NEC

Preliminary User’s Manual

μPD79F9211

16-bit Single-Chip Microcontrollers

Date Published February 2008 NS CP(K)

[MEMO]

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 V_{IL} (MAX) and V_{IH} (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 V_{IL} (MAX) and V_{IH} (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 V_{DD} 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.

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

- **The information contained in this document is being issued in advance of the production cycle for the product. The parameters for the product may change before final production or NEC Electronics Corporation, at its own discretion, may withdraw the product prior to its production.**
- **Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC 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 NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific". The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics products depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

  "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.

  "Special": 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, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M5D 02.11-1

INTRODUCTION

Readers

This manual is intended for user engineers who wish to understand the functions of the μPD79F9211 and design and develop application systems and programs for these devices.
The target products are as follows.

μPD79F9210, μPD79F9211

Purpose

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

Organization

The μPD79F9211 manual is separated into two parts: this manual and the instructions edition (common to the 78K0R Microcontroller Series).

μPD79F9211
User's Manual
(This Manual)

- Pin functions
- Internal block functions
- Interrupts
- Other on-chip peripheral functions
- Electrical specifications (target)

78K0R Microcontroller
User's Manual
Instructions

- 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**.
- 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 Series 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.
μPD79F9211 User's Manual	This manual
78K0R Microcontroller Instructions User's Manual	U17792E

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

Document Name		Document No.
CC78K0R Ver. 1.00 C Compiler	Operation	U17838E
	Language	U17837E
RA78K0R Ver. 1.00 Assembler Package	Operation	U17836E
	Language	U17835E
SM+ System Simulator	Operation	U18010E
PM+ Ver. 6.20		U17990E
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-78K0RKX3L In-Circuit Emulator	To be prepared

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP4 Flash Memory Programmer User's Manual	U15260E
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

Note See the “Semiconductor Device Mount Manual” website (http://www.necel.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.

CONTENTS

CHAPTER 1 OUTLINE

The μPD79F9211 is a 16-bit single-chip microcontroller that uses a 78K0R CPU core and incorporates peripheral functions, such as ROM/RAM, a multi-function timer, a multi-function serial interface, an A/D converter, a programmable gain amplifier (PGA), a comparator, a real-time counter, and a watchdog timer.

This product has been developed for inverter control applications that are controlled by using two chips. The multi-function timer (timer array unit TAUS) that can perform various PWM operations operates at a maximum resolution of 40 MHz and is provided with several functions, such as a PWM (complementary PWM × 2 channels) output function with dead time, a 6-phase PWM output function with dead time, and a DC inverter real-time output function, and can perform inverter control. Furthermore, the multi-function timer is equipped with two channels of comparators for fail-safe applications and can set PWM to high impedance when an overcurrent occurs.

In consideration of motor applications, the internal high-speed oscillator (CPU clock 20 MHz, timer 40 MHz) is mounted. This can solve problems such as the resonator coming off.

The μPD79F9211 provides high cost performance in various situations.

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 ultra low-speed (61 μs: @ 32.768 kHz operation with subsystem clock)
- 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)
μPD79F9211[Note]		16 KB	1 KB

Note Under development

- 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 internal low-speed oscillation clock)
- On-chip multiplier/divider (16 bits × 16 bits, 32 bits ÷ 32 bits)
- On-chip BCD adjustment
- I/O ports: 37
- Timer: 14 channels
  - 16-bit timer: 12 channels
  - Watchdog timer: 1 channel
  - Real-time counter: 1 channel
  - On-chip motor control option unit
- On-chip comparator/operational amplifier function
- Serial interface
  - CSI: 2 channels/UART (LIN-bus supported): 1 channel
  - CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel
- 10-bit resolution A/D converter (AV_{REF} = 2.7 to 5.5 V)
  10 channels
- Power supply voltage: V_{DD} = 2.7 to 5.5 V
- Operating ambient temperature: T_A = −40 to +85°C

1.2 Applications

○ Electric bicycles

1.3 Ordering Information

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

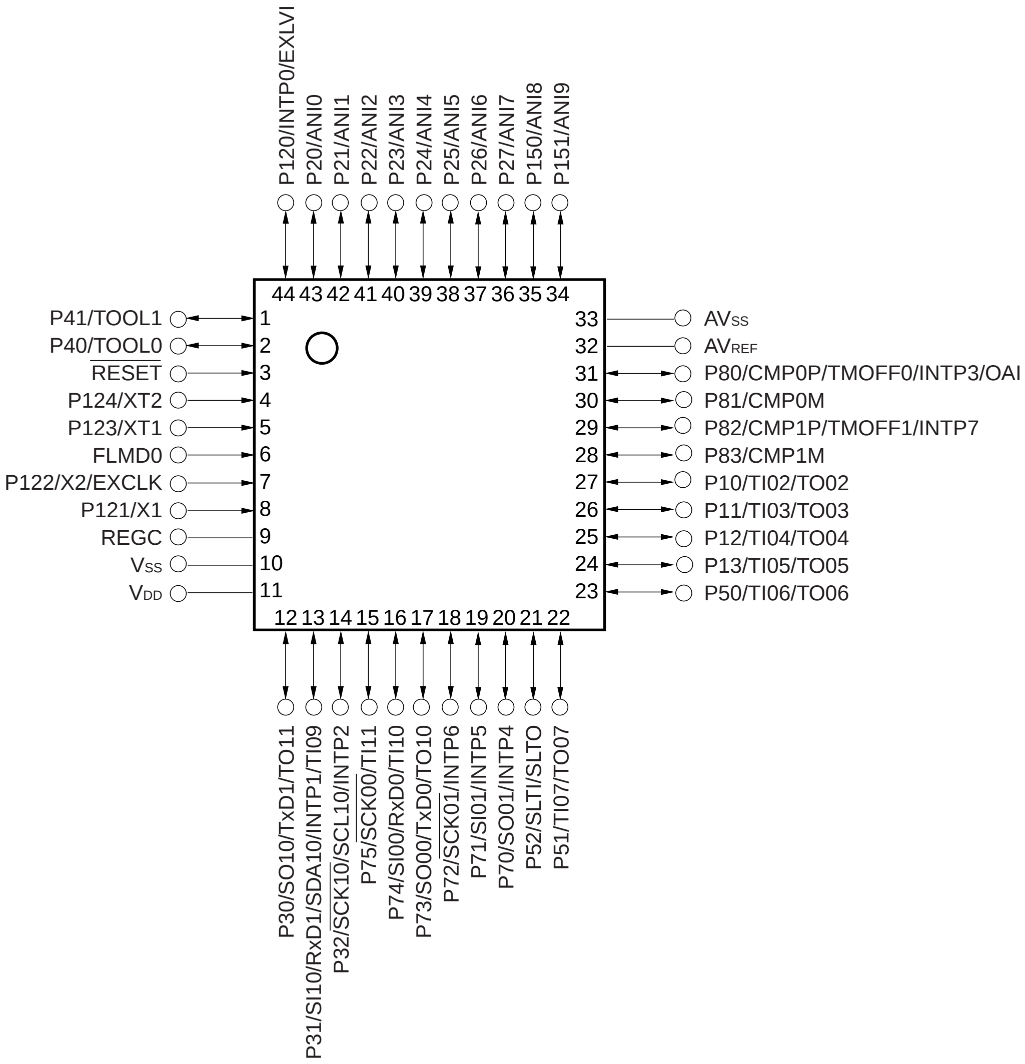
Part Number	Package
μPD79F9211GB-GAF-AX**Note**	44-pin plastic LQFP (10 × 10)

Note Under development

Caution The μPD79F9211 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

1.4 Pin Configuration (Top View)

- 44-pin plastic LQFP (10 × 10)[Note]



Note Under development

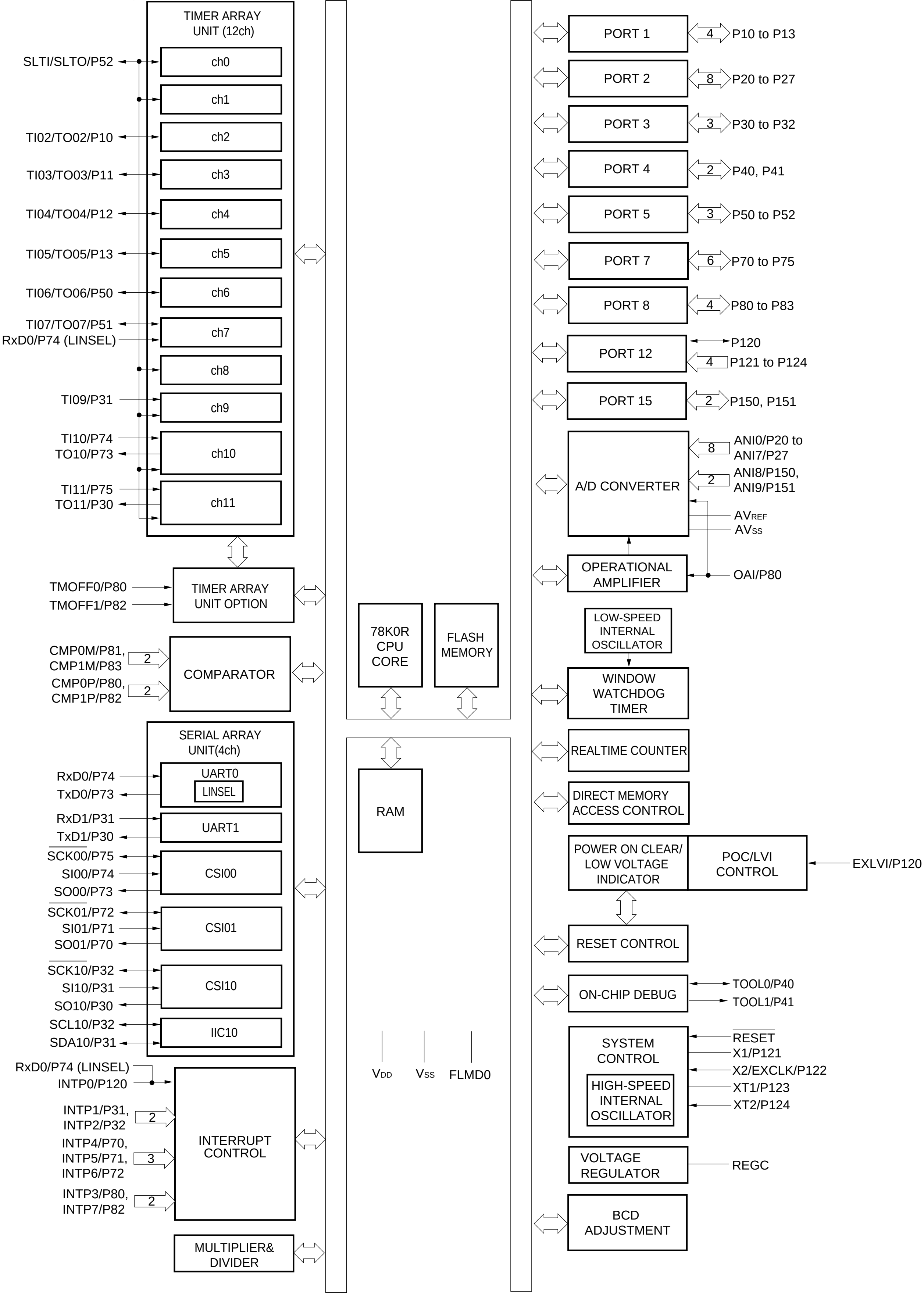
Cautions 1. Make AVSS the same potential as VSS.

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

Pin Identification

ANI0 to ANI9:	Analog input	$\overline{\text{RESET}}$:	Reset
AV$_{\text{REF}}$:	Analog reference voltage	RxD0, RxD1:	Receive data
AV$_{\text{SS}}$:	Analog ground	$\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$:	Serial clock input/output
CMP0M, CMP1M:	Comparator input (minus)	SCL10:	Serial clock input/output
CMP0P, CMP1P:	Comparator input (plus)	SDA10:	Serial data input/output
EXCLK:	External clock input	SI00, SI01, SI10:	Serial data input
	(main system clock)	SLTI:	Selectable timer input
EXLVI:	External potential input	SLTO:	Selectable timer output
	for low-voltage detector	SO00, SO01, SO10:	Serial data output
FLMD0:	Flash programming mode	TI02 to TI07, TI09 to TI11:	Timer input
INTP0 to INTP7:	External interrupt input	TMOFF0, TMOFF1:	Timer off input
OAI:	Operational amplifier input	TO02 to TO07, TO10, TO11:	Timer output
P10 to P13:	Port 1	TOOL0:	Data input/output for tool
P20 to P27:	Port 2	TOOL1:	Clock output for tool
P30 to P32:	Port 3	TxD0, TxD1:	Transmit data
P40, P41:	Port 4	V$_{\text{DD}}$:	Power supply
P50 to P52:	Port 5	V$_{\text{SS}}$:	Ground
P70 to P75:	Port 7	X1, X2:	Crystal oscillator (main system
P80 to P83:	Port 8		clock)
P120 to P124:	Port 12	XT1, XT2:	Crystal oscillator (subsystem
P150, P151:	Port 15		clock)
REGC:	Regulator capacitance		

1.5 Block Diagram



1.6 Outline of Functions

(1/2)

Item		μPD79F9211[Note]
Internal memory	Flash memory (self-programming supported)	16K
	RAM	1K
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} = 2.7 to 5.5 V
	Internal high-speed oscillation clock	Internal oscillation 1 MHz, 8 MHz, 20 MHz (TYP.): V_{DD} = 2.7 to 5.5 V
		Double-speed mode internal oscillation 40 MHz (TYP.): V_{DD} = 2.7 to 5.5 V
Subsystem clock (Oscillation frequency)		XT1 (crystal) oscillation 32.768 kHz (TYP.): V_{DD} = 2.7 to 5.5 V
Internal low-speed oscillation clock (For WDT)		Internal oscillation 30 kHz (TYP.): V_{DD} = 2.7 to 5.5 V
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)
Minimum instruction execution time		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)
		0.125 μs (Internal high-speed oscillation clock: f_{IH} = 8 MHz (TYP.) operation)
		61 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)
Instruction set		• 8-bit operation, 16-bit operation • Multiply (16 bits × 16 bits) • Division (32 bits ÷ 32 bits) • Bit manipulation (Set, reset, test, and Boolean operation), etc.
I/O port	TOTAL	37
	CMOS I/O	33
	CMOS Input	4
Timer		• 16-bit timer: 12 channels • Watchdog timer: 1 channel • Real-time counter: 1 channel
	Timer outputs	9 (PWM output: 9)
A/D converter		10-bit resolution × 10 channels (AV_{REF} = 2.7 to 5.5 V)

Note Under development

(2/2)

Item		μPD79F9211[Note]
Serial interface		• CSI: 2 channels/UART (LIN-bus supported): 1 channel • CSI: 1 channel/UART: 1 channel/simplified I^{2}C: 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	33
	External	8
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]
Power-on-clear circuit		• Power-on-reset: 1.61 ±0.09 V • Power-down-reset: 1.59 ±0.09 V
Low-voltage detector		1.91 V to 4.22 V (16 stages)
On-chip debug function		Provided
Power supply voltage		V$_{DD}$ = 2.7 to 5.5 V
Operating ambient temperature		T$_A$ = −40 to +85°C
Package		44-pin plastic LQFP (10 × 10) (0.8 mm pitch)

Notes 1. Under development

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.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

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

Table 2-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27, P150, P151, P80 to P83
V_{DD}	• Port pins other than P20 to P27, P150 to P151, P80 to P83 • Pins other than port pins

(1) Port functions (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P10	I/O	Port 1. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI02/TO02
P11				TI03/TO03
P12				TI04/TO04
P13				TI05/TO05
P20 to P27	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0 to ANI7
P30	I/O	Port 3. 3-bit I/O port. Input of P31 and P32 can be set to TTL buffer. Output of P30 to P32 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SO10/TxD1/TO11
P31				SI10/RxD1/SDA10/ INTP1/TI09
P32				$\overline{\text{SCK10}}$/SCL10/ INTP2
P40 Note	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	I/O	Port 5. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI06/TO06
P51			Input port	TI07/TO07
P52			Input port	SLTI/SLTO
P70	I/O	Port 7. 6-bit I/O port. Input of P71, P72, P74, and P75 can be set to TTL buffer. Output of P70, P72, P73, and P75 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SO01/INTP4
P71				SI01/INTP5
P72				$\overline{\text{SCK01}}$/INTP6
P73				SO00/TxD0/TO10
P74				SI00/RxD0/TI10
P75				$\overline{\text{SCK00}}$/TI11
P80	I/O	Port 8. 4-bit I/O port. Inputs/output can be specified in 1-bit units. Inputs of P80 to P83 can be set as comparator inputs or operational amplifier inputs.	Input port	CMP0P/TMOFF0/ INTP3/OAI
P81				CMP0M
P82				CMP1P/TMOFF1/ INTP7
P83				CMP1M

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 **2.2.4 P40, P41 (port 4)**).

(1) Port functions (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P120	I/O	Port 12. 1-bit I/O port and 4-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
P122				X2/EXCLK
P123				XT1
P124				XT2
P150, P151	I/O	Port 15. 2-bit I/O port. Inputs/output can be specified in 1-bit units.	Digital input port	ANI8, ANI9

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

Function Name	I/O	Function	After Reset	Alternate Function
ANI0 to ANI7	Input	A/D converter analog input	Digital input port	P20 to P27
ANI8, ANI9				P150, P151
CMP0M	Input	Input voltage on the side of comparator 0 (–)	Analog input	P81
CMP0P	Input	Input voltage on the side of comparator 0 (+)		P80/TMOFF0/INTP3/OAI
CMP1M	Input	Input voltage on the side of comparator 1 (–)		P83
CMP1P	Input	Input voltage on the side of comparator 1 (+)		P82/TMOFF1/INTP7
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
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				P31/SI10/RxD1/SDA10/TI09
INTP2				P32/$\overline{\text{SCK10}}$/SCL10
INTP3				P80/CMP0P/TMOFF0/OAI
INTP4				P70/SO01
INTP5				P71/SI01
INTP6				P72/$\overline{\text{SCK01}}$
INTP7				P82/CMP1P/TMOFF1
OAI	Input	Operational amplifier input	Input port	P80/CMP0P/TMOFF0/INTP3
REGC	–	Connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF: target).	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RxD0	Input	Serial data input to UART0	Input port	P74/SI00/TI10
RxD1		Serial data input to UART1		P31/SI10/SDA10/INTP1/TI09
$\overline{\text{SCK00}}$	I/O	Clock input/output for CSI00, CSI01, and CSI10	Input port	P75/TI11
$\overline{\text{SCK01}}$				P72/INTP6
$\overline{\text{SCK10}}$				P32/SCL10/INTP2
SCL10	I/O	Clock input/output for simplified I^2C	Input port	P32/$\overline{\text{SCK10}}$/INTP2
SDA10	I/O	Serial data I/O for simplified I^2C	Input port	P31/SI10/RxD1/INTP1/TI09
SI00	Input	Serial data input to CSI00, CSI01, and CSI10	Input port	P74/RxD0/TI10
SI01				P71/INTP5
SI10				P31/RxD1/SDA10/INTP1/TI09
SLTI	Input	16-bit timer 00, 01, 08, 09, 10, 11 input	Input port	P52/SLTO
SLTO	Output	16-bit timer 00, 01, 08, 09, 10, 11 output	Input port	P52/SLTI

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

Function Name	I/O	Function	After Reset	Alternate Function
SO00	Output	Serial data output from CSI00, CSI01, and CSI10	Input port	P73/TxD0/TO10
SO01				P70/INTP4
SO10				P30/TxD1/TO11
TI02	Input	External count clock input to 16-bit timer 02	Input port	P10/TO02
TI03		External count clock input to 16-bit timer 03		P11/TO03
TI04		External count clock input to 16-bit timer 04		P12/TO04
TI05		External count clock input to 16-bit timer 05		P13/TO05
TI06		External count clock input to 16-bit timer 06		P50/TO06
TI07		External count clock input to 16-bit timer 07		P51/TO07
TI09		External count clock input to 16-bit timer 09		P31/SI10/RxD1/ SDA10/INTP1
TI10		External count clock input to 16-bit timer 10		P74/SI00/RxD0
TI11		External count clock input to 16-bit timer 11		P75/$\overline{\text{SCK00}}$
TMOFF0	Input	Timer pin Hi-Z control input	Input port	P80/CMP0P/INTP3/OAI
TMOFF1				P82/CMP1P/INTP7
TO02	Output	16-bit timer 02 output	Input port	P10/TI02
TO03		16-bit timer 03 output		P11/TI03
TO04		16-bit timer 04 output		P12/TI04
TO05		16-bit timer 05 output		P13/TI05
TO06		16-bit timer 06 output		P50/TI06
TO07		16-bit timer 07 output		P51/TI07
TO10		16-bit timer 10 output		P73/SO00/TxD0
TO11		16-bit timer 11 output		P30/SO10/TxD1
TxD0	Output	Serial data output from UART0	Input port	P73/SO00/TO10
TxD1		Serial data output from UART1		P30/SO10/TO11
X1	–	Resonator connection for main system clock	Input port	P121
X2	–		Input port	P122/EXCLK
EXCLK	Input	External clock input for main system clock	Input port	P122/X2
XT1	–	Resonator connection for subsystem clock	Input port	P123
XT2	–		Input port	P124
V_{DD}	–	Positive power supply (Port pins other than P20 to P27, P80 to P83, P150, P151, and other than ports)	–	–
AV_{REF}	–	• A/D converter and comparator reference voltage input • Positive power supply for P20 to P27, P80 to P83, P150, P151, A/D converter, operational amplifier, and comparator	–	–
V_{SS}	–	Ground potential (Port pins other than P20 to P27, P80 to P83, P150, P151, and other than ports)	–	–
AV_{SS}	–	Ground potential for A/D converter, operational amplifier, comparator, P20 to P27, P80 to P83, P150, and P151	–	–
FLMD0	–	Flash memory programming mode setting	–	–
TOOL0	I/O	Data I/O for flash memory programmer/debugger	Input port	P40
TOOL1	Output	Clock output for debugger	Input port	P41

2.2 Description of Pin Functions

2.2.1 P10 to P13 (port 1)

P10 to P13 function as an 4-bit I/O port. These pins also function as timer I/O.

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

(1) Port mode

P10 to P13 function as an 4-bit I/O port. P10 to P13 can be set to input or output port in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

(2) Control mode

P10 to P13 function as timer I/O.

(a) TI02 to TI05

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 02 to 05.

(b) TO02 to TO05

These are the timer output pins of 16-bit timers 02 to 05.

2.2.2 P20 to P27 (port 2)

P20 to P27 function as an 8-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

P20 to P27 function as an 8-bit I/O port. P20 to P27 can be set to input or output port in 1-bit units using port mode register 2 (PM2).

(2) Control mode

P20 to P27 function as A/D converter analog input pins (ANI0 to ANI7). When using these pins as analog input pins, see **11.6 (5) ANI0/P20 to ANI7/P27, ANI8/P150 to ANI9/P151**.

Caution ANI0/P20 to ANI7/P27 are set in the digital input (general-purpose port) mode after release of reset.

2.2.3 P30 to P32 (port 3)

P30 to P32 function as a 3-bit I/O port. These pins also function as serial interface data I/O, clock I/O, external interrupt request input, and timer I/O.

Input to the P30 and P31 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units, using port input mode register 3 (PIM3).

Output from the P30 to P32 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 3 (POM3).

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

(1) Port mode

P30 to P32 function as a 3-bit I/O port. P30 to P32 can be set to input or output port in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

(2) Control mode

P30 to P32 function as serial interface data I/O, clock I/O, external interrupt request input, and timer I/O.

(a) SI10

This is a serial data input pin of serial interface CSI10.

(b) SO10

This is a serial data output pin of serial interface CSI10.

(c) $\overline{\text{SCK10}}$

This is a serial clock I/O pin of serial interface CSI10.

(d) TxD1

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

(e) RxD1

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

(f) SDA10

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

(g) SCL10

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

(h) 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.

(i) TI09

This is a pin for inputting an external count clock/capture trigger to 16-bit timer 09.

(j) TO11

This is a timer output pin of 16-bit timer 11.

Caution To use P30/SO10/TxD1/TO11 and P32/$\overline{\text{SCK10}}$/SCL10/INTP2 as general-purpose ports, set serial communication operation setting register 02 (SCR02) to the default status (0087H). In addition, clear port output mode register 3 (POM3) to 00H.

2.2.4 P40 and P41 (port 4)

P40 and P41 function as an 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 an 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 V_{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 V_{DD} via an external resistor.

2.2.5 P50 to P52 (port 5)

P50 to P52 function as an 3-bit I/O port. These pins also function as timer I/O.

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

(1) Port mode

P50 to P52 function as an 3-bit I/O port. P50 to P52 can be set to input or output port in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

(2) Control mode

P50 to P52 function as timer I/O.

(a) TI06, TI07

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 06 and 07.

(b) TO06, TO07

These are the timer output pins of 16-bit timers 06 and 07.

(c) SLTI

This is used as a pin for inputting an external count clock or a capture trigger to 16-bit timers 00, 01, 08, 09, 10, and 11, by setting the input switching control register (ISC).

(d) SLTO

This is used as a timer output pin of 16-bit timers 00, 01, 08, 09, 10, and 11, by setting the input switching control register (ISC).

2.2.6 P70 to P75 (port 7)

P70 to P75 function as a 6-bit I/O port. These pins also function as serial interface data I/O, clock I/O, external interrupt request input, and timer I/O.

Input to the P71, P72, P74, and P75 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units, using port input mode register 7 (PIM7).

Output from the P70, P72, P73, and P75 pins can be specified as normal CMOS output or N-ch open-drain output (V_{DD} tolerance) in 1-bit units, using port output mode register 7 (POM7).

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

(1) Port mode

P70 to P75 function as a 6-bit I/O port. P70 to P75 can be set to input or output port in 1-bit units using port mode register 7 (PM7). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

(2) Control mode

P70 to P75 function as serial interface data I/O, clock I/O, external interrupt request input, and timer I/O.

(a) TI10, TI11

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 10 and 11.

(b) TO10

This is a timer output pin of 16-bit timer 10.

(c) SI00, SI01

These are the serial data input pin of serial interface CSI00 and CSI01.

(d) SO00, SO01

These are the serial data output pin of serial interface CSI00 and CSI01.

(e) $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$

These are the serial clock I/O pins of serial interface CSI00 and CSI01.

(f) TxD0

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

(g) RxD0

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

(h) INTP4 to INTP6

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

Caution To use P70/SO01/INTP4, P72/$\overline{\text{SCK01}}$/INTP6, P73/SO00/TxD0/TO10, and P75/$\overline{\text{SCK00}}$/TI11 as general-purpose ports, set serial communication operation setting registers 00 and 01 (SCR00 and SCR01) to the default status (0087H). In addition, clear port output mode register 7 (POM7) to 00H.

2.2.7 P80 to P83 (port 8)

P80 to P83 function as an 4-bit I/O port. These pins also function as input voltages on the side of comparators 0 and 1 (+), input voltages on the side of comparators 0 and 1 (–), timer pin Hi-Z control inputs, external interrupt request inputs, and operational amplifier inputs.

Inputs to the P80 to P83 pins must be enabled or disabled in 1-bit units using port input mode register 8 (PIM8).

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

(1) Port mode

P80 to P83 function as an 4-bit I/O port. P80 to P83 can be set to input port or output port in 1-bit units using port mode register 8 (PM8).

(2) Control mode

P80 to P83 function as input voltages on the side of comparators 0 and 1 (+), input voltages on the side of comparators 0 and 1 (–), timer pin Hi-Z control inputs, external interrupt request inputs, and operational amplifier inputs.

(a) CMP0P, CMP1P

These are the input voltage pins on the side of comparators 0 and 1 (+).

(b) CMP0M, CMP1M

These are the input voltage pins on the sides of comparators 0 and 1 (–).

(c) TMOFF0, TMOFF1

These are the timer pin Hi-Z control input pins.

(d) INTP3, INTP7

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

(e) OAI

This is an operational amplifier input pin.

2.2.8 P120 to P124 (port 12)

P120 functions as a 1-bit I/O port. P121 to P124 function as a 4-bit input port. These pins also function as external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, and external clock input 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 to P124 function as a 4-bit input port.

(2) Control mode

P120 to P124 function as external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, and external clock input 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, X2

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

(d) XT1, XT2

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

(e) EXCLK

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

2.2.9 P150 and P151 (port 15)

P150 and P151 function as an 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

P150 and P151 function as an 2-bit I/O port. P150 and P151 can be set to input or output port in 1-bit units using port mode register 15 (PM15).

(2) Control mode

P150 and P151 function as A/D converter analog input pins (ANI8 and ANI9). When using these pins as analog input pins, see **11.6 (5) ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI9/P151**.

Caution ANI8/P150 and ANI9/P151 are set in the digital input (general-purpose port) mode after release of reset.

2.2.10 AV$_{REF}$

This is the A/D converter and comparator reference voltage input pin and the positive power supply pin of P20 to P27, P150 to P151, P80 to P83, A/D converter, operational amplifier, and comparator.

When all pins of port 2, port 15**Note**, and port 8 are used as the analog port pins, make the potential of AV$_{REF}$ be such that 2.7 V ≤ AV$_{REF}$ ≤ V$_{DD}$. When one or more of the pins of port 2, port 15**Note**, and port 8 are used as the digital port pins or when the A/D converter, operational amplifier, and comparator are not used, make AV$_{REF}$ the same potential as V$_{DD}$.

2.2.11 AV$_{SS}$

This is the ground potential pin of A/D converter, operational amplifier, comparator, P20 to P27, P150 to P151, and P80 to P83. Even when the A/D converter, operational amplifier, and comparator are not used, always use this pin with the same potential as V$_{SS}$.

2.2.12 $\overline{\text{RESET}}$

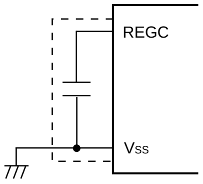
This is the active-low system reset input pin.

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

2.2.13 REGC

This is the pin for connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF: target). However, when using the STOP mode that has been entered since operation of the internal high-speed oscillation clock and external main system clock, 0.47 μF is recommended.

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



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

2.2.14 V_{DD}

V_{DD} is the positive power supply pin for port pins other than P20 to P27, P150 to P151, and P80 to P83, and other than ports.

2.2.15 V_{SS}

V_{SS} is the ground potential pin for port pins other than P20 to P27, P150 to P151, and P80 to P83, and other than ports.

2.2.16 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} 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.

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

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

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P10/TI02/TO02	8-R	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P11/TI03/TO03			
P12/TI04/TO04			
P13/TI05/TO05			
P20/ANI0 to P27/ANI7[Note]	11-G		Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.
P30/SO10/TxD1/TO11	5-AG		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <When N-ch open-drain> Output • Set the port output latch to 0: Leave open. • Set the port output latch to 1: Independently connect to V_{DD} or V_{SS} via a resistor.
P31/SI10/RxD1/SDA10/INTP1/TI09	5-AN		
P32/$\overline{\text{SCK10}}$/SCL10/INTP2			
P40/TOOL0	8-R		<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} or V_{SS} via a resistor. Output: Leave open.
P41/TOOL1			Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P50/TI06/TO06			
P51/TI07/TO07			
P52/SLTI/SLTO			

Note P20/ANI0 to P27/ANI7 are set in the digital input port mode after release of reset.

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P70/SO01/INTP4	8-R	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <When N-ch open-drain> Output • Set the port output latch to 0: Leave open. • Set the port output latch to 1: Independently connect to V_{DD} or V_{SS} via a resistor.
P71/SI01/INTP5	5-AN		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P72/$\overline{\text{SCK01}}$/INTP6			Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <When N-ch open-drain> Output • Set the port output latch to 0: Leave open. • Set the port output latch to 1: Independently connect to V_{DD} or V_{SS} via a resistor.
P73/SO00/TxD0/TO10	8-R		
P74/SI00/RxD0/TI10	5-AN		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P75/$\overline{\text{SCK00}}$/TI11			Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <When N-ch open-drain> Output • Set the port output latch to 0: Leave open. • Set the port output latch to 1: Independently connect to V_{DD} or V_{SS} via a resistor.
P80/CMP0P/TMOFF0/ INTP3/OAI	11-J		Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.
P81/CMP0M	11-H		
P82/CMP1P/TOMOFF1/ INTP7	11-I		
P83/CMP1M	11-H		
P120/INTP0/EXLVI	8-R		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121/X1[Note 1]	37-C	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK[Note 1]			
P123/XT1[Note 1]	37-B		
P124/XT2[Note 1]			
P150/ANI8, P151/ANI9[Note 2]	11-G	I/O	Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.

Notes 1. Use recommended connection above in input port mode (see **Figure 5-2 Format of Clock Operation Mode Control Register (CMC)**) when these pins are not used.

2. P150/ANI8 and P151/ANI9 are set in the digital input port mode after release of reset.

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
AV_{REF}	–	–	<When one or more of P20 to P27, P150, P151, or P80 to P83 are set as a digital port> Make this pin the same potential as V_{DD}. <When all of P20 to P27, P150, P151, and P80 to P83 are set as analog ports> Make this pin to have a potential where 2.7 V $\leq AV_{REF} \leq V_{DD}$.
AV_{SS}	–	–	Make this pin the same potential as the V_{SS}.
FLMD0	2-W	–	Leave open or connect to V_{SS} via a resistor of 100 kΩ or more.
$\overline{\text{RESET}}$	2	Input	Connect directly to V_{DD} or via a resistor.
REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF: target).

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

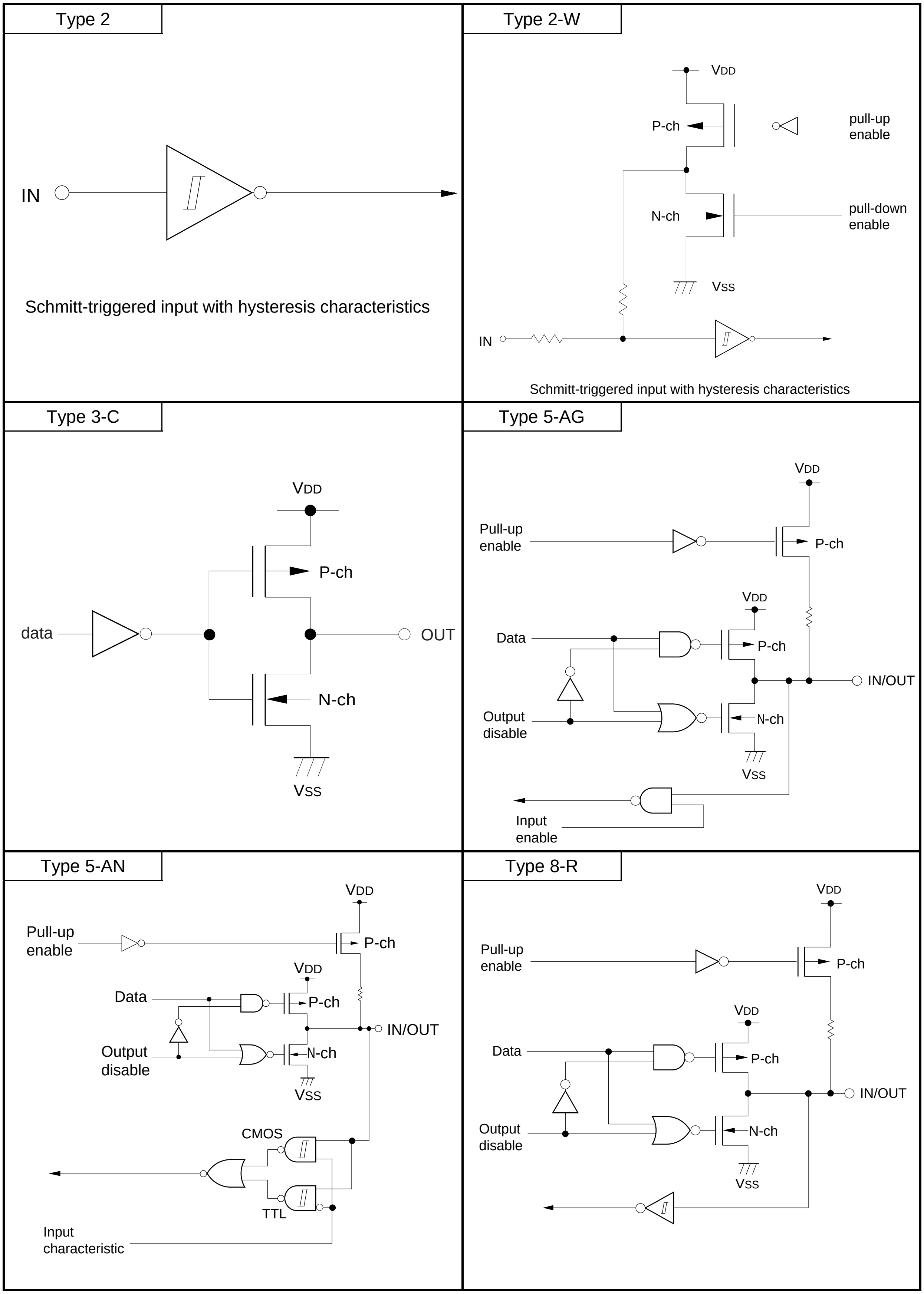


Figure 2-1. Pin I/O Circuit List (2/3)

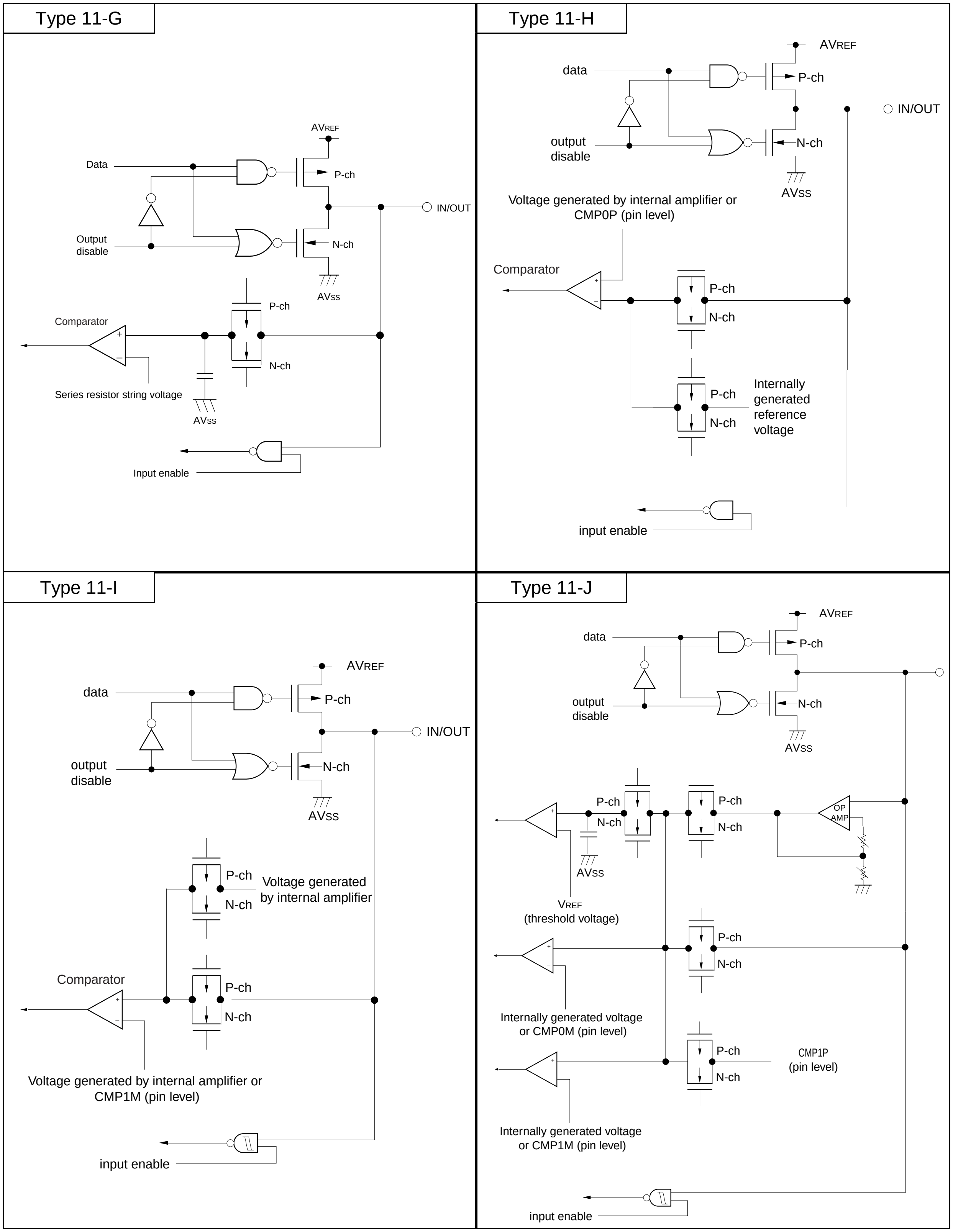
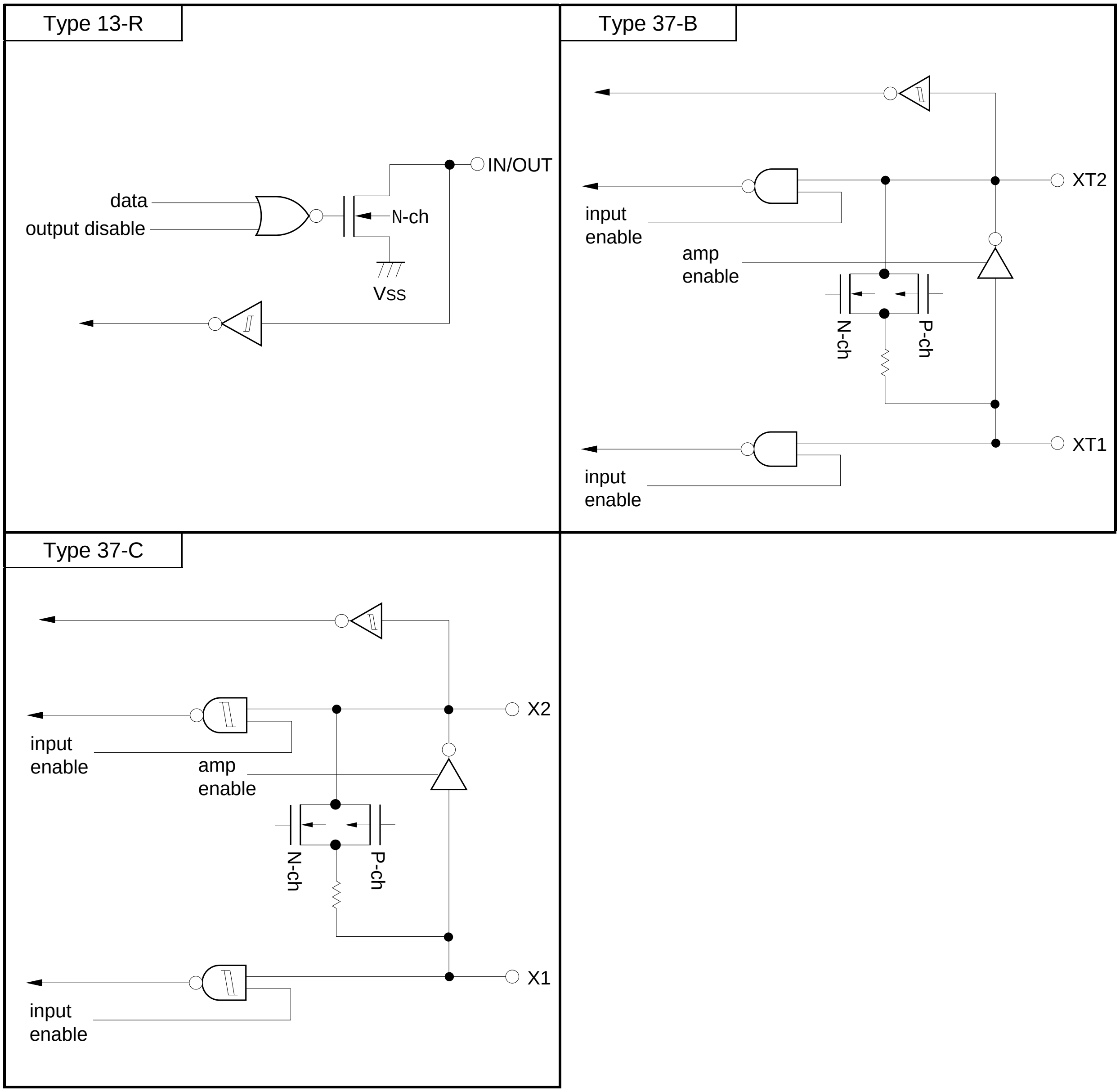


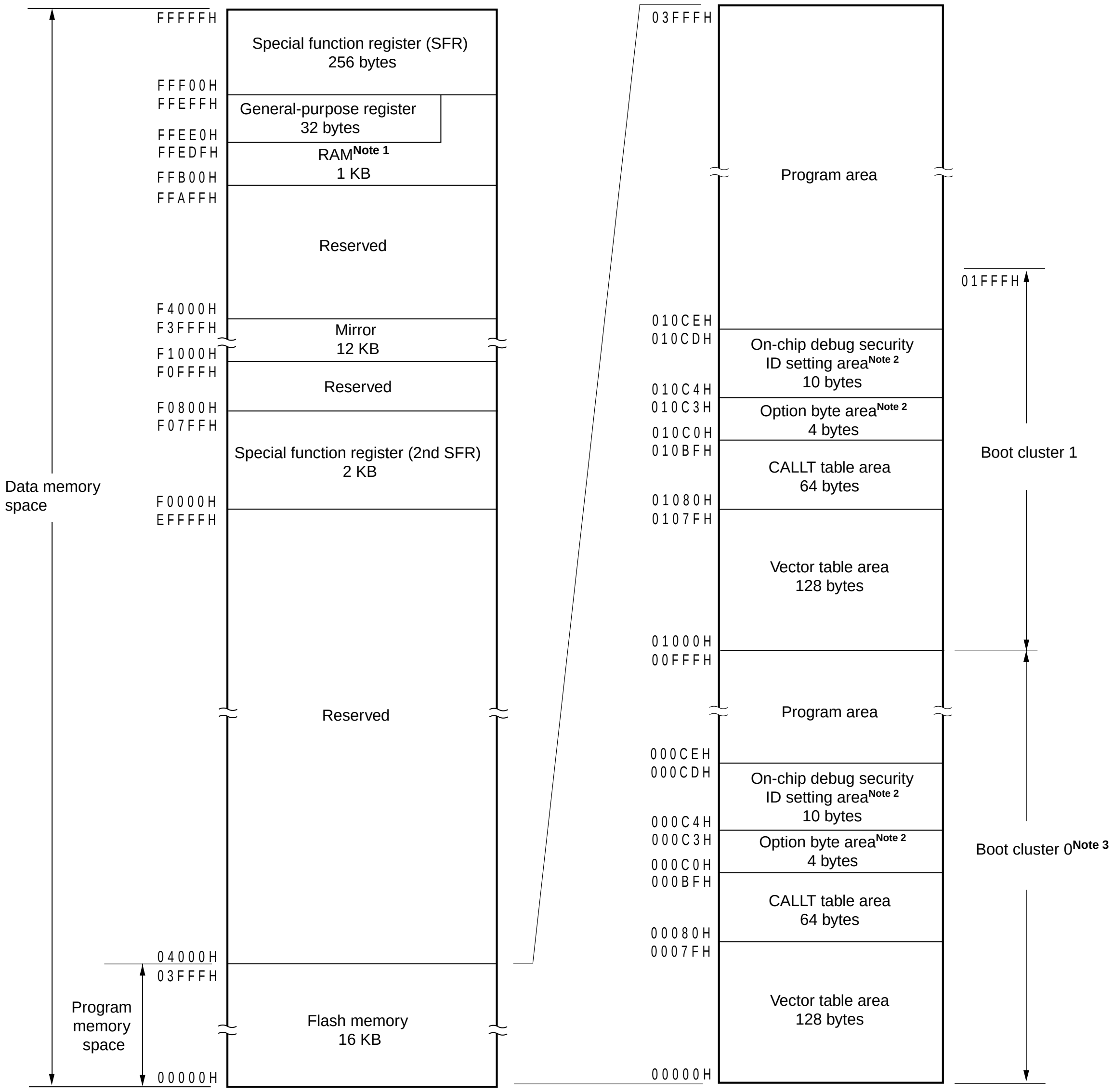
Figure 2-1. Pin I/O Circuit List (3/3)



3.1 Memory Space

Products in the μPD79F9211 can access a 1 MB memory space. Figures 3-1 shows the memory maps.

Figure 3-1. Memory Map (μPD79F9211)



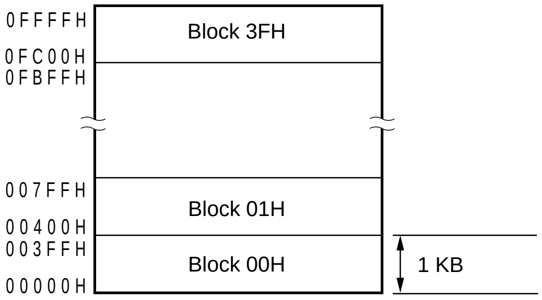
Notes 1. Use of the area FFE20H to FFEDFH is prohibited when using the self-programming function. Since this area is used for self-programming library.

2. 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.

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



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

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

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

Remark μPD79F9211: Block numbers 00H to 0FH

3.1.1 Internal program memory space

The internal program memory space stores the program and table data. Normally, it is addressed with the program counter (PC).

μPD79F9211 products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD79F9211	Flash memory	16384 × 8 bits (00000H to 03FFFH)

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.

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 3-3. 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	00036H	INTRTC
0000CH	INTP2	00038H	INTRTCI
0000EH	INTP3/INTTMOFF0	0003CH	INTTMM0
00010H	INTP4	0003EH	INTTMV0
00012H	INTP5	00040H	INTMD
00014H	INTTMAD	00042H	INTTM04
00016H	INTCMP0	00044H	INTTM05
00018H	INTCMP1	00046H	INTTM06
0001AH	INTDMA0	00048H	INTTM07
0001CH	INTDMA1	0004AH	INTP6
0001EH	INTST0/INTCSI00	0004CH	INTP7/INTTMOFF1
00020H	INTSR0/INTCSI01	0004EH	INTTMM1
00022H	INTSRE0	00050H	INTTMV1
00024H	INTST1/INTCSI10/INTIIC10	00052H	INTTM08
00026H	INTSR1	00054H	INTTM09
00028H	INTSRE1	00056H	INTTM10
		00058H	INTTM11

(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**.

3.1.2 Mirror area

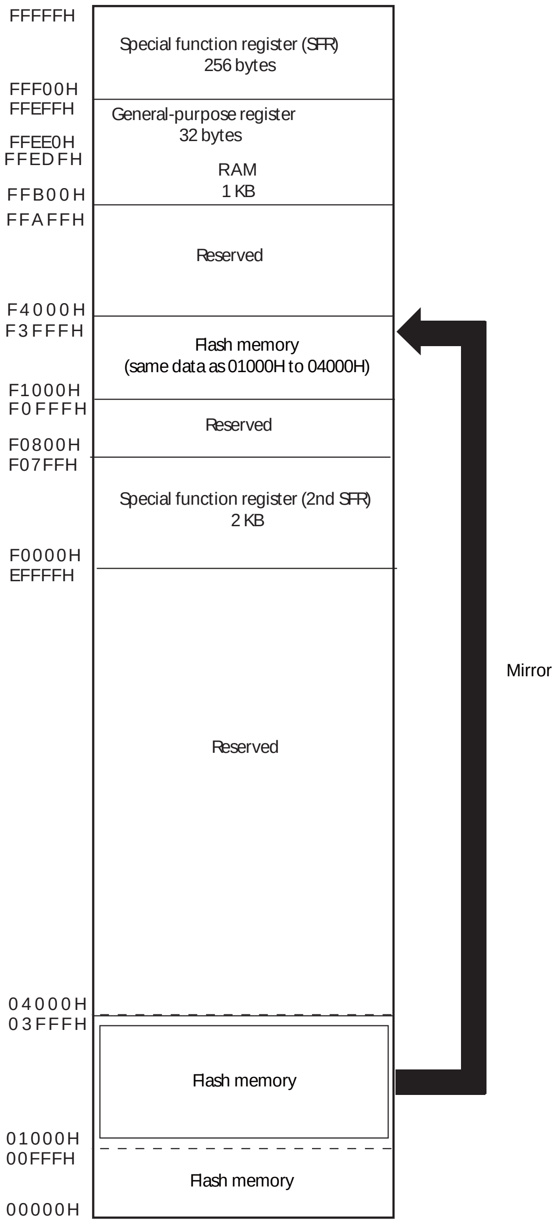
The μPD79F9211 mirrors the data flash area of 00000H to 0FFFFH, to F0000H to FFFFFH (set by the processor mode control register (PMC)).

By reading data from F0000H to FFFFFH, 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.

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

The following show examples.

Example μPD79F9211 (Flash memory: 16 KB, RAM: 1 KB)



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 3-2. Format of **Configuration of Processor Mode Control Register (PMC)**

Address: FFFFEH After reset: 00H R/W

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

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

Cautions 1. Be sure to set bit 0 (MAA) of this register to 0.

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

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

3.1.3 Internal data memory space

μPD79F9211 products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
μPD79F9211	1024 × 8 bits (FEB00H to FFEFFH)

The 32-byte area FFEE0H to FFEFFH is assigned to four general-purpose register banks consisting of eight 8-bit registers per bank.

This area can be used as a program area where instructions are written and executed. However, executing instructions is disabled in the general-purpose register.

The internal high-speed RAM can also be used as a stack memory.

Caution While using the self-programming function, the areas FFE20H to FFEDFH cannot be used as stack memory.

3.1.4 Special function register (SFR) area

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

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

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

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

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

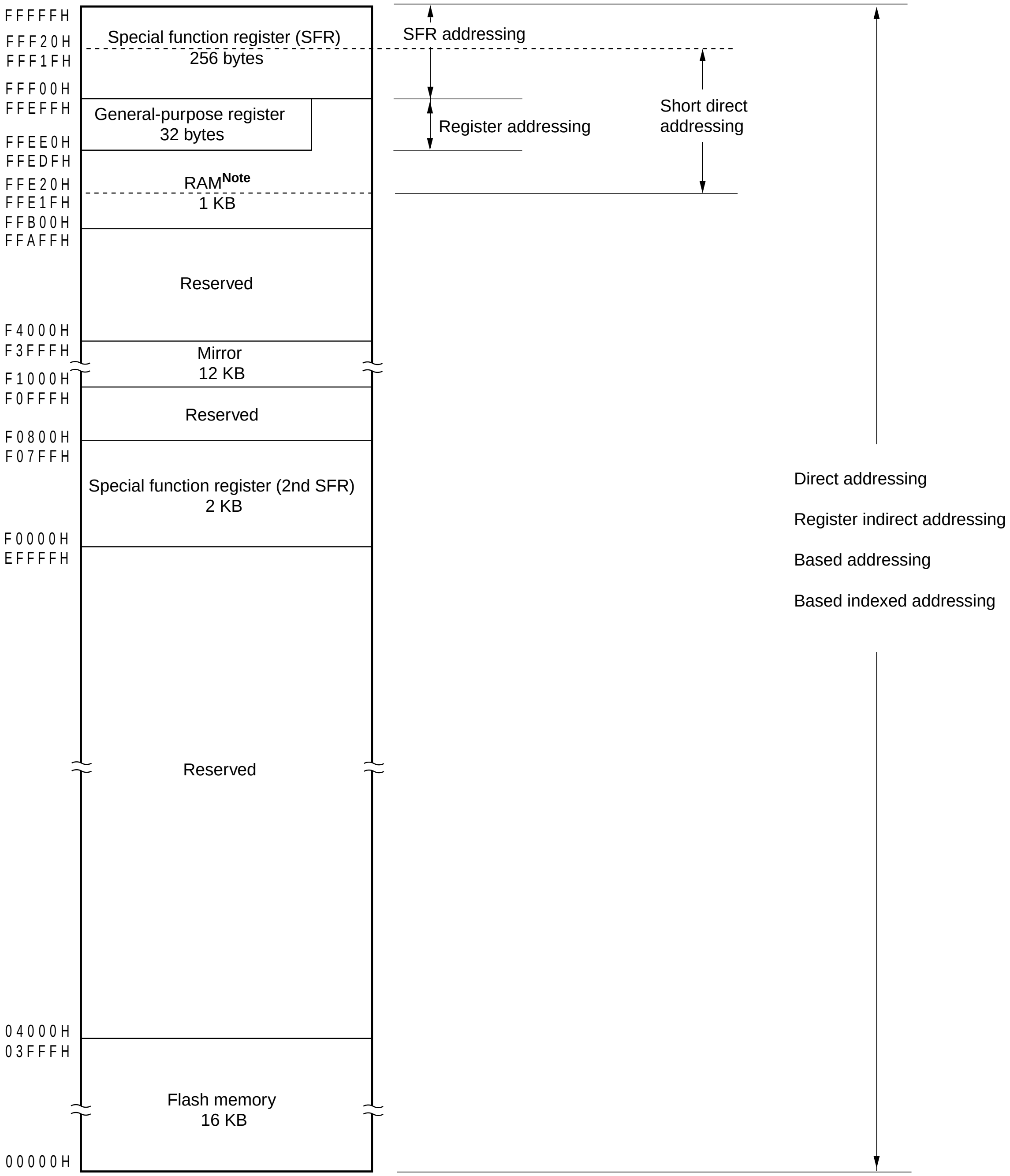
Caution Do not access addresses to which extended SFRs are not assigned.

3.1.6 Data memory addressing

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

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD79F9211, 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 3-3 shows correspondence between data memory and addressing.

Figure 3-3. Correspondence Between Data Memory and Addressing (μPD79F9211)



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.

3.2 Processor Registers

The μPD79F9211 products incorporate the following processor registers.

3.2.1 Control registers

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

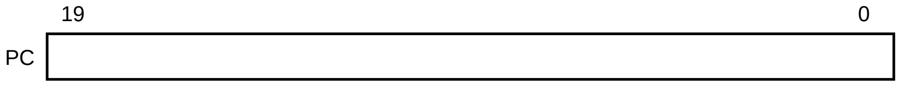
(1) Program counter (PC)

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

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

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

Figure 3-4. Format of Program Counter



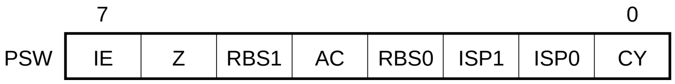
(2) Program status word (PSW)

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

Program status word contents are stored in the stack area upon interrupt request generation 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 3-5. Format of Program Status Word



(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)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

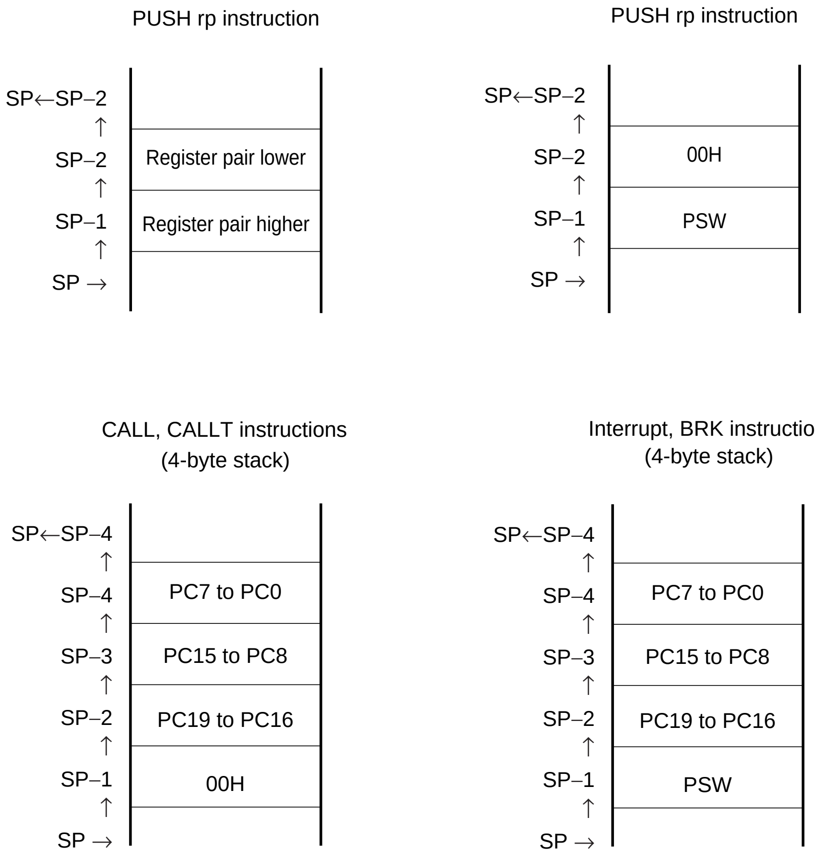
Figure 3-6. Format of Stack Pointer

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

The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.
Each stack operation saves data as shown in Figure 3-7.

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

Figure 3-7. Data to Be Saved to Stack Memory



3.2.2 General-purpose registers

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

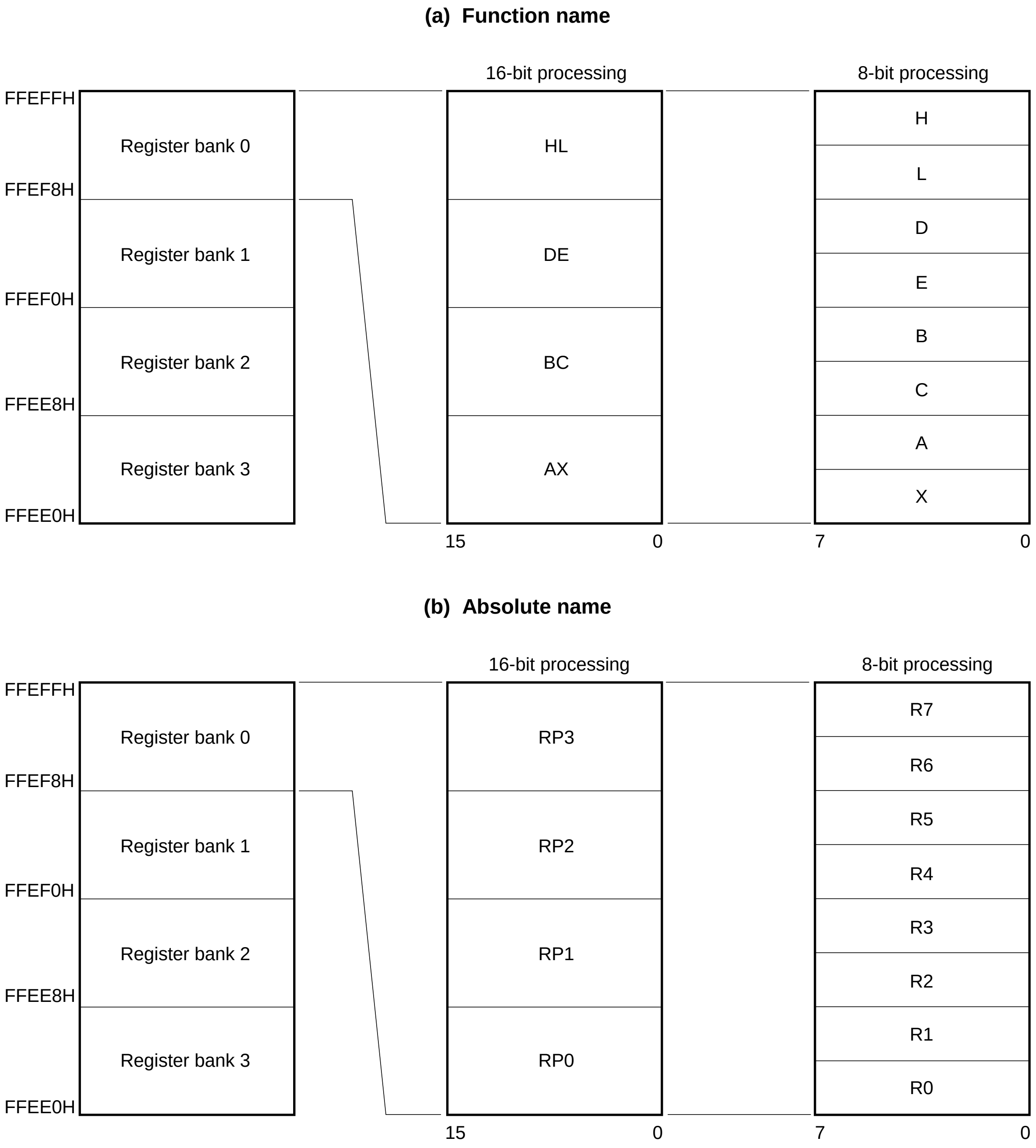
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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

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

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

Figure 3-8. Configuration of General-Purpose Registers



(a) Function name

(b) Absolute name

3.2.3 ES and CS registers

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

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

Figure 3-9. Configuration of ES and CS Registers

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

	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CP2	CP1	CP0

3.2.4 Special function registers (SFRs)

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

SFRs are allocated to the FFF00H to FFFFFH area.

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

Each manipulation bit unit can be specified as follows.

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

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

- Symbol
  Symbol indicating the address of a special function register. It is a reserved word in the 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 extended SFRs are not assigned.

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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
FFF07H	Port register 7	P7		R/W	√	√	–	00H
FFF08H	Port register 8	P8		R/W	√	√	–	00H
FFF0CH	Port register 12	P12		R/W	√	√	–	Undefined
FFF0FH	Port register 15	P15		R/W	√	√	–	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	–	√	√	0000H
FFF11H		–			–	–		
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–		
FFF18H	Timer data register 00	TDR00		R/W	–	–	√	0000H
FFF19H								
FFF1AH	Timer data register 01	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
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
FFF27H	Port mode register 7	PM7		R/W	√	√	–	FFH
FFF28H	Port mode register 8	PM8		R/W	√	√	–	FFH
FFF2CH	Port mode register 12	PM12		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
FFF3EH	Timer input select register 0	TIS0		R/W	√	√	–	00H
FFF42H	A/D converter mode register 1	ADM1		R/W	√	√	–	00H

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	Serial data register 03	RXD1	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
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 08	TDR08		R/W	–	–	√	0000H
FFF71H								
FFF72H	Timer data register 09	TDR09		R/W	–	–	√	0000H
FFF73H								
FFF74H	Timer data register 10	TDR10		R/W	–	–	√	0000H
FFF75H								
FFF76H	Timer data register 11	TDR11		R/W	–	–	√	0000H
FFF77H								
FFF90H	Sub-count register	RSUBC		R	–	–	√	0000H
FFF91H								
FFF92H	Second count register	SEC		R/W	–	√	–	00H
FFF93H	Minute count register	MIN		R/W	–	√	–	00H
FFF94H	Hour count register	HOUR		R/W	–	√	–	12H [Note]
FFF95H	Week count register	WEEK		R/W	–	√	–	00H
FFF96H	Day count register	DAY		R/W	–	√	–	01H
FFF97H	Month count register	MONTH		R/W	–	√	–	01H
FFF98H	Year count register	YEAR		R/W	–	√	–	00H
FFF99H	Watch error correction register	SUBCUD		R/W	–	√	–	00H
FFF9AH	Alarm minute register	ALARMWM		R/W	–	√	–	00H
FFF9BH	Alarm hour register	ALARMWH		R/W	–	√	–	12H
FFF9CH	Alarm week register	ALARMWW		R/W	–	√	–	00H

Note. The value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF9DH	Real-time counter control register 0	RTCC0		R/W	√	√	–	00H
FFF9EH	Real-time counter control register 1	RTCC1		R/W	√	√	–	00H
FFF9FH	Real-time counter control register 2	RTCC2		R/W	√	√	–	00H
FFFA0H	Clock operation mode control register	CMC		R/W	–	√	–	00H
FFFA1H	Clock operation status control register	CSC		R/W	√	√	–	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	√	√	–	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	–	√	–	07H
FFFA4H	Clock control register	CKC		R/W	√	√	–	09H
FFFA5H	Clock output select register 0	CKS0		R/W	√	√	–	00H
FFFA8H	Reset control flag register	RESF		R	–	√	–	00H[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

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 3-5. SFR List (4/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	Interrupt request flag register 1H	IF1H		R/W	√	√		00H
FFFE4H	Interrupt mask flag register 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	Interrupt mask flag register 0H	MK0H		R/W	√	√		FFH
FFFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	Interrupt mask flag register 1H	MK1H		R/W	√	√		FFH
FFFE8H	Priority specification flag register 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	Priority specification flag register 00H	PR00H		R/W	√	√		FFH
FFFEAH	Priority specification flag register 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
FFFF0H	Multiplication/division data register A (L)	MDAL/MULA		R/W	–	–	√	0000H
FFFF1H								
FFFF2H	Multiplication/division data register A (H)	MDAH/MULB		R/W	–	–	√	0000H
FFFF3H								
FFFF4H	Multiplication/division data register B (H)	MDBH/MULOH		R/W	–	–	√	0000H
FFFF5H								
FFFF6H	Multiplication/division data register B (L)	MDBL/MULOL		R/W	–	–	√	0000H
FFFF7H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H

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

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

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

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

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

Each manipulation bit unit can be specified as follows.

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

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

- Symbol
  Symbol indicating the address of an extended SFR. It is a reserved word in the 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 extended SFRs are not assigned.

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

Table 3-6. Extended SFR (2nd SFR) List (1/5)

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
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
F0037H	Pull-up resistor option register 7	PU7		R/W	√	√	–	00H
F003CH	Pull-up resistor option register 12	PU12		R/W	√	√	–	00H
F0043H	Port input mode register 3	PIM3		R/W	√	√	–	00H
F0047H	Port input mode register 7	PIM7		R/W	√	√	–	00H
F0048H	Port input mode register 8	PIM8		R/W	√	√	–	00H
F0053H	Port output mode register 3	POM3		R/W	√	√	–	00H
F0057H	Port output mode register 7	POM7		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
F00F1H	Peripheral enable register 1	PER1		R/W	√	√	–	00H
F00F2H	Peripheral enable register 2	PER2		R/W	√	√	–	00H
F00F3H	Operation speed mode control register	OSMC		R/W	–	√	–	00H
F00F4H	Regulator mode control register	RMC		R/W	–	√	–	00H
F00F6H	Double speed clock control register	DSCCTL		R/W	√	√	–	00H
F00FEH	BCD adjust result register	BCDADJ		R	–	√	–	00H
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–	–		
F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–	–		
F0104H	Serial status register 02	SSR02L	SSR02	R	–	√	√	0000H
F0105H		–			–	–		
F0106H	Serial status register 03	SSR03L	SSR03	R	–	√	√	0000H
F0107H		–			–	–		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0109H		–			–	–		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H
F010BH		–			–	–		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	–	√	√	0000H
F010DH		–			–	–		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	–	√	√	0000H
F010FH		–			–	–		

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	–	–	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	–	–	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	–	–	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	–	–	√	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		–			–	–		
F0122H	Serial channel start trigger register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		–			–	–		
F0124H	Serial channel stop trigger 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		–			–	–		
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								

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F018EH	Timer counter register 07	TCR07	R	–	–	√	FFFFH
F018FH							
F0190H	Timer mode register 00	TMR00	R/W	–	–	√	0000H
F0191H							
F0192H	Timer mode register 01	TMR01	R/W	–	–	√	0000H
F0193H							
F0194H	Timer mode register 02	TMR02	R/W	–	–	√	0000H
F0195H							
F0196H	Timer mode register 03	TMR03	R/W	–	–	√	0000H
F0197H							
F0198H	Timer mode register 04	TMR04	R/W	–	–	√	0000H
F0199H							
F019AH	Timer mode register 05	TMR05	R/W	–	–	√	0000H
F019BH							
F019CH	Timer mode register 06	TMR06	R/W	–	–	√	0000H
F019DH							
F019EH	Timer mode register 07	TMR07	R/W	–	–	√	0000H
F019FH							
F01A0H	Timer status register 00	TSR00	R	–	–	√	0000H
F01A1H							
F01A2H	Timer status register 01	TSR01	R	–	–	√	0000H
F01A3H							
F01A4H	Timer status register 02	TSR02	R	–	–	√	0000H
F01A5H							
F01A6H	Timer status register 03	TSR03	R	–	–	√	0000H
F01A7H							
F01A8H	Timer status register 04	TSR04	R	–	–	√	0000H
F01A9H							
F01AAH	Timer status register 05	TSR05	R	–	–	√	0000H
F01ABH							
F01ACH	Timer status register 06	TSR06	R	–	–	√	0000H
F01ADH							
F01AEH	Timer status register 07	TSR07	R	–	–	√	0000H
F01AFH							
F01B0H	Timer channel enable status register 0	TE0	R	–	–	√	0000H
F01B1H							
F01B2H	Timer channel start trigger register 0	TS0	R/W	–	–	√	0000H
F01B3H							
F01B4H	Timer channel stop trigger register 0	TT0	R/W	–	–	√	0000H
F01B5H							
F01B6H	Timer clock select register 0	TPS0	R/W	–	–	√	0000H
F01B7H							
F01B8H	Timer channel output register 0	TO0	R/W	–	–	√	0000H
F01B9H							

Table 3-6. Extended SFR (2nd SFR) List (4/5)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F01BAH	Timer channel output enable register 0	TOE0	R/W	–	–	√	0000H
F01BBH							
F01BCH	Timer channel output level register 0	TOL0	R/W	–	–	√	0000H
F01BDH							
F01BEH	Timer channel output mode register 0	TOM0	R/W	–	–	√	0000H
F01BFH							
F01C0H	Timer counter register 08	TCR08	R	–	–	√	FFFFH
F01C1H							
F01C2H	Timer counter register 09	TCR09	R	–	–	√	FFFFH
F01C3H							
F01C4H	Timer counter register 10	TCR10	R	–	–	√	FFFFH
F01C5H							
F01C6H	Timer counter register 11	TCR11	R	–	–	√	FFFFH
F01C7H							
F01C8H	Timer mode register 08	TMR08	R/W	–	–	√	0000H
F01C9H							
F01CAH	Timer mode register 09	TMR09	R/W	–	–	√	0000H
F01CBH							
F01CCH	Timer mode register 10	TMR10	R/W	–	–	√	0000H
F01CDH							
F01CEH	Timer mode register 11	TMR11	R/W	–	–	√	0000H
F01CFH							
F01D0H	Timer status register 08	TSR08	R	–	–	√	0000H
F01D1H							
F01D2H	Timer status register 09	TSR09	R	–	–	√	0000H
F01D3H							
F01D4H	Timer status register 10	TSR10	R	–	–	√	0000H
F01D5H							
F01D6H	Timer status register 11	TSR11	R	–	–	√	0000H
F01D7H							
F01E8H	Timer triangle wave output mode register 0	TOT0	R/W	–	–	√	0000H
F01EAH	Timer real-time output enable register 0	TRE0	R/W	–	–	√	0000H
F01ECH	Timer real-time output register 0	TRO0	R/W	–	–	√	0000H
F01EEH	Timer real-time control register 0	TRC0	R/W	–	–	√	0000H
F01F0H	Timer modulation output enable register 0	TME0	R/W	–	–	√	0000H
F01F2H	Timer dead-time output enable register 0	TDE0	R/W	–	–	√	0000H
F0220H	TAU option mode register	OPMR	R/W	–	–	√	0000H
F0222H	TAU option status register	OPSR	R	–	–	√	0000H
F0224H	TAU option Hi-Z start trigger register	OPHS	R/W	–	–	√	0000H
F0226H	TAU option Hi-Z stop trigger register	OPHT	R/W	–	–	√	0000H

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0234H	Slave address register	SVA	R/W	–	√	–	00H
F0240H	Operational amplifier control register	OAM	R/W	√	√	–	00H
F0241H	Comparator 0 control register	C0CTL	R/W	√	√	–	00H
F0242H	Comparator 0 internal reference voltage setting register	C0RVM	R/W	√	√	–	00H
F0243H	Comparator 1 control register	C1CTL	R/W	√	√	–	00H
F0244H	Comparator 1 internal reference voltage setting register	C1RVM	R/W	√	√	–	00H

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

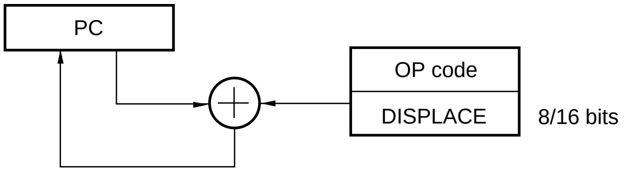
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

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

Figure 3-10. Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

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

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

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

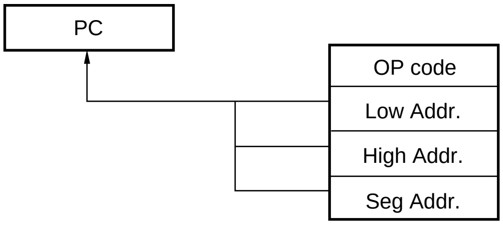
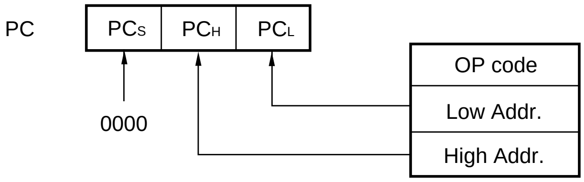


Figure 3-12. Example of CALL !addr16/BR !addr16



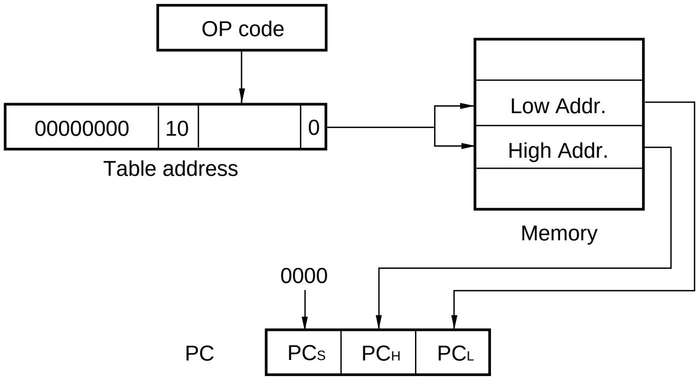
3.3.3 Table indirect addressing

[Function]

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

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

Figure 3-13. Outline of Table Indirect Addressing

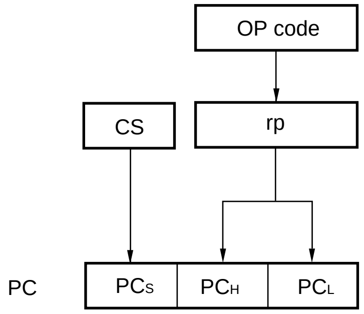


3.3.4 Register direct addressing

[Function]

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

Figure 3-14. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

[Function]

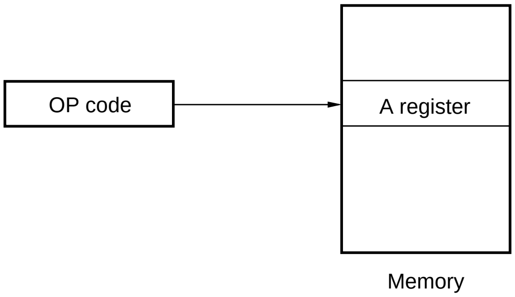
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

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

Implied addressing can be applied only to MULU X.

Figure 3-15. Outline of Implied Addressing



3.4.2 Register addressing

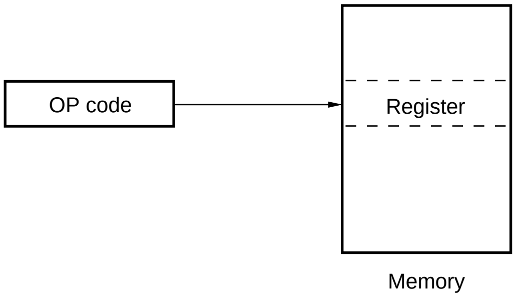
[Function]

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

[Operand format]

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

Figure 3-16. Outline of Register Addressing



3.4.3 Direct addressing

[Function]

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

[Operand format]

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

Figure 3-17. Example of ADDR16

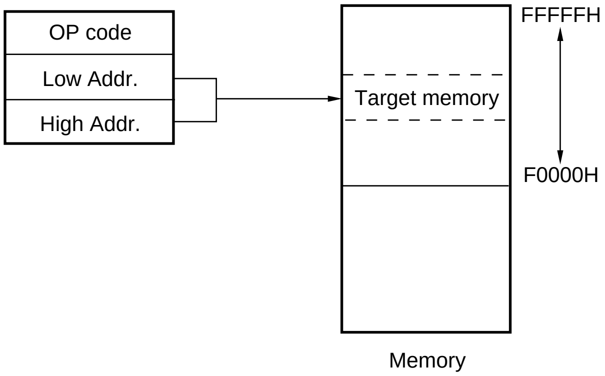
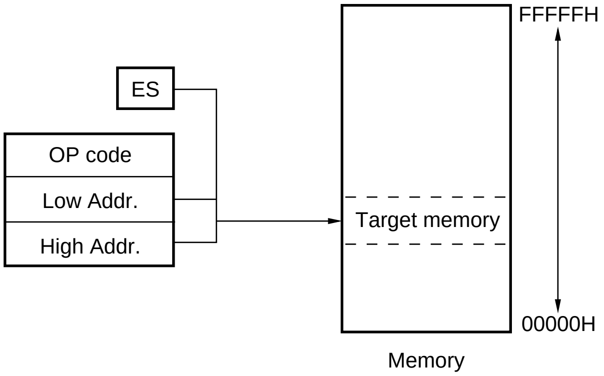


Figure 3-18. Example of ES:ADDR16



3.4.4 Short direct addressing

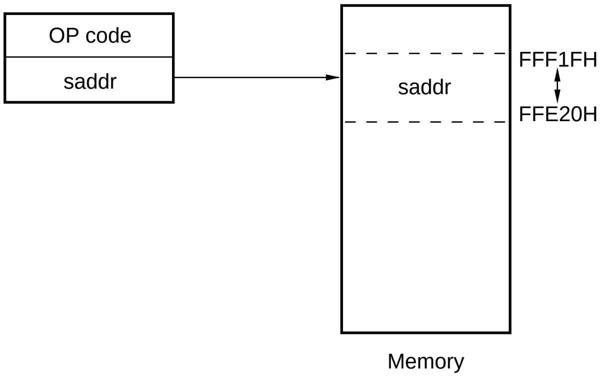
[Function]

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

[Operand format]

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

Figure 3-19. Outline of Short Direct Addressing



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

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

3.4.5 SFR addressing

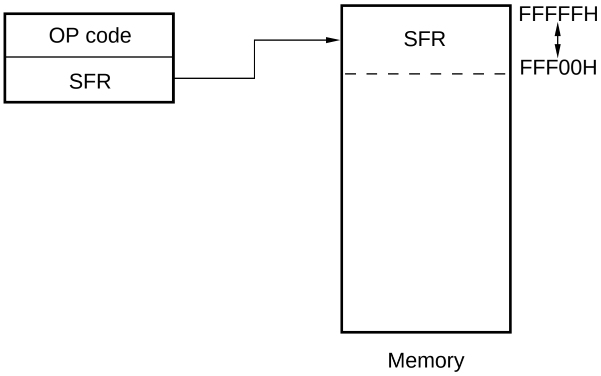
[Function]

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

[Operand format]

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

Figure 3-20. Outline of SFR Addressing



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

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

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

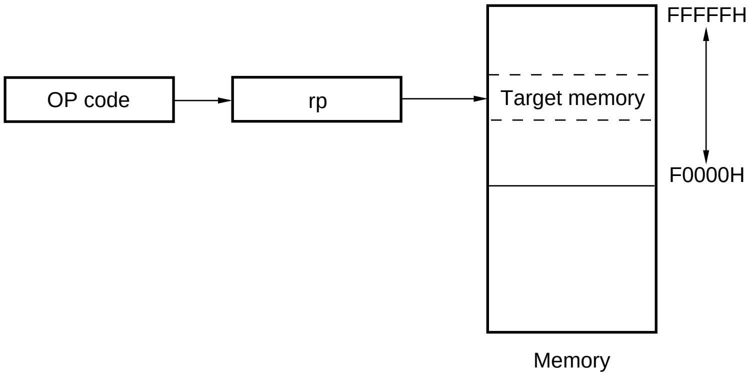
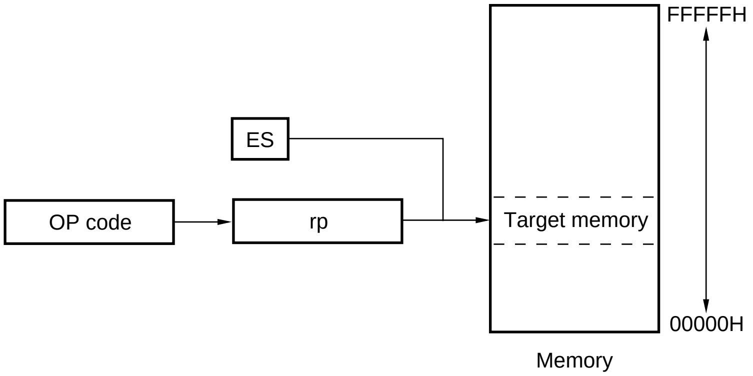


Figure 3-22. Example of ES:[DE], ES:[HL]



3.4.7 Based addressing

[Function]

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

[Operand format]

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

Figure 3-23. Example of [SP+byte]

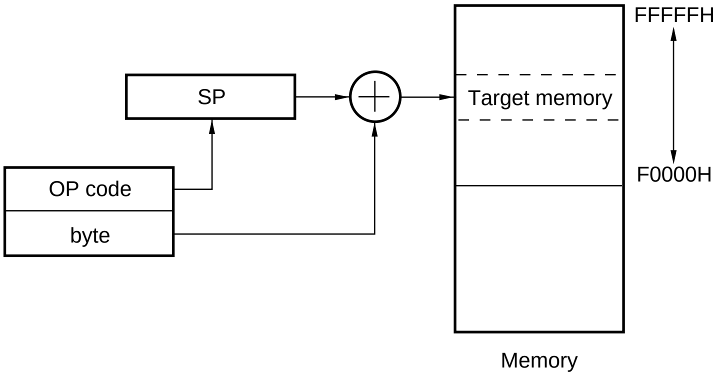


Figure 3-24. Example of [HL + byte], [DE + byte]

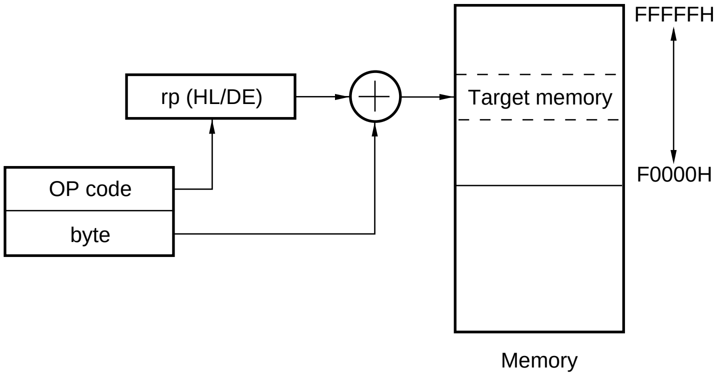


Figure 3-25. Example of word[B], word[C]

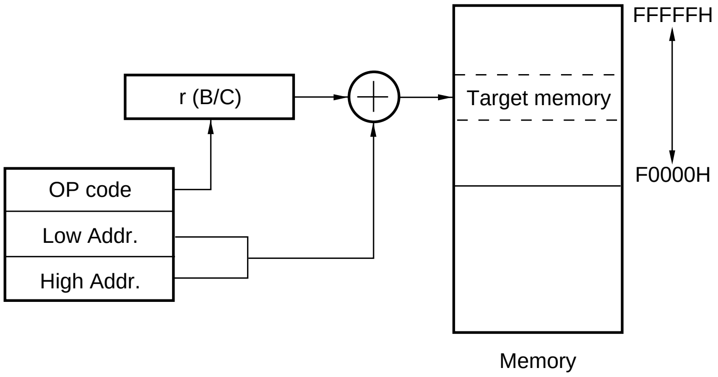


Figure 3-26. Example of word[BC]

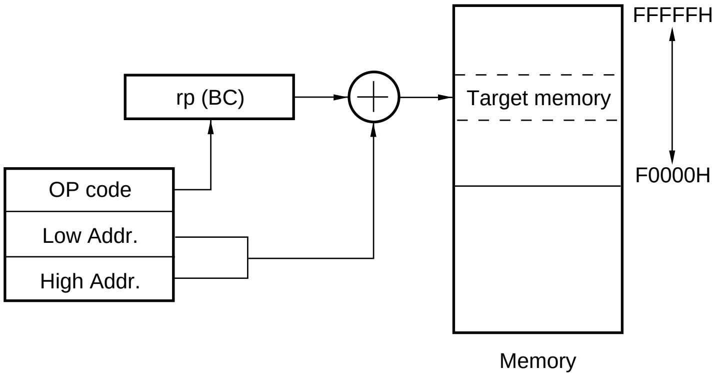


Figure 3-27. Example of ES:[HL + byte], ES:[DE + byte]

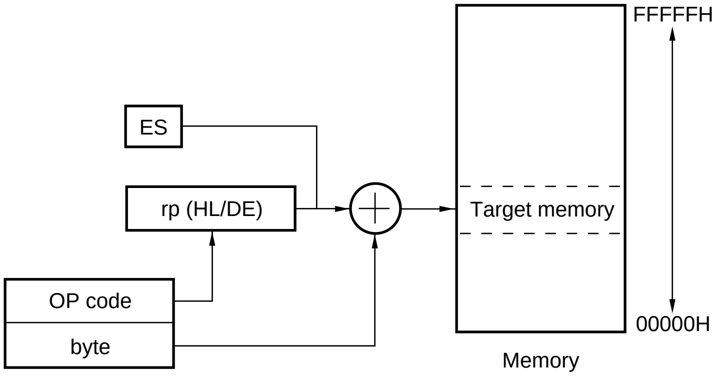


Figure 3-28. Example of ES:word[B], ES:word[C]

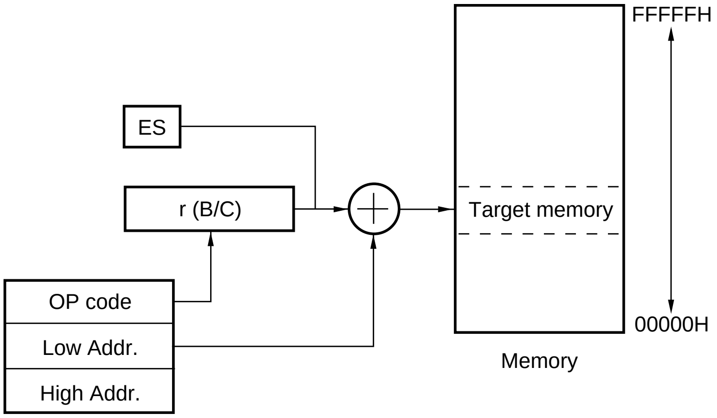
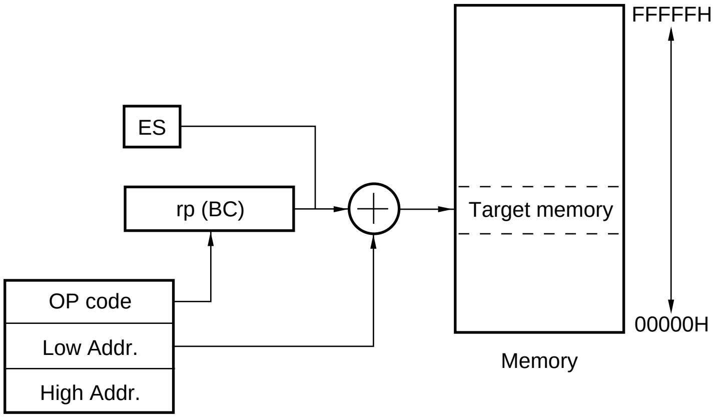


Figure 3-29. Example of ES:word[BC]



3.4.8 Based indexed addressing

[Function]

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

[Operand format]

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

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

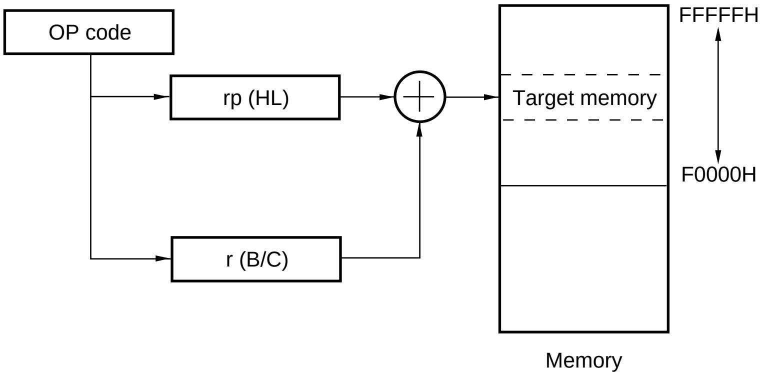
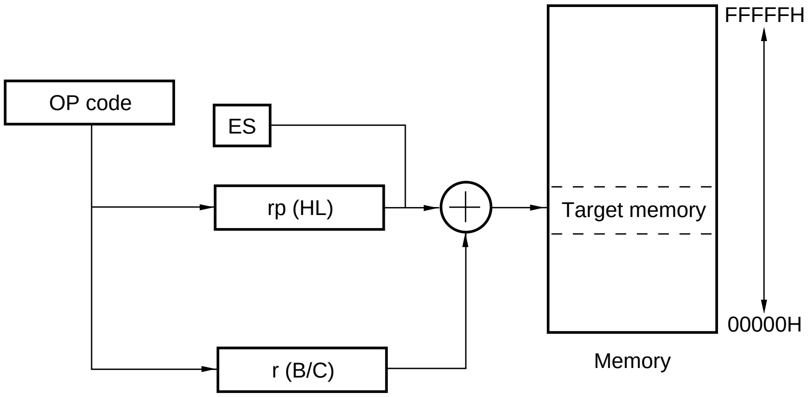


Figure 3-31. Example of ES:[HL+B], ES:[HL+C]



3.4.9 Stack addressing

[Function]

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

Stack addressing is applied only to the internal RAM area.

[Operand format]

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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

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

Table 4-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27, P150, P151, P80 to P83
V_{DD}	• Port pins other than P20 to P27, P150 to P151, P80 to P83 • Pins other than port pins

μPD79F9211 products are provided with the ports shown in Figures 4-1, which enable variety of control operations. The functions of each port are shown in Table 4-2.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 2 PIN FUNCTIONS**.

Figure 4-1. Port Types

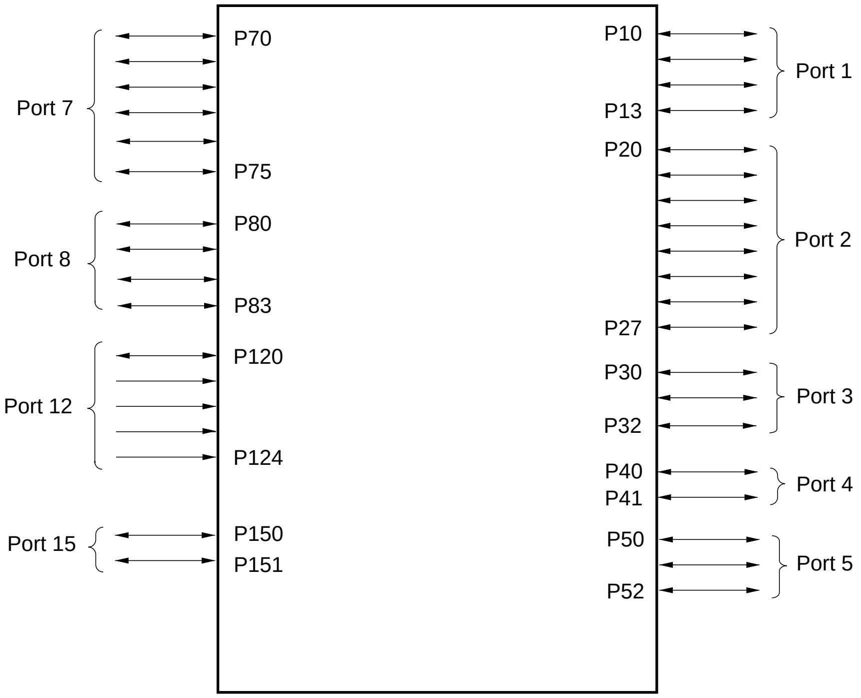


Table 4-2. Port Functions (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P10	I/O	Port 1. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI02/TO02
P11				TI03/TO03
P12				TI04/TO04
P13				TI05/TO05
P20 to P27	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0 to ANI7
P30	I/O	Port 3. 3-bit I/O port. Input of P31 and P32 can be set to TTL buffer. Output of P30 to P32 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SO10/TxD1/TO11
P31				SI10/RxD1/SDA10/INTP1/TI09
P32				$\overline{\text{SCK10}}$/SCL10/INTP2
P40 Note	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	I/O	Port 5. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI06/TO06
P51			Input port	TI07/TO07
P52			Input port	SLTI/SLTO
P70	I/O	Port 7. 6-bit I/O port. Input of P71, P72, P74, and P75 can be set to TTL buffer. Output of P70, P72, P73, and P75 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SO01/INTP4
P71				SI01/INTP5
P72				$\overline{\text{SCK01}}$/INTP6
P73				SO00/TxD0/TO10
P74				SI00/RxD0/TI10
P75				$\overline{\text{SCK00}}$/TI11
P80	I/O	Port 8. 4-bit I/O port. Inputs/output can be specified in 1-bit units. Inputs of P80 to P83 can be set as comparator inputs or operational amplifier inputs.	Input port	CMP0P/TMOFF0/INTP3/OAI
P81				CMP0M
P82				CMP1P/TMOFF1/INTP7
P83				CMP1M

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 **2.2.4 P40, P41 (port 4)**).

Table 4-2. Port Functions (44-pin products) (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P120	I/O	Port 12. 1-bit I/O port and 4-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
P122				X2/EXCLK
P123				XT1
P124				XT2
P150, P151	I/O	Port 15. 2-bit I/O port. Inputs/output can be specified in 1-bit units.	Digital input port	ANI8, ANI9

4.2 Port Configuration

Ports include the following hardware.

Table 4-3. Port Configuration

Item	Configuration
Control registers	Port mode registers (PM1 to PM5, PM7, PM8, PM12, PM15) Port registers (P1 to P5, P7, P8, P12, P15) Pull-up resistor option registers (PU1, PU3 to PU5, PU7, PU12) Port input mode registers (PIM3, PIM7, PIM8) Port output mode registers (POM3, POM7) A/D port configuration register (ADPC)
Port	Total: 37 (CMOS I/O: 33, CMOS input: 4)
Pull-up resistor	Total: 19

4.2.1 Port 1

Port 1 is a 4-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 P10 to P13 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).

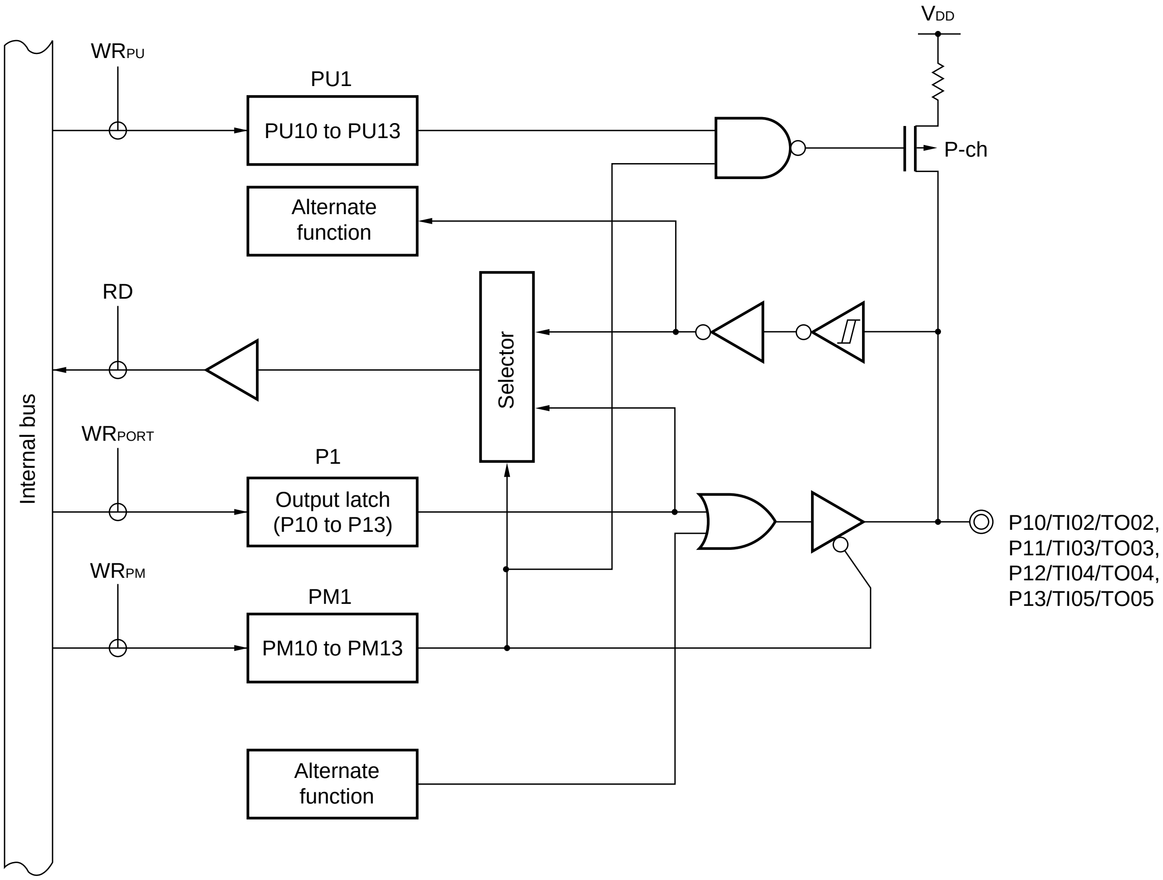
This port can also be used for timer I/O.

Reset signal generation sets port 1 to input mode.

Figure 4-2 shows a block diagram of port 1.

Caution To use P10/TI02/TO02, P11/TI03/TO03, P12/TI04/TO04, or P13/TI05/TO05 as a general-purpose port, set bits 2 to 5 (TO02 to TO05) of timer output register 0 (TO0) and bits 2 to 5 (TOE02 to TOE05) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.

Figure 4-2. Block Diagram of P10 to P13



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

4.2.2 Port 2

Port 2 is a 8-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 P20/ANI0 to P27/ANI7 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using PM2. Use these pins starting from the lower bit.

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

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

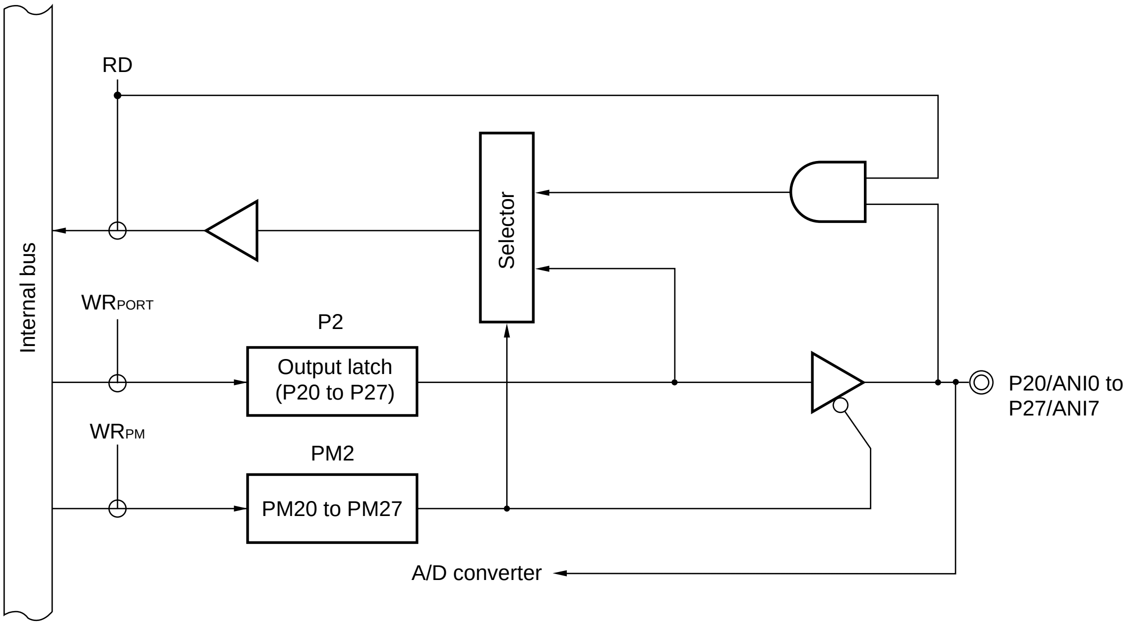
ADPC	PM2	ADS	P20/ANI0 to P27/ANI7 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P20/ANI0 to P27/ANI7 are set in the digital input mode when the reset signal is generated.

Figure 4-3 shows a block diagram of port 2.

Caution Make the AV$_{REF}$ pin the same potential as the V$_{DD}$ pin when port 2 is used as a digital port.

Figure 4-3. Block Diagram of P20 to P27



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

4.2.3 Port 3

Port 3 is a 3-bit I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to P32 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

Input to the P31 and P32 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 3 (PIM3).

Output from the P30 to P32 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 3 (POM3).

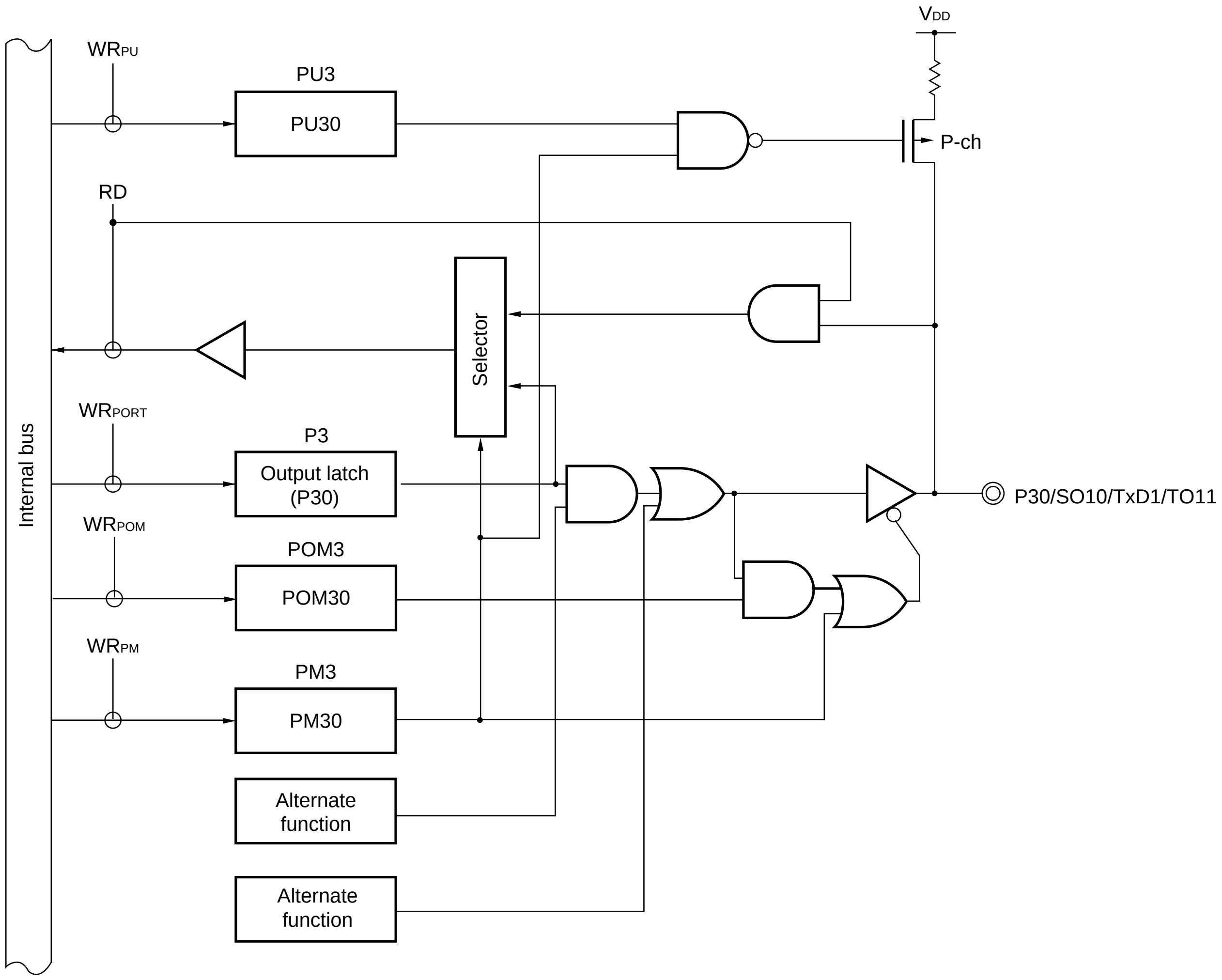
This port can also be used for serial interface data I/O, clock I/O, external interrupt request input, and timer I/O.

Reset signal generation sets port 3 to input mode.

Figures 4-6 and 4-7 show block diagrams of port 3.

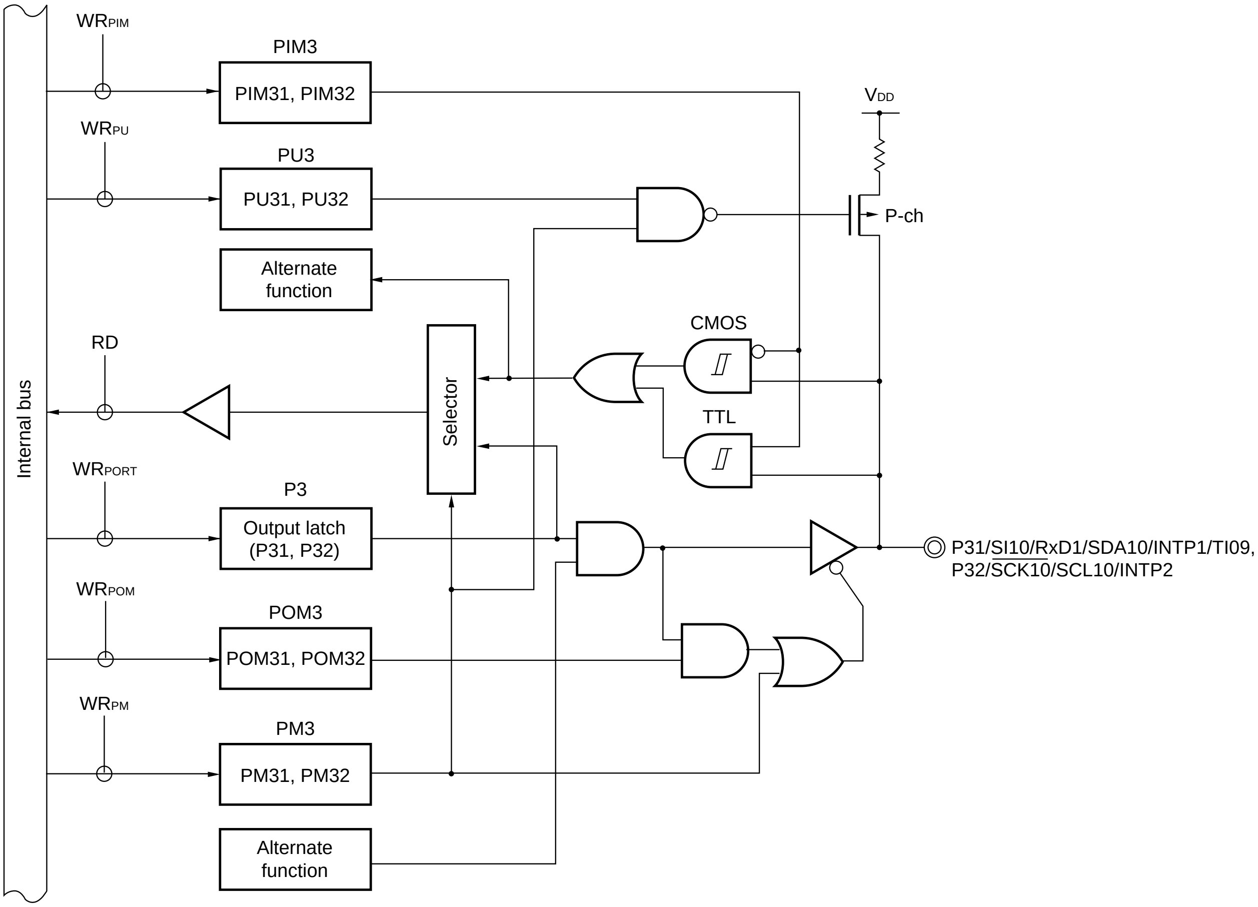
Caution To use P30/SO10/TxD1/TO11, P31/SI10/RxD1/SDA10/INTP1/TI09, P32/$\overline{\text{SCK10}}$/SCL10/INTP2 as a general-purpose port, note the serial array unit setting. For details, refer to Table 13-9 Relationship Between Register Settings and Pins (Channel 2: CSI10, UART1 Transmission, IIC10) and Table 13-10 Relationship Between Register Settings and Pins (Channel 3: UART1 Reception).

Figure 4-4. Block Diagram of P30



P3:	Port register 3
PU3:	Pull-up resistor option register 3
POM3:	Port output mode register 3
PM3:	Port mode register 3
RD:	Read signal
WR××:	Write signal

Figure 4-5. Block Diagram of P31 and P32



P3:	Port register 3
PU3:	Pull-up resistor option register 3
PIM3:	Port input mode register 3
POM3:	Port output mode register 3
PM3:	Port mode register 3
RD:	Read signal
WR××:	Write signal

4.2.4 Port 4

Port 4 is a 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 and 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.

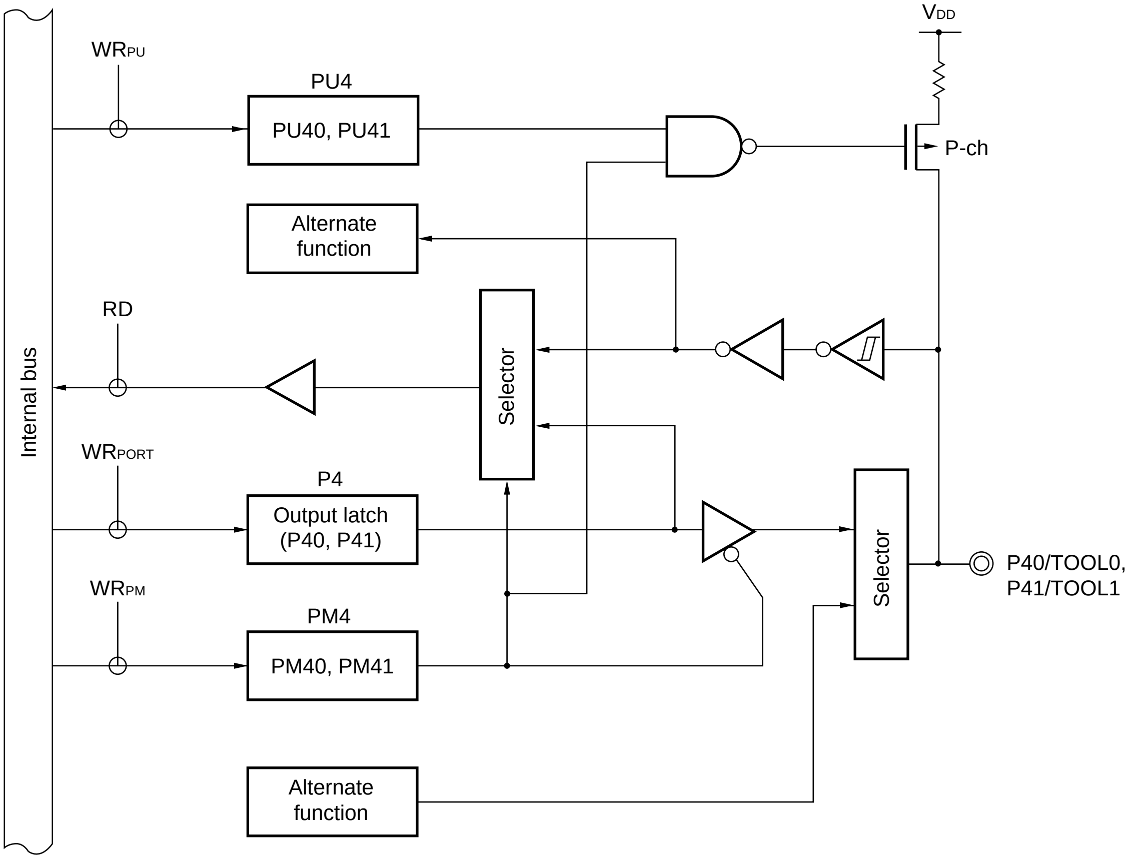
Figure 4-6 shows a block diagram of port 4.

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 4-6. Block Diagram of P40 and P41



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

4.2.5 Port 5

Port 5 is a 3-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P52 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

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

Reset signal generation sets port 5 to input mode.

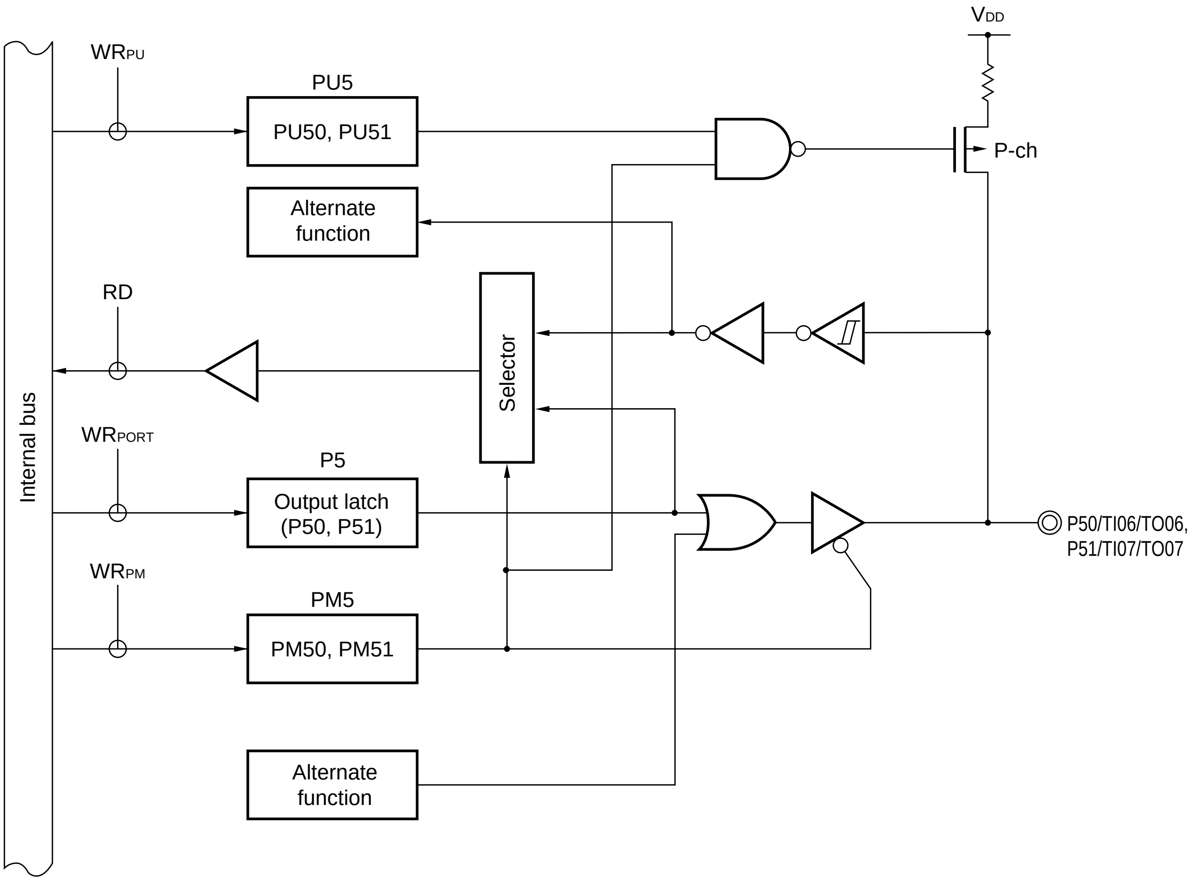
Figures 4-9 and 4-10 show a block diagrams of port 5.

Cautions 1. To use P50/TI06/TO06 or P51/TI07/TO07 as a general-purpose port, set bits 6 and 7 (TO06, TO07) of timer output register 0 (TO0) and bits 6 and 7 (TOE06, TOE07) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

2. To use P52/SLTI/SLTO as a general-purpose port, check which timer I/O pin of which channel n is selected in the input switching control register (ISC) setting. Also, set bit n (TO0n) of timer output register 0 (TO0) and bit n (TOE0n) of timer output enable register 0 (TOE0) to "0", which is the same setting as in the initial state of each.

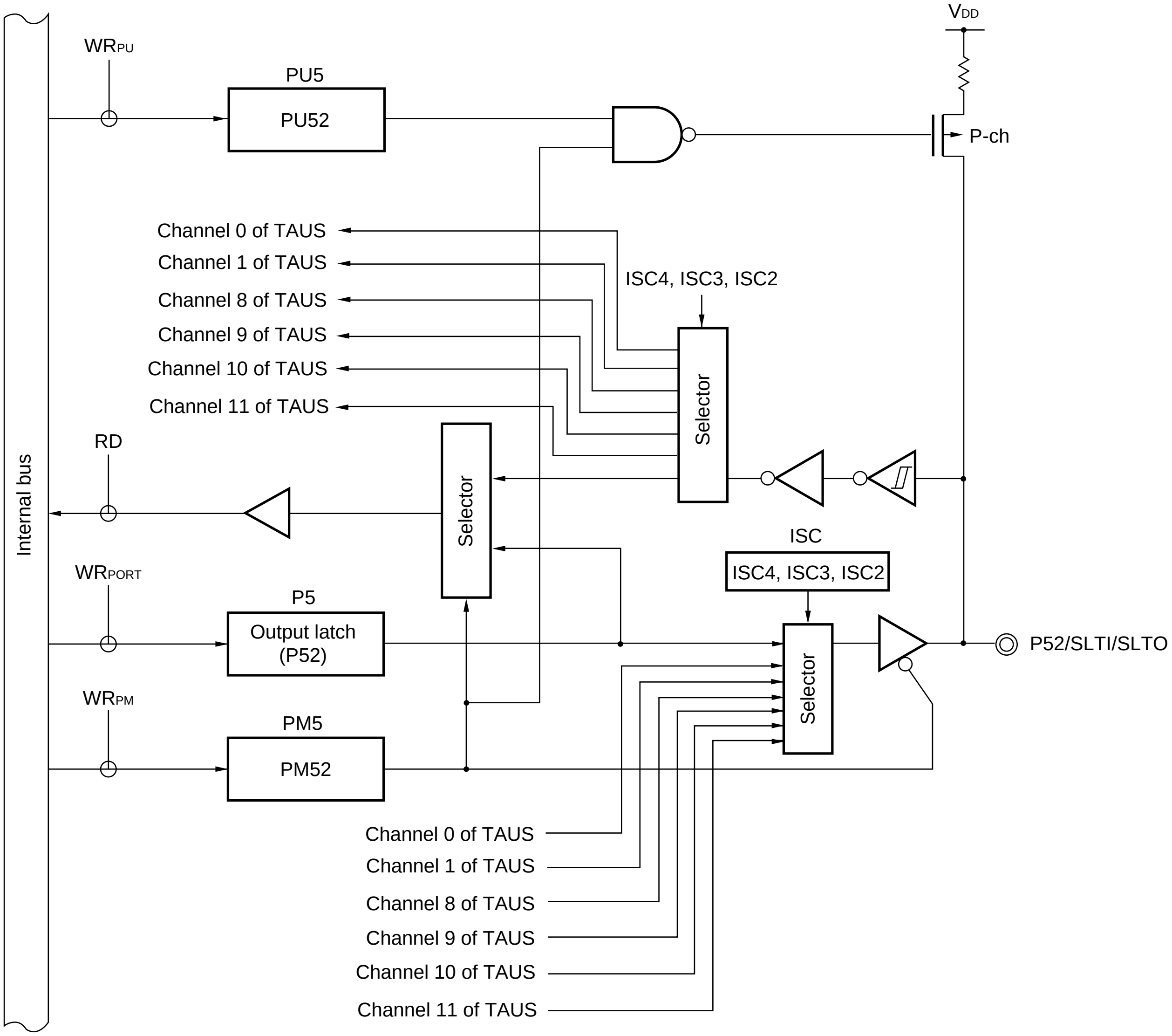
Remark n = 00, 01, 08 to 11

Figure 4-7. Block Diagram of P50 and P51



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

Figure 4-8. Block Diagram of P52



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

4.2.6 Port 7

Port 7 is a 6-bit I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 and P75 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

Input to the P71, P72, P74, and P75 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 7 (PIM7).

Output from the P70, P72, P73, and P75 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 7 (POM7).

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

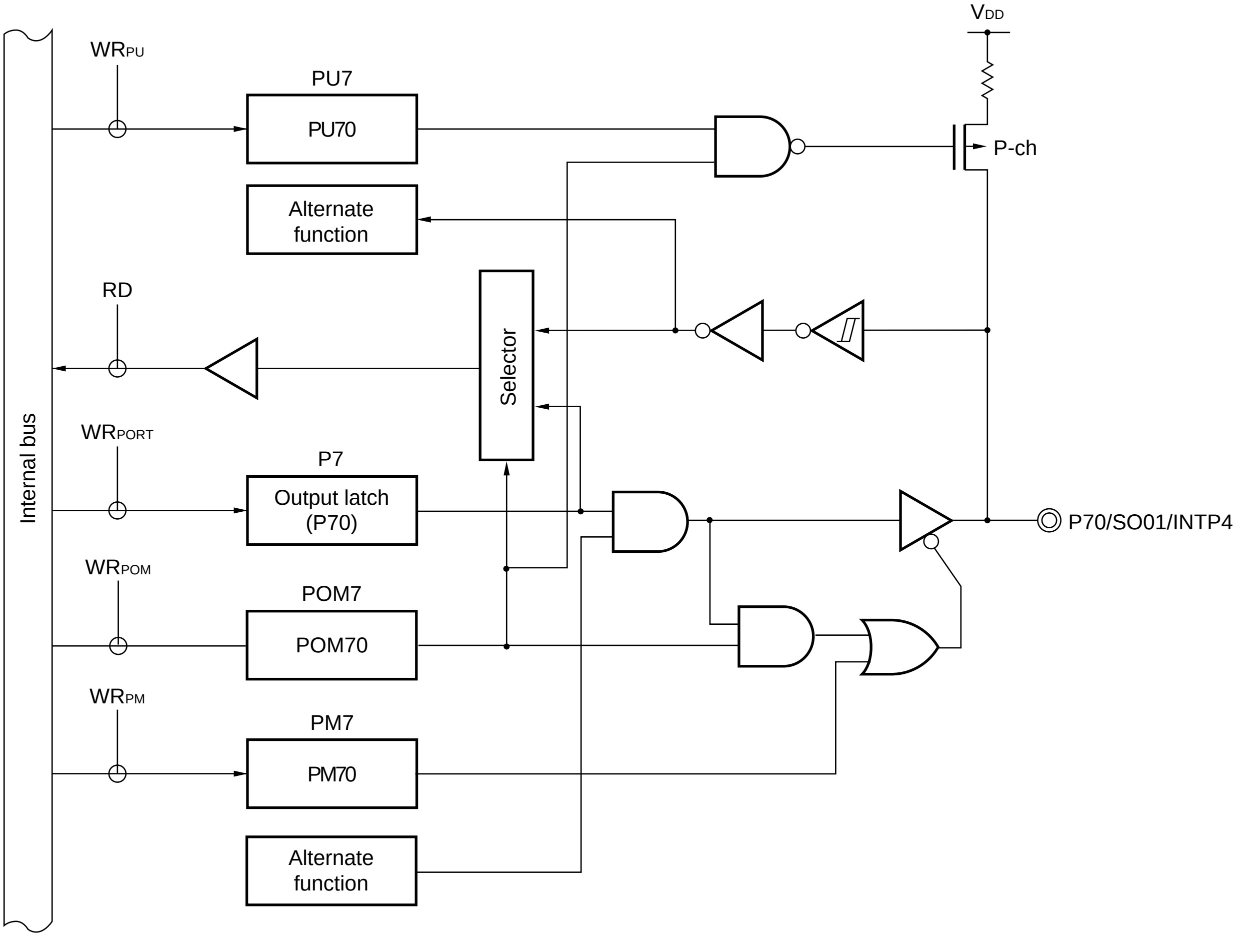
Reset signal generation sets port 7 to input mode.

Figures 4-14 to 4-17 show block diagrams of port 7.

Cautions 1. To use P70/SO01/INTP4, P71/SI01/INTP5, P72/$\overline{\text{SCK01}}$/INTP6, P73/SO00/TxD0/TO10, P74/SI00/RxD0/TI10, P75/$\overline{\text{SCK00}}$/TI11 as a general-purpose port, note the serial array unit setting. For details, refer to Table 13-6 Relationship Between Register Settings and Pins (Channel 0: CSI00, UART0 Transmission) and Table 13-8 Relationship Between Register Settings and Pins (Channel 1: CSI01, UART0 Reception).

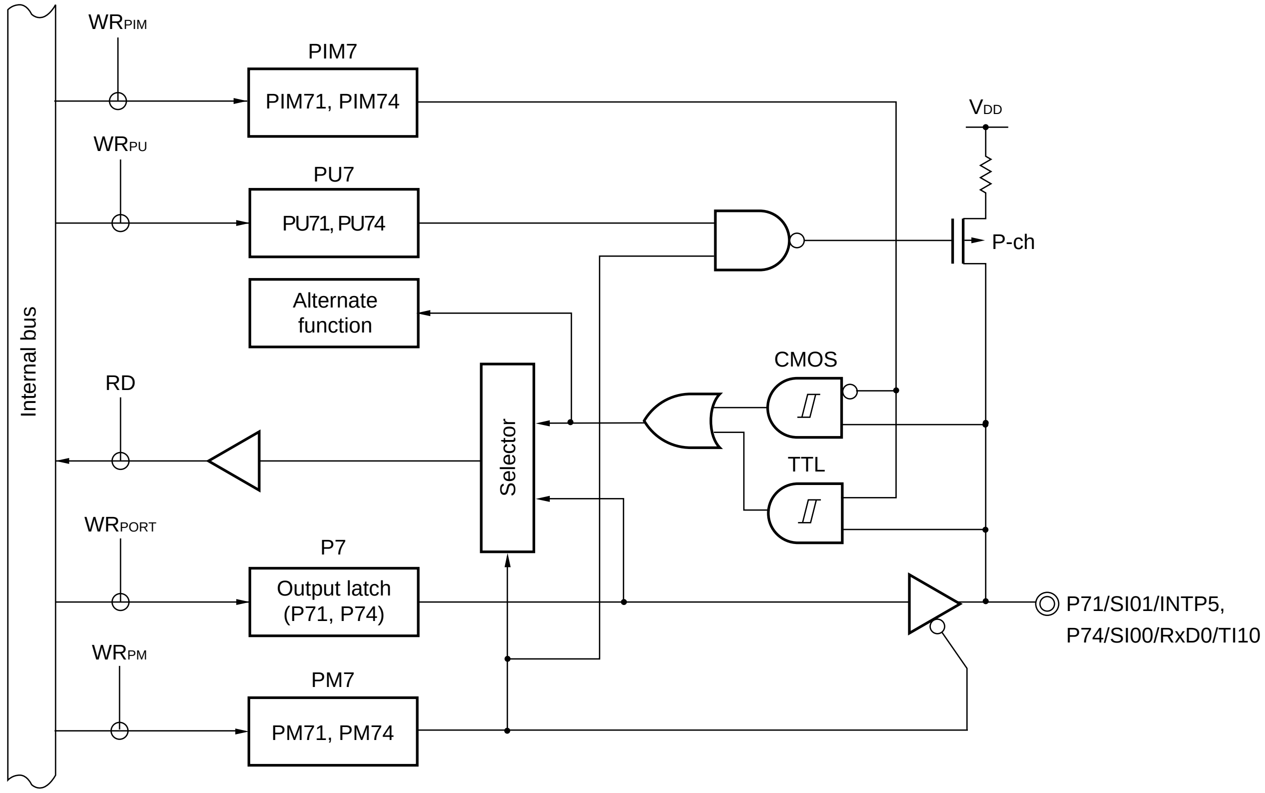
2. To use P73/SO00/TxD0/TO10 as a general-purpose port, set bit 10 (TO10) of timer output register 0 (TO0) and bit 10 (TOE10) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.

Figure 4-9. Block Diagram of P70



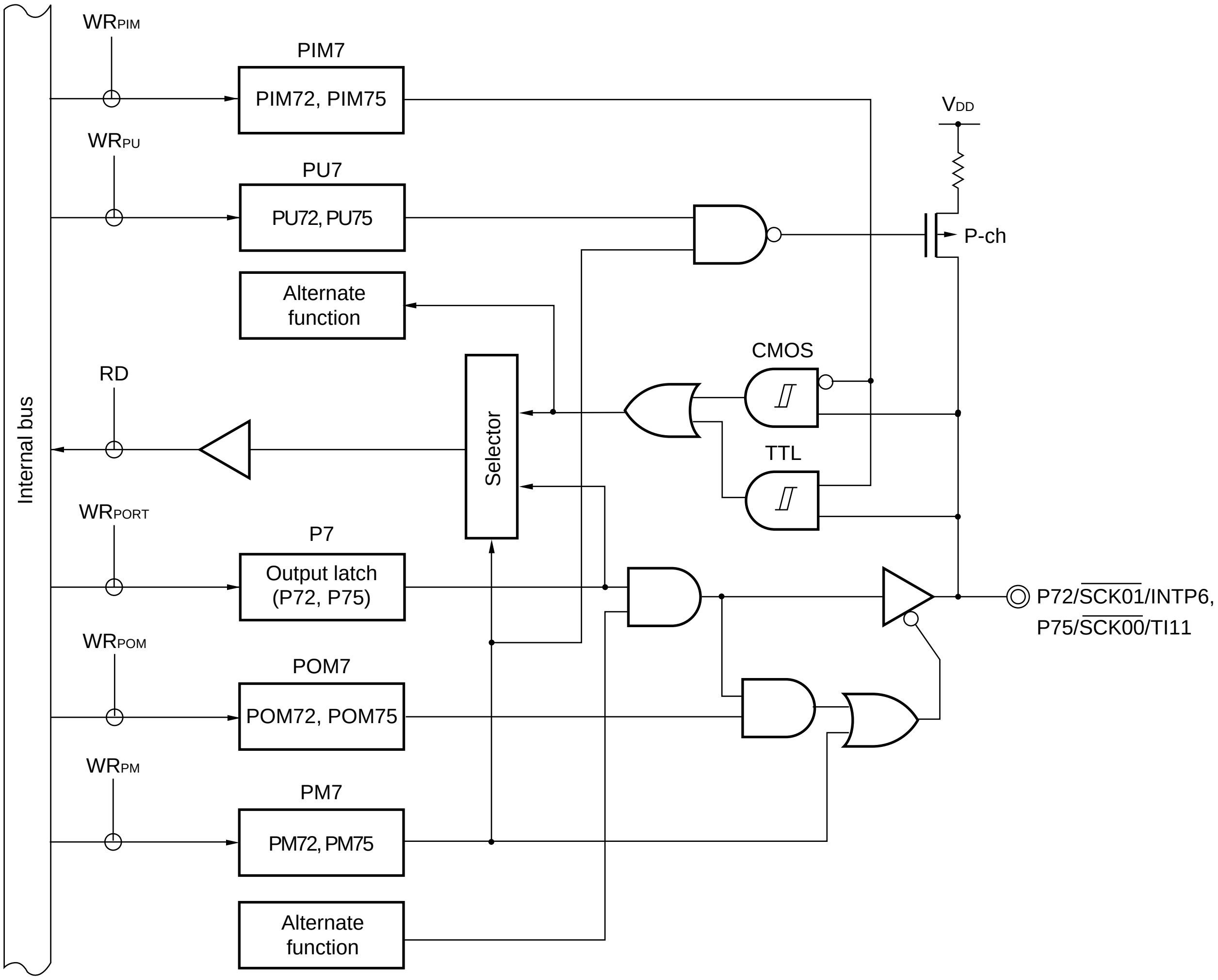
P7:	Port register 7
PU7:	Pull-up resistor option register 7
POM7:	Port output mode register 7
PM7:	Port mode register 7
RD:	Read signal
WR××:	Write signal

Figure 4-10. Block Diagram of P71 and P74



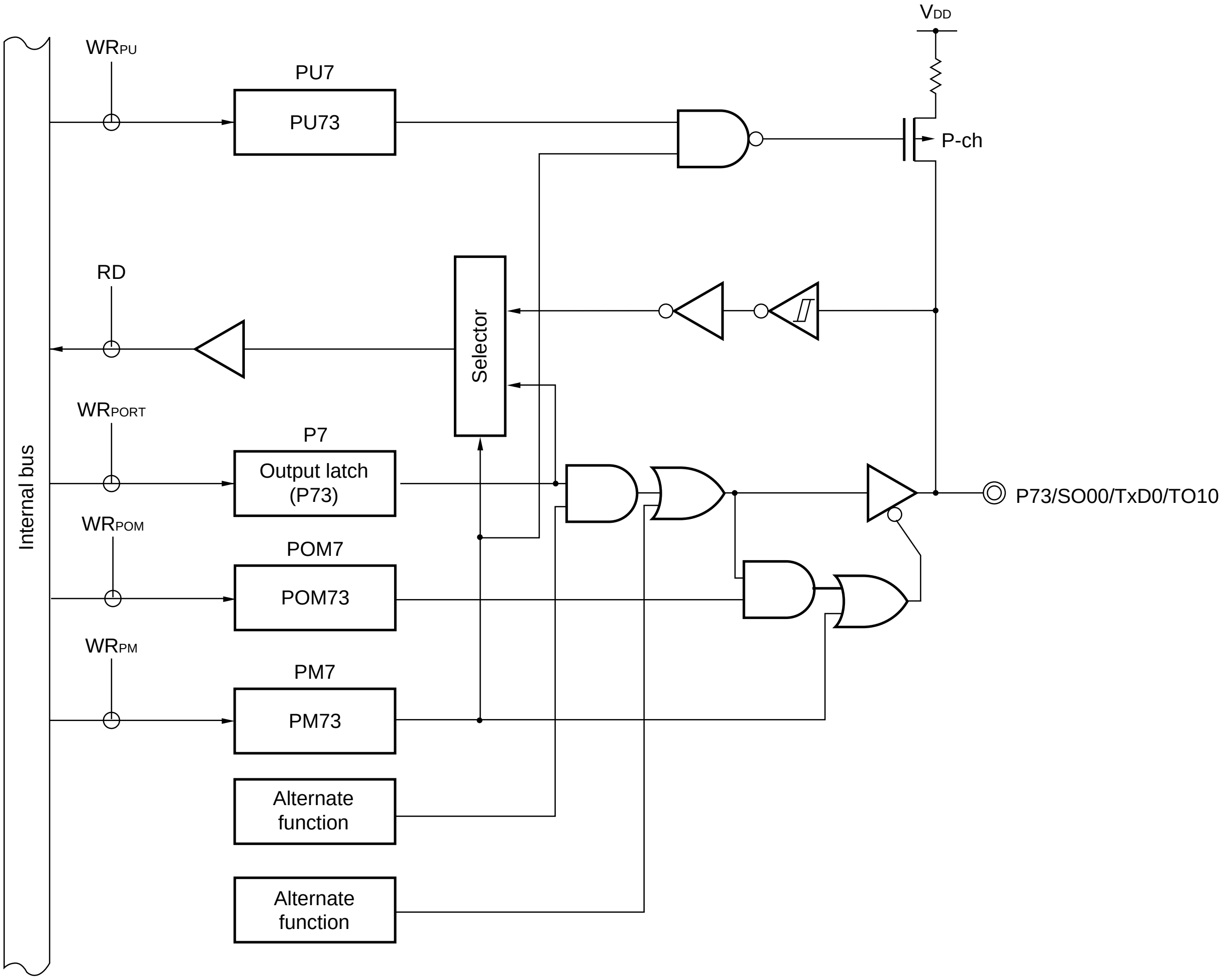
P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
PIM7:	Port input mode register 7
RD:	Read signal
WR××:	Write signal

Figure 4-11. Block Diagram of P72 and P75



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
PIM7:	Port input mode register 7
POM7:	Port output mode register 7
RD:	Read signal
WR××:	Write signal

Figure 4-12. Block Diagram of P73



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

4.2.7 Port 8

Port 8 is a 4-bit I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8).

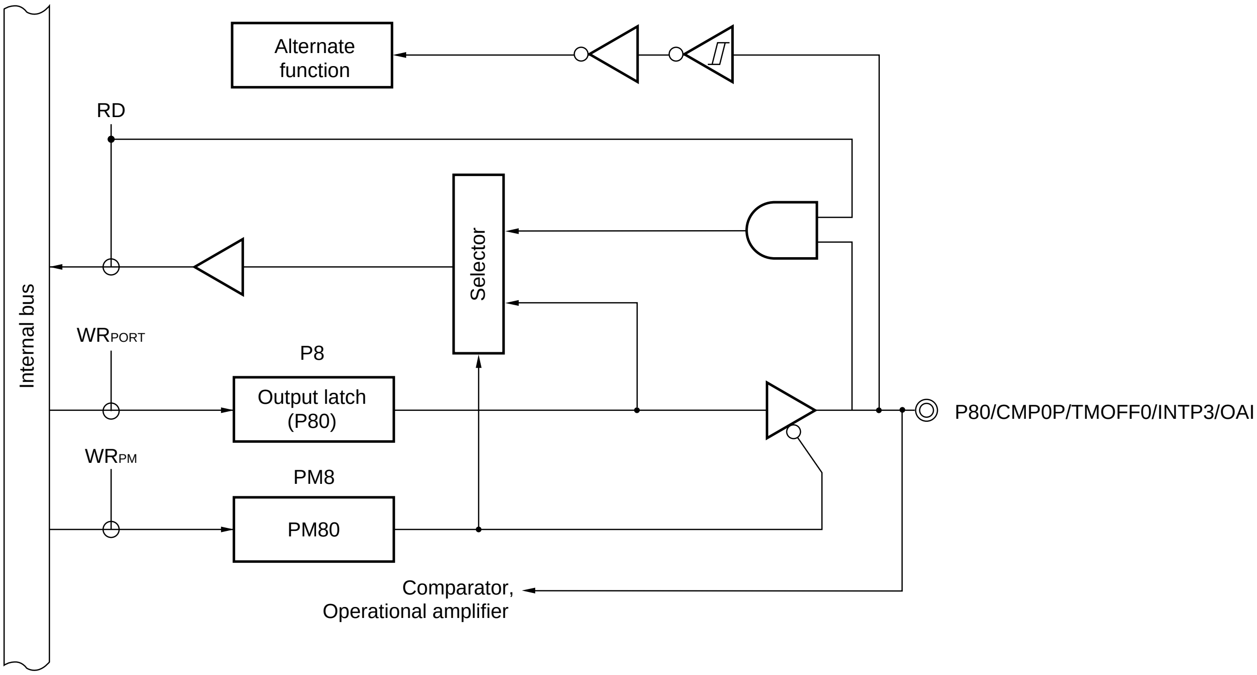
Inputs to the P80 to P83 pins must be enabled or disabled in 1-bit units using port input mode register 8 (PIM8).

This port can also be used for an input voltage on the side of comparators 0 and 1 (+), an input voltage on the sides of comparators 0 and 1 (–), a timer pin Hi-Z control input, an external interrupt request input, and an operational amplifier input.

Reset signal generation sets port 8 to input mode.

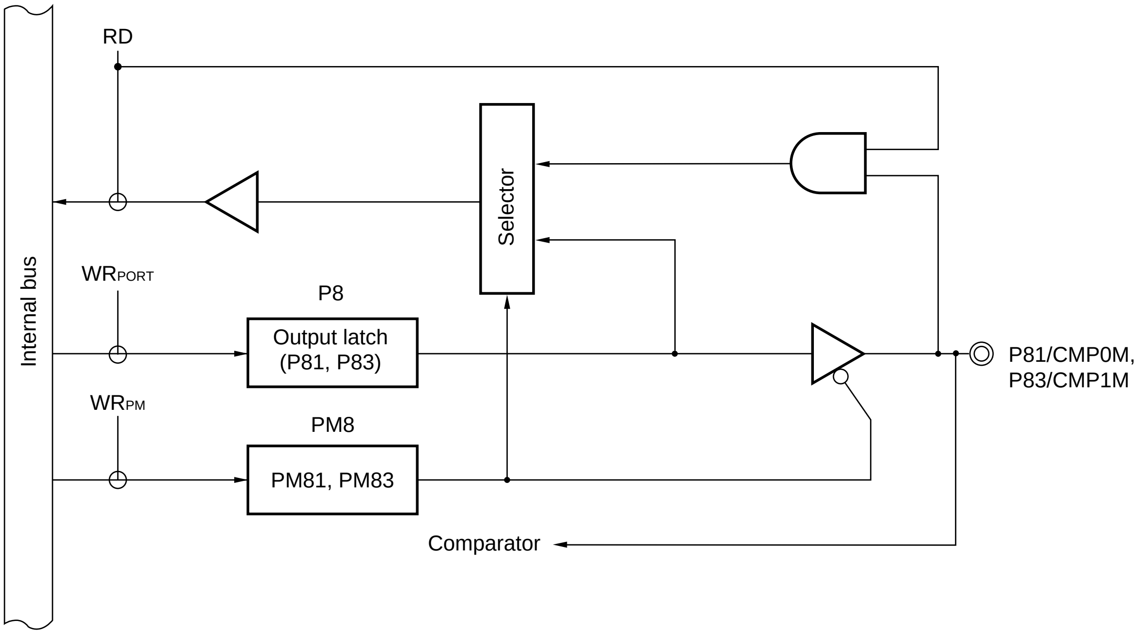
Figures 4-18 to 4-20 show block diagrams of port 8.

Figure 4-13. Block Diagram of P80



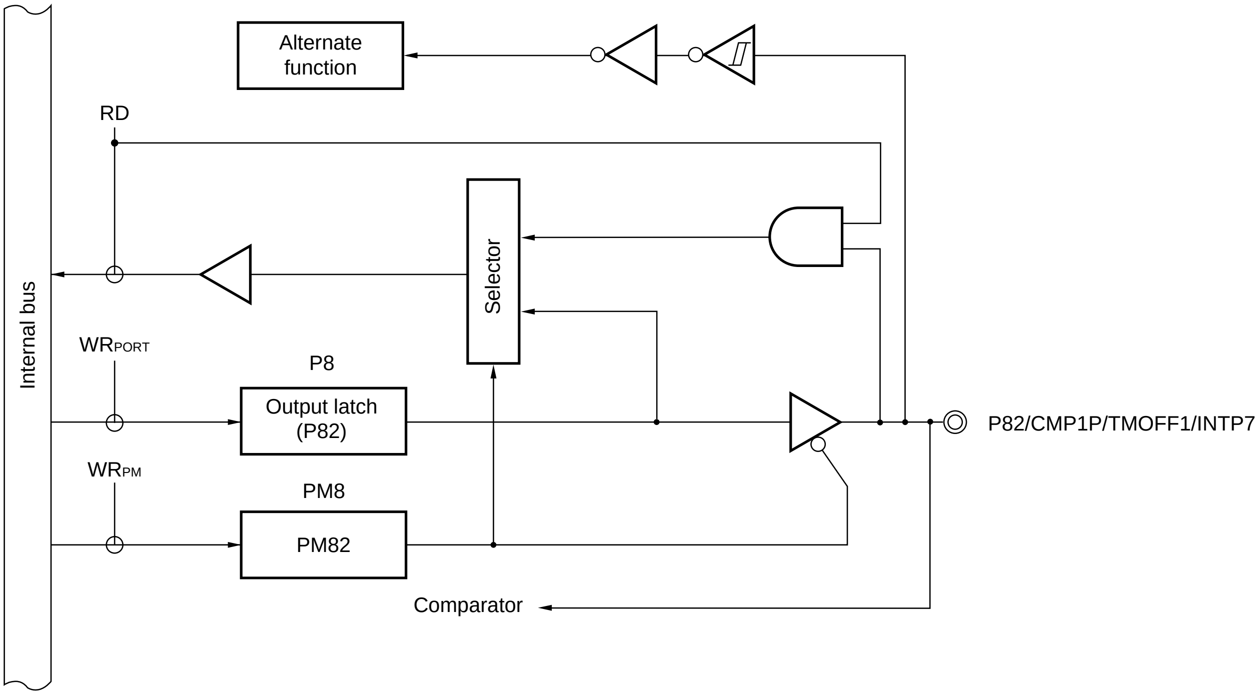
P8:	Port register 8
PM8:	Port mode register 8
RD:	Read signal
WR××:	Write signal

Figure 4-14. Block Diagram of P81 and P83



P8:	Port register 8
PM8:	Port mode register 8
RD:	Read signal
WR××:	Write signal

Figure 4-15. Block Diagram of P82



P8:	Port register 8
PM8:	Port mode register 8
RD:	Read signal
WR××:	Write signal

4.2.8 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 to P124 are 4-bit input ports.

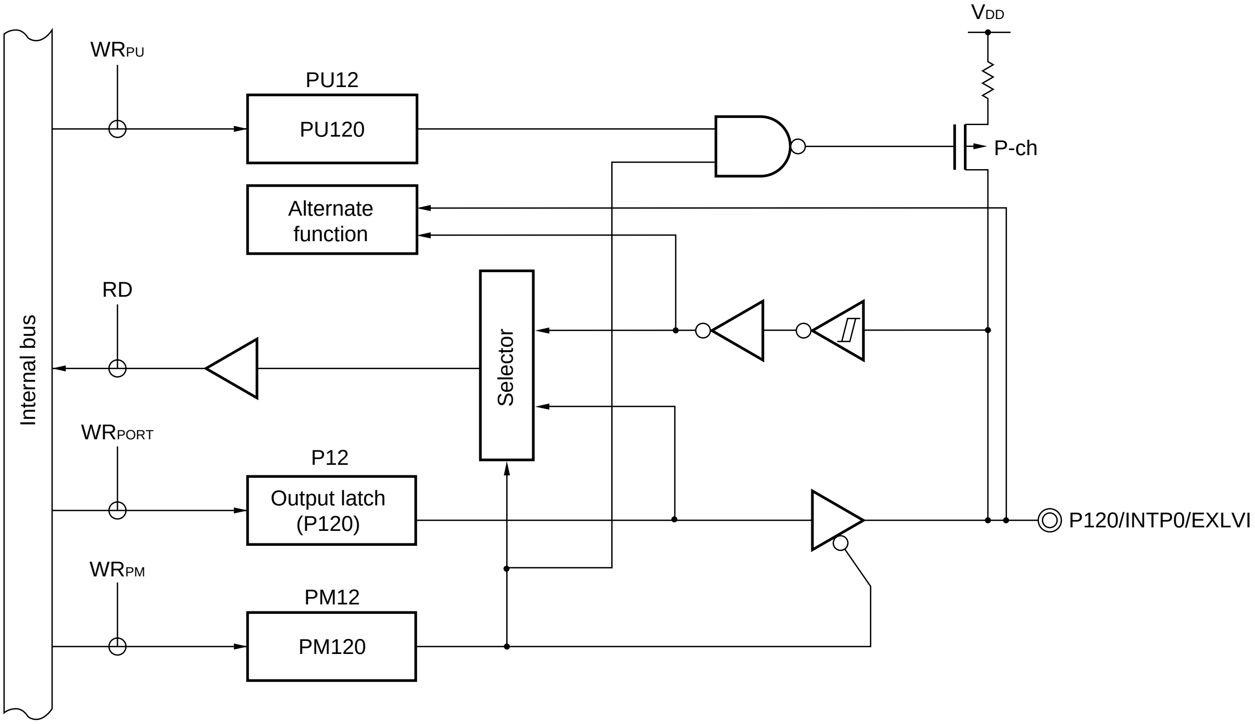
This port can also be used for external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, and external clock input for main system clock.

Reset signal generation sets port 12 to input mode.

Figures 4-21 and 4-24 show block diagrams of port 12.

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

Figure 4-16. Block Diagram of P120



P12: Port register 12

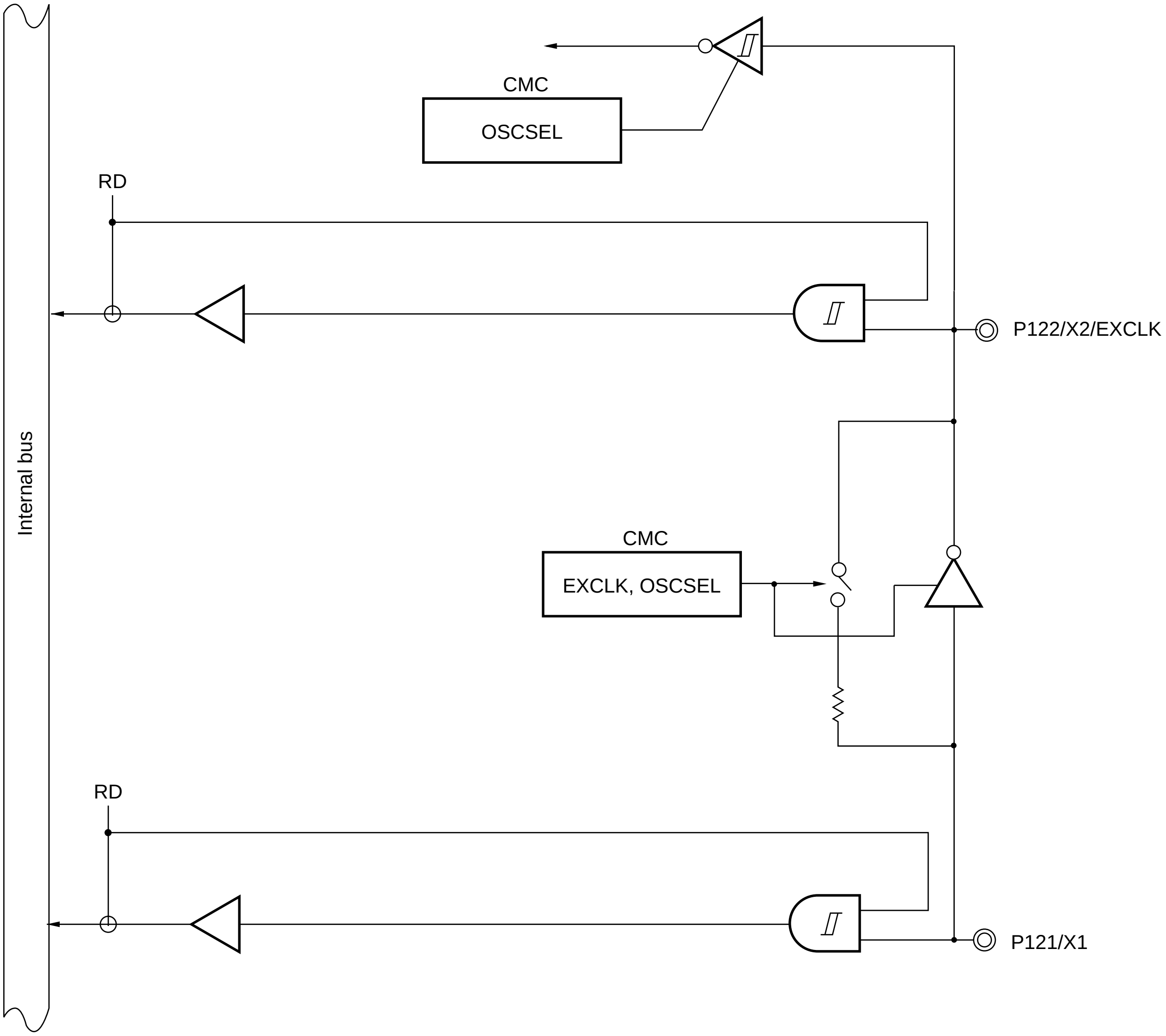
PU12: Pull-up resistor option register 12

PM12: Port mode register 12

RD: Read signal

WR××: Write signal

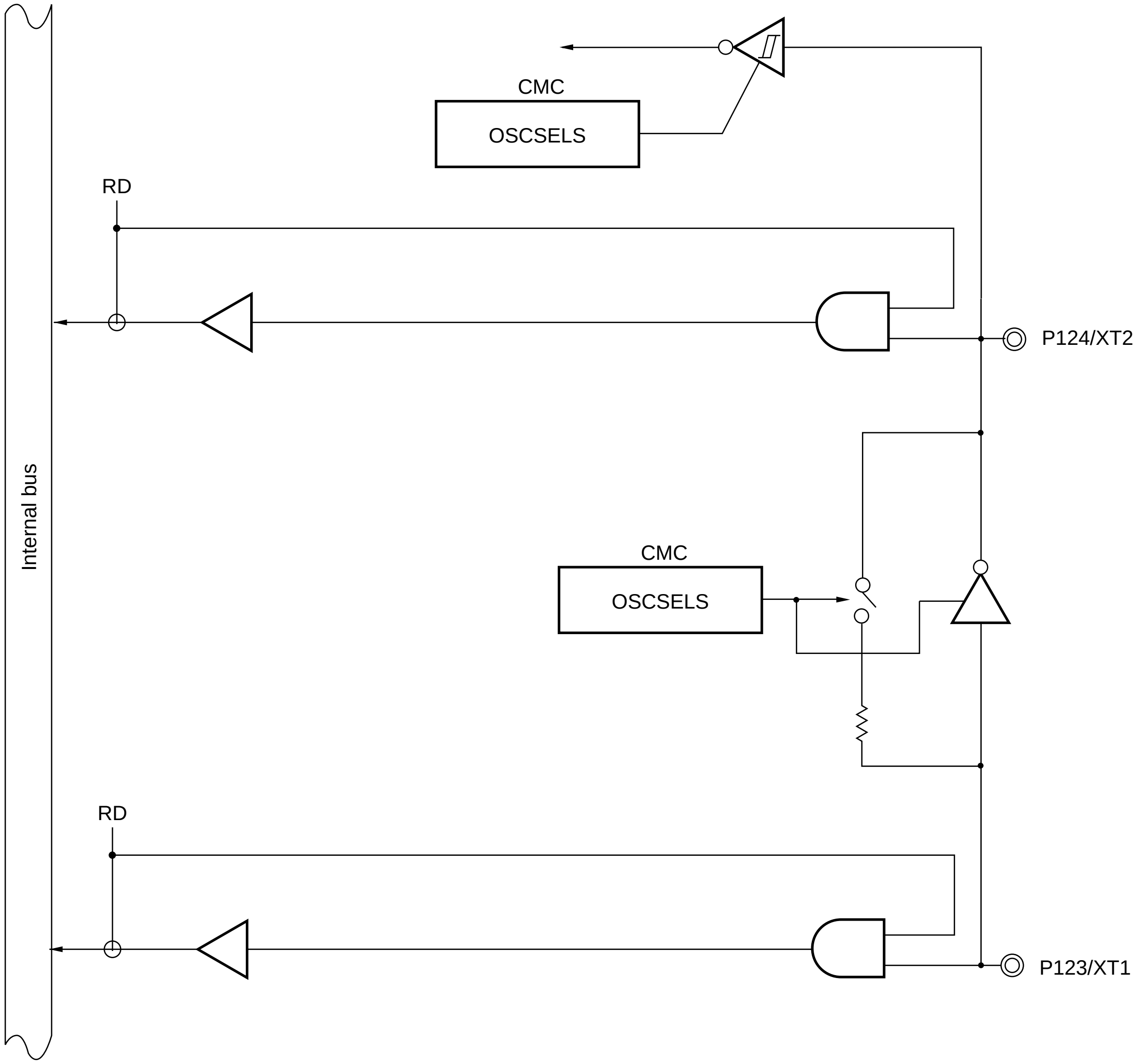
Figure 4-17. Block Diagram of P121 and P122



CMC: Clock operation mode control register

RD: Read signal

Figure 4-18. Block Diagram of P123 and P124



CMC: Clock operation mode control register
RD: Read signal

4.2.9 Port 15

Port 15 is an 2-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 and P151/ANI9 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 and P151/ANI9 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM15.

Table 4-5. Setting Functions of P150/ANI8 and P151/ANI9 Pins

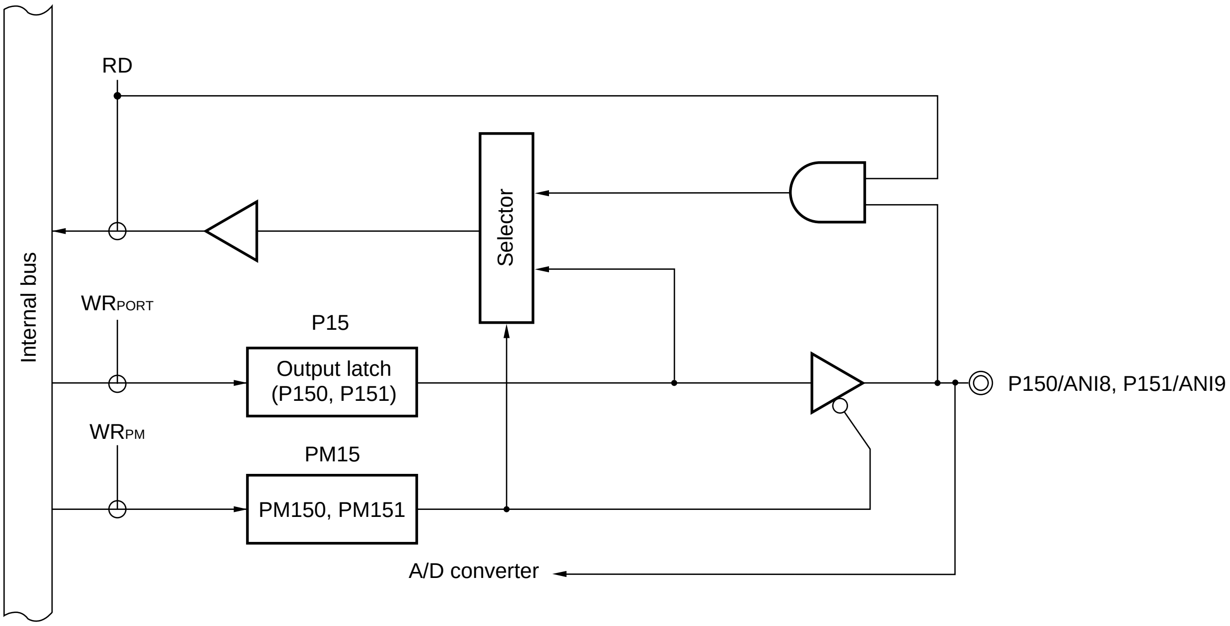
ADPC	PM15	ADS	P150/ANI8 and P151/ANI9 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P150/ANI8 and P151/ANI9 are set in the digital input mode when the reset signal is generated.

Figure 4-19 shows block diagram of port 15.

Caution Make the AV$_{REF0}$ pin the same potential as the V$_{DD}$ pin when port 15 is used as a digital port.

Figure 4-19. Block Diagram of P150 and P151



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

4.3 Registers Controlling Port Function

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

- Port mode registers (PM1 to PM5, PM7, PM8, PM12, PM15)
- Port registers (P1 to P5, P7, P8, P12, P15)
- Pull-up resistor option registers (PU1, PU3 to PU5, PU7, PU12)
- Port input mode registers (PIM3, PIM7, PIM8)
- Port output mode registers (POM3, POM7)
- A/D port configuration register (ADPC)

(1) Port mode registers (PM1 to PM8, PM12, 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 **4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function**.

Figure 4-20. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM1	1	1	1	1	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	1	1	1	PM52	PM51	PM50	FFF25H	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM8	1	1	1	1	PM83	PM82	PM81	PM80	FFF28H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM15	1	1	1	1	1	1	PM151	PM150	FFF2FH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 1 to 8, 12, 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set the following bits to 1.

- **Bits 4 to 7 of the PM1 register**
- **Bits 3 to 7 of the PM3 register**
- **Bits 2 to 7 of the PM4 register**
- **Bits 3 to 7 of the PM5 register**
- **Bits 6 and 7 of the PM7**
- **Bits 4 to 7 of the PM8 register**
- **Bits 1 to 7 of the PM12 register**
- **Bits 2 to 7 of the PM15 register**

(2) Port registers (P1 to P8, P12, 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 4-21. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P1	0	0	0	0	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	0	P32	P31	P30	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	P52	P51	P50	FFF05H	00H (output latch)	R/W
P7	0	0	P75	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P8	0	0	0	0	P83	P82	P81	P80	FFF08H	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W[Note]
P15	0	0	0	0	0	0	P151	P150	FFF0FH	00H (output latch)	R/W

Pmn	m = 1 to 8, 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

Note . P121 to P124 are read-only.

(3) Pull-up resistor option registers (PU1, PU3 to PU5, PU7, PU12)

These registers specify whether the on-chip pull-up resistors of P10 to P13, P30 to P32, P40, P41, P50 to P52, P70 to P75, or P120 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 PU1, PU3 to PU5, PU7, and PU12. 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 PU1, PU3 to PU5, PU7, and PU12.

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

Reset signal generation clears these registers to 00H.

Figure 4-22. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU1	0	0	0	0	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	PU32	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	00H	R/W
PU5	0	0	0	0	0	PU52	PU51	PU50	F0035H	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W

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

(4) Port input mode registers (PIM3, PIM7, PIM8)

PIM3 and PIM7 registers set the input buffer of P31, P32, P71, P72, P74, or P75 in 1-bit units.

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

PIM8 is used to enable or disable the inputs to P80 to P83 in 1-bit units. When using a comparator or an operational amplifier, the inputs are disabled by software processing. To use port functions and alternative functions, the inputs must be enabled, because they are disabled by default.

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

Reset signal generation clears these registers to 00H.

Figure 4-23. Format of Port Input Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM3	0	0	0	0	0	PIM32	PIM31	0	F0043H	00H	R/W
PIM7	0	0	PIM75	PIM74	0	PIM72	PIM71	0	F0047H	00H	R/W
PIM8	0	0	0	0	PIM83	PIM82	PIM81	PIM80	F0048H	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 3 and 7; n = 1, 2, 4, 5)
0	Normal input buffer
1	TTL input buffer

PIM8n	P8n pin input buffer selection (n = 0 to 3)
0	Disables input
1	Enables input

(5) Port output mode registers (POM3, POM7)

These registers set the output mode of P30 to P32, P70[Note], P72[Note], P73, or P75[Note] in 1-bit units.

N-ch open drain output (V_{DD} tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA10 pin during simplified I^{2}C communication with an external device of the same potential.

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

Reset signal generation clears these registers to 00H.

Figure 4-24. Format of Port Input Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM3	0	0	0	0	0	POM32	POM31	POM30	F0053H	00H	R/W
POM7	0	0	POM75	0	POM73	POM72	0	POM70	F0057H	00H	R/W

POMmn	Pmn pin output mode selection (m = 3 and 7; n = 0 to 3 and 5)
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 P20/ANI0 to P27/ANI7 and P150/ANI8 to P151/ANI9Note 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 4-25. 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

ADP C4	ADP C3	ADP C2	ADP C1	ADP C0	Analog input (A)/digital I/O (D) switching									
					Port 5		Port 2							
					ANI9 /P151	ANI8 /P150	ANI7 /P27	ANI6 /P26	ANI5 /P25	ANI4 /P24	ANI3 /P23	ANI2 /P22	ANI1 /P21	ANI0 /P20
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	A	A	D	D	D	D	D	D
0	0	1	1	1	A	A	A	D	D	D	D	D	D	D
0	1	0	0	0	A	A	D	D	D	D	D	D	D	D
0	1	0	0	1	A	D	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D
Other than the 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 analog input channel specification register (ADS).

4.4 Port Function Operations

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

4.4.1 Writing to I/O port

(1) Output mode

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

(2) Input mode

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

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on I/O port

(1) Output mode

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

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change.
The data of the output latch is cleared when a reset signal is generated.

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

When parts of port 3 and 7 operate with V_{DD} = 4.0 V to 5.5 V, I/O connections with an external device that operates on a 2.5V or 3 V power supply voltage are possible.

Regarding inputs, CMOS/TTL switching is possible on a bit-by-bit basis by port input mode registers (PIM3 and PIM7).

Moreover, regarding outputs, different power potentials can be supported by switching the output buffer to the N-ch open drain (V_{DD} withstand voltage) by the port output mode registers (POM3 and POM7).

(1) Setting procedure when using I/O pins of UART0, UART1 CSI00, CSI01, and CSI10 functions

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

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

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

In case of UART0:	P74
In case of UART1:	P31
In case of CSI00:	P74, P75
In case of CSI01:	P71, P72
In case of CSI10:	P31, P32

<3> Set the corresponding bit of the PIMn register to 1 to switch to the TTL input buffer.

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

Remark n = 3 and 7

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

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

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

In case of UART0:	P73
In case of UART1:	P30
In case of CSI00:	P73, P75
In case of CSI01:	P70, P72
In case of CSI10:	P30, P32

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

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

<5> Set the output mode by manipulating the PMn register.
At this time, the output data is high level, so the pin is in the Hi-Z state.

<6> Operation is done only in the low level according to the operating status of the serial array unit.

Remark n = 3 and 7

(2) Setting procedure when using I/O pins of simplified IIC10 functions

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

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

In case of simplified IIC10: P31, P32

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

<4> Set the corresponding bit of the POM3 register to 1 to set the N-ch open drain output (V_{DD} withstand voltage) mode.

<5> Set the corresponding bit of the PM3 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 I^2C mode.

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

To use the alternate function of a port pin, set the port mode register and output latch as shown in Table 4-6.

Table 4-6. Settings of Port Mode Register and Output Latch When Using Alternate Function (1/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P10	TI02	Input	1	×
	TO02	Output	0	0
P11	TI03	Input	1	×
	TO03	Output	0	0
P12	TI04	Input	1	×
	TO04	Output	0	0
P13	TI05	Input	1	×
	TO05	Output	0	0
P20 to P27[Note]	ANI0 to ANI7[Note]	Input	1	×
P30	SO10	Output	0	1
	TxD1	Output	0	1
	TO11	Output	0	0
P31	SI10	Input	1	×
	RxD1	Input	1	×
	SDA10	I/O	0	1
	INTP1	Input	1	×
	TI09	Input	1	×
P32	$\overline{\text{SCK10}}$	Input	1	×
		Output	0	1
	SCL10	I/O	0	1
	INTP2	Input	1	×
P40	TOOL0	I/O	×	×
P41	TOOL1	Output	×	×
P50	TI06	Input	1	×
	TO06	Output	0	0
P51	TI07	Input	1	×
	TO07	Output	0	0
P52	SLTI	Input	1	×
	SLTO	Output	0	0

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

(**Note** is listed on the next page after next.)

Table 4-6. Settings of Port Mode Register and Output Latch When Using Alternate Function (2/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P70	SO01	Output	0	1
	INTP4	Input	1	×
P71	SI01	Input	1	×
	INTP5	Input	1	×
P72	$\overline{\text{SCK01}}$	Input	1	×
		Output	0	1
	INTP6	Input	1	×
P73	SO00	Output	0	1
	TxD0	Output	0	1
	TO10	Output	0	0
P74	SI00	Input	1	×
	RxD0	Input	1	×
	TI11	Input	1	×
P75	$\overline{\text{SCK00}}$	Input	1	×
		Output	0	1
	TI11	Input	1	×
P80[Note]	CMP0P	Input	1	×
	TMOFF0	Input	1	×
	INTP3	Input	1	×
	OAI[Note]	Input	1	×
P81	CMP0M	Input	1	×
P82	CMP1P	Input	1	×
	TMOFF1	Input	1	×
	INTP7	Input	1	×
P83	CMP1M	Input	1	×
P120	INTP0	Input	1	×
	EXLVI	Input	1	×
P150 to P151 [Note]	ANI8 to ANI9[Note]	Input	1	×

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

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

Note. The function of the ANI0/P20 to ANI7/P27, ANI8/P150 to ANI9/P151, and OAI/P80 pins can be selected by using the A/D port configuration register (ADPC), the analog input channel specification register (ADS), PM2, PM15, and PM8.

Table 4-7. Setting Functions of ANI0/P20 to ANI7/P27, ANI8/P150 to ANI9/P151, and OAI/P80 Pins

ADPC	PM2, PM15, PM8	ADS	ANI0/P20 to ANI7/P27, ANI8/P150 to ANI9/P151, and OAI/P80 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

4.6 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 P13 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 0FH.

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 μPD79F9211.

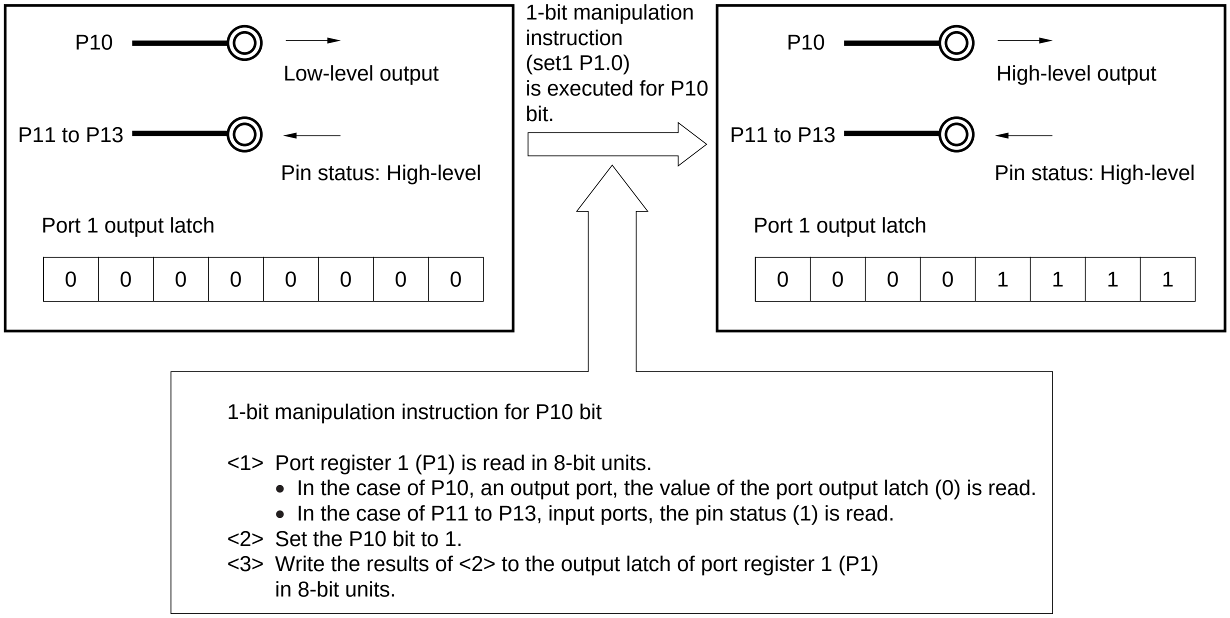
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P13, which are input ports, are read. If the pin statuses of P11 to P13 are high level at this time, the read value is 0EH.

The value is changed to 0FH by the manipulation in <2>.

0FH is written to the output latch by the manipulation in <3>.

Figure 4-26. Bit Manipulation Instruction (P10)



0	0	0	0	0	0	0	0

0	0	0	0	1	1	1	1

1-bit manipulation instruction for P10 bit

<1> Port register 1 (P1) is read in 8-bit units.
- In the case of P10, an output port, the value of the port output latch (0) is read.
- In the case of P11 to P13, input ports, the pin status (1) is read.

<2> Set the P10 bit to 1.
<3> Write the results of <2> to the output latch of port register 1 (P1) in 8-bit units.

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following 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

This circuit oscillates clocks of f_{IH} = 1, 8, and 20 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> Double-speed mode internal high-speed oscillation clock oscillator

This circuit oscillates a clock of f_{DSC} = 40 MHz (TYP.). Oscillation can be stopped by executing the STOP instruction or setting DSCON (bit 0 of DSCCTL).

An external main system clock (f_{EX} = 2 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of 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)).
Furthermore, the double-speed mode internal high-speed oscillation clock can be selected by setting SELDSC (bit 2 of the double-speed operation control register (DSCCTL)).

(2) Subsystem clock

- **XT1 clock oscillator**

This circuit oscillates a clock of f_{SUB} = 32.768 kHz by connecting a 32.768 kHz resonator to XT1 and XT2. Oscillation can be stopped by setting XTSTOP (bit 6 of CSC).

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{DSC}: Double-speed mode internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{SUB}: Subsystem clock frequency

(3) Internal low-speed oscillation clock (clock for 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

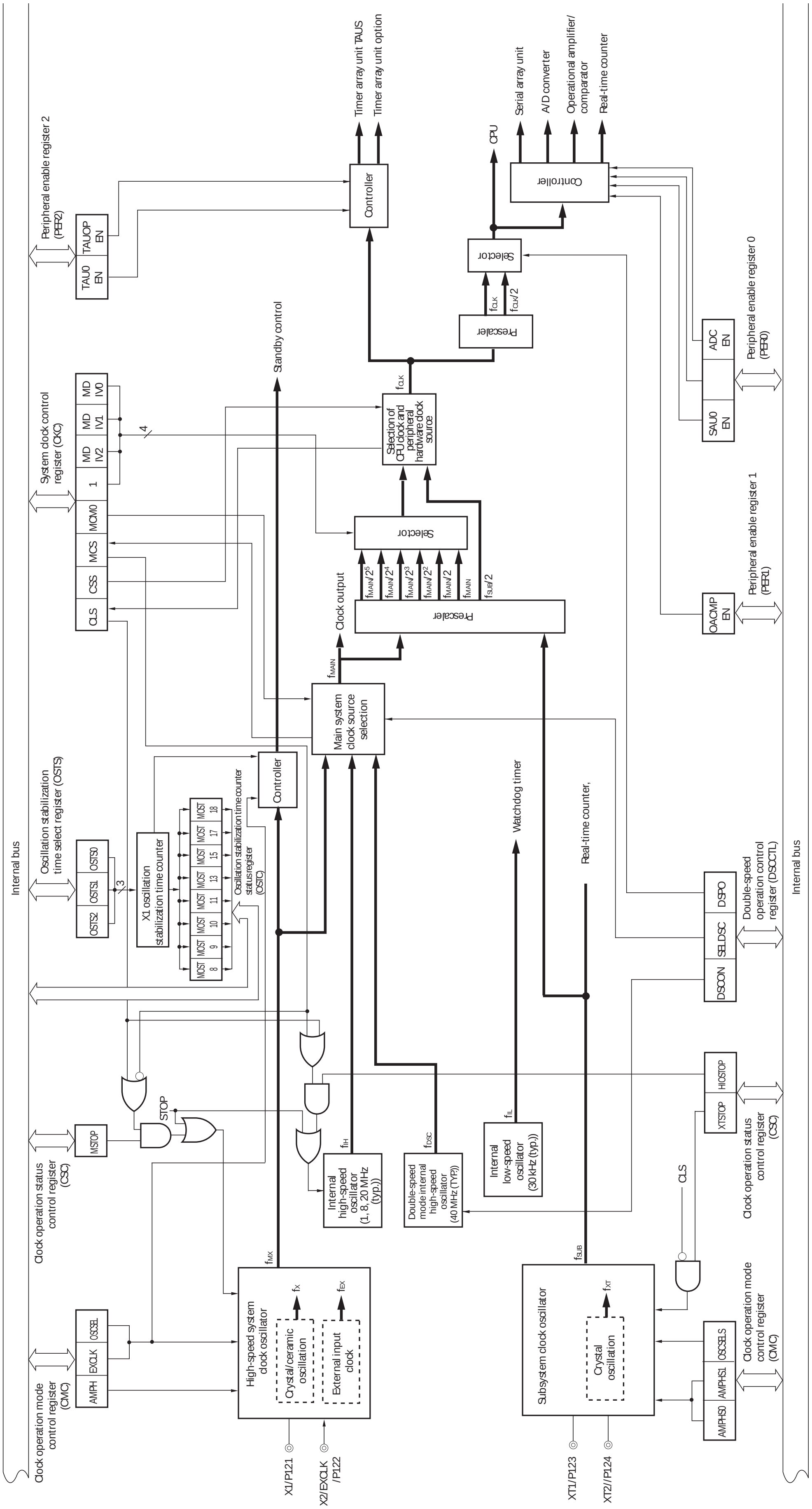
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) System clock control register (CKC) Double-speed operation control register (DSCCTL) Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2) Operation speed mode control register (OSMC)
Oscillators	X1 oscillator XT1 oscillator Internal high-speed oscillator Internal low-speed oscillator

Figure 5-1. Block Diagram of Clock Generator


Internal bus
Clock operation mode control register (CMC)
AMPH
EXCLK
OSCSEL
Clock operation status control register (CSC)
MSTOP
Oscillation stabilization time select register (OSTS)
OSTS2
OSTS1
OSTS0
3
X1 oscillation stabilization time counter
MOST 8
MOST 9
MOST 10
MOST 11
MOST 13
MOST 15
MOST 17
MOST 18
Oscillation stabilization time counter status register (OSTC)
System clock control register (CKC)
CLS
CSS
MCS
MCM0
1
MDIV2
MDIV1
MDIV0
4
Peripheral enable register 2 (PER2)
TAU0EN
TAUOPEN
STOP
Controller
Standby control
Main system clock source selection
fMAIN
Clock output
Prescaler
fMAIN/2^5
fMAIN/2^4
fMAIN/2^3
fMAIN/2^2
fMAIN/2
fMAIN
fSUB/2
Selector
Selection of CPU clock and peripheral hardware clock source
fCLK
Prescaler
fCLK
fCLK/2
Selector
Controller
Timer array unit TAUS
Timer array unit option
CPU
Controller
Serial array unit
A/D converter
Operational amplifier/comparator
Real-time counter
High-speed system clock oscillator
Crystal/ceramic oscillation
External input clock
fX
fEX
fMX
X1/P121
X2/EXCLK/P122
Internal high-speed oscillator (1, 8, 20 MHz (typ.))
fIH
Double-speed mode internal high-speed oscillator (40 MHz (TYP.))
fDSC
Internal low-speed oscillator (30 kHz (typ.))
fIL
Watchdog timer
Real-time counter,
Subsystem clock oscillator
Crystal oscillation
fXT
fSUB
XT1/P123
XT2/P124
CLS
Clock operation mode control register (CMC)
AMPHS0
AMPHS1
OSCSELS
Clock operation status control register (CSC)
XTSTOP
HIOSTOP
Double-speed operation control register (DSCCTL)
DSCON
SELDSC
DSPO
Internal bus
Peripheral enable register 1 (PER1)
OACMPEN
Peripheral enable register 0 (PER0)
SAU0EN
ADCEN

Remark f_{X}: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{DSC}: Double-speed mode 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_{XT}: XT1 clock oscillation frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Internal low-speed oscillation clock frequency

5.3 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)
- Double-speed operation control register (DSCCTL)
- Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2)
- 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, X2/EXCLK/P122, XT1/P123, and XT2/P124 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 5-2. Format of Clock Operation Mode Control Register (CMC)

Address: FFFA0H After reset: 00H R/W

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

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

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

AMPHS1	AMPHS0	Control of the XT1 oscillator dimensions
0	0	Guarantees an oscillation margin of 200 kΩ.
0	1	Guarantees an oscillation margin of 500 kΩ.
1	×	Prioritizes low-current consumption (guarantees an oscillation margin of 50 kΩ).

AMPH	Control of high-speed system clock oscillation frequency
0	2 MHz ≤ f_{MX} ≤ 10 MHz
1	10 MHz < f_{MX} ≤ 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 or XT1 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. **It is recommended to set the default value (00H) to CMC after reset release, even when the register is used at the default value, in order to prevent malfunctioning during a program loop.**

Remark f_{MX}: High-speed system clock frequency

(2) Clock operation status control register (CSC)

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

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

Reset signal generation sets this register to C0H.

Figure 5-3. Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H After reset: C0H R/W

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

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

XTSTOP	Subsystem clock operation control	
	XT1 oscillation mode	Input port mode
0	XT1 oscillator operating	–
1	XT1 oscillator stopped	

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

Cautions 1. After reset release, set the clock operation mode control register (CMC) before starting X1 oscillation as set by MSTOP or XT1 oscillation as set by XTSTOP.

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 OSC register.

Caution 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 follows.

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

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock	• CLS = 0 and MCS = 0 • CLS = 1 (CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		
Subsystem clock	• CLS = 0 (CPU and peripheral hardware clocks operate with a clock other than the subsystem clock.)	XTSTOP = 1
Internal high-speed oscillation clock	• CLS = 0 and MCS = 1 • CLS = 1 (CPU and peripheral hardware clocks operate with a clock other than the internal high-speed oscillator clock.)	HIOSTOP = 1

(3) 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 or subsystem 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 signal is generated, the STOP instruction and MSTOP (bit 7 of CSC register) = 1 clear OSTC to 00H.

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

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

Figure 5-4. 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.

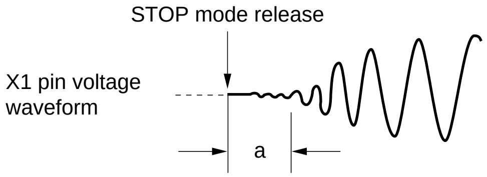
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

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

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 register after the oscillation starts.

- **If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem 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 f_X: X1 clock oscillation frequency

(4) 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 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 OSTS to 07H.

Figure 5-5. 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 register before executing the STOP instruction.

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

3. To change the setting of the OSTS register, be sure to confirm that the counting operation of the OSTC register has been completed.

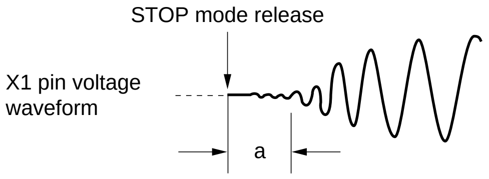
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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

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 register after the oscillation starts.

- **If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem 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

(5) System clock control register (CKC)

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

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

Reset signal generation sets this register to 09H.

Figure 5-6. 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	CLS	CSS	MCS	MCM0	1	MDIV2	MDIV1	MDIV0

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

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

CSS	MCM0	MDIV2	MDIV1	MDIV0	Selection of CPU/peripheral hardware clock (f_{CLK})
0	0	0	0	0	f_{IH}
		0	0	1	$f_{IH}/2$ (default)
		0	1	0	$f_{IH}/2^2$
		0	1	1	$f_{IH}/2^3$
		1	0	0	$f_{IH}/2^4$
		1	0	1	$f_{IH}/2^5$
0	1	0	0	0	f_{MX}
		0	0	1	$f_{MX}/2$
		0	1	0	$f_{MX}/2^2$
		0	1	1	$f_{MX}/2^3$
		1	0	0	$f_{MX}/2^4$
		1	0	1	$f_{MX}/2^5$ [Note 2]
1 [Note 3]	× [Note 3]	×	×	×	$f_{SUB}/2$
Other than above					Setting prohibited

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

2. Setting is prohibited when f_{MX} < 4 MHz.

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

Remarks 1. f_{IH}: Internal high-speed oscillation clock frequency

f_{MX}: High-speed system clock frequency

f_{SUB}: Subsystem clock frequency

2. ×: don’t care

(**Cautions 1** to **3** are listed on the next page.)

Cautions 1. Be sure to set bit 3 to 1.

2. The clock set by CSS, MCM0, and MDIV2 to MDIV0 is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the real-time counter, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral operating hardware clock.

3. If the peripheral hardware clock is used as the subsystem clock, the operations of the A/D converter is not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 28 ELECTRICAL SPECIFICATIONS (TARGET).

The fastest instruction can be executed in 1 clock of the CPU clock in the μPD79F9211. Therefore, the relationship between the CPU clock (f_{CLK}) and the minimum instruction execution time is as shown in Table 5-3.

Table 5-3. 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)			Subsystem Clock (CSS = 1)
	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 32.768 kHz Operation
f_{MAIN}	0.1 μs	0.05 μs	0.125 μs (TYP.)	–
$f_{MAIN}/2$	0.2 μs	0.1 μs	0.25 μs (TYP.) (default)	–
$f_{MAIN}/2^2$	0.4 μs	0.2 μs	0.5 μs (TYP.)	–
$f_{MAIN}/2^3$	0.8 μs	0.4 μs	1.0 μs (TYP.)	–
$f_{MAIN}/2^4$	1.6 μs	0.8 μs	2.0 μs (TYP.)	–
$f_{MAIN}/2^5$	3.2 μs	1.6 μs	4.0 μs (TYP.)	–
$f_{SUB}/2$	–		–	61 μs

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

f_{SUB}: Subsystem clock frequency

(6) Double-speed operation control register (DSCCTL)

This register controls the double-speed mode internal high-speed oscillation clock (DSC) function.

It can be used to select whether to use the double-speed mode internal high-speed oscillation clock (f_{DSC}) as a peripheral hardware clock that supports the double-speed mode.

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

Reset signal generation clears this register to 00H.

Figure 5-7. Format of Double-Speed Operation 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	DSPO	DSCON

DSCS	CPU/peripheral hardware clock (f_{CLK}) supply status flag
0	Clock through mode
1	DSC mode

SELDSC	Selection of whether to use DSC output for the CPU/peripheral hardware clock (f_{CLK})
0	Supplies the clock selected by the CKC register (clock through mode).
1	Supplies the double-speed mode internal high-speed oscillation clock (f_{DSC}) (DSC mode).

DSPO	Double-speed mode function enable/disable
0	Disables the double-speed mode function.
1	Enables the double-speed mode function.

DSCON	Double-speed mode internal oscillation clock (f_{DSC}) operation enable/disable
0	Disables operation.
1	Enables operation.

Note Bit 3 is read-only.

The bits related to the selection of the CPU/peripheral hardware clock (f_{CLK}) are shown below.

Table 5-4. Relationship Between CPU/Peripheral Hardware Clock (f_{CLK}) and Bit Settings

Main System Clock				Subsystem Clock (f_{SUB})	CPU/Peripheral Hardware Clock (f_{CLK})
High-Speed System Clock (f_{MX})	Internal High-Speed Oscillation Clock (f_{IH})	Double-Speed Mode Internal High-Speed Oscillation Clock (f_{DSC})			
2 to 20 MHz	1, 8, 20 MHz (TYP.)	40 MHz (TYP.)		32.768 kHz	
MCM0		DSCON	SELDSC	CSS	
1		×	0	0	High-speed system clock
0		×	0	0	Internal high-speed oscillation clock
× Note 1		0	0	1	Subsystem Clock
×		1	1	0	Double-speed mode internal high-speed oscillation clock Note 2

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

2. The supply of a clock of 20 MHz or more is disabled, because a clock supplied to a CPU or peripheral hardware that does not support the double-speed mode is halved ($f_{CLK}/2$) by setting DSPO (bit 1 of the DSCCTL register) to 1.

(7) Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2)

These registers are used to enable or disable use of each peripheral hardware macro. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

PER0, PER1, PER2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears theses registers to 00H.

Figure 5-8. Format of Peripheral Enable Registers (1/3)

Address: F00F0H After reset: 00H R/W

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

Address: F00F1H After reset: 00H R/W

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

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TAUOPEN	TAU0EN

RTCEN	Control of real-time counter (RTC) input clock
0	Stops input clock supply. • SFR used by the real-time counter (RTC) cannot be written (can be read). • Operation of the real-time counter (RTC) continues.
1	Supplies input clock. • SFR used by the real-time counter (RTC) can be read and written.

ADCEN	Control of A/D converter input clock
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.

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

Caution Be sure to clear the following bits to 0.

- **Bits 0, 1, 3, 4, and 6 of the PER0 register**
- **Bits 0 to 2 and 4 to 7 of the PER1 register**
- **Bits 2 to 7 of the PER2 register**

Figure 5-8. Format of Peripheral Enable Registers (2/3)

Address: F00F0H After reset: 00H R/W

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

Address: F00F1H After reset: 00H R/W

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

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TAUOPEN	TAU0EN

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

OACMPEN	Control of operational amplifier input clock
0	Stops input clock supply. • SFR used by the operational amplifier cannot be written. • The operational amplifier is in the reset status.
1	Supplies input clock. • SFR used by the operational amplifier can be read and written.

Caution Be sure to clear the following bits to 0.

- **Bits 0, 1, 3, 4, and 6 of the PER0 register**
- **Bits 0 to 2 and 4 to 7 of the PER1 register**
- **Bits 2 to 7 of the PER2 register**

Figure 5-8. Format of Peripheral Enable Registers (3/3)

Address: F00F0H After reset: 00H R/W

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

Address: F00F1H After reset: 00H R/W

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

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TAUOPEN	TAU0EN

TAUOPEN	Control of inverter control function input clock
0	Stops input clock supply. • SFR used by the inverter control function cannot be written. • The inverter control function is in the reset status.
1	Supplies input clock. • SFR used by the inverter control function can be read and written.

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

Caution Be sure to clear the following bits to 0.

- **Bits 0, 1, 3, 4, and 6 of the PER0 register**
- **Bits 0 to 2 and 4 to 7 of the PER1 register**
- **Bits 2 to 7 of the PER2 register**

(8) Operation speed mode control register (OSMC)

This register is used to control the step-up circuit of the flash memory for high-speed operation.

If the microcontroller operates at a low speed with a system clock of 10 MHz or less, the power consumption can be lowered by setting this register to the default value, 00H.

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

Reset signal generation clears this register to 00H.

Figure 5-9. 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	0	FSEL

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

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

2. 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.**

3. The CPU waits when “1” is written to the FSEL flag.

The wait time is 15 μs to 20 μs (target) when $f_{CLK} = f_{IH}$, and 30 μs to 40 μs (target) when $f_{CLK} = f_{IH}/2$.

However, counting the oscillation stabilization time of f_X can continue even while the CPU is waiting.

4. To increase f_{CLK} to 10 MHz or higher, set FSEL to “1”, then change f_{CLK} after two or more clocks have elapsed.

5. Even when set to FSEL = 1, the system clock can be operated at a frequency of 10 MHz or less.

When setting FSEL to “1”, however, do so while $V_{DD} \geq 2.25$ V.

When set to FSEL = 1, make sure that $V_{DD} \geq 2.25$ V at the following timings, even if f_{CLK} is divided.

- **When releasing f_{IH} or f_{EX} from the STOP mode selected for f_{CLK}**
- **When switching f_{CLK} from f_{SUB} to f_{MAIN}**

5.4 System Clock Oscillator

5.4.1 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.

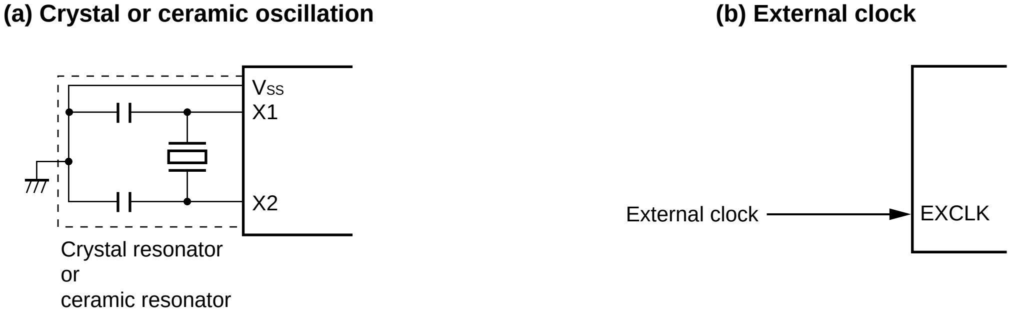
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

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 2-2 Connection of Unused Pins**.

Figure 5-10 shows an example of the external circuit of the X1 oscillator.

Figure 5-10. Example of External Circuit of X1 Oscillator



Cautions are listed on the next page.

5.4.2 XT1 oscillator

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

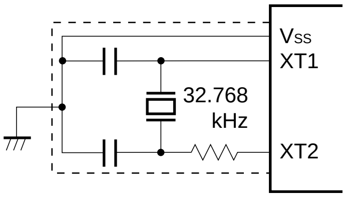
To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

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

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

Figure 5-11 shows an example of the external circuit of the XT1 oscillator.

Figure 5-11. Example of External Circuit of XT1 Oscillator (Crystal Oscillation)



Cautions are listed on the next page.

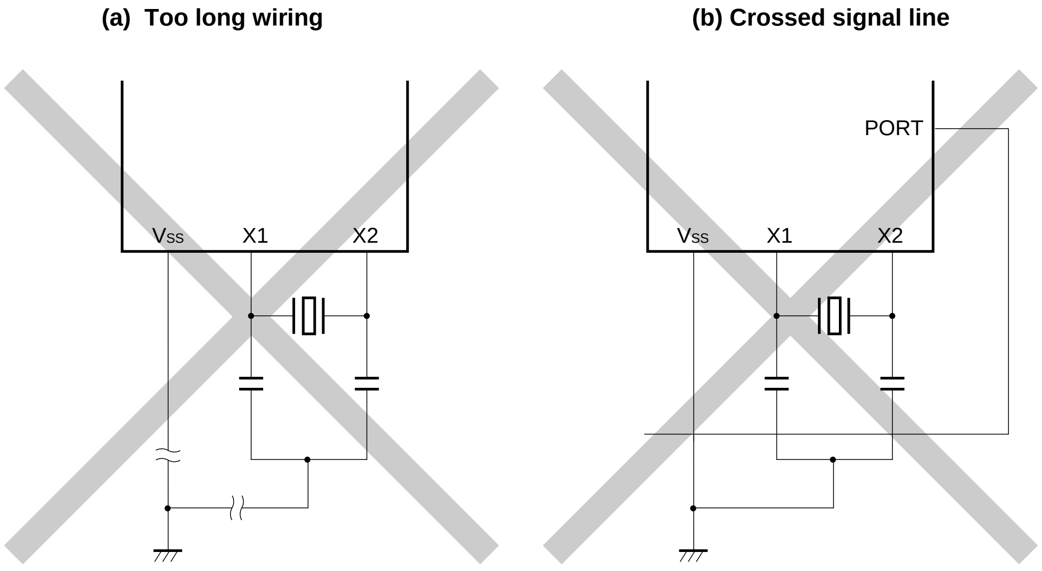
Caution 1. When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-10 and 5-11 to avoid an adverse effect from wiring capacitance.

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

Note that the XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption.

Figure 5-12 shows examples of incorrect resonator connection.

Figure 5-12. Examples of Incorrect Resonator Connection (1/2)

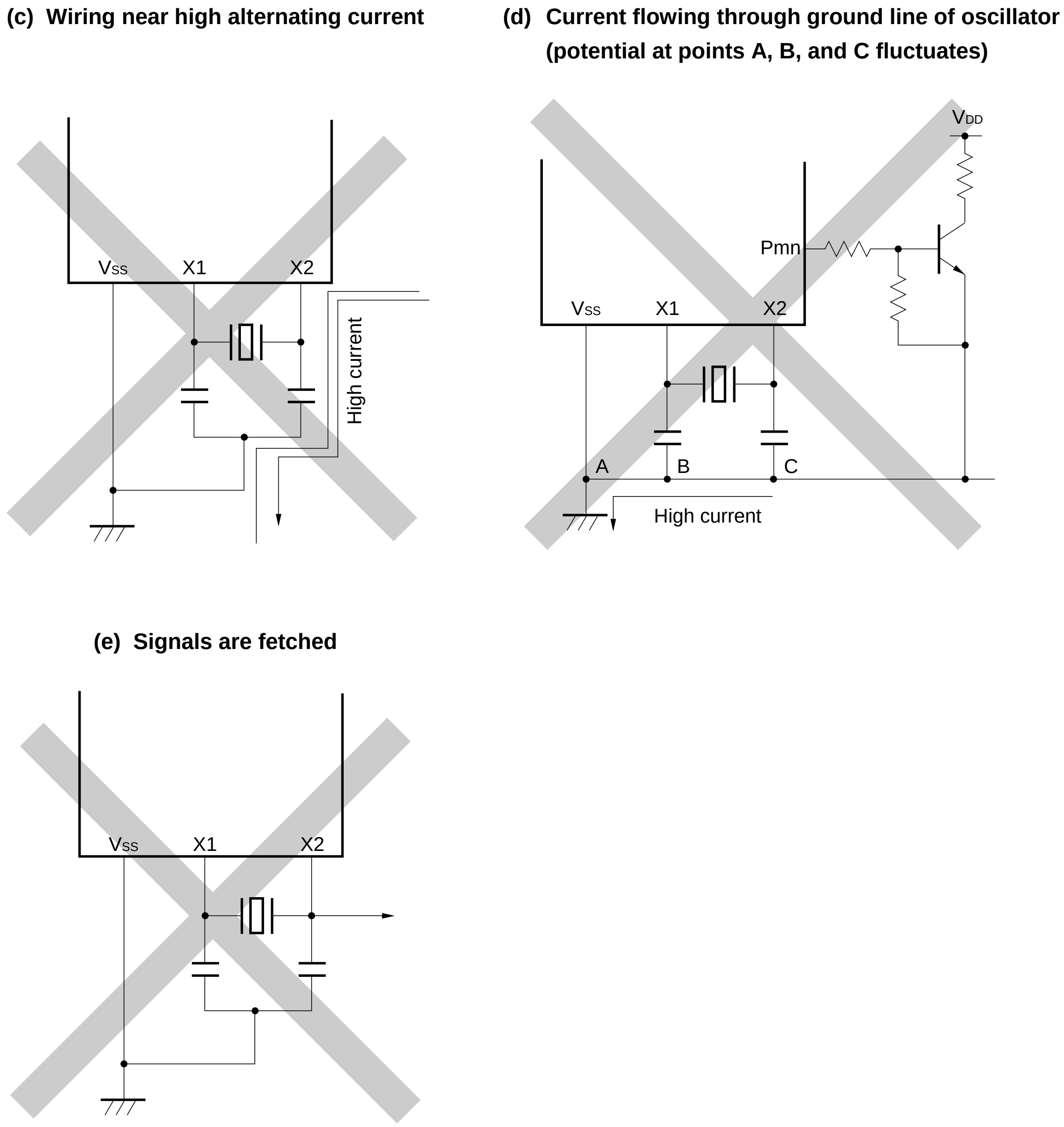


(a) Too long wiring

(b) Crossed signal line

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

Figure 5-12. Examples of Incorrect Resonator Connection (2/2)



(c) Wiring near high alternating current

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

(e) Signals are fetched

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

Caution 2. When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.

5.4.3 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the μPD79F9211 (1, 8, 20 MHz (TYP.)). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

After a reset release, the internal high-speed oscillator automatically starts oscillation.

5.4.4 Double-speed mode internal high-speed oscillator

The double-speed clock oscillator is incorporated in the μPD79F9211 (40 MHz (TYP.)). Oscillation can be controlled by bit 0 (DSCON) of the double-speed operation control register (DSCCTL).

After a reset release, the double-speed mode internal high-speed oscillator automatically starts oscillation.

5.4.5 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the μPD79F9211.

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.

5.4.6 Prescaler

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

5.5 Clock Generator Operation

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

- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - Internal high-speed oscillation clock f_{IH}
- Subsystem clock f_{SUB}
- Double-speed mode internal high-speed oscillation clock f_{DSC}
- 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 μPD79F9211, thus enabling the following.

(1) Enhancement of security function

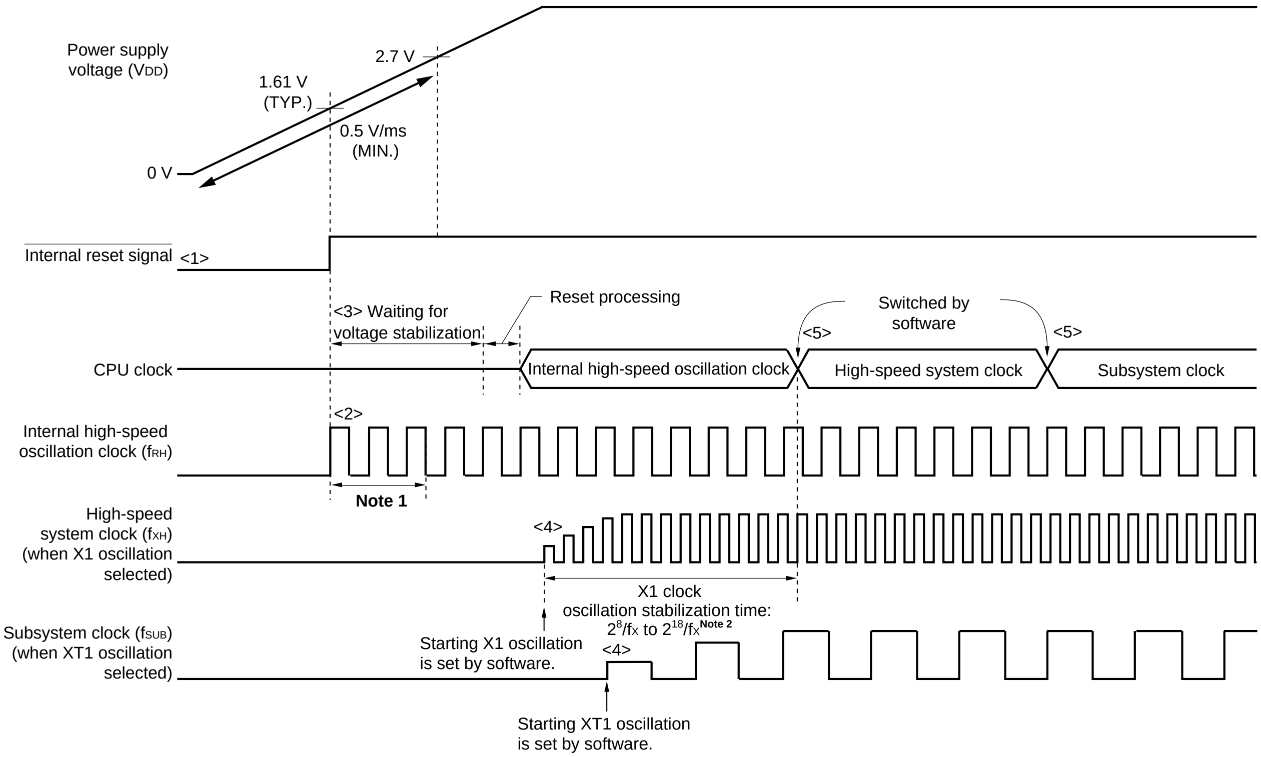
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started by the internal high-speed oscillation clock after a reset release. As a result, reset sources can be detected by software and the minimum amount of safety processing can be done during anomalies to ensure that the system terminates safely.

(2) Improvement of performance

Because the CPU can be started without waiting for the X1 clock oscillation stabilization time, the total performance can be improved.

When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-13 to Figure 5-16.

Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))


Power supply voltage (VDD)
2.7 V
1.61 V (TYP.)
0.5 V/ms (MIN.)
0 V
Internal reset signal
<1>
<3> Waiting for voltage stabilization
Reset processing
Switched by software
<5>
<5>
CPU clock
Internal high-speed oscillation clock
High-speed system clock
Subsystem clock
<2>
Internal high-speed oscillation clock (fRH)
Note 1
High-speed system clock (fXH) (when X1 oscillation selected)
<4>
X1 clock oscillation stabilization time: 2^8/fX to 2^{18}/fX Note 2
Starting X1 oscillation is set by software.
<4>
Subsystem clock (fSUB) (when XT1 oscillation selected)
Starting XT1 oscillation is set by software.

<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 automatically starts oscillation.

<3> When the power supply voltage rises with a slope of 0.5 V/ms (MIN.), the CPU starts operation on the internal high-speed oscillation clock after the reset is released and after the stabilization times for the voltage of the power supply and regulator have elapsed, and then reset processing is performed.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem clock)**.

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **5.6.1 Example of controlling high-speed system clock** and **(2)** in **5.6.3 Example of controlling subsystem clock**).

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

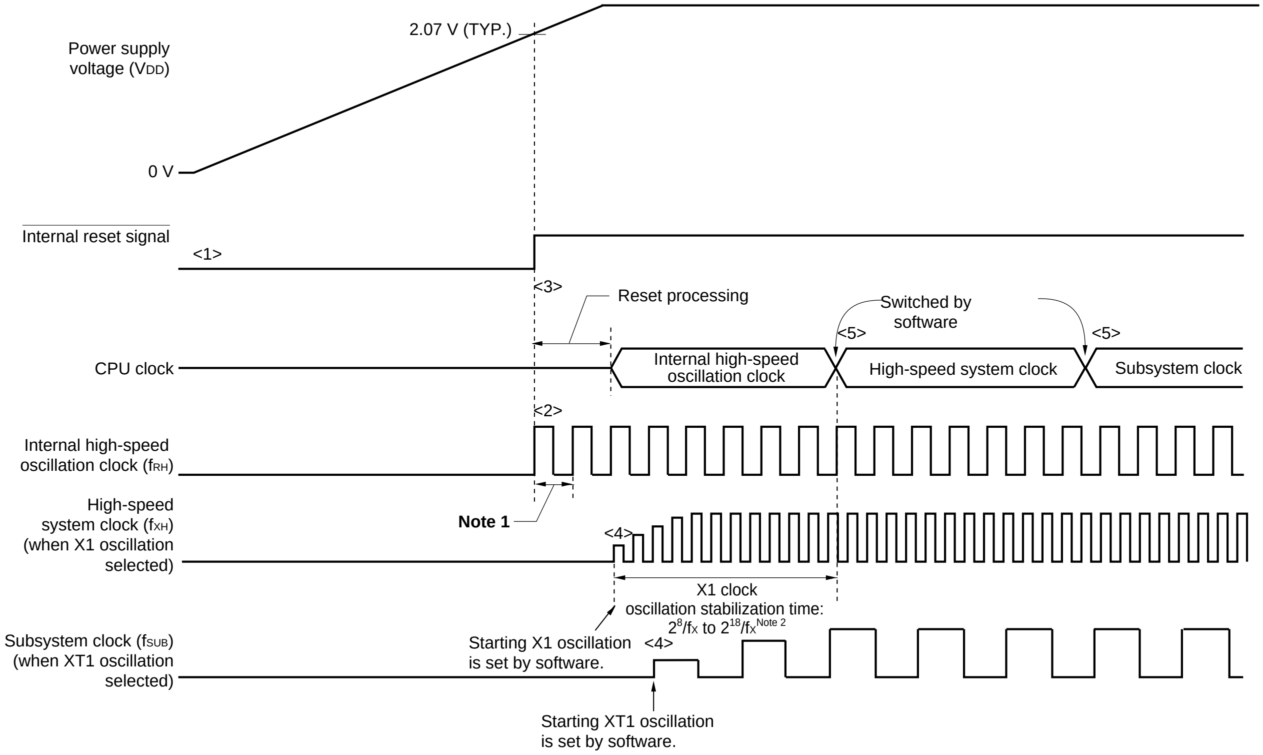
2. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

Cautions 1. If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 2.7 V, input a low level to the $\overline{\text{RESET}}$ pin from power application until the voltage reaches 2.7 V, or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 5-14). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-13 after reset release by the $\overline{\text{RESET}}$ pin.

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

Remark While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see **(4)** in **5.6.1 Example of controlling high-speed system clock**, **(3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**, and **(3)** in **5.6.3 Example of controlling subsystem clock**).

Figure 5-14. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0))



Power supply voltage (VDD)
2.07 V (TYP.)
0 V
$\overline{\text{Internal reset signal}}$
<1>
<3>
Reset processing
Switched by software
<5>
<5>
CPU clock
Internal high-speed oscillation clock
High-speed system clock
Subsystem clock
<2>
Internal high-speed oscillation clock (fRH)
High-speed system clock (fXH) (when X1 oscillation selected)
Note 1
<4>
X1 clock oscillation stabilization time: $2^8/f_X$ to $2^{18}/f_X$ Note 2
Starting X1 oscillation is set by software.
<4>
Subsystem clock (fSUB) (when XT1 oscillation selected)
Starting XT1 oscillation is set by software.

<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 2.07 V (TYP.), the reset is released and the internal high-speed oscillator 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.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem clock**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **5.6.1 Example of controlling high-speed system clock** and **(2)** in **5.6.3 Example of controlling subsystem clock**).

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

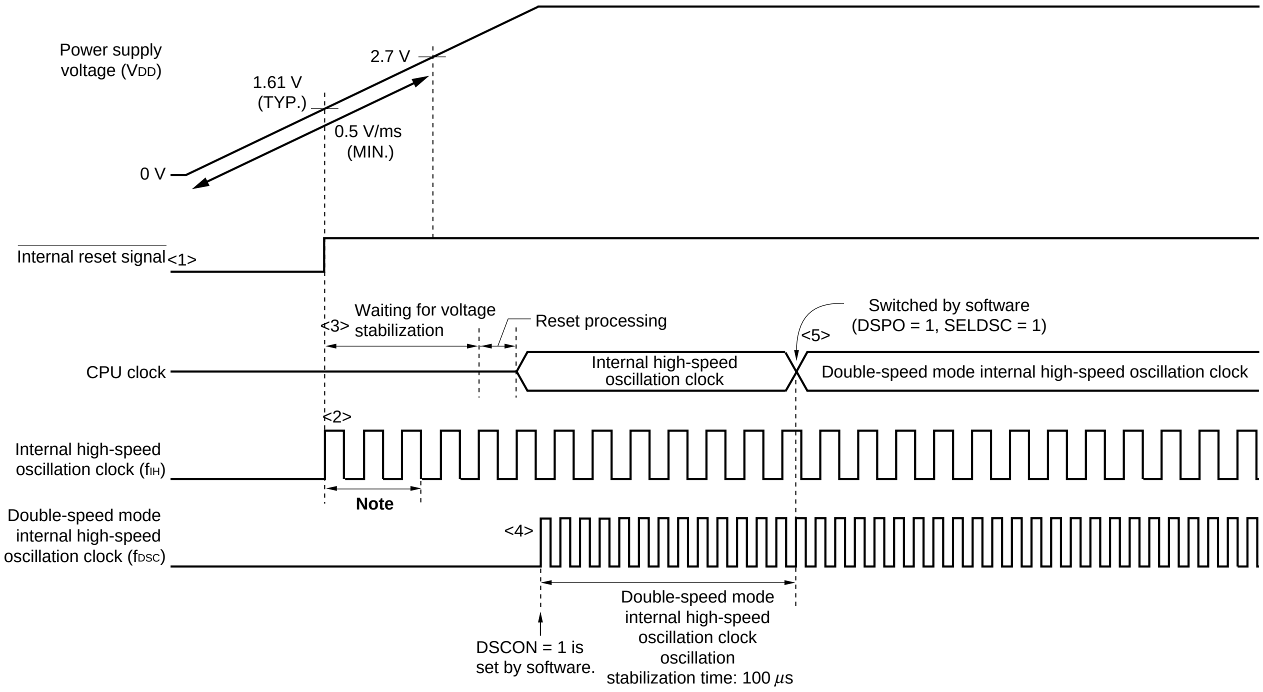
2. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

Cautions 1. A voltage oscillation stabilization time is required after the supply voltage reaches 1.61 V (TYP.). If the supply voltage rises from 1.61 V (TYP.) to 2.07 V (TYP.) within the power supply oscillation stabilization time, the power supply oscillation stabilization time is automatically generated before reset processing.

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

Remark While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see **(4)** in **5.6.1 Example of controlling high-speed system clock, (3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**, and **(3)** in **5.6.3 Example of controlling subsystem clock**).

Figure 5-15. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1) and Changing to Double-Speed Mode Internal High-Speed Oscillation Clock)


Power supply voltage (VDD)
2.7 V
1.61 V (TYP.)
0.5 V/ms (MIN.)
0 V
Internal reset signal
<1>
<3>
Waiting for voltage stabilization
Reset processing
Switched by software (DSPO = 1, SELDSC = 1)
<5>
CPU clock
Internal high-speed oscillation clock
Double-speed mode internal high-speed oscillation clock
<2>
Internal high-speed oscillation clock (fIH)
Note
Double-speed mode internal high-speed oscillation clock (fDSC)
<4>
DSCON = 1 is set by software.
Double-speed mode internal high-speed oscillation clock oscillation stabilization time: 100 μs

<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 automatically starts oscillation.

<3> When the power supply voltage rises with a slope of 0.5 V/ms (MIN.), the CPU starts operation on the internal high-speed oscillation clock after the reset is released and after the stabilization times for the voltage of the power supply and regulator have elapsed, and then reset processing is performed.

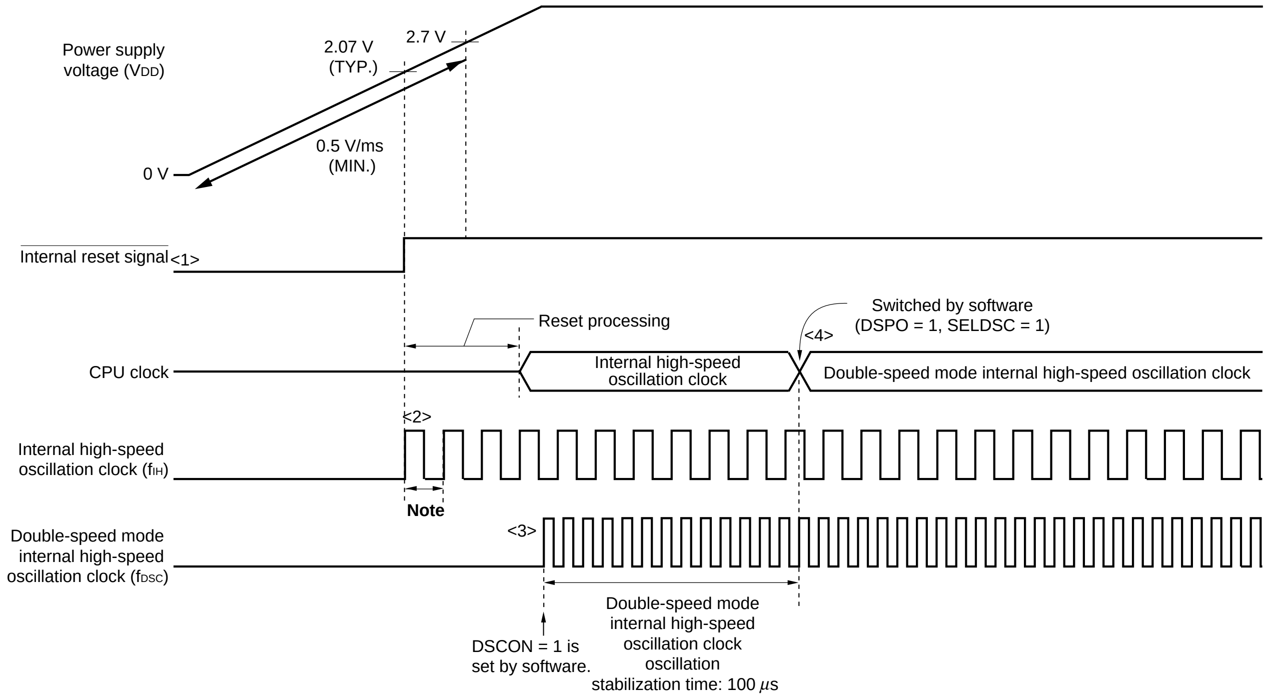
<4> Set DSCON = 1 by software.

<5> Switch the clock by setting DSPO = 1 and SELDSC = 1 by software after waiting for 100 μs.

Note The internal voltage stabilization time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

Caution If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 2.7 V, input a low level to the $\overline{\text{RESET}}$ pin from power application until the voltage reaches 2.7 V, or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 5-16). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-15 after reset release by the $\overline{\text{RESET}}$ pin.

Figure 5-16. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0) and Changing to Double-Speed Mode Internal High-Speed Oscillation Clock)


Power supply voltage (VDD)
2.07 V (TYP.)
2.7 V
0.5 V/ms (MIN.)
0 V
Internal reset signal
<1>
Reset processing
Switched by software (DSPO = 1, SELDSC = 1)
<4>
CPU clock
Internal high-speed oscillation clock
Double-speed mode internal high-speed oscillation clock
<2>
Internal high-speed oscillation clock (fIH)
Note
<3>
Double-speed mode internal high-speed oscillation clock (fDSC)
DSCON = 1 is set by software.
Double-speed mode internal high-speed oscillation clock oscillation stabilization time: 100 μs

<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 2.07 V (TYP.), the reset is released and the internal high-speed oscillator 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.

<4> Set DSCON = 1 by software.

<5> Switch the clock by setting DSPO = 1 and SELDSC = 1 by software after waiting for 100 μs.

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

Caution A voltage oscillation stabilization time is required after the supply voltage reaches 1.61 V (TYP.). If the supply voltage rises from 1.61 V (TYP.) to 2.07 V (TYP.) within the power supply oscillation stabilization time, the power supply oscillation stabilization time is automatically generated before reset processing.

5.6 Controlling Clock

5.6.1 Example of controlling high-speed system clock

The following two types of high-speed system clocks are available.

• X1 clock:	Crystal/ceramic resonator is connected to the X1 and X2 pins.
• External main system clock:	External clock is input to the EXCLK pin.

When the high-speed system clock is not used, the X1/P121 and X2/EXCLK/P122 pins can be used as input port pins.

Caution The X1/P121 and X2/EXCLK/P122 pins are in the input port mode after a reset release.

The following describes examples of setting procedures for the following cases.

(1) When oscillating X1 clock

(2) When using external main system clock

(3) When using high-speed system clock as CPU/peripheral hardware clock

(4) When stopping high-speed system clock

(1) Example of setting procedure when oscillating the X1 clock

<1> Setting P121/X1 and P122/X2/EXCLK pins and setting oscillation frequency (CMC register)

• 2 MHz ≤ f_X ≤ 10 MHz

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0	1	0	0/1	0	0/1	0/1	0

• 10 MHz < f_X ≤ 20 MHz

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0	1	0	0/1	0	0/1	0/1	1

Remarks 1. f_X: X1 clock oscillation frequency

2. For setting of the P123/XT1 and P124/XT2 pins, see **5.6.3 Example of controlling subsystem clock**.

<2> Controlling oscillation of X1 clock (CSC register)
If MSTOP is cleared to 0, the X1 oscillator starts oscillating.

<3> Waiting for the stabilization of the oscillation of X1 clock
Check the OSTC register and wait for the necessary time.
During the wait time, other software processing can be executed with the internal high-speed oscillation clock.

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.
Therefore, it is necessary to also set the value of the OSCSELS bit at the same time. For OSCSELS bit, see 5.6.3 Example of controlling subsystem clock.

2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 26 ELECTRICAL SPECIFICATIONS (TARGET)).

(2) Example of setting procedure when using the external main system clock

<1> Setting P121/X1 and P122/X2/EXCLK pins (CMC register)

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
1	1	0	0/1	0	0/1	0/1	0/1

Remark For setting of the P123/XT1 and P124/XT2 pins, see **5.6.3 (1) Example of setting procedure when oscillating the subsystem clock**.

<2> Controlling external main system clock input (CSC register)
When MSTOP is cleared to 0, the input of the external main system clock is enabled.

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.
Therefore, it is necessary to also set the value of the OSCSELS bits at the same time. For OSCSELS bits, see 5.6.3 Example of controlling subsystem clock.

2. Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 26 ELECTRICAL SPECIFICATIONS (TARGET)).

(3) Example of setting procedure when using high-speed system clock as CPU/peripheral hardware clock

<1> Setting high-speed system clock oscillation**Note**
(See **5.6.1 (1) Example of setting procedure when oscillating the X1 clock** and **(2) Example of setting procedure when using the external main system clock.**)

Note The setting of <1> is not necessary when high-speed system clock is already operating.

<2> Setting the high-speed system clock as the source clock of the CPU/peripheral hardware clock and setting the division ratio of the set clock (CKC register)

MCM0	MDIV2	MDIV1	MDIV0	Selection of CPU/Peripheral Hardware Clock (f_{CLK})
1	0	0	0	f_{MX}
	0	0	1	$f_{MX}/2$
	0	1	0	$f_{MX}/2^2$
	0	1	1	$f_{MX}/2^3$
	1	0	0	$f_{MX}/2^4$
	1	0	1	$f_{MX}/2^5$ **Note**

Note Setting is prohibited when f_{MX} < 4 MHz.

<3> If some peripheral hardware macros are not used, supply of the input clock to each hardware macro can be stopped.

(PER0 register)

RTCEN	0	ADCEN	0	0	SAU0EN	0	0

(PER1 register)

0	0	0	0	OACMPEN	0	0	0

(PER2 register)

0	0	0	0	0	0	TAUOPEN	TAU0EN

xxxEN	Input clock control
0	Stops input clock supply.
1	Supplies input clock.

Caution Be sure to clear the following bits to 0.

- **Bits 0, 1, 3, 4, and 6 of the PER0 register**
- **Bits 0 to 2 and 4 to 7 of the PER1 register**
- **Bits 2 to 7 of the PER2 register**

Remark		
	RTCEN:	Control of the real-time counter input clock
	ADCEN:	Control of the A/D converter input clock
	SAU0EN:	Control of the serial array unit input clock
	OACMPEN:	Control of the operational amplifier input clock
	TAUOPEN:	Control of the inverter control function input clock
	TAU0EN:	Control of the timer array unit TAUS input clock

(4) Example of setting procedure when stopping the high-speed system clock

The high-speed system clock can be stopped (disabling clock input if the external clock is used) in the following two ways.

- Executing the STOP instruction
- Setting MSTOP to 1

(a) To execute a STOP instruction

<1> Setting to stop peripheral hardware

Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 16 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after STOP mode is released

If the X1 clock oscillates before the STOP mode is entered, set the value of the OSTS register before executing the STOP instruction.

<3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and X1 oscillation is stopped (the input of the external clock is disabled).

(b) To stop X1 oscillation (disabling external clock input) by setting MSTOP to 1

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the high-speed system clock.

When CLS = 0 and MCS = 1, the high-speed system clock is supplied to the CPU, so change the CPU clock to the subsystem clock or internal high-speed oscillation clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Setting of X1 clock oscillation stabilization time after restart of X1 clock oscillation**Note**

Prior to setting "1" to MSTOP, set the OSTS register to a value greater than the count value to be confirmed with the OSTS register after X1 clock oscillation is restarted.

<3> Stopping the high-speed system clock (CSC register)

When MSTOP is set to 1, X1 oscillation is stopped (the input of the external clock is disabled).

Note This setting is required to resume the X1 clock oscillation when the high-speed system clock is in the X1 oscillation mode.
This setting is not required in the external clock input mode.

Caution Be sure to confirm that MCS = 0 or CLS = 1 when setting MSTOP to 1. In addition, stop peripheral hardware that is operating on the high-speed system clock.

5.6.2 Example of controlling internal high-speed oscillation clock

The following describes examples of clock setting procedures for the following cases.

(1) When restarting oscillation of the internal high-speed oscillation clock

(2) When using internal high-speed oscillation clock as CPU/peripheral hardware clock

(3) When stopping the internal high-speed oscillation clock

(1) Example of setting procedure when restarting oscillation of the internal high-speed oscillation clock Note****

<1> Setting restart of oscillation of the internal high-speed oscillation clock (CSC register)

When HIOSTOP is cleared to 0, the internal high-speed oscillation clock restarts oscillation.

Note After a reset release, the internal high-speed oscillator automatically starts oscillating and the internal high-speed oscillation clock is selected as the CPU/peripheral hardware clock.

(2) Example of setting procedure when using internal high-speed oscillation clock as CPU/peripheral hardware clock

<1> Restarting oscillation of the internal high-speed oscillation clock**Note**

(See **5.6.2 (1) Example of setting procedure when restarting internal high-speed oscillation clock**).

Note The setting of <1> is not necessary when the internal high-speed oscillation clock is operating.

<2> Setting the internal high-speed oscillation clock as the source clock of the CPU/peripheral hardware clock and setting the division ratio of the set clock (CKC register)

MCM0	MDIV2	MDIV1	MDIV0	Selection of CPU/Peripheral Hardware Clock (f_{CLK})
0	0	0	0	f_{IH}
	0	0	1	$f_{IH}/2$
	0	1	0	$f_{IH}/2^2$
	0	1	1	$f_{IH}/2^3$
	1	0	0	$f_{IH}/2^4$
	1	0	1	$f_{IH}/2^5$

Caution If switching the CPU/peripheral hardware clock from the high-speed system clock to the internal high-speed oscillation clock after restarting the internal high-speed oscillation clock, do so after 10 μs or more have elapsed.

If the switching is made immediately after the internal high-speed oscillation clock is restarted, the accuracy of the internal high-speed oscillation cannot be guaranteed for 10 μs.

(3) Example of setting procedure when stopping the internal high-speed oscillation clock

The internal high-speed oscillation clock can be stopped in the following two ways.

- Executing the STOP instruction
- Setting HIOSTOP to 1

(a) To execute a STOP instruction

<1> Setting of peripheral hardware

Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 16 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after STOP mode is released

If the X1 clock oscillates before the STOP mode is entered, set the value of the OSTS register before executing the STOP instruction.

<3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and internal high-speed oscillation clock is stopped.

(b) To stop internal high-speed oscillation clock by setting HIOSTOP to 1

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the internal high-speed oscillation clock.

When CLS = 0 and MCS = 0, the internal high-speed oscillation clock is supplied to the CPU, so change the CPU clock to the high-speed system clock or subsystem clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the internal high-speed oscillation clock (CSC register)
When HIOSTOP is set to 1, internal high-speed oscillation clock is stopped.

Caution Be sure to confirm that MCS = 1 or CLS = 1 when setting HIOSTOP to 1. In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock.

5.6.3 Example of controlling subsystem clock

The subsystem clock can be oscillated by connecting a crystal resonator to the XT1 and XT2 pins.
When the subsystem clock is not used, the XT1/P123 and XT2/P124 pins can be used as input port pins.

Caution The XT1/P123 and XT2/P124 pins are in the input port mode after a reset release.

The following describes examples of setting procedures for the following cases.
(1) When oscillating subsystem clock
(2) When using subsystem clock as CPU clock
(3) When stopping subsystem clock

Caution When the subsystem clock is used as the CPU clock, the subsystem clock is also supplied to the peripheral hardware (except the real-time counter, and watchdog timer). At this time, the operations of the A/D converter is not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 28 ELECTRICAL SPECIFICATIONS (TARGET).

(1) Example of setting procedure when oscillating the subsystem clock

<1> Setting P123/XT1 and P124/XT2 pins (CMC register)

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0/1	0/1	0	1	0	0/1	0/1	0/1

Remark For setting of the P121/X1 and P122/X2 pins, see **5.6.1 Example of controlling high-speed system clock**.

<2> Controlling oscillation of subsystem clock (CSC register)
If XTSTOP is cleared to 0, the XT1 oscillator starts oscillating.

<3> Waiting for the stabilization of the subsystem clock oscillation
Wait for the oscillation stabilization time of the subsystem clock by software, using a timer function.

Caution The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.
Therefore, it is necessary to also set the value of the EXCLK and OSCSEL bits at the same time. For EXCLK and OSCSEL bits, see 5.6.1 (1) Example of setting procedure when oscillating the X1 clock or 5.6.1 (2) Example of setting procedure when using the external main system clock.

(2) Example of setting procedure when using the subsystem clock as the CPU clock

<1> Setting subsystem clock oscillation^{Note}

(See **5.6.3 (1) Example of setting procedure when oscillating the subsystem clock**.)

Note The setting of <1> is not necessary when while the subsystem clock is operating.

<2> Setting the subsystem clock as the source clock of the CPU clock (CKC register)

CSS	Selection of CPU/Peripheral Hardware Clock (f_{CLK})
1	$f_{SUB}/2$

Caution When the subsystem clock is used as the CPU clock, the subsystem clock is also supplied to the peripheral hardware (except the real-time counter, and watchdog timer). At this time, the operations of the A/D converter is not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 28 ELECTRICAL SPECIFICATIONS (TARGET).

(3) Example of setting procedure when stopping the subsystem clock

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the subsystem clock.

When CLS = 1, the subsystem clock is supplied to the CPU, so change the CPU clock to the internal high-speed oscillation clock or high-speed system clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the subsystem clock (CSC register)

When XTSTOP is set to 1, subsystem clock is stopped.

Cautions 1. Be sure to confirm that CLS = 0 when setting XTSTOP to 1. In addition, stop the peripheral hardware if it is operating on the subsystem clock.

2. The subsystem clock oscillation cannot be stopped using the STOP instruction.

5.6.4 Example of controlling internal low-speed oscillation clock

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

The internal low-speed oscillator automatically starts oscillation after a reset release, 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.

(1) Example of setting procedure when stopping the internal low-speed oscillation clock

The internal low-speed oscillation clock can be stopped in the following two ways.

- Stop the watchdog timer in the HALT/STOP mode by the option byte (bit 0 (WDSTBYON) of 000C0H = 0), and execute the HALT or STOP instruction.
- Stop the watchdog timer by the option byte (bit 4 (WDTON) of 000C0H = 0).

(2) Example of setting procedure when restarting oscillation of the internal low-speed oscillation clock

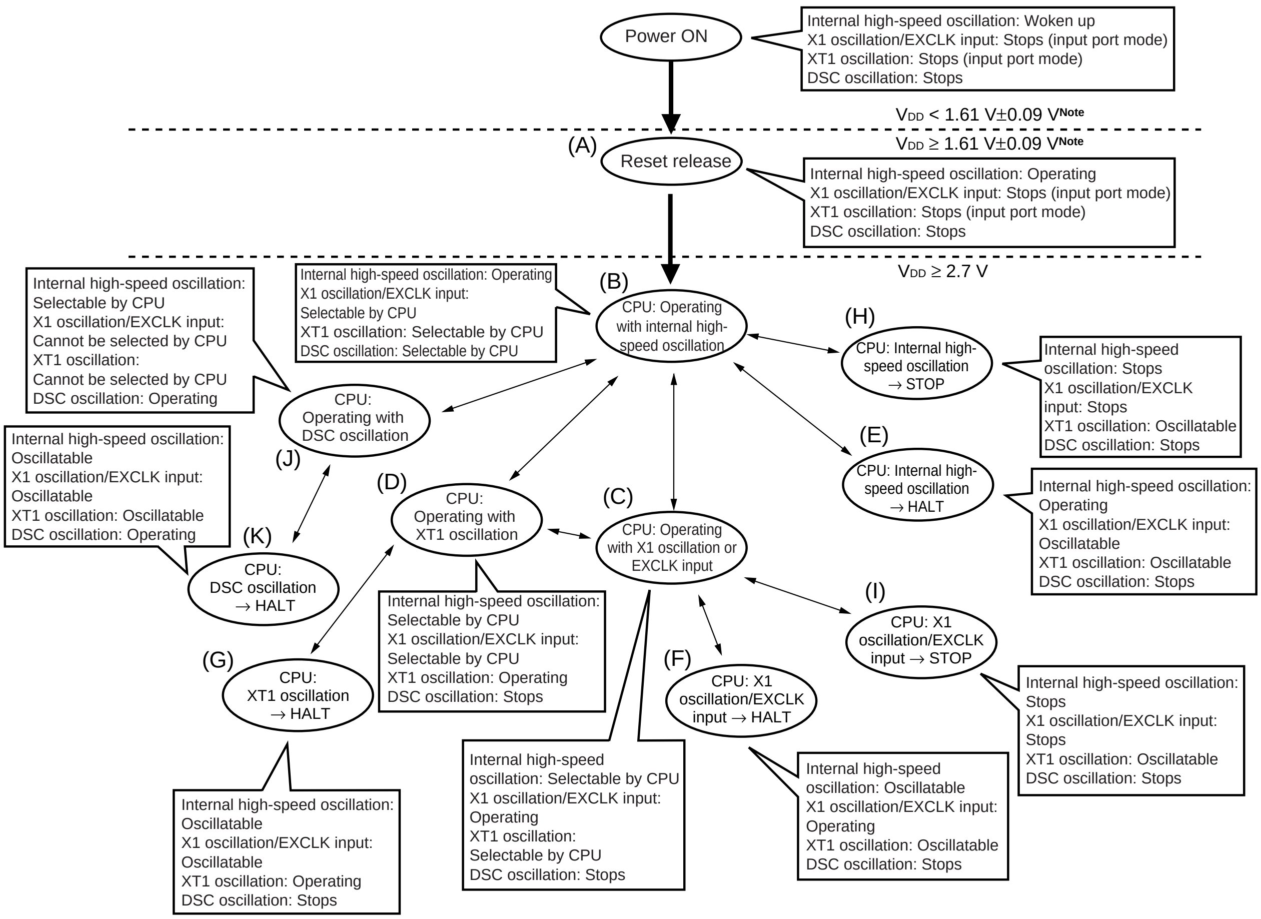
The internal low-speed oscillation clock can be restarted as follows.

- Release the HALT or STOP mode
  (only when the watchdog timer is stopped in the HALT/STOP mode by the option byte (bit 0 (WDSTBYON) of 000C0H) = 0) and when the watchdog timer is stopped as a result of execution of the HALT or STOP instruction).

5.6.5 CPU clock status transition diagram

Figure 5-17 shows the CPU clock status transition diagram of this product.

Figure 5-17. CPU Clock Status Transition Diagram


Power ON
Internal high-speed oscillation: Woken up
X1 oscillation/EXCLK input: Stops (input port mode)
XT1 oscillation: Stops (input port mode)
DSC oscillation: Stops
VDD < 1.61 V±0.09 VNote
VDD ≥ 1.61 V±0.09 VNote
(A)
Reset release
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Stops (input port mode)
XT1 oscillation: Stops (input port mode)
DSC oscillation: Stops
VDD ≥ 2.7 V
Internal high-speed oscillation: Selectable by CPU
X1 oscillation/EXCLK input: Cannot be selected by CPU
XT1 oscillation: Cannot be selected by CPU
DSC oscillation: Operating
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Selectable by CPU
XT1 oscillation: Selectable by CPU
DSC oscillation: Selectable by CPU
(B)
CPU: Operating with internal high-speed oscillation
(H)
CPU: Internal high-speed oscillation → STOP
Internal high-speed oscillation: Stops
X1 oscillation/EXCLK input: Stops
XT1 oscillation: Oscillatable
DSC oscillation: Stops
CPU: Operating with DSC oscillation
(J)
Internal high-speed oscillation: Oscillatable
X1 oscillation/EXCLK input: Oscillatable
XT1 oscillation: Oscillatable
DSC oscillation: Operating
(E)
CPU: Internal high-speed oscillation → HALT
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Oscillatable
XT1 oscillation: Oscillatable
DSC oscillation: Stops
(D)
CPU: Operating with XT1 oscillation
(C)
CPU: Operating with X1 oscillation or EXCLK input
(K)
CPU: DSC oscillation → HALT
Internal high-speed oscillation: Selectable by CPU
X1 oscillation/EXCLK input: Selectable by CPU
XT1 oscillation: Operating
DSC oscillation: Stops
(I)
CPU: X1 oscillation/EXCLK input → STOP
(G)
CPU: XT1 oscillation → HALT
(F)
CPU: X1 oscillation/EXCLK input → HALT
Internal high-speed oscillation: Stops
X1 oscillation/EXCLK input: Stops
XT1 oscillation: Oscillatable
DSC oscillation: Stops
Internal high-speed oscillation: Selectable by CPU
X1 oscillation/EXCLK input: Operating
XT1 oscillation: Selectable by CPU
DSC oscillation: Stops
Internal high-speed oscillation: Oscillatable
X1 oscillation/EXCLK input: Operating
XT1 oscillation: Oscillatable
DSC oscillation: Stops
Internal high-speed oscillation: Oscillatable
X1 oscillation/EXCLK input: Oscillatable
XT1 oscillation: Operating
DSC oscillation: Stops

Note Preliminary value and subject to change.

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**Note**.

After the reset operation, the status will shift to (B) in the above figure.

2. DSC: Double-speed mode internal high-speed oscillation clock

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

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

(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 ≤ f_X ≤ 10 MHz)	0	1	0	0	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < f_X ≤ 20 MHz)	0	1	1	0	1[Note 2]	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	0/1	0	0/1	Must not be checked	1

Notes 1. The CMC and OSMC registers 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 f_X > 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 (TARGET)).

(3) CPU operating with subsystem 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	CMC Register[Note]	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	OSCSELS	XTSTOP		CSS
(A) → (B) → (D)	1	0	Necessary	1

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

Remark (A) to (K) in Table 5-5 correspond to (A) to (K) in Figure 5-17.

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

(4) CPU operating with double-speed mode internal high-speed oscillation clock (J) 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	DSCCTL Register
	DSCON		DSPO	SELDSC
(A) → (B) → (J)	1	Necessary (100 μs)	1	1

(5) 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	0/1	**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 CMC and OSMC registers 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 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 f_X > 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 (TARGET)).

Remark (A) to (K) in Table 5-5 correspond to (A) to (K) in Figure 5-17.

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

(6) CPU clock changing from internal high-speed oscillation clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) ⟶

Setting Flag of SFR Register / Status Transition	CMC Register[Note]	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	OSCSELS	XTSTOP		CSS
(B) → (D)	1	0	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

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

(7) CPU clock changing from internal high-speed oscillation clock (B) to double-speed mode internal high-speed oscillation clock (J)

(Setting sequence of SFR registers) ⟶

Setting Flag of SFR Register / Status Transition	DSCCTL Register[Note]	Waiting for Oscillation Stabilization	DSCCTL Register	DSCCTL Register
	DSCON		DSPO	SELDSC
(B) → (J)	1	Necessary (100 μs)	1	1

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

(8) 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

Remark (A) to (K) in Table 5-5 correspond to (A) to (K) in Figure 5-17.

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

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register	CMC Register[Note]	CSC Register	Waiting for Oscillation Stabilization	CKC Register
Status Transition	OSCSELS	XTSTOP		CSS
(C) → (D)	1	0	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

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

(10) CPU clock changing from subsystem clock (D) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register	CSC Register	CKC Register	
Status Transition	HIOSTOP	MCM0	CSS
(D) → (B)	0	0	0

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

Unnecessary if this register is already set

Remark (A) to (K) in Table 5-5 correspond to (A) to (K) in Figure 5-17.

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

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note 1]			OSTS Register	CSC Register	OSMC Register	OSTC Register	CKC Register	
	EXCLK	OSCSEL	AMPH		MSTOP	FSEL		MCM0	CSS
(D) → (C) (X1 clock: 2 MHz ≤ f_X ≤ 10 MHz)	0	1	0	Note 2	0	0	Must be checked	1	0
(D) → (C) (X1 clock: 10 MHz < f_X ≤ 20 MHz)	0	1	1	Note 2	0	1 [Note 3]	Must be checked	1	0
(D) → (C) (external main clock)	1	1	0/1	Note 2	0	0/1	Must not be checked	1	0

Unnecessary if this register is already set

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

Unnecessary if these registers are already set

Notes 1. The CMC and OSMC registers 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 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 f_X > 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 (TARGET)).

(12) CPU clock changing from double-speed mode internal high-speed oscillation clock (J) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	CKC Register	DSCCTL Register		
	HIOSTOP	MCM0	SELDSC	DSPO	DSCON
(J) → (B)	0	0	0	0	0

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

Unnecessary if this register is already set

Remark (A) to (K) in Table 5-5 correspond to (A) to (K) in Figure 5-17.

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

(13)
- **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 subsystem clock (D)**
- **HALT mode (K) set while CPU is operating with double-speed mode internal high-speed oscillation clock (J)**

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

(14)
- **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) →

<table>
<tr><th colspan="2">Status Transition</th><th colspan="3">Setting</th></tr>
<tr><td rowspan="2">(B) → (H)</td><td>In X1 stop</td><td rowspan="3">Stopping peripheral functions that cannot operate in STOP mode</td><td>–</td><td rowspan="3">Executing STOP instruction</td></tr>
<tr><td>In X1 oscillation</td><td rowspan="2">Sets the OSTS register</td></tr>
<tr><td colspan="2">(C) → (I)</td></tr>
</table>

Remark (A) to (K) in Table 5-5 correspond to (A) to (K) in Figure 5-17.

5.6.6 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 5-6. Changing CPU Clock (1/2)

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	
	Subsystem clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	
	Double-speed mode internal high-speed oscillation clock	Stabilization of DSC oscillation • After elapse of oscillation stabilization time (100 μs) after setting to DSCON = 1 • DSPO = 1, SELDSC = 1	
X1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • RSTOP = 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.)	–
	Subsystem clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1).
	Double-speed mode 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 • RSTOP = 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.)	–
	Subsystem clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1).
	Double-speed mode internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–

Table 5-6. Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Subsystem clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator and selection of internal high-speed oscillation clock as main system clock • HIOSTOP = 0, MCS = 0	XT1 oscillation can be stopped (XTSTOP = 1)
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	Double-speed mode internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
Double-speed mode internal high-speed oscillation clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator and selection of internal high-speed oscillation clock as main system clock • HIOSTOP = 0, MCS = 0 • SELDSC = 0, DSPO = 0	Double-speed mode 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.	–
	Subsystem clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–

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

By setting bits 0 to 2, 4, and 6 (MDIV0 to MDIV2, MCM0, CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), 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 5-7 to Table 5-10).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 7 (CLS) of CKC. 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 5-7. Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Type
f_{IH}	↔	f_{MX}	Type 2 (see Table 5-9)
f_{MAIN}	↔	f_{SUB}	Type 3 (see Table 5-10)
f_{MAIN}	↔ (changing the division ratio)	f_{MAIN}	Type 1 (see Table 5-8)

Table 5-8. Maximum Number of Clocks Required in Type 1

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 5-9. Maximum Number of Clocks Required in Type 2

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})	$f_{MX}>f_{IH}$		1 + f_{MX}/f_{IH} clock
	$f_{MX}<f_{IH}$		$2f_{IH}/f_{MX}$ clock
1 (f_{MAIN} = f_{MX})	$f_{MX}>f_{IH}$	$2f_{MX}/f_{IH}$ clock	
	$f_{MX}<f_{IH}$	1 + f_{MX}/f_{IH} clock	

(**Remarks 1 and 2** are listed on the next page.)

Table 5-10. Maximum Number of Clocks Required in Type 3

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

Remarks 1. The number of clocks listed in Table 5-8 to Table 5-10 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Table 5-8 to Table 5-10 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

5.6.8 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 5-11. 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 or CLS = 1 (The CPU is operating on a clock other than the internal high-speed oscillation clock.)	HIOSTOP = 1
X1 clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		
Subsystem clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock.)	XTSTOP = 1
Double-speed mode internal high-speed oscillation clock	SELDSC = 0, DSPO = 0 (The main system clock is operating on a clock other than the double-speed mode internal high-speed oscillation clock.)	DSCON = 0

CHAPTER 6 TIMER ARRAY UNIT TAUS

Timer array unit TAUS has twelve 16-bit timers per unit. 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.

Single-operation Function (Explained in This Chapter)	Combination-operation Function (Explained in This Chapter)	Inverter Control Function (Refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**)
• Interval timer • Square wave output • External event counter • Divider function • Input pulse interval measurement • Measurement of high-/low-level width of input signal	• PWM output • One-shot pulse output • Multiple PWM output	• Real-time output function (type 1) • Real-time output function (type 2) • 6-phase PWM output function • Triangular wave PWM output function • Triangular wave PWM output function with dead time • 6-phase triangular wave PWM output function with dead time • Interrupt signal thinning function • A/D conversion trigger output function (type 1) • A/D conversion trigger output function (type 2)

Channel 7 can be used to realize LIN-bus reception processing in combination with UART0 of the serial array unit.

6.1 Functions of Timer Array Unit TAUS

Timer array unit TAUS has the following functions.

6.1.1 Functions of each channel when it operates independently

Single-operation functions are those functions that can be used for any channel regardless of the operation mode of the other channel (for details, refer to **6.6.1 Overview of single-operation function and combination-operation function**).

(1) Interval timer

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

(2) Square wave output

A toggle operation is performed each time INTTMn is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOn, SLTO).

(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 (TIn, SLTI) has reached a specific value.

(4) Divider function

A clock input from a timer input pin (TIm) is divided and output from an output pin (TOm).

(5) Input pulse interval measurement

Counting is started by the valid edge of a pulse signal input to a timer input pin (TIn, SLTI). 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.

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

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

Remarks 1. n = 00 to 11
In the case of timer input pin (TIn): n = 02-07, 09-11
In the case of timer output pin (TOn): n = 02-07, 10, 11
2. m = 10, 11

6.1.2 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 (for details, refer to **6.6.1 Overview of single-operation function and combination-operation function**).

(1) PWM (Pulse Width Modulator) output

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

(2) One-shot pulse output

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

(3) Multiple PWM (Pulse Width Modulator) output

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

6.1.3 LIN-bus supporting function (channel 7 only)

(1) Detection of wakeup signal

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD0) of UART0 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 (RxD0) of UART0 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 (RxD0) of UART0 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

6.2 Configuration of Timer Array Unit TAUS

Timer array unit TAUS includes the following hardware.

Table 6-1. Configuration of Timer Array Unit TAUS

Item	Configuration
Timer/counter	Timer counter register n (TCRn)
Register	Timer data register n (TDRn)
Timer input	TI02 to TI07, TI09, TI10, TI11, SLTI pins, RxD0 pin (for LIN-bus)
Timer output	TO02 to TO07, TO10, TO11, SLTO pins, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 2 (PER2) • Timer clock select register 0 (TPS0) • Timer channel enable status register 0 (TE0) • Timer channel start register 0 (TS0) • Timer channel stop register 0 (TT0) • Timer input select register 0 (TIS0) • Timer output enable register 0 (TOE0) • Timer output register 0 (TO0) • Timer output level register 0 (TOL0) • Timer output mode register 0 (TOM0) • Timer triangle wave output mode register 0 (TOT0)Note • Timer dead time output enable register 0 (TDE0)Note • Timer real-time output register 0 (TRO0)Note • Timer real-time output enable register 0 (TRE0)Note • Timer real-time control register 0 (TRC0)Note • Timer modulation output enable register 0 (TME0)Note • TAU option mode register (OPMR)Note • TAU option status register (OPSR)Note • TAU option Hi-Z start trigger register (OPHS)Note • TAU option Hi-Z stop trigger register (OPHT)Note
	<Registers of each channel> • Timer mode register n (TMRn) • Timer status register n (TSRn) • Input switch control register (ISC) • Noise filter enable registers 1, 2 (NFEN1, NFEN2) • Port mode registers 1, 3, 5, 7 (PM1, PM3, PM5, PM7) • Port registers 1, 3, 5, 7 (P1, P3, P5, P7)

Note. These registers are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Remark n: Channel number (n = 00 to 11)

- Timer I/O pin configuration

The P52/SLTI/SLTO pin can be assigned to the timer I/Os of channels 0, 1, and 8 to 11 by setting the input switching control register (ISC). (For details of the input switching control register (ISC), see 6.3 (23) Input switching control register (ISC).)

The following I/O pins can be selected for channels 0, 1, and 8 to 11.

Table 6-2. I/O Pins That Can Be Selected for Channels 0, 1, and 8 to 11

Channel for Which I/O Pin Can Be Selected	Input Pin	Output Pin
Channel 0	P52/SLTI/SLTO pin	P52/SLTI/SLTO pin
Channel 1	P52/SLTI/SLTO pin	P52/SLTI/SLTO pin
Channel 8	P52/SLTI/SLTO pin	P52/SLTI/SLTO pin
Channel 9	• P31/TI09 pin • P52/SLTI/SLTO pin	P52/SLTI/SLTO pin
Channel 10	• P74/TI10 pin • P52/SLTI/SLTO pin	• P73/TO10 pin • P52/SLTI/SLTO pin
Channel 11	• P75/TI11 pin • P52/SLTI/SLTO pin	• P30/TO11 pin • P52/SLTI/SLTO pin

Caution Hereinafter, timer I/O pins are described as TIn and TOn (n = xx), which also includes the selection of the SLTI and SLTO pins.

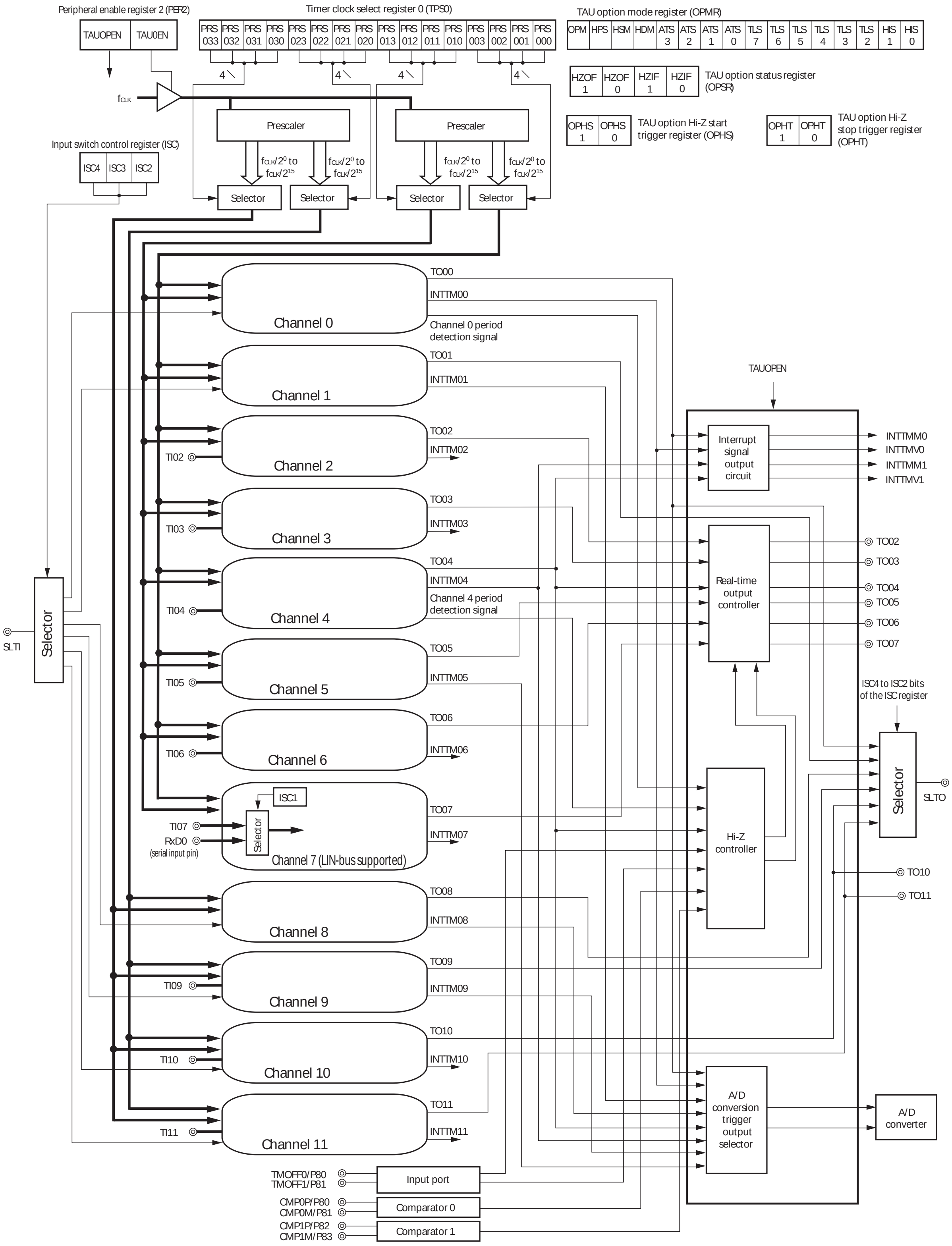
Remarks 1. When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

2. Only one of the above-mentioned channels can be assigned as the timer I/O pin for the P52/SLTI/SLTO pin.

3. The SLTI and SLTO pins cannot be selected as timer I/Os for channels other than those mentioned above (channels 2 to 7).

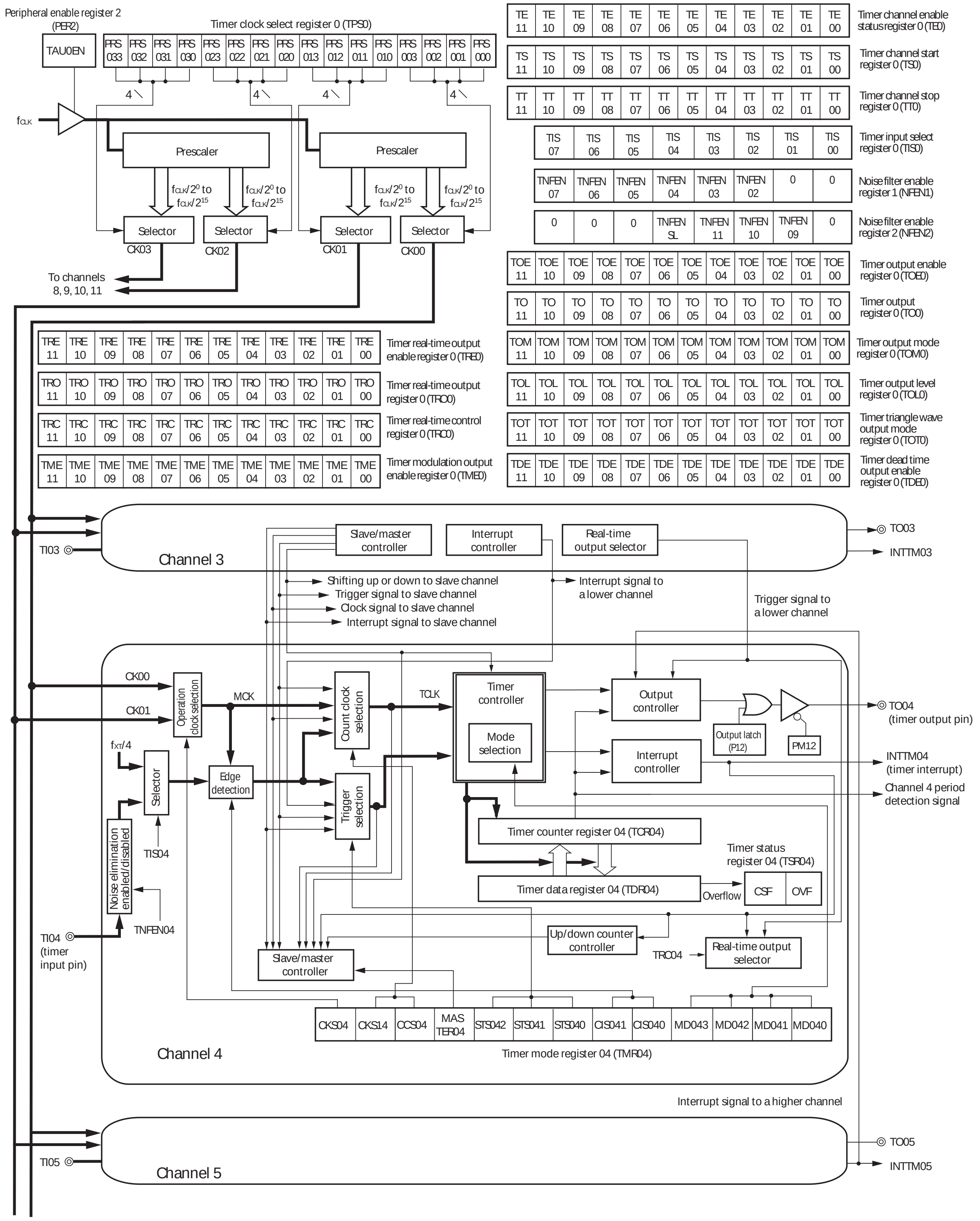
Figure 6-1 shows the block diagram.

Figure 6-1. Entire Configuration of Timer Array Unit TAUS



Remark The configuration diagram in Figure 6-1 also includes the registers and pins used with the inverter control function. For details of the inverter control function, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Figure 6-2. Block Diagram of Each Channel (Example: Channel 4)


Peripheral enable register 2 (PER2)
TAU0EN
Timer clock select register 0 (TPS0)
PRS 033 | PRS 032 | PRS 031 | PRS 030 | PRS 023 | PRS 022 | PRS 021 | PRS 020 | PRS 013 | PRS 012 | PRS 011 | PRS 010 | PRS 003 | PRS 002 | PRS 001 | PRS 000
4
fCLK
Prescaler
fCLK/2^0 to fCLK/2^15
Selector
CK03
CK02
CK01
CK00
To channels 8, 9, 10, 11
TE 11 | TE 10 | TE 09 | TE 08 | TE 07 | TE 06 | TE 05 | TE 04 | TE 03 | TE 02 | TE 01 | TE 00
Timer channel enable status register 0 (TE0)
TS 11 | TS 10 | TS 09 | TS 08 | TS 07 | TS 06 | TS 05 | TS 04 | TS 03 | TS 02 | TS 01 | TS 00
Timer channel start register 0 (TS0)
TT 11 | TT 10 | TT 09 | TT 08 | TT 07 | TT 06 | TT 05 | TT 04 | TT 03 | TT 02 | TT 01 | TT 00
Timer channel stop register 0 (TT0)
TIS 07 | TIS 06 | TIS 05 | TIS 04 | TIS 03 | TIS 02 | TIS 01 | TIS 00
Timer input select register 0 (TIS0)
TNFEN 07 | TNFEN 06 | TNFEN 05 | TNFEN 04 | TNFEN 03 | TNFEN 02 | 0 | 0
Noise filter enable register 1 (NFEN1)
0 | 0 | 0 | TNFEN SL | TNFEN 11 | TNFEN 10 | TNFEN 09 | 0
Noise filter enable register 2 (NFEN2)
TOE 11 | TOE 10 | TOE 09 | TOE 08 | TOE 07 | TOE 06 | TOE 05 | TOE 04 | TOE 03 | TOE 02 | TOE 01 | TOE 00
Timer output enable register 0 (TOE0)
TO 11 | TO 10 | TO 09 | TO 08 | TO 07 | TO 06 | TO 05 | TO 04 | TO 03 | TO 02 | TO 01 | TO 00
Timer output register 0 (TO0)
TRE 11 | TRE 10 | TRE 09 | TRE 08 | TRE 07 | TRE 06 | TRE 05 | TRE 04 | TRE 03 | TRE 02 | TRE 01 | TRE 00
Timer real-time output enable register 0 (TRE0)
TOM 11 | TOM 10 | TOM 09 | TOM 08 | TOM 07 | TOM 06 | TOM 05 | TOM 04 | TOM 03 | TOM 02 | TOM 01 | TOM 00
Timer output mode register 0 (TOM0)
TRO 11 | TRO 10 | TRO 09 | TRO 08 | TRO 07 | TRO 06 | TRO 05 | TRO 04 | TRO 03 | TRO 02 | TRO 01 | TRO 00
Timer real-time output register 0 (TRO0)
TOL 11 | TOL 10 | TOL 09 | TOL 08 | TOL 07 | TOL 06 | TOL 05 | TOL 04 | TOL 03 | TOL 02 | TOL 01 | TOL 00
Timer output level register 0 (TOL0)
TRC 11 | TRC 10 | TRC 09 | TRC 08 | TRC 07 | TRC 06 | TRC 05 | TRC 04 | TRC 03 | TRC 02 | TRC 01 | TRC 00
Timer real-time control register 0 (TRC0)
TOT 11 | TOT 10 | TOT 09 | TOT 08 | TOT 07 | TOT 06 | TOT 05 | TOT 04 | TOT 03 | TOT 02 | TOT 01 | TOT 00
Timer triangle wave output mode register 0 (TOT0)
TME 11 | TME 10 | TME 09 | TME 08 | TME 07 | TME 06 | TME 05 | TME 04 | TME 03 | TME 02 | TME 01 | TME 00
Timer modulation output enable register 0 (TME0)
TDE 11 | TDE 10 | TDE 09 | TDE 08 | TDE 07 | TDE 06 | TDE 05 | TDE 04 | TDE 03 | TDE 02 | TDE 01 | TDE 00
Timer dead time output enable register 0 (TDE0)
TI03
Channel 3
Slave/master controller
Interrupt controller
Real-time output selector
TO03
INTTM03
Shifting up or down to slave channel
Trigger signal to slave channel
Clock signal to slave channel
Interrupt signal to slave channel
Interrupt signal to a lower channel
Trigger signal to a lower channel
CK00
CK01
Operation clock selection
MCK
Count clock selection
TCLK
Timer controller
Mode selection
Output controller
Output latch (P12)
PM12
TO04 (timer output pin)
Interrupt controller
INTTM04 (timer interrupt)
Channel 4 period detection signal
fXT/4
Selector
Edge detection
Trigger selection
TIS04
Noise elimination enabled/disabled
TNFEN04
Timer counter register 04 (TCR04)
Timer status register 04 (TSR04)
Timer data register 04 (TDR04)
Overflow
CSF
OVF
TI04 (timer input pin)
Up/down counter controller
TRC04
Real-time output selector
Slave/master controller
CKS04 | CKS14 | CCS04 | MAS TER04 | STS042 | STS041 | STS040 | CIS041 | CIS040 | MD043 | MD042 | MD041 | MD040
Channel 4
Timer mode register 04 (TMR04)
Interrupt signal to a higher channel
TI05
Channel 5
TO05
INTTM05

Remark The block diagram in Figure 6-2 also includes the registers and pins used with the inverter control function. For details of the inverter control function, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

(1) Timer counter register n (TCRn)

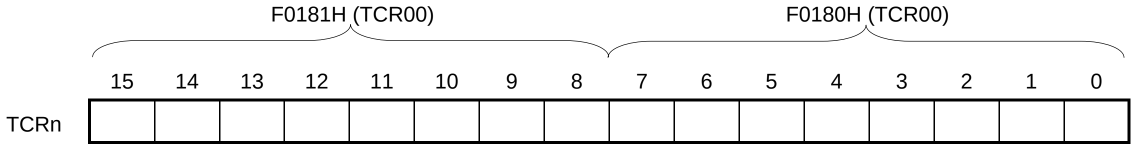
TCRn is a 16-bit read-only register and is used to count clocks.

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 MDn4 to MDn0 bits of TMRn.

Figure 6-3. Format of Timer Counter Register n (TCRn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07) After reset: FFFFH R
F01C0H, F01C1H (TCR08) to F01C6H, F01C7H (TCR11)



(n = 00 to 11)

The count value can be read by reading TCRn.

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

- When the reset signal is generated
- When the TAU0EN bit of peripheral enable register 2 (PER2) is cleared

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 TDRn even when TCRn is read.

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

Table 6-3. TCRn Register Read Value in Various Operation Modes

Operation Mode	Count Mode	TCRn Register Read Value[Note 1]			
		Operation mode change after reset	Operation mode change after count operation paused (TTn = 1)	Operation restart after count operation paused (TTn = 1)	During start trigger wait status 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 TDRn register + 1
Interval one-count mode[Note 2]	Count down	FFFFH	Undefined	Stop value	FFFFH
Up and down count mode[Note 2]	Count down and up	FFFFH	Undefined	Stop value	−

Notes 1. The read values of the TCRn register when TSn has been set to "1" while TEn = 0 are shown. The read value is held in the TCRn register until the count operation starts.

2. These operation modes are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Remark n = 00 to 11

(2) Timer data register n (TDRn)

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 MDn4 to MDn0 bits of TMRn.

The value of TDRn 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 6-4. Format of Timer Data Register n (TDRn)

Address: FFF18H, FFF19H (TDR00), FFF1AH, FFF1BH (TDR01), After reset: 0000H R/W
FFF64H, FFF65H (TDR02) to FFF76H, FFF77H (TDR11)

	FFF19H (TDR00)								FFF18H (TDR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRn																

(n = 00 to 11)

(i) When TDRn is used as compare register

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

Caution TDRn does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When TDRn is used as capture register

The count value of TCRn is captured to TDRn when the capture trigger is input.

A valid edge of the TIn pin can be selected as the capture trigger. This selection is made by TMRn.

Remark n = 00 to 11

6.3 Registers Controlling Timer Array Unit TAUS

Timer array unit TAUS is controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Timer clock select register 0 (TPS0)
- Timer mode register n (TMRn)
- Timer status register n (TSRn)
- Timer channel enable status register 0 (TE0)
- Timer channel start register 0 (TS0)
- Timer channel stop register 0 (TT0)
- Timer input select register 0 (TIS0)
- Timer output enable register 0 (TOE0)
- Timer output register 0 (TO0)
- Timer output level register 0 (TOL0)
- Timer output mode register 0 (TOM0)
- Timer triangle wave output mode register 0 (TOT0)**Note**
- Timer dead time output enable register 0 (TDE0)**Note**
- Timer real-time output register 0 (TRO0)**Note**
- Timer real-time output enable register 0 (TRE0)**Note**
- Timer real-time control register 0 (TRC0)**Note**
- Timer modulation output enable register 0 (TME0)**Note**
- TAU option mode register (OPMR)**Note**
- TAU option status register (OPSR)**Note**
- TAU option Hi-Z start trigger register (OPHS)**Note**
- TAU option Hi-Z stop trigger register (OPHT)**Note**
- Input switch control registers (ISC)
- Noise filter enable registers 1, 2 (NFEN1, NFEN2)
- Port mode registers 1, 3, 5, 7 (PM1, PM3, PM5, PM7)
- Port registers 1, 3, 5, 7 (P1, P3, P5, P7)

Note These registers are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Remark n = 00 to 11

(1) Peripheral enable register 2 (PER2)

PER2 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

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

When the inverter control function is used, be sure to set bit 1 (TAUOPEN) to 1. (For details of the inverter control function, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.)

PER2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Cautions 1. When setting timer array unit TAUS, be sure to set TAU0EN to 1 first. If TAU0EN = 0, writing to a control register of timer array unit TAUS is ignored, and all read values are default values. Similarly, when using the inverter control function, set TAUOPEN to 1 first.

2. Be sure to clear bits 2 to 7 of the PER2 register to 0.

Figure 6-5. Format of Peripheral Enable Register 2 (PER2)

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TAUOPEN	TAU0EN

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

TAUOPEN	Control of inverter control block input clock
0	Stops supply of input clock. • SFR used by the inverter control block cannot be written. • The inverter control block is in the reset status.
1	Supplies input clock. • SFR used by the inverter control block can be read/written.

(2) Timer clock select register 0 (TPS0)

TPS0 is a 16-bit register that is used to select four types of operation clocks (CK00, CK01, CK02, CK03) that are commonly supplied to each channel.
The operation clocks that can be set with each bit are as follows.

PRS000 to PRS003: CK00 (settable to timer channels 00 to 07)
PRS010 to PRS013: CK01 (settable to timer channels 00 to 07)
PRS020 to PRS023: CK02 (settable to timer channels 08 to 11)
PRS030 to PRS033: CK03 (settable to timer channels 08 to 11)

Rewriting of TPS0 during timer operation is possible only in the following cases.

PRS000 to PRS003 bits: When all the channels set to CKSn = 0 are in the operation stopped state (TEn = 0) (n = 00 to 07)
PRS010 to PRS013 bits: When all the channels set to CKSn = 1 are in the operation stopped state (TEn = 0) (n = 00 to 07)
PRS020 to PRS023 bits: When all the channels set to CKSn = 0 are in the operation stopped state (TEn = 0) (n = 08 to 11)
PRS030 to PRS033 bits: When all the channels set to CKSn = 1 are in the operation stopped state (TEn = 0) (n = 08 to 11)

TPS0 can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0000H.

Figure 6-6. Format of Timer Clock Select Register 0 (TPS0)

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

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0	PRS 033	PRS 032	PRS 031	PRS 030	PRS 023	PRS 022	PRS 021	PRS 020	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 0m3	PRS 0m2	PRS 0m1	PRS 0m0	Selection of operation clock (CK0m) Note				
					f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 40 MHz
0	0	0	0	f_{CLK}	5 MHz	10 MHz	20 MHz	40 MHz
0	0	0	1	$f_{CLK}/2$	2.5 MHz	5 MHz	10 MHz	20 MHz
0	0	1	0	$f_{CLK}/2^2$	1.25 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	1	$f_{CLK}/2^3$	625 kHz	1.25 MHz	2.5 MHz	5 MHz
0	1	0	0	$f_{CLK}/2^4$	312.5 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	1	$f_{CLK}/2^5$	156.2 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	1	0	$f_{CLK}/2^6$	78.1 kHz	156.2 kHz	312.5 kHz	625 kHz
0	1	1	1	$f_{CLK}/2^7$	39.1 kHz	78.1 kHz	156.2 kHz	312.5 kHz
1	0	0	0	$f_{CLK}/2^8$	19.5 kHz	39.1 kHz	78.1 kHz	156.2 kHz
1	0	0	1	$f_{CLK}/2^9$	9.76 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	4.88 kHz	9.76 kHz	19.5 kHz	39.1 kHz
1	0	1	1	$f_{CLK}/2^{11}$	2.44 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	1	0	0	$f_{CLK}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	9.76 kHz
1	1	0	1	$f_{CLK}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	1	0	$f_{CLK}/2^{14}$	305 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	1	$f_{CLK}/2^{15}$	153 Hz	305 Hz	610 Hz	1.22 kHz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit TAUS (TT0 = 00FFH).

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

2. m = 0 to 3

(3) Timer mode register n (TMRn)

TMRn sets an operation mode of channel n. It is used to select an operation clock (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, capture & one-count, interval one-count[Note], or up and down count[Note]).

Rewriting TMRn is prohibited when the register is in operation (when TE0 = 1). However, bits 7 and 6 (CISn1, CISn0) can be rewritten even while the register is operating with some functions (when TE0 = 1) (for details, see **6.7 Operation of Timer Array Unit TAUS as Independent Channel** and **6.8 Operation of Plural Channels of Timer Array Unit TAUS**).

TMRn can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note These modes are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Figure 6-7. Format of Timer Mode Register n (TMRn) (1/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR08) to F01CEH, F01CFH (TMR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKS n	0	CCS 1n	CCS 0n	MAST ERn	STS n2	STS n1	STS n0	CIS n1	CIS n0	0	MD n4	MD n3	MD n2	MD n1	MD n0

CKS n	Selection of operation clock (MCK) of channel n
0	Operation clock CK00 set by PRS register: timer channels 0 to 7 Operation clock CK02 set by PRS register: timer channels 8 to 11
1	Operation clock CK01 set by PRS register: timer channels 0 to 7 Operation clock CK03 set by PRS register: timer channels 8 to 11
Operation clock MCK is used by the edge detector. A count clock (TCLK) is generated depending on the setting of the CCS1n and CCS0n bits.	

CCS 1n	CCS 0n	Selection of count clock (TCLK) of channel n
0	0	Operation clock MCK specified by CKSn bit
0	1	Valid edge of input signal input from TIn pin
1	0	Selects master channel count clock (when the channel is used as a slave channel with the combination-operation function)[Note].
1	1	Selects master channel interrupt signal (when the channel is used as a slave channel with the combination-operation function)[Note].
Count clock (TCLK) is used for the timer/counter, output controller, and interrupt controller.		

Note These settings are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Caution Be sure to clear bits 14 and 5 to “0”.

Remark n = 00 to 11

Figure 6-7. Format of Timer Mode Register n (TMRn) (2/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR08) to F01CEH, F01CFH (TMR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn	0	CCS1n	CCS0n	MASTERn	STSn2	STSn1	STSn0	CISn1	CISn0	0	MDn4	MDn3	MDn2	MDn1	MDn0

MASTERn	Selection of operation in single-operation function or as slave channel in combination-operation function /operation as master channel in combination-operation function of channel n
0	Operates in single-operation function or as slave channel in combination-operation function.
1	Operates as master channel in combination-operation function.

Only the even channel can be set as a master channel (MASTERn = 1).
Be sure to use the odd channel as a slave channel (MASTERn = 0).
Clear MASTERn to 0 for a channel that is used with the single-operation function.

STSn2	STSn1	STSn0	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 TIn pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of TIn 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 combination-operation function).
1	0	1	The interrupt signal of the higher channel, regardless of the master channel setting is used$^{\text{Note}}$.
1	1	0	The trigger of the dead time control trigger generation channel is used$^{\text{Note}}$.
1	1	1	The up and down control trigger of the master channel is used$^{\text{Note}}$.
Other than above			Setting prohibited

CISn1	CISn0	Selection of TIn 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 STSn2 to STSn0 bits is other than 010B, set the CISn1 to CISn0 bits to 10B.

Note These settings are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Remark n = 00 to 11

Figure 6-7. Format of Timer Mode Register n (TMRn) (3/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR08) to F01CEH, F01CFH (TMR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn	0	CCS1n	CCS0n	MASTERn	STSn2	STSn1	STSn0	CISn1	CISn0	0	MDn4	MDn3	MDn2	MDn1	MDn0

MDn4	MDn3	MDn2	MDn1	MDn0	Operation mode of channel n	Count operation of TCR	Independent operation
0	0	0	0	1/0	Interval timer mode	Counting down	Possible
0	0	1	0	1/0	Capture mode	Counting up	Possible
0	0	1	1	0	Event counter mode	Counting down	Possible
0	1	0	0	1/0	One-count mode	Counting down	Impossible
0	1	1	0	0	Capture & one-count mode	Counting up	Possible
1	0	0	0	0	Interval one-count mode[Note 1]	Counting down	Impossible
1	0	0	1	0	Up and down count mode[Note 1]	Counting up and down	Impossible
Other than above					Setting prohibited		
The operation of the MDn0 bit varies depending on each operation mode (see table below).							

Operation mode (Value set by the MDn4 to MDn1 bits (see table above))	MDn0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0, 0) • Capture mode (0, 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, 0, 1, 1) • Up and down count mode[Note 1] (1, 0, 0, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode (0, 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.
• Interval one-count mode[Note 1] (1, 0, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
• Capture & one-count mode (0, 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. These settings are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

2. If the start trigger (TSn = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

Remark n = 00 to 11

(4) Timer status register n (TSRn)

TSRn indicates the overflow status of the counter of channel n.

TSRn is valid only in the capture mode (MDn4 to MDn1 = 0010B) and capture & one-count mode (MDn4 to MDn1 = 0110B$^{\text{Note}}$). It will not be set in any other mode.

Furthermore, CSF is valid only in the up and down count mode (MDn4 to MDn1 = 1001B)$^{\text{Note}}$. It will not be set in any other mode.

See Table 6-4 for the operation of the OVF bit in each operation mode and set/clear conditions.

TSRn can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-8. Format of Timer Status Register n (TSRn)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) After reset: 0000H R
F01D0H, F01D1H (TSR08) to F01D6H, F01D7H (TSR11)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSF Note	OVF

CSF Note	Count-up or count-down status of count clock (up and down count mode only)
0	Indicates that the count clock is counting up.
1	Indicates that the count clock is counting down.

OVF	Counter overflow status of channel n (capture mode and capture & one-count mode only)
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.	

Note This operation mode or bit is used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Table 6-4. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF	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 • Interval one-count mode$^{\text{Note}}$ • Up and down count mode$^{\text{Note}}$	clear	– (Use prohibited, not set and not cleared)
	set	

Note These operation modes are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Remark The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

(5) Timer channel enable status register 0 (TE0)

TE0 is used to enable or stop the timer operation of each channel.

When a bit of timer channel start register 0 (TS0) is set to 1, the corresponding bit of this register is set to 1.

When a bit of timer channel stop register 0 (TT0) is set to 1, the corresponding bit of this register is cleared to 0.

TE0 can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-9. Format of Timer Channel Enable Status Register 0 (TE0)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	TE11	TE10	TE09	TE08	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

TEn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.

Remark n = 00 to 11

(6) Timer channel start register 0 (TS0)

TS0 is a trigger register that is used to clear a timer counter (TCRn) and start the counting operation of each channel.

When a bit (TSn) of this register is set to 1, the corresponding bit (TEn) of timer channel enable status register 0 (TE0) is set to 1. TSn is a trigger bit and cleared immediately when TEn = 1.

TS0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-10. Format of Timer Channel Start Register 0 (TS0)

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	TS11	TS10	TS09	TS08	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

TSn	Operation enable (start) trigger of channel n
0	No trigger operation
1	TEn is set to 1 and the count operation becomes enabled. The TCRn count operation start in the count operation enabled state varies depending on each operation mode (see Table 6-5).

Caution Be sure to clear bits 15 to 12 to “0”.

Remarks 1. When the TS0 register is read, 0 is always read.

2. n = 00 to 11

Table 6-5. Operations from Count Operation Enabled State to TCRn Count Start (1/2)

Timer operation mode	Operation when TSn = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TSn = 1) until count clock generation. The first count clock loads the value of TDRn to TCRn and the subsequent count clock performs count down operation (see **6.3 (6) (a) Start timing in interval timer mode and up and down count mode**).
• Event counter mode	Writing 1 to TSn bit loads the value of TDRn to TCRn. The subsequent count clock performs count down operation. The external trigger detection selected by STSn2 to STSn0 bits in the TMRn register does not start count operation (see **6.3 (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 TCRn and the subsequent count clock performs count up operation (see **6.3 (6) (c) Start timing in capture mode**).

Table 6-5. Operations from Count Operation Enabled State to TCRn Count Start (2/2)

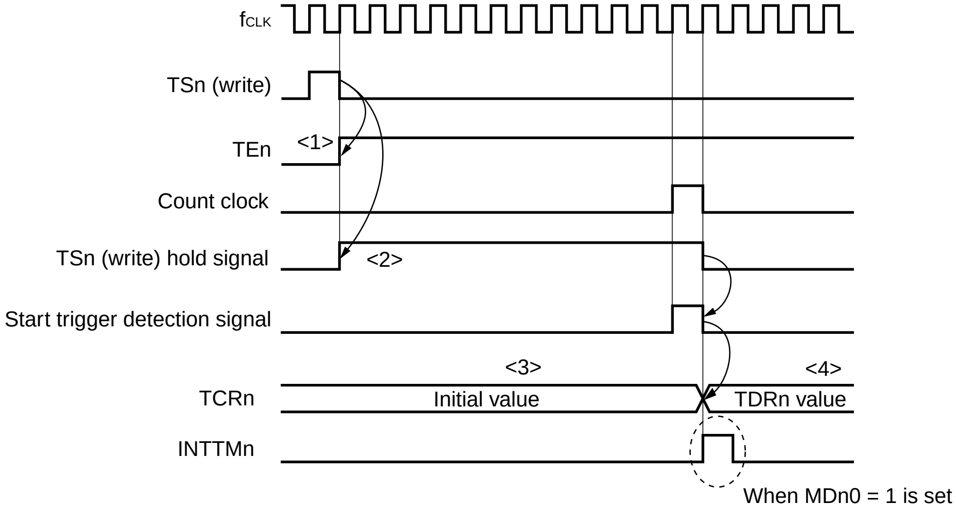
Timer operation mode	Operation when TSn = 1 is set
• One-count mode	When TEn = 0, writing 1 to TSn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of TDRn to TCRn and the subsequent count clock performs count down operation (see **6.3 (6) (d) Start timing in one-count mode**).
• Capture & one-count mode	When TEn = 0, writing 1 to TSn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to TCRn and the subsequent count clock performs count up operation (see **6.3 (6) (e) Start timing in capture & one-count mode and interval one-count mode**).
• Interval one-count mode[Note]	When TEn = 0, writing 1 to TSn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to TCRn and the subsequent count clock performs count up operation (see **6.3 (6) (e) Start timing in capture & one-count mode and interval one-count mode**).
• Up and down count mode[Note]	No operation is carried out from start trigger detection (TSn = 1) until count clock generation. The first count clock loads the value of TDRn to TCRn and the subsequent count clock performs count down operation (see **6.3 (6) (a) Start timing in interval timer mode and up and down count mode**).

Note These operation modes are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

(a) Start timing in interval timer mode and up and down count mode[Note]

<1> Writing 1 to TSn sets TEn = 1.
<2> The write data to TSn is held until count clock generation.
<3> TCRn holds the initial value until count clock generation.
<4> On generation of count clock, the "TDRn value" is loaded to TCRn and count starts.

Figure 6-11. Start Timing (In Interval Timer Mode and Up and Down Count Mode[Note])



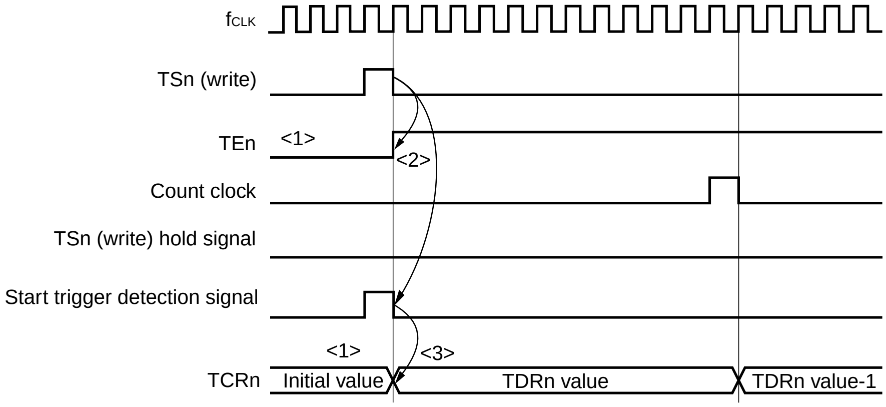
Note These operation modes are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Caution In the first cycle operation of count clock after writing TSn, 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 MDn0 = 1.

(b) Start timing in event counter mode

<1> While TEn is set to 0, TCRn holds the initial value.
<2> Writing 1 to TSn sets 1 to TEn.
<3> As soon as 1 has been written to TSn and 1 has been set to TEn, the "TDRn value" is loaded to TCRn to start counting.
<4> After that, the TCRn value is counted down according to the count clock.

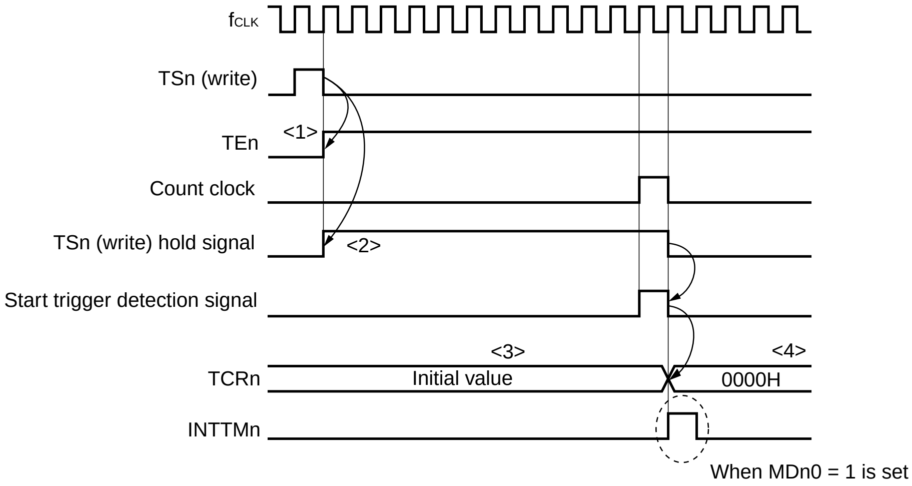
Figure 6-12. Start Timing (In Event Counter Mode)



(c) Start timing in capture mode

<1> Writing 1 to TSn sets TEn = 1.
<2> The write data to TSn is held until count clock generation.
<3> TCRn holds the initial value until count clock generation.
<4> On generation of count clock, 0000H is loaded to TCRn and count starts.

Figure 6-13. Start Timing (In Capture Mode)

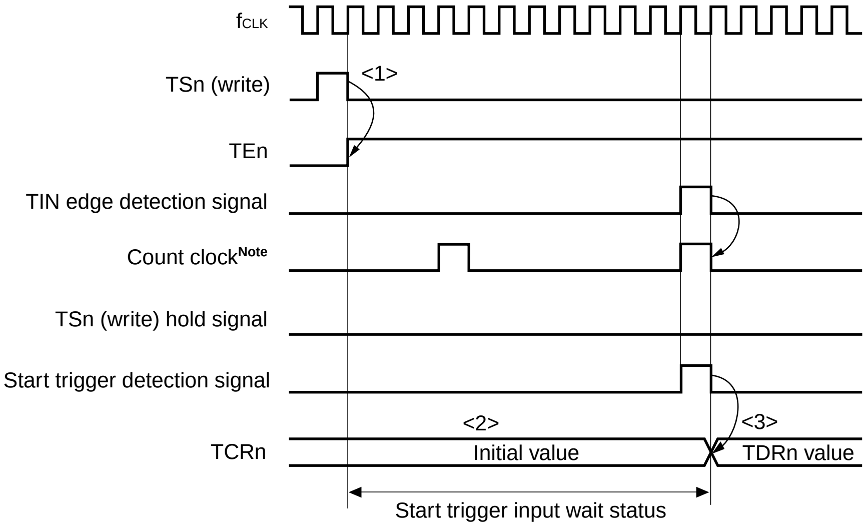


Caution In the first cycle operation of count clock after writing TSn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDn0 = 1.

(d) Start timing in one-count mode

<1> Writing 1 to TSn sets TEn = 1.
<2> Enters the start trigger input wait status, and TCRn holds the initial value.
<3> On start trigger detection, the "TDRn value" is loaded to TCRn and count starts.

Figure 6-14. Start Timing (In One-count Mode)



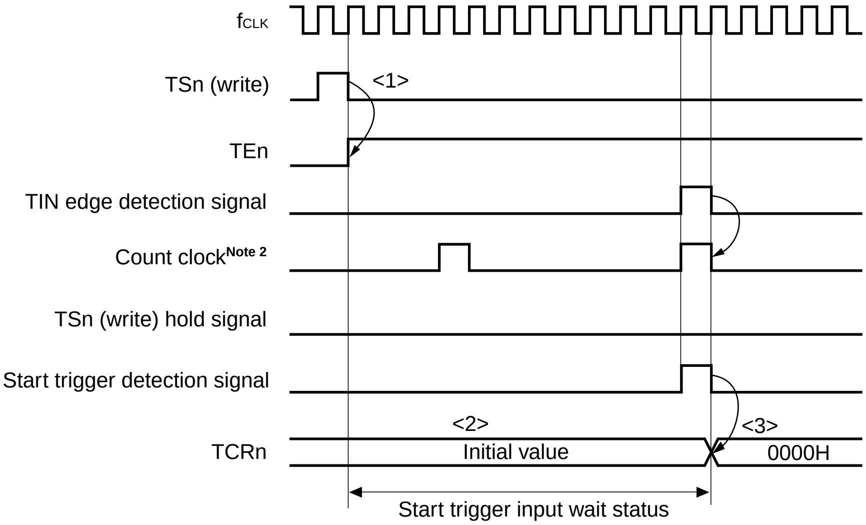
Note When the one-count mode is set, the operation clock (MCK) is selected as count clock (CCS1n, CCS0n = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TIn is used).

(e) Start timing in capture & one-count mode and interval one-count mode[Note 1]

<1> Writing 1 to TSn sets TEn = 1.
<2> Enters the start trigger input wait status, and TCRn holds the initial value.
<3> On start trigger detection, 0000H is loaded to TCRn and count starts.

Figure 6-15. Start Timing (In Capture & One-count Mode and Interval One-count Mode[Note 1])



Notes 1. This operation mode is used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

2. When the capture & one-count mode or interval one-count mode is set, the operation clock (MCK) is selected as count clock (CCS1n, CCS0n = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TIn is used).

(7) Timer channel stop register 0 (TT0)

TT0 is a trigger register that is used to clear a timer counter (TCRn) and start the counting operation of each channel.

When a bit (TTn) of this register is set to 1, the corresponding bit (TEn) of timer channel enable status register 0 (TE0) is cleared to 0. TTn is a trigger bit and cleared to 0 immediately when TEn = 0.

TT0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TT0 can be set with a 1-bit or 8-bit memory manipulation instruction with TT0L.

Reset signal generation clears this register to 0000H.

Figure 6-16. Format of Timer Channel Stop Register 0 (TT0)

Address: F01B4H, F01B5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	TT11	TT10	TT09	TT08	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

TTn	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 12 to “0”.

Remarks 1. When the TT0 register is read, 0 is always read.

2. n = 00 to 11

(8) Timer input select register 0 (TIS0)

TIS0 is used to select whether a signal input to the timer input pin (TIn) or the subsystem clock divided by four ($f_{XT}/4$) is valid for each channel.

TIS0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-17. Format of Timer Input Select Register 0 (TIS0)

Address: FFF3EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

TISn	Selection of timer input/subsystem clock used with channel n
0	Input signal of timer input pin (TIn)
1	Subsystem clock divided by 4 ($f_{XT}/4$)

Remark n = 00 to 07

(9) Timer output enable register 0 (TOE0)

TOE0 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 TOn bit of the timer output register (TO0) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TOn).

TOE0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-18. Format of Timer Output Enable Register 0 (TOE0)

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	TOE 11	TOE 10	TOE 09	TOE 08	TOE 07	TOE 06	TOE 05	TOE 04	TOE 03	TOE 02	TOE 01	TOE 00

TOE n	Timer output enable/disable of channel n
0	The TOn operation stopped by count operation (timer channel output bit). Writing to the TOn bit is enabled. The TOn pin functions as data output, and it outputs the level set to the TOn bit. The output level of the TOn pin can be manipulated by software.
1	The TOn operation enabled by count operation (timer channel output bit). Writing to the TOn bit is disabled (writing is ignored). The TOn pin functions as timer output, and the TOEn is set or reset depending on the timer operation. The TOn pin outputs the square-wave or PWM depending on the timer operation.

Caution Be sure to clear bits 15 to 12 to “0”.

Remark n = 00 to 11

(10) Timer output register 0 (TO0)

TO0 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 (TOn) of each channel.

This register can be rewritten by software only when timer output is disabled (TOEn = 0). When timer output is enabled (TOEn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P10/TO02, P11/TO03, P12/TO04, P13/TO05, P50/TO06, P51/TO07, P73/TO10, P30/TO11, or P52/SLTO pin as a port function pin, set the corresponding TOn bit to “0”.

TO0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-19. Format of Timer Output Register 0 (TO0)

Address: F01B8H, F01B9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	TO11	TO10	TO09	TO08	TO07	TO06	TO05	TO04	TO03	TO02	TO01	TO00

TOn	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 12 to “0”.

Remark n = 00 to 11

(11) Timer output level register 0 (TOL0)

TOL0 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 (TOEn = 1) in the combination-operation mode (TOMn = 1). In the toggle mode (TOMn = 0), this register setting is invalid.

TOL0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-20. Format of Timer Output Level Register 0 (TOL0)

Address: F01BCH, F01BDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	TOL11	TOL10	TOL09	TOL08	TOL07	TOL06	TOL05	TOL04	TOL03	TOL02	TOL01	TOL00

TOLn	Control of timer output level of channel n
0	Positive logic output (active-high) Adds dead time to the positive logic side when TDEn = 1 of the timer dead time output enable register 0 (TDE0) is 1[Note].
1	Inverted output (active-low) Adds dead time to the inverted logic side when TDEn = 1 of the timer dead time output enable register 0 (TDE0) is 1[Note].

Note These settings are used with the inverter control function. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Caution Be sure to clear bits 15 to 12 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. n = 00 to 11

(12) Timer output mode register 0 (TOM0)

TOM0 is used to control the timer output mode of each channel.

When a channel is used for the single-operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the combination-operation function, 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 (TOEn = 1)[Note].

TOM0 can be rewritten when timer operation is stopped (TE0 = 1).

TOM0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note The setting of each channel n when the inverter control function is used is reflected when TREn = 0 when timer output is enabled (TOEn = 1), or at the timing the timer output signal is set or reset when TREn or TMEn = 1.

Figure 6-21. Format of Timer Output Mode Register 0 (TOM0)

Address: F01BEH, F01BFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	TOM 11	TOM 10	TOM 09	TOM 08	TOM 07	TOM 06	TOM 05	TOM 04	TOM 03	TOM 02	TOM 01	TOM 00

TOM n	Control of timer output mode of channel n
0	Toggle operation mode (to produce toggle output by timer interrupt request signal (INTTMn))
1	Selects the combination-operation mode by the timer triangle wave output mode register (TOT0) setting[Note].

Note When the inverter control function is used, the combination-operation mode is selected by the timer triangle wave output mode register (TOT) setting. For details, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

Caution Be sure to clear bits 15 to 12 to "0".

Remark n: Channel number, m: Slave channel number
n = 00 to 11 (n = 00, 02, 04, 06, 08, 10 for master channel)
n < m ≤ 11 (For details of the relation between the master channel and slave channel, refer to **6.6.3 Applicable range of basic rules of combination-operation function**.)

(13) Timer triangle wave output mode register 0 (TOT0)
(14) Timer real-time output enable register 0 (TRE0)
(15) Timer real-time output register 0 (TRO0)
(16) Timer real-time control register 0 (TRC0)
(17) Timer dead time output enable register 0 (TDE0)
(18) Timer modulation output enable register 0 (TME0)
(19) TAU option mode register (OPMR)
(20) TAU option status register (OPSR)
(21) TAU option Hi-Z start trigger register (OPHS)
(22) TAU option Hi-Z stop trigger register (OPHT)

The above-mentioned registers are used with the inverter control function. For the setting and inverter control of each register, refer to **CHAPTER 7 INVERTER CONTROL FUNCTIONS**.

(23) Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to perform LIN-bus communication operation by using channel 7 in association with the serial array unit.

When ISC1 bit is set to 1, the input signal of the serial data input pin (RxD0) is selected as a timer input signal.

The ISC4 to ISC2 bits are set when selecting the P52/SLTI/SLTO pin as the timer I/O pin of timer channels 0, 1, and 8 to 11.

ISC 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 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	ISC4	ISC3	ISC2	ISC1	ISC0

ISC4	ISC3	ISC2	Selecting P52/SLTI/SLTO Pin as Timer I/O Pin											
			Channel 0		Channel 1		Channel 8		Channel 9		Channel 10		Channel 11	
			Input pin	Output pin	Input pin	Output pin	Input pin	Output pin	Input pin	Output pin	Input pin	Output pin	Input pin	Output pin
0	0	0	–	–	P52/SLTI	P52/SLTO	–	–	P31/TI09	–	P74/TI10	P73/TO10	P75/TI11	P30/TO11
0	0	1	P52/SLTI	P52/SLTO	–	–	–	–	P31/TI09	–	P74/TI10	P73/TO10	P75/TI11	P30/TO11
0	1	0	–	–	–	–	P52/SLTI	P52/SLTO	P31/TI09	–	P74/TI10	P73/TO10	P75/TI11	P30/TO11
0	1	1	–	–	–	–	–	–	P52/SLTI	P52/SLTO	P74/TI10	P73/TO10	P75/TI11	P30/TO11
1	0	0	–	–	–	–	–	–	P31/TI09		P52/SLTI	P52/SLTO	P75/TI11	P30/TO11
1	0	1	–	–	–	–	–	–	P31/TI09	–	P74/TI10	P73/TO10	P52/SLTI	P52/SLTO
Other than the above			Setting prohibited											

ISC1	Switching channel 7 input of timer array unit TAUS
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of RxD0 pin is used as timer input (wakeup signal detection).

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 RxD0 pin as an external interrupt (to measure the pulse widths of the sync break field and sync field).

Caution Be sure to clear bits 7 to 5 to “0”.

Remark When the LIN-bus communication function is used, select the input signal of the RxD0 pin by setting ISC1 to 1.

(24) Noise filter enable registers 1, 2 (NFEN1, NFEN2)

NFEN1 and NFEN2 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 and synchronization of the 2 clocks is performed with the CPU/peripheral hardware clock (f_{CLK}). When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f_{CLK}).

NFEN1 and NFEN2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 6-23. Format of Noise Filter Enable Registers 1, 2 (NFEN1, NFEN2) (1/2)

Address: F0061H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	0	0

Address: F0062H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN2	0	0	0	TNFENSL	TNFEN11	TNFEN10	TNFEN09	0

TNFENSL	Enable/disable using noise filter of SLTI/SLTO/P52 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN11	Enable/disable using noise filter of TI$\overline{\text{11/SCK}}$00/P75 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN10	Enable/disable using noise filter of TI10/SI00/RxD0/P74 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN09	Enable/disable using noise filter of TI09/SI10/RxD1/SDA10/INTP1/P31 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN07	Enable/disable using noise filter of TI07/TO07/P51 pin or RxD0/SI00/P74 pin input signal[Note]
0	Noise filter OFF
1	Noise filter ON

TNFEN06	Enable/disable using noise filter of TI06/TO06/P50 pin input signal
0	Noise filter OFF
1	Noise filter ON

Note. The applicable pin can be switched by setting ISC1 of the ISC register.

ISC1 = 0: Whether or not to use the noise filter of TI07 pin can be selected.
ISC1 = 1: Whether or not to use the noise filter of RxD0 pin can be selected.

Figure 6-23. Format of Noise Filter Enable Registers 1, 2 (NFEN1, NFEN2) (2/2)

Address: F0061H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	0	0

Address: F0062H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN2	0	0	0	TNFENSL	TNFEN11	TNFEN10	TNFEN09	0

TNFEN05	Enable/disable using noise filter of TI05/TO05/P13 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04/TO04/P12 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03/TO03/P11 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02/TO02/P10 pin input signal
0	Noise filter OFF
1	Noise filter ON

(25) Port mode registers 1, 3, 5, 7 (PM1, PM3, PM5, PM7)

These registers set input/output of ports 1, 3, 5 and 7 in 1-bit units.

When using the P10/TO02/TI02, P11/TO03/TI03, P12/TO04/TI04, P13/TO05/TI05, P50/TO06/TI06, P51/TO07/TI07, P73/TO10/SO00/TxD00, and P30/TO11/SO10/TxD1 pins for timer output, set PM10 to PM13, PM30, PM50, PM51, and PM73, and the output latches of P10 to P13, P30, P50, P51, and P73 to 0.

When using the P10/TO02/TI02, P11/TO03/TI03, P12/TO04/TI04, P13/TO05/TI05, P50/TO06/TI06, P51/TO07/TI07, P31/TI09/SI10/RxD1/SDA10/INTP1, P74/TI10/SI00/RxD0 and P75/TI11/$\overline{\text{SCK00}}$ pins for timer input, set PM10 to PM13, PM31, PM50, PM51, P74 and PM75 to 1. At this time, the output latches of P10 to P13, P31, P50, P51, P74 and P75 may be 0 or 1.

PM1, PM3, PM5, and PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 6-24. Format of Port Mode Registers 1, 3, 5, and 7 (PM1, PM3, PM5, PM7)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	1	1	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	PM32	PM31	PM30

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	1	1	1	1	1	PM52	PM51	PM50

Address: FFF27H After reset: FFH R/W

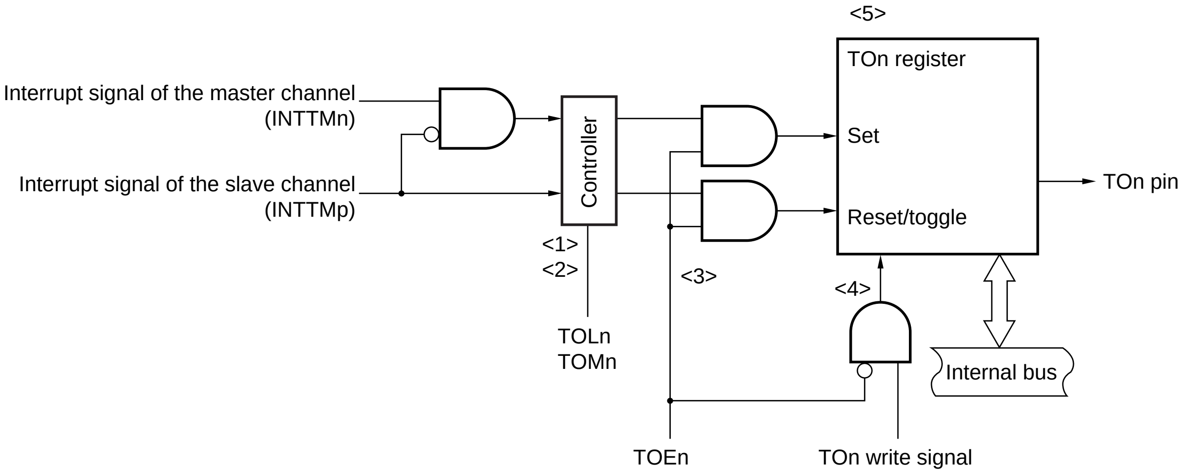
Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PMmn	Pmn pin I/O mode selection (m = 1, 3, 5, 7; n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Channel Output (TOn pin) Control

6.4.1 TOn pin output circuit configuration

Figure 6-25. Output Circuit Configuration



The following describes the TOn pin output circuit.

<1> When TOMn = 0 (toggle mode), the set value of the TOLn register is ignored and only INTTMp (slave channel timer interrupt) is transmitted to the TOn register.

<2> When TOMn = 1 (combination-operation mode), both INTTMn (master channel timer interrupt) and INTTMp (slave channel timer interrupt) are transmitted to the TOn register.
At this time, the TOLn register becomes valid and the signals are controlled as follows:

When TOLn = 0: Forward operation (INTTMn → set, INTTMp → reset)
When TOLn = 1: Reverse operation (INTTMn → reset, INTTMp → set)

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

<3> When TOEn = 1, INTTMn (master channel timer interrupt) and INTTMp (slave channel timer interrupt) are transmitted to the TOn register. Writing to the TOn register (TOn write signal) becomes invalid.
When TOEn = 1, the TOn pin output never changes with signals other than interrupt signals.
To initialize the TOn pin output level, it is necessary to set TOEn = 0 and to write a value to TOn.

<4> When TOEn = 0, writing to TOn bit to the target channel (TOn write signal) becomes valid. When TOEn = 0, neither INTTMn (master channel timer interrupt) nor INTTMp (slave channel timer interrupt) is transmitted to TOn register.

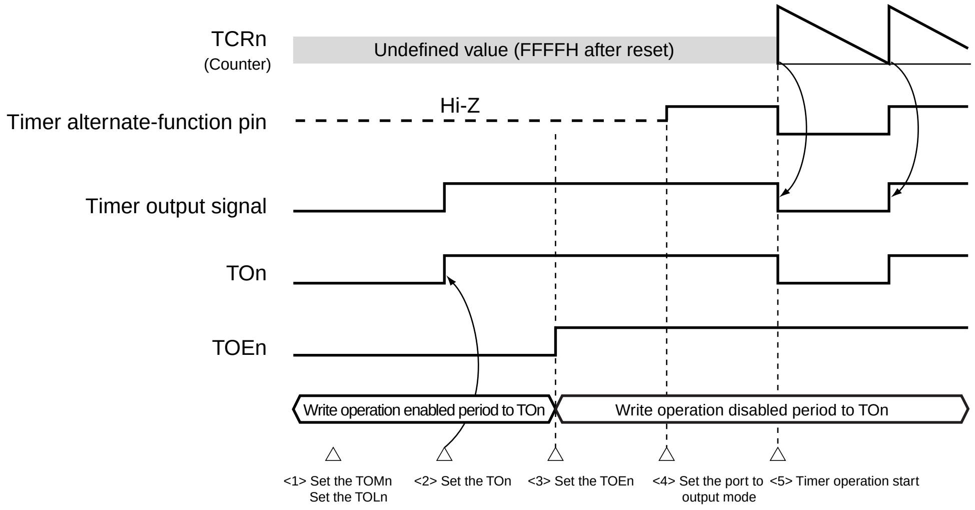
<5> The TOn register can always be read, and the TOn pin output level can be checked.

Remark n: Channel number, p: Slave channel number
n = 00 to 11 (However, n = 00, 02, 04, 06, 08, 10 in the case of the master channel)
$n < p \leq 11$ (For details of the relation between the master channel and slave channel, refer to **6.6.3 Applicable range of basic rules of combination-operation function**.)

6.4.2 TOn pin output setting

The following figure shows the procedure and status transition of TOn out put pin from initial setting to timer operation start.

Figure 6-26. Status Transition from Timer Output Setting to Operation Start


TCRn
(Counter)
Undefined value (FFFFH after reset)
Timer alternate-function pin
Hi-Z
Timer output signal
TOn
TOEn
Write operation enabled period to TOn
Write operation disabled period to TOn
<1> Set the TOMn
Set the TOLn
<2> Set the TOn
<3> Set the TOEn
<4> Set the port to
output mode
<5> Timer operation start

<1> The operation mode of timer output is set.

- TOMn bit (0: Toggle mode, 1: Combination-operation mode)
- TOLn bit (0: Forward output, 1: Reverse output)

<2> The timer output signal is set to the initial status by setting TOn.
<3> The timer output operation is enabled by writing 1 to TOEn (writing to TOn is disabled).
<4> The port I/O setting is set to output (see **6.3 (25) Port mode registers 1, 3, 5, 7**).
<5> The timer operation is enabled (TSn = 1).

Remark n = 00 to 11

6.4.3 Cautions on channel output operation

(1) Changing values set in registers TO0, TOE0, and TOL0 during timer operation

Since the timer operations (operations of TCRn and TDRn) are independent of the TOn output circuit and changing the values set in TO0, TOE0, and TOL0 does not affect the timer operation, the values can be changed during timer operation.
When the values set in TOE0, TOL0, and TOM0 (except for TO0) are changed close to the timer interrupt (INTTMn), the waveform output to the TOn pin may be different depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMn) signal generation timing.

Remark n = 00 to 11

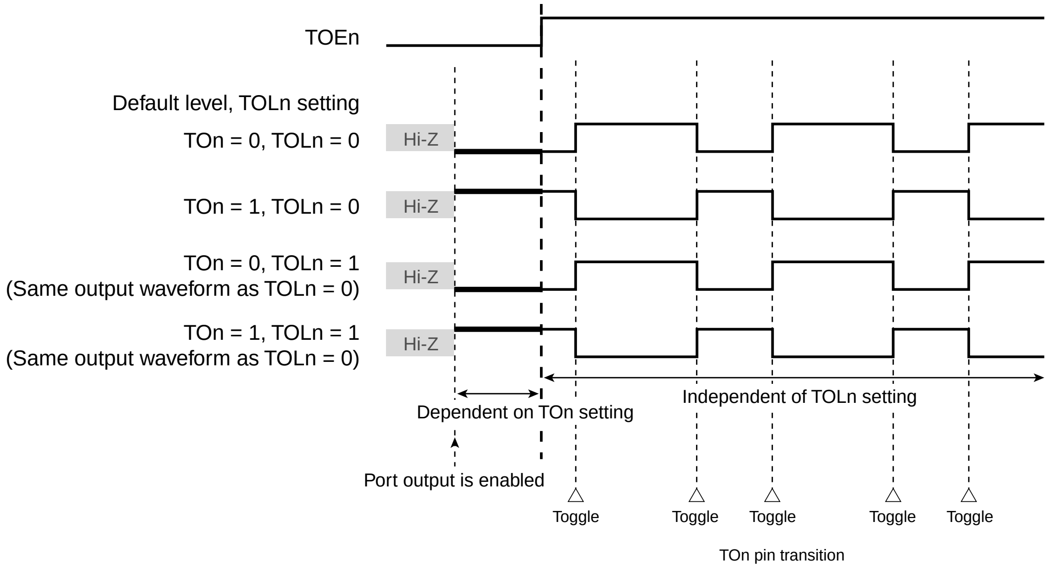
(2) Default level of TOn pin and output level after timer operation start

The following figure shows the TOn pin output level transition when writing has been done in the state of TOEn = 0 before port output is enabled and TOEn = 1 is set after changing the default level.

(a) When operation starts with TOMn = 0 setting (toggle output)

The setting of TOLn is invalid when TOMn = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TOn pin is reversed.

Figure 6-27. TOn Pin Output Status at Toggle Output (TOMn = 0)



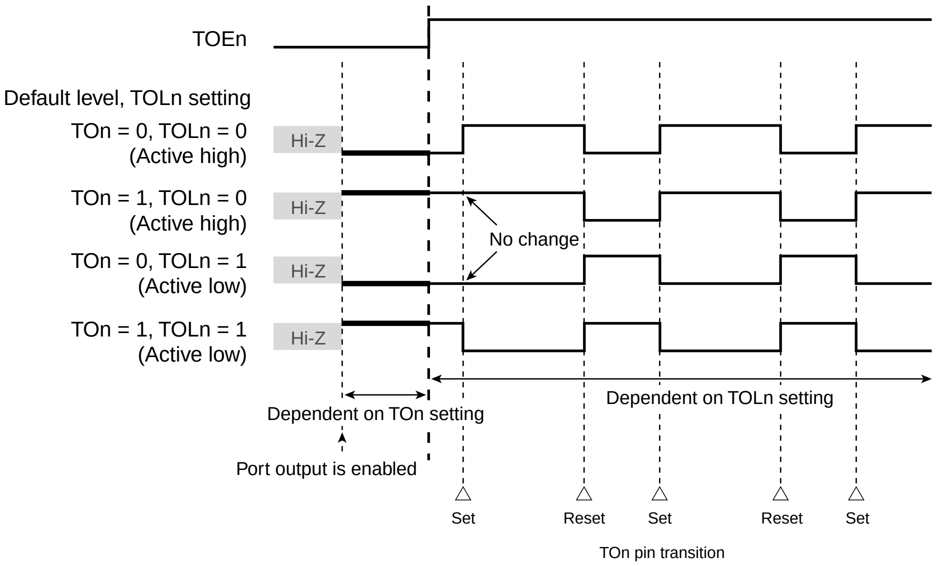
Remarks 1. Toggle: Reverse TOn pin output status

2. n = 00 to 11

(b) When operation starts with TOMn = 1 setting (Combination-operation mode (PWM output))

When TOMn = 1, the active level is determined by TOLn setting.

Figure 6-28. TOn Pin Output Status at PWM Output (TOMn = 1)



Remarks 1. Set: The output signal of TOn pin changes from inactive level to active level.
Reset: The output signal of TOn pin changes from active level to inactive level.

2. n = 00 to 11

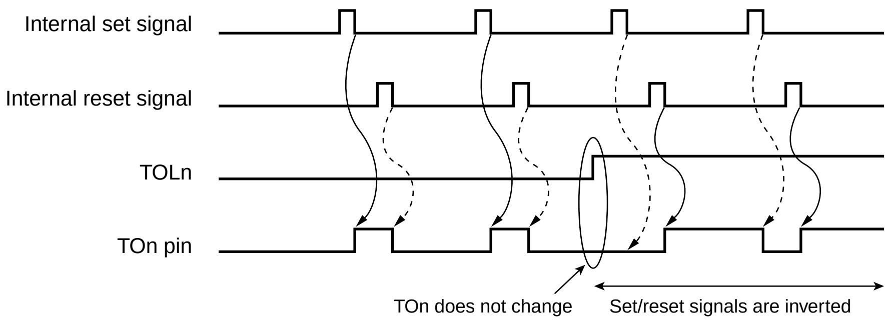
(3) Operation of TOn pin in combination-operation mode (TOMn = 1)

(a) When TOLn setting has been changed during timer operation

When the TOLn setting has been changed during timer operation, the setting becomes valid at the generation timing of TOn change condition. Rewriting TOLn does not change the output level of TOn.

The following figure shows the operation when the value of TOLn has been changed during timer operation (TOMn = 1).

Figure 6-29. Operation when TOLn Has Been Changed during Timer Operation



Remarks 1. Set: The output signal of TOn pin changes from inactive level to active level.
Reset: The output signal of TOn pin changes from active level to inactive level.

2. n = 00 to 11

(b) Set/reset timing

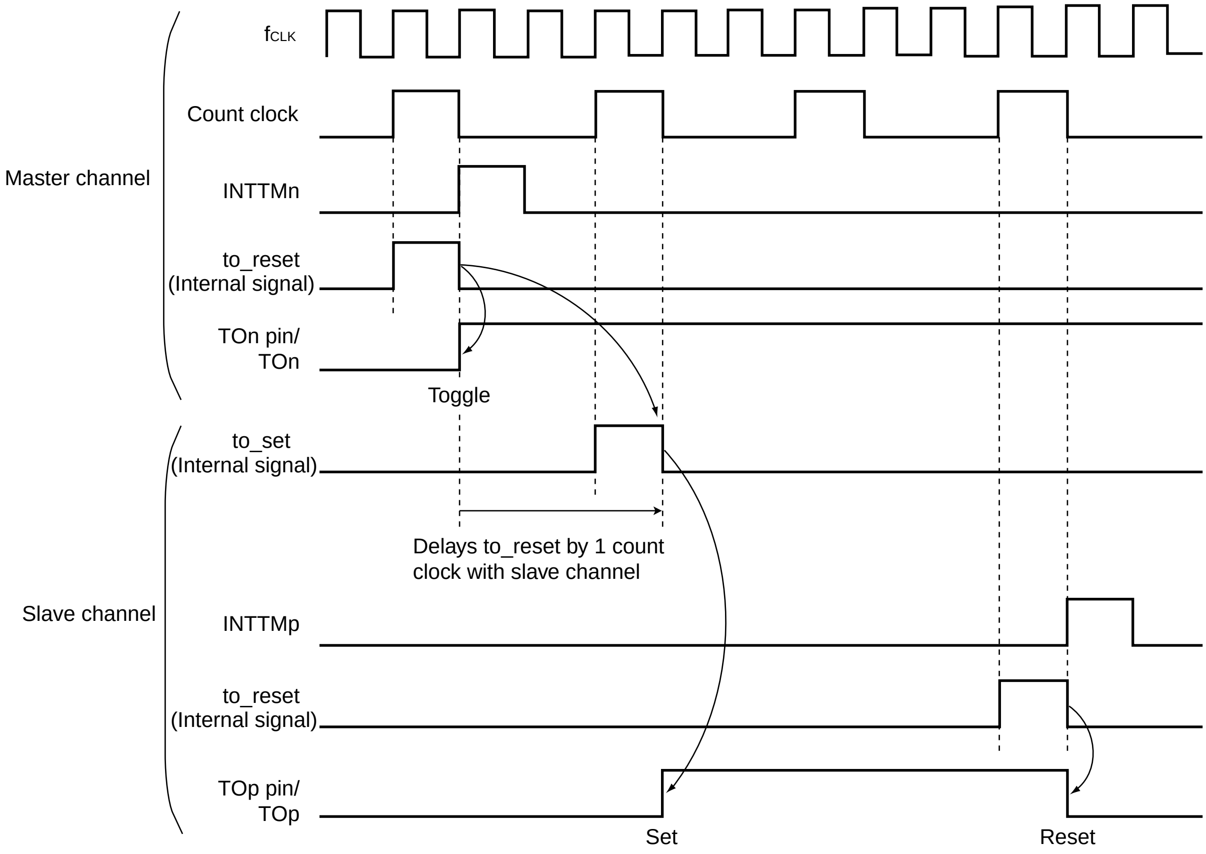
To realize 0%/100% output at PWM output, the TOn pin/TOn set timing at master channel timer interrupt (INTTMn) generation is delayed by 1 count clock by the slave channel timer interrupt (INTTMp).

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

Figure 6-30 shows the set/reset operating statuses where the master/slave channels are set as follows.

Master channel:	TOEn = 1, TOMn = 0, TOLn = 0
Slave channel:	TOEp = 1, TOMp = 1, TOLp = 0

Figure 6-30. Set/Reset Timing Operating Statuses



Remarks 1. to_reset: TOn pin reset/toggle signal

to_set: TOn pin set signal

2. n: Channel number, p: Slave channel number

n = 00 to 11 (However, n = 00, 02, 04, 06, 08, 10 in the case of the master channel)

n < p ≤ 11 (For details of the relation between the master channel and slave channel, refer to **6.6.3 Applicable range of basic rules of combination-operation function**.)

6.4.4 Collective manipulation of TOn bits

In the TO0 register, the setting bits for all the channels are located in one register in the same way as the TS0 register (channel start trigger). Therefore, TOn of all the channels can be manipulated collectively. Only specific bits can also be manipulated by setting the corresponding TOEn = 0 to a target TOn (channel output).

Figure 6-31. Example of TOn Bits Collective Manipulation

Before writing

TO0	0	0	0	0	TO11 0	TO10 1	TO09 0	TO08 0	TO07 0	TO06 0	TO05 1	TO04 0	TO03 0	TO02 0	TO01 1	TO00 0
TOE0	0	0	0	0	TOE11 1	TOE10 1	TOE09 1	TOE08 0	TOE07 0	TOE06 0	TOE05 1	TOE04 0	TOE03 1	TOE02 1	TOE01 1	TOE00 1

Data to be written

0	0	0	0	1	0	1	1	1	1	0	0	0	0	1	1
				×	×	×	O	O	O	×	O	×	×	×	×

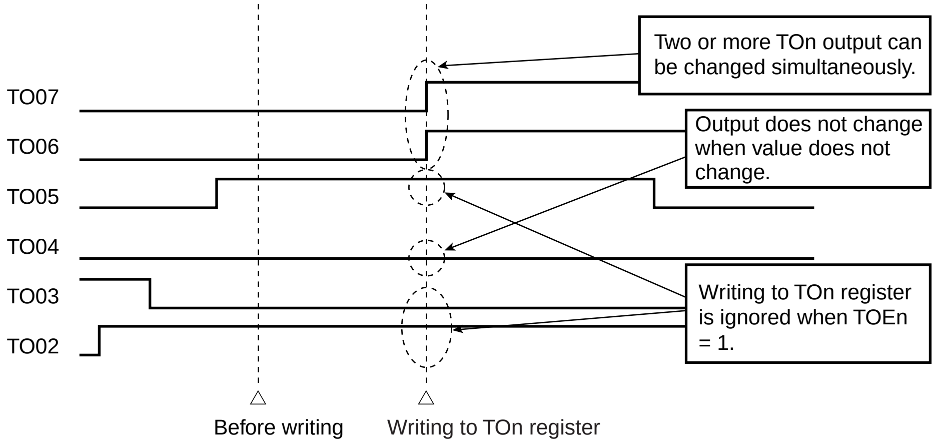
After writing

TO0	0	0	0	0	TO11 0	TO10 1	TO09 0	TO08 1	TO07 1	TO06 1	TO05 1	TO04 0	TO03 0	TO02 0	TO01 1	TO00 0

Writing is done only to TOn bits with TOEn = 0, and writing to TOn bits with TOEn = 1 is ignored.

TOn (channel output) to which TOEn = 1 is set is not affected by the write operation. Even if the write operation is done to TOn, it is ignored and the output change by timer operation is normally done.

Figure 6-32. TOn Pin Statuses by Collective Manipulation of TOn Bits


Two or more TOn output can be changed simultaneously.
Output does not change when value does not change.
Writing to TOn register is ignored when TOEn = 1.
TO07
TO06
TO05
TO04
TO03
TO02
Before writing
Writing to TOn register

Caution When TOEn = 1, even if the output by timer interrupt of each timer (INTTMn) contends with writing to TOn, output is normally done to TOn pin.

Remark n = 00 to 11

6.4.5 Timer interrupt and TOn pin output at operation start

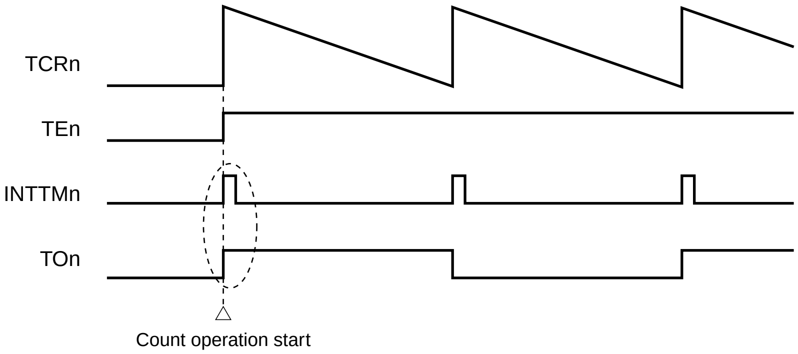
In the interval timer mode or capture mode, the MDn0 bit in the TMRn register sets whether or not to generate a timer interrupt at count start.

When MDn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMn) generation.

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

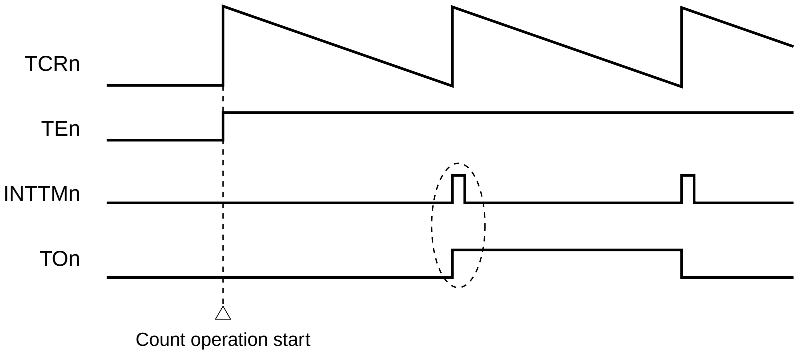
Figures 6-32 and 6-33 show operation examples when the interval timer mode (TOEn = 1, TOMn = 0) is set.

Figure 6-33. When MDn0 is set to 1



When MDn0 is set to 1, a timer interrupt (INTTMn) is output at count operation start, and TOn performs a toggle operation.

Figure 6-34. When MDn0 is set to 0



When MDn0 is set to 0, a timer interrupt (INTTMn) is not output at count operation start, and TOn does not change either. After counting one cycle, INTTMn is output and TOn performs a toggle operation.

Remark n = 00 to 11

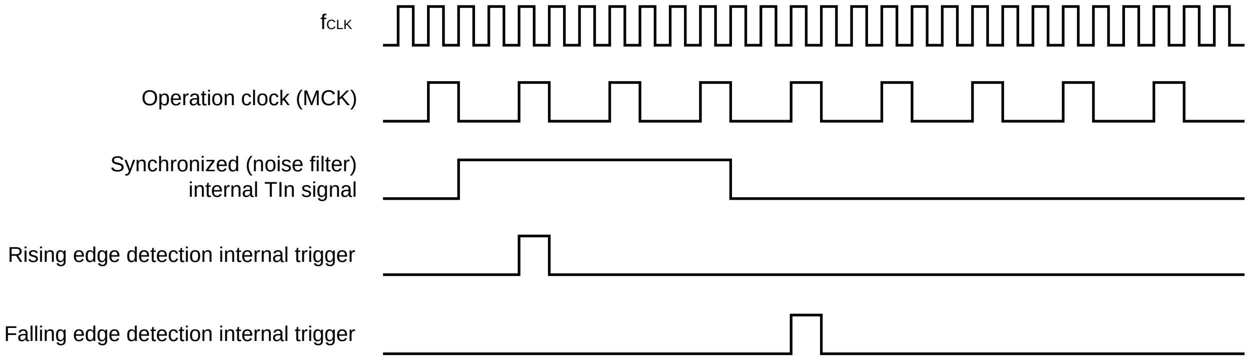
6.5 Channel Input (TIn Pin) Control

6.5.1 TIn edge detection circuit

(1) Edge detection basic operation timing

Edge detection circuit sampling is done in accordance with the operation clock (MCK).

Figure 6-35. Edge Detection Basic Operation Timing


f_{CLK}
Operation clock (MCK)
Synchronized (noise filter) internal TIn signal
Rising edge detection internal trigger
Falling edge detection internal trigger

Remark n = 00 to 11

6.6 Basic Function of Timer Array Unit TAUS

6.6.1 Overview of single-operation function and combination-operation function

Timer array unit TAUS consists of several channels and has a single-operation function that allows each channel to operate independently, and a combination-operation function that uses two or more channels in combination.

The single-operation function can be used for any channel, regardless of the operation mode of the other channels.

The combination-operation function is realized by combining a master channel (reference timer that mainly counts periods) and a slave channel (timer that operates in accordance with the master channel), and several rules must be observed when using this function.

6.6.2 Basic rules of combination-operation function

The basic rules of using the combination-operation function are as follows.

(1) Only an even channel (channel 0, 2, 4, etc.) can be set as a master channel.

(2) Any channel, except channel 0, can be set as a slave channel.

(3) The slave channel must be lower than the master channel.

Example: If channel 2 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel.

(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.

Example: If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

(6) A master channel can transmit INTTMn (interrupt), start software trigger, and count clock to the lower channels.

(7) A slave channel can use the INTTMn (interrupt), start software trigger, and count clock of the master channel, but it cannot transmit its own INTTMn (interrupt), start software trigger, and count clock to the lower channel.

(8) A master channel cannot use the INTTMn (interrupt), start software trigger, and count clock from the higher master channel.

(9) To simultaneously start channels that operate in combination, the TSn bit of the channels in combination must be set at the same time.

(10) To stop the channels in combination simultaneously, the TTn bit of the channels in combination must be set at the same time.

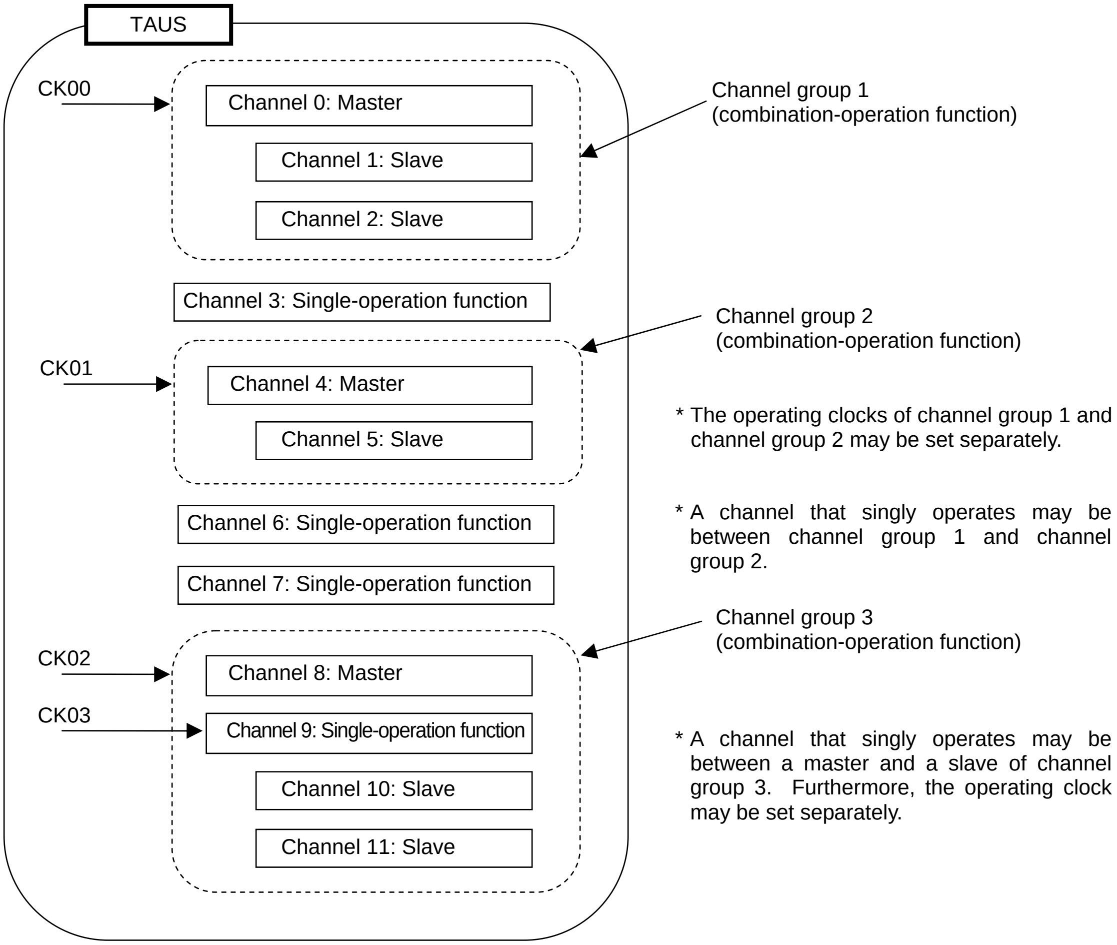
Remark n = 00 to 11 (This is, however, n = 00, 02, 04, 06, 08, 10 in the case of the master channel.)

6.6.3 Applicable range of basic rules of combination-operation function

The rules of the combination-operation function are applied in a channel group (a master channel and slave channels forming one combination-operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the combination-operation function in **6.6.2 Basic rules of combination-operation function** do not apply to the channel groups.

Example


TAUS
CK00
Channel 0: Master
Channel 1: Slave
Channel 2: Slave
Channel 3: Single-operation function
CK01
Channel 4: Master
Channel 5: Slave
Channel 6: Single-operation function
Channel 7: Single-operation function
CK02
CK03
Channel 8: Master
Channel 9: Single-operation function
Channel 10: Slave
Channel 11: Slave
Channel group 1
(combination-operation function)
Channel group 2
(combination-operation function)
Channel group 3
(combination-operation function)

* The operating clocks of channel group 1 and channel group 2 may be set separately.

* A channel that singly operates may be between channel group 1 and channel group 2.

* A channel that singly operates may be between a master and a slave of channel group 3. Furthermore, the operating clock may be set separately.

6.7 Operation of Timer Array Unit TAUS as Independent Channel

6.7.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMn (timer interrupt) at fixed intervals.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMn (timer interrupt) = Period of count clock × (Set value of TDRn + 1)

(2) Operation as square wave output

TOn performs a toggle operation as soon as INTTMn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOn can be calculated by the following expressions.

- Period of square wave output from TOn = Period of count clock × (Set value of TDRn + 1) × 2
- Frequency of square wave output from TOn = Frequency of count clock/{(Set value of TDRn + 1) × 2}

TCRn operates as a down counter in the interval timer mode.

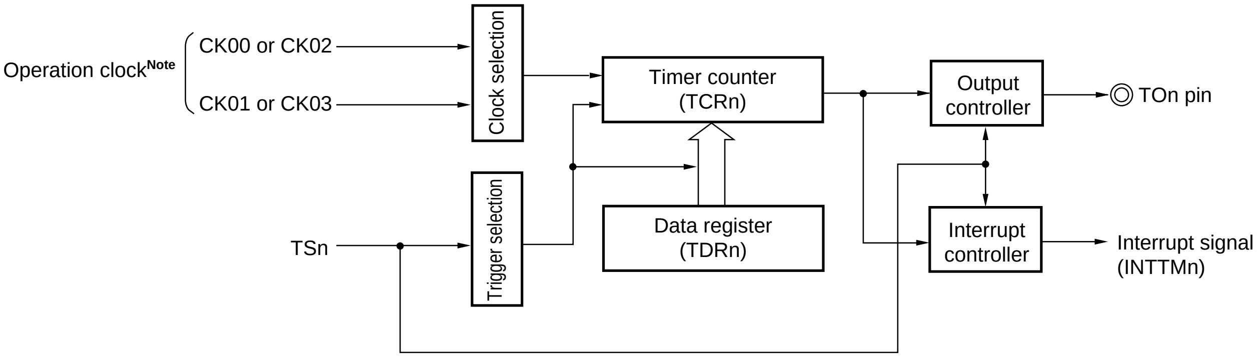
TCRn loads the value of TDRn at the first count clock after the channel start trigger bit (TSn) is set to 1. If MDn0 of TMRn = 0 at this time, INTTMn is not output and TOn is not toggled. If MDn0 of TMRn = 1, INTTMn is output and TOn is toggled.

After that, TCRn count down in synchronization with the count clock.

When TCRn = 0000H, INTTMn is output and TOn is toggled at the next count clock. At the same time, TCRn loads the value of TDRn again. After that, the same operation is repeated.

TDRn can be rewritten at any time. The new value of TDRn becomes valid from the next period.

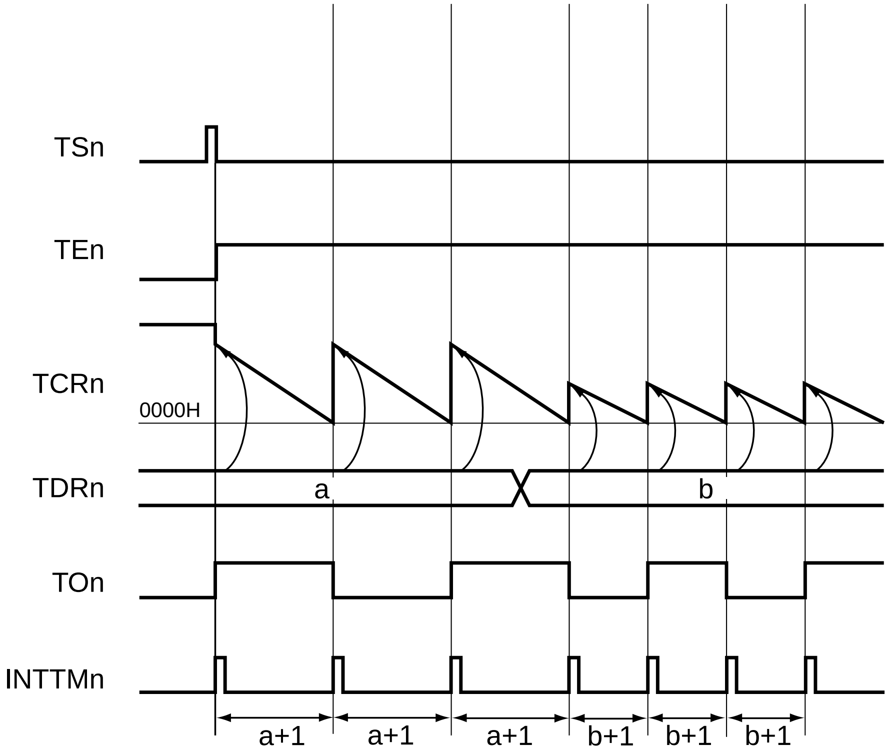
Figure 6-36. Block Diagram of Operation as Interval Timer/Square Wave Output



Note The operation clocks of channels 0 to 7 are selected from CK00 and CK01, and those of channels 8 to 11 from CK02 and CK03.

Remark n = 00 to 11

Figure 6-37. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDn0 = 1)



Remark n = 00 to 11

Figure 6-38. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output

(a) Timer mode register n (TMRn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel n
0000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMn nor inverts timer output when counting is started.
1: Generates INTTMn and inverts timer output when counting is started.

Selection of TIn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK02 (channels 8 to 11) as operation clock of channel n.
1: Selects CK03 (channels 8 to 11) as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n
TO0	TOn 1/0

0: Outputs 0 from TOn.
1: Outputs 1 from TOn.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOEn 1/0

0: Stops the TOn output operation by counting operation.
1: Enables the TOn output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOLn 0

0: Cleared to 0 when TOMn = 0 (toggle mode)

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOMn 0

0: Sets toggle mode.

Remark n = 00 to 11

Figure 6-39. Operation Procedure of Interval Timer/Square Wave Output Function

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	Sets the TMRn register (determines operation mode of channel). Sets interval (period) value to the TDRn register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOn output Clears the TOMn bit of the TOM0 register to 0 (toggle mode). Clears the TOLn bit to 0.	The TOn pin goes into Hi-Z output state.
	Sets the TOn bit and determines default level of the TOn output.	The TOn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets TOEn to 1 and enables operation of TOn.	TOn does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TOn pin outputs the TOn set level.
Operation start	Sets TOEn to 1 (only when operation is resumed). Sets the TSn bit to 1. The TSn bit automatically returns to 0 because it is a trigger bit.	TEn = 1, and count operation starts. Value of TDRn is loaded to TCRn at the count clock input. INTTMn is generated and TOn performs toggle operation if the MDn0 bit of the TMRn register is 1.
During operation	Set values of TMRn, TOM0, and TOL0 registers cannot be changed. Set value of the TDRn register can be changed. The TCRn register can always be read. The TSRn register is not used. Set values of the TO0 and TOE0 registers can be changed.	Counter (TCRn) counts down. When count value reaches 0000H, the value of TDRn is loaded to TCRn again and the count operation is continued. By detecting TCRn = 0000H, INTTMn is generated and TOn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTn bit is set to 1. The TTn bit automatically returns to 0 because it is a trigger bit.	TEn = 0, and count operation stops. TCRn holds count value and stops. The TOn output is not initialized but holds current status.
	TOEn is cleared to 0 and value is set to TOn bit.	The TOn pin outputs the TOn set level.
TAUS stop	To hold the TOn pin output level Clears TOn bit to 0 after the value to be held is set to the port register.	The TOn pin output level is held by port function.
	When holding the TOn pin output level is not necessary Switches the port mode register to input mode.	The TOn pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER2 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOn bit is cleared to 0 and the TOn pin is set to port mode.)

Operation is resumed.

Remark n = 00 to 11

6.7.2 Operation as external event counter

The timer array unit can be used as an external event counter that counts the number of times the valid input edge (external event) is detected in the TIn 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 TDRn + 1

TCRn operates as a down counter in the event counter mode.

When the channel start trigger bit (TSn) is set to 1, TCRn loads the value of TDRn.

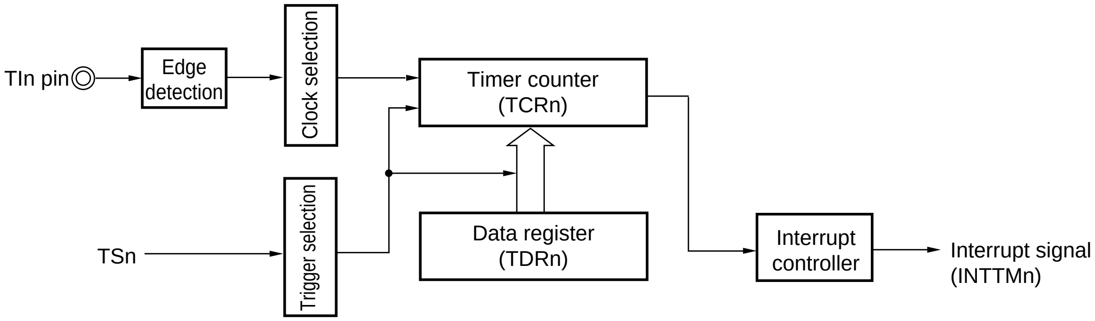
TCRn counts down each time the valid input edge of the TIn pin has been detected. When TCRn = 0000H, TCRn loads the value of TDRn again, and outputs INTTMn.

After that, the above operation is repeated.

TOn must not be used because its waveform depends on the external event and irregular.

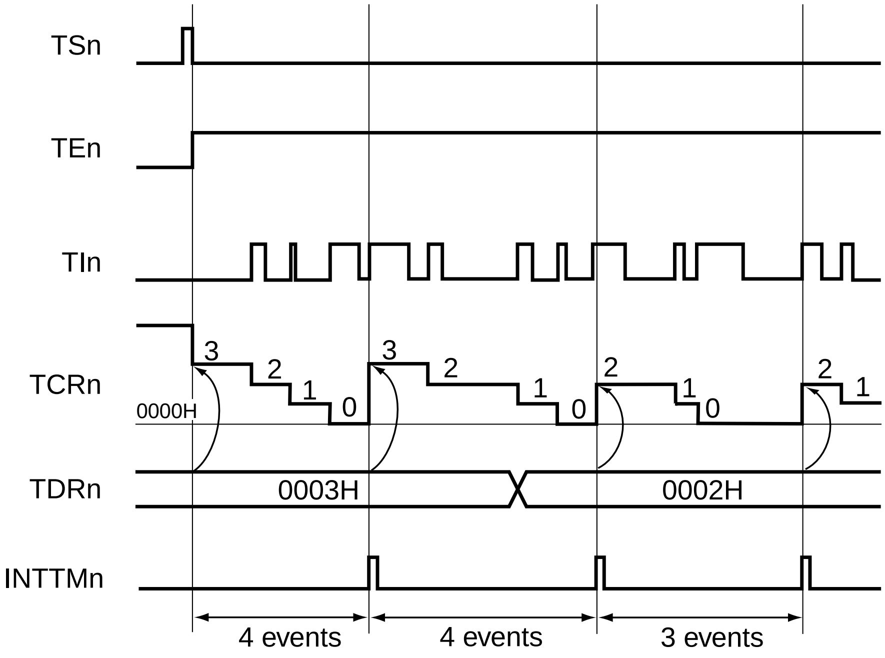
TDRn can be rewritten at any time. The new value of TDRn becomes valid during the next count period.

Figure 6-40. Block Diagram of Operation as External Event Counter



Remark n = 00 to 11

Figure 6-41. Example of Basic Timing of Operation as External Event Counter



Remark n = 00 to 11

Figure 6-42. Example of Set Contents of Registers in External Event Counter Mode

(a) Timer mode register n (TMRn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	1	1	0

Operation mode of channel n
0011B: Event count mode

Setting of operation when counting is started
0: Neither generates INTTMn nor inverts timer output when counting is started.

Selection of TIn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
01B: Selects the TIn pin input valid edge.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel n.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n
TO0	TOn
	0

0: Outputs 0 from TOn.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOEn
	0

0: Stops the TOn output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOLn
	0

0: Cleared to 0 when TOMn = 0 (toggle mode).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOMn
	0

0: Sets toggle mode.

Remark n = 00 to 11

Figure 6-43. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	Sets the TMRn register (determines operation mode of channel). Sets number of counts to the TDRn register. Clears the TOEn bit of the TOE0 register to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSn bit to 1. The TSn bit automatically returns to 0 because it is a trigger bit.	TEn = 1, and count operation starts. Value of TDRn is loaded to TCRn and detection of the TIn pin input edge is awaited.
During operation	Set value of the TDRn register can be changed. The TCRn register can always be read. The TSRn register is not used. Set values of TMRn, TOM0, TOL0, TO0, and TOE0 registers cannot be changed.	Counter (TCRn) counts down each time input edge of the TIn pin has been detected. When count value reaches 0000H, the value of TDRn is loaded to TCRn again, and the count operation is continued. By detecting TCRn = 0000H, the INTTMn output is generated. After that, the above operation is repeated.
Operation stop	The TTn bit is set to 1. The TTn bit automatically returns to 0 because it is a trigger bit.	TEn = 0, and count operation stops. TCRn holds count value and stops.
TAUS stop	The TAU0EN bit of the PER2 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n = 00 to 11

6.7.3 Operation as frequency divider

The timer array unit can be used as a frequency divider that divides a clock input to the TIn pin and outputs the result from TOn.

The divided clock frequency output from TOn can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDRn + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDRn + 1)

TCRn operates as a down counter in the interval timer mode.

After the channel start trigger bit (TSn) is set to 1, TCRn loads the value of TDRn when the TIn valid edge is detected. If MDn0 of TMRn = 0 at this time, INTTMn is not output and TOn is not toggled. If MDn0 of TMRn = 1, INTTMn is output and TOn is toggled.

After that, TCRn counts down at the valid edge of TIn. When TCRn = 0000H, it toggles TOn. At the same time, TCRn loads the value of TDRn again, and continues counting.

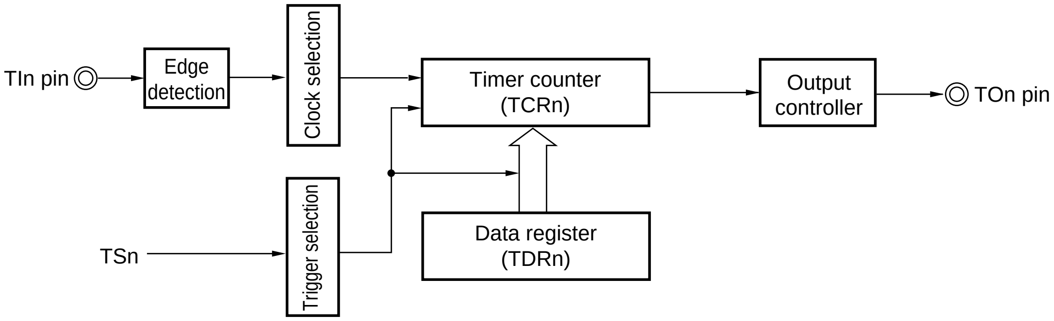
If detection of both the edges of TIn is selected, the duty factor error of the input clock affects the divided clock period of the TOn output.

The period of the TOn output clock includes a sampling error of one period of the operation clock.

Clock period of TOn output = Ideal TOn output clock period ± Operation clock period (error)

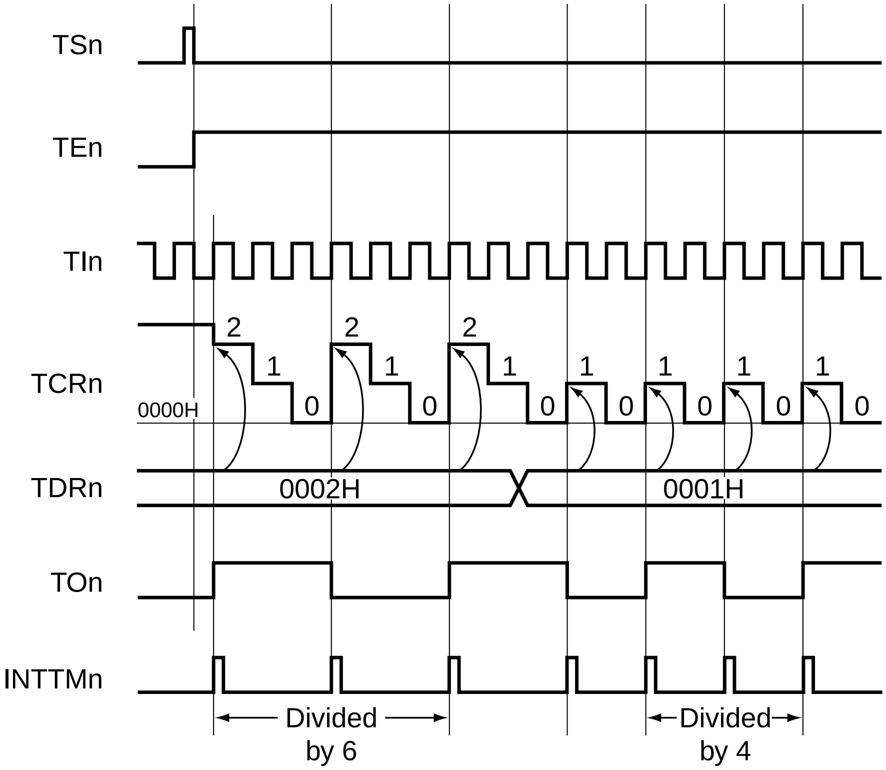
TDRn can be rewritten at any time. The new value of TDRn becomes valid during the next count period.

Figure 6-44. Block Diagram of Operation as Frequency Divider



Remark n = 10, 11

Figure 6-45. Example of Basic Timing of Operation as Frequency Divider (MDn0 = 1)



Remark n = 10, 11

Figure 6-46. Example of Set Contents of Registers During Operation as Frequency Divider

(a) Timer mode register n (TMRn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	0	0	1/0

Operation mode of channel n
0000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMn nor inverts timer output when counting is started.
1: Generates INTTMn and inverts timer output when counting is started.

Selection of TIn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
01B: Selects the TIn pin input valid edge.

Operation clock selection
0:Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel n.
1:Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n
TO0	TOn
	1/0

0: Outputs 0 from TOn.
1: Outputs 1 from TOn.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOEn
	1/0

0: Stops the TOn output operation by counting operation.
1: Enables the TOn output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOLn
	0

0: Cleared to 0 when TOMn = 0 (toggle mode)

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOMn
	0

0: Sets toggle mode.

Remark n = 10, 11

Figure 6-47. Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	Sets the TMRn register (determines operation mode of channel). Sets interval (period) value to the TDRn register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOMn bit of the TOM0 register to 0 (toggle mode). Clears the TOLn bit to 0.	The TOn pin goes into Hi-Z output state.
	Sets the TOn bit and determines default level of the TOn output.	The TOn default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEn to 1 and enables operation of TOn.	TOn does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TOn pin outputs the TOn set level.
Operation start	Sets the TOEn to 1 (only when operation is resumed).	
	Sets the TSn bit to 1. The TSn bit automatically returns to 0 because it is a trigger bit.	TEn = 1, and count operation starts. Value of TDRn is loaded to TCRn at the count clock input. INTTMn is generated and TOn performs toggle operation if the MDn0 bit of the TMRn register is 1.
During operation	Set value of the TDRn register can be changed. The TCRn register can always be read. The TSRn register is not used. Set values of TO0 and TOE0 registers can be changed. Set values of TMRn, TOM0, and TOL0 registers cannot be changed.	Counter (TCRn) counts down. When count value reaches 0000H, the value of TDRn is loaded to TCRn again, and the count operation is continued. By detecting TCRn = 0000H, INTTMn is generated and TOn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTn bit is set to 1. The TTn bit automatically returns to 0 because it is a trigger bit.	TEn = 0, and count operation stops. TCRn holds count value and stops. The TOn output is not initialized but holds current status.
	TOEn is cleared to 0 and value is set to the TO0n bit.	The TOn pin outputs the TOn set level.
TAUS stop	To hold the TOn pin output level Clears TOn bit to 0 after the value to be held is set to the port register.	The TOn pin output level is held by port function.
	When holding the TOn pin output level is not necessary Switches the port mode register to input mode.	The TOn pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER2 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOn bit is cleared to 0 and the TOn pin is set to port mode).

Operation is resumed.

Remark n = 10, 11

6.7.4 Operation as input pulse interval measurement

The count value can be captured at the TIn valid edge and the interval of the pulse input to TIn can be measured.

The pulse interval can be calculated by the following expression.

TIn input pulse interval = Period of count clock × ((10000H × TSRn: OVF) + (Capture value of TDRn + 1))

Caution The TIn pin input is sampled using the operating clock selected with the CKSn bit of the TMRn register, so an error equal to the number of operating clocks occurs.

TCRn operates as an up counter in the capture mode.

When the channel start trigger (TSn) is set to 1, TCRn counts up from 0000H in synchronization with the count clock.

When the TIn pin input valid edge is detected, the count value is transferred (captured) to TDRn and, at the same time, the counter (TCRn) is cleared to 0000H, and the INTTMn is output. If the counter overflows at this time, the OVF bit of the TSRn register is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

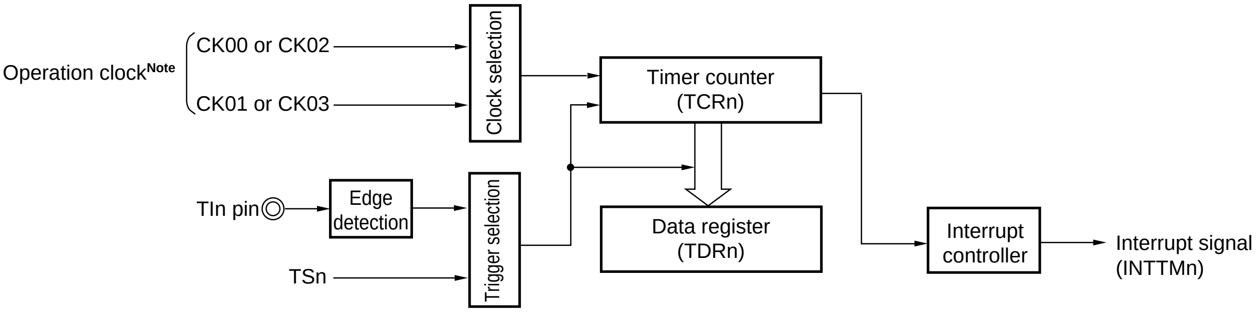
As soon as the count value has been captured to the TDRn register, the OVF bit of the TSRn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

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 TSRn 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 STSn2 to STSn0 of the TMRn register to 001B to use the valid edges of TIn as a start trigger and a capture trigger.

When TEn = 1, instead of the TIn pin input, a software operation (TSn = 1) can be used as a capture trigger.

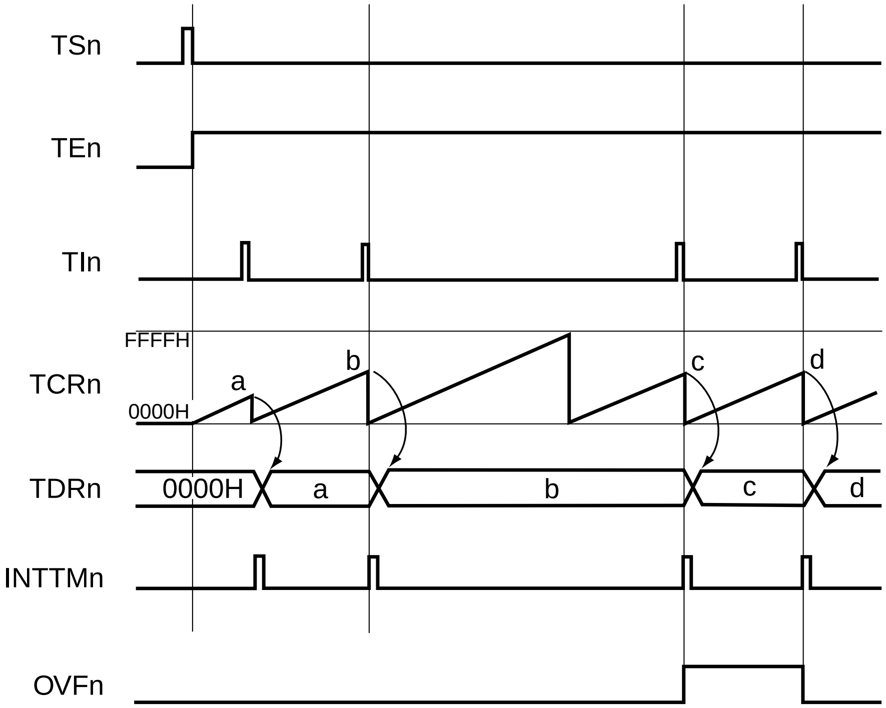
Figure 6-48. Block Diagram of Operation as Input Pulse Interval Measurement



Note The operation clocks of channels 0 to 7 are selected from CK00 and CK01, and those of channels 8 to 11 from CK02 and CK03.

Remark n = 00 to 11

Figure 6-49. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDn0 = 0)



Remark n = 00 to 11

Figure 6-50. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register n (TMRn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	0	0	0	0	1	1/0	1/0	0	0	0	1	0	1/0

Operation mode of channel n
0010B: Capture mode

Setting of operation when counting is started
0: Does not generate INTTMn when counting is started.
1: Generates INTTMn when counting is started.

Selection of TIn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Capture trigger selection
001B: Selects the TIn pin input valid edge.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel n.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n
TO0	TOn
	0

0: Outputs 0 from TOn.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOEn
	0

0: Stops TOn output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOLn
	0

0: Cleared to 0 when TOMn = 0 (toggle mode).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOMn
	0

0: Sets toggle mode.

Remark n = 00 to 11

Figure 6-51. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	Sets the TMRn register (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSn bit to 1. The TSn bit automatically returns to 0 because it is a trigger bit.	TEn = 1, and count operation starts. TCRn is cleared to 0000H at the count clock input. When the MDn0 bit of the TMRn register is 1, INTTMn is generated.
During operation	Set values of only the CISn1 and CISn0 bits of the TMRn register can be changed. The TDRn register can always be read. The TCRn register can always be read. The TSRn register can always be read. Set values of TOM0, TOL0, TO0, and TOE0 registers cannot be changed.	Counter (TCRn) counts up from 0000H. When the TIn pin input valid edge is detected, the count value is transferred (captured) to TDRn. At the same time, TCRn is cleared to 0000H, and the INTTMn signal is generated. If an overflow occurs at this time, the OVF bit of the TSRn register is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TTn bit is set to 1. The TTn bit automatically returns to 0 because it is a trigger bit.	TEn = 0, and count operation stops. TCRn holds count value and stops. The OVF bit of the TSRn register is also held.
TAUS stop	The TAU0EN bit of the PER2 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n = 00 to 11

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

By starting counting at one edge of TIn and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TIn can be measured. The signal width of TIn can be calculated by the following expression.

Signal width of TIn input = Period of count clock × ((10000H × TSRn: OVF) + (Capture value of TDRn + 1))

Caution The TIn pin input is sampled using the operating clock selected with the CKSn bit of the TMRn register, so an error equal to the number of operating clocks occurs.

TCRn operates as an up counter in the capture & one-count mode.

When the channel start trigger (TSn) is set to 1, TEn is set to 1 and the TIn pin start edge detection wait status is set.

When the TIn start valid edge (rising edge of TIn when the high-level width is to be measured) is detected, the counter counts up in synchronization with the count clock. When the valid capture edge (falling edge of TIn when the high-level width is to be measured) is detected later, the count value is transferred to TDRn and, at the same time, INTTMn is output. If the counter overflows at this time, the OVF bit of the TSRn register is set to 1. If the counter does not overflow, the OVF bit is cleared. TCRn stops at the value “value transferred to TDRn + 1”, and the TIn 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 TDRn register, the OVF bit of the TSRn 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 TSRn 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.

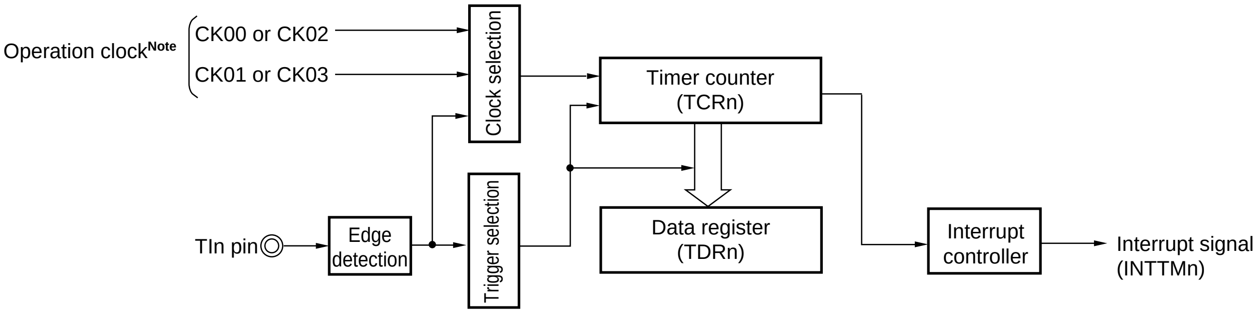
Whether the high-level width or low-level width of the TIn pin is to be measured can be selected by using the CISn1 and CISn0 bits of the TMRn register.

Because this function is used to measure the signal width of the TIn pin input, TSn cannot be set to 1 while TEn is 1.

CISn1, CISn0 of TMRn = 10B: Low-level width is measured.
CISn1, CISn0 of TMRn = 11B: High-level width is measured.

Remark n = 00 to 11

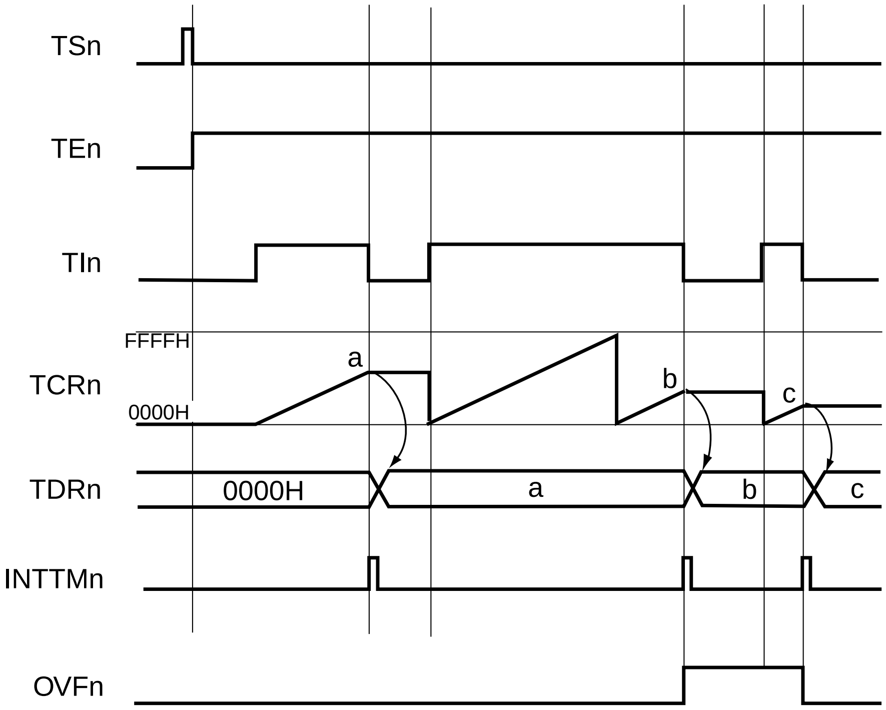
Figure 6-52. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Note The operation clocks of channels 0 to 7 are selected from CK00 and CK01, and those of channels 8 to 11 from CK02 and CK03.

Remark n = 00 to 11

Figure 6-53. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remark n = 00 to 11

Figure 6-54. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register n (TMRn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	0	0	0	1	0	1	1/0	0	0	1	1	0	0

Operation mode of channel n
0110B: Capture & one-count

Setting of operation when counting is started
0: Does not generate INTTMn when counting is started.

Selection of TIn pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TIn pin input valid edge.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel n.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n
TO0	TOn 0

0: Outputs 0 from TOn.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOEn 0

0: Stops the TOn output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOLn 0

0: Cleared to 0 when TOMn = 0 (toggle mode).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOMn 0

0: Sets toggle mode.

Remark n = 00 to 11

Figure 6-55. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	Sets the TMRn register (determines operation mode of channel). Clears TOEn to 0 and stops operation of TOn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSn bit to 1. The TSn bit automatically returns to 0 because it is a trigger bit.	TEn = 1, and the TIn pin start edge detection wait status is set.
	Detects TIn pin input count start valid edge.	Clears TCRn to 0000H and starts counting up.
During operation	Set value of the TDRn register can be changed. The TCRn register can always be read. The TSRn register is not used. Set values of TMRn, TOM0, TOL0, TO0, and TOE0 registers cannot be changed.	When the TIn pin start edge is detected, the counter (TCRn) counts up from 0000H. If a capture edge of the TIn pin is detected, the count value is transferred to TDRn and INTTMn is generated. If an overflow occurs at this time, the OVF bit of the TSRn register is set; if an overflow does not occur, the OVF bit is cleared. TCRn stops the count operation until the next TIn pin start edge is detected. After that, the above operation is repeated.
Operation stop	The TTn bit is set to 1. TTn bit automatically returns to 0 because it is a trigger bit.	TEn = 0, and count operation stops. TCRn holds count value and stops. The OVF bit of the TSRn register is also held.
TAUS stop	The TAU0EN bit of PER2 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n = 00 to 11

6.8 Operation of Plural Channels of Timer Array Unit TAUS

6.8.1 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRm (slave)}/{Set value of TDRn (master) + 1} × 100
0% output: Set value of TDRm (slave) = 0000H
100% output: Set value of TDRm (slave) ≥ {Set value of TDRn (master) + 1}

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

The master channel operates in the interval timer mode and counts the periods. When the channel start trigger (TSn) is set to 1, INTTMn is output. TCRn counts down starting from the loaded value of TDRn, in synchronization with the count clock. When TCRn = 0000H, INTTMn is output. TCRn loads the value of TDRn again. After that, it continues the similar operation.

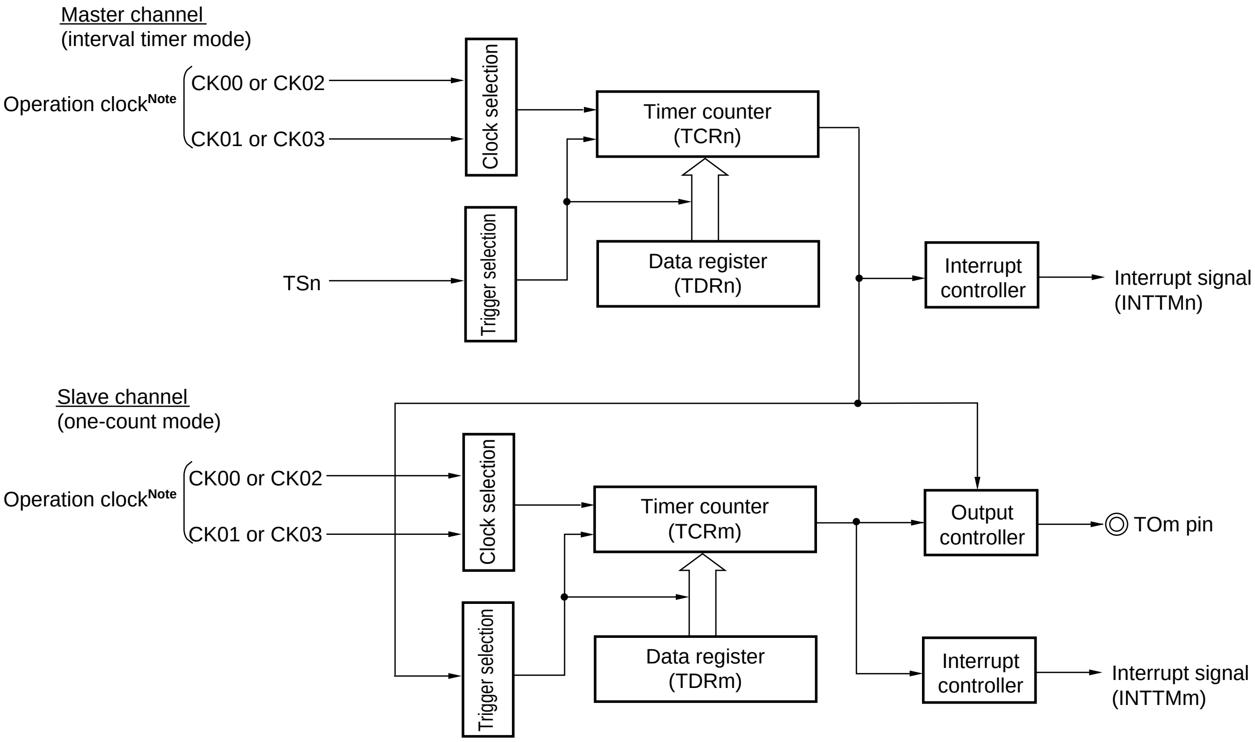
TCRm of a slave channel operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOm pin. TCRm of the slave channel loads the value of TDRm, using INTTMn of the master channel as a start trigger, and stops counting until the next start trigger (INTTMn of the master channel) is input.

The output level of TOm becomes active one count clock after generation of INTTMn from the master channel, and inactive when TCRm = 0000H.

Caution To rewrite both TDRn of the master channel and TDRm of the slave channel, a write access is necessary two times. The timing at which the values of TDRn and TDRm are loaded to TCRn and TCRm is upon occurrence of INTTMn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMn of the master channel, the TOm pin cannot output the expected waveform. To rewrite both TDRn of the master and TDRm of the slave, therefore, be sure to rewrite both the registers immediately after INTTMn is generated from the master channel.

Remark n = 00, 02, 04, 06, 08, 10
m = n + 1

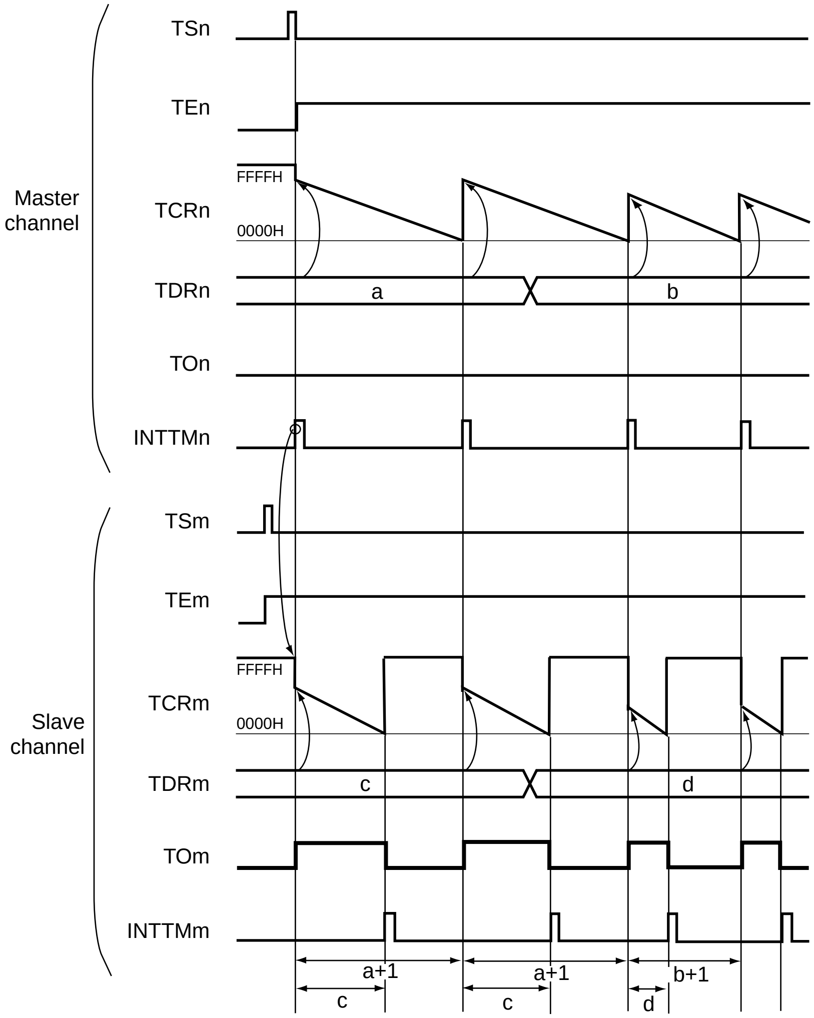
Figure 6-56. Block Diagram of Operation as PWM Function



Note The operation clocks of channels 0 to 7 are selected from CK00 and CK01, and those of channels 8 to 11 from CK02 and CK03.

Remark n = 00, 02, 04, 06, 08, 10
m = n + 1

Figure 6-57. Example of Basic Timing of Operation as PWM Function



Remark n = 00, 02, 04, 06, 08, 10
m = n + 1

Figure 6-58. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register n (TMRn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn 1/0	0	CCS1n 0	CCS0n 0	MAS TERn 1	STSn2 0	STSn1 0	STSn0 0	CISn1 0	CISn0 0	0	MDn4 0	MDn3 0	MDn2 0	MDn1 0	MDn0 1

Operation mode of channel n
0000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMn when counting is started.

Selection of TIn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel n.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n
TO0	TOn 0

0: Outputs 0 from TOn.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOEn 0

0: Stops the TOn output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOLn 0

0: Cleared to 0 when TOMn = 0 (toggle mode).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOMn 0

0: Sets toggle mode.

Remark n = 00, 02, 04, 06, 08, 10

Figure 6-59. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register m (TMRm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRm	CKSm 1/0	 0	CCS1m 0	CCS0m 0	MAS TERm 0	STSm2 1	STSm1 0	STSm0 0	CISm1 0	CISm0 0	 0	MDm4 0	MDm3 1	MDm2 0	MDm1 0	MDm0 1

Operation mode of channel m
0100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TIm pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMn of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel m.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channel m.
* Make the same setting as master channel.

(b) Timer output register 0 (TO0)

	Bit m
TO0	TOm 1/0

0: Outputs 0 from TOm.
1: Outputs 1 from TOm.

(c) Timer output enable register 0 (TOE0)

	Bit m
TOE0	TOEm 1/0

0: Stops the TOm output operation by counting operation.
1: Enables the TOm output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit m
TOL0	TOLm 1/0

0: Positive logic output (active-high)
1: Inverted output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit m
TOM0	TOMm 1

1: Sets the combination-operation mode.

Remark n = 00, 02, 04, 06, 08, 10
m = n + 1

Figure 6-60. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	Sets the TMRn and TMRm registers of two channels to be used (determines operation mode of channels). An interval (period) value is set to the TDRn register of the master channel, and a duty factor is set to the TDRm register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMm bit of the TOM0 register is set to 1 (combination-operation mode). Sets the TOLm bit. Sets the TOm bit and determines default level of the TOm output.	The TOm pin goes into Hi-Z output state. The TOm default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEm to 1 and enables operation of TOm.	TOm does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TOm pin outputs the TOm set level.

Remark n = 00, 02, 04, 06, 08, 10
m = n + 1

Figure 6-60. Operation Procedure When PWM Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	Sets TOEm (slave) to 1 (only when operation is resumed). The TSn (master) and TSm (slave) bits of the TS0 register are set to 1 at the same time. The TSn and TSm bits automatically return to 0 because they are trigger bits.	TEn = 1, TEm = 1 When the master channel starts counting, INTTMn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRn and TMRm registers cannot be changed. Set values of the TDRn and TDRm registers can be changed after INTTMn of the master channel is generated. The TCRn and TCRm registers can always be read. The TSRn and TSRm registers are not used. Set values of the TOL0, TO0, and TOE0 registers cannot be changed.	The counter of the master channel loads the TDRn value to TCRn, and counts down. When the count value reaches TCRn = 0000H, INTTMn output is generated. At the same time, the value of the TDRn register is loaded to TCRn, and the counter starts counting down again. At the slave channel, the value of TDRm is loaded to TCRm, triggered by INTTMn of the master channel, and the counter starts counting down. The output level of TOm becomes active one count clock after generation of the INTTMn output from the master channel. It becomes inactive when TCRm = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTn (master) and TTm (slave) bits are set to 1 at the same time. The TTn and TTm bits automatically return to 0 because they are trigger bits.	TEn, TEm = 0, and count operation stops. TCRn and TCRm hold count value and stops. The TOm output is not initialized but holds current status.
	TOEm of slave channel is cleared to 0 and value is set to the TOm bit.	The TOm pin outputs the TOm set level.
TAUS stop	To hold the TOm pin output level Clears TOm bit to 0 after the value to be held is set to the port register. When holding the TOm pin output level is not necessary Switches the port mode register to input mode.	The TOm pin output level is held by port function. The TOm pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER2 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOm bit is cleared to 0 and the TOm pin is set to port mode.)

Operation is resumed.

Remark n = 00, 02, 04, 06, 08, 10
m = n + 1

6.8.2 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 TIn pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDRn (master) + 2} × Count clock period
Pulse width = {Set value of TDRm (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. TCRn of the master channel starts operating upon start trigger detection and TCRn loads the value of TDRn. TCRn counts down from the value of TDRn it has loaded, in synchronization with the count clock. When TCRn = 0000H, it outputs INTTMn and stops counting until the next start trigger is detected.

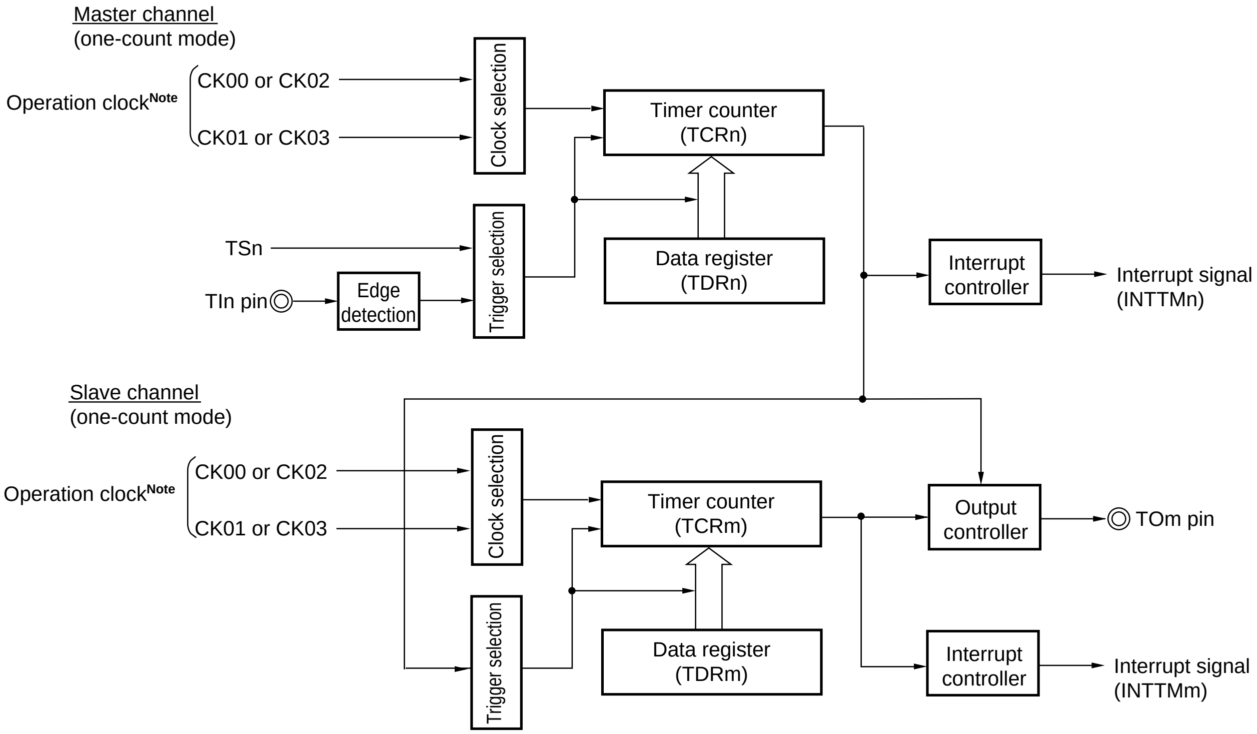
The slave channel operates in the one-count mode and counts the pulse width. TCRm of the slave channel starts operation using INTTMn of the master channel as a start trigger, and loads the TDRm value. TCRm counts down from the value of TDRm it has loaded, in synchronization with the count value. When TCRm = 0000H, it outputs INTTMm and stops counting until the next start trigger (INTTMn of the master channel) is detected. The output level of TOm becomes active one count clock after generation of INTTMn from the master channel, and inactive when TCRm = 0000H.

Instead of using the TIn pin input, a one-shot pulse can also be output using the software operation (TSn = 1) as a start trigger.

Caution The timing of loading of TDRn of the master channel is different from that of TDRm of the slave channel. If TDRn and TDRm are rewritten during operation, therefore, an illegal waveform is output. Be sure to rewrite TDRn and TDRm after INTTMn of the channel to be rewritten is generated.

Remark n = 02, 04, 06, 10
m = n + 1

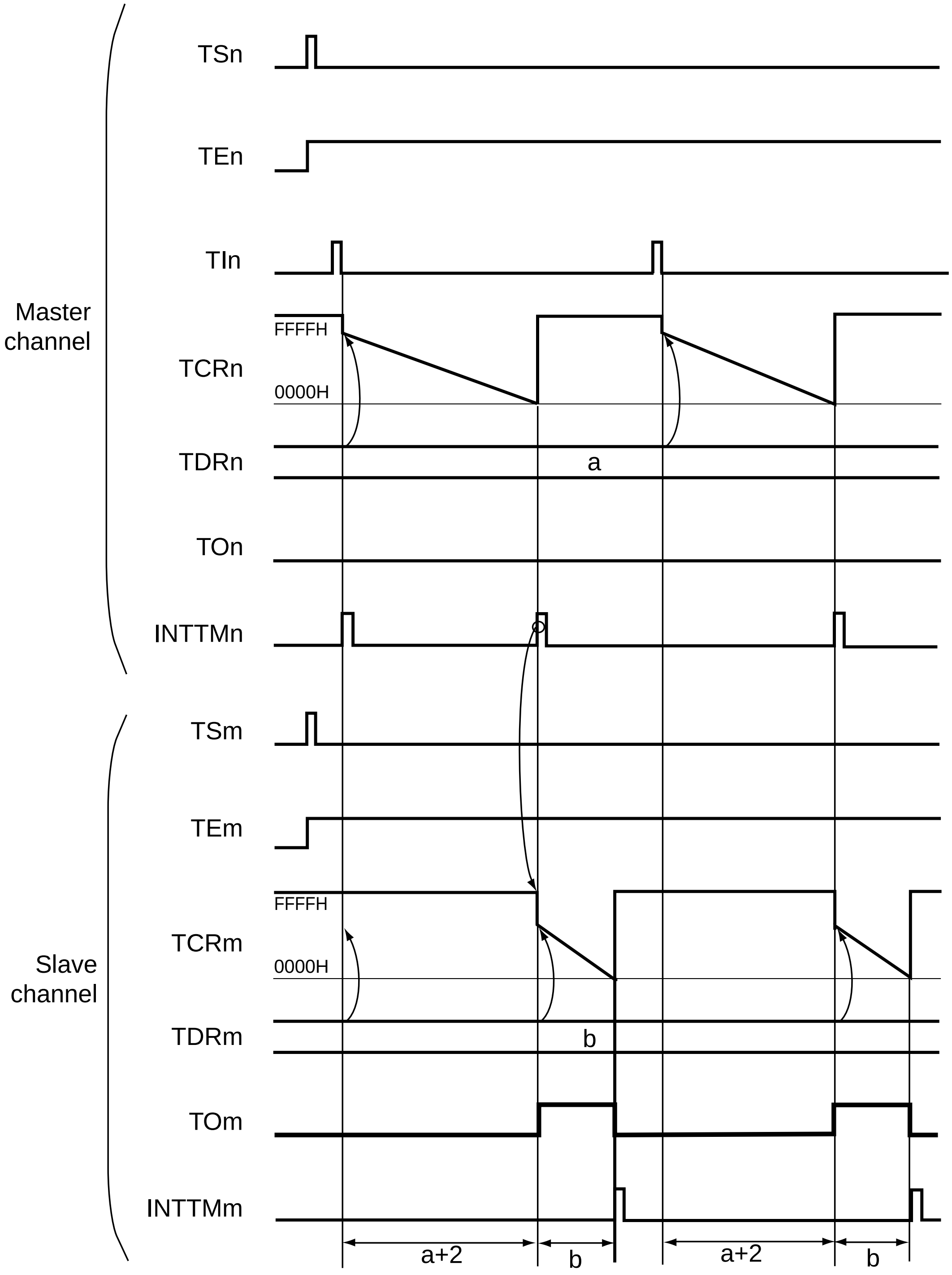
Figure 6-61. Block Diagram of Operation as One-Shot Pulse Output Function



Note The operation clocks of channels 0 to 7 are selected from CK00 and CK01, and those of channels 8 to 11 from CK02 and CK03.

Remark n = 02, 04, 06, 10
m = n + 1

Figure 6-62. Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remark n = 02, 04, 06, 10
m = n + 1

Figure 6-63. Example of Set Contents of Registers When One-Shot Pulse Output Function (Master Channel) Is Used

(a) Timer mode register n (TMRn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	0	1	0	0	1	1/0	1/0	0	0	1	0	0	0

Operation mode of channel n
0100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TIn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TIn pin input valid edge.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel n.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n
TO0	TOn
	0

0: Outputs 0 from TOn.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOEn
	0

0: Stops the TOn output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOLn
	0

0: Cleared to 0 when TOMn = 0 (toggle mode).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOMn
	0

0: Sets toggle mode.

Remark n = 02, 04, 06, 10

Figure 6-64. Example of Set Contents of Registers When One-Shot Pulse Output Function (Slave Channel) Is Used

(a) Timer mode register m (TMRm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRm	CKSm		CCS1m	CCS0m	MAS TERm	STSm2	STSm1	STSm0	CISm1	CISm0		MDm4	MDm3	MDm2	MDm1	MDm0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Operation mode of channel m
0100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TIm pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMn of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel m.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channel m.
* Make the same setting as master channel.

(b) Timer output register 0 (TO0)

	Bit m
TO0	TOm
	1/0

0: Outputs 0 from TOm.
1: Outputs 1 from TOm.

(c) Timer output enable register 0 (TOE0)

	Bit m
TOE0	TOEm
	1/0

0: Stops the TOm output operation by counting operation.
1: Enables the TOm output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit m
TOL0	TOLm
	1/0

0: Positive logic output (active-high)
1: Inverted output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit m
TOM0	TOMm
	1

1: Sets the combination-operation mode.

Remark n = 02, 04, 06, 10
m = n + 1

Figure 6-65. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	Sets the TMRn and TMRm registers of two channels to be used (determines operation mode of channels). An output delay is set to the TDRn register of the master channel, and a pulse width is set to the TDRm register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMm bit of the TOM0 register is set to 1 (combination-operation mode). Sets the TOLm bit.	The TOm pin goes into Hi-Z output state.
	Sets the TOm bit and determines default level of the TOm output.	The TOm default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEm to 1 and enables operation of TOm.	TOm does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TOm pin outputs the TOm set level.

Remark n = 02, 04, 06, 10
m = n + 1

Figure 6-65. Operation Procedure of One-Shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets TOEm (slave) to 1 (only when operation is resumed). The TSn (master) and TSm (slave) bits of the TS0 register are set to 1 at the same time. The TSn and TSm bits automatically return to 0 because they are trigger bits.	TEn and TEm are set to 1 and the master channel enters the TIn input edge detection wait status. Counter stops operating.
	Detects the TIn pin input valid edge of master channel.	Master channel starts counting.
During operation	Set values of only the CISn1 and CISn0 bits of the TMRn register can be changed. Set values of the TMRm, TDRn, TDRm, and TOM0 registers cannot be changed. The TCRn and TCRm registers can always be read. The TSRn and TSRm registers are not used. Set values of the TOL0, TO0, and TOE0 registers can be changed.	Master channel loads the value of TDRn to TCRn when the TIn pin valid input edge is detected, and the counter starts counting down. When the count value reaches TCRn = 0000H, the INTTMn output is generated, and the counter stops until the next valid edge is input to the TIn pin. The slave channel, triggered by INTTMn of the master channel, loads the value of TDRm to TCRm, and the counter starts counting down. The output level of TOm becomes active one count clock after generation of INTTMn from the master channel. It becomes inactive when TCRm = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTn (master) and TTm (slave) bits are set to 1 at the same time. The TTn and TTm bits automatically return to 0 because they are trigger bits.	TEn, TEm = 0, and count operation stops. TCRn and TCRm hold count value and stops. The TOm output is not initialized but holds current status.
	TOEm of slave channel is cleared to 0 and value is set to the TOm bit.	The TOm pin outputs the TOm set level.
TAUS stop	To hold the TOm pin output level Clears TOm bit to 0 after the value to be held is set to the port register. When holding the TOm pin output level is not necessary Switches the port mode register to input mode.	The TOm pin output level is held by port function. The TOm pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER2 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOm bit is cleared to 0 and the TOm pin is set to port mode.)

Remark n = 02, 04, 06, 10
m = n + 1

6.8.3 Operation as multiple PWM output function

By extending the PWM function and using two or more slave channels, many PWM output signals can be produced.

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 TDRn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRp (slave 1)}/{Set value of TDRn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRq (slave 2)}/{Set value of TDRn (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDRp (slave 1) > {set value of TDRn (master) + 1} or if the {set value of TDRq (slave 2)} > {set value of TDRn (master) + 1}, it is summarized into 100% output.

TCRn of the master channel operates in the interval timer mode and counts the periods.

TCRp of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOp pin. TCRp loads the value of TDRp to TCRp, using INTTMn of the master channel as a start trigger, and start counting down. When TCRp = 0000H, TCRp outputs INTTMp and stops counting until the next start trigger (INTTMn of the master channel) has been input. The output level of TOp becomes active one count clock after generation of INTTMn from the master channel, and inactive when TCRp = 0000H.

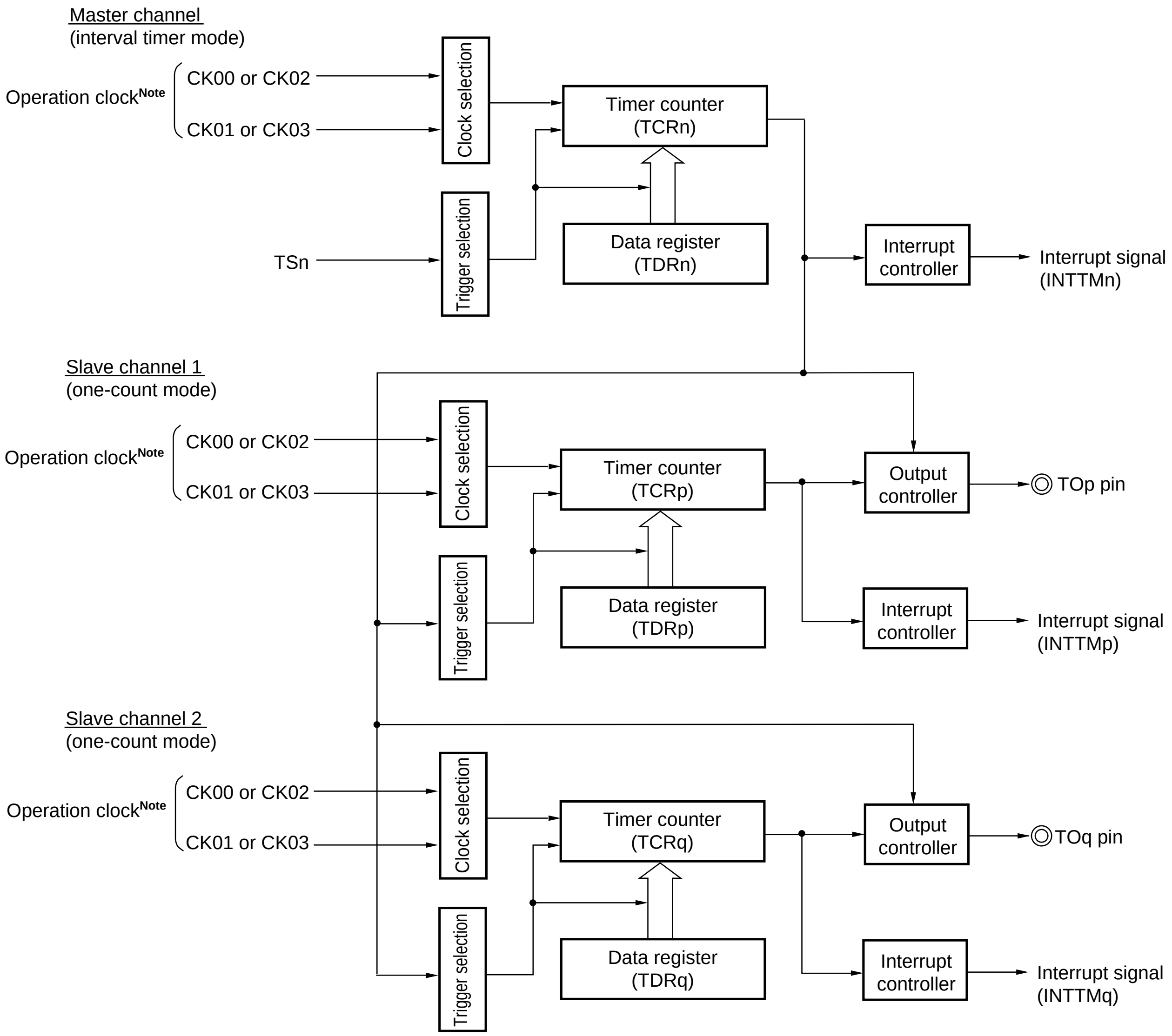
In the same way as TCRp of the slave channel 1, TCRq of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOq pin. TCRq loads the value of TDRq to TCRq, using INTTMn of the master channel as a start trigger, and starts counting down. When TCRq = 0000H, TCRq outputs INTTMq and stops counting until the next start trigger (INTTMn of the master channel) has been input. The output level of TOq becomes active one count clock after generation of INTTMn from the master channel, and inactive when TCRq = 0000H.

When channel 0 is used as the master channel as above, up to seven types of PWM signals can be output at the same time.

Caution To rewrite both TDRn of the master channel and TDRp of the slave channel 1, write access is necessary at least twice. Since the values of TDRn and TDRp are loaded to TCRn and TCRp after INTTMn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMn from the master channel, the TOp pin cannot output the expected waveform. To rewrite both TDRn of the master and TDRp of the slave, be sure to rewrite both the registers immediately after INTTMn is generated from the master channel (This applies also to TDRq of the slave channel 2).

Remark n = 00, 02, 04, 06, 08, 10
$n < p < q \leq 11$
However, p and q are consecutive integers.

Figure 6-66. Block Diagram of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



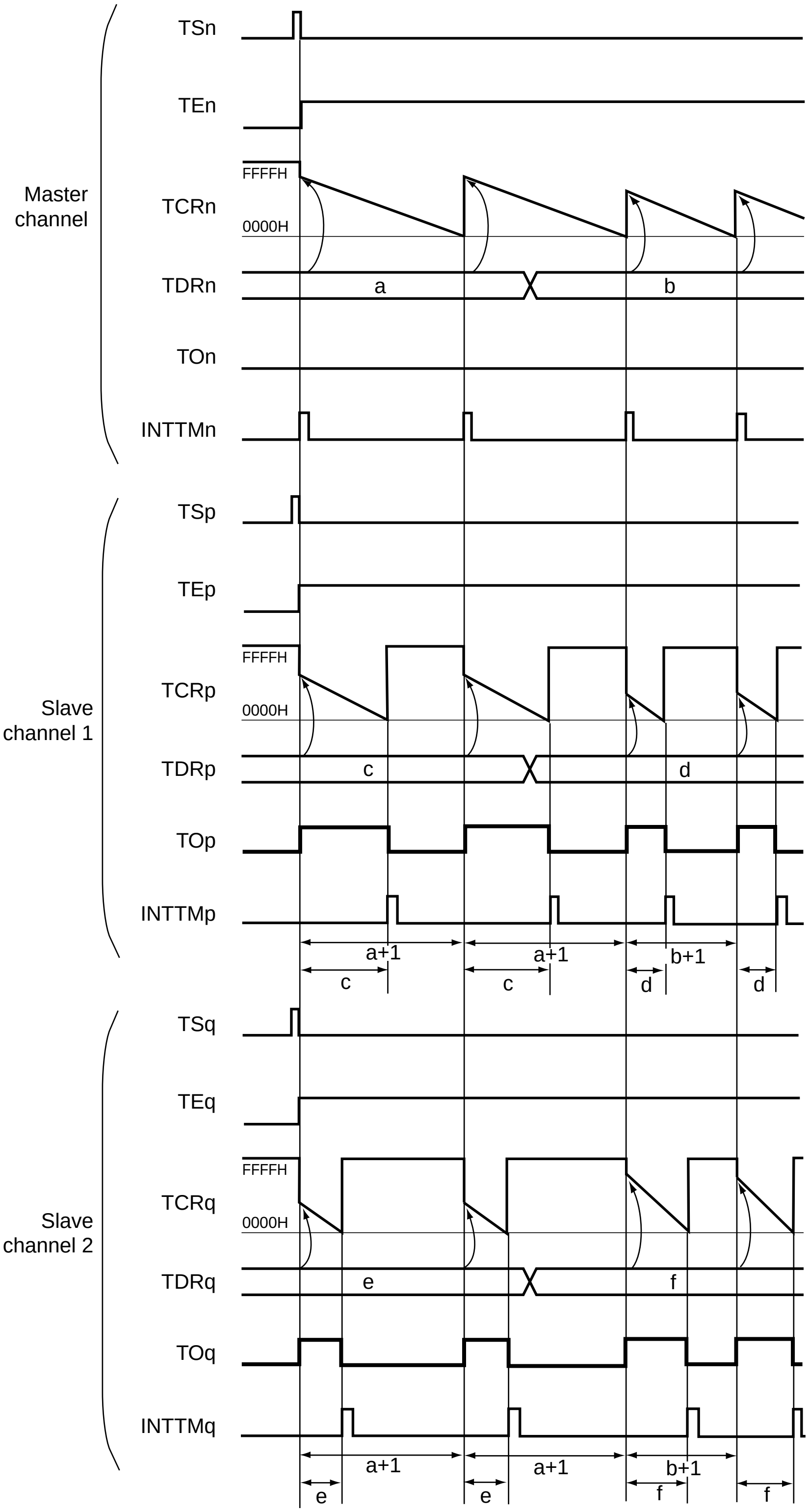
Note The operation clocks of channels 0 to 7 are selected from CK00 and CK01, and those of channels 8 to 11 from CK02 and CK03.

Remark n = 00, 02, 04, 06, 08, 10

$n < p < q \leq 11$

However, p and q are consecutive integers.

Figure 6-67. Example of Basic Timing of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



Remark n = 00, 02, 04, 06, 08, 10
$n < p < q \leq 11$
However, p and q are consecutive integers.

Figure 6-68. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register n (TMRn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
0000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMn when counting is started.

Selection of TIn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel n.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n
TO0	TOn
	0

0: Outputs 0 from TOn.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOEn
	0

0: Stops the TOn output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOLn
	0

0: Cleared to 0 when TOMn = 0 (toggle mode).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOMn
	0

0: Sets toggle mode.

Remark n = 00, 02, 04, 06, 08, 10

Figure 6-69. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs)

(a) Timer mode register p, q (TMRp, TMRq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRp	CKSp		CCS1p	CCS0p	MAS TERp	STSp2	STSp1	STSp0	CISp1	CISp0		MDp4	MDp3	MDp2	MDp1	MDp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRq	CKSq		CCS1q	CCS0q	MAS TERq	STSq2	STSq1	STSq0	CISq1	CISq0		MDq4	MDq3	MDq2	MDq1	MDq0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p, q
0100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TIp and TIq pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMn of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channels p and q.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channels p and q.
* Make the same setting as master channel.

(b) Timer output register 0 (TO0)

	Bit q	Bit p
TO0	TOq	TOp
	1/0	1/0

0: Outputs 0 from TOp or TOq.
1: Outputs 1 from TOp or TOq.

(c) Timer output enable register 0 (TOE0)

	Bit q	Bit p
TOE0	TOEq	TOEp
	1/0	1/0

0: Stops the TOp or TOq output operation by counting operation.
1: Enables the TOp or TOq output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit q	Bit p
TOL0	TOLq	TOLp
	1/0	1/0

0: Positive logic output (active-high)
1: Inverted output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit q	Bit p
TOM0	TOMq	TOMp
	1	1

1: Sets the combination-operation mode.

Remark n = 00, 02, 04, 06, 08, 10 n < p < q ≤ 11 (However, p and q are consecutive integers.)

Figure 6-70. Operation Procedure When Multiple PWM Output Function Is Used (Output Two Types of PWMs) (1/2)

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER2 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	Sets the TMRn, TMRp, and TMRq registers of each channel to be used (determines operation mode of channels). An interval (period) value is set to the TDRn register of the master channel, and a duty factor is set to the TDRp and TDRq registers of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMp and TOMq bits of the TOM0 register is set to 1 (combination-operation mode). Clears the TOLp and TOLq bits to 0.	The TOp and TOq pins go into Hi-Z output state.
	Sets the TOp and TOq bits and determines default level of the TOp and TOq outputs. →	The TOp and TOq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets TOEp or TOEq to 1 and enables operation of TOp or TOq. →	TOp or TOq does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOp and TOq pins output the TOp and TOq set levels.

Remark n = 00, 02, 04, 06, 08, 10
n < p < q ≤ 11
However, p and q are consecutive integers.

Figure 6-70. Operation Procedure When Multiple PWM Output Function Is Used (Output Two Types of PWMs (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets TOEp and TOEq (slave) to 1 (only when operation is resumed). The TSn (master), TSp, and TSq (slave) bits of the TS0 register are set to 1 at the same time. The TSn, TSp, and TSq bits automatically return to 0 because they are trigger bits.	TEn = 1, TEp, TEq = 1 When the master channel starts counting, INTTMn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRn, TMRp, TMRq, TOM0, and TOE0 registers cannot be changed. Set values of the TDRn, TDRp, and TDRq registers can be changed after INTTMn of the master channel is generated. The TCRn, TCRp, and TCRq registers can always be read. The TSRn, TSRp, and TSRq registers are not used. Set values of the TOM0, TOL0, TO0, and TOE0 registers can be changed.	The counter of the master channel loads the TDRn value to TCRn and counts down. When the count value reaches TCRn = 0000H, INTTMn is generated. At the same time, the value of the TDRn register is loaded to TCRn, and the counter starts counting down again. At the slave channel 1, the values of TDRp are transferred to TCRp, triggered by INTTMn of the master channel, and the counter starts counting down. The output levels of TOp become active one count clock after generation of the INTTMn output from the master channel. It becomes inactive when TCRp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of TDRq are transferred to TCRq, triggered by INTTMn of the master channel, and the counter starts counting down. The output levels of TOq become active one count clock after generation of the INTTMn output from the master channel. It becomes inactive when TCRq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTn bit (master), TTp, and TTq (slave) bits are set to 1 at the same time. The TTn, TTp, and TTq bits automatically return to 0 because they are trigger bits.	TEn, TEp, TEq = 0, and count operation stops. TCRn, TCRp, and TCRq hold count value and stops. The TOp and TOq output is not initialized but holds current status.
	TOEp or TOEq of slave channel is cleared to 0 and value is set to the TOp and TOq bits.	The TOp and TOq pins output the TOp and TOq set levels.
TAUS stop	To hold the TOp and TOq pin output levels Clears TOp and TOq bits to 0 after the value to be held is set to the port register. When holding the TOp and TOq pin output levels is not necessary Switches the port mode register to input mode.	The TOp and TOq pin output levels are held by port function. The TOp and TOq pin output levels go into Hi-Z output state.
	The TAU0EN bit of the PER2 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOp and TOq bits are cleared to 0 and the TOp and TOq pins are set to port mode.)

Remark n = 00, 02, 04, 06, 08, 10

$n < p < q \leq 11$

However, p and q are consecutive integers.

CHAPTER 7 INVERTER CONTROL FUNCTIONS

7.1 Outline of Functions

The μPD79F9211 can be used as an inverter control function and for motor control by using timer array unit TAUS (hereinafter referred to as "TAUS") and the TAUS option unit. It can be used with an operation clock having a 40 MHz maximum resolution (when the internal high-speed oscillation clock is used). Furthermore, the A/D converter start timing can be generated.

The following operations can be performed by using the inverter control function.

(1) Complementary PWM output function

- 6-phase PWM output function (triangular wave modulation, upper arm, lower arm, six outputs)
  A waveform of an arbitrary period, the duty, and the dead time can be generated by using channel 7 of the TAUS in combination.
- Half-bridge output function (two outputs × 2)
  A waveform of an arbitrary period, the duty, and the dead time can be generated by using channel 4 of the TAUS in combination.
- Full-bridge output function (four outputs)
  This function uses the above-mentioned half-bridge output.

(2) Non-complementary PWM output function

- 6-phase PWM output function (sawtooth wave modulation or triangular wave modulation, six outputs)
  A waveform of an arbitrary frequency, the duty, and one without dead time can be generated by using channel 7 of the TAUS in combination.

(3) Interrupt-thinning function
The interrupts of channels 0 and 4 that are used for setting the period can be thinned.

(4) Arbitrary dead time setting function
An arbitrary dead time can be set.

(5) A/D converter start timing setting function (Four types of timings can be generated.)
The A/D converter start timing can be output by using channels 8 and 9 of the TAUS.

(6) 0% and 100% output can be performed.
0% and 100% outputs can be performed both with the complementary PWM output function and non-complementary PWM output function.

(7) Forward and reverse settings of the timer output can be performed for each pin.

(8) Real-time output function (PWM modulation can be performed with this function)

(9) Forcible output stop function

The outputs of TO02 to TO07 can be set to high impedance during detection of the valid edge by the internal comparator.

The outputs of TO02 to TO07 can be set to high impedance during detection of the valid edge by an external pin input (TMOFF0, TMOFF1).

7.2 Configuration of Inverter Control Function

The inverter function can be achieved by adding functions to timer array unit TAUS.

The hardware configuration of timer array unit TAUS and the inverter control function block is shown below.

Table 7-1. Configuration of Timer Array Unit TAUS and Inverter Control Function Block

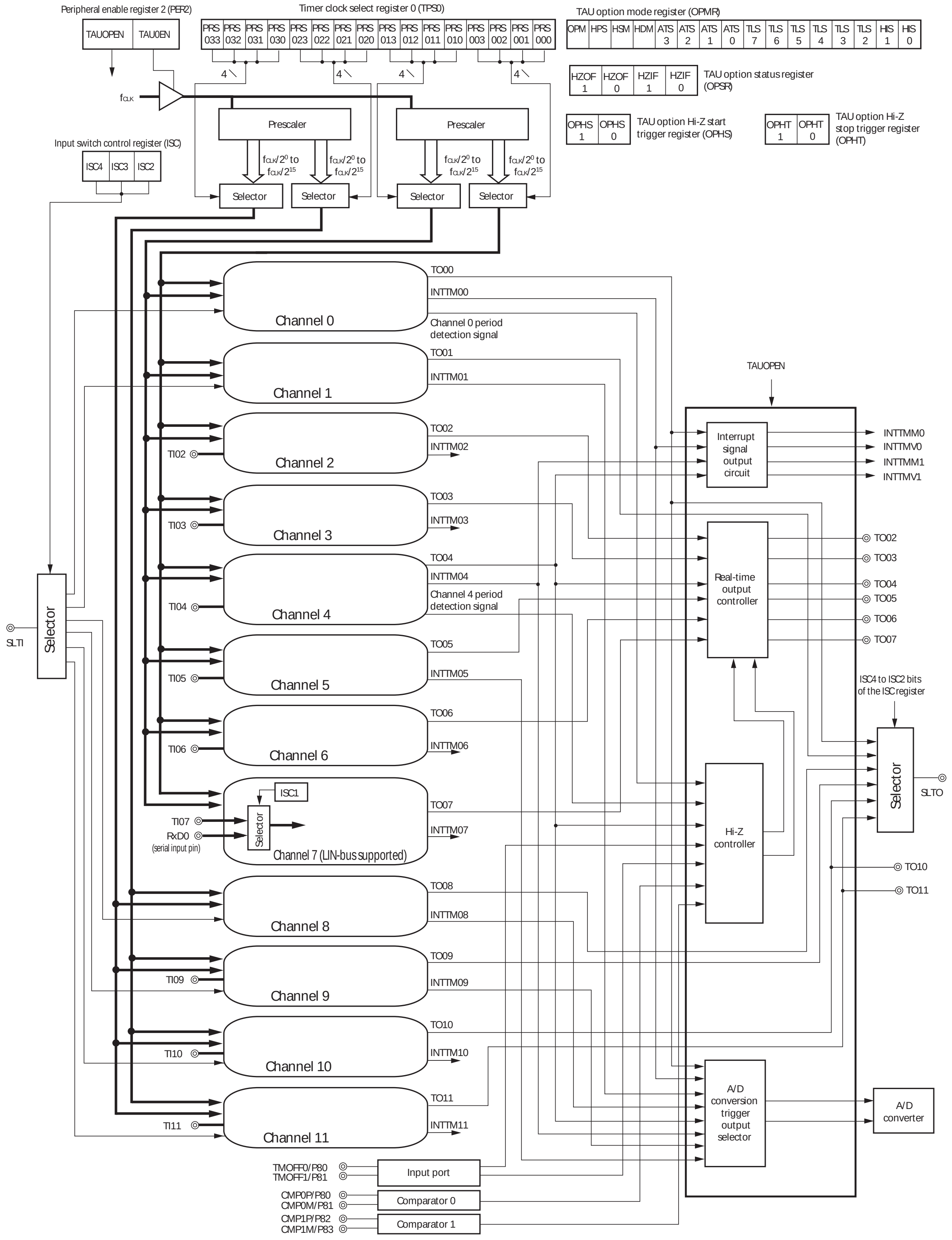
Item	Configuration
Timer/counter	Timer counter register n (TCRn)Note
Register	Timer data register n (TDRn)Note
Timer input	TI02 to TI07, TI109, TI10, TI11, SLTI pins, RxD0 pin (for LIN-bus)
Timer output	TO02 to TO07, TO10, TO11, SLTO pins, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 2 (PER2)Note • Timer clock select register 0 (TPS0)Note • Timer channel enable status register 0 (TE0)Note • Timer channel start register 0 (TS0)Note • Timer channel stop register 0 (TT0)Note • Timer input select register 0 (TIS0) Note • Timer output enable register 0 (TOE0)Note • Timer output register 0 (TO0)Note • Timer output level register 0 (TOL0)Note • Timer output mode register 0 (TOM0)Note • Timer triangle wave output mode register 0 (TOT0) • Timer dead time output enable register 0 (TDE0) • Timer real-time output register 0 (TRO0) • Timer real-time output enable register 0 (TRE0) • Timer real-time control register 0 (TRC0) • Timer modulation output enable register 0 (TME0) • TAU option mode register (OPMR) • TAU option status register (OPSR) • TAU option Hi-Z start trigger register (OPHS) • TAU option Hi-Z stop trigger register (OPHT)
	<Registers of each channel> • Timer mode register n (TMRn)Note • Timer status register n (TSRn)Note • Input switch control register (ISC)Note Used when the LIN-bus is supported • Noise filter enable registers 1, 2 (NFEN1, NFEN2)Note • Port mode registers 1, 3, 5, 7 (PM1, PM3, PM5, PM7)Note • Port registers 1, 3, 5, 7 (P1, P3, P5, P7)Note

Note. The inverter control function can be achieved by adding option functions to timer array unit TAUS. In this chapter, only the registers to be used with the inverter control function are described. For other registers that are to be used in common with timer array unit TAUS, refer to **CHAPTER 6 TIMER ARRAY UNIT TAUS**.

Remark n: Channel number (n = 00 to 11)

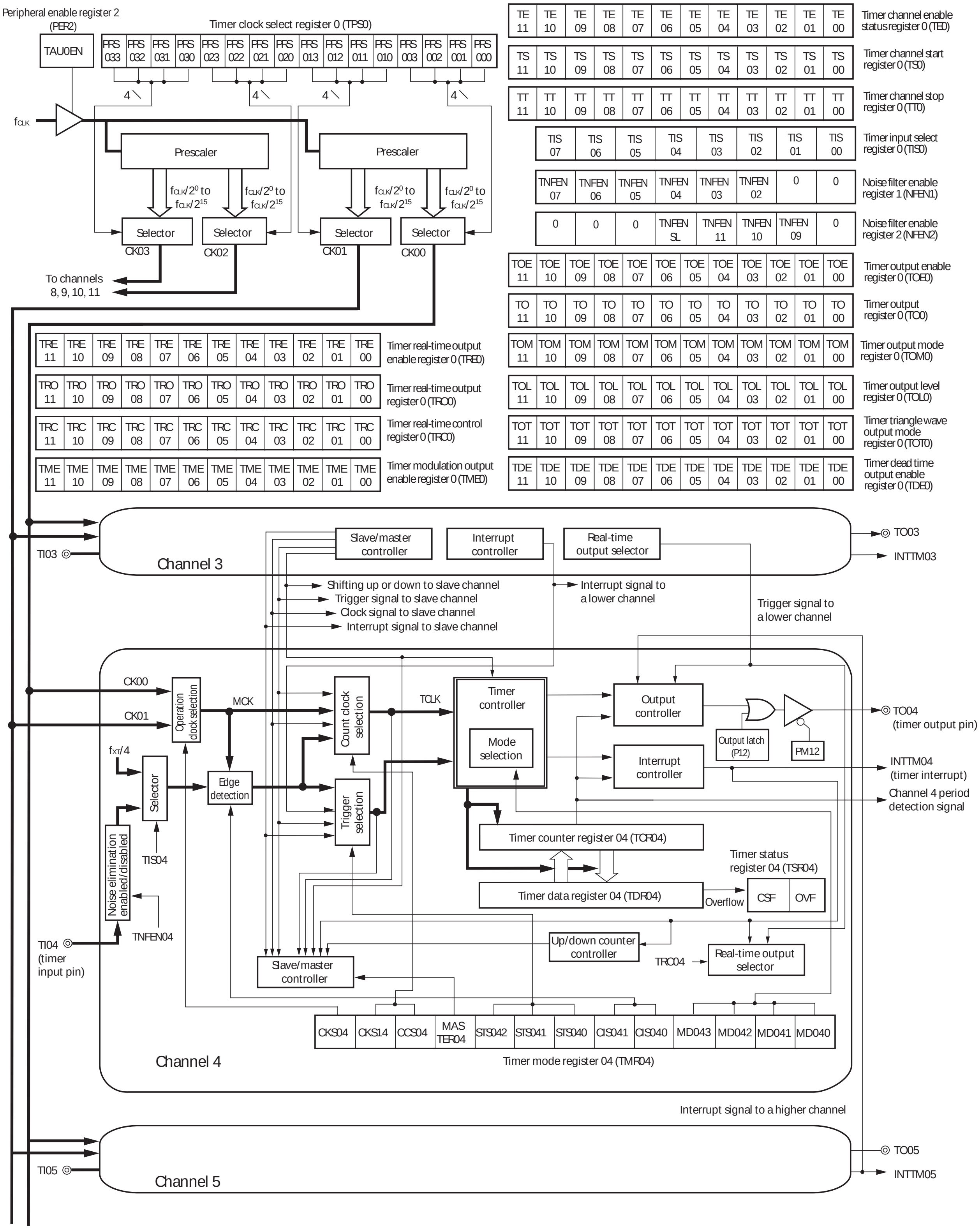
Figure 7-1 shows a block diagram.

Figure 7-1. Entire Configuration of Timer Array Unit TAUS and Inverter Control Function Block



Remark The configuration diagram in Figure 7-1 includes the registers and pins that are to be used in common with timer array unit TAUS. For details of timer array unit TAUS, refer to **CHAPTER 6 TIMER ARRAY UNIT TAUS**.

Figure 7-2. Block Diagram of Each Channel


Peripheral enable register 2 (PER2)
TAU0EN
Timer clock select register 0 (TPS0)
PRS 033 | PRS 032 | PRS 031 | PRS 030 | PRS 023 | PRS 022 | PRS 021 | PRS 020 | PRS 013 | PRS 012 | PRS 011 | PRS 010 | PRS 003 | PRS 002 | PRS 001 | PRS 000
4
fCLK
Prescaler
Prescaler
fCLK/2^0 to fCLK/2^15
Selector
Selector
Selector
Selector
CK03
CK02
CK01
CK00
To channels 8, 9, 10, 11
TE 11 | TE 10 | TE 09 | TE 08 | TE 07 | TE 06 | TE 05 | TE 04 | TE 03 | TE 02 | TE 01 | TE 00
Timer channel enable status register 0 (TE0)
TS 11 | TS 10 | TS 09 | TS 08 | TS 07 | TS 06 | TS 05 | TS 04 | TS 03 | TS 02 | TS 01 | TS 00
Timer channel start register 0 (TS0)
TT 11 | TT 10 | TT 09 | TT 08 | TT 07 | TT 06 | TT 05 | TT 04 | TT 03 | TT 02 | TT 01 | TT 00
Timer channel stop register 0 (TT0)
TIS 07 | TIS 06 | TIS 05 | TIS 04 | TIS 03 | TIS 02 | TIS 01 | TIS 00
Timer input select register 0 (TIS0)
TNFEN 07 | TNFEN 06 | TNFEN 05 | TNFEN 04 | TNFEN 03 | TNFEN 02 | 0 | 0
Noise filter enable register 1 (NFEN1)
0 | 0 | 0 | TNFEN SL | TNFEN 11 | TNFEN 10 | TNFEN 09 | 0
Noise filter enable register 2 (NFEN2)
TOE 11 | TOE 10 | TOE 09 | TOE 08 | TOE 07 | TOE 06 | TOE 05 | TOE 04 | TOE 03 | TOE 02 | TOE 01 | TOE 00
Timer output enable register 0 (TOE0)
TO 11 | TO 10 | TO 09 | TO 08 | TO 07 | TO 06 | TO 05 | TO 04 | TO 03 | TO 02 | TO 01 | TO 00
Timer output register 0 (TO0)
TRE 11 | TRE 10 | TRE 09 | TRE 08 | TRE 07 | TRE 06 | TRE 05 | TRE 04 | TRE 03 | TRE 02 | TRE 01 | TRE 00
Timer real-time output enable register 0 (TRE0)
TOM 11 | TOM 10 | TOM 09 | TOM 08 | TOM 07 | TOM 06 | TOM 05 | TOM 04 | TOM 03 | TOM 02 | TOM 01 | TOM 00
Timer output mode register 0 (TOM0)
TRO 11 | TRO 10 | TRO 09 | TRO 08 | TRO 07 | TRO 06 | TRO 05 | TRO 04 | TRO 03 | TRO 02 | TRO 01 | TRO 00
Timer real-time output register 0 (TRO0)
TOL 11 | TOL 10 | TOL 09 | TOL 08 | TOL 07 | TOL 06 | TOL 05 | TOL 04 | TOL 03 | TOL 02 | TOL 01 | TOL 00
Timer output level register 0 (TOL0)
TRC 11 | TRC 10 | TRC 09 | TRC 08 | TRC 07 | TRC 06 | TRC 05 | TRC 04 | TRC 03 | TRC 02 | TRC 01 | TRC 00
Timer real-time control register 0 (TRC0)
TOT 11 | TOT 10 | TOT 09 | TOT 08 | TOT 07 | TOT 06 | TOT 05 | TOT 04 | TOT 03 | TOT 02 | TOT 01 | TOT 00
Timer triangle wave output mode register 0 (TOT0)
TME 11 | TME 10 | TME 09 | TME 08 | TME 07 | TME 06 | TME 05 | TME 04 | TME 03 | TME 02 | TME 01 | TME 00
Timer modulation output enable register 0 (TME0)
TDE 11 | TDE 10 | TDE 09 | TDE 08 | TDE 07 | TDE 06 | TDE 05 | TDE 04 | TDE 03 | TDE 02 | TDE 01 | TDE 00
Timer dead time output enable register 0 (TDE0)
TI03
Channel 3
Slave/master controller
Interrupt controller
Real-time output selector
TO03
INTTM03
Shifting up or down to slave channel
Trigger signal to slave channel
Clock signal to slave channel
Interrupt signal to slave channel
Interrupt signal to a lower channel
Trigger signal to a lower channel
CK00
CK01
Operation clock selection
MCK
Count clock selection
TCLK
Timer controller
Mode selection
Output controller
Interrupt controller
Output latch (P12)
PM12
TO04 (timer output pin)
INTTM04 (timer interrupt)
Channel 4 period detection signal
fXT/4
Selector
Edge detection
Trigger selection
Noise elimination enabled/disabled
TIS04
TNFEN04
TI04 (timer input pin)
Timer counter register 04 (TCR04)
Timer status register 04 (TSR04)
Timer data register 04 (TDR04)
Overflow
CSF
OVF
Up/down counter controller
Slave/master controller
TRC04
Real-time output selector
CKS04 | CKS14 | CCS04 | MAS TER04 | STS042 | STS041 | STS040 | CIS041 | CIS040 | MD043 | MD042 | MD041 | MD040
Timer mode register 04 (TMR04)
Channel 4
Interrupt signal to a higher channel
TI05
Channel 5
TO05
INTTM05

Remark The block diagram in Figure 7-2 includes the registers and pins that are to be used in common with timer array unit TAUS. For details of timer array unit TAUS, refer to **CHAPTER 6 TIMER ARRAY UNIT TAUS**.

(1) Timer counter register n (TCRn)

(2) Timer data register n (TDRn)

The above-mentioned registers are used in common with timer array unit TAUS.
For details, refer to **6.2 Configuration of Timer Array Unit TAUS**.

7.3 Registers Controlling Timer Array Unit TAUS and Inverter Control Function Block

The following registers control timer array unit TAUS and the inverter control function block.

- Peripheral enable register 2 (PER2)
- Timer clock select register 0 (TPS0)[Note]
- Timer mode register n (TMRn)[Note]
- Timer status register n (TSRn)[Note]
- Timer channel enable status register 0 (TE0)[Note]
- Timer channel start register 0 (TS0)[Note]
- Timer channel stop register 0 (TT0)[Note]
- Timer input select register 0 (TIS0) [Note]
- Timer output enable register 0 (TOE0)[Note]
- Timer output register 0 (TO0)[Note]
- Timer output level register 0 (TOL0)[Note]
- Timer output mode register 0 (TOM0)[Note]
- Timer triangle wave output mode register 0 (TOT0)
- Timer dead time output enable register 0 (TDE0)
- Timer real-time output register 0 (TRO0)
- Timer real-time output enable register 0 (TRE0)
- Timer real-time control register 0 (TRC0)
- Timer modulation output enable register 0 (TME0)
- TAU option mode register (OPMR)
- TAU option status register (OPSR)
- TAU option Hi-Z start trigger register (OPHS)
- TAU option Hi-Z stop trigger register (OPHT)
- Input switch control register (ISC)[Note]
- Noise filter enable registers 1, 2 (NFEN1, NFEN2)[Note]
- Port mode registers 1, 3, 5, 7 (PM1, PM3, PM5, PM7)[Note]
- Port registers 1, 3, 5, 7 (P1, P3, P5, P7)[Note]

Note The inverter control function can be achieved by adding option functions to timer array unit TAUS. In this chapter, only the registers to be used with the inverter control function are described. For other registers that are to be used in common with timer array unit TAUS, refer to **CHAPTER 6 TIMER ARRAY UNIT TAUS**.

Remark n = 00 to 11

(1) Peripheral enable register 2 (PER2)

PER2 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When using timer array unit TAUS and the inverter control function, be sure to set bit 0 (TAU0EN) and bit 1 (TAUOPEN) of this register to 1.

PER2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Cautions 1. When setting timer array unit TAUS and the inverter control function, be sure to set TAU0EN and TAUOPEN to 1 first. If TAU0EN and TAUOPEN are set to 0, writing to a control register of timer array unit TAUS and the inverter control function is ignored, and all read values are default values.

2. Be sure to clear bits 2 to 7 of the PER2 register to 0.

Figure 7-3. Format of Peripheral Enable Register 2 (PER2)

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TAUOPEN	TAU0EN

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

TAUOPEN	Control of inverter control block input clock
0	Stops supply of input clock. • SFR used by the inverter control block cannot be written. • The inverter control block is in the reset status.
1	Supplies input clock. • SFR used by the inverter control block can be read/written.

(2) Timer clock select register 0 (TPS0)

(3) Timer mode register n (TMRn)

(4) Timer status register n (TSRn)

(5) Timer channel enable status register 0 (TE0)

(6) Timer channel start register 0 (TS0)

(7) Timer channel stop register 0 (TT0)

(8) Timer input select register 0 (TIS0)

(9) Timer output enable register 0 (TOE0)

(10) Timer output register 0 (TO0)

(11) Timer output level register 0 (TOL0)

(12) Timer output mode register 0 (TOM0)

The above-mentioned registers (2) to (12) are used in common with timer array unit TAUS.
For details, refer to **6.2 Configuration of Timer Array Unit TAUS**.

(13) Timer triangle wave output mode register 0 (TOT0)

TOT0 controls the timer output mode of the channel that is set by setting TOMn of the TOM0 register to 1.

The setting of each channel n by this register is reflected at the timing that the timer output signal is set or reset, when TREn and TMEn are set to 0 or TREn and TMEn are set to 1, when timer output is enabled (TOEn = 1) and TOMn is set to 1.

TOT0 can be rewritten when timer operation is stopped (TE0 = 0).

TOT0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-4. Format of Timer Triangle Wave Output Mode Register 0 (TOT0)

Address: F01E8H, F01E9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOT0	0	0	0	0	TOT 11	TOT 10	TOT 09	TOT 08	TOT 07	TOT 06	TOT 05	TOT 04	TOT 03	TOT 02	TOT 01	TOT 00

TOT n	Selection of combination-operation mode of channel n
0	Sets by a master channel timer interrupt request signal (INTTMn) and resets by a slave channel timer interrupt request signal (INTTMm).
1	Sets by a timer interrupt request signal during a down status (INTTMn) and resets by a timer interrupt request signal during an up status (INTTMm)Note.

Note Set the slave channel to TOTn = 1 when triangle wave PWM has been generated.

Caution Be sure to set bits 15 to 12 to 0.

Remark n: Channel number, m: Slave channel number
n = 00 to 11 (master channel: n = 00, 02, 04, 06, 08, 10)
n < m ≤ 11

(14) Timer real-time output enable register 0 (TRE0)

TRE0 enables or stops the timer output of each channel when the real-time output function is used.

TRE0 can be rewritten when timer operation is stopped (TE0 = 0).

TRE0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-5. Format of Timer Real-Time Output Enable Register 0 (TRE0)

Address: F01EAH, F01EBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRE0	0	0	0	0	TRE11	TRE10	TRE09	TRE08	TRE07	TRE06	TRE05	TRE04	TRE03	TRE02	TRE01	TRE00

TREn	Selection of combination-operation mode of channel n
0	Stops real-time output.
1	Enables real-time output.

Caution Be sure to set bits 15 to 12 to 0.

Remark n = 00 to 11

(15) Timer real-time output register 0 (TRO0)

TRO0 is a timer output buffer register for the real-time output function. The value of each bit of this register is output from the timer output pin (TOn) of each channel when the real-time output is enabled.

The TRO0 setting does not affect timer operation when the real-time output is stopped (TREn = 0). The TOn pin output is changed by the real-time output and timer operation when the real-time output is enabled (TREn = 1).

TRO0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-6. Format of Timer Real-Time Output Register 0 (TRO0)

Address: F01ECH, F01EDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRO0	0	0	0	0	TRO11	TRO10	TRO09	TRO08	TRO07	TRO06	TRO05	TRO04	TRO03	TRO02	TRO01	TRO00

TROn	Real-time output level
0	Low level
1	High level

Caution Be sure to set bits 15 to 12 to 0.

Remark n = 00 to 11

(16) Timer real-time control register 0 (TRC0)

TRC0 sets the channel generated by the real-time output trigger.

TRC0 can be rewritten when timer operation is stopped (TE0 = 0).

TRC0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-7. Format of Timer Real-Time Control Register 0 (TRC0)

Address: F01EEH, F01EFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRC0	0	0	0	0	TRC11	TRC10	TRC09	TRC08	TRC07	TRC06	TRC05	TRC04	TRC03	TRC02	TRC01	TRC00

TRCn	Selection of real-time output trigger function
0	Does not operate as a real-time output trigger generation channel. The timer interrupt request signal (INTTMn) of the channel that is set to TRCn = 1 at a higher channel becomes the real-time output trigger.
1	Operates as a real-time output trigger generation channel. The timer interrupt request signal (INTTMn) of the channel becomes the real-time output trigger of a lower channel.

Caution Be sure to set bits 15 to 12 to 0.

Remark n = 00 to 11

(17) Timer dead time output enable register 0 (TDE0)

TDE0 enables or disables dead time control of the timer output of each channel.

The setting of each channel n by this register is reflected at the timing that the timer output signal is set or reset, when timer output is enabled (TOEn = 1), and TOMn and TOTn are set to 1.

The TDEn value must be set the same for the even-number channel and odd-number channel for which dead time control is performed, because dead time control is performed for a set of an even-number channel and odd-number channel (even-number channel + 1).

TDE0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-8. Format of Timer Dead Time Output Enable Register 0 (TDE0)

Address: F01F2H, F01F3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDE0	0	0	0	0	TDE11	TDE10	TDE09	TDE08	TDE07	TDE06	TDE05	TDE04	TDE03	TDE02	TDE01	TDE00

TDEn	Selection of dead time control
0	Disables dead time control.
1	Enables dead time control.

Caution Be sure to set bits 15 to 12 to 0.

Remark n = 00 to 11

(18) Timer modulation output enable register 0 (TME0)

TME0 enables or disables the operation of the modulated-output function of timer output and real-time output.

TME0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-9. Format of Timer Modulation Output Enable Register 0 (TME0)

Address: F01F0H, F01F1H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TME0	0	0	0	0	TME11	TME10	TME09	TME08	TME07	TME06	TME05	TME04	TME03	TME02	TME01	TME00

TMEn	Real-time output level
0	Disables modulated output of timer output and real-time output.
1	Enables modulated output of timer output and real-time output.

Caution Be sure to set bits 15 to 12 to 0.

Remark n = 00 to 11

(19) TAU option mode register (OPMR)

OPMR sets the operation mode of the inverter control function option unit.

OPMR can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-10. Format of TAU Option Mode Register (OPMR) (1/2)

Address: F0220H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPMR	OPM	HPS	HSM	HDM	ATS 3	ATS 2	ATS 1	ATS 0	TLS 7	TLS 6	TLS 5	TLS 4	TLS 3	TLS 2	HIS0 1	HIS 0

OPM	Operation mode selection
0	6-phase output control mode (TO02 to TO07 become Hi-Z control targets, and Hi-Z control and cancellation are set by the HDM bit.)
1	Half-bridge output control mode (when channel 0 and channel 4 are the period registers) (TO02 and TO03 are set to Hi-Z by the TMOFF0 pin or internal comparator CMP0. TO06 and TO07 are set to Hi-Z by the TMOFF1 pin or internal comparator CMP1. A Hi-Z state is cancelled by the HSM bit.)

HPS	Hi-Z input pin selection
0	Uses the TMOFF0 and TMOFF1 pins as the Hi-Z control signal.
1	Uses the internal comparator output signal as the Hi-Z control signal.

HSM	Hi-Z cancellation method selection (when OPM = 1)
0	A Hi-Z state can be cancelled in synchronization with the period after the inactive edge of an internal comparator (CMP0/CMP1) or TMOFF0, TMOFF1 is detected.
1	A Hi-Z state can be cancelled in synchronization with the period after the edge by a software write is detected.

HDM	Hi-Z cancellation method selection (when OPM = 0)
0	2-stage overcurrent detection mode (A Hi-Z state is set when the active edge of internal comparator 0 (CMP0 side) or TMOFF0 is detected, and the Hi-Z state is cancelled in synchronization with the period after an inactive edge is detected. Furthermore, a Hi-Z state is set when the active edge of internal comparator 1 (CMP1 side) or TMOFF1 is detected, and the Hi-Z state is cancelled in synchronization with the period after the edge by a software write is detected.)
1	Overcurrent/electromotive force detection mode (A Hi-Z state is set by reversing the internal comparator 0 output or reversing the TMOFF0 active edge detection, and thus detecting the overcurrent side (high-potential CMP1 or TMOFF1) and electromotive force side (low-potential CMP0 or TMOFF0). The Hi-Z state can be cancelled in synchronization with the period after inactive edge detection of an internal comparator or TMOFF0, TMOFF1.)

Figure 7-10. Format of TAU Option Mode Register (OPMR) (2/2)

Address: F0220H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPMR	OPM	HPS	HSM	HDM	ATS 3	ATS 2	ATS 1	ATS 0	TLS 7	TLS 6	TLS 5	TLS 4	TLS 3	TLS 2	HIS 1	HIS 0

ATS 3	ATS 2	A/D1 conversion output trigger selection
		OPM = 0: uses channel 9 for interrupt generation
		OPM = 1: uses channel 5 for interrupt generation

0	0	Generates an A/D trigger for the match interrupt during a down status period of the master channel.
0	1	Generates an A/D trigger for the match interrupt during an up status period of the master channel.
1	0	Match interrupt during an up or a down status period of the master channel
1	1	Match interrupt during an up or a down status period of the master channel + valley interrupt of the master channel

ATS 1	ATS 0	A/D0 conversion output trigger selection
		OPM = 0: uses channel 8 for interrupt generation
		OPM = 1: uses channel 1 for interrupt generation

0	0	Generates an A/D trigger for the match interrupt during a down status period of the master channel.
0	1	Generates an A/D trigger for the match interrupt during an up status period of the master channel.
1	0	Match interrupt during an up or a down status period of the master channel
1	1	Match interrupt during an up or a down status period of the master channel + valley interrupt of the master channel

TLSn	Output reversal control
0	Performs forward output of timer output (TO0n).
1	Performs reverse output of timer output (TO0n).

HIS1	TMOFF1 valid edge selection
0	Sets the falling edge as valid.
1	Sets the rising edge as valid.

HIS0	TMOFF0 valid edge selection
0	Sets the falling edge as valid.
1	Sets the rising edge as valid.

Remark n = 2 to 7

(20) TAU option status register (OPSR)

OPSR displays various statuses of the motor control option unit.

OPSR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-11. Format of TAU Option Status Register (OPSR)

Address: F0222H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPSR	0	0	0	0	0	0	HZO F1	HZO F0	0	0	0	0	0	0	HZIF 1	HZIF 0

HZO F1	Hi-Z output signal 1 operation status
0	Normal status (timer output)
1	Hi-Z output status

HZO F0	Hi-Z output signal 0 operation status
0	Normal status (timer output)
1	Hi-Z output status

HZIF 1	TMOFF1 pin/internal comparator 0 output signal status
0	TMOFF1 pin is at low level when HPS of OPMR is 0. Internal comparator 1 output signal is at low level when HPS of OPMR is 1.
1	TMOFF1 pin is at high level when HPS of OPMR is 0. Internal comparator 1 output signal is at high level when HPS of OPMR is 1.

HZIF 0	TMOFF0 pin/internal comparator 0 output signal status
0	TMOFF0 pin is at low level when HPS of OPMR is 0. Internal comparator 0 output signal is at low level when HPS of OPMR is 1.
1	TMOFF0 pin is at high level when HPS of OPMR is 0. Internal comparator 0 output signal is at high level when HPS of OPMR is 1.

(21) TAU option Hi-Z start trigger register (OPHS)

OPHS sets the Hi-Z controller software trigger. Set this register to 1 to start the Hi-Z output of the TOn pin when the software trigger is set as valid.

OPHS can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-12. Format of TAU Option Hi-Z Start Trigger Register (OPHS)

Address: F0224H After reset: 0000H W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPHS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OPH S1	OPH S0

OPH S1	Hi-Z control output signal 1
0	–
1	High level (output port Hi-Z output)

OPH S0	Hi-Z control output signal 0
0	–
1	High level (output port Hi-Z output)

Caution Be sure to set bits 15 to 2 to 0.

Remark n = 02 to 07

(22) TAU option Hi-Z stop trigger register (OPHT)

OPHT sets the Hi-Z controller software trigger. Set this register to 1 to cancel the Hi-Z status of the TOn pin when the software trigger is set as valid.

OPHT can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 7-13. Format of TAU Option Hi-Z Stop Trigger Register (OPHT)

Address: F0226H After reset: 0000H W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPHT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OPHT1	OPHT0

OPHT1	Hi-Z control output signal 1
0	–
1	Low level (output port Hi-Z cancellation)

OPHT0	Hi-Z control output signal 0
0	–
1	Low level (output port Hi-Z cancellation)

Caution Be sure to set bits 15 to 2 to 0.

Remark n = 02 to 07

(23) Input switch control register (ISC)
(24) Noise filter enable registers 1, 2 (NFEN1, NFEN2)
(25) Port mode registers 1, 3, 5, 7 (PM1, PM3, PM5, PM7)

The above-mentioned registers (23) to (25) are used in common with timer array unit TAUS.
For details, refer to **6.2 Configuration of Timer Array Unit TAUS**.

7.4 Operation Using Inverter Control Function

7.4.1 Operation as real-time output function (type 1)

The values of TROn and TROm can be output from TOn and TOm by using the INTTMn output of the real-time output trigger generation channel.

This function is an extension of the function described in **6.7.1 Operation as interval timer/square wave output**.

The real-time output trigger generation channel (channel to which TRCn = 1 is set) outputs INTTMn at a fixed interval and generates a real-time output trigger.

The real-time output channel (channel to which TRCm = 0 is set) outputs the set value of TROm from TOm by the real-time output trigger.

The interrupt generation period can be calculated by the following expression.

$$\boxed{\text{INTTMn generation period} = \text{Count clock period} \times (\text{Set value of TDRn} + 1)}$$

The channel to which TRCn = 1 was set becomes the real-time output trigger generation channel and operates in the interval timer mode.

TCRn loads the value of TDRn at the first count clock, after the channel start trigger bit (TSn) is set to 1. At this time, INTTMn is not output and TOn is not toggled when MDn0 of TMRn is 0. INTTMn is output and TOn is toggled when MDn0 of TMRn is 1.

Afterward, TCRn counts down along with the count clock.

When TCRn has become 0000H, INTTMn is output and TOn is toggled upon the next count clock. TCRn loads the value of TDRn again at the same timing. Similar operation is continued hereafter.

The set value of TROn is output from TOn at the INTTMn output timing of the real-time output trigger generation channel.

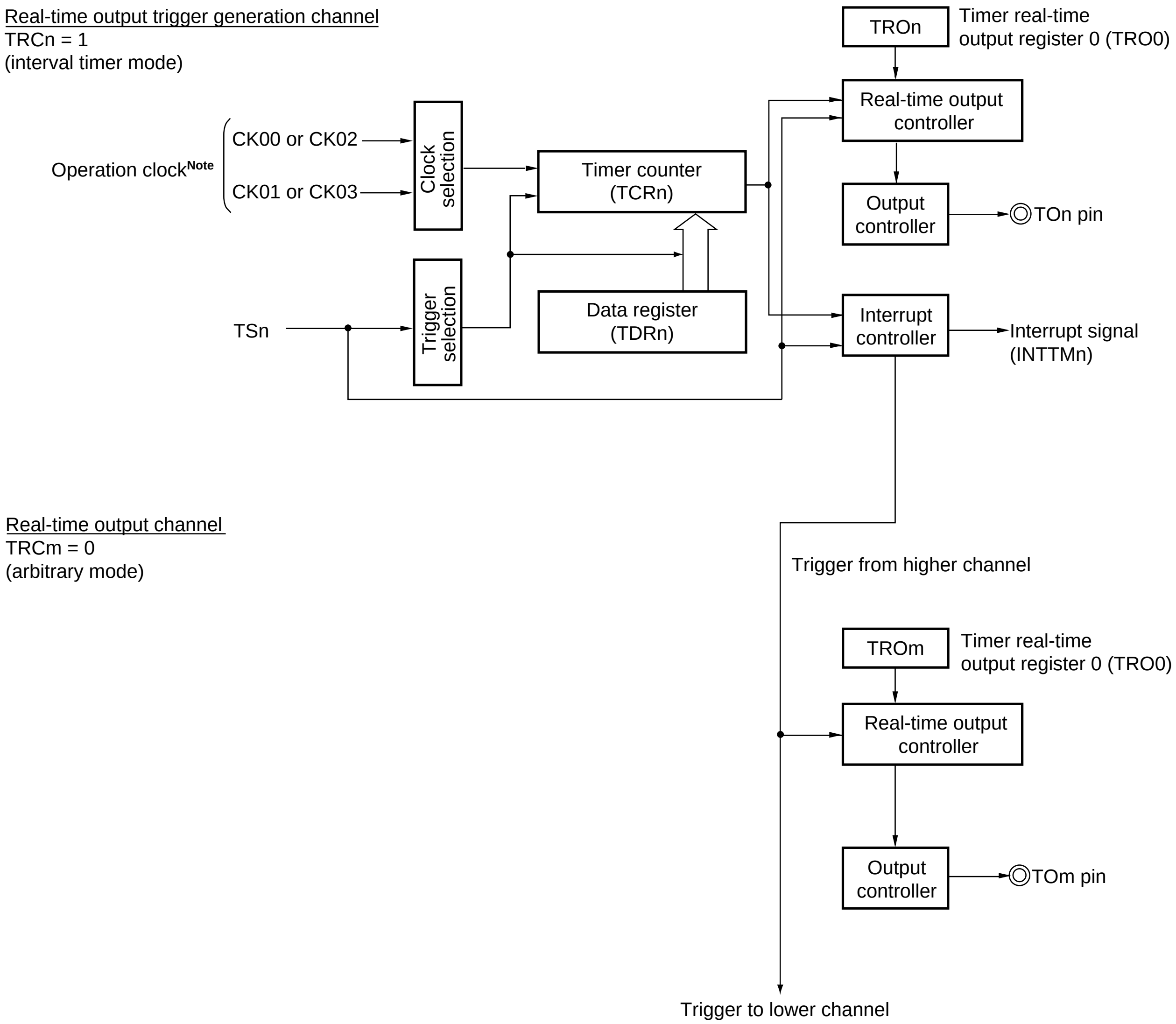
TOm of the lower channel (real-time output channel (TRCm = 0)) of the real-time output trigger generation channel (TRCn = 1) is controlled by the TREm and TRCm bits. The TOm output level will not change by only rewriting TROm.

When TREm of the real-time output channel (TRCm = 0) is 1, TOm outputs the set value of TROm at the INTTMn output timing of the real-time output trigger generation channel. When TREm or TRCm of the lower channel is 0 or 1, TOm is not toggled at the INTTMn output timing of the real-time output trigger generation channel.

When this function is used, TCRm, TDRm, and INTTMm of the lower channel can be operated as different functions.

Remark n = 01 to 10
m = 02 to 11

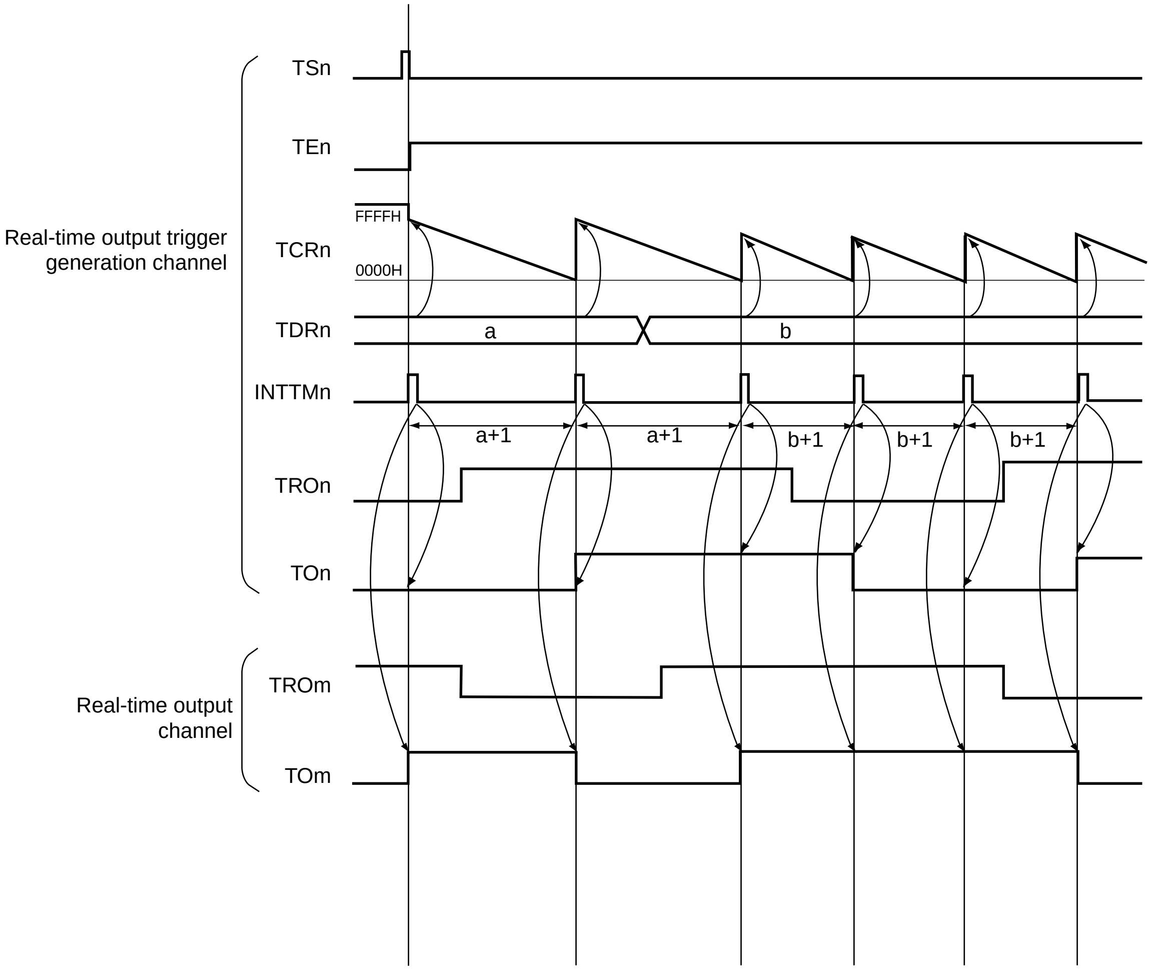
Figure 7-14. Block Diagram of Operation as Real-Time Output Function (Type 1)



Note The operation clocks of channels 0 to 7 are selected from CK00 and CK01, and those of channels 8 to 11 from CK02 and CK03.

Remark n = 01 to 10
m = 02 to 11

Figure 7-15. Example of Basic Timing of Operation as Real-Time Output Function (Type 1) (MDn0 = 1)



Remark n = 01 to 10

m = 02 to 11

Figure 7-16. Example of Set Contents of Registers During Operation as Real-Time Output Function (Type 1) (1/2)

(a) Timer mode register n (TMRn) of real-time output trigger generation channel (TRCn = 1)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel n
0000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMn nor inverts timer output when counting is started.
1: Generates INTTMn and inverts timer output when counting is started.

Selection of TIn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel n.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channel n.

(b) Other registers of real-time output trigger generation channel (TRCn = 1)

TOE0:TOEn	0: Stops the TOn output operation by counting operation. 1: Enables the TOn output operation by counting operation.
TO0:TOn	0: Outputs a low level from TOn. 1: Outputs a high level from TOn.
TOM0:TOMn	0: Toggle mode
TOT0:TOTn	0: Sets 0 when TOMn = 0 (toggle mode).
TOL0:TOLn	0: Sets 0 when TOMn = 0 (toggle mode).
TDE0:TDEn	0: Stops dead time control.
TRE0:TREn	0: Stops real-time output. 1: Enables real-time output.
TRO0:TROn	0: Outputs a low level as real-time output. 1: Outputs a high level as real-time output.
TRC0:TRCn	1: Operates as the real-time output trigger generation channel.
TME0:TMEn	0: Stops modulated output.

Remark n = 01 to 10

Figure 7-16. Example of Set Contents of Registers During Operation as Real-Time Output Function (Type 1) (2/2)

(c) Timer mode register m (TMRm) of real-time output channel (TRCm = 0)

With the real-time output function (type 1), TMRm of the channel when TRCm is set to 0 can be set arbitrarily.

(d) Other registers of real-time output channel (TRCm = 0)

TOE0:TOEm	0: Stops the TOm output operation by counting operation. 1: Enables the TOm output operation by counting operation.
TO0:TOm	0: Outputs a low level from TOm. 1: Outputs a high level from TOm.
TOM0:TOMm	0: Sets 0 when TREm = 1 (enables real-time output).
TOT0:TOTm	0: Sets 0 when TOMm = 0 (toggle mode).
TOL0:TOLm	0: Sets 0 when TOMm = 0 (toggle mode).
TDE0:TDEm	0: Stops dead time control.
TRE0:TREm	1: Enables real-time output.
TRO0:TROm	0: Outputs a low level as real-time output. 1: Outputs a high level as real-time output.
TRC0:TRCm	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEm	0: Stops modulated output.

Remark m = 02 to 11

Figure 7-17. Operation Procedure of Real-Time Output Function (Type 1) (1/2)

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN and TAUOPEN bits of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	[Real-time output trigger generation channel (TRCn = 1)] Sets the TMRn register (determines operation mode of channel). Sets interval (period) value to the TDRn register. Sets the TRCm bit to 1 (trigger generation channel). Sets the TREm bit to 1 (real-time output enable). [Real-time output channel (TRCm = 0)] Sets the TRCm bit to 0 (non-trigger generation channel). Sets the TREm bit to 1 (real-time output enable).	Channel stops operating. (Clock is supplied and some power is consumed.) The TOn and TOm pins go into Hi-Z output state.
	Sets the TOEn and TOEm bits to 1 and enables output of TOn and TOm. Sets the port to output mode.	TOn and TOm do not change because channel has stopped operating. The TOn and TOm pins of the product output the TOn and TOm set levels.
Operation start	Sets the TOEn and TOEm bits to 1 (only when operation is resumed). Sets the TSn bit of the trigger generation channel to 1. The TSn bit automatically returns to 0 because it is a trigger bit.	[Real-time output trigger generation channel (TRCn = 1)] TEn = 1, and count operation starts. Value of TDRn is loaded to TCRn at the count clock input. INTTMn is generated if the MDn0 bit of the TMRn register is 1.
During operation	Set value of the TDRn register can be changed. The TCRn register can always be read. Set values of the TROn and TROm bits can be changed.	Counter (TCRn) counts down. When count value reaches 0000H, the value of TDRn is loaded to TCRn again and the count operation is continued. By detecting TCRn = 0000H, INTTMn is generated. After that, the above operation is repeated. The set value of TROm of the real-time output channel is output from TOm at the INTTMn output timing.
Operation stop	The TTn bit is set to 1. The TTn bit automatically returns to 0 because it is a trigger bit.	TEn = 0, and count operation stops. TCRn holds count value and stops. The TOn output is not initialized but holds current status and stops.
	The TOEn and TOEm bits are cleared to 0 and values are set to TOn and TOm.	The set values of TOn and TOm initialize the outputs of TOn and TOm.

Operation is resumed.

Remark n = 01 to 10, m = 02 to 11

Figure 7-17. Operation Procedure of Real-Time Output Function (Type 1) (2/2)

	Software Operation	Hardware Status
TAUS stop	To hold the TOn and TOm pin output levels Clears the TOn and TOm bits to 0 after the values to be held are set to the port register.	The TOn and TOm pin output levels are held by port function.
	When holding the TOn and TOm pin output levels is not necessary Switches the port mode register to input mode.	The TOn and TOm pin output levels go into Hi-Z output state.
	The TAU0EN and TAUOPEN bits of the PER2 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOn and TOm bits are cleared to 0 and the TOn and TOm pins are set to port mode.)

Remark n = 01 to 10, m = 02 to 11

7.4.2 Operation as real-time output function (type 2)

The values of TROn and TROm can be output from TOn and TOm by using the INTTMn output of the real-time output trigger generation channel. Real-time output by external pin input edge detection or software input detection can be performed.

The real-time output function (type 2) is an extension of the function described in **6.7.4 Operation as input pulse interval measurement**.

The real-time output trigger generation channel (channel to which TRCn = 1 is set) outputs INTTMn and generates a real-time output trigger by valid edge detection of the TIn pin input or setting 1 to the channel start trigger (TSn).

The real-time output channel (channel to which TRCm = 0 is set) outputs the set value of TROm from TOm by the real-time output trigger. The TCRn value is captured to TDRn at the INTTMn generation timing, but the TDRn value has no meaning.

The channel to which TRCn = 1 was set becomes the real-time output trigger generation channel and operates as an up counter in the capture mode.

TCRn starts counting up from 0000H along with the count clock, when the channel start trigger (TSn) is set to 1 while TEn is 0 or when the valid edge of the TIn input is detected.

At this time, INTTMn is not output and TOn is not toggled when MDn0 of the TMRn register is 0. INTTMn is output and TOn is toggled when MDn0 of the TMRn register is 1.

The counter (TCRn) is cleared to 0000H and INTTMn is output at the same time the count value is transferred (captured) to TDRn, when the valid edge of the TIn pin input is detected or when the channel start trigger (TSn) is set to 1.

The set value of TROn is output from TOn at the INTTMn output timing of the real-time output trigger generation channel.

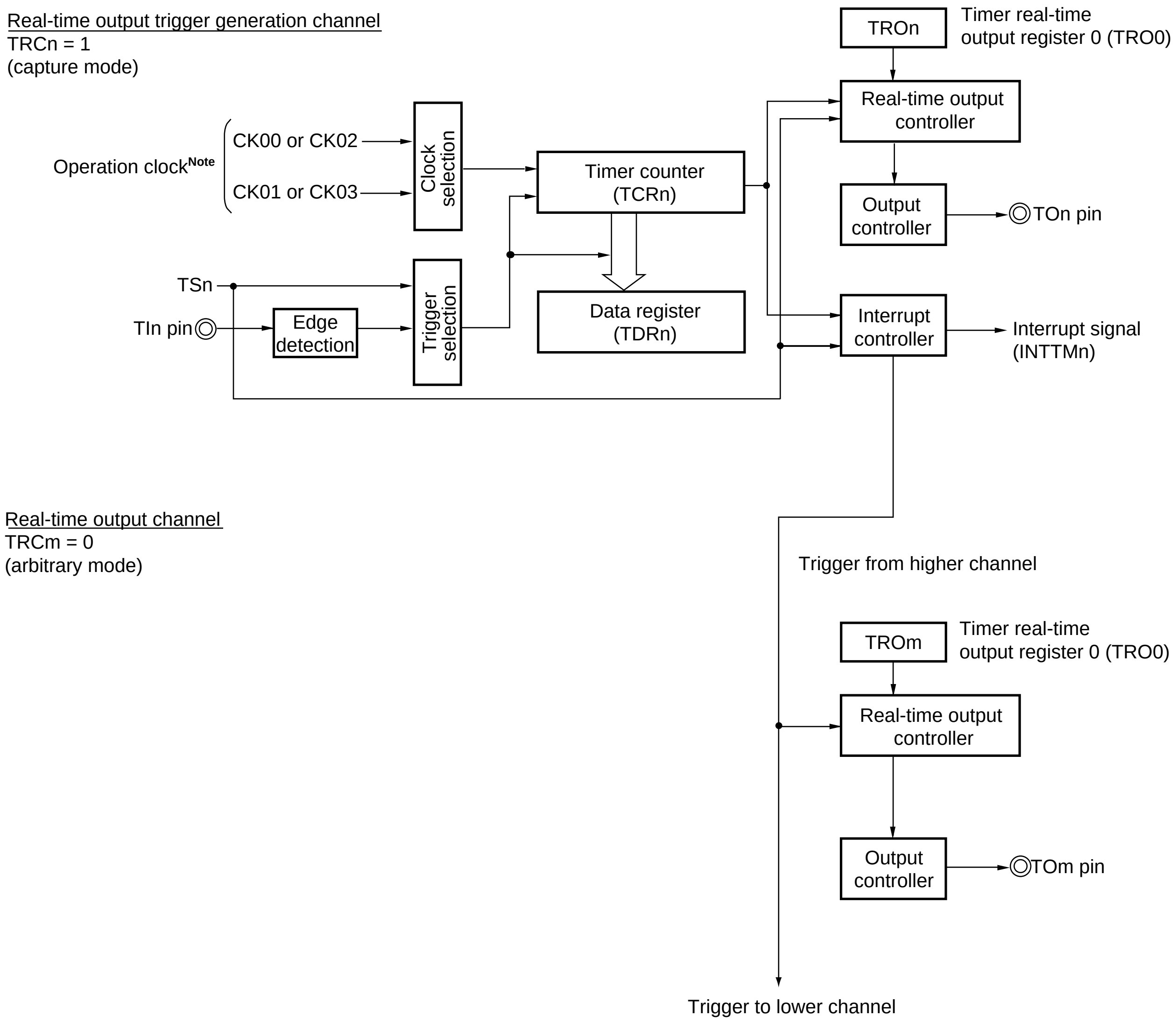
TOm of the lower channel (real-time output channel) of the real-time output trigger generation channel (TRCn = 1) is controlled by the TREm bit. The TOm output level will not change by only rewriting TROm.

When TREm of the real-time output channel (TRCm = 0) is 1, TOm outputs the set value of TROm at the INTTMn output timing of the real-time output trigger generation channel. When TREm or TRCm of the lower channel is 0 or 1, real-time output is not performed at the INTTMn output timing of the real-time output trigger generation channel.

When the real-time output function (type 2) is used, TCRm, TDRm, and INTTMn of the lower channel can be operated as different functions.

Remark n = 01 to 10
m = 02 to 11

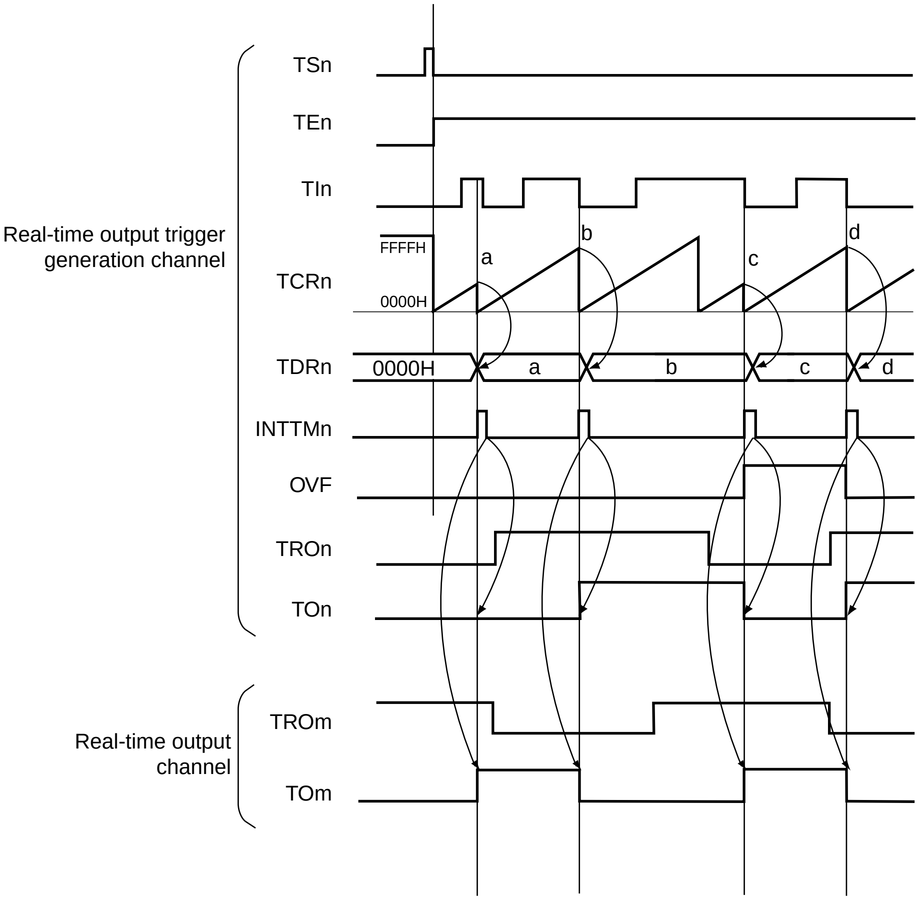
Figure 7-18. Block Diagram of Operation as Real-Time Output Function (Type 2)



Note The operation clocks of channels 0 to 7 are selected from CK00 and CK01, and those of channels 8 to 11 from CK02 and CK03.

Remark n = 01 to 10
m = 02 to 11

Figure 7-19. Example of Basic Timing of Operation as Real-Time Output Function (Type 2) (MDn0 = 1)



Remark n = 01 to 10
m = 02 to 11

Figure 7-20. Example of Set Contents of Registers During Operation as Real-Time Output Function (Type 2) (1/2)

(a) Timer mode register n (TMRn) of real-time output trigger generation channel (TRCn = 1)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	0	0	0	0	1	1/0	1/0	0	0	0	1	0	1/0

Operation mode of channel n
0010B: Capture mode

Setting of operation when counting is started
0: Neither generates INTTMn nor inverts timer output when counting is started.
1: Generates INTTMn and inverts timer output when counting is started.

Selection of TIn pin input edge
00B: Falling edge detection
01B: Rising edge detection
10B: Both-edge detection
11B: Setting prohibited

Capture trigger selection
001B: Selects valid edge of TIn pin input.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channels 1 to 7.
1: Selects CK01 (channels 0 to 7) or CK03 (channels 8 to 11) as operation clock of channels 1 to 7.

(b) Other registers of real-time output trigger generation channel (TRCn = 1)

TOE0:TOEn	0: Stops the TOn output operation by counting operation. 1: Enables the TOn output operation by counting operation.
TO0:TOn	0: Outputs a low level from TOn. 1: Outputs a high level from TOn.
TOM0:TOMn	0: Toggle mode
TOT0:TOTn	0: Sets 0 when TOMn = 0 (toggle mode).
TOL0:TOLn	0: Sets 0 when TOMn = 0 (toggle mode).
TDE0:TDEn	0: Stops dead time control.
TRE0:TREn	0: Stops real-time output. 1: Enables real-time output.
TRO0:TROn	0: Outputs a low level as real-time output. 1: Outputs a high level as real-time output.
TRC0:TRCn	1: Operates as the real-time output trigger generation channel.
TME0:TMEn	0: Stops modulated output.

Remark n = 01 to 10

Figure 7-20. Example of Set Contents of Registers During Operation as Real-Time Output Function (Type 2) (2/2)

(c) Timer mode register m (TMRm) of real-time output channel (TRCm = 0)

With the real-time output function (type 2), TMRm of the channel when TRCm is set to 0 can be set arbitrarily.

(d) Other registers of real-time output channel (TRCm = 0)

TOE0:TOEm	0: Stops the TOm output operation by counting operation. 1: Enables the TOm output operation by counting operation.
TO0:TOm	0: Outputs a low level from TOm. 1: Outputs a high level from TOm.
TOM0:TOMm	0: Sets 0 when TREm = 1 (enables real-time output).
TOT0:TOTm	0: Sets 0 when TOMm = 0 (toggle mode).
TOL0:TOLm	0: Sets 0 when TOMm = 0 (toggle mode).
TDE0:TDEm	0: Stops dead time control.
TRE0:TREm	1: Enables real-time output.
TRO0:TROm	0: Outputs a low level as real-time output. 1: Outputs a high level as real-time output.
TRC0:TRCm	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEm	0: Stops modulated output.

Remark m = 02 to 11

Figure 7-21. Operation Procedure of Real-Time Output Function (Type 2) (1/2)

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN and TAUOPEN bits of the PER2 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	Real-time output trigger generation channel (TRCn = 1) Sets the TMRn register (determines operation mode of channel). Sets the TRCm bit to 1 (trigger generation channel). Sets the TREm bit to 1 (real-time output enable). Real-time output channel (TRCm = 0) Sets the TRCm bit to 0 (non-trigger generation channel). Sets the TREm bit to 1 (real-time output enable).	Channel stops operating. (Clock is supplied and some power is consumed.) The TOn and TOm pins go into Hi-Z output state.
	Sets the TOEn and TOEm bits to 1 and enables output of TOn and TOm. Clears the port register and port mode register to 0. →	TOn and TOm do not change because channel has stopped operating. The TOn and TOm pins of the product output the TOn and TOm set levels.
Operation start	Sets the TOEn and TOEm bits to 1 (only when operation is resumed). Sets the TSn bit of the trigger generation channel to 1. → The TSn bit automatically returns to 0 because it is a trigger bit.	Real-time output trigger generation channel (TRCn = 1) TEn = 1, and count operation starts. Clears TCRn to 0000H by count clock input. INTTMn is generated if the MDn0 bit of the TMRn register is 1.
During operation	The TMRn register can only change the set values of the CISn1 and CISn0 bits. Set values of the TROn and TROm bits can be changed.	Counter (TCRn) counts up from 0000H and transfers (captures) the count value to TDRn when the valid edge of the TIn pin input is detected. At the same time, TCRn is cleared to 0000H and INTTMn is generated. The OVF bit of the TSRn register is set or cleared when an overflow occurs or does not occur at this time. After that, the above operation is repeated. The set value of TROm of the real-time output channel is output from TOm at the INTTMn output timing.
Operation stop	The TTn bit is set to 1. → The TTn bit automatically returns to 0 because it is a trigger bit.	TEn = 0, and count operation stops. TCRn holds count value and stops. TCRn also holds the OVF bit of the TSRn register. The TOn output is not initialized but holds current status and stops.
	The TOEn and TOEm bits are cleared to 0 and values are set to the TOn and TOm bits. →	The TOn and TOm pins output the set levels of TOn and TOm.

Operation is resumed.

Remark n = 01 to 10
m = 02 to 11

Figure 7-21. Operation Procedure of Real-Time Output Function (Type 2) (2/2)

	Software Operation	Hardware Status
TAUS stop	To hold the TOn and TOm pin output levels	
	Clears the TOp and TOq bits to 0 after the values to be held are set to the port register.	The TOn and TOm pin output levels are held by port function.
	When holding the TOn and TOm pin output levels is not necessary	
	Switches the port mode register to input mode.	The TOn and TOm pin output levels go into Hi-Z output state.
	The TAU0EN and TAUOPEN bits of the PER2 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOn and TOm bits are cleared to 0 and the TOn and TOm pins are set to port mode.)

Remark n = 01 to 10
m = 02 to 11

7.4.3 Operation as 6-phase PWM output function

By extending the PWM function and using eight channels in combination, a 6-phase PWM waveform can be output.

A total of six PWM output signals, one each from slave channel 2, slave channel 3, slave channel 4, slave channel 5, slave channel 6, slave channel 7 are output. Slave channel 1 can be operated in any operation mode. (With this function, the operation mode of slave channel 1 will not be fixed.)

The period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR00 (master) + 1} × Count clock period
Duty [%] = {Set value of TDRm (slave)}/{Set value of TDR00 (master) + 1} × 100
0% output: Set value of TDRm (slave) = 0000H
100% output: Set value of TDRm (slave) ≥ Set value of TDR00 (master) + 1

Remark Although the duty value exceeds 100% if the set value of TDRm (slave) > {set value of TDR00 (master) + 1}, it is summarized into 100% output.

TCR00 of the master channel operates in the interval timer mode and counts the periods.

With the 6-phase PWM output function, the operation mode of slave channel 1 will not be fixed and can be set freely.

(To use the modulated-output function, slave channel 1 is used as the real-time output trigger generation channel.)

TCRm of slave channels 2 to 7 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOm pin. TCRm loads the value of TDRm to TCRm, using INTTM00 of the master channel as a start trigger, and starts counting down. When TCRm = 0000H, TCRm outputs INTTMm and stops counting until the next start trigger (INTTM00 of the master channel) has been input. The output level of TOm becomes active one count clock after generation of INTTM00 from the master channel, and inactive when TCRm = 0000H.

TDR00 and TDRm of the master channel and slave channel become valid from the next period (generation of INTTM00 of the master channel).

Caution To rewrite both TDR00 of the master channel and TDRm of slave channels 2 to 7, write access is necessary at least twice. Since the values of TDR00 and TDRm are loaded to TCR00 and TCRm after INTTM00 is generated from the master channel, if rewriting is performed separately before and after generation of INTTM00 from the master channel, the TOm pin cannot output the expected waveform. To rewrite both TDR00 of the master and TDRm of the slave, be sure to rewrite both the registers immediately after INTTM00 is generated from the master channel.

Remark m = 02 to 07

Figure 7-22. Block Diagram of Operation as 6-Phase PWM Output Function

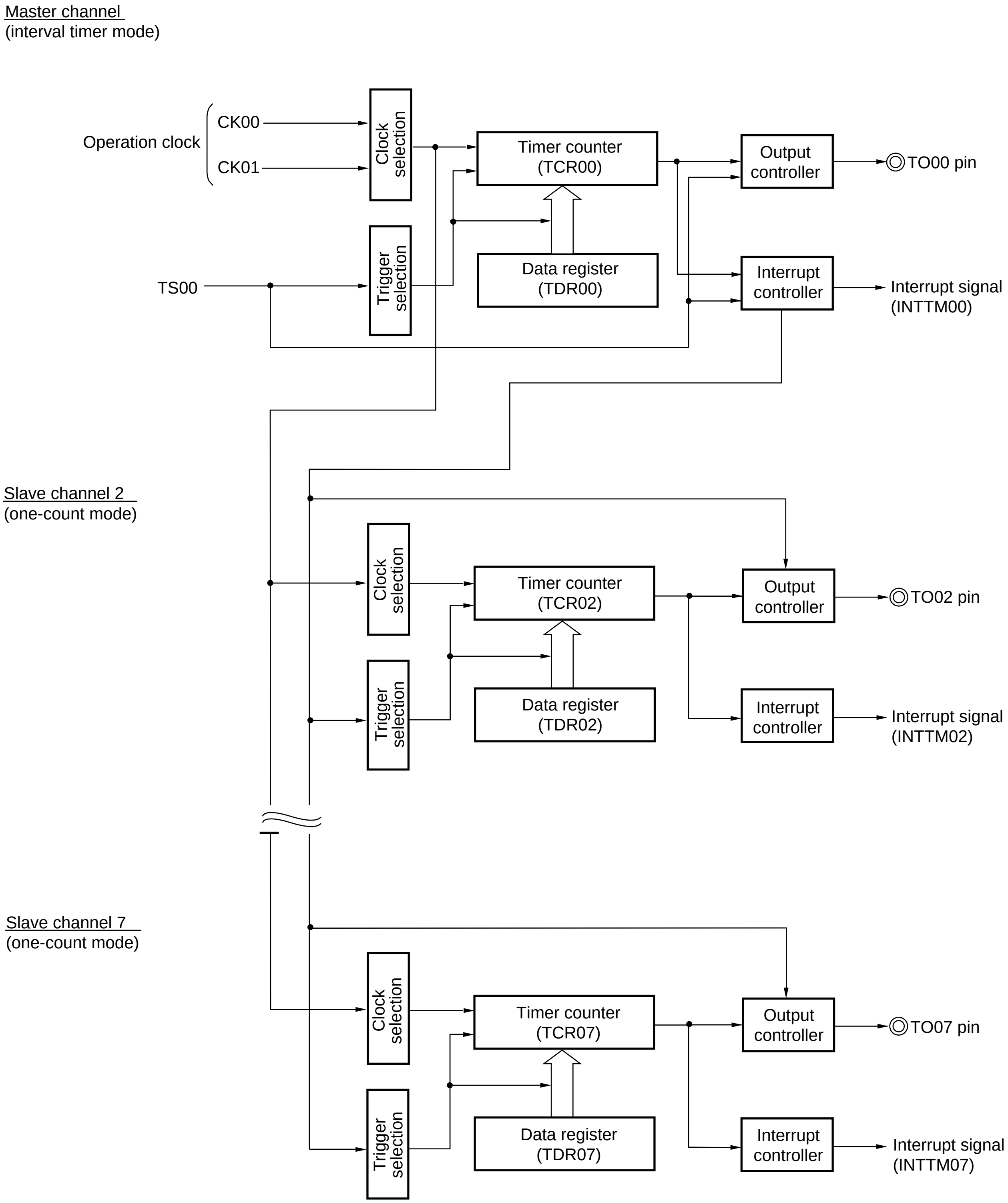


Figure 7-23. Example of Basic Timing of Operation as 6-Phase PWM Output Function

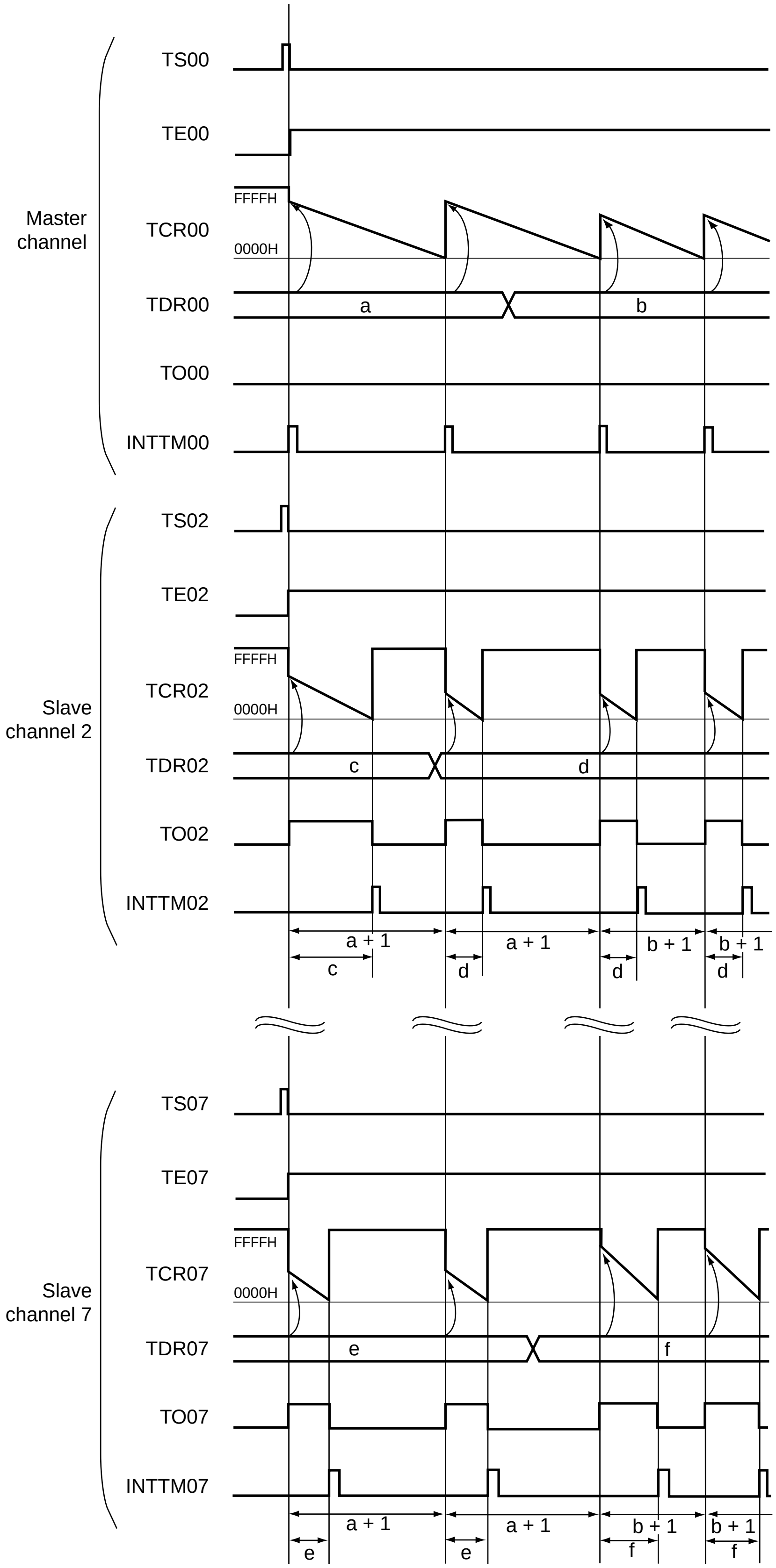
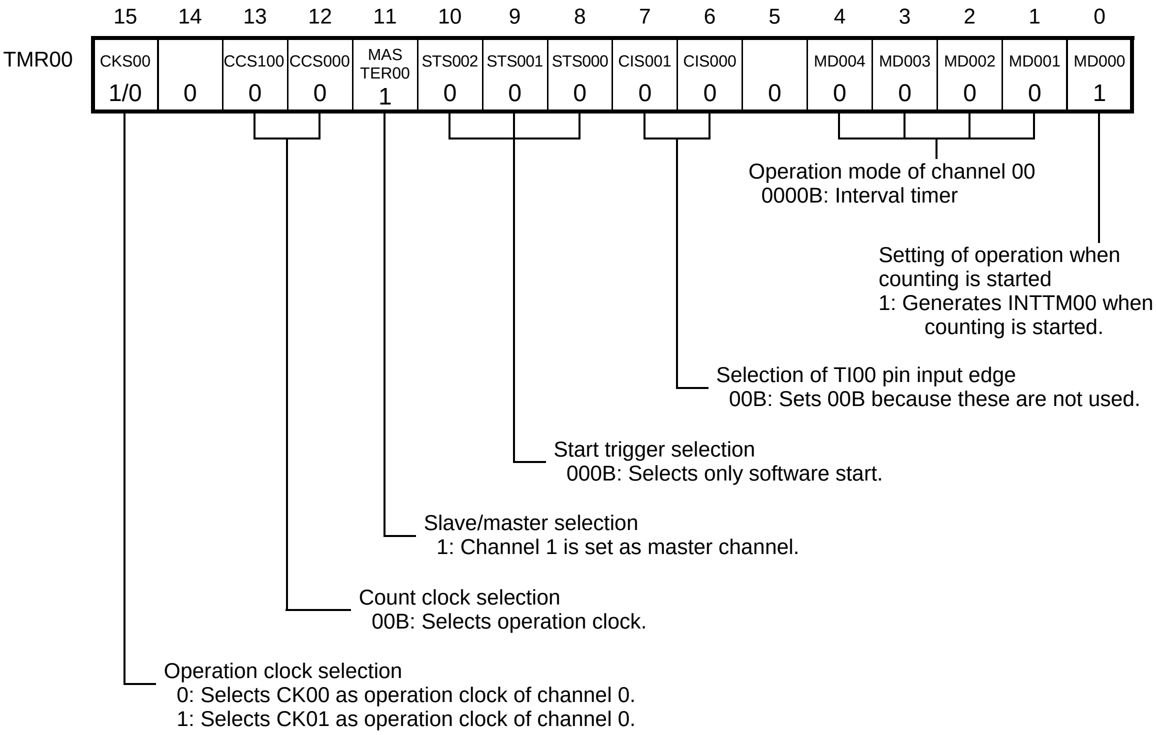


Figure 7-24. Example of Set Contents of Registers When 6-Phase PWM Output Function (Master Channel) Is Used

(a) Timer mode register 00 (TMR00)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR00	CKS00		CCS100	CCS000	MAS TER00	STS002	STS001	STS000	CIS001	CIS000		MD004	MD003	MD002	MD001	MD000
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel 00
0000B: Interval timer

Setting of operation when counting is started
1: Generates INTTM00 when counting is started.

Selection of TI00 pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
00B: Selects operation clock.

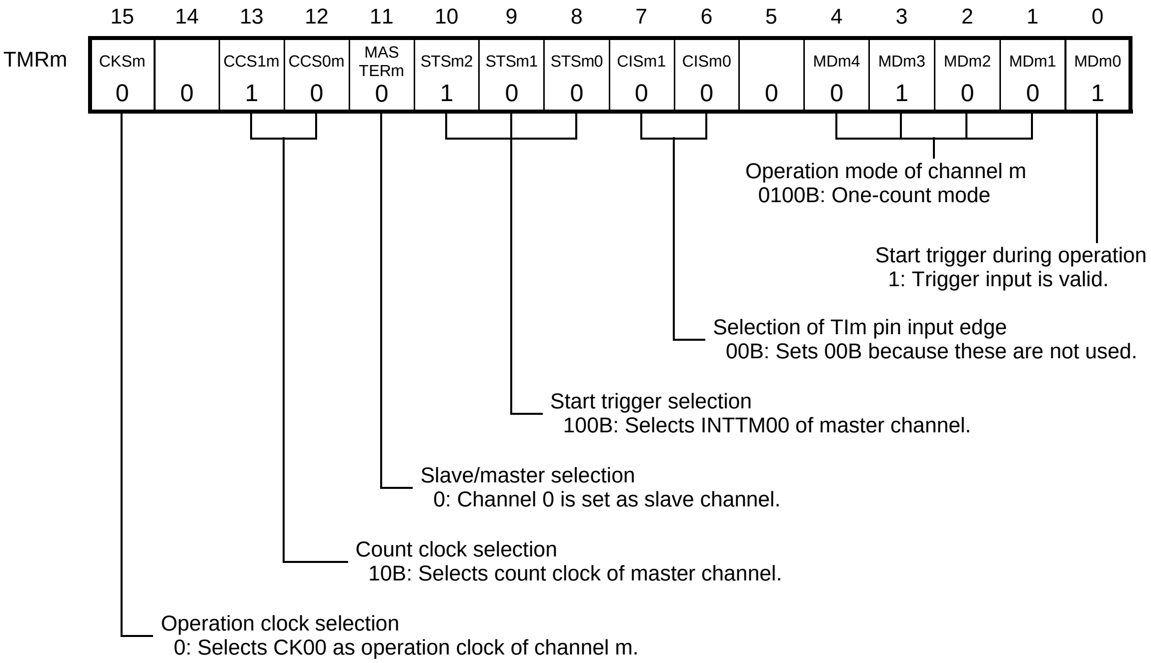
Operation clock selection
0: Selects CK00 as operation clock of channel 0.
1: Selects CK01 as operation clock of channel 0.

(b) Other registers

TOE0:TOE00	0: Stops the TO00 output operation by counting operation.
TO0:TO00	0: Outputs a low level from TO00.
TOM0:TOM00	0: Sets 0 when TOE00 = 0 (stops the TO00 output operation by counting operation).
TOT0:TOT00	0: Sets 0 when TOM00 = 0 (toggle mode).
TOL0:TOL00	0: Sets 0 when TOM00 = 0 (toggle mode).
TDE0:TDE00	0: Stops dead time control.
TRE0:TRE00	0: Stops real-time output.
TRO0:TRO00	0: Sets 0 when TRE00 = 0 (stops real-time output).
TRC0:TRC00	0: Does not operate as the real-time output trigger generation channel.
TME0:TME00	0: Stops modulated output.

Figure 7-25. Example of Set Contents of Registers When 6-Phase PWM Output Function (Slave Channels 2 to 7) Is Used

(a) Timer mode register m (TMRm)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRm	CKSm		CCS1m	CCS0m	MAS TERm	STSm2	STSm1	STSm0	CISm1	CISm0		MDm4	MDm3	MDm2	MDm1	MDm0
	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel m
0100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TIm pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTM00 of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
10B: Selects count clock of master channel.

Operation clock selection
0: Selects CK00 as operation clock of channel m.

(b) Other registers

TOE0:TOEm	0: Stops the TOm output operation by counting operation. 1: Enables the TOm output operation by counting operation.
TO0:TOm	0: Outputs a low level from TOm. 1: Outputs a high level from TOm.
TOM0:TOMm	1: Sets combination-operation mode.
TOT0:TOTm	0: Generates other than triangular wave PWM output.
TOL0:TOLm	0: Positive logic output (active-high) 1: Inverted output (active-low)
TDE0:TDEm	0: Stops dead time control.
TRE0:TREm	0: Stops real-time output.
TRO0:TROm	0: Sets 0 when TREm = 0 (stops real-time output).
TRC0:TRCm	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEm	0: Stops modulated output.

Remark m = 02 to 07

Figure 7-26. Operation Procedure When 6-Phase PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN and TAUOPEN bits of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR00 and TMRm registers of each channel to be used (determines operation mode of channels). An interval (period) value is set to the TDR00 register of the master channel, and a duty factor is set to the TDRm register of slave channels 2 to 7.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels 2 to 7. Sets the TOMm bit to 1 (combination-operation mode). Sets the TOTm bit to 0 (generates other than triangular wave PWM output). Sets the TOLm bit and determines the active level of the TOm output. Sets the TOm bit and determines default level of the TOm output. Sets the TOEm bit to 1 and enables operation of TOm. Clears the port register and port mode register to 0.	The TOm pin goes into Hi-Z output state. The TOm default setting level is output when the port mode register is in output mode and the port register is 0. TOm does not change because channel has stopped operating. The TOm pin outputs the TOm set level.
Operation start	Sets TOEm (slaves 2 to 7) to 1 (only when operation is resumed). The TS00 (master) and TSm (slaves 2 to 7) bits of the TS0 register are set to 1 at the same time. The TS00 and TSm bits automatically return to 0 because they are trigger bits.	TE00 = 1, TEm = 1 When the master channel starts counting, INTTM00 is generated. Triggered by this interrupt, the slave channels 2 to 7 also start counting.
During operation	Set values of the TDR00 and TDRm registers can be changed after INTTM00 of the master channel is generated. The TCR00 and TCRm registers can always be read. Set values of the TOL0, TO0, and TOE0 registers can be changed.	The counter of the master channel loads the TDR00 value to TCR00 and counts down. When the count value reaches TCR00 = 0000H, INTTM00 is generated. At the same time, the value of the TDR00 register is loaded to TCR00, and the counter starts counting down again. At slave channels 2 to 7, the values of the TDRm register are transferred to TCRm, triggered by INTTM00 of the master channel, and the counter starts counting down. The output levels of TOm become active one count clock after generation of the INTTM00 output from the master channel. It becomes inactive when TCRm = 0000H, and the counting operation is stopped. After that, the above operation is repeated.

Operation is resumed. (from next page)

Remark m = 02 to 07

Figure 7-26. Operation Procedure When 6-Phase PWM Output Function Is Used (2/2)

Operation is resumed.
(to forward page)

	Software Operation	Hardware Status
Operation stop	The TT00 (master) and TTm (slaves 2 to 7) bits are set to 1 at the same time. The TT00 and TTm bits automatically return to 0 because they are trigger bits.	TE00, TEm = 0, and count operation stops. TCR00 and TCRm hold count value and stops. The TOm output is not initialized but holds current status.
	The TOEm bits of slave channels 2 to 7 are cleared to 0 and value is set to the TOm bit.	The TOm pin outputs the TOm set level.
TAUS stop	To hold the TOm pin output level Clears the TOm bit to 0 after the value to be held is set to the port register. When holding the TOm pin output level is not necessary Switches the port mode register to input mode.	The TOm pin output level is held by port function. The TOm pin output level goes into Hi-Z output state.
	The TAU0EN and TAUOPEN bits of the PER2 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO00 and TOm bits are cleared to 0 and the TO00 and TOm pins are set to port mode.)

Remark m = 02 to 07

7.4.4 Operation as triangular wave PWM output function

Multiple channels can be used in combination to output a triangular wave PWM for motor control.

The period is set by the master channel and a triangular wave PWM is output by the slave channel. When multiple triangular wave PWMs are output for a period, the triangular wave PWM output can be added by adding a slave channel.

The output pulse period and duty factor can be calculated by the following expression.

Pulse period (down/up) = {Set value of TDR00 (master) + 1} × 2 × Count clock period

Duty factor [%] = {Set value of TDR00 (master) + 1 − Set value of TDRm (slave)}/{Set value of TDR00 (master) + 1} × 100

0% output: Set value of TDRm (slave) ≥ {Set value of TDR00 (master) + 1}

100% output: Set value of TDRm (slave) = 0000H

Remark Although the duty factor exceeds 0% if the set value of TDRm (slave) > {set value of TDR00 (master) + 1}, it is summarized into 0% output.

The master channel operates in the interval timer mode and counts the periods.

TCR00 loads the value of TDR00 at the first count clock, after the channel start trigger bit (TS00) is set to 1. At this time, INTTM00 is not output and TO00 is not toggled when MDn0 of TMR00 is 0. INTTM00 is output and TO00 is toggled when MDn0 of TMRn is 1.

Afterward, TCRn counts down along with the count clock.

When TCR00 has become 0000H, INTTM00 is output and TO00 is toggled upon the next count clock. TCR00 loads the value of TDR00 again at the same timing. Similar operation is continued hereafter.

A carrier period is generated in two periods of the master channel count.

The count operation of the slave channel is controlled by defining the first period of the master channel as a down status of the slave channel and the second period as an up status of the slave channel.

TO00 of the master channel outputs up and down statuses.

TO00 of the TO0 register must be manipulated while TOE00 of the TOE0 register is 0 and the default level must be set, because up and down statuses are output.

TO00 of TO0 is set to 1 when MD000 of the TMR00 register is 0, and TO00 is set to 0 when MD000 is 1.

By setting the default level, a high level is output from TO00 during a down status and a low level is output during an up status.

TCRm of the slave channel operates in the up and down count mode, and counts the duty.

TCRm loads the value of TDRm at the first count clock, after the channel start trigger bit (TSm) is set to 1. Hereafter, counting up and counting down is switched in accordance with the operation of the master channel. INTTMm is output when TCRm becomes 0000H.

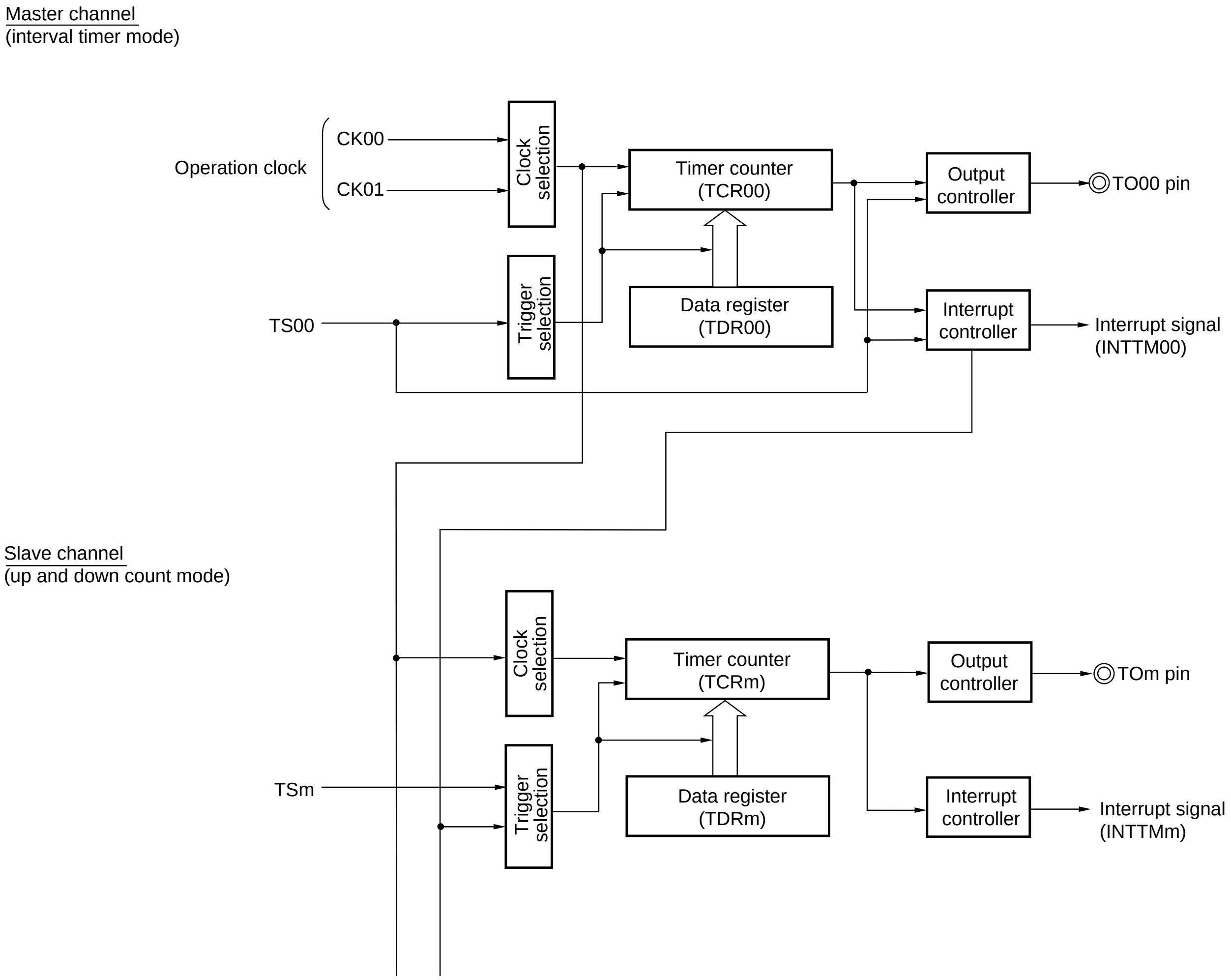
The TOm output becomes an active level when TCRm generates INTTMm while counting down, and it becomes an inactive level when TCRm generates INTTMm while counting up.

TCRm loads the value of TDRm again when INTTM00 is generated in an up status of the master channel. Similar operation is continued hereafter.

Remark m = 02 to 07

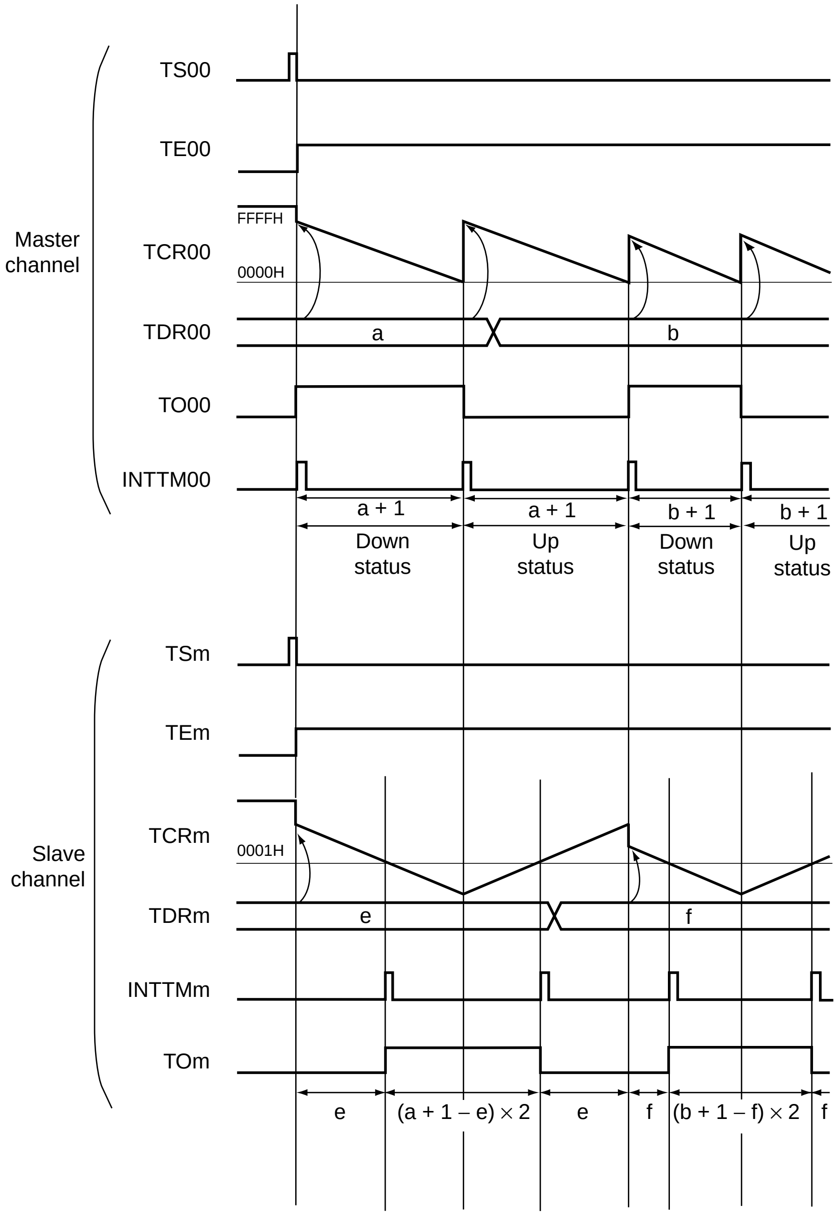
Caution TDR00 of the master channel must be rewritten during an up status period of the slave channel. When the value of TDR00 is rewritten separately during an up status and a down status, the periods of the up status and down status differ and the TO00 pin cannot output an expected waveform, because the value of TDR00 of the rewritten master channel becomes valid during the next down status period.

Figure 7-27. Block Diagram of Operation as Triangular Wave PWM Output Function



Remark m = 02 to 07

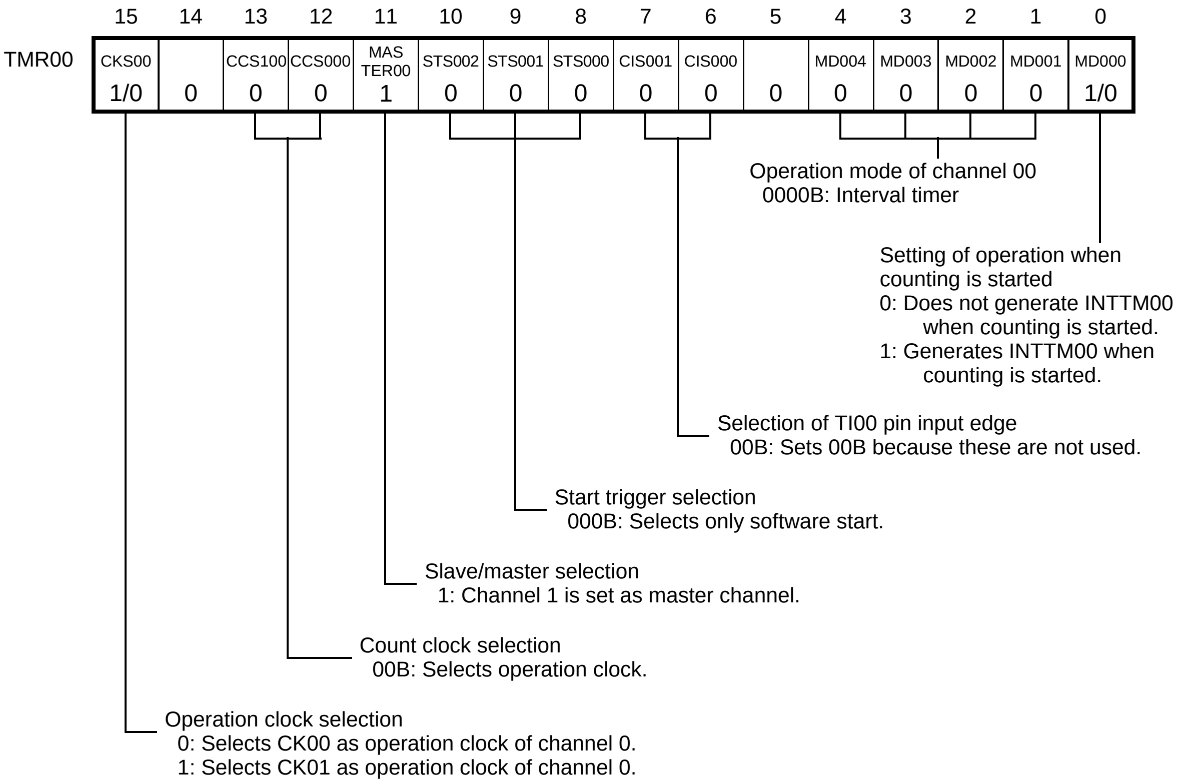
Figure 7-28. Example of Basic Timing of Operation as Triangular Wave PWM Output Function



Remark m = 02 to 07

Figure 7-29. Example of Set Contents of Registers When Triangular Wave PWM Output Function (Master Channel) Is Used

(a) Timer mode register 00 (TMR00)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR00	CKS00		CCS100	CCS000	MAS TER00	STS002	STS001	STS000	CIS001	CIS000		MD004	MD003	MD002	MD001	MD000
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel 00
0000B: Interval timer

Setting of operation when counting is started
0: Does not generate INTTM00 when counting is started.
1: Generates INTTM00 when counting is started.

Selection of TI00 pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
00B: Selects operation clock.

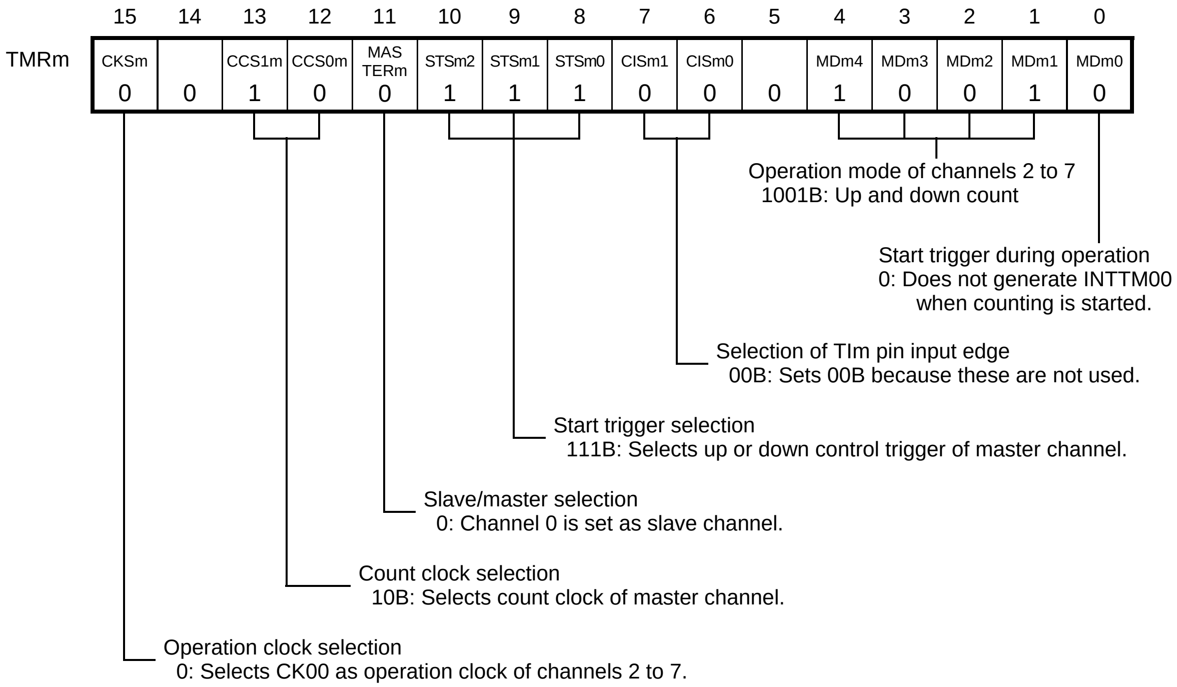
Operation clock selection
0: Selects CK00 as operation clock of channel 0.
1: Selects CK01 as operation clock of channel 0.

(b) Other registers

TOE0:TOE00	0: Stops the TO00 output operation by counting operation. 1: Enables the TO00 output operation by counting operation.
TO0:TO00	0: Outputs a low level from TO00. 1: Outputs a high level from TO00.
TOM0:TOM00	0: Sets toggle mode.
TOT0:TOT00	0: Sets 0 when TOM00 = 0 (toggle mode).
TOL0:TOL00	0: Sets 0 when TOM00 = 0 (toggle mode).
TDE0:TDE00	0: Stops dead time control.
TRE0:TRE00	0: Stops real-time output.
TRO0:TRO00	0: Sets 0 when TRE00 = 0 (stops real-time output).
TRC0:TRC00	0: Does not operate as the real-time output trigger generation channel.
TME0:TME00	0: Stops modulated output.

Figure 7-30. Example of Set Contents of Registers When Triangular Wave PWM Output Function (Slave Channel) Is Used

(a) Timer mode register m (TMRm)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRm	CKSm		CCS1m	CCS0m	MAS TERm	STSm2	STSm1	STSm0	CISm1	CISm0		MDm4	MDm3	MDm2	MDm1	MDm0
	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0

Operation mode of channels 2 to 7
1001B: Up and down count

Start trigger during operation
0: Does not generate INTTM00 when counting is started.

Selection of TIm pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
111B: Selects up or down control trigger of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
10B: Selects count clock of master channel.

Operation clock selection
0: Selects CK00 as operation clock of channels 2 to 7.

(b) Other registers

TOE0:TOEm	0: Stops the TOm output operation by counting operation. 1: Enables the TOm output operation by counting operation.
TO0:TOm	0: Outputs a low level from TOm. 1: Outputs a high level from TOm.
TOM0:TOMm	1: Sets combination-operation mode.
TOT0:TOTm	1: Sets triangular wave PWM output.
TOL0:TOLm	0: Positive logic output (active-high) 1: Inverted output (active-low)
TDE0:TDEm	0: Stops dead time control.
TRE0:TREm	0: Stops real-time output.
TRO0:TROm	0: Sets 0 when TREm = 0 (stops real-time output).
TRC0:TRCm	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEm	0: Stops modulated output.

Remark m = 02 to 07

Figure 7-31. Operation Procedure When Triangular Wave PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN and TAUOPEN bits of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR00 and TMRm registers of each channel to be used (determines operation mode of channels). An interval (period) value is set to the TDR00 register of the master channel, and a duty factor is set to the TDRm register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets the master channel. Sets the TOM00 bit of the TOM0 register to 0 (toggle mode). Sets the slave channel. Sets the TOMm bit of the TOM0 register to 1 (combination-operation mode). Sets the TOTm bit of the TOT0 register to 1 (triangular wave PWM output). Sets the TOLm bit to 0 (positive logic output). Sets the TOm bit and determines default level of the TOm output.	The TO00 and TOm pins go into Hi-Z output states. The TOn default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE00 and TOEm bits to 1 and enables operation of TO00 and TOm. Clears the port register and port mode register to 0.	TO00 and TOm do not change because channel stops operating. The TO00 and TOm pins output the TO00 and TOm set levels.
Operation start	Sets the TOE00 (master) and TOEm (slave) bits to 1 (only when operation is resumed). The TS00 (master) and TSm (slave) bits of the TS0 register are set to 1 at the same time. The TS00 and TSm bits automatically return to 0 because they are trigger bits.	TE00 = 1, TEm = 1 When the master and slave channels starts counting and the MD000 bit of the TMR00 register is 1, INTTM00 is generated.
During operation	The set value of the TDR00 (master) register must be changed during an up status period. The set value of the TDRm (slave) register can be changed. The TCR00 and TCRm registers can always be read. The TSRm (slave) register can always be read.	At the master channel, a period is generated and count operation of the slave channel is controlled. TCR00 loads the value of TDR00 and counts down. When the count value reaches TCR00 = 0000H, INTTM00 is generated. At the same time, the value of the TDR00 register is loaded to TCR00, and the counter starts counting down again. At the slave channel, INTTM00 of the master channel is used as the trigger to switch counting down and counting up. INTTMm is generated upon detection of TCRm = 0001H and TOm outputs a triangular wave PWM. At the master channel, TCR00 loads the value of TDR00 again and count operation is continued by the generation of INTTM00 during an up status. After that, the above operation is repeated.

Operation is resumed. (from next page)

Remark m = 02 to 07

Operation is resumed. (to forward page)

Figure 7-31. Operation Procedure When Triangular Wave PWM Output Function Is Used (2/2)

	Software Operation	Hardware Status
Operation stop	The TT00 (master) and TTm (slave) bits are set to 1 at the same time. The TT00 and TTm bits automatically return to 0 because they are trigger bits.	TE00, TEm = 0, and count operation stops. TCR00 and TCRm hold count value and stops. The TO00 and TOm outputs are not initialized but hold current statuses.
	The TOE00 and TOEm bits are cleared to 0 and values are set to the TO00 and TOm bits.	The TO00 and TOm pins output the TO00 and TOm set levels.
TAUS stop	To hold the TO00 and TOm pin output levels Clears the TO00 and TOm bits to 0 after the value to be held is set to the port register. When holding the TO00 and TOm pin output levels is not necessary Switches the port mode register to input mode.	The TO00 and TOm pin output levels are held by port function. The TO00 and TOm pin output levels go into Hi-Z output states.
	The TAU0EN and TAUOPEN bits of the PER2 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO00 and TOm bits are cleared to 0 and the TO00 and TOm pins are set to port mode.)

Remark m = 02 to 07

7.4.5 Operation as triangular wave PWM output function with dead time

The triangular wave PWM output function with dead time uses four channels of channels 0 to 3 or channels 4 to 7 in combination to output a triangular wave PWM waveform (with dead time).

It outputs triangular wave PWM output signals with dead times from slave channels 2 and 3, and slave channels 6 and 7. Slave channels 1 and 5 can be operated in any operation mode. (With this function, the operation modes of slave channels 1 and 5 will not be fixed.)

The output pulse period, duty factor (positive phase), and duty factor (reverse phase) can be calculated by the following expression.

Pulse period (down/up) = {Set value of TDRn (master) + 1} × 2 × Count clock period
Duty factor (positive phase) [%] = {{Set value of TDRn (master) + 1} – {Set value of TDRp (slave p)} × 2 – {Set value of TDRq (slave q) + 1}} × Count clock period
Duty factor (reverse phase) [%] = {{Set value of TDRn (master) + 1} – {Set value of TDRp (slave p)} × 2 + {Set value of TDRq (slave q) + 1}} × Count clock period

Errors will be included in the output waveforms when the dead time function is used. The output width of a positive-phase wave will be shortened by the amount of dead time, and the output width of a reverse- phase wave will be extended by the amount of dead time. The linearity of output transition will be lost in the neighborhood of 0% and 100% outputs due to the errors.

0% output: Set value of TDRp (slave p) ≥ Set value of TDRn (master) + 1
100% output: Set value of TDRp (slave p) = 0000H

At the master channel, channels 0 and 4 are used.

TCRn operates as a down counter in the interval timer mode.

TCRn loads the value of TDRn at the first count clock, after the channel start trigger bit (TSn) is set to 1. At this time, INTTMn is not output and TOn is not toggled when MDn0 of TMRn is 0. INTTMn is output and TOn is toggled when MDn0 of TMRn is 1.

Afterward, TCRn counts down along with the count clock.

When TCRn has become 0000H, INTTMn is output and TOn is toggled upon the next count clock. TCRn loads the value of TDRn again at the same timing. Similar operation is continued hereafter.

A carrier period is generated in two periods of the master channel count.

The count operation of the slave channel is controlled by defining the first period of the master channel as a down status of the slave channel and the second period as an up status of the slave channel.

TOn of the master channel outputs up and down statuses.

TOn of the TO0 register must be manipulated while TOEn of the TOE0 register is 0 and the default level must be set, because up and down statuses are output.

TOn of the TO0 register is set to 1 when MDn0 of the TMRn register is 0, and TOn is set to 0 when MDn0 is 1.

By setting the default level, a high level is output from TO00 during a down status and a low level is output during an up status.

Remark n = 00, 04
p = 02, 06
q = 03, 07

Slave channels 1 and 5 are not used as PWM output functions with dead times.

Dead time is controlled by using slave channel p (p = 2, 6) and slave channel q (q = 3, 7) in combination. The triangular wave PWM output function with dead time uses master channel 0, slave channel 2, and slave channel 3, and master channel 4, slave channel 6, and slave channel 7 in combination.

TCRp of slave channel p operates in the up and down count mode, and counts the duty. TCRp loads the value of TDRp at the first count clock, after the channel start trigger bit (TSp) is set to 1. Hereafter, counting up and counting down is switched in accordance with the operation of the master channel. INTTMp is output when TCRp becomes 0001H.

TCRp loads the value of TDRp again when INTTMn is generated in an up status of the master channel. Similar operation is continued hereafter.

TCRq of slave channel q operates in the one-count mode, and counts the dead time.

TCRq loads the value of TDRq and counts down by using INTTMp of slave channel p as the start trigger. When TCRq becomes 0000H, it outputs INTTMq and stops counting until the next start trigger is input (INTTMp of slave channel p).

A triangular wave PWM waveform with dead time is output by changing TOp and TOq by the count operation (INTTMp, INTTMq) of slave channel p (duty) and slave channel q (dead time). A positive-phase waveform and a reverse-phase waveform are output by controlling the TOLp and TOLq bits of the TOL0 registers of slave channel p and slave channel q.

The set condition of TOp (TOLp = 0) is the generation of INTTMq by the operation of slave channel q, which uses the generation of INTTMp while the TCRp register counts down as the start trigger. The reset condition of TOp (TOLp = 0) is the generation of INTTMp of slave channel p while TCRp counts up.

The set condition of TOq (TOLq = 1) is the generation of INTTMp while the TCRp register counts down. The reset condition of TOq (TOLq = 0) is the generation of INTTMq by the operation of slave channel q, which uses the generation of INTTMp while TCRp counts down as the start trigger.

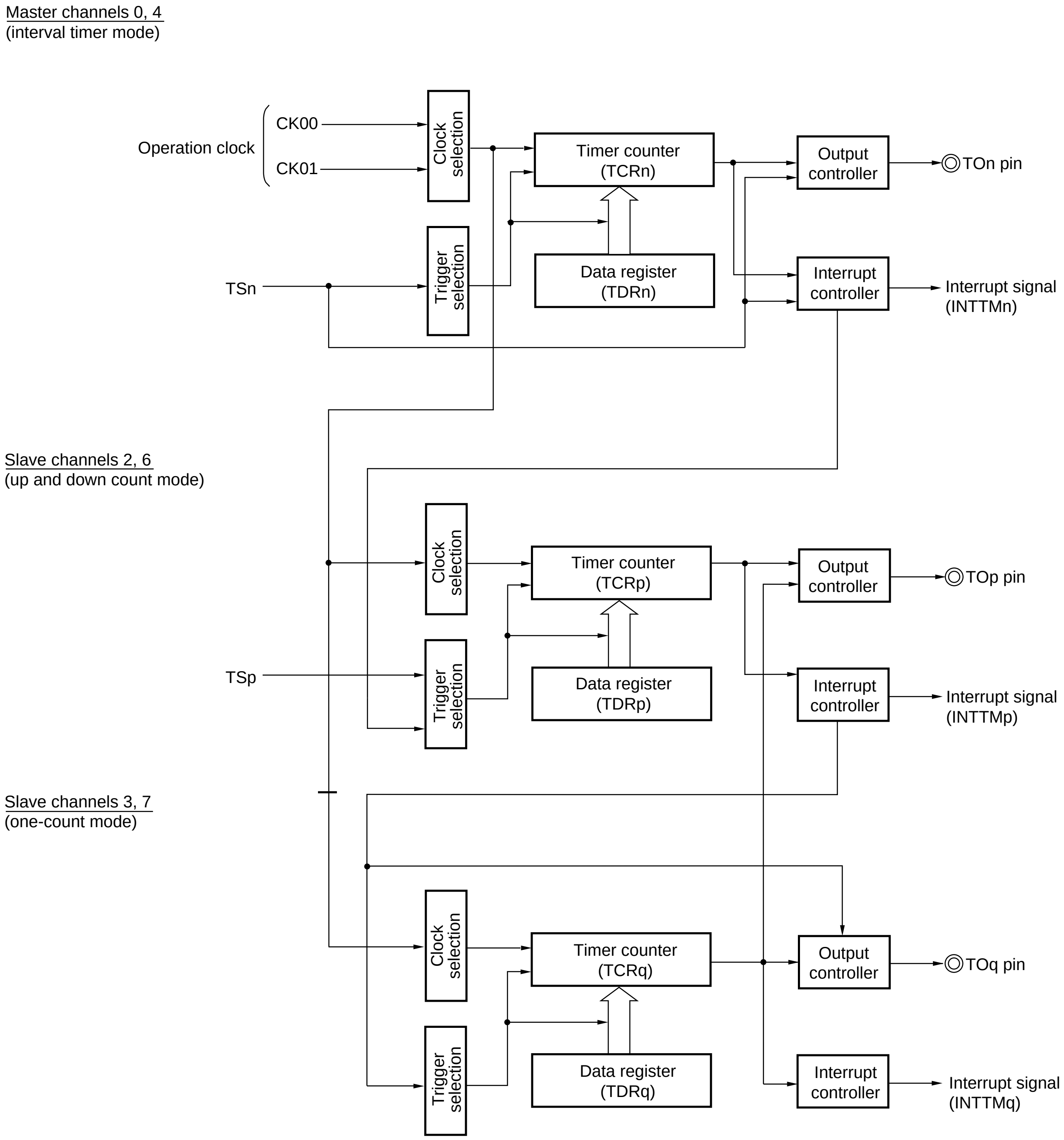
Caution TDRn of master channel n must be rewritten during an up status period of slave channel p. When the value of TDRn is rewritten during a down status period, the periods of the down status and up status differ and an expected waveform cannot be output, because the value of TDRn of the rewritten master channel becomes valid at the next period.

The value of TDRp of slave channel p becomes valid from the next carrier period (up and down trigger detection).

The value of TDRq of slave channel q becomes valid from the next start timing (dead time control trigger detection).

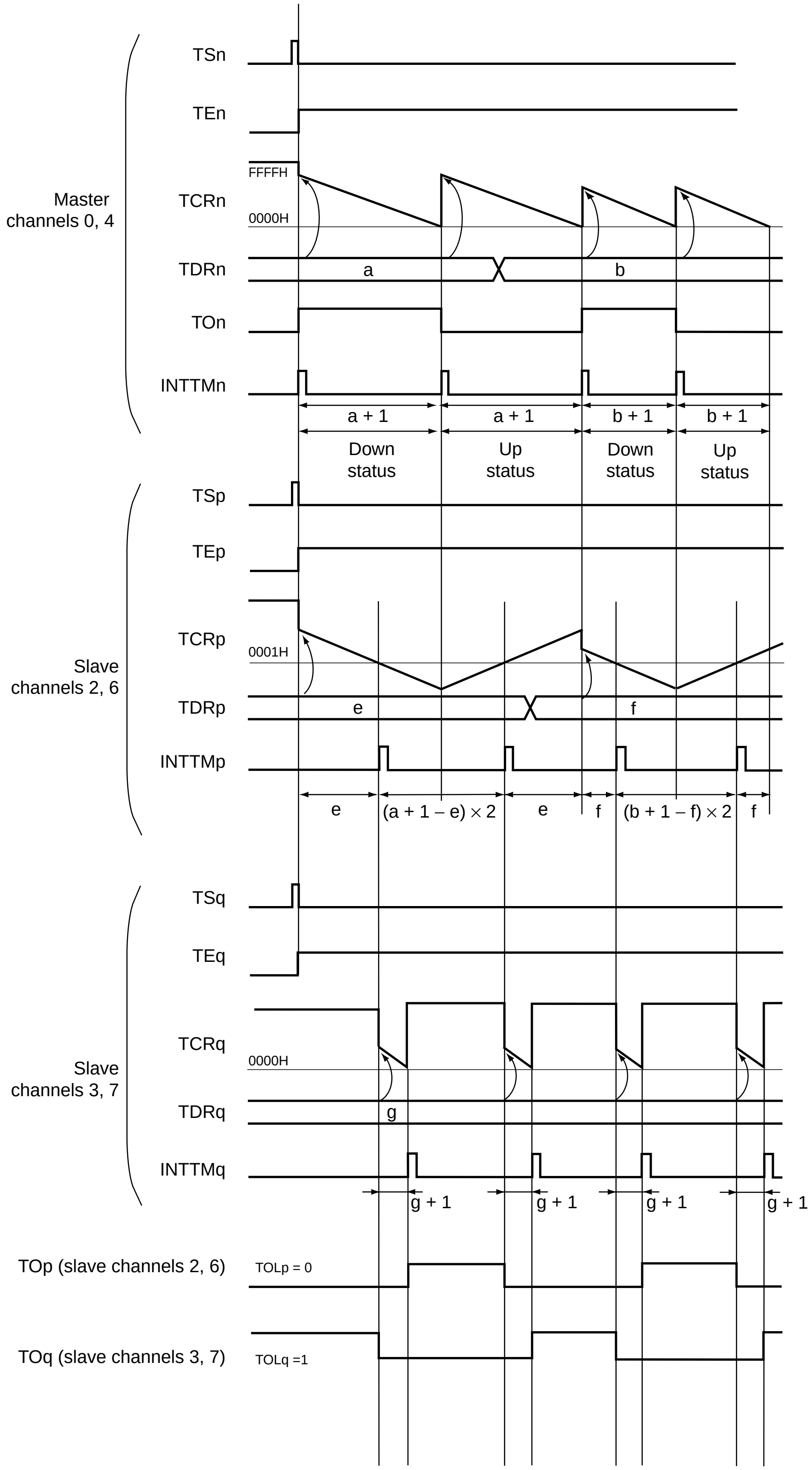
Remark n = 00, 04
p = 02, 06
q = 03, 07

Figure 7-32. Block Diagram of Operation as Triangular Wave PWM Output Function with Dead Time



Remark n = 00, 04
p = 02, 06
q = 03, 07

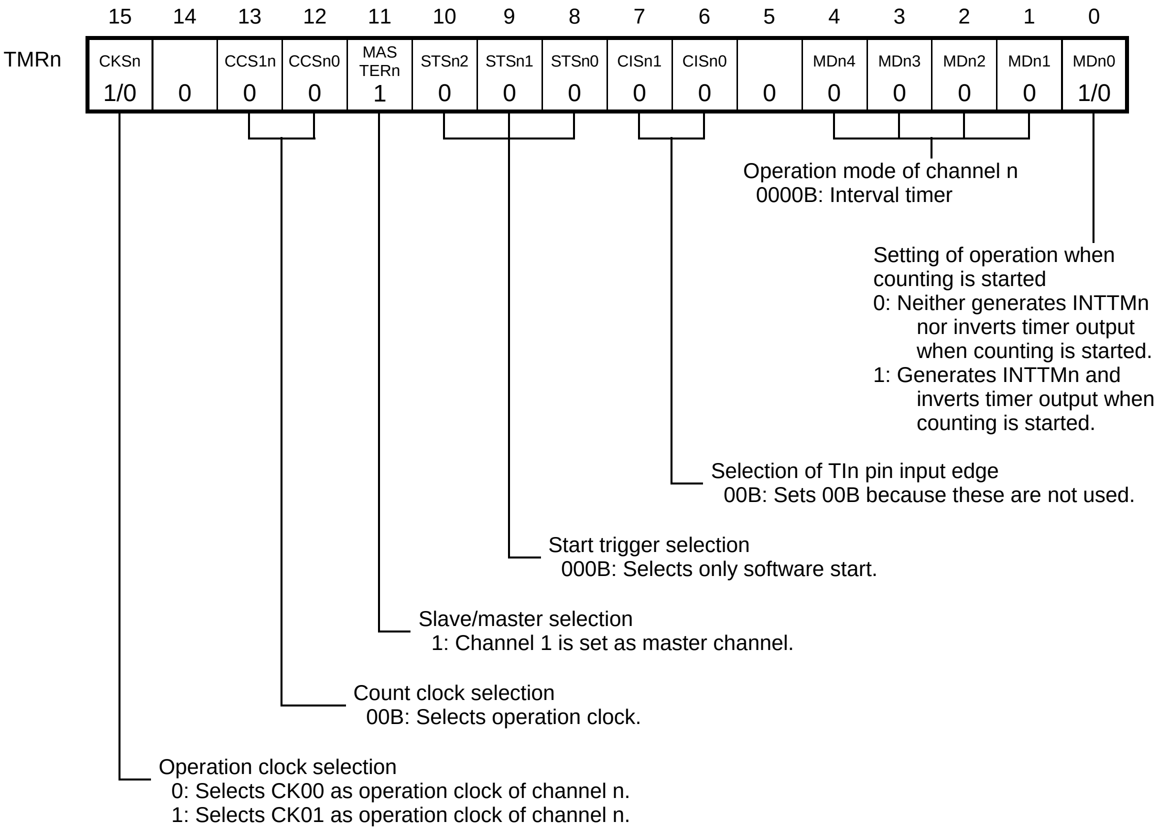
Figure 7-33. Example of Basic Timing of Operation as Triangular Wave PWM Output Function with Dead Time



Remark n = 00, 04
p = 02, 06
q = 03, 07

Figure 7-34. Example of Set Contents of Registers When Triangular Wave PWM Output Function with Dead Time (Master Channel n) Is Used

(a) Timer mode register n (TMRn)



Operation mode of channel n
0000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMn nor inverts timer output when counting is started.
1: Generates INTTMn and inverts timer output when counting is started.

Selection of TIn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

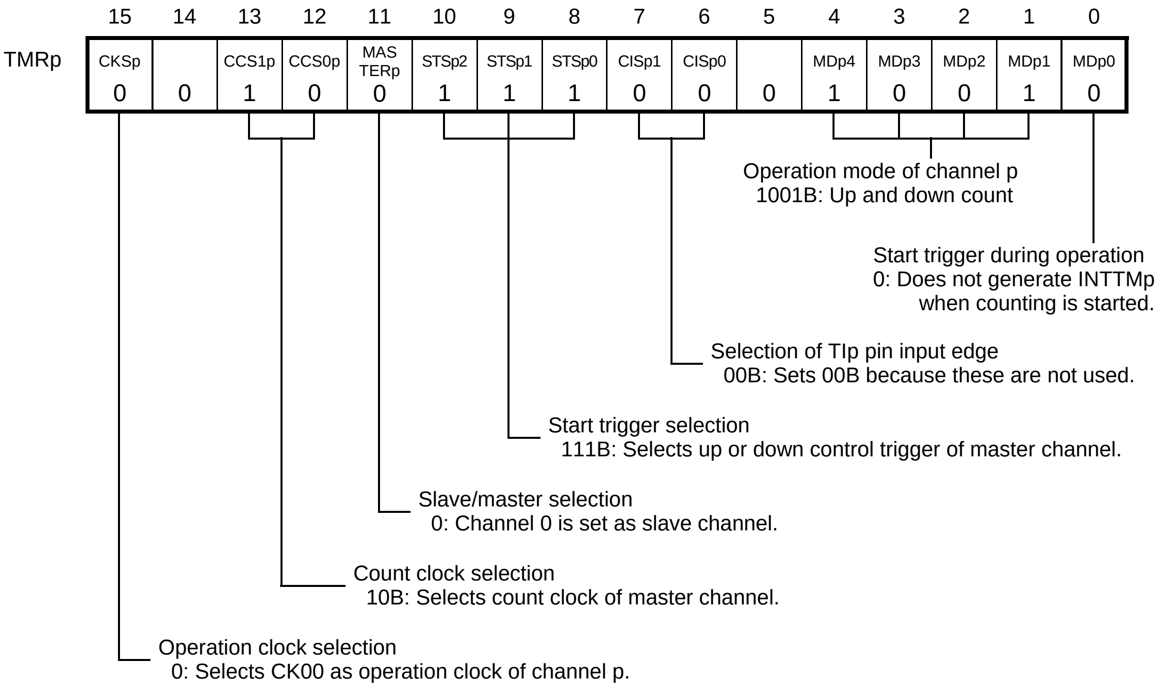
(b) Other registers

TOE0:TOEn	0: Stops the TOn output operation by counting operation. 1: Enables the TOn output operation by counting operation.
TO0:TOn	0: Outputs a low level from TOn. 1: Outputs a high level from TOn.
TOM0:TOMn	0: Sets toggle mode.
TOT0:TOTn	0: Sets 0 when TOMn = 0 (toggle mode).
TOL0:TOLn	0: Sets 0 when TOMn = 0 (toggle mode).
TDE0:TDEn	0: Stops dead time control.
TRE0:TREn	0: Stops real-time output.
TRO0:TROn	0: Sets 0 when TREn = 0 (stops real-time output).
TRC0:TRCn	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEn	0: Stops modulated output.

Remark n = 00, 04

Figure 7-35. Example of Set Contents of Registers When Triangular Wave PWM Output Function with Dead Time (Slave Channel p) Is Used

(a) Timer mode register p (TMRp)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRp	CKSp		CCS1p	CCS0p	MAS TERp	STSp2	STSp1	STSp0	CISp1	CISp0		MDp4	MDp3	MDp2	MDp1	MDp0
	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0

Operation mode of channel p
1001B: Up and down count

Start trigger during operation
0: Does not generate INTTMp when counting is started.

Selection of TIp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
111B: Selects up or down control trigger of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
10B: Selects count clock of master channel.

Operation clock selection
0: Selects CK00 as operation clock of channel p.

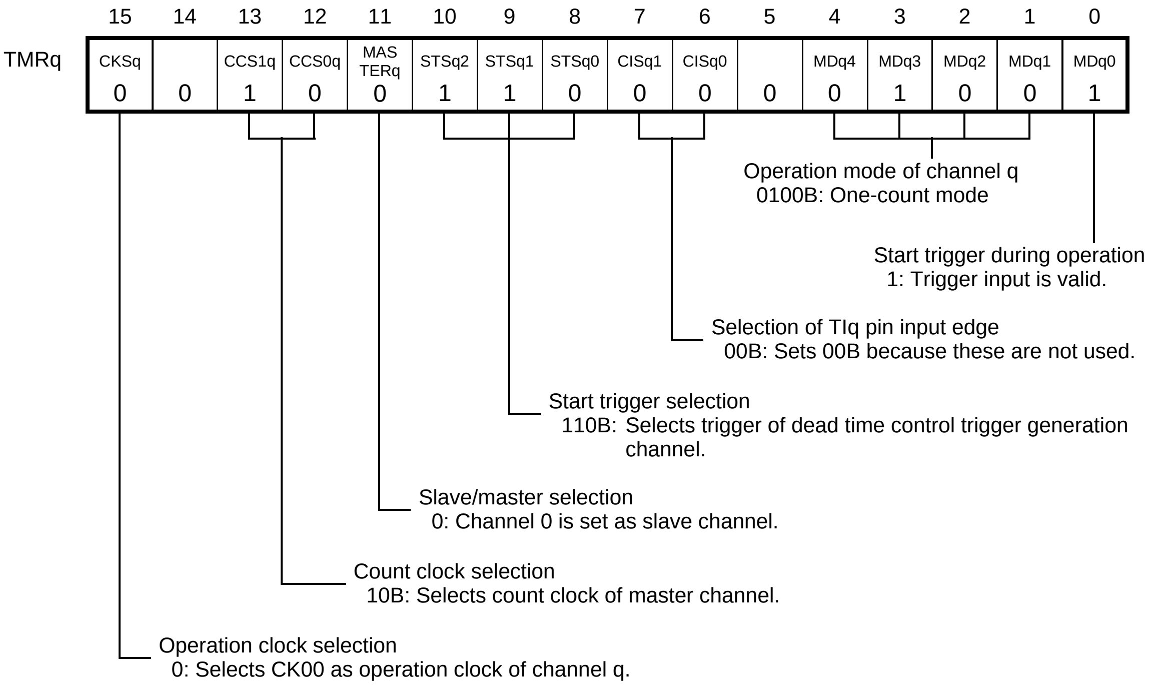
(b) Other registers

TOE0:TOEp	0: Stops the TOp output operation by counting operation. 1: Enables the TOp output operation by counting operation.
TO0:TOp	0: Outputs a low level from TOp. 1: Outputs a high level from TOp.
TOM0:TOMp	1: Sets combination-operation mode.
TOT0:TOTp	1: Generates triangular wave PWM output.
TOL0:TOLp	0: Positive logic output (active-high) 1: Inverted output (active-low)
TDE0:TDEp	1: Enables dead time control.
TRE0:TREp	0: Stops real-time output.
TRO0:TROp	0: Sets 0 when TREp = 0 (stops real-time output).
TRC0:TRCp	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEp	0: Stops modulated output.

Remark p = 02, 06

Figure 7-36. Example of Set Contents of Registers When Triangular Wave PWM Output Function with Dead Time (Slave Channel q) Is Used

(a) Timer mode register q (TMRq)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRq	CKSq		CCS1q	CCS0q	MAS TERq	STSq2	STSq1	STSq0	CISq1	CISq0		MDq4	MDq3	MDq2	MDq1	MDq0
	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	1

Operation mode of channel q
0100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TIq pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
110B: Selects trigger of dead time control trigger generation channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
10B: Selects count clock of master channel.

Operation clock selection
0: Selects CK00 as operation clock of channel q.

(b) Other registers

TOE0:TOEq	0: Stops the TOq output operation by counting operation. 1: Enables the TOq output operation by counting operation.
TO0:TOq	0: Outputs a low level from TOq. 1: Outputs a high level from TOq.
TOM0:TOMq	1: Sets combination-operation mode.
TOT0:TOTq	1: Generates triangular wave PWM.
TOL0:TOLq	0: Positive logic output (active-high) 1: Inverted output (active-low)
TDE0:TDEq	1: Enables dead time control.
TRE0:TREq	0: Stops real-time output.
TRO0:TROq	0: Sets 0 when TREq = 0 (stops real-time output).
TRC0:TRCq	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEq	0: Stops modulated output.

Remark q = 03, 07

Figure 7-37. Operation Procedure When Triangular Wave PWM Output Function with Dead Time Is Used (1/2)

Operation is resumed. (from next page)

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN and TAUOPEN bits of the PER2 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMRn, TMRp, and TMRq registers of four channels to be used (determines operation mode of channels). An interval (period) value is set to the TDRn register of the master channel, a duty factor is set to the TDRp register of slave channel p, and a dead time width is set to the TDRq register of slave channel q.	Channel stops operating. (Clock is supplied and some power is consumed.) The TO00, TOp, and TOq pins go into Hi-Z output states.
	Sets slave channels p and q. The TOMp and TOMq bits of the TOM0 register, and the TOTp and TOTq bits of the TOT0 register are set to 1 (triangular wave PWM generation). Sets the TOLp and TOLq bits, and determines the active levels of the TOp and TOq outputs. Sets the TDEp and TDEq bits to 1 (dead time control enable).	
	Sets the TOn, TOp, and TOq bits, and determines default levels of TOn, TOp, and TOq. →	The TOn, TOp, and TOq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOEn, TOEp, and TOEq bits to 1 and enables operation of TOn, TOp, and TOq. →	TOn, TOp, and TOq do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TOn, TOp, and TOq pins output the TOn, TOp, and TOq set levels.
Operation start	Sets the TOEm (master), and TOEp and TOEq (slaves) bits to 1 (only when operation is resumed). The TSn (master), and TSp and TSq (slaves) bits of the TS0 register are set to 1 at the same time. → The TSn, TSp, and TSq bits automatically return to 0 because they are trigger bits.	TEn = 1, TEp = 1, TEq = 1 When the master channel and slave channel p start counting, and when the MDn0 bit of the TMRn register is set to 1, INTTMn is generated. Slave channel q waits until slave channel p detects INTTMp.
During operation	The set value of the TDRn (master) register must be changed during an up status period of slave channel p. The set values of the TDRp and TDRq (slaves) register can be changed. The TCRn, TCRp, and TCRq registers can always be read. The TSRp (slave) register can always be read.	At the master channel, a period is generated and count operation of slave channels p and q are controlled. A PWM duty is generated at slave channel p, and dead time is generated at slave channel q. Triangular wave PWM waveforms with dead times are output from the TOp and TOq pins by a combined operation of slave channel p and slave channel q.

Remark n = 00, 04 p = 02, 06 q = 03, 07

Figure 7-37. Operation Procedure When Triangular Wave PWM Output Function with Dead Time Is Used (2/2)

Operation is resumed.
(to forward page)

	Software Operation	Hardware Status
Operation stop	The TTn (master), and TTp and TTq (slaves) bits are set to 1 at the same time. The TTn, TTp, and TTq bits automatically return to 0 because they are trigger bits.	TEn, TEp, and TEq = 0, and count operation stops. TCRn, TCRp, and TCRq hold count values and stop. The TOn, TOp, and TOq outputs are not initialized but hold current statuses.
	The TOEn, TOEp, and TOEq bits are cleared to 0 and values are set to the TOn, TOp, and TOq bits.	The TO00, TOp, and TOq pins output the TO00, TOp, and TOq set levels.
TAUS stop	To hold the TOn, TOp, and TOq pin output levels Clears the TOn, TOp, and TOq bits to 0 after the value to be held is set to the port register. When holding the TOn, TOp, and TOq pin output levels is not necessary Switches the port mode register to input mode.	The TOn, TOp, and TOq pin output levels are held by port function. The TOn, TOp, and TOq pin output levels go into Hi-Z output states.
	The TAU0EN and TAUOPEN bits of the PER2 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOn, TOp, and TOq bits are cleared to 0 and the TOn, TOp, and TOq pins are set to port mode.)

Remark n = 00, 04
p = 02, 06
q = 03, 07

7.4.6 Operation as 6-phase triangular wave PWM output function

The 6-phase triangular wave PWM output function uses eight channels in combination to output a 6-phase triangular wave PWM waveform (with dead time).

The 6-phase triangular wave PWM output function is an extension of the function described in **7.4.5 Operation as triangular wave PWM output function with dead time**.

It outputs 6-phase triangular wave PWM output signals from slave channel 2, slave channel 3, slave channel 4, slave channel 5, slave channel 6, and slave channel 7. Slave channel 1 can be operated in any operation mode. (With this function, the operation mode of slave channel 1 will not be fixed.)

The output pulse period, duty factor (positive phase), and duty factor (reverse phase) can be calculated by the following expression.

Pulse period (down/up) = {Set value of TDR00 (master) + 1} × 2 × Count clock period
Duty factor (positive phase) [%] = {{Set value of TDR00 (master) + 1} − {Set value of TDRp (slave p)} × 2 − {Set value of TDRq (slave q) + 1}} × Count clock period
Duty factor (reverse phase) [%] = {{Set value of TDR00 (master) + 1} − {Set value of TDRp (slave p)} × 2 + {Set value of TDRq (slave q) + 1}} × Count clock period

Errors will be included in the output waveforms when the dead time function is used. The output width of a positive-phase wave will be shortened by the amount of dead time, and the output width of a reverse- phase wave will be extended by the amount of dead time. The linearity of output transition will be lost in the neighborhood of 0% and 100% outputs due to the errors.

0% output: Set value of TDRp (slave p) ≥ Set value of TDR00 (master) + 1
100% output: Set value of TDRp (slave p) = 0000H

The master channel operates in the interval timer mode and counts the periods.

A carrier period is generated in two periods of the master channel count.

The count operation of the slave channel is controlled by defining the first period of the master channel as a down status of the slave channel and the second period as an up status of the slave channel.

TO00 of the master channel outputs up and down statuses.

TO00 of the TO0 register must be manipulated while TOE00 of the TOE0 register is 0 and the default level must be set, because up and down statuses are output.

TO00 of the TO0 register is set to 1 when MD000 of the TMR0 register is 0, and TO00 is set to 0 when MD000 is 1.

By setting the default level, a high level is output from TO00 during a down status and a low level is output during an up status.

Slave channel 1 is not used as a 6-phase PWM output function.

(To use the modulation output function, slave channel 1 is used as the real-time output trigger generation channel.)

Dead time is controlled by using slave channel p (p = 2, 4, 6) and slave channel q (q = 3, 5, 7) in combination. The 6-phase triangular wave PWM output function uses slave channels 2 and 3, slave channels 4 and 5, and slave channels 6 and 7 in combination. The output operations of TOp and TOq are explained next.

TCRp of slave channel p operates in the up and down count mode, and counts the duty. TCRp loads the value of TDRp at the first count clock, after the channel start trigger bit (TSp) is set to 1. Hereafter, counting up and counting down is switched in accordance with the operation of the master channel. INTTMp is output when TCRp becomes 0001H.

TCRp loads the value of TDRp again when INTTM00 is generated in an up status of the master channel. Similar operation is continued hereafter.

TCRq of slave channel q operates in the one-count mode, and counts the dead time.

TCRq loads the value of TDRq and counts down by using INTTMp of slave channel p as the start trigger. When TCRq becomes 0000H, it outputs INTTMq and stops counting until the next start trigger is input (INTTMq of slave channel q).

A 6-phase triangular wave PWM waveform is output by changing TOp and TOq by the count operations (INTTMp, INTTMq) of slave channel p (duty) and slave channel q (dead time). A positive-phase waveform and a reverse-phase waveform are output by controlling the TOLp and TOLq bits of the TOL0 registers of slave channel p and slave channel q.

The set condition of TOp (TOLp = 0) is the generation of INTTMq by the operation of slave channel q, which uses the generation of INTTMp while the TCRp register counts down as the start trigger. The reset condition of TOp (TOLp = 0) is the generation of INTTMp of slave channel p while TCRp counts up.

The set condition of TOq (TOLq = 1) is the generation of INTTMp while the TCRp register counts down. The reset condition of TOq (TOLq = 0) is the generation of INTTMq by the operation of slave channel q, which uses the generation of INTTMp while TCRp counts up as the start trigger.

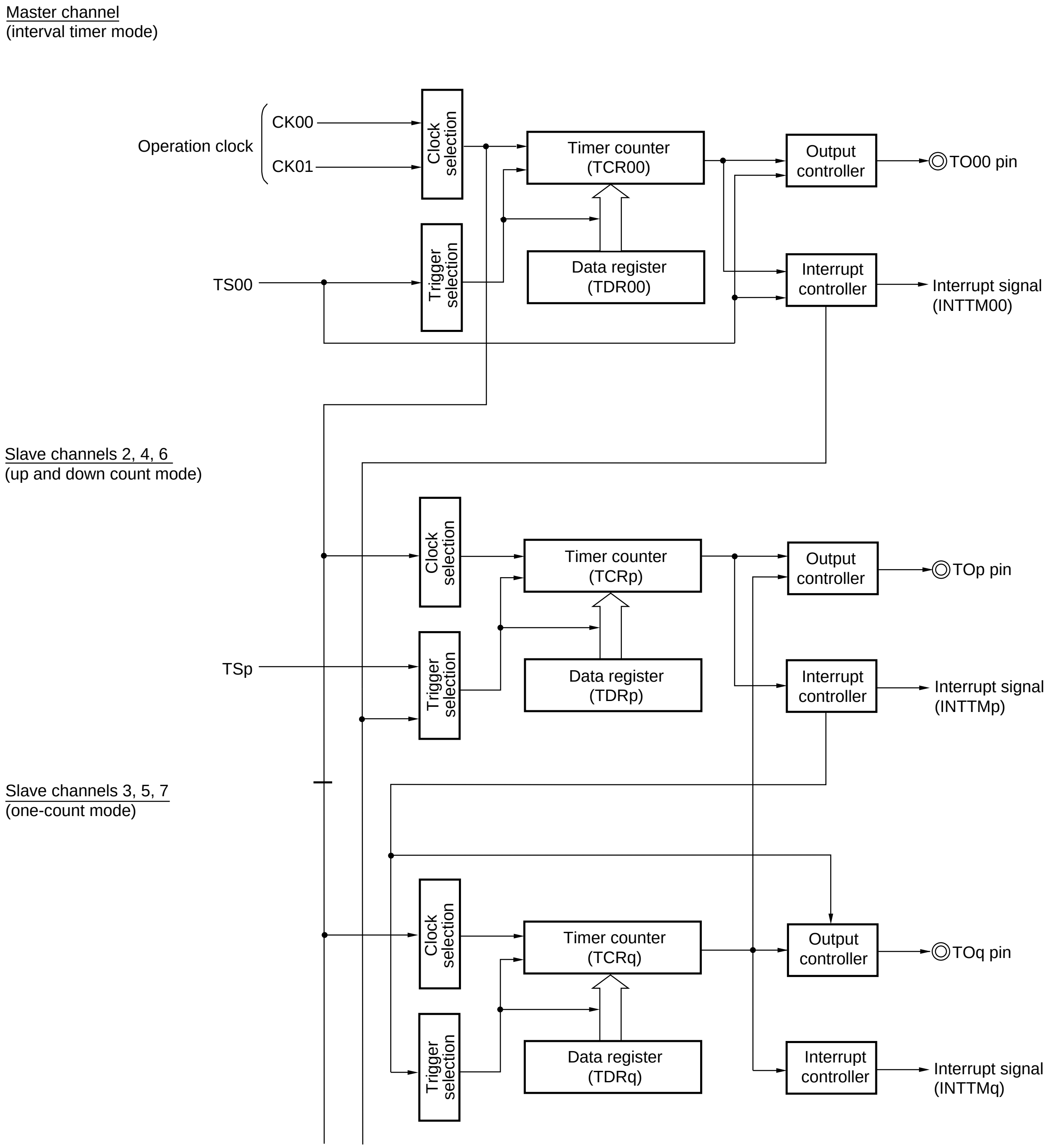
Caution TDR00 of the master channel must be rewritten during an up status period of slave channel p. When the value of TDR00 is rewritten during a down status period, the periods of the down status and up status differ and an expected waveform cannot be output, because the value of TDRn of the rewritten master channel becomes valid at the next period.

The value of TDRp of slave channel p becomes valid from the next carrier period (up and down trigger detection).

The value of TDRq of slave channel q becomes valid from the next start timing (dead time control trigger detection).

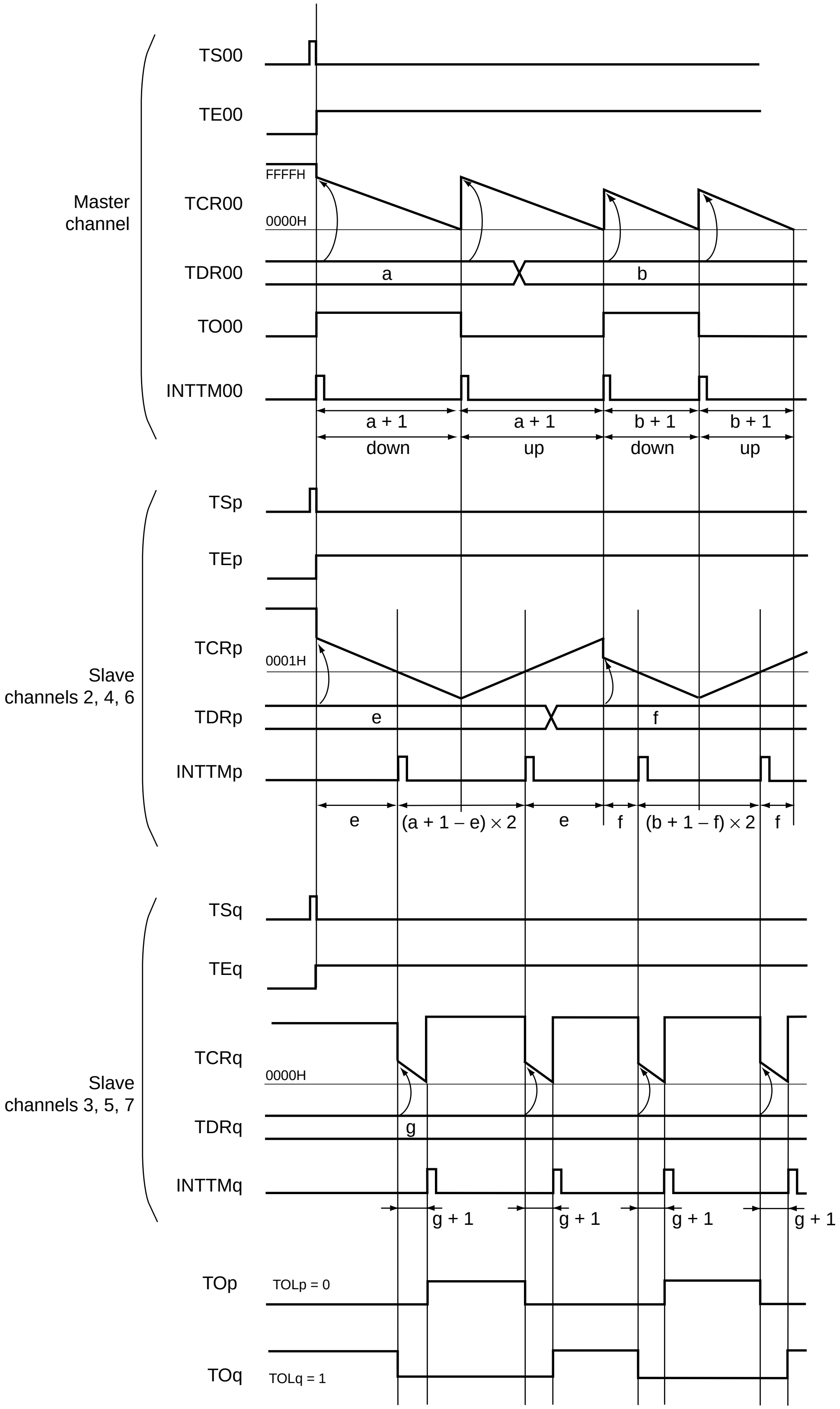
Remark p = 02, 04, 06
q = 03, 05, 07

Figure 7-38. Block Diagram of Operation as 6-Phase Triangular Wave PWM Output Function



Remark p = 02, 04, 06
q = 03, 05, 07

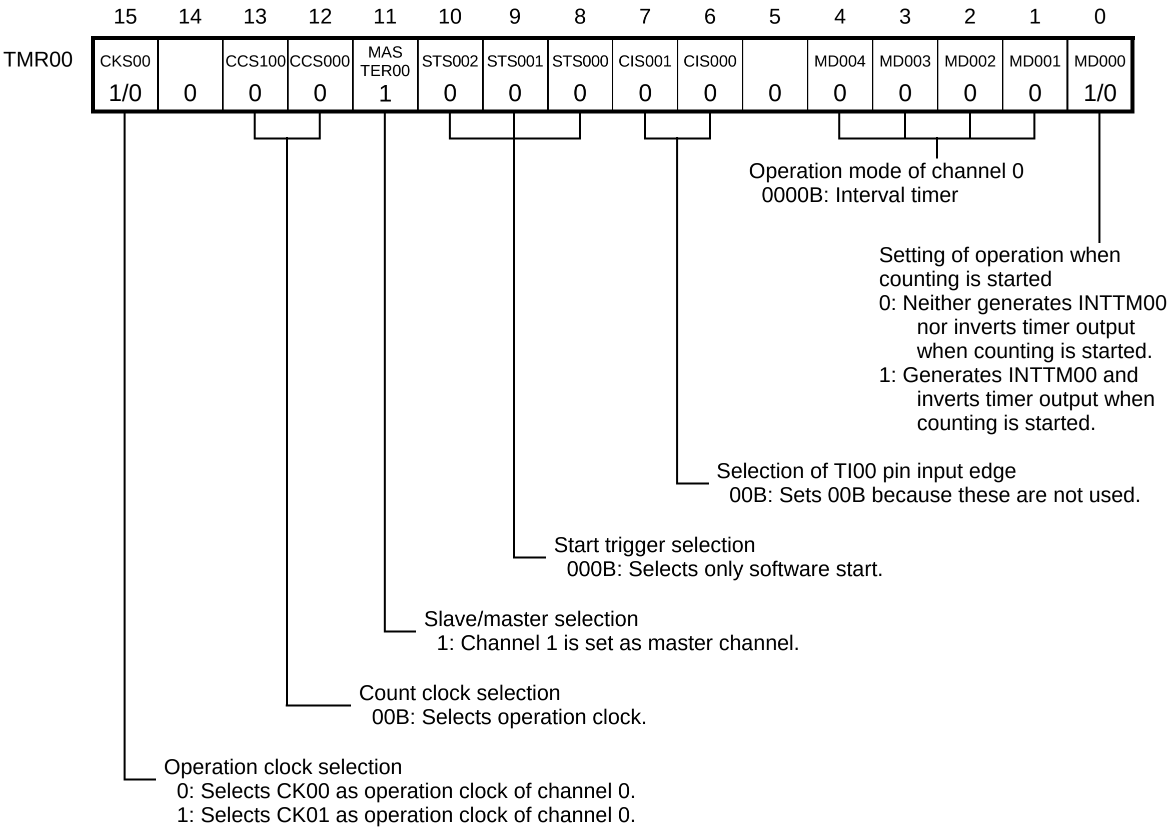
Figure 7-39. Example of Basic Timing of Operation as 6-Phase Triangular Wave PWM Output Function



Remark p = 02, 04, 06
q = 03, 05, 07

Figure 7-40. Example of Set Contents of Registers of 6-Phase Triangular Wave PWM Output Function (Master Channel)

(a) Timer mode register 00 (TMR00)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR00	CKS00		CCS100	CCS000	MAS TER00	STS002	STS001	STS000	CIS001	CIS000		MD004	MD003	MD002	MD001	MD000
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel 0
0000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTM00 nor inverts timer output when counting is started.
1: Generates INTTM00 and inverts timer output when counting is started.

Selection of TI00 pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
00B: Selects operation clock.

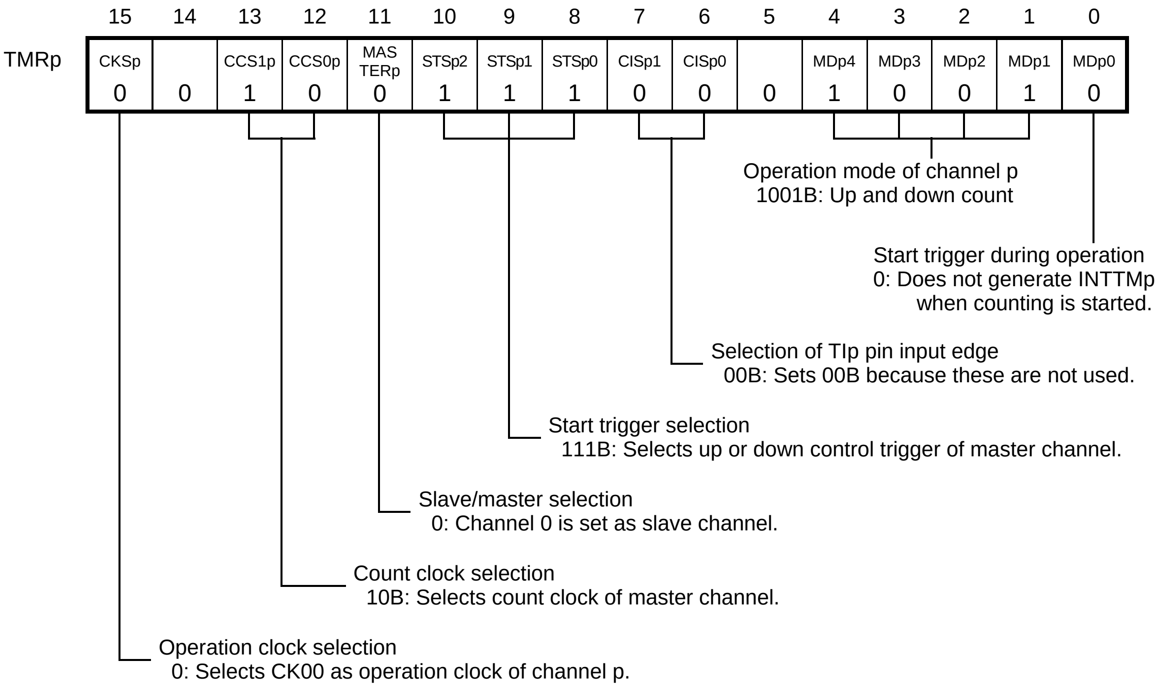
Operation clock selection
0: Selects CK00 as operation clock of channel 0.
1: Selects CK01 as operation clock of channel 0.

(b) Other registers

TOE0:TOE00	0: Stops the TO00 output operation by counting operation. 1: Enables the TO00 output operation by counting operation.
TO0:TO00	0: Outputs a low level from TO00. 1: Outputs a high level from TO00.
TOM0:TOM00	0: Sets toggle mode.
TOT0:TOT00	0: Sets 0 when TOM00 = 0 (toggle mode).
TOL0:TOL00	0: Sets 0 when TOM00 = 0 (toggle mode).
TDE0:TDE00	0: Stops dead time control.
TRE0:TRE00	0: Stops real-time output.
TRO0:TRO00	0: Sets 0 when TRE00 = 0 (stops real-time output).
TRC0:TRC00	0: Does not operate as the real-time output trigger generation channel.
TME0:TME00	0: Stops modulated output.

Figure 7-41. Example of Set Contents of Registers of 6-Phase Triangular Wave PWM Output Function (Slave Channel p)

(a) Timer mode register p (TMRp)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRp	CKSp 0	0	CCS1p 1	CCS0p 0	MAS TERp 0	STSp2 1	STSp1 1	STSp0 1	CISp1 0	CISp0 0	0	MDp4 1	MDp3 0	MDp2 0	MDp1 1	MDp0 0

Operation mode of channel p
1001B: Up and down count

Start trigger during operation
0: Does not generate INTTMp when counting is started.

Selection of TIp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
111B: Selects up or down control trigger of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
10B: Selects count clock of master channel.

Operation clock selection
0: Selects CK00 as operation clock of channel p.

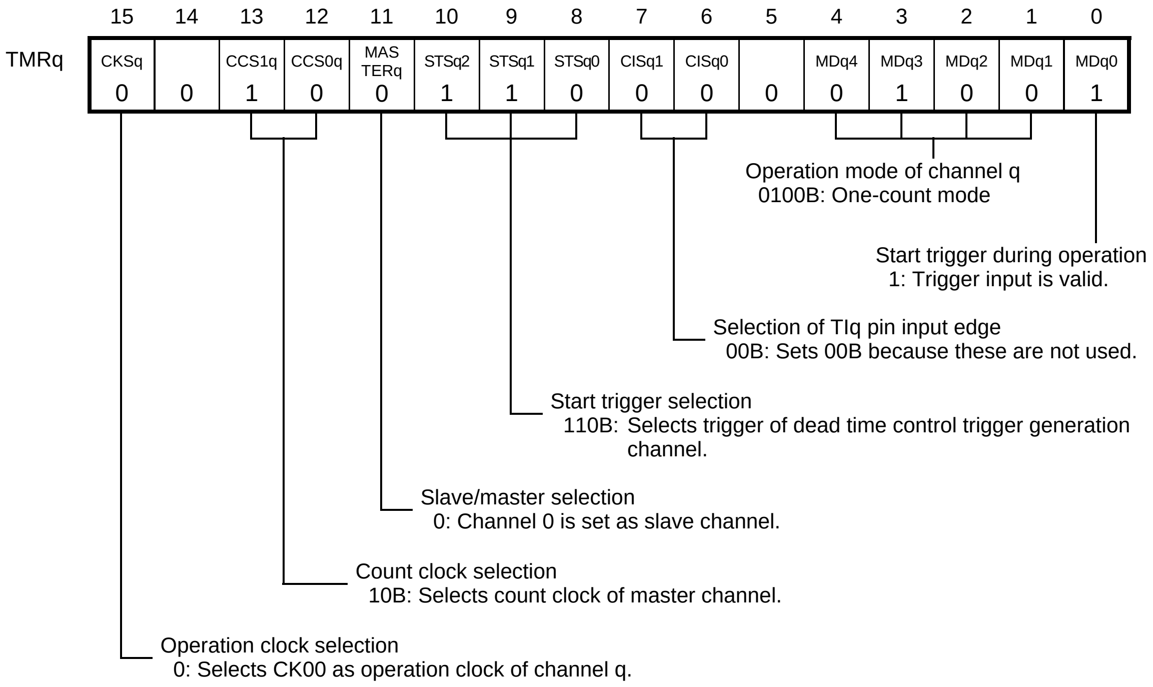
(b) Other registers

TOE0:TOEp	0: Stops the TOp output operation by counting operation. 1: Enables the TOp output operation by counting operation.
TO0:TOp	0: Outputs a low level from TOp. 1: Outputs a high level from TOp.
TOM0:TOMp	1: Sets combination-operation mode.
TOT0:TOTp	1: Sets triangular wave PWM output.
TOL0:TOLp	0: Positive logic output (active-high) 1: Inverted output (active-low)
TDE0:TDEp	1: Enables dead time control.
TRE0:TREp	0: Stops real-time output.
TRO0:TROp	0: Sets 0 when TREp = 0 (stops real-time output).
TRC0:TRCp	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEp	0: Stops modulated output.

Remark p = 02, 04, 06

Figure 7-42. Example of Set Contents of Registers of 6-Phase Triangular Wave PWM Output Function (Slave Channel q)

(a) Timer mode register q (TMRq)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRq	CKSq		CCS1q	CCS0q	MAS TERq	STSq2	STSq1	STSq0	CISq1	CISq0		MDq4	MDq3	MDq2	MDq1	MDq0
	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	1

Operation mode of channel q
0100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TIq pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
110B: Selects trigger of dead time control trigger generation channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
10B: Selects count clock of master channel.

Operation clock selection
0: Selects CK00 as operation clock of channel q.

(b) Other registers

TOE0:TOEq	0: Stops the TOq output operation by counting operation. 1: Enables the TOq output operation by counting operation.
TO0:TOq	0: Outputs a low level from TOq. 1: Outputs a high level from TOq.
TOM0:TOMq	1: Sets combination-operation mode.
TOT0:TOTq	1: Sets triangular wave PWM output.
TOL0:TOLq	0: Positive logic output (active-high) 1: Inverted output (active-low)
TDE0:TDEq	1: Enables dead time control.
TRE0:TREq	0: Stops real-time output.
TRO0:TROq	0: Sets 0 when TREq = 0 (stops real-time output).
TRC0:TRCq	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEq	0: Stops modulated output.

Remark q = 03, 05, 07

Figure 7-43. Operation Procedure When 6-Phase Triangular Wave PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN and TAUOPEN bits of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR00, TMRp, and TMRq registers of the channels to be used (determines operation mode of channels). An interval (period) value is set to the TDR00 register of the master channel, a duty factor is set to the TDRp register of slave channel p, and a dead time width is set to the TDRq register of slave channel q.	Channel stops operating. (Clock is supplied and some power is consumed.) The TO00, TOp, and TOq pins go into Hi-Z output states.
	Sets slave channels p and q. The TOMp and TOMq bits of the TOM0 register, and the TOTp and TOTq bits of the TOT0 register are set to 1 (triangular wave PWM output). Sets the TOLp and TOLq bits, and determines the active levels of TOp and TOq. Sets the TDEp and TDEq bits of the TDE0 register to 1 (dead time control enable).	
	Sets the TO00, TOp, and TOq bits, and determines default levels of TO00, TOp, and TOq. Sets the TOE00, TOEp, and TOEq bits to 1 and enables operation of TO00, TOp and TOq. Clears the port register and port mode register to 0.	The TO00, TOp, and TOq default setting levels are output when the port mode register is in output mode and the port register is 0. TO00, TOp, and TOq do not change because channels stop operating. The TO00, TOp, and TOq pins output the TOp and TOq set levels.
Operation start	Sets the TOE00 (master), and TOEp and TOEq (slaves) bits to 1 (only when operation is resumed). The TS00 (master), and TSp and TSq (slaves) bits of the TS0 register are set to 1 at the same time. The TS00, TSp,and TSq bits automatically return to 0 because they are trigger bits.	TE00 = 1, TEp = 1, TEq = 1 When the master channel and slave channel q start counting, and when the MD000 bit of the TMR00 register is set to 1, INTTM00 is generated.
During operation	The set value of the TDR00 (master) register must be changed during an up status period. The set values of the TDRp and TDRq (slaves) register can be changed. The TCR00, TCRp, and TCRq registers can always be read. The TSRp (slave) register can always be read.	At the master channel, a period is generated and count operation of slave channels p and q are controlled. A PWM duty is generated at slave channel p, and dead time is generated at slave channel q. Triangular wave PWM waveforms with dead times are output from the TOp and TOq pins by a combined operation of slave channel p and slave channel q.

Operation is resumed. (from next page)

Remark p = 02, 04, 06
q = 03, 05, 07

Figure 7-43. Operation Procedure When 6-Phase Triangular Wave PWM Output Function Is Used (2/2)

Operation is resumed.
(to forward page)

	Software Operation	Hardware Status
Operation stop	The TT00 (master), and TTp and TTq (slaves) bits are set to 1 at the same time. The TT00, TTp, and TTq bits automatically return to 0 because they are trigger bits.	TE00, TEp, and TEq = 0, and count operation stops. TCR00, TCRp, and TCRq hold count values and stop. The TO00, TOp, and TOq outputs are not initialized but hold current statuses.
	The TOE00, TOEp, and TOEq bits are cleared to 0 and values are set to the TO00, TOp, and TOq bits.	The TO00, TOp, and TOq pins output the TO00, TOp, and TOq set levels.
TAUS stop	To hold the TO00, TOp, and TOq pin output levels Clears the TO00, TOp, and TOq bits to 0 after the value to be held is set to the port register. When holding the TO00, TOp, and TOq pin output levels is not necessary Switches the port mode register to input mode.	The TO00, TOp, and TOq pin output levels are held by port function. The TO00, TOp, and TOq pin output levels go into Hi-Z output states.
	The TAU0EN and TAUOPEN bits of the PER2 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO00, TOp, and TOq bits are cleared to 0 and the TO00, TOp, and TOq pins are set to port mode.)

Remark p = 02, 04, 06
q = 03, 05, 07

7.4.7 Interrupt signal thinning function

The interrupt signal thinning function uses two channels in combination to output INTTMm that is INTTMn of the master channel being thinned by the specified number of times from the slave channel.

It assumes the slave channel to be used as a sub-function of the 6-phase triangular wave PWM output function (7.4.6). The setting of the master channel is therefore the same as in 6-phase triangular wave PWM output function (7.4.6). The number of interrupts to be thinned can be calculated by the following expression.

Number of interrupts to be thinned = Set value of TDRm (slave channel)
→ Outputting INTTMn of the master channel from INTTMm of the slave channel every {Set value of TDRm (slave) + 1} times

TCRn of the master channel counts down in the interval timer mode.

TCRn loads the value of TDRn by setting the channel start trigger bit (TSn) to 1. At this time, INTTMn is not output and TOn is not toggled when MDn0 of TMRn is 0. INTTMn is output and TOn is toggled when MDn0 of TMRn is 1. Afterward, TCRn counts down along with the count clock. When TCRn has become 0000H, INTTMn is output and TOn is toggled upon the next count clock. TCRn loads the value of TDRn again at the same timing. Similar operation is continued hereafter.

The slave channel operates as a down counter in the event counter mode and controls the thinning of INTTMn signals of the master channel.

TCRm loads the value of TDRm by setting the channel start trigger bit (TSm) to 1.

TCRm counts down along with the INTTMn output of the master channel, and loads the value of TDRm again and outputs INTTMm when TCRm becomes 0000H. Similar operation is continued hereafter.

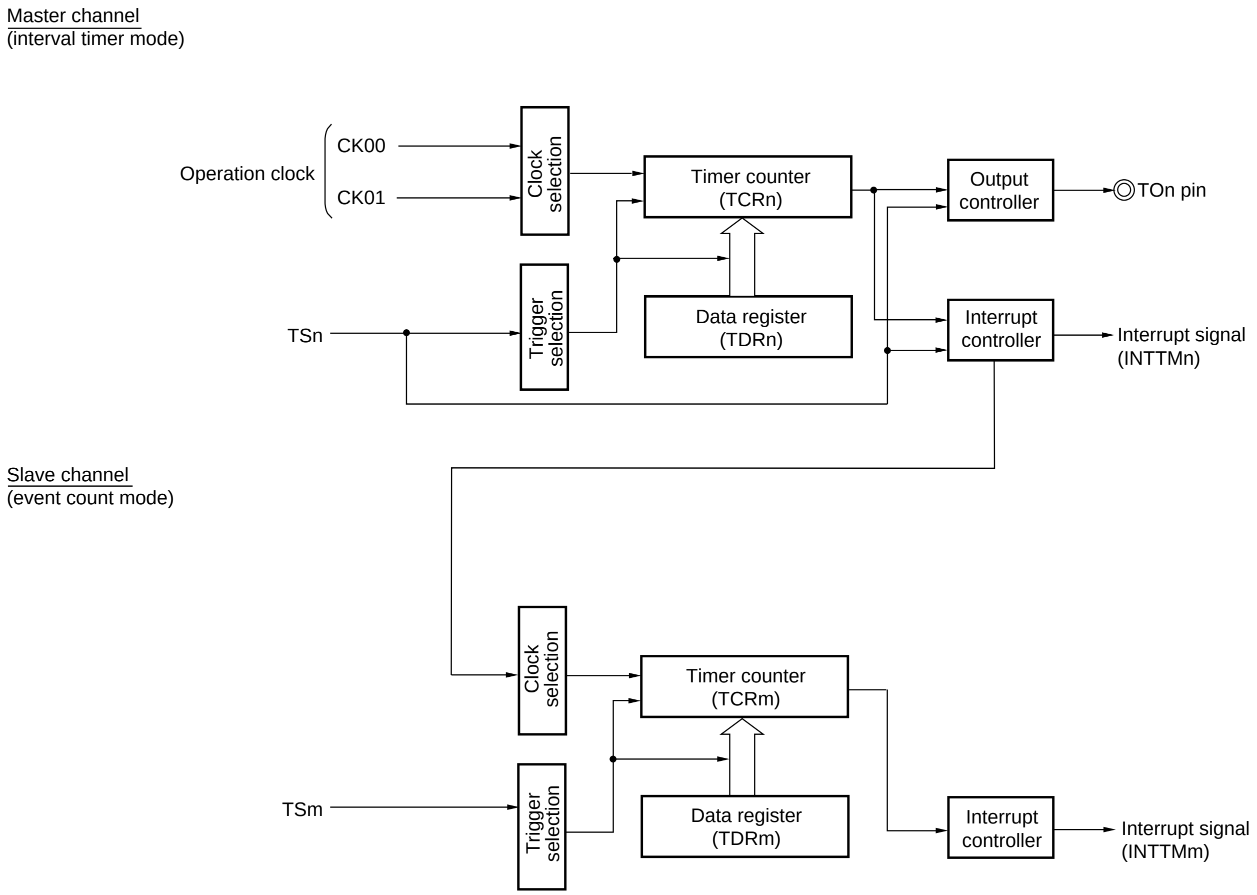
TOn cannot be used, because TOn becomes an irregular waveform that depends on external events.

TDRn of the master channel becomes valid from the next start timing (master channel INTTMn generation).

TDRm of the slave channel becomes valid from the next start timing (slave channel INTTMm generation).

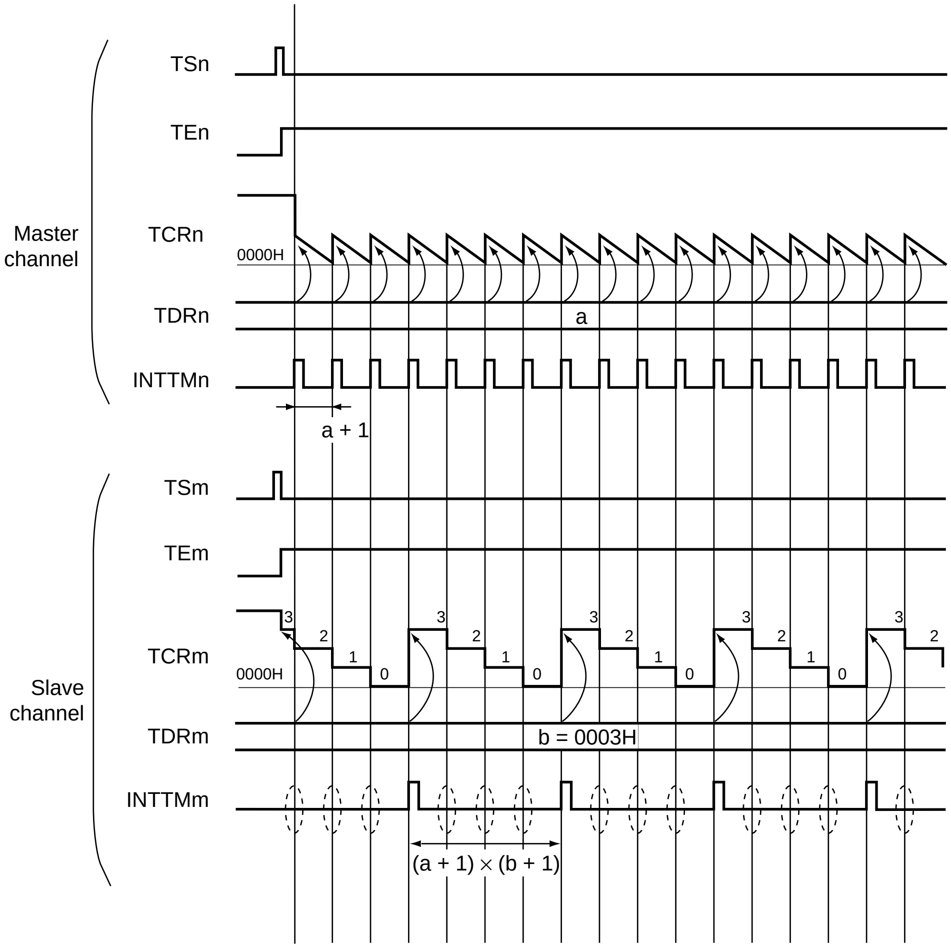
Remark n = 00
m = 01

Figure 7-44. Block Diagram of Operation as Interrupt Signal Thinning Function



Remark n = 00
m = 01

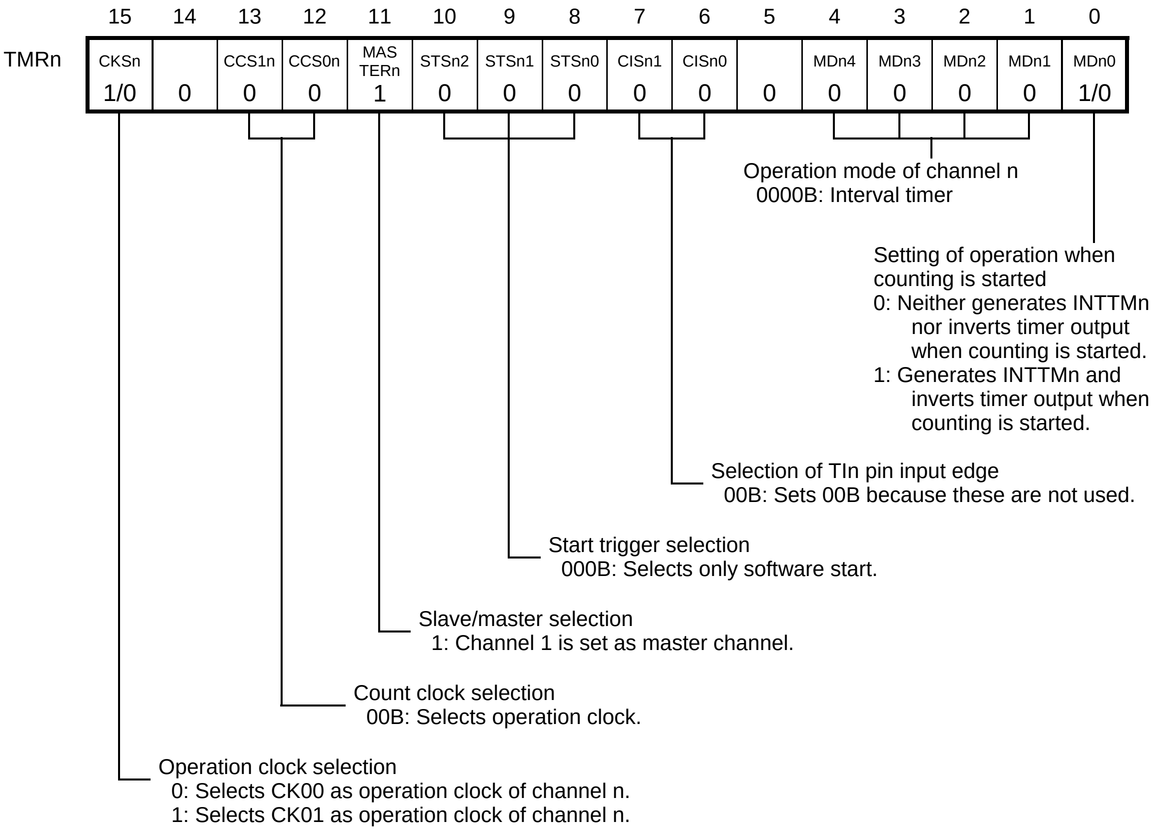
Figure 7-45. Example of Basic Timing of Operation as Interrupt Signal Thinning Function



Remark n = 00
m = 01

Figure 7-46. Example of Set Contents of Registers When Interrupt Signal Thinning Function (Master Channel) Is Used

(a) Timer mode register n (TMRn)



Operation mode of channel n
0000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMn nor inverts timer output when counting is started.
1: Generates INTTMn and inverts timer output when counting is started.

Selection of TIn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

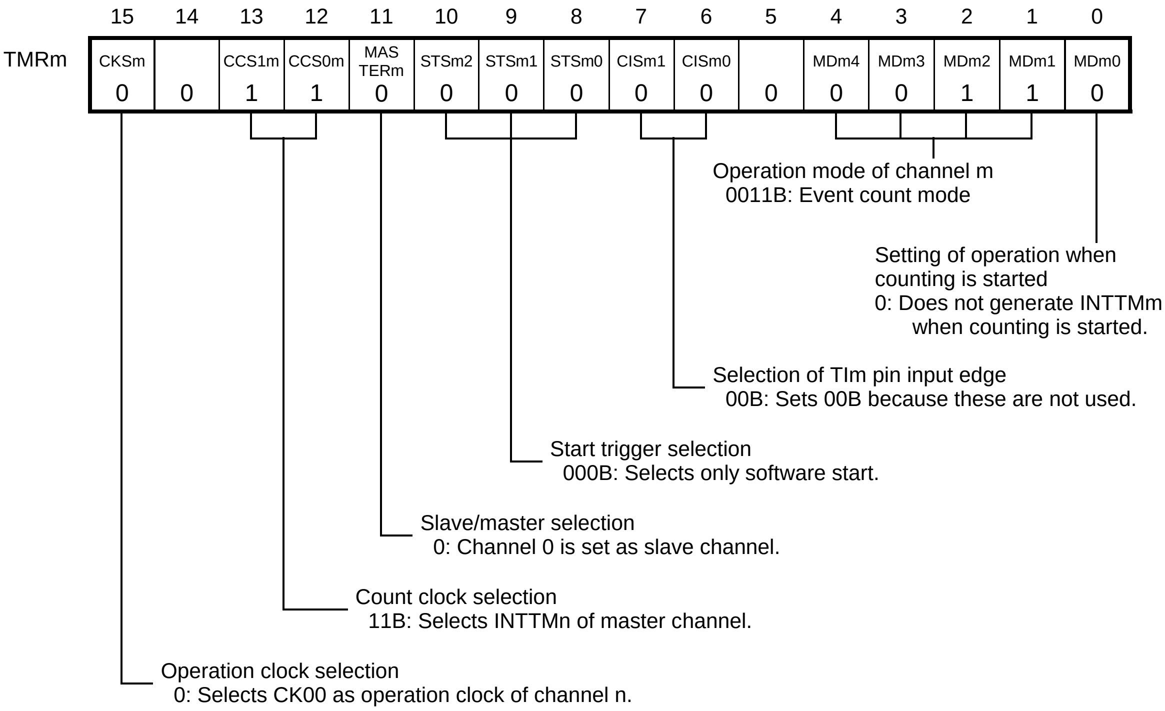
(b) Other registers

TOE0:TOEn	0: Stops the TOn output operation by counting operation.
TO0:TOn	0: Outputs a low level from TOn.
TOM0:TOMn	0: Sets toggle mode.
TOT0:TOTn	0: Sets 0 when TOMn = 0 (toggle mode).
TOL0:TOLn	0: Sets 0 when TOMn = 0 (toggle mode).
TDE0:TDEn	0: Stops dead time control.
TRE0:TREn	0: Stops real-time output.
TRO0:TROn	0: Sets 0 when TREn = 0 (stops real-time output).
TRC0:TRCn	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEn	0: Stops modulated output.

Remark n = 00

Figure 7-47. Example of Set Contents of Registers When Interrupt Signal Thinning Function (Slave Channel) Is Used

(a) Timer mode register m (TMRm)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRm	CKSm		CCS1m	CCS0m	MAS TERm	STSm2	STSm1	STSm0	CISm1	CISm0		MDm4	MDm3	MDm2	MDm1	MDm0
	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0

Operation mode of channel m
0011B: Event count mode

Setting of operation when counting is started
0: Does not generate INTTMm when counting is started.

Selection of TIm pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
11B: Selects INTTMn of master channel.

Operation clock selection
0: Selects CK00 as operation clock of channel n.

(b) Other registers

TOE0:TOEm	0: Stops the TOm output operation by counting operation.
TO0:TOm	0: Outputs a low level from TOm.
TOM0:TOMm	0: Sets 0 when TOEm = 0 (stops the TOm operation by counting operation).
TOT0:TOTm	0: Sets 0 when TOMm = 0 (toggle mode).
TOL0:TOLm	0: Sets 0 when TOMm = 0 (toggle mode).
TDE0:TDEm	0: Stops dead time control.
TRE0:TREm	0: Stops real-time output.
TRO0:TROm	0: Sets 0 when TREm = 0 (stops real-time output).
TRC0:TRCm	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEm	0: Stops modulated output.

Remark n = 00
m = 01

Figure 7-48. Operation Procedure When Interrupt Signal Thinning Function Is Used

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN and TAUOPEN bits of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMRn and TMRm registers of two channels to be used (determines operation mode of channels). An interval (period) value is set to the TDRn register of the master channel, and the number of interrupts to be thinned is set to the TDRm register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TOEn bit (master channel) to 1 (only when operation is resumed). Sets the TSn (master) and TSm (slave) bits to 1 at the same time. The TSn and TSm bits automatically return to 0 because they are trigger bits.	TEn = 1, TEm = 1 At the master channel, TCRn loads the value of TDRn by count clock input. INTTMn is generated and TOn is toggled when the MDn0 bit of the TMRn register is 1. At the slave channel, TCRm loads the value of TDRm and enters a state to wait for detection of INTTMn of the master channel.
During operation	The set values of the TDRn and TDRm registers can be changed. The TCRn and TCRm registers can always be read.	The counter (TCRn) of the master channel counts down. When the count value reaches TCRn = 0000H, the value of TDRn is loaded to TCRn again and the count operation is continued. By detecting TCRn = 0000H, INTTMn is generated and TOn performs toggle operation. The counter (TCRm) of the slave channel counts down every time an INTTMn signal of the master channel is detected. When the count value reaches TCRm = 0000H, the value of TDRm is loaded to TCRm again and the count operation is continued. By detecting TCRm = 0000H, INTTMm is generated.
Operation stop	The TTn (master) and TTm (slave) bits are set to 1 at the same time. The TTn and TTm bits automatically return to 0 because they are trigger bits.	TEn and TEm = 0, and count operation stops. TCRn and TCRm hold count values and stop.
TAUS stop	The TAU0EN and TAUOPEN bits of the PER2 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOn and TOm bits are cleared to 0 and the TOn and TOm pins are set to port mode.)

Operation is resumed.

7.4.8 Operation as A/D conversion trigger output function (type 1)

The A/D conversion trigger output function uses two channels in combination to output A/D conversion triggers.

It outputs A/D conversion trigger signals from slave channels.

Multiple slave channels can be used to increase the number of A/D conversion trigger outputs.

The A/D conversion trigger output function assumes the master channel to be used as a sub-function of the 6-phase PWM output function (7.4.3). The setting of the master channel is therefore the same as in 6-phase PWM output function (7.4.3). The A/D conversion trigger pulse generation period can be calculated by the following expression.

A/D conversion trigger pulse generation period (interval from the start of the carrier period to INTTMn detection)
= {Set value of TDRm (slave) + 1} × Count clock period

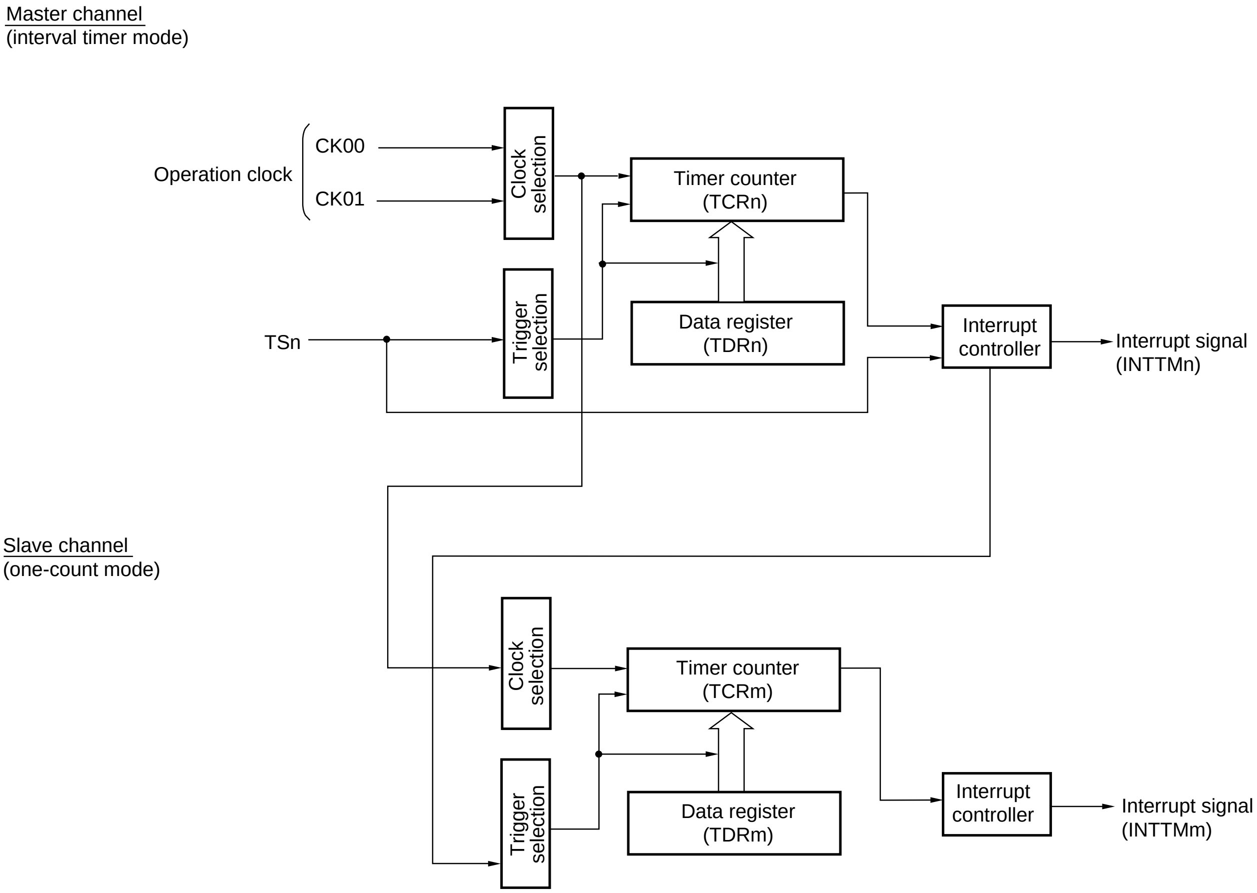
TCRn of the master channel operates in the interval timer mode and counts the periods.

TCRm of the slave channel operates in one-count mode and counts the duty. TCRm of the master channel loads the value of TDRm by using INTTMn of the master channel as a start trigger, and counts down. When TCRm has become 0000H, TCRm outputs INTTMm and stops counting until the next start trigger (INTTMn of the master channel) is input.

TDRn and TDRm of the master channel and slave channel become valid from the next period (master channel INTTMn generation).

Remark When n = 00: m = 01 or m = 08, 09
When n = 04: m = 05

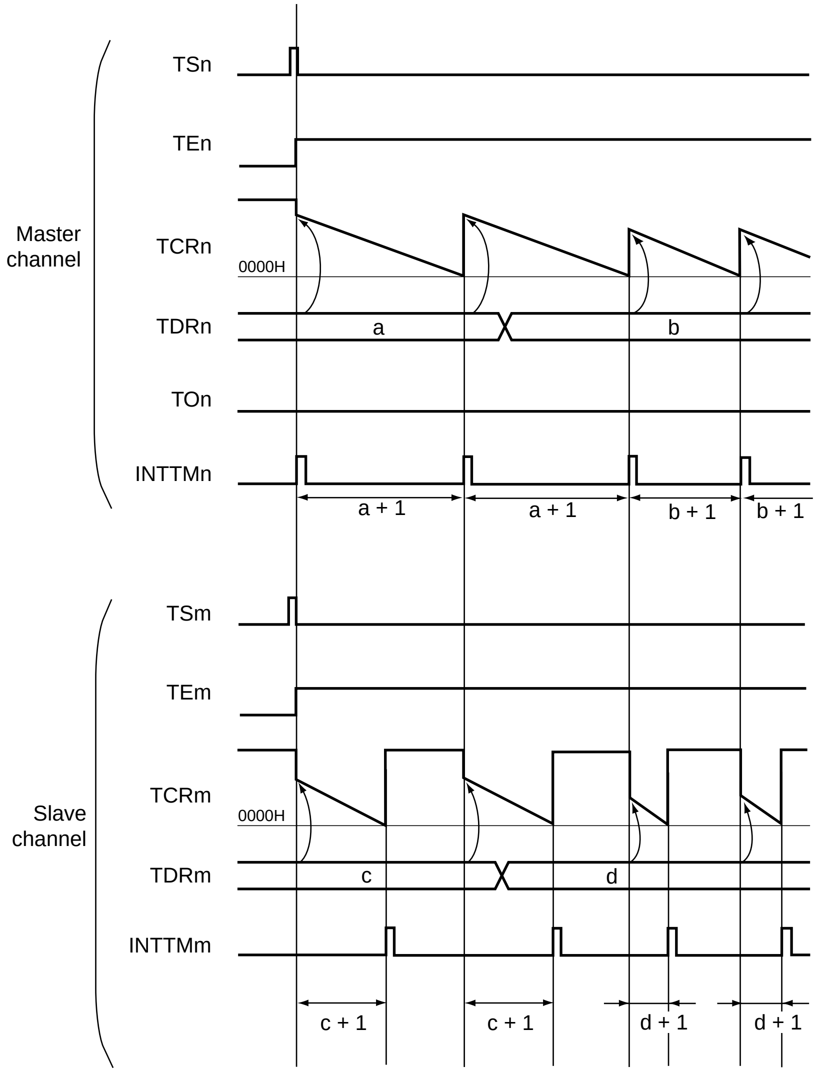
Figure 7-49. Block Diagram of Operation as A/D Conversion Trigger Output Function (Type 1)



Remark When n = 00: m = 01 or m = 08, 09

When n = 04: m = 05

Figure 7-50. Example of Basic Timing of Operation as A/D Conversion Trigger Output Function (Type 1)

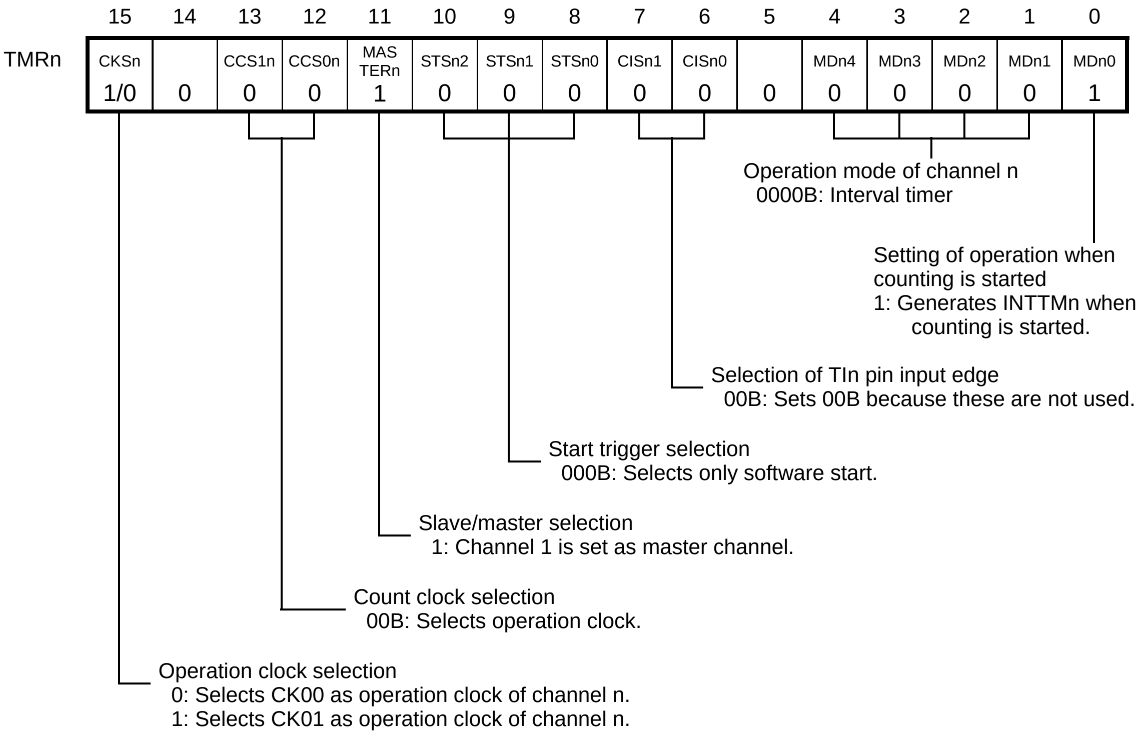


Remark When n = 00: m = 01 or m = 08, 09

When n = 04: m = 05

Figure 7-51. Example of Set Contents of Registers When A/D Conversion Trigger Output Function (Type 1) (Master Channel) Is Used

(a) Timer mode register n (TMRn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
0000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMn when counting is started.

Selection of TIn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

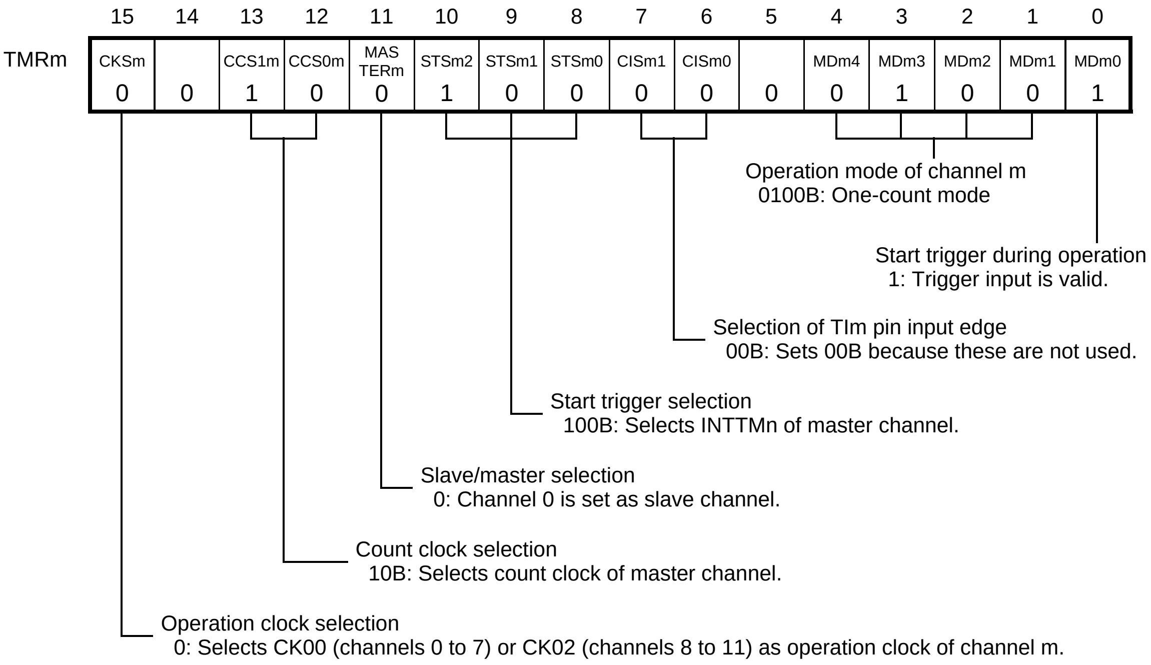
(b) Other registers

TOE0:TOEn	0: Stops the TOn output operation by counting operation.
TO0:TOn	0: Outputs a low level from TOn.
TOM0:TOMn	0: Sets 0 when TOEn = 0 (stops the TOn output operation by counting operation).
TOT0:TOTn	0: Sets 0 when TOMn = 0 (toggle mode).
TOL0:TOLn	0: Sets 0 when TOMn = 0 (toggle mode).
TDE0:TDEn	0: Stops dead time control.
TRE0:TREn	0: Stops real-time output.
TRO0:TROn	0: Sets 0 when TREn = 0 (stops real-time output).
TRC0:TRCn	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEn	0: Stops modulated output.

Remark When n = 00:m = 01 or m = 08, 09
When n = 04:m = 05

Figure 7-52. Example of Set Contents of Registers When A/D Conversion Trigger Output Function (Type 1) (Slave Channel) Is Used

(a) Timer mode register m (TMRm)



Operation mode of channel m
0100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TIm pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMn of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
10B: Selects count clock of master channel.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel m.

(b) Other registers

TOE0:TOEm	0: Stops the TOn output operation by counting operation.
TO0:TOm	0: Outputs a low level from TOm.
TOM0:TOMm	0: Sets 0 when TOEm = 0 (stops the TOm output operation by counting operation).
TOT0:TOTm	0: Sets 0 when TOMm = 0 (toggle mode).
TOL0:TOLm	0: Sets 0 when TOMm = 0 (toggle mode).
TDE0:TDEm	0: Stops dead time control.
TRE0:TREm	0: Stops real-time output.
TRO0:TROm	0: Sets 0 when TREm = 0 (stops real-time output).
TRC0:TRCm	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEm	0: Stops modulated output.

Remark When n = 00: m = 01 or m = 08, 09
When n = 04: m = 05

Figure 7-53. Operation Procedure When A/D Conversion Trigger Output Function (Type 1) Is Used

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN and TAUOPEN bits of the PER2 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines the clock frequencies of CK00 and CK01 for channels 0 to 7, and those of CK02 and CK03 for channels 8 to 11.	
Channel default setting	Sets the TMRn and TMRm registers of two channels to be used (determines operation mode of channels). An interval (period) value is set to the TDRn register of the master channel, and a duty factor is set to the TDRm register of the slave channel.	Channel operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSn (master) and TSm (slave) bits of the TS0 register to 1 at the same time. → The TSn and TSm bits automatically return to 0 because they are trigger bits.	TEn = 1, TEm = 1 The master channel starts counting and INTTMn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	The set values of the TDRn and TDRm registers can be changed after generation of INTTMn of the master channel. The TCRn and TCRm registers can always be read.	At the master channel, TCRn loads the value of TDRn and counts down. When the count value reaches TCRn = 0000H, INTTMn is generated. At the same time, the value of TDRn is loaded to TCRn, and the counter starts counting down again. At the slave channel, the value of the TDRm register is transferred to TCRm, triggered by the INTTMn signal of the master channel, and the counter starts counting down. INTTMm is generated when TCRm = 0000H is detected, and the count operation is stopped. After that, the above operation is repeated.
Operation stop	The TTn (master) and TTm (slave) bits are set to 1 at the same time. → The TTn and TTm bits automatically return to 0 because they are trigger bits.	TEn and TEm = 0, and count operation stops. TCRn and TCRm hold count values and stop.
TAUS stop	The TAU0EN and TAUOPEN bits of the PER2 register are cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark When n = 00: m = 01 or m = 08, 09

When n = 04: m = 05

7.4.9 Operation as A/D conversion trigger output function (type 2)

The A/D conversion trigger output function uses two channels in combination to output A/D conversion triggers.

It outputs A/D conversion trigger signals from slave channels.

Multiple slave channels can be used to increase the number of A/D conversion trigger outputs.

The A/D conversion trigger output function assumes the slave channel to be used as a sub-function of the 6-phase triangular wave PWM output function (7.4.6). The setting of the master channel is therefore the same as in 6-phase triangular wave PWM output function (7.4.6). The A/D conversion trigger pulse generation period can be calculated by the following expression.

A/D conversion trigger pulse generation period (interval from the start of the carrier period to INTTMn detection during a down status) = {Set value of TDRm (slave) + 1} × Count clock period

Setting range of TDRm (slave): 0000H < TDRm (slave) < {Set value of TDRn (master) + 1}

* Interval from INTTMm detection during a down status to INTTMm detection during an up status
= {{Set value of TDRn (master) + 1} – {Set value of TDRm (slave)}} × 2 × Count clock period

TCRn of the master channel operates in the interval timer mode and counts the periods.

TCRn loads the value of TDRn by setting the channel start trigger bit (TSn) to 1.

At this time, INTTMn is not output and TOn is not toggled when MDn0 of TMRn is 0. INTTMn is output and TOn is toggled when MDn0 of TMRn is 1. Afterward, TCRn counts down along with the count clock. When TCRn has become 0000H, INTTMn is output and TOn is toggled upon the next count clock. TCRn loads the value of TDRn again at the same timing. Similar operation is continued hereafter.

A carrier period is generated in two periods of the master channel count.

The count operation of the slave channel is controlled by defining the first period of the master channel as a down status of the slave channel and the second period as an up status of the slave channel.

TOn of the master channel outputs up and down statuses.

TOn of the TO0 register must be manipulated while TOEn of the TOE0 register is 0 and the default level must be set, because up and down statuses are output.

TOn of the TO0 register is set to 1 when MDn0 of the TMR0 register is 0, and TOn is set to 0 when MDn0 is 1.

By setting the default level, a high level is output from TOn during a down status and a low level is output during an up status.

TCRm of slave channel m operates in the up and down count mode, and counts the duty. TRm loads the value of TDRm at the first count clock, after the channel start trigger bit (TSm) is set to 1. Hereafter, counting up and counting down is switched in accordance with the operation of the master channel. INTTMm is output when TCRm becomes 0001H.

TCRm loads the value of TDRm again when INTTMn is generated in an up status of the master channel. Similar operation is continued hereafter.

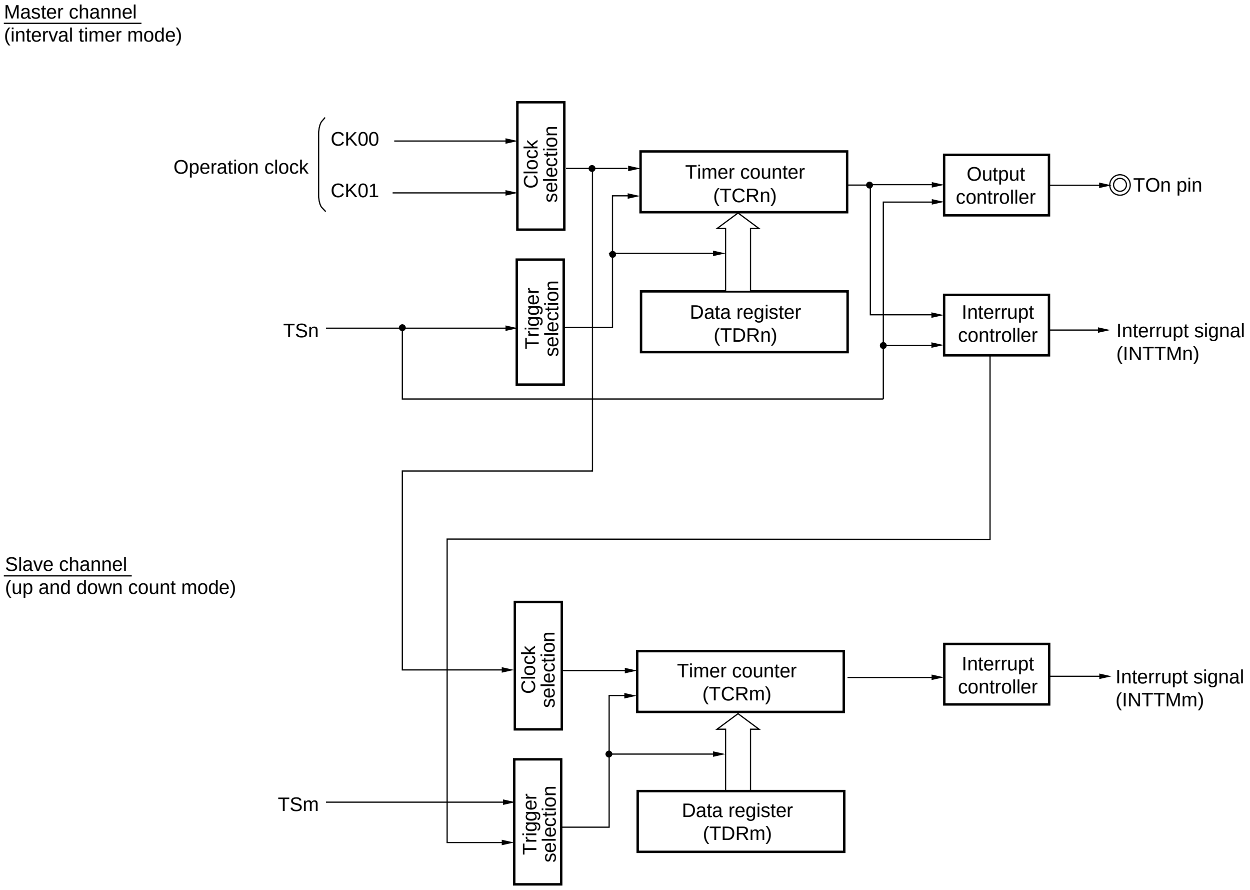
Remark When n = 00: m = 01 or m = 08, 09
When n = 04: m = 05

Caution TDRn of the master channel must be rewritten during an up status period of slave channel m. When the value of TDRn is rewritten during a down status period, the periods of the down status and up status differ and an expected waveform cannot be output, because the value of TDRn of the rewritten master channel becomes valid at the next period.

TDRm of the slave channel becomes valid from the next carrier period (up and down trigger detection).

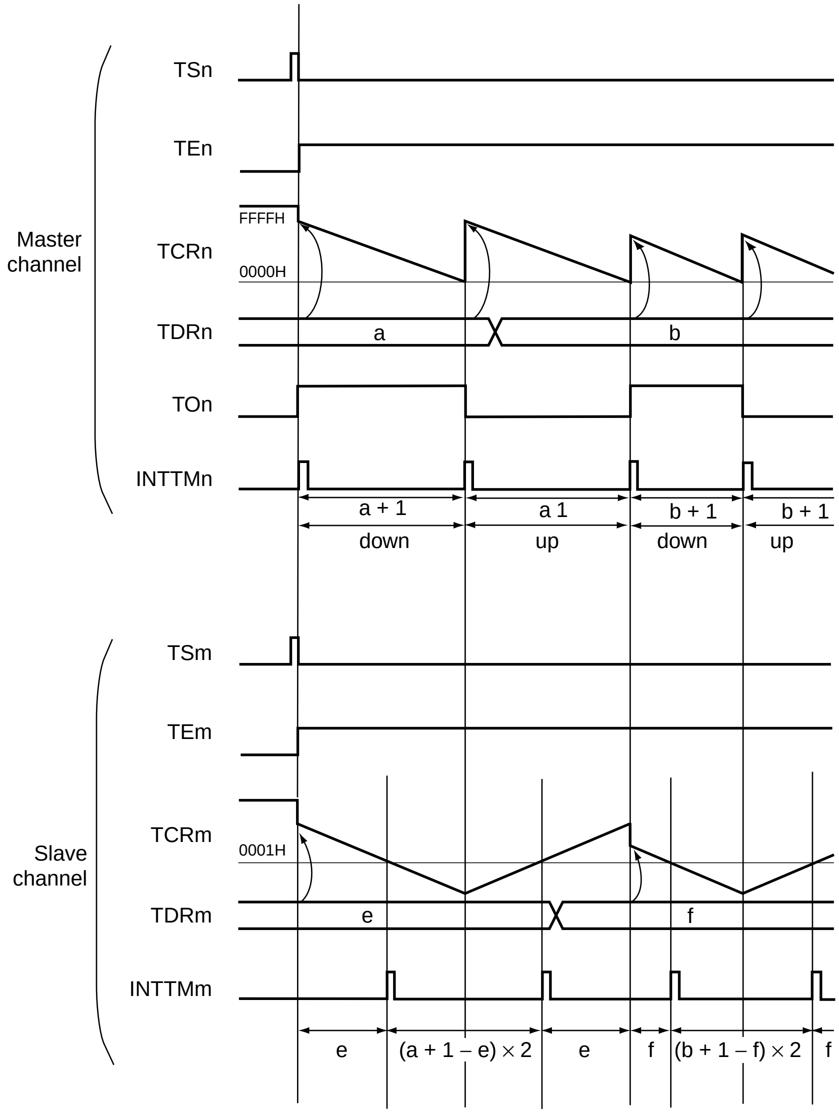
Remark When n = 00: m = 01 or m = 08, 09
When n = 04: m = 05

Figure 7-54. Block Diagram of Operation as A/D Conversion Trigger Output Function (Type 2)



Remark When n = 00: m = 01 or m = 08, 09
When n = 04: m = 05

Figure 7-55. Example of Basic Timing of Operation as A/D Conversion Trigger Output Function (Type 2)

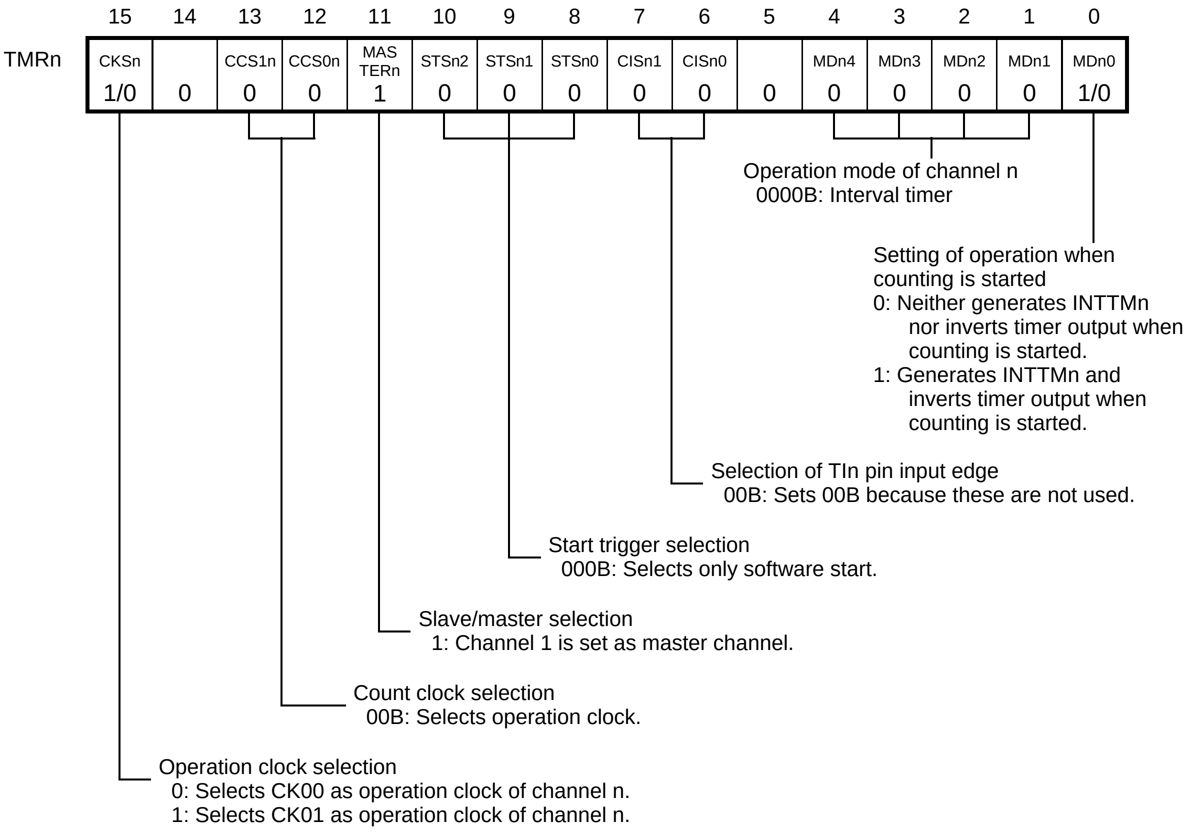


Remark When n = 00: m = 01 or m = 08, 09

When n = 04: m = 05

Figure 7-56. Example of Set Contents of Registers When A/D Conversion Trigger Output Function (Type 2) (Master Channel) Is Used

(a) Timer mode register n (TMRn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn		CCS1n	CCS0n	MAS TERn	STSn2	STSn1	STSn0	CISn1	CISn0		MDn4	MDn3	MDn2	MDn1	MDn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel n
0000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMn nor inverts timer output when counting is started.
1: Generates INTTMn and inverts timer output when counting is started.

Selection of TIn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
00B: Selects operation clock.

Operation clock selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

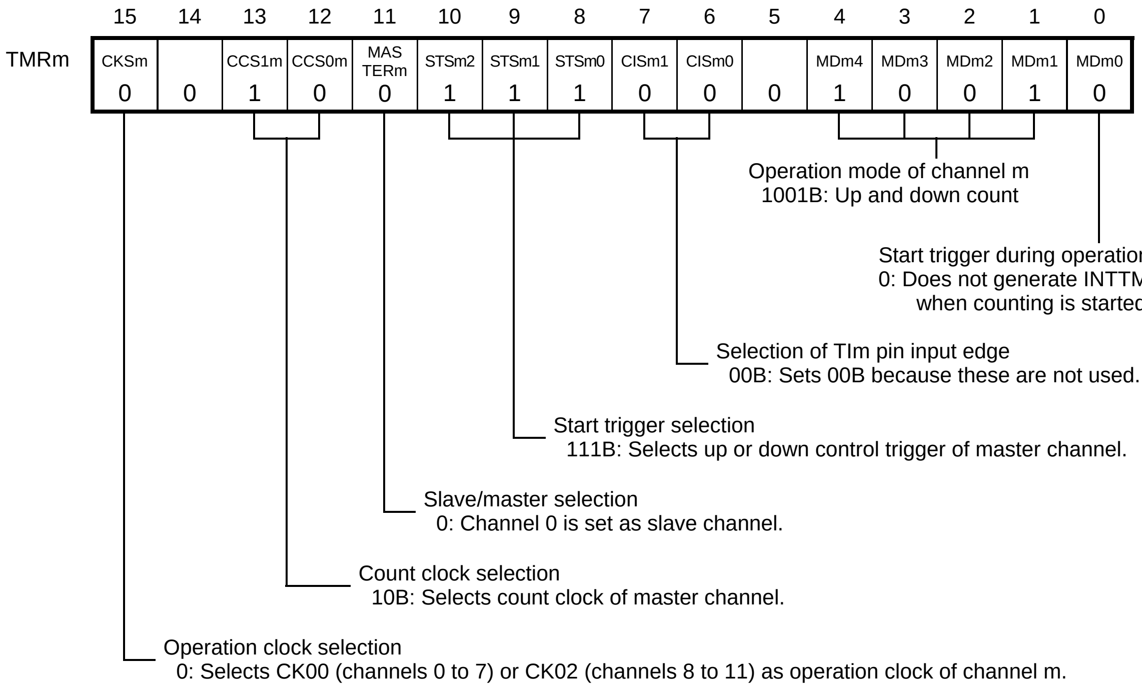
(b) Other registers

TOE0:TOEn	0: Stops the TOn output operation by counting operation. 1: Enables the TOn output operation by counting operation.
TO0:TOn	0: Outputs a low level from TOn. 1: Outputs a high level from TOn.
TOM0:TOMn	0: Sets toggle mode.
TOT0:TOTn	0: Sets 0 when TOMn = 0 (toggle mode).
TOL0:TOLn	0: Sets 0 when TOMn = 0 (toggle mode).
TDE0:TDEn	0: Stops dead time control.
TRE0:TREn	0: Stops real-time output.
TRO0:TROn	0: Sets 0 when TREn = 0 (stops real-time output).
TRC0:TRCn	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEn	0: Stops modulated output.

Remark When n = 00: m = 01 or m = 08, 09
When n = 04: m = 05

Figure 7-57. Example of Set Contents of Registers When A/D Conversion Trigger Output Function (Type 2) (Slave Channel m) Is Used

(a) Timer mode register m (TMRm)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRm	CKSm		CCS1m	CCS0m	MAS TERm	STSm2	STSm1	STSm0	CISm1	CISm0		MDm4	MDm3	MDm2	MDm1	MDm0
	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0

Operation mode of channel m
1001B: Up and down count

Start trigger during operation
0: Does not generate INTTMm when counting is started.

Selection of TIm pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
111B: Selects up or down control trigger of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
10B: Selects count clock of master channel.

Operation clock selection
0: Selects CK00 (channels 0 to 7) or CK02 (channels 8 to 11) as operation clock of channel m.

(b) Other registers

TOE0:TOEm	0: Stops the TOn output operation by counting operation.
TO0:TOm	0: Outputs a low level from TOm.
TOM0:TOMm	0: Sets 0 when TOEm = 0 (stops the TOn operation by counting operation).
TOT0:TOTm	0: Sets 0 when TOMm = 0 (toggle mode).
TOL0:TOLm	0: Sets 0 when TOMm = 0 (toggle mode).
TDE0:TDEm	0: Stops dead time control.
TRE0:TREm	0: Stops real-time output.
TRO0:TROm	0: Sets 0 when TREm = 0 (stops real-time output).
TRC0:TRCm	0: Does not operate as the real-time output trigger generation channel.
TME0:TMEm	0: Stops modulated output.

Remark When n = 00: m = 01 or m = 08, 09
When n = 04: m = 05

Figure 7-58. Operation Procedure When A/D Conversion Trigger Output Function (Type 2) Is Used (1/2)

Operation is resumed. (from next page)

	Software Operation	Hardware Status
TAUS default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN and TAUOPEN bits of the PER2 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMRn and TMRm registers of two channels to be used (determines operation mode of channels). An interval (period) value is set to the TDRn register of the master channel, and an interrupt width is set to the TDRm register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.) The TOn pin goes into Hi-Z output states.
	Sets the master channel. Sets the TOMn bit of the TOM0 register to 0 (toggle mode). Sets the TDEn bit of the TDE0 register to 0 (dead time control enable).	
	Sets the TOn bit, and determines default levels of the TOn bit. Sets the TOEn bit to 1 and enables operation of TOn. Clears the port register and port mode register to 0.	The TOn default setting level is output when the port mode register is in output mode and the port register is 0. TOn does not change because channel stop operating. The TOn pin outputs the set level of TOn.
Operation start	Sets the TOEn (master) bit to 1 (only when operation is resumed). Sets the TSn (master) and TSm (slave) bits of the TS0 register to 1 at the same time. The TSn and TSm bits automatically return to 0 because they are trigger bits.	TEn = 1, TEm = 1 When the master and slave channels start counting and the MDn0 bit of the TMRn register is 1, INTTMn is generated.
During operation	The set value of the TDRn (master) register must be changed during an up status period. The set value of the TDRm (slave) register can be changed. The TCRn and TCRm registers can always be read. The TSRm (slave) register can always be read.	At the master channel, TCRn loads the value of TDRn and counts down. When the count value reaches TCRn = 0000H, INTTMn is generated. At the same time, the value of TDRn is loaded to TCRn, and the counter starts counting down again. At the slave channel, TCRm loads the value of TDRm, and counting down and up are switched according to the operation of the master channel. INTTMm is generated and count operation is stopped upon detection of TCRm = 0001H. TCRm loads the value of TDRm again and count operation is continued by the generation of INTTMn during an up status of the master channel.

Remark When n = 00: m = 01 or m = 08, 09
When n = 04: m = 05

Operation is resumed.
(to forward page)

Figure 7-58. Operation Procedure When A/D Conversion Trigger Output Function (Type 2) Is Used (2/2)

	Software Operation	Hardware Status
Operation stop	The TTn (master) and TTm (slave) bits are set to 1 at the same time. The TTn and TTm bits automatically return to 0 because they are trigger bits.	TEn and TEm = 0, and count operation stops. TCRn and TCRm hold count values and stop. The TOn output is not initialized but hold current statuses.
	Sets the TOEn (master) bit to 0, TOn bit to value.	The TOn pin outputs the set level of TOn.
TAUS stop	To hold the TOn pin output level Clears the TOn bit to 0 after the value to be held is set to the port register.	The TOn pin output level is held by port function.
	When holding the TOn pin output level is not necessary Switches the port mode register to input mode.	The TOn pin output level goes into Hi-Z output states.
	The TAU0EN and TAUOPEN bits of the PER2 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOn bit is cleared to 0 and the TOn pin is set to port mode.)

Remark When n = 00: m = 01 or m = 08, 09
When n = 04: m = 05

CHAPTER 8 COMPARATORS/OPERATIONAL AMPLIFIERS

8.1 Functions of Comparator and Operational Amplifier

The operational amplifiers and comparators have the following functions.

○ Comparators

- A comparator is equipped with two channels (CMP0, CMP1).
- Negative-side input pins (CMP0M, CMP1M) and a positive-side input pin (CMP0P, CMP1P) can be connected.
- The output signal of an operational amplifier can be used as the positive-side input signal of a comparator. (In this case, the output signal is simultaneously input to both channels of comparators 0 and 1.)
- CMP0M and CMP1M pin inputs and the internal generation reference voltage (6 combinations for each comparator) can be selected as the reference voltage.
- The elimination width of the noise elimination digital filter can be selected.
- An interrupt request is generated when an overvoltage is detected (INTCMP0 and INTCMP1).
- The output signal of a comparator is connected to the timer array unit and sets the timer output pin (TOn) to a Hi-Z state.

○ Operational amplifiers

- An operational amplifier amplifies and outputs an analog voltage that is input. One among eight amplification factors can be selected.
- The output signal of an operational amplifier can be used as the positive-side input signal of a comparator. (In this case, the output signal is simultaneously input to both channels of comparators 0 and 1.)
- The output signal of an operational amplifier can be selected as the analog input of an A/D converter.

○ Overvoltage detection function

The timer output pin (TOn) can be set to a Hi-Z state while an overcurrent flows by using an operational amplifier and a comparator. Furthermore, a function to set the pin to a Hi-Z state can be selected from the following two functions.

- 2-stage overcurrent detection function
- Overcurrent/electromotive force detection function

Both functions can be used by setting two values (comparator 0 < comparator 1) of the reference voltage that set the pin to a Hi-Z state, and inputting the same signal to the positive-side inputs of comparators 0 and 1.

- 2-stage overcurrent detection function

  <1> Reference voltage of comparator 0 < input signal voltage < reference voltage of comparator 1

  → Set the TOn pin to a Hi-Z state.

  Timer output is automatically restarted in synchronization with the timer cycle when the input signal voltage is lower than the reference voltage of comparator 0.

<2> Reference voltage of comparator 1 < input signal voltage

→ Set the TOn pin to a Hi-Z state.

Timer output is automatically restarted in synchronization with the next timer cycle by setting a register, when the input signal voltage is lower than the reference voltage of comparator 0.

- Overcurrent/electromotive force detection function

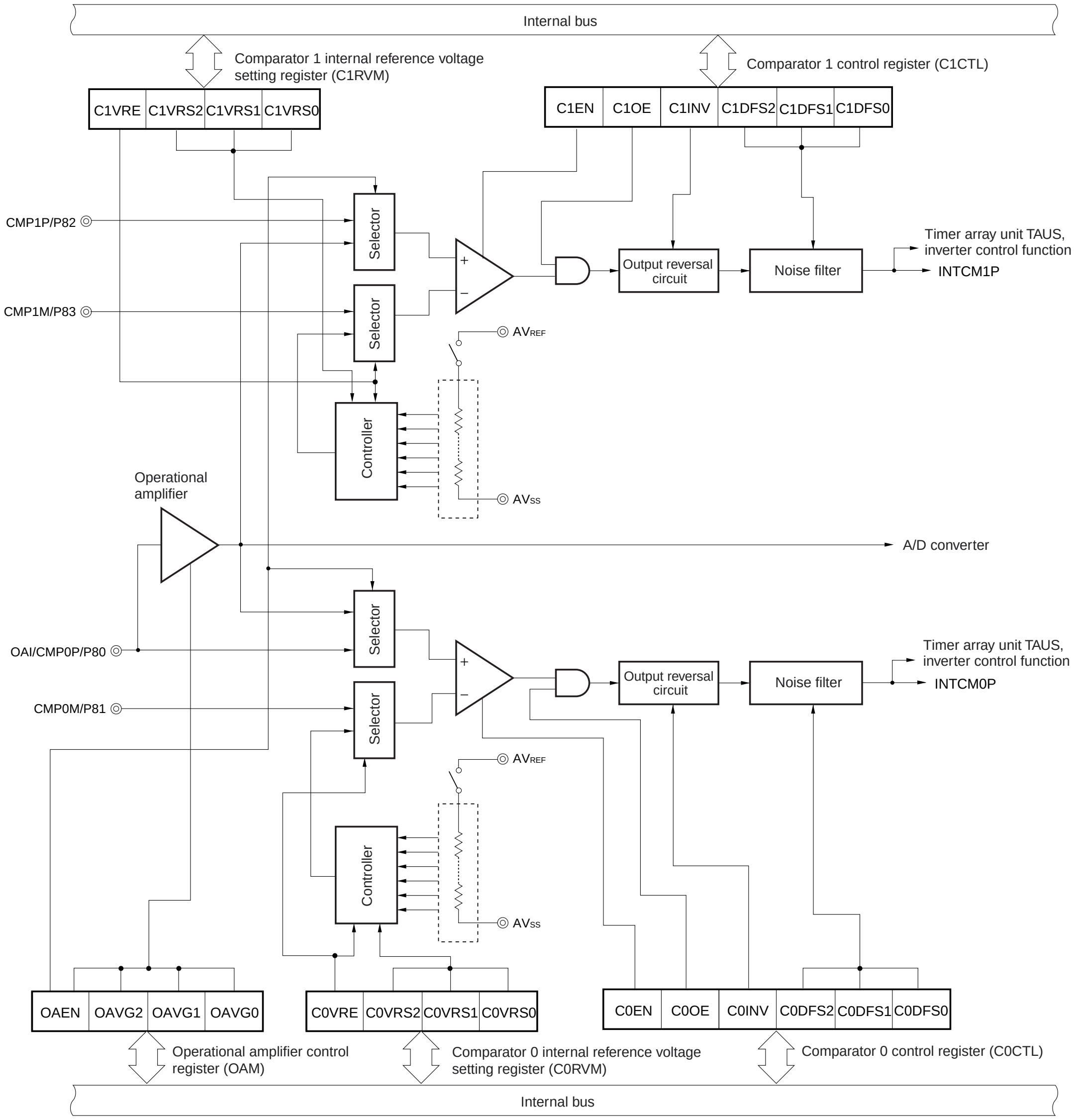
<1> Input signal voltage (electromotive force) < comparator 0, or comparator 1 < input signal voltage (overcurrent)

→ Set the TOn pin to a Hi-Z state.

Timer output is automatically restarted in synchronization with the timer cycle when the input signal voltage is higher than the reference voltage of comparator 0 or when it is lower than the reference voltage of comparator 1.

Remark n: Timer channel number (n = 02 to 07)

Figure 8-1. Block Diagram of Comparator/Operational Amplifier



8.2 Configurations of Comparator and Operational Amplifier

The comparators and operational amplifiers consist of the following hardware.

Table 8-1. Configurations of Comparator and Operational Amplifier

Item	Configuration
Control registers	Peripheral enable register 1 (PER1) Operational amplifier control register (OAM) Comparator 0 and 1 control registers (C0CTL, C1CTL) Comparator 0 and 1 internal reference voltage setting registers (C0RVM, C1RVM) Port input mode register 8 (PIM8) Port mode register 8 (PM8)

8.3 Registers Controlling Comparators and Operational Amplifiers

The comparators and operational amplifiers use the following eight registers.

- Peripheral enable register 1 (PER1)
- Operational amplifier control register (OAM)
- Comparator 0 and 1 control registers (C0CTL, C1CTL)
- Comparator 0 and 1 internal reference voltage setting registers (C0RVM, C1RVM)
- Port input mode register 8 (PIM8)
- Port mode register 8 (PM8)

(1) Peripheral enable register 1 (PER1)

This register is used to set whether each peripheral hardware macro can be used. Power consumption and noise are reduced by stopping the clock supply to unused hardware.

Make sure to set bit 3 (OACMPEN) to 1 to use a comparator or an operational amplifier.

PER1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Cautions 1. Make sure to set OACMPEN to 1 first, when setting the comparator or operational amplifier. Writing to the control register of the comparator or operational amplifier will be ignored and all values read will be initialized when OACMPEN is set to 0.

2. Make sure to set bits 0 to 2 and bits 4 to 7 of the PER1 register to “0”.

Figure 8-2. Format of Peripheral Enable Register 1 (PER1)

Address: F00F1H After reset: 00H R/W

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

OACMPEN	Comparator/operational amplifier input clock control
0	Stops input clock supply • The SFR used with the comparator or operational amplifier cannot be written. • The comparator or operational amplifier is reset.
1	Supplies an input clock • The SFR used with the comparator or operational amplifier can be read and written.

(2) Operational amplifier control register (OAM)

This register is used to enable or disable the operation of an operational amplifier and set the amplification factor.

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

Reset signal generation clears this register to 00H.

Figure 8-3. Format of Operational Amplifier Control Register (OAM)

Address: F0240H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
OAM	OAEN	0	0	0	0	OAVG2	OAVG1	OAVG0

OAEN	Operational amplifier operation control
0	Stops operation
1	Enables operation Enables external input from the operational amplifier input pin (OAI) Inputs the operational amplifier output signal as the positive-side input voltage of comparators 0 and 1

OAVG2	OAVG1	OAVG0	Input voltage amplification factor setting
0	0	0	×1
0	0	1	×4
0	1	0	×6
0	1	1	×8
1	0	0	×10
1	0	1	×12
Other than the above			Setting prohibited

Cautions 1. Set the amplification factor before enabling (OAEN = 1) the operation of the operational amplifier. Changing the amplification factor setting in the operation enabled state (OAEN = 1) is prohibited.

2. Comparators 0 and 1 may perform false detection when CnEN of the CnCTL register is set to 1, because the operational amplifier output signal becomes the positive-side input voltage of comparators 0 and 1, at the same time as when the operational amplifier enters the operation enable state. Therefore, set OAM while the comparator output signal is being stopped.

Remark n = 0, 1

(3) Comparator n control register (CnCTL)

This register is used to control the operation of comparator n, enable or disable comparator output, reverse the output, and set the noise elimination width.

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

Reset signal generation clears this register to 00H.

Figure 8-4. Format of Comparator n Control Register (CnCTL)

Address: F0241H (C0CTL), F0242H (C1CTL) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CnCTL	CnEN	0	0	CnOE	CnINV	CnDFS2	CnDFS1	CnDFS0

CnEN	Comparator operation control
0	Stops operation
1	Enables operation Enables input to the external pins on the positive and negative sides of comparator n

CnOE	Enabling or disabling of comparator output
0	Disables output (output signal = fixed to low level)
1	Enables output

CnINV	Output reversal setting
0	Forward
1	Reverse

CnDFS2	CnDFS1	CnDFS0	Noise elimination width setting (f_{CLK} = 20 MHz)
0	0	0	Noise filter unused
0	0	1	250 ns
0	1	0	500 ns
0	1	1	1 μs
1	0	0	2 μs
Other than the above			Setting prohibited

Cautions 1. Rewrite CnINV and CnDFS2 to CnDFS0 after setting the comparator output to the disabled state (CnOE = 0).

2. With the noise elimination width, an extra CPU clock (f_{CLK}) may be eliminated from the setting value.
(Example: When f_{CLK} = 20 MHz, CnDFS2 to CnDFS0 = 001, noise elimination width = 250 to 300 ns)

3. To operate the comparator in combination with an operational amplifier, set the operation of the comparator after setting the operation of the operational amplifier (see Figure 8-10 and Figure 8-11).

4. The negative-side external pin input of the comparator will be cutoff when CnVRE of the CnRVM register is set (1), regardless of the value that enables or disables the comparator operation (CnEN).

Remarks 1. f_{CLK}: CPU or peripheral hardware clock frequency

2. n = 0, 1

(4) Comparator n internal reference voltage selection register (CnRVM)

This register is used to set the internal reference voltage of comparator n. The internal reference voltage can be selected from six voltages that use AV_{REF}.

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

Reset signal generation clears this register to 00H.

Figure 8-5. Format of Comparator n Internal Reference Voltage Selection Register (CnRVM)

Address: F0243H (C0RVM), F0244H (C1RVM) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CnRVM	CnVRE	0	0	0	0	CnVRS2	CnVRS1	CnVRS0

CnVRE	Internal reference voltage operation control
0	Stops operation
1	Enables operation Connects the internal reference voltage to the negative-side input of comparator n

CnVRS2	CnVRS1	CnVRS0	Reference voltage setting	
			Reference voltage settable with comparator 0 (n = 0)	Reference voltage settable with comparator 1 (n = 1)
0	0	0	Setting prohibited	
0	0	1	$2AV_{REF}/16$	$3AV_{REF}/16$
0	1	0	$4AV_{REF}/16$	$5AV_{REF}/16$
0	1	1	$6AV_{REF}/16$	$7AV_{REF}/16$
1	0	0	$8AV_{REF}/16$	$9AV_{REF}/16$
1	0	1	$10AV_{REF}/16$	$11AV_{REF}/16$
1	1	0	$12AV_{REF}/16$	$13AV_{REF}/16$
1	1	1	Setting prohibited	

Cautions 1. The operation of the comparator is controlled by CnEN when the operation of the internal reference voltage is stopped (CnVRE = 0).

2. The negative-side external pin input of the comparator will be cutoff when CnVRE is set (1), regardless of the value that enables or disables the comparator operation (CnEN).

3. Set the reference voltage before enabling the operation of the internal reference voltage (CnVRE = 1). Changing the reference voltage setting in the operation enabled state (CnVRE = 1) is prohibited.

Remark n = 0, 1

(5) Port input mode register 8 (PIM8)

This register is used to enable or disable port 8 input in 1-bit units.

Set to input disable to use a comparator or an operational amplifier. Set to input enable to use the port function or the external interrupt function and timer Hi-Z control function, because input disable is set by default.

PIM8 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 Port Input Mode Register 8 (PIM8)

Address: F0048H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM8	0	0	0	0	PIM83	PIM82	PIM81	PIM80

PIM8n	Selection of enabling or disabling P8n pin input (n = 0 to 3)
0	Disables input
1	Enables input

(6) Port mode register 8 (PM8)

This register is used to set port 8 input or output in 1-bit units.

Set the PM80 to PM83 bits to 1 to use the P80/CMP0P/TMOFF0/INTP3/OAI, P81/CMP0M, P82/CMP1P/TMOFF1/INTP7, or P83/CMP1M pin as the positive-side or negative-side input function of the comparator, or the operational amplifier input function.

The output latches of P80 to P83 may be 0 or 1 at this time.

PM80 to PM83 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 8-7. Format of Port Mode Register 8 (PM8)

Address: FFF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	1	1	PM83	PM82	PM81	PM80

PM8n	P8n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution The port function that is alternatively used as the CMP0M, CMP1M pin can be used in the input mode, when the CMP0P, CMP1P pin is selected as the positive-side input of the comparator, and the internal reference voltage is used on the negative side. Using the output mode, however, is prohibited. Furthermore accessing port register 8 (P8) is also prohibited.

8.4 Operations of Comparator and Operational Amplifier

8.4.1 Starting comparator and operational amplifier operation

The procedures for starting the operation of a comparator and an operational amplifier are described below, separately for each use method.

○ Using only a comparator
- Using the external pin input for the comparator reference voltage (Figure 8-8)
- Using the internal reference voltage for the comparator reference voltage (Figure 8-9)

○ Using a comparator and an operational amplifier (using the operational amplifier output voltage as the comparator compare voltage input)
- Using the external pin input for the comparator reference voltage (Figure 8-10)
- Using the internal reference voltage for the comparator reference voltage (Figure 8-11)

○ Using the operational amplifier output voltage as the A/D converter analog input (Figure 8-12)

Figure 8-8. Using the External Pin Input for the Comparator Reference Voltage (Using Only a Comparator)

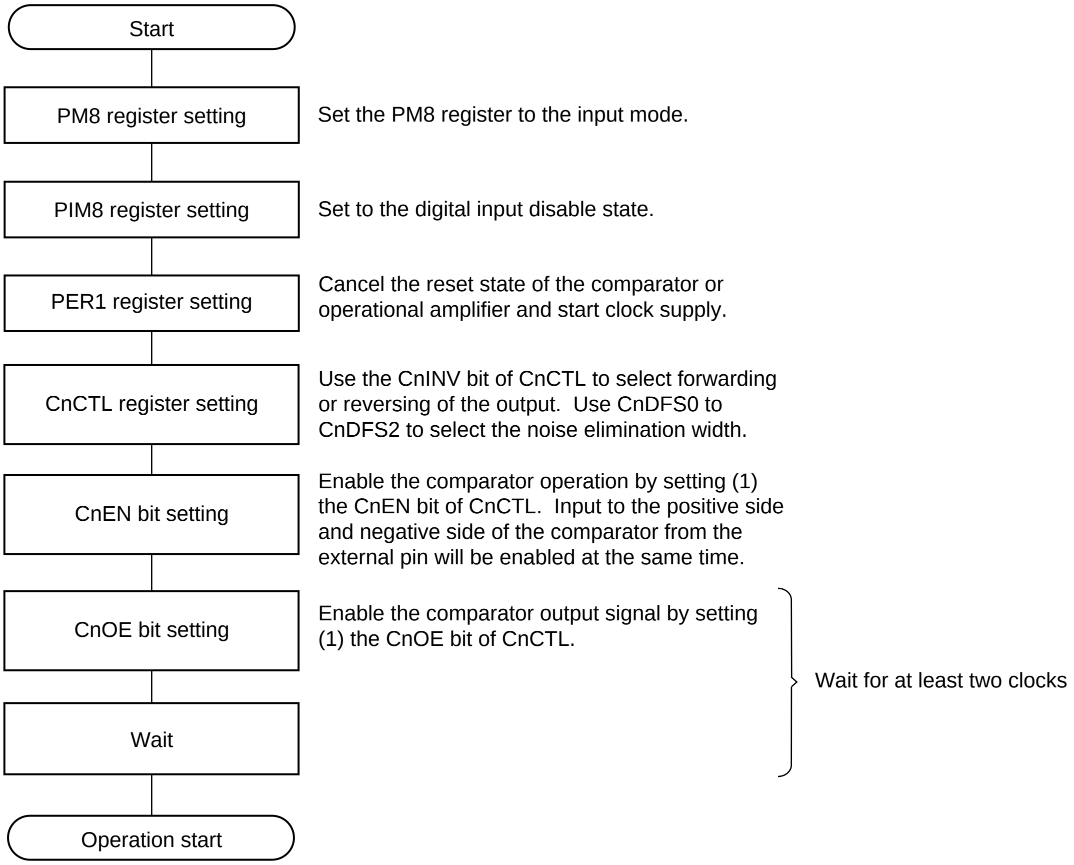
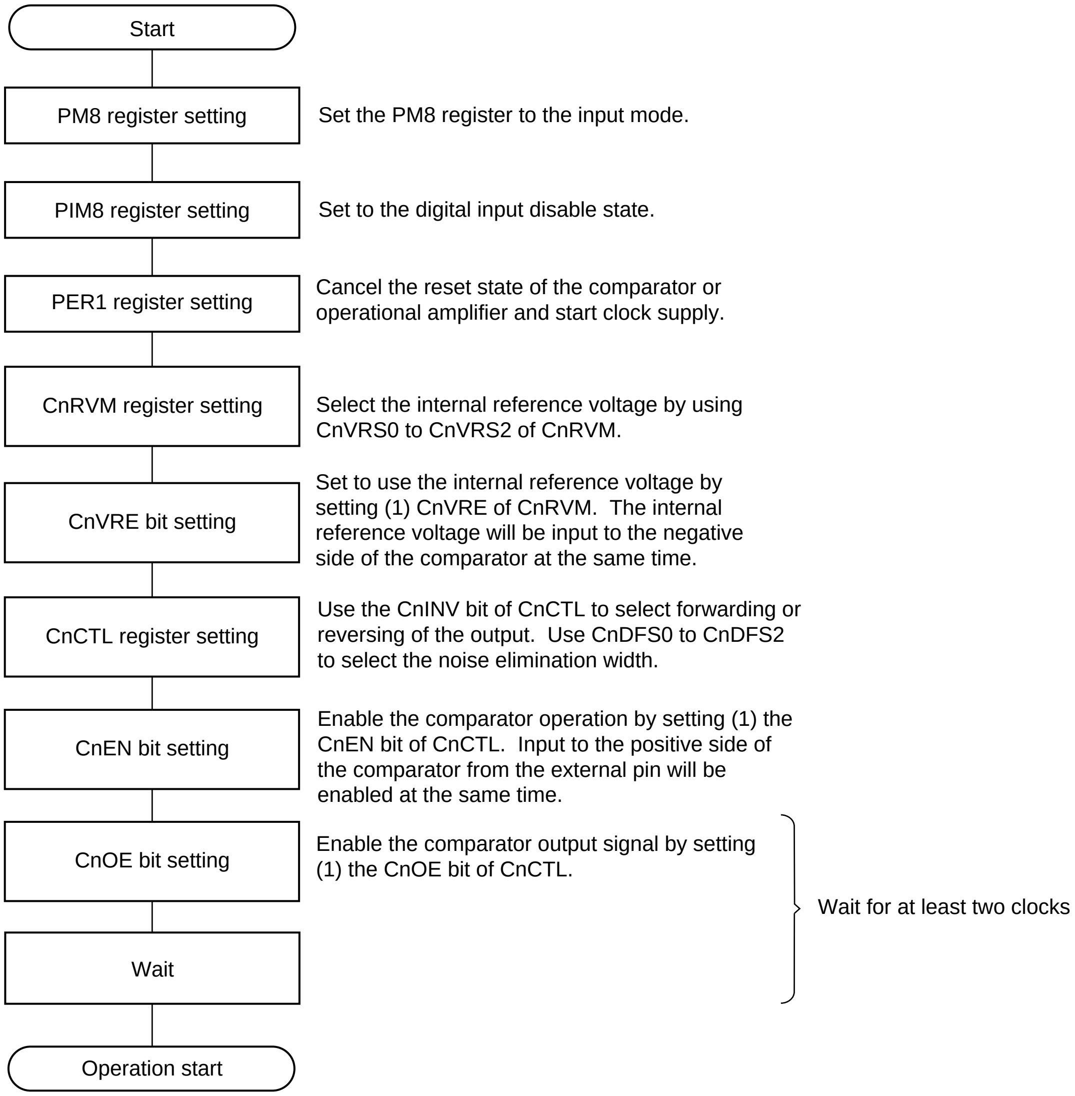

Start
PM8 register setting
Set the PM8 register to the input mode.
PIM8 register setting
Set to the digital input disable state.
PER1 register setting
Cancel the reset state of the comparator or operational amplifier and start clock supply.
CnCTL register setting
Use the CnINV bit of CnCTL to select forwarding or reversing of the output. Use CnDFS0 to CnDFS2 to select the noise elimination width.
CnEN bit setting
Enable the comparator operation by setting (1) the CnEN bit of CnCTL. Input to the positive side and negative side of the comparator from the external pin will be enabled at the same time.
CnOE bit setting
Enable the comparator output signal by setting (1) the CnOE bit of CnCTL.
Wait
Wait for at least two clocks
Operation start

Figure 8-9. Using the Internal Reference Voltage for the Comparator Reference Voltage (Using Only a Comparator)



Set the PM8 register to the input mode.

Set to the digital input disable state.

Cancel the reset state of the comparator or operational amplifier and start clock supply.

Select the internal reference voltage by using CnVRS0 to CnVRS2 of CnRVM.

Set to use the internal reference voltage by setting (1) CnVRE of CnRVM. The internal reference voltage will be input to the negative side of the comparator at the same time.

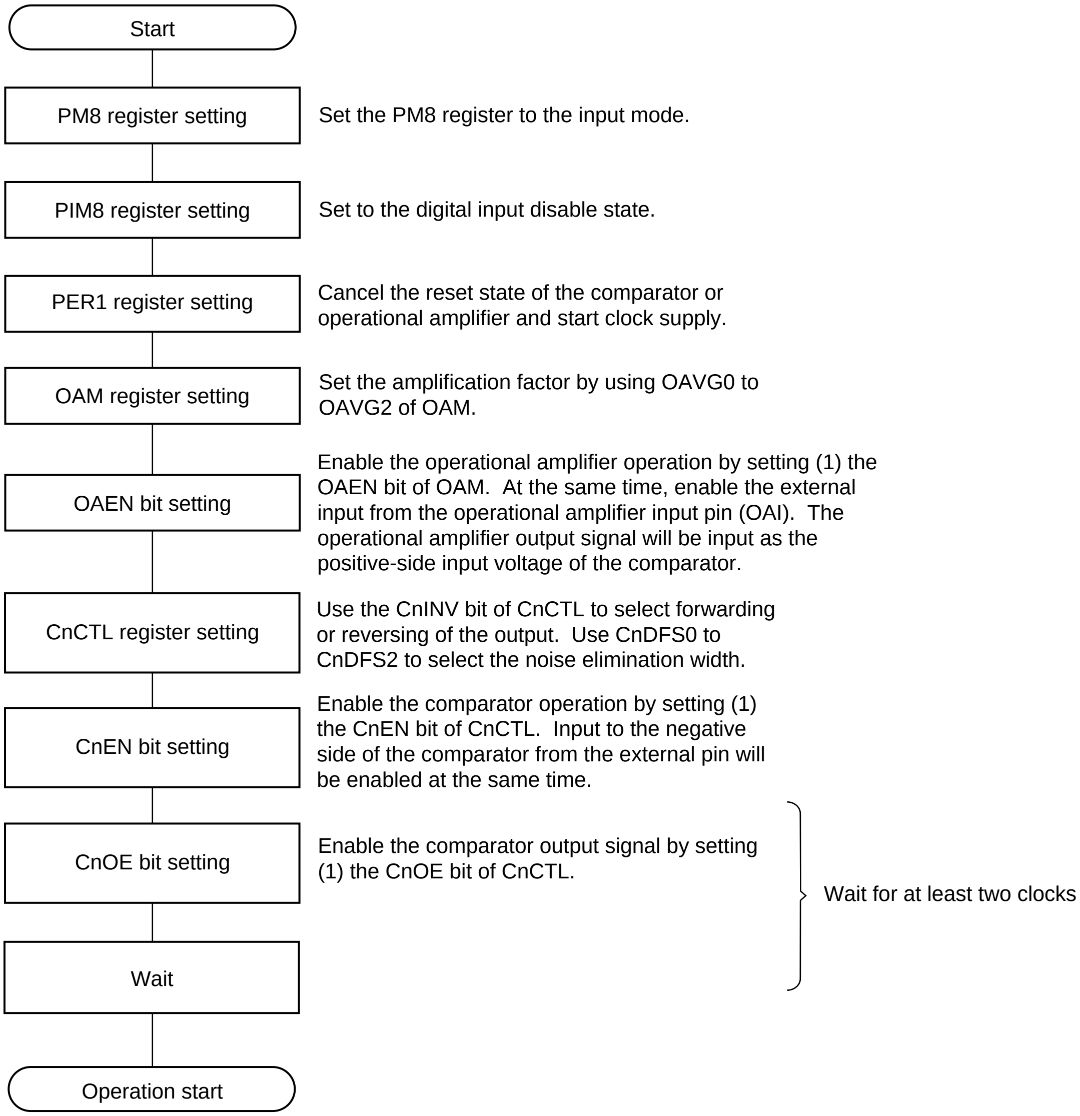
Use the CnINV bit of CnCTL to select forwarding or reversing of the output. Use CnDFS0 to CnDFS2 to select the noise elimination width.

Enable the comparator operation by setting (1) the CnEN bit of CnCTL. Input to the positive side of the comparator from the external pin will be enabled at the same time.

Enable the comparator output signal by setting (1) the CnOE bit of CnCTL.

Wait for at least two clocks

Figure 8-10. Using the External Pin Input for the Comparator Reference Voltage (Using a Comparator and an Operational Amplifier)



Set the PM8 register to the input mode.

Set to the digital input disable state.

Cancel the reset state of the comparator or operational amplifier and start clock supply.

Set the amplification factor by using OAVG0 to OAVG2 of OAM.

Enable the operational amplifier operation by setting (1) the OAEN bit of OAM. At the same time, enable the external input from the operational amplifier input pin (OAI). The operational amplifier output signal will be input as the positive-side input voltage of the comparator.

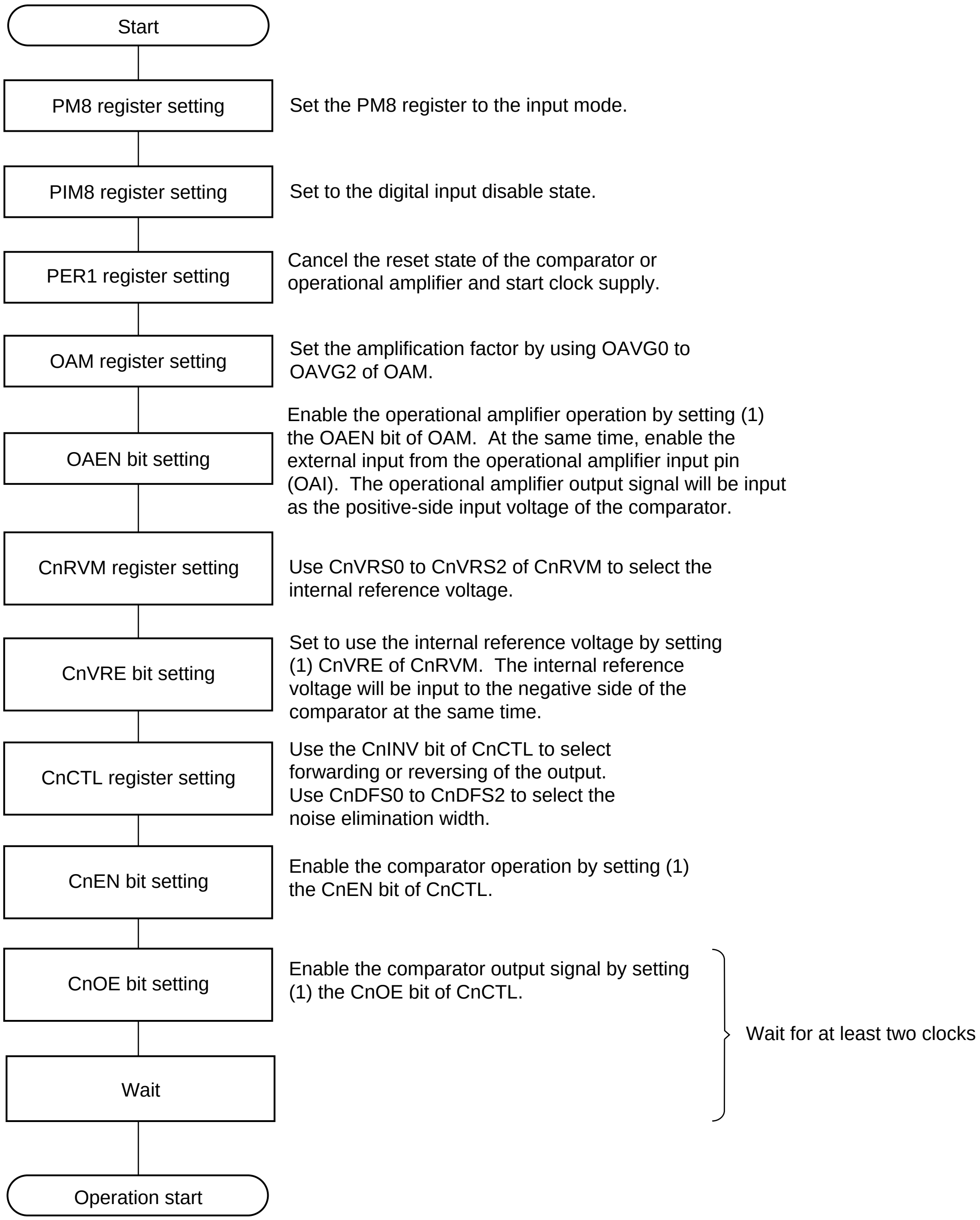
Use the CnINV bit of CnCTL to select forwarding or reversing of the output. Use CnDFS0 to CnDFS2 to select the noise elimination width.

Enable the comparator operation by setting (1) the CnEN bit of CnCTL. Input to the negative side of the comparator from the external pin will be enabled at the same time.

Enable the comparator output signal by setting (1) the CnOE bit of CnCTL.

Remark n = 0, 1

Figure 8-11. Using the Internal Reference Voltage for the Comparator Reference Voltage (Using a Comparator and an Operational Amplifier)



Set the PM8 register to the input mode.

Set to the digital input disable state.

Cancel the reset state of the comparator or operational amplifier and start clock supply.

Set the amplification factor by using OAVG0 to OAVG2 of OAM.

Enable the operational amplifier operation by setting (1) the OAEN bit of OAM. At the same time, enable the external input from the operational amplifier input pin (OAI). The operational amplifier output signal will be input as the positive-side input voltage of the comparator.

Use CnVRS0 to CnVRS2 of CnRVM to select the internal reference voltage.

Set to use the internal reference voltage by setting (1) CnVRE of CnRVM. The internal reference voltage will be input to the negative side of the comparator at the same time.

Use the CnINV bit of CnCTL to select forwarding or reversing of the output.
Use CnDFS0 to CnDFS2 to select the noise elimination width.

Enable the comparator operation by setting (1) the CnEN bit of CnCTL.

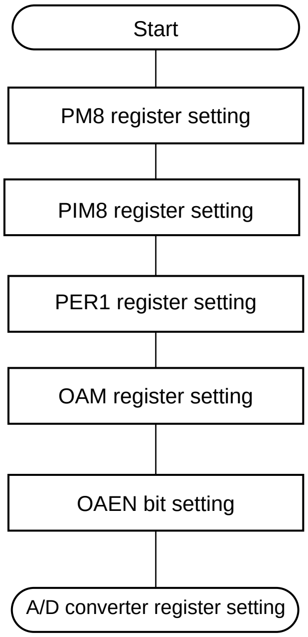
Enable the comparator output signal by setting (1) the CnOE bit of CnCTL.

Wait for at least two clocks

Remark n = 0, 1

Perform the following settings before selecting the operational amplifier output signal as the analog input by using the analog input channel specification register (ADS) of the A/D converter (refer to **12.4.1 Basic operations of A/D converter**).

Figure 8-12. Using the Operational Amplifier Output Voltage as the A/D Converter Analog Input



Set the PM8 register to the input mode.

Set to the digital input disable state.

Cancel the reset state of the comparator or operational amplifier and start clock supply.

Set the amplification factor by using OAVG0 to OAVG2 of OAM.

Enable the operational amplifier operation by setting (1) the OAEN bit of OAM. At the same time, enable the external input from the operational amplifier input pin (OAI). The operational amplifier output signal will be input as the positive-side input voltage of the comparator.

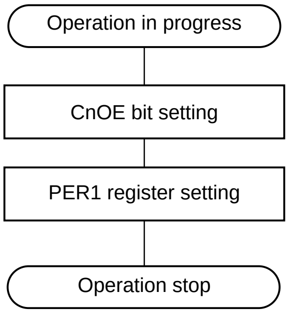
Remark n = 0, 1

8.4.2 Stopping comparator and operational amplifier operation

The procedures for stopping the operation of a comparator and an operational amplifier are described below, separately for each use method.

○ Using only a comparator (Figure 8-13)
○ Using the operational amplifier output voltage as the comparator compare voltage input (Figure 8-14)
○ Using the operational amplifier output voltage as the A/D converter analog input (Figure 8-15)

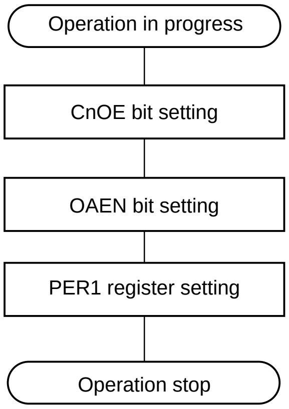
Figure 8-13. Using Only a Comparator



Clear (0) the CnOE bit of CnCTL.

Stop the clock supply to the comparator or operational amplifier and initialize the setting register.

Figure 8-14. Using the Operational Amplifier Output Voltage as the Comparator Compare Voltage Input

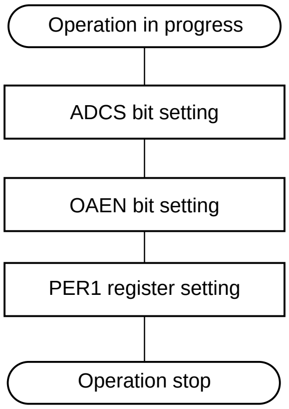


Clear (0) the CnOE bit of CnCTL.

Clear (0) the OAEN bit of OAM and prohibit the operational amplifier input.

Stop the clock supply to the comparator or operational amplifier and initialize the setting register.

Figure 8-15. Using the Operational Amplifier Output Voltage as the A/D Converter Analog Input



Clear (0) the ADCS bit of A/D converter mode register (ADM).

Clear (0) the OAEN bit of OAM and prohibit the operational amplifier input.

Stop the clock supply to the comparator or operational amplifier and initialize the setting register.

Remark n = 0, 1

CHAPTER 9 REAL-TIME COUNTER

9.1 Functions of Real-Time Counter

The real-time counter has the following features.

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

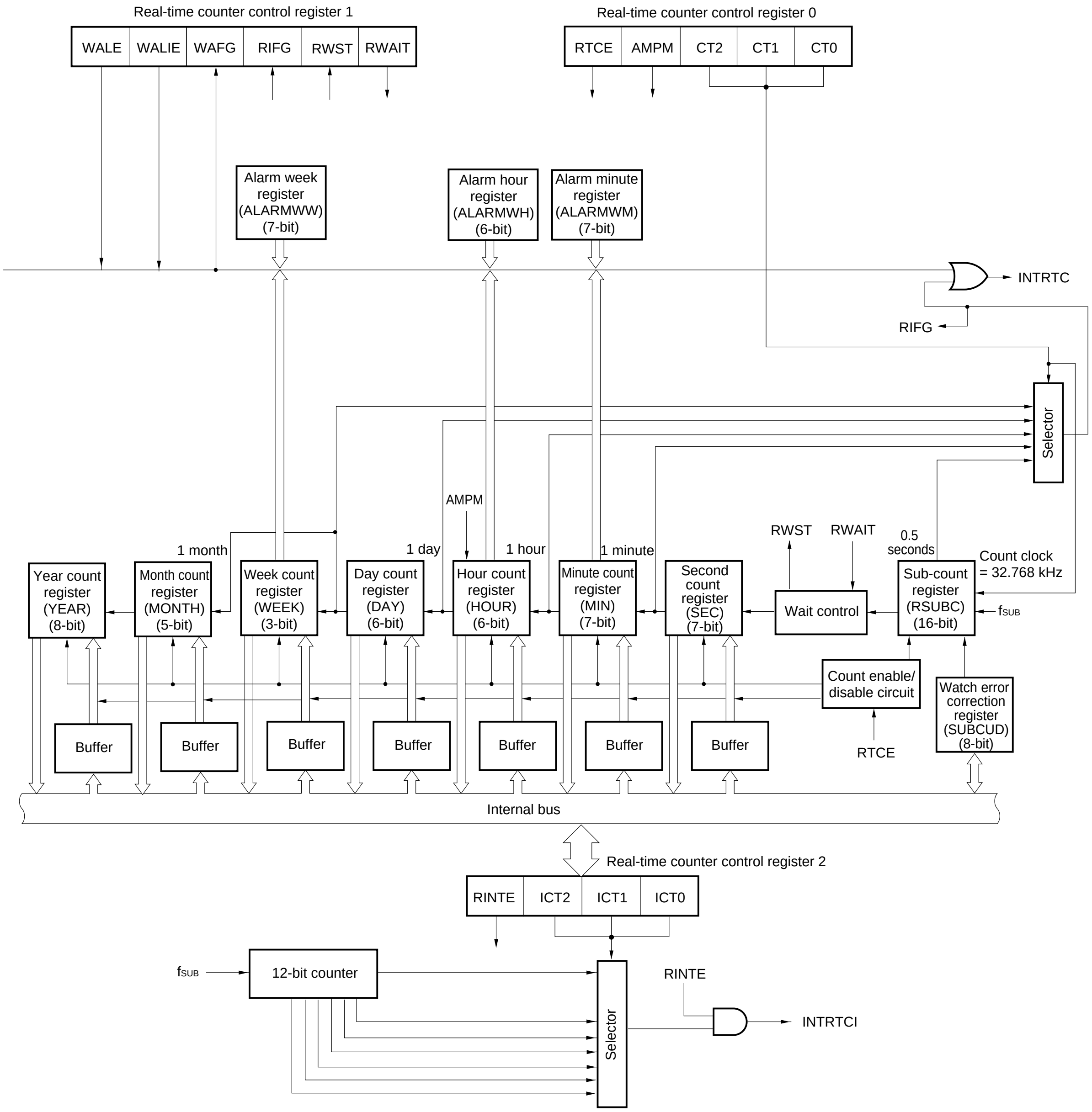
9.2 Configuration of Real-Time Counter

The real-time counter includes the following hardware.

Table 9-1. Configuration of Real-Time Counter

Item	Configuration
Control registers	Peripheral enable register 0 (PER0)
	Real-time counter control register 0 (RTCC0)
	Real-time counter control register 1 (RTCC1)
	Real-time counter control register 2 (RTCC2)
	Sub-count register (RSUBC)
	Second count register (SEC)
	Minute count register (MIN)
	Hour count register (HOUR)
	Day count register (DAY)
	Week count register (WEEK)
	Month count register (MONTH)
	Year count register (YEAR)
	Watch error correction register (SUBCUD)
	Alarm minute register (ALARMWM)
	Alarm hour register (ALARMWH)
	Alarm week register (ALARMWW)

Figure 9-1. Block Diagram of Real-Time Counter


Real-time counter control register 1
WALE
WALIE
WAFG
RIFG
RWST
RWAIT
Real-time counter control register 0
RTCE
AMPM
CT2
CT1
CT0
Alarm week register (ALARMWW) (7-bit)
Alarm hour register (ALARMWH) (6-bit)
Alarm minute register (ALARMWM) (7-bit)
INTRTC
RIFG
Selector
AMPM
RWST
RWAIT
0.5 seconds
1 month
1 day
1 hour
1 minute
Count clock = 32.768 kHz
Year count register (YEAR) (8-bit)
Month count register (MONTH) (5-bit)
Week count register (WEEK) (3-bit)
Day count register (DAY) (6-bit)
Hour count register (HOUR) (6-bit)
Minute count register (MIN) (7-bit)
Second count register (SEC) (7-bit)
Wait control
Sub-count register (RSUBC) (16-bit)
fSUB
Count enable/ disable circuit
Watch error correction register (SUBCUD) (8-bit)
RTCE
Buffer
Buffer
Buffer
Buffer
Buffer
Buffer
Buffer
Internal bus
Real-time counter control register 2
RINTE
ICT2
ICT1
ICT0
fSUB
12-bit counter
Selector
RINTE
INTRTCI

9.3 Registers Controlling Real-Time Counter

Timer real-time counter is controlled by the following 16 registers.

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the real-time counter is used, be sure to set bit 7 (RTCEN) 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 9-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

RTCEN	Control of real-time counter (RTC) input clock [Note]
0	Stops supply of input clock. • SFR used by the real-time counter (RTC) cannot be written. • The real-time counter (RTC) is in the reset status.
1	Supplies input clock. • SFR used by the real-time counter (RTC) can be read/written.

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

Cautions 1. When using the real-time counter, first set RTCEN to 1, while oscillation of the subsystem clock (f_{SUB}) is stable. If RTCEN = 0, writing to a control register of the real-time counter is ignored, and, even if the register is read, only the default value is read.

2. Be sure to clear bits 0, 1, 3, 4, and 6 of PER0 register to 0.

(2) Real-time counter control register 0 (RTCC0)

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

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

Reset signal generation clears this register to 00H.

Figure 9-3. Format of Real-Time Counter Control Register 0 (RTCC0)

Address: FFF9DH After reset: 00H R/W

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

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

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• To change the value of AMPM, set RWAIT (bit 0 of RTCC1) to 1, and re-set the hour count register (HOUR). • Table 9-2 shows the displayed time digits that are displayed.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
After changing the values of CT2 to CT0, clear the interrupt request flag.			

Remark ×: don’t care

Table 9-2. Displayed Time Digits

24-Hour System	12-Hour System	24-Hour System	12-Hour System
00	12 (AM12)	12	32 (PM12)
01	01 (AM1)	13	21 (PM1)
02	02 (AM2)	14	22 (PM2)
03	03 (AM3)	15	23 (PM3)
04	04 (AM4)	16	24 (PM4)
05	05 (AM5)	17	25 (PM5)
06	06 (AM6)	18	26 (PM6)
07	07 (AM7)	19	27 (PM7)
08	08 (AM8)	20	28 (PM8)
09	09 (AM9)	21	29 (PM9)
10	10 (AM10)	22	30 (PM10)
11	11 (AM11)	23	31 (PM11)

(3) Real-time counter control register 1 (RTCC1)

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

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

Reset signal generation clears this register to 00H.

Figure 9-4. Format of Real-Time Counter Control Register 1 (RTCC1) (1/2)

Address: FFF9EH After reset: 00H R/W

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

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
To set the registers of alarm (WALIE flag of RTCC1, ALARMWM register, ALARMWH register, and ALARMWW register), disable WALE (clear it to “0”).	

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

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one clock (32 kHz) after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

Figure 9-4. Format of Real-Time Counter Control Register 1 (RTCC1) (2/2)

<table>
<tr><th>RIFG</th><th>Constant-period interrupt status flag</th></tr>
<tr><td>0</td><td>Constant-period interrupt is not generated.</td></tr>
<tr><td>1</td><td>Constant-period interrupt is generated.</td></tr>
<tr><td colspan="2">This flag indicates the status of generation of the constant-period interrupt. When the constant-period interrupt is generated, it is set to “1”.
This flag is cleared when “0” is written to it. Writing “1” to it is invalid.</td></tr>
</table>

<table>
<tr><th>RWST</th><th>Wait status flag of real-time counter</th></tr>
<tr><td>0</td><td>Counter is operating.</td></tr>
<tr><td>1</td><td>Mode to read or write counter value</td></tr>
<tr><td colspan="2">This status flag indicates whether the setting of RWAIT is valid.
Before reading or writing the counter value, confirm that the value of this flag is 1.</td></tr>
</table>

<table>
<tr><th>RWAIT</th><th>Wait control of real-time counter</th></tr>
<tr><td>0</td><td>Sets counter operation.</td></tr>
<tr><td>1</td><td>Stops SEC to YEAR counters. Mode to read or write counter value</td></tr>
<tr><td colspan="2">This bit controls the operation of the counter.
Be sure to write “1” to it to read or write the counter value.
Because RSUBC continues operation, complete reading or writing of it in 1 second, and clear this bit back to 0.
When RWAIT = 1, it takes up to 1 clock (32 kHz) until the counter value can be read or written.
If RSUBC overflows when RWAIT = 1, it counts up after RWAIT = 0. If the second count register is written, however, it does not count up because RSUBC is cleared.</td></tr>
</table>

Caution If writing is performed to the WAFG flag with a 1-bit manipulation instruction, the RIFG flag may be cleared. Therefore, to perform writing to the WAFG flag, be sure to use an 8-bit manipulation instruction, and at this time, set 1 to the RIFG flag to invalidate writing. In the same way, to perform writing to the RIFG flag, use an 8-bit manipulation instruction and set 1 the WAFR flag.

Remark Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

(4) Real-time counter control register 2 (RTCC2)

The RTCC2 register is an 8-bit register that is used to control the interval interrupt function.

RTCC2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-5. Format of Real-Time Counter Control Register 2 (RTCC2)

Address: FFF9FH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
RTCC2	RINTE	0	0	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	Interval interrupt (INTRTCI) selection
0	×	×	×	Interval interrupt is not generated.
1	0	0	0	$2^6/f_{XT}$ (1.953125 ms)
1	0	0	1	$2^7/f_{XT}$ (3.90625 ms)
1	0	1	0	$2^8/f_{XT}$ (7.8125 ms)
1	0	1	1	$2^9/f_{XT}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{XT}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{XT}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{XT}$ (125 ms)
Change ICT2, ICT1, and ICT0 when RINTE = 0.				

(5) Sub-count register (RSUBC)

The RSUBC register is a 16-bit register that counts the reference time of 1 second of the real-time counter. It takes a value of 0000H to 7FFFH and counts 1 second with a clock of 32.768 kHz.

RSUBC can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Cautions 1. When a correction is made by using the SUBCUD register, the value may become 8000H or more.

2. This register is also cleared by reset effected by writing the second count register.

3. The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.

Figure 9-6. Format of Sub-Count Register (RSUBC)

Address: FFF90H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

Address: FFF91H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8

(6) Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the sub-counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

Figure 9-7. Format of Second Count Register (SEC)

Address: FFF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

(7) Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

Figure 9-8. Format of Minute Count Register (MIN)

Address: FFF93H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

(8) Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 0 to 23 or 1 to 12 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Figure 9-9. Format of Hour Count Register (HOUR)

Address: FFF94H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution Bit 5 (HOUR20) of HOUR indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

(9) Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It counts up when the hour counter overflows.

This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 01 to 31 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 01H.

Figure 9-10. Format of Day Count Register (DAY)

Address: FFF96H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

(10) Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 06 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

Figure 9-11. Format of Week Count Register (WEEK)

Address: FFF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

(11) Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 01 to 12 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 01H.

Figure 9-12. Format of Month Count Register (MONTH)

Address: FFF97H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

(12) Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month counter overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 99 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

Figure 9-13. Format of Year Count Register (YEAR)

Address: FFF98H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

(13) Watch error correction register (SUBCUD)

This register is used to correct the count value of the sub-count register (RSUBC).

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

Reset signal generation clears this register to 00H.

Figure 9-14. Format of Watch Error Correction Register (SUBCUD)

Address: FFF99H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40.
1	Corrects watch error only when the second digits are at 00.

F6	Setting of watch error correction method
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.
When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100).	

(14) Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

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

Reset signal generation clears this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 9-15. Format of Alarm Minute Register (ALARMWM)

Address: FFF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

(15) Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 9-16. Format of Alarm Hour Register (ALARMWH)

Address: FFF9BH After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of ALARMWH indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

(16) Alarm week register (ALARMWW)

This register is used to set date of alarm.

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

Reset signal generation clears this register to 00H.

Figure 9-17. Format of Alarm Week Register (ALARMWW)

Address: FFF9CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

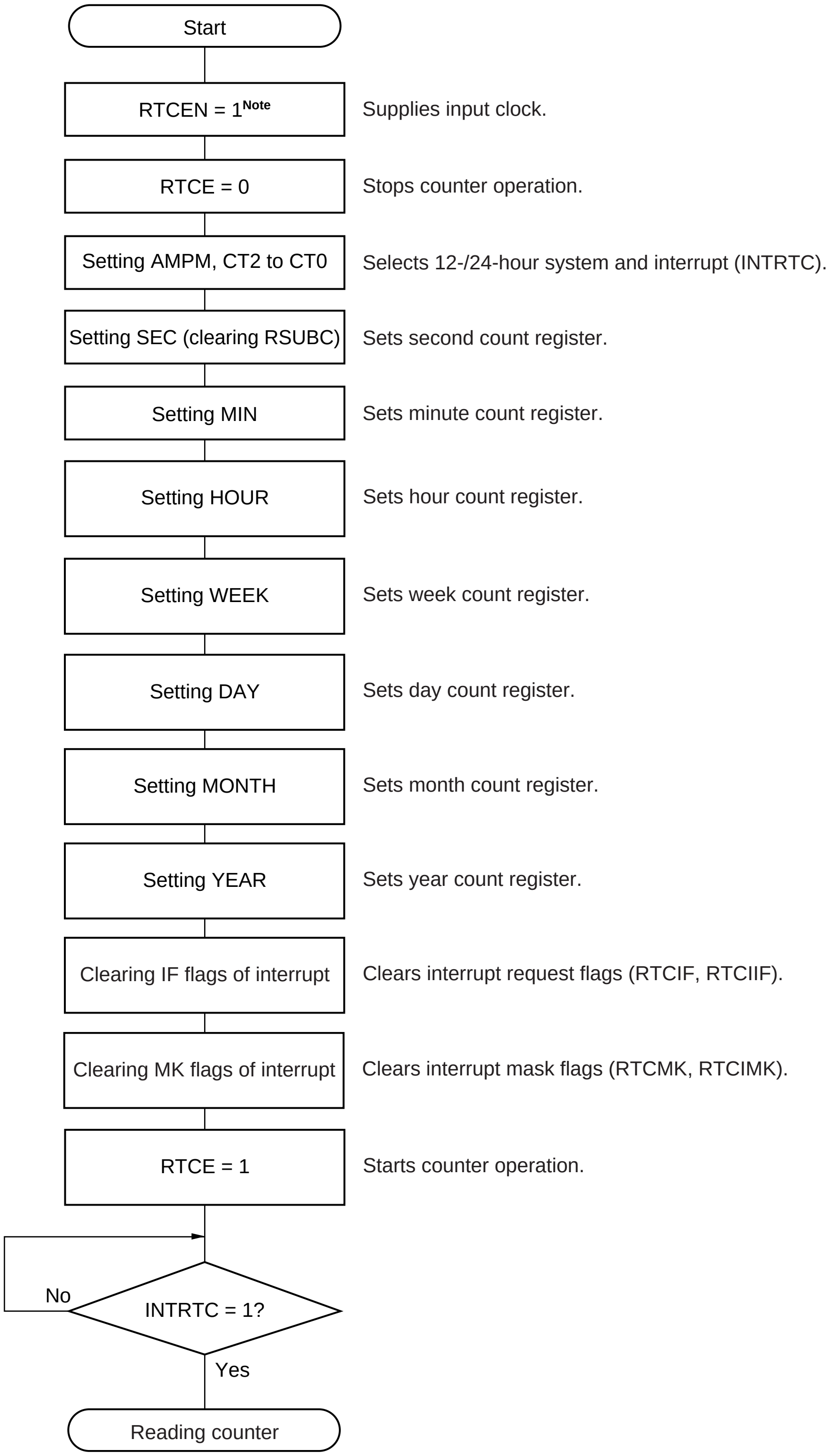
Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

9.4 Real-Time Counter Operation

9.4.1 Starting operation of real-time counter

Figure 9-18. Procedure for Starting Operation of Real-Time Counter

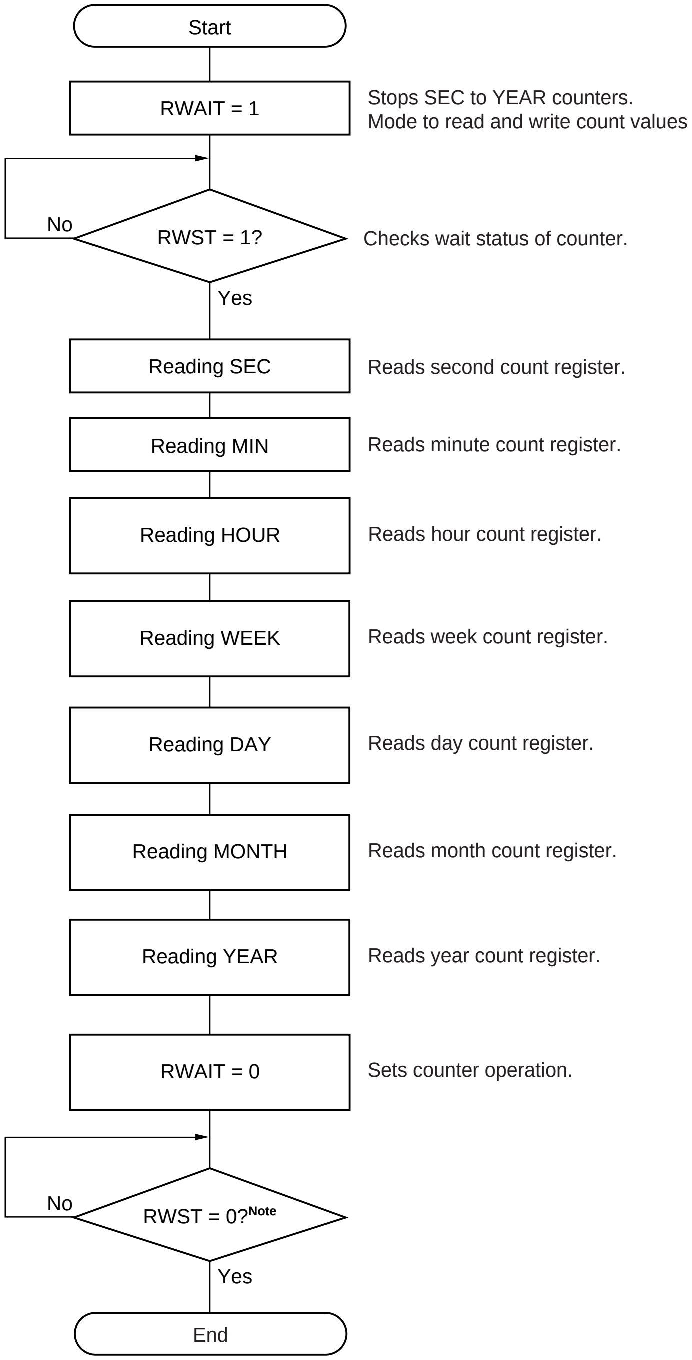

Start
RTCEN = 1Note
Supplies input clock.
RTCE = 0
Stops counter operation.
Setting AMPM, CT2 to CT0
Selects 12-/24-hour system and interrupt (INTRTC).
Setting SEC (clearing RSUBC)
Sets second count register.
Setting MIN
Sets minute count register.
Setting HOUR
Sets hour count register.
Setting WEEK
Sets week count register.
Setting DAY
Sets day count register.
Setting MONTH
Sets month count register.
Setting YEAR
Sets year count register.
Clearing IF flags of interrupt
Clears interrupt request flags (RTCIF, RTCIIF).
Clearing MK flags of interrupt
Clears interrupt mask flags (RTCMK, RTCIMK).
RTCE = 1
Starts counter operation.
No
INTRTC = 1?
Yes
Reading counter

Note First set RTCEN to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

9.4.2 Reading/writing real-time counter

Read or write the counter when RWAIT = 1.

Figure 9-19. Procedure for Reading Real-Time Counter

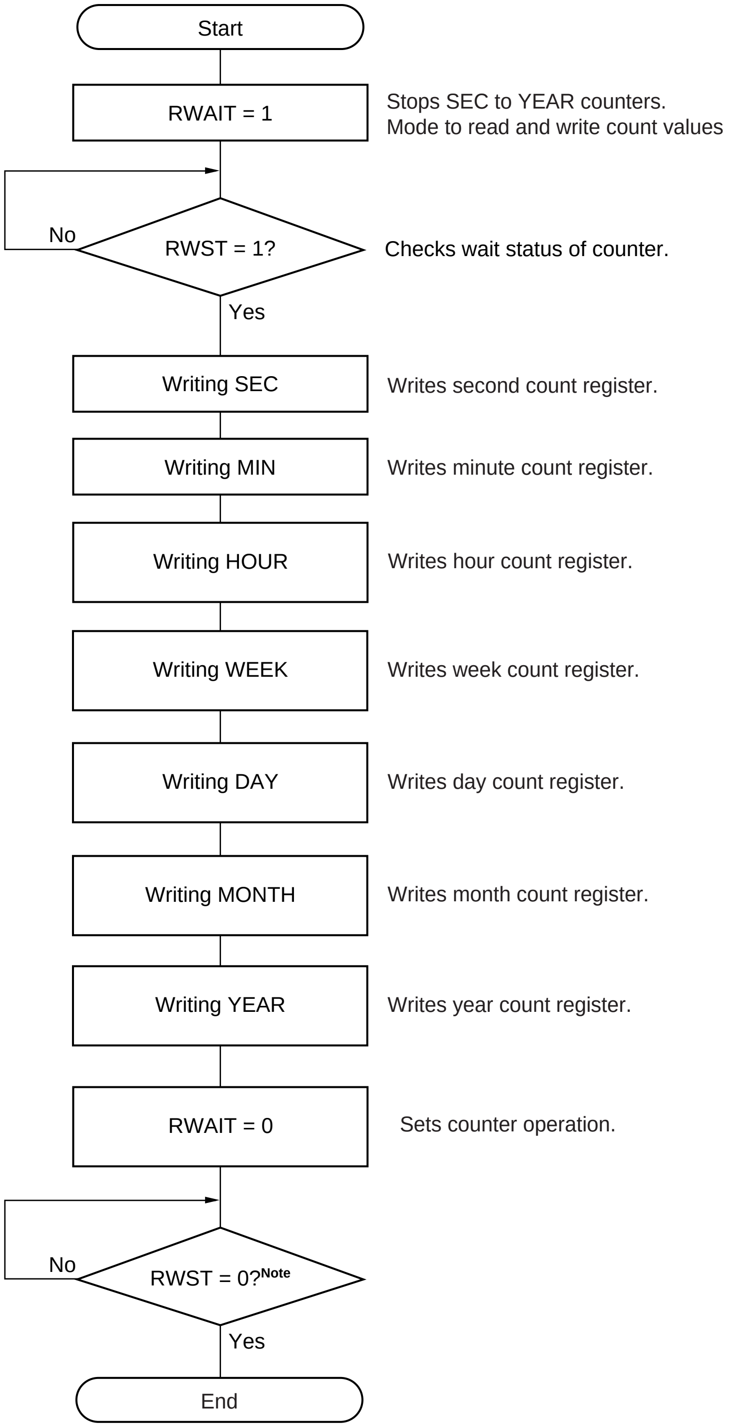

Start
RWAIT = 1
Stops SEC to YEAR counters.
Mode to read and write count values
No
RWST = 1?
Checks wait status of counter.
Yes
Reading SEC
Reads second count register.
Reading MIN
Reads minute count register.
Reading HOUR
Reads hour count register.
Reading WEEK
Reads week count register.
Reading DAY
Reads day count register.
Reading MONTH
Reads month count register.
Reading YEAR
Reads year count register.
RWAIT = 0
Sets counter operation.
No
RWST = 0?Note
Yes
End

Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence.
All the registers do not have to be set and only some registers may be read.

Figure 9-20. Procedure for Writing Real-Time Counter


Start
RWAIT = 1
Stops SEC to YEAR counters.
Mode to read and write count values
No
RWST = 1?
Checks wait status of counter.
Yes
Writing SEC
Writes second count register.
Writing MIN
Writes minute count register.
Writing HOUR
Writes hour count register.
Writing WEEK
Writes week count register.
Writing DAY
Writes day count register.
Writing MONTH
Writes month count register.
Writing YEAR
Writes year count register.
RWAIT = 0
Sets counter operation.
No
RWST = 0?Note
Yes
End

Note Be sure to confirm that RWST = 0 before setting STOP mode.

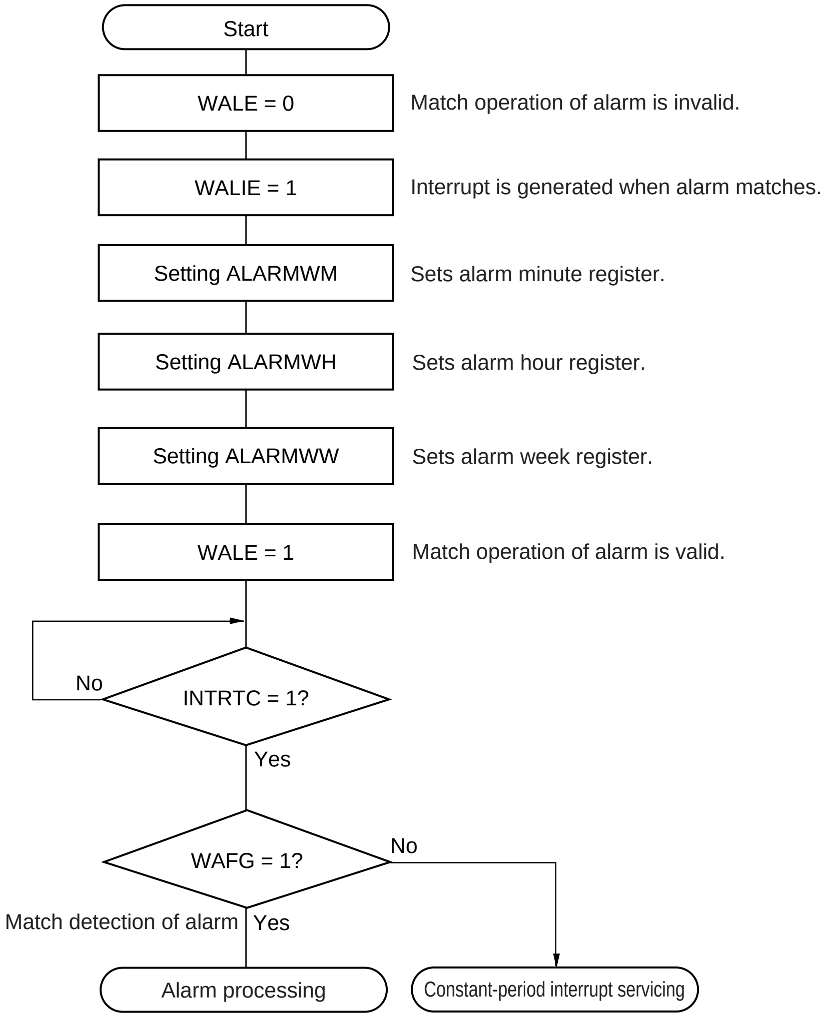
Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be written in any sequence.
All the registers do not have to be set and only some registers may be written.

9.4.3 Setting alarm of real-time counter

Set time of alarm when WALE = 0.

Figure 9-21. Alarm Setting Procedure


Start
WALE = 0
Match operation of alarm is invalid.
WALIE = 1
Interrupt is generated when alarm matches.
Setting ALARMWM
Sets alarm minute register.
Setting ALARMWH
Sets alarm hour register.
Setting ALARMWW
Sets alarm week register.
WALE = 1
Match operation of alarm is valid.
No
INTRTC = 1?
Yes
WAFG = 1?
No
Match detection of alarm
Yes
Alarm processing
Constant-period interrupt servicing

Remarks 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

CHAPTER 10 WATCHDOG TIMER

10.1 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.

10.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 10-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

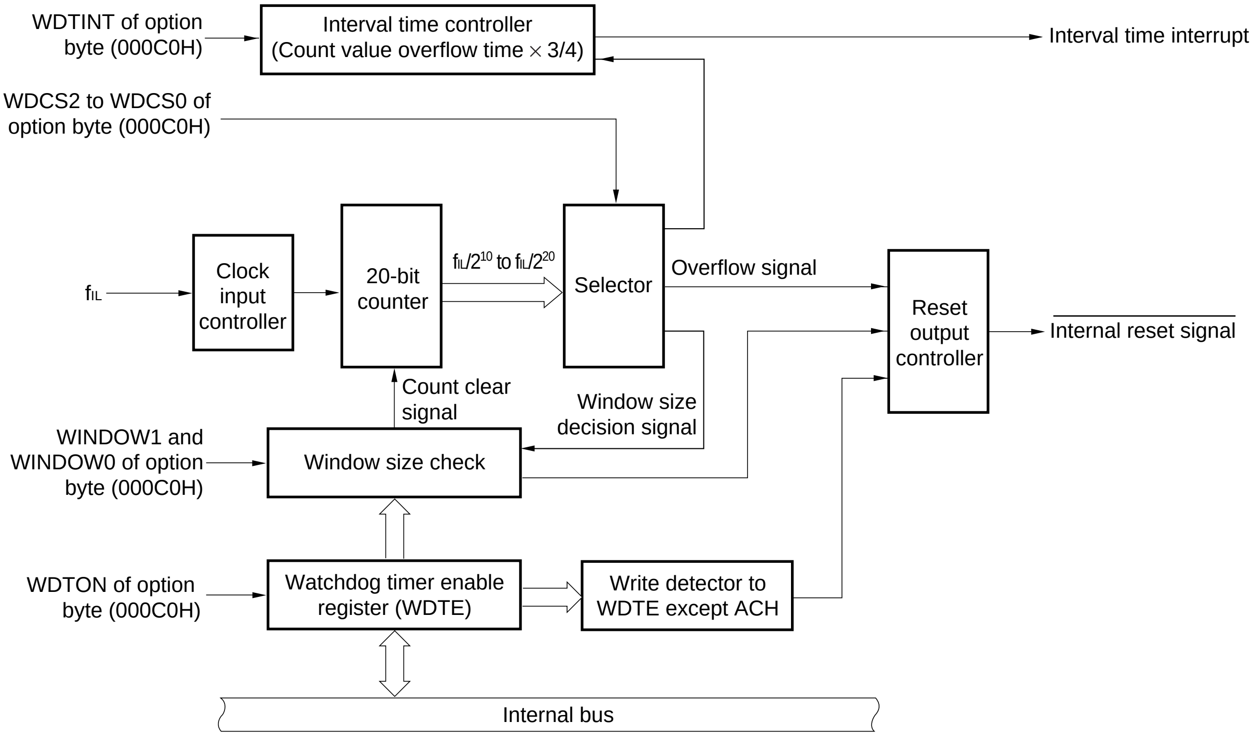
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 10-2. 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 10-1. Block Diagram of Watchdog Timer


WDTINT of option byte (000C0H)
Interval time controller (Count value overflow time × 3/4)
Interval time interrupt
WDCS2 to WDCS0 of option byte (000C0H)
fIL
Clock input controller
20-bit counter
fIL/2^10 to fIL/2^20
Selector
Overflow signal
Reset output controller
Internal reset signal
Count clear signal
Window size decision signal
WINDOW1 and WINDOW0 of option byte (000C0H)
Window size check
WDTON of option byte (000C0H)
Watchdog timer enable register (WDTE)
Write detector to WDTE except ACH
Internal bus

10.3 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 10-2. 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)).

10.4 Operation of Watchdog Timer

10.4.1 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 **10.4.2** and **CHAPTER 21**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **10.4.3** 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.
   A 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.

<Example> When the overflow time is set to $2^{10}/f_{IL}$, writing “ACH” is valid up to count value 3FH.

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.

10.4.2 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 time is set.

Table 10-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 33 kHz (MAX.))
0	0	0	$2^{10}/f_{IL}$ (31.03 ms)
0	0	1	$2^{11}/f_{IL}$ (62.06 ms)
0	1	0	$2^{12}/f_{IL}$ (124.1 ms)
0	1	1	$2^{13}/f_{IL}$ (248.2 ms)
1	0	0	$2^{15}/f_{IL}$ (992.9 ms)
1	0	1	$2^{17}/f_{IL}$ (3.971 s)
1	1	0	$2^{18}/f_{IL}$ (7.943 s)
1	1	1	$2^{20}/f_{IL}$ (31.17 s)

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.

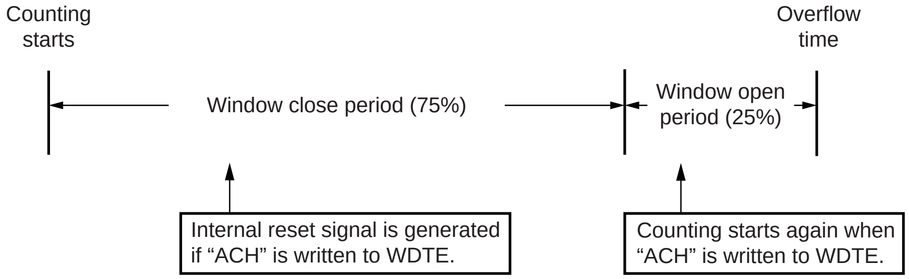
Remark f_{IL}: Internal low-speed oscillation clock frequency

10.4.3 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 25%


Counting starts
Overflow time
Window close period (75%)
Window open period (25%)
Internal reset signal is generated if "ACH" is written to WDTE.
Counting starts again when "ACH" is written to WDTE.

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 to be set is as follows.

Table 10-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	25%
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.

3. Do not set the window open period to 25% if the watchdog timer corresponds to either of the conditions below.

- **When stopping all main system clocks (internal high-speed oscillation clock, X1 clock, and external main system clock) by use of the STOP mode or software.**

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			
	25%	50%	75%	100%
Window close time	0 to 28.44 ms	0 to 18.96 ms	0 to 9.48 ms	None
Window open time	28.44 to 31.03 ms	18.96 to 31.03 ms	9.48 to 31.03 ms	0 to 31.03 ms

<When window open period is 25%>

- Overflow time:
  $2^{10}/f_{IL}$ (MAX.) = 2^{10}/33 kHz (MAX.) = 31.03 ms
- Window close time:
  0 to $2^{10}/f_{IL}$ (MIN.) × (1 − 0.25) = 0 to 2^{10}/27 kHz (MIN.) × 0.75 = 0 to 28.44 ms
- Window open time:
  $2^{10}/f_{IL}$ (MIN.) × (1 − 0.25) to $2^{10}/f_{IL}$ (MAX.) = 2^{10}/27 kHz (MIN.) × 0.75 to 2^{10}/33 kHz (MAX.) = 28.44 to 31.03 ms

10.4.4 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 10-5. 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 11 A/D CONVERTER

11.1 Function of A/D Converter

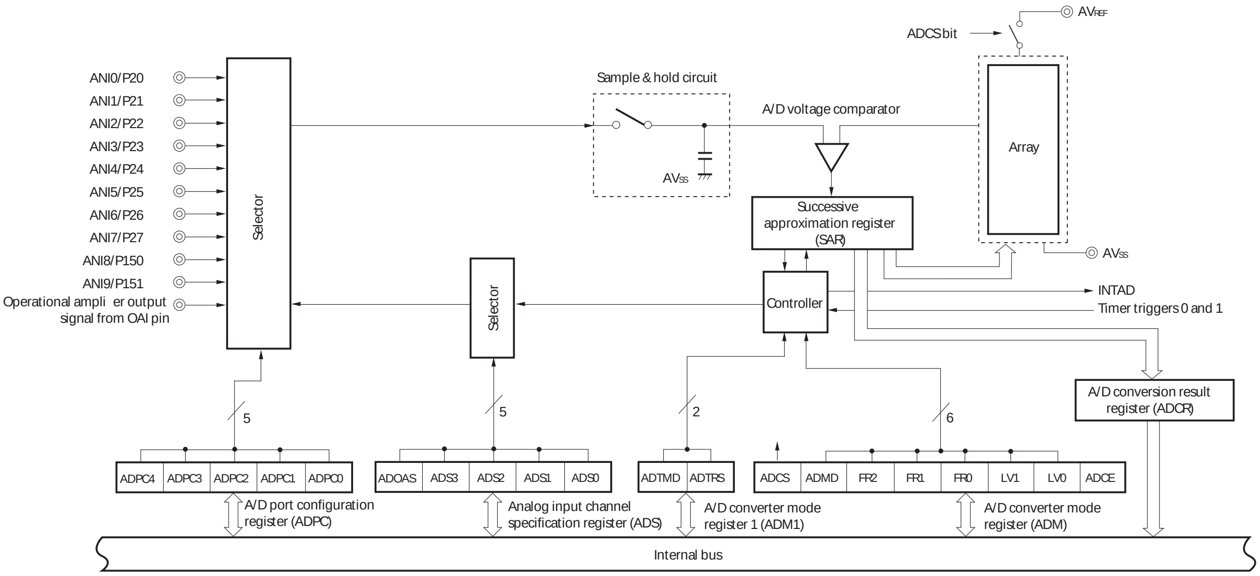
The A/D converter is a 10-bit resolution converter that converts analog input signals into digital values, and is configured to control a total of twelve channels of analog inputs, including up to eleven channels of A/D converter analog inputs (ANI0 to ANI9) and an internal operational amplifier output (OAI).

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 ANI0 to ANI19. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 11-1. Block Diagram of A/D Converter



11.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI9 pins

These are the analog input pins of the A/D converter. μPD79F9211 have 10 channels (ANI0 to ANI9) of analog input pins. 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) OAI pin

This is the output signal pin of the operational amplifier. The A/D converter can perform A/D conversion by selecting the output signal of the operational amplifier as the analog input.

(3) 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.

(4) 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.

(5) Array

The array generates the comparison voltage input from an analog input pin.

(6) 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.

(7) 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).

(8) 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.

(9) 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.

(10) AV_{REF} pin

This pin inputs the reference voltage of the A/D converter, the operational amplifier, the power supply pins and A/D converter of the comparator, and the comparator. When all pins of ports 2, 8, and 15 are used as the analog port pins, make the potential of AV_{REF} be such that $2.7\ V \leq AV_{REF} \leq V_{DD}$. When one or more of the pins of ports 2, 8, and 15 are used as the digital port pins, make AV_{REF} the same potential as V_{DD}.

The analog signal input to ANI0 to ANI9 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

(11) 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.

11.3 Registers Used in A/D Converter

The A/D converter uses the following eight registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register (ADM)
- A/D converter mode register 1 (ADM1)
- A/D port configuration register (ADPC)
- Analog input channel specification register (ADS)
- Port mode registers 2, 8, 15 (PM2, PM8, PM15)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. 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 11-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

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	Supplies input clock. • 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.

2. Be sure to clear bits 0, 1, 3, 4, and 6 of PER0 register to 0.

(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 11-3. 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	ADMD	FR2[Note 1]	FR1[Note 1]	FR0[Note 1]	LV1[Note 1]	LV0[Note 1]	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

ADMD	A/D conversion operation mode specification
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control[Note 2]
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation (A/D voltage comparator: 1/2AV$_{\text{REF}}$ operation)

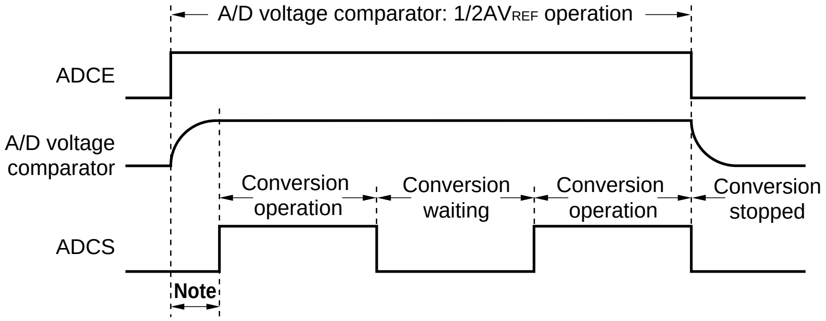
Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 11-2 A/D Conversion Time Selection**.

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, when ADCS is set to 1 after 1 µs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Table 11-1. 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 (A/D voltage comparator: 1/2AV$_{\text{REF}}$ operation, only comparator consumes power)
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator: 1/2AV$_{\text{REF}}$ operation)

Figure 11-4. 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 falling 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 11-2. A/D Conversion Time Selection

(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	Standard	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	Voltage boost	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

(2) 2.7 V ≤ AV_{REF} ≤ 5.5 V

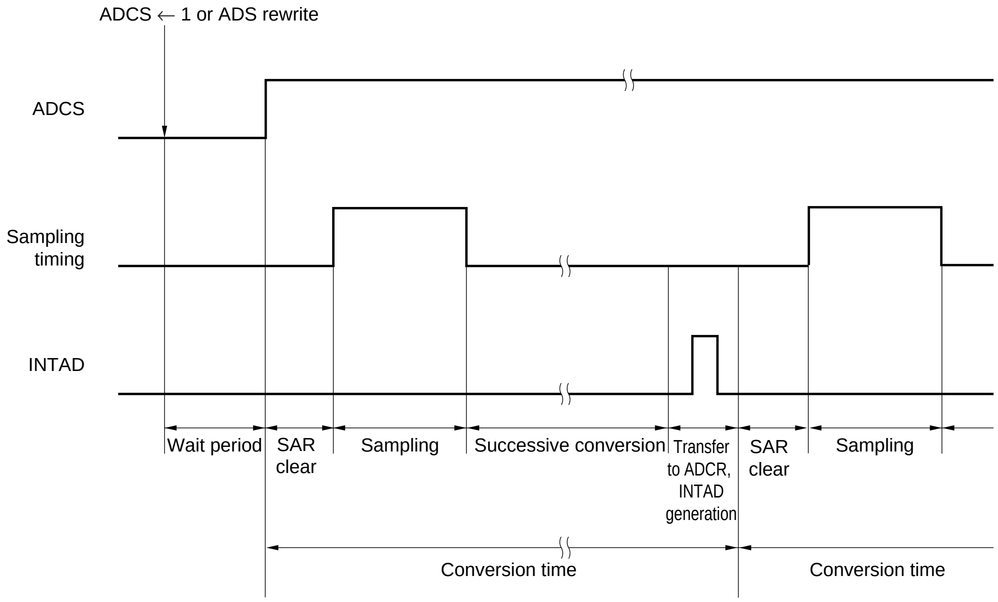
A/D Converter Mode Register (ADM)					Mode	Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	
0	0	0	0	0	Standard	Setting prohibited	Setting prohibited	34.2 μs	17.1 μs	f_{CLK}/20
0	0	1					34.4 μs	17.2 μs	8.6 μs	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	Voltage boost	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

Figure 11-5. A/D Converter Sampling and A/D Conversion Timing



(3) A/D converter mode register 1 (ADM1)

This register sets the A/D conversion start trigger.

ADM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-6. Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF42H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM1	ADTMD	0	0	0	0	0	0	ADTRS

ADTMD	A/D trigger mode selection
0	Software trigger mode
1	Timer trigger mode (hardware trigger mode)

ADTRS	Timer trigger signal selection
0	Timer trigger signal 0
1	Timer trigger signal 1

Caution Rewriting ADM1 during A/D conversion is prohibited. Rewrite it when conversion operation is stopped (ADCS = 0).

Remark For details of the timer trigger signals, refer to **7.4.8 Operation as A/D conversion trigger output function (type 1)** and **7.4.9 Operation as A/D conversion trigger output function (type 2)**.

(4) 12-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. 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 11-7. 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.

(5) 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 11-8. 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.

(6) 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 11-9. 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	ADOAS	0	0	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADOAS	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	ANI0	P20/ANI0 pin
0	0	0	0	1	ANI1	P21/ANI1 pin
0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	1	0	1	ANI5	P25/ANI5 pin
0	0	1	1	0	ANI6	P26/ANI6 pin
0	0	1	1	1	ANI7	P27/ANI7 pin
0	1	0	0	0	ANI8	P150/ANI8 pin
0	1	0	0	1	ANI9	P151/ANI9 pin
1	×	×	×	×	OAI	P80/OAI pin
Other than the above					Setting prohibited	

(Notes 1, 2 and Caution are listed on the next page.)

○ Scan mode (ADMD = 1)

ADOAS	ADS3	ADS2	ADS1	ADS0	Analog input channel			
					Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	0	1	1	0	ANI6	ANI7	ANI8	ANI9
1	0	0	0	0	OAI	ANI0	ANI1	ANI2
1	0	0	0	1	OAI	ANI1	ANI2	ANI3
1	0	0	1	0	OAI	ANI2	ANI3	ANI4
1	0	0	1	1	OAI	ANI3	ANI4	ANI5
1	0	1	0	0	OAI	ANI4	ANI5	ANI6
1	0	1	0	1	OAI	ANI5	ANI6	ANI7
1	0	1	1	0	OAI	ANI6	ANI7	ANI8
1	0	1	1	1	OAI	ANI7	ANI8	ANI9
Other than the above					ANI0	ANI1	ANI2	ANI3

Cautions 1. Be sure to clear bits 4, 5, and 7 to “0”.

2 Set a channel to be used for A/D conversion in the input mode by using port mode registers 2, 8, and 15 (PM2, PM8, PM15).

3. Do not set the pin that is set by ADPC as digital I/O by ADS.

4. Select the output signal (OAI pin) of the operational amplifier as the analog input after setting the operation of the operational amplifier (refer to 11.4.1 Basic operations of A/D converter).

Remark ×: don’t care

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

This register switches the ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI9/P151 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 11-10. 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 5		Port 2							
					ANI9 /P151	ANI8 /P150	ANI7 /P27	ANI6 /P26	ANI5 /P25	ANI4 /P24	ANI3 /P23	ANI2 /P22	ANI1 /P21	ANI0 /P20
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	A	A	D	D	D	D	D	D
0	0	1	1	1	A	A	A	D	D	D	D	D	D	D
0	1	0	0	0	A	A	D	D	D	D	D	D	D	D
0	1	0	0	1	A	D	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D
Other than the 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.

Note 1 →

(8) Port input mode register 8 (PIM8)

This register enables or disables the input of port 8 in 1-bit units.

Disable the input to use the OAI pin as the analog input. Enable the input to use the port function, or the external interrupt and timer Hi-Z control functions, because the input is disabled in the initial state.

PIM8 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-11. Format of Port Input Mode Register 8 (PIM8)

Address: F0048H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM8	0	0	0	0	PIM83	PIM82	PIM81	PIM80

PIM8n	Selection of enabling or disabling P8n pin input (n = 0 to 3)
0	Disables input
1	Enables input

(9) Port mode registers 2, 8, and 15 (PM2, PM8, PM15)

When using the ANI0/P20 to ANI7/P27, ANI8/P150 to ANI9/P151, and OAI/P80 [Note] pins for analog input port, set PM20 to PM27, PM80, and PM150 to PM151 to 1. The output latches of P20 to P27, P80, and P150 to P151 at this time may be 0 or 1.

If PM20 to PM27, PM80, and PM150 to PM151 are set to 0, they cannot be used as analog input port pins.

PM2, PM8, 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 11-12. Formats of Port Mode Registers 2, 8, and 15 (PM2, PM8, PM15)

Address: FFF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

Address: FFF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	1	1	PM83	PM82	PM81	PM80

Address: FFF2FH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM15	1	1	1	1	1	1	PM151	PM150

PMmn	Pmn pin I/O mode selection (mn = 20 to 27, 80, 150 to 151)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Cautions .Be sure to set bit 2 of PM15 to “1”, and bit 2 of P15 to “0”.

The ANI0/P20 to ANI7/P27, OAI/P80, and ANI8/P150 to ANI9/P151 pins are as shown below depending on the settings of ADPC, ADS, PM2, PM8, and PM15.

Table 11-3. Setting Functions of ANI0/P20 to ANI7/P27, OAI/P80, and ANI8/P150 to ANI9/P151 Pins

ADPC	PM2, PM8, and PM15	ADS	ANI0/P20 to ANI7/P27, OAI/P80, and ANI8/P150 to ANI9/P151Pins
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.	

11.4 A/D Converter Operations

11.4.1 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, PM8, and PM15).

<5> Set the operational amplifier operation to set the operational amplifier output (OAI pin) for the analog input channel (refer to **8.4.1 Starting comparator and operational amplifier operation**).

<6> Select one channel for A/D conversion using the analog input channel specification register (ADS).

<7> Use the A/D converter mode register 1 (ADM1) to set the trigger mode.

<8> 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>.
(<9> to <15> are operations performed by hardware.)

<9> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<10> 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.

<11>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.

<12> 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.

<13> 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

<14> Comparison is continued in this way up to bit 0 of SAR.

<15> 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.

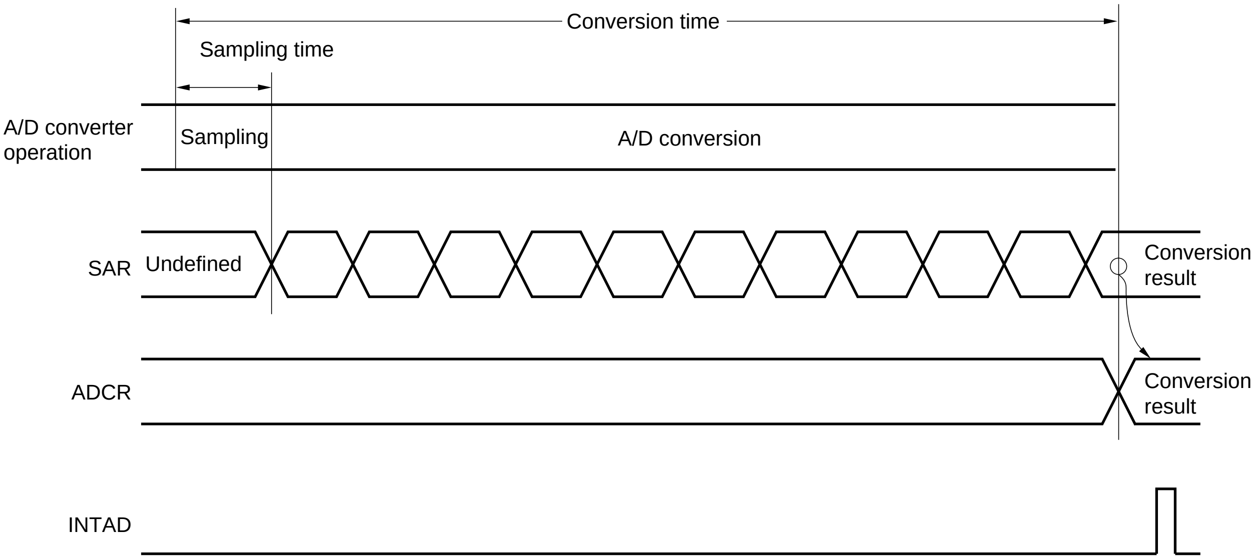
<16> Repeat steps <9> to <15>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <8>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <8>. To change a channel of A/D conversion, start from <6>.

Caution Make sure the period of <3> to <8> 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 11-13. 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.

11.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI9, OAI) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$\mathrm{SAR} = \mathrm{INT}\left(\frac{V_{\mathrm{AIN}}}{AV_{\mathrm{REF}}} \times 1024 + 0.5\right)$$

$$\mathrm{ADCR} = \mathrm{SAR} \times 64$$

or

$$\left(\frac{\mathrm{ADCR}}{64} - 0.5\right) \times \frac{AV_{\mathrm{REF}}}{1024} \leq V_{\mathrm{AIN}} < \left(\frac{\mathrm{ADCR}}{64} + 0.5\right) \times \frac{AV_{\mathrm{REF}}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

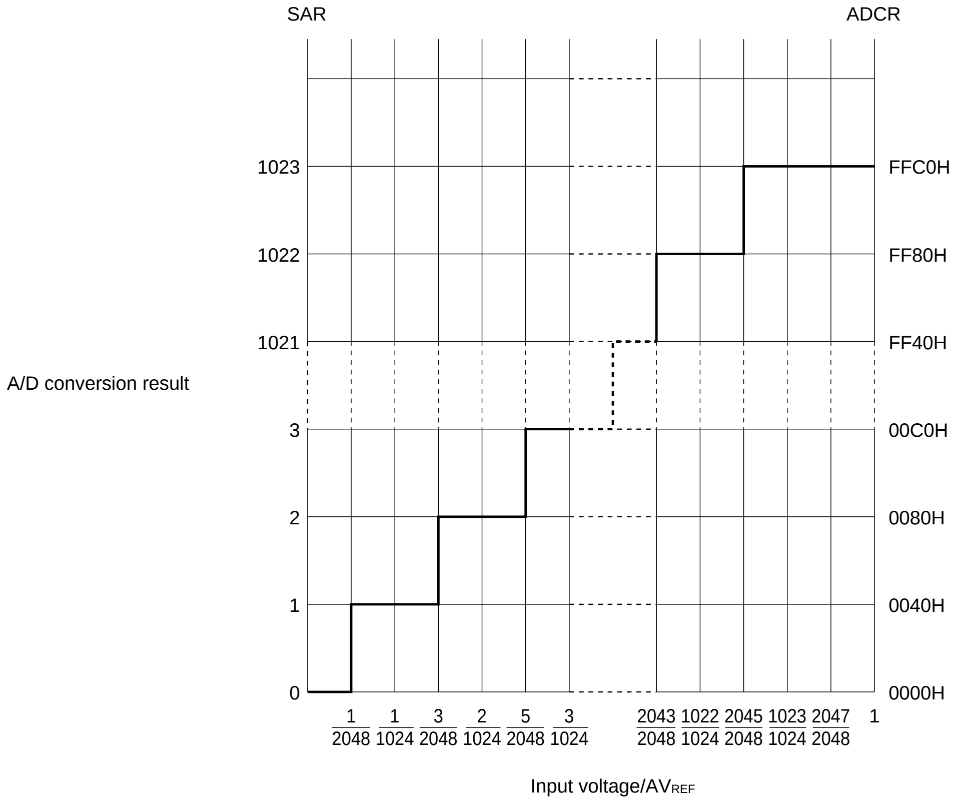
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 11-14 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 11-14. Relationship Between Analog Input Voltage and A/D Conversion Result



11.4.3 Trigger mode selection

The following two trigger modes that set the A/D conversion start timing are provided. These trigger modes are set by the ADM1 register.

- Software trigger mode
- Timer trigger mode (hardware trigger mode)

(1) Software trigger mode

This mode is used to start A/D conversion of the analog input channels (ANI0 to ANI9, OAI), which have been selected by the analog input channel specification register (ADS), by setting ADCS to 1.

A/D conversion is repeatedly performed as long as the ADCS bit is not cleared to 0, after completion of A/D conversion.

If the ADM, ADM1, or ADS register is written during conversion, A/D conversion is aborted. In this case, A/D conversion is started again from the beginning in the select mode, and A/D conversion is started again from scan 0 in the scan mode.

(2) Timer trigger mode (hardware trigger mode)

This mode is used to start A/D conversion of the analog input channels (ANI0 to ANI9, operational amplifier output), which have been selected by the analog input channel specification register (ADS), by setting ADCS to 1 and detecting timer trigger signals 0 and 1.

A/D conversion is repeatedly performed as long as the ADCS bit is not cleared to 0, after completion of A/D conversion.

If a timer trigger signal is generated during A/D conversion or if the ADM, ADM1, or ADS register is written during conversion, A/D conversion is aborted. In this case, A/D conversion is started again from the beginning in the select mode, and A/D conversion is started again from scan 0 in the scan mode.

11.4.4 A/D converter operation modes

The select mode and scan mode are provided as the A/D converter operation modes.

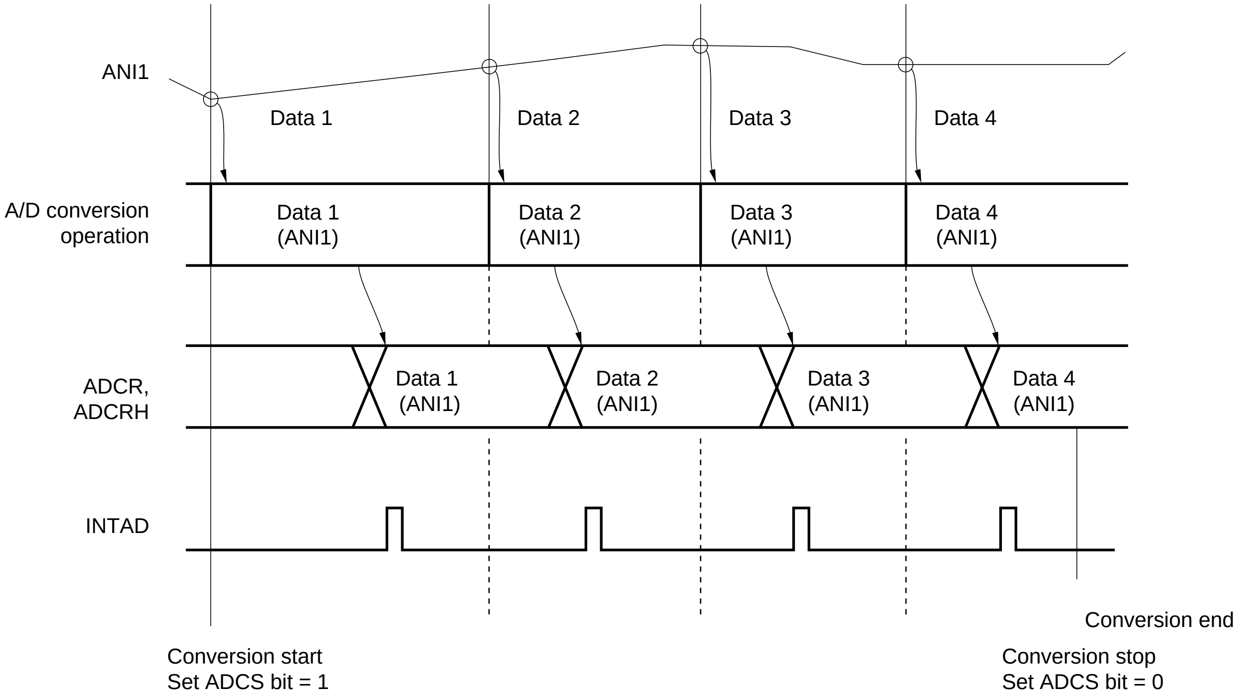
(1) Select mode

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.

If anything is written to ADM, ADM1, or ADS during conversion, A/D conversion is aborted. In this case, A/D conversion is started again from the beginning, regardless of being in the software trigger mode or timer trigger mode (hardware trigger mode).

Figure 11-15. Example of Select Mode Operation Timing



(2) Scan mode

The four analog input channels of scans 0 to 3, which are specified by the analog input channel specification register (ADS), while the ADMD bit of A/D converter mode register (ADM) is 1, are A/D converted successively. A/D conversion is performed in sequence, starting from the analog input channel specified by scan 0.

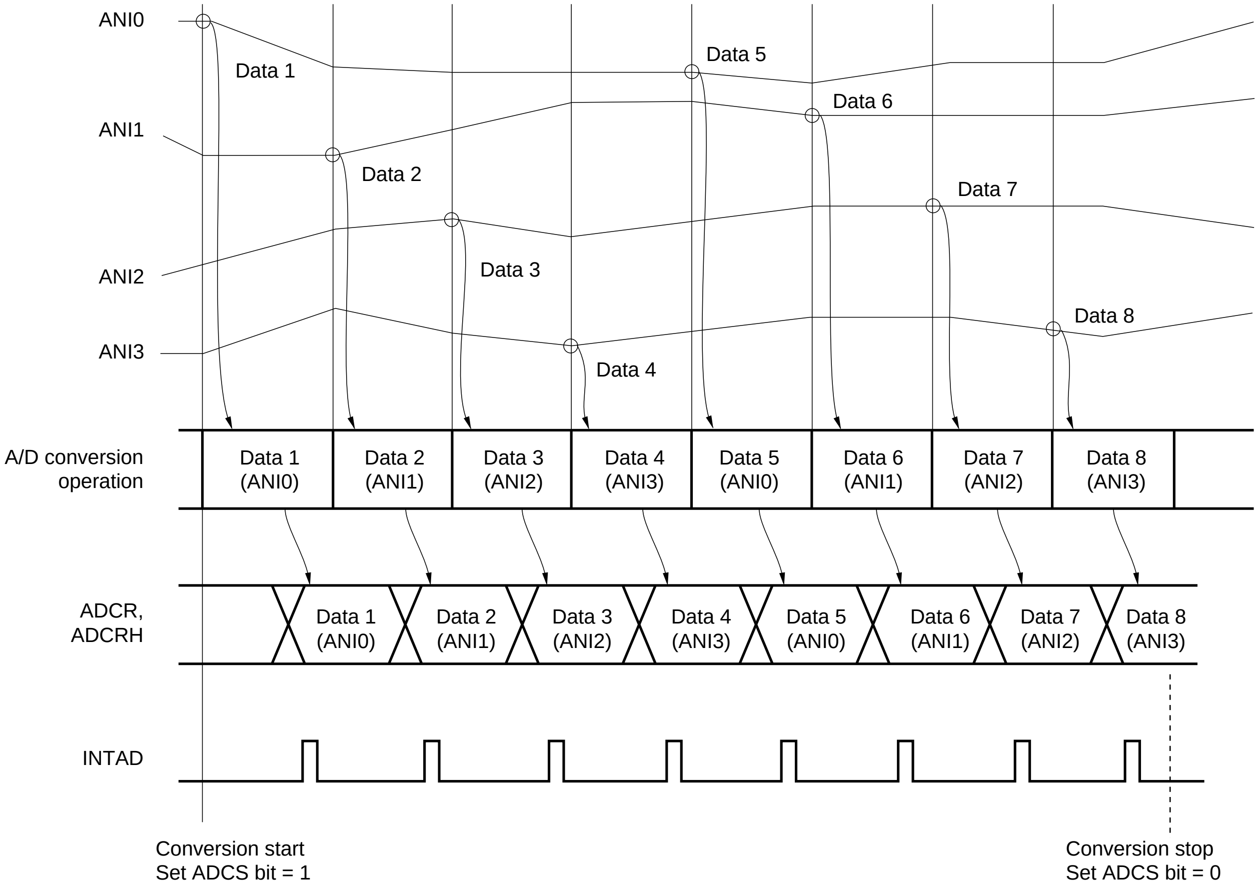
When A/D conversion of one analog input is complete, the conversion result is stored in the A/D conversion result register (ADCR) and the A/D conversion end interrupt request signal (INTAD) is generated.

The A/D conversion results of all the analog input channels are stored in ADCR. It is therefore recommended to save the contents of ADCR to RAM, once A/D conversion of one analog input channel has been completed.

When one A/D conversion ends, the next A/D conversion is started successively, regardless of being set to the trigger mode.

If anything is written to ADM, ADM1, or ADS during conversion, A/D conversion is aborted. In this case, A/D conversion is started again from the analog input channel of scan 0, regardless of being in the software trigger mode or timer trigger mode (hardware trigger mode).

Figure 11-16. Example of Scan 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 to 0 (PM27 to PM20) of port mode register 2 (PM2), bit 0 (PM80) of port mode register 8 (PM8), and bits 2 to 0 (PM151 to PM150) of port mode register 15 (PM15).

<5> Set the operational amplifier operation to set the operational amplifier output (OAI pin) for the analog input channel (refer to **8.4.1 Starting comparator and operational amplifier operation**).

<6> Select a channel to be used by using bits 6 and 3 to 0 (ADOAS, ADS3 to ADS0) of the analog input channel specification register (ADS).

<7> Use bits 0 and 7 (ADTRS, ADTMD) of A/D converter mode register 1 (ADM1) to set the trigger mode.

<8> Set bit 7 (ADCS) of ADM to 1 to start A/D conversion.

<9> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.

<10> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Change the channel>

<11> Change the channel using bits 6 and 3 to 0 (ADOAS, ADS3 to ADS0) of ADS to start A/D conversion.

<12> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.

<13> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>

<14> Clear ADCS to 0.

<15> Clear ADCE to 0.

<16> Clear bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 0.

Cautions 1. Make sure the period of <3> to <8> is 1 μs or more.

2. <3> may be done between <4> and <6>.

3. <3> can be omitted. However, ignore data of the first conversion after <8> in this case.

4. The period from <9> to <12> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <11> to <12> is the conversion time set using FR2 to FR0, LV1, and LV0.

11.5 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.

$1\text{LSB} = 1/2^{10} = 1/1024$
$= 0.098\%\text{FSR}$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, 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 11-17. Overall Error

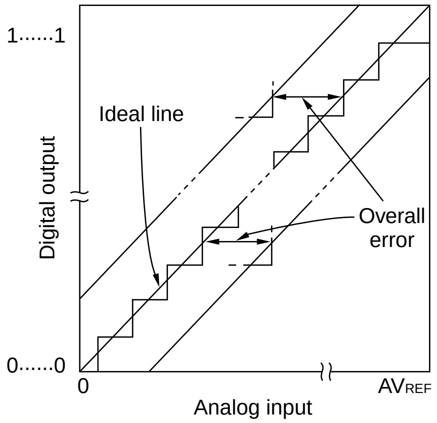
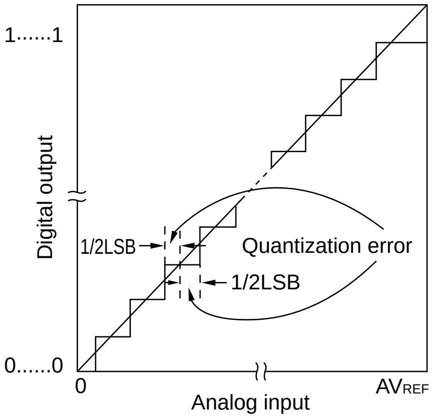


Figure 11-18. Quantization Error



(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 11-19. Zero-Scale Error

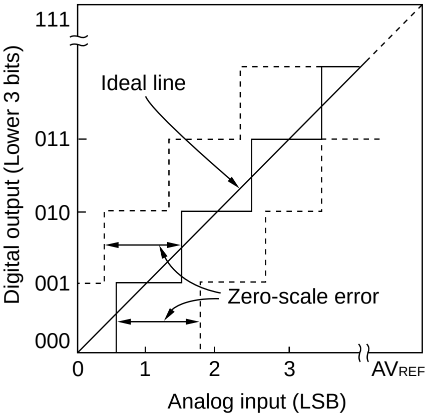


Figure 11-20. Full-Scale Error

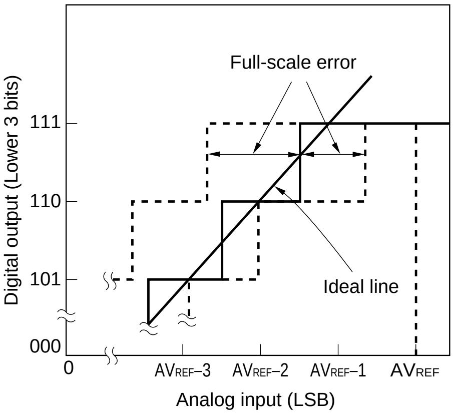


Figure 11-21. Integral Linearity Error

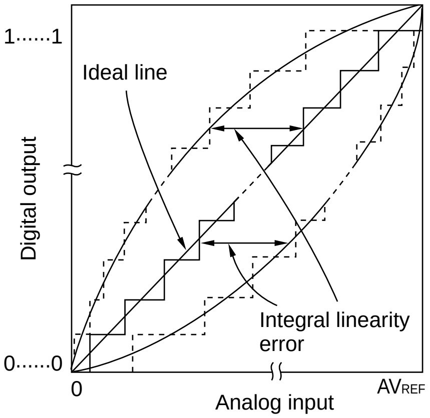
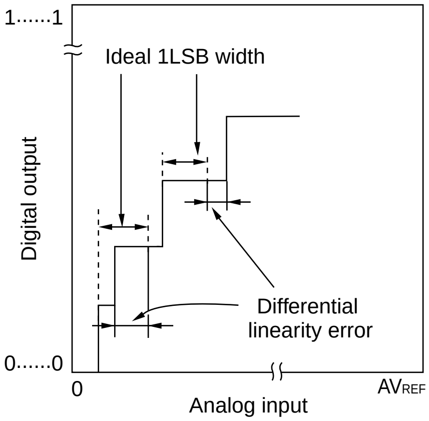


Figure 11-22. Differential Linearity Error



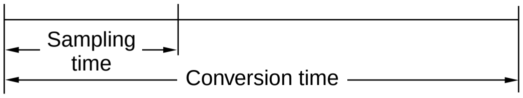
(8) Conversion time

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.



11.6 Cautions for A/D Converter

(1) Operating current in STOP mode

The A/D converter stops operating in the STOP mode. At this time, the operating current can be reduced by clearing bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register (ADM) to 0.

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 ANI0 to ANI9

Observe the rated range of the ANI0 to ANI9 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 ANI0 to ANI9.

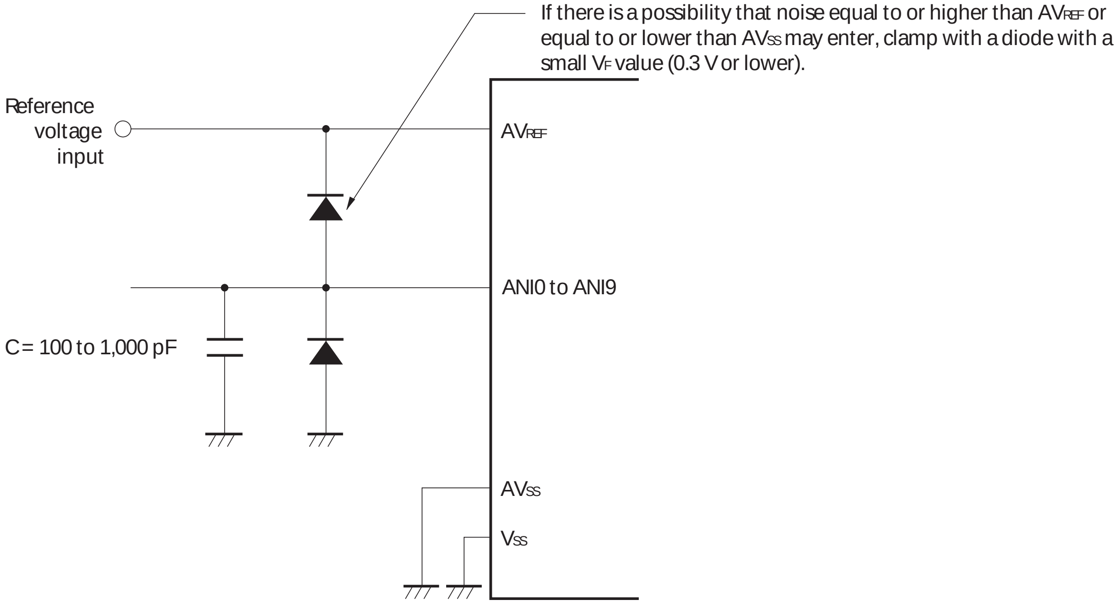
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<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 11-23 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 11-23. Analog Input Pin Connection


If there is a possibility that noise equal to or higher than AVREF or equal to or lower than AVSS may enter, clamp with a diode with a small VF value (0.3 V or lower).
Reference voltage input
AVREF
ANI0 to ANI9
C = 100 to 1,000 pF
AVSS
VSS

(5) ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI9/P151

<1> The analog input pins (ANI0 to ANI7) are also used as input port pins (P20 to P27).
The analog input pins (ANI8 to ANI19) are also used as input port pins (P150 to P151).
When A/D conversion is performed with any of ANI0 to ANI9 selected, do not access P20 to P27 and P150 to P151 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P20 to P27 and P150 to P151 starting with the ANI0/P20 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 ANI0 to ANI9 pins

This A/D converter charges a sampling capacitor for sampling during sampling time.

Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.

To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI9 pins (see **Figure 11-23**).

(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.

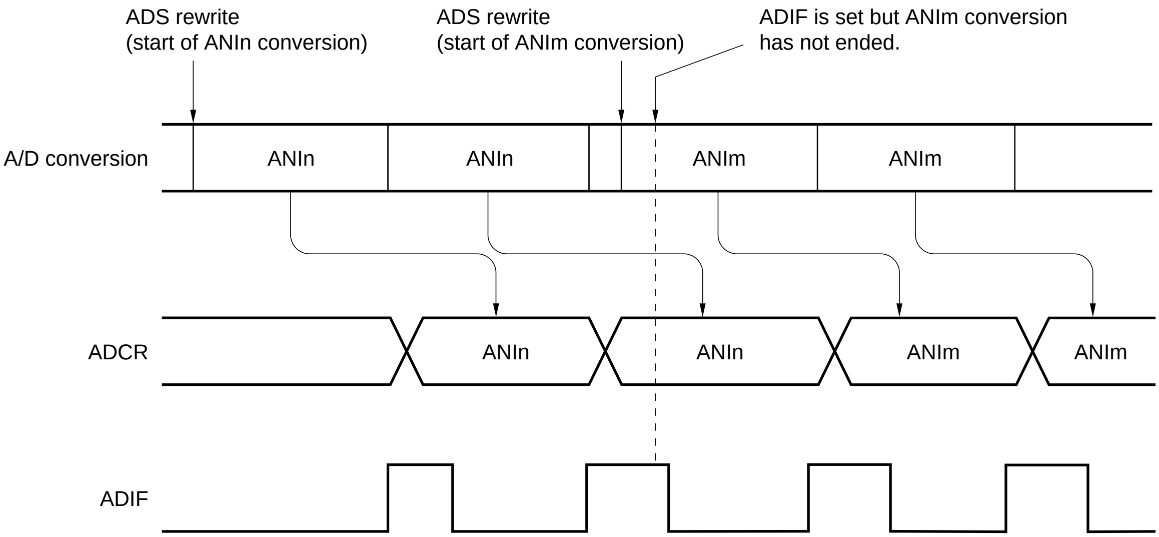
(8) Interrupt request flag (ADIF)

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 11-24. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 9

2. m = 0 to 9

(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, or if the ADCS bit is set to 1 with the ADCE bit = 0. 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), A/D converter mode register 1 (ADM1), 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, ADM1, 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 11-25. Internal Equivalent Circuit of ANIn Pin

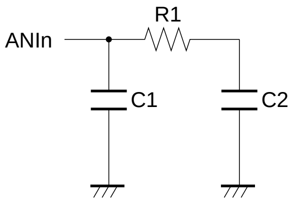


Table 11-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	Mode	R1	C1	C2
4.0 V ≤ V_{DD} ≤ 5.5 V	Standard	5.2 kΩ	8 pF	6.26 pF
	High speed 1	5.2 kΩ		
	High speed 2	7.8 kΩ		
2.7 V ≤ V_{DD} < 4.0 V	Standard	18.6 kΩ		
	High speed 2	7.8 kΩ		

Remarks 1. The resistance and capacitance values shown in Table 11-4 are not guaranteed values.

2. n = 0 to 9

CHAPTER 12 SERIAL ARRAY UNIT

The serial array unit has four serial channels per unit and can use two or more of various serial interfaces (3-wire serial (CSI), UART, and simplified I^2C) in combination.

Function assignment of each channel supported by the μPD79F9211 is as shown below.

Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	CSI00	UART0 (supporting LIN-bus)	–
1	CSI01		–
2	CSI10	UART1	IIC10
3	–		–

(Example of combination) When “UART0” is used for channels 0 and 1, CSI00[Note] and CSI01[Note] cannot be used, but CSI10, UART1, or IIC10 can be used.

12.1 Functions of Serial Array Unit

Each serial interface supported by the μPD79F9211 has the following features.

12.1.1 3-wire serial I/O (CSI00, CSI01, CSI10)

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- 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

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

12.1.2 UART (UART0, UART1)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[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 UART0 (0 and 1 channels)

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

} External interrupt (INTP0) or timer array unit TAUS is used.

12.1.3 Simplified I^{2}C (IIC10)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I^{2}C 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 and does not have a function to detect wait states.

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 and ACK detection functions
- 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 I^{2}C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

12.2 Configuration of Serial Array Unit

Serial array unit includes the following hardware.

Table 12-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits
Buffer register	Lower 8 bits of serial data register 0n (SDR0n)[Note]
Serial clock I/O	SCK00, SCK01, SCK10 pins (for 3-wire serial I/O), SCL10 pin (for simplified I^2C)
Serial data input	SI00, SI01, SI10 pins (for 3-wire serial I/O), R_XD0 pin (for UART supporting LIN-bus), R_XD1 pins (for UART)
Serial data output	SO00, SO01, SO10 pins (for 3-wire serial I/O), T_XD0 pin (for UART supporting LIN-bus), T_XD1 pin (for UART), output controller
Serial data I/O	SDA10 pin (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register 0 (SPS0) • Serial channel enable status register 0 (SE0) • Serial channel start register 0 (SS0) • Serial channel stop register 0 (ST0) • Serial output enable register 0 (SOE0) • Serial output register 0 (SO0) • Serial output level register 0 (SOL0) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register 0n (SDR0n) • Serial mode register 0n (SMR0n) • Serial communication operation setting register 0n (SCR0n) • Serial status register 0n (SSR0n) • Serial flag clear trigger register 0n (SIR0n) • Port input mode registers 3, 7 (PIM3, PIM7) • Port output mode registers 3, 7 (POM3, POM7) • Port mode registers 3, 7 (PM3, PM7) • Port registers 3, 7 (P3, P7)

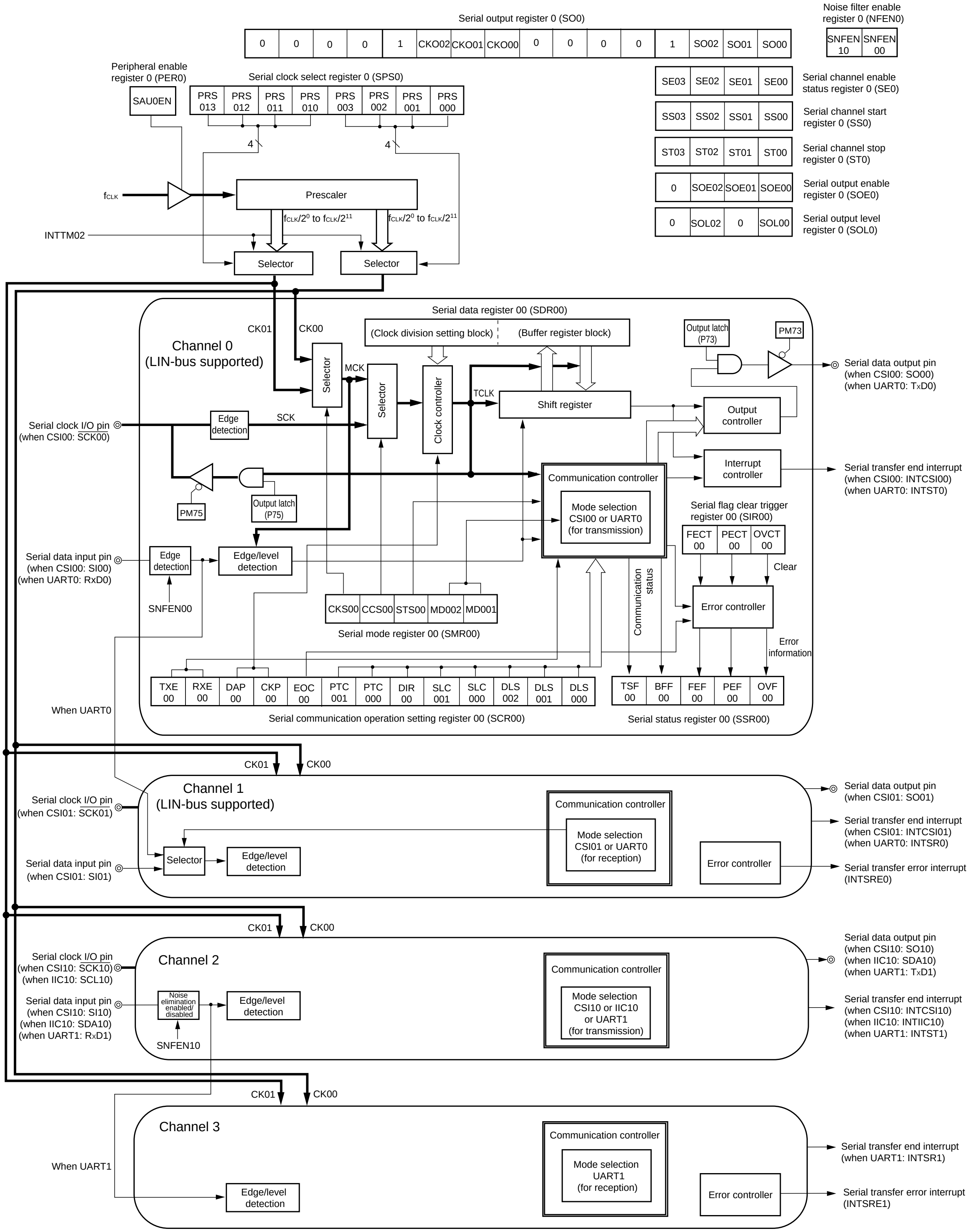
Notes 1. The lower 8 bits of the serial data register 0n (SDR0n) can be read or written as the following SFR, depending on the communication mode.

- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)
- IIC10 communication … SIO10 (IIC10 data register)

Remark n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10),
q: UART number (q = 0, 1)

Figure 12-1 shows the block diagram of serial array unit.

Figure 12-1. Block Diagram of Serial Array Unit 1


Serial output register 0 (SO0)
0 0 0 0 1 CKO02 CKO01 CKO00 0 0 0 0 1 SO02 SO01 SO00
Noise filter enable register 0 (NFEN0)
SNFEN 10 SNFEN 00
Peripheral enable register 0 (PER0)
SAU0EN
Serial clock select register 0 (SPS0)
PRS 013 PRS 012 PRS 011 PRS 010 PRS 003 PRS 002 PRS 001 PRS 000
4
4
SE03 SE02 SE01 SE00
Serial channel enable status register 0 (SE0)
SS03 SS02 SS01 SS00
Serial channel start register 0 (SS0)
ST03 ST02 ST01 ST00
Serial channel stop register 0 (ST0)
0 SOE02 SOE01 SOE00
Serial output enable register 0 (SOE0)
0 SOL02 0 SOL00
Serial output level register 0 (SOL0)
fCLK
Prescaler
fCLK/2^0 to fCLK/2^11
fCLK/2^0 to fCLK/2^11
INTTM02
Selector
Selector
Serial data register 00 (SDR00)
(Clock division setting block)
(Buffer register block)
CK01
CK00
Channel 0
(LIN-bus supported)
Output latch (P73)
PM73
Selector
MCK
Selector
Clock controller
TCLK
Shift register
Serial data output pin (when CSI00: SO00) (when UART0: TxD0)
Output controller
Serial clock I/O pin (when CSI00: SCK00)
Edge detection
SCK
Interrupt controller
Serial transfer end interrupt (when CSI00: INTCSI00) (when UART0: INTST0)
Communication controller
Mode selection CSI00 or UART0 (for transmission)
PM75
Output latch (P75)
Serial flag clear trigger register 00 (SIR00)
FECT 00 PECT 00 OVCT 00
Clear
Serial data input pin (when CSI00: SI00) (when UART0: RxD0)
Edge detection
Edge/level detection
SNFEN00
Communication status
Error controller
CKS00 CCS00 STS00 MD002 MD001
Serial mode register 00 (SMR00)
Error information
TXE 00 RXE 00 DAP 00 CKP 00 EOC 00 PTC 001 PTC 000 DIR 00 SLC 001 SLC 000 DLS 002 DLS 001 DLS 000
Serial communication operation setting register 00 (SCR00)
TSF 00 BFF 00 FEF 00 PEF 00 OVF 00
Serial status register 00 (SSR00)
When UART0
CK01
CK00
Channel 1
(LIN-bus supported)
Serial clock I/O pin (when CSI01: SCK01)
Serial data input pin (when CSI01: SI01)
Selector
Edge/level detection
Communication controller
Mode selection CSI01 or UART0 (for reception)
Error controller
Serial data output pin (when CSI01: SO01)
Serial transfer end interrupt (when CSI01: INTCSI01) (when UART0: INTSR0)
Serial transfer error interrupt (INTSRE0)
CK01
CK00
Channel 2
Serial clock I/O pin (when CSI10: SCK10) (when IIC10: SCL10)
Serial data input pin (when CSI10: SI10) (when IIC10: SDA10) (when UART1: RxD1)
Noise elimination enabled/ disabled
SNFEN10
Edge/level detection
Communication controller
Mode selection CSI10 or IIC10 or UART1 (for transmission)
Serial data output pin (when CSI10: SO10) (when IIC10: SDA10) (when UART1: TxD1)
Serial transfer end interrupt (when CSI10: INTCSI10) (when IIC10: INTIIC10) (when UART1: INTST1)
CK01
CK00
Channel 3
When UART1
Edge/level detection
Communication controller
Mode selection UART1 (for reception)
Error controller
Serial transfer end interrupt (when UART1: INTSR1)
Serial transfer error interrupt (INTSRE1)

(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 0n (SDR0n).

	7	6	5	4	3	2	1	0
Shift register								

(2) Lower 8 bits of the serial data register 0n (SDR0n)

SDR0n 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 (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 (DLS0n0 to DLS0n2) of the SCR0n register, regardless of the output sequence of the data.

- 5-bit data length (stored in bits 0 to 4 of SDR0n register) (settable in UART mode only)
- 7-bit data length (stored in bits 0 to 6 of SDR0n register)
- 8-bit data length (stored in bits 0 to 7 of SDR0n register)

SDR0n can be read or written in 16-bit units.

The lower 8 bits of SDR0n of SDR0n can be read or written[Note] as the following SFR, depending on the communication mode.

- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)
- IIC10 communication … SIO10 (IIC10 data register)

Reset signal generation clears this register to 0000H.

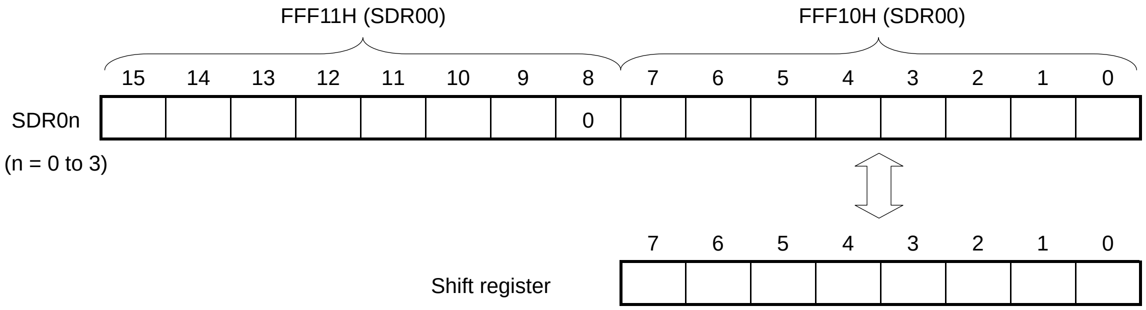
Note Writing in 8-bit units is prohibited when the operation is stopped (SE0n = 0).

Remarks 1. After data is received, “0” is stored in bits 0 to 7 in bit portions that exceed the data length.

2. n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10),
q: UART number (q = 0, 1)

Figure 12-2. Format of Serial Data Register 0n (SDR0n)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), After reset: 0000H R/W
FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03)



Caution Be sure to clear bit 8 to “0”.

Remarks 1. For the function of the higher 7 bits of SDR0n, see **12.3 Registers Controlling Serial Array Unit**.

2. n: Channel number (n = 0 to 3),

12.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register 0 (SPS0)
- Serial mode register 0n (SMR0n)
- Serial communication operation setting register 0n (SCR0n)
- Serial data register 0n (SDR0n)
- Serial status register 0n (SSR0n)
- Serial flag clear trigger register 0n (SIR0n)
- Serial channel enable status register 0 (SE0)
- Serial channel start register 0 (SS0)
- Serial channel stop register 0 (ST0)
- Serial output enable register 0 (SOE0)
- Serial output level register 0 (SOL0)
- Serial output register 0 (SO0)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 3, 7 (PIM3, PIM7)
- Port output mode registers 3, 7 (POM3, POM7)
- Port mode registers 3, 7 (PM3, PM7)
- Port registers 3, 7 (P3, P7)

Remark n: Channel number (n = 0 to 3)

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. 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 is used, be sure to set bit 2 (SAU0EN) 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 12-3. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

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

Cautions 1. When setting serial array unit, be sure to set SAU0EN to 1 first. If SAU0EN = 0, writing to a control register of serial array unit is ignored, and, even if the register is read, only the default value is read (except for input switch control register (ISC), noise filter enable register (NFEN0), port input mode registers (PIM3, PIM7), port output mode registers (POM3, POM7), port mode registers (PM3, PM7), and port registers (P3, P7)).

2. After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

3. Be sure to clear bits 0, 1, 3, 4, and 6 of PER0 register to 0.

(2) Serial clock select register 0 (SPS0)

SPS0 is a 16-bit register that is used to select two types of operation clocks (CK00, CK01) that are commonly supplied to each channel. CK01 is selected by bits 7 to 4 of SPS0, and CK00 is selected by bits 3 to 0.

Rewriting SPS0 is prohibited when the register is in operation (when SE0n = 1).

SPS0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SPS0 can be set with an 8-bit memory manipulation instruction with SPS0L.

Reset signal generation clears this register to 0000H.

Figure 12-4. Format of Serial Clock Select Register 0 (SPS0)

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
SPS0	0	0	0	0	0	0	0	0	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 0p3	PRS 0p2	PRS 0p1	PRS 0p0	Section of operation clock (CK0p) 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 Note 2				
Other than above				Setting prohibited				

Notes1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (ST0 = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 for the operation clock, also stop the timer array unit TAUS (TT0 = 00FFH).

2. SAU can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by setting the TIS02 bit of the TIS0 register of TAUS to 1, selecting $f_{SUB}/4$ for the input clock, and selecting INTTM02 using the SPS0 register. When changing f_{CLK}, however, SAU and TAUS must be stopped as described in Note 1 above.

Cautions 1. Be sure to clear bits 15 to 8 to "0".

2. After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

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

f_{SUB}: Subsystem clock frequency

2. p = 0, 1

(3) Serial mode register 0n (SMR0n)

SMR0n is a register that sets an operation mode of channel n. It is also used to select an operation clock (MCK), specify whether the serial clock (SCK) may be input or not, set a start trigger, an operation mode (CSI, UART, or I^{2}C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting SMR0n is prohibited when the register is in operation (when SE0n = 1). However, the MD0n0 bit can be rewritten during operation.

SMR0n can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets this register to 0020H.

Figure 12-5. Format of Serial Mode Register 0n (SMR0n) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS 0n	CCS 0n	0	0	0	0	0	STS 0n	0	SIS 0n0	1	0	0	MD 0n2	MD 0n1	MD 0n0

CKS 0n	Selection of operation clock (MCK) of channel n
0	Prescaler output clock CK00 set by PRS register
1	Prescaler output clock CK01 set by PRS register
Operation clock MCK is used by the edge detector. In addition, depending on the setting of the CCS0n bit and the higher 7 bits of the SDR0n register, a transfer clock (TCLK) is generated.	

CCS 0n	Selection of transfer clock (TCLK) of channel n
0	Divided operation clock MCK specified by CKS0n bit
1	Clock input from SCK pin (slave transfer in CSI mode)
Transfer clock TCLK is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCS0n = 0, the division ratio of MCK is set by the higher 7 bits of the SDR0n register.	

STS 0n	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^{2}C).
1	Valid edge of R$_X$D pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SS0 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 n: Channel number (n = 0 to 3)

Figure 12-5 Format of Serial Mode Register 0n (SMR0n) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS 0n	CCS 0n	0	0	0	0	0	STS 0n	0	SIS 0n0	1	0	0	MD 0n2	MD 0n1	MD 0n0

SIS 0n0	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 0n2	MD 0n1	Setting of operation mode of channel n
0	0	CSI mode
0	1	UART mode
1	0	Simplified I^{2}C mode
1	1	Setting prohibited

MD 0n0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt
For successive transmission, the next transmit data is written by setting MD0n0 to 1 when SDR0n data has run out.	

Remark n: Channel number (n = 0 to 3)

(4) Serial communication operation setting register 0n (SCR0n)

SCR0n 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 SCR0n is prohibited when the register is in operation (when SE0n = 1).

SCR0n can be set by a 16-bit memory manipulation instruction.

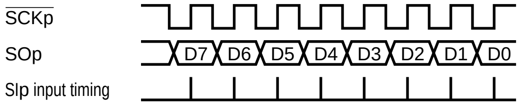
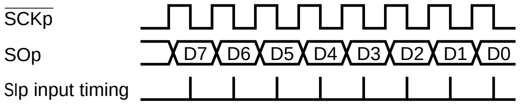
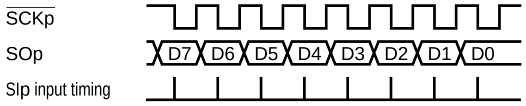
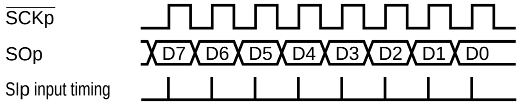
Reset signal generation sets this register to 0087H.

Figure 12-6. Format of Serial Communication Operation Setting Register 0n (SCR0n) (1/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n	RXE0n	DAP0n	CKP0n	0	EOC0n	PTC0n1	PTC0n0	DIR0n	0	SLC0n1	SLC0n0	0	DLS0n2	DLS0n1	DLS0n0

TXE0n	RXE0n	Setting of operation mode of channel n
0	0	Does not start communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP0n	CKP0n	Selection of data and clock phase in CSI mode
0	0	
0	1	
1	0	
1	1	
Be sure to set DAP0n, CKP0n = 0, 0 in the UART mode and simplified I^{2}C mode.		

Caution Be sure to clear bits 3, 6, and 11 to “0”. Be sure to set bit 2 to “1”.

Remark n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 01, 10)

Figure 12-6. Format of Serial Communication Operation Setting Register 0n (SCR0n) (2/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE 0n	RXE 0n	DAP 0n	CKP 0n	0	EOC 0n	PTC 0n1	PTC 0n0	DIR 0n	0	SLC 0n1	SLC 0n0	0	DLS 0n2	DLS 0n1	DLS 0n0

EOC 0n	Selection of masking of error interrupt signal (INTSREx (x = 0 , 1))
0	Masks error interrupt INTSREx (INTSRx is not masked).
1	Enables generation of error interrupt INTSREx (INTSRx is masked if an error occurs).
Set EOC0n = 0 in the CSI mode, simplified I^{2}C mode, and during UART transmission[Note]. Set EOC0n = 1 during UART reception.	

PTC 0n1	PTC 0n0	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.	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 PTC0n1, PTC0n0 = 0, 0 in the CSI mode and simplified I^{2}C mode.			

DIR 0n	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 DIR0n = 0 in the simplified I^{2}C mode.	

SLC 0n1	SLC 0n0	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 (SLC0n1, SLC0n0 = 0, 1) during UART reception and in the simplified I^{2}C mode. Set no stop bit (SLC0n1, SLC0n0 = 0, 0) in the CSI mode.		

Note When using CSI01 not with EOC01 = 0, error interrupt INTSRE0 may be generated.

Caution Be sure to clear bits 3, 6, and 11 to “0”. Be sure to set bit 2 to “1”.

Remark n: Channel number (n = 0 to 3)

Figure 12-6. Format of Serial Communication Operation Setting Register 0n (SCR0n) (3/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE 0n	RXE 0n	DAP 0n	CKP 0n	0	EOC 0n	PTC 0n1	PTC 0n0	DIR 0n	0	SLC 0n1	SLC 0n0	0	DLS 0n2	DLS 0n1	DLS 0n0

DLS 0n2	DLS 0n1	DLS 0n0	Setting of data length in CSI and UART modes
1	0	0	5-bit data length (stored in bits 0 to 4 of SDR0n register) (settable in UART mode only)
1	1	0	7-bit data length (stored in bits 0 to 6 of SDR0n register)
1	1	1	8-bit data length (stored in bits 0 to 7 of SDR0n register)
Other than above			Setting prohibited
Be sure to set DLS0n0 = 1 in the simplified I^{2}C mode.			

Caution Be sure to clear bits 3, 6, and 11 to “0”. Be sure to set bit 2 to “1”.

Remark n: Channel number (n = 0 to 3)

(5) Higher 7 bits of the serial data register 0n (SDR0n)

SDR0n 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 (MCK).

If the CCS0n bit of serial mode register 0n (SMR0n) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of SDR0n is used as the transfer clock.

For the function of the lower 8 bits of SDR0n, see **12.2 Configuration of Serial Array Unit**.

SDR0n 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 (SE0n = 0). During operation (SE0n = 1), a value is written only to the lower 8 bits of SDR0n. When SDR0n is read during operation, 0 is always read.

Reset signal generation clears this register to 0000H.

Figure 12-7. Format of Serial Data Register 0n (SDR0n)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), After reset: 0000H R/W
FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03)

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n								0								

SDR0n[15:9]							Transfer clock setting by dividing the operating clock (MCK)
0	0	0	0	0	0	0	MCK/2
0	0	0	0	0	0	1	MCK/4
0	0	0	0	0	1	0	MCK/6
0	0	0	0	0	1	1	MCK/8
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	MCK/254
1	1	1	1	1	1	1	MCK/256

Cautions 1. Be sure to clear bit 8 to “0”.

2. Setting SDR0n[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

Remarks 1. For the function of the lower 8 bits of SDR0n, see **12.2 Configuration of Serial Array Unit**.

2. n: Channel number (n = 0 to 3)

(6) Serial status register 0n (SSR0n)

SSR0n 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.

SSR0n can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of SSR0n can be set with an 8-bit memory manipulation instruction with SSR0nL.

Reset signal generation clears this register to 0000H.

Figure 12-8. Format of Serial Status Register 0n (SSR0n) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR0n	0	0	0	0	0	0	0	0	0	TSF 0n	BFF 0n	0	0	FEF 0n	PEF 0n	OVF 0n

TSF 0n	Communication status indication flag of channel n
0	Communication is not under execution.
1	Communication is under execution.
Because this flag is an updating flag, it is automatically cleared when the communication operation is completed. This flag is cleared also when the ST0n/SS0n bit is set to 1.	

BFF 0n	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDR0n register.
1	Valid data is stored in the SDR0n register.
This is an updating flag. It is automatically cleared when transfer from the SDR0n register to the shift register is completed. During reception, it is automatically cleared when data has been read from the SDR0n register. This flag is cleared also when the ST0n/SS0n bit is set to 1. This flag is automatically set if transmit data is written to the SDR0n register when the TXE0n bit of the SCR0n register = 1 (transmission or reception mode in each communication mode). It is automatically set if receive data is stored in the SDR0n register when the RXE0n bit of the SCR0n register = 1 (transmission or reception mode in each communication mode). It is also set in case of a reception error. If data is written to the SDR0n register when BFF0n = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVF0n = 1) is detected.	

Remark n: Channel number (n = 0 to 3)

Figure 12-8. Format of Serial Status Register 0n (SSR0n) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR0n	0	0	0	0	0	0	0	0	0	TSF 0n	BFF 0n	0	0	FEF 0n	PEF 0n	OVF 0n

FEF 0n	Framing error detection flag of channel n
0	No error occurs.
1	A framing error occurs during UART reception. <Framing error cause> A framing error occurs if the stop bit is not detected upon completion of UART reception.
This is a cumulative flag and is not cleared until 1 is written to the FECT0n bit of the SIR0n register.	

PEF 0n	Parity error detection flag of channel n
0	Error does not occur.
1	A parity error occurs during UART reception or ACK is not detected during I^2C transmission. <Parity error cause> • A parity error occurs if the parity of transmit data does not match the parity bit on completion of UART reception. • ACK is not detected if the ACK signal is not returned from the slave in the timing of ACK reception during I^2C transmission.
This is a cumulative flag and is not cleared until 1 is written to the PECT0n bit of the SIR0n register.	

OVF 0n	Overrun error detection flag of channel n
0	No error occurs.
1	An overrun error occurs. <Causes of overrun error> • Receive data stored in the SDR0n register is not read and transmit data is written or the next receive data is written. • Transmit data is not ready for slave transmission or reception in the CSI mode.
This is a cumulative flag and is not cleared until 1 is written to the OVCT0n bit of the SIR0n register.	

Remark n: Channel number (n = 0 to 3)

(7) Serial flag clear trigger register 0n (SIR0n)

SIR0n is a trigger register that is used to clear each error flag of channel n.

When each bit (FECT0n, PECT0n, OVCT0n) of this register is set to 1, the corresponding bit (FEF0n, PEF0n, OVF0n) of serial status register 0n is cleared to 0. Because SIR0n is a trigger register, it is cleared immediately when the corresponding bit of SSR0n is cleared.

SIR0n can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SIR0n can be set with an 8-bit memory manipulation instruction with SIR0nL.

Reset signal generation clears this register to 0000H.

Figure 12-9. Format of Serial Flag Clear Trigger Register 0n (SIR0n)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIR0n	0	0	0	0	0	0	0	0	0	0	0	0	0	FEC T0n	PEC T0n	OVC T0n

FEC T0n	Clear trigger of framing error of channel n
0	No trigger operation
1	Clears the FEF0n bit of the SSR0n register to 0.

PEC T0n	Clear trigger of parity error flag of channel n
0	No trigger operation
1	Clears the PEF0n bit of the SSR0n register to 0.

OVC T0n	Clear trigger of overrun error flag of channel n
0	No trigger operation
1	Clears the OVF0n bit of the SSR0n register to 0.

Caution Be sure to clear bits 15 to 3 to “0”.

Remarks 1. n: Channel number (n = 0 to 3)

2. When the SIR0n register is read, 0000H is always read.

(8) Serial channel enable status register 0 (SE0)

SE0 indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written a bit of serial channel start register 0 (SS0), the corresponding bit of this register is set to 1.

When 1 is written a bit of serial channel stop register 0 (ST0), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of CKO0n of the serial output register 0 (SO0) 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 CKO0n of the SO0 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.

SE0 can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of SE0 can be set with an 1-bit or 8-bit memory manipulation instruction with SE0L.

Reset signal generation clears this register to 0000H.

Figure 12-10. Format of Serial Channel Enable Status Register 0 (SE0)

Address: F0120H, F0121H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03	SE02	SE01	SE00

SE0n	Indication of operation enable/stop status of channel n
0	Operation stops (stops with the values of the control register and shift register, and the statuses of the serial clock I/O pin, serial data output pin, and the FEF, PEF, and OVF error flags retained[Note]).
1	Operation is enabled.

Note Bits 6 and 5 (TSF0n, BFF0n) of the SSR0n register are cleared.

Remark n: Channel number (n = 0 to 3)

(9) Serial channel start register 0 (SS0)

SS0 is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SS0n), the corresponding bit (SE0n) of serial channel enable status register 0 (SE0) is set to 1. Because SS0n is a trigger bit, it is cleared immediately when SE0n = 1.

SS0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SS0 can be set with an 1-bit or 8-bit memory manipulation instruction with SS0L.

Reset signal generation clears this register to 0000H.

Figure 12-10. Format of Serial Channel Start Register 0 (SS0)

Address: F0122H, F0123H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

SS0n	Operation start trigger of channel n
0	No trigger operation
1	Sets SE0n to 1 and enters the communication wait status (if a communication operation is already under execution, the operation is stopped and the start condition is awaited).

Caution Be sure to clear bits 15 to 4 to “0”.

Remarks 1. n: Channel number (n = 0 to 3)

2. When the SS0 register is read, 0000H is always read.

(10) Serial channel stop register 0 (ST0)

ST0 is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (ST0n), the corresponding bit (SE0n) of serial channel enable status register 0 (SE0) is cleared to 0. Because ST0n is a trigger bit, it is cleared immediately when SE0n = 0.

ST0 can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of ST0 can be set with an 1-bit or 8-bit memory manipulation instruction with ST0L.

Reset signal generation clears this register to 0000H.

Figure 12-12. Format of Serial Channel Stop Register 0 (ST0)

Address: F0124H, F0125H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

ST0n	Operation stop trigger of channel n
0	No trigger operation
1	Clears SE0n to 0 and stops the communication operation. (Stops with the values of the control register and shift register, and the statuses of the serial clock I/O pin, serial data output pin, and the FEF, PEF, and OVF error flags retained[Note]).

Note Bits 6 and 5 (TSF0n, BFF0n) of the SSR0n register are cleared.

Caution Be sure to clear bits 15 to 4 to "0".

Remarks **1.** n: Channel number (n = 0 to 3)

2. When the ST0 register is read, 0000H is always read.

(11) Serial output enable register 0 (SOE0)

SOE0 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 SO0n of the serial output register 0 (SO0) 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 SO0n value of the SO0 register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and stop condition can be created by software.

SOE0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SOE0 can be set with an 1-bit or 8-bit memory manipulation instruction with SOE0L.

Reset signal generation clears this register to 0000H.

Figure 12-13. Format of Serial Output Enable Register 0 (SOE0)

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	SOE 02	SOE 01	SOE 00

SOE 0n	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 3of SOE0 to “0”.

Remark n: Channel number (n = 0 to 2)

(12) Serial output register 0 (SO0)

SO0 is a buffer register for serial output of each channel.

The value of bit n of this register is output from the serial data output pin of channel n.

The value of bit (n + 8) of this register is output from the serial clock output pin of channel n.

SO0n of this register can be rewritten by software only when serial output is disabled (SOE0n = 0). When serial output is enabled (SOE0n = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

CKO0n of this register can be rewritten by software only when the channel operation is stopped (SE0n = 0). While channel operation is enabled (SE0n = 1), rewriting by software is ignored, and the value of CKO0n can be changed only by a serial communication operation.

To use the following pins as a port function pin, set the corresponding CKO0n and SO0n bits to “1”.

- P30/SO10/TxD1/TO11, P31/SI10/RxD1/SDA10/INTP1/TI09,
  32/$\overline{\text{SCK10}}$/SCL10/INTP2, P70/SO01/INTP4, P72/$\overline{\text{SCK01}}$/INTP6,
  P73/SO00/TxD0/TO10, P75/$\overline{\text{SCK00}}$/TI11

SO0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0F0FH.

Figure 12-14. Format of Serial Output Register 0 (SO0)

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	CKO 02	CKO 01	CKO 00	0	0	0	0	1	SO 02	SO 01	SO 00

CKO 0n	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SO 0n	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 and 3of SO0 to “1”. And be sure to clear bits 15 to 12 and 7 to 4 of SO0 to “0”.

Remark n: Channel number (n = 0 to 2)

(13) Serial output level register 0 (SOL0)

SOL0 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^{2}C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOE0n = 1). When serial output is disabled (SOE0n = 0), the value of the SO0n bit is output as is.

Rewriting SOL0 is prohibited when the register is in operation (when SE0n = 1).

SOL0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SOL0 can be set with an 8-bit memory manipulation instruction with SOL0L.

Reset signal generation clears this register to 0000H.

Figure 12-15. Format of Serial Output Level Register 0 (SOL0)

Address: F0134H, F0135H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 02	0	SOL 00

SOL 0n	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 n: Channel number (n = 0, 2)

(14) Input switch control register (ISC)

ISC is used to realize a LIN-bus communication operation by UART0 in coordination with an external interrupt and the timer array unit TAUS.

When bit 0 is set to 1, the input signal of the serial data input (RxD0) 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 (RxD0) pin is selected as a timer input, so that the pulse widths of a sync break field and a 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 12-16. 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	ISC4	ISC3	ISC2	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit TAUS
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of RxD0 pin is used as timer input (wakeup signal detection).

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 RxD0 pin as an external interrupt (to measure the pulse widths of the sync break field and sync field).

Caution Be sure to clear bits 7 to 5 to “0”.

(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^{2}C 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 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 12-17. 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	0	0	0	0	SNFEN10	0	SNFEN00

SNFEN10	Use of noise filter of RxD1/SDA10/SI10/INTP1/TI09/P31 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN10 to 1 to use the RxD1 pin. Clear SNFEN10 to 0 to use the SDA10, SI10, INTP1, TI09, and P31 pins.	

SNFEN00	Use of noise filter of RxD0/SI00/TI10/P74 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN00 to 1 to use the RxD0 pin. Clear SNFEN00 to 0 to use the SI00, TI10, and P74 pins.	

Caution Be sure to clear bits 7 to 3, and 1 to “0”.

(16) Port input mode registers 3, 7 (PIM3, PIM7)

These registers set the input buffer of ports 3 and 7 in 1-bit units.

PIM3 and PIM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 12-18. Format of Port Input Mode Registers 3 and 7 (PIM3 and PIM7)

Address F0043H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM3	0	0	0	0	0	PIM32	PIM31	0

Address F0047H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM7	0	0	PIM75	PIM74	0	PIM72	PIM71	0

PIMmn	Pmn pin input buffer selection (m = 3, 7; n = 1, 2, 4, 5)
0	Normal input buffer
1	TTL input buffer

(17) Port output mode registers 3, 7 (POM3, POM7)

These registers set the output mode of ports 3 and 7 in 1-bit units.

POM3 and POM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 12-19. Format of Port Output Mode Registers 3 and 7 (POM3 and POM7)

Address F0053H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM3	0	0	0	0	0	POM32	POM31	POM30

Address F0057H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM7	0	0	POM75	0	POM73	POM72	0	POM70

POMmn	Pmn pin output buffer selection (m = 3, 7; n = 0 to 3, 5)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

(18) Port mode registers 3, 7 (PM3, PM7)

These registers set input/output of ports 3 and 7 in 1-bit units.

When using the P30/SO10/TxD1/TO11, P31/SI10/RxD1/SDA10/INTP1/TI09, P32/$\overline{\text{SCK10}}$/SCL10/INTP2, P70/SO01/INTP4, P72/$\overline{\text{SCK01}}$/INTP6, P73/SO00/TxD0/TO10, and P75/$\overline{\text{SCK00}}$/TI11 pins for serial data output or serial clock output, clear the PM30 to PM32, PM70, PM72, PM73, and PM75 bits to 0, and set the output latches of P30 to P32, P70, P72, P73, and P75 to 1.

When using the P31/SI10/RxD1/SDA10/INTP1/TI09, P32/$\overline{\text{SCK10}}$/SCL10/INTP2, P71/SI01/INTP5, P72/$\overline{\text{SCK01}}$/INTP6, P74/SI00/RxD0/TI10, and P75/$\overline{\text{SCK00}}$/TI11 pins for serial data input or serial clock input, set the PM31, PM32, PM71, PM72, PM74, and PM75 bits to 1. At this time, the output latches of P31, P32, P71, P72, P74, and P75 may be 0 or 1.

PM3 and PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 12-20. Format of Port Mode Registers 3 and 7 (PM3 and PM7)

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	PM32	PM31	PM30

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PMmn	Pmn pin I/O mode selection (m = 3, 7; n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

12.4 Operation stop mode

Each serial interface of serial array unit has the operation stop mode.

In this mode, serial communication cannot be executed, thus reducing the power consumption.

In addition, the following pins can be used as ordinary port pins in this mode.

- P30/$\overline{\text{SO10}}$/TxD1/TO11, P31/SI10/RxD1/SDA10/INTP1/TI09, P32/SCK10/SCL10/INTP2, P70/$\overline{\text{SO01}}$/INTP4, P71/SI01/INTP5, P72/SCK01/INTP6, P73/SO00/TxD0/TO10, P74/SI00/RxD0/TI10, P75/$\overline{\text{SCK00}}$/TI11

12.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

PER0 is used to enable or disable use of each peripheral hardware macro. 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, set bit 2 (SAU0EN) to 0.

Figure 12-21. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0)

	7	6	5	4	3	2	1	0
PER0	RTCEN ×	0	ADCEN ×	0	0	SAU0EN 0/1	0	0

Control of SAU input clock

0: Stops supply of input clock

1: Supplies input clock

Cautions 1. If SAU0EN = 0, writing to a control register of serial array unit is ignored, and, even if the register is read, only the default value is read (except for input switch control register (ISC), noise filter enable register (NFEN0), port input mode registers (PIM3, PIM7), port output mode registers (POM3, POM7), port mode registers (PM3, PM7), and port registers (P3, P7)).

2. Be sure to clear bits 0, 1, 3, 4, and 6 of PER0 register to 0.

Remark ▨: 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

12.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 12-22. Each Register Setting When Stopping the Operation by Channels (1/2)

(a) Serial Channel Enable Status Register 0 (SE0) … 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
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03 0/1	SE02 0/1	SE01 0/1	SE00 0/1

0: Operation stops

* The SE0 register is a read-only status register, whose operation is stopped by using the ST0 register.
With a channel whose operation is stopped, the value of CKO0n of the SO0 register can be set by software.

(b) Serial channel stop register 0 (ST0) … 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
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03 0/1	ST02 0/1	ST01 0/1	ST00 0/1

1: Clears SE0n to 0 and stops the communication operation

* Because ST0n is a trigger bit, it is cleared immediately when SE0n = 0.

(c) Serial output enable register 0 (SOE0) … 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	SOE02 0/1	SOE01 0/1	SOE00 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO0n value of the SO0 register can be set by software.

(d) Serial output register 0 (SO0) …This register is a buffer register for serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1	CKO01 0/1	CKO00 0/1	0	0	0	0	1	SO02 0/1	SO01 0/1	SO00 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 CKO0n and SO0n bits to “1”.

Remark 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

12.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- 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

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

The channels supporting 3-wire serial I/O (CSI00, CSI01, CSI10) are channels 0 to 2 of SAU and channel.

Channel	Used as CSI	Used as UART	Used as Simplified I^{2}C
0	CSI00	UART0 (supporting LIN-bus)	–
1	CSI01		–
2	CSI10	UART1	IIC10
3	–		–

3-wire serial I/O (CSI00, CSI01, CIS10) performs the following six types of communication operations.

- Master transmission (See **12.5.1**.)
- Master reception (See **12.5.2**.)
- Master transmission/reception (See **12.5.3**.)
- Slave transmission (See **12.5.4**.)
- Slave reception (See **12.5.5**.)
- Slave transmission/reception (See **12.5.6**.)

12.5.1 Master transmission

Master transmission is that the μPD79F9210/1 outputs a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10
Target channel	Channel 0 of SAU	Channel 1 of SAU	Channel 2 of SAU
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	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. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] [Note] f_{CLK}: System clock frequency		
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data output starts from the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 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 (TARGET)**).

Remark n: Channel number (n = 0 to 2)

(1) Register setting

Figure 12-23. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0						CKO02	CKO01	CKO00						SO02	SO01	SO00
	0	0	0	0	1	0/1	0/1	0/1	0	0	0	0	1	0/1	0/1	0/1

Communication starts when these bits are 1 if the data phase is forward (CKP0n = 0). If the phase is reversed (CKP0n = 1), communication starts when these bits are 0.

(b) Serial output enable register 0 (SOE0) … 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
SOE0														SOE02	SOE01	SOE00
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1

(c) Serial channel start register 0 (SS0) … 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
SS0													SS03	SS02	SS01	SS00
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	0/1	0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n	CCS0n						STS0n		SIS00				MD0n2	MD0n1	MD0n0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0	DIR0n		SLC0n1	SLC0n0		DLS0n2	DLS0n1	DLS0n0
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

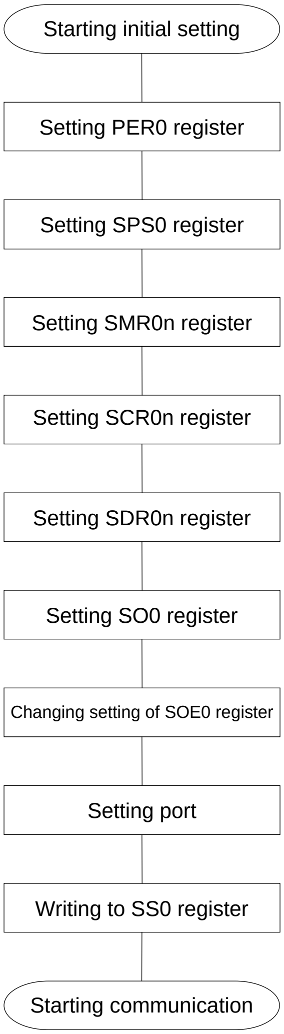
(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	Baud rate setting							0	Transmit data setting							
									SIOp							

Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)
□ : 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 12-24. Initial Setting Procedure for Master Transmission



Release the serial array unit from the reset status and start clock supply.

Set the prescaler.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate.

Manipulate the SO0n and CKO0n bits and set an initial output level.

Set the SOE0n bit to 1 and enable data output of the target channel.

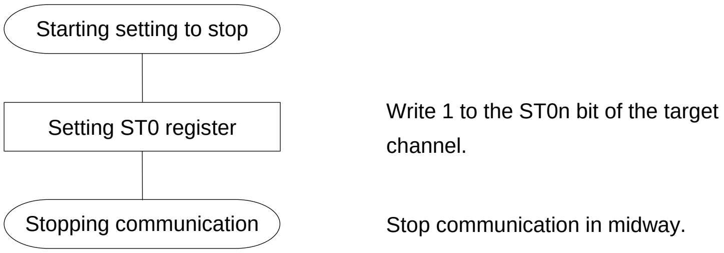
Enable data output and clock output of the target channel by setting a port register and a port mode register.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

Set transmit data to the SIOp register (bits 7 to 0 of the SDR0n register) and start communication.

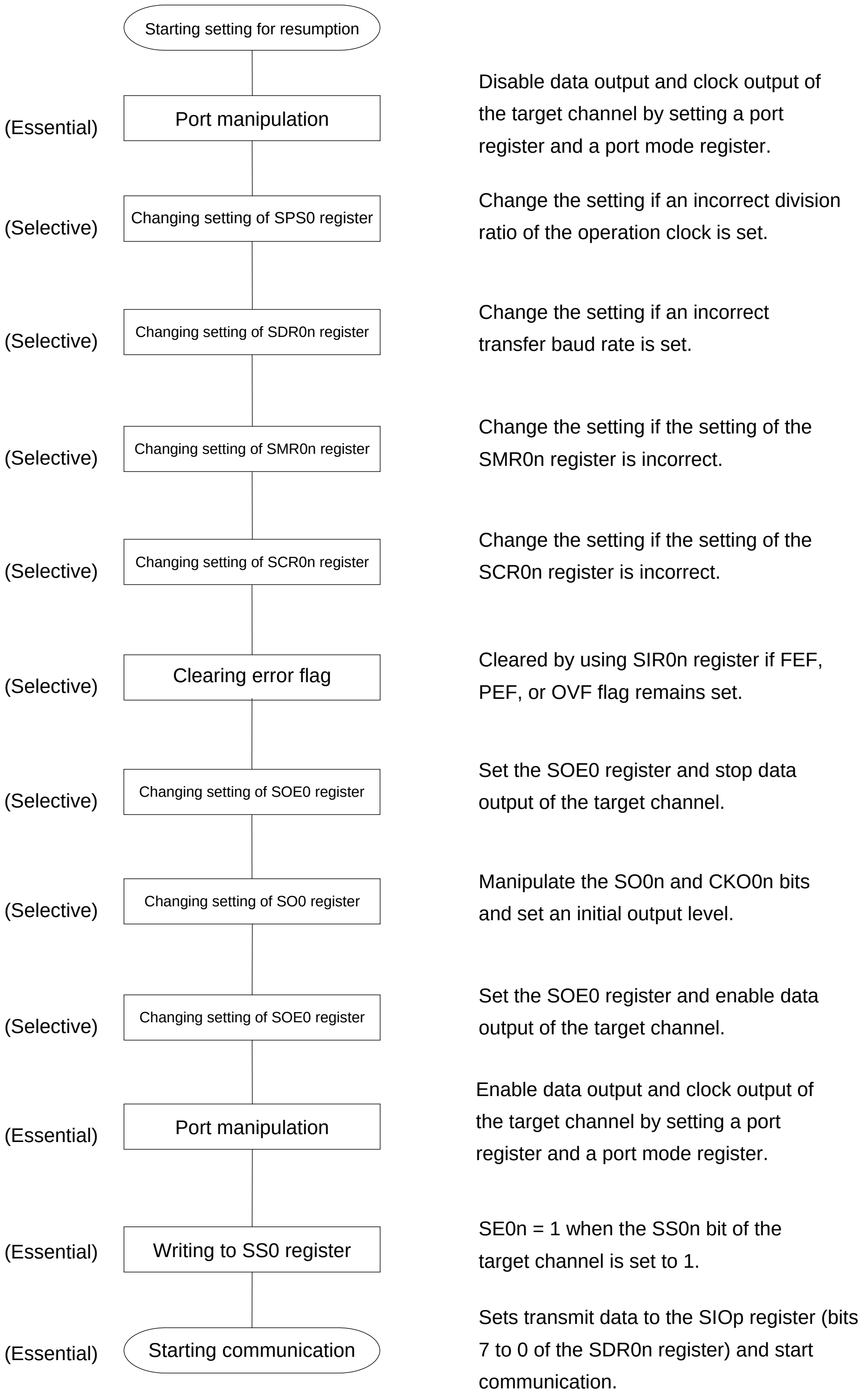
Cautions **After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.**

Figure 12-25. Procedure for Stopping Master Transmission


Starting setting to stop
Setting ST0 register
Write 1 to the ST0n bit of the target channel.
Stopping communication
Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 12-26 Procedure for Resuming Master Transmission**).

Figure 12-26. Procedure for Resuming Master Transmission


Starting setting for resumption
(Essential)
Port manipulation
(Selective)
Changing setting of SPS0 register
(Selective)
Changing setting of SDR0n register
(Selective)
Changing setting of SMR0n register
(Selective)
Changing setting of SCR0n register
(Selective)
Clearing error flag
(Selective)
Changing setting of SOE0 register
(Selective)
Changing setting of SO0 register
(Selective)
Changing setting of SOE0 register
(Essential)
Port manipulation
(Essential)
Writing to SS0 register
(Essential)
Starting communication

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect transfer baud rate is set.

Change the setting if the setting of the SMR0n register is incorrect.

Change the setting if the setting of the SCR0n register is incorrect.

Cleared by using SIR0n register if FEF, PEF, or OVF flag remains set.

Set the SOE0 register and stop data output of the target channel.

Manipulate the SO0n and CKO0n bits and set an initial output level.

Set the SOE0 register and enable data output of the target channel.

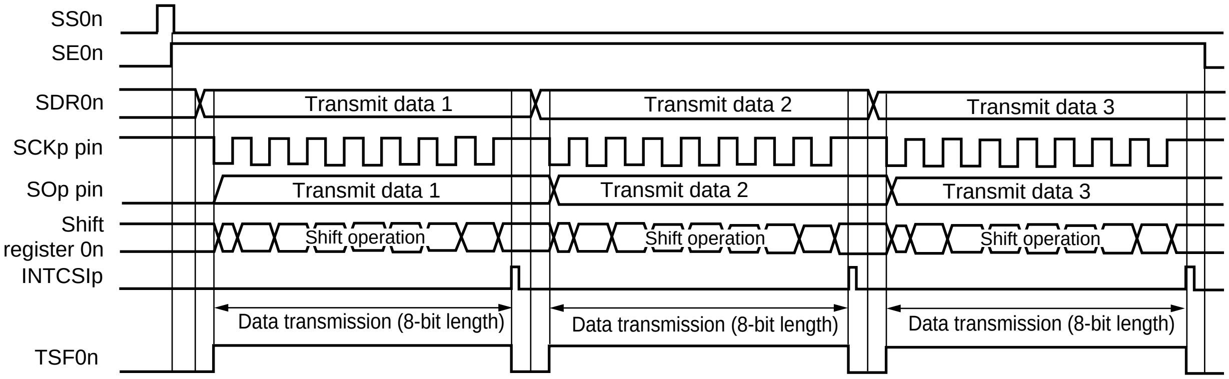
Enable data output and clock output of the target channel by setting a port register and a port mode register.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

Sets transmit data to the SIOp register (bits 7 to 0 of the SDR0n register) and start communication.

(3) Processing flow (in single-transmission mode)

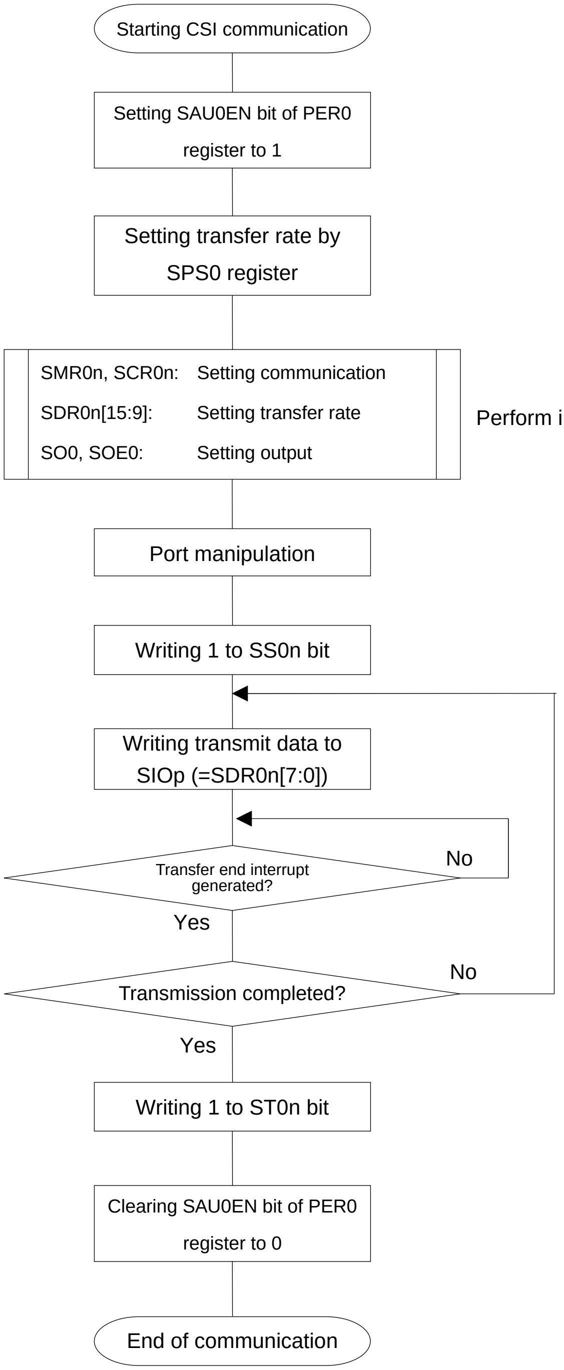
Figure 12-27. Timing Chart of Master Transmission (in Single-Transmission Mode)



Remark n: Channel number (n = 0 to 2)

p: CSI number (p = 00, 01, 10)

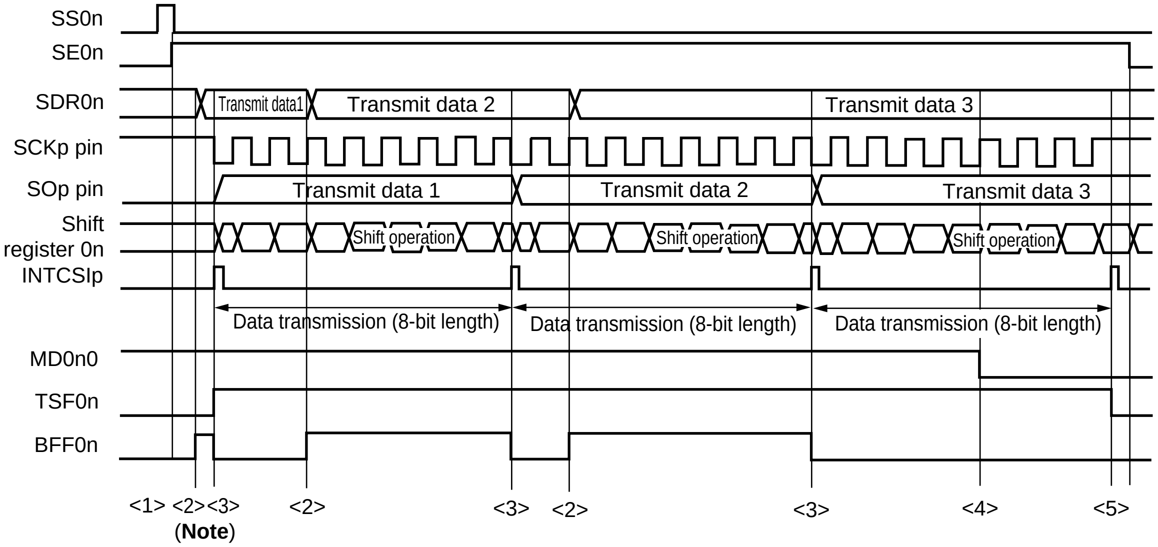
Figure 12-28. Flowchart of Master Transmission (in Single-Transmission Mode)


Starting CSI communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0, SOE0: Setting output
Perform initial setting when SE0n = 0.
Port manipulation
Writing 1 to SS0n bit
Writing transmit data to SIOp (=SDR0n[7:0])
Transfer end interrupt generated?
No
Yes
Transmission completed?
No
Yes
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 12-29. Timing Chart of Master Transmission (in Continuous Transmission Mode)

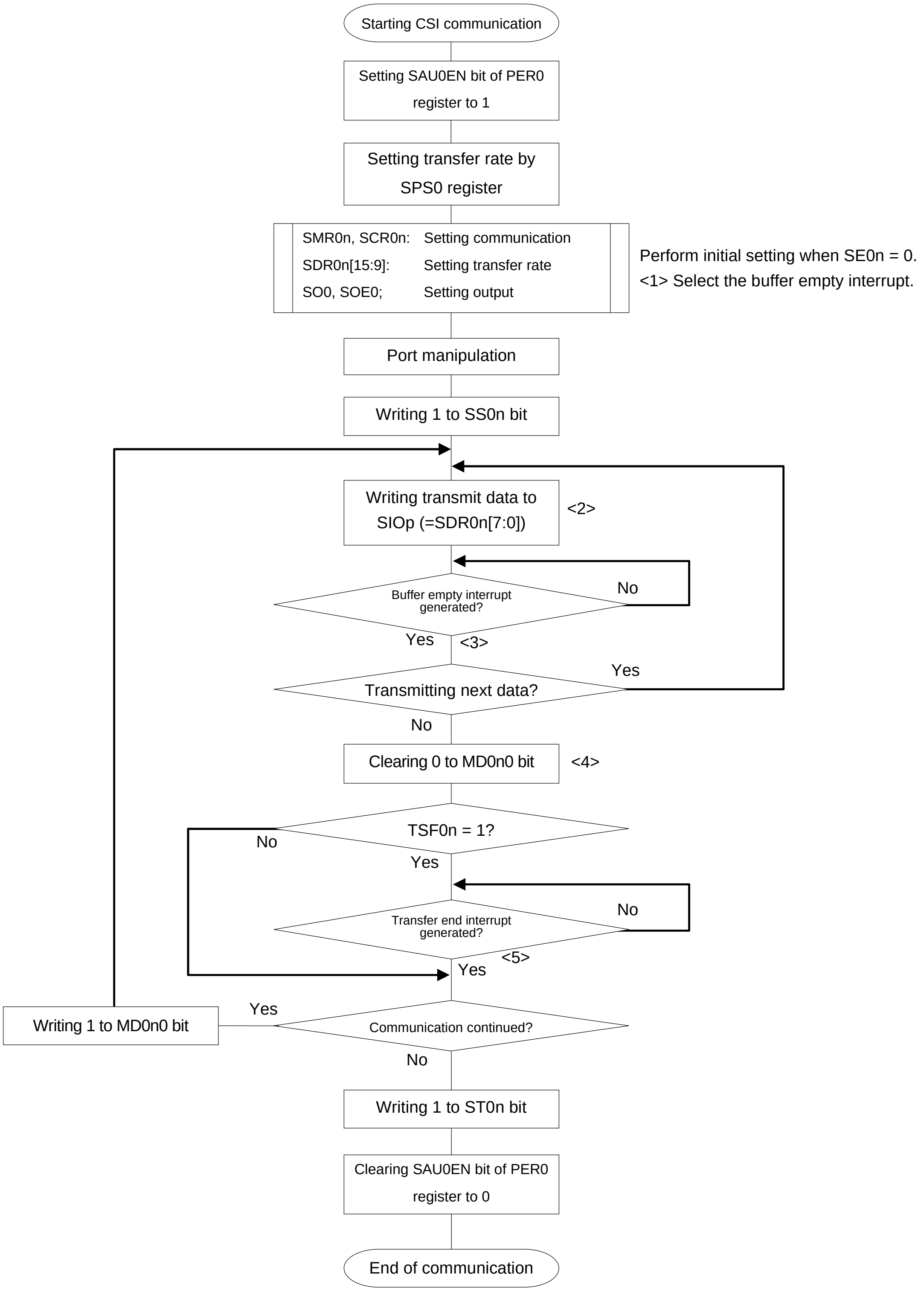


Note When transmit data is written to the SDR0n register while BFF0n = 1, the transmit data is overwritten.

Caution The MD0n0 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.

Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)

Figure 12-30. Flowchart of Master Transmission (in Continuous Transmission Mode)


Starting CSI communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0, SOE0; Setting output
Perform initial setting when SE0n = 0.
<1> Select the buffer empty interrupt.
Port manipulation
Writing 1 to SS0n bit
Writing transmit data to SIOp (=SDR0n[7:0])
<2>
Buffer empty interrupt generated?
No
Yes
<3>
Transmitting next data?
Yes
No
Clearing 0 to MD0n0 bit
<4>
TSF0n = 1?
No
Yes
Transfer end interrupt generated?
No
<5>
Yes
Writing 1 to MD0n0 bit
Yes
Communication continued?
No
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions **After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.**

Remark <1> to <5> in the figure correspond to <1> to <5> in **Figure 12-29 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

12.5.2 Master reception

Master reception is that the μPD79F9210/1 outputs a transfer clock and receives data from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI10
Target channel	Channel 0 of SAU	Channel 1 of SAU	Channel 2 of SAU
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error detection flag	Overrun error detection flag (OVF0n) only		
Transfer data length	7 or 8 bits		
Transfer rate	Max. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] Note fCLK: System clock frequency		
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data input starts from the start of the operation of the serial clock. • DAP0n = 1: Data input starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 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 (TARGET)**).

Remark n: Channel number (n = 0 to 2)

(1) Register setting

Figure 12-31. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0						CKO02	CKO01	CKO00						SO02	SO01	SO00
	0	0	0	0	1	0/1	0/1	0/1	0	0	0	0	1	×	×	×

Communication starts when these bits are 1 if the data phase is forward (CKP0n = 0). If the phase is reversed (CKP0n = 1), communication starts when these bits are 0.

(b) Serial output enable register 0 (SOE0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0														SOE02	SOE01	SOE00
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1

Set these bits to 0 when using them for CSI master reception

(c) Serial channel start register 0 (SS0) … 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
SS0													SS03	SS02	SS01	SS00
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	0/1	0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n	CCS0n						STS0n		SIS0n				MD0n2	MD0n1	MD0n0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Operation mode of channel n
0: Transfer end interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0	DIR0n		SLC0n1	SLC0n0		DLS0n2	DLS0n1	DLS0n0
	0	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

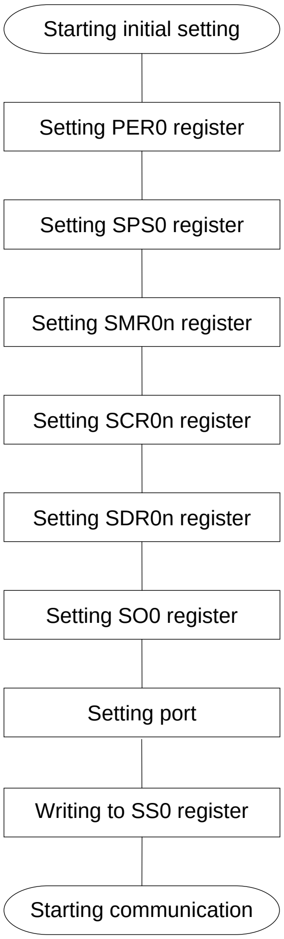
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	Baud rate setting							0	Receive data register (Write FFH as dummy data.)							

SIOp

Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)
□ : 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 12-32. Initial Setting Procedure for Master Reception



Release the serial array unit from the reset status and start clock supply.

Set the prescaler.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate.

Manipulate the CKO0n bit and set an initial output level.

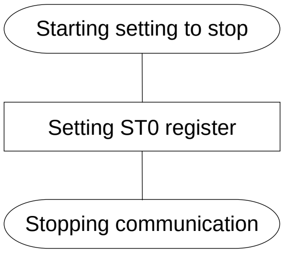
Enable clock output of the target channel by setting a port register and a port mode register.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

Set dummy data to the SIOp register (bits 7 to 0 of the SDR0n register) and start communication.

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 12-33. Procedure for Stopping Master Reception

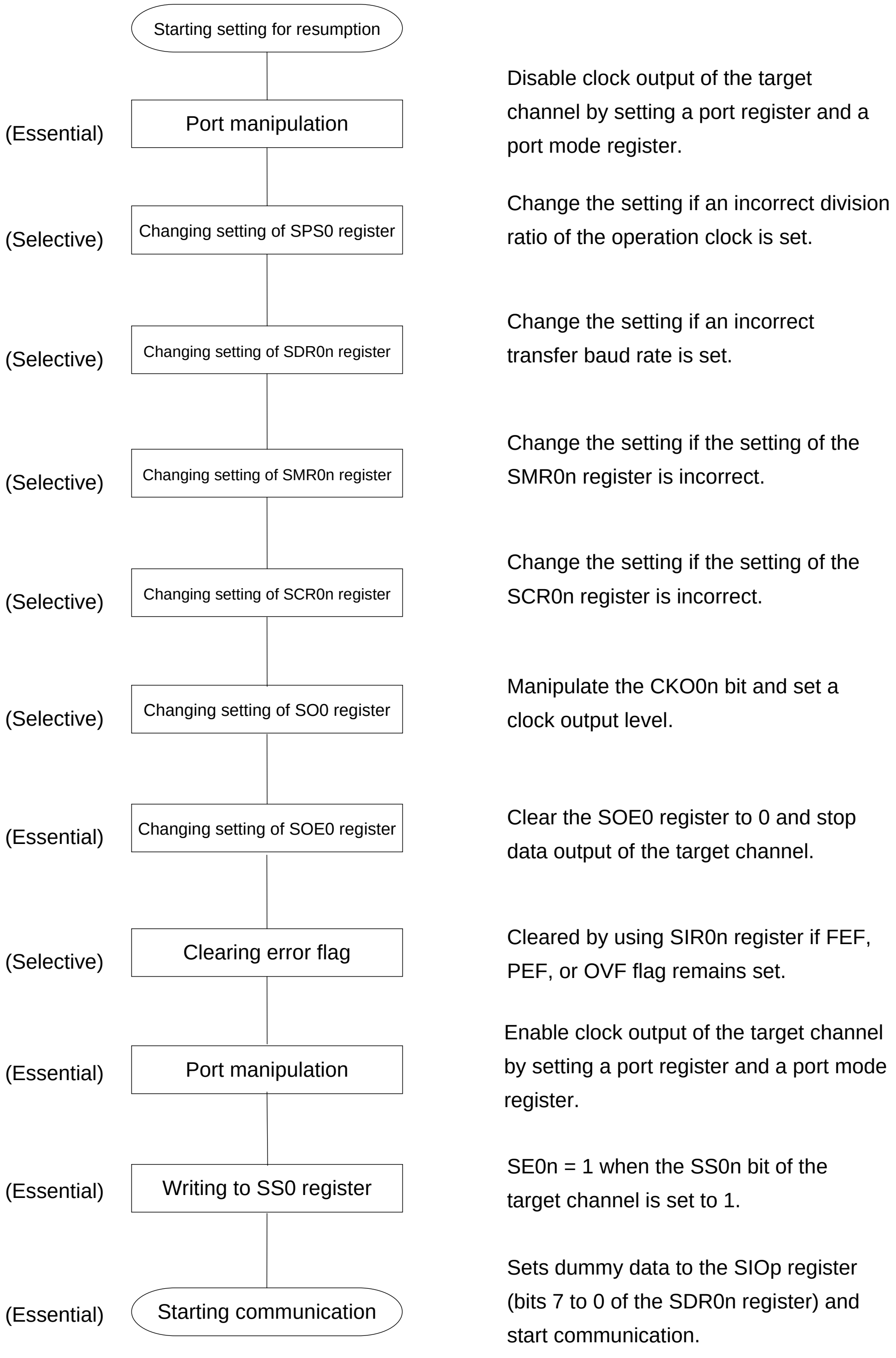


Write 1 to the ST0n bit of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 12-34 Procedure for Resuming Master Reception**).

Figure 12-34. Procedure for Resuming Master Reception


Starting setting for resumption
(Essential)
Port manipulation
(Selective)
Changing setting of SPS0 register
(Selective)
Changing setting of SDR0n register
(Selective)
Changing setting of SMR0n register
(Selective)
Changing setting of SCR0n register
(Selective)
Changing setting of SO0 register
(Essential)
Changing setting of SOE0 register
(Selective)
Clearing error flag
(Essential)
Port manipulation
(Essential)
Writing to SS0 register
(Essential)
Starting communication

Disable clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect transfer baud rate is set.

Change the setting if the setting of the SMR0n register is incorrect.

Change the setting if the setting of the SCR0n register is incorrect.

Manipulate the CKO0n bit and set a clock output level.

Clear the SOE0 register to 0 and stop data output of the target channel.

Cleared by using SIR0n register if FEF, PEF, or OVF flag remains set.

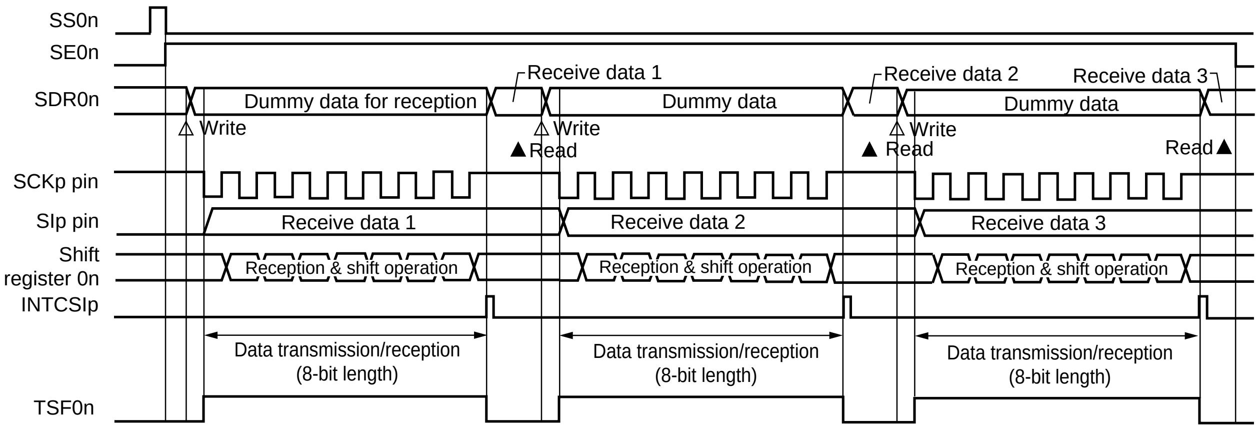
Enable clock output of the target channel by setting a port register and a port mode register.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

Sets dummy data to the SIOp register (bits 7 to 0 of the SDR0n register) and start communication.

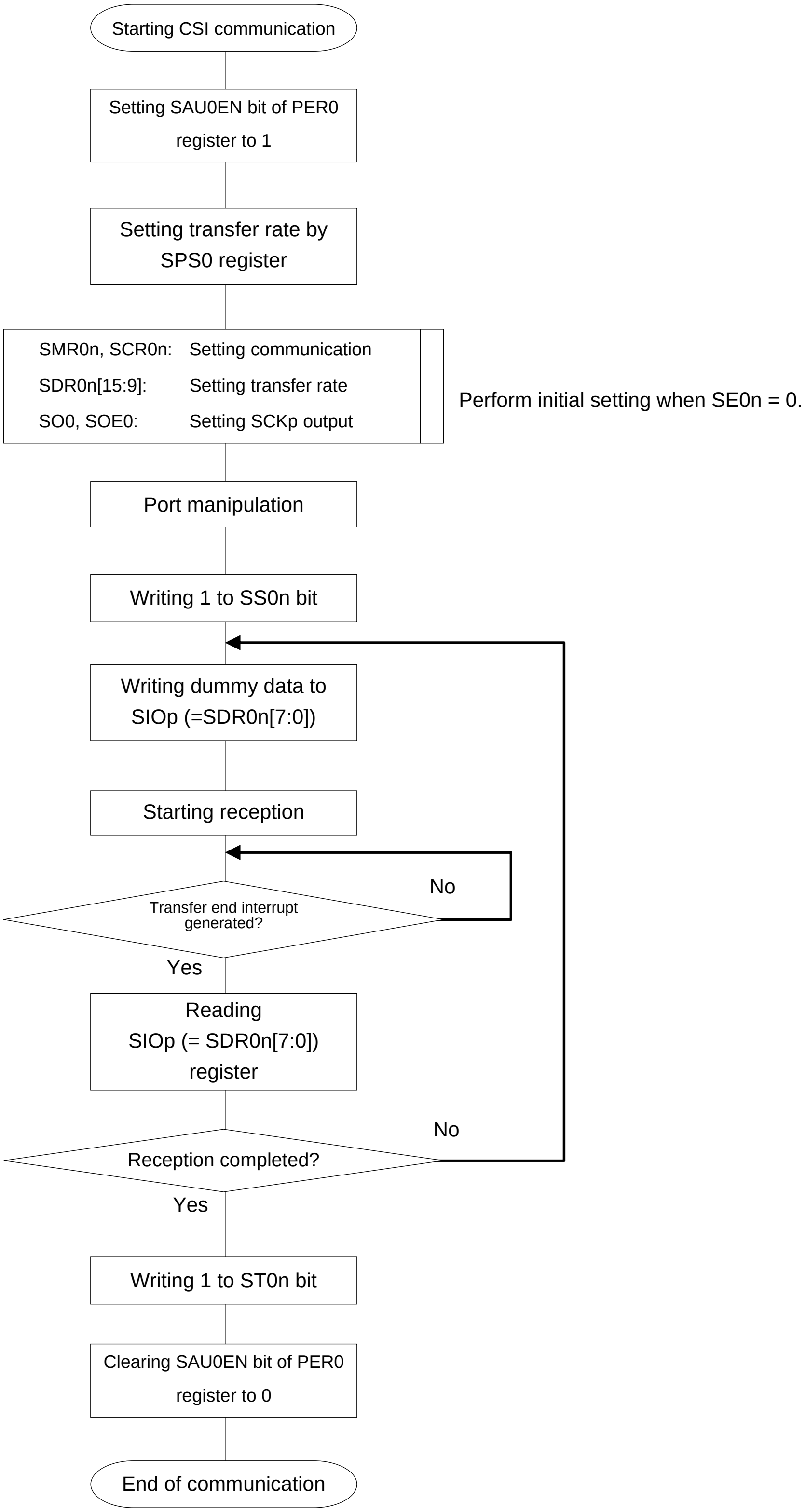
(3) Processing flow (in single-reception mode)

Figure 12-35. Timing Chart of Master Reception (in Single-Reception Mode)



Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)

Figure 12-36. Flowchart of Master Reception (in Single-Reception Mode)


Starting CSI communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0, SOE0: Setting SCKp output
Perform initial setting when SE0n = 0.
Port manipulation
Writing 1 to SS0n bit
Writing dummy data to SIOp (=SDR0n[7:0])
Starting reception
Transfer end interrupt generated?
No
Yes
Reading SIOp (= SDR0n[7:0]) register
Reception completed?
No
Yes
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions **After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.**

12.5.3 Master transmission/reception

Master transmission/reception is that the μPD79F9210/1 outputs a transfer clock and transmits/receives data to/from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI10
Target channel	Channel 0 of SAU	Channel 1 of SAU	Channel 2 of SAU
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	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 (OVF0n) only		
Transfer data length	7 or 8 bits		
Transfer rate	Max. f_{CLK}/4 [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] Note f_{CLK}: System clock frequency		
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data I/O starts at the start of the operation of the serial clock. • DAP0n = 1: Data I/O starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 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 (TARGET)**).

Remark n: Channel number (n = 0 to 2)

(1) Register setting

Figure 12-37. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0						CKO02	CKO01	CKO00						SO02	SO01	SO00
	0	0	0	0	1	0/1	0/1	0/1	0	0	0	0	1	0/1	0/1	0/1

Communication starts when these bits are 1 if the data phase is forward (CKP0n = 0). If the phase is reversed (CKP0n = 1), communication starts when these bits are 0.

(b) Serial output enable register 0 (SOE0) … 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
SOE0														SOE02	SOE01	SOE00
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1

(c) Serial channel start register 0 (SS0) … 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
SS0													SS03	SS02	SS01	SS00
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	0/1	0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n	CCS0n						STS0n		SIS0n0				MD0n2	MD0n1	MD0n0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0	DIR0n		SLC0n1	SLC0n0		DLS0n2	DLS0n1	DLS0n0
	1	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

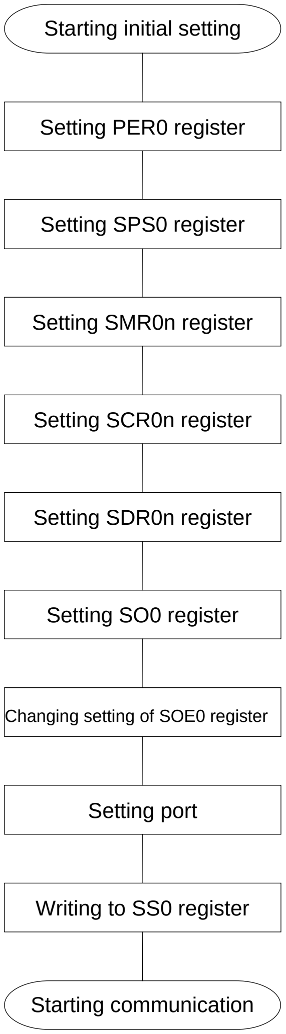
(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	Baud rate setting							0	Transmit data setting/receive data register							
									SIOp							

Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)
□ : 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 12-38. Initial Setting Procedure for Master Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the prescaler.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate.

Manipulate the SO0n and CKO0n bits and set an initial output level.

Set the SOE0n bit to 1 and enable data output of the target channel.

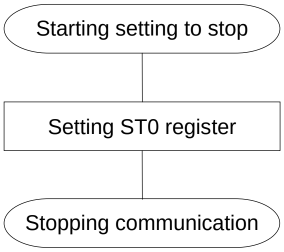
Enable data output and clock output of the target channel by setting a port register and a port mode register.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

Set transmit data to the SIOp register (bits 7 to 0 of the SDR0n register) and start communication.

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 12-39. Procedure for Stopping Master Transmission/Reception

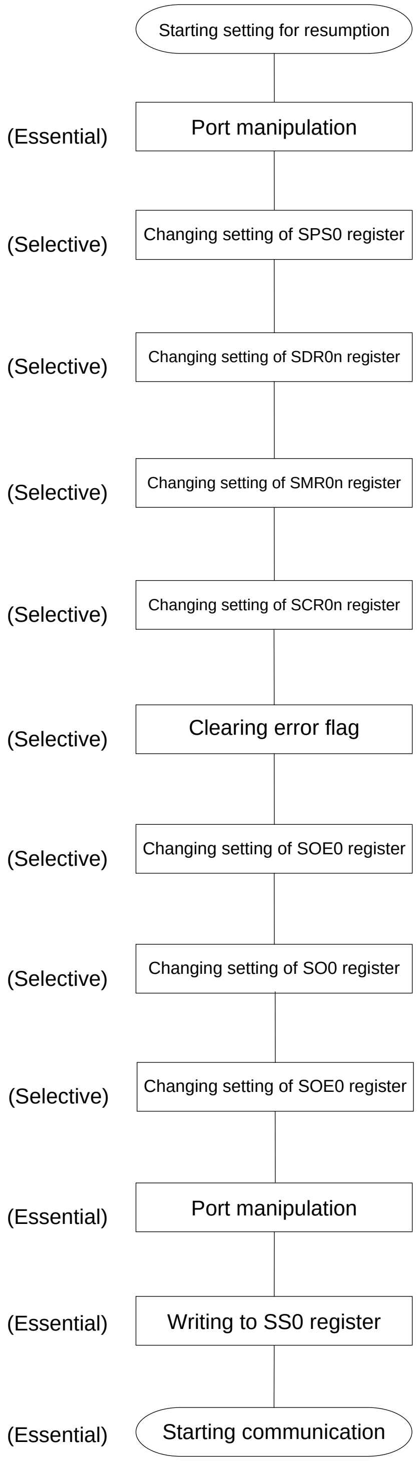


Write 1 to the ST0n bit of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 12-40 Procedure for Resuming Master Transmission/Reception**).

Figure 12-40. Procedure for Resuming Master Transmission/Reception


Starting setting for resumption
(Essential)
Port manipulation
(Selective)
Changing setting of SPS0 register
(Selective)
Changing setting of SDR0n register
(Selective)
Changing setting of SMR0n register
(Selective)
Changing setting of SCR0n register
(Selective)
Clearing error flag
(Selective)
Changing setting of SOE0 register
(Selective)
Changing setting of SO0 register
(Selective)
Changing setting of SOE0 register
(Essential)
Port manipulation
(Essential)
Writing to SS0 register
(Essential)
Starting communication

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect transfer baud rate is set.

Change the setting if the setting of the SMR0n register is incorrect.

Change the setting if the setting of the SCR0n register is incorrect.

Cleared by using SIR0n register if FEF, PEF, or OVF flag remains set.

Set the SOE0 register and stop data output of the target channel.

Manipulate the SO0n and CKO0n bits and set an initial output level.

Set the SOE0 register and enable data output of the target channel.

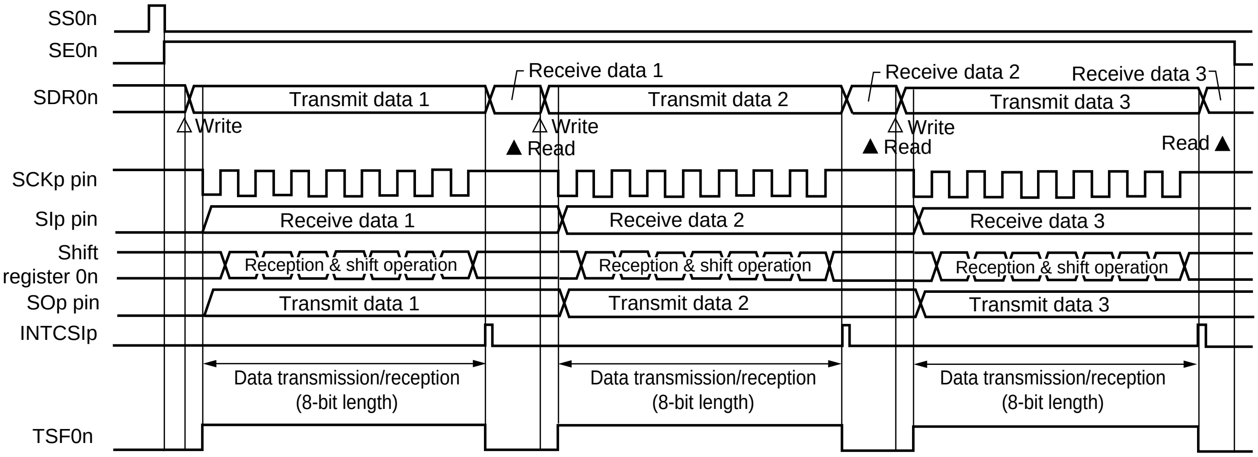
Enable data output and clock output of the target channel by setting a port register and a port mode register.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

Sets transmit data to the SIOp register (bits 7 to 0 of the SDR0n register) and start communication.

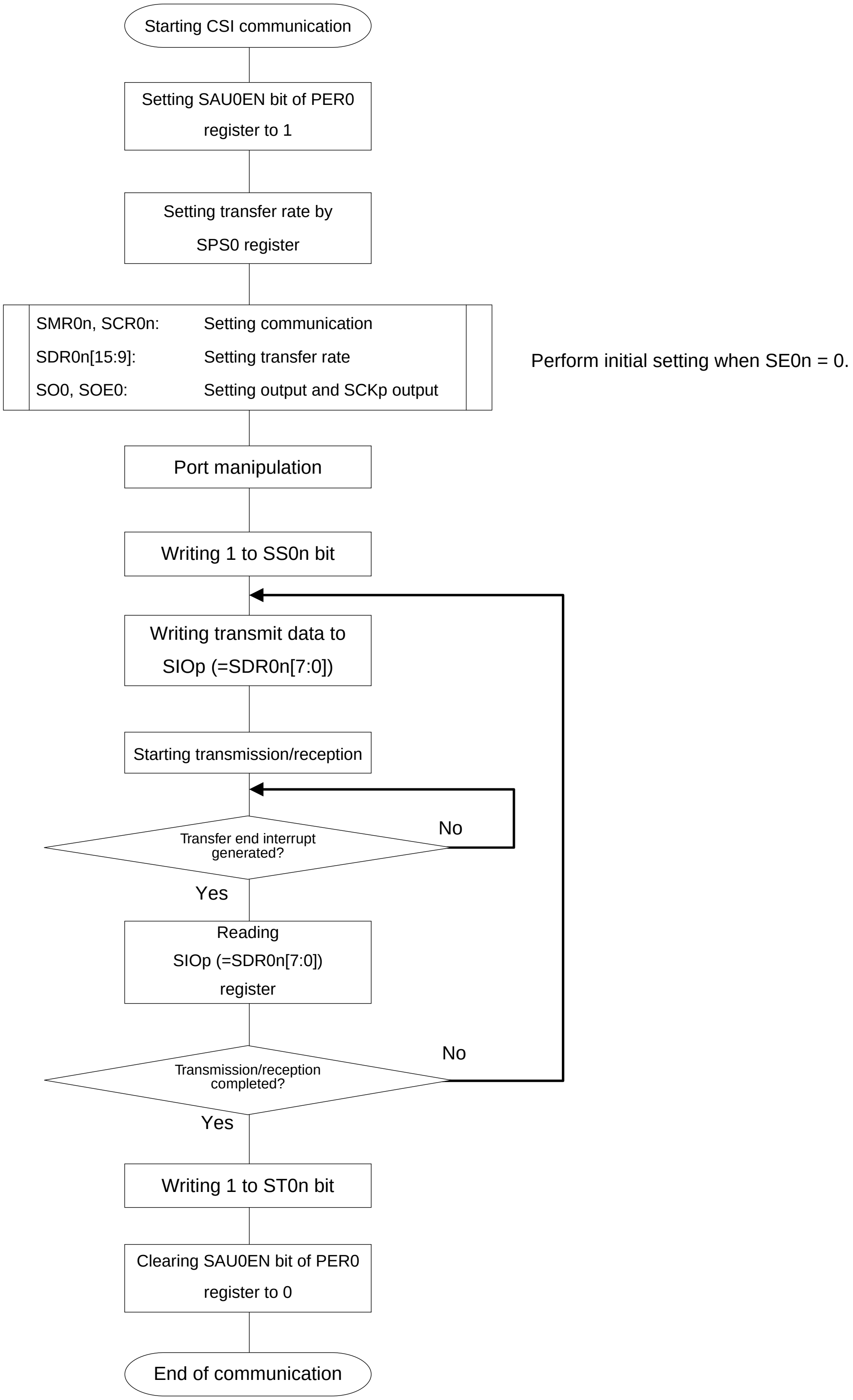
(3) Processing flow (in single-transmission/reception mode)

Figure 12-41. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode)



Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)

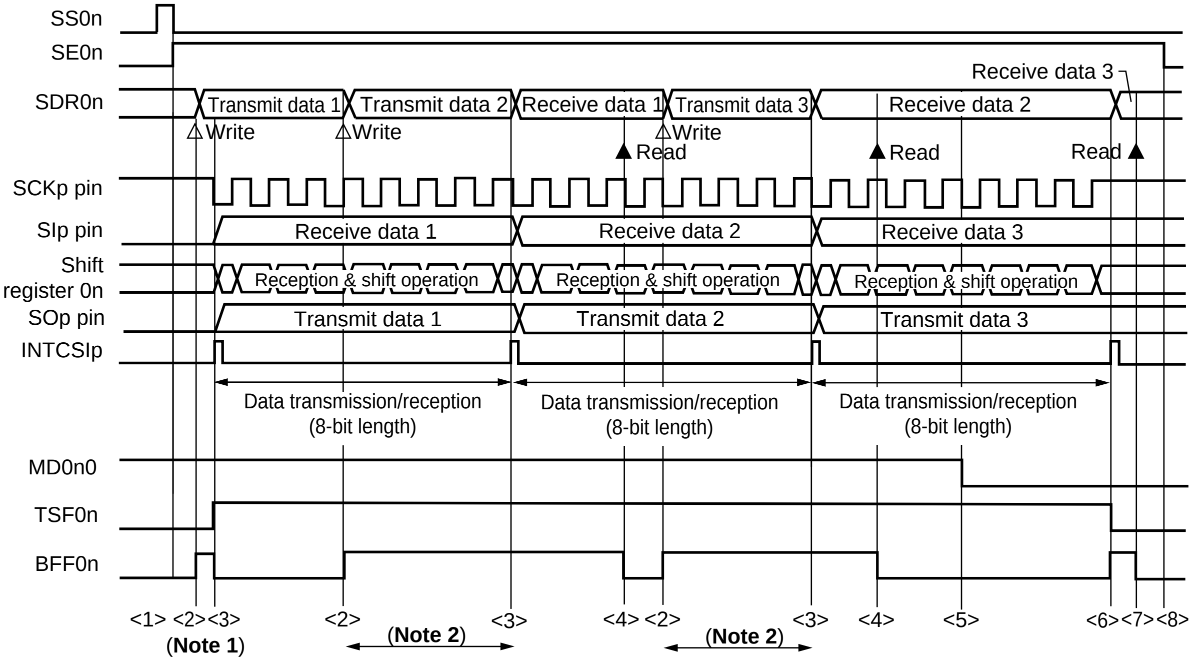
Figure 12-42. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)


Starting CSI communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0, SOE0: Setting output and SCKp output
Perform initial setting when SE0n = 0.
Port manipulation
Writing 1 to SS0n bit
Writing transmit data to SIOp (=SDR0n[7:0])
Starting transmission/reception
Transfer end interrupt generated?
No
Yes
Reading SIOp (=SDR0n[7:0]) register
Transmission/reception completed?
No
Yes
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions **After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.**

(4) Processing flow (in continuous transmission/reception mode)

Figure 12-43. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Notes 1. When transmit data is written to the SDR0n register while BFF0n = 1, the transmit data is overwritten.

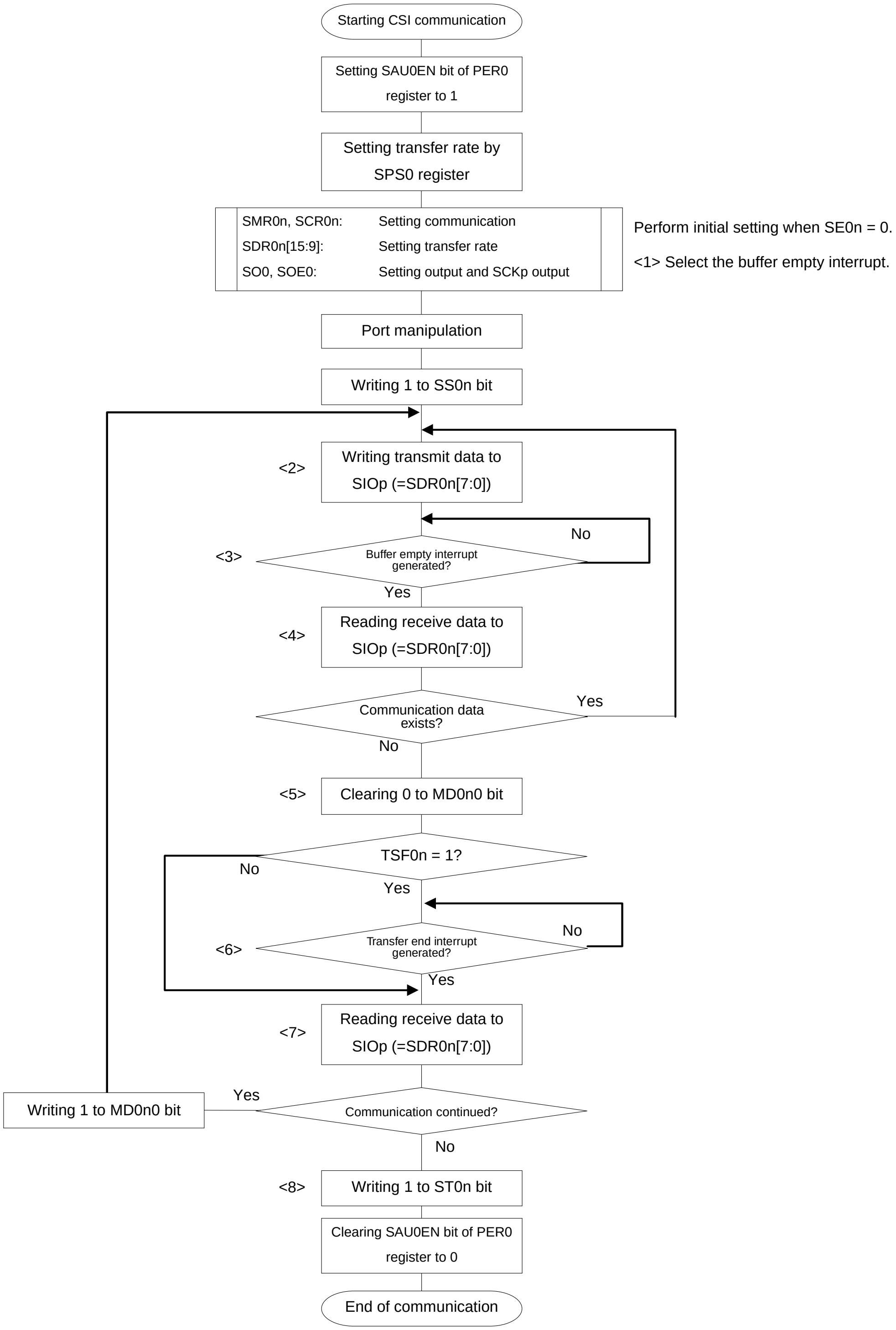
2. The transmit data can be read by reading the SDR0n register during this period. At this time, the transfer operation is not affected.

Caution The MD0n0 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.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-44 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode**).

2. n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)

Figure 12-44. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)


Starting CSI communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0, SOE0: Setting output and SCKp output
Perform initial setting when SE0n = 0.
<1> Select the buffer empty interrupt.
Port manipulation
Writing 1 to SS0n bit
<2>
Writing transmit data to SIOp (=SDR0n[7:0])
<3>
Buffer empty interrupt generated?
No
Yes
<4>
Reading receive data to SIOp (=SDR0n[7:0])
Communication data exists?
Yes
No
<5>
Clearing 0 to MD0n0 bit
TSF0n = 1?
No
Yes
<6>
Transfer end interrupt generated?
No
Yes
<7>
Reading receive data to SIOp (=SDR0n[7:0])
Writing 1 to MD0n0 bit
Yes
Communication continued?
No
<8>
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-43 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

12.5.4 Slave transmission

Slave transmission is that the μPD79F9210/1 transmits data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10
Target channel	Channel 0 of SAU	Channel 1 of SAU	Channel 2 of SAU
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	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 (OVF0n) only		
Transfer data length	7 or 8 bits		
Transfer rate	The smaller of $f_{CLK}/6$ [MHz] and $f_{MCK}/2$ [MHz] is the maximum transfer rate Notes1,2.		
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data output starts from the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 1: Reverse		
Data direction	MSB or LSB first		

Notes 1. Because the external serial clock input to pins SCK00, SCK01, and SCK10 is sampled internally and used, the fastest baud rate is the smaller of $f_{CLK}/6$ [MHz] and $f_{MCK}/2$ [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 (TARGET)**).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 0 to 2)

(1) Register setting

Figure 12-45. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0						CKO02	CKO01	CKO00						SO02	SO01	SO00
	0	0	0	0	1	×	×	×	0	0	0	0	1	0/1	0/1	0/1

(b) Serial output enable register 0 (SOE0) … 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
SOE0														SOE02	SOE01	SOE00
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1

(c) Serial channel start register 0 (SS0) … 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
SS0													SS03	SS02	SS01	SS00
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	0/1	0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n	CCS0n						STS0n		SIS0n0				MD0n2	MD0n1	MD0n0
	0/1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0	DIR0n		SLC0n1	SLC0n0		DLS0n2	DLS0n1	DLS0n0
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

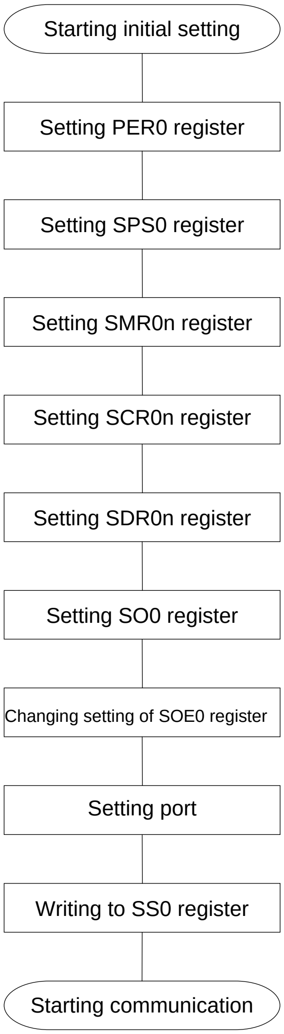
(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	Baud rate setting							0	Transmit data setting							
									SIOp							

Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)
□ : 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 12-46. Initial Setting Procedure for Slave Transmission



Release the serial array unit from the reset status and start clock supply.

Set the prescaler.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Manipulate the SO0n bit and set an initial output level.

Set the SOE0n bit to 1 and enable data output of the target channel.

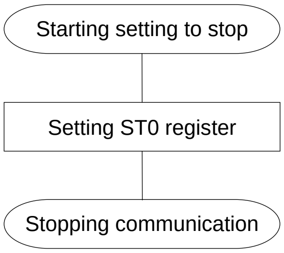
Enable data output of the target channel by setting a port register and a port mode register.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

Set transmit data to the SIOp register (bits 7 to 0 of the SDR0n register) and wait for a clock from the master.

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 12-47. Procedure for Stopping Slave Transmission

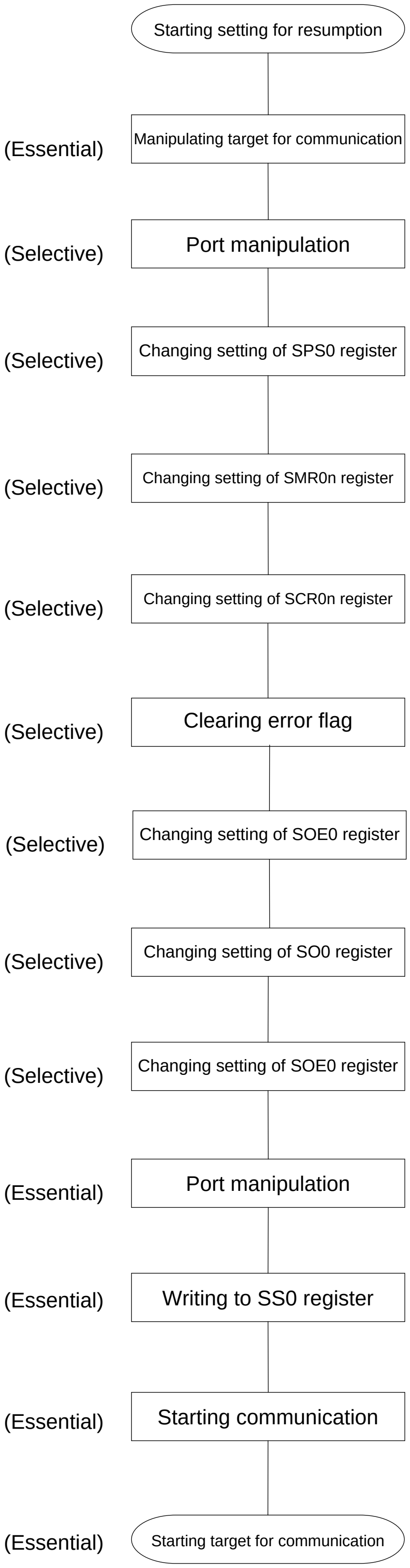


Write 1 to the ST0n bit of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 12-48 Procedure for Resuming Slave Transmission**).

Figure 12-48. Procedure for Resuming Slave Transmission


Starting setting for resumption
(Essential)
Manipulating target for communication
(Selective)
Port manipulation
(Selective)
Changing setting of SPS0 register
(Selective)
Changing setting of SMR0n register
(Selective)
Changing setting of SCR0n register
(Selective)
Clearing error flag
(Selective)
Changing setting of SOE0 register
(Selective)
Changing setting of SO0 register
(Selective)
Changing setting of SOE0 register
(Essential)
Port manipulation
(Essential)
Writing to SS0 register
(Essential)
Starting communication
(Essential)
Starting target for communication

Stop the target for communication or wait until the target completes its operation.

Disable data output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if the setting of the SMR0n register is incorrect.

Change the setting if the setting of the SCR0n register is incorrect.

Cleared by using SIR0n register if FEF, PEF, or OVF flag remains set.

Set the SOE0 register and stop data output of the target channel.

Manipulate the SO0n and CKO0n bits and set an initial output level.

Set the SOE0 register 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.

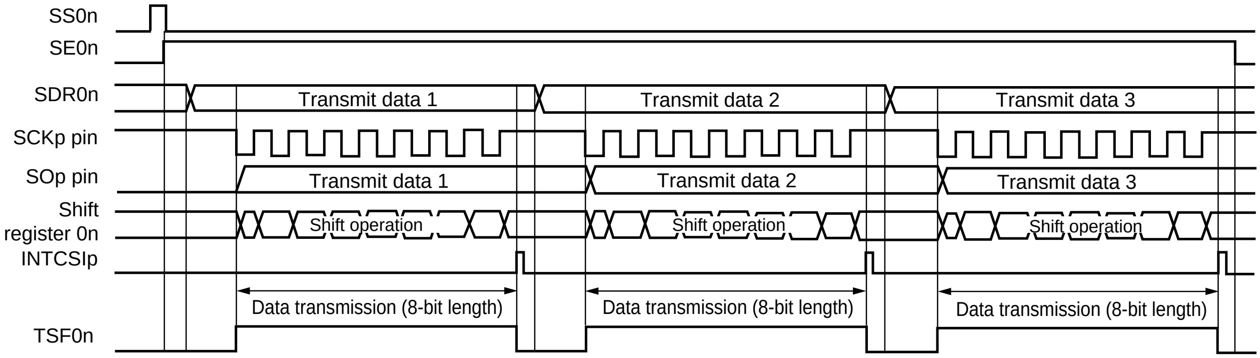
Set the SS0n bit of the target channel to 1 and set SE0n to 1.

Sets transmit data to the SIOp register (bits 7 to 0 of the SDR0n register) and wait for a clock from the master.

Starts the target for communication.

(3) Processing flow (in single-transmission mode)

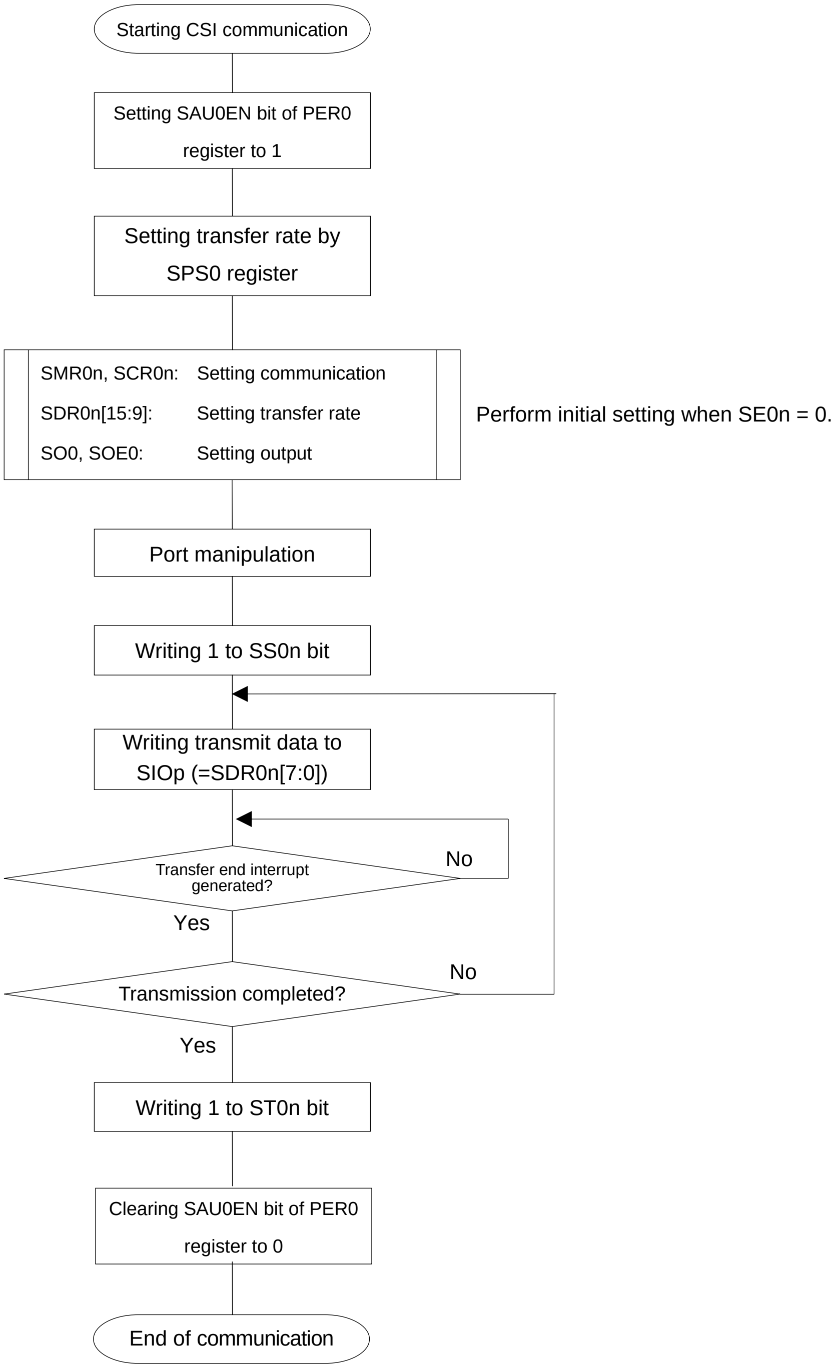
Figure 12-49. Timing Chart of Slave Transmission (in Single-Transmission Mode)



Remark n: Channel number (n = 0 to 2)

p: CSI number (p = 00, 01, 10)

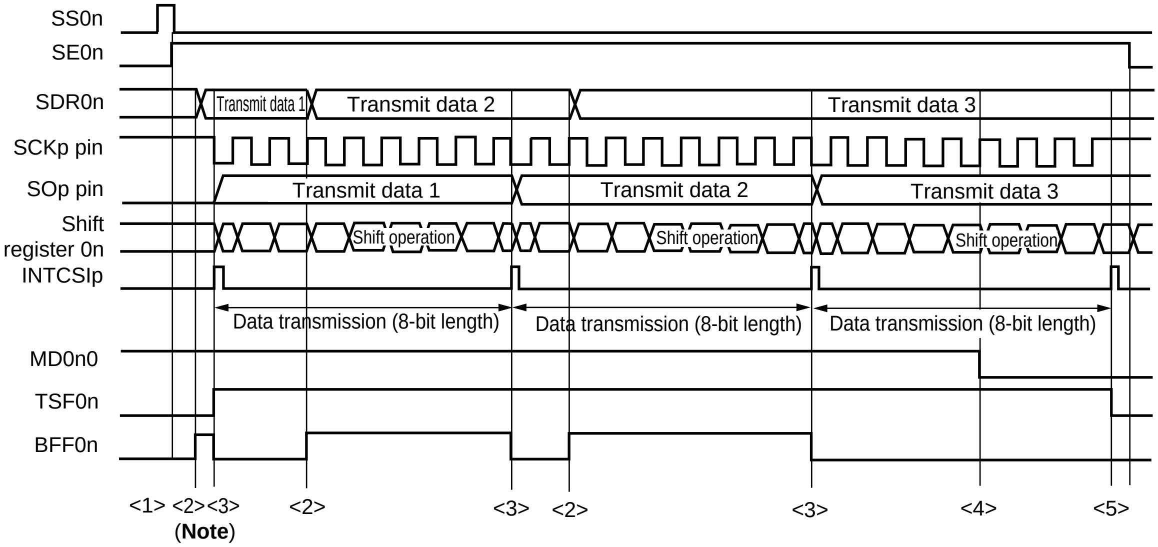
Figure 12-50. Flowchart of Slave Transmission (in Single-Transmission Mode)


Starting CSI communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0, SOE0: Setting output
Perform initial setting when SE0n = 0.
Port manipulation
Writing 1 to SS0n bit
Writing transmit data to SIOp (=SDR0n[7:0])
Transfer end interrupt generated?
No
Yes
Transmission completed?
No
Yes
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 12-51. Timing Chart of Slave Transmission (in Continuous Transmission Mode)

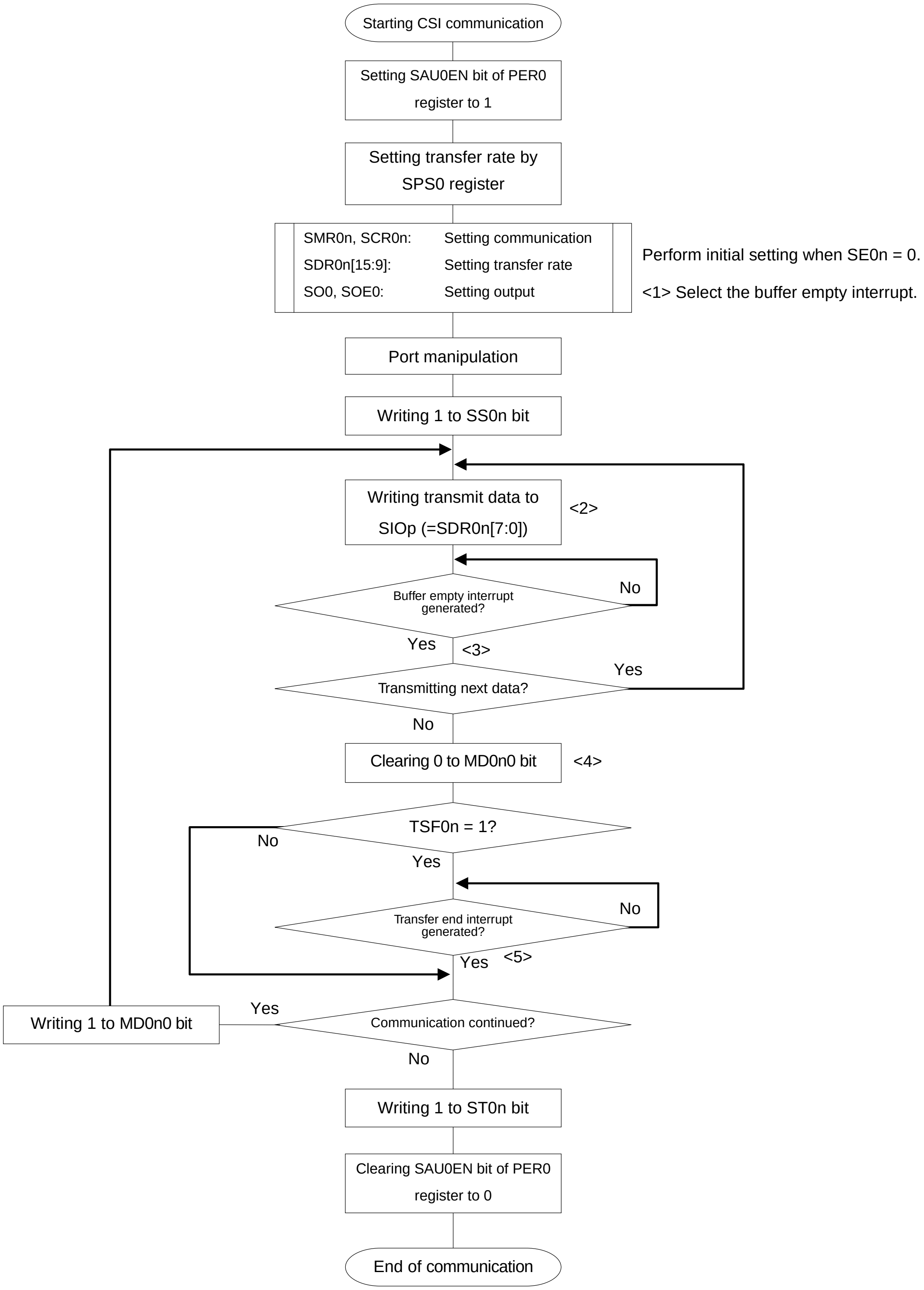


Note When transmit data is written to the SDR0n register while BFF0n = 1, the transmit data is overwritten.

Caution The MD0n0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)

Figure 12-52. Flowchart of Slave Transmission (in Continuous Transmission Mode)


Starting CSI communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0, SOE0: Setting output
Perform initial setting when SE0n = 0.
<1> Select the buffer empty interrupt.
Port manipulation
Writing 1 to SS0n bit
Writing transmit data to SIOp (=SDR0n[7:0])
<2>
Buffer empty interrupt generated?
No
Yes
<3>
Transmitting next data?
Yes
No
Clearing 0 to MD0n0 bit
<4>
TSF0n = 1?
No
Yes
Transfer end interrupt generated?
No
Yes
<5>
Communication continued?
Yes
Writing 1 to MD0n0 bit
No
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions **After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.**

Remark <1> to <5> in the figure correspond to <1> to <5> in **Figure 12-51 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

12.5.5 Slave reception

Slave reception is that the μPD79F9210/1 receives data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10
Target channel	Channel 0 of SAU	Channel 1 of SAU	Channel 2 of SAU
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error detection flag	Overrun error detection flag (OVF0n) only		
Transfer data length	7 or 8 bits		
Transfer rate	The smaller of f_{CLK}/6 [MHz] and f_{MCK}/2 [MHz] is the maximum transfer rate [Notes 1, 2].		
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data input starts from the start of the operation of the serial clock. • DAP0n = 1: Data input starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 1: Reverse		
Data direction	MSB or LSB first		

Notes 1. Because the external serial clock input to pins SCK00, SCK01, and SCK10 is sampled internally and used, the fastest baud rate is the smaller of f_{CLK}/6 [MHz] and f_{MCK}/2 [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 (TARGET)**).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 0 to 2)

(1) Register setting

Figure 12-53. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10)

(a) Serial output register 0 (SO0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0						CKO02	CKO01	CKO00						SO02	SO01	SO00
	0	0	0	0	1	×	×	×	0	0	0	0	1	×	×	×

(b) Serial output enable register 0 (SOE0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0														SOE02	SOE01	SOE00
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1

Set these bits to 0 when using them for CSI slave reception

(c) Serial channel start register 0 (SS0) … 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
SS0													SS03	SS02	SS01	SS00
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	0/1	0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n	CCS0n						STS0n		SIS0n0				MD0n2	MD0n1	MD0n0
	0/1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Operation mode of channel n
0: Transfer end interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0	DIR0n		SLC0n1	SLC0n0		DLS0n2	DLS0n1	DLS0n0
	0	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

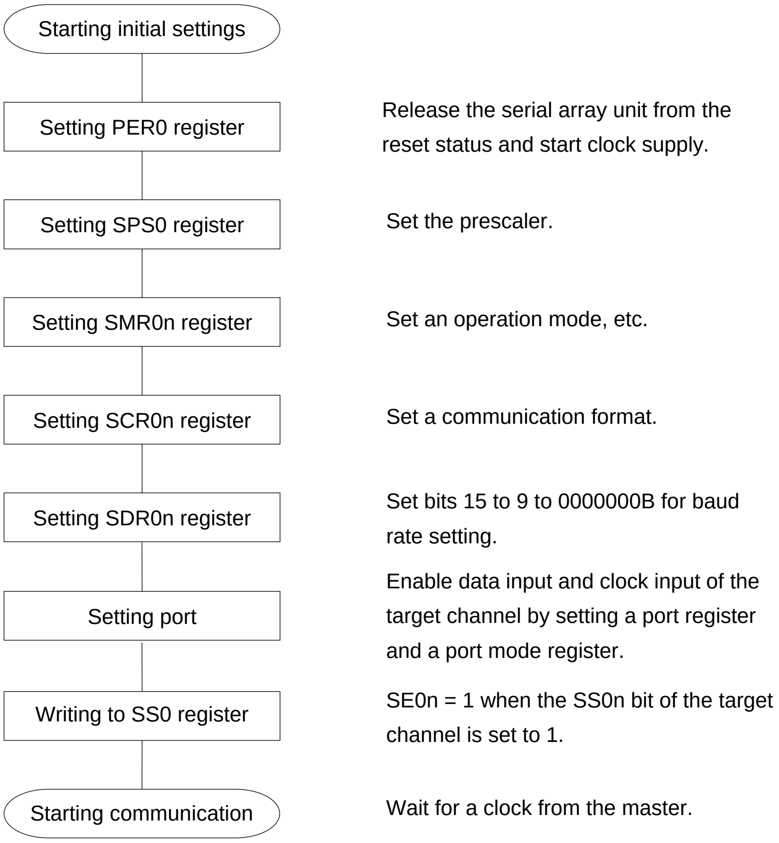
(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	0000000 (baud rate setting)							0	Receive data register							
									SIOp							

Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)
□ : 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 12-54. Initial Setting Procedure for Slave Reception


Starting initial settings
Setting PER0 register
Release the serial array unit from the reset status and start clock supply.
Setting SPS0 register
Set the prescaler.
Setting SMR0n register
Set an operation mode, etc.
Setting SCR0n register
Set a communication format.
Setting SDR0n register
Set bits 15 to 9 to 0000000B for baud rate setting.
Setting port
Enable data input and clock input of the target channel by setting a port register and a port mode register.
Writing to SS0 register
SE0n = 1 when the SS0n bit of the target channel is set to 1.
Starting communication
Wait for a clock from the master.

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 12-55. Procedure for Stopping Slave Reception

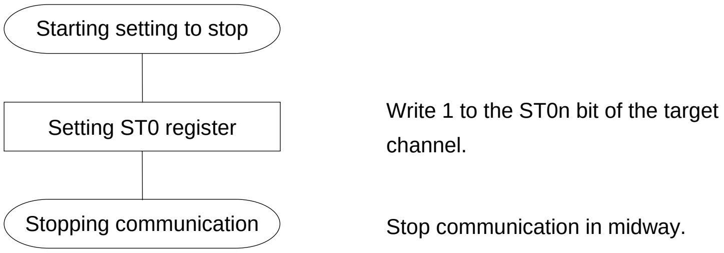
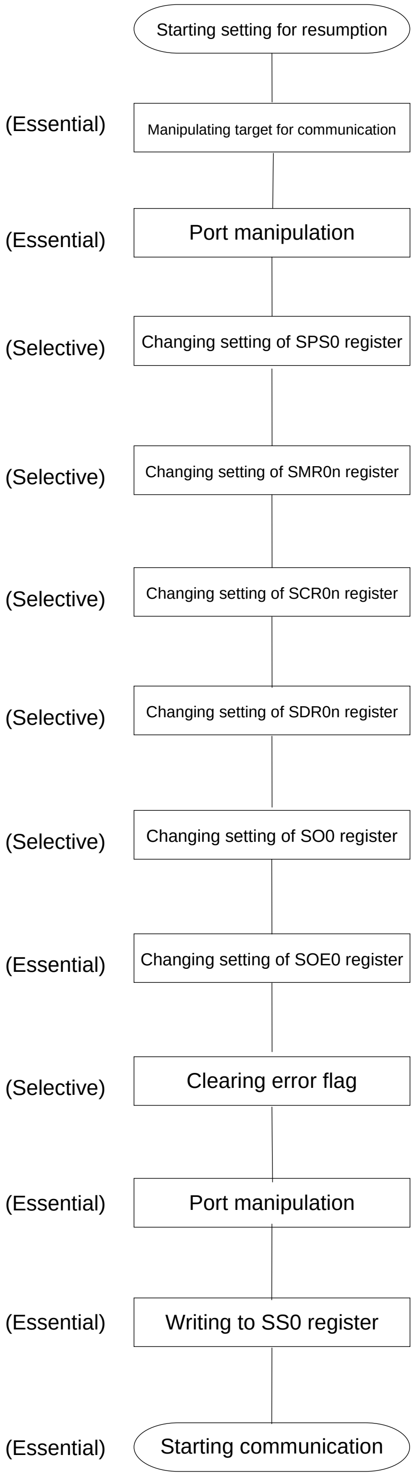

Starting setting to stop
Setting ST0 register
Write 1 to the ST0n bit of the target channel.
Stopping communication
Stop communication in midway.

Figure 12-56. Procedure for Resuming Slave Reception


Starting setting for resumption
(Essential)
Manipulating target for communication
(Essential)
Port manipulation
(Selective)
Changing setting of SPS0 register
(Selective)
Changing setting of SMR0n register
(Selective)
Changing setting of SCR0n register
(Selective)
Changing setting of SDR0n register
(Selective)
Changing setting of SO0 register
(Essential)
Changing setting of SOE0 register
(Selective)
Clearing error flag
(Essential)
Port manipulation
(Essential)
Writing to SS0 register
(Essential)
Starting communication

Stop the target for communication or wait until the target completes its operation.

Disable clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if the setting of the SMR0n register is incorrect.

Change the setting if the setting of the SCR0n register is incorrect.

Change the setting if the setting of the SDR0n register is incorrect.

Manipulate the CKO0n bit and enable reception.

Clear the SOE0 register to 0 and stop data output of the target channel.

Cleared by using SIR0n register if FEF, PEF, or OVF flag remains set.

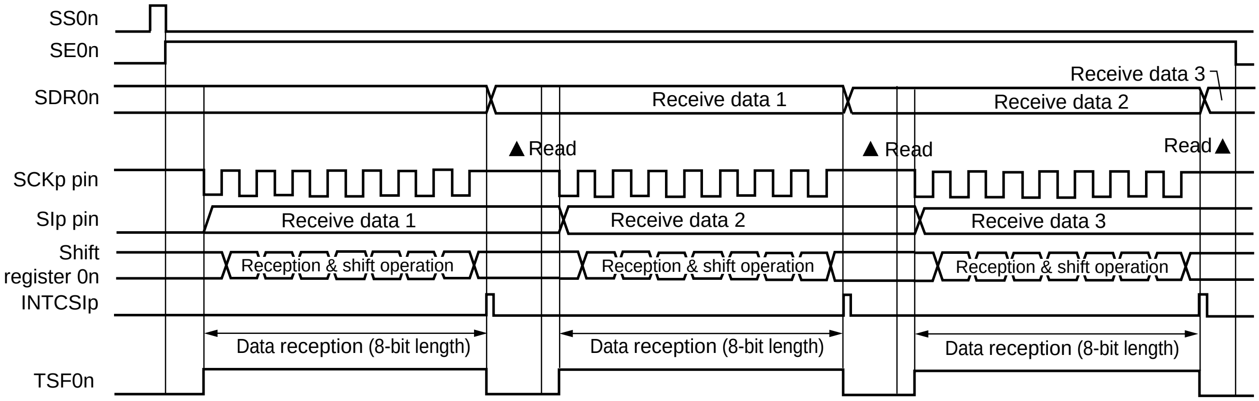
Enable clock output of the target channel by setting a port register and a port mode register.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

Wait for a clock from the master.

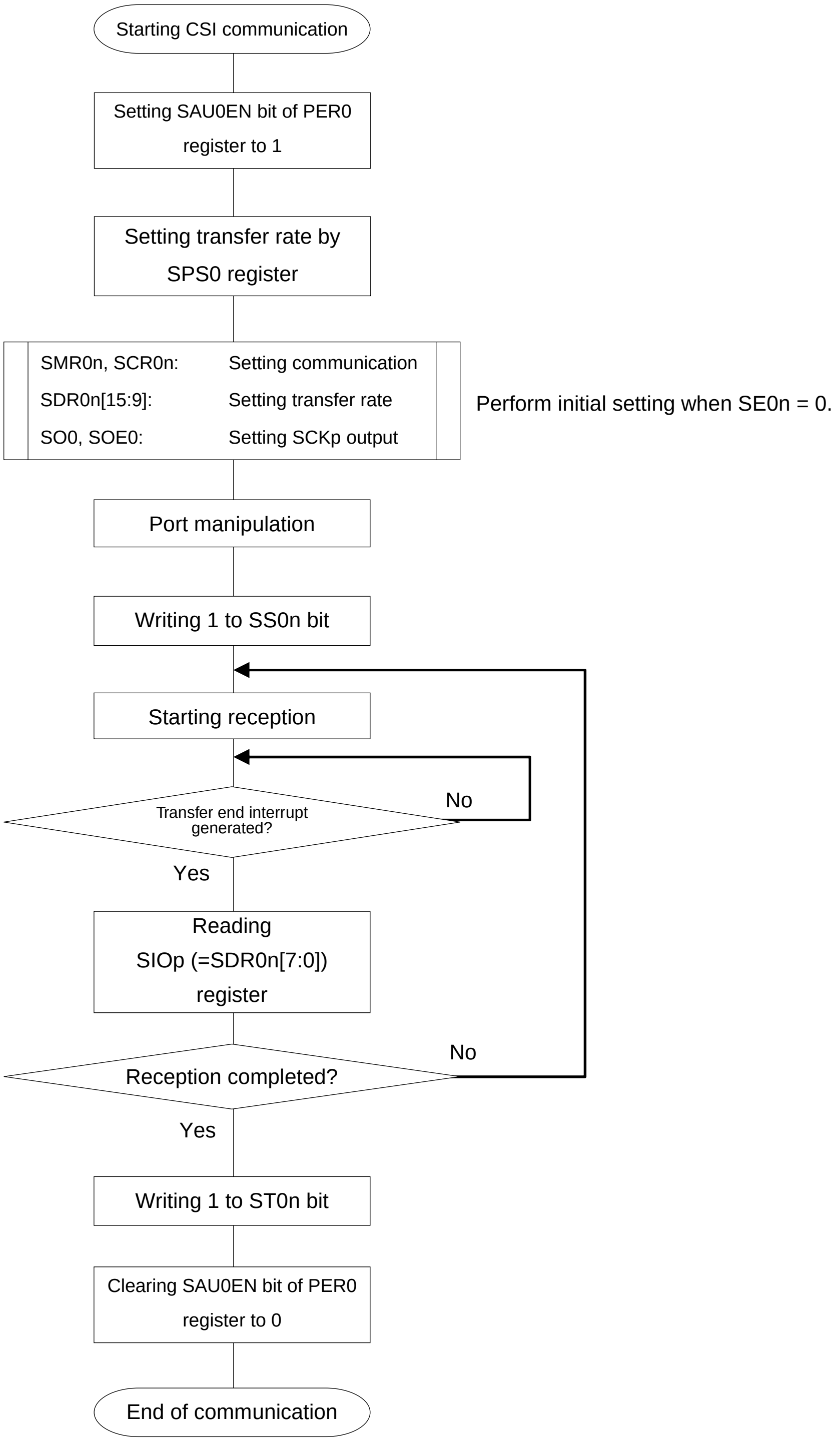
(3) Processing flow (in single-reception mode)

Figure 12-57. Timing Chart of Slave Reception (in Single-Reception Mode)


SS0n
SE0n
SDR0n
Receive data 1
Receive data 2
Receive data 3
Read
Read
Read
SCKp pin
SIp pin
Receive data 1
Receive data 2
Receive data 3
Shift register 0n
Reception & shift operation
Reception & shift operation
Reception & shift operation
INTCSIp
Data reception (8-bit length)
Data reception (8-bit length)
Data reception (8-bit length)
TSF0n

Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)

Figure 12-58. Flowchart of Slave Reception (in Single-Reception Mode)


Starting CSI communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0, SOE0: Setting SCKp output
Perform initial setting when SE0n = 0.
Port manipulation
Writing 1 to SS0n bit
Starting reception
Transfer end interrupt generated?
No
Yes
Reading SIOp (=SDR0n[7:0]) register
Reception completed?
No
Yes
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

12.5.6 Slave transmission/reception

Slave transmission/reception is that the μPD79F9210/1 transmits/receives data to/from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10
Target channel	Channel 0 of SAU	Channel 1 of SAU	Channel 2 of SAU
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10
Interrupt	INTCSI00	INTCSI01	INTCSI10
	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 (OVF0n) only		
Transfer data length	7 or 8 bits		
Transfer rate	The smaller of $f_{CLK}/6$ [MHz] and $f_{MCK}/2$ [MHz] is the maximum transfer rate Notes 1, 2.		
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data I/O starts from the start of the operation of the serial clock. • DAP0n = 1: Data I/O starts half a clock before the start of the serial clock operation.		
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 1: Reverse		
Data direction	MSB or LSB first		

Notes 1. Because the external serial clock input to pins SCK00, SCK01, and SCK10 is sampled internally and used, the fastest baud rate is the smaller of $f_{CLK}/6$ [MHz] and $f_{MCK}/2$ [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 (TARGET)**).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 0 to 2)

(1) Register setting

Figure 12-59. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01 CSI10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 0/1	SO01 0/1	SO00 0/1

(b) Serial output enable register 0 (SOE0) … 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
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	SOE01 0/1	SOE00 0/1

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 0/1	SS00 0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 1	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 1	RXE0n 1	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0n1 0	PTC0n0 0	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0n1 1	DLS0n0 0/1

(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

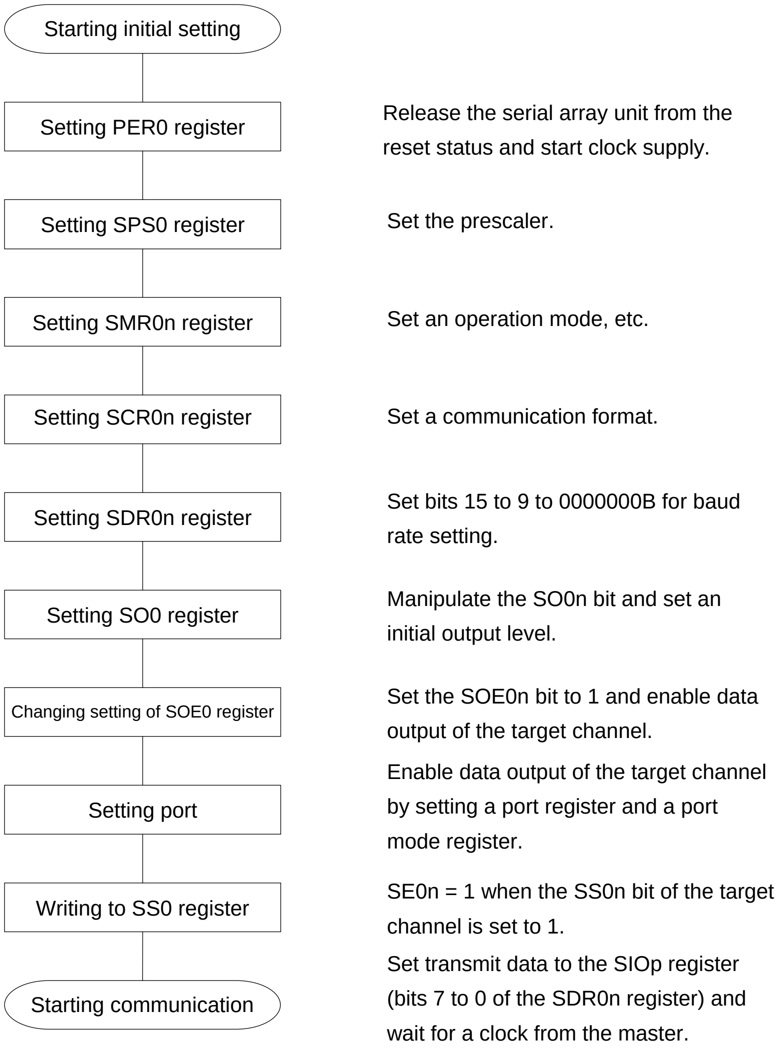
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	0000000 (baud rate setting)							0	Transmit data setting/receive data register							

SIOp

Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)
□ : 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 12-60. Initial Setting Procedure for Slave Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the prescaler.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Manipulate the SO0n bit and set an initial output level.

Set the SOE0n bit to 1 and enable data output of the target channel.

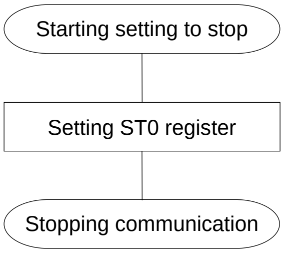
Enable data output of the target channel by setting a port register and a port mode register.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

Set transmit data to the SIOp register (bits 7 to 0 of the SDR0n register) and wait for a clock from the master.

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 12-61. Procedure for Stopping Slave Transmission/Reception

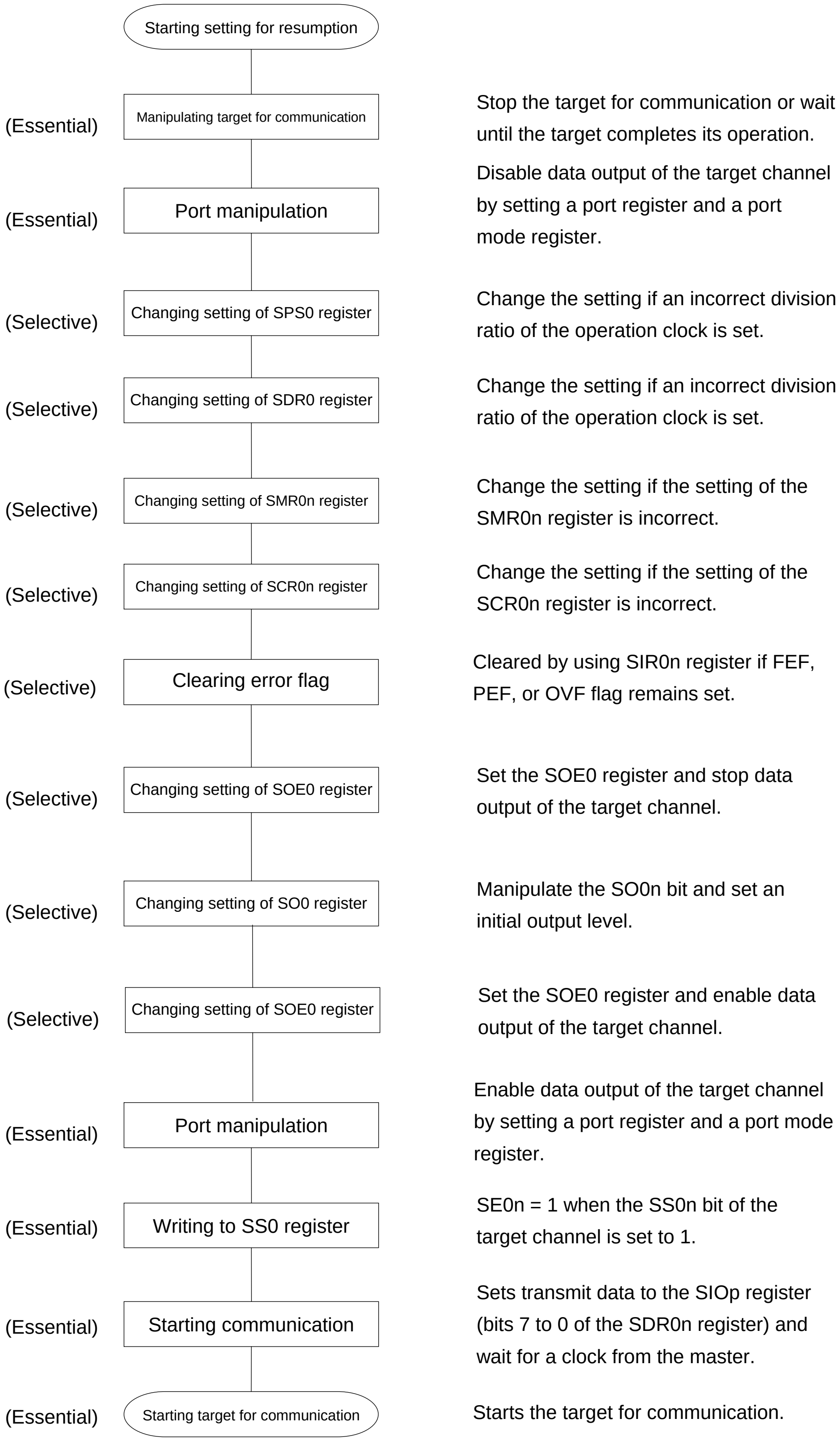


Write 1 to the ST0n bit of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 12-62 Procedure for Resuming Slave Transmission/Reception**).

Figure 12-62. Procedure for Resuming Slave Transmission/Reception


Starting setting for resumption
(Essential)
Manipulating target for communication
(Essential)
Port manipulation
(Selective)
Changing setting of SPS0 register
(Selective)
Changing setting of SDR0 register
(Selective)
Changing setting of SMR0n register
(Selective)
Changing setting of SCR0n register
(Selective)
Clearing error flag
(Selective)
Changing setting of SOE0 register
(Selective)
Changing setting of SO0 register
(Selective)
Changing setting of SOE0 register
(Essential)
Port manipulation
(Essential)
Writing to SS0 register
(Essential)
Starting communication
(Essential)
Starting target for communication

Stop the target for communication or wait until the target completes its operation.

Disable data output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if the setting of the SMR0n register is incorrect.

Change the setting if the setting of the SCR0n register is incorrect.

Cleared by using SIR0n register if FEF, PEF, or OVF flag remains set.

Set the SOE0 register and stop data output of the target channel.

Manipulate the SO0n bit and set an initial output level.

Set the SOE0 register 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.

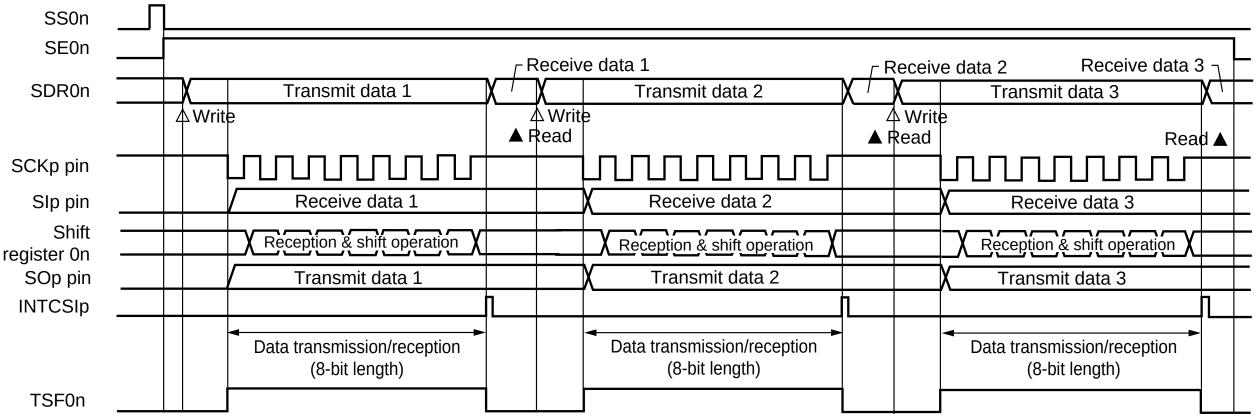
SE0n = 1 when the SS0n bit of the target channel is set to 1.

Sets transmit data to the SIOp register (bits 7 to 0 of the SDR0n register) and wait for a clock from the master.

Starts the target for communication.

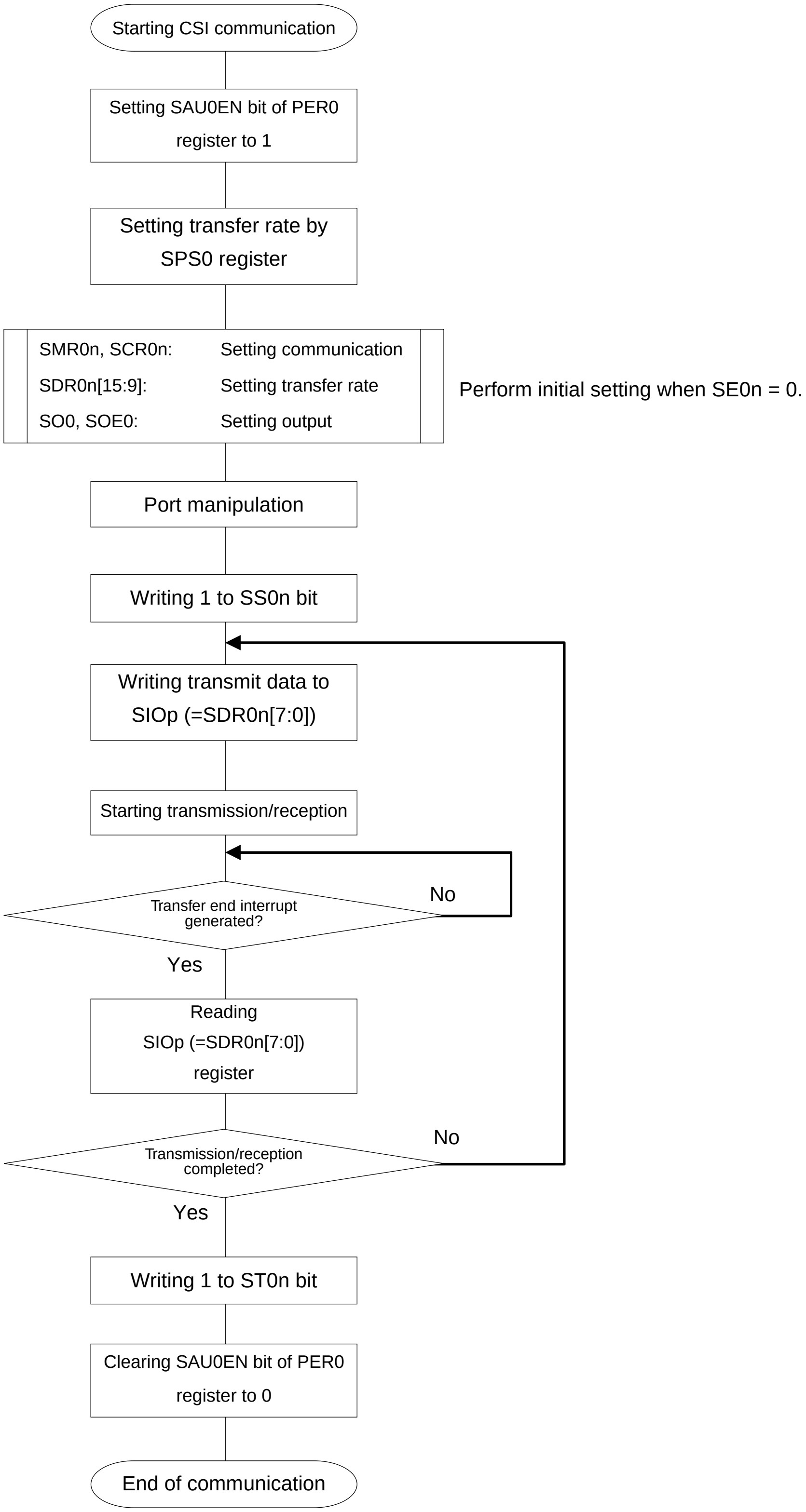
(3) Processing flow (in single-transmission/reception mode)

Figure 12-63. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



Remark n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)

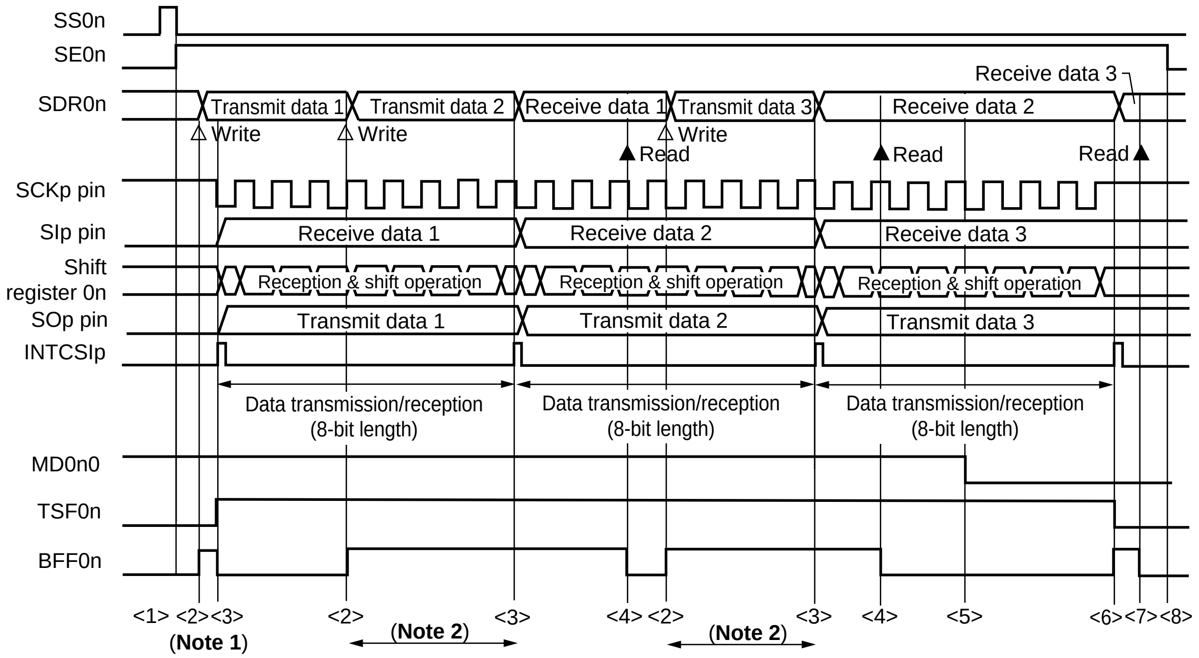
Figure 12-64. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)


Starting CSI communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0, SOE0: Setting output
Perform initial setting when SE0n = 0.
Port manipulation
Writing 1 to SS0n bit
Writing transmit data to SIOp (=SDR0n[7:0])
Starting transmission/reception
Transfer end interrupt generated?
No
Yes
Reading SIOp (=SDR0n[7:0]) register
Transmission/reception completed?
No
Yes
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions **After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.**

(4) Processing flow (in continuous transmission/reception mode)

Figure 12-65. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Notes 1. When transmit data is written to the SDR0n register while BFF0n = 1, the transmit data is overwritten.

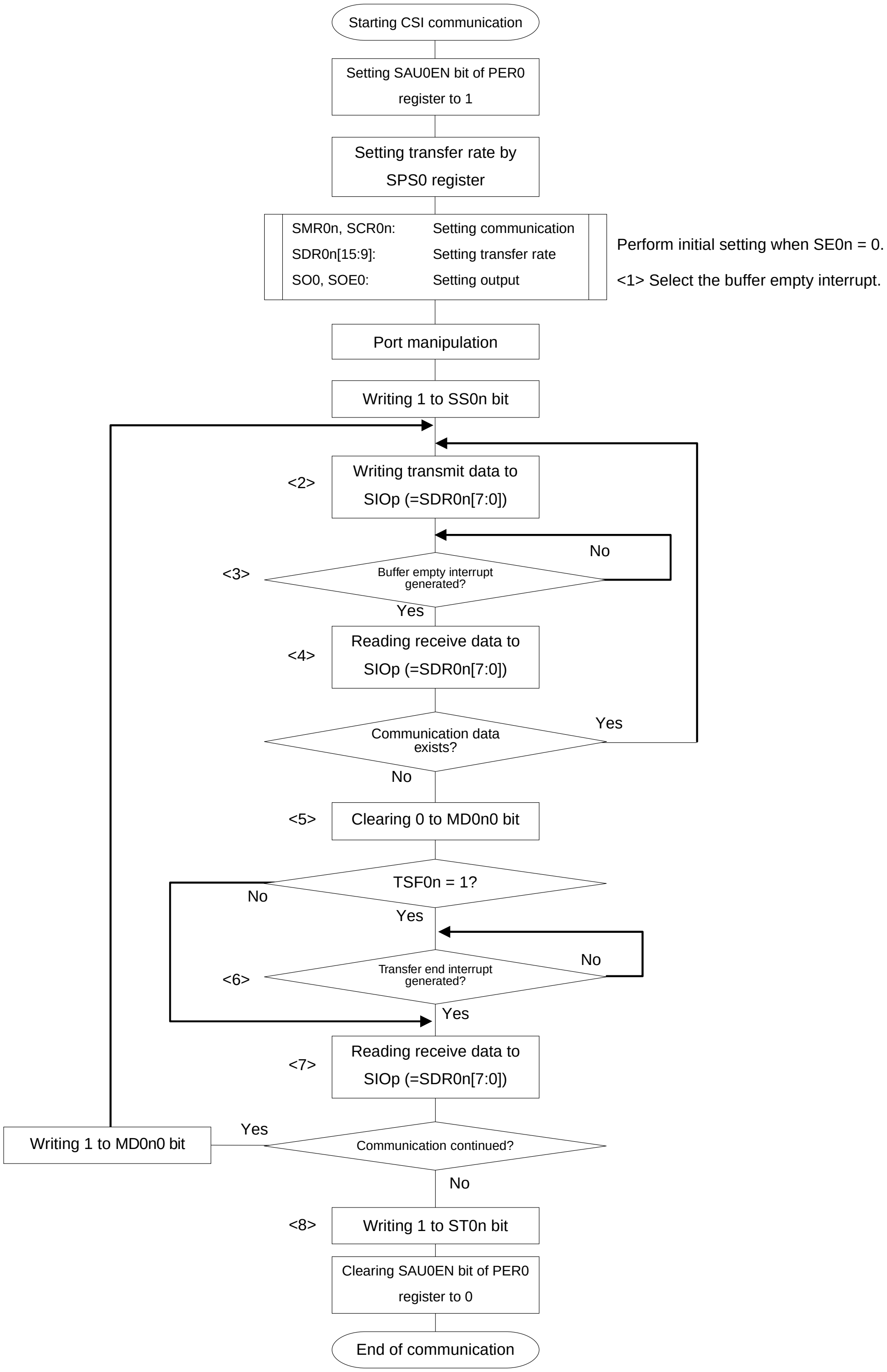
2. The transmit data can be read by reading the SDR0n register during this period. At this time, the transfer operation is not affected.

Caution The MD0n0 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. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-66 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode**).

2. n: Channel number (n = 0 to 2)
p: CSI number (p = 00, 01, 10)

Figure 12-66. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)


Starting CSI communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0, SOE0: Setting output
Perform initial setting when SE0n = 0.
<1> Select the buffer empty interrupt.
Port manipulation
Writing 1 to SS0n bit
<2>
Writing transmit data to SIOp (=SDR0n[7:0])
<3>
Buffer empty interrupt generated?
No
Yes
<4>
Reading receive data to SIOp (=SDR0n[7:0])
Communication data exists?
Yes
No
<5>
Clearing 0 to MD0n0 bit
TSF0n = 1?
No
Yes
<6>
Transfer end interrupt generated?
No
Yes
<7>
Reading receive data to SIOp (=SDR0n[7:0])
Writing 1 to MD0n0 bit
Yes
Communication continued?
No
<8>
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-65 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

12.5.7 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI01, CSI10) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (MCK) frequency of target channel} ÷ (SDR0n[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master}[Note] [Hz]

Note. The permissible maximum frequency is the smaller of f_{CLK}/6 [MHz] and f_{MCK}/2 [MHz].

Remarks 1. The value of SDR0n[15:9] is the value of bits 15 to 9 of the SDR0n register (0000000B to 1111111B) and therefore is 0 to 127.

2. n: Channel number (n = 0 to 2)

The operation clock (MCK) is determined by serial clock select register 0 (SPS0) and bit 15 (CKS0n) of serial mode register 0n (SMR0n).

Table 12-2. Selection of operation clock

SMR0n Register	SPS0 Register								Operation Clock (MCK) [Note1]	
CKS0n	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000		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[Note2]	
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[Note2]	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (ST0 = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 for the operation clock, also stop the timer array unit TAUS (TT0 = 00FFH).

2. SAU can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by setting the TIS02 bit of the TIS0 register of TAUS to 1, selecting $f_{SUB}/4$ for the input clock, and selecting INTTM02 using the SPS0 register. When changing f_{CLK}, however, SAU and TAUS must be stopped as described in Note 1 above.

Remarks 1. X: Don’t care

2. n: Channel number (n = 0 to 2)

12.6 Operation of UART (UART0, UART1) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[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 UART0 (0, 1 channels of unit)

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

External interrupt (INTP0) or timer array unit TAUS is used.

UART0 uses channels 0 and 1 of SAU.

UART1 uses channels 2 and 3 of SAU.

Channel	Used as CSI	Used as UART	Used as Simplified I^{2}C
0	CSI00	UART0 (supporting LIN-bus)	–
1	CSI01		–
2	CSI10	UART1	IIC10
3	–		–

UART performs the following four types of communication operations.

- UART transmission (See **12.6.1**.)
- UART reception (See **12.6.2**.)
- LIN transmission (UART0 only) (See **12.6.3**.)
- LIN reception (UART0 only) (See **12.6.4**.)

12.6.1 UART transmission

UART transmission is an operation to transmit data from the μPD79F9210/1 to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0	UART1
Target channel	Channel 0 of SAU	Channel 2 of SAU
Pins used	TxD0	TxD1
Interrupt	INTST0	INTST1
	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] (SDR0n [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 (TARGET)**

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel

f_{CLK}: System clock frequency

2. n: Channel number (n = 0, 2)

(1) Register setting

Figure 12-67. Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (1/2)

(a) Serial output register 0 (SO0) … 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
SO0	0	0	0	0	1	CKO02 ×	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 0/1Note	SO01 ×	SO00 0/1Note

(b) Serial output enable register 0 (SOE0) … 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
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	SOE01 ×	SOE00 0/1

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 0/1

(d) Serial output level register 0 (SOL0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL02 0/1	0	SOL00 0/1

0: Forward (normal) transmission
1: Reverse transmission

(e) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 0	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 1	MD0n0 0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

Note. Before transmission is started, be sure to set to 1 when the SOL0n bit of the target channel is set to 0, and set to 0 when the SOL0n 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 12-67. Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (2/2)

(f) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 1	RXE0n 0	DAP0n 0	CKP0n 0	0	EOC0n 0	PTC0n1 0/1	PTC0n0 0/1	DIR0n 0/1	0	SLC0n1 0/1	SLC0n0 0/1	0	DLS0n2 1	DLS0n1 0/1	DLS0n0 0/1

Setting of parity bit
00B: No parity
01B: 0 parity
10B: Even parity
11B: Odd parity

Setting of stop bit
01B: Appending 1 bit
10B: Appending 2 bits

(g) Serial data register 0n (SDR0n) (lower 8 bits: TXDq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	Baud rate setting							0	Transmit data setting							
									TXDq							

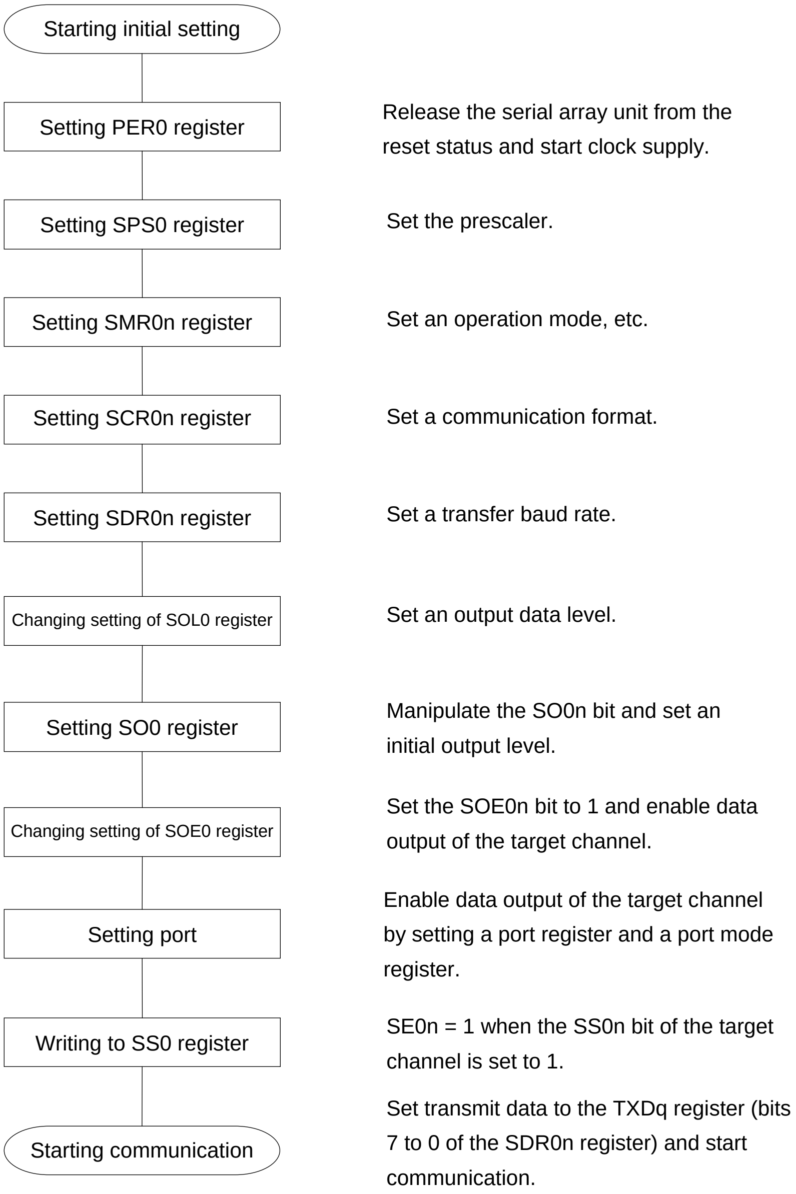
Remark n: Channel number (n = 0, 2), q: UART number (q = 0, 1)

□: Setting is fixed in the UART transmission mode, ▨ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-68. Initial Setting Procedure for UART Transmission



Release the serial array unit from the reset status and start clock supply.

Set the prescaler.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate.

Set an output data level.

Manipulate the SO0n bit and set an initial output level.

Set the SOE0n bit to 1 and enable data output of the target channel.

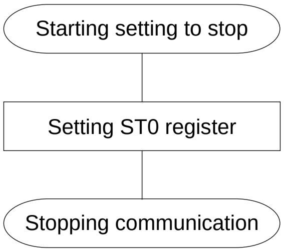
Enable data output of the target channel by setting a port register and a port mode register.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

Set transmit data to the TXDq register (bits 7 to 0 of the SDR0n register) and start communication.

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 12-69. Procedure for Stopping UART Transmission

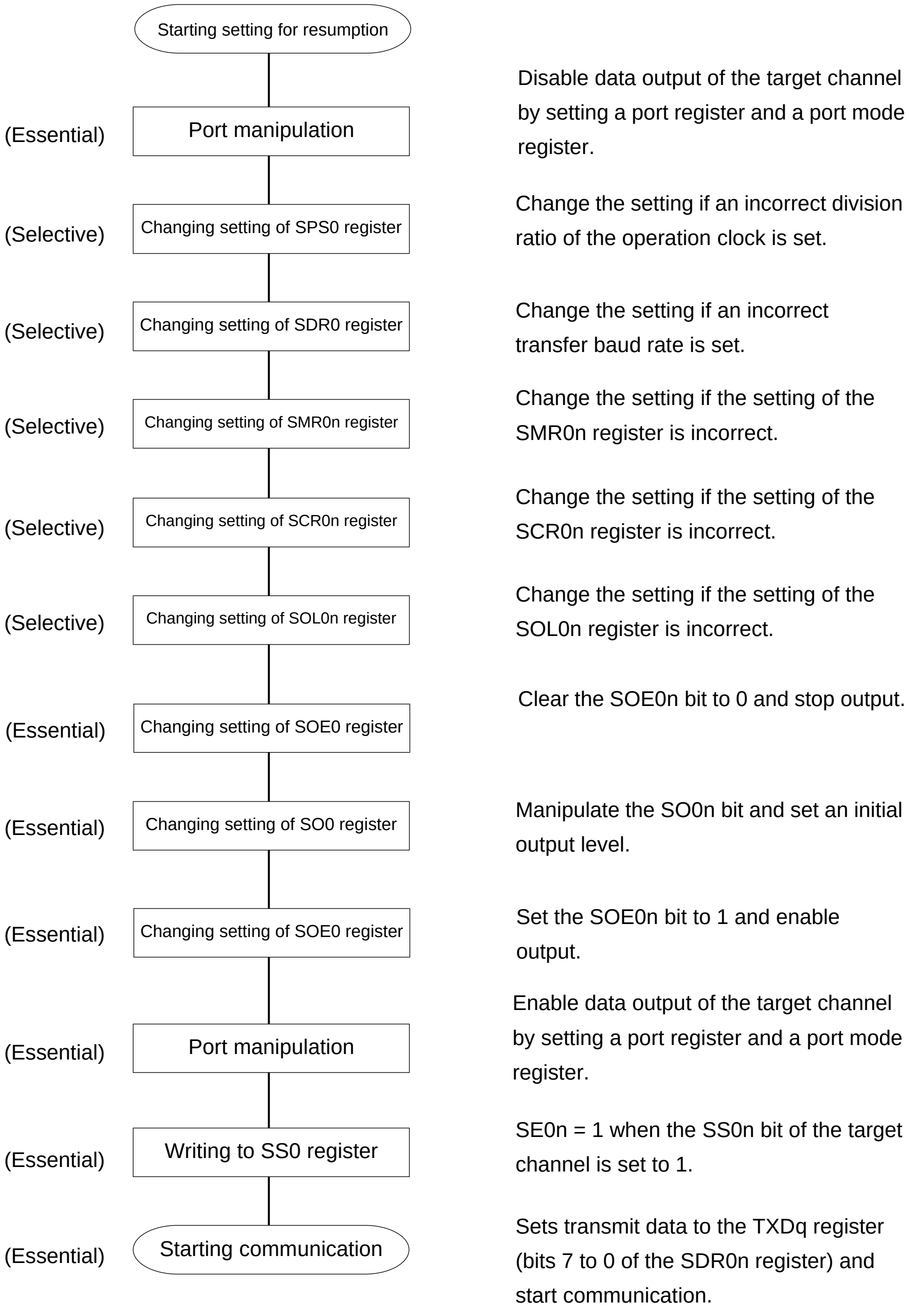


Write 1 to the ST0n bit of the target channel.

Stop communication in midway.

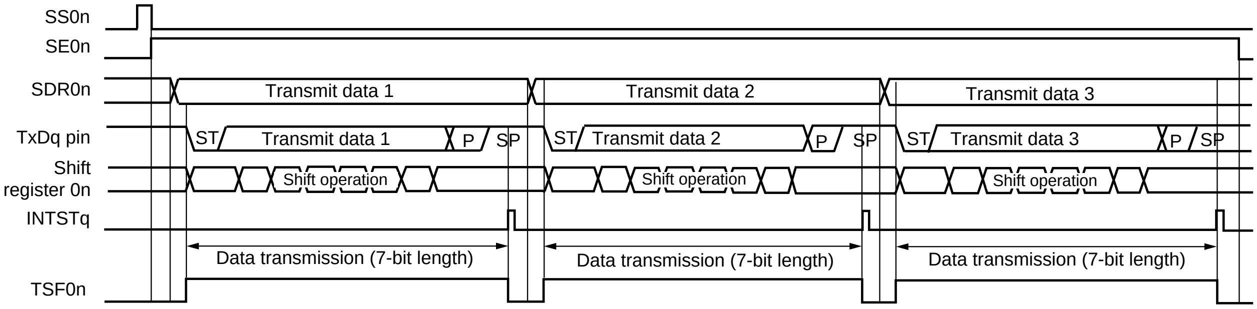
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 12-70 Procedure for Resuming UART Transmission**).

Figure 12-70. Procedure for Resuming UART Transmission


Starting setting for resumption
(Essential)
Port manipulation
Disable data output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of SPS0 register
Change the setting if an incorrect division ratio of the operation clock is set.
(Selective)
Changing setting of SDR0 register
Change the setting if an incorrect transfer baud rate is set.
(Selective)
Changing setting of SMR0n register
Change the setting if the setting of the SMR0n register is incorrect.
(Selective)
Changing setting of SCR0n register
Change the setting if the setting of the SCR0n register is incorrect.
(Selective)
Changing setting of SOL0n register
Change the setting if the setting of the SOL0n register is incorrect.
(Essential)
Changing setting of SOE0 register
Clear the SOE0n bit to 0 and stop output.
(Essential)
Changing setting of SO0 register
Manipulate the SO0n bit and set an initial output level.
(Essential)
Changing setting of SOE0 register
Set the SOE0n bit to 1 and enable output.
(Essential)
Port manipulation
Enable data output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to SS0 register
SE0n = 1 when the SS0n bit of the target channel is set to 1.
(Essential)
Starting communication
Sets transmit data to the TXDq register (bits 7 to 0 of the SDR0n register) and start communication.

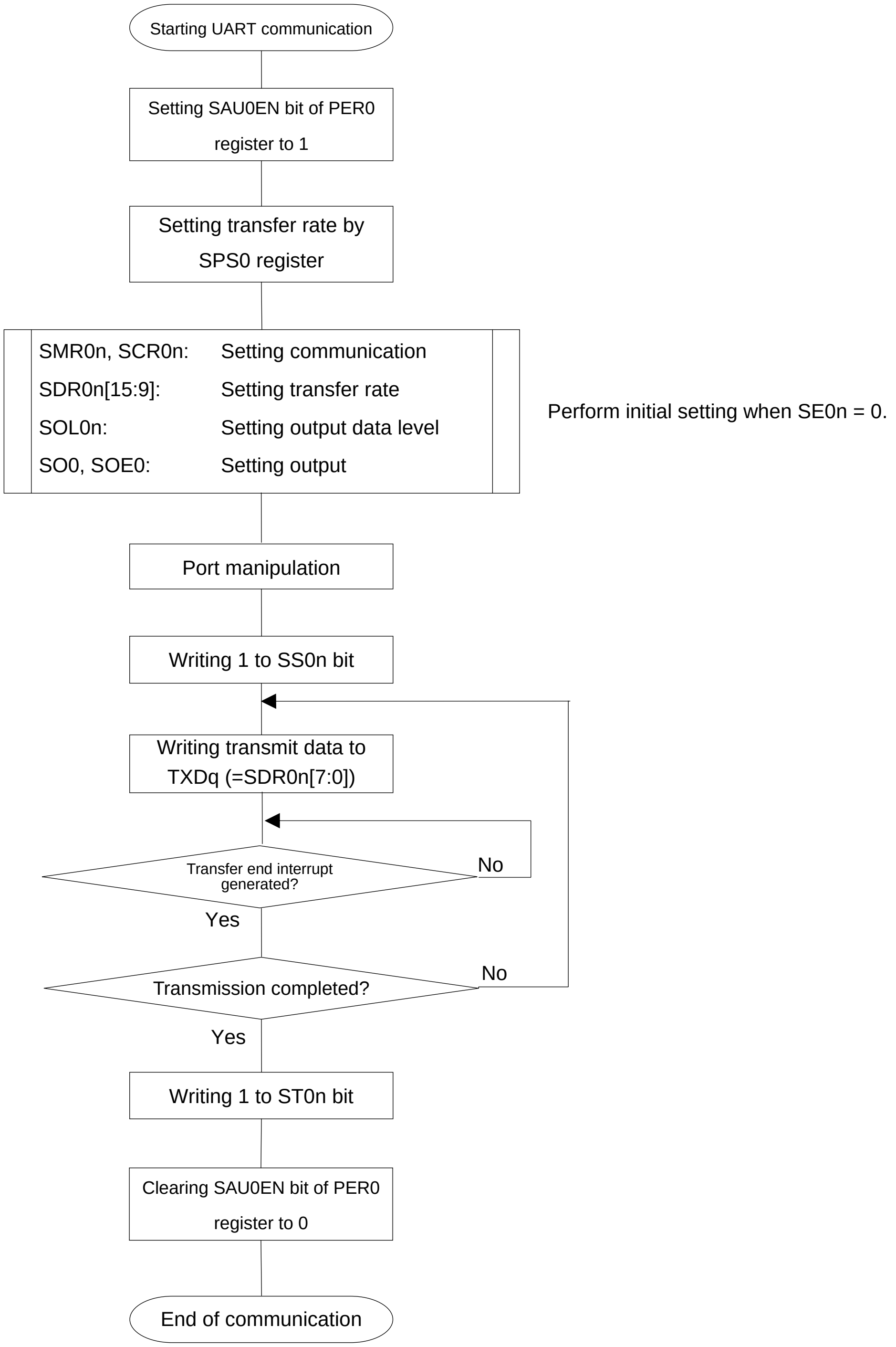
(3) Processing flow (in single-transmission mode)

Figure 12-71. Timing Chart of UART Transmission (in Single-Transmission Mode)



Remark n: Channel number (n = 0, 2), q: UART number (q = 0, 1)

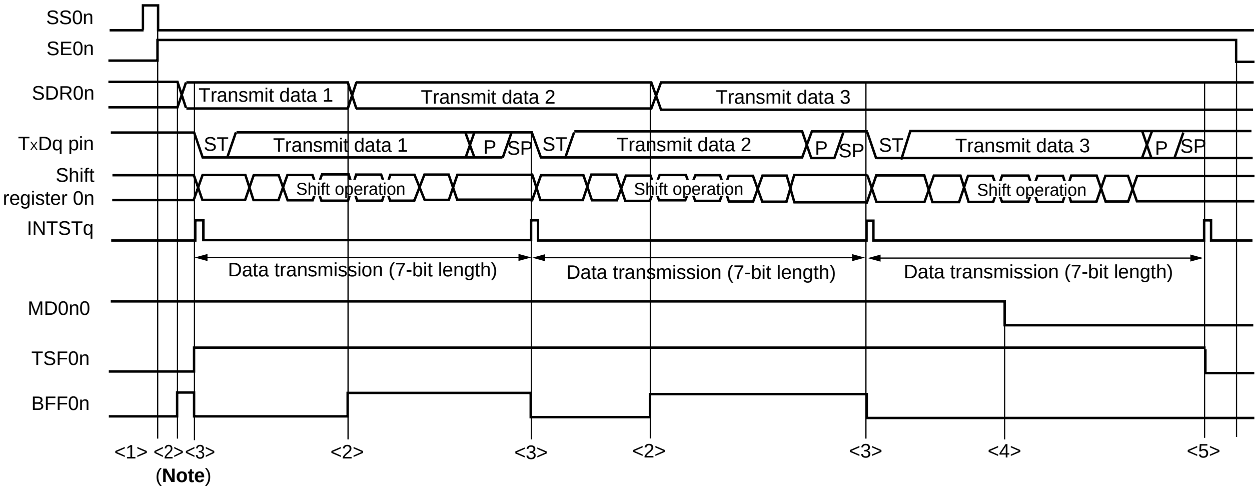
Figure 12-72. Flowchart of UART Transmission (in Single-Transmission Mode)


Starting UART communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SOL0n: Setting output data level
SO0, SOE0: Setting output
Perform initial setting when SE0n = 0.
Port manipulation
Writing 1 to SS0n bit
Writing transmit data to TXDq (=SDR0n[7:0])
Transfer end interrupt generated?
No
Yes
Transmission completed?
No
Yes
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 12-73. Timing Chart of UART Transmission (in Continuous Transmission Mode)

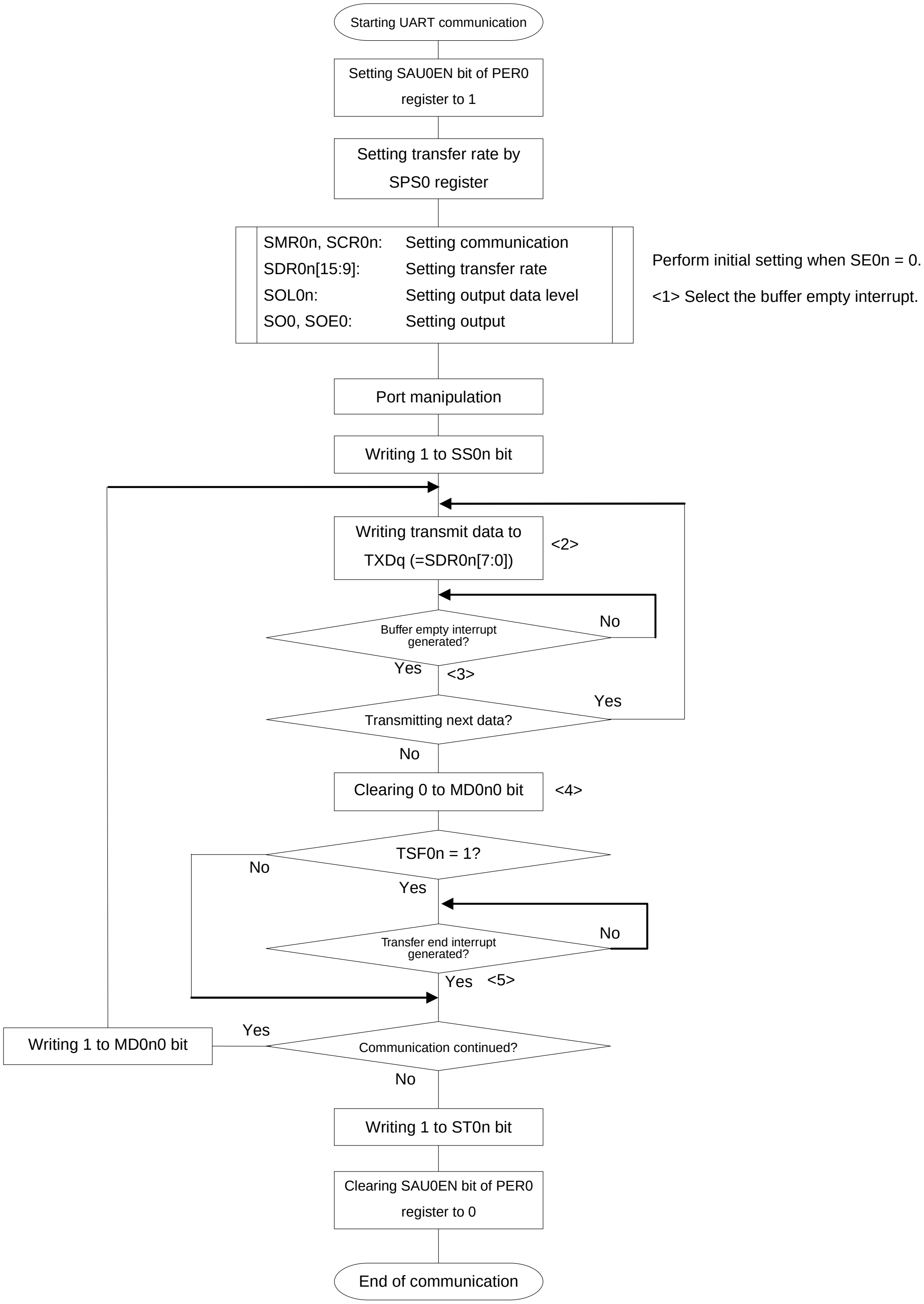


Note When transmit data is written to the SDR0n register while BFF0n = 1, the transmit data is overwritten.

Caution The MD0n0 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 = 0, 1)

Figure 12-74. Flowchart of UART Transmission (in Continuous Transmission Mode)


Starting UART communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SOL0n: Setting output data level
SO0, SOE0: Setting output
Perform initial setting when SE0n = 0.
<1> Select the buffer empty interrupt.
Port manipulation
Writing 1 to SS0n bit
Writing transmit data to TXDq (=SDR0n[7:0])
<2>
Buffer empty interrupt generated?
No
Yes
<3>
Transmitting next data?
Yes
No
Clearing 0 to MD0n0 bit
<4>
TSF0n = 1?
No
Yes
Transfer end interrupt generated?
No
Yes
<5>
Writing 1 to MD0n0 bit
Yes
Communication continued?
No
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of communication

Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Remark <1> to <5> in the figure correspond to <1> to <5> in **Figure 12-73 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

12.6.2 UART reception

UART reception is an operation wherein the μPD79F9210/1 asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd channel of the two channels used for UART is used.

UART	UART0	UART1
Target channel	Channel 1 of SAU	Channel 3 of SAU
Pins used	RxD0	RxD1
Interrupt	INTSR0	INTSR1
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE0	INTSRE1
Error detection flag	• Framing error detection flag (FEF0n) • Parity error detection flag (PEF0n) • Overrun error detection flag (OVF0n)	
Transfer data length	5, 7 or 8 bits	
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR0n [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note	
Data phase	Forward output (default: high level) Reverse output (default: low level)	
Parity bit	The following selectable • No parity bit (no parity check) • Appending 0 parity (no parity check) • Appending even parity • Appending odd parity	
Stop bit	Appending 1 bit	
Data direction	MSB or LSB first	

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (TARGET)**).

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 12-75. Example of Contents of Registers for UART Reception of UART (UART0, UART1) (1/2)

(a) Serial output register 0 (SO0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 ×	SO01 ×	SO00 ×

(b) Serial output enable register 0 (SOE0)

	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	SOE02 ×	SOE01 0/1	SOE00 ×

Set this bit to 0 when using them for UART0 reception

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 0/1	SS02 ×	SS01 0/1	SS00 ×

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 0	0	0	0	0	0	STS0n 1	0	SIS0n0 0/1	1	0	0	MD0n2 0	MD0n1 1	MD0n0 0

0: Forward (normal) reception
1: Reverse reception

Operation mode of channel n
0: Transfer end interrupt

(e) Serial mode register 0r (SMR0r)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0r	CKS0r 0/1	CCS0r 0	0	0	0	0	0	STS0r 0	0	SIS0r0 0	1	0	0	MD0r2 0	MD0r1 1	MD0r0 0/1

Same setting value as CKS0n

Operation mode of channel r
0: Transfer end interrupt
1: Buffer empty interrupt

Caution For the UART reception, be sure to set SMR0r of channel r that is to be paired with channel n.

Remark n: Channel number (n = 1, 3), r: Channel number (r = n – 1)

□ : 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 12-75. Example of Contents of Registers for UART Reception of UART (UART0, UART1) (2/2)

(f) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0	DIR0n		SLC0n1	SLC0n0		DLS0n2	DLS0n1	DLS0n0
	0	1	0	0	0	1	0/1	0/1	0/1	0	0	1	0	1	0/1	0/1

(g) Serial data register 0n (SDR0n) (lower 8 bits: RXDq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	Baud rate setting							0	Receive data register							
									RXDq							

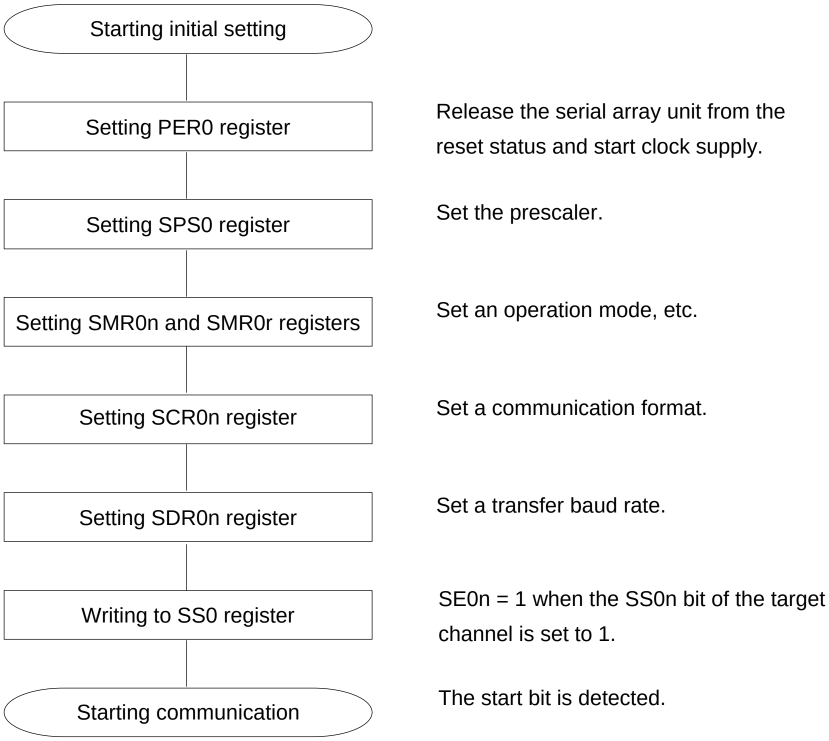
Remark n: Channel number (n = 1, 3), q: UART number (q = 0, 1)

□ : Setting is fixed in the UART reception mode, ▨ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-76. Initial Setting Procedure for UART Reception


Starting initial setting
Setting PER0 register
Release the serial array unit from the reset status and start clock supply.
Setting SPS0 register
Set the prescaler.
Setting SMR0n and SMR0r registers
Set an operation mode, etc.
Setting SCR0n register
Set a communication format.
Setting SDR0n register
Set a transfer baud rate.
Writing to SS0 register
SE0n = 1 when the SS0n bit of the target channel is set to 1.
Starting communication
The start bit is detected.

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 12-77. Procedure for Stopping UART Reception

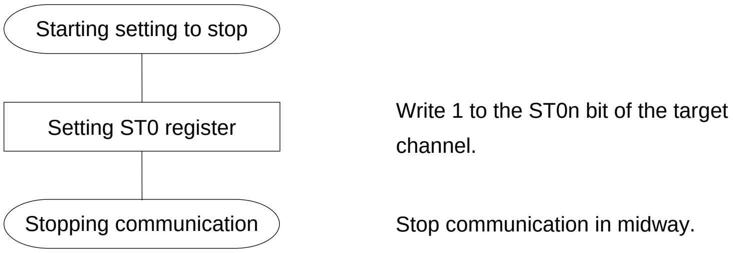
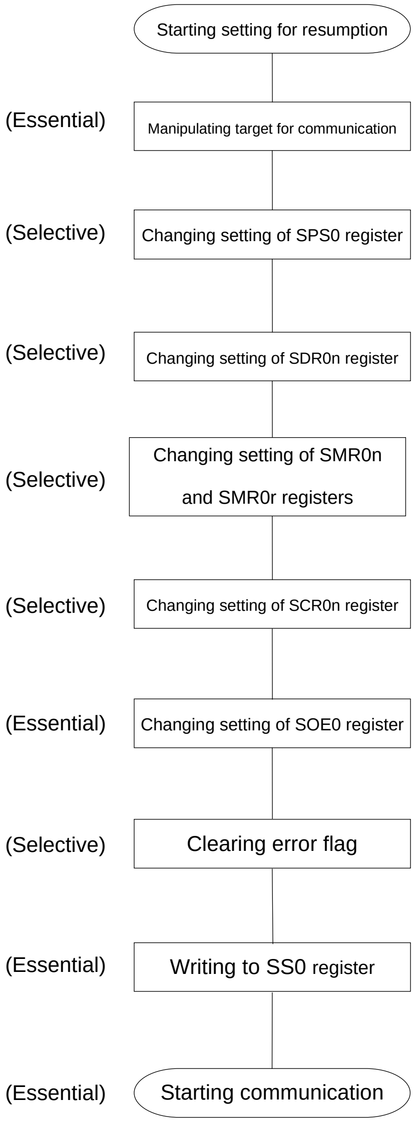

Starting setting to stop
Setting ST0 register
Write 1 to the ST0n bit of the target channel.
Stopping communication
Stop communication in midway.

Figure 12-78. Procedure for Resuming UART Reception


Starting setting for resumption
(Essential)
Manipulating target for communication
(Selective)
Changing setting of SPS0 register
(Selective)
Changing setting of SDR0n register
(Selective)
Changing setting of SMR0n and SMR0r registers
(Selective)
Changing setting of SCR0n register
(Essential)
Changing setting of SOE0 register
(Selective)
Clearing error flag
(Essential)
Writing to SS0 register
(Essential)
Starting communication

Stop the target for communication or wait until the target completes its operation.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect transfer baud rate is set.

Change the setting if the setting of the SMR0n and SMR0r registers is incorrect.

Change the setting if the setting of the SCR0n register is incorrect.

Clear the SOE0 register to 0 and stop data output of the target channel.

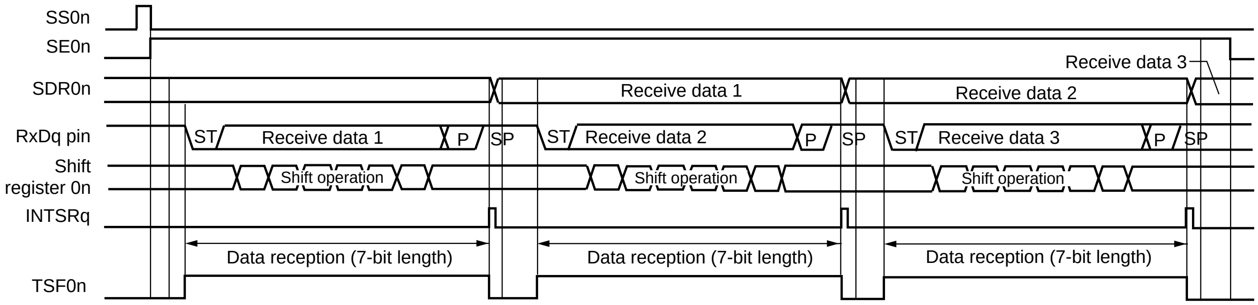
Cleared by using SIR0n register if FEF, PEF, or OVF flag remains set.

SE0n = 1 when the SS0n bit of the target channel is set to 1.

The start bit is detected.

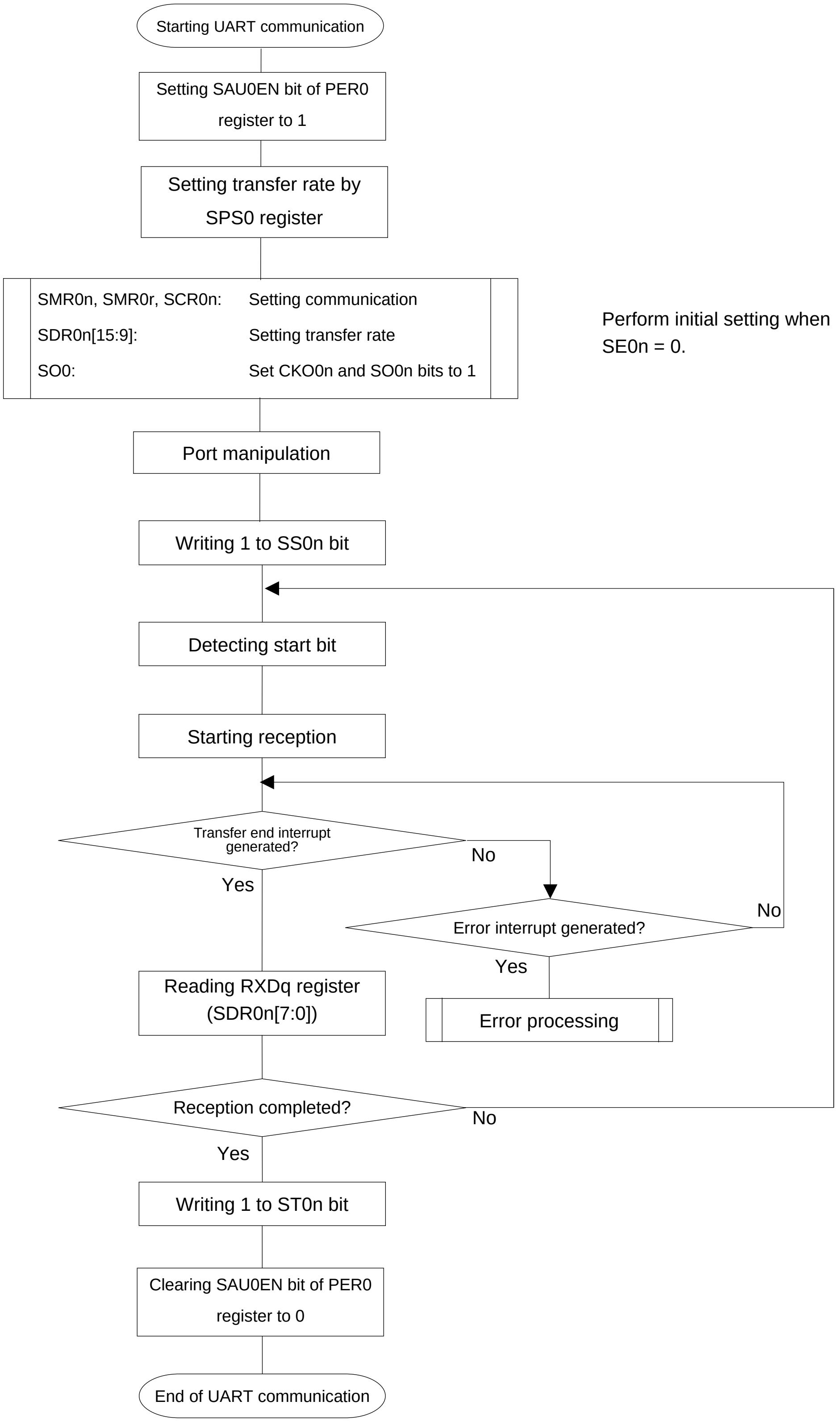
(3) Processing flow

Figure 12-79. Timing Chart of UART Reception



Remark n: Channel number (n = 1, 3), q: UART number (q = 0, 1)

Figure 12-80. Flowchart of UART Reception


Starting UART communication
Setting SAU0EN bit of PER0 register to 1
Setting transfer rate by SPS0 register
SMR0n, SMR0r, SCR0n: Setting communication
SDR0n[15:9]: Setting transfer rate
SO0: Set CKO0n and SO0n bits to 1
Perform initial setting when SE0n = 0.
Port manipulation
Writing 1 to SS0n bit
Detecting start bit
Starting reception
Transfer end interrupt generated?
No
Yes
Error interrupt generated?
No
Yes
Reading RXDq register (SDR0n[7:0])
Error processing
Reception completed?
No
Yes
Writing 1 to ST0n bit
Clearing SAU0EN bit of PER0 register to 0
End of UART communication

Cautions **After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.**

12.6.3 LIN transmission

Of UART transmission, UART0 supports LIN communication.

For LIN transmission, channel 0 of unit (SAU) is used.

UART	UART0	UART1
Support of LIN communication	Supported	Not supported
Target channel	Channel 0 of SAU	–
Pins used	TxD0	–
Interrupt	INTST0	–
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	8 bits	
Transfer rate	Max. f_{MCK}/6 [bps] (SDR00 [15:9] = 2 or more), Min. f_{CLK}/(2 × 2^{11} × 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 (TARGET)**).

Remark f_{MCK}: Operation clock (MCK) 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.

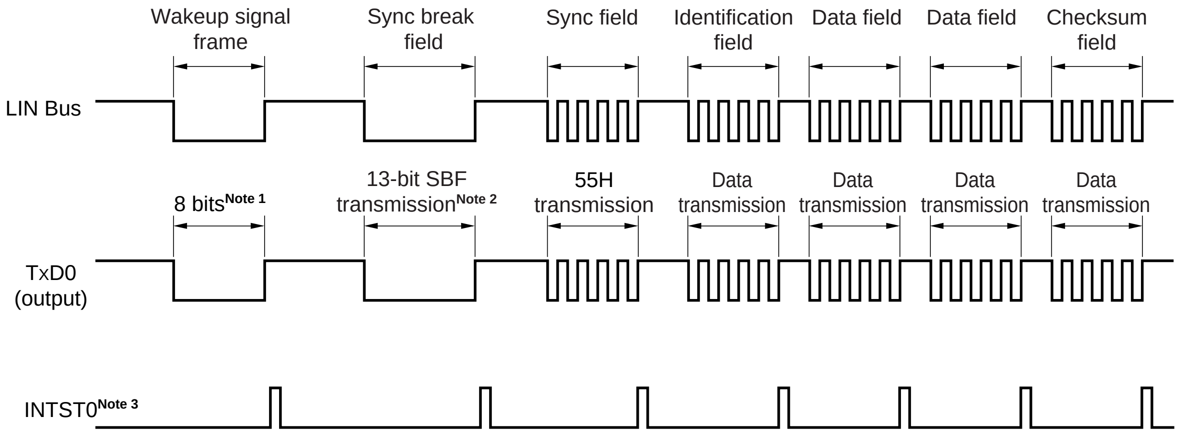
Usually, the master is connected to a network such as CAN (Controller Area Network).

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 12-81 outlines a transmission operation of LIN.

Figure 12-81. 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.

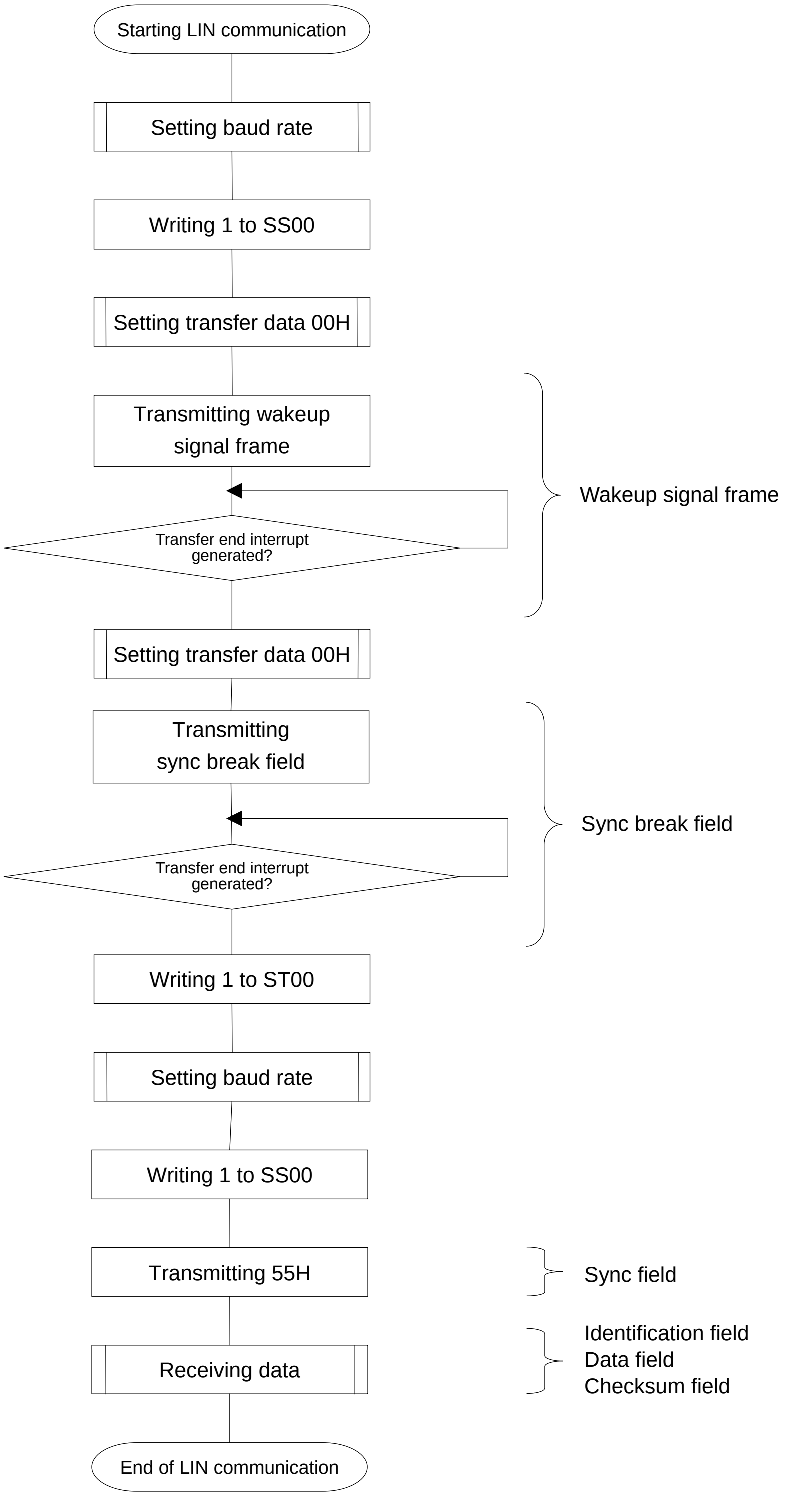
$\boxed{\text{(Baud rate of sync break field)} = 9/13 \times N}$

By transmitting data of 00H at this baud rate, a sync break field is generated.

3. INTST0 is output upon completion of transmission. INTST0 is also output when SBF transmission is executed.

Remark The interval between fields is controlled by software.

Figure 12-82. Flowchart for LIN Transmission



12.6.4 LIN reception

Of UART reception, UART0 supports LIN communication.

For LIN reception, channel 1 of unit (SAU) is used.

UART	UART0	UART1
Support of LIN communication	Supported	Not supported
Target channel	Channel 1 of SAU	–
Pins used	RxD0	–
Interrupt	INTSR0	–
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE0	–
Error detection flag	• Framing error detection flag (FEF01) • Parity error detection flag (PEF01) • Overrun error detection flag (OVF01)	
Transfer data length	8 bits	
Transfer rate	Max. f_{MCK}/6 [bps] (SDR01 [15:9] = 2 or more), Min. f_{CLK}/(2 × 2^{11} × 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	

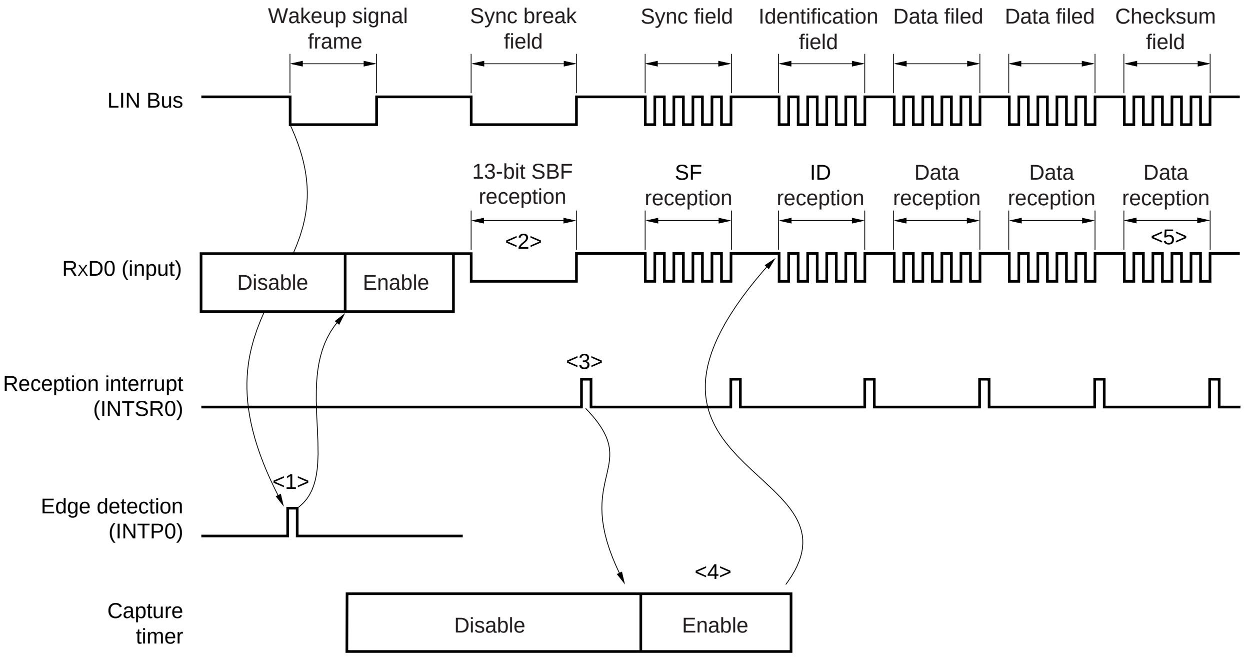
Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (TERGET)**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel

f_{CLK}: System clock frequency

Figure 12-83 outlines a reception operation of LIN.

Figure 12-83. 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 UART0 (RXE01 = 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 RXD0 register (= bits 7 to 0 of the serial data register 01 (SDR01)) at the set baud rate. When the stop bit is detected, the reception end interrupt request (INTSR0) 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 TAUS and measure the bit interval (pulse width) of the sync field (see **6.7.5 Operation as input signal high-/low-level width measurement**).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART0 once and adjust (re-set) the baud rate.

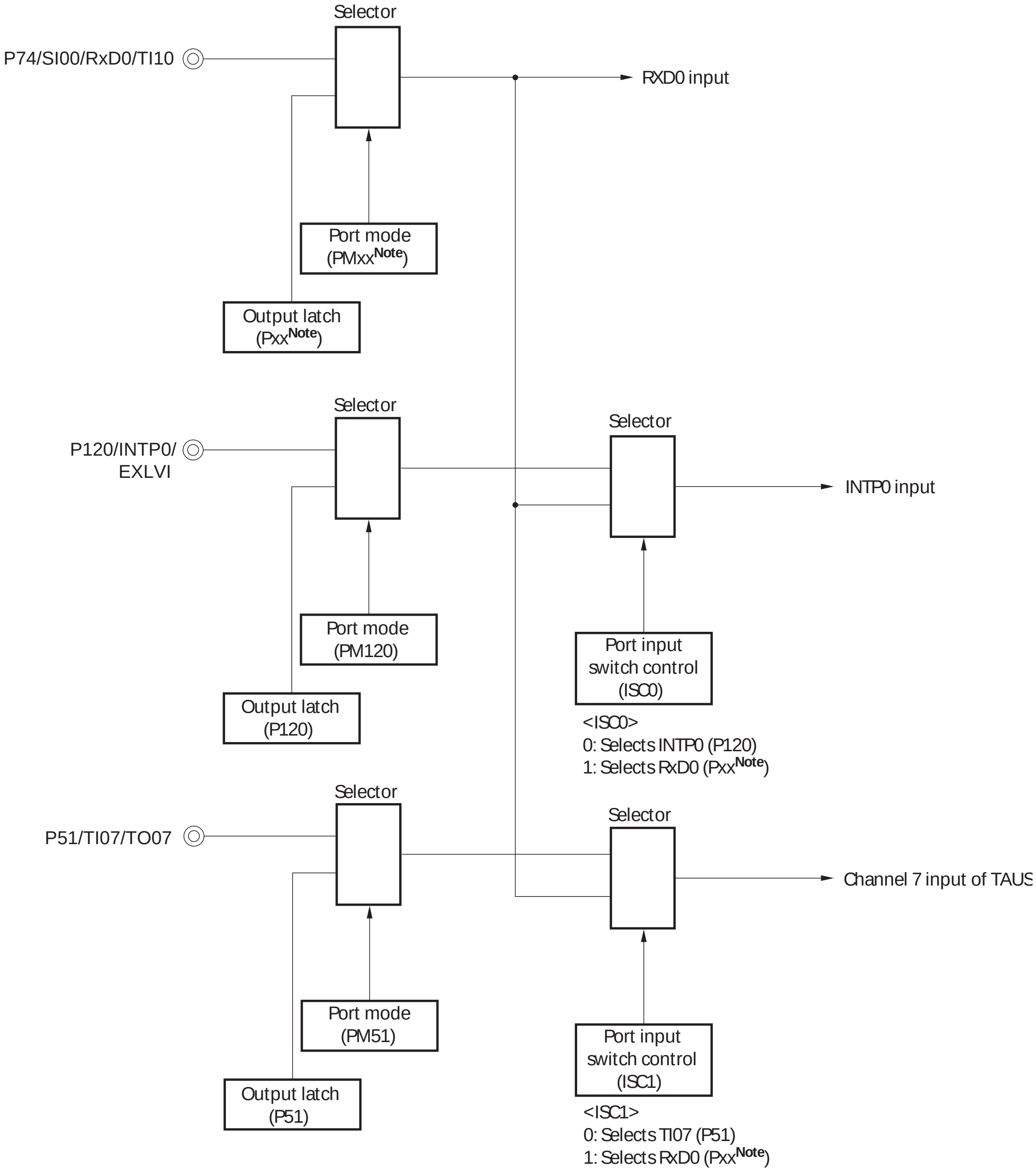
<5> The checksum field should be distinguished by software. In addition, processing to initialize UART0 after the checksum field is received and to wait for reception of SBF should also be performed by software.

Figure 12-84 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 TAUS to calculate a baud-rate error.

By controlling switch of port input (ISC0/ISC1), the input source of port input (RxD0) for reception can be input to the external interrupt pin (INTP0) and timer array unit TAUS.

Figure 12-84. Port Configuration for Manipulating Reception of LIN



Note xx = 74

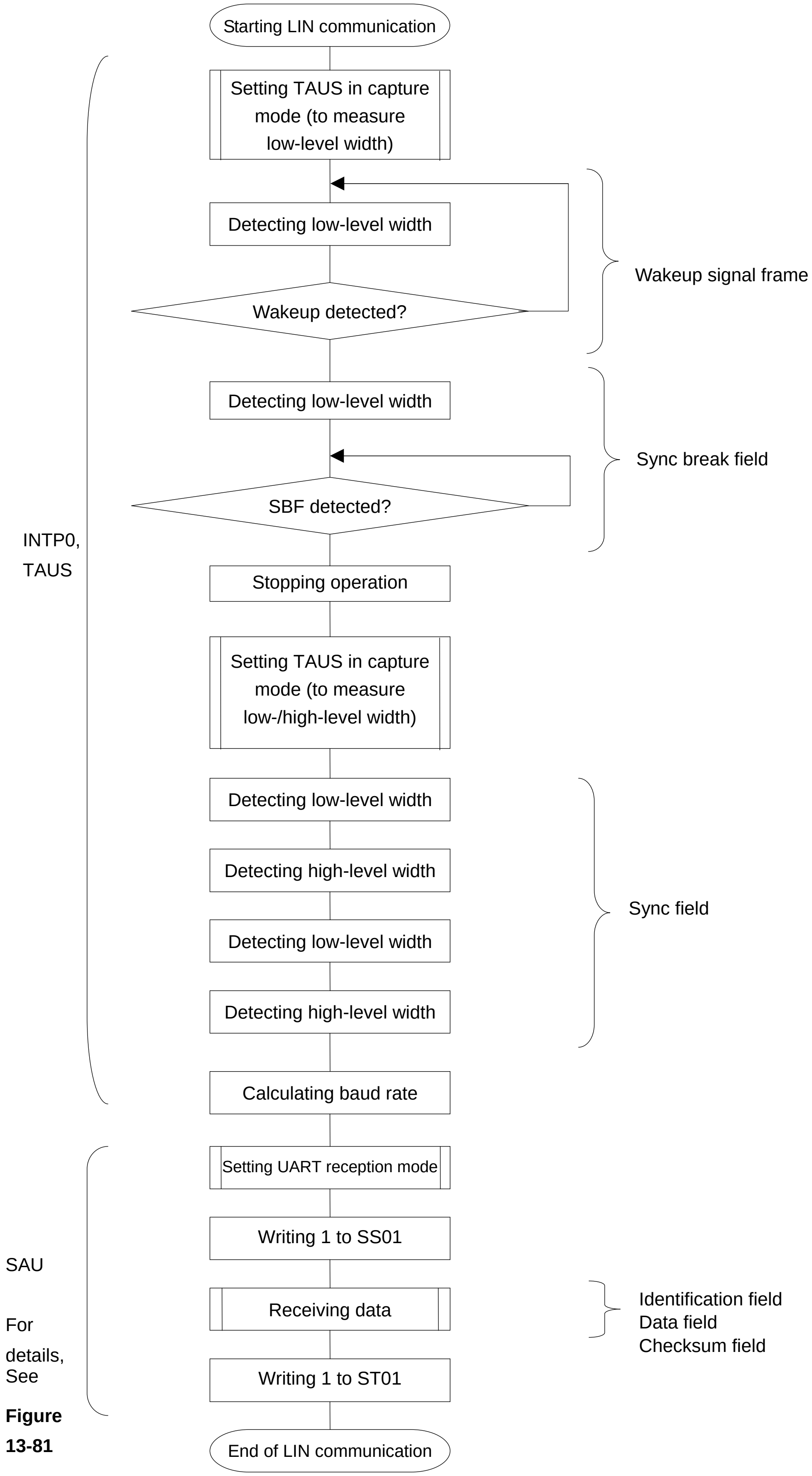
Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 12-16**.)

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 TAUS; 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 RxD0 is measured in the capture mode.)
- Channels 0 and 1 (UART0) of serial array unit (SAU)

Figure 12-85. Flowchart of LIN Reception


Starting LIN communication
Setting TAUS in capture mode (to measure low-level width)
Detecting low-level width
Wakeup detected?
Wakeup signal frame
Detecting low-level width
SBF detected?
Sync break field
INTP0, TAUS
Stopping operation
Setting TAUS in capture mode (to measure low-/high-level width)
Detecting low-level width
Detecting high-level width
Detecting low-level width
Detecting high-level width
Sync field
Calculating baud rate
Setting UART reception mode
Writing 1 to SS01
Receiving data
Identification field
Data field
Checksum field
Writing 1 to ST01
End of LIN communication
SAU
For details, See
Figure 13-81

12.6.5 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0, UART1) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (MCK) frequency of target channel} ÷ (SDR0n[15:9] + 1) ÷ 2 [bps]

Caution Setting SDR0n [15:9] = (0000000B, 0000001B) is prohibited.

Remarks 1. When UART is used, the value of SDR0n[15:9] is the value of bits 15 to 9 of the SDR0n register (0000010B to 1111111B) and therefore is 2 to 127.

2. n: Channel number (n = 0 to 3)

The operation clock (MCK) is determined by serial clock select register 0 (SPS0) and bit 15 (CKS0n) of serial mode register 0n (SMR0n).

Table 12-3. Selection of operation clock

SMR0n Register	SPS0 Register								Operation Clock (MCK) [Note1]	
CKS0n	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000		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[Note2]	
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[Note2]	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (ST0 = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 for the operation clock, also stop the timer array unit TAUS (TT0 = 00FFH).

2. SAU can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by setting the TIS02 bit of the TIS0 register of TAUS to 1, selecting $f_{SUB}/4$ for the input clock, and selecting INTTM02 using the SPS0 register. When changing f_{CLK}, however, SAU and TAUS must be stopped as described in Note 1 above.

Remarks 1. X: Don’t care

2. n: Channel number (n = 0 to 3)

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART1) 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)	SDR0n[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 n: Channel number (n = 0, 2)

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0, UART1) 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.

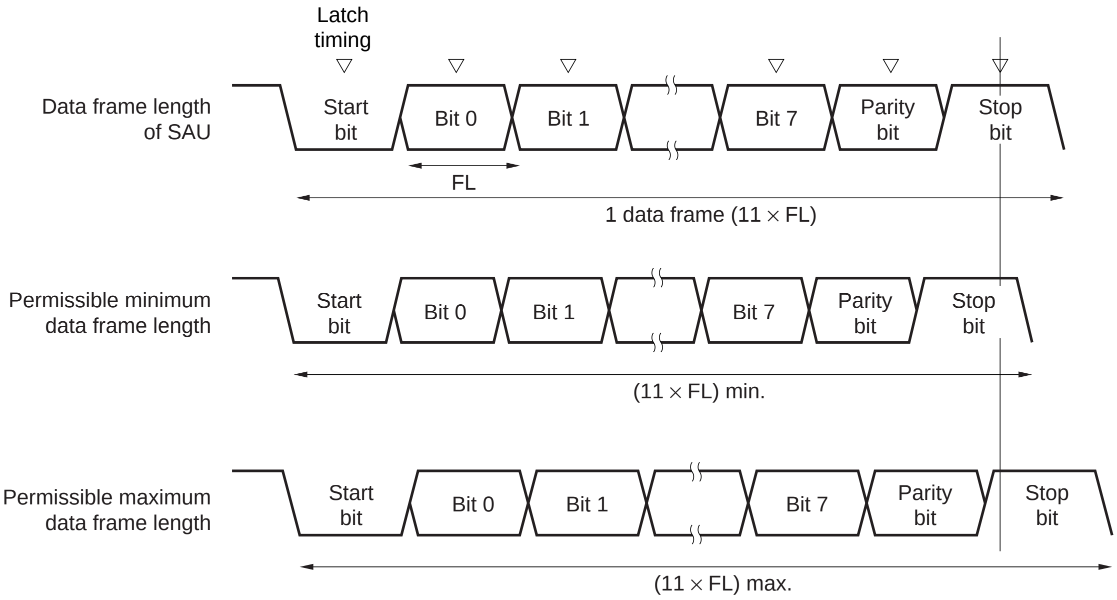
$$(\text{Maximum receivable baud rate}) = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$(\text{Minimum receivable baud rate}) = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **12.6.5 (1) Baud rate calculation expression**.)
k: SDR0n[15:9] + 1
Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark n: Channel number (n = 1, 3)

Figure 12-86. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 12-86, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of the serial data register 0n (SDR0n) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

12.7 Operation of Simplified I^{2}C (IIC10) 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 and ACK detection functions
- 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 I^{2}C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection function

The channel supporting simplified I^{2}C (IIC10) is channel 2 of SAU.

Channel	Used as CSI	Used as UART	Used as Simplified I^{2}C
0	CSI00	UART0 (supporting LIN-bus)	–
1	CSI01		–
2	CSI10	UART1	IIC10
3	–		–

Simplified I^{2}C (IIC10) performs the following four types of communication operations.

- Address field transmission (See **12.7.1**.)
- Data transmission (See **12.7.2**.)
- Data reception (See **12.7.3**.)
- Stop condition generation (See **12.7.4**.)

12.7.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^{2}C 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^{2}C	IIC10
Target channel	Channel 2 of SAU
Pins used	SCL10, SDA10
Interrupt	INTIIC10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Parity error detection flag (PEF02)
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_{CLK}/4 MHz f_{CLK}: System clock frequency However, the following condition must be satisfied in each mode of I^{2}C. • 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

(1) Register setting

Figure 12-87. Example of Contents of Registers for Address Field Transmission of Simplified I^{2}C (IIC10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 0/1	SO01 ×	SO00 ×

Start condition is generated by manipulating the SO02 bit.

(b) Serial output enable register 0 (SOE0) … Sets only the bits of the target 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	SOE02 0/1	SOE01 ×	SOE00 ×

SOE02 = 0 until the start condition is generated, and SOE02 = 1 after generation.

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 ×

(d) Serial mode register 02 (SMR02)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR02	CKS02 0/1	CCS02 0	0	0	0	0	0	STS02 0	0	SIS020 0	1	0	0	MD022 1	MD021 0	MD020 0

Operation mode of channel 2
0: Transfer end interrupt

(e) Serial communication operation setting register 02 (SCR02)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02 1	RXE02 0	DAP02 0	CKP02 0	0	EOC02 0	PTC021 0	PTC020 0	DIR02 0	0	SLC021 0	SLC020 1	0	DLS022 1	DLS021 1	DLS020 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

(f) Serial data register 02 (SDR02) (lower 8 bits: SIO10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02	Baud rate setting							0	Transmit data setting (address + R/W)							
									SIO10							

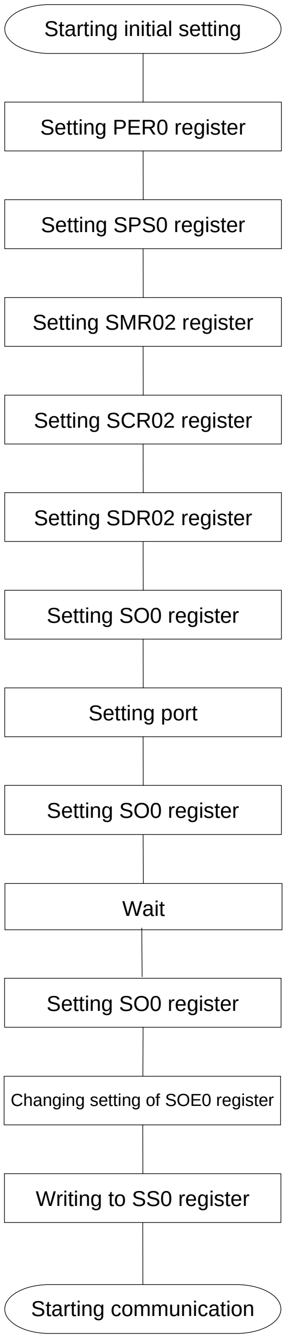
Remark □: Setting is fixed in the IIC mode, ▨: Setting disabled (set to the initial value)

×: 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 12-88. Initial Setting Procedure for Address Field Transmission



Release the serial array unit from the reset status and start clock supply.

Set the prescaler.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate.

Manipulate the SO02 and CKO02 bits and set an initial output level.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Clear the SO02 bit to 0 to generate the start condition.

Secure a wait time so that the specifications of I^{2}C on the slave side are satisfied.

Clear the CKO02 bit to 0 to lower the clock output level.

Set the SOE02 bit to 1 and enable data output of the target channel.

SE02 = 1 when the SS02 bit of the target channel is set to 1.

Set address and R/W to the SIO10 register (bits 7 to 0 of the SDR02 register) and start communication.

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

(3) Processing flow

Figure 12-89. Timing Chart of Address Field Transmission

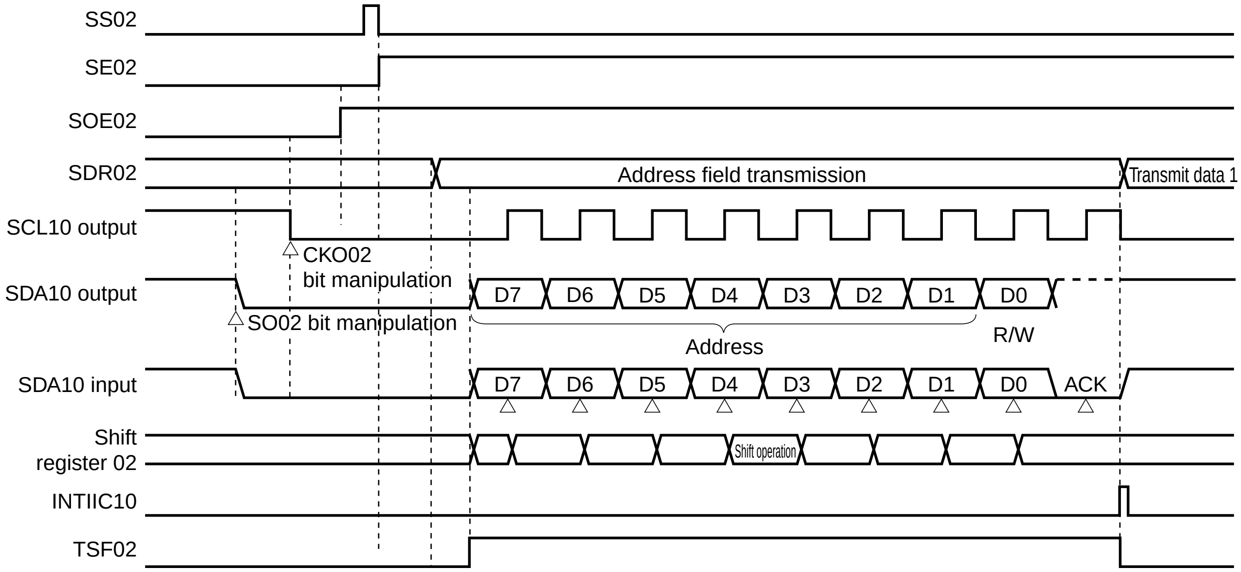
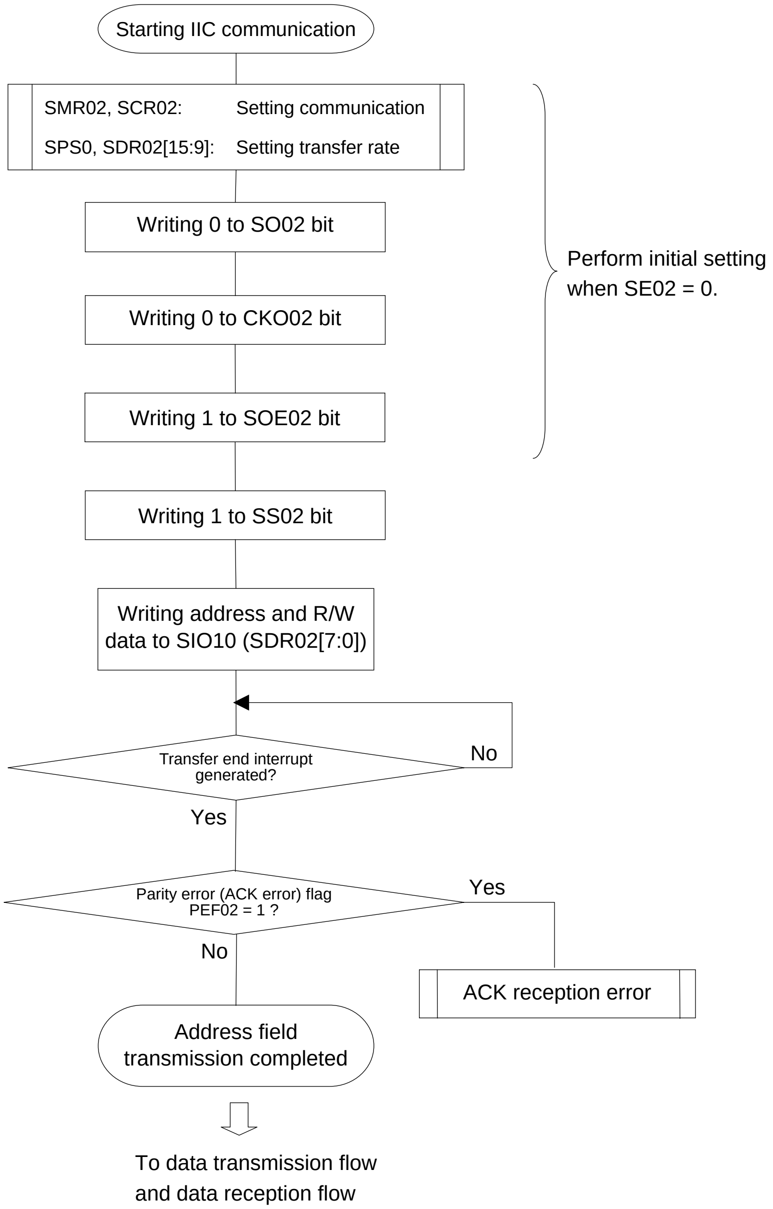


Figure 12-90. Flowchart of Address Field Transmission


Starting IIC communication
SMR02, SCR02: Setting communication
SPS0, SDR02[15:9]: Setting transfer rate
Writing 0 to SO02 bit
Writing 0 to CKO02 bit
Writing 1 to SOE02 bit
Perform initial setting when SE02 = 0.
Writing 1 to SS02 bit
Writing address and R/W data to SIO10 (SDR02[7:0])
Transfer end interrupt generated?
No
Yes
Parity error (ACK error) flag PEF02 = 1 ?
Yes
No
ACK reception error
Address field transmission completed
To data transmission flow and data reception flow

12.7.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC10
Target channel	Channel 2 of SAU
Pins used	SCL10, SDA10
Interrupt	INTIIC10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Parity error detection flag (PEF02)
Transfer data length	8 bits
Transfer rate	Max. f_{CLK}/4 MHz f_{CLK}: System clock frequency 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

(1) Register setting

Figure 12-91. Example of Contents of Registers for Data Transmission of Simplified I^{2}C (IIC10)

(a) Serial output register 0 (SO0) … 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
SO0						CKO02	CKO01	CKO00						SO02	SO01	SO00
	0	0	0	0	1	0/1 Note	×	×	0	0	0	0	1	0/1 Note	×	×

(b) Serial output enable register 0 (SOE0) … 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
SOE0														SOE02	SOE01	SOE00
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	×	×

(c) Serial channel start register 0 (SS0) … 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
SS0													SS03	SS02	SS01	SS00
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	×	×

(d) Serial mode register 02 (SMR02) … 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
SMR02	CKS02	CCS02						STS02		SIS020				MD022	MD021	MD020
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(e) Serial communication operation setting register 02 (SCR02) … Do not manipulate the bits of this register, except the TXE02 and RXE02 bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02	RXE02	DAP02	CKP02		EOC02	PTC021	PTC020	DIR02		SLC021	SLC020		DLS022	DLS021	DLS020
	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(f) Serial data register 02 (SDR02) (lower 8 bits: SIO10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02	Baud rate setting							0	Transmit data setting							
									SIO10							

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 12-92. Timing Chart of Data Transmission

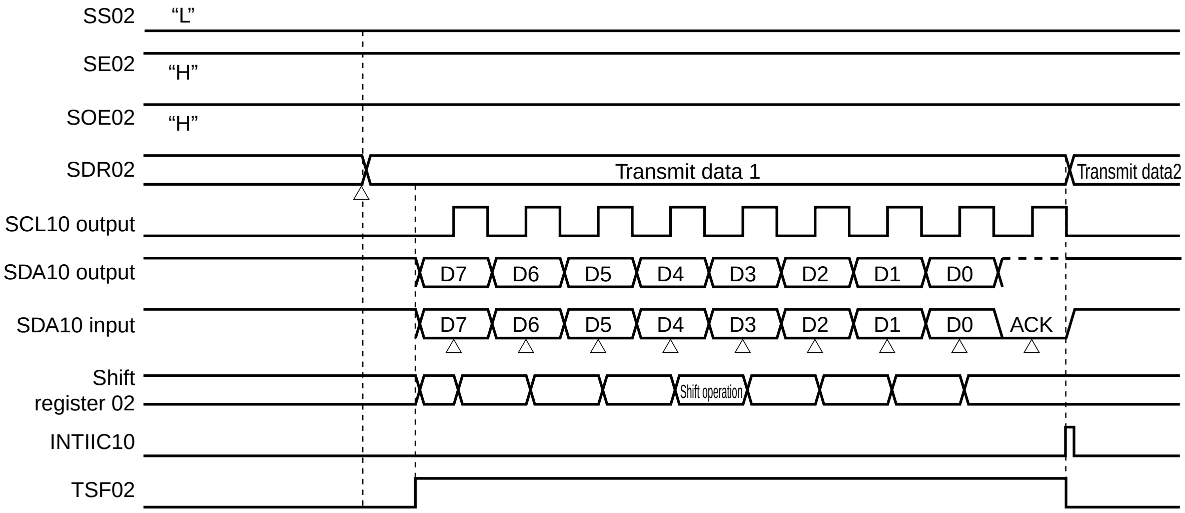
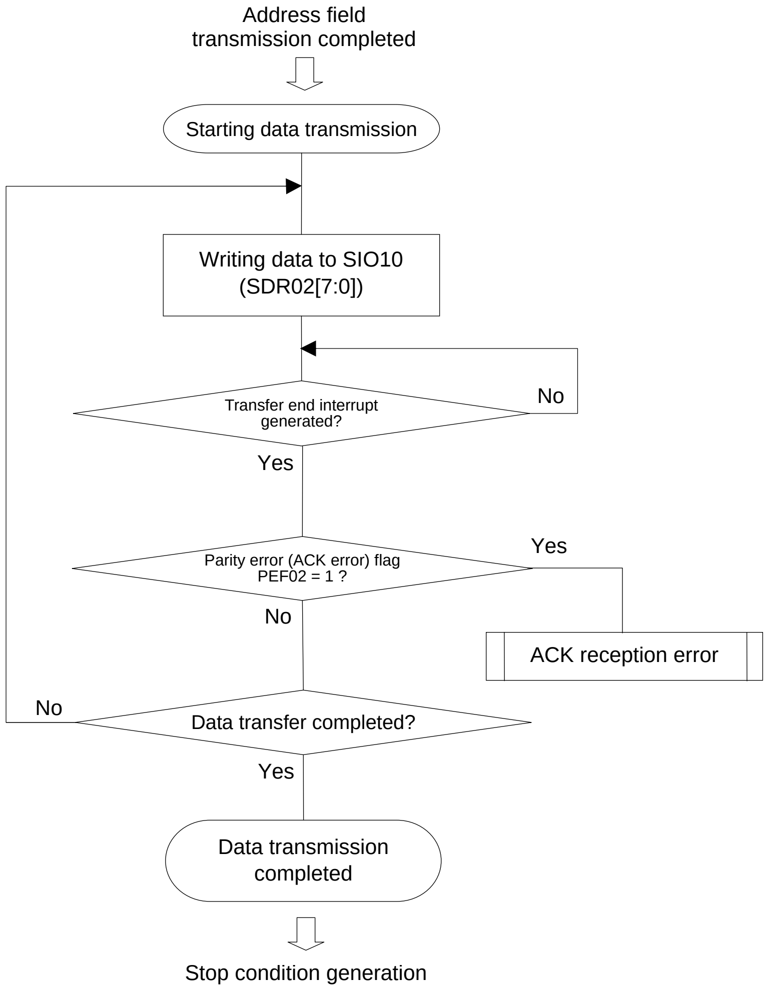


Figure 12-93. Flowchart of Data Transmission


Address field transmission completed
Starting data transmission
Writing data to SIO10 (SDR02[7:0])
Transfer end interrupt generated?
No
Yes
Parity error (ACK error) flag PEF02 = 1 ?
Yes
No
ACK reception error
No
Data transfer completed?
Yes
Data transmission completed
Stop condition generation

12.7.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC10
Target channel	Channel 2 of SAU
Pins used	SCL10, SDA10
Interrupt	INTIIC10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	None
Transfer data length	8 bits
Transfer rate	Max. f_{CLK}/4 MHz f_{CLK}: System clock frequency 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

(1) Register setting

Figure 12-94. Example of Contents of Registers for Data Reception of Simplified I^{2}C (IIC10)

(a) Serial output register 0 (SO0) … 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
SO0	0	0	0	0	1	CKO02 0/1[Note]	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 0/1[Note]	SO01 ×	SO00 ×

(b) Serial output enable register 0 (SOE0) … 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
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 1	SOE01 ×	SOE00 ×

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 ×

(d) Serial mode register 02 (SMR02) … 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
SMR02	CKS02 0/1	CCS02 0	0	0	0	0	0	STS02 0	0	SIS020 0	1	0	0	MD022 1	MD021 0	MD020 0

(e) Serial communication operation setting register 02 (SCR02) … Do not manipulate the bits of this register, except the TXE02 and RXE02 bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02 0	RXE02 1	DAP02 0	CKP02 0	0	EOC02 0	PTC021 0	PTC020 0	DIR02 0	0	SLC021 0	SLC020 1	0	DLS022 1	DLS021 1	DLS020 1

(f) Serial data register 02 (SDR02) (lower 8 bits: SIO10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02	Baud rate setting							0	Dummy transmit data setting (FFH)							
									SIO10							

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 12-95. Timing Chart of Data Reception

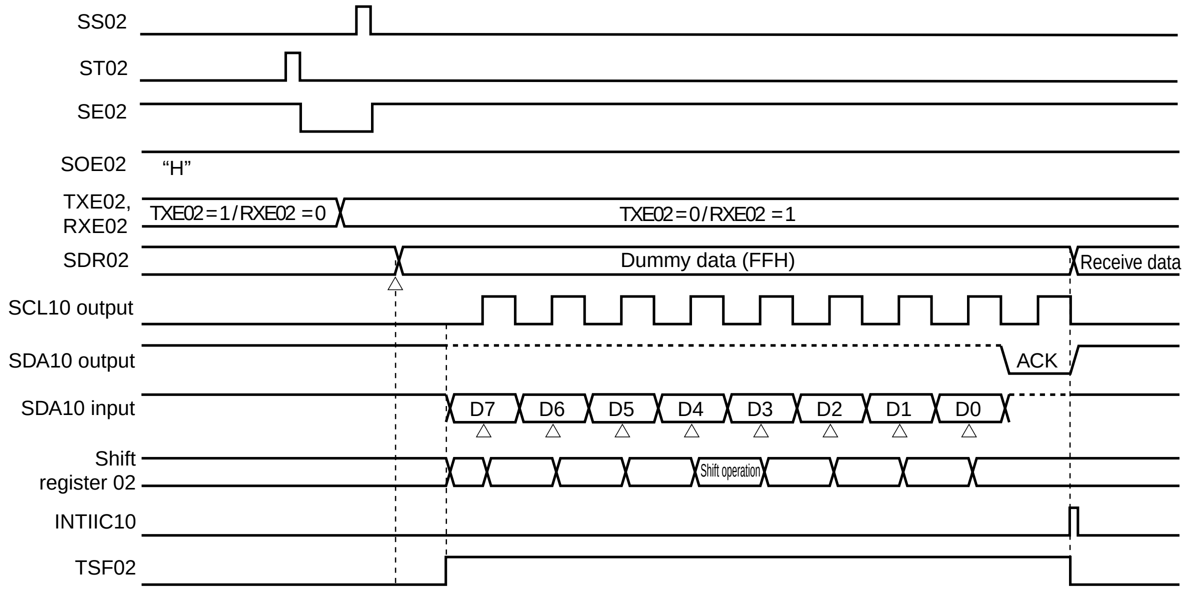
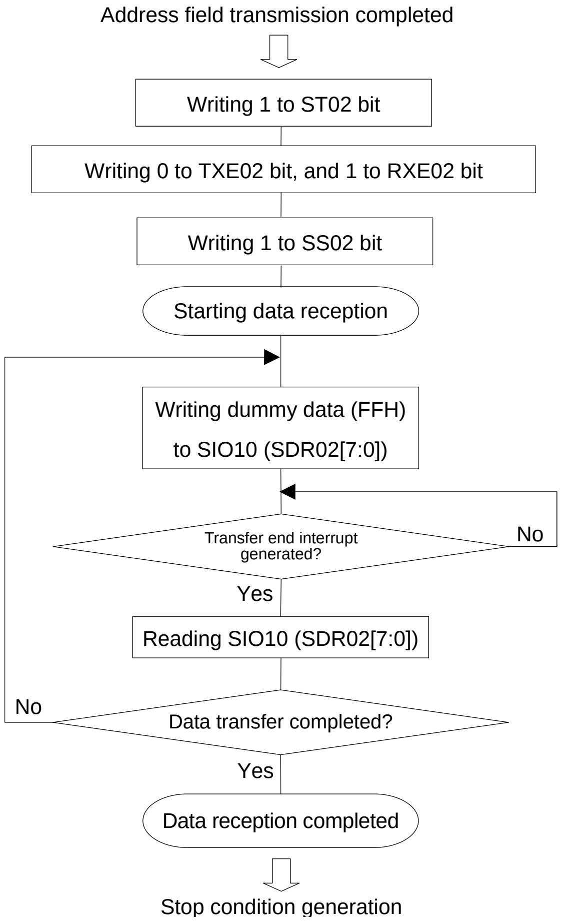


Figure 12-96. Flowchart of Data Reception



Caution ACK is also output when the last data is received. Communication is then completed by setting "1" to the ST02 bit to stop operation and generating a stop condition.

12.7.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

(1) Processing flow

Figure 12-97. Timing Chart of Stop Condition Generation

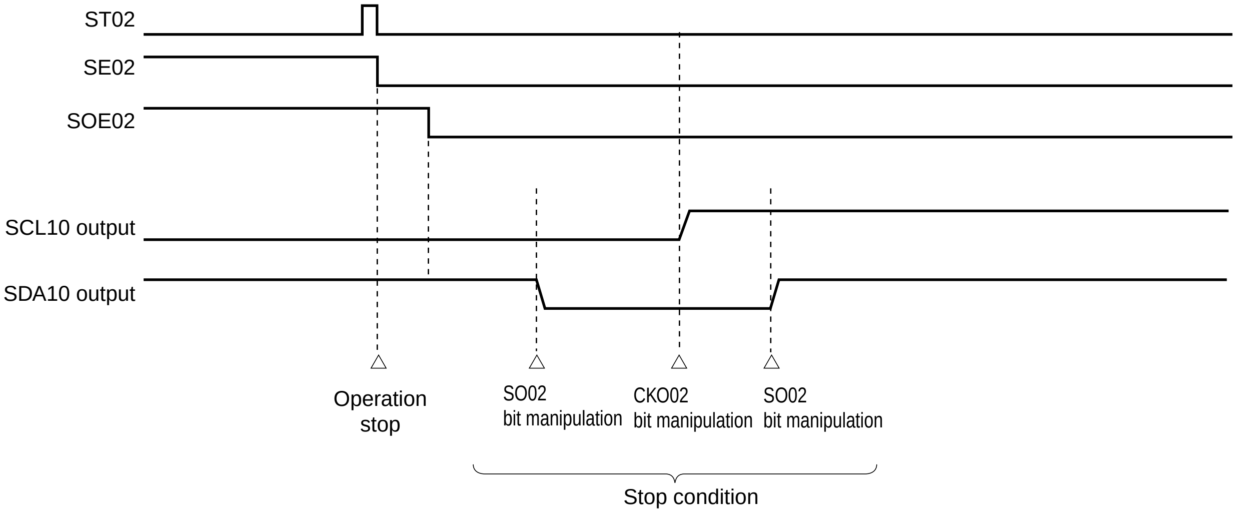
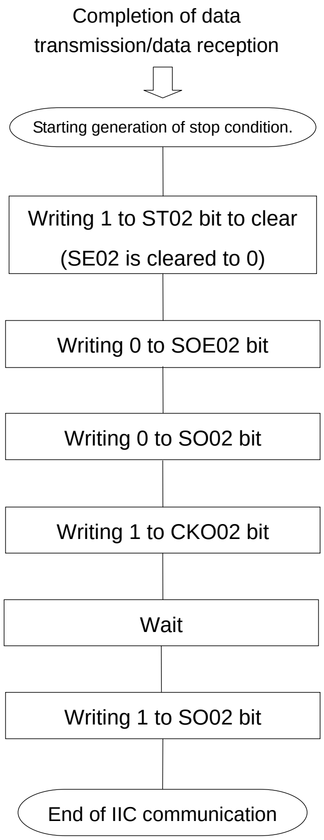


Figure 12-98. Flowchart of Stop Condition Generation


Completion of data transmission/data reception
Starting generation of stop condition.
Writing 1 to ST02 bit to clear (SE02 is cleared to 0)
Writing 0 to SOE02 bit
Writing 0 to SO02 bit
Writing 1 to CKO02 bit
Wait
Writing 1 to SO02 bit
End of IIC communication

Secure a wait time so that the specifications of I^2C on the slave side are satisfied.

12.7.5 Calculating transfer rate

The transfer rate for simplified I^{2}C (IIC10) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (MCK) frequency of target channel} ÷ (SDR02[15:9] + 1) ÷ 2

Remark The value of SDR02[15:9] is the value of bits 15 to 9 of the SDR02 register (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (MCK) is determined by serial clock select register 0 (SPS0) and bit 15 (CKS02) of serial mode register 02 (SMR02).

Table 12-4. Selection of operation clock

SMR02 Register	SPS0 Register								Operation Clock (MCK) [Note1]	
CKS02	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000		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[Note2]	
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[Note2]	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (ST0 = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 for the operation clock, also stop the timer array unit TAUS (TT0 = 00FFH).

2. SAU can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by setting the TIS02 bit of the TIS0 register of TAUS to 1, selecting $f_{SUB}/4$ for the input clock, and selecting INTTM02 using the SPS0 register. When changing f_{CLK}, however, SAU and TAUS must be stopped as described in Note 1 above.

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 (MCK)	SDR02[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%

12.8 Processing Procedure in Case of Error

The processing procedure to be followed if an error of each type occurs is described in Figures 12-99 to 12-101.

Figure 12-99. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads SDR0n register. →	BFF = 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 SSR0n register.		Error type is identified and the read value is used to clear error flag.
Writes SIR0n register. →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR0n register to the SIR0n register without modification.

Figure 12-100. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads SDR0n register. →	BFF = 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 SSR0n register.		Error type is identified and the read value is used to clear error flag.
Writes SIR0n register. →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR0n register to the SIR0n register without modification.
Sets ST0n bit to 1. →	SE0n = 0, and channel n stops operation.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets SS0n bit to 1. →	SE0n = 1, and channel n is enabled to operate.	

Remark n: Channel number (n = 0 to 3)

Figure 12-101. Processing Procedure in Case of Parity Error (ACK error) in Simplified I^{2}C Mode

Software Manipulation	Hardware Status	Remark
Reads SDR02 register.	BFF = 0, and channel 2 is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads SSR02 register.		Error type is identified and the read value is used to clear error flag.
Writes SIR02 register.	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR02 register to the SIR02 register without modification.
Sets ST02 bit to 1.	SE02 = 0, and channel 2 stops operation.	Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets SS02 bit to 1.	SE02 = 1, and channel 2 is enabled to operate.	

12.9 Relationship Between Register Settings and Pins

Tables 12-5 to 12-8 show the relationship between register settings and pins for each channel of serial array units 0 and 1.

Table 12-5. Relationship between register settings and pins (Channel 0: CSI00, UART0 transmission)

SE00 Note 1	MD002	MD001	SOE00	SO00	CKO00	TXE00	RXE00	PM75	P75	PM74 Note 2	P74 Note 2	PM73	P73	Operation mode	Pin Function		
															$\overline{\text{SCK00}}$/ TI11/P75	SI00/ RxD0/TI10/ P74 Note 2	SO00/ TxD0/TO10/ P73
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	TI11/P75	TI10/P74	TO10/P73
	0	1														RxD0/TI10/ P74	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI00 reception	$\overline{\text{SCK00}}$ (input)	SI00	TO10/P73
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI00 transmission	$\overline{\text{SCK00}}$ (input)	TI10/P74	SO00
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI00 transmission /reception	$\overline{\text{SCK00}}$ (input)	SI00	SO00
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI00 reception	$\overline{\text{SCK00}}$ (output)	SI00	TO10/P73
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI00 transmission	$\overline{\text{SCK00}}$ (output)	TI10/P74	SO00
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI00 transmission /reception	$\overline{\text{SCK00}}$ (output)	SI00	SO00
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART0 transmission Note 5	TI11/P75	RxD0/TI10/ P74	TxD0

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.

2. When channel 1 is set to UART0 reception, this pin becomes an RxD0 function pin (refer to **Table 12-6**). In this case, operation stop mode or UART0 transmission must be selected for channel 0.

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 **12.3 (12) Serial output register 0 (SO0)**.

5. When using UART0 transmission and reception in a pair, set channel 1 to UART0 reception (refer to **Table 12-6**).

Remark X: Don’t care

Table 12-6. Relationship between register settings and pins (Channel 1: CSI01, UART0 reception)

SE01 Note 1	MD012	MD011	SOE01	SO01	CKO01	TXE01	RXE01	PM72	P72	PM71	P71	PM70	P70	PM74 Note 2	P74 Note 2	Operation mode	Pin Function			
																	$\overline{\text{SCK01}}$/INTP6/P72	SI01/INTP5/P71	SO01/INTP4/P70	SI00/RxD0/TI10/P74 Note 2
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	INTP6/P72	INTP5/P71	INTP4/P70	SI00/TI10/P74
	0	1																		
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	× Note 3	× Note 3	Slave CSI01 reception	$\overline{\text{SCK01}}$ (input)	SI01	INTP4/P70	SI00/TI10/P74
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	× Note 3	× Note 3	Slave CSI01 transmission	$\overline{\text{SCK01}}$ (input)	INTP5/P71	SO01	SI00/TI10/P74
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	× Note 3	× Note 3	Slave CSI01 transmission /reception	$\overline{\text{SCK01}}$ (input)	SI01	SO01	SI00/TI10/P74
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	× Note 3	× Note 3	Master CSI01 reception	$\overline{\text{SCK01}}$ (output)	SI01	INTP4/P70	SI00/TI10/P74
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	× Note 3	× Note 3	Master CSI01 transmission	$\overline{\text{SCK01}}$ (output)	INTP5/P71	SO01	SI00/TI10/P74
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	× Note 3	× Note 3	Master CSI01 transmission /reception	$\overline{\text{SCK01}}$ (output)	SI01	SO01	SI00/TI10/P74
	0	1	0	1	1	0	1	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	1	×	UART0 reception Notes 5, 6	INTP6/P72	INTP5/P71	INTP4/P70	RxD0

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.

2. When channel 1 is set to UART0 reception, this pin becomes an RxD0 function pin. In this case, set channel 0 to operation stop mode or UART0 transmission (refer to **Table 12-5**).
When channel 0 is set to CSI00, this pin cannot be used as an RxD0 function pin. In this case, set channel 1 to operation stop mode or CSI01.

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 **12.3 (12) Serial output register 0 (SO0)**.

5. When using UART0 transmission and reception in a pair, set channel 0 to UART0 transmission (refer to **Table 12-5**).

6. The SMR00 register of channel 0 must also be set during UART0 reception. For details, refer to **12.5.2 (1) Register setting**.

Remark X: Don’t care

Table 12-7. Relationship between register settings and pins (Channel 2: CSI10, UART1 transmission, IIC10)

SE02 Note 1	MD022	MD021	SOE02	SO02	CKO02	TXE02	RXE02	PM32	P32	PM31 Note 2	P31 Note 2	PM30	P30	Operation mode	Pin Function		
															$\overline{\text{SCK10}}$/ SCL10/ INTP2/P32	SI10/SDA10/ RxD1/INTP1/ TI09/P31 Note 2	SO10/ TxD1/ TO11/P30
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	INTP2/P32	INTP1/TI09/P31	TO11/P30
	0	1														RxD1/INTP1/ TI09/P31	
	1	0														INTP1/TI09/P31	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI10 reception	$\overline{\text{SCK10}}$ (input)	SI10	TO11/P30
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI10 transmission	$\overline{\text{SCK10}}$ (input)	INTP1/TI09/P31	SO10
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI10 transmission /reception	$\overline{\text{SCK10}}$ (input)	SI10	SO10
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI10 reception	$\overline{\text{SCK10}}$ (output)	SI10	TO11/P30
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI10 transmission	$\overline{\text{SCK10}}$ (output)	INTP1/TI09/P31	SO10
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI10 transmission /reception	$\overline{\text{SCK10}}$ (output)	SI10	SO10
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART1 transmission Note5	INTP2/P32	RxD1/INTP1/ TI09/P31	TxD1
0	1	0	0	0/1 Note 6	0/1 Note 6	0 1 0	0 0 1	0	1	0	1	× Note 3	× Note 3	IIC10 start condition	SCL10	SDA10	TO11/P30
1			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC10 address field transmission	SCL10	SDA10	TO11/P30
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC10 data transmission	SCL10	SDA10	TO11/P30
			1	0/1 Note 4	0/1 Note 4	0	1	0	1	0	1	× Note 3	× Note 3	IIC10 data reception	SCL10	SDA10	TO11/P30
0			0	0/1 Note 7	0/1 Note 7	0 1 0	0 0 1	0	1	0	1	× Note 3	× Note 3	IIC10 stop condition	SCL10	SDA10	TO11/P30

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.

2. When channel 3 is set to UART1 reception, this pin becomes an RxD1 function pin (refer to **Table 12-8**). In this case, operation stop mode or UART1 transmission must be selected for channel 2.

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 **12.3 (12) Serial output register 0 (SO0)**.

5. When using UART1 transmission and reception in a pair, set channel 3 to UART1 reception (refer to **Table 12-8**).

6. Set the CKO02 bit to 1 before a start condition is generated. Clear the SO02 bit from 1 to 0 when the start condition is generated.

7. Set the CKO02 bit to 1 before a stop condition is generated. Clear the SO02 bit from 0 to 1 when the stop condition is generated.

Remark X: Don’t care

Table 12-8. Relationship between register settings and pins (Channel 3: UART1 reception)

SE03 Note 1	MD032	MD031	TXE03	RXE03	PM31 Note 2	P31 Note 2	Operation mode	Pin Function
								SI10/SDA10/RxD1/INTP1/TI09/P31 Note 2
0	0	1	0	0	× Note 3	× Note 3	Operation stop mode	SI10/SDA10/INTP1/TI09/P31 Note 2
1	0	1	0	1	1	×	UART1 reception Notes 4, 5	RxD1

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.

2. When channel 3 is set to UART1 reception, this pin becomes an RxD1 function pin. In this case, set channel 2 to operation stop mode or UART1 transmission (refer to **Table 12-7**).

When channel 2 is set to CSI10 or IIC10, this pin cannot be used as an RxD1 function pin. In this case, set channel 3 to operation stop mode.

3. This pin can be set as a port function pin.

4. When using UART1 transmission and reception in a pair, set channel 2 to UART1 transmission (refer to **Table 12-7**).

5. The SMR02 register of channel 2 must also be set during UART1 reception. For details, refer to **12.5.2 (1) Register setting**.

Remark X: Don’t care

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

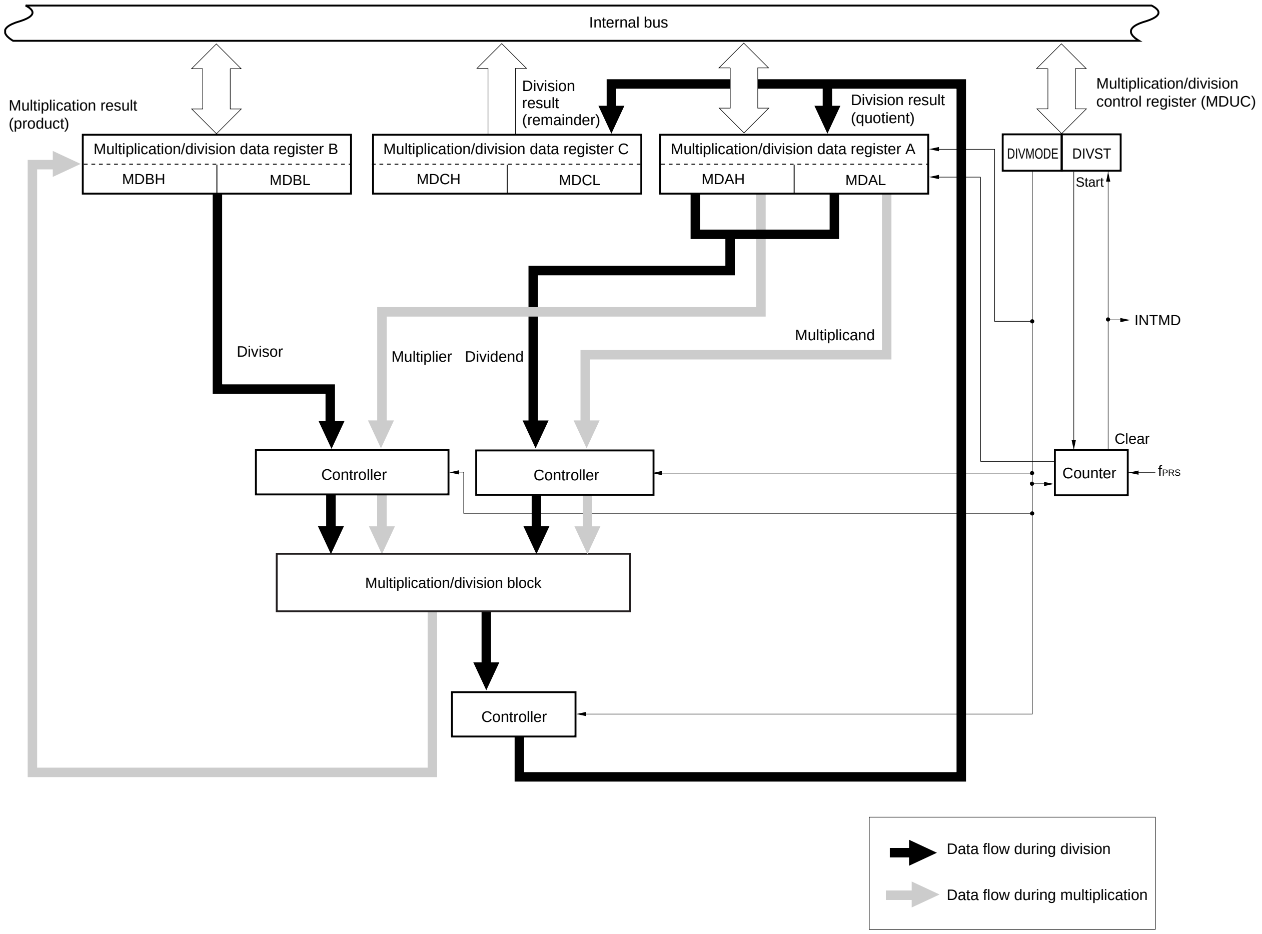
The multiplier/divider consists of the following hardware.

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/W

Symbol	F00E3H								F00E2H							
MDCH	MDCH 15	MDCH 14	MDCH 13	MDCH 12	MDCH 11	MDCH 10	MDCH 9	MDCH 8	MDCH 7	MDCH 6	MDCH 5	MDCH 4	MDCH 3	MDCH 2	MDCH 1	MDCH 0

Symbol	F00E1H								F00E0H							
MDCL	MDCL 15	MDCL 14	MDCL 13	MDCL 12	MDCL 11	MDCL 10	MDCL 9	MDCL 8	MDCL 7	MDCL 6	MDCL 5	MDCL 4	MDCL 3	MDCL 2	MDCL 1	MDCL 0

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>.)

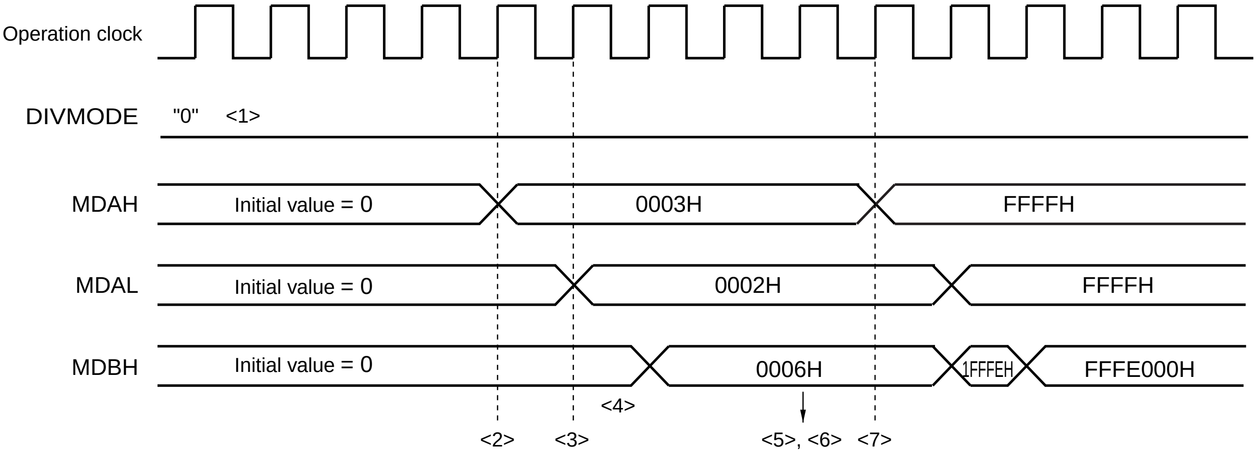
- Next operation

  <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

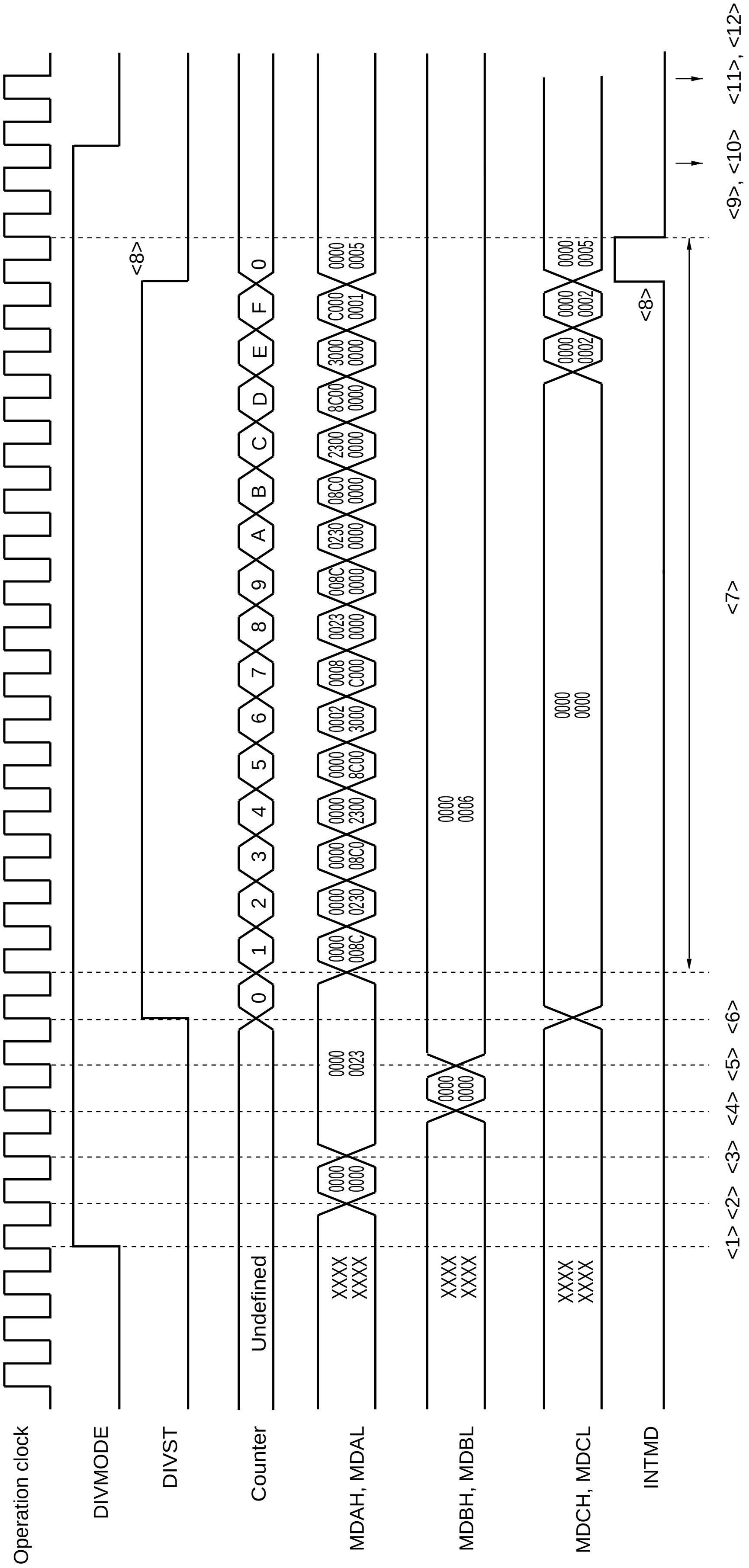
  <8> DIVST is cleared (0) and an interrupt request signal (INTMD) is generated (end of operation).
  <9> Read the quotient (lower 16 bits) from MDAL.
  <10> Read the quotient (higher 16 bits) from MDAH.
  <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  <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 μPD79F9211 has an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between the peripheral hardware supporting DMA, SFRs, and internal RAM without via CPU.

As a result, the normal internal operation of the CPU and data transfer can be executed in parallel with transfer between the SFR and internal RAM, and therefore, a large capacity of data can be processed. In addition, real-time control using communication, timer, and A/D can also be realized.

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 (CSI00, CSI01, CSI10, UART0, UART1, or IIC10)
- 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 register 0, 1 (DRC0, DRC1)

(1) DMA SFR address register n (DSAn)

This is an 8-bit register that is used to set an SFR address that is the transfer source or destination of DMA channel n.

Set the lower 8 bits of the SFR addresses FFF00H to FFFFFH[Note].

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.

Note Except for address FFFFEH because the PMC register is allocated there.

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 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.

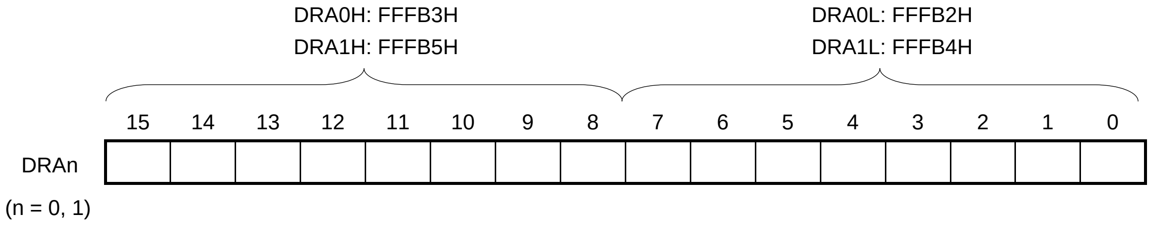
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

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



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 (bits 15 to 8) DBC0L: FFFB6H
DBC1L: FFFB8H (bits 7 to 0)

	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 to 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]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).

DMA transfer is started 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	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.

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

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 stat 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	Timer channel 0 interrupt
0	0	1	1	INTTM01	Timer channel 1 interrupt
0	1	0	0	INTTM04	Timer channel 4 interrupt
0	1	0	1	INTTM05	Timer channel 5 interrupt
0	1	1	0	INTST0/INTCSI00	UART0 transmission end interrupt/CSI00 transfer end interrupt
0	1	1	1	INTSR0/INTCSI01	UART0 reception end interrupt/CSI01 transfer end interrupt
1	0	0	0	INTST1/INTCSI10/INTIIC10	UART1 transmission end interrupt/ CSI10 transfer end interrupt/ IIC10 transfer end interrupt
1	0	0	1	INTSR1	UART1 reception 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.
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1).	

DSTn	DMA transfer mode flag
0	DMA transfer of DMA channel n is completed.
1	DMA transfer of DMA channel n is not completed (still under execution).
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by 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 14.5.5 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, CBCn, and DMCn registers.

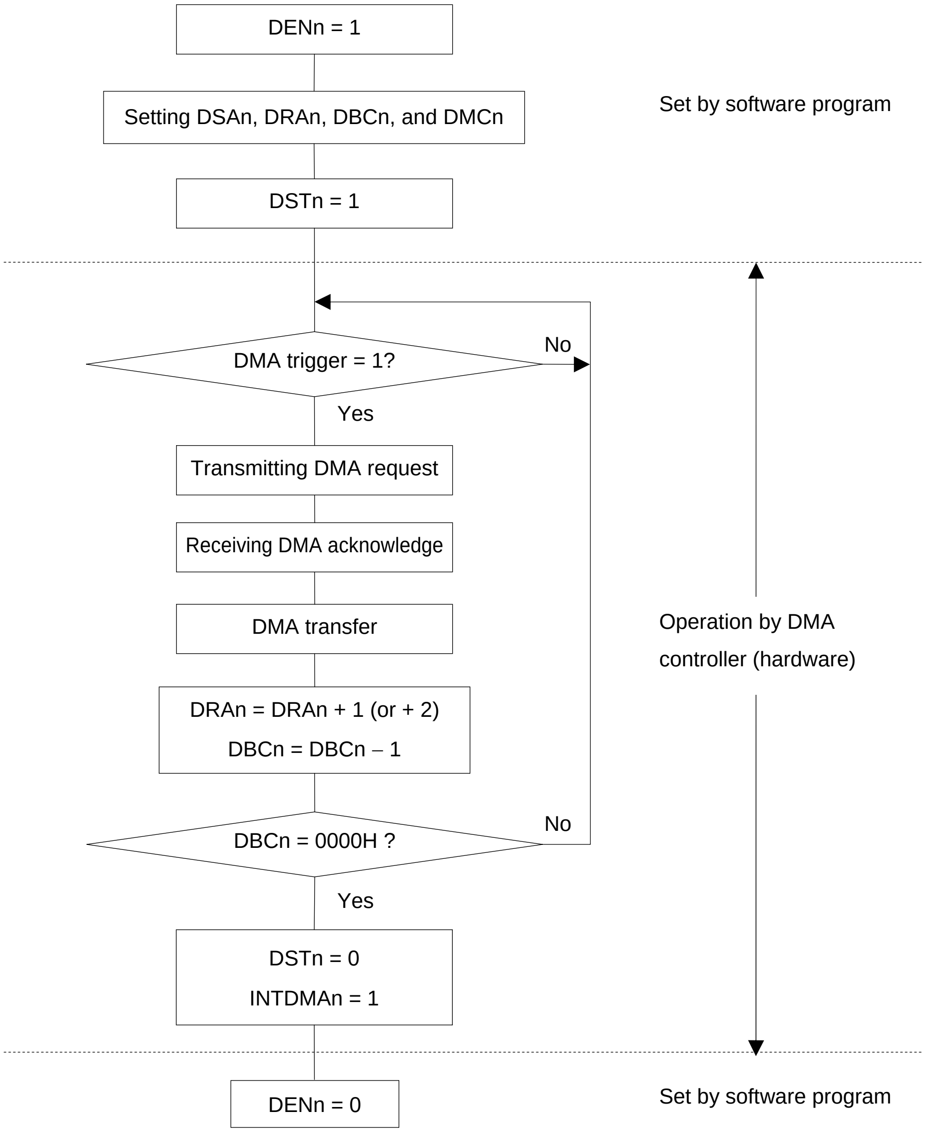
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.

<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)

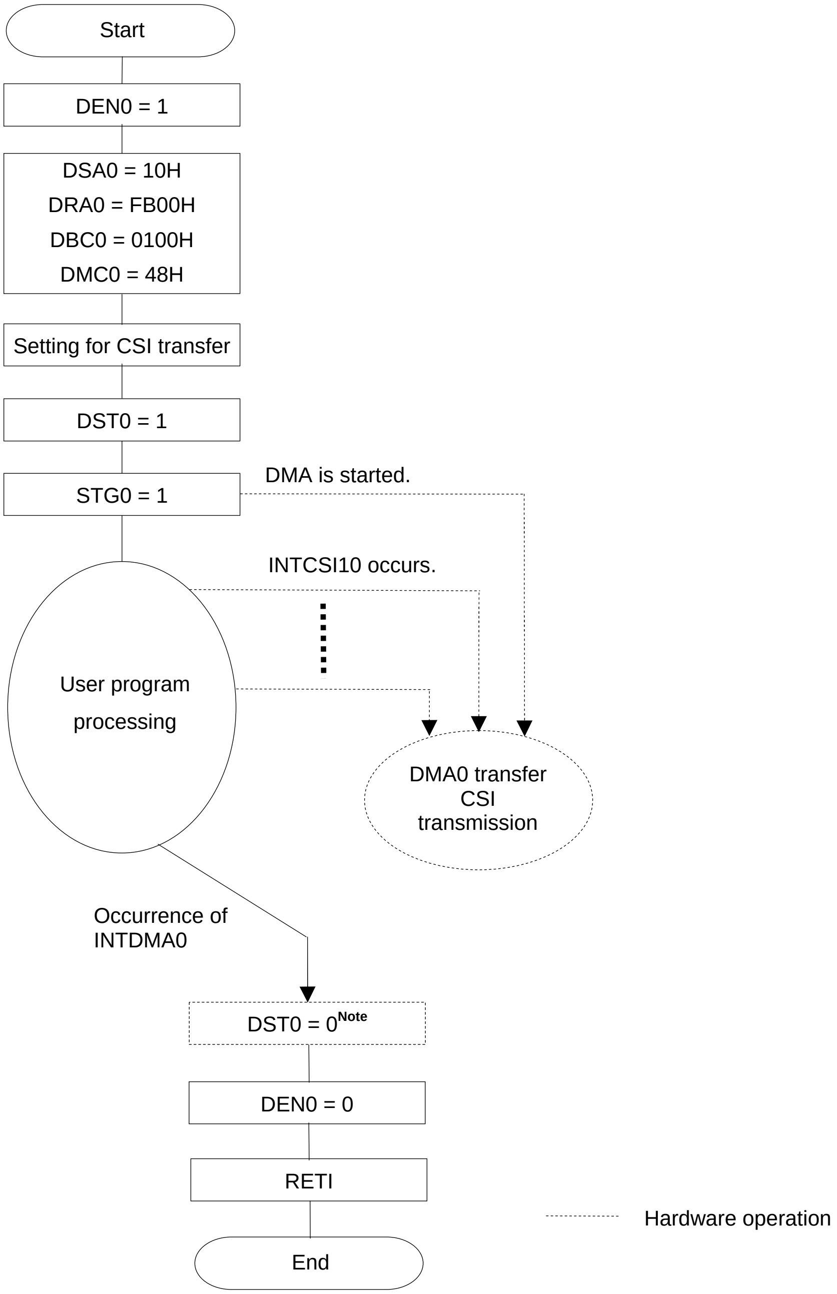
14.5 Example of Setting of DMA Controller

14.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission of CSI10
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI10 (software trigger (STG0) only for the first start source)
- Interrupt of CSI10 is specified by IFC03 to IFC00 (bits 3 to 0 of the DMC0 register) = 1000B.
- Transfers FFB00H to FFBFFH (256 bytes) of RAM to FFF10H of the transmit buffer (SIO10) of CSI.

Figure 14-7. Example of Setting for CSI Consecutive Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set DST0 to 0 and then DEN0 to 0 (for details, refer to **14.5.5 Forced termination by software**).

The fist trigger for consecutive transmission is not started by the interrupt of CSI. Start it by a software trigger.

CSI transmission of the second time and onward is automatically executed.

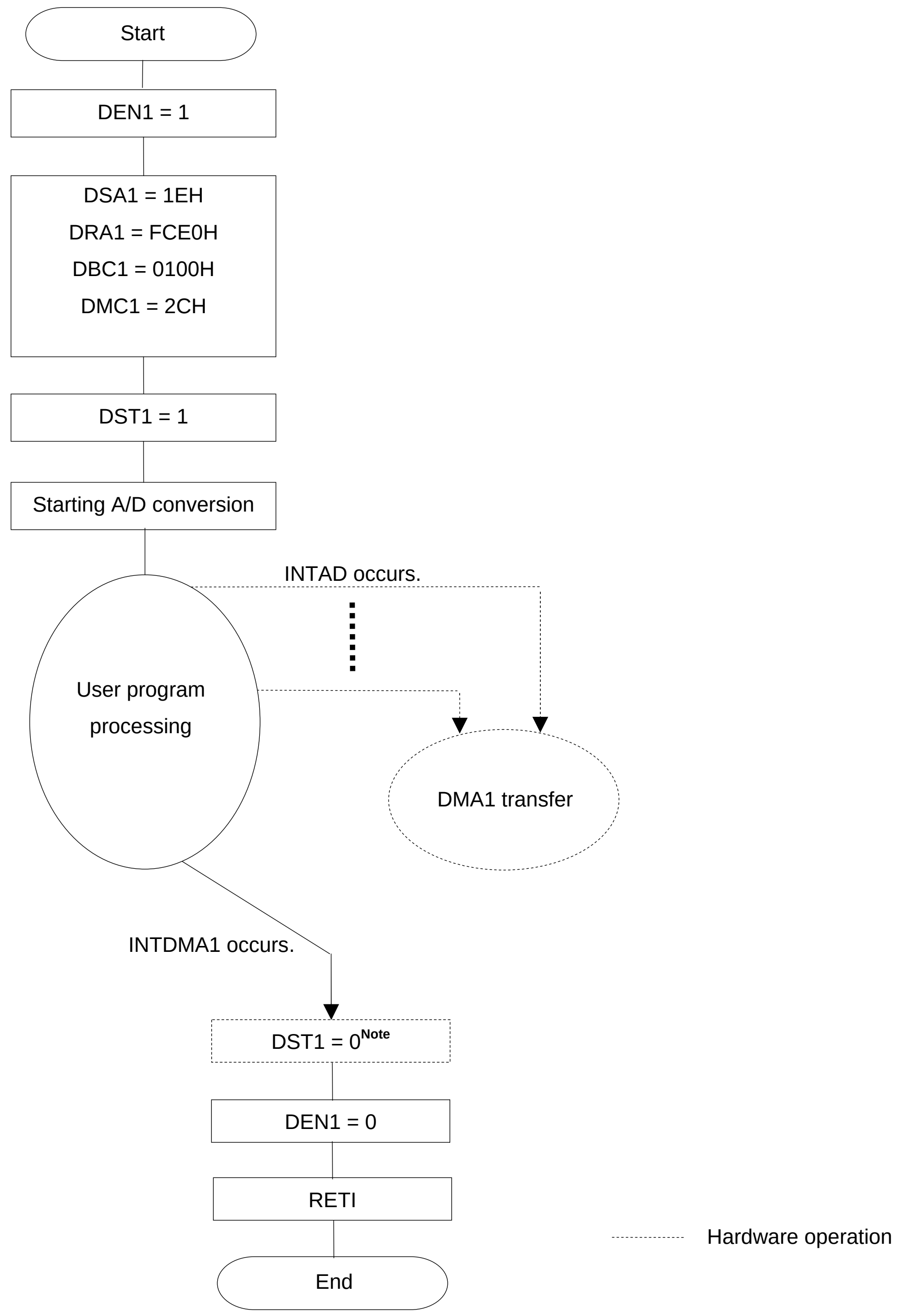
The DMA interrupt (INTDMA0) is generated as soon as the last data has been written to the transmit buffer. At this point, the last data of CSI is being transmitted. To start DMA transfer again, therefore, wait until transfer of CSI is completed.

14.5.2 Consecutive capturing of A/D conversion results

A flowchart of an example of setting for consecutively capturing A/D conversion results is shown below.

- Consecutive capturing of A/D conversion results.
- DMA channel 1 is used for DMA transfer.
- DMA start source: INTAD
- Interrupt of A/D is specified by IFC13 to IFC10 (bits 3 to 0 of the DMC1 register) = 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.

Figure 14-8. Example of Setting of Consecutively Capturing A/D Conversion Results



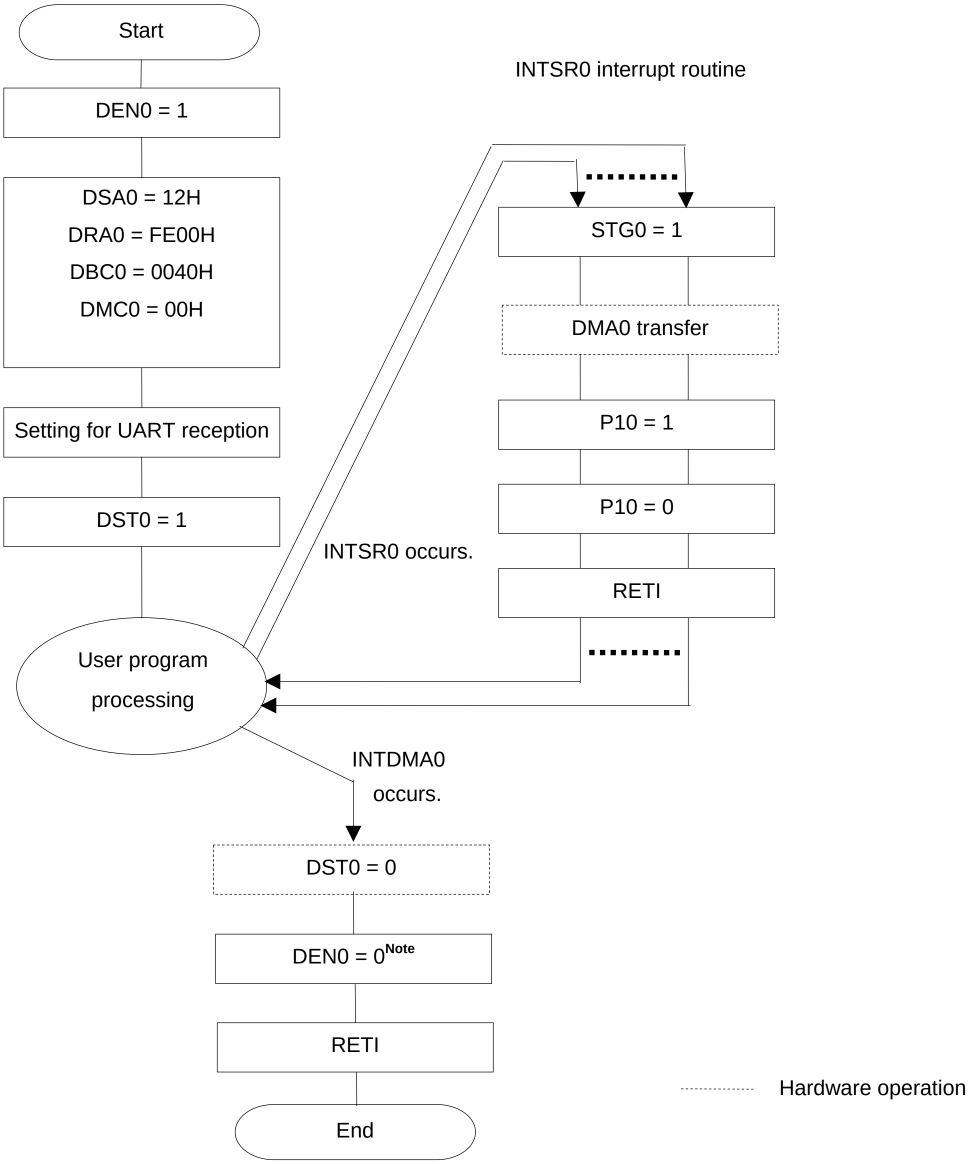
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set DST1 to 0 and then DEN1 to 0 (for details, refer to **14.5.5 Forced termination by software**).

14.5.3 UART consecutive reception + ACK transmission

A flowchart illustrating an example of setting for UART consecutive reception + ACK transmission is shown below.

- Consecutively receives data from UART0 and outputs ACK to P10 on completion of reception.
- DMA channel 0 is used for DMA transfer.
- DMA start source: Software trigger (DMA transfer on occurrence of an interrupt is disabled.)
- Transfers FFF12H of UART receive data register 0 (RXD0) to 64 bytes of FFE00H to FFE3FH of RAM.

Figure 14-9. Example of Setting for UART Consecutive Reception + ACK Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set DST0 to 0 and then DEN0 to 0 (for details, refer to **14.5.5 Forced termination by software**).

Remark This is an example where a software trigger is used as a DMA start source.
If ACK is not transmitted and if only data is consecutively received from UART, the UART reception end interrupt (INTSR0) can be used to start DMA for data reception.

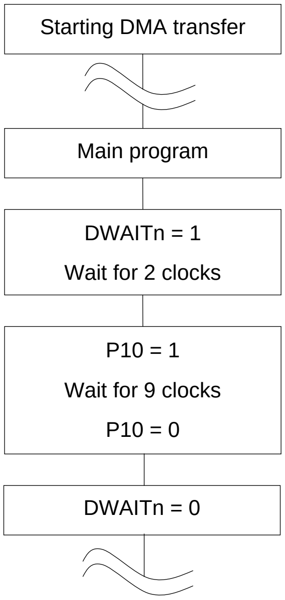
14.5.4 Holding DMA transfer pending by DWAITn

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.

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 14-10. Example of Setting for Holding DMA Transfer Pending by DWAITn



Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

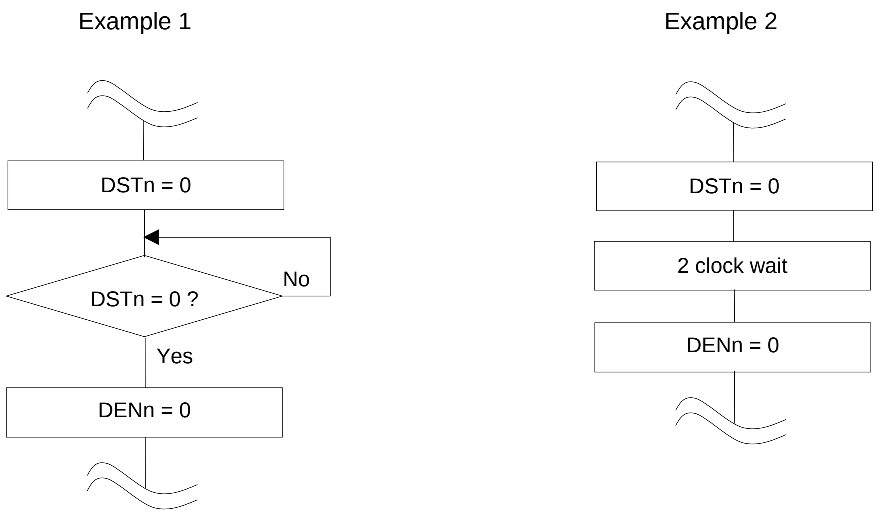
14.5.5 Forced termination by software

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.

- 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.

Figure 14-11. Forced Termination of DMA Transfer



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. When the requests from either of the DMA channels are successively generated in a short period [Note], they are successively transferred, and on completion of that, the requests from the other DMA channel are executed. In this case, one or tow instructions are executed between the first DMA transfer and next DMA transfer.

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.

Note The short period refers to a period of eight or fewer CPU clocks. The relationship between the lengths of clock period and DMA operations is as follows.

1 clock period: Setting disabled DMA request cannot be accepted.

2 to 4 clock period: DMA transfer of the channel where requests are successively generated is executed.

5 to 8 clock period: Whether DMA transfer of the channel where requests are successively generated is executed or DMA requests from the other channel are executed depends on the number of times CPU instructions are 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	4 clocks	10 clocks

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

In the following cases, however, DMA transfer may be delayed further. The number of clocks by which DMA transfer is delayed differs depending on the condition.

- Instruction execution by RAM
- Execution of DMA pending instruction

(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 &!addr16
- 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, and 8-bit manipulation instructions with operands including ES registers

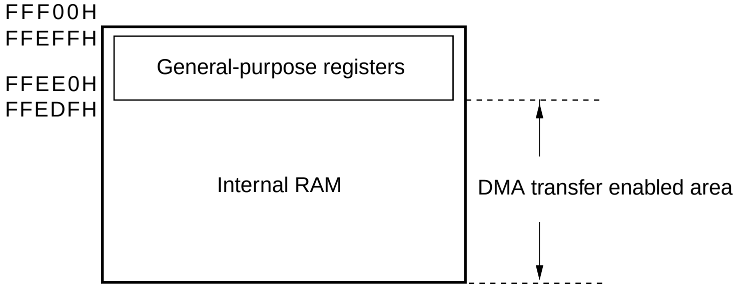
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

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: 8, internal: 33.

(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 μPD79F9211 has a total of 43 interrupt sources including maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Table 15-1**).

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			000AH	
	4	INTP2			000CH	
	5	INTP3/ INTTMOFF0	Pin input edge detection/timer Hi-Z control interrupt 0		000EH	
	6	INTP4	Pin input edge detection		0010H	
	7	INTP5			0012H	
	8	INTTMAD	A/D conversion timer trigger	Internal	0014H	(A)
	9	INTCMP0	CMP0 detection		0016H	
	10	INTCMP1	CMP1 detection		0018H	
	11	INTDMA0	End of DMA0 transfer		001AH	
	12	INTDMA1	End of DMA1 transfer		001CH	
	13	INTST0/ INTCSI00	End of UART0 transmission/ end of CSI00 communication		001EH	
	14	INTSR0/ INTCSI01	End of UART0 reception/ end of CSI01 communication		0020H	
	15	INTSRE0	UART0 communication error occurrence		0022H	
	16	INTST1 /INTCSI10 /INTIIC10	End of UART1 transmission/ end of CSI10 communication/ end of IIC10 communication		0024H	
	17	INTSR1	End of UART1 reception		0026H	
	18	INTSRE1	UART1 communication error occurrence		0028H	
	19	INTTM00	End of timer channel 0 count or capture		002CH	
	20	INTTM01	End of timer channel 1 count or capture		002EH	
	21	INTTM02	End of timer channel 2 count or capture		0030H	
	22	INTTM03	End of timer channel 3 count or capture		0032H	
	23	INTAD	End of A/D conversion		0034H	

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 41 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.

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	24	INTRTC	Fixed-cycle signal of real-time counter/alarm match detection	Internal	0036H	(A)
	25	INTRTCI	Interval signal detection of real-time counter		0038H	
	26	INTTMM0	Timer array unit mountain interrupt signal detection 0		003CH	
	27	INTTMV0	Timer array unit valley interrupt signal detection 0		003EH	
	28	INTMD	End of division operation		0040H	
	29	INTTM04	End of timer channel 4 count or capture		0042H	
	30	INTTM05	End of timer channel 5 count or capture		0044H	
	31	INTTM06	End of timer channel 6 count or capture		0046H	
	32	INTTM07	End of timer channel 7 count or capture		0048H	
	33	INTP6	Pin input edge detection	External	004AH	(B)
	34	INTP7/ INTTMOFF1	Pin input edge detection/timer Hi-Z control interrupt 0		004CH	
	35	INTTMM1	Timer array unit mountain interrupt signal detection 1	Internal	004EH	(A)
	36	INTTMV1	Timer array unit valley interrupt signal detection 1		0050H	
	37	INTTM08	End of timer channel 8 count or capture		0052H	
	38	INTTM09	End of timer channel 9 count or capture		0054H	
	39	INTTM10	End of timer channel 10 count or capture		0056H	
	40	INTTM11	End of timer channel 11 count or capture		0058H	
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 41 indicates the lowest priority.

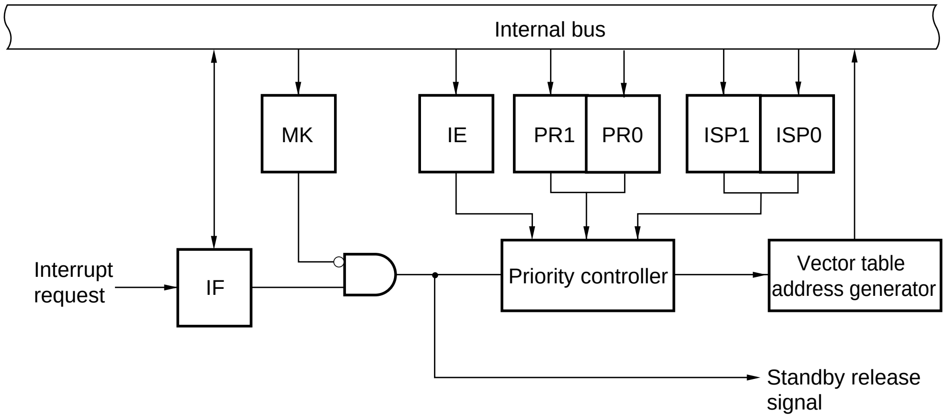
2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 15-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

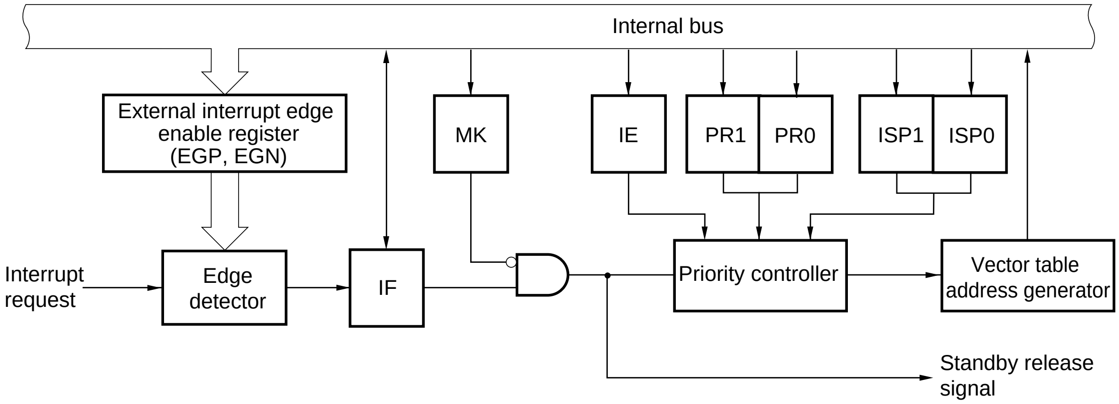
4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 15-1. Basic Configuration of Interrupt Function

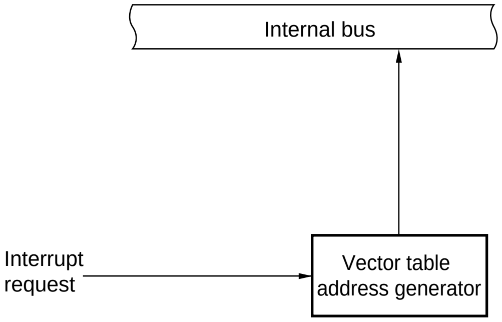
(A) Internal maskable interrupt



(B) External maskable interrupt



(C) Software interrupt



IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP0:	In-service priority flag 0
ISP1:	In-service priority flag 1
MK:	Interrupt mask flag
PR0:	Priority specification flag 0
PR1:	Priority specification flag 1

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	
INTP3[Note 1]	PIF3[Note 1]		PMK3[Note 1]		PPR03, PPR13[Note 1]	
INTTMOFF0[Note 1]	TMOFFIF0[Note 1]		TMOFFMK0[Note 1]		TMOFFPR00, TMOFFPR10[Note 1]	
INTP4	PIF4		PMK4		PPR04, PPR14	
INTP5	PIF5		PMK5		PPR05, PPR15	
INTTMAD	TMADIF	IF0H	TMADMK	MK0H	TMADPR0, TMADPR1	PR00H, PR10H
INTCMP0	CMPIF0		CMPMK0		CMPPR00, CMPPR10	
INTCMP1	CMPIF1		CMPMK1		CMPPR01, CMPPR11	
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
INTST0 [Note 2]	STIF0 [Note 2]		STMK0 [Note 2]		STPR00, STPR10 [Note 2]	
INTCSI00 [Note 2]	CSIIF00 [Note 2]		CSIMK00 [Note 2]		CSIPR000, CSIPR100 [Note 2]	
INTSR0 [Note 3]	SRIF0 [Note 3]		SRMK0 [Note 3]		SRPR00, SRPR10 [Note 3]	
INTCSI01 [Note 3]	CSIIF01 [Note 3]		CSIMK01 [Note 3]		CSIPR001, CSIPR101 [Note 3]	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	

Notes 1. Do not use INTP3 and INTMOFF0 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTP3 and INTMOFF0 is generated, bit 5 of IF0L is set to 1. Bit 5 of MK0L, PR00L, and PR10L supports these two interrupt sources.

2. Do not use UART0 and CSI00 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INST0 and INTCSI00 is generated, bit 5 of IF0H is set to 1. Bit 5 of MK0H, PR00H, and PR10H supports these two interrupt sources.

3. Do not use UART0 and CSI01 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INSR0 and INTCSI01 is generated, bit 6 of IF0H is set to 1. Bit 6 of MK0H, PR00H, and PR10H supports these two interrupt sources.

Table 15-2. Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTST1[Note 1]	STIF1[Note 1]	IF1L	STMK1[Note 1]	MK1L	STPR01, STPR11[Note 1]	PR01L, PR11L
INTCSI10[Note 1]	CSIIF10[Note 1]		CSIMK10[Note 1]		CSIPR010, CSIPR110[Note 1]	
INTIIC10[Note 1]	IICIF10[Note 1]		IICMK10[Note 1]		IICPR010, IICPR110 [Note 1]	
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11	
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1	
INTRTCI	RTCIIF		RTCIMK		RTCIPR0, RTCIPR1	
INTTMM0	TMMIF0		TMMMK0		TMMPR00, TMMPR10	
INTTMV0	TMVIF0		TMVMK0		TMVPR00, TMVPR10	
INTMD	MDIF		MDMK		MDPR0, MDPR1	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTP6	PIF6		PMK6		PPR06, PPR16	
INTP7 [Note 2]	PIF7 [Note 2]		PMK7 [Note 2]		PPR07, PPR17 [Note 2]	
INTTMOFF1 [Note 2]	TMOFFIF1 [Note 2]		TMOFFMK1 [Note 2]		TMOFFPR01, TMOFFPR11 [Note 2]	
INTTMM1	TMMIF1		TMMMK1		TMMPR01, TMMPR11	
INTTMV1	TMVIF1		TMVMK1		TMVPR01, TMVPR11	
INTTM08	TMIF08		TMMK08		TMPR008, TMPR108	
INTTM09	TMIF09	IF2H	TMMK09	MK2H	TMPR009, TMPR109	PR02H, PR12H
INTTM10	TMIF10		TMMK10		TMPR010, TMPR110	
INTTM11	TMIF11		TMMK11		TMPR011, TMPR111	

Notes 1. Do not use UART1, CSI10, and IIC10 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST1, INTCSI10, and INTIIC10 is generated, bit 0 of IF1L is set to 1. Bit 0 of MK1L, PR01L, and PR11L supports these three interrupt sources.

2. Do not use INTP7 and INTMOFF1 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTP7 and INTMOFF1 is generated, bit 4 of IF2L is set to 1. Bit 4 of MK2L, PR02L, and PR12L supports these two interrupt sources.

(1) Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

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) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3 TMOFFIF0	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0	SRIF0 CSIIF01	STIF0 CSIIF00	DMAIF1	DMAIF0	CMPIF1	CMPIF0	TMADIF

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	0	SREIF1	SRIF1	STIF1 CSIIF10 IICIF10

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
IF1H	TMIF04	MDIF	TMVIF0	TMMIF0	0	RTCIIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	TMIF08	TMVIF1	TMMIF1	PIF7 TMOFFIF1	PIF6	TMIF07	TMIF06	TMIF05

Figure 15-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
IF2H	0	0	0	0	0	TMIF11	TMIF10	TMIF09

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. Be sure to clear bit 3 of IF1H, and bits 3 to 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	PMK3 TMOFFMK0	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0	SRMK0 CSIMK01	STMK0 CSIMK00	DMAMK1	DMAMK0	CMPMK1	CMPMK0	TMADMK

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	1	SREMK1	SRMK1	STMK1 CSIMK10 IICMK10

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
MK1H	TMMK04	MDMK	TMVMK0	TMMMK0	1	RTCIMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	TMMK08	TMVMK1	TMMMK1	PMK7 TMOFFMK1	PMK6	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
MK2H	1	1	1	1	1	TMMK11	TMMK10	TMMK09

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bit 3 of MK1H, and bits 3 to 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) (1/3)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03 TMOFFPR00	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13 TMOFFPR10	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00	SRPR00 CSIPR001	STPR00 CSIPR000	DMAPR01	DMAPR00	CMPPR01	CMPPR00	TMADPR0

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	SRPR10 CSIPR101	STPR10 CSIPR100	DMAPR11	DMAPR10	CMPPR11	CMPPR10	TMADPR1

Figure 15-4. Format of Priority Specification Flag Registers
(PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/3)

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	1	SREPR01	SRPR01	STPR01 CSIPR010 IICPR010

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	1	SREPR11	SRPR11	STPR11 CSIPR110 IICPR110

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR01H	TMPR004	MDPR0	TMVPR00	TMMPR00	1	RTCIPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR11H	TMPR104	MDPR1	TMVPR10	TMMPR10	1	RTCIPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	TMPR008	TMVPR01	TMMPR01	PPR07 TMOFFPR01	PPR06	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	TMPR108	TMVPR11	TMMPR11	PPR17 TMOFFPR11	PPR16	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR02H	1	1	1	1	1	TMPR011	TMPR010	TMPR009

Address: FFFDDH After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR12H	1	1	1	1	1	TMPR111	TMPR110	TMPR109

Figure 15-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (3/3)

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 3 of PR01H and PR11H, and bits 3 to 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 to INTP7.

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	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
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 Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P31	INTP1
EGP2	EGN2	P32	INTP2
EGP3	EGN3	P80	INTP3
EGP4	EGN4	P70	INTP4
EGP5	EGN5	P71	INTP5
EGP6	EGN6	P72	INTP6
EGP7	EGN7	P82	INTP7

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 to 7

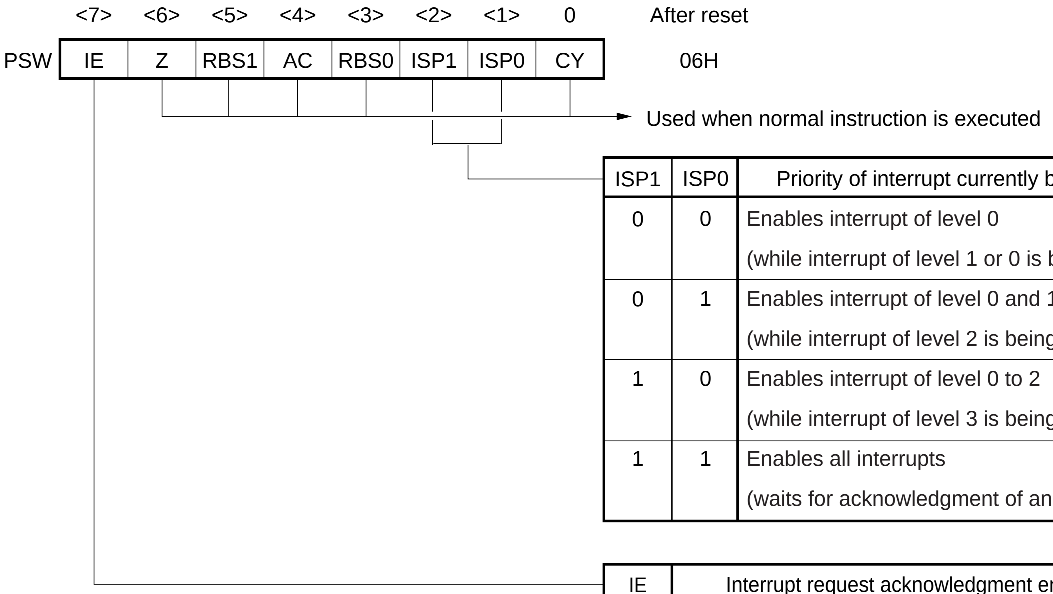
(5) Program status word (PSW)

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.

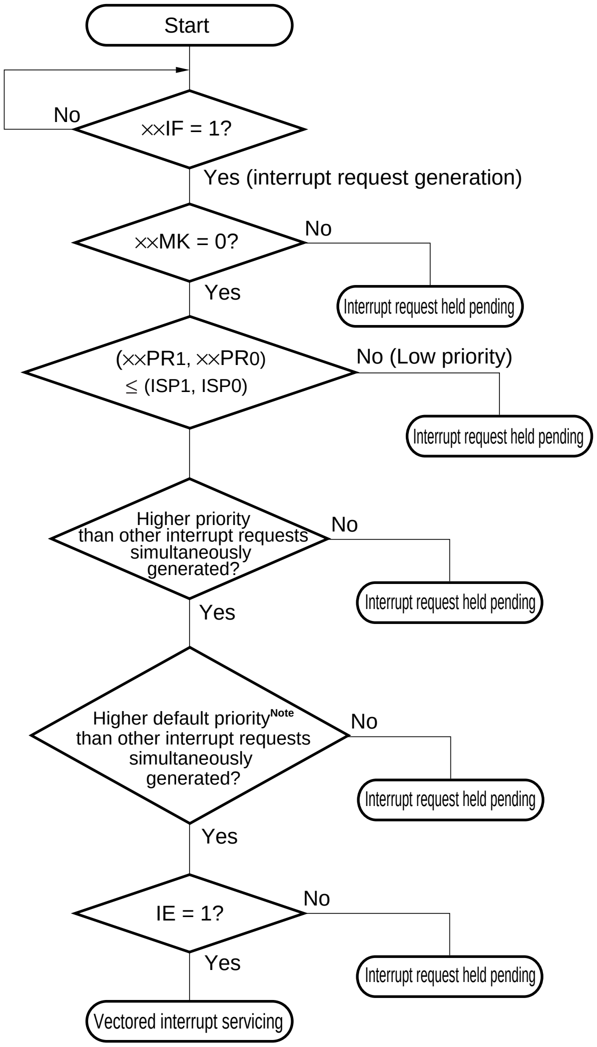
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

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.

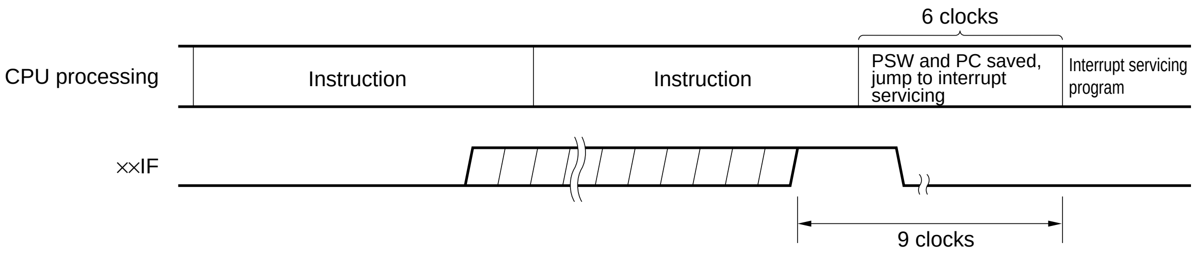
Figure 15-7. Interrupt Request Acknowledgment Processing Algorithm


Start
××IF = 1?
No
Yes (interrupt request generation)
××MK = 0?
No
Yes
Interrupt request held pending
(××PR1, ××PR0)
≤ (ISP1, ISP0)
No (Low priority)
Interrupt request held pending
Higher priority than other interrupt requests simultaneously generated?
No
Yes
Interrupt request held pending
Higher default priority[Note] than other interrupt requests simultaneously generated?
No
Yes
Interrupt request held pending
IE = 1?
No
Yes
Interrupt request held pending
Vectored interrupt servicing

××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**)

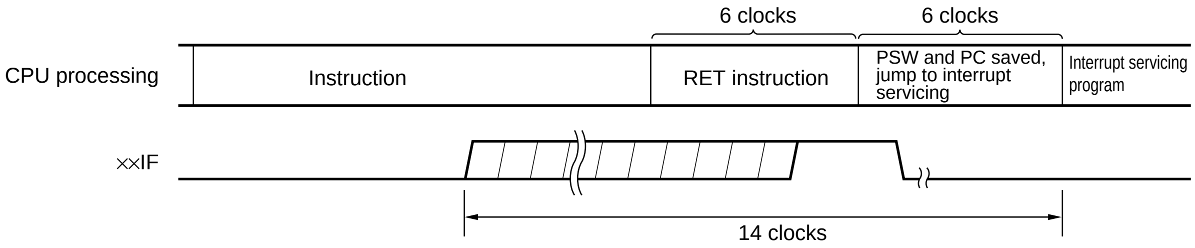
Note For the default priority, refer to **Table 15-1 Interrupt Source List**.

Figure 15-8. Interrupt Request Acknowledgment Timing (Minimum Time)


6 clocks
CPU processing
Instruction
Instruction
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
××IF
9 clocks

Remark 1 clock: 1/fCLK (fCLK: CPU clock)

Figure 15-9. Interrupt Request Acknowledgment Timing (Maximum Time)


6 clocks
6 clocks
CPU processing
Instruction
RET instruction
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
××IF
14 clocks

Remark 1 clock: 1/fCLK (fCLK: 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 execution.

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.

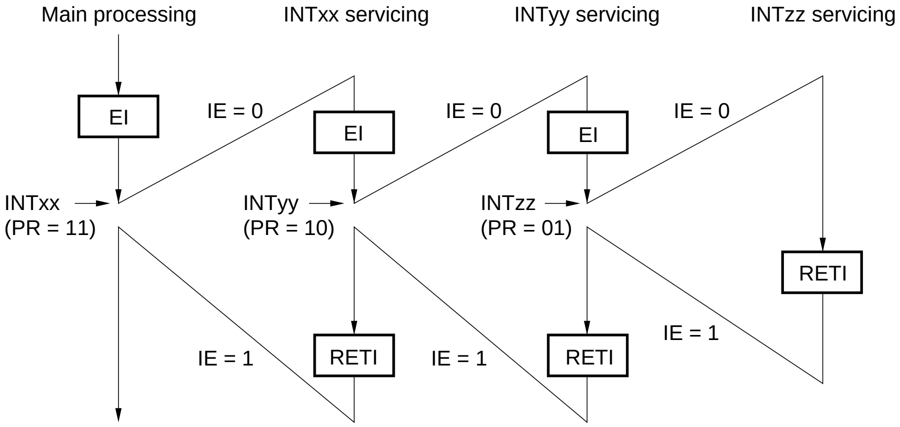
ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.
ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.
ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.
ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment.
IE = 0: Interrupt request acknowledgment is 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 1 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

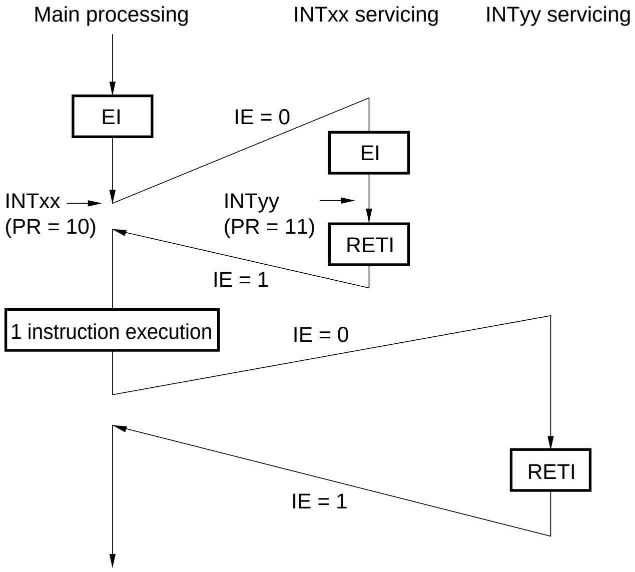
Figure 15-10. Examples of Multiple Interrupt Servicing (1/2)

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

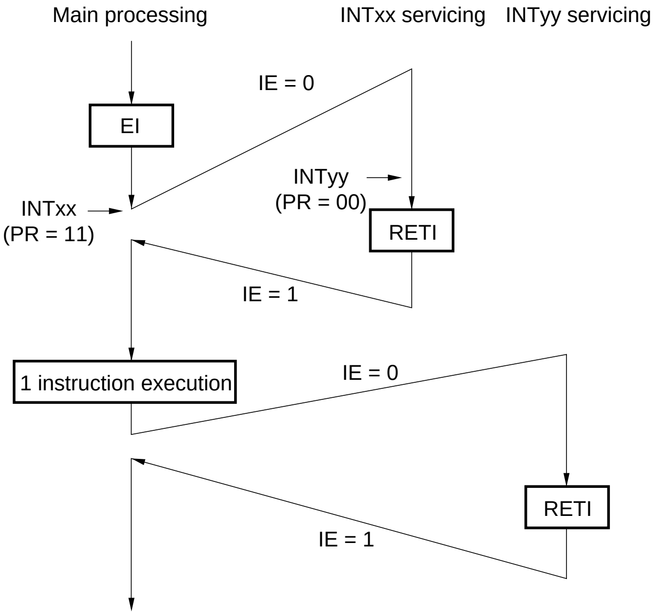


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

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 1 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 1 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

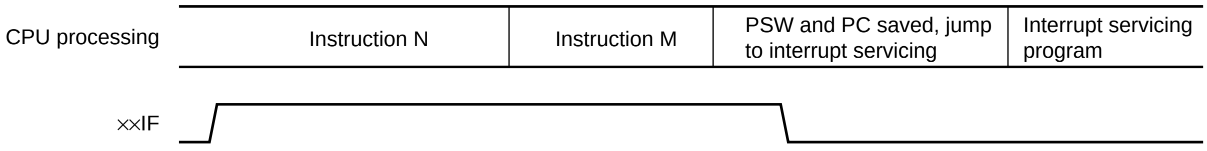
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr8
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- 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


CPU processing
Instruction N
Instruction M
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
××IF

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, double-speed mode internal high-speed oscillator, or subsystem clock oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and 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. The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.

3. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

4. It can be selected by the option byte whether the internal low-speed oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 21 OPTION BYTE.

5. The STOP instruction cannot be executed when the CPU operates on the double-speed mode 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 5 CLOCK GENERATOR**.

(1) Oscillation stabilization time counter status register (OSTC)

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

- If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem 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.

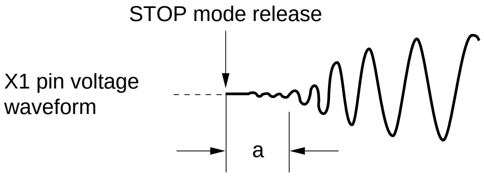
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

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.

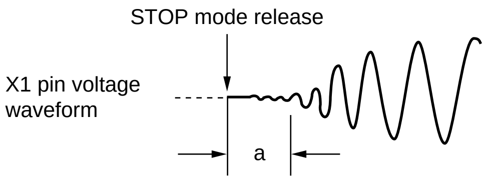
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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, double-speed mode internal high-speed oscillation clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Table 16-1. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{IH}) or Double-Speed Mode Internal High-Speed Oscillation Clock (f_{DSC})	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_{DSC}	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)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Oscillates • WTON = 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 TAUS			Operable		
Inverter control function					
Real-time counter (RTC)					
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Operates • WTON = 1 and WDSTBYON = 0: Stops		
A/D converter					
Operational amplifier					
Comparator					
Serial array unit (SAU)					
Multiplier/divider					
DMA controller					
Power-on-clear function					
Low-voltage detection function					
External interrupt					

Remark f_{IH}: Internal high-speed oscillation clock
f_{DSC}: Double-speed mode internal high-speed oscillation clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{IL}: Internal low-speed oscillation clock

Table 16-1. Operating Statuses in HALT Mode (2/2)

Item / HALT Mode Setting			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
			When CPU Is Operating on XT1 Clock (f_{XT})
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{IH}, f_{DSC}	Status before HALT mode was set is retained
		f_{X}	
		f_{EX}	Operates or stops by external clock input
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Oscillates • WTON = 1 and WDSTBYON = 0: Stops
CPU			Operation stopped
Flash memory			Operation stopped (wait state in low-current consumption mode)
RAM			The value is retained
Port (latch)			Status before HALT mode was set is retained
Timer array unit TAUS			Operable
Inverter control function			Cannot operate
Real-time counter (RTC)			Operable
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Operates • WTON = 1 and WDSTBYON = 0: Stops
A/D converter			Cannot operate
Operational amplifier			Operable
Comparator			
Serial array unit (SAU)			
Multiplier/divider			Operable
DMA controller			
Power-on-clear function			
Low-voltage detection function			
External interrupt			

Remark f_{IH}: Internal high-speed oscillation clock
f_{DSC}: Double-speed mode internal high-speed oscillation clock
f_{X}: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{IL}: Internal low-speed oscillation clock

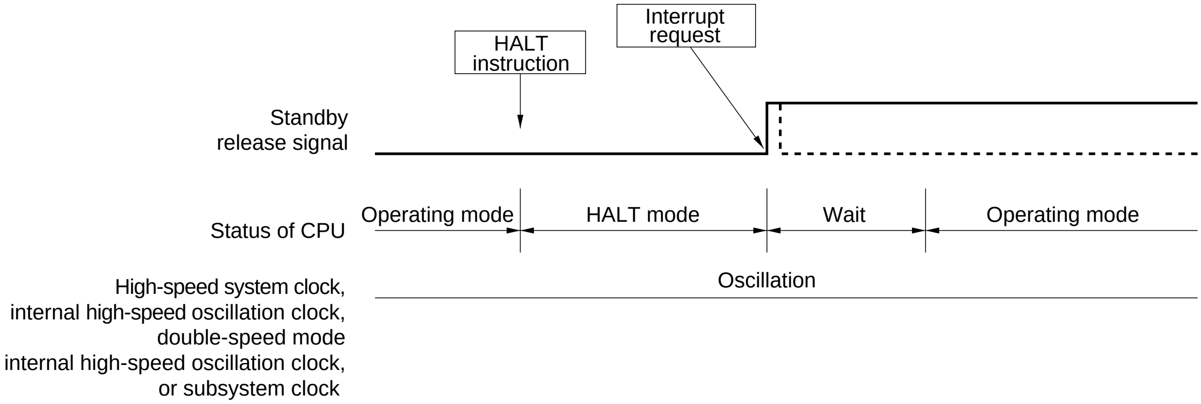
(2) HALT mode release

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


Interrupt request
HALT instruction
Standby release signal
Status of CPU
Operating mode
HALT mode
Wait
Operating mode
High-speed system clock, internal high-speed oscillation clock, double-speed mode internal high-speed oscillation clock, or subsystem clock
Oscillation

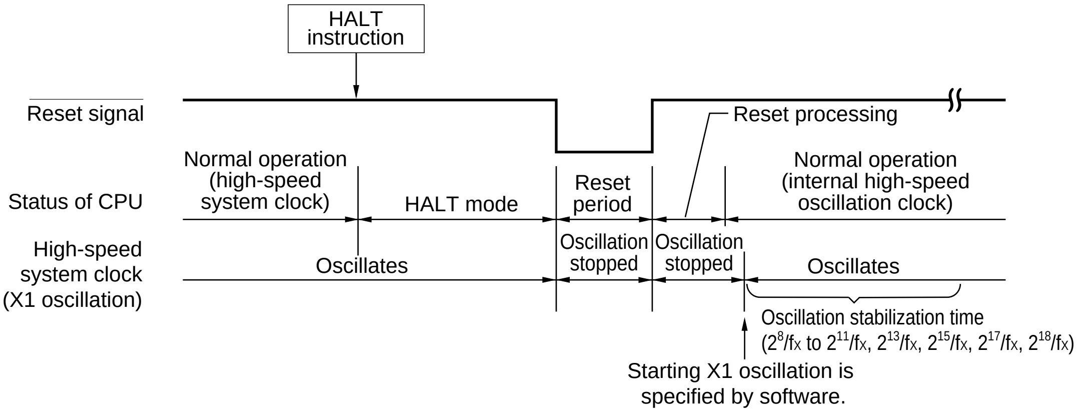
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

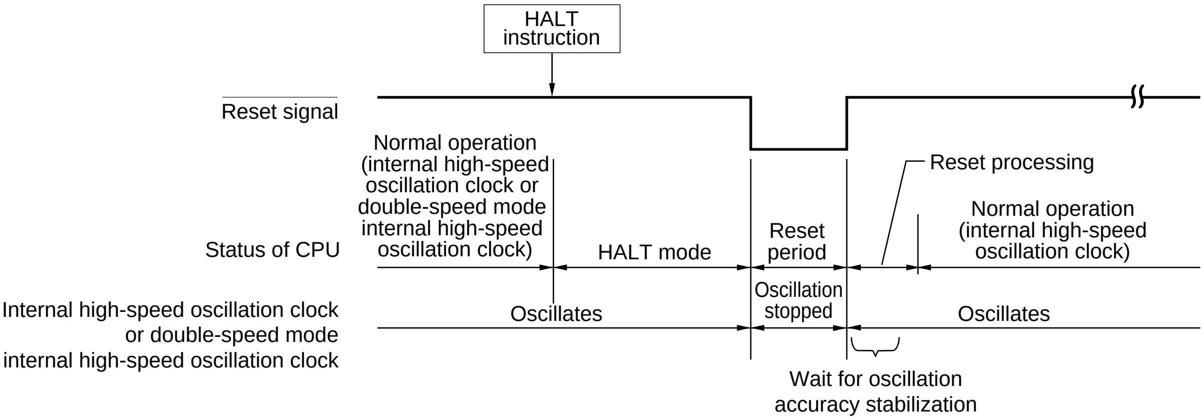
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 16-4. HALT Mode Release by Reset

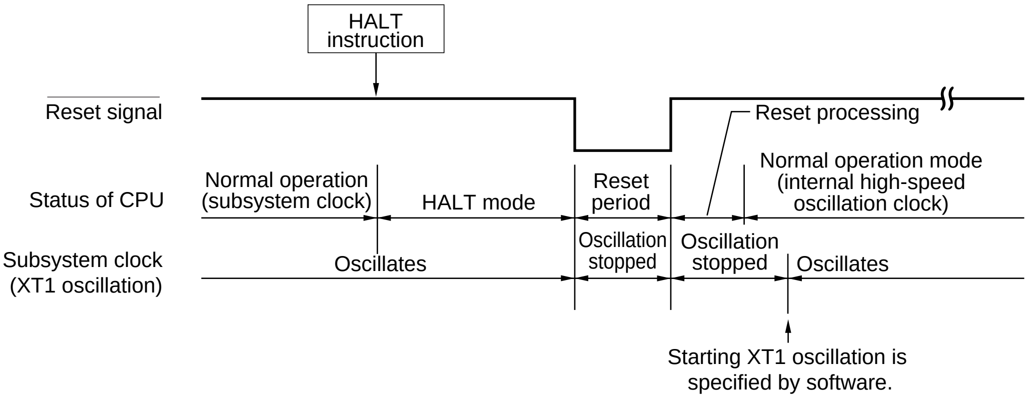
(1) When high-speed system clock is used as CPU clock


HALT instruction
Reset signal
Reset processing
Status of CPU
Normal operation (high-speed system clock)
HALT mode
Reset period
Normal operation (internal high-speed oscillation clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time
(2^8/fx to 2^11/fx, 2^13/fx, 2^15/fx, 2^17/fx, 2^18/fx)
Starting X1 oscillation is specified by software.

(2) When internal high-speed oscillation clock or double-speed mode internal high-speed oscillation clock is used as CPU clock


HALT instruction
Reset signal
Reset processing
Status of CPU
Normal operation (internal high-speed oscillation clock or double-speed mode internal high-speed oscillation clock)
HALT mode
Reset period
Normal operation (internal high-speed oscillation clock)
Internal high-speed oscillation clock or double-speed mode internal high-speed oscillation clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

(3) When subsystem clock is used as CPU clock


HALT instruction
Reset signal
Reset processing
Status of CPU
Normal operation (subsystem clock)
HALT mode
Reset period
Normal operation mode (internal high-speed oscillation clock)
Subsystem clock (XT1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Starting XT1 oscillation is specified by software.

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 CPU clock before the setting was the internal high-speed oscillation clock, X1 clock, or external main system clock.

Cautions 1. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

2. The STOP instruction cannot be executed when the CPU operates on the double-speed mode 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}			
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained		
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Oscillates • WTON = 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 TAUS			Operation disabled		
Inverter control function					
Real-time counter (RTC)			Operable		
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Operates • WTON = 1 and WDSTBYON = 0: Stops		
A/D converter			Operation disabled		
Operational amplifier					
Comparator					
Serial array unit (SAU)					
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_{XT}: XT1 clock

f_{IL}: Internal low-speed oscillation clock

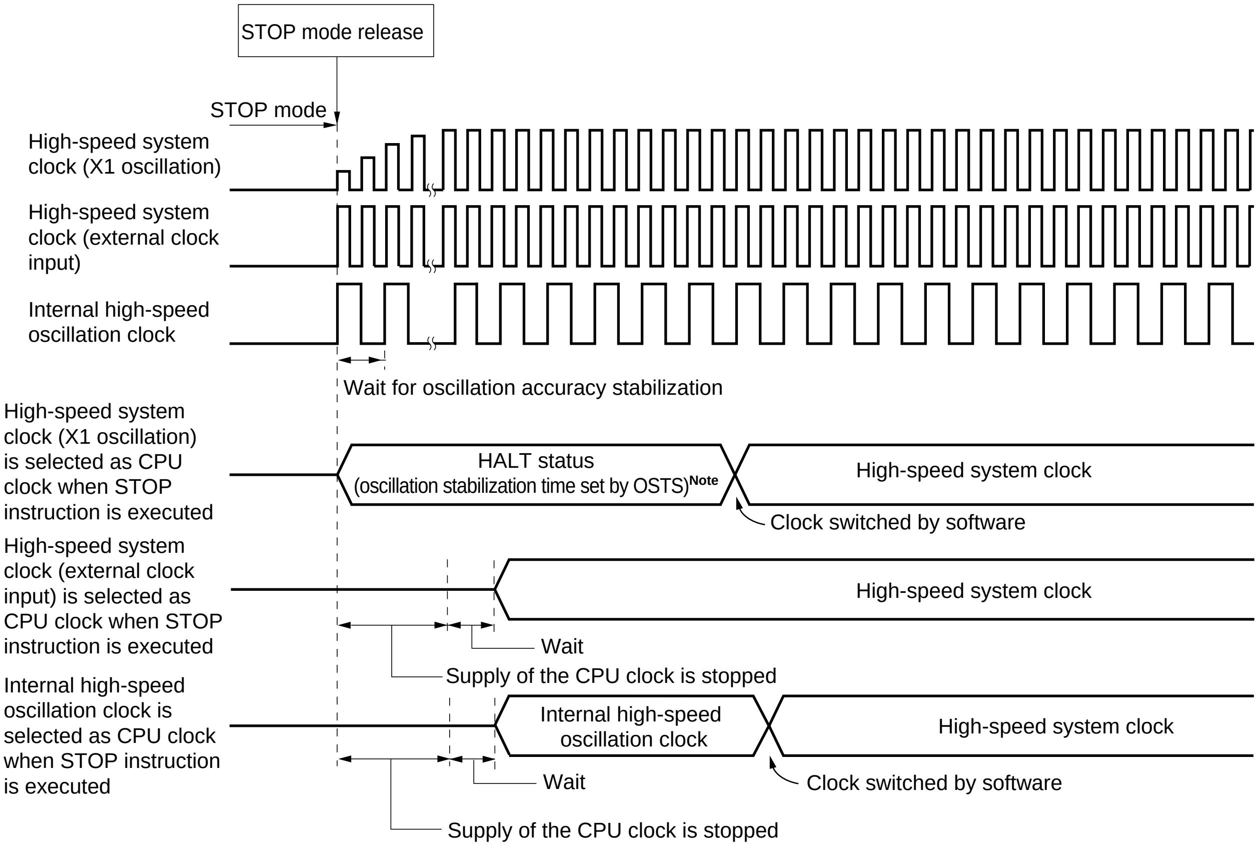
Cautions 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

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 next 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).

(2) STOP mode release

Figure 16-5. Operation Timing When STOP Mode Is Released


STOP mode release
STOP mode
High-speed system clock (X1 oscillation)
High-speed system clock (external clock input)
Internal high-speed oscillation clock
Wait for oscillation accuracy stabilization
High-speed system clock (X1 oscillation) is selected as CPU clock when STOP instruction is executed
HALT status
(oscillation stabilization time set by OSTS)Note
High-speed system clock
Clock switched by software
High-speed system clock (external clock input) is selected as CPU clock when STOP instruction is executed
High-speed system clock
Wait
Supply of the CPU clock is stopped
Internal high-speed oscillation clock is selected as CPU clock when STOP instruction is executed
Internal high-speed oscillation clock
High-speed system clock
Wait
Clock switched by software
Supply of the CPU clock is stopped

Note When the oscillation stabilization time set by OSTS is equal to or shorter than 61 μs, the HALT status is retained to a maximum of “61μs + wait time.”

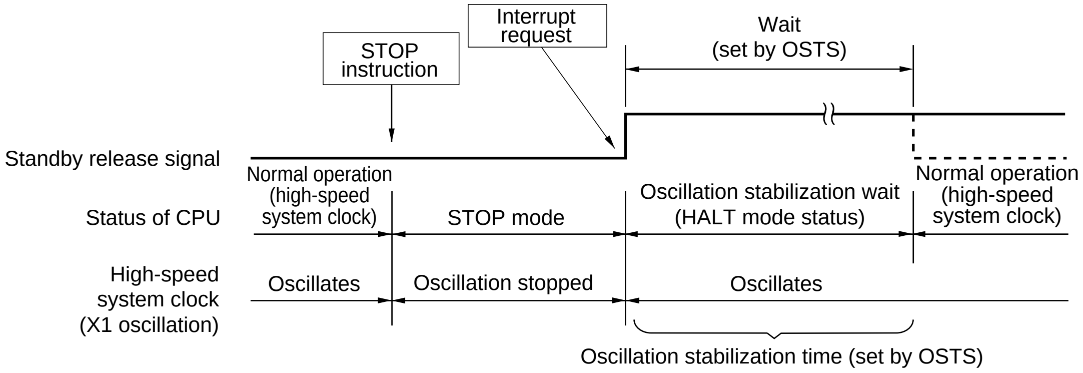
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

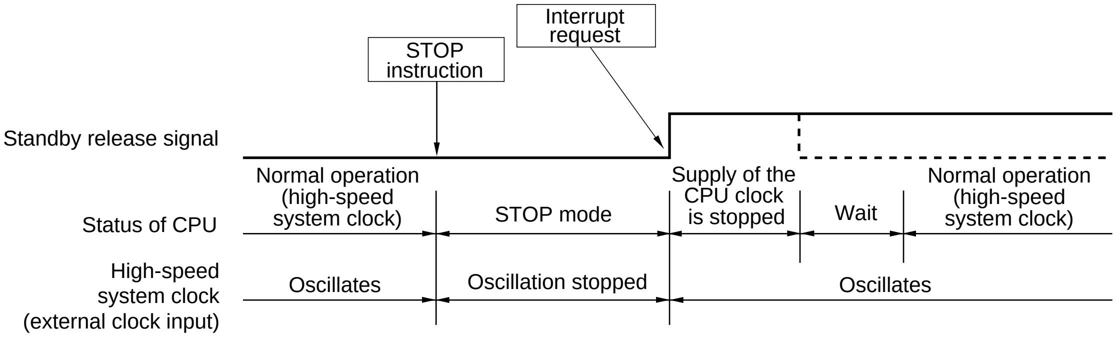
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 16-6. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock


Interrupt request
STOP instruction
Wait (set by OSTS)
Standby release signal
Normal operation (high-speed system clock)
Status of CPU
STOP mode
Oscillation stabilization wait (HALT mode status)
Normal operation (high-speed system clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillates
Oscillation stabilization time (set by OSTS)

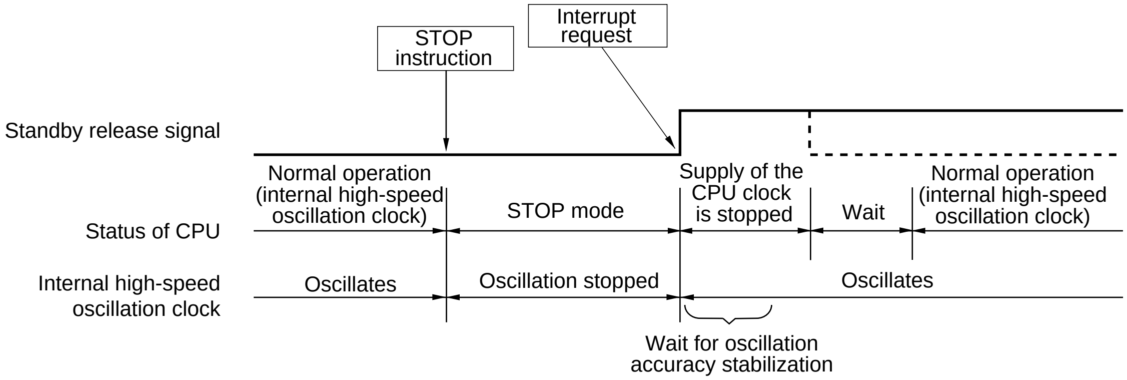
(2) When high-speed system clock (external clock input) is used as CPU clock


Interrupt request
STOP instruction
Standby release signal
Normal operation (high-speed system clock)
Status of CPU
STOP mode
Supply of the CPU clock is stopped
Wait
Normal operation (high-speed system clock)
High-speed system clock (external clock input)
Oscillates
Oscillation stopped
Oscillates

Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 16-6. STOP Mode Release by Interrupt Request Generation (2/2)

(3) When internal high-speed oscillation clock is used as CPU clock


Interrupt request
STOP instruction
Standby release signal
Normal operation (internal high-speed oscillation clock)
STOP mode
Supply of the CPU clock is stopped
Wait
Normal operation (internal high-speed oscillation clock)
Status of CPU
Internal high-speed oscillation clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

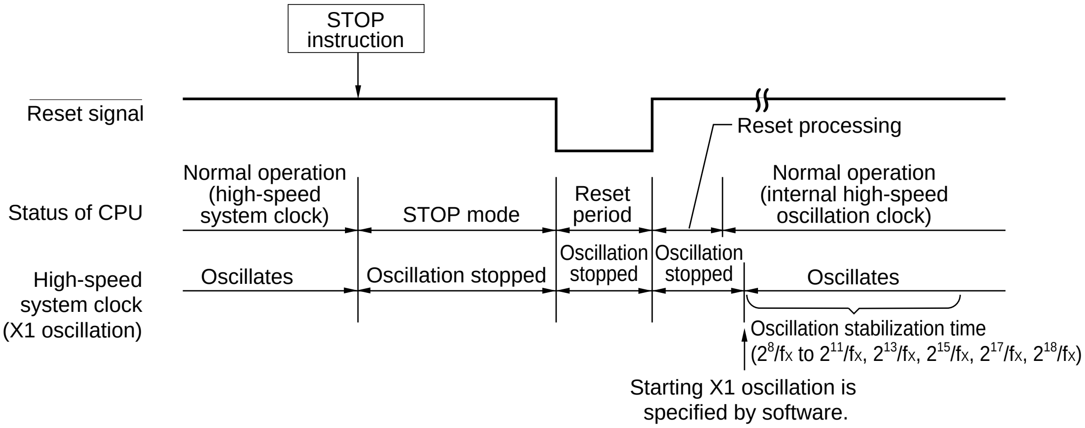
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

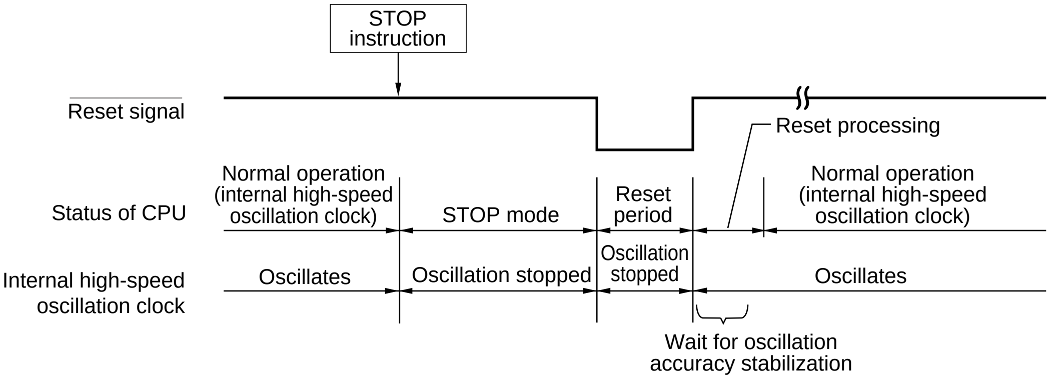
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 16-7. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock


STOP instruction
Reset signal
Reset processing
Status of CPU
Normal operation (high-speed system clock)
STOP mode
Reset period
Normal operation (internal high-speed oscillation clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time (2^8/fx to 2^{11}/fx, 2^{13}/fx, 2^{15}/fx, 2^{17}/fx, 2^{18}/fx)
Starting X1 oscillation is specified by software.

(2) When internal high-speed oscillation clock is used as CPU clock


STOP instruction
Reset signal
Reset processing
Status of CPU
Normal operation (internal high-speed oscillation clock)
STOP mode
Reset period
Normal operation (internal high-speed oscillation clock)
Internal high-speed oscillation clock
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

Remark fx: X1 clock oscillation frequency

CHAPTER 17 RESET FUNCTION

The following five 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 of the low-voltage detector (LVI) or input voltage (EXLVI) from external input pin, and detection voltage
(5) Internal reset by execution of illegal instruction[Note]

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 power supply detection is automatically released when $V_{DD} \geq V_{POC}$ 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 21 LOW-VOLTAGE DETECTOR**) after reset processing.

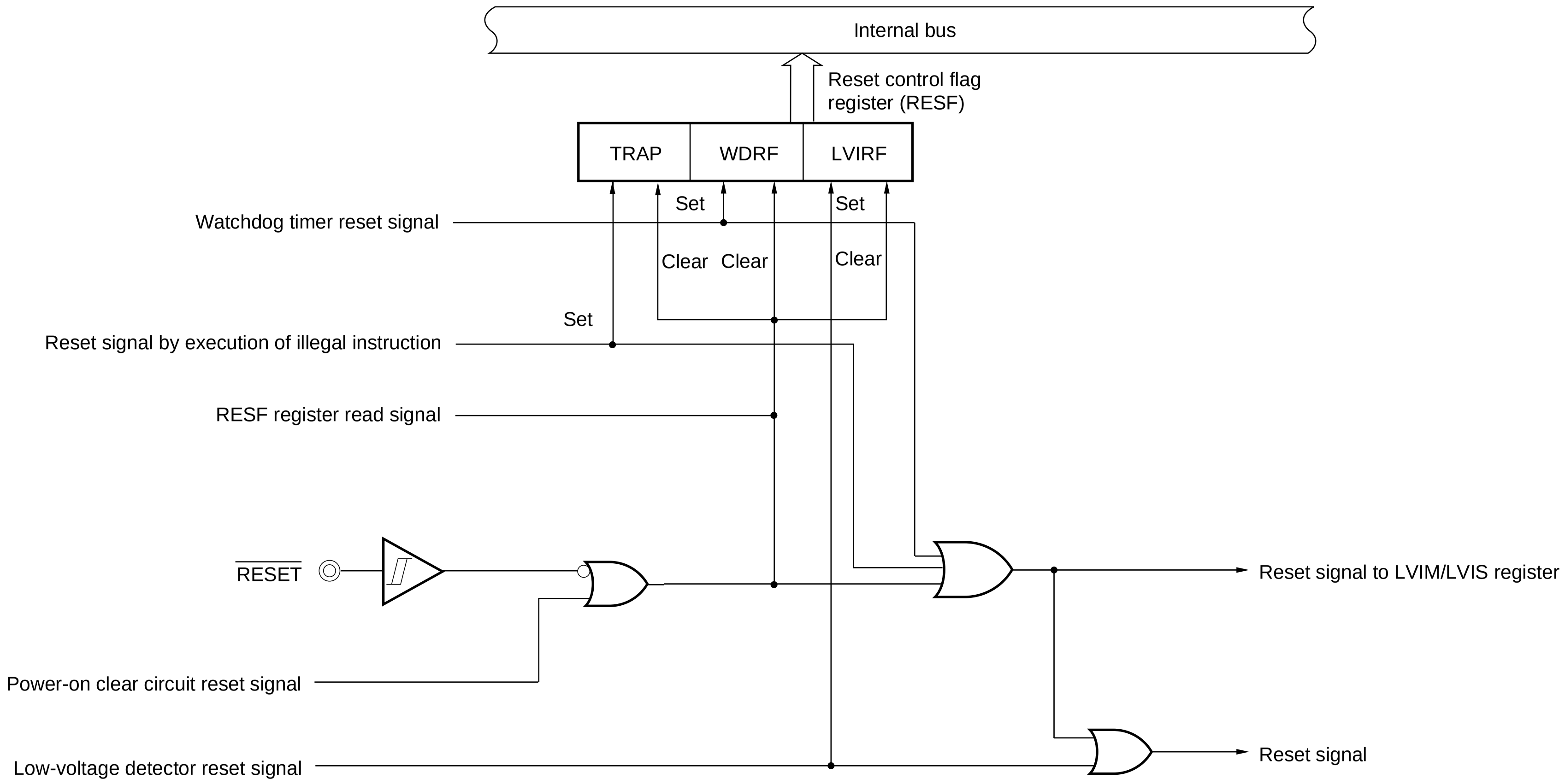
Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
(If an external reset is effected upon power application, the period during which the supply voltage is outside the operating range (V_{DD} < 2.7 V) is not counted in the 10 μs. However, the low-level input may be continued before POC is released.)

2. During reset input, the X1 clock, XT1 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. However, because SFR and 2nd SFR are initialized, the port pins become high-impedance.

Figure 17-1. Block Diagram of Reset Function


Internal bus
Reset control flag register (RESF)
TRAP
WDRF
LVIRF
Set
Set
Clear
Clear
Clear
Set
Watchdog timer reset signal
Reset signal by execution of illegal instruction
RESF register read signal
RESET
Power-on clear circuit reset signal
Low-voltage detector reset signal
Reset signal to LVIM/LVIS register
Reset signal

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

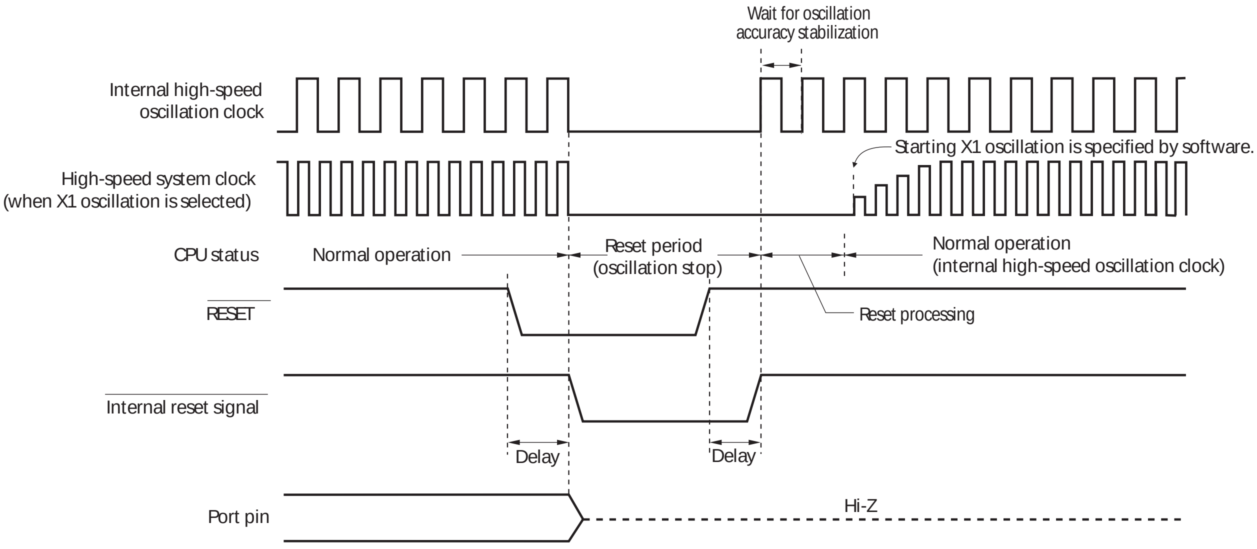
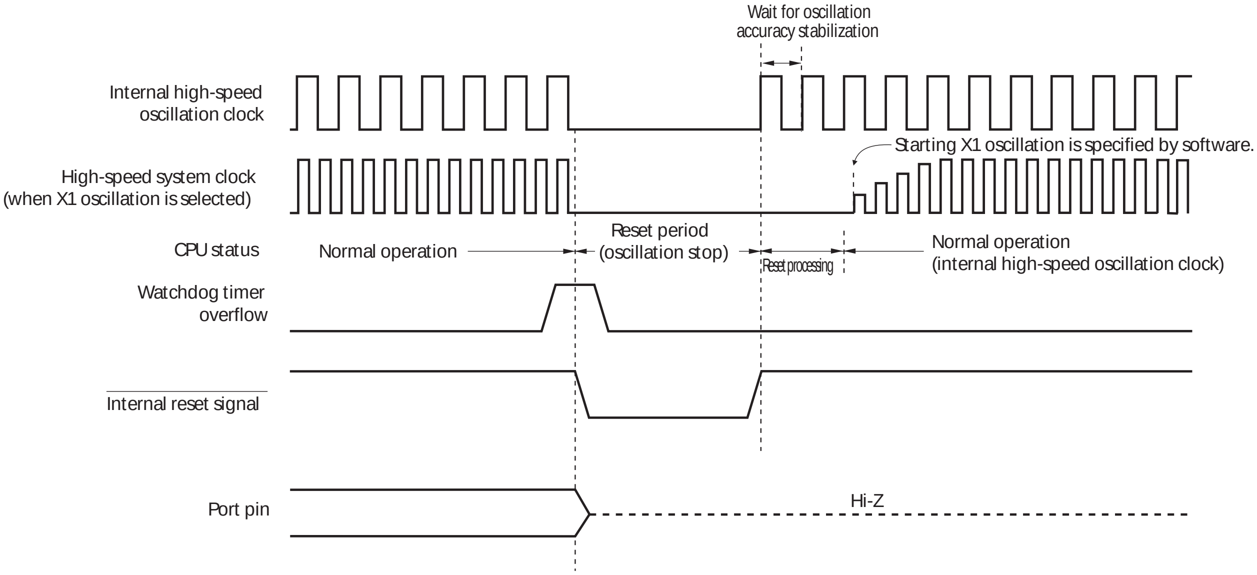
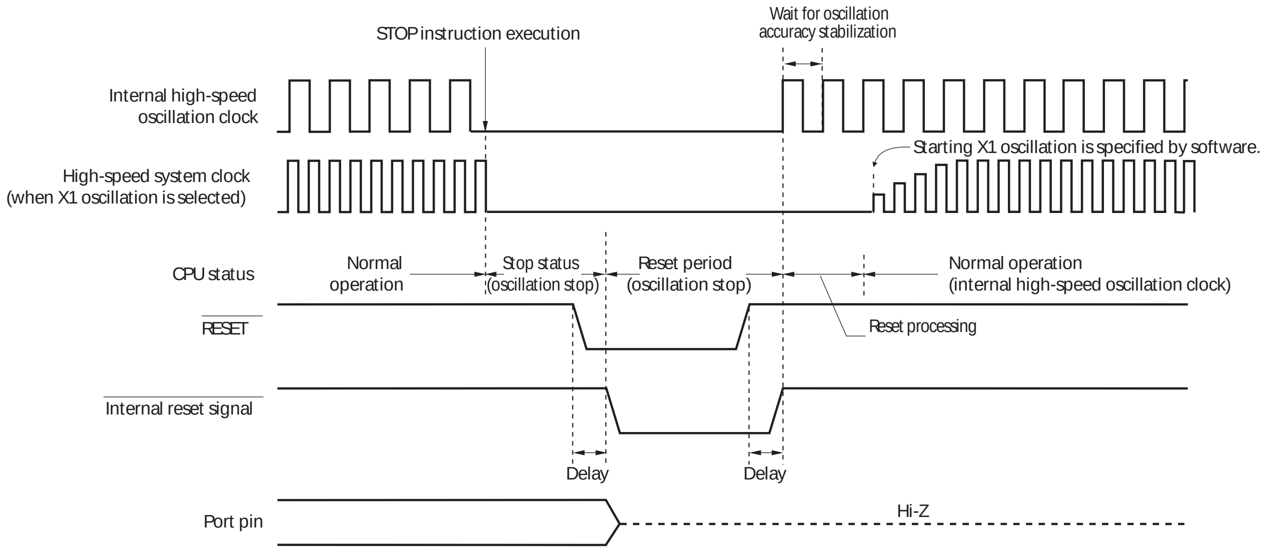

Wait for oscillation accuracy stabilization
Internal high-speed oscillation clock
Starting X1 oscillation is specified by software.
High-speed system clock (when X1 oscillation is selected)
CPU status
Normal operation
Reset period (oscillation stop)
Normal operation (internal high-speed oscillation clock)
Reset processing
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Figure 17-3. Timing of Reset Due to Watchdog Timer Overflow


Wait for oscillation accuracy stabilization
Internal high-speed oscillation clock
Starting X1 oscillation is specified by software.
High-speed system clock (when X1 oscillation is selected)
CPU status
Normal operation
Reset period (oscillation stop)
Reset processing
Normal operation (internal high-speed oscillation clock)
Watchdog timer overflow
Internal reset signal
Port pin
Hi-Z

Caution A watchdog timer internal reset resets the watchdog timer.

Figure 17-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input


STOP instruction execution
Wait for oscillation accuracy stabilization
Internal high-speed oscillation clock
High-speed system clock (when X1 oscillation is selected)
Starting X1 oscillation is specified by software.
CPU status
Normal operation
Stop status (oscillation stop)
Reset period (oscillation stop)
Normal operation (internal high-speed oscillation clock)
RESET
Reset processing
Internal reset signal
Delay
Delay
Port pin
Hi-Z

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}	Operation stopped
		f_X	Operation stopped (X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (pin is input port mode)
	Subsystem clock	f_{XT}	Operation stopped (XT1 and XT2 pins are 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 TAUS			Operation stopped
Real-time counter (RTC)			
Watchdog timer			
A/D converter			
Operational amplifier			
Comparator			
Serial array unit (SAU)			
Multiplier/divider			
DMA controller			
Power-on-clear function			Detection operation is 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_X: X1 oscillation clock

f_{EX}: External main system clock

f_{XT}: XT1 oscillation clock

f_{IL}: Internal low-speed oscillation clock

Table 17-2. Hardware Statuses After Reset Acknowledgment (1/4)

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]
Port registers (P1 to P5 , P7, P8, P12, P15) (output latches)		00H
Port mode registers (PM1 to PM5, PM7, PM8, PM12, PM15)		FFH
Port input mode registers (PIM3, PIM7, PIM8)		00H
Port output mode registers (POM3, POM7)		00H
Pull-up resistor option registers (PU1, PU3 to PU5, PU7, PU12)		00H
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
Processor mode control register (PMC)		00H
System clock control register (CKC)		09H
Double-speed operation control register (DSCCTL)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		07H
Noise filter enable registers 0, 1, 2 (NFEN0, NFEN1, NFEN2)		00H
Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2)		00H
Operation speed mode control register (OSMC)		00H
Timer array unit (TAUS)	Timer data registers 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11 (TDR00, TDR01, TDR02, TDR03, TDR04, TDR05, TDR06, TDR07, TDR08, TDR09, TDR10, TDR11)	0000H
	Timer mode registers 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11 (TMR00, TMR01, TMR02, TMR03, TMR04, TMR05, TMR06, TMR07, TMR08, TMR09, TMR10, TMR11)	0000H
	Timer status registers 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11 (TSR00, TSR01, TSR02, TSR03, TSR04, TSR05, TSR06, TSR07, TSR08, TSR09, TSR10, TSR11)	0000H
	Timer input select register 0 (TIS0)	00H
	Timer counter registers 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11 (TCR00, TCR01, TCR02, TCR03, TCR04, TCR05, TCR06, TCR07, TCR08, TCR09, TCR10, TCR11)	FFFFH
	Timer channel enable status register 0 (TE0)	0000H
	Timer channel start trigger register 0 (TS0)	0000H
	Timer channel stop trigger register 0 (TT0)	0000H
	Timer clock select register 0 (TPS0)	0000H
	Timer channel output register 0 (TO0)	0000H
	Timer channel output enable register 0 (TOE0)	0000H
	Timer channel output level register 0 (TOL0)	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 (2/4)

Hardware		Status After Reset Acknowledgment[Note 1]
Timer array unit (TAUS)	Timer channel output mode register 0 (TOM0)	0000H
	Timer output 3-phase mode register 0 (TOT0)	0000H
	Timer real-time output enable register 0 (TRE0)	0000H
	Timer real-time output register 0 (TRO0)	0000H
	Timer real-time control register 0 (TRC0)	0000H
	Timer modulation output enable register 0 (TME0)	0000H
	Timer dead time output enable register 0 (TDE0)	0000H
	TAU option mode register (OPMR)	0000H
	TAU option status register (OPSR)	0000H
	TAU option Hi-Z start trigger register (OPHS)	0000H
	TAU option Hi-Z stop trigger register (OPHT)	0000H
Real-time counter	Sub-count register (RSUBC)	0000H
	Second count register (SEC)	00H
	Minute count register (MIN)	00H
	Hour count register (HOUR)	12H
	Week count register (WEEK)	00H
	Day count register (DAY)	01H
	Month count register (MONTH)	01H
	Year count register (YEAR)	00H
	Watch error correction register (SUBCUD)	00H
	Alarm minute register (ALARMWM)	00H
	Alarm hour register (ALARMWH)	12H
	Alarm week register ALARMWW)	00H
	Real-time counter control register 0 (RTCC0)	00H
	Real-time counter control register 1 (RTCC1)	00H
	Real-time counter control register 2 (RTCC2)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 2]
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	Mode register (ADM)	00H
	Mode register 1 (ADM1)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	10H

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 (3/4)

Hardware		Status After Reset Acknowledgment[Note]
Serial array unit (SAU)	Serial data registers 00, 01, 02, 03 (SDR00, SDR01, SDR02, SDR03)	0000H
	Serial status registers 00, 01, 02, 03 (SSR00, SSR01, SSR02, SSR03)	0000H
	Serial flag clear trigger registers 00, 01, 02, 03 (SIR00, SIR01, SIR02, SIR03)	0000H
	Serial mode registers 00, 01, 02, 03 (SMR00, SMR01, SMR02, SMR03)	0020H
	Serial communication operation setting registers 00, 01, 02, 03 (SCR00, SCR01, SCR02, SCR03)	0087H
	Serial channel enable status register 0 (SE0)	0000H
	Serial channel start trigger register 0 (SS0)	0000H
	Serial channel stop trigger register 0 (ST0)	0000H
	Serial clock select register 0 (SPS0)	0000H
	Serial output register 0 (SO0)	0F0FH
	Serial output enable register 0 (SOE0)	0000H
	Input switch control register (ISC)	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

Note. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Table 17-2. Hardware Statuses After Reset Acknowledgment (4/4)

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]
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
Operational amplifier	Operational amplifier control register (OAM)	00H
Comparator	Comparator 0 control register (C0CTL)	00H
	Comparator 0 internal reference voltage setting register (C0RVM)	00H
	Comparator 1 control register (C1CTL)	00H
	Comparator 1 internal reference voltage setting register (C1RVM)	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. These values vary depending on the reset source.

Reset Source Register		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by LVI
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held
	WDRF bit			Held	Set (1)	Held
	LVIRF bit			Held	Held	Set (1)
LVIS		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 μPD79F9211. 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 RESF set RESF to 00H.

Figure 17-5. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: 00H[Note 1] R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDRF	0	0	0	LVIRF

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.

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-chipdebug 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 LVI
TRAP	Cleared (0)	Cleared (0)	Set (1)	Held	Held
WDRF			Held	Set (1)	Held
LVIRF			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[Note].

  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[Note].

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V[Note]), generates internal reset signal when $V_{DD} < V_{PDR}$.

Note These are preliminary values and subject to change.

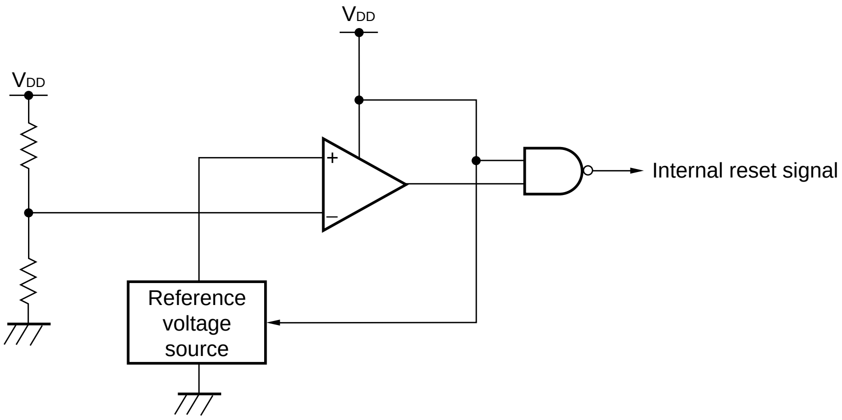
Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

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 or LVI.
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$^{\textbf{Note}}$), 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$^{\textbf{Note}}$.

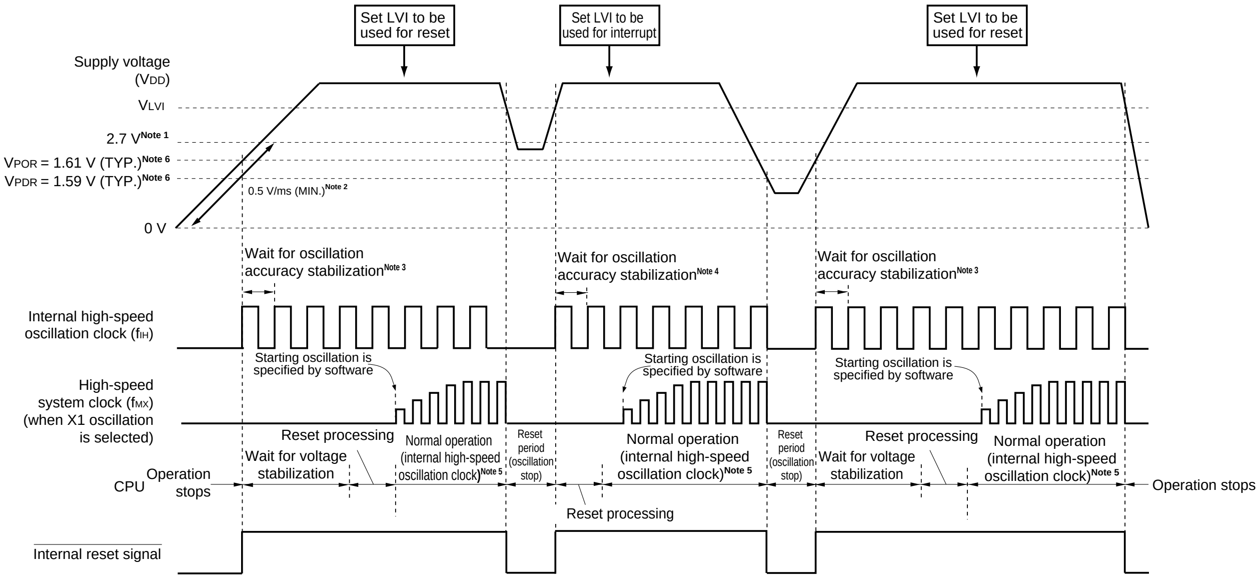
- The supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V$^{\textbf{Note}}$) are compared. When $V_{DD} < V_{PDR}$, the internal reset signal is generated.

Note These are preliminary values and subject to change.

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)


Set LVI to be used for reset
Set LVI to be used for interrupt
Set LVI to be used for reset
Supply voltage (VDD)
VLVI
2.7 V Note 1
VPOR = 1.61 V (TYP.) Note 6
VPDR = 1.59 V (TYP.) Note 6
0.5 V/ms (MIN.) Note 2
0 V
Wait for oscillation accuracy stabilization Note 3
Wait for oscillation accuracy stabilization Note 4
Wait for oscillation accuracy stabilization Note 3
Internal high-speed oscillation clock (fIH)
Starting oscillation is specified by software
Starting oscillation is specified by software
Starting oscillation is specified by software
High-speed system clock (fMX) (when X1 oscillation is selected)
Reset processing
Normal operation (internal high-speed oscillation clock) Note 5
Reset period (oscillation stop)
Normal operation (internal high-speed oscillation clock) Note 5
Reset period (oscillation stop)
Reset processing
Normal operation (internal high-speed oscillation clock) Note 5
CPU
Operation stops
Wait for voltage stabilization
Wait for voltage stabilization
Operation stops
Reset processing
Internal reset signal

Notes 1. The operation guaranteed range is 2.7 V ≤ V_{DD} ≤ 5.5 V. Make sure to perform normal operation after the supply voltage has become at least 2.7 V. To make the state at lower than 2.7 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 2.7 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 2.7 V, or set LVI to ON by default by using an option byte (option byte: LVIOFF = 0).

3. The internal voltage stabilization time 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 or subsystem 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. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

6. This is a preliminary value and subject to change.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 19 LOW-VOLTAGE DETECTOR).

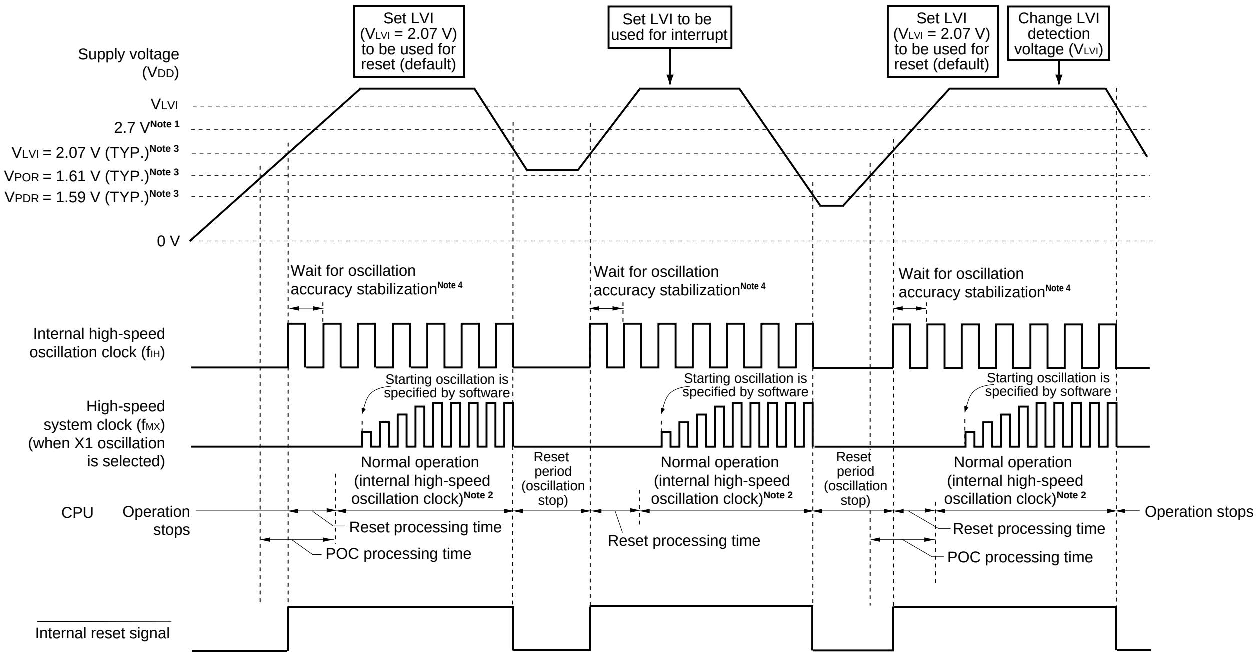
Remark V_{LVI}: LVI detection voltage

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)


Set LVI (VLVI = 2.07 V) to be used for reset (default)
Set LVI to be used for interrupt
Set LVI (VLVI = 2.07 V) to be used for reset (default)
Change LVI detection voltage (VLVI)
Supply voltage (VDD)
VLVI
2.7 VNote 1
VLVI = 2.07 V (TYP.)Note 3
VPOR = 1.61 V (TYP.)Note 3
VPDR = 1.59 V (TYP.)Note 3
0 V
Wait for oscillation accuracy stabilizationNote 4
Wait for oscillation accuracy stabilizationNote 4
Wait for oscillation accuracy stabilizationNote 4
Internal high-speed oscillation clock (fIH)
Starting oscillation is specified by software
Starting oscillation is specified by software
Starting oscillation is specified by software
High-speed system clock (fMX) (when X1 oscillation is selected)
Normal operation (internal high-speed oscillation clock)Note 2
Reset period (oscillation stop)
Normal operation (internal high-speed oscillation clock)Note 2
Reset period (oscillation stop)
Normal operation (internal high-speed oscillation clock)Note 2
CPU
Operation stops
Operation stops
Reset processing time
POC processing time
Reset processing time
Reset processing time
POC processing time
Internal reset signal

Notes 1. The operation guaranteed range is $2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$. Make sure to perform normal operation after the supply voltage has become at least 2.7 V. To make the state at lower than 2.7 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 or subsystem 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. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

3. These are preliminary values and subject to change.

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

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 19 LOW-VOLTAGE DETECTOR).

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

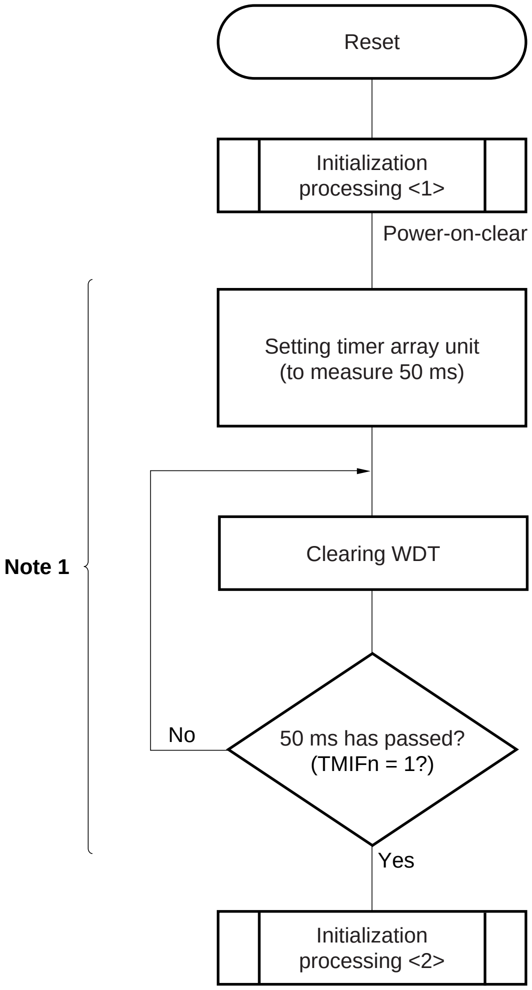
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POR}, V_{PDR}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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


Reset
Initialization processing <1>
Power-on-clear
Setting timer array unit (to measure 50 ms)
Clearing WDT
Note 1
No
50 ms has passed? (TMIFn = 1?)
Yes
Initialization processing <2>

; Check the reset source, etc.[Note 2]

; f_{CLK} = Internal high-speed oscillation clock (8.4 MHz (MAX.)) (default)
Source: f_{CLK} (8.4 MHz (MAX.))/2^{12},
where comparison value = 102: ≅ 50 ms
Timer starts (TSn = 1).

; Initial setting for port.
Setting of division ratio of system clock,
such as setting of timer or A/D converter.

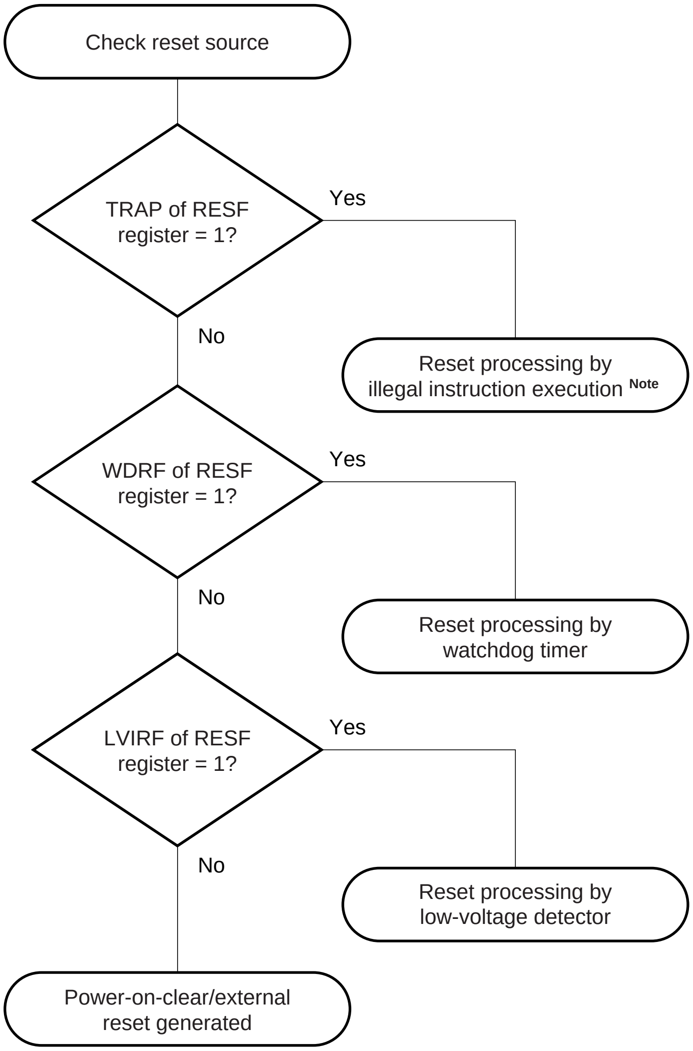
Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Remark n = 00 to 11

Figure 18-3. Example of Software Processing After Reset Release (2/2)

- Checking reset source


Check reset source
TRAP of RESF register = 1?
Yes
No
Reset processing by illegal instruction execution Note
WDRF of RESF register = 1?
Yes
No
Reset processing by watchdog timer
LVIRF of RESF register = 1?
Yes
No
Reset processing by low-voltage detector
Power-on-clear/external reset generated

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^Note^), 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^Note^). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V^Note^).
- 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},16 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

Note This is a preliminary value and subject to change.

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

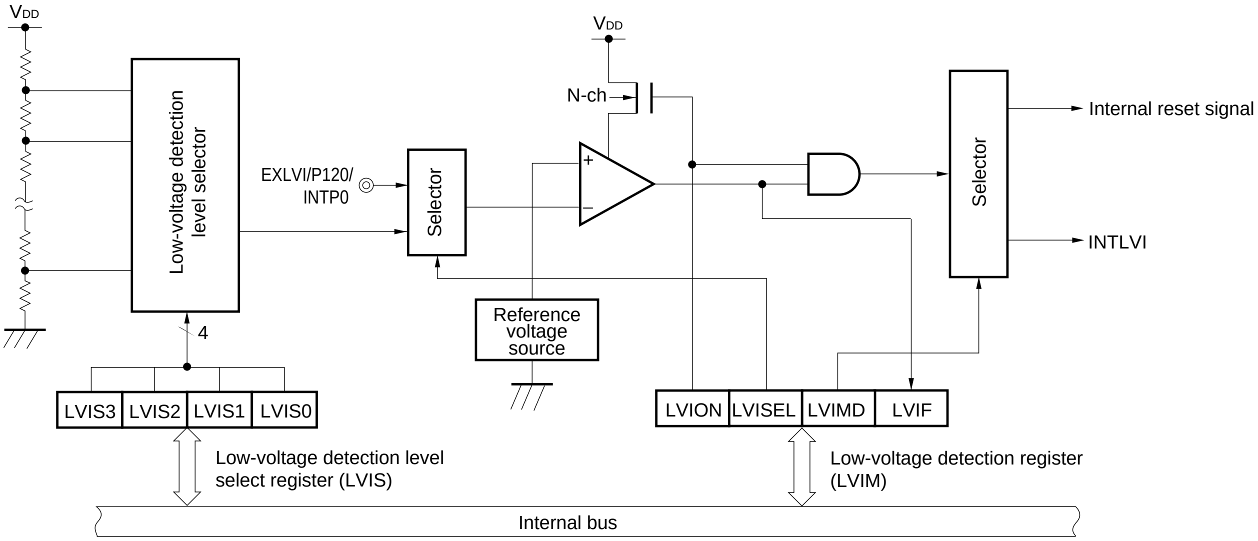
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

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	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.))
- Detection delay time (200 μs (MAX.))

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, follow either of the procedures below.

- **When using 8-bit memory manipulation instruction: Write 00H to LVIM.**
- **When using 1-bit memory manipulation instruction: Clear LVION to 0.**

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.

(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 1] 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)[Note 2]
0	0	0	1	V_{LVI1} (4.07 ±0.1 V)[Note 2]
0	0	1	0	V_{LVI2} (3.92 ±0.1 V)[Note 2]
0	0	1	1	V_{LVI3} (3.76 ±0.1 V)[Note 2]
0	1	0	0	V_{LVI4} (3.61 ±0.1 V)[Note 2]
0	1	0	1	V_{LVI5} (3.45 ±0.1 V)[Note 2]
0	1	1	0	V_{LVI6} (3.30 ±0.1 V)[Note 2]
0	1	1	1	V_{LVI7} (3.15 ±0.1 V)[Note 2]
1	0	0	0	V_{LVI8} (2.99 ±0.1 V)[Note 2]
1	0	0	1	V_{LVI9} (2.84 ±0.1 V)[Note 2]
1	0	1	0	V_{LVI10} (2.68 ±0.1 V)[Note 2]
1	0	1	1	V_{LVI11} (2.53 ±0.1 V)[Note 2]
1	1	0	0	V_{LVI12} (2.38 ±0.1 V)[Note 2]
1	1	0	1	V_{LVI13} (2.22 ±0.1 V)[Note 2]
1	1	1	0	V_{LVI14} (2.07 ±0.1 V)[Note 2]
1	1	1	1	V_{LVI15} (1.91 ±0.1 V)[Note 2]

Notes 1. 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” if a reset other than by LVI is effected.

2. These are preliminary values and subject to change.

Caution 1. Be sure to clear bits 4 to 7 to “0”.

Cautions 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 it with software because an LVIIF flag 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 it with software because an LVIIF flag 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.

(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}), 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$^{\textbf{Note}}$). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V$^{\textbf{Note}}$).

(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$^{\textbf{Note}}$). 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).

Note This is a preliminary value and subject to change.

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 410 μs).

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))
- Detection delay time (200 μs (MAX.))

<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.

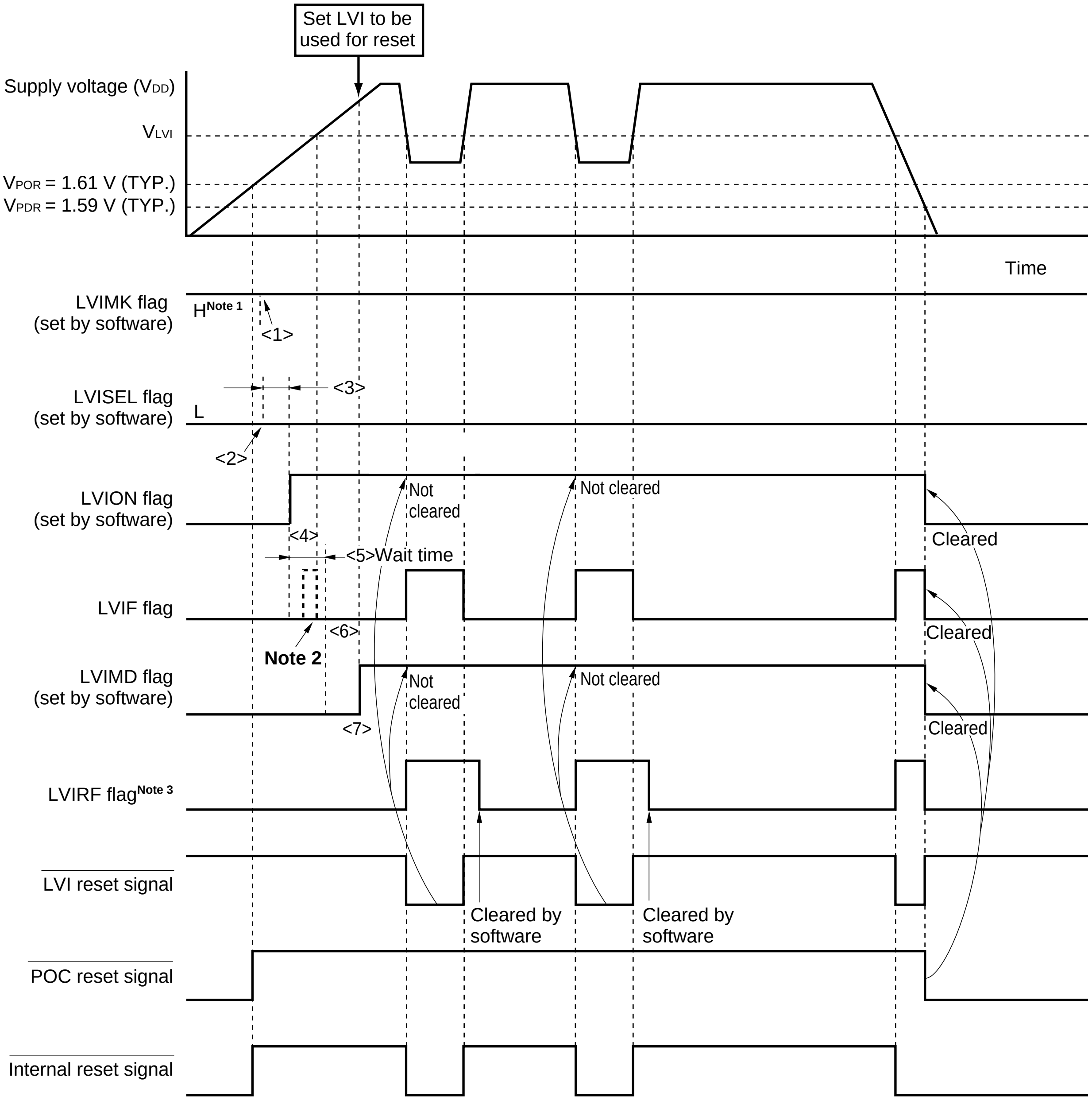
- When stopping operation

Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:
  Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:
  Clear LVIMD to 0 and then LVION to 0.

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. 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

  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

  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

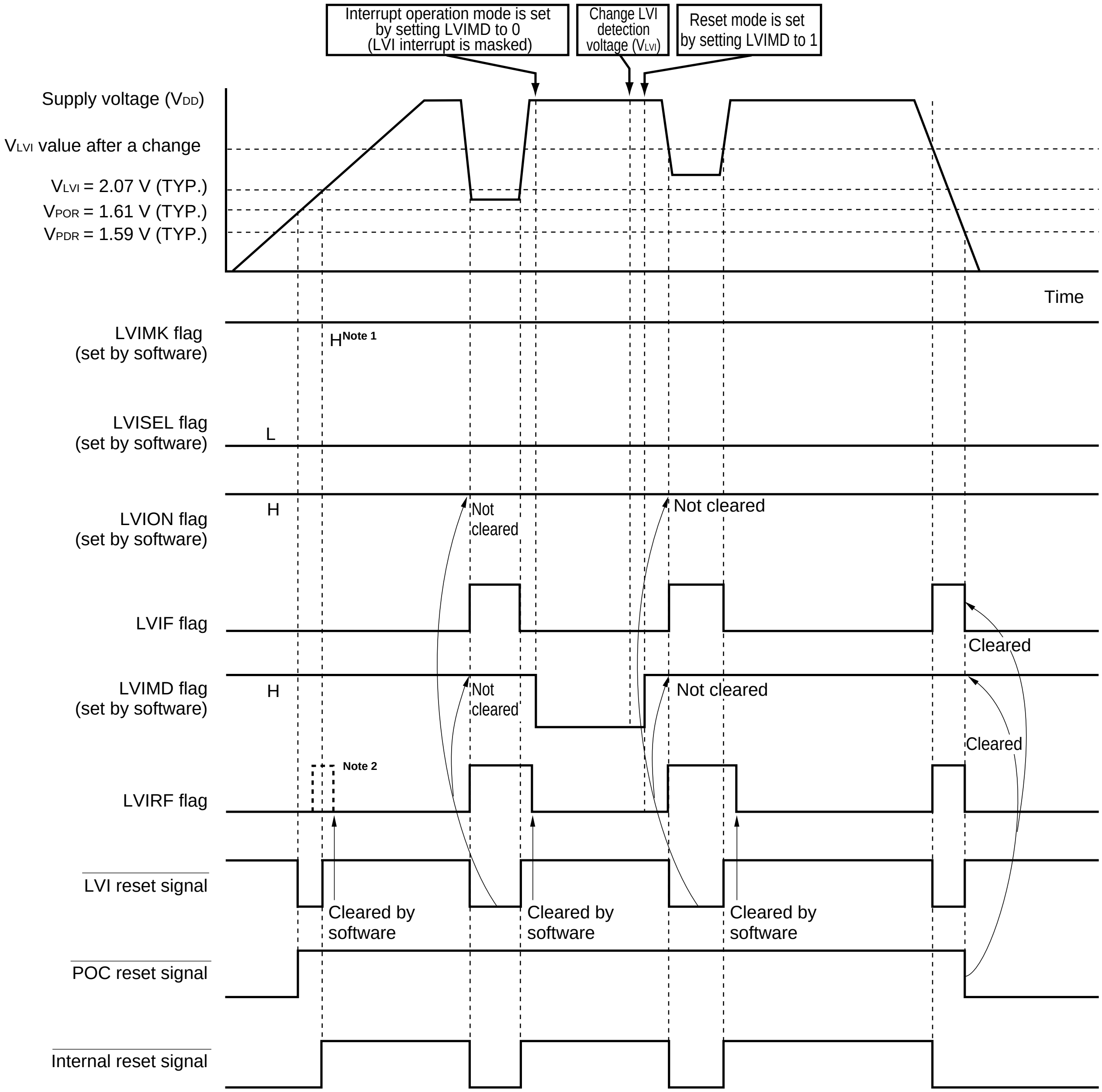
  - When using 1-bit memory manipulation instruction:
    Clear LVIMD to 0 and then LVION to 0.

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)


Interrupt operation mode is set by setting LVIMD to 0 (LVI interrupt is masked)
Change LVI detection voltage (VLVI)
Reset mode is set by setting LVIMD to 1
Supply voltage (VDD)
VLVI value after a change
VLVI = 2.07 V (TYP.)
VPOR = 1.61 V (TYP.)
VPDR = 1.59 V (TYP.)
Time
LVIMK flag (set by software)
H Note 1
LVISEL flag (set by software)
L
LVION flag (set by software)
H
Not cleared
Not cleared
LVIF flag
Cleared
LVIMD flag (set by software)
H
Not cleared
Not cleared
Cleared
LVIRF flag
Note 2
LVI reset signal
Cleared by software
Cleared by software
Cleared by software
POC reset signal
Internal reset signal

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 410 μs).

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))
- Detection delay time (200 μs (MAX.))

<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}.

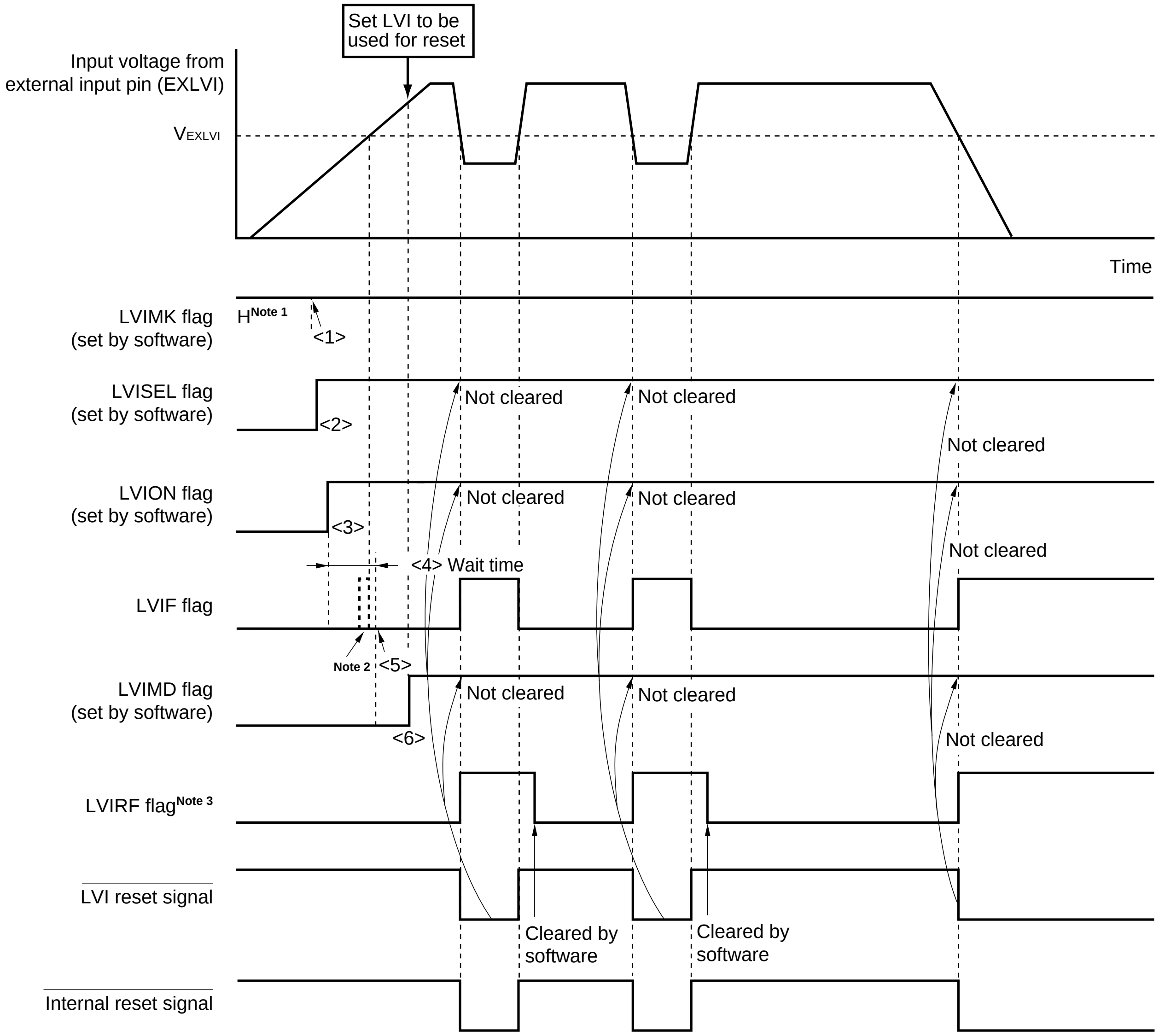
- When stopping operation

Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:
  Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:
  Clear LVIMD to 0 and then LVION to 0.

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 410 μs).

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))
- Detection delay time (200 μs (MAX.))

<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).

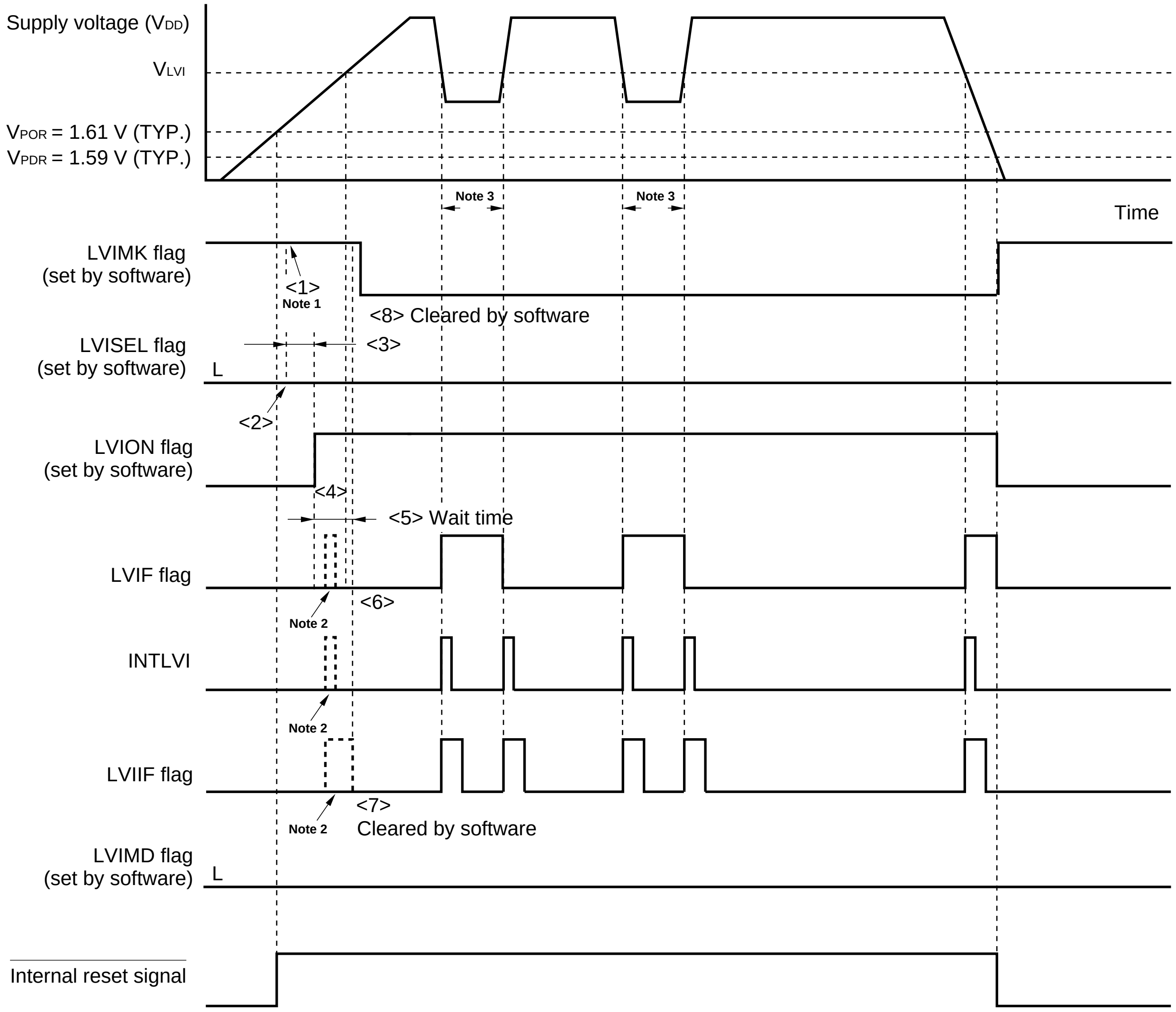
Figure 19-8 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <8> above.

- When stopping operation
Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:
Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:
Clear LVION to 0.

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

  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:

    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:

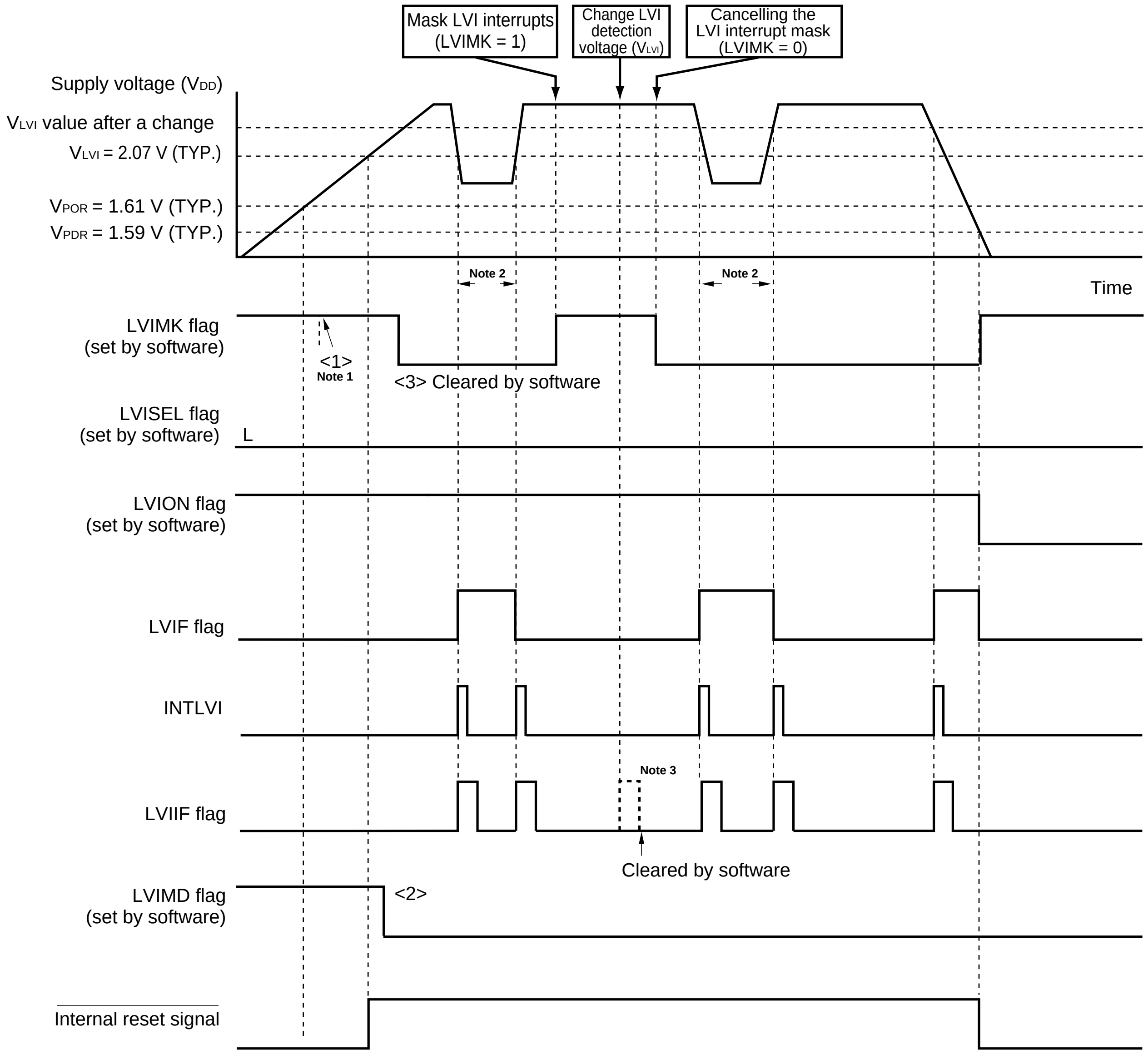
    Clear LVION to 0.

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)


Mask LVI interrupts (LVIMK = 1)
Change LVI detection voltage (VLVI)
Cancelling the LVI interrupt mask (LVIMK = 0)
Supply voltage (VDD)
VLVI value after a change
VLVI = 2.07 V (TYP.)
VPOR = 1.61 V (TYP.)
VPDR = 1.59 V (TYP.)
Note 2
Note 2
Time
LVIMK flag (set by software)
<1>
Note 1
<3> Cleared by software
LVISEL flag (set by software)
L
LVION flag (set by software)
LVIF flag
INTLVI
Note 3
LVIIF flag
Cleared by software
LVIMD flag (set by software)
<2>
Internal reset signal

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 410 μs).

  - Operation stabilization time (10 μs (MAX.))
  - Minimum pulse width (200 μs (MIN.))
  - Detection delay time (200 μs (MAX.))

  <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.

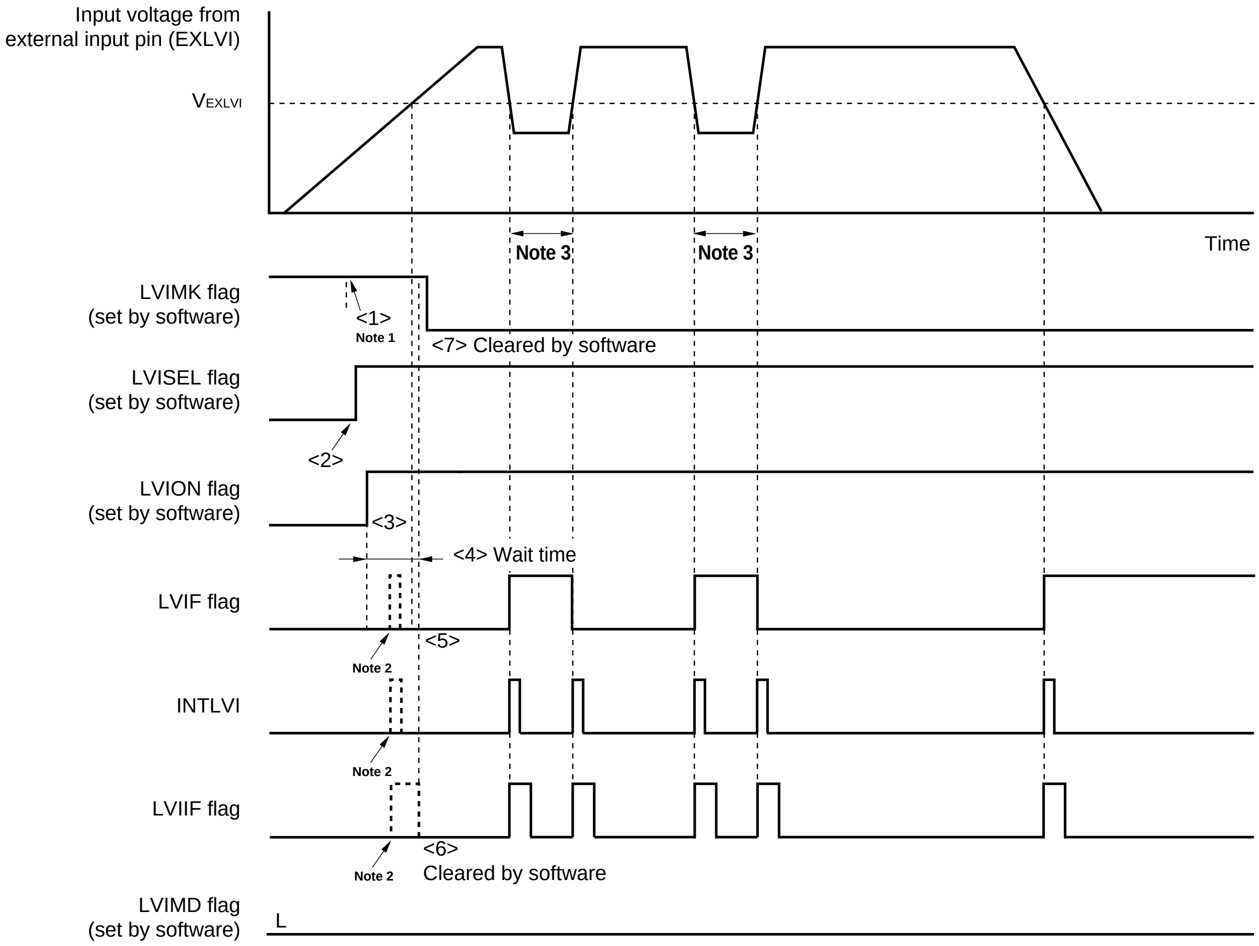
  Caution Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

Figure 19-10. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 1)


Input voltage from external input pin (EXLVI)
VEXLVI
Time
Note 3
Note 3
LVIMK flag (set by software)
<1>
Note 1
<7> Cleared by software
LVISEL flag (set by software)
<2>
LVION flag (set by software)
<3>
<4> Wait time
LVIF flag
Note 2
<5>
INTLVI
Note 2
LVIIF flag
Note 2
<6>
Cleared by software
LVIMD flag (set by software)
L

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>

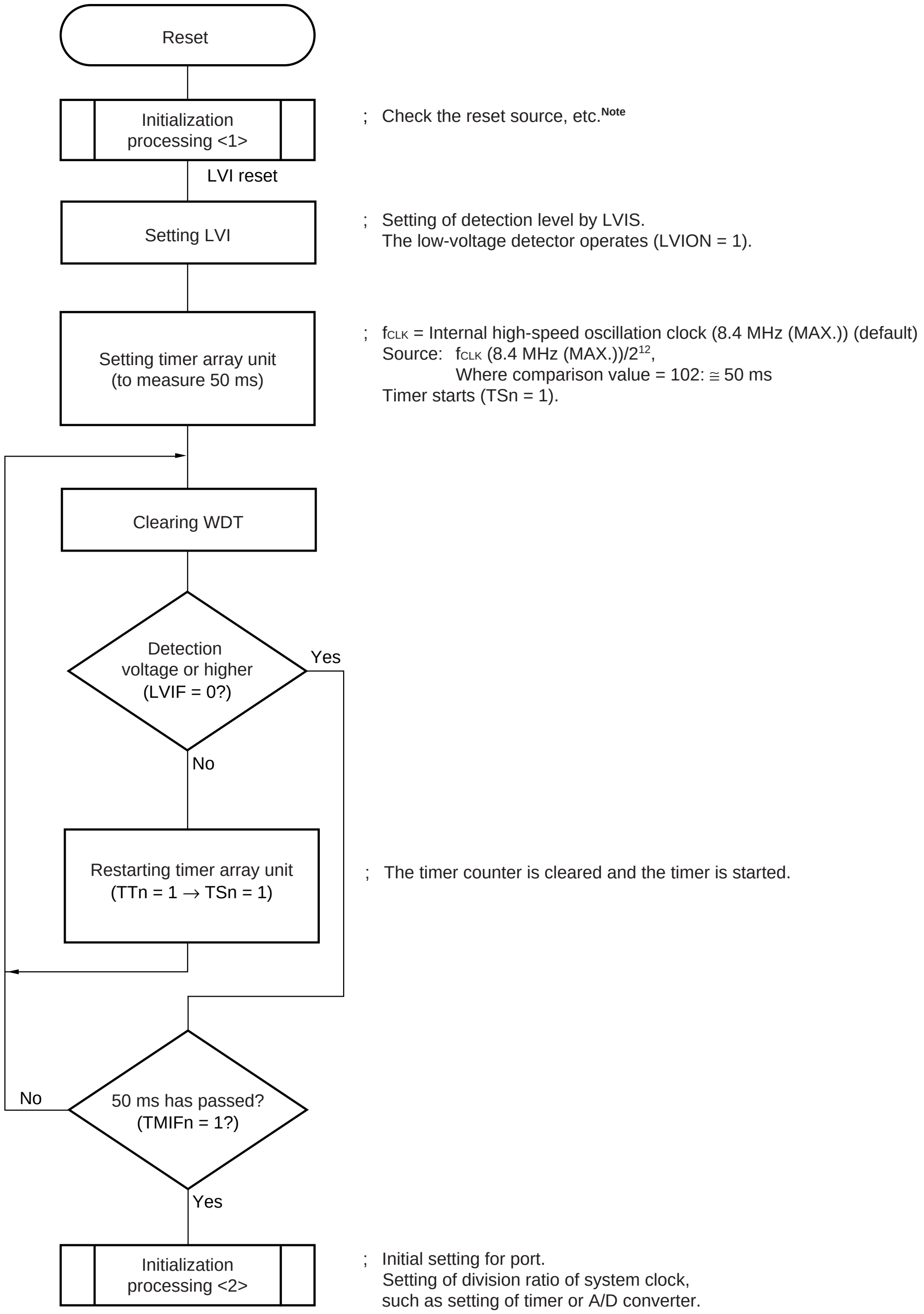
After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 19-11**).

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


Reset
Initialization processing <1>
; Check the reset source, etc.Note
LVI reset
Setting LVI
; Setting of detection level by LVIS.
The low-voltage detector operates (LVION = 1).
Setting timer array unit (to measure 50 ms)
; fCLK = Internal high-speed oscillation clock (8.4 MHz (MAX.)) (default)
Source: fCLK (8.4 MHz (MAX.))/2^12,
Where comparison value = 102: ≅ 50 ms
Timer starts (TSn = 1).
Clearing WDT
Detection voltage or higher (LVIF = 0?)
Yes
No
Restarting timer array unit (TTn = 1 → TSn = 1)
; The timer counter is cleared and the timer is started.
No
50 ms has passed? (TMIFn = 1?)
Yes
Initialization processing <2>
; Initial setting for port.
Setting of division ratio of system clock,
such as setting of timer or A/D converter.

Note A flowchart is shown on the next page.

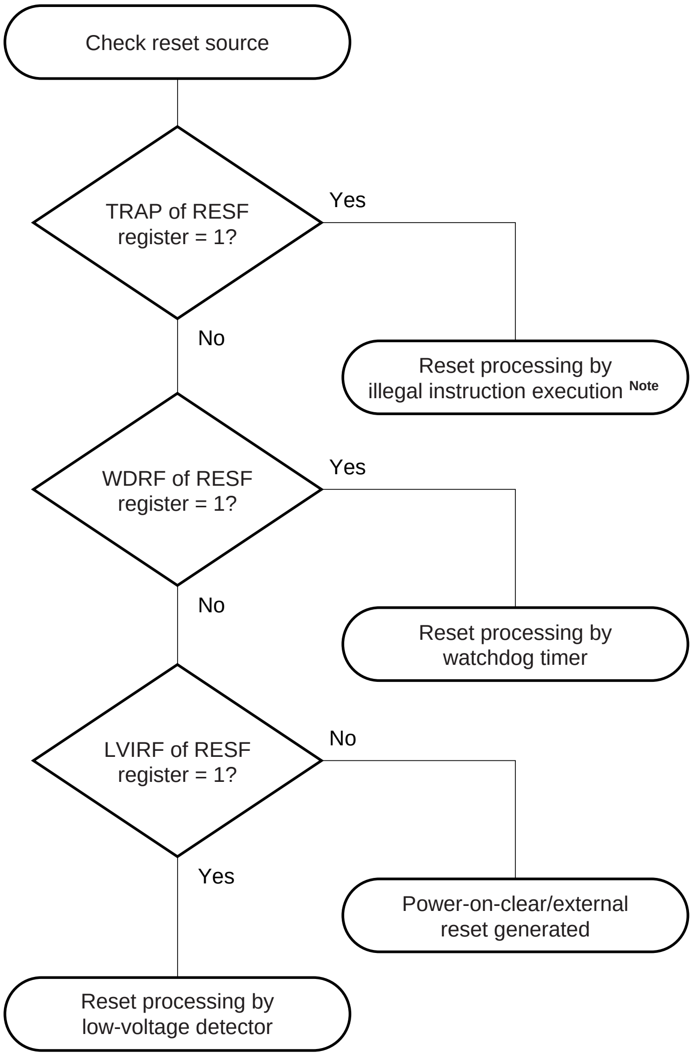
Remarks 1. If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

2. n = 00 to 11

Figure 19-11. Example of Software Processing After Reset Release (2/2)

- Checking reset source


Check reset source
TRAP of RESF register = 1?
Yes
No
Reset processing by illegal instruction execution Note
WDRF of RESF register = 1?
Yes
No
Reset processing by watchdog timer
LVIRF of RESF register = 1?
No
Yes
Power-on-clear/external reset generated
Reset processing by low-voltage detector

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.

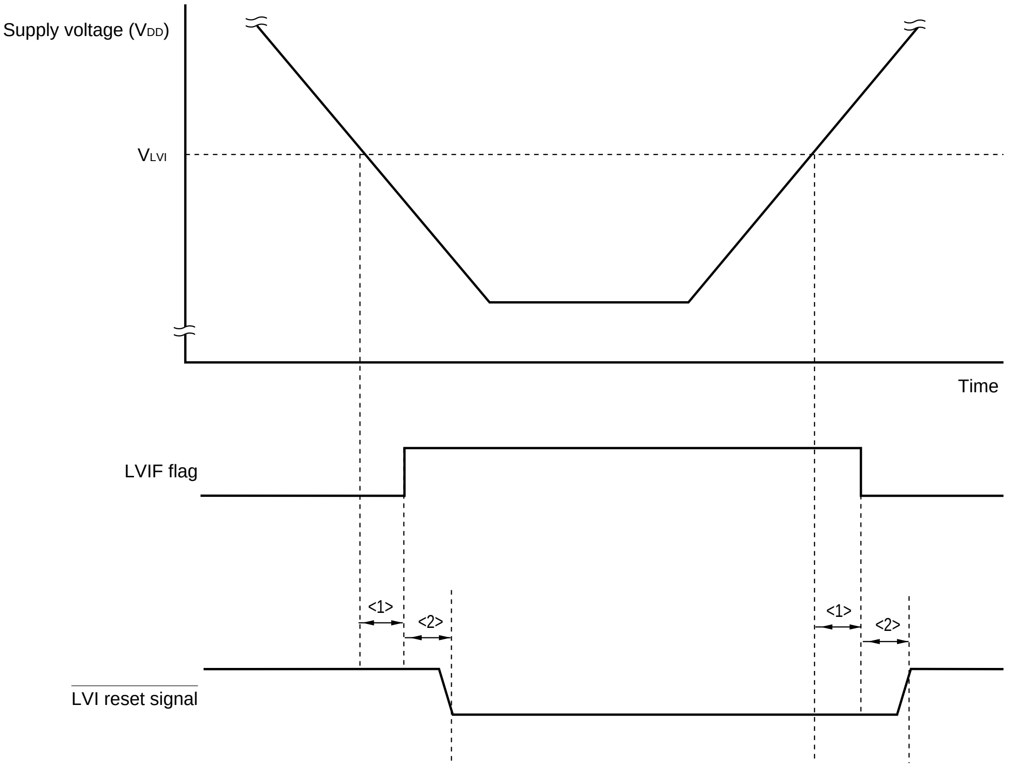
- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

(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.))

<2>: Detection delay time (200 μs (MAX.))

CHAPTER 20 REGULATOR

20.1 Regulator Overview

The μPD79F9211 contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF: target). However, when using the STOP mode that has been entered since operation of the internal high-speed oscillation clock and external main system clock, 0.47 μF is recommended. 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

Cautions 1. The RMC register can be rewritten only in the low consumption current mode (refer to Table 20-1). In other words, rewrite this register during CPU operation with the subsystem clock (f_{XT}) while the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the double-speed mode internal high-speed oscillation clock (f_{DSC}) are both stopped.

2. 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 $f_{CLK} \leq 5$ MHz
<When the high-speed internal oscillation clock, external input clock, or subsystem clock are selected for the CPU clock> $f_{CLK} \leq 5$ MHz

3. The self-programming function is disabled in the low consumption current mode.

Table 20-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
Low consumption current mode	1.8 V	During system reset
		In STOP mode (except during OCD mode)
		When both the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the double-speed mode internal high-speed oscillation clock (f_{DSC}) are stopped during CPU operation with the subsystem clock (f_{XT})
		When both the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the double-speed mode internal high-speed oscillation clock (f_{DSC}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{XT}) has been set
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 μPD79F9211 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).

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
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note 2]
0	0	25%
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} = 33 kHz (MAX.))
0	0	0	$2^{10}/f_{IL}$ (31.03 ms)
0	0	1	$2^{11}/f_{IL}$ (62.06 ms)
0	1	0	$2^{12}/f_{IL}$ (124.1 ms)
0	1	1	$2^{13}/f_{IL}$ (248.2 ms)
1	0	0	$2^{15}/f_{IL}$ (992.9 ms)
1	0	1	$2^{17}/f_{IL}$ (3.971 s)
1	1	0	$2^{18}/f_{IL}$ (7.943 s)
1	1	1	$2^{20}/f_{IL}$ (31.17 s)

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 Option Byte (000C1H/010C1H)

Address: 000C1H/010C1H[Note]

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

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)

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Cautions 1. Be sure to set bits 7 to 1 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	Erases data of flash memory in case of failures in enabling on-chip debugging and authenticating on-chip debug security ID.
1	1	Does not erases data of flash memory in case of failures in enabling on-chip debugging and 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

Set the user option byte and on-chip debug option byte using a linker option of assembler package RA78K0R.
For how to set the option byte, refer to **RA78K0R Assembler Package User's Manual**.

Remark The option byte is referenced during reset processing. For the timing of reset processing, see **CHAPTER 17 RESET FUNCTION**.

The μPD79F9211 incorporates 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

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 μPD79F9211 has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the μPD79F9211 is mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 22-1. Wiring Between μPD79F9211 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer			44-pin GB Package	
Signal Name	I/O	Pin Function	Pin Name	Pin No.
SI/RxD	Input	Receive signal	TOOL0/P40	2
SO/TxD	Output	Transmit signal		
SCK	Output	Transfer clock	–	–
CLK	Output	Clock output	–	–
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	3
FLMD0	Output	Mode signal	FLMD0	6
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	11
			AV_{REF}	32
GND	–	Ground	V_{SS}	10
			AV_{SS}	33

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 (MC Package)

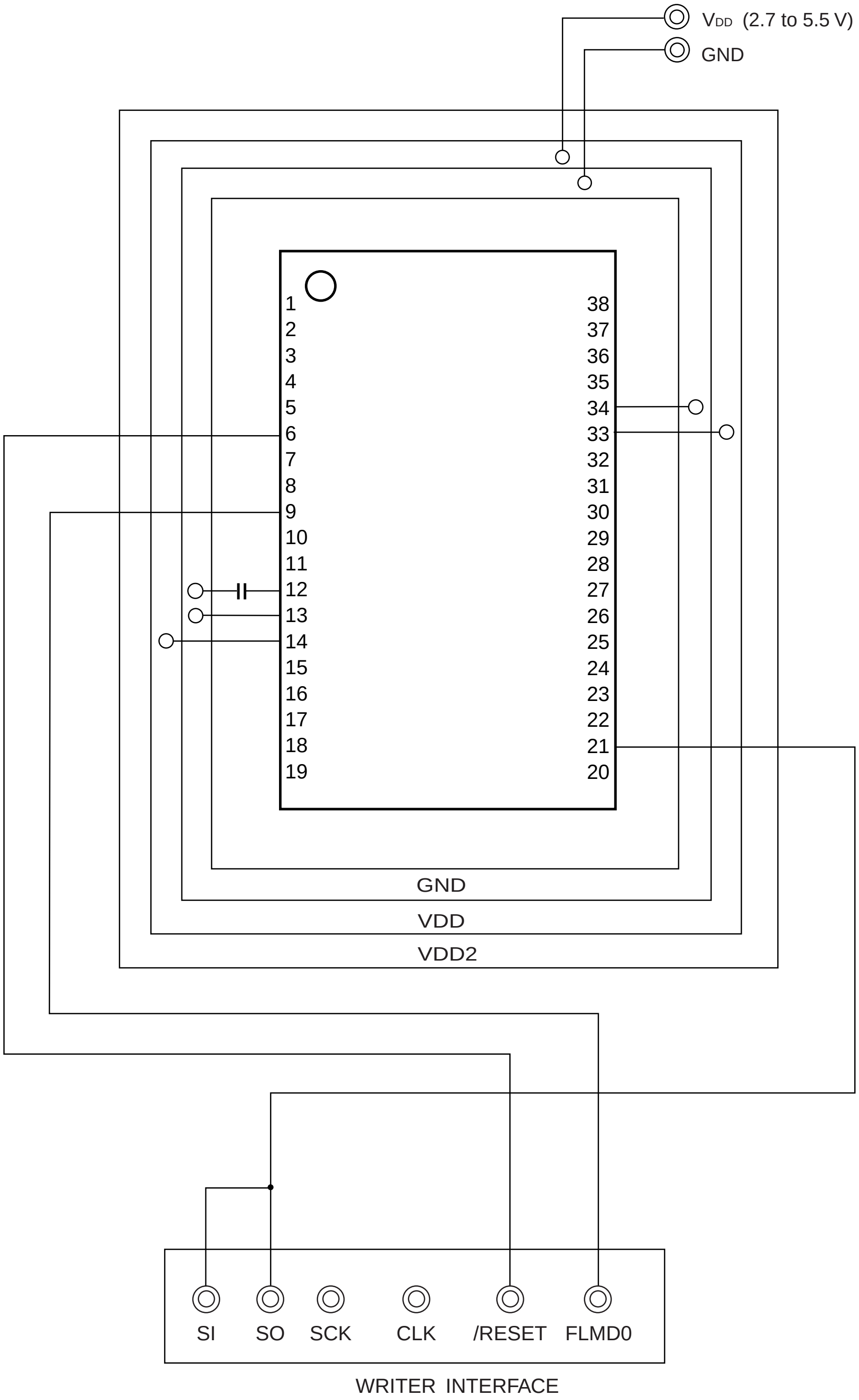


Figure 22-2. Example of Wiring Adapter for Flash Memory Writing (GB Package)

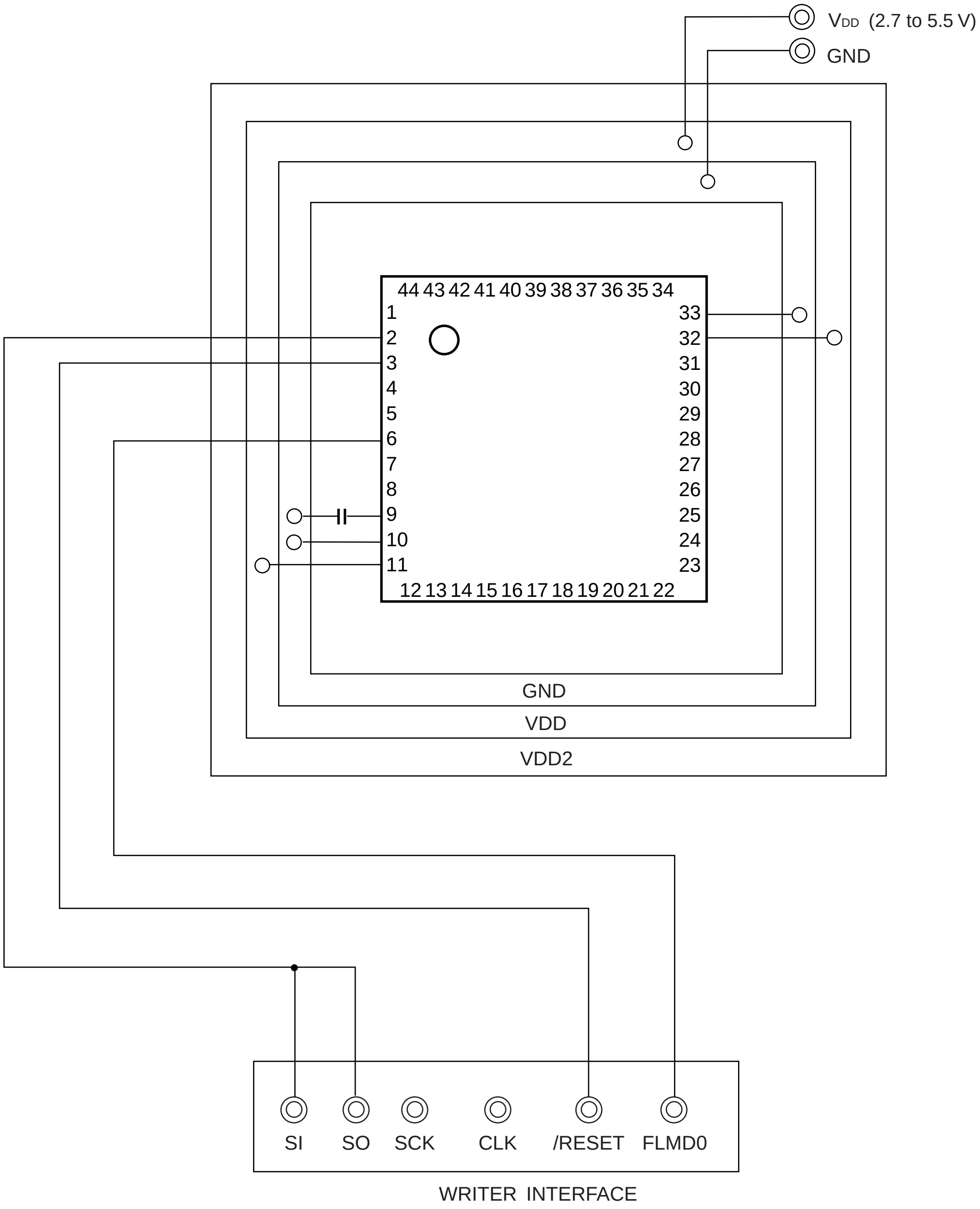
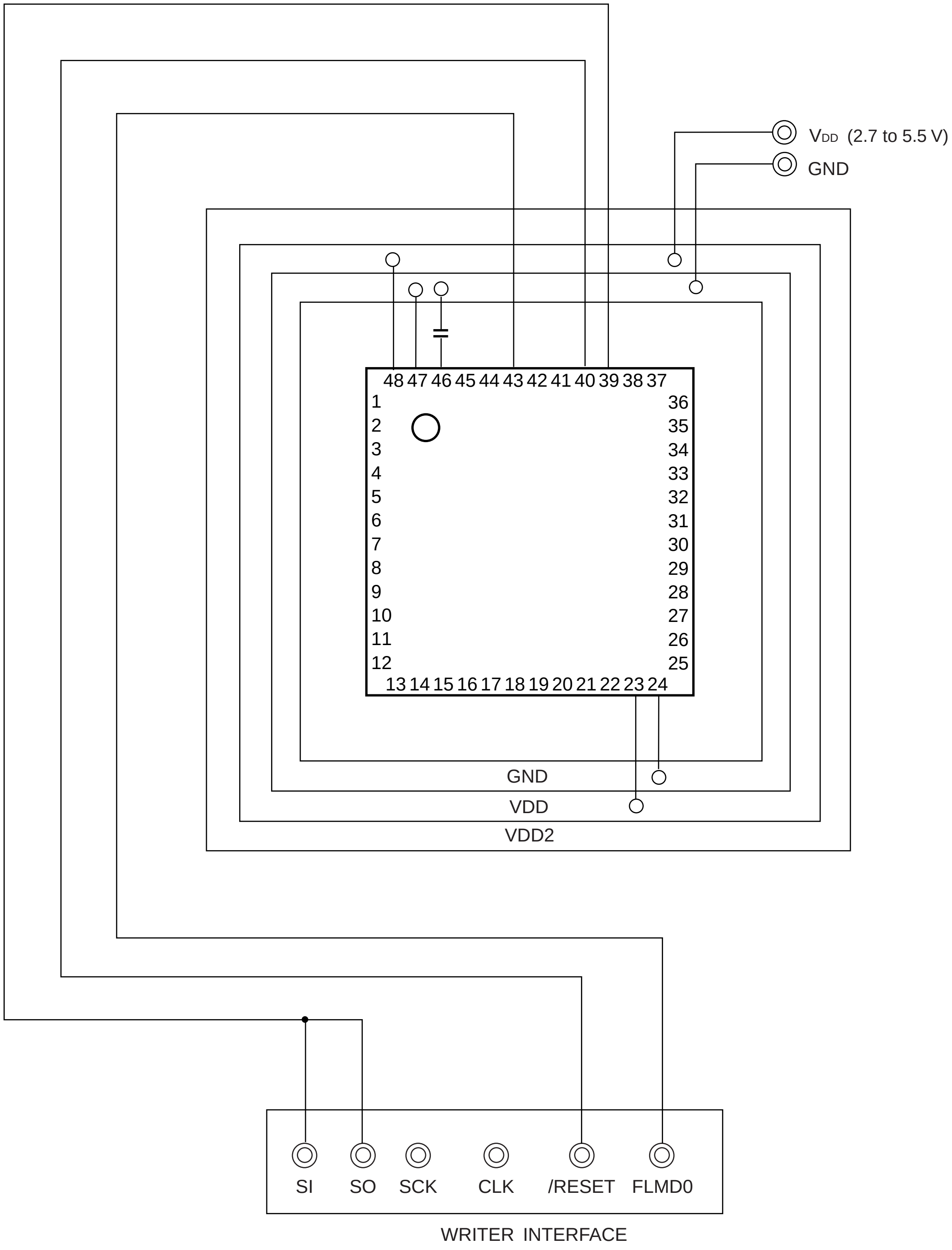


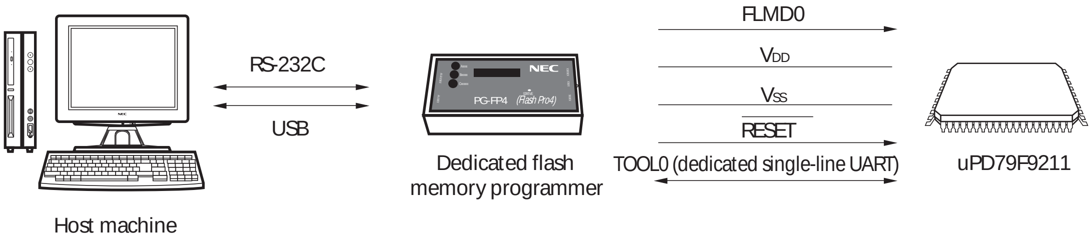
Figure 22-3. Example of Wiring Adapter for Flash Memory Writing (GA Package)



22.2 Programming Environment

The environment required for writing a program to the flash memory of the μPD79F9211 is illustrated below.

Figure 22-4. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

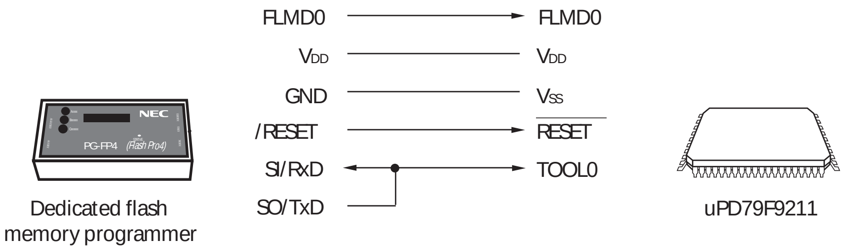
To interface between the dedicated flash memory programmer and the μPD79F9211, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

22.3 Communication Mode

Communication between the dedicated flash memory programmer and the μPD79F9211 is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the μPD79F9211.

Transfer rate: 115,200 bps to 1,000,000 bps

Figure 22-5. Communication with Dedicated Flash Memory Programmer



When using the FlashPro4 as the dedicated flash memory programmer, the FlashPro4 generates the following signals for the μPD79F9211. For details, refer to the user's manual for the FlashPro4.

Table 22-2. Pin Connection

FlashPro4			μPD79F9211	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}, AV_{REF}	◎
GND	–	Ground	V_{SS}, AV_{SS}	◎
CLK	Output	Clock output	–	×
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎
SI/RxD	Input	Receive signal	TOOL0	◎
SO/TxD	Output	Transmit signal		
SCK	Output	Transfer clock	–	×

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.

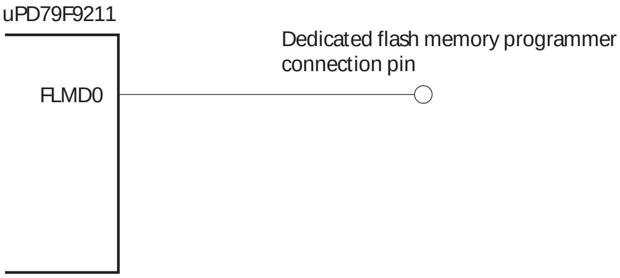
Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS} pin.

(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 22-6. 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} via an external resistor.

When on-chip debugging is enabled in the normal operation mode, pull this pin up by connecting it to V_{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).

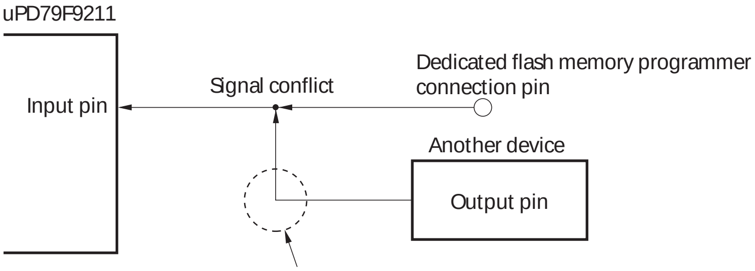
Remark The SAU pins are not used for communication between the μPD79F9211 and dedicated flash memory programmer, because single-line UART is used.

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 22-7. Signal Conflict ($\overline{\text{RESET}}$ Pin)



In the flash memory programming mode, a signal output by another device will conflict with the signal output by the dedicated flash memory programmer. Therefore, isolate the signal of another device.

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} via a resistor.

22.4.5 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF: target) in the same manner as during normal operation. However, when using the STOP mode that has been entered since operation of the internal high-speed oscillation clock and external main system clock, 0.47 μF is recommended. 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} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

However, when using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

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 22-8. Format of Background Event Control Register (BECTL)

Address: FFFBEH After reset: 00H R/W

Symbol	$\boxed{7}$	6	5	4	3	2	1	0
BECTL	FLMDPUP	0	0	0	0	0	0	0

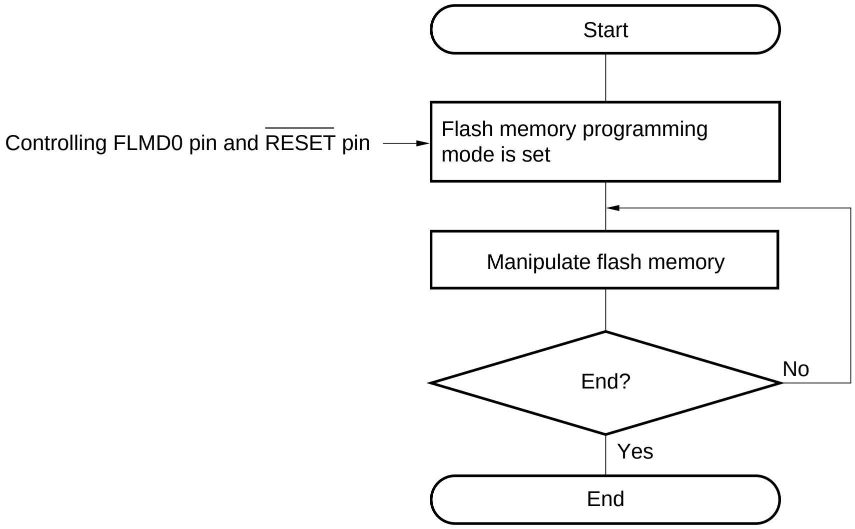
FLMDPUP	Software control of FLMD0 pin
0	Selects pull-down
1	Selects pull-up

22.6 Programming Method

22.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 22-9. Flash Memory Manipulation Procedure


Start
Controlling FLMD0 pin and RESET pin
Flash memory programming mode is set
Manipulate flash memory
End?
No
Yes
End

22.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the μPD79F9211 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 22-10. Flash Memory Programming Mode

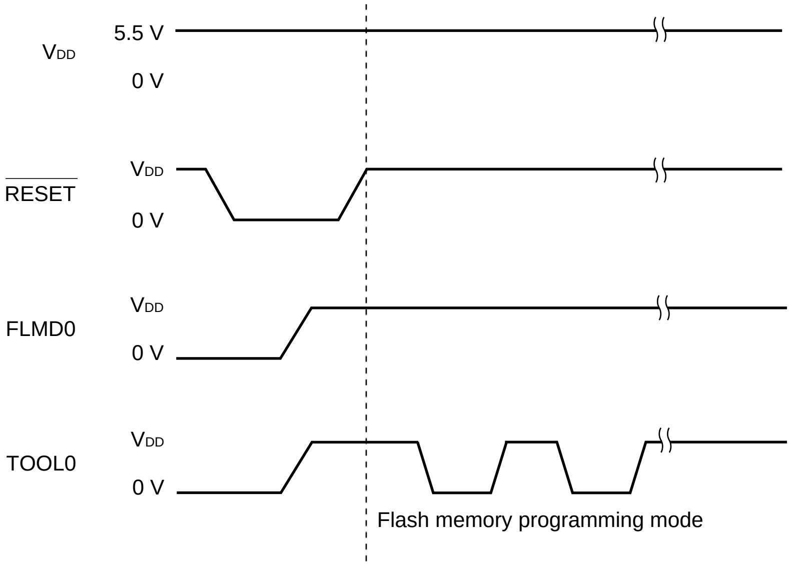


Table 22-3. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0	Normal operation mode
V_{DD}	Flash memory programming mode

22.6.3 Selecting communication mode

Communication mode of the μPD79F9211 as follows.

Table 22-4. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed	Frequency	Multiply Rate	
1-line mode (dedicated single-line UART)	UART-ch0	1 Mbps[Note 2]	–	–	TOOL0

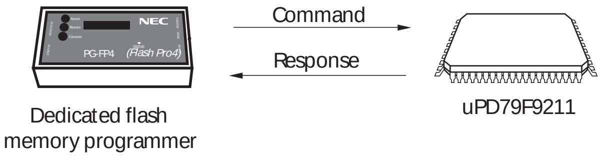
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

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 μPD79F9211S communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the μPD79F9211 are called commands, and the signals sent from the μPD79F9211 to the dedicated flash memory programmer are called response.

Figure 22-11. Communication Commands



The flash memory control commands of the μPD79F9211 are listed in the table below. All these commands are issued from the programmer and the μPD79F9211 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 μPD79F9211 information (such as the part number and flash memory configuration).
	Version Get	Gets the μPD79F9211 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 μPD79F9211 return a response for the command issued by the dedicated flash memory programmer. The response names sent from the μPD79F9211 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 μPD79F9211 supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command. The security setting is valid when the programming mode is set next.

- Disabling 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 batch erase (chip erase) command, block erase command, and write command on boot cluster 0 in the flash memory is prohibited by this setting.

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 μPD79F9211 security function is enabled.

Remark To prohibit writing and erasing during self-programming, use the flash sealed window function (see **22.8.2** for detail).

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 sealed window function (see **22.8.2** for detail).

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 μPD79F9211 supports a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the μPD79F9211 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. The self-programming function cannot be used when the CPU operates with the subsystem clock.

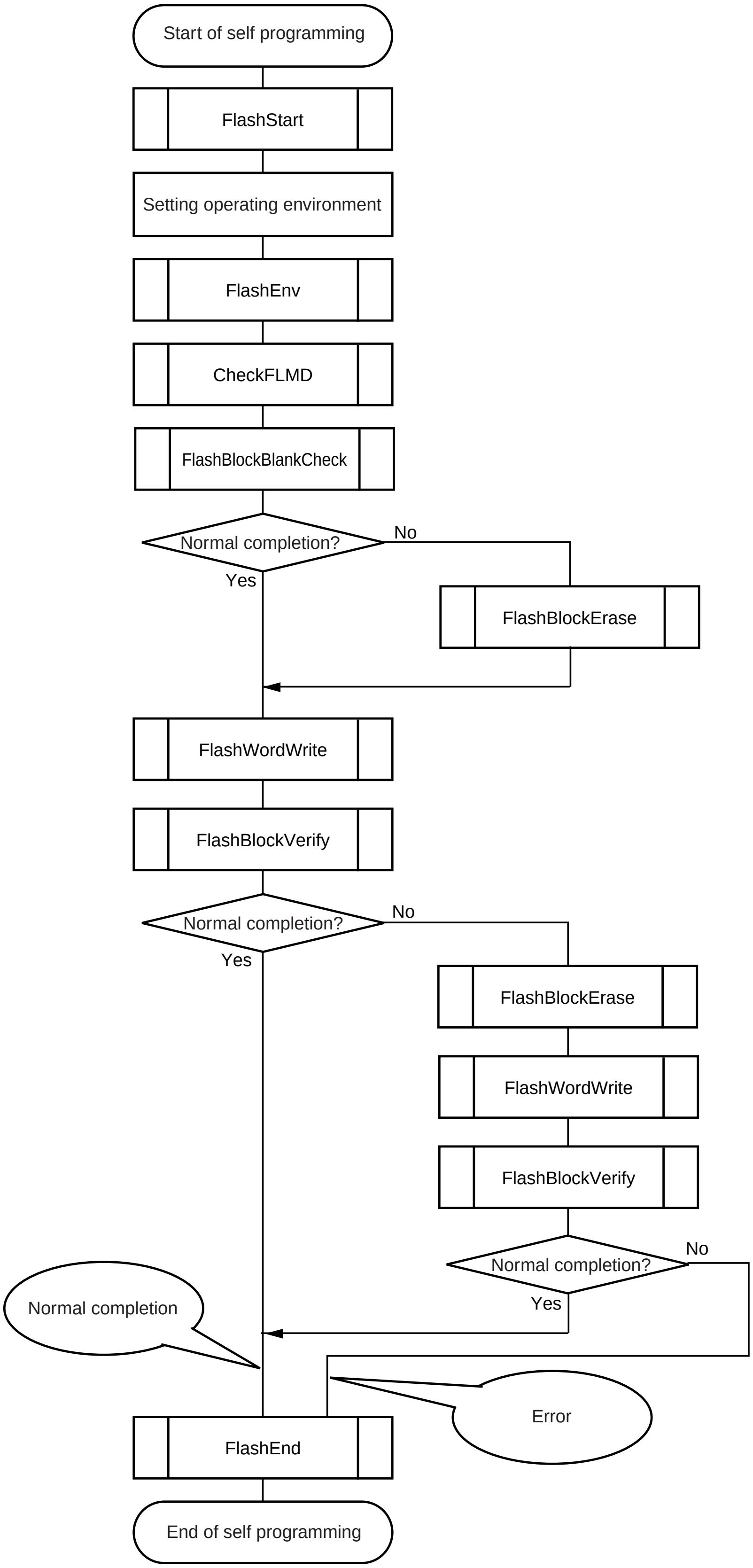
2. In the self-programming mode, call the self-programming start library (FlashStart).

3. 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.

4. The self-programming function is disabled in the low consumption current mode. For details of the low consumption current mode, see CHAPTER 20 REGULATOR.

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

Figure 22-12. Flow of Self Programming (Rewriting Flash Memory)



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.

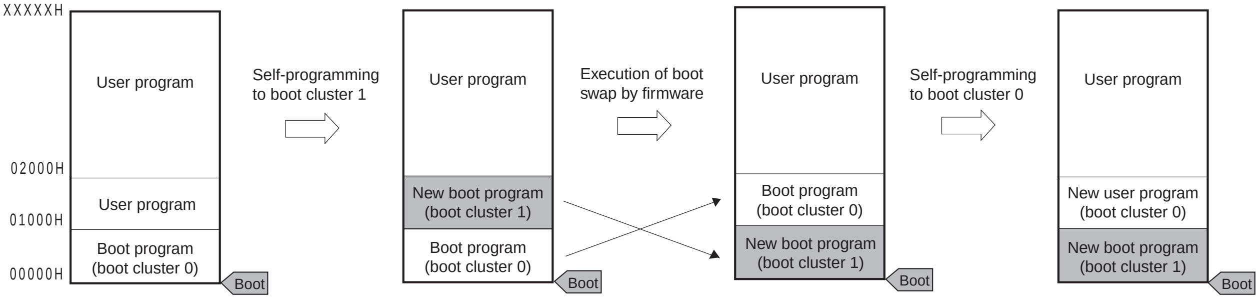
The boot swap function is used to avoid this problem.

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 μPD79F9211, 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 22-13. Boot Swap Function

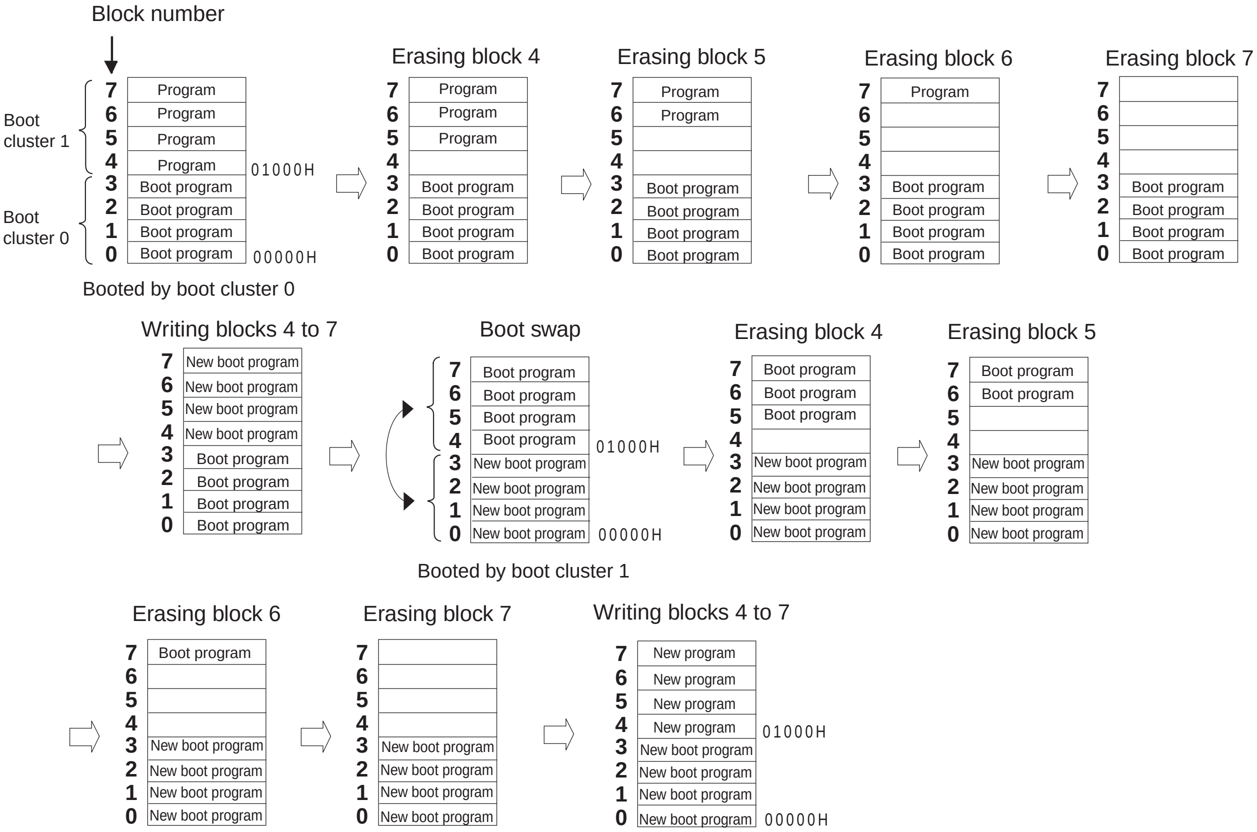

XXXXXH
02000H
01000H
00000H
User program
User program
Boot program (boot cluster 0)
Boot
Self-programming to boot cluster 1
User program
New boot program (boot cluster 1)
Boot program (boot cluster 0)
Boot
Execution of boot swap by firmware
User program
Boot program (boot cluster 0)
New boot program (boot cluster 1)
Boot
Self-programming to boot cluster 0
User program
New user program (boot cluster 0)
New boot program (boot cluster 1)
Boot

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 22-14. 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.

Writing and erasing to the flash memory within the range specified as a window are enabled during self-programming, and writing and erasing to the flash memory outside the specified range are prohibited.

The window range can be expanded or reduced by setting and change during on-board/off-board programming and self-programming. However, the shield function becomes effective only during self-programming. In on-board/off-board programming, writing and erasing to the flash memory outside the window range are enabled.

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 range of window range.
On-board/Off-board programming	Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc.	Block erasing is enabled also outside the window range.	Writing is enabled also outside the window range.

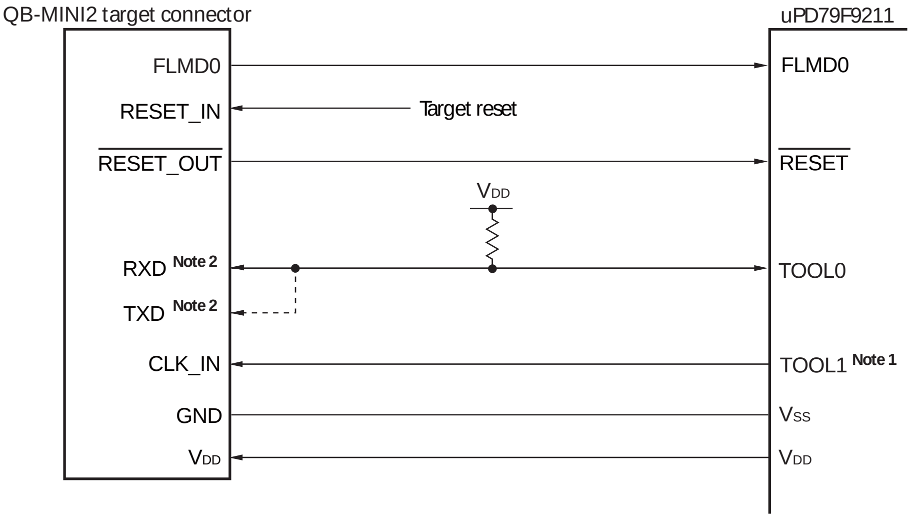
Remark See **22.7 Security Settings** to prohibit writing/erasing during on-board/off-board programming.

23.1 Connecting QB-MINI2 to μPD79F9211

The μPD79F9211 uses the V_{DD}, FLMD0, $\overline{\text{RESET}}$, TOOL0, TOOL1[Note 1], and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2).

Caution The μPD79F9211 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 23-1. Connection Example of QB-MINI2 and μPD79F9211



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 2-2 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.

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. Lists the Differences Between 1-line Mode and 2-line Mode.

Communicat ion mode	Flash memory programming function	Debugging function
1-line mode	Available	• Pseudo real-time RAM monitor (RRM) function not supported. • DMM function (rewriting memory in RUN) not supported. • The debugger speed is two to four times slower than 2-line mode.
2-line mode	None	• Pseudo real-time RAM monitor (RRM) function supported • DMM function (rewriting memory in RUN) supported

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 μPD79F9211 has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 21 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 010C3H and 010C4H to 010CDH in advance, because 000C3H, 000C4H to 000CDH and 010C3H, and 010C4H to 010CDH are switched.

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

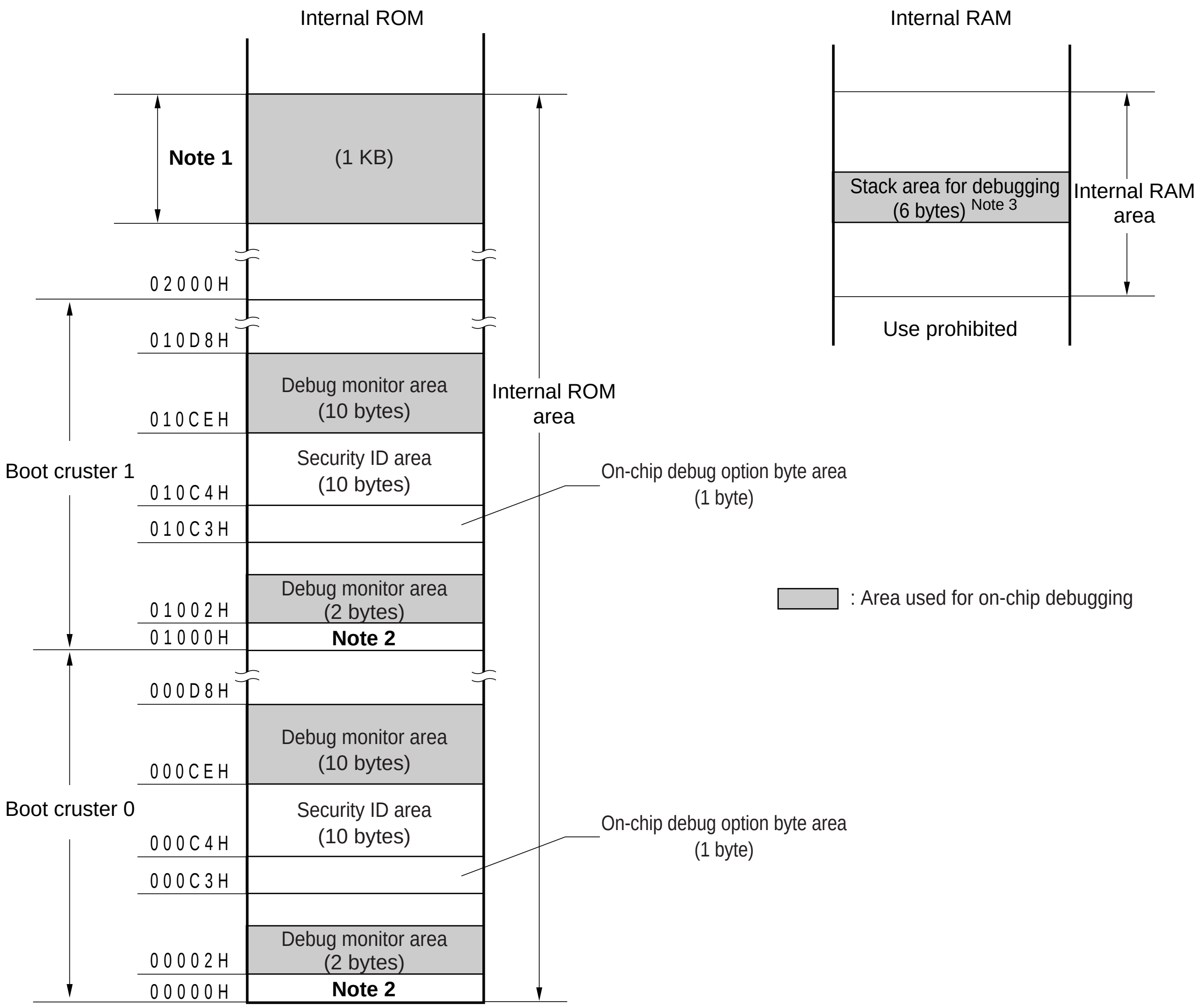
To perform communication between the μPD79F9211 and QB-MINI2, as well as each debug function, the securing of memory space must be done beforehand.

If NEC Electronics assembler RA78K0R or compiler CC78K0R is used, the items can be set by using linker options.

(1) Securement of memory space

The shaded portions in Figure 23-2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 23-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products ( ): Internal ROM	Address of **Note 1**
μPD79F9211 (16 KB)	03C00H-03FFFH

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 register.

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 Register	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 Register	AC Flag	BCDADJ Register
`MOV A, #85H        ; <1>`	85H	–	–	–
`ADD A, #15H        ; <2>`	9AH	0	0	06H
`ADD A, !BCDADJ     ; <3>`	00H	1	1	–

Examples 3: 80 + 80 = 160

Instruction	A Register	CY Register	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 register.

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 Register	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) Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only^Note)
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 For special-function register symbol, see **Table 3-5 SFR List** and **Table 3-6 Extended SFR (2nd SFR) List**.

25.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 25-2. Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
⊻	Exclusive logical sum (exclusive OR)
¯	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

25.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

Table 25-3. Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
×	Set/cleared according to the result
R	Previously saved value is restored

25.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except rp = AX

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except rp = AX

4. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 ← A_0, A_{m-1} ← A_m) × 1			×
	ROL	A, 1	2	1	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			×
	RORC	A, 1	2	1	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
	ROLC	A, 1	2	1	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			×
	ROLWC	AX,1	2	1	–	(CY ← AX_{15}, AX_0 ← CY, AX_{m+1} ← AX_m) × 1			×
		BC,1	2	1	–	(CY ← BC_{15}, BC_0 ← CY, BC_{m+1} ← 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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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 ← $\overline{\text{CY}}$			×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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$, PC$_S$ ← 0000, PC$_H$ ← (0000, addr5 + 1), PC$_L$ ← (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)$_L$, PC$_S$ ← 0000, PC$_H$ ← (0007FH), PC$_L$ ← (0007EH), SP ← SP − 4, IE ← 0			
	RET	–	1	6	–	PC$_L$ ← (SP), PC$_H$ ← (SP + 1), PC$_S$ ← (SP + 2), SP ← SP + 4			
	RETI	–	2	6	–	PC$_L$ ← (SP), PC$_H$ ← (SP + 1), PC$_S$ ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R
	RETB	–	2	6	–	PC$_L$ ← (SP), PC$_H$ ← (SP + 1), PC$_S$ ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f$_{CPU}$) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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]	6/8	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/9	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

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, $addr20	4	3/5$^{\text{Note 3}}$	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5$^{\text{Note 3}}$	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5$^{\text{Note 3}}$	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5$^{\text{Note 3}}$	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5$^{\text{Note 3}}$	6/8	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6$^{\text{Note 3}}$	7/9	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5$^{\text{Note 3}}$	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5$^{\text{Note 3}}$	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5$^{\text{Note 3}}$	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	5/7$^{\text{Note 3}}$	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5$^{\text{Note 3}}$	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6$^{\text{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 or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. This indicates the number of clocks "when condition is not met/when condition is met".

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

3. n indicates the number of register banks (n = 0 to 3)

Cautions 1. These specifications show target values, which may change after device evaluation.

2. The 78K0R/IC3 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.5 to +6.5	V
	V_{SS}		−0.5 to +0.3	V
	AV_{REF}		−0.5 to V_{DD} +0.3Note 1	V
	AV_{SS}		−0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	−0.3 to 3.6 and −0.3 to V_{DD} +0.3Note 2	V
Input voltage	V_{I1}	P10 to P13, P30 to P32, P40, P41, P50 to P52, P70 to P75, P120 to P124, $\overline{\text{RESET}}$, FLMD0	−0.3 to V_{DD} +0.3Note 1	V
	V_{I3}	P20 to P27, P80 to P83, P150 to P151	−0.3 to AV_{REF} +0.3Note 1 and −0.3 to V_{DD} +0.3Note 1	V
Output voltage	V_{O1}	P10 to P13, P30 to P32, P40, P41, P50 to P52, P70 to P75, P120	−0.3 to V_{DD} +0.3Note 1	V
	V_{O2}	P20 to P27, P80 to P83, P150 to P151	−0.3 to AV_{REF} +0.3	V

Notes 1. Must be 6.5 V or lower.

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

Remarks . 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)

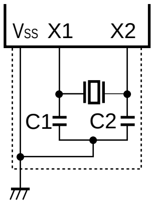
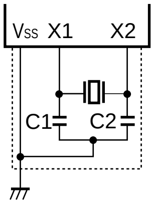
Parameter	Symbols	Conditions		Ratings	Unit
Analog input voltage	V_{AN}	ANI0 to ANI9, OAI, CMP0M, CMP0P, CMP1M, CMP1P		–0.3 to AV_{REF} +0.3[Note] and –0.3 to V_{DD} +0.3[Note]	V
Output current, high	I_{OH1}	Per pin	P10 to P13, P30 to P32, P40, P41, P50 to P52, P70 to P75, P120	–10	mA
		Total of all pins –80 mA	P40, P41, P120	–25	mA
			P10 to P13, P30 to P32, P50 to P52, P70 to P75	–55	mA
	I_{OH2}	Per pin	P20 to P27, P80 to P83, P150 to P151	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	P10 to P13, P30 to P32, P40, P41, P50 to P52, P70 to P75, P120	30	mA
		Total of all pins 200 mA	P40, P41, P120	60	mA
			P10 to P13, P30 to P32, P50 to P52, P70 to P75	140	mA
	I_{OL2}	Per pin	P20 to P27, P80 to P83, P150 to P151	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

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.**

Remarks. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

X1 Oscillator Characteristics
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (f_X)[Note]	2.7 V ≤ V_{DD} ≤ 5.5 V	2.0		20.0	MHz
Crystal resonator		X1 clock oscillation frequency (f_X)[Note]	2.7 V ≤ V_{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}.**
- **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.

Internal Oscillator Characteristics
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
Internal high-speed oscillation clock frequency Note 1	f_{IH1M}	2.7 V ≤ V_{DD} ≤ 5.5 V			1		MHz
	f_{IH8M}	2.7 V ≤ V_{DD} ≤ 5.5 V			8		MHz
	f_{IH20M}	2.7 V ≤ V_{DD} ≤ 5.5 V			20		MHz
	f_{IH40M}	2.7 V ≤ V_{DD} ≤ 5.5 V			40		MHz
Internal low-speed oscillation clock frequency	f_{IL}	Normal current mode	2.7 V ≤ V_{DD} ≤ 5.5 V	27	30	33	kHz
		Low consumption current mode Note 2		15	30	45	kHz

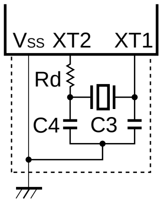
Notes 1. This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.

2. Regulator output is set to low consumption current mode in the following cases:

- When the RMC register is set to 5AH.
- During $\overline{\text{RESET}}$ pin reset.
- In STOP mode (except during OCD mode).
- When both the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the double-speed mode internal high-speed oscillation clock (f_{DSC}) are stopped during CPU operation with the subsystem clock (f_{XT})
- When both the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the double-speed mode internal high-speed oscillation clock (f_{DSC}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{XT}) has been set.

Remark For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 22 REGULATOR**.

XT1 Oscillator Characteristics
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Items	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})[Note]		32	32.768	35	kHz

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Cautions 1. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (1/8)

(T_A = –40 to +85°C, $2.7\ V \le V_{DD} \le 5.5\ V$, $2.7\ V \le AV_{REF} \le V_{DD}$, $V_{SS} = AV_{SS} = 0\ V$)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P10 to P13, P30 to P32, P40, P41, P50 to P52, P70 to P75, P120	$4.0\ V \le V_{DD} \le 5.5\ V$			–3.0	mA
			$2.7\ V \le V_{DD} < 4.0\ V$			–1.0	mA
		Total of P40, P41, P120 (When duty = 70% [Note 2])	$4.0\ V \le V_{DD} \le 5.5\ V$			–20.0	mA
			$2.7\ V \le V_{DD} < 4.0\ V$			–10.0	mA
		Total of P10 to P13, P30 to P32, P50 to P52, P70 to P75 (When duty = 70% [Note 2])	$4.0\ V \le V_{DD} \le 5.5\ V$			–30.0	mA
			$2.7\ V \le V_{DD} < 4.0\ V$			–19.0	mA
		Total of all pins (When duty = 60% [Note 2])	$4.0\ V \le V_{DD} \le 5.5\ V$			–50.0	mA
			$2.7\ V \le V_{DD} < 4.0\ V$			–29.0	mA
	I_{OH2}	Per pin for P20 to P27, P80 to P83, P150 to P151	$AV_{REF} \le V_{DD}$			–0.1	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} pin to an output pin.

2. Specification under conditions where the duty factor is 60% or 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_{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.

Caution P30 to P32, P70, P72, P73, and P75 do not output high level in N-ch open-drain mode.

Remarks . Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (2/8)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P30 to P32, P40, P41, P52, P70 to P75, P120	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			1.0	mA
		Per pin for P10 to P13, P50, P51	4.0 V ≤ V_{DD} ≤ 5.5 V			10.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			1.0	mA
		Total of P40, P41, P120 (When duty = 70% [Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
		Total of P10 to P13, P30 to P32, P50 to P52, P70 to P75 (When duty = 70% [Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			45.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			35.0	mA
		Total of all pins (When duty = 70% [Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			65.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			40.0	mA
	I_{OL2}	Per pin for P20 to P27, P80 to P83, P150 to P151	AV_{REF} ≤ V_{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} and AV_{SS} pin.

2. Specification under conditions where the duty factor is 60% or 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.

Remarks . Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (3/8)
(T_A = –40 to +85°C, 2.7 V $\leq V_{DD} \leq$ 5.5 V, 2.7 V $\leq AV_{REF} \leq V_{DD}$, $V_{SS} = AV_{SS}$ = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P30, P123, P124		$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	P10 to P13, P31, P32, P40, P41, P50 to P52, P70 to P75, P120 to P122, EXCLK, $\overline{\text{RESET}}$	Normal input buffer	$0.8V_{DD}$		V_{DD}	V
	V_{IH3}	P31, P32, P71, P72, P74, P75	TTL input buffer 4.0 V $\leq V_{DD} \leq$ 5.5 V	2.2		V_{DD}	V
			TTL input buffer 2.7 V $\leq V_{DD}$ < 4.0 V	2.0		V_{DD}	V
	V_{IH4}	P20 to P27, P81, P83, P150, P151	2.7 V $\leq AV_{REF} \leq V_{DD}$	$0.7AV_{REF}$		AV_{REF}	V
	V_{IH5}	P80, P82	2.7 V $\leq AV_{REF} \leq V_{DD}$	$0.8AV_{REF}$		AV_{REF}	V
	V_{IH7}	FLMD0		$0.9V_{DD}$ **Note**		V_{DD}	V

Note Must be $0.9V_{DD}$ or higher when used in the flash memory programming mode.

Caution The maximum value of V_{IH} of pins P30 to P32, P70, P72, P73, and P75 is V_{DD}, even in the N-ch open-drain mode.

Remark. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (4/8)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, low	V_{IL1}	P30, P123, P124		0		0.3V_{DD}	V
	V_{IL2}	P10 to P13, P31, P32, P40, P41, P50 to P52, P70 to P75, P120 to P122, EXCLK, $\overline{\text{RESET}}$	Normal input buffer	0		0.2V_{DD}	V
	V_{IL3}	P31, P32, P71, P72, P74, P75	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 2.7 V ≤ V_{DD} < 4.0 V	0		0.5	V
	V_{IL4}	P20 to P27, P81, P83, P150 to P151	2.7 V ≤ AV_{REF} ≤ V_{DD}	0		0.3AV_{REF}	V
	V_{IL5}	P80, P82	2.7 V ≤ AV_{REF} ≤ V_{DD}	0		0.2AV_{REF}	V
	V_{IL7}	FLMD0 Note		0		0.1V_{DD}	V

Note When disabling writing of the flash memory, connect the FLMD0 pin processing directly to V_{SS}, and maintain a voltage less than 0.1V_{DD}.

Remarks . Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (5/8)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P10 to P13, P30 to P32, P40, P41, P50 to P52, P70 to P75, P120	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = – 3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.0 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P27, P80 to P83, P150 to P151	AV_{REF} ≤ V_{DD}, I_{OH2} = –0.1 mA	AV_{REF} – 0.5			V
Output voltage, low	V_{OL1}	P10 to P13, P30 to P32, P40, P41, P50 to P52, P70 to P75, P120	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.0 mA			0.5	V
	V_{OL2}	P10 to P13, P50, P51	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 10.0 mA			1.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.0 mA			0.5	V
	V_{OL3}	P20 to P27, P80 to P83, P150 to P151	AV_{REF} = V_{DD}, I_{OL2} = 0.4 mA			0.4	V

Caution P30 to P32, P70, P72, P73, and P75 do not output high level in N-ch open-drain mode.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (6/8)
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

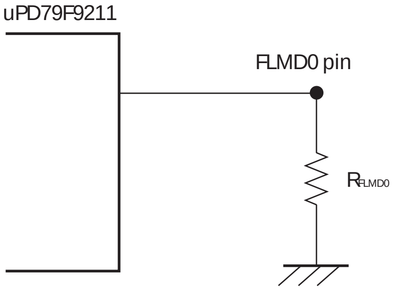
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P10 to P13, P30 to P32, P40, P41, P50 to P52 P70 to P75, P120, FLMD0, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P20 to P27, P80 to P83, P150 to P151	$V_I = AV_{REF}$, $AV_{REF} = V_{DD}$				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{DD}$	In input port			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P10 to P13, P30 to P32, P40, P41, P50 to P52, P70 to P75, P120, FLMD0, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL2}	P20 to P27, P80 to P83, P150 to P151	$V_I = V_{SS}$, $AV_{REF} = V_{DD}$				–1	μA
	I_{LIL3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{SS}$	In input port			–1	μA
				In resonator connection			–10	μA

Remarks . Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (7/8)
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
O-chip pll-up resistance	R_U	P10 to P13, P30 to P32, P40, P41, P50 to P52, P120	V_I = V_{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.

DC Characteristics (8/8)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1}	Operating mode				T.B.D	T.B.D	mA
	I_{DD2}	HALT mode				T.B.D	T.B.D	mA
	I_{DD3}	STOP mode				T.B.D	T.B.D	μA
RTC operating current	I_{RTC}[Notes 1, 2]	f_{SUB} = 32.768 kHz		V_{DD} = 3.0 V		0.2	1.0	μA
				V_{DD} = 2.0 V		0.2	1.0	μA
Watchdog timer operating current	I_{WDT}[Notes 2, 3]	f_{IL} = 30 kHz				5	10	μA
A/D converter operating current	I_{ADC}[Note 4]	During conversion at maximum speed,	High speed mode 1	AV_{REF} = V_{DD} = 5.0 V		T.B.D	T.B.D	mA
			High speed mode 2	AV_{REF} = V_{DD} = 3.0 V		T.B.D	T.B.D	mA
			Normal mode 1	AV_{REF} = V_{DD} = 5.0 V		T.B.D	T.B.D	mA
Operational amplifier operating current	I_{AMP} [Note 5]					T.B.D	T.B.D	μA
Comparator operating current	I_{CMP}[Note 6]	Per channel when the internal reference voltage is not used		AV_{REF} = V_{DD} = 5.0 V		T.B.D	T.B.D	μA
				AV_{REF} = V_{DD} = 3.0 V		T.B.D	T.B.D	μA
		Per channel when the internal reference voltage is used		AV_{REF} = V_{DD} = 5.0 V		T.B.D	T.B.D	μA
				AV_{REF} = V_{DD} = 3.0 V		T.B.D	T.B.D	μA
LVI operating current	I_{LVI} [Note 7]					9	18	μA

Notes 1. Current flowing only to the real-time counter (excluding the operating current of the XT1 oscillator). The current value of the 78K0R/IC3 is the TYP. value, the sum of the TYP. values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time counter operates in operation mode or HALT mode. The I_{DD1} and I_{DD2} MAX. values also include the real-time counter operating current.

2. When internal high-speed oscillator and high-speed system clock are stopped.

3. Current flowing only to the watchdog timer (including the operating current of the 30 kHz internal oscillator). The current value of the 78K0R/IC3 is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when f_{CLK} = f_{SUB}/2 when the watchdog timer operates in STOP mode.

4. Current flowing only to the A/D converter (AV_{REF} pin). The current value of the 78K0R/IC3 is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

5. Current flowing only to the operational amplifier (AV_{REF} pin). The current value of the 78K0R/IC3 is the sum of I_{DD1} or I_{DD2} and I_{AMP} when the operational amplifier operates in an operation mode or the HALT mode.

6. Current flowing only to the comparator (AV_{REF} pin). The current value of the 78K0R/IC3 is the sum of I_{DD1} or I_{DD2} and I_{CMP} when the comparator operates in an operation mode or the HALT mode.

7. Current flowing only to the LVI circuit. The current value of the 78K0R/IC3 is 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.

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

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

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

AC Characteristics

(1) Basic operation (1/6)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

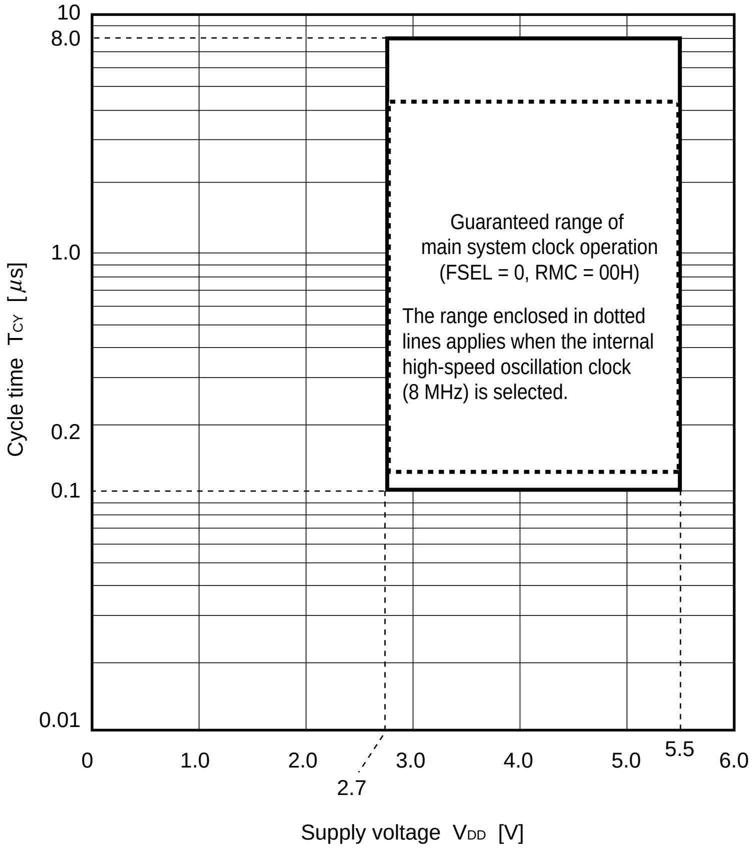
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	Normal current mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		8	μs
			Low consumption current mode		0.2		8	μs
		Subsystem clock (f_{SUB}) operation			57.2	61	62.5	μs
		In the self programming mode	Normal current mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		0.5	μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			2.0		20.0	MHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V			24			ns
TI02 to TI07, TI09 to TI11, SLTI input high-level width, low-level width	t_{TIH}, t_{TIL}	f_{MCK} = f_{IHM40}			2/f_{MCK}+10			ns
		Other than above			1/f_{MCK}+10			ns
TO02 to TO07, TO10, TO11, SLTO output frequency	f_{TO}	2.7 V ≤ V_{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}: Macro operation clock frequency
(Operation clock to be set by the CKSn bit of the TMRn register. n: Channel number (n = 00 to 11))

2. For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 22 REGULATOR**.

(1) Basic operation (2/6)

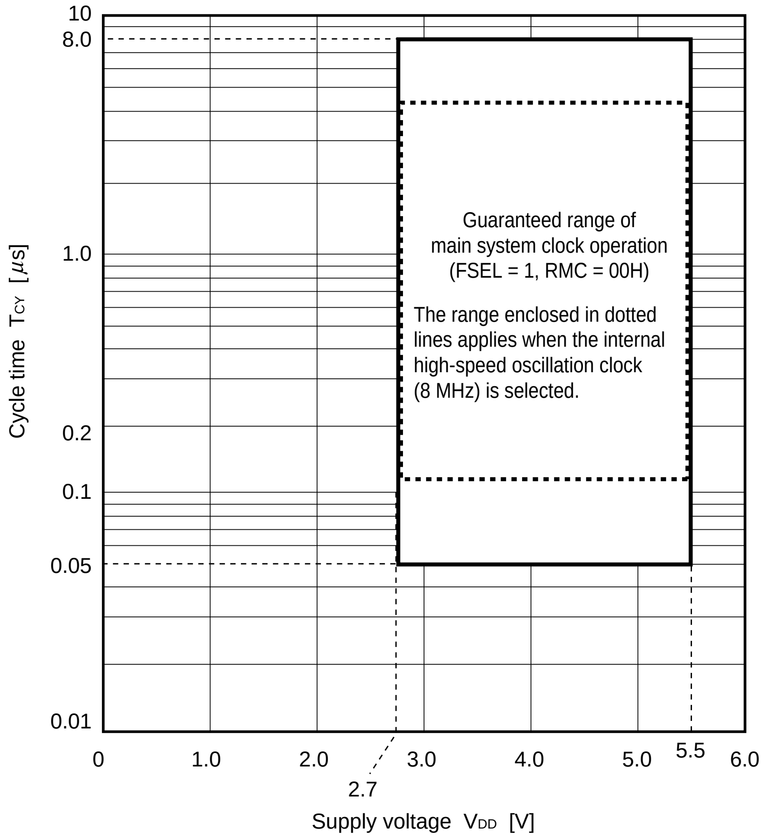
Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 00H)



Remark FSEL: Bit 0 of the operation speed mode control register (OSMC)

(1) Basic operation (3/6)

Minimum instruction execution time during main system clock operation (FSEL = 1, RMC = 00H)



Caution The following operations are prohibited when V_{DD} is less than 2.7 V.

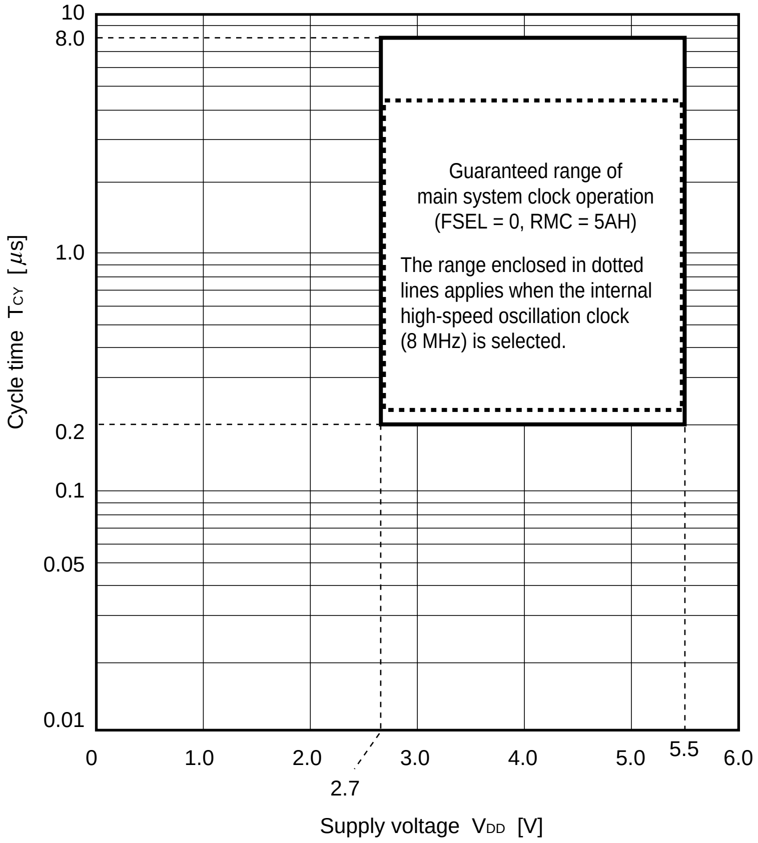
- **Operation rewriting FSEL from 0 to 1 when V_{DD} is less than 2.7 V**
- **Releasing STOP mode during f_{EX} operation and f_{IH} operation, when V_{DD} is less than 2.3 V and FSEL is set to 1**

(This must not be performed even if the frequency is divided. The STOP mode may be released during f_X operation.)

Remarks 1.	FSEL:	Bit 0 of the operation speed mode control register (OSMC)
2.	f_{IH}:	Internal high-speed oscillation clock frequency
	f_X:	X1 clock oscillation frequency
	f_{EX}:	External main system clock frequency

(1) Basic operation (4/6)

Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 5AH)

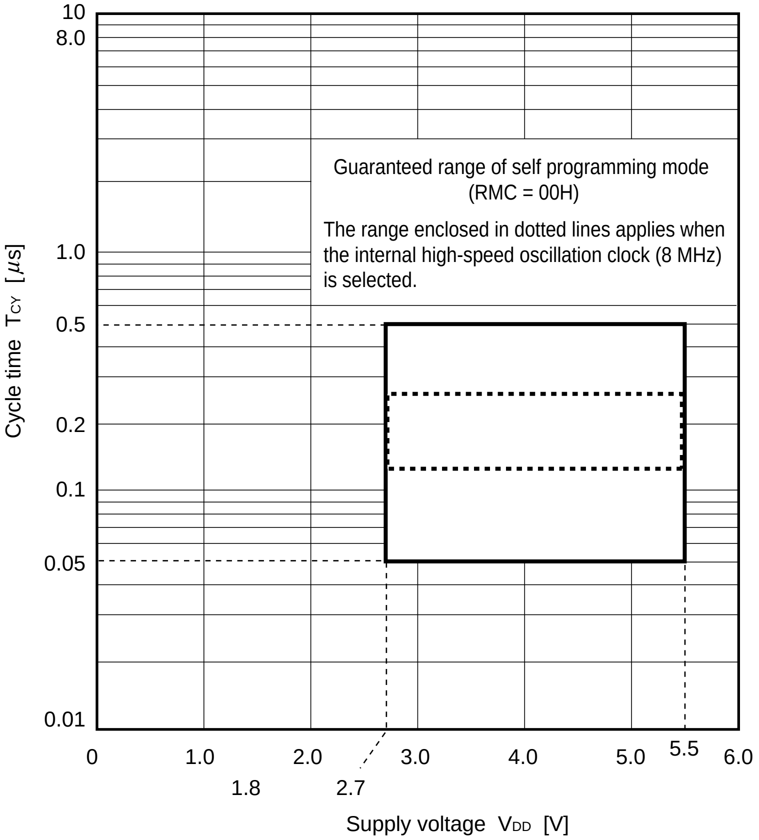


Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)

2. The entire voltage range is 5 MHz (MAX.) when RMC is set to 5AH.

(1) Basic operation (5/6)

Minimum instruction execution time during self programming mode (RMC = 00H)

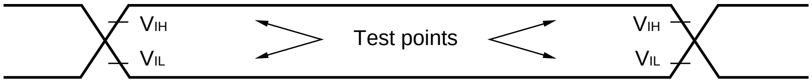


Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)

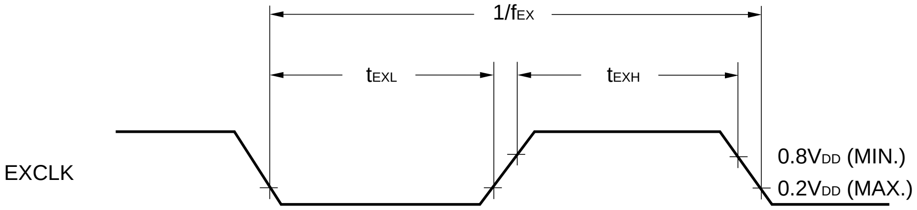
2. The self programming function cannot be used when RMC is set to 5AH or the CPU operates with the subsystem clock.

(1) Basic operation (6/6)

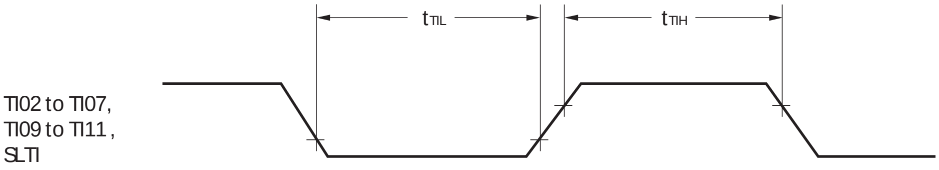
AC Timing Test Points



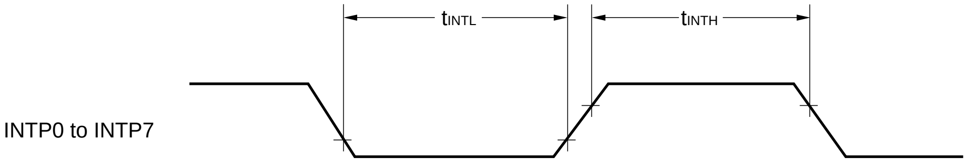
External Main System Clock Timing



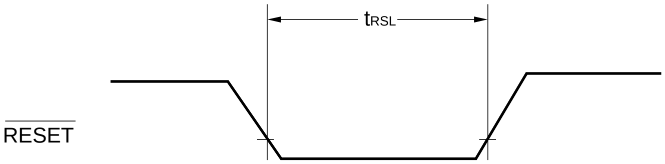
TI Timing



Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing

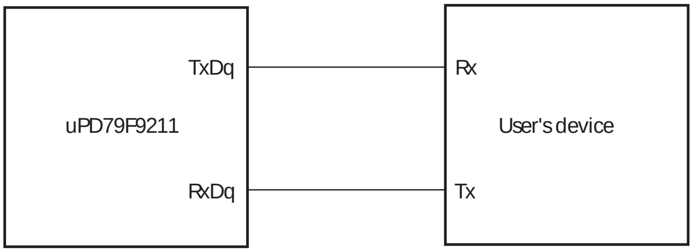


(2) Serial interface: Serial array unit (1/17)
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

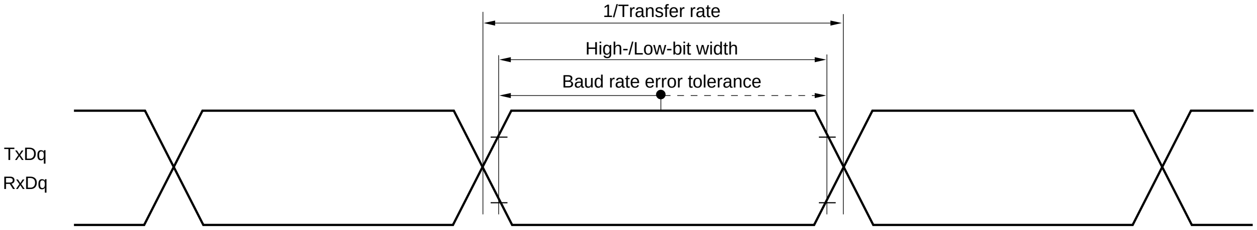
(a) During communication at same potential (UART mode) (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					$f_{MCK}/6$	Mbps
		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)



Caution Select the normal input buffer for RxDq and the normal output mode for TxDq by using the PIMg and POMg registers.

Remarks 1. q: UART number (q = 0, 1) , g: PIM and POM number (g = 3, 7)

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 to 3))

(2) Serial interface: Serial array unit (2/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

(b) During communication at same potential (CSI mode) (master mode, $\overline{\text{SCKp}}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V_{DD} < 4.0 V	400			ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	t_{KCY1}/2 – 20			ns
		2.7 V ≤ V_{DD} < 4.0 V	t_{KCY1}/2 – 35			ns
SIp setup time (to $\overline{\text{SCKp}}$↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V	70			ns
		2.7 V ≤ V_{DD} < 4.0 V	100			ns
SIp hold time (from $\overline{\text{SCKp}}$↑) Note 2	t_{KSI1}		30			ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output Note 3	t_{KSO1}	C = 50 pF Note 4			40	ns

Notes 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes “to $\overline{\text{SCKp}}$↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp hold time becomes “from $\overline{\text{SCKp}}$↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

3. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes “from $\overline{\text{SCKp}}$↑” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

4. C is the load capacitance of the $\overline{\text{SCKp}}$ and SOp output lines.

Caution Select the normal input buffer for SIp and the normal output mode for SOp and $\overline{\text{SCKp}}$ by using the PIMg and POMg registers.

Remarks 1. p: CSI number (p = 00, 01, 10),
g: PIM and POM number (g = 3, 7)

2. n: Channel number (n = 0 to 2)

(2) Serial interface: Serial array unit (3/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

(c) During communication at same potential (CSI mode) (slave mode, $\overline{\text{SCKp}}$... external clock input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY2}	16 MHz < f_{MCK}		$8/f_{MCK}$			ns
		f_{MCK} ≤ 16 MHz		$6/f_{MCK}$			ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH2}, t_{KL2}			$f_{KCY2}/2$			ns
SIp setup time (to $\overline{\text{SCKp}}$↑) Note 1	t_{SIK2}			$1/f_{MCK}$+80			ns
SIp hold time (from $\overline{\text{SCKp}}$↑) Note 2	t_{KSI2}			50			ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output Note 3	t_{KSO2}	C = 50 pF Note 4	4.0 V ≤ V_{DD} ≤ 5.5 V			$1/f_{MCK}$+120	ns
			2.7 V ≤ V_{DD} < 4.0 V			$1/f_{MCK}$+120	ns

Notes 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes “to $\overline{\text{SCKp}}$↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp hold time becomes “from $\overline{\text{SCKp}}$↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

3. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes “from $\overline{\text{SCKp}}$↑” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

4. C is the load capacitance of the $\overline{\text{SCKp}}$ and SOp output lines.

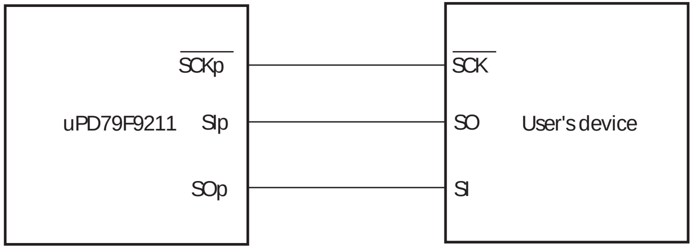
Caution Select the normal input buffer for SIp and $\overline{\text{SCKp}}$ and the normal output mode for SOp by using the PIMg and POMg registers.

Remarks 1. p: CSI number (p = 00, 01, 10),
g: PIM and POM number (g = 3, 7)

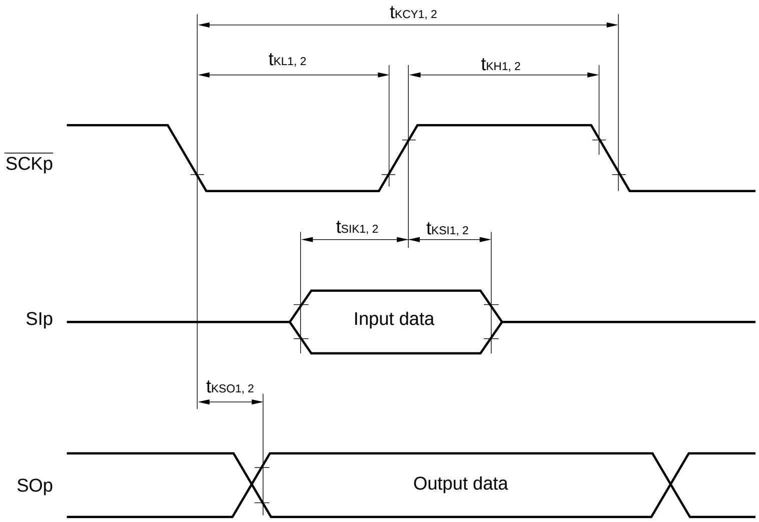
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of the SMR0n register. n: Channel number (n = 0 to 2))

(2) Serial interface: Serial array unit (4/17)

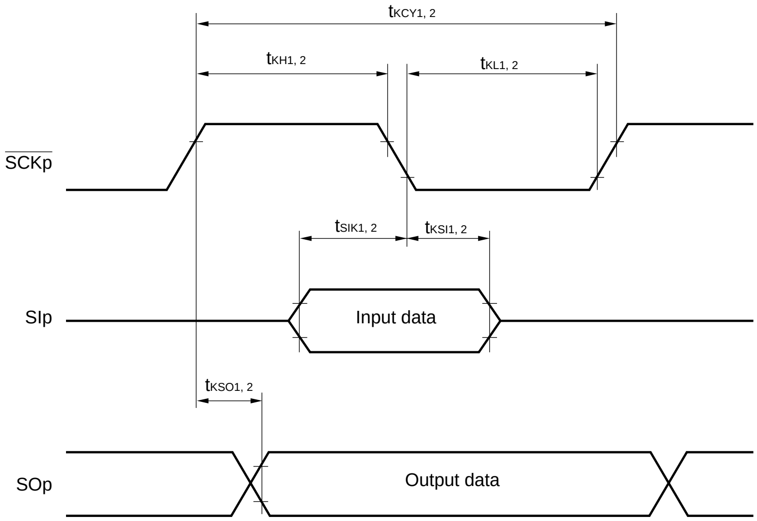
CSI mode connection diagram (during communication at same potential)



CSI mode serial transfer timing (during communication at same potential)
(When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1.)



CSI mode serial transfer timing (during communication at same potential)
(When DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.)



Remarks 1. p: CSI number (p = 00, 01, 10),

2. n: Channel number (n = 0 to 2))

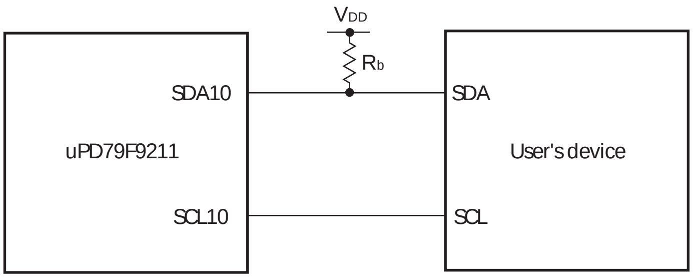
(2) Serial interface: Serial array unit (5/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

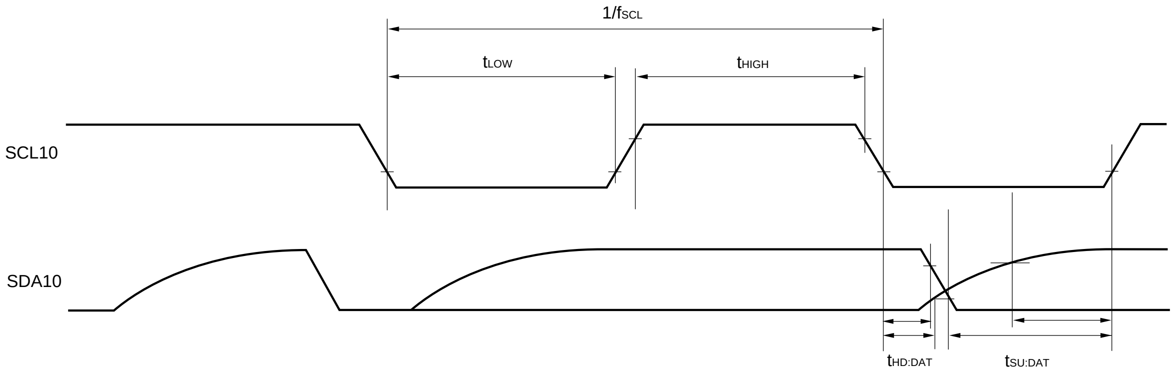
(d) During communication at same potentia (simplified I^{2}C mode)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCL10 clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ		1.0	MHz
		2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		480	kHz
Hold time when SCL10 = “L”	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	475		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	995		ns
Hold time when SCL10 = “H”	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	475		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	995		ns
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	1/f_{MCK}+85		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK}+120		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	0	130	ns
		2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	160	ns

Simplified I^{2}C 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} tolerance) mode for SDA10 and the normal output mode for SCL10 by using the PIM3 and POM3 registers.

Remarks 1. R_b[Ω]:Communication line (SDA10) pull-up resistance,
C_b[F]: Communication line (SCL10, SDA10) load capacitance

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS02 bit of the SMR02 register.)

(2) Serial interface: Serial array unit (6/17)
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

(e) Communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output) (1/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		reception	4.0 V ≤ V_{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
			2.7 V ≤ V_{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 RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMg registers.

Remarks 1. q: UART number (q = 0, 1) , g: PIM and POM number (g = 3, 7)

2. V_b[V]: Communication line voltage

3. 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 to 3))

4. 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} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (7/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

(e) Communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output) (2/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		transmission	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V				**Note 1**	bps
				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
			2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V				**Note 3**	bps
				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} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1- \frac{2.2}{V_b})\} \times 3}\ [\text{bps}]$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100\ [\%]$$

* 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 2.7 V ≤ V_{DD} ≤ 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1- \frac{2.0}{V_b})\} \times 3}\ [\text{bps}]$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100\ [\%]$$

* 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 RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMg registers.

(Remarks are given on the next page.)

(2) Serial interface: Serial array unit (8/17)

Remarks 1. R_b[Ω]:Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

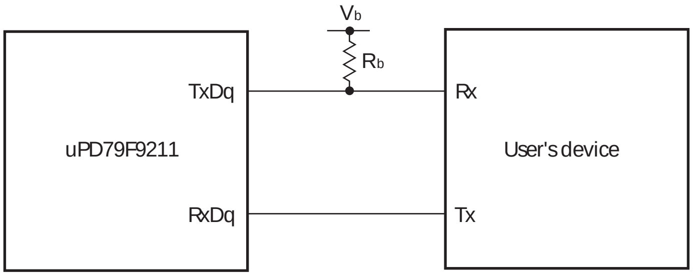
2. q: UART number (q = 0, 1) , g: PIM and POM number (g = 3, 7)

3. 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 to 3))

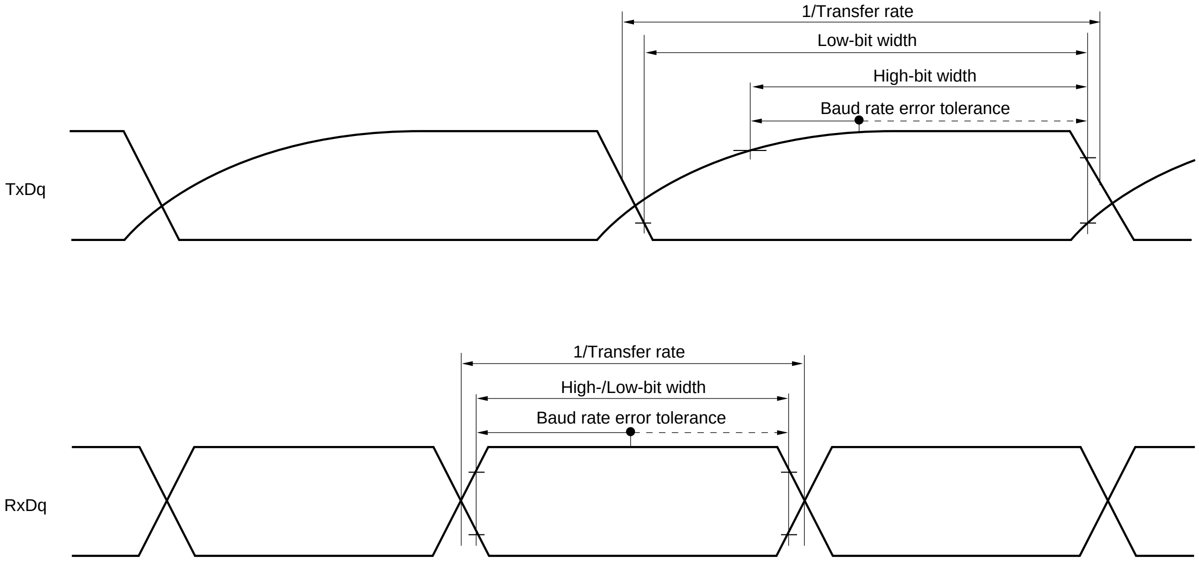
4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.
$4.0 V \leq V_{DD} \leq 5.5 V$, $2.7 V \leq V_b \leq 4.0 V$: $V_{IH} = 2.2 V$, $V_{IL} = 0.8 V$
$2.7 V \leq V_{DD} \leq 4.0 V$, $2.3 V \leq V_b \leq 2.7 V$: $V_{IH} = 2.0 V$, $V_{IL} = 0.5 V$

(2) Serial interface: Serial array unit (9/17)

UART mode connection diagram (communication at different potential)



UART mode bit width (communication at different potential) (reference)



Caution Select the TTL input buffer for RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMg registers.

Remarks 1. R_b[Ω]:Communication line (TxDq) pull-up resistance, V_b[V]: Communication line voltage

2. q: UART number (q = 0, 1) , g: PIM and POM number (g = 3, 7)

(2) Serial interface: Serial array unit (10/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

(f) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{\text{SCKp}}$... internal clock output) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ	500			ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1000			ns
$\overline{\text{SCKp}}$ high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 120			ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 275			ns
$\overline{\text{SCKp}}$ low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 20			ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 35			ns
SIp setup time (to $\overline{\text{SCKp}}$↑) Note	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ	195			ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	380			ns
SIp hold time (from $\overline{\text{SCKp}}$↑) Note	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ	30			ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output Note	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ			165	ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ			320	ns

Note When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1.

Caution Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{\text{SCKp}}$ by using the PIMg and POMg registers.

Remarks 1. p: CSI number (p = 00, 01, 10),
g: PIM and POM number (g = 3, 7)

2. n: Channel number (n = 0 to 2)

3. R_b[Ω]:Communication line ($\overline{\text{SCKp}}$, SOp) pull-up resistance,
C_b[F]: Communication line (SIp, SOp, $\overline{\text{SCKp}}$) load capacitance, V_b[V]: Communication line voltage

4. 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} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

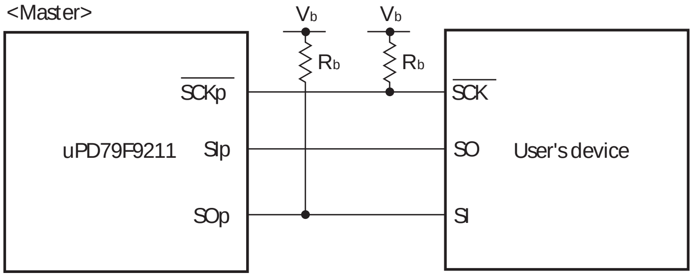
(2) Serial interface: Serial array unit (11/17)
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

(f) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{\text{SCKp}}$... internal clock output) (2/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SIp setup time (to $\overline{\text{SCKp}}$↓) [Note]	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ	70			ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	100			ns
SIp hold time (from $\overline{\text{SCKp}}$↓) [Note]	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ	30			ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{\text{SCKp}}$↑ to SOp output [Note]	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ			40	ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ			40	ns

Note When DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

CSI mode connection diagram (communication at different potential)



Caution Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{\text{SCKp}}$ by using the PIMg and POMg registers.

Remarks 1. p: CSI number (p = 00, 01, 10),
g: PIM and POM number (g = 3, 7)

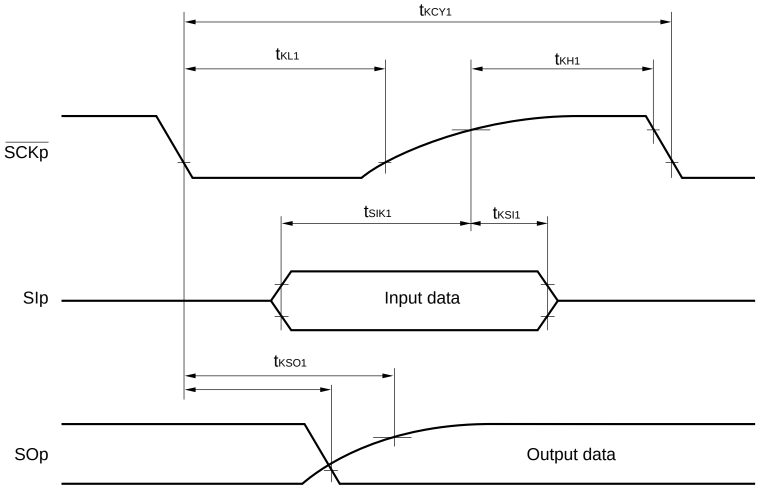
2. n: Channel number (n = 0 to 2)

3. R_b[Ω]:Communication line ($\overline{\text{SCKp}}$, SOp) pull-up resistance,
C_b[F]: Communication line (SIp, SOp, $\overline{\text{SCKp}}$) load capacitance, V_b[V]: Communication line voltage

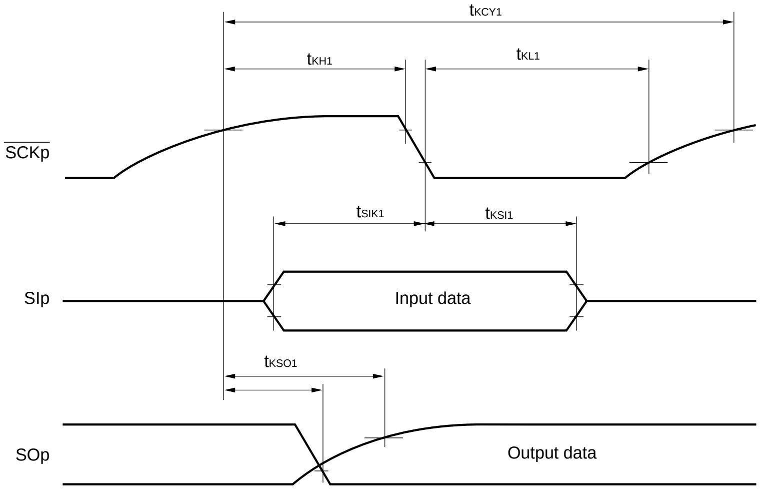
4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (12/17)

CSI mode serial transfer timing (communication at different potential)
(When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1.)



CSI mode serial transfer timing (communication at different potential)
(When DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.)



Caution Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{\text{SCKp}}$ by using the PIMg and POMg registers.

Remarks 1. p: CSI number (p = 00, 01, 10),
g: PIM and POM number (g = 3, 7)

2. n: Channel number (n = 0 to 2)

(2) Serial interface: Serial array unit (13/17)
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

(g) Communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{\text{SCKp}}$... external clock input)

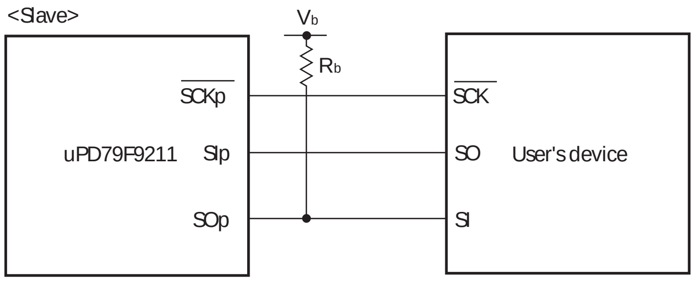
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	16.6 MHz < f_{MCK}	12/f_{MCK}			ns
			12.5 MHz < f_{MCK} ≤ 16.6 MHz	10/f_{MCK}			ns
			8.3 MHz < f_{MCK} ≤ 12.5 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 8.3 MHz	6/f_{MCK}			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	17.5 MHz < f_{MCK}	18/f_{MCK}			ns
			15 MHz < f_{MCK} ≤ 17.5 MHz	16/f_{MCK}			ns
			12.5 MHz < f_{MCK} ≤ 15 MHz	14/f_{MCK}			ns
			10 MHz < f_{MCK} ≤ 12.5 MHz	12/f_{MCK}			ns
			7.5 MHz < f_{MCK} ≤ 10 MHz	10/f_{MCK}			ns
			5 MHz < f_{MCK} ≤ 7.5 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 5 MHz	6/f_{MCK}			ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		f_{KCY2}/2 – 20			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{KCY2}/2 – 35			ns
SIp setup time (to $\overline{\text{SCKp}}$↑) Note 1	t_{SIK2}			1/f_{MCK} + 90			ns
SIp hold time (from $\overline{\text{SCKp}}$↑) Note 2	t_{KSI2}			50			ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output Note 3	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ				1/f_{MCK} + 245	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ				1/f_{MCK} + 400	ns

Notes 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes "to $\overline{\text{SCKp}}$↓" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp hold time becomes "from $\overline{\text{SCKp}}$↓" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

3. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes "from $\overline{\text{SCKp}}$↑" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

CSI mode connection diagram (communication at different potential)



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

(2) Serial interface: Serial array unit (14/17)

Caution Select the TTL input buffer for SIp and $\overline{\text{SCKp}}$ and the N-ch open drain output (V_{DD} tolerance) mode for SOp by using the PIMg and POMg registers.

Remarks 1. p: CSI number (p = 00, 01, 10),
g: PIM and POM number (g = 3, 7)

2. R_b[Ω]:Communication line (SOp) pull-up resistance,
C_b[F]: Communication line (SOp, $\overline{\text{SCKp}}$) load capacitance, V_b[V]: Communication line voltage

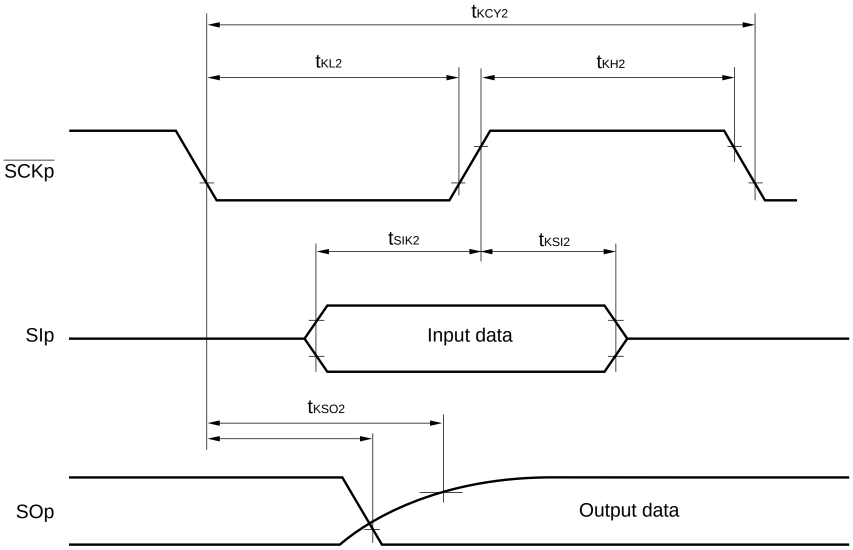
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of the SMR0n register. n: Channel number (n = 0 to 2))

4. 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.

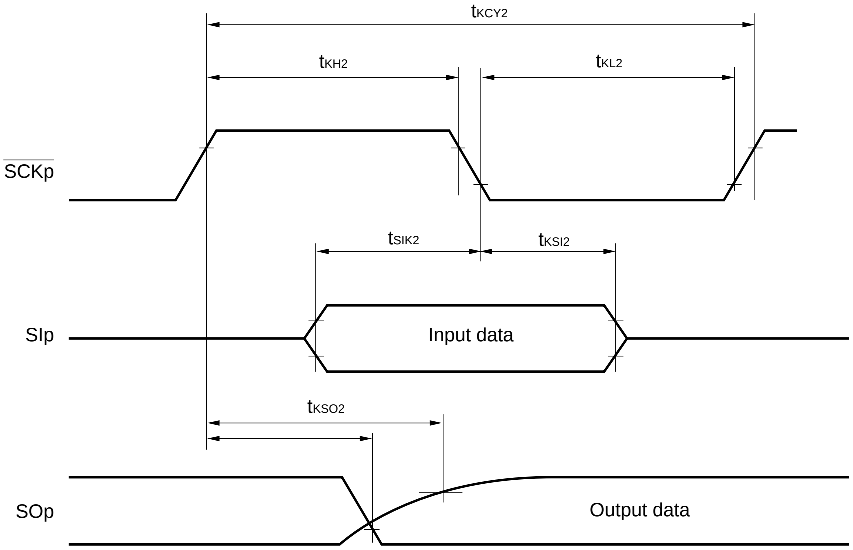
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (15/17)

CSI mode serial transfer timing (communication at different potential)
(When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1.)



CSI mode serial transfer timing (communication at different potential)
(When DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.)



Caution Select the TTL input buffer for SIp and $\overline{\text{SCKp}}$ and the N-ch open drain output (V_{DD} tolerance) mode for SOp by using the PIMg and POMg registers.

Remarks 1. p: CSI number (p = 00, 01, 10),
g: PIM and POM number (g = 3, 7)

2. n: Channel number (n = 0 to 2)

(2) Serial interface: Serial array unit (16/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

(h) Communication at different potential (2.5 V, 3 V) (simplified I^{2}C mode)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCL10 clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ		1.0	MHz
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		480	kHz
Hold time when SCL10 = “L”	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	525		ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1065		ns
Hold time when SCL10 = “H”	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	235		ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	445		ns
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	1/f_{MCK} + 135		ns
		2.7 V ≤ V_{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} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	0	140	ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	160	ns

Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDA10 and the N-ch open drain output (V_{DD} tolerance) mode for SCL10 by using the PIM3 and POM3 registers.

Remarks 1. R_b[Ω]:Communication line (SDA10, SCL10) pull-up resistance,
C_b[F]: Communication line (SDA10, SCL10) load capacitance, V_b[V]: Communication line voltage

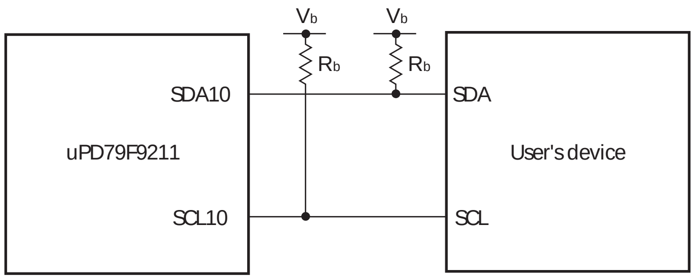
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS02 bit of the SMR02 register.)

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^{2}C mode mode.

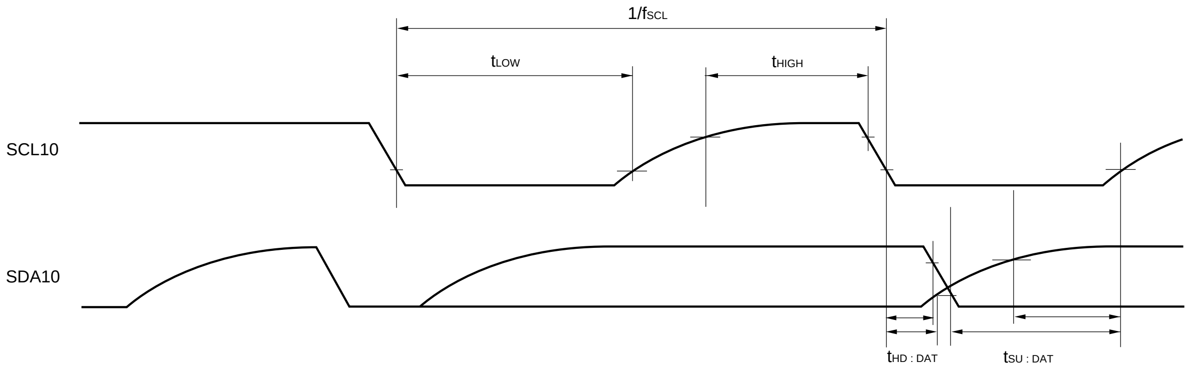
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (17/17)

Simplified I²C mode connection diagram (communication at different potential)



Simplified I²C mode serial transfer timing (communication at different potential)



Caution **Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDA10 and the N-ch open drain output (V_{DD} tolerance) mode for SCL10 by using the PIM3 and POM3 registers.**

Remark R_b[Ω]:Communication line (SDA10, SCL10) pull-up resistance, V_b[V]: Communication line voltage

(3) Serial interface: On-chip debug (UART)
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

(a) On-chip debug (UART)

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}	2.7 V ≤ V_{DD} ≤ 5.5 V			10	MHz

A/D Converter Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}					10	bit
Overall error[Notes 1, 2]	AINL	4.0 V ≤ AV_{REF} ≤ 5.5 V				±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V				±0.6	%FSR
Conversion time	t_{CONV}	4.0 V ≤ AV_{REF} ≤ 5.5 V	High speed mode 1	2.5		66.6	μs
			Normal mode	6.1		66.6	μs
		2.7 V ≤ AV_{REF} < 4.0 V	High speed mode 1	4.5		66.6	μs
			Normal mode	12.2		66.6	μs
Zero-scale error[Notes 1, 2]	EZS	4.0 V ≤ AV_{REF} ≤ 5.5 V	Normal mode			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V	Normal mode			±0.6	%FSR
Full-scale error[Notes 1, 2]	EFS	4.0 V ≤ AV_{REF} ≤ 5.5 V	Normal mode			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V	Normal mode			±0.6	%FSR
Integral non-linearity error[Note 1]	ILE	4.0 V ≤ AV_{REF} ≤ 5.5 V				±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V				±4.5	LSB
Differential non-linearity error [Note 1]	DLE	4.0 V ≤ AV_{REF} ≤ 5.5 V				±1.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V				±2.0	LSB
Analog input voltage	V_{AIN}	2.7 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.

Operational amplifier characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input offset voltage	VIO_{AMP}				T.B.D		mV
Input voltage range	VI_{AMP}	×1 gain		0.1AV_{REF}		0.45AV_{REF}	V
		Other than above		0.1AV_{REF} /gain		0.9AV_{REF} /gain	V
Maximum output voltage	VO_{AMP}			0.1AV_{REF}		0.9AV_{REF}	V
Slew rate	SR_F	Rising edge	4.0 V ≤ AV_{REF} ≤ 5.5 V		T.B.D		V/μs
			2.7 V ≤ AV_{REF} < 4.0 V		T.B.D		V/μs
	SR_R	Falling edge	4.0 V ≤ AV_{REF} ≤ 5.5 V		T.B.D		V/μs
			2.7 V ≤ AV_{REF} < 4.0 V		T.B.D		V/μs
Gain	RG			1 to 12			times
Operation stabilization wait time	t_{AMP}					3	μs

Remark Slew rate: The change with respect to the rise or fall of the output voltage

V/μs: The change in voltage per 1 μs

Operation stabilization wait time: Time required until a state is entered where the DC and AC specifications of the operational amplifier are satisfied after the operation of the operational amplifier has been enabled (OAEN = 1)

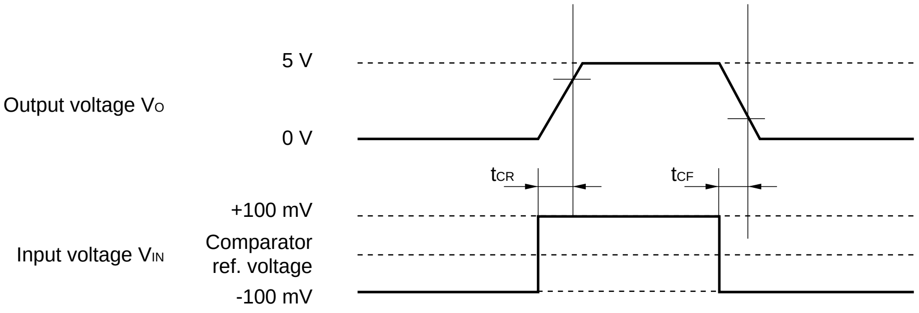
Comparator characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input offset voltage	VIO_{CMP}			T.B.D		mV
Input voltage range	VI_{CMP}		0.1AV_{REF}		0.9AV_{REF}	V
Internal reference voltage deviation	ΔV_{IREF}			T.B.D		%
Response time	t_{CR}	Input amplitude = ±100 mV, at rising edge Note 1		T.B.D		ns
	t_{CF}	Input amplitude = ±100 mV, at rising edge Note 2		T.B.D		ns
Operation stabilization wait time	t_{CMP}				1	μs
Reference voltage stabilization wait time	t_{VR}				1	μs

Notes 1. Characteristics of pulse response when CMP0P and CMP1P input or operational amplifier output changes from the comparator reference voltage –100 mV to the comparator reference voltage +100 mV.

2. Characteristics of pulse response when CMP0P and CMP1P input or operational amplifier output changes from the comparator reference voltage +100 mV to the comparator reference voltage –100 mV.



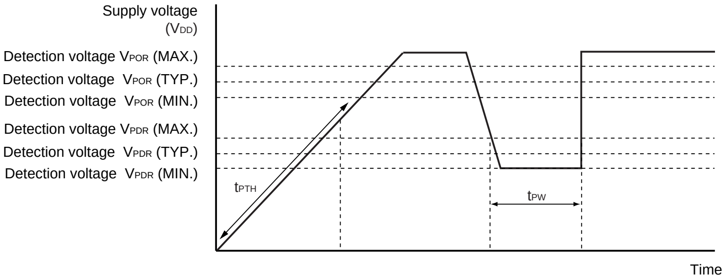
Remark Operation stabilization wait time: Time required until a state is entered where the DC and AC specifications of the comparator are satisfied after the operation of the comparator has been enabled (CnEN = 1)

stabilization wait time: Time required until the voltage level of the internal reference voltage circuit reaches 99% of the ideal value after the internal reference voltage has been enabled (CnREN = 1)

POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}		1.52	1.61	1.70	V
	V_{PDR}		1.5	1.59	1.68	V
Power supply voltage rise inclination	t_{PTH}	Change inclination of V_{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


Supply voltage
(V_{DD})
Detection voltage V_{POR} (MAX.)
Detection voltage V_{POR} (TYP.)
Detection voltage V_{POR} (MIN.)
Detection voltage V_{PDR} (MAX.)
Detection voltage V_{PDR} (TYP.)
Detection voltage V_{PDR} (MIN.)
t_{PTH}
t_{PW}
Time

Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)

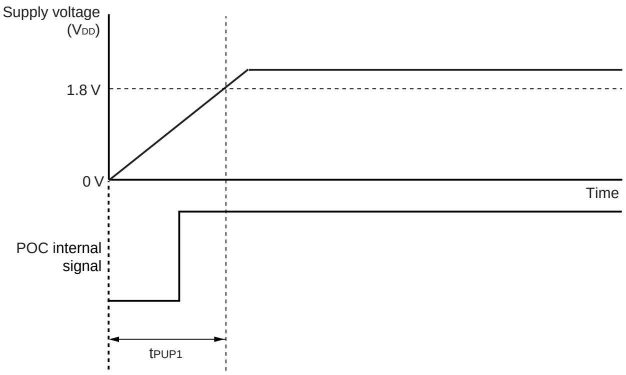
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 1.8 V (V_{DD} (MIN.)) Note (V_{DD}: 0 V → 1.8 V)	t_{PUP1}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when RESET input is not used			3.6	ms
Maximum time to rise to 1.8 V (V_{DD} (MIN.)) Note (releasing RESET input → V_{DD}: 1.8 V)	t_{PUP2}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when RESET input is used			1.88	ms

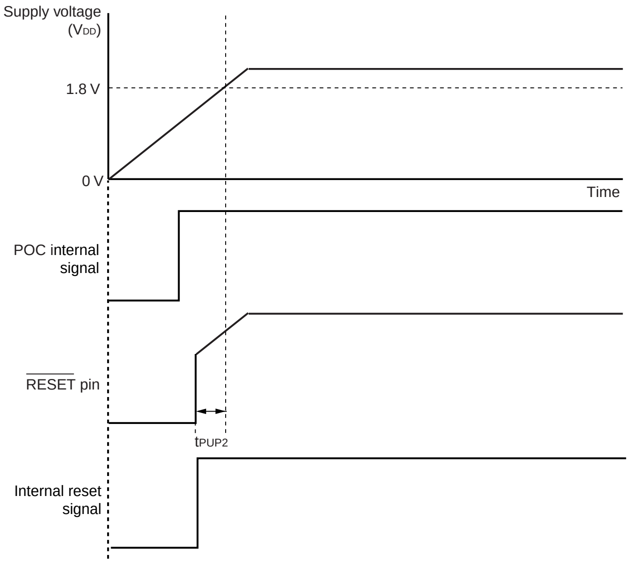
Note Make sure to raise the power supply in a shorter time than this.

Supply Voltage Rise Time Timing

- When RESET pin input is not used

- When RESET pin input is used (when external reset is released by the RESET pin, after POC has been released)





LVI Circuit Characteristics (T_A = –40 to +85°C, $V_{PDR} \le V_{DD} \le 5.5$ V, $V_{SS} = AV_{SS} = 0$ V)

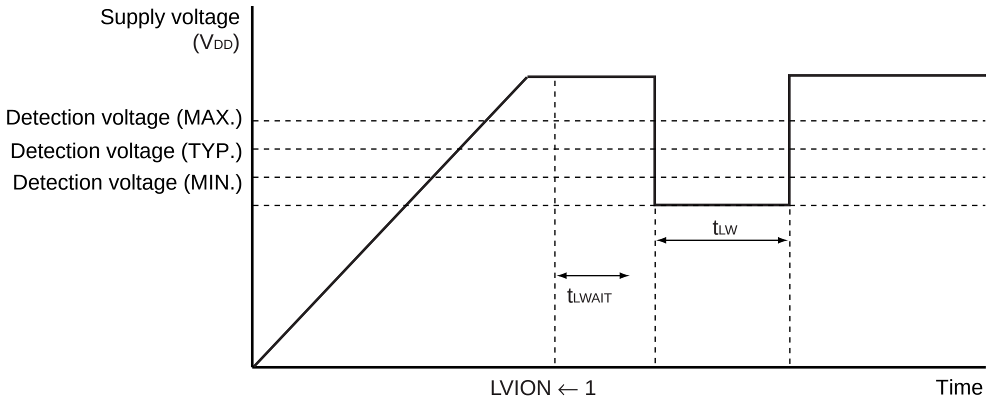
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.12	4.22	4.32	V
		V_{LVI1}		3.97	4.07	4.17	V
		V_{LVI2}		3.82	3.92	4.02	V
		V_{LVI3}		3.66	3.76	3.86	V
		V_{LVI4}		3.51	3.61	3.71	V
		V_{LVI5}		3.35	3.45	3.55	V
		V_{LVI6}		3.20	3.30	3.40	V
		V_{LVI7}		3.05	3.15	3.25	V
		V_{LVI8}		2.89	2.99	3.09	V
		V_{LVI9}		2.74	2.84	2.94	V
		V_{LVI10}		2.58	2.68	2.78	V
		V_{LVI11}		2.43	2.53	2.63	V
		V_{LVI12}		2.28	2.38	2.48	V
		V_{LVI13}		2.12	2.22	2.32	V
		V_{LVI14}		1.97	2.07	2.17	V
		V_{LVI15}		1.81	1.91	2.01	V
	External input pin[Note 1]	V_{EXLVI}	EXLVI < V_{DD}, 2.7 V ≤ V_{DD} ≤ 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 15

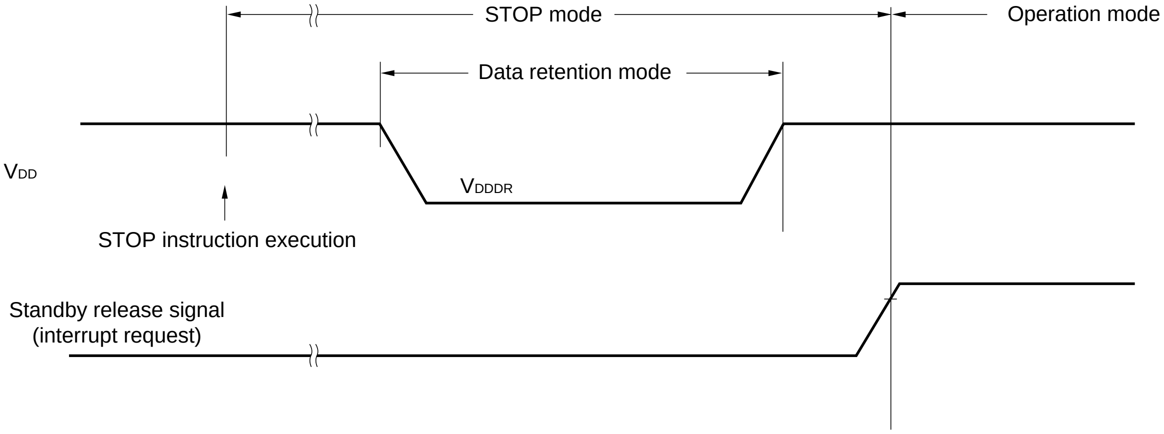
LVI Circuit Timing



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 until a POC reset is effected, but data is not retained when a POC reset is effected.

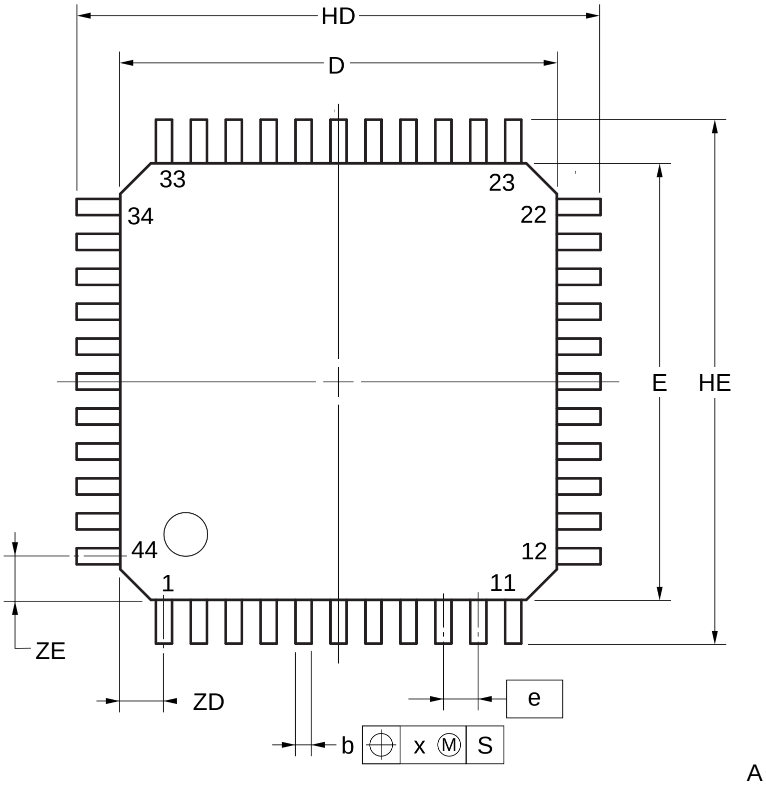


Flash Memory Programming Characteristics
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

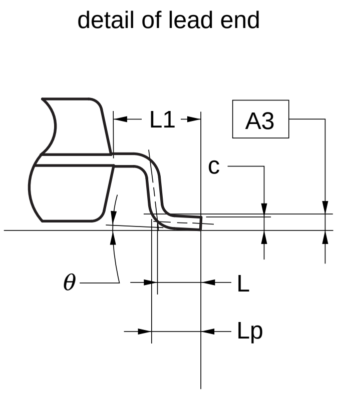
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply current	I_{DD}	Typ. = 10 MHz, Max. = 20 MHz		6	20	mA
CPU/peripheral hardware clock frequency	f_{CLK}		2		20	MHz
Number of rewrites per chip	C_{erwr}	Retention: 15 years 1 erase + 1 write after erase = 1 rewrite[Note]	100			Times

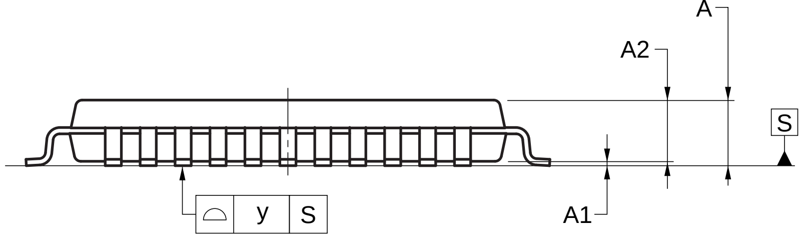
Note When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

44-PIN PLASTIC LQFP (10x10)



detail of lead end





(UNIT:mm)

ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
$\boxed{\text{A3}}$	0.25
b	$0.35^{+0.08}_{-0.04}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3°^{+5°}_{-3°}$
$\boxed{\text{e}}$	0.80
x	0.20
y	0.10
ZD	1.00
ZE	1.00

P44GB-80-GAF

NOTE
Each lead centerline is located within 0.20 mm of its true position at maximum material condition.

[MEMO]

For further information, please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

Shanghai Branch
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai, P.R.China P.C:200120
Tel:021-5888-5400
http://www.cn.necel.com/

Shenzhen Branch
Unit 01, 39/F, Excellence Times Square Building,
No. 4068 Yi Tian Road, Futian District, Shenzhen,
P.R.China P.C:518048
Tel:0755-8282-9800
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/