User's Manual

8

The sales of these products are limited for China, Hong Kong, and India.

μPD79F7025, 79F7026

User's Manual: Hardware

8-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.1.10 Nov 2013

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

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

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

(2012.4)

NOTES FOR CMOS DEVICES

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

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

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

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

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

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

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the μPD79F7025, 79F7026 and design and develop application systems and programs for these devices.

The target products are as follows.

- μPD79F7025, 79F7026

Purpose

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

Organization

The μPD79F7025, 79F7026 manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

μPD79F7025, 79F7026 User's Manual Hardware	**78K/0 Series User's Manual Instructions**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**.
- 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 RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0.
- To know details of the 78K0 microcontroller instructions:
  → Refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

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.
μPD79F7025, 79F7026 User's Manual: Hardware	This manual
78K/0 Series User's Manual Instructions	U12326E
78K0/Kx2 Flash Memory Programming (Programmer) Application Note	U17739E

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

Document Name	Document No.
PG-FP5 Flash Memory Programmer User's Manual	R20UT0008E
QB-MINI2 On-Chip Debug Emulator with Programming User's Manual	R20UT0449E
QB-Programmer Programming GUI Operation	U18527E

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

Document Name	Document No.
QB-MINI2 On-Chip Debug Emulator with Programming User's Manual	R20UT0449E

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

Documents Related to Development Tools (Software)

Document Name		Document No.
RA78K0 Ver.3.80 Assembler Package User's Manual[Note 1]	Operation	U17199E
	Language	U17198E
	Structured Assembly Language	U17197E
78K0 Assembler Package RA78K0 Ver.4.01 Operating Precautions (Notification Document)[Note 1]		ZUD-CD-07-0181-E
CC78K0 Ver.3.70 C Compiler User's Manual[Note 2]	Operation	U17201E
	Language	U17200E
78K0 C Compiler CC78K0 Ver. 4.00 Operating Precautions (Notification Document)[Note 2]		ZUD-CD-07-0103-E
ID78K0-QB Ver.2.94 Integrated Debugger User's Manual	Operation	U18330E
ID78K0-QB Ver.3.00 Integrated Debugger User's Manual	Operation	U18492E
PM plus Ver.5.20[Note 3] User's Manual		U16934E
PM+ Ver.6.30[Note 4] User's Manual		U18416E

Notes 1. This document is installed into the PC together with the tool when installing RA78K0 Ver. 4.01. For descriptions not included in "78K0 Assembler Package RA78K0 Ver. 4.01 Operating Precautions", refer to the user's manual of RA78K0 Ver. 3.80.

2. This document is installed into the PC together with the tool when installing CC78K0 Ver. 4.00. For descriptions not included in "78K0 C Compiler CC78K0 Ver. 4.00 Operating Precautions", refer to the user's manual of CC78K0 Ver. 3.70.

3. PM plus Ver. 5.20 is the integrated development environment included with RA78K0 Ver. 3.80.

4. PM+ Ver. 6.30 is the integrated development environment included with RA78K0 Ver. 4.01. Software tool (assembler, C compiler, debugger, and simulator) products of different versions can be managed.

Other Documents

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

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

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

Contents

CHAPTER 1 OUTLINE

1.1 Features

- 78K0 CPU core
- I/O ports, ROM and RAM capacities

Item / Products	I/O ports	Program Memory (Flash Memory)	Data Memory (Internal High-Speed RAM)
μPD79F7025	28 (CMOS I/O: 27, CMOS input: 1)	8 KB	512 bytes
μPD79F7026		16 KB	768 bytes

- Low power consumption
  - Internal high-speed oscillation mode: 850 μA (TYP.) (at V_{DD} = 5.0 V, f_{IH} = 4 MHz operation)
  - STOP mode: 1.0 μA (TYP.) (at V_{DD} = 5.0 V)
- Clock
  - High-speed system clock … Selected from the following three sources
    - Ceramic/crystal oscillator: 1 to 10 MHz
    - External clock: 1 to 10 MHz
    - Internal high-speed oscillator: 4 MHz ± 3 % (T_A = –40 to 85 °C)
  - Low-speed system oscillator 240 kHz ± 10 % … Watchdog timer, timer clock in intermittent operation
- Power-on-clear (POC) circuit
- Low-voltage detector (LVI) (An interrupt/reset (selectable) is generated when the detection voltage is reached))
  - Detection voltage: Selectable from three levels between 3.93 and 4.24 V
- Single-power-supply flash memory
  - Software protection function: Protected from outside party copying (no flash reading command)
- Safety function
  - Watchdog timer operated by clock independent from CPU
    … A hang-up can be detected even if the system clock stops
  - Supply voltage drop detectable by LVI
    … Appropriate processing can be executed before the supply voltage drops below the operation voltage
  - Equipped with option byte function
    … Important system operation settings set in hardware

○ Timer
- 16-bit timer/event counter … PPG output, capture input, external event counter input
- 8-bit timers H0, H1 … PWM output, operable with low-speed internal oscillation clock
- 8-bit timer/event counter 5 … External event counter input
- Watchdog timer … Operable with low-speed internal oscillation clock

Item / Products	16-bit timer/event counter	8-bit timer	Watchdog timer
µPD79F7025	1 ch	Timer H: 2 ch Timer 5: 1 ch	1 ch
µPD79F7026			

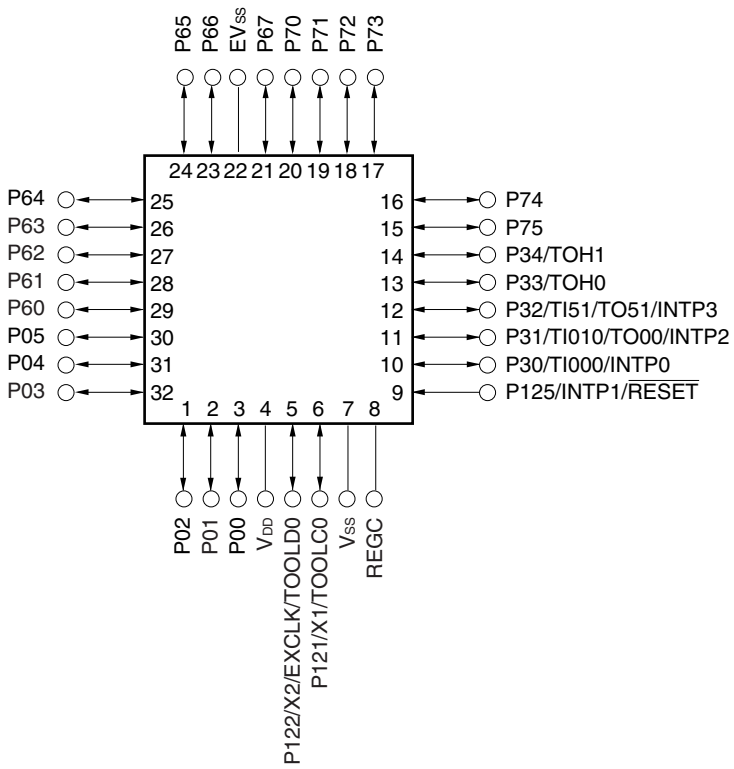
○ On-chip debug function …Available to control for the target device, and to reference memory
○ Assembler and C language supported
○ Development tools
- Support for simplified emulator (MINICUBE2)

○ Power supply voltage: V_{DD} = 4.5 to 5.5 V
○ Operating ambient temperature: T_A = –40 to +85°C

1.2 Ordering Information

Pin count	Package	ROM	RAM	Semiconductor	Part Number
32-pin	32-pin plastic LQFP (7 × 7)	8 KB	512 B	Product contains no lead in any area (Terminal finish is Ni/Pd/Au plating)	µ PD79F7025GA-GBT-AT
		16 KB	768 B		µ PD79F7026GA-GBT-AT

1.3 Pin Configuration (Top View)



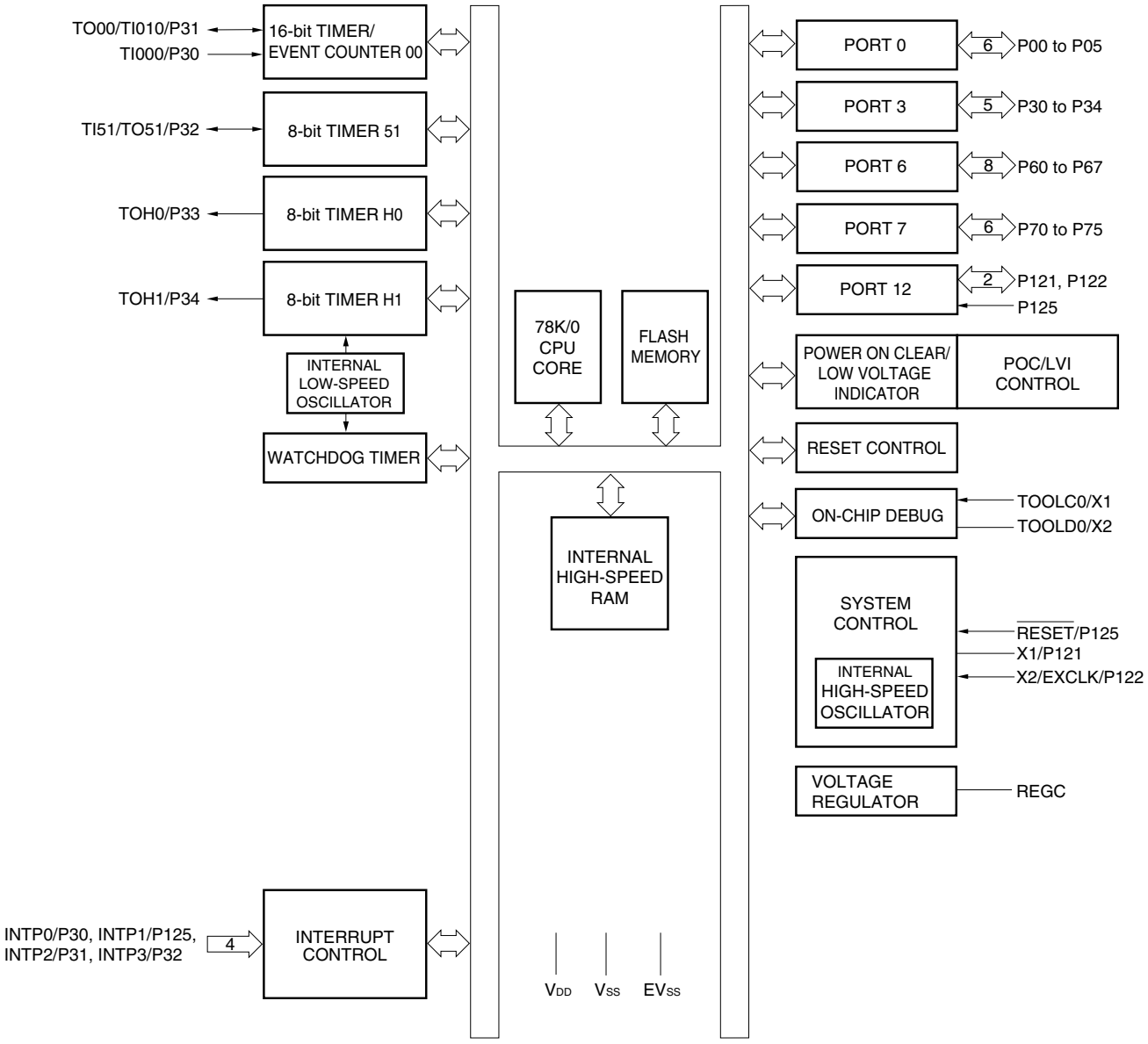
EVSS	: Ground	REGC	: Regulator Capacitance
EXCLK	: External Clock Input (Main System Clock)	$\overline{\text{RESET}}$	: Reset
		TI000, TI010, TI51	: Timer Input
INTP0 to INTP3	: External Interrupt Input	TO00, TO51, TOH0, TOH1	: Timer Output
P00 to P05	: Port 0	TOOLC0	: Clock Input for Tool
P30 to P34	: Port 3	TOOLD0	: Data Input/Output for Tool
P60 to P67	: Port 3	VDD	: Power Supply
P70 to P75	: Port 3	VSS	: Ground
P121, P122, P125	: Port 12	X1, X2	: Crystal Oscillator (Main System Clock)

Cautions 1. Be sure to connect VSS to a stabilized GND (= 0 V).

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

3. $\overline{\text{RESET}}$/P125 immediately after release of reset is set in the external reset input.

1.4 Block Diagram



Cautions 1. Be sure to connect V_{SS} to a stabilized GND (= 0 V).

2. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 µF).

3. $\overline{\text{RESET}}$/P125 immediately after release of reset is set in the external reset input.

1.5 Outline of Functions

Item			µPD79F7025	µPD79F7026
Internal memory		Flash memory	8 KB	16 KB
		High-Speed RAM	512 bytes	768 bytes
Memory space			64 KB	
Clock	Main	High-speed system (crystal/ceramic oscillation, external clock input)	1 to 10 MHz: V_{DD} = 4.5 to 5.5 V	
		Internal high-speed oscillation	4 MHz ± 3 % (T_A = –40 to 85 °C)	
	Internal low-speed oscillation		240 kHz ± 10 %	
General-purpose registers			8 bits × 32 registers (8 bits × 8 registers × 4 banks)	
Instruction set			• 8-bit operation, 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.	
I/O ports (total)			28	
	CMOS I/O		27	
	CMOS input		1	
Timer	16 bits (TM0)		1 ch (PPG output: 1, capture input: 2)	
	8 bits (TM5)		1 ch	
	8 bits (TMH0, TMH1)		2 ch (PWM output: 2)	
	Watchdog (WDT)		1 ch	
Vectored interrupt Internal sources		External	4	
		Internal	6	
Reset			• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector	
On-chip debug function			Provided	
Power supply voltage			V_{DD} = 4.5 to 5.5 V	
Operating ambient temperature			T_A = –40 to +85°C	
Package			32-pin plastic LQFP (7 × 7)	

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

This is one type of pin I/O buffer power supplies: V_{DD}. The relationship between this power supplies and the pins is shown below.

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

Power Supply	Corresponding Pins
V_{DD}	All pins

(1) Port functions

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 6-bit I/O port. Output of P00 to P05 can be set to P-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	–
P01				–
P02				–
P03				–
P04				–
P05				–
P30	I/O	Port 3. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI000/INTP0
P31				TI010/TO00/INTP2
P32				TI51/TO51/INTP3
P33				TOH0
P34				TOH1
P60	I/O	Port 6. 8-bit I/O port. Output of P60 to P67 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/pull-down resistor can be specified by a software setting.	Input port	–
P61				–
P62				–
P63				–
P64				–
P65				–
P66				–
P67				–
P70	I/O	Port 7. 6-bit I/O port. Output of P70 to 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/pull-down resistor can be specified by a software setting.	Input port	–
P71				–
P72				–
P73				–
P74				–
P75				–
P121	I/O	Port 12. P121 and P122 are 2-bit I/O port. P125 is 1-bit input-only port. For only P125, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	X1/TOOLC0
P122				X2/EXCLK/TOOLD0
P125	Input		Reset input	INTP1/$\overline{\text{RESET}}$

(2) Non-port functions

Function Name	I/O	Function	After Reset	Alternate Function
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	P30/TI000
INTP1				P125/$\overline{\text{RESET}}$
INTP2				P31/TI010/TO00
INTP3				P32/TI51/TO51
REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 µF).	–	–
$\overline{\text{RESET}}$	Input	System reset input	Reset input	P125/INTP1
TI000	Input	External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture registers (CR000, CR010) of 16-bit timer/event counter 00	Input port	P30/INTP0
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P31/TO00/INTP2
TI51	Input	External count clock input to 8-bit timer/event counter 51	Input port	P32/TO51/INTP3
TO00	Output	16-bit timer/event counter 00 output	Input port	P31/TI010/INTP2
TO51	Output	8-bit timer/event counter 51 output	Input port	P32/TI51/INTP3
TOH0	Output	8-bit timer H0 output	Input port	P33
TOH1	Output	8-bit timer H1 output	Input port	P34
X1	–	Connecting resonator for main system clock	Input port	P121/TOOLC0
X2				P122/EXCLK/TOOLD0
EXCLK	Input	External clock input for main system clock	Input port	P122/X2/TOOLD0
V_{DD}	–	Positive power supply for pins	–	–
EV_{SS}	–	Ground potential of other than P60 to P67 and P70 to P75	–	–
V_{SS}	–	Ground potential of P60 to P67 and P70 to P75	–	–
TOOLC0	Input	Clock input for flash memory programmer/on-chip debugger	Input port	P121/X1
TOOLD0	I/O	Data I/O for flash memory programmer/on-chip debugger		P122/X2/EXCLK

2.2 Description of Pin Functions

2.2.1 P00 to P05 (port 0)

P00 to P05 function as an I/O port.

Output from the P00 to P05 pins can be specified as normal CMOS output or P-ch open-drain output (V_{DD} tolerance) in 1-bit units, using port output mode register 0 (POM0).

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

(1) Port mode

P00 to P05 function as an I/O port. P00 to P05 can be set to input or output port in 1-bit units using port mode register 0 (PM0). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

2.2.2 P30 to P34 (port 3)

P30 to P34 function as an I/O port. These pins also function as pins for external interrupt request input, and timer I/O.

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

(1) Port mode

P30 to P34 function as an I/O port. P30 to P34 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 P34 function as external interrupt request input, and timer I/O.

(a) INTP0, INTP2, INTP3

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

(b) TI000

This is a pin for inputting an external count clock to 16-bit timer/event counter 00 and is also for inputting a capture trigger signal to the capture registers (CR000, CR010) of 16-bit timer/event counter 00.

(c) TI010

This is a pin for inputting a capture trigger signal to the capture register (CR000) of 16-bit timer/event counter 00.

(d) TO00

This is a timer output pin of 16-bit timer/event counter 00.

(e) TI51

This is an external count clock input pin to 8-bit timer/event counter 51.

(f) TO51

This is a timer output pin to 8-bit timer/event counter 51.

(f) TOH0, TOH1

This is a timer output pin of 8-bit timer H0, H1.

2.2.3 P60 to P67 (port 6)

P60 to P67 function as an I/O port.

Output from the P60 to P67 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 6 (POM6).

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

(1) Port mode

P60 to P67 function as an I/O port. P60 to P67 can be set to input or output port in 1-bit units using port mode register 6 (PM6). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 6 (PU6). Use of an on-chip pull-down resistor can be specified by pull-down resistor option register 6 (PD6).

2.2.4 P70 to P75 (port 7)

P70 to P75 function as an I/O port.

Output from the P70 to 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 an 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). Use of an on-chip pull-down resistor can be specified by pull-down resistor option register 7 (PD7).

2.2.5 P121, P122, P125 (port 12)

P121, P122 function as an I/O port. P125 functions as an Input port. These pins also function as connecting resonator for main system clock, external clock input for main system clock, external reset input, clock input and data I/O for flash memory programmer/on-chip debugger, and external interrupt request input.

Set bit 5 (RSTM) of the reset pin mode register (RSTMASK) to 1 when using P125/$\overline{\text{RESET}}$ as an input port, and clear RSTM to 0 when using P125/$\overline{\text{RESET}}$ as an external reset input.

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

(1) Port mode

P121, P122, and P125 function as an I/O port. P121 and P122 can be set to input or output port using port mode register 12 (PM12). Only for P125, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

(2) Control mode

P121, P122, and P125 function as connecting resonator for main system clock, external clock input for main system clock, external reset input, clock input and data I/O for flash memory programmer/on-chip debugger, and external interrupt request input.

(a) X1, X2

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

(b) EXCLK

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

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

This is an active-low system reset input pin.

(d) TOOLC0

This is a clock input pin for flash memory programmer/on-chip debugger.

(e) TOOLD0

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

(f) INTP1

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

Caution Because $\overline{\text{RESET}}$/P125 is set in the external reset input immediately after release of reset, if a reset signal is generated during low level input, the reset status continues until the input rises to the high level.

Remark For how to connect a flash memory programmer using TOOLC0/X1, TOOLD0/X2, refer to **CHAPTER 17 FLASH MEMORY**. For how to connect TOOLC0/X1, TOOLD0/X2 and an on-chip debug emulator, refer to **CHAPTER 18 ON-CHIP DEBUG FUNCTION**.

2.2.6 V_{DD}, V_{SS}, EV_{SS}

These are the power supply/ground pins.

(a) V_{DD}

V_{DD} is a positive power supply pin.

(b) V_{SS}, EV_{SS}

V_{SS} and EV_{SS} are a ground potential pins [Note].

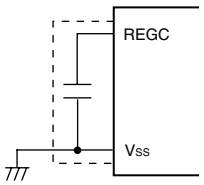
Note Be sure to connect V_{SS} to a stabilized GND (= 0 V).

2.2.7 REGC

This is a pin for connecting regulator output stabilization capacitance for internal operation and an internally connected pin.

(a) REGC

This is a pin for connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF). 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.

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

Table 2-1 shows the types of pin I/O circuits and the recommended connections of unused pins.
Refer to **Figure 2-1** for the configuration of the I/O circuit of each type.

Table 2-1. Pin I/O Circuit Types

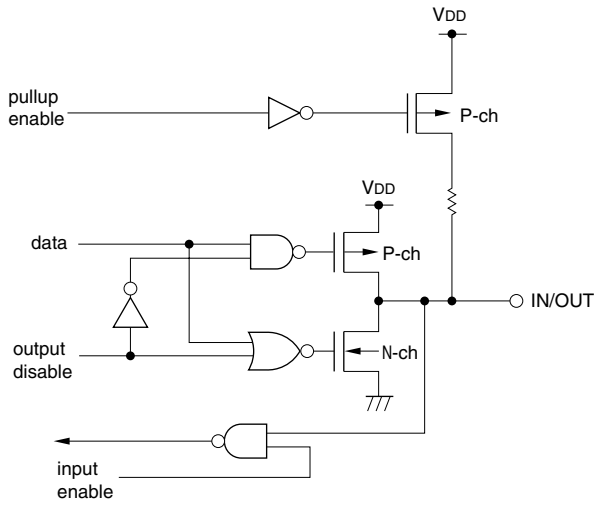
Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00 to P05	5-BF	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P30/TI000/INTP0	5-AH		
P31/TI010/TO00/INTP2			
P32/TI51/TO51/INTP3			
P33/TOH0	5-A		
P34/TOH1			
P60 to P67	5-BE		Input: Independently connect to V_{DD} or EV_{SS} via a resistor. Output: Leave open.
P70 to P75			
P121/X1/TOOLC0[Note]	37-H		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P122/X2/EXCLK/TOOLD0[Note]			
P125/$\overline{\text{RESET}}$/INTP1	42-B	Input	Connect directly to V_{DD} or via a resistor.
REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).

Note Use recommended connection above in input port mode (refer to **Figure 5-2 Format of Clock Operation Mode Select Register (OSCCTL)**) when these pins are not used.

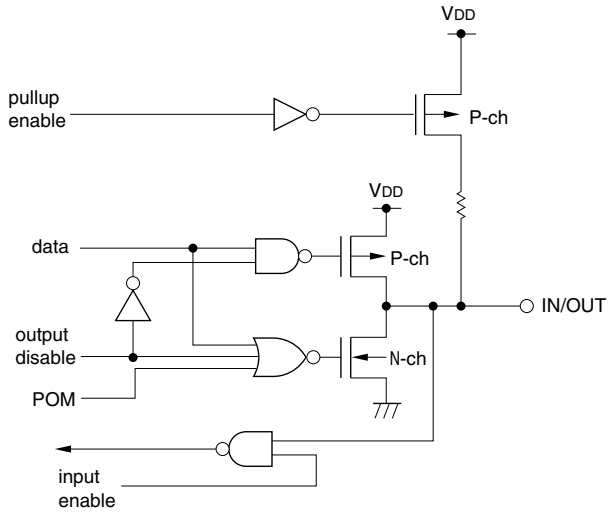
Caution Because $\overline{\text{RESET}}$/P125 is set in the external reset input immediately after release of reset, if a reset signal is generated during low level input, the reset status continues until the input rises to the high level.

Figure 2-1. Pin I/O Circuit List

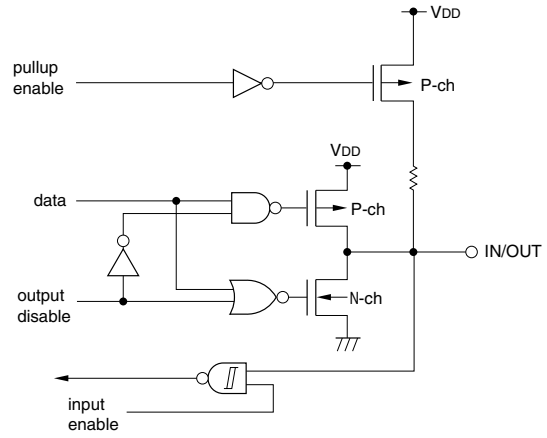
Type 5-A



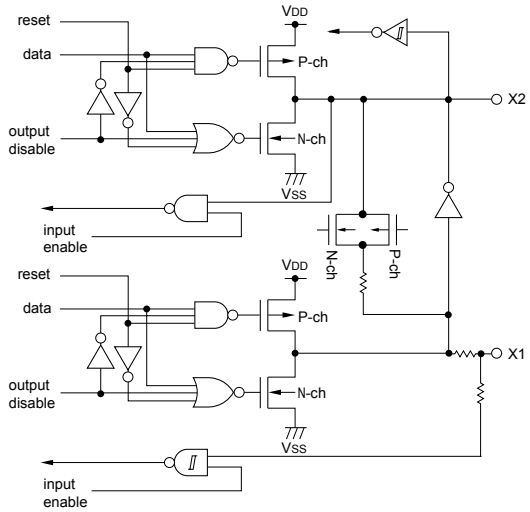
Type 5-BF



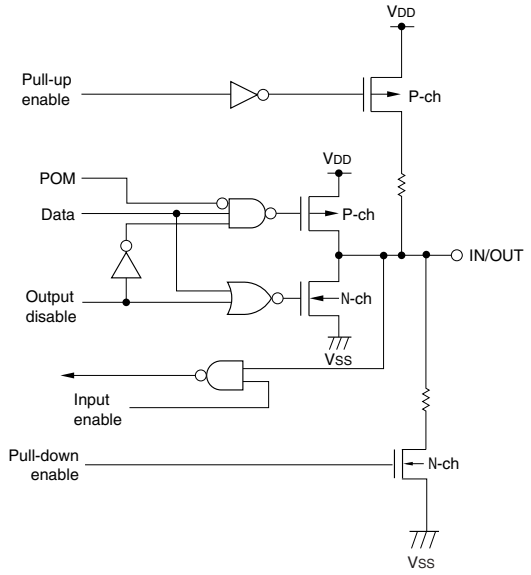
Type 5-AH



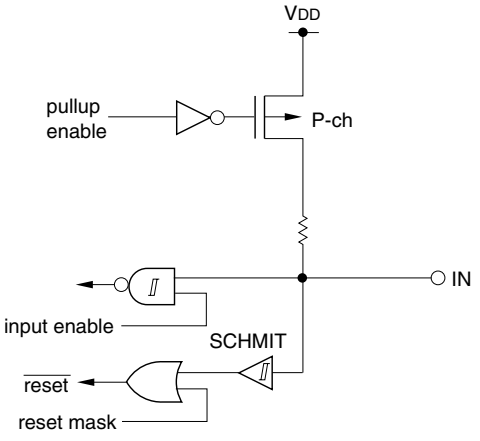
Type 37-H



Type 5-BE



Type 42-B



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

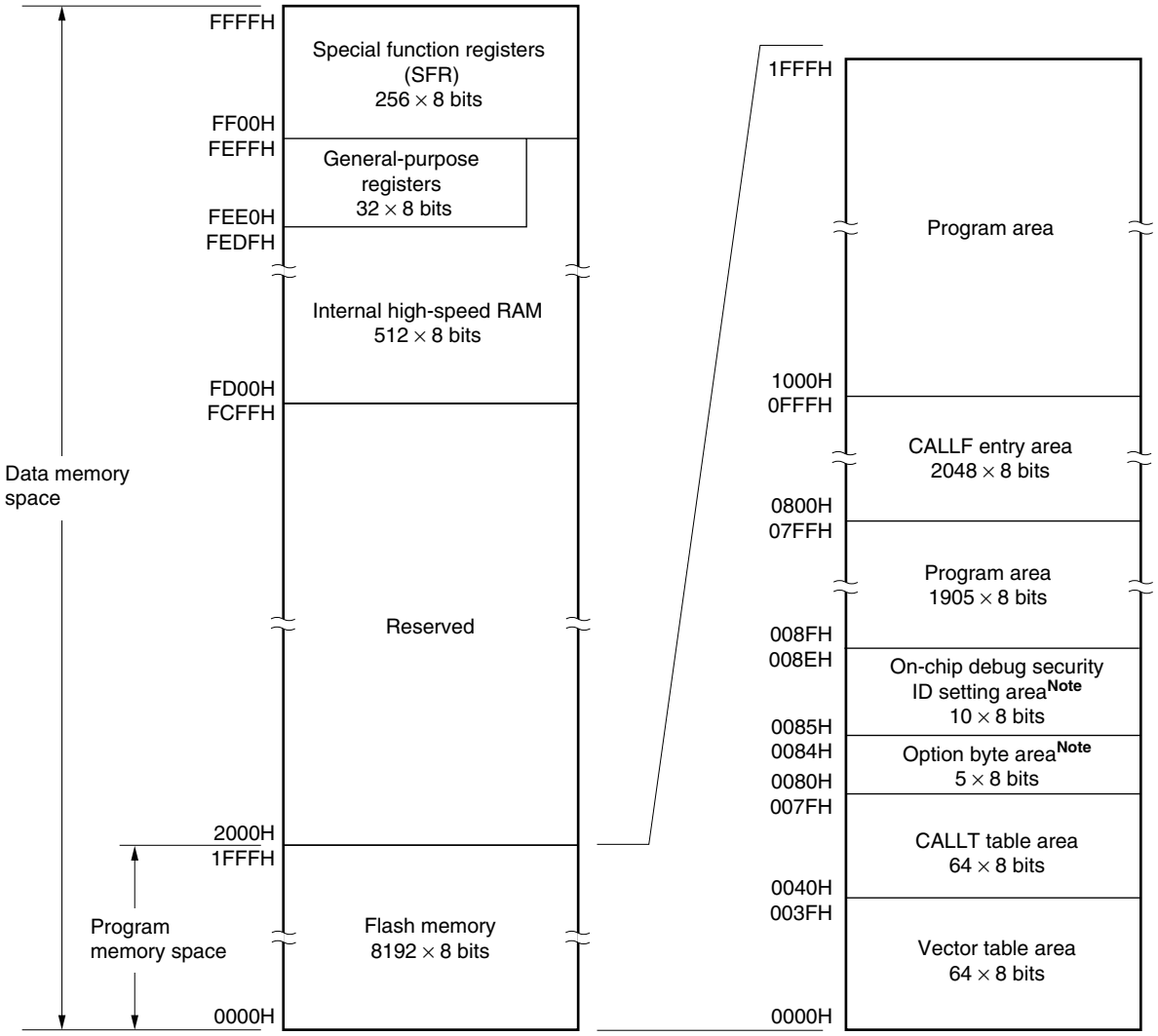
Products in the μPD79F7025, 79F7026 can access a 64 KB memory space. Figures 3-1 and 3-2 show the memory maps.

Caution Reset signal generation makes the setting of the ROM area undefined. Therefore, set the value corresponding to each product as indicated below after release of reset.

Table 3-1. Set Values of Internal Memory Size Switching Register (IMS)

Products	IMS	ROM Capacity	Internal High-Speed RAM Capacity
μPD79F7025	42H	8 KB	512 bytes
μPD79F7026	04H	16 KB	768 bytes

Figure 3-1. Memory Map (µPD79F7025)



Note Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, refer to **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

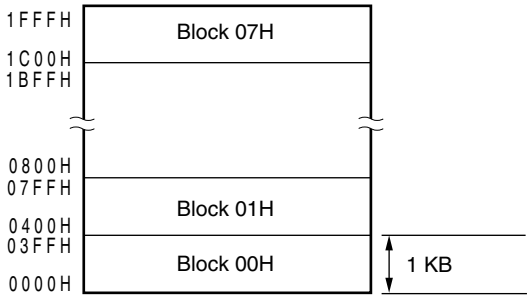
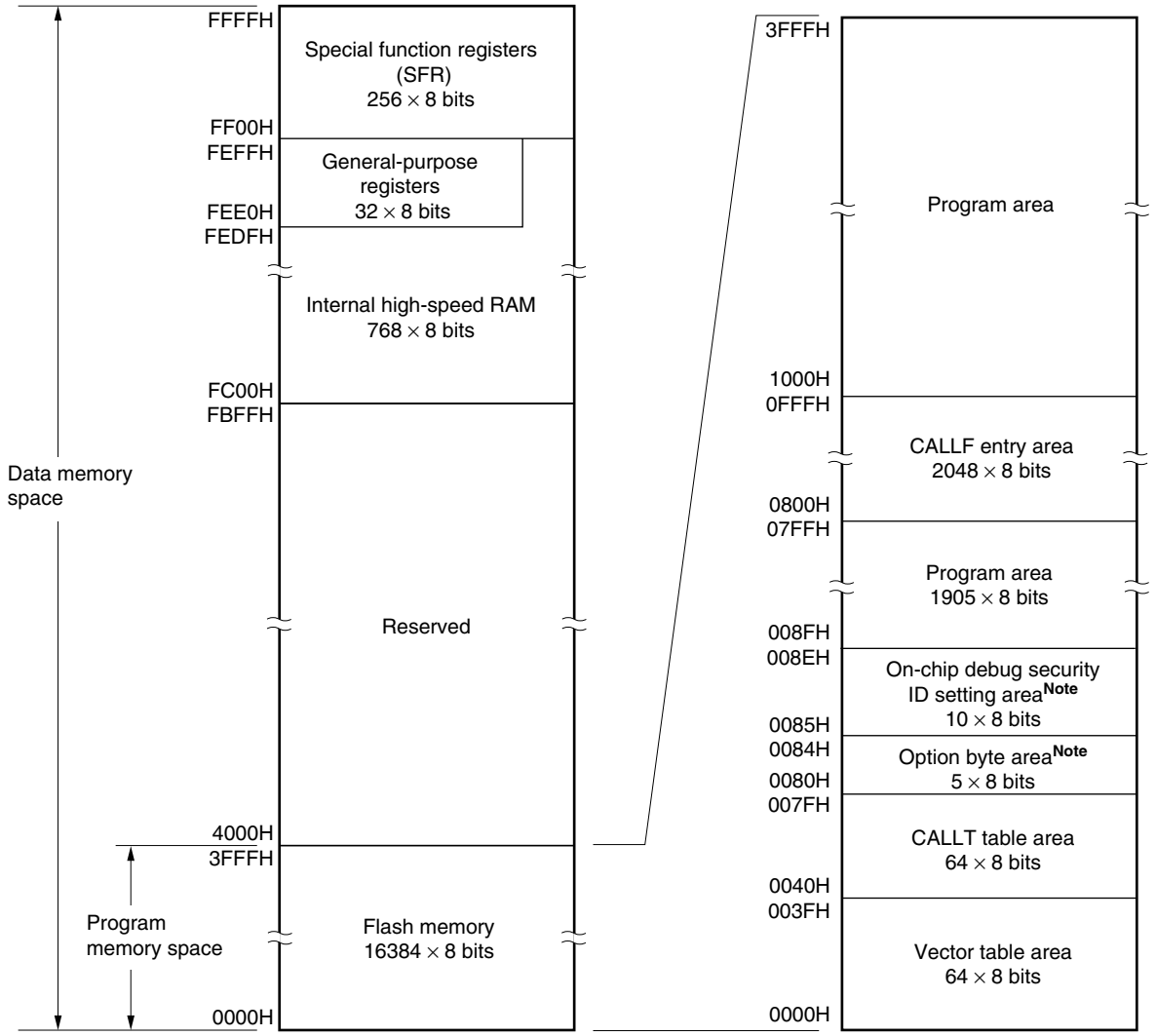
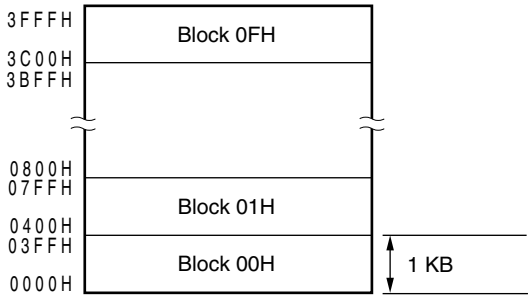


Figure 3-2. Memory Map (µPD79F7026)



Note Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, refer to **Table 3-2 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 3-2. Correspondence Between Address Values and Block Numbers in Flash Memory

Address Value	Block Number
0000H to 03FFH	00H
0400H to 07FFH	01H
0800H to 0BFFH	02H
0C00H to 0FFFH	03H
1000H to 13FFH	04H
1400H to 17FFH	05H
1800H to 1BFFH	06H
1C00H to 1FFFH	07H
2000H to 23FFH	08H
2400H to 27FFH	09H
2800H to 2BFFH	0AH
2C00H to 2FFFH	0BH
3000H to 33FFH	0CH
3400H to 37FFH	0DH
3800H to 3BFFH	0EH
3C00H to 3FFFH	0FH

Remark μPD79F7025: Block numbers 00H to 07H
μPD79F7026: 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).

μPD79F7025, 79F7026 incorporate internal ROM (flash memory), as shown below.

Table 3-3. Internal ROM Capacity

Product	Internal ROM	
	Structure	Capacity
μPD79F7025	Flash memory	8192 × 8 bits (0000H to 1FFFH)
μPD79F7026		16384 × 8 bits (0000H to 3FFFH)

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

(1) Vector table area

The 64-byte area 0000H to 003FH 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 3-4. Vector Table

Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT
0004H	INTLVI
0006H	INTP0
0008H	INTP1
000AH	INTP2
000CH	INTP3
001AH	INTTMH1
001CH	INTTMH0
0020H	INTTM000
0022H	INTTM010
002AH	INTTM51
003EH	BRK

(2) CALLT instruction table area

The 64-byte area 0040H to 007FH can store the subroutine entry address of a 1-byte call instruction (CALLT).

(3) Option byte area

A 5-byte area of 0080H to 0084H can be used as an option byte area. Set the option byte at 0080H to 0084H. For details, refer to **CHAPTER 16 OPTION BYTE**.

(4) On-chip debug security ID setting area

A 10-byte area of 0085H to 008EH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 0085H to 008EH. For details, refer to **CHAPTER 18 ON-CHIP DEBUG FUNCTION**.

(5) CALLF instruction entry area

The area 0800H to 0FFFH can perform a direct subroutine call with a 2-byte call instruction (CALLF).

3.1.2 Internal data memory space

μPD79F7025, 79F7026 incorporate the following RAMs.

(1) Internal high-speed RAM

Table 3-5. Internal High-Speed RAM Capacity

Product	Internal High-Speed RAM
μPD79F7025	512 × 8 bits (FD00H to FEFFH)
μPD79F7026	768 × 8 bits (FC00H to FEFFH)

The 32-byte area FEE0H to FEFFH is assigned to four general-purpose register banks consisting of eight 8-bit registers per bank.

This area cannot be used as a program area in which instructions are written and executed.

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

3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (refer to **Table 3-6 Special Function Register List** in **3.2.3 Special function registers (SFRs)**).

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

3.1.4 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 µPD79F7025, 79F7026 microcontrollers, 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 and 3-4 show correspondence between data memory and addressing. For details of each addressing mode, refer to **3.4 Operand Address Addressing**.

Figure 3-3. Correspondence Between Data Memory and Addressing (µPD79F7025)

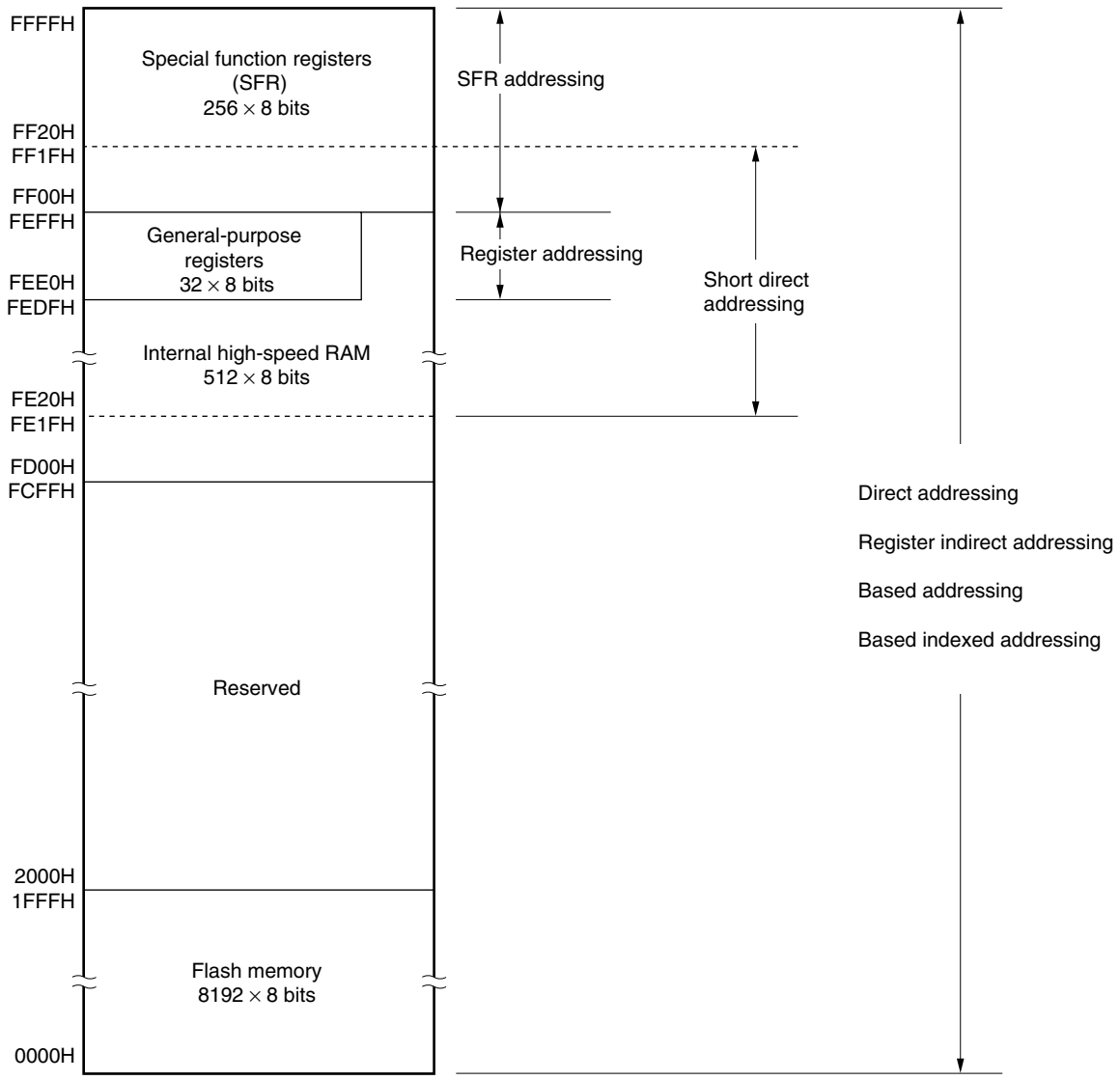
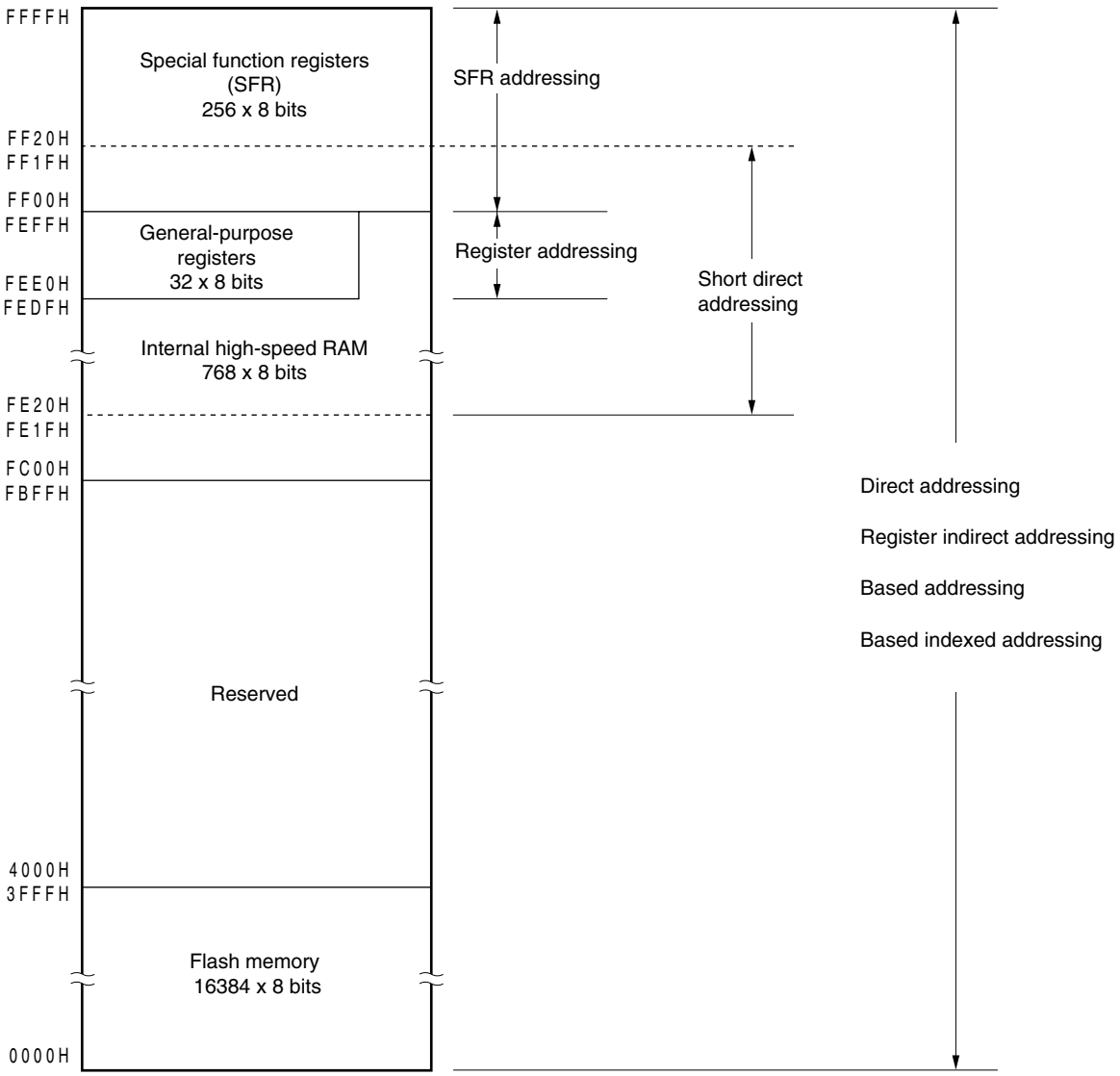


Figure 3-4. Correspondence Between Data Memory and Addressing (µPD79F7026)



3.2 Processor Registers

The μPD79F7025, 79F7026 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 16-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 3-5. Format of Program Counter

	15															0
PC	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

(2) Program status word (PSW)

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

Program status word contents are stored in the stack area upon vectored interrupt request acknowledge or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets PSW to 02H.

Figure 3-6. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

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

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgment is controlled with an in-service priority flag (ISP), 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 and 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 flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupt requests specified by a priority specification flag register (PR0L, PR0H, PR1L) (refer to **10.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

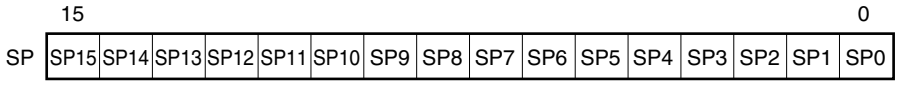
(f) Carry flag (CY)

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 high-speed RAM area can be set as the stack area.

Figure 3-7. Format of Stack Pointer



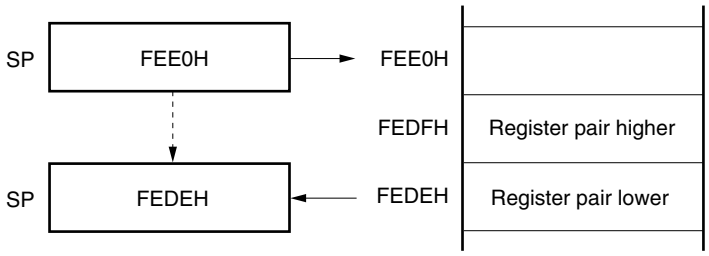
The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

Each stack operation saves/restores data as shown in Figures 3-8 and 3-9.

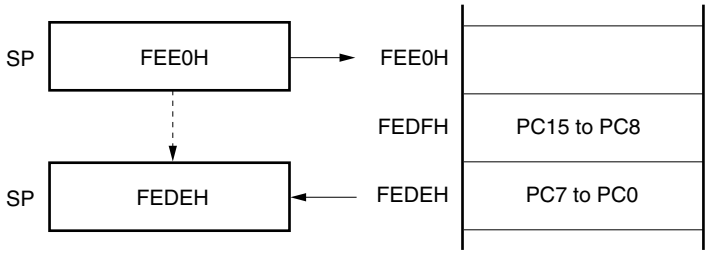
Caution Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

Figure 3-8. Data to Be Saved to Stack Memory

(a) PUSH rp instruction (when SP = FEE0H)



(b) CALL, CALLF, CALLT instructions (when SP = FEE0H)



(c) Interrupt, BRK instructions (when SP = FEE0H)

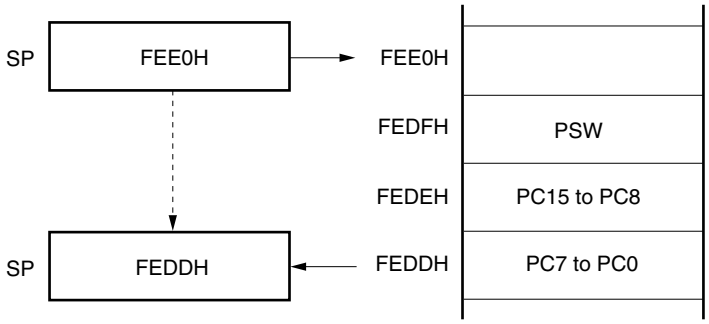
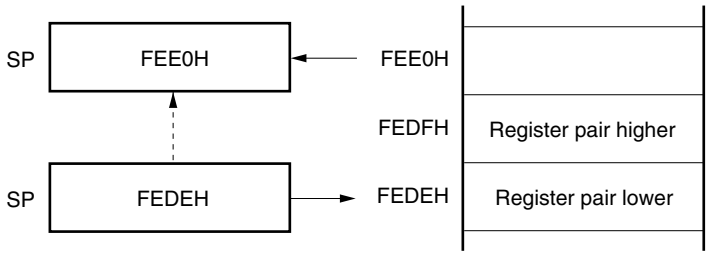
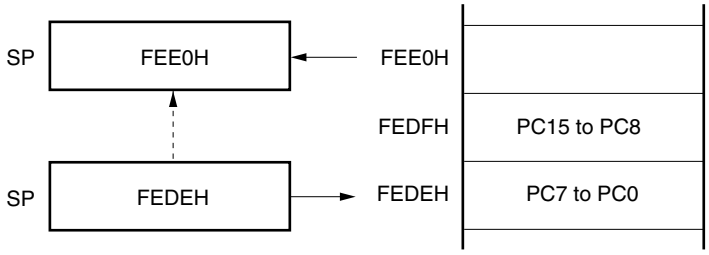


Figure 3-9. Data to Be Restored from Stack Memory

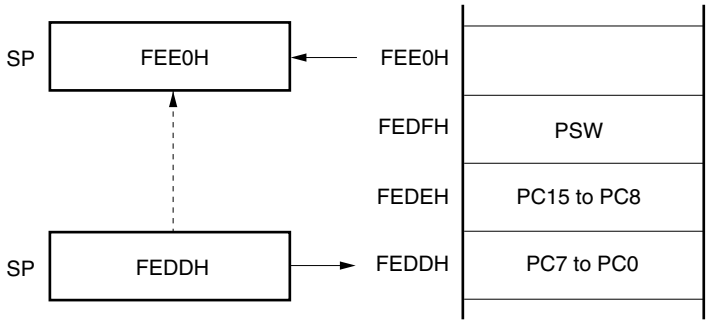
(a) POP rp instruction (when SP = FEDEH)



(b) RET instruction (when SP = FEDEH)



(c) RETI, RETB instructions (when SP = FEDDH)



3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FEE0H to FEFFH) 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).

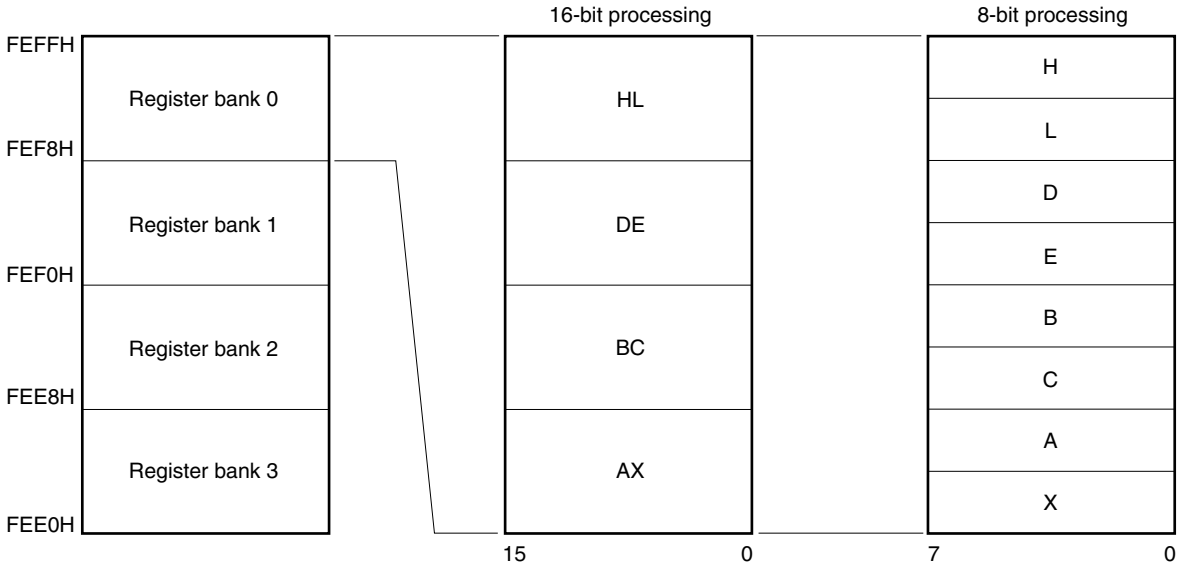
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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).

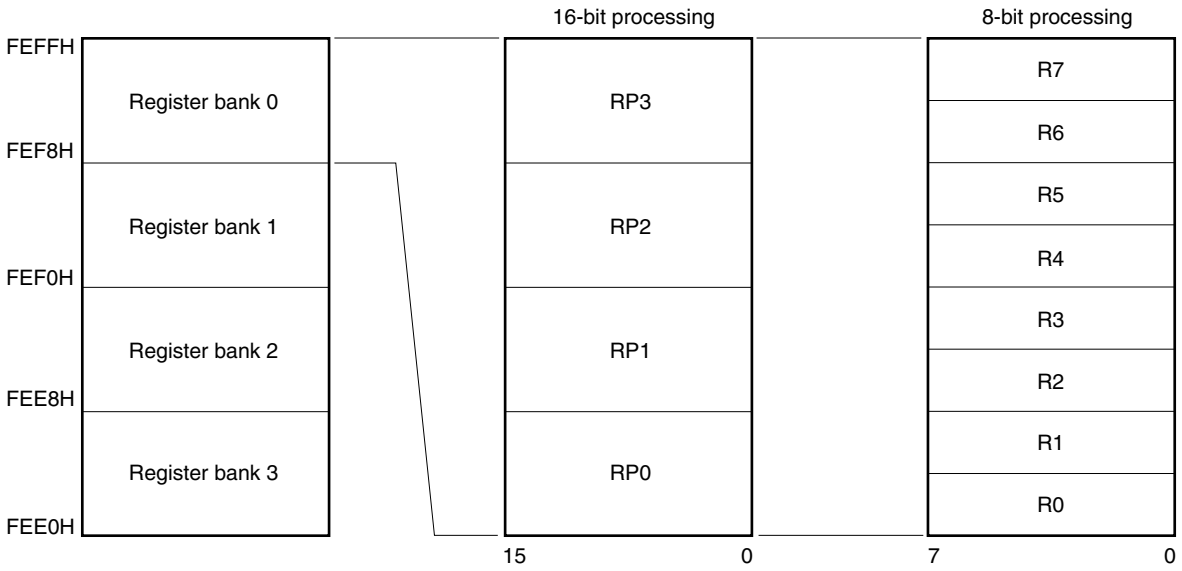
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-10. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



3.2.3 Special function registers (SFRs)

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

SFRs are allocated to the FF00H to FFFFH area.

Special function registers can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulatable bit units, 1, 8, and 16, depend on the special function register 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 3-6 gives lists of the special function registers. 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 RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0. When using the RA78K0, ID78K0-QB, and system simulator, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding special function register can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulatable bit units
  Indicates the manipulatable 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.

Table 3-6. Special Function Register List (1/4)

Address	Symbol	Bit No.								R/W	Number of Bits Manipulated Simultaneously			After Reset	Reference page
		7	6	5	4	3	2	1	0		1	8	16		
FF00H	P0	0	0	P05	P04	P03	P02	P01	P00	R/W	√	√	–	00H	
FF01H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF02H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF03H	P3	0	0	0	P34	P33	P32	P31	P30	R/W	√	√	–	00H	
FF04H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF05H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF06H	P6	P67	P66	P65	P64	P63	P62	P61	P60	R/W	√	√	–	00H	
FF07H	P7	0	0	P75	P74	P73	P72	P71	P70	R/W	√	√	–	00H	
FF08H to FF0BH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF0CH	P12	0	0	P125	0	0	P122	P121	0	R/W	√	√	–	00H	
FF0DH to FF0FH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF10H	TM00	–	–	–	–	–	–	–	–	R	–	–	√	0000H	
FF11H		–	–	–	–	–	–	–	–						
FF12H	CR000	–	–	–	–	–	–	–	–	R/W	–	–	√	0000H	
FF13H		–	–	–	–	–	–	–	–						
FF14H	CR010	–	–	–	–	–	–	–	–	R/W	–	–	√	0000H	
FF15H		–	–	–	–	–	–	–	–						
FF16H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF17H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF18H	CMP00	–	–	–	–	–	–	–	–	R/W	–	√	–	00H	
FF19H	CMP10	–	–	–	–	–	–	–	–	R/W	–	√	–	00H	
FF1AH	CMP01	–	–	–	–	–	–	–	–	R/W	–	√	–	00H	
FF1BH	CMP11	–	–	–	–	–	–	–	–	R/W	–	√	–	00H	
FF1CH to FF1EH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF1FH	TM51	–	–	–	–	–	–	–	–	R	–	√	–	00H	
FF20H	PM0	1	1	PM05	PM04	PM03	PM02	PM01	PM00	R/W	√	√	–	FFH	
FF21H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF22H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF23H	PM3	1	1	1	PM34	PM33	PM32	PM31	PM30	R/W	√	√	–	FFH	
FF24H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF25H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF26H	PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	R/W	√	√	–	FFH	
FF27H	PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	R/W	√	√	–	FFH	
FF28H to FF2BH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Remark For a bit name enclosed in angle brackets (<>), the bit name is defined as a reserved word in the RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0.

Table 3-6. Special Function Register List (2/4)

Address	Symbol	Bit No.								R/W	Number of Bits Manipulated Simultaneously			After Reset	Reference page
		7	6	5	4	3	2	1	0		1	8	16		
FF2CH	PM12	1	1	1	1	1	PM122	PM121	1	R/W	√	√	–	FFH	
FF2DH	RSTMASK	0	0	RSTM	0	0	0	0	0	R/W	√	√	–	00H	
FF2EH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF2FH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF30H	PU0	0	0	PU05	PU04	PU03	PU02	PU01	PU00	R/W	√	√	–	00H	
FF31H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF32H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF33H	PU3	0	0	0	PU34	PU33	PU32	PU31	PU30	R/W	√	√	–	00H	
FF34H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF35H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF36H	PU6	PU67	PU66	PU65	PU64	PU63	PU62	PU61	PU60	R/W	√	√	–	00H	
FF37H	PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	R/W	√	√	–	00H	
FF38H	PD6	PD67	PD66	PD65	PD64	PD63	PD62	PD61	PD60	R/W	√	√	–	00H	
FF39H	PD7	0	0	PD75	PD74	PD73	PD72	PD71	PD70	R/W	√	√	–	00H	
FF3AH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF3BH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF3CH	PU12	0	0	PU125	0	0	0	0	0	R/W	√	√	–	20H	
FF3DH to FF47H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF48H	EGP	0	0	0	0	EGP3	EGP2	EGP1	EGP0	R/W	√	√	–	00H	
FF49H	EGN	0	0	0	0	EGN3	EGN2	EGN1	EGN0	R/W	√	√	–	00H	
FF4AH to FF4FH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF50H	POM0	0	0	POM05	POM04	POM03	POM02	POM01	POM00	R/W	√	√	–	00H	
FF51H to FF55H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF56H	POM6	POM67	POM66	POM65	POM64	POM63	POM62	POM61	POM60	R/W	√	√	–	00H	
FF57H	POM7	0	0	POM75	POM74	POM73	POM72	POM71	POM70	R/W	√	√	–	00H	
FF58H to FF5FH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF60H	TMHMD0	<TMHE0>	CKS02	CKS01	CKS00	TMMD01	TMMD00	<TOLEV0>	<TOEN0>	R/W	√	√	–	00H	
FF61H to FF6FH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF70H	TMHMD1	<TMHE1>	CKS12	CKS11	CKS10	TMMD11	TMMD10	<TOLEV1>	<TOEN1>	R/W	√	√	–	00H	
FF71H	TMCYC1	0	0	0	0	0	RMC1	NRZB1	<NRZ1>	R/W	√	√	–	00H	
FF72H to FF85H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Remark For a bit name enclosed in angle brackets (<>), the bit name is defined as a reserved word in the RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0.

Table 3-6. Special Function Register List (3/4)

Address	Symbol	Bit No.								R/W	Number of Bits Manipulated Simultaneously			After Reset	Reference page
		7	6	5	4	3	2	1	0		1	8	16		
FF86H	TMC00	0	0	0	0	TMC003	TMC002	TMC001	<OVF00>	R/W	√	√	–	00H	
FF87H	PRM00	ES110	ES100	ES010	ES000	0	0	PRM001	PRM000	R/W	√	√	–	00H	
FF88H	CRC00	0	0	0	0	0	CRC002	CRC001	CRC000	R/W	√	√	–	00H	
FF89H	TOC00	0	<OSPT00>	<OSPE00>	TOC004	<LVS00>	<LVR00>	TOC001	<TOE00>	R/W	√	√	–	00H	
FF8AH to FF98H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF99H	WDTE	RUN	–	–	–	–	–	–	–	R/W	–	√	–	1AH/ 9AH[Note1]	
FF9AH to FF9EH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FF9FH	OSCCTL	<EXCLK>	<OSCSEL>	0	0	0	0	0	0	R/W	√	√	–	00H	
FFA0H	RCM	<RSTS>	0	0	0	0	0	<LSRSTOP>	<RSTOP>	R/W	√	√	–	80H[Note2]	
FFA1H	MCM	0	0	0	0	0	<XSEL>	<MCS>	<MCM0>	R/W	√	√	–	00H	
FFA2H	MOC	<MSTOP>	0	0	0	0	0	0	0	R/W	√	√	–	80H	
FFA3H	OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16	R	√	√	–	00H	
FFA4H	OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	R/W	–	√	–	05H	
FFA5H to FFABH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FFACH	RESF	0	0	0	WDTRF	0	0	0	LVIRF	R	–	√	–	00H[Note3]	
FFADH to FFB0H	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FFB1H	CR51	–	–	–	–	–	–	–	–	R/W	–	√	–	00H	
FFB2H	TCL51	0	0	0	0	0	TCL512	TCL511	TCL510	R/W	√	√	–	00H	
FFB3H	TMC51	<TCE51>	TMC516	0	0	LVS51	LVR51	TMC511	TOE51	R/W	√	√	–	00H	
FFB4H to FFBDH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FFBEH	LVIM	<LVION>	0	0	0	0	0	<LVIMD>	<LVIF>	R/W	√	√	–	00H[Note4]	
FFBFH	LVIS	0	0	0	0	0	0	LVIS1	LVIS0	R/W	√	√	–	00H[Note5]	
FFC0H to FFDFH	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Notes 1. The reset value of WDTE is determined by setting of option byte.

2. The value of this register is 00H immediately after a reset release but automatically changes to 80H after oscillation accuracy stabilization of high-speed internal oscillator has been waited.

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

4. The reset values of LVIM vary depending on the reset source and setting of option byte.

5. The reset values of LVIS vary depending on the reset source.

Remark For a bit name enclosed in angle brackets (<>), the bit name is defined as a reserved word in the RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0.

Table 3-6. Special Function Register List (4/4)

Address	Symbol		Bit No.								R/W	Number of Bits Manipulated Simultaneously			After Reset	Reference page
			7	6	5	4	3	2	1	0		1	8	16		
FFE0H	IF0	IF0L	0	0	0	<PIF3>	<PIF2>	<PIF1>	<PIF0>	<LVIIF>	R/W	√	√	√	00H	
FFE1H		IF0H	<TMIF010>	<TMIF000>	0	<TMIFH0>	<TMIFH1>	0	0	0	R/W	√	√		00H	
FFE2H	IF1	IF1L	0	0	0	0	<TMIF51>	0	0	0	R/W	√	√	√	00H	
FFE3H		–	0	0	0	0	0	0	0	0	–	–	–		00H	
FFE4H	MK0	MK0L	1	1	1	<PMK3>	<PMK2>	<PMK1>	<PMK0>	<LVIMK>	R/W	√	√	√	FFH	
FFE5H		MK0H	<TMMK010>	<TMMK000>	1	<TMMKH0>	<TMMKH1>	1	1	1	R/W	√	√		FFH	
FFE6H	MK1	MK1L	1	1	1	1	<TMMK51>	1	1	1	R/W	√	√	√	FFH	
FFE7H		–	1	1	1	1	1	1	1	1	–	–	–		FFH	
FFE8H	PR0	PR0L	1	1	1	<PPR3>	<PPR2>	<PPR1>	<PPR0>	<LVIPR>	R/W	√	√	√	FFH	
FFE9H		PR0H	<TMPR010>	<TMPR000>	1	<TMPRH0>	<TMPRH1>	1	1	1	R/W	√	√		FFH	
FFEAH	PR1	PR1L	1	1	1	1	<TMPR51>	1	1	1	R/W	√	√	√	FFH	
FFEBH		–	1	1	1	1	1	1	1	1	–	–	–		FFH	
FFECH to FFEFH	–		–	–	–	–	–	–	–	–	–	–	–	–	–	–
FFF0H	IMS		RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0	R/W	–	√	–	CFH Note	
FFF1H to FFFAH	–		–	–	–	–	–	–	–	–	–	–	–	–	–	–
FFFBH	PCC		0	0	0	0	0	PCC2	PCC1	PCC0	R/W	√	√	–	01H	

Note Reset signal generation makes the setting of the ROM area undefined. Therefore, set the value corresponding to each product as indicated in Table 3-1 after release of reset.

Remark For a bit name enclosed in angle brackets (<>), the bit name is defined as a reserved word in the RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0.

3.3 Instruction Address Addressing

An instruction address is determined by contents of the program counter (PC) and is normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to PC and branched by the following addressing (for details of instructions, refer to the **78K/0 Series Instructions User's Manual (U12326E)**).

3.3.1 Relative addressing

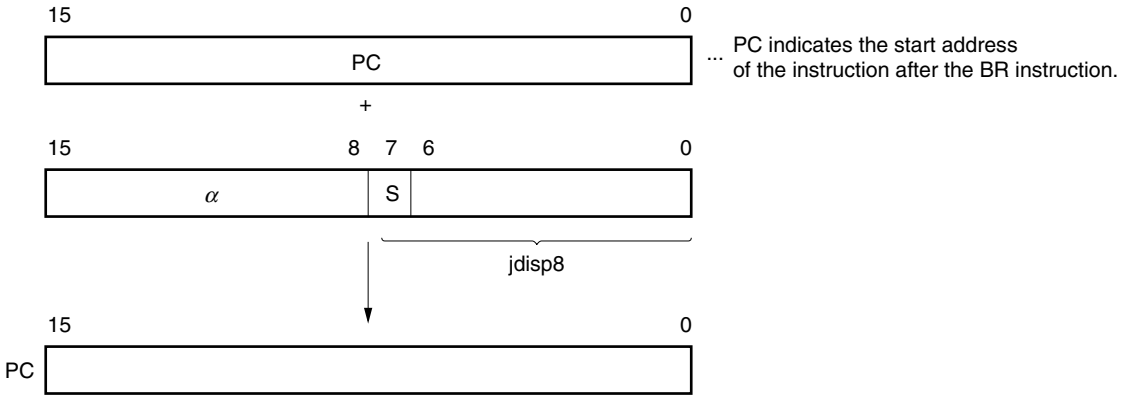
[Function]

The value obtained by adding 8-bit immediate data (displacement value: jdisp8) of an instruction code to the start address of the following instruction is transferred to the program counter (PC) and branched. The displacement value is treated as signed two's complement data (−128 to +127) and bit 7 becomes a sign bit.

In other words, relative addressing consists of relative branching from the start address of the following instruction to the −128 to +127 range.

This function is carried out when the BR $addr16 instruction or a conditional branch instruction is executed.

[Illustration]



When S = 0, all bits of α are 0.
When S = 1, all bits of α are 1.

3.3.2 Immediate addressing

[Function]

Immediate data in the instruction word is transferred to the program counter (PC) and branched.

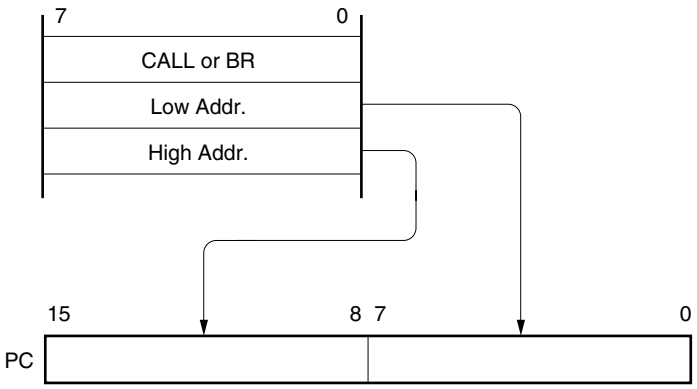
This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed.

CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space.

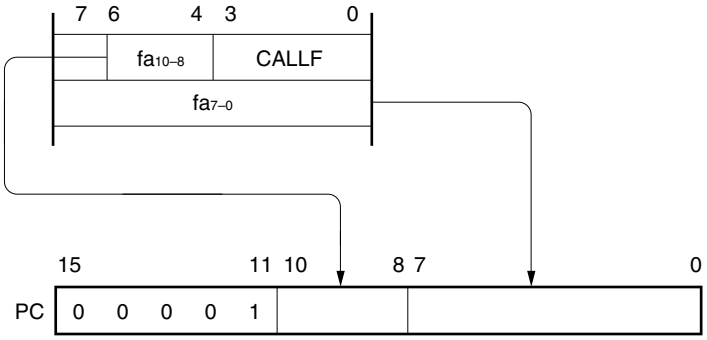
The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

In the case of CALL !addr16 and BR !addr16 instructions



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

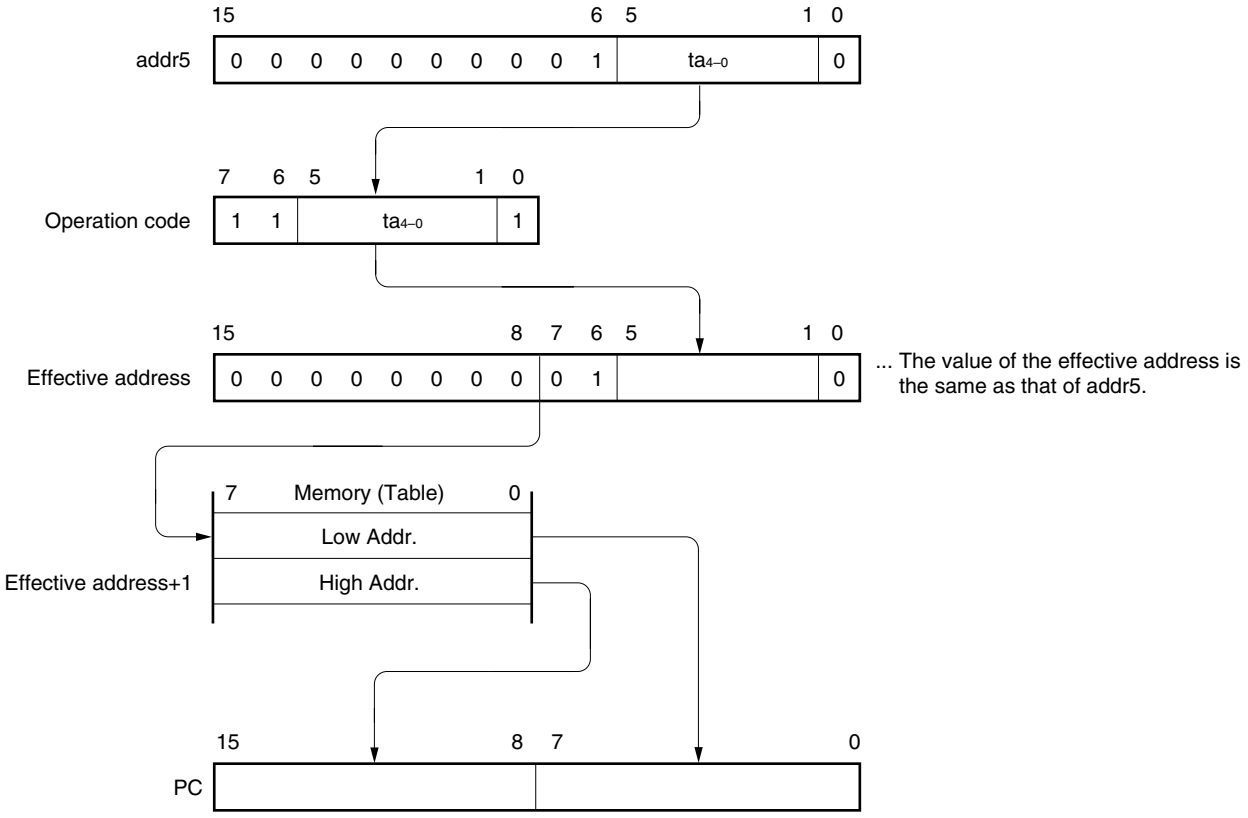
[Function]

Table contents (branch destination address) of the particular location to be addressed by bits 1 to 5 of the immediate data of an operation code are transferred to the program counter (PC) and branched.

This function is carried out when the CALLT [addr5] instruction is executed.

This instruction references the address that is indicated by addr5 and is stored in the memory table from 0040H to 007FH, and allows branching to the entire memory space.

[Illustration]



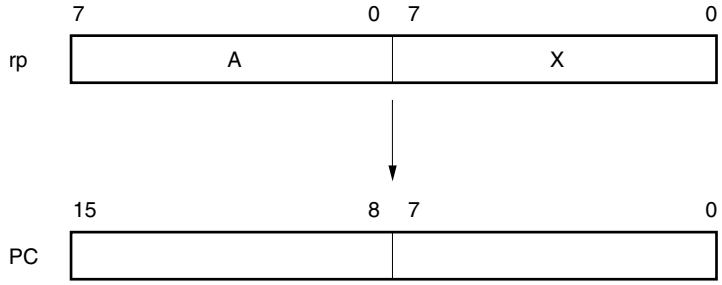
3.3.4 Register addressing

[Function]

Register pair (AX) contents to be specified with an instruction word are transferred to the program counter (PC) and branched.

This function is carried out when the BR AX instruction is executed.

[Illustration]



3.4 Operand Address Addressing

The following methods are available to specify the register and memory (addressing) to undergo manipulation during instruction execution.

3.4.1 Implied addressing

[Function]

The register that functions as an accumulator (A and AX) among the general-purpose registers is automatically (implicitly) addressed.

Of the μPD79F7025, 79F7026 instruction words, the following instructions employ implied addressing.

Instruction	Register to Be Specified by Implied Addressing
MULU	A register for multiplicand and AX register for product storage
DIVUW	AX register for dividend and quotient storage
ADJBA/ADJBS	A register for storage of numeric values that become decimal correction targets
ROR4/ROL4	A register for storage of digit data that undergoes digit rotation

[Operand format]

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

[Description example]

In the case of MULU X

With an 8-bit $\times$ 8-bit multiply instruction, the product of the A register and X register is stored in AX. In this example, the A and AX registers are specified by implied addressing.

3.4.2 Register addressing

[Function]

The general-purpose register to be specified is accessed as an operand with the register bank select flags (RBS0 to RBS1) and the register specify codes of an operation code.

Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

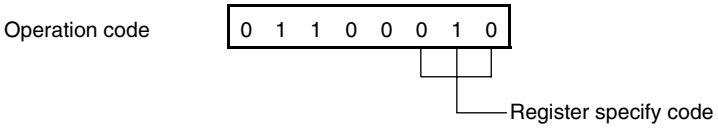
[Operand format]

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

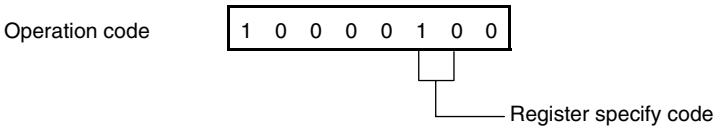
'r' and 'rp' can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

The memory to be manipulated is directly addressed with immediate data in an instruction word becoming an operand address.

This addressing can be carried out for all of the memory spaces.

[Operand format]

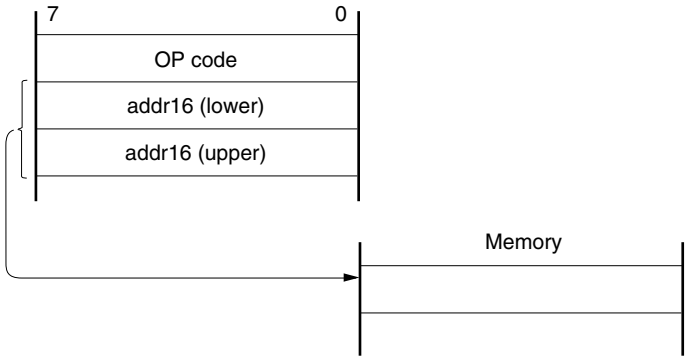
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

MOV A, !0FE00H; when setting !addr16 to FE00H

Operation code	1 0 0 0 1 1 1 0	OP code
	0 0 0 0 0 0 0 0	00H
	1 1 1 1 1 1 1 0	FEH

[Illustration]



3.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word.

This addressing is applied to the 256-byte space FE20H to FF1FH. Internal high-speed RAM and special function registers (SFRs) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the overall SFR area. Ports that are frequently accessed in a program and compare and capture registers of the timer/event counter are mapped in this area, allowing SFRs to be manipulated with a small number of bytes and clocks.

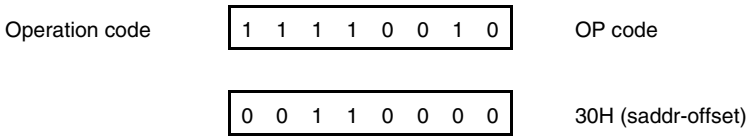
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. Refer to the **[Illustration]** shown below.

[Operand format]

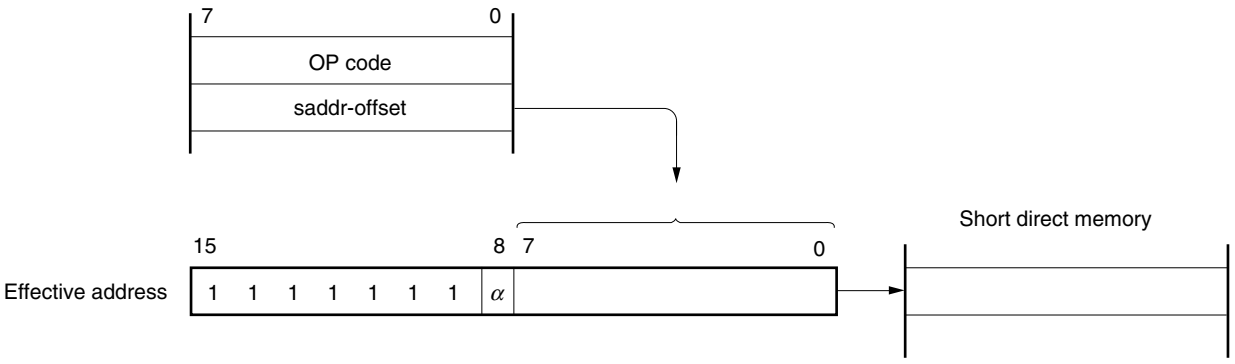
Identifier	Description
saddr	Immediate data that indicate label or FE20H to FF1FH
saddrp	Immediate data that indicate label or FE20H to FF1FH (even address only)

[Description example]

```
LB1 EQU 0FE30H   ; Defines FE30H by LB1.
      :
MOV LB1, A       ; When LB1 indicates FE30H of the saddr area and the value of register A is transferred to that
                   address
```



[Illustration]



When 8-bit immediate data is 20H to FFH, $\alpha = 0$

When 8-bit immediate data is 00H to 1FH, $\alpha = 1$

3.4.5 Special function register (SFR) addressing

[Function]

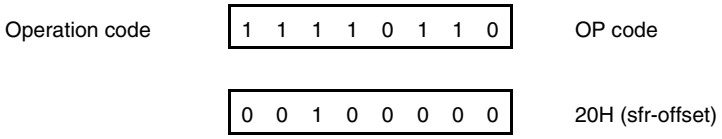
A memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word. This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

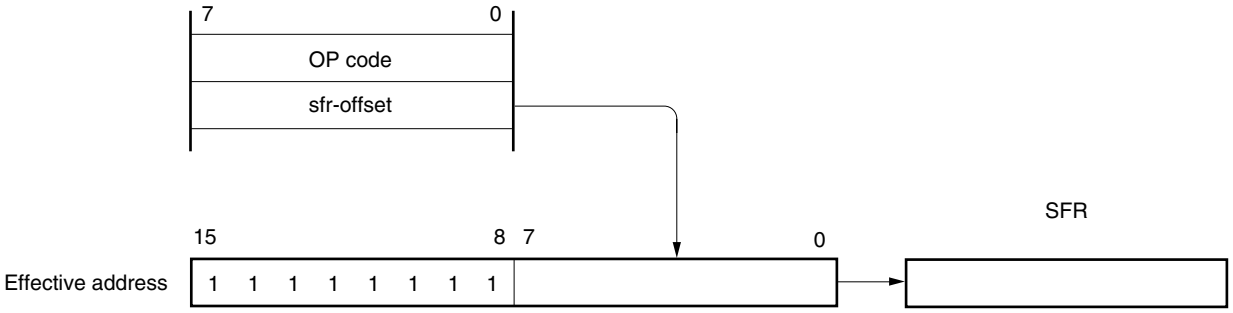
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

MOV PM0, A; when selecting PM0 (FF20H) as sfr



[Illustration]



3.4.6 Register indirect addressing

[Function]

Register pair contents specified by a register pair specify code in an instruction word and by a register bank select flag (RBS0 and RBS1) serve as an operand address for addressing the memory.

This addressing can be carried out for all of the memory spaces.

[Operand format]

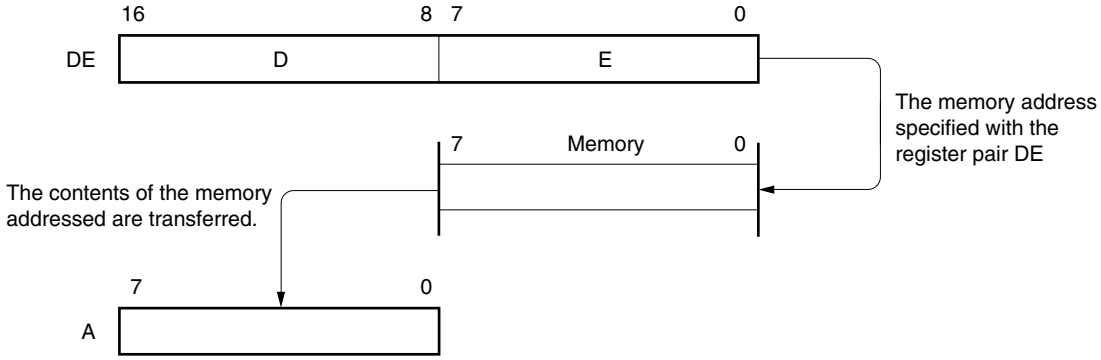
Identifier	Description
–	[DE], [HL]

[Description example]

MOV A, [DE]; when selecting [DE] as register pair

Operation code	1	0	0	0	0	1	0	1

[Illustration]



3.4.7 Based addressing

[Function]

8-bit immediate data is added as offset data to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored.

This addressing can be carried out for all of the memory spaces.

[Operand format]

Identifier	Description
–	[HL + byte]

[Description example]

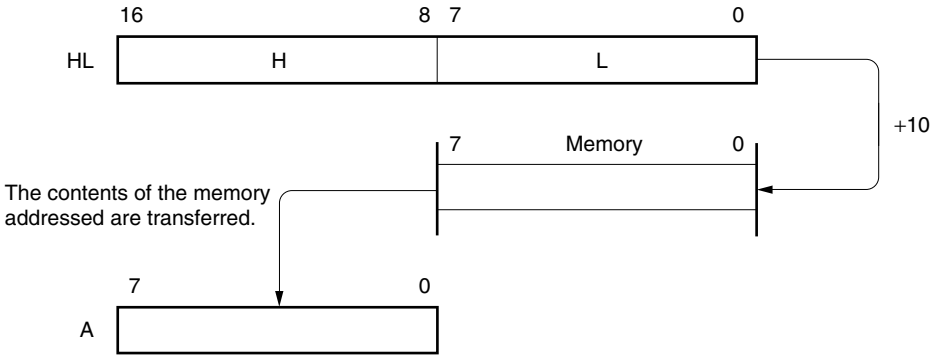
MOV A, [HL + 10H]; when setting byte to 10H

Operation code

1	0	1	0	1	1	1	0

0	0	0	1	0	0	0	0

[Illustration]



3.4.8 Based indexed addressing

[Function]

The B or C register contents specified in an instruction word are added to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the B or C register contents as a positive number to 16 bits. A carry from the 16th bit is ignored.

This addressing can be carried out for all of the memory spaces.

[Operand format]

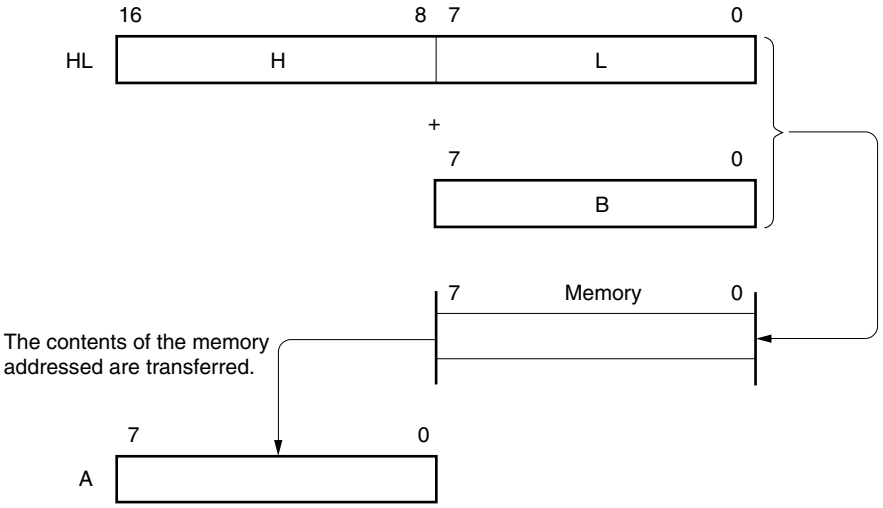
Identifier	Description
–	[HL + B], [HL + C]

[Description example]

MOV A, [HL +B]; when selecting B register

Operation code	1	0	1	0	1	0	1	1

[Illustration]



3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) contents.

This addressing method is automatically employed when the PUSH, POP, subroutine call and return instructions are executed or the register is saved/reset upon generation of an interrupt request.

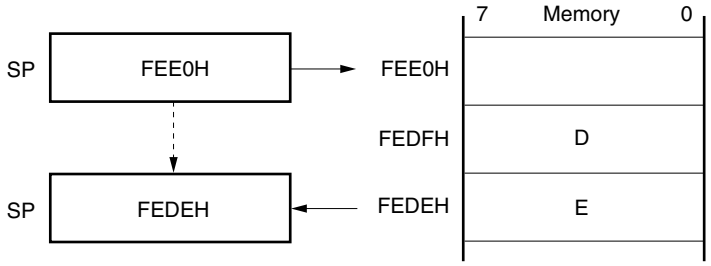
With stack addressing, only the internal high-speed RAM area can be accessed.

[Description example]

PUSH DE; when saving DE register

Operation code	1	0	1	1	0	1	0	1

[Illustration]



CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

This is one type of pin I/O buffer power supplies: V_{DD}. The relationship between this power supplies and the pins is shown below.

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

Power Supply	Corresponding Pins
V_{DD}	All pins

µPD79F7025, 79F7026 microcontrollers are provided with digital I/O ports, 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, refer to **CHAPTER 2 PIN FUNCTIONS**.

Table 4-2. Port Functions

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 6-bit I/O port. Output of P00 to P05 can be set to P-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	–
P01				–
P02				–
P03				–
P04				–
P05				–
P30	I/O	Port 3. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI000/INTP0
P31				TI010/TO00/INTP2
P32				TI51/TO51/INTP3
P33				TOH0
P34				TOH1
P60	I/O	Port 6. 8-bit I/O port. Output of P60 to P67 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	–
P61				–
P62				–
P63				–
P64				–
P65				–
P66				–
P67				–
P70	I/O	Port 7. 6-bit I/O port. Output of P70 to 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	–
P71				–
P72				–
P73				–
P74				–
P75				–
P121	I/O	Port 12. P121, P122 is 2-bit I/O port. P125 is 1-bit input-only port. For only P125, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	X1/TOOLC0
P122				X2/EXCLK/TOOLD0
P125	Input		Reset input	INTP1/$\overline{\text{RESET}}$

4.2 Port Configuration

Ports include the following hardware.

Table 4-3. Port Configuration

Item	Configuration
Control registers	Port mode registers (PMxx): PM0, PM3, PM6, PM7, PM12 Port registers (Pxx): P0, P3, P6, P7, P12 Pull-up resistor option registers (PUxx): PU0, PU3, PU6, PU7, PU12 Pull-down resistor option registers (PDxx): PD6, PD7 Reset pin mode register (RSTMASK)
Port	Total: 28 (CMOS I/O: 27, CMOS input: 1)
Pull-up resistor	Total: 26
Pull-down resistor	Total: 14

4.2.1 Port 0

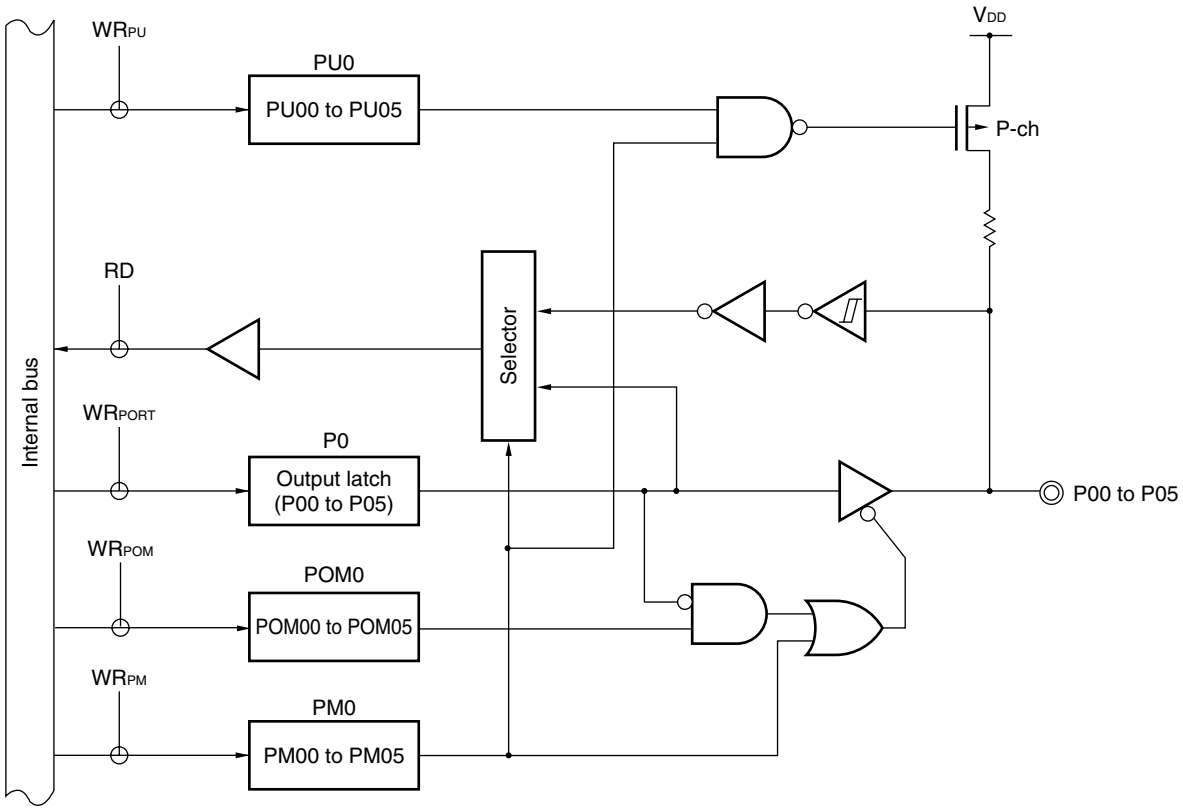
Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P05 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Output from the P00 to P05 pins can be specified as P-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 0 (POM0).

Reset signal generation sets port 0 to input mode.

Figure 4-1 show block diagram of port 0.

Figure 4-1. Block Diagram of P00 to P05



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

4.2.2 Port 3

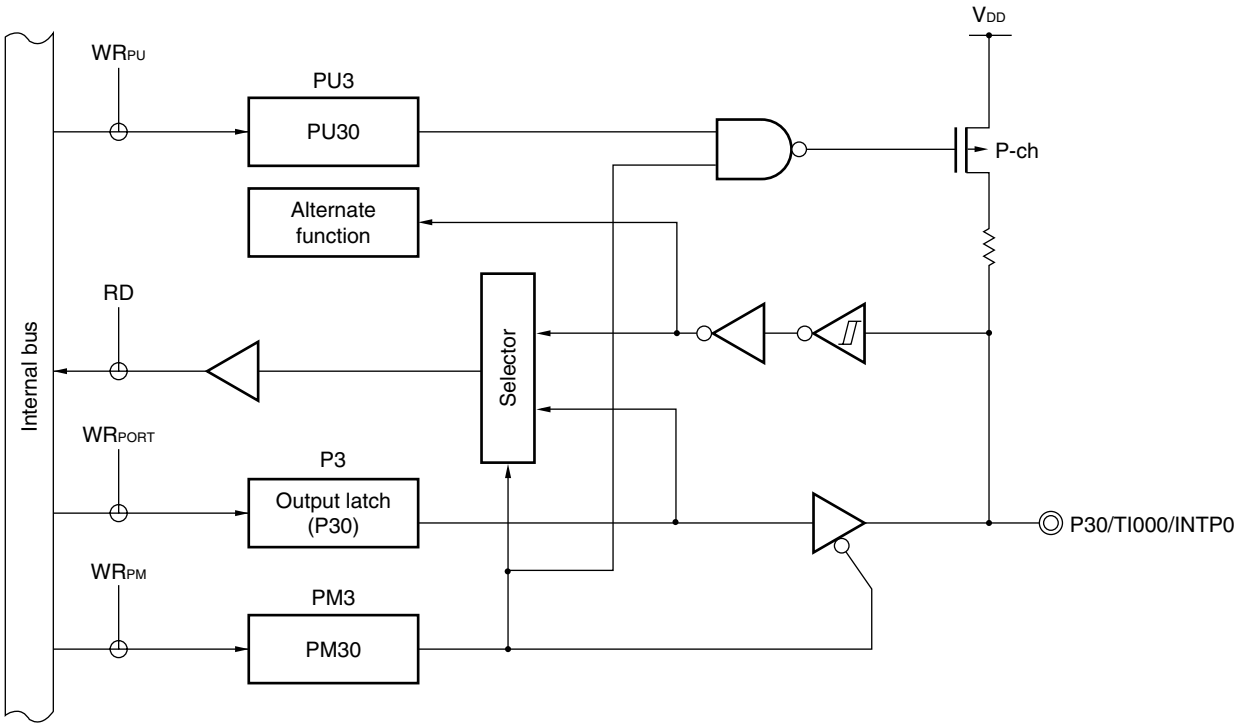
Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to P34 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

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

Reset signal generation sets port 3 to input mode.

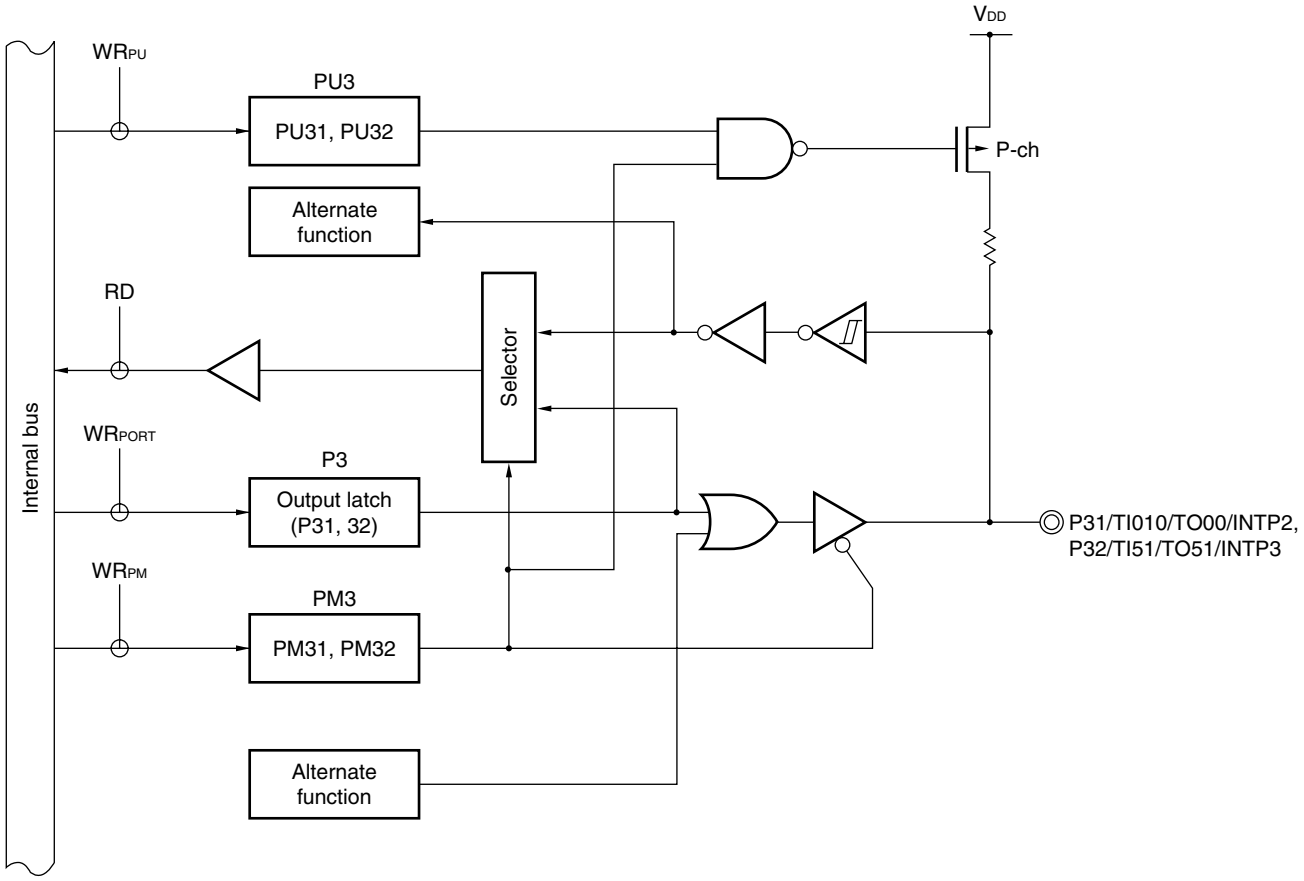
Figures 4-2 to 4-4 show block diagrams of port 3.

Figure 4-2. Block Diagram of P30



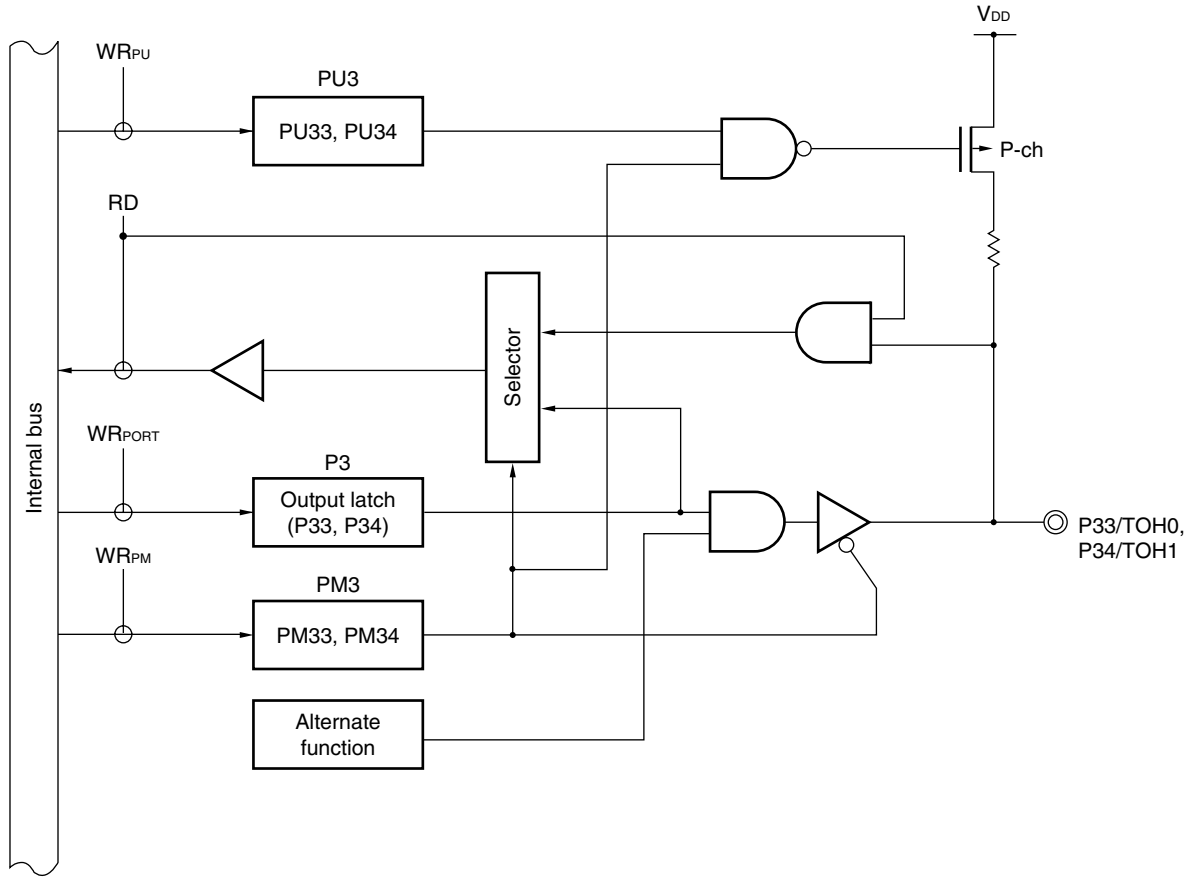
P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-3. Block Diagram of P31 and P32



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

Figure 4-4. Block Diagram of P34



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

4.2.3 Port 6

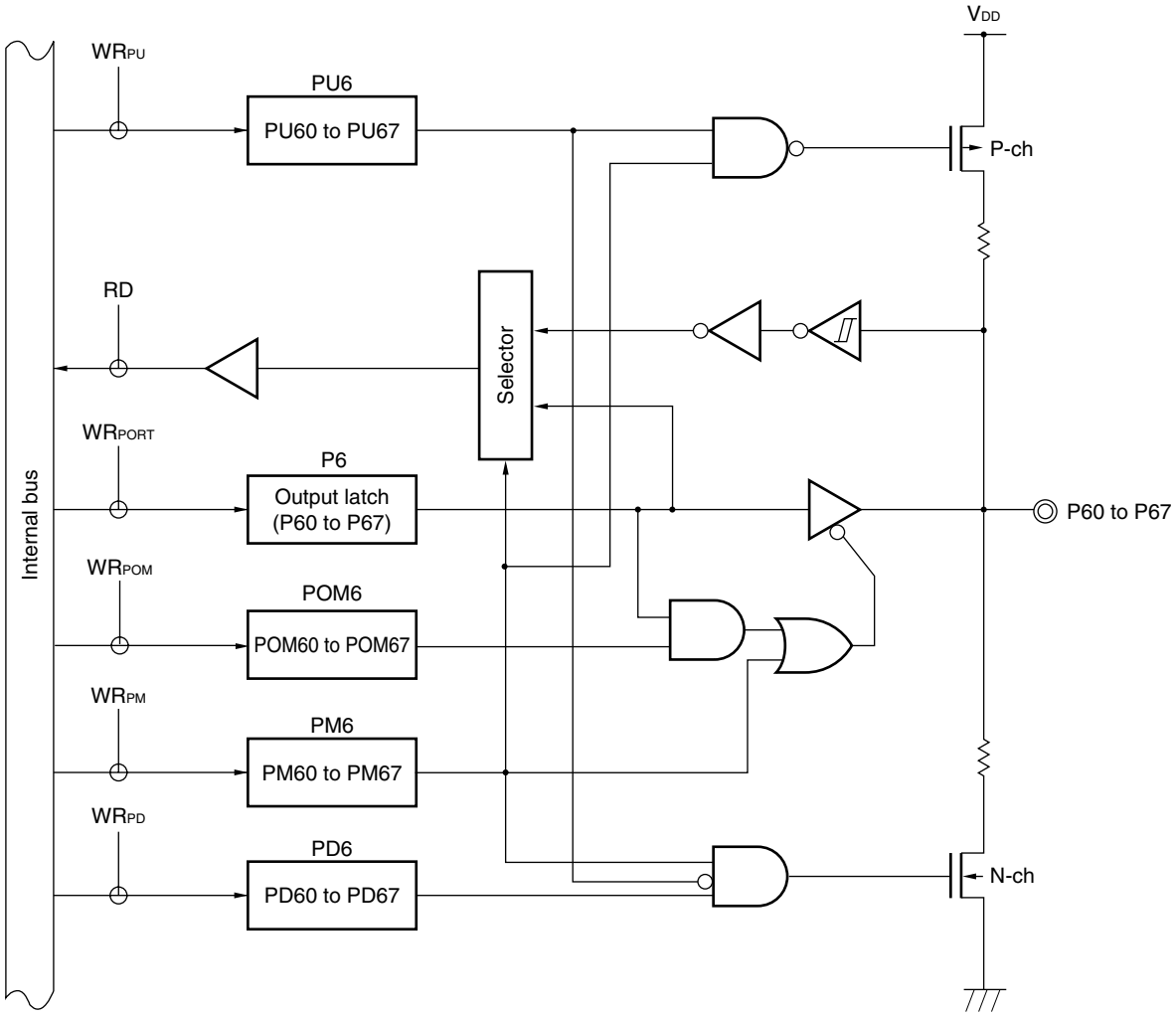
Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6). When the P60 to P67 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 6 (PU6).

Output from the P60 to P67 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 6 (POM6).

Reset signal generation sets port 6 to input mode.

Figure 4-5 show block diagram of port 6.

Figure 4-5. Block Diagram of P60 to P67



P6: Port register 6
PU6: Pull-up resistor option register 6
PD6: Pull-down resistor option register 6
PM6: Port mode register 6
POM6: Port output mode register 6
RD: Read signal
WR××: Write signal

4.2.4 Port 7

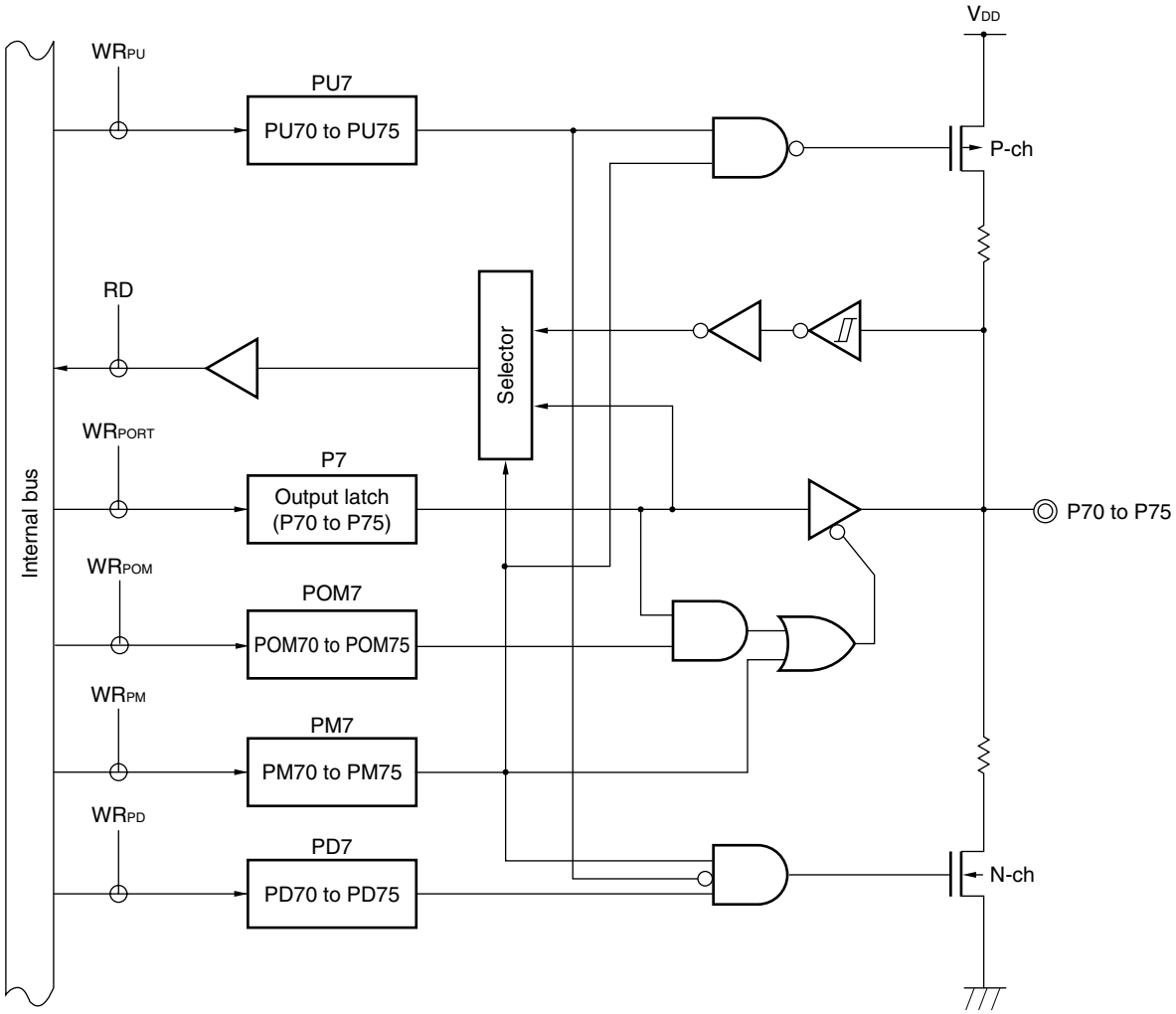
Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P75 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7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

Output from the P70 to 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).

Reset signal generation sets port 7 to input mode.

Figure 4-6 show block diagram of port 7.

Figure 4-6. Block Diagram of P70 to P75



P7: Port register 7
PU7: Pull-up resistor option register 7
PD7: Pull-down resistor option register 7
PM7: Port mode register 7
POM7: Port output mode register 7
RD: Read signal
WR××: Write signal

4.2.5 Port 12

P121 and P122 function as an I/O port with an output latch. P121 and P122 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12).

P125 functions as an Input port.

When used as an input port for P125, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

This port can also be used as pins for connecting resonator for main system clock, external clock input for main system clock, external reset input, clock input and data I/O for flash memory programmer/on-chip debugger, and external interrupt request input.

Set bit 5 (RSTM) of the reset pin mode register (RSTMASK) to 1 when using P125/INTP1/$\overline{\text{RESET}}$ as an input port, and clear RSTM to 0 when using P125/INTP1/$\overline{\text{RESET}}$ as an external reset input.

Reset signal generation sets port 12 to input mode.

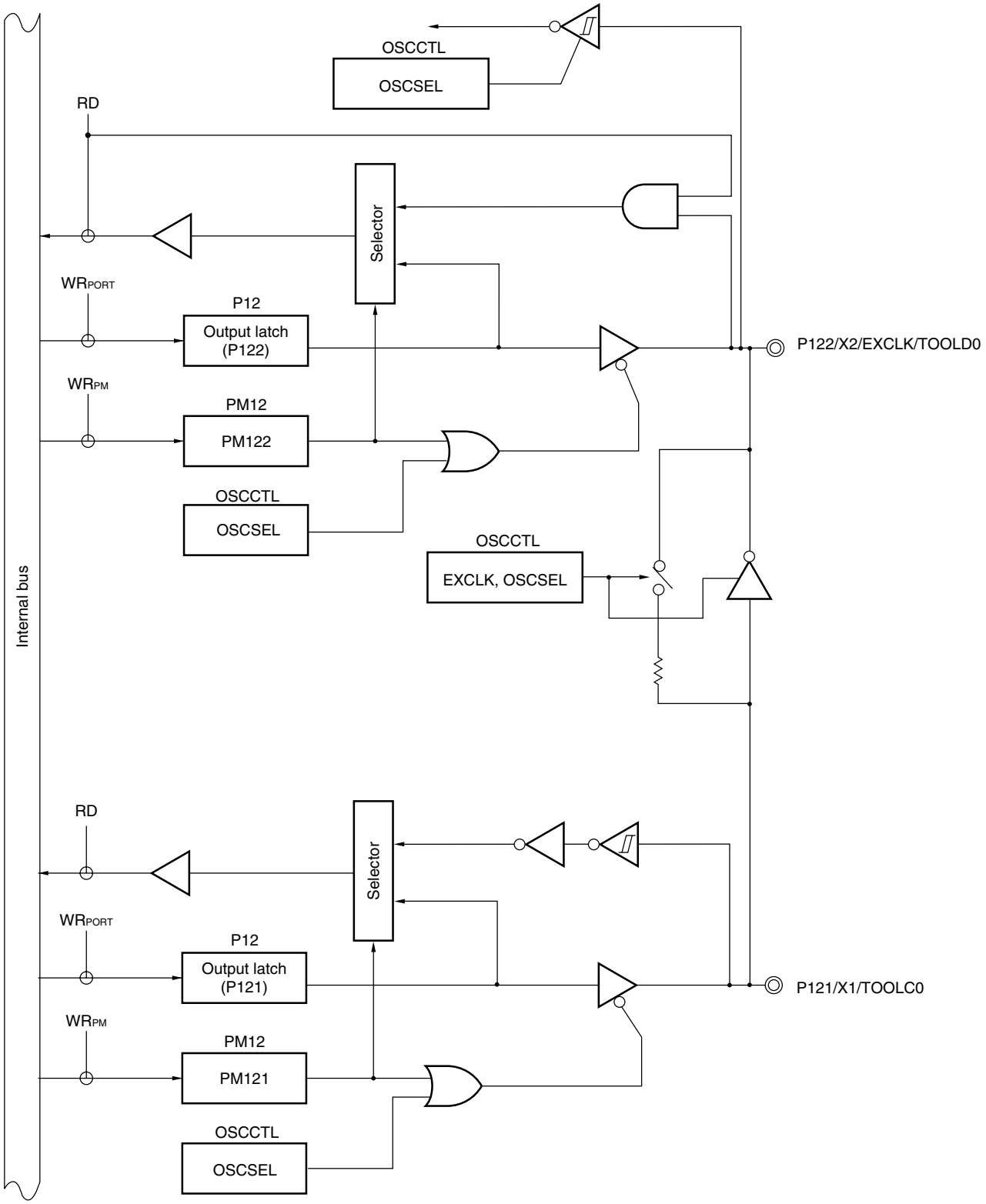
Figures 4-7, 4-8 show block diagrams of port 12.

Cautions 1. When using the P121, P122 pins to connect a resonator for the main system clock (X1, X2) or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode or external clock input mode must be set by using the clock operation mode select register (OSCCTL) (for details, refer to 5.3 (1) Clock operation mode select register (OSCCTL)). The reset value of OSCCTL is 00H (all of the P121, P122 pins are input port pins).

2. $\overline{\text{RESET}}$/INTP1/P125 is set in an external reset input after a reset release.

3. Because $\overline{\text{RESET}}$/INTP1/P125 is set in the external reset input immediately after release of reset, if a reset signal is generated during low level input, the reset status continues until the input rises to the high level.

Figure 4-7. Block Diagram of P121, P122



P12: Port register 12

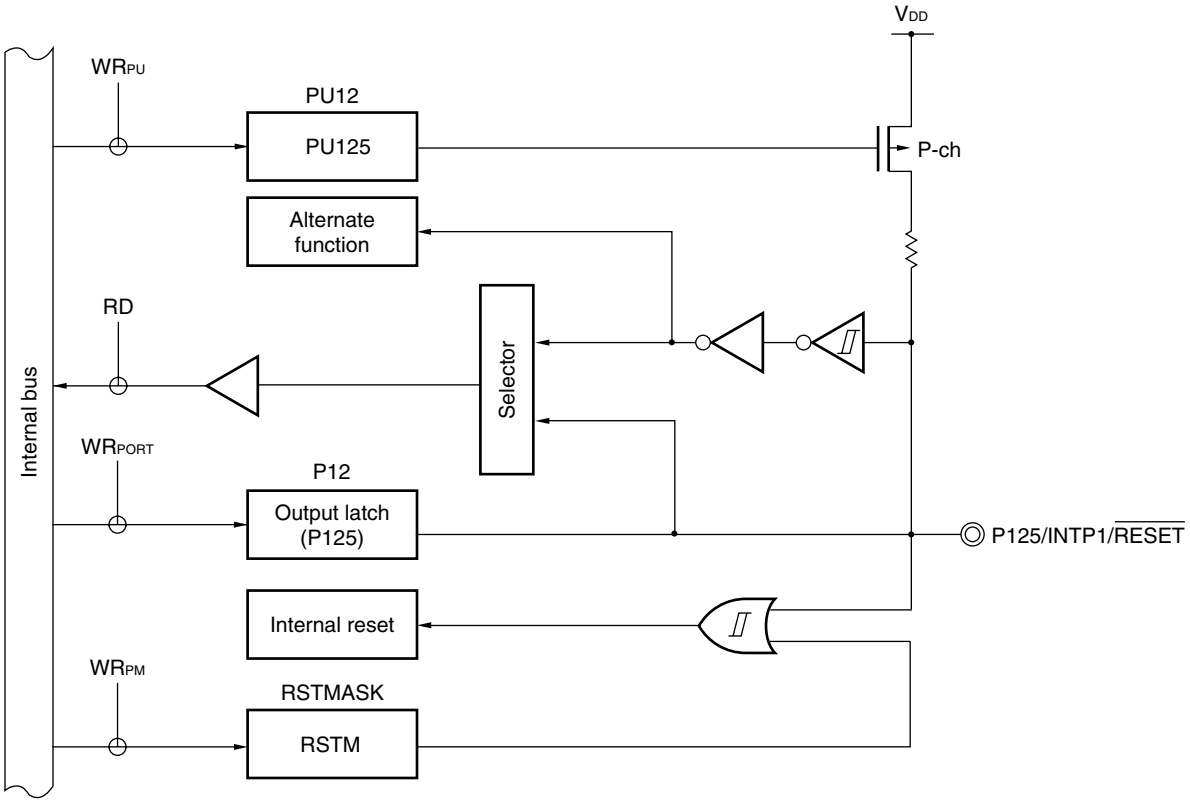
PM12: Port mode register 12

OSCCTL: Clock operation mode select register

RD: Read signal

WR××: Write signal

Figure 4-8. Block Diagram of P125



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

Caution Because $\overline{\text{RESET}}$/P125 is set in the external reset input immediately after release of reset, if a reset signal is generated during low level input, the reset status continues until the input rises to the high level.

Remark After reset, the external reset function and the pull-up resistor are enabled (RSTM = 0, PU125 = 1). Set RSTM bit to 1 when using as a port function.

4.3 Registers Controlling Port Function

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

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Pull-down resistor option registers (PDxx)
- Reset pin mode register (RSTMASK)

(1) Port mode registers (PMxx)

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

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

Reset signal generation sets these registers to FFH.

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

Figure 4-9. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FF26H	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FF27H	FFH	R/W
PM12	1	1	1	1	1	PM122	PM121	1	FF2CH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 3, 6, 7, 12; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 6 and 7 of PM0, bits 5 to 7 of PM3, bits 6 and 7 of PM7, bits 0 and 3 to 7 of PM12 to 1.

(2) Port registers (Pxx)

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.

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

Reset signal generation clears these registers to 00H.

Figure 4-10. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	P05	P04	P03	P02	P01	P00	FF00H	00H (output latch)	R/W
P3	0	0	0	P34	P33	P32	P31	P30	FF03H	00H (output latch)	R/W
P6	P67	P66	P65	P64	P63	P62	P61	P60	FF06H	00H (output latch)	R/W
P7	0	0	P75	P74	P73	P72	P71	P70	FF07H	00H (output latch)	R/W
P12	0	0	P125	0	0	P122Note	P121Note	0	FF0CH	00H (output latch)	R/W

Pmn	m = 0, 3, 6, 7, 12; 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 “0” is always read from the output latch of the pin in the X1 oscillation mode or external clock input mode.

(3) Pull-up resistor option registers (PUxx)

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

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

Reset signal generation clears these registers to 00H (sets only PU12 to 20H).

Figure 4-11. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	PU05	PU04	PU03	PU02	PU01	PU00	FF30H	00H	R/W
PU3	0	0	0	PU34	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU6	PU67	PU66	PU65	PU64	PU63	PU62	PU61	PU60	FF36H	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	FF37H	00H	R/W
PU12	0	0	PU125	0	0	0	0	0	FF3CH	20H	R/W

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

Caution When same bit of both PU6 and PD6 or PU7 and PD7 are set to 1, connection of the on-chip pull-up resistor is valid, but the pull-down resistor is invalid.

(4) Pull-down resistor option registers (PDxx)

These registers specify whether the on-chip pull-down resistors are to be used or not. On-chip pull-down 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-down resistor has been specified in these registers. On-chip pull-down resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of these registers.

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

Reset signal generation clears these registers to 00H.

Figure 4-12. Format of Pull-down Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PD6	PD67	PD66	PD65	PD64	PD63	PD62	PD61	PD60	FF38H	00H	R/W
PD7	0	0	PD75	PD74	PD73	PD72	PD71	PD70	FF39H	00H	R/W

PDmn	Pmn pin on-chip pull-down resistor selection (m = 6, 7; n = 0 to 7)
0	On-chip pull-down resistor not connected
1	On-chip pull-down resistor connected

Caution When same bit of both PU6 and PD6 or PU7 and PD7 are set to 1, connection of the on-chip pull-up resistor is valid, but the pull-down resistor is invalid.

(5) Port output mode registers (POM0, POM6, POM7)

These registers set the output mode of P00 to P05, P60 to P67, P70 to P75 in 1-bit units.

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

Reset signal generation clears these registers to 00H.

Figure 4-13. Format of Port Output Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	POM05	POM04	POM03	POM02	POM01	POM00	F0050H	00H	R/W

POM0n	P0n pin output mode selection (n = 0 to 5)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

POM6	POM67	POM66	POM65	POM64	POM63	POM62	POM61	POM60	F0056H	00H	R/W
POM7	0	0	POM75	POM74	POM73	POM72	POM71	POM70	F0057H	00H	R/W

POMmn	Pmn pin output mode selection (m = 6, 7; n = 0 to 7)
0	CMOS output mode (normal I/O output capacity)
1	P-ch open-drain output (V_{DD} tolerance) mode (high current output capability)

(6) Reset pin mode register (RSTMASK)

This register sets the pin function of $\overline{\text{RESET}}$/INTP1/P125 (external reset input/input-dedicated port).

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

Reset signal generation clears this register to 00H.

Figure 4-14. Format of Reset Pin Mode Register (RSTMASK)

Address: FF2DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RSTMASK	0	0	RSTM	0	0	0	0	0

RSTM	$\overline{\text{RESET}}$/INTP1/P125 pin function selection
0	Using as external reset input ($\overline{\text{RESET}}$)
1	Using as input-dedicated port (P125/INTP1)

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.

4.5 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 4-4**.

Table 4-4. Settings of Port Mode Register and Output Latch When Using Alternate Function

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P30	TI000	Input	1	×
	INTP0	Input	1	×
P31	TI010	Input	1	×
	TO00	Output	0	0
	INTP2	Input	1	×
P32	TI51	Input	1	×
	TO51	Output	0	0
	INTP3	Input	1	×
P33	TOH0	Output	0	0
P34	TOH1	Output	0	0
P121	X1 Note 1	–	×	×
	TOOLC0	Input	×	×
P122	X2 Note 1	–	×	×
	EXCLK Note 1	Input	×	×
	TOOLD0	I/O	×	×
P125	INTP1	Input	1	×
	$\overline{\text{RESET}}$ Note 2	Input	×	×

Notes 1. When using the P121 to P124 pins to connect a resonator for the main system clock (X1, X2), or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode, or external clock input mode must be set by using OSCCTL register (for details, refer to 5.3 (1) Clock operation mode select register (OSCCTL). The reset value of OSCCTL is 00H (all of the P121 and P122 pins are Input port pins).

2. Clear RSTM bit (bit 5 of RSTMASK register) to 0 when using P125 as an external reset input ($\overline{\text{RESET}}$).

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

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 P60 is an output port, P61 to P67 are input ports (all pin statuses are high level), and the port latch value of port 6 is 00H, if the output of output port P60 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 6 is FFH.

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

A 1-bit manipulation instruction is executed in the following order in the µPD79F7025, 79F7026 microcontrollers.

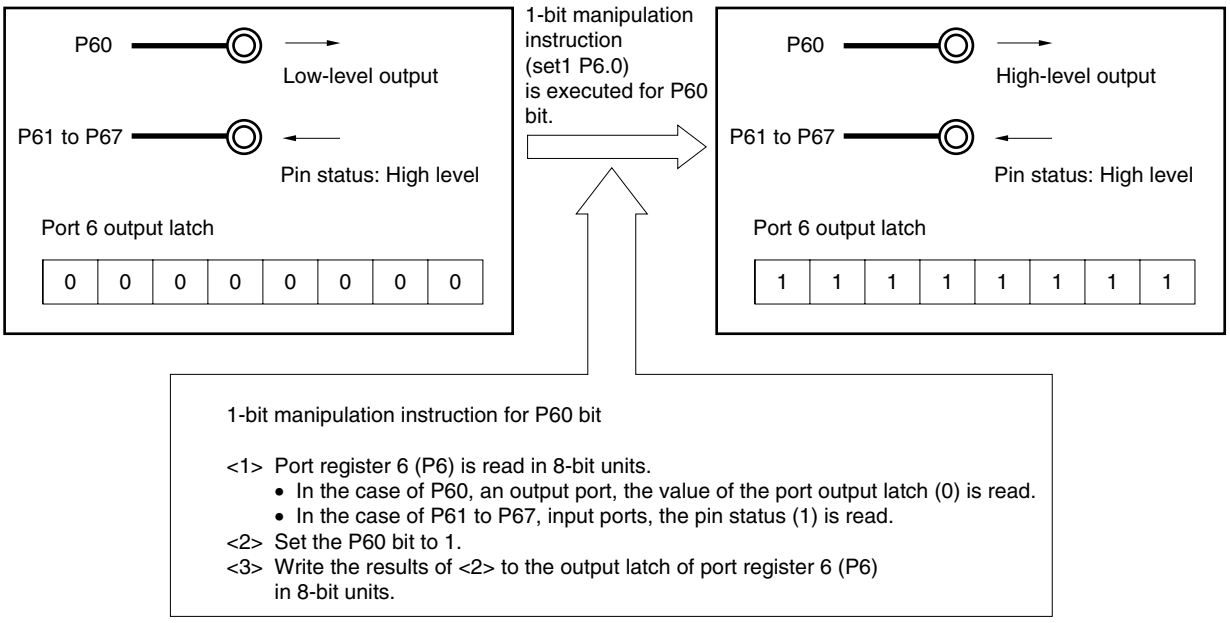
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P60, which is an output port, is read, while the pin statuses of P61 to P67, which are input ports, are read. If the pin statuses of P61 to P67 are high level at this time, the read value is FEH.

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

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

Figure 4-15 1-Bit Manipulation Instruction (P60)



Remark The following instructions are 1-bit manipulation instructions.

- MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1

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 = 1 to 10 MHz by connecting a resonator to X1 and X2.
Oscillation can be stopped by executing the STOP instruction or using the main OSC control register (MOC).

<2> Internal high-speed oscillator

This circuit oscillates a clock of f_{IH} = 4 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 using the internal oscillation mode register (RCM).

An external main system clock (f_{EXCLK} = 1 to 10 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 using RCM.
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 using the main clock mode register (MCM).

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{EXCLK}: External main system clock frequency

(2) Internal low-speed oscillation clock (clock for watchdog timer)

- **Internal low-speed oscillator**

This circuit oscillates a clock of f_{IL} = 240 kHz (TYP.). After a reset release, the internal low-speed oscillation clock always starts operating.
Oscillation can be stopped by using the internal oscillation mode register (RCM) when “internal low-speed oscillator can be stopped by software” is set by option byte.
The internal low-speed oscillation clock cannot be used as the CPU clock. The following hardware operates with the internal low-speed oscillation clock.

- Watchdog timer
- 8-bit timer H1 (when f_{IL}, $f_{IL}/2^7$, or $f_{IL}/2^9$ is selected)

Remark f_{IL}: Internal low-speed oscillation clock frequency

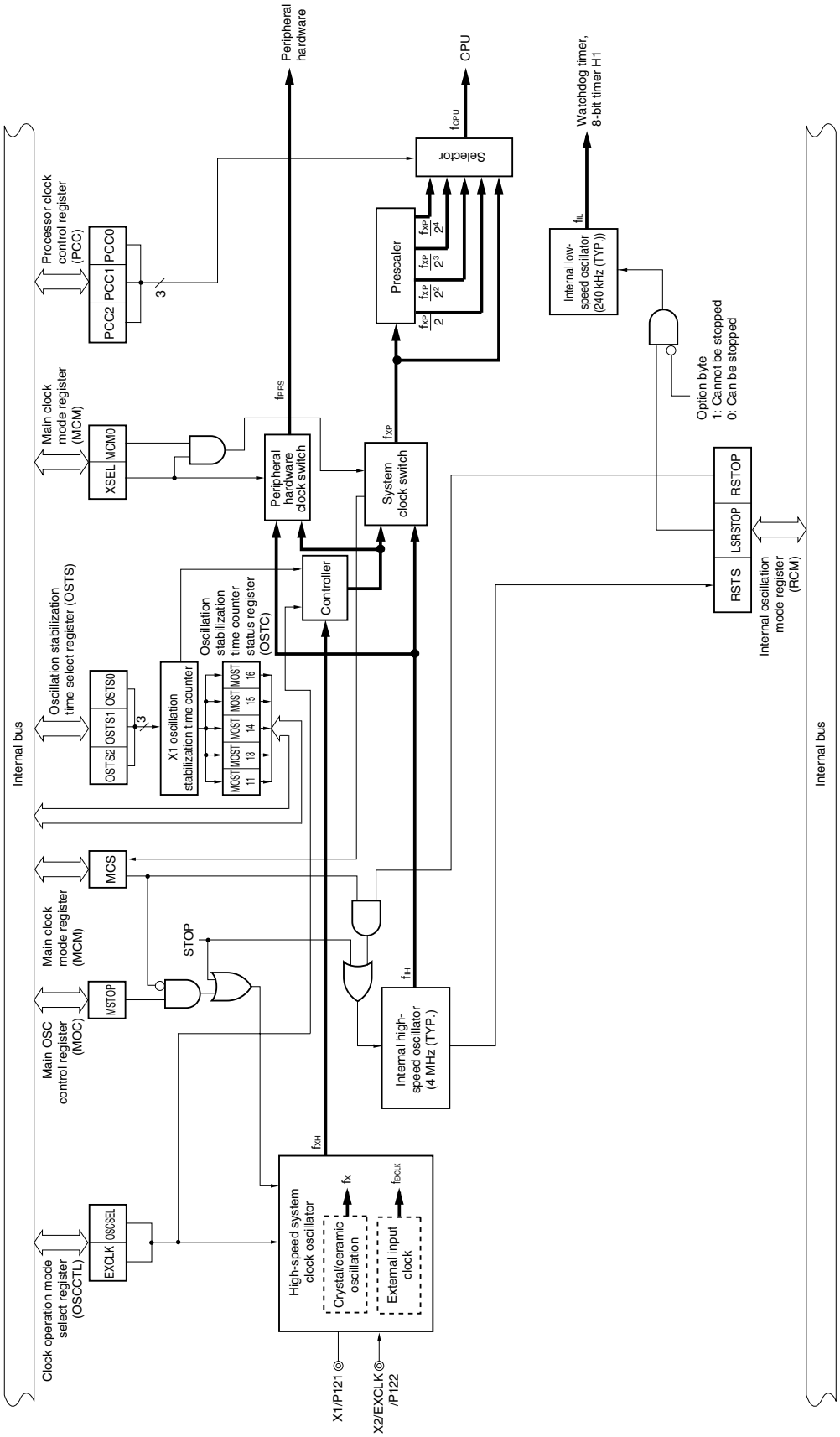
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode select register (OSCCTL) Processor clock control register (PCC) Internal oscillation mode register (RCM) Main OSC control register (MOC) Main clock mode register (MCM) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS)
Oscillators	X1 oscillator Internal high-speed oscillator Internal low-speed oscillator

Figure 5-1. Block Diagram of Clock Generator



Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{EXCLK}: External main system clock frequency
f_{XH}: High-speed system clock frequency
f_{XP}: Main system clock frequency
f_{PRS}: Peripheral hardware clock frequency
f_{CPU}: CPU clock frequency
f_{IL}: Internal low-speed oscillation clock frequency

5.3 Registers Controlling Clock Generator

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

- Clock operation mode select register (OSCCTL)
- Processor clock control register (PCC)
- Internal oscillation mode register (RCM)
- Main OSC control register (MOC)
- Main clock mode register (MCM)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

(1) Clock operation mode select register (OSCCTL)

This register selects the operation modes of the high-speed system clock.

OSCCTL can be set 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 Select Register (OSCCTL)

Address: FF9FH After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	2	1	0
OSCCTL	EXCLK	OSCSEL	0	0	0	0	0	0

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

Cautions 1. To change the value of EXCLK and OSCSEL, be sure to confirm that bit 7 (MSTOP) of the main OSC control register (MOC) is 1 (the X1 oscillator stops or the external clock from the EXCLK pin is disabled).

2. Be sure to clear bits 0 to 5 to 0.

(2) Processor clock control register (PCC)

This register is used to select the CPU clock and the division ratio.

PCC is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PCC to 01H.

Figure 5-3. Format of Processor Clock Control Register (PCC)

Address: FFFBH After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	PCC2	PCC1	PCC0

PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection
0	0	0	f_{XP}
0	0	1	f_{XP}/2 (default)
0	1	0	$f_{XP}/2^2$
0	1	1	$f_{XP}/2^3$
1	0	0	$f_{XP}/2^4$
Other than above			Setting prohibited

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

2. The peripheral hardware clock (f_{PRS}) is not divided when the division ratio of the PCC is set.

Remark f_{XP}: Main system clock oscillation frequency

The fastest instruction can be executed in 2 clocks of the CPU clock in the µPD79F7025, 79F7026 microcontrollers. Therefore, the relationship between the CPU clock (f_{CPU}) and the minimum instruction execution time is as shown in Table 5-2.

Table 5-2. Relationship between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$	
	Main System Clock	
	High-Speed System Clock[Note]	Internal High-Speed Oscillation Clock[Note]
	At 10 MHz Operation	At 4 MHz (TYP.) Operation
f_{XP}	0.2 µs	0.5 µs (TYP.)
$f_{XP}/2$	0.4 µs	1.0 µs (TYP.)
$f_{XP}/2^2$	0.8 µs	2.0 µs (TYP.)
$f_{XP}/2^3$	1.6 µs	4.0 µs (TYP.)
$f_{XP}/2^4$	3.2 µs	8.0 µs (TYP.)

Note The main clock mode register (MCM) is used to set the main system clock supplied to CPU clock (high-speed system clock/internal high-speed oscillation clock) (refer to **Figure 5-3**).

(3) Internal oscillation mode register (RCM)

This register sets the operation mode of internal oscillator.

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

Reset signal generation sets this register to 80H [Note 1].

Figure 5-4. Format of Internal Oscillation Mode Register (RCM)

Address: FFA0H After reset: 80H[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	2	<1>	<0>
RCM	RSTS	0	0	0	0	0	LSRSTOP	RSTOP

RSTS	Status of internal high-speed oscillator
0	Waiting for accuracy stabilization of internal high-speed oscillator
1	Stability operating of internal high-speed oscillator

LSRSTOP	Internal low-speed oscillator oscillating/stopped
0	Internal low-speed oscillator oscillating
1	Internal low-speed oscillator stopped

RSTOP	Internal high-speed oscillator oscillating/stopped
0	Internal high-speed oscillator oscillating
1	Internal high-speed oscillator stopped

Notes 1. The value of this register is 00H immediately after a reset release but automatically changes to 80H after internal high-speed oscillator has been stabilized.

2. Bit 7 is read-only.

Caution When setting RSTOP to 1, be sure to confirm that the CPU operates with a clock other than the internal high-speed oscillation clock. Specifically, set under the following condition.

- **When MCS = 1 (when CPU operates with the high-speed system clock)**

In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock before setting RSTOP to 1.

(4) Main OSC control register (MOC)

This register selects the operation mode of the high-speed system clock.

This register is used to stop the X1 oscillator or to disable an external clock input from the EXCLK pin when the CPU operates with a clock other than the high-speed system clock.

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

Reset signal generation sets this register to 80H.

Figure 5-5. Format of Main OSC Control Register (MOC)

Address: FFA2H After reset: 80H R/W

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

MSTOP	Control of high-speed system clock operation	
	X1 oscillation mode	External clock input mode
0	X1 oscillator operating	External clock from EXCLK pin is enabled
1	X1 oscillator stopped	External clock from EXCLK pin is disabled

Cautions 1. When setting MSTOP to 1, be sure to confirm that the CPU operates with a clock other than the high-speed system clock. Specifically, set under the following condition.

- **When MCS = 0 (when CPU operates with the internal high-speed oscillation clock)**

In addition, stop peripheral hardware that is operating on the high-speed system clock before setting MSTOP to 1.

2. Do not clear MSTOP to 0 while bit 6 (OSCSEL) of the clock operation mode select register (OSCCTL) is 0 (input port mode).

3. The peripheral hardware cannot operate when the peripheral hardware clock is stopped. To resume the operation of the peripheral hardware after the peripheral hardware clock has been stopped, initialize the peripheral hardware.

(5) Main clock mode register (MCM)

This register selects the main system clock supplied to CPU clock and clock supplied to peripheral hardware clock.

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

Reset signal generation clears this register to 00H.

Figure 5-6. Format of Main Clock Mode Register (MCM)

Address: FFA1H After reset: 00H R/W[Note]

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

XSEL	MCM0	Selection of clock supplied to main system clock and peripheral hardware	
		Main system clock (f_{XP})	Peripheral hardware clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{IH})	Internal high-speed oscillation clock (f_{IH})
0	1		
1	0		High-speed system clock (f_{XH})
1	1	High-speed system clock (f_{XH})	

MCS	Main system clock status
0	Operates with internal high-speed oscillation clock
1	Operates with high-speed system clock

Note Bit 1 is read-only.

Cautions 1. XSEL can be changed only once after a reset release.

2. A clock other than f_{PRS} is supplied to the following peripheral functions regardless of the setting of XSEL and MCM0.

- **Watchdog timer (operates with internal low-speed oscillation clock)**
- **When "f_{IL}", "$f_{IL}/2^7$", or "$f_{IL}/2^9$" is selected as the count clock for 8-bit timer H1 (operates with internal low-speed oscillation clock)**
- **Peripheral hardware selects the external clock as the clock source (Except when the external count clock of TM00 is selected (TI000 pin valid edge))**

(6) Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter. When X1 clock oscillation starts with the internal high-speed oscillation clock used as the CPU clock, the X1 clock oscillation stabilization time can be checked.

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, and WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Figure 5-7. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status	
						fx = 10 MHz
1	0	0	0	0	2^{11}/fx min.	204.8 µs min.
1	1	0	0	0	2^{13}/fx min.	819.2 µs min.
1	1	1	0	0	2^{14}/fx min.	1.64 ms min.
1	1	1	1	0	2^{15}/fx min.	3.27 ms min.
1	1	1	1	1	2^{16}/fx min.	6.55 ms min.

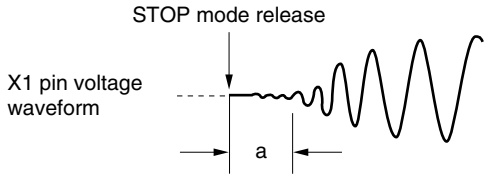
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

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 fx: X1 clock oscillation frequency

(7) 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 OSTS to 05H.

Figure 5-8. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection	
				fx = 10 MHz
0	0	1	2^{11}/fx	204.8 µs
0	1	0	2^{13}/fx	819.2 µs
0	1	1	2^{14}/fx	1.64 ms
1	0	0	2^{15}/fx	3.27 ms
1	0	1	2^{16}/fx	6.55 ms
Other than above			Setting prohibited	

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

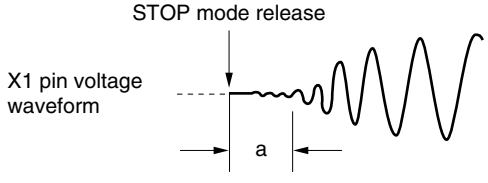
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

3. 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.

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



Remark fx: X1 clock oscillation frequency

5.4 System Clock Oscillator

5.4.1 X1 oscillator

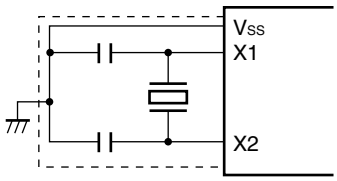
The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 10 MHz) connected to the X1 and X2 pins.

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

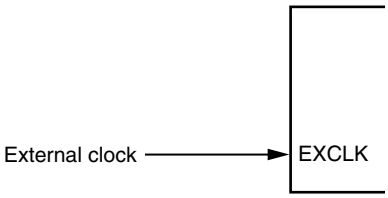
Figure 5-9 shows an example of the external circuit of the X1 oscillator.

Figure 5-9. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Cautions are listed on the next page.

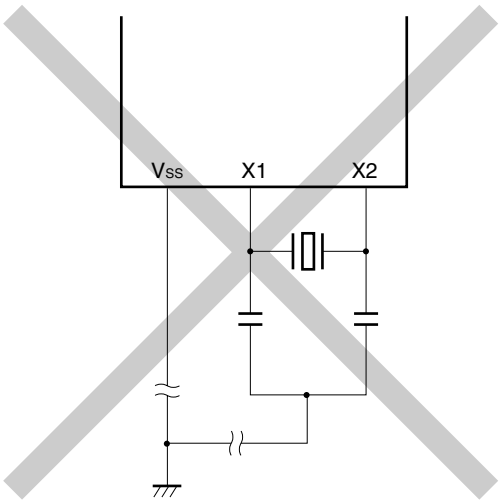
Caution When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the Figure 5-10 to avoid an adverse effect from wiring capacitance.

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

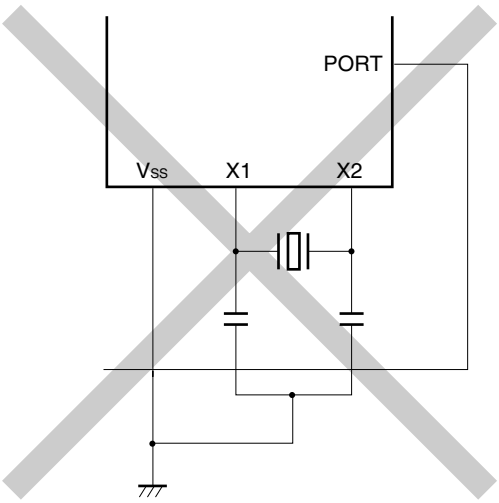
Figure 5-10 shows examples of incorrect resonator connection.

Figure 5-10. Examples of Incorrect Resonator Connection (1/2)

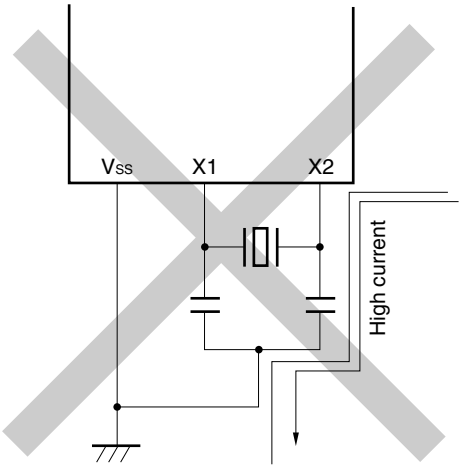
(a) Too long wiring



(b) Crossed signal line



(c) Wiring near high alternating current



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

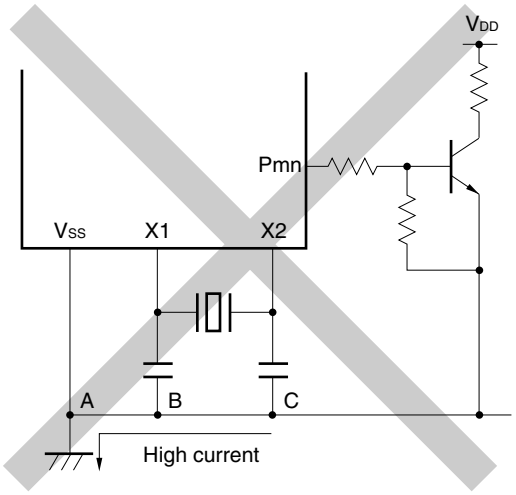
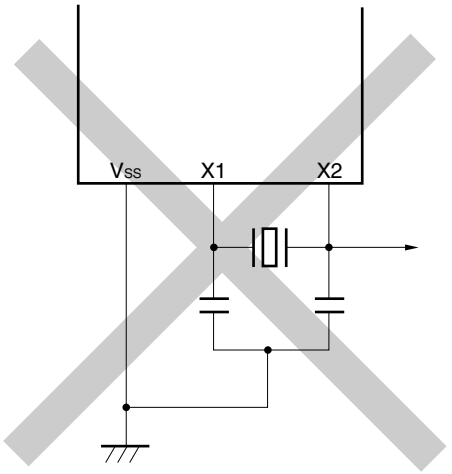


Figure 5-10. Examples of Incorrect Resonator Connection (2/2)

(e) Signals are fetched



5.4.2 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the µPD79F7025, 79F7026. Oscillation can be controlled by the internal oscillation mode register (RCM).

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

5.4.3 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the µPD79F7025, 79F7026.

The internal low-speed oscillation clock is only used as the watchdog timer and the clock of 8-bit timer H1. The internal low-speed oscillation clock cannot be used as the CPU clock.

“Can be stopped by software” or “Cannot be stopped” can be selected by the option byte. When “Can be stopped by software” is set, oscillation can be controlled by the internal oscillation mode register (RCM).

After a reset release, the internal low-speed oscillator automatically starts oscillation, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation is enabled using the option byte.

5.4.4 Prescaler

The prescaler generates the CPU clock by dividing the main system clock when the main system clock is selected as the clock to be supplied to the CPU.

5.5 Clock Generator Operation

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

- Main system clock f_{XP}
  - High-speed system clock f_{XH}
    X1 clock f_X
    External main system clock f_{EXCLK}
  - Internal high-speed oscillation clock f_{IH}
- Internal low-speed oscillation clock f_{IL}
- CPU clock f_{CPU}
- Peripheral hardware clock f_{PRS}

The CPU starts operation when the internal high-speed oscillator starts outputting after a reset release in the μPD79F7025, 79F7026 microcontrollers, thus enabling the following.

(1) Enhancement of security function

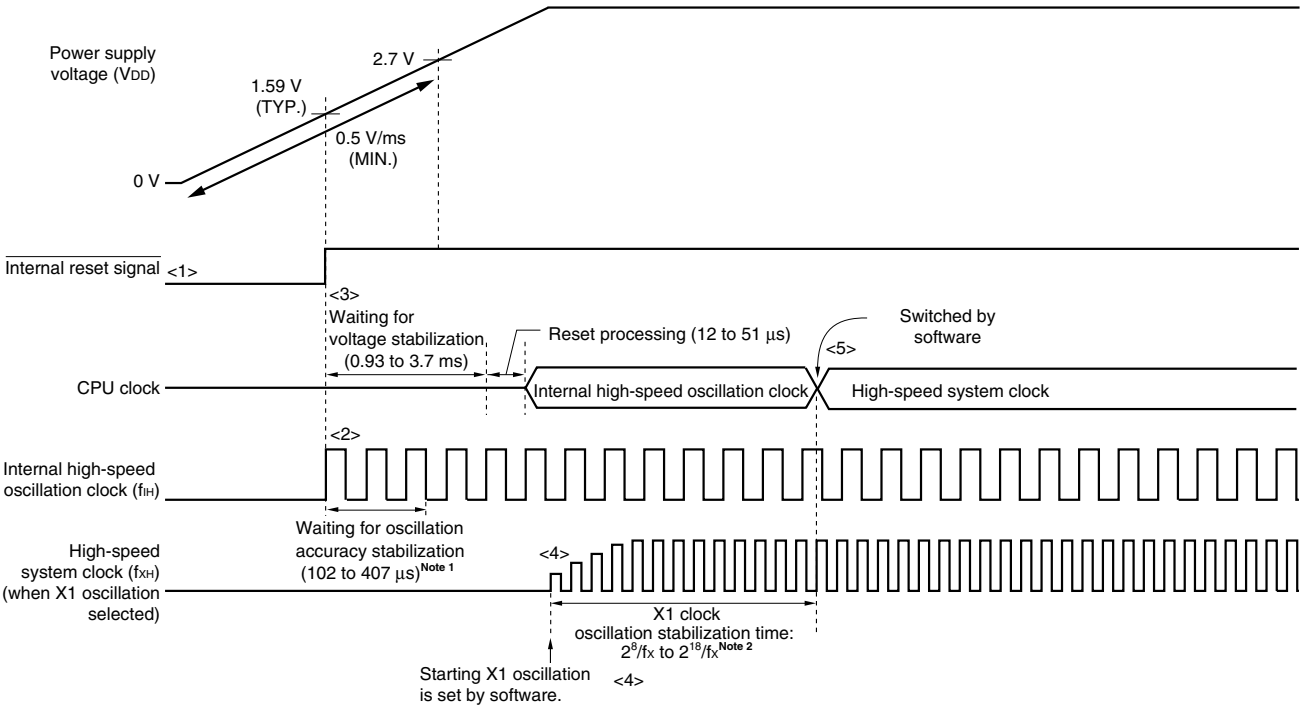
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started by the internal high-speed oscillation clock after a reset release. Consequently, the system can be safely shut down by performing a minimum operation, such as acknowledging a reset source by software or performing safety processing when there is a malfunction.

(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 Figures 5-11 and 5-12.

Figure 5-11. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVISTART = 0))



<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 1.59 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 clock via software (refer to **(1)** in **5.6.1 Example of controlling high-speed system clock**).

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

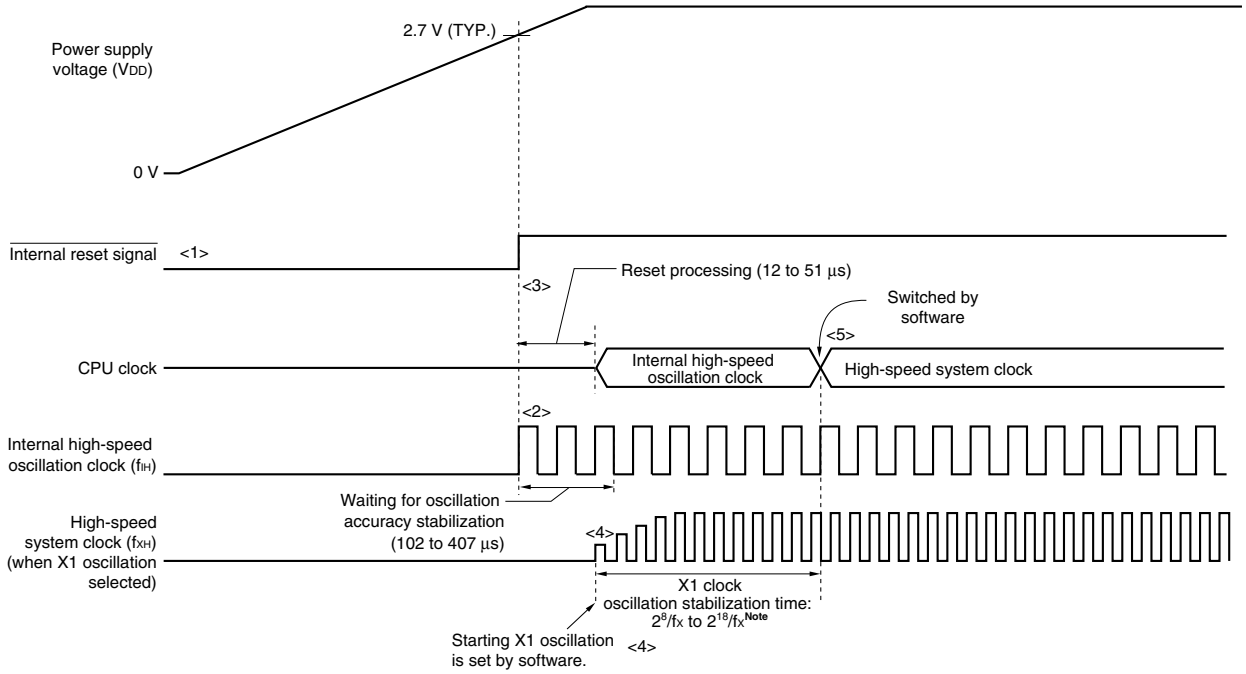
Cautions 1. If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 2.7 V, input a low level to the $\overline{\text{RESET}}$ pin from power application until the voltage reaches 2.7 V, or set the LVI default start function enabled by using the option byte (LVISTART = 1) (refer to Figure 5-12). When a low level has been input to the $\overline{\text{RESET}}$ pin until the voltage reaches 2.7 V, the CPU operates with the same timing as <2> and thereafter in Figure 5-11, after the reset has been released 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.

Remarks 1. Operation to be initiated at $V_{DD} \leq 2.7$ V, recommend that LVI default start function is set to ON.

2. 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 (refer to **(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**).

Figure 5-12. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVISTART = 1))



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

<2> When the power supply voltage exceeds 2.7 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 clock via software (refer to **(1)** in **5.6.1 Example of controlling high-speed system clock**).

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

Note 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 (0.93 to 3.7 ms) is required after the supply voltage reaches 1.59 V (TYP.). If the supply voltage rises from 1.59 V (TYP.) to 2.7 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.

Remarks 1. Operation to be initiated at $V_{DD} \leq 2.7$ V, recommend that LVI default start function is set to ON.

2. 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 (refer to **(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**).

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 across 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 clock and 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 selecting X1 clock or external clock (OSCCTL register)
When EXCLK is cleared to 0 and OSCSEL is set to 1, the mode is switched from port mode to X1 oscillation mode.

EXCLK	OSCSEL	Operation Mode of High-Speed System Clock Pin	P121/X1 Pin	P122/X2/EXCLK Pin
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	

<2> Controlling oscillation of X1 clock (MOC 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. Do not change the value of EXCLK and OSCSEL while the X1 clock is operating.
2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (refer to CHAPTER 20 ELECTRICAL SPECIFICATIONS).

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

<1> Setting P121/X1 and P122/X2/EXCLK pins and selecting operation mode (OSCCTL register)
When EXCLK and OSCSEL are set to 1, the mode is switched from port mode to external clock input mode.

EXCLK	OSCSEL	Operation Mode of High-Speed System Clock Pin	P121/X1 Pin	P122/X2/EXCLK Pin
1	1	External clock input mode	Input port	External clock input

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

Cautions 1. Do not change the value of EXCLK and OSCSEL while the external main system clock is operating.
2. Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (refer to CHAPTER 20 ELECTRICAL SPECIFICATIONS).

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

<1> Setting high-speed system clock oscillation[Note]
(Refer to **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 main system clock (MCM register)

When XSEL and MCM0 are set to 1, the high-speed system clock is supplied as the main system clock and peripheral hardware clock.

XSEL	MCM0	Selection of Main System Clock and Clock Supplied to Peripheral Hardware	
		Main System Clock (f_{XP})	Peripheral Hardware Clock (f_{PRS})
1	1	High-speed system clock (f_{XH})	High-speed system clock (f_{XH})

Caution If the high-speed system clock is selected as the main system clock, a clock other than the high-speed system clock cannot be set as the peripheral hardware clock.

<3> Setting the main system clock as the CPU clock and selecting the division ratio (PCC register)

The main system clock is supplied to the CPU. To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
0	0	0	f_{XP}
0	0	1	f_{XP}/2 (default)
0	1	0	$f_{XP}/2^2$
0	1	1	$f_{XP}/2^3$
1	0	0	$f_{XP}/2^4$
Other than above			Setting prohibited

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

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

- Executing the STOP instruction and stopping the X1 oscillation (disabling clock input if the external clock is used)
- Setting MSTOP to 1 and stopping the X1 oscillation (disabling clock input if the external clock is used)

(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, refer to **CHAPTER 11 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after standby release

When the CPU is operating on the X1 clock, set the value of the OSTS register before the STOP instruction is executed.

<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 (PCC and MCM registers)

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 a clock other than the high-speed system clock.

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

<2> Stopping the high-speed system clock (MOC register)

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

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 clock, and internal high-speed oscillation clock or high-speed system clock as 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]

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

When RSTOP is cleared to 0, the internal high-speed oscillation clock starts operating.

<2> Waiting for the oscillation accuracy stabilization time of internal high-speed oscillation clock (RCM register)

Wait until RSTS is set to 1[Note 2].

Notes 1. After a reset release, the internal high-speed oscillator automatically starts oscillating and the internal high-speed oscillation clock is selected as the CPU clock.

2. This wait time is not necessary if high accuracy is not necessary for the CPU clock and peripheral hardware clock.

(2) Example of setting procedure when using internal high-speed oscillation clock as CPU clock, and internal high-speed oscillation clock or high-speed system clock as peripheral hardware clock

<1> • Restarting oscillation of the internal high-speed oscillation clock[Note]
(Refer to **5.6.2 (1) Example of setting procedure when restarting oscillation of the internal high-speed oscillation clock**).

• Oscillating the high-speed system clock[Note]
(This setting is required when using the high-speed system clock as the peripheral hardware clock. Refer to **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 the internal high-speed oscillation clock or high-speed system clock is already operating.

<2> Selecting the clock supplied as the main system clock and peripheral hardware clock (MCM register)
Set the main system clock and peripheral hardware clock using XSEL and MCM0.

XSEL	MCM0	Selection of Main System Clock and Clock Supplied to Peripheral Hardware	
		Main System Clock (f_{XP})	Peripheral Hardware Clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{IH})	Internal high-speed oscillation clock (f_{IH})
0	1		
1	0		High-speed system clock (f_{XH})

<3> Selecting the CPU clock division ratio (PCC register)
The main system clock is supplied to the CPU. To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
0	0	0	f_{XP}
0	0	1	$f_{XP}/2$ (default)
0	1	0	$f_{XP}/2^2$
0	1	1	$f_{XP}/2^3$
1	0	0	$f_{XP}/2^4$
Other than above			Setting prohibited

(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 to set the STOP mode
- Setting RSTOP to 1 and stopping the internal high-speed oscillation clock

(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, refer to **CHAPTER 11 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after standby release

When the CPU is operating on the X1 clock, set the value of the OSTS register before the STOP instruction is executed. To operate the CPU immediately after the STOP mode has been released, set MCM0 to 0, switch the CPU clock to the internal high-speed oscillation clock, and check that RSTS is 1.

<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 RSTOP to 1

<1> Confirming the CPU clock status (PCC and MCM registers)

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 a clock other than the internal high-speed oscillation clock.

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

<2> Stopping the internal high-speed oscillation clock (RCM register)

When RSTOP is set to 1, internal high-speed oscillation clock is stopped.

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

5.6.3 Example of controlling internal low-speed oscillation clock

The internal low-speed oscillation clock cannot be used as the CPU clock.

Only the following peripheral hardware can operate with this clock.

- Watchdog timer
- 8-bit timer H1 (if f_{IL} is selected as the count clock)

In addition, the following operation modes can be selected by the option byte.

- Internal low-speed oscillator cannot be stopped
- Internal low-speed oscillator can be stopped by software

The internal low-speed oscillator automatically starts oscillation after a reset release, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation has been enabled by the option byte.

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

<1> Setting LSRSTOP to 1 (RCM register)

When LSRSTOP is set to 1, the internal low-speed oscillation clock is stopped.

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

<1> Clearing LSRSTOP to 0 (RCM register)

When LSRSTOP is cleared to 0, the internal low-speed oscillation clock is restarted.

Caution If "Internal low-speed oscillator cannot be stopped" is selected by the option byte, oscillation of the internal low-speed oscillation clock cannot be controlled.

5.6.4 Clocks supplied to CPU and peripheral hardware

The following table shows the relation among the clocks supplied to the CPU and peripheral hardware, and setting of registers.

Table 5-4. Clocks Supplied to CPU and Peripheral Hardware, and Register Setting

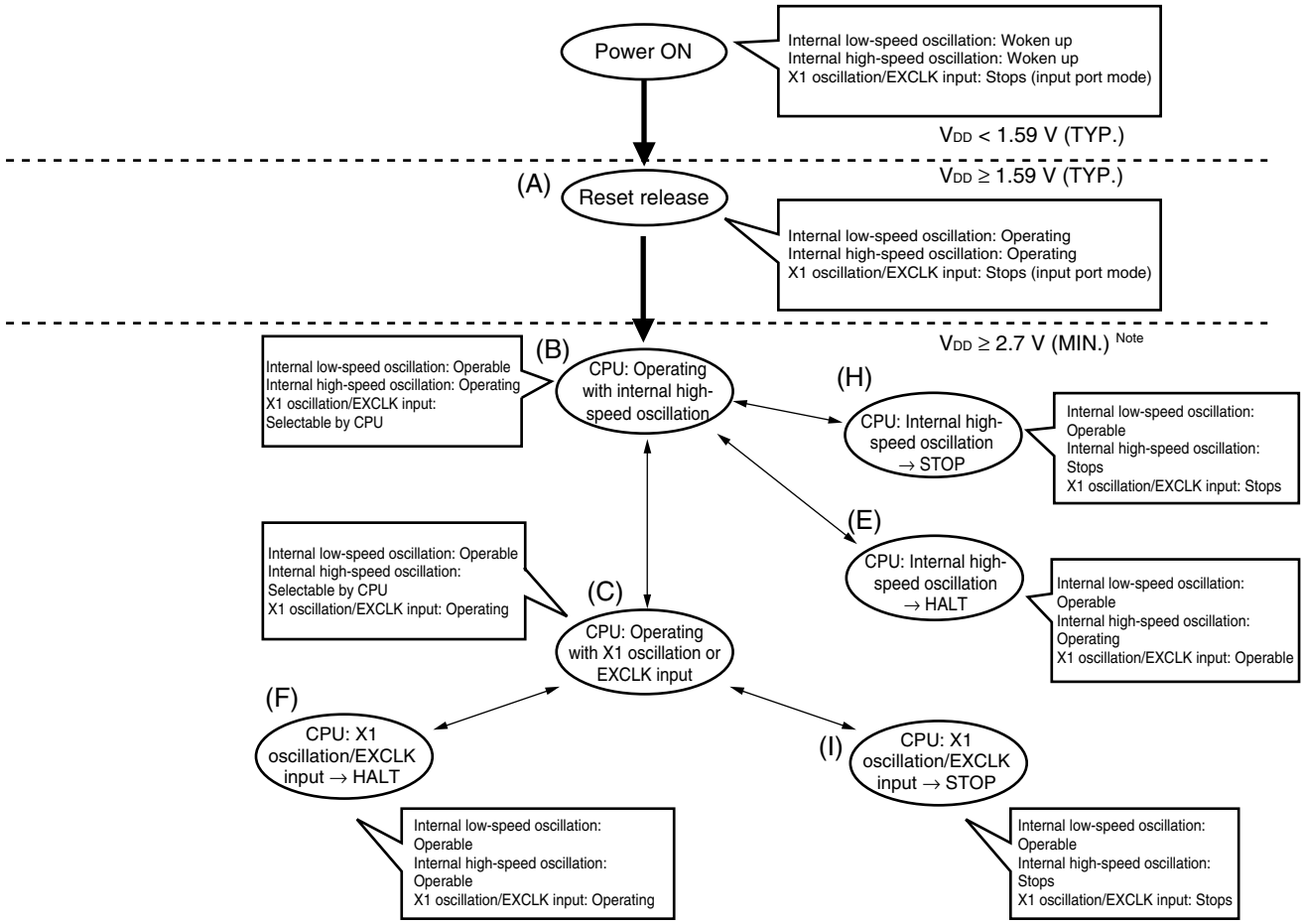
Supplied Clock		XSEL	MCM0	EXCLK
Clock Supplied to CPU	Clock Supplied to Peripheral Hardware			
Internal high-speed oscillation clock		0	×	×
Internal high-speed oscillation clock	X1 clock	1	0	0
	External main system clock	1	0	1
X1 clock		1	1	0
External main system clock		1	1	1

Remark XSEL: Bit 2 of the main clock mode register (MCM)
MCM0: Bit 0 of MCM
EXCLK: Bit 7 of the clock operation mode select register (OSCCTL)
×: don't care

5.6.5 CPU clock status transition diagram

Figure 5-13 shows the CPU clock status transition diagram of this product.

Figure 5-13. CPU Clock Status Transition Diagram (When LVI Default Start Mode Function Stopped Is Set (Option Byte: LVISTART = 0))



Note When LVI default start valid

Remark When LVI default start function enabled is set (option byte: LVISTART = 1), the CPU clock status changes to (A) in the above figure when the supply voltage exceeds 2.7 V (TYP.), and to (B) after reset processing (12 to 51 μs).

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 (1/2)

(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 (B) immediately after a reset release.)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0
(A) → (B) → (C) (X1 clock)	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main system clock)	1	1	0	Must not be checked	1	1

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (refer to CHAPTER 20 ELECTRICAL SPECIFICATIONS).

(3) 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	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL[Note]	MCM0
(B) → (C) (X1 clock)	0	1	0	Must be checked	1	1
(B) → (C) (external main system clock)	1	1	0	Must not be checked	1	1

Unnecessary if these registers are already set

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

Note The value of this flag can be changed only once after a reset release. This setting is not necessary if it has already been set.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (refer to CHAPTER 20 ELECTRICAL SPECIFICATIONS).

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-13.

2. EXCLK, OSCSEL: Bits 7 and 6 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)

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

(4) 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	RSTOP	RSTS	MCM0
(C) → (B)	0	Confirm this flag is 1.	0

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

(5) • 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)

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

(6) • STOP mode (H) set while CPU is operating with internal high-speed oscillation clock (B)
• STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

Status Transition	Setting	
(B) → (H) (C) → (I)	Stopping peripheral functions that cannot operate in STOP mode	Executing STOP instruction

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-13.

2. MCM0: Bit 0 of the main clock mode register (MCM)
RSTS, RSTOP: Bits 7 and 0 of the internal oscillation mode register (RCM)

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

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Internal high-speed oscillation clock	X1 clock	Stabilization of X1 oscillation • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • After elapse of oscillation stabilization time	Internal high-speed oscillator can be stopped (RSTOP = 1).
	External main system clock	Enabling input of external clock from EXCLK pin • MSTOP = 0, OSCSEL = 1, EXCLK = 1	Internal high-speed oscillator can be stopped (RSTOP = 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			External main system clock input can be disabled (MSTOP = 1).

5.6.7 Time required for switchover of main system clock

By setting bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC), the division ratio of the main system clock can be changed.

The actual switchover operation is not performed immediately after rewriting to PCC; operation continues on the pre-switchover clock for several clocks (refer to **Table 5-7**).

Table 5-7. Time Required for Switchover of Main System Clock Cycle Division Factor

Set Value Before Switchover			Set Value After Switchover														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0				16 clocks			16 clocks			16 clocks			16 clocks		
0	0	1	8 clocks						8 clocks			8 clocks			8 clocks		
0	1	0	4 clocks			4 clocks						4 clocks			4 clocks		
0	1	1	2 clocks			2 clocks			2 clocks						2 clocks		
1	0	0	1 clock			1 clock			1 clock			1 clock					

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

By setting bit 0 (MCM0) of the main clock mode register (MCM), the main system clock can be switched (between the internal high-speed oscillation clock and the high-speed system clock).

The actual switchover operation is not performed immediately after rewriting to MCM0; operation continues on the pre-switchover clock for several clocks (refer to **Table 5-8**).

Whether the CPU is operating on the internal high-speed oscillation clock or the high-speed system clock can be ascertained using bit 1 (MCS) of MCM.

Table 5-8. Maximum Time Required for Main System Clock Switchover

Set Value Before Switchover	Set Value After Switchover	
MCM0	MCM0	
	0	1
0		$1 + 2f_{IH}/f_{XH}$ clock
1	$1 + 2f_{XH}/f_{IH}$ clock	

Caution When switching the internal high-speed oscillation clock to the high-speed system clock, bit 2 (XSEL) of MCM must be set to 1 in advance. The value of XSEL can be changed only once after a reset release.

Remarks 1. The number of clocks listed in Table 5-8 is the number of main system clocks before switchover.

2. Calculate the number of clocks in Table 5-8 by removing the decimal portion.

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

$1 + 2f_{IH}/f_{XH} = 1 + 2 \times 8/10 = 1 + 2 \times 0.8 = 1 + 1.6 = 2.6 \rightarrow 2$ 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 5-9. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
Internal high-speed oscillation clock	MCS = 1 (The CPU is operating on the high-speed system clock)	RSTOP = 1
X1 clock	MCS = 0 (The CPU is operating on the internal high-speed oscillation clock)	MSTOP = 1
External main system clock		

5.6.9 Peripheral hardware and source clocks

The following lists peripheral hardware and source clocks incorporated in the μPD79F7025, 79F7026 microcontrollers.

Table 5-10. Peripheral Hardware and Source Clocks

Peripheral Hardware \ Source Clock		Peripheral Hardware Clock (f_{PRS})	Internal Low-Speed Oscillation Clock (f_{IL})	External Clock from Peripheral Hardware Pins
16-bit timer/event counter 00		Y	N	Y (TI000 pin)[Note]
8-bit timer/event counter	51	Y	N	Y (TI51 pin)[Note]
8-bit timer	H0	Y	N	N
	H1	Y	Y	N
Watchdog timer		N	Y	N

Note Do not start the peripheral hardware operation with the external clock from peripheral hardware pins when in the STOP mode.

Remark Y: Can be selected, N: Cannot be selected

CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00

6.1 Functions of 16-Bit Timer/Event Counter 00

16-bit timer/event counter 00 has the following functions.

(1) Interval timer

16-bit timer/event counter 00 generates an interrupt request at the preset time interval.

(2) Square-wave output

16-bit timer/event counter 00 can output a square wave with any selected frequency.

(3) External event counter

16-bit timer/event counter 00 can measure the number of pulses of an externally input signal.

(4) One-shot pulse output

16-bit timer event counter 00 can output a one-shot pulse whose output pulse width can be set freely.

(5) PPG output

16-bit timer/event counter 00 can output a rectangular wave whose frequency and output pulse width can be set freely.

(6) Pulse width measurement

16-bit timer/event counter 00 can measure the pulse width of an externally input signal.

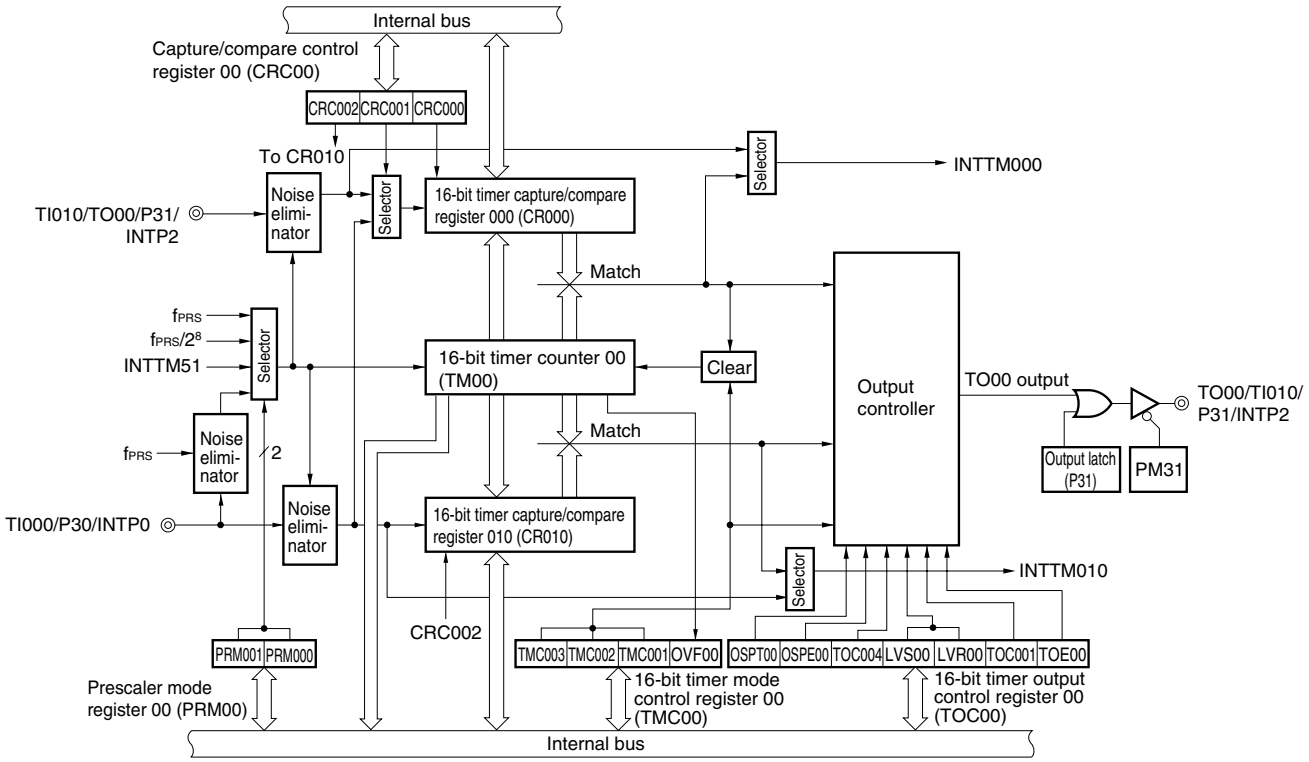
6.2 Configuration of 16-Bit Timer/Event Counter 00

16-bit timer/event counter 00 includes the following hardware.

Table 6-1. Configuration of 16-Bit Timer/Event Counter 00

Item	Configuration
Time/counter	16-bit timer counter 00 (TM00)
Register	16-bit timer capture/compare registers 000, 010 (CR000, CR010)
Timer input	TI000, TI010
Timer output	TO00, output controller
Control registers	16-bit timer mode control register 00 (TMC00) 16-bit timer capture/compare control register 00 (CRC00) 16-bit timer output control register 00 (TOC00) Prescaler mode register 00 (PRM00) Port mode register 3 (PM3) Port register 3 (P3)

Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter 00



Cautions 1. The valid edge of TI010 and timer output (TO00) cannot be used for the P34 pin at the same time. Select either of the functions.

2. If clearing of bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) to 00 and input of the capture trigger conflict, then the captured data is undefined.

3. To change the mode from the capture mode to the comparison mode, first clear the TMC003 and TMC002 bits to 00, and then change the setting.

A value that has been once captured remains stored in CR000 unless the device is reset. If the mode has been changed to the comparison mode, be sure to set a comparison value.

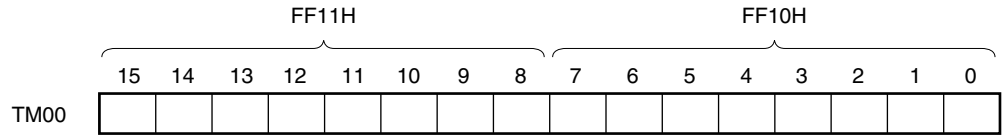
(1) 16-bit timer counter 00 (TM00)

TM00 is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 6-2. Format of 16-Bit Timer Counter 00 (TM00)

Address: FF10H, FF11H After reset: 0000H R



The count value of TM00 can be read by reading TM00 when the value of bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) is other than 00. The value of TM00 is 0000H if it is read when TMC003 and TMC002 = 00.

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

- At reset signal generation
- If TMC003 and TMC002 are cleared to 00
- If the valid edge of the TI000 pin is input in the mode in which the clear & start occurs when inputting the valid edge to the TI000 pin
- If TM00 and CR000 match in the mode in which the clear & start occurs when TM00 and CR000 match
- OSPT00 is set to 1 in one-shot pulse output mode or the valid edge is input to the TI000 pin

Caution Even if TM00 is read, the value is not captured by CR010.

(2) 16-bit timer capture/compare register 000 (CR000), 16-bit timer capture/compare register 010 (CR010)

CR000 and CR010 are 16-bit registers that are used with a capture function or comparison function selected by using CRC00.

Change the value of CR000 while the timer is stopped (TMC003 and TMC002 = 00).

The value of CR010 can be changed during operation if the value has been set in a specific way. For details, refer to **6.5.1 Rewriting CR010 during TM00 operation**.

These registers can be read or written in 16-bit units.

Reset signal generation clears these registers to 0000H.

Figure 6-3. Format of 16-Bit Timer Capture/Compare Register 000 (CR000)

Address: FF12H, FF13H After reset: 0000H R/W

	FF13H								FF12H							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR000																

(i) When CR000 is used as a compare register

The value set in CR000 is constantly compared with the TM00 count value, and an interrupt request signal (INTTM000) is generated if they match. The value is held until CR000 is rewritten.

Caution CR000 does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.

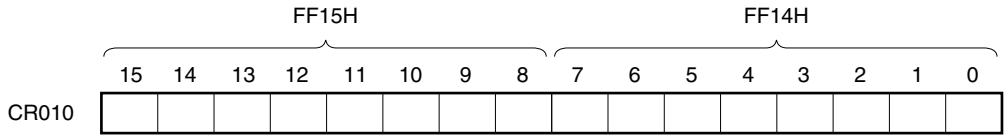
(ii) When CR000 is used as a capture register

The count value of TM00 is captured to CR000 when a capture trigger is input.

As the capture trigger, an edge of a phase reverse to that of the TI000 pin or the valid edge of the TI010 pin can be selected by using CRC00 or PRM00.

Figure 6-4. Format of 16-Bit Timer Capture/Compare Register 010 (CR010)

Address: FF14H, FF15H After reset: 0000H R/W



(i) When CR010 is used as a compare register

The value set in CR010 is constantly compared with the TM00 count value, and an interrupt request signal (INTTM010) is generated if they match.

Caution CR010 does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.

(ii) When CR010 is used as a capture register

The count value of TM00 is captured to CR010 when a capture trigger is input.

It is possible to select the valid edge of the TI000 pin as the capture trigger. The TI000 pin valid edge is set by PRM00.

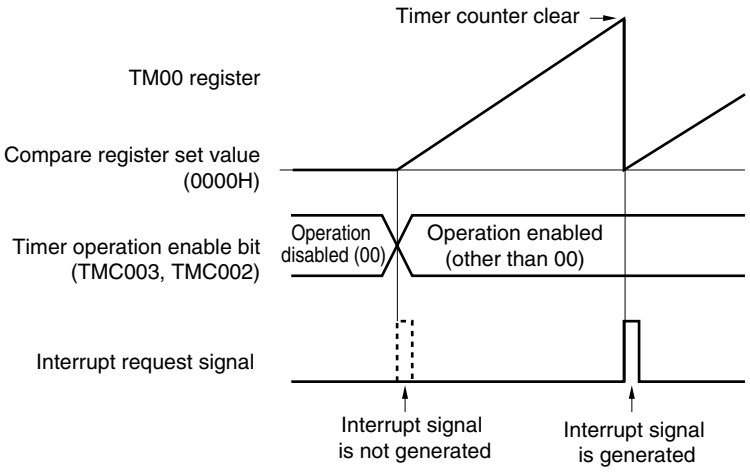
(iii) Setting range when CR000 or CR010 is used as a compare register

When CR000 or CR010 is used as a compare register, set it as shown below.

Operation	CR000 Register Setting Range	CR010 Register Setting Range
Operation as interval timer	0000H < N ≤ FFFFH	0000H[Note] ≤ M ≤ FFFFH Normally, this setting is not used. Mask the match interrupt signal (INTTM010).
Operation as square-wave output		
Operation as external event counter		
Operation in the clear & start mode entered by TI000 pin valid edge input	0000H[Note] ≤ N ≤ FFFFH	0000H[Note] ≤ M ≤ FFFFH
Operation as free-running timer		
Operation as PPG output	M < N ≤ FFFFH	0000H[Note] ≤ M < N
Operation as one-shot pulse output	0000H[Note] ≤ N ≤ FFFFH (N ≠ M)	0000H[Note] ≤ M ≤ FFFFH (M ≠ N)

Note When 0000H is set, a match interrupt immediately after the timer operation does not occur and timer output is not changed, and the first match timing is as follows. A match interrupt occurs at the timing when the timer counter (TM00 register) is changed from 0000H to 0001H.

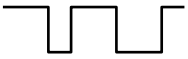
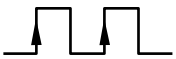
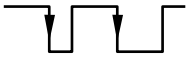
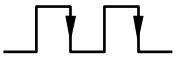
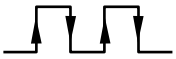
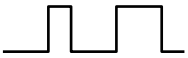
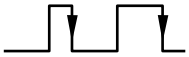
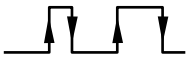
- When the timer counter is cleared due to overflow
- When the timer counter is cleared due to TI000 pin valid edge (when clear & start mode is entered by TI000 pin valid edge input)
- When the timer counter is cleared due to compare match (when clear & start mode is entered by match between TM00 and CR000 (CR000 = other than 0000H, CR010 = 0000H))



Remarks 1. N: CR000 register set value, M: CR010 register set value

2. For details of the operation enable bits (bits 3 and 2 (TMC003 and TMC002)), refer to **6.3 (1) 16-bit timer mode control register 00 (TMC00)**.

Table 6-2. Capture Operation of CR000 and CR010

External Input Signal / Capture Operation	TI000 Pin Input		TI010 Pin Input	
Capture operation of CR000	CRC001 = 1 TI000 pin input (reverse phase) 	Set values of ES010 and ES000 Position of edge to be captured	CRC001 bit = 0 TI010 pin input 	Set values of ES110 and ES100 Position of edge to be captured
		01: Rising 		01: Rising 
		00: Falling 		00: Falling 
		11: Both edges (cannot be captured)		11: Both edges 
	Interrupt signal	INTTM000 signal is not generated even if value is captured.	Interrupt signal	INTTM000 signal is generated each time value is captured.
Capture operation of CR010	TI000 pin input[Note] 	Set values of ES010 and ES000 Position of edge to be captured		
		01: Rising 		
		00: Falling 		
		11: Both edges 		
	Interrupt signal	INTTM010 signal is generated each time value is captured.		

Note The capture operation of CR010 is not affected by the setting of the CRC001 bit.

Caution To capture the count value of the TM00 register to the CR000 register by using the phase reverse to that input to the TI000 pin, the interrupt request signal (INTTM000) is not generated after the value has been captured. If the valid edge is detected on the TI010 pin during this operation, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal. To not use the external interrupt, mask the INTTM000 signal.

Remark CRC001: Refer to **6.3 (2) Capture/compare control register 00 (CRC00)**.
ES110, ES100, ES010, ES000: Refer to **6.3 (4) Prescaler mode register 00 (PRM00)**.

6.3 Registers Controlling 16-Bit Timer/Event Counter 00

Registers used to control 16-bit timer/event counter 00 are shown below.

- 16-bit timer mode control register 00 (TMC00)
- Capture/compare control register 00 (CRC00)
- 16-bit timer output control register 00 (TOC00)
- Prescaler mode register 00 (PRM00)
- Port mode register 3 (PM3)
- Port register 3 (P3)

(1) 16-bit timer mode control register 00 (TMC00)

TMC00 is an 8-bit register that sets the 16-bit timer/event counter 00 operation mode, TM00 clear mode, and output timing, and detects an overflow.

Rewriting TMC00 is prohibited during operation (when TMC003 and TMC002 = other than 00). However, it can be changed when TMC003 and TMC002 are cleared to 00 (stopping operation) and when OVF00 is cleared to 0.

TMC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TMC00 to 00H.

Caution 16-bit timer/event counter 00 starts operation at the moment TMC003 and TMC002 are set to values other than 00 (operation stop mode), respectively. Set TMC003 and TMC002 to 00 to stop the operation.

Figure 6-5. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address: FF86H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	Operation enable of 16-bit timer/event counter 00
0	0	Disables 16-bit timer/event counter 00 operation. Stops supplying operating clock. Clears 16-bit timer counter 00 (TM00).
0	1	Free-running timer mode
1	0	Clear & start mode entered by TI000 pin valid edge input[Note]
1	1	Clear & start mode entered upon a match between TM00 and CR000

TMC001	Condition to reverse timer output (TO00)
0	• Match between TM00 and CR000 or match between TM00 and CR010
1	• Match between TM00 and CR000 or match between TM00 and CR010 • Trigger input of TI000 pin valid edge

OVF00	TM00 overflow flag
Clear (0)	Clears OVF00 to 0 or TMC003 and TMC002 = 00
Set (1)	Overflow occurs.
OVF00 is set to 1 when the value of TM00 changes from FFFFH to 0000H in all the operation modes (free-running timer mode, clear & start mode entered by TI000 pin valid edge input, and clear & start mode entered upon a match between TM00 and CR000). It can also be set to 1 by writing 1 to OVF00.	

Note The TI000 pin valid edge is set by bits 5 and 4 (ES010, ES000) of prescaler mode register 00 (PRM00).

(2) Capture/compare control register 00 (CRC00)

CRC00 is the register that controls the operation of CR000 and CR010.

Changing the value of CRC00 is prohibited during operation (when TMC003 and TMC002 = other than 00).

CRC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CRC00 to 00H.

Figure 6-6. Format of Capture/Compare Control Register 00 (CRC00)

Address: FF88H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	CR010 operating mode selection
0	Operates as compare register
1	Operates as capture register

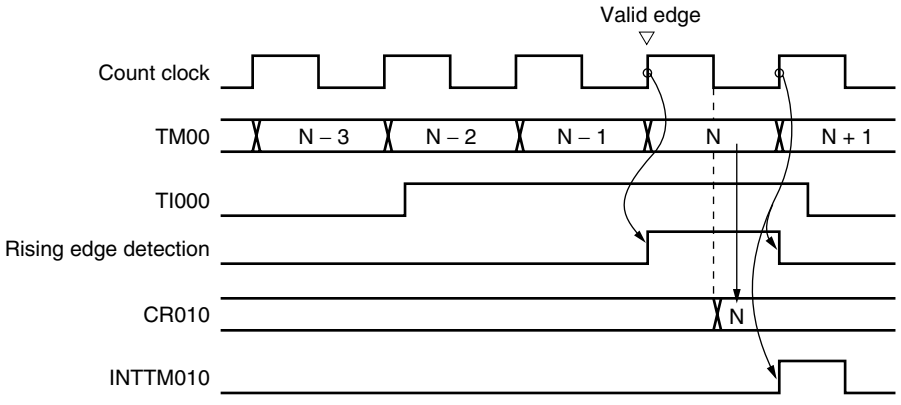
CRC001	CR000 capture trigger selection
0	Captures on valid edge of TI010 pin
1	Captures on valid edge of TI000 pin by reverse phase[Note]
The valid edge of the TI010 and TI000 pin is set by PRM00. If ES010 and ES000 are set to 11 (both edges) when CRC001 is 1, the valid edge of the TI000 pin cannot be detected.	

CRC000	CR000 operating mode selection
0	Operates as compare register
1	Operates as capture register
If TMC003 and TMC002 are set to 11 (clear & start mode entered upon a match between TM00 and CR000), be sure to set CRC000 to 0.	

Note When the valid edge is detected from the TI010 pin, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal.

Caution To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 00 (PRM00).

Figure 6-7. Example of CR010 Capture Operation (When Rising Edge Is Specified)



(3) 16-bit timer output control register 00 (TOC00)

TOC00 is an 8-bit register that controls the TO00 output.

TOC00 can be rewritten while only OSPT00 is operating (when TMC003 and TMC002 = other than 00). Rewriting the other bits is prohibited during operation.

However, TOC004 can be rewritten during timer operation as a means to rewrite CR010 (refer to **6.5.1 Rewriting CR010 during TM00 operation**).

TOC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TOC00 to 00H.

Caution Be sure to set TOC00 using the following procedure.

<1> Set TOC004 and TOC001 to 1.

<2> Set only TOE00 to 1.

<3> Set either of LVS00 or LVR00 to 1.

Figure 6-8. Format of 16-Bit Timer Output Control Register 00 (TOC00)

Address: FF89H After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00

OSPT00	One-shot pulse output trigger via software
0	–
1	One-shot pulse output
The value of this bit is always “0” when it is read. Do not set this bit to 1 in a mode other than the one-shot pulse output mode. If it is set to 1, TM00 is cleared and started.	

OSPE00	One-shot pulse output operation control
0	Successive pulse output
1	One-shot pulse output
One-shot pulse output operates correctly in the free-running timer mode or clear & start mode entered by TI000 pin valid edge input. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM00 and CR000.	

TOC004	TO00 output control on match between CR010 and TM00
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM010) is generated even when TOC004 = 0.	

LVS00	LVR00	Setting of TO00 output status
0	0	No change
0	1	Initial value of TO00 output is low level (TO00 output is cleared to 0).
1	0	Initial value of TO00 output is high level (TO00 output is set to 1).
1	1	Setting prohibited

- LVS00 and LVR00 can be used to set the initial value of the TO00 output level. If the initial value does not have to be set, leave LVS00 and LVR00 as 00.
- Be sure to set LVS00 and LVR00 when TOE00 = 1.
  LVS00, LVR00, and TOE00 being simultaneously set to 1 is prohibited.
- LVS00 and LVR00 are trigger bits. By setting these bits to 1, the initial value of the TO00 output level can be set. Even if these bits are cleared to 0, TO00 output is not affected.
- The values of LVS00 and LVR00 are always 0 when they are read.
- For how to set LVS00 and LVR00, refer to **6.5.2 Setting LVS00 and LVR00**.
- The actual TO00/TI010/P01 pin output is determined depending on PM01 and P01, besides TO00 output.

TOC001	TO00 output control on match between CR000 and TM00
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM000) is generated even when TOC001 = 0.	

TOE00	TO00 output control
0	Disables output (TO00 output fixed to low level)
1	Enables output

(4) Prescaler mode register 00 (PRM00)

PRM00 is the register that sets the TM00 count clock and TI000 and TI010 pin input valid edges.

Rewriting PRM00 is prohibited during operation (when TMC003 and TMC002 = other than 00).

PRM00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears PRM00 to 00H.

Cautions 1. Do not apply the following setting when setting the PRM001 and PRM000 bits to 11 (to specify the valid edge of the TI000 pin as a count clock).

- **Clear & start mode entered by the TI000 pin valid edge**
- **Setting the TI000 pin as a capture trigger**

2. If the operation of the 16-bit timer/event counter 00 is enabled when the TI000 or TI010 pin is at high level and when the valid edge of the TI000 or TI010 pin is specified to be the rising edge or both edges, the high level of the TI000 or TI010 pin is detected as a rising edge. Note this when the TI000 or TI010 pin is pulled up. However, the rising edge is not detected when the timer operation has been once stopped and then is enabled again.

3. The valid edge of TI010 and timer output (TO00) cannot be used for the P34 pin at the same time. Select either of the functions.

Figure 6-9. Format of Prescaler Mode Register 00 (PRM00)

Address: FF87H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES110	ES100	ES010	ES000	0	0	PRM001	PRM000

ES110	ES100	TI010 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES010	ES000	TI000 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM001	PRM000	Count clock selection		
			f_{PRS} = 2 MHz	f_{PRS} = 5 MHz
0	0	f_{PRS}	2 MHz	5 MHz
0	1	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz
1	0	INTTM51		
1	1	TI000 valid edge Notes 1, 2		

Notes 1. The external clock from the TI000 pin requires a pulse longer than twice the cycle of the peripheral hardware clock (f_{PRS}).

2. Do not start timer operation with the external clock from the TI000 pin when in the STOP mode.

Remark f_{PRS}: Peripheral hardware clock frequency

(5) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P31/TO00/TI010/INTP2 pin for timer output, set PM31 and the output latches of P31 to 0.

When using the P30/TI000/INTP1 and P31/TI010/TO00/INTP2 pins for timer input, set PM30 and PM31 to 1. At this time, the output latches of P30 and P31 may be 0 or 1.

PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM3 to FFH.

Figure 6-10. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Operation of 16-Bit Timer/Event Counter 00

6.4.1 Interval timer operation

If bits 3 and 2 (TMC003 and TMC002) of the 16-bit timer mode control register (TMC00) are set to 11 (clear & start mode entered upon a match between TM00 and CR000), the count operation is started in synchronization with the count clock.

When the value of TM00 later matches the value of CR000, TM00 is cleared to 0000H and a match interrupt signal (INTTM000) is generated. This INTTM000 signal enables TM00 to operate as an interval timer.

Remarks 1. For the setting of I/O pins, refer to **6.3 (5) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

Figure 6-11. Block Diagram of Interval Timer Operation

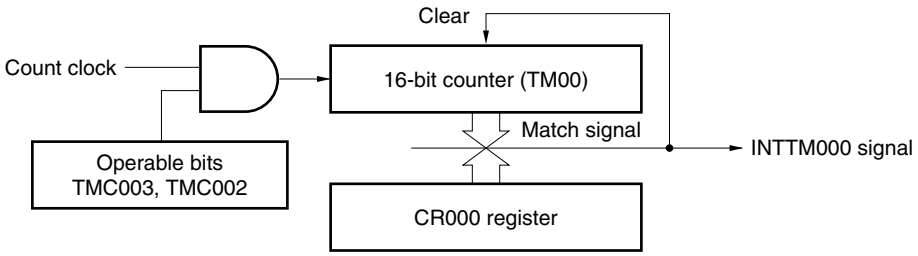


Figure 6-12. Basic Timing Example of Interval Timer Operation

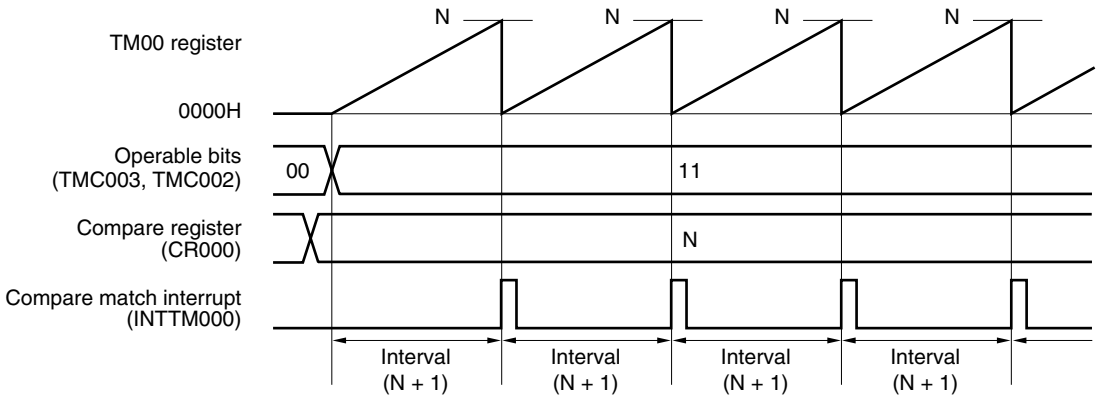


Figure 6-13. Example of Register Settings for Interval Timer Operation

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	1	1	0	0

Clears and starts on match between TM00 and CR000.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	0	0	0

CR000 used as compare register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0	0	0	0	0

(d) Prescaler mode register 00 (PRM00)

ES110	ES100	ES010	ES000	3	2	PRM001	PRM000
0	0	0	0	0	0	0/1	0/1

Selects count clock

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interval time is as follows.

- Interval time = (M + 1) × Count clock cycle

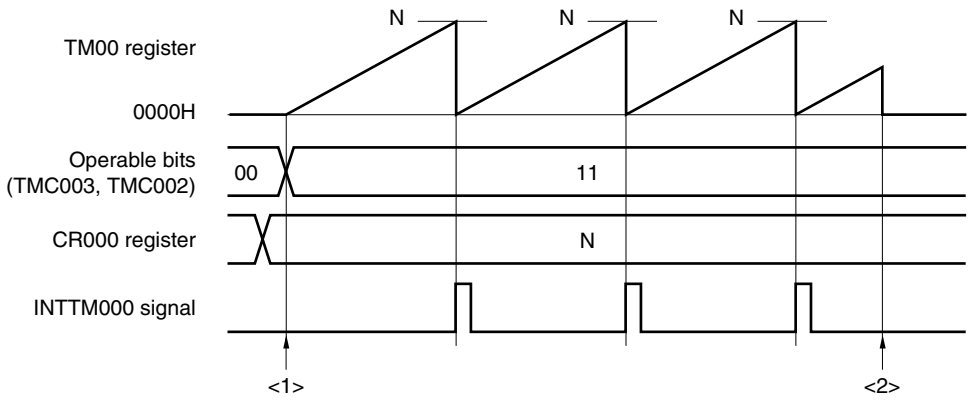
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

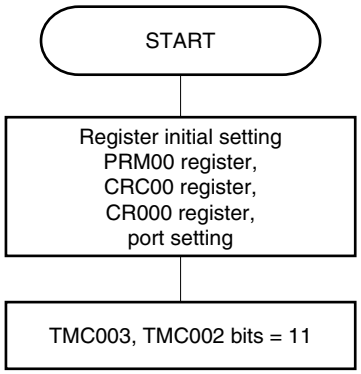
Usually, CR010 is not used for the interval timer function. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

Figure 6-14. Example of Software Processing for Interval Timer Function



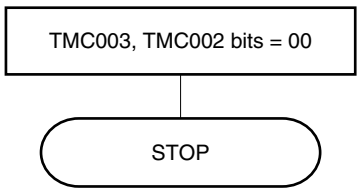
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

6.4.2 Square-wave output operation

When 16-bit timer/event counter 00 operates as an interval timer (refer to **6.4.1**), a square wave can be output from the TO00 pin by setting the 16-bit timer output control register 00 (TOC00) to 03H.

When TMC003 and TMC002 are set to 11 (count clear & start mode entered upon a match between TM00 and CR000), the counting operation is started in synchronization with the count clock.

When the value of TM00 later matches the value of CR000, TM00 is cleared to 0000H, an interrupt signal (INTTM000) is generated, and TO00 output is inverted. This TO00 output that is inverted at fixed intervals enables TO0n to output a square wave.

Remarks 1. For the setting of I/O pins, refer to **6.3 (5) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

Figure 6-15. Block Diagram of Square-Wave Output Operation

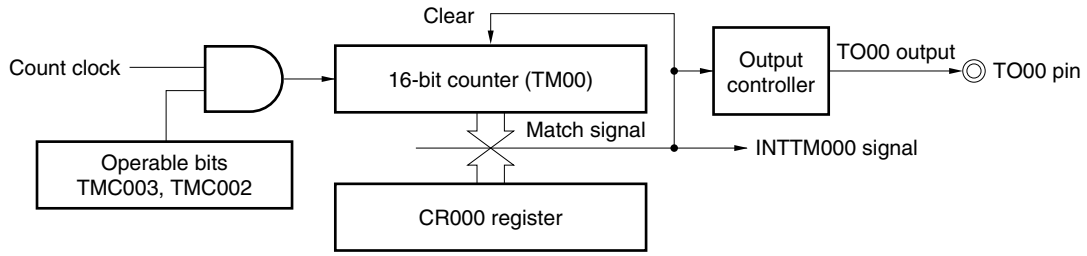


Figure 6-16. Basic Timing Example of Square-Wave Output Operation

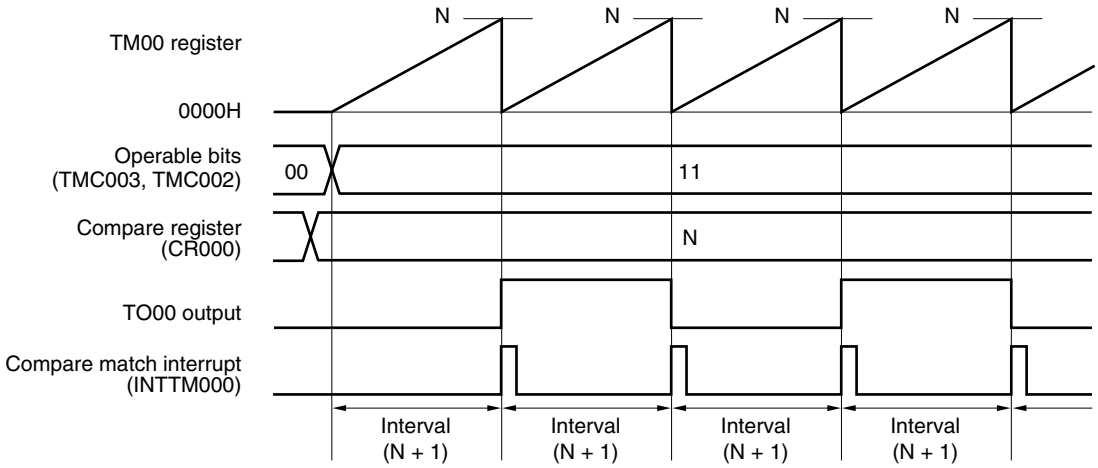
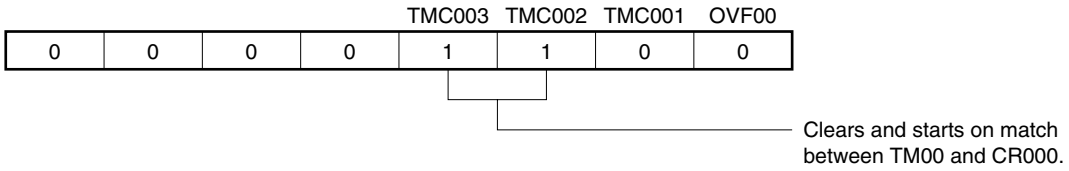
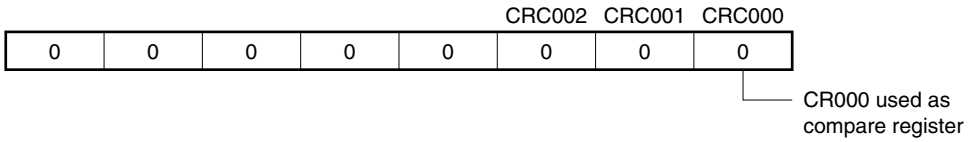


Figure 6-17. Example of Register Settings for Square-Wave Output Operation

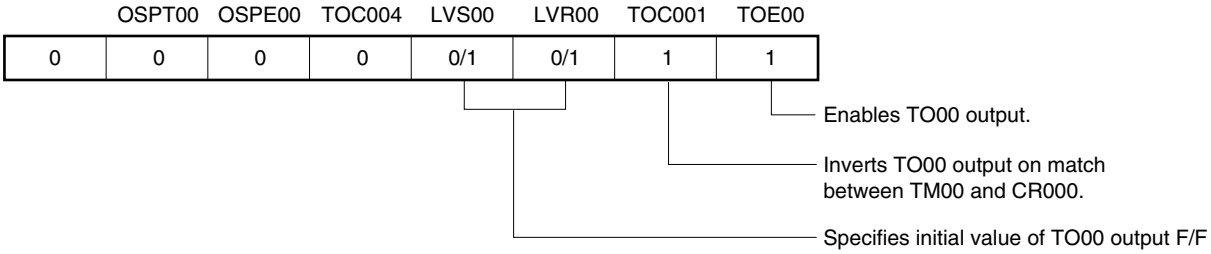
(a) 16-bit timer mode control register 00 (TMC00)



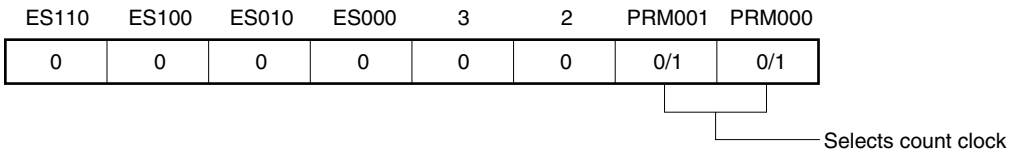
(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interval time is as follows.

- Square wave frequency = 1 / [2 × (M + 1) × Count clock cycle]

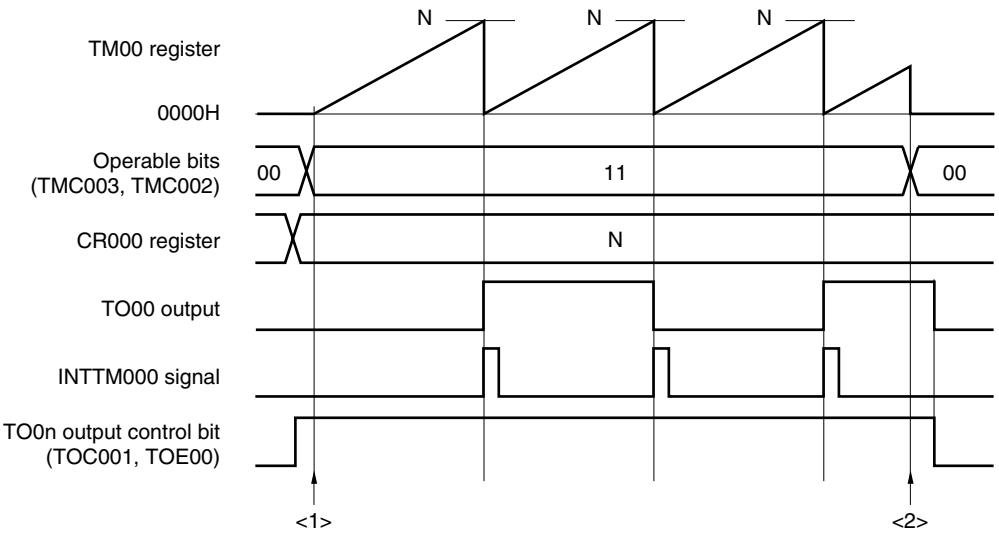
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

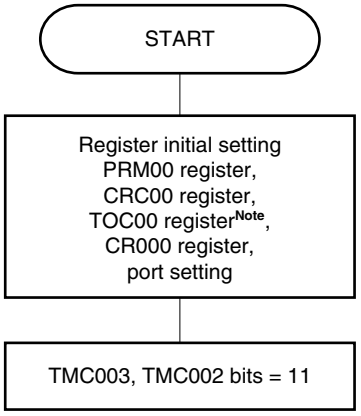
Usually, CR010 is not used for the square-wave output function. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

Figure 6-18. Example of Software Processing for Square-Wave Output Function



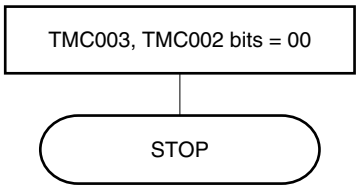
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

Note Care must be exercised when setting TOC00. For details, refer to **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.3 External event counter operation

When bits 1 and 0 (PRM001 and PRM000) of the prescaler mode register 00 (PRM00) are set to 11 (for counting up with the valid edge of the TI000 pin) and bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 11, the valid edge of an external event input is counted, and a match interrupt signal indicating matching between TM00 and CR000 (INTTM000) is generated.

To input the external event, the TI000 pin is used. Therefore, the timer/event counter cannot be used as an external event counter in the clear & start mode entered by the TI000 pin valid edge input (when TMC003 and TMC002 = 10).

The INTTM000 signal is generated with the following timing.

- Timing of generation of INTTM000 signal (second time or later)
  = Number of times of detection of valid edge of external event × (Set value of CR000 + 1)

However, the first match interrupt immediately after the timer/event counter has started operating is generated with the following timing.

- Timing of generation of INTTM000 signal (first time only)
  = Number of times of detection of valid edge of external event input × (Set value of CR000 + 2)

To detect the valid edge, the signal input to the TI000 pin is sampled during the clock cycle of f_{PRS}. The valid edge is not detected until it is detected two times in a row. Therefore, a noise with a short pulse width can be eliminated.

Remarks 1. For the setting of I/O pins, refer to **6.3 (5) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

Figure 6-19. Block Diagram of External Event Counter Operation

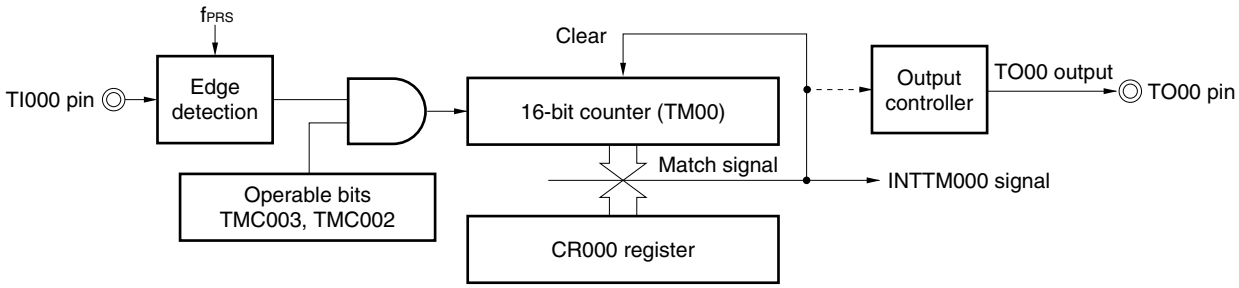
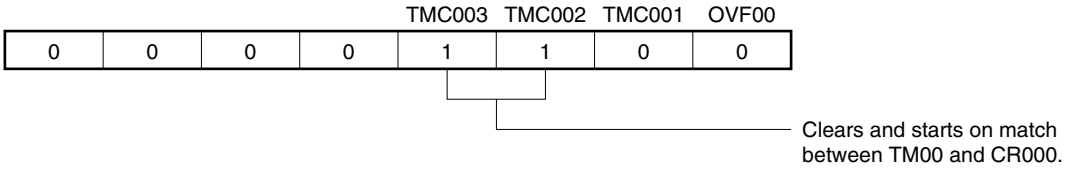
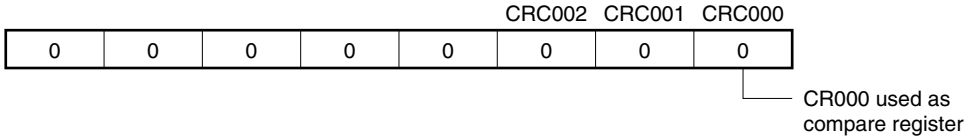


Figure 6-20. Example of Register Settings in External Event Counter Mode (1/2)

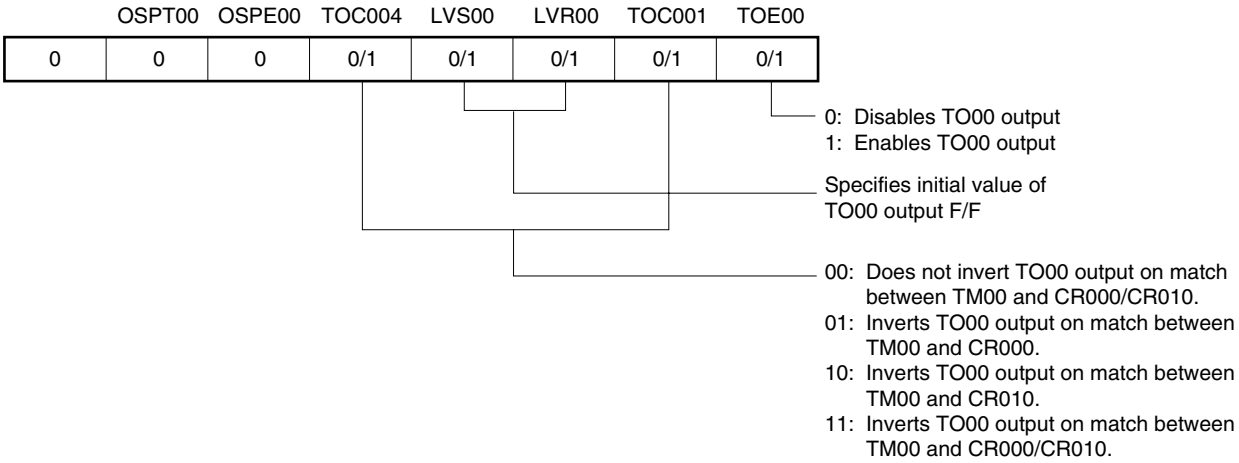
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)

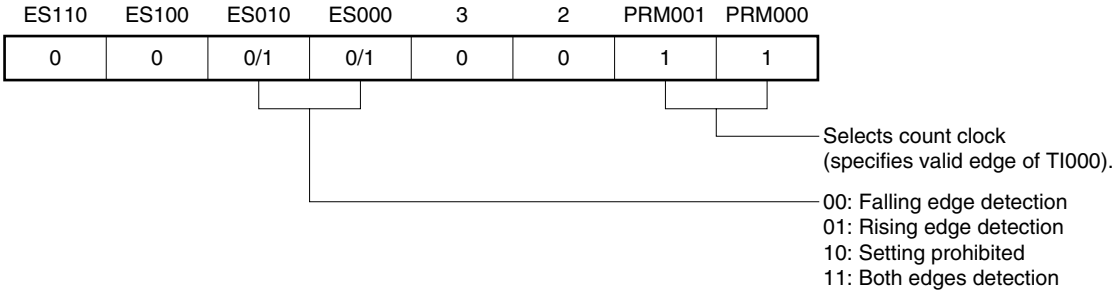


Figure 6-20. Example of Register Settings in External Event Counter Mode (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interrupt signal (INTTM000) is generated when the number of external events reaches (M + 1).

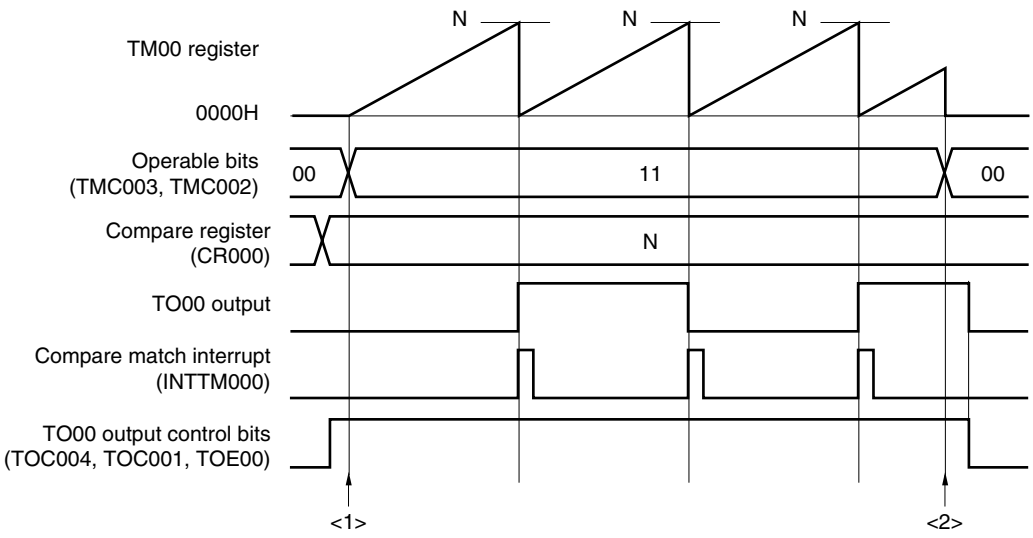
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

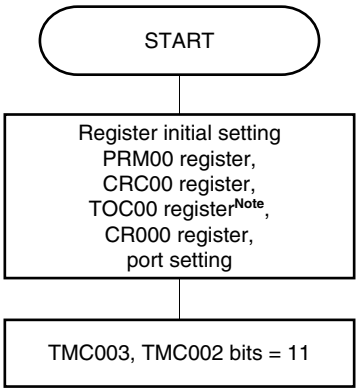
Usually, CR010 is not used in the external event counter mode. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

Figure 6-21. Example of Software Processing in External Event Counter Mode



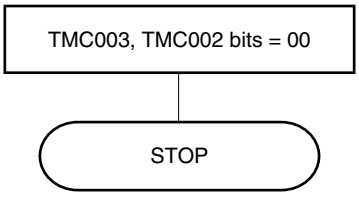
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

Note Care must be exercised when setting TOC00. For details, refer to **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.4 Operation in clear & start mode entered by TI000 pin valid edge input

When bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 10 (clear & start mode entered by the TI000 pin valid edge input) and the count clock (set by PRM00) is supplied to the timer/event counter, TM00 starts counting up. When the valid edge of the TI000 pin is detected during the counting operation, TM00 is cleared to 0000H and starts counting up again. If the valid edge of the TI000 pin is not detected, TM00 overflows and continues counting.

The valid edge of the TI000 pin is a cause to clear TM00. Starting the counter is not controlled immediately after the start of the operation.

CR000 and CR010 are used as compare registers and capture registers.

(a) When CR000 and CR010 are used as compare registers

Signals INTTM000 and INTTM010 are generated when the value of TM00 matches the value of CR000 and CR010.

(b) When CR000 and CR010 are used as capture registers

The count value of TM00 is captured to CR000 and the INTTM000 signal is generated when the valid edge is input to the TI010 pin (or when the phase reverse to that of the valid edge is input to the TI000 pin).

When the valid edge is input to the TI000 pin, the count value of TM00 is captured to CR010 and the INTTM010 signal is generated. As soon as the count value has been captured, the counter is cleared to 0000H.

Caution Do not set the count clock as the valid edge of the TI000 pin (PRM001 and PRM000 = 11). When PRM001 and PRM000 = 11, TM00 is cleared.

Remarks 1. For the setting of the I/O pins, refer to **6.3 (5) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

(1) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: compare register, CR010: compare register)

Figure 6-22. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Compare Register)

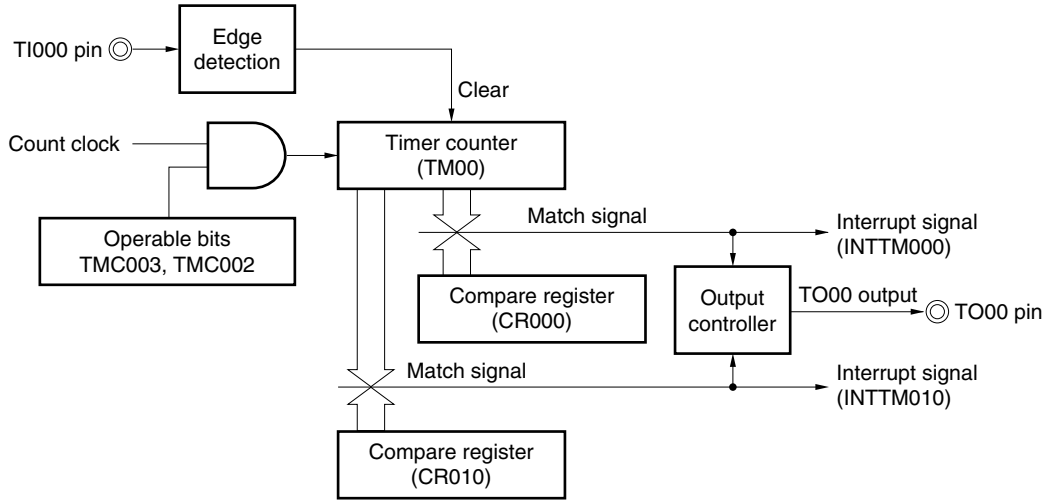
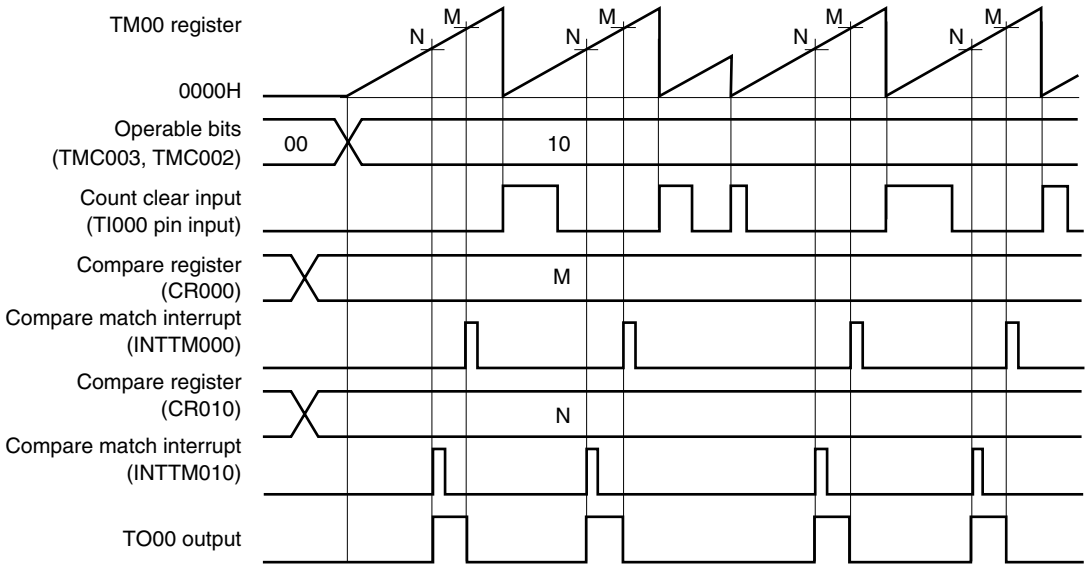
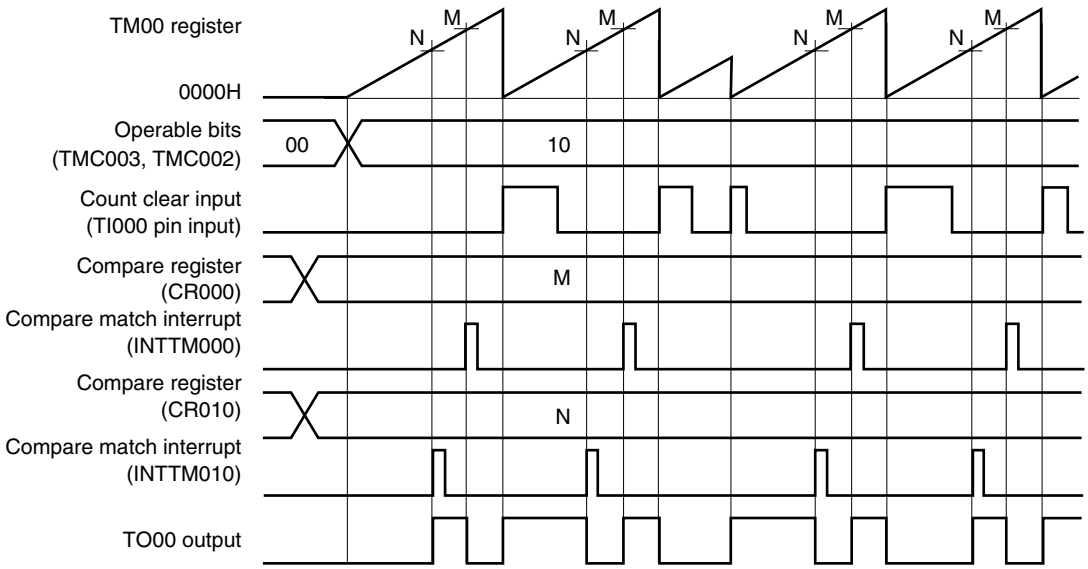


Figure 6-23. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Compare Register)

(a) TOC00 = 13H, PRM00 = 10H, CRC00 = 00H, TMC00 = 08H



(b) TOC00 = 13H, PRM00 = 10H, CRC00 = 00H, TMC00 = 0AH



(a) and (b) differ as follows depending on the setting of bit 1 (TMC001) of the 16-bit timer mode control register 00 (TMC00).

(a) The TO00 output level is inverted when TM00 matches a compare register.

(b) The TO00 output level is inverted when TM00 matches a compare register or when the valid edge of the TI000 pin is detected.

(2) Operation in clear & start mode entered by TI000 pin valid edge input
(CR000: compare register, CR010: capture register)

Figure 6-24. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register)

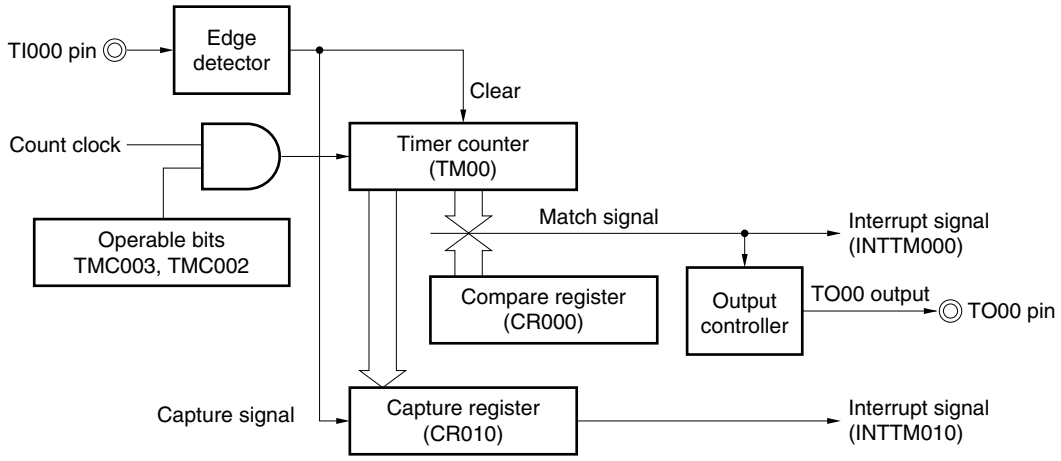
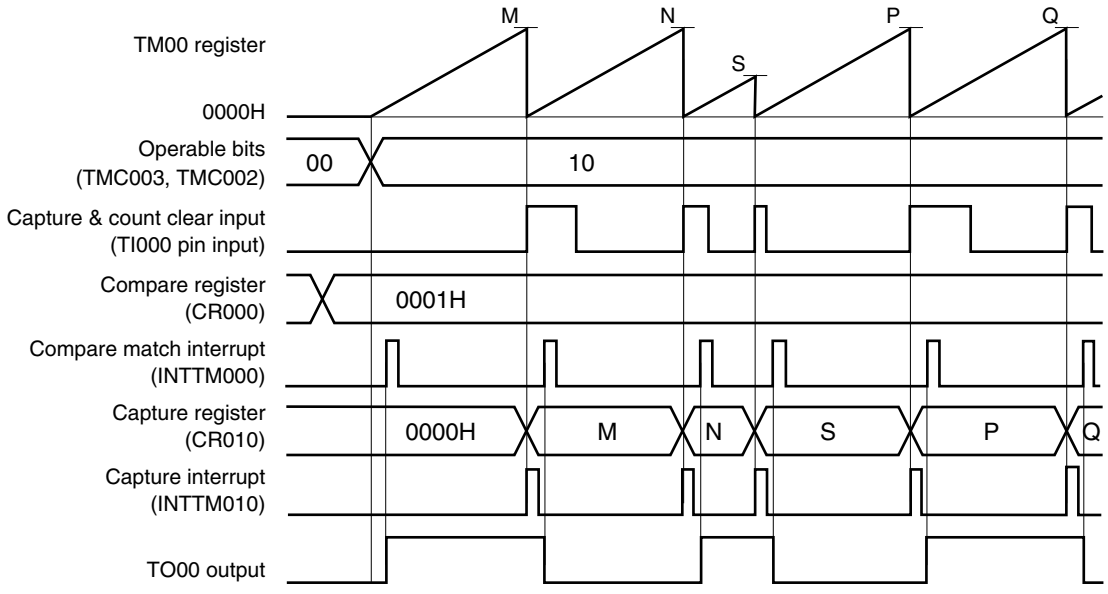


Figure 6-25. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 08H, CR000 = 0001H

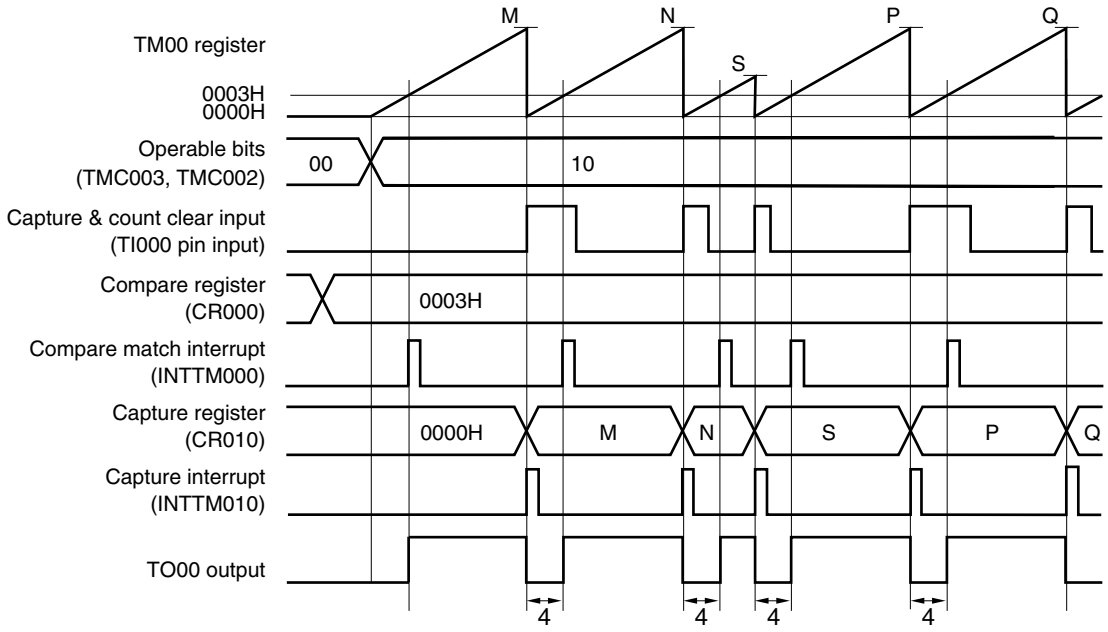


This is an application example where the TO00 output level is inverted when the count value has been captured & cleared.

The count value is captured to CR010 and TM00 is cleared (to 0000H) when the valid edge of the TI000 pin is detected. When the count value of TM00 is 0001H, a compare match interrupt signal (INTTM000) is generated, and the TO00 output level is inverted.

Figure 6-25. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register) (2/2)

(b) TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 0AH, CR000 = 0003H



This is an application example where the width set to CR000 (4 clocks in this example) is to be output from the TO00 pin when the count value has been captured & cleared.

The count value is captured to CR010, a capture interrupt signal (INTTM010) is generated, TM00 is cleared (to 0000H), and the TO00 output level is inverted when the valid edge of the TI000 pin is detected. When the count value of TM00 is 0003H (four clocks have been counted), a compare match interrupt signal (INTTM000) is generated and the TO00 output level is inverted.

(3) Operation in clear & start mode by entered TI000 pin valid edge input (CR000: capture register, CR010: compare register)

Figure 6-26. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register)

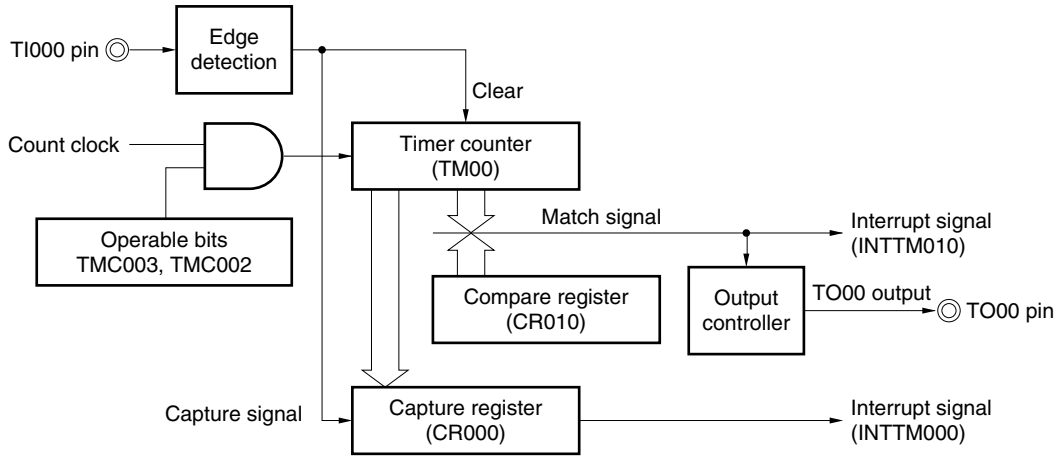
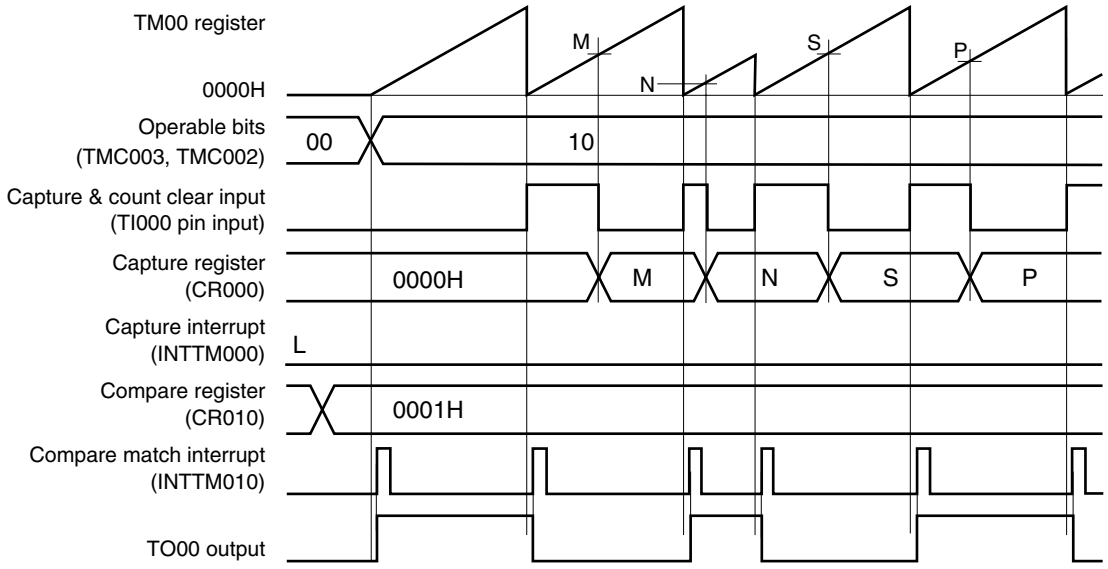


Figure 6-27. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00 = 03H, TMC00 = 08H, CR010 = 0001H



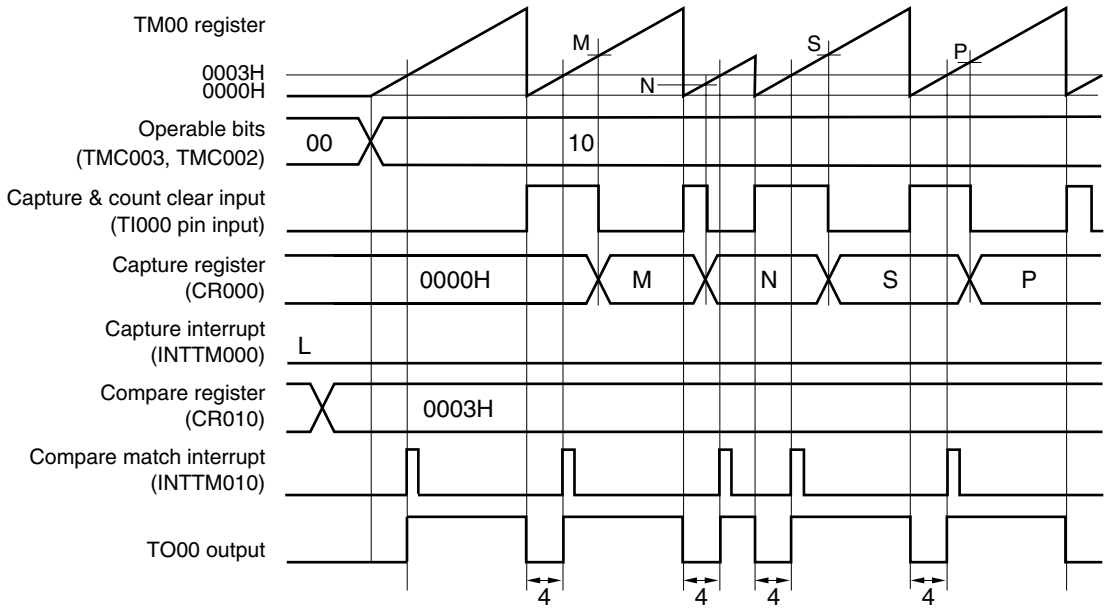
This is an application example where the TO00 output level is to be inverted when the count value has been captured & cleared.

TM00 is cleared at the rising edge detection of the TI000 pin and it is captured to CR000 at the falling edge detection of the TI000 pin.

When bit 1 (CRC001) of capture/compare control register 00 (CRC00) is set to 1, the count value of TM00 is captured to CR000 in the phase reverse to that of the signal input to the TI000 pin, but the capture interrupt signal (INTTM000) is not generated. However, the INTTM000 signal is generated when the valid edge of the TI010 pin is detected. Mask the INTTM000 signal when it is not used.

Figure 6-27. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register) (2/2)

(b) TOC00 = 13H, PRM00 = 10H, CRC00 = 03H, TMC00 = 0AH, CR010 = 0003H



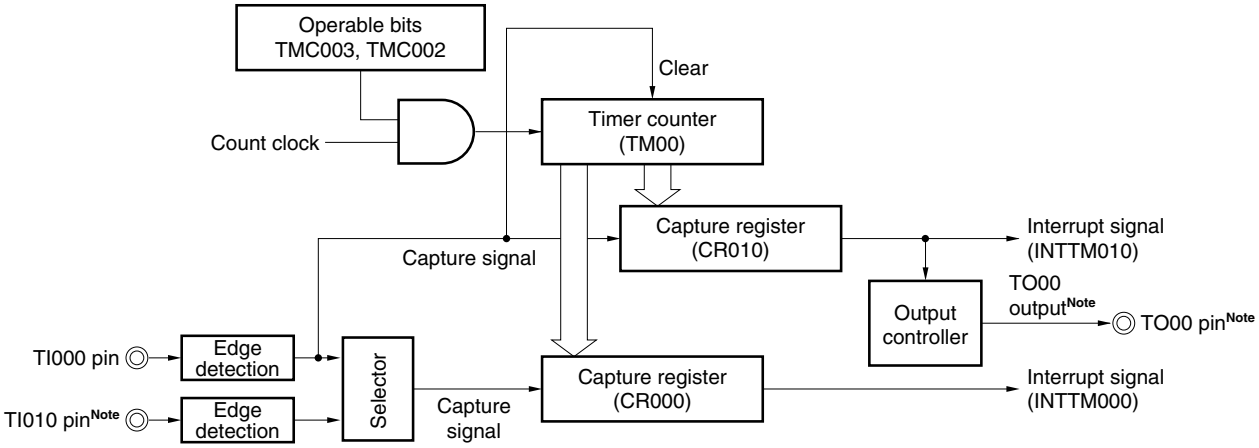
This is an application example where the width set to CR010 (4 clocks in this example) is to be output from the TO00 pin when the count value has been captured & cleared.

TM00 is cleared (to 0000H) at the rising edge detection of the TI000 pin and captured to CR000 at the falling edge detection of the TI000 pin. The TO00 output level is inverted when TM00 is cleared (to 0000H) because the rising edge of the TI000 pin has been detected or when the value of TM00 matches that of a compare register (CR010).

When bit 1 (CRC001) of capture/compare control register 00 (CRC00) is 1, the count value of TM00 is captured to CR000 in the phase reverse to that of the input signal of the TI000 pin, but the capture interrupt signal (INTTM000) is not generated. However, the INTTM000 interrupt is generated when the valid edge of the TI010 pin is detected. Mask the INTTM000 signal when it is not used.

(4) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: capture register, CR010: capture register)

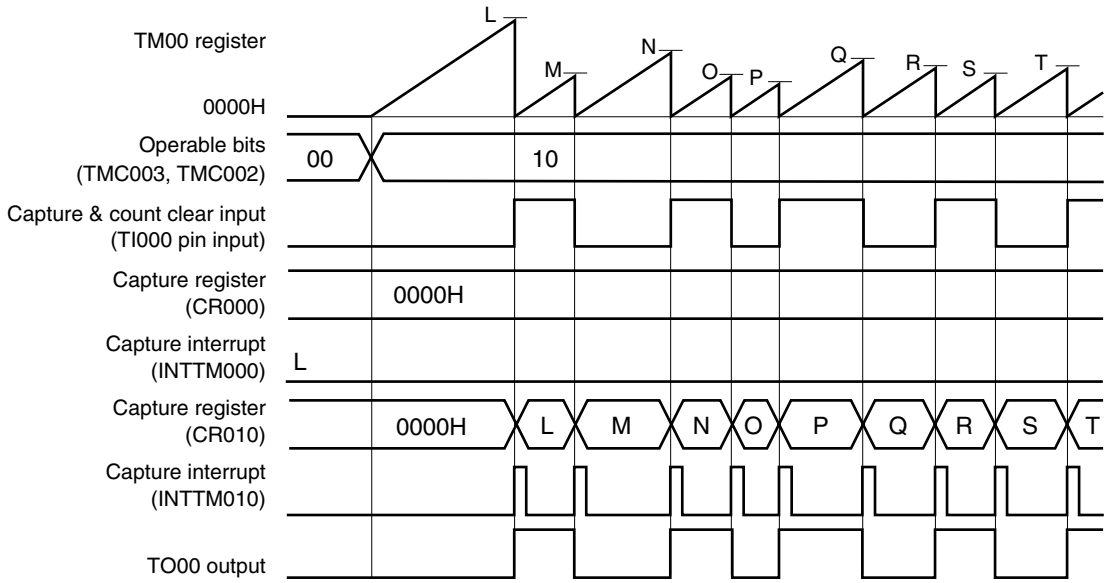
Figure 6-28. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register)



Note The timer output (TO00) cannot be used when detecting the valid edge of the TI010 pin is used.

Figure 6-29. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (1/3)

(a) TOC00 = 13H, PRM00 = 30H, CRC00 = 05H, TMC00 = 0AH

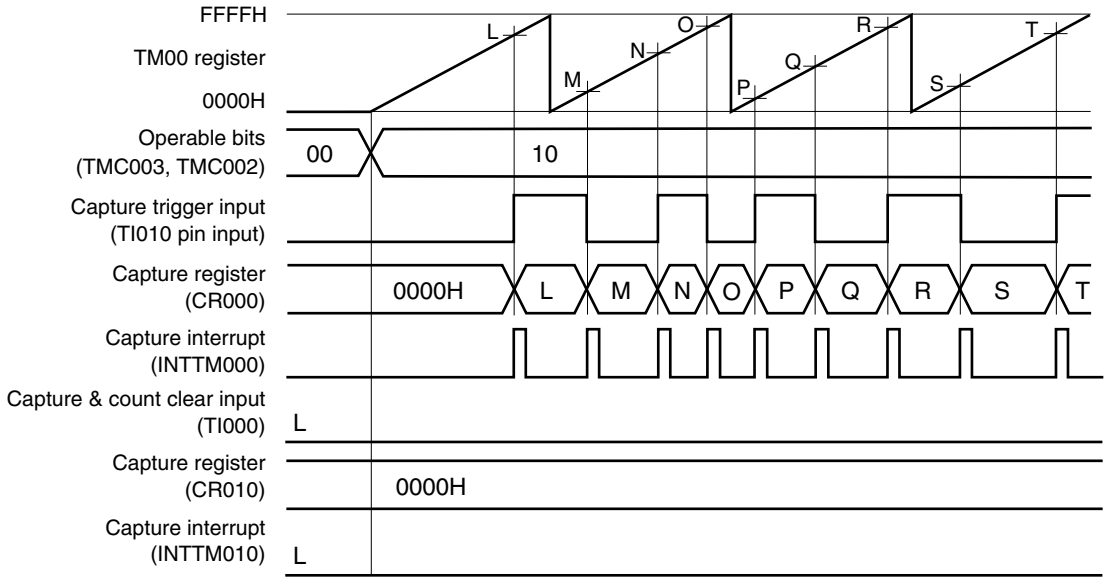


This is an application example where the count value is captured to CR010, TM00 is cleared, and the TO00 output is inverted when the rising or falling edge of the TI000 pin is detected.

When the edge of the TI010 pin is detected, an interrupt signal (INTTM000) is generated. Mask the INTTM000 signal when it is not used.

Figure 6-29. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (2/3)

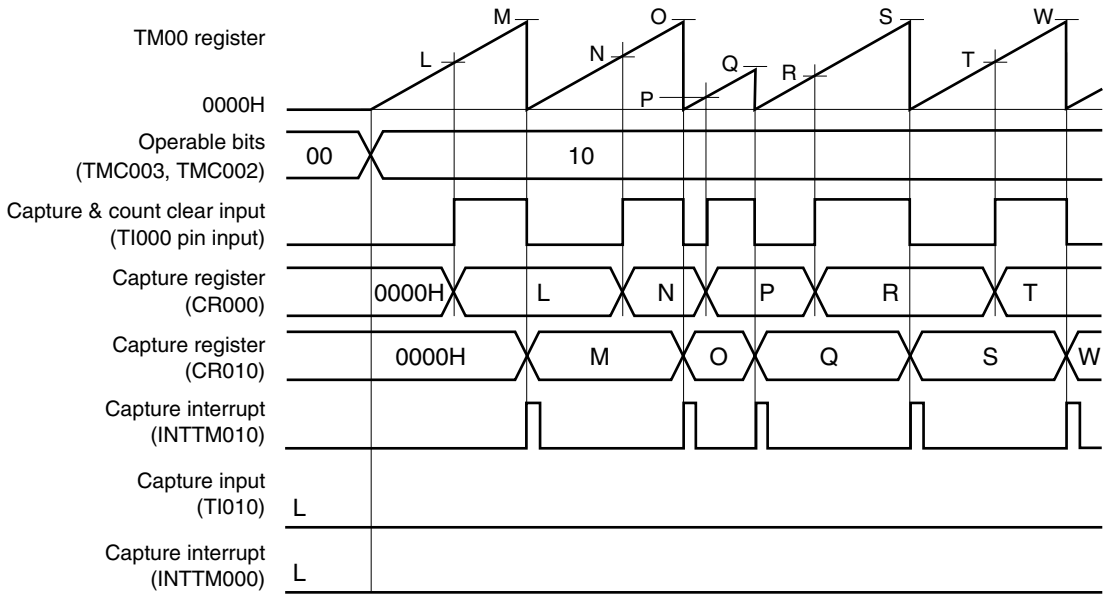
(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 0AH



This is a timing example where an edge is not input to the TI000 pin, in an application where the count value is captured to CR000 when the rising or falling edge of the TI010 pin is detected.

Figure 6-29. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (3/3)

(c) TOC00 = 13H, PRM00 = 00H, CRC00 = 07H, TMC00 = 0AH



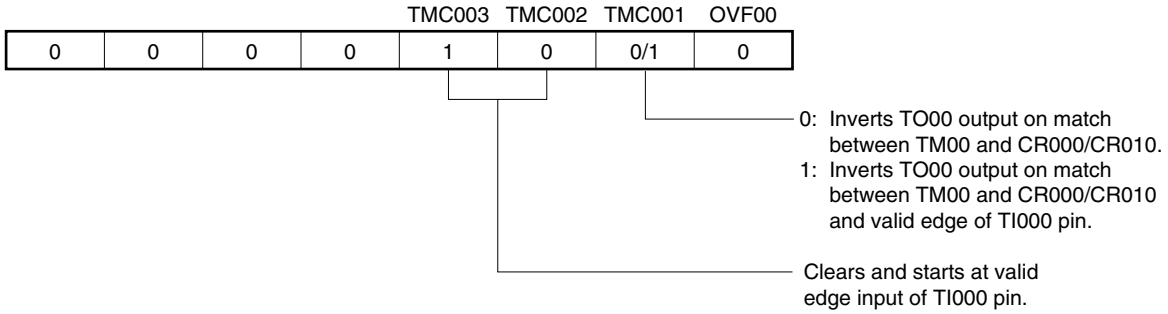
This is an application example where the pulse width of the signal input to the TI000 pin is measured.
By setting CRC00, the count value can be captured to CR000 in the phase reverse to the falling edge of the TI000 pin (i.e., rising edge) and to CR010 at the falling edge of the TI000 pin.
The high- and low-level widths of the input pulse can be calculated by the following expressions.

- High-level width = [CR010 value] – [CR000 value] × [Count clock cycle]
- Low-level width = [CR000 value] × [Count clock cycle]

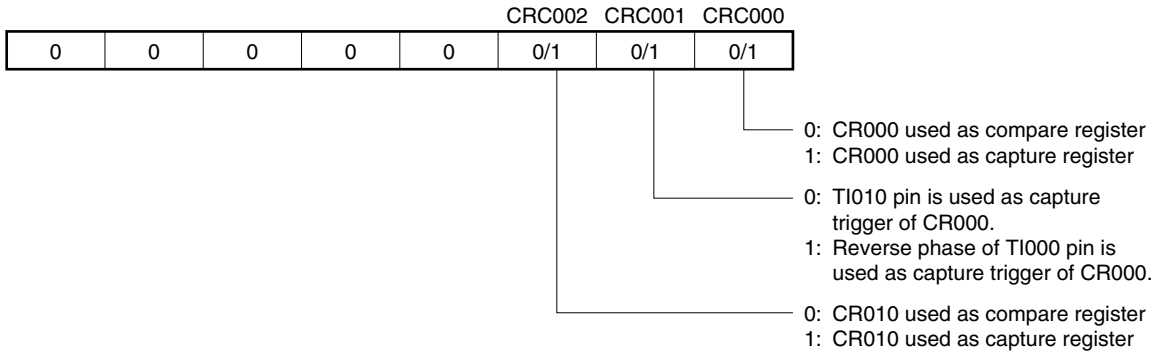
If the reverse phase of the TI000 pin is selected as a trigger to capture the count value to CR000, the INTTM000 signal is not generated. Read the values of CR000 and CR010 to measure the pulse width immediately after the INTTM010 signal is generated.
However, if the valid edge specified by bits 6 and 5 (ES110 and ES100) of prescaler mode register 00 (PRM00) is input to the TI010 pin, the count value is not captured but the INTTM000 signal is generated. To measure the pulse width of the TI000 pin, mask the INTTM000 signal when it is not used.

Figure 6-30. Example of Register Settings in Clear & Start Mode Entered by TI000 Pin Valid Edge Input (1/2)

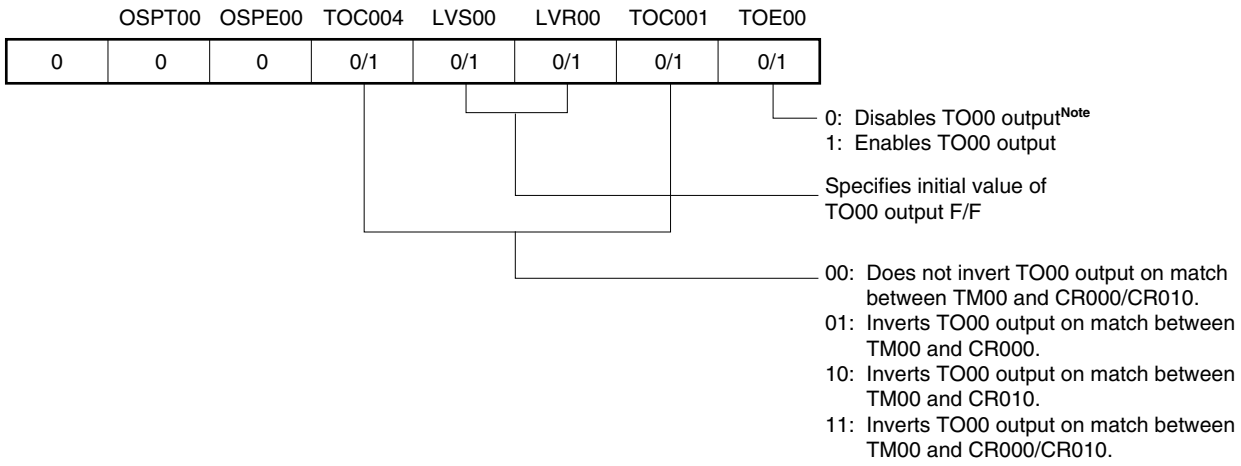
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



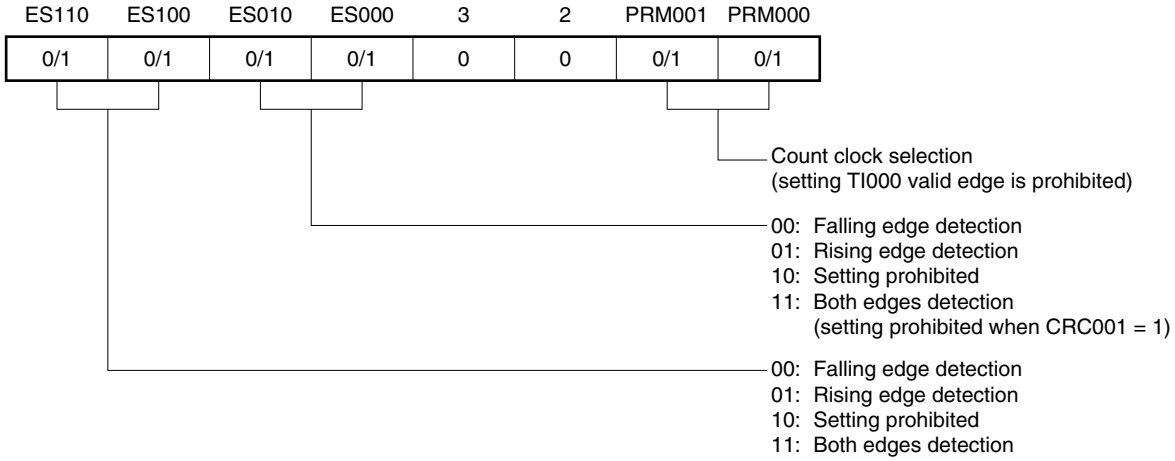
(c) 16-bit timer output control register 00 (TOC00)



Note The timer output (TO00) cannot be used when detecting the valid edge of the TI010 pin is used.

Figure 6-30. Example of Register Settings in Clear & Start Mode Entered by TI000 Pin Valid Edge Input (2/2)

(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM000) is generated. The count value of TM00 is not cleared.

To use this register as a capture register, select either the TI000 or TI010 pin [Note] input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

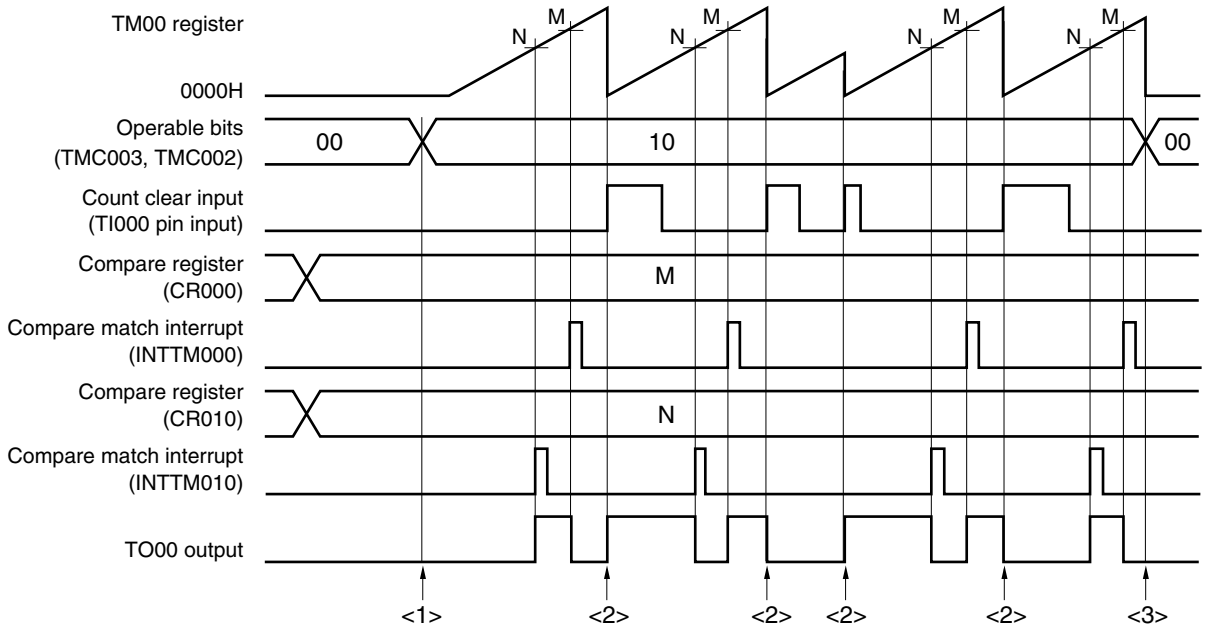
Note The timer output (TO00) cannot be used when detection of the valid edge of the TI010 pin is used.

(g) 16-bit capture/compare register 010 (CR010)

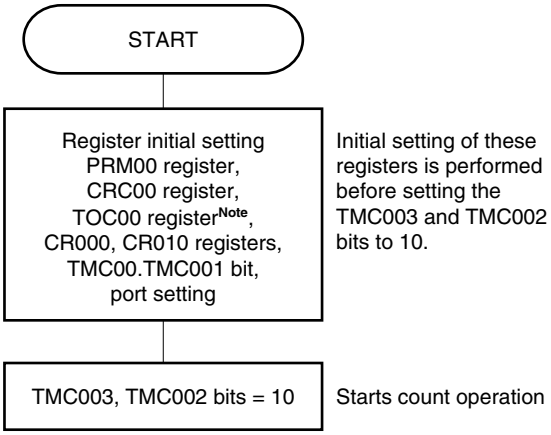
When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM010) is generated. The count value of TM00 is not cleared.

When this register is used as a capture register, the TI000 pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR010.

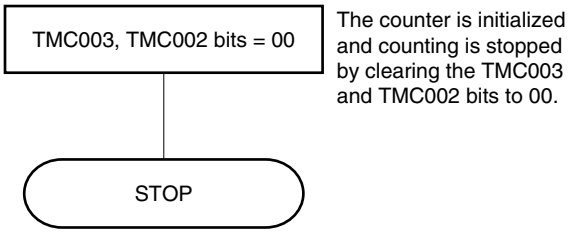
Figure 6-31. Example of Software Processing in Clear & Start Mode Entered by TI000 Pin Valid Edge Input



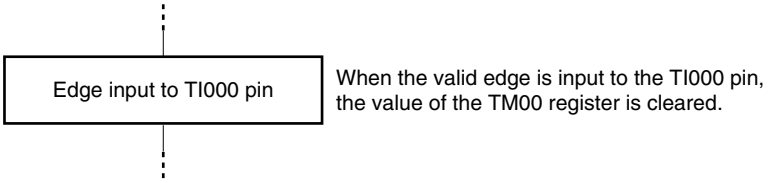
<1> Count operation start flow



<3> Count operation stop flow



<2> TM00 register clear & start flow



Note Care must be exercised when setting TOC00. For details, refer to **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.5 Free-running timer operation

When bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 01 (free-running timer mode), 16-bit timer/event counter 00 continues counting up in synchronization with the count clock. When it has counted up to FFFFH, the overflow flag (OVF00) is set to 1 at the next clock, and TM00 is cleared (to 0000H) and continues counting. Clear OVF00 to 0 by executing the CLR instruction via software.

The following three types of free-running timer operations are available.

- Both CR000 and CR010 are used as compare registers.
- One of CR000 or CR010 is used as a compare register and the other is used as a capture register.
- Both CR000 and CR010 are used as capture registers.

Remarks 1. For the setting of the I/O pins, refer to **6.3 (5) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

(1) Free-running timer mode operation

(CR000: compare register, CR010: compare register)

Figure 6-32. Block Diagram of Free-Running Timer Mode
(CR000: Compare Register, CR010: Compare Register)

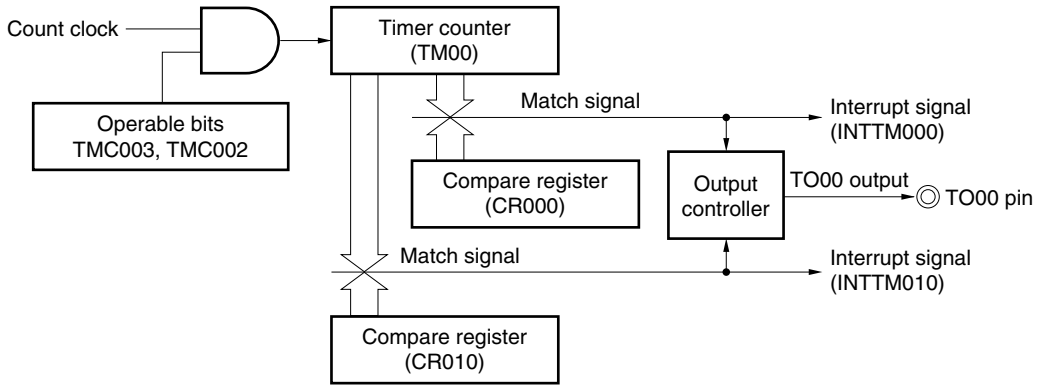
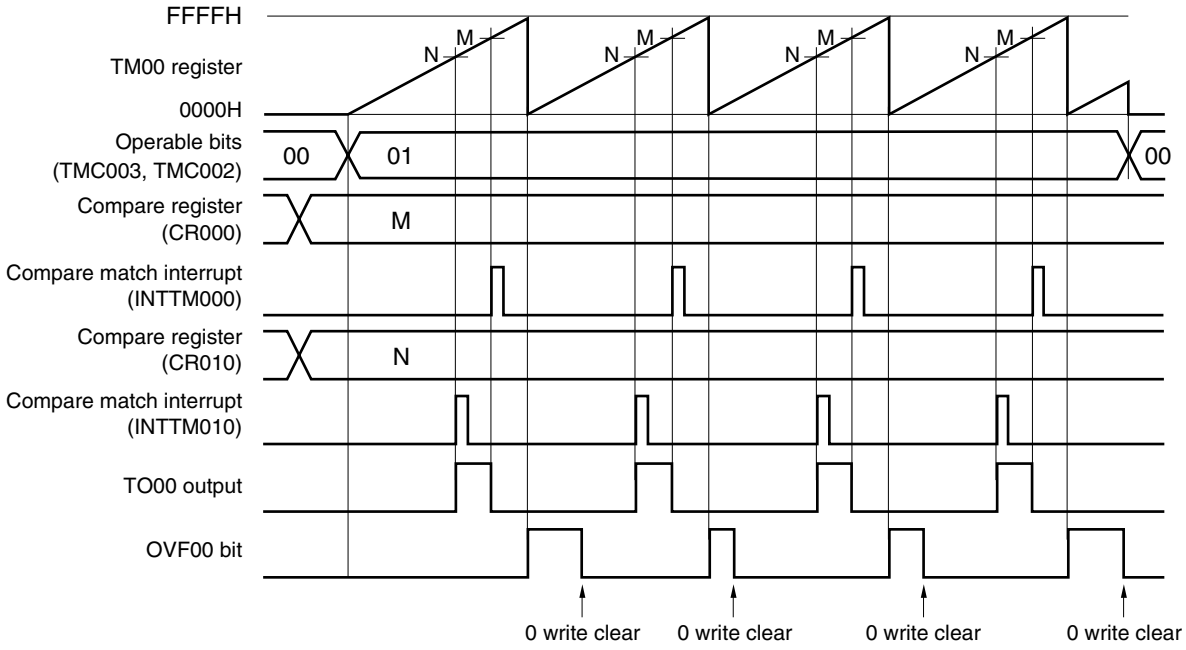


Figure 6-33. Timing Example of Free-Running Timer Mode
(CR000: Compare Register, CR010: Compare Register)

• TOC00 = 13H, PRM00 = 00H, CRC00 = 00H, TMC00 = 04H



This is an application example where two compare registers are used in the free-running timer mode.

The TO00 output level is reversed each time the count value of TM00 matches the set value of CR000 or CR010. When the count value matches the register value, the INTTM000 or INTTM010 signal is generated.

(2) Free-running timer mode operation
(CR000: compare register, CR010: capture register)

Figure 6-34. Block Diagram of Free-Running Timer Mode
(CR000: Compare Register, CR010: Capture Register)

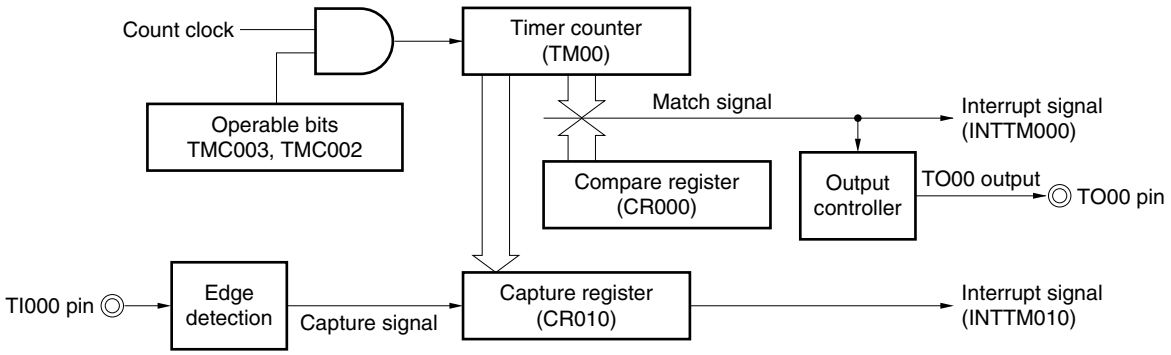
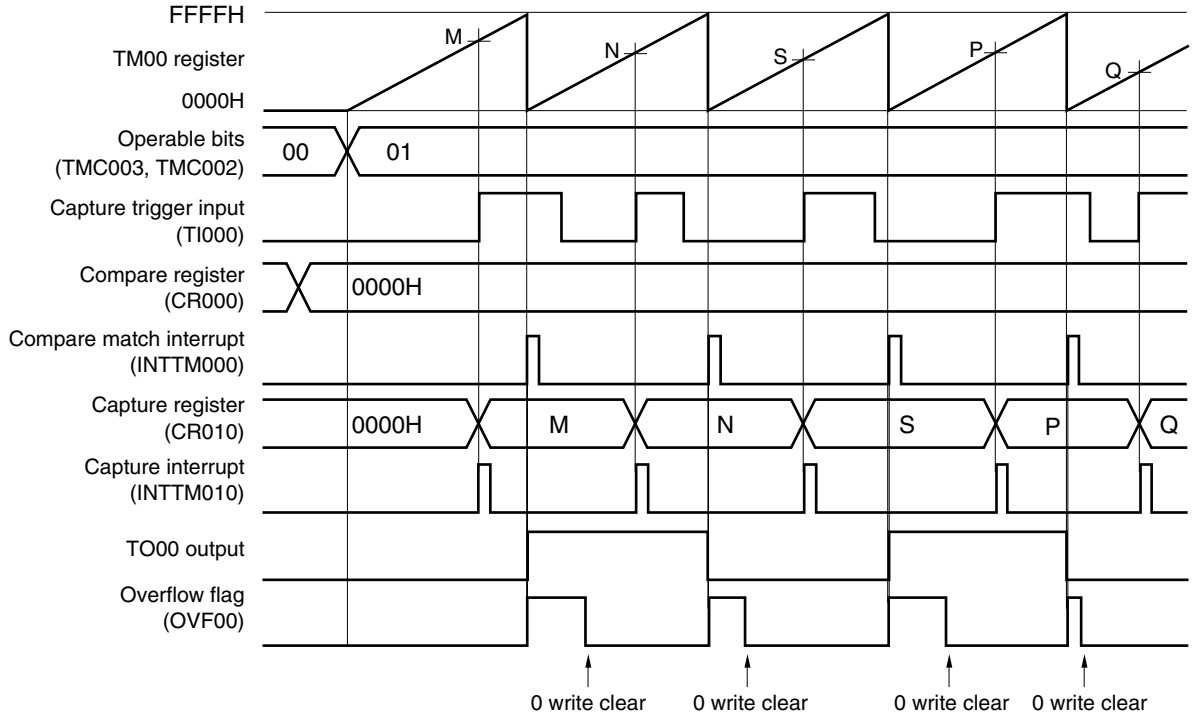


Figure 6-35. Timing Example of Free-Running Timer Mode
(CR000: Compare Register, CR010: Capture Register)

• TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 04H



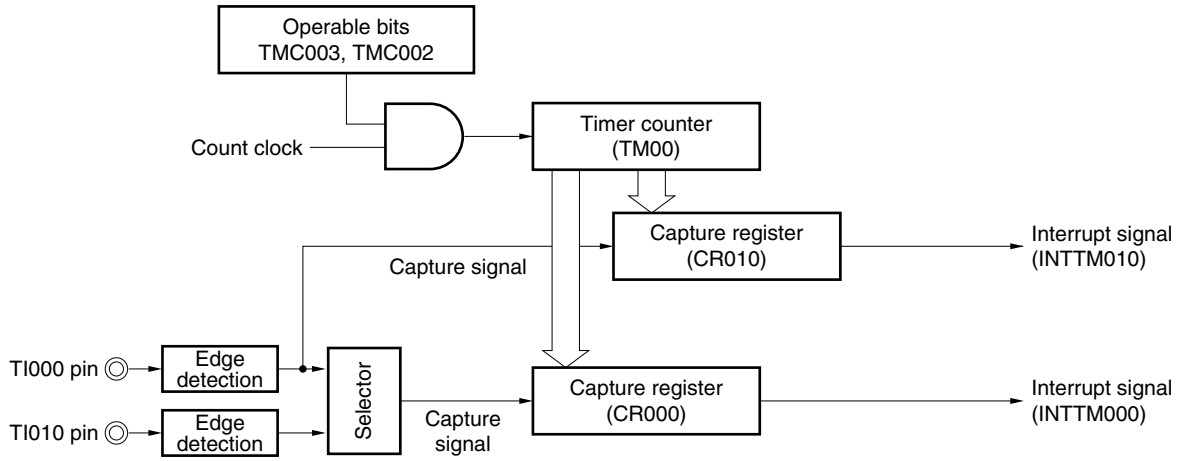
This is an application example where a compare register and a capture register are used at the same time in the free-running timer mode.

In this example, the INTTM000 signal is generated and the TO00 output level is reversed each time the count value of TM00 matches the set value of CR000 (compare register). In addition, the INTTM010 signal is generated and the count value of TM00 is captured to CR010 each time the valid edge of the TI000 pin is detected.

(3) Free-running timer mode operation

(CR000: capture register, CR010: capture register)

Figure 6-36. Block Diagram of Free-Running Timer Mode
(CR000: Capture Register, CR010: Capture Register)

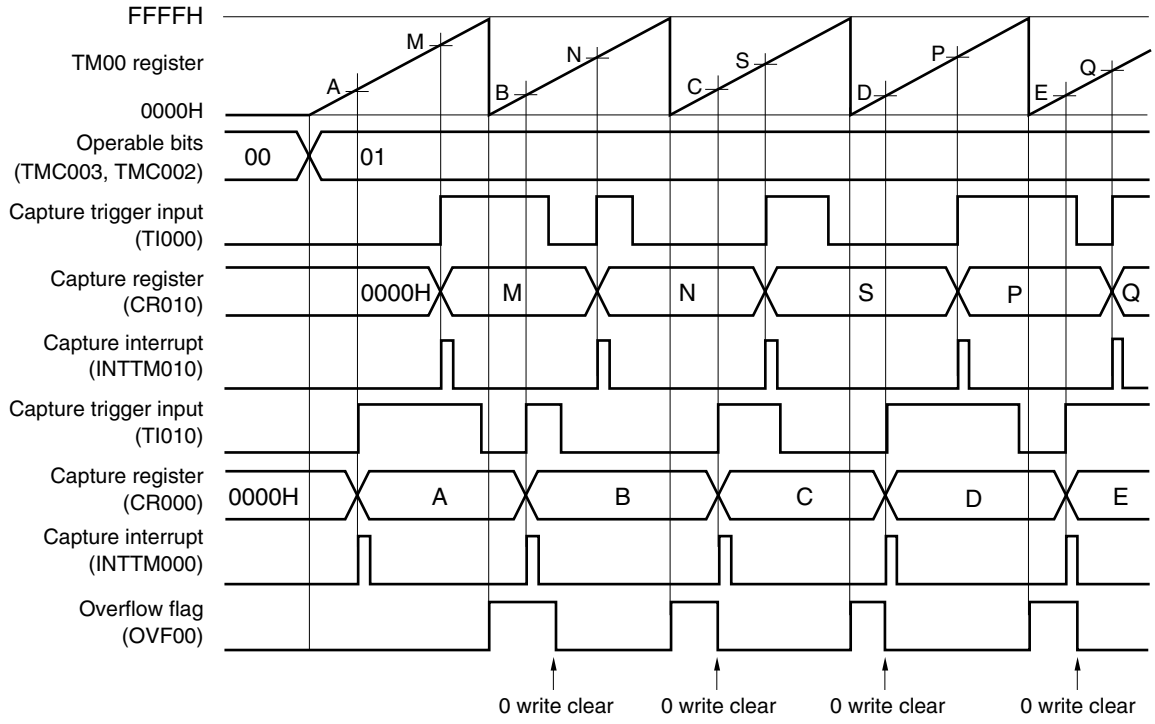


Remark If both CR000 and CR010 are used as capture registers in the free-running timer mode, the TO00 output level is not inverted.

However, it can be inverted each time the valid edge of the TI000 pin is detected if bit 1 (TMC001) of 16-bit timer mode control register 00 (TMC00) is set to 1.

Figure 6-37. Timing Example of Free-Running Timer Mode (CR000: Capture Register, CR010: Capture Register) (1/2)

(a) TOC00 = 13H, PRM00 = 50H, CRC00 = 05H, TMC00 = 04H

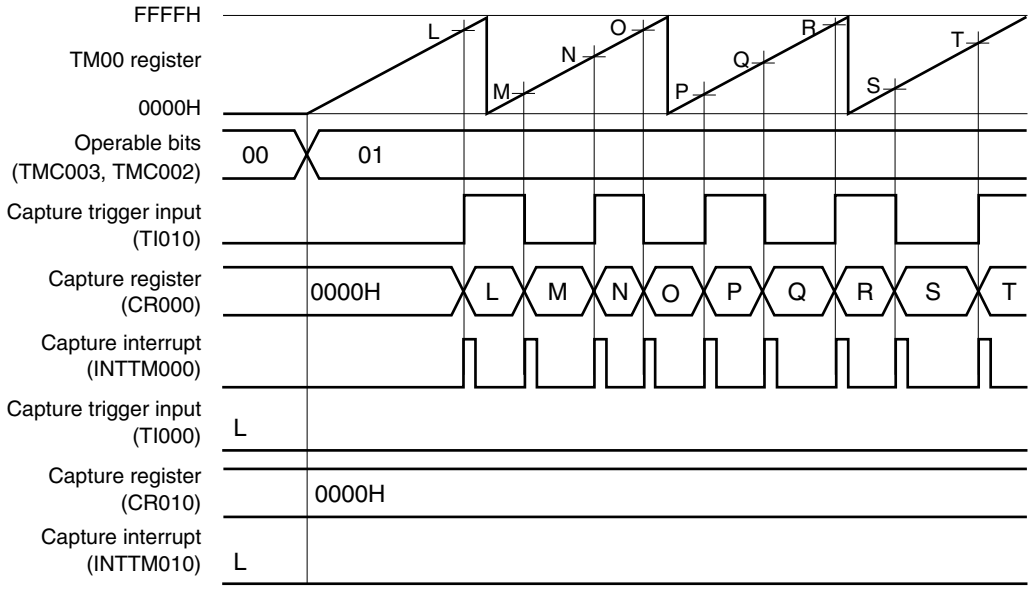


This is an application example where the count values that have been captured at the valid edges of separate capture trigger signals are stored in separate capture registers in the free-running timer mode.

The count value is captured to CR010 when the valid edge of the TI000 pin input is detected and to CR000 when the valid edge of the TI010 pin input is detected.

Figure 6-37. Timing Example of Free-Running Timer Mode (CR000: Capture Register, CR010: Capture Register) (2/2)

(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 04H

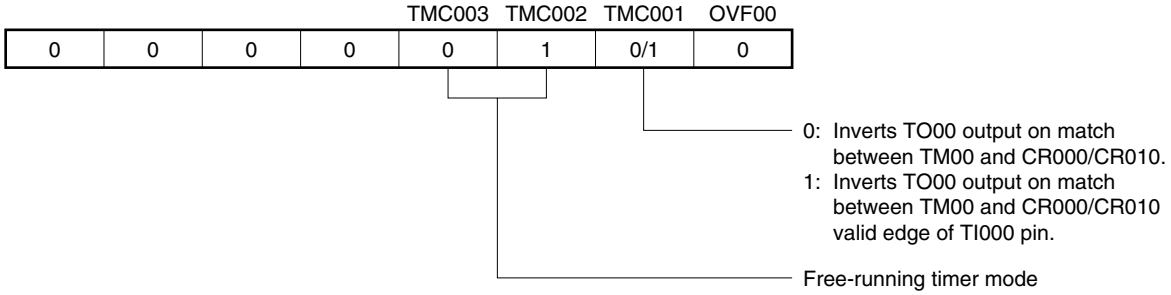


This is an application example where both the edges of the TI010 pin are detected and the count value is captured to CR000 in the free-running timer mode.

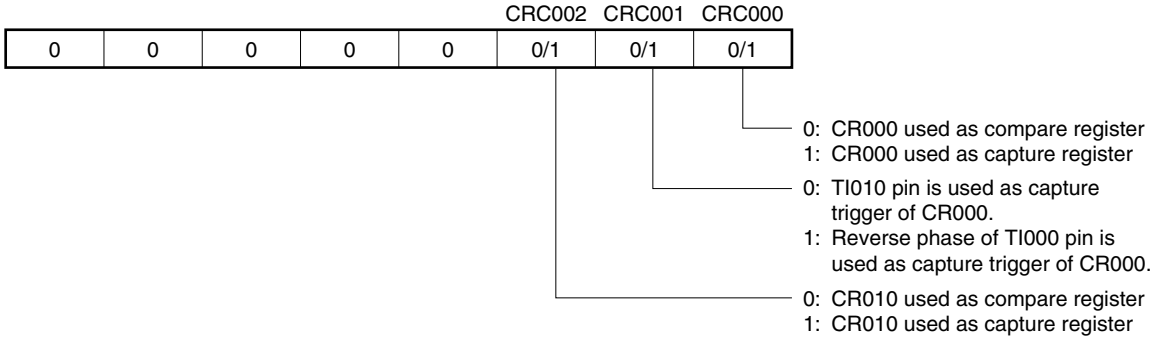
When both CR000 and CR010 are used as capture registers and when the valid edge of only the TI010 pin is to be detected, the count value cannot be captured to CR010.

Figure 6-38. Example of Register Settings in Free-Running Timer Mode (1/2)

(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)

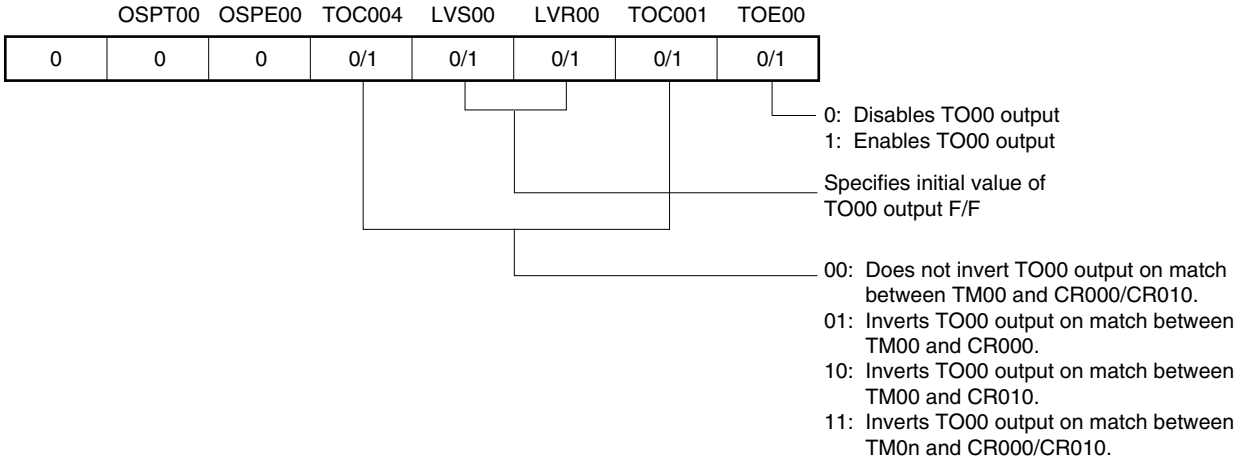
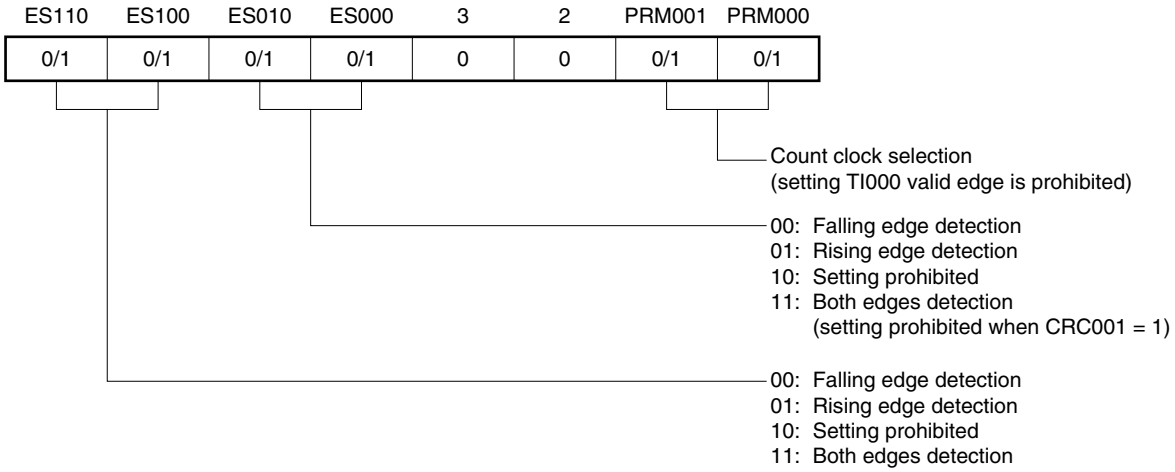


Figure 6-38. Example of Register Settings in Free-Running Timer Mode (2/2)

(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM000) is generated. The count value of TM00 is not cleared.

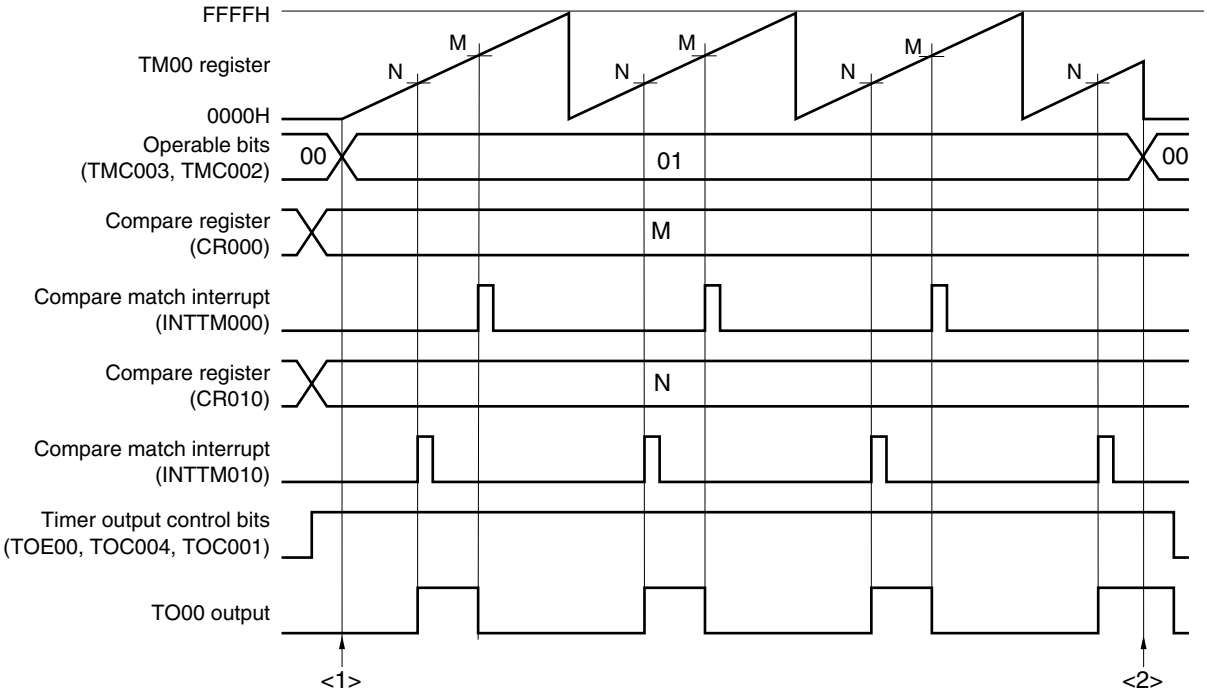
To use this register as a capture register, select either the TI000 or TI010 pin input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

(g) 16-bit capture/compare register 010 (CR010)

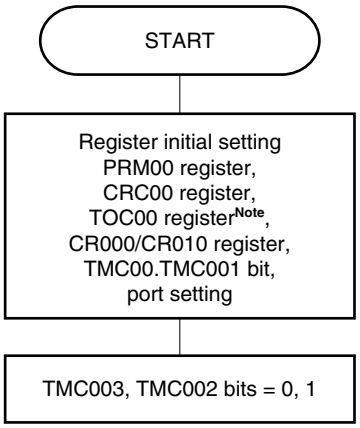
When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM010) is generated. The count value of TM00 is not cleared.

When this register is used as a capture register, the TI000 pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR010.

Figure 6-39. Example of Software Processing in Free-Running Timer Mode



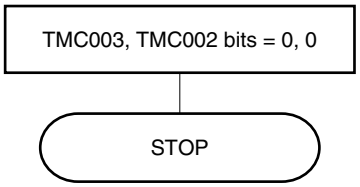
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 01.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

Note Care must be exercised when setting TOC00. For details, refer to **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.6 PPG output operation

A square wave having a pulse width set in advance by CR010 is output from the TO00 pin as a PPG (Programmable Pulse Generator) signal during a cycle set by CR000 when bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 11 (clear & start upon a match between TM00 and CR000).

The pulse cycle and duty factor of the pulse generated as the PPG output are as follows.

- Pulse cycle = (Set value of CR000 + 1) × Count clock cycle
- Duty = (Set value of CR010 + 1) / (Set value of CR000 + 1)

Caution To change the duty factor (value of CR010) during operation, refer to 6.5.1 Rewriting CR010 during TM00 operation.

Remarks 1. For the setting of I/O pins, refer to **6.3 (5) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

Figure 6-40. Block Diagram of PPG Output Operation

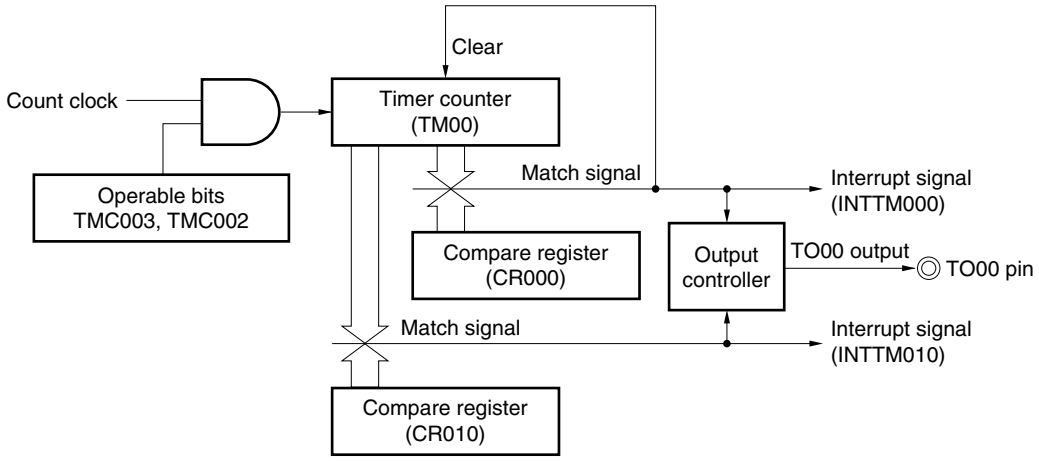
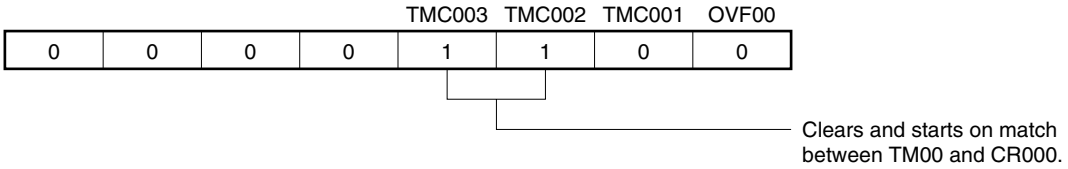
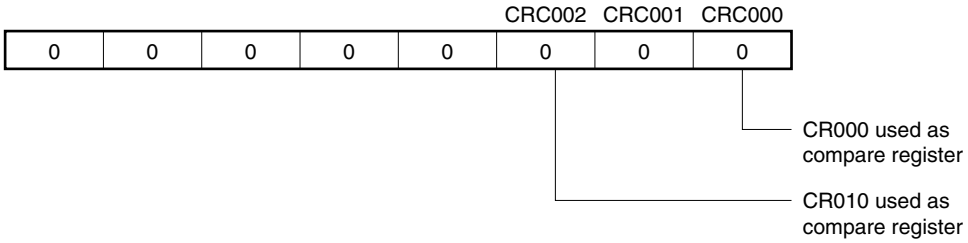


Figure 6-41. Example of Register Settings for PPG Output Operation (1/2)

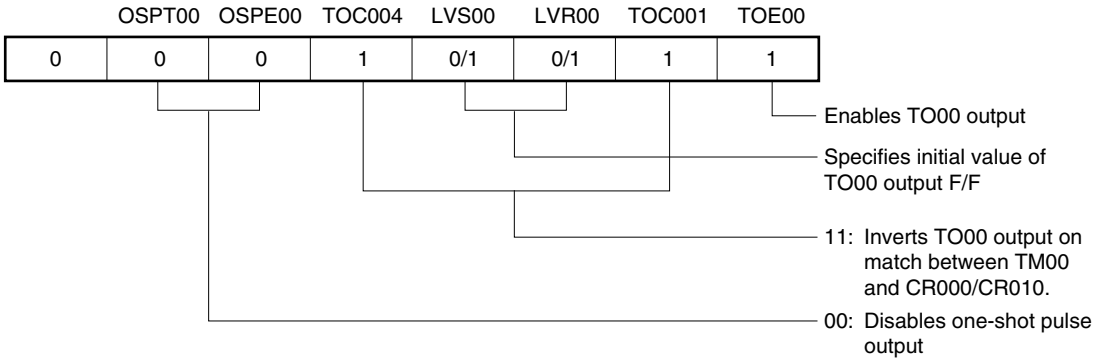
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)

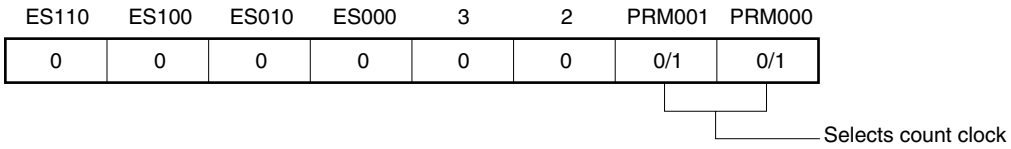


Figure 6-41. Example of Register Settings for PPG Output Operation (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

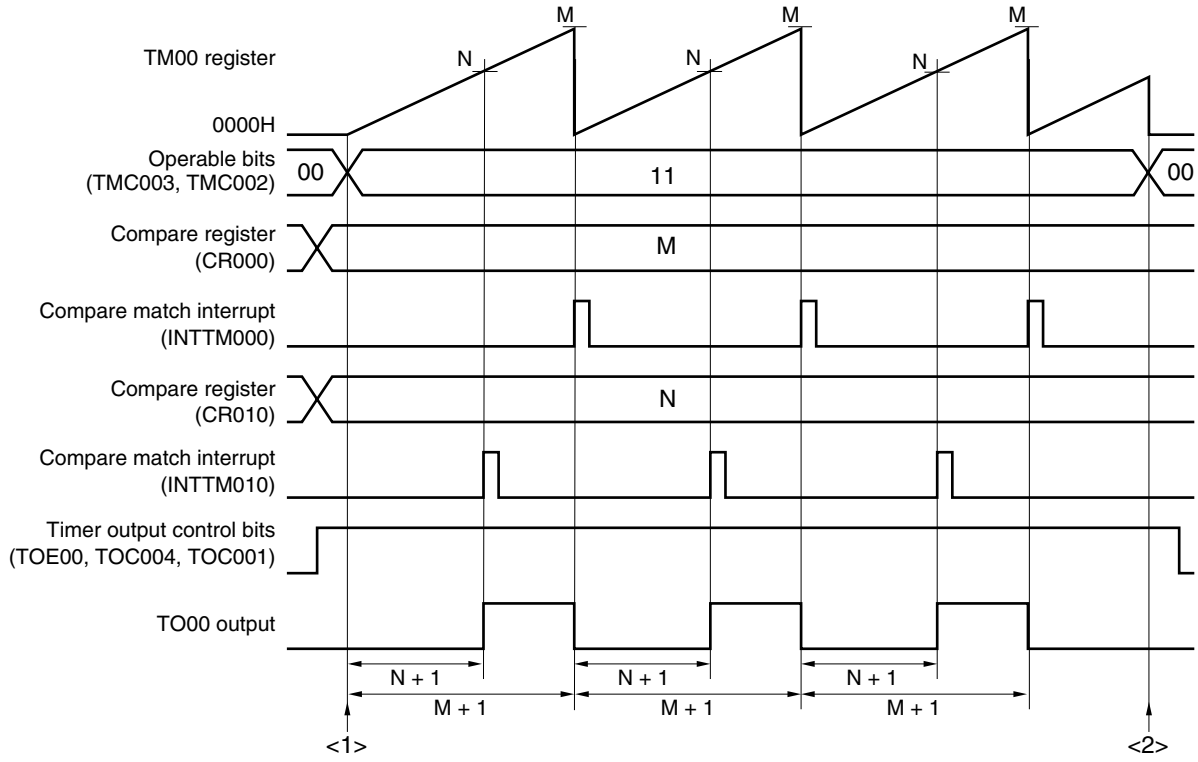
An interrupt signal (INTTM000) is generated when the value of this register matches the count value of TM00. The count value of TM00 is cleared.

(g) 16-bit capture/compare register 010 (CR010)

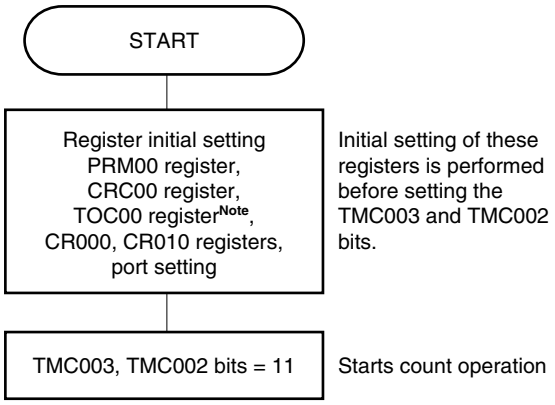
An interrupt signal (INTTM010) is generated when the value of this register matches the count value of TM00. The count value of TM00 is not cleared.

Caution Set values to CR000 and CR010 such that the condition 0000H ≤ CR010 < CR000 ≤ FFFFH is satisfied.

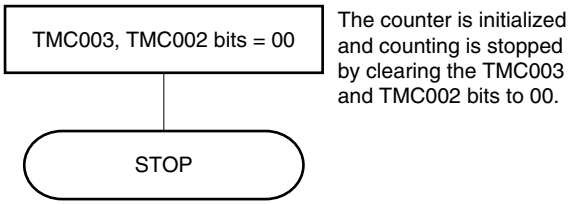
Figure 6-42. Example of Software Processing for PPG Output Operation



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, refer to **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

Remark PPG pulse cycle = (M + 1) × Count clock cycle
PPG duty = (N + 1)/(M + 1)

6.4.7 One-shot pulse output operation

A one-shot pulse can be output by setting bits 3 and 2 (TMC003 and TMC002) of the 16-bit timer mode control register 00 (TMC00) to 01 (free-running timer mode) or to 10 (clear & start mode entered by the TI000 pin valid edge) and setting bit 5 (OSPE00) of 16-bit timer output control register 00 (TOC00) to 1.

When bit 6 (OSPT00) of TOC00 is set to 1 or when the valid edge is input to the TI000 pin during timer operation, clearing & starting of TM00 is triggered, and a pulse of the difference between the values of CR000 and CR010 is output only once from the TO00 pin.

Cautions 1. Do not input the trigger again (setting OSPT00 to 1 or detecting the valid edge of the TI000 pin) while the one-shot pulse is output. To output the one-shot pulse again, generate the trigger after the current one-shot pulse output has completed.

2. To use only the setting of OSPT00 to 1 as the trigger of one-shot pulse output, do not change the level of the TI000 pin or its alternate function port pin. Otherwise, the pulse will be unexpectedly output.

Remarks 1. For the setting of the I/O pins, refer to **6.3 (5) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

Figure 6-43. Block Diagram of One-Shot Pulse Output Operation

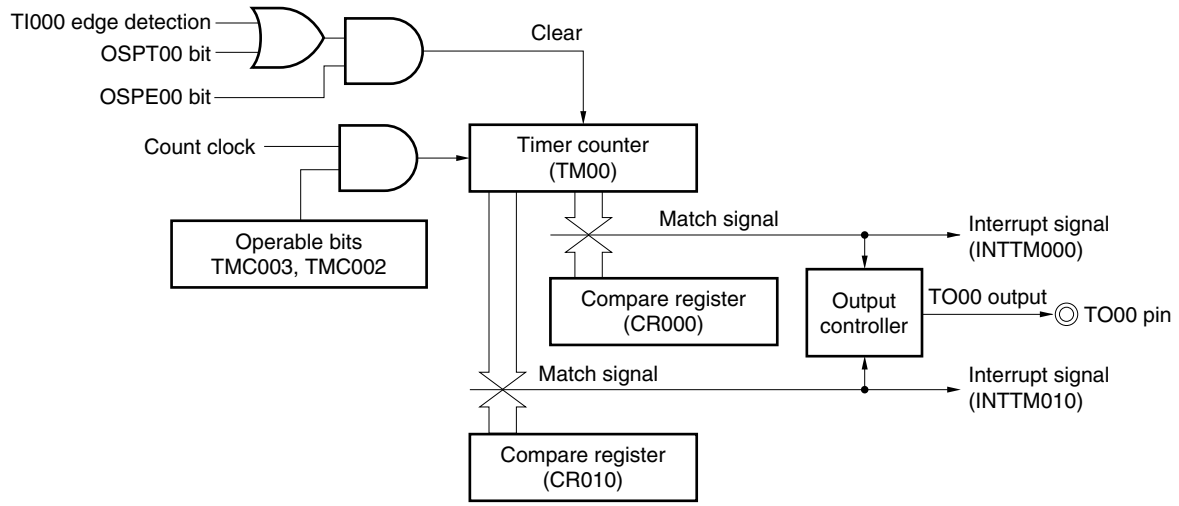
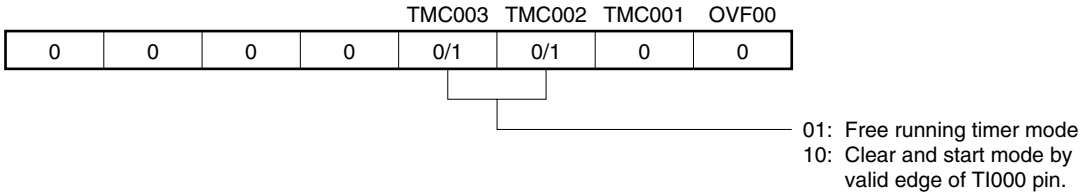
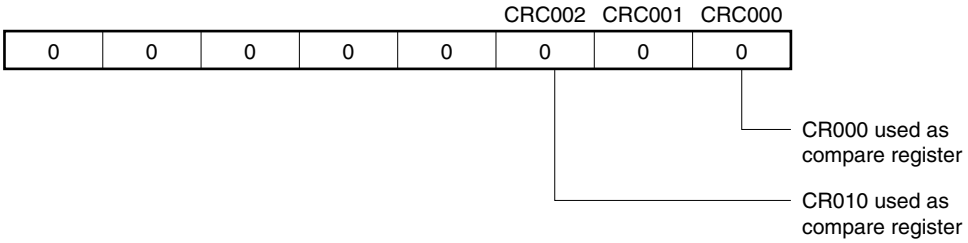


Figure 6-44. Example of Register Settings for One-Shot Pulse Output Operation (1/2)

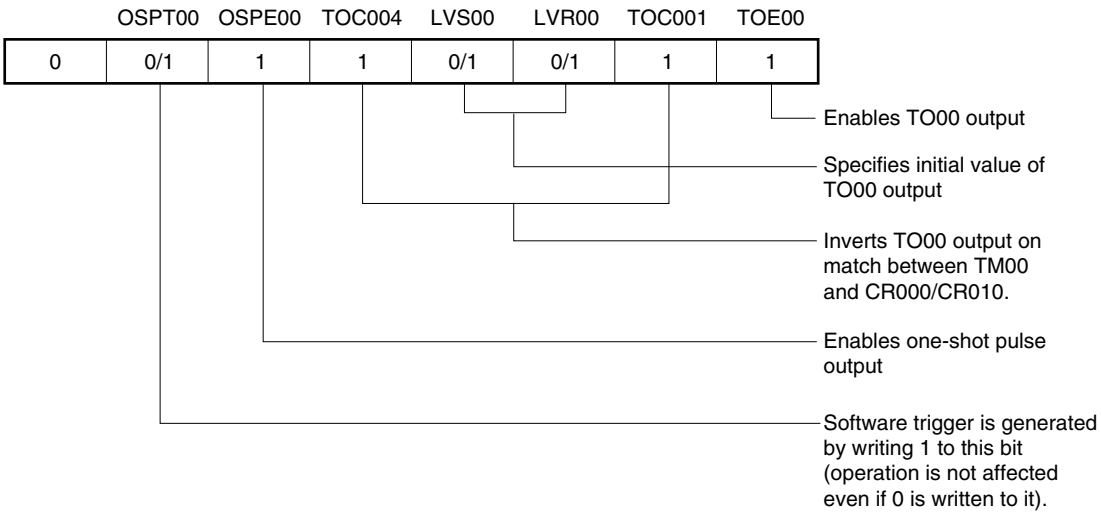
(a) 16-bit timer mode control register 00 (TMC00)



(b) Capture/compare control register 00 (CRC00)



(c) 16-bit timer output control register 00 (TOC00)



(d) Prescaler mode register 00 (PRM00)

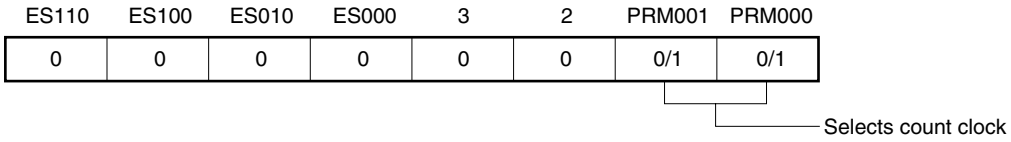


Figure 6-44. Example of Register Settings for One-Shot Pulse Output Operation (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

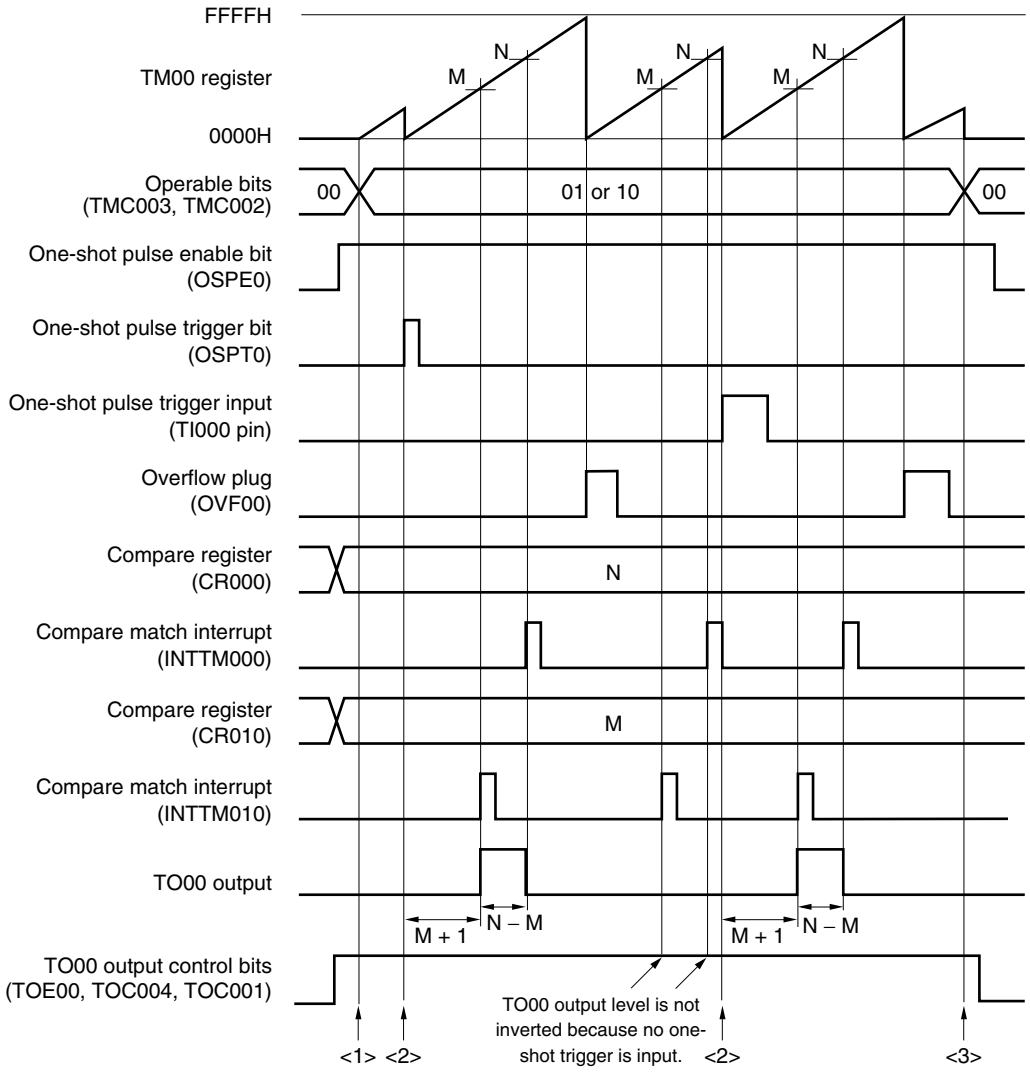
This register is used as a compare register when a one-shot pulse is output. When the value of TM00 matches that of CR000, an interrupt signal (INTTM000) is generated and the TO00 output level is inverted.

(g) 16-bit capture/compare register 010 (CR010)

This register is used as a compare register when a one-shot pulse is output. When the value of TM00 matches that of CR010, an interrupt signal (INTTM010) is generated and the TO00 output level is inverted.

Caution Do not set the same value to CR000 and CR010.

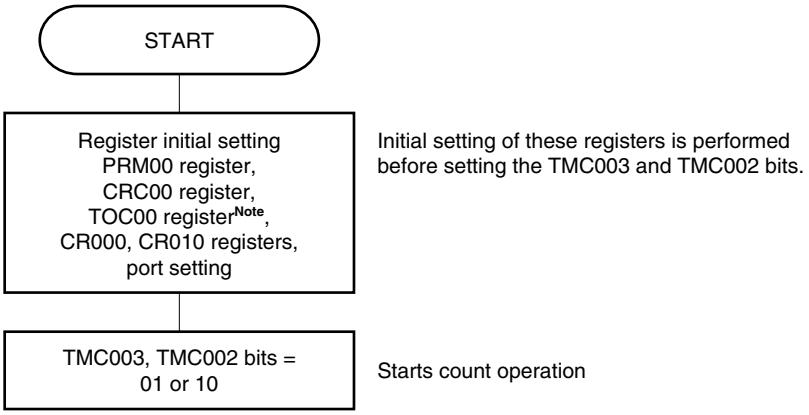
Figure 6-45. Example of Software Processing for One-Shot Pulse Output Operation (1/2)



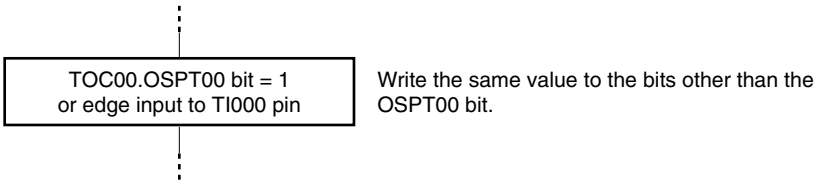
- Time from when the one-shot pulse trigger is input until the one-shot pulse is output
  = (M + 1) × Count clock cycle
- One-shot pulse output active level width
  = (N – M) × Count clock cycle

Figure 6-45. Example of Software Processing for One-Shot Pulse Output Operation (2/2)

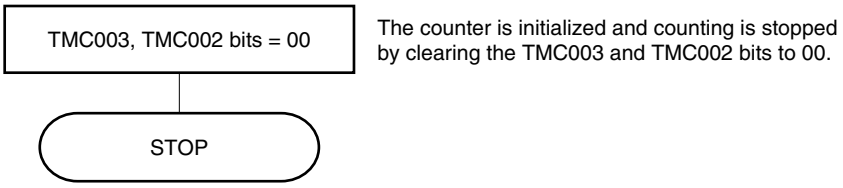
<1> Count operation start flow



<2> One-shot trigger input flow



<3> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, refer to **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.8 Pulse width measurement operation

TM00 can be used to measure the pulse width of the signal input to the TI000 and TI010 pins.

Measurement can be accomplished by operating the 16-bit timer/event counter 00 in the free-running timer mode or by restarting the timer in synchronization with the signal input to the TI000 pin.

When an interrupt is generated, read the value of the valid capture register and measure the pulse width. Check bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00). If it is set (to 1), clear it to 0 by software.

Figure 6-46. Block Diagram of Pulse Width Measurement (Free-Running Timer Mode)

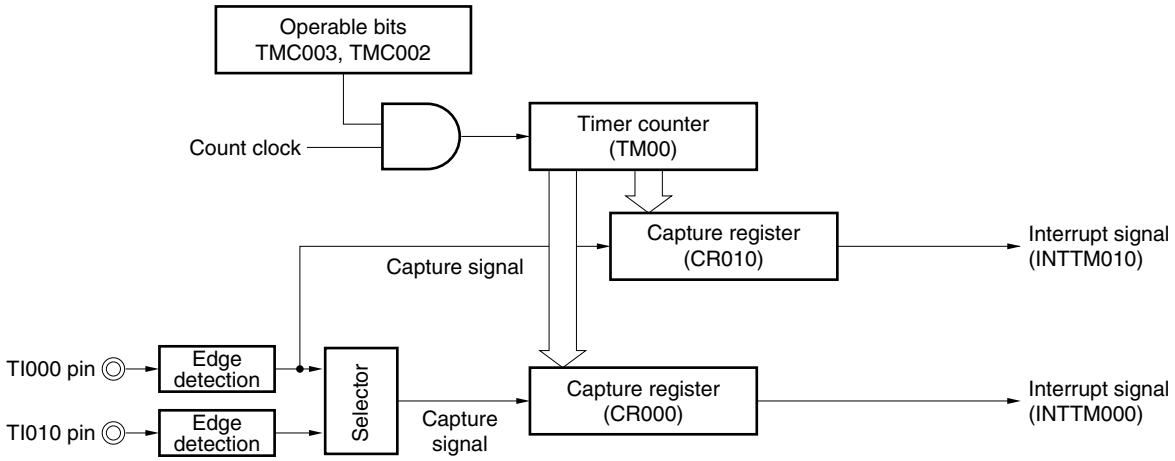
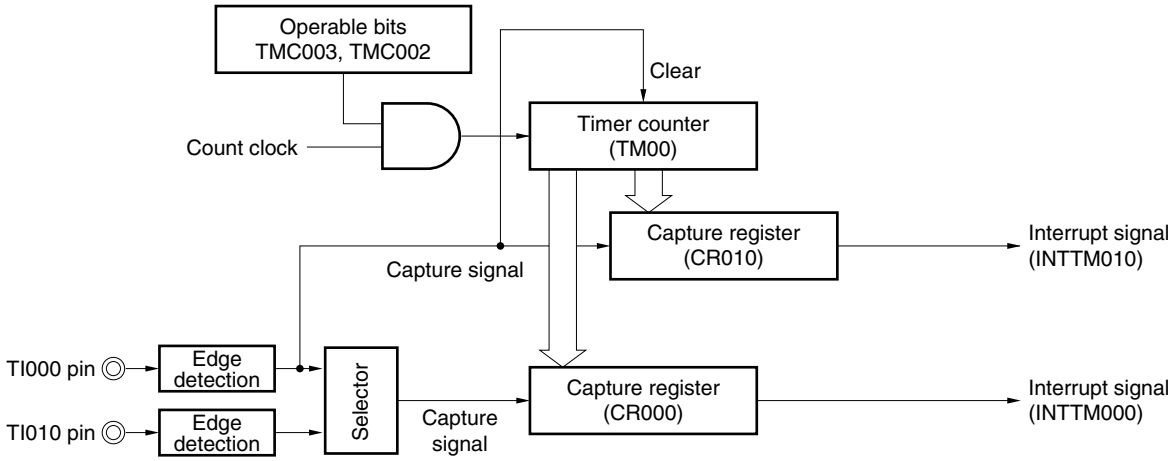


Figure 6-47. Block Diagram of Pulse Width Measurement
(Clear & Start Mode Entered by TI000 Pin Valid Edge Input)



A pulse width can be measured in the following three ways.

- Measuring the pulse width by using two input signals of the TI000 and TI010 pins (free-running timer mode)
- Measuring the pulse width by using one input signal of the TI000 pin (free-running timer mode)
- Measuring the pulse width by using one input signal of the TI000 pin (clear & start mode entered by the TI000 pin valid edge input)

Remarks 1. For the setting of the I/O pins, refer to **6.3 (5) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

(1) Measuring the pulse width by using two input signals of the TI000 and TI010 pins (free-running timer mode)

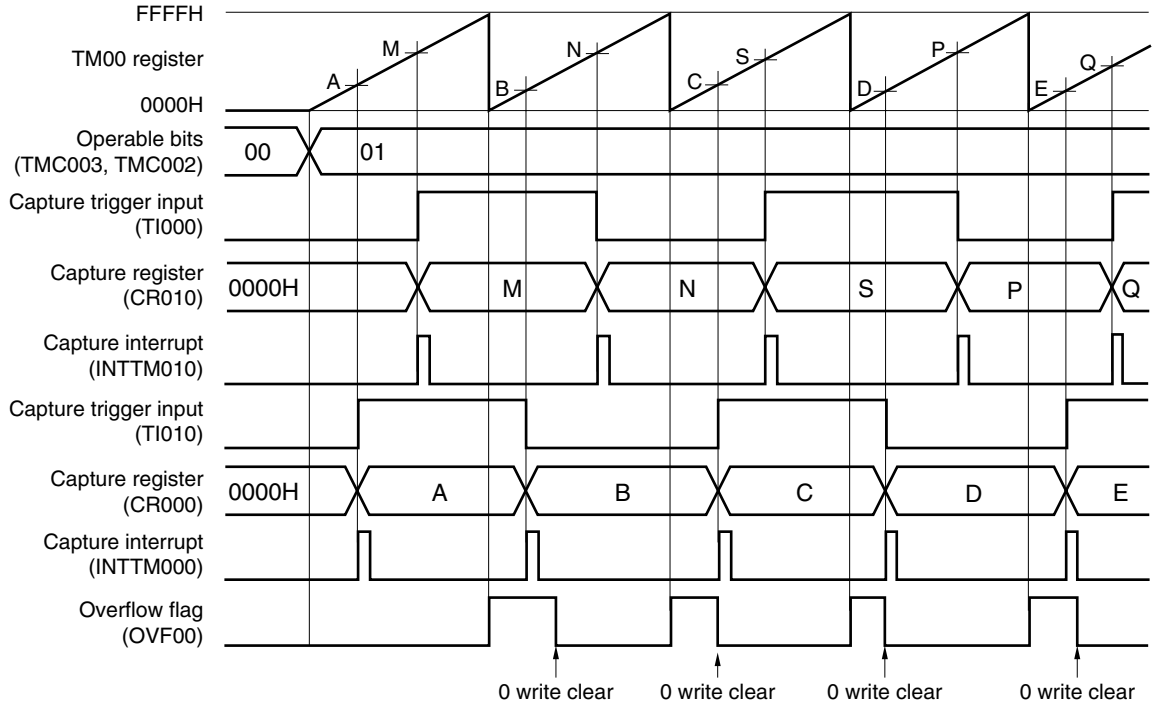
Set the free-running timer mode (TMC003 and TMC002 = 01). When the valid edge of the TI000 pin is detected, the count value of TM00 is captured to CR010. When the valid edge of the TI010 pin is detected, the count value of TM00 is captured to CR000. Specify detection of both the edges of the TI000 and TI010 pins.

By this measurement method, the previous count value is subtracted from the count value captured by the edge of each input signal. Therefore, save the previously captured value to a separate register in advance.

If an overflow occurs, the value becomes negative if the previously captured value is simply subtracted from the current captured value and, therefore, a borrow occurs (bit 0 (CY) of the program status word (PSW) is set to 1). If this happens, ignore CY and take the calculated value as the pulse width. In addition, clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 6-48. Timing Example of Pulse Width Measurement (1)

• TMC00 = 04H, PRM00 = F0H, CRC00 = 05H



(2) Measuring the pulse width by using one input signal of the TI000 pin (free-running timer mode)

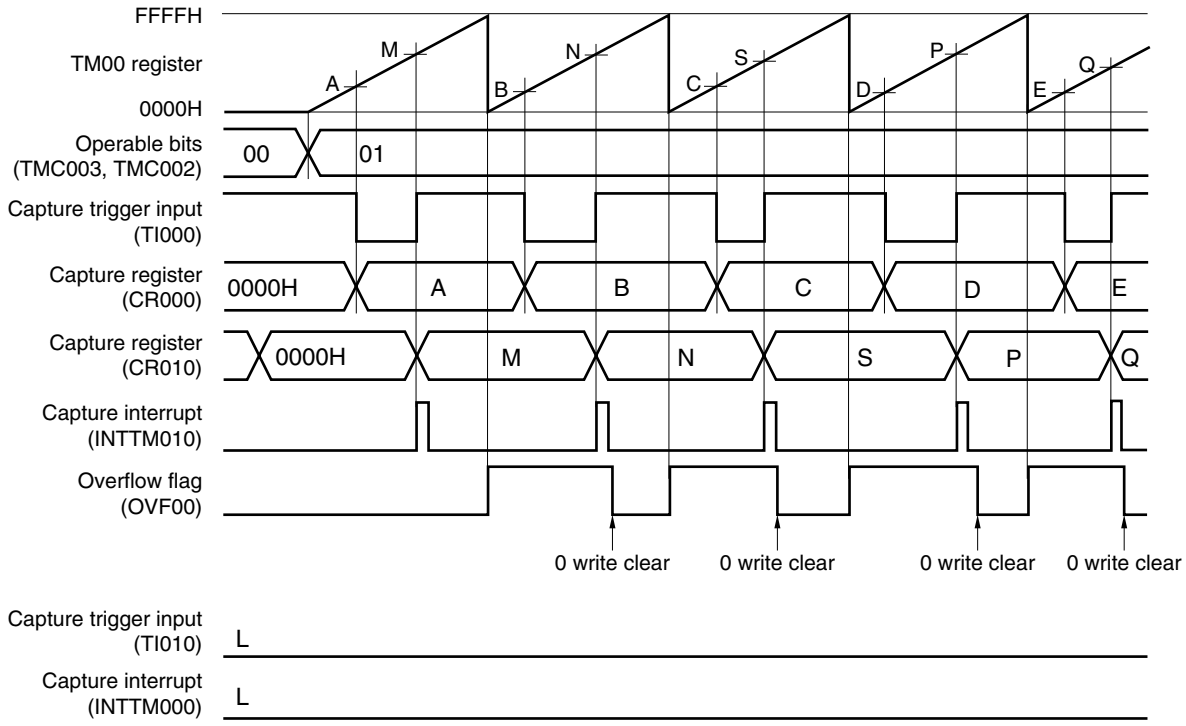
Set the free-running timer mode (TMC003 and TMC002 = 01). The count value of TM00 is captured to CR000 in the phase reverse to the valid edge detected on the TI000 pin. When the valid edge of the TI000 pin is detected, the count value of TM00 is captured to CR010.

By this measurement method, values are stored in separate capture registers when a width from one edge to another is measured. Therefore, the capture values do not have to be saved. By subtracting the value of one capture register from that of another, a high-level width, low-level width, and cycle are calculated.

If an overflow occurs, the value becomes negative if one captured value is simply subtracted from another and, therefore, a borrow occurs (bit 0 (CY) of the program status word (PSW) is set to 1). If this happens, ignore CY and take the calculated value as the pulse width. In addition, clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 6-49. Timing Example of Pulse Width Measurement (2)

• TMC00 = 04H, PRM00 = 10H, CRC00 = 07H



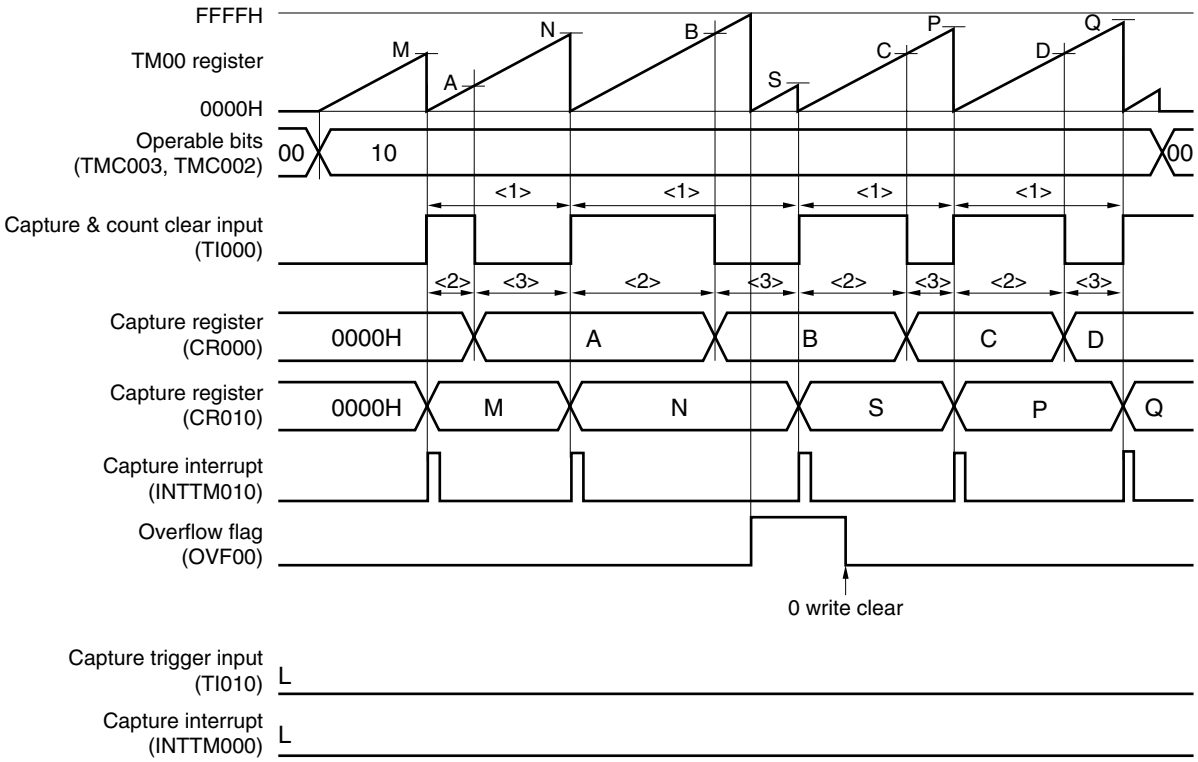
(3) Measuring the pulse width by using one input signal of the TI000 pin (clear & start mode entered by the TI000 pin valid edge input)

Set the clear & start mode entered by the TI000 pin valid edge (TMC003 and TMC002 = 10). The count value of TM00 is captured to CR000 in the phase reverse to the valid edge of the TI000 pin, and the count value of TM00 is captured to CR010 and TM00 is cleared (0000H) when the valid edge of the TI000 pin is detected. Therefore, a cycle is stored in CR010 if TM00 does not overflow.

If an overflow occurs, take the value that results from adding 10000H to the value stored in CR010 as a cycle. Clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 6-50. Timing Example of Pulse Width Measurement (3)

• TMC00 = 08H, PRM00 = 10H, CRC00 = 07H



<1> Pulse cycle = (10000H × Number of times OVF00 bit is set to 1 + Captured value of CR010) × Count clock cycle

<2> High-level pulse width = (10000H × Number of times OVF00 bit is set to 1 + Captured value of CR000) × Count clock cycle

<3> Low-level pulse width = (Pulse cycle − High-level pulse width)

Figure 6-51. Example of Register Settings for Pulse Width Measurement (1/2)

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	0/1	0/1	0	0

01: Free running timer mode
10: Clear and start mode entered by valid edge of TI000 pin.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	1	0/1	1

1: CR000 used as capture register

0: TI010 pin is used as capture trigger of CR000.
1: Reverse phase of TI000 pin is used as capture trigger of CR000.

1: CR010 used as capture register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0	0	0	0	0

(d) Prescaler mode register 00 (PRM00)

ES110	ES100	ES010	ES000	3	2	PRM001	PRM000
0/1	0/1	0/1	0/1	0	0	0/1	0/1

Selects count clock
(setting valid edge of TI000 is prohibited)

00: Falling edge detection
01: Rising edge detection
10: Setting prohibited
11: Both edges detection
(setting when CRC001 = 1 is prohibited)

00: Falling edge detection
01: Rising edge detection
10: Setting prohibited
11: Both edges detection

Figure 6-51. Example of Register Settings for Pulse Width Measurement (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

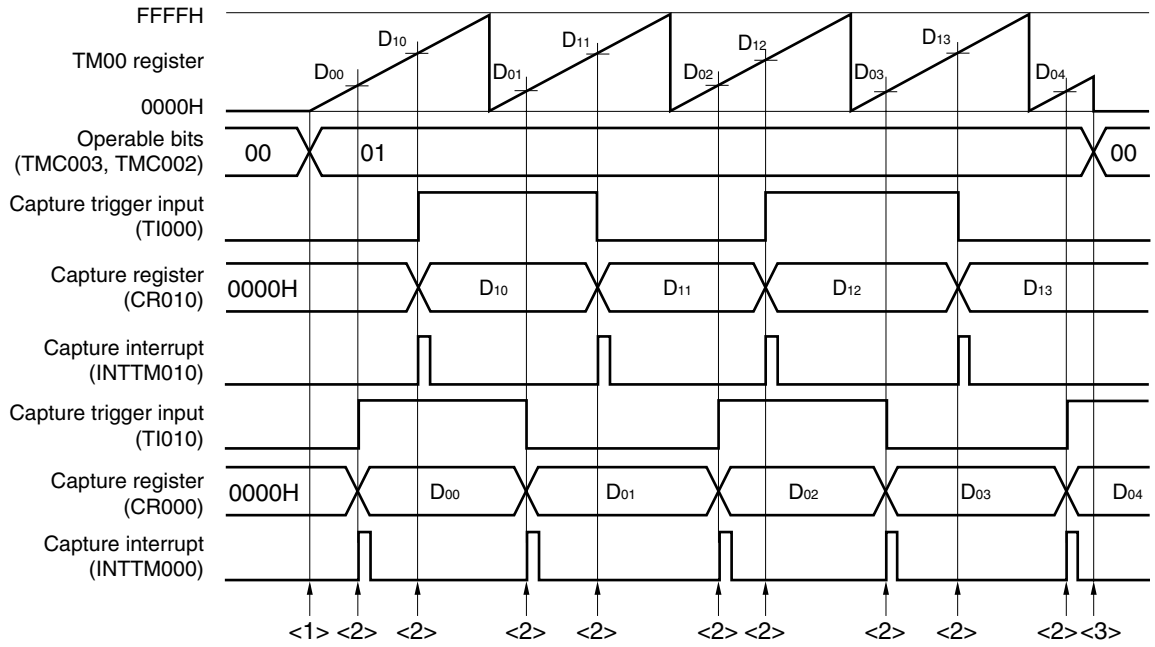
This register is used as a capture register. Either the TI000 or TI010 pin is selected as a capture trigger. When a specified edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

(g) 16-bit capture/compare register 010 (CR010)

This register is used as a capture register. The signal input to the TI000 pin is used as a capture trigger. When the capture trigger is detected, the count value of TM00 is stored in CR010.

Figure 6-52. Example of Software Processing for Pulse Width Measurement (1/2)

(a) Example of free-running timer mode



(b) Example of clear & start mode entered by TI000 pin valid edge

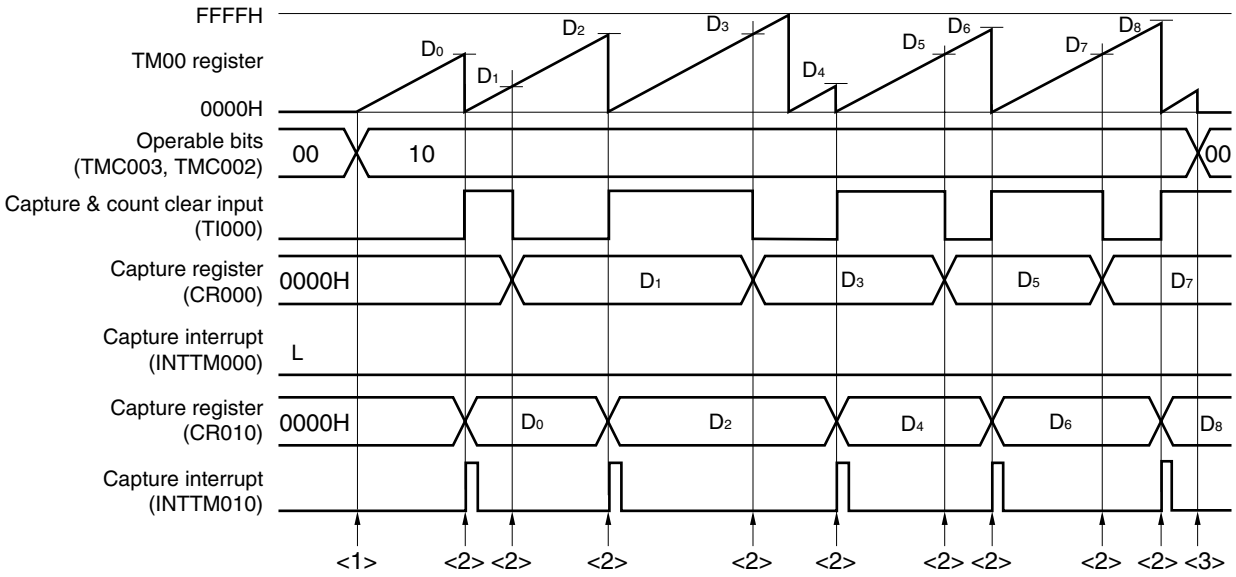
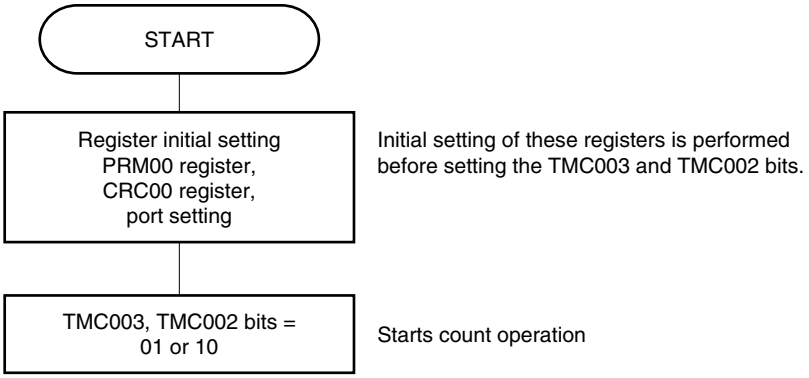
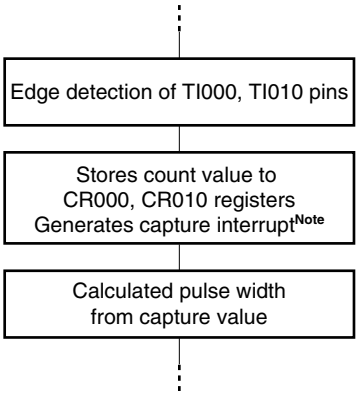


Figure 6-52. Example of Software Processing for Pulse Width Measurement (2/2)

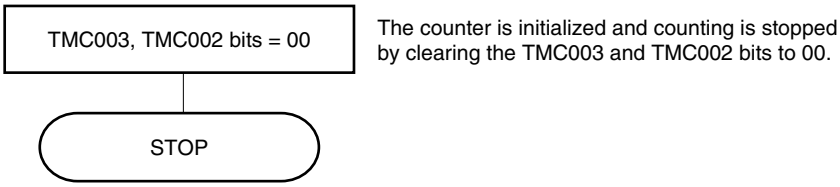
<1> Count operation start flow



<2> Capture trigger input flow



<3> Count operation stop flow



Note The capture interrupt signal (INTTM000) is not generated when the reverse-phase edge of the TI000 pin input is selected to the valid edge of CR000.

6.5 Special Use of TM00

6.5.1 Rewriting CR010 during TM00 operation

In principle, rewriting CR000 and CR010 of the μPD79F7023, 79F7024 microcontrollers when they are used as compare registers is prohibited while TM00 is operating (TMC003 and TMC002 = other than 00).

However, the value of CR010 can be changed, even while TM00 is operating, using the following procedure if CR010 is used for PPG output and the duty factor is changed. (When changing the value of CR010 to a smaller value than the current one, rewrite it immediately after its value matches the value of TM00. When changing the value of CR010 to a larger value than the current one, rewrite it immediately after the values of CR000 and TM00 match. If the value of CR010 is rewritten immediately before a match between CR010 and TM00, or between CR000 and TM00, an unexpected operation may be performed.).

Procedure for changing value of CR010

<1> Disable interrupt INTTM010 (TMMK010 = 1).
<2> Disable reversal of the timer output when the value of TM00 matches that of CR010 (TOC004 = 0).
<3> Change the value of CR010.
<4> Wait for one cycle of the count clock of TM00.
<5> Enable reversal of the timer output when the value of TM00 matches that of CR010 (TOC004 = 1).
<6> Clear the interrupt flag of INTTM010 (TMIF010 = 0) to 0.
<7> Enable interrupt INTTM010 (TMMK010 = 0).

Remark For TMIF010 and TMMK010, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

6.5.2 Setting LVS00 and LVR00

(1) Usage of LVS00 and LVR00

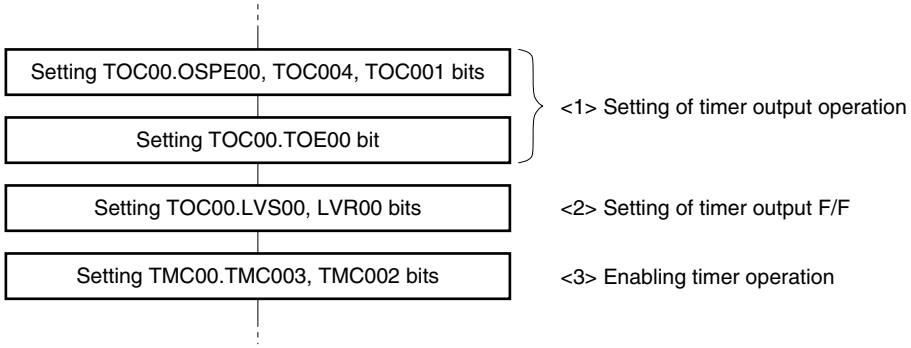
LVS00 and LVR00 are used to set the default value of the TO00 output and to invert the timer output without enabling the timer operation (TMC003 and TMC002 = 00). Clear LVS00 and LVR00 to 00 (default value: low-level output) when software control is unnecessary.

LVS00	LVR00	Timer Output Status
0	0	Not changed (low-level output)
0	1	Cleared (low-level output)
1	0	Set (high-level output)
1	1	Setting prohibited

(2) Setting LVS00 and LVR00

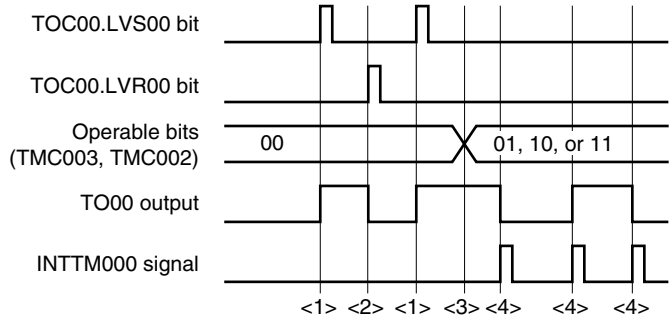
Set LVS00 and LVR00 using the following procedure.

Figure 6-53. Example of Flow for Setting LVS00 and LVR00 Bits



Caution Be sure to set LVS00 and LVR00 following steps <1>, <2>, and <3> above.
Step <2> can be performed after <1> and before <3>.

Figure 6-54. Timing Example of LVR00 and LVS00



<1> The TO00 output goes high when LVS00 and LVR00 = 10.

<2> The TO00 output goes low when LVS00 and LVR00 = 01 (the pin output remains unchanged from the high level even if LVS00 and LVR00 are cleared to 00).

<3> The timer starts operating when TMC003 and TMC002 are set to 01, 10, or 11. Because LVS00 and LVR00 were set to 10 before the operation was started, the TO00 output starts from the high level. After the timer starts operating, setting LVS00 and LVR00 is prohibited until TMC003 and TMC002 = 00 (disabling the timer operation).

<4> The TO00 output level is inverted each time an interrupt signal (INTTM000) is generated.

6.6 Cautions for 16-Bit Timer/Event Counter 00

(1) Restrictions for each channel of 16-bit timer/event counter 00

Table 6-3 shows the restrictions for each channel.

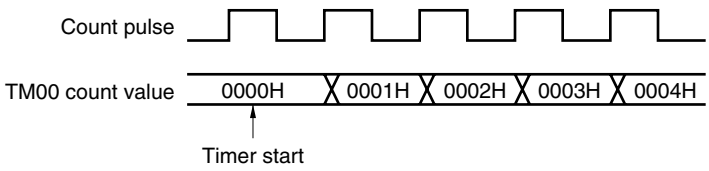
Table 6-3. Restrictions for Each Channel of 16-Bit Timer/Event Counter 00

Operation	Restriction
As interval timer	–
As square-wave output	
As external event counter	
As clear & start mode entered by TI000 pin valid edge input	Using timer output (TO00) is prohibited when detection of the valid edge of the TI010 pin is used. (TOC00 = 00H)
As free-running timer	–
As PPG output	0000H ≤ CP010 < CR000 ≤ FFFFH
As one-shot pulse output	Setting the same value to CR000 and CP010 is prohibited.
As pulse width measurement	Using timer output (TO00) is prohibited (TOC00 = 00H)

(2) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because counting TM00 is started asynchronously to the count pulse.

Figure 6-55. Start Timing of TM00 Count



(3) Setting of CR000 and CR010 (clear & start mode entered upon a match between TM00 and CR000)

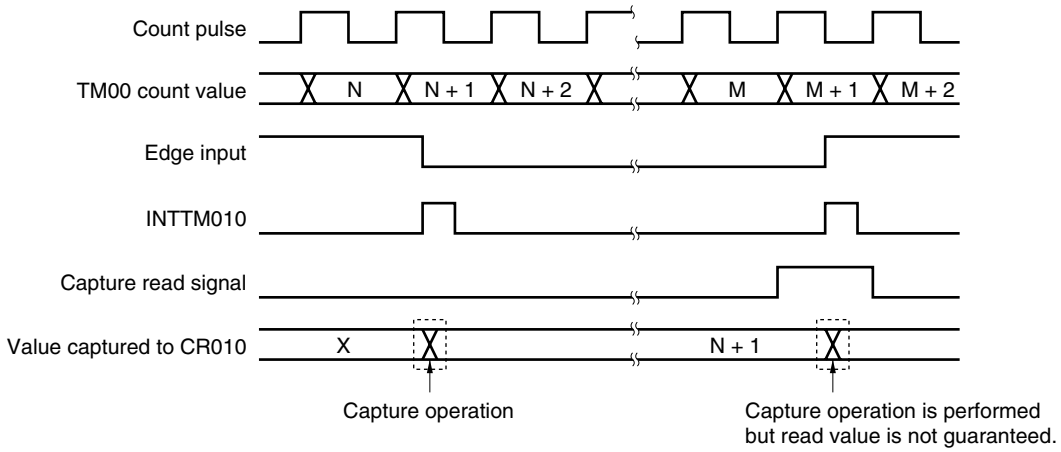
Set a value other than 0000H to CR000 and CR010 (TM00 cannot count one pulse when it is used as an external event counter).

(4) Timing of holding data by capture register

(a) When the valid edge is input to the TI000/TI010 pin and the reverse phase of the TI000 pin is detected while CR000/CR010 is read, CR010 performs a capture operation but the read value of CR000/CR010 is not guaranteed. At this time, an interrupt signal (INTTM000/INTTM010) is generated when the valid edge of the TI000/TI010 pin is detected (the interrupt signal is not generated when the reverse-phase edge of the TI000 pin is detected).

When the count value is captured because the valid edge of the TI000/TI010 pin was detected, read the value of CR000/CR010 after INTTM000/INTTM010 is generated.

Figure 6-56. Timing of Holding Data by Capture Register



(b) The values of CR000 and CR010 are not guaranteed after 16-bit timer/event counter 00 stops.

(5) Setting valid edge

Set the valid edge of the TI000 pin while the timer operation is stopped (TMC003 and TMC002 = 00). Set the valid edge by using ES000 and ES010.

(6) Re-triggering one-shot pulse

Make sure that the trigger is not generated while an active level is being output in the one-shot pulse output mode. Be sure to input the next trigger after the current active level is output.

(7) Operation of OVF00 flag

(a) Setting OVF00 flag (1)

The OVF00 flag is set to 1 in the following case, as well as when TM00 overflows.

Select the clear & start mode entered upon a match between TM00 and CR000.

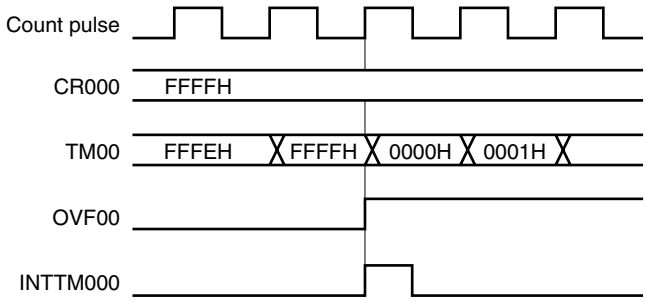
↓

Set CR000 to FFFFH.

↓

When TM00 matches CR000 and TM00 is cleared from FFFFH to 0000H

Figure 6-57. Operation Timing of OVF00 Flag



(b) Clearing OVF00 flag

Even if the OVF00 flag is cleared to 0 after TM00 overflows and before the next count clock is counted (before the value of TM00 becomes 0001H), it is set to 1 again and clearing is invalid.

(8) One-shot pulse output

One-shot pulse output operates correctly in the free-running timer mode or the clear & start mode entered by the TI000 pin valid edge. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM00 and CR000.

(9) Capture operation

(a) When valid edge of TI000 is specified as count clock

When the valid edge of TI000 is specified as the count clock, the capture register for which TI000 is specified as a trigger does not operate correctly.

(b) Pulse width to accurately capture value by signals input to TI010 and TI000 pins

To accurately capture the count value, the pulse input to the TI000 and TI010 pins as a capture trigger must be wider than two count clocks selected by PRM00 (refer to **Figure 6-7**).

(c) Generation of interrupt signal

The capture operation is performed at the falling edge of the count clock but the interrupt signals (INTTM000 and INTTM010) are generated at the rising edge of the next count clock (refer to **Figure 6-7**).

(d) Note when CRC001 (bit 1 of capture/compare control register 00 (CRC00)) is set to 1

When the count value of the TM00 register is captured to the CR000 register in the phase reverse to the signal input to the TI000 pin, the interrupt signal (INTTM000) is not generated after the count value is captured. If the valid edge is detected on the TI010 pin during this operation, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal. Mask the INTTM000 signal when the external interrupt is not used.

(10) Edge detection

(a) Specifying valid edge after reset

If the operation of the 16-bit timer/event counter 00 is enabled after reset and while the TI000 or TI010 pin is at high level and when the rising edge or both the edges are specified as the valid edge of the TI000 or TI010 pin, then the high level of the TI000 or TI010 pin is detected as the rising edge. Note this when the TI000 or TI010 pin is pulled up. However, the rising edge is not detected when the operation is once stopped and then enabled again.

(b) Sampling clock for eliminating noise

The sampling clock for eliminating noise differs depending on whether the valid edge of TI000 is used as the count clock or capture trigger. In the former case, the sampling clock is fixed to f_{PRS}. In the latter, the count clock selected by PRM00 is used for sampling.

When the signal input to the TI000 pin is sampled and the valid level is detected two times in a row, the valid edge is detected. Therefore, noise having a short pulse width can be eliminated (refer to **Figure 6-7**).

(11) Timer operation

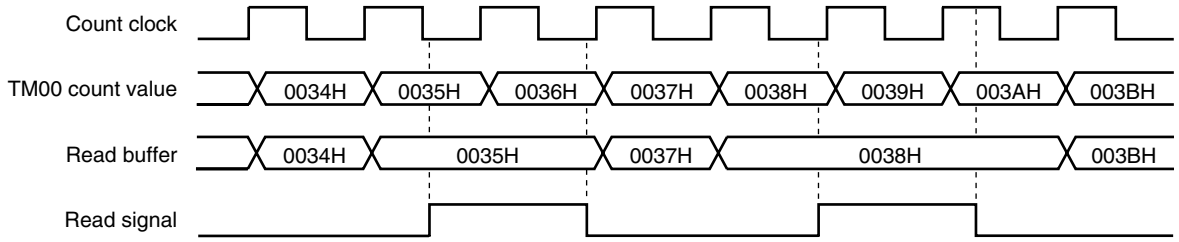
The signal input to the TI000/TI010 pin is not acknowledged while the timer is stopped, regardless of the operation mode of the CPU.

Remark f_{PRS}: Peripheral hardware clock frequency

(12) Reading of 16-bit timer counter 00 (TM00)

TM00 can be read without stopping the actual counter, because the count values captured to the buffer are fixed when it is read. The buffer, however, may not be updated when it is read immediately before the counter counts up, because the buffer is updated at the timing the counter counts up.

Figure 6-58. 16-bit Timer Counter 00 (TM00) Read Timing



CHAPTER 7 8-BIT TIMER/EVENT COUNTER 51

7.1 Functions of 8-Bit Timer/Event Counter 51

8-bit timer/event counter 51 has the following functions.

(1) Interval timer
(2) External event counter
(3) Square-wave output
(4) PWM output

7.2 Configuration of 8-Bit Timer/Event Counter 51

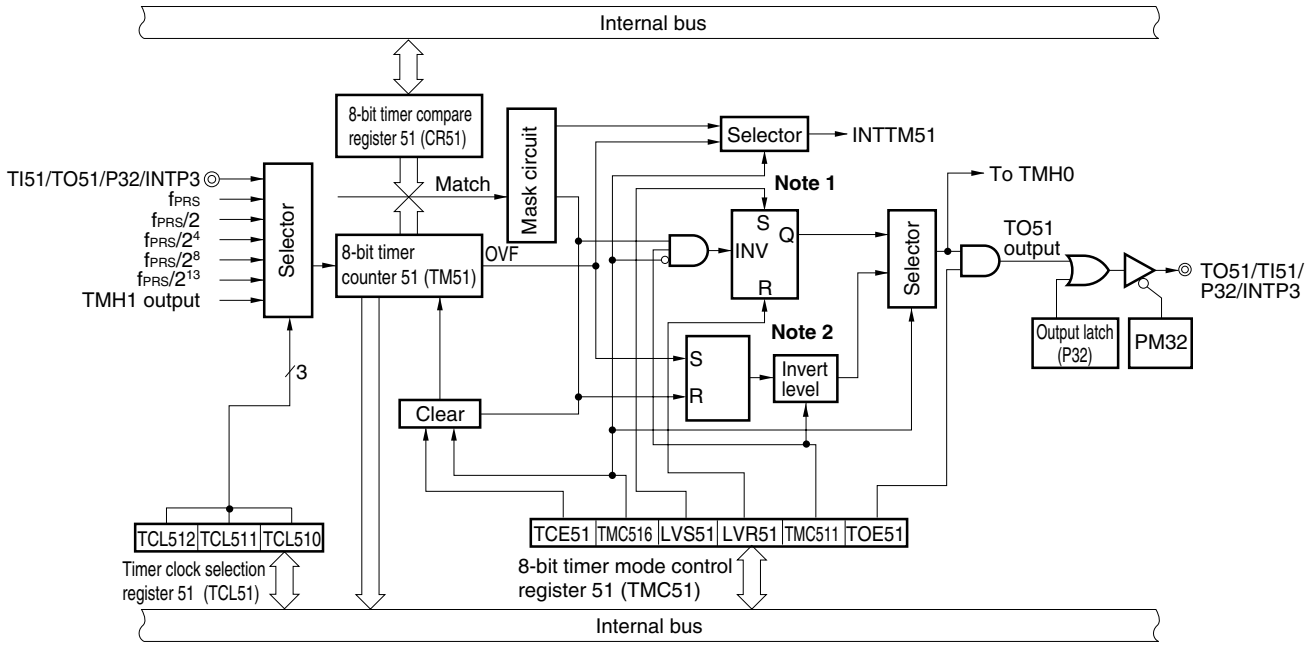
8-bit timer/event counter 51 includes the following hardware.

Table 7-1. Configuration of 8-Bit Timer/Event Counter 51

Item	Configuration
Timer register	8-bit timer counter 51 (TM51)
Timer input	TI51
Register	8-bit timer compare register 51 (CR51)
Control registers	Timer clock selection register 51 (TCL51) 8-bit timer mode control register 51 (TMC51) Port mode register 3 (PM3) Port register 3 (P3)

Figure 7-1 shows the block diagrams of 8-bit timer/event counter 51.

Figure 7-1. Block Diagram of 8-Bit Timer 51



(1) 8-bit timer counter 51 (TM51)

TM51 is an 8-bit register that counts the count pulses and is read-only.

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 7-2. Format of 8-Bit Timer Counter 51 (TM51)

Address: FF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TM51								

In the following situations, the count value is cleared to 00H.

<1> Reset signal generation

<2> When TCE51 is cleared

<3> When TM51 and CR51 match in the mode in which clear & start occurs upon a match of the TM51 and CR51.

(2) 8-bit timer compare register 51 (CR51)

CR51 can be read and written by an 8-bit memory manipulation instruction.

The value set in CR51 is constantly compared with the 8-bit timer counter 51 (TM51) count value, and an interrupt request (INTTM51) is generated if they match.

The value of CR51 can be set within 00H to FFH.

Reset signal generation clears CR51 to 00H.

Figure 7-3. Format of 8-Bit Timer Compare Register 51 (CR51)

Address: FFB1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR51								

Caution In PWM mode, make the CR51 rewrite period 3 count clocks of the count clock (clock selected by TCL51) or more.

7.3 Registers Controlling 8-Bit Timer/Event Counter 51

The following four registers are used to control 8-bit timer/event counter 51.

- Timer clock selection register 51 (TCL51)
- 8-bit timer mode control register 51 (TMC51)
- Port mode register 3 (PM3)
- Port register 3 (P3)

(1) Timer clock selection register 51 (TCL51)

This register sets the count clock of 8-bit timer/event counter 51 and the valid edge of the TI51 pin input.

TCL51 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TCL51 to 00H.

Figure 7-4. Format of Timer Clock Selection Register 51 (TCL51)

Address: FFB2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count clock selection			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	TI51 pin falling edge[Note]			
0	0	1	TI51 pin rising edge[Note]			
0	1	0	f_{PRS}	2 MHz	5 MHz	10 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz
1	0	1	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz
1	1	0	$f_{PRS}/2^{13}$	0.244 kHz	0.61 kHz	1.22 kHz
1	1	1	TMH1 output			

Note Do not start timer operation with the external clock from the TI51 pin when in the STOP mode.

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

2. Be sure to clear bits 3 to 7 to "0".

Remark f_{PRS}: Peripheral hardware clock frequency

(2) 8-bit timer mode control register 51 (TMC51)

TMC51 is a register that performs 8-bit timer counter 51 (TM51) count operation control.

TMC51 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-5. Format of 8-Bit Timer Mode Control Register 51 (TMC51)

Address: FFB3H After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC51	TCE51	TMC516	0	0	LVS51	LVR51	TMC511	TOE51

TCE51	TM51 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC516	TM51 operating mode selection
0	Mode in which clear & start occurs on a match between TM51 and CR51
1	PWM (free-running) mode

LVS51	LVR51	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F clear (0) (default value of TO51 output: low)
1	0	Timer output F/F set (1) (default value of TO51 output: high)
1	1	Setting prohibited

TMC511	In other modes (TMC516 = 0)	In PWM mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE51	Timer output control
0	Output disabled (TO51 output is low level)
1	Output enabled

Cautions 1. Perform <1> to <2> below in the following order, not at the same time.

<1> Set TMC511, TMC516: Operation mode setting

<2> Set TCE51

2. When TCE51 = 1, setting the other bits of TMC51 is prohibited.

Remark In PWM mode, PWM output is made inactive by clearing TCE51 to 0.

(4) Port mode registers 3 (PM3)

This register sets port 3 input/output in 1-bit units.

PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

When using the P32/TI51/TO51/INTP3 pin for timer input, set PM30 to 1. The output latches of P32 at this time may be 0 or 1.

Figure 7-6. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 Operations of 8-Bit Timer/Event Counter 51

7.4.1 Operation as interval timer

8-bit timer/event counter 51 operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 51 (CR51).

When the count value of 8-bit timer counter 51 (TM51) matches the value set to CR51, counting continues with the TM51 value cleared to 0 and an interrupt request signal (INTTM51) is generated.

The count clock of TM51 can be selected with bits 0 to 2 (TCL510 to TCL512) of timer clock selection register 51 (TCL51).

Setting

<1> Set the registers.
- TCL51: Select the count clock.
- CR51: Compare value
- TMC51: Stop the count operation, select the mode in which clear & start occurs on a match of TM51 and CR51.
  (TMC51 = 0000×××0B × = Don't care)

<2> After TCE51 = 1 is set, the count operation starts.

<3> If the values of TM51 and CR51 match, INTTM51 is generated (TM51 is cleared to 00H).

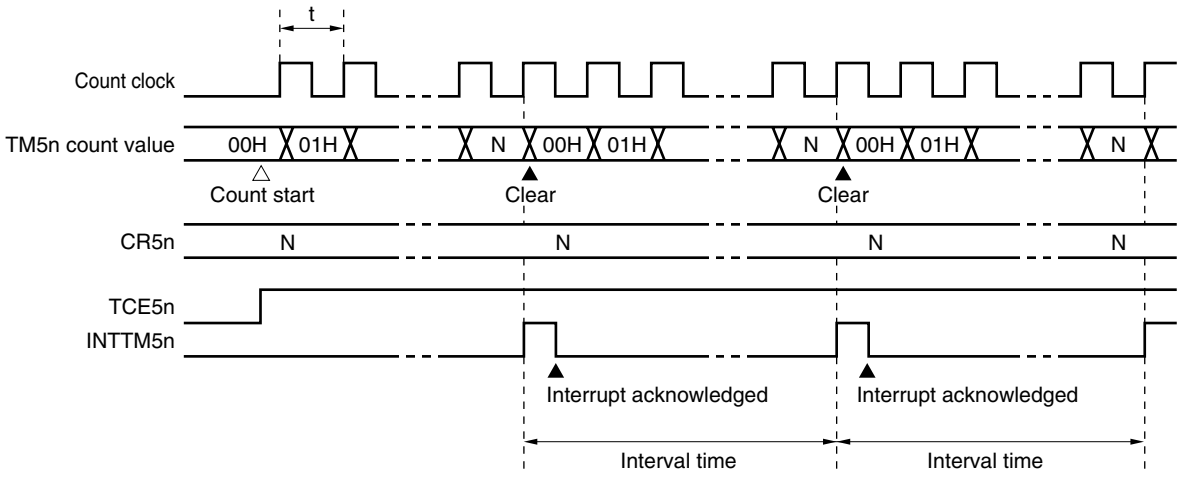
<4> INTTM51 is generated repeatedly at the same interval.
Set TCE51 to 0 to stop the count operation.

Caution Do not write other values to CR51 during operation.

Remark For how to enable the INTTM51 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

Figure 7-7. Interval Timer Operation Timing (1/2)

(a) Basic operation

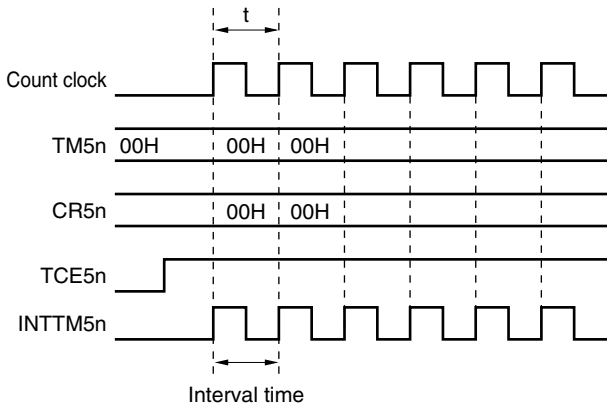


Remarks 1. Interval time = (N + 1) × t, N = 01H to FFH

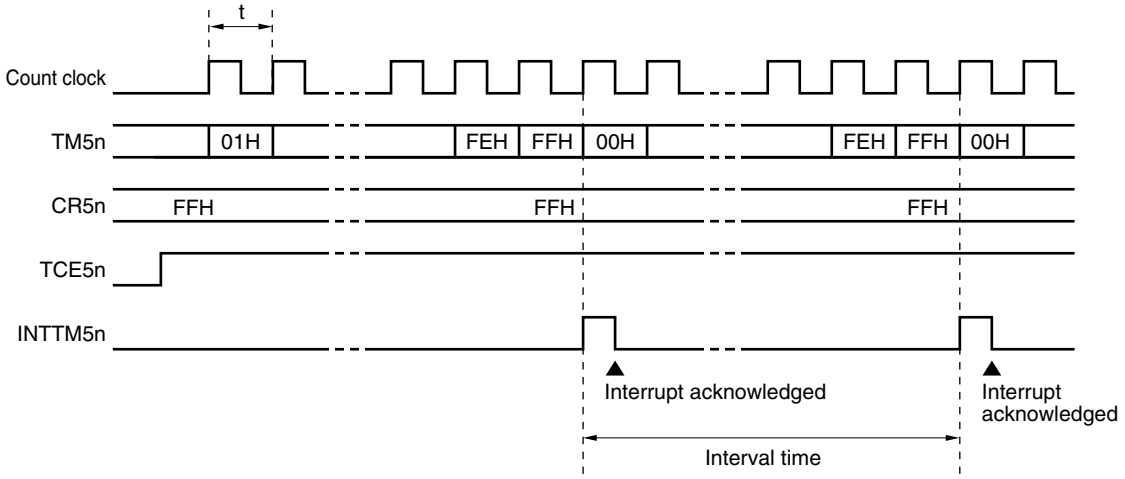
2. n = 1

Figure 7-7. Interval Timer Operation Timing (2/2)

(b) When CR51 = 00H



(c) When CR51 = FFH



Remark n = 1

7.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses to be input to the TI51 pin by 8-bit timer counter 51 (TM51).

TM51 is incremented each time the valid edge specified by timer clock selection register 51 (TCL51) is input. Either the rising or falling edge can be selected.

When the TM51 count value matches the value of 8-bit timer compare register 51 (CR51), TM51 is cleared to 0 and an interrupt request signal (INTTM51) is generated.

Whenever the TM51 value matches the value of CR51, INTTM51 is generated.

Setting

<1> Set each register.

- Set the port mode register (PM30) to 1.
- TCL51: Select TI51 pin input edge.
  TI51 pin falling edge → TCL51 = 00H
  TI51 pin rising edge → TCL51 = 01H
- CR51: Compare value
- TMC51: Stop the count operation, select the mode in which clear & start occurs on match of TM51 and CR51, disable the timer F/F inversion operation, disable timer output.
  (TMC51 = 00000000B)

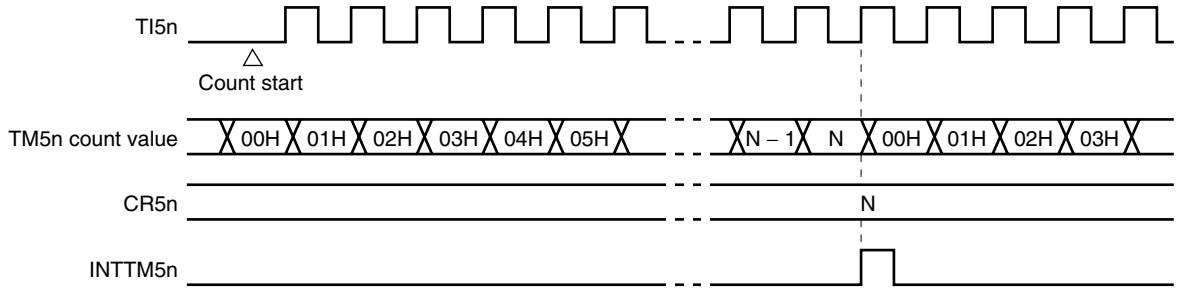
<2> When TCE51 = 1 is set, the number of pulses input from the TI51 pin is counted.

<3> When the values of TM51 and CR51 match, INTTM51 is generated (TM51 is cleared to 00H).

<4> After these settings, INTTM51 is generated each time the values of TM51 and CR51 match.

Remark For how to enable the INTTM51 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

Figure 7-8. External Event Counter Operation Timing (with Rising Edge Specified)



Remarks 1. N = 00H to FFH

2. n = 1

7.4.3 Square-wave output operation

A square wave with any selected frequency is output at intervals determined by the value preset to 8-bit timer compare register 51 (CR51).

The TO51 pin output status is inverted at intervals determined by the count value preset to CR51 by setting bit 0 (TOE51) of 8-bit timer mode control register 51 (TMC51) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

Setting

<1> Set each register.

- Clear the port output latch (P30) and port mode register (PM30) to 0.
- TCL51: Select the count clock.
- CR51: Compare value
- TMC51: Stop the count operation, select the mode in which clear & start occurs on a match of TM51 and CR51. (TMC5n = 00001011B or 00000111B)

<2> After TCE51 = 1 is set, the count operation starts.

<3> The timer output F/F is inverted by a match of TM51 and CR51. After INTTM51 is generated, TM51 is cleared to 00H.

<4> After these settings, the timer output F/F is inverted at the same interval and a square wave is output from TO5n. The frequency is as follows.

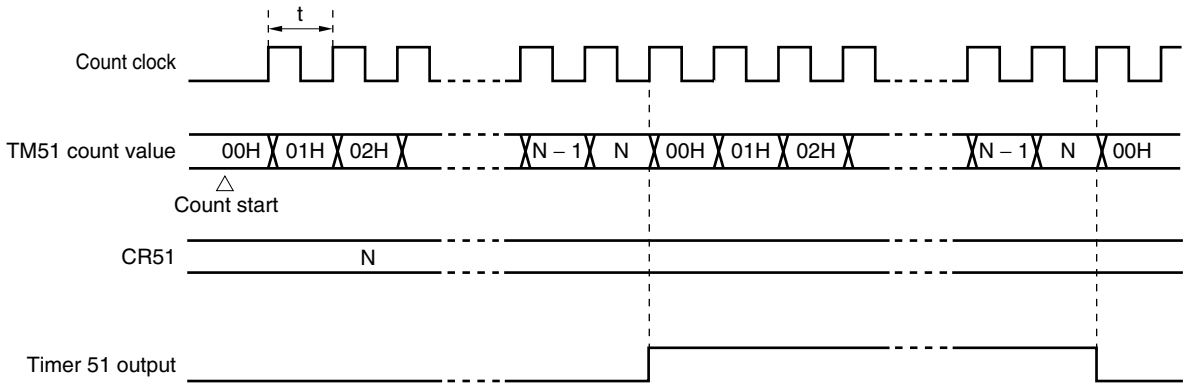
- Frequency = 1/2t (N + 1)

  (N: 00H to FFH)

Caution Do not write other values to CR51 during operation.

Remark For how to enable the INTTM51 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

Figure 7-9. Square-Wave Output Operation Timing



7.4.4 PWM output operation

8-bit timer/event counter 51 operates as a PWM output when bit 6 (TMC516) of 8-bit timer mode control register 51 (TMC51) is set to 1.

The duty pulse determined by the value set to 8-bit timer compare register 51 (CR51) is output from TO51.

Set the active level width of the PWM pulse to CR51; the active level can be selected with bit 1 (TMC511) of TMC51.

The count clock can be selected with bits 0 to 2 (TCL510 to TCL512) of timer clock selection register 51 (TCL51).

PWM output can be enabled/disabled with bit 0 (TOE51) of TMC51.

Caution In PWM mode, make the CR51 rewrite period 3 count clocks of the count clock (clock selected by TCL51) or more.

(1) PWM output basic operation

Setting

<1> Set each register.

- Clear the port output latch (P30) and port mode register (PM30) to 0.
- TCL51: Select the count clock.
- CR51: Compare value
- TMC51: Stop the count operation, select PWM mode.

TMC511	Active Level Selection
0	Active-high
1	Active-low

(TMC51 = 01000000B or 01000010B)

<2> The count operation starts when TCE51 = 1.
Clear TCE51 to 0 to stop the count operation.

PWM output operation

<1> PWM output (TO51 output) outputs an inactive level until an overflow occurs.

<2> When an overflow occurs, the active level is output. The active level is output until CR51 matches the count value of 8-bit timer counter 51 (TM51).

<3> After the CR51 matches the count value, the inactive level is output until an overflow occurs again.

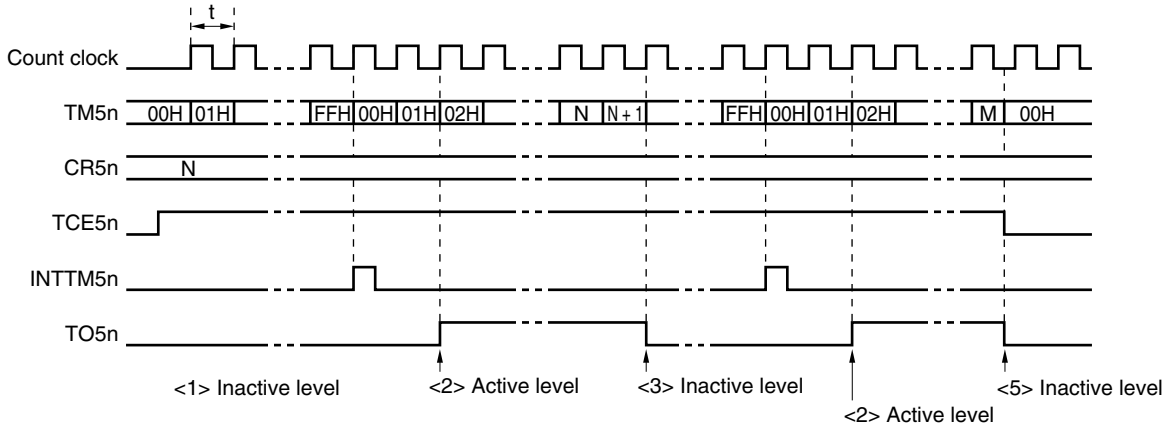
<4> Operations <2> and <3> are repeated until the count operation stops.

<5> When the count operation is stopped with TCE51 = 0, PWM output becomes inactive.
For details of timing, refer to **Figures 7-10** and **7-11**.
The cycle, active-level width, and duty are as follows.

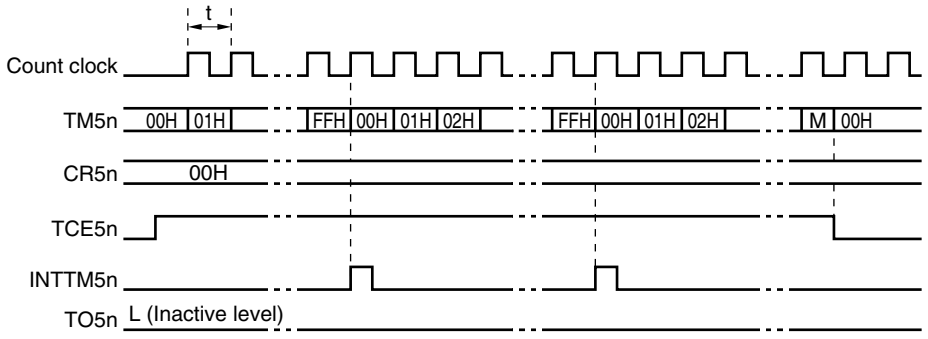
- Cycle = 2^8t
- Active-level width = Nt
- Duty = N/2^8
  (N = 00H to FFH)

Figure 7-10. PWM Output Operation Timing

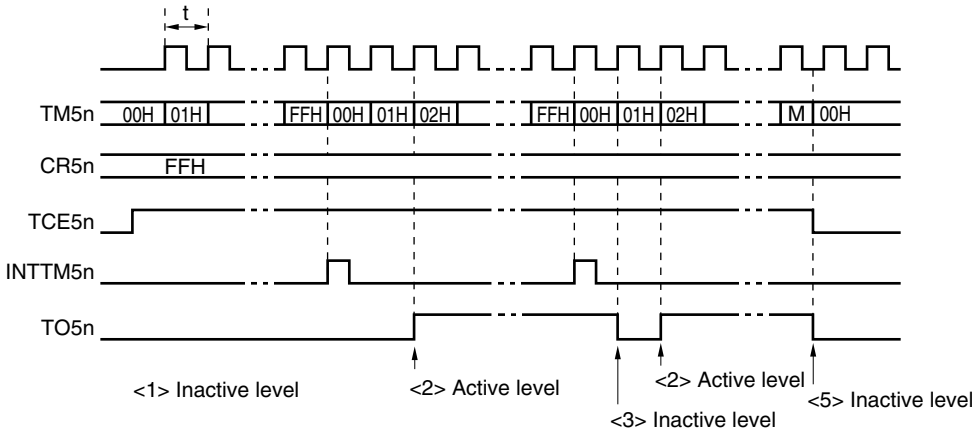
(a) Basic operation (active level = H)



(b) CR5n = 00H



(c) CR5n = FFH



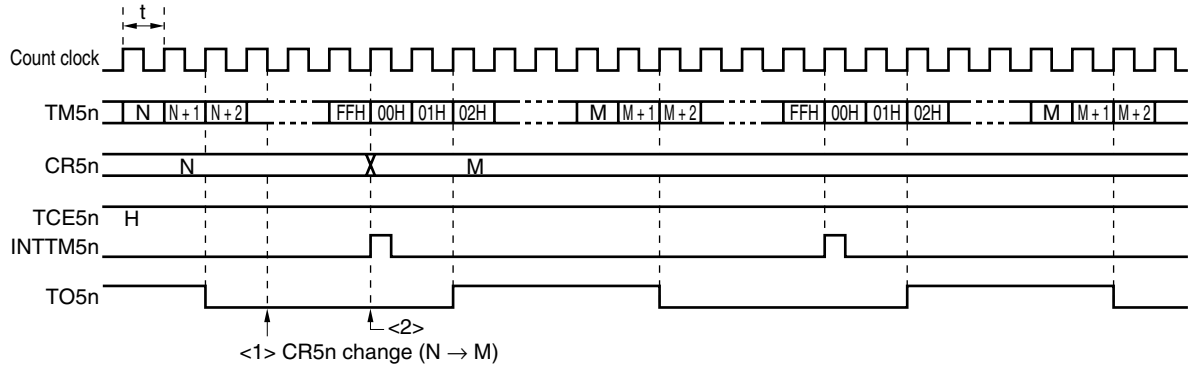
Remarks 1. <1> to <3> and <5> in Figure 7-10 (a) and (c) correspond to <1> to <3> and <5> in PWM output operation in **7.4.4 (1) PWM output basic operation**.

2. n = 1

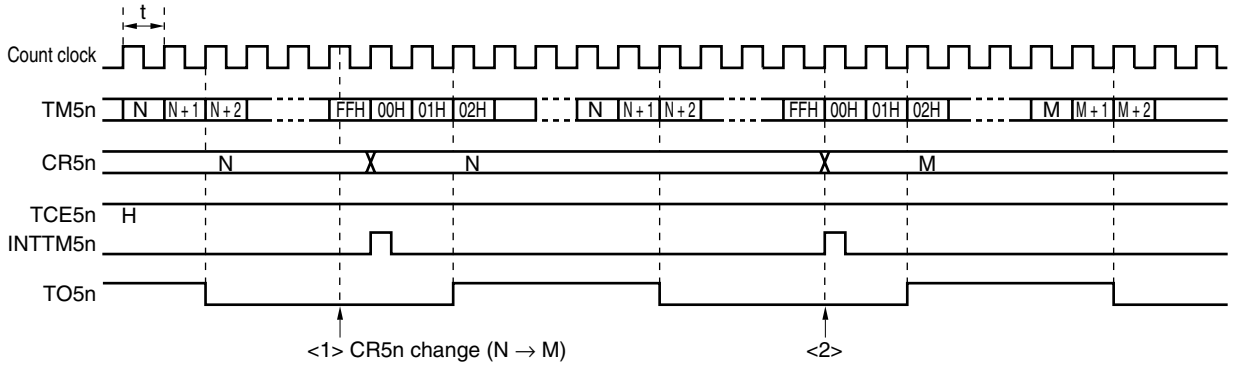
(2) Operation with CR51 changed

Figure 7-11. Timing of Operation with CR51 Changed

(a) CR51 value is changed from N to M before clock rising edge of FFH
→ Value is transferred to CR51 at overflow immediately after change.



(b) CR51 value is changed from N to M after clock rising edge of FFH
→ Value is transferred to CR51 at second overflow.



Caution When reading from CR51 between <1> and <2> in Figure 7-11, the value read differs from the actual value (read value: M, actual value of CR51: N).

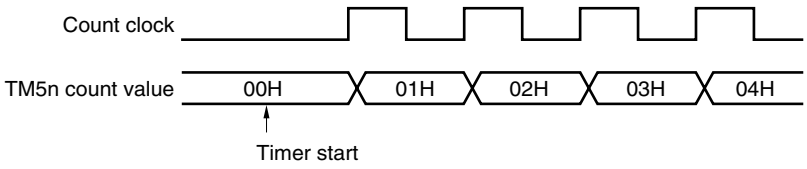
Remark n = 1

7.5 Cautions for 8-Bit Timer/Event Counter 51

(1) Timer start error

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counter 51 (TM51) is started asynchronously to the count clock.

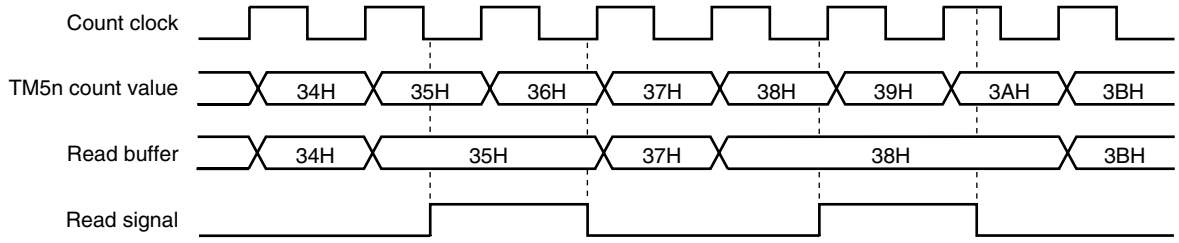
Figure 7-12. 8-Bit Timer Counter 51 (TM51) Start Timing



(2) Reading of 8-bit timer counter 51 (TM51)

TM51 can be read without stopping the actual counter, because the count values captured to the buffer are fixed when it is read. The buffer, however, may not be updated when it is read immediately before the counter counts up, because the buffer is updated at the timing the counter counts up.

Figure 7-13. 8-bit Timer Counter 51 (TM51) Read Timing



Remark n = 1

CHAPTER 8 8-BIT TIMERS H0 AND H1

8.1 Functions of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 have the following functions.

- Interval timer
- Square-wave output
- PWM output
- Carrier generator (8-bit timer H1 only)

8.2 Configuration of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 include the following hardware.

Table 8-1. Configuration of 8-Bit Timers H0 and H1

Item	Configuration
Timer register	8-bit timer counter Hn
Registers	8-bit timer H compare register 0n (CMP0n) 8-bit timer H compare register 1n (CMP1n)
Timer output	TOHn, output controller
Control registers	8-bit timer H mode register n (TMHMDn) 8-bit timer H carrier control register 1 (TMCYC1)Note Port mode register 3 (PM3) Port register 3 (P3)

Note 8-bit timer H1 only

Remark n = 0, 1

Figure 8-1. Block Diagram of 8-Bit Timer H0

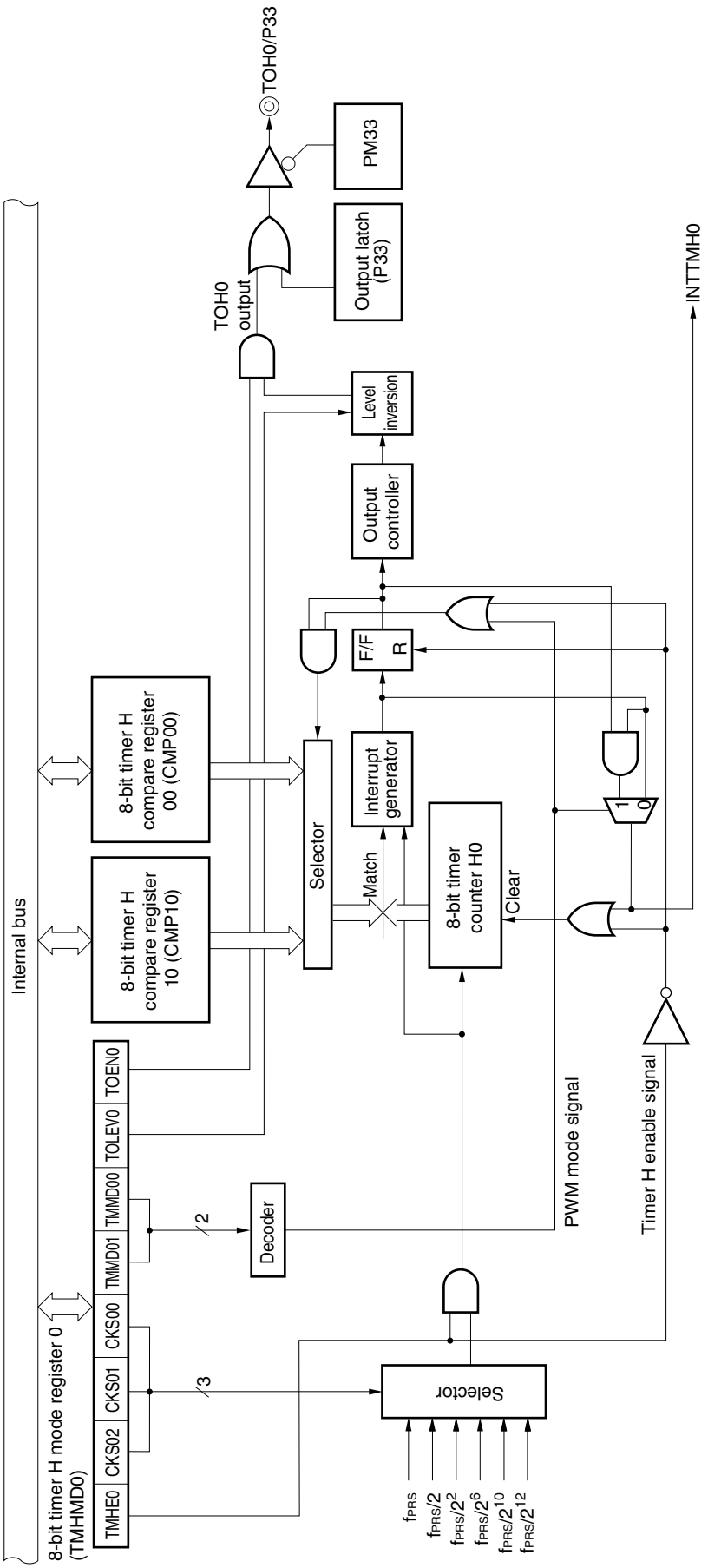
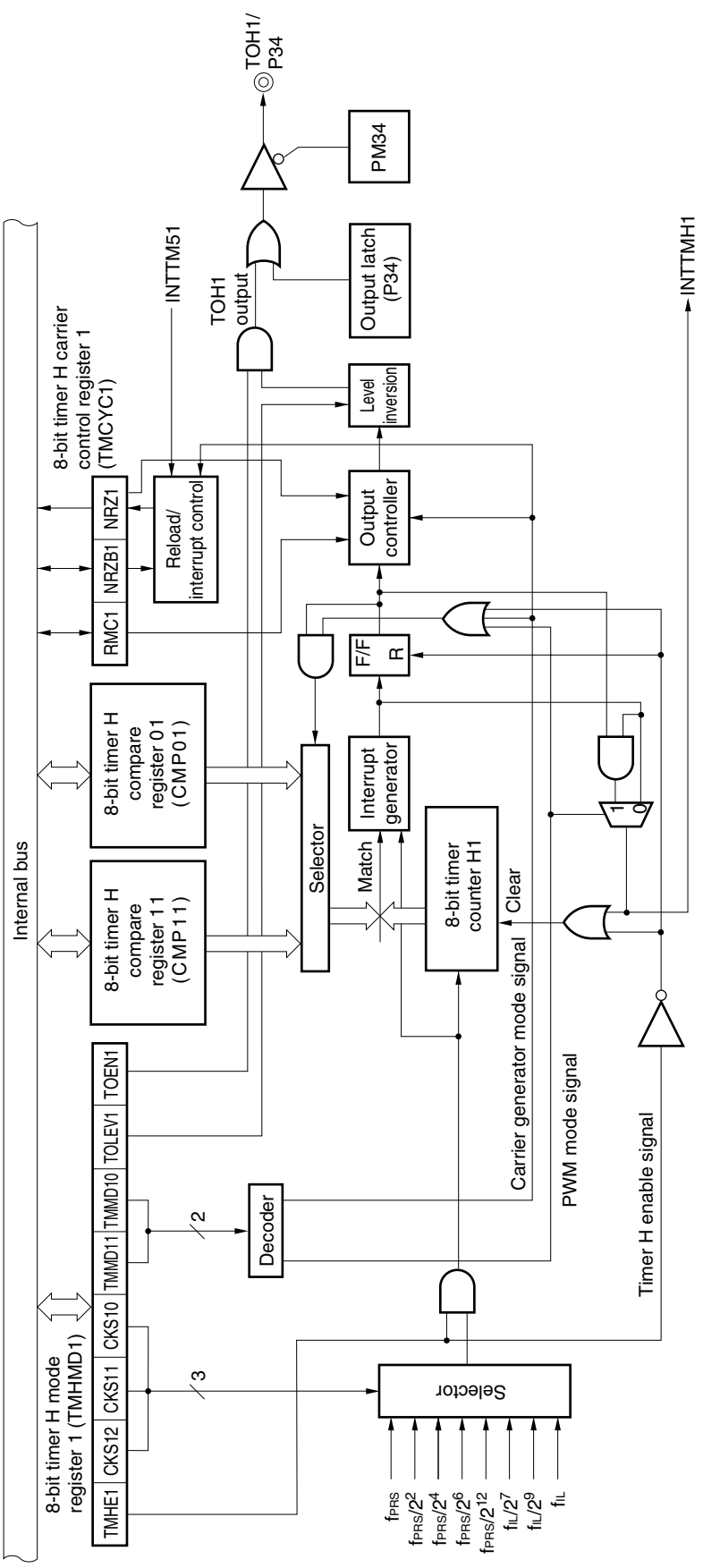


Figure 8-2. Block Diagram of 8-Bit Timer H1



(1) 8-bit timer H compare register 0n (CMP0n)

This register can be read or written by an 8-bit memory manipulation instruction. This register is used in all of the timer operation modes.

This register constantly compares the value set to CMP0n with the count value of the 8-bit timer counter Hn and, when the two values match, generates an interrupt request signal (INTTMHn) and inverts the output level of TOHn.

Rewrite the value of CMP0n while the timer is stopped (TMHEn = 0).

A reset signal generation clears this register to 00H.

Figure 8-3. Format of 8-Bit Timer H Compare Register 0n (CMP0n)

Address: FF18H (CMP00), FF1AH (CMP01) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP0n								

Caution CMP0n cannot be rewritten during timer count operation. CMP0n can be refreshed (the same value is written) during timer count operation.

(2) 8-bit timer H compare register 1n (CMP1n)

This register can be read or written by an 8-bit memory manipulation instruction. This register is used in the PWM output mode and carrier generator mode.

In the PWM output mode, this register constantly compares the value set to CMP1n with the count value of the 8-bit timer counter Hn and, when the two values match, inverts the output level of TOHn. No interrupt request signal is generated.

In the carrier generator mode, the CMP1n register always compares the value set to CMP1n with the count value of the 8-bit timer counter Hn and, when the two values match, generates an interrupt request signal (INTTMHn). At the same time, the count value is cleared.

CMP1n can be refreshed (the same value is written) and rewritten during timer count operation.

If the value of CMP1n is rewritten while the timer is operating, the new value is latched and transferred to CMP1n when the count value of the timer matches the old value of CMP1n, and then the value of CMP1n is changed to the new value. If matching of the count value and the CMP1n value and writing a value to CMP1n conflict, the value of CMP1n is not changed.

A reset signal generation clears this register to 00H.

Figure 8-4. Format of 8-Bit Timer H Compare Register 1n (CMP1n)

Address: FF19H (CMP10), FF1BH (CMP11) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP1n								

Cautions 1. In the PWM output mode and carrier generator mode, be sure to set CMP1n when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to CMP1n).

2. In the interval timer mode, do not use CMP1n.

Remark n = 0, 1

8.3 Registers Controlling 8-Bit Timers H0 and H1

The following five registers are used to control 8-bit timers H0 and H1.

- 8-bit timer H mode register n (TMHMDn)
- 8-bit timer H carrier control register 1 (TMCYC1)[Note]
- Port mode register 3 (PM3)
- Port register 3 (P3)

Note 8-bit timer H1 only

(1) 8-bit timer H mode register n (TMHMDn)

This register controls the mode of timer H.

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

Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 8-5. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)

Address: FF69H After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS02	CKS01	CKS00	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	f_{PRS}	2 MHz	5 MHz	10 MHz
0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz
0	1	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz
1	0	0	$f_{PRS}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz
1	0	1	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz
Other than above			Setting prohibited			

TMMD01	TMMD00	Timer operation mode
0	0	Interval timer mode
1	0	PWM output mode
Other than above		Setting prohibited

TOLEV0	Timer output level control (in default mode)
0	Low level
1	High level

TOEN0	Timer output control
0	Disables output
1	Enables output

Note Note the following points when selecting the TM50 output as the count clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Cautions 1. When TMHE0 = 1, setting the other bits of TMHMD0 is prohibited. However, TMHMD0 can be refreshed (the same value is written).

2. In the PWM output mode, be sure to set the 8-bit timer H compare register 10 (CMP10) when starting the timer count operation (TMHE0 = 1) after the timer count operation was stopped (TMHE0 = 0) (be sure to set again even if setting the same value to CMP10).

3. The actual TOH0/P15 pin output is determined depending on PM33 and P33, besides TOH0 output.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)

3. TMC501: Bit 1 of TMC50

Figure 8-6. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FF6CH After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS12	CKS11	CKS10	Count clock selection			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	f_{PRS}	2 MHz	5 MHz	10 MHz
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz
0	1	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz
0	1	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz
1	0	0	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz
1	0	1	$f_{IL}/2^6$	1.88 kHz (TYP.)		
1	1	0	$f_{IL}/2^{15}$	0.47 kHz (TYP.)		
1	1	1	f_{IL}	240 kHz (TYP.)		

TMMD11	TMMD10	Timer operation mode
0	0	Interval timer mode
0	1	Carrier generator mode
1	0	PWM output mode
1	1	Setting prohibited

TOLEV1	Timer output level control (in default mode)
0	Low level
1	High level

TOEN1	Timer output control
0	Disables output
1	Enables output

Cautions 1. When TMHE1 = 1, setting the other bits of TMHMD1 is prohibited. However, TMHMD1 can be refreshed (the same value is written).

2. In the PWM output mode and carrier generator mode, be sure to set the 8-bit timer H compare register 11 (CMP11) when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to CMP11).

3. When the carrier generator mode is used, set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

4. The actual TOH1/P34 pin output is determined depending on PM34 and P34, besides TOH1 output.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

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

(2) 8-bit timer H carrier control register 1 (TMCYC1)

This register controls the remote control output and carrier pulse output status of 8-bit timer H1.

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

Reset signal generation clears this register to 00H.

Figure 8-7. Format of 8-Bit Timer H Carrier Control Register 1 (TMCYC1)

Address: FF6DH After reset: 00H R/W[Note]

	7	6	5	4	3	2	1	<0>
TMCYC1	0	0	0	0	0	RMC1	NRZB1	NRZ1

RMC1	NRZB1	Remote control output
0	0	Low-level output
0	1	High-level output at rising edge of INTTM51 signal input
1	0	Low-level output
1	1	Carrier pulse output at rising edge of INTTM51 signal input

NRZ1	Carrier pulse output status flag
0	Carrier output disabled status (low-level status)
1	Carrier output enabled status (RMC1 = 1: Carrier pulse output, RMC1 = 0: High-level status)

Note Bit 0 is read-only.

Caution Do not rewrite RMC1 when TMHE = 1. However, TMCYC1 can be refreshed (the same value is written).

(3) Port mode register 3 (PM3)

This register sets port 0 input/output, port 1 input/output, and port 3 input/output in 1-bit units.

When using the P33/TOH0 pin for timer output, clear PM33 and the output latches of P33 to 0.

When using the P00/TOH1 pin for timer output, clear PM34 and the output latches of P34 to 0.

PM0, PM1, and PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 8-8. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Operation of 8-Bit Timers H0 and H1

8.4.1 Operation as interval timer/square-wave output

When the 8-bit timer counter Hn and compare register 0n (CMP0n) match, an interrupt request signal (INTTMHn) is generated and the 8-bit timer counter Hn is cleared to 00H.

Compare register 1n (CMP1n) is not used in interval timer mode. Since a match of the 8-bit timer counter Hn and the CMP1n register is not detected even if the CMP1n register is set, timer output is not affected.

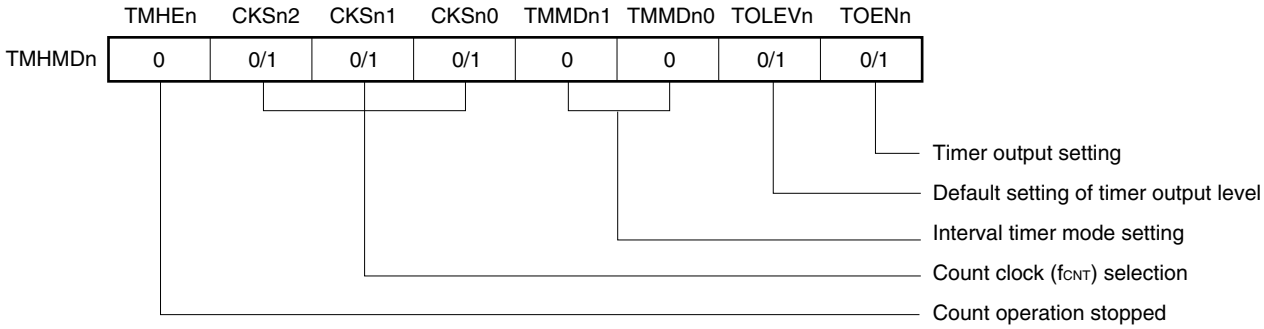
By setting bit 0 (TOENn) of timer H mode register n (TMHMDn) to 1, a square wave of any frequency (duty = 50%) is output from TOHn.

Setting

<1> Set each register.

Figure 8-9. Register Setting During Interval Timer/Square-Wave Output Operation

(i) Setting timer H mode register n (TMHMDn)



(ii) CMP0n register setting

The interval time is as follows if N is set as a comparison value.

- Interval time = (N +1)/f_{CNT}

<2> Count operation starts when TMHEn = 1.

<3> When the values of the 8-bit timer counter Hn and the CMP0n register match, the INTTMHn signal is generated and the 8-bit timer counter Hn is cleared to 00H.

<4> Subsequently, the INTTMHn signal is generated at the same interval. To stop the count operation, clear TMHEn to 0.

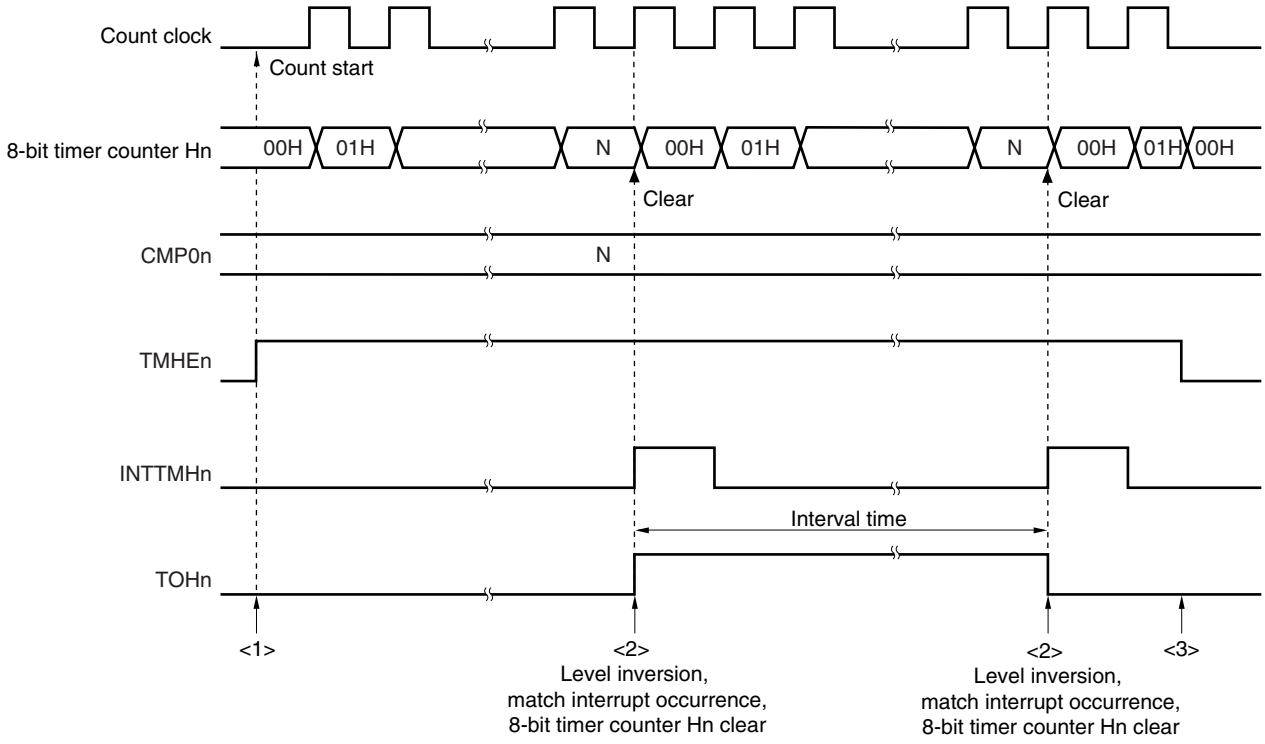
Remarks 1. For the setting of the output pin, refer to **8.3 (3) Port mode register 3 (PM3)**.

2. For how to enable the INTTMHn signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

3. n = 0, 1

Figure 8-10. Timing of Interval Timer/Square-Wave Output Operation (1/2)

(a) Basic operation (Operation When 01H ≤ CMP0n ≤ FEH)



<1> The count operation is enabled by setting the TMHEn bit to 1. The count clock starts counting no more than 1 clock after the operation is enabled.

<2> When the value of the 8-bit timer counter Hn matches the value of the CMP0n register, the value of the timer counter is cleared, and the level of the TOHn output is inverted. In addition, the INTTMHn signal is output at the rising edge of the count clock.

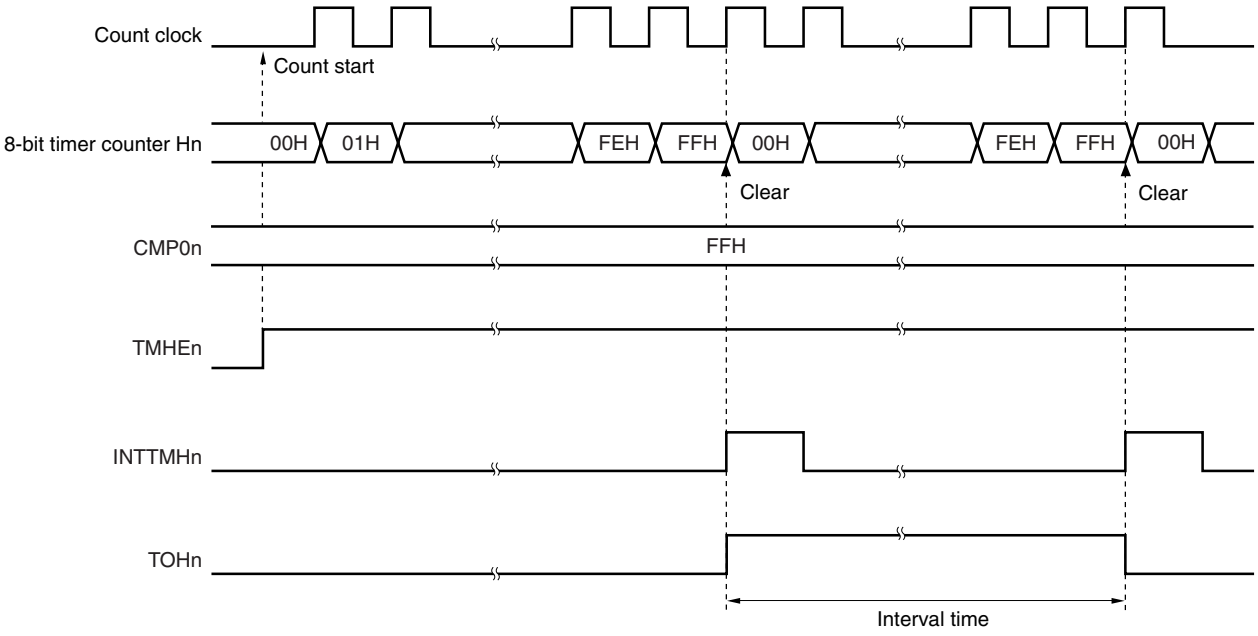
<3> If the TMHEn bit is cleared to 0 while timer H is operating, the INTTMHn signal and TOHn output are set to the default level. If they are already at the default level before the TMHEn bit is cleared to 0, then that level is maintained.

Remarks 1. 01H ≤ N ≤ FEH

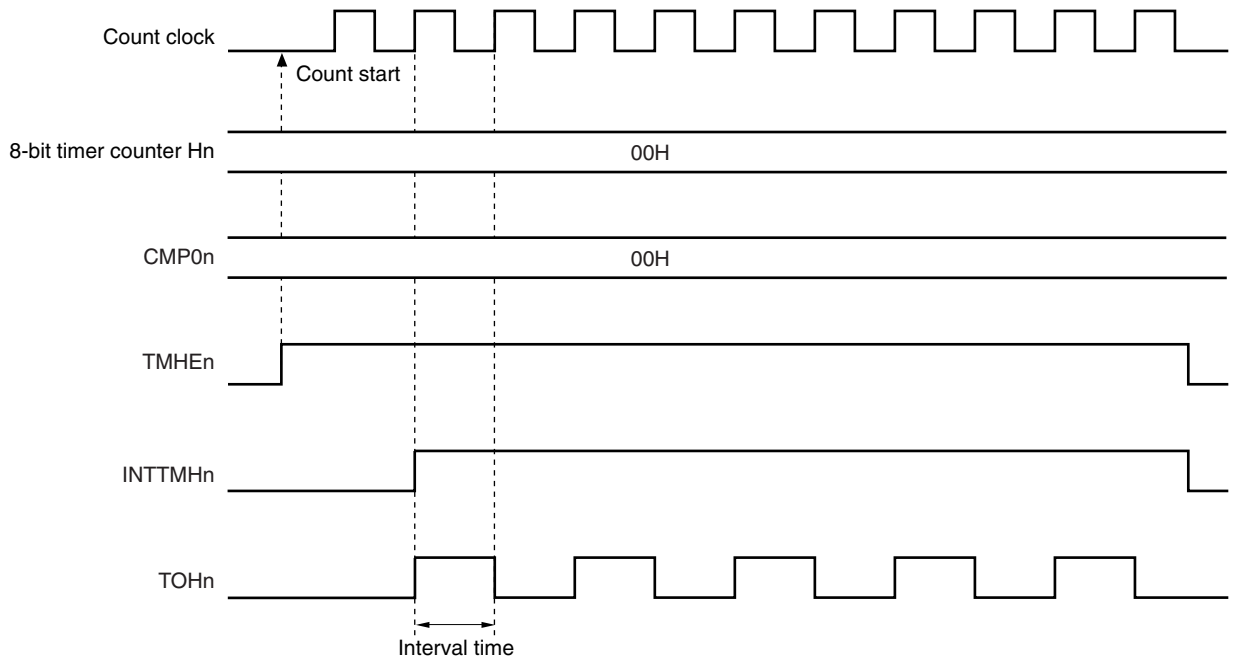
2. n = 0, 1

Figure 8-10. Timing of Interval Timer/Square-Wave Output Operation (2/2)

(b) Operation when CMP0n = FFH



(c) Operation when CMP0n = 00H



Remark n = 0, 1

8.4.2 Operation as PWM output

In PWM output mode, a pulse with an arbitrary duty and arbitrary cycle can be output.

The 8-bit timer compare register 0n (CMP0n) controls the cycle of timer output (TOHn). Rewriting the CMP0n register during timer operation is prohibited.

The 8-bit timer compare register 1n (CMP1n) controls the duty of timer output (TOHn). Rewriting the CMP1n register during timer operation is possible.

The operation in PWM output mode is as follows.

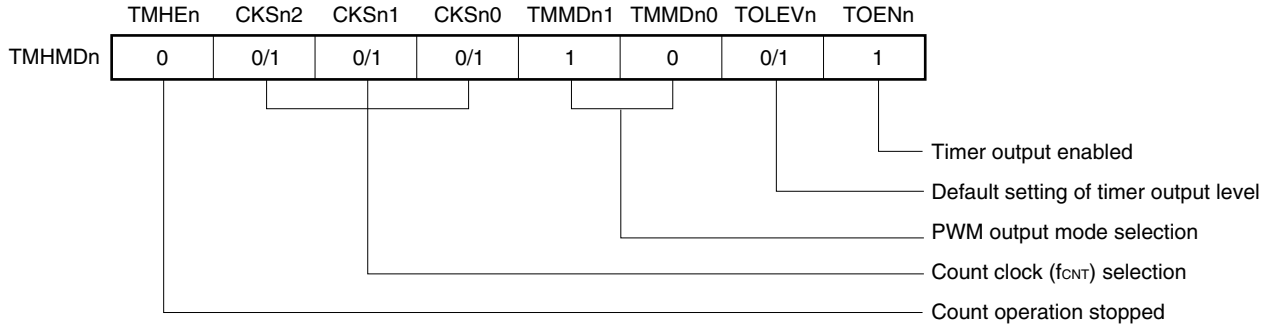
PWM output (TOHn output) outputs an active level and 8-bit timer counter Hn is cleared to 0 when 8-bit timer counter Hn and the CMP0n register match after the timer count is started. PWM output (TOHn output) outputs an inactive level when 8-bit timer counter Hn and the CMP1n register match.

Setting

<1> Set each register.

Figure 8-11. Register Setting in PWM Output Mode

(i) Setting timer H mode register n (TMHMDn)



(ii) Setting CMP0n register

- Compare value (N): Cycle setting

(iii) Setting CMP1n register

- Compare value (M): Duty setting

Remarks 1. n = 0, 1

2. 00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

<2> The count operation starts when TMHEn = 1.

<3> The CMP0n register is the compare register that is to be compared first after counter operation is enabled. When the values of the 8-bit timer counter Hn and the CMP0n register match, the 8-bit timer counter Hn is cleared, an interrupt request signal (INTTMHn) is generated, and an active level is output. At the same time, the compare register to be compared with the 8-bit timer counter Hn is changed from the CMP0n register to the CMP1n register.

<4> When the 8-bit timer counter Hn and the CMP1n register match, an inactive level is output and the compare register to be compared with the 8-bit timer counter Hn is changed from the CMP1n register to the CMP0n register. At this time, the 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty can be obtained.

<6> To stop the count operation, set TMHEn = 0.

If the setting value of the CMP0n register is N, the setting value of the CMP1n register is M, and the count clock frequency is f_{CNT}, the PWM pulse output cycle and duty are as follows.

- PWM pulse output cycle = (N + 1)/f_{CNT}
- Duty = (M + 1)/(N + 1)

Cautions 1. The set value of the CMP1n register can be changed while the timer counter is operating. However, this takes a duration of three operating clocks (signal selected by the CKSn2 to CKSn0 bits of the TMHMDn register) from when the value of the CMP1n register is changed until the value is transferred to the register.

2. Be sure to set the CMP1n register when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to the CMP1n register).

3. Make sure that the CMP1n register setting value (M) and CMP0n register setting value (N) are within the following range.
00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

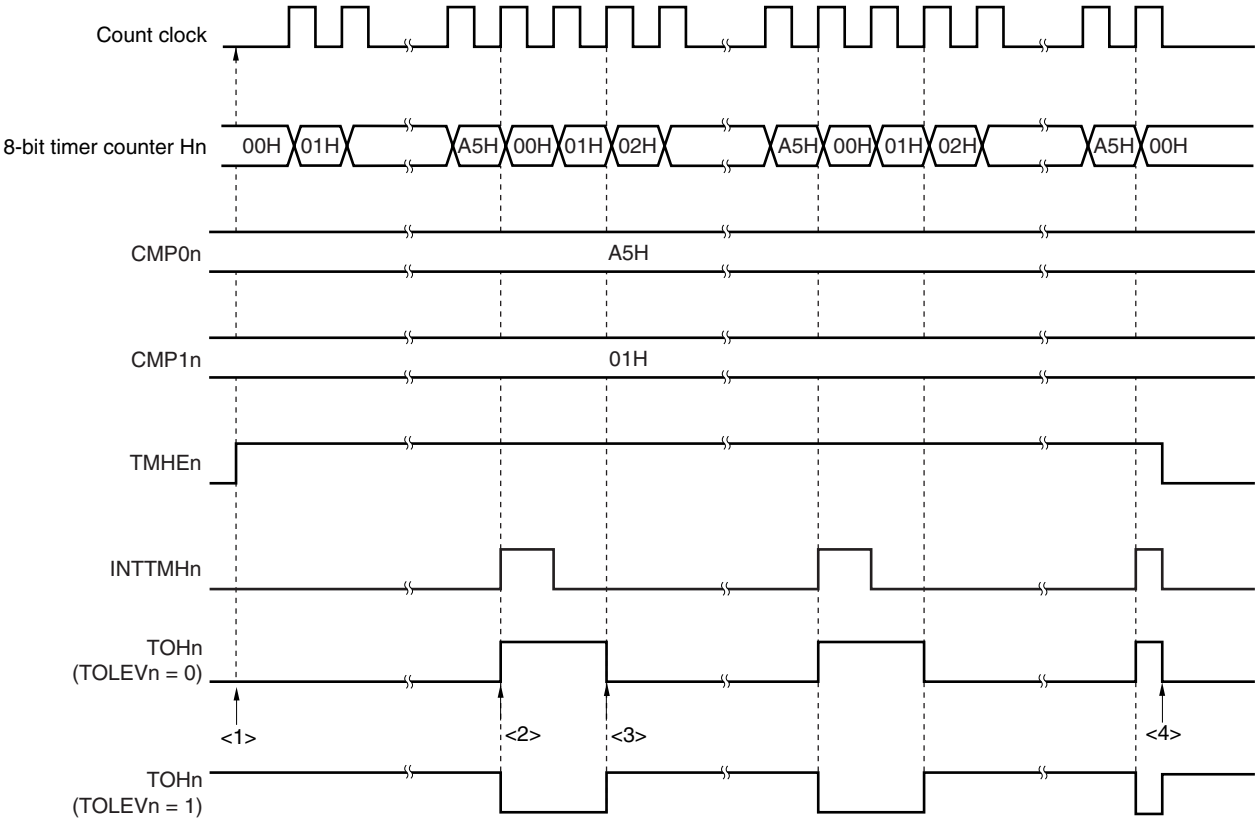
Remarks 1. For the setting of the output pin, refer to **8.3 (3) Port mode register 3 (PM3)**.

2. For details on how to enable the INTTMHn signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

3. n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (1/4)

(a) Basic operation



<1> The count operation is enabled by setting the TMHEn bit to 1. Start the 8-bit timer counter Hn by masking one count clock to count up. At this time, PWM output outputs an inactive level.

<2> When the values of the 8-bit timer counter Hn and the CMP0n register match, an active level is output. At this time, the value of the 8-bit timer counter Hn is cleared, and the INTTMHn signal is output.

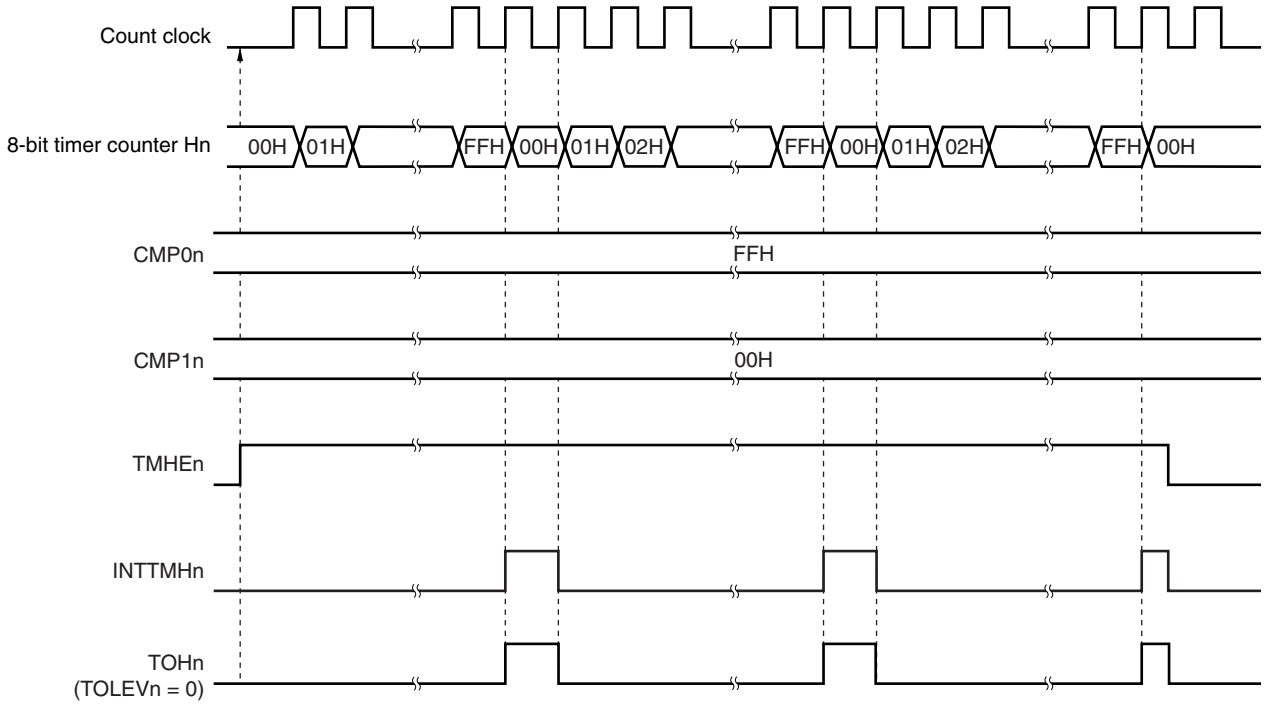
<3> When the values of the 8-bit timer counter Hn and the CMP1n register match, an inactive level is output. At this time, the 8-bit timer counter value is not cleared and the INTTMHn signal is not output.

<4> Clearing the TMHEn bit to 0 during timer Hn operation sets the INTTMHn signal to the default and PWM output to an inactive level.

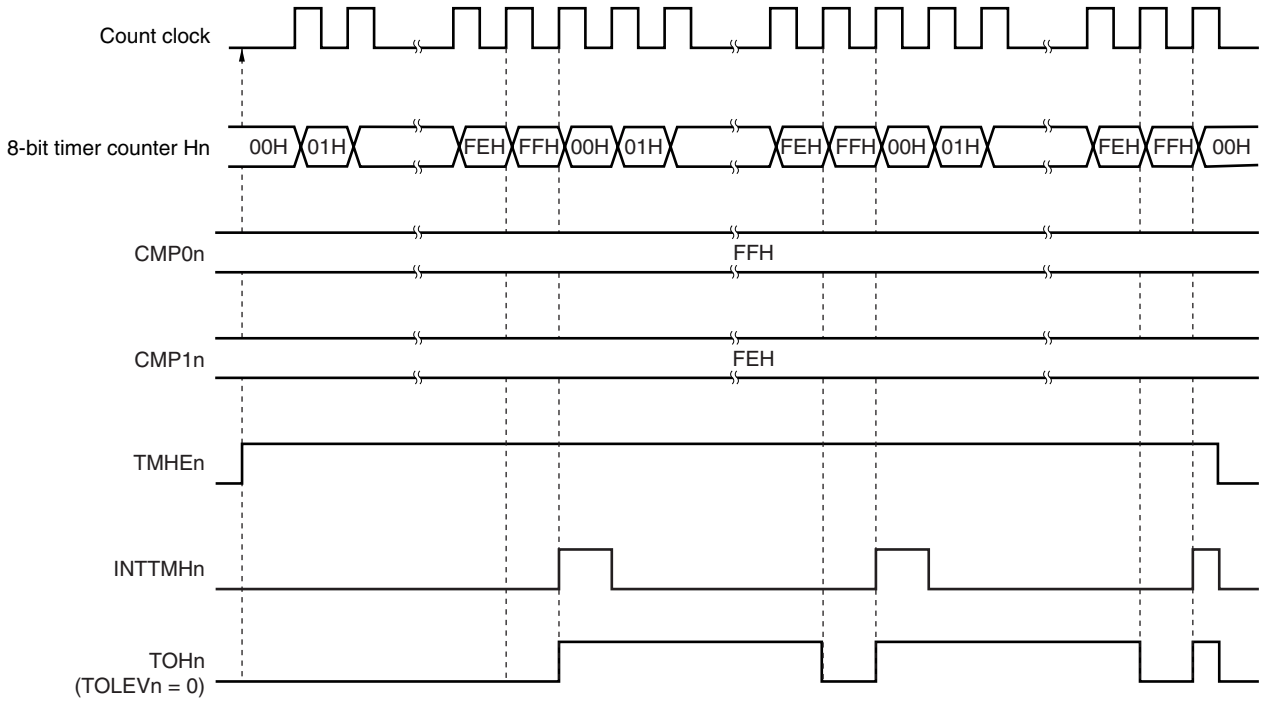
Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (2/4)

(b) Operation when CMP0n = FFH, CMP1n = 00H



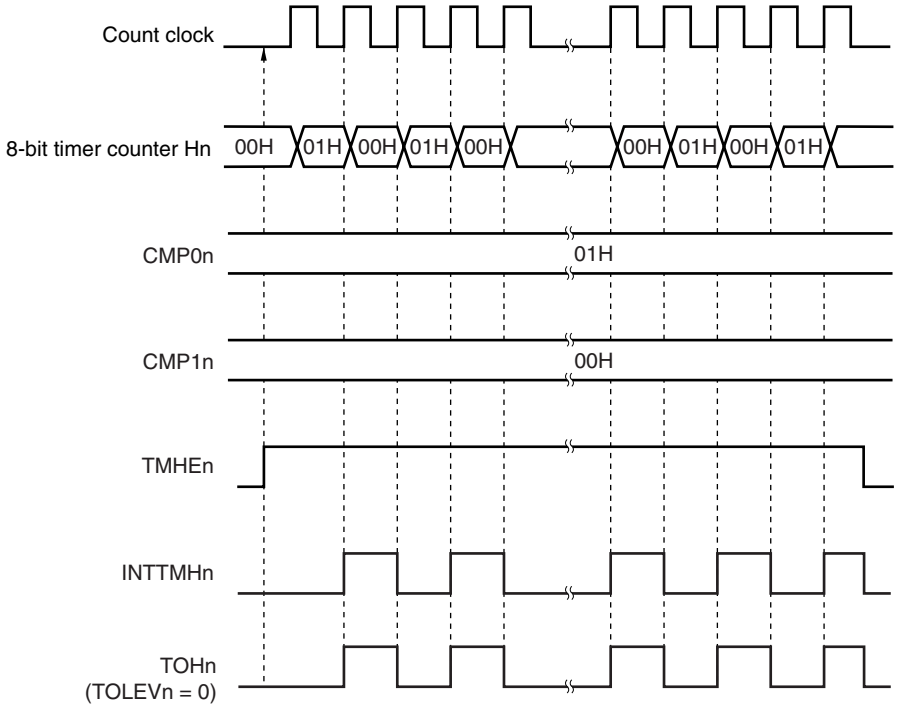
(c) Operation when CMP0n = FFH, CMP1n = FEH



Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (3/4)

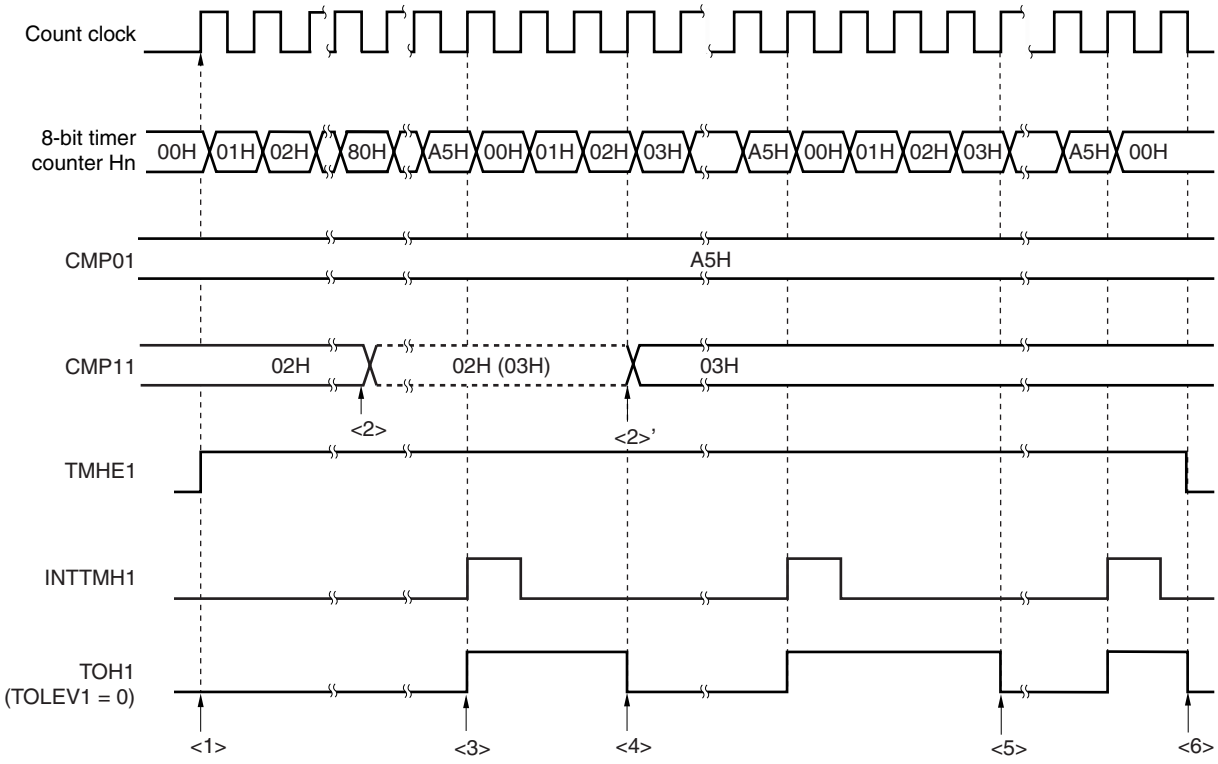
(d) Operation when CMP0n = 01H, CMP1n = 00H



Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (4/4)

(e) Operation by changing CMP1n (CMP1n = 02H → 03H, CMP0n = A5H)



<1> The count operation is enabled by setting TMHEn = 1. Start the 8-bit timer counter Hn by masking one count clock to count up. At this time, PWM output outputs an inactive level.

<2> The CMP1n register value can be changed during timer counter operation. This operation is asynchronous to the count clock.

<3> When the values of the 8-bit timer counter Hn and the CMP0n register match, the value of the 8-bit timer counter Hn is cleared, an active level is output, and the INTTMHn signal is output.

<4> If the CMP1n register value is changed, the value is latched and not transferred to the register. When the values of the 8-bit timer counter Hn and the CMP1n register before the change match, the value is transferred to the CMP1n register and the CMP1n register value is changed (<2>').

However, three count clocks or more are required from when the CMP1n register value is changed to when the value is transferred to the register. If a match signal is generated within three count clocks, the changed value cannot be transferred to the register.

<5> When the values of the 8-bit timer counter Hn and the CMP1n register after the change match, an inactive level is output. The 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<6> Clearing the TMHEn bit to 0 during timer Hn operation sets the INTTMHn signal to the default and PWM output to an inactive level.

Remark n = 0, 1

8.4.3 Carrier generator operation (8-bit timer H1 only)

In the carrier generator mode, the 8-bit timer H1 is used to generate the carrier signal of an infrared remote controller, and the 8-bit timer/event counter 51 is used to generate an infrared remote control signal (time count).

The carrier clock generated by the 8-bit timer H1 is output in the cycle set by the 8-bit timer/event counter 51.

In carrier generator mode, the output of the 8-bit timer H1 carrier pulse is controlled by the 8-bit timer/event counter 51, and the carrier pulse is output from the TOH1 output.

(1) Carrier generation

In carrier generator mode, the 8-bit timer H compare register 01 (CMP01) generates a low-level width carrier pulse waveform and the 8-bit timer H compare register 11 (CMP11) generates a high-level width carrier pulse waveform.

Rewriting the CMP11 register during the 8-bit timer H1 operation is possible but rewriting the CMP01 register is prohibited.

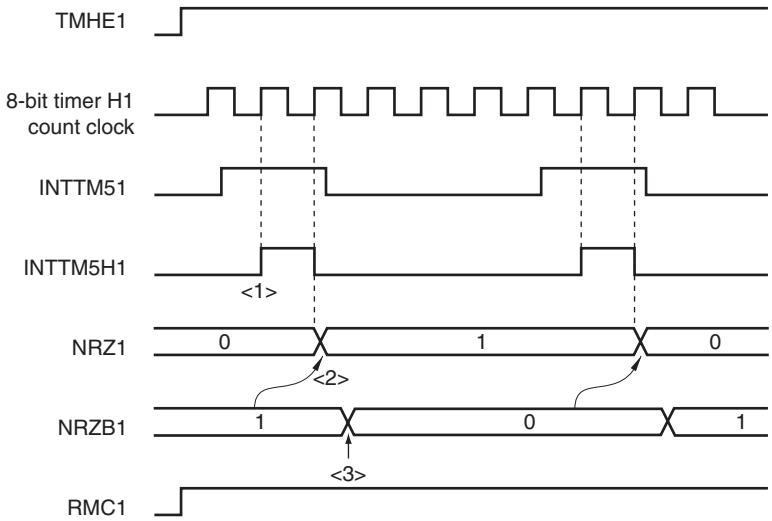
(2) Carrier output control

Carrier output is controlled by the interrupt request signal (INTTM51) of the 8-bit timer/event counter 51 and the NRZB1 and RMC1 bits of the 8-bit timer H carrier control register (TMCYC1). The relationship between the outputs is shown below.

RMC1 Bit	NRZB1 Bit	Output
0	0	Low-level output
0	1	High-level output at rising edge of INTTM51 signal input
1	0	Low-level output
1	1	Carrier pulse output at rising edge of INTTM51 signal input

To control the carrier pulse output during a count operation, the NRZ1 and NRZB1 bits of the TMCYC1 register have a master and slave bit configuration. The NRZ1 bit is read-only but the NRZB1 bit can be read and written. The INTTM51 signal is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal of the NRZ1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit. The timing for transfer from the NRZB1 bit to the NRZ1 bit is as shown below.

Figure 8-13. Transfer Timing



<1> The INTTM51 signal is synchronized with the count clock of the 8-bit timer H1 and is output as the INTTM5H1 signal.

<2> The value of the NRZB1 bit is transferred to the NRZ1 bit at the second clock from the rising edge of the INTTM5H1 signal.

<3> Write the next value to the NRZB1 bit in the interrupt servicing program that has been started by the INTTM5H1 interrupt or after timing has been checked by polling the interrupt request flag. Write data to count the next time to the CR51 register.

Cautions 1. Do not rewrite the NRZB1 bit again until at least the second clock after it has been rewritten, or else the transfer from the NRZB1 bit to the NRZ1 bit is not guaranteed.

2. When the 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated at the timing of <1>. When the 8-bit timer/event counter 51 is used in a mode other than the carrier generator mode, the timing of the interrupt generation differs.

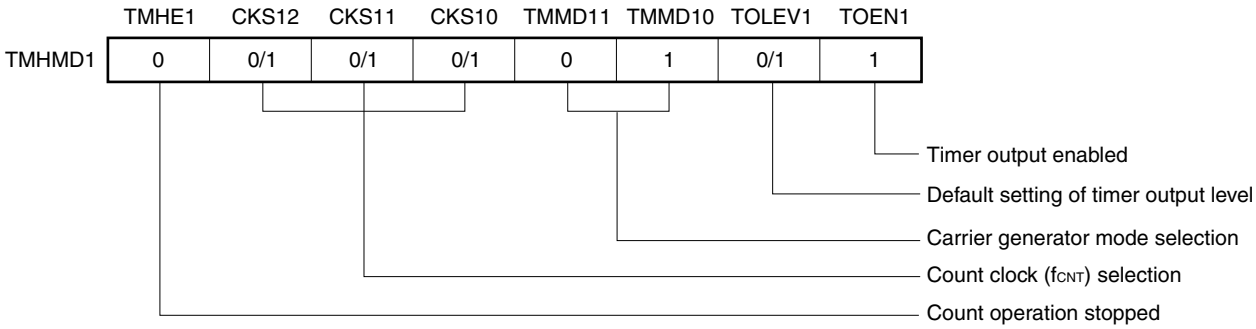
Remark INTTM5H1 is an internal signal and not an interrupt source.

Setting

<1> Set each register.

Figure 8-14. Register Setting in Carrier Generator Mode

(i) Setting 8-bit timer H mode register 1 (TMHMD1)



(ii) CMP01 register setting

- Compare value

(iii) CMP11 register setting

- Compare value

(iv) TMCYC1 register setting

- RMC1 = 1 ... Remote control output enable bit
- NRZB1 = 0/1 ... carrier output enable bit

(v) TCL51 and TMC51 register setting

- Refer to **7.3 Registers Controlling 8-Bit Timer/Event Counter 51**.

<2> When TMHE1 = 1, the 8-bit timer H1 starts counting.

<3> When TCE51 of the 8-bit timer mode control register 51 (TMC51) is set to 1, the 8-bit timer/event counter 51 starts counting.

<4> After the count operation is enabled, the first compare register to be compared is the CMP01 register. When the count value of the 8-bit timer counter H1 and the CMP01 register value match, the INTTMH1 signal is generated, the 8-bit timer counter H1 is cleared. At the same time, the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register.

<5> When the count value of the 8-bit timer counter H1 and the CMP11 register value match, the INTTMH1 signal is generated, the 8-bit timer counter H1 is cleared. At the same time, the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register.

<6> By performing procedures <4> and <5> repeatedly, a carrier clock is generated.

<7> The INTTM51 signal is synchronized with count clock of the 8-bit timer H1 and output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<8> Write the next value to the NRZB1 bit in the interrupt servicing program that has been started by the INTTM5H1 interrupt or after timing has been checked by polling the interrupt request flag. Write data to count the next time to the CR51 register.

<9> When the NRZ1 bit is high level, a carrier clock is output by TOH1 output.

<10> By performing the procedures above, an arbitrary carrier clock is obtained. To stop the count operation, clear TMHE1 to 0.

If the setting value of the CMP01 register is N, the setting value of the CMP11 register is M, and the count clock frequency is f_{CNT}, the carrier clock output cycle and duty are as follows.

- Carrier clock output cycle = (N + M + 2)/f_{CNT}
- Duty = High-level width/carrier clock output width = (M + 1)/(N + M + 2)

Cautions 1. Be sure to set the CMP11 register when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).

2. Set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

3. Set the values of the CMP01 and CMP11 registers in a range of 01H to FFH.

4. The set value of the CMP11 register can be changed while the timer counter is operating. However, it takes the duration of three operating clocks (signal selected by the CKS12 to CKS10 bits of the TMHMD1 register) since the value of the CMP11 register has been changed until the value is transferred to the register.

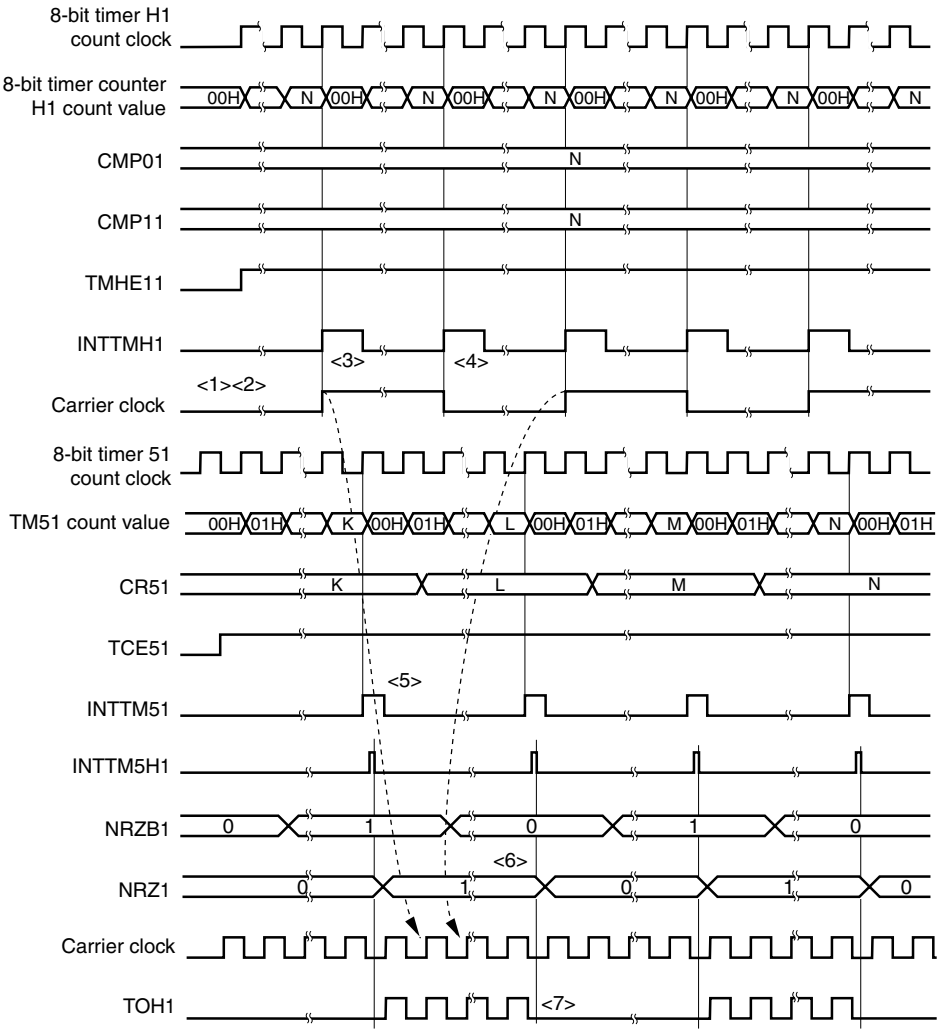
5. Be sure to set the RMC1 bit before the count operation is started.

Remarks 1. For the setting of the output pin, refer to **8.3 (2) Port mode register 3 (PM3)**.

2. For how to enable the INTTMH1 signal interrupt, refer to **CHAPTER 10 INTERRUPT FUNCTIONS**.

Figure 8-15. Carrier Generator Mode Operation Timing (1/3)

(a) Operation when CMP01 = N, CMP11 = N



<1> When TMHE1 = 0 and TCE51 = 0, the 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, the 8-bit timer counter H1 starts a count operation. At that time, the carrier clock remains default.

<3> When the count value of the 8-bit timer counter H1 matches the CMP01 register value, the first INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. The 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of the 8-bit timer counter H1 matches the CMP11 register value, the INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. The 8-bit timer counter H1 is cleared to 00H. By performing procedures <3> and <4> repeatedly, a carrier clock with duty fixed to 50% is generated.

<5> When the INTTM51 signal is generated, it is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal.

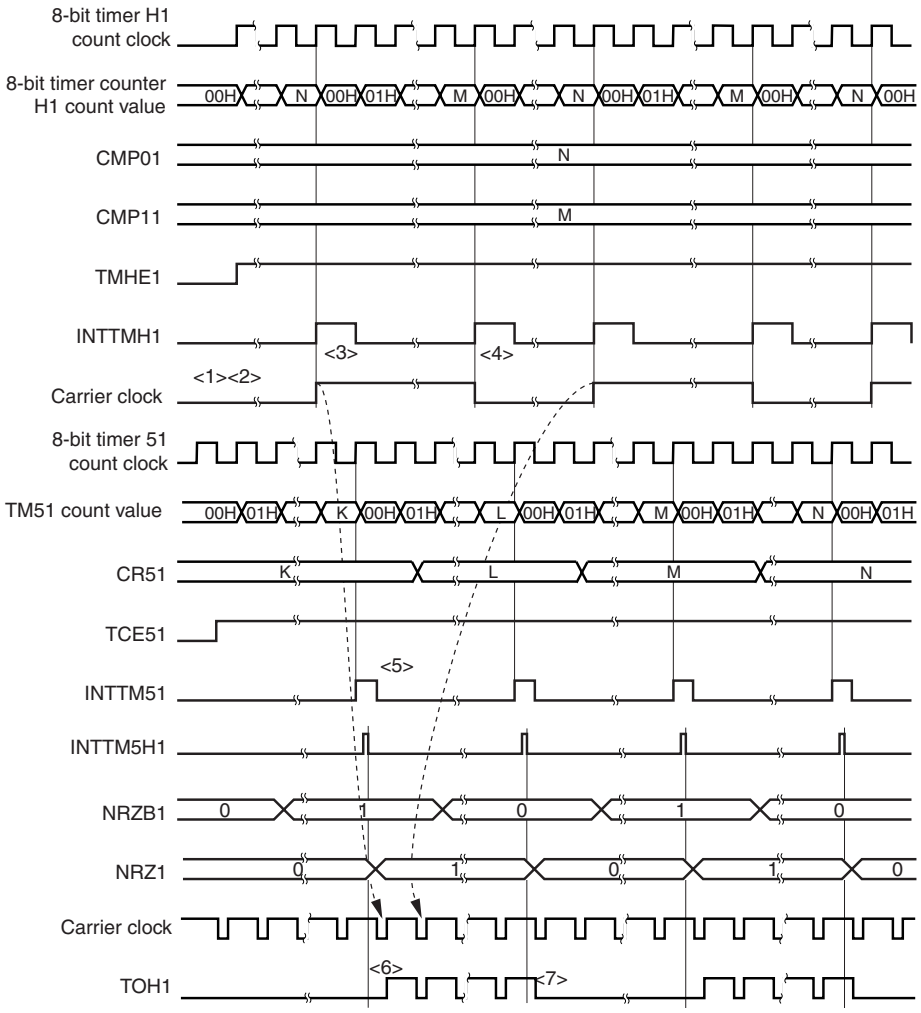
<6> The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<7> When NRZ1 = 0 is set, the TOH1 output becomes low level.

Remark INTTM5H1 is an internal signal and not an interrupt source.

Figure 8-15. Carrier Generator Mode Operation Timing (2/3)

(b) Operation when CMP01 = N, CMP11 = M



<1> When TMHE1 = 0 and TCE51 = 0, the 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, the 8-bit timer counter H1 starts a count operation. At that time, the carrier clock remains default.

<3> When the count value of the 8-bit timer counter H1 matches the CMP01 register value, the first INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. The 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of the 8-bit timer counter H1 matches the CMP11 register value, the INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. The 8-bit timer counter H1 is cleared to 00H. By performing procedures <3> and <4> repeatedly, a carrier clock with duty fixed to other than 50% is generated.

<5> When the INTTM51 signal is generated, it is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal.

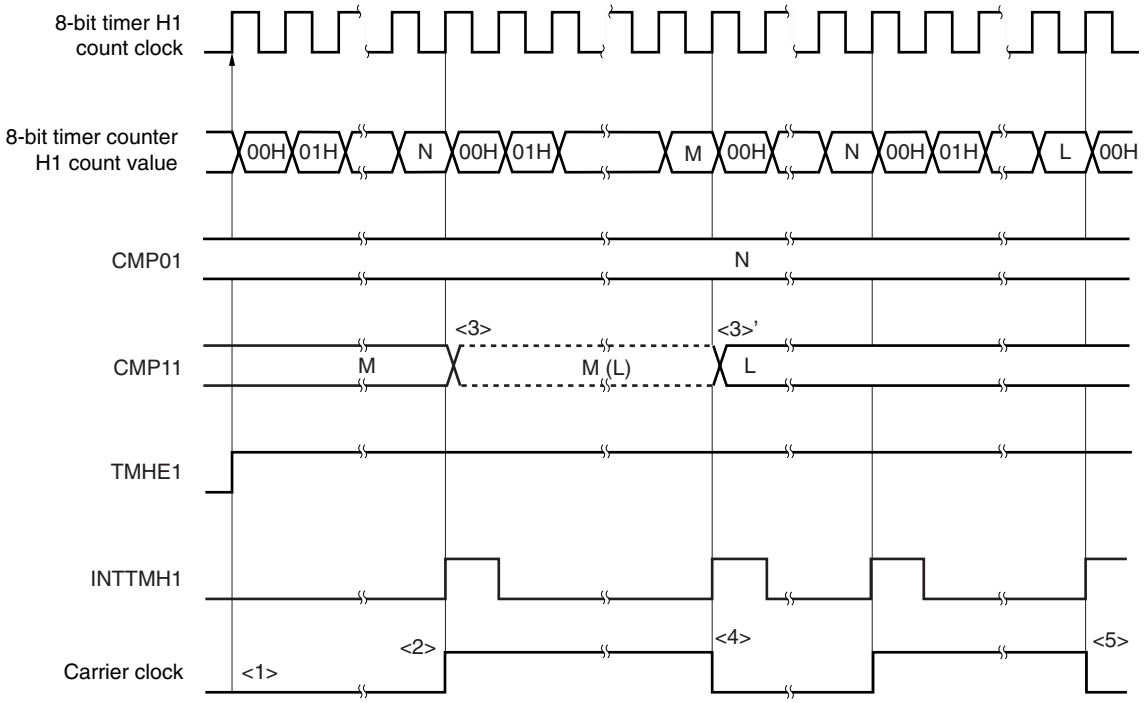
<6> A carrier signal is output at the first rising edge of the carrier clock if NRZ1 is set to 1.

<7> When NRZ1 = 0, the TOH1 output is held at the high level and is not changed to low level while the carrier clock is high level (from <6> and <7>, the high-level width of the carrier clock waveform is guaranteed).

Remark INTTM5H1 is an internal signal and not an interrupt source.

Figure 8-15. Carrier Generator Mode Operation Timing (3/3)

(c) Operation when CMP11 is changed



<1> When TMHE1 = 1 is set, the 8-bit timer H1 starts a count operation. At that time, the carrier clock remains default.

<2> When the count value of the 8-bit timer counter H1 matches the value of the CMP01 register, the INTTMH1 signal is output, the carrier signal is inverted, and the timer counter is cleared to 00H. At the same time, the compare register whose value is to be compared with that of the 8-bit timer counter H1 is changed from the CMP01 register to the CMP11 register.

<3> The CMP11 register is asynchronous to the count clock, and its value can be changed while the 8-bit timer H1 is operating. The new value (L) to which the value of the register is to be changed is latched. When the count value of the 8-bit timer counter H1 matches the value (M) of the CMP11 register before the change, the CMP11 register is changed (<3>').

However, it takes three count clocks or more since the value of the CMP11 register has been changed until the value is transferred to the register. Even if a match signal is generated before the duration of three count clocks elapses, the new value is not transferred to the register.

<4> When the count value of 8-bit timer counter H1 matches the value (M) of the CMP11 register before the change, the INTTMH1 signal is output, the carrier signal is inverted, and the timer counter is cleared to 00H. At the same time, the compare register whose value is to be compared with that of the 8-bit timer counter H1 is changed from the CMP11 register to the CMP01 register.

<5> The timing at which the count value of the 8-bit timer counter H1 and the CMP11 register value match again is indicated by the value after the change (L).

CHAPTER 9 WATCHDOG TIMER

9.1 Functions of Watchdog Timer

The watchdog timer operates on the internal low-speed oscillation clock.

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to WDTE
- If data is written to WDTE during a window close period
- If the instruction is fetched from an area not set by the IMS register (detection of an invalid check while the CPU hangs up)
- If the CPU accesses an area that is not set by the IMS register (excluding FB00H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of RESF, refer to **CHAPTER 12 RESET FUNCTION**.

9.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

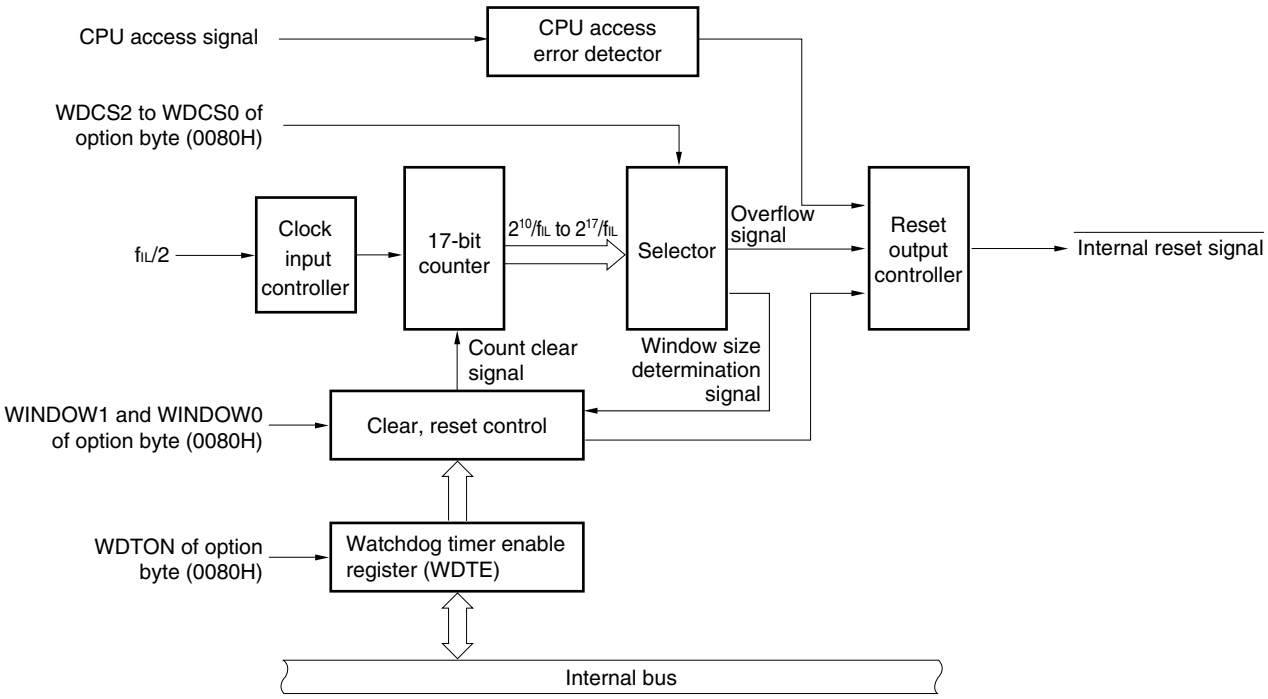
How the counter operation is controlled, overflow time, and window open period are set by the option byte.

Table 9-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (0080H)
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)

Remark For the option byte, refer to **CHAPTER 16 OPTION BYTE**.

Figure 9-1. Block Diagram of Watchdog Timer



9.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

(1) Watchdog timer enable register (WDTE)

Writing ACH to WDTE clears the watchdog timer counter and starts counting again.

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

Reset signal generation sets this register to 9AH or 1AH[Note].

Figure 9-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FF99H 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 (0080H). 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. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

3. The value read from WDTE is 9AH/1AH (this differs from the written value (ACH)).

9.4 Operation of Watchdog Timer

9.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (0080H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (0080H) to 1 (the counter starts operating after a reset release) (for details, refer to **CHAPTER 16**).

WDTON	Operation Control of Watchdog Timer Counter/Illegal Access Detection
0	Counter operation disabled (counting stopped after reset), illegal access detection operation disabled
1	Counter operation enabled (counting started after reset), illegal access detection operation enabled

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (0080H) (for details, refer to **9.4.2** and **CHAPTER 16**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (0080H) (for details, refer to **9.4.3** and **CHAPTER 16**).

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
   - If the instruction is fetched from an area not set by the IMS register (detection of an invalid check during a CPU program loop)
   - If the CPU accesses an area not set by the IMS register (excluding FB00H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

Cautions 1. The first writing to WDTE after a reset release clears the watchdog timer, if it is made before the overflow time regardless of the timing of the writing, 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 (FFFFH).

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 (LSROSC) of the option byte.

	LSROSC = 0 (Internal Low-Speed Oscillator Can Be Stopped by Software)	LSROSC = 1 (Internal Low-Speed Oscillator Cannot Be Stopped)
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		

If LSROSC = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is not cleared to 0 but starts counting from the value at which it was stopped.

If oscillation of the internal low-speed oscillator is stopped by setting LSRSTOP (bit 1 of the internal oscillation mode register (RCM) = 1) when LSROSC = 0, the watchdog timer stops operating. At this time, the counter is not cleared to 0.

9.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (0080H).

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 9-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer
0	0	0	$2^{10}/f_{IL}$ (3.88 ms)
0	0	1	$2^{11}/f_{IL}$ (7.76 ms)
0	1	0	$2^{12}/f_{IL}$ (15.52 ms)
0	1	1	$2^{13}/f_{IL}$ (31.03 ms)
1	0	0	$2^{14}/f_{IL}$ (62.06 ms)
1	0	1	$2^{15}/f_{IL}$ (124.12 ms)
1	1	0	$2^{16}/f_{IL}$ (248.24 ms)
1	1	1	$2^{17}/f_{IL}$ (496.48 ms)

Caution The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

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

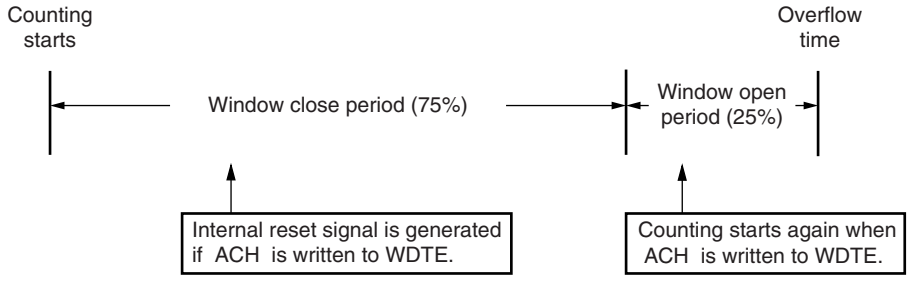
2. ( ): f_{IL} = 264 kHz (MAX.)

9.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (0080H). 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%



Caution The first writing to WDTE after a reset release clears the watchdog timer, if it is made before the overflow time regardless of the timing of the writing, and the watchdog timer starts counting again.

The window open period to be set is as follows.

Table 9-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	25%
0	1	50%
1	0	75%
1	1	100%

Caution The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

Remark If the overflow time is set to $2^{11}/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period			
	25%	50%	75%	100%
Window close time	0 to 7.11 ms	0 to 4.74 ms	0 to 2.37 ms	None
Window open time	7.11 to 7.76 ms	4.74 to 7.76 ms	2.37 to 7.76 ms	0 to 7.76 ms

<When window open period is 25%>

- Overflow time:
  $2^{11}/f_{IL}$ (MAX.) = 2^{11}/264 kHz (MAX.) = 7.76 ms
- Window close time:
  0 to $2^{11}/f_{IL}$ (MIN.) × (1 − 0.25) = 0 to 2^{11}/216 kHz (MIN.) × 0.75 = 0 to 7.11 ms
- Window open time:
  $2^{11}/f_{IL}$ (MIN.) × (1 − 0.25) to $2^{11}/f_{IL}$ (MAX.) = 2^{11}/216 kHz (MIN.) × 0.75 to 2^{11}/264 kHz (MAX.) = 7.11 to 7.76 ms

CHAPTER 10 INTERRUPT FUNCTIONS

10.1 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 a high interrupt priority group and a low interrupt priority group by setting the priority specification flag registers (PR0L, PR0H, PR1L).

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, refer to **Table 10-1**.

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.

(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.

10.2 Interrupt Sources and Configuration

The interrupt sources consist of maskable interrupts and software interrupts. In addition, they also have up to four reset sources (refer to **Table 10-1**).

Table 10-1. Interrupt Source List (1/2)

Interrupt Type	Internal/ External	Basic Configuration Type Note 1	Default Priority Note 2	Interrupt Source		Vector Table Address
				Name	Trigger	
Maskable	Internal	(A)	0	INTLM	Low-voltage detection Note 3	0004H
	External	(B)	1	INTP0	Pin input edge detection	0006H
			2	INTP1		0008H
			3	INTP2		000AH
			4	INTP3		000CH
	-	-	5	-	-	000EH
			6	-		0010H
			7	-		0012H
			8	-		0014H
			9	-		0016H
			10	-		0018H
	Internal	(A)	11	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)	001AH
			12	INTTMH0	Match between TMH0 and CMP00 (when compare register is specified)	001CH
	-	-	13	-	-	001EH
	Internal	(A)	14	INTTM000	Match between TM00 and CR000 (when compare register is specified), TI010 pin valid edge detection (when capture register is specified)	0020H
			15	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)	0022H
	-	-	16	-	-	0024H
			17			0026H
			18			0028H
	Internal	(A)	19	INTTM51 Note 4	Match between TM51 and CR51 (when compare register is specified)	002AH
	-	-	20	-	-	002CH

Notes 1. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 10-1**.

2. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 28 indicates the lowest priority.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

4. When 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (refer to **Figure 8-13 Transfer Timing**).

Table 10-1. Interrupt Source List (2/2)

Interrupt Type	Internal/ External	Basic Configuration Type[Note 1]	Default Priority[Note 2]	Interrupt Source		Vector Table Address
				Name	Trigger	
Maskable	-	-	21	-	-	002EH
			22			0030H
			23			0032H
			24			0034H
			25			0036H
			26			0038H
			27			003AH
			28			003CH
Software	–	(C)	–	BRK	BRK instruction execution	003EH
Reset	–	–	–	RESET	Reset input	0000H
				POC	Power-on clear	
				LVI	Low-voltage detection[Note 3]	
				WDT	WDT overflow	

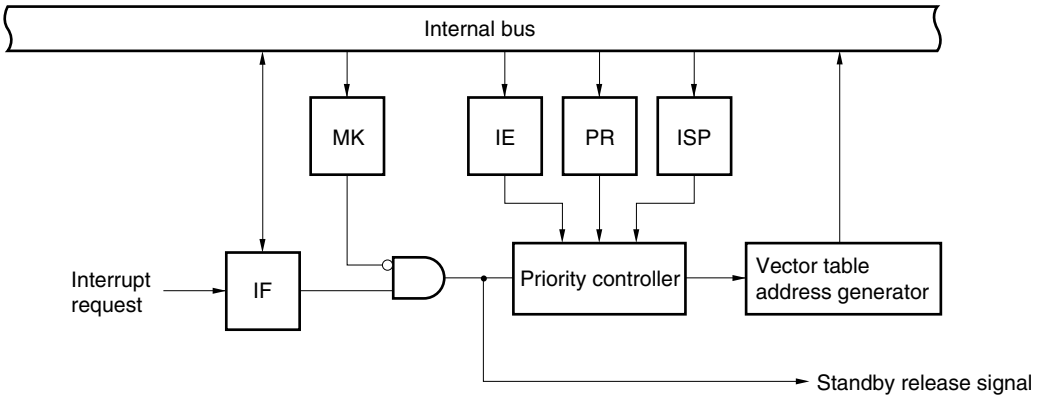
Notes 1. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 10-1**.

2. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 28 indicates the lowest priority.

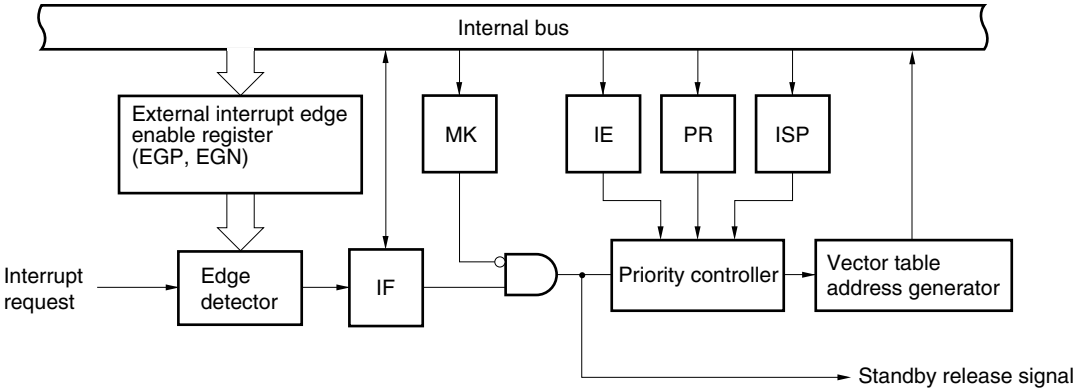
3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

Figure 10-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

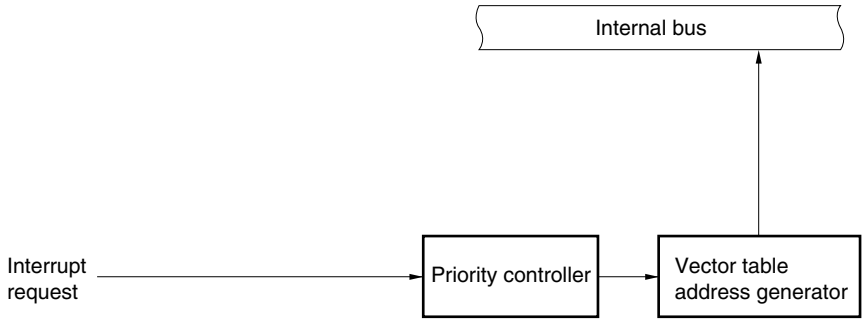


Remark n = 0 to 3

IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

Figure 10-1. Basic Configuration of Interrupt Function (2/2)

(C) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

10.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L)
- Priority specification flag registers (PR0L, PR0H, PR1L)
- External interrupt rising edge enable registers (EGP)
- External interrupt falling edge enable registers (EGN)
- Program status word (PSW)

Table 10-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 10-2. Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTLVI	LVIIF	IF0L	LVIMK	MK0L	LVIPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTTMH1	TMIFH1	IF0H	TMMKH1	MK0H	TMPRH1	PR0H
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTTM51[Note]	TMIF51	IF1L	TMMK51	MK1L	TMPR51	PR1L

Note When 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (refer to **Figure 8-13 Transfer Timing**).

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

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, and IF1L are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they are set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Cautions 1. When operating a timer after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of 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.

Figure 10-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L)

Address: FFE0H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF0L	0	0	0	PIF3	PIF2	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	1	0
IF0H	TMIF010	TMIF000	0	TMIFH0	TMIFH1	0	0	0

Address: FFE2H After reset: 00H R/W

Symbol	7	6	5	4	<3>	2	1	0
IF1L	0	0	0	0	TMIF51	0	0	0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution Be sure to clear bits 5 and 7 of IF0L, bits 0 to 2, and 5 of IF0H, and bits 0 to 2, 4 to 7 of IF1L to 0.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, and MK1L are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H are combined to form 16-bit register MK0, they are set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 10-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L)

Address: FFE4H After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK0L	1	1	1	PMK3	PMK2	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	2	1	0
MK0H	TMMK010	TMMK000	1	TMMKH0	TMMKH1	1	1	1

Address: FFE6H After reset: FFH R/W

Symbol	7	6	5	4	<3>	2	1	0
MK1L	1	1	1	1	TMMK51	1	1	1

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bits 5 to 7 of MK0L, bits 0 to 2, and 5 of MK0H, and bits 0 to 2, 4 to 7 of MK1L to 1.

(3) Priority specification flag registers (PR0L, PR0H, PR1L)

The priority specification flag registers are used to set the corresponding maskable interrupt priority order.

PR0L, PR0H, and PR1L are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are combined to form 16-bit register PR0, they are set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 10-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L)

Address: FFE8H After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR0L	1	1	1	PPR3	PPR2	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	5	<4>	<3>	2	1	0
PR0H	TMPR010	TMPR000	1	TMPRH0	TMPRH1	1	1	1

Address: FFEAH After reset: FFH R/W

Symbol	7	6	5	4	<3>	2	1	0
PR1L	1	1	1	1	TMPR51	1	1	1

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Caution Be sure to set bits 5 to 7 of PR0L, bits 0 to 2, and 5 of PR0H, and bits 0 to 2, 4 to 7 of PR1L to 1.

(4) External interrupt rising edge enable registers (EGP), external interrupt falling edge enable registers (EGN)

These registers specify the valid edge for INTPn.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 10-5. Format of External Interrupt Rising Edge Enable Registers (EGP) and External Interrupt Falling Edge Enable Registers (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	0	0	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	0	0	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Caution Be sure to clear bits 3 to 7 of EGP and EGN to 0.

Remark n = 0 to 3

Table 10-3 shows the ports corresponding to EGPn and EGNn.

Table 10-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P30	INTP0
EGP1	EGN1	P125	INTP1
EGP2	EGN2	P31	INTP2
EGP3	EGN3	P32	INTP3

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 3

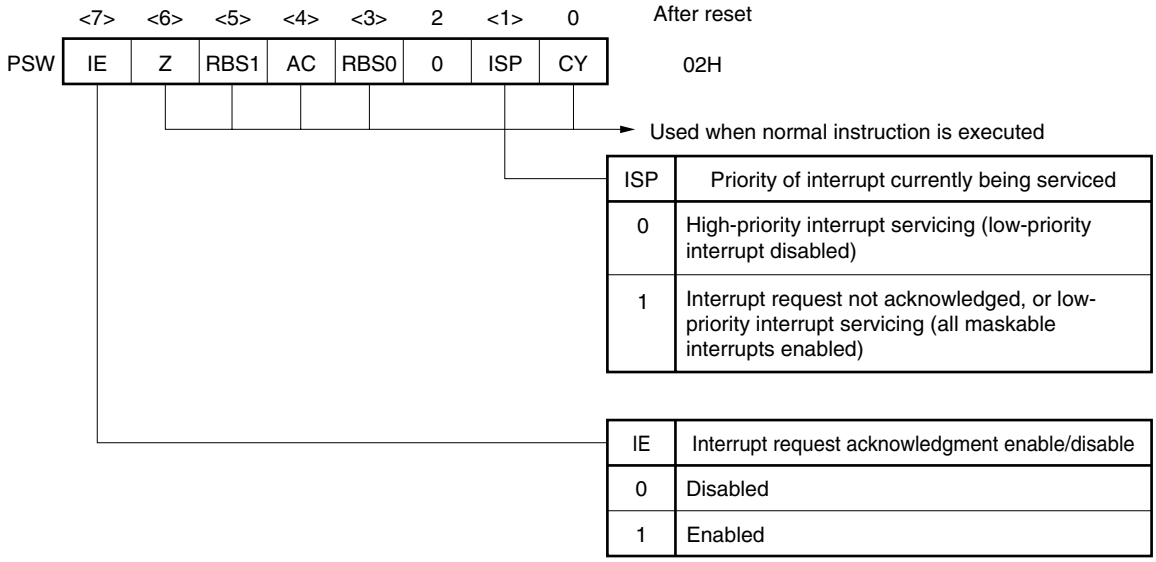
(5) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP flag 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 ISP flag. 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 02H.

Figure 10-6. Format of Program Status Word



ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupt disabled)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (all maskable interrupts enabled)

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

10.4 Interrupt Servicing Operations

10.4.1 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 (when the ISP flag is reset to 0).

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 10-4 below.

For the interrupt request acknowledgment timing, refer to **Figures 10-8** and **10-9**.

Table 10-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: $1/f_{CPU}$ (f_{CPU}: 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.

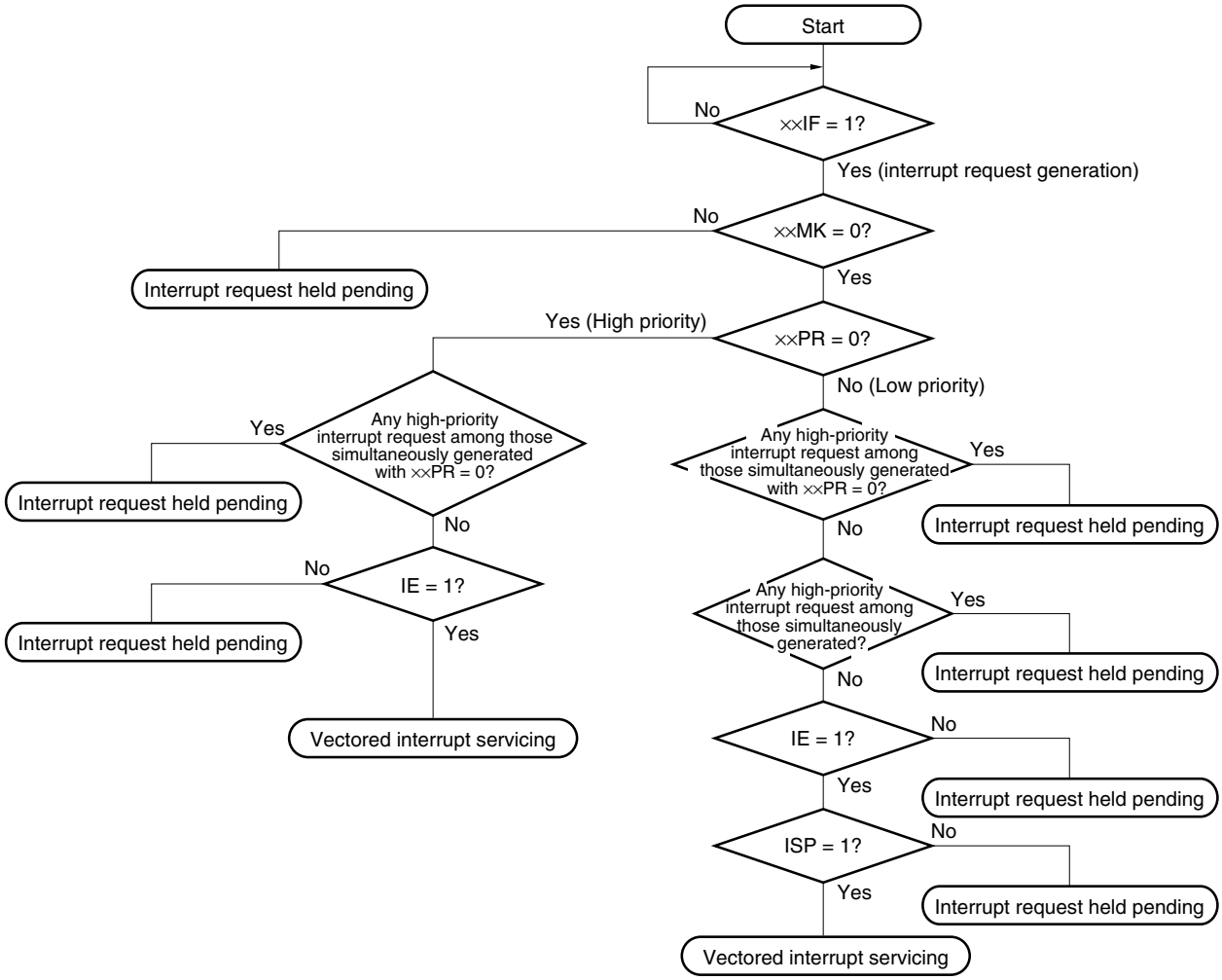
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 10-7 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 ISP flag. The vector table data determined for each interrupt request is the loaded into the PC and branched.

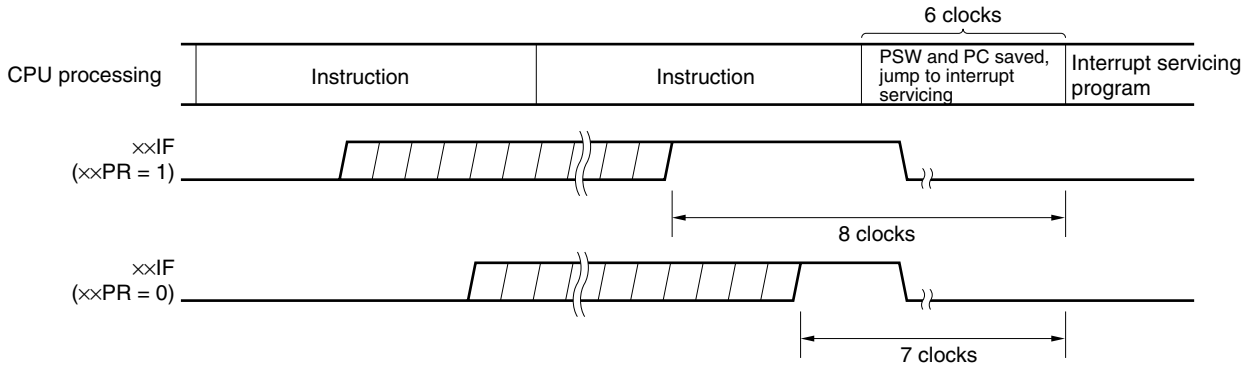
Restoring from an interrupt is possible by using the RETI instruction.

Figure 10-7. Interrupt Request Acknowledgment Processing Algorithm



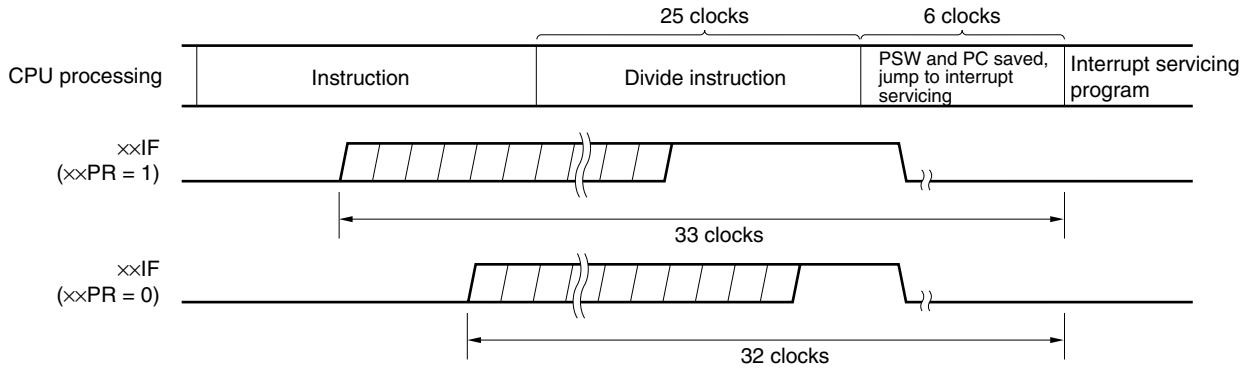
××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR: Priority specification flag
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 10-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 10-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

10.4.2 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 (003EH, 003FH) 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.

10.4.3 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 10-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 10-10 shows multiple interrupt servicing examples.

Table 10-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request				Software Interrupt Request
		PR = 0		PR = 1		
		IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP = 0	○	×	×	×	○
	ISP = 1	○	×	○	×	○
Software interrupt		○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP and IE are flags contained in the PSW.

ISP = 0: An interrupt with higher priority is being serviced.

ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

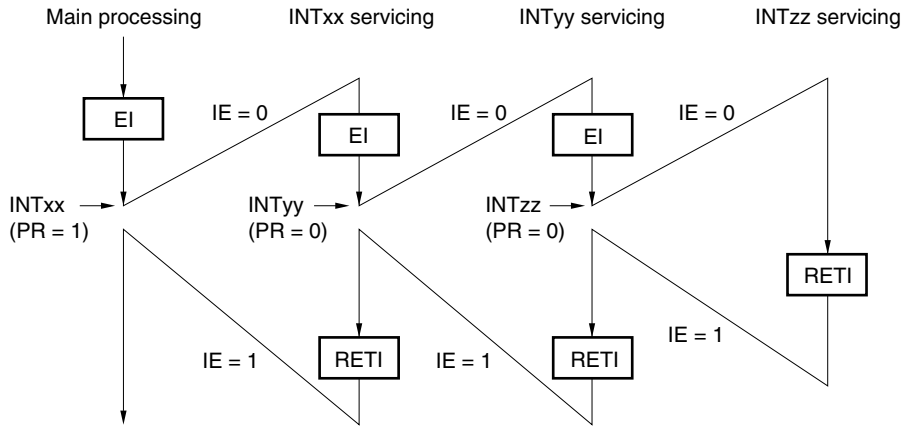
4. PR is a flag contained in PR0L, PR0H, and PR1L.

PR = 0: Higher priority level

PR = 1: Lower priority level

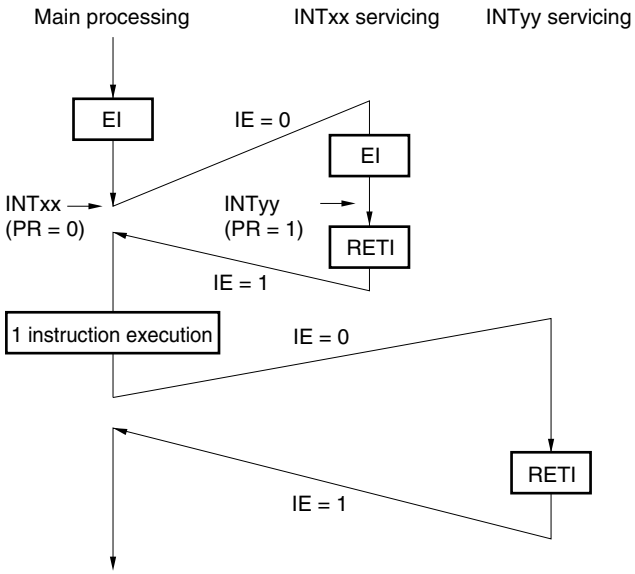
Figure 10-10. Examples of Multiple Interrupt Servicing (1/2)

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

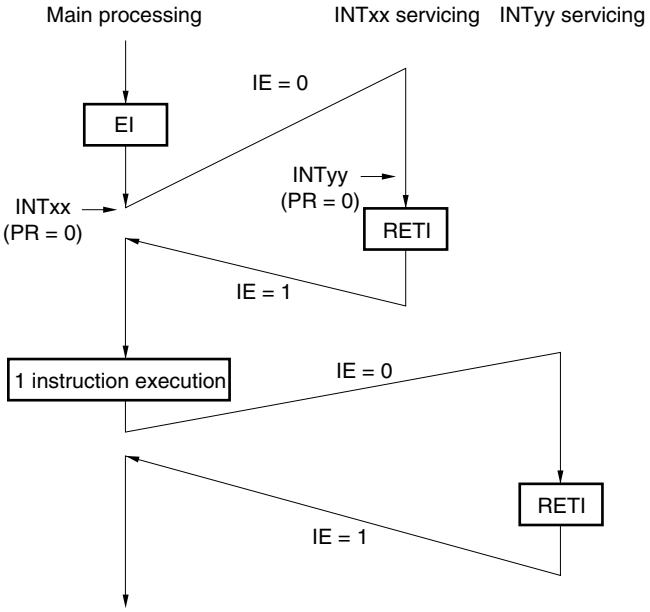


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 10-10. 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 = 0: Higher priority level
IE = 0: Interrupt request acknowledgment disabled

10.4.4 Interrupt request hold

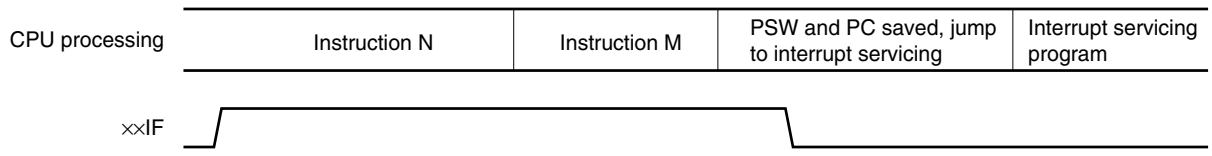
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, $addr16
- BF PSW. bit, $addr16
- BTCLR PSW. bit, $addr16
- EI
- DI
- Manipulation instructions for the IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, and PR1L 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 10-11 shows the timing at which interrupt requests are held pending.

Figure 10-11. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 11 STANDBY FUNCTION

11.1 Standby Function and Configuration

11.1.1 Standby function

The standby function is designed to reduce the operating current of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, internal high-speed oscillator, or internal low-speed oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and internal high-speed oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Caution When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.

11.1.2 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, refer to **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. When X1 clock oscillation starts with the internal high-speed oscillation clock used as the CPU clock, the X1 clock oscillation stabilization time can be checked.

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, and WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1, clear OSTC to 00H.

Figure 11-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status	
						fx = 10 MHz
1	0	0	0	0	2^{11}/fx min.	204.8 µs min.
1	1	0	0	0	2^{13}/fx min.	819.2 µs min.
1	1	1	0	0	2^{14}/fx min.	1.64 ms min.
1	1	1	1	0	2^{15}/fx min.	3.27 ms min.
1	1	1	1	1	2^{16}/fx min.	6.55 ms min.

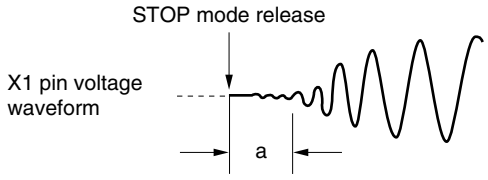
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

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 fx: 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 OSTS to 05H.

Figure 11-2. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection	
				fx = 10 MHz
0	0	1	2^{11}/fx	204.8 µs
0	1	0	2^{13}/fx	819.2 µs
0	1	1	2^{14}/fx	1.64 ms
1	0	0	2^{15}/fx	3.27 ms
1	0	1	2^{16}/fx	6.55 ms
Other than above			Setting prohibited	

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

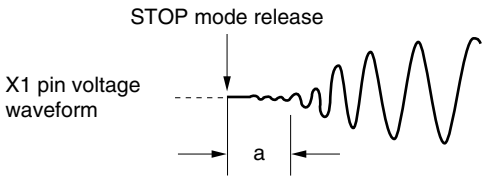
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

3. 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.

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



Remark fx: X1 clock oscillation frequency

11.2 Standby Function Operation

11.2.1 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, or internal high-speed oscillation clock.

The operating statuses in the HALT mode are shown below.

Table 11-1. Operating Statuses in HALT Mode

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{IH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	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)	Status before HALT mode was set is retained
		f_{EXCLK}	Operates or stops by external clock input		Operation continues (cannot be stopped)
	f_{IL}		Status before HALT mode was set is retained		
CPU			Operation stopped		
Flash memory					
RAM			Status before HALT mode was set is retained		
Port (latch)					
16-bit timer/event counter 00			Operable		
8-bit timer/event counter51					
8-bit timers		H0			
		H1			
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.		
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Remark f_{IH}: Internal high-speed oscillation clock, f_X: X1 clock
f_{EXCLK}: External main system clock, f_{IL}: Internal low-speed oscillation clock

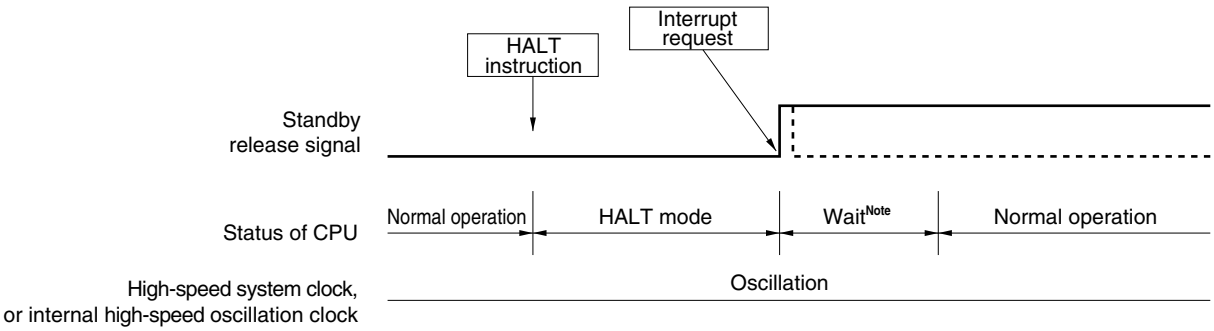
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 11-3. HALT Mode Release by Interrupt Request Generation



Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 11 or 12 clocks
- When vectored interrupt servicing is not carried out: 4 or 5 clocks

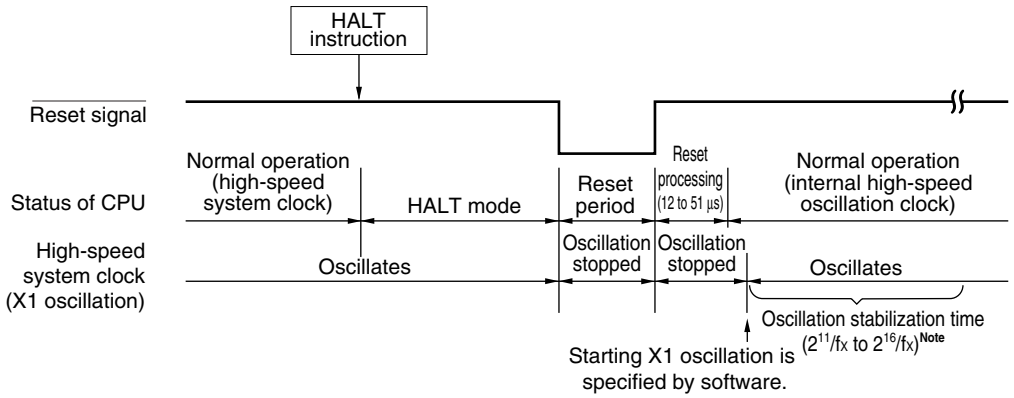
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

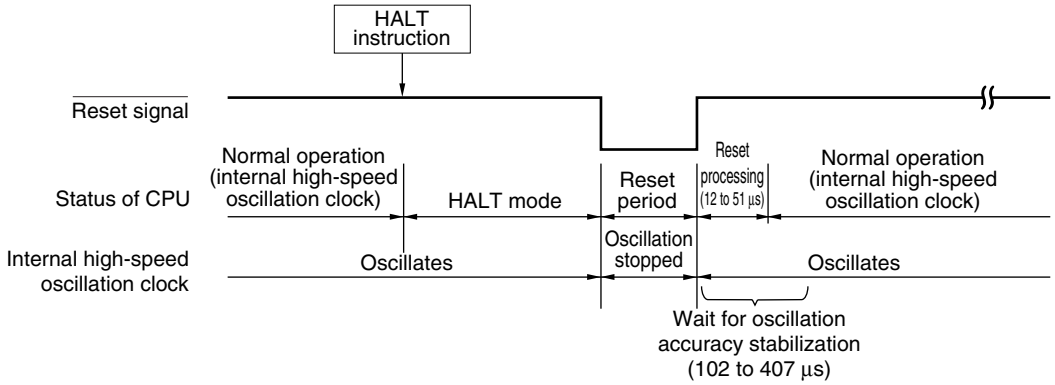
Figure 11-4. HALT Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



Note Oscillation stabilization time is not required when using the external main system clock (f_{EXCLK}) as the high-speed system clock.

(2) When internal high-speed oscillation clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

Table 11-2. Operation in Response to Interrupt Request in HALT Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	HALT mode held
Reset	–	–	×	×	Reset processing

×: don't care

11.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the main system clock.

Caution 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.

The operating statuses in the STOP mode are shown below.

Table 11-3. 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_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Stopped		
		f_X			
		f_{EXCLK}	Input invalid		
	f_{IL}		Status before STOP mode was set is retained		
CPU			Operation stopped		
Flash memory					
RAM			Status before STOP mode was set is retained		
Port (latch)					
16-bit timer/event counter 00			Operation stopped		
8-bit timer/event counter51			Operable only when TI51 is selected as the count clock		
8-bit timer		H0	Operation stopped		
		H1	Operable only when f_{IL}, $f_{IL}/2^7$, $f_{IL}/2^9$ is selected as the count clock		
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.		
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Remark f_{IH}: Internal high-speed oscillation clock, f_X: X1 clock
f_{EXCLK}: External main system clock, f_{IL}: Internal low-speed oscillation clock

Cautions 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

2. Even if "internal low-speed oscillator can be stopped by software" is selected by the option byte, the internal low-speed oscillation clock continues in the STOP mode in the status before the STOP mode is set. To stop the internal low-speed oscillator's oscillation in the STOP mode, stop it by software 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), switch the CPU clock to the internal high-speed oscillation clock before the execution of the STOP instruction using the following procedure.

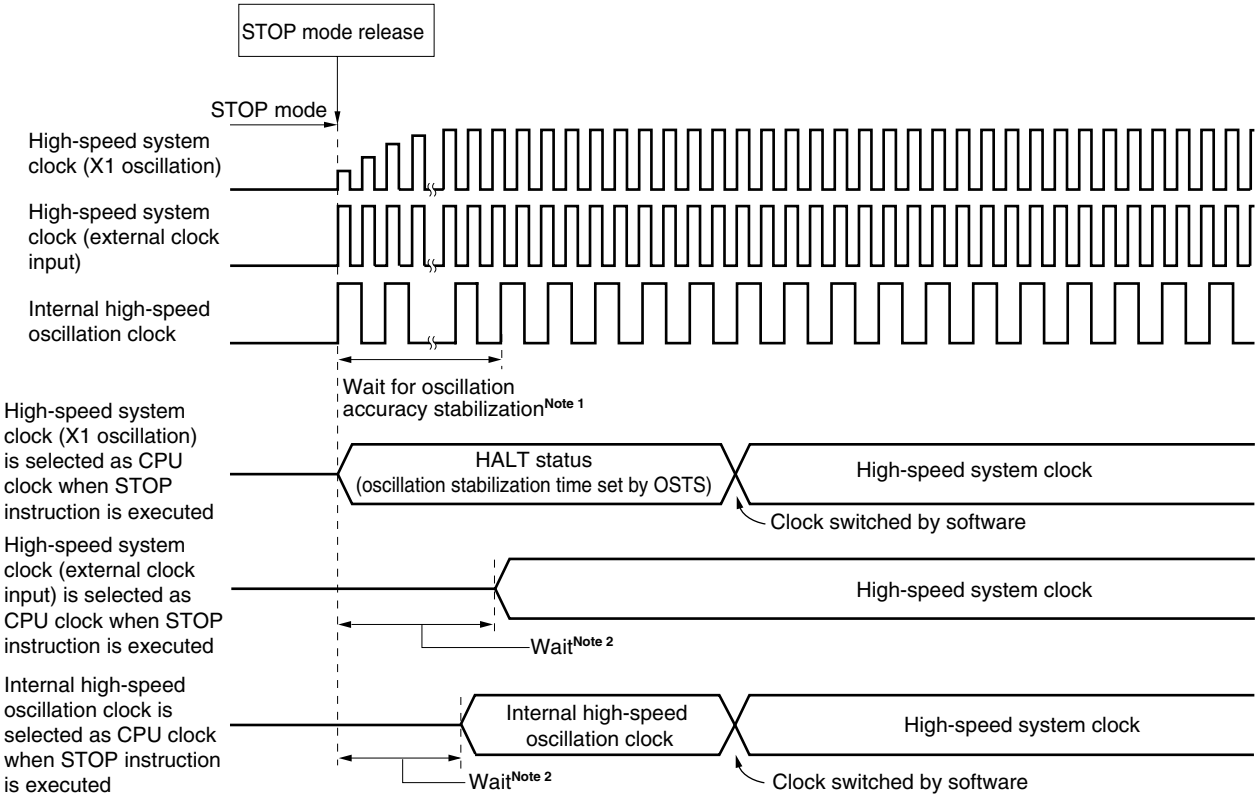
<1> Set RSTOP to 0 (starting oscillation of the internal high-speed oscillator) → <2> Set MCM0 to 0 (switching the CPU from X1 oscillation to internal high-speed oscillation) → <3> Check that MCS is 0 (checking the CPU clock) → <4> Check that RSTS is 1 (checking internal high-speed oscillation operation) → <5> Execute the STOP instruction

Before changing the CPU clock from the internal high-speed oscillation clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

4. Execute the STOP instruction after having confirmed that the internal high-speed oscillator is operating stably (RSTS = 1).

(2) STOP mode release

Figure 11-5. Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)



Notes 1. The wait time for oscillation accuracy stabilization is 102 to 407 μs.

2. The wait time is as follows:

- When vectored interrupt servicing is carried out: 17 or 18 clocks
- When vectored interrupt servicing is not carried out: 11 or 12 clocks

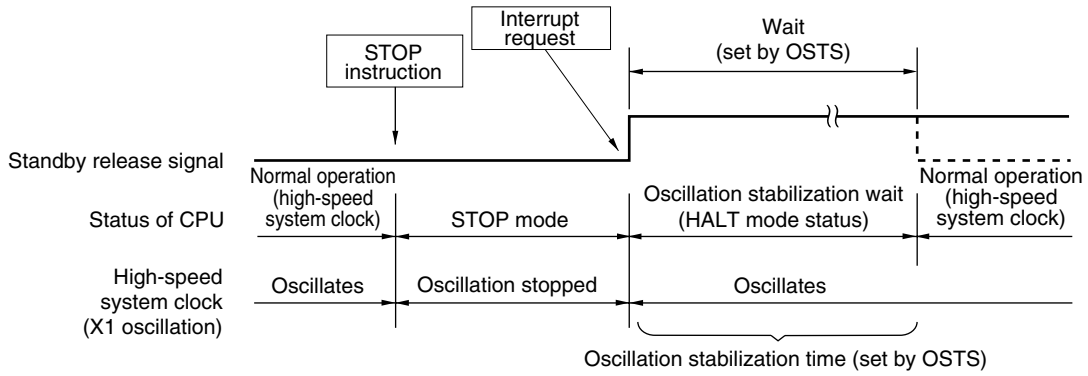
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

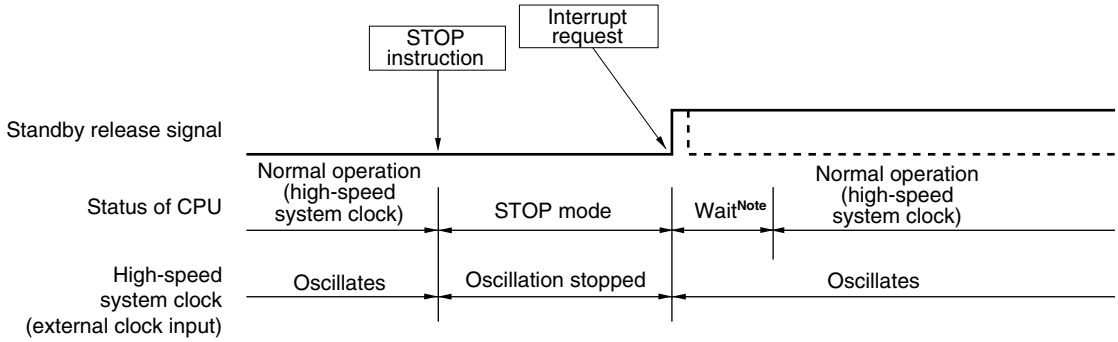
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 11-6. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock



(2) When high-speed system clock (external clock input) is used as CPU clock



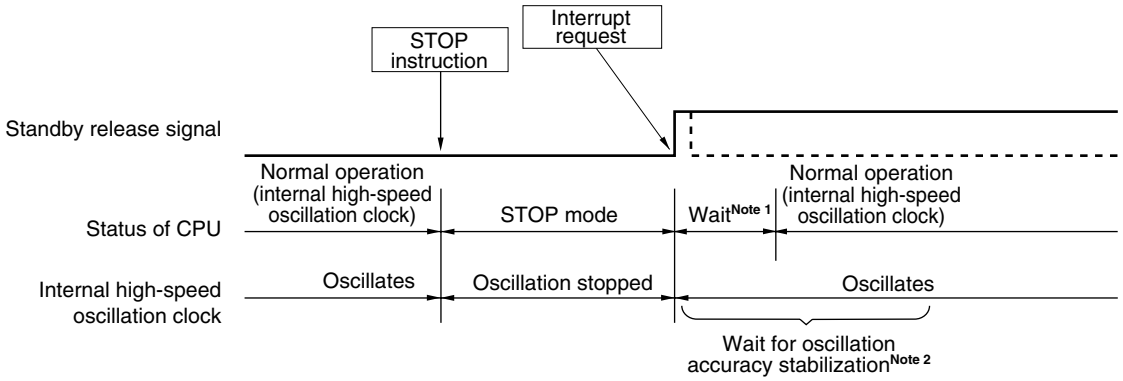
Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 17 or 18 clocks
- When vectored interrupt servicing is not carried out: 11 or 12 clocks

Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 11-6. STOP Mode Release by Interrupt Request Generation (2/2)

(3) When internal high-speed oscillation clock is used as CPU clock



Notes 1. The wait time is as follows:

- When vectored interrupt servicing is carried out: 17 or 18 clocks
- When vectored interrupt servicing is not carried out: 11 or 12 clocks

2. The wait time for oscillation accuracy stabilization is 102 to 407 μs.

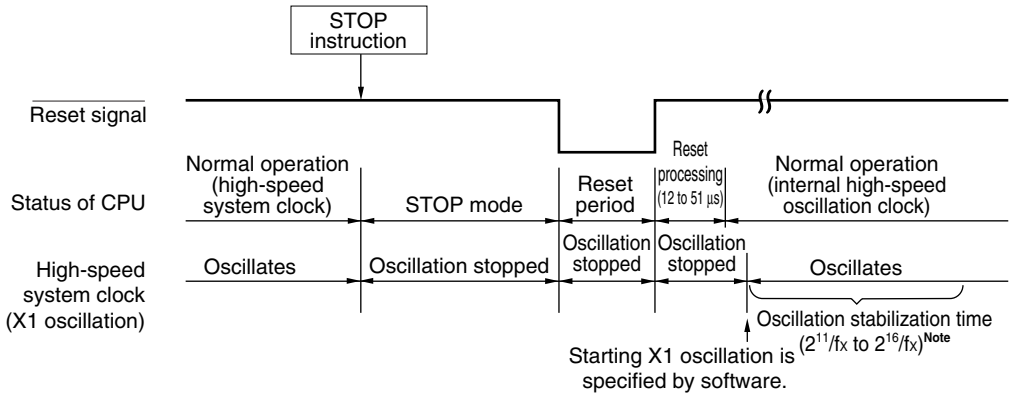
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(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 11-7. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



Note Oscillation stabilization time is not required when using the external main system clock (f_{EXCLK}) as the high-speed system clock.

Remark f_X: X1 clock oscillation frequency

(2) When internal high-speed oscillation clock is used as CPU clock

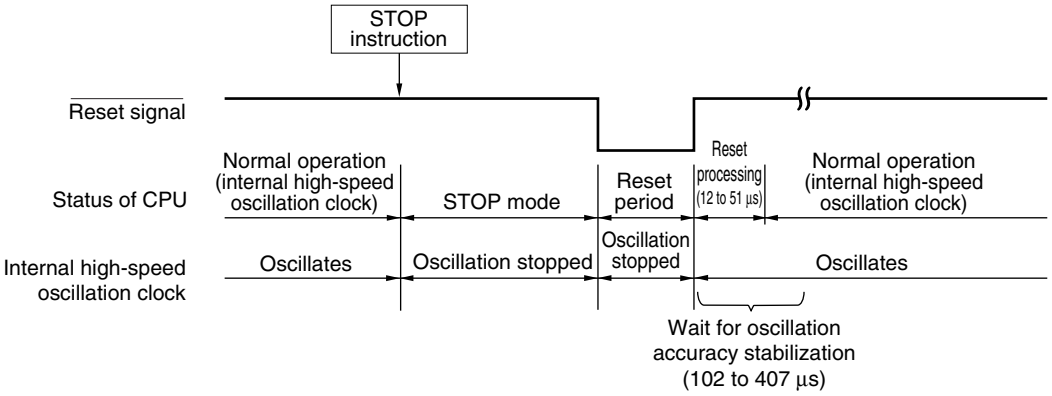


Table 11-4. Operation in Response to Interrupt Request in STOP Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	STOP mode held
Reset	–	–	×	×	Reset processing

×: don't care

CHAPTER 12 RESET FUNCTION

The following four 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) and detection voltage

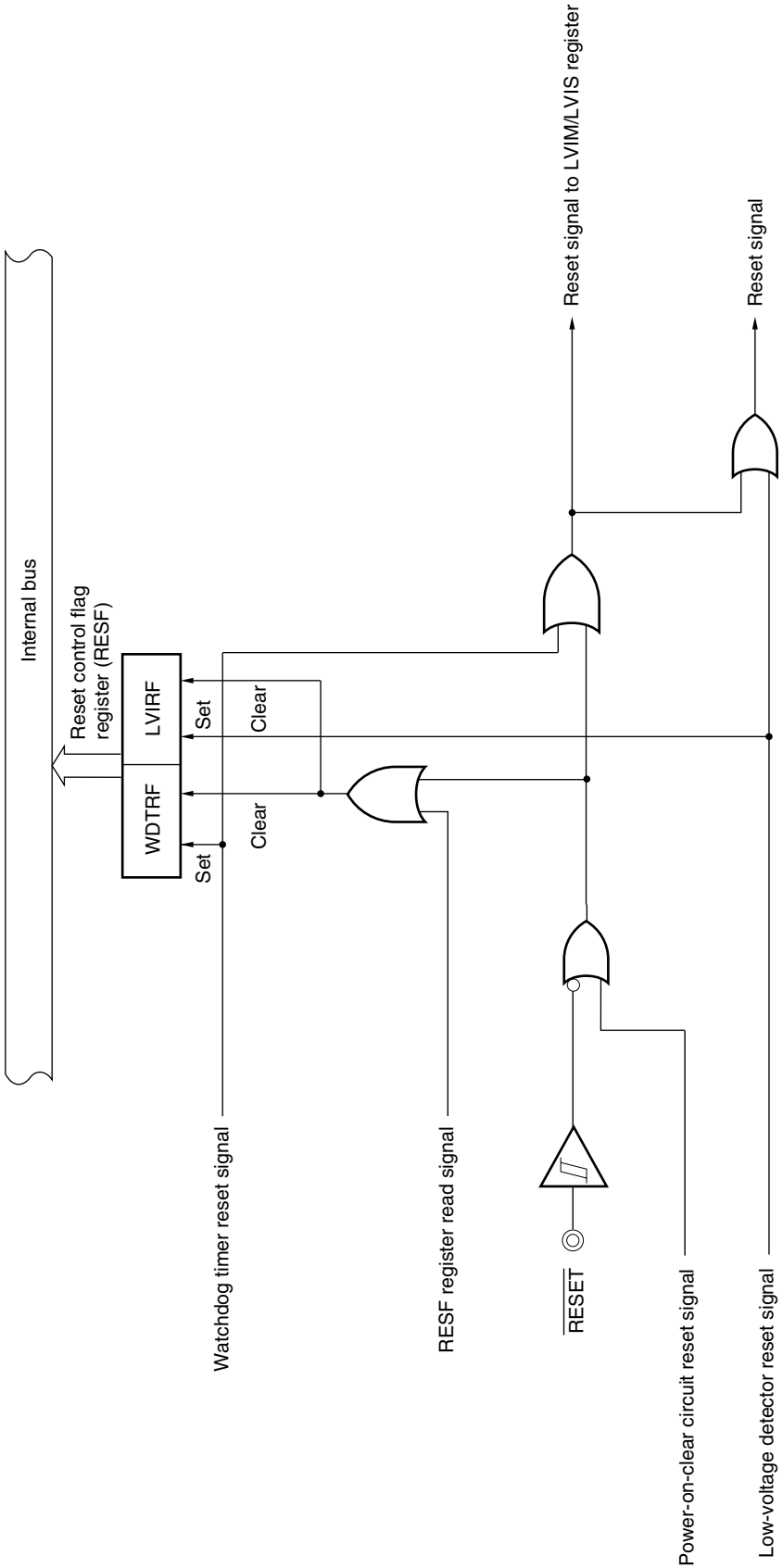
External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Tables 12-1 and 12-2. 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 (refer to **Figures 12-2** to **12-4**) after reset processing. Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (refer to **CHAPTER 13 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 14 LOW-VOLTAGE DETECTOR**) after reset processing.

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} < 4.5 V) is not counted in the 10 µs. However, the low-level input may be continued before POC is released.)
2. During reset signal generation, the X1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.
3. When the STOP mode is released by a reset, the RAM contents in the STOP mode are held during reset input. However, because SFR is initialized, the port pins become high-impedance.

Figure 12-1. Block Diagram of Reset Function



Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level selection register

Figure 12-2. Timing of Reset by $\overline{\text{RESET}}$ Input

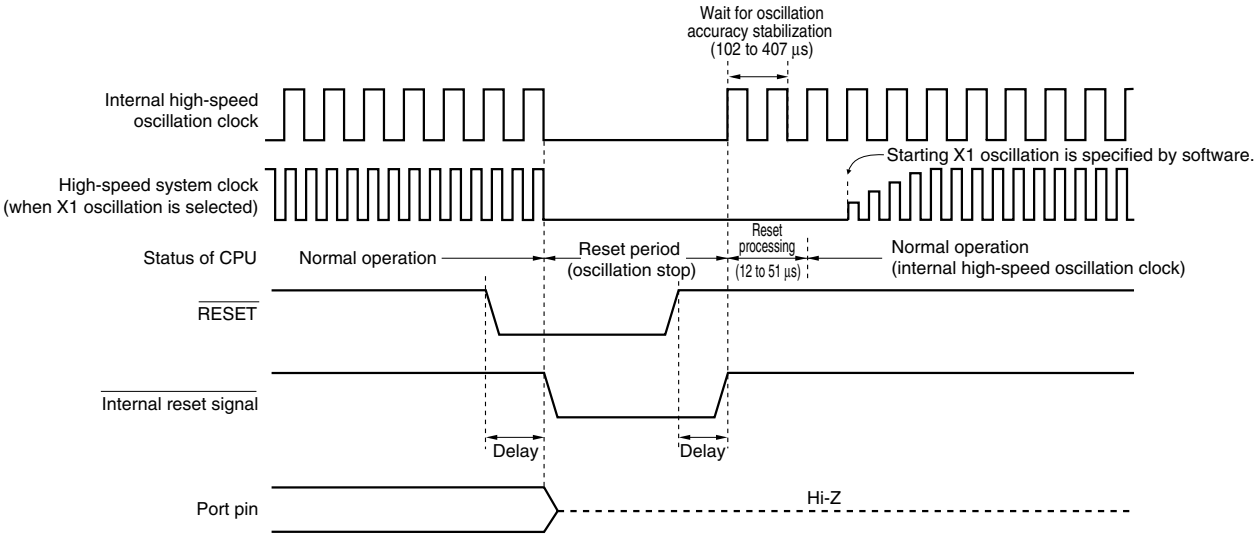
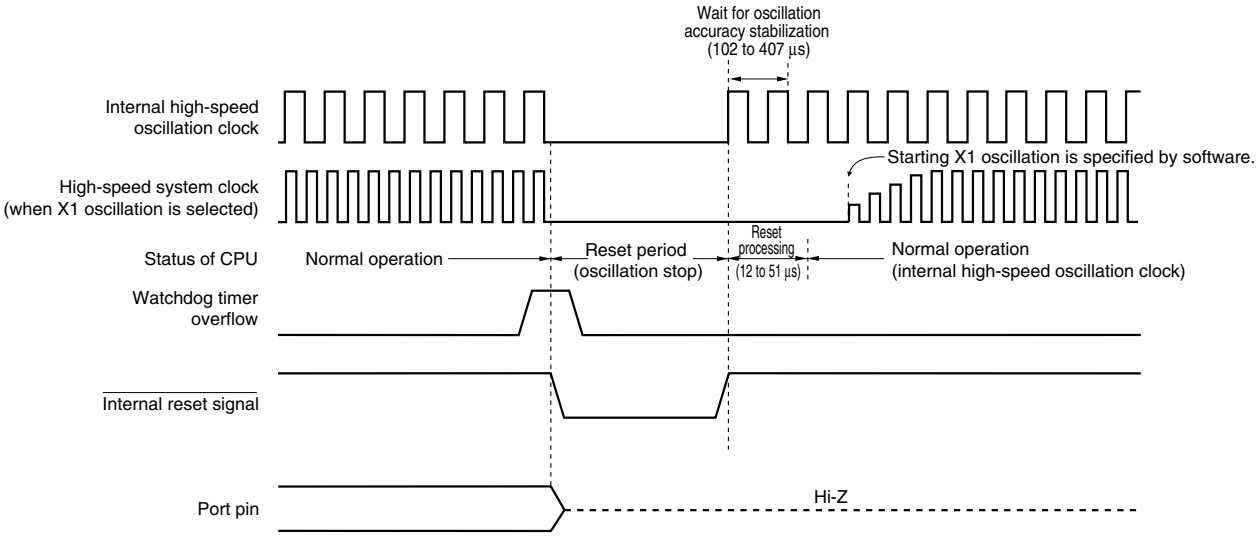
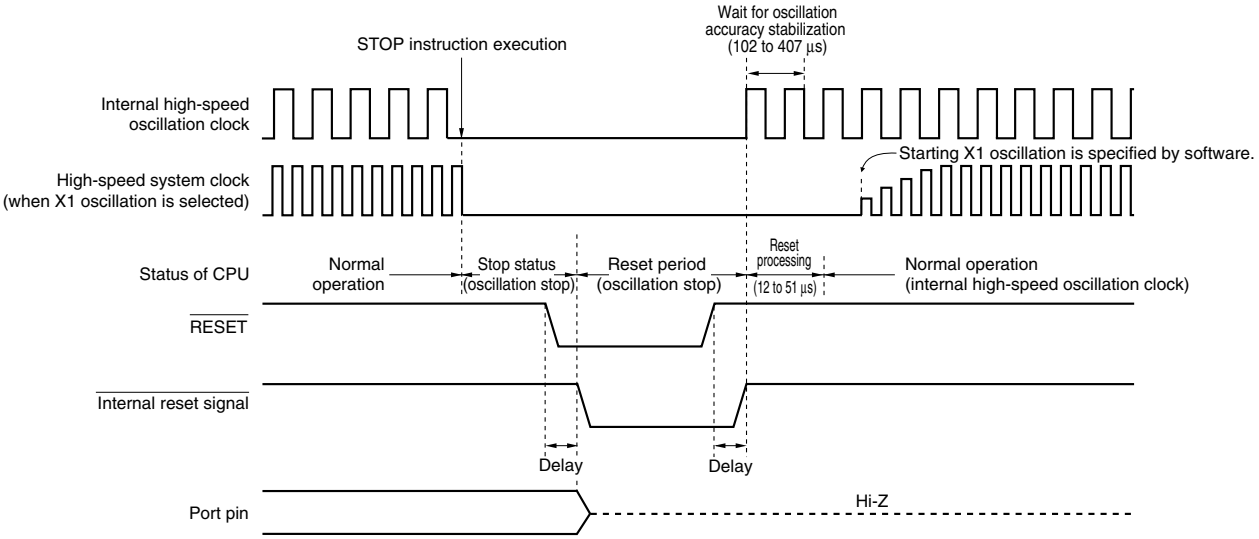


Figure 12-3. Timing of Reset Due to Watchdog Timer Overflow



Caution A watchdog timer internal reset resets the watchdog timer.

Figure 12-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Remark For the reset timing of the power-on-clear circuit and low-voltage detector, refer to **CHAPTER 13 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 14 LOW-VOLTAGE DETECTOR**.

Table 12-1. 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_{EXCLK}	Clock input invalid (EXCLK pin is input port mode)
	f_{IL}		Operation stopped
CPU			
Flash memory			
RAM			Operation stopped (The value, however, is retained when the voltage is at least the power-onclear detection voltage.)
Port (latch)			Operation stopped
16-bit timer/event counter 00			
8-bit timer/event counter51			
8-bit timer		H0	
		H1	
Watchdog timer			
External interrupt			
Power-on-clear function			Operable
Low-voltage detection function			Operation stopped (however, operation continues at LVI reset)
On-chip debug function			Operation stopped

Remark f_{IH}: Internal high-speed oscillation clock, f_X: X1 clock
f_{EXCLK}: External main system clock, f_{IL}: Internal low-speed oscillation clock

Table 12-2. Hardware Statuses After Reset Acknowledgment (1/3)

Hardware		After Reset Acknowledgment[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers 0, 3, 6, 7, 12 (P0, P3, P6, P7, P12) (output latches)		00H
Port mode registers 0, 3, 6, 7, 12 (PM0, PM3, PM6, PM7, PM12)		FFH
Pull-up resistor option registers 0, 3, 6, 7 (PU0, PU3, PU6, PU7)		00H
Pull-up resistor option register 12 (PU12)		20H
Pull-down resistor option registers 6, 7 (PD6, PD7)		00H
Reset pin mode register (RSTMASK)		00H
Internal memory size switching register (IMS)		CFH[Note 3]
Clock operation mode select register (OSCCTL)		00H
Processor clock control register (PCC)		01H
Internal oscillation mode register (RCM)		80H
Main OSC control register (MOC)		80H
Main clock mode register (MCM)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		05H
16-bit timer/event counter 00	Timer counter 00 (TM00)	0000H
	Capture/compare registers 000, 010 (CR000, CR010)	0000H
	Mode control register 00 (TMC00)	00H
	Prescaler mode register 00 (PRM00)	00H
	Capture/compare control register 00 (CRC00)	00H
	Timer output control register 00 (TOC00)	00H
8-bit timer/event counter 51	Timer counter 51 (TM51)	00H
	Compare register 51 (CR51)	00H
	Timer clock selection register 51 (TCL51)	00H
	Mode control register 51 (TMC51)	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. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. Reset signal generation makes the setting of the ROM area undefined. Therefore, set the value corresponding to each product as indicated below after release of reset.

Products	IMS	ROM Capacity	Internal High-Speed RAM Capacity
µPD79F7025	42H	8 KB	512 bytes
µPD79F7026	04H	16 KB	768 bytes

Table 12-2. Hardware Statuses After Reset Acknowledgment (2/3)

Hardware		Status After Reset Acknowledgment[Note 1]
8-bit timer H0, H1	Compare registers 00, 01, 10, 11 (CMP00, CMP01, CMP10, CMP11)	00H
	Mode register (TMHMD0, TMHMD1)	00H
	Carrier control register 1 (TMCYC1)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 2]
Reset function	Reset control flag register (RESF)	00H[Note 3]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 3]
	Low-voltage detection level selection register (LVIS)	00H[Note 3]
Interrupt	Request flag registers 0L, 0H, 1L (IF0L, IF0H, IF1L)	00H
	Mask flag registers 0L, 0H, 1L (MK0L, MK0H, MK1L)	FFH
	Priority specification flag registers 0L, 0H, 1L (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

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

3. These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI (Except Reset by LVI Default Start Function)	Reset by LVI Default Start Function
RESF	WDTRF flag	Cleared (0)	Cleared (0)	Set (1)	Held	Cleared (0)
	LVIRF flag			Held	Set (1)	
LVIM		Cleared (00H)	Cleared (00H)	Cleared (00H)	Held	Cleared (00H)
LVIS						

12.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the µPD79F7025, 79F7026 microcontrollers. 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 12-5. Format of Reset Control Flag Register (RESF)

Address: FFACH After reset: 00H[Note] R

Symbol	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	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.

Note The value after reset varies depending on the reset source.

Caution Do not read data by a 1-bit memory manipulation instruction.

The status of RESF when a reset request is generated is shown in Table 12-3.

Table 12-3. RESF Status When Reset Request Is Generated

Flag \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI (Except Reset by LVI Default Start Function)	Reset by LVI Default Start Function
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held	Cleared (0)
LVIRF			Held	Set (1)	

CHAPTER 13 POWER-ON-CLEAR CIRCUIT

13.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

(1) In 1.59 V POC mode (option byte: LVISTART = 0)

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{POC} = 1.59 V ±0.15 V), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated. It is released when $V_{DD} \geq V_{POC}$.

(2) In 2.7 V/1.59 V POC mode (option byte: LVISTART = 1)

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{DDPOC} = 2.7 V ±0.2 V), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated. It is released when $V_{DD} \geq V_{DDPOC}$.

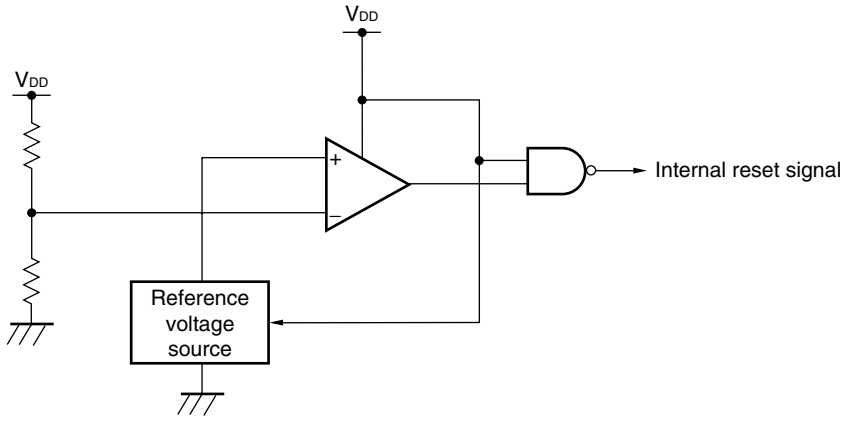
Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

Remark The µPD79F7025, 79F7026 microcontrollers incorporate 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) and low-voltage-detector (LVI). 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, refer to **CHAPTER 12 RESET FUNCTION**.

13.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 13-1.

Figure 13-1. Block Diagram of Power-on-Clear Circuit



13.3 Operation of Power-on-Clear Circuit

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds detection voltage (V_{POC} = 1.59 V ±0.15 V), the reset status is released.

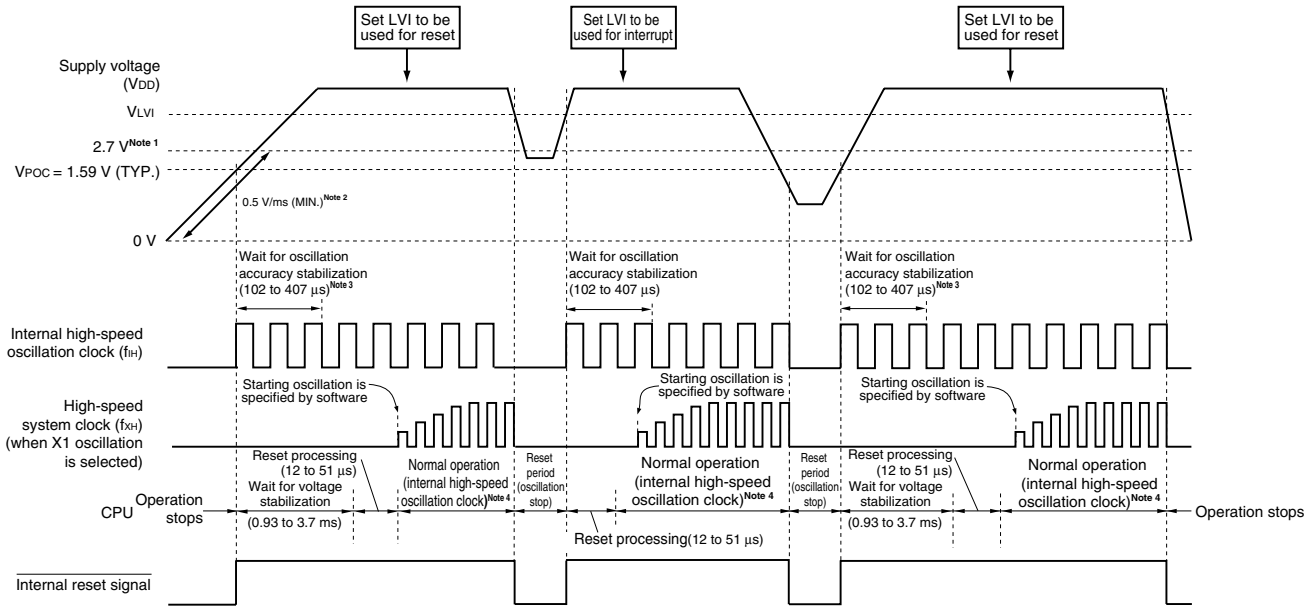
Caution If the LVI default function enabled is set by using an option byte, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.7 V ±0.1 V.

- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{PDR}$, the internal reset signal is generated.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

Figure 13-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) In 1.59 V POC mode (option byte: LVISTART = 0)



Notes 1. The operation guaranteed range is 2.7 V ≤ V_{DD} ≤ 5.5 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.

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

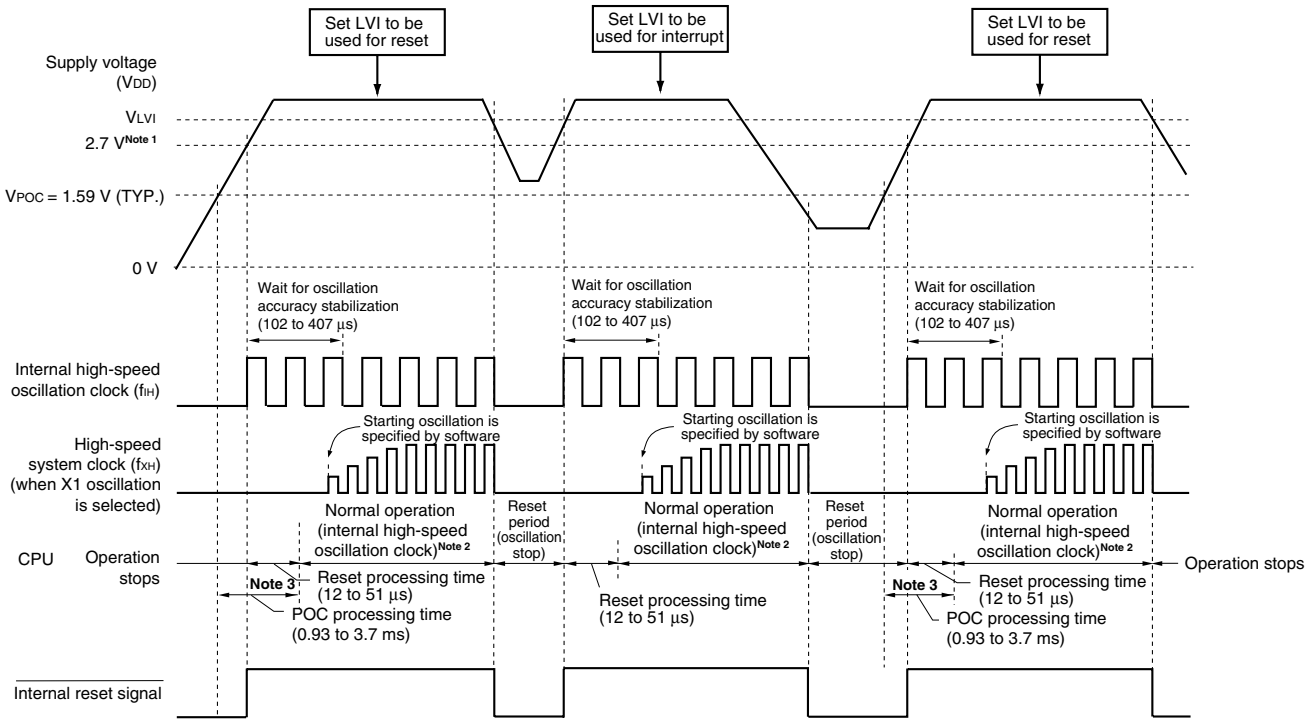
4. The internal high-speed oscillation clock or high-speed system clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time.

Caution Set the low-voltage detector by software after the reset status is released (refer to CHAPTER 14 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POC}: Detection voltage

Figure 13-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) In 2.7 V/1.59 V POC mode (option byte: LVISTART = 1)



Notes 1. The operation guaranteed range is 2.7 V ≤ V_{DD} ≤ 5.5 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 or high-speed system clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time.

3. The following times are required between reaching the POC detection voltage (1.59 V (TYP.)) and starting normal operation.

- When the time to reach 2.7 V (TYP.) from 1.59 V (TYP.) is less than 3.7 ms:
  A processing time of about 1.0 to 3.8 ms is required between reaching 1.59 V (TYP.) and starting normal operation.
- When the time to reach 2.7 V (TYP.) from 1.59 V (TYP.) is greater than 3.7 ms:
  A reset processing time of about 12 to 51 μs is required between reaching 2.7 V (TYP.) and starting normal operation.

Caution Set the low-voltage detector by software after the reset status is released (refer to CHAPTER 14 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POC}: Detection voltage

13.4 Cautions for Power-on-Clear Circuit

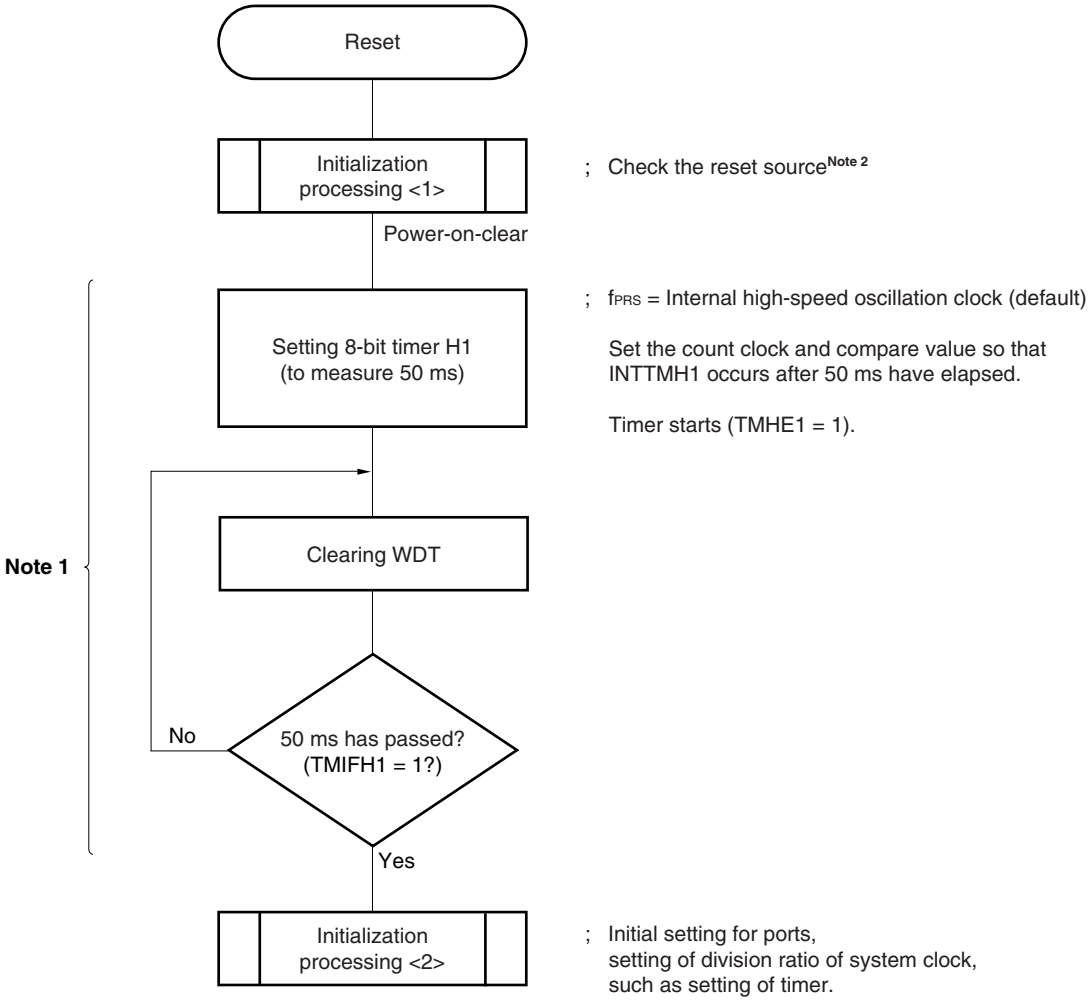
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can 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 13-3. Example of Software Processing After Reset Release (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of detection voltage

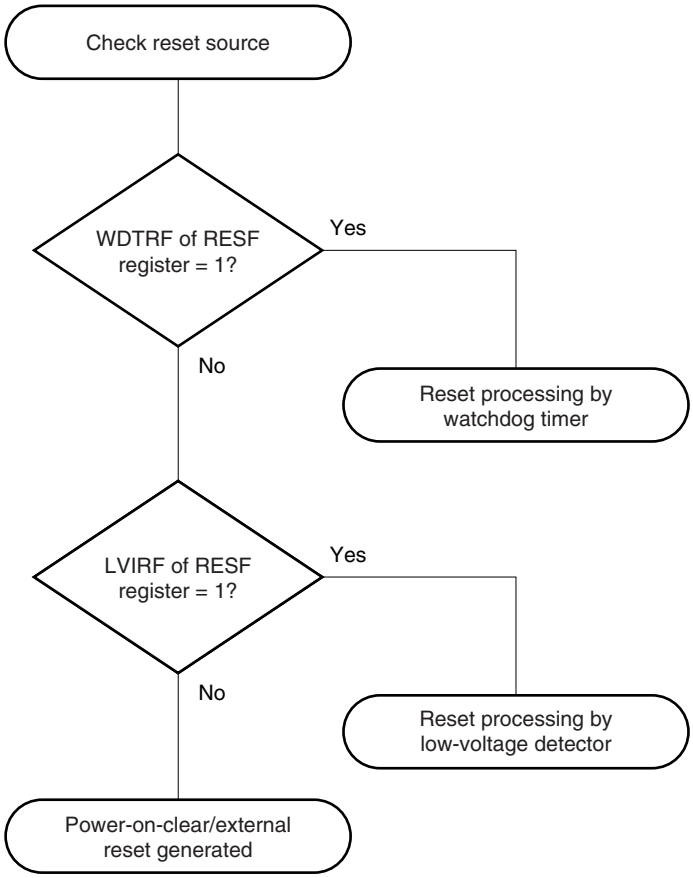


Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Figure 13-3. Example of Software Processing After Reset Release (2/2)

• Checking reset source



CHAPTER 14 LOW-VOLTAGE DETECTOR

14.1 Functions of Low-Voltage Detector

The low-voltage detector has the following functions.

- The LVI circuit compares the supply voltage (V_{DD}) with the LVI detection voltage (V_{LVI}) 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 detection voltage (V_{POC} = 1.59 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < the LVI detection voltage (V_{LVI} = 2.7 V ±0.1 V). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < the LVI detection voltage (V_{LVI} = 2.7 V ±0.1 V).
- A reset or an interrupt can be selected to be generated after detection by software.
- Detection levels (V_{LVI}, 3 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

The reset and interrupt signals are generated as follows depending on selection by software.

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 ($V_{DD} \geq V_{LVI}$).

Remark LVIMD: Bit 1 of LVIM

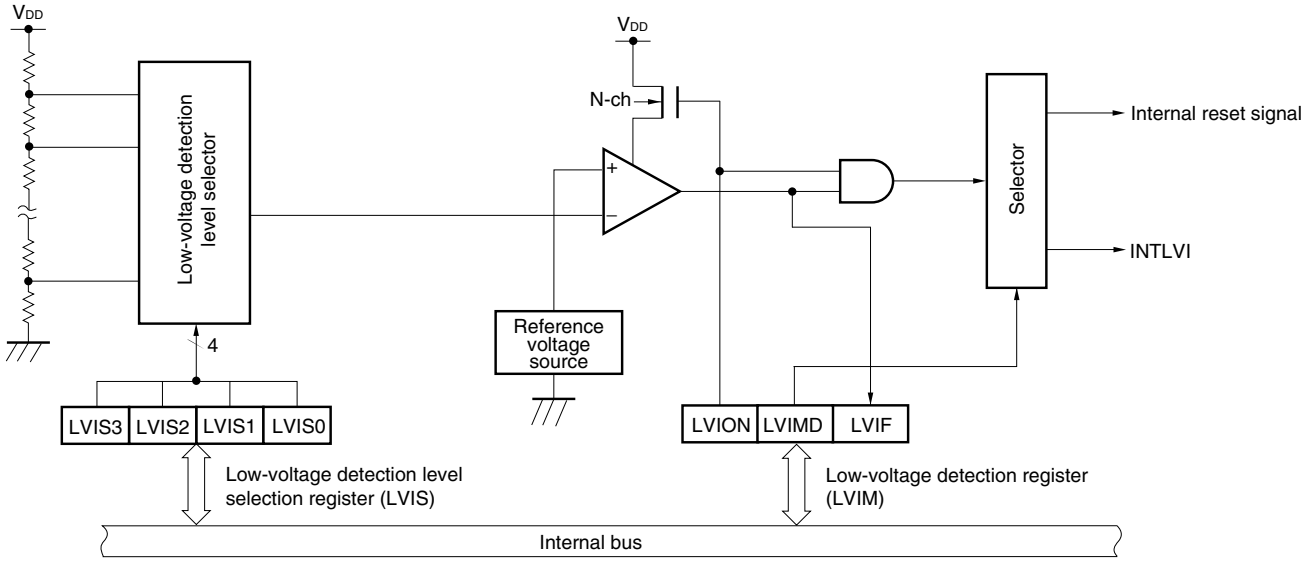
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

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, refer to **CHAPTER 12 RESET FUNCTION**.

14.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 14-1.

Figure 14-1. Block Diagram of Low-Voltage Detector



14.3 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)

(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.

The generation of a reset signal other than an LVI reset clears this register to 00H.

Figure 14-2. Format of Low-Voltage Detection Register (LVIM)

Address: FFBEH After reset: 00H[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION[Notes 3, 4]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVIMD[Note 3]	Low-voltage detection operation mode (interrupt/reset) selection
0	Generates an internal interrupt signal when the supply voltage (V_{DD}) drops lower than the LVI detection voltage (V_{LVI}) ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$).
1	Generates an internal reset signal when the supply voltage (V_{DD}) < the LVI detection voltage (V_{LVI}) and releases the reset signal when $V_{DD} \geq V_{LVI}$.

LVIF	Low-voltage detection flag
0	Supply voltage (V_{DD}) ≥ LVI detection voltage (V_{LVI}), or when LVI operation is disabled
1	Supply voltage (V_{DD}) < LVI detection voltage (V_{LVI})

Notes 1. The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by LVI resets (except resets by the LVI default start function).
The value of this register is reset to “00H” by other resets.

2. Bit 0 is read-only.

3. LVION and LVIMD 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.

4. When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to wait for an operation stabilization time (10 μs (MAX.)) from when LVION is set to 1 until operation is stabilized. After the operation stabilizes, an external input (minimum pulse width: 200 μs) of 200 μs or more is required until LVIF is set (1) after the voltage drops to the LVI detection voltage or less.

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. If LVI operation is disabled (clears LVION) when LVI is used in interrupt mode (LVIMD = 0) and 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.

(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 00H.

Figure 14-3. Format of Low-Voltage Detection Level Select Register (LVIS)

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

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	LVIS1	LVIS0

LVIS1	LVIS0	Detection level
0	0	V_{LVI0} (4.24 ±0.1 V)
0	1	V_{LVI1} (4.09 ±0.1 V)
1	0	V_{LVI2} (3.93 ±0.1 V)
Other than above		Setting prohibited

Note The reset value changes depending on the reset source.

If the LVIS register is reset by LVI resets (except resets by the LVI default start function), it is not reset but holds the current value. The value of this register is reset to “00H” by other resets.

Cautions 1. Be sure to clear bits 2 to 7 to 0.

2. Do not change the value of LVIS during LVI operation.

14.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

(1) Used as reset (LVIMD = 1)

- Compares the supply voltage (V_{DD}) and LVI detection voltage (V_{LVI}), generates an internal reset signal when $V_{DD} < V_{LVI}$, and releases internal reset when $V_{DD} \geq V_{LVI}$.

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 detection voltage (V_{POC} = 1.59 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.70 V ±0.1 V). Operation to be initiated at $V_{DD} \leq 2.7$ V, recommend that LVI default start function is set to ON.

(2) Used as interrupt (LVIMD = 0)

- Compares the supply voltage (V_{DD}) and LVI 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).

While the low-voltage detector is operating, whether the supply voltage 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)

14.4.1 When used as reset

(1) In 1.59 V POC mode (LVISTART = 0)

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set the LVI detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for an operation stabilization time (10 μs (MAX.)).
  - <5> Wait until it is checked that (supply voltage (V_{DD}) ≥ LVI detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.
  - <6> Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected).

  Figure 14-4 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 supply voltage (V_{DD}) ≥ LVI 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:

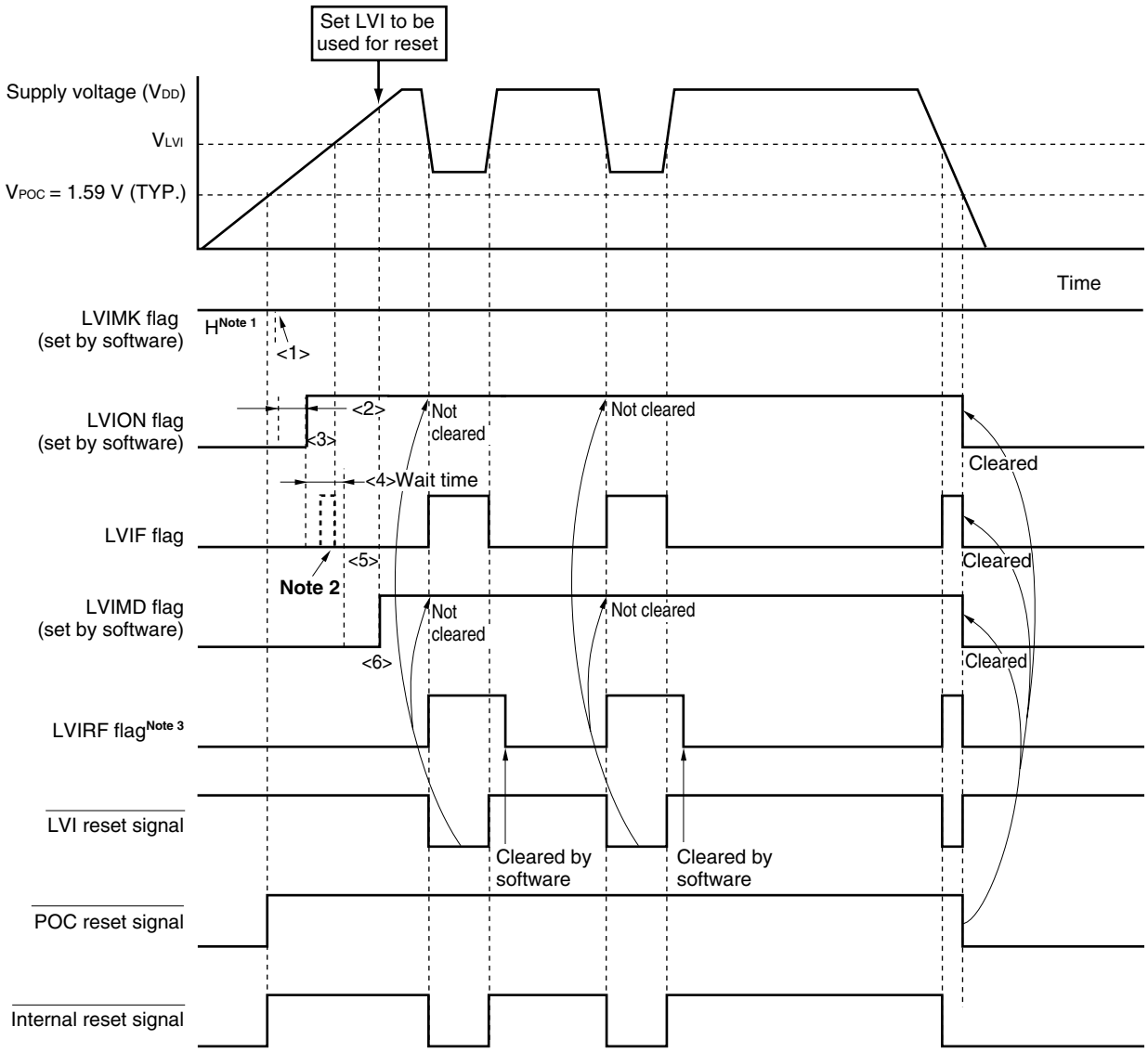
  Write 00H to LVIM.

  • When using 1-bit memory manipulation instruction:

  Clear LVIMD to 0 and then LVION to 0.

Remark Operation to be initiated at V_{DD} ≤ 2.7 V, recommend that LVI default start function is set to ON.

Figure 14-4. Timing of Low-Voltage Detector Internal Reset Signal Generation (LVISTART = 0)



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, refer to **CHAPTER 12 RESET FUNCTION**.

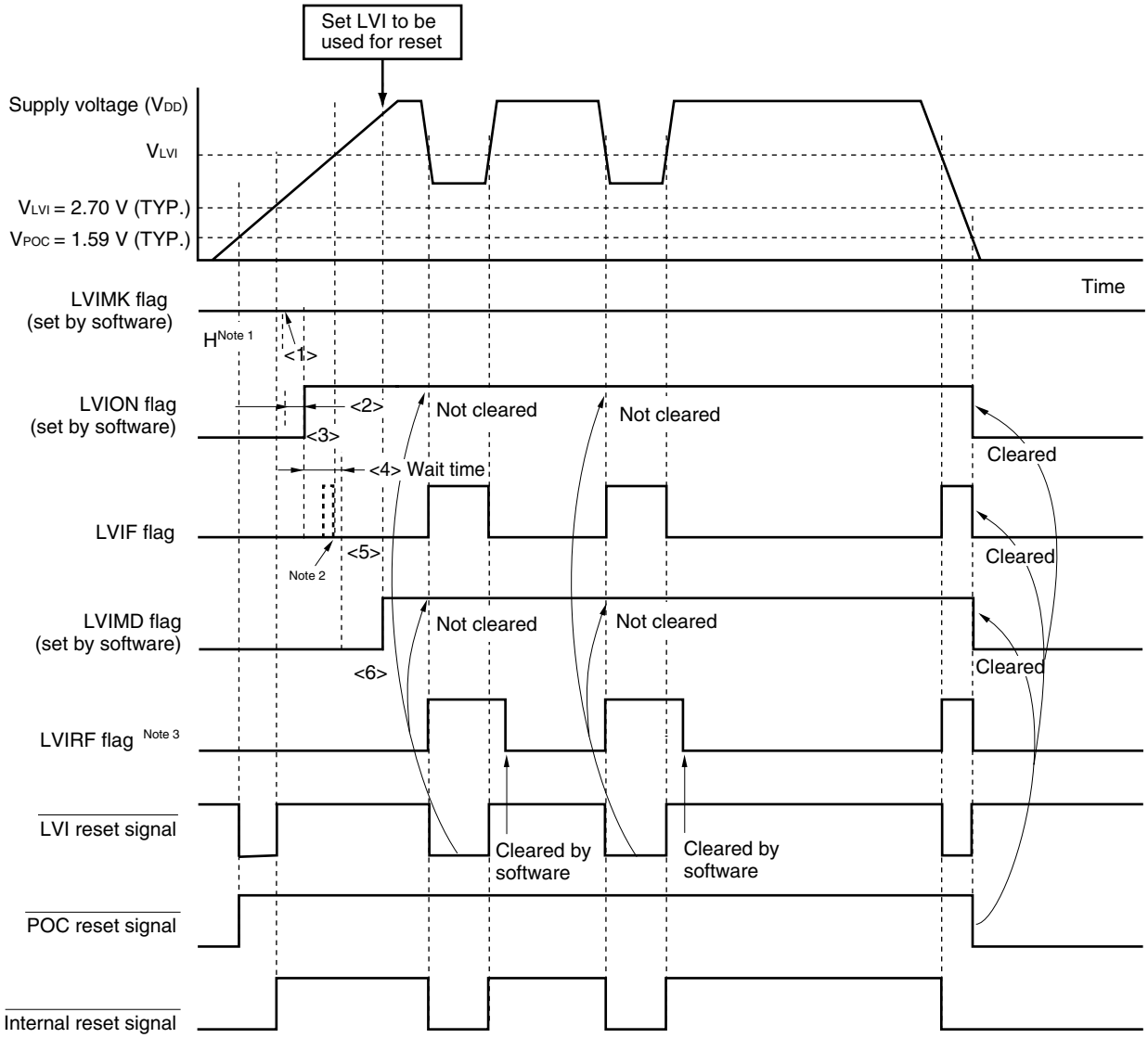
Remarks 1. <1> to <6> in Figure 14-4 above correspond to <1> to <6> in the description of “When starting operation” in **14.4.1 (1) In 1.59 V POC mode (LVISTART = 0).**

2. VPOC: Detection voltage

(2) In 2.7 V/1.59 V POC mode (LVISTART = 1)

The setting when operation starts and when operation stops is the same as that described in **14.4.1 (1) When LVI default start function stopped is set (LVISTART = 0)**.

Figure 14-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (LVISTART = 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, refer to **CHAPTER 12 RESET FUNCTION**.

Remarks 1. <1> to <6> in Figure 14-5 above correspond to <1> to <6> in the description of “When starting operation” in **14.4.1 (1) In 1.59 V POC mode (LVISTART = 0)**.

2. VPOC: Detection voltage

14.4.2 When used as interrupt

(1) When LVI default start function stopped is set (LVISTART = 0)

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set the LVI detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).
  - <3> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <5> Use software to wait for an operation stabilization time (10 μs (MAX.)).
  - <6> Confirm that "supply voltage (V_{DD}) ≥ LVI detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < LVI 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 14-6 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.

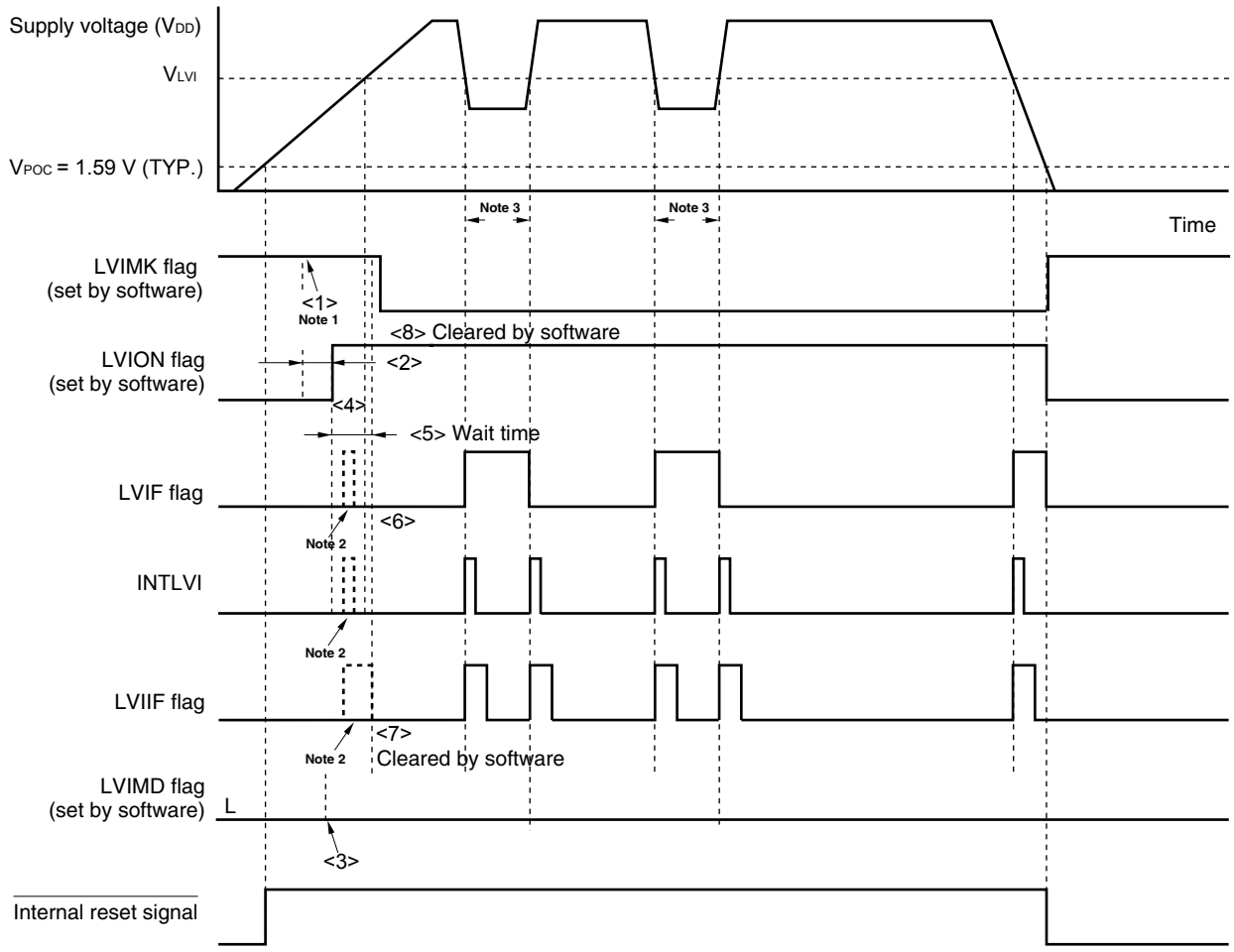
- When using 8-bit memory manipulation instruction:

Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:

Clear LVION to 0.

Figure 14-6. Timing of Low-Voltage Detector Interrupt Signal Generation (LVISTART = 0)



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 (clears LVION) 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.

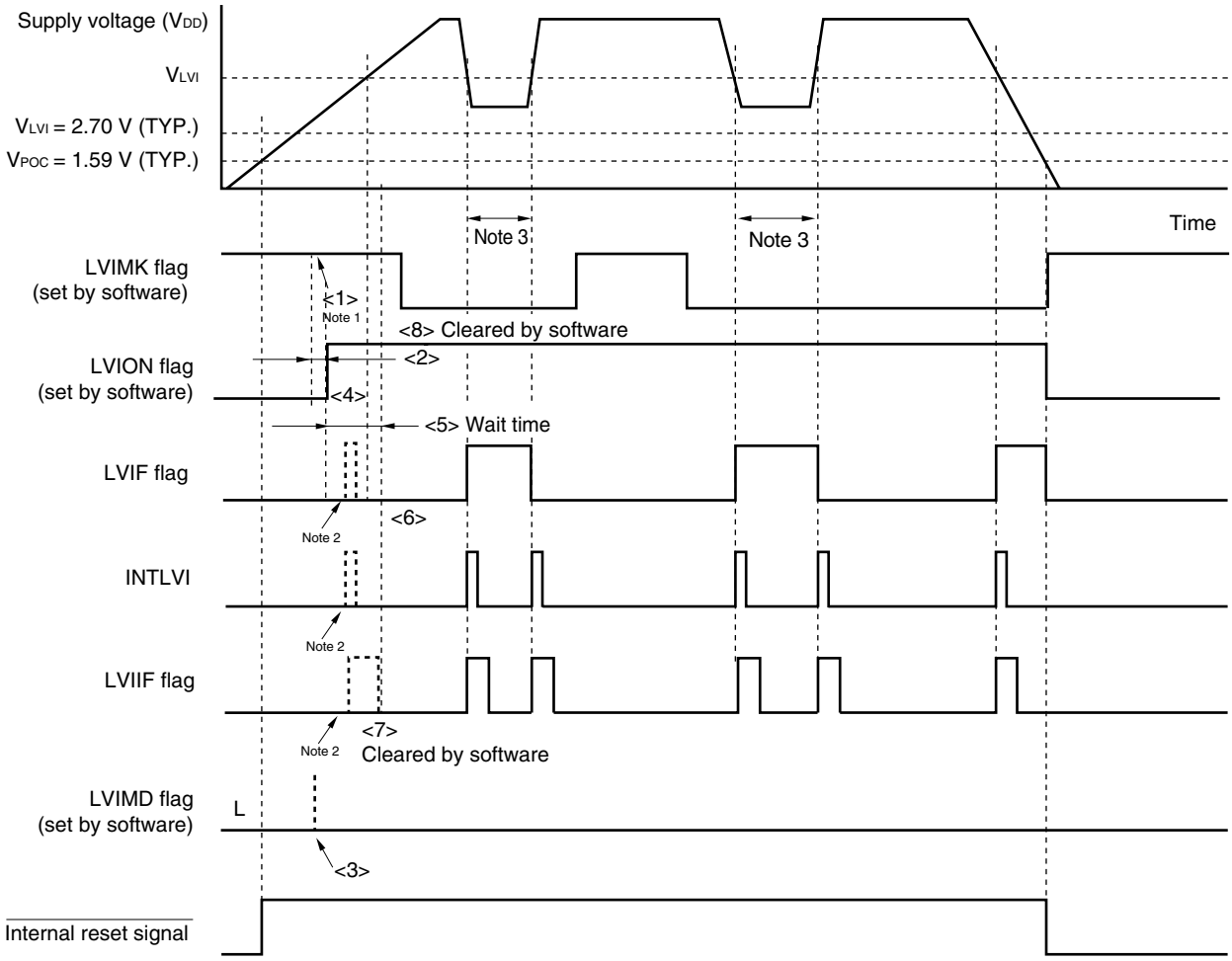
Remarks 1. <1> to <8> in Figure 14-6 above correspond to <1> to <8> in the description of "When starting operation" in **14.4.2 (1) When LVI default start function stopped is set (LVISTART = 0).**

2. V_{POC}: Detection voltage

(2) When LVI default start function enabled is set (LVISTART = 1)

The setting when operation starts and when operation stops is the same as that described in **14.4.2 (1) When LVI default start function stopped is set (LVISTART = 0)**.

Figure 14-7. Timing of Low-Voltage Detector Interrupt Signal Generation (LVISTART = 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. If LVI operation is disabled (clears LVION) 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 14-7 above correspond to <1> to <8> in the description of "When starting operation" in **14.4.2 (1) When LVI default start function enabled is set (LVISTART = 1).**

2. V_{POC}: Detection voltage

14.5 Cautions for Low-Voltage Detector

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.

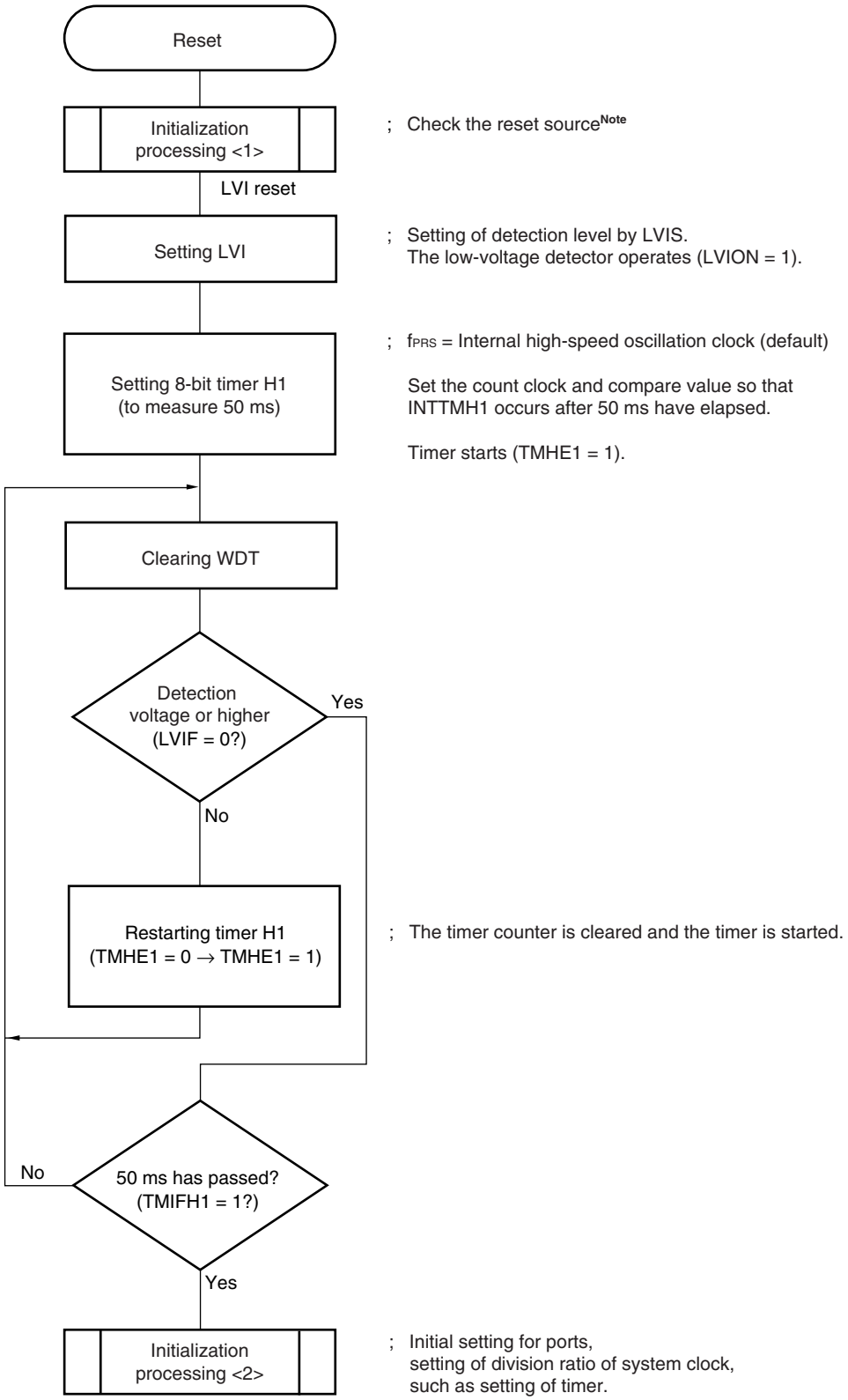
The time from reset release through microcontroller operation start can be set arbitrarily by the following action.

<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 (refer to **Figure 14-8**).

Figure 14-8. Example of Software Processing After Reset Release (1/2)

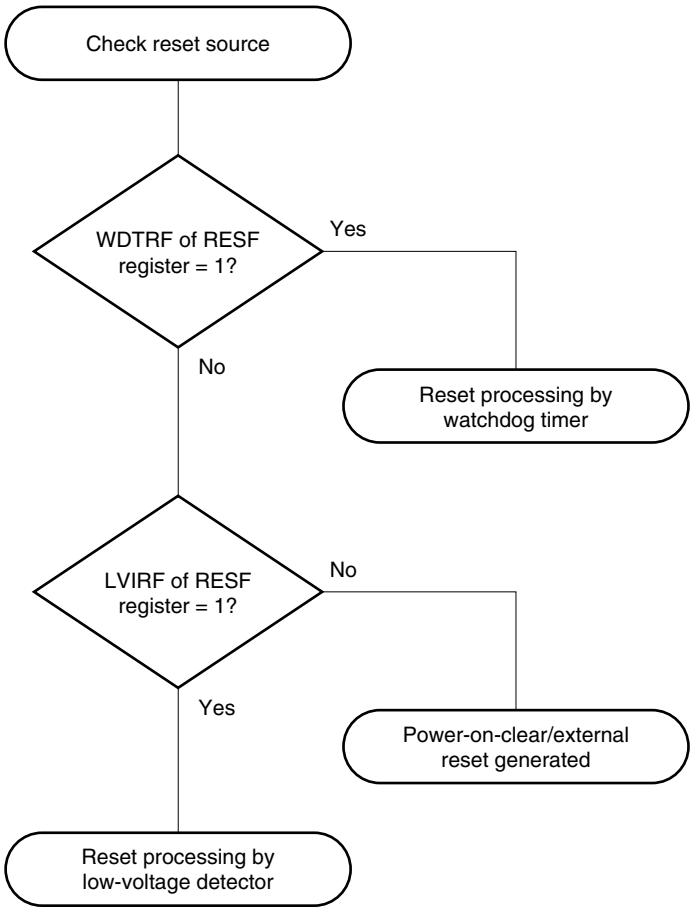
• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



Note A flowchart is shown on the next page.

Figure 14-8. Example of Software Processing After Reset Release (2/2)

• Checking reset source



Operation example 2: When used as interrupt

Interrupt requests may be generated frequently.

Take the following action.

<Action>

Confirm that "supply voltage (V_{DD}) ≥ LVI detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < LVI 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.

CHAPTER 15 REGULATOR

15.1 Regulator Overview

The μPD79F7025, 79F7026 microcontrollers contain a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). 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 2.5 V (TYP.).

CHAPTER 16 OPTION BYTE

16.1 Functions of Option Bytes

The flash memory at 0080H to 0084H of the µPD79F7025, 79F7026 is an option byte area. When power is turned on or when the device is restarted from the reset status, the device automatically references the option bytes and sets specified functions. When using the product, be sure to set the following functions by using the option bytes.

(1) 0080H

- ○ Internal low-speed oscillator operation
  - Can be stopped by software
  - Cannot be stopped
- ○ Watchdog timer interval time setting
- ○ Watchdog timer counter operation
  - Enabled counter operation
  - Disabled counter operation
- ○ Watchdog timer window open period setting

(2) 0081H

- ○ LVI default start operation control
  - During LVI default start function enabled (LVISTART = 1)
    The device is in the reset state after reset release or upon power application and until the supply voltage reaches 2.7 V (TYP.). It is released from the reset state when the voltage exceeds 2.7 V (TYP.).
    If the supply voltage rises to 2.7 V after reset release or power application at a rate slower than 0.5 V/ms (MIN.), LVI default start function operation is recommended.
  - During LVI default start function stopped (LVISTART = 0)
    The device is in the reset state after reset release or upon power application and until the supply voltage reaches 1.59 V (TYP.). It is released from the reset state when the voltage exceeds 1.59 V (TYP.).

Caution LVISTART can only be written by using a dedicated flash memory programmer.

Remark Operation to be initiated at $V_{DD} \leq 2.7$ V, recommend that LVI default start function is set to ON.

(3) 0082H

Be sure to set to 01H.

(4) 0083H

- ○ On-chip debug mode
  - Disabling on-chip debug mode
  - Forcibly setting to on-chip debug mode

- ○ Clock supply to OCD when STOP instruction is executed
  - Supply
  - Stop

(5) 0084H

- ○ On-chip debug operation control
  - Disabling on-chip debug operation
  - Enabling on-chip debug operation and erasing data of the flash memory in case authentication of the on-chip debug security ID fails
  - Enabling on-chip debug operation and not erasing data of the flash memory even in case authentication of the on-chip debug security ID fails

16.2 Format of Option Byte

The format of the option byte is shown below.

Figure 16-1. Format of Option Byte (1/3)

Address: 0080H

7	6	5	4	3	2	1	0
0	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	LSROSC

WINDOW1	WINDOW0	Watchdog timer window open period
0	0	25%
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter/illegal access detection
0	Counter operation disabled (counting stopped after reset), illegal access detection operation disabled
1	Counter operation enabled (counting started after reset), illegal access detection operation enabled

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time
0	0	0	$2^{10}/f_{IL}$ (3.88 ms)
0	0	1	$2^{11}/f_{IL}$ (7.76 ms)
0	1	0	$2^{12}/f_{IL}$ (15.52 ms)
0	1	1	$2^{13}/f_{IL}$ (31.03 ms)
1	0	0	$2^{14}/f_{IL}$ (62.06 ms)
1	0	1	$2^{15}/f_{IL}$ (124.12 ms)
1	1	0	$2^{16}/f_{IL}$ (248.24 ms)
1	1	1	$2^{17}/f_{IL}$ (496.48 ms)

LSROSC	Internal low-speed oscillator operation
0	Can be stopped by software (stopped when 1 is written to bit 1 (LSRSTOP) of RCM register)
1	Cannot be stopped (not stopped even if 1 is written to LSRSTOP bit)

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. If LSROSC = 0 (oscillation can be stopped by software), the count clock is not supplied to the watchdog timer in the HALT and STOP modes, regardless of the setting of bit 0 (LSRSTOP) of the internal oscillation mode register (RCM).

When 8-bit timer H1 operates with the internal low-speed oscillation clock, the count clock is supplied to 8-bit timer H1 even in the HALT/STOP mode.

3. Be sure to clear bit 7 to 0.

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

2. ( ): f_{IL} = 264 kHz (MAX.)

Figure 16-1. Format of Option Byte (2/3)

Address: 0081H[Notes 1, 2]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	LVISTART

LVISTART	LVI default start operation control
0	LVI is OFF by default upon power application (LVI default start function stopped)
1	LVI is ON by default upon power application (LVI default start function enabled)

Notes 1. LVISTART can only be written by using a dedicated flash memory programmer.

2. To change the setting for the LVI default start, set the value to 0081H again after batch erasure (chip erasure) of the flash memory. The setting cannot be changed after the memory of the specified block is erased.

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

Remark Operation to be initiated at $V_{DD} \leq 2.7$ V, recommend that LVI default start function is set to ON.

Address: 0082H[Note]

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

Note Be sure to set to 01H.

Address: 0083H

7	6	5	4	3	2	1	0
0	0	0	OCDCKSTP	1	1	1	OCDONB

OCDCKSTP	Clock supply to OCD when STOP instruction is executed in on-chip debug mode
0	If STOP instruction is executed, internal oscillator is not stopped, and clock supply to OCD is continued. CPU and peripheral circuits are stopped.
1	Internal oscillator is stopped and clock supply to OCD is stopped.

OCDONB	On-chip debug mode
0	Disables on-chip debug mode
1	Forcibly sets to on-chip debug mode

Caution Be sure to clear bits 7 to 5 and 0 to "0" and set bits 3 to 1 to "1".

Figure 16-1. Format of Option Byte (3/3)

Address: 0084H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCDEN1	OCDEN0

OCDEN1	OCDEN0	On-chip debug operation control
0	0	Operation disabled
0	1	Setting prohibited
1	0	Operation enabled. Does not erase data of the flash memory in case authentication of the on-chip debug security ID fails.
1	1	Operation enabled. Erases data of the flash memory in case authentication of the on-chip debug security ID fails.

Caution Be sure to clear bits 7 to 2 to "0".

Remark For the on-chip debug security ID, refer to **CHAPTER 18 ON-CHIP DEBUG FUNCTION**.

Here is an example of description of the software for setting the option bytes.

```
OPT      CSEG   AT 0080H
OPTION:  DB     30H      ; Enables watchdog timer operation (illegal access detection operation),
                         ; Window open period of watchdog timer: 50%,
                         ; Overflow time of watchdog timer: 2^10/fIL,
                         ; Internal low-speed oscillator can be stopped by software.
         DB     00H      ; LVI default start function stopped
         DB     01H      ; Internal high-speed oscillation clock frequency 4 MHz (TYP.)
         DB     1FH      ; Shifting to the on-chip debug mode, after reset release
                         ; When STOP instruction is executed in on-chip debug mode,
                         ; internal oscillator is stopped and clock supply to OCD is stopped.
         DB     02H      ; Operation enabled.  Does not erase data of the flash memory in case
                         ; authentication of the on-chip debug security ID fails.
```

Remark Referencing of the option byte is performed during reset processing. For the reset processing timing, refer to **CHAPTER 12 RESET FUNCTION**.

CHAPTER 17 FLASH MEMORY

The μPD79F7025, 79F7026 incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

17.1 Internal Memory Size Switching Register

Select the internal memory capacity using the internal memory size switching register (IMS).
IMS is set by an 8-bit memory manipulation instruction.
Reset signal generation sets IMS to CFH.

Caution Reset signal generation makes the setting of the ROM area undefined. Therefore, set the value corresponding to each product as indicated Table 17-1 after release of reset.

Figure 17-1. Format of Internal Memory Size Switching Register (IMS)

Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	0	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM1	RAM0	Internal high-speed RAM capacity selection
0	0	768 bytes
1	0	512 bytes
Other than above		Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
0	0	1	0	8 KB
0	1	0	0	16 KB
1	1	1	1	(Default value)
Other than above				Setting prohibited

Table 17-1. Set Values of Internal Memory Size Switching Register

Products	IMS Setting
μPD79F7025	42H
μPD79F7026	04H

17.2 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 μPD79F7025, 79F7026 have been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

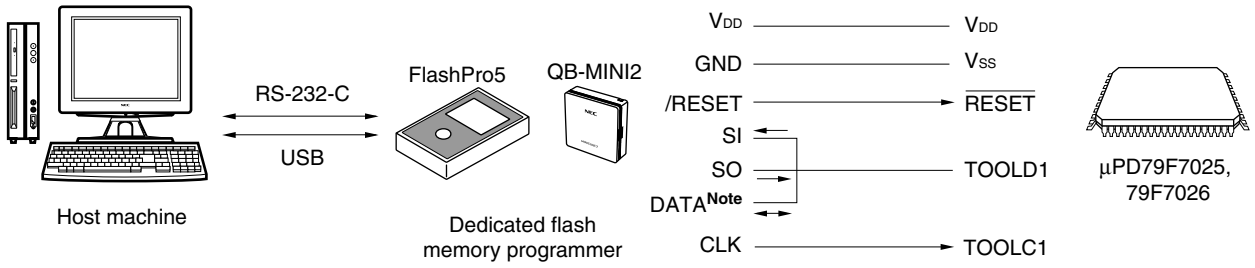
Data can be written to the flash memory with a dedicated program adapter (FA series) before the μPD79F7025, 79F7026 are mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

17.3 Programming Environment

The environment required for writing a program to the flash memory of the µPD79F7025, 79F7026 are illustrated below.

Figure 17-2. Environment for Writing Program to Flash Memory



Note QB-MINI2 only

A host machine that controls the dedicated flash memory programmer is necessary.

To interface between the dedicated flash memory programmer and the µPD79F7025, 79F7026, the TOOLD0 pins 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.

Table 17-2. Pin Connection

Dedicated Flash memory programmer			µPD79F7025, 79F7026 microcontrollers
Signal Name	I/O	Pin Function	Pin Name
CLK	Output	Clock output to µPD79F7025, 79F7026 microcontrollers	TOOLC0/
SI	Input	Receive signal	TOOLD0
SO	Output	Transmit signal	
DATA[Note]	I/O	Input/output signal for data communication during debugging	
/RESET	Output	Reset signal	$\overline{\text{RESET}}$
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND	–	Ground	V_{SS}

Note QB-MINI2 only

17.4 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.

17.4.1 TOOL pins

The pins used for communication in flash memory programming mode are shown in the table below.

Table 17-3. Pins Used for Communication in Flash Memory Programming Mode

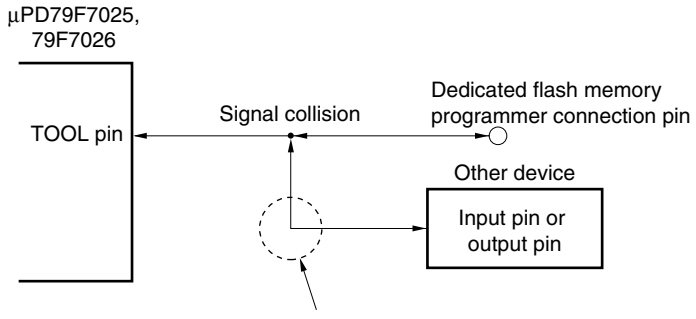
Pin Name	Connection of Pins
TOOLC0	Connect this pin directly to the dedicated flash memory programmer or pull it down by connecting it to V_{SS} via a resistor (10 kΩ)
TOOLD0	Connect this pin directly to the dedicated flash memory programmer or pull it up by connecting it to V_{DD} via a resistor (3 k to 10 kΩ)

To connect the dedicated flash memory programmer to the pins of a serial interface that is connected to another device on the board, care must be exercised so that signals do not collide or that the other device does not malfunction.

(1) Signal collision

If the dedicated flash memory programmer is connected to the TOOL pin that is connected to another device, signal collision takes place. To avoid this collision, either isolate the connection with the other device, or make the other device go into a high-impedance state.

Figure 17-3. Signal Collision (TOOL Pin)



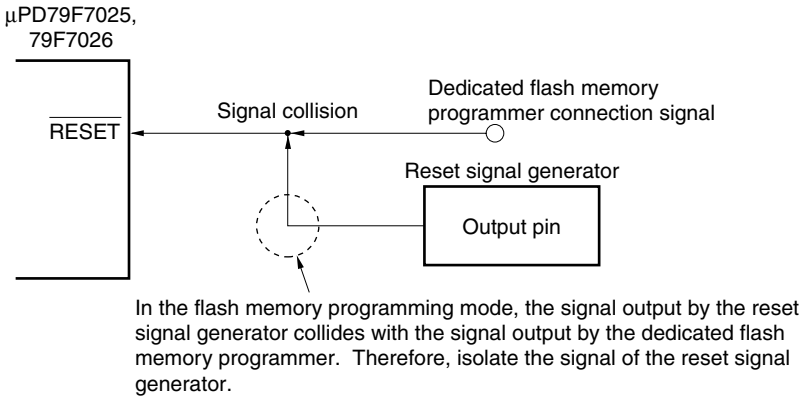
In the flash memory programming mode, the signal of the other device collides with the signal of the dedicated flash programmer. Therefore, isolate the signal of the other device.

17.4.2 $\overline{\text{RESET}}$ pin

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, signal collision takes place. To prevent this collision, isolate the connection with the reset signal generator.

If the reset signal is input from the user system while the flash memory programming mode is set, the flash memory will not be correctly programmed. Do not input any signal other than the reset signal of the dedicated flash memory programmer.

Figure 17-4. Signal Collision ($\overline{\text{RESET}}$ Pin)



17.4.3 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD} or V_{SS} via a resistor.

17.4.4 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF) in the same manner as during normal operation. 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.

17.4.5 Other signal pins

Connect X1, 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.

17.4.6 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

However, 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, even when using the on-board supply voltage.

Supply the same other power supplies (V_{SS}) as those in the normal operation mode.

17.4.7 On-board writing when connecting crystal/ceramic resonator

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 processed as described below.

When using the X1 (TOOLC0) and X2 (TOOLD0) pins as the serial interface for flash memory programming, signals will collide if an external device is connected. To prevent the conflict of signals, isolate the connection with the external device.

Similarly, when a capacitor is connected to the X1 and X2 pins, the waveform during communication is changed, and thus communication may be disabled depending on the capacitor capacitance. Make sure to isolate the connection with the capacitor during flash programming.

In cases when a crystal or ceramic resonator has been selected to generate the system clock, and the decision has been made to execute on-board flash programming with the resonator mounted on the device because it is difficult to isolate the resonator, be sure to thoroughly evaluate the flash memory programming with the resonator mounted on the device before executing the processing described next.

- Mount the minimum-possible test pads between the device and the resonator, and connect the programmer via the test pad. Keep the wiring as short as possible (refer to **Figure 17-5** and **Table 17-4**).

Figure 17-5. Example of Mounting Test Pads

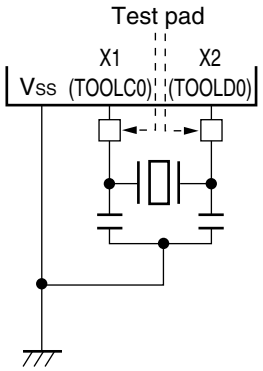


Table 17-4. Clock to Be Used and Mounting of Test Pads

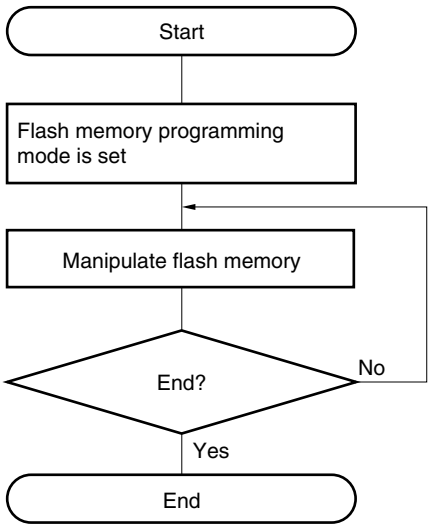
Clock to Be Used		Mounting of Test Pads
High-speed internal oscillation clock		Not required
External clock		
Crystal/ceramic oscillation clock	Before resonator is mounted	
	After resonator is mounted	Required

17.5 Programming Method

17.5.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 17-6. Flash Memory Manipulation Procedure



17.5.2 Flash memory programming mode

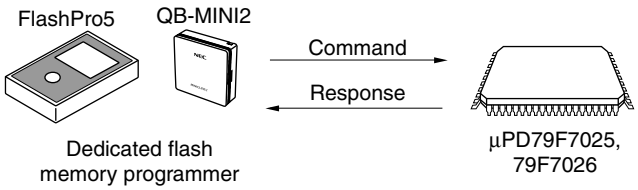
To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the µPD79F7025, 79F7026 in the flash memory programming mode. The system switches to the flash memory programming mode once the dedicated flash memory programmer is connected and communication starts.

Change the mode by using a jumper when writing the flash memory on-board.

17.5.3 Communication commands

The µPD79F7025, 79F7026 communicate with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the µPD79F7025, 79F7026 are called commands, and the signals sent from the µPD79F7025, 79F7026 to the dedicated flash memory programmer are called response.

Figure 17-7. Communication Commands



The flash memory control commands of the µPD79F7025, 79F7026 are listed in the table below. All these commands are issued from the programmer and the µPD79F7025, 79F7026 perform processing corresponding to the respective commands.

Table 17-5. 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 μPD79F7025, 79F7026 information (such as the part number and flash memory configuration).
	Version Get	Gets the μPD79F7025, 79F7026 version and 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 μPD79F7025, 79F7026 return a response for the command issued by the dedicated flash memory programmer. The response names sent from the μPD79F7025, 79F7026 are listed below.

Table 17-6. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

CHAPTER 18 ON-CHIP DEBUG FUNCTION

18.1 Connecting QB-MINI2 to µPD79F7025, 79F7026

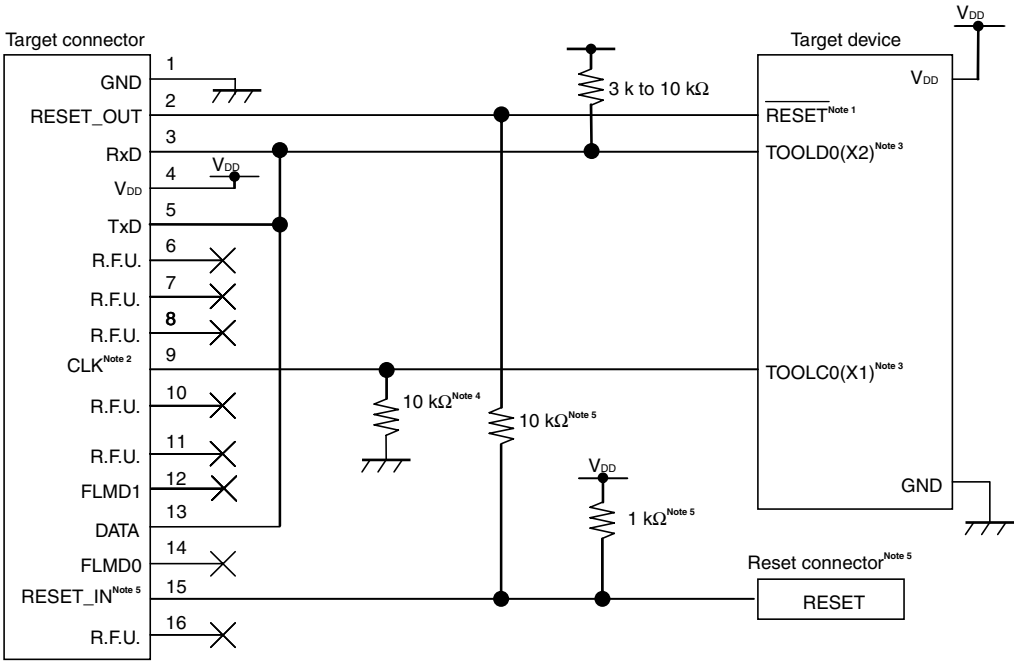
The µPD79F7025, 79F7026 use the V_{DD}, $\overline{RESET}$, TOOLC0/X1, TOOLD0/X2, and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2).

Cautions 1. The µPD79F7025, 79F7026 have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. When transitioning to STOP mode during on-chip debugging, oscillation of the internal high-speed oscillator continues, but the on-chip debug operation is not affected.

Figure 18-1. Connection Example of QB-MINI2 and µPD79F7025, 79F7026 (1/2)

(1) When using the TOOLC0 and TOOLD0 pins (X1 oscillator or EXCLK input clock is not used, both debugging and programming are performed)



Notes 1. If there are capacitance elements such as capacitors, on-chip debugging might not operate normally.

2. A clock signal provided on the 78K0-OCD board, a 4 or 8 MHz clock signal generated in QB-MINI2, or the clock signal generated by the internal high-speed oscillator of the device can be used for the clock signal of the target device during on-chip debugging.
Only the internal high-speed oscillator of the device can be used during flash programming.

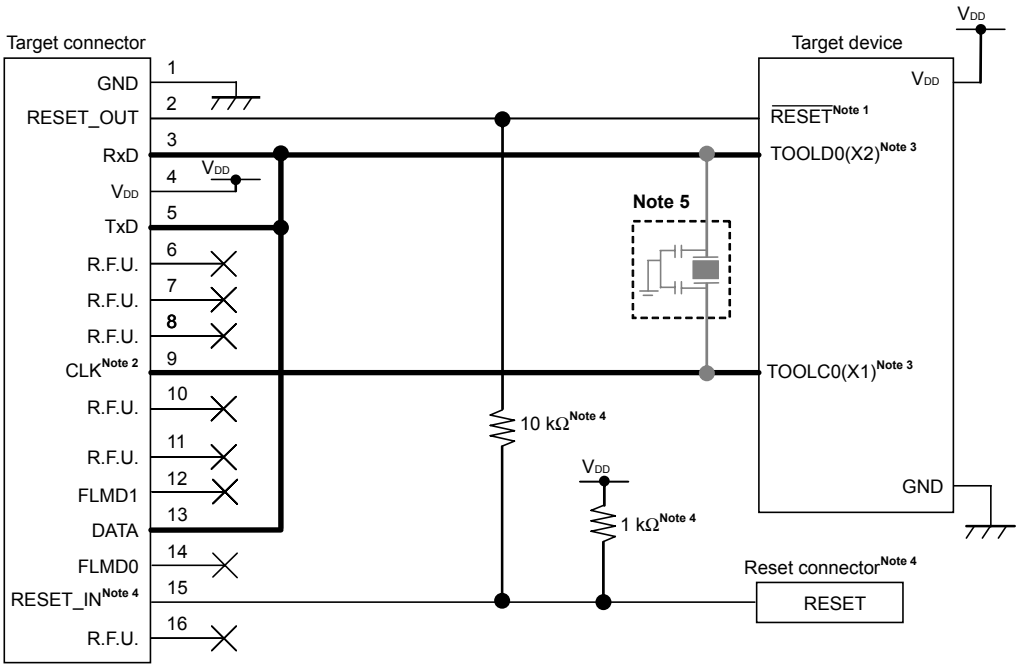
3. During on-chip debugging, the settings specified by the user program are ignored, because these pins are used as pins dedicated to on-chip debugging. However, if the pins are specified as input pins, the pins must be processed (because they are left open when QB-MINI2 is not connected.)

4. This is the processing for the pin that is unused (the input is left open) when the target device operates (when QB-MINI2 is not connected). (This processing is not required if an oscillator circuit is used.)

5. This connection is designed assuming that the reset signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less).

Figure 18-1. Connection Example of QB-MINI2 and µPD79F7025, 79F7026 (2/2)

(2) When using the TOOLC0 and TOOLD0 pins (with X1/X2 oscillator is used, both debugging and programming are performed)



Notes 1. If there are capacitance elements such as capacitors, on-chip debugging might not operate normally.

2. A clock signal provided on the 78K0-OCD board, a 4 or 8 MHz clock signal generated in QB-MINI2, or the clock signal generated by the internal high-speed oscillator of the device can be used for the clock signal of the target device during on-chip debugging.
Only the internal high-speed oscillator of the device can be used during flash programming.

3. During on-chip debugging, the settings specified by the user program are ignored, because these pins are used as pins dedicated to on-chip debugging. However, if the pins are specified as input pins, the pins must be processed (because they are left open when QB-MINI2 is not connected.)

4. This connection is designed assuming that the reset signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **4.1.3 Connection of reset pin** of QB-MINI2 On-Chip Debug Emulator with Programming Function (18371E).

5. Never connect an oscillation circuit to the 78K0-OCD board during on-chip debugging and flash programming.
To prevent an oscillation circuit from not oscillating due to wiring capacitance when the target device operates (when QB-MINI2 is not connected), also consider countermeasures such as disconnecting the oscillation circuit from the target connectors by setting the jumpers.
A program that was downloaded using the debugger does not operate when QB-MINI2 is not connected.

Caution The bold lines in the figure (TOOLD0 and TOOLC0) must be designed so that the device pins are less than 30 mm from the QB-MINI2 connectors or the paths must be shielded by connecting them to GND.

18.2 On-Chip Debug Security ID

The μPD79F7025, 79F7026 have an on-chip debug operation control bit in the flash memory at 0084H (refer to **CHAPTER 16 OPTION BYTE**) and an on-chip debug security ID setting area at 0085H to 008EH, to prevent third parties from reading memory content.

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 18-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
0085H to 008EH	Any ID code of 10 bytes
1085H to 108EH	

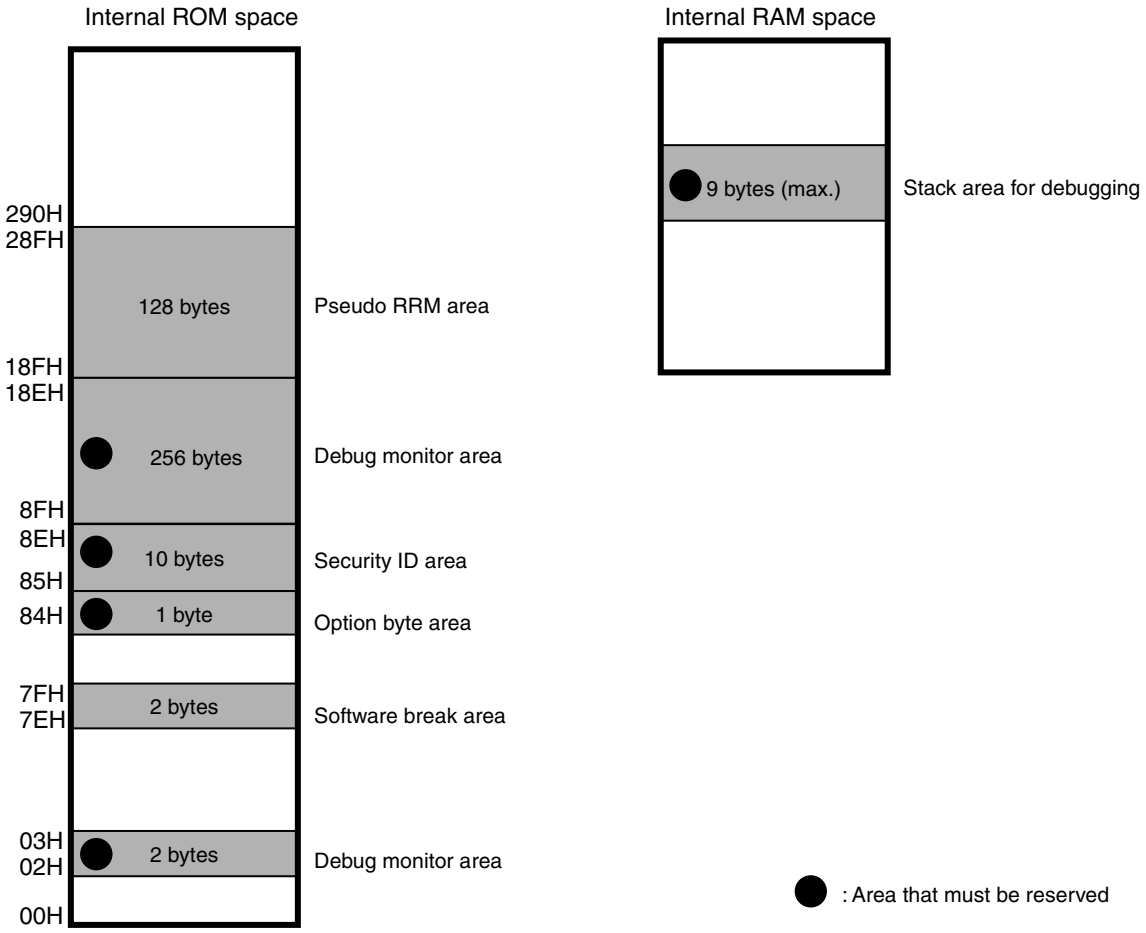
18.3 Securing of User Resources

QB-MINI2 uses the user memory spaces (shaded portions in Figure 18-2) to implement communication with the target device, or each debug functions. The areas marked with a dot (•) are always used for debugging, and other areas are used for each debug function used.

These areas can be secured by using user programs or the linker option.

For details on the securing of these areas, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

Figure 18-2. Reserved Area Used by QB-MINI2



CHAPTER 19 INSTRUCTION SET

This chapter lists each instruction set of the µPD79F7025, 79F7026 in table form. For details of each operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U11926E)**.

19.1 Conventions Used in Operation List

19.1.1 Operand identifiers and specification methods

Operands are written in the "Operand" column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more methods, select one of them. Uppercase letters and the symbols #, !, $ and [ ] are keywords and must be written as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] 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 specification.

Table 19-1. Operand Identifiers and Specification Methods

Identifier	Specification Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special function register symbol[Note]
sfrp	Special function register symbol (16-bit manipulatable register even addresses only)[Note]
saddr	FE20H to FF1FH Immediate data or labels
saddrp	FE20H to FF1FH Immediate data or labels (even address only)
addr16	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions)
addr11	0800H to 0FFFH Immediate data or labels
addr5	0040H to 007FH Immediate data or labels (even address only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, refer to **Table 3-6 Special Function Register List**.

19.1.2 Description of operation column

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
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: Higher 8 bits and lower 8 bits of 16-bit register
∧: Logical product (AND)
∨: Logical sum (OR)
∨: Exclusive logical sum (exclusive OR)
$\overline{\ \ }$: Inverted data
addr16: 16-bit immediate data or label
jdisp8: Signed 8-bit data (displacement value)

19.1.3 Description of flag operation column

(Blank): Not affected
0: Cleared to 0
1: Set to 1
×: Set/cleared according to the result
R: Previously saved value is restored

19.2 Operation List

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	–	7	sfr ← byte	
		A, r Note 3	1	2	–	A ← r	
		r, A Note 3	1	2	–	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	–	5	A ← sfr	
		sfr, A	2	–	5	sfr ← A	
		A, !addr16	3	8	9	A ← (addr16)	
		!addr16, A	3	8	9	(addr16) ← A	
		PSW, #byte	3	–	7	PSW ← byte	× × ×
		A, PSW	2	–	5	A ← PSW	
		PSW, A	2	–	5	PSW ← A	× × ×
		A, [DE]	1	4	5	A ← (DE)	
		[DE], A	1	4	5	(DE) ← A	
		A, [HL]	1	4	5	A ← (HL)	
		[HL], A	1	4	5	(HL) ← A	
		A, [HL + byte]	2	8	9	A ← (HL + byte)	
		[HL + byte], A	2	8	9	(HL + byte) ← A	
		A, [HL + B]	1	6	7	A ← (HL + B)	
		[HL + B], A	1	6	7	(HL + B) ← A	
		A, [HL + C]	1	6	7	A ← (HL + C)	
		[HL + C], A	1	6	7	(HL + C) ← A	
	XCH	A, r Note 3	1	2	–	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	–	6	A ↔ (sfr)	
		A, !addr16	3	8	10	A ↔ (addr16)	
		A, [DE]	1	4	6	A ↔ (DE)	
		A, [HL]	1	4	6	A ↔ (HL)	
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)	
		A, [HL + B]	2	8	10	A ↔ (HL + B)	
		A, [HL + C]	2	8	10	A ↔ (HL + C)	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM 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 processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	–	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	–	8	AX ← sfrp	
		sfrp, AX	2	–	8	sfrp ← AX	
		AX, rp Note 3	1	4	–	AX ← rp	
		rp, AX Note 3	1	4	–	rp ← AX	
		AX, !addr16	3	10	12	AX ← (addr16)	
		!addr16, AX	3	10	12	(addr16) ← AX	
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp	
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r	× × ×
		r, A	2	4	–	r, CY ← r + A	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr)	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16)	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL)	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	× × ×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r + CY	× × ×
		r, A	2	4	–	r, CY ← r + A + CY	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16) + C	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	× × ×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	× × ×
		A, r Note 3	2	4	–	A, CY ← A – r	× × ×
		r, A	2	4	–	r, CY ← r – A	× × ×
		A, saddr	2	4	5	A, CY ← A – (saddr)	× × ×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	× × ×
		A, [HL]	1	4	5	A, CY ← A – (HL)	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	× × ×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	× × ×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	× × ×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	× × ×
		r, A	2	4	–	r, CY ← r – A – CY	× × ×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	× × ×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	× × ×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	× × ×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	× × ×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	× × ×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×
		A, r Note 3	2	4	–	A ← A ∧ r	×
		r, A	2	4	–	r ← r ∧ A	×
		A, saddr	2	4	5	A ← A ∧ (saddr)	×
		A, !addr16	3	8	9	A ← A ∧ (addr16)	×
		A, [HL]	1	4	5	A ← A ∧ (HL)	×
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	×
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	×
		A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM 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 processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A ∨ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∨ byte	×
		A, r Note 3	2	4	–	A ← A ∨ r	×
		r, A	2	4	–	r ← r ∨ A	×
		A, saddr	2	4	5	A ← A ∨ (saddr)	×
		A, !addr16	3	8	9	A ← A ∨ (addr16)	×
		A, [HL]	1	4	5	A ← A ∨ (HL)	×
		A, [HL + byte]	2	8	9	A ← A ∨ (HL + byte)	×
		A, [HL + B]	2	8	9	A ← A ∨ (HL + B)	×
		A, [HL + C]	2	8	9	A ← A ∨ (HL + C)	×
	XOR	A, #byte	2	4	–	A ← A ⊻ byte	×
		saddr, #byte	3	6	8	(saddr) ← (saddr) ⊻ byte	×
		A, r Note 3	2	4	–	A ← A ⊻ r	×
		r, A	2	4	–	r ← r ⊻ A	×
		A, saddr	2	4	5	A ← A ⊻ (saddr)	×
		A, !addr16	3	8	9	A ← A ⊻ (addr16)	×
		A, [HL]	1	4	5	A ← A ⊻ (HL)	×
		A, [HL + byte]	2	8	9	A ← A ⊻ (HL + byte)	×
		A, [HL + B]	2	8	9	A ← A ⊻ (HL + B)	×
		A, [HL + C]	2	8	9	A ← A ⊻ (HL + C)	×
	CMP	A, #byte	2	4	–	A – byte	× × ×
		saddr, #byte	3	6	8	(saddr) – byte	× × ×
		A, r Note 3	2	4	–	A – r	× × ×
		r, A	2	4	–	r – A	× × ×
		A, saddr	2	4	5	A – (saddr)	× × ×
		A, !addr16	3	8	9	A – (addr16)	× × ×
		A, [HL]	1	4	5	A – (HL)	× × ×
		A, [HL + byte]	2	8	9	A – (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A – (HL + B)	× × ×
		A, [HL + C]	2	8	9	A – (HL + C)	× × ×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM 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 processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	× × ×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	× × ×
	CMPW	AX, #word	3	6	–	AX – word	× × ×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X	
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C	
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	× ×
		saddr	2	4	6	(saddr) ← (saddr) + 1	× ×
	DEC	r	1	2	–	r ← r – 1	× ×
		saddr	2	4	6	(saddr) ← (saddr) – 1	× ×
	INCW	rp	1	4	–	rp ← rp + 1	
	DECW	rp	1	4	–	rp ← rp – 1	
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time	×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time	×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time	×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time	×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$	
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$	
BCD adjustment	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	× × ×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	× × ×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← sfr.bit	×
		CY, A.bit	2	4	–	CY ← A.bit	×
		CY, PSW.bit	3	–	7	CY ← PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← (HL).bit	×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY	
		sfr.bit, CY	3	–	8	sfr.bit ← CY	
		A.bit, CY	2	4	–	A.bit ← CY	
		PSW.bit, CY	3	–	8	PSW.bit ← CY	× ×
		[HL].bit, CY	2	6	8	(HL).bit ← CY	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit	×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit	×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit	×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)	×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit	×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit	×
		CY, PSW. bit	3	–	7	CY ← CY ⊻ PSW.bit	×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit	×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1	
		sfr.bit	3	–	8	sfr.bit ← 1	
		A.bit	2	4	–	A.bit ← 1	
		PSW.bit	2	–	6	PSW.bit ← 1	× × ×
		[HL].bit	2	6	8	(HL).bit ← 1	
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0	
		sfr.bit	3	–	8	sfr.bit ← 0	
		A.bit	2	4	–	A.bit ← 0	
		PSW.bit	2	–	6	PSW.bit ← 0	× × ×
		[HL].bit	2	6	8	(HL).bit ← 0	
	SET1	CY	1	2	–	CY ← 1	1
	CLR1	CY	1	2	–	CY ← 0	0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Call/return	**CALL**	!addr16	3	7	–	(SP – 1) ← (PC + 3)H, (SP – 2) ← (PC + 3)L, PC ← addr16, SP ← SP – 2	
	CALLF	!addr11	2	5	–	(SP – 1) ← (PC + 2)H, (SP – 2) ← (PC + 2)L, PC15 – 11 ← 00001, PC10 – 0 ← addr11, SP ← SP – 2	
	CALLT	[addr5]	1	6	–	(SP – 1) ← (PC + 1)H, (SP – 2) ← (PC + 1)L, PCH ← (addr5 + 1), PCL ← (addr5), SP ← SP – 2	
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 1)H, (SP – 3) ← (PC + 1)L, PCH ← (003FH), PCL ← (003EH), SP ← SP – 3, IE ← 0	
	RET		1	6	–	PCH ← (SP + 1), PCL ← (SP), SP ← SP + 2	
	RETI		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R
	RETB		1	6	–	PCH ← (SP + 1), PCL ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R
Stack manipulate	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1	
		rp	1	4	–	(SP – 1) ← rpH, (SP – 2) ← rpL, SP ← SP – 2	
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R R R
		rp	1	4	–	rpH ← (SP + 1), rpL ← (SP), SP ← SP + 2	
	MOVW	SP, #word	4	–	10	SP ← word	
		SP, AX	2	–	8	SP ← AX	
		AX, SP	2	–	8	AX ← SP	
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16	
		$addr16	2	6	–	PC ← PC + 2 + jdisp8	
		AX	2	8	–	PCH ← A, PCL ← X	
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (fCPU) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW. bit = 0	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	10	12	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0	
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0	
		saddr, $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0	
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1 (Enable Interrupt)	
	DI		2	–	6	IE ← 0 (Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

19.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except “r = A”

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand / First Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand / First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand / First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

CHAPTER 20 ELECTRICAL SPECIFICATIONS

Cautions **The µPD79F7025, 79F7026 microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.**

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
	EV_{SS}		–0.5 to + 0.3	V
REGC pin input voltage Note 2	V_{IREGC}		–0.5 to + 3.6 and –0.5 to V_{DD} + 0.3	V
Input voltage	V_{I1}	P00 to P05, P30 to P34, P60 to P67, P70 to P75, P121, P122, P125, X1, X2, $\overline{RESET}$	–0.3 to V_{DD} + 0.3 Note 1	V
Output voltage	V_{O1}	P00 to P05, P30 to P34, P60 to P67, P70 to P75, P121, P122	–0.3 to V_{DD} + 0.3 Note 1	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). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions			1Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00 to P05	POM0n = 0	−10	mA
				POM0n = 1	−250	mA
		Total of all pins			−300	mA
	I_{OH2}	Per pin	P30 to P34, P60 to P67, P70 to P75		−10	mA
		Total of all pins			−55	mA
	I_{OH3}	Per pin	P121, P122		−1	mA
		Total of all pins			−4	mA
Output current, low	I_{OL1}	Per pin	P00 to P05, P30 to P34		30	mA
		Total of all pins 120 mA	P00 to P05		60	mA
			P30 to P34		60	mA
	I_{OL2}	Per pin	P60 to P67, P70 to P75		30	mA
		Total of all pins			300	mA
	I_{OH3}	Per pin	P121		4	mA
			P122		9	mA
		Total of all pins	P121, P122		10	mA
Operating ambient temperature	T_A	Normal operation			−40 to +85	°C
		Flash memory programming			10 to 40	°C
Storage temperature	T_{stg}				−65 to +150	°C

Cautions 1. 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.

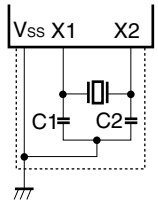
2. The value of the current that can be run per pin must satisfy the value of the current per pin and the total value of the currents of all pins.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. n = 0 to 5

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator, crystal resonator		X1 clock oscillation frequency (f_X)Note	4.5 V ≤ V_{DD} ≤ 5.5 V	1.0		10.0	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 High-speed Oscillator Characteristics

(T_A = –40 to +85°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Resonator	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
Internal high-speed oscillator	Oscillation frequency (f_{IH} = 4 MHz) Note	RSTS = 1	T_A = –40 to +85°C (± 3 %)	3.88	4	4.12	MHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Internal Low-speed Oscillator Characteristics

(T_A = –40 to +85°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Internal low-speed oscillator	Oscillation frequency (f_{IL})		216	240	264	kHz

DC Characteristics (1/4)

(T_A = 25°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P00 to P05	POM0n = 0			−3.0	mA
			POM0n = 1			−200.0 Note 3	mA
		Total of P00 to P05 Note 4				−206.0	mA
	I_{OH2}	Per pin for P30 to P34, P60 to P67, P70 to P75				−3.0	mA
		Total of P30 to P34, P60 to P67, P70 to P75				−33.0	mA
	I_{OH3}	Per pin for P121, P122				−100	µA
Output current, low Note 2	I_{OL1}	Per pin for P00 to P05, P30 to P34				8.5	mA
		Total of P00 to P05 Note 4				25.0	mA
		Total of P30 to P34 Note 4				40.0	mA
		Total of P00 to P05, P30 to P34 Note 4				65.0	mA
	I_{OL2}	Per pin for P60 to P67, P70 to P75				15.0	mA
		Total of P60 to P67, P70 to P75				210.0	mA
	I_{OL3}	Per pin for P121				400	µA
		Per pin for P122				8.5	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} to an output pin.

2. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

3. Specification under conditions where the duty factor is 25%

4. Specification under conditions where the duty factor is 70% (time for which current is output is 0.7 × t and time for which current is not output is 0.3 × t, where t is a specific time). The total output current of the pins at a duty factor of other than 70% can be calculated by the following expression.

- Where the duty factor of I_{OH} is n%: Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where the duty factor is 50%, 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.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. n = 0 to 5

DC Characteristics (2/4)

(T_A = –40 to +85°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00 to P05, P33, P34, P60 to P67, P70 to P75, P122 Note 1		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P121, P125 Note 2		0.7V_{DD}		V_{DD}	V
	V_{IH3}	P30 to P32, $\overline{RESET}$, EXCLK		0.8V_{DD}		V_{DD}	V
	V_{IH4}	X1, X2		V_{DD} – 0.1		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P05, P33, P34, P60 to P67, P70 to P75, P122 Note 1		0		0.3V_{DD}	V
	V_{IL2}	P121, P125 Note 2		0		0.3V_{DD}	V
	V_{IL3}	P30 to P32, $\overline{RESET}$, EXCLK		0		0.2V_{DD}	V
	V_{IL4}	X1, X2		0		0.1	V
Output voltage, high	V_{OH1}	P00 to P05 Note 3, P30 to P34, P60 to P67, P70 to P75	I_{OH1} = –3.0 mA, T_A = 25°C	V_{DD} – 0.7			V
	V_{OH2}	P00 to P05 Note 4	I_{OH2} = –200 mA, V_{DD} = 5 V, T_A = 25°C	V_{DD} – 1.5			V
	V_{OH3}	P121, P122	I_{OH3} = –100 µA, T_A = 25°C	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P00 to P05, P30 to P34, P122	I_{OL1} = 8.5 mA, T_A = 25°C			0.7	V
	V_{OL2}	P60 to P67, P70 to P75	I_{OL2} = 15.0 mA, T_A = 25°C			0.6	V
	V_{OL3}	P121	I_{OL3} = 400 µA, T_A = 25°C			0.4	V

Notes 1. V_{IH} and V_{IL} of P122/EXCLK differ between the input port mode and external clock mode.

2. V_{IH} and V_{IL} of P125/$\overline{RESET}$ differ between the input port mode and external $\overline{RESET}$ mode.

3. POMn = 0

4. POMn = 1

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. n = 0 to 5

DC Characteristics (3/4)

(T_A = –40 to +85°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P05, P30 to P34, P60 to P67, P70 to P75, P125/$\overline{\text{RESET}}$	$V_I = V_{DD}$				3	µA
	I_{LIH2}	P121, P122	$V_I = V_{DD}$	I/O port mode			3	µA
		X1, X2		OSC mode			20	µA
Input leakage current, low	I_{LIL1}	P00 to P05, P30 to P34, P60 to P67, P70 to P75, P125/$\overline{\text{RESET}}$	$V_I = V_{DD}$				–3	µA
	I_{LIL3}	P121, P122	$V_I = V_{DD}$	I/O port mode			–3	µA
		X1, X2		OSC mode			–20	µA
Pull-up resistor	R_{PLU1}	P00 to P05, P30 to P34, P60 to P67, P70 to P75	$V_I = V_{DD}$		10	20	100	kΩ
	R_{PLU2}	P125/$\overline{\text{RESET}}$			75	150	300	kΩ
Pull-down resistor	R_{PLU1}	P60 to P67, P70 to P75	$V_I = V_{DD}$		100	200		kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (4/4)

(T_A = –40 to +85°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	f_{XH} = 10 MHz, V_{DD} = 5.0 V	Square wave input		1.6	3.7	mA
				Resonator connection		2.3	5.1	mA
			f_{XH} = 5 MHz, V_{DD} = 5.0 V	Square wave input		0.9	2.1	mA
				Resonator connection		1.3	2.6	mA
			f_{IH} = 4 MHz, V_{DD} = 5.0 V			0.85	2.4	mA
	I_{DD2}	HALT mode	f_{XH} = 10 MHz, V_{DD} = 5.0 V	Square wave input		0.4	1.7	mA
				Resonator connection		1.0	3.2	mA
			f_{XH} = 5 MHz, V_{DD} = 5.0 V	Square wave input		0.2	0.85	mA
				Resonator connection		0.6	1.5	mA
			f_{IH} = 4 MHz, V_{DD} = 5.0 V			0.25	0.83	mA
	I_{DD3}	STOP mode	V_{DD} = 5.0 V, CREG + POC only			1.0	20	µA
Watchdog timer operating current[Note 2]	I_{WDT}	V_{DD} = 5.0 V	In 240 kHz internal low-speed oscillation clock operation			5	10	µA
LVI operating current[Note 3]	I_{LVI}					9	18	µA

Notes 1. Total current flowing into the internal power supply (V_{DD}), including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors, the pull-down resistors and the output current of the port are not included. Below the TYP. column are the values when V_{DD} = 3.0 V and the CPU is operating. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the watchdog timer, and LVI.

2. Current flowing only to the watchdog timer (including the operating current of the 240 kHz internal oscillator). The current value of the µPD79F7025, 79F7026 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates.

3. Current flowing only to the LVI circuit. The current value of the µPD79F7025, 79F7026 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

AC Characteristics

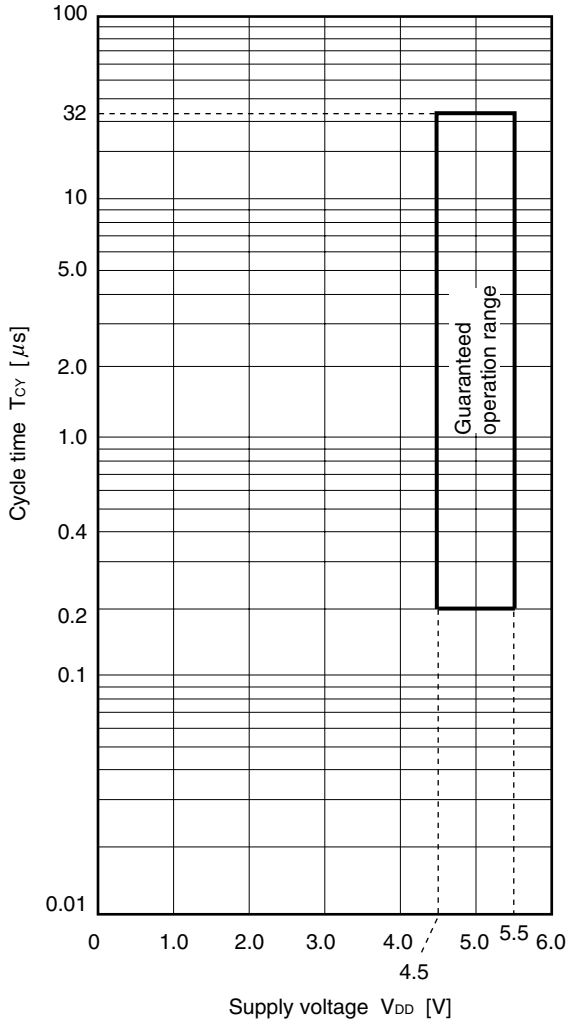
(1) Basic operation

(T_A = –40 to +85°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

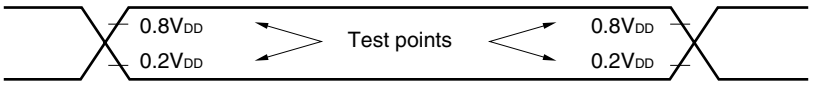
Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{XP}) operation	0.2		32	µs
Peripheral hardware clock frequency	f_{PRS}	f_{PRS} = f_{XP}			10	MHz
		f_{PRS} = f_{IH}	3.88		4.12	MHz
External main system clock frequency	f_{EXCLK}		1.0		10.0	MHz
External main system clock input high-level width, low-level width	t_{EXCLKH}, t_{EXCLKL}		(1/f_{EXCLK} ×1/2) –1			ns
TI000, TI010 input high-level width, low-level width	t_{TIH0}, t_{TIL0}		2/f_{sam}+0.2[Note]			µs
TI51 input frequency	f_{TI5}				10.0	MHz
TI51 input high-level width, low-level width	t_{TIH5}		50			ns
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}		1			µs
$\overline{\text{RESET}}$ low-level width	t_{RSL}		10			µs

Note Selection of f_{sam} = f_{PRS}, f_{PRS}/4, f_{PRS}/256 is possible using bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00). Note that when selecting the TI000 valid edge as the count clock, f_{sam} = f_{PRS}.

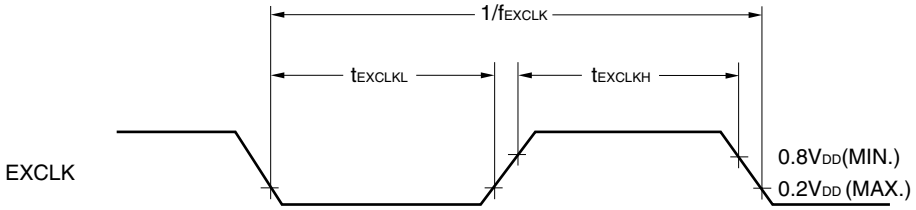
T_{CY} vs. V_{DD} (Main System Clock Operation)



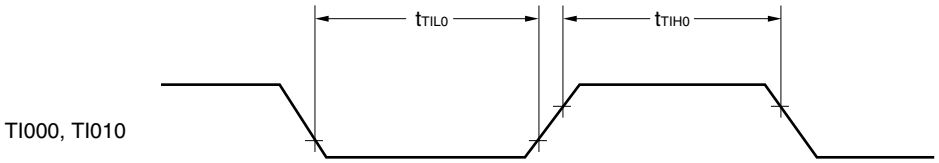
AC Timing Test Points

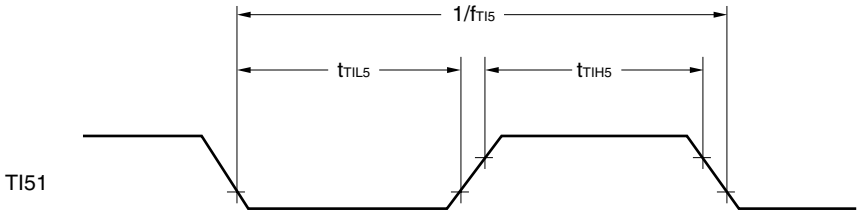


External Main System Clock Timing

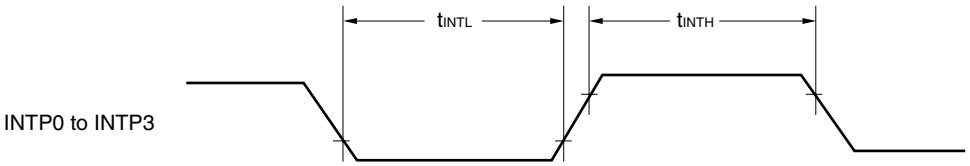


TI Timing

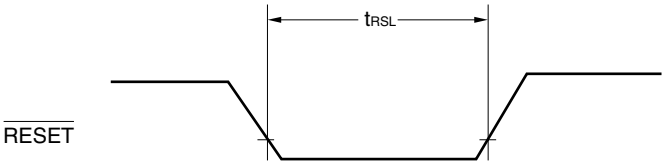




Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing



(2) On-chip debug

(T_A = –40 to +85°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate			f_{CLK}/32		f_{CLK}/8	bps
		In OCD mode (f_{CLK} = 4 MHz, V_{DD} ≥ 4.5 V, C_b = 50 pF)	125/250/500			kbps
		In Writer mode (f_{CLK} = 4 MHz, V_{DD} ≥ 4.5 V, C_b = 50 pF)	125/250/500			kbps
X2/output frequency	f_{OCDB}	V_{DD} ≥ 4.5 V			10	MHz

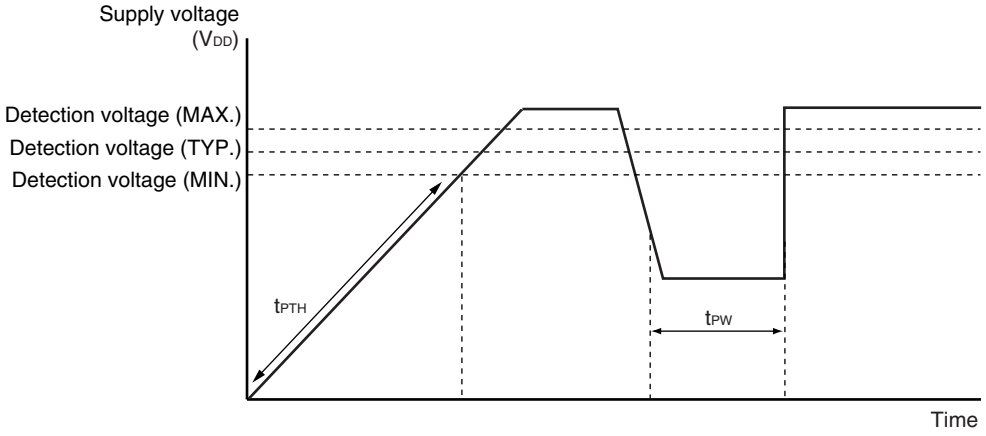
Analog Characteristics

(1) POC

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}		1.44	1.59	1.74	V
Power supply voltage rise inclination	t_{PTH}	Change inclination of V_{DD}: 0 V → V_{POC0}	0.5			V/ms
Minimum pulse width	t_{PW}	When the voltage drops	200			µs

POC Circuit Timing



(2) Supply Voltage Rise Time

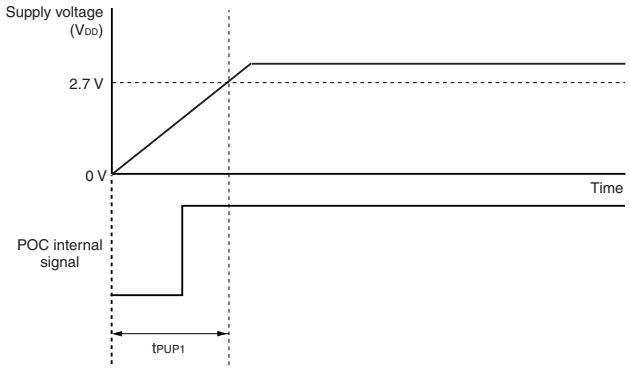
(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 4.5 V (V_{DD} (MIN.))[Note] (V_{DD}: 0 V → 4.5 V)	t_{PUP1}	LVI default start function stopped is set (LVISTART (Option Byte) = 0), when $\overline{\text{RESET}}$ input is not used			9.0	ms
Maximum time to rise to 4.5 V (V_{DD} (MIN.))[Note] (releasing $\overline{\text{RESET}}$ input → V_{DD}: 4.5 V)	t_{PUP2}	LVI default start function stopped is set (LVISTART (Option Byte) = 0), when $\overline{\text{RESET}}$ input is used			1.9	ms

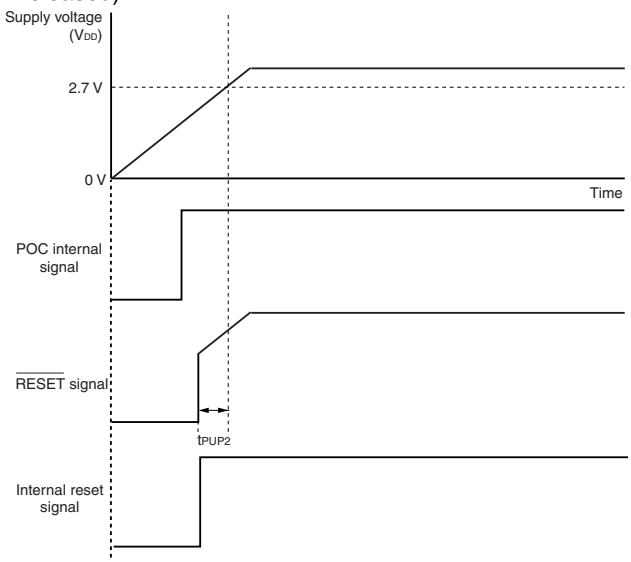
Note Make sure to raise the power supply in a shorter time than this.

Supply Voltage Rise Time Timing

- When $\overline{\text{RESET}}$ pin input is not used



- When $\overline{\text{RESET}}$ pin input is used (when external reset is released by the $\overline{\text{RESET}}$ pin, after POC has been released)



(3) LVI

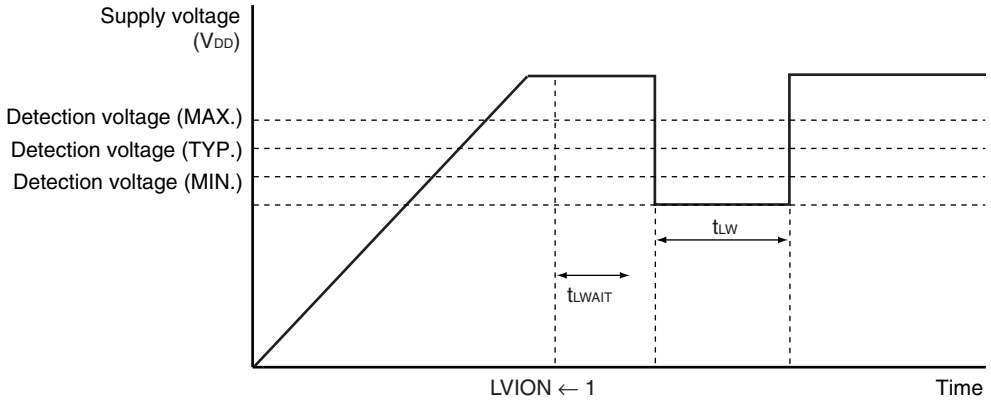
(T_A = –40 to +85°C, $V_{PDR} \le V_{DD} \le 5.5$ V, V_{SS} =0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}			4.24±0.1		V
		V_{LVI1}			4.09±0.1		V
		V_{LVI2}			3.93±0.1		V
	Supply voltage when power supply voltage is turned on	V_{DDLVI}	When LVI default start function enabled is set (LVISTART = 1)	2.5	2.7	2.9	V
Minimum pulse width		t_{LW}		200			μs
Operation stabilization wait time[Note]		t_{LWAIT}				10	μs

Note Time required from setting bit 7 (LVION) of the low-voltage detection register (LVIM) to 1 to operation stabilization

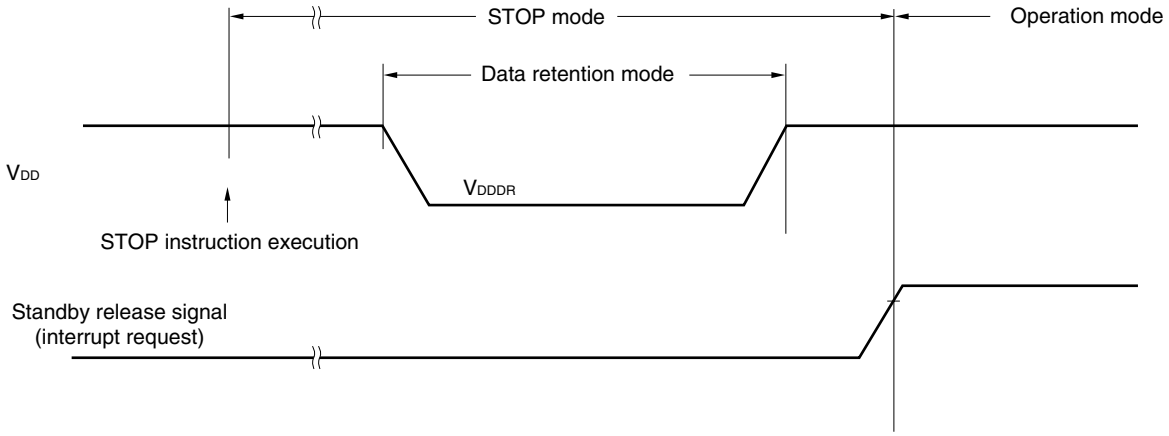
Remark $V_{LVI(n-1)} > V_{LVIn}$: n = 1, 2

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}		3.8		5.5	V



Flash Memory Programming Characteristics

(T_A = –40 to +85°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

• Basic characteristics

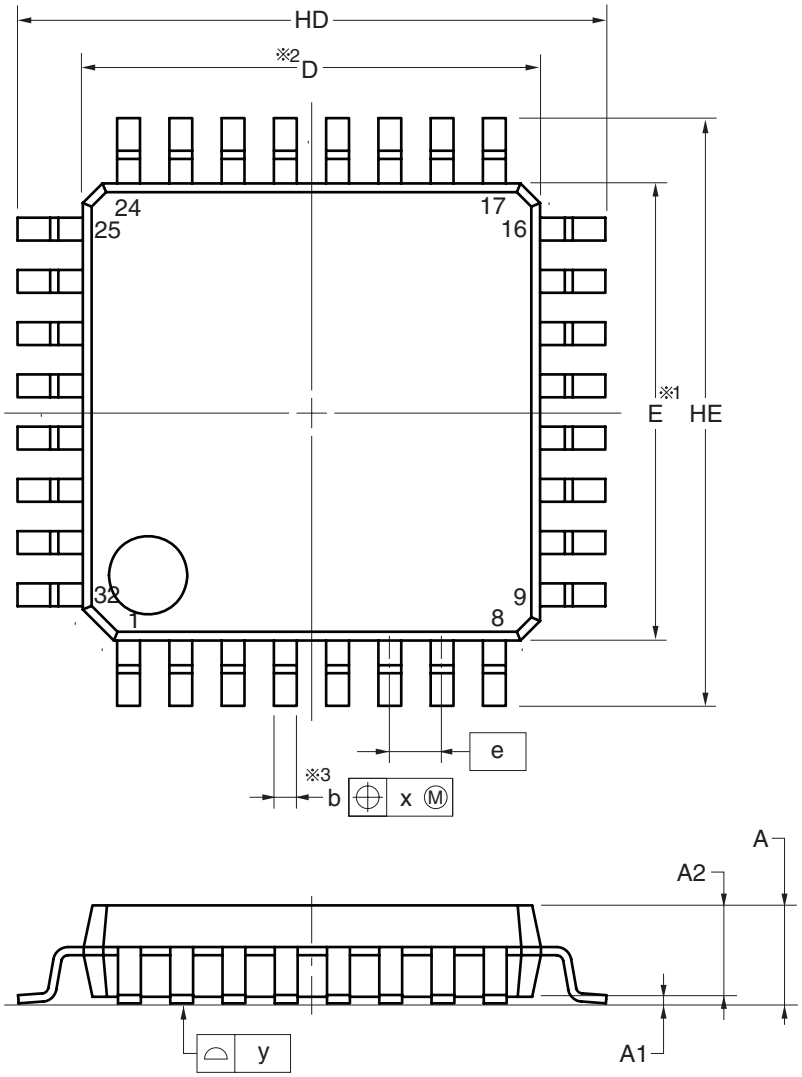
Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
V_{DD} supply current	I_{DD}					4.5	11.0	mA
Write time (in unit of 8 bit)	t_{wrwa}					10	100	µs
Number of rewrites per chip[Note]	C_{erwr}	1 erase + 1 write after erase = 1 rewrite	When a flash memory programmer is used	Retention: 15 years	10			Times
Operating temperature		When a flash memory programmer is used : 10 to 40 °C						

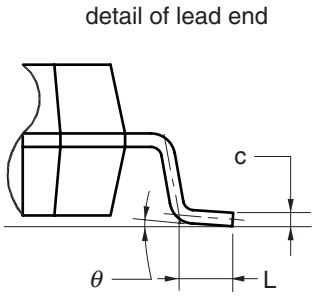
Note When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

CHAPTER 21 PACKAGE DRAWINGS

• μPD79F7025GA-GBT, 79F7026GA-GBT

32-PIN PLASTIC LQFP(7x7)





(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.10
E	7.00±0.10
HD	9.00±0.20
HE	9.00±0.20
A	1.70 MAX.
A1	0.10±0.10
A2	1.40
b	0.37±0.05
c	0.145±0.055
L	0.50±0.20
θ	0° to 8°
e	0.80
x	0.20
y	0.10

P32GA-80-GBT

NOTE

1.Dimensions "※1" and "※2" do not include mold flash.

2.Dimension "※3" does not include trim offset.

REVISION HISTORY	μPD79F7025, 79F7026 User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
1.00	Apr 25, 2012	—	First Edition issued.
1.01	Nov 14. 2012	276, 278	Information of the address. 0082H added to **Chapter 16, Option Byte**.
1.10	Nov 29. 2013	—	Modification of Caution on cover.

μPD79F7025, 79F7026 User's Manual: Hardware

Publication Date: Rev.1.10 Nov 29, 2013

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.3

μPD79F7025, 79F7026

Renesas Electronics Corporation

R01UH0328EJ0110