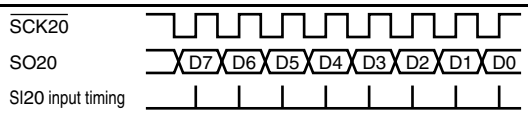
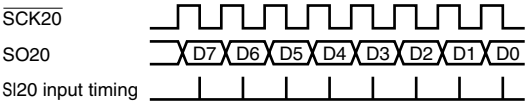
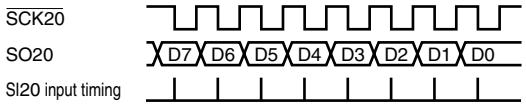
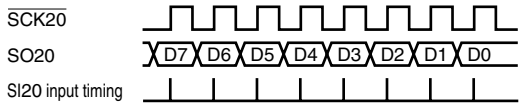
Reset signal generation sets this register to 0087H.

Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/3)

Address: F0118H, F0119H (SCR00), F011AH, F011BH (SCR01), F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in CSI mode	Type
0	0		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.			

Caution Be sure to clear bits 3, 6, and 11 to "0". Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)

Address: F0118H, F0119H (SCR00), F011AH, F011BH (SCR01), F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

EOC mn	Selection of masking of error interrupt signal (INTSREx (x = 0, 2, 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 1]. Set EOCmn = 1 during UART reception.	

PTC mn1	PTC mn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity[Note 2].	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTC001, PTC000 = 0, 0 in the CSI mode and simplified I^2C mode.			

DIR mn	Selection of data transfer sequence in CSI and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I^2C mode.	

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 and in the simplified I^2C mode. Set no stop bit (SLC001, SLC000 = 0, 0) in the CSI mode.		

Notes 1. When using CSI01 not with EOC01 = 0, error interrupt INTSRE0 may be generated.

2. 0 is always added regardless of the data contents.

Caution Be sure to clear bits 3, 6, and 11 to "0". Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (3/3)

Address: F0118H, F0119H (SCR00), F011AH, F011BH (SCR01), F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

DLS mn2	DLS mn1	DLS mn0	Setting of data length in CSI and UART modes
1	0	0	5-bit data length (stored in bits 0 to 4 of SDRmn register) (settable in UART mode only)
1	1	0	7-bit data length (stored in bits 0 to 6 of SDRmn register)
1	1	1	8-bit data length (stored in bits 0 to 7 of SDRmn register)
Other than above			Setting prohibited
Be sure to set DLSmn0 = 1 in the simplified I^2C mode.			

Caution Be sure to clear bits 3, 6, and 11 to "0". Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

(5) Serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 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 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8 bits.

SDRmn can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8 bits of SDRmn. When SDRmn is read during operation, 0 is always read.

Reset signal generation clears this register to 0000H.

Figure 11-8. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11), FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13) 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								0								

SDRmn[15:9]							Transfer clock setting by dividing the operating clock (f_{MCK})
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

Cautions 1. Be sure to clear bit 8 to "0".

2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

3. Setting SDR10[15:9] = 0000000B is prohibited when simplified I^2C is used. Setting SDR10[15:9] = 0000001B or more.

4. Do not write to the lower eight bits when operation is stopped (SEmn = 0). (Doing so clears the higher seven bits to 0.)

Remarks 1. For the function of the lower 8 bits of SDRmn, see **11.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 3)
mn = 00, 01, 10 to 13

(6) Serial flag clear trigger register mn (SIRmn)

SIRmn 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 SIRmn is a trigger register, it is cleared immediately when the corresponding bit of SSRmn is cleared.

SIRmn can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SIRmn can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears this register to 0000H.

Figure 11-9. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00), F010AH, F010BH (SIR01), F0148H, F0149H (SIR10) to F014EH, F014FH (SIR13) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn	PECTmn	OVCTmn

FECTmn	Clear trigger of framing error of channel n
0	No trigger operation
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	No trigger operation
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	No trigger operation
1	Clears the OVFmn bit of the SSRmn register to 0.

Caution Be sure to clear bits 15 to 3 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 10, 13

2. When the SIRmn register is read, 0000H is always read.

(7) Serial status register mn (SSRmn)

SSRmn 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.

SSRmn can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of SSRmn can be set with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears this register to 0000H.

Figure 11-10. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01), F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> • Transferring transmit data from the SDRmn register to the shift register ends during transmission. • Reading receive data from the SDRmn register ends during reception. • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

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), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

Figure 11-10. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01), F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn	PEF mn	OVF mn

FEF mn	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear condition> • 1 is written to the FECTmn bit of the SIRmn register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEF mn	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I^2C transmission).
<Clear condition> • 1 is written to the PECTmn bit of the SIRmn register. <Set condition> • The parity of the transmit data and the parity bit do not match when UART reception ends (parity error). • No ACK signal is returned from the slave channel at the ACK reception timing during I^2C transmission (ACK is not detected).	

OVF mn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> • 1 is written to the OVCTmn bit of the SIRmn register. <Set condition> • Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • Transmit data is not ready for slave transmission or transmission and reception in CSI mode.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

(8) Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

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

The lower 8 bits of the SSm register can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

Figure 11-11. Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H (SS0), F0162H, F0163H (SS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status[Note].

Note If a communication operation is already under execution, the operation is stopped.

Caution Be sure to clear bits 15 to 4 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

2. When the SSm register is read, 0000H is always read.

(9) Serial channel stop register m (STm)

The STm is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

The STm register can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of the STm register can be set with an 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears the STm register to 0000H.

Figure 11-12. Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H (ST0), F0164H, F0165H (ST1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm3	STm2	STm1	STm0

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation[Note].

Note 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 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

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 an 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears the SEm register to 0000H.

Figure 11-13. Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H (SE0), F0160H, F0161H (SE1) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm3	SEm2	SEm1	SEm0

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

(11) Serial output enable register m (SOEm)

SOEm 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 SOmn of the serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOmn 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.

SOEm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SOEm can be set with an 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears this register to 0000H.

Figure 11-14. Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE00

Address: F016AH, F016BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12	0	SOE10

SOEmn	Serial output enable/disable 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 1 of SOE0, and bits 1 and 15 to 3 of SOE1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2),
mn = 00, 10, 12

(12) Serial output register m (SOm)

SOm is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

SOmn of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

CKOmn of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of CKOmn can be changed only by a serial communication operation.

To use the P13/TxD3, P142/$\overline{\text{SCK20}}$/SCL20, P143/SI20/SDA20/RxD2, or P144/SO20/TxD2 pin as a port function pin, set the corresponding CKOmn and SOmn bits to "1".

SOm can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0F0FH.

Figure 11-15. Format of Serial Output Register m (SOm)

Address: F0128H, F0129H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	SO00

Address: F0168H, F0169H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	SO12	1	SO10

CKOmn	Serial clock output of channel n
0	Serial clock output value is "0".
1	Serial clock output value is "1".

SOmn	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 to 8 and 3 to 1 of SO0, and bits 11 to 9, 3, and 1 of SO1 to "1". And be sure to clear bits 15 to 12 and 7 to 4 of SOm to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2),
mn = 00, 10, 12

(13) Serial output level register m (SOLm)

SOLm 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 (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting SOLm is prohibited when the register is in operation (when SEmn = 1).

SOLm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SOLm can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears this register to 0000H.

Figure 11-16. Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H (SOL0), F0174H, F0175H (SOL1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL m2	0	SOL m0

SOL mn	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 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 10, 12

(14) Input switch control register (ISC)

ISC is used to realize a LIN-bus communication operation by UART3 in coordination with an external interrupt and the timer array unit 0.

When bit 0 is set to 1, the input signal of the serial data input (RxD3) 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 (RxDk) 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.

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

Reset signal generation clears this register to 00H.

Figure 11-17. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	1	ISC0

ISC0	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 RxD3 pin as an external interrupt (wakeup signal detection).

Caution Be sure to clear bits 7 to 2 to “0”. And be sure to set bit 1 to “1”.

(15) Noise filter enable register 0 (NFEN0)

NFEN0 is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for CSI or simplified I^2C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, CPU/peripheral hardware operating clock (f_{CLK}) is synchronized with 2-clock match detection.

NFEN0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-18. Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0060H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	SNFEN30	0	SNFEN20	0	0	0	SNFEN00

SNFEN30	Use of noise filter of RxD3 pin (RxD3/P14)
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the RxD3 pin. Clear SNFEN30 to 0 to use the other than RxD3 pin.	

SNFEN20	Use of noise filter of RxD2 pin (RxD2/SDA20/SI20/P143)
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.	

SNFEN00	Use of noise filter of RxD0/P11 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN00 to 1 to use the RxD0 pin (communicating with IO-Link transceiver (IO-Link mode[Note])). Clear SNFEN00 to 0 to use the P11 pin (communicating with IO-Link transceiver (SIO mode[Note])).	

Note For details of the IO-Link mode and SIO mode, refer to **4. 3. 3 IO-Link mode and standard I/O (SIO) mode**.

Caution Be sure to clear bits 7, 5, and 3 to 1 to "0".

(16) Port input mode register 14 (PIM14)

This register sets the input buffer of port 14 in 1-bit units.

PIM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-19. Format of Port Input Mode Register 14 (PIM14)

Address F004EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM14	0	0	0	0	PIM143	PIM142	0	0

PIM14n	P14n pin input buffer selection (n = 2, 3)
0	Normal input buffer
1	TTL input buffer

(17) Port output mode register 14 (POM14)

This register sets the output mode of port 14 in 1-bit units.

POM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-20. Format of Port Output Mode Register 14 (POM14)

Address F005EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM14	0	0	0	POM144	POM143	POM142	0	0

POM14n	P14n pin output buffer selection (n = 2 to 4)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

(18) Port mode registers 1, 14 (PM1, PM14)

These registers set input/output of ports 1 and 14 in 1-bit units.

When using the P12/TxD0, P13/TxD3, P142/SCK20/SCL20, P143/SI20/SDA20/RxD2, and P144/SO20/TxD2 pins for serial data output or serial clock output, clear the PM12, PM13, PM142, PM143, and PM144 bits to 0, and set the output latches of P12, P13, P142, P143, and P144 to 1.

When using the P11/RxD0, P14/RxD3, P142/SCK20/SCL20, and P143/SI20/RxD2/SDA20 pins for serial data input or serial clock input, set the PM11, PM14, PM142, and PM143 bits to 1. At this time, the output latches of P11, P14, P142, and P143 may be 0 or 1.

PM10 and PM12 bits are port mode register of an internal pin in the MCU.

PM1 and PM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 11-21. Format of Port Mode Registers 1 and 14 (PM1, PM14)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12[Note]	PM11	PM10[Note]

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	PM144	PM143	PM142	1	PM140

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

Note Port mode register of an internal connection pin between the MCU and IO-Link transceiver (For the pin settings, see **Table 2-1 Settings of Internal Connection Pins**).

Caution Be sure to clear PM140 to 0 after a reset release (see 2.2 Initial Setting of Unused Internal Pins in MCU).

11.4 Operation stop mode

Each serial interface of serial array unit has the operation stop mode.

In this mode, serial communication cannot be executed, thus reducing the power consumption.

In addition, the P11/RxD0, P12/TxD0, P13/TxD3, P14/RxD3, P142/SCK20/SCL20, P143/SI20/SDA20/RxD2, or P144/SO20/TxD2 pin can be used as ordinary port pins in this mode.

11.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0.

To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

Figure 11-22. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	0	0	ADCEN ×	IICAEN ×	SAU1EN 0/1	SAU0EN 0/1	TAU1EN ×	TAU0EN ×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

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 (except for noise filter enable register (NFEN0), port input mode register (PIM14), port output mode register (POM14), port mode registers (PM1, PM14), and port registers (P1, P14)).

2. Be sure to clear bits 6 and 7 to 0.

Remark m: Unit number (m = 0, 1), ▨: Setting disabled (fixed by hardware)
×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

11.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 11-23. Each Register Setting When Stopping the Operation by Channels (1/2)

(a) Serial channel stop register m (STm) … This register is a trigger register that is used to enable stopping communication/count by each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm3 0/1	STm2 0/1	STm1 0/1	STm0 0/1

1: Clears SEmn to 0 and stops the communication operation

* Because STmn is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm3 0/1	SEm2 0/1	SEm1 0/1	SEm0 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 CKO10 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.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE00 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO00 value of the SO0 register can be set by software.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12 0/1	0	SOE10 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO10 and SO12 value of the SO1 register can be set by software.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
□ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-23. Each Register Setting When Stopping the Operation by Channels (2/2)

(d) Serial output register m (SOm) …This register is a buffer register for serial output of each channel.

SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	SO00 0/1

1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding SO00 bits to “1”.

SO1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	CKO10 0/1	0	0	0	0	1	SO12 0/1	1	SO10 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 CKO10, SO10, and SO12 bits to “1”.

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

□ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

11.5 Operation of 3-Wire Serial I/O (CSI20) 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
- 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: Max. $f_{CLK}/4$, during slave communication: Max. $f_{MCK}/6$ Note

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note Use the clocks within a range satisfying the $\overline{SCK}$ cycle time (t_{KCY}) characteristics (see **CHAPTER 26 ELECTRICAL SPECIFICATIONS**).

The channels supporting 3-wire serial I/O (CSI20) is channel 0 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	–	UART0 (dedicated to IO-Link communication)	–
	1	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

3-wire serial I/O (CSI20) performs the following six types of communication operations.

- Master transmission (See **11.5.1**.)
- Master reception (See **11.5.2**.)
- Master transmission/reception (See **11.5.3**.)
- Slave transmission (See **11.5.4**.)
- Slave reception (See **11.5.5**.)
- Slave transmission/reception (See **11.5.6**.)

11.5.1 Master transmission

Master transmission is that the μPD78F8040, 78F8041, 78F8042, 78F8043 output a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{\text{SCK20}}$, SO20
Interrupt	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	None
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] Note f_{CLK}: System clock frequency
Data phase	Selectable by DAP10 bit of the SCR10 register • DAP10 = 0: Data output starts from the start of the operation of the serial clock. • DAP10 = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKP10 bit of the SCR10 register • CKP10 = 0: Forward • CKP10 = 1: Reverse
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 26 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-24. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI20)

(a) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10	CCS10	0	0	0	0	0	STS10	0	SIS100	1	0	0	MD102	MD101	MD100
	0/1	0						0		0				0	0	0/1

Operation clock (f_{MCK}) of channel 0
0: Prescaler output clock CK10 set by the SPS1 register
1: Prescaler output clock CK11 set by the SPS1 register

Interrupt source of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10	RXE10	DAP10	CKP10		EOC10	PTC101	PTC100	DIR10		SLC101	SLC100		DLS102	DLS101	DLS100
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **11.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data (Transmit data setting)							

SIO20

(d) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	1	1	SO10
								0/1								0/1

Communication starts when these bits are 1 if the data phase is forward (CKP10 = 0). If the phase is reversed (CKP10 = 1), communication starts when these bits are 0.

(e) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE10
																0/1

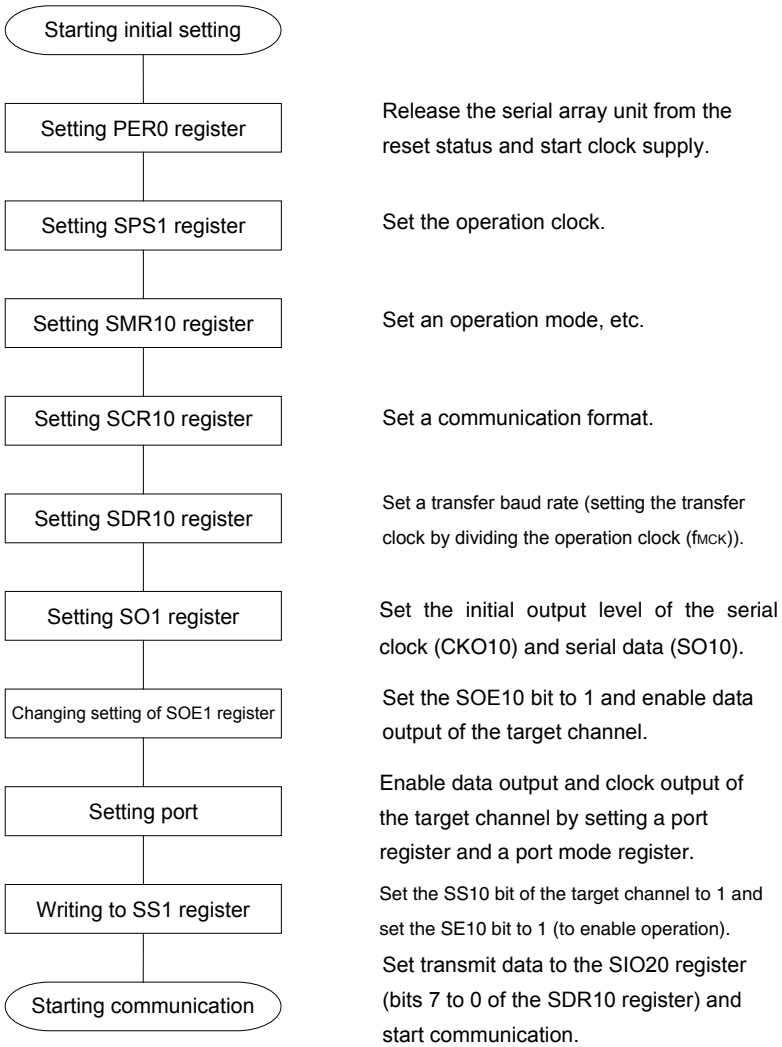
(f) Serial channel start register 1 (SS1) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13	SS12	SS11	SS10
													×	×	×	0/1

Remark □: Setting is fixed in the CSI master transmission mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

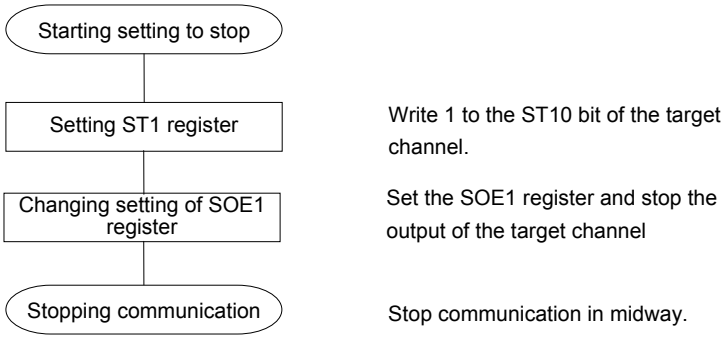
(2) Operation procedure

Figure 11-25. Initial Setting Procedure for Master Transmission



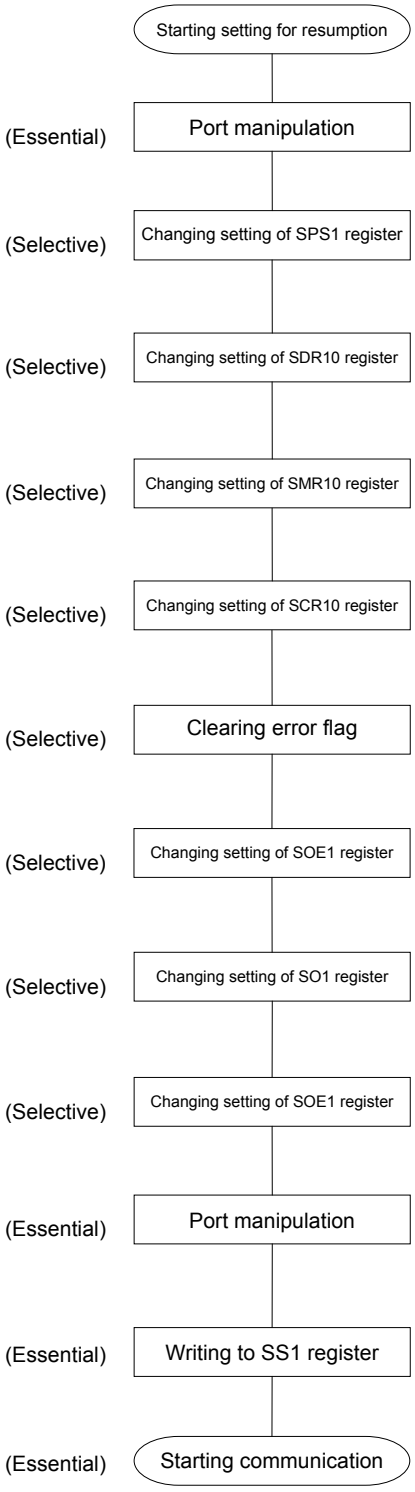
Caution After setting the SAU1EN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register 1 (SPS1) after 4 or more f_{CLK} clocks have elapsed.

Figure 11-26. Procedure for Stopping Master Transmission



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO1 register (see **Figure 11-27 Procedure for Resuming Master Transmission**).

Figure 11-27. Procedure for Resuming Master Transmission



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})).

Change the setting if the setting of the SMR10 register is incorrect.

Change the setting if the setting of the SCR10 register is incorrect.

If the FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register mn (SIR10).

Set the SOE10 bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKO10) and serial data (SO10).

Set the SOE10 bit to 1 and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SS10 bit of the target channel to 1 and set the SE10 bit to 1 (to enable operation).

Set transmit data to the SIO20 register (bits 7 to 0 of the SDR10 register) and start communication.

(3) Processing flow (in single-transmission mode)

Figure 11-28. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAP10 = 0, CKP10 = 0)

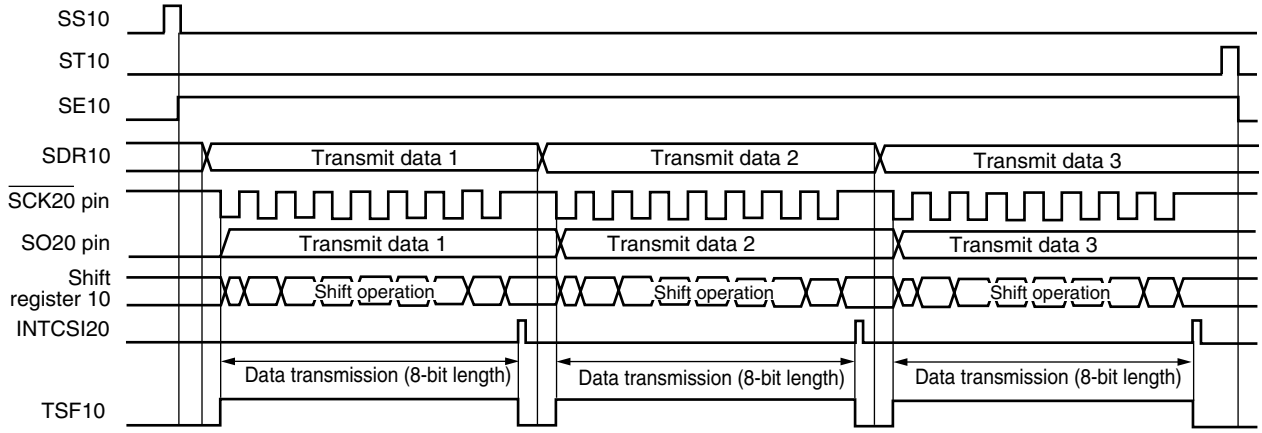
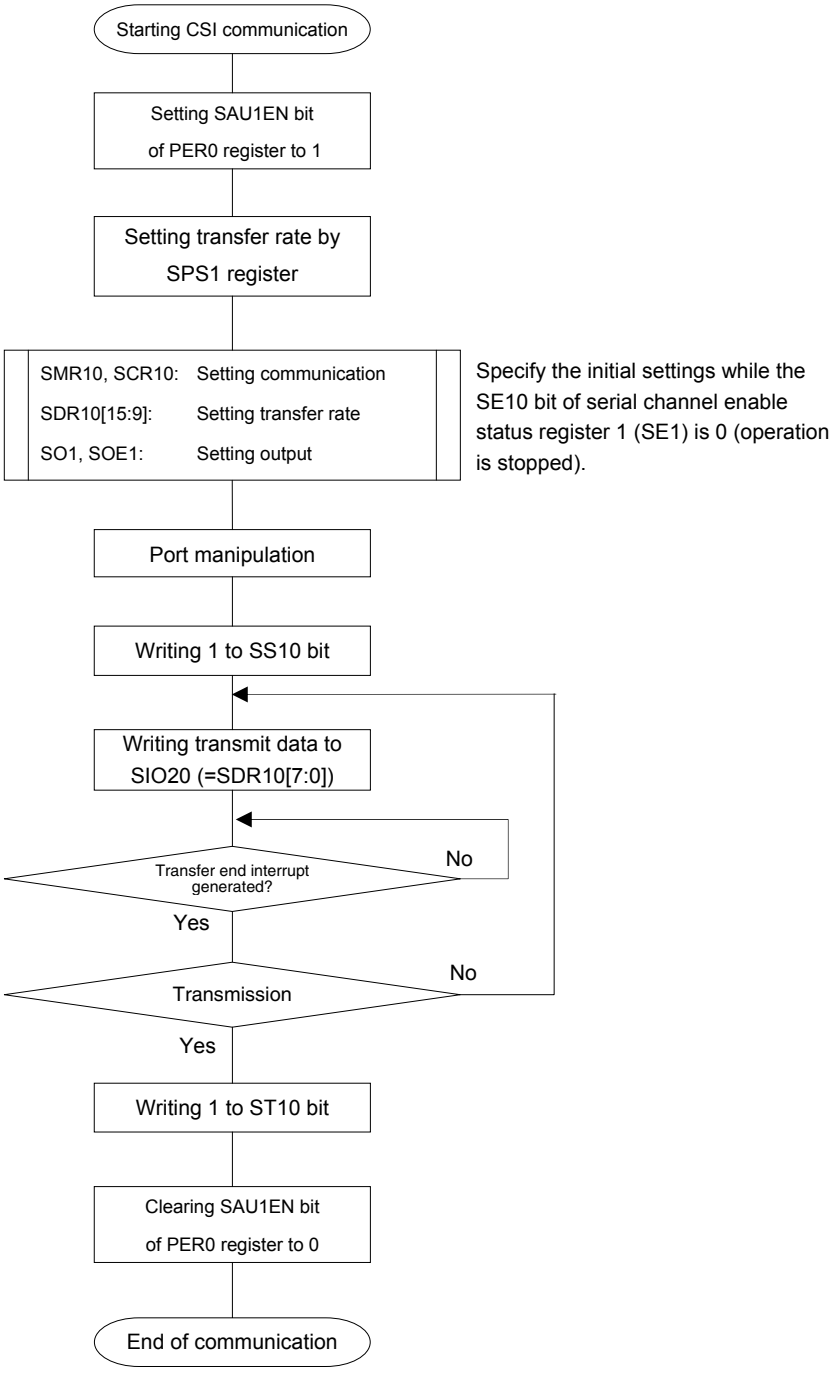


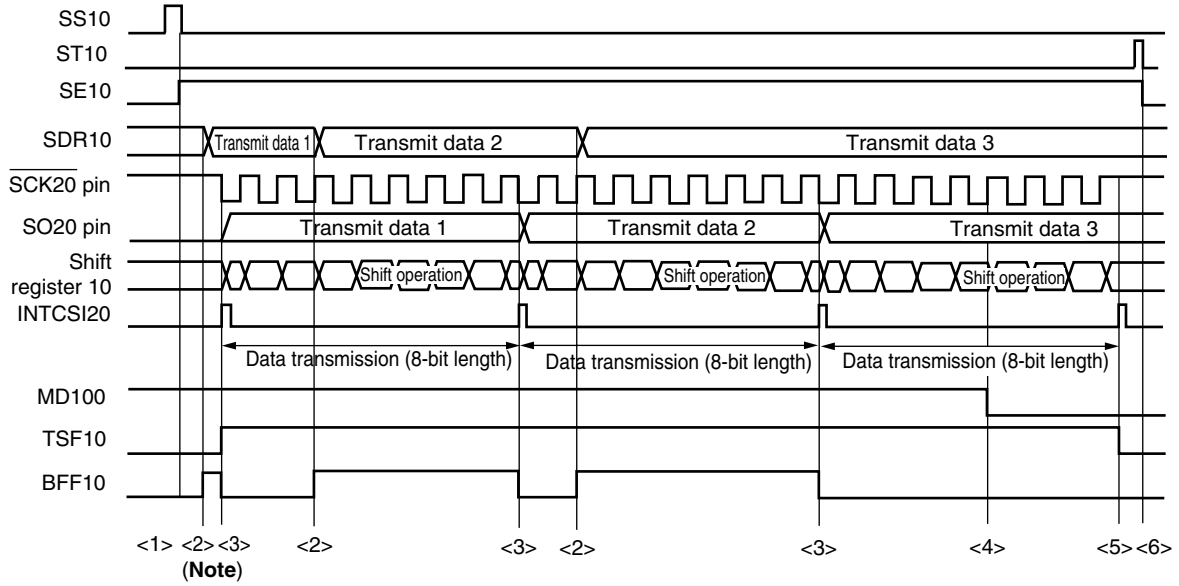
Figure 11-29. Flowchart of Master Transmission (in Single-Transmission Mode)



Caution After setting the SAU1EN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register 1 (SPS1) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

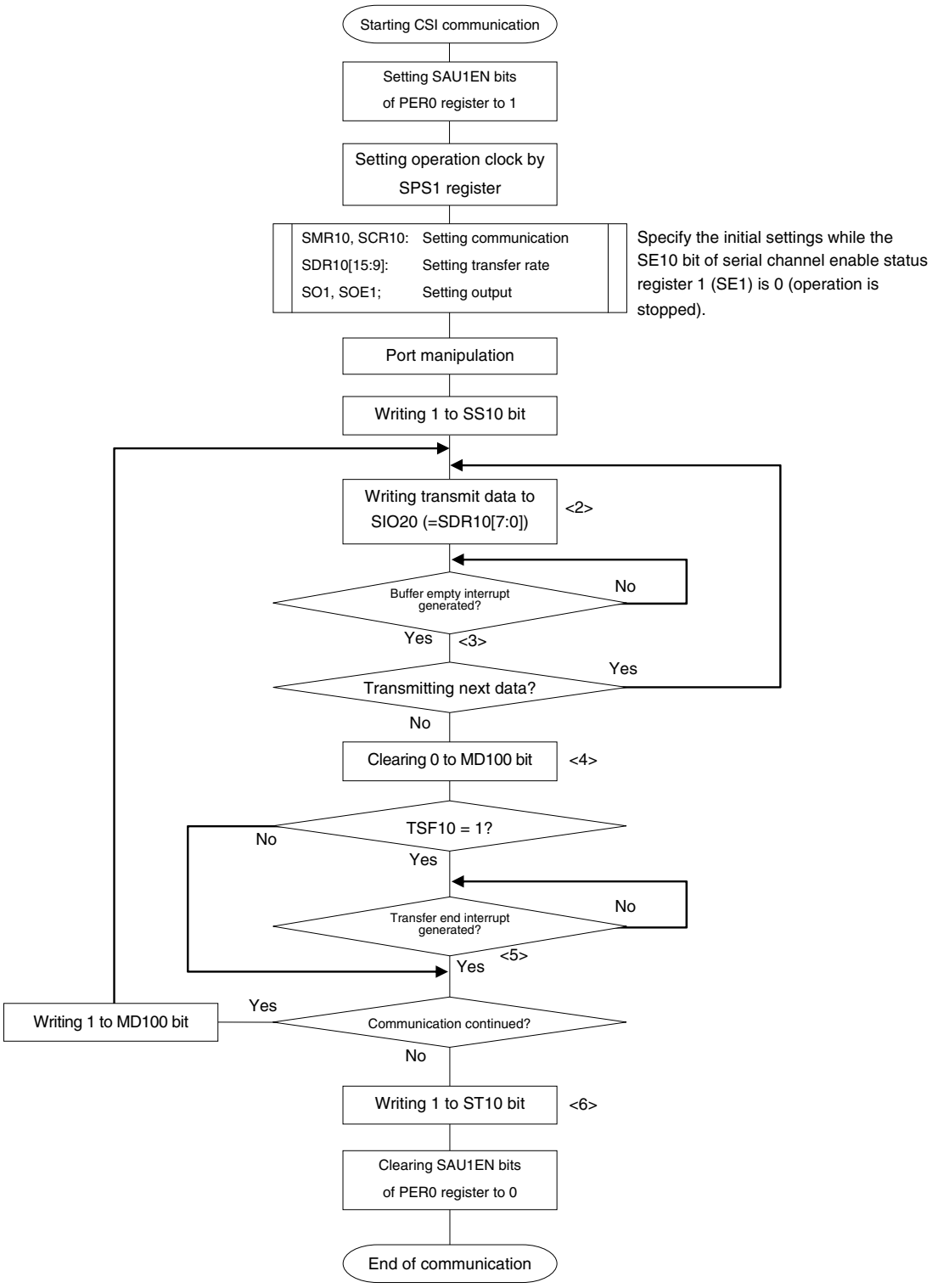
Figure 11-30. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAP10 = 0, CKP10 = 0)



Note If transmit data is written to the SDR10 register while the BFF10 bit is 1 (valid data is stored in serial data register 10 (SDR10)), the transmit data is overwritten.

Caution The MD100 bit can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Figure 11-31. Flowchart of Master Transmission (in Continuous Transmission Mode)



Caution After setting the SAU1EN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register 1 (SPS1) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-30 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

11.5.2 Master reception

Master reception is that the μPD78F8040, 78F8041, 78F8042, 78F8043 output a transfer clock and receives data from other device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{\text{SCK20}}$, SI20
Interrupt	INTCSI20
	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 (OVF10) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] Note f_{CLK}: System clock frequency
Data phase	Selectable by DAP10 bit • DAP10 = 0: Data input starts from the start of the operation of the serial clock. • DAP10 = 1: Data input starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKP10 bit • CKP10 = 0: Forward • CKP10 = 1: Reverse
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 26 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-32. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI20)

(a) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10	CCS10	0	0	0	0	0	STS10	0	SIS100	1	0	0	MD102	MD101	MD100
	0/1	0						0		0				0	0	0/1

Operation clock (f_{MCK}) of channel 0
0: Prescaler output clock CK10 set by the SPS1 register
1: Prescaler output clock CK11 set by the SPS1 register

Interrupt source of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10	RXE10	DAP10	CKP10		EOC10	PTC101	PTC100	DIR10		SLC101	SLC100		DLS102	DLS101	DLS100
	0	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **11.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Receive data (Write FFH as dummy data.)							

SIO20

(d) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	1	1	SO10
								0/1								×

Communication starts when these bits are 1 if the data phase is forward (CKP10 = 0). If the phase is reversed (CKP10 = 1), communication starts when these bits are 0.

(e) Serial output enable register 1 (SOE1) … The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE10
																×

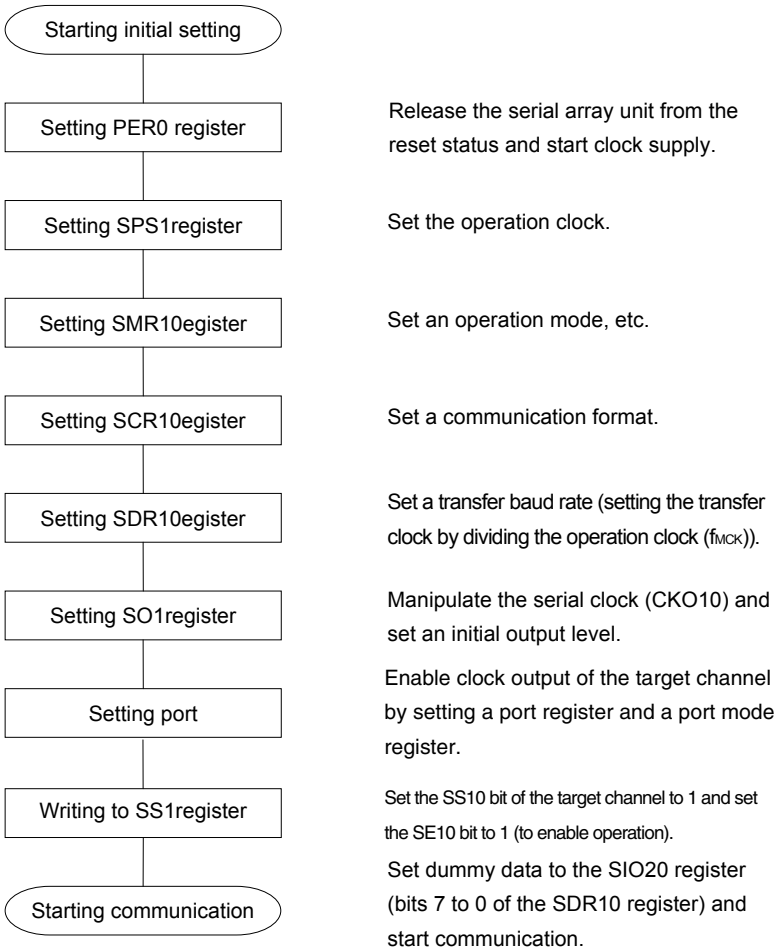
(f) Serial channel start register 1 (SS1) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS11	SS10
															×	0/1

Remark □: Setting is fixed in the CSI master reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

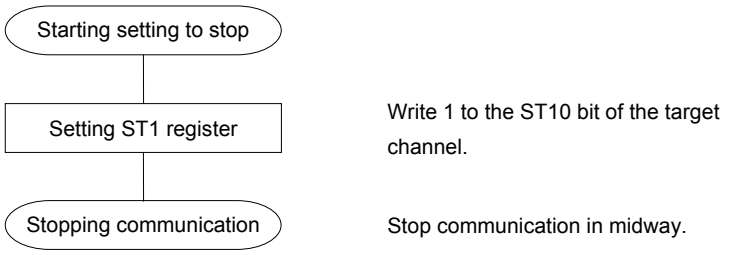
(2) Operation procedure

Figure 11-33. Initial Setting Procedure for Master Reception



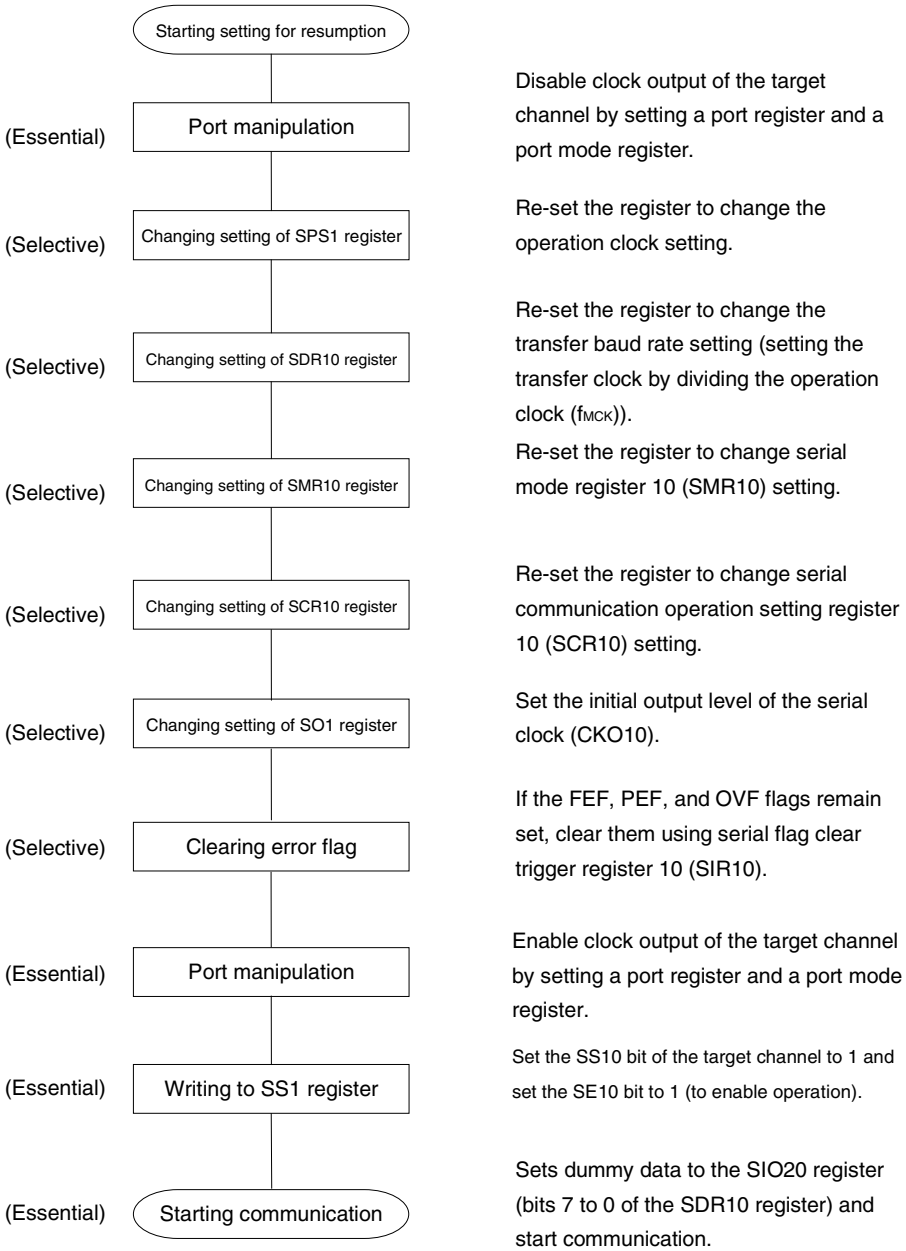
Caution After setting the SAU1EN bit of peripheral enable register 0 (PER0) to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

Figure 11-34. Procedure for Stopping Master Reception



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO1 register (see **Figure 11-35 Procedure for Resuming Master Reception**).

Figure 11-35. Procedure for Resuming Master Reception



(3) Processing flow (in single-reception mode)

Figure 11-36. Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAP10 = 0, CKP10 = 0)

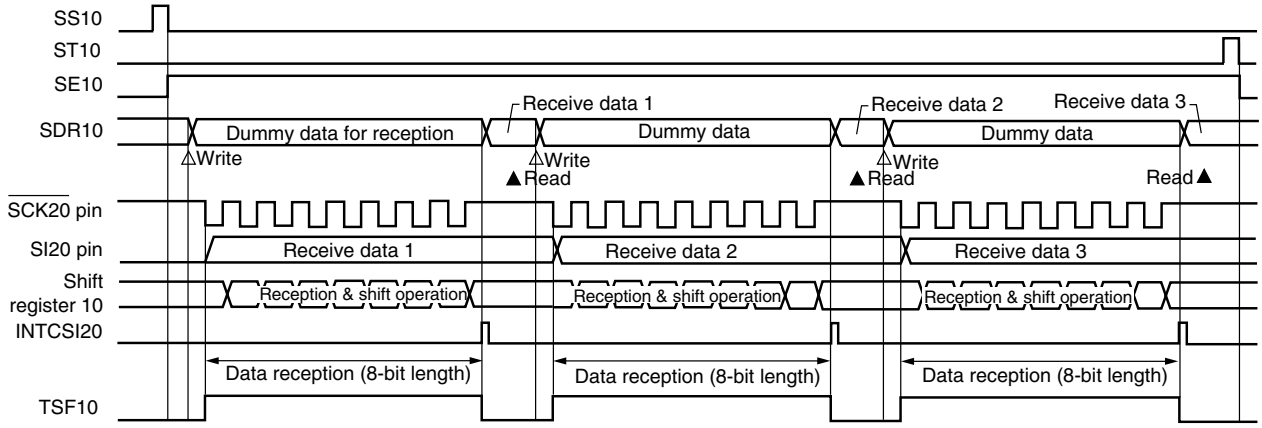
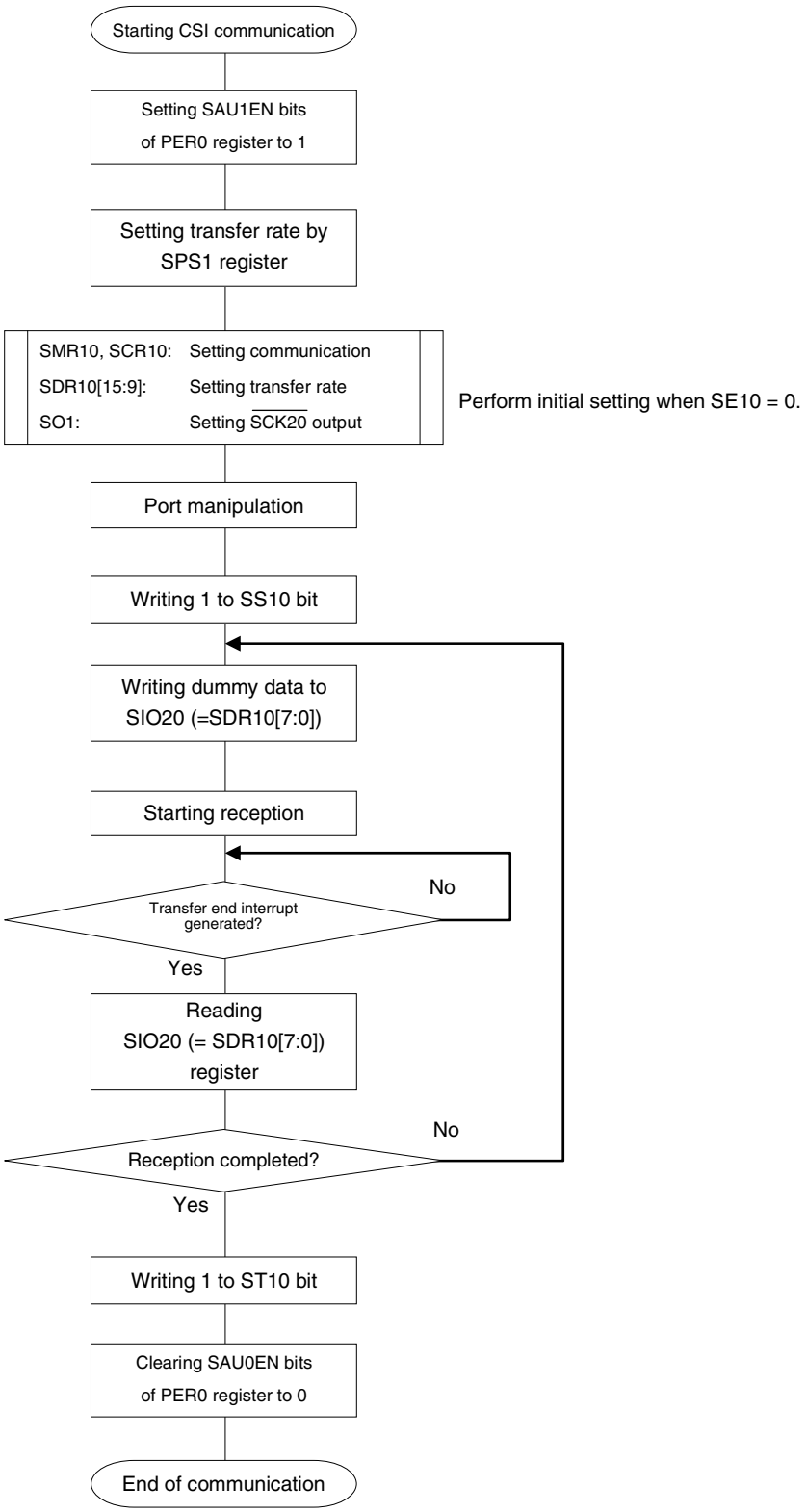


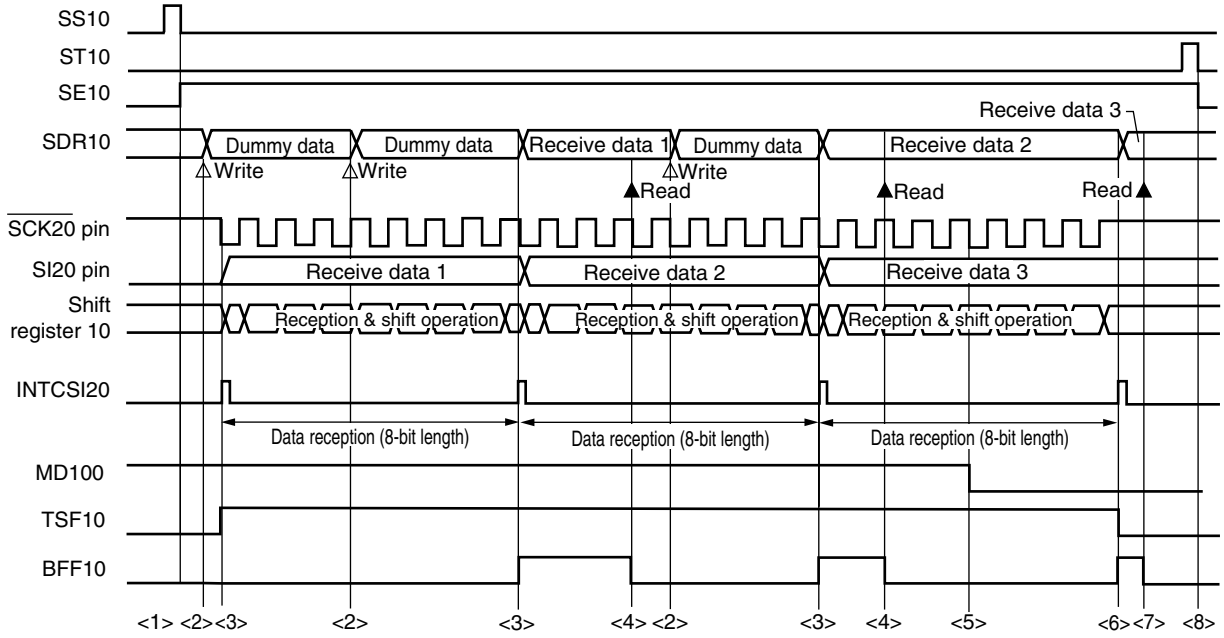
Figure 11-37. Flowchart of Master Reception (in Single-Reception Mode)



Caution After setting the PER0 register to 1, be sure to set the SPS1 register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous reception mode)

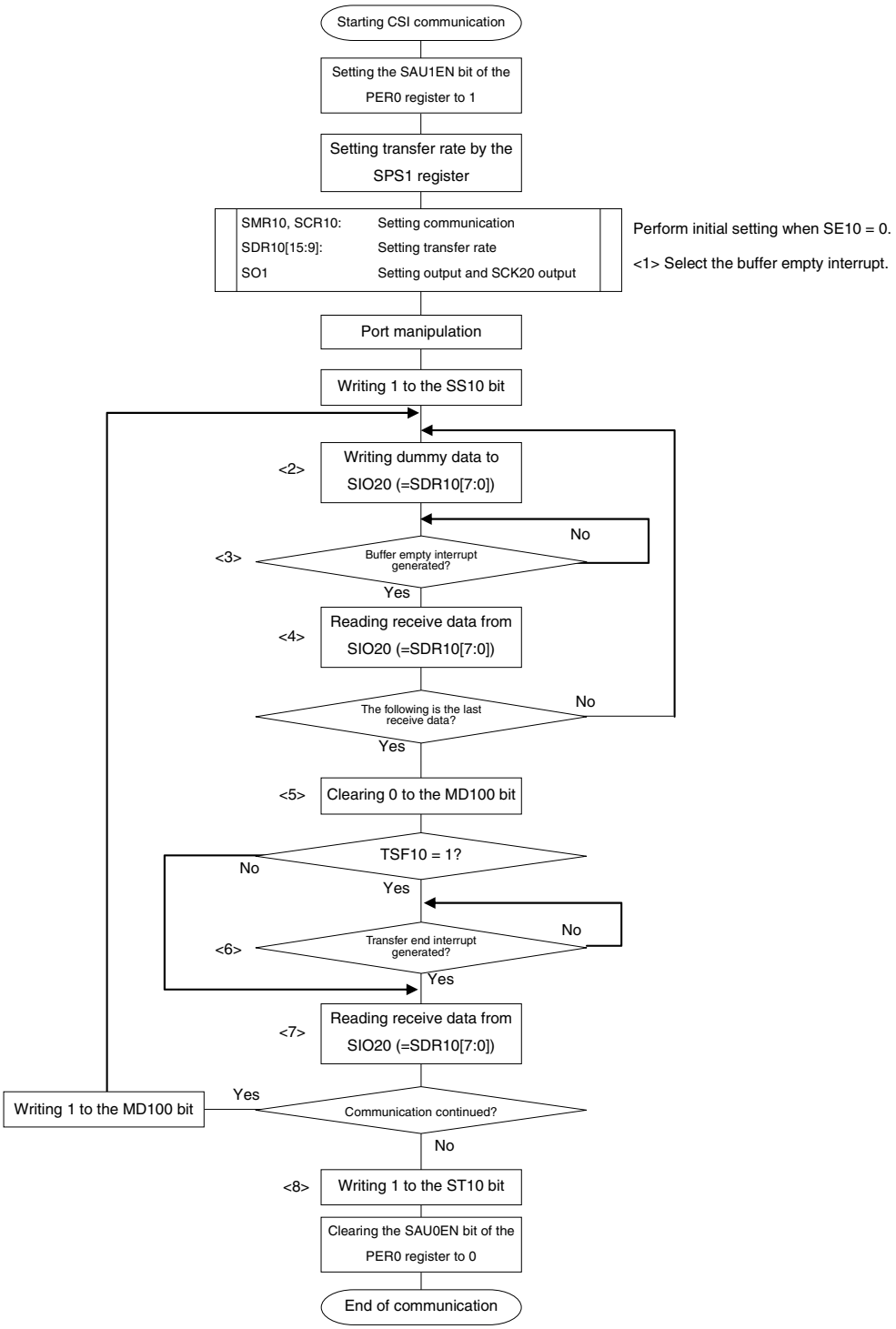
Figure 11-38. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAP10 = 0, CKP10 = 0)



Caution The MD100 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-39 Flowchart of Master Reception (in Continuous Reception Mode)**.

Figure 11-39. Flowchart of Master Reception (in Continuous Reception Mode)



Caution After setting the PER0 register to 1, be sure to set the SPS1 register after 4 or more clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-38 Timing Chart of Master Reception (in Continuous Reception Mode)**.

11.5.3 Master transmission/reception

Master transmission/reception is that the μPD78F8040, 78F8041, 78F8042, 78F8043 output a transfer clock and transmits/receives data to/from other device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{\text{SCK20}}$, SI20, SO20
Interrupt	INTCSI20
	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 (OVF10) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] Note f_{CLK}: System clock frequency
Data phase	Selectable by DAP10 bit • DAP10 = 0: Data I/O starts at the start of the operation of the serial clock. • DAP10 = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKP10 bit • CKP10 = 0: Forward • CKP10 = 1: Reverse
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 26 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-40. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI20)

(a) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10	CCS10	0	0	0	0	0	STS10	0	SIS100	1	0	0	MD102	MD101	MD100
	0/1	0						0		0				0	0	0/1

Operation clock (f_{MCK}) of channel 0
0: Prescaler output clock CK10 set by the SPS1 register
1: Prescaler output clock CK11 set by the SPS1 register

Interrupt source of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10	RXE10	DAP10	CKP10	0	EOC10	PTC101	PTC100	DIR10	0	SLC101	SLC100	0	DLS102	DLS101	DLS100
	1	1	0/1	0/1		0	0	0	0/1		0	0		1	1	0/1

Selection of the data and clock phase (For details about the setting, see **11.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data setting/receive data register							

SIO20

(d) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	1	1	SO10
								0/1								0/1

Communication starts when these bits are 1 if the data phase is forward (CKP10 = 0). If the phase is reversed (CKP10 = 1), communication starts when these bits are 0.

(e) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE10
																0/1

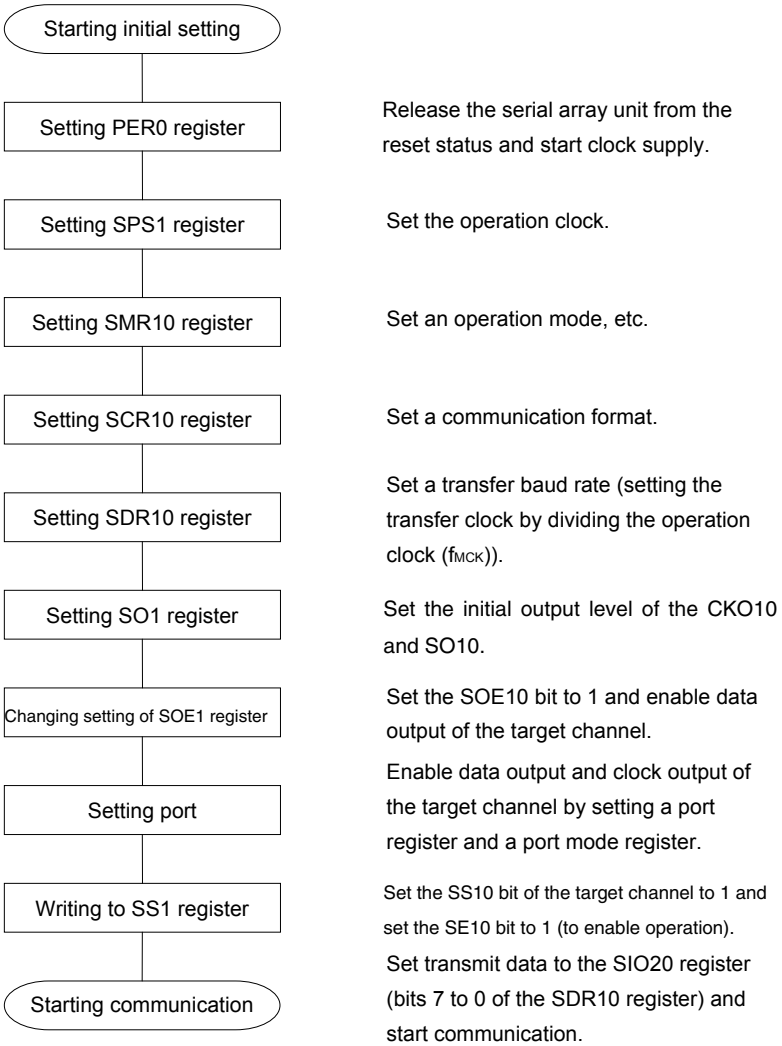
(f) Serial channel start register m (SS1) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1	SSm0
															×	0/1

Remark □: Setting is fixed in the CSI master transmission/reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

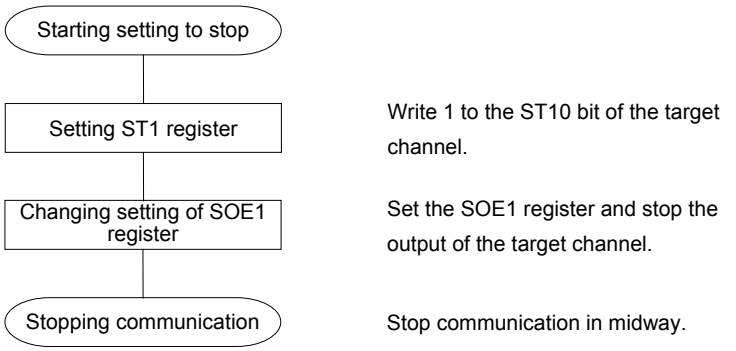
(2) Operation procedure

Figure 11-41. Initial Setting Procedure for Master Transmission/Reception



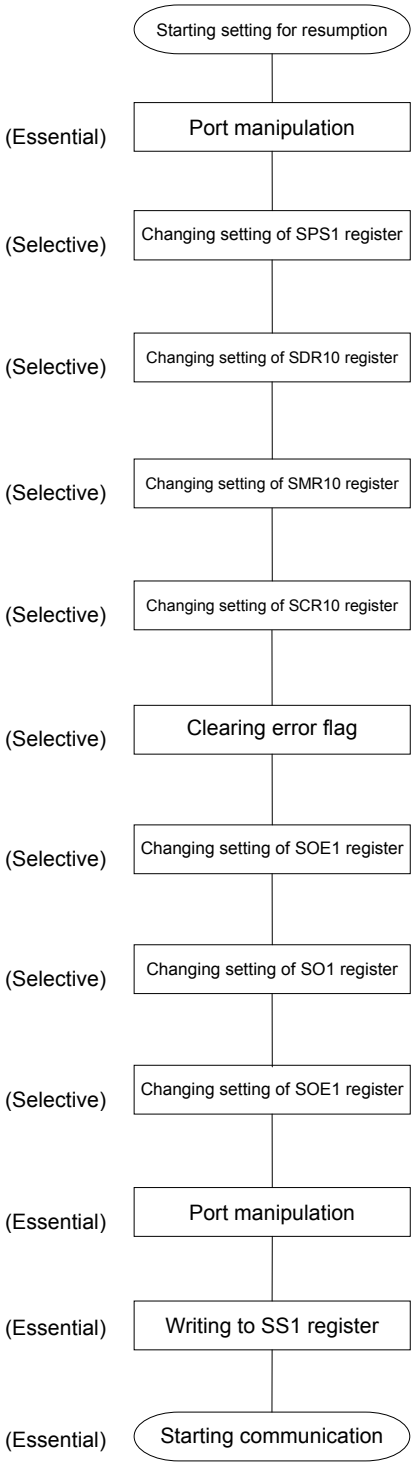
Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

Figure 11-42. Procedure for Stopping Master Transmission/Reception



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO1 register (see **Figure 11-43 Procedure for Resuming Master Transmission/Reception**).

Figure 11-43. 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 SMR10 setting.

Re-set the register to change SCR10 setting.

If the FEF, PEF, and OVF flags remain set, clear them using SIR10.

Set the SOE10 bit to 0 to stop output from the target channel.

Set the initial output level of the CKO10 and SO10.

Set the SOE10 bit to 1 and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SS10 bit of the target channel to 1 and set the SE10 bit to 1 (to enable operation).

Set transmit data to the SIO20 register (bits 7 to 0 of the SDR10 register) and start communication.

(3) Processing flow (in single-transmission/reception mode)

Figure 11-44. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAP10 = 0, CKP10 = 0)

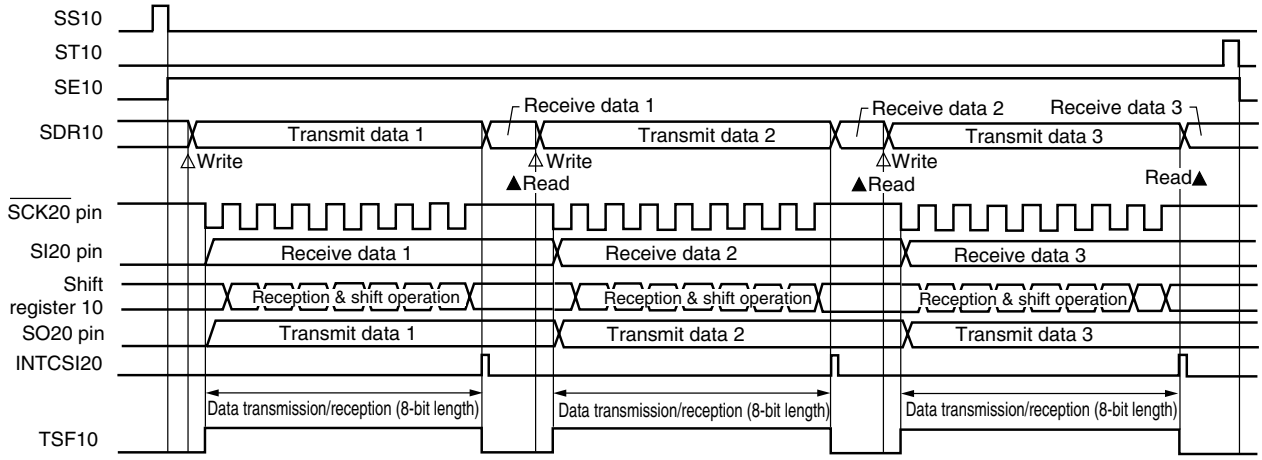
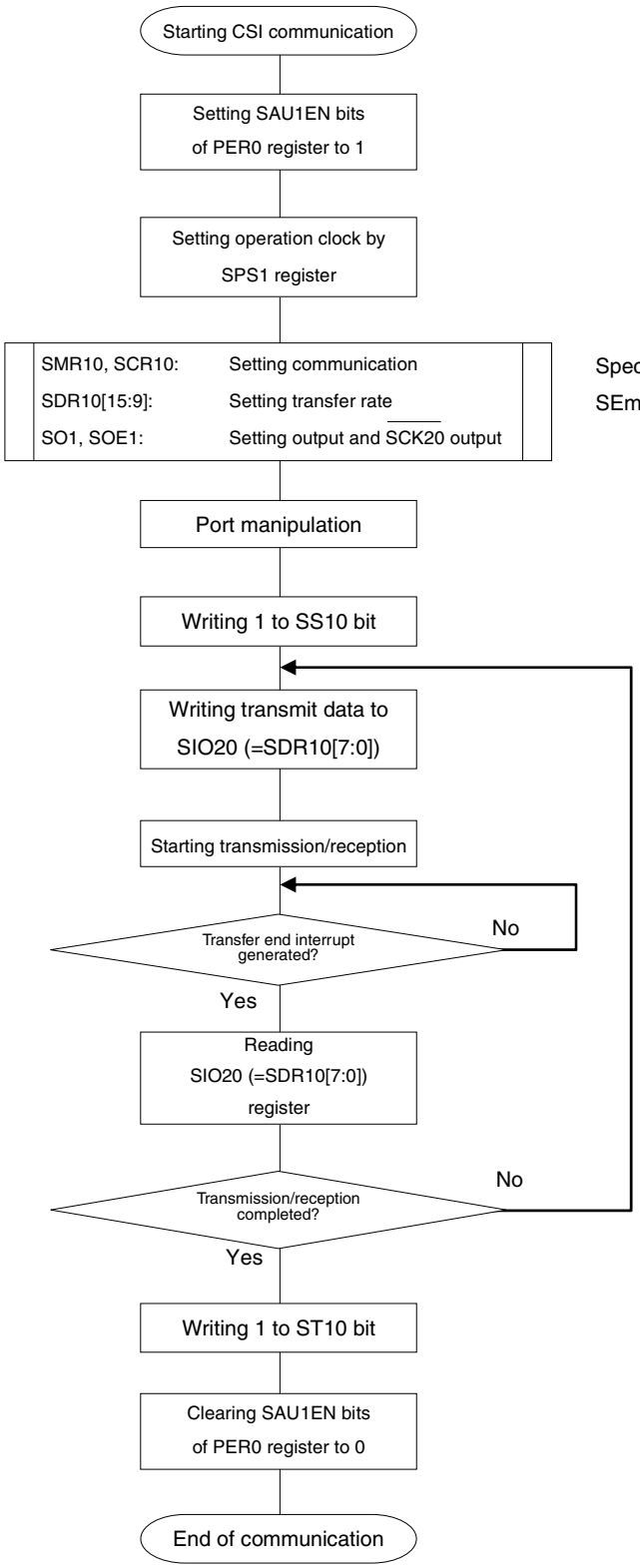


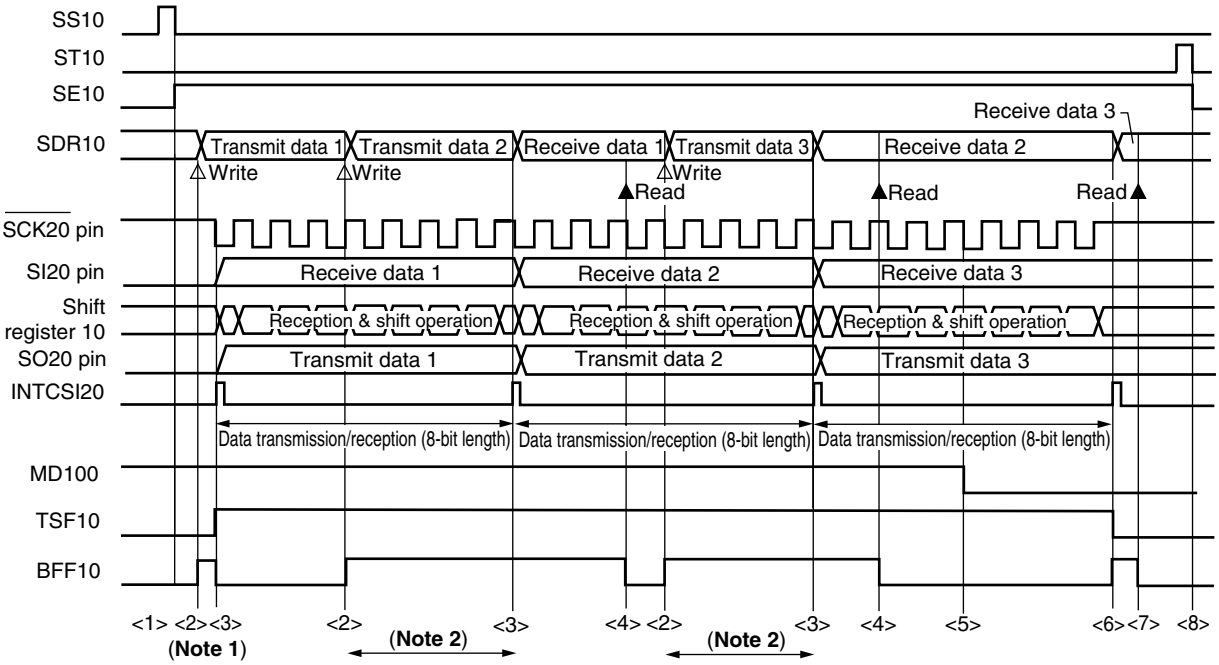
Figure 11-45. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-46. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAP10 = 0, CKP10 = 0)



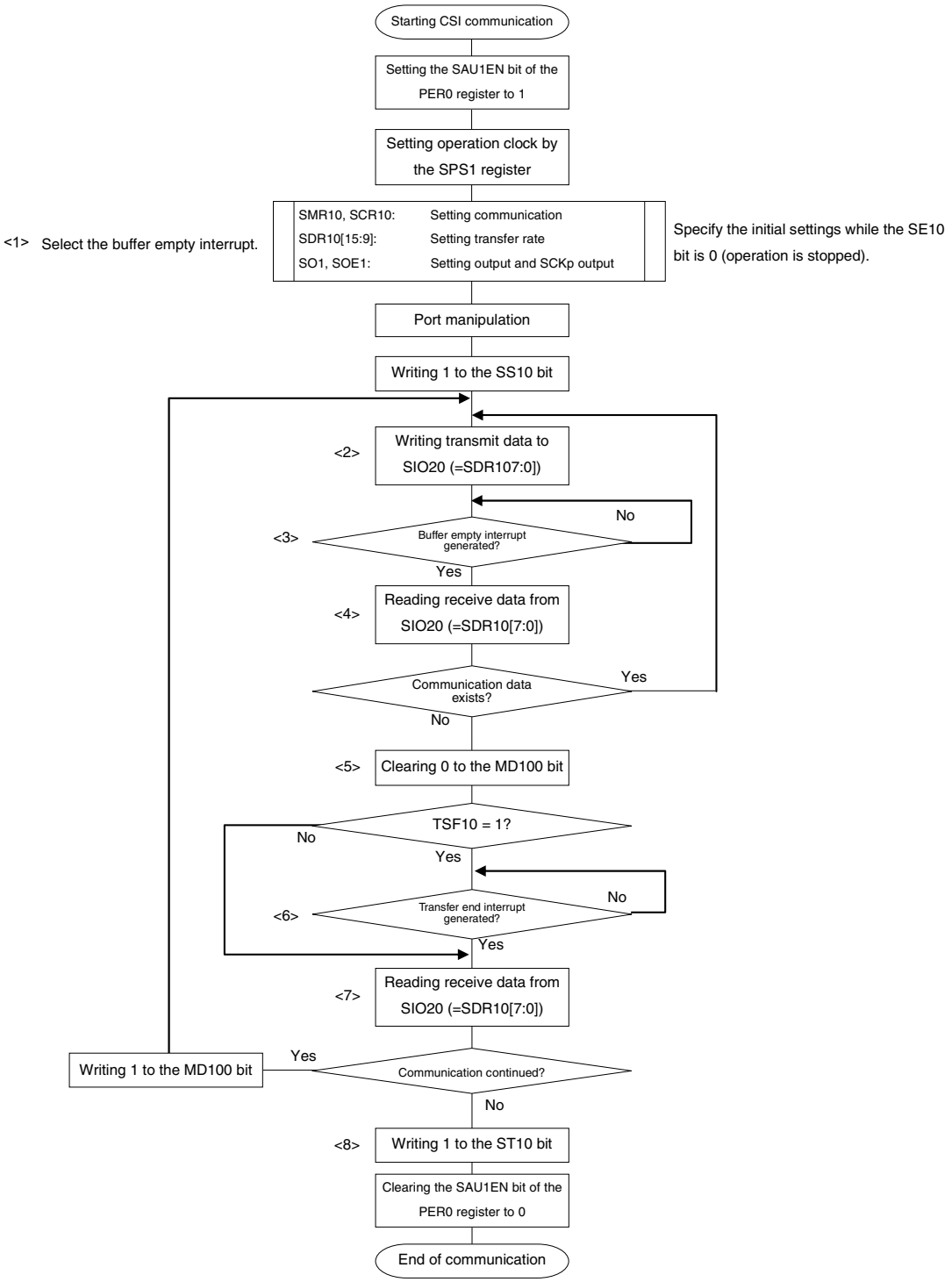
Notes 1. If transmit data is written to the SDR10 register while the BFF10 bit is 1 (valid data is stored in SDR10), the transmit data is overwritten.

2. The transmit data can be read by reading the SDR10 register during this period. At this time, the transfer operation is not affected.

Caution The MD100 bit can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-47 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

Figure 11-47. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-46 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

11.5.4 Slave transmission

Slave transmission is that the μPD78F8040, 78F8041, 78F8042, 78F8043 transmit data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{\text{SCK20}}$, SO20
Interrupt	INTCSI20
	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 (OVF10) only
Transfer data length	7 or 8 bits
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2
Data phase	Selectable by DAP10 bit • DAP10 = 0: Data output starts from the start of the operation of the serial clock. • DAP10 = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKP10 bit • CKP10 = 0: Forward • CKP10 = 1: Reverse
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to pins $\overline{\text{SCK20}}$ is sampled internally and used, the maximum transfer rate is f_{MCK}/6 [Hz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 26 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel

(1) Register setting

Figure 11-48. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI20)

(a) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10	CCS10	0	0	0	0	0	STS10	0	SIS100	1	0	0	MD102	MD101	MD100
	0/1	1						0		0				0	0	0/1

Operation clock (f_{MCK}) of channel 0
0: Prescaler output clock CK10 set by the SPS1 register
1: Prescaler output clock CK11 set by the SPS1 register

Interrupt source of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10	RXE10	DAP10	CKP10		EOC10	PTC101	PTC100	DIR10		SLC101	SLC100		DLS102	DLS101	DLS100
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **11.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Transmit data setting							

SIO20

(d) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	1	1	SO10
								×								0/1

(e) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE10
																0/1

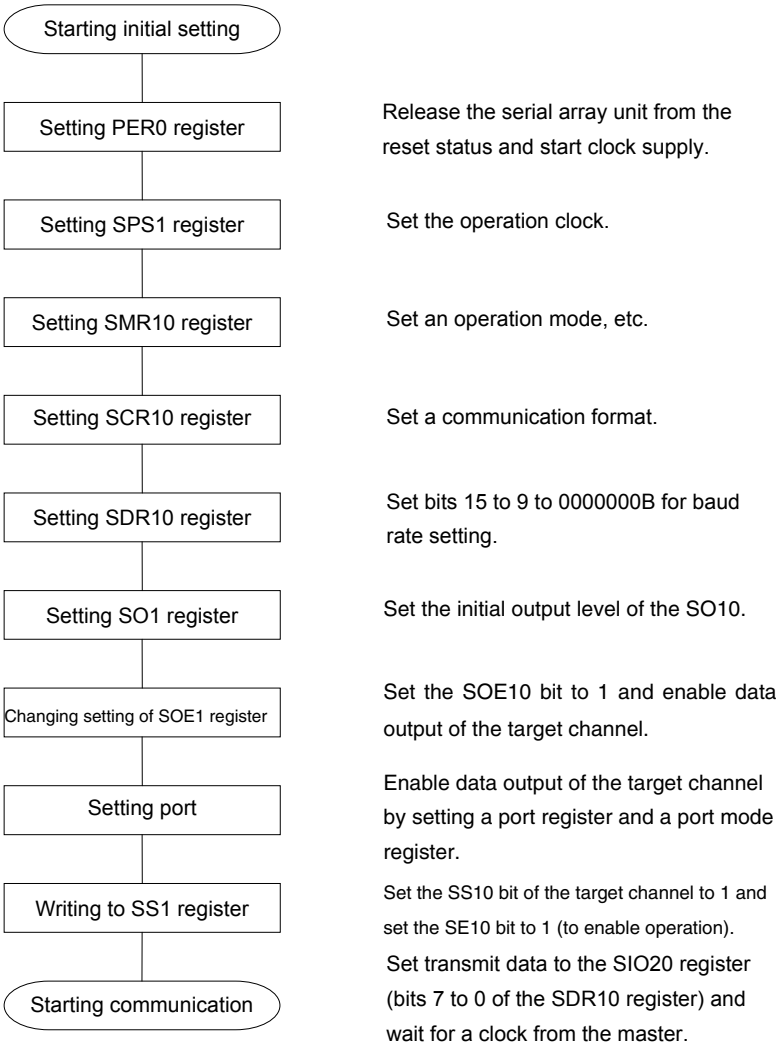
(f) Serial channel start register 1 (SS1) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS11	SS10
															×	0/1

Remark □: Setting is fixed in the CSI slave transmission mode, ■ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

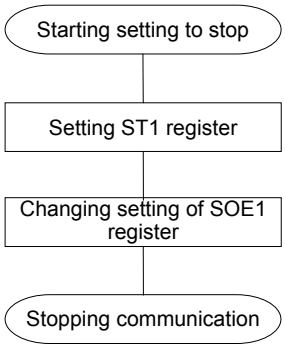
(2) Operation procedure

Figure 11-49. Initial Setting Procedure for Slave Transmission



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

Figure 11-50. Procedure for Stopping Slave Transmission



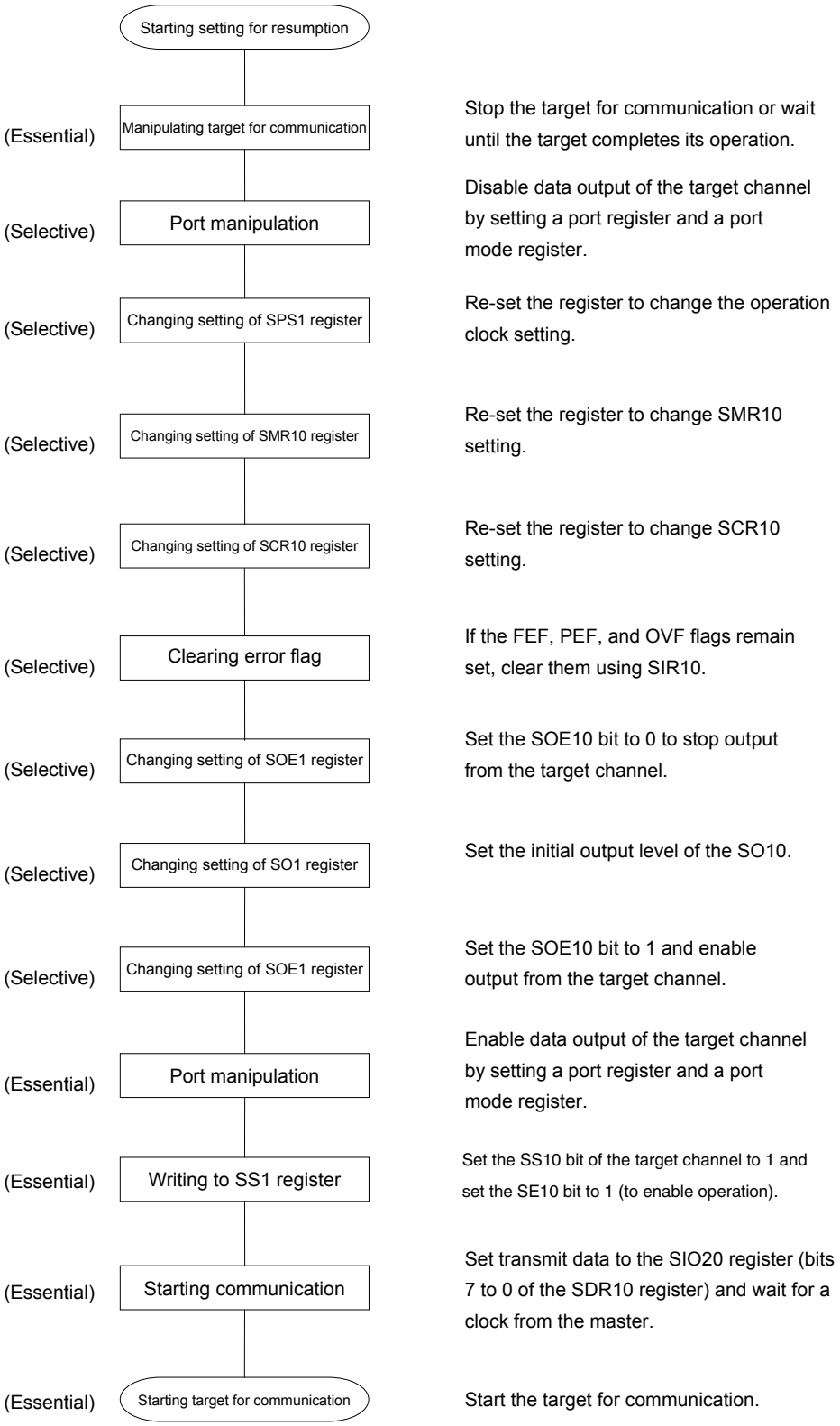
Write 1 to the ST10 bit of the target channel.

Set the SOE1 register and stop the output of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO1 register (see **Figure 11-51 Procedure for Resuming Slave Transmission**).

Figure 11-51. Procedure for Resuming Slave Transmission



(3) Processing flow (in single-transmission mode)

Figure 11-52. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAP10 = 0, CKP10 = 0)

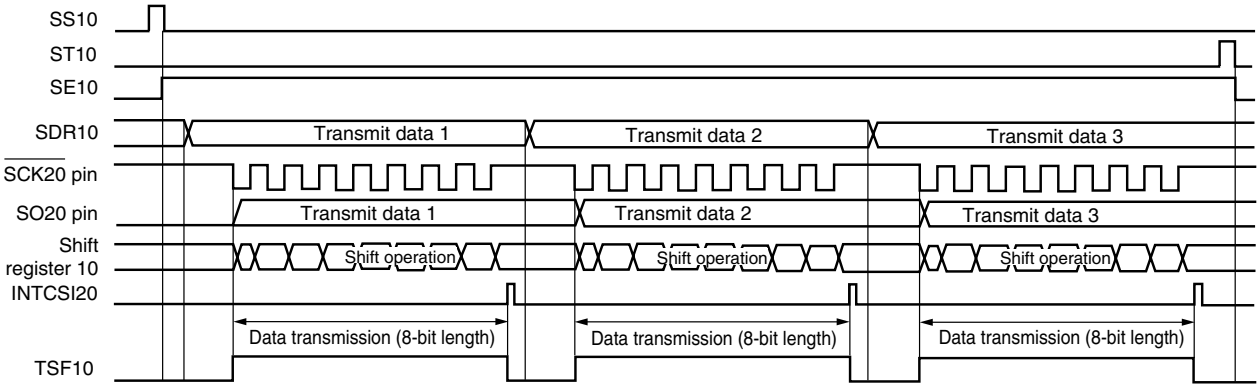
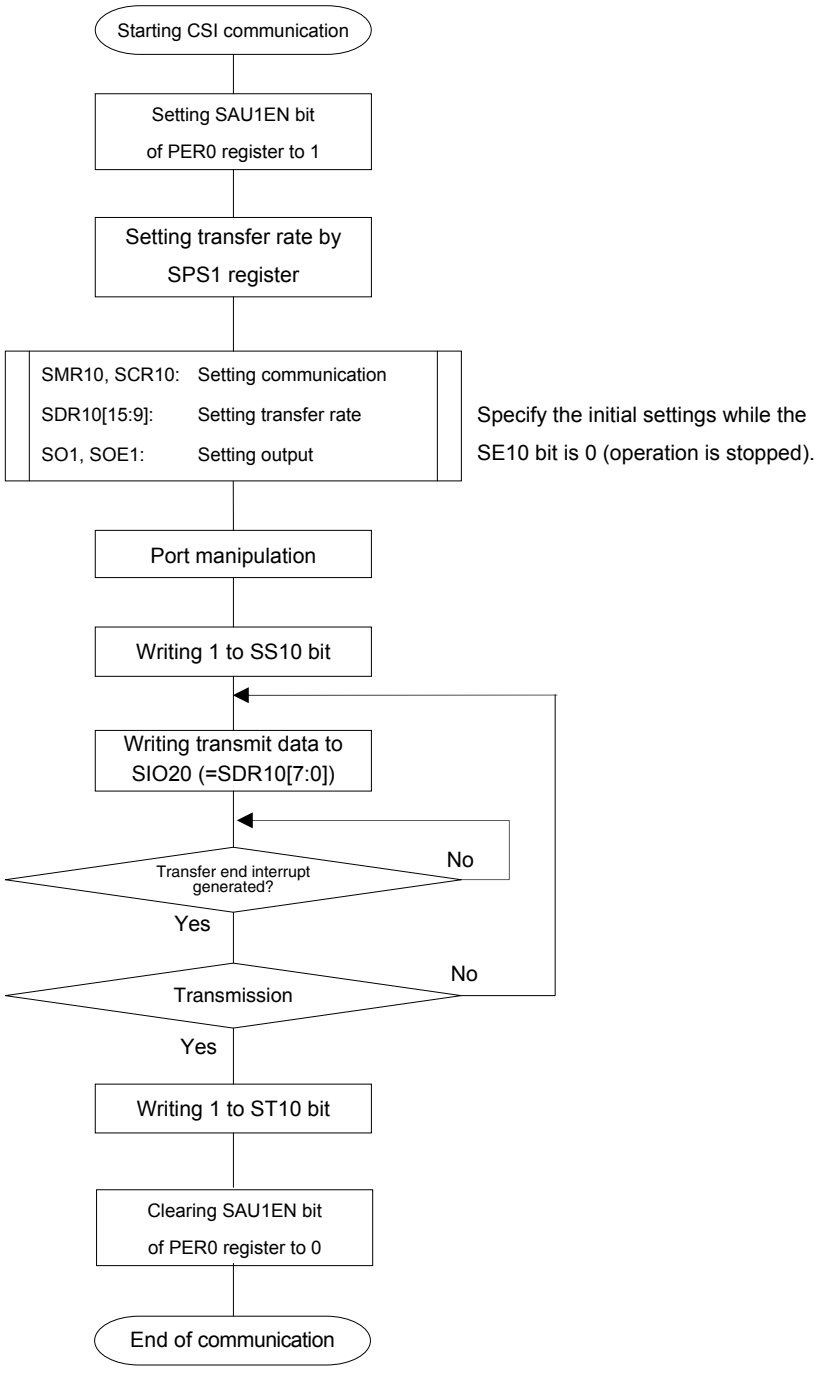


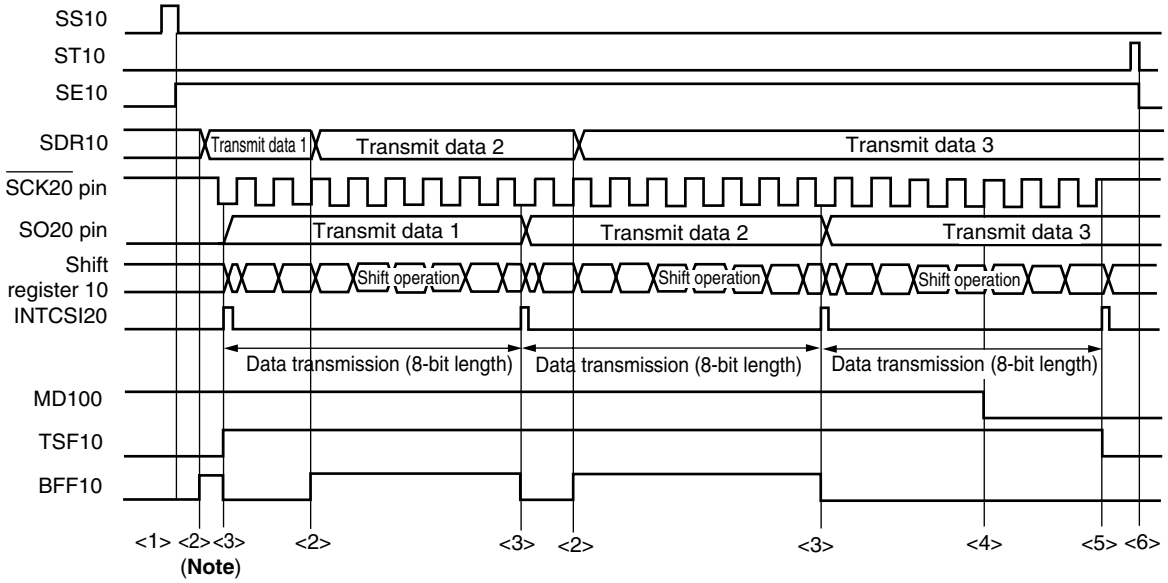
Figure 11-53. Flowchart of Slave Transmission (in Single-Transmission Mode)



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

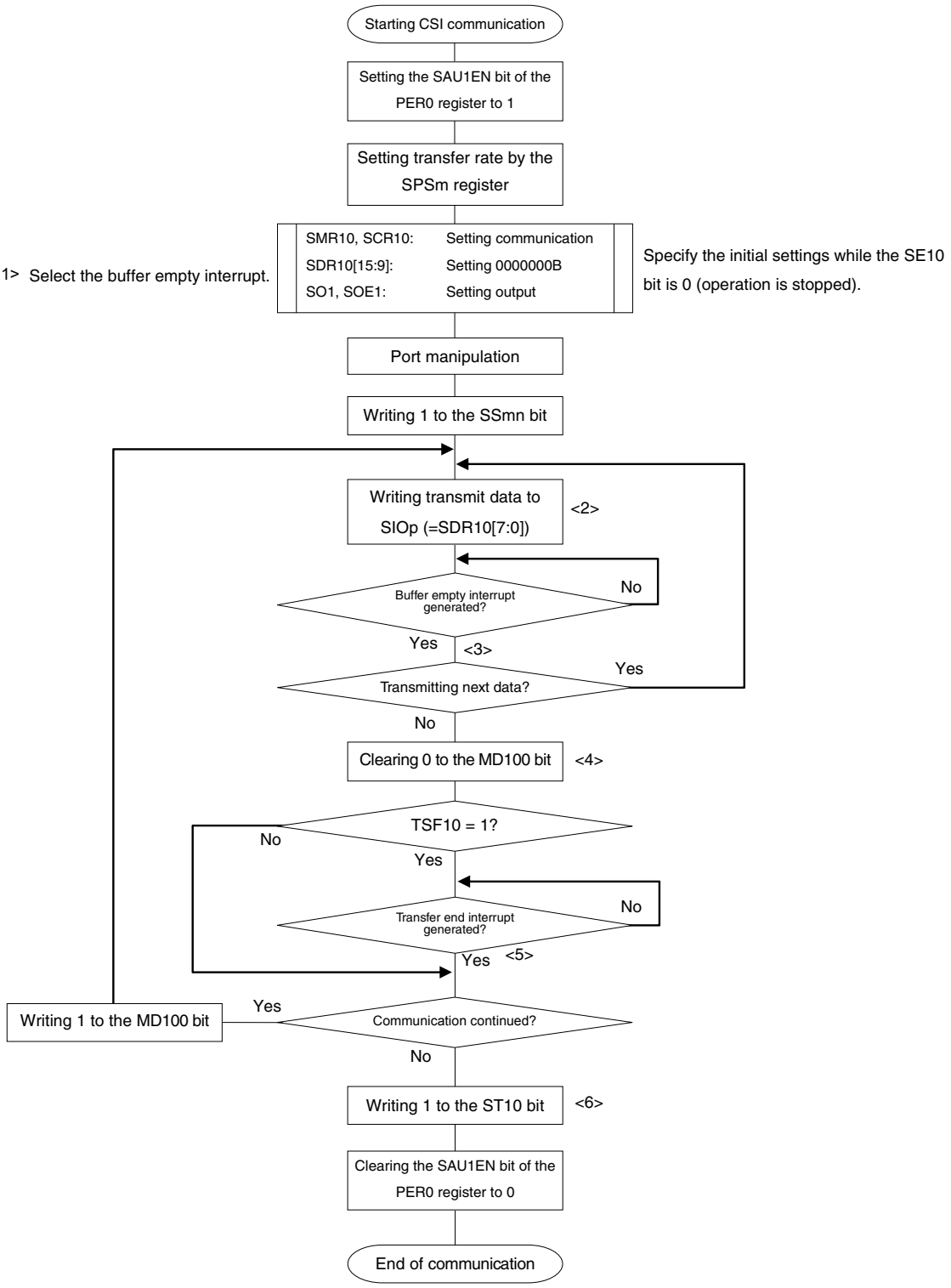
Figure 11-54. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAP10 = 0, CKP10 = 0)



Note If transmit data is written to the SDR10 register while the BFF10 bit is 1 (valid data is stored in serial data register mn (SDR10)), the transmit data is overwritten.

Caution The MD100 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Figure 11-55. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-54 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

11.5.5 Slave reception

Slave reception is that the μPD78F8040, 78F8041, 78F8042, 78F8043 receive data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{SCK20}$, SI20
Interrupt	INTCSI20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Overrun error detection flag (OVF10) only
Transfer data length	7 or 8 bits
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2
Data phase	Selectable by DAP10 bit • DAP10 = 0: Data input starts from the start of the operation of the serial clock. • DAP10 = 1: Data input starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKP10 bit • CKP10 = 0: Forward • CKP10 = 1: Reverse
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to pins $\overline{SCK20}$ is sampled internally and used, the maximum transfer rate is f_{MCK}/6 [Hz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 26 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel

(1) Register setting

Figure 11-56. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI20)

(a) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10	CCS10	0	0	0	0	0	STS10	0	SIS100	1	0	0	MD102	MD101	MD100
	0/1	1						0		0				0	0	0

Operation clock (f_{MCK}) of channel 0
0: Prescaler output clock CK10 set by the SPS1 register
1: Prescaler output clock CK11 set by the SPS1 register

Interrupt source of channel 0
0: Transfer end interrupt

(b) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10	RXE10	DAP10	CKP10		EOC10	PTC101	PTC100	DIR10		SLC101	SLC100		DLS102	DLS101	DLS100
	0	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **11.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	0000000 (baud rate setting)							0	Receive data register							

SIO20

(d) Serial output register 1 (SO1) … The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	1	1	SO10
								×								×

(e) Serial output enable register 1 (SOE1) … The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE10
																×

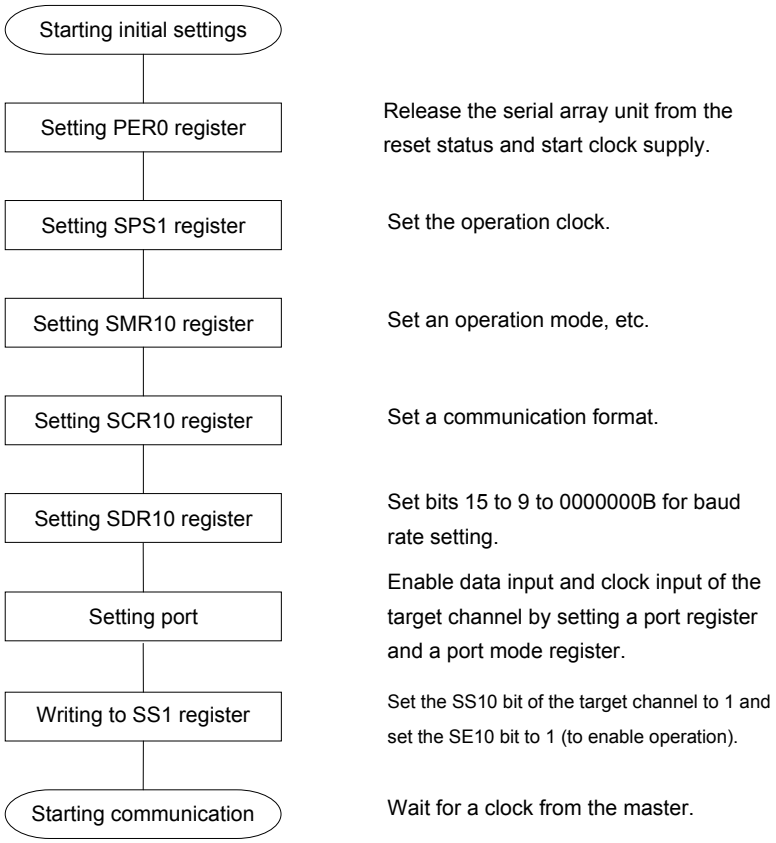
(f) Serial channel start register 1 (SS1) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS11	SS10
															×	0/1

Remark ☐: Setting is fixed in the CSI slave reception mode, ▒ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-57. Initial Setting Procedure for Slave Reception



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

Figure 11-58. Procedure for Stopping Slave Reception

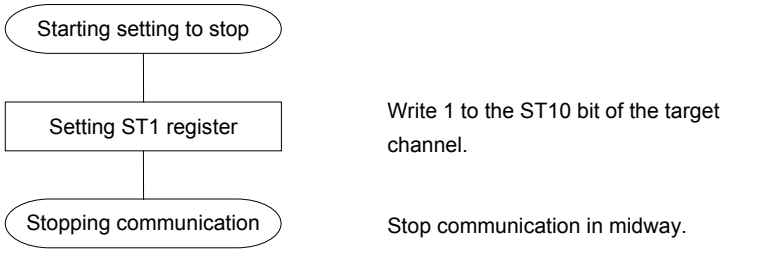
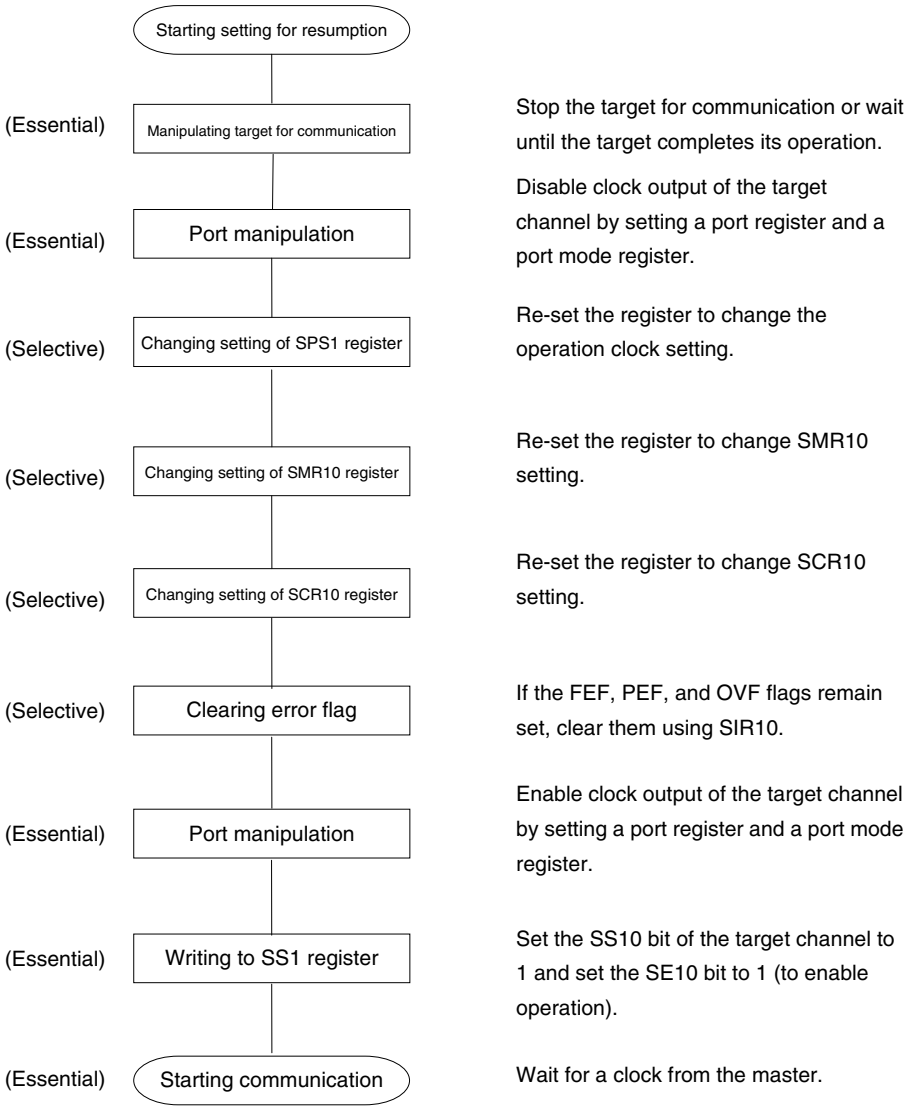


Figure 11-59. Procedure for Resuming Slave Reception



(3) Processing flow (in single-reception mode)

Figure 11-60. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAP10 = 0, CKP10 = 0)

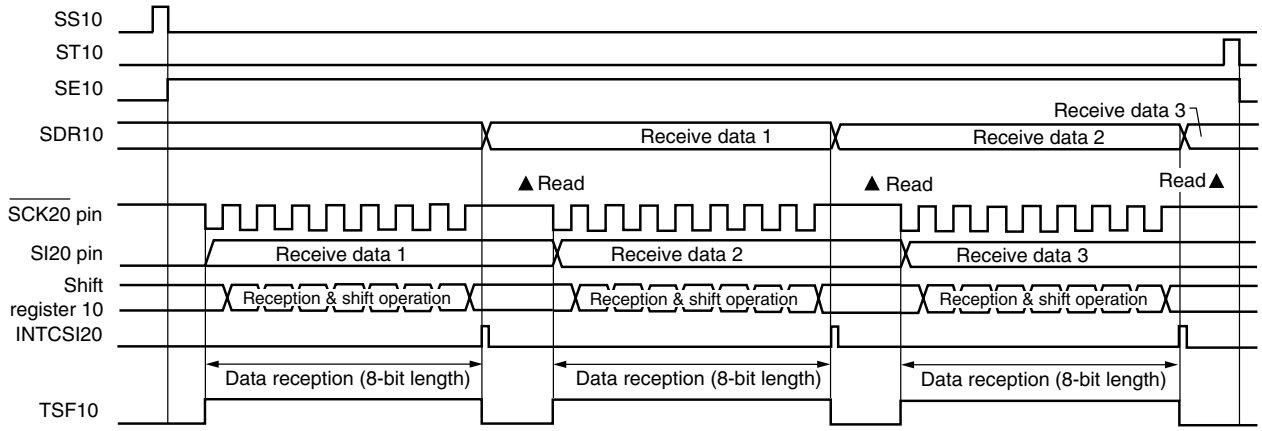
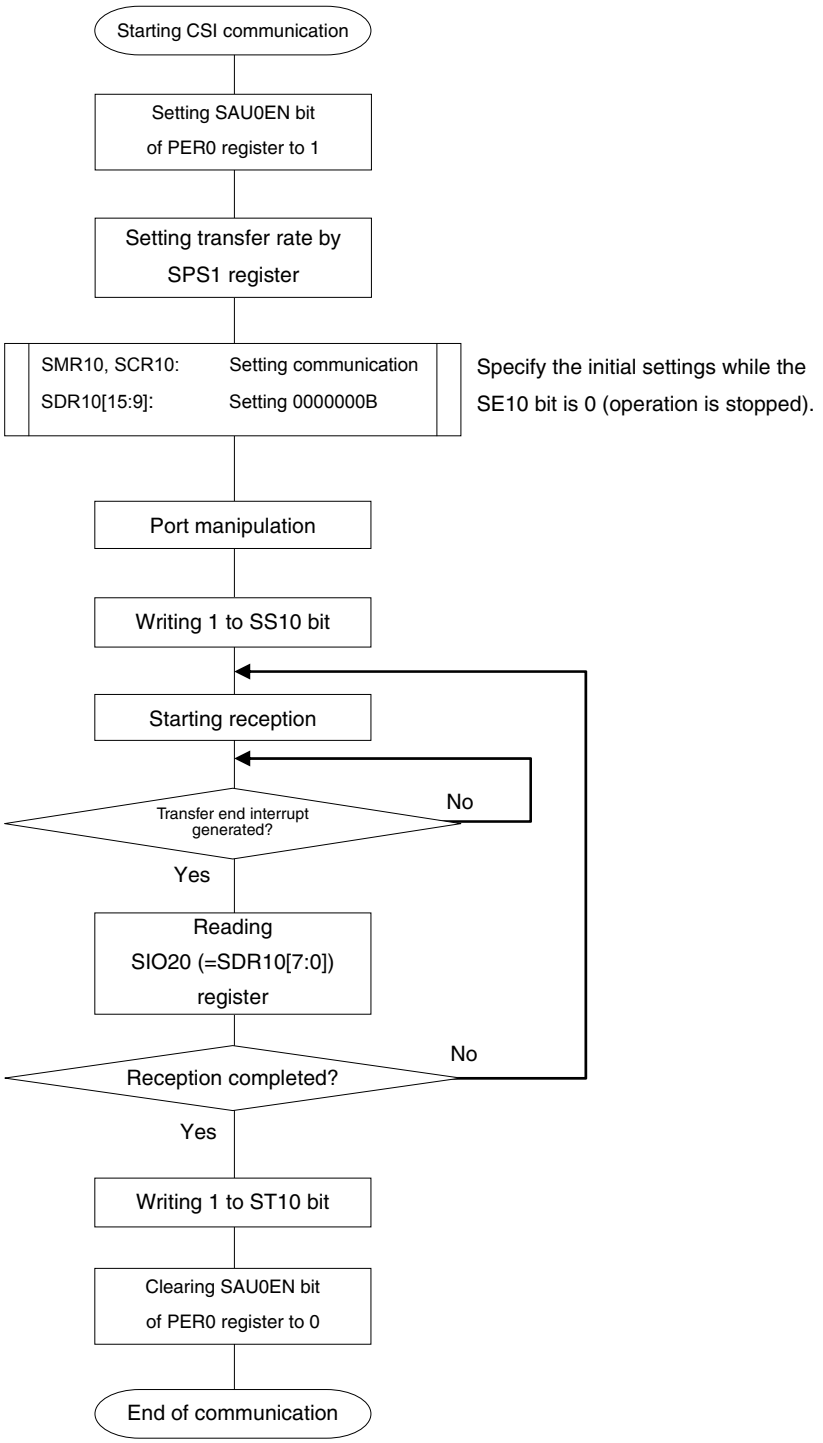


Figure 11-61. Flowchart of Slave Reception (in Single-Reception Mode)



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

11.5.6 Slave transmission/reception

Slave transmission/reception is that the μPD78F8040, 78F8041, 78F8042, 78F8043 transmit/receive data to/from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI20
Target channel	Channel 0 of SAU1
Pins used	$\overline{SCK20}$, SI20, SO20
Interrupt	INTCSI20
	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 (OVF10) only
Transfer data length	7 or 8 bits
Transfer rate	Max. f_{MCK}/6 [Hz]Notes 1, 2
Data phase	Selectable by DAP10 bit • DAP10 = 0: Data I/O starts from the start of the operation of the serial clock. • DAP10 = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by CKP10 bit • CKP10 = 0: Forward • CKP10 = 1: Reverse
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to pins $\overline{SCK20}$ is sampled internally and used, the maximum transfer rate is f_{MCK}/6 [MHz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 26 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel

(1) Register setting

Figure 11-62. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI20)

(a) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10	CCS10	0	0	0	0	0	STS10	0	SIS100	1	0	0	MD102	MD101	MD100
	0/1	1						0		0				0	0	0/1

Operation clock (f_{MCK}) of channel 0
0: Prescaler output clock CK10 set by the SPS1 register
1: Prescaler output clock CK11 set by the SPS1 register

Interrupt source of channel 0
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10	RXE10	DAP10	CKP10		EOC10	PTC101	PTC100	DIR10		SLC101	SLC100		DLS102	DLS101	DLS100
	1	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **11.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	0000000 (baud rate setting)							0	Transmit data setting/receive data register							

SIO20

(d) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	1	1	SO10
								×								0/1

(e) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE10
																0/1

(f) Serial channel start register 1 (SS1) … Sets only the bits of the target channel to 1.

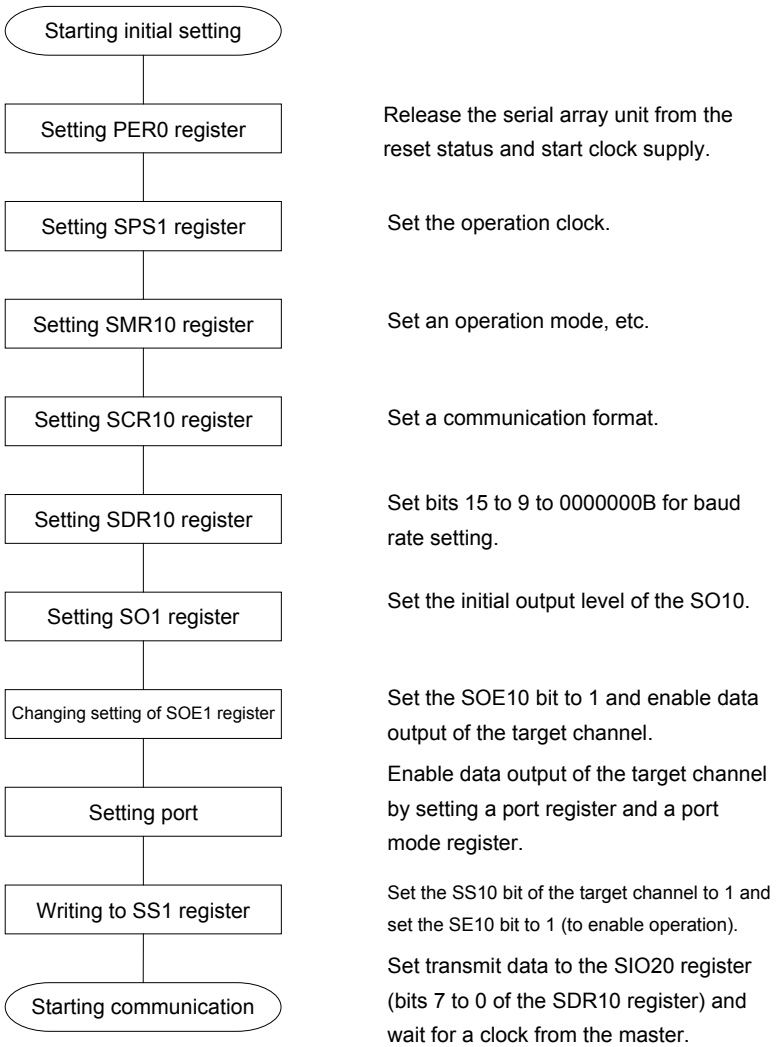
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS11	SS10
															×	0/1

Caution Be sure to set transmit data to the SIO20 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

(2) Operation procedure

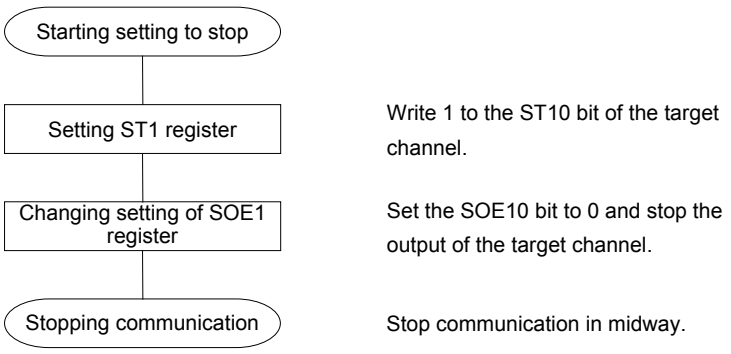
Figure 11-63. Initial Setting Procedure for Slave Transmission/Reception



Cautions 1. After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

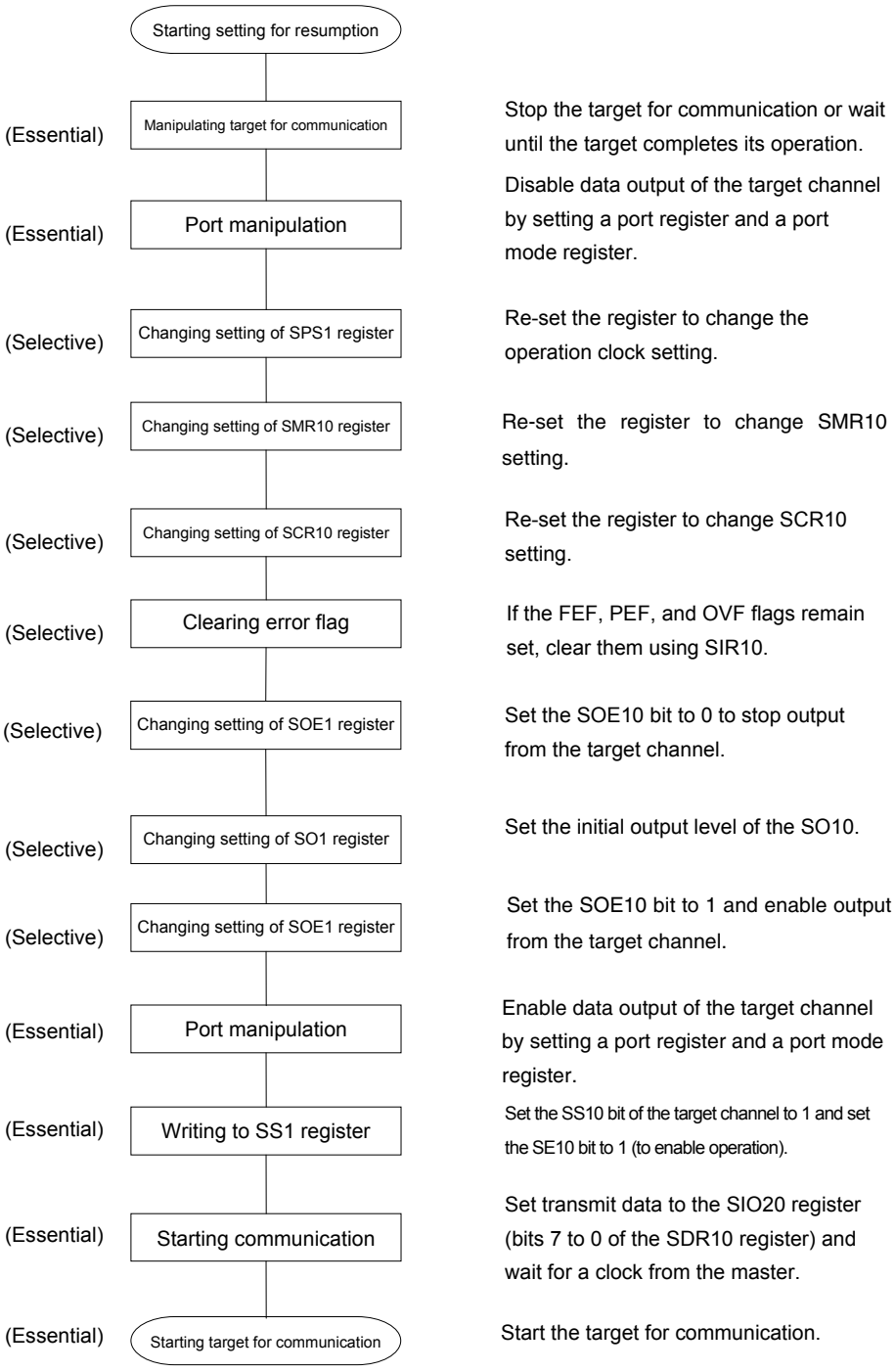
2. Be sure to set transmit data to the SIO20 register before the clock from the master is started.

Figure 11-64. Procedure for Stopping Slave Transmission/Reception



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO1 register (see **Figure 11-65 Procedure for Resuming Slave Transmission/Reception**).

Figure 11-65. Procedure for Resuming Slave Transmission/Reception



Caution Be sure to set transmit data to the SIO20 register before the clock from the master is started.

(3) Processing flow (in single-transmission/reception mode)

Figure 11-66. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAP10 = 0, CKP10 = 0)

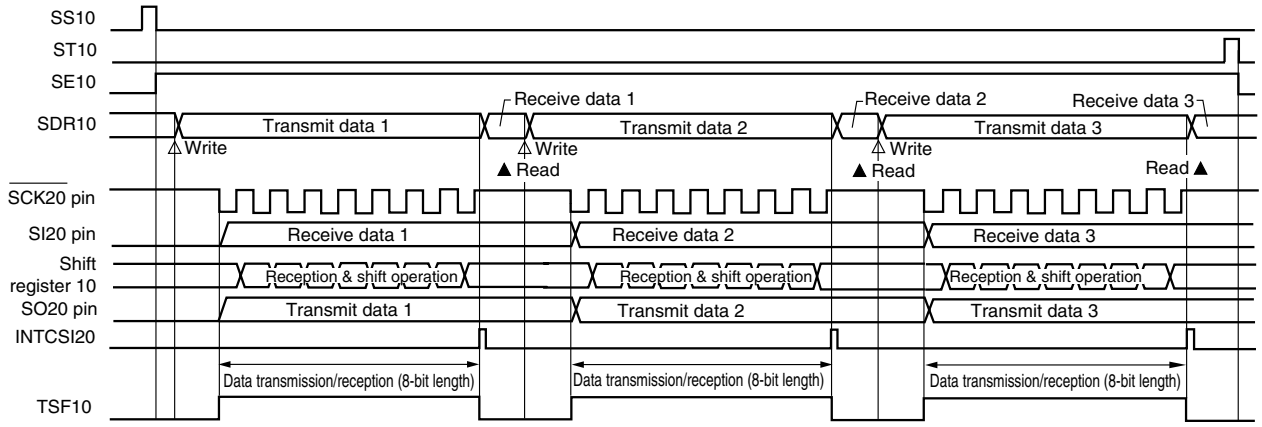
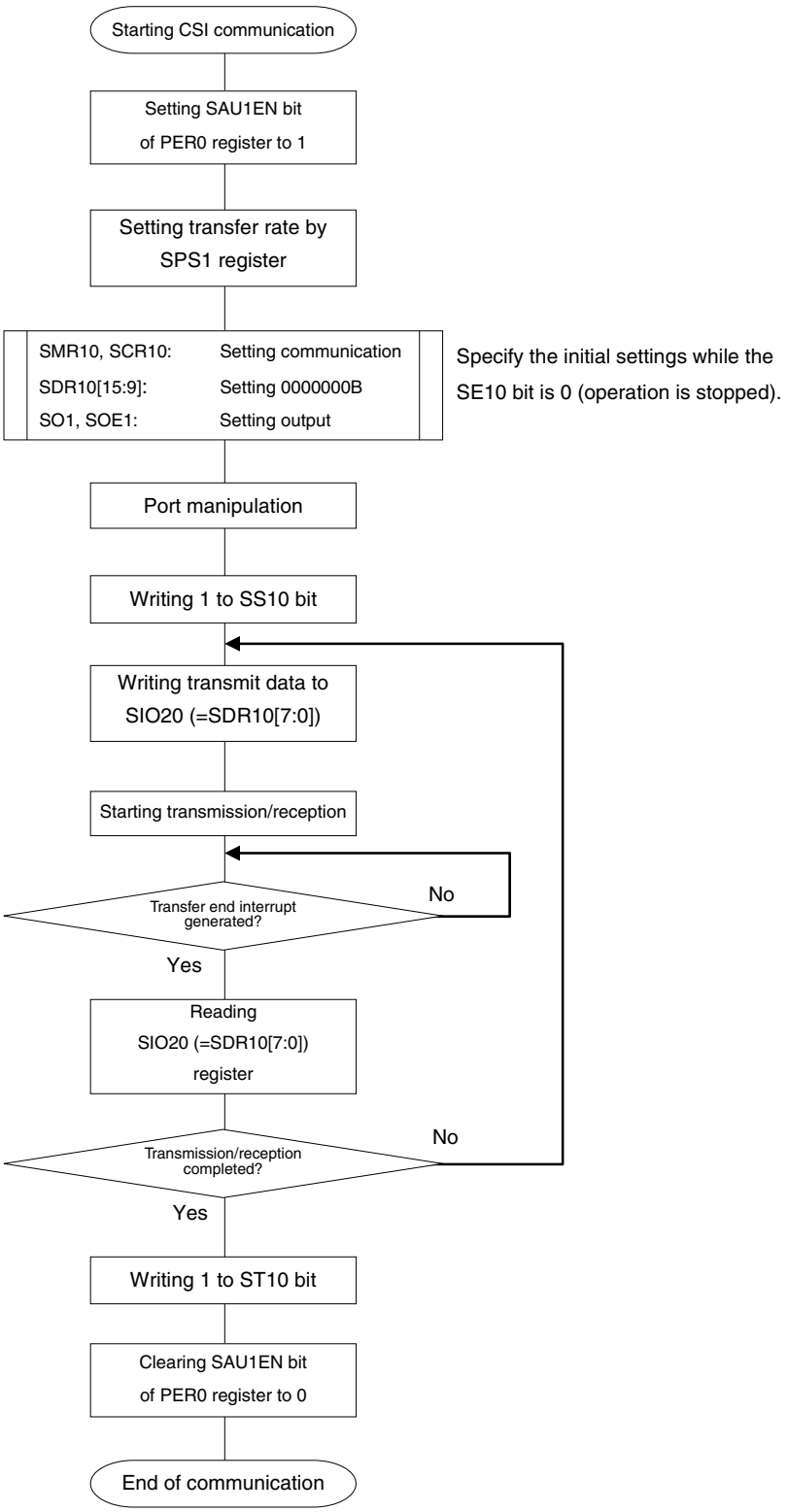


Figure 11-67. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

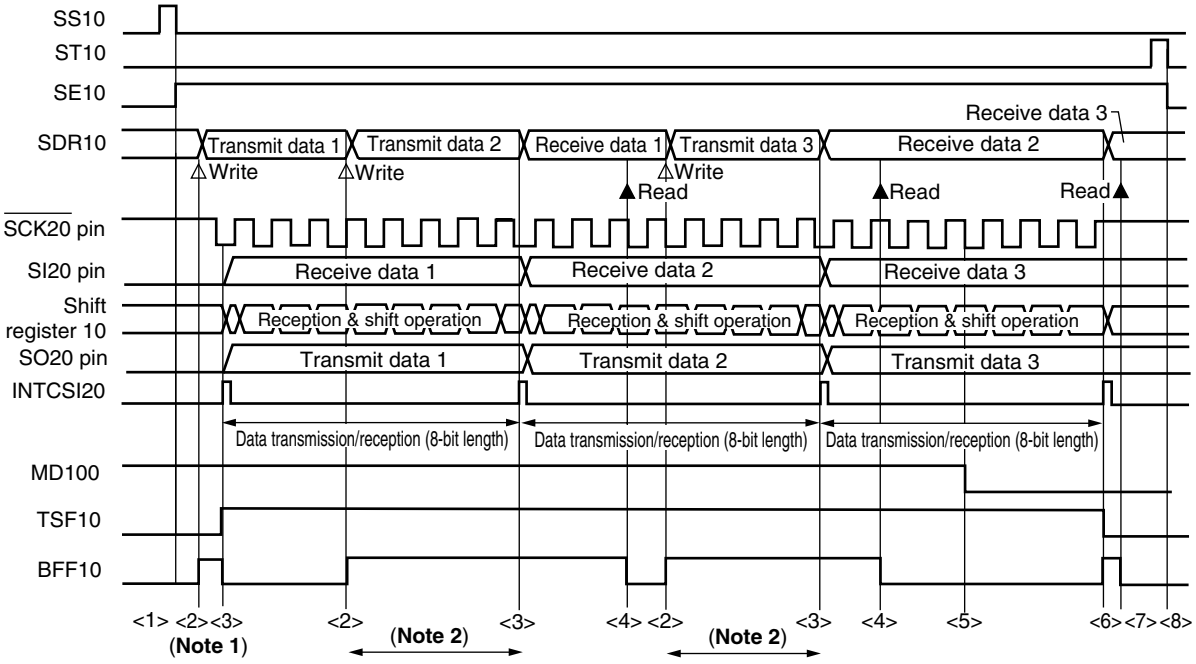


Cautions 1. After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

2. Be sure to set transmit data to the SIO20 register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-68. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAP10 = 0, CKP10 = 0)



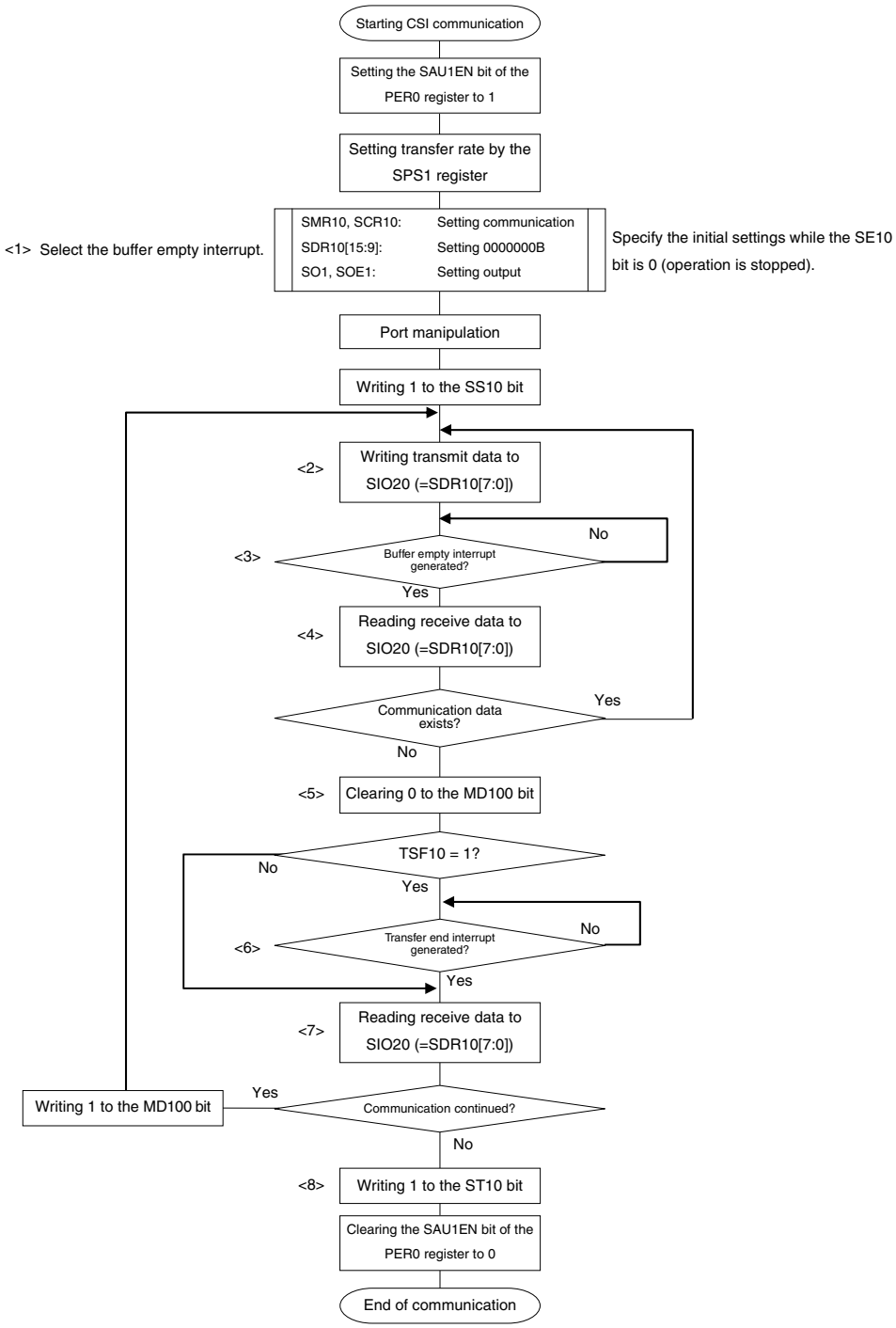
Notes 1. If transmit data is written to the SDR10 register while the BFF10 bit is 1 (valid data is stored in SDR10), the transmit data is overwritten.

2. The transmit data can be read by reading the SDR10 register during this period. At this time, the transfer operation is not affected.

Caution The MD100 bit 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.

Remarks <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-69 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

Figure 11-69. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Cautions 1. After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

2. Be sure to set transmit data to the SIO20 register before the clock from the master is started.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-68 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

11.5.7 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI20) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDR10[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 SDR10[15:9] is the value of bits 15 to 9 of the SDR10 register (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register 1 (SPS1) and bit 15 (CKS10) of serial mode register 10 (SMR10).

Table 11-2. Selection of operation clock

SMR10 Register	SPS1 Register								Operation Clock (f_{MCK}) Note	
CKS10	PRS 113	PRS 112	PRS 111	PRS 110	PRS 103	PRS 102	PRS 101	PRS 100		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM03	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM03	
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 (STm = 000FH) the operation of the serial array unit (SAU). When selecting INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

Remark X: Don't care

11.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSI20) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI20) communication is described in Figure 11-70.

Figure 11-70. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads SDR10 register. →	The BFF10 bit 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 SSR10 register.		Error type is identified and the read value is used to clear error flag.
Writes 1 to SIR10 register. →	Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSR10 register to the SIR10 register without modification.

11.6 Operation of UART (UART0, UART2, UART3) 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 5, 7, or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]
- Framing error, parity error, or overrun error

The LIN-bus is supported in UART3 (2, 3 channels of unit 1)

[LIN-bus functions]
- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

} External interrupt (INTP0) or timer array unit (TAU0) is used.

Caution UART0 (0 and 1 channels of unit 0) is dedicated to IO-Link communication (see 11.7.3 to 11.7.5).

UART0 uses channels 0 and 1 of SAU0.
UART2 uses channels 0 and 1 of SAU1.
UART3 uses channels 2 and 3 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	–	UART0 (dedicated to IO-Link communication)	–
	1	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

Caution When using serial array units 0 and 1 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 **11.6.1**.)
- UART reception (See **11.6.2**.)
- LIN transmission (UART3 only) (See **11.7.1**.)
- LIN reception (UART3 only) (See **11.7.2**.)
- IO-Link transmission/reception (UART0 only) (See **11.7.5**.)

11.6.1 UART transmission

UART transmission is an operation to transmit data from the μPD78F8040, 78F8041, 78F8042, 78F8043 to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART2	UART3
Target channel	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	TxD2	TxD3
Interrupt	INTST2	INTST3
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	5, 7, or 8 bits	
Transfer rate	Max. f_{MCK}/6 [bps] (SDR1n [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 • 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 26 ELECTRICAL SPECIFICATIONS**).

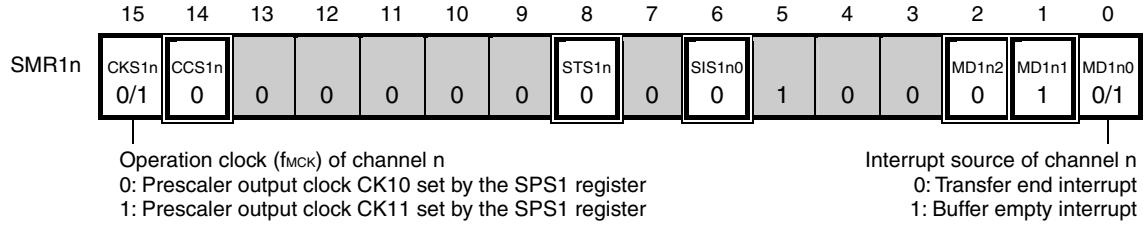
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 0, 2)

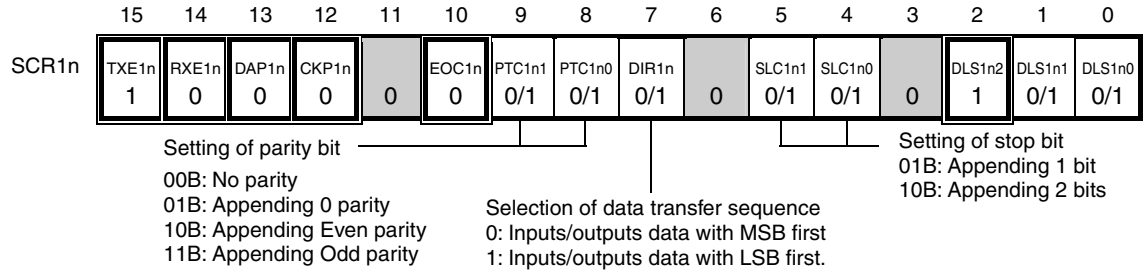
(1) Register setting

Figure 11-71. Example of Contents of Registers for UART Transmission of UART (UART2, UART3) (1/2)

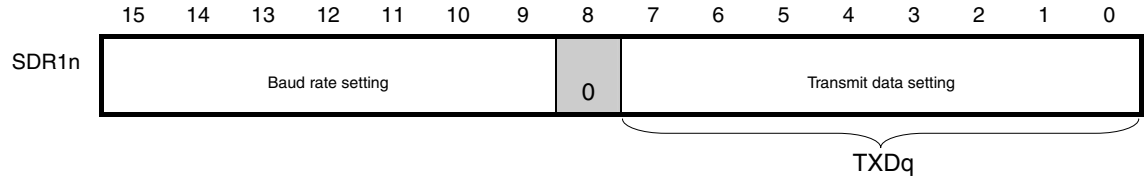
(a) Serial mode register 1n (SMR1n)



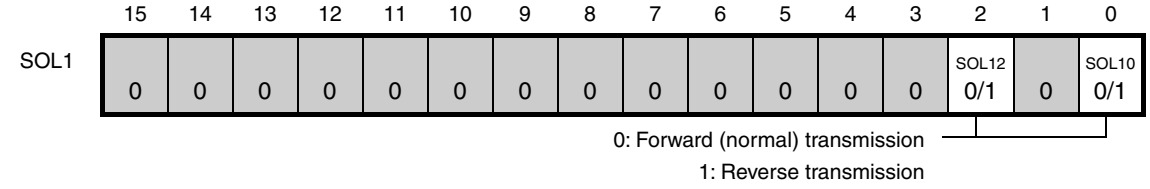
(b) Serial communication operation setting register 1n (SCR1n)



(c) Serial data register 1n (SDR1n) (lower 8 bits: TXDq)



(d) Serial output level register 1 (SOL1) … Sets only the bits of the target channel.



Note Before transmission is started, be sure to set to 1 when the SOL1n bit of the target channel is set to 0, and set to 0 when the SOL1n bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remark n: Channel number (n = 0, 2)

□: Setting is fixed in the UART transmission mode, ■ : Setting disabled (fixed by hardware)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-71. Example of Contents of Registers for UART Transmission of UART (UART2, UART3) (2/2)

(e) Serial output register 1 (SO1) … Sets only the bits of the target channel.

SO1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	CKO10 ×	0	0	0	0	1	SO12 0/1Note	1	SO10 0/1Note

0: Serial data output value is “0”
1: Serial data output value is “1”

(f) Serial output enable register 1 (SOE1) … Sets only the bits of the target channel to 1.

SOE1

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	SOE12 0/1	0	SOE10 0/1

(g) Serial channel start register 1 (SS1) … Sets only the bits of the target channel to 1.

SS1

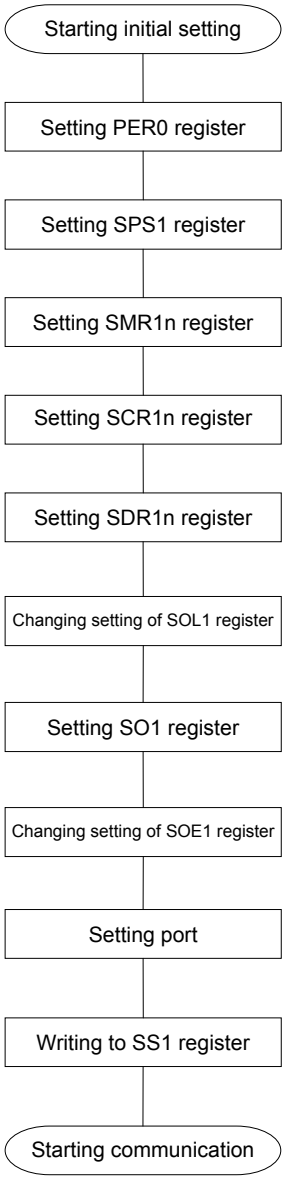
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SS13 ×	SS12 0/1	SS11 ×	SS10 0/1

Note Before transmission is started, be sure to set to 1 when the SOL1n bit of the target channel is set to 0, and set to 0 when the SOL1n bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remark n: Channel number (n = 0, 2)
□: Setting is fixed in the UART transmission mode, ▒ : Setting disabled (fixed by hardware)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-72. Initial Setting Procedure for UART Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set an output data level.

Set the initial output level of the SO1n bit.

Set the SOE1n bit to 1 and enable data output of the target channel.

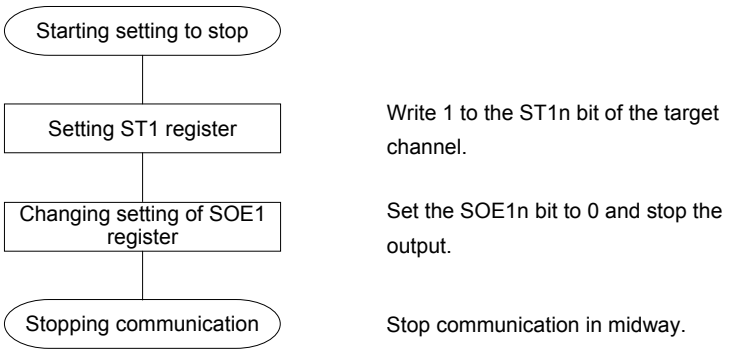
Enable data output of the target channel by setting a port register and a port mode register.

Set the SS1n bit of the target channel to 1 and set the SE1n bit to 1 (to enable operation).

Set transmit data to the TXDq register (bits 7 to 0 of the SDR1n register) and start communication.

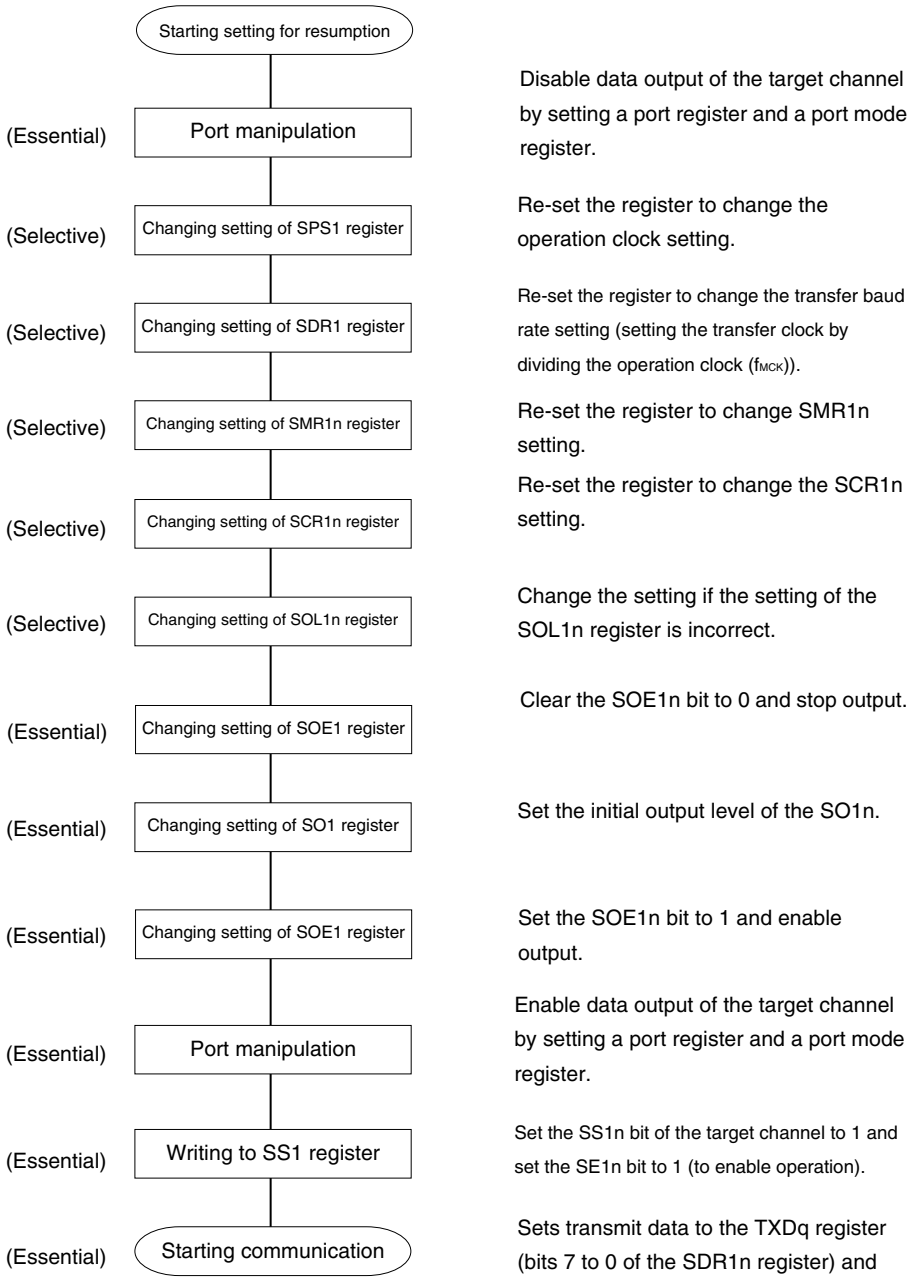
Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

Figure 11-73. Procedure for Stopping UART Transmission



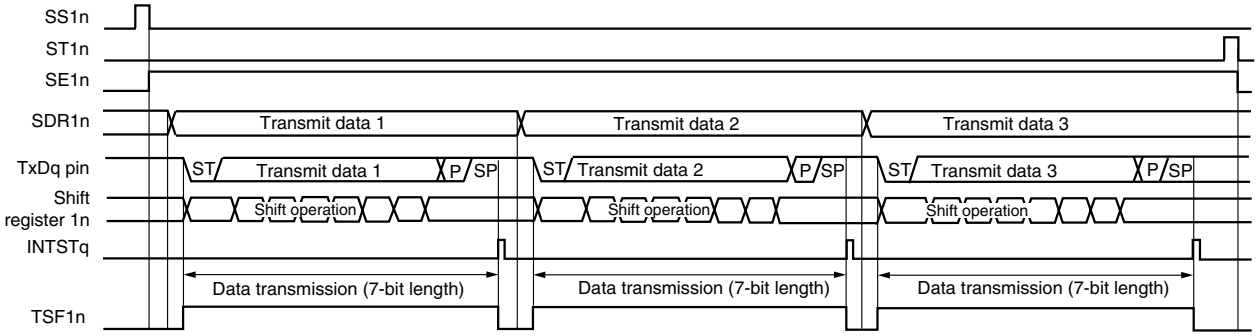
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO1 register (see **Figure 11-74 Procedure for Resuming UART Transmission**).

Figure 11-74. Procedure for Resuming UART Transmission



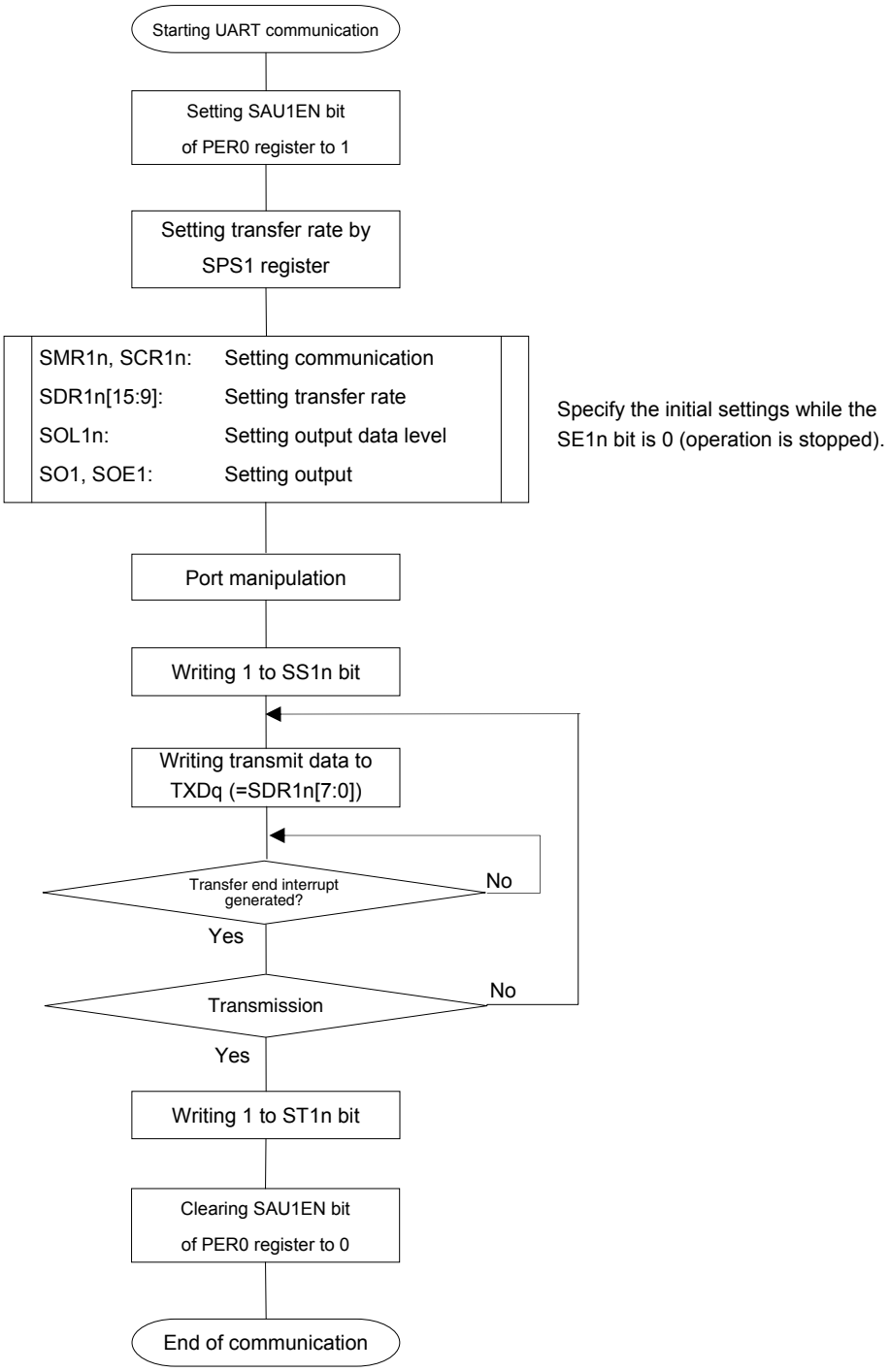
(3) Processing flow (in single-transmission mode)

Figure 11-75. Timing Chart of UART Transmission (in Single-Transmission Mode)



Remark n: Channel number (n = 0, 2), q: UART number (q = 2, 3)

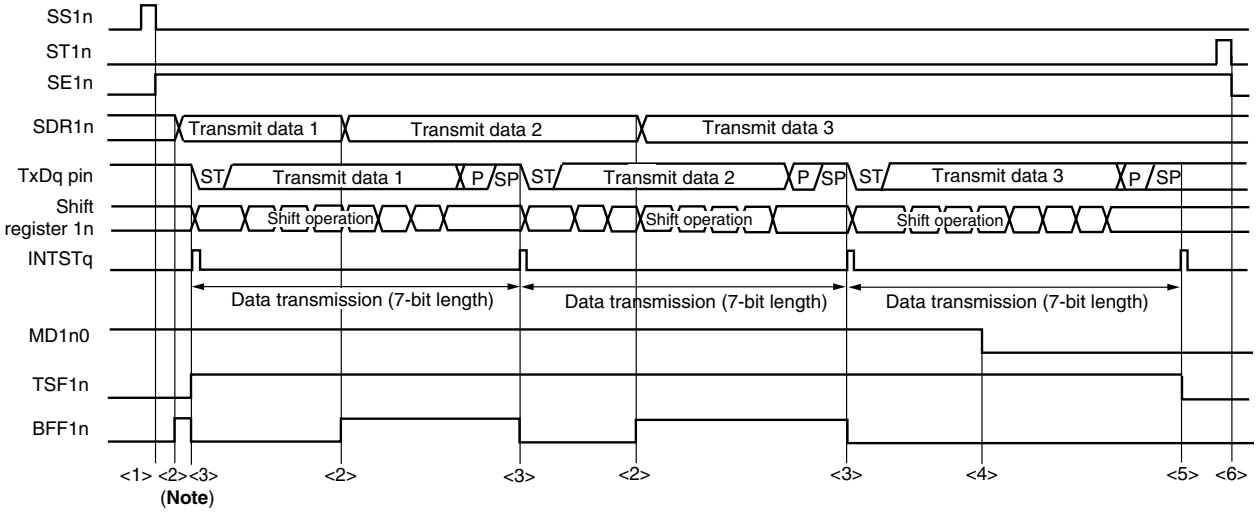
Figure 11-76. Flowchart of UART Transmission (in Single-Transmission Mode)



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set serial clock select register 1 (SPS1) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-77. Timing Chart of UART Transmission (in Continuous Transmission Mode)

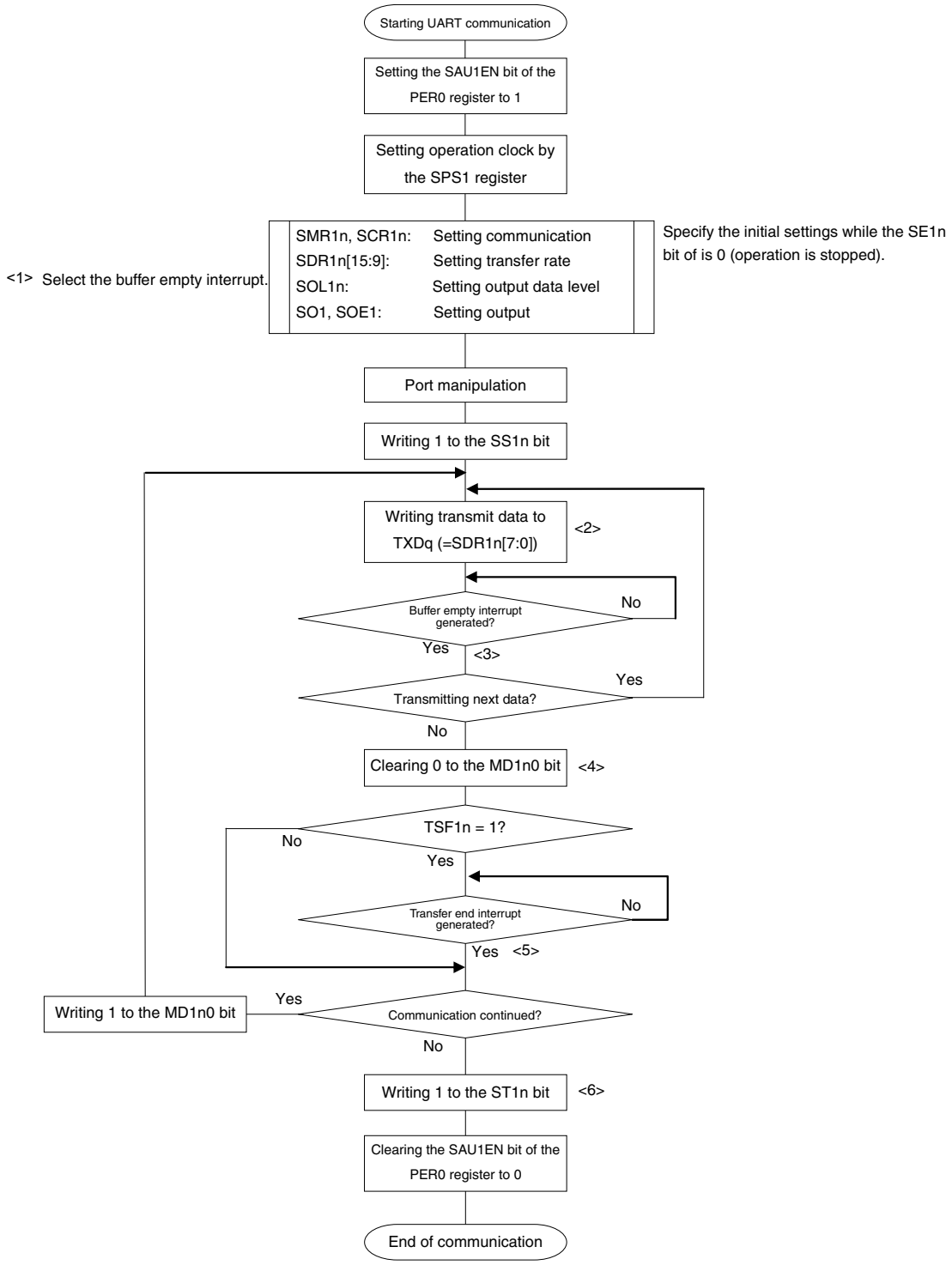


Note If transmit data is written to the SDR1n register while the BFF1n bit is 1 (valid data is stored in SDR1n), the transmit data is overwritten.

Caution The MD1n0 bit can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remark n: Channel number (n = 0, 2), q: UART number (q = 2, 3)

Figure 11-78. Flowchart of UART Transmission (in Continuous Transmission Mode)



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-77 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

11.6.2 UART reception

UART reception is an operation wherein the μPD78F8040, 78F8041, 78F8042, 78F8043 asynchronously receive 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 even-numbered channels must be set.

UART	UART2	UART3
Target channel	Channel 1 of SAU1	Channel 3 of SAU1
Pins used	RxD2	RxD3
Interrupt	INTSR2	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE2	INTSRE3
Error detection flag	• Framing error detection flag (FEF1n) • Parity error detection flag (PEF1n) • Overrun error detection flag (OVF1n)	
Transfer data length	5, 7 or 8 bits	
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR1n [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 (The parity bit is not checked.) • Appending 0 parity (The parity bit is not checked.) • Even-parity check • Odd-parity check	
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 26 ELECTRICAL SPECIFICATIONS**).

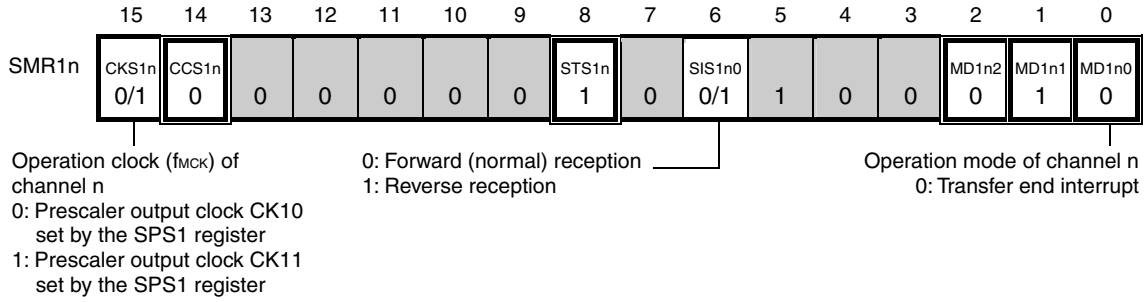
Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 1, 3)

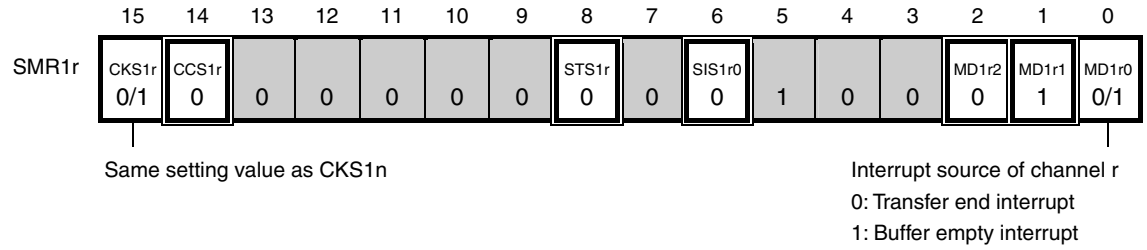
(1) Register setting

Figure 11-79. Example of Contents of Registers for UART Reception of UART (UART2, UART3) (1/2)

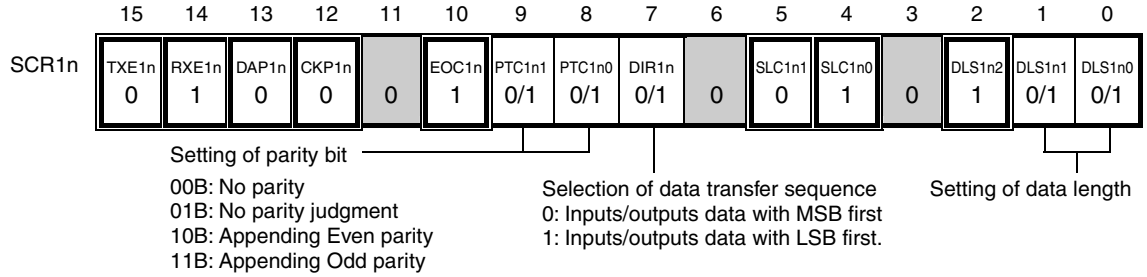
(a) Serial mode register 1n (SMR1n)



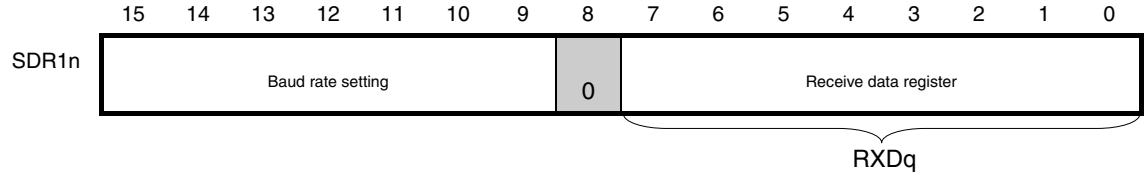
(b) Serial mode register 1r (SMR1r)



(c) Serial communication operation setting register 1n (SCR1n)



(d) Serial data register 1n (SDR1n) (lower 8 bits: RXDq)



Caution For the UART reception, be sure to set SMR1r of channel r that is to be paired with channel n.

Remark n: Channel number (n = 1, 3), r: Channel number (r = n – 1), q: UART number (q = 2, 3)

□: Setting is fixed in the UART reception mode, ▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-79. Example of Contents of Registers for UART Reception of UART (UART2, UART3) (2/2)

(e) Serial output register 1 (SO1) … The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10 ×	0	0	0	0	1	SO12 ×	1	SO10 ×

(f) Serial output enable register 1 (SOE1) … The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12 ×	0	SOE10 ×

(g) Serial channel start register 1 (SS1) … Sets only the bits of the target channel is 1.

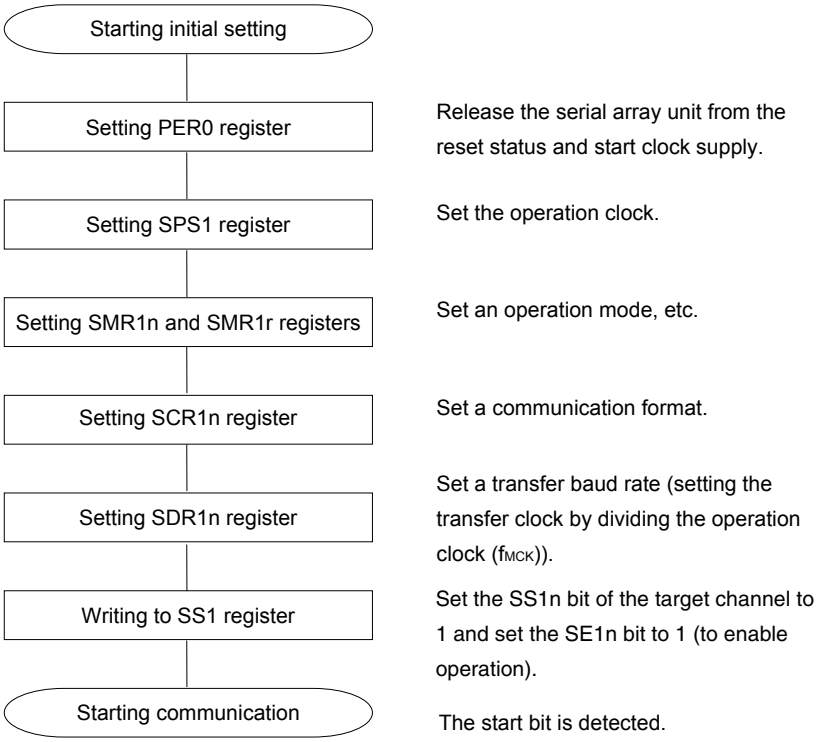
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 0/1	SS12 ×	SS11 0/1	SS10 ×

Caution For the UART reception, be sure to set SMR1r of channel r that is to be paired with channel n.

Remark n: Channel number (n = 1, 3), r: Channel number (r = n – 1), q: UART number (q = 2, 3)
□: 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 11-80. Initial Setting Procedure for UART Reception



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

Figure 11-81. Procedure for Stopping UART Reception

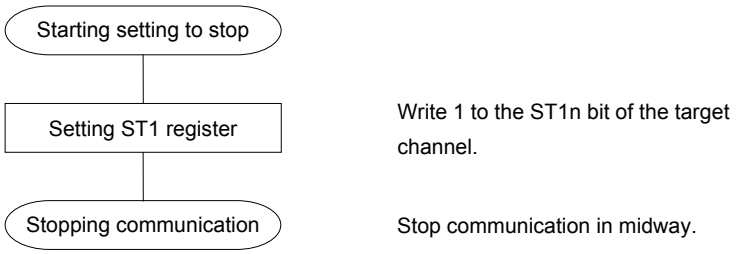
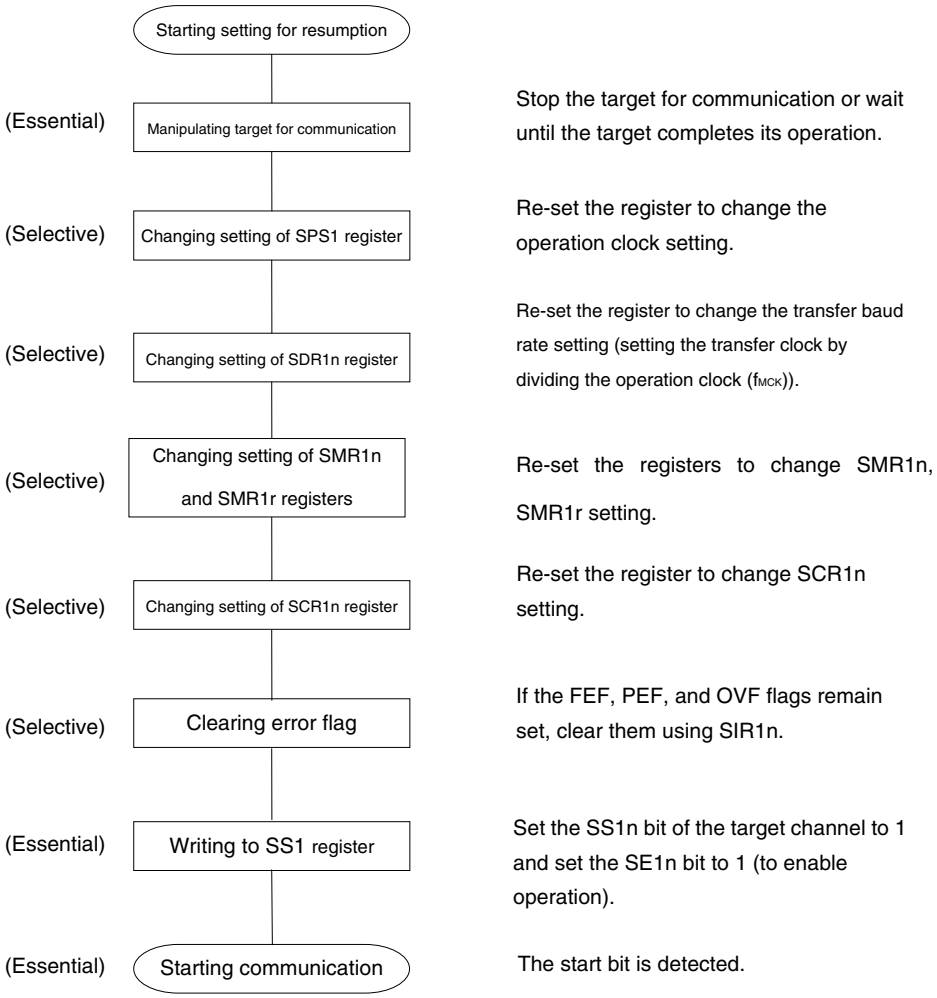
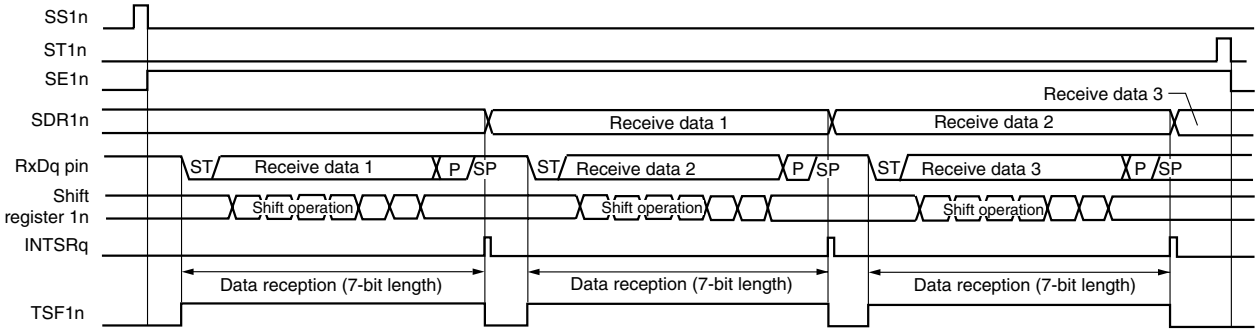


Figure 11-82. Procedure for Resuming UART Reception



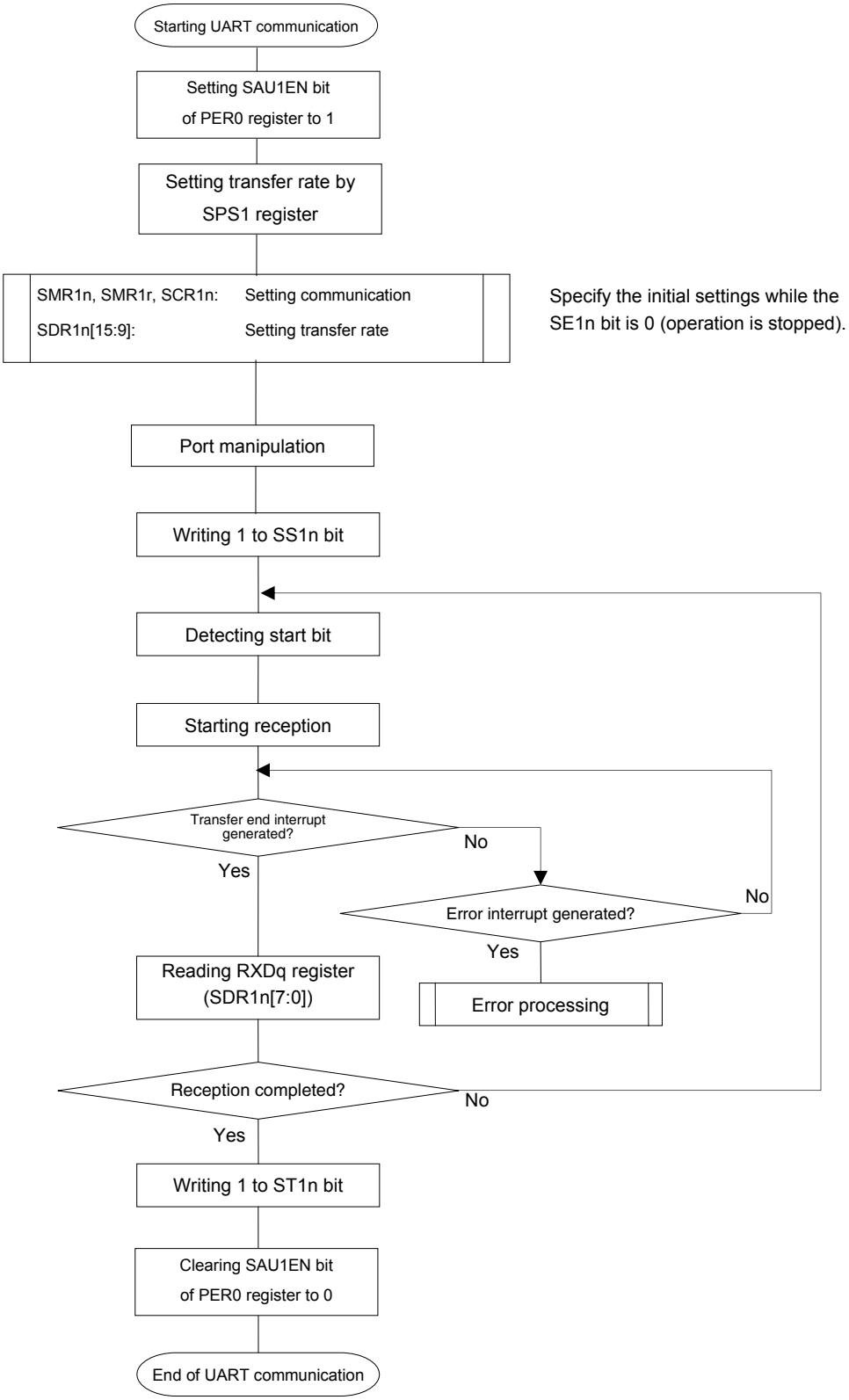
(3) Processing flow

Figure 11-83. Timing Chart of UART Reception



Remark n: Channel number (n = 1, 3), q: UART number (q = 2, 3)

Figure 11-84. Flowchart of UART Reception



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

11.6.3 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0, UART2, UART3) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting SDRmn [15:9] = (0000000B, 0000001B) is prohibited.

Remarks 1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

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

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-3. Selection of operation clock

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} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0, INTTM03 if m = 1	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0, INTTM03 if m = 1	
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 (STm = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

Remarks 1. X: Don't care

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

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART2, UART3) 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} = 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 20 MHz			
	Operation Clock (MCK)	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	39	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	64	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	64	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	64	153846 bps	+0.16 %
312500 bps	f_{CLK}	31	312500 bps	±0.0 %

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0, UART2, UART3) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

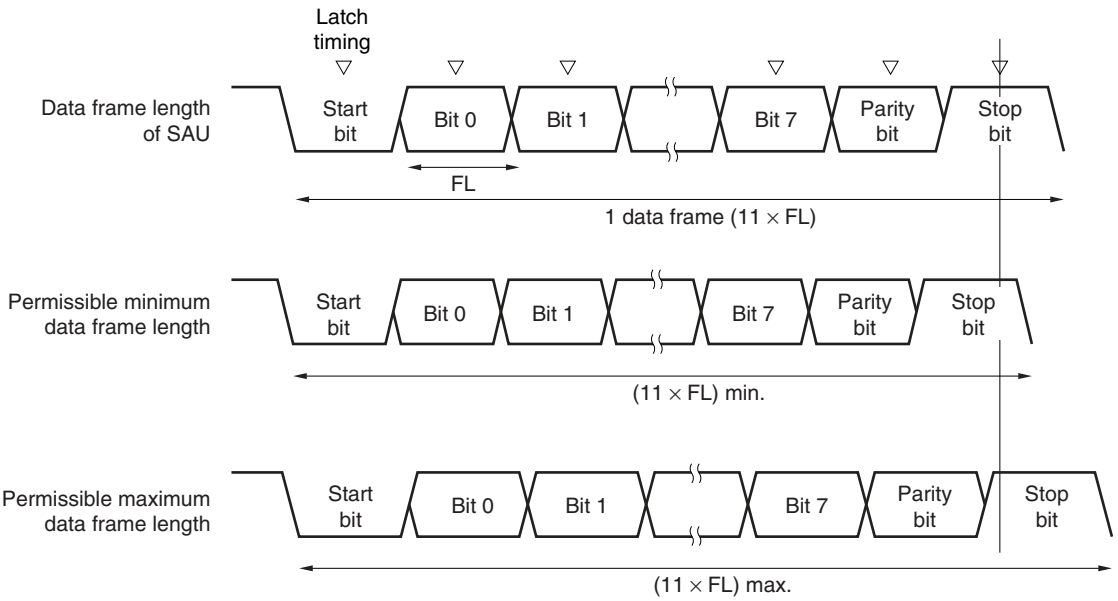
Brate: Calculated baud rate value at the reception side (See **11.6.3 (1) Baud rate calculation expression**.)

k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

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

Figure 11-85. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 11-85, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of the serial data register mn (SDRmn) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

11.6.4 Procedure for processing errors that occurred during UART (UART0, UART2, UART3) communication

The procedure for processing errors that occurred during UART (UART0, UART2, UART3) communication is described in Figures 11-86 and 11-87.

Figure 11-86. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. →	The BFFmn bit 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes 1 to SIRmn register. →	Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 01, 10 to 13

Figure 11-87. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. →	The BFFmn bit 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. →	Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets STmn bit to 1. →	The SEmn bit 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.
Writes 1 to SSmn bit. →	The SEmn bit 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) , mn = 00, 01, 10 to 13

11.7 LIN Communication Operation

11.7.1 LIN transmission

Of UART transmission, UART3 supports LIN communication.

For LIN transmission, channel 2 of unit 1 (SAU1) is used.

UART	UART2	UART3
Support of LIN communication	Not supported	Supported
Target channel	–	Channel 2 of SAU1
Pins used	–	TxD3
Interrupt	–	INTST3
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	8 bits	
Transfer rate	Max. f_{MCK}/6 [bps] (SDR12 [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 • 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 26 ELECTRICAL SPECIFICATIONS**).

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.

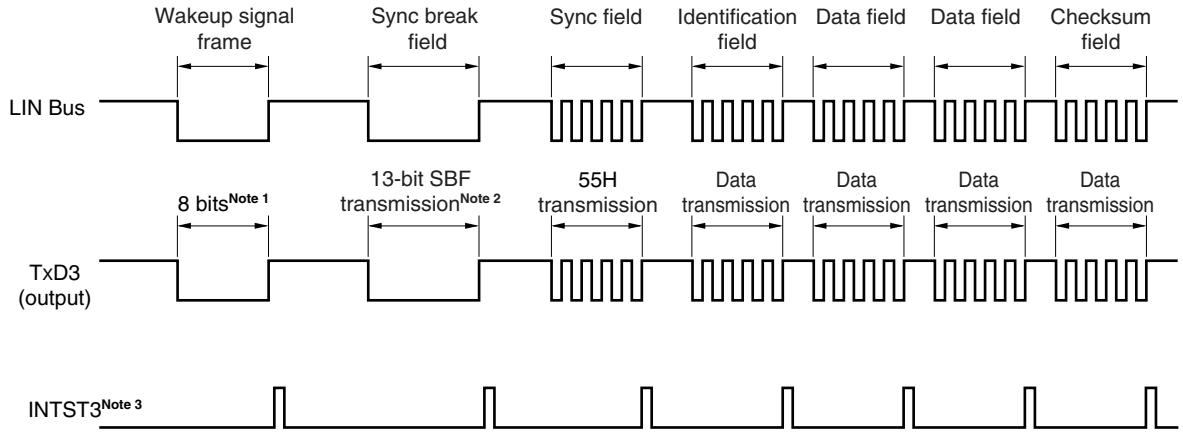
Usually, the master is connected to a network such as CAN (Controller Area Network).

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 11-88 outlines a transmission operation of LIN.

Figure 11-88. 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 sync 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 sync break field is calculated as follows.

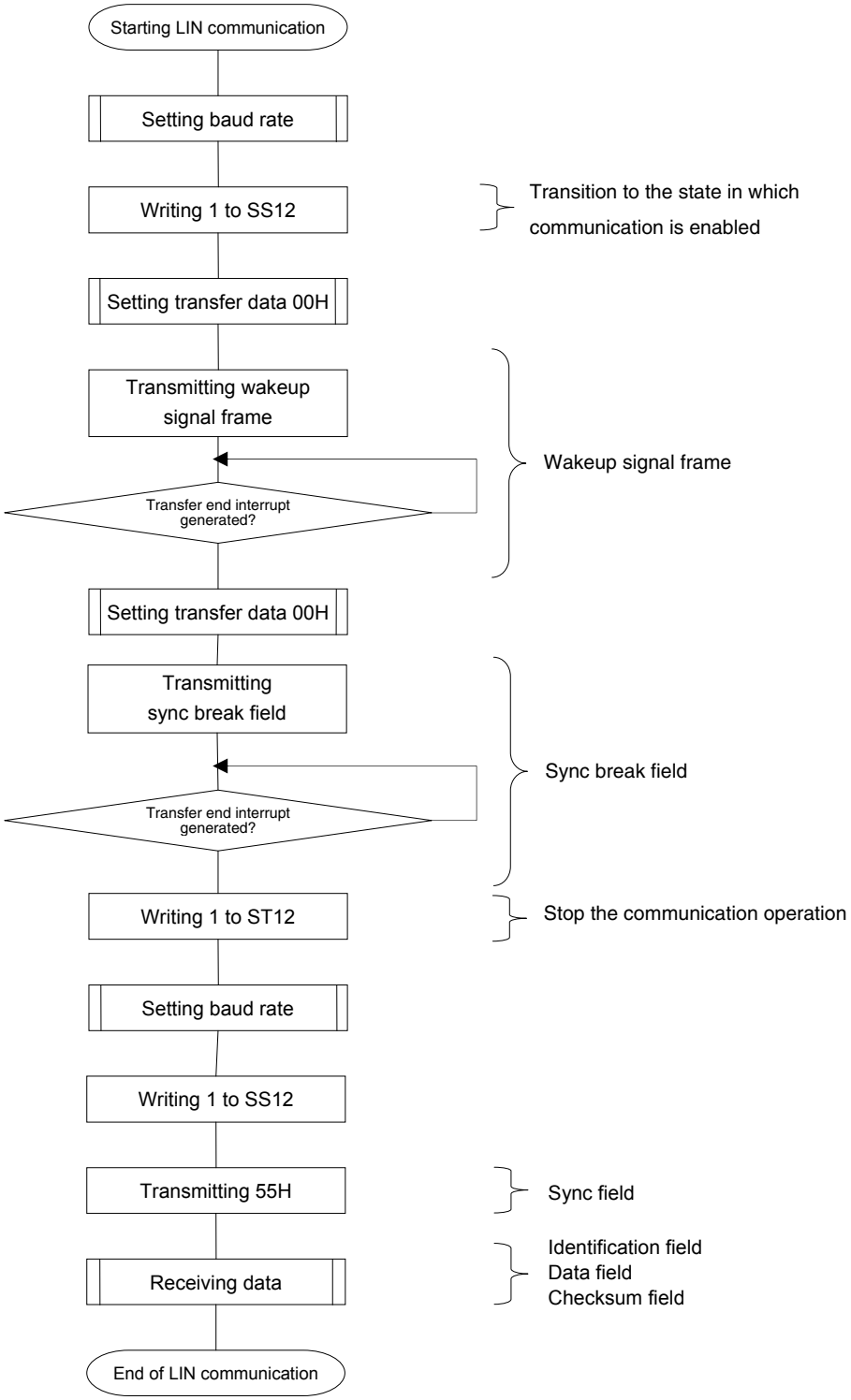
(Baud rate of sync break field) = 9/13 × N

By transmitting data of 00H at this baud rate, a sync break field is generated.

3. INTST3 is output upon completion of transmission. INTST3 is also output when SBF transmission is executed.

Remark The interval between fields is controlled by software.

Figure 11-89. Flowchart for LIN Transmission



11.7.2 LIN reception

Of UART reception, UART3 supports LIN communication.

For LIN reception, channel 3 of unit 1 (SAU1) is used.

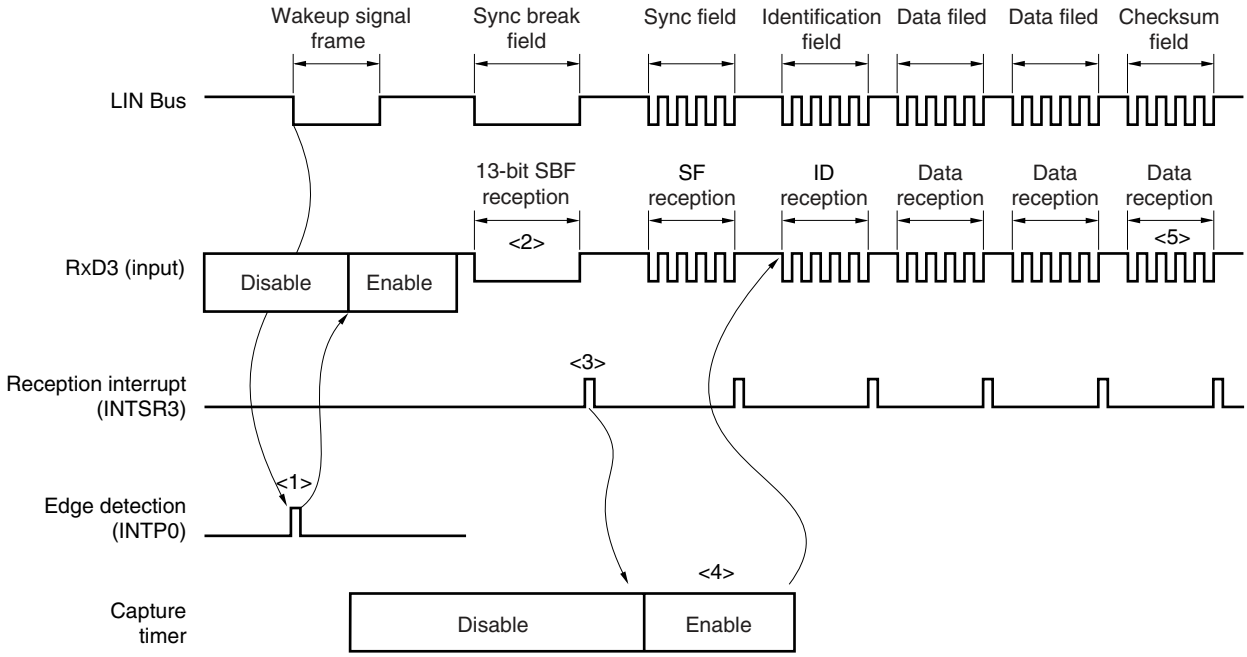
UART	UART2	UART3
Support of LIN communication	Not supported	Supported
Target channel	–	Channel 3 of SAU1
Pins used	–	RxD3
Interrupt	–	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	–	INTSRE3
Error detection flag	• Framing error detection flag (FEF13) • Parity error detection flag (PEF13) • Overrun error detection flag (OVF13)	
Transfer data length	8 bits	
Transfer rate	Max. f_{MCK}/6 [bps] (SDR13 [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 (The parity bit is not checked.) • Appending 0 parity (The parity bit is not checked.) • Even-parity check • Odd-parity check	
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 26 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Figure 11-90 outlines a reception operation of LIN.

Figure 11-90. 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, enable reception of UART3 (RXE13 = 1) and wait for SBF reception.

<2> When the start bit of SBF is detected, reception is started and serial data is sequentially stored in the RXD3 register (= bits 7 to 0 of the serial data register 13 (SDR13)) at the set baud rate. When the stop bit is detected, the reception end interrupt request (INTSR3) is generated. When data of low levels of 11 bits or more is detected as SBF, it is judged that SBF reception has been correctly completed. If data of low levels of less than 11 bits is detected as SBF, it is judged that an SBF reception error has occurred, and the system returns to the SBF reception wait status.

<3> When SBF reception has been correctly completed, start channel 7 of the timer array unit TAU0 and measure the bit interval (pulse width) of the sync field (see **8.7.4 Operation as input signal high-/low-level width measurement**).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART3 once and adjust (re-set) the baud rate.

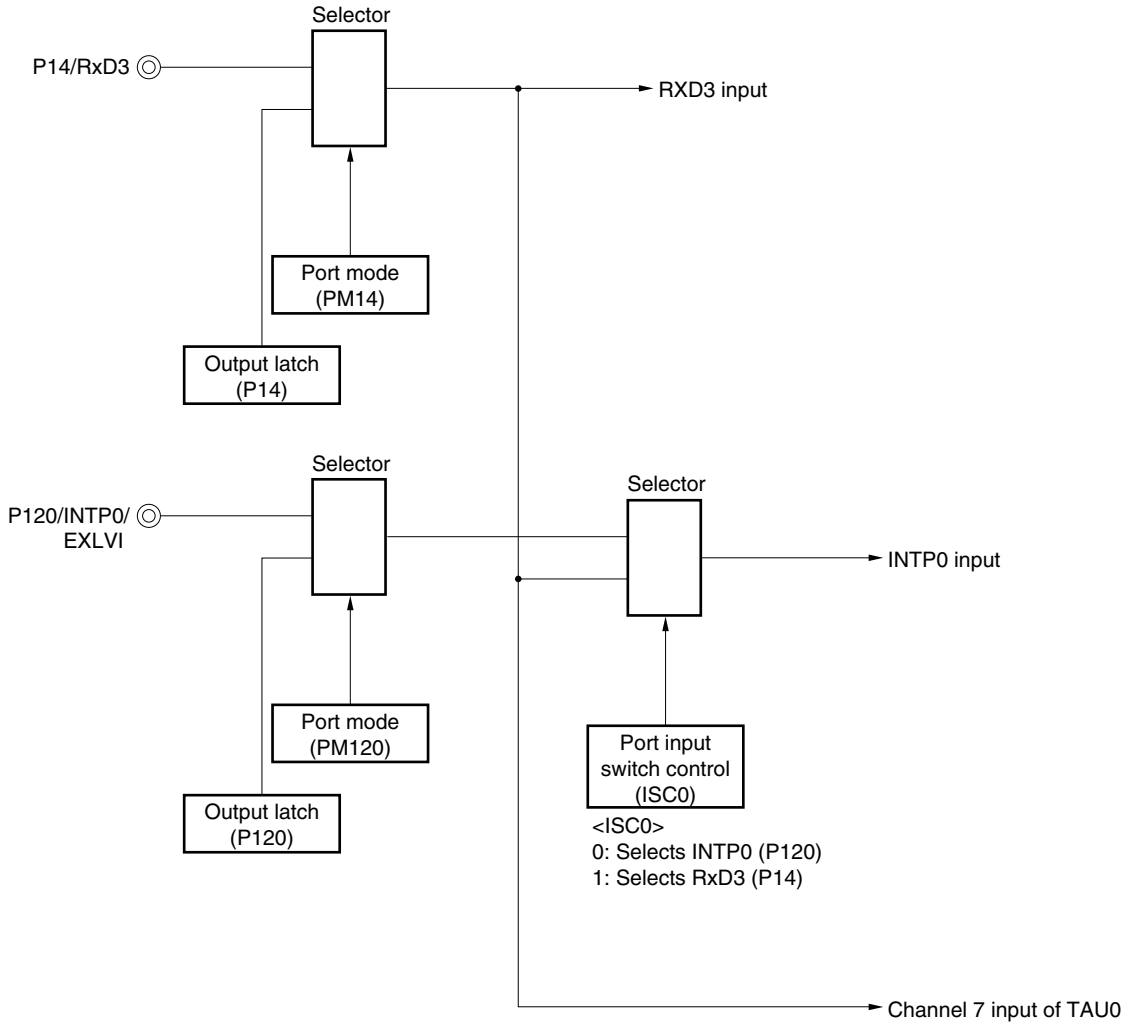
<5> The checksum field should be distinguished by software. In addition, processing to initialize UART3 after the checksum field is received and to wait for reception of SBF should also be performed by software.

Figure 11-91 shows 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 (TAU0) to calculate a baud-rate error.

By controlling switch of port input (ISC0), the input source of port input (RxD3) for reception can be input to the external interrupt pin (INTP0) and timer array unit (TAU0).

Figure 11-91. Port Configuration for Manipulating Reception of LIN



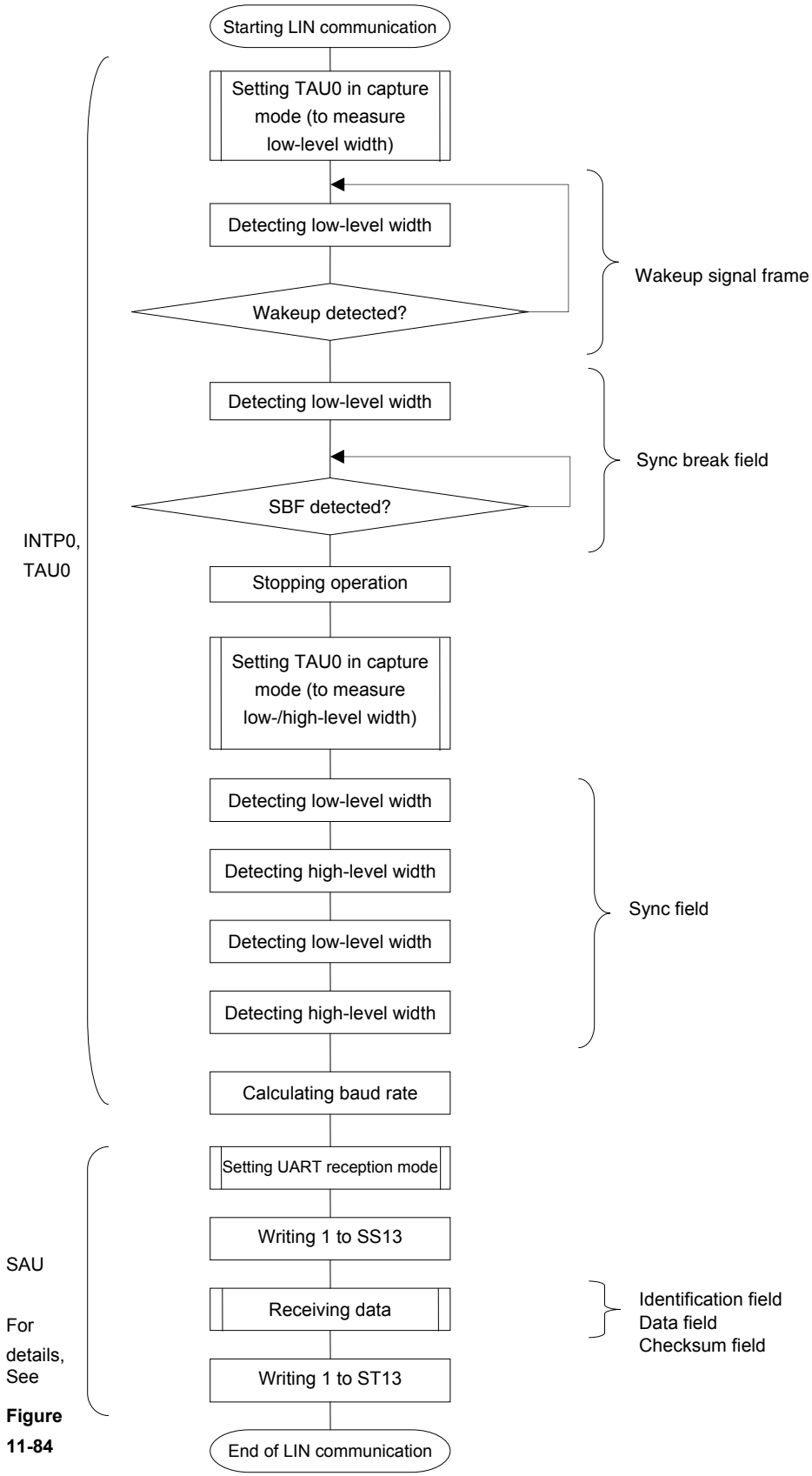
Remark ISC0: Bit 0 of the input switch control register (ISC) (See **Figure 11-17**.)

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 0 ; Baud rate error detection
  Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect an error (The interval of the edge input to RxD3 is measured in the capture mode.)
- Channels 2 and 3 (UART3) of serial array unit (SAU1)

Figure 11-92. Flowchart of LIN Reception



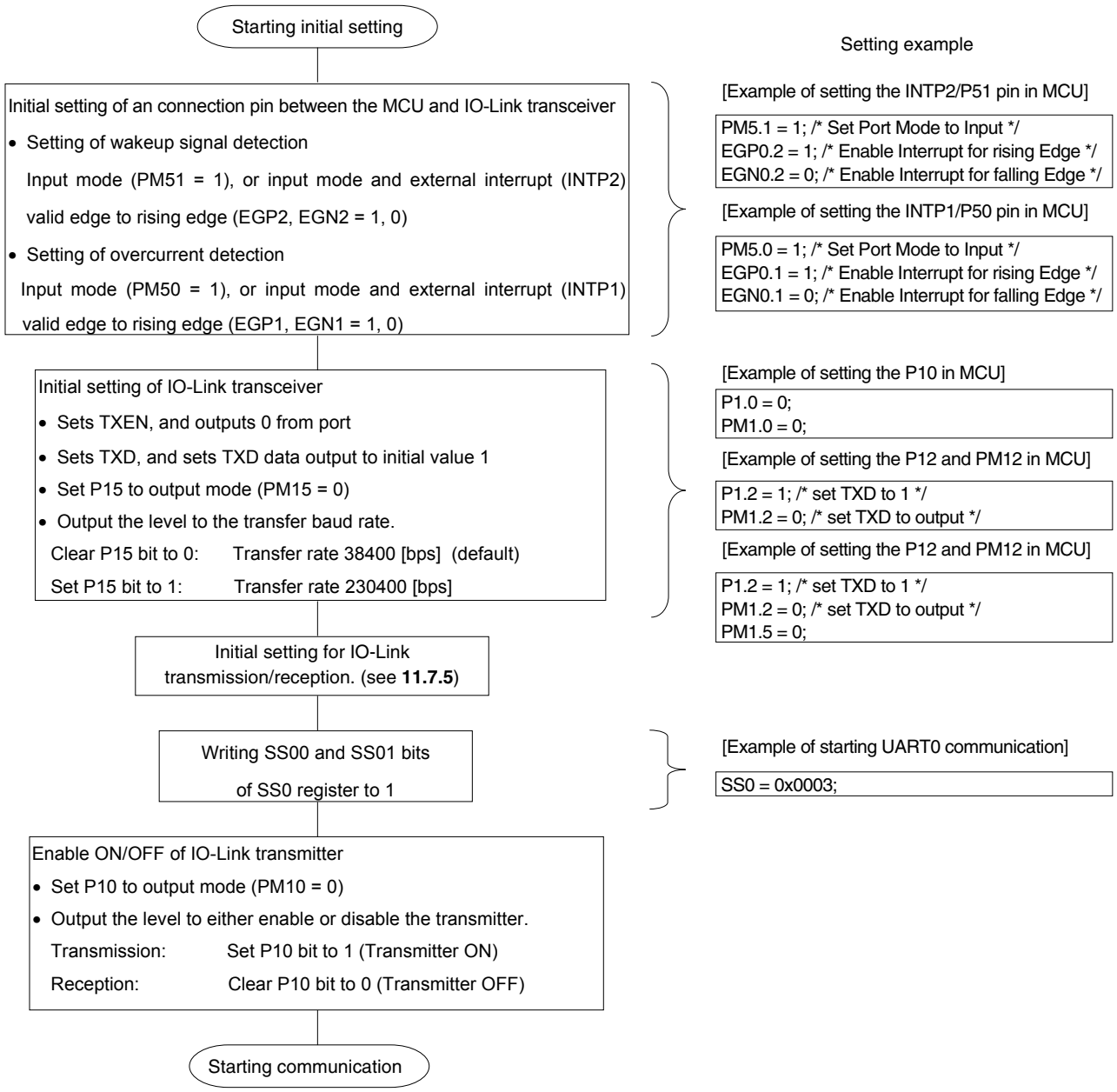
11.7.3 Communicating with IO-Link Transceiver

Communication with the IO-Link transceiver is performed by using UART0 of the MCU.

The MCU ports must be set to appropriate modes or levels before communicating with the IO-Link transceiver, because they are set to input mode after a reset release (See **CHAPTER 2 CONNECTION BETWEEN MCU AND IO-LINK TRANSCEIVER**).

11.7.4 Setting procedure for communicating with IO-Link transceiver

Figure 11-93. Setting Procedure for Communicating with IO-Link Transceiver



11.7.5 IO-Link transmission/reception

IO-Link transmission/reception is an operation to transmit/receive data between the MCU and IO-Link transceiver asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for IO-Link transmission. For IO-Link reception, the odd channel of the two channels used for IO-Link is used. In the case of IO-Link reception, however, the SMR register of both the odd- and even-numbered channels must be set.

UART	UART0	
	Transmission	Reception
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	TxD0 (internal pin)	RxD0
Interrupt	INTST0	INTSR0
		Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error interrupt	None	INTSRE0
Error detection flag	None	• Framing error detection flag (FEF11) • Parity error detection flag (PEF11) • Overrun error detection flag (OVF11)
Transfer data length	8 bits	
Transfer rate	38400 [bps] or 230400 [bps] Notes 1, 2	
Data phase	Reverse output	None
Parity bit	Appending odd parity	Odd-parity check
Stop bit	Appending 1 bit	Appending 1 bit
Data direction	LSB first	

Notes 1. Example 1:

Target baud rate: 38400 [bps]
Operation clock: $f_{CLK}/2^2$ (at f_{CLK} = 20 MHz) (SDR00[15:9] = 64)
Caluculated baud rate: 38461.5 [bps]
Error from target baud rate: +0.16 [%]

[Setting example]

```
SPS0 = 0x0002; /* set CK0 Clock to 5 MHz */
SDR00 = 0x8000; /* set clock for channel 0 (38,4 Kbit/s) */
SDR01 = 0x8000; /* set clock for channel 1 (38,4 Kbit/s) */
```

Example 2:

Target baud rate: 230400 [bps]
Operation clock: f_{CLK} (at f_{CLK} = 20 MHz) (SDR00[15:9] = 42)
Caluculated baud rate: 232558 [bps]
Error from target baud rate: +0.93 [%]

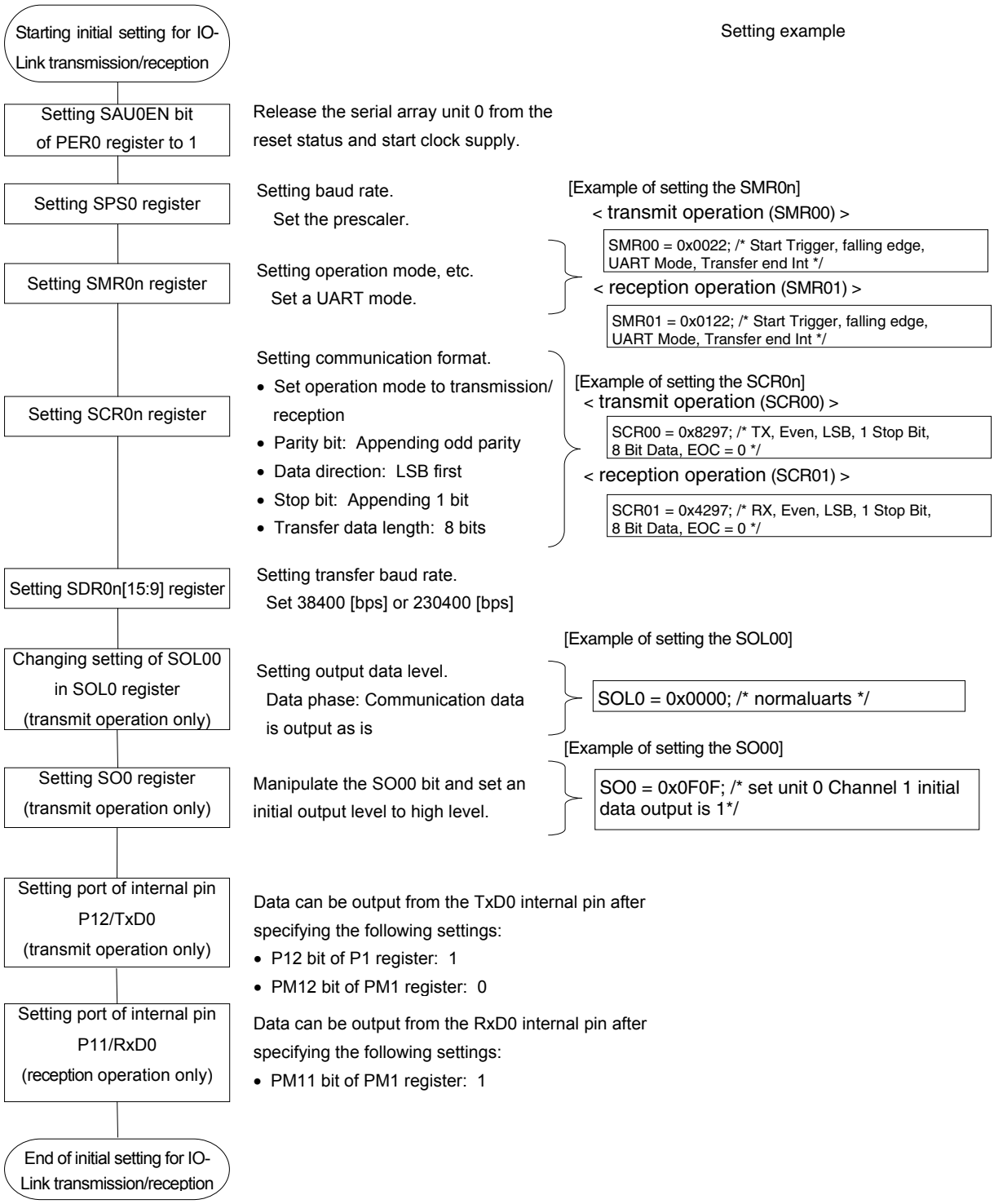
[Setting example]

```
SPS0 = 0x0000; /* set CK0 Clock to 20 MHz */
SDR00 = 0x5400; /* set clock for channel 0 (230,4 Kbit/s) */
SDR01 = 0x5400; /* set clock for channel 1 (230,4 Kbit/s) */
```

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 26 ELECTRICAL SPECIFICATIONS**).

Remark f_{CLK}: System clock frequency

Figure 11-94. Initial Setting Procedure for UART Transmission



Note n = 0, 1

Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS0 register after 4 or more f_{CLK} clocks have elapsed.

11.8 Operation of Simplified I²C (IIC20) 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 and does not have a wait detection function. Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note] and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- 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 function

Note An ACK is not output when the last data is being received by writing 0 to the SOE10 (SOE1 register) bit and stopping the output of serial communication data. See **11.8.3 (2) Processing flow** for details.

Remark To use the full-function I²C bus, see **CHAPTER 12 SERIAL INTERFACE IICA**.

The channels supporting simplified I²C (IIC20) is channel 0 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C
0	0	–	UART0 (dedicated to IO-Link communication)	–
	1	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

Simplified I²C (IIC20) performs the following four types of communication operations.

- Address field transmission (See **11.8.1**.)
- Data transmission (See **11.8.2**.)
- Data reception (See **11.8.3**.)
- Stop condition generation (See **11.8.4**.)

11.8.1 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	IIC20
Target channel	Channel 0 of SAU1
Pins used	SCL20, SDA20 Note
Interrupt	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Parity error detection flag (PEF10)
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}/4 [Hz] (SDR10 [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 (first mode) • Max. 100 kHz (standard mode)
Data level	Forward 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 (POM143 = 1) for the port output mode register (POM14) (see **6.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POM142 = 1) also for the clock input/output pins (SCL20) (see **6.4.4 Connecting to external device with different potential (2.5 V, 3 V)** for details).

(1) Register setting

Figure 11-95. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC20)

(a) Serial mode register 10 (SMR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR10	CKS10	CCS10	0	0	0	0	0	STS10	0	SIS100	1	0	0	MD102	MD101	MD100
	0/1	0						0		0				1	0	0

Operation clock (f_{MCK}) of channel 0
0: Prescaler output clock CK10 set by the SPS1 register
1: Prescaler output clock CK11 set by the SPS1 register

Operation mode of channel 0
0: Transfer end interrupt

(b) Serial communication operation setting register 10 (SCR10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10	RXE10	DAP10	CKP10		EOC10	PTC101	PTC100	DIR10		SLC101	SLC100		DLS102	DLS101	DLS100
	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

(c) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Transmit data setting (address + R/W)							

SIO20

(d) Serial output register 1 (SO1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	SO12		SO10
								0/1						×	1	0/1

Start condition is generated by manipulating the SO10 bit.

(e) Serial output enable register 1 (SOE1) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12		SOE10
														×	0	0/1

SOE10 = 0 until the start condition is generated, and SOE10 = 1 after generation.

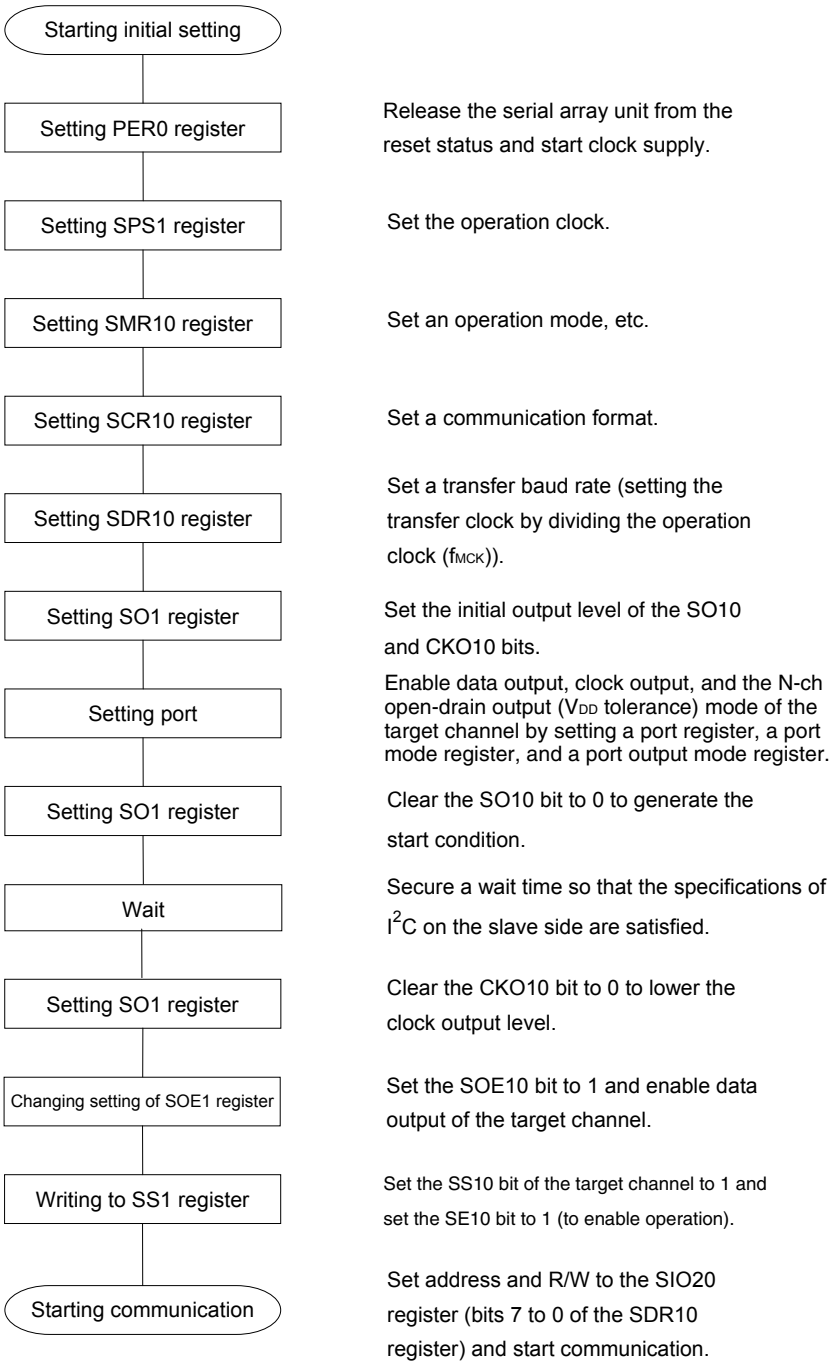
(f) Serial channel start register 1 (SS1) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13	SS12	SS11	SS10
													×	×	×	0/1

Remark □: Setting is fixed in the IIC mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-96. Initial Setting Procedure for Address Field Transmission



Caution After setting the SAU1EN bit of PER0 to 1, be sure to set SPS1 after 4 or more f_{CLK} clocks have elapsed.

(3) Processing flow

Figure 11-97. Timing Chart of Address Field Transmission

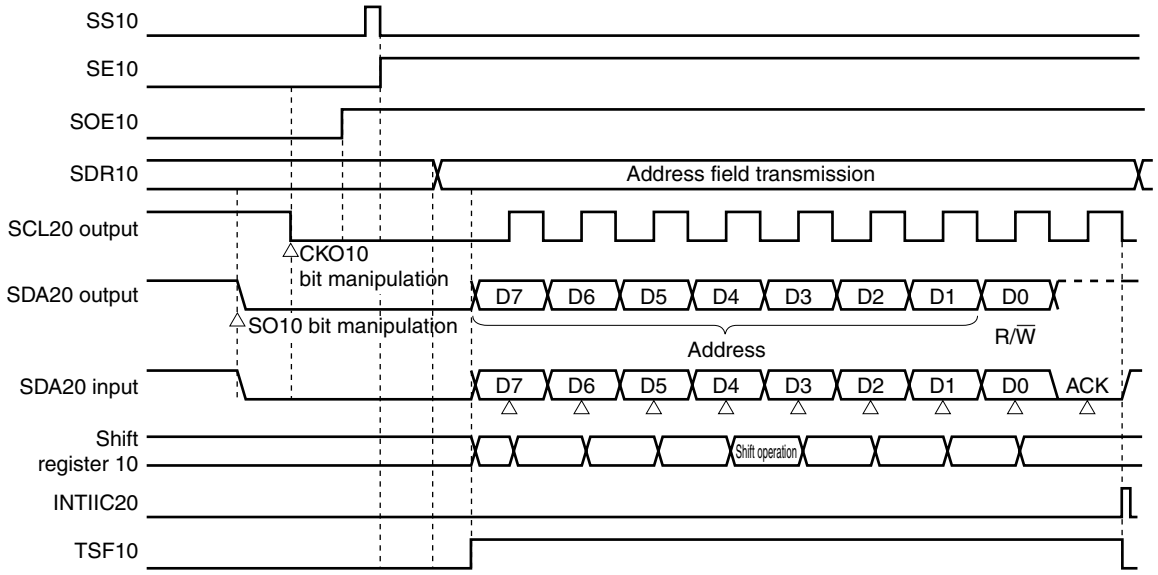
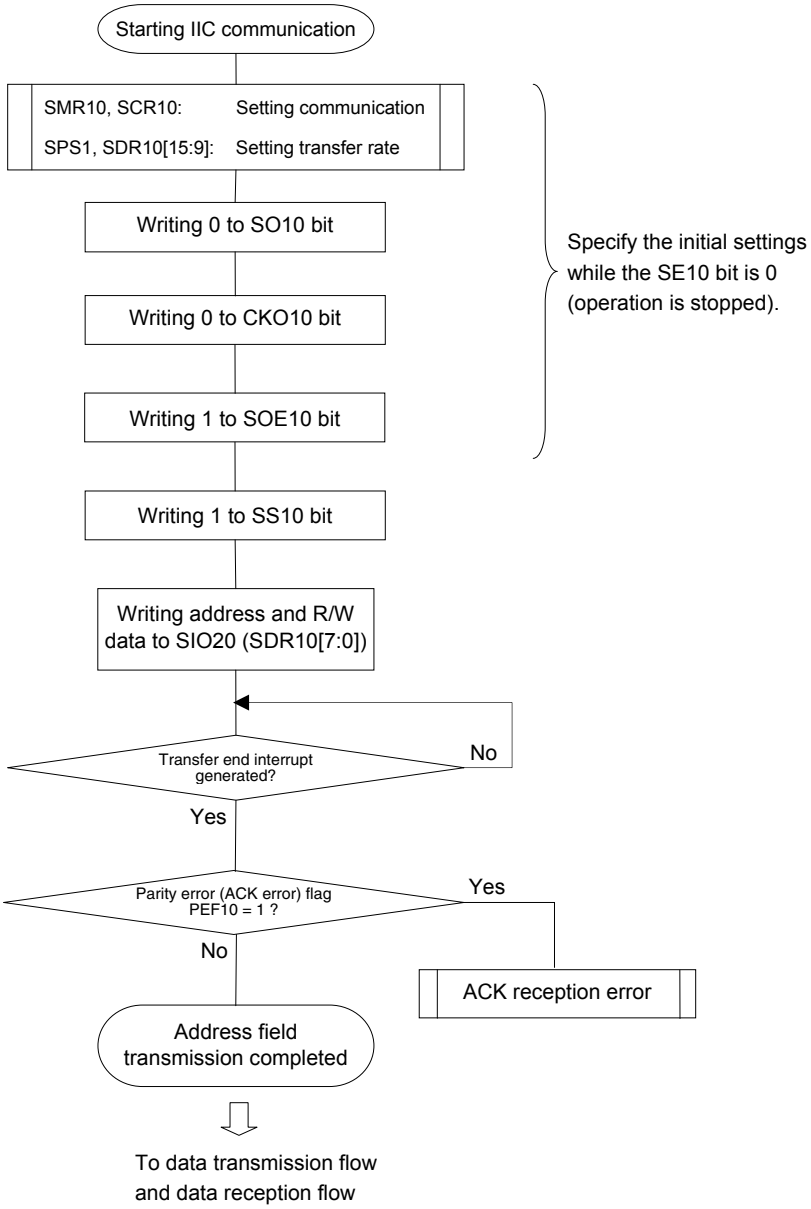


Figure 11-98. Flowchart of Address Field Transmission



11.8.2 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	IIC20
Target channel	Channel 0 of SAU1
Pins used	SCL20, SDA20 Note
Interrupt	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Parity error detection flag (PEF10)
Transfer data length	8 bits
Transfer rate	Max. f_{MCK}/4 [Hz] (SDR10 [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 (first mode) • Max. 100 kHz (standard mode)
Data level	Forward 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 (POM143 = 1) for the port output mode register (POM14) (see **6.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POM142 = 1) also for the clock input/output pins (SCL20) (see **6.4.4 Connecting to external device with different potential (2.5 V, 3 V)** for details).

(1) Register setting

Figure 11-99. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC20)

(a) Serial mode register 10 (SMR10) … 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
SMR10	CKS10	CCS10	0	0	0	0	0	STS10	0	SIS100	1	0	0	MD102	MD101	MD100
	0/1	0						0		0				1	0	0

(b) Serial communication operation setting register 10 (SCR10) … Do not manipulate the bits of this register, except the TXE10 and RXE10 bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10	RXE10	DAP10	CKP10	0	EOC10	PTC101	PTC100	DIR10	0	SLC101	SLC100	0	DLS102	DLS101	DLS100
	1	0	0	0		0	0	0	0		0	1		1	1	1

(c) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Transmit data setting							

SIO20

(d) Serial output register 1 (SO1) … 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
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	SO12	1	SO10
								0/1 Note						×		0/1 Note

(e) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12	0	SOE10
														×		0/1

(f) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13	SS12	SS11	SS10
													×	×	×	0/1

Note The value varies depending on the communication data during communication operation.

Remark □: Setting is fixed in the IIC mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Processing flow

Figure 11-100. Timing Chart of Data Transmission

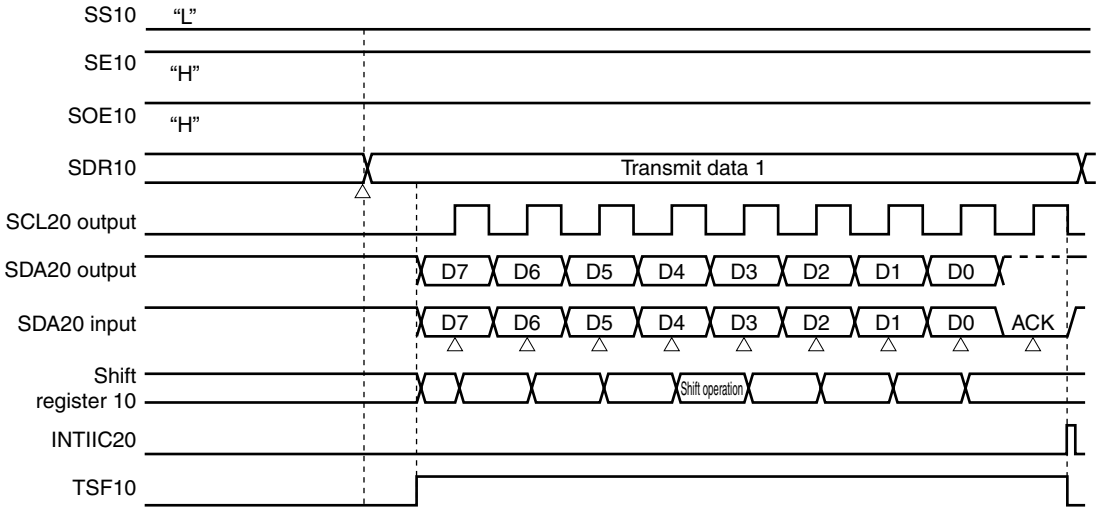
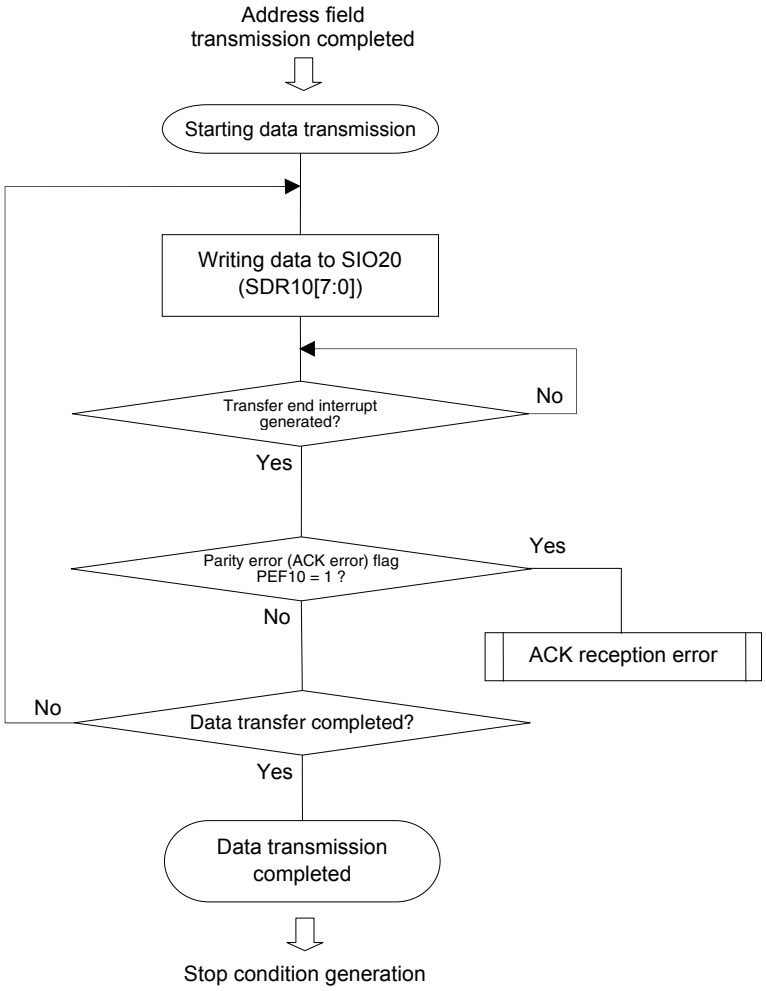


Figure 11-101. Flowchart of Data Transmission



11.8.3 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	IIC20
Target channel	Channel 0 of SAU1
Pins used	SCL20, SDA20 Note
Interrupt	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Overrun error detection flag (OVF10) only
Transfer data length	8 bits
Transfer rate	Max. f_{MCK}/4 [Hz] (SDR10 [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 (first mode) • Max. 100 kHz (standard mode)
Data level	Forward 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 (POM143 = 1) for the port output mode registers (POM14) (see **6.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POM142 = 1) also for the clock input/output pins (SCL20) (see **6.4.4 Connecting to external device with different potential (2.5 V, 3 V)** for details).

(1) Register setting

Figure 11-102. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC20)

(a) Serial mode register 10 (SMR10) … 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
SMR10	CKS10	CCS10	0	0	0	0	0	STS10	0	SIS100	1	0	0	MD102	MD101	MD100
	0/1	0						0		0				1	0	0

(b) Serial communication operation setting register 10 (SCR10) … Do not manipulate the bits of this register, except the TXE10 and RXE10 bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR10	TXE10	RXE10	DAP10	CKP10	0	EOC10	PTC101	PTC100	DIR10	0	SLC101	SLC100	0	DLS102	DLS101	DLS100
	0	1	0	0		0	0	0	0		0	1		1	1	1

(c) Serial data register 10 (SDR10) (lower 8 bits: SIO20)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR10	Baud rate setting							0	Dummy transmit data setting (FFH)							

(d) Serial output register 1 (SO1) … 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
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	SO12	1	SO10
								0/1Note						×		0/1Note

(e) Serial output enable register 1 (SOE1) … 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
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12	0	SOE10
														×		0/1

(f) Serial channel start register 1 (SS1) … 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
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13	SS12	SS11	SS10
													×	×	×	0/1
									SIO20							

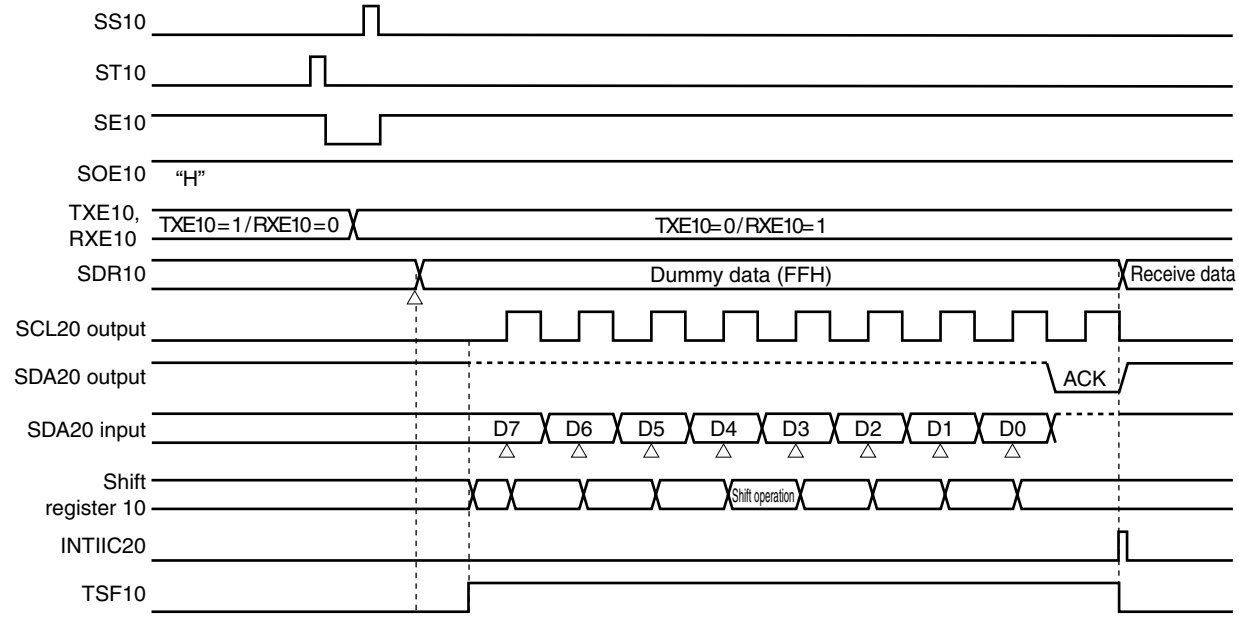
Note The value varies depending on the communication data during communication operation.

Remark ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Processing flow

Figure 11-103. Timing Chart of Data Reception

(a) When starting data reception



(b) When receiving last data

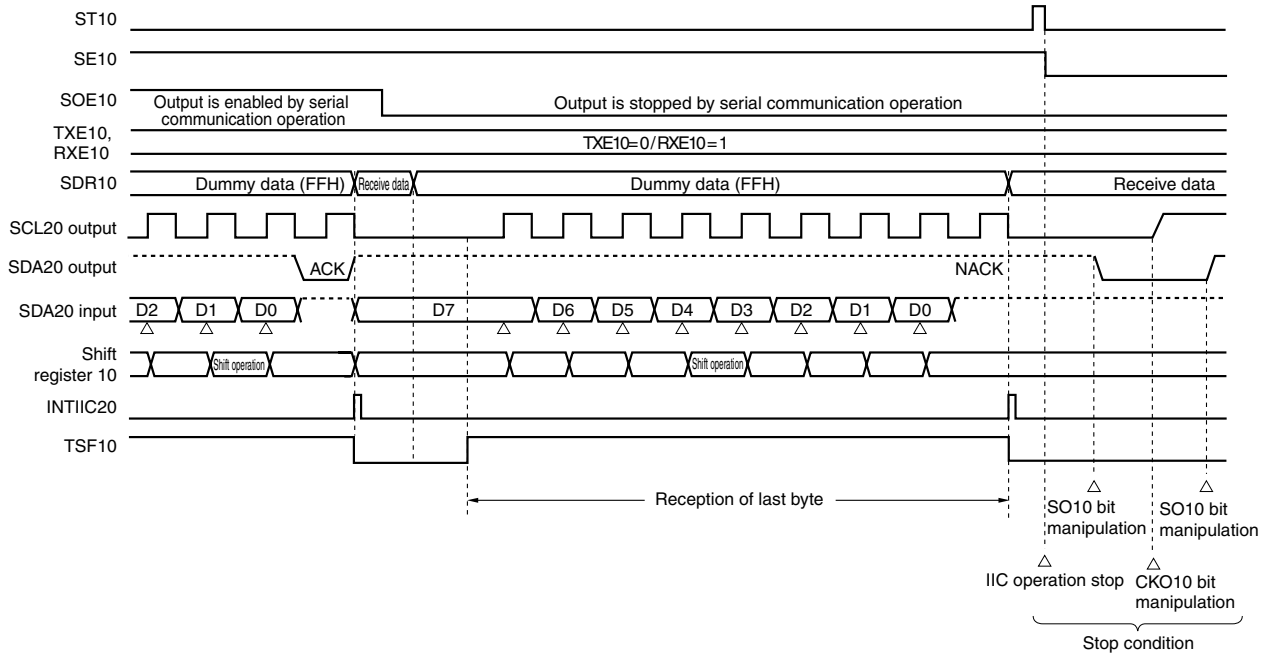
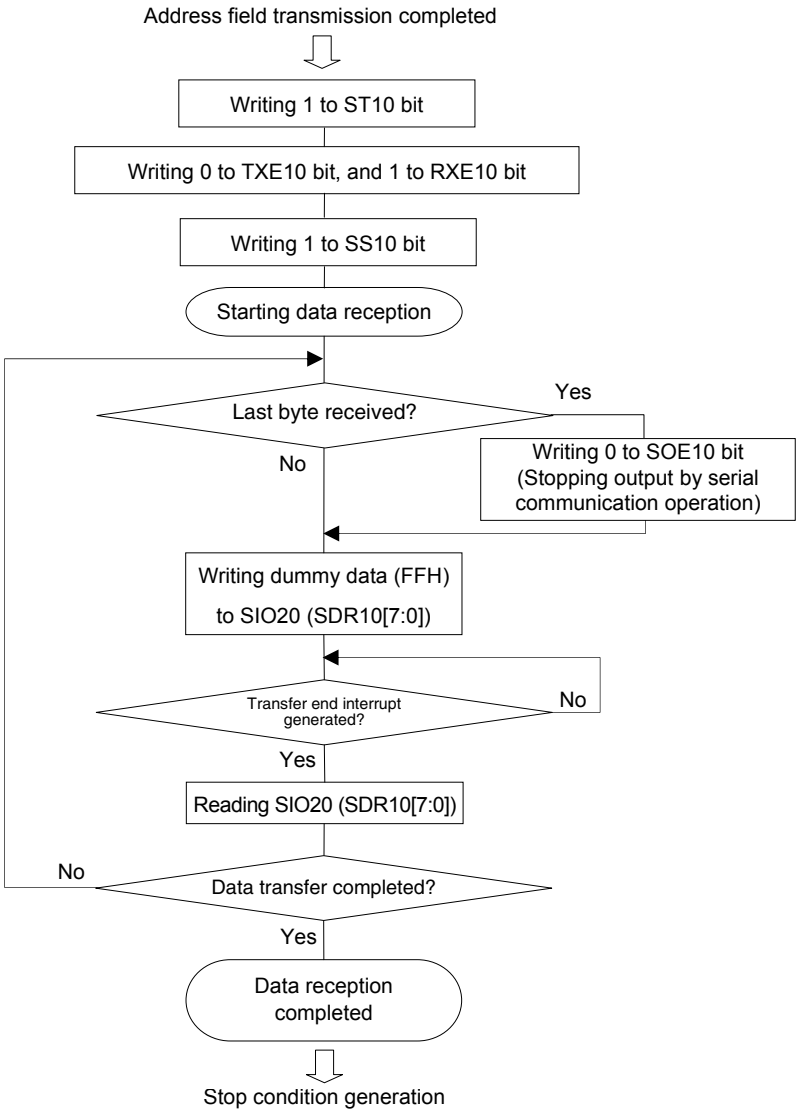


Figure 11-104. Flowchart of Data Reception



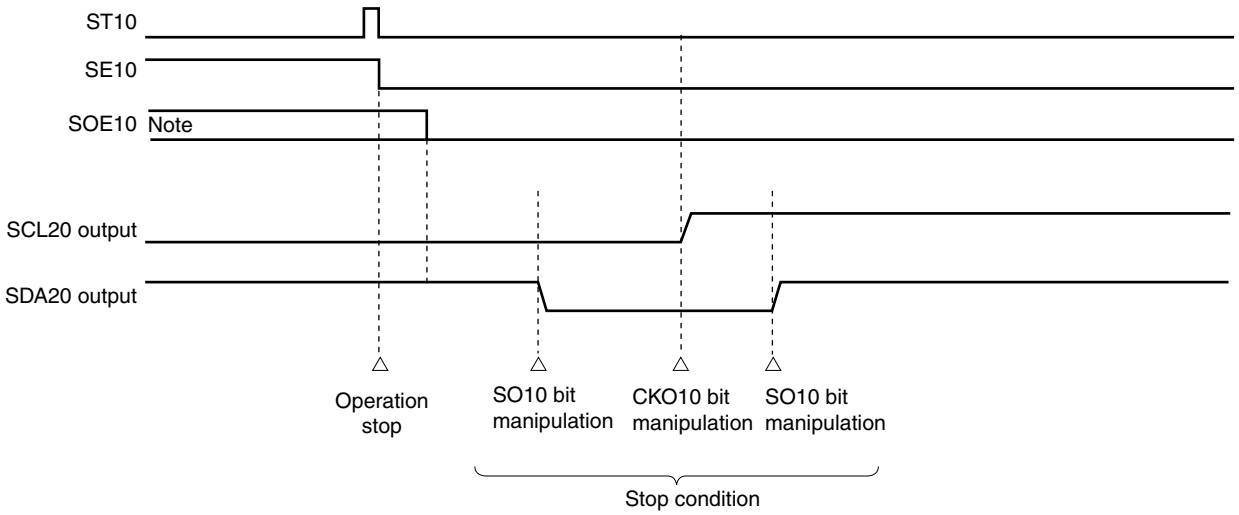
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the ST10 bit to stop operation and generating a stop condition.

11.8.4 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.

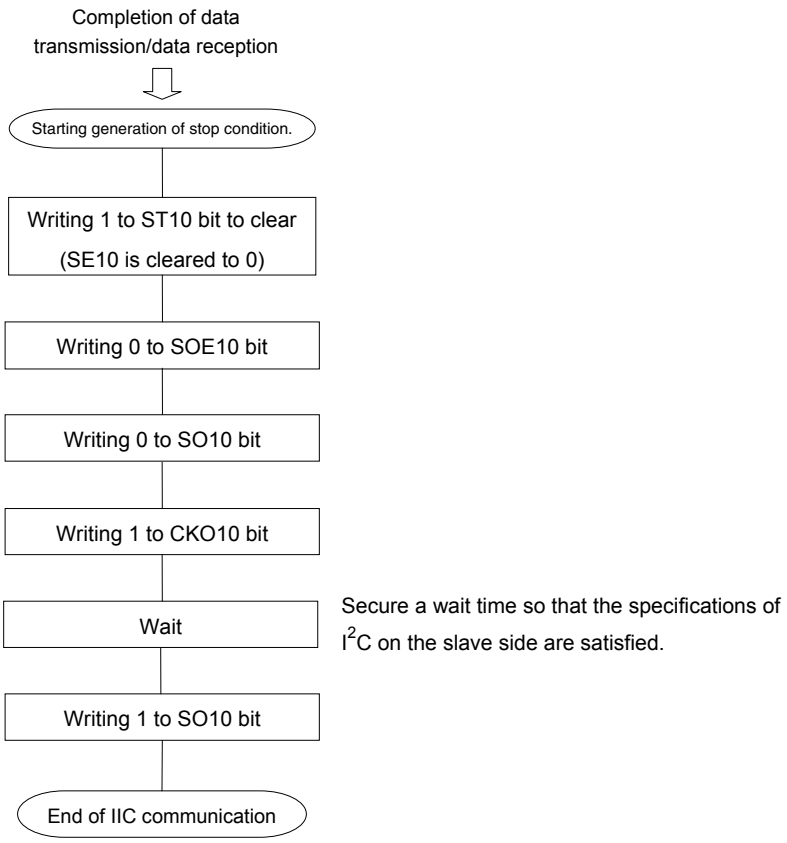
(1) Processing flow

Figure 11-105. Timing Chart of Stop Condition Generation



Note During the receive operation, the SOE10 bit is set to 0 before receiving the last data.

Figure 11-106. Flowchart of Stop Condition Generation



11.8.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC20) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (MCK) frequency of target channel} ÷ (SDR10[15:9] + 1) ÷ 2

Caution Setting SDR10[15:9] = 0000000B is prohibited. Setting SDR10[15:9] = 0000001B or more.

Remark The value of SDR10[15:9] is the value of bits 15 to 9 of the SDR10 register (0000001B to 1111111B) and therefore is 1 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register 1 (SPS1) and bit 15 (CKS10) of serial mode register 10 (SMR10).

Table 11-4. Selection of operation clock

SMR10 Register	SPS1 Register								Operation Clock (f_{MCK}) Note	
CKS10	PRS 113	PRS 112	PRS 111	PRS 110	PRS 103	PRS 102	PRS 101	PRS 100		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM03	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM03	
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 (ST1 = 000FH) the operation of the serial array unit (SAU). When selecting INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

Remark X: Don't care

Here is an example of setting an IIC transfer rate where MCK = f_{CLK} = 20 MHz.

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 20 MHz			
	Operation Clock (f_{MCK})	SDR10[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	f_{CLK}	99	100 kHz	0.0%
400 kHz	f_{CLK}	24	400 kHz	0.0%

11.8.6 Procedure for processing errors that occurred during simplified I²C (IIC20) communication

The procedure for processing errors that occurred during simplified I²C (IIC20) communication is described in Figures 11-107 and 11-108.

Figure 11-107. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads SDR10 register. →	The BFF10 bit is set to 0 and channel 0 is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads SSR10 register.		Error type is identified and the read value is used to clear error flag.
Writes 1 to SIR10 register. →	Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSR10 register to the SIR10 register without modification.

Figure 11-108. Processing Procedure in Case of Parity Error (ACK error) in Simplified I²C Mode

Software Manipulation	Hardware Status	Remark
Reads SDR10 register. →	The BFF10 bit is set to 0 and channel 0 is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads SSR10 register.		Error type is identified and the read value is used to clear error flag.
Writes SIR10 register. →	Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSR10 register to the SIR10 register without modification.
Sets ST10 bit to 1. →	The BFF10 bit is set to 0 and channel 0 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 retry from address transmission.
Creates stop condition.		
Creates start condition.		
Sets SS10 bit to 1. →	The BFF10 bit is set to 1 and channel 0 is enabled to receive data.	

11.9 Relationship Between Register Settings and Pins

Tables 11-5 to 11-8 show the relationship between register settings and pins for each channel of serial array units 0 and 1.

Table 11-5. Relationship between register settings and pins
(Channel 0 of unit 1: CSI20, UART2 transmission, IIC20)

SE10 Note 1	MD102	MD101	SOE10	SO10	CKO10	TXE10	RXE10	PM142	P142	PM143 Note 2	P143 Note 2	PM144	P144	Operation mode	Pin Function		
															$\overline{\text{SCK20}}$/SCL20/P142	SI20/SDA20/RxD2/P143 Note 2	SO20/TxD2/P144
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	P142	P143	P144
	0	1														P143/RxD2	
	1	0														P143	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI20 reception	$\overline{\text{SCK20}}$ (input)	SI20	P144
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI20 transmission	$\overline{\text{SCK20}}$ (input)	P143	SO20
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI20 transmission/reception	$\overline{\text{SCK20}}$ (input)	SI20	SO20
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI20 reception	$\overline{\text{SCK20}}$ (output)	SI20	P144
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI20 transmission	$\overline{\text{SCK20}}$ (output)	P143	SO20
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI20 transmission/reception	$\overline{\text{SCK20}}$ (output)	SI20	SO20
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART2 transmission Note 5	P142	P143/RxD2	TxD2
0	1	0	0	0/1 Note 6	0/1 Note 6	0	0	0	1	0	1	× Note 3	× Note 3	IIC20 start condition	SCL20	SDA20	P144
						1	0										
						0	1										
1			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC20 address field transmission	SCL20	SDA20	P144
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC20 data transmission	SCL20	SDA20	P144
			1	0/1 Note 4	0/1 Note 4	0	1	0	1	0	1	× Note 3	× Note 3	IIC20 data reception	SCL20	SDA20	P144
0			0	0/1 Note 7	0/1 Note 7	0	0	0	1	0	1	× Note 3	× Note 3	IIC20 stop condition	SCL20	SDA20	P144
						1	0										
						0	1										

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. When channel 1 of unit 1 is set to UART2 reception, this pin becomes an RxD2 function pin (refer to **Table 11-6**). In this case, operation stop mode or UART2 transmission must be selected for channel 0 of unit 1.

3. This pin can be set as a port function pin.

4. This is 0 or 1, depending on the communication operation. For details, refer to **11.3 (12) Serial output register m (SOm)**.

5. When using UART2 transmission and reception in a pair, set channel 1 of unit 1 to UART2 reception (refer to **Table 11-6**).

6. Set the CKO10 bit to 1 before a start condition is generated. Clear the SO10 bit from 1 to 0 when the start condition is generated.

7. Set the CKO10 bit to 1 before a stop condition is generated. Clear the SO10 bit from 0 to 1 when the stop condition is generated.

Remark X: Don't care

Table 11-6. Relationship between register settings and pins (Channel 1 of unit 1: UART2 reception)

SE11 Note 1	MD112	MD111	TXE11	RXE11	PM143 Note 2	P143 Note 2	Operation mode	Pin Function
								SI20/SDA20/RxD2/P143 Note 2
0	0	1	0	0	× Note 3	× Note 3	Operation stop mode	SI20/SDA20/P143
1	0	1	0	1	1	×	UART2 reception Notes 4, 5	RxD2

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. When channel 1 of unit 1 is set to UART2 reception, this pin becomes an RxD2 function pin. In this case, set channel 0 of unit 1 to operation stop mode or UART2 transmission (refer to **Table 11-5**).
When channel 0 of unit 1 is set to CSI20 or IIC20, this pin cannot be used as an RxD2 function pin. In this case, set channel 1 of unit 1 to operation stop mode.

3. This pin can be set as a port function pin.

4. When using UART2 transmission and reception in a pair, set channel 0 of unit 1 to UART2 transmission (refer to **Table 11-5**).

5. The SMR10 register of channel 0 of unit 1 must also be set during UART2 reception. For details, refer to **11.6.2 (1) Register setting**.

Remark X: Don't care

Table 11-7. Relationship between register settings and pins (Channel 2 of unit 1: UART3 transmission)

SE12 Note1	MD122	MD121	SOE12	SO12	TXE12	RXE12	PM13 Note 2	P13 Note 2	Operation mode	Pin Function
										TxD3/P13
0	0	1	0	1	0	0	× Note 3	× Note 3	Operation stop mode	P13
1	0	1	1	0/1 Note 4	1	0	0	1	UART3 transmission Note 5	TxD3

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. When channel 3 of unit 1 is set to UART1 reception, this pin becomes an RxD3 function pin (refer to **Table 11-8**). In this case, set channel 2 of unit 1 to operation stop mode or UART2 transmission.

3. This pin can be set as a port function pin.

4. This is 0 or 1, depending on the communication operation. For details, refer to **11.3 (12) Serial output register m (SOm)**.

5. When using UART3 transmission and reception in a pair, set channel 3 of unit 1 to UART3 reception (refer to **Table 11-8**).

Remark X: Don't care

Table 11-8. Relationship between register settings and pins (Channel 3 of unit 1: UART3 reception)

SE13 Note 1	MD132	MD131	TXE13	RXE13	PM14 Note 2	P14 Note 2	Operation mode	Pin Function
								RxD3/P14 Note 2
0	0	1	0	0	× Note 3	× Note 3	Operation stop mode	P14
1	0	1	0	1	1	×	UART3 reception Notes 4, 5	RxD3

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. When channel 1 of unit 1 is set to UART3 reception, this pin becomes an RxD3 function pin. In this case, set channel 2 of unit 1 to operation stop mode or UART3 transmission (refer to T**able 11-7**).

3. This pin can be set as a port function pin.

4. When using UART3 transmission and reception in a pair, set channel 2 of unit 1 to UART3 transmission (refer to **Table 11-7**).

5. The SMR12 register of channel 2 of unit 1 must also be set during UART3 reception. For details, refer to **11.6.2 (1) Register setting**.

Remark X: Don't care

CHAPTER 12 SERIAL INTERFACE IICA

12.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL0) line and a serial data bus (SDA0) 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 SCL0 and SDA0 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 (INTIICA) 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 WUP bit of IICA control register 1 (IICCTL1).

Figure 12-1 shows a block diagram of serial interface IICA.

Figure 12-1. Block Diagram of Serial Interface IICA

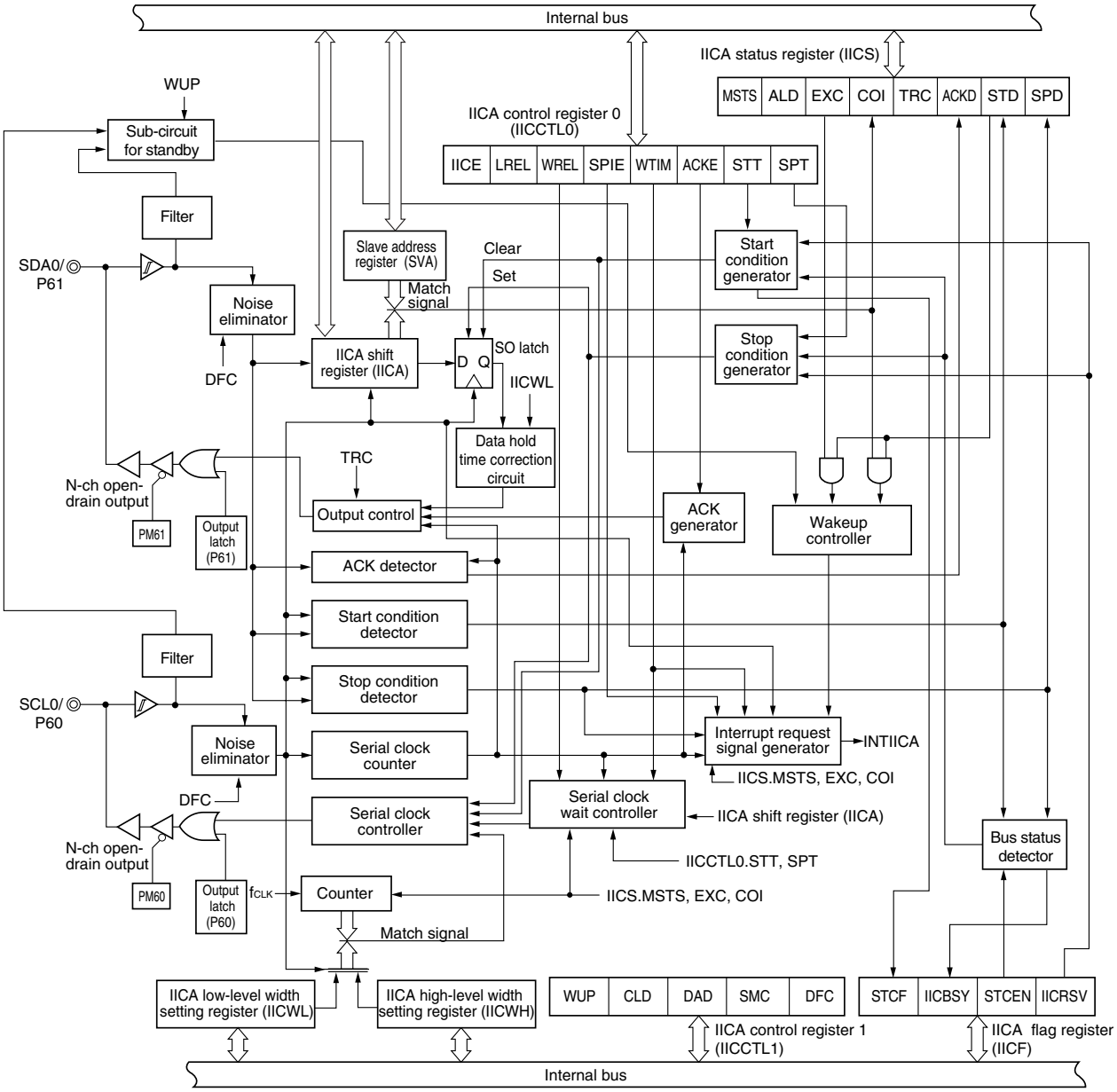
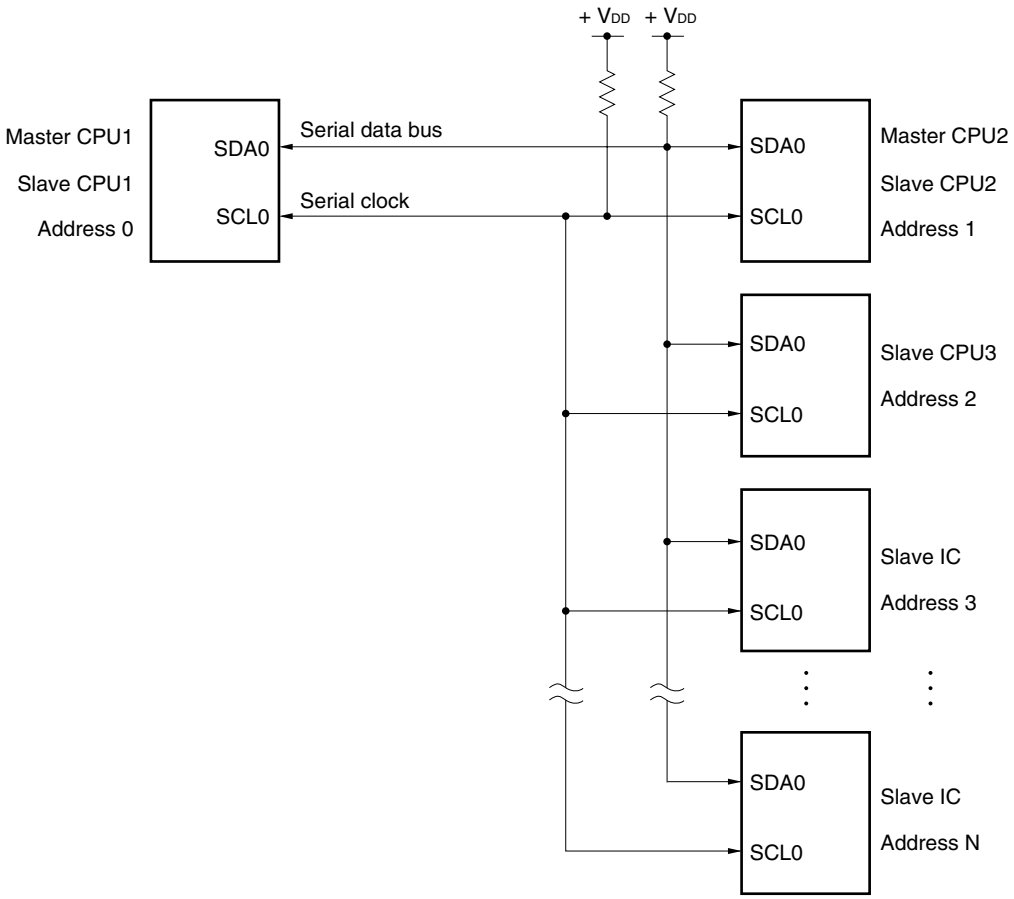


Figure 12-2 shows a serial bus configuration example.

Figure 12-2. Serial Bus Configuration Example Using I²C Bus



12.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 12-1. Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register (IICA) Slave address register (SVA)
Control registers	Peripheral enable register 0 (PER0) IICA control register 0 (IICCTL0) IICA status register (IICS) IICA flag register (IICF) IICA control register 1 (IICCTL1) IICA low-level width setting register (IICWL) IICA high-level width setting register (IICWH) Port mode register 6 (PM6) Port register 6 (P6)

(1) IICA shift register (IICA)

IICA register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. IICA register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to IICA register.

Cancel the wait state and start data transfer by writing data to IICA register during the wait period.

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

Reset signal generation clears IICA to 00H.

Figure 12-3. Format of IICA Shift Register (IICA)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA								

Cautions 1. Do not write data to IICA register during data transfer.

2. Write or read IICA register only during the wait period. Accessing IICA register in a communication state other than during the wait period is prohibited. When the device serves as the master, however, IICA register can be written only once after the communication trigger bit (STT) is set to 1.

3. When communication is reserved, write data to IICA register after the interrupt triggered by a stop condition is detected.

(2) Slave address register (SVA)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

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

However, rewriting to this register is prohibited while STD = 1 (while the start condition is detected).

Reset signal generation clears SVA register to 00H.

Figure 12-4. Format of Slave Address Register (SVA)

Address: F0234H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA	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 SDA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA) when the address received by this register matches the address value set to the slave address register (SVA) 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 (INTIICA).

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 WTIM bit)
- Interrupt request generated when a stop condition is detected (set by SPIE bit)

Remark WTIM bit: Bit 3 of IICA control register 0 (IICCTL0)
SPIE bit: Bit 4 of IICA control register 0 (IICCTL0)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCL0 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 STT 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 SPT 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 STT bit: Bit 1 of IICA control register 0 (IICCTL0)
SPT bit: Bit 0 of IICA control register 0 (IICCTL0)
IICRSV bit: Bit 0 of IICA flag register (IICF)
IICBSY bit: Bit 6 of IICA flag register (IICF)
STCF bit: Bit 7 of IICA flag register (IICF)
STCEN bit: Bit 1 of IICA flag register (IICF)

12.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following eight registers.

- Peripheral enable register 0 (PER0)
- IICA control register 0 (IICCTL0)
- IICA flag register (IICF)
- IICA status register (IICS)
- IICA control register 1 (IICCTL1)
- IICA low-level width setting register (IICWL)
- IICA high-level width setting register (IICWH)
- Port mode register 6 (PM6)
- Port register 6 (P6)

(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 (IICAEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 12-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

IICAEN	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.

Cautions 1. When setting serial interface IICA, be sure to set IICAEN bit to 1 first. If IICAEN = 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) and port register 6 (P6)).

2. Be sure to clear bits 6 and 7 of the PER0 register to 0.

(2) IICA control register 0 (IICCTL0)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

IICCTL0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE, WTIM, and ACKE bits while IICE = 0 or during the wait period. These bits can be set at the same time when the IICE bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Figure 12-6. Format of IICA Control Register 0 (IICCTL0) (1/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTL0	IICE	LREL	WREL	SPIE	WTIM	ACKE	STT	SPT

IICE	I^2C operation enable
0	Stop operation. Reset the IICA status register (IICS)[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCL0 and SDA0 lines are at high level.	
Condition for clearing (IICE = 0)	Condition for setting (IICE = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL[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 SCL0 and SDA0 lines are set to high impedance. The following flags of IICA control register 0 (IICCTL0) and IICA status register (IICS) are cleared to 0. • STT • SPT • MSTS • EXC • COI • TRC • ACKD • STD
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 (LREL = 0)	Condition for setting (LREL = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL[Notes 2, 3]	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When WREL bit is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC = 1), the SDA0 line goes into the high impedance state (TRC = 0).	
Condition for clearing (WREL = 0)	Condition for setting (WREL = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICA status register (IICS), the STCF and IICBSY bits of the IICA flag register (IICF), and the CLD and DAD bits of the IICA control register 1 (IICCTL1) are reset.

2. The signal of this bit is invalid while IICE0 is 0.

3. When the LREL and WREL bits are read, 0 is always read.

Caution If the operation of I^2C is enabled (IICE = 1) when the SCL0 line is high level, the SDA0 line is low level, and the digital filter is turned on (DFC bit of the IICCTL1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LREL bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICE = 1).

Figure 12-6. Format of IICA Control Register 0 (IICCTL0) (2/4)

SPIE[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If WUP bit of the IICA control register 1 (IICCTL1) is 1, no stop condition interrupt will be generated even if SPIE = 1.	
Condition for clearing (SPIE = 0)	Condition for setting (SPIE = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM[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 (WTIM = 0)	Condition for setting (WTIM = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE[Notes 1, 2]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDA0 line is set to low level.
Condition for clearing (ACKE = 0)	Condition for setting (ACKE = 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 12-6. Format of IICA Control Register 0 (IICCTL0) (3/4)

STT[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 STT bit is cleared and the STT 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 ACKE bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPT).
- Setting STT bit to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (STT = 0)	Condition for setting (STT = 1)
• Cleared by setting STT bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL = 1 (exit from communications) • When IICE = 0 (operation stop) • Reset	• Set by instruction

Note The signal of this bit is invalid while IICE0 is 0.

Remarks 1. Bit 1 (STT) becomes 0 when it is read after data setting.

2. IICRSV: Bit 0 of IIC flag register (IICF)
STCF: Bit 7 of IIC flag register (IICF)

Figure 12-6. Format of IICA Control Register 0 (IICCTL0) (4/4)

SPT	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 ACKE bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as start condition trigger (STT).
- SPT bit can be set to 1 only when in master mode.
- When WTIM bit has been cleared to 0, if SPT 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. WTIM bit should be changed from 0 to 1 during the wait period following the output of eight clocks, and SPT bit should be set to 1 during the wait period that follows the output of the ninth clock.
- Setting SPT bit to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (SPT = 0)	Condition for setting (SPT = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE = 0 (operation stop) • Reset	• Set by instruction

Caution When bit 3 (TRC) of the IICA status register (IICS) is set to 1 (transmission status), bit 5 (WREL) of the IICA control register 0 (IICCTL0) is set to 1 during the ninth clock and wait is canceled, after which TRC bit is cleared (reception status) and the SDA0 line is set to high impedance. Release the wait performed while TRC bit is 1 (transmission status) by writing to the IICA shift register.

Remark Bit 0 (SPT) becomes 0 when it is read after data setting.

(3) IICA status register (IICS)

This register indicates the status of I^2C.

IICS register is read by a 1-bit or 8-bit memory manipulation instruction only when STT = 1 and during the wait period.

Reset signal generation clears this register to 00H.

Caution Reading the IICS register while the address match wakeup function is enabled (WUP = 1) in STOP mode is prohibited. When WUP bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA 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 (SPIE = 1) the interrupt generated by detecting a stop condition and read the IICS register after the interrupt has been detected.

Remark STT: bit 1 of the IICA control register 0 (IICCTL0)
WUP: bit 7 of the IICA control register 1 (IICCTL1)

Figure 12-7. Format of IICA Status Register (IICS) (1/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS	MSTS	ALD	EXC	COI	TRC	ACKD	STD	SPD

MSTS	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS = 0)	Condition for setting (MSTS = 1)
• When a stop condition is detected • When ALD = 1 (arbitration loss) • Cleared by LREL = 1 (exit from communications) • When IICE bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. MSTS bit is cleared.

Condition for clearing (ALD = 0)	Condition for setting (ALD = 1)
• Automatically cleared after IICS register is read[Note] • When IICE 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 IICS register. Therefore, when using the ALD bit, read the data of this bit before the data of the other bits.

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

Figure 12-7. Format of IICA Status Register (IICS) (2/3)

EXC	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC = 0)	Condition for setting (EXC = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE 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).

COI	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI = 0)	Condition for setting (COI = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE bit changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register (SVA)) (set at the rising edge of the eighth clock).

TRC	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDA0 line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRC = 0)	Condition for setting (TRC = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE bit changes from 1 to 0 (operation stop) • Cleared by WREL = 1[Note] (wait cancel) • When ALD bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS, EXC, COI = 0) <Master> • When “1” is output to the 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 (TRC) of the IICA status register (IICS) is set to 1 (transmission status), bit 5 (WREL) of the IICA control register 0 (IICCTL0) is set to 1 during the ninth clock and wait is canceled, after which TRC bit is cleared (reception status) and the SDA0 line is set to high impedance. Release the wait performed while TRC bit is 1 (transmission status) by writing to the IICA shift register.

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

Figure 12-7. Format of IICA Status Register (IICS) (3/3)

ACKD	Detection of acknowledge ($\overline{ACK}$)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD = 0)	Condition for setting (ACKD = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL = 1 (exit from communications) • When IICE bit changes from 1 to 0 (operation stop) • Reset	• After the SDA0 line is set to low level at the rising edge of SCL0 line's ninth clock

STD	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect.

Condition for clearing (STD = 0)	Condition for setting (STD = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL = 1 (exit from communications) • When IICE bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD	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 (SPD = 0)	Condition for setting (SPD = 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 IICE bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

(4) IICA flag register (IICF)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

IICF register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STT 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.

STCEN bit can be used to set the initial value of the IICBSY bit.

IICRSV and STCEN bits can be written only when the operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) = 0). When operation is enabled, the IICF register can be read.

Reset signal generation clears this register to 00H.

Figure 12-8. Format of IICA Flag Register (IICF)

Address: FFF52H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT flag

Condition for clearing (STCF = 0)	Condition for setting (STCF = 1)
• Cleared by STT = 1 • When IICE = 0 (operation stop) • Reset	• Generating start condition unsuccessful and STT bit cleared to 0 when communication reservation is disabled (IICRSV = 1).

IICBSY	I^2C bus status flag
0	Bus release status (communication initial status when STCEN = 1)
1	Bus communication status (communication initial status when STCEN = 0)

Condition for clearing (IICBSY = 0)	Condition for setting (IICBSY = 1)
• Detection of stop condition • When IICE = 0 (operation stop) • Reset	• Detection of start condition • Setting of IICE bit when STCEN = 0

STCEN	Initial start enable trigger
0	After operation is enabled (IICE = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN = 0)	Condition for setting (STCEN = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSV	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV = 0)	Condition for setting (IICRSV = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to STCEN bit only when the operation is stopped (IICE = 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 (STT = 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 (IICE = 0).

Remark STT: Bit 1 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

(5) IICA control register 1 (IICCTL1)

This register is used to set the operation mode of I^2C and detect the statuses of the SCL0 and SDA0 pins.

IICCTL1 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD and DAD bits are read-only.

Set the IICCTL1 register, except the WUP bit, while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation clears this register to 00H.

Figure 12-9. Format of IICA Control Register 1 (IICCTL1) (1/2)

Address: F0231H After reset: 00H R/W[Note 1]

Symbol	7	6	<5>	<4>	<3>	<2>	1	0
IICCTL1	WUP	0	CLD	DAD	SMC	DFC	0	0

WUP	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 WUP = 1, execute the STOP instruction at least three clocks after setting (1) WUP bit (see **Figure 12-22 Flow When Setting WUP = 1**).

Clear (0) WUP bit after the address has matched or an extension code has been received. The subsequent communication can be entered by clearing (0) WUP bit. (The wait must be released and transmit data must be written after WUP bit has been cleared (0).)

The interrupt timing when the address has matched or when an extension code has been received, while WUP = 1, is identical to the interrupt timing when WUP = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP = 1, a stop condition interrupt is not generated even if the SPIE bit is set to 1.

When WUP = 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 STT bit, without waiting for the detection of the subsequent start condition or stop condition.

Condition for clearing (WUP = 0)	Condition for setting (WUP = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when MSTS, EXC, and COI bits are “0”, and STD bit also “0” (communication not entered))[Note 2]

Notes 1. Bits 4 and 5 are read-only.

2. The status of IICA status register (IICS) must be checked and WUP bit must be set during the period shown below.

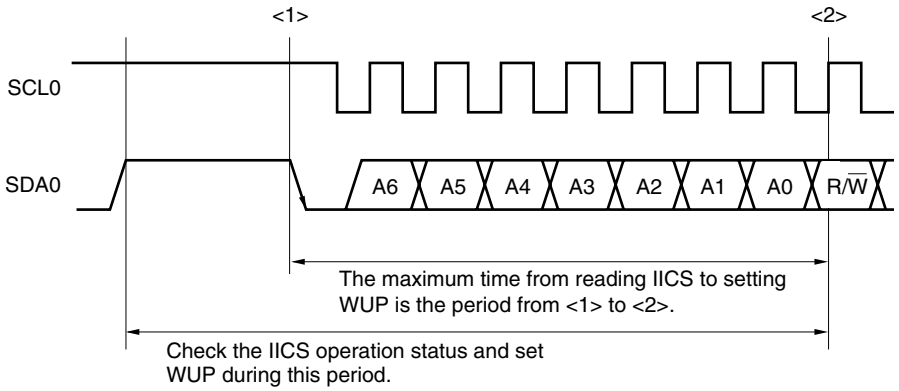


Figure 12-9. Format of IICA Control Register 1 (IICCTL1) (2/2)

CLD	Detection of SCL0 pin level (valid only when IICE = 1)
0	The SCL0 pin was detected at low level.
1	The SCL0 pin was detected at high level.
Condition for clearing (CLD = 0)	Condition for setting (CLD = 1)
• When the SCL0 pin is at low level • When IICE = 0 (operation stop) • Reset	• When the SCL0 pin is at high level

DAD	Detection of SDA0 pin level (valid only when IICE = 1)
0	The SDA0 pin was detected at low level.
1	The SDA0 pin was detected at high level.
Condition for clearing (DAD = 0)	Condition for setting (DAD = 1)
• When the SDA0 pin is at low level • When IICE = 0 (operation stop) • Reset	• When the SDA0 pin is at high level

SMC	Operation mode switching
0	Operates in standard mode.
1	Operates in fast mode.

DFC	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 DFC bit being set (1) or cleared (0). The digital filter is used for noise elimination in fast mode.	

Remark IICE: Bit 7 of IICA control register 0 (IICCTL0)

(6) IICA low-level width setting register (IICWL)

This register is used to set the low-level width of the SCL0 pin signal that is output by serial interface IICA.

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

Set IICWL register while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation sets this register to FFH.

Figure 12-10. Format of IICA Low-Level Width Setting Register (IICWL)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWL								

(7) IICA high-level width setting register (IICWH)

This register is used to set the high-level width of the SCL0 pin signal that is output by serial interface IICA.

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

Set IICWL register while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation sets this register to FFH.

Figure 12-11. Format of IICA High-Level Width Setting Register (IICWH)

Address: F0233H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWH								

Remark For how to set the transfer clock by using the IICWL and IICWH registers, see **12.4.2 Setting transfer clock by using IICWL and IICWH registers**.

(8) Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCL0 pin as clock I/O and the P61/SDA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set IICE bit (bit 7 of IICA control register 0 (IICCTL0)) to 1 before setting the output mode because the P60/SCL0 and P61/SDA0 pins output a low level (fixed) when IICE bit is 0.

PM6 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 12-12. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	0	0	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1, 4 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

12.4 I²C Bus Mode Functions

12.4.1 Pin configuration

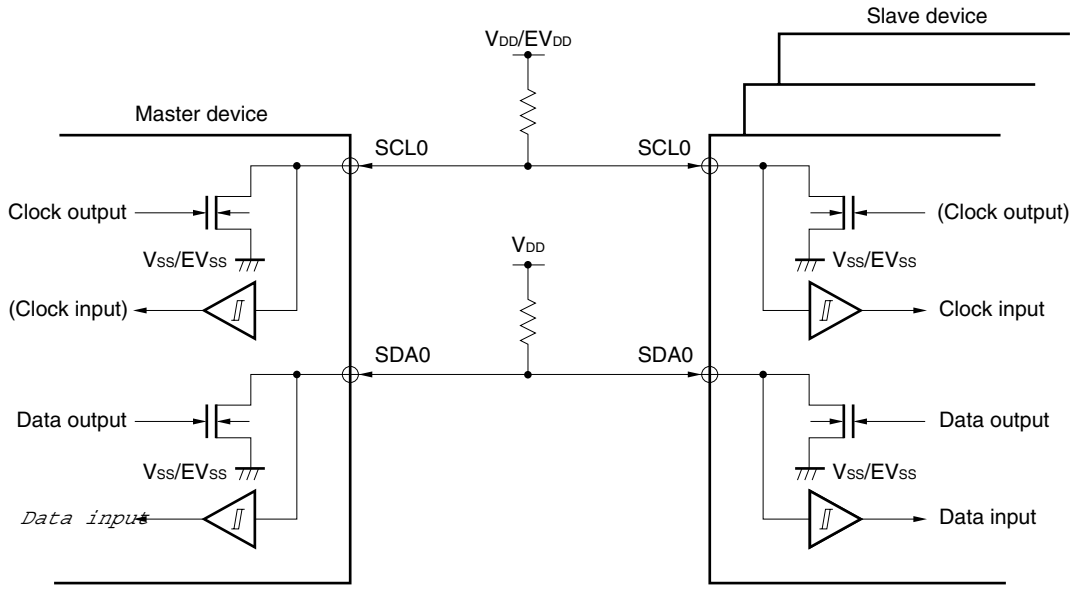
The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

(1) SCL0 ..... 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) SDA0..... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 12-13. Pin Configuration Diagram



12.4.2 Setting transfer clock by using IICWL and IICWH registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{CLK}}{IICWL + IICWH + f_{CLK}(t_R + t_F)}$$

At this time, the optimal setting values of IICWL and IICWH registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWL = \frac{0.52}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{CLK}$$

- When the normal mode

$$IICWL = \frac{0.47}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{CLK}$$

(2) Setting IICWL and IICWH registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICWL = 1.3\ \mu s \times f_{CLK}$$
$$IICWH = (1.2\ \mu s - t_R - t_F) \times f_{CLK}$$

- When the normal mode

$$IICWL = 4.7\ \mu s \times f_{CLK}$$
$$IICWH = (5.3\ \mu s - t_R - t_F) \times f_{CLK}$$

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.)
Standard mode: f_{CLK} = 1 MHz (MIN.)

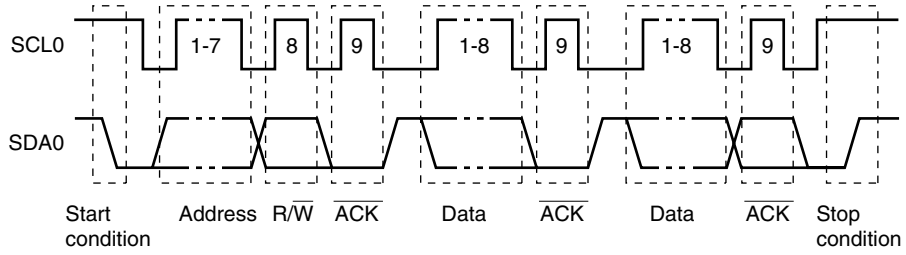
Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDA0 and SCL0 signals separately, because they differ depending on the pull-up resistance and wire load.

2. IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
t_R: SDA0 and SCL0 signal rising times
f_{CLK}: CPU/peripheral hardware clock frequency

12.5 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. Figure 12-14 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 12-14. I²C Bus Serial Data Transfer Timing



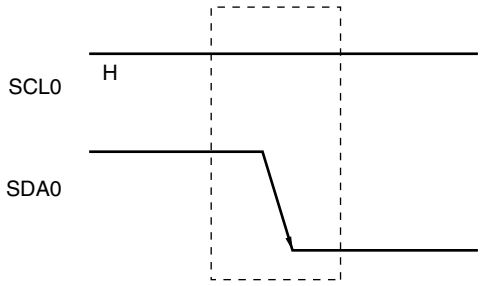
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL0) is continuously output by the master device. However, in the slave device, the SCL0 pin low level period can be extended and a wait can be inserted.

12.5.1 Start conditions

A start condition is met when the SCL0 pin is at high level and the SDA0 pin changes from high level to low level. The start conditions for the SCL0 pin and SDA0 pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 12-15. Start Conditions



A start condition is output when bit 1 (STT) of IICA control register 0 (IICCTL0) is set (1) after a stop condition has been detected (SPD: Bit 0 of the IICA status register (IICS) = 1). When a start condition is detected, bit 1 (STD) of IICS register is set (1).

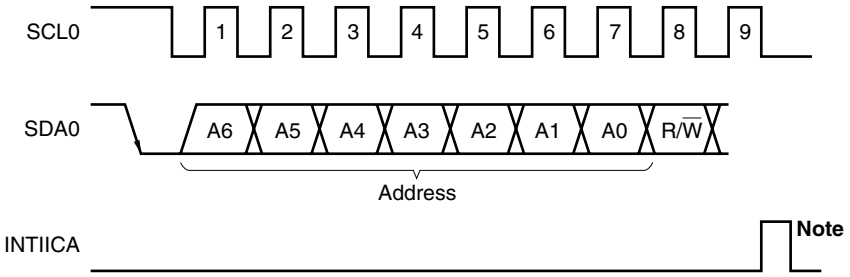
12.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register (SVA). If the address data matches the SVA register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 12-16. Address



Note INTIICA is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **12.5.3 Transfer direction specification** are written to the IICA shift register (IICA). The received addresses are written to IICA register.

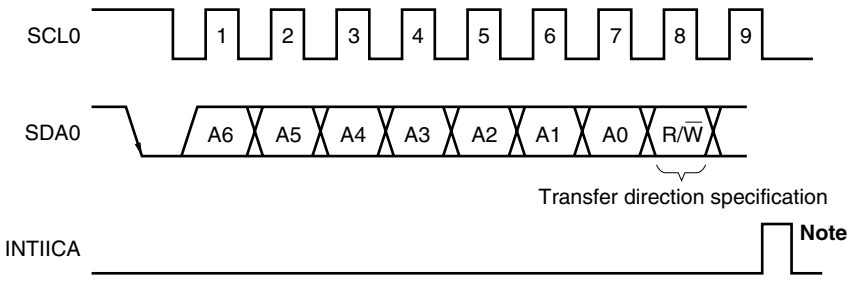
The slave address is assigned to the higher 7 bits of IICA register.

12.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

Figure 12-17. Transfer Direction Specification



Note INTIICA is not issued if data other than a local address or extension code is received during slave device operation.

12.5.4 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 (ACKD) of the IICA status register (IICS).

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.

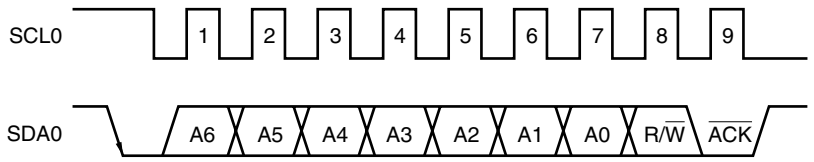
To generate $\overline{ACK}$, the reception side makes the SDA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{ACK}$ is enabled by setting bit 2 (ACKE) of IICA control register 0 (IICCTL0) to 1. Bit 3 (TRC) of the IICS register is set by the data of the eighth bit that follows 7-bit address information. Usually, set ACKE bit to 1 for reception (TRC = 0).

If a slave can receive no more data during reception (TRC = 0) or does not require the next data item, then the slave must inform the master, by clearing ACKE bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC = 0), it must clear ACKE 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 12-18. $\overline{ACK}$



When the local address is received, $\overline{ACK}$ is automatically generated, regardless of the value of ACKE 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 ACKE 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.

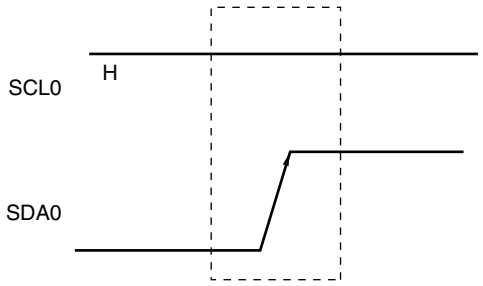
- When 8-clock wait state is selected (bit 3 (WTIM) of IICCTL0 register = 0):
  By setting ACKE bit to 1 before releasing the wait state, $\overline{ACK}$ is generated at the falling edge of the eighth clock of the SCL0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM) of IICCTL0 register = 1):
  $\overline{ACK}$ is generated by setting ACKE bit to 1 in advance.

12.5.5 Stop condition

When the SCL0 pin is at high level, changing the SDA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 12-19. Stop Condition



A stop condition is generated when bit 0 (SPT) of IICA control register 0 (IICCTL0) is set to 1. When the stop condition is detected, bit 0 (SPD) of the IICA status register (IICS) is set to 1 and INTIICA is generated when bit 4 (SPIE) of IICCTL0 register is set to 1.

12.5.6 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL0 pin to low level notifies the communication partner of the wait state. When wait state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 12-20. 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 ACKE = 1)

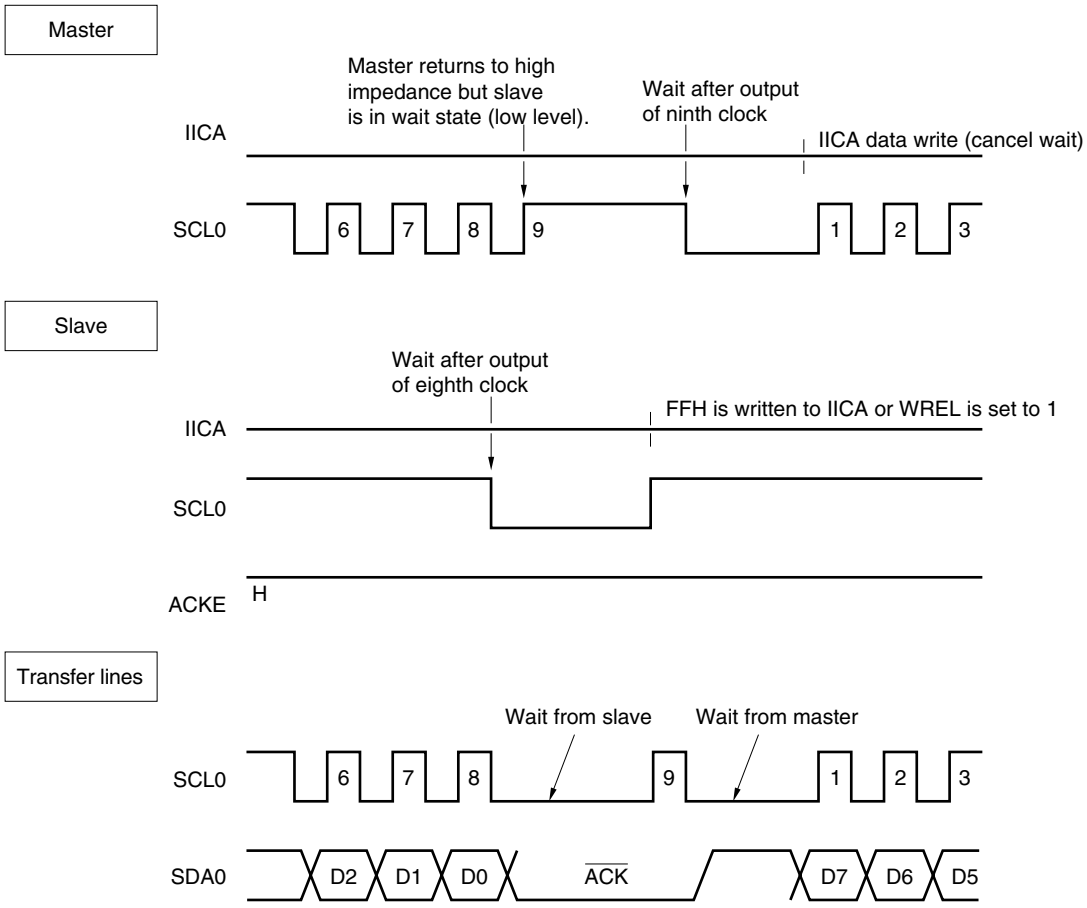
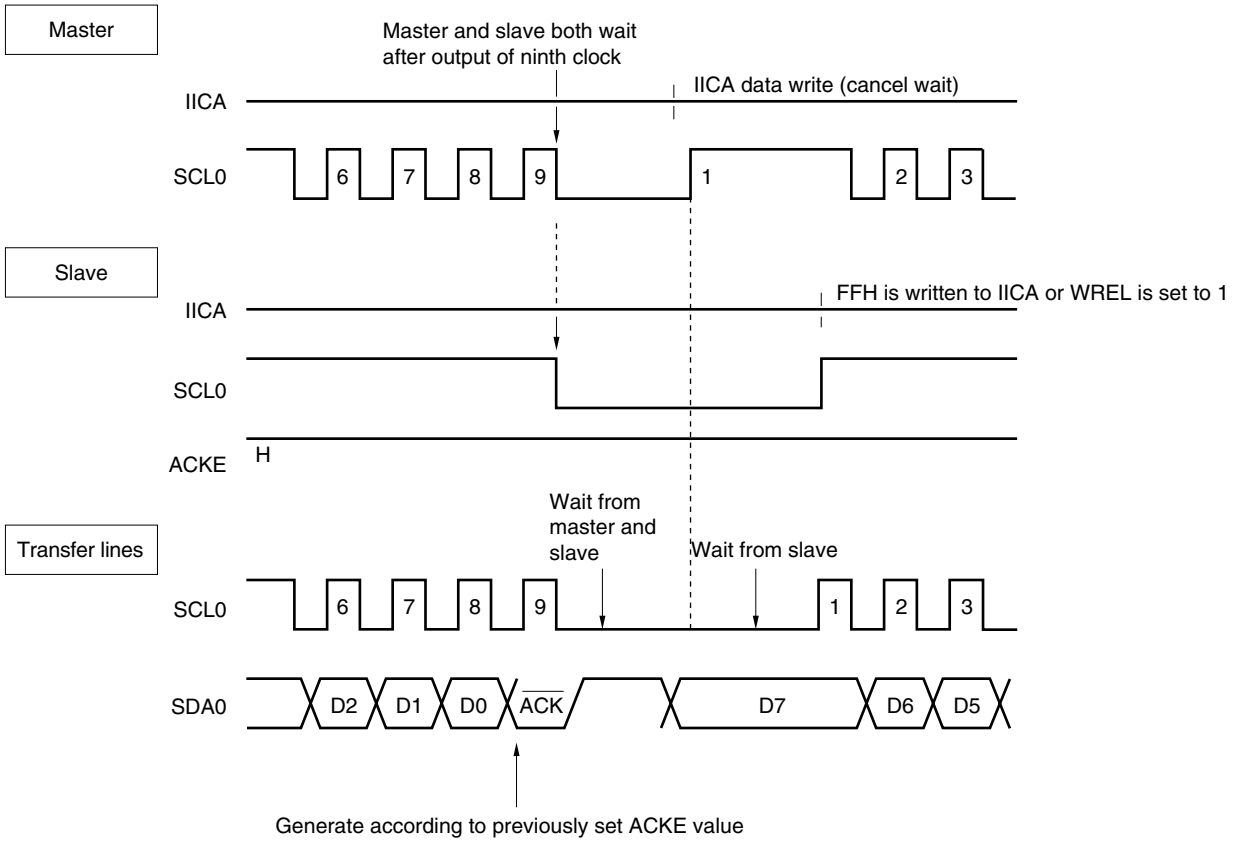


Figure 12-20. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE = 1)



Remark ACKE: Bit 2 of IICA control register 0 (IICCTL0)
WREL: Bit 5 of IICA control register 0 (IICCTL0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM) of IICA control register 0 (IICCTL0).

Normally, the receiving side cancels the wait state when bit 5 (WREL) of IICCTL0 register is set to 1 or when FFH is written to the IICA shift register (IICA), and the transmitting side cancels the wait state when data is written to IICA register.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT) of IICCTL0 register to 1
- By setting bit 0 (SPT) of IICCTL0 register to 1

12.5.7 Canceling wait

The I^2C usually cancels a wait state by the following processing.

- Writing data to IICA shift register (IICA)
- Setting bit 5 (WREL) of IICA control register 0 (IICCTL0) (canceling wait)
- Setting bit 1 (STT) of IICCTL0 register (generating start condition)[Note]
- Setting bit 0 (SPT) of IICCTL0 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 IICA register.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL) of IICCTL0 register to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT) of IICCTL0 register to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT) of IICCTL0 register to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to IICA register after canceling a wait state by setting WREL bit to 1, an incorrect value may be output to SDA0 line because the timing for changing the SDA0 line conflicts with the timing for writing IICA register.

In addition to the above, communication is stopped if IICE 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 (LREL) of IICCTL0 register, so that the wait state can be canceled.

Caution If a processing to cancel a wait state is executed when WUP = 1, the wait state will not be canceled.

12.5.8 Interrupt request (INTIICA) generation timing and wait control

The setting of bit 3 (WTIM) of IICA control register 0 (IICCTL0) determines the timing by which INTIICA is generated and the corresponding wait control, as shown in Table 12-2.

Table 12-2. INTIICA Generation Timing and Wait Control

WTIM	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 INTIICA 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 (SVA).
At this point, $\overline{ACK}$ is generated regardless of the value set to IICCTL0 register's bit 2 (ACKE). For a slave device that has received an extension code, INTIICA occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICA 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 (SVA) and extension code is not received, neither INTIICA 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 WTIM bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to IICA shift register (IICA)
- Setting bit 5 (WREL) of IICA control register 0 (IICCTL0) (canceling wait)
- Setting bit 1 (STT) of IICCTL0 register (generating start condition)Note
- Setting bit 0 (SPT) of IICCTL0 register (generating stop condition)Note

Note Master only.

When an 8-clock wait has been selected (WTIM = 0), the presence/absence of $\overline{ACK}$ generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIICA is generated when a stop condition is detected (only when SPIE = 1).

12.5.9 Address match detection method

In I²C 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 (INTIICA) occurs when the address set to the slave address register (SVA) matches the slave address sent by the master device, or when an extension code has been received.

12.5.10 Error detection

In I²C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by the IICA shift register (IICA) 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.

12.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC) is set to 1 for extension code reception and an interrupt request (INTIICA) is issued at the falling edge of the eighth clock. The local address stored in the slave address register (SVA) 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 SVA register is set to 11110xx0. Note that INTIICA occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC = 1
- Seven bits of data match: COI = 1

Remark EXC: Bit 5 of IICA status register (IICS)
COI: Bit 4 of IICA status register (IICS)

(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 (LREL) of the IICA control register 0 (IICCTL0) to 1 to set the standby mode for the next communication operation.

Table 12-3. Bit Definitions of Major Extension Codes

Slave Address	R/W Bit	Description
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 I²C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

12.5.12 Arbitration

When several master devices simultaneously generate a start condition (when STT bit is set to 1 before STD bit is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD) in the IICA status register (IICS) is set (1) via the timing by which the arbitration loss occurred, and the SCL0 and SDA0 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 ALD = 1 setting that has been made by software.

For details of interrupt request timing, see **12.5.8 Interrupt request (INTIICA) generation timing and wait control**.

Remark STD: Bit 1 of IICA status register (IICS)
STT: Bit 1 of IICA control register 0 (IICCTL0)

Figure 12-21. Arbitration Timing Example

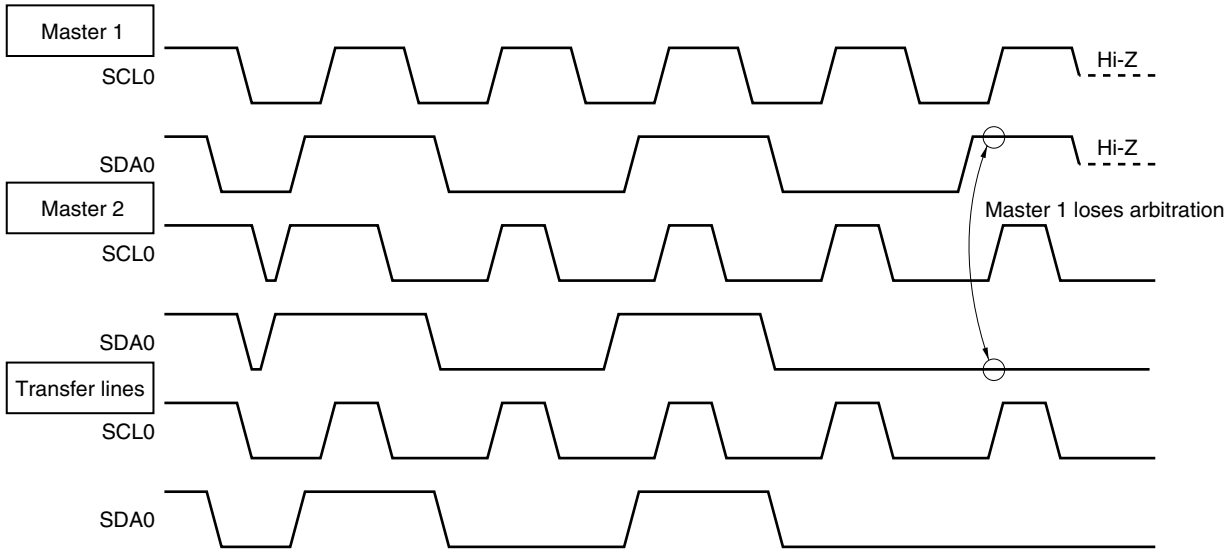


Table 12-4. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ 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 SPIE = 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 SPIE = 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 SCL0 is at low level while attempting to generate a restart condition	

Notes 1. When WTIM bit (bit 3 of IICA control register 0 (IICCTL0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM = 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 SPIE = 1 for master device operation.

Remark SPIE: Bit 4 of IICA control register 0 (IICCTL0)

12.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICA) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICA 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.

However, when a stop condition is detected, bit 4 (SPIE) of IICA control register 0 (IICCTL0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

To use the wakeup function in the STOP mode, set WUP bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA) 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 WUP bit after this interrupt has been generated.

Figure 12-22 shows the flow for setting WUP = 1 and Figure 12-23 shows the flow for setting WUP = 0 upon an address match.

Figure 12-22. Flow When Setting WUP = 1

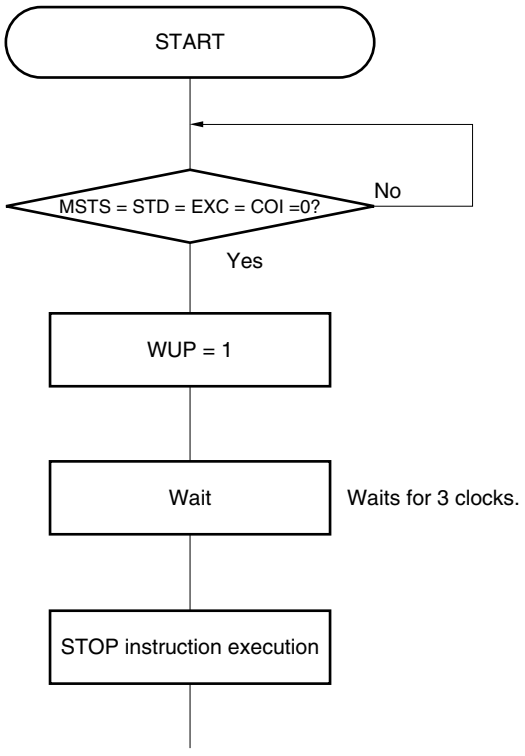
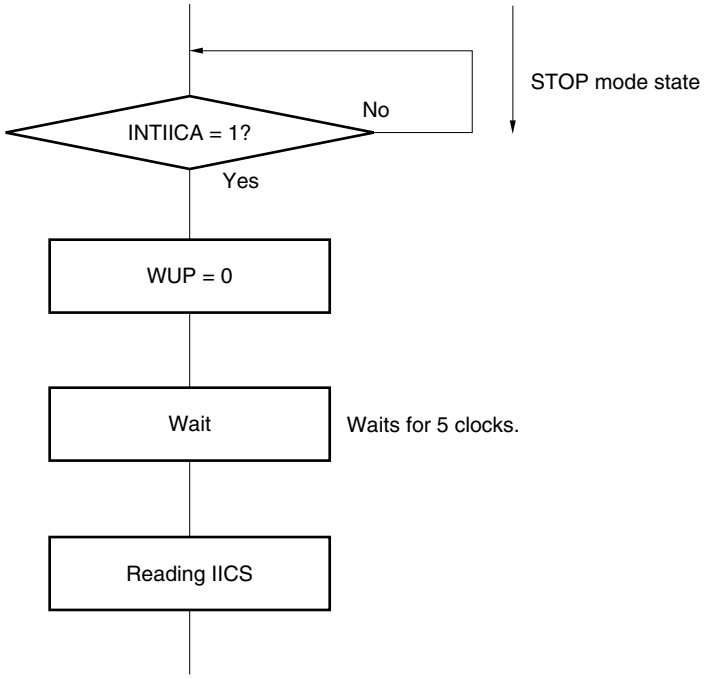


Figure 12-23. Flow When Setting WUP = 0 upon Address Match (Including Extension Code Reception)

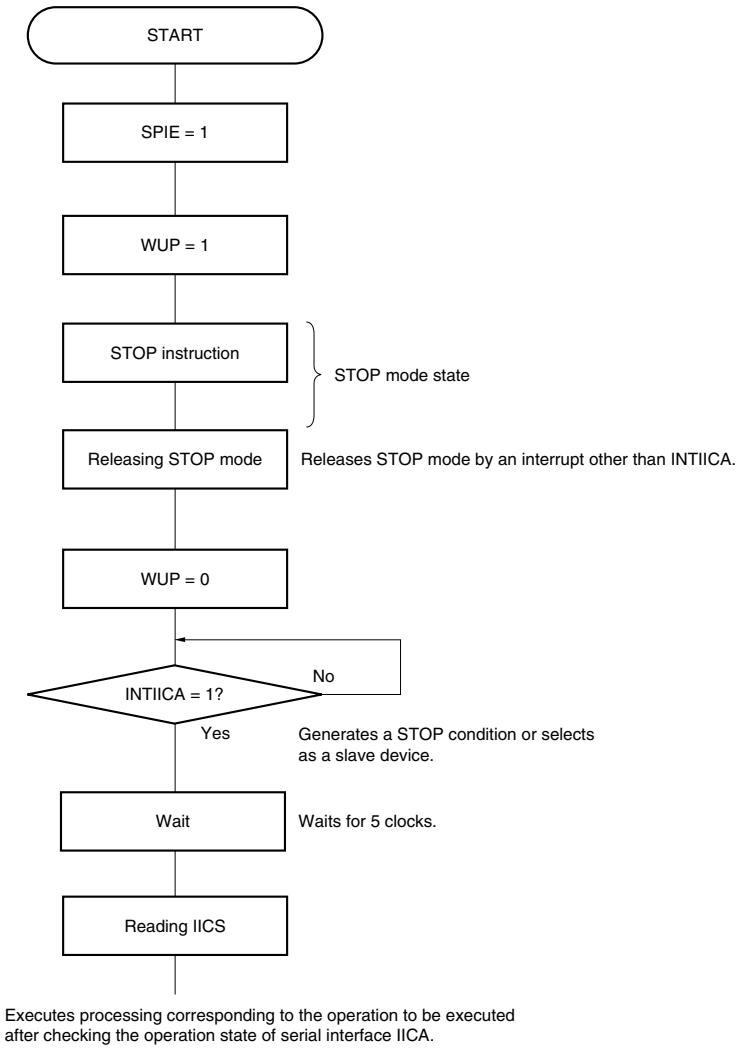


Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA) generated from serial interface IICA.

- Master device operation: Flow shown in Figure 12-24
- Slave device operation: Same as the flow in Figure 12-23

Figure 12-24. When Operating as Master Device after Releasing STOP Mode other than by INTIICA



12.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IICA flag register (IICF) = 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 (LREL) of IICA control register 0 (IICCTL0) to 1 and saving communication).

If bit 1 (STT) of IICCTL0 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 (IICA) after bit 4 (SPIE) of IICCTL0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICA) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to IICA register before the stop condition is detected is invalid.
When STT bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode)..... communication reservation

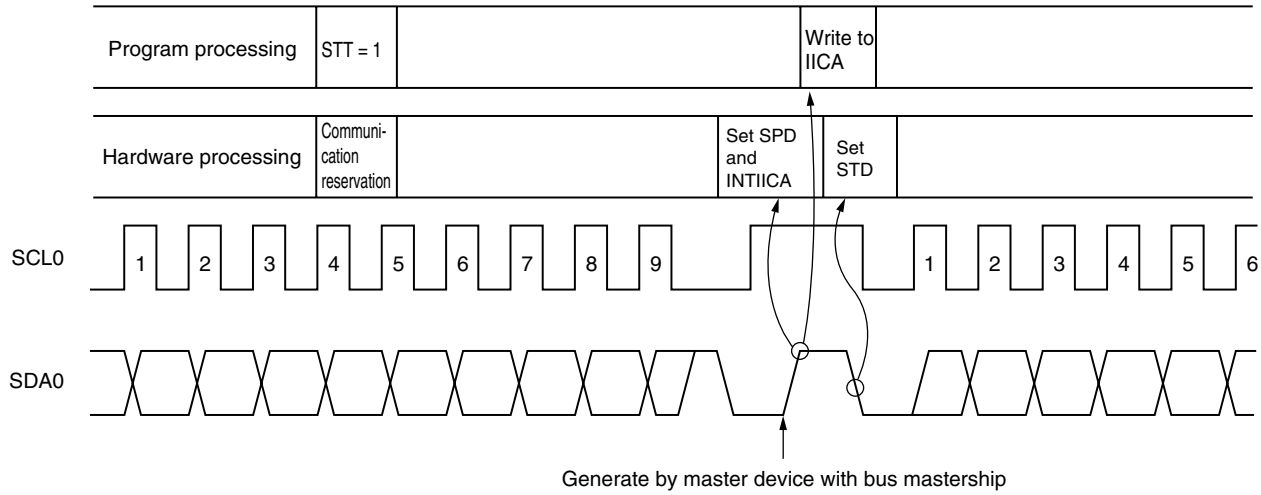
Check whether the communication reservation operates or not by using MSTS bit (bit 7 of the IICA status register (IICS)) after STT bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT = 1 to checking the MSTS flag: (IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clocks]

Remark IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 12-25 shows the communication reservation timing.

Figure 12-25. Communication Reservation Timing



Remark IICA: IICA shift register
STT: Bit 1 of IICA control register 0 (IICCTL0)
STD: Bit 1 of IICA status register (IICS)
SPD: Bit 0 of IICA status register (IICS)

Communication reservations are accepted via the timing shown in Figure 12-26. After bit 1 (STD) of the IICA status register (IICS) is set to 1, a communication reservation can be made by setting bit 1 (STT) of IICA control register 0 (IICCTL0) to 1 before a stop condition is detected.

Figure 12-26. Timing for Accepting Communication Reservations

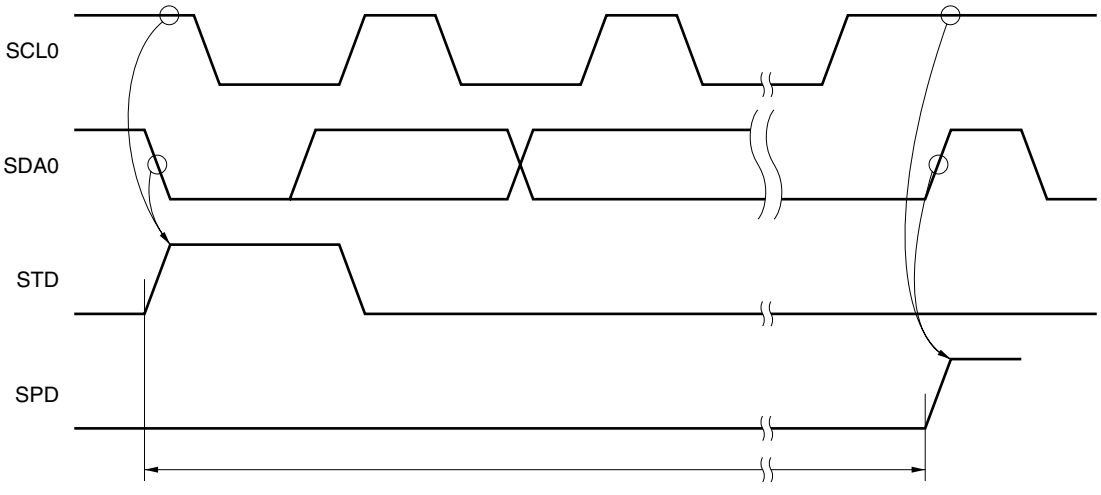
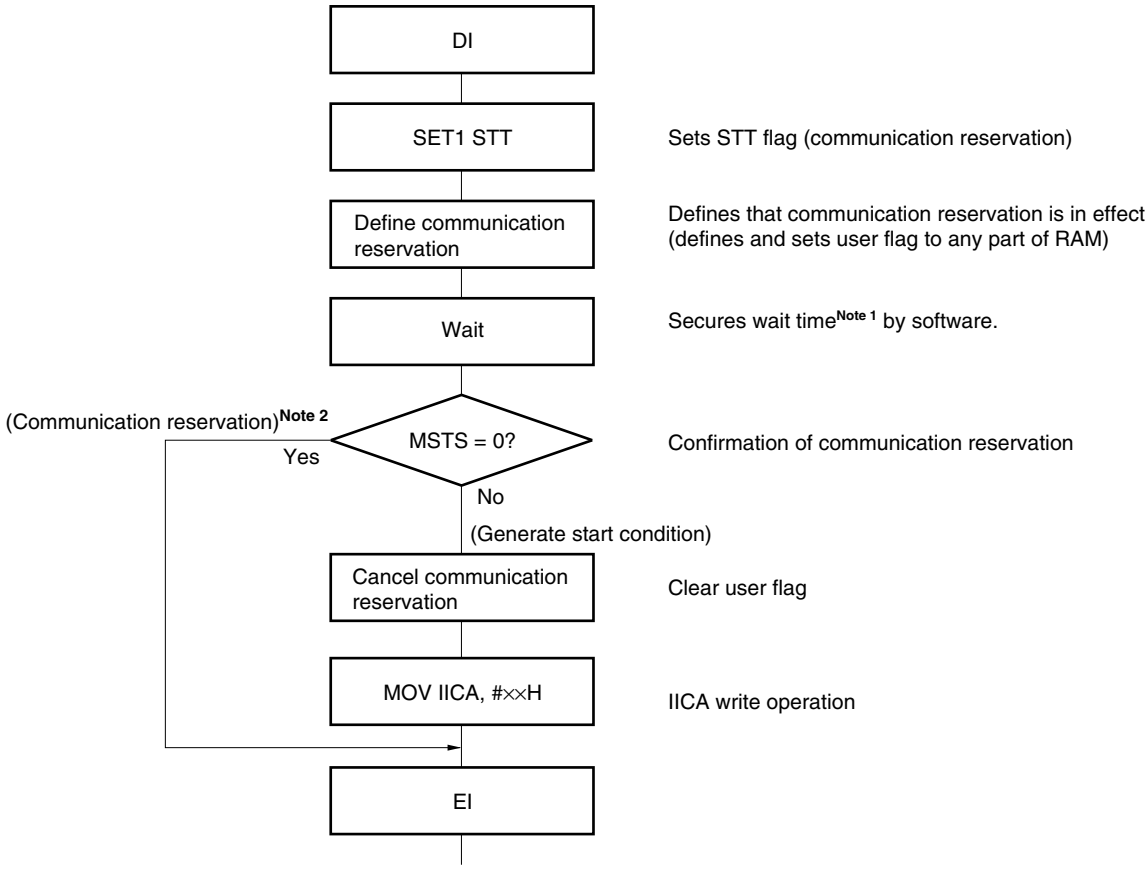


Figure 12-27 shows the communication reservation protocol.

Figure 12-27. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.
(IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clocks]

2. The communication reservation operation executes a write to the IICA shift register (IICA) when a stop condition interrupt request occurs.

Remark STT: Bit 1 of IICA control register 0 (IICCTL0)
MSTS: Bit 7 of IICA status register (IICS)
IICA: IICA shift register
IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IICA flag register (IICF) = 1)

When bit 1 (STT) of IICA control register 0 (IICCTL0) 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 ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL) of IICCTL0 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of IICF register). It takes up to 5 clocks until STCF bit is set to 1 after setting STT = 1. Therefore, secure the time by software.

12.5.15 Cautions

(1) When STCEN = 0

Immediately after I^2C operation is enabled (IICE = 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 1 (IICCTL1).
<2> Set bit 7 (IICE) of IICA control register 0 (IICCTL0) to 1.
<3> Set bit 0 (SPT) of IICCTL0 register to 1.

(2) When STCEN = 1

Immediately after I^2C operation is enabled (IICE = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I^2C communications are already in progress

If I^2C operation is enabled and the device participates in communication already in progress when the SDA0 pin is low and the SCL0 pin is high, the macro of I^2C recognizes that the SDA0 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 (SPIE) of IICCTL0 register to 0 to disable generation of an interrupt request signal (INTIICA) when the stop condition is detected.
<2> Set bit 7 (IICE) of IICCTL0 register to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL) of IICCTL0 register to 1 before $\overline{ACK}$ is returned (4 to 80 clocks after setting IICE bit to 1), to forcibly disable detection.

(4) Setting STT and SPT bits (bits 1 and 0 of IICCTL0 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set SPIE bit (bit 4 of 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 IICA shift register (IICA) 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 SPIE bit to 1 when MSTS bit (bit 7 of IICA status register (IICS)) is detected by software.

12.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the μPD78F8040, 78F8041, 78F8042, 78F8043 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), theμPD78F8040, 78F8041, 78F8042, 78F8043 take 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 μPD78F8040, 78F8041, 78F8042, 78F8043 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

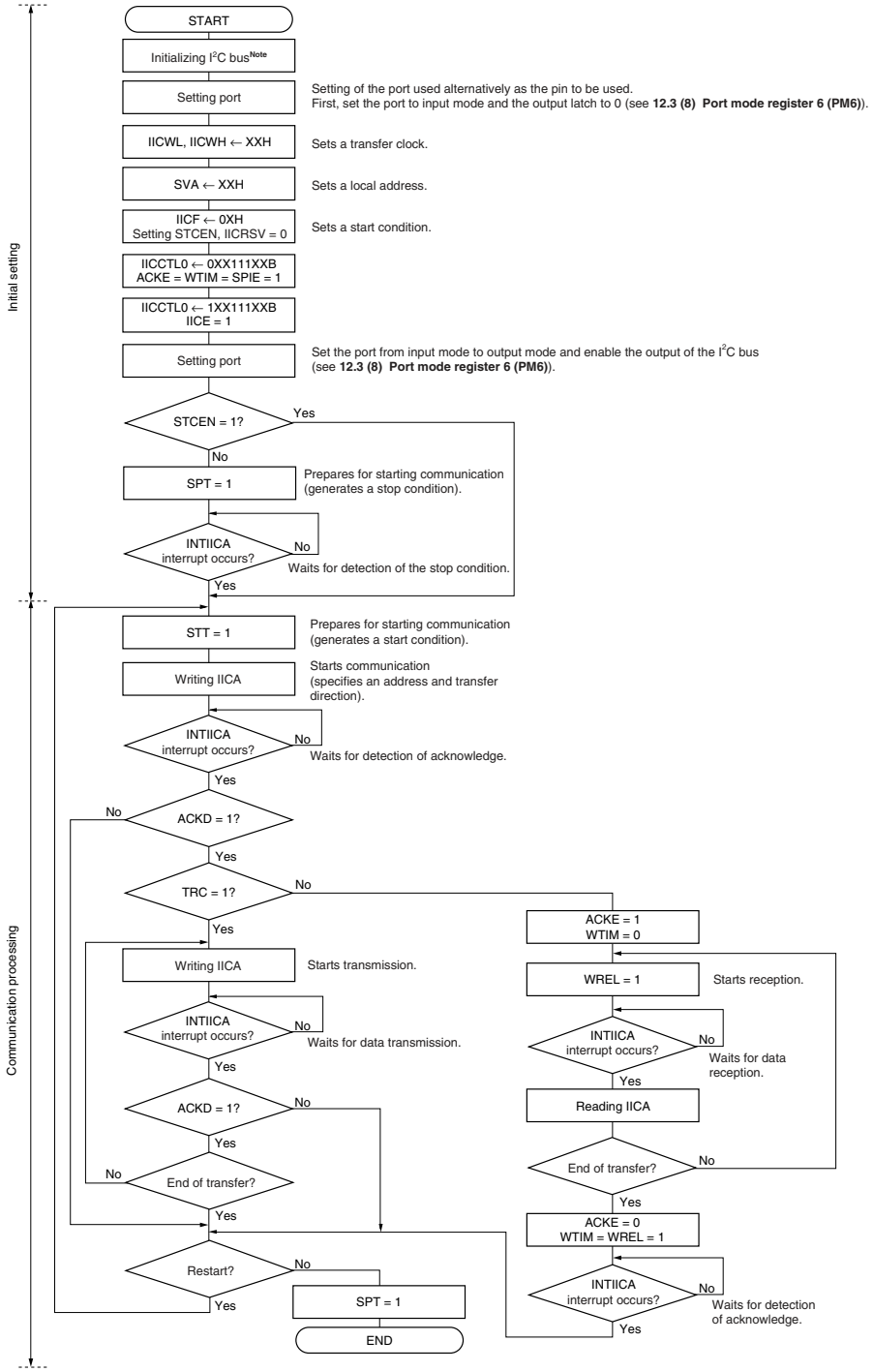
An example of when the μPD78F8040, 78F8041, 78F8042, 78F8043 are used as the I^2C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA interrupt occurrence (communication waiting). When an INTIICA 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 12-28. Master Operation in Single-Master System

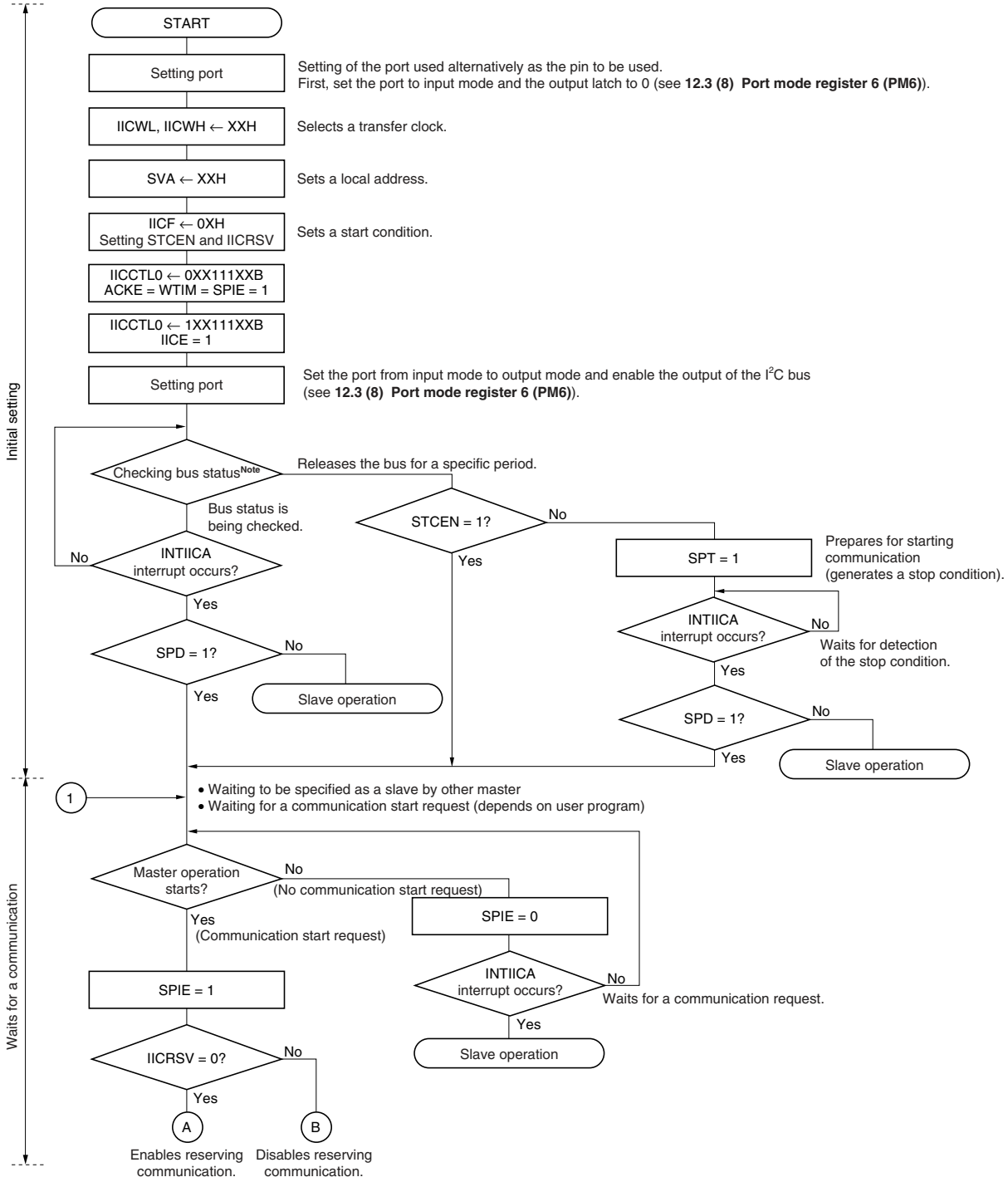


Note Release (SCL0 and SDA0 pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDA0 pin, for example, set the SCL0 pin in the output port mode, and output a clock pulse from the output port until the SDA0 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.

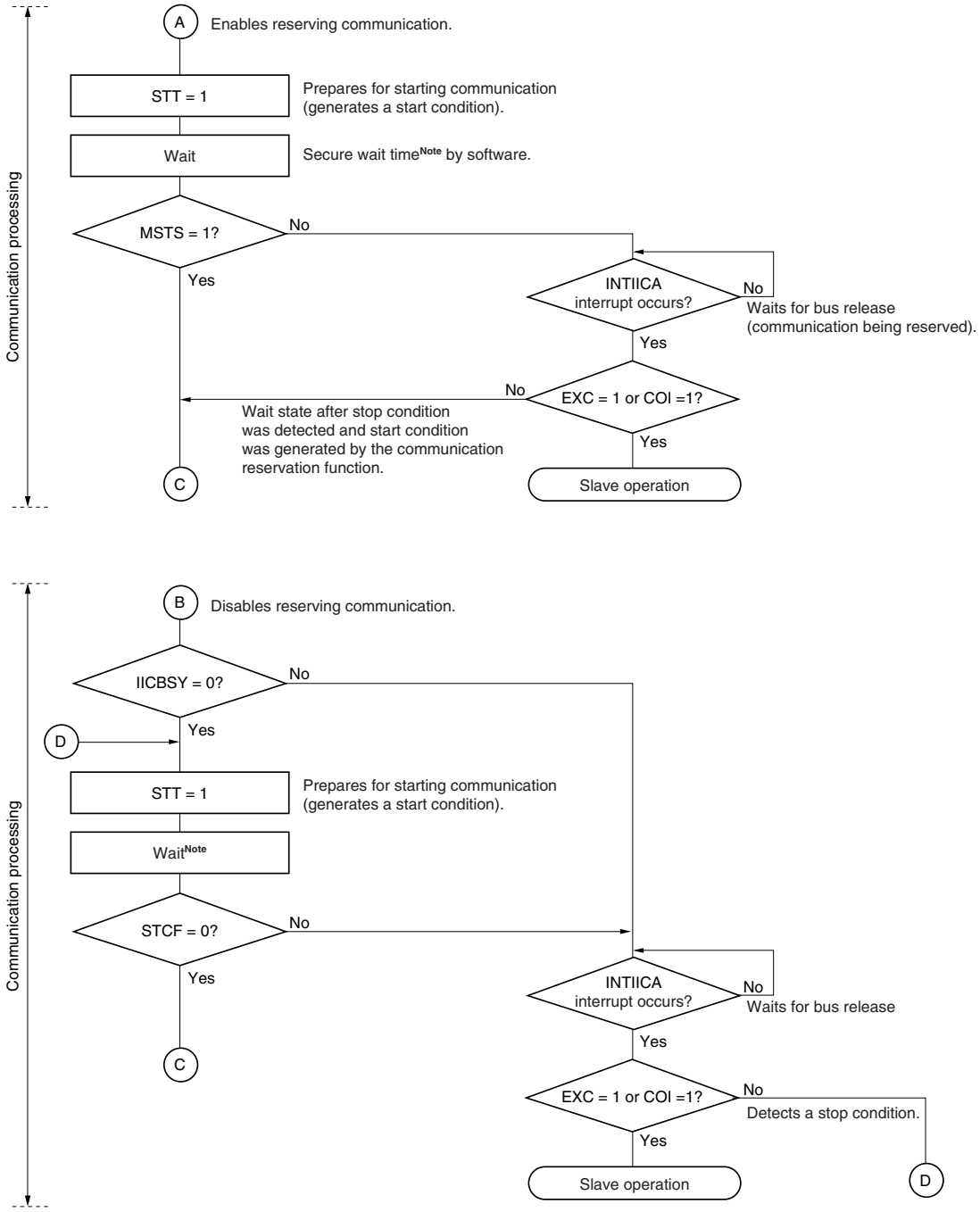
(2) Master operation in multi-master system

Figure 12-29. Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLD bit = 1, DAD bit = 1) for a specific period (for example, for a period of one frame). If the SDA0 pin is constantly at low level, decide whether to release the I²C bus (SCL0 and SDA0 pins = high level) in conformance with the specifications of the product that is communicating.

Figure 12-29. Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.

(IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clocks]

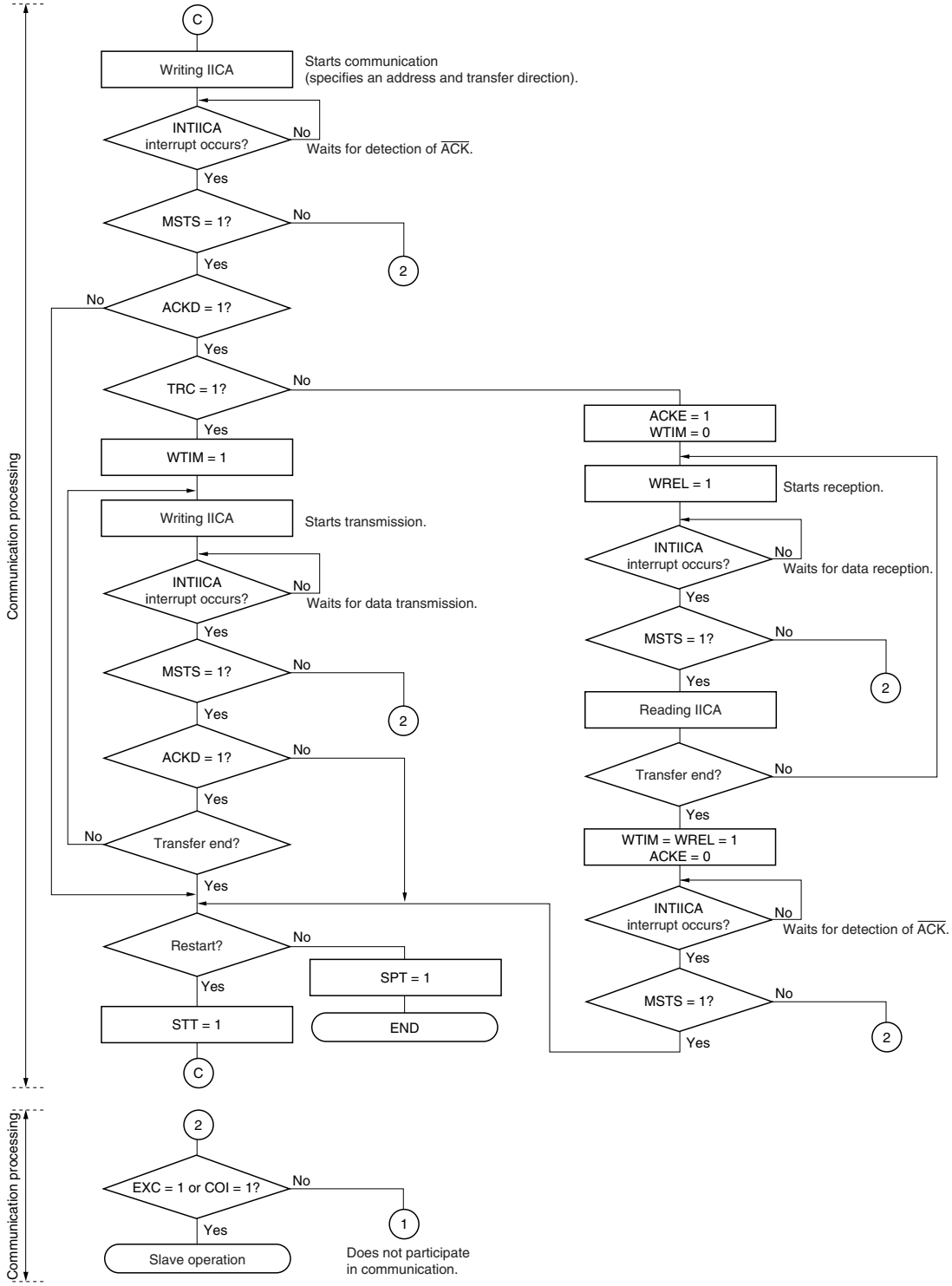
Remark IICWL: IICA low-level width setting register

IICWH: IICA high-level width setting register

t_F: SDA0 and SCL0 signal falling times

f_{CLK}: CPU/peripheral hardware clock frequency

Figure 12-29. 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 MSTS bit each time interrupt INTIICA has occurred to check the arbitration result.

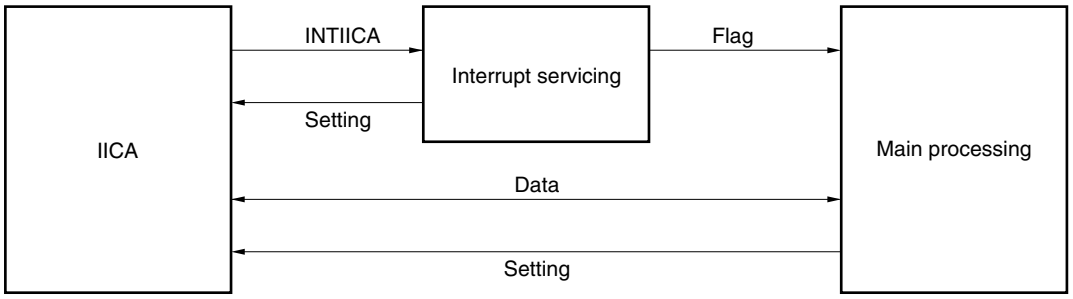
3. To use the device as a slave in a multi-master system, check the status by using the IICA status register (IICS) and IICA flag register (IICF) each time interrupt INTIICA has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA 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 INTIICA 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 INTIICA.

<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 INTIICA 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 TRC bit.

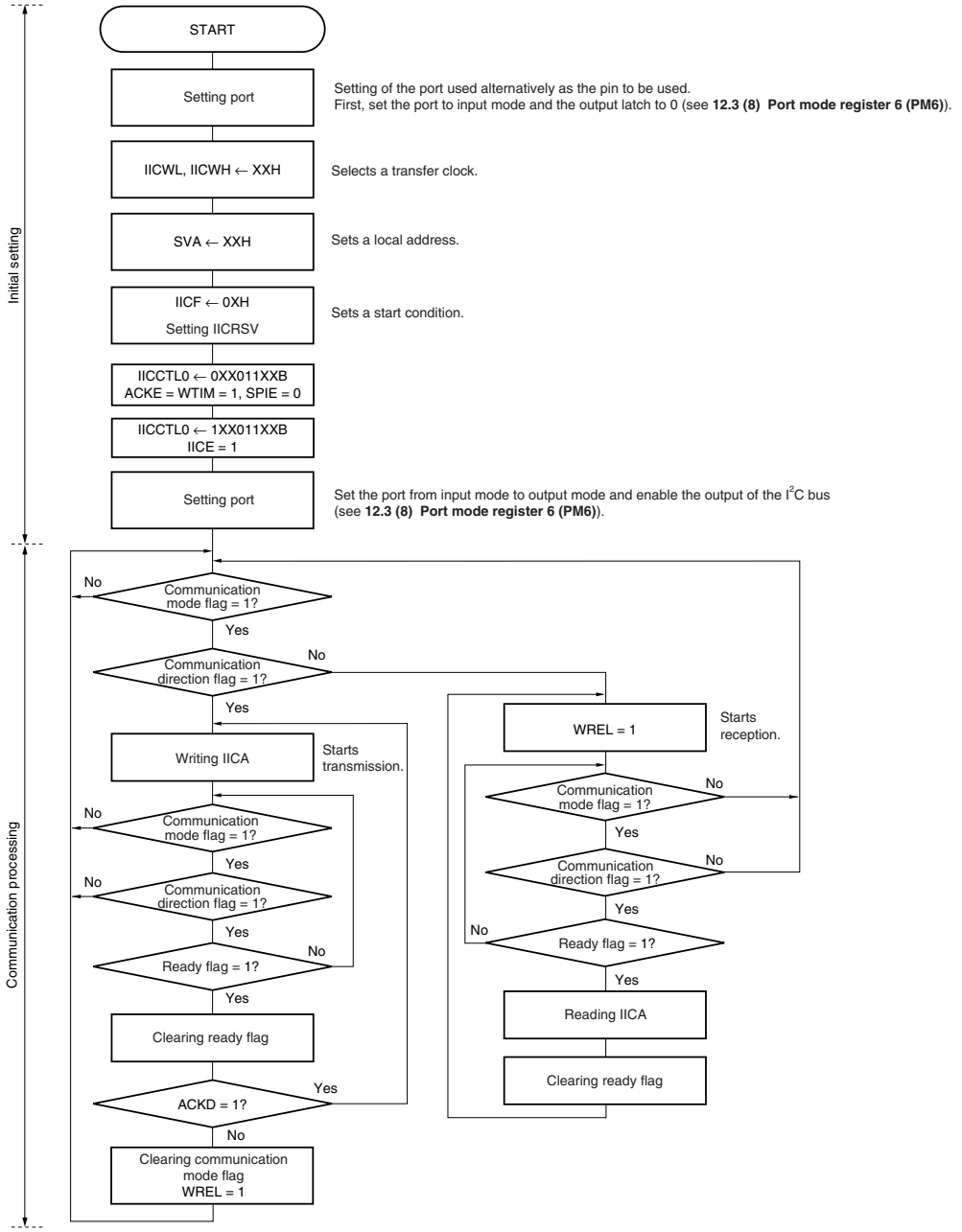
The main processing of the slave operation is explained next.

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 $\overline{ACK}$. If $\overline{ACK}$ is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, $\overline{ACK}$ is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 12-30. Slave Operation Flowchart (1)



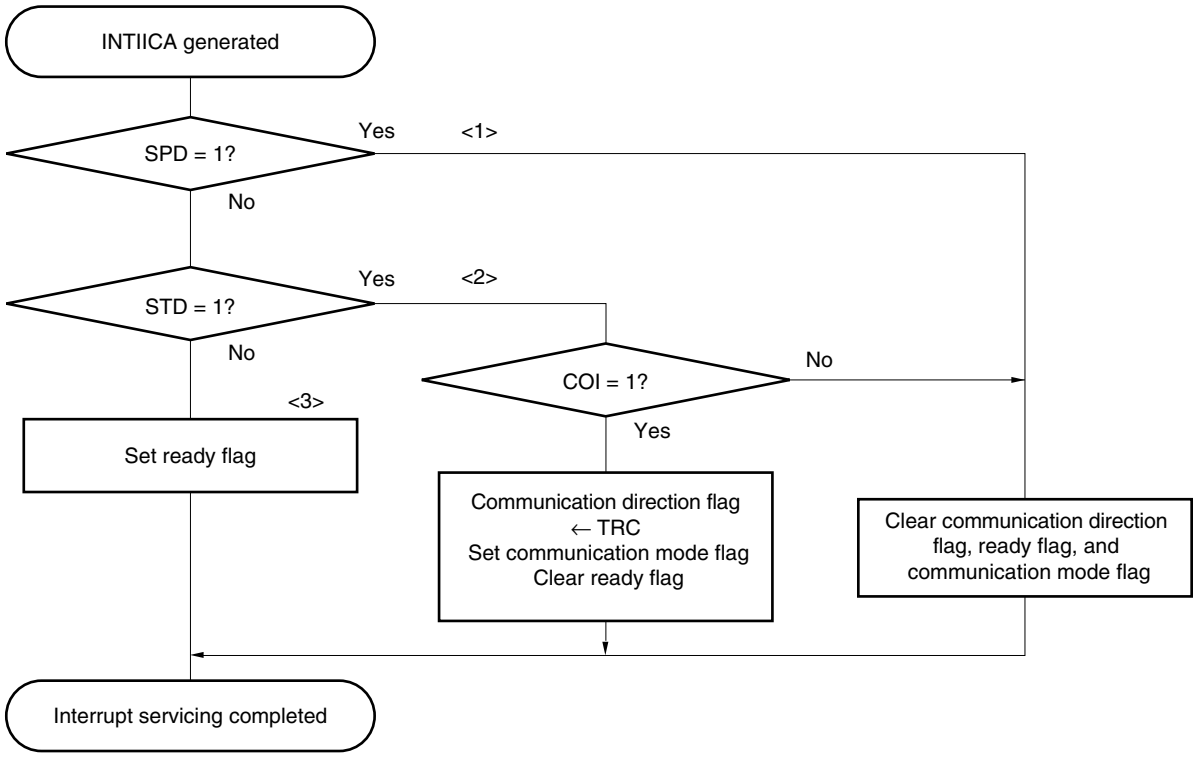
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA interrupt checks the status, and the following operations are performed.

<1> Communication is stopped if the stop condition is issued.
<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in Figure 12-31 Slave Operation Flowchart (2).

Figure 12-31. Slave Operation Flowchart (2)



12.5.17 Timing of I²C interrupt request (INTIICA) occurrence

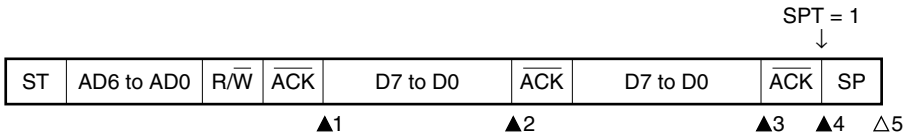
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA, and the value of the IICA status register (IICS) when the INTIICA signal is generated are shown below.

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 WTIM = 0

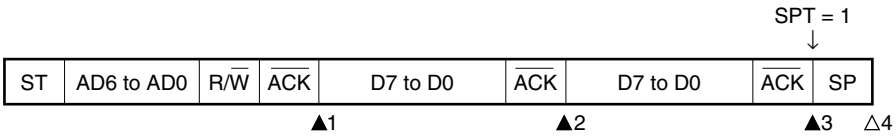


▲1: IICS = 1000×110B
▲2: IICS = 1000×000B
▲3: IICS = 1000×000B (Sets WTIM bit to 1)Note
▲4: IICS = 1000××00B (Sets SPT bit to 1)Note
△5: IICS = 00000001B

Note To generate a stop condition, set WTIM bit to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

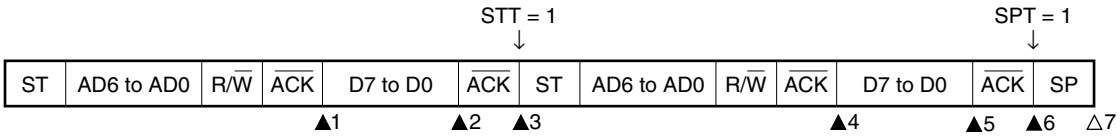


▲1: IICS = 1000×110B
▲2: IICS = 1000×100B
▲3: IICS = 1000××00B (Sets SPT bit to 1)
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM = 0

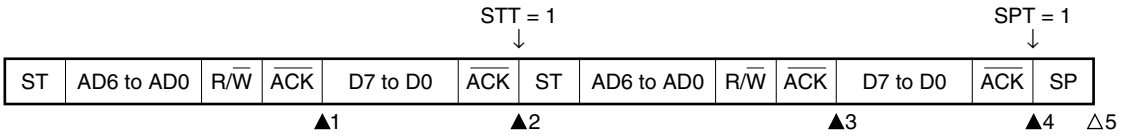


▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM bit to 1)[Note 1]
▲3: IICS = 1000××00B (Clears WTIM bit to 0[Note 2], sets STT bit to 1)
▲4: IICS = 1000×110B
▲5: IICS = 1000×000B (Sets WTIM bit to 1)[Note 3]
▲6: IICS = 1000××00B (Sets SPT bit to 1)
△7: IICS = 00000001B

Notes 1. To generate a start condition, set WTIM bit to 1 and change the timing for generating the INTIICA interrupt request signal.
2. Clear WTIM bit to 0 to restore the original setting.
3. To generate a stop condition, set WTIM bit to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

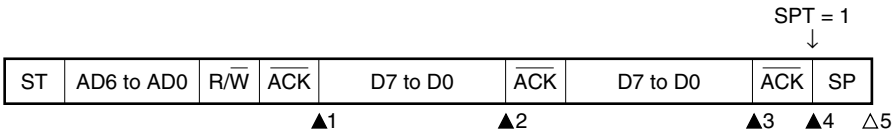


▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (Sets STT bit to 1)
▲3: IICS = 1000×110B
▲4: IICS = 1000××00B (Sets SPT bit to 1)
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM = 0

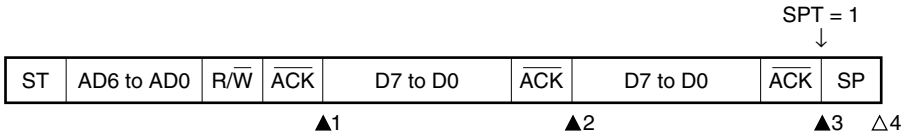


▲1: IICS = 1010×110B
▲2: IICS = 1010×000B
▲3: IICS = 1010×000B (Sets WTIM bit to 1)[Note]
▲4: IICS = 1010××00B (Sets SPT bit to 1)
△5: IICS = 00000001B

Note To generate a stop condition, set WTIM bit to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



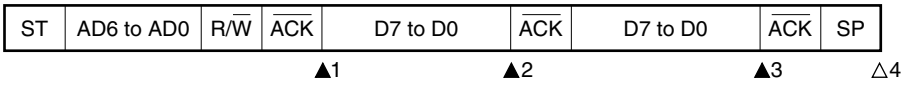
▲1: IICS = 1010×110B
▲2: IICS = 1010×100B
▲3: IICS = 1010××00B (Sets SPT bit to 1)
△4: IICS = 00001001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

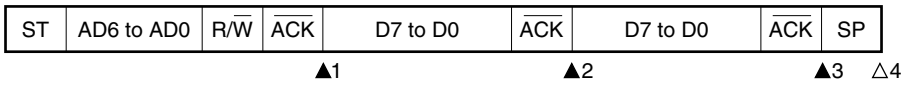
(i) When WTIM = 0



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

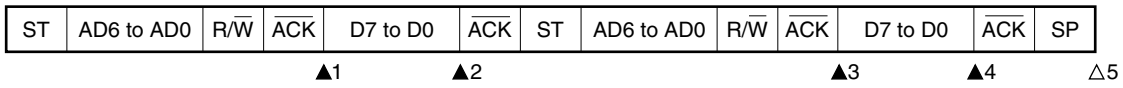


▲1: IICS = 0001×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

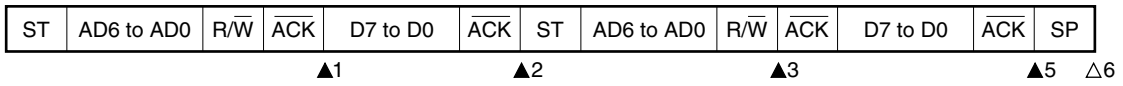
(i) When WTIM = 0 (after restart, matches with SVA)



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×110B
▲4: IICS = 0001×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, matches with SVA)

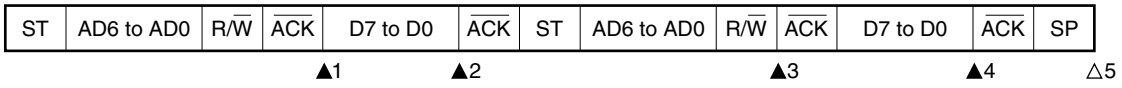


▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 0001×110B
▲4: IICS = 0001××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

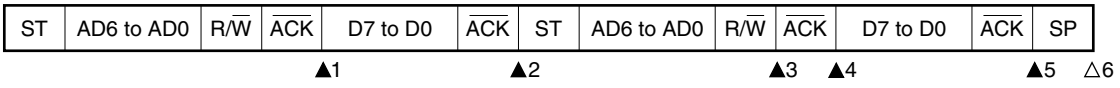
(i) When WTIM = 0 (after restart, does not match address (= extension code))



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0010×010B
▲4: IICS = 0010×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= extension code))



▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 0010×010B
▲4: IICS = 0010×110B
▲5: IICS = 0010××00B
△6: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

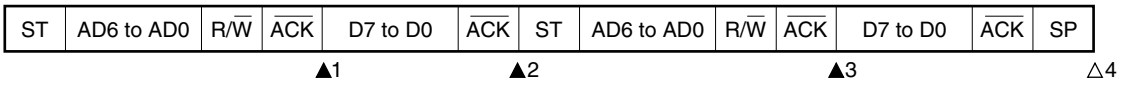
(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM = 0 (after restart, does not match address (= not extension code))

▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 00000110B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= not extension code))



▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 00000110B
△4: IICS = 00000001B

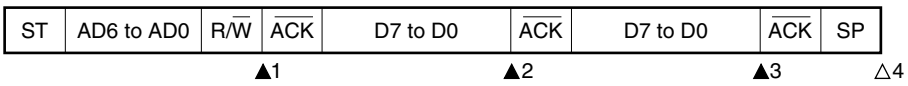
Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(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

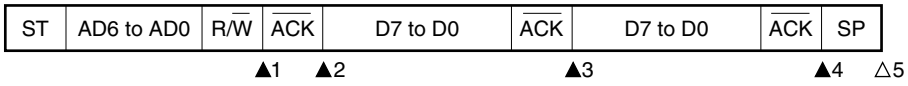
(i) When WTIM = 0



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

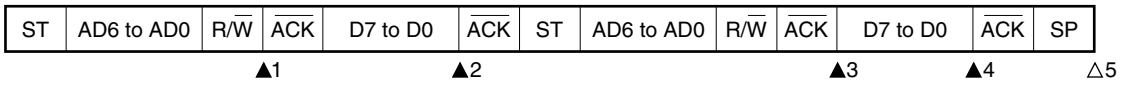


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010×100B
▲4: IICS = 0010××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

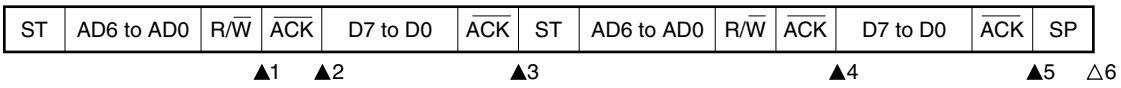
(i) When WTIM = 0 (after restart, matches SVA)



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0001×110B
▲4: IICS = 0001×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, matches SVA)

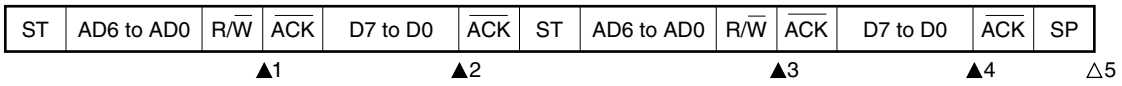


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010××00B
▲4: IICS = 0001×110B
▲5: IICS = 0001××00B
△6: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

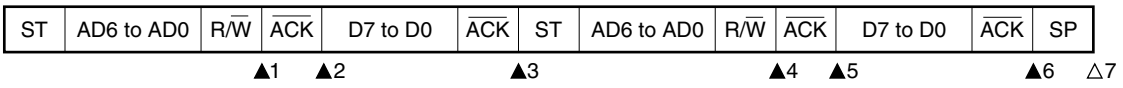
(i) When WTIM = 0 (after restart, extension code reception)



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×010B
▲4: IICS = 0010×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, extension code reception)

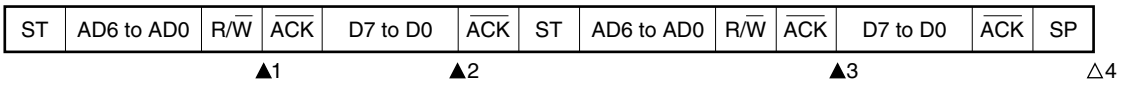


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010××00B
▲4: IICS = 0010×010B
▲5: IICS = 0010×110B
▲6: IICS = 0010××00B
△7: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

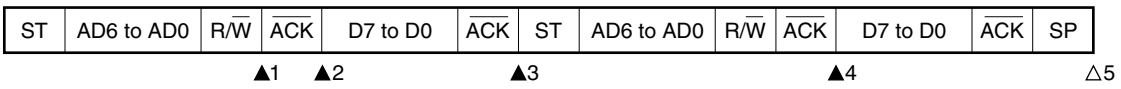
(i) When WTIM = 0 (after restart, does not match address (= not extension code))



▲1: IICS = 00100010B
▲2: IICS = 00100000B
▲3: IICS = 00000110B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= not extension code))

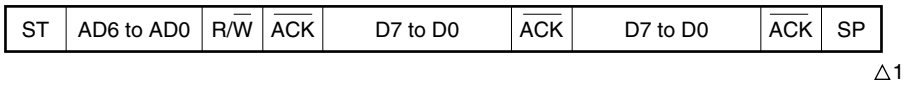


▲1: IICS = 00100010B
▲2: IICS = 00100110B
▲3: IICS = 00100×00B
▲4: IICS = 00000110B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICS = 00000001B

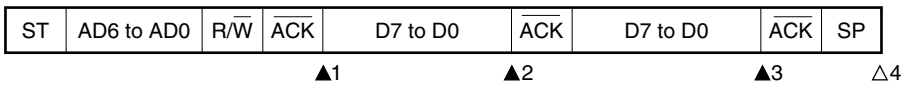
Remark △: Generated only when SPIE = 1

(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 MSTS bit each time interrupt request signal INTIICA has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

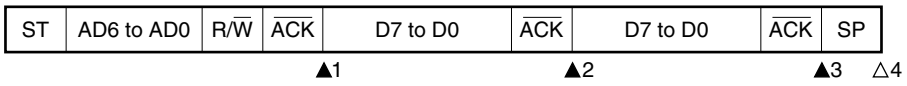
(i) When WTIM = 0



▲1: IICS = 0101×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



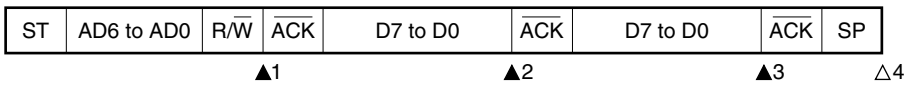
▲1 ▲2 ▲3 △4

▲1: IICS = 0101×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIM = 0

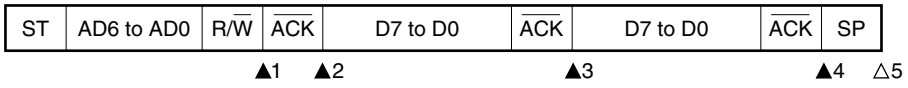


▲1 ▲2 ▲3 △4

▲1: IICS = 0110×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



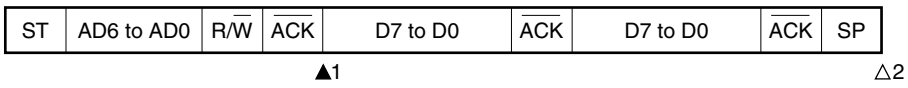
▲1: IICS = 0110×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010×100B
▲4: IICS = 0010××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 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 MSTS bit each time interrupt request signal INTIICA has occurred to check the arbitration result.

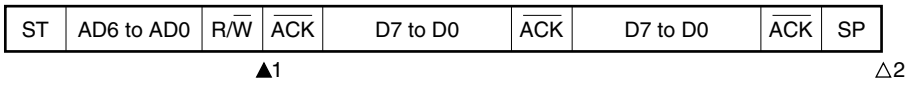
(a) When arbitration loss occurs during transmission of slave address data (when WTIM = 1)



▲1: IICS = 01000110B
△2: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(b) When arbitration loss occurs during transmission of extension code



▲1: IICS = 0110×010B

Sets LREL = 1 by software

△2: IICS = 00000001B

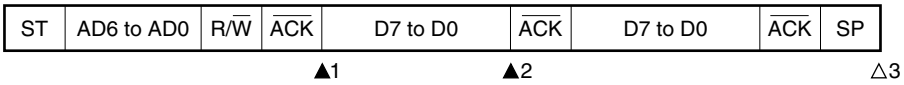
Remark ▲: Always generated

△: Generated only when SPIE = 1

×: Don't care

(c) When arbitration loss occurs during transmission of data

(i) When WTIM = 0



▲1: IICS = 10001110B

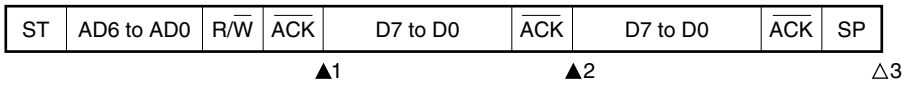
▲2: IICS = 01000000B

△3: IICS = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE = 1

(ii) When WTIM = 1

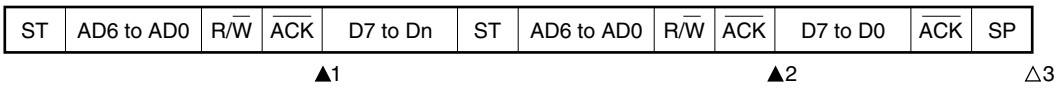


▲1: IICS = 10001110B
▲2: IICS = 01000100B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(d) When loss occurs due to restart condition during data transfer

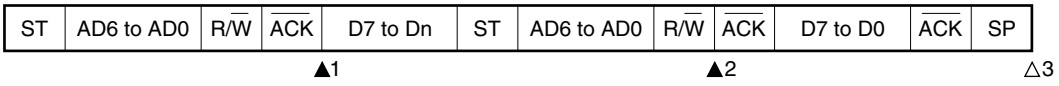
(i) Not extension code (Example: unmatches with SVA)



▲1: IICS = 1000×110B
▲2: IICS = 01000110B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care
n = 6 to 0

(ii) Extension code



▲1: IICS = 1000×110B

▲2: IICS = 01100010B

Sets LREL = 1 by software

△3: IICS = 00000001B

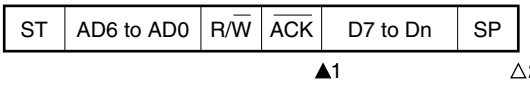
Remark ▲: Always generated

△: Generated only when SPIE = 1

×: Don't care

n = 6 to 0

(e) When loss occurs due to stop condition during data transfer



▲1: IICS = 10000110B

△2: IICS = 01000001B

Remark ▲: Always generated

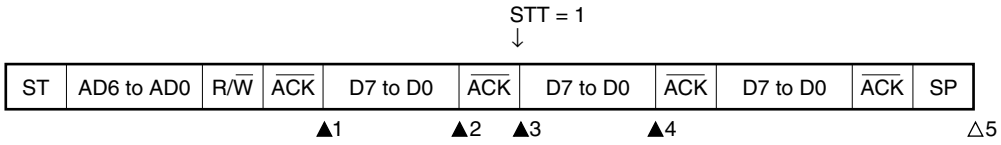
△: Generated only when SPIE = 1

×: Don't care

n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

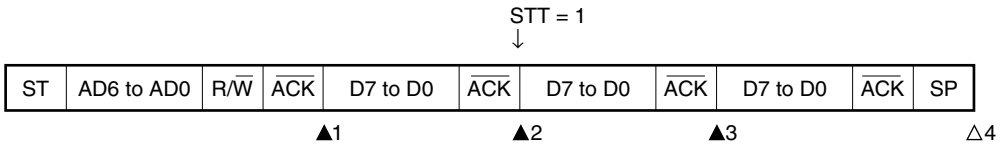
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM bit to 1)
▲3: IICS = 1000×100B (Clears WTIM bit to 0)
▲4: IICS = 01000000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

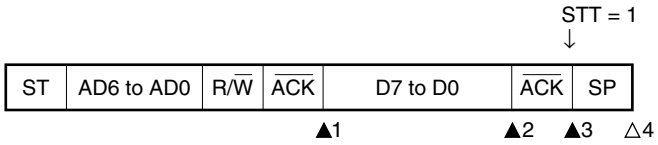


▲1: IICS = 1000×110B
▲2: IICS = 1000×100B (Sets STT bit to 1)
▲3: IICS = 01000100B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

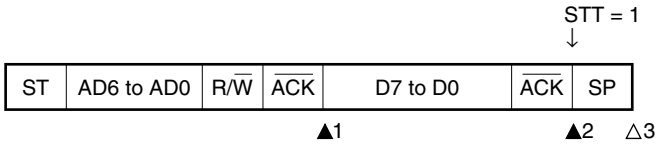
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM bit to 1)
▲3: IICS = 1000××00B (Sets STT bit to 1)
△4: IICS = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

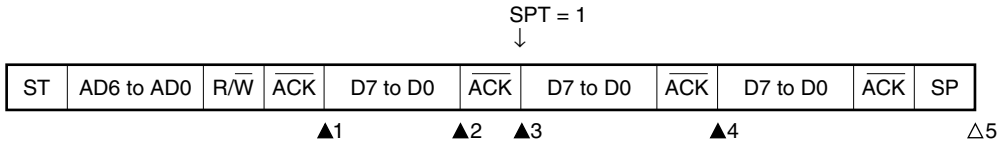


▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (Sets STT bit to 1)
△3: IICS = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

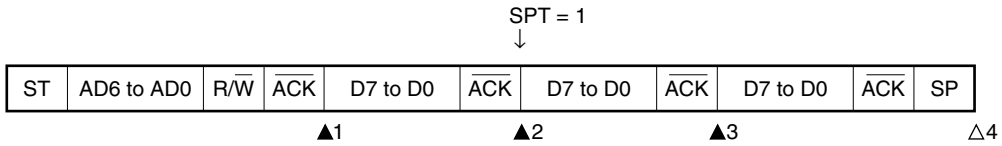
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM bit to 1)
▲3: IICS = 1000×100B (Clears WTIM bit to 0)
▲4: IICS = 01000100B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



▲1: IICS = 1000×110B
▲2: IICS = 1000×100B (Sets SPT bit to 1)
▲3: IICS = 01000100B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

12.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC bit (bit 3 of the IICA status register (IICS)), which specifies the data transfer direction, and then starts serial communication with the slave device.

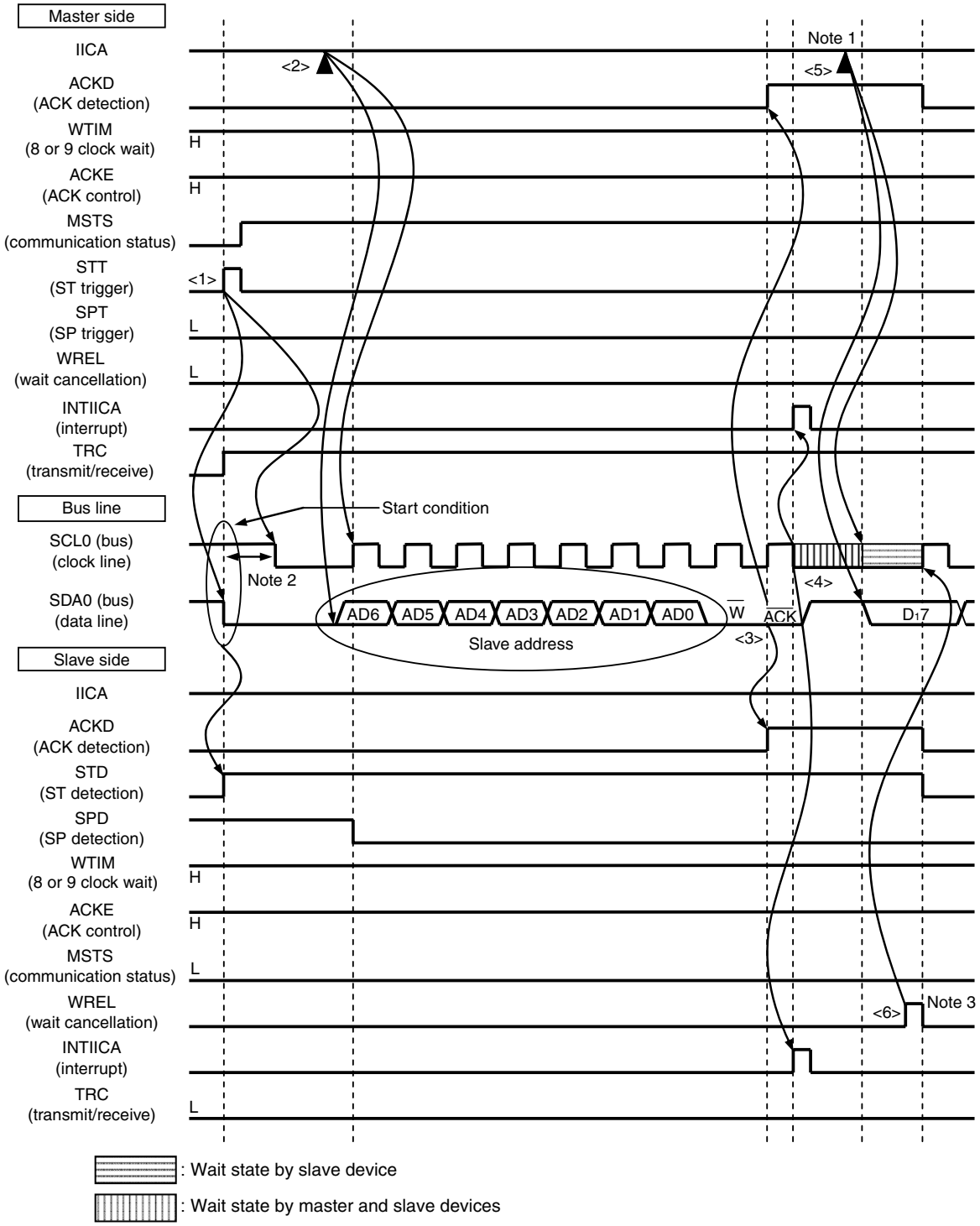
Figures 12-32 and 12-33 show timing charts of the data communication.

The IICA shift register (IICA)'s shift operation is synchronized with the falling edge of the serial clock (SCL0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA0 pin.

Data input via the SDA0 pin is captured into IICA at the rising edge of SCL0.

Figure 12-32. 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 IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. Make sure that the time between the fall of the SDA0 pin signal and the fall of the SCL0 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 slave wait, write "FFH" to IICA or set the WREL bit.

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 12-32 are explained below.

<1> The start condition trigger is set by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus data line goes low (SDA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS = 1). The master device is ready to communicate once the bus clock line goes low (SCL0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register (IICA) 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 (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: 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 (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA register and releases the wait status that it set by the master device.

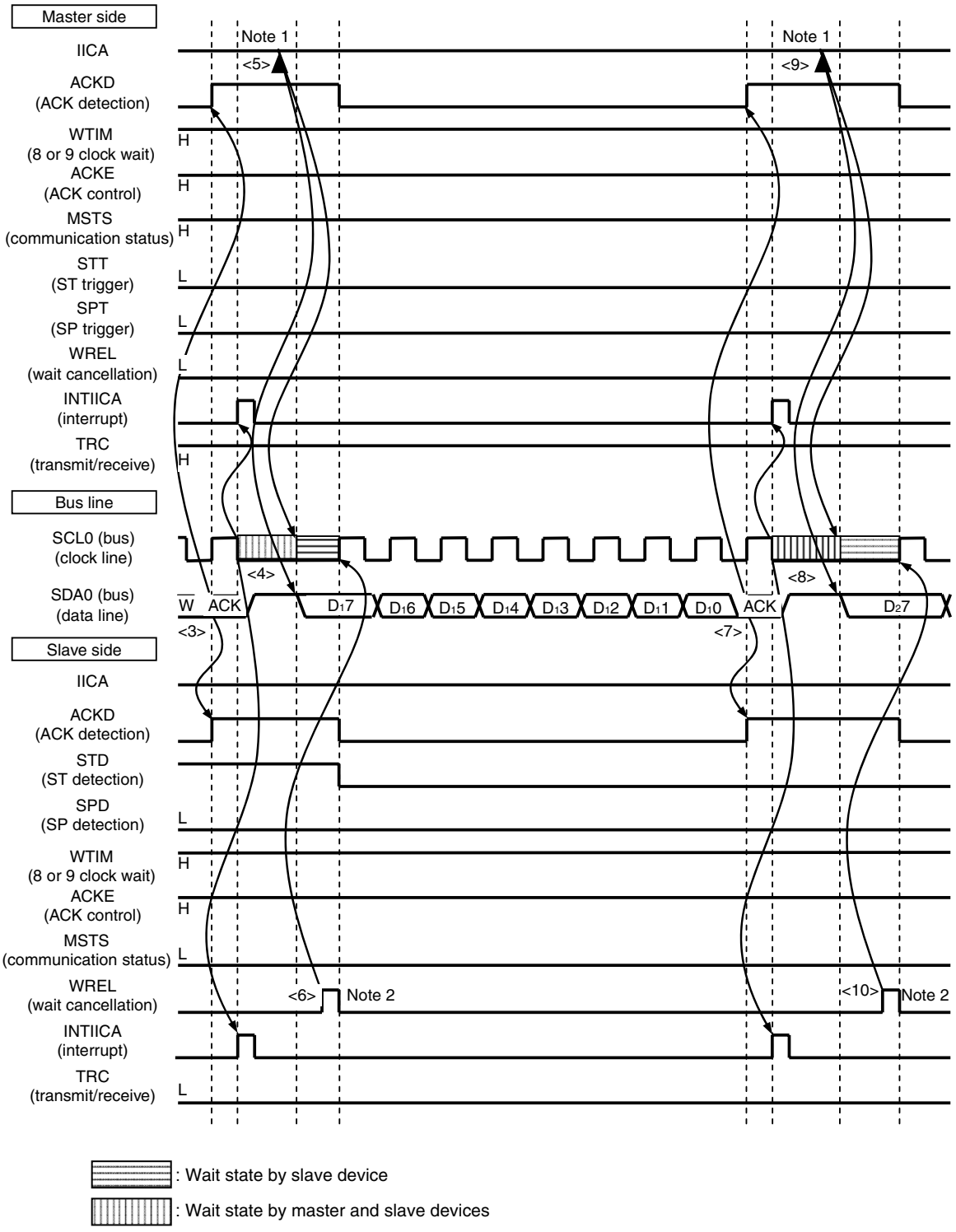
<6> If the slave device releases the wait status (WREL = 1), the master device starts transferring data to the slave device.

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: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 12-32 represent the entire procedure for communicating data using the I^2C bus. Figure 12-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 12-32. 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 IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 12-32 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 (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: 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 (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WREL = 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 (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<9> The master device writes the data to transmit to the IICA register and releases the wait status that it set by the master device.

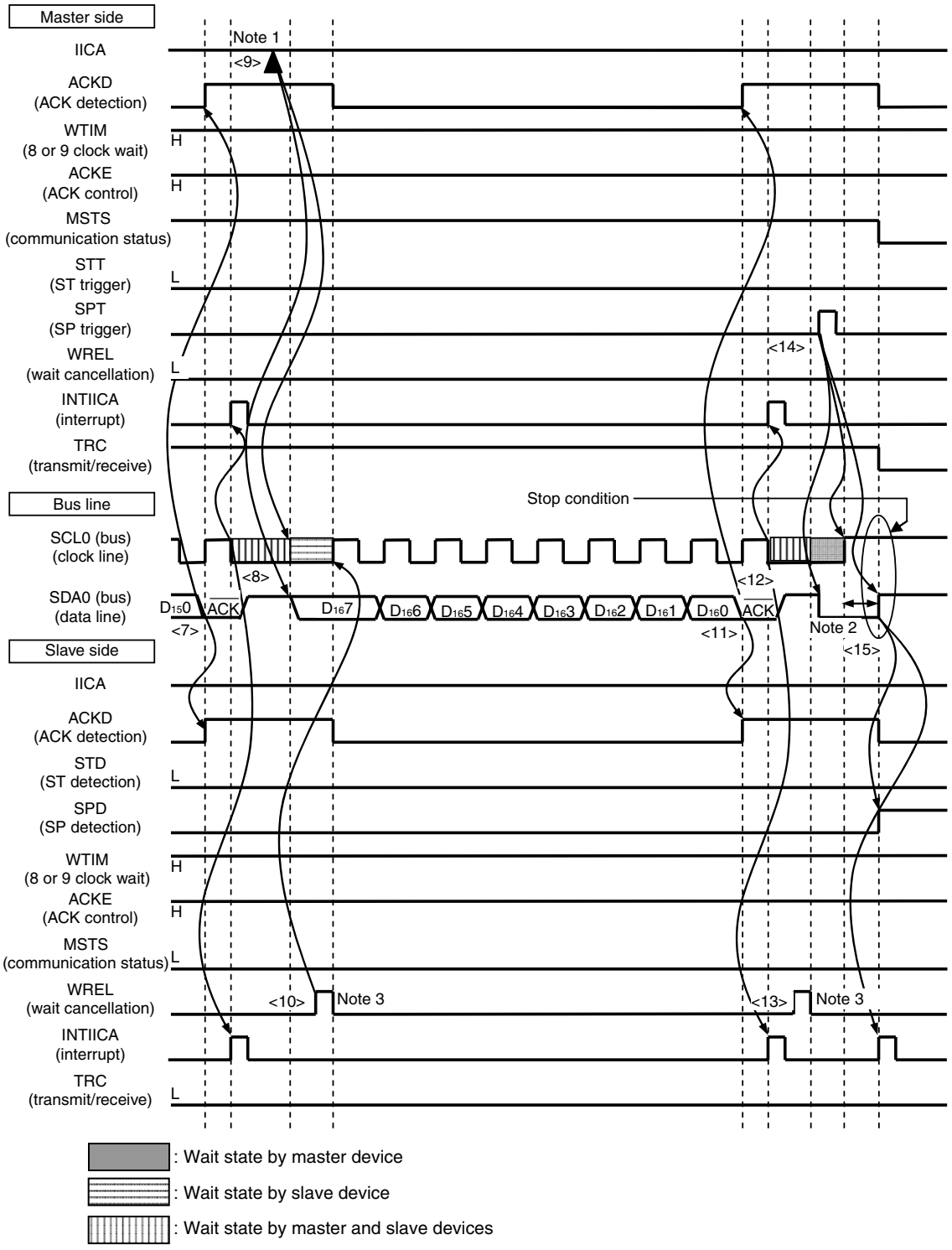
<10> The slave device reads the received data and releases the wait status (WREL = 1). The master device then starts transferring data to the slave device.

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: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 12-32 represent the entire procedure for communicating data using the I^2C bus. Figure 12-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 12-32. 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 IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. Make sure that the time between the rise of the SCL0 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 slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 12-32 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 (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register (IICA) 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 (WREL = 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 (ACKD = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<13> The slave device reads the received data and releases the wait status (WREL = 1).

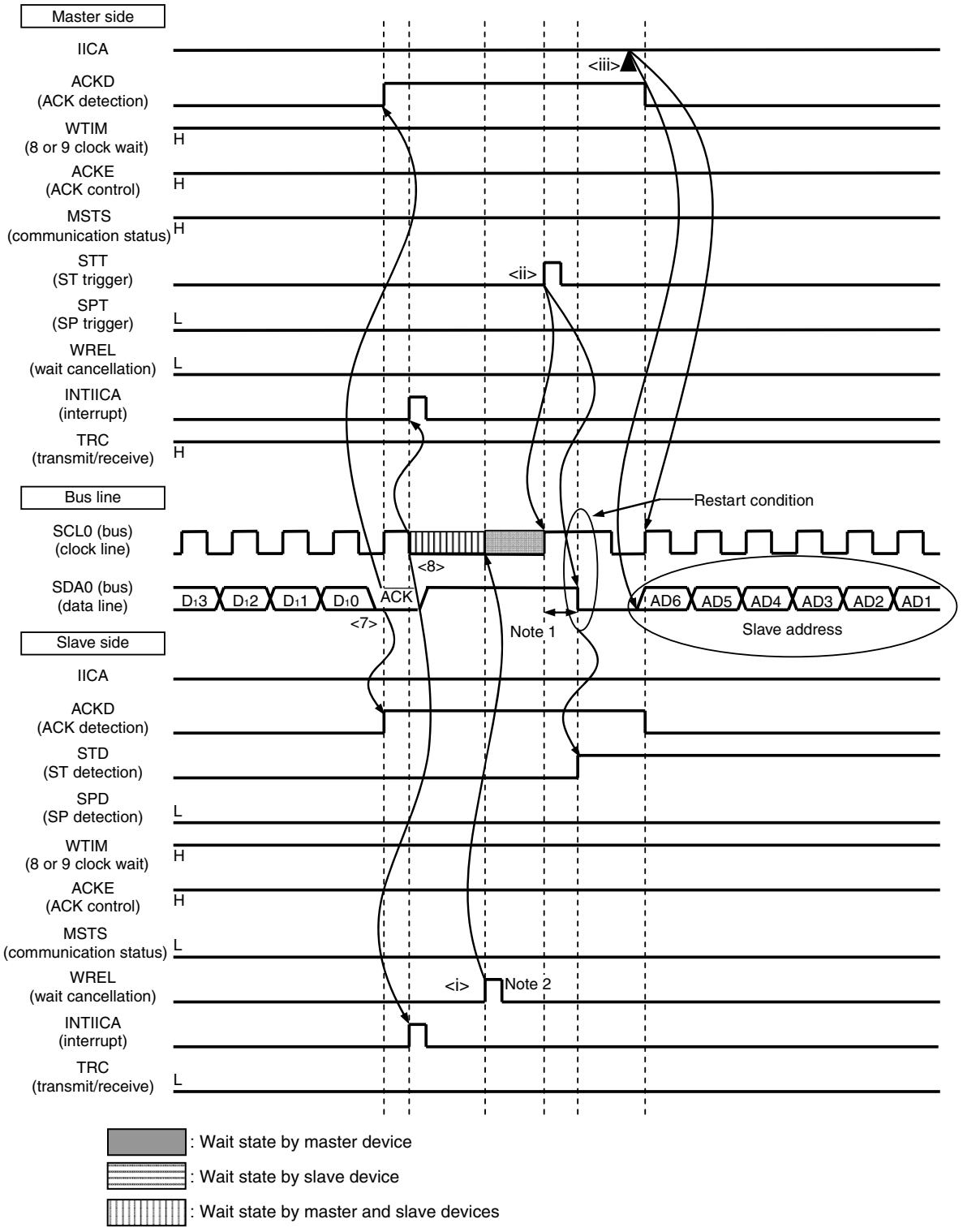
<14> After a stop condition trigger is set, the bus data line is cleared (SDA0 = 0) and the bus clock line is set (SCL0 = 1). The stop condition is then generated by setting the bus data line (SDA0 = 1) after the stop condition setup time has elapsed.

<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICA: stop condition).

Remark <1> to <15> in Figure 12-32 represent the entire procedure for communicating data using the I^2C bus. Figure 12-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 12-32. 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 SCL0 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 slave wait, write “FFH” to IICA or set the WREL bit.

The following describes the operations in Figure 12-32 (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 (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

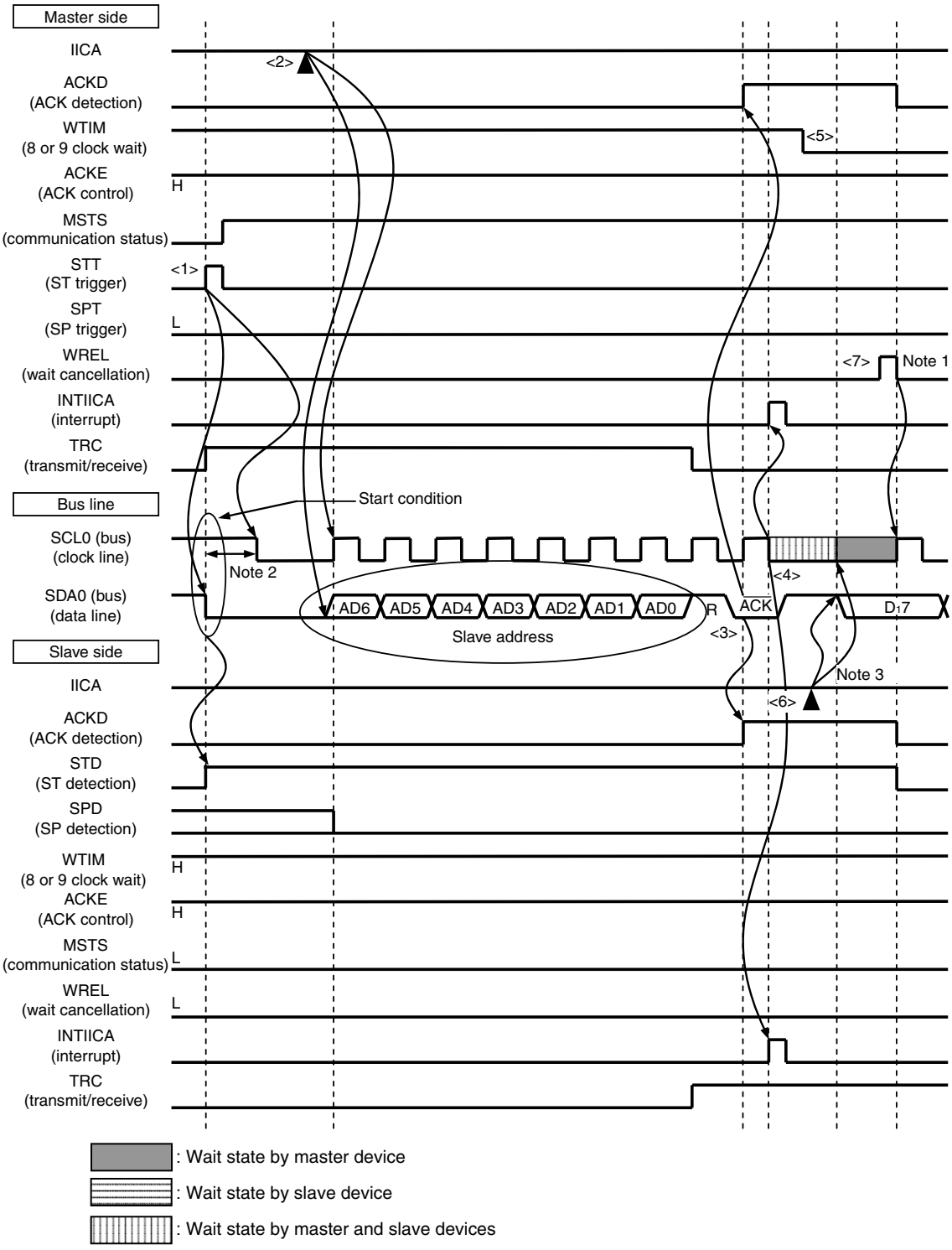
<i> The slave device reads the received data and releases the wait status (WREL = 1).

<ii> The start condition trigger is set again by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus clock line goes high (SCL0 = 1) and the bus data line goes low (SDA0 = 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 (SCL0 = 0) after the hold time has elapsed.

<iii> The master device writes the address + R/W (transmission) to the IICA shift register (IICA) and transmits the slave address.

Figure 12-33. 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 master wait, write “FFH” to IICA or set the WREL bit.

2. Make sure that the time between the fall of the SDA0 pin signal and the fall of the SCL0 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 IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 12-33 are explained below.

<1> The start condition trigger is set by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus data line goes low (SDA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS = 1). The master device is ready to communicate once the bus clock line goes low (SCL0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register (IICA) 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 (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: 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 (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM = 0).

<6> The slave device writes the data to transmit to the IICA register and releases the wait status that it set by the slave device.

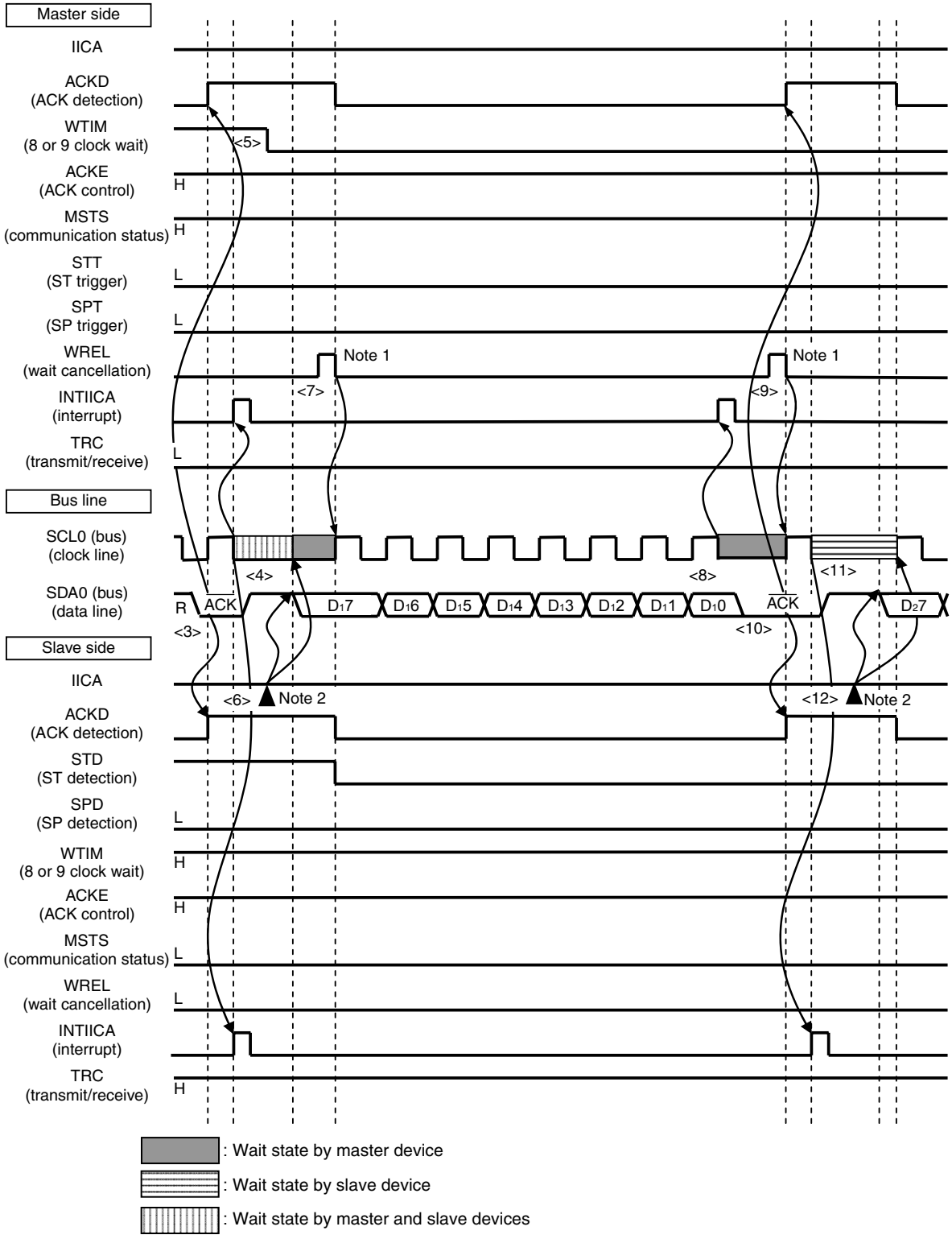
<7> If the master device releases the wait status (WREL = 1), the slave device starts transferring data to the master device.

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: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 12-33 represent the entire procedure for communicating data using the I^2C bus. Figure 12-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 12-33. 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 master wait, write “FFH” to IICA or set the WREL bit.

2. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 12-33 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 (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: 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 (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM = 0).

<6> The slave device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the slave device.

<7> If the master device releases the wait status (WREL = 1), the slave device starts transferring data to the master device.

<8> The master device sets a wait status (SCL0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA: 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 (WREL = 1).

<10> The ACK is detected by the slave device (ACKD = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA: end of transfer).

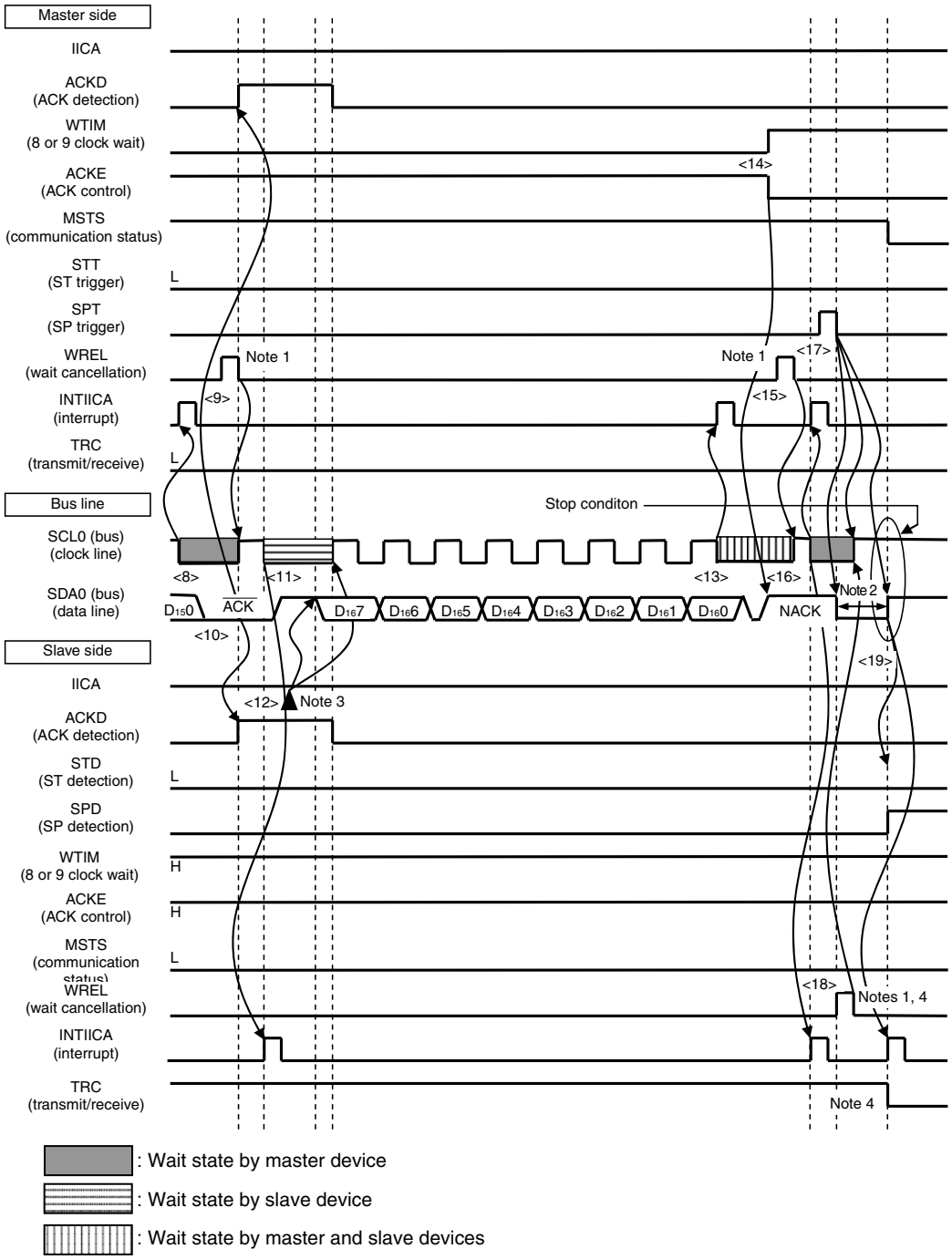
<12> The slave device writes the data to transmit to the IICA register and releases the wait status that it set by the slave device. The slave device then starts transferring data to the master device.

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: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 12-33 represent the entire procedure for communicating data using the I^2C bus. Figure 12-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 12-33. 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



Notes 1. To cancel a wait state, write “FFH” to IICA or set the WREL bit.

2. Make sure that the time between the rise of the SCL0 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 IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

4. If a wait state during slave transmission is canceled by setting the WREL bit, the TRC bit will be cleared.

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 12-33 are explained below.

<8> The master device sets a wait status (SCL0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA: 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 (WREL = 1).

<10> The ACK is detected by the slave device (ACKD = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA: end of transfer).

<12> The slave device writes the data to transmit to the IICA shift register (IICA) 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 (INTIICA: end of transfer) at the falling edge of the 8th clock, and sets a wait status (SCL0 = 0). Because ACK control (ACKE = 1) is performed, the bus data line is at the low level (SDA0 = 0) at this stage.

<14> The master device sets NACK as the response (ACKE = 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 (WREL = 1), the slave device detects the NACK (ACK = 0) at the rising edge of the 9th clock.

<16> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<17> When the master device issues a stop condition (SPT = 1), the bus data line is cleared (SDA0 = 0) and the master device releases the wait status. The master device then waits until the bus clock line is set (SCL0 = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the wait status (WREL = 1) to end communication. Once the slave device releases the wait status, the bus clock line is set (SCL0 = 1).

<19> Once the master device recognizes that the bus clock line is set (SCL0 = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDA0 = 1) and issues a stop condition. The slave device detects the generated stop condition and both the master device and slave device issue an interrupt (INTIICA: stop condition).

Remark <1> to <19> in Figure 12-33 represent the entire procedure for communicating data using the I^2C bus. Figure 12-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

CHAPTER 13 MULTIPLIER/DIVIDER

13.1 Functions of Multiplier/Divider

The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)

13.2 Configuration of Multiplier/Divider

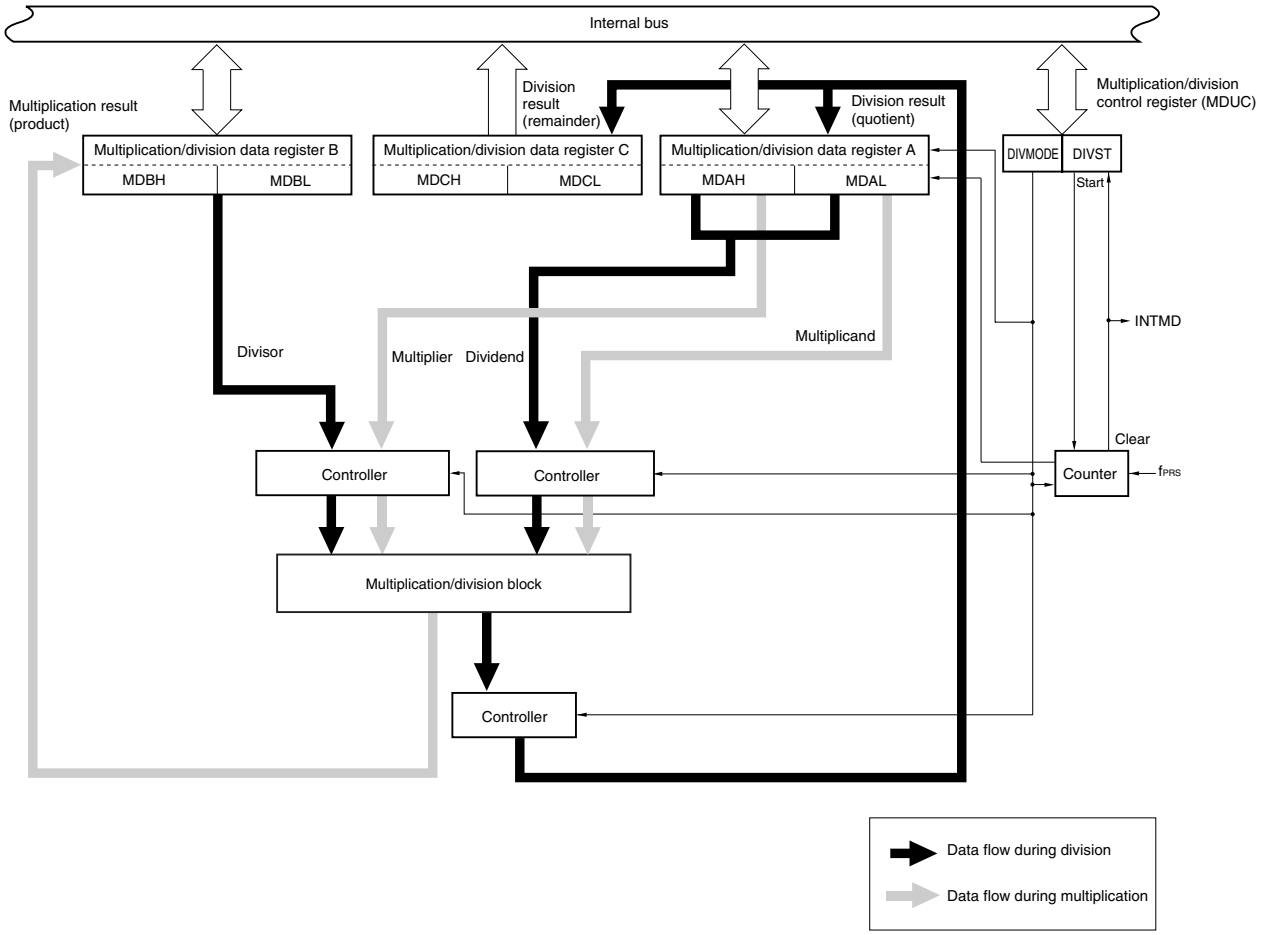
The multiplier/divider consists of the following hardware.

Table 13-1. Configuration of Multiplier/Divider

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 13-1 shows a block diagram of the multiplier/divider.

Figure 13-1. Block Diagram of Multiplier/Divider



(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, 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.

MDAH and MDAL can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 13-2. 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 values during division operation processing (while the multiplication/division control register (MDUC) is 81H). The operation will be executed in this case, but the operation result will be an undefined value.

2. The MDAH and MDAL values read during division operation processing (while MDUC is 81H) will not be guaranteed.

The following table shows the functions of MDAH and MDAL during operation execution.

Table 13-2. Functions of MDAH and MDAL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	MDAH: Multiplier MDAL: Multiplicand	–
1	Division mode	MDAH: Divisor (higher 16 bits) MDAL: Dividend (lower 16 bits)	MDAH: Division result (quotient) Higher 16 bits MDAL: Division result (quotient) Lower 16 bits

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

(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 set the divisor data in the division mode.

MDBH and MDBL can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 13-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol	FFFF7H								FFFF6H							
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	FFFF5H								FFFF4H							
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	MDBHL 3	MDBL 2	MDBL 1	MDBL 0

Cautions 1. Do not rewrite the MDBH and MDBL values during division operation processing (while the multiplication/division control register (MDUC) is 81H). The operation result will be an undefined value.

2. Do not set MDBH and MDBL to 0000H in the division mode. If they are set, the operation result will be an undefined value.

The following table shows the functions of MDBH and MDBL during operation execution.

Table 13-3. Functions of MDBH and MDBL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	–	MDBH: Multiplication result (product) Higher 16 bits MDBL: Multiplication result (product) Lower 16 bits
1	Division mode	MDBH: Divisor (higher 16 bits) MDBL: Dividend (lower 16 bits)	–

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

(3) Multiplication/division data register C (MDCL, MDCH)

The MDCH and MDCL registers store remainder value of the operation result in the division mode. They are not used in the multiplication mode.

MDCH and MDCL can be read by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 13-4. Format of Multiplication/Division Data Register C (MDCH, MDCL)

Address: F00E0H, F00E1H, F00E2H, F00E3H After reset: 0000H, 0000H R

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

Caution The MDCH and MDCL values read during division operation processing (while the multiplication/division control register (MDUC) is 81H) will not be guaranteed.

Table 13-4. Functions of MDCH and MDCL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	–	–
1	Division mode	–	MDCH: Remainder (higher 16 bits) MDCL: Remainder (lower 16 bits)

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

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 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)]

13.3 Register Controlling Multiplier/Divider

The multiplier/divider is controlled by using the multiplication/division control register (MDUC).

(1) Multiplication/division control register (MDUC)

MDUC is an 8-bit register that controls the operation of the multiplier/divider.

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

Reset signal generation clears this register to 00H.

Figure 13-5. 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	0	0	0	0	0	0	DIVST

DIVMODE	Operation mode (multiplication/division) selection
0	Multiplication mode
1	Division mode

DIVST[Note]	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

Note DIVST can only be set (1) in the division mode. In the division mode, division operation is started by setting (1) DIVST. DIVST is automatically cleared (0) when the operation ends. In the multiplication mode, operation is automatically started by setting the multiplier and multiplicand to MDAH and MDAL, respectively.

Cautions 1. Do not rewrite DIVMODE during operation processing (while DIVST is 1). If it is rewritten, the operation result will be an undefined value.

2. DIVST cannot be cleared (0) by using software during division operation processing (while DIVST is 1).

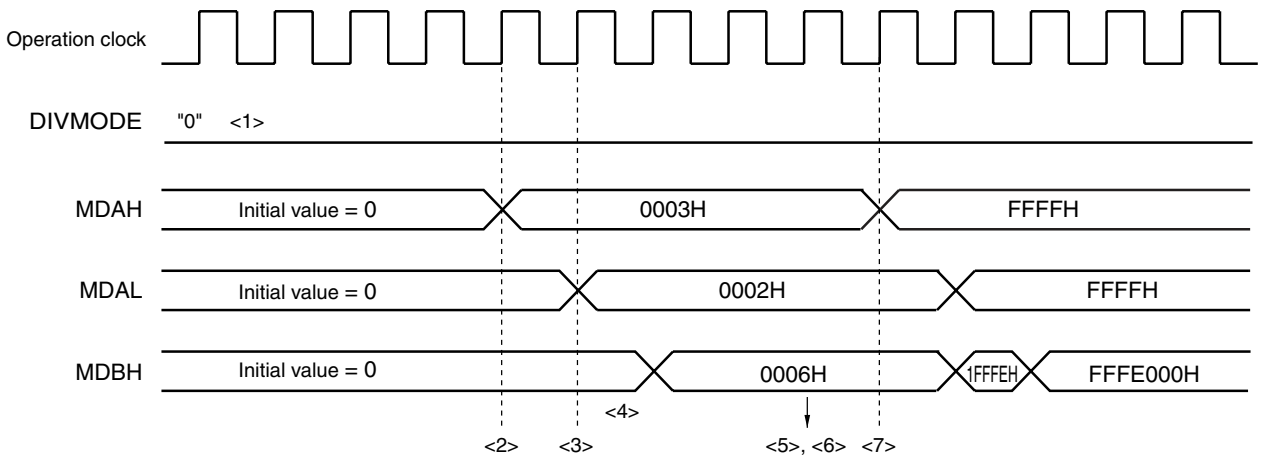
13.4 Operations of Multiplier/Divider

13.4.1 Multiplication operation

- Initial setting
  - <1> Set bit 7 (DIVMODE) of the multiplication/division control register (MDUC) to 0.
  - <2> Set the multiplicand to the multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to the 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 MDAH and MDAL, 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 the multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from the 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 operation next, start from the “Initial setting” for multiplication operation.
  - <8> To execute division operation next, start from the “Initial setting” in **13.4.2 Division operation**.

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 13-6.

Figure 13-6. Timing Diagram of Multiplication Operation (0003H × 0002H)



13.4.2 Division operation

- Initial setting
  - <1> Set bit 7 (DIVMODE) of the multiplication/division control register (MDUC) to 1.
  - <2> Set the dividend (higher 16 bits) to the multiplication/division data register A (H) (MDAH).
  - <3> Set the dividend (lower 16 bits) to the multiplication/division data register A (L) (MDAL).
  - <4> Set the divisor (higher 16 bits) to the multiplication/division data register B (H) (MDBH).
  - <5> Set the divisor (lower 16 bits) to the multiplication/division data register B (L) (MDBL).
  - <6> Set bit 0 (DIVST) of MDUC to 1.

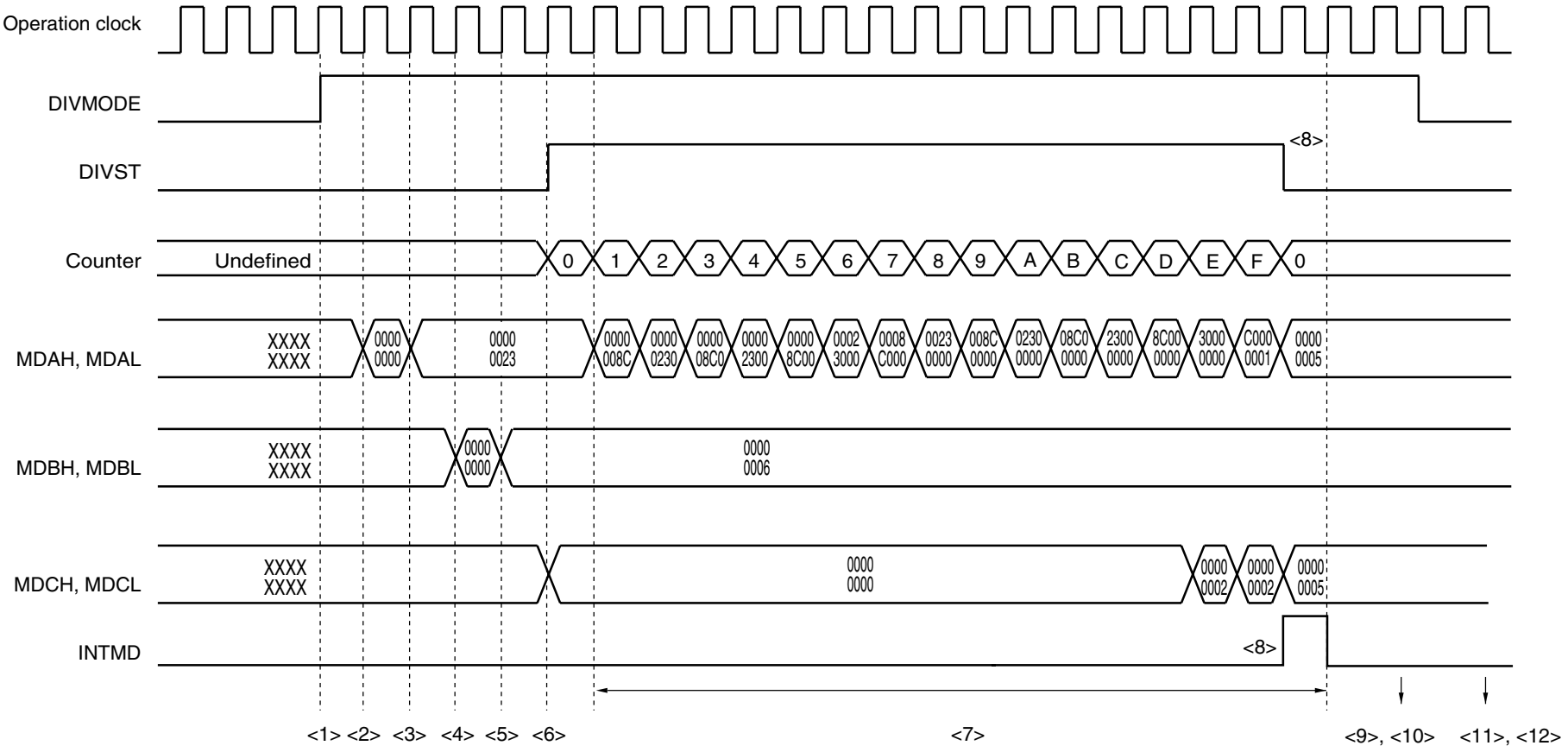
    (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  - <7> The operation will end when one of the following processing is completed.
    - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
    - A check whether DIVST has been cleared
    - Generation of a division completion interrupt (INTMD)

    (The read values of MDBL, MDBH, MDCH, and MDCL during operation processing are not guaranteed.)
- Operation end
  - <8> DIVST is cleared (0) and an interrupt request signal (INTMD) is generated (end of operation).
  - <9> Read the quotient (lower 16 bits) from MDAL.
  - <10> Read the quotient (higher 16 bits) from MDAH.
  - <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  - <12> Read the remainder (higher 16 bits) from the 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 operation next, start from the “Initial setting” in **13.4.1 Multiplication operation**.
  - <14> To execute division operation next, start from the “Initial setting” for division operation.

Remark Steps <1> to <12> correspond to <1> to <12> in Figure 13-7.

Figure 13-7. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



CHAPTER 14 DMA CONTROLLER

The μPD78F8040, 78F8041, 78F8042, 78F8043 have an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between SFRs of the peripheral hardware supporting DMA 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.

14.1 Functions of DMA Controller

- Number of DMA channels: 2
- 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 (UART0, or UART3)
  - Timer (channel 0, 1, 4, or 5)
- 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

14.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 14-1. 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 registers 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.

DSAn 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 14-1. 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 (FE700H to FFEDFH in the case of the μPD78F8042) 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, DRAn 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.

DRAn 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 14-2. Format of DMA RAM Address Register n (DRAn)

Address: FFFB2H, FFFB3H (DRA0), FFFB4H, FFFB5H (DRA1) After reset: 0000H R/W

DRA0H: FFFB3H
DRA1H: FFFB5H

DRA0L: FFFB2H
DRA1L: FFFB4H

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

(n = 0, 1)

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.

DBCn 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 14-3. 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)

14.3 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)

DMCn 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 DMCn is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 14-4. 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 STGn 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 DWAITn to 0. It takes 2 clocks to actually hold DMA transfer pending when the value of DWAITn is set to 1.	

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

2. When DMA transfer is held pending while using both DMA channels, be sure to hold the DMA transfer pending for both channels (by setting the DWAIT0 and DWAIT1 bits to 1).

Remark n: DMA channel number (n = 0, 1)

Figure 14-4. 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	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	INTTM04	End of timer channel 4 count or capture end interrupt
0	1	0	1	INTTM05	End of timer channel 5 count or capture end interrupt
0	1	1	0	INTST0	UART0 transmission transfer end or buffer empty interrupt
0	1	1	1	INTSR0	UART0 reception transfer end interrupt
1	0	1	0	INTST3	UART3 transmission transfer end or buffer empty interrupt
1	0	1	1	INTSR3	UART3 reception transfer end interrupt
1	1	0	0	INTAD	A/D conversion end interrupt
Other than above				Setting prohibited	

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

Remark n: DMA channel number (n = 0, 1)

(2) DMA operation control register n (DRCn)

DRCn 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).

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

Reset signal generation clears this register to 00H.

Figure 14-5. 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.
The DMA controller 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).
The DMA controller 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 IFCn3 to IFCn0 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.	

Cautions 1. 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 DSTn to 0 and then DENn to 0 (for details, refer to 14.5.4 Forced termination by software).

2. When the FSEL bit of the OSMC register has been set to 1, do not enable (DENn = 1) DMA operation for at least three clocks after the setting.

Remark n: DMA channel number (n = 0, 1)

14.4 Operation of DMA Controller

14.4.1 Operation procedure

<1> The DMA controller is enabled to operate when DENn = 1. Before writing the other registers, be sure to set DENn 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 the DSAn, DRAn, DBCn, and DMCn registers.

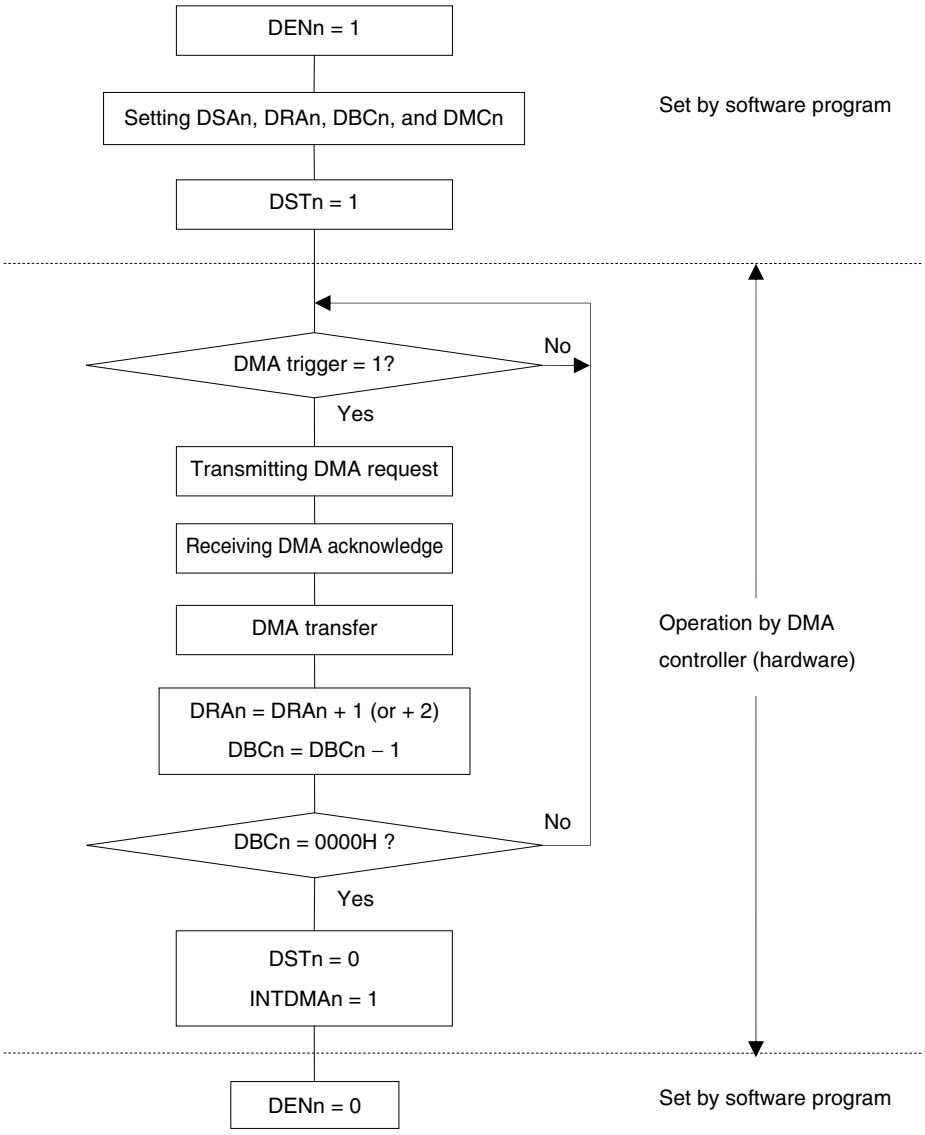
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.

<4> When a software trigger (STGn) or a start source trigger specified by IFCn3 to IFCn0 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 DENn to 0 when the DMA controller is not used.

Figure 14-6. Operation Procedure



Remark n: DMA channel number (n = 0, 1)

14.4.2 Transfer mode

The following four modes can be selected for DMA transfer by using bits 6 and 5 (DRSn and DSn) of the DMCn register.

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.

14.4.3 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, the DBCn and DRAn registers 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)

14.5 Example of Setting of DMA Controller

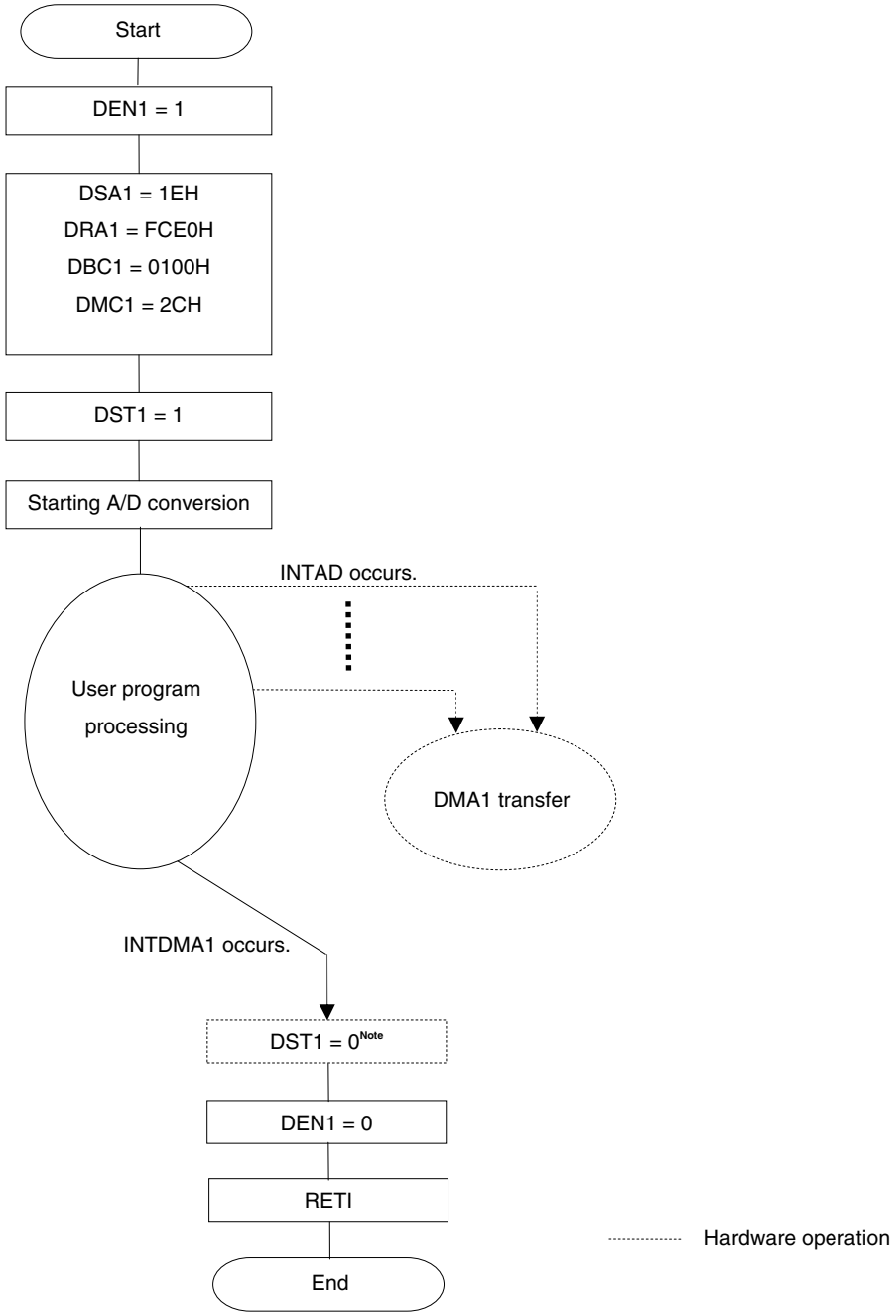
14.5.1 Consecutive capturing of A/D conversion results

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 = 1100B.
- Transfers FFF1EH and FFF1FH (2 bytes) of the 10-bit A/D conversion result register 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 14-7. Example of Setting of Consecutively Capturing A/D Conversion Results



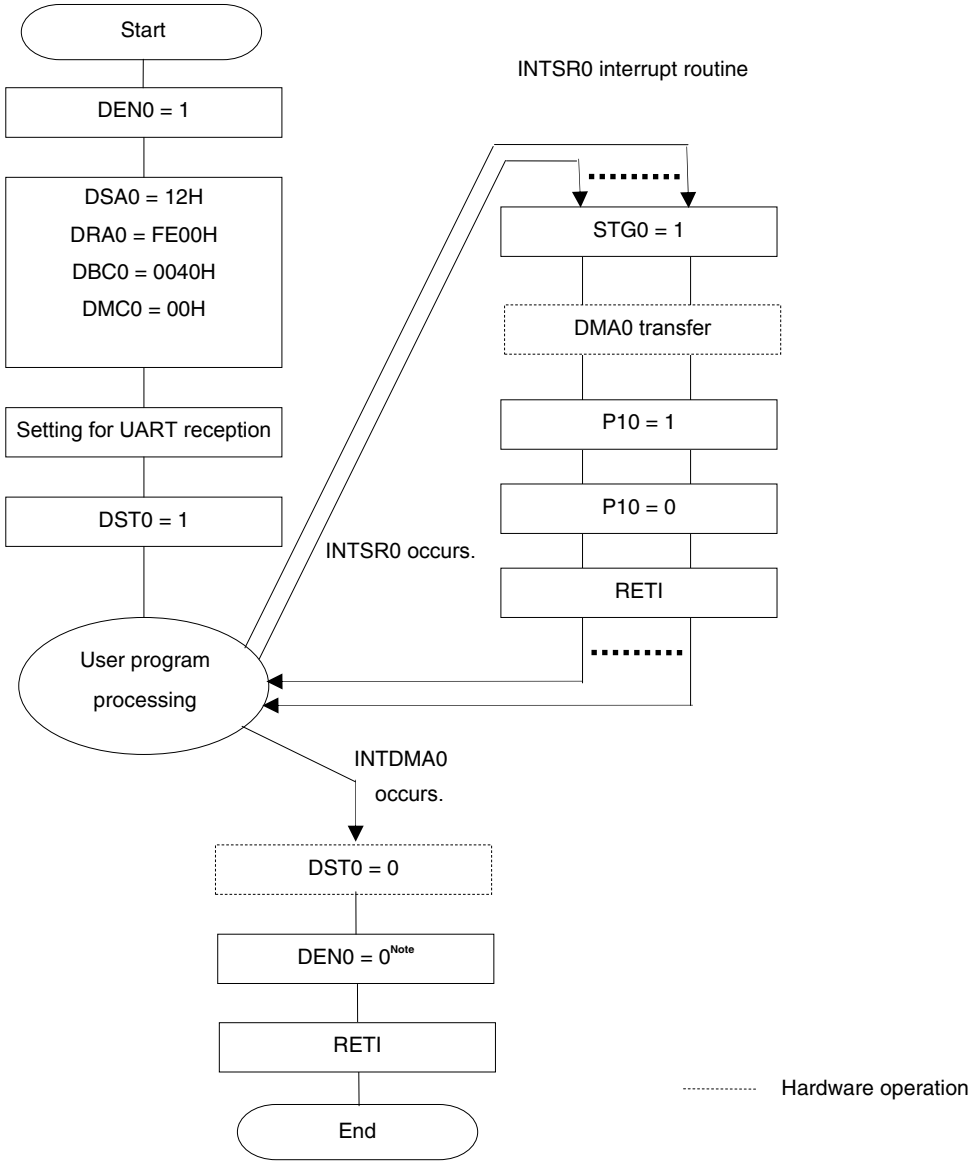
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set DST1 to 0 and then DEN1 to 0 (for details, refer to **14.5.4 Forced termination by software**).

14.5.2 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 14-8. 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 DST0 to 0 and then DEN0 to 0 (for details, refer to **14.5.4 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.

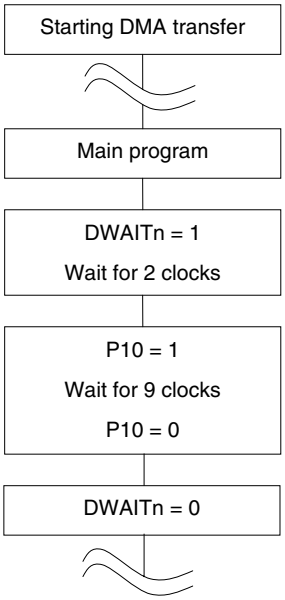
14.5.3 Holding DMA transfer pending by DWAITn

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 DWAITn 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 DWAITn to 1.

After setting DWAITn to 1, it takes two clocks until a DMA transfer is held pending.

Figure 14-9. Example of Setting for Holding DMA Transfer Pending by DWAITn



Caution When DMA transfer is held pending while using both DMA channels, be sure to held the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 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)

14.5.4 Forced termination by software

After DSTn is set to 0 by software, it takes up to 2 clocks until a DMA transfer is actually stopped and DSTn 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>

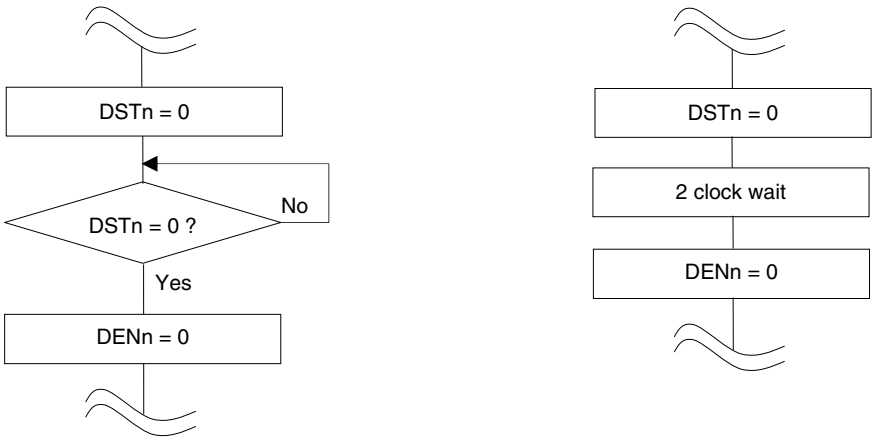
- Set DSTn to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that DSTn has actually been cleared to 0, and then set DENn to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- Set DSTn to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set DENn to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

<When using both DMA channels>

- To terminate a DMA transfer by using software when two channels are used, set the DWAITn bit corresponding to each channel to hold the DMA transfer pending, clear the DSTn0 bit, clear the DWAITn bit for both channels to cancel the pending state, and then clear the DENn bit for both channels.

Figure 14-10. Forced Termination of DMA Transfer (1/2)



Example 1 Example 2

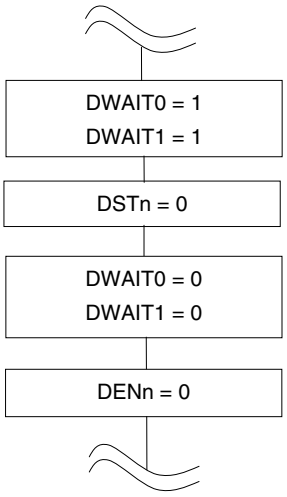
Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

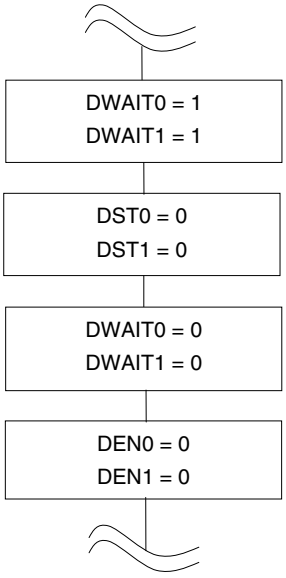
Figure 14-10. Forced Termination of DMA Transfer (2/2)

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 DWAITn is set to 1. In addition, the system does not have to wait two clock cycles after clearing DSTn to 0, because more than two clock cycles elapse from when DSTn is cleared to 0 to when DENn is cleared to 0.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

14.6 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 DMA requests are generated at the same time, however, DMA channel 0 takes priority over DMA channel 1.

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 14-2. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time[Note]	3 clocks	10 clocks

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 14.6 (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 14-3. 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.

(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

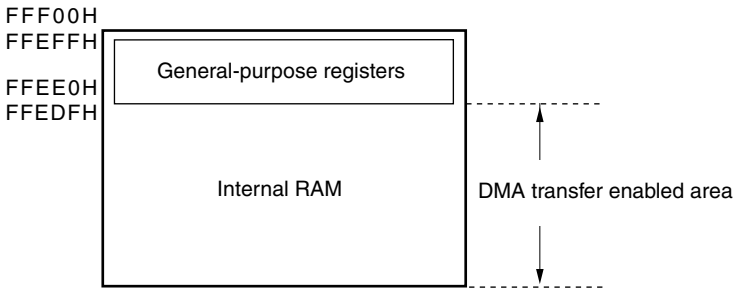
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

The address indicated by DRA0n 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.



CHAPTER 15 INTERRUPT FUNCTIONS

15.1 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 15-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

External: 5, internal: 28

(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.

15.2 Interrupt Sources and Configuration

The μPD78F8040, 78F8041, 78F8042, 78F8043 have a total of 34 interrupt sources including maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Table 15-1**). 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 15-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	0	INTWDTI	Watchdog timer interval[Note 3] (75% of overflow time)	Internal	0004H	(A)
	1	INTLVI	Low-voltage detection[Note 4]		0006H	
	2	INTP0	Pin input edge detection	External	0008H	(B)
	3	INTP1	Oercurrents detection of IO-Link[Note 5]		000AH	
	4	INTP2	Wakeup signal detection of IO-Link[Note 5]		000CH	
	5	INTP4	Pin input edge detection		0010H	
	6	INTP5			0012H	
	7	INTST3	UART3 transmission transfer end or buffer empty interrupt	Internal	0014H	(A)
	8	INTSR3	UART3 reception transfer end		0016H	
	9	INTSRE3	UART3 reception communication error occurrence		0018H	
	10	INTDMA0	End of DMA0 transfer		001AH	
	11	INTDMA1	End of DMA1 transfer		001CH	
	12	INTST0	UART0 transmission transfer end or buffer empty interrupt		001EH	
	13	INTSR0	UART0 reception transfer end		0020H	
	14	INTSRE0	UART0 reception communication error occurrence		0022H	
	15	INTIICA	End of IICA communication		002AH	
	16	INTTM00	End of timer array unit 0 channel 0 count or capture		002CH	
	17	INTTM01	End of timer array unit 0 channel 1 count or capture		002EH	
	18	INTTM02	End of timer array unit 0 channel 2 count or capture		0030H	
	19	INTTM03	End of timer array unit 0 channel 3 count or capture		0032H	
	20	INTAD	End of A/D conversion		0034H	
	21	INTST2 /INTCSI20 /INTIIC20	UART2 transmission transfer end or buffer empty interrupt/CSI20 transfer end or buffer empty interrupt/IIC20 transfer end		003CH	

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 32 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 15-1.

3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

4. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

5. INTP1/P50 and INTP2/P51 are used for IO-Link communication.

Table 15-1. Interrupt Source List (2/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	22	INTTM13	End of timer array unit 1 channel 3 count or capture	Internal	0040H	(A)
	23	INTTM04	End of timer array unit 0 channel 4 count or capture		0042H	
	24	INTTM05	End of timer array unit 0 channel 5 count or capture		0044H	
	25	INTTM06	End of timer array unit 0 channel 6 count or capture		0046H	
	26	INTTM07	End of timer array unit 0 channel 7 count or capture		0048H	
	27	INTSR2	UART2 reception transfer end		004AH	
	28	INTTM10	End of timer array unit 1 channel 0 count or capture		0056H	
	29	INTTM11	End of timer array unit 1 channel 1 count or capture		0058H	
	30	INTTM12	End of timer array unit 1 channel 2 count or capture		005AH	
	31	INTSRE2	UART2 reception communication error occurrence		005CH	
	32	INTMD	End of division operation		005EH	
Software	–	BRK	Execution of BRK instruction	–	007EH	(C)
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–
		POC	Power-on-clear			
		LVI	Low-voltage detection[Note 3]			
		WDT	Overflow of watchdog timer			
		TRAP	Execution of illegal instruction[Note 4]			

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 32 indicates the lowest priority.

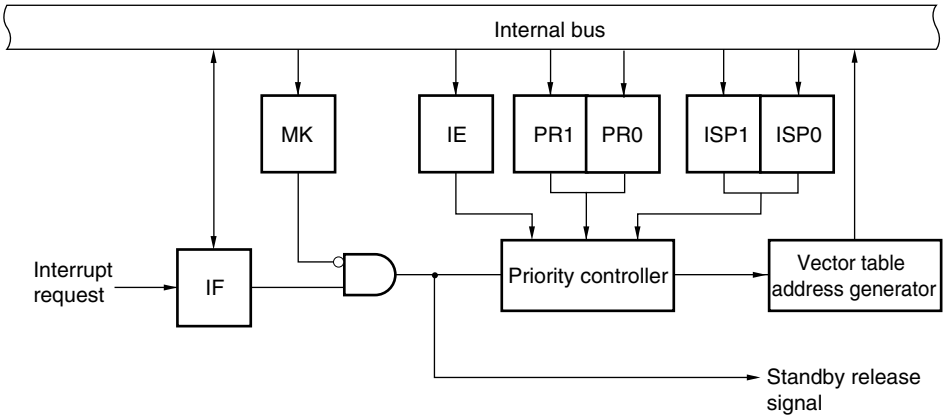
2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 15-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

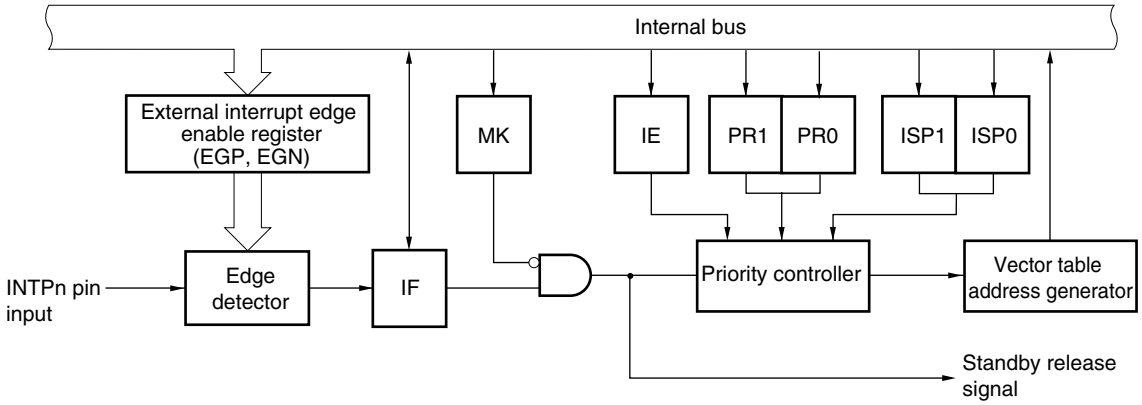
4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 15-1. Basic Configuration of Interrupt Function

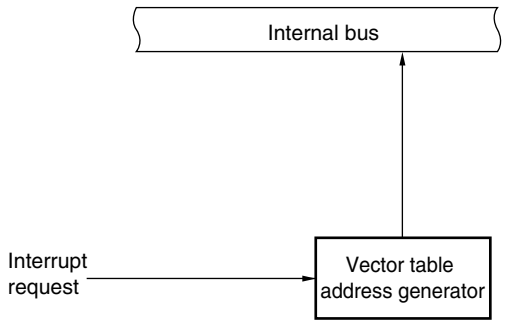
(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)



(C) Software interrupt



Remarks 1. 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

2. n = 0, 1, 2, 4, 5

15.3 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 register (EGP0)
- External interrupt falling edge enable register (EGN0)
- Program status word (PSW)

Table 15-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 15-2. Flags Corresponding to Interrupt Request Sources (1/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1	
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP1	PIF1		PMK1		PPR01, PPR11	
INTP2	PIF2		PMK2		PPR02, PPR12	
INTP4	PIF4		PMK4		PPR04, PPR14	
INTP5	PIF5		PMK5		PPR05, PPR15	
INTST3	STIF3	IF0H	STMK3	MK0H	STPR03, STPR13	PR00H, PR10H
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13	
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13	
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
INTST0	STIF0		STMK0		STPR00, STPR10	
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	

Table 15-2. Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register
INTIICA	IICAIF	IF1L	IICAMK	MK1L	IICAPR0, IICAPR1	PR01L, PR11L
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H
INTST2[Note]	STIF2[Note]		STMK2[Note]		STPR02, STPR12[Note]	
INTCSI20[Note]	CSIIF20[Note]		CSIMK20[Note]		CSIPR020, CSIPR120[Note]	
INTIIC20[Note]	IICIF20[Note]		IICMK20[Note]		IICPR020, IICPR120[Note]	
INTTM13	TMIF13		TMMK13		TMPR013, TMPR113	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12	
INTTM10	TMIF10	IF2H	TMMK10	MK2H	TMPR010, TMPR110	PR02H, PR12H
INTTM11	TMIF11		TMMK11		TMPR011, TMPR111	
INTTM12	TMIF12		TMMK12		TMPR012, TMPR112	
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12	
INTMD	MDIF		MDMK		MDPR0, MDPR1	

Note 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 4 of IF1H is set to 1. Bit 4 of MK1H, PR01H, and PR11H supports these three 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.

IF0L, IF0H, IF1L, IF1H, IF2L, and IF2H can be set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, IF1L and IF1H, and IF2L and IF2H 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 15-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	0	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0	SRIF0	STIF0	DMAIF1	DMAIF0	SREIF3	SRIF3	STIF3

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICAIF	0	0	0

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	3	2	1	<0>
IF1H	TMIF04	TMIF13	0	STIF2 CSIIF20 IICIF20	0	0	0	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
IF2L	0	0	0	0	SRIF2	TMIF07	TMIF06	TMIF05

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
IF2H	0	0	MDIF	SREIF2	TMIF12	TMIF11	TMIF10	0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

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

Cautions 1. Be sure to clear bit 5 of IF0L and bits 0 to 2 of IF1L and bits 1 to 3 and 5 of IF1H and bits 4 to 7 of IF2L and bits 0, 6, and 7 of IF2H 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 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.

MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H can be set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, MK1L and MK1H, and MK2L and MK2H 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 15-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	1	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0	SRMK0	STMK0	DMAMK1	DMAMK0	SREMK3	SRMK3	STMK3

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICAMK	1	1	1

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	3	2	1	<0>
MK1H	TMMK04	TMMK13	1	STMK2 CSIMK20 IICMK20	1	1	1	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
MK2L	1	1	1	1	SRMK2	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
MK2H	1	1	MDMK	SREMK2	TMMK12	TMMK11	TMMK10	1

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to clear bit 5 of MK0L and bits 0 to 2 of MK1L and bits 1 to 3 and 5 of MK1H and bits 4 to 7 of MK2L and bits 0, 6, and 7 of MK2H 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).

PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H can be set by a 1-bit or 8-bit memory manipulation instruction. If PR00L and PR00H, PR01L and PR01H, PR02L and PR02H, PR10L and PR10H, PR11L and PR11H, and PR12L and PR12H 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 15-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	1	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	1	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00	SRPR00	STPR00	DMAPR01	DMAPR00	SREPR03	SRPR03	STPR03

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	SRPR10	STPR10	DMAPR11	DMAPR10	SREPR13	SRPR13	STPR13

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICAPR0	1	1	1

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR1	1	1	1

Figure 15-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	3	2	1	<0>
PR01H	TMPR004	TMPR013	1	STPR02 CSIPR020 IICPR020	1	1	1	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	3	2	1	<0>
PR11H	TMPR104	TMPR113	1	STPR12 CSIPR120 IICPR120	1	1	1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
PR02L	1	1	1	1	SRPR02	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
PR12L	1	1	1	1	SRPR12	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
PR02H	1	1	MDPR0	SREPR02	TMPR012	TMPR011	TMPR010	1

Address: FFFDDH After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
PR12H	1	1	MDPR1	SREPR12	TMPR112	TMPR111	TMPR110	1

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 Be sure to set bit 5 of PR00L and PR10L and bits 0 to 2 of PR01L and PR11L and bits 1 to 3 and 5 of PR01H and PR11H bits 4 to 7 of PR02L and PR12L and bits 0, 6, and 7 of PR02H and PR12H to 1.

(4) External interrupt rising edge enable register (EGP0), external interrupt falling edge enable register (EGN0)

These registers specify the valid edge for INTP0, INTP1, INTP2, INTP4, and INTP5.

EGP0 and EGN0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 15-5. Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	0	EGP5	EGP4	0	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	0	EGN5	EGN4	0	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0, 1, 2, 4, 5)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 15-3 shows the ports corresponding to EGPn and EGNn.

Table 15-3. Ports Corresponding to EGPn and EGNn

Detection Enable Bit		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P50	INTP1
EGP2	EGN2	P51	INTP2
EGP4	EGN4	P31	INTP4
EGP5	EGN5	P16	INTP5

Caution Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark n = 0, 1, 2, 4, 5

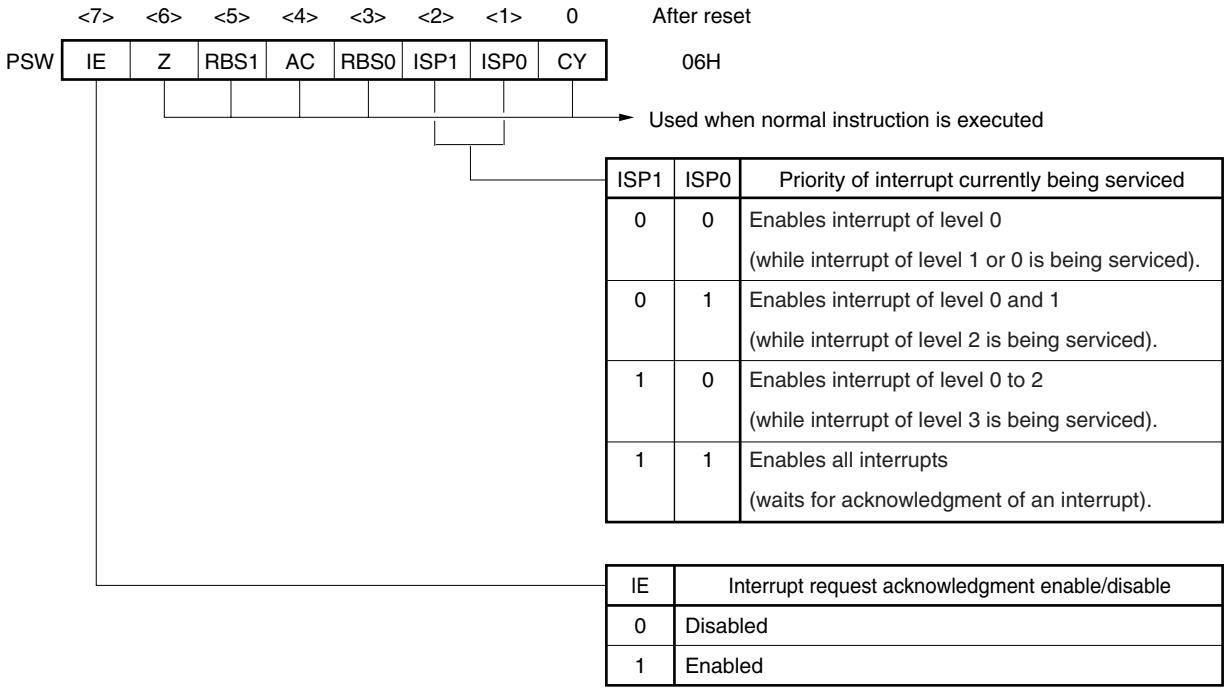
(5) Program status word (PSW)

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 15-6. 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

15.4 Interrupt Servicing Operations

15.4.1 Maskable interrupt acknowledgment

A maskable interrupt 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 15-4 below.

For the interrupt request acknowledgment timing, see **Figures 15-8** and **15-9**.

Table 15-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	9 clocks	14 clocks

Note If an interrupt request is generated just before the RET instruction, the wait time becomes longer.

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.

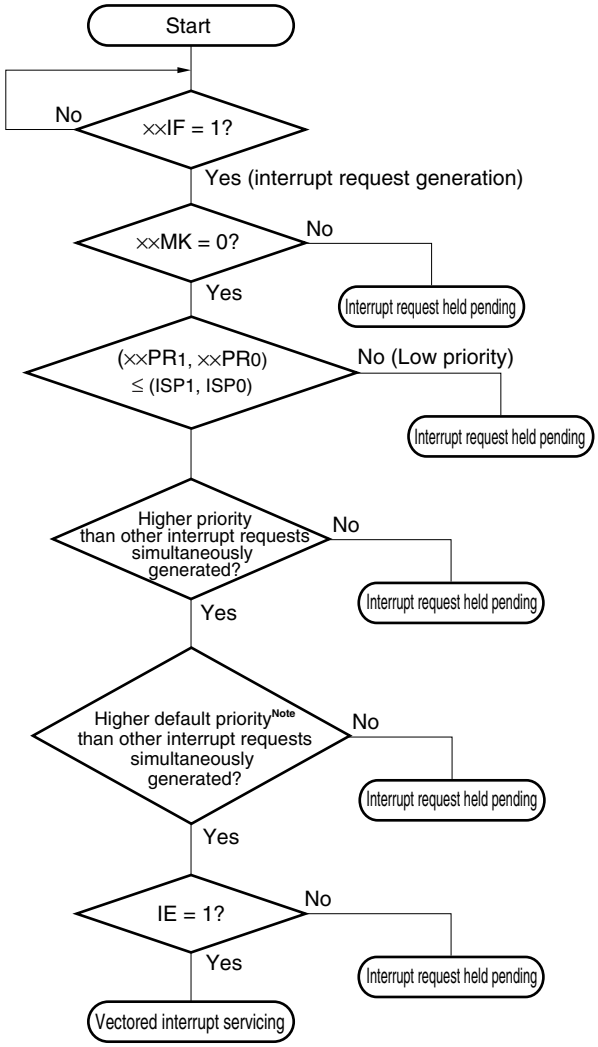
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 15-7 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.

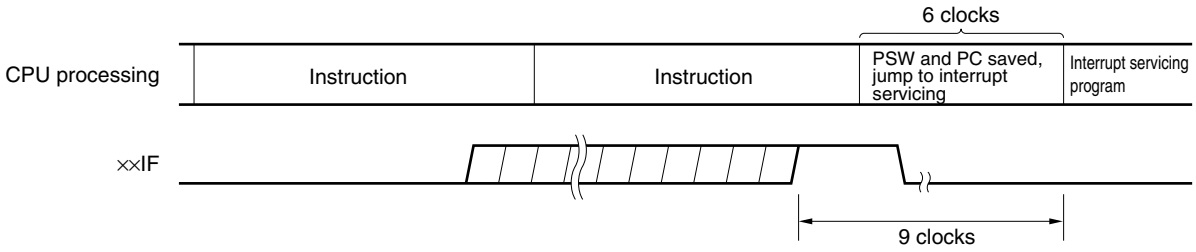
Figure 15-7. Interrupt Request Acknowledgment Processing Algorithm



××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 15-6**)

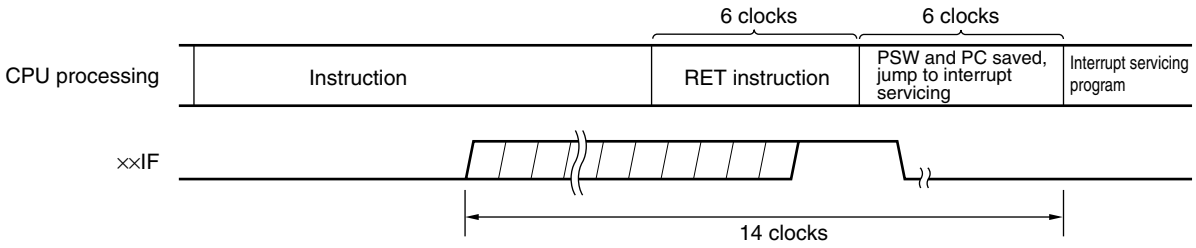
Note For the default priority, refer to **Table 15-1 Interrupt Source List**.

Figure 15-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 15-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

15.4.2 Software interrupt request acknowledgment

A software interrupt acknowledge 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.

15.4.3 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.

Table 15-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 15-10 shows multiple interrupt servicing examples.

Table 15-5. 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 PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)

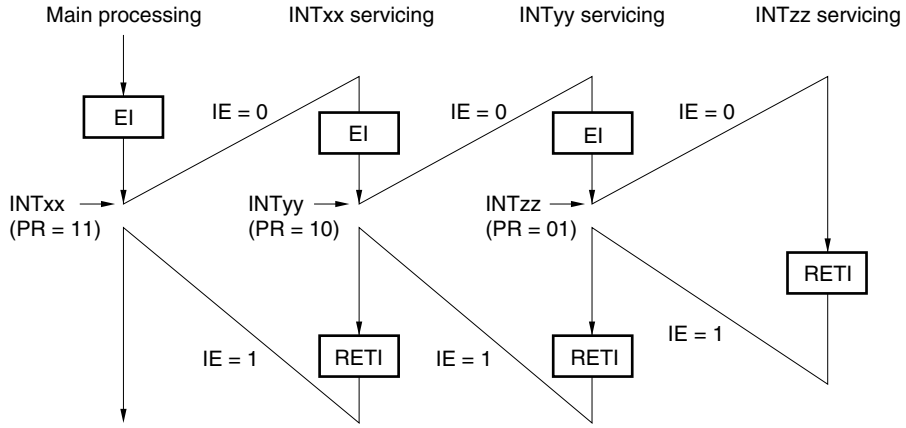
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

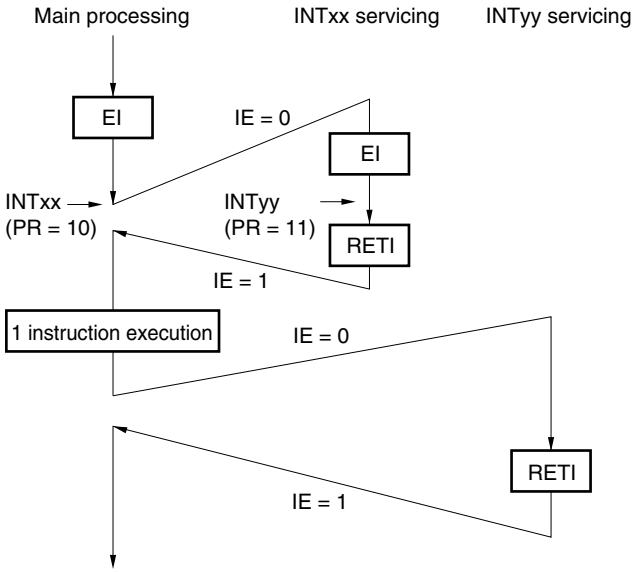
Figure 15-10. Examples of Multiple Interrupt Servicing (1/2)

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

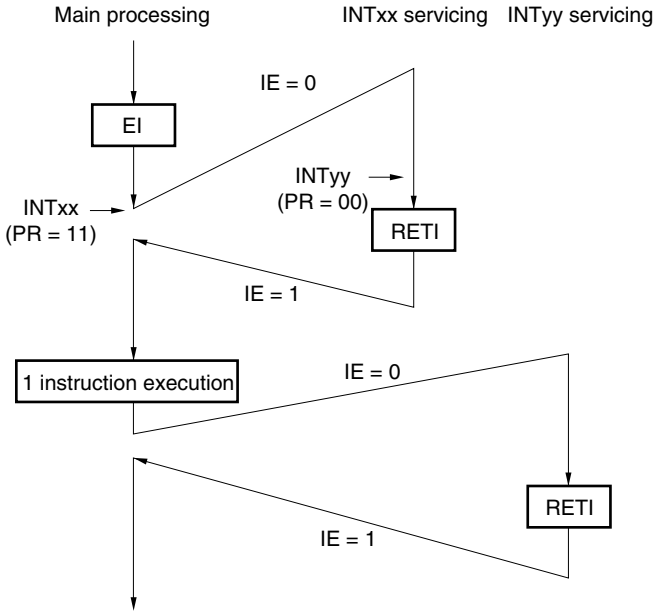


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

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 15-10. 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.

15.4.4 Interrupt request hold

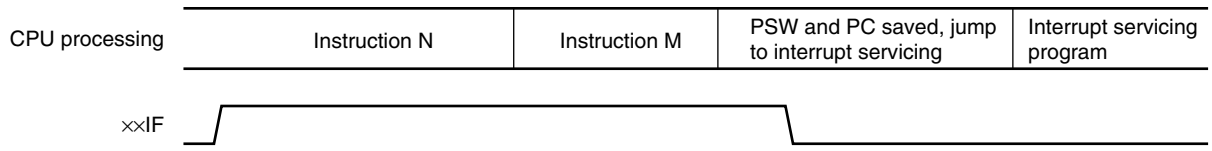
There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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 15-11 shows the timing at which interrupt requests are held pending.

Figure 15-11. 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 16 STANDBY FUNCTION

16.1 Standby Function and Configuration

16.1.1 Standby function

The standby function reduces the operating current of the system, and the following two 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, internal high-speed oscillator, or 20 MHz internal high-speed 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 internal high-speed 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.

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. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.

2. 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 the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

3. It can be selected by the option byte whether the internal low-speed oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 21 OPTION BYTE.

4. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

16.1.2 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 7 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 internal high-speed oscillation clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock with the X1 clock oscillating.

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

When reset is released (reset by $\overline{\text{RESET}}$ input, POC, LVI, WDT, and executing an illegal instruction), the STOP instruction and MSTOP (bit 7 of CSC register) = 1 clear this register to 00H.

Figure 16-1. 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.

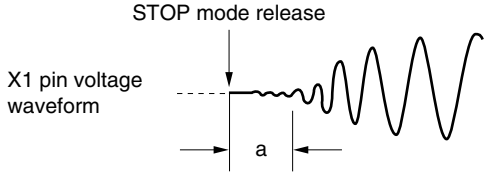
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC 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 OSTS after the STOP mode is released.

When the internal high-speed oscillation clock is selected as the CPU clock, confirm with 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 OSTC.

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

Reset signal generation sets this register to 07H.

Figure 16-2. 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	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	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 OSTS 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.

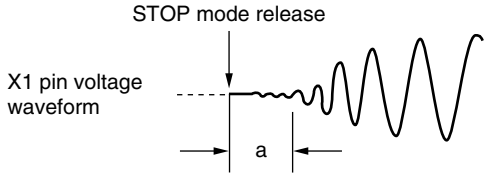
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

5. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC 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

16.2 Standby Function Operation

16.2.1 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, internal high-speed oscillation clock, or 20 MHz internal high-speed oscillation clock.

The operating statuses in the HALT mode are shown below.

Table 16-1. Operating Statuses in HALT Mode

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{IH}) or 20 MHz Internal High-Speed Oscillation Clock (f_{IH20})	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}, f_{IH20}	Operation continues (cannot be stopped)	Status before HALT mode was set is retained	
		f_X	Status before HALT mode was set is retained	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • WDTON = 1 and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			The value is retained		
Port (latch)			Status before HALT mode was set is retained		
Timer array unit TAU			Operable		
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Operates • WDTON = 1 and WDSTBYON = 0: Stops		
A/D converter			Operable		
Serial array unit (SAU)					
Serial interface (IICA)					
Multiplier/divider					
DMA controller					
Power-on-clear function					
Low-voltage detection function					
External interrupt					

Remark f_{IH}: Internal high-speed oscillation clock
f_{IH20}: 20 MHz internal high-speed oscillation clock
f_X: X1 clock
f_{EX}: External main system clock
f_{IL}: Internal low-speed oscillation clock

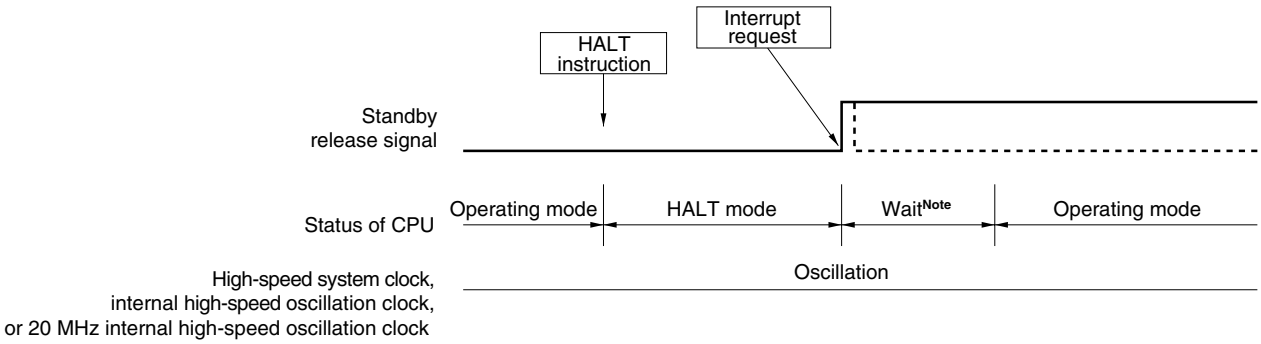
(2) HALT mode release

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 16-3. HALT Mode Release by Interrupt Request Generation



Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 10 to 12 clocks
- When vectored interrupt servicing is not carried out: 5 or 6 clocks

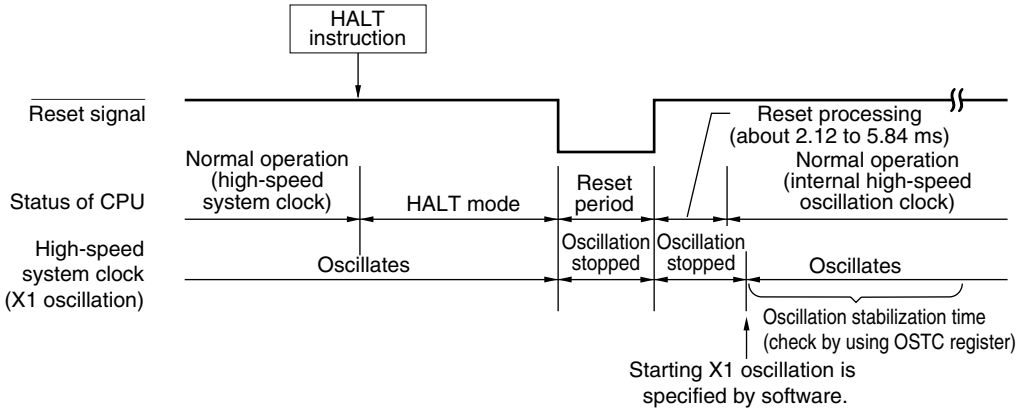
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

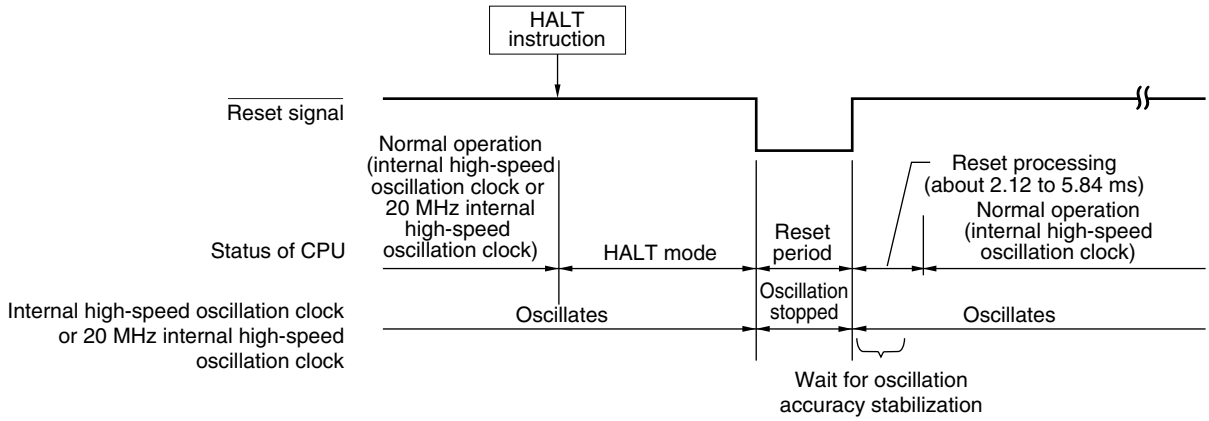
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 16-4. HALT Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock or 20 MHz internal high-speed oscillation clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

16.2.2 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 external main system clock.

Cautions 1. Because the interrupt request signal is used to release 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 released 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. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

The operating statuses in the STOP mode are shown below.

Table 16-2. Operating Statuses in STOP Mode

STOP Mode Setting / Item			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation 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}			
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • WDTON = 1 and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			The value is retained		
Port (latch)			Status before STOP mode was set is retained		
Timer array unit TAU			Operation disabled		
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Operates • WDTON = 1 and WDSTBYON = 0: Stops		
A/D converter			Operation disabled		
Serial array unit (SAU)					
Serial interface (IICA)			Wakeup by address match operable		
Multiplier/divider			Operation disabled		
DMA controller					
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Remark f_{IH}: Internal high-speed oscillation clock
f_X: X1 clock
f_{EX}: External main system clock
f_{IL}: Internal low-speed oscillation 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 internal low-speed oscillation 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 internal high-speed oscillation clock before the execution of the STOP instruction. Before changing the CPU clock from the internal high-speed oscillation 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).

4. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

(2) STOP mode release

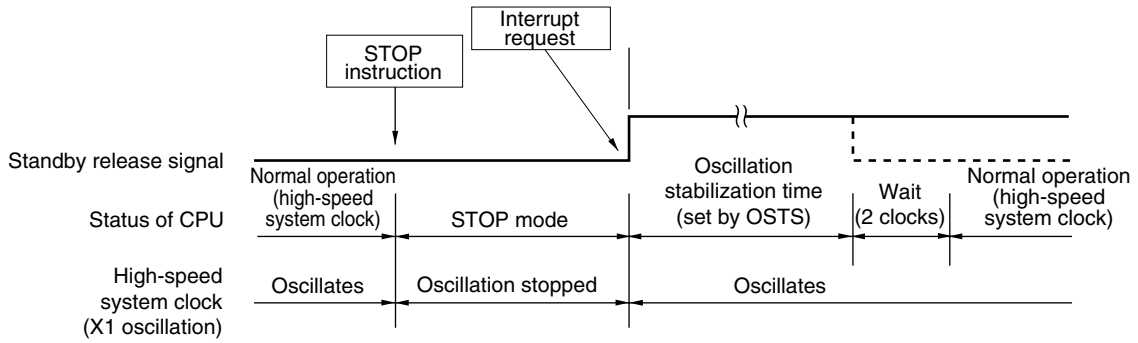
The STOP mode can be released by the following two sources.

(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 16-5. STOP Mode Release by Interrupt Request Generation (1/2)

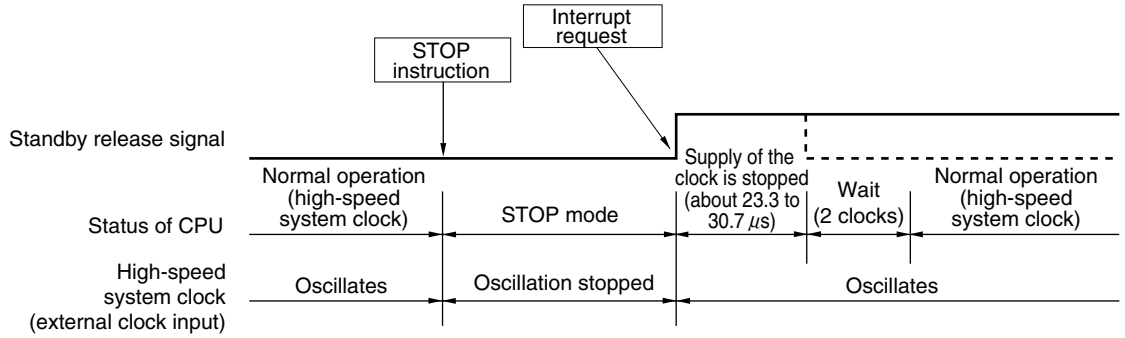
(1) When high-speed system clock (X1 oscillation) is used as CPU clock



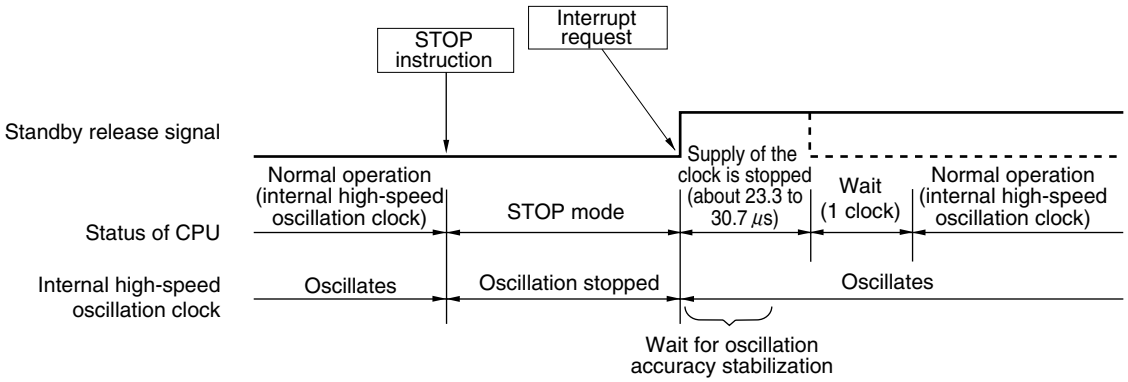
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 16-5. STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (external clock input) is used as CPU clock



(3) When internal high-speed oscillation clock is used as CPU clock



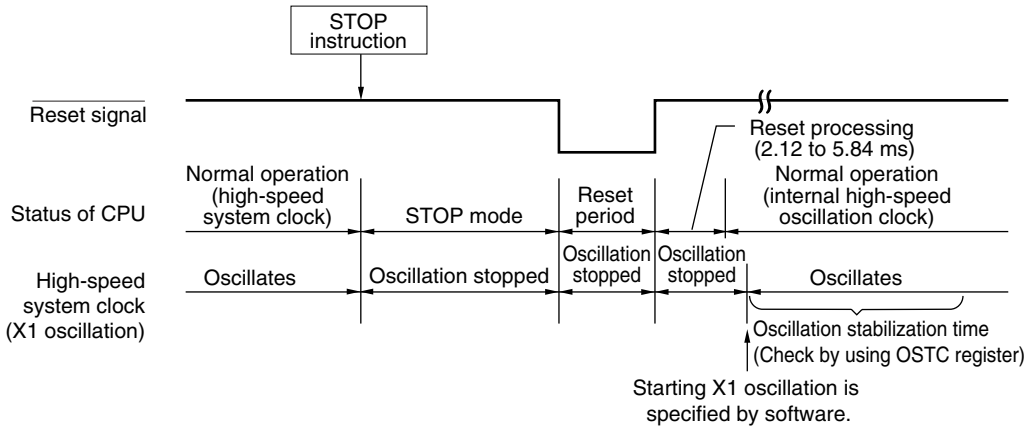
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

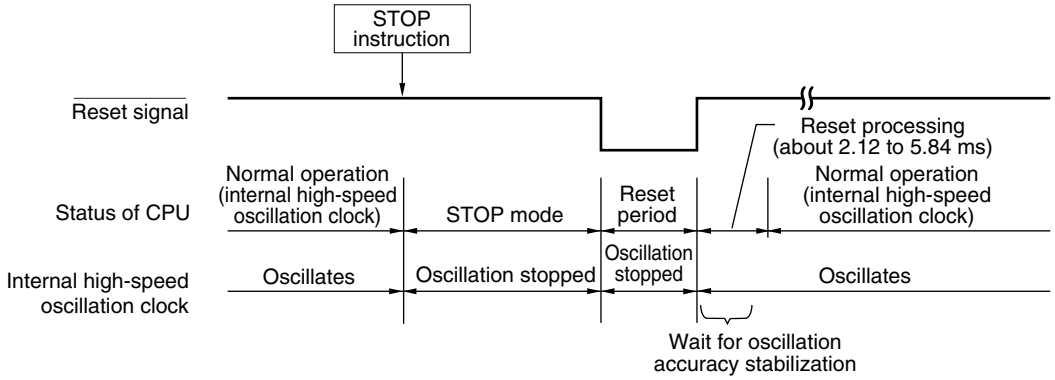
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 16-6. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

CHAPTER 17 RESET FUNCTION

The following six 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-clear (POC) circuit
(4) Internal reset by comparison of supply voltage or input voltage (EXLVI) from external input pin, and detection voltage of the low-voltage detector (LVI)
(5) Internal reset by execution of illegal instruction[Note]
(6) Internal reset by a reset processing check error

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 POC and LVI circuit voltage detection or execution of illegal instruction[Note], and each item of hardware is set to the status shown in Tables 17-1 and 17-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release.

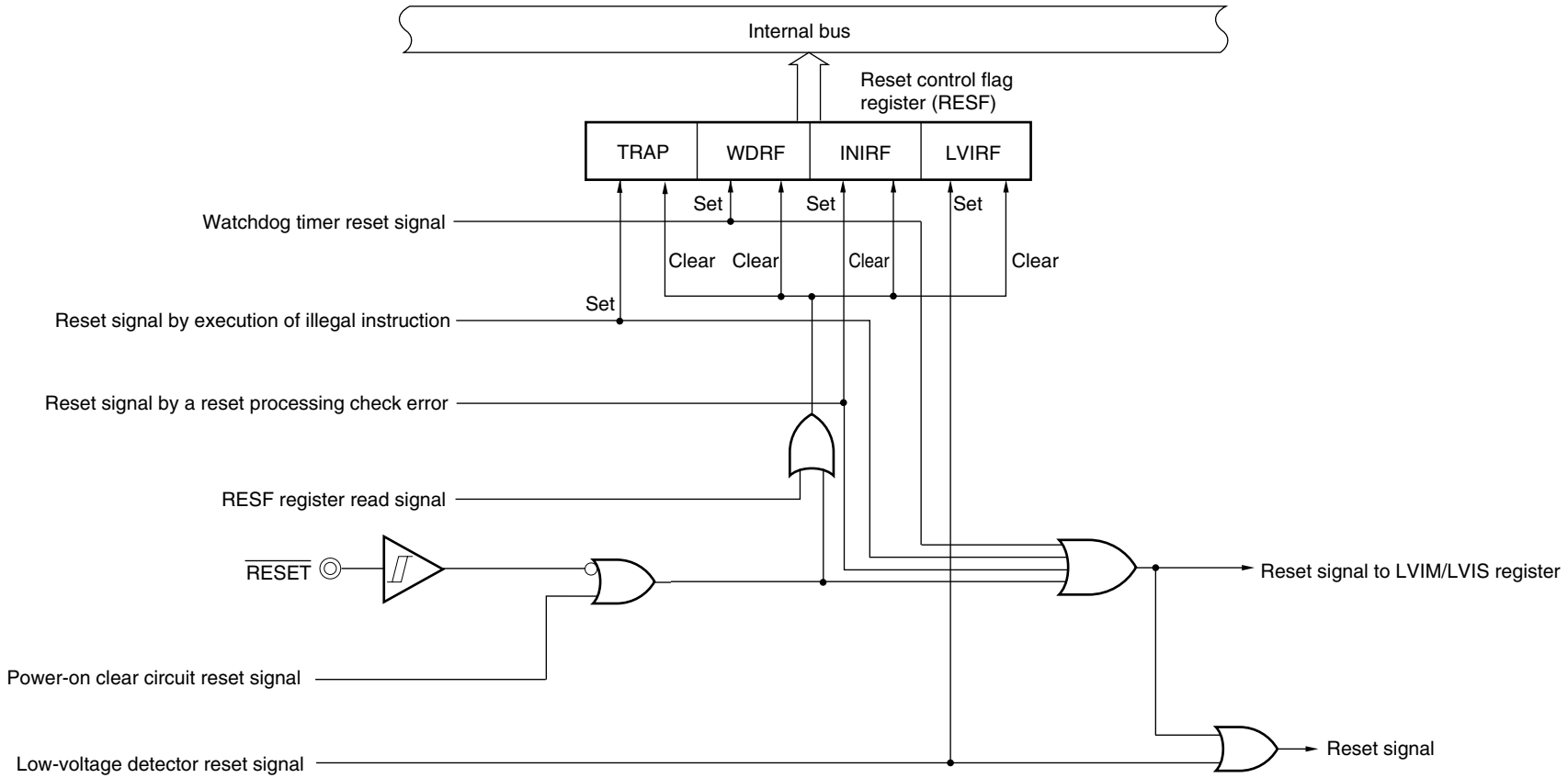
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 17-2** to **17-4**) after reset processing. Reset by POC and LVI circuit 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 internal high-speed oscillation clock (see **CHAPTER 18 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 19 LOW-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 ($V_{DD} \geq 3.0$ V).)
2. During reset input, the X1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.
3. When the STOP mode is released by a reset, the RAM contents in the STOP mode are held during reset input.
4. When reset is effected, port pins become high-impedance, because each SFR and 2nd SFR are initialized.

Remark V_{POR}: POC power supply rise detection voltage

Figure 17-1. Block Diagram of Reset Function



Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level select register

Figure 17-2. Timing of Reset by $\overline{\text{RESET}}$ Input

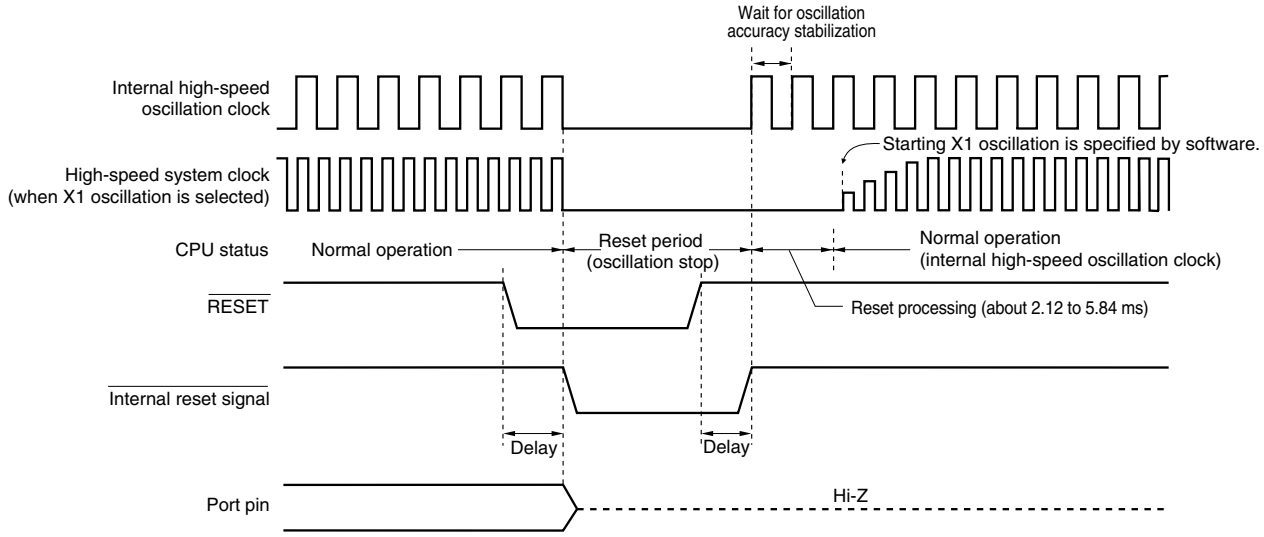
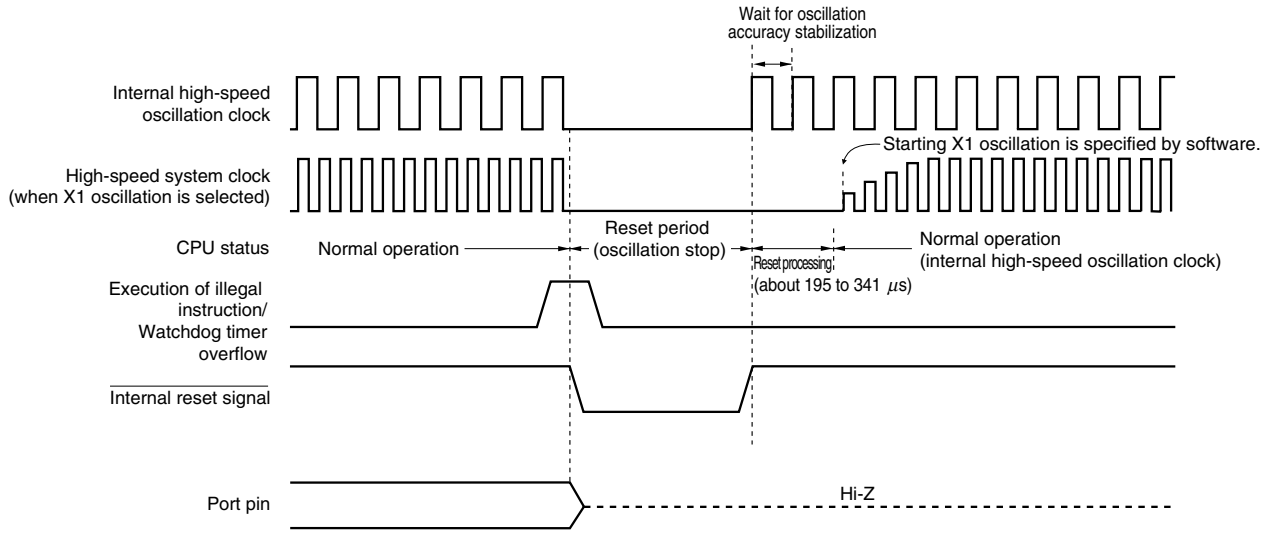
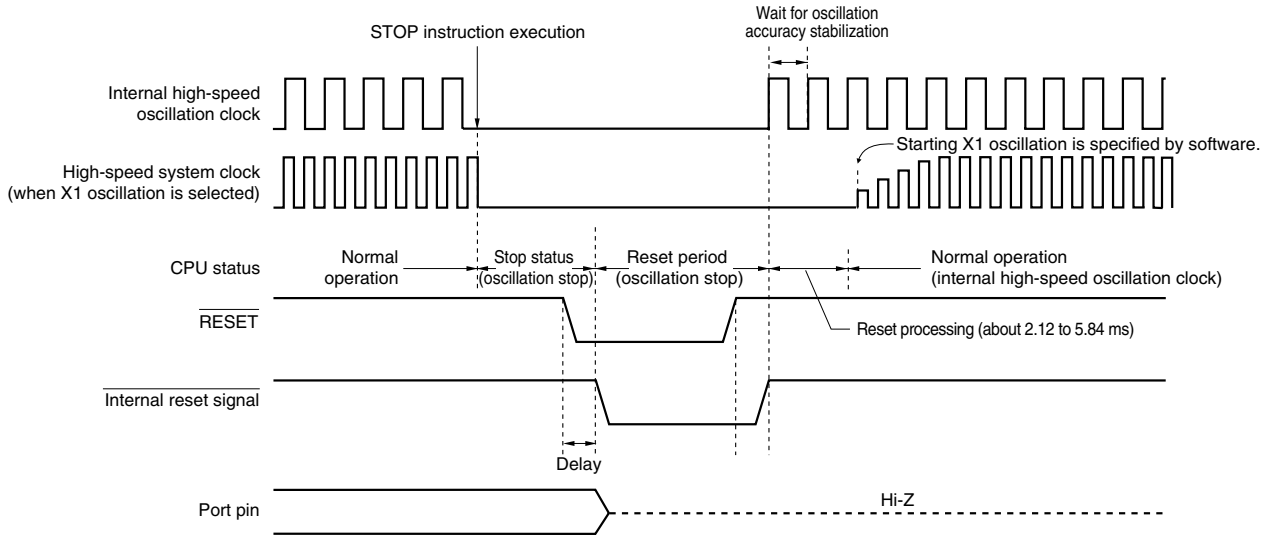


Figure 17-3. Timing of Reset Due to Execution of Illegal Instruction or Watchdog Timer Overflow



Caution A watchdog timer internal reset also resets the watchdog timer.

Figure 17-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Remark For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 18 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 19 LOW-VOLTAGE DETECTOR**.

Table 17-1. Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}, f_{IH20}	Operation stopped
		f_X	Operation stopped (X1 pin is input port mode)
		f_{EX}	Clock input invalid (pin is input port mode)
	f_{IL}		Operation stopped
CPU			
Flash memory			
RAM			Operation stopped (The value, however, is retained when the voltage is at least the power-on-clear detection voltage.)
Port (latch)			The port pins become high impedance.
Timer array unit TAU			Operation stopped
Watchdog timer			
A/D converter			
Serial array unit (SAU)			
Serial interface (IICA)			
Multiplier/divider			
DMA controller			
Power-on-clear function			Detection operation possible
Low-voltage detection function			Operation stopped (however, operation continues at LVI reset)
External interrupt			Operation stopped

Remark f_{IH}: Internal high-speed oscillation clock
f_{IH20}: 20 MHz internal high-speed oscillation clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{IL}: Internal low-speed oscillation clock

Table 17-2. Hardware Statuses After Reset Acknowledgment (1/3)

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[Note 2]
	General-purpose registers	Undefined[Note 2]
Processor mode control register (PMC)		00H
Port registers (P0 to P7, P11, P12, P14, P15) (output latches)		00H
Port mode registers (PM0 to PM7, PM11 PM12, PM14, PM15)		FFH
Port input mode register 14 (PIM14)		00H
Port output mode register 14 (POM14)		00H
Pull-up resistor option registers (PU0, PU1, PU3 to PU6, PU12, PU14)		00H
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
System clock control register (CKC)		09H
20 MHz internal high-speed oscillation control register (DSCCTL)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		07H
Noise filter enable registers 0 to 2 (NFEN0 to NFEN2)		00H
Peripheral enable register 0 (PER0)		00H
Operation speed mode control register (OSMC)		00H
Timer array unit (TAU)	Timer data registers 00 to 07, 10 to 13 (TDR00 to TDR07, TDR10 to TDR13)	0000H
	Timer mode registers 00 to 07, 10 to 13 (TMR00 to TMR07, TMR10 to TMR13)	0000H
	Timer status registers 00 to 07, 10 to 13 (TSR00 to TSR07, TSR10 to TSR13)	0000H
	Timer counter registers 00 to 07, 10 to 13 (TCR00 to TCR07, TCR10 to TCR13)	FFFFH
	Timer channel enable status registers 0, 1 (TE0, TE1)	0000H
	Timer channel start registers 0, 1 (TS0, TS1)	0000H
	Timer channel stop registers 0, 1 (TT0, TT1)	0000H
	Timer clock select registers 0, 1 (TPS0, TPS1)	0000H
	Timer output registers 0, 1 (TO0, TO1)	0000H
	Timer output enable registers 0 , 1(TOE0, TOE1)	0000H
	Timer output level registers 0, 1 (TOL0, TOL1)	0000H
	Timer output mode registers 0, 1 (TOM0, TOM1)	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. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

Table 17-2. Hardware Statuses After Reset Acknowledgment (2/3)

Hardware		Status After Reset Acknowledgment[Note 1]
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
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	10H
Serial array unit (SAU)	Serial data registers 00, 01, 10 to 13 (SDR00, SDR01, SDR10 to SDR13)	0000H
	Serial status registers 00, 01, 10 to 13 (SSR00, SSR01, SSR10 to SSR13)	0000H
	Serial flag clear trigger registers 00, 01, 10 to 13 (SIR00, SIR01, SIR10 to SIR13)	0000H
	Serial mode registers 00, 01, 10 to 13 (SMR00, SMR01, SMR10 to SMR13)	0020H
	Serial communication operation setting registers 00, 01, 10 to 13 (SCR00, SCR01, SCR10 to SCR13)	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 level registers 0, 1 (SOL0, SOL1)	0000H
Serial interface IICA	IICA shift register (IICA)	00H
	IICA status register (IICS)	00H
	IICA flag register (IICF)	00H
	IICA control register 0 (IICCTL0)	00H
	IICA control register 1 (IICCTL1)	00H
	IICA low-level width setting register (IICWL)	FFH
	IICA high-level width setting register (IICWH)	FFH
	Slave address register (SVA)	00H
Multiplier/divider	Multiplication/division data register A (L) (MDAL)	0000H
	Multiplication/division data register A (H) (MDAH)	0000H
	Multiplication/division data register B (L) (MDBL)	0000H
	Multiplication/division data register B (H) (MDBH)	0000H
	Multiplication/division data register C (L) (MDCL)	0000H
	Multiplication/division data register C (H) (MDCH)	0000H
	Multiplication/division control register (MDUC)	00H

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

Table 17-2. Hardware Statuses After Reset Acknowledgment (3/3)

Hardware		Status After Reset Acknowledgment[Note 1]
Reset function	Reset control flag register (RESF)	00H[Note 2]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 3]
	Low-voltage detection level select register (LVIS)	0EH[Note 2]
Regulator	Regulator mode control register (RMC)	00H
DMA controller	SFR address registers 0, 1 (DSA0, DSA1)	00H
	RAM address registers 0L, 0H, 1L, 1H (DRA0L, DRA0H, DRA1L, DRA1H)	00H
	Byte count registers 0L, 0H, 1L, 1H (DBC0L, DBC0H, DBC1L, DBC1H)	00H
	Mode control registers 0, 1 (DMC0, DMC1)	00H
	Operation control registers 0, 1 (DRC0, DRC1)	00H
Interrupt	Request flag registers 0L, 0H, 1L, 1H, 2L, 2H (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)	00H
	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, PR10L, PR10H, PR11L, PR11H, PR02L, PR02H, PR12L, PR12H)	FFH
	External interrupt rising edge enable register 0 (EGP0)	00H
	External interrupt falling edge enable register 0 (EGN0)	00H
BCD correction circuit	BCD correction result register (BCDAJ)	Undefined

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 POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by INIRF	Reset by LVI
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held
	WDRF bit			Held	Set (1)	Held	Held
	INIRF bit			Held	Held	Set (1)	Held
	LVIRF bit			Held	Held	Held	Set (1)
LVIS		Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Held

3. This value varies depending on the reset source and the option byte.

17.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the μPD78F8040, 78F8041, 78F8042, 78F8043. The reset control flag register (RESF) is used to store which source has generated the reset request.

RESF can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset by power-on-clear (POC) circuit, and reading the RESF register clear TRAP, WDRF, INIRF, and LVIRF flags.

Figure 17-5. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP Note 1	Undefined	Undefined	WDRF Note 1	Undefined	Undefined	INIRF	LVIRF Note 1

TRAP	Internal reset request by execution of illegal instruction Note 2
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

WDRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

INIRF	Internal rese request t by a reset processing check error
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF 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. When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, LVIRF flag may become 1 from the beginning depending on the power-on waveform.

The status of RESF when a reset request is generated is shown in Table 17-3.

Table 17-3. RESF Status When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by INIRF	Reset by LVI
TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held
WDRF bit			Held	Set (1)	Held	Held
INIRF bit			Held	Held	Set (1)	Held
LVIRF bit			Held	Held	Held	Set (1)

CHAPTER 18 POWER-ON-CLEAR CIRCUIT

18.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds 1.61 V ±0.09 V.

  Caution If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V), generates internal reset signal when $V_{DD} < V_{PDR}$.

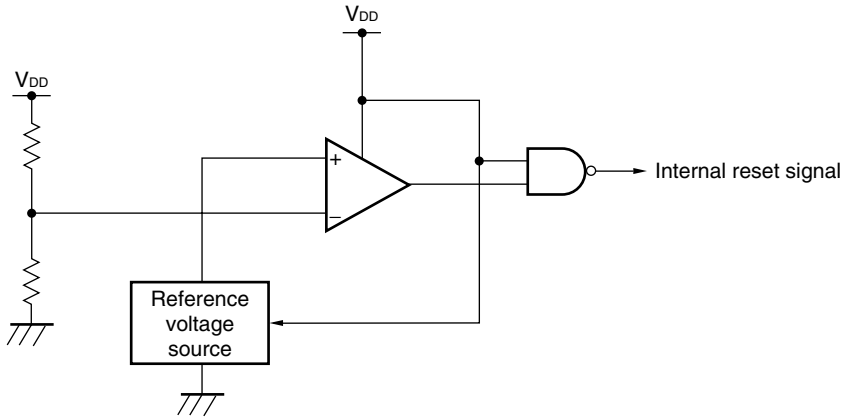
  Caution If an internal reset signal is generated in the POC circuit, TRAP, WDRF, INIRF, and LVIRF of the reset control flag register (RESF) is cleared.

  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), low-voltage-detector (LVI), or illegal instruction execution. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT, LVI, or illegal instruction execution.
  For details of RESF, see **CHAPTER 17 RESET FUNCTION**.

18.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 18-1.

Figure 18-1. Block Diagram of Power-on-Clear Circuit



18.3 Operation of Power-on-Clear Circuit

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{PDR} = 1.61 V ±0.09 V), the reset status is released.

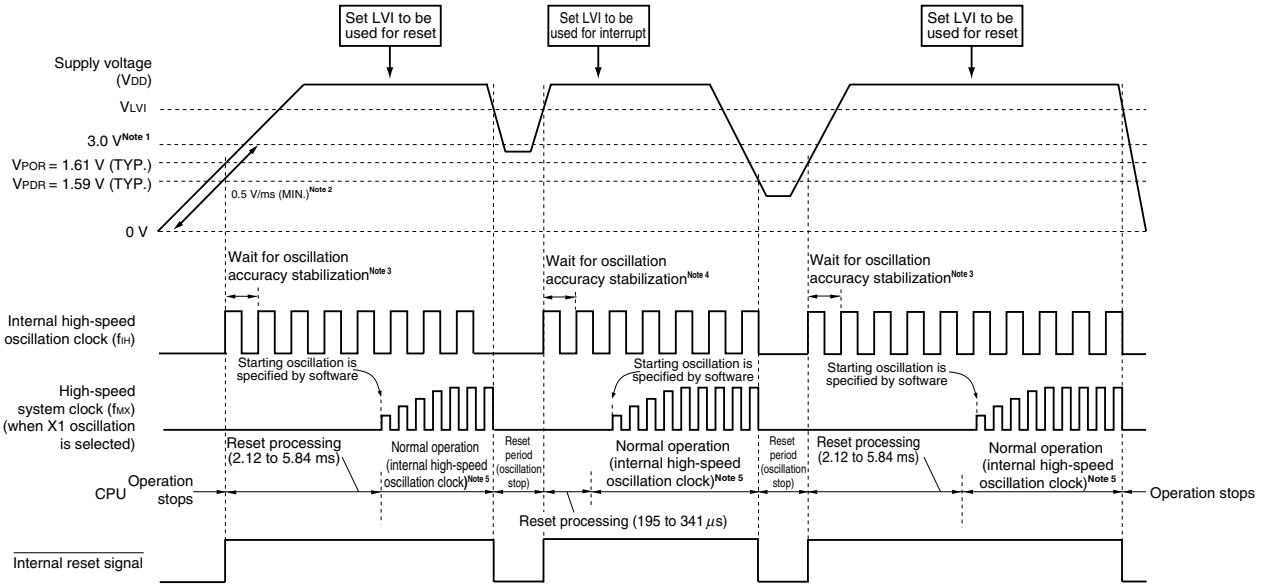
  Caution If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V.

- The supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 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-clear circuit and low-voltage detector is shown below.

Figure 18-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) When LVI is OFF upon power application (option byte: LVIOFF = 1)



Notes 1. The operation guaranteed range is 3.0 V ≤ V_{DD} ≤ 5.5 V. Perform normal operation after the supply voltage reaches 3.0 V or more.

To make the state at lower than 3.0 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. If the rate at which the voltage rises to 3.0 V after power application is slower than 0.5 V/ms (MIN.), input a low level to the $\overline{\text{RESET}}$ pin before the voltage reaches to 3.0 V, or set LVI to ON by default by using an option byte (option byte: LVIOFF = 0).

3. The reset processing time, such as when waiting for internal voltage stabilization, includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

4. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

5. The internal high-speed oscillation clock and a high-speed system clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time.

Cautions 1. Set the low-voltage detector by software after the reset status is released (see CHAPTER 19 LOW-VOLTAGE DETECTOR).

2. Some operations can also be executed while V_{DD} < 3.0 V (For details, figures of CHAPTER 26 ELECTRICAL SPECIFICATIONS AC Characteristics (1) Basic operation).

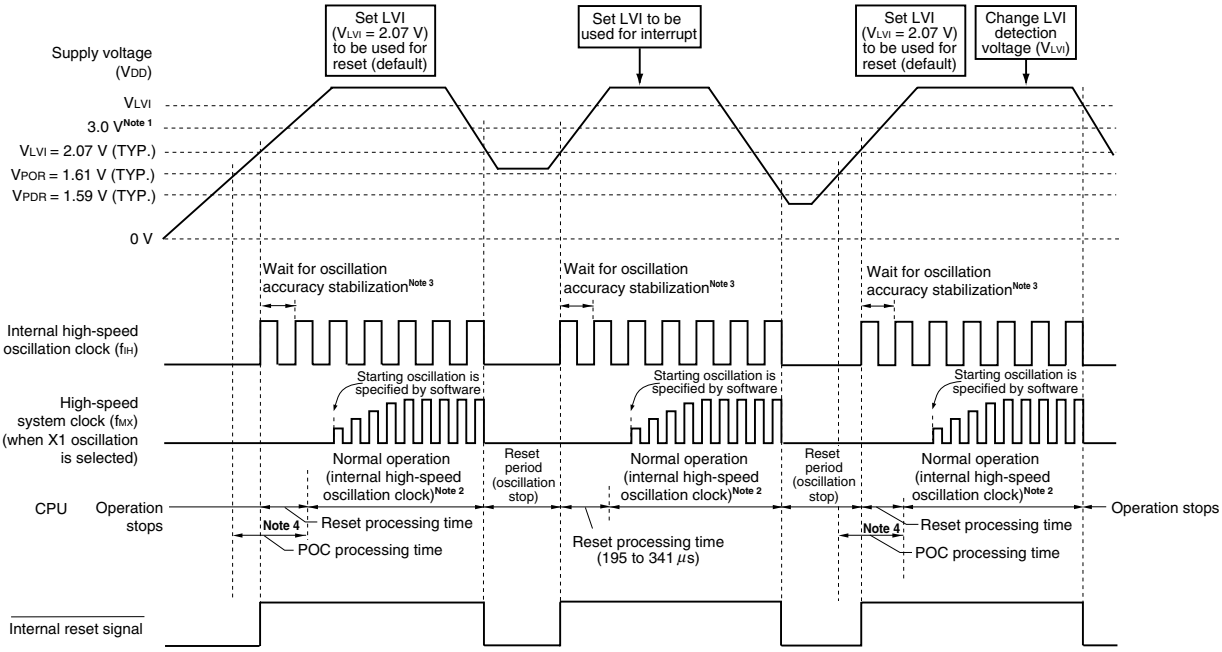
Remark V_{LVI}: LVI detection voltage

V_{POR}: POC power supply rise detection voltage

V_{PDR}: POC power supply fall detection voltage

Figure 18-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) When LVI is ON upon power application (option byte: LVIOFF = 0)



Notes 1. The operation guaranteed range is 3.0 V ≤ V_{DD} ≤ 5.5 V. Perform normal operation after the supply voltage reaches 3.0 V or more.

To make the state at lower than 3.0 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. The internal high-speed oscillation clock and a high-speed system clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time.

3. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

4. The following times are required between reaching the POC detection voltage (1.61 V (TYP.)) and starting normal operation.

- When the time to reach 2.07 V (TYP.) from 1.61 V (TYP.) is less than 5.8 ms:
  A POC processing time of 2.12 to 5.84 ms is required between reaching 1.61 V (TYP.) and starting normal operation.
- When the time to reach 2.07 V (TYP.) from 1.61 V (TYP.) is greater than 5.8 ms:
  A reset processing time of 195 to 341 μs is required between reaching 2.07 V (TYP.) and starting normal operation.

Cautions 1. Set the low-voltage detector by software after the reset status is released (see CHAPTER 19 LOW-VOLTAGE DETECTOR).

2. Some operations can also be executed while V_{DD} < 3.0 V (For details, figures of CHAPTER 26 ELECTRICAL SPECIFICATIONS AC Characteristics (1) Basic operation).

Remark V_{LVI}: LVI detection voltage
V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

18.4 Cautions for Power-on-Clear Circuit

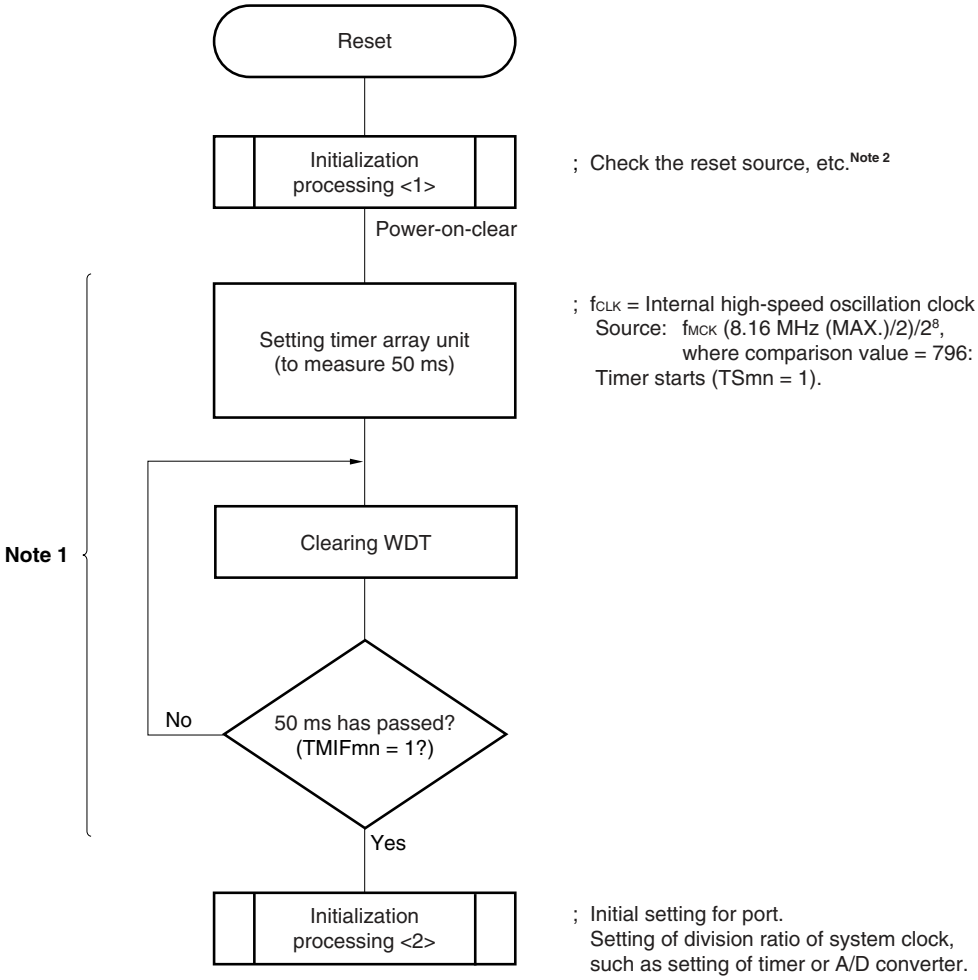
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POR}, V_{PDR}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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 18-3. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage



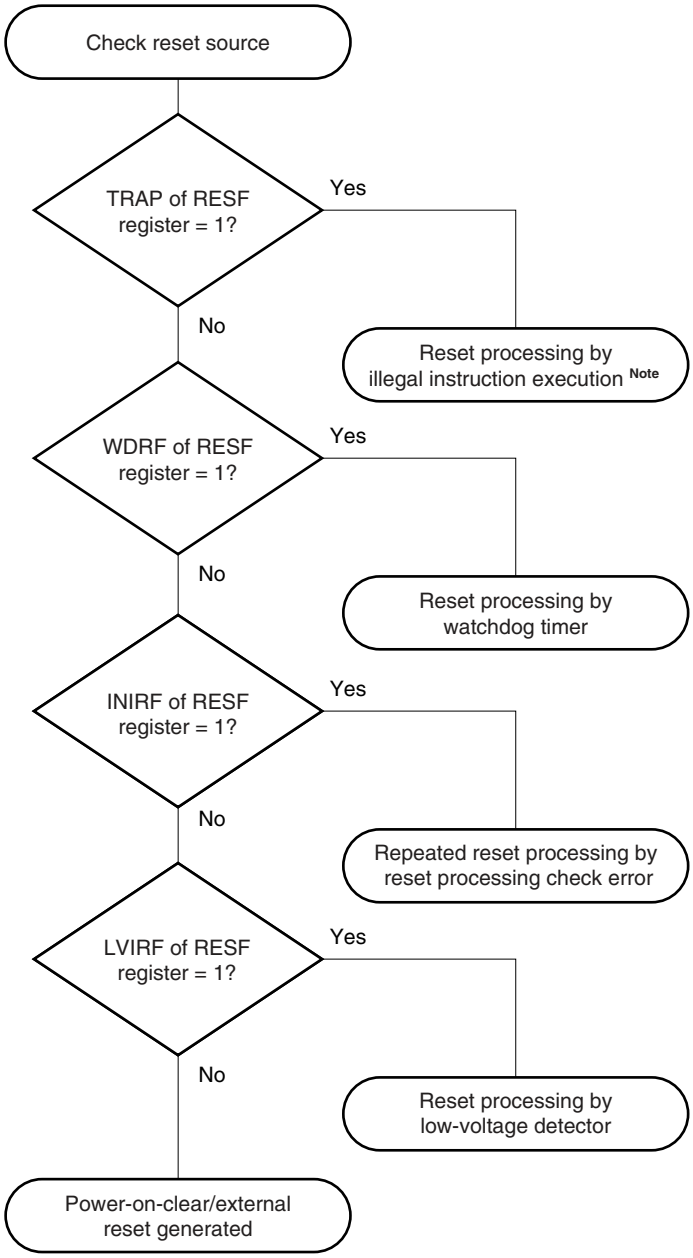
Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Remark m = 0, 1, n = 0 to 7, mn = 00 to 07, 10 to 13

Figure 18-3. 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 19 LOW-VOLTAGE DETECTOR

19.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has the following functions.

- The LVI circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVI}) or the input voltage from an external input pin (EXLVI) with the detection voltage (V_{EXLVI} = 1.21 V ±0.1 V), and generates an internal reset or internal interrupt signal.
- The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage (V_{POR} = 1.61 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.2 V). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V).
- The supply voltage (V_{DD}) or the input voltage from the external input pin (EXLVI) can be selected to be detected by software.
- A reset or an interrupt can be selected to be generated after detection by software.
- Detection levels (V_{LVI}, 8 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

The reset and interrupt signals are generated as follows depending on selection by software.

Selection of Level Detection of Supply Voltage (V_{DD}) (LVISEL = 0)		Selection Level Detection of Input Voltage from External Input Pin (EXLVI) (LVISEL = 1)	
Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).	Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).
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 ($V_{DD} \geq V_{LVI}$).	Generates an internal reset signal when EXLVI < V_{EXLVI} and releases the reset signal when EXLVI ≥ V_{EXLVI}.	Generates an internal interrupt signal when EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).

Remark LVISEL: Bit 2 of low-voltage detection register (LVIM)
LVIMD: Bit 1 of LVIM

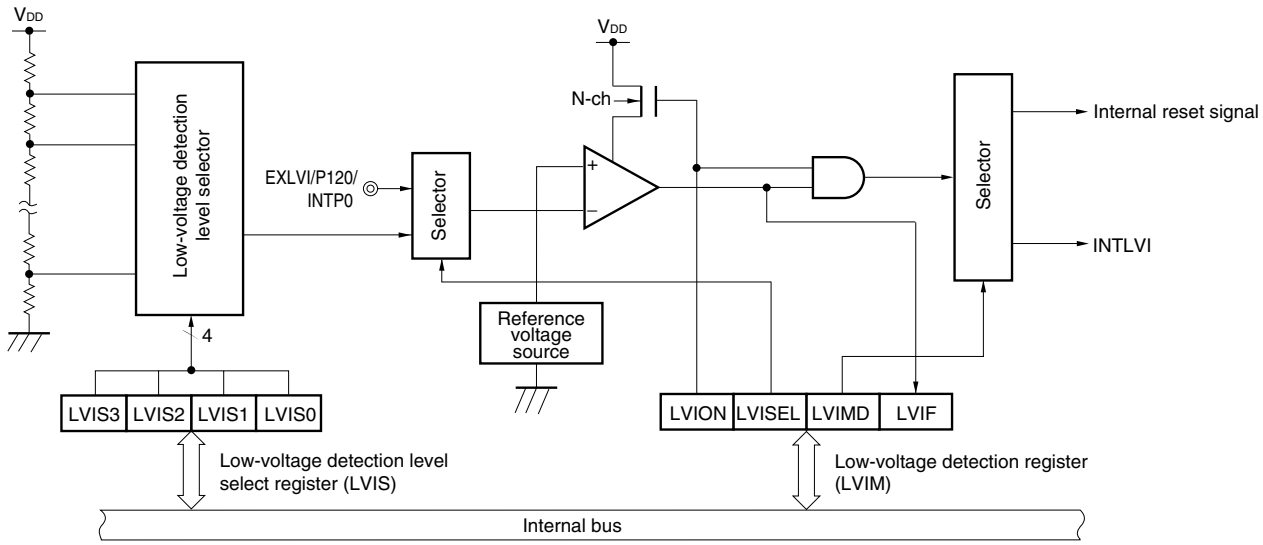
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, see **CHAPTER 17 RESET FUNCTION**.

19.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 19-1.

Figure 19-1. Block Diagram of Low-Voltage Detector



19.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level select register (LVIS)
- Port mode register 12 (PM12)

(1) Low-voltage detection register (LVIM)

This register sets low-voltage detection and the operation mode.

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

Reset signal generation clears this register to 00H.

Figure 19-2. Format of Low-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	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION[Notes 3, 4]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVISEL[Note 3]	Voltage detection selection
0	Detects level of supply voltage (V_{DD})
1	Detects level of input voltage from external input pin (EXLVI)

LVIMD[Note 3]	Low-voltage detection operation mode (interrupt/reset) selection
0	• LVISEL = 0: Generates an internal interrupt signal when the supply voltage (V_{DD}) drops lower than the detection voltage (V_{LVI}) ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$). • LVISEL = 1: Generates an interrupt signal when the input voltage from an external input pin (EXLVI) drops lower than the detection voltage (V_{EXLVI}) (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).
1	• LVISEL = 0: Generates an internal reset signal when the supply voltage (V_{DD}) < detection voltage (V_{LVI}) and releases the reset signal when $V_{DD} \geq V_{LVI}$. • LVISEL = 1: Generates an internal reset signal when the input voltage from an external input pin (EXLVI) < detection voltage (V_{EXLVI}) and releases the reset signal when EXLVI ≥ V_{EXLVI}.

LVIF	Low-voltage detection flag
0	• LVISEL = 0: Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}), or when LVI operation is disabled • LVISEL = 1: Input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI}), or when LVI operation is disabled
1	• LVISEL = 0: Supply voltage (V_{DD}) < detection voltage (V_{LVI}) • LVISEL = 1: Input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI})

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 LVI reset.
It is set to “82H” when a reset signal other than LVI is applied if option byte LVIOFF = 0, and to “00H” if option byte LVIOFF = 1.

2. Bit 0 is read-only.

3. LVION, LVIMD, and LVISEL are cleared to 0 in the case of a reset other than an LVI reset. These are not cleared to 0 in the case of an LVI reset.

Note 4. When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to wait for the following periods of time, between when LVION is set to 1 and when the voltage is confirmed with LVIF.

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

The LVIF value for these periods may be set/cleared regardless of the voltage level, and can therefore not be used. Also, the LVIIF interrupt request flag may be set to 1 in these periods.

Cautions 1. To stop LVI, be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

2. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

3. When LVI is used in interrupt mode (LVIMD = 0) and LVISEL is set to 0, an interrupt request signal (INTLVI) that disables LVI operation (clears LVION) when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}) (if LVISEL = 1, input voltage of external input pin (EXLVI) is less than or equal to the detection voltage (V_{EXLVI})) is generated and LVIIF may be set to 1.

4. To read LVIM after writing this register, secure the time of one or more clock.

(2) Low-voltage detection level select register (LVIS)

This register selects the low-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 0EH.

Figure 19-3. Format of Low-Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 0EH[Note] R/W

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	Detection level
0	0	0	0	V_{LVI0} (4.22 ±0.1 V)
0	0	0	1	V_{LVI1} (4.07 ±0.1 V)
0	0	1	0	V_{LVI2} (3.92 ±0.1 V)
0	0	1	1	V_{LVI3} (3.76 ±0.1 V)
0	1	0	0	V_{LVI4} (3.61 ±0.1 V)
0	1	0	1	V_{LVI5} (3.45 ±0.1 V)
0	1	1	0	V_{LVI6} (3.30 ±0.1 V)
0	1	1	1	V_{LVI7} (3.15 ±0.1 V)

Note The reset value changes depending on the reset source.

If the LVIS register is reset by LVI, it is not reset but holds the current value. The value of this register is reset to “0EH” (default value: V_{LVI} = 2.07±0.1 V) if a reset other than by LVI is effected.

Cautions 1. Be sure to clear bits 4 to 7 to “0”.

2. Change the LVIS value with either of the following methods.

- **When changing the value after stopping LVI**
  - **<1> Stop LVI (LVION = 0).**
  - **<2> Change the LVIS register.**
  - **<3> Set to the mode used as an interrupt (LVIMD = 0).**
  - **<4> Mask LVI interrupts (LVIMK = 1).**
  - **<5> Enable LVI operation (LVION = 1).**
  - **<6> Before cancelling the LVI interrupt mask (LVIMK = 0), clear an LVIIF flag with software because it may be set when LVI operation is enabled.**
- **When changing the value after setting to the mode used as an interrupt (LVIMD = 0)**
  - **<1> Mask LVI interrupts (LVIMK = 1).**
  - **<2> Set to the mode used as an interrupt (LVIMD = 0).**
  - **<3> Change the LVIS register.**
  - **<4> Before cancelling the LVI interrupt mask (LVIMK = 0), clear an LVIIF flag with software because it may be set when the LVIS register is changed.**

3. When an input voltage from the external input pin (EXLVI) is detected, the detection voltage (V_{EXLVI}) is fixed. Therefore, setting of LVIS is not necessary.

4. To read LVIM after writing this register, secure the time of one or more clock.

(3) Port mode register 12 (PM12)

When using the P120/EXLVI/INTP0 pin for external low-voltage detection potential input, set PM120 to 1. At this time, the output latch of P120 may be 0 or 1.

PM12 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 19-4. Format of Port Mode Register 12 (PM12)

Address: FFF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	1	1	PM120

PM120	P120 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

19.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

(1) Used as reset (LVIMD = 1)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), generates an internal reset signal when $V_{DD} < V_{LVI}$, and releases internal reset when $V_{DD} \geq V_{LVI}$.
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI} = 1.21 V ±0.1 V), generates an internal reset signal when EXLVI < V_{EXLVI}, and releases internal reset when EXLVI ≥ V_{EXLVI}.

Remark The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage (V_{POR} = 1.61 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.2 V). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V).

(2) Used as interrupt (LVIMD = 0)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}). When V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$), generates an interrupt signal (INTLVI).
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI} = 1.21 V ±0.1 V). When EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}), generates an interrupt signal (INTLVI).

While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

Remark LVIMD: Bit 1 of low-voltage detection register (LVIM)
LVISEL: Bit 2 of LVIM

19.4.1 When used as reset

(1) When detecting level of supply voltage (V_{DD})

(a) When LVI default start function stopped is set (LVIOFF = 1)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).

<3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<5> Use software to wait for the following periods of time (Total 210 μs).

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

<6> Wait until it is checked that (supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.

<7> Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected).

Figure 19-5 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

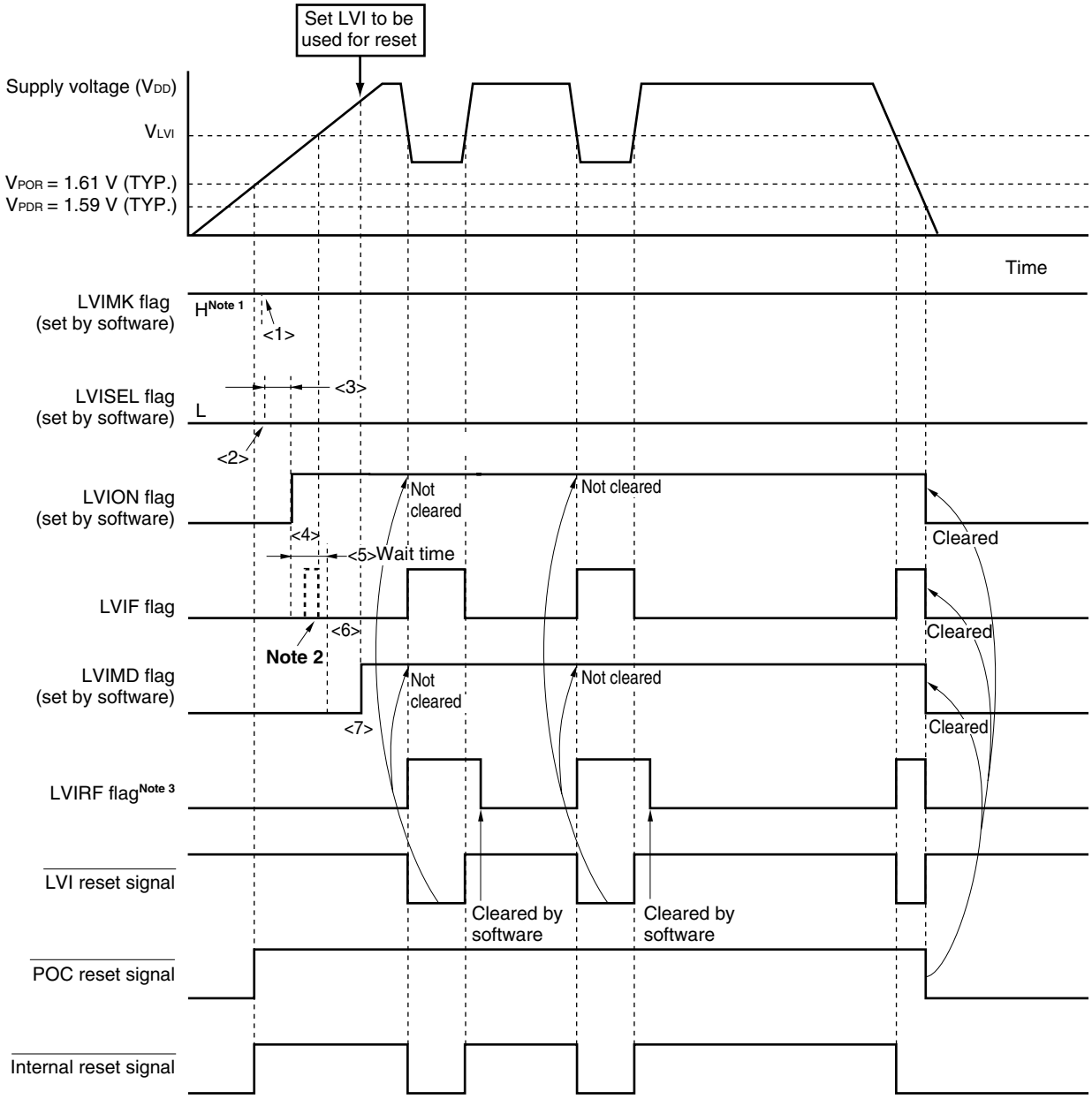
Cautions 1. Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.

2. If supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}) when LVIMD is set to 1, an internal reset signal is not generated.

- When stopping operation

Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

Figure 19-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The LVIIF flag of the interrupt request flag registers and the LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 17 RESET FUNCTION**.

Remarks 1. <1> to <7> in Figure 19-5 above correspond to <1> to <7> in the description of "When starting operation" in **19.4.1 (1) (a) When LVI default start function stopped is set (LVIOFF = 1).**

2. VPOR: POC power supply rise detection voltage
VPDR: POC power supply fall detection voltage

(b) When LVI default start function enabled is set (LVIOFF = 0)

- When starting operation

  Start in the following initial setting state.

  - Set bit 7 (LVION) of LVIM to 1 (enables LVI operation)
  - Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD}))
  - Set the low-voltage detection level selection register (LVIS) to 0EH (default value: V_{LVI} = 2.07 V ±0.1 V).
  - Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected)
  - Set bit 0 (LVIF) of LVIM to 0 (“Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})”)

  Figure 19-6 shows the timing of the internal reset signal generated by the low-voltage detector.

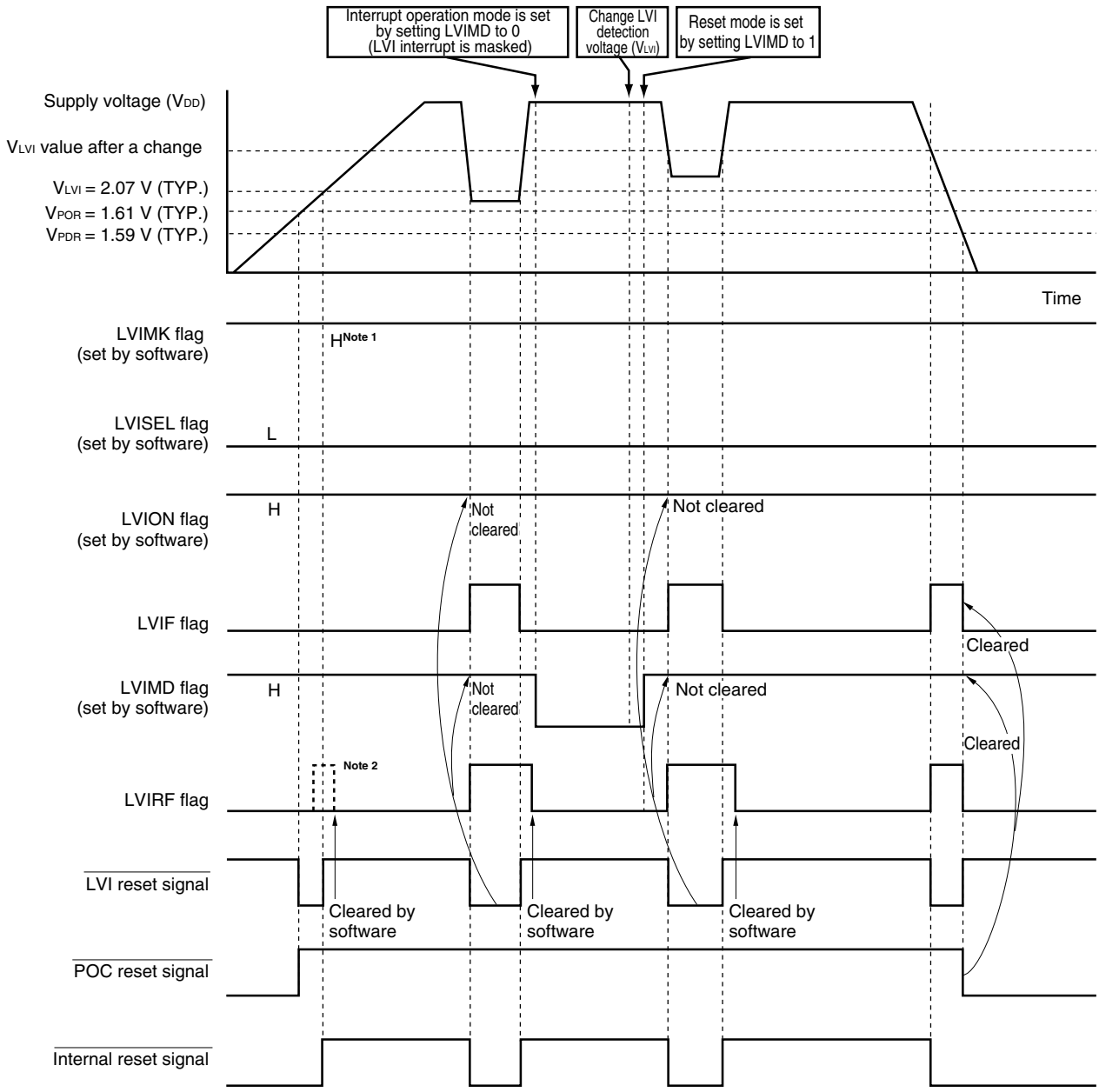
- When stopping operation

  Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

Caution Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.**
  This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

Figure 19-6. Timing of Low-Voltage Detector Internal Reset Signal Generation
(Bit: LVISEL = 0, Option Byte: LVIOFF = 0)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. LVIRF is bit 0 of the reset control flag register (RESF).
When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform.
For details of RESF, see **CHAPTER 17 RESET FUNCTION**.

Remark V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(2) When detecting level of input voltage from external input pin (EXLVI)

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for the following periods of time (Total 210 μs).
    - Operation stabilization time (10 μs (MAX.))
    - Minimum pulse width (200 μs (MIN.))
  - <5> Wait until it is checked that (input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.))) by bit 0 (LVIF) of LVIM.
  - <6> Set bit 1 (LVIMD) of LVIM to 1 (generates reset signal when the level is detected).

  Figure 19-7 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <6> above.

  Cautions 1. Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.

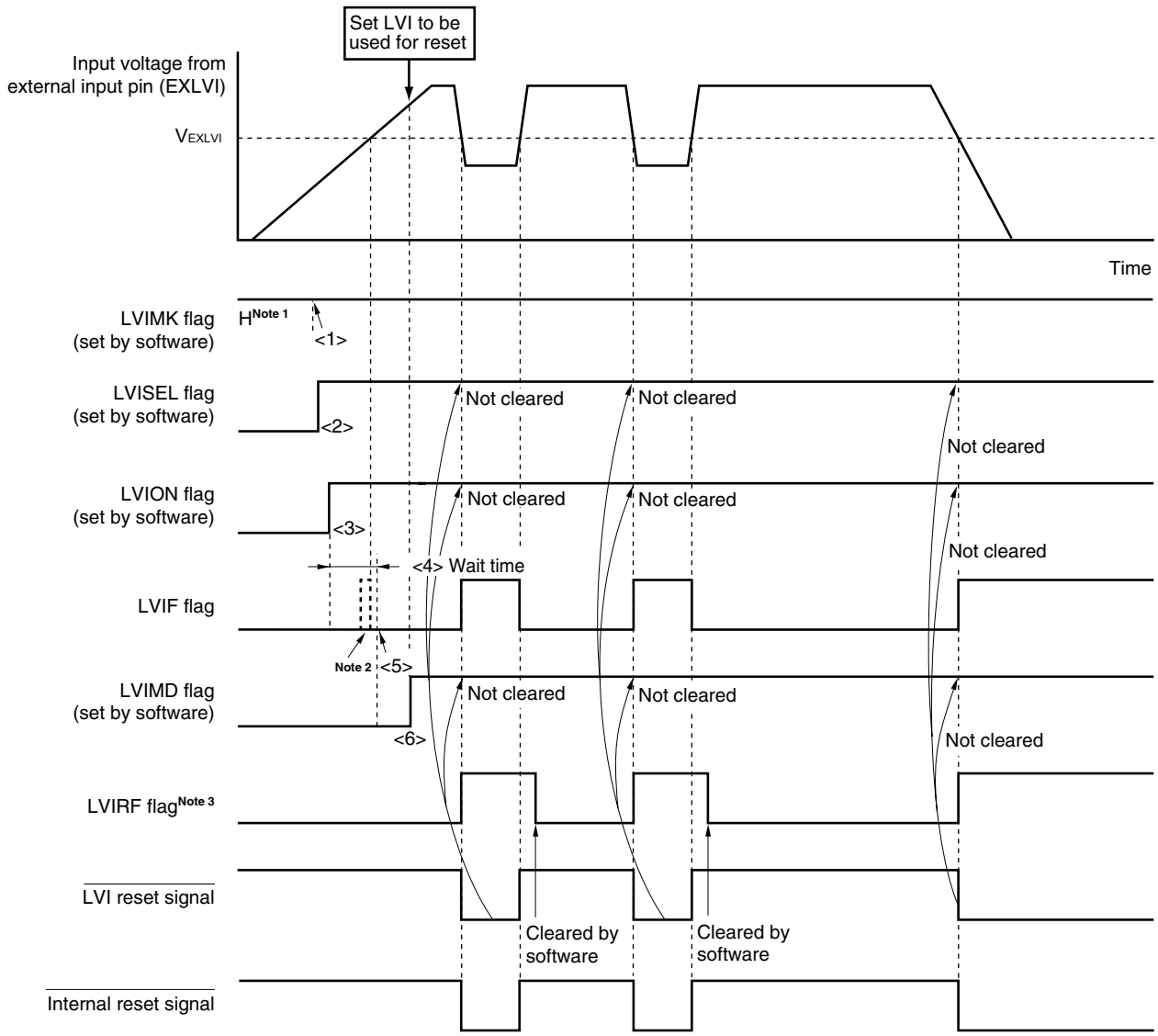
  2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.

  3. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation

  Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

Figure 19-7. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The LVIIF flag of the interrupt request flag registers and the LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 17 RESET FUNCTION**.

Remark <1> to <6> in Figure 19-7 above correspond to <1> to <6> in the description of "When starting operation" in **19.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

19.4.2 When used as interrupt

(1) When detecting level of supply voltage (V_{DD})

(a) When LVI default start function stopped is set (LVIOFF = 1)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).
Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).

<3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<5> Use software to wait for the following periods of time (Total 210 μs).
- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

<6> Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, at bit 0 (LVIF) of LVIM.

<7> Clear the interrupt request flag of LVI (LVIIF) to 0.

<8> Release the interrupt mask flag of LVI (LVIMK).

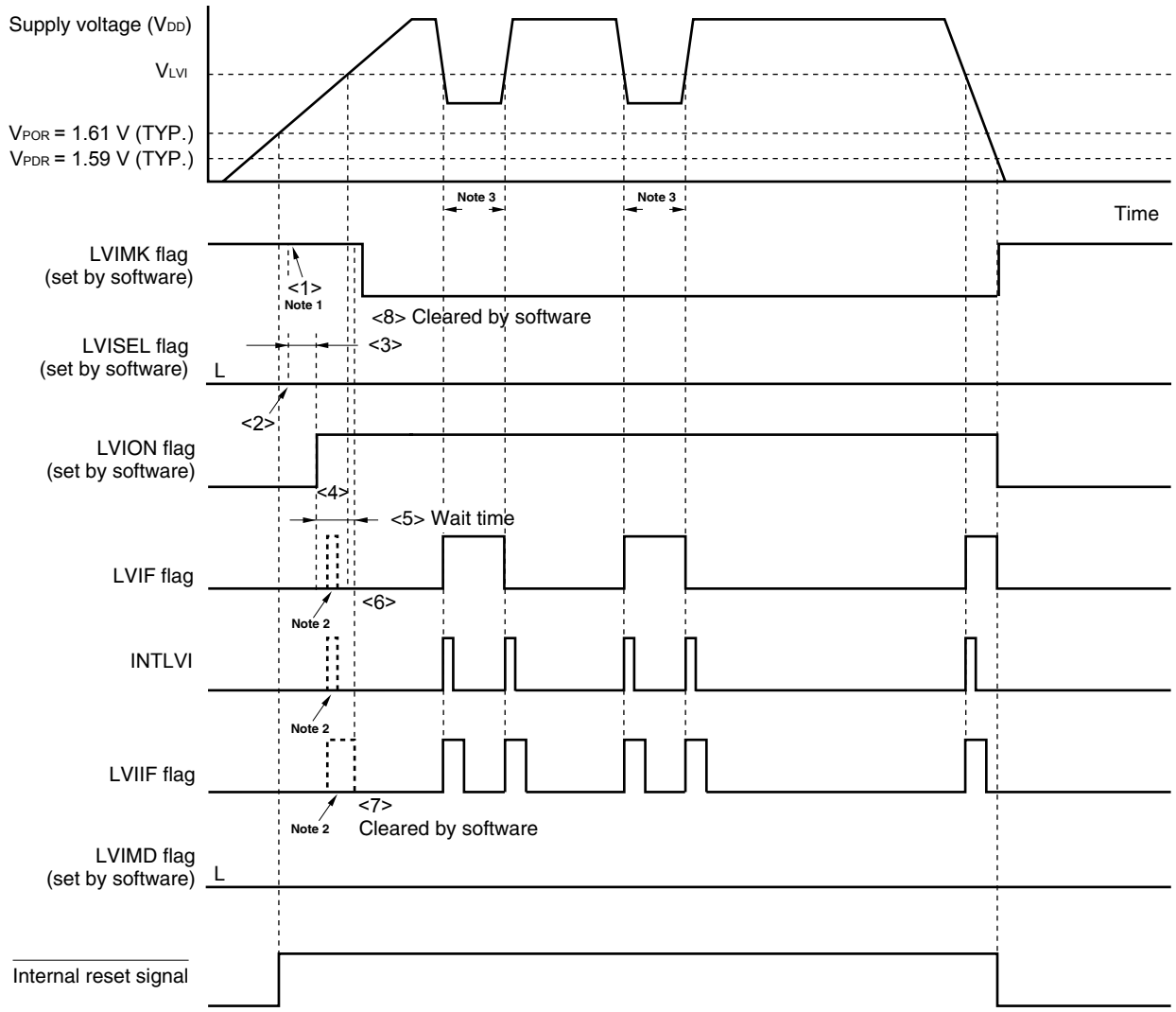
<9> Execute the EI instruction (when vector interrupts are used).

Figure 19-8 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <8> above.

- When stopping operation

Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

Figure 19-8. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVI operation is disabled when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

Remarks 1. <1> to <8> in Figure 19-8 above correspond to <1> to <8> in the description of "When starting operation" in **19.4.2 (1) (a) When LVI default start function stopped is set (LVIOFF = 1).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(b) When LVI default start function enabled is set (LVIOFF = 0)

- When starting operation

<1> Start in the following initial setting state.

- Set bit 7 (LVION) of LVIM to 1 (enables LVI operation)
- Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD}))
- Set the low-voltage detection level selection register (LVIS) to 0EH (default value: V_{LVI} = 2.07 V ±0.1 V).
- Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected)
- Set bit 0 (LVIF) of LVIM to 0 (Detects falling edge “Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})”)

<2> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).

<3> Release the interrupt mask flag of LVI (LVIMK).

<4> Execute the EI instruction (when vector interrupts are used).

Figure 19-9 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <3> above.

- When stopping operation

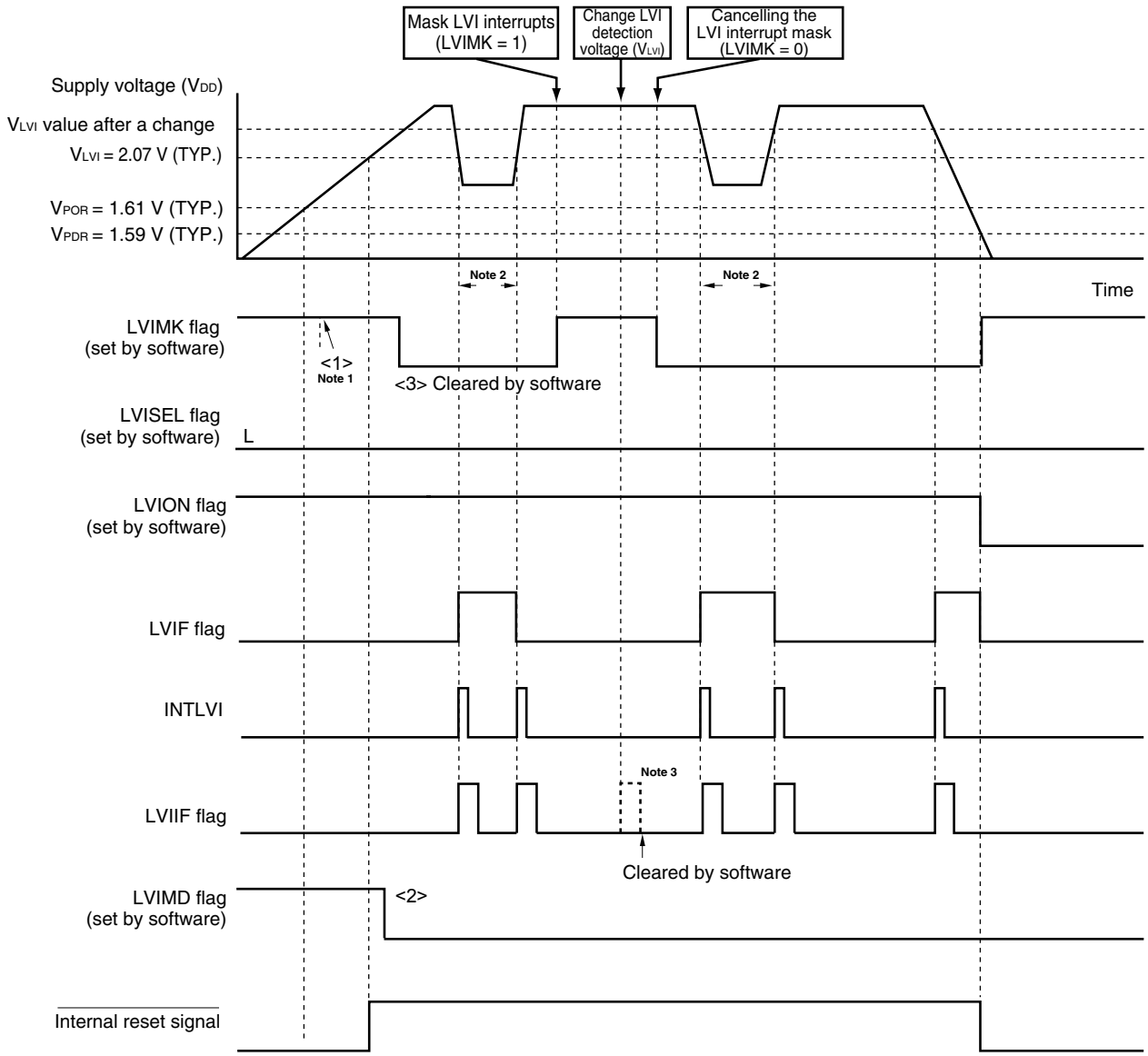
Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

Cautions 1. Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.**
  This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

2. When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform.
For details of RESF, see CHAPTER 17 RESET FUNCTION.

Figure 19-9. Timing of Low-Voltage Detector Interrupt Signal Generation
(Bit: LVISEL = 0, Option Byte: LVIOFF = 0)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. If LVI operation is disabled when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

3. The LVIIF flag may be set when the LVI detection voltage is changed.

Remarks 1. <1> to <3> in Figure 19-9 above correspond to <1> to <3> in the description of "When starting operation" in **19.4.2 (1) (b) When LVI default start function enabled is set (LVIOFF = 0).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(2) When detecting level of input voltage from external input pin (EXLVI)

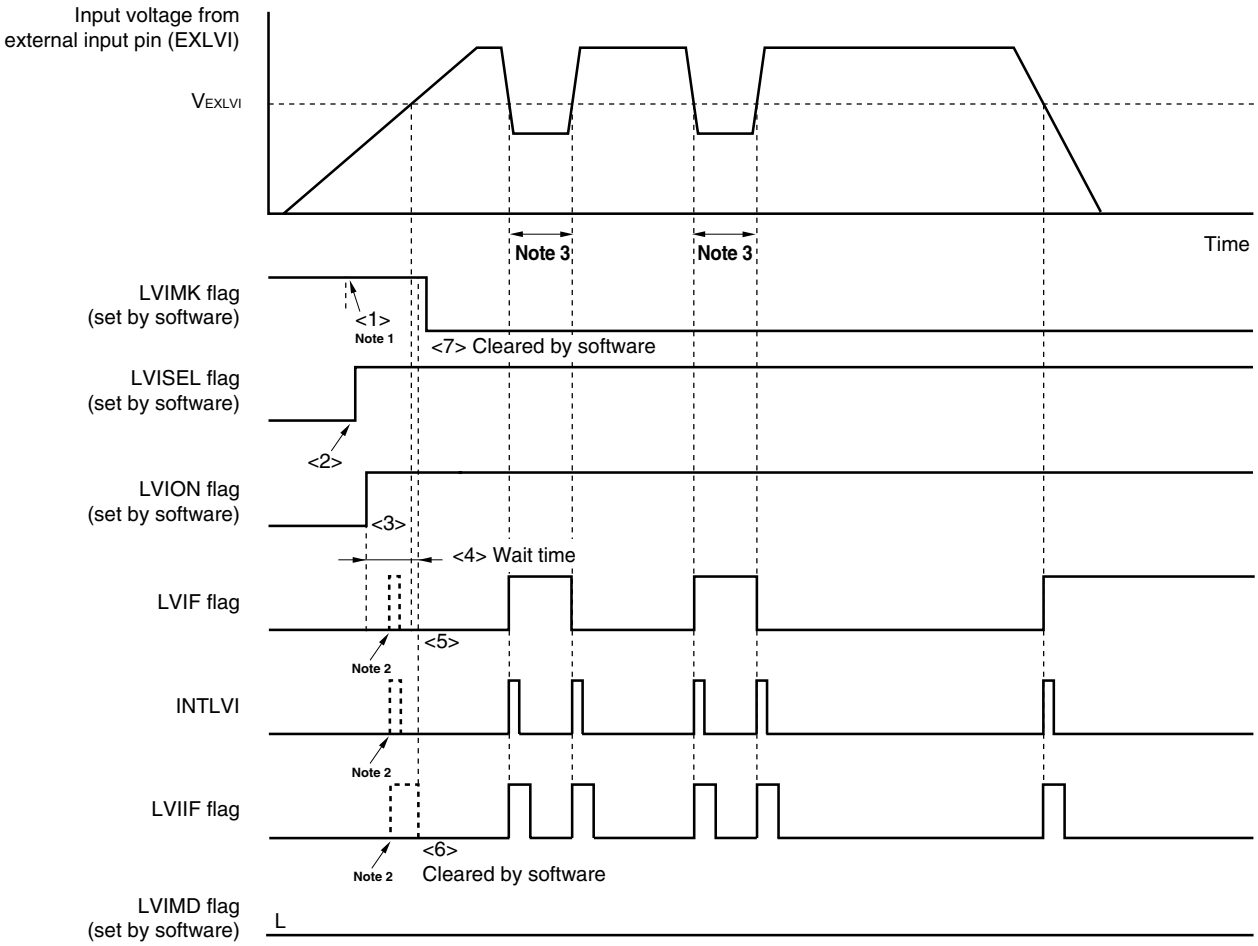
- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
    Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for the following periods of time (Total 210 μs).
    - Operation stabilization time (10 μs (MAX.))
    - Minimum pulse width (200 μs (MIN.))
  - <5> Confirm that "input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.))" when detecting the falling edge of EXLVI, or "input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI} = 1.21 V (TYP.))" when detecting the rising edge of EXLVI, at bit 0 (LVIF) of LVIM.
  - <6> Clear the interrupt request flag of LVI (LVIIF) to 0.
  - <7> Release the interrupt mask flag of LVI (LVIMK).
  - <8> Execute the EI instruction (when vector interrupts are used).

Figure 19-10 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

Caution Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

Figure 19-10. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVI operation is disabled when the input voltage of external input pin (EXLVI) is less than or equal to the detection voltage (V_{EXLVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

Remark <1> to <7> in Figure 19-10 above correspond to <1> to <7> in the description of "When starting operation" in **19.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**.

19.5 Cautions for Low-Voltage Detector

(1) Measures method when supply voltage (V_{DD}) frequently fluctuates in the vicinity of the LVI detection voltage (V_{LVI})

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

Operation example 1: When used as reset

The system may be repeatedly reset and released from the reset status.

The time from reset release through microcontroller operation start can be set arbitrarily by the following action.

<Action>

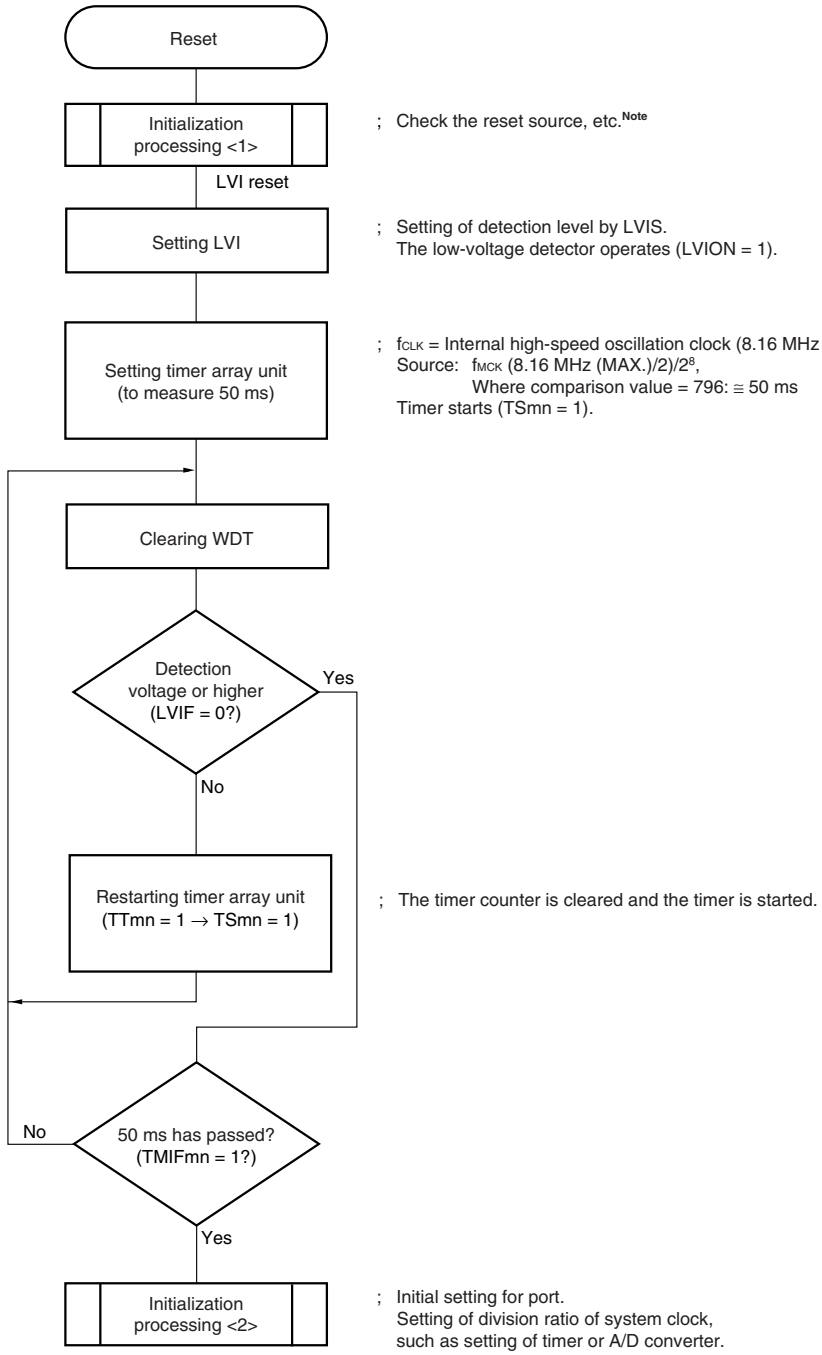
After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 19-11**).

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Figure 19-11. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



Note A flowchart is shown on the next page.

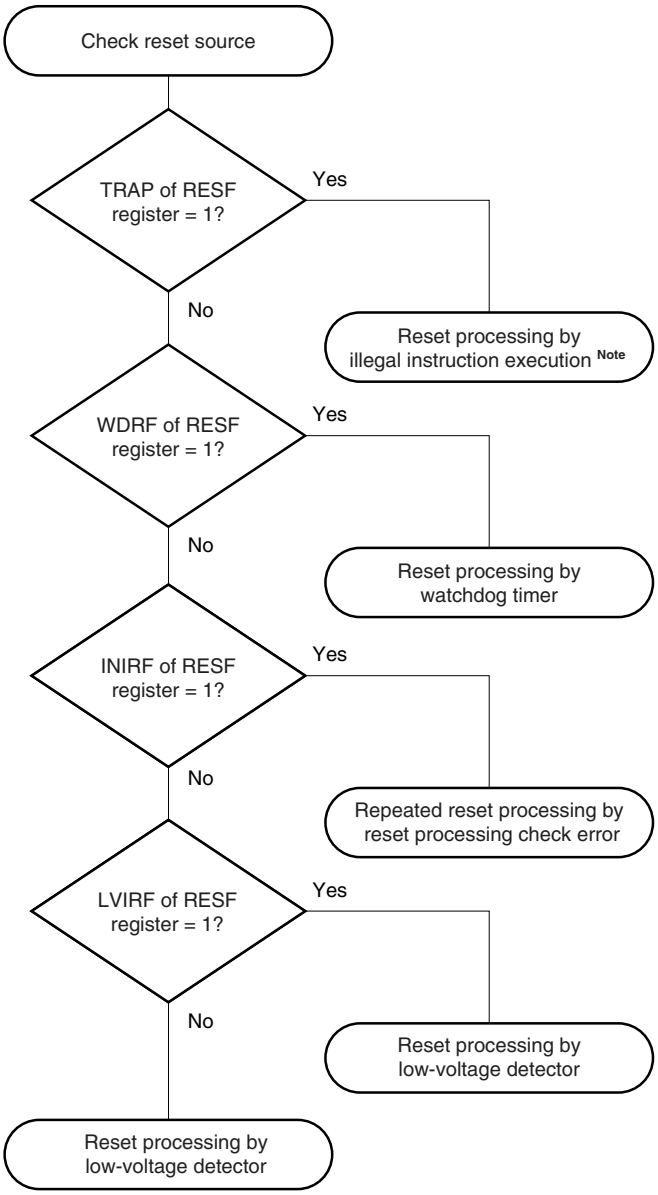
Remarks 1. If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

2. m = 0, 1, n = 0 to 7, mn = 00 to 07, 10 to 13

Figure 19-11. Example of Software Processing After Reset Release (2/2)

• Checking reset source



Note 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.

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Operation example 2: When used as interrupt

Interrupt requests may be generated frequently.

Take the following action.

<Action>

Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 1 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

For a system with a long supply voltage fluctuation period near the LVI detection voltage, take the above action after waiting for the supply voltage fluctuation time.

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

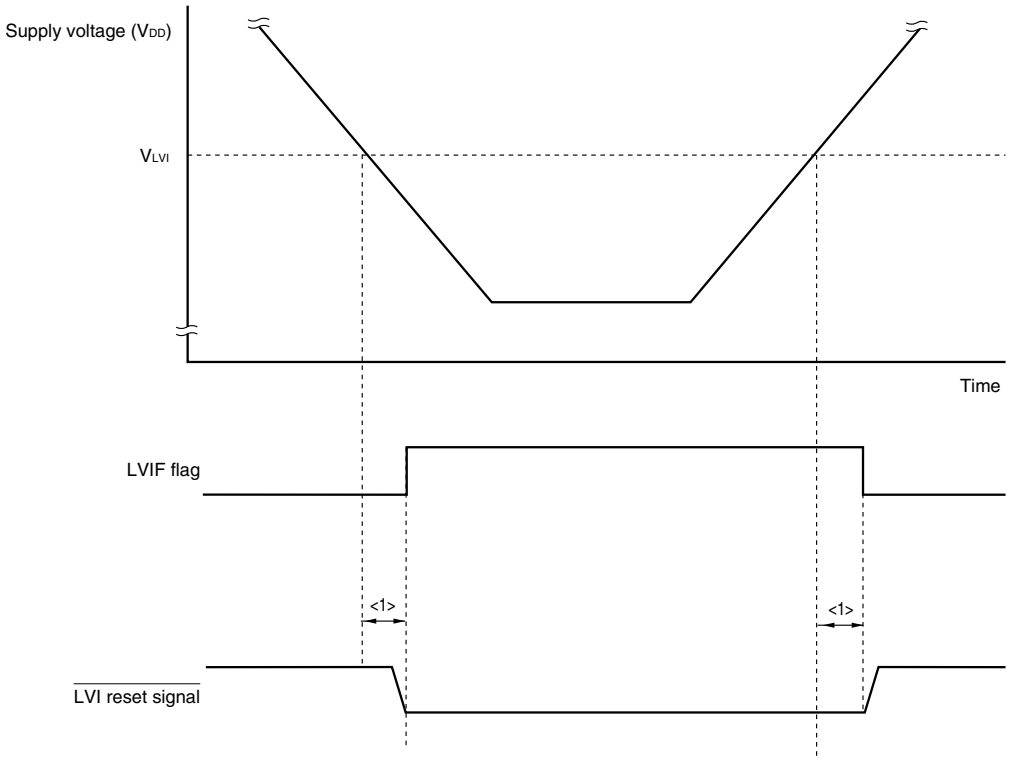
- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

(2) Delay from the time LVI reset source is generated until the time LVI reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVI detection voltage (V_{LVI}) until the time LVI reset has been generated.

In the same way, there is also some delay from the time LVI detection voltage (V_{LVI}) ≤ supply voltage (V_{DD}) until the time LVI reset has been released (see **Figure 19-12**).

Figure 19-12. Delay from the time LVI reset source is generated until the time LVI reset has been generated or released



<1>: Minimum pulse width (200 μs (MIN.))

CHAPTER 20 REGULATOR

20.1 Regulator Overview

The μPD78F8040, 78F8041, 78F8042, 78F8043 contain 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}/EV_{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 2.4 V (typ.), and in the low consumption current mode, 1.8 V (typ.).

20.2 Registers Controlling Regulator

(1) Regulator mode control register (RMC)

This register sets the output voltage of the regulator.

RMC is set with an 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 20-1. Format of Regulator Mode Control Register (RMC)

Address: F00F4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMC								

RMC[7:0]	Control of output voltage of regulator
5AH	Fixed to low consumption current mode (1.8 V)
00H	Switches normal current mode (2.4 V) and low consumption current mode (1.8 V) according to the condition (refer to **Table 20-1**)
Other than above	Setting prohibited

Caution 1. When using the setting fixed to the low consumption current mode, the RMC register can be used in the following cases.

<When X1 clock is selected as the CPU clock> $f_X \leq 5$ MHz and $f_{CLK} \leq 1$ MHz

<When the internal high-speed oscillation clock, or external input clock are selected for the CPU clock> $f_{CLK} \leq 1$ MHz

(**Caution** is given on the next page.)

Caution 2. A wait is required to change the operation speed mode control register (OSMC) after changing the RMC register. Wait for 3.5 ms by software when setting to low consumption current mode and 10 μs when setting to normal current mode, as described in the procedure shown below.

- **When setting to low consumption current mode**
  - **<1> Select a frequency of 1 MHz for f_{CLK}.**
  - **<2> Set RMC to 5AH (set the regulator to low consumption current mode).**
  - **<3> Wait for 3.5 ms.**
  - **<4> Set FLPC and FSEL of OSMC to 1 and 0, respectively.**

- **When setting to normal current mode**
  - **<1> Set RMC to 00H (set the regulator to normal current mode).**
  - **<2> Wait for 10 μs.**
  - **<3> Change FLPC and FSEL of OSMC.**
  - **<4> Change the f_{CLK} frequency.**

Table 20-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
Low consumption current mode	1.8 V	In STOP mode (except during OCD mode)
Normal current mode	2.4 V	Other than above

CHAPTER 21 OPTION BYTE

21.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the μPD78F8040, 78F8041, 78F8042, 78F8043 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.

Caution Be sure to set FFH to 000C2H (000C2H/010C2H when the boot swap operation is used).

21.1.1 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 LVI upon reset release (upon power application)
  - LVI is ON or OFF by default upon reset release (reset by $\overline{\text{RESET}}$ pin excluding LVI, POC, WDT, or illegal instructions).
- ○ Setting of internal high-speed oscillator frequency
  - Select from 1 MHz, 8 MHz or 20 MHz.

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

- ○ Be sure to set FFH, as these addresses are reserved areas.

Caution Set FFH to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

21.1.2 On-chip debug option byte (000C3H/ 010C3H)

○ Control of on-chip debug operation
- On-chip debug operation is disabled or enabled.

○ Handling of data of flash memory in case of failure in on-chip debug security ID authentication
- Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

Caution Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

21.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 21-1. Format of User Option Byte (000C0H/010C0H) (1/2)

Address: 000C0H/010C0H[Note 1]

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% 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} = 34.5 kHz (MAX.))
0	0	0	$2^7/f_{IL}$ (3.71 ms)
0	0	1	$2^8/f_{IL}$ (7.42 ms)
0	1	0	$2^9/f_{IL}$ (14.84 ms)
0	1	1	$2^{10}/f_{IL}$ (29.68 ms)
1	0	0	$2^{12}/f_{IL}$ (118.72 ms)
1	0	1	$2^{14}/f_{IL}$ (474.90 ms)
1	1	0	$2^{15}/f_{IL}$ (949.80 ms)
1	1	1	$2^{17}/f_{IL}$ (3799.19 ms)

Figure 21-1. Format of User Option Byte (000C0H/010C0H) (2/2)

Address: 000C0H/010C0H[Note 1]

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

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 WINDOW1 and WINDOW0.

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}: Internal low-speed oscillation clock frequency

Figure 21-2. Format of User Option Byte (000C1H/010C1H)

Address: 000C1H/010C1H[Note 1]

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2	FRQSEL1	LVIOFF

FRQSEL2	FRQSEL1	Internal high-speed oscillator frequency
0	1	8 MHz/20 MHz [Note 2]
1	0	1 MHz [Note 3]
Other than the above		Setting prohibited

LVIOFF	Setting of LVI on power application
0	LVI is ON by default (LVI default start function enabled) upon reset release (upon power application)
1	LVI is OFF by default (LVI default start function stopped) upon reset release (upon power application)

Notes 1. Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

2. When 8 MHz or 20 MHz has been selected, the 8 MHz internal high-speed oscillator automatically starts oscillating after reset release. To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1 with $V_{DD} \geq 3.0$ V. The circuit cannot be changed to a 1 MHz internal high-speed oscillator while the microcontroller operates.

3. When 1 MHz has been selected, the microcontroller operates on the 1 MHz internal high-speed oscillator after reset release. The circuit cannot be changed to an 8 MHz or 20 MHz internal highspeed oscillator while the microcontroller operates.

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

Cautions 1. Be sure to set bits 7 to 3 to "1".

2. Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.**
  This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

Figure 21-3. Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H[Note]

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1

Note Be sure to set FFH to 000C2H, as these addresses are reserved areas. Also set FFH to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

21.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 21-4. 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 debug operation. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debug operation. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Note Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

Caution Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

21.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the RA78K0R or PM+ 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.

See the RA78K0R Assembler Package User's Manual for how to set the linker option.

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 2^10/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          0FBH    ; Select 8 MHz or 20 MHz for internal high-speed oscillator
                          ; Stops LVI default start function
      DB          0FFH    ; Reserved area
      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          0FBH    ; Select 8 MHz or 20 MHz for internal high-speed oscillator
                          ; Stops LVI default start function
      DB          0FFH    ; Reserved area
      DB          85H     ; Enables on-chip debug operation, does not erase flash memory
                          ; data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 010C0H to 010C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 22 FLASH MEMORY

The μPD78F8040, 78F8041, 78F8042, 78F8043 incorporate the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

22.1 Writing with Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the μPD78F8040, 78F8041, 78F8042, 78F8043.

- PG-FP5, FL-PR5
- QB-MINI2

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 μPD78F8040, 78F8041, 78F8042, 78F8043 have 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 μPD78F8040, 78F8041, 78F8042, 78F8043 are mounted on the target system.

Remark FL-PR5 and FA series is product of Naito Densei Machida Mfg. Co., Ltd.

<R> **Table 22-1. Wiring Between μPD78F8040, 78F8041, 78F8042, 78F8043 and Dedicated Flash Memory Programmer**

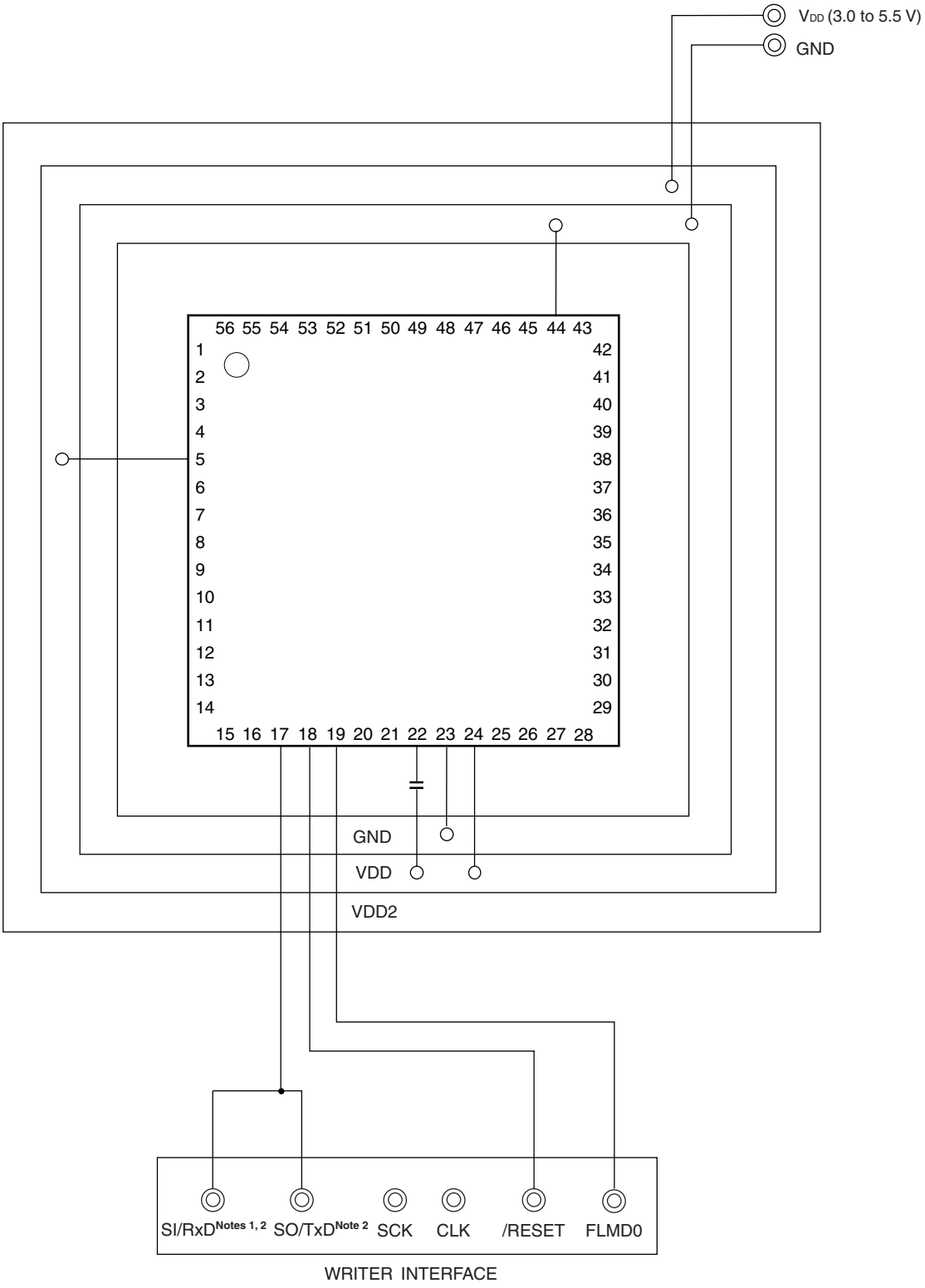
Pin Configuration of Dedicated Flash Memory Programmer			Pin Name	Pin No. (for QFN package products)	Pin No. (for FBGA package products)
Signal Name	I/O	Pin Function			
SI/RxD Notes 1, 2	Input	Receive signal	TOOL0/P40	17	L5
SO/TxD Note 2	Output	Transmit signal			
$\overline{SCK}$	Output	Transfer clock	–	–	–
CLK	Output	Clock output	–	–	–
/RESET	Output	Reset signal	$\overline{RESET}$	18	K5
FLMD0	Output	Mode signal	FLMD0	19	K6
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}/EV_{DD}	24	E5, F5
			AV_{REF}	5	L1
GND	–	Ground	V_{SS}/EV_{SS}	23	G5, H5
			AV_{SS}	44	M1

Notes 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.

2. Connect SI/RxD or SO/TxD when using QB-MINI2.

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

Figure 22-1. Example of Wiring Adapter for Flash Memory Writing (QFN Package)



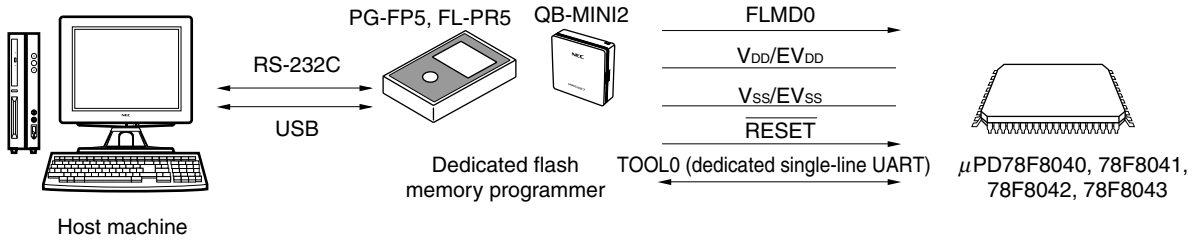
Notes 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.

2. Connect SI/RxD or SO/TxD when using QB-MINI2.

22.2 Programming Environment

The environment required for writing a program to the flash memory of the μPD78F8040, 78F8041, 78F8042, 78F8043 are illustrated below.

Figure 22-2. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

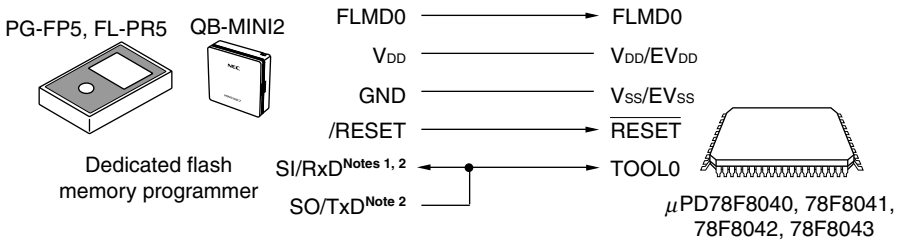
To interface between the dedicated flash memory programmer and the μPD78F8040, 78F8041, 78F8042, 78F8043, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

22.3 Communication Mode

Communication between the dedicated flash memory programmer and the μPD78F8040, 78F8041, 78F8042, 78F8043 are established by serial communication using the TOOL0 pin via a dedicated single-line UART of the μPD78F8040, 78F8041, 78F8042, 78F8043.

Transfer rate: 115,200 bps, 250,000 bps, 500,000 bps, 1 Mbps

Figure 22-3. Communication with Dedicated Flash Memory Programmer



Notes 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.

2. Connect SI/RxD or SO/TxD when using QB-MINI2.

The dedicated flash memory programmer generates the following signals for the μPD78F8040, 78F8041, 78F8042, 78F8043. See the manual of PG-FP5, FL-PR5, or MINICUBE2 for details.

Table 22-2. Pin Connection

Dedicated Flash Memory Programmer			μPD78F8040, 78F8041, 78F8042, 78F8043	Connection
Signal Name	I/O	Pin Function	Pin Name	
FLMD0	Output	Mode signal	FLMD0	◎
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}/EV_{DD}, AV_{REF}	◎
GND	–	Ground	V_{SS}/EV_{SS}, AV_{SS}	◎
CLK	Output	Clock output	–	×
/RESET	Output	Reset signal	$\overline{RESET}$	◎
SI/RxD Notes 1, 2	Input	Receive signal	TOOL0	◎
SO/TxD Note 2	Output	Transmit signal		
$\overline{SCK}$	Output	Transfer clock	–	×

Notes 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.

2. Connect SI/RxD or SO/TxD when using QB-MINI2.

Remark ◎: Be sure to connect the pin.

×: The pin does not have to be connected.

22.4 Connection of Pins on Board

To write the flash memory on-board, 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.

22.4.1 FLMD0 pin

(1) In flash memory programming mode

Directly connect this pin to a flash memory programmer when data is written by the flash memory programmer. This supplies a writing voltage of the V_{DD} level to the FLMD0 pin.

The FLMD0 pin does not have to be pulled down externally because it is internally pulled down by reset. To pull it down externally, use a resistor of 1 kΩ to 200 kΩ.

(2) In normal operation mode

It is recommended to leave this pin open during normal operation.

The FLMD0 pin must always be kept at the V_{SS} level before reset release but does not have to be pulled down externally because it is internally pulled down by reset. However, pulling it down must be kept selected (i.e., FLMDPUP = "0", default value) by using bit 7 (FLMDPUP) of the background event control register (BECTL) (see **22.5 (1) Back ground event control register**). To pull it down externally, use a resistor of 200 kΩ or smaller.

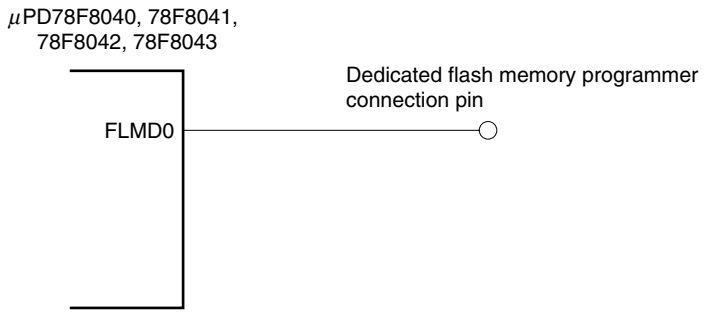
Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS}/EV_{SS} pin.

(3) In self programming mode

It is recommended to leave this pin open when using the self programming function. To pull it down externally, use a resistor of 100 kΩ to 200 kΩ.

In the self programming mode, the setting is switched to pull up in the self programming library.

Figure 22-4. FLMD0 Pin Connection Example



22.4.2 TOOL0 pin

In the flash memory programming mode, connect this pin directly to the dedicated flash memory programmer or pull it up by connecting it to V_{DD}/EV_{DD} via an external resistor.

When on-chip debugging is enabled in the normal operation mode, pull this pin up by connecting it to EV_{DD} via an external resistor, and be sure to keep inputting the V_{DD} level to the TOOL0 pin before reset is released (pulling down this pin is prohibited).

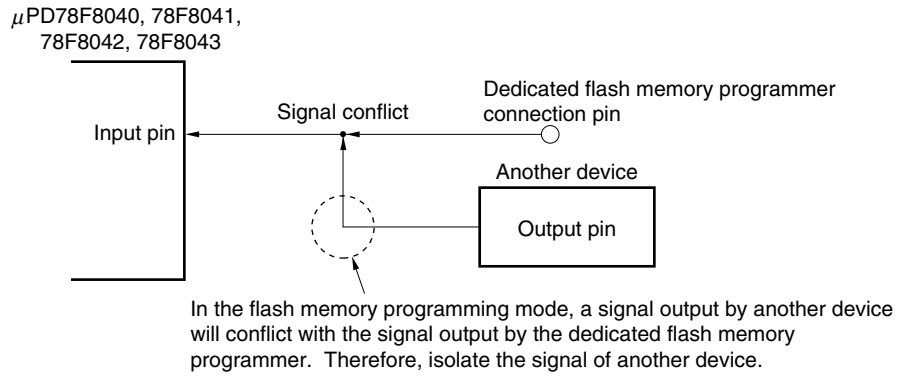
Remark The SAU and IICA pins are not used for communication between the μPD78F8040, 78F8041, 78F8042, 78F8043 and dedicated flash memory programmer, because single-line UART is used.

22.4.3 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer is 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.

Figure 22-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



22.4.4 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 V_{DD} or V_{SS}/EV_{SS} via a resistor.

22.4.5 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.

22.4.6 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 internal high-speed oscillation clock (f_{IH}) is used.

22.4.7 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD}/EV_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS}/EV_{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 using the on-board supply voltage, be sure to connect the V_{DD}/EV_{DD} and V_{SS}/EV_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

Supply the same other power supplies (AV_{REF} and AV_{SS}) as those in the normal operation mode.

22.5 Registers that Control Flash Memory

(1) Background event control register (BECTL)

Even if the FLMD0 pin is not controlled externally, it can be controlled by software with the BECTL register to set the self-programming mode.

However, depending on the processing of the FLMD0 pin, it may not be possible to set the self-programming mode by software. When using BECTL, leaving the FLMD0 pin open is recommended. When pulling it down externally, use a resistor with a resistance of 100 kΩ or more. In addition, in the normal operation mode, use BECTL with the pull down selection. In the self-programming mode, the setting is switched to pull up in the self- programming library.

The BECTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 22-6. Format of Background Event Control Register (BECTL)

Address: FFFBEH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
BECTL	FLMDPUP	0	0	0	0	0	0	0

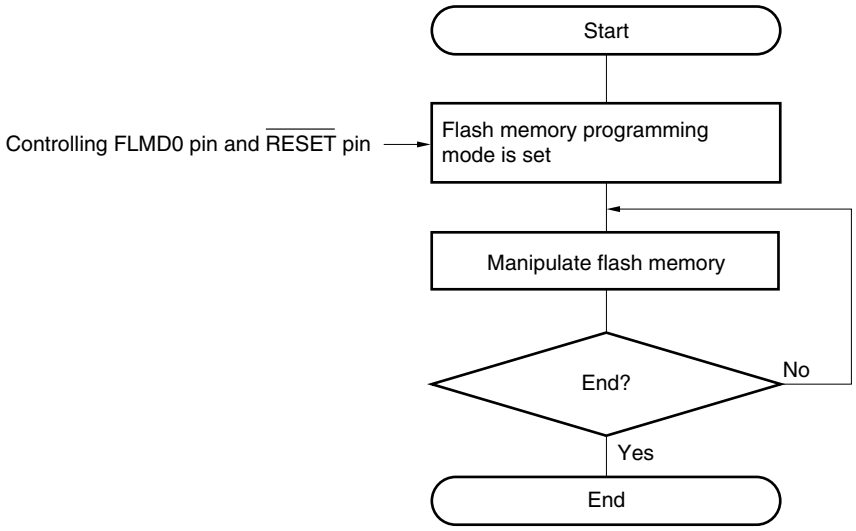
FLMDPUP	Software control of FLMD0 pin
0	Selects pull-down
1	Selects pull-up

22.6 Programming Method

22.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 22-7. Flash Memory Manipulation Procedure



22.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the μPD78F8040, 78F8041, 78F8042, 78F8043 in the flash memory programming mode. To set the mode, set the FLMD0 pin and TOOL0 pin to V_{DD} and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

Figure 22-8. Flash Memory Programming Mode

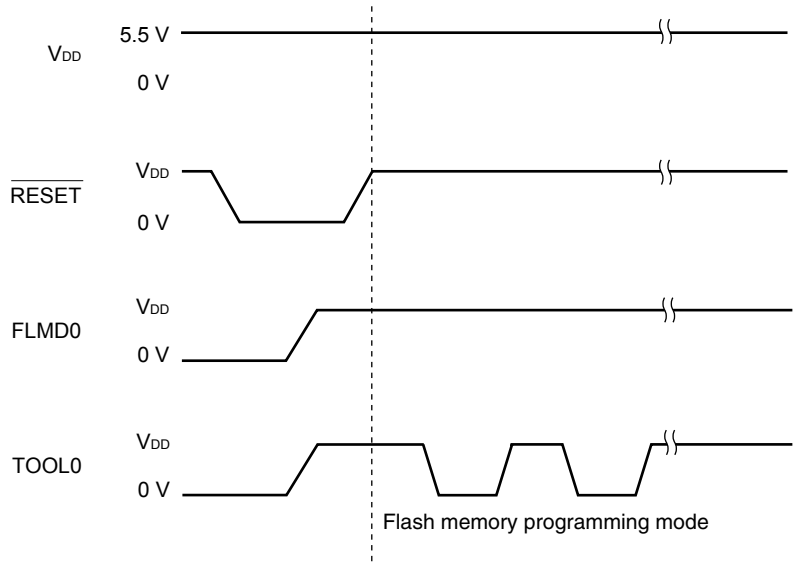


Table 22-3. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0 V	Normal operation mode
V_{DD}	Flash memory programming mode

22.6.3 Selecting communication mode

Communication mode of the μPD78F8040, 78F8041, 78F8042, 78F8043 are as follows.

Table 22-4. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
1-line mode (single-line UART)	UART	115,200 bps, 250,000 bps, 500,000 bps, 1 Mbps	–	–	TOOL0

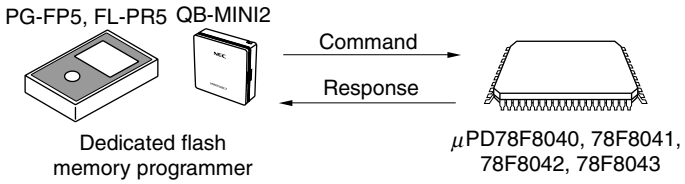
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

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.

22.6.4 Communication commands

The μPD78F8040, 78F8041, 78F8042, 78F8043 communicate with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the μPD78F8040, 78F8041, 78F8042, 78F8043 are called commands, and the signals sent from the μPD78F8040, 78F8041, 78F8042, 78F8043 to the dedicated flash memory programmer are called response.

Figure 22-9. Communication Commands



The flash memory control commands of the μPD78F8040, 78F8041, 78F8042, 78F8043 are listed in the table below. All these commands are issued from the programmer and the μPD78F8040, 78F8041, 78F8042, 78F8043 perform processing corresponding to the respective commands.

Table 22-5. 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	Chip Erase	Erases the entire flash memory.
	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 μPD78F8040, 78F8041, 78F8042, 78F8043 information (such as the part number and flash memory configuration).
	Version Get	Gets the μPD78F8040, 78F8041, 78F8042, 78F8043 firmware version.
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

The μPD78F8040, 78F8041, 78F8042, 78F8043 return a response for the command issued by the dedicated flash memory programmer. The response names sent from the μPD78F8040, 78F8041, 78F8042, 78F8043 are listed below.

Table 22-6. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

22.7 Security Settings

The μPD78F8040, 78F8041, 78F8042, 78F8043 support 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 batch erase (chip erase)
  Execution of the block erase and batch erase (chip erase) commands for entire blocks in the flash memory is prohibited by this setting during on-board/off-board programming. Once execution of the batch erase (chip erase) command is prohibited, all of the prohibition settings (including prohibition of batch erase (chip erase)) can no longer be cancelled.

  Caution After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.

- 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 and block erase commands 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. In addition, execution of the batch erase (chip erase) command.

The batch erase (chip erase), 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.

All the security settings are cleared by executing the batch erase (chip erase) command.

Table 22-7 shows the relationship between the erase and write commands when the μPD78F8040, 78F8041, 78F8042, 78F8043 security function is enabled.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **22.8.2** for details).

Table 22-7. Relationship Between Enabling Security Function and Command

(1) During on-board/off-board programming

Valid Security	Executed Command		
	Batch Erase (Chip Erase)	Block Erase	Write
Prohibition of batch erase (chip erase)	Cannot be erased in batch	Blocks cannot be erased.	Can be performed[Note].
Prohibition of block erase	Can be erased in batch.		Can be performed.
Prohibition of writing			Cannot be performed.
Prohibition of rewriting boot cluster 0	Cannot be erased in batch	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 batch erase (chip 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 batch erase (chip erase)	Blocks can be erased.	Can be performed.
Prohibition of block erase		
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **22.8.2** for details).

Table 22-8. Setting Security in Each Programming Mode

(1) On-board/off-board programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command
Prohibition of writing		
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

(2) Self programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set by using information library.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command during on-board/off-board programming (cannot be disabled during self programming)
Prohibition of writing		
Prohibition of rewriting boot cluster 0		

22.8 Flash Memory Programming by Self-Programming

The μPD78F8040, 78F8041, 78F8042, 78F8043 support 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 μPD78F8040, 78F8041, 78F8042, 78F8043 self-programming library, it can be used to upgrade the program in the field.

If an interrupt occurs during self-programming, self-programming can be temporarily stopped and interrupt servicing can be executed. If an unmasked interrupt request is generated in the EI state, the request branches directly from the self-programming library to the interrupt routine. After the self-programming mode is later restored, self-programming can be resumed. However, the interrupt response time is different from that of the normal operation mode.

Cautions 1. In the self-programming mode, call the self-programming start library (FlashStart).

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.

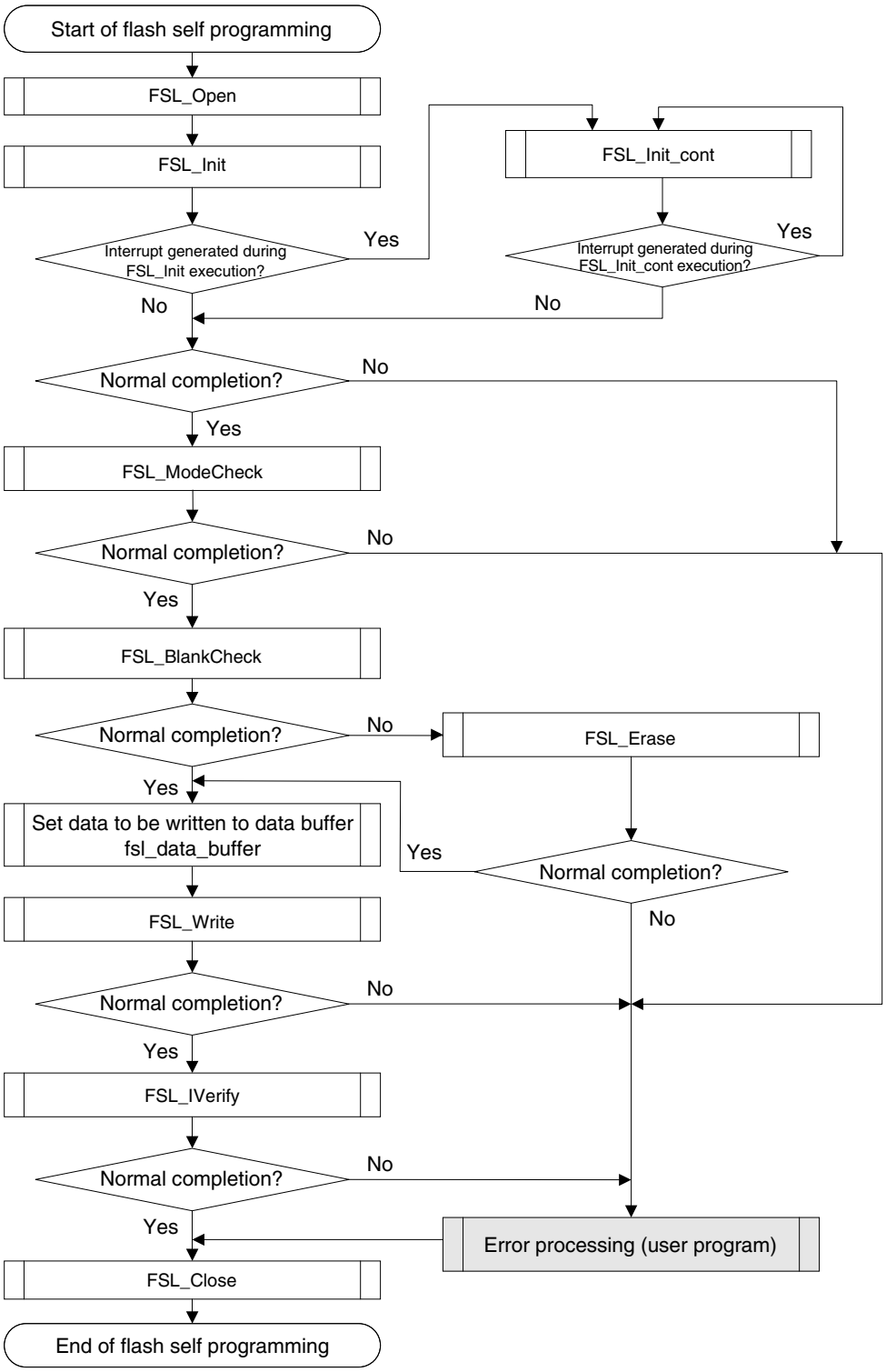
3. Disable DMA operation (DENn = 0) during the execution of self programming library functions.

Remarks 1. For details of the self-programming function and the μPD78F8040, 78F8041, 78F8042, 78F8043 self-programming library, refer to **78K0R Microcontroller Self Programming Library Type02 User's Manual (U19193E).**

2. For details of the time required to execute self programming, see the notes on use that accompany the flash self programming library tool.

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

Figure 22-10. Flow of Self Programming (Rewriting Flash Memory)



Remark For details of the self programming library, refer to **78K0R Microcontroller Self Programming Library Type02 User's Manual (U19193E)**.

22.8.1 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.

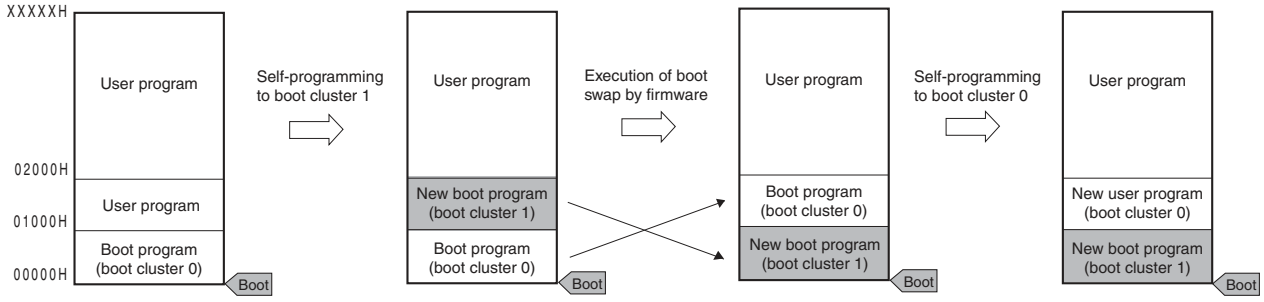
The boot swap function is used to avoid this problem.

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 μPD78F8040, 78F8041, 78F8042, 78F8043, 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 22-11. Boot Swap Function

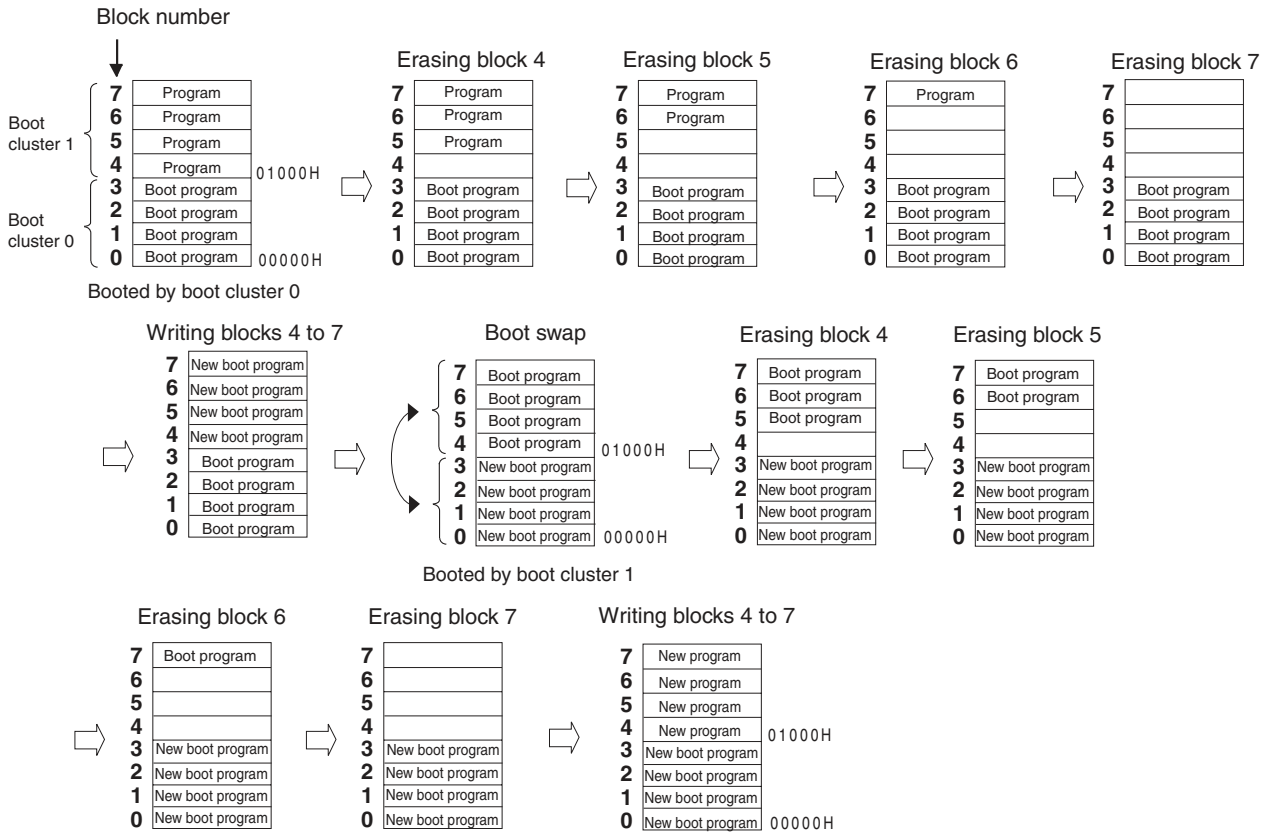


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 22-12. Example of Executing Boot Swapping



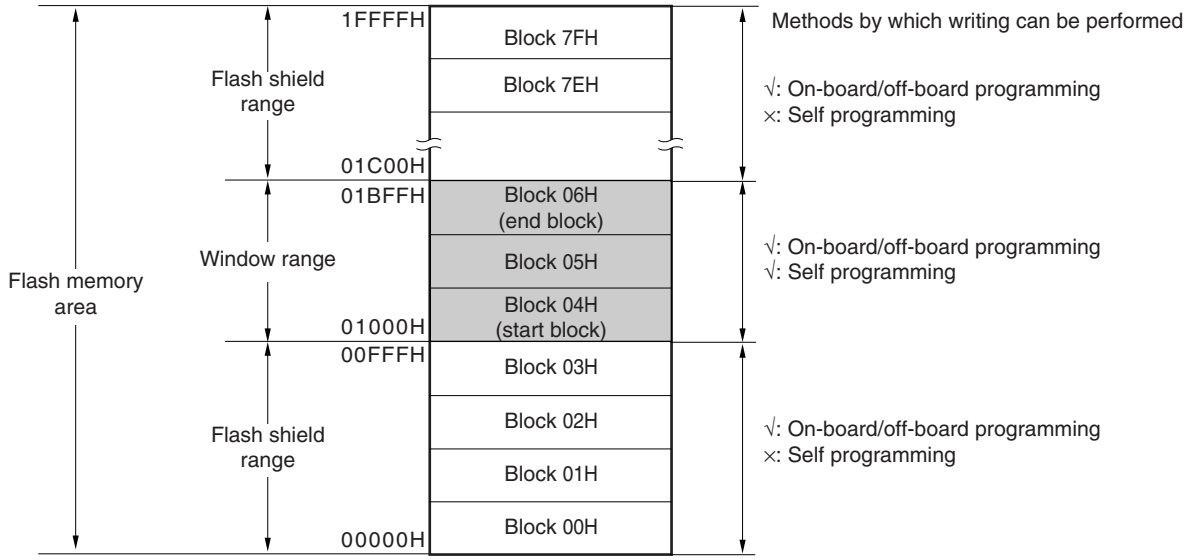
22.8.2 Flash shield window function

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 22-13. Flash Shield Window Setting Example
(Target Devices: μPD78F8043, Start Block: 04H, End Block: 06H)



Caution 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.

Table 22-9. Relationship Between Flash Shield Window Function Setting/Change Methods and Commands

Programming Conditions	Window Range Setting/Change Methods	Execution Commands	
		Block Erase	Write
Self-programming	Specify the starting and ending blocks by the set information library.	Block erasing is enabled only within the window range.	Writing is enabled only within the 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 **22.7 Security Settings** to prohibit writing/erasing during on-board/off-board programming.

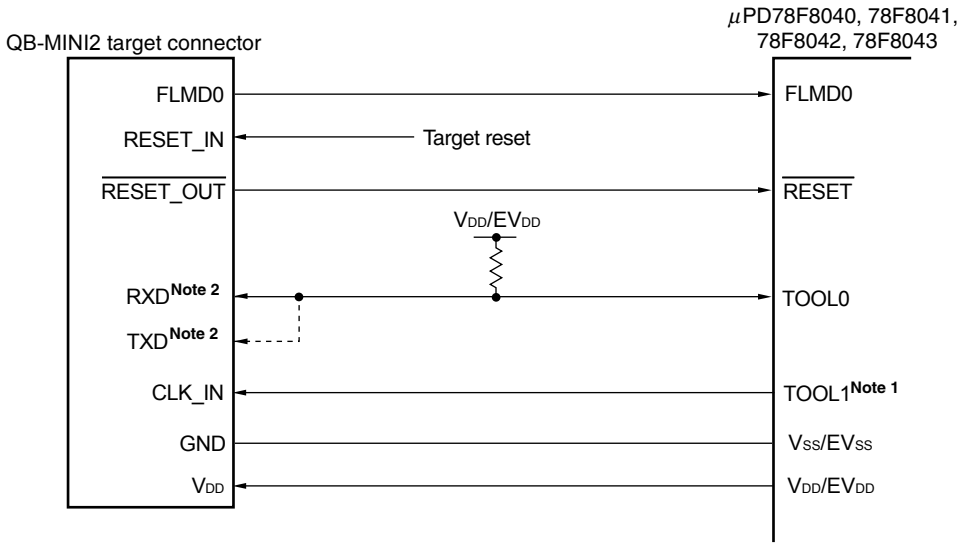
CHAPTER 23 ON-CHIP DEBUG FUNCTION

23.1 Connecting QB-MINI2 to μPD78F8040, 78F8041, 78F8042, 78F8043

The μPD78F8040, 78F8041, 78F8042, 78F8043 use the V_{DD}/EV_{DD}, FLMD0, $\overline{RESET}$, TOOL0, TOOL1[Note 1], and V_{SS}/EV_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2).

Caution The μPD78F8040, 78F8041, 78F8042, 78F8043 have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 23-1. Connection Example of QB-MINI2 and μPD78F8040, 78F8041, 78F8042, 78F8043



Notes 1. Connection is not required for communication in 1-line mode but required for communication in 2-line mode. At this time, perform necessary connections according to **Table 3-3 Connection of Unused Pins** since TOOL1 is an unused pin when QB-MINI2 is unconnected.

2. Connecting the dotted line is not necessary since RXD and TXD are shorted within QB-MIN2. When using the other flash memory programmer, RXD and TXD may not be shorted within the programmer. In this case, they must be shorted on the target system.

Caution When communicating in 2-line mode, a clock with a frequency of half that of the CPU clock frequency is output from the TOOL1 pin. A resistor or ferrite bead can be used as a countermeasure against fluctuation of the power supply caused by that clock.

Remark The FLMD0 pin is recommended to be open for self-programming in on-chip debugging. To pull down externally, use a resistor of 100 kΩ or more.

1-line mode (single line UART) using the TOOL0 pin or 2-line mode using the TOOL0 and TOOL1 pins is used for serial communication. For flash memory programming, 1-line mode is used. 1-line mode or 2-line mode is used for on-chip debugging. Table 23-1 lists the differences between 1-line mode and 2-line mode.

Table 23-1. Differences Between 1-Line Mode and 2-Line Mode

Communication Mode	Flash Memory Programming Function
1-line mode	Available
2-line mode	None

Remark 2-line mode is not used for flash programming, however, even if TOOL1 pin is connected with CLK_IN of QB-MINI2, writing is performed normally with no problem.

23.2 On-Chip Debug Security ID

The μPD78F8040, 78F8041, 78F8042, 78F8043 have an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 21 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 010C3H and 010C4H to 010CDH in advance, because 000C3H, 000C4H to 000CDH and 010C3H, and 010C4H to 010CDH are switched.

For details on the on-chip debug security ID, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

Table 23-2. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

23.3 Securing of User Resources

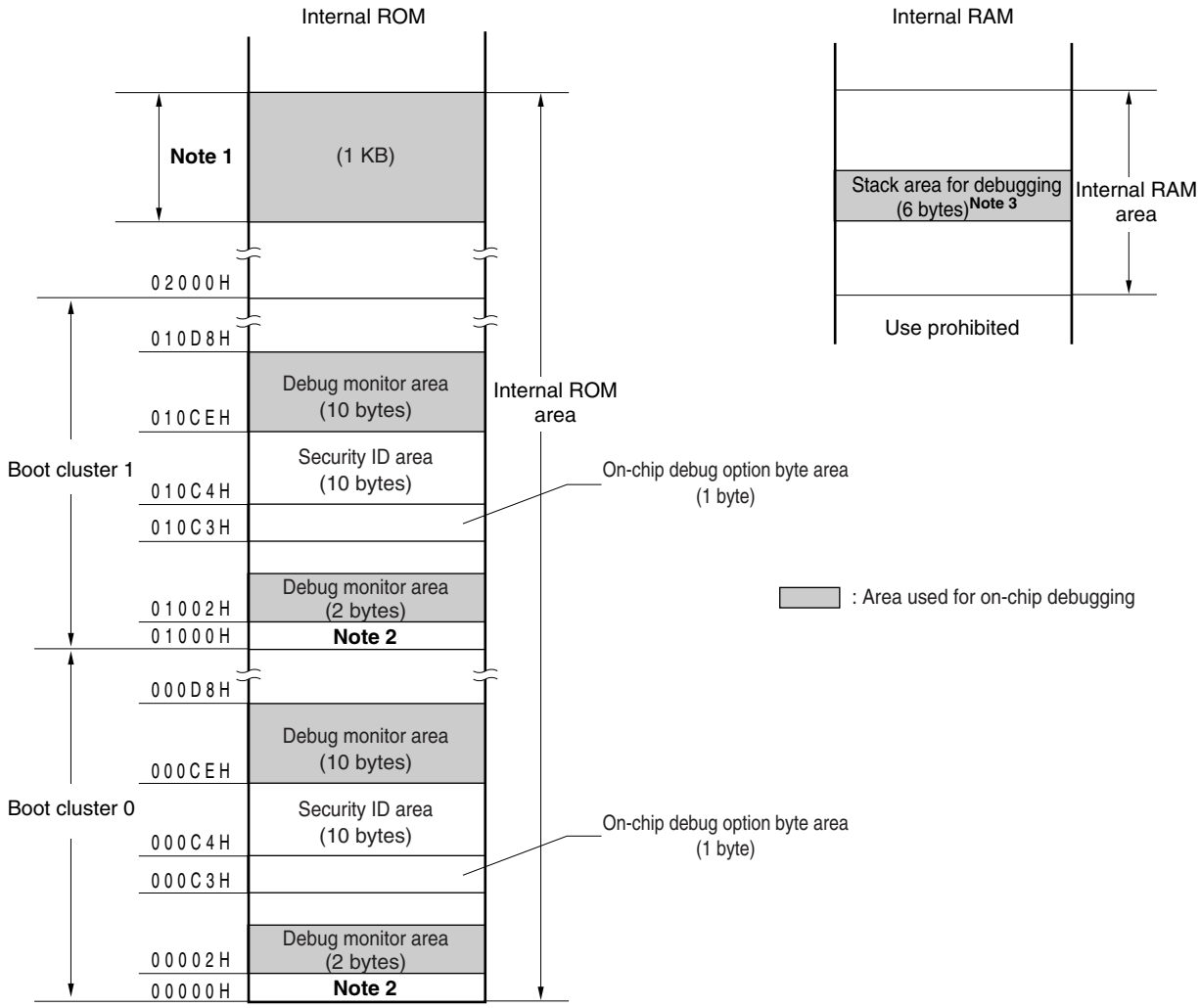
To perform communication between the μPD78F8040, 78F8041, 78F8042, 78F8043 and QB-MINI2, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler RA78K0R or compiler CC78K0R is used, the items can be set by using linker options.

(1) Securing of memory space

The shaded portions in Figure 23-2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 23-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products	Internal ROM	Address
μPD78F8040	32 KB	07C00H to 07FFFH
μPD78F8041	64 KB	0FC00H to 0FFFFH
μPD78F8042	96 KB	17C00H to 17FFFH
μPD78F8043	128 KB	1FC00H to 1FFFFH

2. In debugging, reset vector is rewritten to address allocated to a monitor program.

3. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 6 extra bytes are consumed for the stack area used.

For details of the way to secure of the memory space, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

CHAPTER 24 BCD CORRECTION CIRCUIT

24.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCDADJ register.

24.2 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.

BCDADJ is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

Figure 24-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

24.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCDADJ register.

<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 BCDADJ register.

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 − 52 = 39

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #91H ; <1>`	91H	−	−	−
`SUB A, #52H ; <2>`	3FH	0	1	06H
`SUB A, !BCDADJ ; <3>`	39H	0	0	−

CHAPTER 25 INSTRUCTION SET

This chapter lists the instructions in the 78K0R microcontroller instruction set. For details of each operation and operation code, refer to the separate document **78K0R Microcontrollers Instructions User's Manual (U17792E)**.

Remark The shaded parts of the tables in **Table 25-5 Operation List** indicate the operation or instruction format that is newly added for the 78K0R microcontrollers.

25.1 Conventions Used in Operation List

25.1.1 Operand identifiers and specification methods

Operands are described in the "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, $, $!, [ ], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- $: 8-bit relative address specification
- $!: 16-bit relative address specification
- [ ]: Indirect address specification
- ES:: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, $, $!, [ ], and ES: symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 25-1. Operand Identifiers and Specification Methods

Identifier	Description Method
r 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 symbol (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 5-5 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 5-6 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

25.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 25-2. Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
∨	Exclusive logical sum (exclusive OR)
¯	Inverted data
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

25.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

Table 25-3. Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
×	Set/cleared according to the result
R	Previously saved value is restored

25.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

An interrupt or DMA transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 25-4. Use Example of PREFIX Operation Code

Instruction	Opcode				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	–
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	–	–	–	–
MOV A, ES:[HL]	11H	8BH	–	–	–

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

25.2 Operation List

Table 25-5. Operation List (1/17)

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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 25-5. 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 (fCLK) 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 25-5. 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_{CLK}) 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 25-5. 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_{CLK}) 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 26 ELECTRICAL SPECIFICATIONS

Cautions 1. The μPD78F8040, 78F8041, 78F8042, 78F8043 have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

<R> **2. These are target specifications for the FBGA package products (under development).**

26.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}/EV_{DD}		−0.5 to +6.5	V
	V_{SS}/EV_{SS}		−0.5 to +0.3	V
	AV_{REF}		−0.5 to V_{DD}/EV_{DD} +0.3 Note 1	V
	AV_{SS}		−0.5 to +0.3	V
	V_{DDH}		−0.3 to +40	V
	V_{DD_IO}		−0.3 to +5.5	V
REGC pin input voltage	V_{IREGC}	REGC	−0.3 to 3.6 and −0.3 to V_{DD}/EV_{DD} +0.3 Note 2	V
Input voltage	V_{I1}	P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P65, P67, P120, P121, P142 to P144, EXCLK, $\overline{RESET}$, FLMD0	−0.3 to V_{DD}/EV_{DD} +0.3 Note 1	V
	V_{I2}	P60, P61 (N-ch open-drain)	−0.3 to +6.5	V
	V_{I3}	P26, P27, P150 to P153	−0.3 to AV_{REF} +0.3 and −0.3 to V_{DD}/EV_{DD} +0.3 Note 1	V
Output voltage	V_{O1}	P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P60, P61, P65, P67, P120, P142 to P144	−0.3 to V_{DD}/EV_{DD} +0.3 Note 1	V
	V_{O2}	P26, P27, P150 to P153	−0.3 to AV_{REF} +0.3	V
CQ pin input voltage	V_{AI}	Not supplied	−40 to +40	V
		Supplied	−20 to +40	

Notes 1. Must be 6.5 V or lower.

2. Connect the REGC pin to Vss/EVss via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Analog input voltage	V_{AN}	ANI6 to ANI11		−0.3 to AV_{REF} +0.3[Note] and −0.3 to V_{DD}/EV_{DD} +0.3[Note]	V
Output current, high	I_{OH1}	Per pin	P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P65, P67, P120, P142 to P144	−10	mA
		Total of all pins −80 mA	P40, P41, P120, P142 to P144	−25	mA
			P05, P11, P13, P14, P16, P17, P31, P50, P51, P65, P67	−55	mA
	I_{OH2}	Per pin	P26, P27, P150 to P153	−0.5	mA
		Total of all pins		−2	mA
Output current, low	I_{OL1}	Per pin	P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P60, P61, P65, P67, P120, P142 to P144	30	mA
		Total of all pins 200 mA	P40, P41, P120, P142 to P144	60	mA
			P05, P11, P13, P14, P16, P17, P31, P50, P51, P60, P61, P65, P67	140	mA
	I_{OL2}	Per pin	P26, P27, P150 to P153	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
Maximum permissible current	POW			748	mW

Note 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 port pins.

26.2 Recommended Operation Conditions of IO-Link Transceiver

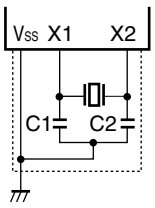
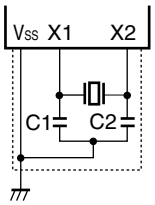
Recommended Operation Conditions of IO-Link Transceiver

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Supply voltage	V_{DDH}		8	24	36	V
		For IO-Link communication	18	24	30	V
	V_{DD_IO}	3.3 V interface supply	3	3.3	3.6	V
		5 V interface supply	4.5	5	5.5	V
Operating ambient temperature	T_A	In normal operation mode	−40 to +85			°C

26.3 Oscillator Characteristics

26.3.1 Main system clock oscillator characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (f_X)Note	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V	2.0		20.0	MHz
Crystal resonator		X1 clock oscillation frequency (f_X)Note	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V	2.0		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}/EV_{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 internal high-speed oscillation 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 oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be 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.

26.3.2 Internal oscillator characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2= 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
Internal high-speed oscillation clock frequency Note	f_{IH1M}			1.0		MHz
	f_{IH8M}			8.0		MHz
	f_{IH20M}			20		MHz
Internal low-speed oscillation clock frequency	f_{IL}	Normal current mode	27	30	33	kHz
		Low consumption current mode	25.5	30	34.5	kHz

Note This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Remark For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 20 REGULATOR**.

26.4 DC Characteristics

26.4.1 Pin Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P65, P67, P120, P142 to P144	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			–3.0	mA
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V			–1.0	mA
		Total of P40, P41, P120, P142 to P144 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			–20.0	mA
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V			–10.0	mA
		Total of P05, P11, P13, P14, P16, P17, P31, P50, P51, P65, P67 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			–30.0	mA
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V			–19.0	mA
		Total of all pins[Note 3] (When duty = 60%[Note 2])	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			–50.0	mA
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V			–29.0	mA
<R>		Total of all pins[Note 4] (When duty = 60%[Note 2])	3.0 V ≤ V_{DD}/EV_{DD} < 5.5 V			–16.0	mA
	I_{OH2}	Per pin for P26, P27, P150 to P153	AV_{REF} = V_{DD}/EV_{DD}			–0.1	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD}/EV_{DD} pin to an output pin.

2. Specification under conditions where the duty factor is 60%.
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%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 50% and I_{OH} = –20.0 mA
Total output current of pins = (–20.0 × 0.7)/(50 × 0.01) = –28.0 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.

<R> **3.** When the power is supplied from the external NPN transistor which is connecting the V_{REGO} pin to V_{DD}/EV_{DD}, V_{DD_IO}. Or when it supplies a power supply from the outside and does not use the built-in regulator of the IO-Link transceiver.

<R> **4.** When V_{REGO} pin is directly connected with IV_{DD}, and the power is supplied from the built-in regulator of the IO-Link transceiver to V_{DD}/EV_{DD}, V_{DD_IO}.

Caution P40 and P142 to P144 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P65, P67, P120, P142 to P144	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			8.5	mA
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V			1.0	mA
		Per pin for P60, P61	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			15.0	mA
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V			3.0	mA
		Total of P40, P41, P120 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			20.0	mA
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V			15.0	mA
		Total of P05, P11, P13, P14, P16, P17, P31, P50, P51, P60, P61, P65, P67 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			45.0	mA
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V			35.0	mA
		Total of all pins[Note 3] (When duty = 60%[Note 2])	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			65.0	mA
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V			50.0	mA
<R>		Total of all pins[Note 4] (When duty = 60%[Note 2])	3.0 V ≤ V_{DD}/EV_{DD} < 5.5 V			20.0	mA
	I_{OL2}	Per pin for P26, P27, P150 to P153	AV_{REF} = V_{DD}/EV_{DD}			0.4	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to V_{SS}/EV_{SS}, AV_{SS}, and GND1 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%).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 50% and I_{OL} = 20.0 mA

Total output current of pins = (20.0 × 0.7)/(50 × 0.01) = 28.0 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.

<R> **3.** When the power is supplied from the external NPN transistor which is connecting the V_{REGO} pin to V_{DD}/EV_{DD}, V_{DD_IO}. Or when it supplies a power supply from the outside and does not use the built-in regulator of the IO-Link transceiver.

<R> **4.** When V_{REGO} pin is directly connected with IV_{DD}, and the power is supplied from the built-in regulator of the IO-Link transceiver to V_{DD}/EV_{DD}, V_{DD_IO}.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P13, P41, P144		0.7 V_{DD}/EV_{DD}		V_{DD}/EV_{DD}	V
	V_{IH2}	P05, P11, P14, P16, P17, P31, P40, P50, P51, P65, P67, P120, P121, P142, P143, EXCLK, $\overline{\text{RESET}}$	Normal input buffer	0.8 V_{DD}/EV_{DD}		V_{DD}/EV_{DD}	V
	V_{IH3}	P11, P142, P143	TTL input buffer 4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V	2.2		V_{DD}/EV_{DD}	V
			TTL input buffer 3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V	2.0		V_{DD}/EV_{DD}	V
	V_{IH4}	P26, P27, P150 to P153	AV_{REF} = V_{DD}/EV_{DD}	0.7AV_{REF}		AV_{REF}	V
	V_{IH5}	P60, P61		0.7 V_{DD}/EV_{DD}		6.0	V
	V_{IH6}	FLMD0		0.9 V_{DD}/EV_{DD} Note		V_{DD}/EV_{DD}	V
<R>	V_{IH7}	SILM		0.7 V_{DD_IO}			V
	V_{THHs}	CQ	18 V < V_{DDH} < 30 V	10.5		13	V

Note Must be 0.9V_{DD}/EV_{DD} or higher when used in the flash memory programming mode.

Cautions 1. The maximum value of V_{IH} of pins P40 and P142 to P144 is V_{DD}/EV_{DD}, even in the N-ch open-drain mode.

2. For EXCLK, the value of V_{IH} and V_{IL} differs according to the input port mode or external clock mode. Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, low	V_{IL1}	P13, P41, P144		0		0.3 V_{DD}/EV_{DD}	V
	V_{IL2}	P05, P11, P14, P16, P17, P31, P40, P50, P51, P65, P67, P120, P121, P142, P143, EXCLK, $\overline{\text{RESET}}$	Normal input buffer	0		0.2 V_{DD}/EV_{DD}	V
	V_{IL3}	P11, P142, P143	TTL input buffer 4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V	0		0.5	V
	V_{IL4}	P26, 27, P150 to P153	AV_{REF} = V_{DD}/EV_{DD}	0		0.3AV_{REF}	V
	V_{IL5}	P60, P61		0		0.3 V_{DD}/EV_{DD}	V
	V_{IL6}	FLMD0 Note		0		0.1 V_{DD}/EV_{DD}	V
<R>	V_{IL7}	SILM				0.2 V_{DD_IO}	V
	V_{THLS}	CQ	18 V < V_{DDH} < 30 V	8		11.5	V

Note When disabling writing of the flash memory, connect the FLMD0 pin processing directly to V_{SS}/EV_{SS}, and maintain a voltage less than 0.1V_{DD}/EV_{DD}.

Caution For EXCLK, the value of V_{IH} and V_{IL} differs according to the input port mode or external clock mode. Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

	Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
	Output voltage, high	V_{OH1}	P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P65, P67, P120, P142 to P144	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, I_{OH1} = – 3.0 mA	V_{DD}/EV_{DD} – 0.7			V
				3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, I_{OH1} = –1.0 mA	V_{DD}/EV_{DD} – 0.5			V
		V_{OH2}	P26, P27, P150 to P153	AV_{REF} = V_{DD}/EV_{DD}, I_{OH2} = –0.1 mA	AV_{REF} – 0.5			V
<R>		V_{OH3}	RXD, WAKE, ILIM	I_{OH1} = –2.0 mA	0.8 V_{DD_IO}			V
		V_{RQHs}	CQ	I_{QHs} = –100 mA	V_{SUP} – 1.7			V
				I_{QHs} = –200 mA	V_{SUP} – 3.0			V
	Output voltage, low	V_{OL1}	P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P60, P61, P65, P67, P120, P142 to P144	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
				3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, I_{OL1} = 1.0 mA			0.5	V
		V_{OL2}	P26, P27, P150 to P153	AV_{REF} = V_{DD}/EV_{DD}, I_{OL2} = 0.4 mA			0.4	V
		V_{OL3}	P60, P61	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, I_{OL1} = 15.0 mA			2.0	V
				4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, I_{OL1} = 5.0 mA			0.4	V
				3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.4	V
<R>		V_{OL4}	RXD, WAKE, ILIM	I_{OL1} = 2.0 mA			0.2 V_{DD_IO}	V
		V_{RQLs}	CQ	I_{QHs} = 100 mA			1.5	V
				I_{QHs} = 200 mA			3.0	V

Remarks 1. V_{SUP}: Supply voltage

2. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

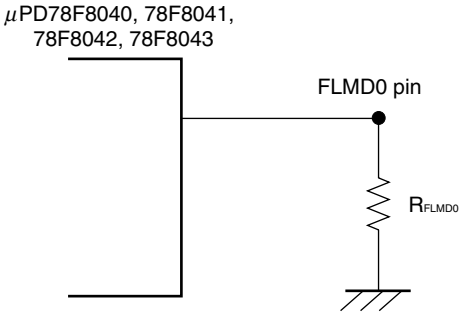
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P60, P61, P65, P67, P120, P142 to P144, FLMD0, $\overline{\text{RESET}}$	V_I = V_{DD}/EV_{DD}				1	μA
	I_{LIH2}	P26, P27, P150 to P153	V_I = AV_{REF}				1	μA
	I_{LIH3}	P121 (X1)	V_I = V_{DD}/EV_{DD}	In input port			1	μA
				In resonator connection			10	μA
		X2	V_I = V_{DD}/EV_{DD}				10	μA
Input leakage current, low	I_{LIL1}	P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P60, P61, P65, P67, P120, P142 to P144, FLMD0, $\overline{\text{RESET}}$	V_I = V_{SS}/EV_{SS}				–1	μA
	I_{LIL2}	P26, P27, P150 to P153	V_I = V_{SS}/EV_{SS} AV_{REF} = V_{DD}/EV_{DD}				–1	μA
	I_{LIL3}	P121 (X1)	V_I = V_{SS}/EV_{SS}	In input port			–1	μA
				In resonator connection			–10	μA
		X2	V_I = V_{SS}/EV_{SS}				–10	μA

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
On-chip pll-up resistance	R_U	P05, P11, P13, P14, P16, P17, P31, P40, P41, P50, P51, P65, P67, P120, P142 to P144	V_I = V_{SS}/EV_{SS}, In input port	10	20	100	kΩ
FLMD0 pin external pull-down resistance[Note]	R_{FLMD0}	When enabling the self-programming mode setting with software		100			kΩ

Note It is recommended to leave the FLMD0 pin open. If the pin is required to be pulled down externally, set R_{FLMD0} to 100 kΩ or more.



Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

26.4.2 Supply current characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1} Note 1	Operating mode	f_{MX} = 20 MHz Note 2, V_{DD}/EV_{DD} = 5.0 V	Square wave input		5.9	8.3	mA
				Resonator connection		6.2	8.6	mA
			f_{MX} = 20 MHz Note 2, V_{DD}/EV_{DD} = 3.0 V	Square wave input		5.9	8.3	mA
				Resonator connection		6.2	8.6	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD}/EV_{DD} = 5.0 V	Square wave input		3.3	4.8	mA
				Resonator connection		3.4	4.9	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD}/EV_{DD} = 3.0 V	Square wave input		3.3	4.8	mA
				Resonator connection		3.4	4.9	mA
			f_{MX} = 5 MHz Notes 2, 3, V_{DD}/EV_{DD} = 3.0 V	Square wave input		1.8	2.7	mA
				Resonator connection		1.9	2.8	mA
			f_{IH20} = 20 MHz Note 4	V_{DD}/EV_{DD} = 5.0 V		6.1	8.6	mA
				V_{DD}/EV_{DD} = 3.0 V		6.1	8.6	mA
			f_{IH} = 8 MHz Note 4	V_{DD}/EV_{DD} = 5.0 V		2.6	3.8	mA
				V_{DD}/EV_{DD} = 3.0 V		2.6	3.8	mA
			f_{IH} = 1 MHz Note 4	V_{DD}/EV_{DD} = 3.0 V		200	389	μA

Notes 1. Total current flowing into V_{DD}/EV_{DD}, and AV_{REF}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD}/EV_{DD} or V_{SS}/EV_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. When internal high-speed oscillator and 20 MHz internal high-speed oscillator are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0, FLPC and FSEL (bits 1 and 0 of operation speed mode control register (OSMC)) = 0 and 0.

4. When high-speed system clock and subsystem clock are stopped.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

f_{IH20}: 20 MHz internal high-speed oscillation clock frequency

f_{IH}: Internal high-speed oscillation clock frequency

2. Temperature condition of the TYP. value is T_A = 25°C

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD2} Note 1	HALT mode	f_{MX} = 20 MHz Note 2, V_{DD}/EV_{DD} = 5.0 V	Square wave input		1.2	3.6	mA
				Resonator connection		1.5	3.9	mA
			f_{MX} = 20 MHz Note 2, V_{DD}/EV_{DD} = 3.0 V	Square wave input		1.2	3.6	mA
				Resonator connection		1.5	3.9	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD}/EV_{DD} = 5.0 V	Square wave input		0.70	2.1	mA
				Resonator connection		0.80	2.2	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD}/EV_{DD} = 3.0 V	Square wave input		0.70	2.1	mA
				Resonator connection		0.80	2.2	mA
			f_{MX} = 5 MHz Notes 2, 3, V_{DD}/EV_{DD} = 3.0 V	Square wave input		0.41	1.8	mA
				Resonator connection		0.46	1.8	mA
			f_{IH20} = 20 MHz Note 4	V_{DD}/EV_{DD} = 5.0 V		1.4	3.9	mA
				V_{DD}/EV_{DD} = 3.0 V		1.4	3.9	mA
			f_{IH} = 8 MHz Note 4	V_{DD}/EV_{DD} = 5.0 V		0.48	1.8	mA
				V_{DD}/EV_{DD} = 3.0 V		0.48	1.8	mA
			f_{IH} = 1 MHz Note 4	V_{DD}/EV_{DD} = 3.0 V		55	168	μA
	I_{DD3} Note 5	STOP mode	T_A = –40 to +70 °C			0.37	5.2	μA
			T_A = –40 to +85 °C			0.37	7.9	μA

Notes 1. Total current flowing into V_{DD}/EV_{DD} and AV_{REF}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD}/EV_{DD} or V_{SS}/EV_{SS}. The maximum value include the peripheral operation current. However, not including the current flowing into the A/D converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. During HALT instruction execution by flash memory.

2. When internal high-speed oscillator and 20 MHz internal high-speed oscillator are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0, FLPC and FSEL (bits 1 and 0 of operation speed mode control register (OSMC)) = 0 and 0.

4. When high-speed system clock is stopped.

5. Total current flowing into V_{DD}/EV_{DD} and AV_{REF}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD}/EV_{DD} or V_{SS}/EV_{SS}. The maximum value includes the peripheral operation current and STOP leakage current. However, not including the current flowing into the A/D converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. When watchdog timer is stopped.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

f_{IH20}: 20 MHz internal high-speed oscillation clock frequency

f_{IH}: Internal high-speed oscillation clock frequency

2. Temperature condition of the TYP. value is T_A = 25°C

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Watchdog timer operating current	I_{WDT} Notes 1, 2	f_{IL} = 30 kHz				0.31	0.35	μA
A/D converter operating current	I_{ADC} Note 3	During conversion at maximum speed	High speed mode 1	AV_{REF} = V_{DD}/EV_{DD} = 5.0 V		1.72	3.2	mA
			High speed mode 2	AV_{REF} = V_{DD}/EV_{DD} = 3.0 V		0.72	1.6	mA
			Normal mode	AV_{REF} = V_{DD}/EV_{DD} = 5.0 V		0.86	1.9	mA
			Low voltage mode	AV_{REF} = V_{DD}/EV_{DD} = 3.0 V		0.37	0.8	mA
LVI operating current	I_{LVI} Note 4					9	18	μA
IO-Link transceiver operating current Note 5	IV_{DDH}	Supply current at V_{DDH} Note 6		No external load at IV_{DD}/V_{REGO} Note 6, TXEN = low		1.5	4.0	mA
	IIV_{DD}	Supply current at IV_{DD} Note 7		IV_{DD} = 5 V		0.6	2	mA
	IV_{DD_IO}	Supply current at V_{DD_IO}		Static condition		20.0	50.0	μA

Notes 1. When internal high-speed oscillator and high-speed system clock are stopped.

2. Current flowing only to the watchdog timer (including the operating current of the 30 kHz internal oscillator). The current value of the μPD78F8040, 78F8041, 78F8042, 78F8043 are the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates during STOP mode.

3. Current flowing only to the A/D converter (AV_{REF} pin). The current value of the μPD78F8040, 78F8041, 78F8042, 78F8043 are 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.

4. Current flowing only to the LVI circuit. The current value of the μPD78F8040, 78F8041, 78F8042, 78F8043 are the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates in the Operating, HALT or STOP mode.

5. The current path varies depending on the power supply configuration. (For details, see **Figure of the current paths for the different power supply configurations**.)

6. This is the current when V_{REGO} and IV_{DD} are connected. A current from V_{DDH} that includes the current flowing to IV_{DD} is supplied to the internal regulator of the IO-Link transceiver.

7. This is the current when supplying 5 V from an external power supply to IV_{DD} or when using an external NPN transistor. It does not include the current flowing to V_{DDH}.

Remarks 1. f_{IL}: Internal low-speed oscillation clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

2. Temperature condition of the TYP. value is T_A = 25°C

Figure of the current paths for the different power supply configurations

Figure 26-1 Current Paths for Different Power Supply Configurations (when V_{REGO} and IV_{DD} are connected)

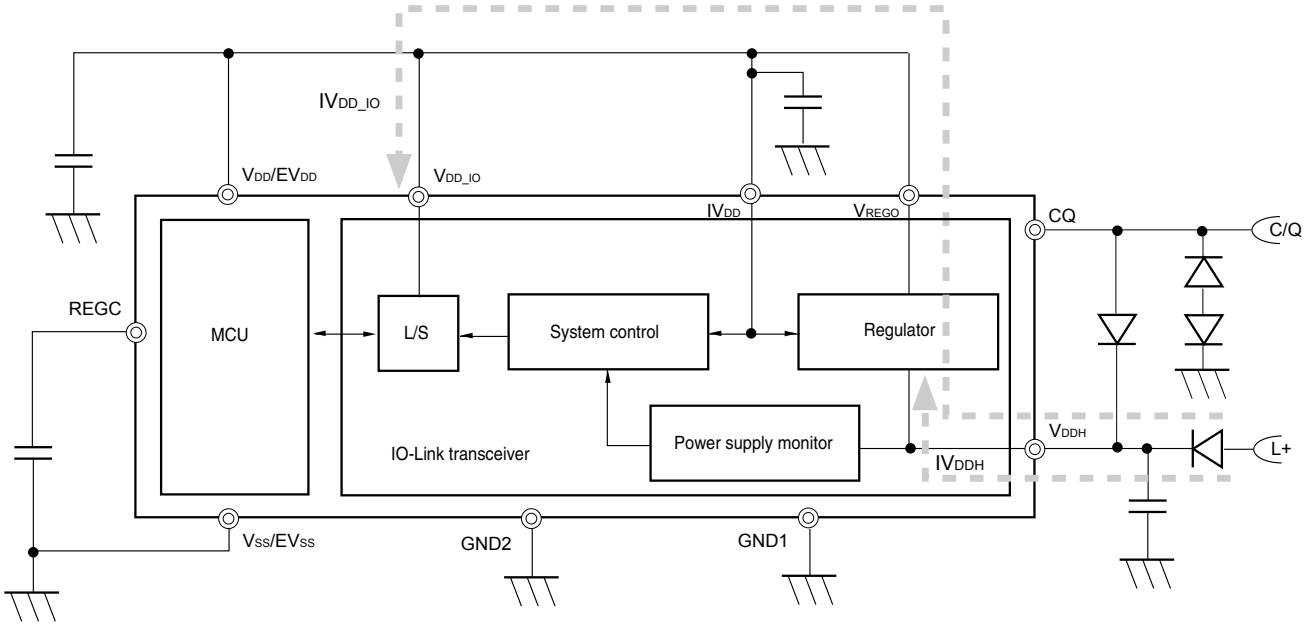
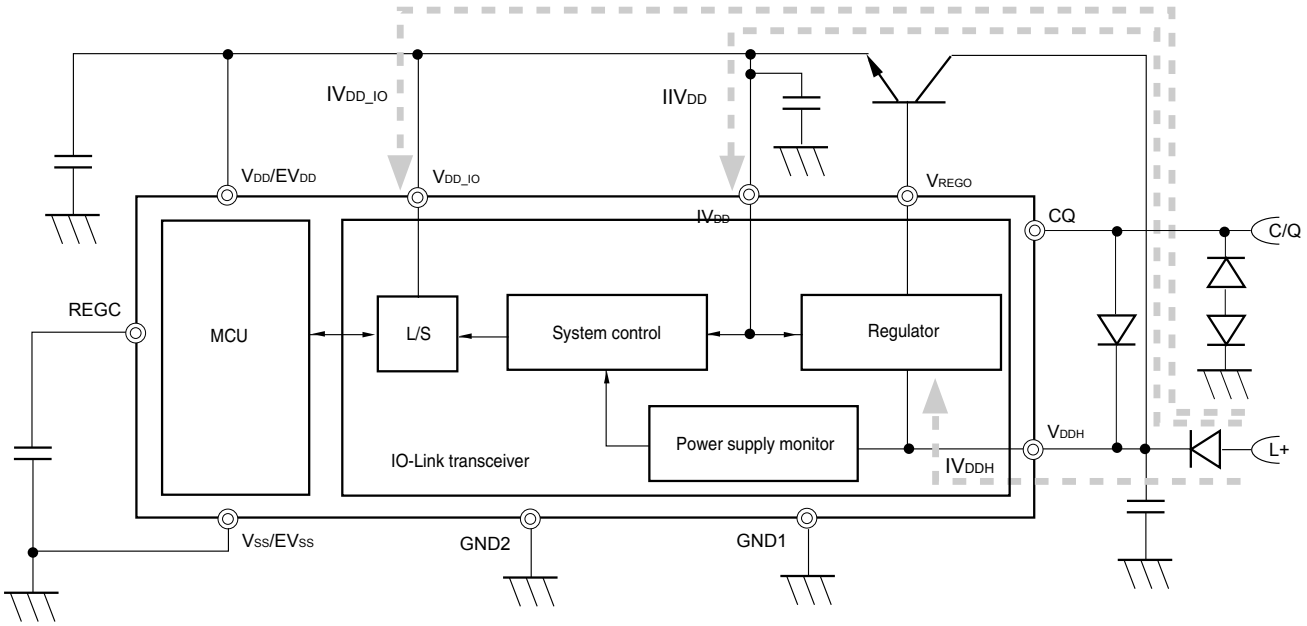


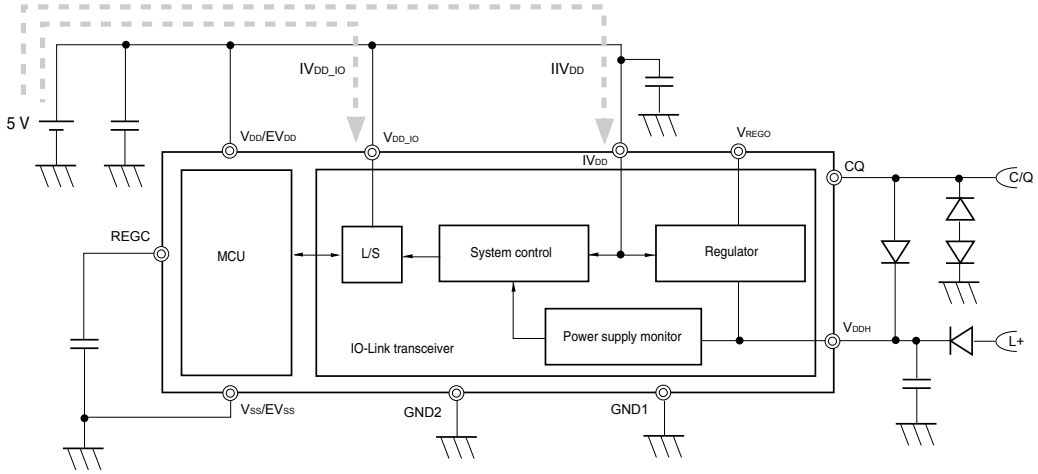
Figure 26-2 Current Paths for Different Power Supply Configurations (when using an external NPN transistor)



Cautions 1. Make QFN package exposed die pad (GND3) the same potential as V_{SS}/EV_{SS}.

<R> **2. Be sure to connect an NPN transistor externally when the internal regulator is used in the FBGA package poducts (refer to figure 26-2). The power supply configuration of figure 26-1 is prohibited in the FBGA package poducts.**

Figure 26-3 Current Paths for Different Power Supply Configurations (when supplying 5 V from an external power supply to IV_{DD})



Caution Make QFN package exposed die pad (GND3) the same potential as V_{SS}/EV_{SS}.

26.5 AC Characteristics

26.5.1 Basic operation

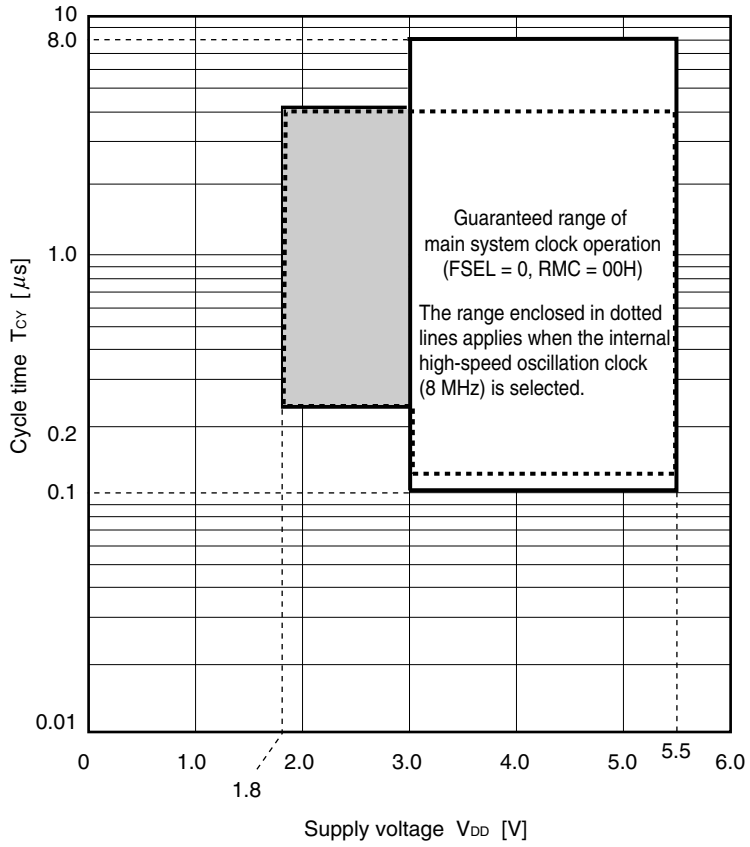
(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	Normal current mode	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V	0.05		8	μs
			Low consumption current mode		1		8	μs
		In the self programming mode	Normal current mode	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V	0.05		1	μs
External main system clock frequency	f_{EX}	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			2.0		20.0	MHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			24			ns
TI01 to TI03, TI05, TI11, TI13 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK}+10			ns
TO01 to TO03, TO05, TO11, TO13 output frequency	f_{TO}	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V					10	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}				1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Remarks 1. f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the TMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7))

2. For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 20 REGULATOR**.

Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 00H)

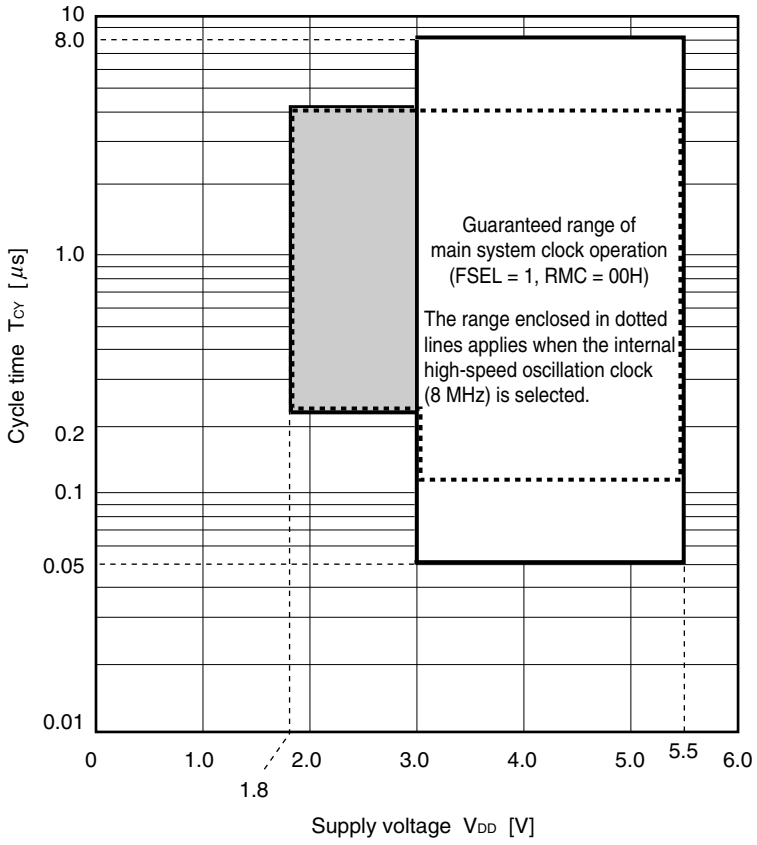


Caution Only the following operations are possible in the shaded section above:

- **Using the CPU (executing instructions)**
  The CPU clock is a fraction of the 8 MHz internal high-speed oscillation clock ($f_{IH}/2^5$ to $f_{IH}/2$ (250 kHz to 4 MHz)).
- **Reading or writing the internal RAM**
- **Using the low-voltage detector (LVI)**
- **Using the interval timer function of the timer array unit (TAU)**
- **Specifying the mode of the standby function (STOP or HALT mode)**
- **Setting up the control registers of the clock generator**
  However, the clock can be switched only if the clock cycle time after the clock has been switched will be within the guaranteed operating range.
- **Using the watchdog timer (WDT) (including the internal low-speed oscillator)**

Remark FSEL: Bit 0 of the operation speed mode control register (OSMC)
RMC: Regulator mode control register

Minimum instruction execution time during main system clock operation (FSEL = 1, RMC = 00H)



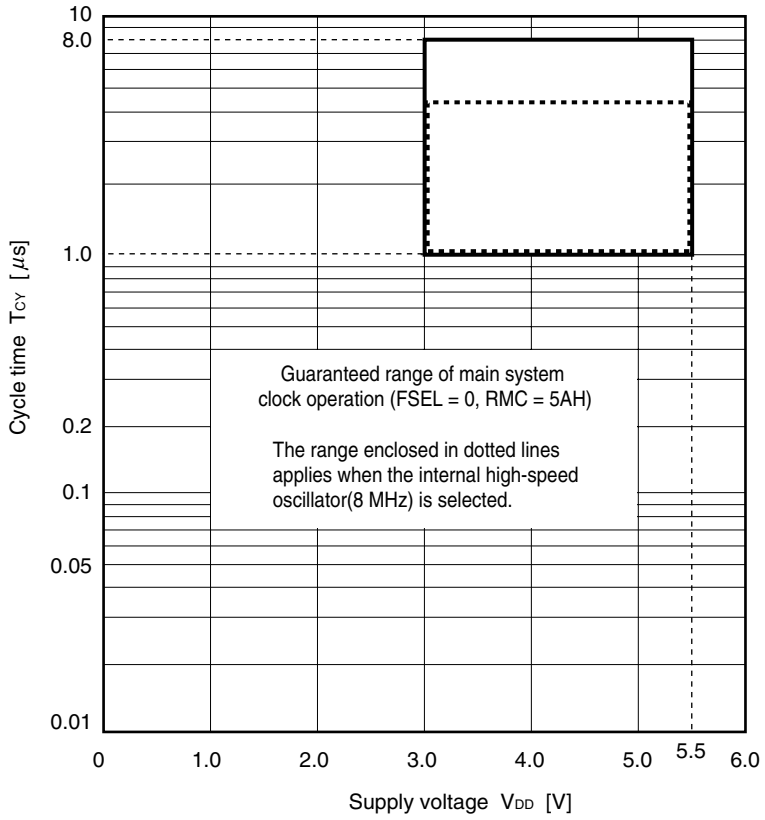
Cautions 1. To set FSEL = 0 when the clock is operating at 10 MHz or less.

2. Only the following operations are possible in the shaded section above:

- **Using the CPU (executing instructions)**
  The CPU clock is a fraction of the 8 MHz internal high-speed oscillation clock ($f_{IH}/2^5$ to $f_{IH}/2$ (250 kHz to 4 MHz)).
- **Reading or writing the internal RAM**
- **Using the low-voltage detector (LVI)**
- **Using the interval timer function of the timer array unit (TAU)**
- **Specifying the mode of the standby function (STOP or HALT mode)**
- **Setting up the control registers of the clock generator**
  However, the clock can be switched only if the clock cycle time after the clock has been switched will be within the guaranteed operating range.
- **Using the watchdog timer (WDT) (including the internal low-speed oscillator)**

Remark FSEL: Bit 0 of the operation speed mode control register (OSMC)
RMC: Regulator mode control register

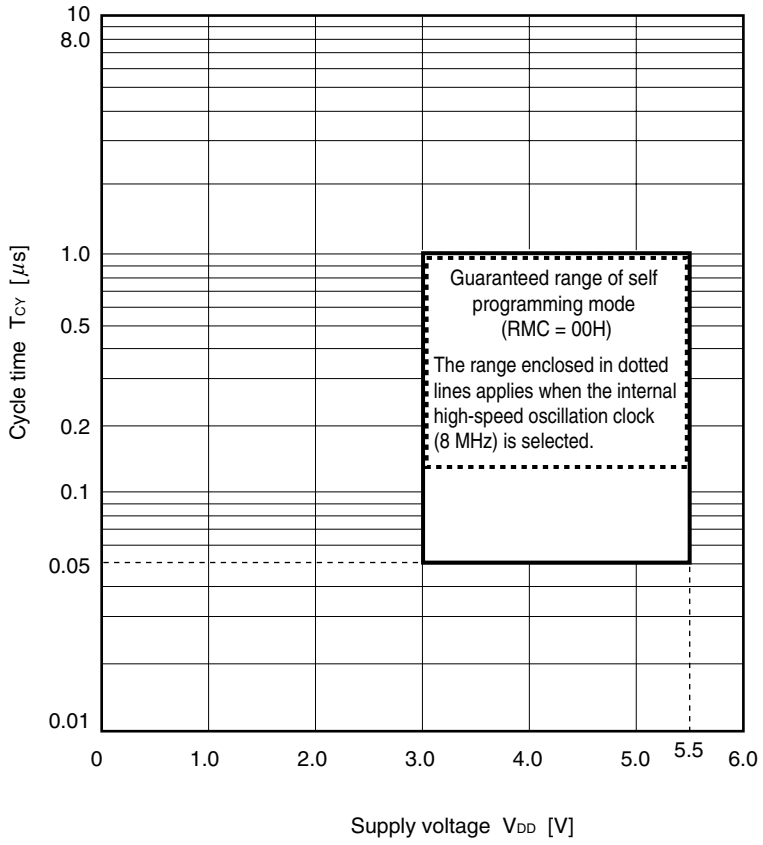
Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 5AH)



Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)
RMC: Regulator mode control register

2. The entire voltage range is 1 MHz (MAX.) when RMC is set to 5AH.

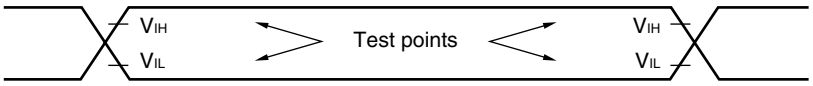
Minimum instruction execution time during self programming mode (RMC = 00H)



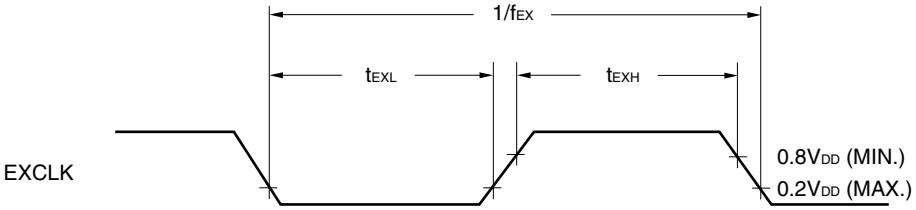
Remark RMC: Regulator mode control register

26.5.2 Measurement conditions

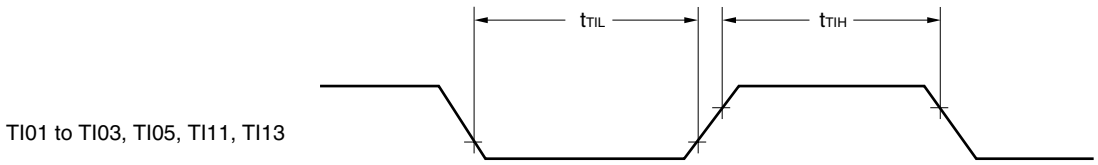
AC Timing Test Points



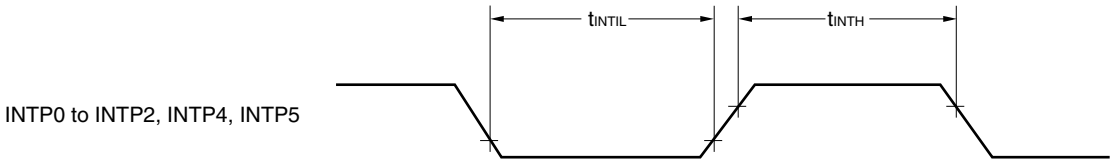
External Main System Clock Timing



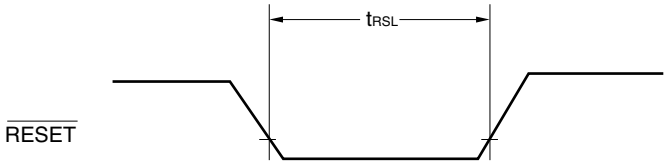
TI Timing



Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing



26.6 Peripheral Functions Characteristics

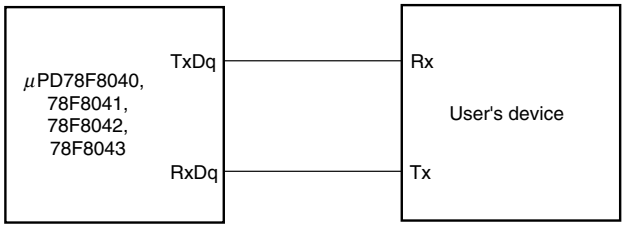
26.6.1 Serial array unit

(1) During communication at same potential (UART mode) (dedicated baud rate generator output)

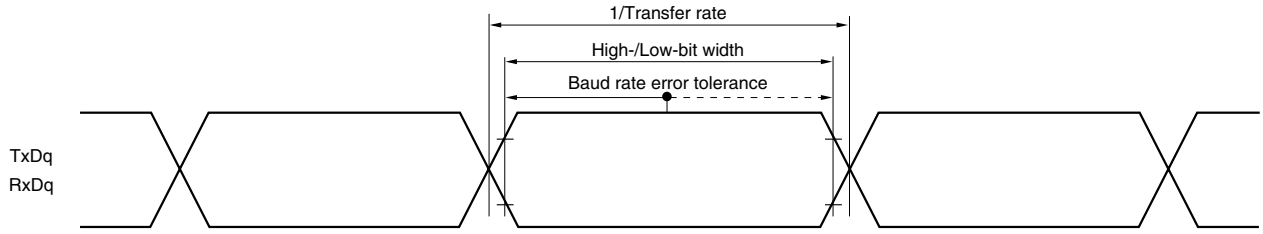
(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					f_{MCK}/6	bps
		f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}			3.3	Mbps

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Cautions 1. Select the normal input buffer for RxDq and the normal output mode for TxDq by using the PIM14 and POM14 registers.

2. UART0 (0 and 1 channels of unit 0) is dedicated to IO-Link communication.

Remarks 1. q: UART number (q = 0, 2, 3)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3))

(2) During communication at same potential (CSI mode) (master mode, $\overline{SCK20}$... internal clock output)

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK20}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V	200 Note 1			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V	300 Note 1			ns
$\overline{SCK20}$ high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V	t_{KCY1}/2 – 20			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V	t_{KCY1}/2 – 35			ns
SI20 setup time (to $\overline{SCK20}$↑) Note 2	t_{SIK1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V	70			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V	100			ns
SI20 hold time (from $\overline{SCK20}$↑) Note 2	t_{KSI1}		30			ns
Delay time from $\overline{SCK20}$↓ to SO20 output Note 3	t_{KSO1}	C = 30 pF Note 4			40	ns

Notes 1. The value must also be 4/f_{CLK} or more.

2. When DAP10 = 0 and CKP10 = 0, or DAP10 = 1 and CKP10 = 1. The SI20 setup time becomes "to $\overline{SCK20}$↓" when DAP10 = 0 and CKP10 = 1, or DAP10 = 1 and CKP10 = 0.

3. When DAP10 = 0 and CKP10 = 0, or DAP10 = 1 and CKP10 = 1. The SI20 hold time becomes "from $\overline{SCK20}$↓" when DAP10 = 0 and CKP10 = 1, or DAP10 = 1 and CKP10 = 0.

4. C is the load capacitance of the $\overline{SCK20}$ and SO20 output lines.

Caution Select the normal input buffer for SI20 and the normal output mode for SO20 and $\overline{SCK20}$ by using the PIM14 and POM14 registers.

(3) During communication at same potential (CSI mode) (slave mode, $\overline{SCK20}$... external clock input)

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK20}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V		6/f_{MCK}			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V	16 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
$\overline{SCK20}$ high-/low-level width	t_{KH2}, t_{KL2}			f_{KCY2}/2			ns
SI20 setup time (to $\overline{SCK20}$↑) Note 1	t_{SIK2}			80			ns
SI20 hold time (from $\overline{SCK20}$↑) Note 1	t_{KSI2}			1/f_{MCK}+50			ns
Delay time from $\overline{SCK20}$↓ to SO20 output Note 2	t_{KSO2}	C = 30 pF Note 3	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			2/f_{MCK}+45	ns
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V			2/f_{MCK}+57	ns

Notes 1. When DAP10 = 0 and CKP10 = 0, or DAP10 = 1 and CKP10 = 1. The SI20 setup time becomes "to $\overline{SCK20}$↓" when DAP10 = 0 and CKP10 = 1, or DAP10 = 1 and CKP10 = 0.

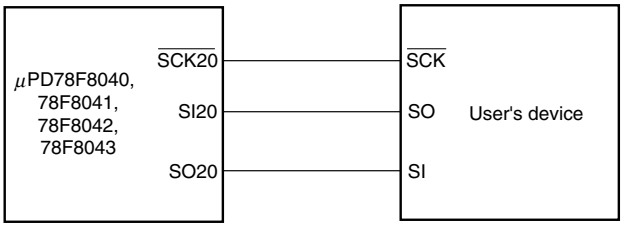
2. When DAP10 = 0 and CKP10 = 0, or DAP10 = 1 and CKP10 = 1. The delay time to SO20 output becomes "from $\overline{SCK20}$↑" when DAP10 = 0 and CKP10 = 1, or DAP10 = 1 and CKP10 = 0.

3. C is the load capacitance of the SO20 output lines.

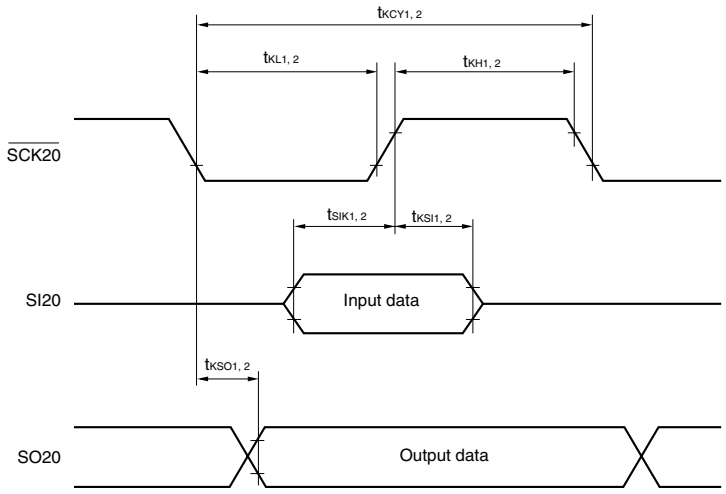
Caution Select the normal input buffer for SI20 and $\overline{SCK20}$ and the normal output mode for SO20 by using the PIM14 and POM14 registers.

Remark f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS10 bit of the SMR10 register.)

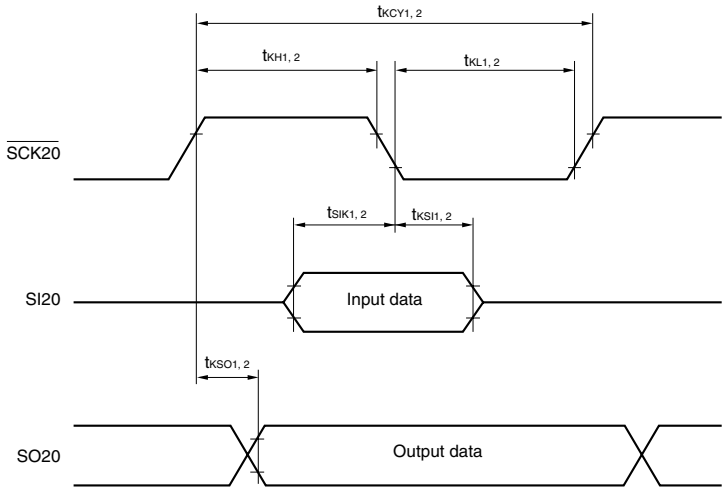
CSI mode connection diagram (during communication at same potential)



CSI mode serial transfer timing (during communication at same potential)
(When DAP10 = 0 and CKP10 = 0, or DAP10 = 1 and CKP10 = 1.)



CSI mode serial transfer timing (during communication at same potential)
(When DAP10 = 0 and CKP10 = 1, or DAP10 = 1 and CKP10 = 0.)



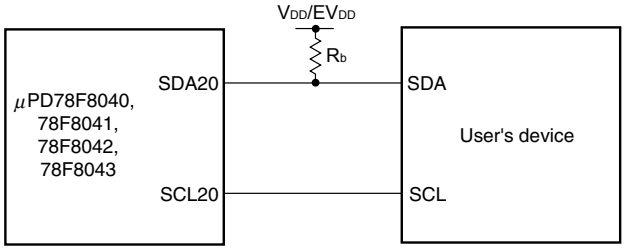
(4) During communication at same potential (simplified I²C mode)

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

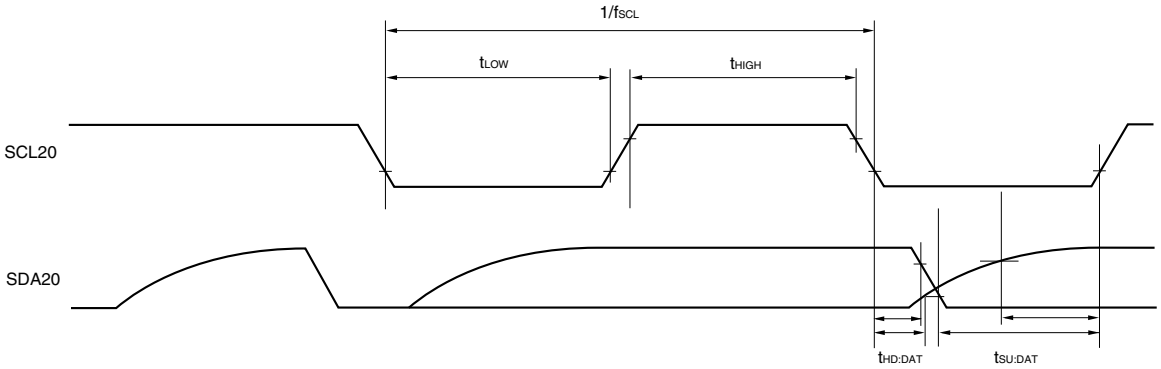
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCL20 clock frequency	f_{SCL}	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400[Note]	kHz
Hold time when SCL20 = “L”	t_{LOW}	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1200		ns
Hold time when SCL20 = “H”	t_{HIGH}	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1200		ns
Data setup time (reception)	$t_{SU:DAT}$	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK}+120		ns
Data hold time (transmission)	$t_{HD:DAT}$	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	660	ns

Note The value must also be f_{MCK}/4 or less.

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 normal input buffer and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for SDA20 and the normal output mode for SCL20 by using the PIM14 and POM14 registers.

Remarks 1. R_b[Ω]:Communication line (SDA20) pull-up resistance,
C_b[F]: Communication line (SCL20, SDA20) load capacitance

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS10 bit of the SMR10 register.)

(5) During Communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output) (1/2)

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		reception	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V				f_{MCK}/6	bps
				f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}			3.3	Mbps
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V				f_{MCK}/6	bps
				f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}			3.3	Mbps

Caution Select the TTL input buffer for RxD2 and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for TxD2 by using the PIM14 and POM14 registers.

Remarks 1. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS1n bit of the SMR1n register. n: Channel number (n = 0, 1))

2. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.

4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V

3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

3. UART0 and UART3 cannot communicate at different potential. Use UART2 for communication at different potential.

(5) During Communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output) (2/2)
(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		transmission	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V				**Note 1**	
				f_{CLK} = 16.8 MHz, f_{MCK} = f_{CLK}, C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V			2.8 Note 2	Mbps
			3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V				**Note 3**	
				f_{CLK} = 19.2 MHz, f_{MCK} = f_{CLK}, C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V			1.2 Note 4	Mbps

Notes 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

2. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to Note 1 above to calculate the maximum transfer rate under conditions of the customer.

3. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

4. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to Note 3 above to calculate the maximum transfer rate under conditions of the customer.

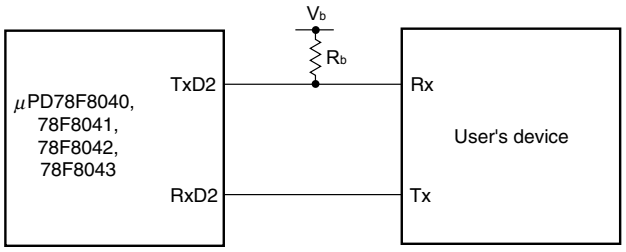
Caution Select the TTL input buffer for RxD2 and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for TxD2 by using the PIM14 and POM14 registers.

Remarks 1. R_b[Ω]:Communication line (TxD2) pull-up resistance,
C_b[F]: Communication line (TxD2) load capacitance, V_b[V]: Communication line voltage

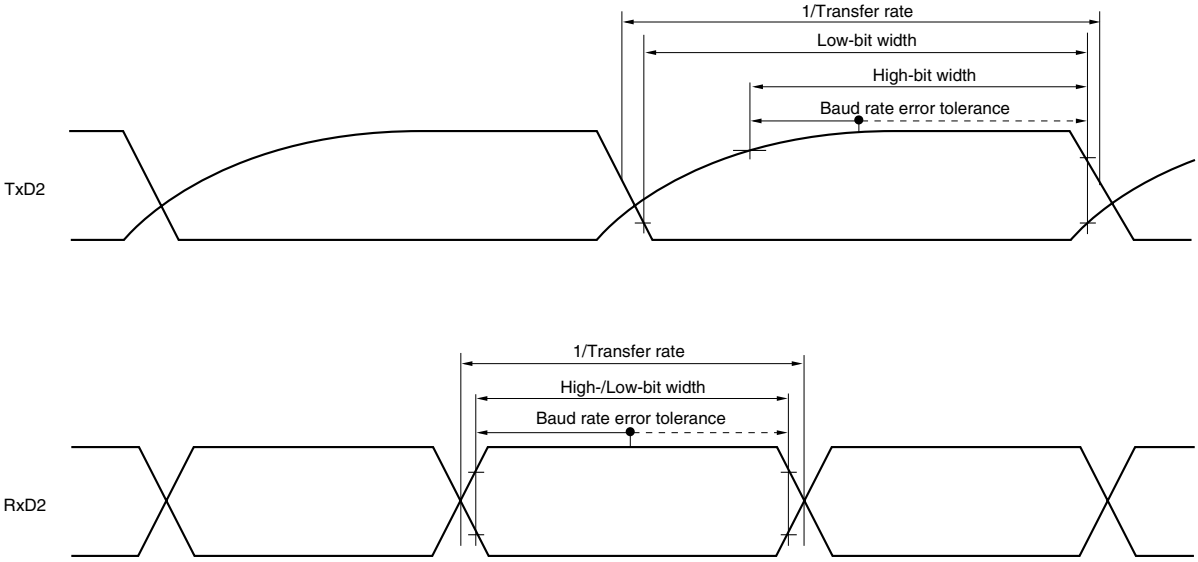
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of the SMR0n register. n: Channel number (n = 0, 1))

3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.
4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Caution Select the TTL input buffer for RxD2 and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for TxD2 by using the PIM14 and POM14 registers.

Remark R_b[Ω]:Communication line (TxD2) pull-up resistance, V_b[V]: Communication line voltage

(6) During Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{SCK20}$... internal clock output) (1/2)

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK20}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	400 Note 1			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	800 Note 1			ns
$\overline{SCK20}$ high-level width	t_{KH1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 75			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 170			ns
$\overline{SCK20}$ low-level width	t_{KL1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 20			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 35			ns
SI20 setup time (to $\overline{SCK20}$↑) Note	t_{SIK1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	150			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	275			ns
SI20 hold time (from $\overline{SCK20}$↑) Note	t_{KSI1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	30			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCK20}$↓ to SO20 output Note	t_{KSO1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			120	ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			215	ns

Notes 1. The value must also be 4/f_{CLK} or more.

2. When DAP10 = 0 and CKP10 = 0, or DAP10 = 1 and CKP10 = 1.

Caution Select the TTL input buffer for SI20 and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for SO20 and $\overline{SCK20}$ by using the PIM14 and POM14 registers.

Remarks 1. R_b[Ω]:Communication line ($\overline{SCK20}$, SO20) pull-up resistance,
C_b[F]: Communication line (SO20, $\overline{SCK20}$) load capacitance, V_b[V]: Communication line voltage

2. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.

4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V

3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

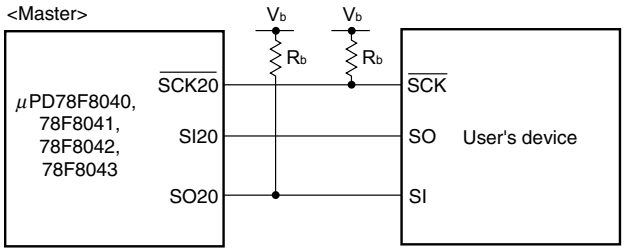
(6) During Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{SCK20}$... internal clock output) (2/2)

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SI20 setup time (to $\overline{SCK20}$↓) Note	t_{SIK1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	70			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	100			ns
SI20 hold time (from $\overline{SCK20}$↓) Note	t_{KSI1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	30			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCK20}$↑ to SO20 output Note	t_{KSO1}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			40	ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			40	ns

Note When DAP10 = 0 and CKP10 = 1, or DAP10 = 1 and CKP10 = 0.

CSI mode connection diagram (during communication at different potential)



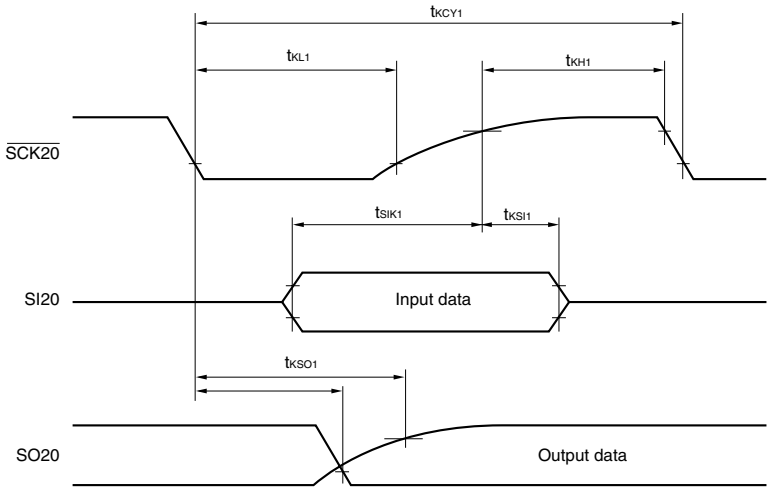
Caution Select the TTL input buffer for SI20 and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for SO20 and $\overline{SCK20}$ by using the PIM14 and POM14 registers.

Remarks 1. R_b[Ω]:Communication line ($\overline{SCK20}$, SO20) pull-up resistance,
C_b[F]: Communication line (SO20, $\overline{SCK20}$) load capacitance, V_b[V]: Communication line voltage

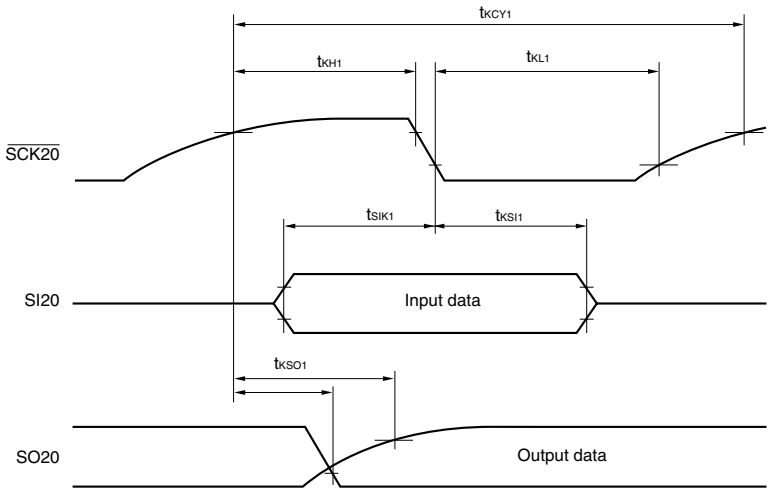
2. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.

4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAP10 = 0 and CKP10 = 0, or DAP10 = 1 and CKP10 = 1.)



CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAP10 = 0 and CKP10 = 1, or DAP10 = 1 and CKP10 = 0.)



Caution Select the TTL input buffer for SI20 and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for SO20 and $\overline{SCK20}$ by using the PIM14 and POM14 registers.

(7) During Communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{SCK20}$... external clock input)

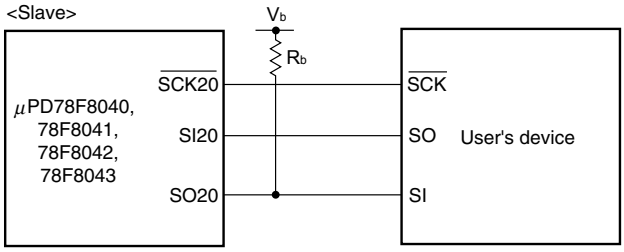
(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK20}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	13.6 MHz < f_{MCK}	10/f_{MCK}			ns
			6.8 MHz < f_{MCK} ≤ 13.6 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 6.8 MHz	6/f_{MCK}			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	18.5 MHz < f_{MCK}	16/f_{MCK}			ns
			14.8 MHz < f_{MCK} ≤ 18.5 MHz	14/f_{MCK}			ns
			11.1 MHz < f_{MCK} ≤ 14.8 MHz	12/f_{MCK}			ns
			7.4 MHz < f_{MCK} ≤ 11.1 MHz	10/f_{MCK}			ns
			3.7 MHz < f_{MCK} ≤ 7.4 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 3.7 MHz	6/f_{MCK}			ns
$\overline{SCK20}$ high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 – 20			ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 – 35			ns
SI20 setup time (to $\overline{SCK20}$↑) Note 1	t_{SIK2}			90			ns
SI20 hold time (from $\overline{SCK20}$↑) Note 1	t_{KSI2}			1/f_{MCK} + 50			ns
Delay time from $\overline{SCK20}$↓ to SO20 output Note 2	t_{KSO2}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ				2/f_{MCK} + 120	ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ				2/f_{MCK} + 230	ns

Notes 1. When DAP10 = 0 and CKP10 = 0, or DAP10 = 1 and CKP10 = 1. The SI20 setup time becomes "to $\overline{SCK20}$↓" when DAP10 = 0 and CKP10 = 1, or DAP10 = 1 and CKP10 = 0.

2. When DAP10 = 0 and CKP10 = 0, or DAP10 = 1 and CKP10 = 1. The delay time to SO20 output becomes "from $\overline{SCK20}$↑" when DAP10 = 0 and CKP10 = 1, or DAP10 = 1 and CKP10 = 0.

CSI mode connection diagram (during communication at different potential)



(**Caution** and **Remark** are given on the next page.)

Caution Select the TTL input buffer for SI20 and $\overline{\text{SCK20}}$ and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for SO20 by using the PIM14 and POM14 registers.

Remarks 1. R_b[Ω]:Communication line (SO20) pull-up resistance,
C_b[F]: Communication line (SO20) load capacitance, V_b[V]: Communication line voltage

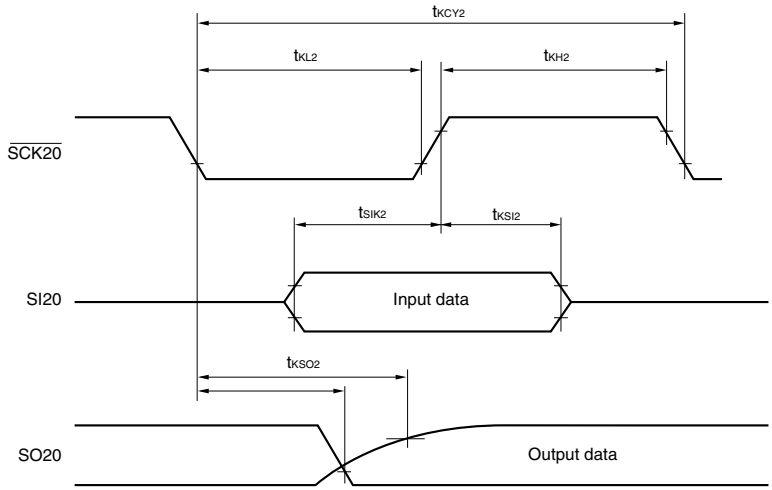
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS10 bit of the SMR10 register.)

3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.

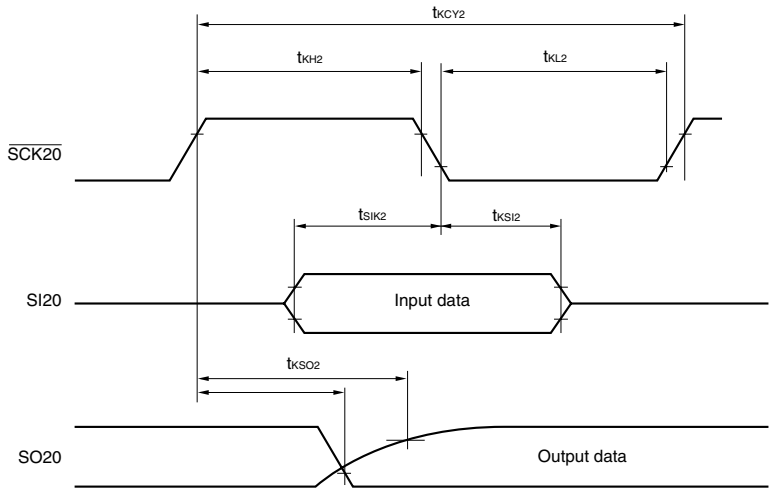
$4.0\ V \le V_{DD}/EV_{DD} \le 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$: $V_{IH} = 2.2\ V$, $V_{IL} = 0.8\ V$

$3.0\ V \le V_{DD}/EV_{DD} < 4.0\ V$, $2.3\ V \le V_b \le 2.7\ V$: $V_{IH} = 2.0\ V$, $V_{IL} = 0.5\ V$

CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAP10 = 0 and CKP10 = 0, or DAP10 = 1 and CKP10 = 1.)



CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAP10 = 0 and CKP10 = 1, or DAP10 = 1 and CKP10 = 0.)



Caution Select the TTL input buffer for SI20 and $\overline{\text{SCK20}}$ and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for SO20 by using the PIM14 and POM14 registers.

(8) During Communication at different potential (2.5 V, 3 V) (simplified I^2C mode)

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCL20 clock frequency	f_{SCL}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ		400[Note]	kHz
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400[Note]	kHz
Hold time when SCL20 = “L”	t_{LOW}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	1275		ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1275		ns
Hold time when SCL20 = “H”	t_{HIGH}	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	655		ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	655		ns
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	1/f_{MCK} + 190		ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 190		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	0	640	ns
		3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	660	ns

Note The value must also be f_{MCK}/4 or less.

Caution Select the TTL input buffer and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for SDA20 and the N-ch open drain output (V_{DD} tolerance) mode for SCL20 by using the PIM14 and POM14 registers.

Remarks 1. R_b[Ω]:Communication line (SDA20, SCL20) pull-up resistance,
C_b[F]: Communication line (SDA20, SCL20) load capacitance, V_b[V]: Communication line voltage

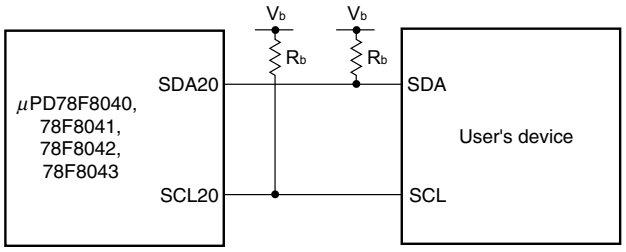
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS10 bit of the SMR10 register.)

3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified I^2C mode mode.

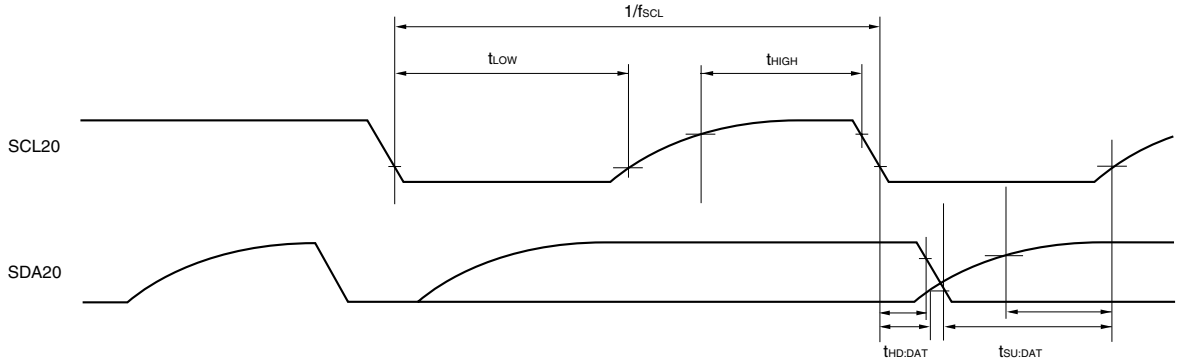
4.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V

3.0 V ≤ V_{DD}/EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

Simplified I²C mode connection diagram (during communication at different potential)



Simplified I²C mode serial transfer timing (during communication at different potential)



Caution Select the TTL input buffer and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for SDA20 and the N-ch open drain output (V_{DD}/EV_{DD} tolerance) mode for SCL20 by using the PIM14 and POM14 registers.

Remark R_b[Ω]:Communication line (SDA20, SCL20) pull-up resistance, V_b[V]: Communication line voltage

26.6.2 Serial interface IICA

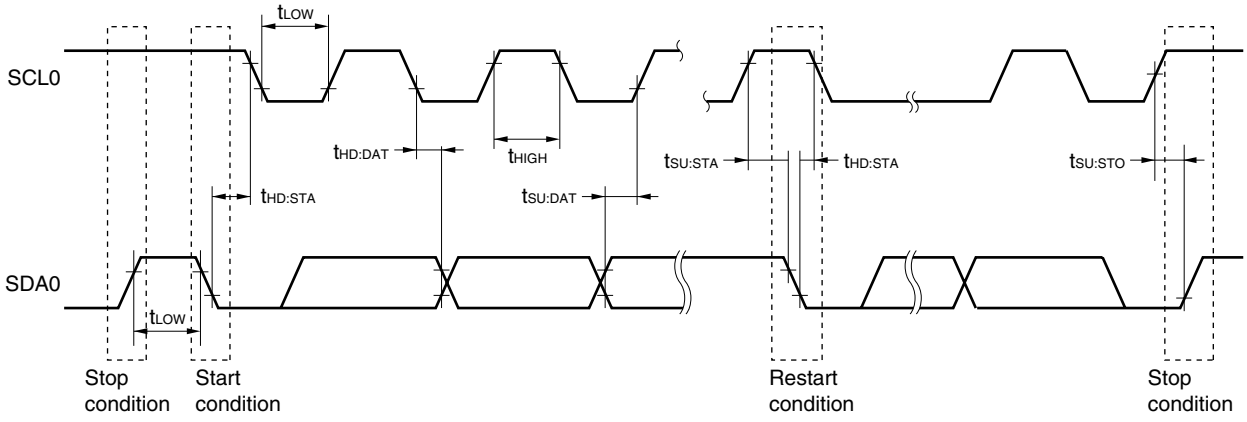
(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	Standard Mode		Fast Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz Standard mode: f_{CLK} ≥ 1 MHz	0	100	0	400	kHz
Setup time of restart condition Note 1	$t_{SU:STA}$		4.7		0.6		μs
Hold time	$t_{HD:STA}$		4.0		0.6		μs
Hold time when SCL0 = "L"	t_{LOW}		4.7		1.3		μs
Hold time when SCL0 = "H"	t_{HIGH}		4.0		0.6		μs
Data setup time (reception)	$t_{SU:DAT}$		250		100		ns
Data hold time (transmission) Note 2	$t_{HD:DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$		4.0		0.6		μs
Bus-free time	t_{BUF}		4.7		1.3		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.

IICA serial transfer timing



26.6.3 On-chip debug (UART)

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, V_{SS}/EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate			$f_{CLK}/2^{12}$		$f_{CLK}/6$	bps
		Flash memory programming mode			3.33	Mbps
TOOL1 output frequency	f_{TOOL1}	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V			10	MHz

26.6.4 A/D converter characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					10	bit
Overall error[Notes 1, 2]	AINL					±0.35	%FSR
Conversion time	tCONV	4.0 V ≤ AV_{REF} ≤ 5.5 V	High speed mode 1	2.5		66.6	μs
			Normal mode	5.2		66.6	μs
		2.7 V ≤ AV_{REF} < 5.5 V	High speed mode 2	3.5		66.6	μs
			Normal mode	8.6		66.6	μs
		1.8 V ≤ AV_{REF} < 4.0 V	Low voltage mode	24.1		66.6	μs
Zero-scale error[Notes 1, 2]	EZS					±0.25	%FSR
Full-scale error[Notes 1, 2]	EFS					±0.25	%FSR
Integral non-linearity error[Note 1]	ILE					±2.5	LSB
Differential non-linearity error[Note 1]	DLE					±1.5	LSB
Analog input voltage	VAIN	1.8 V ≤ AV_{REF} ≤ 5.5 V		AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

26.6.5 IO-Link transceiver characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output voltage of internal voltage regulator	IV_{DD}	8 V ≤ V_{DDH} ≤ 36 V, IV_{DD} = V_{REGO}	4.75	5	5.25	V
	V_{REGO}	When using an external NPN transistor		IV_{DD}+0.7		V
Output current of internal voltage regulator	IV_{REGO}	IV_{DD} = V_{REGO}			20	mA
IV_{DD} undervoltage threshold	IV_{DD_UV}		3.5		4.5	V
V_{DD_IO} undervoltage threshold	$V_{DD\_IO\_UV}$		1.5		3	V

<R> **(1) Transmitter**

(a) DC Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

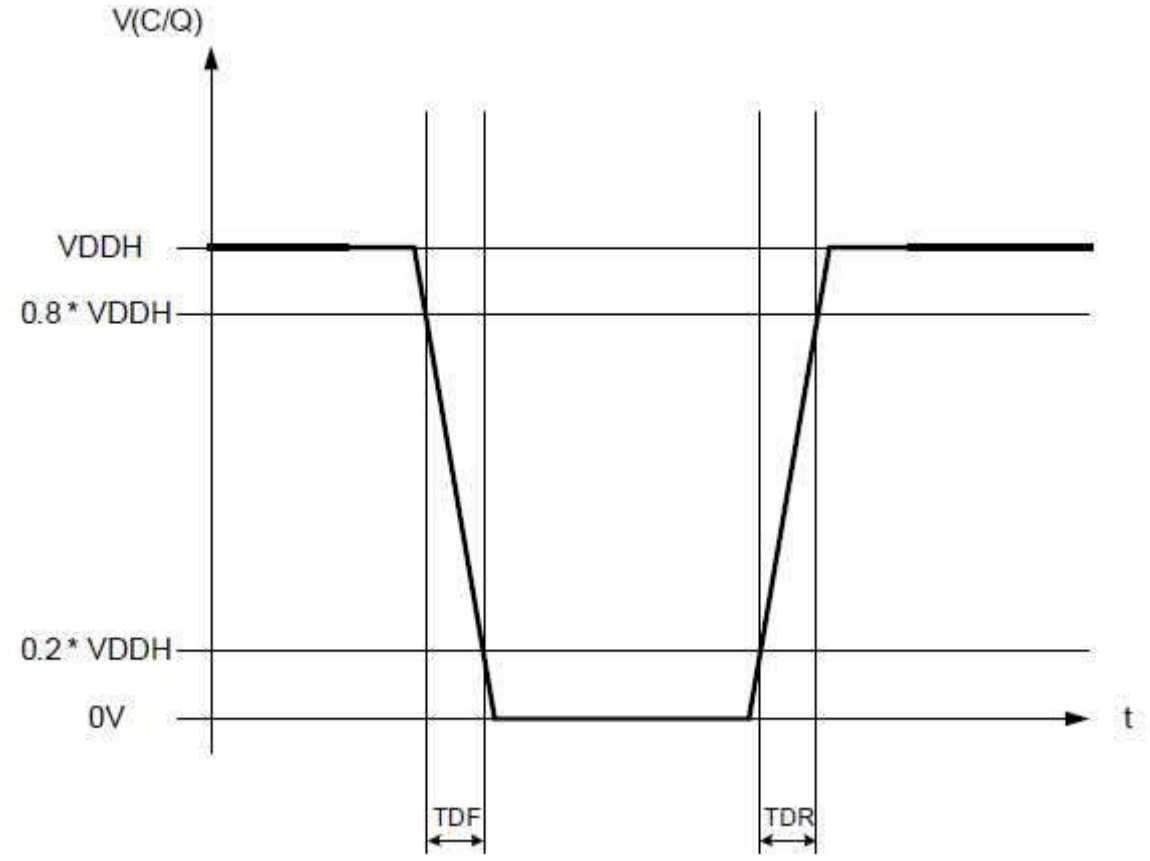
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
DC driver residual voltage low	V_{RQLs}	TXD = high	I_{QLs} = 100 mA			1.5	V
			I_{QLs} = 200 mA 8 V ≤ V_{DDH} < 15 V			2.5	V
			I_{QLs} = 200 mA 15 V ≤ V_{DDH} ≤ 36 V			2.0	V
DC driver residual voltage high	V_{RQHs}	TXD = low	I_{QHs} = –100 mA	V_{SUP} –1.7			V
		TXD = high	I_{QHs} = –200 mA 8 V ≤ V_{DDH} < 15 V	V_{SUP} –2.5			V
			I_{QHs} = –200 mA 15 V ≤ V_{DDH} ≤ 36 V	V_{SUP}–2.0			V
Overcurrent shutoff threshold low	I_{THL_OFF}	Driver current low, TXD = high, SILIM = low		220	350	480	mA
Overcurrent shutoff threshold high	I_{THH_OFF}	Driver current high, TXD = low, SILIM = low		–480	–350	–220	mA

Remark V_{SUP}: Supply voltage

(b) AC Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage rise time 230.4 kBaud	T_{DR}	C_{load} = 5 nF, R_{load} = 2 kΩ, TXD high to low transition	SPEED = high			896	ns
Output voltage rise time 38.4 kBaud			SPEED = low			5.2	μs
Output voltage fall time 230.4 kBaud	T_{DF}	C_{load} = 5 nF, R_{load} = 2 kΩ, TXD low to high transition	SPEED = high			896	ns
Output voltage fall time 38.4 kBaud			SPEED = low			5.2	μs
On time with overload	T_{ON_OL}	Short to supply, single over load		5		75	μs
Off time after overload Detection	$T_{OFF\_OL\_8V}$	V_{DDH} = 8 V		5	12	25	T_{ON_OL}
Off time after overload Detection	$T_{OFF\_OL\_36V}$	V_{DDH} = 36 V		15	35	80	T_{ON_OL}
Setup time TXD stable before transition TXEN = Low to TXEN = High	T_{setup}	Application information		1			μs
Hold time TXD stable after transition TXEN = High to TXEN = Low	T_{hold}	Application information		1			μs
Propagation Delay TXEN to transmitter enable after transition TXEN = Low to TXEN = High	T_{prop_txen}					1	μs



<R> **(2) Receiver**

(a) DC Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input threshold high	V_{THHs}	18 V < V_{DDH} < 30 V	10.5		13	V
Input threshold low	V_{THLs}	18 V < V_{DDH} < 30 V	8		11.5	V
Input threshold hysteresis	V_{HYSs}	18 V < V_{DDH} < 30 V	1	2.5	4	V
Receiver input	R_{RX}	–3 V < VCQ < V_{SUP} + 3 V	10	20	40	kΩ
V_{DDH} voltage range for IO-Link conform communication	V_{DDH}		18		30	V

Remark V_{SUP}: Supply voltage

(b) AC Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Receiver delay	T_{D_RX}	Information parameter		200	300	ns
Accepted minimum bit length (debounce window)	T_{BIT_MIN}		250		1000	ns

<R> **(3) Wake-up**

(a) DC Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Supply voltage at V_{DDH} for wake-up functionality	V_{DDH}		18		32	V

(b) AC Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Wake-up debounce time	T_{wu}	Single event of overload with receiver level change opposite to TXD	20		74	μs

<R> **(4) Temperature monitor**

(a) DC Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} = V_{DD_IO} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}/EV_{DD}, V_{SS}/EV_{SS} = AV_{SS} = GND1 = GND2 = 0 V)

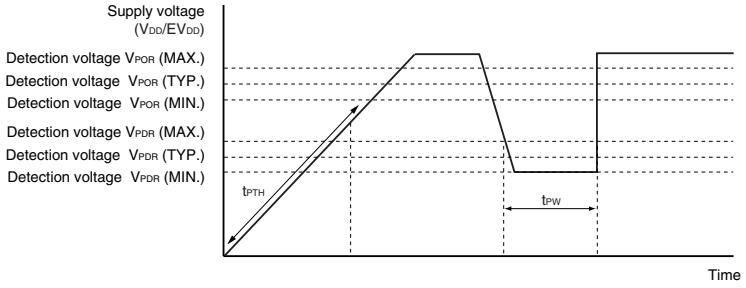
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Overtemperature threshold	T_{over}		155	175	200	°C

26.6.6 POC circuit characteristics

(T_A = –40 to +85°C, V_{SS}/EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.52	1.61	1.70	V
	V_{PDR}	Power supply fall time	1.50	1.59	1.68	V
Power supply voltage rise inclination	t_{PTH}	Change inclination of V_{DD}/EV_{DD}: 0 V → V_{POR}	0.5			V/ms
Minimum pulse width	t_{PW}	When the voltage drops	200			μs
Detection delay time					200	μs

POC Circuit Timing



26.6.7 Supply voltage rise time

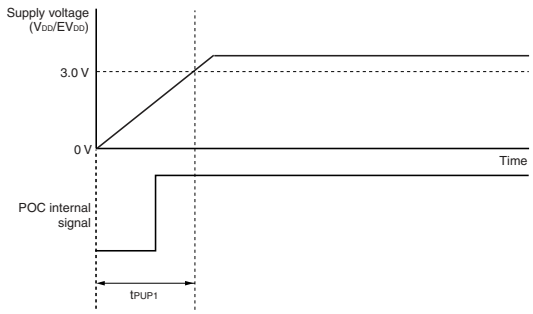
(T_A = –40 to +85°C, V_{SS}/EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 3.0 V (V_{DD}/EV_{DD} (MIN.))[Note] (V_{DD}/EV_{DD}: 0 V → 3.0 V)	t_{PUP1}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when $\overline{RESET}$ input is not used			6.0	ms
Maximum time to rise to 3.0 V (V_{DD}/EV_{DD} (MIN.))[Note] ($\overline{RESET}$ input → V_{DD}/EV_{DD}: 3.0 V)	t_{PUP2}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when $\overline{RESET}$ input is used			1.88	ms

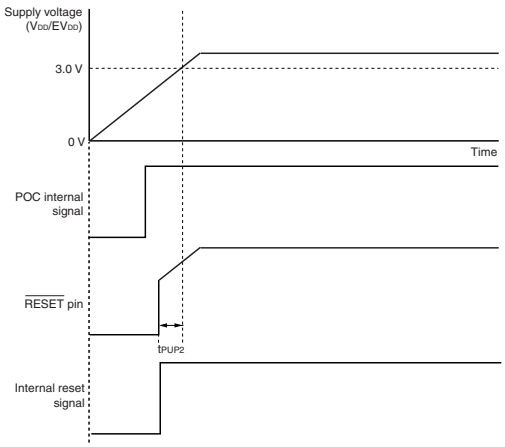
Note Make sure to raise the power supply in a shorter time than this.

Supply Voltage Rise Time Timing

- **When $\overline{RESET}$ pin input is not used**



- **When $\overline{RESET}$ pin input is used (when external reset is released by the $\overline{RESET}$ pin, after POC has been released)**



26.6.8 LVI circuit characteristics

LVI Circuit Characteristics (T_A = –40 to +85°C, $V_{PDR} \le V_{DD}/EV_{DD} = V_{DD\_IO} \le 5.5$ V, 4.75 V ≤ $IV_{DD} \le 5.25$ V, V_{SS}/EV_{SS} = GND1 = GND2 = 0 V)

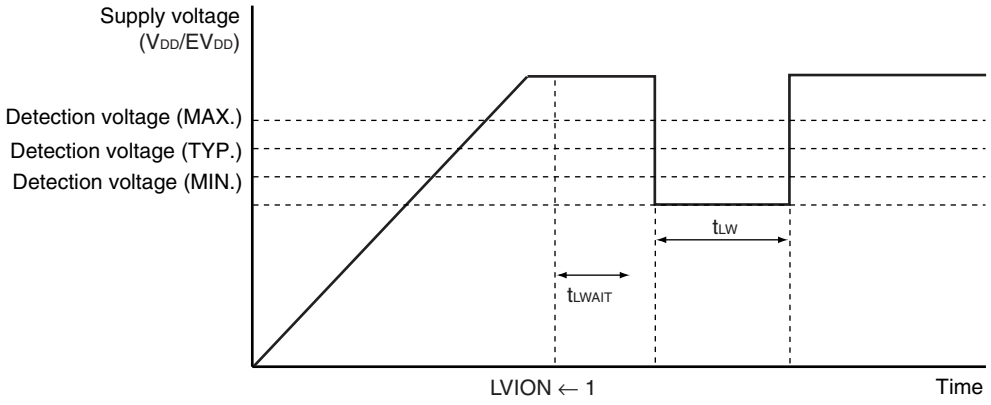
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.12	4.22	4.32	V
		V_{LVI1}		3.97	4.07	4.17	V
		V_{LVI2}		3.82	3.92	4.02	V
		V_{LVI3}		3.66	3.76	3.86	V
		V_{LVI4}		3.51	3.61	3.71	V
		V_{LVI5}		3.35	3.45	3.55	V
		V_{LVI6}		3.20	3.30	3.40	V
		V_{LVI7}		3.05	3.15	3.25	V
	External input pin[Note 1]	V_{EXLVI}	EXLVI < V_{DD}/EV_{DD}, 3.0 V ≤ $V_{DD}/EV_{DD} \le 5.5$ V	1.11	1.21	1.31	V
	Power supply voltage on power application	V_{PUPLVI}	When LVI default start function enabled is set	1.87	2.07	2.27	V
Minimum pulse width		t_{LW}		200			μs
Detection delay time						200	μs
Operation stabilization wait time[Note 2]		t_{LWAIT}				10	μs

Notes 1. The EXLVI/P120/INTP0 pin is used.

2. Time required from setting bit 7 (LVION) of the low-voltage detection register (LVIM) to 1 to operation stabilization

Remark $V_{LVI(n-1)} > V_{LVIn}$: n = 1 to 7

LVI Circuit Timing

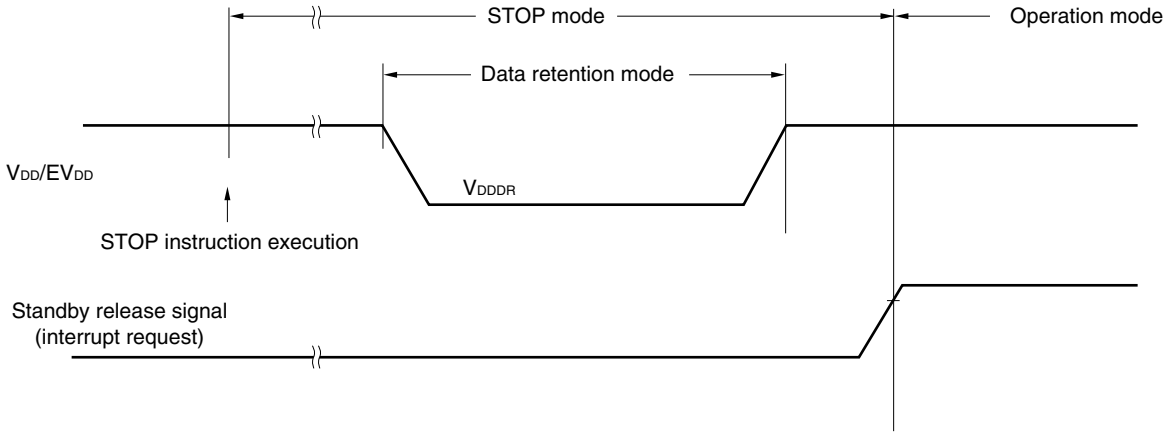


26.7 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics

(T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.5[Note]		5.5	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained before a POC reset is effected, but data is not retained when a POC reset is effected.



26.8 Flash Memory Programming Characteristics

(T_A = –40 to +85°C, 3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V, 4.75 V ≤ IV_{DD} ≤ 5.25 V, V_{SS}/EV_{SS} = GND1 = GND2 = 0 V)

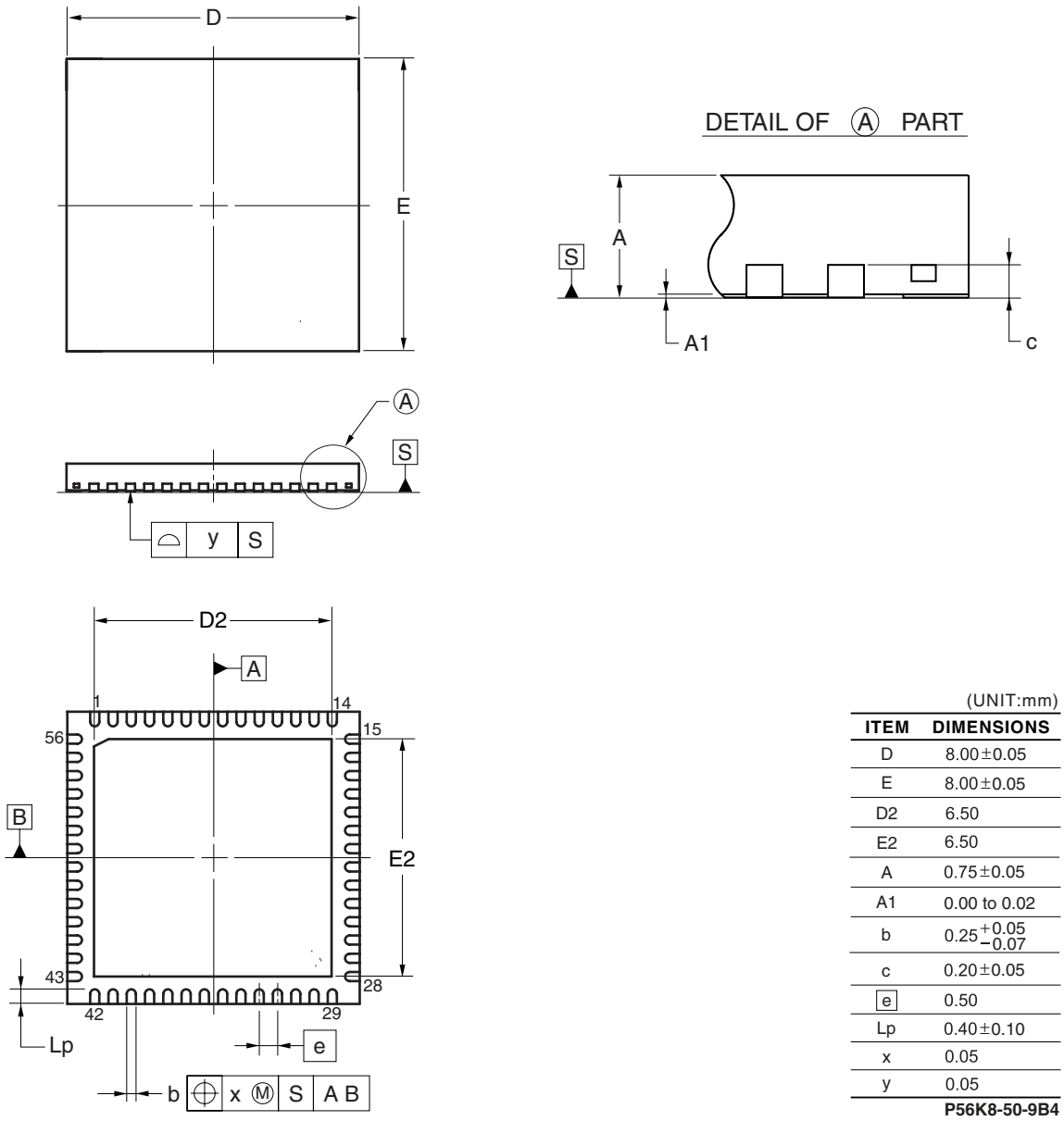
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
V_{DD}/EV_{DD} supply current	I_{DD}	Typ. = 10 MHz, Max. = 20 MHz			6	20	mA
CPU/peripheral hardware clock frequency	f_{CLK}	3.0 V ≤ V_{DD}/EV_{DD} ≤ 5.5 V		2		20	MHz
Number of rewrites (number of deletes per block)	C_{erwr}	Used for updating programs When using flash memory programmer and Renesas Electronics self programming library	Retained for 15 years	1,000			Times
		Used for updating data When using Renesas Electronics EEPROM emulation library	Retained for 5 years	10,000			Times

Remark When updating data multiple times, use the flash memory as one for updating data.

CHAPTER 27 PACKAGE DRAWINGS

(1) PD78F8040K8-9B4-AX, PD78F8041K8-9B4-AX, PD78F8042K8-9B4-AX, PD78F8043K8-9B4-AX

56-PIN PLASTIC WQFN(8x8)



(UNIT:mm)

ITEM	DIMENSIONS
D	8.00±0.05
E	8.00±0.05
D2	6.50
E2	6.50
A	0.75±0.05
A1	0.00 to 0.02
b	$0.25^{+0.05}_{-0.07}$
c	0.20±0.05
e	0.50
Lp	0.40±0.10
x	0.05
y	0.05

P56K8-50-9B4

(2) PD78F8040F1-AD1-AX, PD78F8041F1-AD1-AX, PD78F8042F1-AD1-AX, PD78F8043F1-AD1-AX

T.B.D.

<R> CHAPTER 28 RECOMMENDED SOLDERING CONDITIONS

These products should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, please contact a Renesas Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www2.renesas.com/pkg/en/mount/index.html)

Table 28-1. Surface Mounting Type Soldering Conditions

(1) 56-pin plastic QFN (8x8)

PD78F8040K8-9B4-AX, 78F8041K8-9B4-AX, 78F8042K8-9B4-AX, 78F8043K8-9B4-AX

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: 3 times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	IR60-107-3

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Cautions 1. Do not use different soldering methods together (except for partial heating).

2. The PD78F8040, 78F8041, 78F8042, 78F8043 microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

(2) 56-pin plastic FBGA (4x7)

PD78F8040F1-AD1-AX, 78F8041F1-AD1-AX, 78F8042F1-AD1-AX, 78F8043F1-AD1-AX

T.B.D.

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the μPD78F8040, 78F8041, 78F8042, 78F8043.

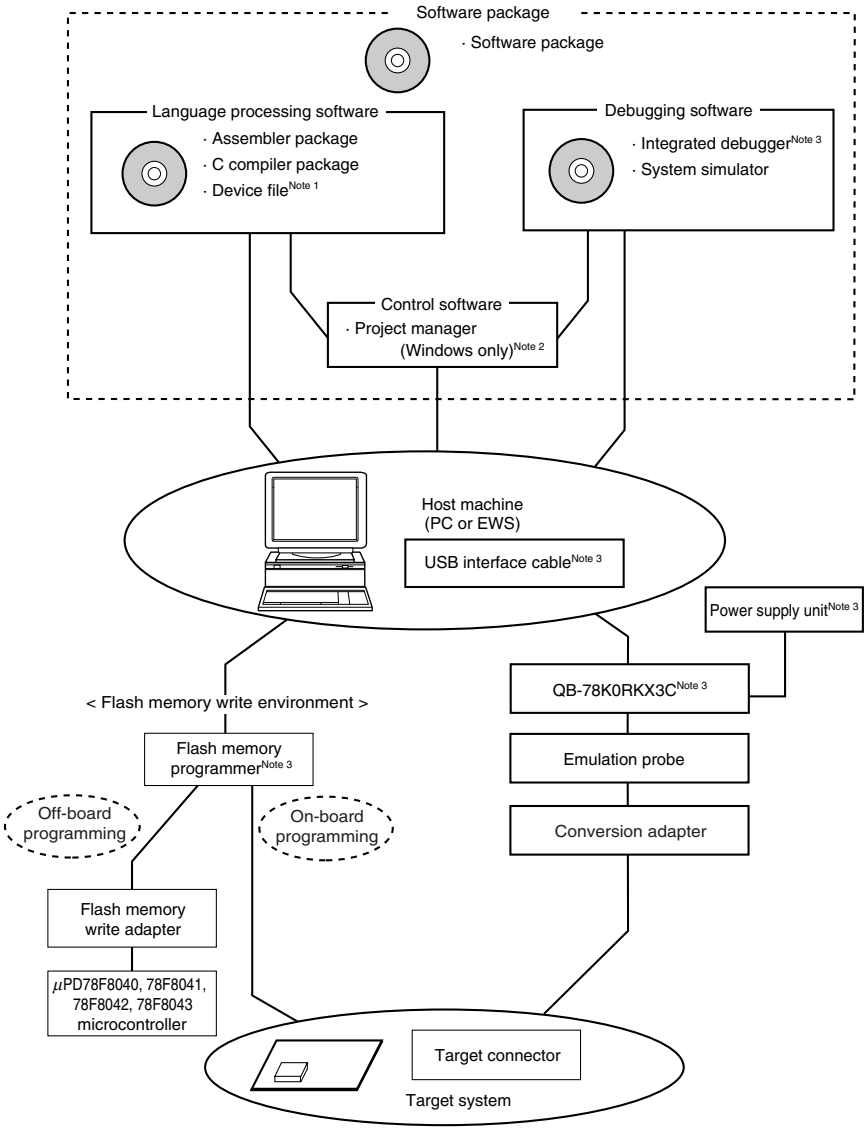
Figure A-1 shows the development tool configuration.

- **Support for PC98-NX series**
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows™**
  Unless otherwise specified, “Windows” means the following OSs.
  - Windows 98
  - Windows NT™
  - Windows 2000
  - Windows XP

Figure A-1. Development Tool Configuration (1/2)

(1) When using the in-circuit emulator QB-78K0RKX3C



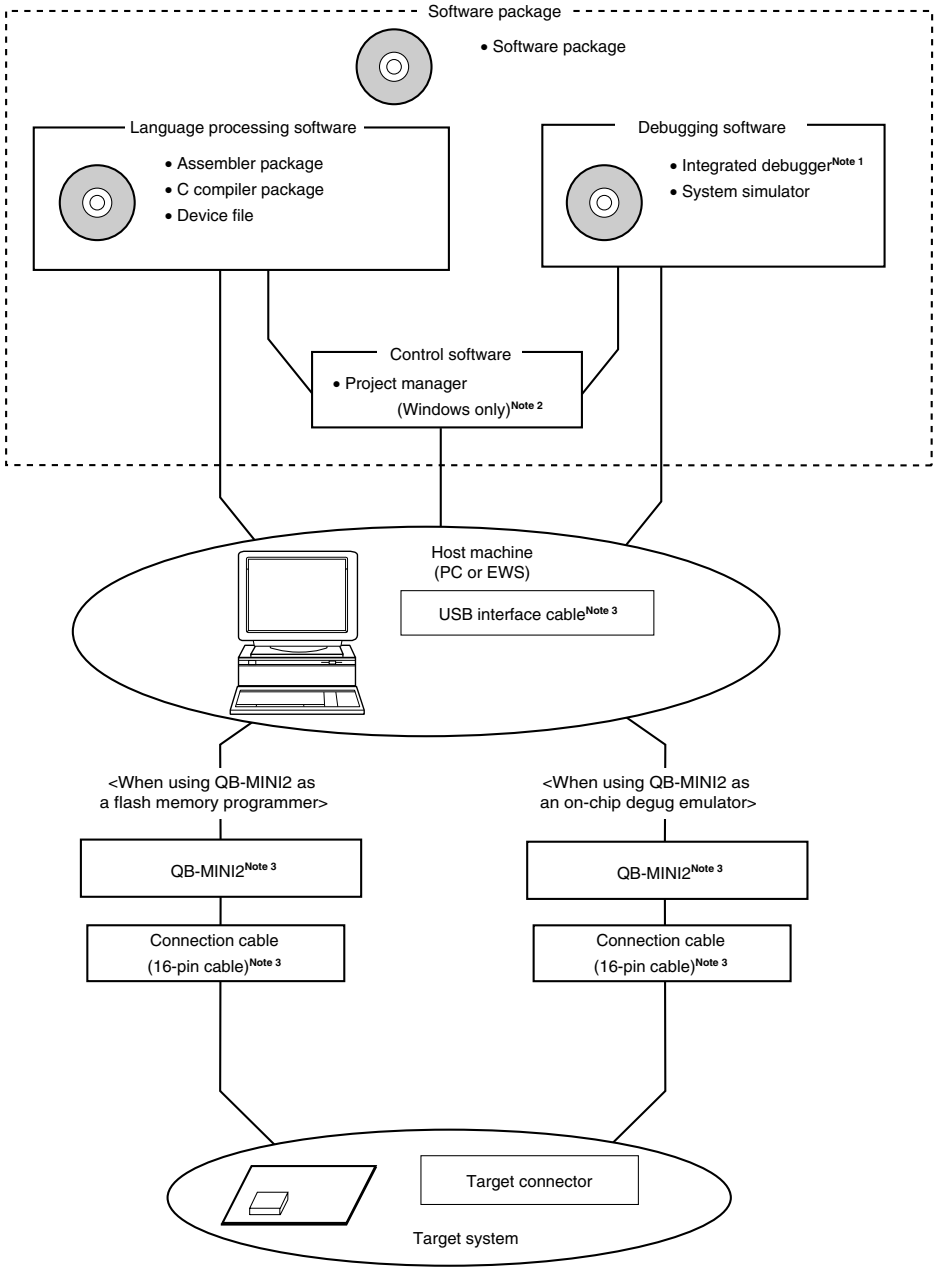
Notes 1. Download the device file for the μPD78F8040 to 78F8043 (DF788043) from the download site for development tools (http://www2.renesas.com/micro/ods/eng/index.html). <R>

2. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

3. In-circuit emulator QB-78K0RKX3C is supplied with integrated debugger ID78K0R-QB, on-chip debug emulator with programming function QB-MINI2 and USB interface cable. Any other products are sold separately.

Figure A-1. Development Tool Configuration (2/2)

(2) When using the on-chip debug emulator with programming function QB-MINI2



Notes 1. Download the integrated debugger (ID78K0R-QB) from the download site for development tools
<R> (http://www2.renesas.com/micro/ods/eng/index.html).

2. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

3. On-chip debug emulator QB-MINI2 is supplied with USB interface cable, connection cables (10-pin cable and 16-pin cable), and 78K0-OCD board. Any other products are sold separately. In addition, download the software for operating the QB-MINI2 from the download site for MINICUBE2
<R> (http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html).

A.1 Software Package

SP78K0R 78K0R Series software package	Development tools (software) common to the 78K0R microcontrollers are combined in this package.
	Part number: μS××××SP78K0R

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP78K0R

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

A.2 Language Processing Software

RA78K0R Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. This assembler is also provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF788043[Note]). **<Precaution when using RA78K0R in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××RA78K0R
CC78K0R C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file (both sold separately). **<Precaution when using CC78K0R in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××CC78K0R
DF788043[Note] Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0R, CC78K0R, SM+ for 78K0R, and ID78K0R-QB) (all sold separately). The corresponding OS and host machine differ depending on the tool to be used.
	Part number: μS××××DF788043

Note The DF788043 can be used in common with the RA78K0R, CC78K0R, SM+ for 78K0R, and ID78K0R-QB.

Remark ××××in the part number differs depending on the host machine and OS used.

μS××××RA78K0R
μS××××CC78K0R

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

μS××××DF788043

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	

A.3 Control Software

PM+ Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from the project manager. **<Caution>** The project manager is included in the assembler package (RA78K0R). It can only be used in Windows.

A.4 Flash Memory Programming Tools

A.4.1 When using flash memory programmers PG-FP5, FL-PR5

PG-FP5, FL-PR5 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
<R> FA-78F8043K8-9B4-RX FA-78F8043F1-AD1-RX Note Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use.

Note Under development

Remark The FL-PR5 and FA-78F8043K8-9B4-RX are products of Naito Densei Machida Mfg. Co., Ltd.

A.4.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This is a flash memory programmer dedicated to microcontrollers with on-chip flash memory. It is available also as on-chip debug emulator which serves to debug hardware and software when developing application systems using the 78K0R. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use μPD78F8040, 78F8041, 78F8042, 78F8043, use USB interface cable and 16-pin connection cable.

Remark Download the software for operating the QB-MINI2 from the download site for MINICUBE2
<R> (http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html).

A.5 Debugging Tools (Hardware)

A.5.1 When using in-circuit emulator QB-78K0RKX3C

QB-78K0RKX3C In-circuit emulator	This in-circuit emulator serves to debug hardware and software when developing application systems using the μPD78F8040, 78F8041, 78F8042, 78F8043. It supports to the integrated debugger (ID78K0R-QB). This emulator should be used in combination with a power supply unit and emulation probe, and the USB is used to connect this emulator to the host machine.
QB-COMMON-PW-××[Note 1]	This power supply unit can be used in common with the all products of in-circuit emulator IECUBE and the flash memory programmer PG-FP5.
QB-144-CA-01 Check pin adapter	This check pin adapter is used in waveform monitoring using the oscilloscope, etc.
QB-144-EP-02S Emulation probe	This emulation probe is flexible type and used to connect the in-circuit emulator and target system.
<R> QB-78F8043-EA-01T QB-78F8043-EA-02T[Note 2] Exchange adapter	This exchange adapter is used to perform pin conversion from the in-circuit emulator to target connector. QB-78F8043-EA-01T: For QFN package products QB-78F8043-EA-02T: For FBGA package products
<R> QB-56K8-NQ-01T QB-56F1-NQ-01T[Note 2] Target connector	This target connector is used to mount on the target system. QB-56K8-NQ-01T: For QFN package products QB-56F1-NQ-01T: For FBGA package products

Notes 1. ×× differs depending on the target area of each product.

2. Under development

Remark The QB-78K0RKX3C is supplied with a USB interface cable, integrated debugger ID78K0R-QB, and on-chip debug emulator with programming function QB-MINI2.

A.5.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0R microcontrollers. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use μPD78F8040, 78F8041, 78F8042, 78F8043, use USB interface cable and 16-pin connection cable.

Remark Download the software for operating the QB-MINI2 from the download site for MINICUBE2
<R> (http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html).

A.6 Debugging Tools (Software)

SM+ for 78K0R System simulator	SM+ for 78K0R is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of SM+ for 78K0R allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. SM+ for 78K0R should be used in combination with the device file (DF788043).
	Part number: μS××××SM781000
ID78K0R-QB Integrated debugger	This debugger supports the in-circuit emulators for the 78K0R microcontrollers. The ID78K0R-QB is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. ID78K0R-QB should be used in combination with the device file.
	Part number: μS××××ID78K0R-QB

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM781000
μS××××ID78K0R-QB

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

APPENDIX B REVISION HISTORY

B.1 Major Revisions in This Edition

Page	Description	Classification
CHAPTER 1 OUTLINE		
pp.2, 4, 9	Addition of the FBGA package products	(d)
CHAPTER 2 CONNECTION BETWEEN MCU AND IO-LINK TRANSCEIVER		
p.13	Addition of **Caution** to **2.3.2 Using internal regulator of IO-Link transceiver**	(c)
CHAPTER 22 FLASH MEMORY		
p.593	Addition of Pin No. (for FBGA package products) to **Table 22-1. Wiring Between μPD78F8040, 78F8041, 78F8042, 78F8043 and Dedicated Flash Memory Programmer**	(d)
CHAPTER 26 ELECTRICAL SPECIFICATIONS		
p.637	Addition of **Caution** of target specifications for the FBGA package products	(d)
pp.641, 642	Addition of I_{OH1}, I_{OL1} specificatons and **Cautions** to **26.4.1 Pin Characteristics**	(b)
pp.643, 644	Addition of V_{IH7}, V_{IL7} specificatons to **26.4.1 Pin Characteristics**	(b)
p.645	Addition of V_{OH3}, V_{OL4} specificatons to **26.4.1 Pin Characteristics**	(b)
p.651	Addition of **Caution** to **Figure of the current paths for the different power supply configurations**	(b)
pp.677 to 679	Addition of specificatons to **26.6.5 IO-Link transceiver characteristics**	(b)
CHAPTER 28 RECOMMENDED SOLDERING CONDITIONS		
Throughout	Addition of chapter	(b)
APPENDIX A DEVELOPMENT TOOLS		
pp.687, 688, 690, 691	Change of URL of Renesas Electronics website	(c)
p.690	Addition of a flash memory programming adapter to **A.4.1 When using flash memory programmers PG-FP5, FL-PR5**	(d)
p.691	Addition of an exchange adapter and a target connector to **A.5.1 When using in-circuit emulator QB-78K0RKX3C**	(d)

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

B.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/5)

Edition	Description	Chapter
2nd Edition	Change of specifications from target specifications to formal specifications, deletion of "target" in capacitance values of the capacitor connecting to REGC and IV_{DD} pins, and deletion of TIS0, TIS1, PIM1, and POM1 registers	**Throughout**
	Change of **Documents Related to Devices**	**INTRODUCTION**
	Change of on-chip internal high-speed oscillation clocks in **1.1 Features**	**CHAPTER 1 OUTLINE**
	Addition of Internal reset by a reset processing check error	
	Addition of description to **3.2.15 $\overline{\text{RESET}}$**	**CHAPTER 3 PIN FUNCTIONS**
	Addition of **Note** to **Figure 5-1** to **Figure 5-4**	**CHAPTER 5 CPU ARCHITECTURE**
	Deletion of TIS0 and TIS1 register of **Table 5-5. SFR List (2/4)**	
	Change of **Figure 6-19. Format of Port Mode Register**	**CHAPTER 6 PORT FUNCTIONS**
	Change of **Figure 6-20. Format of Port Register**	
	Addition of **Caution 3** to **Figure 6-24. Format of A/D Port Configuration Register (ADPC)**	
	Change of description of AMPH bit in **Figure 7-2. Format of Clock Operation Mode Control Register (CMC)**	**CHAPTER 7 CLOCK GENERATOR**
	Change of **Figure 7-3. Format of System Clock Control Register (CKC)**	
	Change of **Table 7-2. Relationship Between CPU Clock and Minimum Instruction Execution Time**	
	Addition of **Note** and **Caution 5** to **Figure 7-4. Format of Clock Operation Status Control Register (CSC)**	
	Change of **Caution 3** of **Figure 7-6. Format of Oscillation Stabilization Time Select Register (OSTS)**	
	Addition of **Caution 3** to **Figure 7-7. Format of 20 MHz Internal High-Speed Oscillation Control Register (DSCCTL)**	
	Addition of description to **7.3 (7) Peripheral enable register 0 (PER0)**	
	Change of **Caution 2, 3, and 6** of **Figure 7-9. Format of Operation Speed Mode Control Register (OSMC)**	
	Change of **7.4.2 Internal high-speed oscillator**	
	Change of **Figure 7-12. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))**	
	Change of **Figure 7-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0))**	
	Change of **7.6 Controlling Clock**	
	Addition of **Note 2** to **7.6.5 CPU clock status transition diagram**	
	Change of **Table 7-4. CPU Clock Transition and SFR Register Setting Examples (1/3) (2) CPU operating with high-speed system clock (C) after reset release (A)** and Addition of **Remark**	
	Change of **Table 7-4. CPU Clock Transition and SFR Register Setting Examples (2/3) (4) CPU clock changing from internal high-speed oscillation clock (B) to high-speed system clock (C)** and Addition of **Remark**	
	Change of **Table 7-8. Maximum Number of Clocks Required for $f_{IH} \leftrightarrow f_{MX}$**	

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

(2/5)

Edition	Description	Chapter
2nd Edition	Change of **8.1.2 Functions of each channel when it operates with another channel**	**CHAPTER 8 TIMER ARRAY UNIT**
	Change of **Figure 8-1. Entire Configuration of Timer Array Unit 0**	
	Change of **Figure 8-2. Entire Configuration of Timer Array Unit 1**	
	Addition of **Figure 8-3. Internal Block Diagram of Channel of Timer Array Unit 0**	
	Change of **Figure 8-7. Format of Timer Clock Select Register m (TPSm)**	
	Addition of **Cautions** to **Figure 8-8. Format of Timer Mode Register mn (TMRmn) (1/3)**	
	Change of **Remark** of **Figure 8-21. Format of Timer Output Mode Register m (TOMm)**	
	Change of **Figure 8-22. Format of Input Switch Control Register (ISC)**	
	Change of **Figure 8-25. Format of Port Mode Registers 0, 1, 3, 6 (PM0, PM1, PM3, PM6)**	
	Change of **Remark** of **8.5.1 TOmn pin output circuit configuration**	
	Change of **Figure 8-40. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)**	
	Change of **Figure 8-44. Operation Procedure When External Event Counter Function Is Used**	
	Change of **Figure 8-48. Operation Procedure When Input Pulse Interval Measurement Function Is Used**	
	Addition of **Caution** to **8.7.4 Operation as input signal high-/low-level width measurement**	
	Change of **Figure 8-52. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used**	
	Change of **Remark** of **8.8 Simultaneous Channel Operation Function of Timer Array Unit**	
	Change of description of **8.8.2 Operation as PWM function**	
	Change of **Table 9-3. Setting of Overflow Time of Watchdog Timer**	**CHAPTER 9 WATCHDOG TIMER**
	Change of **9.4.3 Setting window open period of watchdog timer (**Deletion of if the window open period is 25%**)**	
	Change of **Figure 10-5. A/D Converter Sampling and A/D Conversion Timing**	**CHAPTER 10 A/D CONVERTER**
	Change of **Figure 10-10. Formats of Port Mode Registers 2 and 15 (PM2, PM15)**	
	Change of **Figure 10-11. Basic Operation of A/D Converter**	
	Change of **Figure 10-13. Example of Select Mode Operation Timing**	
	Change of description of **10.6 (9) Conversion results just after A/D conversion start**	
	Change of **Table 10-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**	
	Addition of **10.6 (12) Starting the A/D converter**	
	Deletion of PIM1 and POM1 registers	**CHAPTER 11 SERIAL ARRAY UNIT**
	Change of **Figure 11-2. Block Diagram of Serial Array Unit 1**	
	Change of **Figure 11-5. Format of Serial Clock Select Register m (SPSm)**	
	Addition of **Caution** to **Figure 11-6. Format of Serial Mode Register mn (2/2) (SMRmn)**	
	Addition of **Note** to **Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)**	
	Change of description of **11.3 (5) Serial data register mn (SDRmn)**	
	Change of **Figure 11-9. Format of Serial Flag Clear Trigger Register mn (SIRmn)**	
	Change of **Figure 11-10. Format of Serial Status Register mn (SSRmn)**	

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

(3/5)

Edition	Description	Chapter
2nd Edition	Change of **Figure 11-11. Format of Serial Channel Start Register m (SSm)** and addition of **Note**	**CHAPTER 11 SERIAL ARRAY UNIT (continuation)**
	Change of **Figure 11-18. Format of Noise Filter Enable Register 0 (NFEN0)**	
	Change of **Figure 11-24. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI20)**	
	Change of **11.5.2 Master reception**	
	Change of **Figure 11-32. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI20)**	
	Modification of **Figure 11-37. Flowchart of Master Reception (in Single-Reception Mode)**	
	Addition of **Figure 11-38. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAP10 = 0, CKP10 = 0)**	
	Addition of **Figure 11-39. Flowchart of Master Reception (in Continuous Reception Mode)**	
	Change of **Figure 11-40. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI20)**	
	Change of **Figure 11-45. Flowchart of Master Transmission/Reception (in Single-Transmission/Reception Mode)**	
	Change of **Figure 11-47. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**	
	Change of **Figure 11-48. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI20)**	
	Modification of **Figure 11-53. Flowchart of Slave Transmission (in Single-Transmission Mode)**	
	Modification of **Figure 11-55. Flowchart of Slave Transmission (in Continuous Transmission Mode)**	
	Change of **Figure 11-56. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI20)**	
	Change of **Figure 11-62. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI20)** and addition of **Caution**	
	Addition of **Caution** to **Figure 11-63. Initial Setting Procedure for Slave Transmission/Reception**	
	Addition of **Caution** to **Figure 11-65. Procedure for Resuming Slave Transmission/Reception**	
	Modification of **Figure 11-67. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)** and addition of **Caution**	
	Modification of **Figure 11-69. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)** and addition of **Caution**	
	Change of description of **11.6 Operation of UART (UART0, UART2, UART3) Communication**	
	Change of **Figure 11-71. Example of Contents of Registers for UART Transmission of UART (UART2, UART3) (1/2)**	
	Change of **Figure 11-79. Example of Contents of Registers for UART Reception of UART (UART2, UART3) (1/2)**	
	Change of **Figure 11-89. Flowchart for LIN Transmission**	
	Change of **Figure 11-93. Setting Procedure for Communicating with IO-Link Transceiver**	
	Change of **11.7.5 IO-Link transmission**	
	Change of **Figure 11-94. Initial Setting Procedure for UART Transmission**	
	Change of **Figure 11-95. Example of Contents of Registers for Address Field Transmission of Simplified I^2C (IIC20)**	

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

(4/5)

Edition	Description	Chapter
2nd Edition	Change of **12.6 Timing Charts**	**CHAPTER 12 SERIAL INTERFACE IICA**
	Addition of **Note** to **Figure 14-4. Format of DMA Mode Control Register n (DMCn) (1/2)**	**CHAPTER 14 DMA CONTROLLER**
	Change of **Caution 2** of **14.6 (2) DMA response time**	
	Change of **14.6 (4) DMA pending instruction**	
	Change of **15.4.4 Interrupt request hold**	**CHAPTER 15 INTERRUPT FUNCTIONS**
	Change of **Table 16-1. Operating Statuses in HALT Mode**	**CHAPTER 16 STANDBY FUNCTION**
	Change of **Figure 16-4. HALT Mode Release by Reset**	
	Change of **Table 16-2. Operating Statuses in STOP Mode**	
	Change of **Figure 16-5. STOP Mode Release by Interrupt Request Generation (1/2)**	
	Change of **Figure 16-6. STOP Mode Release by Reset**	
	Addition of (6) Internal reset by a reset processing check error	**CHAPTER 17 RESET FUNCTION**
	Change of **Figure 17-1. Block Diagram of Reset Function**	
	Change of **Figure 17-2. Timing of Reset by $\overline{\text{RESET}}$ Input**	
	Change of **Figure 17-3. Timing of Reset Due to Execution of Illegal Instruction or Watchdog Timer Overflow**	
	Change of **Figure 17-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input**	
	Change of **Table 17-2. Hardware Statuses After Reset Acknowledgment (3/3)**	
	Change of **Note 2** of **Table 17-2. Hardware Statuses After Reset Acknowledgment (3/3)**	
	Change of **Figure 17-5. Format of Reset Control Flag Register (RESF)**	
	Change of **Table 17-3. RESF Status When Reset Request Is Generated**	
	Change of **Figure 18-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector**	**CHAPTER 18 POWER-ON-CLEAR CIRCUIT**
	Change of **Figure 18-3. Example of Software Processing After Reset Release**	
	Change of **Figure 19-11. Example of Software Processing After Reset Release**	**CHAPTER 19 LOW-VOLTAGE DETECTOR**
	Change of description of **20.1 Regulator Overview**	**CHAPTER 20 REGULATOR**
	Change of **Caution 2** of **Figure 20-1. Format of Regulator Mode Control Register (RMC)**	
	Change of **Figure 21-1. Format of User Option Byte (000C0H/010C0H)**	**CHAPTER 21 OPTION BYTE**
	Change of **21.4 Setting of Option Byte**	
	Addition of PG-FP5, FL-PR5, and QB-MINI2 as dedicated flash memory programmer and deletion of PG-FP4 and FL-PR4	**CHAPTER 22 FLASH MEMORY**
	Change of transfer rate of **22.3 Communication Mode**	
	Change of description of **22.4.5 REGC pin**	
	Change of description of **22.8 Flash Memory Programming by Self-Programming** and addition of **Remark**	
	Change of **Figure 22-10. Flow of Self Programming (Rewriting Flash Memory)**	

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

(5/5)

Edition	Description	Chapter
2nd Edition	Change of **Table 23-1. Differences Between 1-Line Mode and 2-Line Mode**	**CHAPTER 23 ON-CHIP DEBUG FUNCTION**
	Addition of maximum permissible current to **Absolute Maximum Ratings (T_A = 25°C) (2/2)**	**CHAPTER 26 ELECTRICAL SPECIFICATIONS**
	Change of internal low-speed oscillation clock frequency of **26.3.2 Internal oscillator characteristics**	
	Change of **26.4.2 Supply current characteristics**	
	Change of instruction cycle (minimum instruction execution time), TI01 to TI03, TI05, TI11, TI13 input high-level width, low-level width , and TO01 to TO03, TO05, TO11, TO13 output frequency of **26.5.1 Basic operation**	
	Change of **Minimum instruction execution time during self programming mode (RMC = 00H)** in **26.5 AC Characteristics**	
	Change of SCL0 clock frequency of **26.6.2 Serial interface IICA**	
	Change of **26.6.4 A/D converter characteristics**	
	Change of **(1) When using the in-circuit emulator QB-78K0RKX3C** of **Figure A-1. Development Tool Configuration**	**APPENDIX A DEVELOPMENT TOOLS**
	Change of **(2) When using the on-chip debug emulator with programming function QB-MINI2** of **Figure A-1. Development Tool Configuration**	
	Change of **A.2 Language Processing Software**	
	Change of **A.4.1 When using flash memory programmers PG-FP5, FL-PR5**	
	Change of **A.5.1 When using in-circuit emulator QB-78K0RKX3C**	
	Addition of chapter	**APPENDIX B REVISION HISTORY**

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

PD78F8040, 78F8041, 78F8042, 78F8043
User's Manual: Hardware

Publication Date: Rev.0.01 Jun 13, 2009
Rev.3.00 Sep 28, 2010

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, 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 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

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.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

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.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.0

PD78F8040,
78F8041,
78F8042,
78F8043

Renesas Electronics Corporation

R01UH0119EJ0300
(Previous Number: U19680EJ2V0UD00)