FUJITSU SEMICONDUCTOR
CONTROLLER MANUAL

CM25-10135-5E

F^{2}MC-8L

8-BIT MICROCONTROLLER

MB89530/530H/530A Series HARDWARE MANUAL

FUJITSU

F^2MC-8L

8-BIT MICROCONTROLLER

MB89530/530H/530A Series HARDWARE MANUAL

FUJITSU LIMITED

PREFACE

■ Objectives and Intended Readers

Thank you for purchasing Fujitsu semiconductor products.

The MB89530/530H/530A series of microcontrollers was developed as a general-purpose product in the F^{2}MC ®*-8L series that comprises the proprietary 8-bit one-chip microcontroller that supports application specific ICs (ASICs). This product is designed for a broad range of uses, from consumer products to industrial equipment, such as portable devices.

This manual describes the functions and operations of the MB89530/530H/530A series of microcontrollers for engineers who develop products using this series. Be sure to read this manual thoroughly.

For details on the instructions, see the F^{2}MC Programming Manual.

■ Trademark

F^{2}MC, an abbreviation for FUJITSU Flexible Microcontroller, is a registered trademark of Fujitsu Limited.

■ Organization of this manual

This manual consists of the following 23 chapters and appendix:

CHAPTER 1 "OVERVIEW"

This chapter explains the features and basic specifications of the MB89530/530H/530A series of microcontrollers.

CHAPTER 2 "HANDLING DEVICE"

This chapter describes the precautions to be taken when using the MB89530/530H/530A series of microcontrollers.

CHAPTER 3 "CPU"

This chapter describes the functions and operations of the CPU.

CHAPTER 4 "I/O PORTS"

This chapter describes the functions and operations of the I/O port.

CHAPTER 5 "TIMEBASE TIMER"

This chapter describes the functions and operations of the timebase timer.

CHAPTER 6 "WATCHDOG TIMER"

This chapter describes the functions and operations of the watchdog timer.

CHAPTER 7 "WATCH PRESCALER"

This chapter describes the functions and operations of the watch prescaler.

CHAPTER 8 "2-CHANNEL 8-BIT PWM TIMERS"

This chapter describes the functions and operations of the 2-channel 8-bit PWM timer.

CHAPTER 9 "PULSE-WIDTH COUNT TIMER (PWC)"

This chapter describes the functions and operations of the pulse width count timer (PWC).

CHAPTER 10 "6-BIT PPG TIMER"

This chapter describes the functions and operations of 6-bit PPG timer.

CHAPTER 11 "12-BIT PPG TIMER"

This chapter describes the functions and operations of the 12-bit PPG timer.

CHAPTER 12 "16-BIT TIMER/COUNTER"

This chapter describes the functions and operations of the 16-bit timer/counter.

CHAPTER 13 "EXTERNAL INTERRUPT CIRCUIT 1 (EDGE)"

This chapter describes the functions and operations of external interrupt circuit 1 (edge).

CHAPTER 14 "EXTERNAL INTERRUPT CIRCUIT 2 (LEVEL)"

This chapter describes the functions and operations of external interrupt circuit 2 (level).

CHAPTER 15 "A/D CONVERTER"

This chapter describes the functions and operations of the A/D converter.

CHAPTER 16 "UART/SIO"

This chapter describes the functions and operations of the UART/SIO.

CHAPTER 17 "HIGH-SPEED UART"

This chapter describes the functions and operations of the high-speed UART.

CHAPTER 18 "8-BIT SERIAL I/O"

This chapter describes the functions and operations of the 8-bit serial I/O.

CHAPTER 19 "I^{2}C INTERFACE"

This chapter describes the functions and operations of the I^{2}C bus Interface.

CHAPTER 20 "WILD REGISTER FUNCTION"

This chapter describes the functions and operations of the wild register function.

CHAPTER 21 "CLOCK OUTPUT"

This chapter describes the functions and operations of the clock output function.

CHAPTER 22 "FLASH MEMORY"

This chapter describes the functions and operations of the flash memory.

CHAPTER 23 "MB89F538/F538L SERIAL PROGRAMMING"

This chapter describes an example of serial writing connection.

APPENDIX

This appendix lists the I/O map and instructions.

- The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering.
- The information and circuit diagrams in this document are presented as examples of semiconductor device applications, and are not intended to be incorporated in devices for actual use. Also, FUJITSU is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of this information or circuit diagrams.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).

  Please note that Fujitsu will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

HOW TO USE THIS MANUAL

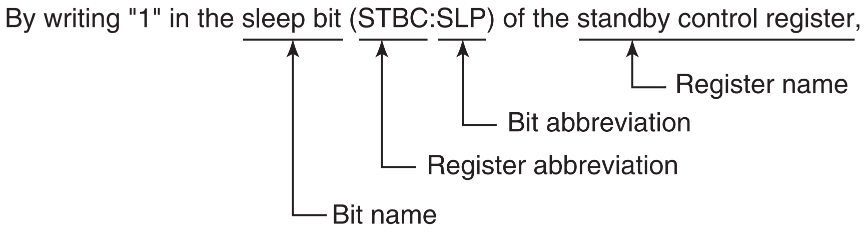
■ Page Configuration of This Manual

Each section of this manual has a summary below the title. This is to enable the reader to acquire an understanding of the outline of the product.

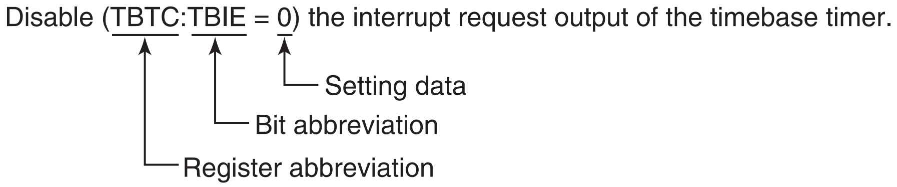
The titles of the sections are also described in the subsections. This is to let the reader know what section of the manual he or she is consulting.

■ Example of Notations for the Register and Bit Names

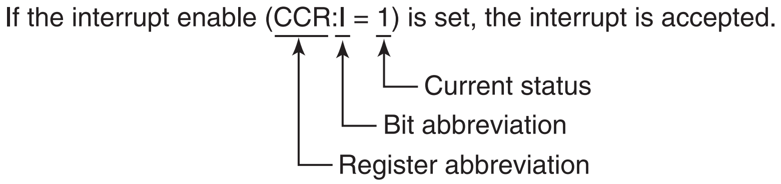
❍ Example of the notations for the register and bit names



By writing "1" in the sleep bit (STBC:SLP) of the standby control register,



Disable (TBTC:TBIE = 0) the interrupt request output of the timebase timer.



If the interrupt enable (CCR:I = 1) is set, the interrupt is accepted.

❍ Example of the notation for the convertible pin

P34/PT02 pin

A convertible pin is available which can be used by switching its functions by program settings. The convertible pin is represented by separating the name of each function by "/".

CONTENTS

CHAPTER 1 OVERVIEW

This chapter describes the features and basic specifications of the MB89530/530H/530A series of microcontrollers.

1.1 Features of the MB89530/530H/530A Series

The MB89530/530H/530A series is a one-chip microcontroller that uses the F^{2}MC®-8L core to support the low voltage and high-speed operation. The internal peripheral functions include the timers, serial interfaces, A/D converter, external interrupts, and so on. This series is a general-purpose one-chip microcontroller that is suitable for a broad range of uses, from consumer products to industrial equipment as well as for portable devices.

■ Features of the MB89530/530H/530A series

❍ **Full package development**

- Two types of QFP packages (1-mm pitch, 0.65-mm pitch)
- LQFP package (0.5-mm pitch)
- SH-DIP package

❍ **High-speed operation with low voltage**

- Minimum instruction execution time 0.32 µs (for oscillation 12.5 MHz)

❍ **F^{2}MC®-8L CPU core**

Optimal instruction system for the controller

- Multiplication and division instructions
- 16-bit arithmetic operations
- Branch instructions with bit test
- Bit manipulation instructions

❍ **Five-system timer**

- 8-bit PWM timer: 2-channel timer (can be used as the interval timer or PWM timer)
- Pulse-width count timer (supports the usages such as continuous measurement and remote control reception)
- 16-bit timer counter
- 21-bit timebase timer
- Watch prescaler (17 bits)

❍ **UART**

- Clock synchronization and asynchronization can be switched.

❍ **Two serial interfaces (serial I/O)**

- Transfer direction can be selected (specification of MSB first or LSB first), enabling communication with a variety of devices.

❍ **10-bit A/D converter (8 channels)**

- The start of the A/D converter is supported with external clock input and timebase timer output. (The A/D converter cannot be started with external clock input for MB89F538/F538L.)

❍ **Two programmable pulse generators (PPGs)**

- 6-bit PPG that enables a pulse width and cycle to be selected with a program
- 12-bit PPG that enables a pulse width and cycle to be selected with a program (2 channels)

❍ **I^{2}C interface circuit**

❍ **External interrupt 1 (single-clock system product: 4 channels, dual-clock system product: 3 channels)**

- Four or three inputs are independent, enabling releases from standby mode. (With the edge detection function)

❍ **External interrupt 2 (product other than MB89F538/F538L: 8 channels, MB89F538/F538L: 7 channels)**

- Eight or seven inputs are independent, enabling releases from standby mode. (With the level edge detection function)

❍ **Standby mode (low-power consumption mode)**

- Stop mode (current consumption hardly takes place because oscillation stops)
- Sleep mode (current consumption becomes about 1/3 of the usual amount because the CPU stops)
- Subclock mode
- Watch mode

❍ **Watchdog timer reset**

❍ **I/O ports**

- Maximum number of ports

  Single-clock system product

  Other than MB89F538/F538L: 53 ports

  MB89F538/F538L: 52 ports

  Dual-clock system product

  Other than MB89F538/F538L: 51 ports

  MB89F538/F538L: 50 ports

- General-purpose I/O port (CMOS)

  Other than MB89F538/F538L: 38 ports

  MB89F538/F538L: 37 ports
- General-purpose I/O port (N-channel open drain): 2 ports
- General-purpose output port (N-channel open drain): 8 ports
- General-purpose input port (CMOS)

  Single-clock system product: 5 ports

  Dual-clock system product: 3 ports

1.2 Available Models of the MB89530/530H/530A Series

Four models of the MB89530/530H/530A series are supported. Table 1.2-1 "Available models of the MB89530/530H/530A series" lists the available models and Table 1.2-2 "CPU and peripheral functions of the MB89530/530H/530A series" lists the CPU and peripheral functions.

■ Available models of the MB89530/530H/530A series

Table 1.2-1 Available models of the MB89530/530H/530A series

Item	Model					
	MB89535A	**MB89537/537C** **MB89537H/537HC** **MB89537A/537AC**	**MB89538/538C** **MB89538H/538HC** **MB89538A/538AC**	**MB89F538/F538L**	**MB89P538**	**MB89PV530**
Classification	Mass production (mask ROM product)			Flash Memory	PROM	EVA
ROM capacity	16 KB x 8 bits (Internal ROM)	32 KB x 8 bits (Internal ROM)	48 KB x 8 bits (Internal ROM)	48 KB x 8 bits (Internal flash memory)	48 KB x 8 bits (Internal PROM)	48 KB x 8 bits (External ROM) (*2)
RAM capacity	512 x 8 bits	1 KB x 8 bits	2 KB x 8 bits			
Operating voltage	2.2V to 3.6V(*1) (MB89537/538/537C/538C) 3.5V to 5.5V(*1) (MB89537H/538H/537HC/538HC) 2.2V to 5.5V(*1) (MB89535A/537A/538A/537AC/538AC)			MB89F538: 3.5V to 5.5V(*1) MB89F538L: 2.4V to 3.6V(*1)(*3) 2.7V to 3.6V(*1)(*4)	2.7V to 5.5V	

*1 Depends on the operating frequency.
*2 MBM27C512 is used as the external ROM.
*3 Flash memory read assurance voltage.
*4 Flash memory read/write assurance voltage.

Note:

MB89537/538/537H/538H/535A/537A/538A contain no I^{2}C function.
To use the I^{2}C function, use MB89PV530/MB89P538/MB89F538/MB89F538L/MB89537C/538C/537HC/538HC/537AC/538AC.

Table 1.2-2 CPU and peripheral functions of the MB89530/530H/530A series

Item		Specifications
CPU function		Number of basic instructions : 136 Instruction bit length : 8 bits Instruction length : 1 to 3 bytes Data bit length : 1, 8, and 16 bits Minimum instruction execution time : 0.32 µs/12.5 MHz Minimum interrupt processing time : 2.88 µs/12.5 MHz
Peripheral function	Port	Input port • Single-clock system product : 5 ports (Four ports also serve as external interrupts.) • Dual-clock system product : 3 ports (Three ports also serve as external interrupts.) Output-dedicated port (N-channel) : 8 ports (Eight ports are also used as ADC input.) I/O port (N-channel open drain) : 2 ports (Two ports also serve as S02/SDA and SI2/SCL.) I/O port (CMOS) • Other than MB89F538/F538L : 38 pins • MB89F538/F538L : 37 pins (Twenty-one ports do not serve as other functions.) Total • Single-clock system product Other than MB89F538/F538L: 53 ports MB89F538/F538L: 52 ports • Dual-clock system product Other than MB89F538/F538L: 51 ports MB89F538/F538L: 50 ports
	Timebase timer	21 bits Interrupt cycle for main clock original oscillation 12.5 MHz (approx. 0.655 ms, 2.621 ms, 20.97 ms, and 335.5 ms)
	Watchdog timer	Reset cycle for main clock original oscillation 12.5 MHz (approx. 167.8 to 335.6 ms) Reset cycle for subclock original oscillation 32.768 KHz (approx. 500 to 1,000 ms)
	PWM timer	8-bit interval timer operation (Rectangular output supported, operating clock cycle: $1t_{inst}$, $8t_{inst}$, $16t_{inst}$, and $64t_{inst}$) 8-bit resolution pulse width measurement (conversion cycle: $2^8 t_{inst}$ to 2^8 x $64t_{inst}$) 2 channels (also usable as the interval timer), usable as the channel 1 output and channel 2 count clock.
Watch prescaler		Interval time for 17-bit subclock original oscillation 32.768 KHz (Approx. 31.25 ms, 0.25 s, 0.50 s, 1.00 s, 2.00 s, and 4.00 s)

Table 1.2-2 CPU and peripheral functions of the MB89530/530H/530A series (Continued)

Item		Specifications
Peripheral functions	Pulse width count timer	8-bit one-shot timer operation (Underflow output supported, operating clock cycle: $1t_{inst}$, $4t_{inst}$, $32t_{inst}$, and external) 8-bit reload timer operation (rectangular output supported, operating clock cycle: $1t_{inst}$, $4t_{inst}$, $32t_{inst}$, and external) 8-bit pulse width measurement operation (the following measurements are possible: continuous measurement, H width measurement, L width measurement, measurement from ↑ to ↑, measurement from ↓ to ↓, and both H width measurement and measurement from ↑ to ↑)
	16-bit timer/ counter	16-bit timer operation (operating clock cycle: $1t_{inst}$, and external) 16-bit event counter operation (rising edge, falling edge, or both selectable) 16 bits x 1 channel
	Serial I/O	8-bit length LSB or MSB precedence selectable Transfer clock ($2t_{inst}$, $8t_{inst}$, $32t_{inst}$, and external)
	UART/SIO	CLK synchronization/CLK asynchronization data transferable (bits 8 and 9 with parity bits, bits 7 and 8 without parity bits) 14 baud rates can be selected with the internal baud rate generator.
	UART	CLK synchronization/CLK asynchronization data transferable (bits 4, 6, 7, and 8 with parity bits, bits 5, 7, 8, and 9 without parity bits) 14 baud rates can be selected with the internal baud rate generator. Baud rates can also be set with external clock input and 2-channel 8-bit PWM timer output.
	External interrupt 1	Single-clock system product: independent 4 channels, dual-clock system product: independent 3 channels Rising edge, falling edge, or both can be selected. Usable for release from standby mode (edge detection can also be used in stop mode).
	External interrupt 2	Other than MB89F538/F538L: independent 8 channel L level detection, MB89F538/F538L: independent 7 channel L level detection Usable for release from standby mode.
	6-bit PPG, 12-bit PPG	Rectangular waves whose cycle is programmable can be generated. 6 bits x 1 channel and 12 bits x 2 channels
	I^2C bus interface	1 channel, conforms to the system management bus revised edition 1.0 of the Intel Corporation and to the I^2C specifications of Philips Electronics. 2-wire system communication (contained only in the following: MB89PV530/P538/F538/F538L/537C/538C/537HC/538HC/537AC/538AC)
	A/D converter	10-bit resolution x 8 channels A/D conversion function (conversion time: $60t_{inst}$) Repeated start is supported with external or internal clocks (MB89F538/MB89F538L is excluded from repeated start with external clocks.) Reference voltage input available (AVR)
Standby mode (low-power mode)		Sleep mode, stop mode, subclock mode, and watch mode
Process		CMOS

t_{inst}: Instruction cycle (execution time). 1/4, 1/8, 1/16, or 1/64 of the main clock or 1/2 of the subclock can be selected (see Section 3.6 "Clock").

1.3 Differences among Models and the Precautions to Be Taken When Selecting Models

This section describes the differences among five models of the MB89530/530H/530A series and the precautions to be taken when selecting models.

■ Differences among models and precautions to be taken when selecting models

Table 1.3-1 Packages of the supported models

Package	Model				
	MB89535A	MB89537/537C MB89537H/537HC MB89537A/537AC	MB89538/538C MB89538H/538HC MB89538A/538AC	MB89P538 MB89F538 MB89F538L	MB89PV530
DIP-64P-M01	o	o	o	o	x
FPT-64P-M03	o	o	o	x	x
FPT-64P-M06	o	o	o	o	x
FPT-64P-M09	o	o	o	o	x
MDP-64C-P02	x	x	x	x	o
MQP-64C-P01	x	x	x	x	o

[o]: Usable
[x]: Unusable
The conversion socket (manufactured by Sunhayato Corporation) can be used to convert a pin pitch.
Where to make contact: Sunhayato Corporation
Tel. 03-3986-0403

❍ **Memory space**

To use a piggyback model for evaluation, carefully confirm the differences between this model and the model to be actually used. Pay special attention to the following points. (See Section 3.1 "Memory Space.")

- The program ROM area begins from 4000H on MB89P538, MB89F538, MB89F538L, and MB89PV530.
- The stack area is set in the upper limit of the RAM

❍ **Current consumption**

- For MB89PV530, the current consumed by EPROM connected to the top socket is added.
- During low-speed operation, the current consumption of the model mounted on the one-time PROM or the EPROM is greater than that of the model mounted on the mask ROM. However, the current consumption in sleep or stop mode is equal for both cases.

For details, see the electrical characteristics of the data sheet.

❍ Mask option

The functions that can be specified with options and the method of specifying the options depend on the model. Before using the options, first refer to Appendix C "Mask Options" for details.

❍ Wild register function

Table 1.3-2 "Spaces in which the wild register function can be used" lists the spaces in which the wild register function can be used.

Table 1.3-2 Spaces in which the wild register function can be used

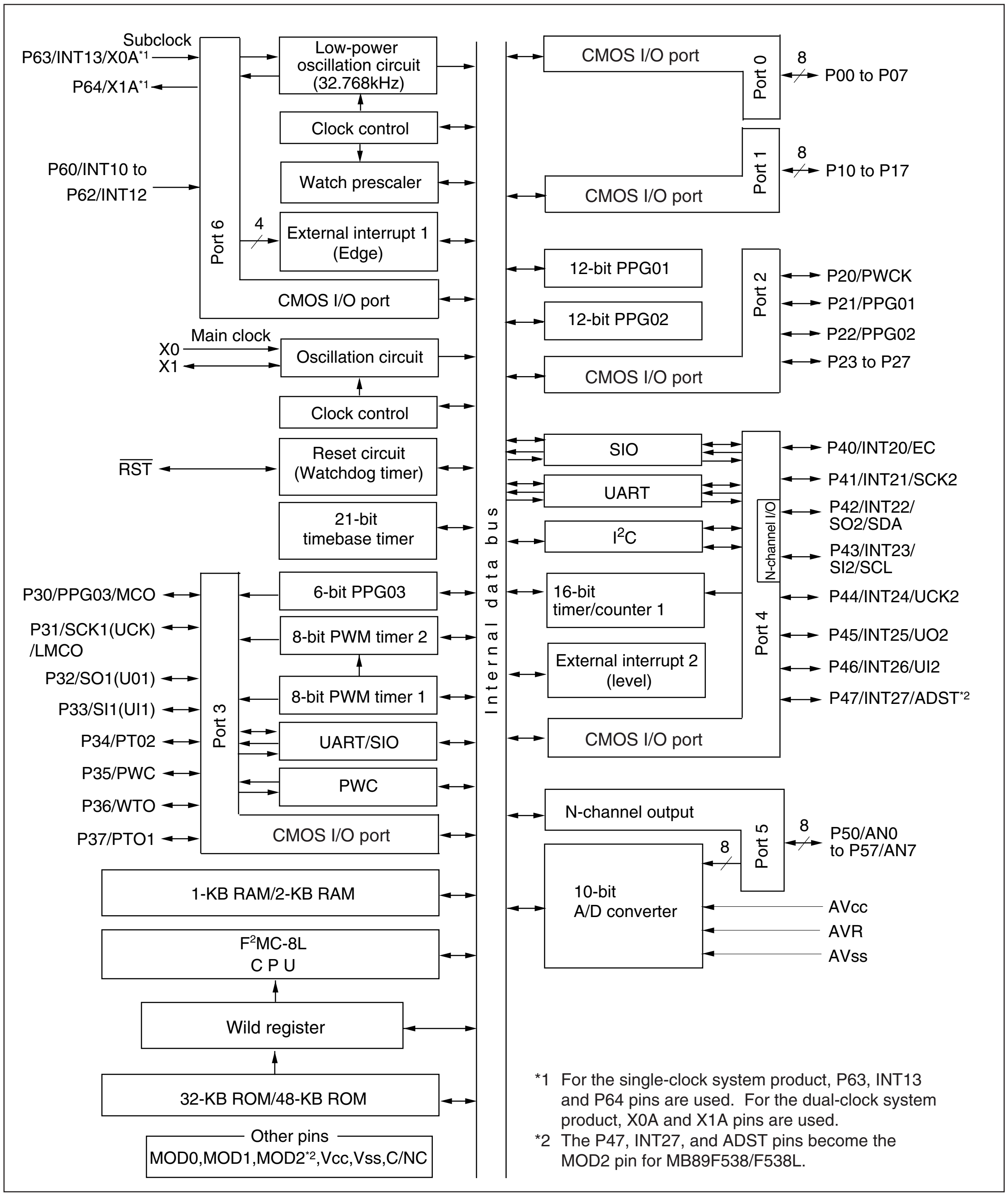
Model name	Address space
MB89PV530	4000_H to $FFFF_H$
MB89P538/F538/F538L	4000_H to $FFFF_H$
MB89537/537C/537H/537HC/537A/537AC	8000_H to $FFFF_H$
MB89538/538C/538H/538HC/538A/538AC	4000_H to $FFFF_H$
MB89535A	$C000_H$ to $FFFF_H$

1.4 Block Diagram of the MB89530/530H/530A Series

Figure 1.4-1 "Entire block diagram of the MB89530/530H/530A series" shows the block diagram of the MB89530/530H/530A series.

■ Entire block diagram of the MB89530/530H/530A series

Figure 1.4-1 Entire block diagram of the MB89530/530H/530A series



*1 For the single-clock system product, P63, INT13 and P64 pins are used. For the dual-clock system product, X0A and X1A pins are used.

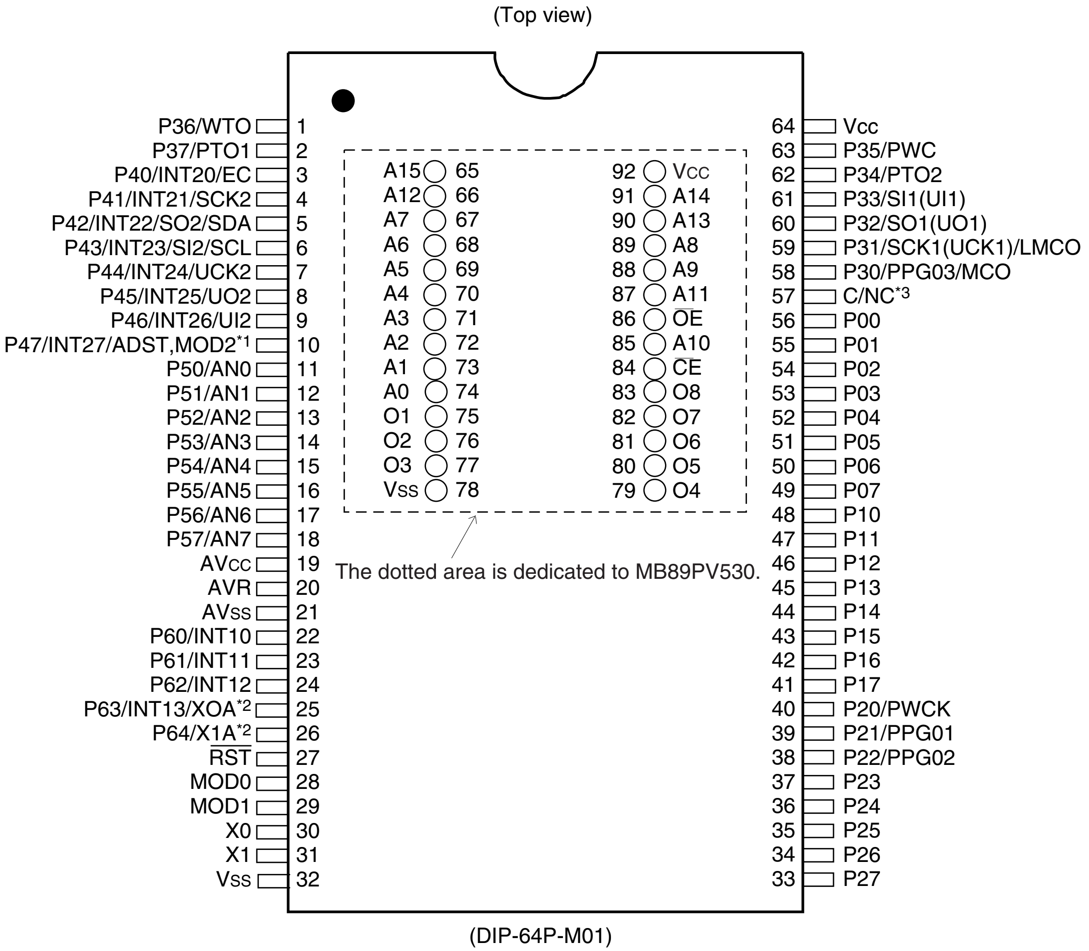
*2 The P47, INT27, and ADST pins become the MOD2 pin for MB89F538/F538L.

1.5 Pin Layout

Figure 1.5-1 "Pin layout for DIP-64P-M01 and MDP-64C-P02", Figure 1.5-2 "Pin layout for FPT-64P-M03 and FPT-64P-M09", and Figure 1.5-3 "Pin layout diagram for FPT-64P-M06 and MQP-64C-P01" show the pin layouts of the MB89530/530H/530A series.

■ Pin layout for DIP-64P-M01 and MDP-64C-P02

Figure 1.5-1 Pin layout for DIP-64P-M01 and MDP-64C-P02



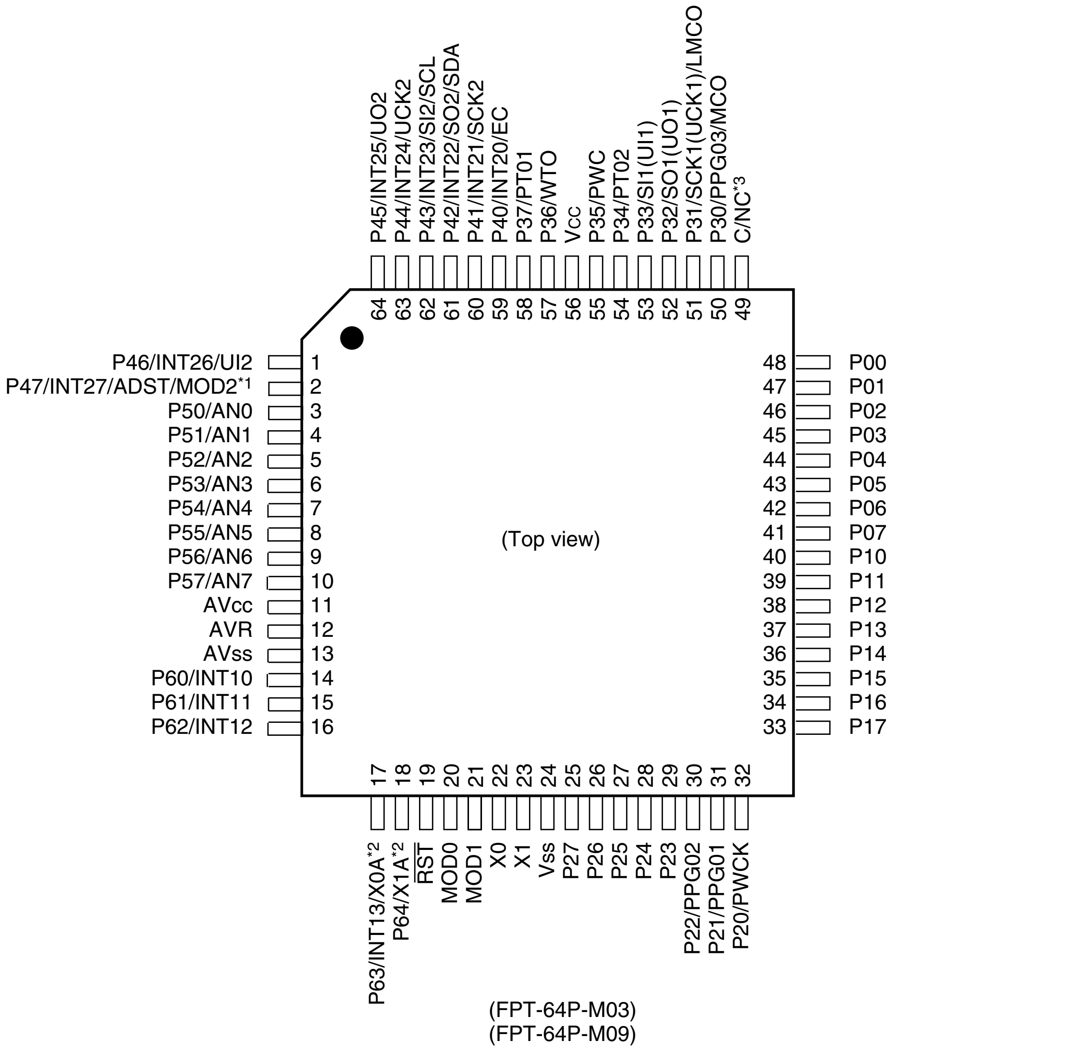
*1 Pin 10 is the MOD2 pin for MB89F538/F538L and pin 10 is the P47, INT27, and ADST pins for a product other than MB89F538/538L.

*2 Pins 25 and 26 are P63, INT13, and P64 pins for the single-clock system product. Pins 25 and 26 are X0A and X1A pins for the dual-clock system product.

*3 The pin function of pin 57 depends on the model. For details, see Chapter 2, "Handling Device."

■ Pin layout for FPT-64P-M03 and FPT-64P-M09

Figure 1.5-2 Pin layout for FPT-64P-M03 and FPT-64P-M09



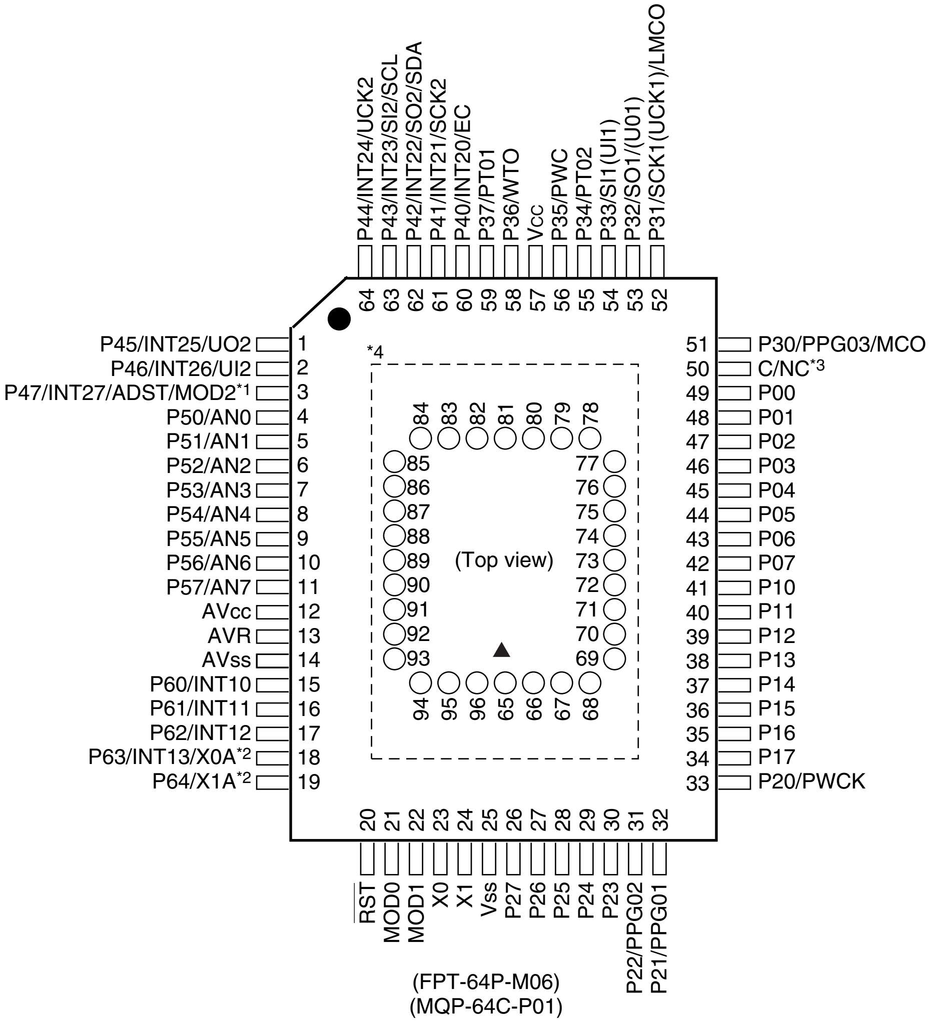
*1 Pin 2 is the MOD2 pin for MB89F538/F538L. Pin 2 is the P47, INT27, and ADST pins for a product other than MB89F538/538L.

*2 Pins 17 and 18 are P63, INT13, and P64 pins for the single-clock system product. Pins 17 and 18 are X0A and X1A pins for the dual-clock system product.

*3 The pin function of pin 49 depends on the model. For details, see Chapter 2, "Handling Device."

■ Pin layout for FPT-64P-M06 and MQP-64C-P01

Figure 1.5-3 Pin layout diagram for FPT-64P-M06 and MQP-64C-P01



*1 Pin 3 is the MOD2 pin for MB89F538/F538L. Pin 3 is the P47, INT27, and ADST pins for a product other than MB89F538/538L.

*2 Pins 18 and 19 are P63, INT13, and P64 pins for the single-clock system product. Pins 18 and 19 are the X0A and X1A pins for the dual-clock system product.

*3 The pin function of pin 50 depends on the model. For details, see Chapter 2, "Handling Device."

*4 Pin layout (dedicated to MB89PV530) in the upper part of the package

Pin number	Pin symbol	Pin number	Pin symbol	Pin number	Pin symbol	Pin number	Pin symbol
65	N.C.	73	A2	81	N.C.	89	OE
66	A15	74	A1	82	04	90	N.C.
67	A12	75	A0	83	05	91	A11
68	A7	76	N.C.	84	06	92	A9
69	A6	77	01	85	07	93	A8
70	A5	78	02	86	08	94	A13
71	A4	79	03	87	CE	95	A14
72	A3	80	V_{SS}	88	A10	96	V_{CC}

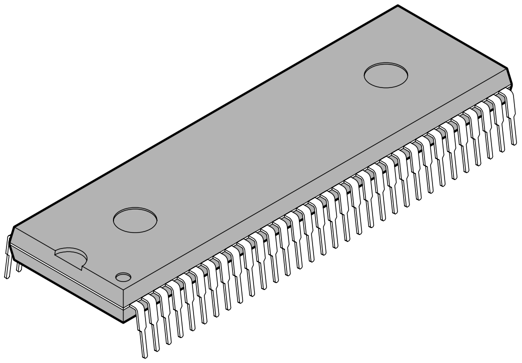
N.C.: This pin is unusable because it is used for internal connection.

1.6 External Dimension Diagram

Six packages are provided for the MB89530/530H/530A series. Figure 1.6-1 "External dimensions for DIP-64P-M01" to Figure 1.6-6 "External dimensions for MQP-64C-P01" show their external dimensions.

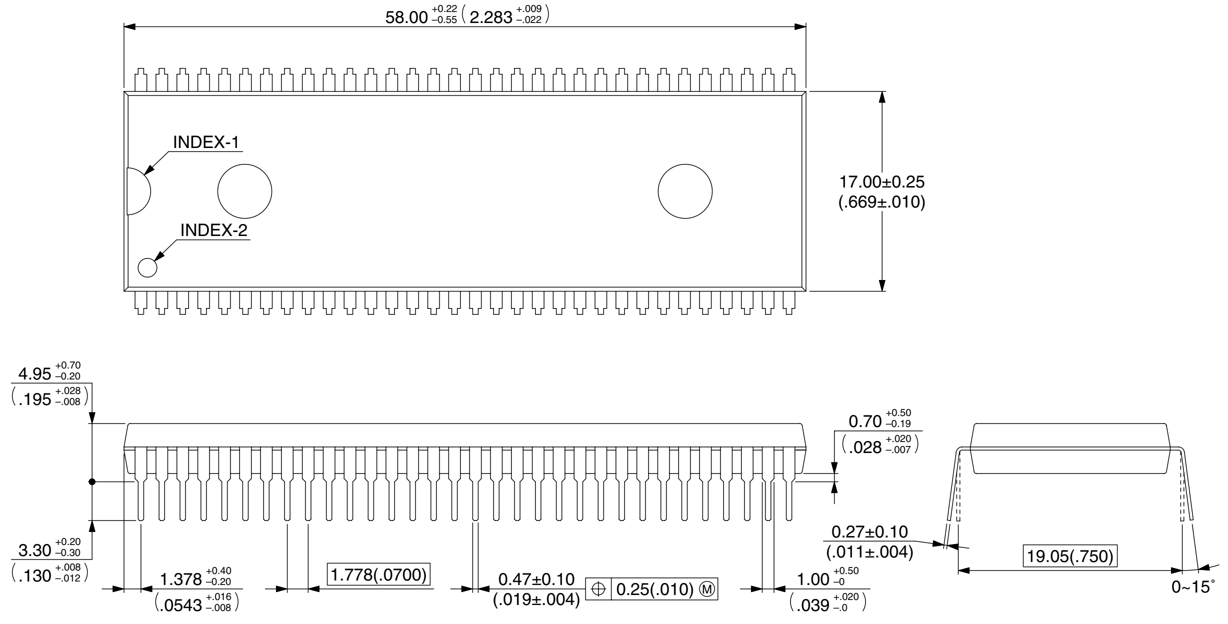
■ External dimensions for DIP-64P-M01

Figure 1.6-1 External dimensions for DIP-64P-M01

Plastic SH-DIP 64 pins  (DIP-64P-M01)	Lead pitch	1.778mm
	Package width × package length	17 × 58 mm
	Packaging method	Plastic mold
	Mounting height	5.65 mm MAX

Plastic SH-DIP 64 pins
(DIP-64P-M01)

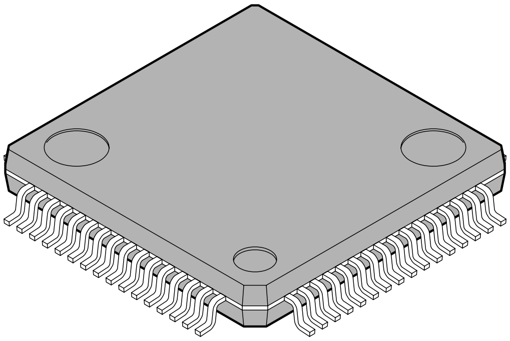
Note: The pin width and pin thickness include the plating thickness.



Unit: mm (inches)

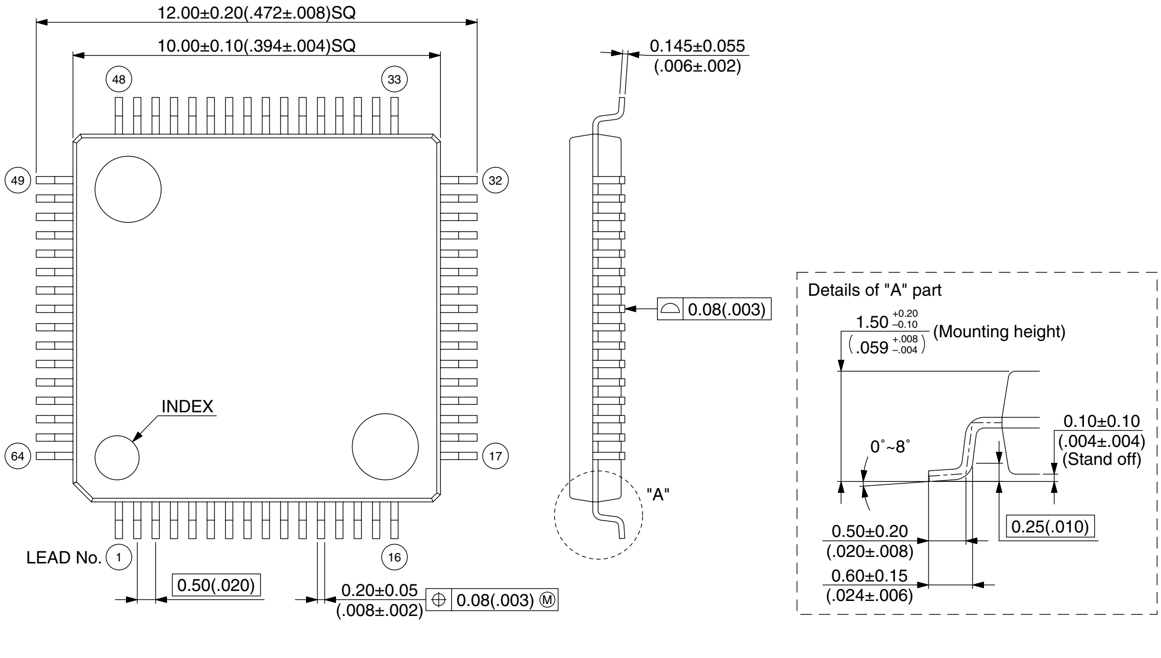
■ External dimensions for FPT-64P-M03

Figure 1.6-2 External dimensions for FPT-64P-M03

Plastic LQFP 64 pins  (FPT-64P-M03)	Lead pitch	0.50 mm
	Package width × package length	10.0 × 10.0 mm
	Lead shape	Gullwing
	Packaging method	Plastic mold
	Mounting height	1.70 mm MAX
	Weight	0.32g

Plastic LQFP 64 pins
(FPT-64P-M03)

Note: The pin width and pin thickness include the plating thickness.

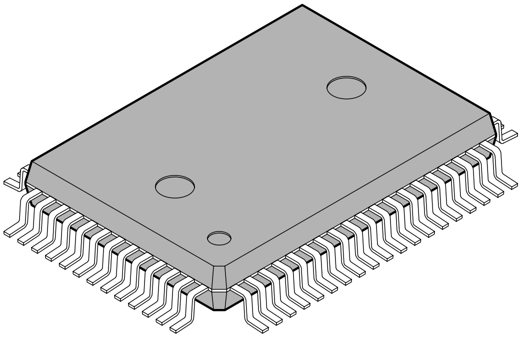


© 2000 FUJITSU LIMITED F64009S-c-4-7

Unit: mm (inches)

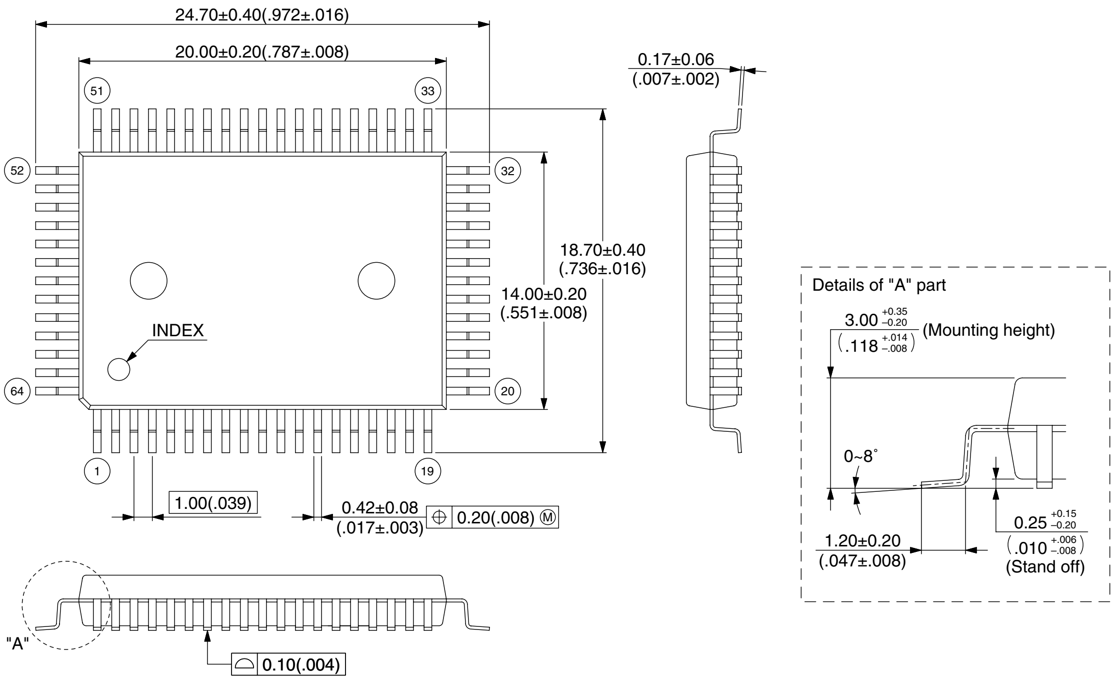
■ External dimensions for FPT-64P-M06

Figure 1.6-3 External dimensions for FPT-64P-M06

Plastic QFP 64 pins  (FPT-64P-M06)	Lead pitch	1.00 mm
	Package width × package length	14 × 20 mm
	Lead shape	Gullwing
	Packaging method	Plastic mold
	Mounting height	3.35 mm MAX

Plastic QFP 64 pins
(FPT-64P-M06)

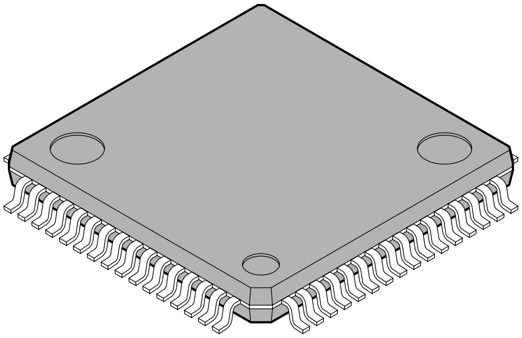
Note: The pin width and pin thickness include the plating thickness.



Unit: mm (inches)

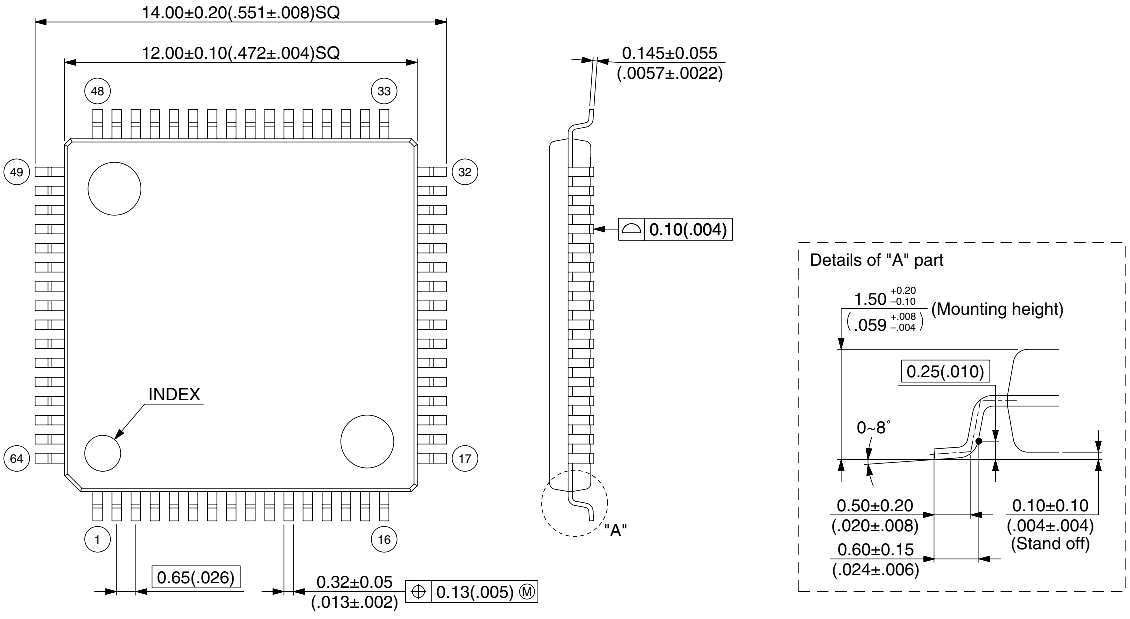
■ External dimensions for FPT-64P-M09

Figure 1.6-4 External dimensions for FPT-64P-M09

Plastic LQFP 64 pins  (FPT-64P-M09)	Lead pitch	0.65 mm
	Package width × package length	12 × 12 mm
	Lead shape	Gullwing
	Packaging method	Plastic mold
	Mounting height	1.70 mm MAX

Plastic LQFP 64 pins
(FPT-64P-M09)

Note: The pin width and pin thickness include the plating thickness.

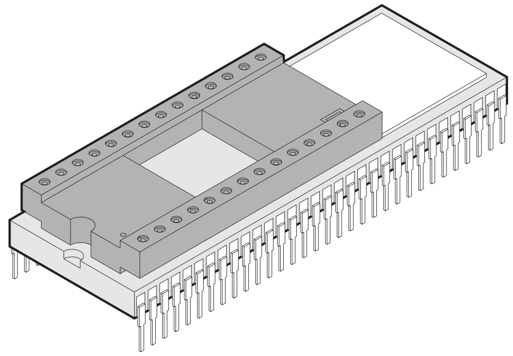


Ⓒ 2001 FUJITSU LIMITED F64018S-c-2-4

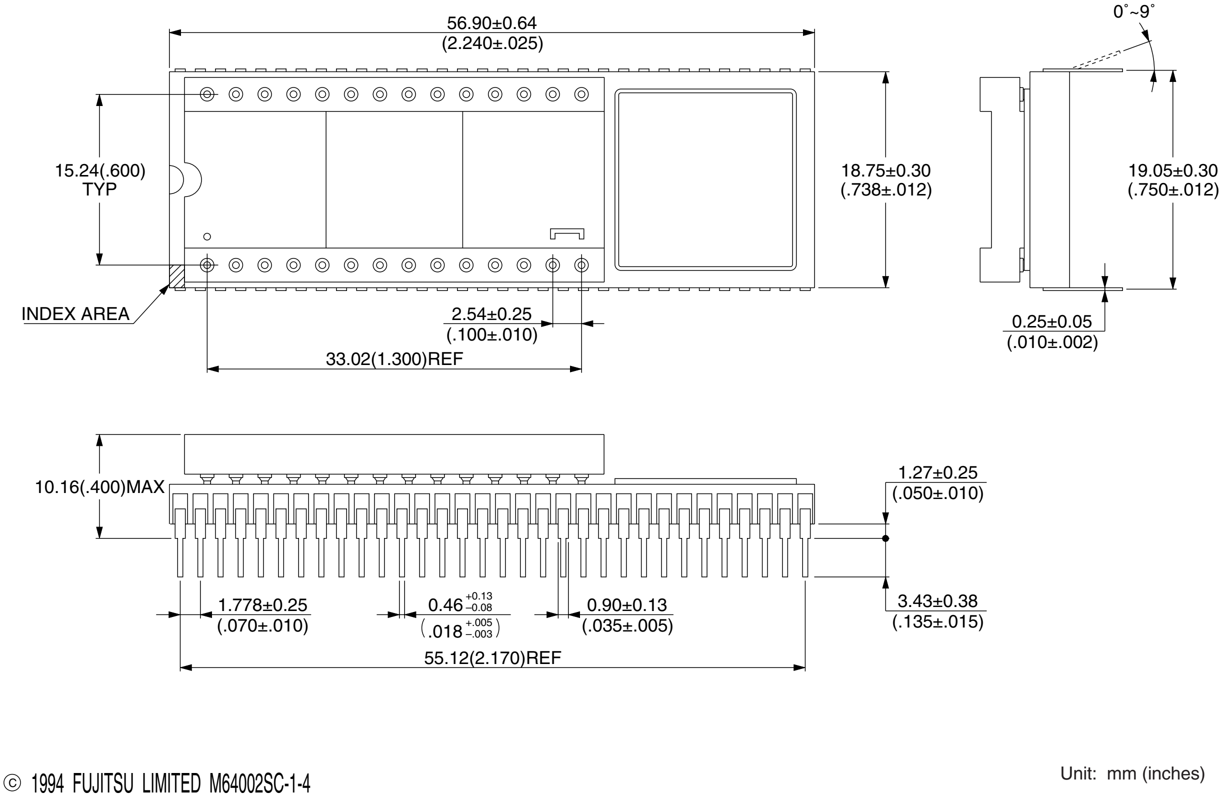
Unit: mm (inches)

■ External dimensions for MDP-64C-P02

Figure 1.6-5 External dimensions for MDP-64C-P02

Ceramic MDIP 64 pins  (MDP-64C-P02)	Lead pitch	70 mil
	Row spacing	750 mil
	Mother board material	Ceramic
	Connection socket material	Plastic

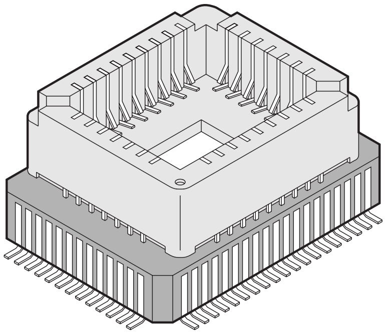
Ceramic MDIP 64 pins
(MDP-64C-P02)



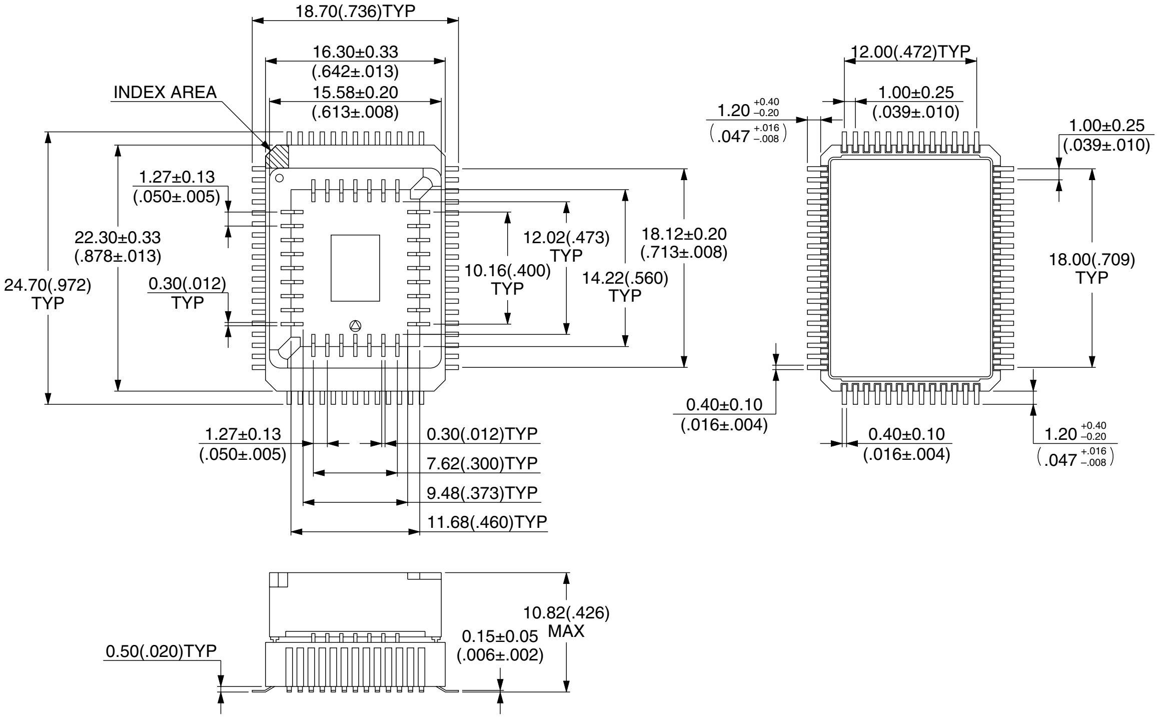
Unit: mm (inches)

■ External dimensions for MQP-64C-P01

Figure 1.6-6 External dimensions for MQP-64C-P01

Ceramic MQFP 64 pins  (MQP-64C-P01)	Lead pitch	1.00 mm
	Lead shape	Straight
	Motherboard material	Ceramic
	Material of the socket mounted	Plastic

Ceramic MQFP 64 pins
(MQP-64C-P01)



Unit: mm (inches)

1.7 Explanations of the Pin Functions

Table 1.7-1 "Explanations of the pin functions" and Table 1.7-2 "Explanation of the external EPROM socket pin functions (MB89PV530 only)" list the I/O pins and their functions of the MB89530/530H/530A series. The alphabetic characters in the I/O circuit format column in Table 1.7-1 "Explanations of the pin functions" correspond to those in the classification column in Table 1.8-1 "I/O circuit format."

■ Explanations of the pin functions

Table 1.7-1 Explanations of the pin functions

Pin number			Pin name	I/O circuit format	Explanation of functions

SH-DIP(*1)
MDIP(*2)
	QFP(*3)
	MQFP(*4)
		LQFP(*5)
		QFP(*6)			

30	23	22	X0	A	Connection pin for the crystal oscillation circuit or other oscillation circuits An external clock can be connected to X0. In this case, X1 must be opened.
31	24	23	X1		
28	21	20	MOD0	B	Input pin for setting the memory access Connected directly to V_{SS}.
29	22	21	MOD1		
27	20	19	$\overline{\text{RST}}$	C	Reset I/O pin. This pin is used as the CMOS input/output and hysteresis input with pull-up resistor. "L" is output from the pin in accordance with an internal reset request. The internal circuit is initialized with input of the "L" level.
56 to 49	49 to 42	48 to 41	P00 to P07	D	General-purpose I/O port
48 to 41	41 to 34	40 to 33	P10 to P17	D	General-purpose I/O port
40	33	32	P20/PWCK	E	General-purpose I/O port Resource I/O pin (input: hysteresis) Hysteresis input. This pin is also used as the PWC input.
39	32	31	P21/ PPG01	D	General-purpose I/O port This pin is also used as the PPG01 output.
38	31	30	P22/ PPG02	D	General-purpose I/O port This pin is also used as the PPG02 output.
37	30	29	P23	D	General-purpose I/O port
36	29	29	P24	D	General-purpose I/O port
35	28	27	P25	D	General-purpose I/O port

Table 1.7-1 Explanations of the pin functions (Continued)

Pin number			Pin name	I/O circuit format	Explanation of functions

SH-DIP(*1)
MDIP(*2)
	QFP(*3)
	MQFP(*4)
		LQFP(*5)
		QFP(*6)			

34	27	26	P26	D	General-purpose I/O port
33	26	25	P27	D	General-purpose I/O port
58	51	50	P30/ PPG03/ MCO	D	General-purpose I/O port This pin is also used as the PPG03 output.
59	52	51	P31/SCK1 (UCK1)/ LMCO	E	General-purpose I/O port Resource I/O pin (input: hysteresis) This pin is also used as the UART/SIO clock I/O.
60	53	52	P32/ SO1(UO1)	D	General-purpose I/O port This pin is also used as the UART/SIO serial data output.
61	54	53	P33/ SI1(UI1)	E	General-purpose I/O port Resource I/O pin (input: hysteresis) This pin is also used as the UART/SIO serial data input.
62	55	54	P34/PTO2	D	General-purpose I/O port This pin is also used as the PWM timer 2 output.
63	56	55	P35/PWC	E	General-purpose I/O port Resource I/O pin (input: hysteresis) This pin is also used as the PWC input.
1	58	57	P36/WTO	D	General-purpose I/O port The resource is output. This pin is also used as the PWC output.
2	59	58	P37/PTO1	D	General-purpose I/O port The resource is output. This pin is also used as the PWM timer 1 output.
3	60	59	P40/ INT20/EC	E	General-purpose I/O port Resource I/O pin (input: hysteresis) This pin is also used as the external interrupt input and 16-bit timer/counter input.
4	61	60	P41/ INT21/ SCK2	E	General-purpose I/O port Resource I/O pin (input: hysteresis) This pin is also used as the external interrupt input and SIO clock I/O.
5	62	61	P42/ INT22/ SO2/SDA	G	N-channel open-drain output Resource I/O pin (input: hysteresis) This pin is also used as the external interrupt input, SIO serial data output, and I^{2}C data line.

Table 1.7-1 Explanations of the pin functions (Continued)

Pin number			Pin name	I/O circuit format	Explanation of functions	

SH-DIP(*1)
MDIP(*2)
	QFP(*3)
	MQFP(*4)
		LQFP(*5)
		QFP(*6)				

6	63	62	P43/ INT23/S12/ SCL	G	N-channel open-drain output Resource I/O pin (input: hysteresis) This pin is also used as the external interrupt input, SIO serial data input, and I^2C clock I/O.	
7	64	63	P44/ INT24/ UCK2	E	General-purpose I/O port Resource I/O pin (input: hysteresis) This pin is also used as the external interrupt input and UART clock I/O.	
8	1	64	P45/ INT25/UO2	E	General-purpose I/O port Resource I/O pin (input: hysteresis) This pin is also used as the external interrupt input and UART data output.	
9	2	1	P46/ INT26/UI2	E	General-purpose I/O port External interrupt I/O pin (input: hysteresis) This pin is also used as the external interrupt input and UART data input.	
10	3	2	MOD2	B	MB89F538/ MB89F538L	Input pin for setting memory access mode Connected directly to V_{SS}.
			P47/ INT27/ ADST	E	Other than above	General-purpose I/O port Resource I/O pin (input: hysteresis) This pin is also used as the external interrupt input and A/D converter clock input pin.
11 to 18	4 to 11	3 to 10	P50/AN0 to P57/AN7	H	N-channel open-drain output port This pin is also used as the A/D converter analog input.	
22 to 24	15 to 17	14 to 16	P60/INT10 to P62/INT12	I	General-purpose input port Resource I/O pin (input: hysteresis) This pin is also used as the external interrupt input.	
25	18	17	P63/INT13	I	Single-clock system product	General-purpose input port Resource I/O pin (input: hysteresis) Also used as the external interrupt input.
			X0A	A	Dual-clock system product	Connection pin for subclocks

Table 1.7-1 Explanations of the pin functions (Continued)

Pin number			Pin name	I/O circuit format	Explanation of functions	

SH-DIP(*1)
MDIP(*2)
	QFP(*3)
	MQFP(*4)
		LQFP(*5)
		QFP(*6)				

26	19	18	P64	J	Single-clock system product	General-purpose input port
			X1A	A	Dual-clock system product	Connection pin for subclocks
64	57	56	V_{CC}	-	Power supply pin	
32	25	24	V_{SS}	-	Power supply pin (GND)	
19	12	11	AV_{CC}	-	A/D converter power supply pin	
20	13	12	AVR	-	A/D converter reference voltage input	
21	14	13	AV_{SS}	-	A/D converter power supply pin Used with the same voltage level with V_{SS}.	
57	50	49	C	-	MB89537H/537HC MB89538H/538HC MB89F538	Capacitor connection pin for power supply stabilization Connect the ceramic capacitor of about 0.1 µF to the exterior.
					MB89P538	If "Available" is selected for the step-down circuit stabilization time, V_{CC} is fixed. If "Unavailable" is selected for the step-down circuit stabilization time, V_{SS} is fixed.
					MB89PV530/535A/F538L MB89537/537C/537A/537AC MB89538/538C/538A/538AC	N.C. pin

*1:DIP-64P-M01 *2:MDP-64C-P02 *3:FPT-64P-M06
*4:MQP-64C-P01 *5:FPT-64P-M03 *6:FPT-64P-M09

Table 1.7-2 Explanation of the external EPROM socket pin functions (MB89PV530 only)

Pin number		Pin name	I/O circuit format	Explanation of functions
MDIP(*1)	MQFP(*2)			
65 66 67 68 69 70 71 72 73 74	66 67 68 69 70 71 72 73 74 75	A15 A12 A7 A6 A5 A4 A3 A2 A1 A0	O	Address output pin
75 76 77	77 78 79	01 02 03	I	Data input pin
78	80	V_{SS}	O	Power supply pin (GND)
79 80 81 82 83	82 83 84 85 86	04 05 06 07 08	I	Data input pin
84	87	$\overline{CE}$	O	ROM chip enable pin. "H" is always output in standby mode.
85	88	A10	O	Address output pin
86	89	$\overline{OE}$	O	ROM output enable pin. "L" is always output.
87 88 89	91 92 93	A11 A9 A8	O	Address output pin
90	94	A13	O	
91	95	A14	O	
92	96	V_{CC}	O	EPROM power supply pin
-	65 76 81 90	N.C.	O	Internal connection pin. Have it open at all times.

*1: MDP-64C-P02
*2: MQP-64C-P01

1.8 I/O Circuit Format

Table 1.8-1 "I/O circuit format" shows the I/O circuit format.
The alphabetic characters of the classification column in Table 1.7-1 "Explanations of the pin functions" correspond to those of the I/O circuit format column in Table 1.8-1 "I/O circuit format."

■ I/O circuit format

Table 1.8-1 I/O circuit format

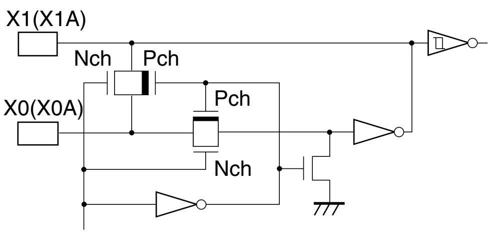
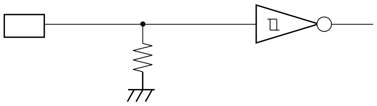
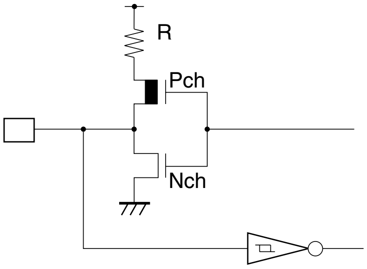
Classification	Circuit	Remarks
A		Oscillation feedback resistor • High-speed side = approx. 1 MΩ • Low-speed side = approx. 10 MΩ
B		• Hysteresis input • Pull-down resistor Contained in MB89535A, MB89537/537C, MB89538/538C, MB8537H/537HC, MB89538H/538HC, MB89537A/537AC, and MB89538A/538AC.
C		• Pull-up resistor is approx. 50 kΩ • Hysteresis input

Table 1.8-1 I/O circuit format (Continued)

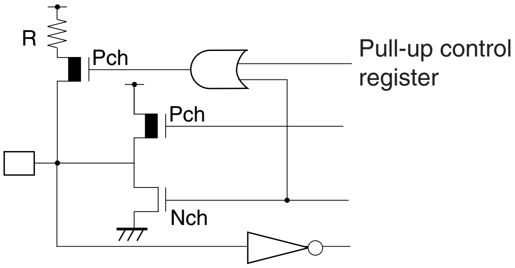
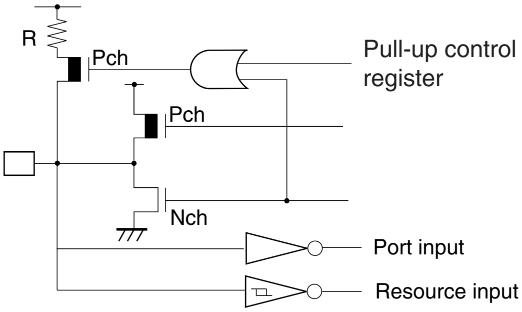
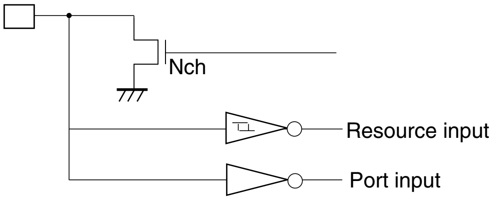
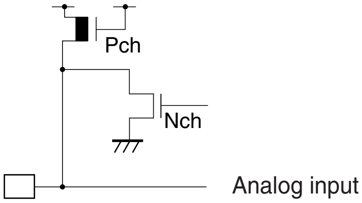
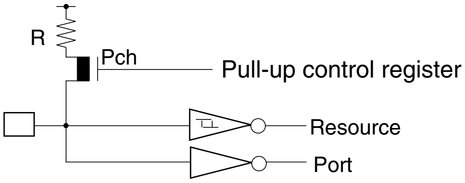
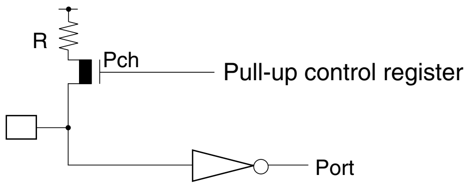
Classification	Circuit	Remarks
D		• CMOS I/O • Software pull-up resistor can be used. Approx. 50 kΩ
E	 R Pch Pch Nch Pull-up control register Port input Resource input	• CMOS I/O • Software pull-up resistor can be used. Approx. 50 kΩ
G		• N-channel open-drain output • Hysteresis input • CMOS input
H		• N-channel open-drain output • Analog input (A/D converter)
I		• Hysteresis input • CMOS input • Software pull-up resistor can be used. Approx. 50 kΩ

Table 1.8-1 I/O circuit format (Continued)

Classification	Circuit	Remarks
J		• CMOS input • Software pull-up resistor can be used. Approx. 50 kΩ

CHAPTER 2 HANDLING DEVICE

This chapter describes the precautions to be taken when using the MB89530/530H/530A series.

2.1 Notes on Handling Devices

This section describes the precautions to be taken when handling the power supply voltage and pins of the device.

■ Notes on Handling Devices

❍ **Maximum Rated Voltage (Preventing Latchup)**

The maximum rated voltage must not be exceeded.

Latchup may occur on CMOS ICs if voltage higher than V_{CC} or lower than V_{SS} is applied to input and output pins other than medium- to high-voltage pins, or if voltage higher than ratings is applied between V_{CC} and V_{SS}.

When latchup occurs, power supply current increases rapidly and might thermally damage elements. When using, take great care not to exceed the absolute maximum ratings.

Also, take care to prevent the analog power supply (AV_{CC}, AVR, DVR) and analog input from exceeding the digital power supply ($V_{CC)}$ when the analog system power supply is turned on and off.

❍ **Power Supply Voltage Fluctuations**

The power supply voltage must be made as stable as possible.

Although V_{CC} power supply voltage is assured to operate within the rated range, a rapid fluctuation of voltage could cause malfunctions, even if it occurs within the rated range. Stabilizing voltage supplied to the IC is therefore important. As stabilization guidelines, it is recommended to control power so that V_{CC} ripple fluctuations (P-P value) will be less than 10% of the standard V_{CC} value at the commercial frequency (50 to 60 Hz) and the transient fluctuation rate will be less than 0.1 V/ms at the time of a momentary fluctuation such as when power is switched.

❍ **Treatment of Unused Input Pins**

Leaving unused input pins open could cause malfunctions. They should be connected to a pull-up or pull-down resistor.

❍ **Treatment of Unused Input Pins**

Be sure to leave (internally connected) N.C. pins open.

❍ **Treatment of Power Supply Pins on Microcontroller with A/D and D/A Converters**

Connect to be $AV_{CC} = V_{CC}$ and AV_{SS} = AVR = DVR = V_{SS} even if the A/D and D/A converters are not in use.

❍ **Precautions when Using an External Clock**

Even when an external clock is used, an oscillation stabilization time is required after a power-on reset and exit from subclock mode or stop mode.

❍ Program Execution in RAM

Debugging of a program that is executed in RAM cannot be done even if MB89PV530 is used.

❍ Wild Register Function

Debugging cannot be made for the wild register with MB89PV530 and the tool. Check the operation with MB89P538, MB89F538, or MB89F538L on an actual machine.

❍ Detailed Processing of the C Pin of the MB89530/530H/530A Series

The MB89530/530H/530A series consists of the products listed in Table 2.1-1 "Pin Processing for the Products With and Without a Step-Down Circuit." The operation characteristic depends on whether a product contains a step-down circuit.

❍ Note to Noise in the External Reset Pin ($\overline{\text{RST}}$)

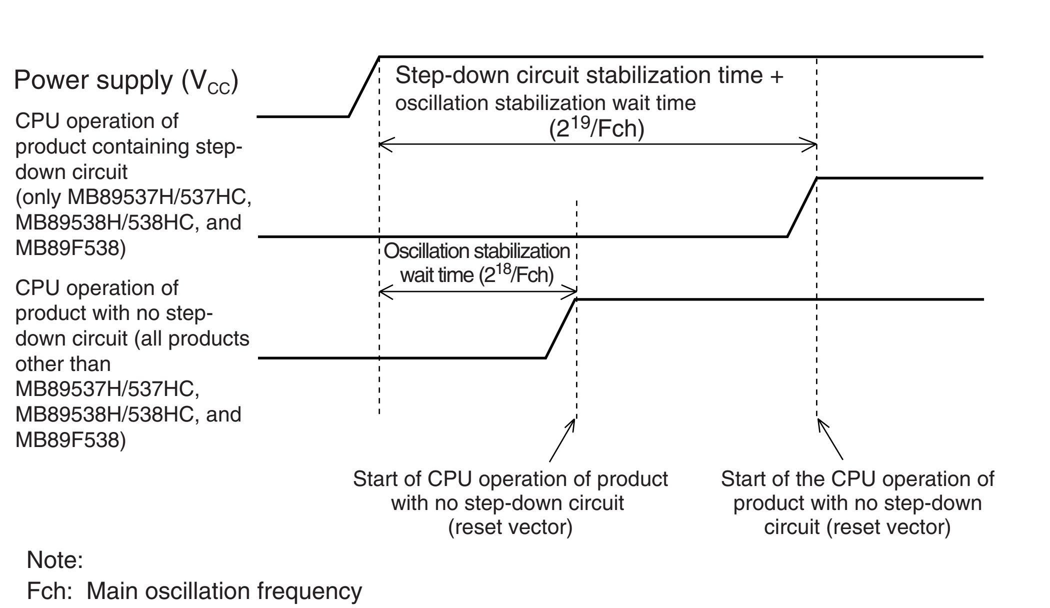
If the reset pulse applied to the external reset pin ($\overline{\text{RST}}$) does not meet the specifications, it may cause malfunctions. Use caution so that the reset pulse less than the specifications will not be fed to the external reset pin ($\overline{\text{RST}}$).

Table 2.1-1 Pin Processing for the Products With and Without a Step-Down Circuit

Product name	Operating voltage	Step-down circuit	Pin type	Pin processing
MB89PV530	2.7V to 5.5V	Not contained.	N.C pin	Not required.
MB89P538	2.7V to 5.5V	Contained.	C pin	V_{CC} fixed
		Not contained.		V_{SS} fixed
MB89F538	3.5V to 5.5V	Contained.	C pin	0.1 µF capacitor connection
MB89537H/537HC	3.5V to 5.5V	Contained.	C pin	0.1 µF capacitor connection
MB89538H/538HC	3.5V to 5.5V	Contained.	C pin	0.1 µF capacitor connection
MB89537/537C	2.2V to 3.6V	Not contained.	N.C pin	Not required.
MB89538/538C	2.2V to 3.6V	Not contained.	N.C pin	Not required.
MB89F538L	2.3V to 3.6V	Not contained.	N.C pin	Not required.
MB89537A/537AC	2.2V to 5.5V	Not contained.	N.C pin	Not required.
MB89538A/538AC	2.2V to 5.5V	Not contained.	N.C pin	Not required.
MB89535A	2.2V to 5.5V	Not contained.	N.C pin	Not required.

These products use the same internal resources. However, the operation sequence after power-on reset depends on whether a product contains a step-down circuit. Figure 2.1-1 "Operation Sequences After Power-On Reset Between Product Types" shows the sequence of operations after power-up for each model.

Figure 2.1-1 Operation Sequences After Power-On Reset Between Product Types


Power supply (V_{CC})
CPU operation of product containing step-down circuit (only MB89537H/537HC, MB89538H/538HC, and MB89F538)
CPU operation of product with no step-down circuit (all products other than MB89537H/537HC, MB89538H/538HC, and MB89F538)
Step-down circuit stabilization time + oscillation stabilization wait time (2^{19}/Fch)
Oscillation stabilization wait time (2^{18}/Fch)
Start of CPU operation of product with no step-down circuit (reset vector)
Start of the CPU operation of product with no step-down circuit (reset vector)
Note:
Fch: Main oscillation frequency

As shown in Figure 2.1-1 "Operation Sequences After Power-On Reset Between Product Types", the start of CPU operation of a product with a step-down circuit is slower than that of the product with no step-down circuit. This is because extra time is required for the step-down circuit to stabilize prior to normal operation of the step-down circuit.

For MB89P538, a product either with or without a step-down circuit can be selected by C pin processing. Therefore, use it depending on the mask version to be used.

CHAPTER 3 CPU

This chapter describes the functions and operations of the CPU.

3.1 Memory Space

The memory space of the MB89530/530H/530A series is 64 Kbytes and is made up of the I/O area, RAM area, and ROM area.
Some areas in the memory space, such as the general-purpose registers and vector table, are used for specific applications.

■ Configuration of the Memory Space

❍ I/O area (address: 0000_H - $007F_H$)

- This area is allocated to the control registers and data registers of the built-in peripheral devices.
- Since the I/O area is allocated to a part of the memory space, it can be accessed like normal memory. The area can be accessed faster using direct addressing.

❍ Extended I/O area (address: $C80_H$ - $C91_H$)

- The high-order byte address register, low-order byte address register, and data register of the wild register are allocated to this area.

❍ RAM area

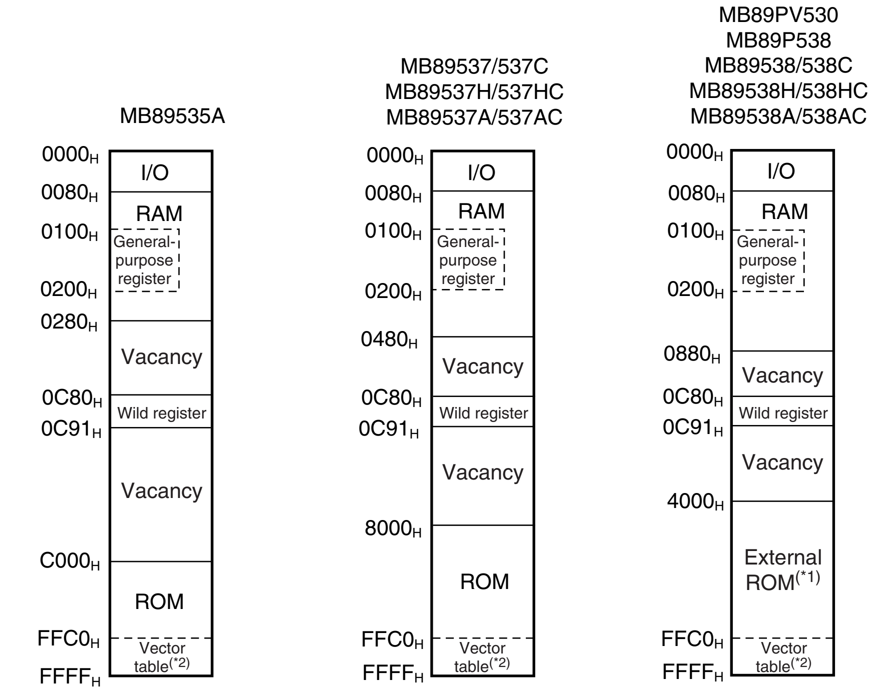
- Static RAM is contained as a built-in data area.
- The internal RAM size is dependent on the part number.
- 80_H to FF_H can be accessed faster using direct addressing.
- 100_H to $1FF_H$ can be used as a general-purpose register area.
- If a reset occurs during a write operation to RAM, data at the address to which data is being written cannot be guaranteed.

❍ ROM area

- ROM is contained as an internal program area.
- The internal ROM size is dependent on the part number.
- $FFC0_H$ to $FFFF_H$ are used as, for example, a vector table.

■ Memory Map

Figure 3.1-1 Memory Map



*1 The external ROM area is used only for MB89PV530.
*2 Vector table (reset, interrupt, and vector call

3.1.1 Special Areas

In addition to the I/O area, the general-purpose register area and vector table area are available as areas for specific applications.

■ General-purpose Register Area (Address: 0100_H - $01FF_H$)

- This area is used for 8-bit arithmetic operations and transfer. Supplementary registers are provided.
- Since this area is allocated to a part of the RAM area, it can also be used as normal RAM.
- When this area is used as a general-purpose register, it can be accessed faster using shorter instructions by general-purpose register addressing.

For details, see Section 3.2.2 "Register Bank Pointer (RP)" and Section 3.3 "General-purpose Registers".

■ Vector Table Area (Address: $FFC0_H$ - $FFFF_H$)

- This area is used as vector tables of the vector call instructions, interrupts, and reset.
- This area is allocated to the highest ranges of the ROM area, and the start address of the corresponding processing routine is set to the address of each vector table.

Table 3.1-1 "Vector Table" lists the addresses of the vector tables referenced corresponding to the vector call instructions, interrupts, and reset.

For details, see Section 3.4 "Interrupts", Section 3.5 "Reset", and "CALLV #vct" of Appendix B.3 "Special Instructions".

Table 3.1-1 Vector Table

Vector call instruction	Vector table address	
	High	Low
CALLV #0	$FFC0_H$	$FFC1_H$
CALLV #1	$FFC2_H$	$FFC3_H$
CALLV #2	$FFC4_H$	$FFC5_H$
CALLV #3	$FFC6_H$	$FFC7_H$
CALLV #4	$FFC8_H$	$FFC9_H$
CALLV #5	$FFCA_H$	$FFCB_H$
CALLV #6	$FFCC_H$	$FFCD_H$
CALLV #7	$FFCE_H$	$FFCF_H$

Table 3.1-1 Vector Table (Continued)

Interrupt name	Vector table address	
	High	Low
IRQF	$FFDC_H$	$FFDD_H$
IRQE	$FFDE_H$	$FFDF_H$
IRQD	$FFE0_H$	$FFE1_H$
IRQC	$FFE2_H$	$FFE3_H$
IRQB	$FFE4_H$	$FFE5_H$
IRQA	$FFE6_H$	$FFE7_H$
IRQ9	$FFE8_H$	$FFE9_H$
IRQ8	$FFEA_H$	$FFEB_H$
IRQ7	$FFEC_H$	$FFED_H$
IRQ6	$FFEE_H$	$FFEF_H$
IRQ5	$FFF0_H$	$FFF1_H$
IRQ4	$FFF2_H$	$FFF3_H$
IRQ3	$FFF4_H$	$FFF5_H$
IRQ2	$FFF6_H$	$FFF7_H$
IRQ1	$FFF8_H$	$FFF9_H$
IRQ0	$FFFA_H$	$FFFB_H$
Mode data	-- (*1)	$FFFD_H$
Reset vector	$FFFE_H$	$FFFF_H$

*1: $FFFC_H$ is not available (Set FF_H)

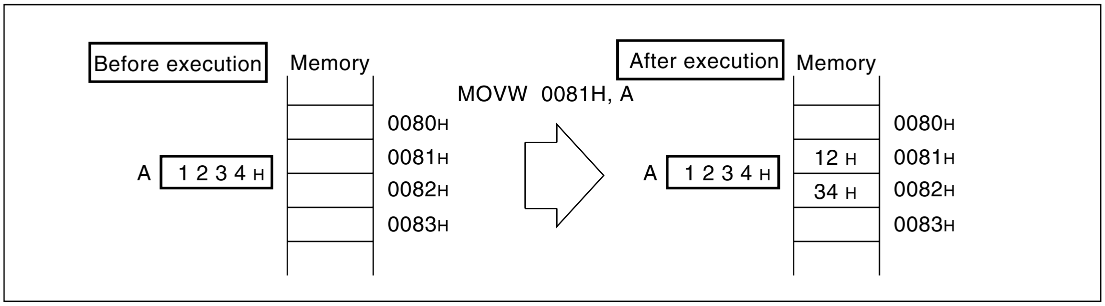
3.1.2 Storing 16-bit Data in Memory

Higher data of 16-bit data and stacks are stored in the areas of smaller address values on memory.

■ Storage of 16-bit Data on RAM

When writing 16-bit data into memory, the higher byte of the data is stored at the lower address. The lower byte of the data is stored at the next address. When reading memory, the same procedure is executed.Figure 3.1-2 "Storing 16-bit Data in Memory" shows the storing 16-bit data in memory.

Figure 3.1-2 Storing 16-Bit Data in Memory



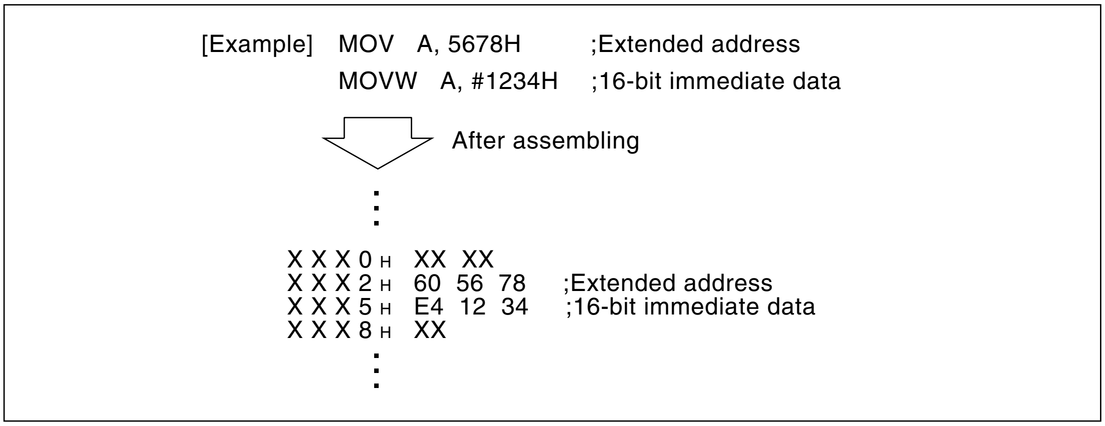
■ Storage of a 16-bit Operand

Also when 16 bits are specified in an operand of an instruction, the higher byte is stored at the nearby operation code (instruction) and the lower byte is stored at the next address.

This is the same if the operand points to a memory address or is 16-bit immediate data.

Figure 3.1-3 "16-bit Data in Instructions" shows the storing 16-bit data in instructions.

Figure 3.1-3 16-Bit Data in Instructions



```
[Example]  MOV   A, 5678H      ;Extended address
           MOVW  A, #1234H     ;16-bit immediate data
```

```
    :
XXX0H  XX  XX
XXX2H  60  56  78    ;Extended address
XXX5H  E4  12  34    ;16-bit immediate data
XXX8H  XX
    :
```

■ Storage of 16-bit Data on the Stack

Data of the 16-bit length register saved on the stack due, for example, to an interrupt, is also stored in the same manner, with the higher byte at the smaller address.

3.2 Dedicated Registers

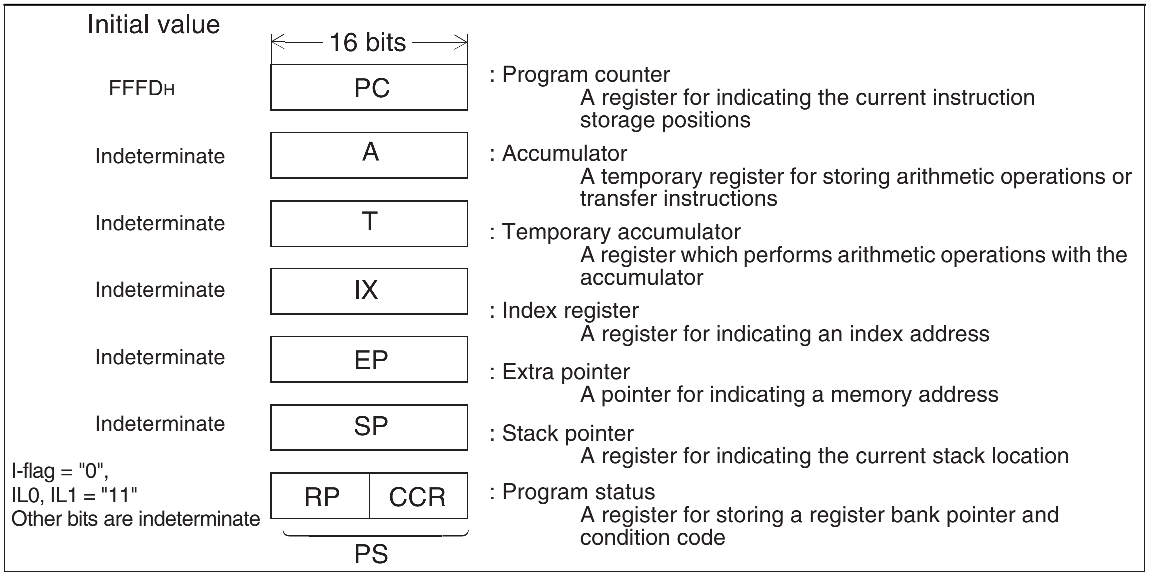
The dedicated registers in the CPU consist of the program counter (PC), two arithmetic operation registers (A and T), three address pointers (IX, EP, and SP), and the program status (PS). All registers are 16 bits.

■ Dedicated Register Configuration

The dedicated registers in the CPU consist of seven 16-bit registers. Some of these registers are also able to be used as 8-bit registers, using the lower 8 bits only.

Figure 3.2-1 "Dedicated Register Configuration" shows the structure of the dedicated registers.

Figure 3.2-1 Dedicated Register Configuration



: Program counter
A register for indicating the current instruction storage positions

: Accumulator
A temporary register for storing arithmetic operations or transfer instructions

: Temporary accumulator
A register which performs arithmetic operations with the accumulator

: Index register
A register for indicating an index address

: Extra pointer
A pointer for indicating a memory address

: Stack pointer
A register for indicating the current stack location

: Program status
A register for storing a register bank pointer and condition code

■ Dedicated Register Functions

❍ Program counter (PC)

The program counter is a 16-bit counter that indicates the memory address of the instruction currently being executed by the CPU. Instruction execution, interrupts, resets, and similar update the contents of the program counter. The initial value during a reset is the read address of the mode data (FFFD$_H$).

❍ Accumulator (A)

The accumulator is a 16-bit arithmetic operation register. The accumulator is used to perform arithmetic operations and data transfers with data in memory or in other registers such as the temporary accumulator (T). The content of the accumulator can be treated as either word (16-bit) or byte (8-bit) data. Only the lower 8 bits (AL) of the accumulator are used for byte arithmetic operations or transfers. In this case, the upper 8 bits (AH) remain unchanged. The content of the accumulator after a reset is indeterminate.

❍ Temporary accumulator (T)

The temporary accumulator is an auxiliary 16-bit arithmetic operation register used to perform arithmetic operations with the data in the accumulator (A). The content of the temporary accumulator is treated as word data (16-bit) for word-length arithmetic operations with the accumulator and as byte data (8-bit) for byte-length arithmetic operations. For byte-length arithmetic operations, only the lower 8 bits of the temporary accumulator (TL) are used and the upper 8 bits (TH) are not used.

Executing a transfer instruction to transfer data to the accumulator (A) automatically transfer the previous content of the accumulator to the temporary accumulator. In this case also, a byte transfer leaves the upper 8 bits of the temporary accumulator (TH) unchanged. The content of the temporary accumulator after a reset is indeterminate.

❍ Index register (IX)

The index register is a 16-bit register used to hold the index address. The index register is used in conjunction with a single byte offset value (-128 to +127). Adding the sign-extended offset value to the index address generates the memory address for data access. The content of the index register after a reset is indeterminate.

❍ Extra pointer (EP)

The extra pointer is a 16-bit register used to hold a memory address for data access. The content of the extra pointer after a reset is indeterminate.

❍ Stack pointer (SP)

The stack pointer is a 16-bit register used to hold the address referenced during operations such as interrupts, subroutine calls, and the stack save and restore instructions. The value of the stack pointer during program execution is the address of the most recently saved data on the stack. The content of the stack pointer after a reset is indeterminate.

❍ Program status (PS)

The program status is a 16-bit control register. The upper 8 bits contain the register bank pointer (RP) which points to the address of the current general-purpose register bank.

The lower 8 bits contain the condition code register (CCR) which contains flags indicating the current CPU status. The two 8-bit registers which form the program status cannot be accessed independently (the program status can only be accessed by the MOVW A,PS and MOVW PS,A instructions).

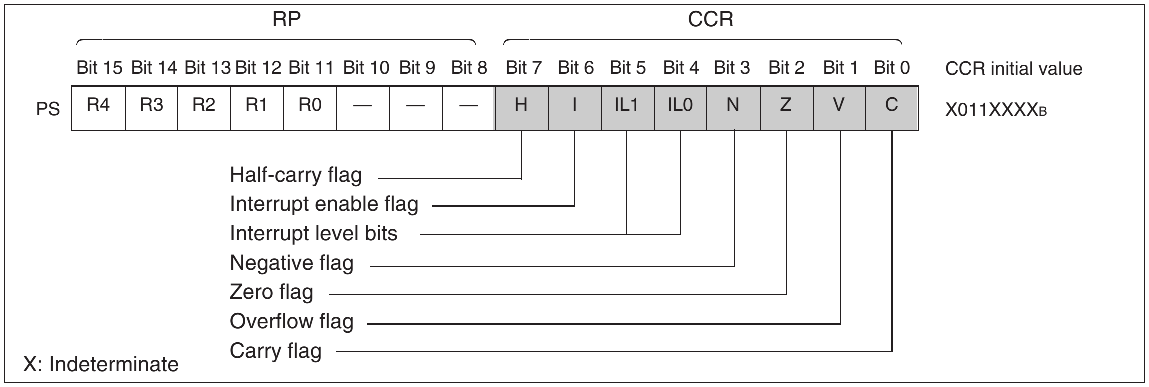
Refer to the "F^{2}MC-8L Programming Manual" for details on using the dedicated registers

3.2.1 Condition Code Register (CCR)

The condition code register (CCR) located in the lower 8 bits of the program status (PS) consists of the C, V, Z, N, and H bits indicating the results of arithmetic operations and the contents of transfer data, and the I, IL1, and IL0 bits for control whether or not the CPU accepts interrupt requests.

■ Structure of Condition Code Register (CCR)

Figure 3.2-2 Structure of Condition Code Register



Note:

The condition code register is part of the program status (PS) and cannot be accessed independently.

Reference:

In practice, the flag bits are rarely fetched and used directly. Instead, the bits are used indirectly by instructions such as branch instructions (such as BNZ) or the decimal adjustment instructions (DAA, DAS). The content of the flags after a reset is indeterminate.

■ Arithmetic Operation Result Bits

❍ Half-carry flag (H)

Set when a carry from bit 3 to bit 4 or a borrow from bit 4 to bit 3 occurs as a result of an arithmetic operation. Cleared otherwise. As this flag is for the decimal adjustment instructions, do not use this flag in cases other than addition or subtraction.

❍ Negative flag (N)

Set if the most significant bit (MSB) is set to 1 as a result of an arithmetic operation. Cleared when the bit is set to 0.

❍ Zero flag (Z)

Set when an arithmetic operation results in 0. Cleared otherwise.

❍ Overflow flag (V)

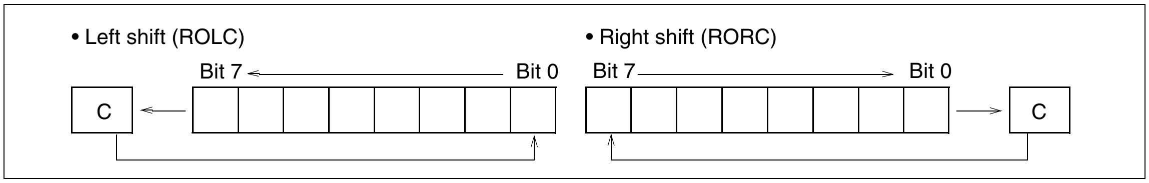
Set if the complement on 2 overflows as a result of an arithmetic operation. Reset if the overflow does not occur.

❍ Carry flag (C)

Set when a carry from bit 7 or borrow to bit 7 occurs as a result of an arithmetic operation. Cleared otherwise. Set to the shift-out value in case of a shift instruction.

Figure 3.2-3 "Change of Carry Flag by Shift Instruction" shows the change of the carry flag by a shift instruction.

Figure 3.2-3 Change of Carry Flag by Shift Instruction



■ Interrupt Acceptance Control Bit

❍ Interrupt enable flag (I)

Interrupt is enabled when this flag is set to "1" and the CPU accepts interrupt. Interrupt is prohibited when this flag is set to "0" and the CPU does not accept interrupt.

The initial value after a reset is "0".

Normal practice is to set the flag to "1" by the SETI instruction and clear to "0" by the CLRI instruction.

❍ Interrupt level bits (IL1, IL0)

These bits indicate the level of the interrupt currently being accepted by the CPU. The value is compared with the interrupt level setting registers (ILR1 to ILR4) which have a setting for each peripheral function interrupt request (IRQ0 to IRQF).

Given that the interrupt enable flag is enabled (I = "1"), the CPU only performs interrupt processing for interrupt requests with an interrupt level value that is less than the value of these bits. Table 3.2-1 "Interrupt Level" lists the interrupt level priorities. The initial value after a reset is "11_B".

Table 3.2-1 Interrupt Level

IL1	IL0	Interrupt level	High-low
0	0	1	High ↕ Low (no interrupt)
0	1		
1	0	2	
1	1	3	

Reference:

The interrupt level bits (IL1, IL0) are normally "11" when the CPU is not processing an interrupt (during main program execution).

See Section 3.4 "Interrupts" for details on interrupts.

3.2.2 Register Bank Pointer (RP)

The register bank pointer (RP) located in the upper 8 bits of the program status (PS) indicates the address of the general-purpose register bank currently in use. The RP is converted to form the actual address in general-purpose register addressing.

■ Structure of Register Bank Pointer (RP)

Figure 3.2-4 "Structure of Register Bank Pointer" shows the structure of the register bank pointer.

Figure 3.2-4 Structure of Register Bank Pointer

	RP								CCR								
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RP initial value
PS	R4	R3	R2	R1	R0	—	—	—	H	I	IL1	IL0	N	Z	V	C	XXXXXXXX$_B$

X: Indeterminate

The register bank pointer indicates the address of the register bank currently in use. Figure 3.2-5 "Rule for Conversion of Actual Addresses of General-purpose Register Area" shows the relationship between the pointer contents and the actual address is based on the conversion rule.

Figure 3.2-5 Rule for Conversion of Actual Addresses of General-purpose Register Area

									Upper bits of RP					Lower operation codes		
	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"1"	R4	R3	R2	R1	R0	b2	b1	b0
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Generated addresses	A15	A14	A13	A12	A10	A11	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

The register bank pointer points to the memory block (register bank) in the RAM area that is used for general-purpose registers. A total of 32 register banks are available. A register bank is specified by setting a value between 0 and 31 in the upper 5 bits of the register bank pointer. Each register bank contains 8-bit general-purpose registers. Registers are specified by the lower 3 bits of the operation codes.

Using the register bank pointer, the addresses 0100$_H$ to 01FF$_H$ can be used as the general-purpose register area. However, the available area is limited on some products if internal RAM only is used. The initial value after a reset is indeterminate.

Note:

Before using a general-purpose register, set the register bank pointer (RP).

The register bank pointer is part of the program status (PS) and cannot be accessed independently.

3.3 General-purpose Registers

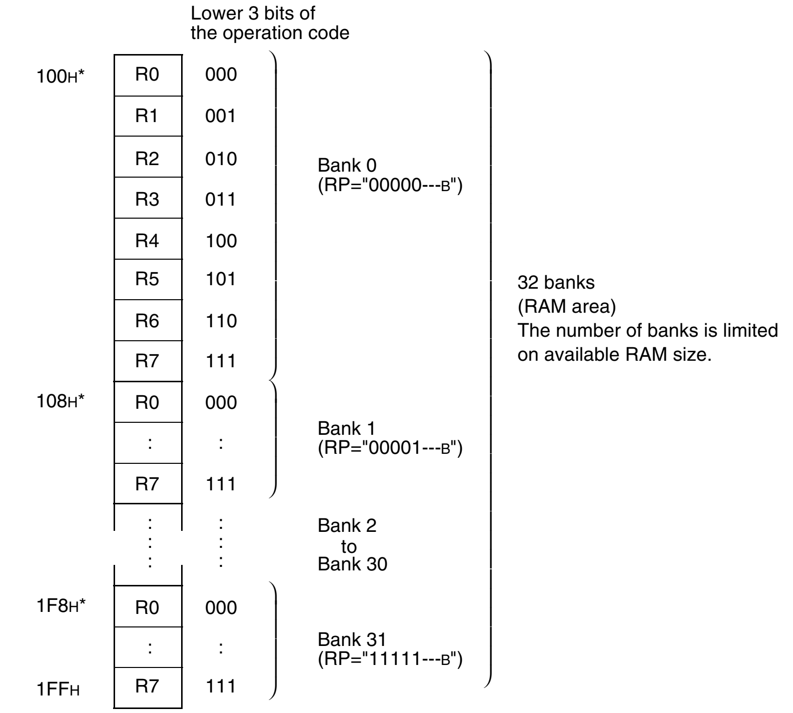
The general-purpose registers are a memory block made up of banks, with 8 x 8-bit registers per bank.
The register bank pointer (RP) is used to specify the register bank.
The function permits the use of up to 32 banks.
Register banks are valid for interrupt processing, vector call processing, and subroutine calls.

■ Structure of General-purpose Registers

- The general-purpose registers are 8 bits and located in the register banks of the general-purpose register area (in RAM).
- One bank contains eight registers (R0 to R7) and up to a total of 32 banks.
- The register bank currently in use is specified by the register bank pointer (RP). The lower three bits of the operation code specify general-purpose register 0 (R0) to general-purpose register 7 (R7).

Figure 3.3-1 "Register Bank Structure" shows the register bank structure.

Figure 3.3-1 Register Bank Structure



*: The top address of a register bank = 0100H + 8 × (upper 5 bits of RP)

see Section 3.1.1 "Special Areas" for the general-purpose register area available for each product.

■ Features of General-purpose Registers

General-purpose registers have the following features:

- RAM can be accessed at high-speed using short instructions (general-purpose register addressing).
- Registers are grouped in blocks in the form of register banks. This simplifies the process of saving register contents and dividing registers by function.

Dedicated register banks can be permanently assigned for each interrupt processing or vector call (CALLV #0 to #7) processing routine by general-purpose register. For example, register bank 4 interrupt 2.

For example, a particular interrupt processing routine only uses a particular register bank which cannot be written to unintentionally by other routines. The interrupt processing routine only needs to specify its dedicated register bank at the start of the routine to effectively save the general-purpose registers in use prior to the interrupt. Therefore, saving the general-purpose registers to the stack or other memory location is not necessary. This allows high-speed interrupt handling while maintaining simplicity.

Also, as an alternative to saving general-purpose registers in subroutine calls, register banks can be used to create reentrant programs (programs that do not use fixed addresses and can be entered more than once) usually made by the index register (IX).

Note:

If an interrupt processing routine changes the register bank pointer (RP), ensure that the program does not also change the interrupt level bits in the condition code register (CCR: IL1, 0) when specifying the register bank.

3.4 Interrupts

The MB89530/530H/530A series has 15 interrupt request input corresponding to peripheral functions. An interrupt level can be set independently.
If an interrupt request output is enabled in the peripheral function, an interrupt request from a peripheral function is compared with the interrupt level in the interrupt controller. The CPU performs interrupt operation according to how the interrupt is accepted. The CPU wakes up from standby modes, and returns to the interrupt or normal operation.

■ Interrupt Requests from Peripheral Functions

Table 3.4-1 "Interrupt Request and Interrupt Vector" lists the interrupt requests corresponding to the peripheral functions. On acceptance of an interrupt, execution branches to the interrupt processing routine. The contents of interrupt the vector table address corresponding to the interrupt request specifies the branch destination address for the interrupt processing routine.

An interrupt processing level can be for each interrupt request in the interrupt level setting registers (ILR1, ILR2, ILR3, ILR4). Three levels are available.

If an interrupt request with the same or lower level occurs during execution of an interrupt processing routine, the letter interrupt is not normally processed until the current interrupt processing routine completes. If interrupt request set the same level occur simultaneously, the highest priority is IRQ0.

Table 3.4-1 Interrupt Requests and Interrupt Vectors

Interrupt request	Vector table address		Bit name of the interrupt level setting register	Priority if interrupt requests with the same level occur simultaneously
	Higher	Lower		
IRQ0 (external interrupt (edge) INT10 to INT11)	FFFA$_H$	FFFB$_H$	L01, L00	High ↑
IRQ1 (external interrupt (edge) INT12 to INT13)	FFF8$_H$	FFF9$_H$	L11, L10	
IRQ2 I^{2}C	FFF6$_H$	FFF7$_H$	L21, L20	
IRQ3 (setting not available)	FFF4$_H$	FFF5$_H$	L31, L30	
IRQ4 (external interrupt (level) INT20 to INT27)	FFF2$_H$	FFF3$_H$	L41, L40	
IRQ5 (PWM timer 1)	FFF0$_H$	FFF1$_H$	L51, L50	
IRQ6 (PWM timer 2)	FFEE$_H$	FFEF$_H$	L61, L60	
IRQ7 (PWC)	FFEC$_H$	FFED$_H$	L71, L70	
IRQ8 (16-bit timer/counter interrupt)	FFEA$_H$	FFEB$_H$	L81, L80	
IRQ9 (8-bit serial I/O)	FFE8$_H$	FFE9$_H$	L91, L90	
IRQA (UART/SIO)	FFE6$_H$	FFE7$_H$	LA1, LA0	
IRQB (UART reception)	FFE4$_H$	FFE5$_H$	LB1, LB0	
IRQC (UART transmission)	FFE2$_H$	FFE3$_H$	LC1, LC0	
IRQD (A/D converter)	FFE0$_H$	FFE1$_H$	LD1, LD0	
IRQE (timebase timer)	FFDE$_H$	FFDF$_H$	LE1, LE0	↓
IRQF (watch prescaler)	FFDC$_H$	FFDD$_H$	LF1, LF0	Low

3.4.1 Interrupt Level Setting Registers (ILR1, ILR2, ILR3, ILR4)

The interrupt level setting registers (ILR1, ILR2, ILR3, ILR4) together contain 16 blocks of 2-bit data, with each data corresponding to an interrupt request from a peripheral function. The interrupt level for each interrupt is set in that interruptis corresponding 2-bit data (interrupt level setting bits).

■ Structure of Interrupt Level Setting Registers (ILR1, ILR2, ILR3, ILR4)

Figure 3.4-1 Structure of the Interrupt Level Setting Register

Register	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
ILR1	0 0 7 BH	L31	L30	L21	L20	L11	L10	L01	L00	11111111B
		W	W	W	W	W	W	W	W	
ILR2	0 0 7 CH	L71	L70	L61	L60	L51	L50	L41	L40	11111111B
		W	W	W	W	W	W	W	W	
ILR3	0 0 7 DH	LB1	LB0	LA1	LA0	L91	L90	L81	L80	11111111B
		W	W	W	W	W	W	W	W	
ILR4	0 0 7 EH	LF1	LF0	LE1	LE0	LD1	LD0	LC1	LC0	11111111B
		W	W	W	W	W	W	W	W	

W: write only

Two bits of the interrupt level setting registers are allocated to each interrupt request. The value of the interrupt level setting bits in these registers sets the interrupt priority (interrupt levels 1 to 3).

The interrupt level setting bits are compared with the interrupt level bits in the condition code register (CCR: IL1, IL0).

The CPU does not accept interrupt requests set to interrupt level 3. Table 3.4-2 "Interrupt Level Setting Bits and Interrupt Level" shows the relationship between the interrupt level setting bits and the interrupt levels.

Table 3.4-2 Interrupt Level Setting Bits and the Interrupt Level

L01 to LF1	L00 to LF0	Request Interrupt level	High-low
0	0	1	High ↕ Low (no interrupt)
0	1		
1	0	2	
1	1	3	

X: 0 to F Associated interrupt numbers

Reference:

The interrupt level bits in the condition code register (CCR: IL1, IL0) are normally "11_B" during main program execution.

Note:

As the IRL1, ILR2, ILR3, and ILR4 registers are write-only, the bit manipulation instructions (SETB, CLRB) cannot be used.

3.4.2 Interrupt Processing

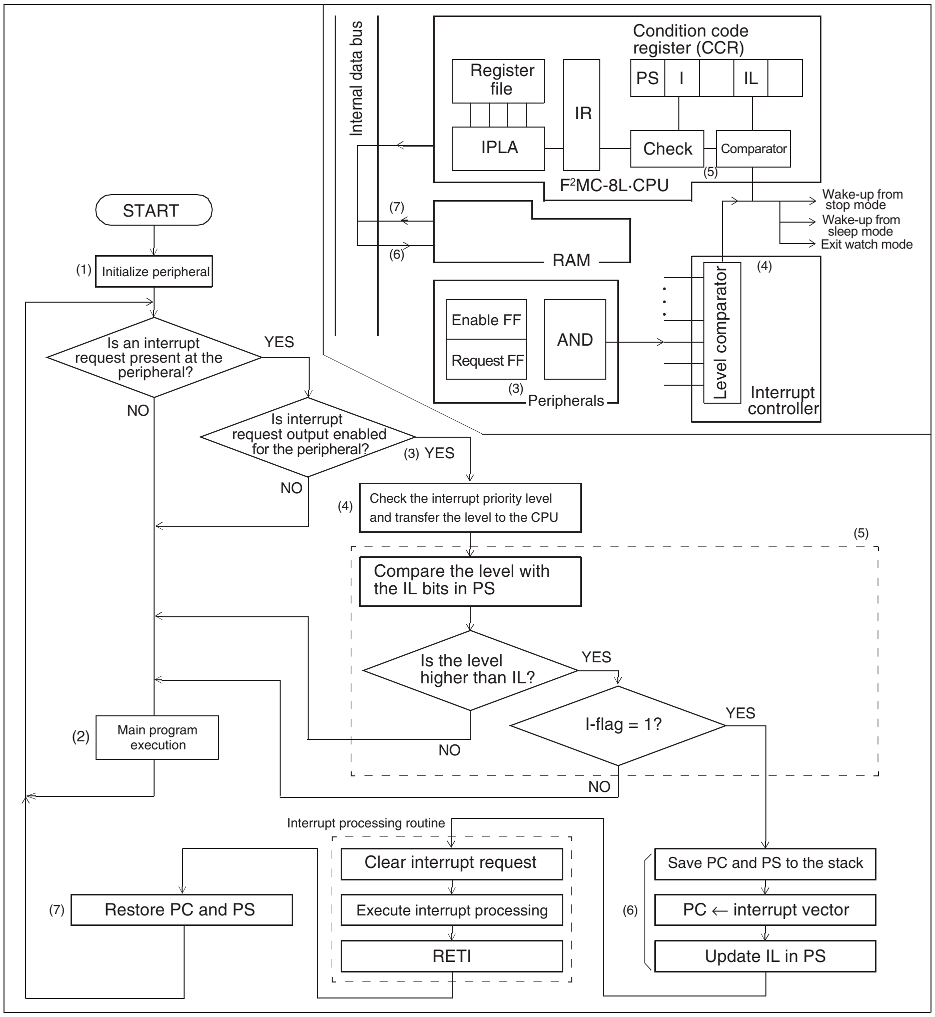
The interrupt controller transmits the interrupt level to the CPU when an interrupt request is generated by a peripheral function. If the CPU is able to receive the interrupt, the CPU temporarily halts the currently executing program and executes the interrupt processing routine.

■ Interrupt Processing

The procedure for interrupt operation is performed in the following order: interrupt source generated at peripheral function, set the interrupt request flag bit (request FF), discriminate the interrupt request enable bit (enable FF), the interrupt level (ILR1, ILR2, ILR3, ILR4 and CCR: IL1, IL0), simultaneously generated interrupt requests with the same level, then check the interrupt enable flag (CCR: I).

Figure 3.4-2 "Interrupt Processing" shows the interrupt processing.

Figure 3.4-2 Interrupt Processing


Internal data bus
Condition code register (CCR)
Register file
IR
PS
I
IL
IPLA
Check
Comparator
(5)
F²MC-8L·CPU
(7)
(6)
RAM
Wake-up from stop mode
Wake-up from sleep mode
Exit watch mode
(4)
Enable FF
Request FF
AND
(3)
Peripherals
Level comparator
Interrupt controller
START
(1) Initialize peripheral
Is an interrupt request present at the peripheral?
YES
NO
Is interrupt request output enabled for the peripheral?
(3) YES
NO
(4) Check the interrupt priority level and transfer the level to the CPU
(5)
Compare the level with the IL bits in PS
Is the level higher than IL?
YES
NO
I-flag = 1?
YES
NO
(2) Main program execution
Interrupt processing routine
Clear interrupt request
Execute interrupt processing
RETI
(7) Restore PC and PS
(6)
Save PC and PS to the stack
PC ← interrupt vector
Update IL in PS

(1) After a reset, all interrupt requests are disabled.
Initialize the peripheral functions that are to generate interrupts in the peripheral function initialization program, set the interrupt levels in the appropriate interrupt level setting registers (ILR1, ILR2, ILR3, ILR4), and start peripheral function. The interrupt level can be set to 1, 2 or 3. Level 1 is the highest priority, followed by level 2. Setting level 3 disables the interrupt for that peripheral function.

(2) Execute the main program (for multiple interrupts, execute the interrupt processing routine).

(3) The interrupt request flag bit (request FF) for a peripheral function is set to "1" when the peripheral function generates an interrupt source. If the interrupt request enable bit for the peripheral function is set to "enable" (enable FF = "1"), the peripheral function outputs the interrupt request to the interrupt controller.

(4) The interrupt controller continuously monitors for interrupt requests from the peripheral functions and passes the interrupt level of the current interrupt request with the highest interrupt level to the CPU. The interrupt controller also evaluates the priority order if requests with the same level are present simultaneously.

(5) If the interrupt level received by the CPU has a higher priority (a lower level value) than the level set in the interrupt level bits in the condition code register (CCR: IL1, IL0), the CPU checks the interrupt enable flag (CCR: I) and receives the interrupt if interrupts are enabled (CCR: I = "1").

(6) The CPU saves the contents of the program counter (PC) and program status (PS) on the stack, reads the top address of the interrupt processing routine from the interrupt vector table for the interrupt, updates the interrupt level bits in the condition code register (CCR: IL1, IL0) with the received interrupt level, and starts execution of the interrupt processing routine.

(7) Finally, on execution of the RETI instruction, the CPU restores the program counter (PC) and program status (PS) values saved on the stack and resumes execution from the instruction following the last instruction executed before the interrupt.

Note:

As the interrupt request flag bit of a peripheral function is not cleared automatically when an interrupt request is received, the bit must be cleared by the program (normally, by writing "0" to the interrupt request flag bit) at interrupt processing routine.

An interrupt wakes up the CPU from standby mode (low-power consumption). see Section 3.7 "Standby Modes (Low-Power Consumption)" for details.

Reference:

If the interrupt request flag bit is cleared at the top of the interrupt processing routine, the peripheral function that has generated the interrupt becomes able to generate another interrupt during execution of the interrupt processing routine (resetting the interrupt request flag bit). However, the interrupts are not normally accepted until the current processing routine completes.

3.4.3 Multiple Interrupts

Multiple interrupts can be performed by setting different interrupt levels to the interrupt level setting register (ILR1 to ILR4) for two or more interrupt requests from peripheral functions.

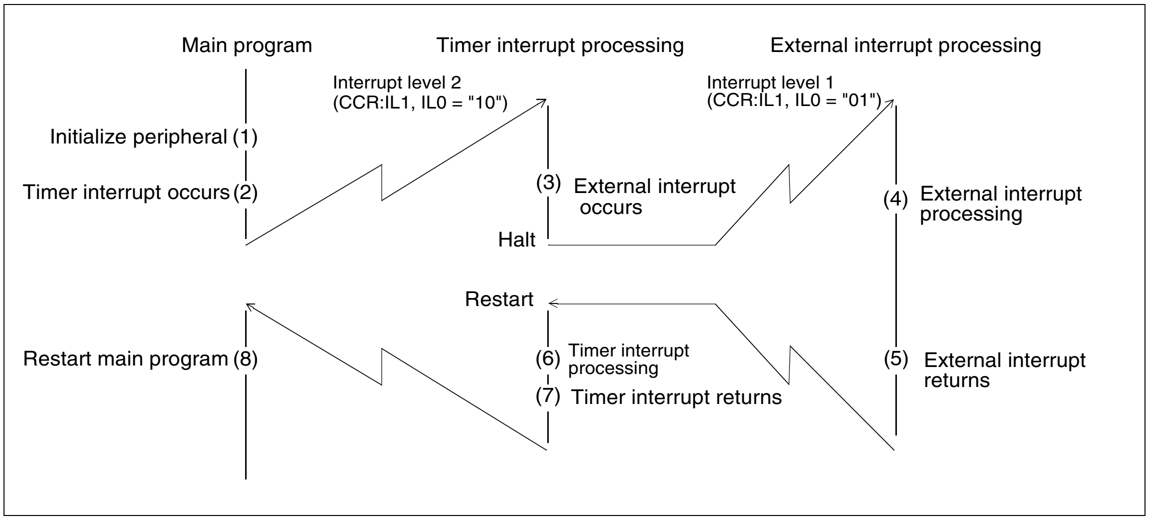
■ Multiple Interrupts

If the interrupt request having the higher interrupt levels occurs during the interrupt processing routines, the CPU halts the current interrupt process and switches to accept the interrupt with the higher priority. Interrupt levels can be set in the range 1 to 3. However, the CPU does not accept interrupt requests set to interrupt level 3.

❍ Example of multiple interrupts

As an example of multiple interrupt processing, assume that an external interrupt has a higher priority than the timer interrupt. The timer interrupt is set to level 2 and the external interrupt is set to level 1. Figure 3.4-3 "Example of Multiple Interrupts" shows the processing when the external interrupt occurs during execution of timer interrupt processing.

Figure 3.4-3 Example of Multiple Interrupts


Main program
Timer interrupt processing
External interrupt processing
Interrupt level 2
(CCR:IL1, IL0 = "10")
Interrupt level 1
(CCR:IL1, IL0 = "01")
Initialize peripheral (1)
Timer interrupt occurs (2)
(3) External interrupt occurs
Halt
(4) External interrupt processing
Restart
Restart main program (8)
(6) Timer interrupt processing
(7) Timer interrupt returns
(5) External interrupt returns

- During execution of timer interrupt processing, the interrupt level bits in the condition code register (CCR:IL1, IL0) are automatically set to the same value as the interrupt level setting register (ILR1, ILR2, ILR3, ILR4) corresponding to the timer interrupt (level 2 in this example). If the interrupt request set to higher interrupt level (level 1 in this example) occurs at this time, the interrupt processing has priority.
- To temporarily disable multiple interrupts during the timer interrupt, the interrupt enable flag in the condition code register is set to "interrupts disabled" (CCR: I = "0") or the interrupt level bits (IL1, IL0) set to "00".
- On execution of the interrupt return instruction (RETI) at the completion of interrupt processing, the CPU restores the program counter (PC) and program status (PS) values saved on the stack and resumes execution of the interrupted program.

  Restoring the program status (PS) returns the condition code register (CCR) to the value prior to the interrupt.

3.4.4 Interrupt Processing Time

The total time from the generation of an interrupt request until control passes to the interrupt processing routine is the sum of the time required to complete execution of the current instruction and the interrupt handling time (the time required to prepare for interrupt processing). The maximum time for this process is 30 instruction cycles.

■ Interrupt Processing Time

When an interrupt request occurs, the time until the interrupt is accepted and the interrupt processing routine is executed includes the interrupt request sampling time and the interrupt handling time.

❍ Interrupt request sampling time

Whether or not an interrupt request has occurred is determined by sampling and testing for interrupt requests during the final cycle of each instruction. Therefore, the CPU is unable to identify interrupt requests during execution of an instruction. The longest delay occurs when an interrupt request is generated immediately after starting execution of a DIVU instruction, which has the longest instruction cycles (21 instruction cycles).

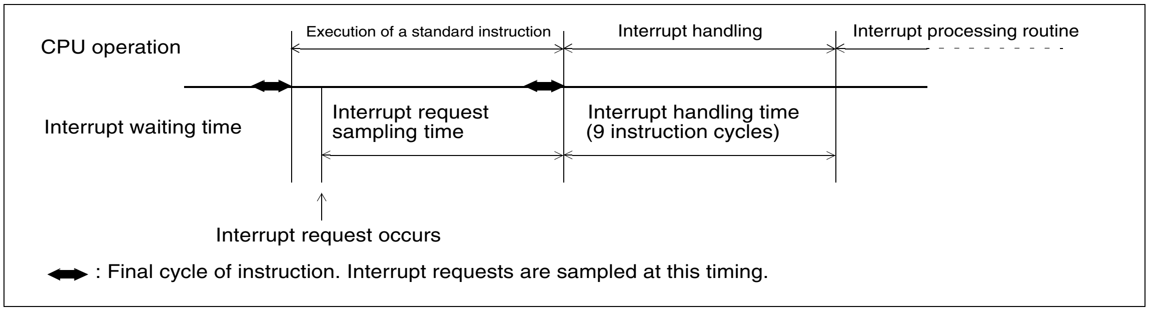
❍ Interrupt handling time

Nine instruction cycles are required to perform the following preparation for interrupt processing after the CPU accepts an interrupt request:

- Save the program counter (PC) and program status (PS).
- Set the top address of the interrupt processing routine (the interrupt vector) in the PC.
- Update the interrupt level bits (PS:CCR: IL1, IL0) in the program status (PS).

Figure 3.4-4 "Interrupt Processing Time" shows the interrupt processing time.

Figure 3.4-4 Interrupt Processing Time


CPU operation
Execution of a standard instruction
Interrupt handling
Interrupt processing routine
Interrupt waiting time
Interrupt request sampling time
Interrupt handling time (9 instruction cycles)
Interrupt request occurs
◀▶ : Final cycle of instruction. Interrupt requests are sampled at this timing.

The total interrupt processing time of 21 + 9 = 30 instruction cycles is required if an interrupt request occurs immediately after starting execution of a DIVU instruction, which has the longest instruction cycles (21 instruction cycles). If, on the other hand, the program does not use the DIVU or MULU instructions, the maximum interrupt processing time is 6 + 9 = 15 instruction cycles.

The time of one instruction cycle changes with the clock mode and the main clock frequency as selected by the "speed-shift" (gear) function. see Section 3.6 "Clock" for details.

3.4.5 Stack Operation during Interrupt Processing

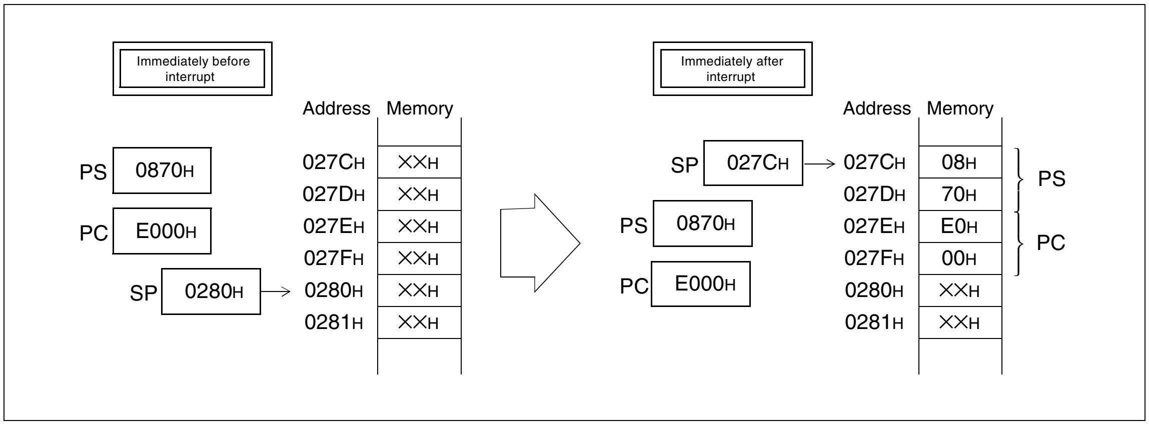
This section describes the saving of the register contents to the stack and restore operation during interrupt processing.

■ Stack Operation at Start of Interrupt Processing

The CPU automatically saves the current contents of the program counter (PC) and program status (PS) to the stack when an interrupt is accepted.

Figure 3.4-5 "Stack Operation at Start of Interrupt Processing" shows the stack operation at the start of interrupt processing.

Figure 3.4-5 Stack Operation at Start of Interrupt Processing



■ Stack Operation at Interrupt Return

On execution of the interrupt return instruction (RETI) at the completion of interrupt processing, the CPU performs the opposite processing to interrupt initiation, restoring first the program status (PS) and then the program counter (PC) from the stack. This returns the PS and PC to their states immediately prior to the start of the interrupt.

Note:

The CPU does not automatically save the accumulator (A) or temporary accumulator (T) contents to the stack. Use the PUSHW and POPW instructions to save and restore A and T contents to and from the stack.

3.4.6 Stack Area for Interrupt Processing

Interrupt processing execution uses the stack area in RAM. The contents of the stack pointer (SP) specifies the top address of the stack area.

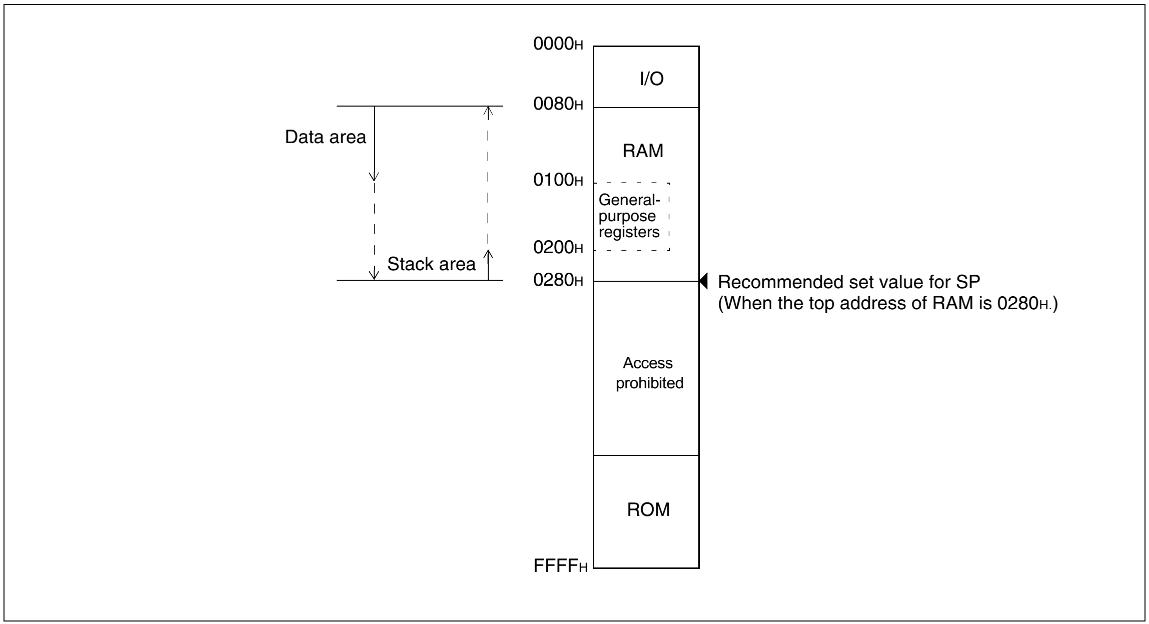
■ Stack Area for Interrupt Processing

The subroutine call instruction (CALL) and vector call instruction (CALLV) use the stack area to save and restore the program counter (PC). The stack area is also used by the PUSHW and POPW instructions to temporarily save and restore registers.

- The stack area is located in RAM along with the data area.
- Initializing the stack pointer (SP) to the top address of RAM and allocating data areas upwards from the bottom RAM address is recommended.

Figure 3.4-6 "Stack Area for Interrupt Processing" shows the example of stack area setting.

Figure 3.4-6 Stack Area for Interrupt Processing


0000H
I/O
0080H
Data area
RAM
0100H
General-purpose registers
0200H
Stack area
0280H
Recommended set value for SP
(When the top address of RAM is 0280H.)
Access prohibited
ROM
FFFFH

Reference:

The stack area is used in the downward direction starting from a high address by functions such as interrupts, subroutine calls, and the PUSHW instruction. Instructions such as return instructions (RETI, RET) and the POPW instruction release stack area in the upward direction. Take care when the stack address is decreased by multiple interrupts or subroutine calls that the stack does not overlap the general-purpose register area or areas containing other data.

3.5 Resets

The resets has the following four types of reset source:

- **External reset**
- **Software reset**
- **Watchdog reset**
- **Power-on reset**

At reset, main clock oscillation stabilization wait time may or may not occur by the operating mode and option settings.

■ Reset Source

Table 3.5-1 Reset Source

Reset source	Reset conditions
External reset	Set the external reset pin to the "L" level.
Software reset	Write "0" to the software reset bit in the standby control register (STBC: RST).
Watchdog reset	Watchdog timer overflow
Power-on reset	Power is turned on (only on products with a power-on reset).

❍ External reset

Inputting an "L" level to the external reset pin ($\overline{\text{RST}}$) generates an external reset. Returning the reset pin to the "H" level wakes up the CPU from the external reset.

The external reset pin can also function as a reset output pin.

❍ Software reset

Writing "0" to the software reset bit in the standby control register (STBC: RST) generates a four-instruction cycle reset.

❍ Watchdog reset

The watchdog reset generates a four-instruction cycle reset if data is not written to the watchdog timer control register (WDTC) within a fixed time after the watchdog timer starts.

❍ Power-on reset

A reset is generated by power-on to initialize the internal circuit.

■ Main Clock Oscillation Stabilization Wait Time and the Reset Source

Whether there will be an oscillation stabilization wait time depends on the operating mode when reset occurs, and the power-on reset option selected.

Following reset, operation always starts out in the normal main clock operating mode, regardless of the kind of reset it was, or the operating mode (the clock mode and standby mode) prior to reset. Therefore, if reset occurs while the main clock oscillator is stopped or in a stabilization wait time, the system will be in a "main clock oscillation stabilization reset" state, and a clock stabilization period will be provided. If the device is set for no power-on reset, however, no main clock oscillation stabilization wait time is provided for power-on or external reset.

In software or watchdog reset, if the reset occurs while the device is in main clock mode, no stabilization time is provided. If it occurs in the subclock mode, however, a stabilization time is provided since the main clock oscillation is stopped.Table 3.5-2 "Reset Source and Oscillation Stabilization Wait Time" shows the relationships between the reset sources and the main clock oscillation stabilization wait time, and reset mode (mode fetch) operations.

Table 3.5-2 Reset Source and Oscillation Stabilization Wait Time

Ret source	Operating state	Reset operation and main clock oscillation stabilization wait time
External reset (*1)	At power on, during stop mode, or subclock mode	After the main clock oscillation stabilization delya time, if the external reset is waked up, reset is operated.(*2)
Software and watchdog reset	Main clock mode	After 4-instruction-cycle reset occurs, reset is operated.(*3)
	Subclock mode	Reset is operated after the main clock oscillation stabilization wait time.(*2)
Power-on reset		Device enters main clock oscillation stabilization wait time at power on. Reset is operated after wait time ends.(*2)

*1: No oscillation stabilization wait time is required for external reset while main clock mode is operating. Reset is operated after external reset is waked up.

*2: If the reset output option is selected, "L" is output at $\overline{\text{RST}}$ pin during the main clock oscillation stabilization wait time.

*3: If the reset output option is selected, "L" level is output at $\overline{\text{RST}}$ pin during 4-instruction-cycle.

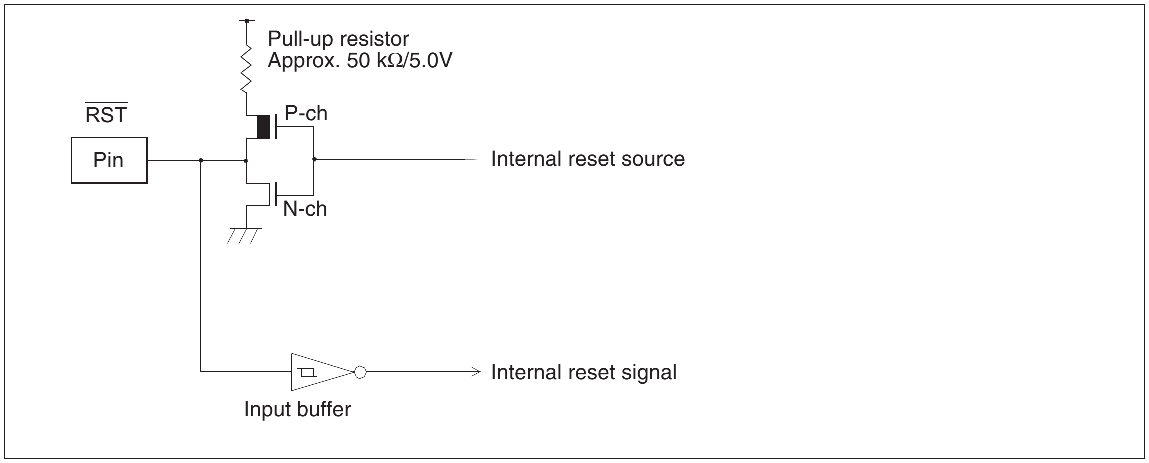
3.5.1 External Reset Pin

Inputting an "L" level to the external reset pin generates a reset. If products are set to with the reset output (optional), the pin outputs an "L" level depending on internal reset sources.

■ Block Diagram of External Reset Pin

The external reset pin ($\overline{\text{RST}}$) is a hysteresis input type and N-ch open-drain output type with a pull-up resistor.

Figure 3.5-1 Block Diagram of External Reset Pin



■ External Reset Pin Functions

Inputting an "L" level to the external reset pin ($\overline{\text{RST}}$) generates an internal reset signal.

On products with the reset output, the pin outputs an "L" level depending on internal reset sources or during the oscillation stabilization wait time due to an external reset. Software reset, watchdog reset, and power-on reset are classed as internal reset sources.

Note:

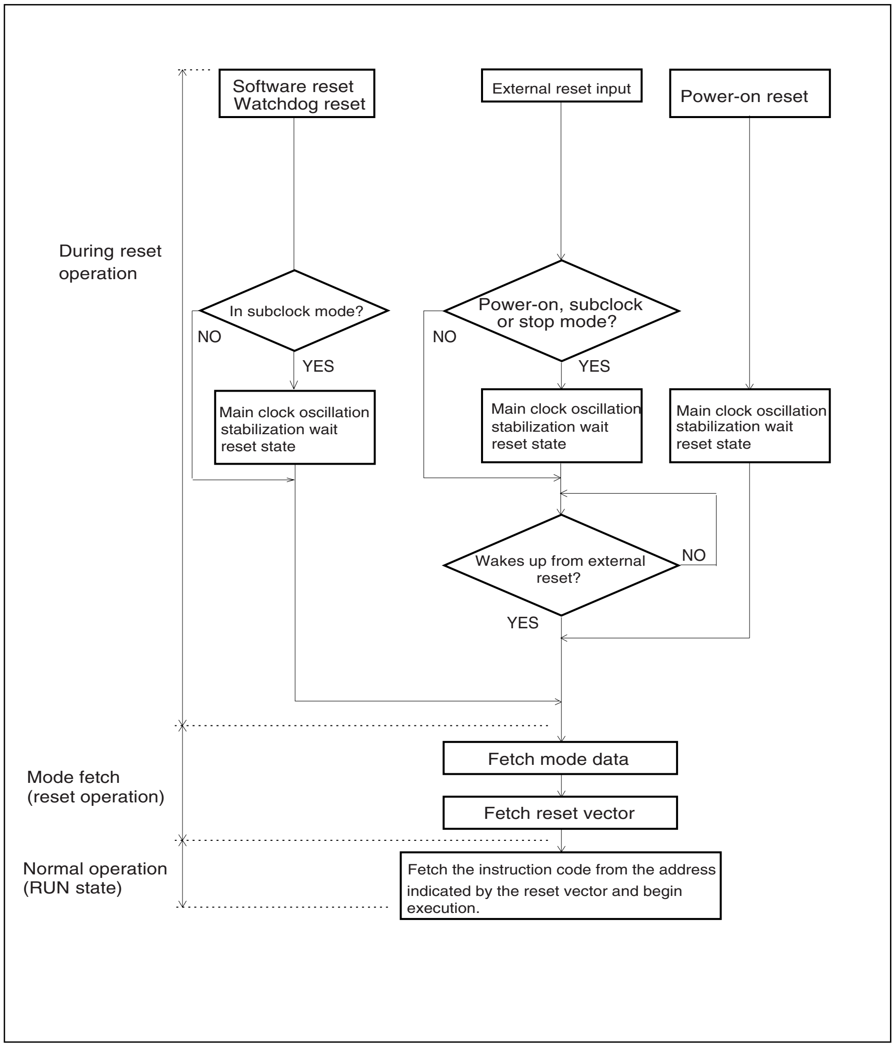
- The external reset input accepts asynchronous with the internal clock. Therefore, initialization of the internal circuit requires a clock. Especially when an external clock is used, a clock is needed to be input at the reset.
- If the reset pulse applied to the external reset pin ($\overline{\text{RST}}$) does not meet the specifications, it may cause malfunctions. Use caution so that the reset pulse less than the specifications will not be fed to the external reset pin ($\overline{\text{RST}}$).

3.5.2 Reset Operation

When the CPU wakes up from a reset, the CPU selects the read address of the mode data and reset vector according to the mode pin settings, then performs a mode fetch. The mode fetch is performed after the oscillation stabilization wait time has passed when power is turned on to a product with power-on reset, or on wake-up from subclock or stop mode by a reset. If reset occurs during a write to RAM, the contents of the RAM address cannot be assured.

■ Overview of Reset Operation

Figure 3.5-2 Reset Operation Flow Diagram


Software reset
Watchdog reset
External reset input
Power-on reset
During reset operation
In subclock mode?
NO
YES
Power-on, subclock or stop mode?
NO
YES
Main clock oscillation stabilization wait reset state
Main clock oscillation stabilization wait reset state
Main clock oscillation stabilization wait reset state
Wakes up from external reset?
NO
YES
Mode fetch (reset operation)
Fetch mode data
Fetch reset vector
Normal operation (RUN state)
Fetch the instruction code from the address indicated by the reset vector and begin execution.

■ Mode Pins

The MB89530/530H/530A series devices are single-chip mode devices. The mode pins (MOD1, MOD2) must be tied to VSS. The mode pin settings determine whether the mode data and reset vector are read from internal ROM.

Do not change the mode pin settings, even after the reset has completed.

■ Mode Fetch

When the CPU wakes up from a reset, the CPU reads the mode data and reset vector from internal ROM.

❍ Mode data (address: FFFD$_H$)

Always set the mode to "00$_H$" (single-chip mode).

❍ Reset vector (address: FFFE$_H$ (upper), FFFF$_H$ (lower))

Contains the address where execution is to start after completion of the reset. The CPU starts executing instructions from the address contained in the reset vector.

■ Oscillation Stabilization Wait Reset State

On products with power-on reset, the reset operation for a power-on reset or external reset in subclock or stop (main/sub) mode starts after the main clock oscillation stabilization wait time selected by the stabilization wait time option. If the CPU has not woken up from the external reset input when the wait time completes, the reset operation does not start until the CPU wakes up from external reset.

As the oscillation stabilization wait time is also required when an external clock is used, a reset requires that the external clock is input.

The main clock oscillation stabilization wait time is timed by the timebase timer.

■ Effect of Reset on RAM Contents

If a reset condition occurs, the operation of the instruction currently being executed is suspended and the device enters the reset status. The contents written in RAM are unchanged before and after a reset. However, if a reset occurs during the writing of 16-bit data, only the high-order byte of the data is written and the low-order byte may not be written. If a reset occurs immediately before or after writing, the contents of the address to which writing is being performed will be unpredictable. After a reset, therefore, all of the RAM to be used must be initialized.

3.5.3 Pin States during Reset

Reset initialized the pin states.

■ Pin States during Reset

When a reset source occurs, with a few exceptions, all I/O pins (peripheral pins) go to the high-impedance state and the mode data is read from internal ROM.

■ Pin States after Reading Mode Data

With a few exceptions, the I/O pins remain in the high-impedance state immediately after reading the mode data (The pin for which "pull-up resistor available" is selected in the pull-up option setting register is set to the "H" level.).

Note:

For devices connected to pins that change to high-impedance state when a reset source occurs take care that malfunction does not occur due to the change in the pin states.

For the pin states under conditions other than the reset state, see Appendix F "Pin Statuses of the MB89530/530H/530A Series".

3.6 Clock

Dual clock oscillation circuits are contained in the clock generator. By connecting each external resonator, the high-speed main clock and low-speed subclock are generated independently (oscillator source). A clock generated externally can also be input.
The speed and supply of the dual clock is controlled by the clock controller according to the clock mode and standby mode.

■ Clock Supply Map

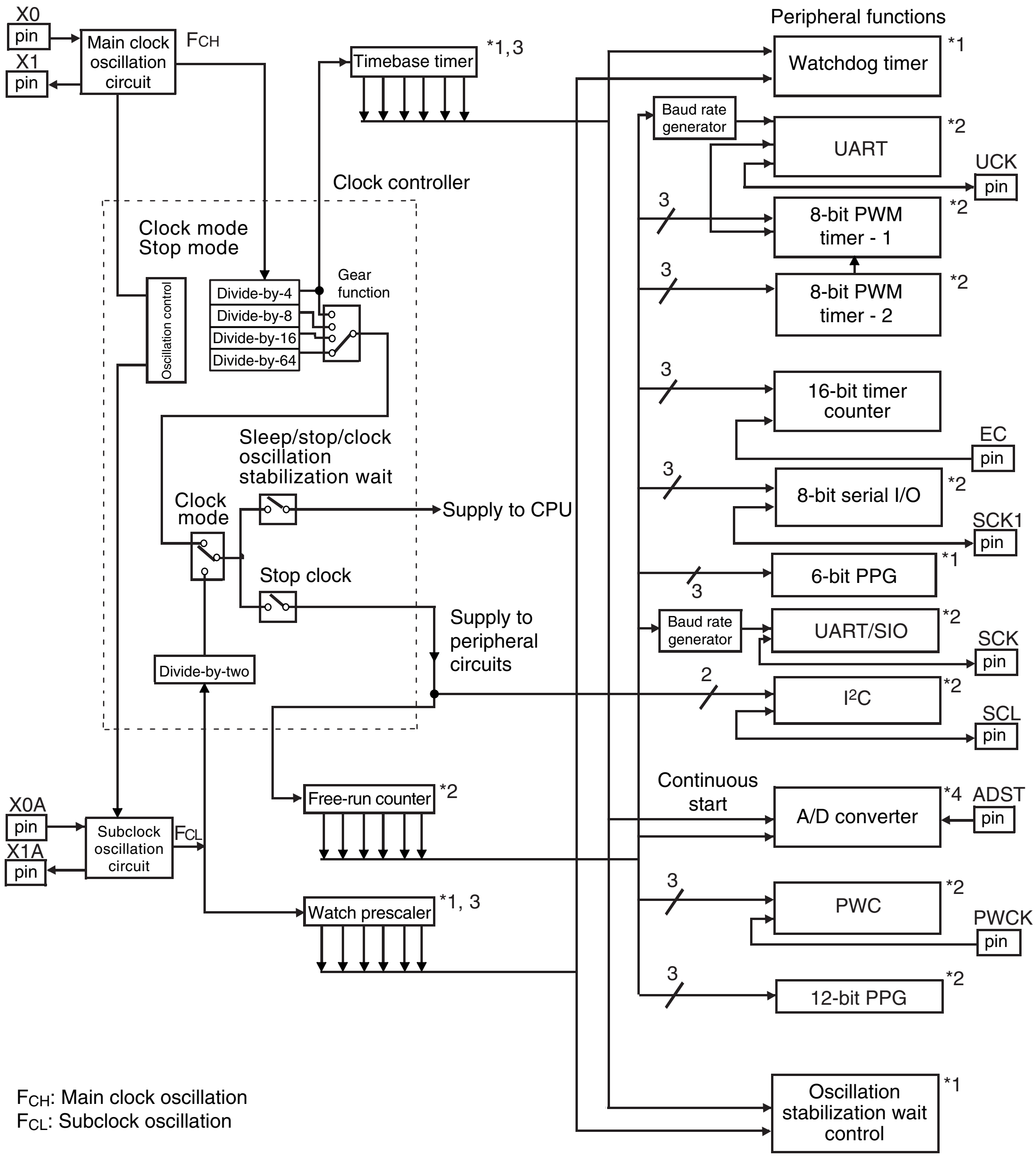
The clock oscillation and the supply to CPU and peripheral circuits (peripheral functions) are controlled by the clock controller. Thus, the operating clock of CPU and peripheral circuits are affected by switching between the main clock and the subclock (clock mode), speed switching of the main clock (gear function), and the standby mode (sleep/stop/watch).

The divide-by output of the free-run counter of the clock for peripheral circuits is supplied to each peripheral function.

The divide-by output of the timebase timer operating at divide-by-two oscillation of the main clock oscillation and peripheral functions to which divide-by output of the watch prescaler operating at the subclock is supplied are available and are not affected by the gear function.

The following Figure 3.6-1 "Clock Supply Map" shows a clock supply map.

Figure 3.6-1 Clock Supply Map



F_{CH}: Main clock oscillation
F_{CL}: Subclock oscillation

*1: Not affected by the clock mode and gear function.
*2: The operating speed is affected by the clock mode and gear function.
*3: Operations stop if the clock (main or sub), which is the source of oscillation, stops.
*4: The timebase timer output can be selected during continuous start of A/D conversion. In other modes, the clock speed is influenced by the clock mode and gear function.

3.6.1 Clock Generator

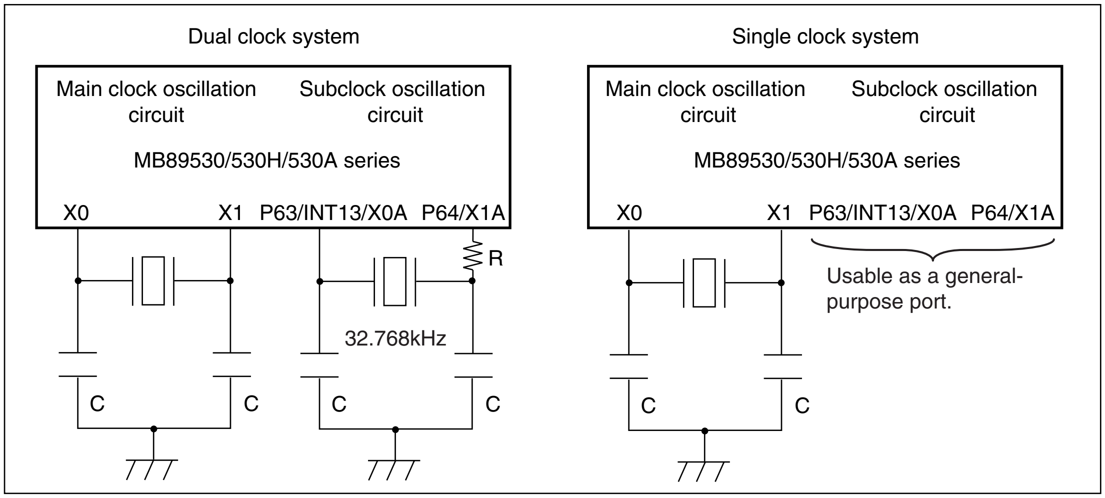
The permission and stop of oscillation of the main clock and subclock are controlled by the clock mode and stop mode.

■ Clock Generator

❍ Crystal resonator or ceramic resonator

Make connections as shown in Figure 3.6-2 "Connection Example of the Crystal and Ceramic Resonator".

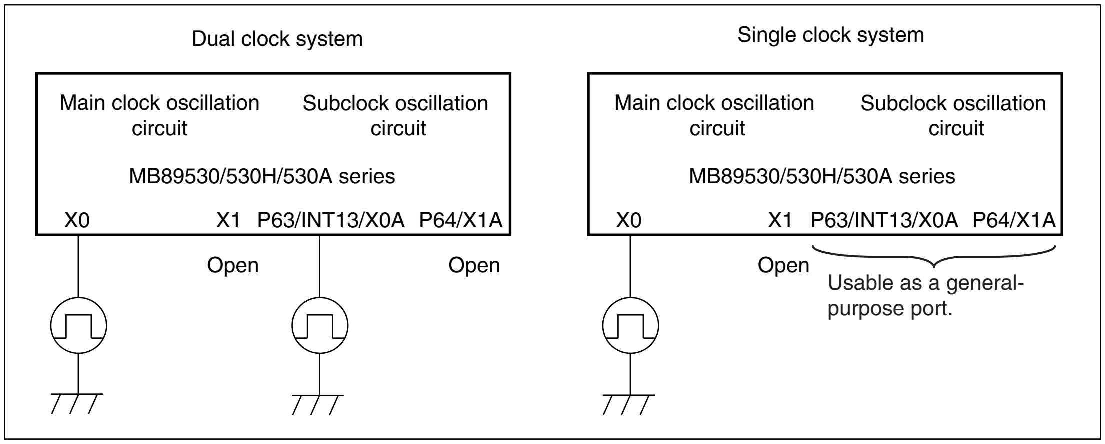
Figure 3.6-2 Connection Example of the Crystal and Ceramic Resonator


Dual clock system
Main clock oscillation circuit
Subclock oscillation circuit
MB89530/530H/530A series
X0
X1
P63/INT13/X0A
P64/X1A
R
32.768kHz
C
C
C
C
Single clock system
Main clock oscillation circuit
Subclock oscillation circuit
MB89530/530H/530A series
X0
X1
P63/INT13/X0A
P64/X1A
Usable as a general-purpose port.
C
C

❍ External clock

Connect the external clock to the X0 pin as shown in Figure 3.6-3 "Connection Example of the External Clock" and open the X1 pin. If the subclock should be supplied externally, connect the external clock to the X0A pin and open the X1A pin.

Figure 3.6-3 Connection Example of the External Clock


Dual clock system
Main clock oscillation circuit
Subclock oscillation circuit
MB89530/530H/530A series
X0
X1
P63/INT13/X0A
P64/X1A
Open
Open
Single clock system
Main clock oscillation circuit
Subclock oscillation circuit
MB89530/530H/530A series
X0
X1
P63/INT13/X0A
P64/X1A
Open
Usable as a general-purpose port.

Note:

The MB89530/530H/530A series can operate with the single clock system. When using only the main clock, a return from subclock mode cannot be performed.

The following table lists the options of the single and dual clock system products:

Table 3.6-1 Part Number Options

Option	Minimum t_{inst}	Clock system
MB89P538-101	4	Single clock system
MB89P538-201	4	Dual clock system

3.6.2 Clock Controller

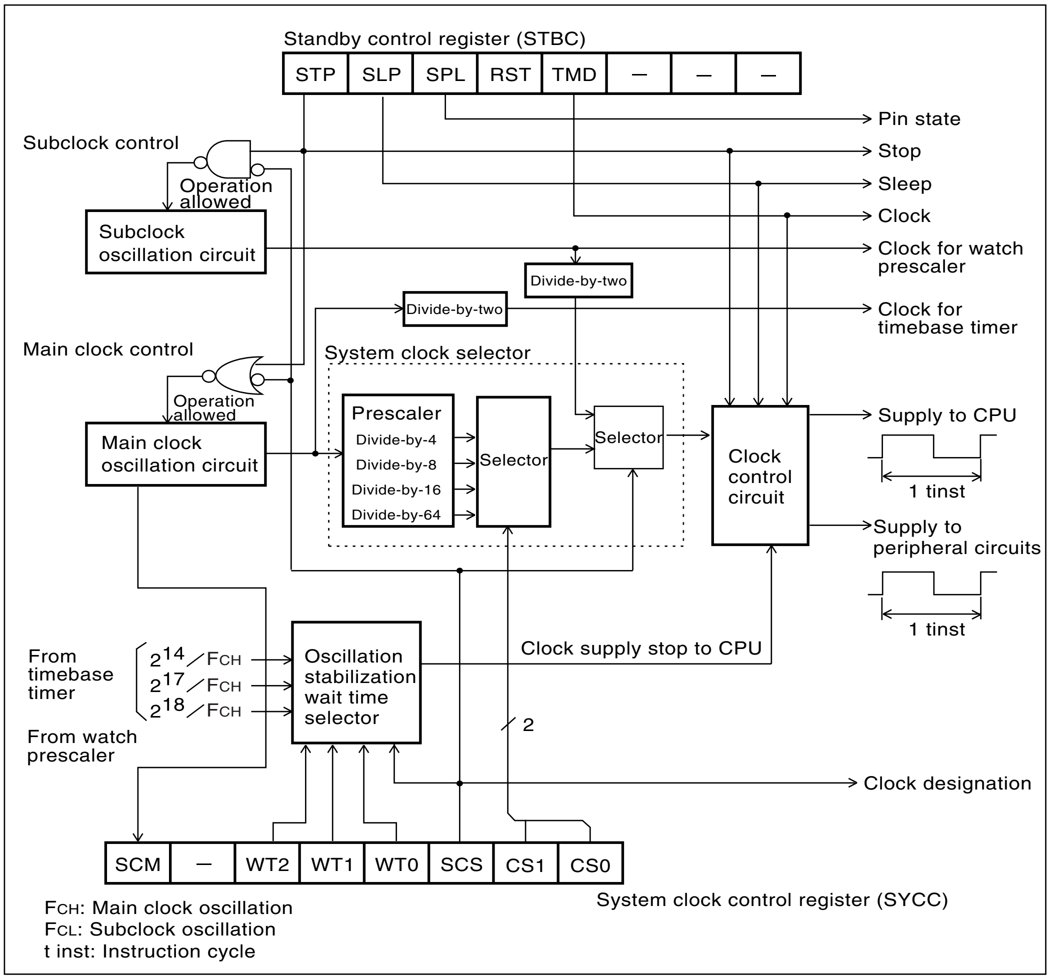
The clock controller is made up of the following seven blocks:

- **Main clock oscillation circuit**
- **Subclock oscillation circuit**
- **System clock selector**
- **Clock control circuit**
- **Oscillation stabilization wait time selector**
- **System clock control register (SYCC)**
- **Standby control register (STBC)**

■ Block Diagram of the Clock Controller

Figure 3.6-4 "Block Diagram of the Clock Controller" shows a block diagram of the clock controller.

Figure 3.6-4 Block Diagram of the Clock Controller



- **Main clock oscillation circuit**

  Oscillation circuit of the main clock. This circuit stops oscillation in main stop mode and subclock mode.

- **Subclock oscillation circuit**

  Oscillation circuit of the subclock. This circuit always oscillates in a mode other than sub-stop mode.

- **System clock selector**

  One type is selected from the four clocks and the subclocks obtained by dividing the oscillation of the main clock to supply it to the clock control circuit.

- **Clock control circuit**

  The operating clock supply to CPU and each peripheral circuit is controlled according to normal operation (RUN) and standby modes (sleep, stop, watch).

  The clock controller stops the supply of clocks to CPU until the clock supply stop signal of the oscillation stabilization wait time selector is released.

- **Oscillation stabilization wait time selector**

  One wait time is selected from the four kinds of oscillation stabilization wait time for the main clock created by the timebase timer and the oscillation stabilization wait time for the subclock created by the watch prescaler for the clock mode, standby mode, and reset and is output as the clock supply stop signal to CPU.

- **System clock control register (SYCC)**

  The selection of the clock mode and main clock speed is performed, and the selection and state confirmation of the oscillation stabilization wait time of the main clock is performed.

- **Standby control register (STBC)**

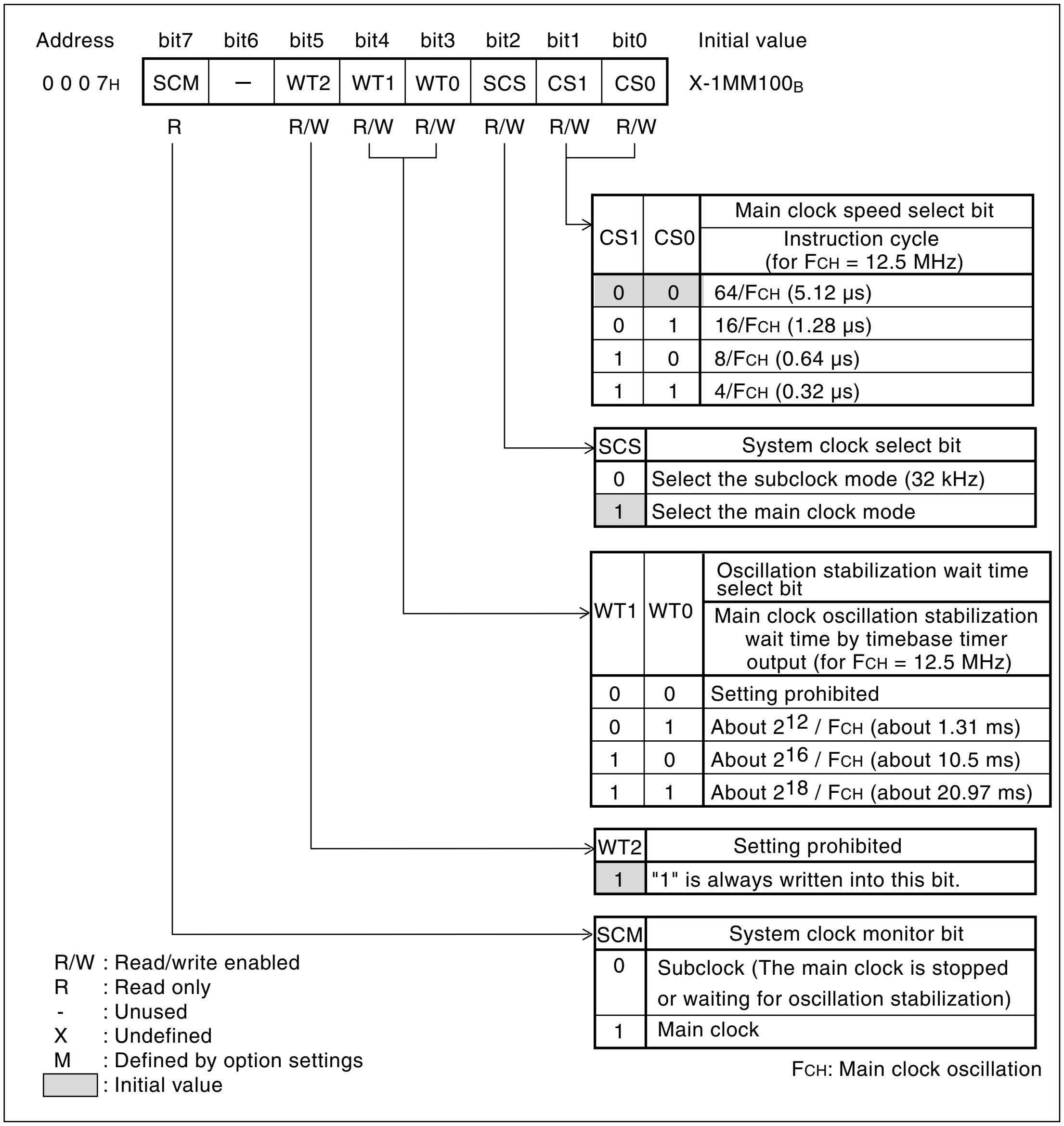
  The transition from normal operation (RUN) to standby mode, pin state settings in stop mode or watch mode, and software reset are performed.

3.6.3 System Clock Control Register (SYCC)

The system clock control register (SYCC) is used to switch the main clock and the subclock, to select the speed of the main clock, and to select the oscillation stabilization wait time.

■ Structure of the System Clock Control Register (SYCC)

Figure 3.6-5 Structure of the System Clock Control Register (SYCC)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 0 7H	SCM	–	WT2	WT1	WT0	SCS	CS1	CS0	X-1MM100B
	R		R/W	R/W	R/W	R/W	R/W	R/W	

CS1	CS0	Main clock speed select bit Instruction cycle (for FCH = 12.5 MHz)
0	0	64/FCH (5.12 µs)
0	1	16/FCH (1.28 µs)
1	0	8/FCH (0.64 µs)
1	1	4/FCH (0.32 µs)

SCS	System clock select bit
0	Select the subclock mode (32 kHz)
1	Select the main clock mode

WT1	WT0	Oscillation stabilization wait time select bit Main clock oscillation stabilization wait time by timebase timer output (for FCH = 12.5 MHz)
0	0	Setting prohibited
0	1	About 2^{12} / FCH (about 1.31 ms)
1	0	About 2^{16} / FCH (about 10.5 ms)
1	1	About 2^{18} / FCH (about 20.97 ms)

WT2	Setting prohibited
1	"1" is always written into this bit.

SCM	System clock monitor bit
0	Subclock (The main clock is stopped or waiting for oscillation stabilization)
1	Main clock

R/W : Read/write enabled
R : Read only
- : Unused
X : Undefined
M : Defined by option settings
▨ : Initial value

FCH: Main clock oscillation

Table 3.6-2 Explanation of the Functions of Each Bit of the System Clock Control Register (SYCC)

Bit name		Function
Bit 7	SCM: System clock monitor bit	• Bit to check the current clock mode (operating clock) • If the bit is "0", the system is operating in subclock mode (The main clock is stopped or waiting for oscillation stabilization to make a transition to the main clock mode). • If the bit is "1", the system is operating in main clock mode. **Reference:** This bit is read-only. Write operation to this bit has no significance and does not affect operations.
Bit 6	Unused bits	• The read value is undefined. • Writing has no effect on operation.
Bit 5	WT2	• "1" is always written into this bit.
Bit 4 Bit 3	WT1, WT0: Oscillation stabilization wait time select bits	• Bits to select the oscillation stabilization wait time of the main clock • The oscillation stabilization wait time selected by these bits is taken when making a transition from the subclock mode to the main clock mode, or returning to normal operation from the main stop mode by an external interrupt. • The initial values of these bits are selected by option settings. Thus, if an oscillation stabilization wait time is taken for a reset, the oscillation stabilization wait time selected by option settings is taken. **Note:** Do not rewrite these bits simultaneously when switching from the subclock to the main clock (SCS=1 --> 0). Before rewriting the bits, check that the oscillation stabilization is not waited upon using the SCM bit.
Bit 2	SCS: System clock select bit	• Bit to specify the clock mode • A transition from the main clock mode to the subclock mode is caused by writing "0" into this bit. • If "1" is written into this bit, the transition from the subclock mode to the main clock mode occurs after taking the oscillation stabilization wait time set by the WT1 and WT0 bits. **Note:** If the single clock system option is selected, this bit has no significance. Set always "1".
Bit 1 Bit 0	CS1, CS0: Main clock speed select bits	• Bit to select the clock speed in main clock mode. • Four different speeds of the operating clock can be selected for CPU and each peripheral function (gear function). However, the operating clock of the timebase timer and watch prescaler is not affected by these bits.

■ Instruction Cycle (t_{inst})

The instruction cycle (minimum execution time) can be selected from the 1/4, 1/8, 1/16, or 1/64 of the main clock and the divide-by-two of the subclock (32.768 kHz) using the system clock select bit (SYCC:SCS) and main clock speed select bits (SYCC:CS1, CS0) of the SYCC register.

The instruction cycle at the maximum speed (SYCC: SCS=1, CS1, CS0=11_B) in main clock mode is $4/F_{CH}$ = about 0.32 µs if the main clock oscillation (F_{CH}) is 12.5 MHz.

The instruction cycle in subclock mode (SCS=0) is $2/F_{CL}$ = about 61.0 µs if the subclock oscillation (F_{CL}) is 32.768 kHz.

3.6.4 Clock Modes

The main clock mode and subclock mode are available as the clock mode.
In main clock mode, the main clock is the main operating clock. The speed of the main clock can be switched by selecting from four kinds of clocks created by dividing its oscillation (gear function).
In subclock mode, the oscillation of the man clock is stopped and the subclock alone becomes the operating clock.

■ Operating State of the Clock Mode

Table 3.6-3 Operating State of the Clock Mode

Clock mode	Main clock speed SYCC register (SYCC: CS1, CS0)		Standby mode	Clock generation		Operation clock in each section				Release source of standby mode (other than resets)
				Main	Sub	CPU	timebase timer	Each peripheral	watch prescaler	
Main clock mode	(1.1)	High speed ↑	RUN	Oscillation	Oscillation	$F_{CH}/4$	$F_{CH}/2$	$F_{CH}/4$	F_{CL}	Various interrupt requests
			Sleep			Stop				
			Stop	Stop			Stop	Stop		External interrupt
	(1.0)		RUN	Oscillation	Oscillation	$F_{CH}/8$	$F_{CH}/2$	$F_{CH}/8$	F_{CL}	Various interrupt requests
			Sleep			Stop				
			Stop	Stop			Stop	Stop		External interrupt
	(0.1)		RUN	Oscillation	Oscillation	$F_{CH}/16$	$F_{CH}/2$	$F_{CH}/16$	F_{CL}	Various interrupt requests
			Sleep			Stop				
			Stop	Stop			Stop	Stop		External interrupt
	(0.0)	↓ Low speed	RUN	Oscillation	Oscillation	$F_{CH}/64$	$F_{CH}/2$	$F_{CH}/64$	F_{CL}	Various interrupt requests
			Sleep			Stop				
			Stop	Stop			Stop	Stop		External interrupt
Subclock mode	-		RUN	Stop	Oscillation	F_{CL}	Stop(*1)	F_{CL}	F_{CL}	Various interrupt requests
			Sleep			Stop				
			Stop		Stop			Stop	Stop	External interrupt
			Watch mode	Stop	Oscillation	Stop	Stop(*1)	Stop	F_{CL}	External interrupt, watch interrupt

F_{CH}: Main clock oscillation
F_{CL}: Subclock oscillation
*1: Since the timebase timer is operated by the main clock, it stops operation in subclock mode.

In each clock mode, a transition can be made to the standby mode corresponding to each mode. For the standby mode, see Section 3.7 "Standby Mode (Low Power Consumption)".

■ Gear Function (Function for Switching the Speed of the Main Clock)

Four different main clock speeds can be selected by writing "00_B" to "11_B" into the main clock speed select bits (SYCC: CS1, CS0) of the system clock control register.

CPU and each peripheral circuit operate at the switched main clock speed. However, the timebase timer and watch prescaler are not affected by the gear function.

By reducing the main clock speed, power consumption can be reduced.

■ Operation in Main Clock Mode

In normal operation in main clock mode (main RUN mode), both the main clock and subclock oscillate. The watch prescaler is operated by the subclock, whereas CPU, the timebase timer, and other peripheral circuits are operated by the main clock.

The speed of the main clock can be switched to a value other than that of the timebase timer during operation in main clock mode (gear function). A transition to the main sleep mode or main stop mode is enabled by specifying the standby mode.

Operation always starts in main RUN mode regardless of the type of reset that occurs (release by the reset in each operation mode).

❍ Transition from the main clock mode to the subclock mode

A transition from the main clock mode to the subclock mode is caused by writing "0" into the system clock select bit (SYCC: SCS) of the system clock control register.

The current operating clock can be checked by reading the system clock monitor bit (SYCC: SCM) of the system clock control register.

Note:

To make a transition to the subclock mode, for example, just after power-on, it is necessary to wait at least the subclock oscillation stabilization wait time created by the watch prescaler using software before making a transition.

■ Operation in Subclock Mode

In normal operation in subclock mode (sub RUN mode), the oscillation of the main clock is stopped and only the subclock is used for operation. By operating in a low speed clock, power consumption can be reduced.

All functions other than the timebase timer operate as in the main clock mode. A transition to the sub-sleep mode, sub-stop mode, or watch mode is enabled by specifying the standby mode during operation in subclock mode.

❍ Return from the subclock mode to the main clock mode

A return from the subclock mode to the main clock mode is caused by writing "1" into the system clock select bit (SYCC: SCS) of the system clock control register.

However, operation in main clock mode starts only after the oscillation stabilization wait time of the main clock passes. The oscillation stabilization wait time can be selected from three different wait times using the oscillation stabilization wait time select bits (SYCC: WT1, WT0) of the system clock control register.

Note:

Do not rewrite the oscillation stabilization wait time select bits (SYCC: WT1, WT0) simultaneously by switching from the subclock to the main clock. Also, do not rewrite the bits when the oscillation stabilization of the main clock is waited upon. In such cases, rewrite the bits after checking that the operating clock has been switched to the main clock (SYCC: SCM=1) by the system clock monitor bit.

After a power-on, if a power-on reset is generated, the oscillation stabilization wait time is allowed to elapse.

For return to the main clock mode from the subclock mode using a reset, the oscillation stabilization wait time is allowed to elapse.

3.6.5 Oscillation Stabilization Wait Time

If the main clock is operated in main RUN mode from a state in which the main clock is stopped, for example, when the power is turned on, or in main stop mode or subclock mode, it is necessary to take the oscillation stabilization wait time of the main clock. Likewise, the oscillation stabilization wait time of the subclock is needed in sub-stop mode because the oscillation of the subclock is stopped.

■ Oscillation Stabilization Wait Time

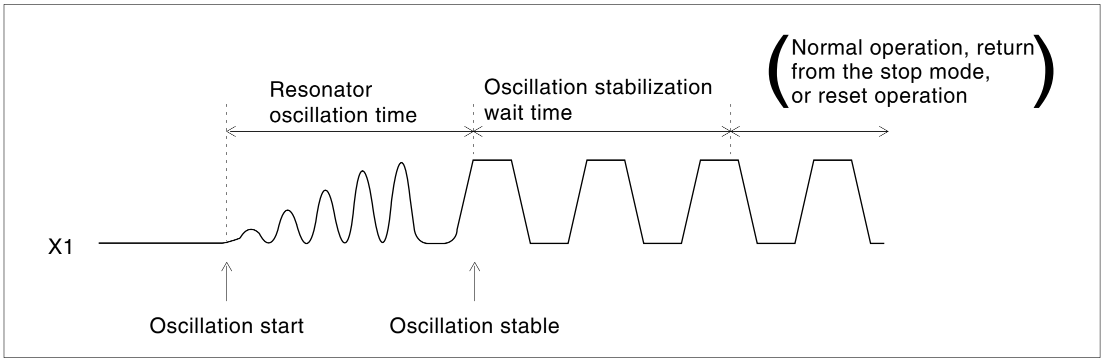
Ceramic and crystal resonators generally take several ms to several dozens of ms to oscillate steadily in natural frequency after starting oscillation.

Thus, CPU operation must be prohibited just after starting oscillation. The clock should be supplied to CPU only when the oscillation is sufficiently stable after the passage of the oscillation stabilization wait time.

Since the time needed to stabilize oscillation is dependent on the type (such as the crystal and ceramic) of resonator connected to the oscillator (clock generator), an oscillation stabilization wait time appropriate to the resonator to be used must be selected.

Figure 3.6-6 "Oscillator Operation after Oscillation Starts" shows an oscillator operation just after the oscillation starts.

Figure 3.6-6 Oscillator Operation after Oscillation Starts


Resonator oscillation time
Oscillation stabilization wait time
(Normal operation, return from the stop mode, or reset operation)
X1
Oscillation start
Oscillation stable

■ Oscillation Stabilization Wait Time of the Main Clock

To start operation in main clock mode from a state in which the main clock is stopped, the oscillation stabilization wait time of the main clock must be taken.

The oscillation stabilization wait time of the main clock is a time interval counted from when the counter of the timebase timer is cleared until the overflow of the specified bit occurs.

❍ Oscillation stabilization wait time during operation

One of the four kinds of oscillation stabilization wait time when returning to the main RUN mode from the main stop mode by an external reset or when making a transition from the subclock mode to the main clock mode can be selected using the oscillation stabilization wait time select bits (SYCC: WT1, WT0) of the system clock control register.

❍ Oscillation stabilization wait time during reset

The oscillation stabilization wait time during reset (SYCC: initial value of WT1 and WT0) can be selected by option settings.

The oscillation stabilization wait time is allowed to elapse when the stop mode is canceled by two or more resets in subclock mode, a power-on reset, or external reset.

Table 3.6-4 "Operation Start Conditions and Oscillation Stabilization Wait Time of the Main Clock Mode" lists the relations between the operation start conditions and oscillation stabilization wait time of the main clock mode.

Table 3.6-4 Operation Start Conditions and Oscillation Stabilization Wait Time of the Main Clock Mode

Start conditions for main clock operation	During power-on	During subclock mode		Release from main stop mode		Transition from the subclock mode to the main clock mode (SYCC: SCS (*1) =1)
		External reset	Software reset and watchdog timer	External reset	External interrupt	
Oscillation stabilization wait time selection	Option settings				SYCC:WT1, WT0 (*2)	

*1: System clock select bit of the system clock control register
*2: Oscillation stabilization wait time select bit of the system clock control register

■ Oscillation Stabilization Wait Time of the Subclock

A certain oscillation stabilization wait time ($2^{15}/F_{CL}$, F_{CL}: subclock oscillation) of the subclock must be taken when returning to the sub RUN mode (subclock oscillation is started) from the sub-stop mode (state in which oscillation of the subclock is stopped) by an external reset.

The oscillation stabilization wait time of the subclock is a time interval from the start of operation in a state in which the watch prescaler is cleared until an overflow occurs.

Since the oscillation stabilization wait time of the subclock is needed during power-on, wait at least the subclock oscillation stabilization wait time using software to make a transition to the subclock mode after power-on.

3.7 Standby Mode (Low Power Consumption)

The sleep mode, stop mode, and watch mode are available as the standby mode. A transition to the standby mode is caused by settings of the standby control register (STBC) both in main clock mode and subclock mode.
In main clock mode, transitions to the sleep mode and stop mode are possible. In subclock mode, transitions to the sleep mode, stop mode, and watch mode are possible. Power consumption can be reduced by stopping operations of CPU and peripheral functions using the standby mode. This section describes the relations between the standby mode and clock mode, and the operating states of each section in standby mode.

■ Standby Mode

In clock mode, power consumption is reduced by reducing the operating clock of CPU and peripheral circuits such as switching of the main clock and subclock or switching of the main clock speed (gear function). In standby mode, however, power consumption is reduced by the clock supply stop (sleep mode) to CPU by the clock controller, clock supply stop (watch mode) to CPU and peripheral circuits, or stop of the oscillation itself (stop mode).

❍ Main sleep mode

The main sleep mode is a mode which stops operations of CPU and the watchdog timer. Peripheral functions excluding the watch prescaler operate on the main clock (Part of the functions can operate on the subclock).

❍ Sub-sleep mode

The sub-sleep mode is a mode which stops the main clock oscillation, CPU operations, and watchdog timer and timebase timer operations. Peripheral functions operate on the subclock.

❍ Main stop mode

The main stop mode is a mode which stops operations of CPU and peripheral functions. The main clock stops oscillation, but the subclock continues oscillation. In this mode, all functions are stopped excluding external interrupts, count operations of the watch prescaler, and some of the functions that run with the subclock.

❍ Sub-stop mode

The sub-stop mode is a mode which stops all functions other than external interrupts. The main clock and the subclock both stop oscillation.

❍ Watch mode

The watch mode is a mode a transition to which is possible only from the subclock mode. All functions other than the watch prescaler (watch interrupt), external interrupts, and part of the functions operating on the subclock stop.

3.7.1 Operating State in Standby Mode

This section describes the operating states of CPU and peripheral functions in standby mode.

■ Operating State in Standby Mode

Table 3.7-1 The Operating States of CPU and Peripheral Functions in Standby Mode

Function		Main clock mode			Subclock mode			
		RUN	Sleep	Stop	RUN	Sleep	Stop	Watch
Main clock		Operating	Operating	Stopped	Stopped	Stopped	Stopped	Stopped
Subclock		Operating	Operating	Operating	Operating	Operating	Stopped	Operating
CPU	Instruction	Operating	Stopped	Stopped	Operating	Stopped	Stopped	Stopped
	ROM	Operating	Retained	Retained	Operating	Retained	Retained	Retained
	RAM							
Peripheral functions	I/O port	Operating	Retained	Retained	Operating	Retained	Retained	Retained
	Watch prescaler	Operating	Operating	Operating(*1)	Operating	Operating	Retained	Operating
	Timebase timer	Operating	Operating	Stopped	Stopped	Stopped	Stopped	Stopped
	16-bit timer/counter	Operating	Operating	Stopped	Operating	Operating	Stopped	Stopped
	8-bit serial I/O	Operating	Operating	Stopped	Operating	Operating	Stopped	Stopped
	UART	Operating	Operating	Stopped	Operating	Operating	Stopped	Stopped
	I^{2}C bus interface	Operating	Operating	Stopped	Operating	Operating	Stopped	Stopped
	UART/SIO	Operating	Operating	Stopped	Operating	Operating	Stopped	Stopped
	8-bit PWM timer	Operating	Operating	Stopped	Operating	Operating	Stopped	Stopped
	A/D converter	Operating	Operating	Stopped	Operating	Operating	Stopped	Stopped
	External interrupt 1, 2	Operating	Operating	Operating	Operating	Operating	Operating	Operating
	12-bit PPG	Operating	Operating	Operating(*2)	Operating(*2)	Operating(*2)	Stopped	Operating(*2)
	Watchdog timer	Operating	Stopped	Stopped	Operating(*2)	Stopped	Stopped	Stopped
	PWC timer	Operating	Operating	Stopped	Operating	Operating	Stopped	Stopped
	6-bit PPG	Operating	Operating	Operating(*2)	Operating(*2)	Operating(*2)	Stopped	Stopped

*1: The watch prescaler carries out the count operation, but no watch interrupt occurs.
*2: This function can operate when the watch prescaler output is selected as the operating clock.

❍ Pin state in standby mode

Most I/O pins can, independent of the clock mode, retain the state just before transition to the stop or watch mode or can be put into high impedance using the pin state designate bit (STBC: SPL) of the standby control register.

For details on pin states in standby mode, see Appendix F "Pin Statuses of the MB89530/530H/530A Series "

3.7.2 Sleep Mode

This section describes the operations in sleep mode.

■ Operations in Sleep Mode

❍ **Transition to the sleep mode**

The sleep mode is a mode which stops the operating clock of CPU. CPU stops by retaining the contents of the registers and RAM just before transition to the sleep mode, but peripheral functions other than the watchdog timer continue their operations.

However, since the main clock oscillation stops in subclock mode, the timebase timer which uses the divide-by-two of the main clock oscillation as its count clock does not operate.

A transition to the sleep mode is caused by writing "1" into the sleep bit (STBC: SLP) of the standby control register. If an interrupt has occurred when "1" is written into the SLP bit, the write operation is ignored and execution of instructions continues without transition to the sleep mode (no transition to sleep mode after the interrupt).

❍ **Sleep mode release**

The sleep mode is released by a reset or an interrupt from the peripheral functions.

If a reset occurs in sub-sleep mode, the reset operation is performed after taking the oscillation stabilization wait time of the main clock.

Pin states are initialized by the reset operation.

If an interrupt request whose interrupt level is higher than "11" comes from a peripheral function or external interrupt circuit, the sleep mode is released regardless of the interrupt enable flag (CCR: I) or interrupt level bit (CCR: IL1, 0) of CPU.

After the release, normal interrupt operations are performed. If an interrupt can be accepted, interrupt processing is performed. If no interrupt can be accepted, processing starts with the instruction following the instruction executed just before the transition to sleep mode.

3.7.3 Stop Mode

This section describes the operations in stop mode.

■ Operations in Stop Mode

❍ Transition to the stop mode

The stop mode is a mode which stops the oscillation. Contents of the registers and RAM just before transition to the stop mode are retained and most functions are stopped.

In main clock mode, the main clock stops the oscillation, but the subclock continues the oscillation. Thus, though the count operation of the watch prescaler and part of the functions operating on the subclock continue their operations, other peripheral functions and CPU, excluding the external interrupt circuits, stop operations.

In subclock mode, both the main clock and subclock stop oscillation, and all functions other than the external interrupt circuits stop their functions. Thus, data can be retained with minimum power consumption.

A transition to the stop mode is caused by writing "1" into the stop bit (STBC: STP) of the standby control register. At this time, if the pin state designate bit (STBC: SPL) is "0", the states of external pins are retained. If the bit is "1", external pins are put into high impedance.

If an interrupt request has occurred when "1" is written into the STP bit, the write operation is ignored and execution of instructions continues without making a transition to the stop mode (No transition to the stop mode occurs even after interrupt processing is completed).

To make a transition to the stop mode in the main clock mode, prohibit (TBTC: TBIE=0) the interrupt request output of the timebase timer if necessary. Likewise, to make a transition to the stop mode in subclock mode, prohibit (WPCR: WIE=0) the watch interrupt request output of the watch prescaler.

❍ Stop mode release

The stop mode can be released by a reset or external interrupt.

If a reset occurs in stop mode, the reset operation is performed after taking the oscillation stabilization wait time of the main clock.

Pin states are initialized by the reset.

If an interrupt request whose interrupt level is higher than "11" comes from an external interrupt circuit in stop mode, the stop mode is released regardless of the interrupt enable flag (CCR: I) and interrupt level bits (CCR: IL1, 0) of CPU. Since peripheral functions are stopped in stop mode, no interrupt requests other than external interrupts occur. Though the watch prescaler operates in main stop mode, no watch interrupts occur.

If the stop mode is released, a normal interrupt operation is performed following the passage of the oscillation stabilization wait time. If the interrupt is accepted, interrupt processing is performed. If the interrupt is not accepted, execution starts with the instruction following the instruction executed just before transition to the stop mode.

If the stop mode is released by an external interrupt, part of the peripheral functions restarts halfway through their operations. Thus, for example, the first interval time of the interval timer function is undefined. Each peripheral function should be initialized after returning from the stop mode.

Note:

The stop mode release by an interrupt can only be caused by an interrupt request of the external interrupt circuits.

3.7.4 Watch Mode

This section describes the operations in watch mode.

■ Operations in watch mode

❍ Transition to the watch mode

The watch mode is a mode which stops the operating clock of CPU and main peripheral circuits. A transition to the watch mode is only possible from the subclock mode (The main clock oscillation is stopped).

Contents of the registers and RAM just before transition to the watch mode are retained and most functions other than the watch prescaler (watch interrupt), external interrupt circuits, and part of the functions operating on the subclock are stopped. Thus, data can be retained with very low power consumption.

A transition to the watch mode is caused by writing "1" into the watch bit (STBC: TMD) of the standby control register when the subclock mode is set by the system clock select bit of the system clock control register.

If the pin state designate bit (STBC: SPL) of the standby control register during transition to the watch mode is "0", the external pin states are retained. If the bit is "1", external pins are put into high impedance.

If an interrupt request has occurred when "1" is written into the TMD bit, the write operation is ignored and execution of instructions continues without making a transition to the watch mode (No transition to the watch mode occurs even after interrupt processing is completed).

❍ Watch mode release

The watch mode can be released by a reset, watch interrupt, or external interrupt.

If a reset occurs in watch mode, the reset operation is performed after taking the oscillation stabilization wait time of the main clock.

Pin states are initialized by the reset.

If an interrupt request whose interrupt level is higher than "11" comes from the watch prescaler or an external interrupt circuit in watch mode, the watch mode is released regardless of the interrupt enable flag (CCR: I) and interrupt level bits (CCR: IL1, 0) of CPU. Since most peripheral functions other than the watch prescaler are stopped in watch mode, no interrupt requests other than watch interrupts and external interrupts occur.

After releasing the watch mode, a normal interrupt operation is performed. If the interrupt is accepted, interrupt processing is performed. If the interrupt is not accepted, execution starts with the instruction following the instruction executed just before transition to the watch mode.

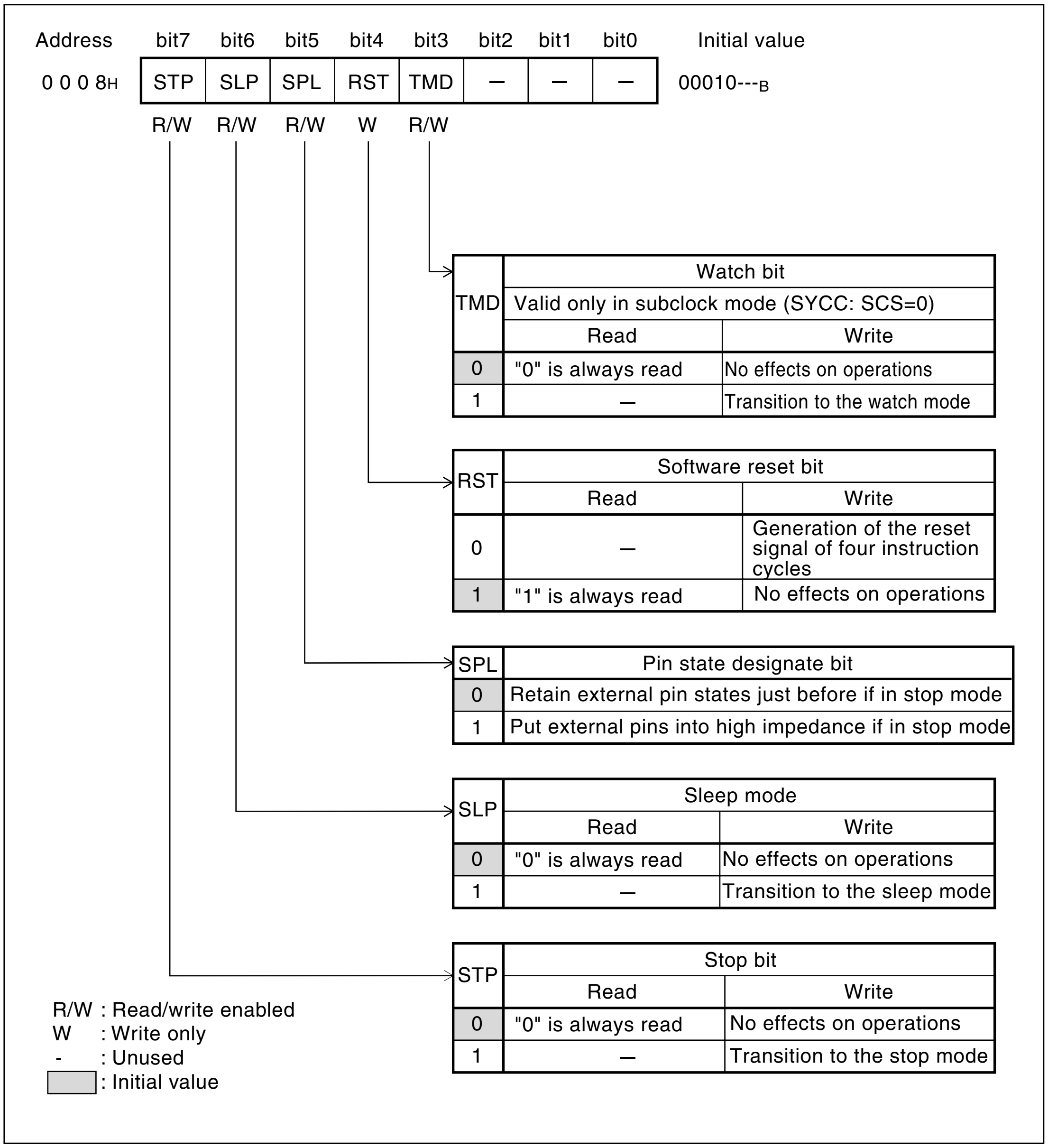
If the watch mode is released, part of the peripheral functions restarts halfway through their operations. Thus, for example, the first interval time of the interval timer function is undefined. Each peripheral function should be initialized after returning from the watch mode.

3.7.5 Standby Control Register (STBC)

The standby control register (STBC) is used to make a transition to the sleep mode/ stop mode/watch mode, set the pin states in watch mode, and reset software.

■ Standby Control Register (STBC)

Figure 3.7-1 Standby Control Register (STBC)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 0 8H	STP	SLP	SPL	RST	TMD	—	—	—	00010---B
	R/W	R/W	R/W	W	R/W				

TMD	Watch bit	
	Valid only in subclock mode (SYCC: SCS=0)	
	Read	Write
0	"0" is always read	No effects on operations
1	—	Transition to the watch mode

RST	Software reset bit	
	Read	Write
0	—	Generation of the reset signal of four instruction cycles
1	"1" is always read	No effects on operations

SPL	Pin state designate bit
0	Retain external pin states just before if in stop mode
1	Put external pins into high impedance if in stop mode

SLP	Sleep mode	
	Read	Write
0	"0" is always read	No effects on operations
1	—	Transition to the sleep mode

STP	Stop bit	
	Read	Write
0	"0" is always read	No effects on operations
1	—	Transition to the stop mode

R/W : Read/write enabled
W : Write only
- : Unused
▨ : Initial value

Table 3.7-2 Explanation of the Functions of Each Bit of the Standby Control Register (STBC)

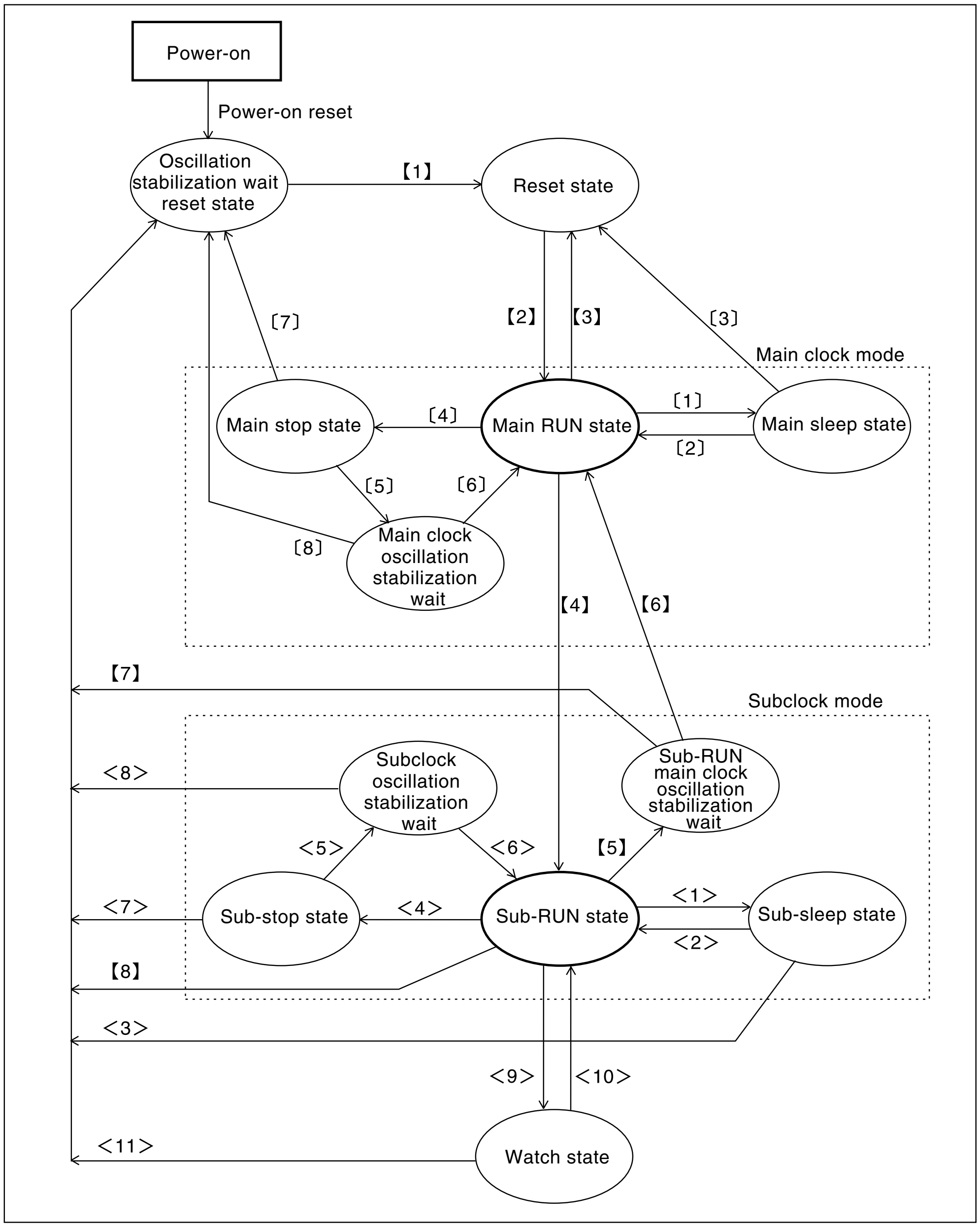
Bit name		Function
Bit 7	STP: Stop bit	• Bit to specify the transition to the stop mode • A transition to the stop mode is caused by writing "1" into this bit. • Operations are not affected if "0" is written into this bit. • When this bit is read, "0" is always read.
Bit 6	SLP: Sleep bit	• Bit to specify the transition to the sleep mode • A transition to the sleep mode is caused by writing "1" into this bit. • Operations are not affected if "0" is written into this bit. • When this bit is read, "0" is always read.
Bit 5	SPL: Pin state designate bit	• Bit to specify the external pin state in stop mode and watch mode • If "0" is written into this bit, the external pin state (level) when making a transition to the stop or watch mode is retained. • If "1" is written into this bit, the external pins are put into high impedance when making a transition to the stop or watch mode (The pin for which "pull-up resistor available" is selected in the pull-up setting register is set to the "H" level.). • This bit is set to "0" by a reset.
Bit 4	RST: Software reset bit	• Bit to specify the software reset • An internal reset source in four instruction cycles caused by writing "0" into this bit. • Operations are not affected if "1" is written into this bit. • When this bit is read, "1" is always read. **Reference:** If the software reset is triggered in subclock mode, operation starts in main clock mode after taking the oscillation stabilization wait time. Thus, the reset signal is output during oscillation stabilization wait time.
Bit 3	TMD: Watch bit	• Bit to specify the transition to the watch mode • Write operation to this bit is valid only in subclock mode (SYCC: SCS=0). • A transition to the watch mode is caused by writing "1" into this bit. • Operations are not affected if "0" is written into this bit. • When this bit is read, "0" is always read.
Bit 2 Bit 1 Bit 0	Unused bits	• The read value is undefined. • Writing has no effect on operation.

3.7.6 State Transition Diagram 1 (Power-On Reset and Dual Clock System)

This section shows a state transition diagram when a power-on reset and the dual clock system are used.

■ State Transition Diagram 1 (Power-On Reset and Dual Clock System)

Figure 3.7-2 State Transition Diagram 1 (Dual Clock)



❍ Transition and release of the clock mode (non-standby mode)

Table 3.7-3 Transition and Release of the Clock Mode (Power-on Reset is Available, Dual Clock System)

State transition	Transition conditions
Transition to the normal state (main RUN) in main clock mode after power-on reset	[1] Main clock oscillation stabilization wait time end (timebase timer output) [2] Release of reset input
Reset in the main RUN state	[3] External reset, software reset, or watchdog reset
Transition from the main RUN state to the sub-RUN state	[4] SYCC:SCS=0(*1)
Return from the sub-RUN state to the main RUN state	[5] SYCC:SCS=1 [6] Main clock oscillation stabilization wait time end (SYCC: can be checked by SCM) [7] External reset, software reset, or watchdog reset
Reset in the sub-RUN state	[8] External reset, software reset, or watchdog reset

SYCC: System clock control register

*1: A transition to the sub-RUN just after power-on occurs after the passage of the subclock oscillation stabilization wait time.

❍ Transition and release of the standby mode

Table 3.7-4 Transition and Release of the Standby Mode (Power-on Reset is Available, Dual Clock System)

State transition	Transition conditions	
	Main clock mode	Subclock mode
Transition to the sleep mode	[1] STBC:SLP=1	<1> STBC:SLP=1
Release of the sleep mode	[2] Interrupt (various types) [3] External reset	<2> Interrupt (various types) <3> External reset
Transition to the stop mode	[4] STBC:STP=1	<4> STBC:STP=1
Release of the stop mode	[5] External interrupt [6] Main clock oscillation stabilization wait time end (timebase timer output) [7] External reset [8] External reset (Waiting for oscillation stabilization)	<5> External interrupt <6> Subclock oscillation stabilization wait time end (watch prescaler output) <7> External reset <8> External reset (Waiting for oscillation stabilization)
Transition to the watch mode	-	<9> STBC:TMD=1(*1)
Release of the watch mode	-	<10> External interrupt or watch interrupt <11> External reset

STBC: Standby control register
*1: A transition to the watch mode is only possible from the sub-RUN state (SYCC: SCS=0).

Note:

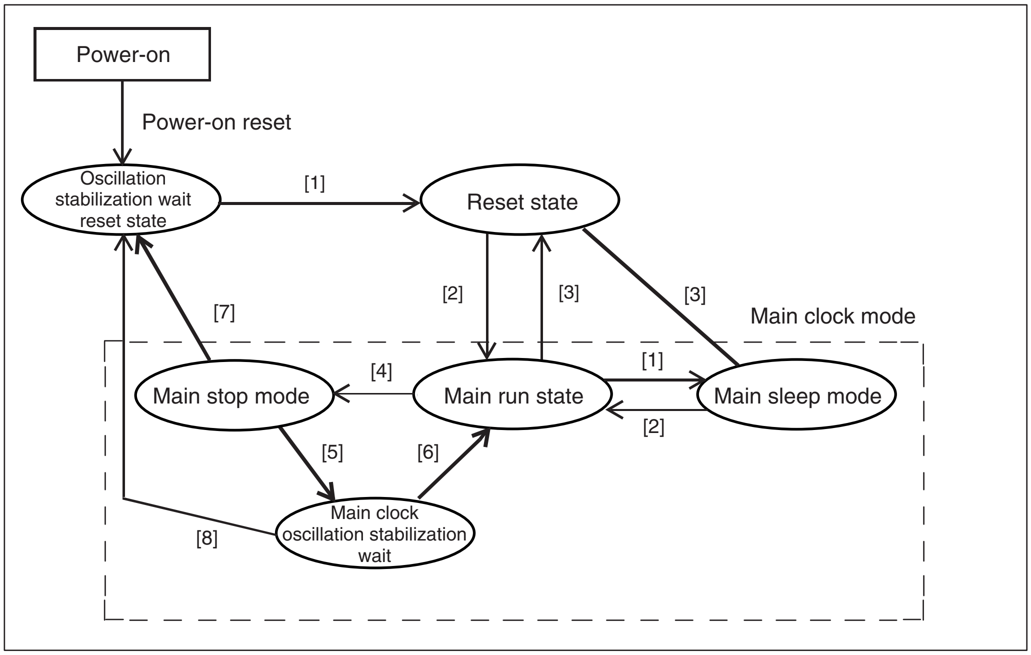
Since CPU and the watchdog timer are stopped in standby mode, no software reset and watchdog reset occur. When a single clock system is used, a transition to the subclock mode is prohibited. If a transition to the subclock mode occurs, CPU stops and there is no other way to return other than resetting.

3.7.7 State Transition Diagram 2 (Single Clock System Option)

This section shows two state transition diagrams for the single clock system option having a power-on reset. If a single clock system is used, the subclock mode and watch mode are not available.

■ State Transition Diagram 2 (Single Clock System Option)

Figure 3.7-3 State Transition Diagram 2



■ Transition to Ordinary State (run) and Release

Table 3.7-5 Transition to Main Clock Mode Run State and Release (Single Clock System Option)

State transition	Transition condition
	Product having a power-on reset (Figure 3.7-3 "State Transition Diagram 2")
Transition to ordinary state (run) after power-on	End of main clock oscillation stabilization wait time (Timebase timer output) Release of the reset input
Reset in the run state	[3] External reset, software reset, and watchdog reset

■ Transition to Standby Mode and Release

Table 3.7-6 Transition to Standby Mode and Release (Single Clock System Option)

State transition	Transition condition
	Product having a power-on reset (Figure 3.7-3 "State Transition Diagram 2")
Transition to the sleep mode	[1] STBC:SLP=1
Release of the sleep mode	Interrupt External reset
Transition to the stop mode	[4] STBC:STP=1
Release of the stop mode	External interrupt End of the main clock oscillation stabilization wait time (Timebase timer output) External reset External reset (during oscillation stabilization wait)

STBC: Standby control register

3.7.8 Notes on Using Standby Mode

A transition to the standby mode does not occur even if the standby mode is set on the standby control register (STBC) when an interrupt request has arrived from a peripheral function. When returning to a normal operation state from the standby mode caused by an interrupt, operations after the return are dependent on whether or not the interrupt request is accepted.

■ Transition to the Standby Mode and Interrupts

When an interrupt request whose interrupt priority is higher than "11_B" arrives at CPU from a peripheral function, a transition to the standby mode does not occur if "1" is written into the stop bit (STBC: STP), sleep bit (STBC: SLP) or watch bit (STBC: TMD) of the standby control register because such write operations are ignored (No transition to the standby mode occurs even after the interrupt processing is completed).

This is not related to whether the interrupt is accepted by CPU.

Even if CPU is processing an interrupt, a transition to the standby mode is possible if the interrupt request flag bit is cleared and there are no other interrupt requests.

■ Release of the Standby Mode by an Interrupt

The standby mode is released if an interrupt request whose interrupt priority is higher than "11" comes from the peripheral functions in sleep mode or stop mode. This is not related to whether the interrupt is accepted by CPU.

After releasing the standby mode, if the priority of the interrupt level setting register (ILR1- ILR4) corresponding to the interrupt request is higher than the interrupt level bits (CCR: IL1, IL0) of the condition code register and if the interrupt enable flag allows the interrupts (CCR: I=1), branching to an interrupt processing routine occurs. If the interrupt is not accepted, execution of the instruction following the instruction starting the standby mode is restarted.

If no branching to an interrupt processing routine just after returning occurs, measures such as an interrupt prohibition are needed before setting the standby mode.

■ Precaution in Setting the Standby Mode

To set the standby mode using the standby control register (STBC), follow Table 3.7-7 "Low Power Consumption Settings by the Standby Control Register (STBC)". The priority order when "1" is written into these bits is the stop mode, watch mode, and sleep mode. However, it is preferable to set "1" to one bit at a time.

Do not make a transition to the stop mode, sleep mode, and watch mode just after switching from the subclock mode to the main clock mode (SYCC: SCS=0 --> 1). Make a transition to these modes after checking that the clock monitor bit (SYCC: SCM) of the system control register is "1".

However, the content written into the watch bit (STBC: TMD) is ignored during operation in main clock mode.

Table 3.7-7 Low Power Consumption Settings by the Standby Control Register (STBC)

STBC register			Mode
STP (bit 7)	SLP (bit 6)	TMD (bit 3)	
0	0	0	Normal
0	0	1	Watch
0	1	0	Sleep
1	0	0	Stop

■ Oscillation Stabilization Wait Time

Since the oscillator for oscillation is stopped in stop mode for both the main clock mode and subclock mode, it is necessary to take the oscillation stabilization wait time after the oscillator in each mode starts operation.

As the oscillation stabilization wait time in main clock mode, take the oscillation stabilization wait time of the main clock created by the timebase timer (Select one from three different kinds of wait time). As the oscillation stabilization wait time in subclock mode, take the oscillation stabilization wait time of the subclock created by the watch prescaler.

In main clock mode, if the selected interval time of the timebase timer is shorter than the oscillation stabilization wait time, an interval timer interrupt request may occur during oscillation stabilization wait time. Before making a transition to the stop mode in main clock mode, prohibit (TBTC: TBIE=0) the interrupt request output of the timebase timer if necessary.

Likewise, a watch interrupt request may occur depending on the selected interval time of the watch prescaler. Before making a transition to the stop mode in subclock mode, prohibit (WPCR: WIE=0) the watch interrupt request output of the watch prescaler if necessary.

3.8 Memory Access Mode

The operation mode for memory access of the MB89530/530H series is the single chip mode only.

■ Single Chip Mode

The single chip mode uses only the internal RAM and ROM. Thus, CPU can access only the internal I/O area, RAM area, and ROM area (internal access).

■ Mode Pin (MOD0, 1)

Set always "Vss" to the mode pin (MOD0 and MOD1).

The mode data and reset vector are read from the internal ROM during reset.

Do not change the settings of the mode pin after the reset operation (during operation) is completed.

Table 3.8-1 "Settings of the Mode Pin" lists the settings of the mode pin.

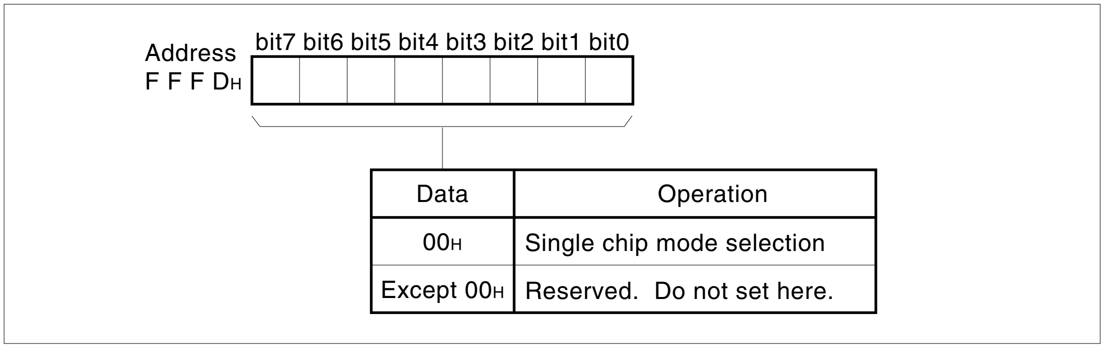
Table 3.8-1 Settings of the Mode Pin

Pin state		Contents
MOD1	MOD0	
Vss	Vss	The mode data and reset vector are read from the internal ROM
Vss	Vcc	Setting prohibited
Vcc	Vss	
Vcc	Vcc	

■ Mode Data

Set always 00_H as the mode data in the internal ROM to select the single chip mode.

Figure 3.8-1 Structure of the Mode Data



Data	Operation
00_H	Single chip mode selection
Except 00_H	Reserved. Do not set here.

■ Selection of the Memory Access Mode

Only the single chip mode can be selected.

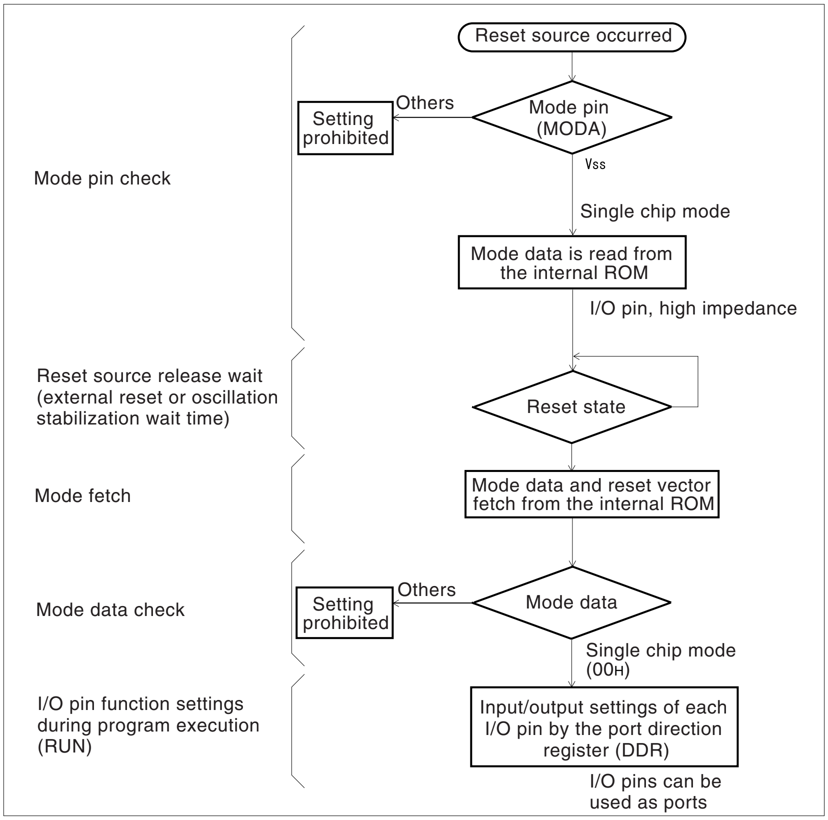
Table 3.8-2 "Mode pins and mode data" lists the mode pins and mode data.

Table 3.8-2 Mode Pins and Mode Data

Memory access mode	Mode pin (MOD0, 1)	Mode data
Single chip mode	Vss	00_H
Other modes	Setting prohibited	Setting prohibited

Figure 3.8-2 "Memory Access Selection Operation" shows the Operation of Memory Access Selection.

Figure 3.8-2 Memory Access Selection Operation


Mode pin check
Reset source release wait (external reset or oscillation stabilization wait time)
Mode fetch
Mode data check
I/O pin function settings during program execution (RUN)
Reset source occurred
Others
Setting prohibited
Mode pin (MODA)
Vss
Single chip mode
Mode data is read from the internal ROM
I/O pin, high impedance
Reset state
Mode data and reset vector fetch from the internal ROM
Others
Setting prohibited
Mode data
Single chip mode (00H)
Input/output settings of each I/O pin by the port direction register (DDR)
I/O pins can be used as ports

CHAPTER 4 I/O PORTS

This chapter describes the functions and operation of the I/O ports.

4.1 Overview of the I/O Ports

Fifty-three general-purpose I/O ports are provided.
Each of the ports can be used for peripheral devices (as pins for peripheral functions).

■ Functions of the I/O ports

The I/O ports output data received from the CPU to their I/O pins and input signals received at their I/O pins into the CPU through the port data registers (PDR). Some of the I/O ports provide a port direction register (DDR) to allow the direction (I/O) of each bit on the I/O pins to be specified.

The following shows the function each port and peripheral device also supported by the port:

- Port 0: General-purpose I/O port
- Port 1: General-purpose I/O port
- Port 2: General-purpose I/O port/Peripheral device (PWCK, 12-bit PPG01, 12-bit PPG02) also supported
- Port 3: General-purpose I/O port/Peripheral device (PPG03/MCO, SCK1(UCK1)/LMCO, SO1(UO1), SI1(UI1), PTO2, PWC, WTO, PTO1) also supported
- Port 4: General-purpose I/O port/Peripheral device (INT20/EC, INT21/SCK2, INT22/SO2/SDA[(*1)], INT23/SI2/SCL[(*1)], INT24/UCK2, INT25/UO2, INT26/UI2, INT27/ADST) also supported

  The functions of SDA and SCL (I^2C) cannot be used on the MB89535A, MB89537, MB89537H, MB89537A, MB89538, MB89538H, and MB89538A.
- Port 5: N-channel open-drain output-only port/Peripheral device (ADC input 8 channels) also supported
- Port 6: CMOS input-only port/Peripheral device (X1A, INT13/X0A, INT10 to INT12) also supported

*1 N-channel open-drain pin

Table 4.1-1 "List of the functions of each port" lists the functions of each port, and Table 4.1-2 "List of the registers on each port" lists the registers of each port.

Table 4.1-1 List of the functions of each port

Port name	Pin name	Input format	Output format	Function	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Port 0	P00 to P07	CMOS	CMOS push-pull	General-purpose I/O port	P07	P06	P05	P04	P03	P02	P01	P00
Port 1	P10 to P17				P17	P16	P15	P14	P13	P12	P11	P10
Port 2	P20/PWCK to P27	CMOS (Resources are hysteretic.)		General-purpose I/O port	P27	P26	P25	P24	P23	P22	P21	P20
				Peripheral device	-	-	-	-	-	PPG02	PPG01	PWCK
Port 3	P30/PPG03/MC to P37/PT01			General-purpose I/O port	P37	P36	P35	P34	P33	P32	P31	P30
				Peripheral device	PTO1	WTO	PWC	PTO2	SI1 (UI1)	SO1 (UO1)	SCK1 (UCK1) /LMCO	PPG03 /MCO
Port 4	P40/INT4/ EC to P47/ INT11/ADST		CMOS push-pull (P42/INT22/ SO2/SDA and P43/ INT23/SI2/ SCL are N-channel open-drains.)	General-purpose I/O port	P47	P46	P45	P44	P43	P42	P41	P40
				Peripheral device	INT27/ ADST	INT26/ UI2	INT25/ UO2	INT24/ UCK2	INT23/ SI2/ SCL(*1)	INT22/ SO2/ SDA(*1)	INT21/ SCK2	INT20/ EC
Port 5	P50/AN0 to P57/AN7	Analog input	N-channel open-drain	Output-only port	P57	P56	P55	P54	P53	P52	P51	P50
				Input to A/D converter	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
Port 6	P60/INT0 to P64/X1A	CMOS (Resources are hysteretic.)	X1A: CMOS push-pull	Input-only port	-	-	-	P64	P63	P62	P61	P60
				Peripheral device	-	-	-	X1A(*2)	INT13/ X0A	INT12	INT11	INT10

*1 Operates with N-channel open-drain output.

*2 Operates with CMOS output.

Table 4.1-2 List of the registers on each port

Register name	Read/Write	Address	Initial value
Port 0 data register (PDR0)	R/W	0000_H	$XXXXXXXX_B$
Port 0 direction register (DDR0)	W(*1)	0001_H	00000000_B
Port 0 pull-up resistor register (PURR0)	R/W	0072_H	11111111_B
Port 1 data register (PDR1)	R/W	0002_H	$XXXXXXXX_B$
Port 1 direction register (DDR1)	W(*1)	0003_H	00000000B
Port 1 pull-up resistor register (PURR1)	R/W	0073_H	11111111_B
Port 2 data register (PDR2)	R/W	$000C_H$	$XXXXXXXX_B$
Port 2 direction register (DDR2)	R/W	$000D_H$	00000000_B
Port 2 pull-up resistor register (PURR2)	R/W	0074_H	11111111_B
Port 3 data register (PDR3)	R/W	$000E_H$	$XXXXXXXX_B$
Port 3 direction register (DDR3)	R/W	$000F_H$	00000000_B
Port 3 pull-up resistor register (PURR3)	R/W	0075_H	11111111_B
Port 4 data register (PDR4)	R/W	0010_H	$XXXXXXXX_B$
Port 4 direction register (DDR4)	R/W	0011_H	$0000\text{--}00_B$
Port 4 pull-up resistor register (PURR4)	R/W	0076_H	$1111\text{--}11_B$
Port 5 data register (PDR5)	R/W	0012_H	11111111_B
Port 6 data register (PDR6)	R	0013_H	$XXXXXXXX_B$
Port 6 pull-up resistor register (PURR6)	R/W	0079_H	$\text{---}11111_B$

*1 DDR0 and DDR1 do not allow bit operation instructions to be used.
R/W: Readable/Writable
R: Read-only
W: Write-only
X: Unfixed
-: Unused

4.2 Port 0 and Port 1

**Port 0 and Port 1 are general-purpose I/O ports.
This section mainly describes the functions of the general-purpose I/O ports.
This section also describes the configurations, pins, and block diagram of the pins, and registers related to Port 0 and Port 1.**

■ Configurations of Port 0 and Port 1

Port 0 and Port 1 consist of the following four elements, respectively:

❍ Port 0

- General-purpose I/O-exclusive pins (P00 to P07)
- Port 0 data register (PDR0)
- Port 0 direction register (DDR0)
- Port 0 pull-up resistor control register (PURR0)

❍ Port 1

- General-purpose I/O-exclusive pins (P10 to P17)
- Port 1 data register (PDR1)
- Port 1 direction register (DDR1)
- Port 1 pull-up resistor control register (PURR1)

■ Pins of Port 0 and Port 1

Port 0 and Port 1 have eight CMOS I/O pins, respectively.

Table 4.2-1 "Pins of Port 0 and Port 1" shows the pins of Port 0 and Port 1.

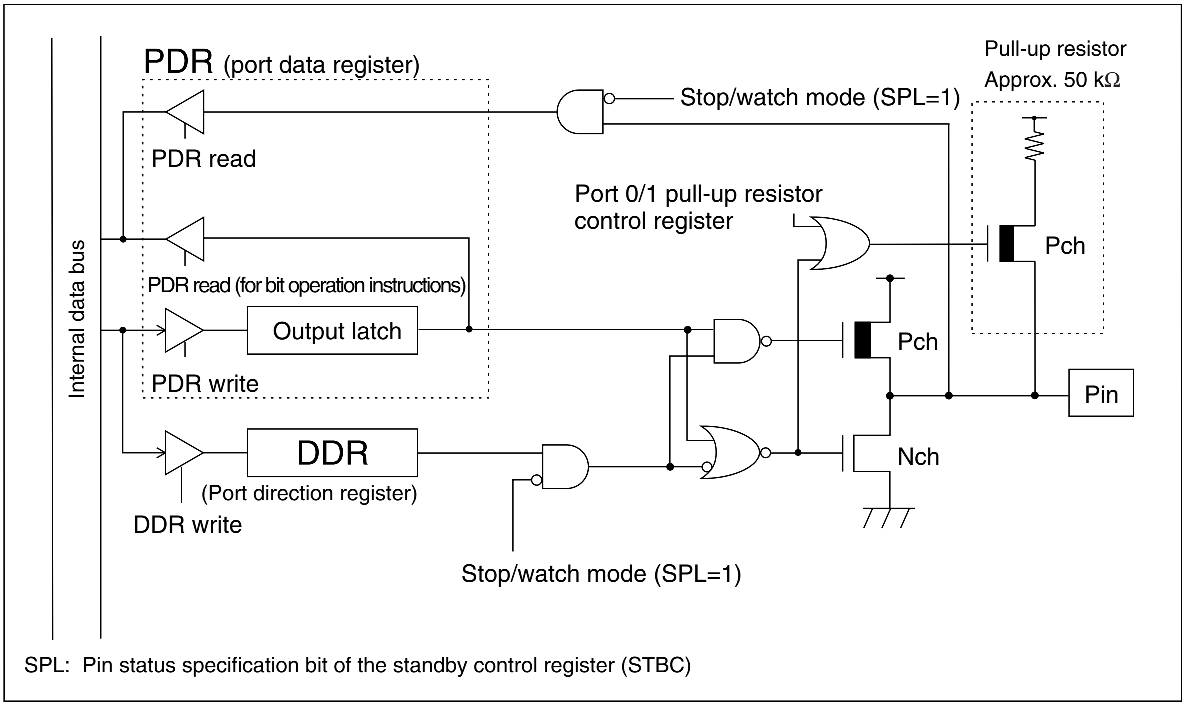
Table 4.2-1 Pins of Port 0 and Port 1

Port name	Pin name	Function	I/O format		Circuit type
			Input	Output	
Port 0	P00 to P07	General-purpose I/O	CMOS	CMOS	D
Port 1	P10 to P17				

For details of the circuit types, see Section 1.7 "Explanations of the Pin Functions."

■ Block diagram of Port 0 and Port 1

Figure 4.2-1 Block diagram of Port 0 and Port 1 pins



SPL: Pin status specification bit of the standby control register (STBC)

■ Registers of Port 0 and Port 1: PDR0 and DDR0

Port 0 has three related registers, PDR0, DDR0, and PURR0.

Port 1 has three related registers, PDR1, DDR1, and PURR1.

Each bit of the registers corresponds to a pin of Port 0 and Port 1.

Table 4.2-2 "Correspondence between bits and pins of the Port 0 and Port 1 registers" shows the correspondence between bits and pins of the Port 0 and Port 1 registers.

Table 4.2-2 Correspondence between bits and pins of the Port 0 and Port 1 registers

Port name	Correspondence between bits and pins of the related registers								
Port 0	PDR0,DDR0,PURR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Corresponding pin	P07	P06	P05	P04	P03	P02	P01	P00
Port 1	PDR1,DDR1,PURR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Corresponding pin	P17	P16	P15	P14	P13	P12	P11	P10

4.2.1 Registers of Port 0 and Port 1 (PDR0, DDR0, PURR0, PDR1, DDR1, PURR1)

This section describes the registers related to Port 0 and Port 1.

■ Functions of the Port 0 and Port 1 registers

❍ **Port 0 and Port 1 data registers (PDR0 and PDR1)**

The PDR0 and PDR1 registers indicate the states of the pins. Therefore, pins which have been set up for output allow for a value (0 or 1) which is the same as that of the output latch to be read. However, those set up for input do not allow for a value of the output latch to be read.

❍ **Port 0 and Port 1 direction registers (DDR0 and DDR1)**

The DDR0 and DDR1 registers set the direction (I/O) of each pin by bit.

Specifying 1 to the bit of a pin sets it up for output, and specifying 0 sets it up for input.

Reference:

The bit operation commands SETB and CLRB executed for registers other than the DDR0 or DDR1 register read states of output latches (not pins). Thus, executing bit operation instructions does not change the states of output latches related to bits not being operated.

Note:

The DDR0 and DDR1 registers are write-only: Do not use the bit operation instructions SETB and CLRB.

Table 4.2-3 "Functions of the Port 0 and Port 1 registers" shows the functions of the Port 0 and Port 1 registers.

Table 4.2-3 Functions of the Port 0 and Port 1 registers

Register name	Value	Reading	Writing	Read/ Write	Address	Initial value
Port 0 data register (PDR0)	0	The pin is Level L.	For pins set up for output, Level L is output to the pins.	R/W	0000_H	$XXXXXXXX_B$
	1	The pin is Level H.	For pins set up for output, Level H is output to the pins.			
Port 0 direction register (DDR0)	0	Unreadable	Operation of the output transistors is disabled, and the pins are set up for input.	W	0001_H	00000000_B
	1	Unreadable	Operation of the output transistors is enabled, and the pins are set up for output.			
Port 1 data register (PDR1)	0	The pin is Level L.	For pins set up for output, Level L is output to the pins.	R/W	0002_H	$XXXXXXXX_B$
	1	The pin is Level H.	For pins set up for output, Level H is output to the pins.			

Table 4.2-3 Functions of the Port 0 and Port 1 registers (Continued)

Register name	Value	Reading	Writing	Read/ Write	Address	Initial value
Port 1 direction register (DDR1)	0	Unreadable (write-only)	Operation of the output transistors is disabled, and the pins are set up for input.	W	0003_H	00000000_B
	1		Operation of the output transistors is enabled, and the pins are set up for output.			

R/W: Readable/Writable
W: Write-only
X: Not specified

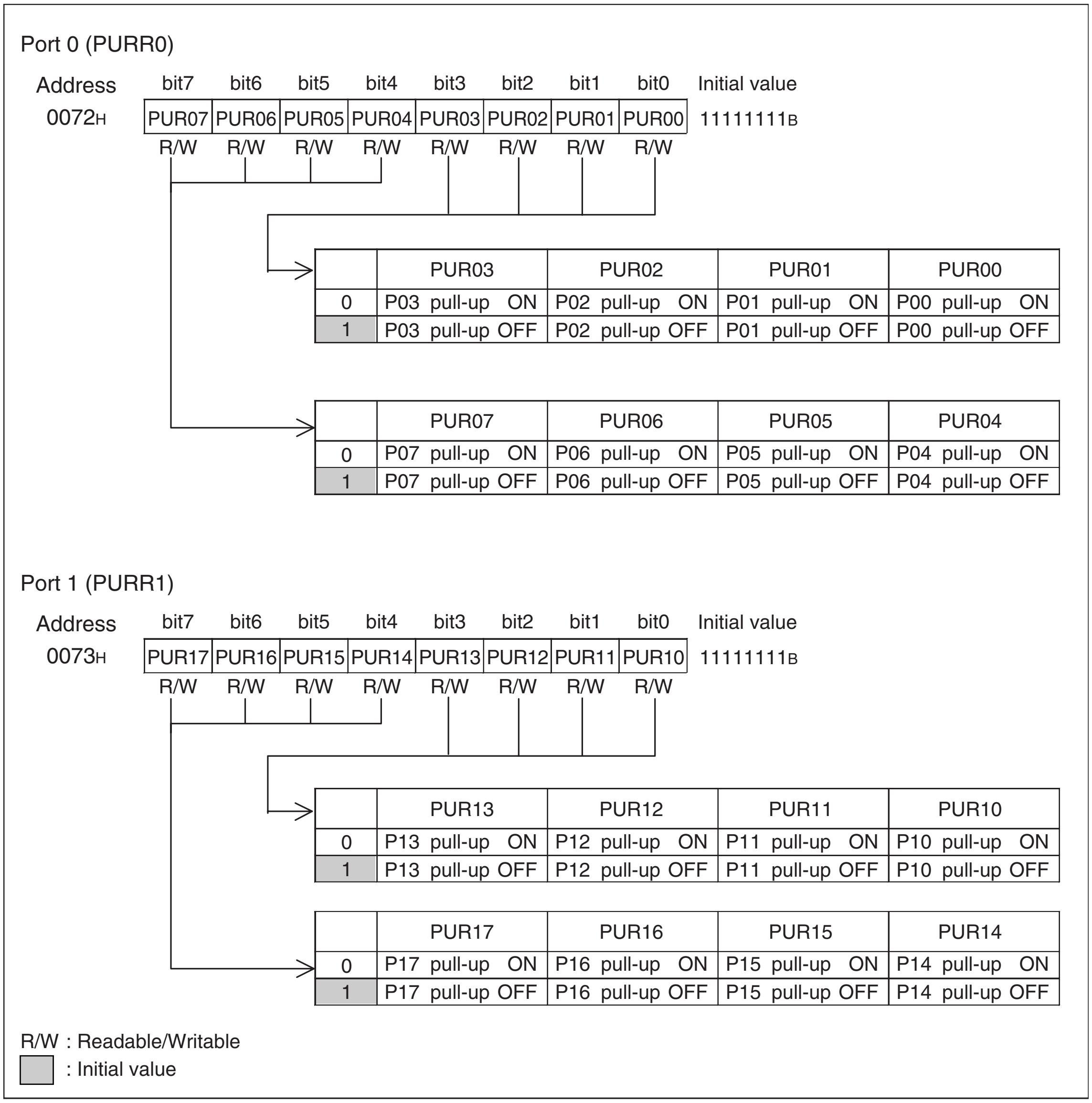
■ Port 0 and Port 1 pull-up resistor control registers (PURR0 and PURR1)

Each Port 0 and Port 1 pin allows a pull-up resistor to be used. Write a pull-up resistor setting to each bit of the Port 0 and Port 1 pull-up resistor control registers.

Enabling pull-up resistors in stop mode or watch mode (STBC:SPL=1) using the Port 0 and Port 1 pull-up resistor control registers does not set the pins to high impedance but rather to Level H (pulled up). However, during a reset, pull-up of the pins is invalidated and the pins are set to high impedance. (The Port 0 and Port 1 pull-up resistor control registers are initialized by a reset.)

Figure 4.2-2 "Settings of the Port 0 and Port 1 pull-up register registers (PURR0 and PURR1)" shows the allowable settings of the Port 0 and Port 1 pull-up resistor control registers.

Figure 4.2-2 Settings of the Port 0 and Port 1 pull-up register registers (PURR0 and PURR1)



Port 0 (PURR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0072H	PUR07	PUR06	PUR05	PUR04	PUR03	PUR02	PUR01	PUR00	11111111B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	PUR03	PUR02	PUR01	PUR00
0	P03 pull-up ON	P02 pull-up ON	P01 pull-up ON	P00 pull-up ON
▨ 1	P03 pull-up OFF	P02 pull-up OFF	P01 pull-up OFF	P00 pull-up OFF

	PUR07	PUR06	PUR05	PUR04
0	P07 pull-up ON	P06 pull-up ON	P05 pull-up ON	P04 pull-up ON
▨ 1	P07 pull-up OFF	P06 pull-up OFF	P05 pull-up OFF	P04 pull-up OFF

Port 1 (PURR1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0073H	PUR17	PUR16	PUR15	PUR14	PUR13	PUR12	PUR11	PUR10	11111111B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	PUR13	PUR12	PUR11	PUR10
0	P13 pull-up ON	P12 pull-up ON	P11 pull-up ON	P10 pull-up ON
▨ 1	P13 pull-up OFF	P12 pull-up OFF	P11 pull-up OFF	P10 pull-up OFF

	PUR17	PUR16	PUR15	PUR14
0	P17 pull-up ON	P16 pull-up ON	P15 pull-up ON	P14 pull-up ON
▨ 1	P17 pull-up OFF	P16 pull-up OFF	P15 pull-up OFF	P14 pull-up OFF

R/W : Readable/Writable

▨ : Initial value

4.2.2 Operation of Port 0 and Port 1

This section describes the operation of Port 0 and Port 1.

■ Operation of Port 0 and Port 1

❍ Operation of pins set up for output

- Specifying 1 to a bit of the DDR0 or DDR1 register sets the corresponding pin for output.
- For an output pin, operation of the output transistor is permitted and data in the output latch is output to the pin.
- Data written to the PDR0 register and/or PDR1 register remains in the output latches and is output to the pins as is.
- Reading the PDR0 and PDR1 registers outputs the values of the pins. (Except for bit operation instructions)

❍ Operation of pins set up for input

- Specifying 0 to a bit of the DDR0 or DDR1 register sets the corresponding pin for input.
- For an input pin, the output transistor is "OFF" and the pin is set to high impedance.
- Data written to the PDR0 register and/or PDR1 register remains in the output latches but is not output to the pins.
- Reading the PDR0 and PDR1 registers outputs the values in the pins.

❍ Operation on a reset

- Resetting the CPU initializes the DDR0 register and DDR1 register with zeros. Therefore, input to the pins is allowed and the pins are set to high impedance.
- The PDR0 and PDR1 registers are not initialized by a reset. Therefore, to use the pins for output, set output data to the PDR0 and PDR1 registers before setting the DDR0 or DDR1 register bits in output mode.

❍ Operation in stop mode and watch mode

When Port 0 and Port1 are put in stop mode or watch mode, they are set to the status specified by the pin status specification bit of the standby control register, irrespective of the values of the DDR0 and DDR1 registers.

Table 4.2-4 "Status of the Port 0 and Port 1 pins" shows the status of the Port 0 and Port 1 pins in each mode.

Table 4.2-4 Status of the Port 0 and Port 1 pins

Pin name	Normal operation Main sleep Main stop (SPL=0) Sub-sleep Sub-stop (SPL=0) Watch mode (SPL=0)	Main stop (SPL=1) Sub-stop (SPL=1) Watch mode (SPL=1)	At reset
P00 to P07	General-purpose I/O port	Hi-z	Hi-z
P10 to P17			

SPL: Pin status specification bit of the standby control register (STBC:SPL)
Hi-z: High impedance

Reference:

When pull-up resistors are enabled by using the Port 0 or Port 1 pull-up resistor control register, the states of the pins do not become high impedance but rather become Level H (pulled-up) in stop mode or watch mode (STBC:SPL=1). However, the pins are not pulled up during a reset and their state becomes Hi-z.

4.3 Port 2

Port 2 is a general-purpose I/O port used and is also used for resource input and output. The function of each pin can be switched between general-purpose I/O port and resource input and output for each bit. This section mainly describes the functions of the general-purpose I/O port.

This section also describes the configuration, pins, and block diagram of the pins, and registers related to Port 2.

■ Configuration of Port 2

Port 2 consists of the following four types of element:

- General-purpose I/O pin (P23 to P27), general-purpose I/O pin/PWCK (P20), general-purpose I/O pin/PPG (P21/PPG01, P22/PPG02)
- Port 2 data register (PDR2)
- Port 2 direction register (DDR2)
- Port 2 pull-up resistor control register (PURR2)

■ Pins of Port 2

Port 2 has eight CMOS I/O pins.

Table 4.3-1 "Pins of Port 2" shows the pins of Port 2.

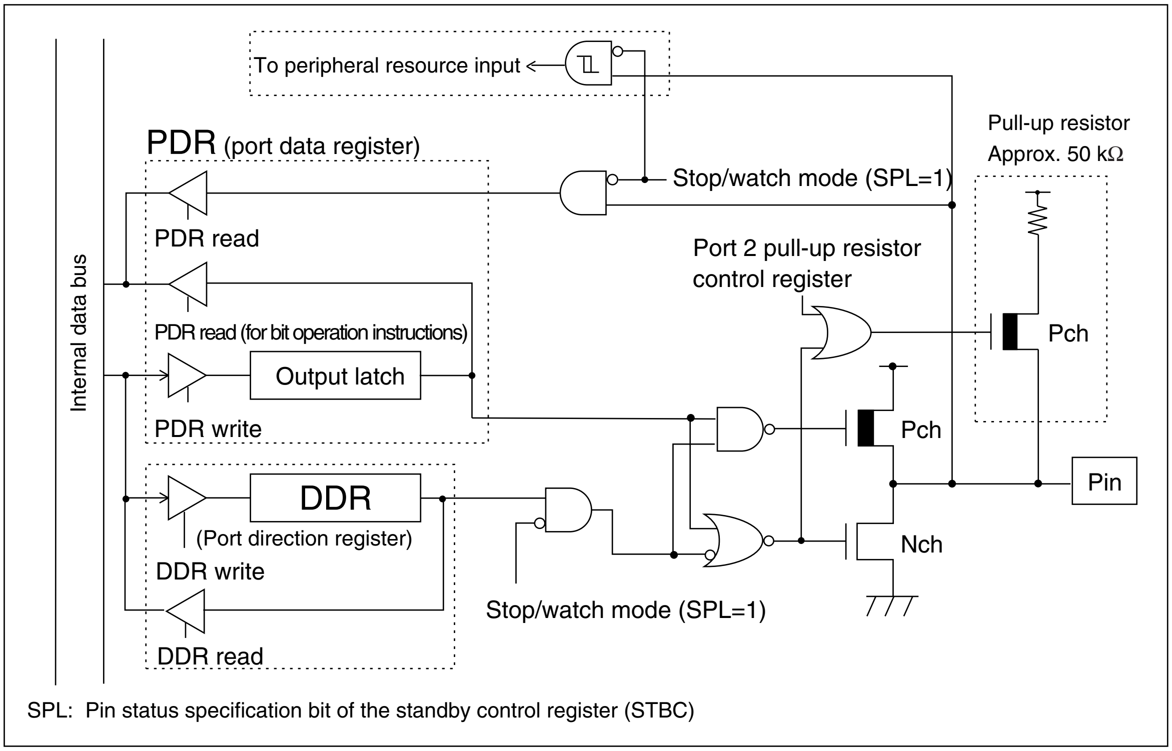
Table 4.3-1 Pins of Port 2

Port name	Pin name	Function	Peripheral device also supported	I/O format		Circuit type
				Input	Output	
Port 2	P23 to P27	P23 to P27 General-purpose I/O	-	CMOS	CMOS	D
	P20/PWCK	P20 General-purpose I/O	PWCK	CMOS (Resources are hysteretic.)		E
	P21/PPG01 to P22/PPG02	P21 toP22 General-purpose I/O	PPG01, PPG02	CMOS		D

For details of the circuit types, see Section 1.7 "Explanations of the Pin Functions."

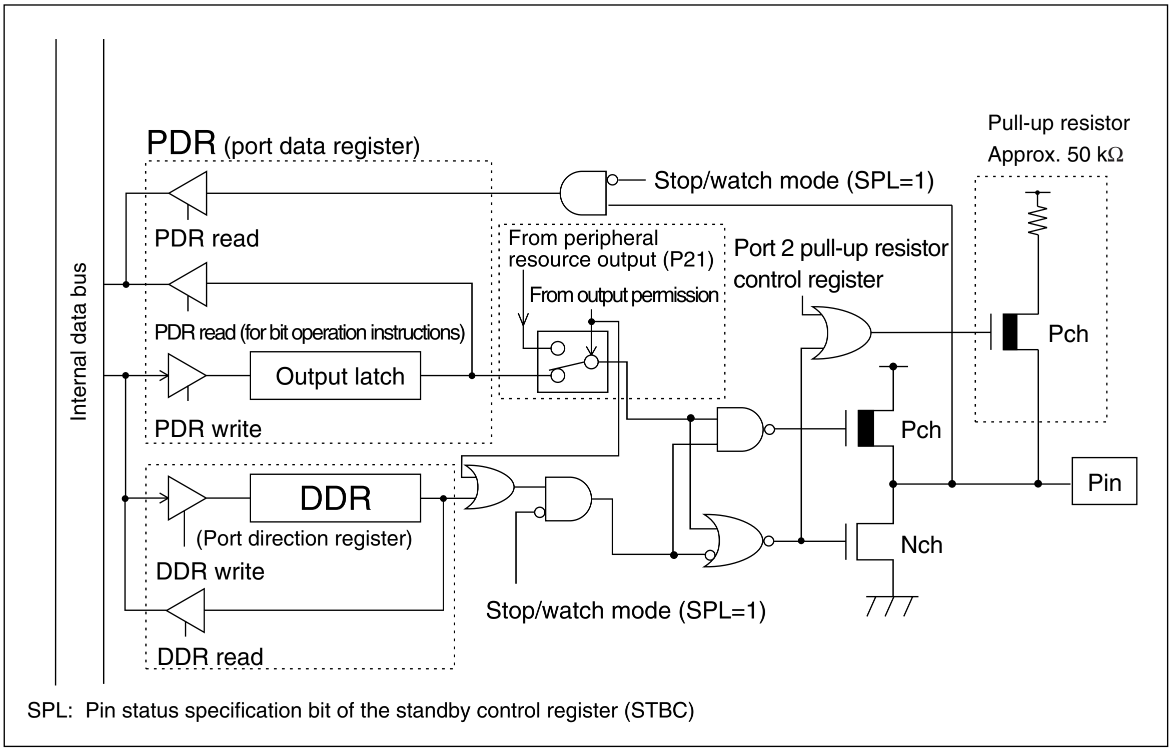
■ Block diagrams of Port 2

Figure 4.3-1 Block diagram of a Port 2 pin (for P20 only)


To peripheral resource input
PDR (port data register)
PDR read
PDR read (for bit operation instructions)
Output latch
PDR write
DDR
(Port direction register)
DDR write
DDR read
Internal data bus
Stop/watch mode (SPL=1)
Port 2 pull-up resistor control register
Pull-up resistor Approx. 50 kΩ
Pch
Pch
Nch
Pin
Stop/watch mode (SPL=1)

SPL: Pin status specification bit of the standby control register (STBC)

Figure 4.3-2 Block diagram of a Port 2 pin (for P21 only)


PDR (port data register)
PDR read
PDR read (for bit operation instructions)
Output latch
PDR write
DDR
(Port direction register)
DDR write
DDR read
Internal data bus
Stop/watch mode (SPL=1)
From peripheral resource output (P21)
From output permission
Port 2 pull-up resistor control register
Pull-up resistor Approx. 50 kΩ
Pch
Pch
Nch
Pin
Stop/watch mode (SPL=1)

SPL: Pin status specification bit of the standby control register (STBC)

Figure 4.3-3 Block diagram of a Port 2 pin (for P22 only)

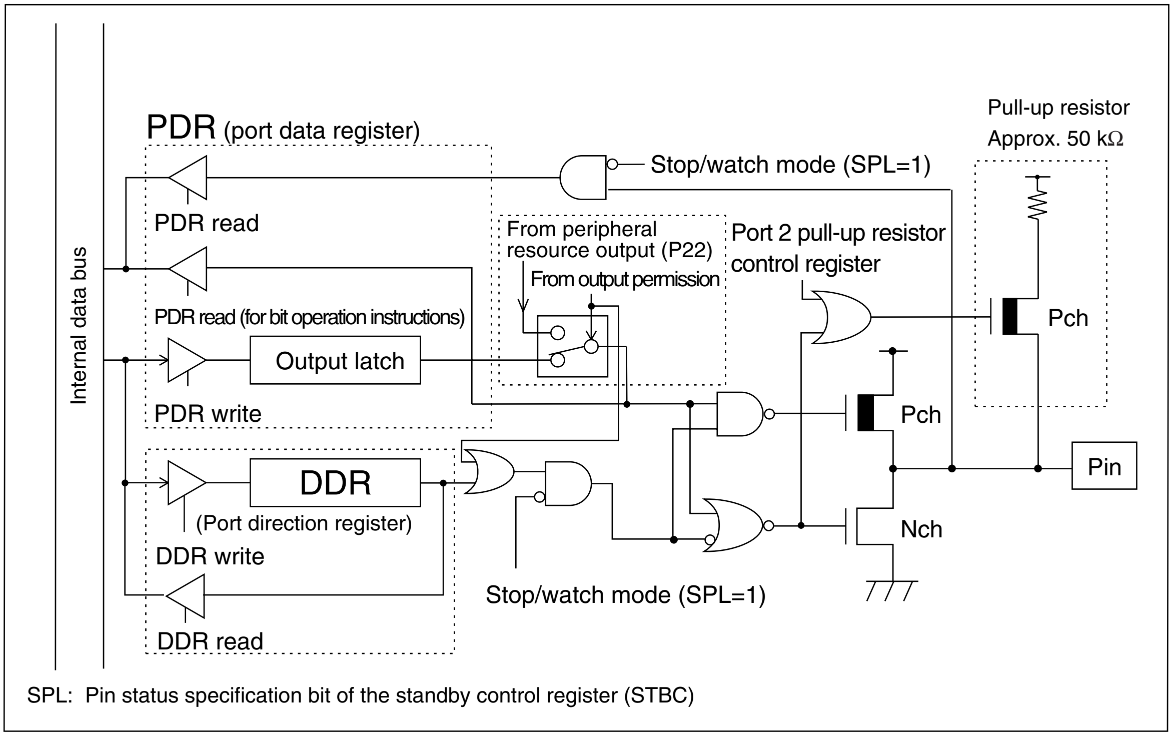
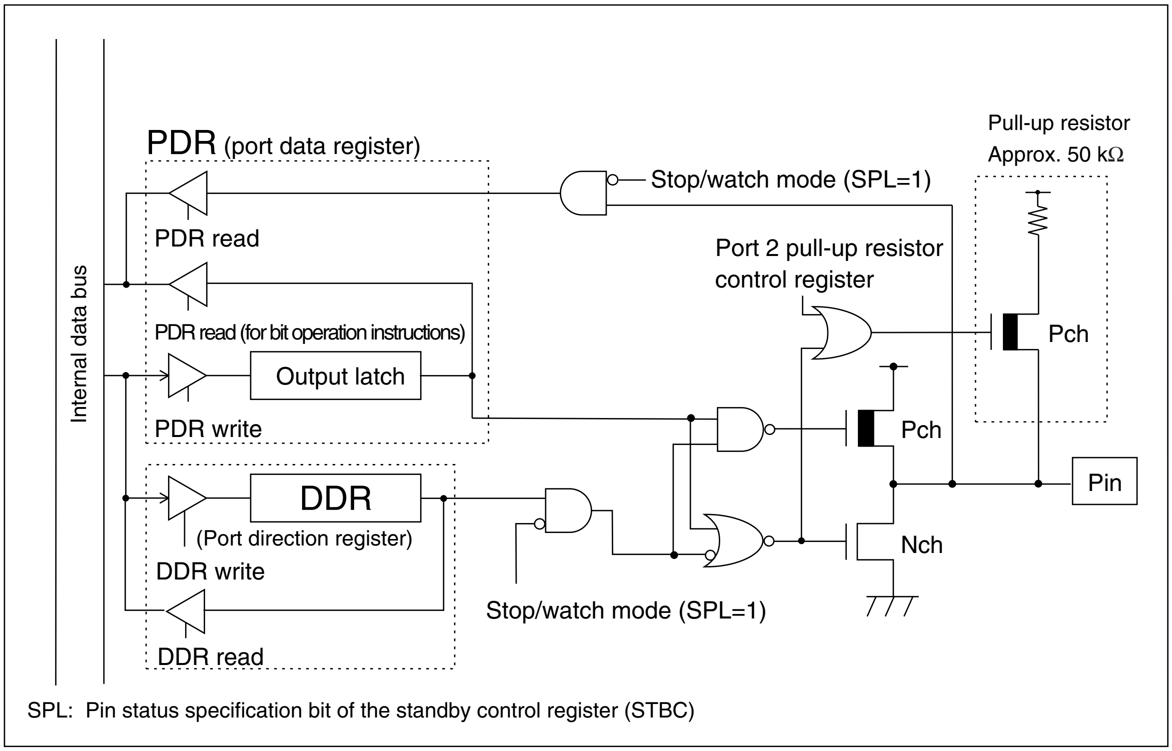

Internal data bus
PDR (port data register)
PDR read
PDR read (for bit operation instructions)
Output latch
PDR write
Stop/watch mode (SPL=1)
From peripheral resource output (P22)
From output permission
Port 2 pull-up resistor control register
Pull-up resistor
Approx. 50 kΩ
Pch
Pch
Pin
Nch
DDR
(Port direction register)
DDR write
DDR read
Stop/watch mode (SPL=1)
SPL: Pin status specification bit of the standby control register (STBC)

Figure 4.3-4 Block diagram of a Port 2 pin (for P23 to P27)


Internal data bus
PDR (port data register)
PDR read
PDR read (for bit operation instructions)
Output latch
PDR write
Stop/watch mode (SPL=1)
Port 2 pull-up resistor control register
Pull-up resistor
Approx. 50 kΩ
Pch
Pch
Pin
Nch
DDR
(Port direction register)
DDR write
DDR read
Stop/watch mode (SPL=1)
SPL: Pin status specification bit of the standby control register (STBC)

Note:

For input levels, CMOS input is applied to the pins and CMOS hysteresis input is applied to the resources. To use pins in input mode, prohibit operation of the corresponding resource.

■ Registers of Port 2

Port 2 has three related registers, PDR2, DDR2, and PURR2.

Each bit of the register corresponds to a pin of Port 2.

Table 4.3-2 "Correspondence between bit and pin of the Port 2 registers" shows the correspondence between bit and pin of the Port 2 registers.

Table 4.3-2 Correspondence between bit and pin of the Port 2 registers

Port name	Correspondence between bits and pins of the related registers								
Port 2	PDR2,DDR2,PURR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Corresponding pin	P27	P26	P25	P24	P23	P22	P21	P20

4.3.1 Registers of Port 2 (PDR2, DDR2, PURR2)

This section describes the registers related to Port 2.

■ Functions of the Port 2 registers

❍ **Port 2 data register (PDR2)**

The PDR2 register indicates the states of the pins. Therefore, pins which have been set up for output allow for a value (0 or 1) which is the same as that of the output latch to be read. However, those set up for input do not allow for a value of the output latch to be read.

❍ **Port 2 direction register (DDR2)**

The DDR2 register sets the direction (I/O) of each pin by bit.

Specifying 1 to the bit of a pin sets it up for output, and specifying 0 sets it up for input.

❍ **Settings for resource output**

To use a resource with output pins, enable each resource output permission bit.

Output from the resource takes priority; thus, the setting values of the PDR2 register and DDR2 which correspond to the resource output pins have no effect irrespective of the resource output values and output permission settings that have been specified.

❍ **Settings for resource input**

To use a resource with input pins, set pins which handle input from the resource for input. Values of the corresponding output latches have no effect.

Reference:

Bit operation commands (SETB, CLRB) do not read values from the pins but rather from output latches; thus, the values of output latches which correspond to the bit to be operated do not change. However, this does not apply to the pin described in the note below.

Note:

Pin exempted: P22 (PPG02) pin

Executing an RMW-related command to the Port 2 data register (PDR2) while the resource is operating reads the level of the pin.

Therefore, bit 2 of PDR2 may change its value.

Table 4.3-3 "Functions of the Port 2 registers" shows the functions of the Port 2 registers.

Table 4.3-3 Functions of the Port 2 registers

Register name	Value	Reading	Writing	Read/ Write	Address	Initial value
Port 2 data register (PDR2)	0	The pin is Level L.	Concerning pins set up for output, Level L is output to the pins.	R/W	$000C_H$	$XXXXXXXX_B$
	1	The pin is Level H.	Concerning pins set up for output, Level H is output to the pins.			
Port 2 direction register (DDR2)	0	Input pin	Operation of the output transistors is disabled, and the pins are set up for input.	R/W	$000D_H$	00000000_B
	1	Output pin	Operation of the output transistors is enabled, and the pins are set up for output.			

R/W: Readable/Writable
W: Write-only
X: Not specified

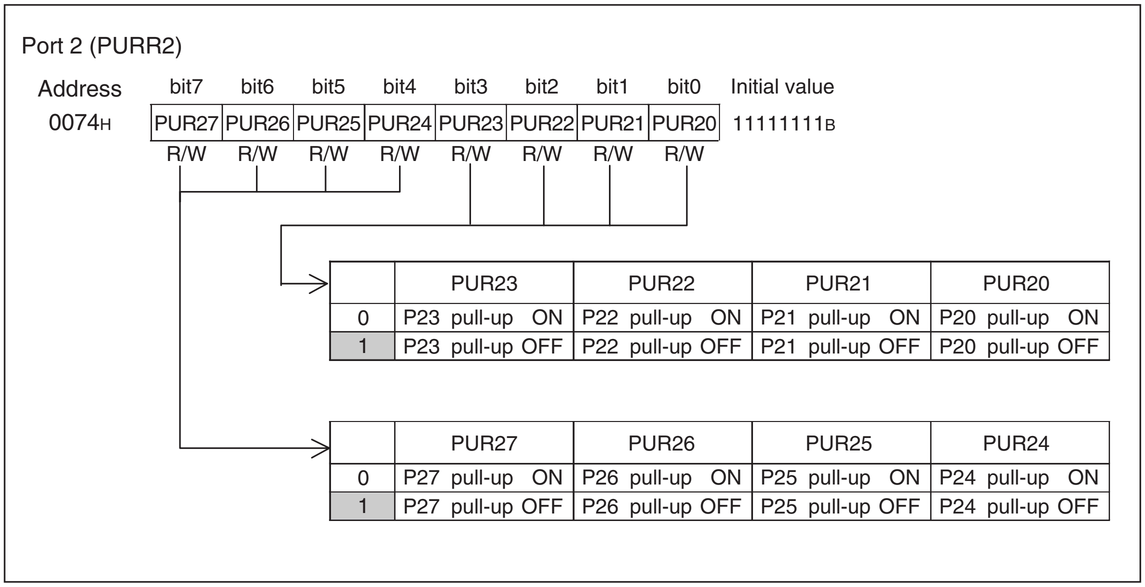
❍ Port 2 pull-up resistor control register (PURR2)

Each Port 2 pin allows a pull-up resistor to be used. Write a pull-up resistor setting to each bit of the Port 2 pull-up resistor control register.

Enabling pull-up resistors in stop mode or watch mode (STBC:SPL=1) using the Port 2 pull-up resistor control register does not set the pins to high impedance but rather to Level H (pulled up). However, during a reset, pull-up of the pins is invalidated and the pins are set to high impedance. (The Port 2 pull-up resistor control registers are initialized by a reset.)

Figure 4.3-5 "Settings of the Port 2 pull-up resistor control register (PURR2)" shows the allowable settings of the Port 2 pull-up resistor control register.

Figure 4.3-5 Settings of the Port 2 pull-up resistor control register (PURR2)



Port 2 (PURR2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0074_H	PUR27	PUR26	PUR25	PUR24	PUR23	PUR22	PUR21	PUR20	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	PUR23	PUR22	PUR21	PUR20
0	P23 pull-up ON	P22 pull-up ON	P21 pull-up ON	P20 pull-up ON
1	P23 pull-up OFF	P22 pull-up OFF	P21 pull-up OFF	P20 pull-up OFF

	PUR27	PUR26	PUR25	PUR24
0	P27 pull-up ON	P26 pull-up ON	P25 pull-up ON	P24 pull-up ON
1	P27 pull-up OFF	P26 pull-up OFF	P25 pull-up OFF	P24 pull-up OFF

4.3.2 Operation of Port 2

This section describes operation of Port 2.

■ Operation of Port 2

❍ Operation of pins set up for output

- Specifying 1 to a bit of the DDR2 register sets the corresponding pin for output.
- For an output pin, operation of the output transistor is permitted and data in the output latch is output to the pin.
- Data written to the PDR2 register remains in the output latches and is output to the pins as is.
- Reading the PDR2 register outputs the values of the pins.

❍ Operation of pins set up for input

- Specifying 0 to a bit of the DDR2 register sets the corresponding pin for input.
- For an input pin, the output transistor is "OFF" and the pin is set to high impedance.
- Data written to the PDR2 register remains in the output latches but is not output to the pins.
- Reading the PDR2 register outputs the values of the pins.

❍ Settings for resource output

- Enabling a resource output permission bit sets up the corresponding pins for resource output.

❍ Settings for resource input

- Specifying 0 to DDR2 register bits which correspond to resource input pins sets the pins for input.
- Values of the pins are always input into resource input (in stop mode or mode other than watch mode).
- Reading the PDR2 register outputs the values of the pins, irrespective of whether the resource is using input pins.

❍ Operation on a reset

- Resetting the CPU initializes the DDR2 register with zeros. Therefore, input to the pins is allowed and the pins are set to high impedance.
- The PDR2 register is not initialized by a reset. Therefore, to use the pins for output, set output data to the PDR2 register before setting the DDR2 register bits in output mode.

❍ Operation in stop mode and watch mode

When Port 2 is put in stop mode or watch mode, it is set to the status specified by the pin status specification bit of the standby control register, irrespective of the values of the DDR2 register.

Table 4.3-4 "Status of the Port 2 pins" shows the status of the Port 2 pins in each mode.

Table 4.3-4 Status of the Port 2 pins

Pin name	Normal operation Main sleep Main stop (SPL=0) Sub-sleep Sub-stop (SPL=0) Watch mode (SPL=0)	Main stop (SPL=1) Sub-stop (SPL=1) Watch mode (SPL=1)	At reset
P20/PWCK,P21/ PPG01,P22/ PPG02,P23 to P27	General-purpose I/O port	Hi-z	Hi-z

SPL: Pin status specification bit of the standby control register (STBC:SPL)
Hi-z: High impedance

Reference:

When pull-up resistors are enabled by using the Port 2 pull-up resistor control register, the states of the pins do not become high impedance but rather become Level H (pulled-up) in stop mode or watch mode (STBC:SPL=1). However, the pins are not pulled up during a reset and their state becomes Hi-z.

4.4 Port 3

Port 3 is a general-purpose I/O port and is also used to input and output resources. The function of each pin can be switched between general-purpose I/O port and resource input and output for each bit. This section mainly describes the functions of the general-purpose I/O port.
This section also describes the configuration, pins, and block diagram of the pins, and registers related to Port 3.

■ Configurations of Port 3

Port 3 consists of the following four types of element:

- General-purpose I/O pin/Resource I/O pins (P30/PPG03/MCO to P37/PT01)
- Port 3 data register (PDR3)
- Port 3 direction register (DDR3)
- Port 3 pull-up resistor control register (PURR3)

■ Pins of Port 3

Port 3 has eight CMOS I/O pins. These pins are used with various types of resources. When these pins are used with some kind of resource, Port 3 cannot be used as a general-purpose output port.

Table 4.4-1 "Pins of Port 3" shows the pins of Port 3.

Table 4.4-1 Pins of Port 3

Port name	Pin name	Function	Peripheral device also supported	I/O format		Circuit type
				Input	Output	
Port 3	P30/PPG03/ MCO	P30 general-purpose I/O	PPG03 output, MCO output	CMOS	CMOS	D
	P31/ SCK1(UCK1) /LMCO	P31 general-purpose I/O	UCK 8-bit UART/ SIO, clock I/O, LMCO output	CMOS (Resources are hysteretic.)		E
	P32/ SO1(UO1)	P32 general-purpose I/O	UO 8-bit UART/ SIO, data output	CMOS		D
	P33/SI1(UI1)	P33 general-purpose I/O	UI 8-bit UART/ SIO, data input	CMOS (Resources are hysteretic.)	CMOS	E
	P34/PTO2	P34 general-purpose I/O	PTO2 output	CMOS		D
	P35/PWC	P35 general-purpose I/O	PWC timer input	CMOS (Resources are hysteretic.)		E

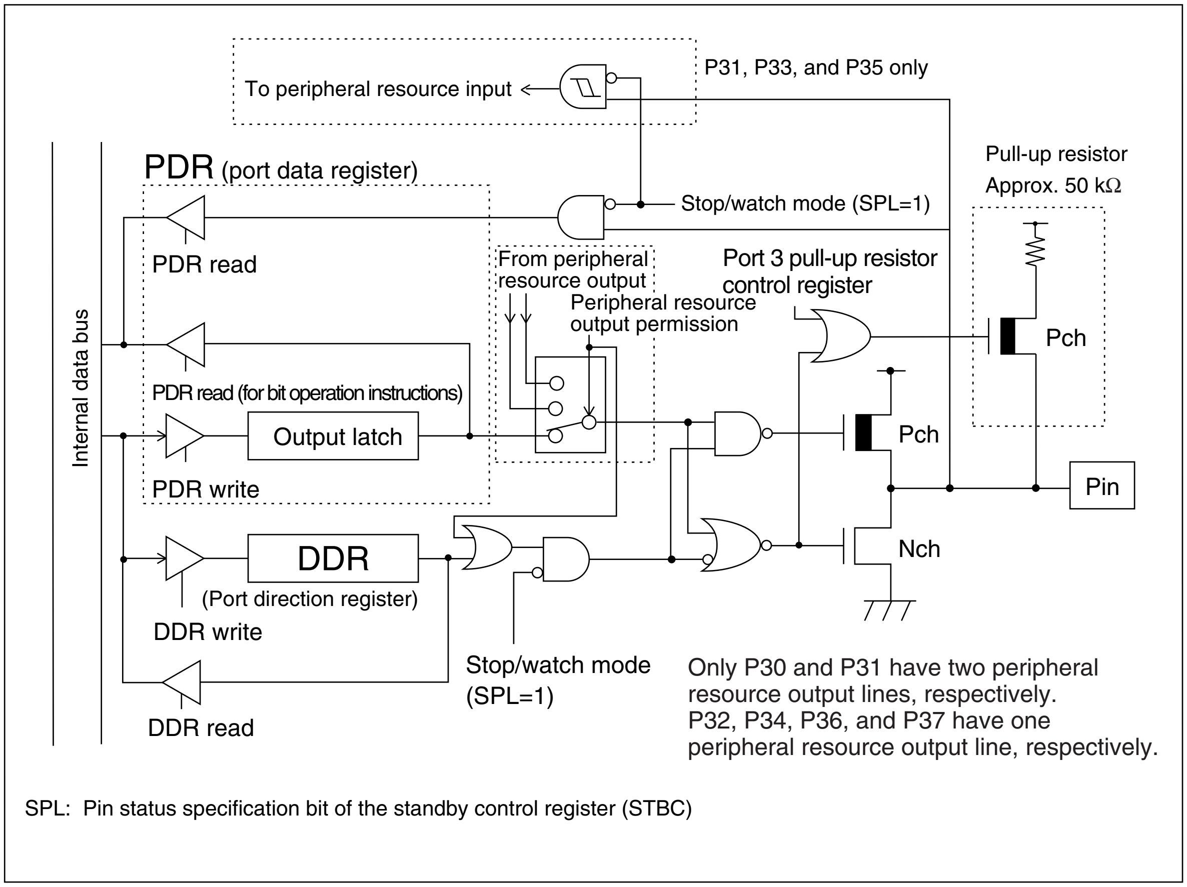
Table 4.4-1 Pins of Port 3 (Continued)

Port name	Pin name	Function	Peripheral device also supported	I/O format		Circuit type
				Input	Output	
Port 3	P36/WTO	P36 general-purpose I/O	PWC timer output	CMOS		D
	P37/PTO1	P37 general-purpose I/O	PTO1 output			D

For details of the circuit types, see Section 1.7 "Explanations of the Pin Functions."

■ Block diagram of Port 3

Figure 4.4-1 Block diagram of a Port 3 pin


To peripheral resource input
P31, P33, and P35 only
PDR (port data register)
Stop/watch mode (SPL=1)
Pull-up resistor
Approx. 50 kΩ
PDR read
From peripheral resource output
Port 3 pull-up resistor control register
Peripheral resource output permission
Internal data bus
Pch
PDR read (for bit operation instructions)
Output latch
Pch
PDR write
Pin
DDR
Nch
(Port direction register)
DDR write
Stop/watch mode (SPL=1)
DDR read
Only P30 and P31 have two peripheral resource output lines, respectively.
P32, P34, P36, and P37 have one peripheral resource output line, respectively.
SPL: Pin status specification bit of the standby control register (STBC)

Reference:

Continuously input values to the peripheral resource input (in mode other than stop mode/ watch mode).

Note:

For input levels, CMOS input is applied to the pins and CMOS hysteresis input is applied to the resources.

■ Registers of Port 3

Port 3 has three related registers, PDR3, DDR3, and PURR3.

Each bit of the register corresponds to a pin of Port 3.

Table 4.4-2 "Correspondence between bit and pin of the Port 3 registers" shows the correspondence between bit and pin of the Port 3 registers.

Table 4.4-2 Correspondence between bit and pin of the Port 3 registers

Port name	Correspondence between bits and pins of the related registers								
Port 3	PDR3,DDR3,PURR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Corresponding pin	P37	P36	P35	P34	P33	P32	P31	P30

4.4.1 Registers of Port 3 (PDR3, DDR3, PURR3)

This section describes the registers related to Port 3.

■ Functions of the Port 3 registers

❍ Port 3 data register (PDR3)

The PDR3 register indicates the states of the pins. Therefore, pins which have been set up for output allow for a value (0 or 1) which is the same as that of the output latch to be read. However, those set up for input do not allow for a value of the output latch to be read.

Reference:

The bit operation instructions SETB and CLRB read states of output latches (not pins). Thus, executing either bit operation command does not change the states of output latches related to bits not being operated.

❍ Port 3 direction register (DDR3)

The DDR3 register sets the direction (I/O) of each pin by bit.

Specifying 1 to the bit of a pin sets it up for output, and specifying 0 sets it up for input.

❍ Settings for resource output

To use a resource with output pins, enable each resource output permission bit.

Output from the resource takes priority; thus, setting values of the PDR3 register and DDR3 which correspond to the resource output pins have no effect irrespective of the resource output values and output permission settings that have been specified.

❍ Settings for resource input

To use a resource with input pins, set pins which handle input from the resource for input. Values of the corresponding output latches have no effect.

Table 4.4-3 "Functions of the Port 3 registers" shows the functions of the Port 3 registers.

Table 4.4-3 Functions of the Port 3 registers

Register name	Value	Reading	Writing	Read/ Write	Address	Initial value
Port 3 data register (PDR3)	0	The pin is Level L.	Concerning pins set up for output, Level L is output to the pins.	R/W	$000E_H$	$XXXXXXXX_B$
	1	The pin is Level H.	Concerning pins set up for output, Level H is output to the pins.			
Port 3 direction register (DDR3)	0	Input pin	Operation of the output transistors is disabled, and the pins are set up for input.	R/W	$000F_H$	00000000_B
	1	Output pin	Operation of the output transistors is enabled, and the pins are set up for output.			

R/W: Readable/Writable
X: Not specified

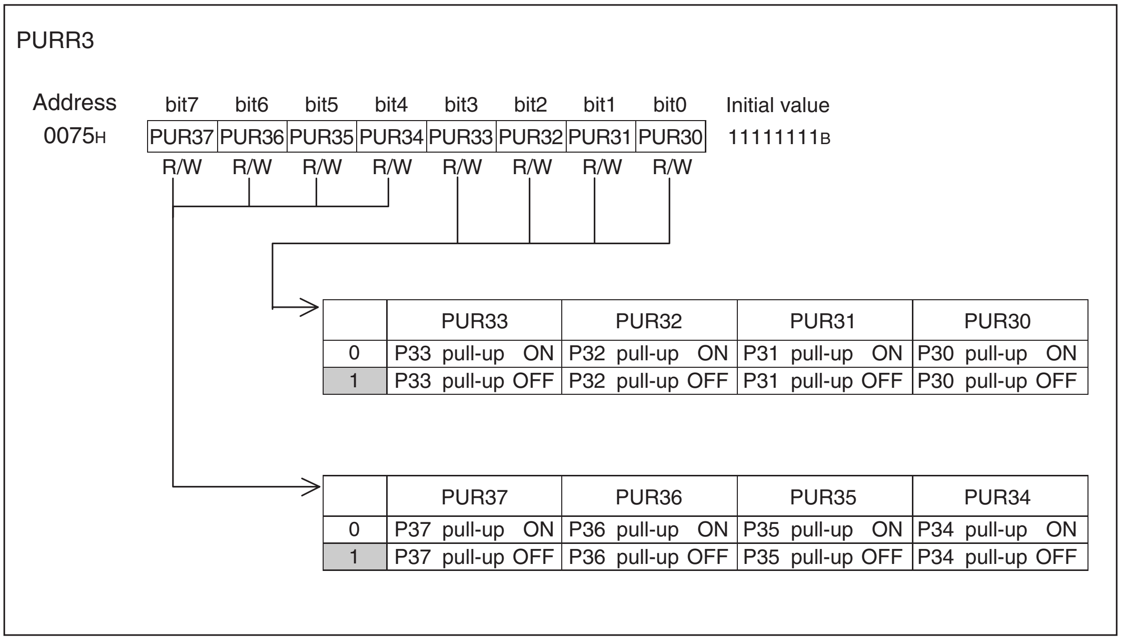
❍ Port 3 pull-up resistor control register (PURR3)

Each Port 3 pin allows a pull-up resistor to be used. Write a pull-up resistor setting to each bit of the Port 3 pull-up resistor control register.

Enabling pull-up resistors in stop mode or watch mode (STBC:SPL=1) using the Port 3 pull-up resistor control register does not set the pins to high impedance but rather to Level H (pulled up). However, during a reset, pull-up of the pins is invalidated and the pins are set to high impedance. (The Port 3 pull-up resistor control registers are initialized by a reset.)

Figure 4.4-2 "Settings of the Port 3 pull-up resistor control register (PURR3)" shows the allowable settings of the Port 3 pull-up resistor control register.

Figure 4.4-2 Settings of the Port 3 pull-up resistor control register (PURR3)



PURR3

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0075_H	PUR37	PUR36	PUR35	PUR34	PUR33	PUR32	PUR31	PUR30	11111111_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	PUR33	PUR32	PUR31	PUR30
0	P33 pull-up ON	P32 pull-up ON	P31 pull-up ON	P30 pull-up ON
1	P33 pull-up OFF	P32 pull-up OFF	P31 pull-up OFF	P30 pull-up OFF

	PUR37	PUR36	PUR35	PUR34
0	P37 pull-up ON	P36 pull-up ON	P35 pull-up ON	P34 pull-up ON
1	P37 pull-up OFF	P36 pull-up OFF	P35 pull-up OFF	P34 pull-up OFF

4.4.2 Operation of Port 3

This section describes operation of Port 3.

■ Operation of Port 3

❍ **Operation of pins set up for output**

- Specifying 1 to a bit of the DDR3 register sets the corresponding pin for output.
- When pins are set up for output, the output transistors are enabled and data of the output latches are output to the pins.
- Data written to the PDR3 register remains in the output latches and is output to the pins as is.
- Reading the PDR3 register outputs the values of the pins.

❍ **Operation of pins set up for input**

- Specifying 0 to a bit of the DDR3 register sets the corresponding pin for input.
- For an input pin, the output transistor is "OFF" and the pin is set to high impedance.
- Data written to the PDR3 register remains in the output latches but is not output to the pins.
- Reading the PDR3 register outputs the values of the pins.

❍ **Settings for resource output**

- Enabling a resource output permission bit sets up the corresponding pins for resource output.
- Even if resource output is enabled, values of the pins can be read by using the PDR3 register; thus, resource output values can be read.

❍ **Settings for resource input**

- Specifying 0 to DDR3 register bits which correspond to resource input pins set the pins for input.
- Reading the PDR3 register outputs the values of the pins, irrespective of whether the resource is using input pins.

❍ **Operation on a reset**

- Resetting the CPU initializes the DDR3 register with zeros. Thus, the output transistors are turned "OFF" (the pins are set for input) and the pins are set to high impedance.
- The PDR3 register is not initialized by a reset. Therefore, to use the pins for output, set output data to the PDR3 register before setting the DDR3 register bits in output mode.

❍ **Operation in stop mode and watch mode**

When Port 3 is put in stop mode or watch mode, it is set to the status specified by the pin status specification bit of the standby control register, irrespective of the values of the DDR3 register.

Table 4.4-4 "Status of the Port 3 pins" shows the status of the Port 3 pins in each mode.

Table 4.4-4 Status of the Port 3 pins

Pin name	Normal operation Main sleep Main stop (SPL=0) Sub-sleep Sub-stop (SPL=0) Watch mode (SPL=0)	Main stop (SPL=1) Sub-stop (SPL=1) Watch mode (SPL=1)	At reset
P30/PPG03/MCO to P37/PT01	General-purpose I/O port/ Resource I/O	Hi-z	Hi-z

SPL: Pin status specification bit of the standby control register (STBC:SPL)
Hi-z: High impedance

Reference:

When pull-up resistors are enabled by using the Port 3 pull-up resistor control register, the states of the pins do not become high impedance but rather become Level H (pulled-up) in stop mode or watch mode (STBC:SPL=1). However, the pins are not pulled up during a reset and their state becomes Hi-z.

4.5 Port 4

Port 4 is a general-purpose I/O port and is also used to input and output resources. The function of each pin can be switched between general-purpose I/O port and resource input and output for each bit. This section mainly describes the functions of the general-purpose I/O port.
This section also describes the configuration, pins, and block diagram of the pins, and registers related to Port 4.

■ Configurations of Port 4

Port 4 consists of the following five types of element:

- General-purpose I/O pin/External interrupt 2/Resource I/O pin (P40/INT20/EC to P47/INT27/ADST)
- Port 4 data register (PDR4)
- Port 4 direction register (DDR4)
- Port 4 pull-up resistor control register (PURR4)
- DDC select register (DDCR)

■ Pins of Port 4

Port 4 has eight CMOS I/O pins.

When P40/INT20/EC to P47/INT27/ADST are used for input, the Port 4 pins can be used as external interrupt input pins.

Table 4.5-1 "Pins of Port 4" shows the pins of Port 4.

Table 4.5-1 Pins of Port 4

Port name	Pin name	Function	Peripheral device also supported	I/O format		Circuit type
				Input	Output	
Port 4	P40/INT20/EC	P40 general-purpose I/O	External interrupt, external clock input pin	CMOS (Resources are hysteretic.)	CMOS	E
	P41/INT21/SCK2	P41 general-purpose I/O	External interrupt, external clock input pin			E
	P42/INT22/SO2/SDA	P42 N-channel open-drain I/O	External interrupt, SO2 output, I^{2}C data pin		N-channel open drain	G
	P43/INT23/SI2/SCL	P43 N-channel open-drain I/O	External interrupt, SI2 input, I^{2}C clock pin			G

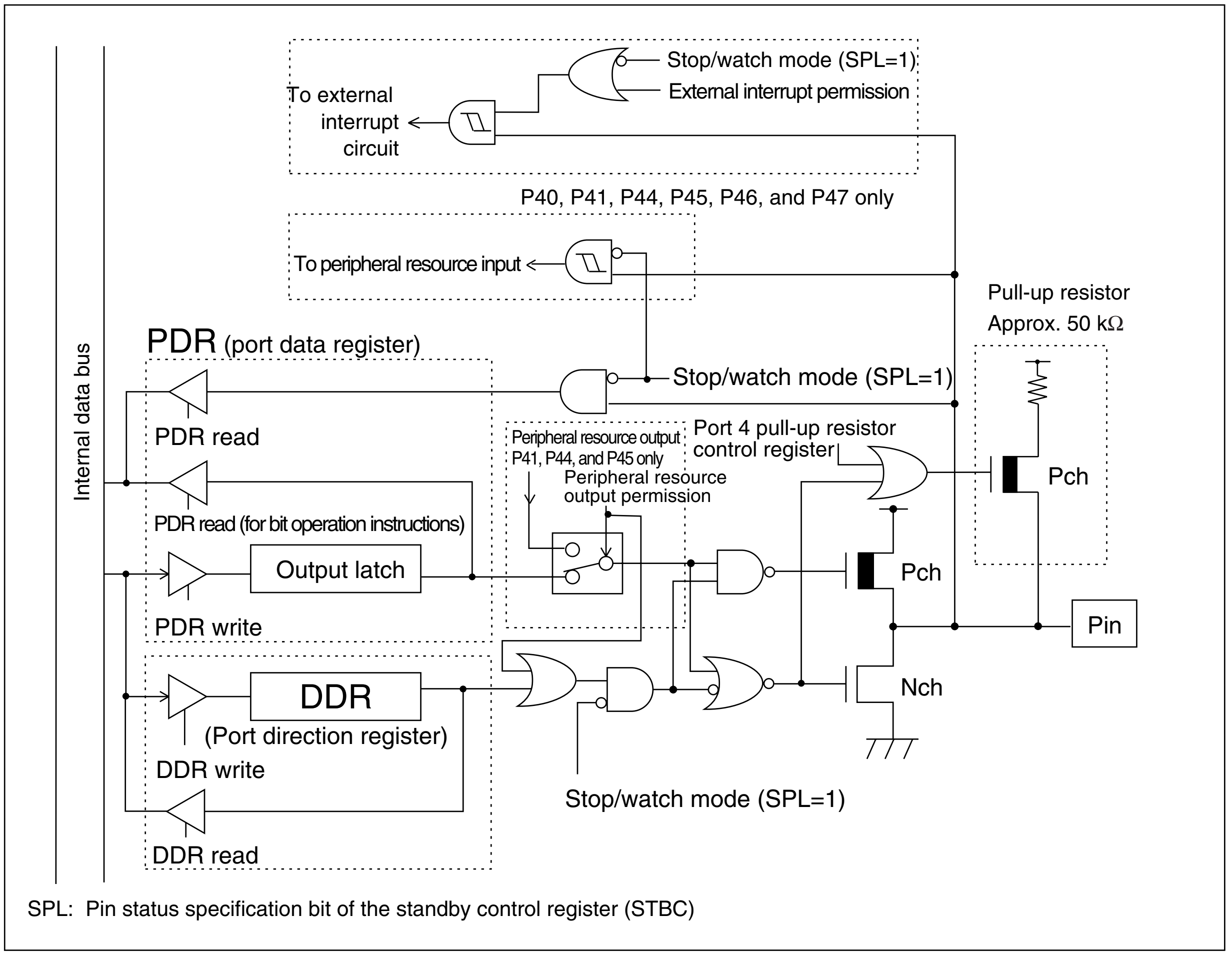
Table 4.5-1 Pins of Port 4 (Continued)

Port name	Pin name	Function	Peripheral device also supported	I/O format		Circuit type
				Input	Output	
Port 4	P44/INT24/ UCK2	P44 general-purpose I/O	External interrupt, UCK 8-bit UART	CMOS (Resources are hysteretic.)	CMOS	E
	P45/INT25/ UO2	P45 general-purpose I/O	External interrupt, UO 8-bit UART			E
	P46/INT26/ UI2	P46 general-purpose I/O	External interrupt, UI 8-bit UART			E
	P47/INT27/ ADST	P47 general-purpose I/O	External interrupt, A/D activation pin			E

For details of the circuit types, see Section 1.7 "Explanations of the Pin Functions."

■ Block diagrams of Port 4

Figure 4.5-1 Block diagram of Port 4 pins (P40, P41, and P44 to P47)


Stop/watch mode (SPL=1)
External interrupt permission
To external interrupt circuit
P40, P41, P44, P45, P46, and P47 only
To peripheral resource input
Pull-up resistor
Approx. 50 kΩ
Internal data bus
PDR (port data register)
Stop/watch mode (SPL=1)
PDR read
Peripheral resource output P41, P44, and P45 only
Port 4 pull-up resistor control register
Peripheral resource output permission
Pch
PDR read (for bit operation instructions)
Output latch
Pch
PDR write
Pin
DDR
Nch
(Port direction register)
DDR write
Stop/watch mode (SPL=1)
DDR read

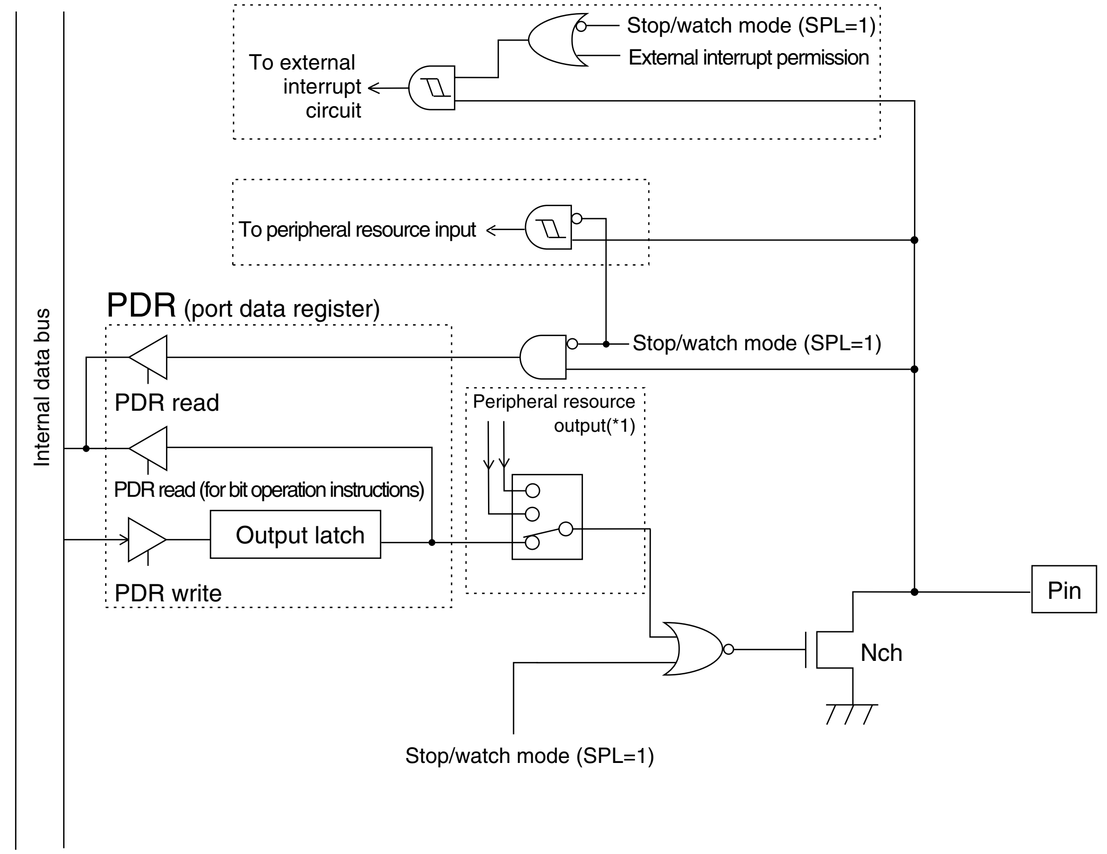
SPL: Pin status specification bit of the standby control register (STBC)

Note:

When Port 4 is used as an ordinary input port, operation of external interrupt circuits which use any of the Port 4 pins must be prohibited. See Chapter 13 "External Interrupt Circuit 1 (Edge)" and Chapter 14 "External Interrupt Circuit 2 (Level)."

For input levels, CMOS input is applied to the pins and CMOS hysteresis input is applied to the resources.

Figure 4.5-2 Block diagram of Port 4 pins (P42, P43)


Stop/watch mode (SPL=1)
External interrupt permission
To external interrupt circuit
To peripheral resource input
Internal data bus
PDR (port data register)
Stop/watch mode (SPL=1)
PDR read
Peripheral resource output(*1)
PDR read (for bit operation instructions)
Output latch
PDR write
Pin
Nch
Stop/watch mode (SPL=1)

SPL: Pin status specification bit of the standby control register (STBC)

*1 P42 has two peripheral resource output lines.
P43 has one peripheral resource output line.

Note:

When Port 4 is used as an ordinary input port, operation of external interrupt circuits which use any of the Port 4 pins must be prohibited. See Chapter 13 "External Interrupt Circuit 1 (Edge)" and Chapter 14 "External Interrupt Circuit 2 (Level)."

For input levels, CMOS input is applied to the pins and CMOS hysteresis input is applied to the resources.

Reference:

To use P42 and P43 as I^2C output pins or N-channel open-drain output pins, pull-up resistors are required on the external pins.

Note that I^2C can be used with the MB89PV530, MB89P538, MB89537C/538C, MB89537HC/538HC, and MB89537AC/538AC only.

■ Registers of Port 4

Port 4 has three related registers, PDR4, DDR4, and PURR4.

Each bit of the register corresponds to a pin of Port 4.

Table 4.5-2 "Correspondence between bit and pin of the Port 4 registers" shows the correspondence between bit and pin of the Port 4 registers.

Table 4.5-2 Correspondence between bit and pin of the Port 4 registers

Port name	Correspondence between bits and pins of the related registers								
Port 4	PDR4,DDR4,PURR4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Corresponding pin	P47	P46	P45	P44	P43(*1)	P42(*1)	P41	P40

*1 Bit 2 and bit 3 of PURR4 are unused. Neither P42 nor P43 have internal pull-up resistors.

4.5.1 Registers of Port 4 (PDR4, DDR4, PURR4, DDCR)

This section describes the registers related to Port 4.

■ Functions of the Port 4 registers.

❍ **Port 4 data register (PDR4)**

The PDR4 register indicates the states of the pins. Therefore, pins which have been set up for output allow for a value (0 or 1) which is the same as that of the output latch to be read. However, those set up for input do not allow for a value of the output latch to be read.

Reference:

Bit operation commands SETB and CLRB read states of output latches (not pins). Thus, executing either bit operation command does not change the states of output latches related to bits not being operated.

❍ **Port 4 direction register (DDR4)**

The DDR4 register sets the direction (I/O) of each pin by bit.

Specifying 1 to the bit of a pin sets it up for output, and specifying 0 sets it up for input. (Note that the DDR4 register does not allow bit 2 and bit 3 to be used.)

❍ **Settings for external interrupt input**

In addition to use of interrupt circuit (external interrupt 1 or 2) permission, to use some of the pins as external interrupt input pins, the pins corresponding to them must be set for input.

(The corresponding output latch data has no effect in this case.)

Table 4.5-3 "Functions of the Port 4 registers" shows the functions of the Port 4 registers.

Table 4.5-3 Functions of the Port 4 registers

Register name	Value	Reading	Writing	Read/ Write	Address	Initial value
Port 4 data register (PDR4)	0	The pin is Level L.	0 is set to the output latches, and Level L is output to the pins set up for output.	R/W	0010_H	$XXXX11XX_B$
	1	The pin is Level H.	1 is set to the output latches, and Level H is output to the pins set up for output.			
Port 4 direction register (DDR4)	0	Input pin	Operation of the output transistors is disabled, and the pins are set up for input.	R/W	0011_H	$0000\text{--}00_B$
	1	Output pin	Operation of the output transistors is enabled, and the pins are set up for output.			

R/W: Readable/Writable
-: Unused
X: Not specified

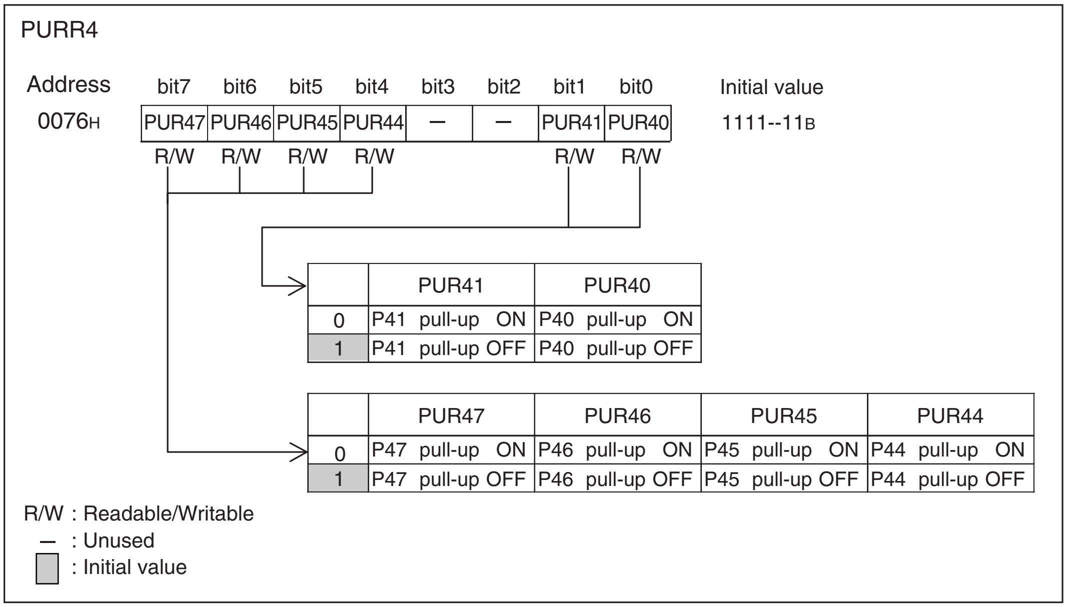
❍ Port 4 pull-up resistor control register (PURR4)

Each Port 4 pin allows a pull-up resistor to be used. Write a pull-up resistor setting to each bit of the Port 4 pull-up resistor control register.

Enabling pull-up resistors in stop mode or watch mode (STBC:SPL=1) using the Port 4 pull-up resistor control register does not set the pins to high impedance but rather to Level H (pulled up). However, during a reset, pull-up of the pins is invalidated and the pins are set to high impedance. (The Port 4 pull-up resistor control registers are initialized by a reset.)

Figure 4.5-3 "Settings of the Port 4 pull-up resistor control register (PURR4)" shows the allowable settings of the Port 4 pull-up resistor control register.

Figure 4.5-3 Settings of the Port 4 pull-up resistor control register (PURR4)



PURR4

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0076_H	PUR47	PUR46	PUR45	PUR44	—	—	PUR41	PUR40	$1111\text{--}11_B$
	R/W	R/W	R/W	R/W			R/W	R/W	

	PUR41	PUR40
0	P41 pull-up ON	P40 pull-up ON
1	P41 pull-up OFF	P40 pull-up OFF

	PUR47	PUR46	PUR45	PUR44
0	P47 pull-up ON	P46 pull-up ON	P45 pull-up ON	P44 pull-up ON
1	P47 pull-up OFF	P46 pull-up OFF	P45 pull-up OFF	P44 pull-up OFF

R/W : Readable/Writable
— : Unused
▨ : Initial value

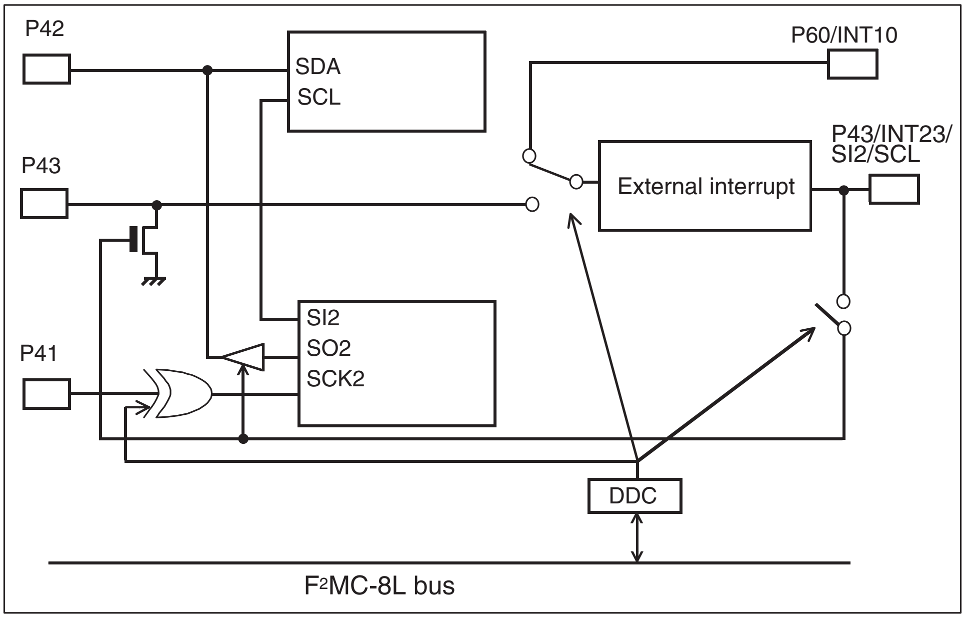
Reference:

Neither P42 nor P43 have internal pull-up resistors. To use them as output pins, external pull-up resistors are required.

■ DDC select register (DDCR)

Figure 4.5-4 "Block diagram of the DDC function" shows the block diagram of the DDC function.

Figure 4.5-4 Block diagram of the DDC function



❍ DDC

This bit is used to select the rising edge or falling edge of serial data output.

Select the source of an external interrupt circuit. Specifying 1 in the DDC bit prohibits serial data output and pulls down the SCL line with an interrupt from P43.

❍ DDC select register (DDCR)

Figure 4.5-5 DDC select register (DDCR)

DDC select register (DDCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0049$_H$	–	–	–	–	–	–	–	DDC	-------0$_B$
								R/W	

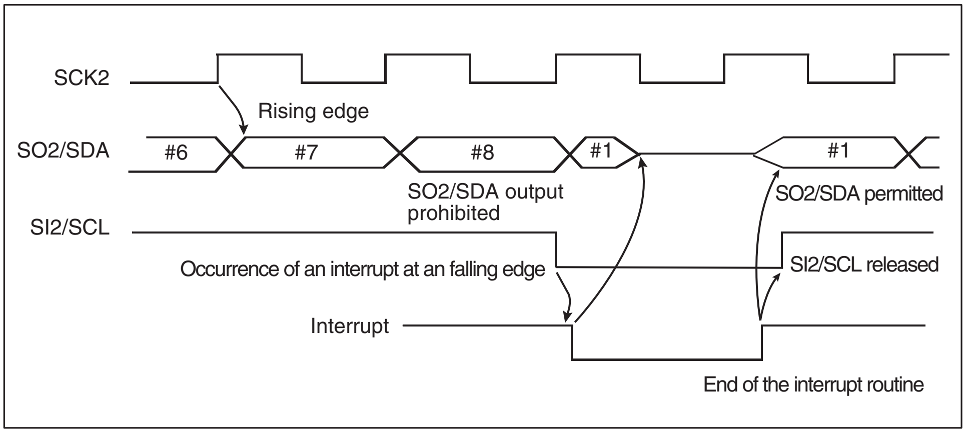
R/W : Readable/Writable
- : Unused

Table 4.5-4 DDC select register (DDCR)

Bit name		Function	
bit1 to bit7	Unused	• Values in the bits are unfixed when they are read. • Writing values in the bits has no effect.	
bit0	DDC	When 0 is specified in the DDC bit:	Serial data outputs falling edges. P60 is used for an external interrupt source.
		When 1 is specified in the DDC bit:	Serial data outputs rising edges. P43 is used for an external interrupt source. When an interrupt takes place, SIO output is prohibited and SCL is pulled down.

❍ **Operation of the DDC function**

Figure 4.5-6 Operation of I^2C and SIO when the DDC function is enabled


SCK2
Rising edge
SO2/SDA
#6
#7
#8
#1
#1
SO2/SDA output prohibited
SO2/SDA permitted
SI2/SCL
Occurrence of an interrupt at an falling edge
SI2/SCL released
Interrupt
End of the interrupt routine

When 1 has been specified in the DDC bit of the DDCR register, serial data is output after a rising edge of SCK2. However, if a Level L signal has been input to the SI2/SCL pin, an external interrupt circuit is activated. In this routine, the interrupt circuit generates an interrupt signal to the CPU. This signal turns on a gate that pulls down the SI2/SCL pin and prohibits SO2 output.

After the interrupt routine ends, the interrupt signal is reset. The connection of the circuits is restored to the previous state.

See Figure 4.5-4 "Block diagram of the DDC function."

4.5.2 Operation of Port 4

This section describes operation of Port 4.

■ Operation of Port 4

❍ **Operation of pins set up for output**

- Specifying 1 to a bit of the DDR4 register sets the corresponding pin for output.
- For an output pin, operation of the output transistor is permitted and data in the output latch is output to the pin.
- Data written to the PDR4 register remains in the output latches and is output to the pins as is.
- Reading the PDR4 register outputs the values of the pins.

❍ **Operation of pins set up for input**

- Specifying 0 to a bit of the DDR4 register sets the corresponding pin for input.
- For an input pin, the output transistor is "OFF" and the pin is set to high impedance.
- Data written to the PDR4 register remains in the output latches but is not output to the pins.
- Reading the PDR4 register outputs the values of the pins.

❍ **Settings for external interrupt input**

- Specifying 0 to DDR4 register bits which correspond to external interrupt input set the pins for input.
- Reading the PDR4 register outputs the values of the pins, irrespective of whether external interrupt input or interrupt request output has been allowed or prohibited.

❍ **Operation on a reset**

- Resetting the CPU initializes the DDR4 register with zeros. Thus, the output transistors are turned "OFF" (all the pins are set for input) and the pins are set to high impedance.
- The PDR4 register is not initialized by a reset. Therefore, to use the pins for output, set output data to the PDR4 register before setting the DDR4 register bits in output mode.

❍ **Operation in stop mode and watch mode**

When Port 4 is put in stop mode or watch mode, it is set to the status specified by the pin status specification bit of the standby control register, irrespective of the values of the DDR4 register.

Table 4.5-5 "Status of the Port 4 pins" shows the status of the Port 4 pins in each mode.

Table 4.5-5 Status of the Port 4 pins

Pin name	Normal operation Main sleep Main stop (SPL=0) Sub-sleep Sub-stop (SPL=0) Watch mode (SPL=0)	Main stop (SPL=1) Sub-stop (SPL=1) Watch mode (SPL=1)	At reset
P40/INT4/EC to P47/INT10/UI2	General-purpose I/O port/ External interrupt input/ Resource I/O	Hi-z (external interrupt input)	Hi-z

SPL: Pin status specification bit of the standby control register (STBC:SPL)
Hi-z: High impedance

Reference:

When pull-up resistors are enabled by using the Port 4 pull-up resistor control register, the states of the pins do not become high impedance but rather become Level H (pulled-up) in stop mode or watch mode (STBC:SPL=1). However, the pins are not pulled up during a reset and their state becomes Hi-z.

Note that this does not apply to P42 and P43.

4.6 Port 5

Port 5 is an N-channel open-drain output port which can also handle analog input. The function of each pin can be switched between analog input and N-channel open-drain output port for each bit. This section mainly describes the functions of the general-purpose I/O port.

This section also describes the configuration, pins, and block diagram of the pins, and registers related to Port 5.

■ Configuration of Port 5

Port 5 consists of the following two types of element:

- Output-only pin (P50/AN0 to P57/AN7)
- Port 5 data register (PDR5)

■ Pins of Port 5

Port 5 has eight N-channel open-drain pins. Do not use them as output-only pins when analog signals are to be input from an A/D converter.

Table 4.6-1 "Pins of Port 5" shows the pins of Port 5.

Table 4.6-1 Pins of Port 5

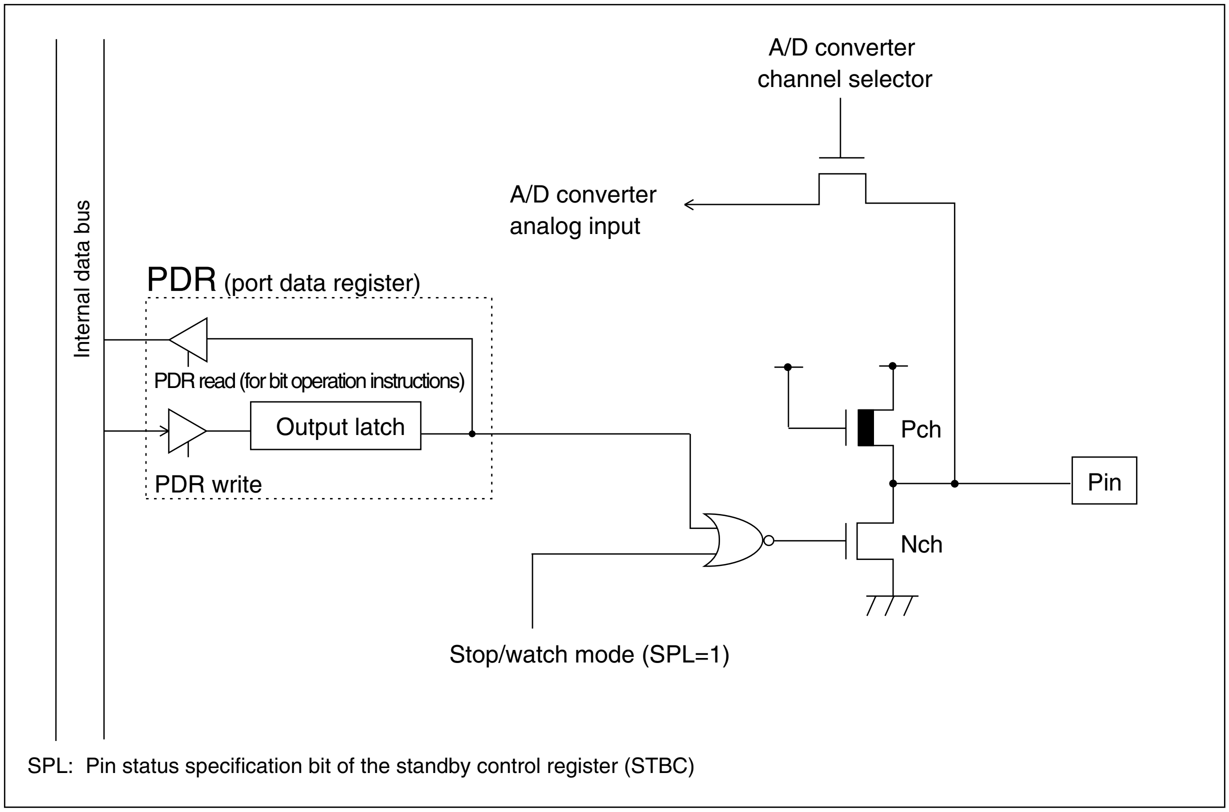
Port name	Pin name	Function	Peripheral device also supported	I/O format		Circuit type
				Input	Output	
Port 5	P50/AN0	P50 N-channel output	AN0 analog input 0	Analog	N-channel open drain output	H
	P51/AN1	P51 N-channel output	AN1 analog input 1			
	P52/AN2	P52 N-channel output	AN2 analog input 2			
	P53/AN3	P53 N-channel output	AN3 analog input 3			
	P54/AN4	P54 N-channel output	AN4 analog input 4			
	P55/AN5	P55 N-channel output	AN5 analog input 5			
	P56/AN6	P56 N-channel output	AN6 analog input 6			
	P57/AN7	P57 N-channel output	AN7 analog input 7			

For details of the circuit types, see Section 1.7 "Explanations of the Pin Functions."

For information about operation of the pins to analog input, see Chapter 15 "A/D Converter."

■ Block diagram of Port 5

Figure 4.6-1 Block diagram of a Port 5 pin



Note:

Do not use the pins for output when analog signals are to be input to Port 5.

■ Register of Port 5

Port 5 has one related register, PDR5.

Each bit of the PDR5 register corresponds to a pin of Port 5.

Table 4.6-2 "Correspondence between bit and pin of the Port 5 register" shows the correspondence between bit and pin of the Port 5 register.

Table 4.6-2 Correspondence between bit and pin of the Port 5 register

Port name	Correspondence between bits and pins of the related registers								
Port 5	PDR5	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Corresponding pin	P57	P56	P55	P54	P53	P52	P51	P50

4.6.1 Register of Port 5 (PDR5)

This section describes the register related to Port 5.

■ Functions of the Port 5 register

❍ Port 5 data register (PDR5)

The PDR5 register indicates values of the output latches. Therefore, the states of the pins cannot be read.

❍ Settings for analog input

To use Port 5 for analog input, write 1 to bits of the PDR5 register. The corresponding output transistors are turned "OFF" and the state of the pins becomes high impedance.

Table 4.6-3 "Functions of the Port 5 register" shows the functions of the Port 5 register.

Table 4.6-3 Functions of the Port 5 register

Register name	Value	Reading	Writing	Read/ Write	Address	Initial value
Port 5 data register (PDR5)	0	The value of the output latch is "0."	Level L is output to the pin. (Set 0 to the output latch to turn on the output transistor.)	R/W	0012_H	11111111_B
	1	The value of the output latch is "1."	The pin is set to high impedance. (Set 1 to the output latch to turn off the output transistor.)			

R/W: Readable/Writable

4.6.2 Operation of Port 5

This section describes operation of Port 5.

■ Operation of Port 5

❍ **Operation of pins set up for output**

- Data written to the PDR5 register remains in the output latches. When the value of an output latch is "0," the output transistor is turned "ON" and Level L is output to the pin. When the value of an output latch is "1," the output transistor is turned "OFF" and state of the pin becomes high impedance. When an output pin is pulled up and if the value of the output latch is "1," it is pulled up.
- Reading the PDR5 register always outputs the values of the output latches.

❍ **Settings for analog input**

- Specify 1 to a bit of the PDR5 register which corresponds to a target analog input pin to turn off the output transistor.
- Reading the PDR5 register always outputs the values of the output latches.

❍ **Operation on a reset**

Resetting the CPU initializes values of the PDR5 register with ones. Thus, all the output transistors are turned "OFF" and the pins are set to high impedance.

❍ **Operation in stop mode and watch mode**

When Port 5 is put in stop mode or watch mode, it is set to the status specified by the pin status specification bit of the standby control register.

Note that input is fixed to prevent leakage due to released input.

Table 4.6-4 "Status of the Port 5 pins" shows the status of the Port 5 pins in each mode.

Table 4.6-4 Status of the Port 5 pins

Pin name	Normal operation Main sleep Main stop (SPL=0) Sub-sleep Sub-stop (SPL=0) Watch mode (SPL=0)	Main stop (SPL=1) Sub-stop (SPL=1) Watch mode (SPL=1)	At reset
P50/AN0 to P57/AN7	General-purpose I/O port/ Analog input	Hi-z	Hi-z

SPL: Pin status specification bit of the standby control register (STBC:SPL)
Hi-z: High impedance

4.7 Port 6

Port 6 is a general-purpose input port which is also used to input external interrupts. This section mainly describes the functions of the general-purpose input port.
This section also describes the configuration, pins, and block diagram of the pins, and registers related to Port 6.

■ Configurations of Port 6

Port 6 consists of the following four types of element:

- General-purpose input pin/External interrupt/Sub-clock related element (P60/INT10 to P62/INT12, P63/INT13/X0A, and P64/X1A)
- Port 6 data register (PDR6)
- Port 6 pull-up register (PURR6)
- DDC select register (DDCR)

■ Pins of Port 6

Port 6 has five CMOS input-only type I/O pins (excluding P64/X1A).

The P60/INT10 to P63/INT13/X0A pins can be used for both input and external interrupt at the same time.

Table 4.7-1 Pins of Port 6

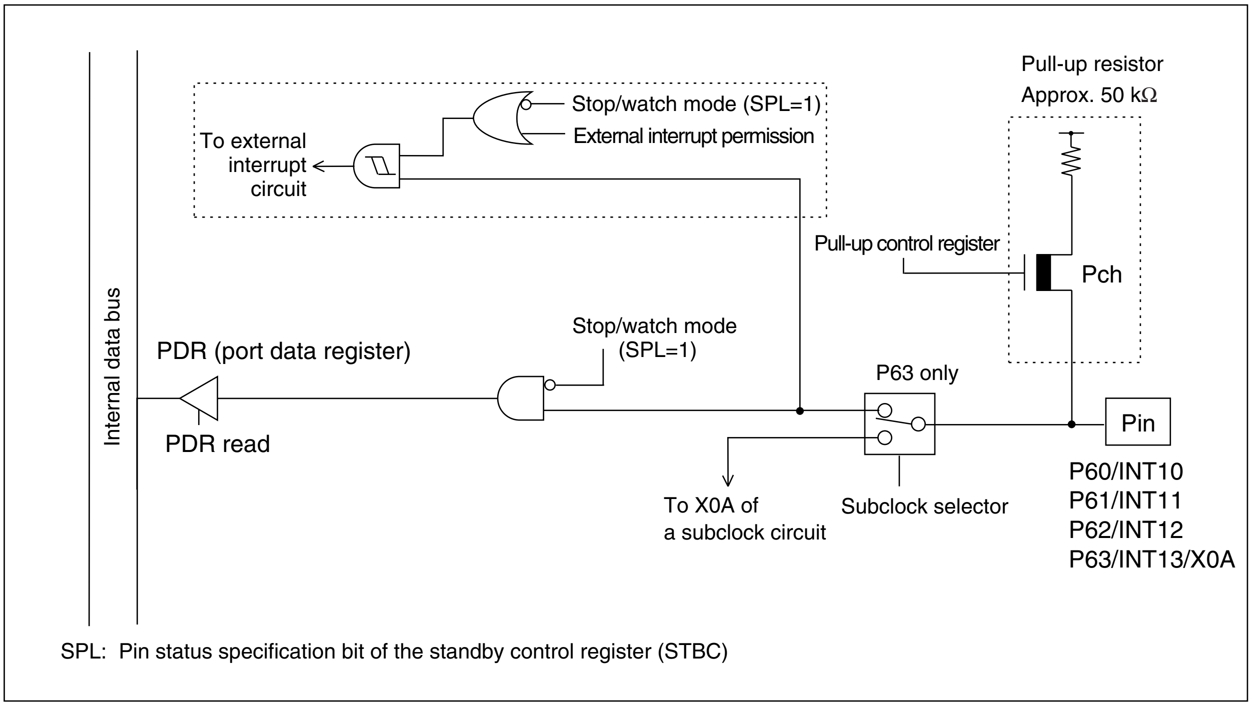
Port name	Pin name	Function	Peripheral device also supported	I/O format		Circuit type
				Input	Output	
Port 6	P60/INT10 to P62/INT12	P60 to P62 General-purpose input	Input external interrupt 1	CMOS (Resources are hysteretic.)	-	I
	P63/INT13/X0A	P63 General-purpose input	External interrupt 1, subclock input pin		-	I/A
	P64/X1A	P64 General-purpose input	Subclock output pin	CMOS	CMOS	J/A

*1 Resources are hysteretic.

For details of the circuit types, see Section 1.7 "Explanations of the Pin Functions."

■ Block diagrams of Port 6

Figure 4.7-1 Block diagram of Port 6 pins (excluding P64/X1A)



SPL: Pin status specification bit of the standby control register (STBC)

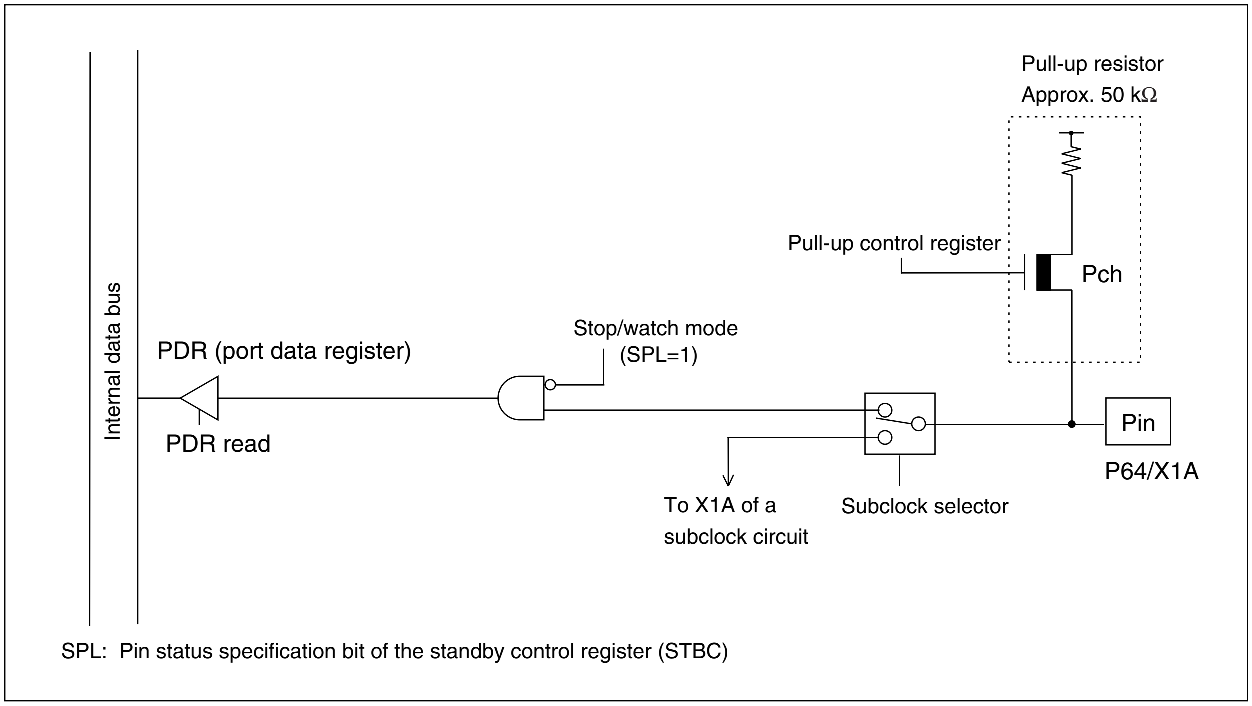
Note:

For a single-clock system device, a port input (P63) or external interrupt input (INT13) can be used.

For a single-clock system device, a pull-up resistor can be set to the port input. However, do not use a pull-up resistor to the external interrupt input. For input levels, ports are CMOS input and resource is CMOS hysteresis input. For a dual-clock system device, a subclock input (X0A) is to be used; thus, no pull-up resistor can be set.

Be sure to disable pull-up resistors by using the pull-up resistor control register.

Figure 4.7-2 Block diagram of P64/X1A



SPL: Pin status specification bit of the standby control register (STBC)

Note:

For a single-clock system device, a port input (P64) can be used. For a single-clock system device, a pull-up resistor can be set to the port input. For a dual-clock system device, a subclock input (X1A) is to be used; thus, no pull-up resistor can be set.

Be sure to disable pull-up resistors by using the pull-up resistor control register.

Port 6 has one related register, PDR6. Each bit of the PDR6 register corresponds to a pin of Port 6.

Table 4.7-2 "Correspondence between bit and pin of the Port 6 register" shows the correspondence between bit and pin of the Port 6 register.

Table 4.7-2 Correspondence between bit and pin of the Port 6 register

Port name	Correspondence between bits and pins of the related registers								
Port 6	PDR6,PURR6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Corresponding pin	-	-	-	P64	P63	P62	P61	P60

4.7.1 Register of Port 6 (PDR6, PURR6, DDCR)

This section describes the register related to Port 6.

■ Functions of the Port 6 register

❍ **Port 6 data register (PDR6)**

The PDR6 register keeps the states of the pins. The pins are input ports; thus, the status of the output latches cannot be read.

Table 4.7-3 "Functions of the Port 6 register" shows the functions of the Port 6 register.

Table 4.7-3 Functions of the Port 6 register

Register name	Value	Reading	Read/Write	Address	Initial value
Port 6 data register (PDR6)	0	The pin is Level L.	R	0013_H	$XXXXXXXX_B$
	1	The pin is Level H.			

R: Read-only
X: Not specified

■ Port 6 pull-up resistor control register (PURR6)

Each Port 6 pin allows a pull-up resistor to be used. Write a pull-up resistor setting to each bit of the Port 6 pull-up resistor control register.

Enabling pull-up resistors in stop mode or watch mode (STBC:SPL=1) using the Port 6 pull-up resistor control register does not set the pins to high impedance but rather to Level H (pulled up). However, during a reset, pull-up of the pins is invalidated and the pins are set to high impedance. (The Port 6 pull-up resistor control registers are initialized by a reset.)

Figure 4.7-3 "Settings of the Port 6 pull-up resistor control register (PURR6)" shows the allowable settings of the Port 6 pull-up resistor control register.

Figure 4.7-3 Settings of the Port 6 pull-up resistor control register (PURR6)

PURR6

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0079H	—	—	—	PUR64	PUR63	PUR62	PUR61	PUR60	---11111B
				R/W	R/W	R/W	R/W	R/W	

	PUR63	PUR62	PUR61	PUR60
0	P63 pull-up ON	P62 pull-up ON	P61 pull-up ON	P60 pull-up ON
1	P63 pull-up OFF	P62 pull-up OFF	P61 pull-up OFF	P60 pull-up OFF

	PUR64
0	P64 pull-up ON
1	P64 pull-up OFF

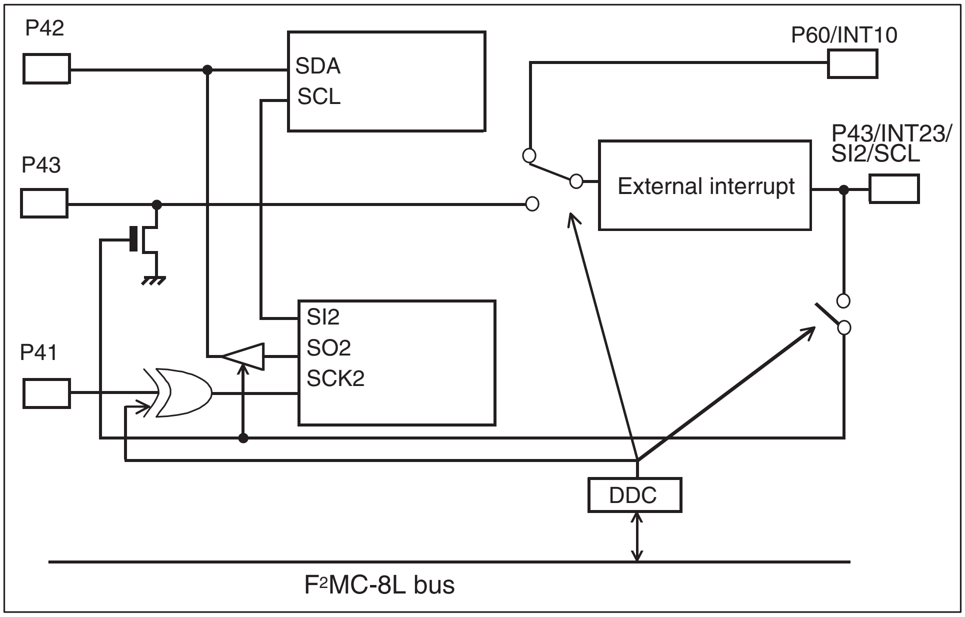
R/W : Readable/Writable

▨ : Initial value

■ DDC select register (DDCR)

Figure 4.7-4 "Block diagram of the DDC function" shows the block diagram of the DDC function.

Figure 4.7-4 Block diagram of the DDC function



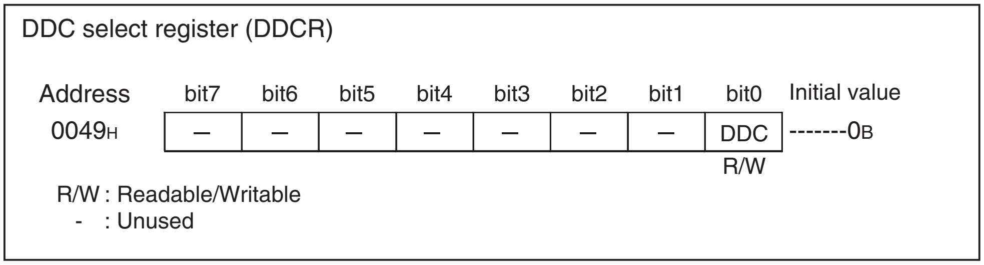
❍ DDC

This bit is used to select the rising edge or falling edge of serial data output.

Select the source of an external interrupt circuit. Specifying 1 in the DDC bit prohibits serial data output and pulls down the SCL line with an interrupt from P43.

❍ DDC select register (DDCR)

Figure 4.7-5 DDC select register (DDCR)



DDC select register (DDCR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0049H	–	–	–	–	–	–	–	DDC	-------0B
								R/W	

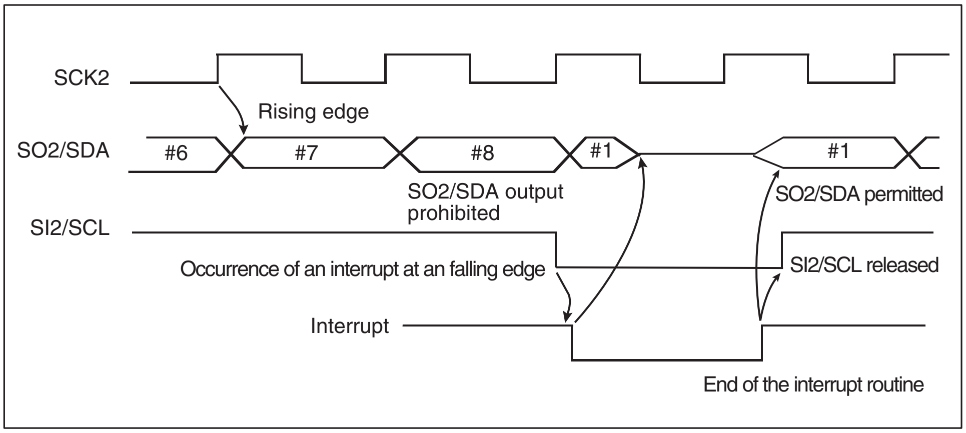
R/W : Readable/Writable
- : Unused

Table 4.7-4 DDC select register (DDCR)

Bit name		Function	
bit1 to bit7	Unused	• Values in the bits are not specified when they are read. • Writing values in the bits has no effect.	
bit0	DDC	When 0 is specified in the DDC bit:	Serial data outputs falling edges. P60 is used for an external interrupt source.
		When 1 is specified in the DDC bit:	Serial data outputs rising edges. P43 is used for an external interrupt source. When an interrupt takes place, SIO output is prohibited and SCL is pulled down.

❍ Operation of the DDC function

Figure 4.7-6 Operation of I^2C and SIO when the DDC function is enabled


SCK2
Rising edge
SO2/SDA
#6
#7
#8
#1
#1
SO2/SDA output prohibited
SO2/SDA permitted
SI2/SCL
Occurrence of an interrupt at an falling edge
SI2/SCL released
Interrupt
End of the interrupt routine

When 1 has been specified in the DDC bit of the DDCR register, serial data is output after a rising edge of SCK2. However, if a Level L signal has been input to the SI2/SCL pin, an external interrupt circuit is activated. In this routine, the interrupt circuit generates an interrupt signal to the CPU. This signal turns on a gate that pulls down the SI2/SCL pin and prohibits SO2 output.

After the interrupt routine ends, the interrupt signal is reset. The connection of the circuits is restored to the previous state.

See Figure 4.7-4 "Block diagram of the DDC function."

4.7.2 Operation of Port 6

This section describes operation of Port 6.

■ Operation of Port 6

❍ **Operation of pins set up for output**

- Reading the PDR6 register outputs the values of the pins.

❍ **Settings for external interrupt input**

- Reading the PDR6 register outputs the values of the pins, irrespective of whether external interrupt input or interrupt request output has been allowed or prohibited.

❍ **Operation on a reset**

- The PDR6 register is not initialized by a reset.

❍ **Operation on a reset**

- The PDR6 register is not initialized by a reset.

❍ **Operation in stop mode and watch mode**

When Port 6 is put in stop mode or watch mode, it is set to the status specified by the pin status specification bit of the standby control register.

Table 4.7-5 "Status of the Port 6 pins" shows the status of the Port 6 pins in each mode.

Table 4.7-5 Status of the Port 6 pins

Pin name	Normal operation Main sleep Main stop (SPL=0) Sub-sleep Sub-stop (SPL=0) Watch mode (SPL=0)	Main stop (SPL=1) Sub-stop (SPL=1) Watch mode (SPL=1)	At reset
P60/INT0 to P64/X1A	General-purpose input port/ External interrupt input	Hi-z (external interrupt input)	Hi-z

SPL: Pin status specification bit of the standby control register (STBC:SPL)
Hi-z: High impedance

Reference:

When all the output register bits are turned "OFF," the states of the pins with pull-up resistors enabled do not become high impedance but rather become Level H (pulled-up).

Note:

When "rising edge," "falling edge," or "both edges" is set in the bit for selecting the edge polarity in stop mode (SPL=1), interrupt input is allowed and not blocked. In this case, keep the electric potential of the pin constant using the pull-up option setting register, external pull-up register, or external-pull-down register.

4.8 Sample I/O Port Program

This section provides a sample of a program with I/O ports used.

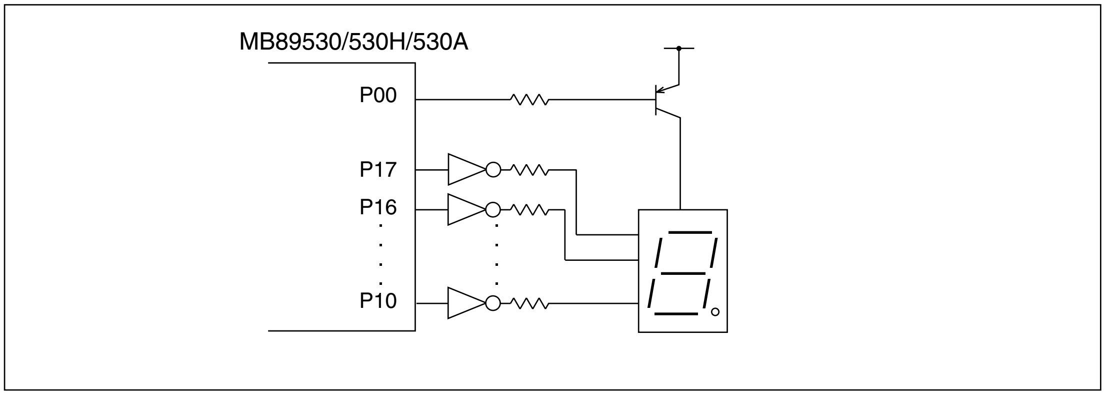
■ Sample I/O port program

❍ Processing specifications

- By using Port 0 and Port 1, light all the 7 segments (8 segments including Dp) of a LED.
- The P00 pin is used for a LED anode common pin, and P10 to P17 pins are used for the segment pins.

Figure 4.8-1 "Example of connection of an 8-segment LED" shows an example of connection of an 8-segment LED.

Figure 4.8-1 Example of connection of an 8-segment LED



❍ Sample coding

```
PDR0   EQU     0000H              ; Address of the port 0 data register
DDR0   EQU     0001H              ; Address of the Port 0 direction register
PDR1   EQU     0002H              ; Address of the Port 1 data register
DDR1   EQU     0003H              ; Address of the Port 1 direction register

;----------Main program-----------------------------------------------------
       CSEG                       ; [CODE SEGMENT]
          :
       CLRB    PDR0:0             ; Set P00 Level L.
       MOV     PDR1, #11111111B   ; Set all the Port 1 pins to Level H.
       MOV     DDR0, #11111111B   ; Set P00 for output; specify #xxxxxxx1B.
       MOV     DDR1, #11111111B   ; Set Port 1 to allow all-bit output.
          :
       ENDS
;--------------------------------------------------------------------------
       END
```

CHAPTER 5 TIMEBASE TIMER

This chapter describes the functions and operations of the timebase timer.

5.1 Overview of the Timebase Timer

The timebase timer is a 21-bit free-run counter that counts up in synchronization with the internal count clock (divide-by-two of the main clock oscillation) and which provides the interval timer function that enables the selection of four types of interval time. The timebase timer also supplies timer output for the oscillation stabilization wait time, an operating clock for the watchdog timer and continuous activation for the A/D converter.

The timebase timer stops its operations in a mode in which the main clock oscillation stops.

■ Interval Timer Function

The interval timer is a function used to generate an interrupt repeatedly at constant intervals.

- An interrupt occurs if the interval timer bit of the counter of the timebase timer overflows.
- The interval timer bit (interval time) can be selected from four kinds of interval time.

Table 5.1-1 "Interval Time of the Timebase Timer" lists the interval time of the timebase timer.

Table 5.1-1 Interval Time of the Timebase Timer

Internal count clock cycle	Interval time
$2/F_{CH}$ (0.2µs)	$2^{13}/F_{CH}$ (Approx. 0.82 ms) $2^{15}/F_{CH}$ (Approx. 3.3 ms) $2^{18}/F_{CH}$ (Approx. 26.2 ms) $2^{22}/F_{CH}$ (Approx. 419.4 ms)

F_{CH}: Main clock oscillation
Values in ( ) shows the interval time when the main clock operates with 10 MHz oscillation.

■ Clock Supply Function

The clock supply function supplies operating clocks for the timer outputs (three types) of the main clock oscillation stabilization wait time and for some peripheral functions.

Table 5.1-2 "Clocks Supplied from the Timebase Timer" lists the cycles of clocks supplied to each peripheral function from the timebase timer.

Table 5.1-2 Clocks Supplied from the Timebase Timer

Clock supply destination	Clock cycle	Remarks
Main clock oscillation stabilization wait time	$2^{14}/F_{CH}$ (Approx. 1.64 ms)	Selected by the oscillation stabilization wait time select bits (SYCC: WT1, WT0) of the system clock control register in the clock controller
	$2^{17}/F_{CH}$ (Approx. 13.10 ms)	
	$2^{18}/F_{CH}$ (Approx. 26.20 ms)	
Watchdog timer	$2^{21}/F_{CH}$ (Approx. 209.7 ms)	Count-up clock of the watchdog timer
A/D converter	$2^{8}/F_{CH}$ (Approx. 25.5 µs)	Clock for continuous start

F_{CH}: Main clock oscillation
Values in ( ) shows the interval time when the main clock operates with 10 MHz oscillation

Note:

The oscillation cycle is unstable just after the oscillation start and the oscillation stabilization wait time serves as a guideline.

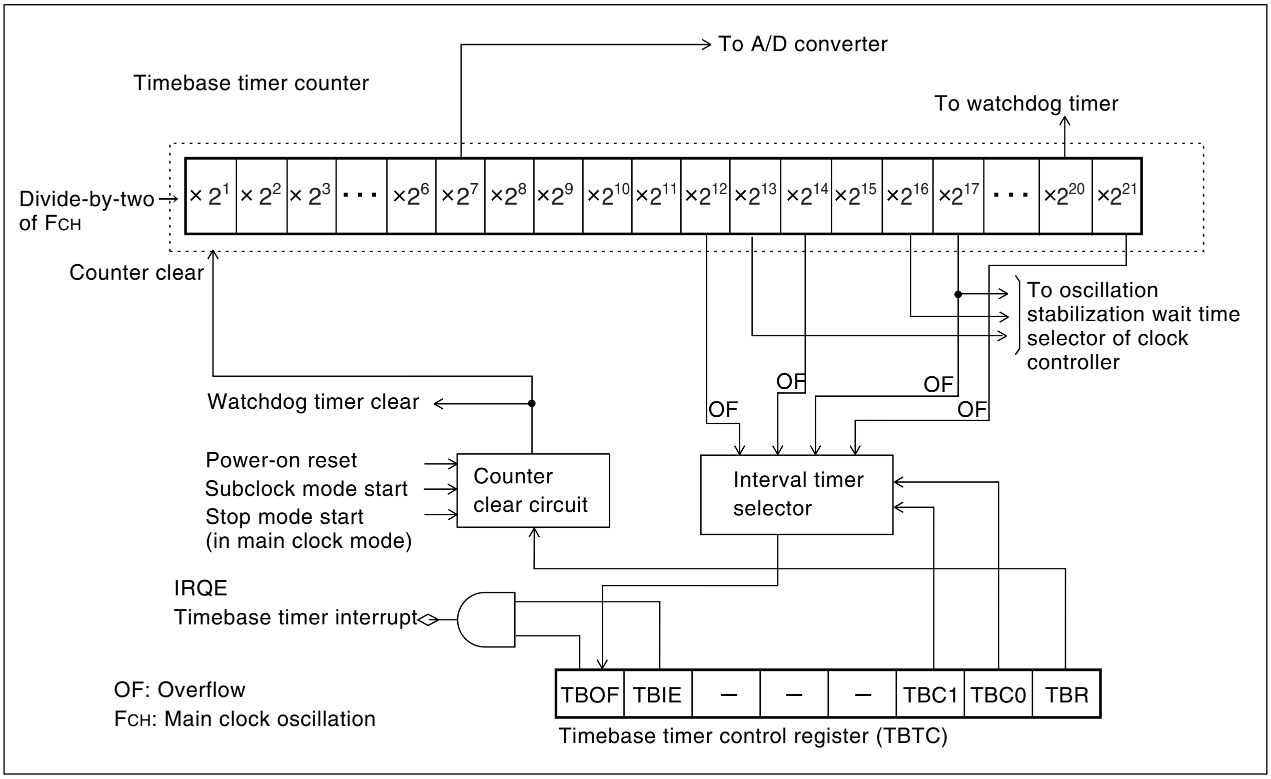
5.2 Configuration of the Timebase Timer

The timebase timer is made up of the following four blocks:

- **Timebase timer counter**
- **Counter clear circuit**
- **Interval timer selector**
- **Timebase timer control register (TBTC)**

■ Block Diagram of the Timebase Timer

Figure 5.2-1 Block Diagram of the Timebase Timer



❍ **Timebase timer counter**

21-bit up-counter using the count clock of divide-by-two of the main clock oscillation. This counter stops operating when the main clock oscillation stops.

❍ **Counter clear circuit**

Clears the counter when, in addition to the setting (TBTC: TBR=0) by the TBTC register, a transition to the main stop mode (STBC: STP=1) or subclock mode (SYCC: SCS=0), or a power-on reset occurs.

- ❍ **Interval timer selector**

  Circuit to select one bit for the interval timer from four bits of the timebase timer counter. The overflow of the selected bit causes an interrupt.

- ❍ **Timebase timer control register (TBTC)**

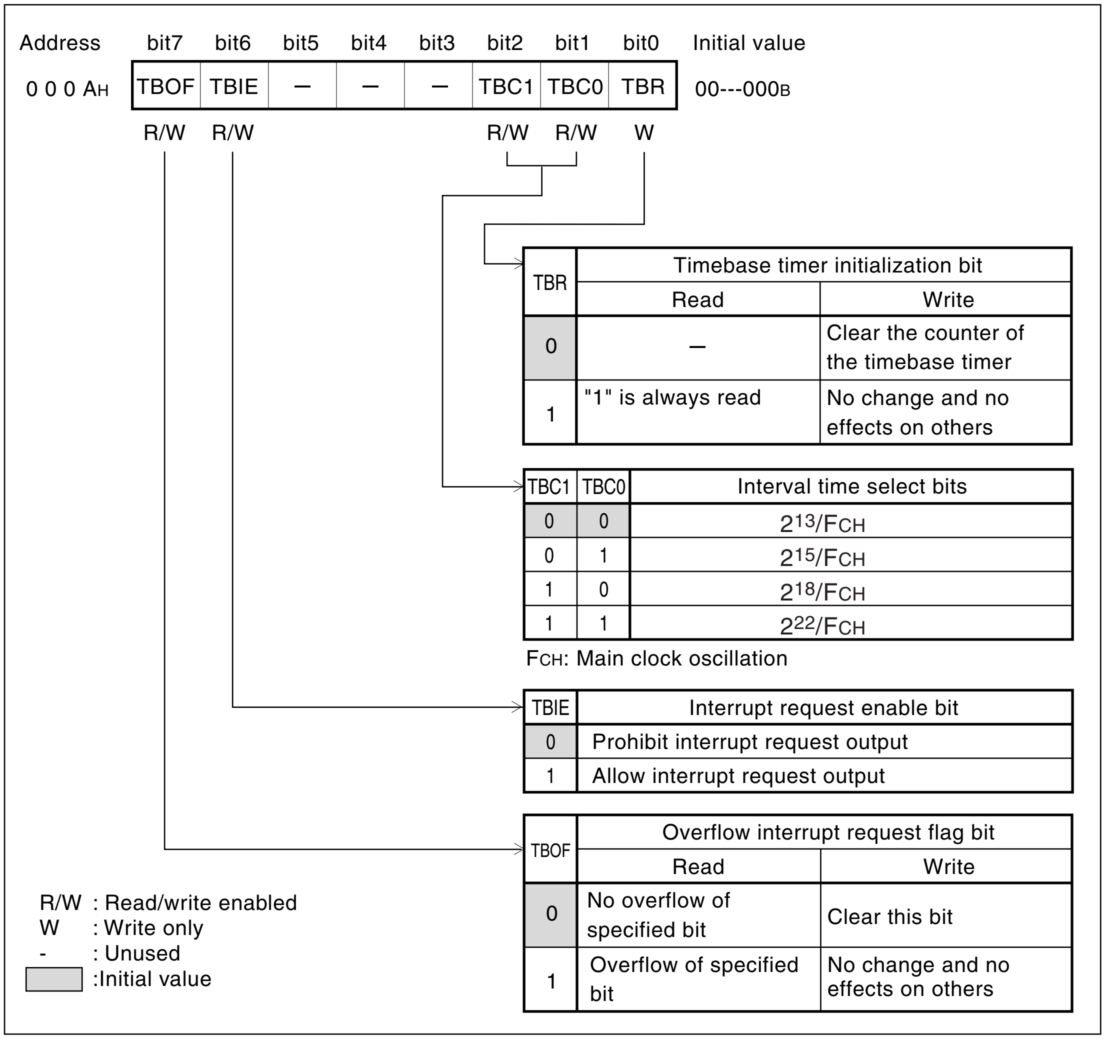
  This register is used to select the interval time, clear the counter, control interrupts, and check the states.

5.3 Timebase Timer Control Register (TBTC)

The timebase timer control register (TBTC) is used to select the interval time, clear the counter, control interrupts, and check the state.

■ Timebase Timer Control Register (TBTC)

Figure 5.3-1 Timebase Timer Control Register (TBTC)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 0 A$_H$	TBOF	TBIE	—	—	—	TBC1	TBC0	TBR	00---000$_B$
	R/W	R/W				R/W	R/W	W	

TBR	Timebase timer initialization bit	
	Read	Write
0	—	Clear the counter of the timebase timer
1	"1" is always read	No change and no effects on others

TBC1	TBC0	Interval time select bits
0	0	$2^{13}/F_{CH}$
0	1	$2^{15}/F_{CH}$
1	0	$2^{18}/F_{CH}$
1	1	$2^{22}/F_{CH}$

F_{CH}: Main clock oscillation

TBIE	Interrupt request enable bit
0	Prohibit interrupt request output
1	Allow interrupt request output

TBOF	Overflow interrupt request flag bit	
	Read	Write
0	No overflow of specified bit	Clear this bit
1	Overflow of specified bit	No change and no effects on others

R/W : Read/write enabled
W : Write only
- : Unused
▨ :Initial value

Table 5.3-1 Explanation of Functions of Each Bit of the Timebase Timer Control Register (TBTC)

Bit name		Function
Bit 7	TBOF: Overflow interrupt request flag bit	• This bit is set to "1" if the specified bit of the counter of the timebase timer overflows. • If both this bit and the interrupt enable bit (TBIE) are "1", an interrupt request is output. • If "0" is written into this bit, the counter is cleared. If "1" is written, no change occurs and operations are not affected.
Bit 6	TBIE: Interrupt request enable bit	• Bit to allow/prohibit interrupt request output to CPU. If both this bit and the overflow interrupt request flag bit (TBOF) are "1", an interrupt request is output.
Bit 5 Bit 4 Bit 3	Unused bits	• The read value is undefined. • Writing has no effect on operation.
Bit 2 Bit 1	TBC1, TBC0: Interval time select bit	• Bits to select the interval timer cycle • Bits for the interval timer of the counter of the timebase timer are specified. • Four kinds of interval time can be selected.
Bit 0	TBR: Timebase timer initialization bit	• Bit to clear the counter of the timebase timer • If "0" is written into this bit, the counter is cleared to "000000_H". If "1" is written, no change occurs and operations are not affected. **Reference:** "1" is always read.

5.4 Timebase Timer Interrupt

As an interrupt source of the timebase timer, an overflow of the specified bit of the timebase timer counter is available (interval timer function).

■ Interrupt when the Interval Timer Function is Active

If an overflow of the selected interval timer bit occurs after the counter is counted up by the internal count clock, the overflow interrupt request flag bit (TBTC: TBOF) is set to "1". At this time, if the interrupt request enable bit is set (TBTC: TBIE=1), an interrupt request to CPU (IRQE) is generated. Clear the interrupt request by writing "0" into the TBOF bit using an interrupt processing routine. The TBOF bit is set whenever an overflow of the specified bit occurs regardless of the value of the TBIE bit.

Note:

To allow interrupt request output (TBTC: TBIE=1) after releasing a reset, clear (TBTC: TBOF=0) the TBOF bit at the same time.

Reference:

If the TBIE bit is changed from prohibition to permission (0 --> 1) when the TBOF bit is "1", an interrupt request is issued immediately.

If the counter clear (TBTC: TBR=0) and an overflow of the selected bit occur at the same time, the TBOF bit is not set.

■ Oscillation Stabilization Wait Time and Timebase Timer Interrupts

If interval time shorter than the oscillation stabilization wait time of the main clock is set, an interval interrupt request (TBTC: TBOF=1) of the timebase timer is generated when the operation in main clock mode starts. In this case, prohibit (TBTC: TBIE=0) interrupts of the timebase timer when making a transition to a mode in which the oscillation of the main clock stops (main stop and subclock modes).

■ Register and Vector Table Related to the Timebase Timer Interrupts

Table 5.4-1 Register and Vector Table Related to the Timebase Timer Interrupts

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Bit to be set		Upper	Lower
IRQE	ILR4 (007E$_H$)	LE1 (bit 5)	LE0 (bit 6)	FFDE$_H$	FFDF$_H$

For the interrupt operations, see Section 3.4.2 "Interrupt Processing".

5.5 Operation of the Timebase Timer

The timebase timer provides the interval timer function and supplies the clock to part of the peripheral functions.

■ Operation of the Interval Timer Function (Timebase Timer)

The setting in Figure 5.5-1 "Setting of the Interval Timer Function" is required for the operation of the interval timer function.

Figure 5.5-1 Setting of the Interval Timer Function

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TBTC	TBOF	TBIE	–	–	–	TBC1	TBC0	TBR
	0	1				◎	◎	0

◎ : Bit used
1 : 1 is set
0 : 0 is set

The counter of the timebase timer continues to count up provided the main clock oscillates in synchronization with the internal count clock (divide-by-two of the main clock oscillation).

If the counter is cleared (TBR=0), it starts counting up from "0". If an overflow of the bit for the interval timer occurs, "1" is set to the overflow interrupt request flag bit (TBOF). That is, starting when clearing occurs, an interrupt request is generated at regular intervals of the selected time.

■ Operation of the Clock Supply Function

The timebase timer is also used as a timer to generate the oscillation stabilization wait time of the main clock. Counting of the oscillation stabilization wait time starts when the counter of the timebase timer is cleared and ends when an overflow of the bit for oscillation stabilization wait time occurs. Three kinds of oscillation stabilization wait time can be selected by the setting of the oscillation stabilization wait time select bits (SYCC: WT1, WT0) of the system clock control register.

The timebase timer supplies the clock to the watchdog timer and A/D converter, and LCD controller/driver. When the counter of the timebase timer is cleared, operations of the continuous activation cycles of the A/D converter are affected. If the timebase timer counter is cleared by changing to main clock operation (STBC:STP = 1) or by changing to subclock operation (STCC:SCS = 0) when timebase timer output is selected (WDTC:CS = 0), the counter of the watchdog timer is cleared at the same time.

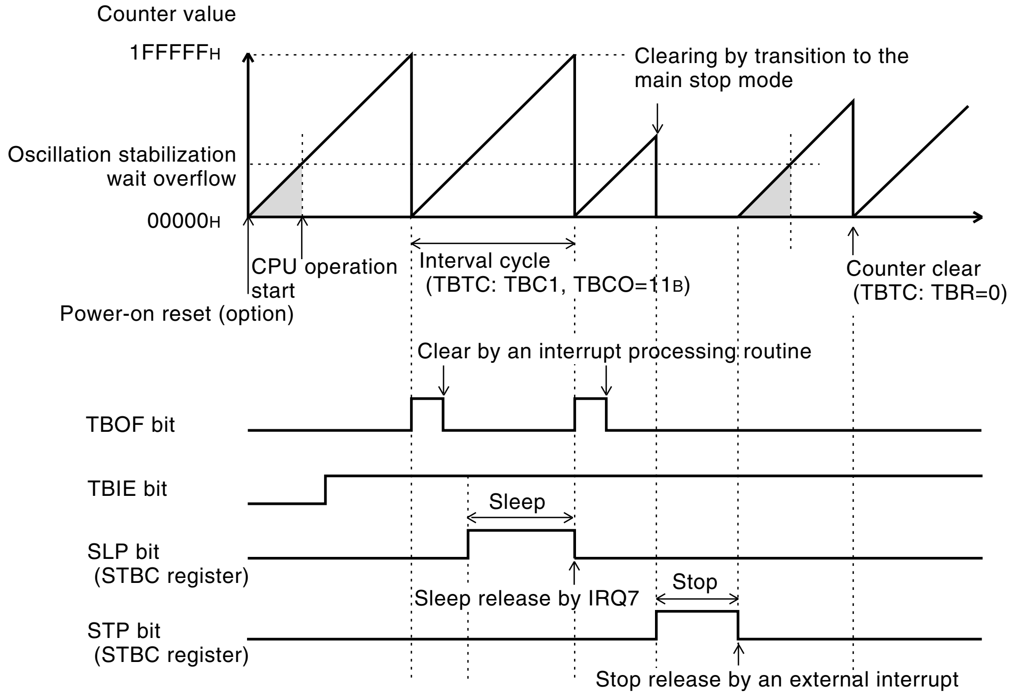
■ Operations of the Time-based Timer

Figure 5.5-2 "Operations of the Timebase Timer" shows the operations in the following states:

- When a power-on reset occurs
- When a transition to the sleep mode occurs during operation of the interval timer function in main clock mode
- When a transition to the main stop mode occurs
- When the counter clear is requested

In subclock mode and main stop mode, the timebase timer is cleared and its operation is stopped. When returning from the subclock mode or main stop mode, the oscillation stabilization wait time is counted by the timebase timer.

Figure 5.5-2 Operations of the Timebase Timer


Counter value
1FFFFFH
Oscillation stabilization wait overflow
00000H
Clearing by transition to the main stop mode
CPU operation start
Power-on reset (option)
Interval cycle (TBTC: TBC1, TBCO=11B)
Counter clear (TBTC: TBR=0)
Clear by an interrupt processing routine
TBOF bit
TBIE bit
Sleep
SLP bit (STBC register)
Sleep release by IRQ7
Stop
STP bit (STBC register)
Stop release by an external interrupt

When "11B" is set to the interval time select bits (TBTC: TBC1, TBC0) of the timebase timer control register ($2^{22}/F_{CH}$).

▨ : Indicates the oscillation stabilization wait time

5.6 Notes on Using the Timebase Timer

The following describes the precautions to take when using the timebase timer.

■ Notes on Using the Timebase Timer

❍ Precautions when setting the timebase timer with programs

Because it is impossible to return from interrupt processing if the interrupt request flag bit (TBTC: TBOF) is "1" and the interrupt request enable bit (TBTC: TBIE=1) is allowed, the TBOF bit must be cleared.

❍ Clearing the timebase timer

The timebase timer is cleared, in addition to clearing by the timebase timer initialization bit (TBTC: TBR=0), when the oscillation stabilization wait time of the main clock is required. If the timebase timer is selected (WDTC: CS=0) as the count clock of the watchdog timer, the watchdog timer is cleared when the timebase timer is cleared.

❍ Using the timebase timer as a timer for the oscillation stabilization wait time

Since the main clock oscillation is stopped when the power is turned on or is in main stop mode or subclock mode, the oscillator takes the oscillation stabilization wait time of the main clock. The appropriate oscillation stabilization wait time must be selected according to the type of resonator connected to the oscillator (clock generator) of the main clock.

For details, see Section 3.6.5 "Oscillation Stabilization Wait Time"

❍ Precautions for peripheral functions to which the clock is supplied from the timebase timer

In a mode in which the main clock oscillation stops, the counter is cleared and the timebase timer stops its operation. In addition, if the counter of the timebase timer is cleared, the "H" level period of the clock supplied by the timebase timer may become shorter or its "L" level period may become longer by half a cycle at the most, since the clock is output from the initial state. The clock for the watchdog timer is also output from the initial state. However, the watchdog timer counter is cleared at the same time and the watchdog timer operates in normal cycles.

5.7 Program Example of the Timebase Timer

The following shows a program example of the timebase timer.

■ Program Example of the Timebase Timer

❍ Processing specifications

Generate the interval timer interrupt of $2^{18}/F_{CH}$ (F_{CH}: main clock oscillation) repeatedly. The interval time at this time is about 26 ms (for 10 MHz operations).

❍ Coding example

```
TBTC    EQU     0000AH           ; Address of the timebase timer control
TBOF    EQU     TBTC:7           ; Definition of overflow interrupt request
                                   flag bit
ILR4    EQU     007EH            ; Address of the interrupt level setting
                                   register 4
INT_V   DSEG    ABS              ; [DATA SEGMENT]
        ORG     0FFDEH
IRQE    DW      WARI             ; Setting interrupt vector
INT_V   ENDS
;----Main program-----------------------------------------------------------
        CSEG                     ; [CODE SEGMENT]
                                 ; Stack pointer (SP) and other are
                                   assumed to have been initialized
          :
        CLRI                     ; Interrupt disable
        MOV     ILR4,#11011111B  ; Setting interrupt level(level 1)
        MOV     TBTC,#01000100B  ; Clearing the overflow interrupt request
                                   flag, enabling the interrupt request
                                   output, selecting $2^{18}/F_{CH}$, and
                                   clearing the timebase timer
        SETI                     ; Interrupt enable
           :
;----Interrupt program------------------------------------------------------
WARI    CLRB    TBOF             ; Clearing interrupt request flag
        PUSHW   A
        XCHW    A,T
        PUSHW   A
          :
        User processing
          :
        POPW    A
        XCHW    A,T
        POPW    A
        RETI
        ENDS
; --------------------------------------------------------------------------
        END
```

CHAPTER 6 WATCHDOG TIMER

This chapter describes the functions and operations of the watchdog timer.

6.1 Overview of the Watchdog Timer

The watchdog timer is a 1-bit counter which accepts the output of either the timebase timer operating with the main clock or the watch prescaler operating with the subclock as the count clock. If the watchdog timer is not cleared for a specified period of time after activation, CPU is reset.

■ Watchdog Timer Function

The watchdog timer is a counter against program runaway. Once the watchdog timer is activated, it is necessary to continue clearing it periodically within a specified period of time. If the watchdog timer is not cleared for a specified period of time, for example, because the program slips into an endless loop, a watchdog reset of the four instruction cycles is generated to CPU.

As the count clock of the watchdog timer, the output of either the timebase timer or watch prescaler can be selected.

The interval time of the watchdog timer is as listed in Table 6.1-1 "Interval Time of the Watchdog Timer". If the watchdog timer is not cleared, a watchdog reset occurs between the minimum and maximum times. Clear the counter within the minimum time of this table.

Table 6.1-1 Interval Time of the Watchdog Timer

	Count clock	
	Timebase timer output (for main clock oscillation 10 MHz)	watch prescaler output (for subclock oscillation 32. 768 kHz)
Minimum time	Approx. 209.7 ms(*1)	500 ms(*2)
maximum time	Approx. 419.4 ms	1000 ms

*1: Divide-by-two of the main clock oscillation (F_{CH}) x count of the timebase timer (2^{21})
*2: Cycle of the subclock oscillation (F_{CL}) x count of the watch prescaler (2^{14})

For the minimum and maximum times of the interval time of the watchdog timer, see Section 6.4 "Operation of the Watchdog Timer"

Note:

The watchdog timer counter is cleared as soon as the timebase timer is cleared (TBTC: TBR = 0) when the timebase timer output is selected for the count clock. The watchdog timer counter is also cleared as soon as the watch prescaler is cleared (WPCR: WCLR = 0) when the watch prescaler is selected for the count clock. Thus, if the counter used as the count clock (timebase timer or watch prescaler) is cleared repeatedly within the watchdog timer interval time, it does not function as the watchdog timer.

Reference:

If a transition to the sleep mode, stop mode, or watch mode occurs, the counter of the watchdog timer is cleared and will not operate until normal operation (RUN state) is resumed.

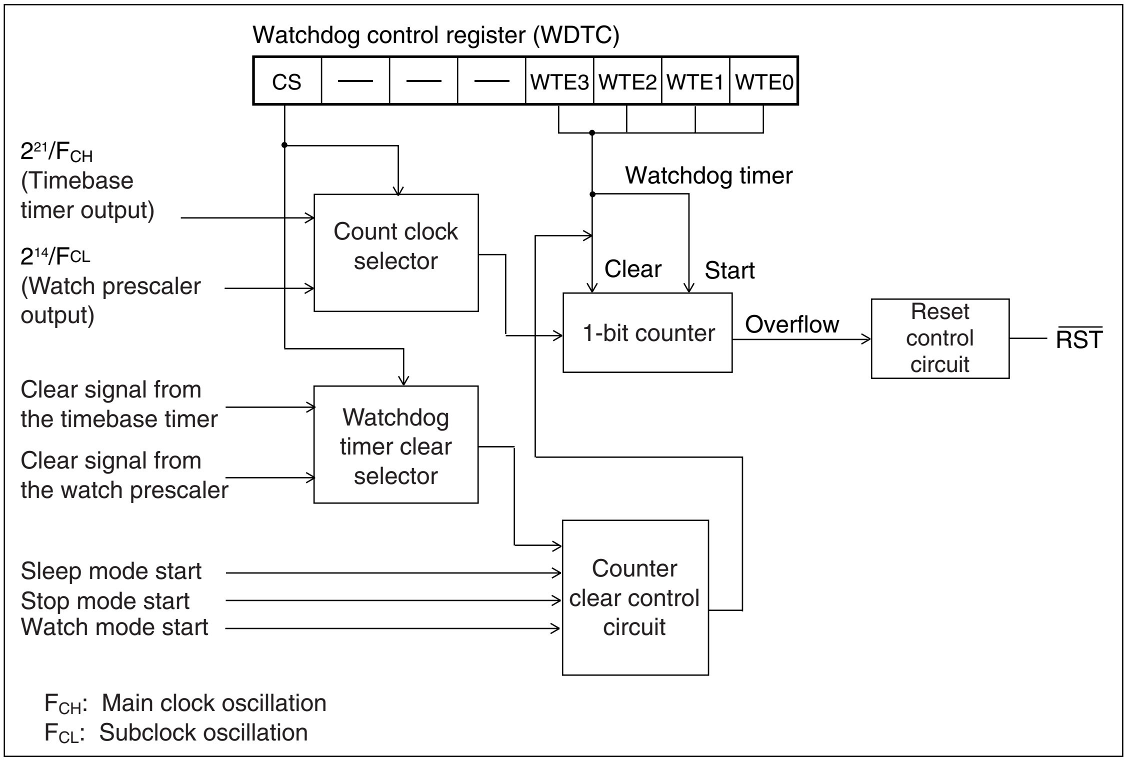
6.2 Configuration of the Watchdog Timer

The watchdog timer is made up of the following six blocks:

- **Count clock selector**
- **Watchdog timer counter**
- **Reset control circuit**
- **Watchdog timer clear selector**
- **Counter clear control circuit**
- **Watchdog timer control register (WDTC)**

■ Block Diagram of the Watchdog Timer

Figure 6.2-1 Block Diagram of the Watchdog Timer


Watchdog control register (WDTC)
CS
—
—
—
WTE3
WTE2
WTE1
WTE0
$2^{21}/F_{CH}$ (Timebase timer output)
$2^{14}/F_{CL}$ (Watch prescaler output)
Count clock selector
Watchdog timer
Clear
Start
1-bit counter
Overflow
Reset control circuit
$\overline{RST}$
Clear signal from the timebase timer
Clear signal from the watch prescaler
Watchdog timer clear selector
Sleep mode start
Stop mode start
Watch mode start
Counter clear control circuit
F_{CH}: Main clock oscillation
F_{CL}: Subclock oscillation

❍ Count clock selector

The count clock selector selects the count clock of the watchdog timer counter. As the count clock, the output of either the timebase timer or the watch prescaler can be selected.

❍ Watchdog timer counter (1-bit counter)

The watchdog timer counter is a 1-bit counter whose count clock is the output of either the timebase timer or the watch prescaler.

❍ **Reset control circuit**

The reset control circuit generates a reset signal to CPU when an overflow of the watchdog timer counter occurs.

❍ **Watchdog timer clear selector**

The watchdog timer clear selector selects the watchdog timer clear signal from the timebase timer or watch prescaler simultaneously with the count clock selector.

❍ **Counter clear control circuit**

The counter clear control circuit controls the watchdog timer counter clearing and operation stop.

❍ **Watchdog timer control register (WDTC)**

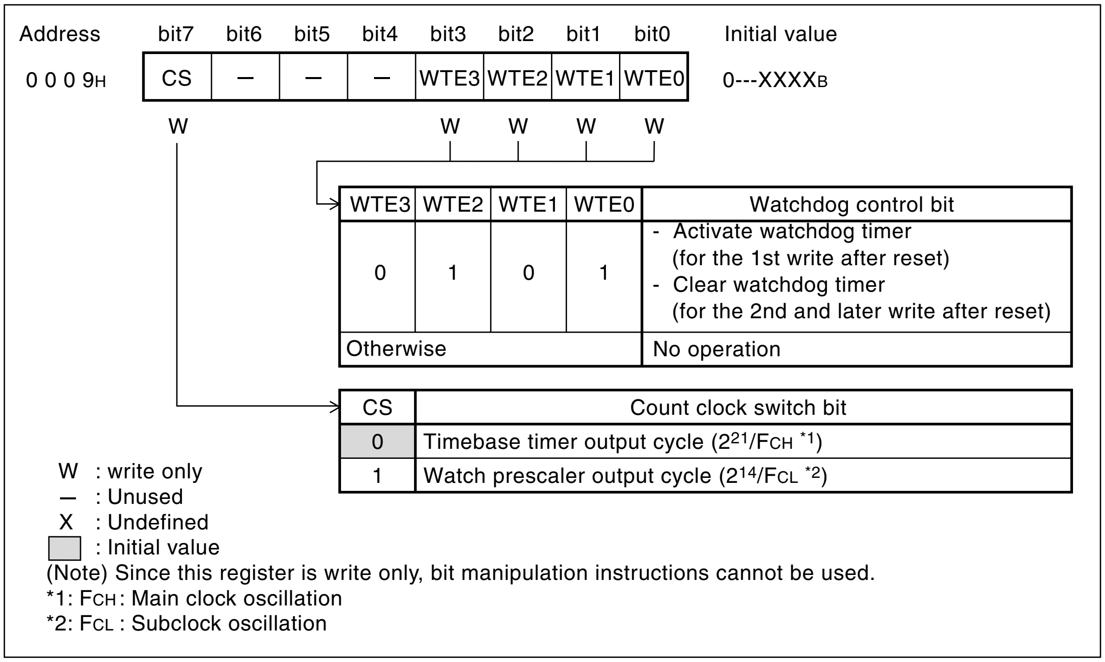
The watchdog timer control register is used to select the count clock and activate/clear the watchdog timer counter. Since this register is write only, bit manipulation instructions cannot be used.

6.3 Watchdog Timer Control Register (WDTC)

The watchdog timer Control Register (WDTC) is used to activate/clear the watchdog timer.

■ Watchdog Timer Control Register (WDTC)

Figure 6.3-1 Watchdog Timer Control Register (WDTC)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 0 9H	CS	—	—	—	WTE3	WTE2	WTE1	WTE0	0---XXXXB
	W				W	W	W	W	

WTE3	WTE2	WTE1	WTE0	Watchdog control bit
0	1	0	1	- Activate watchdog timer (for the 1st write after reset) - Clear watchdog timer (for the 2nd and later write after reset)
Otherwise				No operation

CS	Count clock switch bit
0	Timebase timer output cycle ($2^{21}/F_{CH}$ *1)
1	Watch prescaler output cycle ($2^{14}/F_{CL}$ *2)

W : write only
— : Unused
X : Undefined
▨ : Initial value

(Note) Since this register is write only, bit manipulation instructions cannot be used.
*1: FCH : Main clock oscillation
*2: FCL : Subclock oscillation

Table 6.3-1 Explanation of the Functions of Each Bit of the Watchdog Timer Control Register (WDTC)

Bit name		Function
Bit 7	CS: Count clock switch bit	• Select the count clock of the watchdog timer when activating the watchdog timer. • As the count clock, the output of either the timebase timer or the watch prescaler can be selected. **Note:** • To use the subclock mode, select the output of the watch prescaler. • Select the count clock simultaneously with activation of the watchdog timer and do not change it after the activation. • Bit manipulation instructions cannot be used.
Bit 6 Bit 5 Bit 4	Unused bist	• The read value is undefined. • Writing has no effect on operation.
Bit 3 Bit 2 Bit 1 Bit 0	WTE3, WTE2, WTE1, WTE0: Watchdog control bit	• If "0101_B" is written into these bits, the watchdog timer is activated (the 1st write after reset) or cleared (the 2nd or later write after reset). • Writing anything other than "0101_B" does not affect operations. **Note:** "1111_B" is read. Bit manipulation instructions cannot be used.

6.4 Operation of the Watchdog Timer

The watchdog timer generates a watchdog reset when the watchdog timer counter overflows.

■ Operation of the Watchdog Timer

❍ Activating the watchdog timer

- The watchdog timer can be activated by writing the 1st "0101_B" into the watchdog control bits (WDTC: WTE3 to 0) of the watchdog timer control register after a reset. At this time, specify the count clock switch bit (WDTC: CS) simultaneously.
- A watchdog timer that is activated can only be stopped by a reset.

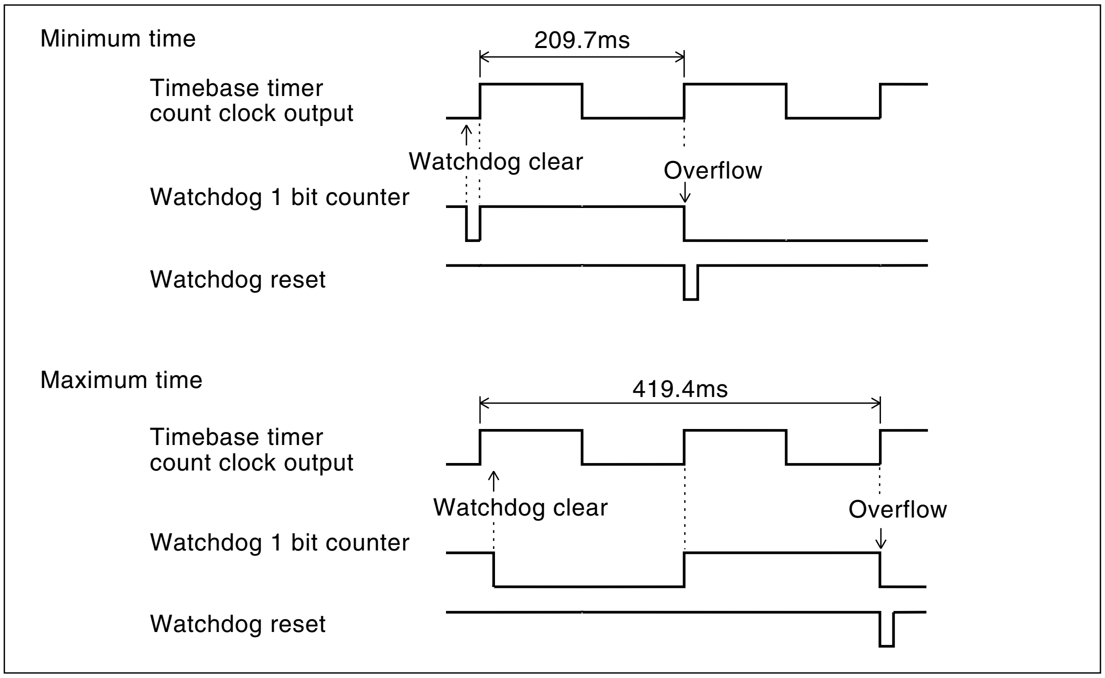
❍ Clearing the watchdog timer

- The counter of the watchdog timer can be cleared by writing the 2nd or subsequent "0101_B" into the watchdog control bits (WDTC: WTE3 to 0) of the watchdog timer control register after a reset.
- If the counter is not cleared within the interval time of the watchdog timer, an overflow of the counter occurs and an internal reset signal of the four instruction cycles is generated.

❍ Watchdog timer interval time

The interval time is changed by the timing of clearing the watchdog timer. Figure 6.4-1 "Watchdog Timer Clearing and Interval Time" shows the relations between the clearing timing of the watchdog timer and the interval time when the output of the timebase timer is selected as the count clock (if the main clock oscillation is 10 MHz).

Figure 6.4-1 Watchdog Timer Clearing and Interval Time



6.5 Notes on Using the Watchdog Timer

The following describes the precautions to take when using the watchdog timer.

■ Notes on Using the Watchdog Timer

❍ Stopping the watchdog timer

A watchdog timer that is activated can only be stopped by a reset.

❍ Selecting the count clock

The count clock switch bit (WDTC: CS) can be rewritten only if "0101_B" is written into the watchdog control bits (WDTC: WTE3 to 0) when the watchdog timer is activated. Thus, a write operation by bit manipulation instructions is not possible. Do not change the settings after activation.

Since the main clock oscillation stops in subclock mode, the timebase timer does not operate.

To enable operation of the watchdog timer in subclock mode, the watch prescaler (WDTC: CS=1) must be selected as the count clock in advance.

❍ Clearing the watchdog timer

- If the counter (timebase timer or watch prescaler) used as the count clock of the watchdog timer is cleared, the counter of the watchdog timer is cleared at the same time.
- If a transition to the sleep mode, stop mode, or watch mode occurs, the counter of the watchdog timer is cleared.

❍ Precautions when creating a program

When creating a program in which the watchdog timer is cleared repeatedly in the main loop, the processing time of the main loop including interrupt processing must be equal to or less than the minimum watchdog timer interval time.

❍ Operations in subclock mode

If a watchdog reset occurs in subclock mode, operation starts in main clock mode after taking the oscillation stabilization wait time. At this time, a reset signal is output during oscillation stabilization wait time.

6.6 Program Example of the Watchdog Timer

The following shows a program example in which the watchdog timer is used.

■ Program Example of the Watchdog Timer

❍ **Processing specifications**

- Select the watch prescaler just after starting the program to activate the watchdog timer.
- Clear the watchdog timer each time in a loop of the main program.
- The main loop must make a round in less than the interval minimum time (about 209.7 ms for 10 MHz operation), including the interrupt processing time, of the watchdog timer.

❍ Coding example (comply with Softune V1)

```
WDTC     EQU     00009H         ; Address of the watchdog timer control register
WDT_CLR  EQU     10000101B
VECT     DSEG    ABS            ; [DATA SEGMENT]
         ORG     0FFFEH
RST_V    DW      PROG           ; Setting reset vector
VECT     ENDS
;----Main program----------------------------------------------------------------
         CSEG                   ; [CODE SEGMENT]
PROG                            ; Initialization routine for reset
         MOVW     SP,#0280H     ; Setting initial value of stack pointer
                                  (for interrupt)
          :
         Initializing interrupt or other peripheral functions
          :
INIT     MOV      WDTC,#WDT_CLR  ; Activating watchdog timer Selection of the
                                   watch prescaler as the count clock
          :
MAIN     MOV      WDTC,#WDT_CLR  ; Clearing watchdog timer
          :
         User processing (interrupt may occur in this processing.)
          :
         JMP      MAIN           ; Ensure that the time necessary for running
                                   the loop is shorter than the minimum time
                                   interval of the watchdog timer.
         ENDS
;--------------------------------------------------------------------------------
         END
```

CHAPTER 7 WATCH PRESCALER

This chapter describes the functions and operations of the watch prescaler.

7.1 Overview of the Watch Prescaler

The watch prescaler is a 17-bit free-run counter that counts up in synchronization with the subclock generated in the clock generator and has an interval timer function that provides for the selection of six kinds of interval time.
The watch prescaler also supplies the timer output of subclock oscillation stabilization wait time and the operating clock of the watchdog and other timers.

■ Interval Timer Function (Watch Interrupt)

The interval timer function is a function used to generate an interrupt repeatedly at regular intervals using the subclock as the count clock.

- An interrupt is generated by divide-by output for the interval timer of the watch prescaler.
- Six kinds of divide-by output (interval time) for the interval timer can be selected.
- The counter of the watch prescaler can be cleared.

Table 7.1-1 "Interval Time of the Watch Prescaler" lists the interval time of the watch prescaler.

Table 7.1-1 Interval Time of the Watch Prescaler

Subclock cycle	Interval time
$1/F_{CL}$ (Approx. 30.5 µs)	$2^{10}/F_{CL}$ (31.25 ms)
	$2^{13}/F_{CL}$ (0.25 s)
	$2^{14}/F_{CL}$ (0.50 s)
	$2^{15}/F_{CL}$ (1.00 s)
	$2^{16}/F_{CL}$ (2.00 s)
	$2^{17}/F_{CL}$ (4.00 s)

F_{CL}: Subclock oscillation
Values in ( ) represent the interval time when the subclock oscillation is operating at 32.768 kHz.

Note:

The watch prescaler cannot be used when the single clock system is selected for option setting.

■ Clock Supply Function

The clock supply function of the watch prescaler is a function used to supply the timer output (one) for oscillation stabilization wait time of the subclock and the clock for the watchdog timer.

Table 7.1-2 "Clocks Supplied from the Watch Prescaler" lists the clock cycles supplied to each peripheral function from the watch prescaler.

Table 7.1-2 Clocks Supplied from the Watch Prescaler

Subclock supply destination	Subclock cycle	Remarks
Subclock oscillation stabilization wait time	$2^{15}/F_{CL}$ (1.00 s)	Do not make a transition to the subclock mode during oscillation stabilization wait time
Watchdog timer	$2^{14}/F_{CL}$ (0.50 s)	Count-up clock of the watchdog timer

F_{CL}: Subclock oscillation
Values in ( ) represent the subclock cycles when the subclock oscillation is operating at 32.768 kHz.

Reference:

Because the oscillation cycles are unstable just after the oscillation starts, the oscillation stabilization wait timer serves as a guideline.

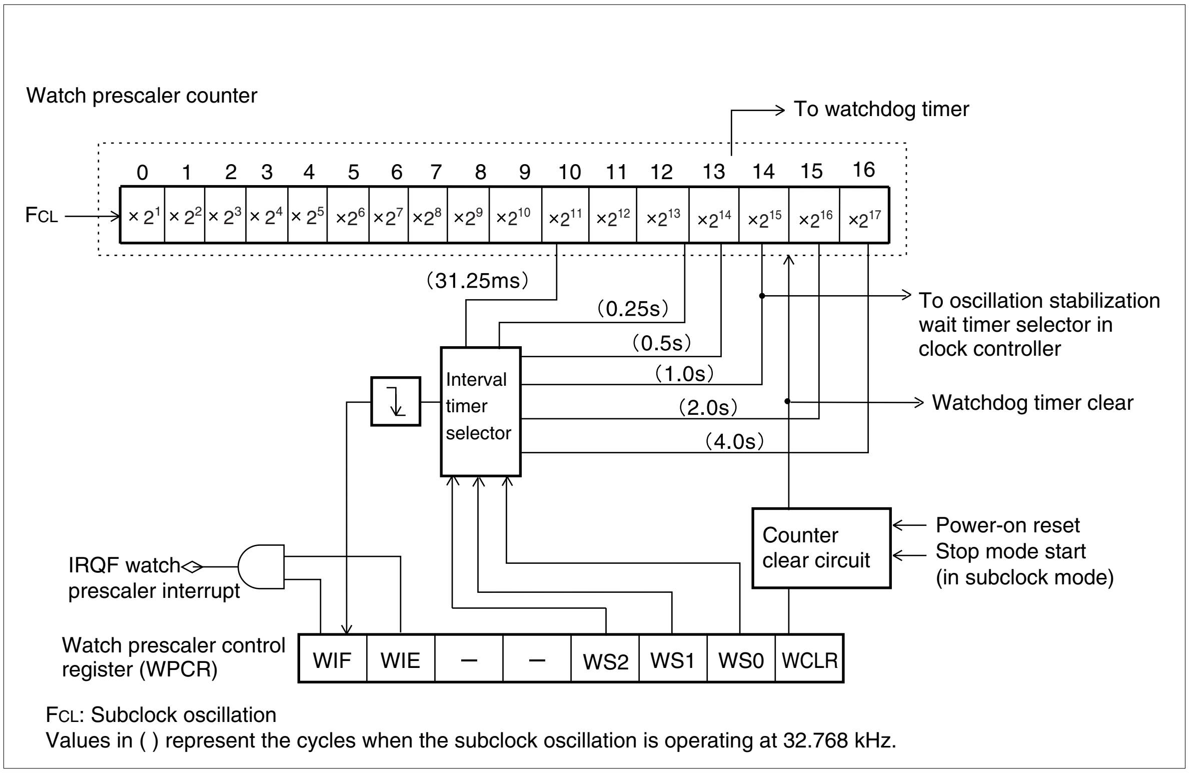
7.2 Configuration of the Watch Prescaler

The watch prescaler comprises the following blocks:

- **watch prescaler counter**
- **Counter clear circuit**
- **Interval timer selector**
- **watch prescaler control register (WPCR)**

■ Block Diagram of the Watch Prescaler

Figure 7.2-1 Block Diagram of the Watch Prescaler



FCL: Subclock oscillation
Values in ( ) represent the cycles when the subclock oscillation is operating at 32.768 kHz.

❍ Watch prescaler counter

17-bit up-counter using the subclock oscillation as the count clock.

❍ Counter clear circuit

The counter clear circuit clears the counter when, in addition to the setting by the WPCR register (WPCR: WCLR=0), a transition to the sub-stop mode (STBC: STP=1) or an optional power-on reset occurs.

❍ Interval timer selector

Circuit to select one divide-by output from four kinds of divide-by output from the watch prescaler counter. The falling edges of the selected divide-by output become an interrupt source.

- **Watch prescaler control register (WPCR)**

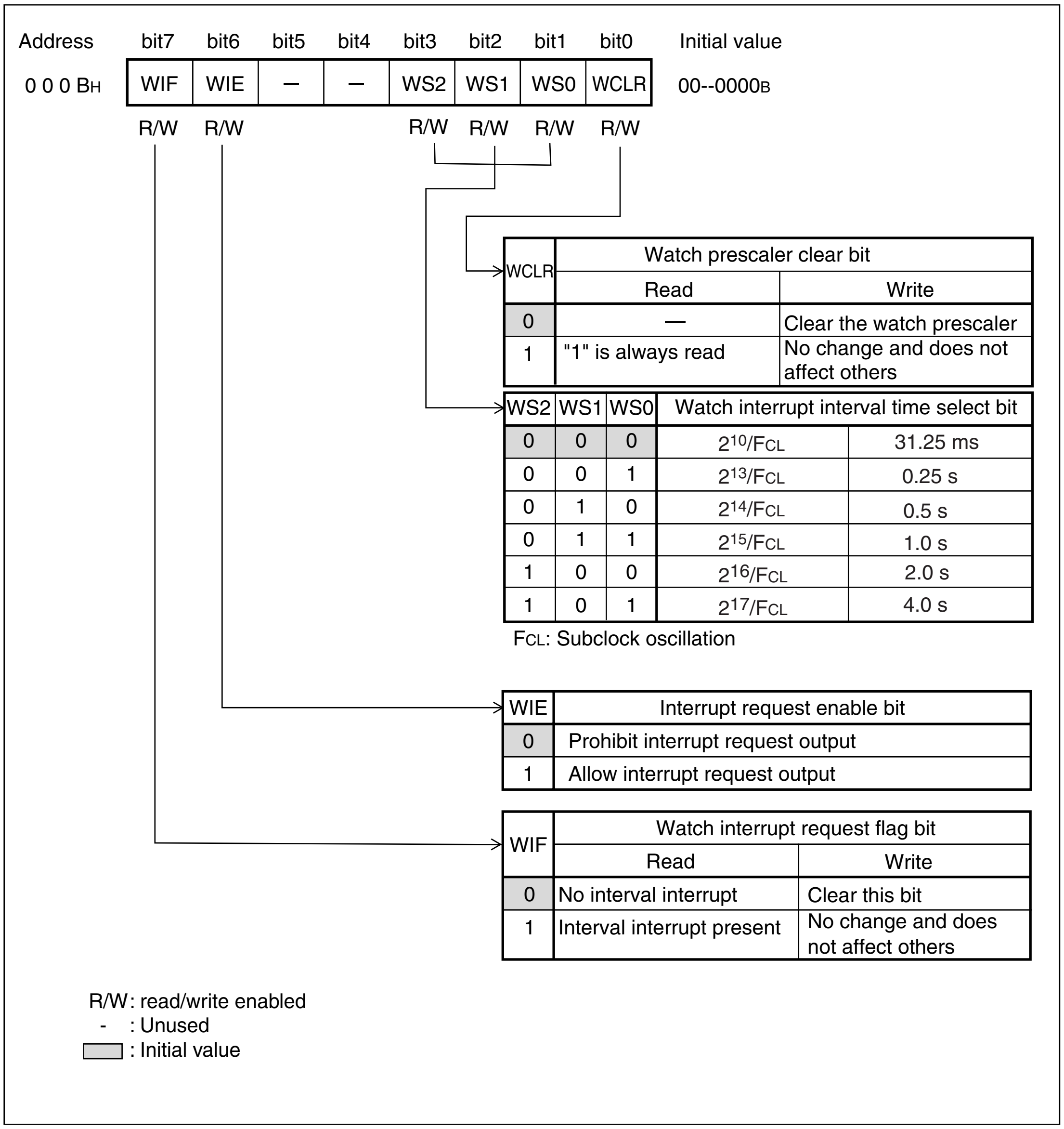
  This register is used to select the interval time, clear the counter, control interrupts, and check status.

7.3 Watch Prescaler Control Register (WPCR)

The watch prescaler control register (WPCR) is a register used to select the interval time, clear the counter, control interrupts, and check status.

■ Watch Prescaler Control Register (WPCR)

Figure 7.3-1 Watch Prescaler Control Register (WPCR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 0 BH	WIF	WIE	—	—	WS2	WS1	WS0	WCLR	00--0000B
	R/W	R/W			R/W	R/W	R/W	R/W	

WCLR	Watch prescaler clear bit	
	Read	Write
0	—	Clear the watch prescaler
1	"1" is always read	No change and does not affect others

WS2	WS1	WS0	Watch interrupt interval time select bit	
0	0	0	2^{10}/FCL	31.25 ms
0	0	1	2^{13}/FCL	0.25 s
0	1	0	2^{14}/FCL	0.5 s
0	1	1	2^{15}/FCL	1.0 s
1	0	0	2^{16}/FCL	2.0 s
1	0	1	2^{17}/FCL	4.0 s

FCL: Subclock oscillation

WIE	Interrupt request enable bit
0	Prohibit interrupt request output
1	Allow interrupt request output

WIF	Watch interrupt request flag bit	
	Read	Write
0	No interval interrupt	Clear this bit
1	Interval interrupt present	No change and does not affect others

R/W: read/write enabled
- : Unused
▨ : Initial value

Table 7.3-1 Explanation of the Functions of Each Bit of the Watch Prescaler Control Register (WPCR)

Bit name		Function
Bit 7	WIF: Watch interrupt request flag bit	• "1" is set by the falling edges of the selected divide-by output for interval timer. • If both this bit and the interrupt request enable bit (WIE) are "1", an interrupt request is output. • This bit is cleared if "0" is written into this bit. If "1" is written, no change occurs and no operation is affected.
Bit 6	WIE: Interrupt request enable bit	• Bit to allow/prohibit interrupt request output to CPU. • If both this bit and the watch interrupt request flag bit (WIF) are "1", an interrupt request is output.
Bit 5 Bit 4	Unused bits	• The read value is undefined. • Writing has no effect on operation.
Bit 3 Bit 2 Bit 1	WS2, WS1, WS0: Watch interrupt interval time select bit	• Bits to select the interval timer cycle • Bits for the interval timer of the counter of the watch prescaler are specified. • Six kinds of interval time can be selected.
Bit 0	WCLR: watch prescaler clear bit	• Bit to clear the counter of the watch prescaler • If "0" is written into this bit, the counter is cleared to "0000_H". If "1" is written, no change occurs and no operation is affected. **Reference:** "1" is always read.

7.4 Watch Prescaler Interrupt

The watch prescaler generates interrupt requests using the falling edges of the selected divide-by output (interval timer function).

■ Interrupt when the Interval Timer Function is Active (Watch Interrupt)

The counter for the watch prescaler counts up using the subclock oscillation. When the specified interval time passes, if not in main stop mode, the watch interrupt request flag bit (WPCR: WIF=1) is set to "1". At this time, if the interrupt request enable bit is set (WPCR: WIE=1), an interrupt request to CPU (IRQF) is issued. Clear the interrupt request to "0" by writing "0" into the WIF bit using an interrupt processing routine. The WIF bit is set whenever the specified divide-by output falls regardless of the value of the WIE bit.

Note:

To allow interrupt request output (WPCR: WIE=1) after releasing a reset, clear (WPCR: WIF=0) the WIF bit at the same time.

Reference:

If the WIE bit is changed from prohibition to permission (WPCR: WIE=0 --> 1) when the WIF bit is "1", an interrupt request is issued immediately.

If the counter clear (WPCR: WCLR=0) and an overflow of the selected bit occur at the same time, the WIF bit is not set.

■ Oscillation Stabilization Wait Time and Watch Interrupts

If an interval time period shorter than the oscillation stabilization wait time of the subclock is set, a watch interrupt request (WPCR: WIF=1) of the watch prescaler is issued when returning from the sub-stop mode following an external interrupt. In this case, prohibit (WPCR: WIE=0) interrupts of the watch prescaler when making a transition to the sub-stop mode.

■ Register and Vector Table Related to the Watch Prescaler Interrupts

Table 7.4-1 "Register and Vector Table Related to the Watch Prescaler Interrupts" lists the register and vector table related to the watch prescaler interrupts.

Table 7.4-1 Register and Vector Table Related to the Watch Prescaler Interrupts

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Bit to be set		Upper	Lower
IRQF	ILR4 (007E$_H$)	LF1 (bit 7)	LF0 (bit 6)	FFDC$_H$	FFDD$_H$

For interrupt operations, see Section 3.4.2 "Interrupt Processing".

7.5 Operation of the Watch Prescaler

The watch prescaler operates to provide the interval timer function .

■ Operation of the Interval Timer Function (Watch Prescaler)

The setting in Figure 7.5-1 "Setting of the Interval Timer Function" is required for the operation of the interval timer function.

Figure 7.5-1 Setting of the Interval Timer Function

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
WPCR	WIF	WIE	—	—	WS2	WS1	WS0	WCLR
	0	1			◎	◎	◎	0

◎ : Bit used
1 : 1 is set
0 : 0 is set

The 17-bit counter of the watch prescaler continues to count up the subclock provided the subclock oscillates.

If the counter is cleared (WPCR: WCLR=0), it starts to count up from "0000_H". When "$1FFFE_H$" is reached, counting continues starting from "0000_H". When the time set with the watch interrupt interval time selection bit is reached during count-up, if the main stop mode is not in effect, "1" is set in the watch interrupt request flag bit (WPCR: WIF). That is, starting with the time when cleared, a watch interrupt request is generated at regular intervals of the selected time.

■ Operation of the Clock Supply Function

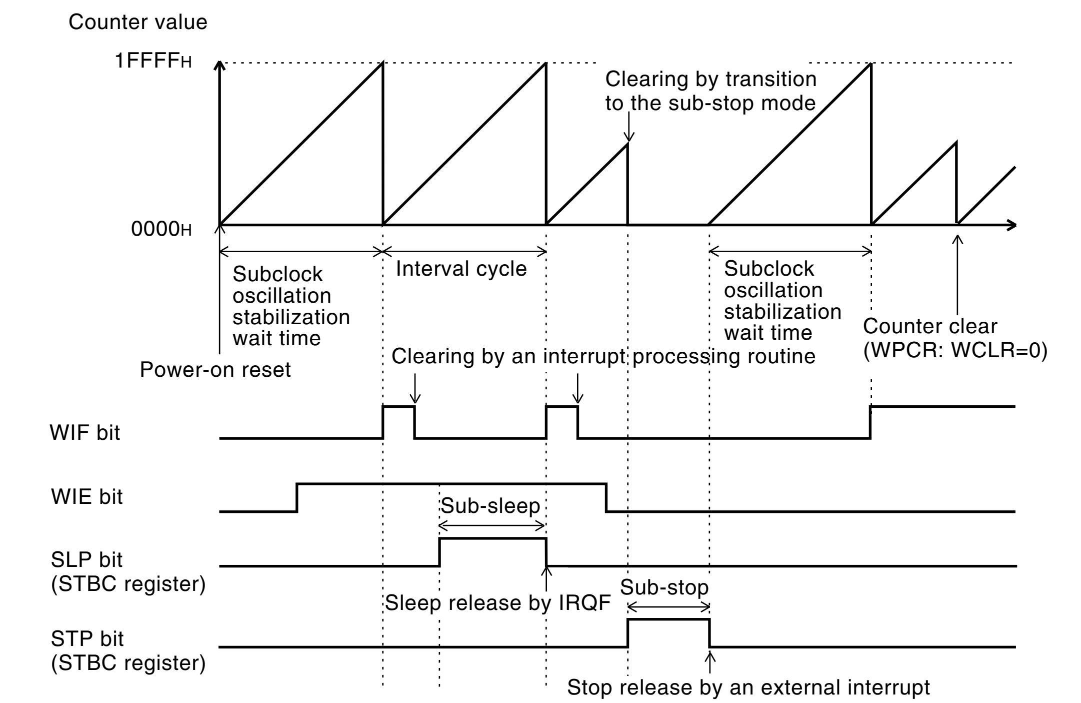
The watch prescaler is also used as a timer to generate the oscillation stabilization wait time of the subclock. Counting of the oscillation stabilization wait time of the subclock ($2^{15}/F_{CL}$, F_{CL}: subclock oscillation) starts when the watch prescaler is cleared and ends when the highest bit falls.

■ Operations of the Watch Prescaler

Figure 7.5-2 "Operations of the Watch Prescaler" shows the counter values if a transition to the sleep mode or stop mode occurs, or the counter clearing is requested when the interval timer function is operating in subclock mode.

The transition to the watch mode is the same as that to the sub-sleep mode.

Figure 7.5-2 Operations of the Watch Prescaler


Counter value
1FFFFH
0000H
Clearing by transition to the sub-stop mode
Subclock oscillation stabilization wait time
Interval cycle
Subclock oscillation stabilization wait time
Counter clear (WPCR: WCLR=0)
Power-on reset
Clearing by an interrupt processing routine
WIF bit
WIE bit
Sub-sleep
SLP bit (STBC register)
Sleep release by IRQF
Sub-stop
STP bit (STBC register)
Stop release by an external interrupt

If "11_B" is set to the interrupt interval time select bits (WPCR: WS1, WS0) of the watch prescaler control register ($2^{15}/F_{CL}$)

7.6 Notes on Using the Watch Prescaler

The following describes the precautions when using the watch prescaler.
The watch prescaler cannot be used when a single clock source is specified with the option setting.

■ Notes on Using the Watch Prescaler

❍ Precautions when setting the watch prescaler in programs

It is impossible to return from interrupt processing if the interrupt request flag bit (WPCR: WIF) is "1" and the interrupt request enable bit is set (WPCR: WIF=1). The WIF bit must be cleared.

❍ Clearing the watch prescaler

The watch prescaler is cleared, in addition to clearing by the watch prescaler clear bit (WPCR: WCLR=0), when the oscillation stabilization wait time of the subclock is required.

If the watch prescaler is selected (WDTC: CS=1) as the count clock of the watchdog timer, the watchdog timer is also cleared when the watch prescaler is cleared.

❍ Using the watch prescaler as a timer for the oscillation stabilization wait time

Since the subclock oscillation is stopped when the power is turned on or operating in sub-stop mode, the oscillator takes the oscillation stabilization wait time using the watch prescaler after activating operations.

Do not make a transition from the main clock mode to the subclock mode during oscillation stabilization wait time, such as just after power-on.

The oscillation stabilization wait time of the subclock is fixed.

For details, see Section 3.6.5 "Oscillation Stabilization Wait Time".

❍ Precautions when using watch interrupts

In main stop mode, the watch prescaler performs a count operation but a watch interrupt (IRQF) does not occur.

❍ Precautions when using the peripheral functions that use clocks supplied from the prescaler.

If the counter of the watch prescaler is cleared, the "H" level of the clock supplied by the watch prescaler is short and its "L" level may be longer by a maximum of 1/2 cycle because the output originates from the initial state.

Though the clock for the watchdog timer is also output from the initial state, the watchdog timer works in normal cycles because the counter of the watchdog timer is cleared simultaneously.

7.7 Program Example of the Watch Prescaler

The following shows a program example of the watch prescaler.

■ Program Example of the Watch Prescaler

❍ Processing specifications

Generate the watch interrupt of $2^{15}/F_{CL}$ (F_{CL}: subclock oscillation) repeatedly. The interval time is about 1 s (for 32.768 kHz operation).

❍ Coding example

```
WPCR   EQU     000BH           ; Address of the watch prescaler
                                 control register
WIF    EQU     WPCR:7          ; Definition of watch interrupt
                                 request flag bit
ILR4   EQU     007EH           ; Address of the interrupt level
                                 setting register
INT_V  DSEG    ABS             ; [DATA SEGMENT]
ORG    0FFDCH
IRQF   DW      WARI            ; Setting interrupt vector
INT_V  ENDS
;----Main program---------------------------------------------------------
       CSEG                    ; [CODE SEGMENT]
                               ; Stack pointer (SP) and other are
                                 assumed to have been initialized
         :
       CLRI                    ; Interrupt disable
       MOV     ILR4,#10111111B ; Setting interrupt level (level 2)
       MOV     WPCR,#01000110B ; Clearing interrupt request flag,
                                 enabling interrupt request output,
                                 selecting  2^15/F_CL, and clearing watch
                                 prescaler
       SETI                    ; Interrupt enable
         :
;----Interrupt program----------------------------------------------------
WARI   CLRB    WIF             ; Clearing interrupt request flag
       PUSHW   A
       XCHW    A,T
       PUSHW   A
         :
       User processing
         :
       POPW    A
       XCHW    A,T
       POPW    A
       RETI
       ENDS
; -----------------------------------------------------------------------
       END
```

CHAPTER 8 2-CHANNEL 8-BIT PWM TIMERS

This chapter describes the functions and operation of the 2-channel 8-bit PWM timer.

8.1 Overview of the 2-Channel 8-Bit PWM Timer (Interval Timer Function)

The 2-channel 8-bit PWM timer consists of two 8-bit PWM timers (CH1 and CH2) that increment the counter value in synchronization with four internal count clocks. CH1 and CH2 each have an interval timer function that outputs square waves and an 8- or 7-bit resolution PWM timer function. CH1 and CH2 can select any of these functions. The interval timer function provides two modes: 8-bit timer mode and CK12 mode. In 8-bit timer mode, CH1 and CH2 are used separately. In CK12 mode, CH1 and CH2 are used combined.
The output cycles of CH1 and CH2 can be used as UART transfer clocks.

■ Interval Timer Function (Square Wave Output Function)

The interval timer function is used to repeatedly generate interrupts at arbitrary time intervals. This function can also output square waves at an arbitrary frequency because it can invert the output level of a pin (PT01, PT02 pin) each time an interrupt occurs.

❍ 8-bit timer mode

- In 8-bit timer mode, the CH1 and CH2 8-bit PWM timers operate independently.
- Interval timer operation is possible with 1 to 2^8 count clock cycles.
- One of four count clocks can be selected.

Table 8.1-1 "Interval Times and Square Wave Output Ranges (CH1, CH2)" lists the interval times and square wave output ranges.

Table 8.1-1 Interval Times and Square Wave Output Ranges (CH1, CH2)

Count clock cycle		Interval time	Square wave output range (Hz)
Internal count clock	$1t_{inst}$	$1t_{inst}$ to $2^{8}t_{inst}$	$1/(2t_{inst})$ to $1/2(^{9}t_{inst})$
	$8t_{inst}$	$2^{3}t_{inst}$ to $2^{11}t_{inst}$	$1/(2^{4}t_{inst})$ to $1/2(^{12}t_{inst})$
	$16t_{inst}$	$2^{4}t_{inst}$ to $2^{12}t_{inst}$	$1/(2^{5}t_{inst})$ to $1/2(^{13}t_{inst})$
	$64t_{inst}$	$2^{6}t_{inst}$ to $2^{14}t_{inst}$	$1/(2^{7}t_{inst})$ to $1/2(^{15}t_{inst})$

t_{inst}: Instruction cycle (affected by the clock mode and other factors)

Reference:

[Example of calculating interval time and square wave frequency]

When the oscillation (F_{CH}) of the main clock is 10 MHz, the value of the PWM compare register (COMR) is DD_H(221), and the count clock cycle is 1 t_{inst}, and the interval timer function is used continuously without modifying the COMR register value, the interval time and the square wave frequency output to the PWM pin are calculated as shown below.

Note, however, that the calculated values are valid only when the fastest clock (SYCC: CS1, CS0 = 11_B, 1 instruction cycle = $4/F_{CH}$) available in main clock mode (SYCC: SCS = 1) is selected with the system clock control register (SYCC).

$$\begin{aligned} \text{Interval time} &= (1 \text{ x } 4/F_{CH}) \text{ x } (\text{COMR register value} + 1) \\ &= (4/10\text{MHz}) \text{ x } (221+1) \\ &= 88.8\mu\text{s} \end{aligned}$$

$$\begin{aligned} \text{Output frequency} &= F_{CH}/\ (1 \text{ x } 8 \text{ x } (\text{COMR register value} + 1)) \\ &= 10\text{MHz}/\ (8 \text{ x } (221 + 1)) \\ &\fallingdotseq 5.63\text{kHz} \end{aligned}$$

❍ CK12 mode

CK12 mode is the mode in which CH1 and CH2 are combined. In this mode, the CH1 square wave output is used as the CH2 count clock.

- For CH1 and CH2, interval timer operation from 1 to 2^8count clock cycles is possible.
- For CH1, one of four count clocks can be selected.
- The CH2 count clock is provided by the CH1 square wave output.

Table 8.1-2 "Interval Times and Square Wave Output Ranges" lists the interval times and square wave output ranges.

Table 8.1-2 Interval Times and Square Wave Output Ranges

Count clock cycle			Interval time	Square wave output range (Hz)
CH1	Internal count clock	$1t_{inst}$	$1t_{inst}$ to $2^{8}t_{inst}$	$1/(2t_{inst})$ to $1/2(^{9}t_{inst})$
		$8t_{inst}$	$2^{3}t_{inst}$ to $2^{11}t_{inst}$	$1/(2^{4}t_{inst})$ to $1/2(^{12}t_{inst})$
		$16t_{inst}$	$2^{4}t_{inst}$ to $2^{12}t_{inst}$	$1/(2^{5}t_{inst})$ to $1/2(^{13}t_{inst})$
		$64t_{inst}$	$2^{6}t_{inst}$ to $2^{14}t_{inst}$	$1/(2^{7}t_{inst})$ to $1/2(^{15}t_{inst})$
CH2	CH1 square wave output	$2t_{inst}$ to $2^{15}t_{inst}$	$2t_{inst}$ to $2^{23}t_{inst}$	$1/(2^{2}t_{inst})$ to $1/2(^{24}t_{inst})$

t_{inst}: Instruction cycle (affected by the clock mode and other factors)

Reference:

[Example of calculating the interval time and square wave frequency]

When the oscillation (F_{CH}) of the main clock is 10 MHz, the values of both PWM compare registers 1 and 2 (COMR1 and COMR2) are "DD_H(221)," and the CH1 count clock cycle is 1 t_{inst}, and the interval timer function is used continuously without modifying the values of the COMR registers, the interval time and the square wave frequency output to the PWM pin are calculated as shown below:

Note, however, that the calculated values are valid only when the fastest clock (CS1, CS0 = 11_B, 1 instruction cycle = $4/F_{CH}$) available in main clock mode (SCS = 1) is selected with the system clock control register (SYCC).

CH1 Interval time = (1 x 4/F_{CH}) x (COMR1 register value + 1)
= (1 x 4/10MHz) x (221+1)
= 88.8μs

CH1 output frequency = F_{CH}/ (1 x 8 x (COMR1 register value + 1))
= 10MHz/ (1 x 8 x (221 + 1))
≒ 5.63kHz

CH2 Interval time = (1 x 4/CH1 output frequency) x (COMR2 register value + 1)
= (1 x 4/5.63kHz) x (221+1)
= 157.7ms

CH2 output frequency = CH1 output frequency/ (1 x 8 x (COMR2 register value + 1))
= 5.63kHz/ (1 x 8 x (221 + 1))
≒ 3.17Hz

8.2 Overview of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)

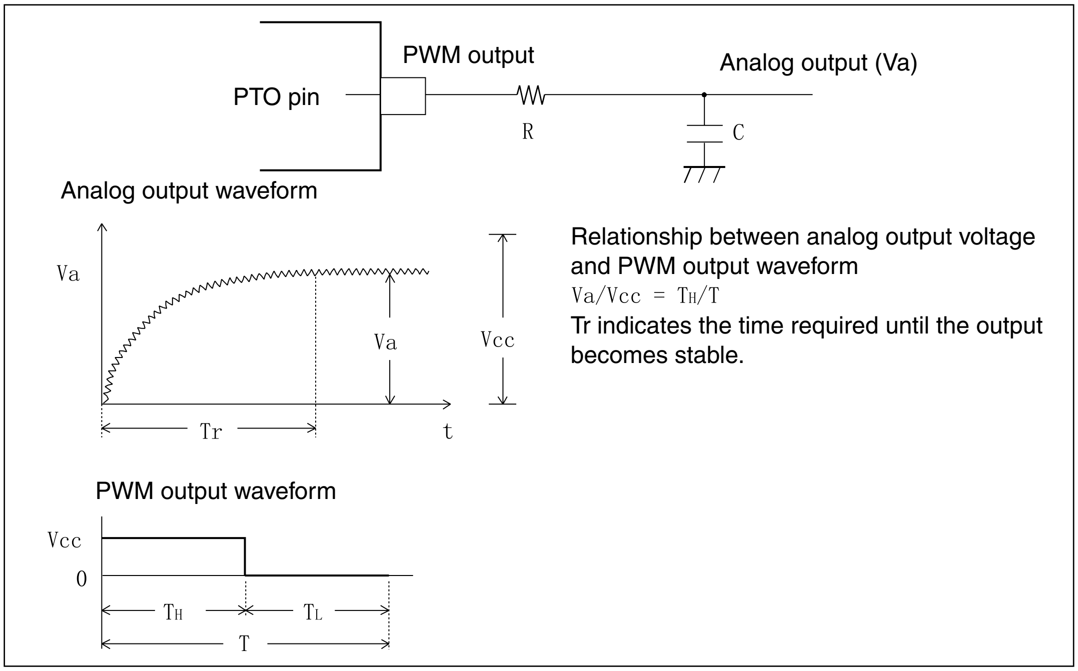
The 2-channel 8-bit PWM timer consists of two 8-bit PWM timers (CH1 and CH2) that increment the counter value in synchronization with four internal count clocks. CH1 and CH2 each have an interval timer function that outputs square waves and an 8- or 7-bit resolution PWM timer function. CH1 and CH2 can select any of these functions.

■ Overview of 2-Channel 8-Bit PWM Timer (PWM Timer Function)

The PWM timer function supports 8-bit PWM mode, 7-bit PWM mode (high-speed mode), and CH12PWM mode. The 8-bit PWM mode uses CH1 and CH2 separately. The CH12PWM mode generates a PWM wave on the assumption that CH1 is the "L" width and CH2 is a cycle. It is also possible to operate CH1 in 8-bit timer mode and select square wave output as the CH2 count clock (CK12PWM mode).

The 2-channel 8-bit PWM timer can be used as a D/A converter by connecting a low-pass filter to the PWM output.

Figure 8.2-1 Example of Using the 2-Channel 8-Bit PWM Timer as a D/A Converter by Connecting a Low-Pass Filter to PWM Output



Relationship between analog output voltage and PWM output waveform
$V_a/V_{cc} = T_H/T$
Tr indicates the time required until the output becomes stable.

Reference:

[Example of calculating the PWM wave (CH12PWM mode)]

When the oscillation (F_{CH}) of the main clock is 10 MHz, the values of the PWM compare registers (COMR1 and COMR2) are 01_H and 03_H, respectively, and each count clock cycle is 1 t_{inst}, the PWM wave is calculated as shown below.

Note, however, that the calculated values are valid only when the fastest clock (SYCC: CS1, CS0 = 11_B, 1 instruction cycle = 4/F_{CH}) of the main clock (SYCC: SCS = 1) mode is selected with the system clock control register (SYCC).

$$\begin{aligned}\text{"L" width} &= (1 \times 4/F_{CH}) \times (\text{COMR1 register value} + 1)\\ &= (1 \times 4/10\text{MHz}) \times (1+1)\\ &= 0.8\mu s\end{aligned}$$

$$\begin{aligned}\text{1-cycle width} &= (1 \times 4/F_{CH}) \times (\text{COMR2 register value} + 1)\\ &= (1 \times 4/10\text{MHz}) \times (3+1)\\ &= 1.6\ \mu s\end{aligned}$$

■ PWM Timer Function

The PWM timer function controls the "H" width of one cycle or controls the "L" width and cycles independently to output a PWM wave to the PT01 or PT02 pin. This function can also be used to provide a D/A converter by connecting a low-pass filter to the PWM output.

When used independently, CH1 and CH2 each can select 8-bit PWM mode and 7-bit PWM mode (high-speed mode).

❍ 8-bit PWM mode

- PWM waves can be output at a duty ratio of 0 to 99.6% because the "H" width of one cycle can be controlled at a resolution of 1/256.
- A PWM wave cycle is 2^8 count clock cycles. Four PWM wave cycles are provided, one of which can be selected.

❍ 7-bit PWM mode (high-speed mode)

- PWM waves can be output at a duty ratio of 0 to 99.2% because the "H" width of one cycle can be controlled at a resolution of 1/128.
- A PWM wave cycle is 2^7 (1/2 of 8-bit PWM mode) times the count clock cycle. Four PWM wave cycles are available, one of which can be selected.

❍ CK12PWM mode (8-bit PWM, 7-bit PWM)

- CH2 can select 8-bit PWM mode or 7-bit PWM mode. However, the count clock is provided by the CH1 square wave output.
- CH1 operates in 8-bit timer mode and can control PWM wave cycles.

❍ CH12PWM mode

- The "L" width of a PWM wave can be controlled for up to 2^8 count clock cycles for each of the four CH1 count clocks.
- PWM wave cycles can be controlled for up to 2^8 count clock cycles for each of the four CH2 count clocks.
- PWM waves can be usually controlled at a resolution of 1/256 and can be output at a duty ratio of 0 to 99.6%.
- PWM waves can be controlled up to a resolution of $1/2^{14}$ (minimum), but the duty ratio is limited.

❍ PWM wave cycles in each mode

Table 8.2-1 PWM Wave Cycles That Can Be Set by the PWM Timer Function

	Count clock cycle		**Independent use of CH1, CH2 (normal mode)**		**CH12PWM mode**	
			Cycle in 8-bit PWM mode	**Cycle in 7-bit PWM mode (high-speed mode)**	**"L" width (CH1)**	**1-cycle width (CH2)**
CH1 and CH2 in non-CK12PWM mode	Internal count clock	$1t_{inst}$	2^8t_{inst}	2^7t_{inst}	$1t_{inst}$ to 2^8t_{inst}	$1t_{inst}$ to 2^8t_{inst}
		$8t_{inst}$	$2^{11}t_{inst}$	$2^{10}t_{inst}$	2^3t_{inst} to $2^{11}t_{inst}$	2^3t_{inst} to $2^{11}t_{inst}$
		$16t_{inst}$	$2^{12}t_{inst}$	$2^{11}t_{inst}$	2^4t_{inst} to $2^{12}t_{inst}$	2^4t_{inst} to $2^{12}t_{inst}$
		$64t_{inst}$	$2^{14}t_{inst}$	$2^{13}t_{inst}$	2^6t_{inst} to $2^{14}t_{inst}$	2^6t_{inst} to $2^{14}t_{inst}$
CH2 in CK12PWM mode	CH1 square wave output	$2t_{inst}$ to $2^{15}t_{inst}$	2^9t_{inst} to $2^{23}t_{inst}$	2^8t_{inst} to $2^{22}t_{inst}$		

t_{inst}: Instruction cycle (affected by the clock mode and other factors)

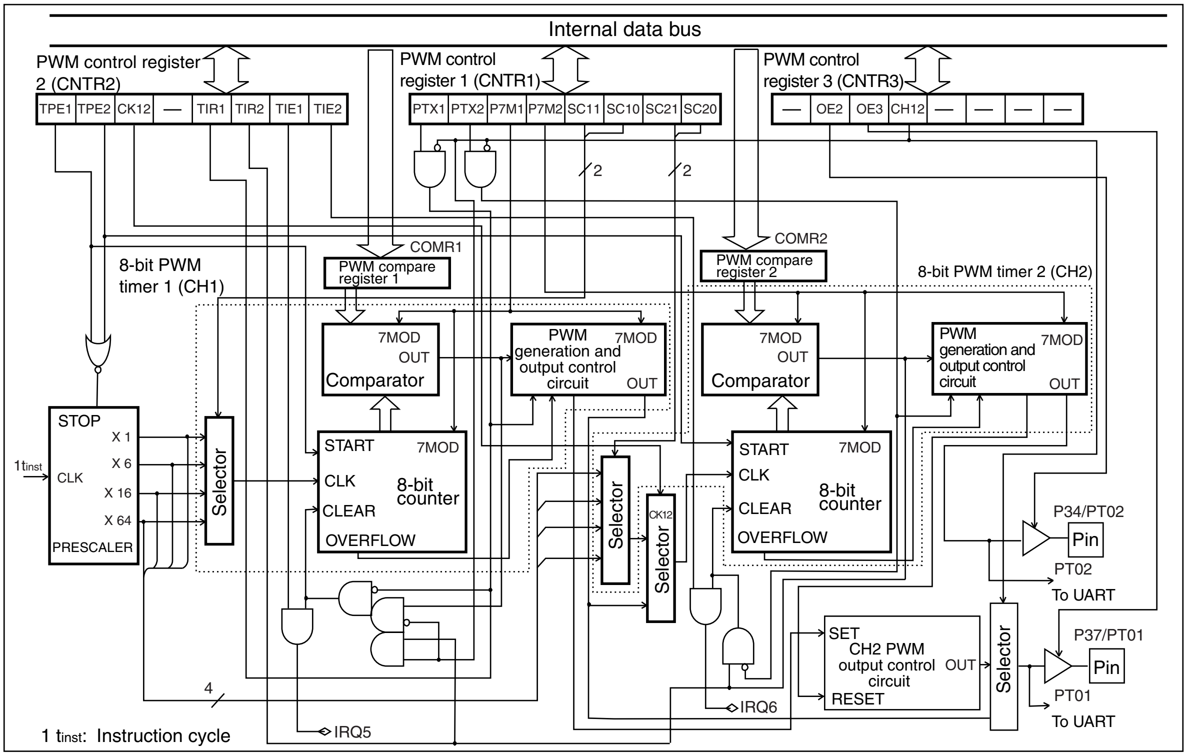
8.3 Configuration of the 2-Channel 8-Bit PWM Timer

The 2-channel 8-bit PWM timer consists of the following seven blocks:

- **Prescaler**
- **8-bit PWM timer 1 (CH1)**
- **8-bit PWM timer 2 (CH2)**
- **PWM compare registers 1 and 2 (COMR1 and COMR2)**
- **PWM control registers 1, 2, and 3 (CNTR1, CNTR2, and CNTR3)**
- **CK12 selector**
- **CH12PWM output control circuit**

■ Block Diagram of 2-Channel 8-Bit PWM Timer

Figure 8.3-1 Block Diagram of the 2-Channel 8-Bit PWM Timer



❍ Prescaler

This circuit divides operating clocks for peripheral circuits.

When any of the counter operation enable bits (CNTR2: TPE1 and TPE2) of PWM control register CNTR2 is "1," the prescaler operates and outputs four internal count clocks.

❍ 8-bit PWM timer 1 (CH1) and 8-bit PWM timer 2 (CH2)

Count clock selector:

This circuit selects the four internal count clocks. The selected clock functions as the clock for incrementing the 8-bit counter.

8-bit counter:

The 8-bit counter is incremented according to the count clock selected by the counter clock selector.

Comparator:

When the value of the 8-bit counter is 00_H, the comparator latches the COMR register value. It also compares the 8-bit counter value with the latched COMR register value to detect a match.

PWM generation and output control circuit:

If a match is detected during operation as an interval timer, an interrupt request is not generated. If output pin control bit 2 (OE2 bit or OE3 bit of CNTR3) is "1" when an interrupt request is generated, the output control circuit inverts the output level of the PWM pin and at the same time clears the 8-bit counter.

If a match is detected during operation as a PWM timer, the PWM generation circuit changes the output level of the PWM pin from High to Low. If the 8-bit counter overflows later, the output level of the PWM pin changes to High.

❍ PWM compare registers 1 and 2 (COMR1 and COMR2)

These registers are used to set the values to be compared with the values of the 8-bit counter.

❍ PWM control registers 1, 2, and 3 (CNTR1, CNTR2, and CNTR3)

CNTR1, CNTR2, and CNTR3 are used to select an operating mode, enable and disable operations, set count clocks, control interrupts, and check interrupt states.

If PWM timer mode (CNTR1: PTX1 or PTX2 = 1) is selected as the operating mode, clearing of an 8-bit counter by the match detection signal from the comparator and interrupt requests are disabled.

❍ CK12 selector

This input clock switching circuit changes the input clock of 8-bit PWM timer 2 (CH2) to the counter clock selector output or the square wave output of 8-bit PWM timer 1 (CH1).

❍ CH12PWM output control circuit

The CH12PWM output control circuit controls the "L" width (L --> H) and cycle (H --> L) of a PWM wave according to outputs from CH1 and CH2 in CH12PWM mode.

❍ Interrupt related to the 2-channel 8-bit PWM timer

- IRQ5: If the output of interrupt requests is enabled (CNTR2: TIEI = 1) when the counter value matches the value set in COMR1 of the CH1 interval timer function, an interrupt request occurs. (Interrupt requests do not occur when the ordinary PWM function is operating.)
- IRQ6: If interrupt requests are enabled (CNTR2: TIE2 = 1) when the counter value matches the value set in the COMR2 register in the CH2 interval timer function, or in CH12PWM mode, an interrupt request occurs. (Interrupt requests do not occur when the ordinary PWM function is operating.)

8.4 Pins of the 2-Channel 8-Bit PWM Timer

This section describes 2-channel 8-bit PWM timer pins and provides a block diagram of the pins.

■ 2-Channel 8-Bit PWM Timer Pins

The 2-channel 8-bit PWM timer pins consist of the P34/PT02 and P37/PT01 pins.

❍ P34/PT02 and P37/PT01 pins

The P34 and P37 pins function as the general-purpose I/O ports. They also serve as interval or PWM timer outputs (PT01 and PT02).

PT01, PT02

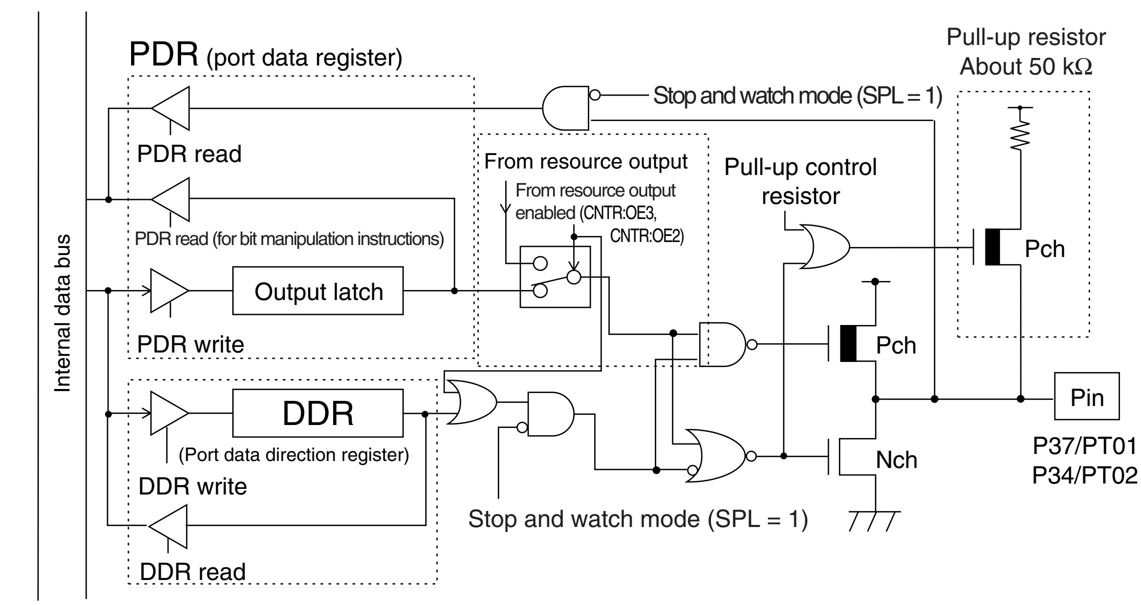
When the interval timer function is used, a square wave is output to these pins.

When the PWM timer function is used, a PWM wave is output to these pins.

When the resource output pin control bit is set to "1" (CNTR3: OE = 1), the P34/PT02 and P37/PT01 pins automatically function as output pins without reference to the values of the port data direction register output latch data (DDR: bit 1). That is, they function as PT01 and PT02 pins.

■ Block Diagram of the 2-Channel 8-Bit PWM Timer Pins

Figure 8.4-1 Block Diagram of the 2-Channel 8-Bit PWM Timer Pins


Internal data bus
PDR (port data register)
PDR read
PDR read (for bit manipulation instructions)
Output latch
PDR write
DDR
(Port data direction register)
DDR write
DDR read
From resource output
From resource output enabled (CNTR:OE3, CNTR:OE2)
Stop and watch mode (SPL = 1)
Pull-up control resistor
Pull-up resistor About 50 kΩ
Pch
Pch
Nch
Pin
P37/PT01
P34/PT02
Stop and watch mode (SPL = 1)

SPL: This bit specifies the pin state of the standby control register (STBC).

Reference:

When "pull-up resistor available" is selected in the pull-up option setting register, the pin state in stop and watch mode (STBC: SPL = 1) is High (pull-up state), not high impedance. During a reset, however, pull-up is invalid and the pin state becomes Hi-z.

8.5 Registers of the 2-Channel 8-Bit PWM Timer

This section describes 2-channel 8-bit PWM timer registers.

■ Registers of the 2-Channel 8-Bit PWM Timer

Figure 8.5-1 Registers of the 2-Channel 8-Bit PWM Timer

CNTR1, 2, 3 (PWM control registers 1, 2, 3)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CNTR1	0 0 2 7H	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20	00000000B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CNTR2	0 0 2 8H	TPE1	TPE2	CK12	—	TIR1	TIR2	TIE1	TIE2	000-0000B
		R/W	R/W	R/W		R/W	R/W	R/W	R/W	

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CNTR3	0 0 2 9H	—	OE2	OE3	CH12	—	—	—	—	-000----B
			R/W	R/W	R/W					

COMR1, 2 (PWM compare registers 1, 2)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
COMR1	0 0 2 AH									XXXXXXXXB
		W	W	W	W	W	W	W	W	

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
COMR2	0 0 2 BH									XXXXXXXXB
		W	W	W	W	W	W	W	W	

R/W : Can be read and written
W : Write only
- : Unused
X : Undefined

Note:

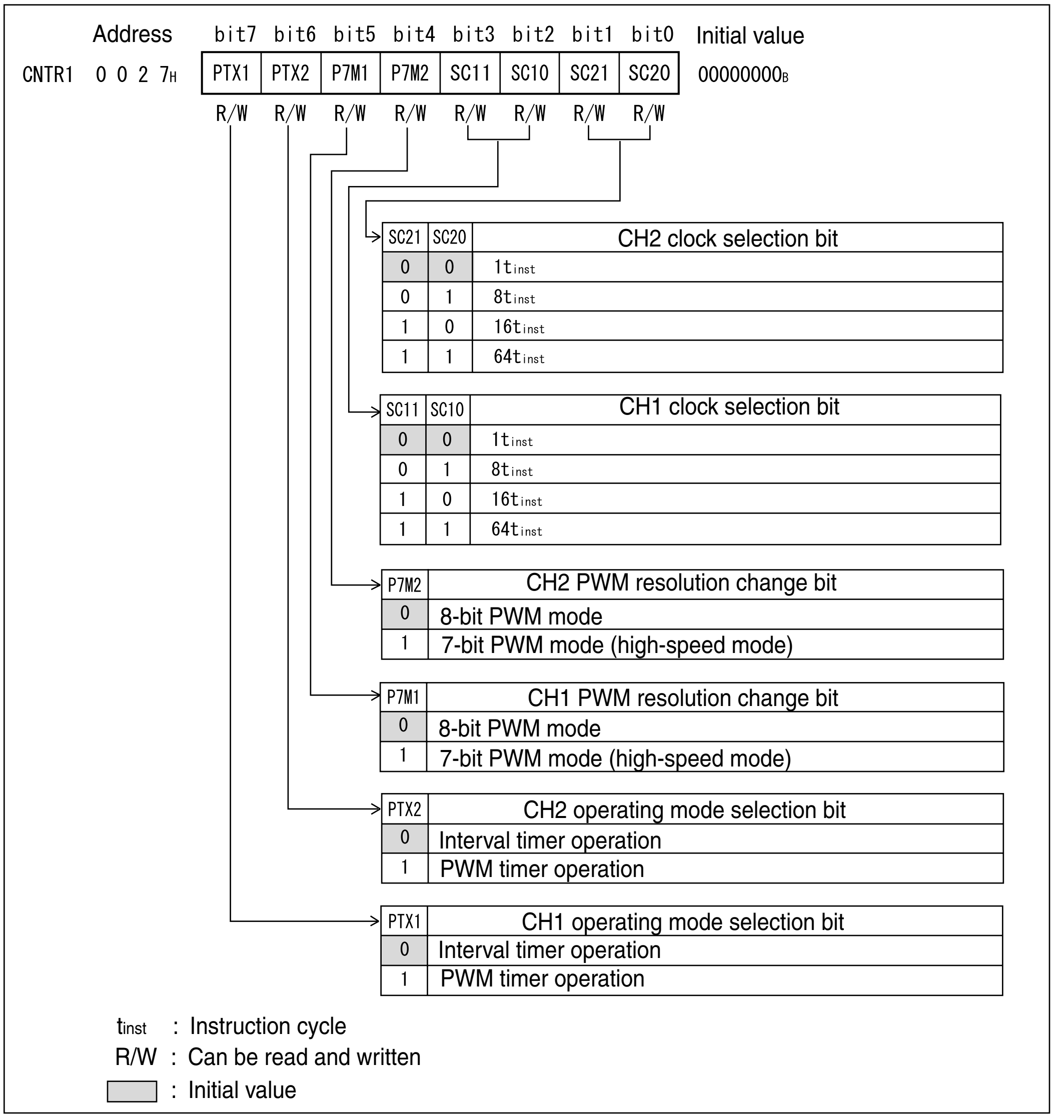
PWM compare registers 1 and 2 (COMR1 and COMR2) cannot use bit manipulation instructions because they are write-only registers.

8.5.1 PWM Control Register 1 (CNTR1)

PWM control register 1 (CNTR1) is used to select an operating mode (interval timer operation or PWM timer operation) for the 2-channel 8-bit PWM timer and count clock. CNTR1 is also used to change the resolution of the PWM timer function.

■ PWM Control Register 1 (CNTR1)

Figure 8.5-2 PWM Control Register 1 (CNTR1)



	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CNTR1	0027H	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20	00000000B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

SC21	SC20	CH2 clock selection bit
0	0	$1t_{inst}$
0	1	$8t_{inst}$
1	0	$16t_{inst}$
1	1	$64t_{inst}$

SC11	SC10	CH1 clock selection bit
0	0	$1t_{inst}$
0	1	$8t_{inst}$
1	0	$16t_{inst}$
1	1	$64t_{inst}$

P7M2	CH2 PWM resolution change bit
0	8-bit PWM mode
1	7-bit PWM mode (high-speed mode)

P7M1	CH1 PWM resolution change bit
0	8-bit PWM mode
1	7-bit PWM mode (high-speed mode)

PTX2	CH2 operating mode selection bit
0	Interval timer operation
1	PWM timer operation

PTX1	CH1 operating mode selection bit
0	Interval timer operation
1	PWM timer operation

t_{inst} : Instruction cycle
R/W : Can be read and written
▨ : Initial value

Table 8.5-1 Function of the PWM Control Register 1 (CNTR1) Bits

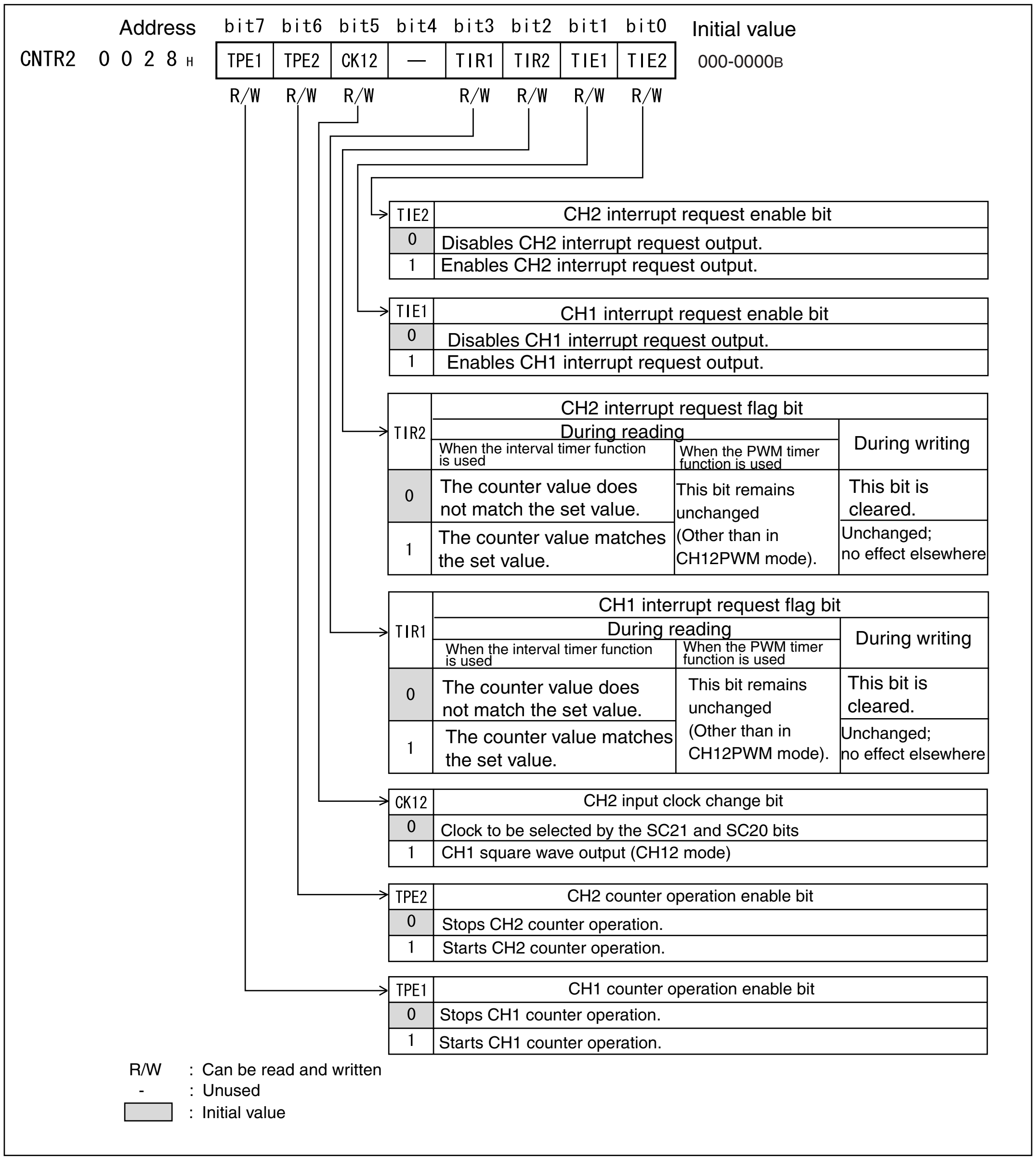
Bit		Description
bit7	PTX1: CH1 operating mode selection bit	• This bit is used to select CH1 interval timer operation or CH1 PWM timer operation. • If this bit is set to "0," CH1 interval timer operation is selected. If this bit is set to "1," CH1 PWM timer operation is selected. **Note:** Before writing this bit, stop counter operation (CNTR2: TPE1 = 0), disable interrupt requests (CNTR2: TIE1 = 0), and clear the request level flag bit (CNTR2: TIR1 = 0). When the CH12 bit of CNTR3 is "1" (CH12 mode), this bit is meaningless.
bit6	PTX2: CH2 operating mode selection bit	• This bit is used to select CH2 interval timer operation or CH2 PWM timer operation. • If this bit is set to "0," CH2 interval timer operation is selected. If this bit is set to "1," CH2 PWM timer operation is selected. **Note:** Before writing this bit, stop counter operation (CNTR2: TPE2 = 0), disable interrupt requests (CNTR2: TIE2 = 0), and clear the request level flag bit (CNTR2: TIR2 = 0). When the CH12 bit of CNTR3 is "1" (CH12 mode), this bit is meaningless.
bit5	P7M1: CH1 PWM resolution change bit	• For CH1 PWM timer operation, this bit is used to change 8-bit PWM mode to 7-bit PWM mode (high-speed mode) or vice versa. • Setting this bit to "0" changes 7-bit PWM mode to 8-bit PWM mode. Setting this bit to "1" changes 8-bit PWM mode to 7-bit PWM mode. **Note:** For interval timer operation, do not set this bit to "1."
bit4	P7M2: CH2 PWM resolution change bit	• For CH2 PWM timer operation, this bit is used to change 8-bit PWM mode to 7-bit PWM mode (high-speed mode) or vice versa. • Setting this bit to "0" changes 7-bit PWM mode to 8-bit PWM mode. Setting this bit to "1" changes 8-bit PWM mode to 7-bit PWM mode. **Note:** For interval timer operation, do not set this bit to "1."
bit3 bit2	SC11, SC10: CH1 clock selection bits	• These bits are used to select the count clocks for the CH1 interval timer function and CH1 PWM timer function. • One out of four internal count clocks can be selected. **Note:** Do not change from the CH1 interval timer function to the CH1 PWM timer function or vice versa while the CH1 counter is operating (CNTR2: TPE1 =1).
bit1 bit0	SC21, SC20: CH2 clock selection bits	• These bits are used to select the count clocks of the CH2 interval timer function and CH2 PWM timer function. • One out of four internal count clocks can be selected. **Note:** Do not change from the CH2 interval timer function to the CH2 PWM timer function or vice versa while the CH2 counter is operating (CNTR2: TPE2 = 1).

8.5.2 PWM Control Register 2 (CNTR2)

PWM control register 2 (CNTR2) is used to enable and disable 2-channel 8-bit PWM timer operation, select a CK12 mode, control interrupts, and check interrupt states.

■ PWM Control Register 2 (CNTR2)

Figure 8.5-3 PWM Control Register 2 (CNTR2)



	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CNTR2	0 0 2 8H	TPE1	TPE2	CK12	—	TIR1	TIR2	TIE1	TIE2	000-0000B
		R/W	R/W	R/W		R/W	R/W	R/W	R/W	

TIE2	CH2 interrupt request enable bit
0	Disables CH2 interrupt request output.
1	Enables CH2 interrupt request output.

TIE1	CH1 interrupt request enable bit
0	Disables CH1 interrupt request output.
1	Enables CH1 interrupt request output.

TIR2	CH2 interrupt request flag bit		
	During reading		During writing
	When the interval timer function is used	When the PWM timer function is used	
0	The counter value does not match the set value.	This bit remains unchanged (Other than in CH12PWM mode).	This bit is cleared.
1	The counter value matches the set value.		Unchanged; no effect elsewhere

TIR1	CH1 interrupt request flag bit		
	During reading		During writing
	When the interval timer function is used	When the PWM timer function is used	
0	The counter value does not match the set value.	This bit remains unchanged (Other than in CH12PWM mode).	This bit is cleared.
1	The counter value matches the set value.		Unchanged; no effect elsewhere

CK12	CH2 input clock change bit
0	Clock to be selected by the SC21 and SC20 bits
1	CH1 square wave output (CH12 mode)

TPE2	CH2 counter operation enable bit
0	Stops CH2 counter operation.
1	Starts CH2 counter operation.

TPE1	CH1 counter operation enable bit
0	Stops CH1 counter operation.
1	Starts CH1 counter operation.

R/W : Can be read and written
- : Unused
▨ : Initial value

Table 8.5-2 Functions of PWM Control Register 2 (CNTR2) Bits

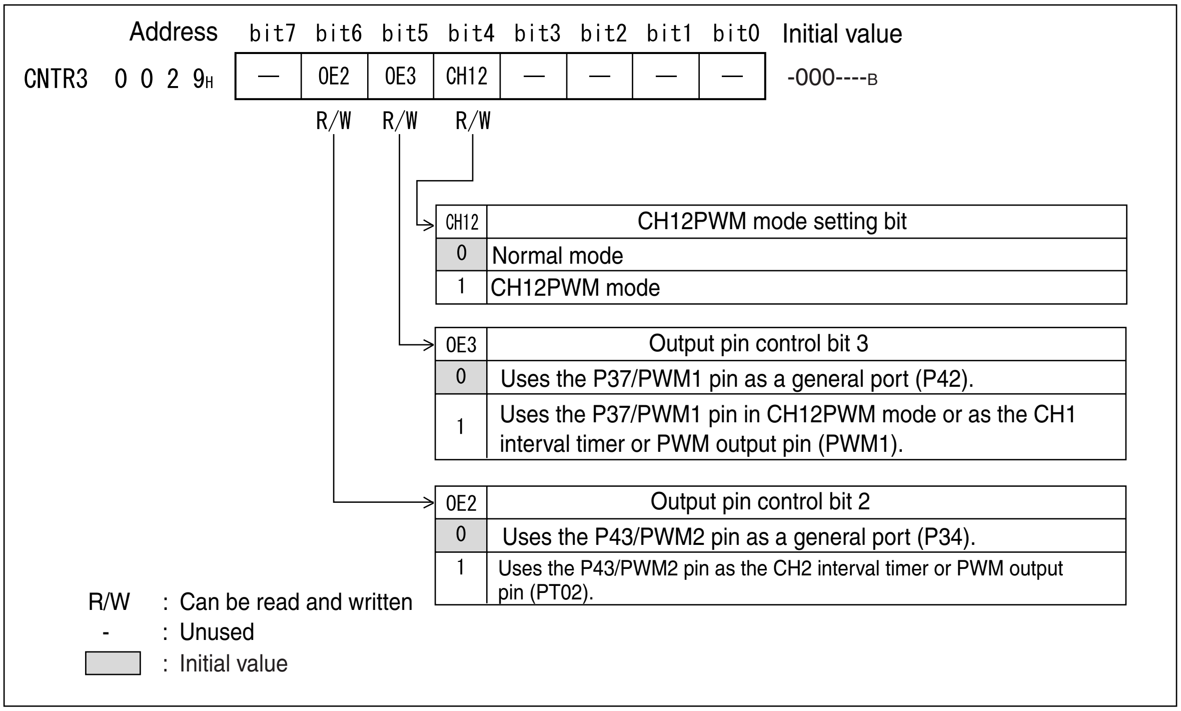
Bit		Description
bit7	TPE1: CH1 counter operation enable bit	• This bit is used to start and stop CH1 interval operation and PWM timer operation. • Setting this bit to "1" starts the count operation. Setting this bit to "0" clears the counter value to 00_H and stops the count operation.
bit6	TPE2: CH2 counter operation enable bit	• This bit is used to start and stop CH2 interval operation and PWM timer operation. • Setting this bit to "1" starts the count operation. Setting this bit to "0" clears the counter value to 00_H and stops the count operation.
bit5	CK12: CH2 input clock change bit	• This bit is used to change the CH2 input clock. • When this bit is set to "0," the clock selected by the SC21 and SC20 bits is used as the input clock. • When this bit is set to "1," CH1 square wave output is used as the input clock, and operation is in CK12 mode without reference to the values of the SC21 and SC20 bits. **Note:** When the CH12 bit of CNTR3 is "1" (CH12PWM mode) or the PTX1 bit of CNTR1 is "1" (CH1 PWM timer operation is selected), do not set this bit to "1."
bit4	Unused	• The value read from this bit is undefined. • Writing to this bit has no effect on operation.
bit3	TIR1: CH1 interrupt request flag bit	• If the count value matches the value in PWM compare register 1 (COMR1) when the CH1 interval timer function is being used, this bit is set to "1." • When this bit and the CH1 interrupt request enable bit (TIE1) are "1," an interrupt request is output to the CPU. • If the PWM timer function is used in a mode other than CH12PWM mode, interrupt requests do not occur. • Writing "0" clears this bit. Writing "1" has no effect.
bit2	TIR2: CH2 interrupt request flag bit	• If the count value matches the value in PWM compare register 2 (COMR2) when the CH2 interval timer function and CH12PWM mode are being used, this bit is set to "1." • When this bit and the CH2 interrupt request enable bit (TIE2) are "1," an interrupt request is output to the CPU. • If the PWM timer function is used in a mode other than CH12PWM mode, interrupt requests do not occur. • Writing "0" clears this bit. Writing "1" has no effect.
bit1	TIE1: CH1 interrupt request enable bit	• This bit is used to enable and disable the output of CH1 interrupt requests to the CPU. • When this bit and the CH1 interrupt request flag bit (TIR1) are "1," an interrupt request is output. **Note:** When the CH12 bit of CNTR3 is "1" (CH12PWM mode), disable the output of interrupt requests (TIE1 = 0).
bit0	TIE2: CH2 interrupt request enable bit	• This bit is used to enable and disable the output of CH2 interrupt requests to the CPU. • When this bit and the CH2 interrupt request flag bit (TIR2) are "1," an interrupt request is output.

8.5.3 PWM Control Register 3 (CNTR3)

PWM control register 3 (CNTR3) is used to select CH12PWM mode for the 2-channel PWM timer and control output pins.

■ PWM Control Register 3 (CNTR3)

Figure 8.5-4 PWM Control Register 3 (CNTR3)



	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CNTR3	0 0 2 9H	—	OE2	OE3	CH12	—	—	—	—	-000----B
			R/W	R/W	R/W					

CH12	CH12PWM mode setting bit
0	Normal mode
1	CH12PWM mode

OE3	Output pin control bit 3
0	Uses the P37/PWM1 pin as a general port (P42).
1	Uses the P37/PWM1 pin in CH12PWM mode or as the CH1 interval timer or PWM output pin (PWM1).

OE2	Output pin control bit 2
0	Uses the P43/PWM2 pin as a general port (P34).
1	Uses the P43/PWM2 pin as the CH2 interval timer or PWM output pin (PT02).

R/W : Can be read and written
- : Unused
▨ : Initial value

Table 8.5-3 Functions of PWM Control Register 3 (CNTR3) Bits

Bit		Description
bit7	Unused	• The value read from this bit is undefined. • Writing to this bit has no effect on operation.
bit6	OE2: Output pin control bit 2	• This bit is used to set whether the P34/PT02 pin is to be used as a general port or a dedicated pin. • When this bit is set to "0," the P34/PT02 pin is used as a general port (P34). When this bit is set to "1," the P34/PT02 pin is used as a dedicated pin (PT02). • The PT02 pin is a CH2 output pin. When the interval timer function is used, a square wave is output. When the PWM timer function is used, a PWM wave is output.
bit5	OE3: Output pin control bit 3	• This bit is used to set whether the P37/PT01 pin is used as a general port or a dedicated pin. • When this bit is set to "0," the P37/PT01 pin is used as a general port (P37). When this bit is set to "1," the P37/PT01 pin is used as a dedicated pin (PT01). • The PT01 pin is a CH1 output pin. When the interval timer function is used, a square wave is output. When the PWM timer function or CH12PWM mode is used, a PWM wave is output.
bit4	CH12: PWM mode setting bit	• This bit is used to switch from normal mode to CH12PWM mode or vice versa. • Setting this bit to "0" sets CH1 and CH2 to operate independently. Setting this bit to "1" selects CH12PWM mode in which the "L" width is specified in CH1 and a cycle is specified in CH2. • The operating mode selection bits (PTX1 and PTX2 bits of CNTR1) are meaningless in CH12PWM mode. **Note:** When the CK12 bit of CNTR2 is "1" (CK12 mode is set), do not set this bit to "1." When the TIE1 or TIE2 bit of CNTR2 is "1" (the CH1 or CH2 counter is operating), do not rewrite this bit.
bit3 bit2 bit1 bit0	Unused	• Values read from these bits are undefined. • Writing to these bits has no effect on operation.

8.5.4 PWM Compare Register 1 (COMR1)

PWM compare register 1 (COMR1) is a CH1 data register. When the interval timer function is operating, the value set in COMR1 defines the interval time. When the normal PWM timer function is operating, the value is the "H" width of a pulse. When CH12PWM mode is used, the value is the "L" width of a pulse.

■ PWM Compare Register 1 (COMR1)

Figure 8.5-5 "PWM Compare Register 1 (COMR1)" shows the bit configuration of COMR1.

Bit manipulation instructions cannot be used for COMR1 because COMR1 is a write-only register.

Figure 8.5-5 PWM Compare Register 1 (COMR1)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
COMR1	0 0 2 AH	*								XXXXXXXXB
		W	W	W	W	W	W	W	W	

W : Write only
X : Undefined
* : This bit is meaningless when 7-bit PWM mode (high-speed mode) is used.

❍ When the interval timer is operating (8-bit timer mode and CK12 mode)

The value to be compared with the counter value is set in COMR1, which is used to specify the interval time (square wave output frequency).

When the value set in COMR1 matches the counter value, the counter is cleared and the interrupt request flag bit is set to "1" (TIR1 bit of CNTR2 = 1).

The value written to COMR1 when the counter is operating takes effect starting with the next cycle (after match detection).

Reference:

The value set in COMR1 when the 8-bit timer or CK12 mode is in effect can be calculated from the expression shown below. Note that the instruction cycle is affected by the clock mode and gear function.

Value set in COMR1 = interval time/(count clock cycle x instruction cycle) - 1

❍ When the PWM timer is operating (8-bit PWM mode and 7-bit PWM mode)

The value to be compared with the counter value is set in COMR1, which is used to specify the "H" width of a pulse.

"H" level is output to the PWM pin until the value set in COMR1 matches the counter value. When the value set in COMR1 matches the counter value, "L" level is output until the counter value overflows.

The value written to COMR1 when the counter is operating takes effect starting with the next cycle (after the counter value has overflowed).

Reference:

The value set in COMR1 when the PWM timer is operating and the PWM wave cycle can be calculated from the expressions shown below. Note that the instruction cycle is affected by the clock mode and gear function.

- 8-bit PWM mode

  Value set in COMR1 = duty ratio (%) x 256

  PWM wave cycle = count clock cycle x instruction cycle x 256

- 7-bit PWM mode

  Value set in COMR1 = duty ratio (%) x 128

  PWM wave cycle = count clock cycle x instruction cycle x 128

❍ During operation in CH12PWM mode

The value to be compared with the counter value is set in COMR1, which is used to specify the "L" width of a pulse.

"L" level is output to the PWM1 pin until the value set in COMR1 matches the counter value.

The value written to COMR1 when the counter is operating takes effect starting with the next cycle.

Reference:

The COMR1 setting value for operation in CH12PWM mode can be calculated from the equation shown below. However, note that the instruction cycle is affected by the clock mode and gear function.

COMR1setting value = "L" width time of PWM wave/(count clock cycle x instruction cycle) - 1

8.5.5 PWM Compare Register 2 (COMR2)

PWM compare register 2 (COMR2) is the CH2 data register. When the interval timer function is in operation, the value in COMR2 defines the interval time. When the normal PWM timer function is in operation, the value defines the "H" width of a pulse. In CH12PWM mode, the value defines the PWM wave cycle.

■ PWM Compare Register 2 (COMR2)

Figure 8.5-6 "PWM Compare Register 2 (COMR2)" shows the bit configuration of COMR2.

Bit manipulation instructions cannot be used for COMR2 because COMR2 is a write-only register.

Figure 8.5-6 PWM Compare Register 2 (COMR2)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
COMR2	0 0 2 B_H	*								$XXXXXXXX_B$
		W	W	W	W	W	W	W	W	

W : Write only
X : Undefined
* : This bit is meaningless when 7-bit PWM mode (high-speed mode) is used.

❍ When the interval timer is operating (8-bit timer mode and CK12 mode)

The value to be compared with the counter value is set in COMR2, which is used to specify the interval time.

When the value set in COMR2 matches the counter value, the counter is cleared and the interrupt request flag bit is set to "1" (TIR2 bit of CNTR2 = 1).

The value written to COMR2 when the counter is operating takes effect starting with the next cycle (after match detection).

Reference:

The COMR2 setting value for the case when the interval timer is in operation can be calculated from the equation shown below. However, note that the instruction cycle is affected by the clock mode and gear function.

- 8-bit timer mode

  COMR2 setting value = interval time/(count clock cycle x instruction cycle) - 1

- CK12 mode

  Value set in COMR2 = interval time/CH1 square wave output - 1

❍ When the PWM timer is operating (8-bit PWM mode and 7-bit PWM mode)

The value to be compared with the counter value is set in COMR2, which is used to specify the "H" width of a pulse.

"H" level is output to the PT02 pin until the value set in COMR2 matches the counter value.

The value written to COMR2 when the counter is operating takes effect starting with the next cycle (after the counter value has overflowed).

Reference:

The value set in COMR2 when the PWM timer is operating and the PWM wave cycle can be calculated as shown below. Note that the instruction cycle is affected by the clock mode and gear function.

- 8-bit PWM mode

  COMR2 setting value = duty ratio (%) x 256

  PWM wave cycle = count clock cycle x instruction cycle x 256

- 7-bit PWM mode

  COMR2 setting value = duty ratio (%) x 128

  PWM wave cycle = count clock cycle x instruction cycle x 128

❍ When CH12PWM mode is in effect

The value to be compared with the counter value is set in COMR2, which is used to specify a PWM wave cycle.

When the value set in COMR2 matches the counter value, the CH1 and CH2 counter values are cleared at the same time and the interrupt request flag bit (TIR2 bit of CNTR2) is set to "1."

In this case, the PT01 pin goes Low.

The value written to COMR2 when the counter is operating takes effect starting with the next cycle.

Reference:

The value set in COMR2 when CH12PWM mode is in effect can be calculated as shown below. Note that the instruction cycle is affected by the clock mode and gear function.

COMR2 setting value = 1-cycle time of PWM wave/(count clock cycle x instruction cycle) - 1

8.6 2-Channel 8-Bit PWM Timer Interrupts

The 2-channel 8-bit PWM timer causes an interrupt if the counter value matches the value set in the PWM compare register when the interval timer function is operating. If the PWM timer function is operating in a mode other than CH12PWM mode, interrupt requests do not occur.

■ Interrupts When the Interval Timer Function Is Operating and CH12PWM Mode Is in Effect

The counter value is incremented starting at 00_H, according to the selected count clock. When the counter value matches the value set in PWM compare register (COMR), the corresponding interrupt request flag bits are set to "1" (TIR1 and TIR2 bits of CNTR2).

In this case, when the interrupt request enable bit is set to "enable" (TIE2 bit of CNTR2 = 1), the 2-channel 8-bit PWM timer generates an interrupt request (IRQ6) for the CPU. Set the TIR bit to "0" and clear the interrupt request with the interrupt processing routine.

When the counter value matches the value set in COMR1 or COMR2, the TIR1 and TIR2 bits are set to "1" irrespective of the values of the TIE1 and TIE2 bits.

Note:

For CH1, interrupts that occur in CH12 PWM mode cannot be used, because interrupts are disabled (TIE1 bit of CNTR = 0). For CH2, interrupts can be used in the same way as during operation with the interval timer function.

Reference:

When the counter value matches the COMR value and at the same time the counter stops (TPE1 and TPE2 bits of CNTR2 = 0), the TIR bit are not set.

If the TIE bit changes from "0" (disabled) to "1" (enabled) when the TIR bit are "1," an interrupt request is generated immediately.

■ Register and Vector Table Related to 2-Channel 8-Bit PWM Timer Interrupts

Table 8.6-1 Register and Vector Table Related to 2-Channel 8-Bit PWM Timer Interrupts

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Setting bit		Upper	Lower
IRQ5	ILR2 ($007C_H$)	L51 (bit3)	L50 (bit2)	$FFF0_H$	$FFF1_H$
IRQ6	ILR2 ($007C_H$)	L61 (bit5)	L60 (bit4)	$FFEE_H$	$FFEF_H$

For information on the operation of interrupts, see Section 3.4.2 "Interrupt Processing."

8.7 Interval Timer Function Operation

This section explains the operation of the interval timer function of the 2-channel 8-bit PWM timer in 8-bit timer mode and CK12 mode.

■ Operation of Interval Timer Function

To use the 2-channel 8-bit PWM timers (CH1 and CH2) as the interval timer function in 8-bit timer mode and CK12 mode, the settings shown in Figure 8.7-1 "Settings of Interval Timer Function" must be made.

Figure 8.7-1 Settings of Interval Timer Function

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR1	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20
CH1-------	0	×	0	×	◎	◎	×	×
CH2-------	×	0	×	0	×	×	◎	◎
CK12-------	0	0	0	0	◎	◎	×	×

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR2	TPE1	TPE2	CK12	—	TIR1	TIR2	TIE1	TIE2
CH1-------	1	×	0		◎	×	◎	×
CH2-------	×	1	0		×	◎	×	◎
CK12-------	1	1	1		◎	◎	◎	◎

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR3	—	OE2	OE3	CH12	—	—	—	—
CH1-------		×	◎	0				
CH2-------		◎	×	0				
CK12-------		◎	◎	1				

COMR1	Setting of CH1 interval time (compare value)
COMR2	Setting of CH2 interval time (compare value)

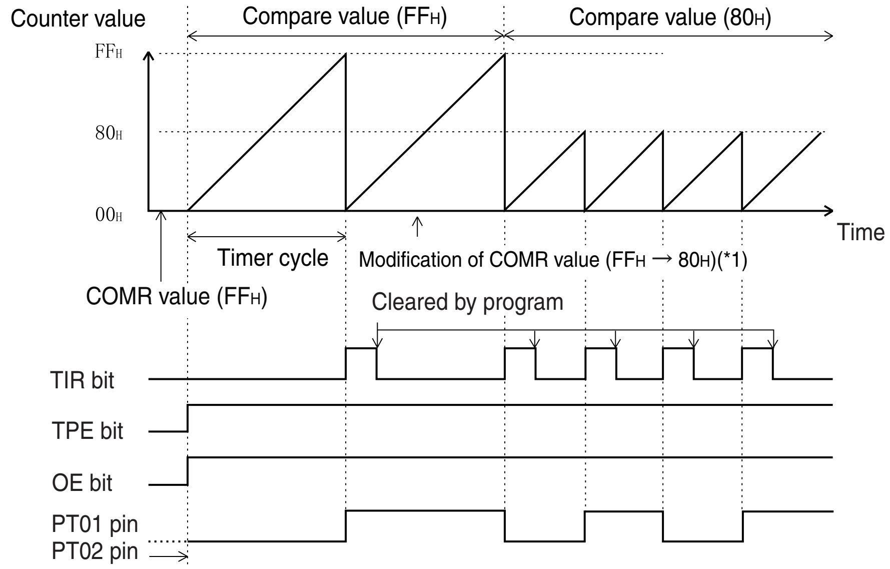
◎	: Used
X	: Unused (0 is set.)
1	: "1" is set.
0	: "0" is set.
CH1	: CH1 in 8-bit timer mode
CH2	: CH2 in 8-bit timer mode
CK12	: CK12 mode

When the counter is activated, incrementing begins on the rising edge of the selected count clock, starting at 00_H. When the counter value matches the value (compare value) set in COMR, the level of the PWM pin is inverted on the rising edge of the next count clock and the counter is cleared. Then, the interrupt request flag bit is set (TIR1 and TIR2 bits of CNTR2 = 1) and the count operation is restarted at 00_H.

In CK12 mode, the CH1 square wave that is output functions as the CH2 count clock.

Figure 8.7-2 "Operation of the 2-Channel 8-Bit PWM Timer" shows operation of the 2-channel 8-bit PWM timer.

Figure 8.7-2 Operation of the 2-Channel 8-Bit PWM Timer



When the output pin control bit (OE) is "0," this pin functions as a general I/O port.

*1 If the compare register (COMR) value is modified while the counter is operating, the interval timer function takes effect starting with the next cycle.

Note:

When the interval timer function is operating (TPE1 and TPE2 bits of CNTR2 = 1), do not change the corresponding count clock cycle (SC11 and SC10 bits or SC21 and SC20 bits of CNTR1). In CK12 mode, however, the SC21 and SC20 bits are meaningless.

Reference:

Setting the COMR value to "00_H" causes the PT01 or PT02 pin output to be inverted in the selected count clock cycle.

When the interval timer function is operating, the output level of the PT01 or PT02 pin when the counter is stopped (TPE1 and TPE2 bits of CNTR2 = 0) is Low.

8.8 Explanation of the 2-Channel 8-Bit PWM Timer Operation in 8-Bit PWM Mode

This section explains the operation of the 2-channel 8-bit PWM timer in 8-bit PWM mode.

■ PWM Timer Function Operation

For operation of the 2-channel 8-bit PWM timers (CH1 and CH2) as the PWM timer function in 8-bit PWM mode, the settings shown in Figure 8.8-1 "Settings of 8-Bit PWM Mode" must be made.

Figure 8.8-1 Settings of 8-Bit PWM Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR1	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20
CH1-------	1	×	0	×	◎	◎	×	×
CH2-------	×	1	×	0	×	×	◎	◎

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR2	TPE1	TPE2	CK12	—	TIR1	TIR2	TIE1	TIE2
CH1-------	1	×	0		×	×	×	×
CH2-------	×	1	0		×	×	×	×

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR3	—	OE2	OE3	CH12	—	—	—	—
CH1-------		×	1	0				
CH2-------		1	×	0				

COMR1	Setting of "H" width of CH1 pulse (compare value)
COMR2	Setting of "H" width of CH2 pulse (compare value)

◎ : Used
X : Unused (0 is set.)
1 : "1" is set.
0 : "0" is set.
CH1: CH1 in 8-bit timer mode
CH2: CH2 in 8-bit timer mode

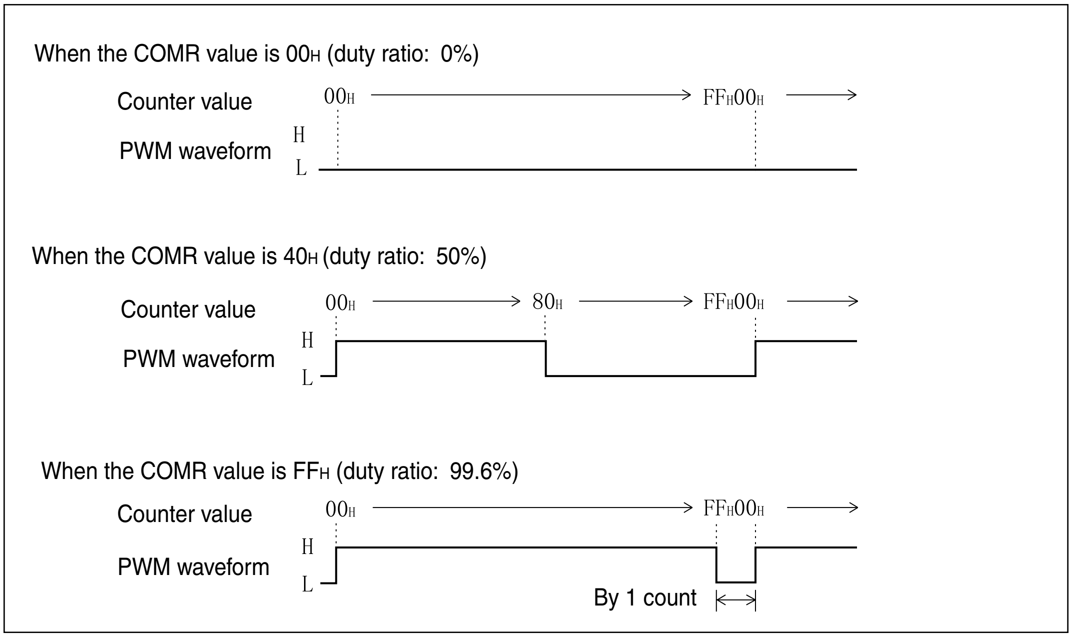
As soon as the counter starts, incrementing begins at the rising edge of the selected count clock, starting at 00_H. The PT01, PT02 pin output (PWM waveform) is kept at the "H"-level till the counter value matches the value in COMR.

After that, the output of the PWM pin is kept at the "L"-level until a counter overflow occurs (FF_H-->00_H).

If CH1 operates in 8-bit timer mode and CH2 operates in the PWM mode after the CK12 bit has been set to "1," the 2-channel 8-bit PWM timer enters CK12PWM mode.

Figure 8.8-2 "Example of PWM Waveform Output in 8-Bit PWM Mode" shows the PWM waveforms output to the PT01, PT02 pin.

Figure 8.8-2 Example of PWM Waveform Output in 8-Bit PWM Mode


When the COMR value is 00H (duty ratio: 0%)
Counter value
00H
FFH00H
PWM waveform
H
L
When the COMR value is 40H (duty ratio: 50%)
Counter value
00H
80H
FFH00H
PWM waveform
H
L
When the COMR value is FFH (duty ratio: 99.6%)
Counter value
00H
FFH00H
PWM waveform
H
L
By 1 count

Note:

When the PWM timer function is operating (TPE1 and TPE2 bits of CNTR2 = 1), do not change the corresponding count clock cycle (SC11 and SC10 bits or SC21 and SC20 bits of CNTR1).

When the CH1 PWM timer is operating, CK12 mode (CK12 bit of CNTR2 = 1) cannot be set.

Reference:

When the PWM timer function is operating, the PT01, PT02 pin output when the counter is stopped (TPE1 bit of CNTR2 = 1, TPE2 bit of CNTR2 = 0) is kept at the level in effect before the counter stopped.

8.9 2-Channel 8-Bit PWM Timer Operation in 7-Bit PWM Mode

This section explains the operation of the 2-channel 8-bit PWM timer in 7-bit PWM mode (high-speed mode).

■ High-Speed PWM Timer Function Operation

For operation of 2-channel 8-bit PWM timers (CH1 and CH2) as the PWM timer function in 7-bit PWM mode, the settings shown in Figure 8.9-1 "Settings of 7-Bit PWM Mode" must be made.

Figure 8.9-1 Settings of 7-Bit PWM Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR1	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20
CH1-------	1	×	1	×	◎	◎	×	×
CH2-------	×	1	×	1	×	×	◎	◎

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR2	TPE1	TPE2	CK12	—	TIR1	TIR2	TIE1	TIE2
CH1-------	1	×	0		×	×	×	×
CH2-------	×	1	0		×	×	×	×

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR3	—	OE2	OE3	CH12	—	—	—	—
CH1-------		×	1	0				
CH2-------		1	×	0				

	bit7	
COMR1	—	Setting of "H" width of CH1 pulse (compare value)

	bit7	
COMR2	—	Setting of "H" width of CH2 pulse (compare value)

◎ : Used
X : Unused (0 is set.)
1 : "1" is set.
0 : "0" is set.
CH1: CH1 in 7-bit timer mode
CH2: CH2 in 7-bit timer mode

When the counter is started, incrementing begins at the rising edge of the selected count clock, starting with 00_H. The PT01, PT02 pin output (PWM waveform) is kept at the "H"-level till the counter value matches the value in COMR.

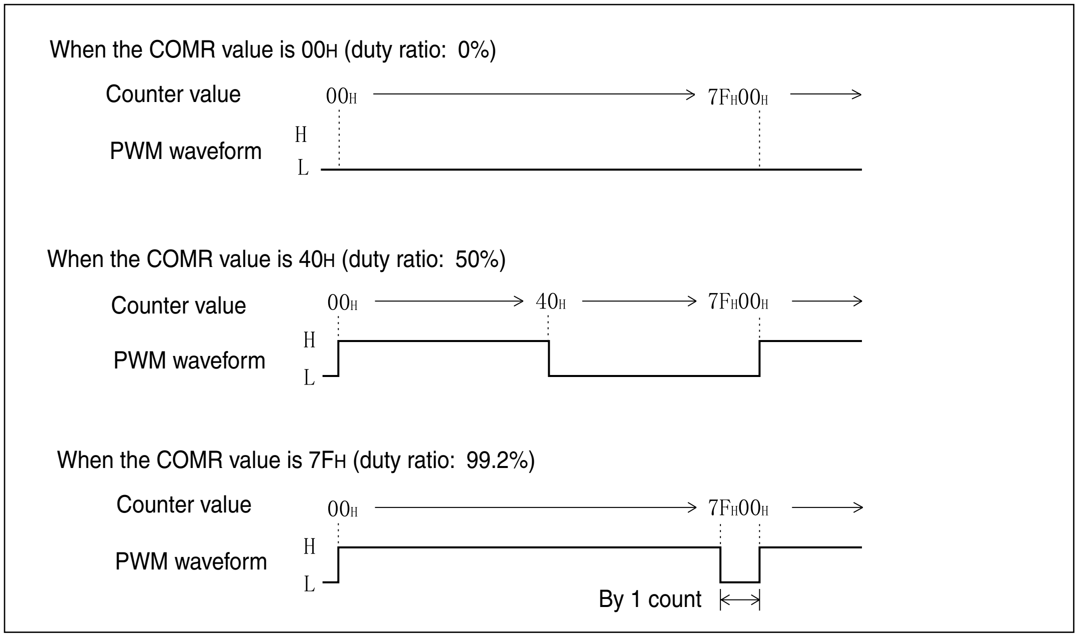
Afterwards, the PWM pin output is kept at the "L"-level till a counter overflow occurs ($7F_H$--> 00_H).

The number of counter bits in 7-bit PWM mode is one bit less than that in the 8-bit PWM mode. For this reason, the operation speed in 7-bit PWM mode becomes twice as fast as that in the 8-bit PWM mode (the cycle time in 7-bit PWM mode is one half of the cycle time in 8-bit PWM mode).

If CH1 is operated in 8-bit timer mode and CH2 is operated in PWM mode after the CK12 bit has been set to "1," the 2-channel 8-bit PWM timer enters the CK12PWM mode.

Figure 8.9-2 "Example of PWM Waveform Output in 7-Bit PWM Mode" shows the PWM waveforms output to the PT01, PT02 pin.

Figure 8.9-2 Example of PWM Waveform Output in 7-Bit PWM Mode


When the COMR value is 00H (duty ratio: 0%)
Counter value
00H
7FH00H
PWM waveform
H
L
When the COMR value is 40H (duty ratio: 50%)
Counter value
00H
40H
7FH00H
PWM waveform
H
L
When the COMR value is 7FH (duty ratio: 99.2%)
Counter value
00H
7FH00H
PWM waveform
H
L
By 1 count

Note:

When the PWM timer function is operating (TPE1 and TPE2 bits of CNTR2 = 1), do not change the corresponding count clock cycle (SC11 and SC10 bits or SC21 and SC20 bits of CNTR1).

When the CH1 PWM timer is operating, CK12 mode (CK12 bit of CNTR2 = 1) cannot be set.

Reference:

When the PWM timer function is operating, the PT01, PT02 pin output when the counter is stopped (TPE1 bit of CNTR2 = 1, TPE2 bit of CNTR2 = 0) is kept at the level in effect before the counter stopped.

8.10 Explanation of the 2-Channel 8-Bit PWM Timer Operation in CH12PWM Mode

This section explains the operation of the 2-channel 8-bit PWM timer in CH12PWM mode.

■ 2-Channel 8-Bit PWM Timer Operation in CH12PWM Mode

For operation of 2-channel 8-bit PWM timers (CH1 and CH2) as the PWM timer function in CH12PWM mode, the settings shown in Figure 8.10-1 "Settings of CH12PWM Mode" must be made.

Figure 8.10-1 Settings of CH12PWM Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR1	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20
	×	×	0	0	◎	◎	◎	◎

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR2	TPE1	TPE2	CK12	—	TIR1	TIR2	TIE1	TIE2
	1	1	0		×	◎	0	◎

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR3	—	OE2	OE3	CH12	—	—	—	—
		0	1	1				

COMR1	Setting of "L" width of PWM wave (compare value)
COMR2	Setting of 1 cycle of PWM wave (compare value)

◎ : Used
X : Unused (0 is set.)
1 : "1" is set.
0 : "0" is set.

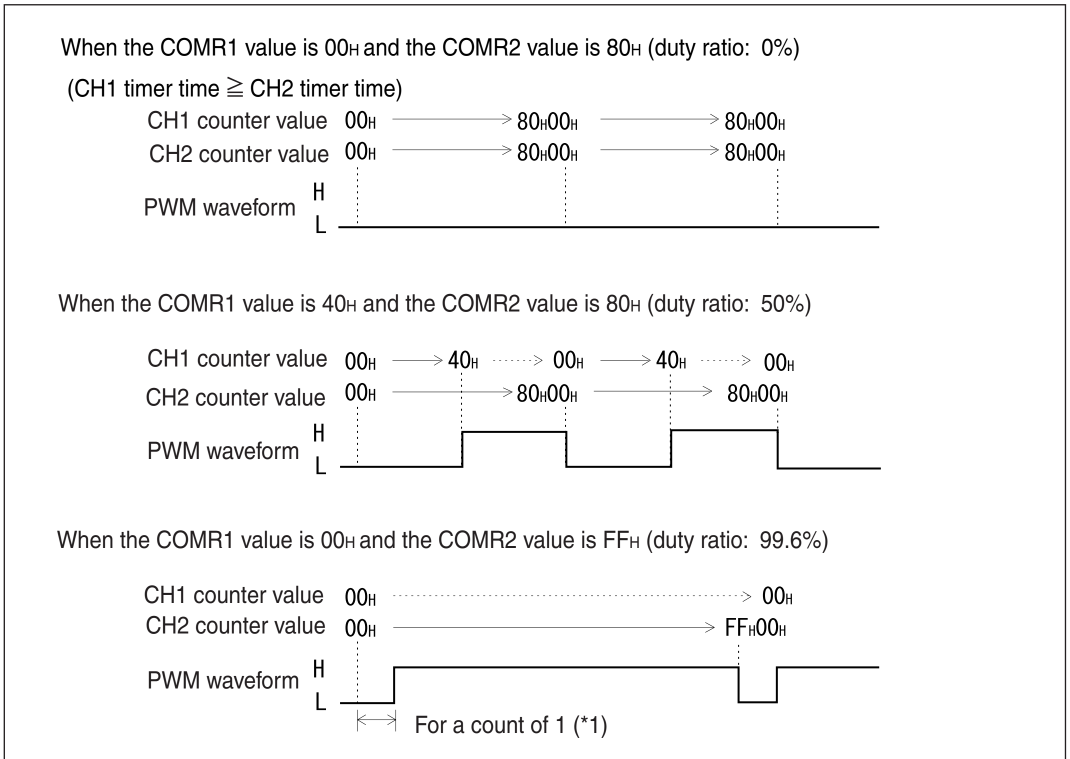
At the start of the CH1 or CH2 counter, incrementing begins with the rising edge of the selected count clock, starting at 00_H. The PT01 pin output (PWM waveform) is kept at "L"-level till the CH1 counter value matches the value in COMR1. When the CH1 counter value matches the value set in COMR1, the PWM pin output becomes "H"-level. When the CH2 counter value matches the value in COMR2, the CH1 and CH2 counters are cleared at the same time and counting is restarted at 00_H. In this case, the PT01 pin output becomes "L"-level and the interrupt request flag bit is set (TIR2 bit of CNTR2 = 1).

Start the CH1 and CH2 counters at the same time (TPE1 and TPE2 bits of CNTR2 = 1). Starting these counters separately would have an effect on the "L" width of the first cycle of a PWM wave or on its cycle time.

If the CH1 timer period that determines the "L" width of the PWM wave becomes greater than the CH2 timer period that determines the cycle of a PWM wave, PWM wave output is disabled.

Figure 8.10-2 "Example of PWM Waveform (PWM pin) Output" shows the PWM waveforms output to the PT01 pin when the same count clock cycle is selected for CH1 and CH2.

Figure 8.10-2 Example of PWM Waveform (PWM pin) Output


When the COMR1 value is 00H and the COMR2 value is 80H (duty ratio: 0%)
(CH1 timer time ≧ CH2 timer time)
CH1 counter value 00H → 80H00H → 80H00H
CH2 counter value 00H → 80H00H → 80H00H
PWM waveform
H
L
When the COMR1 value is 40H and the COMR2 value is 80H (duty ratio: 50%)
CH1 counter value 00H → 40H → 00H → 40H → 00H
CH2 counter value 00H → 80H00H → 80H00H
PWM waveform
H
L
When the COMR1 value is 00H and the COMR2 value is FFH (duty ratio: 99.6%)
CH1 counter value 00H → 00H
CH2 counter value 00H → FFH00H
PWM waveform
H
L
For a count of 1 (*1)

*1 The minimum value of the "L" width can be reduced by making the FCH1 count clock faster than the CH2 count clock.

Note:

When the PWM timer function is operating (TPE1 and TPE2 bits of CNTR2 = 1), do not change the count clock cycle (SC11 and SC10 bits or SC21 and SC20 bits of CNTR1).

When the 2-channel 8-bit PWM timer is operating in CH12PWM mode, CK12 mode (CK12 bits of CNTR21 and CNTR22 = 1) cannot be set.

Disable output of CH2 interrupt requests (TIE1 bit of CNTR2 = 0)

Reference:

When the PWM timer function is operating, the PT01 pin output when the counter is stopped (TPE1 bit of CNTR2 = 1, TPE2 bit of CNTR2 = 0) is kept at the level in effect before the counter stopped.

8.11 Explanation of the Prescaler Operation of 2-Channel 8-Bit PWM Timer

This section explains the prescaler operation of the 2-channel 8-bit PWM timer.

■ Prescaler Operation

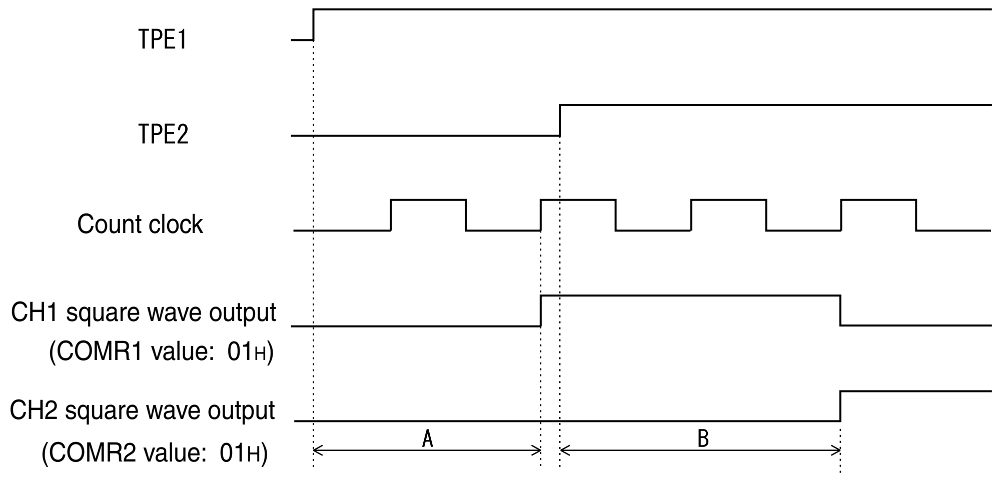
Prescaler operation of the 2-channel 8-bit PWM timer is enabled when the TPE1 or TPE2 bit of CNTR2 (counter operation enable bit) of PWM control register 2 is "1."

For this reason, when the TPE1 and TPE2 bits are set to "1" at the same time, the operations with respect to both CH1 and CH2 are completely identical to those in the first cycle.

However, if the operation of another counter is enabled when the TPE1 or TPE2 bit is already "1," the first cycle will contain a difference of up to one count clock cycle, because incrementing is started asynchronously.

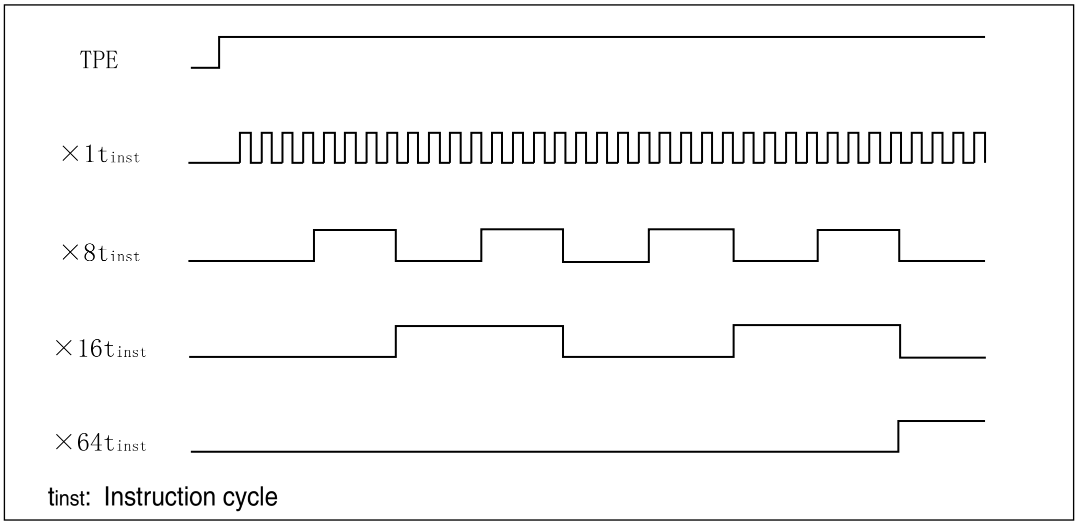
Figure 8.11-1 "Prescaler Operation" shows prescaler operation.

Figure 8.11-1 Prescaler Operation



Note: B produces a difference that is within 1 count clock cycle for A.

Figure 8.11-2 Prescaler Output



tinst: Instruction cycle

8.12 State of the 2-Channel 8-Bit PWM Timer Operation in Each Mode

This section explains the operation of the 2-channel 8-bit PWM timer when the timer switches to sleep or stop mode or when a stop request before completion is issued during operation of the interval timer function or PWM timer function.

■ Operation in Standby Mode and for a Stop before Completion

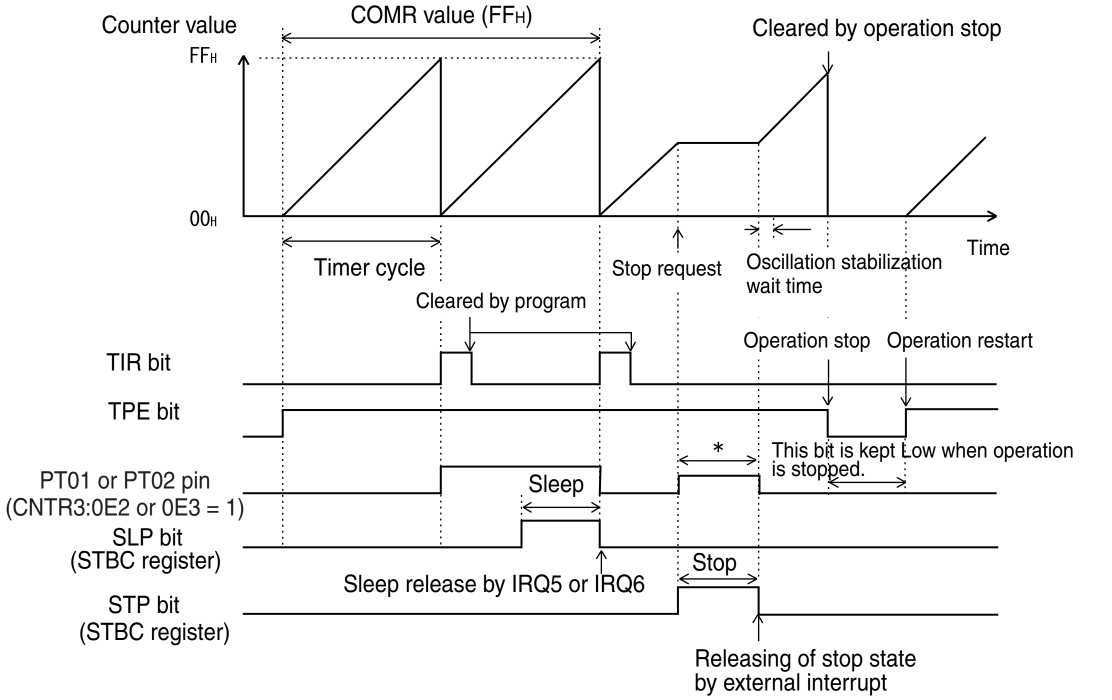
Figure 8.12-1 "Counter Operation in Standby Mode and for a Stop before Completion (When Interval Timer Function is Operating)" and Figure 8.12-2 "Counter Operation in Standby Mode and for a Stop before Completion (When PWM Timer Function Is Operating)" show the counter value states if the 2-channel 8-bit PWM timer switches to sleep or stop mode or a stop request before completion is issued when the interval timer function and PWM timer function are operating.

When the 2-channel 8-bit PWM timer switches to stop mode, the counter retains the value and stops. If stop mode is released by an external interrupt, the counter starts operation at the retained value. For this reason, the first interval time and PWM wave cycle do not become the values that have been set. After stop mode has been released, reinitialize the 8-bit PWM timer.

Release from watch mode (TMD bit of STBC = 1) is performed in the same way as a release from stop mode. Watch mode is released by a watch interrupt or an external interrupt.

❍ When the interval timer function is in operation

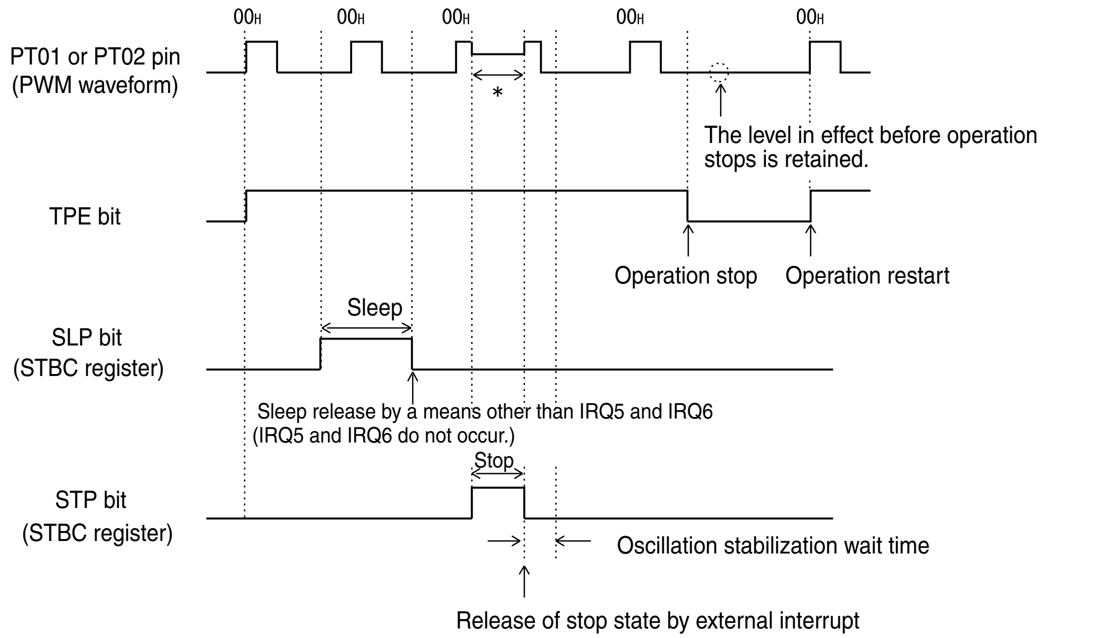
Figure 8.12-1 Counter Operation in Standby Mode and for a Stop before Completion (When Interval Timer Function is Operating)


Counter value
COMR value (FFH)
Cleared by operation stop
FFH
00H
Time
Timer cycle
Stop request
Oscillation stabilization wait time
Cleared by program
Operation stop
Operation restart
TIR bit
TPE bit
*
This bit is kept Low when operation is stopped.
PT01 or PT02 pin (CNTR3:0E2 or 0E3 = 1)
Sleep
SLP bit (STBC register)
Stop
Sleep release by IRQ5 or IRQ6
STP bit (STBC register)
Releasing of stop state by external interrupt

*1 When the pin state specification bit (SPL) of the standby control register (STBC) is "1," the PT01, PT02 pin, which is in stop mode, is set to high impedance.

❍ When the PWM timer function is operating

Figure 8.12-2 Counter Operation in Standby Mode and for a Stop before Completion (When PWM Timer Function Is Operating)


00H
00H
00H
00H
00H
PT01 or PT02 pin
(PWM waveform)
*
The level in effect before operation stops is retained.
TPE bit
Operation stop
Operation restart
Sleep
SLP bit
(STBC register)
Sleep release by a means other than IRQ5 and IRQ6
(IRQ5 and IRQ6 do not occur.)
Stop
STP bit
(STBC register)
Oscillation stabilization wait time
Release of stop state by external interrupt

*1 When the pin state specification bit (SPL) of the standby control register (STBC: SPL) is "1," the PT01 or PT02 pin, which is in stop mode, is set to high impedance.
When the SPL bit is "0," the value in effect before the two-channel 8-bit PWM timer switches to stop mode is retained.

8.13 Notes on Using the 2-Channel 8-Bit PWM Timer Usage

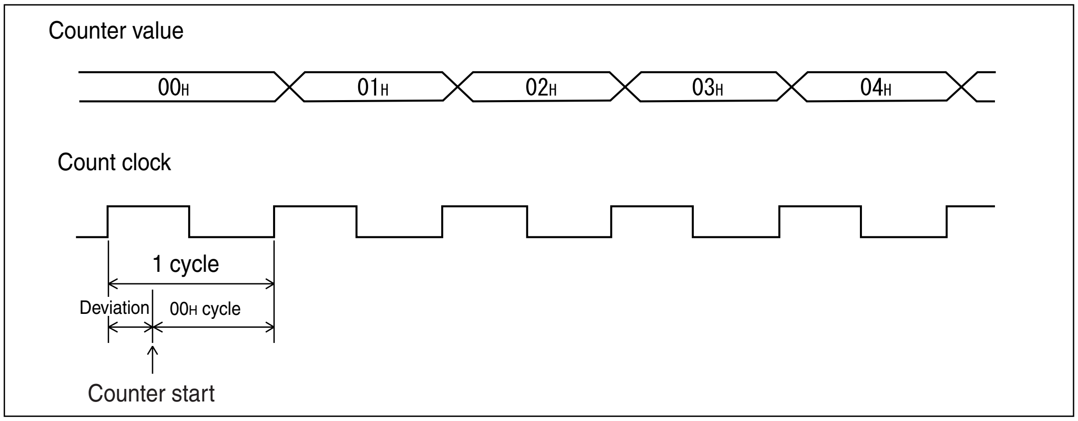
This section provides notes on using the 2-channel 8-bit PWM timer.

■ Notes on Using the 2-Channel 8-Bit PWM Timer

❍ Deviation

Start of the counter by a program is asynchronous with the start of incrementing the counter with the selected count clock. For this reason, the deviation that exists till the counter value matches the value in the PWM compare register (COMR) may become shorter by up to one count clock cycle. Figure 8.13-1 "Deviation that Remains till the Start of Count Operation" shows a deviation that exists till the start of count operation.

Figure 8.13-1 Deviation that Remains till the Start of Count Operation



❍ Notes on program settings

- When the interval timer function and PWM timer function are operating (TPE1 and TPE2 bits of CNTR2 = 1), do not modify the corresponding count clock cycle (SC11 and SC10 bits or SC21 and SC20 bits of CNTR1).
- Switch the interval timer function and PWM timer function (CNTR1:PTX1, PTX2) when the counter is stopped (CNTR2:TPE1 = 0, TPE2 = 0), interrupts are disabled (CNTR2:TIE1 = 0, TIE2 = 0), and interrupt requests have been cleared (CNTR2:TIR1 = 0, TIR2 = 0).
- In CK12 mode (CK12 bit of CNTR2 = 1), do not set CH12PWM mode (CH12 bit of CNTR3 = 1) and CH1 PWM timer operation (PTX1 bit of CNTR1 = 1).
- In CK12PWM mode, disable output of CH2 interrupt requests (TIE1 bit of CNTR2 = 0). Also, do not set CK12 mode.
- If the interrupt request flag bits (TIR1 and TIR2 bits of CNTR2) are "1" and the interrupt request enable bits are set to "enable" (TIE1 and TIE2 bits of CNTR2 = 1), control cannot return from interrupt processing. In this case, be sure to clear the TIR1 or TIR2.
- When the counter value matches the value set in COMR and at the same time the counter stops (TPE1 and TPE2 bits of CNTR2 = 0), the TIR1 or TIR2 bit is not set.

8.14 Program Examples of the 2-Channel 8-Bit PWM Timer (Interval Timer Function)

This section provides examples of 8-bit timer mode programs and CK12 mode programs for the 2-channel 8-bit PWM timer.

■ Examples of 8-Bit Timer Mode Programs

❍ Processing specifications

- CH1 operates as an interval timer in 8-bit timer mode.
- 5 ms interval timer interrupts are generated repeatedly.
- The square wave to be inverted at the interval time is output to the PT01 pin.
- The COMR1 value whose interval time is about 5 ms when the oscillation of the main clock is 10 MHz is shown below. The count clock is 64 t_{inst} of the internal count clock (t_{inst}: Divide-by-4 of oscillation when the main clock speed (gear) is set as the maximum speed).

COMR1 value = 5ms/ (64 x 4/10MHz) -1 = 194.3 (0C2H)

❍ Coding example (comply with Softune V1)

```
CNTR1   EQU    0027H              ;Address of PWM control register 1
CNTR2   EQU    0028H              ;Address of PWM control register 2
CNTR3   EQU    0029H              ;Address of PWM control register 3
COMR1   EQU    002AH              ;Address of PWM compare register 1
TPE1    EQU    CNTR2:7            ;Definition of CH1 counter operation enable bit
TIR1    EQU    CNTR2:3            ;Definition of CH2 interrupt request flag bit
ILR2    EQU    007C               ;Address of interrupt level setting register
INT_V   DSEG   ABS                ; [DATA SEGMENT]
        ORG    OFFE0H
IRQ5    DW     WARI1              ;Setting of interrupt vector
INT_V   ENDS
;-------------Main program-------------------------------------------------------
        CSEG                      ; [CODE SEGMENT]
                                  ;We assume here that the stack pointer (SP)
                                   register is already initialised.
         :
        CLRI                      ;Disable interrupts.
        CLRB    TPE1              ;Stop counter operation.
        MOV     ILR3,#11110111B   ;Set interrupt level (level 1).
        MOV     COMR1,#0C2H       ;Compare the value in COMR1 with the counter
                                   value (interval time).
        MOV     CNTR1,#00001100B  ;Interval timer operation, clearing of the
                                   CH1 clock, and 64tinst selection.
        MOV     CNTR3,#00100000B  ;Enable output to the PWM1 pin.
        MOV     CNTR2,#10000000B  ;Start counter operation and output an
                                   interrupt request.
        SETI                      ;Enable interrupts.
         :
;-----------Interrupt program----------------------------------------------------
WARI1   CLRB    TIR1              ;Clear the interrupt request flag.
        PUSHW   A
        XCHW    A,T               ;Save A and T.
        PUSHW   A
         :
        User processing
         :
        POPW    A
        XCHW    A,T               ;Return A and T.
        POPW    A
        RETI
        ENDS
;--------------------------------------------------------------------------------
```

■ Examples of CK12 Mode Programs

❍ Processing specifications

- CH1 assumes that the interval timer time is 5 ms (square wave output cycle = 10 ms) and that interrupts are not used.
- CH2 causes a 100 ms interval timer interrupt repeatedly on the assumption that CH1 is a count clock.
- The square wave to be inverted at the CH2 interval time is output to the PT02 pin.
- The COMR1 value for which the CH1 timer interval time becomes about 5 ms at a main clock oscillation of 10 MHz is shown below. The count clock is 64 tinst of the internal count clock (tinst: Divide-by-4 of oscillation when the main clock speed (gear) is set to the maximum speed).

COMR1 value = 5ms/ (64 x 4/10MHz) -1 = 194.3 (0C2H)

- The COMR2 value whose CH2 timer interval time is about 100 ms when a CH1 square wave is output is shown below.

COMR2 value = 100ms/ (5 x 4/2) ms = 10 (00AH)

❍ Coding example (comply with Softune V1)

```
CNTR1   EQU    0027H             ;Address of PWM control register 1
CNTR2   EQU    0028H             ;Address of PWM control register 2
CNTR3   EQU    0029H             ;Address of PWM control register 3
COMR1   EQU    002AH             ;Address of PWM compare register 1
COMR2   EQU    002BH             ;Address of PWM compare register 2
TPE1    EQU    CNTR2:7           ;Definition of CH1 counter operation enable bit
TPE2    EQU    CNTR2:6           ;Definition of CH2 counter operation enable bit
TIR2    EQU    CNTR2:2           ;Definition of CH2 interrupt request flag bit
ILR2    EQU    007C              ;Address of interrupt level setting register
INT_V   DSEG   ABS               ; [DATA SEGMENT]
        ORG    OFFEEH
IRQ6    DW     WARI              ;Setting of interrupt vector
INT_V   ENDS
;-------------Main program-------------------------------------------------------
        CSEG                     ; [CODE SEGMENT]
                                 ;We assume here that the stack pointer (SP)
                                  register is already initialised.
         :
        CLRI                     ;Disable interrupts.
        CLRB    TPE1             ;Stop counter operation.
        CLRB    TPE2
        MOV     ILR2,#11110111B  ;Set interrupt level (level 1).
        MOV     COMR1,#0C2H      ;Compare the value in COMR1 with the counter
                                  value (interval time).
        MOV     COMR2,#00AH
        MOV     CNTR1,#00001100B ;Interval timer operation, clearing of the
                                  CH1 clock, and 64tinst selection.
        MOV     CNTR3,#01000000B ;Enable output to the PWM2 pin.
        MOV     CONT2,#11100001B ;Start counter operation and output an
                                  interrupt request.
        SETI                     ;Enable interrupts.
         :
;-----------Interrupt program----------------------------------------------------
WARI1   CLRB    TIR2             ;Clear the CH2 interrupt request flag.
        PUSHW   A
        XCHW    A,T              ;Save A and T.
        PUSHW   A
         :
        User processing
         :
        POPW    A
        XCHW    A,T              ;Return A and T.
        POPW    A
        RETI
        ENDS
;--------------------------------------------------------------------------------
        END
```

8.15 Program Examples of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)

This section provides program examples for the PWM timer function in 8-bit PWM mode, 7-bit PWM mode, and CH12PWM mode PWM of the 2-channel timer.

■ Examples of PWM Timer Function Programs

❍ Processing specifications

- CH1 operates as a PWM timer in 8-bit PWM mode and square waves are output to the PT01 pin.
- CH2 operates as a PWM timer in 7-bit PWM mode (high-speed mode) and square waves are output to the PT02 pin.
- PWM waves with a 50% duty ratio are generated and the duty ratio is subsequently changed to 25%.
- No interrupts are generated.
- If the oscillation of the main clock is 10 MHz, each count clock is 16 t_{inst} of the internal count clock (t_{inst}: Divide-by-4 of oscillation when the main clock speed (gear) is set to the maximum speed), the CH1 PWM wave cycle becomes 16 x 4/10 MHz x 256 = 1.6384 ms and the CH2 PWM wave cycle becomes 16 x 4/10 MHz x 128 = 0.8192 ms.
- The COMR1 value whose duty ratio is 50% in 8-bit PWM mode is shown below.

COMR1 value = 50/100 x 256 = 128 (080H)

- The COMR2 value whose duty ratio is 50% in 7-bit PWM mode is shown below.

COMR2 value = 50/100 x 128 = 64 (040H)

❍ Coding example

```
CNTR1   EQU    0027H              ;Address of PWM control register 1
CNTR2   EQU    0028H              ;Address of PWM control register 2
CNTR3   EQU    0029H              ;Address of PWM control register 3
COMR1   EQU    002AH              ;Address of PWM compare register 1
COMR2   EQU    002BH              ;Address of PWM compare register 2
TPE1    EQU    CNTR2:7            ;Definition of CH1 counter operation enable bit
TPE2    EQU    CNTR2:6            ;Definition of CH2 counter operation enable bit
;-------------Main program------------------------------------------------------
        CSEG                      ; [CODE SEGMENT]
         :
        CLRB    TPE1              ;Stop counter operation.
        CLRB    TPE2
        MOV     COMR1,#80H        ;Specify the "H" width of a pulse and a duty
                                   ratio of 50%.
        MOV     COMR2,#40H        ;Specify 1 cycle of a pulse.
        MOV     CNTR1,#11011010B  ;Select PWM timer, 8/7-bit PWM mode and
                                   16 tinst.
        MOV     CNTR3,#01100000B  ;Enable output to the PT01 and PT02 pin
        MOV     CNTR2,#11000000B  ;Start counter operation and disable output of
                                   interrupt requests.
         :
         :
        MOV     COMR1,#40H        ;Change the duty ratio to 25% (takes effect
                                   starting with the next PWM wave cycle).
        MOV     COMR2,#20H
         :
        ENDS
;-------------------------------------------------------------------------------
        END
```

■ Program Examples for CH12PWM Mode

❍ Processing specifications

- CH1 is set to 8-bit PWM mode to enable CH1 to operate as the PWM timer.
- CH2 is set to 8-bit PWM mode to enable CH2 to operate as the PWM timer.
- The initial duty ratio of the PWM value is set to 50%.
- No interrupts are generated.
- When the oscillation clock frequency is 10 MHz, each clock is 16t_{inst} (t_{inst}: Divided-by-4 value of the oscillation when the main clock speed (gear) is set to the maximum speed).
- The COMR1 register value ("L" width) is set to 40H to set the duty ratio to 50%.
- The COMR2 register value (cycle interval) is set to 80H to set the duty ratio to 50%.

❍ Coding example

```
CNTR1   EQU    0027H               ;Address of PWM control register 1
CNTR2   EQU    0028H               ;Address of PWM control register 2
CNTR3   EQU    0029H               ;Address of PWM control register 3
COMR1   EQU    002AH               ;Address of PWM compare register 1
COMR2   EQU    002BH               ;Address of PWM compare register 2
TPE1    EQU    CNTR2:7             ;Definition of CH1 counter operation enable bit
TPE2    EQU    CNTR2:6             ;Definition of CH2 counter operation enable bit
;----------Main program-----------------------------------------------------------
        CSEG                       ; [CODE SEGMENT]
         :
        CLRB    TPE1               ;Stop counter operation.
        CLRB    TPE2
        MOV     COMR1,#40H         ;Specify the "L" width of a pulse and a duty
                                    ratio of 50%.
        MOV     COMR2,#80H         ;Specify 1 cycle of a pulse.
        MOV     CNTR1,#11001010B   ;Enable the counter to operate as the PWM timer
                                    and select 16tinst.
        MOV     CNTR3,#00110000B   ;Enable output to the PT01 pin and set the
                                    CH12PWM mode.
        MOV     CNTR2,#11000000B   ;Start counter operation and disable output of
                                    interrupt requests.
         :
        ENDS
;---------------------------------------------------------------------------------
        END
```

CHAPTER 9 PULSE WIDTH COUNT TIMER (PWC)

This chapter describes the functions and operations of the pulse width count timer (PWC).

9.1 Overview of the Pulse Width Count Timer

The pulse width count timer (PWC) has two functions: it can act as an interval timer that counts down in synchronization with three types of internal count clock, and it can provide a pulse width measurement function for measuring the width of pulses input to an external pin. Each of the functions can be selected.
These functions allow the PWC to provide the settings for the 8-bit interval timer and to output square waves at a given frequency by using the output of this timer. The PWC can also be used for input capture by repeating external input pulse width measurements continuously.

■ Interval Timer Function (Square Wave Output Function)

In the interval timer function, the PWC repeatedly generates interrupts at given time intervals.

Also, the function can invert the output level of the pin (WT0) at each time interval to output square waves with a given frequency.

- The interval time for the timer function ranges from one clock to 28 clocks of the internal count clock.
- The internal count clock can be selected from three types.
- The timer function can operate in either reload timer mode (in continuous operation) or in one-shot mode (one-time operation).
- The P20/PWCK pin can be selected as the external clock of PWC.
- The timer output bit (PCR2:T0) can be used as the count clock of the 8-bit PWM timer.

Table 9.1-1 "Interval Times and Square Wave Output Ranges" lists the interval times and square wave output ranges.

Table 9.1-1 Interval Times and Square Wave Output Ranges

Internal count clock period	Interval time	Square wave output (Hz)
$1t_{inst}$	$1t_{inst}$ to 2^8t_{inst}	$1/(2t_{inst})$ to $1/(2^9t_{inst})$
$4t_{inst}$	2^2t_{inst} to $2^{10}t_{inst}$	$1/(2^3t_{inst})$ to $1/(2^{11}t_{inst})$
$32t_{inst}$	2^5t_{inst} to $2^{13}t_{inst}$	$1/(2^6t_{inst})$ to $1/(2^{14}t_{inst})$

t_{inst}: Instruction cycle (influenced by clock mode and others)

Reference:

Examples of interval time and square wave output frequency calculations are shown below.

When the main clock source oscillation (F_{CH}) is 10 MHz, the value of the PWC reload buffer register (RLBR) is DD_H (221), and the count clock period is 1 t_{inst}, the interval time and square wave output frequency can be calculated as follows:

Interval value $= (1 \times 4/F_{CH}) \times$ (RLBR register value)
$= (4/10\text{MHz}) \times 221$
$\fallingdotseq 88.8\mu s$

Output frequency $= F_{CH}$ / (1 x 8 x (RLBR register value)
$= 10\text{MHz}/ (8 \times 221)$
$\fallingdotseq 5.6\text{kHz}$

■ Pulse Width Measurement Function

The pulse width measurement function measures the width of "H" or "L" pulses, or one-cycle width of pulses input to an external pin (PWC pin).

- The function also allows consecutive measurement of multiple pulses.
- There are three measurement speeds (internal clock count types) that can be selected.
- Moreover, the width of a long input pulse can be measured as well.

Table 9.1-2 "Pulse Widths That Can Be Measured by the Pulse Width Measurement Function" lists the pulse widths that can be measured by the pulse width measurement function.

Table 9.1-2 Pulse Widths That Can Be Measured by the Pulse Width Measurement Function

Internal count clock period	Interval time
$1t_{inst}$	$1t_{inst}$ to 2^8t_{inst}
$4t_{inst}$	2^2t_{inst} to $2^{10}t_{inst}$
$32t_{inst}$	2^5t_{inst} to $2^{13}t_{inst}$

t_{inst}: Instruction cycle (1/4 of the source oscillation of the main clock)

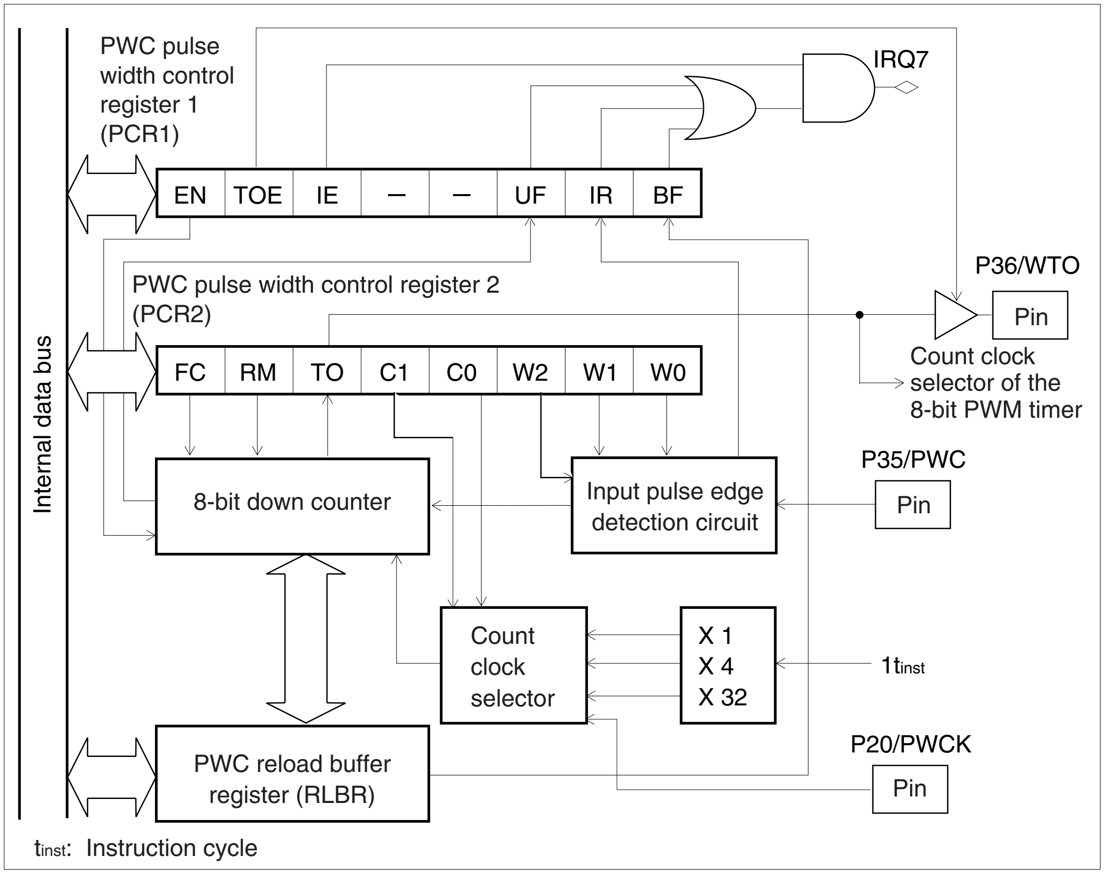
9.2 Configuration of the Pulse Width Count Timer

The pulse width count timer consists of the following six blocks:

- **Count clock selector**
- **8-bit down counter**
- **Input pulse edge detection circuit**
- **PWC reload buffer register (RLBR)**
- **PWC pulse width control register 1 (PCR1)**
- **PWC pulse width control register 2 (PCR2)**

■ Block Diagram of the Pulse Width Count Timer

Figure 9.2-1 Block Diagram of the Pulse Width Count Timer


PWC pulse width control register 1 (PCR1)
IRQ7
EN
TOE
IE
—
—
UF
IR
BF
P36/WTO
PWC pulse width control register 2 (PCR2)
Pin
Count clock selector of the 8-bit PWM timer
Internal data bus
FC
RM
TO
C1
C0
W2
W1
W0
8-bit down counter
Input pulse edge detection circuit
P35/PWC
Pin
Count clock selector
X 1
X 4
X 32
1tinst
P20/PWCK
PWC reload buffer register (RLBR)
Pin
tinst: Instruction cycle

❍ Counter clock selector

This circuit selects the count down clock for the 8-bit down counter from three types of usable internal count clocks and external clock P20/PWCK. To select the external clock, set the P20/ PWCK port as the input port (DDR2:bit 0 = 0).

❍ 8-bit down counter

This counter starts countdown either from the value obtained by subtracting 1 from the value of the PWC reload buffer register (RLBR) (when the interval timer function is used) or from FF_H (when the pulse width is measured). When the counter value changes from 00_H to FF_H, the counter inverts the timer output bit (PCR2:TO). The TO bit is output to the WTO pin by setting the output pin control bit to a dedicated pin (PCR1:TOE = 1).

❍ Input pulse edge detection circuit

This circuit operates when the pulse width measurement function is active. It controls the start and end of a countdown by the 8-bit down counter based on the input edge from the PWC pin that matches the edge set in PWC pulse width control register 2 (PCR2).

❍ PWC reload buffer register (RLBR)

In the reload timer mode of the interval timer function, when the counter value changes from 00_H to FF_H, the value obtained by subtracting 1 from the RLBR register value is again set in the counter. Counting continues.

In the case of the pulse width measurement function, the value of the 8-bit down counter is transferred to the RLBR register upon completion of the measurement.

❍ PWC pulse width control registers 1 and 2 (PCR1 and PCR2)

These registers are used to select the function of the PWC, set the operating conditions, enable or disable operation, exert interrupt control, and check the PWC status.

■ Interrupts Related to Pulse Width Count Timer

IRQ7:

When the count value changes from 01_H to 00_H during operation of the interval timer function or pulse width measurement function, PWC generates an interrupt request if interrupt request output is enabled (PCR1:IE = 1).

Also, PWC generates an interrupt request of the pulse width measurement function when: the pulse width measurement function completes the measurement of the pulse width, or interrupt request output is enabled (PCR1:IE = 1) with the pulse width measurement value retained in the RLBR register.

9.3 Pins of the Pulse Width Count Timer

This section describes the pulse width count timer pins and provides block diagrams of the pins.

■ Pins Related to the Pulse Width Count Timer

The pins related to the pulse width count timer are P35/PWC and P36/WT0.

❍ P35/PWC and P36/WT0 pins

The P35/PWC and P36/WT0 pins are used for multiple functions. P35 and P36 are used as general-purpose I/O ports, the PWC acts as an input pin for pulses to be measured, and WT0 acts as an output pin for the timer output bit (PCR2: T0).

PWC:

When the pulse width measurement function is selected, the counter measures the width of pulses that are input to this pin.

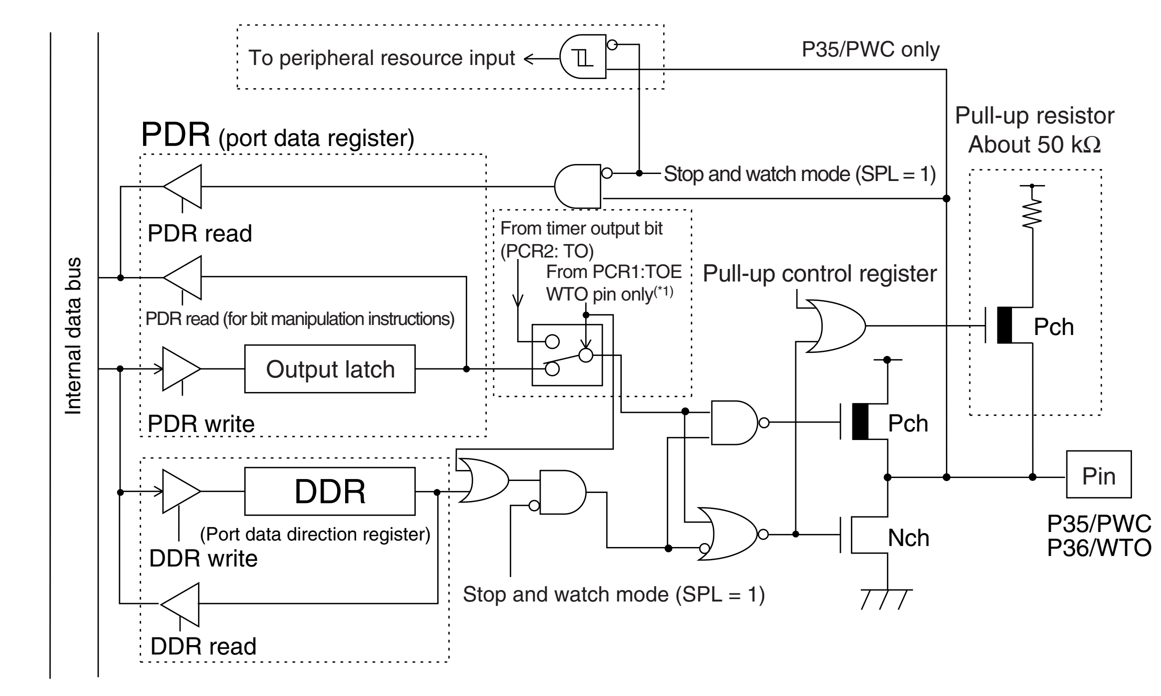
When using this pin as the PWC pin for the pulse width measurement function, set it to the input port (DDR3:bit 5 = 0) by using the port data direction register.

WT0:

When the interval timer function is selected, the output level of this pin is inverted if the counter value changes from 00_H to FF_H. In reload timer mode, the pin outputs square waves. If the output pin control bit is set as a dedicated pin (PCR1: T0E = 1), the P36/WT0 pin automatically serves as an output pin that functions as the WT0 pin, regardless of the value of the port direction register (DDR3: bit 6).

■ Block Diagrams of the Pins Related to the Pulse Width Count Timer

Figure 9.3-1 Block Diagram of the Pins Related to the Pulse Width Count Timer (P35/PWC, P36/WTO)

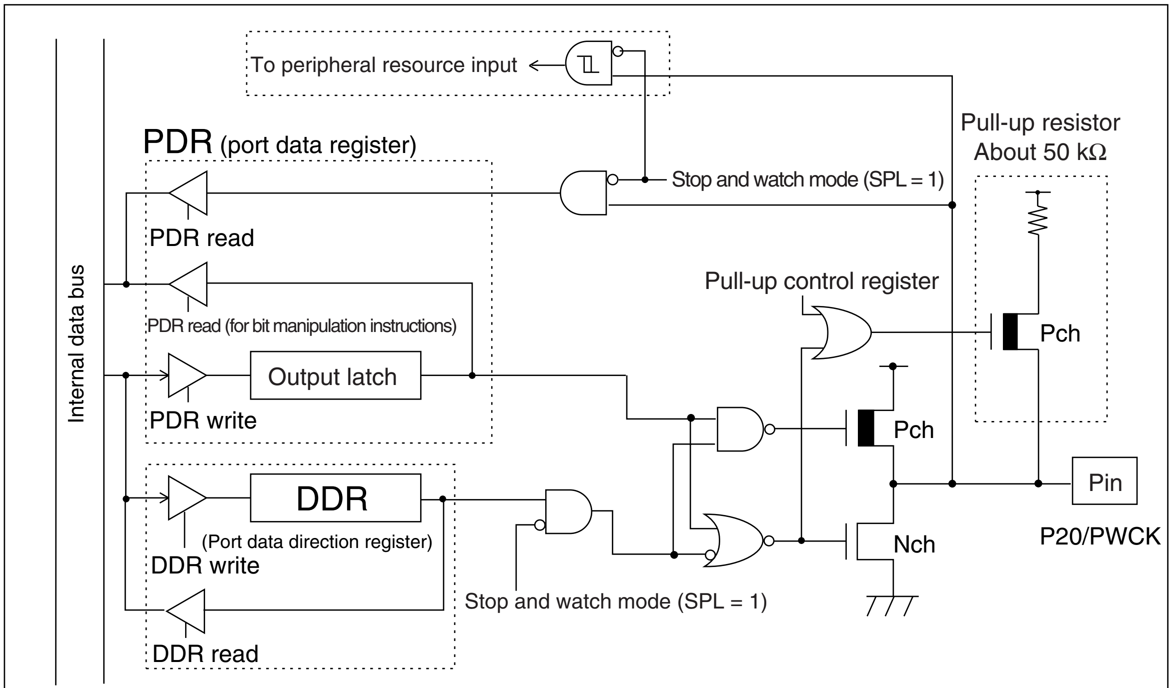


SPL: This bit specifies the pin state of the standby control register (STBC).
*1 P36/WTO only

Reference:

If "pull-up resistor available" is selected in the pull-up option setting register, the pin is set to the "H" level (pull-up state), not the high-impedance state, in stop or watch mode (STBC:SPL = 1). However, the pull-up state is disabled during a reset, in which case the pin enters the Hi-z state.

Figure 9.3-2 Block Diagram of Pins Related to Pulse Width Count Timer (P20/PWCK)



SPL: This bit specifies the pin state of the standby control register (STBC).

Reference:

If "pull-up resistor available" is selected in the pull-up register, the pin is set to the "H" level (pull-up state), not the high-impedance state, in stop or watch mode (STBC:SPL = 1). However, the pull-up state is disabled during a reset, in which case the pin enters the high-impedance state.

9.4 Registers of the Pulse Width Count Timer

This section describes the registers related to the pulse width count timer.

■ Registers Related to the Pulse Width Count Timer

Figure 9.4-1 Registers Related to the Pulse Width Count Timer

PCR1 and PCR2 (PWC pulse width control registers 1 and 2)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PCR1	002CH	EN	TOE	IE	—	—	UF	IR	BF	000XX000B
		R/W	R/W	R/W			R/W	R/W	R	

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PCR2	002DH	FC	RM	TO	C1	CO	W2	W1	WO	00000000B
		R/W	R/W	R/W		R/W	R/W	R/W	R/W	

RLBR (PWC reload buffer register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
002EH									XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	When the interval timer function is selected
	R	R	R	R	R	R	R	R	When the pulse width measurement function is selected

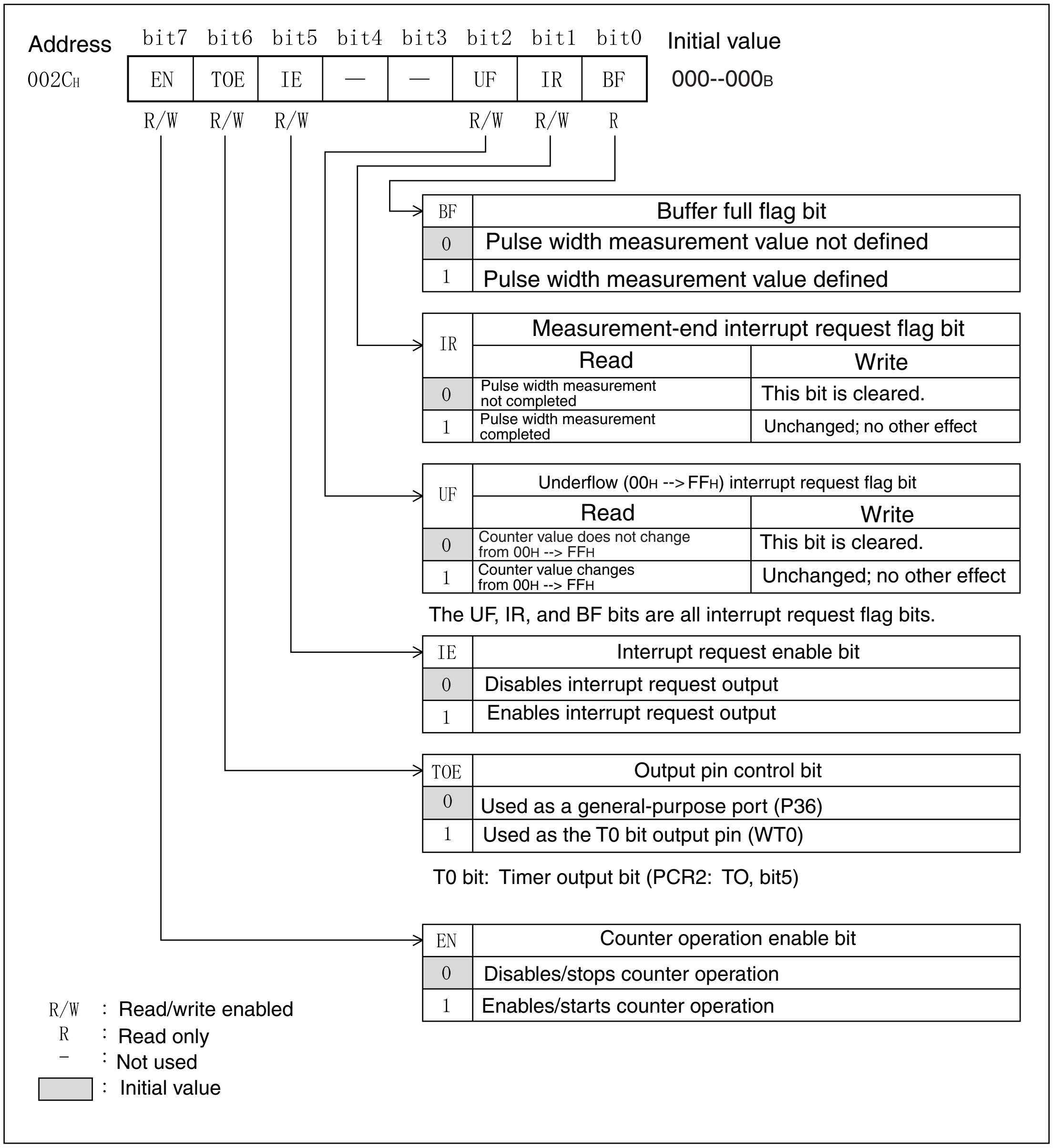
R/W: Read/write enabled
R : Read only
— : Not used
X : Undefined

9.4.1 PWC Pulse Width Control Register 1 (PCR1)

PWC pulse width control register 1 (PCR1) is used to enable or disable the functions of the pulse width count timer, exert interrupt control, and check the PWC status.

■ PWC Pulse Width Control Register 1 (PCR1)

Figure 9.4-2 PWC Pulse Width Control Register 1 (PCR1)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
002CH	EN	TOE	IE	—	—	UF	IR	BF	000--000B
	R/W	R/W	R/W			R/W	R/W	R	

BF	Buffer full flag bit
0	Pulse width measurement value not defined
1	Pulse width measurement value defined

IR	Measurement-end interrupt request flag bit	
	Read	Write
0	Pulse width measurement not completed	This bit is cleared.
1	Pulse width measurement completed	Unchanged; no other effect

UF	Underflow (00H --> FFH) interrupt request flag bit	
	Read	Write
0	Counter value does not change from 00H --> FFH	This bit is cleared.
1	Counter value changes from 00H --> FFH	Unchanged; no other effect

The UF, IR, and BF bits are all interrupt request flag bits.

IE	Interrupt request enable bit
0	Disables interrupt request output
1	Enables interrupt request output

TOE	Output pin control bit
0	Used as a general-purpose port (P36)
1	Used as the T0 bit output pin (WT0)

T0 bit: Timer output bit (PCR2: TO, bit5)

EN	Counter operation enable bit
0	Disables/stops counter operation
1	Enables/starts counter operation

R/W : Read/write enabled
R : Read only
– : Not used
▨ : Initial value

Table 9.4-1 Functions of the PWC Pulse Width Control Register 1 (PCR1) Bits

	Bit name	Function
bit7	EN: Counter operation enable bit	• If this bit is set to "1" when the interval timer function is selected, the counter starts countdown from the value set in the PWC reload buffer register (RLBR). Writing "0" stops counter operation. • If this bit is set to "1" when the pulse width measurement function is selected, measurement with the counter is enabled and, upon detection of the specified edge of the pulse to be measured, the counter starts countdown from FF_H. Writing "0" stops counter operation. **Note:** If the pulse width measurement function is selected and if counter operation is disabled (EN = 0) during measurement, counter operation will stop. However, the counter value is not transferred to the RLBR register in this case. Restarting the counter (EN =1) sets its value to FF_H, enabling counter operation.
bit6	TOE: Output pin control bit	• This bit specifies whether the P36/WT0 pin is to be used as a general-purpose port (T0E = 0) or a dedicated pin (interval timer output) (T0E = 1). • If the pin is set to be used as a dedicated pin, the value of the timer output bit (PCR2: T0) is output to the WT0 pin. **Reference:** If the output pin control bit is set to specify the use of the dedicated pin (T0E = 1), the pin functions as the WT0 pin, regardless of the state of the general-purpose port (P36).
bit5	IE: Interrupt request enable bit	• This bit enables or disables the output of interrupt requests to the CPU. • If this bit or any of the interrupt request flag bits (UF, IR, and BF) is set to "1," an interrupt request is output to the CPU.
bit4 bit3	Unused bits	• The values of these bits are undefined during read operations. • Write operations have no effect on the values of these bits.
bit2	UF: Underflow (00_H-->FF_H) interrupt request flag bit	• This bit is set to "1" if the counter value changes from 00_H to FF_H. • If both this bit and the interrupt request enable bit (IE) are set to "1," an interrupt request is output. • Writing "0" clears this bit. Writing "1" has no effect. **Reference:** • If the counter value changes from 00_H to FF_H when the interval timer function is selected, the timer output bit (PCR2: T0) is inverted. In reload timer mode, the counter continues countdown from the value set in the RLBR register. In one-shot timer mode, the counter automatically stops countdown (EN = 0). • If the counter value changes from 00_H to FF_H while the pulse width measurement function is measuring a long input pulse, the counter sets this bit to "1" and continues countdown.
bit1	IR: Measurement-end interrupt request flag bit	• This bit is used when the pulse width measurement function is selected. • The bit is set to "1" when pulse width measurement is completed. • If this bit and the interrupt request enable bit (IE) are set to "1," an interrupt request is output. • During a write operation, setting "0" clears this bit. Setting "1" has no effect, and the bit is unchanged. • This bit has no meaning when the interval timer function is selected.

Table 9.4-1 Functions of the PWC Pulse Width Control Register 1 (PCR1) Bits (Continued)

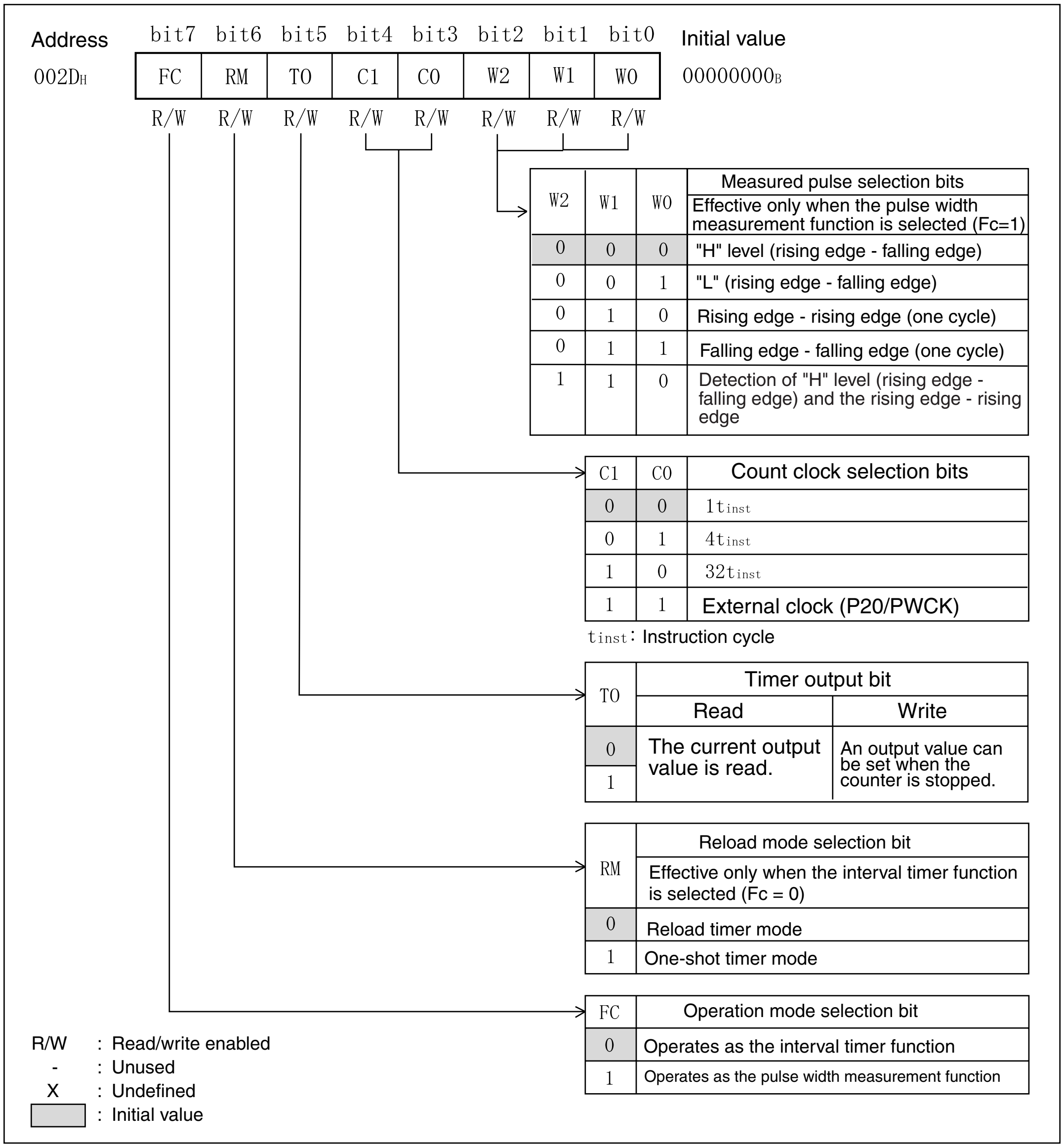
	Bit name	Function
bit0	BF: Buffer full flag bit	• This bit is an interrupt request flag that is set to "1" if the RLBR register retains a measurement value when the pulse width measurement function is selected. • If this bit and the interrupt request enable bit (IE) are set to "1," an interrupt request is output. • The bit is set to "1" when the pulse width measurement is completed. It is cleared to "0" when the measurement value is read from the RLBR register. • This is a read-only bit. Any attempt to write to this bit has no effect on counter operation. • This bit has no meaning when the interval timer function is selected.

9.4.2 PWC Pulse Width Control Register 2 (PCR2)

PWC pulse width control register 2 (PCR2) is used to select the operation mode of the pulse width count timer (pulse width measurement, interval timer operation, etc.), select the count clock type, set the pulse to be measured (measurement edge), and check the timer output status.

■ PWC Pulse Width Control Register 2 (PCR2)

Figure 9.4-3 PWC Pulse Width Control Register 2 (PCR2)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$002D_H$	FC	RM	TO	C1	C0	W2	W1	W0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

W2	W1	W0	Measured pulse selection bits Effective only when the pulse width measurement function is selected (Fc=1)
0	0	0	"H" level (rising edge - falling edge)
0	0	1	"L" (rising edge - falling edge)
0	1	0	Rising edge - rising edge (one cycle)
0	1	1	Falling edge - falling edge (one cycle)
1	1	0	Detection of "H" level (rising edge - falling edge) and the rising edge - rising edge

C1	C0	Count clock selection bits
0	0	$1t_{inst}$
0	1	$4t_{inst}$
1	0	$32t_{inst}$
1	1	External clock (P20/PWCK)

t_{inst}: Instruction cycle

TO	Timer output bit	
	Read	Write
0	The current output value is read.	An output value can be set when the counter is stopped.
1		

RM	Reload mode selection bit Effective only when the interval timer function is selected (Fc = 0)
0	Reload timer mode
1	One-shot timer mode

FC	Operation mode selection bit
0	Operates as the interval timer function
1	Operates as the pulse width measurement function

R/W	:	Read/write enabled
-	:	Unused
X	:	Undefined
▨	:	Initial value

Table 9.4-2 Functions of the PWC Pulse Width Control Register 2 (PCR2) Bits

Bit name		Function
bit7	FC: Operation mode selection bit	• This bit is used to toggle between the interval timer function (FC = 0) and the pulse width measurement function (FC = 1). **Note:** When the pulse width measurement function (FC = 1) is selected, specify the P35/PWC pin as an input port.
bit6	RM: Reload mode selection bit	• When the interval timer function is selected This bit is used to toggle between reload timer mode (RM = 0) and one-shot timer mode (RM = 1). • When the pulse width measurement function is selected This bit is meaningless.
bit5	TO: Timer output bit	• This bit is inverted each time the counter value changes from 00_H to FF_H. • When the output pin control bit (PCR1: T0E) of PWC pulse width control register 1 is set to "1," the value of this bit is output from the WT0 pin. • By counting the number of times this bit is inverted (the counter value changes from 00_H to FF_H), the timer can measure pulses longer than 28 clocks of the selected count clock. When the counter is stopped (PCR1: EN = 0) and when output is enabled (PCR1: T0E = 1), the value this bit is set to become the initial value for the WT0 pin.
bit4 bit3	C1, C0: Count clock selection bits	• These bits select the count clock for the interval timer and pulse width measurement functions. • Three types of internal count clock inputs can be set for port P20/PWCK. **Note:** To use the external clock, set the port (P20/PWCK) as the input port.
bit2 bit1 bit0	W2, W1, W0: Measured pulse selection bits	• When the pulse width measurement function is selected These bits enable selection of the pulse edge type used as the condition for starting or stopping the measurement of the pulse to be measured. These bits enable the setting of five pulse widths and a cycle. • When the interval timer function is selected These bits have no meaning.

Note:

Do not modify the PCR2 register data while the counter is operating (PCR1: EN = 1).

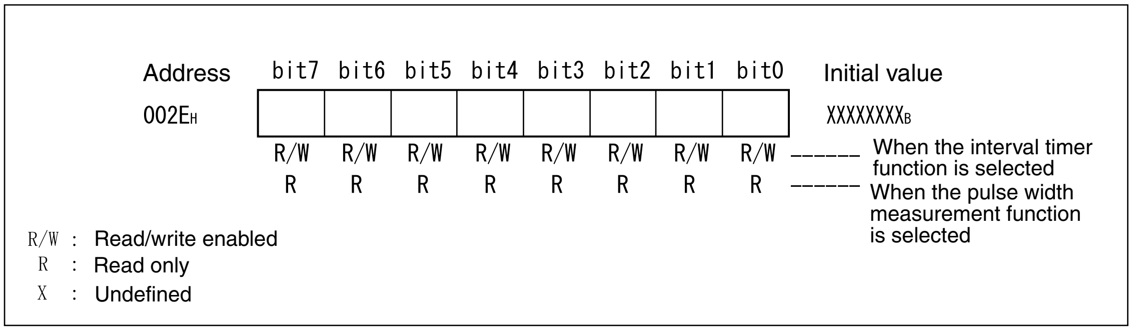
9.4.3 PWC Reload Buffer Register (RLBR)

The PWC reload buffer register (RLBR) functions as a reload register when the interval timer function is selected. When the pulse width measurement function is selected, it serves as a measured value storage register.

■ PWC Reload Buffer Register (RLBR)

Figure 9.4-4 "PWC Reload Buffer Register (RLBR)" shows the bit configuration of the PWC reload buffer register.

Figure 9.4-4 PWC Reload Buffer Register (RLBR)


Address
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
Initial value
002EH
XXXXXXXXB
R/W R/W R/W R/W R/W R/W R/W R/W
When the interval timer function is selected
R R R R R R R R
When the pulse width measurement function is selected
R/W : Read/write enabled
R : Read only
X : Undefined

❍ When the interval timer function is selected

The RLBR register functions as a reload register that specifies the interval time.

When counter operation is enabled (PCR1:EN = 1), the timer starts countdown from the value obtained by subtracting 1 from the value written into this register.

In reload timer mode, when the counter value changes from 00_H to FF_H, the value obtained by subtracting 1 from the RLBR register value is again set (reload) in the counter. The counter continues to count. Also, if a value is written to the RLBR register while the counter is operating, that value becomes valid the next time the value of the RLBR register is reloaded into the counter when the counter value changes from 00_H to FF_H.

Reference:

The value to be set in the RLBR register when the interval timer function is selected can be obtained from the following equation. Note that the instruction cycle is 1/4 ($4/F_{CH}$) of the source oscillation.

RLBR register value = interval time/(count clock period x instruction cycle)

❍ When the pulse width measurement function is selected

The RLBR register serves as a measured value storage register.

When pulse width measurement ends upon detection of the measurement-end edge, the counter value is transferred to this register.

At this time, the buffer full flag bit (PCR1: BF) and the measurement-end interrupt request flag bit (PCR1: IR) are set to "1." The BF bit is cleared to "0" when this register is read.

Reference:

When the pulse width measurement function is selected, the pulse width can be obtained from the RLBR register value using the following equation. Note that the instruction cycle is 1/4 ($4/F_{CH}$) of the source oscillation.

Pulse width = (256 - RLBR register value) x count clock period x instruction cycles

9.5 Pulse Width Count Timer Interrupts

A pulse width count timer interrupt can occur for the following two reasons:

- **The counter value changes from 01_H to FF_H while the interval timer function is operating.**
- **The measurement ends or the buffer becomes full while the pulse width measurement function is operating.**

■ Interrupts That Occur When the Interval Timer Function Is Selected

When the counter value changes from 00_H to FF_H after the counter starts countdown with the selected internal count clock, the underflow (00_H --> FF_H) interrupt request flag bit (PCR1: UF) is set to "1." At this time, an interrupt request (IRQ7) is sent to the CPU provided the interrupt request enable bit is "1" (PCR1: IE = 1). To clear the interrupt request, set the UF bit to "0" by using an interrupt processing routine.

Reference:

The UF bit is not set if a counter stop (PCR1: EN = 0) and a change of the counter value from 00_H to FF_H occur at the same time.

If the IE bit is changed from 0 (disable) to 1 (enable) when the UF bit is set to "1," an interrupt request is generated immediately.

■ Interrupts That Occur When the Pulse Width Measurement Function Is Selected

When the specified measurement-end edge is detected, the measurement-end interrupt request flag bit (PCR1: IR) and the buffer full flag bit (PCR1: BF) are set to "1." When the counter value changes from 00_H to FF_H because of a long pulse, the UF bit is set to "1." At this time, an interrupt request (IRQ7) is sent to the CPU if the interrupt request enable bit is set to 1 (PCR1: IE = 1). To clear the interrupt request, set the IR and UF bits to "0" by using an interrupt processing routine. Alternatively, read the PWC reload buffer register (RLBR) to clear the BF bit to "0."

Reference:

- The IR and BF bits of the PWC pulse width control register 1 are not set if a counter stop (PCR1:EN = 0) and the detection of a measurement-end edge occur at the same time.
- If the IE bit changes from "disabled" to "enabled (1)" when the IR, BF, or UF bit of PWC pulse width control register 1 is set to 1, an interrupt request is generated immediately.

■ Registers and Vector Tables Related to Pulse Width Count Timer Interrupts

Table 9.5-1 Registers and Vector Tables Related to Pulse Width Count Timer Interrupts

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Setting bit		Upper bits	Lower bits
IRQ7	ILR2 ($007C_H$)	L71 (bit7)	L70 (bit6)	$FFEC_H$	$FFED_H$

For information about interrupt processing, see Section 3.4.2 "Interrupt Processing."

9.6 Operation of the Interval Timer Function

This section describes the interval timer function of the pulse width count timer.

■ Operation of the Interval Timer Function

The interval timer function can operate in two modes. In reload timer mode, the function operates the timer continuously. In one-shot mode, the timer stops when one operation cycle is completed.

❍ Reload timer mode

To operate the timer in reload timer mode requires the setting shown in Figure 9.6-1 "Setting the Interval Timer Function (in Reload Timer Mode)"

Figure 9.6-1 Setting the Interval Timer Function (in Reload Timer Mode)

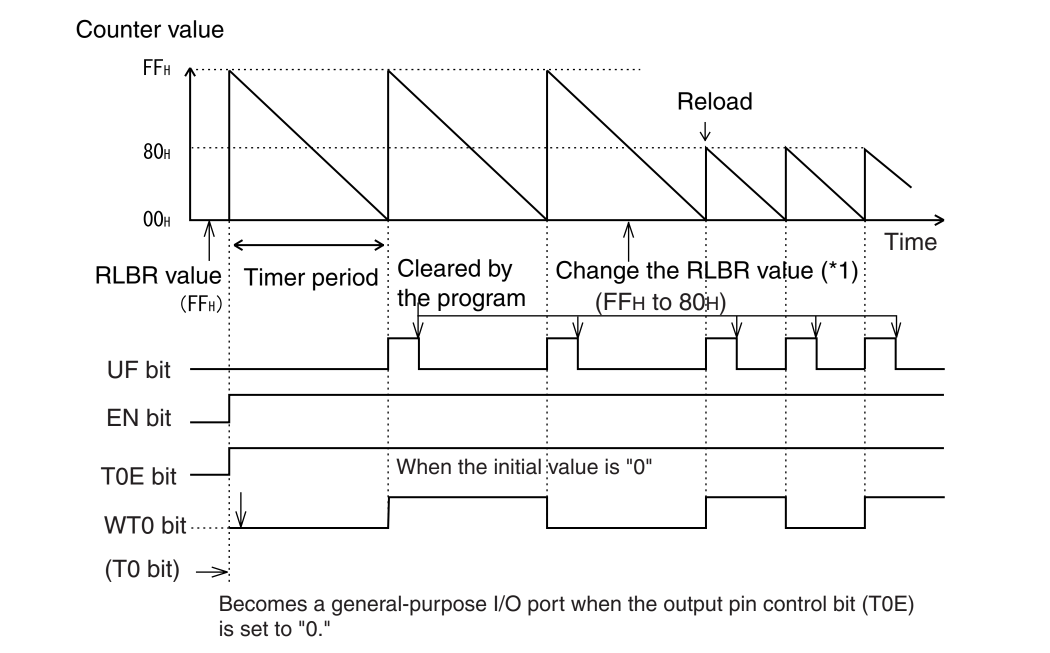
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PCR1	EN	TOE	IE	—	—	UF	IR	BF
	1	1	◎			◎	×	×
PCR2	FC	RM	TO	C1	CO	W2	W1	WO
	0	0	◎	◎	◎	×	×	×
RLBR	Set the interval time (initial counter value)							

◎: Used bit
×: Unused bit
1 : Set to "1"
0 : Set to "0"

When the counter is started, the value obtained by subtracting 1 from the RLBR register value is set in the counter. Countdown starts at the rising edge of the selected count clock. When the counter value changes from 01_H to 00_H, the value of the timer output bit (PCR2: T0) is inverted. After the counter is set again to the value of the RLBR register (the value is reloaded), the underflow (01_H --> 00_H) interrupt request flag bit is set (PCR1: UF = 1) at the rising edge of the next count clock.

Figure 9.6-2 "Timer Operation in Reload Timer Mode" shows the operation of the timer in reload timer mode.

Figure 9.6-2 Timer Operation in Reload Timer Mode


Counter value
FFH
80H
00H
Reload
Time
RLBR value
(FFH)
Timer period
Cleared by the program
Change the RLBR value (*1)
(FFH to 80H)
UF bit
EN bit
T0E bit
When the initial value is "0"
WT0 bit
(T0 bit)
Becomes a general-purpose I/O port when the output pin control bit (T0E) is set to "0."

*1: Any change in the value of the PWC reload buffer register (RLBR) during a cycle of timer operation becomes effective in the next cycle.

Reference:

Setting the value of the RLBR register to 01_H causes the T0 bit value to be inverted for one count clock cycle.

❍ **One-shot timer mode**

To operate the timer in one-shot timer mode requires the setting shown in Figure 9.6-3 "Setting the Interval Timer Function (One-Shot Timer Mode)".

Figure 9.6-3 Setting the Interval Timer Function (One-Shot Timer Mode)

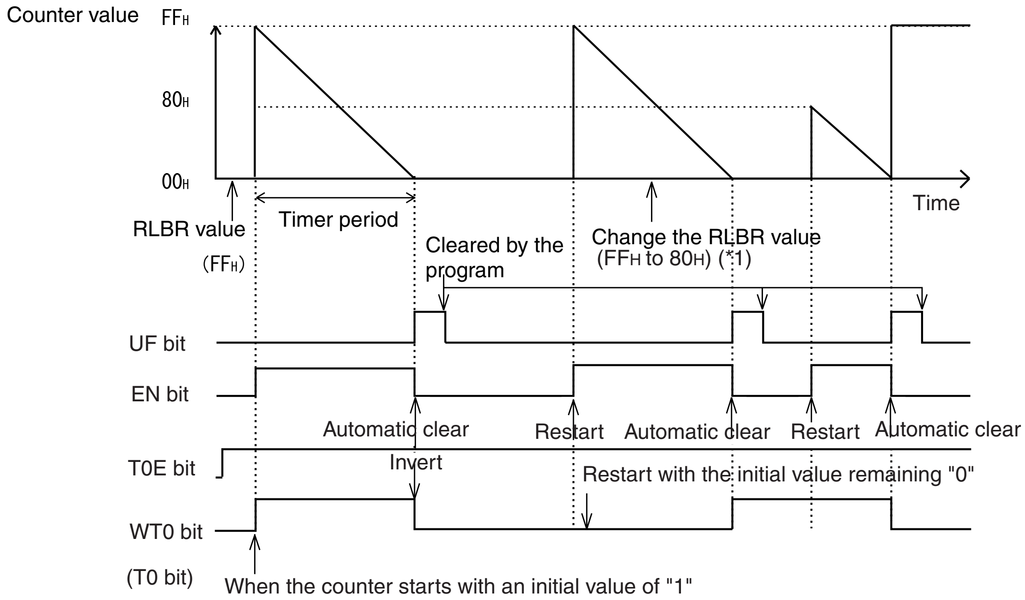
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PCR1	EN	TOE	IE	—	—	UF	IR	BF
	1	1	◎			◎	×	×
PCR2	FC	RM	TO	C1	C0	W2	W1	W0
	0	1	◎	◎	◎	×	×	×
RLBR	Set the interval time (initial counter value)							

◎ : Used bit
× : Unused bit
1 : Set to "1"
0 : Set to "0"

When the counter is started, the value stored in the RLBR register is set in the counter, which then starts countdown at the rising edge of the selected count clock. When the counter value changes from 00_H to FF_H, the value of the timer output bit (PCR2: T0) is inverted and the counter operation enable bit is automatically cleared (PCR1: EN = 0). After the operation of the counter is stopped, the underflow (00_H --> FF_H) interrupt request flag bit is set (PCR1: UF = 1) at the rising edge of the next count clock.

Figure 9.6-4 "Timer Operation in One-Shot Timer Mode" shows the operation of the timer in one-shot timer mode.

Figure 9.6-4 Timer Operation in One-Shot Timer Mode



*1: Any change in the value of the PWC reload buffer register (RLBR) during a cycle of timer operation becomes effective in the next cycle.

Note:

Do not change the PCR2 data while the counter is operating (PCR1: EN = 1).

Reference:

- The UF bit is set to "1" whenever the counter value changes from 00_H to FF_H, regardless of the value of the interrupt request enable bit (PCR1: IE).
- If the counter is stopped (PCR1: EN = 0) when the interval timer function is selected, the T0 bit retains the value stored immediately before counter stops.

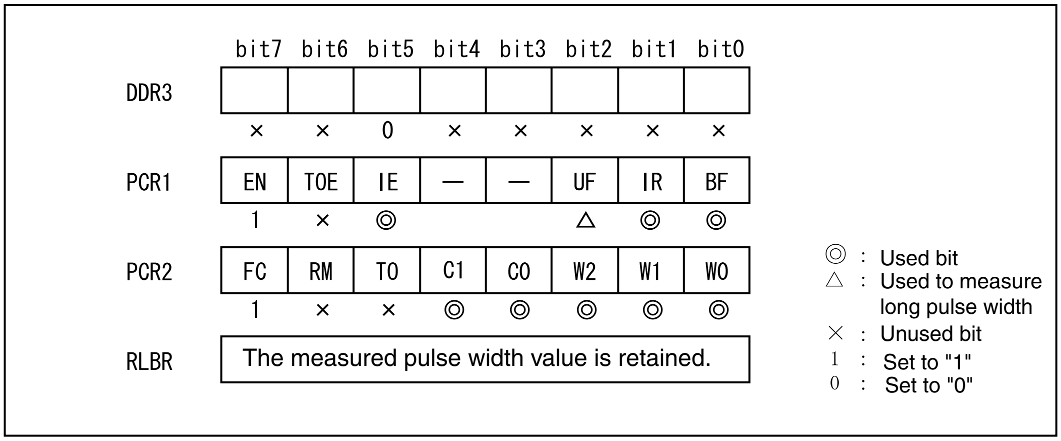
9.7 Operation of the Pulse Width Measurement Function

This section describes the operation of the pulse width measurement function of the pulse width count timer.

■ Operation of the Pulse Width Measurement Function

Operation of the pulse width measurement function requires the settings shown in Figure 9.7-1 "Settings for the Pulse Width Measurement Function".

Figure 9.7-1 Settings for the Pulse Width Measurement Function



	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR3								
	×	×	0	×	×	×	×	×
PCR1	EN	TOE	IE	—	—	UF	IR	BF
	1	×	◎			△	◎	◎
PCR2	FC	RM	TO	C1	C0	W2	W1	W0
	1	×	×	◎	◎	◎	◎	◎
RLBR	The measured pulse width value is retained.							

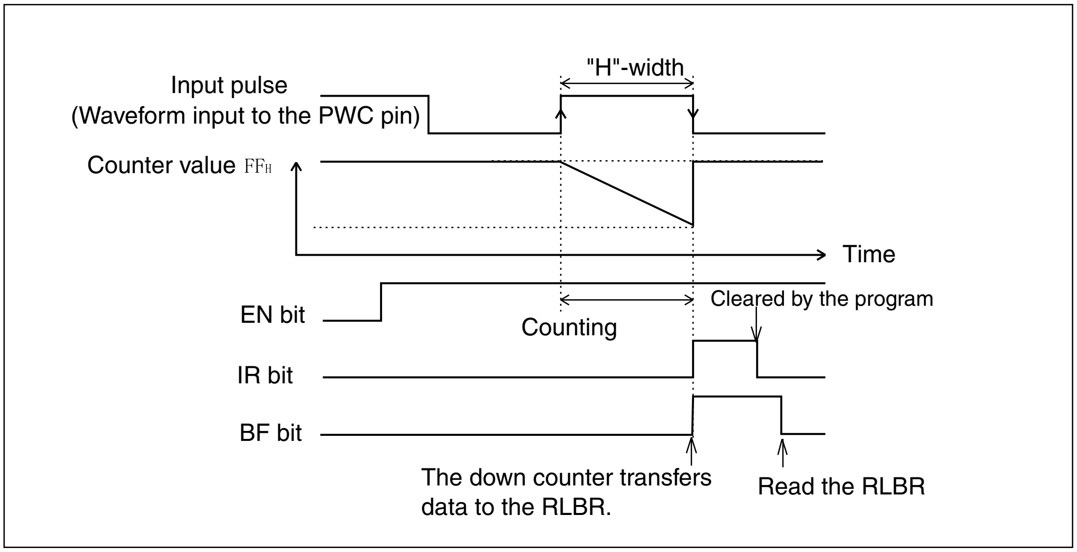
◎ : Used bit
△ : Used to measure long pulse width
× : Unused bit
1 : Set to "1"
0 : Set to "0"

When counter operation has been enabled, the counter starts countdown from FF_H when it detects a measurement-start edge in a pulse input to the PWC pin. (If the input is already "H" for measurement of the "H" width, the counter starts the measurement at the next rising edge.)

When a measurement-end edge is detected, the value of down counter at that point is transferred to the PWC reload buffer register (RLBR). The measurement-end interrupt request flag bit (PCR1: IR) and the buffer full flag bit (PCR1: BF) are set to "1," and counter operation is enabled again. (Since the pulse width measurement can be performed repeatedly, the counter can be used for input capture.)

Figure 9.7-2 "Example of "H"-Width Measurement When the Pulse Width Measurement Function is Selected" shows how the counter operates when the measured pulse selection bits (PCR2: W2, W1, and W0) are set to 000_B ("H" width measurement).

Figure 9.7-2 Example of "H"-Width Measurement When the Pulse Width Measurement Function is Selected



Note:

- If the previous RLBR register value is not read after the pulse width measurement has been performed repeatedly, the BF bit remains "1," retaining the previous measured value. New measured values are discarded.
- Do not change data in the PCR2 register while pulse width measurement is in progress (PCR1: EN = 1).

■ Measurement of Long Pulse Width

To measure a pulse whose width is greater than 28 times the selected count clock period, it is necessary to count either the number of times the value of the WT0 pin (PCR2: T0) is inverted by using an external circuit or the number of times the counter value changes from 00_H to FF_H by using an interrupt processing routine. Counting the number of times the counter value changes from 00_H to FF_H by software requires that a buffer (software counter) be allocated in RAM for storing the number of these events.

When the software counter has been initialized and counter operation is enabled, the counter starts countdown from FF_H when a measurement-start edge is detected in a pulse input to the PWC pin.

An interrupt request is generated if a measurement-end edge is detected or if the counter value changes from 00_H to FF_H. The interrupt processing routine checks the measurement-end interrupt request flag bit (PCR1: IR) and the underflow (00_H --> FF_H) interrupt request flag bit (PCR1: UF). If the UF bit is "1," the bit is set to "0" by writing to clear the interrupt request and, at the same time, the software counter is incremented (counter operation continues).

If the IR bit is set to "1," the pulse width, including the number of times the counter value changes from 00_H to FF_H, is calculated from the software counter value and the PWC reload buffer register (RLBR) value.

When the RLBR register value is 00_H, it is assumed to be 256.

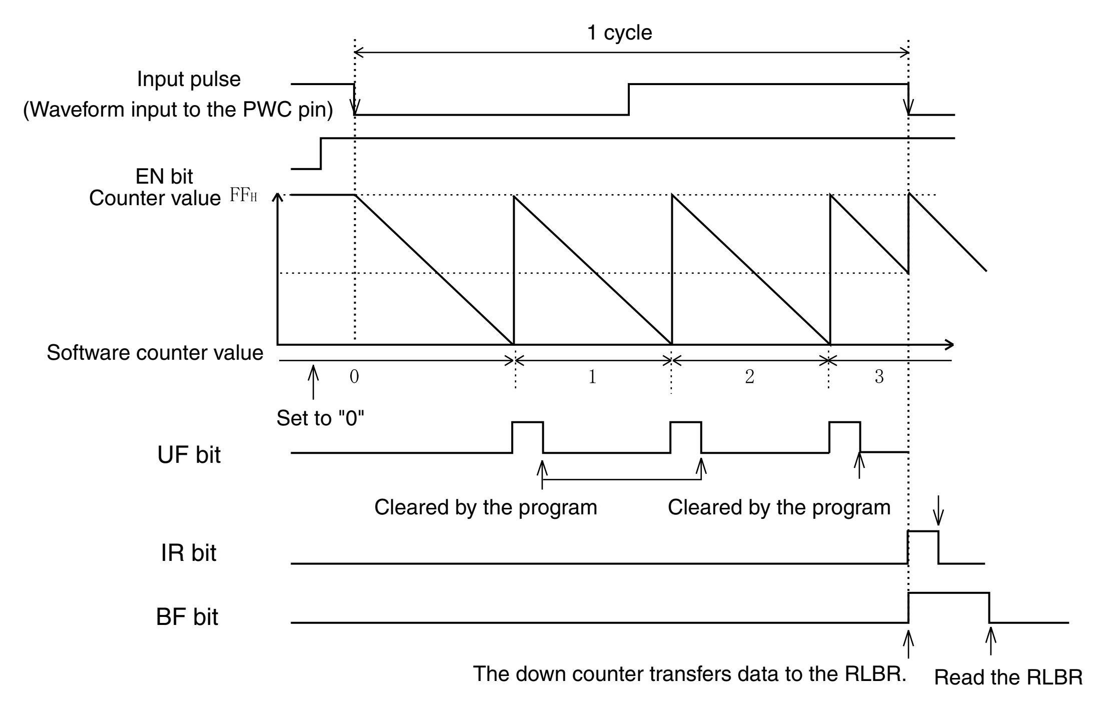
❍ Long pulse width calculation method

Pulse width = [(256 - RLBR register value) + (number of times the counter value changes from 00_H to FF_H x 256)]
x width of 1 cycle of the count clock

Calculate the pulse width before the next counter value change from 00_H to FF_H occurs. If calculated after the next counter value change from 00_H to FF_H, the pulse width value may not be accurate.

Figure 9.7-3 "Long Pulse Width Measurement" shows how the counter operates when the measured pulse selection bits (PCR2: W2, W1, and W0) are set to 011_B (falling edge to falling edge).

Figure 9.7-3 Long Pulse Width Measurement



9.8 Status of the Pulse Width Count Timer in Each Mode

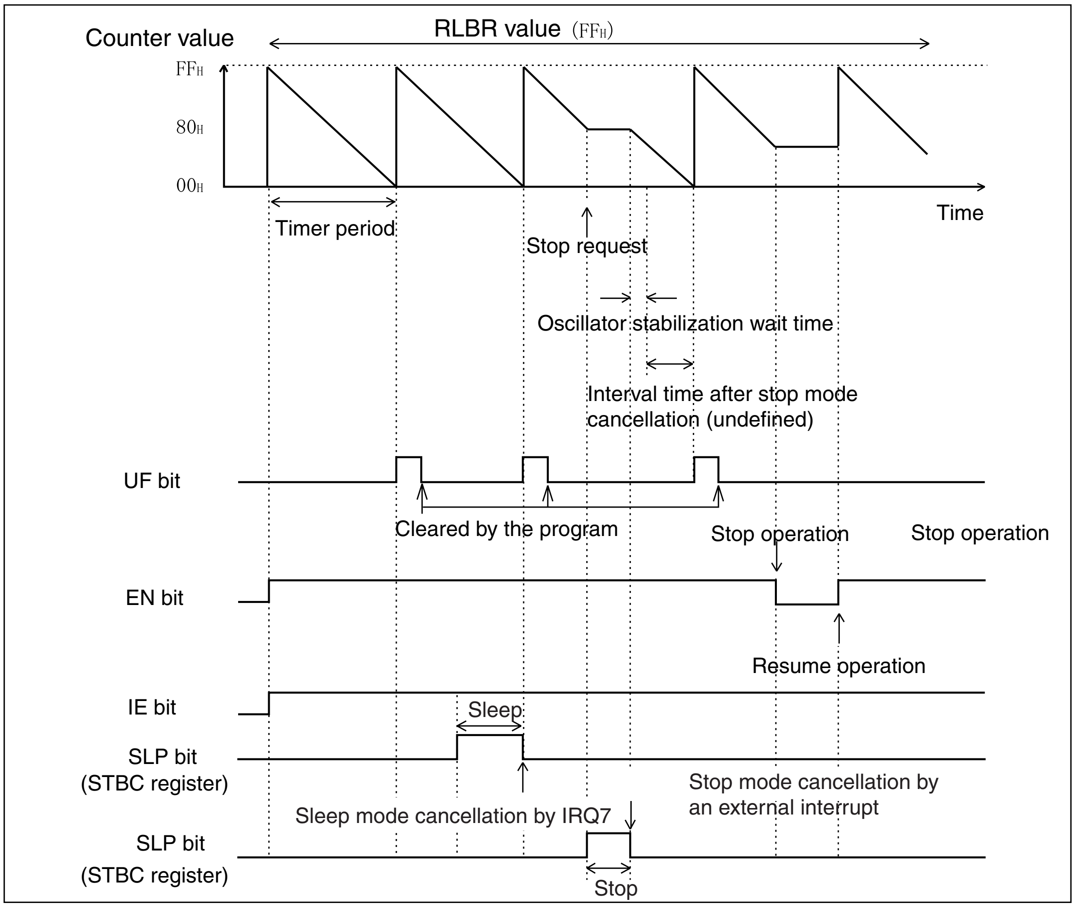
This section describes the operations performed by the pulse width count timer when it enters sleep mode or stop mode, or receives a stop request.

■ Operation in Standby Mode and for a Stop Request

Figure 9.8-1 "Counter Operation in Standby Mode and for a Stop Request" shows how the counter value changes when the counter enters sleep mode or stop mode, or receives a stop request while it is executing the interval timer function or pulse width measurement function.

Upon entering stop mode, the counter halts operation and retains its value. When stop mode is canceled by an external interrupt, the counter resumes operation from the retained value. For this reason, the initial interval time and pulse width values are invalid. After canceling stop mode, initialize the pulse width count timer again.

Figure 9.8-1 Counter Operation in Standby Mode and for a Stop Request



9.9 Notes on Using the Pulse Width Count Timer

This section provides notes on using the pulse width count timer.

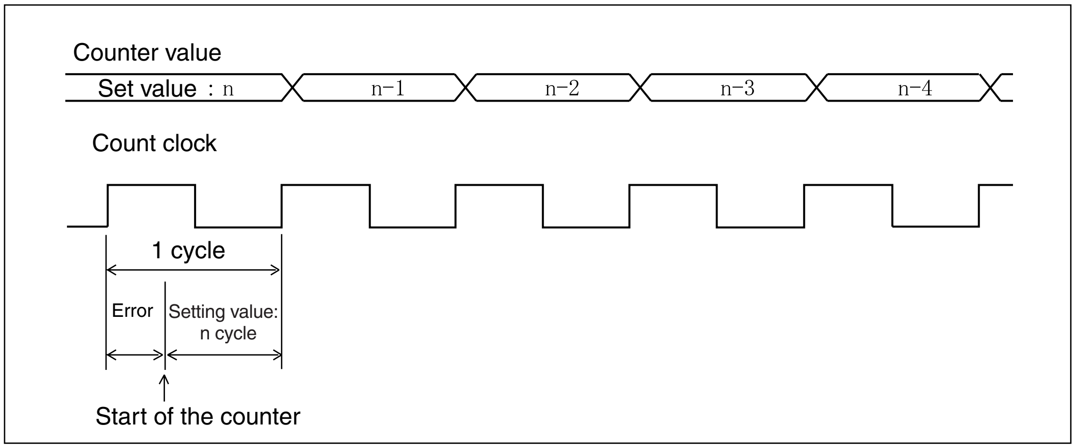
■ Notes on Using the Pulse Width Count Timer

❍ Error

When the interval timer function is used, the start of the counter by the program is not synchronized with the start of countdown by using the selected internal count clock. For this reason, the time from starting the counter until the counter value changes from 00_H to FF_H may be as much as one count clock cycle shorter.

Figure 9.9-1 "Error Occurring before the Operation of the Counter Starts" shows how an error occurs before the operation of the counter starts.

Figure 9.9-1 Error Occurring before the Operation of the Counter Starts



❍ Notes on setting the timer using a program

- Do not change the data of PWC pulse width control register 2 (PCR2) while the interval timer function or pulse width measurement function is operating (PCR1: EN = 1).
- Before switching from the interval timer function to the pulse width measurement function or vice versa (PCR2: FC), set the value of the PCR1 register in such a way that the counter (PCR1: EN = 0), disable interrupts (PCR1: IE = 0), and clear interrupt requests (UF, IR, and BF = 0) are disabled.
- The counter cannot recover from an interrupt if the interrupt request flag bits (PCR1: UF, IR, and BF) and the interrupt request enable bit (PCR1: IE = 1) are set to "1." Be sure to clear the interrupt request flag bits.
- In the case of the pulse width measurement function, if the previous measurement value has not been read out when the next pulse width measurement has been completed, the counter retains the previous value without transferring the new value to the PWC reload buffer register (RLBR). When measuring the long pulse width, read the measured value before the next counter value change from 00_H to FF_H occurs.
- The interrupt request flag bits (PCR1: UF, IR, and BF) are not set if a counter stop (PCR1:

EN = 0) and an interrupt occur at the same time.

9.10 Program Examples for the Interval Timer Function of the Pulse Width Count Timer

This section shows two program examples for the interval timer function of the pulse width count timer.

■ Program Example 1 for the Interval Timer Function (Reload timer mode)

❍ Processing specifications

- An interval timer interrupt occurs repeatedly at 3 ms intervals (reload timer mode).
- A square wave is output to the WT0 pin that is initially at the "L" level. This wave is inverted at the end of each interval.
- The RLBR register value shown below causes the interval time to be about 3 ms when the source oscillation of the main clock is 10 MHz. The count clock is 32 t_{inst} (t_{inst}: 1/4 of the source oscillation of the main clock when the gear is set to the maximum speed).

RLBR register value = 3ms/ (32x4/10MHz) = 234.4 (OEAH)

❍ Coding example

```
PCR1    EQU    002CH             ;Address of PWC pulse width control register 1
PCR2    EQU    002DH             ;Address of PWC pulse width control register 2
RLBR    EQU    002EH             ;Address of the PWC reload buffer register

EN      EQU    PCR1:7            ;Definition of the counter operation enable bit
IE      EQU    PCR1:5            ;Definition of the interrupt request enable bit
UF      EQU    PCR1:2            ;Definition of the underflow (00H--> FFH)
                                  interrupt request flag bit
ILR2    EQU    007CH             ;Address of interrupt level setting register 2
INT_V   DSEG   ABS               ;[DATA SEGMENT]
        ORG    0FFECH
IRQ7    DW     WARI              ;Setting of the interrupt vector
INT_V   ENDS
;----------Main program------------------------------------------------------
        CSEG                     ;[CODE SEGMENT]
                                 ;The stack pointer (SP) and other registers are
                                  assumed to have been initialized.
           :
        CLRI                     ;Disable interrupts
        CLRB    EN               ;Stop counter operation
        CLRB    IE               ;Disable the interrupt request output
        MOV     A,RLBR           ;Clear the buffer full flag (PCR1:bit 0).
        MOV     ILR2,#01111111B  ;Setting of the interrupt level (level 1)
        MOV     RLBR,#0EAH       ;Reloaded counter value (interval time)
        MOV     PCR2,#00010000B  ;Interval timer function, reload timer mode,
                                  initial value of the WTO pin output,
                                  and 32tinst selection
        MOV     PCR1,#11100000B  ;Start counter operation, enable WT0 pin output,
                                  enable interrupt request output, clear the
                                  underflow (00H-->FFH) interrupt request flag,
                                  and clear the measurement-end interrupt request
                                  flag (bit 1)
        SETI                     ;Enable interrupts
           :
;-----------Interrupt processing routine-----------------------------------------
WARI    CLRB    UF               ;Clear the underflow (01H-->00H) interrupt
                                  request flag.
        PUSHW   A
        XCHW    A,T
        PUSHW   A
           :
        User processing
           :
        POPW    A
        XCHW    A,T
        POPW    A
        RETI
        ENDS
;----------------------------------------------------------------------------
        END
```

■ Program Example 2 for the Interval Timer Function (One-shot Timer Mode)

❍ Processing specifications

- One 3 ms interval timer interrupt occurs (one-shot timer mode).
- A pulse wave is output to the WT0 pin that is initially at the "H" level. The wave is inverted at the end of each interval.
- The RLBR register value shown below causes the interval time to be about 3 ms when the source oscillation of the main clock is 10 MHz. The count clock is 32 t_{inst} (t_{inst}: 1/4 of the source oscillation of the main clock when the gear is set to the maximum speed).

❍ Coding example

```
PCR1   EQU    002CH             ;Address of PWC pulse width control register 1
PCR2   EQU    002DH             ;Address of PWC pulse width control register 2
RLBR   EQU    002EH             ;Address of the PWC reload buffer register
EN     EQU    PCR1:7            ;Definition of the counter operation enable bit
IE     EQU    PCR1:5            ;Definition of the interrupt request enable bit
UF     EQU    PCR1:2            ;Definition of the underflow (00$_H$-->FF$_H$) interrupt
                                 request flag bit
ILR2   EQU    007CH             ;Address of interrupt level setting register 2
INT_V  DSEG   ABS               ;[DATA SEGMENT]
       ORG    0FFECH
IRQ7   DW     WARI              ;Setting of the interrupt vector
INT_V  ENDS
;----------Main program------------------------------------------------------
       CSEG                     ;[CODE SEGMENT]
                                ;The stack pointer (SP) and other registers are
                                 assumed to have been initialized.
          :
       CLRI                     ;Disable interrupts
       CLRB   EN                ;Stop counter operation
       CLRB   IE                ;Disable interrupt request output
       MOV    A,RLBR            ;Clear the buffer full flag (PCR1: bit 0)
       MOV    ILR2,#01111111B   ;Setting of the interrupt level (level 1)
       MOV    RLBR,#0EAH        ;Reloaded counter value (interval time)
       MOV    PCR2,#01110000B   ;Interval timer function, one-shot timer mode
                                 Initial value of WT0 pin output, 32 t$_{inst}$
                                 selection
       MOV    PCR1,#11100000B   ;Start counter operation, enable WT0 pin output,
                                 enable interrupt request output, clear the
                                 underflow (00$_H$-->FF$_H$) interrupt request flag,
                                 and clear the measurement-end interrupt request
                                 flag (bit 1)
       SETI                     ;Enable interrupts
          :
;---------Interrupt processing routine----------------------------------------
WARI   CLRB   UF                ;Clear the underflow (01$_H$-->00$_H$) interrupt
                                 request flag.
       PUSHW  A
       XCHW   A,T
       PUSHW  A
          :
       User processing
          :
       POPW   A
       XCHW   A,T
       POPW   A
       RETI
       ENDS
;----------------------------------------------------------------------------
       END
```

9.11 Program Example for the Pulse Width Measurement Function of the Pulse Width Count Timer

This section shows a program example for the pulse width measurement function of the pulse width count timer.

■ Program Example for the Pulse Width Measurement Function Program

❍ Processing specifications

- The "H"-width of pulses input to the PWC pin is measured (pulse width measurement function).
- When the width measurement of a pulse is completed, an interrupt occurs, allowing measurement to continue.
- The relationship between the RLBR register value and the measured pulse width shown below is observed when the count clock is 4 t_{inst} (t_{inst}: 1/4 of the source oscillation of the main clock when the gear is set to the maximum speed) and when the source oscillation of the main clock is 10 MHz.

Pulse width = (256 - RLBR register value) x 4 x 4/10 MHz (measurement range: 1.6 μs to 409.6 μs)

❍ Coding example

```
DDR3   EQU    000FH
PCR1   EQU    002CH            ;Address of PWC pulse width control register 1
PCR2   EQU    002DH            ;Address of PWC pulse width control register 2
RLBR   EQU    002EH            ;Address of the PWC reload buffer register
EN     EQU    PCR1:7           ;Definition of the counter operation enable bit
IE     EQU    PCR1:5           ;Definition of the interrupt request enable bit
IR     EQU    PCR1:1           ;Measurement-end interrupt request flag bit
BF     EQU    PCR1:0           ;Buffer full flag bit
ILR2   EQU    007CH            ;Address of interrupt level setting register 2
INT_V  DSEG   ABS              ;[DATA SEGMENT]
       ORG    0FFE6H
IRQ7   DW     WARI             ;Setting of the interrupt vector
INT_V  ENDS
;----------Main program---------------------------------------------------------
       CSEG                    ;[CODE SEGMENT]
                               ;The stack pointer (SP) and other
                                registers are assumed to have been
                                initialized.
          :
       MOV    DDR3,#00000000B  ;Set the P35/PWC pin to input
       CLRI                    ;Disable interrupts
       CLRB   EN               ;Stop counter operation
       CLRB   IE               ;Disable the interrupt request output
       MOV    A,RLBR           ;Clear the buffer full flag (PCR1:BF)
       MOV    ILR2,#01111111B  ;Setting of the interrupt level (level 1)
       MOV    RLBR,#0E9H       ;Counter reload value (interval time)
       MOV    PCR2,#10001000B  ;Pulse width measurement function, 4 tinst
                                selection, and "H" pulse selection
       MOV    PCR1,#10100000B  ;Enable counter operation, disable WT0 pin
                                output, enable interrupt request output,
                                clear the underflow (00H-->FFH) interrupt
                                request flag, and clear the measurement-end
                                interrupt request flag (IR)
       SETI                    ;Enable interrupts
;----------Interrupt processing routine---------------------------------------
WARI   CLRB   IR               ;Clear the measurement-end interrupt request
                                flag.
       PUSHW  A
       XCHW   A,T
       PUSHW  A
       MOV    A,RLBR           ;Read the pulse width measurement value and clear
                                the BF flag
       User processing
       POPW   A
       XCHW   A,T
       POPW   A
       RETI
       ENDS
;-------------------------------------------------------------------------------
       END
```

CHAPTER 10 6-BIT PPG TIMER

This chapter describes the functions and operations of the 6-bit PPG timer.

10.1 Overview of the 6-Bit PPG Timer

The 6-bit PPG timer is a 6-bit binary counter that can select one out four types of internal count clocks. Because it can set an output waveform cycle and "H" width, it can also be used as the remote control transmission frequency generator or as buzzer output.

■ 6-Bit PPG Timer Function

- The waveforms generated by the 6-bit PPG timer are output to the PPG3 pin.
- This timer can set output waveform cycles and "H" widths independently.
- This timer can select a count clock from among four types of internal clocks.
- This timer can generate frequencies ranging from 2 to 2^{6-1} count clock cycles.

Table 10.1-1 "Output Cycles and Ranges for Adjusting the "H" Width" lists the output cycles and the ranges for adjusting the "H" width.

Table 10.1-1 Output Cycles and Ranges for Adjusting the "H" Width

Internal count clock cycle	Output cycle	Output "H" width (*1)
$1t_{inst}$	$2t_{inst}$ to $63t_{inst}$	$1t_{inst}$ to $62t_{inst}$
$2t_{inst}$	$4t_{inst}$ to $126t_{inst}$	$2t_{inst}$ to $124t_{inst}$
$8t_{inst}$	$16t_{inst}$ to $504t_{inst}$	$8t_{inst}$ to $496t_{inst}$
$32t_{inst}$	$64t_{inst}$ to $2016t_{inst}$	$32t_{inst}$ to $1984t_{inst}$

t_{inst}: Instruction cycle (affected by the clock mode and other factors)

[Example]

An example of calculating a cycle and "H" width of the 6-bit PPG function is given below.

Assume that the count clock cycle is set to 1 t_{inst} for main clock oscillation (F_{CH}) 10 MHz. In this case, if the following is set:

Cycle compare value = 011110_B (30 clock cycles)

"H" width compare value = 001010_B (width of 10 clocks)

The "H" width and cycle of the output waveform are obtained as shown below. However, they are obtained only if the fastest clock available in main clock mode (SCS = 1) is selected (CS1, CS01 instruction cycle = $4/F_{CH}$) with the system clock control register (SYCC).

Cycle = Cycle compare value x count clock cycle
= "011110$_B$" (30 clock cycles) x 1 x 4/F_{CH}
= 30 x 0.4 µs = 12 µs

"H" width = "H" width compare value x count clock cycle
= "001010$_B$" (width of 10 clocks) x 1 x 4/F_{CH}
= 10 x 0.4 µs
= 4 µs

The "H" level is output when an "H" width setting value is equal to or greater than a cycle setting value.

■ 6-bit PPG Timer Function 2

The controllable duty width is about 1.60% to 100%. The smaller the cycle compare value, the lower the resolution (the larger the minimum step width of the duty ratio).

The output cycle and duty ratio can be calculated from the following formulas:

Output cycle = cycle compare value x count clock cycle
Duty ratio = (compare value for "H" width/cycle compare value) x 100 (%)

Table 10.1-2 "Resolutions and Output Cycles for the 6-Bit PPG" lists the minimum steps of the duty ratio and output cycles.

Table 10.1-2 Resolutions and Output Cycles for the 6-Bit PPG

Cycle compare value	"H" width compare value setting range	Output cycle				Resolution	Minimum step of duty ratio
		Count clock = 1 t_{inst}	Count clock = 2 t_{inst}	Count clock = 8 t_{inst}	Count clock = 32 t_{inst}		
0	0	Setting impossible				Output "H"	
1	0 to 2						
2	0 to 2	2 t_{inst}	4 t_{inst}	16 t_{inst}	64 t_{inst}	1/2	50.0%
3	0 to 3	3 t_{inst}	6 t_{inst}	24 t_{inst}	96 t_{inst}	1/3	33.3%
4	0 to 4	4 t_{inst}	8 t_{inst}	32 t_{inst}	128 t_{inst}	1/4	25.0%
5	0 to 5	5 t_{inst}	10 t_{inst}	40 t_{inst}	160 t_{inst}	1/5	20.0%
6	0 to 6	6 t_{inst}	12 t_{inst}	48 t_{inst}	192 t_{inst}	1/6	16.7%
7	0 to 7	7 t_{inst}	14 t_{inst}	56 t_{inst}	224 t_{inst}	1/7	14.3%
8	0 to 8	8 t_{inst}	16 t_{inst}	64 t_{inst}	256 t_{inst}	1/8	12.5%
9	0 to 9	9 t_{inst}	18 t_{inst}	72 t_{inst}	288 t_{inst}	1/9	11.1%
10	0 to 10	10 t_{inst}	20 t_{inst}	80 t_{inst}	320 t_{inst}	1/10	10.0%
:							
15	0 to15	15 t_{inst}	30 t_{inst}	120 t_{inst}	480 t_{inst}	1/15	6.7%
:							
20	0 to 20	20 t_{inst}	40 t_{inst}	160 t_{inst}	640 t_{inst}	1/20	5.0%
:							
25	0 to 25	25 t_{inst}	50 t_{inst}	200 t_{inst}	800 t_{inst}	1/25	4.0%
:							
30	0 to 30	30 t_{inst}	60 t_{inst}	240 t_{inst}	960 t_{inst}	1/30	3.3%
:							
40	0 to 40	40 t_{inst}	80 t_{inst}	320 t_{inst}	1280 t_{inst}	1/40	2.5%
:							
50	0 to 50	50 t_{inst}	100 t_{inst}	400 t_{inst}	1600 t_{inst}	1/50	2.0%
:							
60	0 to 60	60 t_{inst}	120 t_{inst}	480 t_{inst}	1920 t_{inst}	1/60	1.7%
:							
63	0 to 63	63 t_{inst}	126 t_{inst}	504 t_{inst}	2016 t_{inst}	1/63	1.6%

t_{inst}: Instruction cycle

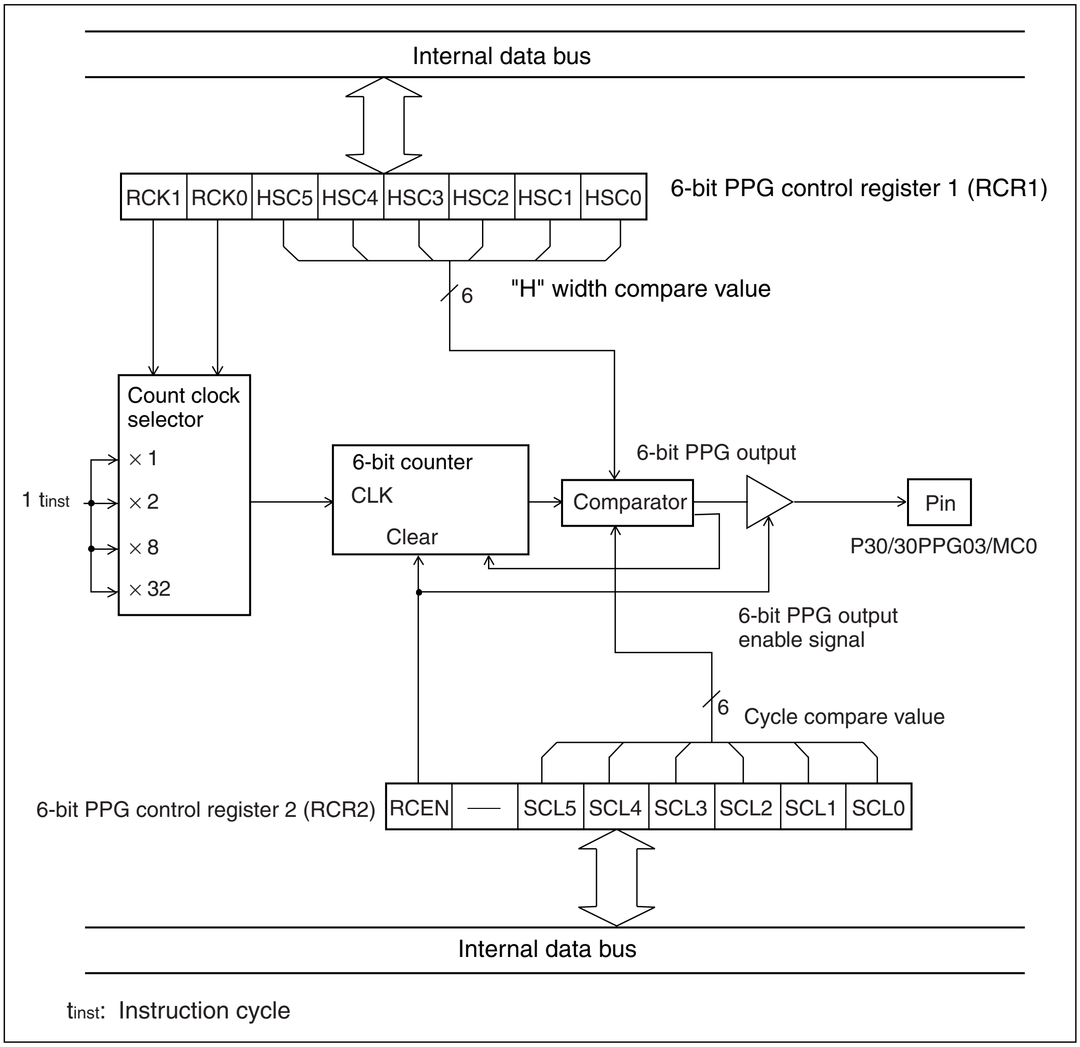
10.2 Configuration of the 6-Bit PPG Timer Circuit

The 6-bit PPG timer consists of the following five blocks:

- **Count clock selector**
- **6-bit counter**
- **Comparator**
- **6-bit PPG control register 1 (RCR1)**
- **6-bit PPG control register 2 (RCR2)**

■ Block Diagram of the 6-Bit PPG Timer

Figure 10.2-1 Block Diagram of the 6-Bit PPG Timer



❍ **Count clock selector**

The count clock selector selects a 6-bit counter count-up clock from among four types of internal count clocks.

❍ **6-bit counter**

The 6-bit counter counts up using the count clock selected by the count clock selector.

This counter is cleared (RCR2:RCEN = 0) with the output enable bit of the RCR2 register.

❍ **Comparator**

The comparator retains the output at "H" until the value of the 6-bit counter matches the value of the register for "H" width comparison.

Thereafter, the comparator keeps the output at "L" until the counter value matches the cycle compare register value. The 6-bit counter is then cleared and counting continues starting at 00_H.

❍ **6-bit PPG control register 1 (RCR1)**

This register selects a count clock of the 6-bit PPG timer and sets a compare value for the "H" width.

❍ **6-bit PPG control register 2 (RCR2)**

This register enables output of the 6-bit PPG timer and sets a cycle compare value.

10.3 Pins of the 6-Bit PPG Timer

This section describes 6-bit PPG timer pins and provides a block diagram of the pins.

■ Pins of the 6-Bit PPG Timer

The 6-bit PPG timer pin is P30/PPG03/MCO.

❍ P30/PPG03/MCO pin

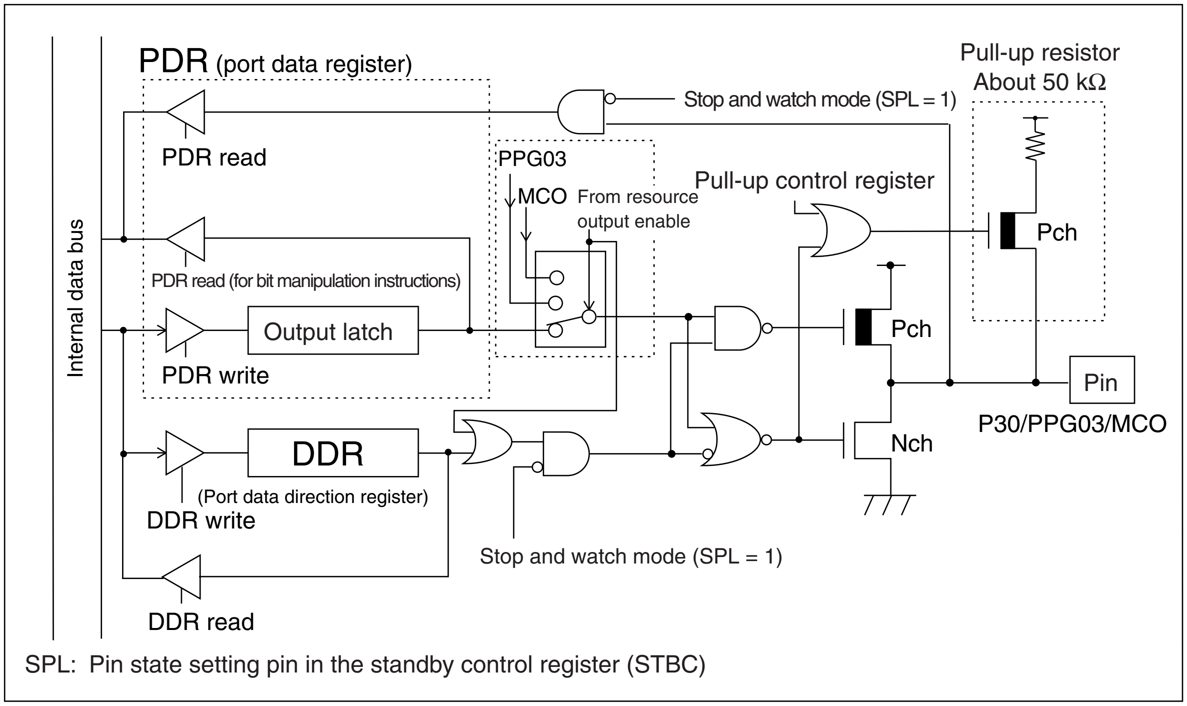
This pin functions as a general-purpose I/O port (P30), 6-bit PPG timer output (PPG03), or main clock output (MCO).

PPG03:

The cycle that was set and the "H" width PPG waveform are output to this pin. The PPG waveform can be output by setting the output enable bit (RCR2:RCEN) of the 6-bit PPG control register to "1."

■ Block Diagram of the 6-Bit PPG Timer Pins

Figure 10.3-1 Block Diagram of the P23/PPG1 Pins



SPL: Pin state setting pin in the standby control register (STBC)

Reference:

If "pull-up resistor available" is selected with the port 3 pull-up resistor control register, the pins are set to the "H" level (pull-up state) in stop/watch mode (SPL = 1). However, the pull-up is disabled during a reset and the pins enter the high-impedance state.

10.4 Registers of the 6-Bit PPG Timer

This section describes the 6-bit PPG timer registers.

■ Registers of the 6-Bit PPG Timer

Figure 10.4-1 6-Bit PPG Timer Registers

RCR1 (6-bit PPG control register 1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 5 8H	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

RCR2 (6-bit PPG control register 2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 5 9H	RCEN	—	SCL5	SCL4	SCL3	SCL2	SCL1	SCL0	0-000000B
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

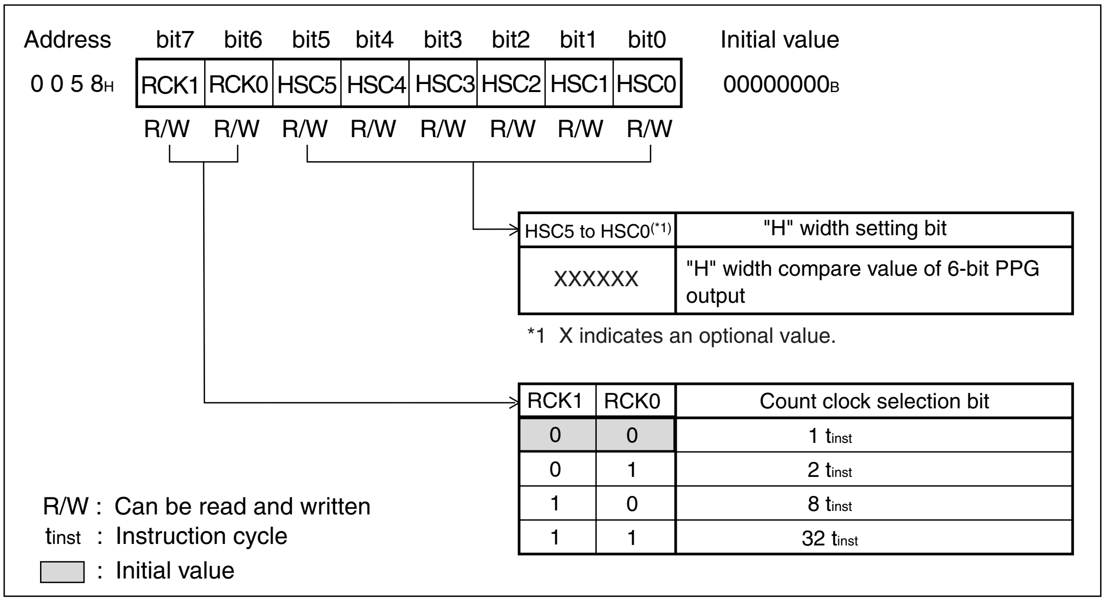
R/W : Can be read and written
— : Undefined

10.4.1 6-Bit PPG Control Register 1 (RCR1)

The 6-bit PPG control register 1 is used to select a 6-bit PPG timer count clock and set the "H" width.

■ 6-Bit PPG Control Register 1 (RCR1)

Figure 10.4-2 6-Bit PPG Control Register 1 (RCR1)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 5 8H	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

HSC5 to HSC0(*1)	"H" width setting bit
XXXXXX	"H" width compare value of 6-bit PPG output

*1 X indicates an optional value.

RCK1	RCK0	Count clock selection bit
0	0	1 t_{inst}
0	1	2 t_{inst}
1	0	8 t_{inst}
1	1	32 t_{inst}

R/W : Can be read and written
t_{inst} : Instruction cycle
▨ : Initial value

Table 10.4-1 Functions of the 6-Bit PPG Control Register 1 (RCR1) Bits

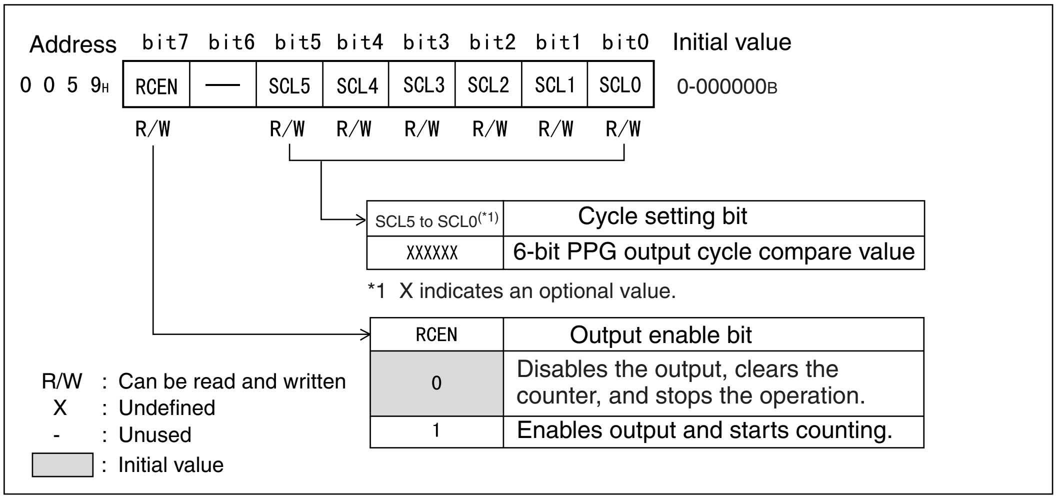
Bit		Function
bit7 bit6	RCK2, RCK1: Count clock selection bits	• These bits select a count clock for the 6-bit PPG timer from among the four types of internal count clocks.
bit5 bit4 bit3 bit2 bit1 bit0	HSC5 to HSC0: "H" width setting bits	• These bits are used to set the count of the "H" width of the 6-bit PPG timer output ("H" width compare value). The count is compared with a counter value. **Note:** Set a value from 01_H to $3E_H$ and be sure to make it smaller than the cycle setting value. If a setting value is equal to or greater than the cycle setting value, the "H" level is always output.

10.4.2 6-Bit PPG Control Register 2 (RCR2)

The 6-bit PPG control register 2 is used to enable the output of 6-bit PPG waveforms and to set a cycle.

■ 6-Bit PPG Control Register 2 (RCR2)

Figure 10.4-3 6-Bit PPG Control Register 2 (RCR2)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 5 9H	RCEN	—	SCL5	SCL4	SCL3	SCL2	SCL1	SCL0	0-000000B
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

SCL5 to SCL0(*1)	Cycle setting bit
XXXXXX	6-bit PPG output cycle compare value

*1 X indicates an optional value.

RCEN	Output enable bit
0	Disables the output, clears the counter, and stops the operation.
1	Enables output and starts counting.

R/W : Can be read and written
X : Undefined
- : Unused
▨ : Initial value

Table 10.4-2 Functions of the 6-Bit PPG Control Register 2 (RCR2) Bits

Bit		Function
bit7	RCEN: Output enable bit	• When this bit is "0," the P30/PPG03/MC0 pin is set as a general-purpose port (P30). When this bit is "1," this pin is set as a 6-bit PPG output pin (PPG03). • Setting this bit to "0" by writing stops and clears the counter. Writing "1" starts the counter operation.
bit6	Unused	• In read operations, the value of this bit is undefined. • Writing this bit has no effect on operation.
bit5 bit4 bit3 bit2 bit1 bit0	SCL5 to SCL0: Cycle setting bits	• These bits set the count for the cycle of the 6-bit PPG output waveform (cycle compare value). This value is compared with the counter value. **Note:** Set a value from 02_H to $3F_H$. If 00_H is set, the state of the preceding pin is retained until the H width compare value is reached. After an H compare match, the H level is always output.

10.5 Operation of the 6-Bit PPG Timer

The 6-bit PPG timer can set cycles and "H" widths to generate a remote control transmission frequency.

■ Operation of the 6-Bit PPG Timer

The settings shown in Figure 10.5-1 "6-Bit PPG Timer Settings" are required for the 6-bit PPG timer to operate.

Figure 10.5-1 6-Bit PPG Timer Settings

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RCR1	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0
	◎	◎	◎	◎	◎	◎	◎	◎

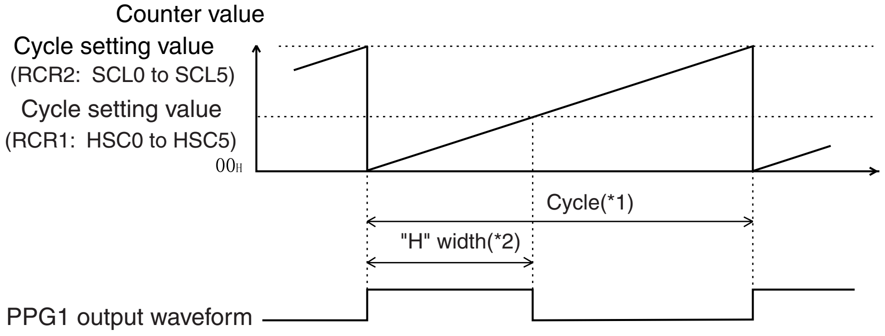
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RCR2	RCEN	—	SCL5	SCL4	SCL3	SCL2	SCL1	SCL0
	1		◎	◎	◎	◎	◎	◎

◎	: Bit used
1	: Bit with 1
0	: Bit with 0
-	: Not used

If output from the 6-bit PPG is enabled, the 6-bit counter starts counting from 00_H in synchronization with the selected count clock. The PPG1 pin is kept at the "H" level until the counter value becomes an "H" width compare value. Next, the PPG1 pin is kept at the "L" level until the counter value becomes the cycle compare value. When a match occurs, the 6-bit counter is cleared and counting starts again from 00_H. The 6-bit PPG timer can be used as a 6-bit PPG because the "H" width and cycle can be set independently.

Figure 10.5-2 "Operation of the 6-Bit PPG Timer" shows 6-bit PPG timer operation.

Figure 10.5-2 Operation of the 6-Bit PPG Timer



*1 Cycle = count clock cycle x count compare value

*2 Cycle = count clock cycle x H width compare value

10.6 Notes on Using the 6-Bit PPG Timer

This section provides notes on using the 6-bit PPG timer.

■ Notes on Using the 6-Bit PPG Timer

❍ Limiting the "H" width setting value

Always set the "H" width setting bit (RCR1:HSC5 to HSC0) value of 6-bit PPG control register 1 in the range from "000010" to "111111" (02 to $3E_H$). If 00_H is set, the PPG1 pin output becomes "H" level output for 0.5 t_{inst}. Always set an "H" width setting value smaller than value of the cycle setting bits (RCR2:SCL5 to SCL0) of 6-bit PPG control register 2. If the former value is equal to or greater than the latter, the PPG1 pin output is always set to the "H" level.

❍ Resolution

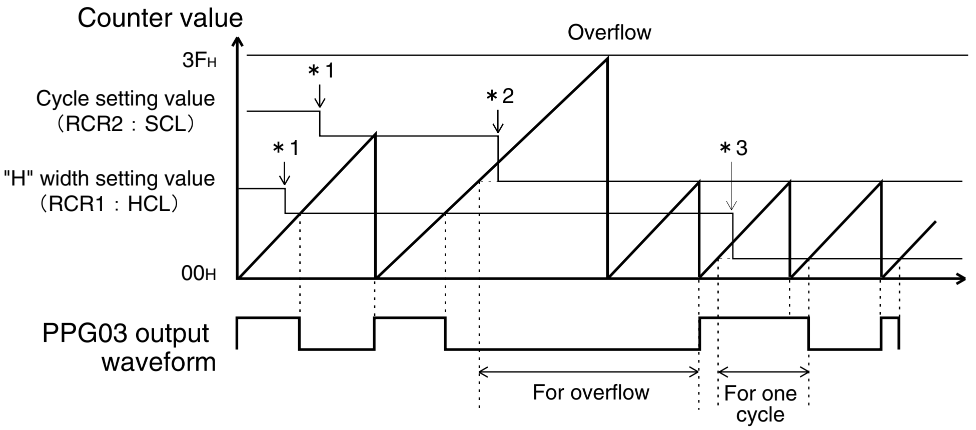
The maximum resolution of the "H" width is 1/63 of the cycle (when the cycle setting value is $3F_H$). If however, a small value is set for this cycle setting value, the minimum resolution of the "H" width is restricted to 1/2 of the cycle (when the cycle setting value is 02_H).

❍ Changing setting values during operation

The following are directly compared with each other: the 6-bit PPG waveform frequency generation 6-bit counter and the "H" width setting bits (RCR1:HSC5 to HSC0), and the counter and cycle setting bits (RCR2:SCL5 to SCL0). Therefore, if a setting value is made small during counter operation, the counter overflows and the cycle may lengthen until matching is detected again. In addition, the "H" width may lengthen until matching is detected in the next cycle.

Figure 10.6-1 "Setting Value Changes during 6-Bit PPG Timer Operation" shows the changes to setting values during 6-bit PPG timer operation.

Figure 10.6-1 Setting Value Changes during 6-Bit PPG Timer Operation



*1 The value of the operating counter is smaller than the changed setting value, so the setting value is valid within the cycle.

*2 Because a cycle smaller than the value of the operating counter is set, no matching is detected and the counter overflows.

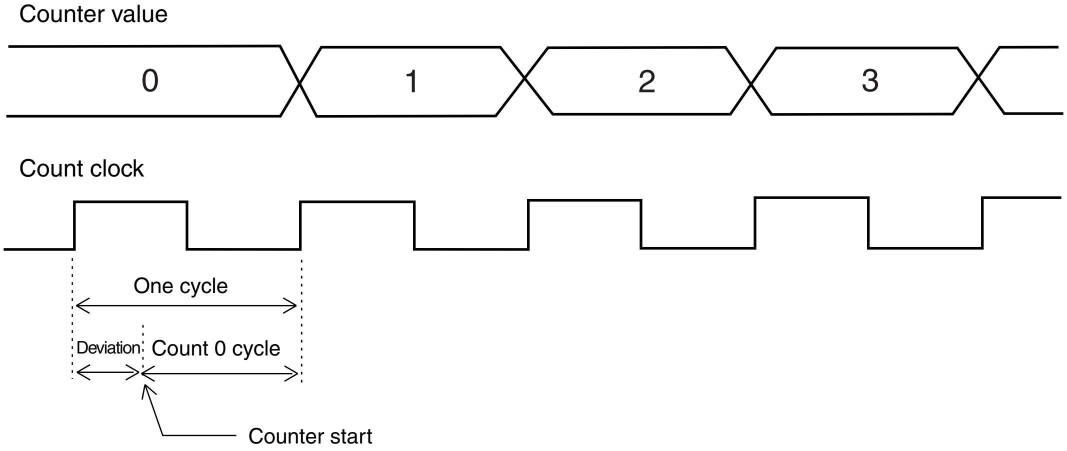
*3 Because an "H" width smaller than the value of the operating counter is set, no matching with the next cycle is detected.

❍ Deviation

The counter start by the program is asynchronous to the start of incrementing with the selected count clock. For this reason, the deviation that exists until a match of the counter value, "H"-width compare value, and cycle compare value is detected may become shorter by up to one count clock cycle.

Figure 10.6-2 "Deviation That Remains till the Start of Count Operation" shows the deviation that exists until the start of count operation.

Figure 10.6-2 Deviation That Remains till the Start of Count Operation



10.7 Program Example of the 6-Bit PPG Timer Programs

This section provides program examples of the 6-bit PPG timer.

■ Program Example of the 6-Bit PPG Timer

❍ Processing specifications

- A remote control transmission frequency is generated with a cycle of about 12 µs and a duty ratio of about 33%.
- Shown below is the cycle compare value for which the cycle becomes 12 µs at the highest main clock speed for a main clock oscillation of 10 MHz (F_{CH}). The count clock is assumed to be 1 t_{inst}. (Time for processing a single instruction: $4/F_{CH}$).

Cycle compare value (RCR2:SCL5 to SCL0) = 12µ s/ (1 x 4/10MHz) = 30

- The "H" width compare value whose duty ratio is approx. 33% is shown below. In this case, the "H" width is approx. 9.5 µs.

"H" width compare value (RCR1:HSC5 to HSC0) = 33/100 x Cycle compare value
= 0.33 x 30
= 10

❍ Coding example (comply with Softune V1)

```
RCR1   EQU    0058H           ;Address of 6-bit PPG control register 1
RCR2   EQU    0059H           ;Address of 6-bit PPG control register 2
;---------Main program------------------------------------------------------------
       CSEG                   ; [CODE SEGMENT]
        :
       MOV    RCR1,#00001010B ;Selects 1 tinst for the count clock; "H" width
                              ;compare value
       MOV    RCR2,#10011110B ;Output is enabled and operation starts,
                              ;cycle compare value
        :
       ENDS
;--------------------------------------------------------------------------------
       END
```

CHAPTER 11 12-BIT PPG TIMER

This chapter describes the functions and operations of the 12-bit PPG timer.

11.1 Overview of the 12-Bit PPG Timer

The 12-bit PPG timer is a 12-bit binary counter that can select one of four clocks as its count clock. This timer can set an output waveform cycle and an "H" width and can be used as a remote control transmission frequency generator or a 12-bit PPG.

■ 12-Bit PPG Timer Function

- This timer generates remote control frequencies and outputs signals to PPG01 or 02.
- This timer can set output waveform cycles and "H" widths independently.
- This timer can select a count clock from among four types of internal clocks.
- Frequencies ranging from 2 to 2^{12-1} count clock cycles can be generated.

Table 11.1-1 "Output Cycles and Ranges for Adjusting the "H" Width" lists the output cycles and the ranges for adjusting the "H" width.

Table 11.1-1 Output Cycles and Ranges for Adjusting the "H" Width

Internal count clock cycle	Output cycle	Output "H" width
$2t_{inst}$	$2t_{inst}$ to $8190t_{inst}$	$1t_{inst}$ to $8188t_{inst}$
$4t_{inst}$	$8t_{inst}$ to $16380t_{inst}$	$2t_{inst}$ to $16376t_{inst}$
$16t_{inst}$	$32t_{inst}$ to $65520t_{inst}$	$16t_{inst}$ to $65504t_{inst}$
$256t_{inst}$	$512t_{inst}$ to $1048.32kt_{inst}$	$128t_{inst}$ to $1048.604kt_{inst}$

t_{inst}: Instruction cycle (affected by the clock mode and others)

Note:

A stable "L" or "H" level can also be output (duty cycle of 0% or 100%).
See the example of calculating the "H" width, shown below.

Reference:

Suppose that 12.5 MHz is selected as the main clock oscillation (F_{CH}) and that a $2t_{inst}$ clock is selected as the count clock cycle. Also suppose that, in main clock mode, the highest clock speed (SYCC:SCS = CS1 = CS0 = 1) is selected from the system clock control register (this causes the instruction cycle time to become $4/F_{CH}$).

[Example]

Cycle = Cycle compare value + count clock cycle
= "011110$_B$" (30-clock cycle) x 2 x 4/F_{CH}
= 30 x 2 x 0.32 µs
= 19.2 µs

"H" width = "H" width compare value x count clock cycle
= "001010$_B$" (10-clock width) x 2 x 4/F_{CH}
= 10 x 2 x 0.32 µs
= 6.4 µs

The "H" level is output when an "H" width setting value is equal to or greater than a cycle setting value.

■ 12-Bit PPG Function

The cycle and "H" width of an output waveform can be set independently, so the 12-bit PPG timer can be used as the remote control transmission output generator. However, the effective range of the "H" width compare setting is a value from "0" (duty ratio of 0%) to the cycle compare setting (duty ratio of 100%). This means that when the cycle compare setting falls (if the cycle of an output waveform is short), the resolution also falls (the step size of the maximum duty ratio becomes large).

he controllable duty width is about 0.02% to 100%. The smaller the cycle compare value, the lower the resolution (the larger the minimum step width of the duty ratio).

The output cycle and duty ratio can be calculated from the following formulas:

Output cycle = cycle compare value x count clock cycle
Duty ratio = "H" width compare value/compare value x 100 (%)

Table 11.1-2 "Resolutions and Output Cycles for the 12-Bit PPG" lists the minimum steps of the duty ratio and output cycles.

Table 11.1-2 Resolutions and Output Cycles for the 12-Bit PPG

Cycle compare value	Output cycle				"H" width compare value setting range	Resolution	Minimum step of duty ratio
	Count clock = 2 t_{inst}	Count clock = 4 t_{inst}	Count clock = 16 t_{inst}	Count clock = 256 t_{inst}			
0	-	-	-	-	-		
1	-	-	-	-	0 to 1		
2	4 t_{inst}	8 t_{inst}	32 t_{inst}	512 t_{inst}	0 to 2	1/2	50.0%
3	6 t_{inst}	12 t_{inst}	48 t_{inst}	768 t_{inst}	0 to 3	1/3	33.3%
4	8 t_{inst}	16 t_{inst}	64 t_{inst}	1024 t_{inst}	0 to 4	1/4	25.0%
5	10 t_{inst}	20 t_{inst}	80 t_{inst}	1280 t_{inst}	0 to 5	1/5	20.0%
6	12 t_{inst}	24 t_{inst}	96 t_{inst}	1536 t_{inst}	0 to 6	1/6	16.7%
7	14 t_{inst}	28 t_{inst}	112 t_{inst}	1792 t_{inst}	0 to 7	1/7	14.3%
8	16 t_{inst}	32 t_{inst}	128 t_{inst}	2048 t_{inst}	0 to 8	1/8	12.5%
9	18 t_{inst}	36 t_{inst}	144 t_{inst}	2304 t_{inst}	0 to 9	1/9	11.1%
10	20 t_{inst}	40 t_{inst}	160 t_{inst}	2560 t_{inst}	0 to 10	1/10	10.0%
:							
20	40 t_{inst}	80 t_{inst}	320 t_{inst}	5120 t_{inst}	0 to 20	1/20	5.0%
:							
100	200 t_{inst}	400 t_{inst}	1600 t_{inst}	25600 t_{inst}	0 to100	1/100	1.0%
:							
500	1000 t_{inst}	2000 t_{inst}	8000 t_{inst}	12800 t_{inst}	0 to 500	1/500	0.2%
:							
1000	2000 t_{inst}	4000 t_{inst}	16000 t_{inst}	256000 t_{inst}	0 to 1000	1/1000	0.1%
:							
2000	4000 t_{inst}	8000 t_{inst}	32000 t_{inst}	512000 t_{inst}	0 to 2000	1/2000	0.05%
:							
3000	6000 t_{inst}	12000 t_{inst}	48000 t_{inst}	768000 t_{inst}	0 to 3000	1/3000	0.03%
:							
4095	8190 t_{inst}	16380 t_{inst}	65520 t_{inst}	1048320 t_{inst}	0 to 4095	1/4095	0.02%

t_{inst}: Instruction cycle

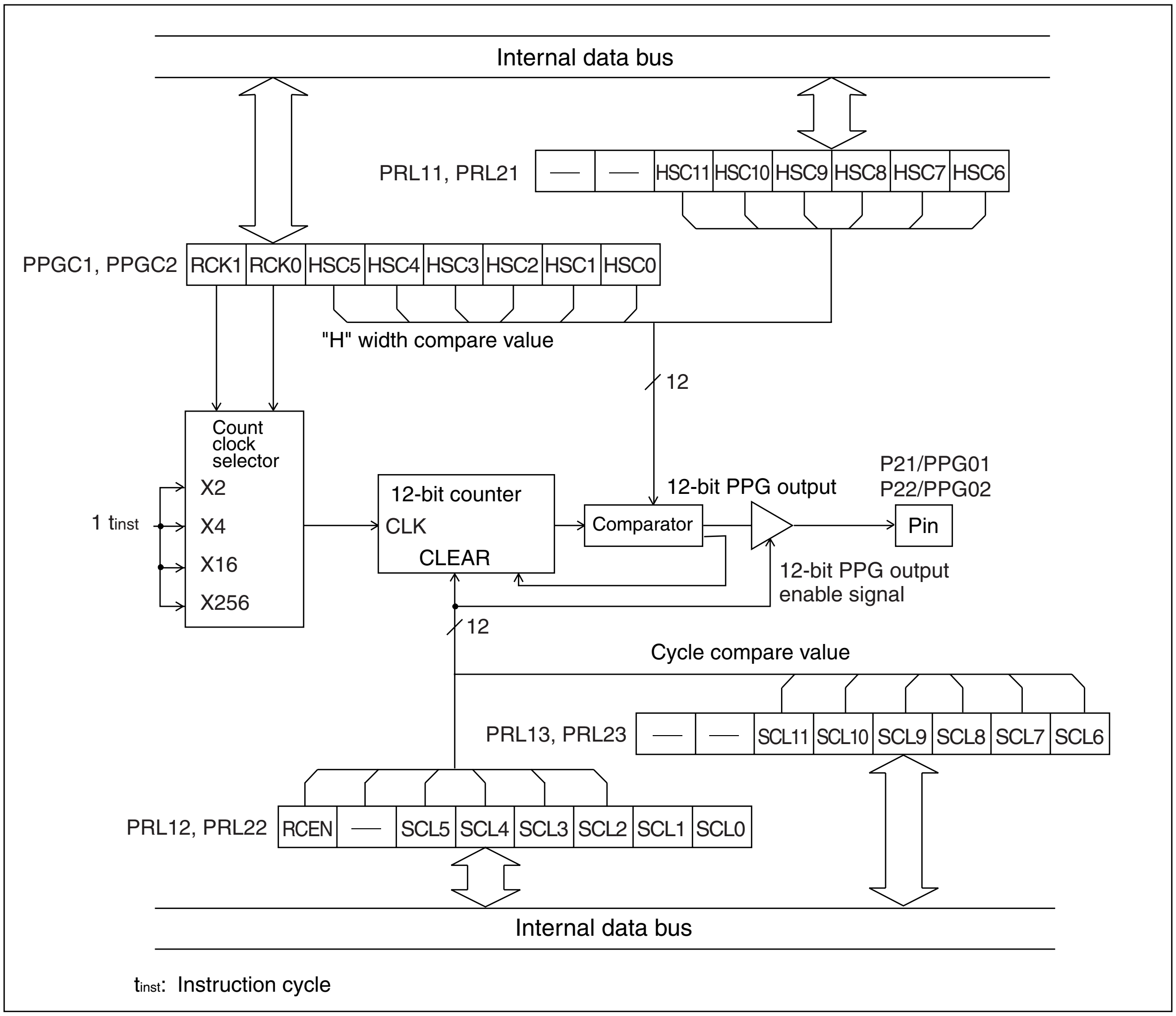
11.2 Configuration of the 12-Bit PPG Timer Circuit

The 12-bit PPG timer consists of the following seven blocks:

- **Count clock selector**
- **12-bit counter**
- **Comparator**
- **PPG1/PPG2 control register (PPGC1/PPGC2)**
- **PPG1/PPG2 reload register 1 (PRL11/PRL21)**
- **PPG1/PPG2 reload register 2 (PRL12/PRL22)**
- **PPG1/PPG2 reload register 3 (PRL13/PRL23)**

■ Block Diagram of the 12-Bit PPG Timer

Figure 11.2-1 Block Diagram of the 12-Bit PPG Timer



❍ Count clock selector

The count clock selector is a circuit that selects four types of internal count clocks and selects a count-up clock for the 12-bit counter.

❍ 12-bit counter

The 12-bit counter counts up using the count clock selected by the count clock selector.

This counter is cleared (RCR23:RCEN = 0) with the output enable bit of the RCR23 register.

❍ Comparator

The comparator keeps the output at "H" until the 12-bit counter value matches the "H" width compare register value.

Thereafter, the comparator keeps the output at "L" until the counter value matches the cycle compare register value that was set. The 12-bit counter is then cleared and counting continues starting at 00_H.

❍ PPG control register (PPGC1/PPGC2), reload register 1 (PRL11/PRL21)

PPGC1/PPGC2 and PRL11/PRL21 are used to select the counter clocks of the 12-bit PPG timer and to set compare values for the output "H" pulse width.

❍ PPG reload register 2 (PRL12/PRL22) and PPG reload register 3 (PRL13/PRL23)

PRL12/PRL22 and PRL23 are used to enable or disable the 12-bit PPG timer and to set compare values for the output cycles.

11.3 Pins of the 12-Bit PPG Timer

This section describes the 12-bit PPG timer pins and provides a block diagram of the pins.

■ 12-Bit PPG Timer Pins

The 12-bit PPG timer pin is either the P21/PPG01 or P22/PPG02 pin.

❍ P21/PPG01 pin and P22/PPG02 pin

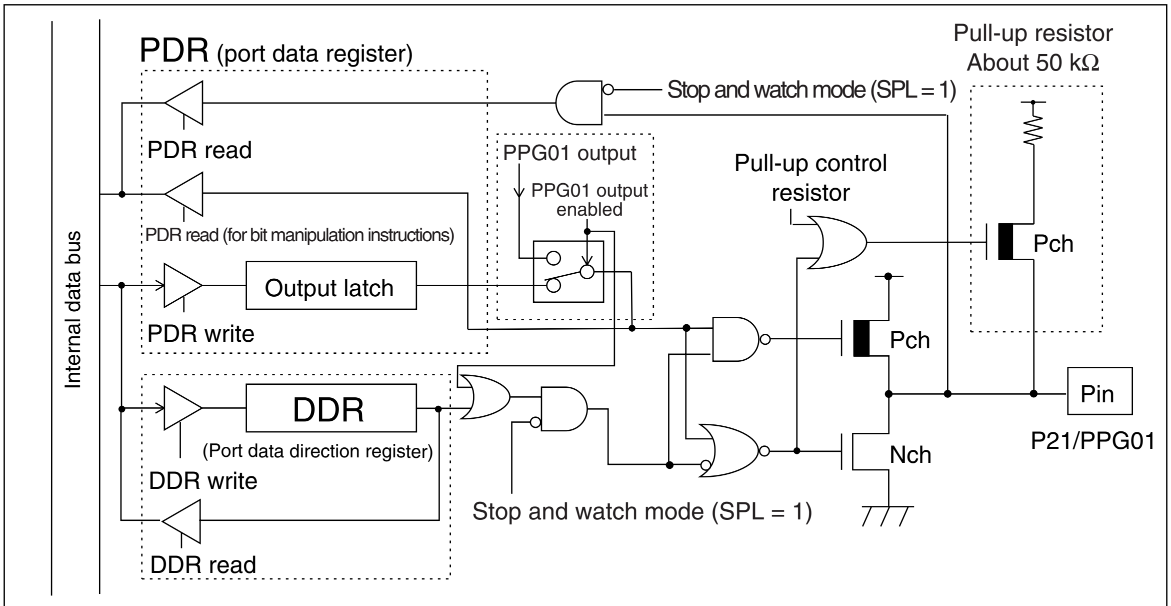
These pins function as the general-purpose CMOS I/O port (P21, P22) or the 12-bit PPG timer output (PPG01, PPG02).

PPG01/PPG02:

The set cycle and the PPG waveform of the "H" width are output to this pin. Setting the output enable bit (RCEN bit of PRL12 or PRL22 = 1) of the 12-bit PPG control register to "1" enables the output of PPG waveforms.

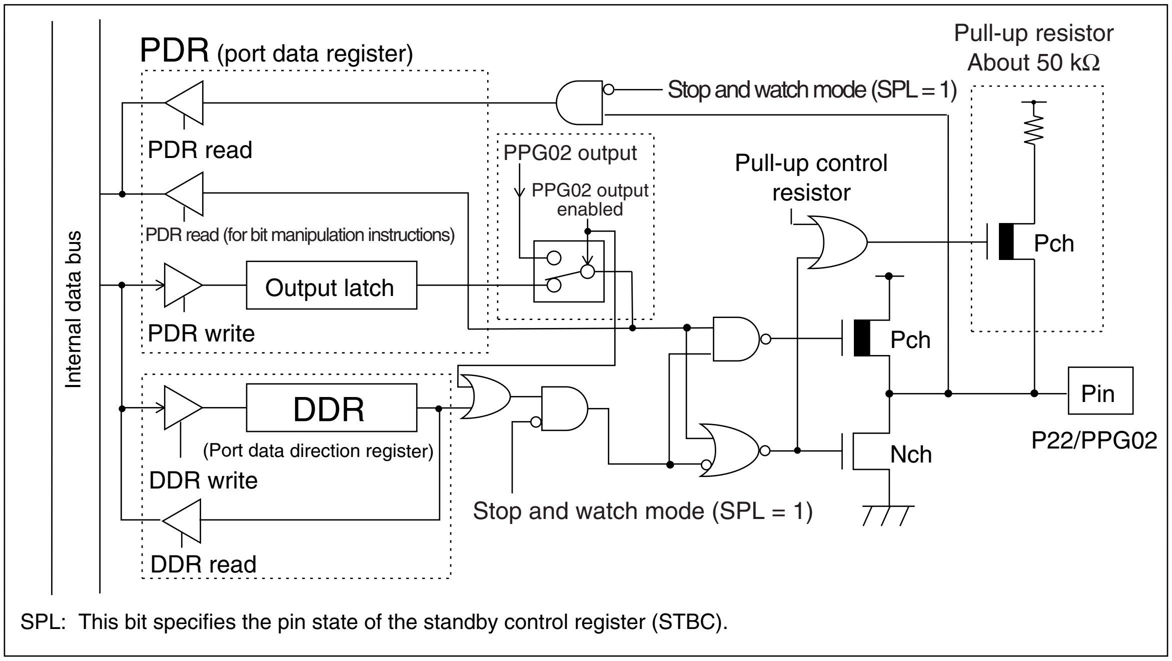
■ Block Diagram of the 12-Bit PPG Timer Pins

Figure 11.3-1 Block Diagram of the P21/PPG01 Pins



SPL: This bit specifies the pin state of the standby control register (STBC).

Figure 11.3-2 Block Diagram of P22/PPG02 Pin


Internal data bus
PDR (port data register)
PDR read
PDR read (for bit manipulation instructions)
Output latch
PDR write
DDR
(Port data direction register)
DDR write
DDR read
PPG02 output
PPG02 output enabled
Stop and watch mode (SPL = 1)
Pull-up control resistor
Pull-up resistor About 50 kΩ
Pch
Pch
Nch
Pin
P22/PPG02
Stop and watch mode (SPL = 1)
SPL: This bit specifies the pin state of the standby control register (STBC).

Reference:

If "pull-up resistor available" is selected in the port 2 pull-up resistor control register, the pins are set to the "H" level (pull-up state) in stop/watch mode (SPL = 1). However, the pull-up is disabled during a reset and the pins enter the high-impedance state.

Note:

If an RMW instruction for the port 2 data register (PDR2) is executed when the 12-bit PPG timer is operating, only the level of the P22/PPG02 pin is read during a read operation.

For this reason, the value of bit 2 in PDR2 may change.

11.4 Registers of the 12-Bit PPG Timer

This section describes the 12-bit PPG timer registers.

■ 12-Bit PPG Timer Registers

Figure 11.4-1 12-bit PPG Time Registers

PPG2 register

PPGC2 (PPG2 control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0038H	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PRL21 (PPG2 reload register 1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
003AH	–	–	HSC11	HSC10	HSC9	HSC8	HSC7	HSC6	--000000B
			R/W	R/W	R/W	R/W	R/W	R/W	

PRL22 (PPG2 reload register 2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0039H	RCEN	–	SCL5	SCL4	SCL3	SCL2	SCL1	SCL0	0-000000B
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

PRL23 (PPG2 reload register 3)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
003BH	–	–	SCL11	SCL10	SCL9	SCL8	SCL7	SCL6	--000000B
			R/W	R/W	R/W	R/W	R/W	R/W	

PPG1 register

PPGC1 (PPG1 control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
004CH	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PRL11 (PPG1 reload register 1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
004EH	–	–	HSC11	HSC10	HSC9	HSC8	HSC7	HSC6	--000000B
			R/W	R/W	R/W	R/W	R/W	R/W	

PRL12 (PPG1 reload register 2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
004DH	RCEN	–	SCL5	SCL4	SCL3	SCL	SCL1	SCL0	0-000000B
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

PRL13 (PPG1 reload register 3)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
004FH	–	–	SCL11	SCL10	SCL9	SCL8	SCL7	SCL6	--000000B
			R/W	R/W	R/W	R/W	R/W	R/W	

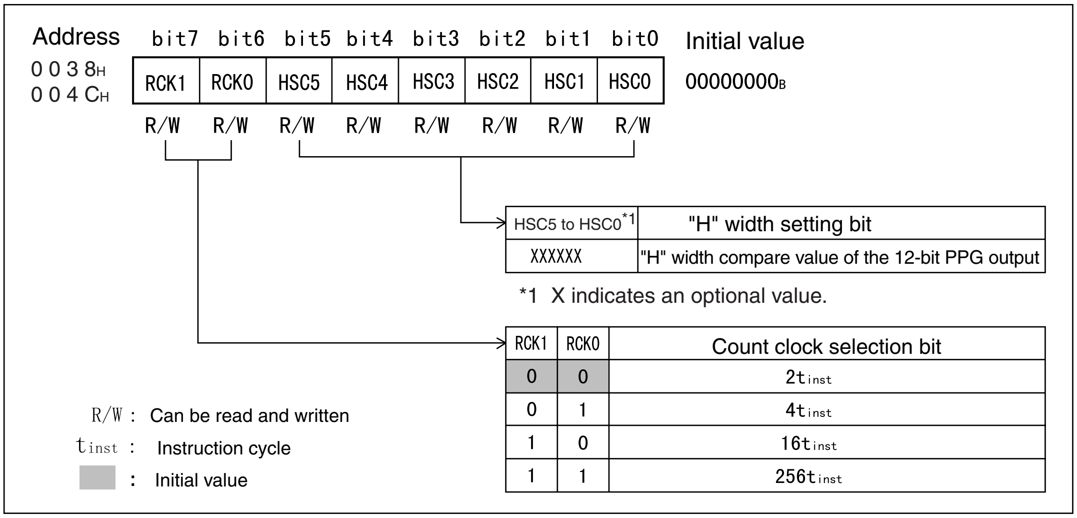
R/W : Can be read and written
- : Unused

11.4.1 12-Bit PPG Control Register 1 (PPGC1/PPGC2)

The 12-bit PPG control register 1 is used to select a 12-bit PPG timer count clock and set the "H" width.

■ 12-Bit PPG Control Register (PPGC1/PPGC2)

Figure 11.4-2 12-bit PPG Control Register 1 (PPGC1/PPGC2)


Address
0 0 3 8H
0 0 4 CH
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
RCK1 RCK0 HSC5 HSC4 HSC3 HSC2 HSC1 HSC0
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value
00000000B
R/W : Can be read and written
tinst : Instruction cycle
: Initial value

HSC5 to HSC0[*1]	"H" width setting bit
XXXXXX	"H" width compare value of the 12-bit PPG output

[*1] X indicates an optional value.

RCK1	RCK0	Count clock selection bit
0	0	$2t_{inst}$
0	1	$4t_{inst}$
1	0	$16t_{inst}$
1	1	$256t_{inst}$

Table 11.4-1 Functions of the 12-bit PPG Control Register 1 (PPGC1/PPGC2) Bits

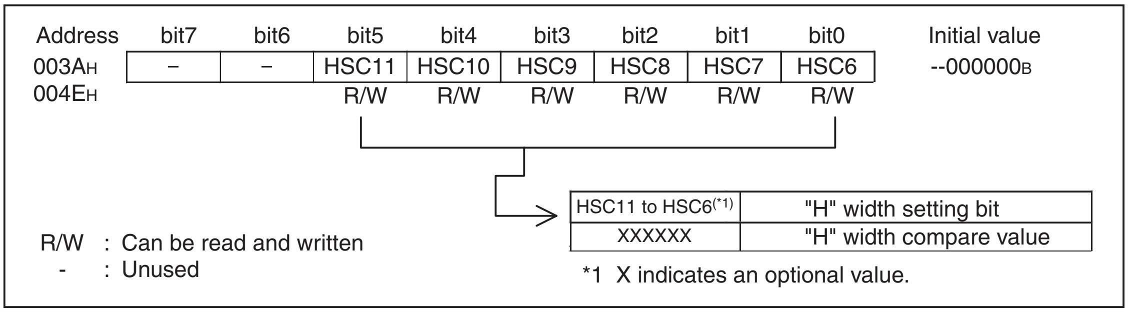
Bit		Function
bit7 bit6	RCK1, RCK0: Count clock selection bits	• These bits select a count clock for the 12-bit PPG timer from among four types of internal count clocks.
bit5 bit4 bit3 bit2 bit1 bit0	HSC5 to HSC0: "H" width setting bits	• These bits and the HCS6 to HSC11 bits of PRL11/PRL21 are used to set the count of the "H" width ("H" width compare value to be compared with the counter value) of the 12-bit PPG timer output.

11.4.2 12-Bit PPG Reload Register 1 (PRL11/PRL21)

The 12-bit PPG reload register 1 is used to set the "H" width.

■ 12-bit PPG Reload Register 1 (PRL11/PRL21)

Figure 11.4-3 12-Bit PPG Reload Register 1 (PRL11/PRL21)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$003A_H$ $004E_H$	–	–	HSC11	HSC10	HSC9	HSC8	HSC7	HSC6	$\text{--}000000_B$
			R/W	R/W	R/W	R/W	R/W	R/W	

HSC11 to HSC6[*1]	"H" width setting bit
XXXXXX	"H" width compare value

R/W : Can be read and written
- : Unused

*1 X indicates an optional value.

Table 11.4-2 Functions of the 12-Bit PPG Reload Register 1 (PRL11/PRL21) Bits

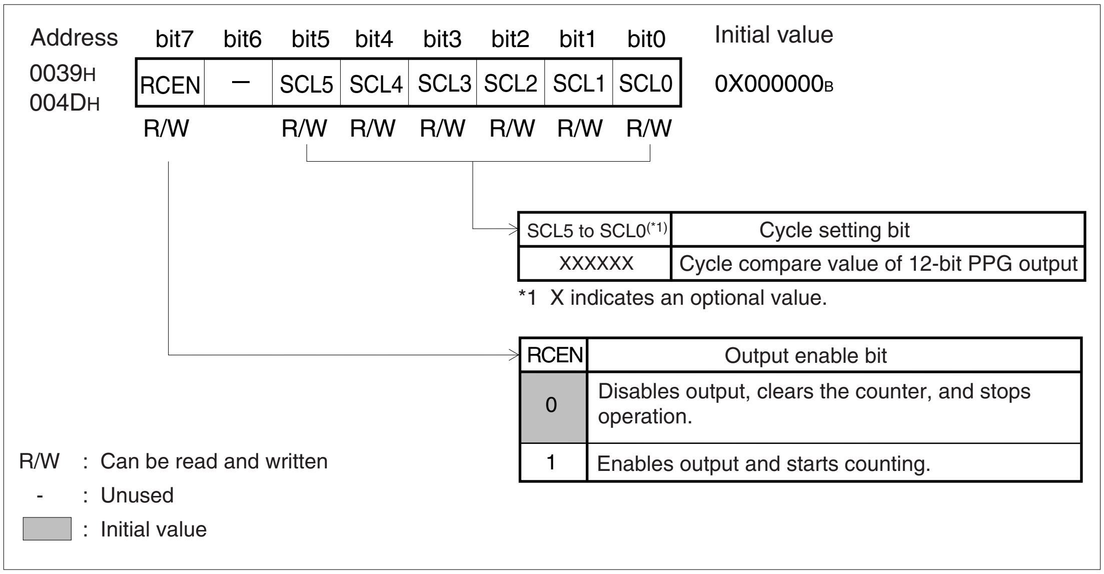
Bit		Function
bit5 bit4 bit3 bit2 bit1 bit0	HSC11 to HSC6: "H" width setting bits	These bits and the HCS0 to HCS5 bits of PPGC1/PPGC2 are used to set the count of the "H" width ("H" width compare value to be compared with the counter value) of the 12-bit PPG timer output.

11.4.3 12-Bit PPG Reload Register 2 (PRL12/PRL22)

The 12-bit PPG reload register 2 is used to enable or disable output and to set an output cycle.

■ 12-bit PPG Reload Register 2 (PRL12/PRL22)

Figure 11.4-4 PPG Reload Register 2 (PRL12/PRL22)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0039H 004DH	RCEN	—	SCL5	SCL4	SCL3	SCL2	SCL1	SCL0	0X000000B
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

SCL5 to SCL0[*1]	Cycle setting bit
XXXXXX	Cycle compare value of 12-bit PPG output

*1 X indicates an optional value.

RCEN	Output enable bit
0	Disables output, clears the counter, and stops operation.
1	Enables output and starts counting.

R/W : Can be read and written

- : Unused

▨ : Initial value

Table 11.4-3 Functions of the 12-Bit PPG Reload Register 2 (PRL12/PRL22)

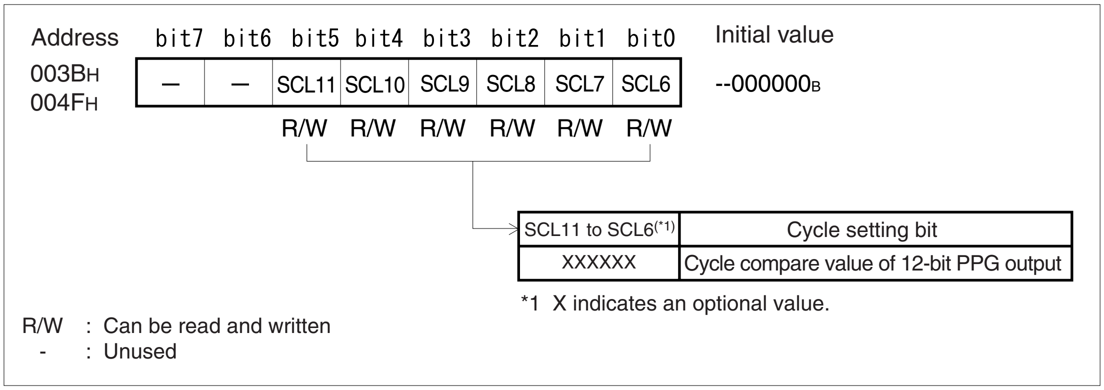
Bit		Function
bit7	RCEN: Output enable bit	When this bit is "0", the pin serves as a general-purpose port. When this bit is "1", the pin serves as a 12-bit PPG output pin. Writing "0" to this bit clears and stops the counter. Writing "1" to this bit starts the counter.
bit6	Unused	• The value read from this bit is undefined. • Writing to this bit has no effect on the operation.
bit5 bit4 bit3 bit2 bit1 bit0	SCL5 to SCL0: Cycle setting bits	These bits and the SCL6 to SCL11 bits of PRL13/PRL23 are used to set the length of the output cycle (cycle compare value to be compared with the counter value) with a counter value. **Note:** Set a value from "000000000010" to "111111111111" (002$_H$ to FFF$_H$). If 000$_H$ is set, output stops in the state existing at setting and that state is retained.

11.4.4 12-Bit PPG Reload Register 3 (PRL13/PRL23)

The 12-bit PPG reload register 3 is used to set an output cycle.

■ 12-bit PPG Reload Register 3 (PRL13/PRL23)

Figure 11.4-5 12-Bit PPG Reload Register 3 (PRL13/PRL23)



SCL11 to SCL6[*1]	Cycle setting bit
XXXXXX	Cycle compare value of 12-bit PPG output

*1 X indicates an optional value.

R/W : Can be read and written
- : Unused

Table 11.4-4 Functions of the 12-Bit PPG Reload Register 3 (PRL13/PRL23)

Bit		Function
bit7 bit6	Unused	• The value read from this bit is undefined. • Writing to this bit has no effect on the operation.
bit5 bit4 bit3 bit2 bit1 bit0	SCL11 to SCL6: Cycle setting bits	These bits and the SCL0 to SCL5 bits of PRL12/PRL22 are used to set the length of the output cycle (cycle compare value to be compared with the counter value) with a counter value. **Note:** Set a value from "000000000010" to "111111111111" (002_H to FFF_H). If 000_H is set, the output stops in the state existing at setting and that state is retained.

11.5 Operation of the 12-Bit PPG Timer

The 12-bit PPG timer generates PPG output that can independently set output cycles and "H" pulse widths.

■ Operation of the 12-Bit PPG Timer

The settings shown in Figure 11.5-1 "12-Bit PPG Timer Settings" are required for operation of the 12-bit PPG timer.

Figure 11.5-1 12-Bit PPG Timer Settings

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PPGC2 PPGC1	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0
	◎	◎	◎	◎	◎	◎	◎	◎

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PRL21 PRL11	–	–	HSC11	HSC10	HSC9	HSC8	HSC7	HSC6
			◎	◎	◎	◎	◎	◎

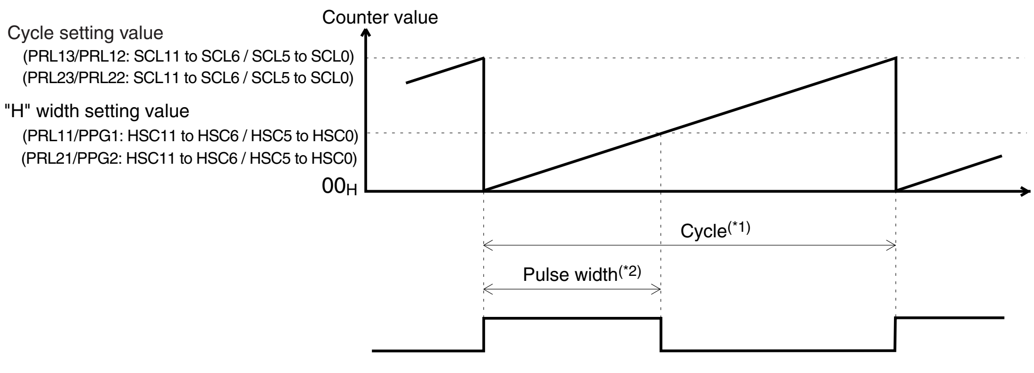
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PRL22 PRL12	RCEN	–	SCL5	SCL4	SCL3	SCL2	SCL1	SCL0
	1		◎	◎	◎	◎	◎	◎

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PRL23 PRL13	–	–	SCL11	SCL10	SCL9	SCL8	SCL7	SCL6
			◎	◎	◎	◎	◎	◎

◎ : Used
1 : Set to "1."
0 : Set to "0."

If output of a 12-bit PPG timer is enabled, the 12-bit counter starts counting from zero in synchronization with the selected count clock and holds the PPG pin at the "H" level until the counter value reaches the "H" width compare value. Next, the 12-bit counter keeps the PPG pin at the "L" level until the counter value reaches the cycle compare value. When the values match, the 12-bit counter is cleared and starts counting again from zero. Because the "H" width and cycle can be set independently, a remote control transmission frequency can also be set.

Figure 11.5-2 Operation of the 12-Bit PPG Timer


Counter value
Cycle setting value
(PRL13/PRL12: SCL11 to SCL6 / SCL5 to SCL0)
(PRL23/PRL22: SCL11 to SCL6 / SCL5 to SCL0)
"H" width setting value
(PRL11/PPG1: HSC11 to HSC6 / HSC5 to HSC0)
(PRL21/PPG2: HSC11 to HSC6 / HSC5 to HSC0)
00H
Cycle(*1)
Pulse width(*2)

*1 Cycle = count clock cycle x cycle compare value
*2 Pulse width = count clock cycle x "H" width compare value

11.6 Notes on Using the 12-Bit PPG Timer

This section provides notes on using the 12-bit PPG timer.

■ Notes on Using the 12-Bit PPG Timer

❍ Limiting the "H" width setting value

Set the "H" width setting bits (HSC5 to HSC0 bits of PPGC1/PPGC2 and HSC11 to HSC6 bits of PRL11/PRL21) of the PPG control register with a value from "000000000010_B" to "111111111111_B" ("001_H" to "FFF_H"). If 00_H is set, the PPG pin outputs the "H" level for $0.5t_{inst}$. Set a value ("H" width) that is smaller than the values of the cycle setting bits (SCL5 to SCL0 of PRL12/PRL22 and SCL11 to SCL6 of PRL13/PRL23). If the set value is greater than or equal to the values of the cycle setting bits, the PPG pin output is held at the "H" level.

❍ Resolution

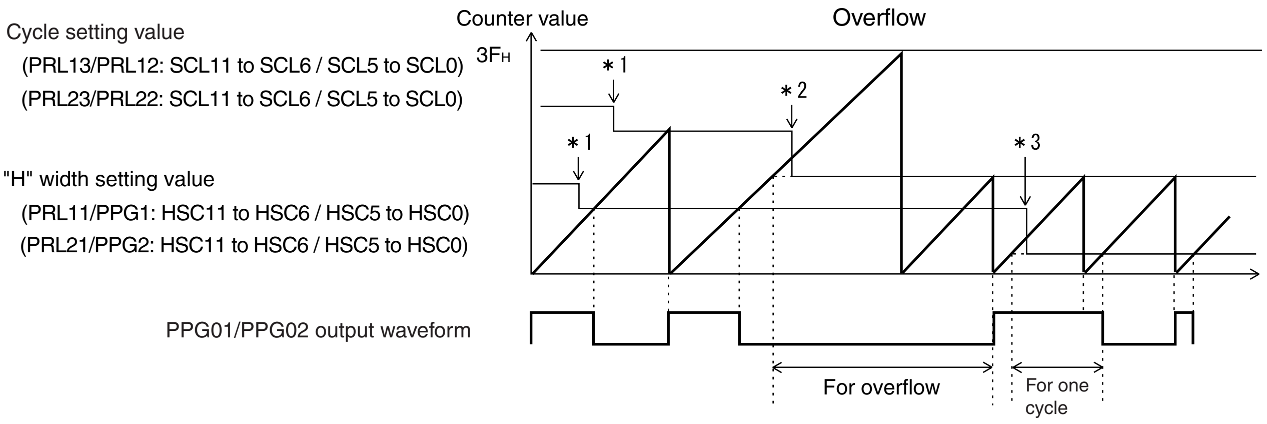
The maximum resolution of the "H" width is 1/4095 of the cycle (when the cycle setting value is "111111111111_B"("FFF_H")). If, however, the cycle setting value is made small, the minimum resolution of the "H" width is restricted to 1/2 of the cycle (when the cycle setting value is "000000000010_B" ("02_H")).

❍ Changing the setting values during operation

The 12-bit PPG timer directly compares the 12-bit counter/"H" width setting bits with the counter/cycle setting bits. For this reason, if a small value is set while the counter is operating, the match detection cycle may lengthen until a match is detected again because the counter will overflow. The "H" width may also lengthen until a match is detected in the next cycle.

Figure 11.6-1 "Setting Value Changes during 12-Bit PPG Timer Operation" shows the setting value changes made while the 12-bit PPG timer is operating.

Figure 11.6-1 Setting Value Changes during 12-Bit PPG Timer Operation

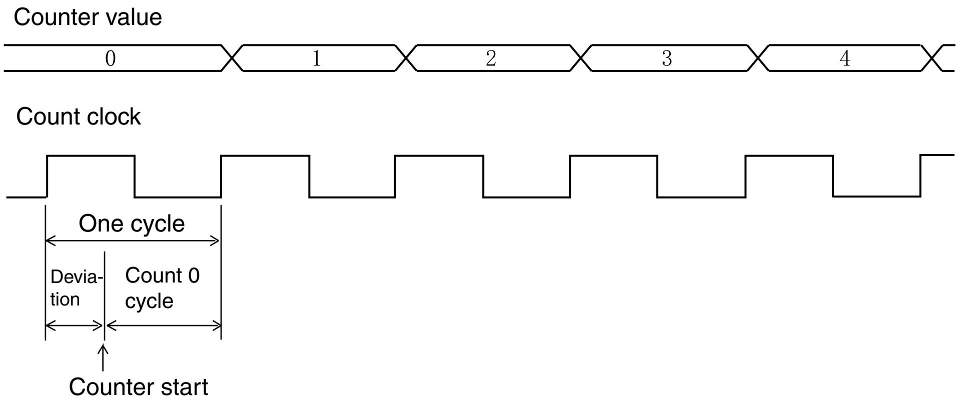

Cycle setting value
(PRL13/PRL12: SCL11 to SCL6 / SCL5 to SCL0)
(PRL23/PRL22: SCL11 to SCL6 / SCL5 to SCL0)
"H" width setting value
(PRL11/PPG1: HSC11 to HSC6 / HSC5 to HSC0)
(PRL21/PPG2: HSC11 to HSC6 / HSC5 to HSC0)
Counter value
3FH
Overflow
*1
*1
*2
*3
PPG01/PPG02 output waveform
For overflow
For one cycle

*1 Since the counter value during operation is smaller than the changed setting value, the setting value is valid within the cycle.
*2 No matching is detected and the counter overflows because the cycle that is set is smaller than the counter value used during operation.
*3 No matching is detected until the next cycle because the "H" width that is set is smaller than the counter value used during operation.

❍ Deviation

The counter start by the program is asynchronous to the start of incrementing with the selected count clock. For this reason, the deviation that exists until a match of the counter value, "H"-width compare value, and cycle compare value is detected may become shorter by up to one count clock cycle.

Figure 11.6-2 Deviation until Counting Starts



11.7 Program Example of the 12-Bit PPG Timer

This section provides program examples of the 12-bit PPG timer.

■ Program Example of the 12-Bit PPG Timer

❍ Program specifications

- A remote transmission frequency is set with a cycle of about 12 µs and a duty ratio of 33%.
- When the selected clock is $2t_{inst}$ at the highest main clock speed for the main clock oscillation of 10 MHz, the cycle compare value when the cycle becomes about 12 µs is shown below.

  Cycle compare value
  (SCL5 to SCL0 bits of PRL22 and SCL11 to SCL6 bits of PRL23) = 12 µs/ (2 x 4/10MHz)
  = 15

- The "H" width compare value when the duty ratio becomes 33% is shown below.

  "H" width compare value
  (HSC5 to HSC0 bits of PPGC2
  and HSC11 to HSC6 bits of PRL21) = 33/100 x cycle compare value
  = 0.33 x 15
  ≒ 4.95

❍ Coding example

```
PPGC2   EQU        0038H         ;  PPG control register
PRL21   EQU        003AH         ;  PPG reload register 1
PRL22   EQU        0039H         ;  PPG reload register 2
PRL23   EQU        003BH         ;  PPG reload register 3
;--------Main program-------------------------------------------------------------
        CSEG                     ;[CODE SEGMENT]
        :
        MOV PPGC2,#01000101B     ;Selects 4tinst as the count clock and sets
                                  the "H" width compare value.
        MOV PRL21,#00H           ;
        MOV PRL22,#10001111B     ;Enables output, starts operation, and sets
                                  a cycle compare value.
        MOV PRL23,#00H
         :
        ENDS
;--------------------------------------------------------------------------------
        END
```

CHAPTER 12 16-BIT TIMER/COUNTER

This chapter describes the functions and operations of the 16-bit timer/counter.

12.1 Overview of the 16-bit Timer/Counter

The 16-bit timer/counter has an interval timer function and a counter function. The interval timer function counts up in synchronization with the internal count clock (the oscillator frequency divided into four cycles). The counter function counts up by detecting a prespecified edge of a pulse that is input to the external pin. One of these functions can be selected.

■ Interval Timer Function

The interval timer function generates an interrupt at prespecified intervals.

The 16-bit counter counts up from a setting value in synchronization with the internal count clock that divides the oscillator frequency into four cycles and generates an interrupt if the counter value overflows.

- Interval timer operation up to an internal count clock times 2^{16} is possible.
- Use the interrupt processing routine to reset the interval time to generate an interrupt repeatedly.

Table 12.1-1 "Range for Interval Time" shows the range for the interval time.

Table 12.1-1 Range for Interval Time

Internal count clock cycle	Interval time
$1t_{ins}$	$1t_{inst}$ to $2^{16}t_{inst}$

t_{inst}: Instruction cycle (the oscillator frequency divided into four cycles)

Reference:

An example of calculating the interval time is shown below.

The interval time at which the main clock oscillation (F_{CH}) is 12.5 MHz and the value of the timer count register (TCR) is "0000_H" can be calculated from the following formula:

$$\begin{aligned}\text{Interval time} &= (4/F_{CH}) \times (2^{16} - \text{TCR register value})\\ &= (4/12.5\ \text{MHz}) \times 65536\\ &\fallingdotseq 21.0\ \text{ms}\end{aligned}$$

■ Counter Function

The counter function detects the edge of a pulse that is input to the external pin (EC pin) and counts it.

- Counts up every time a prespecified edge of the external input is detected and generates an interrupt if the counter value overflows.
- Can detect the pulse width of the external input at a minimum of two instruction cycles.
- Can be set to detect both rising and falling edges.

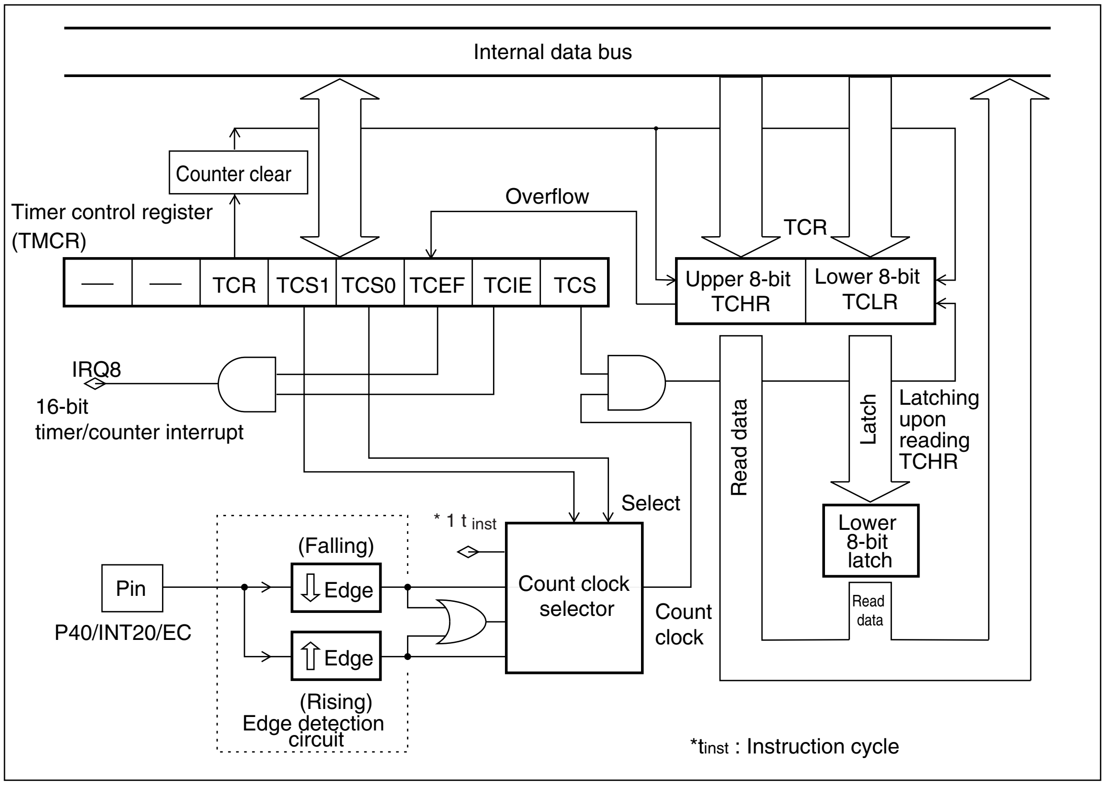
12.2 Configuration of the 16-bit Timer/Counter

The 16-bit timer/counter consists of the following five blocks:

- **Count clock selector**
- **Edge detection circuit**
- **Timer count register (TCR)**
- **Timer control register (TMCR)**
- **Lower 8-bit latch**

■ Block Diagram of the 16-bit Timer/Counter

Figure 12.2-1 Block Diagram of the 16-bit Timer/Counter Section



❍ **Count clock selector**

Selects the internal count clock (1 t_{inst}) in interval timer function mode. Selects the output of the edge detection circuit in counter function mode. The selected signal will be used as the clock by the 16-bit counter (TCR register) to count up.

❍ **Edge detection circuit**

Operates in counter function mode and detects the rising and/or falling edges of a pulse that is input from the EC3 pin.

❍ Timer count register (TCR)

Stores a value from which the 16-bit counter counts up. If the counter value overflows, the TMCR register interrupt request flag bit is set (TMCR: TCEF=1).

❍ Timer control register (TMCR)

Selects a function, enables and disables operation, controls interrupts, and checks the status.

❍ Lower 8-bit latch

Stores the lower 8 bits of the 16-bit counter when the TCR register upper 8-bit value (TCHR) is read. Since the lower 8-bit value of the counter (TCLR) is read from this lower 8-bit latch, a correct 16-bit counter value can be read even while the counter is counting up.

If, while the counter is operating, the lower 8-bit value is read before the upper 8-bit value, a correct value may not be read, depending on the counter carry.

Therefore, always use a word transfer instruction to read the TCR register.

❍ Interrupt related to the timer/counter

IRQ8:

Generates an interrupt request if interrupt request output is enabled (TMCR: TCE = 1) when the counter value overflows either in interval timer or counter function mode.

12.3 Pin of the 16-bit Timer/Counter

This section describes the pin related to the 16-bit timer/counter and shows a block diagram.

■ Pin Related to the 16-bit Timer/Counter

The pin related to the 16-bit timer/counter is the P40/INT20/EC pin. It functions both as a general-purpose input port (P84) and as an external pulse input pin for the counter (EC).

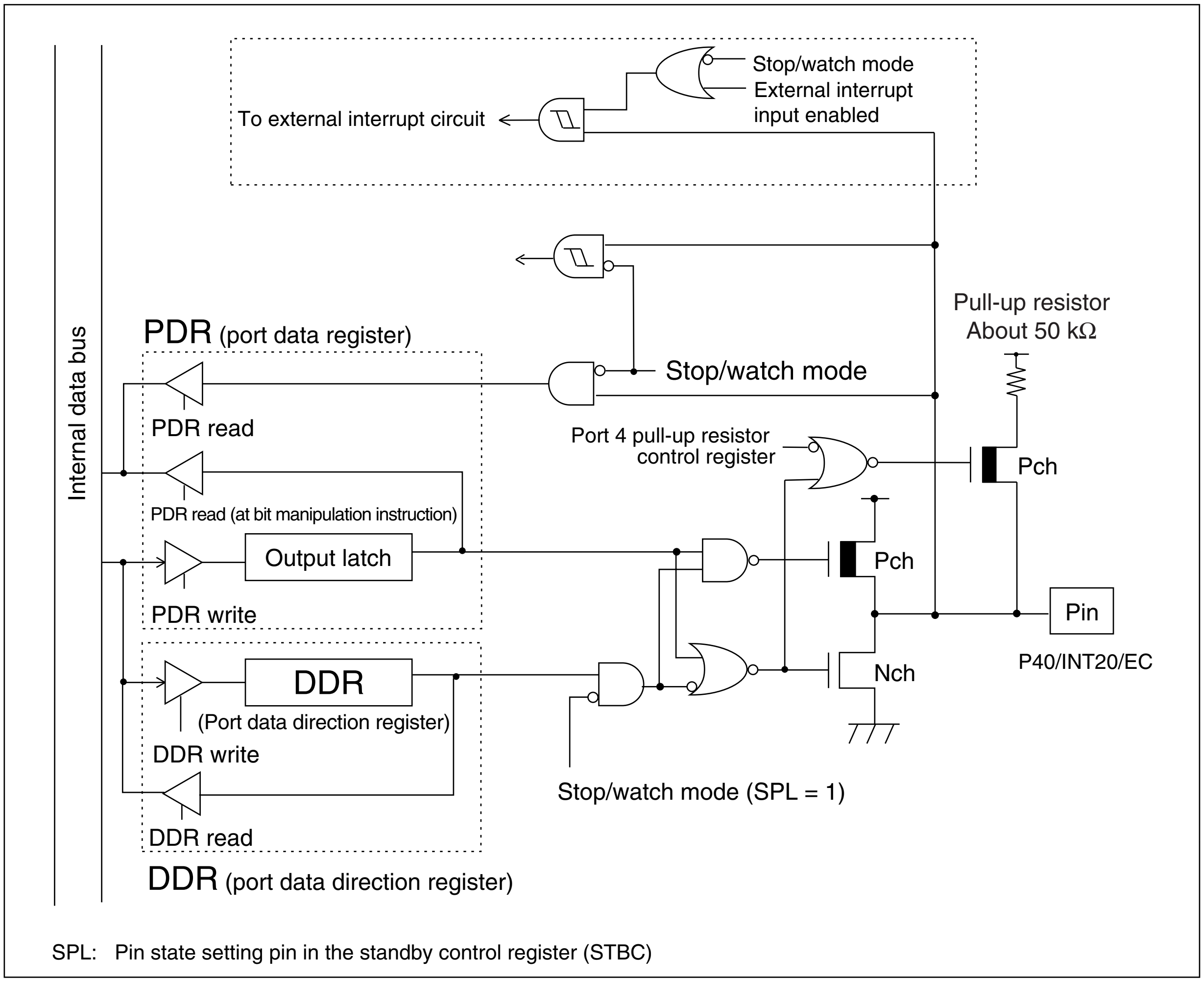
EC:

Counts a prespecified edge of a pulse that is input to this pin in counter function mode.

When using this pin in counter function mode, set the P40/INT20/EC pin as the input port for the port data direction register (bit 0 of DDR4 = 0).

■ Block Diagram of the Pin Related to the 16-bit Timer/Counter

Figure 12.3-1 Block Diagram of 16-Bit Timer/Counter Pins



Stop/watch mode
External interrupt input enabled
To external interrupt circuit
Pull-up resistor About 50 kΩ
Internal data bus
PDR (port data register)
Stop/watch mode
PDR read
Port 4 pull-up resistor control register
Pch
PDR read (at bit manipulation instruction)
Output latch
Pch
Pin
PDR write
P40/INT20/EC
DDR
Nch
(Port data direction register)
DDR write
Stop/watch mode (SPL = 1)
DDR read
DDR (port data direction register)

SPL: Pin state setting pin in the standby control register (STBC)

Reference:

If "pull-up resistor available" is selected in the port 4 pull-up resistor control register, the pins are set to the "H" level (pull-up state), not the high impedance state, in stop/watch mode (SPL bit of STBC = 1). However, the pull-up is disabled during a reset and the pins enter the Hi-z state.

12.4 Registers of the 16-bit Timer/Counter

This section describes the registers related to the 16-bit timer/counter.

■ Registers Related to the 16-bit Timer/Counter

Figure 12.4-1 Registers Related to the 16-bit Timer/Counter

TMCR (timer control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 C H	–	–	TCR	TCS1	TCS0	TCEF	TCIE	TCS	--000000B
			W	R/W	R/W	R/W	R/W	R/W	

TCR (timer count register)

Upper bits (TCHR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 D H									00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	······While the counter is stopped
	R	R	R	R	R	R	R	R	······While the counter is operating

Lower bits (TCLR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 E H									00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	······While the counter is stopped
	R	R	R	R	R	R	R	R	······While the counter is operating

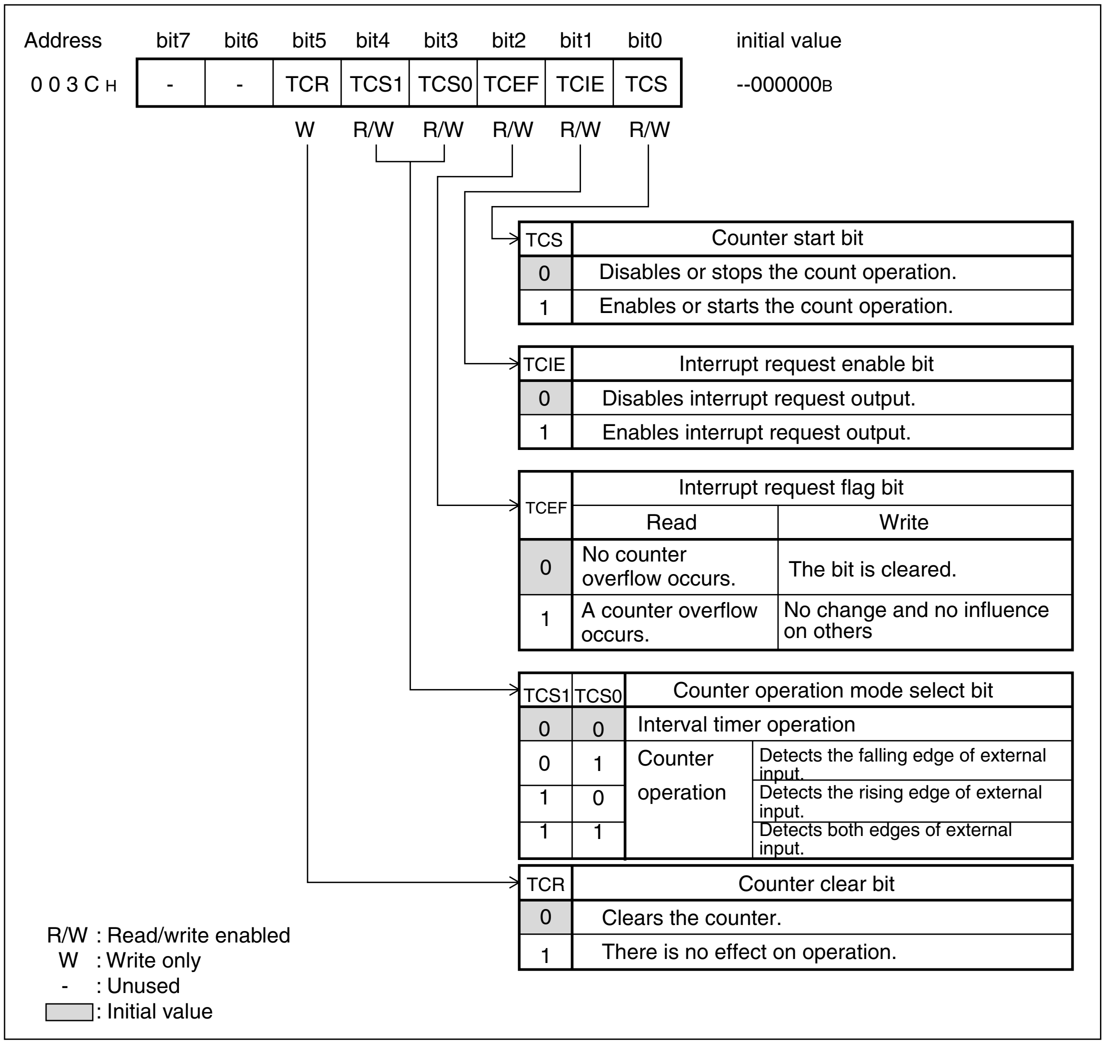
R/W : Read/write enabled
R : Read only
– : Undefined

12.4.1 Timer Control Register (TMCR)

The timer control register (TMCR) selects a 16-bit timer/counter function (either the interval timer or counter function), sets the operating conditions, enables or disables operation, clears the counter, controls interrupts, and checks the status.

■ Timer Control Register (TMCR)

Figure 12.4-2 Timer Control Register (TMCR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	initial value
003CH	-	-	TCR	TCS1	TCS0	TCEF	TCIE	TCS	--000000B
			W	R/W	R/W	R/W	R/W	R/W	

TCS	Counter start bit
0	Disables or stops the count operation.
1	Enables or starts the count operation.

TCIE	Interrupt request enable bit
0	Disables interrupt request output.
1	Enables interrupt request output.

TCEF	Interrupt request flag bit	
	Read	Write
0	No counter overflow occurs.	The bit is cleared.
1	A counter overflow occurs.	No change and no influence on others

TCS1	TCS0	Counter operation mode select bit	
0	0	Interval timer operation	
0	1	Counter operation	Detects the falling edge of external input.
1	0		Detects the rising edge of external input.
1	1		Detects both edges of external input.

TCR	Counter clear bit
0	Clears the counter.
1	There is no effect on operation.

R/W : Read/write enabled
W : Write only
- : Unused
▨ : Initial value

Table 12.4-1 Functions of the Timer Control Register (TMCR) Bits

Bit name		Function
Bit 7 Bit 6	Unused bits	• The read value is undefined. • Writing has no effect on operation.
Bit 5	TCR: Counter clear bit	• Clears the timer count register (TCR). • Writing 0 to this bit clears the timer count register to 0000_B. Writing 1 to this bit has no effect and has no effect on other bits. **Reference:** If this bit is read, the value is always 1.
Bit 4 Bit 3	TCS1, TCS0: Counter operation mode select bits	• Switches the interval timer and counter functions. Setting these bits to 00_B selects the interval timer function that results in operation using the internal count clock. • Selecting an edge to be detected (falling, rising, or both) from the external count clock causes operation as a 16-bit counter. **Note:** When using the counter function (when TCS1 and TCS0 are not 00_B), set the P40/INT20/EC pin as an input port.
Bit 2	TCEF: Interrupt request flag bit	• Set to 1 if the counter overflows. • Setting this bit and the interrupt request enable bit (TCIE) to 1 outputs an interrupt request. • Cleared to 0 for a write. Setting this bit to 1 has no effect and does not affect operation.
Bit 1	TCIE: Interrupt request enable bit	• Enables or disables interrupt request output to the CPU. • Setting this bit and the interrupt request flag bit (TCEF) to 1 outputs an interrupt request.
Bit 0	TCS: Starts or stops the counter.	• Starts and stops the counter. • Writing 1 to this bit starts the count operation of the timer count register (TCR), which counts up according to the count clock. Writing 0 to this bit stops the count operation, and the TCR retains the counter value.

Note:

When clearing the interrupt request flag bit (TMCR: TCEF), do not overwrite the interrupt request enable bit (TMCR: TCIE) at the same time.

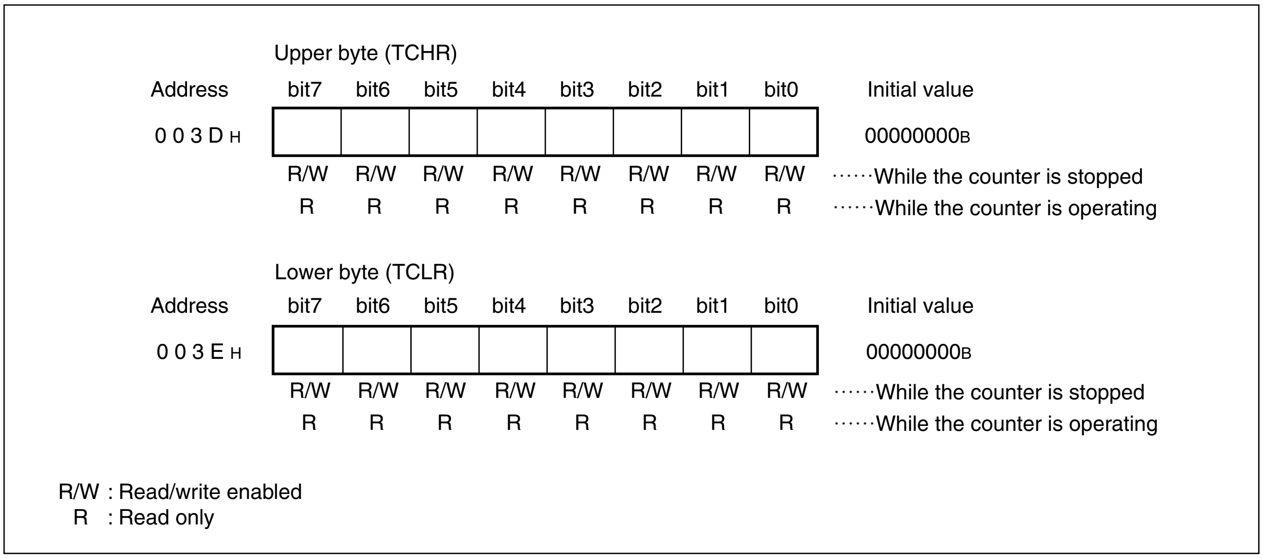
12.4.2 16-bit Timer Count Register (TCR)

The timer count register (TCR) is a 16-bit up counter. The counter counts up from the setting value written in this register.

■ Timer Count Register (TCR)

Figure 12.4-3 "16-bit Timer Count Register (TCR)" shows the bit configuration of the 16-bit timer count register.

Figure 12.4-3 16-bit Timer Count Register (TCR)



Upper byte (TCHR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 D H									00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	……While the counter is stopped
	R	R	R	R	R	R	R	R	……While the counter is operating

Lower byte (TCLR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 E H									00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	……While the counter is stopped
	R	R	R	R	R	R	R	R	……While the counter is operating

R/W : Read/write enabled
R : Read only

Both in the interval timer and counter function modes, set a counter initial value in this register while the counter operation is disabled (TMCR: TCS=0). When the counter operation is enabled (TMCR: TCS=1), the counter counts up from the value written in this register. While the counter is stopped (TMCR: TCS=0), the TCR register maintains its value. If the counter is cleared (TMCR: TCR=0), the TCR register (counter) becomes 0000_H.

After the counter is cleared, writing a value in the TCR register sets the counter to the value written.

You can calculate the value in the TCR register in interval timer mode as follows. Note that the instruction cycle is the oscillator frequency divided into four cycles (4/Fc).

TCR register value = 2^{16} - (Interval time/Instruction cycle)

Set the upper 8 bits as the TCHR register and the lower 8 bits as the TCLR register.

Note:

The value that is set in this register is valid only when the counter is started for the first time. The counter, if it overflows, counts up from 0000_H.

A value must be written in this register while the counter is stopped (TMCR: TCS=0).

A value can be read from this register even while the counter is operating.

Always use a word transfer instruction (such as MOVW A, 003DH) to read this register.

12.5 16-bit Timer/Counter Interrupts

A 16-bit timer/counter interrupt is caused by:

- **An overflow in interval timer function mode ($FFFF_H$ --> 0000_H)**
- **An overflow in the 16-bit counter function mode ($FFFF_H$ --> 0000_H)**

■ Interrupts in Interval Timer Function Mode

If the counter counts up from the defined counter value according to the internal count clock until it overflows, the interrupt request flag bit (TMCR: TCEF) is set to 1. If, at this time, the interrupt request enable bit is set to Enabled (TMCR: TCIE=1), an interrupt request (IRQ8) occurs in the CPU.

Use the interrupt processing routine to write 0 to the TCEF bit and clear the interrupt request.

If the counter is cleared (TMCR: TCR=0) and the counter value overflows at the same time, the TCEF bit is not set. While the TCEF bit is 1, setting the TCIE bit from Disabled to Enabled (0 to 1) immediately causes an interrupt request.

The TCEF bit is set whenever the counter value overflows regardless of the value in the TCIE bit.

■ Interrupts in Counter Function Mode

If the counter counts up from the defined counter value each time preset edge is detected until it overflows, the interrupt request flag bit (TMCR: TCEF) is set to 1. If, at this time, the interrupt request enable bit is set to Enabled (TMCR: TCIE=1), an interrupt request (IRQ8) occurs in the CPU.

Use the interrupt processing routine to write 0 to the TCEF bit and clear the interrupt request.

If the counter is cleared (TMCR: TCR=0) and the counter value overflows at the same time, the TCEF bit is not set. While the TCEF bit is 1, setting the TCIE bit from Disabled to Enabled (0 to 1) immediately causes an interrupt request.

The TCEF bit is set whenever the counter value overflows regardless of the value in the TCIE bit.

■ Register Related to the Interrupts of the 16-bit Timer/Counter and the Vector Table

Table 12.5-1 Register Related to the Interrupts of the 16-bit Timer/Counter and the Vector Table

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Bit to be set		Upper	Lower
IRQ8	ILR3 ($007D_H$)	L81 (bit 1)	L80 (bit 0)	$FFEA_H$	$FFEB_H$

For the operation of interrupts, see Section 3.4.2 "Interrupt Processing".

12.6 Operation of the Interval Timer Function

This section describes the operation of the interval timer function of the 16-bit timer/ counter.

■ Operation of the Interval Timer Function

For interval timer function operation, the setting shown in Figure 12.6-1 "Setting the Interval Timer Function" is necessary.

Figure 12.6-1 Setting the Interval Timer Function

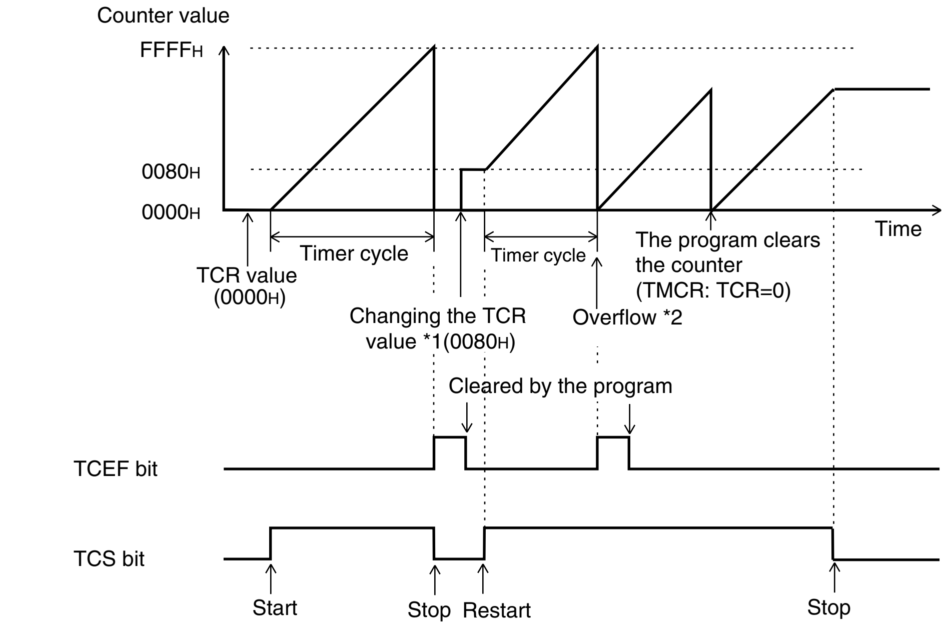
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TMCR	—	—	TCR	TCS1	TCS0	TCEF	TCIE	TCS
			1	0	0	◎	◎	◎
TCHR	Set to the initial value of the counter (upper 8 bits)							
TCLR	Set to the initial value of the counter (lower 8 bits)							

◎ : Used
1 : Set to 1
0 : Set to 0

If the counter is started (TMCR: TCS=1), the counter starts counting up from the value in the TCR register on every rising edge of the internal count clock (1 t_{inst}: the oscillator frequency divided into four cycles). If the counter overflows ($FFFF_H$ --> 0000_H), the interrupt request flag bit is set (TMCR: TCEF=1). After an overflow, the counter starts counting up from 0000_H.

Figure 12.6-2 "Operation of the Interval Timer" shows the operation of the interval timer.

Figure 12.6-2 Operation of the Interval Timer



*1: The timer operation is stopped and the TCR value changes (0000H --> 0080H).
Timer operation then starts again.

*2: The counter, when an overflow occurs, starts counting from 0000H.

Note:

Do not write a value to the TCR register while the interval timer function is operating (TMCR: TCS=1)

12.7 Operation of the Counter Function

This section describes the operation of the counter function of the 16-bit timer/ counter.

■ Operation of the Counter Function

To the counter function operation, the setting shown in Figure 12.7-1 "Setting the Counter Function", "Setting the counter function," is necessary

Figure 12.7-1 Setting the Counter Function

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR8								
	0	×	×	×	×	×	×	×
TMCR	-	-	TCR	TCS1	TCS0	TCEF	TCIE	TCS
			1	Other than 00		◎	◎	◎
TCHR	Set to the initial value of the counter (upper 8 bits)							
TCLR	Set to the initial value of the counter (lower 8 bits)							

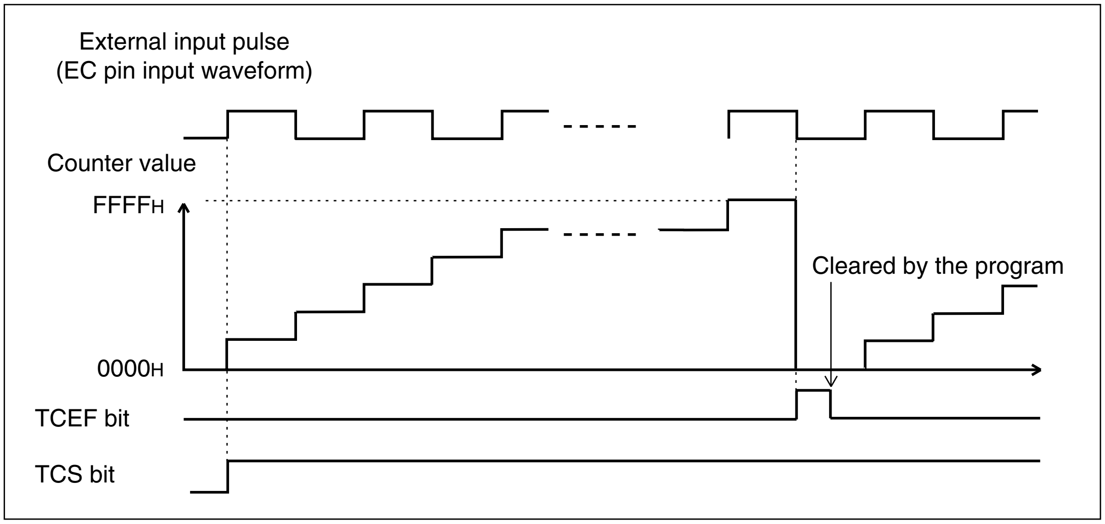
◎ : Used
x : Not used
1 : Set to 1
0 : Set to 0

If the counter is started (TMCR: TCS=1), the counter starts counting up from the value in the TCR register whenever the prespecified edge of a pulse that is input to the EC pin (the external count clock) is detected.

If the counter overflows ($FFFF_H$ --> 0000_H), the interrupt request flag bit is set (TMCR: TCEF=1).

Then, if the next prespecified edge is input, the counter starts counting up from 0000_H.Figure 12.7-2 "16-bit Counter Operation" shows the operation when the counter operation mode select bits (TMCR: TCS1, TCS0) are set to 11_B (detecting both edges) and the TCR register to 0000_H.

Figure 12.7-2 16-bit Counter Operation



Note:

Do not write a value to the TCR register while the counter function is operating (TMCR: TCS=1)

12.8 Status of the 16-bit Timer/Counter in Each Mode

This section describes the operations of switching to the sleep and stop modes and of receiving a suspend request while the 16-bit timer/counter is operating.

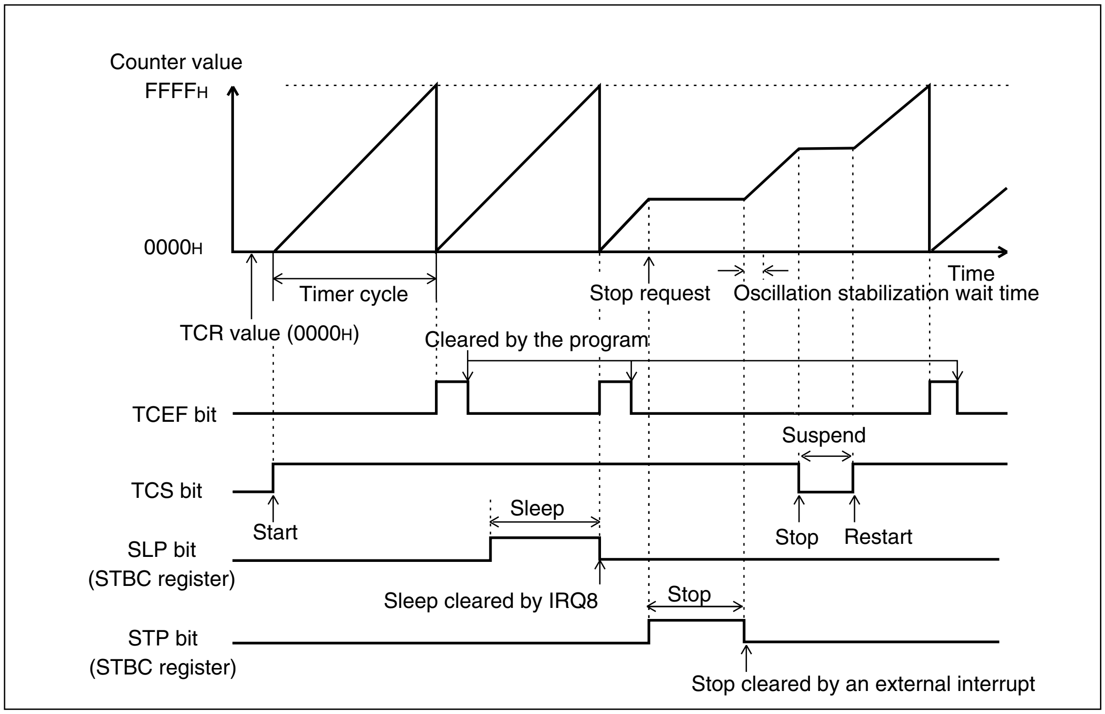
■ Operation in Low Power Consumption (Standby) Mode and when the Counter is Suspended

Figure 12.8-1 "Operation of the Counter in Low Power Consumption (Standby) Mode and when the Counter is Suspended" shows the status of the counter value upon switching to the sleep and stop modes and upon receiving a suspend request while the interval timer or counter function is operating.

In stop mode, the counter stops, retaining the value. If stop mode is cleared by an external interrupt, the counter starts counting from the retained value. Therefore, the initial interval time and the input pulse edge count cannot be correct values. After stop mode is cleared, initialize the 16-bit timer/counter again.

The movement for "transition to watch mode (TMD = 1)" and "release of watch mode (TMD = 0)" is the same as that for "transition to stop mode" and "release of stop mode". Watch mode is released by a clock interrupt and external interrupt.

Figure 12.8-1 Operation of the Counter in Low Power Consumption (Standby) Mode and when the Counter is Suspended



The counter value is retained while the counter is stopped (TMCR: TCS=0).

12.9 Notes on Using the 16-bit Timer/Counter

This section contains notes on using the 16-bit timer/counter.

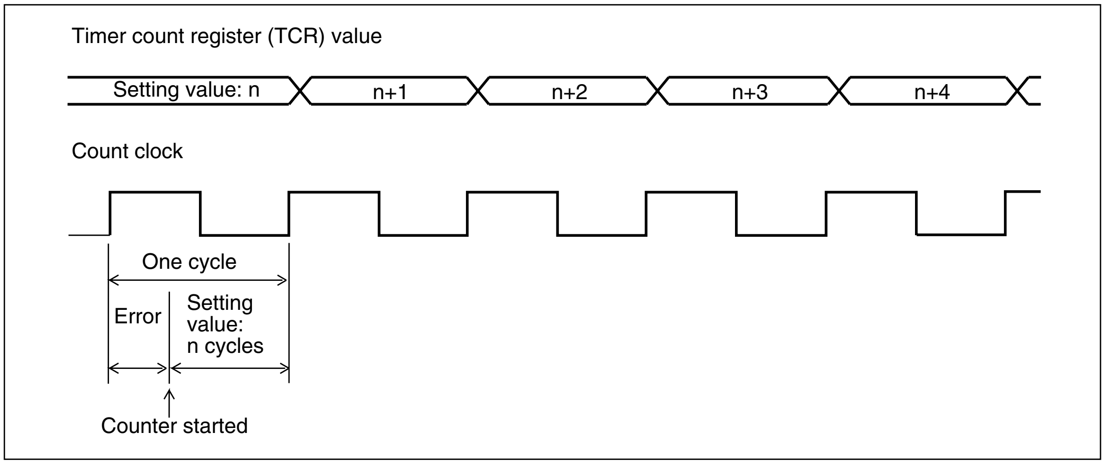
■ Notes on Using the 16-bit Timer/Counter

❍ Error

In interval timer function mode, starting of the counter by the program and starting of count-up by the internal count clock are asynchronous. Thus, an error of one less instruction cycle at the most may exist in the time elapsing until the counter overflows.

Figure 12.9-1 "Error Until the Count Operation is Started" shows an error until the count operation is started.

Figure 12.9-1 Error Until the Count Operation is Started



❍ Notes on setting the program

- Write a value to the TCR register while the counter operation is stopped (TMCR: TCS=0). A value can be read even while the counter is counting. However, always use a word transfer instruction (such as MOVW A, dir) to read this register.
- Change the counter operation mode select bits (TMCR: TCS1, TCS0) while the counter is stopped (TMCR: TCS=0), an interrupt is disabled (TMCR: TCIE=0), and the interrupt request is cleared (TCEF=0).
- If the interrupt request flag bit (TMCR: TCEF) is 1 and the interrupt request is enabled (TMCR: TCIE=1), the counter cannot restored after interrupt processing. Always clear the TCEF bit.
- If the counter is cleared (TMCR: TCR=0) and the counter value overflows at the same time, the interrupt request flag bit (TMCR: TCEF) is not set.

12.10 Programe Example of the 16-bit Timer/Counter

This section contains sample programs for the 16-bit timer/counter.

■ Program Example of the Interval Timer Function

❍ Processing specification

- Generate a 20-ms interval timer interrupt.
- Use the interrupt processing routine to reset the TCR register and generate an interrupt repeatedly.
- The following shows the TCR register value for an interval time of 20 ms when the oscillator frequency is 12.5 MHz.
- TCR register value = 2^{16} - (20 ms / (4/12.5 MHz)) = 3036 (0BDCH)

❍ Coding example

```
TMCR    EQU      003CH             ;Address of the timer control reister
TCHR    EQU      003DH             ;Upper address of the timer count
                                    register
TCLR    EQU      003EH             ;Lower address of the timer count
                                    register
TCEF    EQU      TMCR:2            ;Definition of the interrupt request
                                    flag bit
TCS     EQU      TMCR:0            ;Definition of the count start bit
ILR3    EQU      007DH             ;Address of the interrupt level setting
                                    register
INT_V   DSEG     ABS               ;  [DATA SEGMENT]
        ORG      0FFEAH
IRQ8    DW       WARI              ;Setting interrupt vector
INT_V   ENDS
;----------Main program----------------------------------------------------------
        CSEG                       ;  [CODE SEGMENT]
                                   ;Stack pointer (SP) and other
                                    are assumed to have been initialized
           :
        CLRI                       ;Interrupt disable
        CLRB     TCS               ;Count operation stop
        MOV      ILR3,#11111101B   ;Setting interrupt level (level 1)
        MOV      TCHR,#0BH         ;Set the 25 ms timer data
        MOV      TCLR,#0DCH
        MOV      TMCR,#00100011B   ;Retain the counter value,
                                    set the interval timer operation,
                                    clearing the interrupt request
                                    flag, enabling interrupt request
                                    output, and start counter operation
        SETI                       ;Interrupt enable
           :
;----------Interrupt program----------------------------------------------------
WARI    MOV      TMCR,#00100000B   ;Clearing interrupt request flag
                                    and stop counter operatin
        PUSHW    A
        XCHW     A,T
        PUSHW    A
        MOVW     A,TCHR            ;Add the time from the overflow to the
                                    interrupt acceptance
        MOVW     A,#0BDCH          ;25 ms timer data (at 10 MHz)
        CLRC
        ADDCW    A                 ;Here, an overflow during addition is
                                    not considered
        MOVW     TCHR,A            ;Strictly, the time while the counter
                                    is stopped must be added
        MOV      TMCR,#00100011B   ;Enable the interrupt, and
                                    start counting
           :
        User processing
        POPW     A
        XCHW     A,T
        POPW     A
        RETI
        ENDS
; ---------------------------------------------------------------------------------
        END
```

■ Program Example of the Counter Function

❍ Processing specifications

- Generate an interrupt whenever the rising edge of a pulse being input to the EC pin is counted 10,000 times.
- Use the interrupt processing routine to reset the TCR register and generate an interrupt repeatedly.
- The following shows the TCR register value at which the counter overflows when a rising edge is detected 10,000 times.
    - TCR register value = 2^{16} - 10000 = 65536 - 10000 = 55536 = $D8F0_H$

❍ Coding example

```
DDR4    EQU     0011H
TMCR    EQU     003CH            ;Address of the timer control reister
TCHR    EQU     003DH            ;Upper address of the timer count
                                  register
TCLR    EQU     003EH            ;Lower address of the timer count
                                  register
TCEF    EQU     TMCR:2           ;Definition of the interrupt request
                                  flag bit
TCS     EQU     TMCR:0           ;Definition of the count start bit
ILR3    EQU     007DH            ;Address of the interrupt level setting
                                  register
INT_V   DSEG    ABS              ; [DATA SEGMENT]
        ORG     0FFEAH
IRQ8    DW      WARI             ;Setting interrupt vector
INT_V   ENDS
;----------Main program----------------------------------------------------------
        CSEG                     ; [CODE SEGMENT]
                                 ;Stack pointer (SP) and other
                                  are assumed to have been initialized
          :
        MOV     DDR8,#00000000B  ;P40/INT20/EC
        CLRI                     ;Interrupt disable
        CLRB    TCS              ;Count operation stop
        MOV     ILR3,#11111101B  ;Setting interrupt level (level 1)
        MOV     TCHR,#0D8H       ;Initialize the counter value
        MOV     TCLR,#0F0H
        MOV     TMCR,#00110011B  ;Retain the counter value, set the counter
                                  function (selecting the rising edge of
                                  external input), clearing the interrupt
                                  request flag, enabling the interrupt request
                                  output, and enabling the counter operation
        SETI                     ;Interrupt enable
;----------Interrupt program-----------------------------------------------------
WARI    CLRB    TCEF             ;Clearing interrupt request flag
        PUSHW   A
        XCHW    A,T
        PUSHW   A
        CLRB    TCS              ;Stop counter operation
        MOV     A,#0D8H          ;Initialize the counter value
        MOV     TCHR,A           ;Here, the pulse after overflow is ignored
        MOV     A,#0F0H
        MOV     TCLR,A
        SETB    TCS              ;Restart the count operation, and
                                  start counting 10,000 pulses from here
          :
        User processing
          :
        POPW    A
        XCHW     A,T
        POPW     A
        RETI
        ENDS
; --------------------------------------------------------------------------------
        END
```

CHAPTER 13 EXTERNAL INTERRUPT CIRCUIT 1 (EDGE)

This chapter describes the functions and operation of external interrupt circuit 1 (edge).

13.1 Overview of External Interrupt Circuit 1

External interrupt circuit 1 detects the edge of signals that are input from the eight external interrupt pins and outputs an interrupt request to the CPU.

■ Functions of External Interrupt Circuit 1

External interrupt circuit 1 detects the specified edges of signals that are input to the external interrupt pins and outputs interrupt requests to the CPU. These interrupts can be used to wake up the device from standby mode and restore the normal operating state (the main-RUN state).

- External interrupt pins: 4 pins (P60/INT10 to P63/INT13/X0A)
- External interrupt sources: Input of edge signals to the external interrupt pins (rising or falling edge).
- Interrupt control: Used to enable or disable an interrupt request output according to the interrupt request enable bits (EIE0 to EIE3) of external interrupt 1 control registers 1 and 2 (EIC1, EIC2).
- Interrupt flag: Used to detect the specified edge according to the external interrupt request flag bits (EIR0 to EIR3) of external interrupt 1 control registers 1 and 2 (EIC1, EIC2).
- Interrupt requests: Generated according to external interrupt sources (IRQ0, IRQ1).

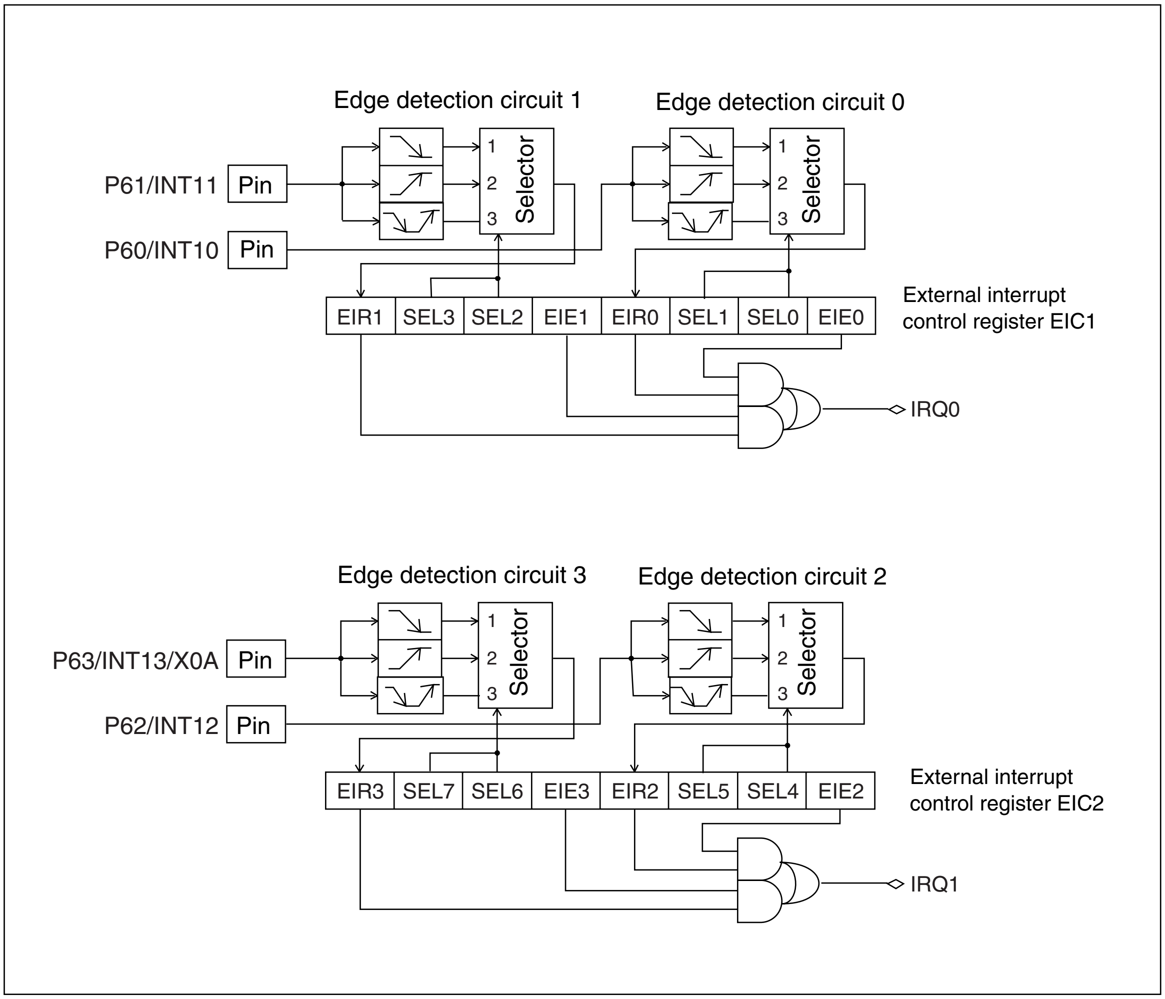
13.2 Configuration of the External Interrupt Circuit 1

External interrupt circuit 1 consists of the following two blocks:

- **Edge detection circuits (0 to 7)**
- **External interrupt control registers 1, 2, 3, and 4 (EIC1, EIC2, EIC3, and EIC4)**

■ Block Diagram of External Interrupt Circuit 1

Figure 13.2-1 Block Diagram of External Interrupt Circuit 1



❍ Edge detection circuit

When the edge polarity of the signal that is input to external interrupt circuit 1 pins (INT10 to INT13) matches the edge polarity selected by the EIC1 or EIC2 register (specified by the SEL0 to SEL7 bits), the corresponding external interrupt request flag bit (EIC1:EIR0, EIR1/EIC2:EIR2, EIR3) is set to "1".

❍ External interrupt control registers (EIC1, EIC2)

The EIC1 to EIC2 registers are used to select edges, enable and disable interrupt requests, and check for interrupt requests.

■ Interrupt sources of external interrupt circuit 1

IRQ0:

This interrupt request is generated if the edge of the selected polarity is input to external interrupt pin INT0/INT1 when the interrupt request output is enabled.

IRQ1:

This interrupt request is generated if the edge of the selected polarity is input to external interrupt pin INT2/INT3 when the interrupt request output is enabled.

13.3 Pins of the External Interrupt Circuit 1

This section describes the pins and provides a block diagram of the pins.

■ Pins for External Interrupt Circuit 1

External interrupt circuit 1 has four external interrupt pins.

These pins function as an external interrupt input (hysteresis input) or general-purpose input/ output port 6.

Pins P60/INT10 to P63/INT13/X0A continuously function as external interrupt inputs. If, however, an interrupt request output is disabled, no interrupt is output. Pin states can always be read directly from the port data register (PDR6).

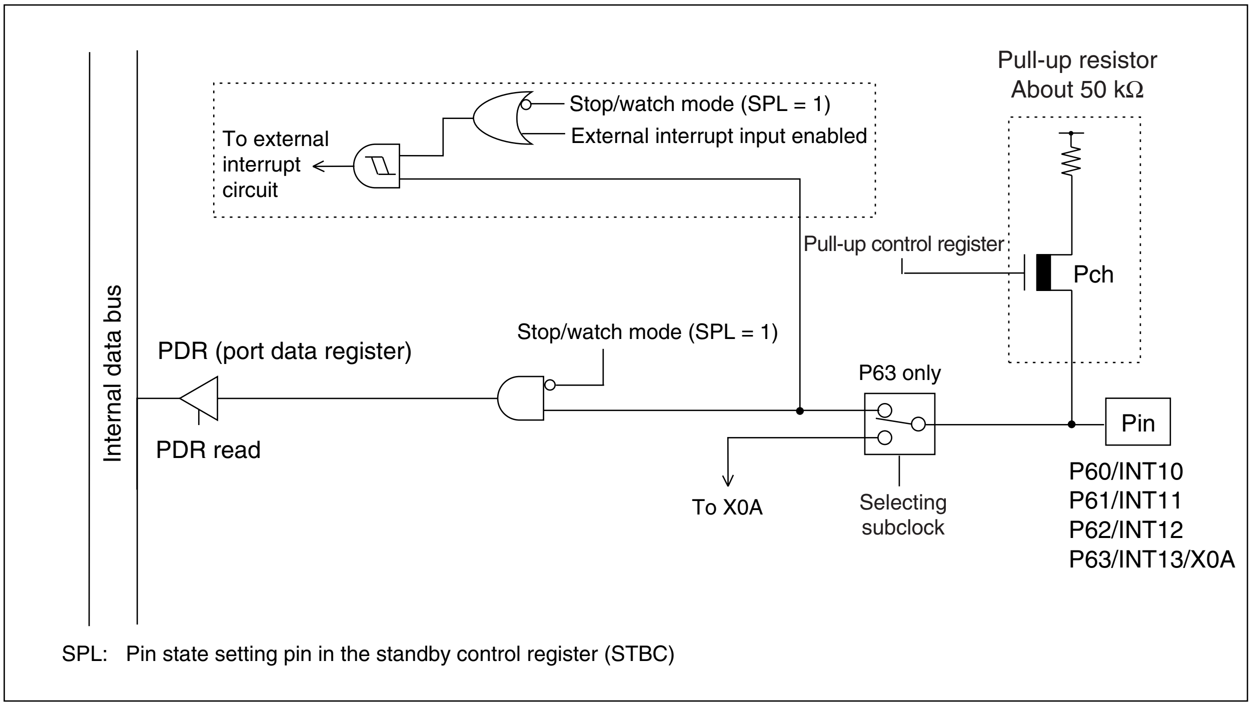
Table 13.3-1 Pins for External Interrupt Circuit 1

External interrupt pin	Used as an external interrupt input (when interrupt request output is enabled)	Used as a general-purpose input/output port (when interrupt request output is disabled)
P60/INT10	INT10 (EIC1:EIE0=1)	P60 (EIC1:EIE0=0)
P61/INT11	INT11 (EIC1:EIE1=1)	P61 (EIC1:EIE1=0)
P62/INT12	INT12 (EIC2:EIE2=1)	P62 (EIC2:EIE2=0)
P63/INT13/X0A	INT13 (EIC2:EIE3=1)	P63 (EIC2:EIE3=0)

INT10 to INT13: These pins generate the corresponding interrupt when an edge with the selected polarity is input to these pins.

■ Block Diagram of Pins for External Interrupt Circuit 1

Figure 13.3-1 Block Diagram of Pins for External Interrupt Circuit 1



SPL: Pin state setting pin in the standby control register (STBC)

Reference:

If "pull-up resistor available" is selected in the pull-up control resistor, the pins are set to the "H" level (pull-up state), not the high impedance state, in stop/watch mode (SPL bit of STBC = 1). However, the pull-up is disabled during a reset and the pins are set to the Hi-z state.

Note:

If "rising edge", "falling edge", or "both edges" is selected with the edge polarity selection bit in stop mode (STBC: SPL = 1), input interrupts continue to be input and are not blocked. Fix the pin voltage via the pull-up option setting register, external pull-up resistor, or external pull-down resistor.

13.4 Registers of the External Interrupt Circuit 1

This section describes the registers for external interrupt circuit 1.

■ External Interrupt Circuit 1 Registers

Figure 13.4-1 Registers for External Interrupts

EIC1 (External interrupt control register 1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 F H	EIR1	SEL3	SEL2	EIE1	EIR0	SEL1	SEL0	EIE0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	IRQ0/INT11				IRQ0/INT10				

EIC2 (External interrupt control register 2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 4 0 H	EIR3	SEL7	SEL6	EIE3	EIR2	SEL5	SEL4	EIE2	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	IRQ1/INT13				IRQ1/INT12				

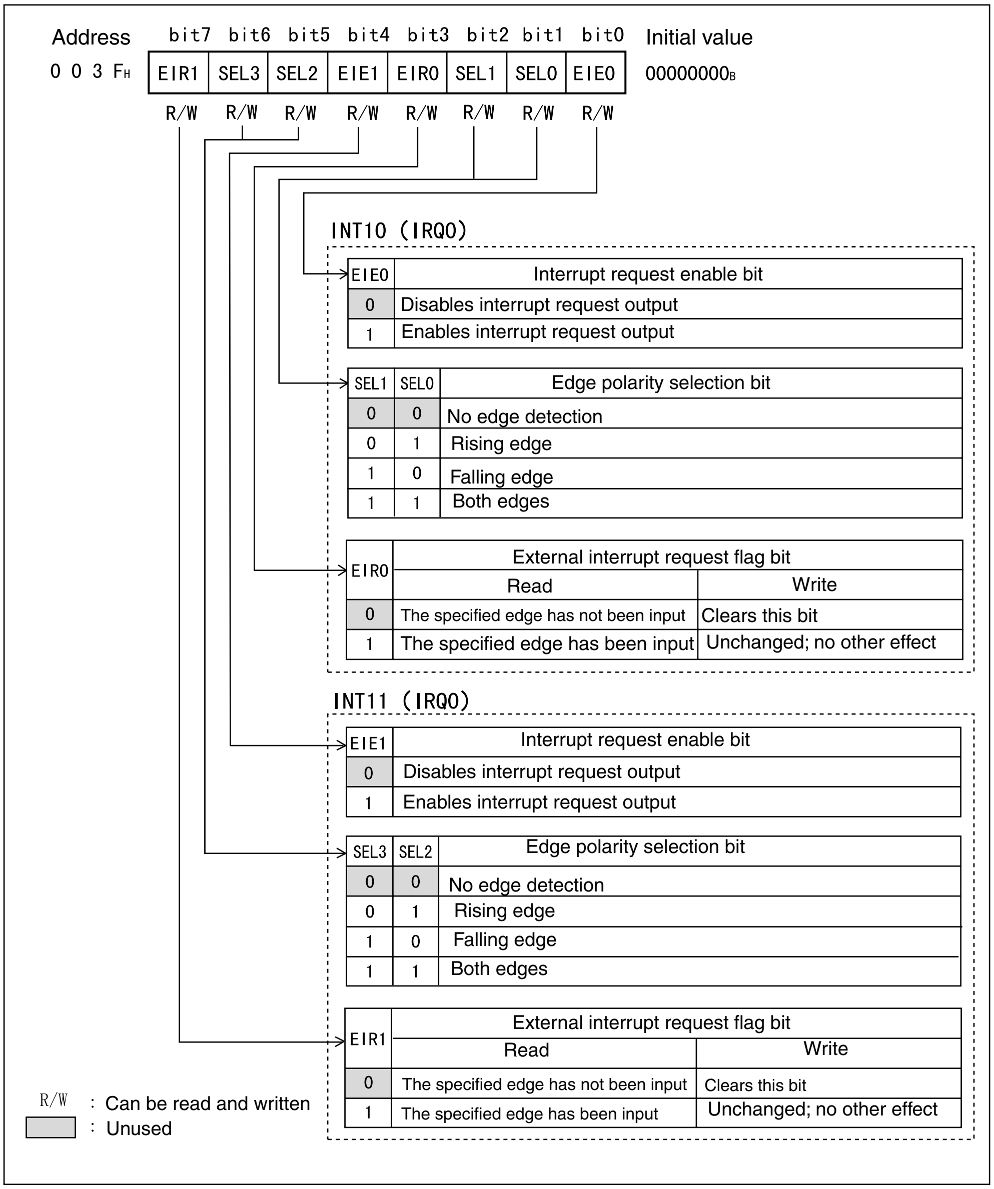
R/W : Can be read and written
X : Undefined

13.4.1 External Interrupt Control Register 1 (EIC1)

External interrupt control register 1 (EIC1) is used to select the edge polarity and control interrupts for external interrupt pins INT10 and INT11.

■ External Interrupt Control Register 1 (EIC1)

Figure 13.4-2 External Interrupt Control Register 1 (EIC1)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 F_H	EIR1	SEL3	SEL2	EIE1	EIR0	SEL1	SEL0	EIE0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

INT10 (IRQ0)

EIE0	Interrupt request enable bit
0	Disables interrupt request output
1	Enables interrupt request output

SEL1	SEL0	Edge polarity selection bit
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

EIR0	External interrupt request flag bit	
	Read	Write
0	The specified edge has not been input	Clears this bit
1	The specified edge has been input	Unchanged; no other effect

INT11 (IRQ0)

EIE1	Interrupt request enable bit
0	Disables interrupt request output
1	Enables interrupt request output

SEL3	SEL2	Edge polarity selection bit
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

EIR1	External interrupt request flag bit	
	Read	Write
0	The specified edge has not been input	Clears this bit
1	The specified edge has been input	Unchanged; no other effect

R/W : Can be read and written
▭ : Unused

Table 13.4-1 Functions of the External Interrupt Control Register 1 (EIC1) Bits

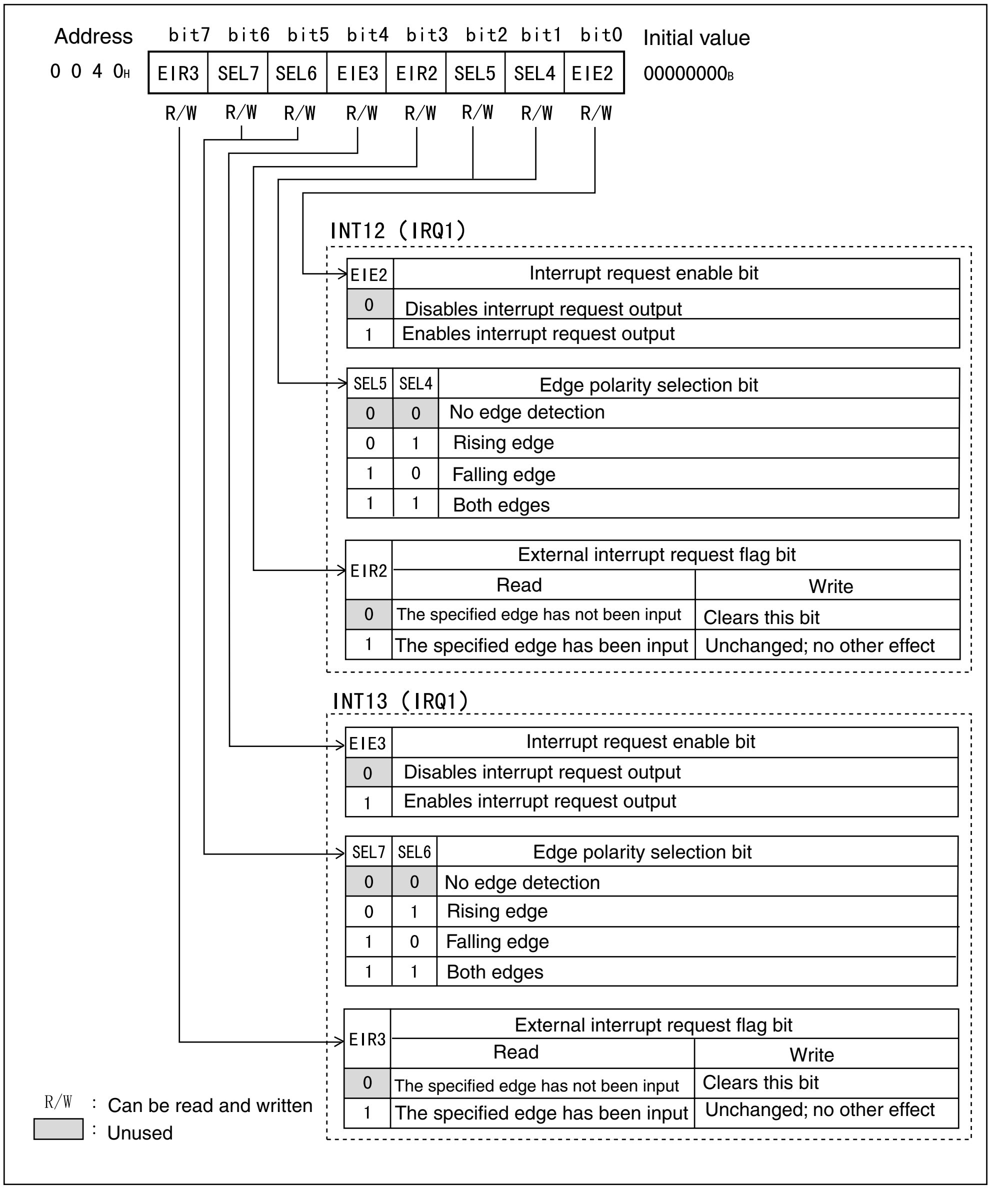
Bit		Description
bit7	EIR1: INT11 external interrupt request flag bit	• This bit is set to "1" when the edge selected by the INT11 edge polarity selection bits (SEL3 and SEL2 bits of EIC1) is input to external interrupt pin INT11. • When this bit and the INT11 interrupt request enable bit (EIE1 bit of EIC1) are "1", an interrupt request is output. • Writing "0" clears this bit. Writing "1" does not change the bit value (has no effect on operation).
bit6 bit5	SEL3, SEL2: INT11 edge polarity selection bits	• These bits are used to select the polarity of the edge that becomes the interrupt source of the pulse input to external interrupt pin INT11. • When "00" is written to these bits, edge detection is not used. When "01_B" is written, rising edge mode is used. • When "10" is written, falling edge mode is used. When "11_B" is written, both rising and falling modes are used. • To change these bits, always write "0" to EIR0.
bit4	EIE1: INT11 interrupt request enable bit	• This bit is used to enable or disable interrupt request output to the CPU. When this bit and the INT11 external interrupt request flag bit (EIR1) are "1", an interrupt request is output.
bit3	EIR0: INT10 external interrupt request flag bit	• This bit is set to "1" when the edge selected by the INT10 edge polarity selection bits (SEL1 and SEL0 bits of EIC1) is input to external interrupt pin INT10. • When this bit and the INT10 interrupt request enable bit (EIE0 bit of EIC1) are "1", an interrupt request is output. • Writing "0" clears this bit. Writing "1" does not change the bit value (has no effect on operation).
bit2 bit1	SEL1, SEL0: INT10 edge polarity selection bits	• These bits are used to select the polarity of the edge that becomes the interrupt source of the pulse input to external interrupt pin INT10. • When "00" is written to these bits, edge detection is not used. When "01_B" is written, rising edge mode is used. • When "10" is written, falling edge mode is used. When "11_B" is written, both rising and falling modes are used. • To change these bits, always write "0" to EIR0.
bit0	EIE0: INT10 interrupt request enable bit	• This bit is used to enable or disable interrupt request output to the CPU. When this bit and the INT10 external interrupt request flag bit (EIR0) are set to "1", an interrupt request is output.

13.4.2 External Interrupt Control Register 2 (EIC2)

External interrupt control register 2 (EIC2) is used to select the edge polarity and control interrupts for external interrupt pins INT12 and INT13.

■ External Interrupt Control Register 2 (EIC2)

Figure 13.4-3 External Interrupt Control Register 2 (EIC2)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 4 0H	EIR3	SEL7	SEL6	EIE3	EIR2	SEL5	SEL4	EIE2	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

INT12 (IRQ1)

EIE2	Interrupt request enable bit
0	Disables interrupt request output
1	Enables interrupt request output

SEL5	SEL4	Edge polarity selection bit
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

EIR2	External interrupt request flag bit	
	Read	Write
0	The specified edge has not been input	Clears this bit
1	The specified edge has been input	Unchanged; no other effect

INT13 (IRQ1)

EIE3	Interrupt request enable bit
0	Disables interrupt request output
1	Enables interrupt request output

SEL7	SEL6	Edge polarity selection bit
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

EIR3	External interrupt request flag bit	
	Read	Write
0	The specified edge has not been input	Clears this bit
1	The specified edge has been input	Unchanged; no other effect

R/W : Can be read and written
▨ : Unused

Table 13.4-2 Functions of the External Interrupt Control Register 2 (EIC2) Bits

Bit		Description
bit7	EIR3: INT13 external interrupt request flag bit	• This bit is set to "1" when the edge selected by the INT13 edge polarity selection bits (SEL7 and SEL6 bits of EIC2) is input to external interrupt pin INT13. • When this bit and the INT13 interrupt request enable bit (EIE3 bit of EIC2) are set to "1", an interrupt request is output. • Writing "0" clears this bit. Writing "1" does not change the bit value (has no effect on the operation).
bit6 bit5	SEL7, SEL6: INT13 edge polarity selection bits	• These bits are used to control the input edge polarity mode of the INT13 pin. • When "00" is written to these bits, edge detection is not used. When "01_B" is written, rising edge mode is used. • When "10" is written, falling edge mode is used. When "11_B" is written, both rising and falling modes are used. • To change these bits, always write "0" to EIR3.
bit4	EIE3: INT13 interrupt request enable bit	• This bit is used to enable or disable interrupt request output to the CPU. When this bit and the INT13 external interrupt request flag bit (EIR3) are "1", an interrupt request is output.
bit3	EIR2: INT12 external interrupt request flag bit	• This bit is set to "1" when the edge selected by the edge polarity selection bits (SEL5 and SEL4 bits of EIC2) is input to external interrupt pin INT12. • When this bit and the INT12 interrupt request enable bit (EIE2 bit of EIC2) are "1", an interrupt request is output. • Writing "0" clears this bit. Writing "1" does not change the bit value (has no effect on operation).
bit2 bit1	SEL5, SEL4: INT12 edge polarity selection bits	• These bits are used to control the input edge polarity mode of the INT12 pin. • When "00" is written to these bits, edge detection is not used. When "01_B" is written, rising edge mode is used. • When "10" is written, falling edge mode is used. When "11_B" is written, both rising and falling modes are used. • To change these bits, always write "0" to EIR2.
bit0	EIE2: INT12 interrupt request enable bit	• This bit is used to enable or disable interrupt request output to the CPU. When this bit and the INT12 external interrupt request flag bit (EIR2) are "1", an interrupt request is output.

13.5 External Interrupt Circuit 1 Interrupts

External interrupt circuit 1 triggers an interrupt when it detects the specified edge in signals that are input to the external interrupt pins.

■ Interrupts during External Interrupt Circuit 1 Operation

When the specified edge of external interrupt input is detected, the corresponding external interrupt request flag bits (EIR0 to EIR3 bits of EIC1 and EIC2) are set to "1". If the corresponding interrupt request enable bits are enabled (EIE0 to EIE3 bits of EIC1 and EIC2 = 1) at this time, an interrupt request (IRQ0 and IRQ1) is output to the CPU. The external interrupt request flag can be cleared by writing "0".

Note:

To enable interrupts (EIE0 and EIE1 bits of EIC1 = 1, EIE2 and EIE3 bits of EIC2 = 1) after releasing a reset, be sure to clear the external interrupt request flag bits (EIE0 and EIE1 bits of EIC1 = 1, EIE2 and EIE3 bits of EIC2 = 1) at a time.

If the external interrupt request flag bit is "1" and the interrupt request enable bit is enabled, the system cannot return from interrupt processing. Always clear the external interrupt request flag bit as part of the interrupt processing routine.

Reference:

Waking up the device from stop mode with an interrupt can be performed by using only external interrupt circuits.

An interrupt request is generated immediately after the interrupt request enable bit is switched from disabled to enabled (0-->1) when the external interrupt request flag bit is "1".

■ Registers and Vector Tables for External Interrupt Circuit 1 Interrupts

Table 13.5-1 Registers and Vector Tables for External Interrupt Circuit 1 Interrupts

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Setting bit		Upper	Lower
IRQ0	ILR1 (007B$_H$)	L01 (bit1)	L00 (bit0)	FFFA$_H$	FFFB$_H$
IRQ1		L11 (bit3)	L10 (bit2)	FFF8$_H$	FFF9$_H$

See 3.4.2 "Interrupt Processing", for more information about the operation during interrupts.

■ Caution When Changing the Edge Polarity Selection

When the INT10 to INT13 edge polarity is changed, always set the corresponding EIR bit to "0". This prevents generation of interrupts by mistake.

13.6 Operation of the External Interrupt Circuit 1

External interrupt circuit 1 can be used to detect the specified edges of signals that are input to the external interrupt pins. This section describes the operation using INT10 as an example.

■ Operation of External Interrupt Circuit 1

To operate INT0 of external interrupt circuit 1, external interrupt circuit 1 must be set as shown in Figure 13.6-1 "Settings of External Interrupt Circuit 1."

Figure 13.6-1 Settings of External Interrupt Circuit 1

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EIC1	EIR1	SEL3	SEL2	EIE1	EIR0	SEL1	SEL0	EIE0
	◎	◎	◎	◎	×	×	×	×

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EIC2	EIR3	SEL7	SEL6	EIE3	EIR2	SEL5	SEL4	EIE2
	×	×	×	×	×	×	×	×

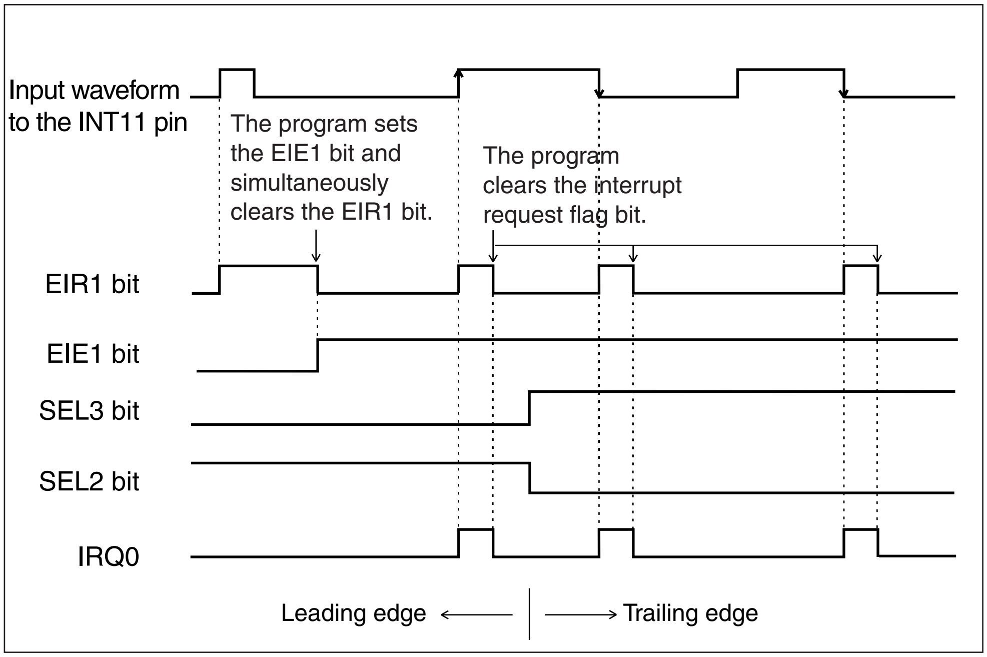
◎ : Used bit
× : Unused bit

If the edge polarity of the signal input from the external interrupt pin (INT11) matches the edge polarity (SEL2 to SEL3) selected by the external interrupt control register, the corresponding external interrupt request flag bit (EIR1) is set to "1".

Without referencing the value of the interrupt request enable bit (EIE1), the external interrupt request bit is set to "1" when the edge polarities match.

Figure 13.6-2 "External Interrupt 1 (INT11) Operation" shows the operation when the INT11 pin is used as an external interrupt pin.

Figure 13.6-2 External Interrupt 1 (INT11) Operation


Input waveform to the INT11 pin
The program sets the EIE1 bit and simultaneously clears the EIR1 bit.
The program clears the interrupt request flag bit.
EIR1 bit
EIE1 bit
SEL3 bit
SEL2 bit
IRQ0
Leading edge
Trailing edge

Reference:

The pin state can be read directly from the port data register (PDR6) even when the pin is used as an external interrupt input.

13.7 Program Example of the External Interrupt Circuit 1

This section shows a program example for external interrupt circuit 1.

■ Program Example for External Interrupt Circuit 1

❍ Processing specifications

Generates an interrupt upon detection of the rising edge of a pulse that is input to the INT11 pin.

❍ Coding example

```
EIC1   EQU      003FH            ;External interrupt control register 1

EIR1   EQU      EIC1:7           ;Definition of the external interrupt request
                                  flag bit
SEL2   EQU      EIC1:5           ;Definition of the edge polarity selection bit
EIE1   EQU      EIC1:4           ;Definition of the interrupt request enable bit

ILR0   EQU      007BH            ;Interrupt level setting register 1 setting

INT_V  DSEG     ABS              ;[DATA SEGMENT]
ORG    0FFFAH
IRQ0   DW       WARI             ;Interrupt vector setting
INT_V  ENDS
;-----------Main program----------------------------------------------------------
       CSEG                      ;[CODE SEGMENT]
                                 ;Assume that the stack pointer (SP) and other
                                  registers are already initialized.
        :
       CLRI                      ;Interrupts disabled
       CLRB     EIR1             ;External interrupt request flag cleared
       MOV      ILR1,#11111101B  ;Interrupt level set to 1
       SETB     SEL2             ;rising edge selected
       SETB     EIE1             ;Interrupt request output enabled
       SETI                      ;Interrupts enabled
        :
;----------Interrupt processing routine-------------------------------------------
WARI   CLRB     EIR1             ;External interrupt request flag cleared
       PUSHW    A
       XCHW     A,T
       PUSHW    A
        :
       User processing
        :
       POPW     A
       XCHW     A,T
       POPW     A
       RETI
       ENDS
;---------------------------------------------------------------------------------
       END
```

CHAPTER 14 EXTERNAL INTERRUPT CIRCUIT 2 (LEVEL)

This chapter describes the functions and operation of external interrupt circuit 2 (level).

14.1 Overview of the External Interrupt Circuit 2 (Level)

External interrupt circuit 2 detects the level of the signals that are input from eight external interrupt pins and outputs an interrupt request to the CPU.

■ Functions of External Interrupt Circuit 2 (Level Detection)

External interrupt circuit 2 detects an "L" level signal that is input from the external interrupt pins and outputs an interrupt request to the CPU. This interrupt can be used to wake up the device from standby mode and restore the normal operating state (the main-RUN or sub-RUN state).

- External interrupt pins: 8 pins (P40/INT20/EC to P47/INT27/ADST)
- External interrupt source: Input of an "L" level signal to an external interrupt pin
- Interrupt control: The external interrupt 2 control register (EIE2) can be used to enable or disable external interrupts.
- Interrupt flag: The external interrupt request flag bit of the external interrupt 2 flag register (EIF2) indicates detection of an "L" level.
- Interrupt requests: An interrupt occurs when the external interrupt pin is "Low" (IRQ4).

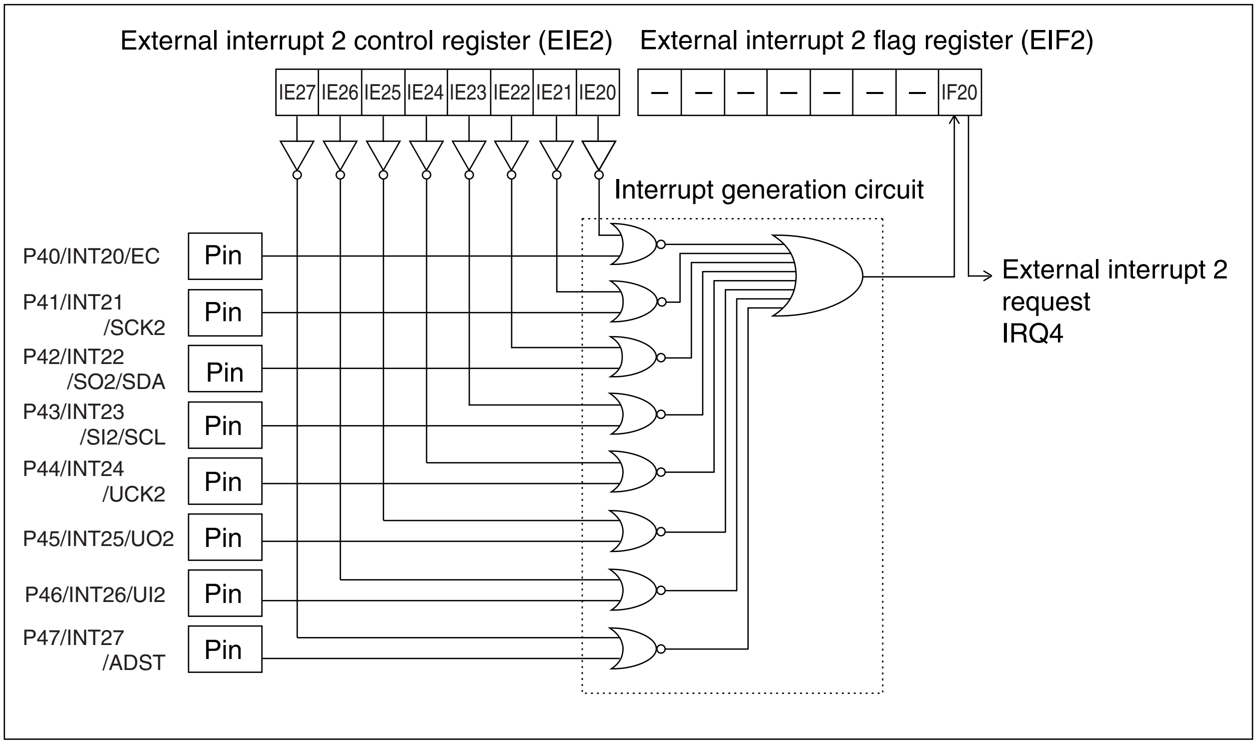
14.2 Configuration of the External Interrupt Circuit 2

External interrupt circuit 2 consists of the following three blocks:

- **Interrupt generation circuit**
- **External interrupt 2 control register (EIE2)**
- **External interrupt 2 flag register (EIF2)**

■ Block Diagram of External Interrupt Circuit 2

Figure 14.2-1 Block Diagram of External Interrupt Circuit 2



❍ Interrupt generation circuit

The interrupt request generation circuit generates an interrupt request signal based on the input signals from the external interrupt pins (INT20 to INT27) and the external interrupt input enable bits.

❍ External interrupt 2 control register (EIE2)

The external interrupt input enable bits (EIE2: IE20 to IE27) enable or disable "L" level inputs from the corresponding external interrupt pins.

❍ External interrupt 2 flag register (EIF2)

The external interrupt flag bit (EIF2: IF20) is used to store or clear interrupt request signals.

❍ External interrupt circuit 2 interrupt sources

IRQ4:

IRQ4 generates an interrupt to the CPU when an "L" level signal is input to any of the external interrupt pins INT20 to INT27 (pins whose interrupt inputs are enabled).

14.3 Pins of the External Interrupt Circuit 2

This section describes the pins, registers, and interrupt sources for external interrupt circuit 2, and provides a block diagram of the pins.

■ Pins for External Interrupt Circuit 2

Eight external interrupt pins are used for external interrupt circuit 2.

These external interrupt pins function as external interrupt inputs (hysteresis inputs), general-purpose input/output ports, or resource pins.

The P40/INT20/EC to P47/INT27/ADST pins function as external interrupt input pins (INT20 to INT27) when the corresponding pin is set as an input port by the port 4 data direction register (DDR4) and the corresponding external interrupt input is enabled by the external interrupt 2 control register (EIE2). If pins are set as input ports, the pin states can always be read from the port data register (PDR4).

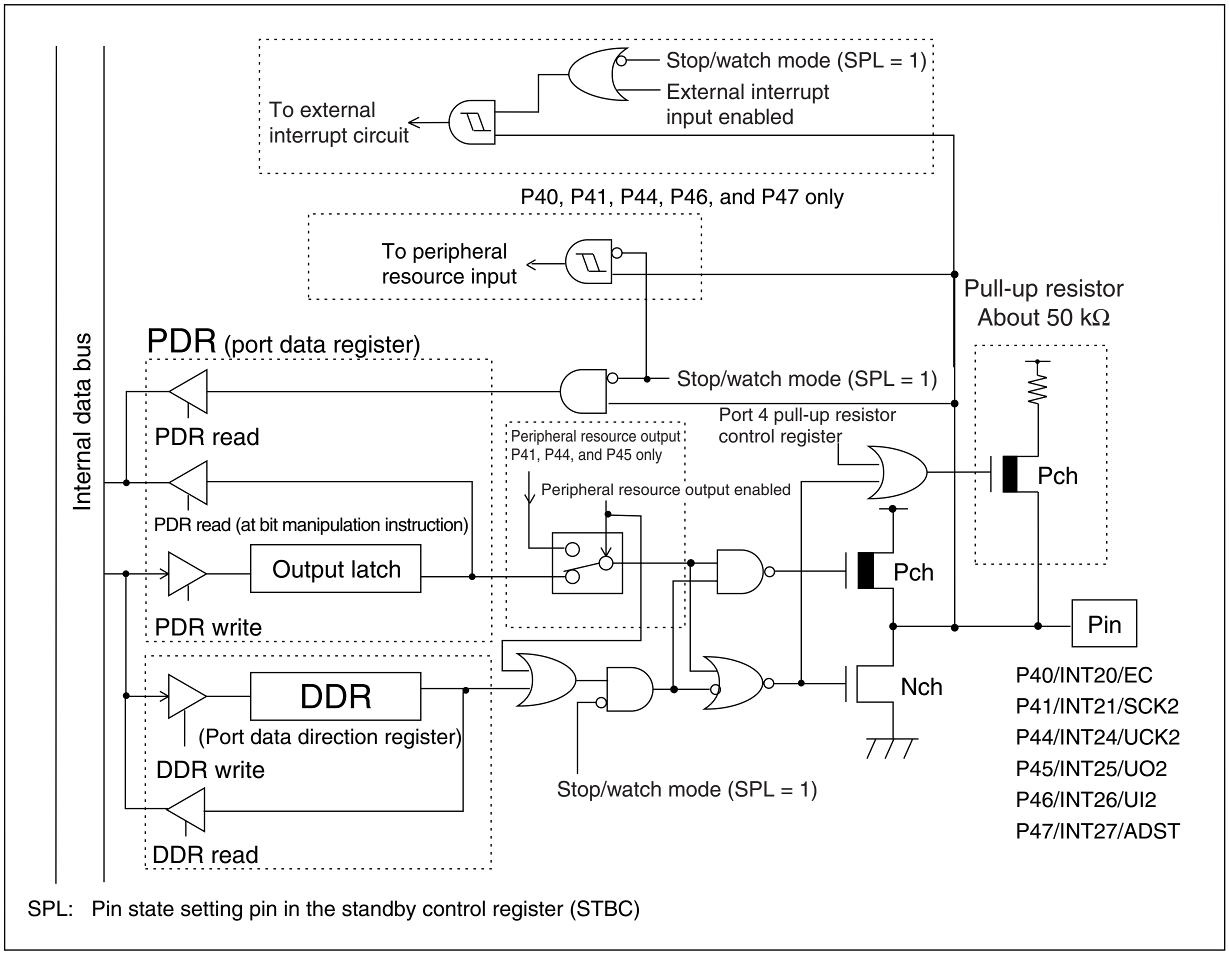
Table 14.3-1 "Pins for External Interrupt Circuit 2" shows the pins for external interrupt circuit 2.

Table 14.3-1 Pins for External Interrupt Circuit 2

External interrupt pin	Used as external interrupt input (interrupt input enabled)	Used as general-purpose input/ output port (interrupt input disabled)
P40/INT20/EC	INT20(EIE2:IE20=1)	P40(EIE2:IE20=0)
P41/INT21/SCK2	INT21(EIE2:IE21=1)	P41(EIE2:IE21=0)
P42/INT22/SO2/SDA	INT22(EIE2:IE22=1)	P42(EIE2:IE22=0)
P43/INT23/S12/SCL	INT23(EIE2:IE23=1)	P43(EIE2:IE23=0)
P44/INT24/UCK2	INT24(EIE2:IE24=1)	P44(EIE2:IE24=0)
P45/INT25/UO2	INT25(EIE2:IE25=1)	P45(EIE2:IE25=0)
P46/INT26/UI2	INT26(EIE2:IE26=1)	P46(EIE2:IE26=0)
P47/INT27/ADST	INT27(EIE2:IE27=1)	P47(EIE2:IE27=0)

■ Block Diagram of Pins for External Interrupt Circuit 2

Figure 14.3-1 Block Diagram of External Interrupt Circuit 2 Pins (INT22 and INT23 Only)

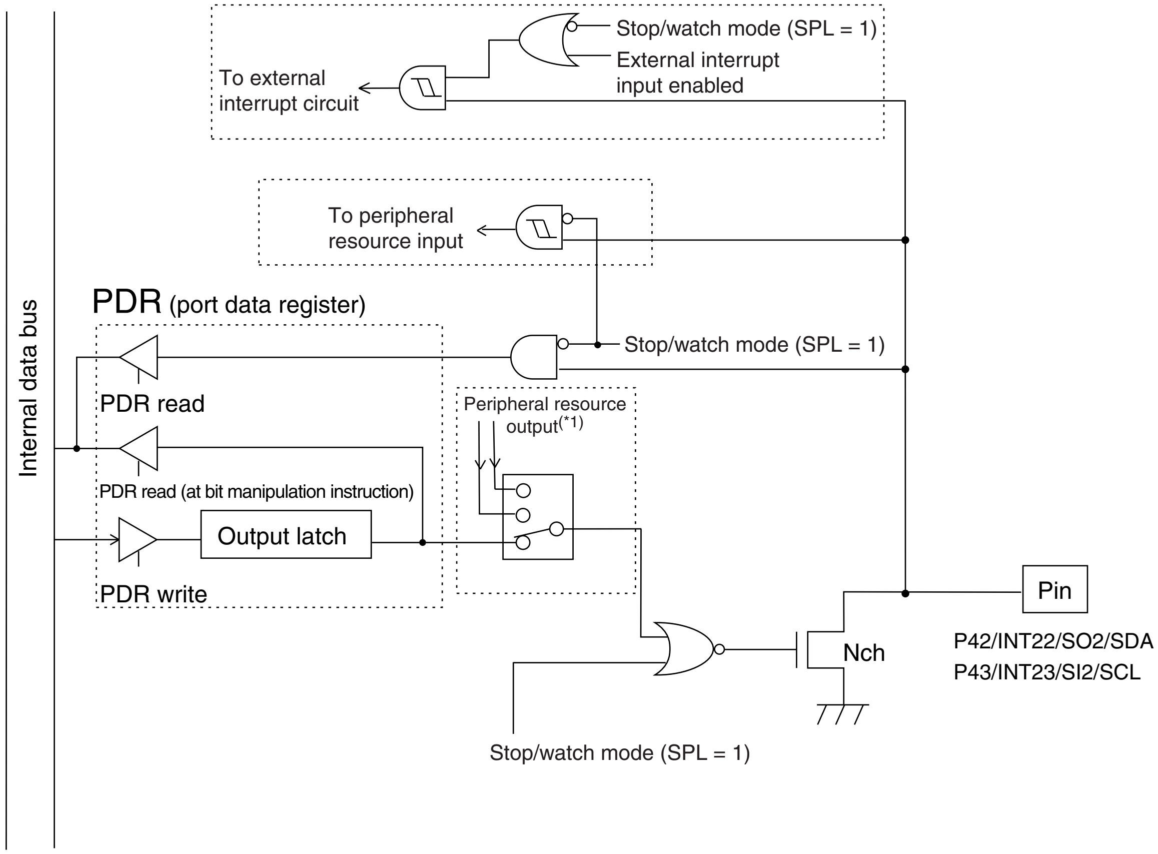

Internal data bus
To external interrupt circuit
Stop/watch mode (SPL = 1)
External interrupt input enabled
P40, P41, P44, P46, and P47 only
To peripheral resource input
Pull-up resistor About 50 kΩ
PDR (port data register)
Stop/watch mode (SPL = 1)
Port 4 pull-up resistor control register
PDR read
Peripheral resource output P41, P44, and P45 only
Peripheral resource output enabled
Pch
PDR read (at bit manipulation instruction)
Output latch
Pch
PDR write
Pin
DDR
Nch
(Port data direction register)
DDR write
Stop/watch mode (SPL = 1)
DDR read
P40/INT20/EC
P41/INT21/SCK2
P44/INT24/UCK2
P45/INT25/UO2
P46/INT26/UI2
P47/INT27/ADST

SPL: Pin state setting pin in the standby control register (STBC)

Reference:

If "pull-up resistor available" is selected in the port 4 pull-up resistor control register, the pins are set to the "H" level (pull-up state), not the high impedance state, in stop/watch mode (SPL bit of STBC = 1). However, the pull-up is disabled during a reset and the pins enter the Hi-z state.

Figure 14.3-2 Block Diagram of External Interrupt Circuit 2 Pins (INT22 and INT23 Only)


Stop/watch mode (SPL = 1)
External interrupt input enabled
To external interrupt circuit
To peripheral resource input
PDR (port data register)
Internal data bus
Stop/watch mode (SPL = 1)
PDR read
Peripheral resource output(*1)
PDR read (at bit manipulation instruction)
Output latch
PDR write
Pin
Nch
P42/INT22/SO2/SDA
P43/INT23/SI2/SCL
Stop/watch mode (SPL = 1)

SPL : Pin state setting pin in the standby control register (STBC)

*1 Two peripheral resources outputs are connected to P42.
One peripheral resource output is connected to P43.

Reference:

When a pin is set as an input pin (bits 2 and 3 of DDR4 = 0), the pin enters the high impedance state. If a high-level input state is initially necessary, an external pull-up resistor must be connected.

14.4 Registers of the External Interrupt Circuit 2

The external interrupt circuit 2 control register (EIE2) is used to enable or disable external interrupt pins.

■ Registers for External Interrupt Circuit 2

Figure 14.4-1 Registers for External Interrupt 2

EIE2 (External interrupt 2 control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 5 6H	IE27	IE26	IE25	IE24	IE23	IE22	IE21	IE20	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

EIF2 (External interrupt 2 flag register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 5 7H	—	—	—	—	—	—	—	IF20	-------0B
								R/W	

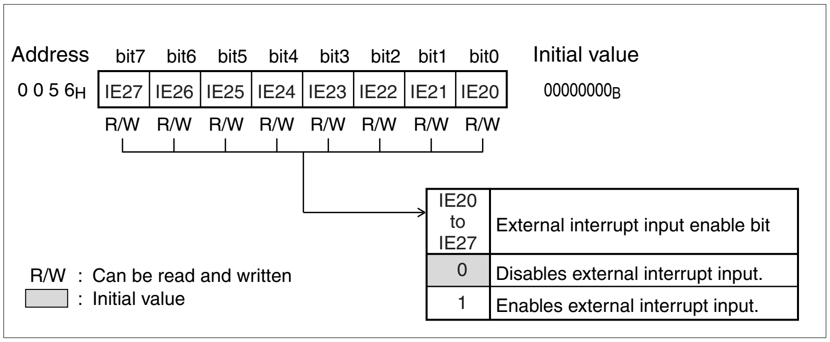
R/W : Can be read and written
— : Unused

14.4.1 External Interrupt 2 Control Register (EIE2)

The external interrupt 2 control register (EIE2) is used to enable or disable interrupt inputs to external interrupt pins INT20 to INT27.

■ External Interrupt 2 Control Register (EIE2)

Figure 14.4-2 External Interrupt 2 Control Register (EIE2)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 5 6H	IE27	IE26	IE25	IE24	IE23	IE22	IE21	IE20	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

IE20 to IE27	External interrupt input enable bit
0	Disables external interrupt input.
1	Enables external interrupt input.

R/W : Can be read and written
▨ : Initial value

Table 14.4-1 Correspondence between the Bits of the External Interrupt 2 Control Register (EIE2) and the External Interrupt Pins

Bit name		External interrupt pin
bit7	IE27	INT27
bit6	IE26	INT26
bit5	IE25	INT25
bit4	IE24	INT24
bit3	IE23	INT23
bit2	IE22	INT22
bit1	IE21	INT21
bit0	IE20	INT20

Table 14.4-2 Functions of the External Interrupt 2 Control Register (EIE2) Bits

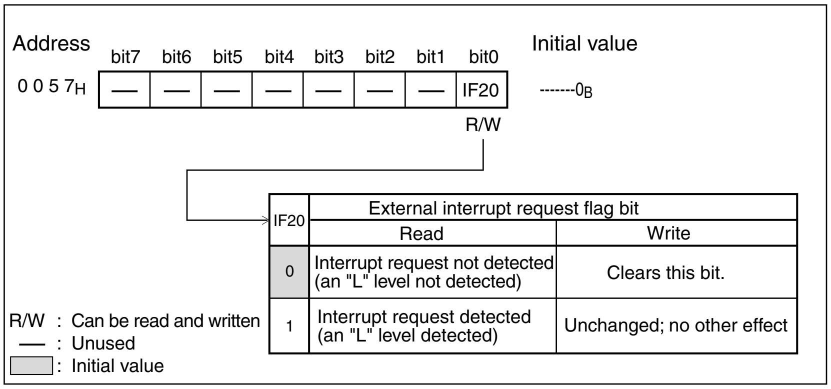
Bit name		Function
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0	IE20 to IE27: External interrupt input enable bits	• These bits enable or disable interrupt inputs from external interrupt pins INT20 to INT27. • Setting these bits to "1" sets the corresponding external interrupt pin to function as an external interrupt input pin and to accept external interrupt inputs. • Setting these bits to "0" sets the corresponding external interrupt pin to function as a general-purpose port and not to accept external interrupt inputs. **Reference:** • If external interrupt pins are used, set the corresponding bit of the port data direction register (DDR4) to "0" to specify that this pin becomes an input. • The state of external interrupt pins can be read directly from the port data register (PDR4) regardless of the state of the external interrupt input enable bits.

14.4.2 External Interrupt 2 Flag Register (EIF2)

The external interrupt 2 flag register (EIF2) is used to indicate detection of a level interrupt, to clear the interrupt request flag, and to store the interrupt state.

■ External Interrupt 2 Flag Register (EIF2)

Figure 14.4-3 External Interrupt 2 Flag Register (EIF2)



IF20	External interrupt request flag bit	
	Read	Write
0	Interrupt request not detected (an "L" level not detected)	Clears this bit.
1	Interrupt request detected (an "L" level detected)	Unchanged; no other effect

Table 14.4-3 Functions of the External Interrupt 2 Flag Register (EIF2) Bits

Bit name		Function
bit7 bit6 bit5 bit4 bit3 bit2 bit1	Unused	• During read operations, the values of these bits are undefined. • Writing these bits has no effect on operation.
bit0	IF20: External interrupt request flag bit	• This bit is set to "1" if an "L" level signal is input to the external interrupt pins (INT20 to INT27) for which external interrupt input is enabled. • Writing "0" clears this bit. Writing "1" does not change the bit and has no other effect. **Note:** Setting the external interrupt input enable bits (EIE2: IE20 to IE27) of the external interrupt 2 control register to "0" only disables external interrupt input. Interrupt requests are output continuously until the IF20 bit is cleared to "0".

14.5 External Interrupt Circuit 2 Interrupts

Input of an "L" level signal to an external interrupt pin triggers an interrupt from external interrupt circuit 2.

■ Interrupts during External Interrupt Circuit 2 Operation

Input of an "L" level to an external interrupt pin for which interrupt input has been enabled sets the external interrupt request flag bit (EIF2: IF20) to "1" and outputs an interrupt request (IRQ4) to the CPU. Write "0" to the IF20 bit in the interrupt handling routine to clear the interrupt request.

If the external interrupt request flag bit (EIF2) is set to "1", interrupt requests are output continuously until the IF20 bit is cleared to "0". Output of the interrupt request continues even if input of external interrupts is disabled by the interrupt enable bits (IE20 to IE27) of the external interrupt 2 control register (EIE2). Therefore, always clear the IF20 bit.

Also, if an external interrupt pin is still at the "L" level when the IF20 bit is cleared with no external interrupt input disabled, the IF20 bit is immediately set again. If necessary, disable external interrupt input or clear the external interrupt source itself.

Reference:

Always clear the IF20 bit before enabling CPU interrupts after a reset. The use of external interrupt circuits 1 and 2 enables return from standby mode.

Note:

An "L" level input to any of the external interrupt pins (INT20 to INT27) generates the same interrupt request (IRQ4). To determine which external interrupt input triggered the interrupt, you must read the port data register (PDR4) before the input changes to an "H" level.

Waking up the device from stop mode with in interrupt can be done using only external interrupt circuits 1 and 2.

■ Register and Vector Table for the External Interrupt Circuit 2 Interrupt

Table 14.5-1 Register and Vector Table for the External Interrupt Circuit 2 Interrupt

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Setting bit		Upper part of the address	Lower part of the address
IRQ4	ILR2 (007C$_H$)	L41 (bit1)	L40 (bit0)	FFF2$_H$	FFF3$_H$

See 3.4.2 "Interrupt Processing", for more information about the operation of interrupts.

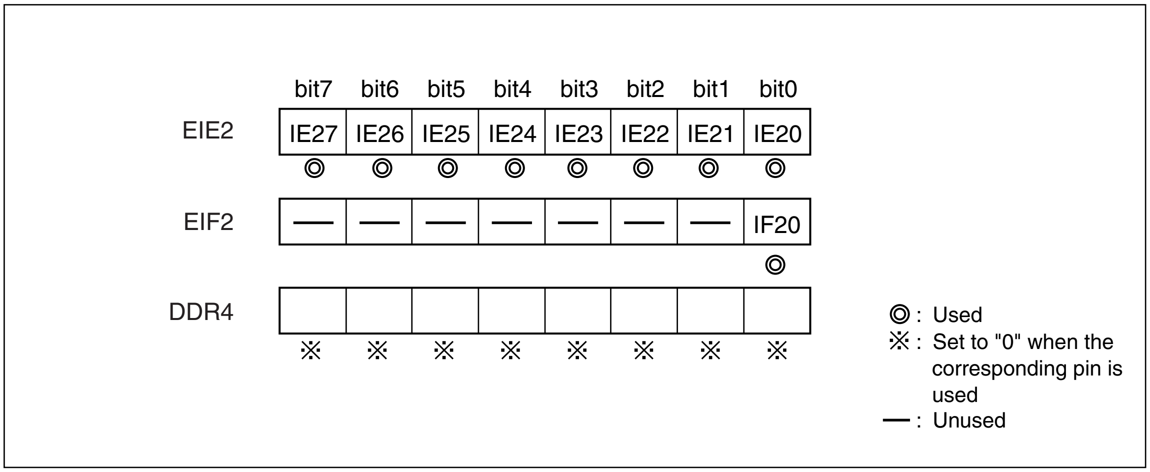
14.6 Operation of the External Interrupt Circuit 2

External interrupt circuit 2 can be used to detect an "L" level on the external interrupt pin and output an interrupt request to the CPU.

■ Operation of External Interrupt Circuit 2

Figure 14.6-1 "External Interrupt Circuit 2 Settings" shows the settings required for operation of the external interrupt circuit.

Figure 14.6-1 External Interrupt Circuit 2 Settings



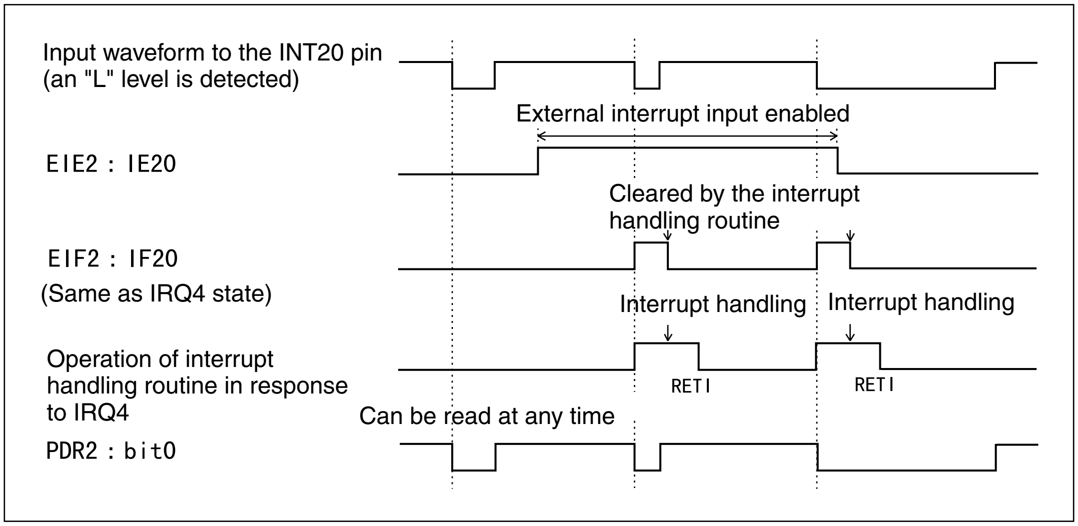
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EIE2	IE27	IE26	IE25	IE24	IE23	IE22	IE21	IE20
	◎	◎	◎	◎	◎	◎	◎	◎
EIF2	—	—	—	—	—	—	—	IF20
								◎
DDR4								
	※	※	※	※	※	※	※	※

◎ : Used
※ : Set to "0" when the corresponding pin is used
— : Unused

Input of an "L" level signal to the corresponding external interrupt pins of INT20 to INT27 for which any of the external interrupt inputs IE20 to IE23 has been enabled outputs an IRQ4 interrupt request to the CPU.

Figure 14.6-2 "Operation of External Interrupt 2 (INT20)" shows the operation of external interrupt circuit 2 (with the INT20 pin used).

Figure 14.6-2 Operation of External Interrupt 2 (INT20)


Input waveform to the INT20 pin
(an "L" level is detected)
External interrupt input enabled
EIE2 : IE20
Cleared by the interrupt handling routine
EIF2 : IF20
(Same as IRQ4 state)
Interrupt handling
Interrupt handling
Operation of interrupt handling routine in response to IRQ4
RETI
RETI
Can be read at any time
PDR2 : bit0

Reference:

The pin state can be read directly from the port data register (PDR4) even when the external interrupt pins are used as external interrupt inputs.

14.7 Program Example of the External Interrupt Circuit 2

This section shows a program example for external interrupt circuit 2.

■ Program Example for External Interrupt Circuit 2

❍ Processing specifications

Generates an interrupt upon detection of an "L" level signal that is input to the INT40 pin.

❍ Coding example

```
DDR4    EQU     0011H             ;Address of the port 4 direction register
EIE2    EQU     0056H             ;Address of the external interrupt 2 control
                                   register
EIF2    EQU     0057H             ;Address of the external interrupt 2 flag
                                   register

IF20    EQU     EIF2:0            ;Definition of the external interrupt request
                                   flag bit

ILR2    EQU     007CH             ;Address of the interrupt level setting register

INT_V   DSEG    ABS               ; [DATA SEGMENT]
ORG     0FFF2H
IRQ4    DW      WARI              ;Interrupt vector setting
INT_V   ENDS
;----------Main program-----------------------------------------------------
CSEG                              ; [CODE SEGMENT]
                                  ;Assume that the stack pointer (SP) and other
                                   registers are already initialized.
          :
        CLRI                      ;Interrupts disabled
        CLRB    IF20              ;External interrupt request flag cleared
        MOV     ILR2,#11111110B   ;Interrupt level set to 2
        MOV     DDR4,#00000000B   ;The P40/INT20 pin set to input
        MOV     EIE2,#00000001B   ;External interrupt input of the P40/INT20 pin
                                   is enabled
        SETI                      ;Interrupts enabled
          :
;----------Interrupt handling routine---------------------------------------
WARI    MOV     EIE2,#00000000B   ;External interrupt input of the INT20 pin
                                   is disabled
        CLRB    IF20              ;External interrupt request flag cleared
        PUSHW   A
        XCHW    A,T
        PUSHW   A
          :
        User processing
          :
        POPW    A
        XCHW    A,T
        POPW    A
        RETI
        ENDS
;---------------------------------------------------------------------------
        END
```

CHAPTER 15 A/D CONVERTER

This chapter describes the functions and operations of the A/D converter.

15.1 Overview of the A/D Converter

The 10-bit sequential comparison type A/D converter selects one input signal from the 8-channel analog input pins. It can be activated by software, internal clocks, or an input from the ADST pin.
However, the MB89F538/F538L cannot use an input (external clock) from the ADST pin.

■ A/D Conversion Function

The A/D conversion function converts the analog voltage (input voltage) input to an analog input pin to an 10-bit digital value.

- One signal can be selected from 8 analog input pins.
- Conversion speed is 60 instruction cycles (24 µs at 10 MHz main clock oscillation).
- Generates an interrupt when A/D conversion completes.
- Conversion completion can also be determined by software.

The following methods are available to activate A/D conversion:

- Software activation
- Continuous activation by a timebase timer output (divide-by-2^8 main clock oscillation)
- Continuous activation (ADST) in synchronization with an external clock

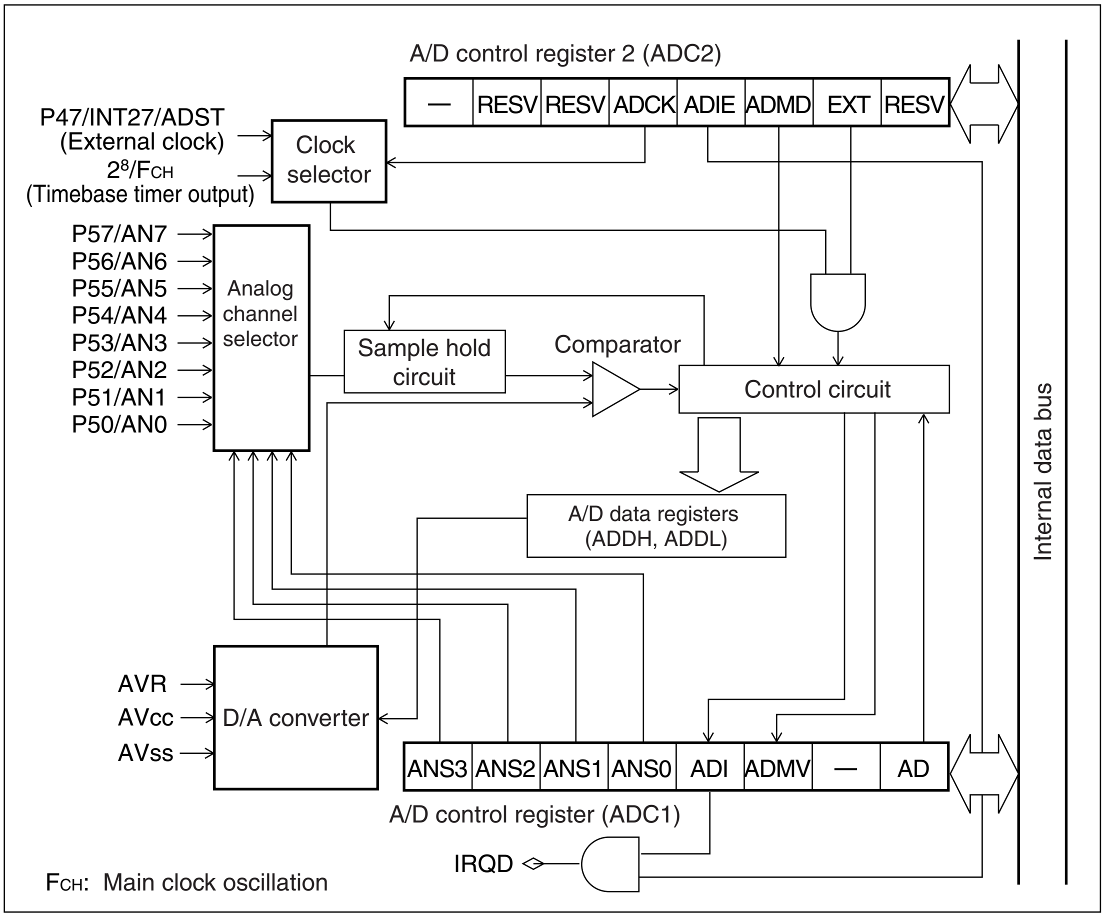
15.2 Configuration of the A/D Converter

The A/D converter consists of the following nine blocks:

- **Clock selector (selects the input clock for activating A/D conversion)**
- **Analog channel selector**
- **Sample and hold circuit**
- **D/A converter**
- **Comparator**
- **Control circuit**
- **A/D data registers (ADDH, ADDL)**
- **A/D control register 1 (ADC1)**
- **A/D control register 2 (ADC2)**

■ Block Diagram of the A/D Converter

Figure 15.2-1 A/D Converter Block Diagram



❍ **Clock selector**

Selects the clock used to activate A/D conversion when continuous activation is enabled (ADC2: EXT = "1").

❍ **Analog channel selector**

This circuit selects one signal from 8 analog input pins.

❍ **Sample and hold circuit**

This circuit holds the input voltage of the channel selected by the analog channel selector. The circuit samples and holds the input voltage immediately after the A/D conversion is activated. This allows A/D conversion to proceed without being affected by input voltage variation.

❍ **D/A converter**

Generates the voltage corresponding to the value set in the ADDH and ADDL registers.

❍ **Comparator**

Compares the sampled and held input voltage with the output voltage of the D/A converter, and determines which voltage is higher or lower.

❍ **Control circuit**

The control circuit successively determines the value of each bit of the 10-bit A/D data register, starting from the most significant bit and proceeding to the least significant bit, based on the greater-than/less-than signal from the comparator. When conversion is complete, the circuit sets the interrupt request flag bit (ADC1: ADI).

❍ **A/D data registers (ADDH/ADDL)**

The upper two bits of the 10-bit A/D data are stored in the ADDH register, and the lower eight bits are stored in the ADDL register.

The result of A/D conversion is stored in the ADDH/ADDL registers.

❍ **A/D control register 1 (ADC1)**

Enables and disables the functions of the A/D converter, selects an analog input pin, checks the state, and controls interrupt.

❍ **A/D control register 2 (ADC2)**

Selects the input clock, enables or disables interrupts, and selects the function.

❍ **AD converter interrupts**

IRQB:

When A/D conversion is complete, an interrupt request is issued if interrupt request output is enabled (ADC2: ADIE = 1).

■ A/D Converter Power Supply Voltage

❍ AVcc

Power supply pin for the A/D converter. Use the same potential that is used for Vcc. If extreme A/D conversion accuracy is required, make sure that AVcc is free of Vcc noise or use another power supply. If the A/D converter is not used, still connect this pin to the power supply.

❍ AVss

Ground pin for the A/D converter. Use same potential as that is used for Vss. If extreme A/D conversion accuracy is required, make sure that AVss is free of Vss noise or use another power supply. If the A/D converter is not used, still connect this pin to ground (GND).

❍ AVR

Pin to inputting the reference voltage for the A/D converter. 10-bit A/D conversion is performed between AVR and AVss. If the A/D converter is not used, connect it to AVss.

15.3 Pins of the A/D Converter

This section describes the pins related to the A/D converter and shows a block diagram.

■ Pins Related to A/D Converter

The pins related to the A/D converter are P47/INT27/ADST and P50/AN0 to P57/AN7.

❍ P47/INT27/ADST

The P47/INT27/ADST pin functions as a general-purpose input/output port (P47), the external clock input pin for A/D conversion activation (ADST), or an external interrupt pin. To enable the P47/INT27/ADST pin to function as the ADST pin, set this pin as an input port (bit 7 = 0), enable continuous activation (EXT bit of ADC2 = 1), and set the input clock selection bit as external clock input (ADCK bit of ADC2 = 1).

ADST:

When performing continuous activation in synchronization with external clock input, input an external clock to this pin.

❍ P50/AN0 to P57/AN7

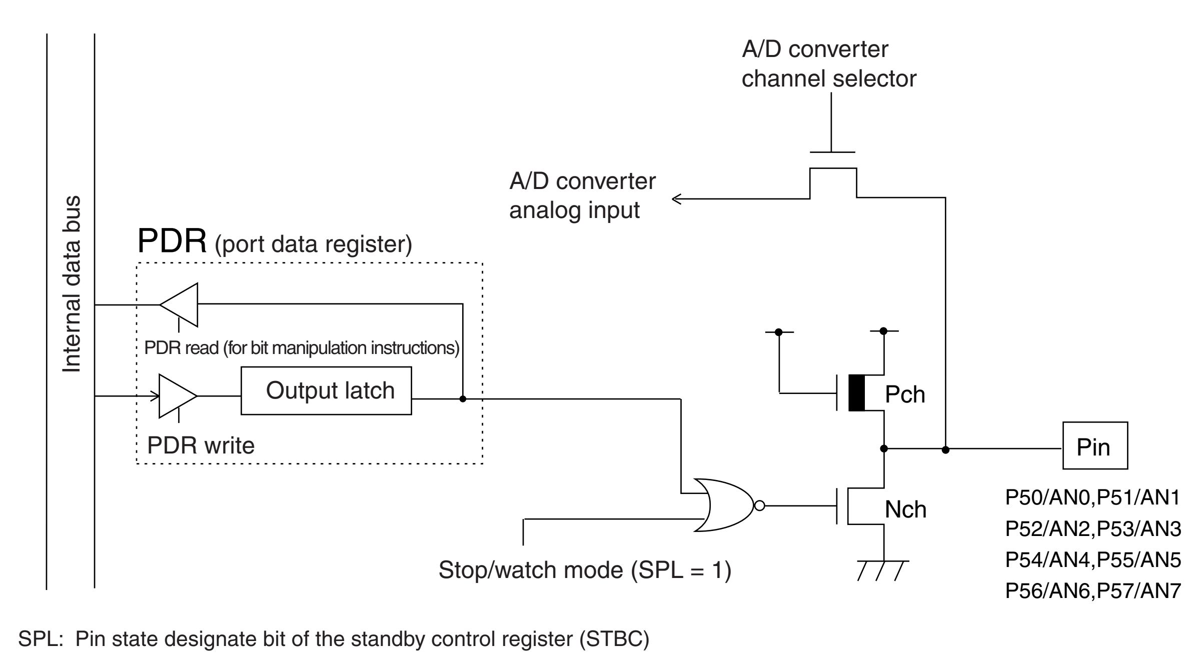
The P50/AN0 to P57/AN7 pins function as the output-only port for N-ch open-drain output (P50 to P57) or analog input pins (AN0 to AN7).

AN0 to AN7:

When using the A/D conversion function, input the analog voltage to be converted to these pins. To enable these pins to function as analog input pins, set "1" in the corresponding bits in the port data register (PDR5), turn off the output transistor, and then select these pins with the analog input channel selection bits (ANS0 to ANS3 of ADC1). Even if the A/D converter is used, those pins that are not used as analog inputs can be used as general-purpose input/output ports.

■ Block Diagram of P50/AN0 to P57/AN7 Pins

Figure 15.3-1 Block Diagram of Pins P50/AN0 to P57/AN7



15.4 Registers of the A/D Converter

This section shows the registers related to the A/D converter.

■ Registers Related to A/D Converter

Figure 15.4-1 Registers Related to the A/D Converter

ADC1 (A/D control register 1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 4H	—	ANS2	ANS1	ANS0	ADI	ADMV	—	AD	X00000X0B
		R/W	R/W	R/W	R/W	R		R/W	

ADC2 (A/D control register 2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 5H	—	RESV	RESV	ADCK	ADIE	ADMD	EXT	RESV	-0000001B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ADDL (A/D data register L)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 6H	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ADDH (A/D data register H)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 7H	—	—	—	—	—	—	D9	D8	------XXB
							R/W	R/W	

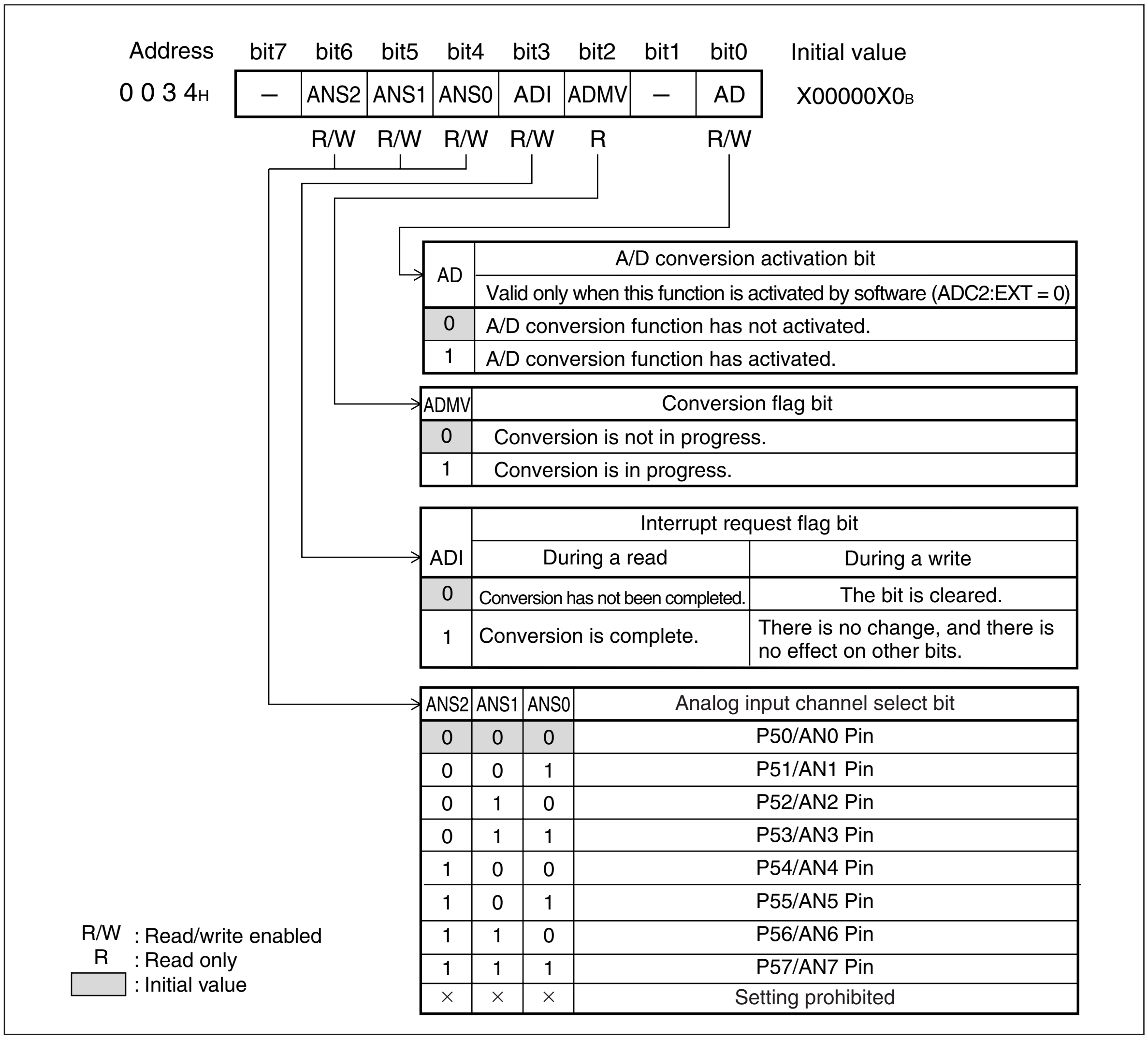
R/W	: Read/write enabled
R	: Read only
—	: Not used
X	: Undefined

15.4.1 A/D Control Register 1 (ADC1)

A/D control register 1 (ADC 1) enables and disables the functions of the A/D converter, selects analog input pins, and checks the operation state.

■ A/D Control Register 1 (ADC1)

Figure 15.4-2 A/D Control Register 1 (ADC1)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0034H	–	ANS2	ANS1	ANS0	ADI	ADMV	–	AD	X00000X0B
		R/W	R/W	R/W	R/W	R		R/W	

AD	A/D conversion activation bit Valid only when this function is activated by software (ADC2:EXT = 0)
0	A/D conversion function has not activated.
1	A/D conversion function has activated.

ADMV	Conversion flag bit
0	Conversion is not in progress.
1	Conversion is in progress.

ADI	Interrupt request flag bit	
	During a read	During a write
0	Conversion has not been completed.	The bit is cleared.
1	Conversion is complete.	There is no change, and there is no effect on other bits.

ANS2	ANS1	ANS0	Analog input channel select bit
0	0	0	P50/AN0 Pin
0	0	1	P51/AN1 Pin
0	1	0	P52/AN2 Pin
0	1	1	P53/AN3 Pin
1	0	0	P54/AN4 Pin
1	0	1	P55/AN5 Pin
1	1	0	P56/AN6 Pin
1	1	1	P57/AN7 Pin
×	×	×	Setting prohibited

R/W : Read/write enabled
R : Read only
▨ : Initial value

Table 15.4-1 Functions of the A/D Control Register 1 (ADC1) bits

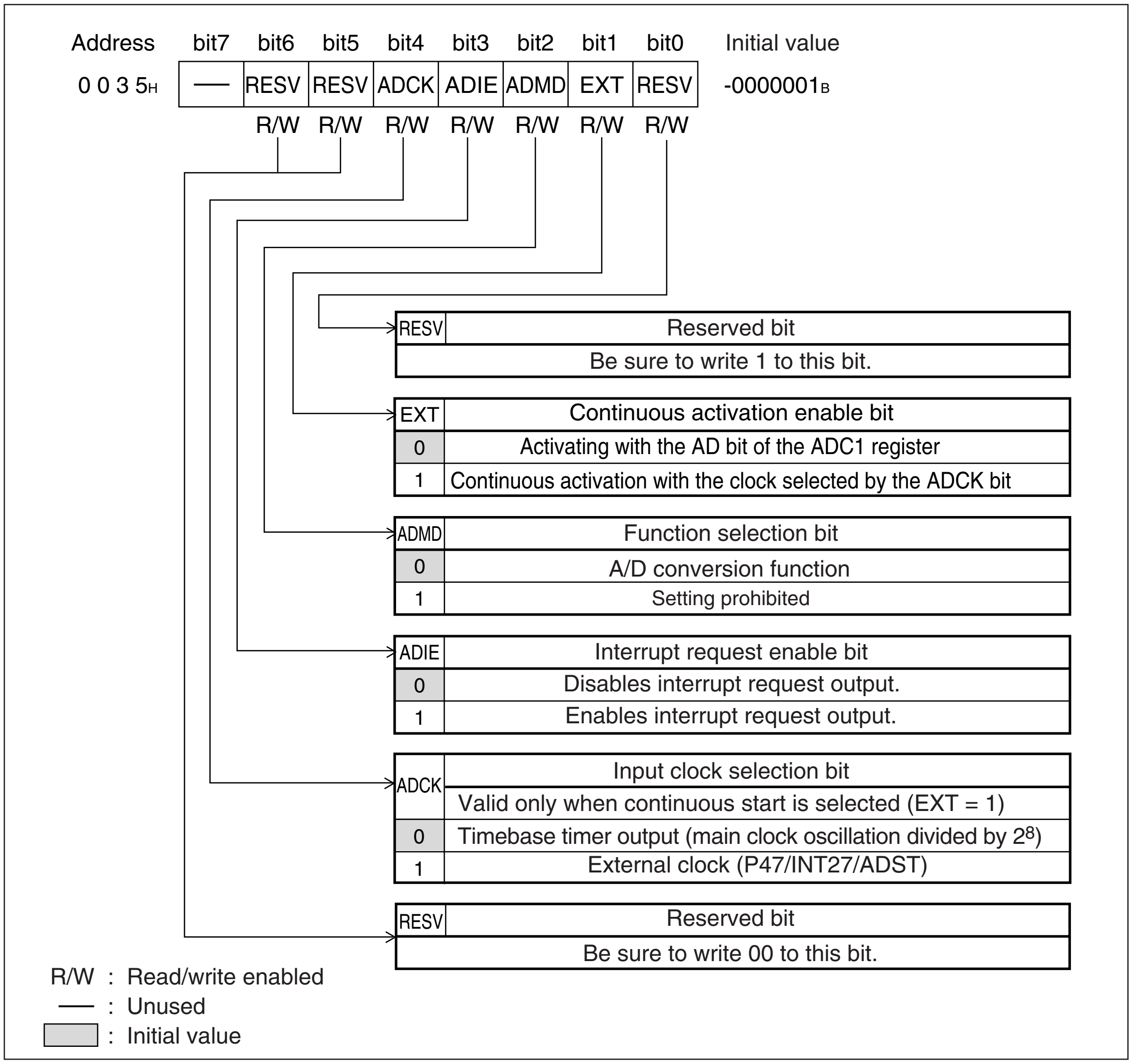
Bit name		Function
Bit 7 Bit 6 Bit 5 Bit 4	ANS2, ANS1, ANS0: Analog input channel select bits	• These bits select a pin to be used as an analog input pin from AN0 to AN7. • When this function is activated by software (ADC2:EXT = 0), these bits can be rewritten when A/D conversion is activated (AD = 1). **Note:** If the ADMV bit is "1", do not rewrite these bits. **Reference:** The pins that are not used as analog input pins can be used as general-purpose ports.
Bit 3	ADI: Interrupt request flag bit	• In A/D conversion function operation, this bit is set to 1 when A/D conversion is completed. • When this bit and the interrupt request enable bit (ADC2:ADIE) are 1, an interrupt request is issued. • For a write operation, writing 0 clears this bit. Writing 1 has no effect and no effects on other bits.
Bit 2	ADMV: Conversion flag bit	• In the A/D conversion function operation, this bit indicates that conversion is being performed. • During conversion, this bit is set to 1. **Reference:** This bit is read only. Writing to this bit has no meaning and has no effect on operation.
Bit 1	Unused bit	• The read value is undefined. • Writing has no effect on operation.
Bit 0	AD: A/D conversion activation bit	• This bit activates the A/D conversion function from software. • When continuous activation is disabled (ADC2:EXT = 0), setting this bit to 1 activates the A/D conversion function. **Note:** • Even though 0 is written to this bit, the operation of the A/D conversion function cannot be stopped. The read value is always 0. • During continuous activation, this bit is ignored.

15.4.2 A/D Control Register 2 (ADC2)

The A/D control register 2 (ADC 2) selects the functions of the A/D converter, selects the input clock, enables and disables interrupts and continuous activation, and checks the state.

■ A/D Control Register 2 (ADC2)

Figure 15.4-3 A/D Control Register 2 (ADC2)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 5H	—	RESV	RESV	ADCK	ADIE	ADMD	EXT	RESV	-0000001B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	

RESV	Reserved bit
	Be sure to write 1 to this bit.

EXT	Continuous activation enable bit
0	Activating with the AD bit of the ADC1 register
1	Continuous activation with the clock selected by the ADCK bit

ADMD	Function selection bit
0	A/D conversion function
1	Setting prohibited

ADIE	Interrupt request enable bit
0	Disables interrupt request output.
1	Enables interrupt request output.

ADCK	Input clock selection bit
	Valid only when continuous start is selected (EXT = 1)
0	Timebase timer output (main clock oscillation divided by 2^8)
1	External clock (P47/INT27/ADST)

RESV	Reserved bit
	Be sure to write 00 to this bit.

R/W : Read/write enabled
— : Unused
▨ : Initial value

Table 15.4-2 Functions the A/D Control Register 2 (ADC2) bits

Bit name		Function
Bit 7	Unused bit	• The read value is undefined. • Writing has no effect on operation.
Bit 6 Bit 5	TIM1, TIM0: Reserved bit	• Be sure to write 00 to this bit.
Bit 4	ADCK: Input clock selection bit	• When continuous activation is enabled (EXT = 1), this bit selects the input clock that activates the A/D conversion function. When this bit is "0", the timerbase timer (divide-by-2^8 output of main clock oscillation) is selected. When this bit is "1", the P47/INT27/ADST pin is selected. **Note:** If the main clock oscillation is stopped in subclock mode, the timerbase timer cannot be used to activate continuous mode conversion.
Bit 3	ADIE: Interrupt request enable bit	• This bit enables and disables interrupt output to the CPU. • When this bit and the interrupt request flag bit (ADC1: ADI) are 1, an interrupt request is output.
Bit 2	ADMD: Operation selection bit	• Reserved bit • Be sure to write 0 to this bit.
Bit 1	EXT: Continuous activation enable bit	• This bit determines whether the A/D conversion function is activated by software or whether it is continuously activated in sync with the input clock. • When this bit is set to 0, activating by software using the A/D conversion activation bit is enabled. When this bit is set to 1, continuous activation synchronized with the rising edge of the clock selected by the input clock selection bit (ADC2:ADCK) is enabled.
Bit 0	RESV: Reserved bit	**Note:** • Be sure to write 1 to this bit. • The read value is always 1.

15.4.3 A/D Data Registers (ADDH, ADDL)

These data registers store the result of A/D conversion.
The upper two bits of the 10-bit data correspond to the ADDH register, and the lower eight bits correspond to the ADDL register.

■ A/D Data Registers (ADDH, ADDL)

Figure 15.4-4 "A/D Data Registers (ADDH, ADDL)" shows the bit configuration of the A/D data register.

Figure 15.4-4 A/D Data Registers (ADDH, ADDL)

ADDH

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0037H	—	—	—	—	—	—	D9	D8	------XXB
							R/W	R/W	

ADDL

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0036H	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Read/write enabled
— : Unused
X : Undefined

The upper two bits of the 10-bit A/D data correspond to bits 1 and 0 of the ADDH register, and the lower eight bits correspond to bits 7 to 0 of the ADDL register.

❍ When A/D conversion function

When A/D conversion is activated, the conversion result data is determined after about 60 instruction cycles and is stored in these registers. Between the completion of A/D conversion and the start of the next A/D conversion cycle, read the contents of these registers (conversion result), write 0 to ADI (bit 3) of the ADC1 register, and clear the flag when conversion is complete.

15.5 A/D Converter Interrupt

The following function is provided as the A/D converter.

- **Completion of conversion in A/D conversion function operation**

■ Interrupt for A/D Conversion Function

When A/D conversion is completed, the interrupt request flag bit (ADC: ADI) is set to 1. If the interrupt request enable bit is then set to enable (ADC2: ADIE=1), an interrupt request to the CPU (IRQD) occurs. Clear the interrupt request by writing 0 to the ADI bit with the interrupt processing routine.

The ADI bit is set when A/D conversion is completed irrespective of the value of the ADIE bit.

Reference:

If the ADIE bit is changed from "disable" to "enable" (0 --> 1) when the ADI bit is set to 1, an interrupt request occurs immediately.

■ Register and Vector Table Related to A/D Converter Interrupt

Table 15.5-1 Register and Vector Table Related to the A/D Converter Interrupt

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Bit to be set		Higher	Lower
IRQD	ILR4 (007E$_H$)	LD1 (bit 3)	LD0 (bit 2)	FFE0$_H$	FFE1$_H$

For the operation of interrupts, see Section 3.4.2 "Interrupt Processing".

15.6 Operation of the A/D Converter

The A/D converter is activated by software.

■ Activating the A/D Conversion Function

❍ Activating software

Activating of the software for the A/D conversion function requires the setting shown in Figure 15.6-1 "Setting the A/D Conversion Function (When Software Activation)".

Figure 15.6-1 Setting the A/D Conversion Function (When Software Activation)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADC1	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	—	AD
	0	◎	◎	◎	◎	◎		1
ADC2	—	RESV	RESV	ADCK	ADIE	ADMD	EXT	RESV
		0	0	×	◎	0	0	1
ADDH	The comparison voltage value is held.							
ADDL	The comparison voltage value is held.							

— : Unused
◎ : Used bit
× : Unused bit
1 : Set 1
0 : Set 0

When A/D conversion is activated, the A/D conversion function starts operation. The A/D conversion function can be restarted even during conversion.

❍ Continuous activation

For continuous activating of the A/D conversion function, the settings shown in Figure 15.6-2 "Settings for the A/D Conversion Function (Continuous Activation)" are required.

Figure 15.6-2 Settings for the A/D Conversion Function (Continuous Activation)

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADC1	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	—	AD
	0	◎	◎	◎	◎	◎		×
ADC2	—	RESV	RESV	ADCK	ADIE	ADMD	EXT	RESV
		0	0	◎	◎	0	1	1
ADDH	The comparison voltage value is held.							
ADDL	The comparison voltage value is held.							

— : Unused
◎ : Used bit
× : Unused bit
1 : Set 1
0 : Set 0

When continuous activation is allowed, A/D conversion starts in sync with the rising edge of the selected input clock and the A/D conversion function starts operating. To disable continuous activation, set the EXT bit of ADC2 to 0. Activating by software is enabled.

■ Operation of A/D Conversion Function

A/D converter operation is described in the following. The time from activation to completion of the A/D conversion is about 60 instruction cycles.

1. When A/D conversion is activated, the conversion flag bit is set (ADC1: ADMV = 1) and the set analog input pin is connected to the sample hold circuit.
2. The voltage on the analog input pin is added to the internal sample hold capacitor for about 16 instruction cycles. This voltage is held until A/D conversion is complete.
3. The comparator compares the voltage added to the sample hold capacitor with the reference voltage for A/D conversion in order from the most significant bit (MSB) to the least significant bit (LSB), and each comparison result is transferred on at a time to the ADDH and ADDL registers.
4. When result transfer ends, the conversion flag bit is cleared (ADMV bit of ADC1 = 0) and the interrupt request flag bit is set (ADI bit of ADC1 = 1).

15.7 Notes on Using the A/D Converter

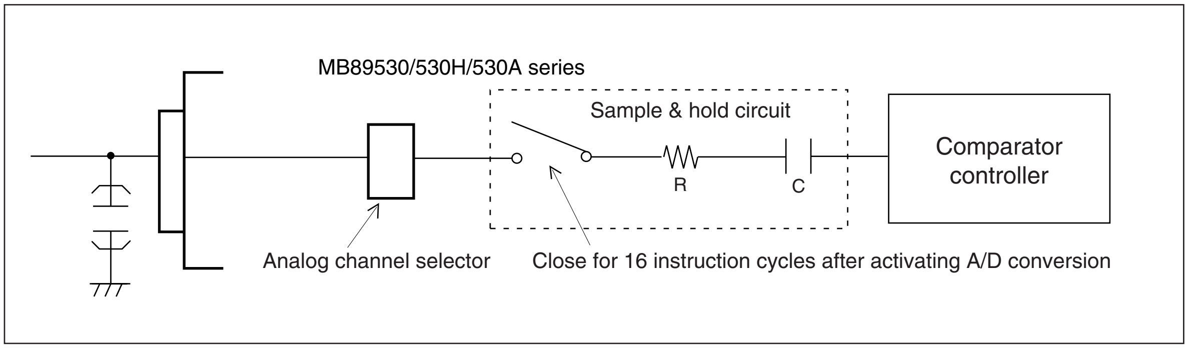
This section contains notes on using the A/D converter.

■ Notes on Using the A/D Converter

❍ Input impedance of the analog input pins

The A/D converter contains the sample hold circuit shown in Figure 15.7-1 "Equivalent Circuit for Analog Input" and adds the voltage of the analog input pin to the sample hold capacitor in about 15 instruction cycles after A/D conversion starts. Therefore, if the output impedance of the external circuit for analog input is high, the analog input voltage may not be stabilized within the analog input sampling period. To avoid this problem, keep the output impedance of the external circuit low enough to stabilize the voltage (less than 10 kΩ). If the output impedance of the external circuit cannot be kept low, it is recommended that an external capacitor of about 0.1 µF be added to the analog input pin.

Figure 15.7-1 Equivalent Circuit for Analog Input



❍ Notes on setting with a program

- In A/D conversion function operations, the ADDH and ADDL registers hold the previous values until A/D conversion is activated. As soon as A/D conversion is activated, the contents of the ADDH and ADDL registers become undefined.
- During operation of the A/D conversion function, do not reselect an analog input channel (ADC1: ANS2 to ANS0). In particular, while continuous activation is in operation, disable continuous activation (ADC2:EXT = 0) and wait for the flag bit (ADC1:ADMV) to change to 0 during conversion.
- The A/D converter is stopped via a reset and activation of the stop mode and watch mode.
- If the interrupt request flag bit (ADC1: ADI) is set to 1 and the interrupt request is enabled (ADC2: ADIE = 1), it is not possible to return from interrupt processing. Be sure to clear the ADI bit.
- For continuous activation by external clocks, input clocks according to the conversion/ comparison time and result read time.

❍ Notes on interrupt requests

If A/D conversion is reactivated (ADC1: AD = 1) and terminated at the same time, the interrupt request flag bit (ADC1: ADI) is not set.

❍ About errors

As |AVR-AVss| decreases, errors become relatively larger.

❍ Order of applying A/D converter power and analog input

Power on the A/D converter (AVcc, AVss) and apply an analog input (AN0 to AN7) at the same time as, or after turning on the digital power supply (Vcc).

To turn off the power, turn off the digital power supply (Vcc) concurrently with or after turning off the power supply of the A/D converter (AVcc, AVss) and the analog input (AN0 to AN7).

When the A/D converter is turned on or off, prevent an analog input from exceeding the voltage of the digital power supply.

❍ Conversion time

The conversion speed of the A/D conversion function is affected by the clock mode, main clock oscillation frequency, or speed switching of main clock (gear function).

❍ Input clock for continuous activation

Although the timebase timer output is not affected by the gear function, this output cannot be used because the oscillation of the main clock stops in subclock mode. In addition, when the timebase timer is cleared, the clock period (one cycle) is affected.

15.8 Program Example of the A/D Converter

This section contains a sample program for the 10-bit A/D converter.

■ Program Example of the A/D Conversion Function

❍ Processing specification

The analog voltage input to the AN0 pin is digitized by activating of the software.

An interrupt is not used, and the completion of conversion is detected by the loop in the program.

❍ Coding example

```
DDR5    EQU     0012H               ;Address of the Port 5 direction register

ADC1    EQU     0034H               ;Address of the A/D control register 1
ADC2    EQU     0035H               ;Address of the A/D control register 2
ADDH    EQU     0037H               ;Address of the A/D data register H
ADDL    EQU     0036H               ;Address of the A/D data register L

AN0     EQU     DDR5:0              ;Definition of the AN0 analog input pin
ADI     EQU     ADC1:3              ;Definition of the interrupt request flag bit
ADMV    EQU     ADC1:2              ;Definition of the conversion flag bit
AD      EQU     ADC1:0              ;Definition of the A/D conversion activation
                                     bit (software activation)
EXT     EQU     ADC2:1              ;Definition for the continuous activation
                                     enable bit
;----------Main program----------------------------------------------------------
        CSEG                        ; [CODE SEGMENT]
          :
        CLRI                        ;Interrupt disable
        SETB    AN0                 ;Specify the P00/AN0 pin as an analog input
                                     pin
        CLRB    EXT                 ;Continuous activation disabled
AD_WAIT
        BBS     ADMV,AD_WAIT        ;A/D converter stop check loop
        MOV     ADC1, #00000000B    ;Select analog input channel 0 (AN0),
                                     clearing the interrupt request flag, and
                                     set that software activating is not used
        MOV     ADC2, #00000001B    ;Disable interrupt request output, and select
                                     the A/D conversion function. Select
                                     activation by software.
        SETI                        ;Interrupt enable
        SETB    AD                  ;Activate the software
AD_CONV
        BBS     ADMV, AD_CONV       ;A/D conversion completion wait loop
                                     [Approx. 24µs (at 10 MHz)]
        CLRB    ADI                 ;clearing the interrupt request flag
        MOVW    A,ADDL              ;Read the A/D conversion data (lower 8 bits)
        MOV     A,ADDH              ;Read the A/D conversion data (upper 2 bits)
           :
        ENDS
;--------------------------------------------------------------------------------
        END
```

CHAPTER 16 UART/SIO

This chapter describes the functions and operations of the UART/SIO.

16.1 Overview of the UART/SIO

The UART/SIO is a general-purpose serial data communication interface. Variable-length serial data can be transferred in clock synchronous or asynchronous mode. The NRZ transfer format is adopted and the transfer rate can be set with the dedicated baud rate generator, the external clock, or the internal timer.

■ Functions of UART/SIO

The UART/SIO functions to transmit/receive serial data (serial I/O) to/from other CPUs and peripheral devices.

- Its full-duplex double buffer allows bidirectional transmission in full-duplex mode.
- A synchronous transfer mode or asynchronous transfer mode can be selected.
- With the built-in baud rate generator, 14 types of baud rates can be selected. In addition, free baud rates can be set using the externally input clock.
- The data length is variable. 7-bits to 8-bits can be set when a parity bit is not attached, and 8-bits to 9-bits can be set when a parity bit is attached (See Table 16.1-1 "Operation Mode of UART/SIO").
- The NRZ (Non Return to Zero) method is adopted as the data transfer format.

Table 16.1-1 Operation Mode of UART/SIO

Operation mode	Data length		Synchronous mode	Stop bit length
	No parity	With parity		
0	7	8	Asynchronous	1 bit or 2 bits
	8	9		
1	8		Synchronous	--

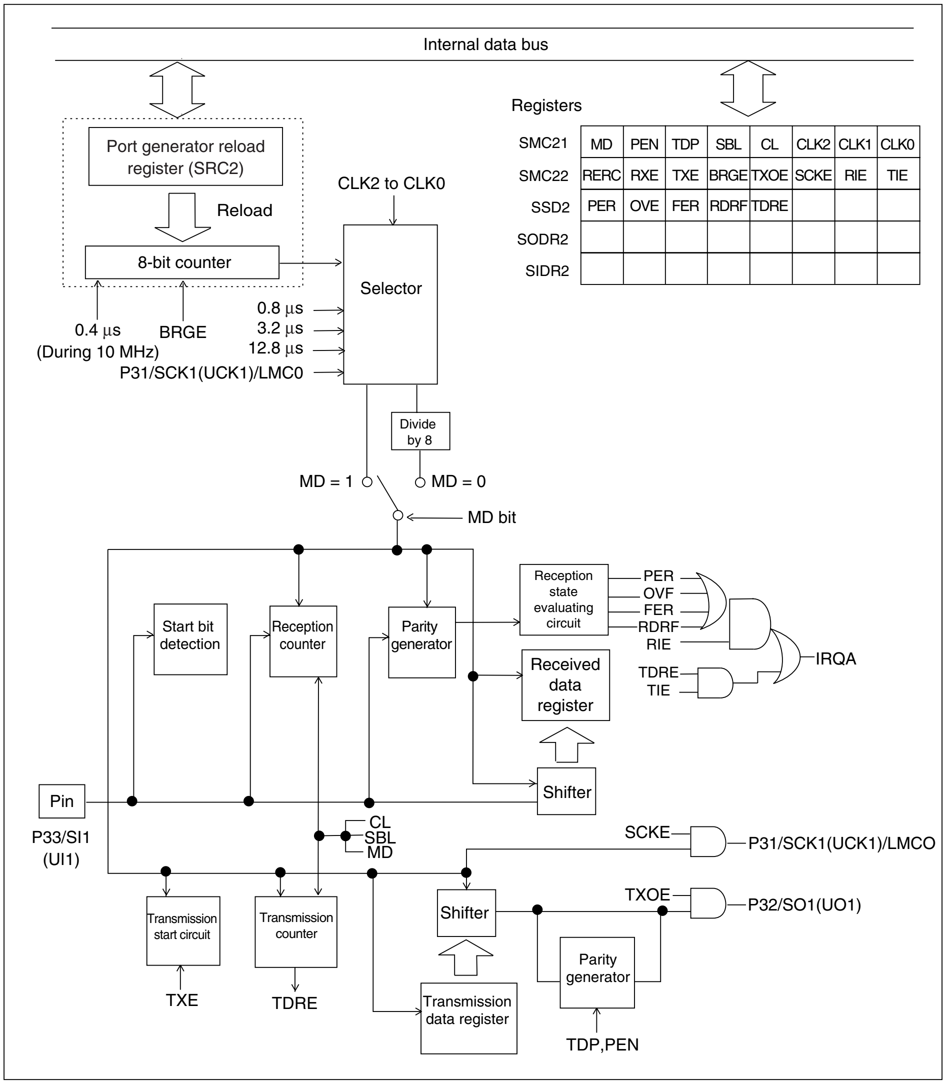
16.2 Configuration of the UART/SIO

The UART/SIO consists of the following six blocks.

- **Serial mode control register 1 (SMC21)**
- **Serial mode control register 2 (SMC22)**
- **Serial rate control register (SRC2)**
- **Serial status/data register (SSD2)**
- **Serial input data register (SIDR2)**
- **Serial output data register (SODR2)**

■ Block Diagram of UART/SIO

Figure 16.2-1 UART/SIO Block Diagram



Registers

SMC21	MD	PEN	TDP	SBL	CL	CLK2	CLK1	CLK0
SMC22	RERC	RXE	TXE	BRGE	TXOE	SCKE	RIE	TIE
SSD2	PER	OVE	FER	RDRF	TDRE			
SODR2								
SIDR2								

- **Serial mode control register 1 (SMC21)**

  A register to control the operation mode of the UART/SIO. This bit is used to select a serial clock and to set a character bit length, stop bit length, parity polarity, parity presence/absence, and clock synchronization/asynchronization.

- **Serial mode control register 2 (SMC22)**

  A register to control the operation mode of the UART/SIO. This register sets the permission/prohibition of serial clock output, permission/prohibition of serial data output, switching between the serial port and the general-purpose port, and permission/prohibition of interrupts.

- **Serial status/data register (SSD2)**

  A register that indicates the state of transmission/reception of the UART/SIO and errors.

- **Serial input data register (SIDR2)**

  A register that holds received data. Serial input is converted and stored in this register.

- **Serial output data register (SODR2)**

  A register that sets transmission data. The data written to this register is converted into serial data and output. When the data length is set to 7-bits, bit 7 is ignored.

- **Port generator reload register (SRC2)**

  A register that controls the UART/SIO data transfer speed (baud rate).

16.3 Pins of the UART/SIO

This section shows the pins related to the UART/SIO and shows a block diagram of the pins.

■ Pins Related to the UART/SIO

The pins related to the UART/SIO are the clock I/O pin (P31/SCK1(UCK1)/LMC0), serial data output pin (P32/SO1(UO1)), and serial data input pin (P33/SI1(UI1)).

P31/SCK1(UCK1)/LMC0

This pin functions as a general-purpose I/O port (P31) or UART/SIO clock I/O pin (hysteresis input). When clock output is enabled (SCKE bit of SMC22 = 1), this pin functions as a UART/SIO clock output pin (SCK1(UCK1)) regardless of the value of the corresponding port data direction register. In this case, do not select an external clock (other than CLK2, CLK1, and CLK0 bits of SMC21 = 100_B). To use this pin as a UART/SIO clock input pin, disable clock output (SCKE bit of SMC22 = 0) and then set this bit as an input port with the corresponding port data direction register (bit 1 of DDR3 = 0). In this case, select an external clock (CLK2, CLK1, and CLK0 bits of SMC21 = 100_B).

P32/SO1(UO1)

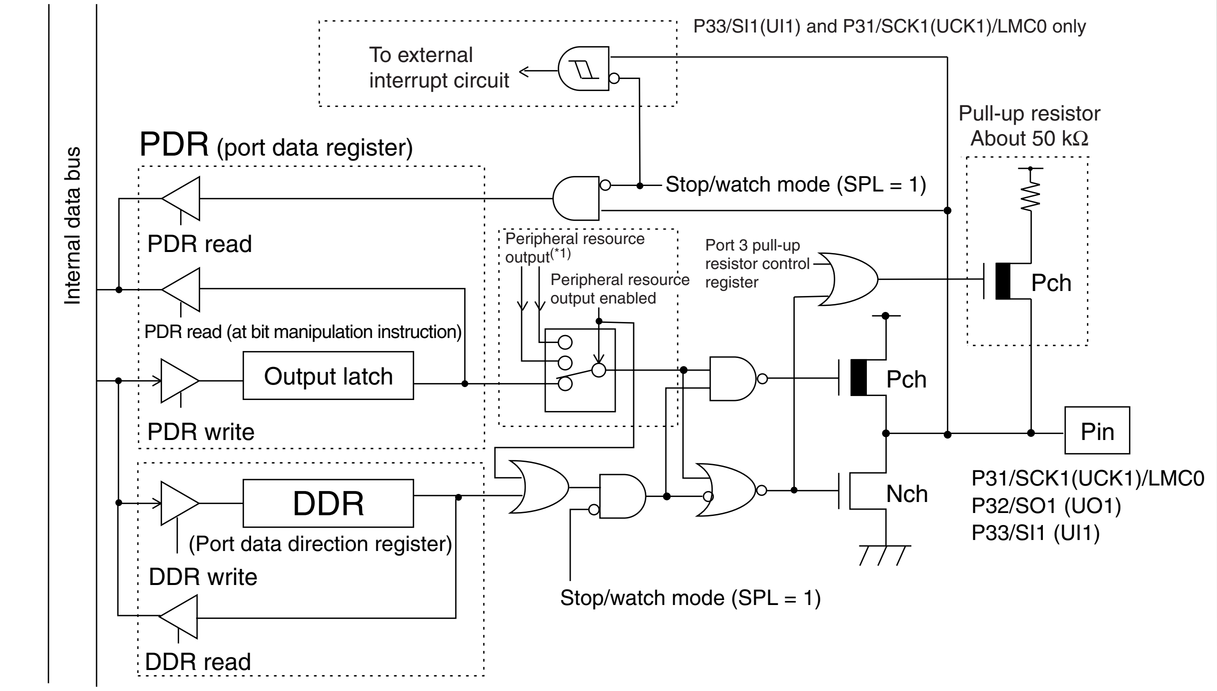
This pin functions as a general-purpose I/O port (P32) or a UART/SIO serial data output pin (S01/U01). When serial data output is enabled (TXOE bit of SMC22 = 1), this pin functions as the UART/SIO serial data output pin (S01/U01) without reference to the value of the corresponding port data direction register.

P33/SI1(UI1)

This pin functions as a general-purpose I/O port (P33) or a UART/SIO serial data input pin (hysteresis input) (SI1/UI1). To use this pin as the UART/SIO serial data input pin, set it as an input port (bit 3 of DDR3 = 0) with the corresponding port data direction register.

■ Block Diagram of Pins Related to UART/SIO

Figure 16.3-1 Block Diagram of Pins Related to UART/SIO



P33/SI1(UI1) and P31/SCK1(UCK1)/LMC0 only
To external interrupt circuit
Internal data bus
PDR (port data register)
Stop/watch mode (SPL = 1)
Pull-up resistor About 50 kΩ
PDR read
Peripheral resource output(*1)
Peripheral resource output enabled
Port 3 pull-up resistor control register
Pch
PDR read (at bit manipulation instruction)
Output latch
Pch
PDR write
Pin
Nch
P31/SCK1(UCK1)/LMC0
P32/SO1 (UO1)
P33/SI1 (UI1)
DDR
(Port data direction register)
DDR write
Stop/watch mode (SPL = 1)
DDR read

SPL: Pin state setting pin in the standby control register (STBC)

*1 Two peripheral resources outputs are connected to P31/SCK1(UCK1)/LMC0.
One peripheral resource output is connected to P32/SO1(UO1).

Reference:

If "pull-up resistor available" is selected in the port 3 pull-up resistor control register, the pins are set to the "H" level (pull-up state), not the Hi-z level, in stop/watch mode (SPL bit of STBC = 1). However, the pull-up is disabled during a reset and the pins are set to the Hi-z level.

16.4 Registers of the UART/SIO

This section shows the registers related to the UART/SIO.

■ Registers Related to UART/SIO

Figure 16.4-1 Registers Related to UART/SIO

SMC21 (Serial mode control register 1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 F H	MD	PEN	TDP	SBL	CL	CLK2	CLK1	CLK0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

SMC22 (Serial mode control register 2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 0 H	RERC	RXE	TXE	BRGE	TXOE	SCKE	RIE	TIE	00000000B
	W	R/W	R/W	W	R/W	R/W	R/W	R/W	

SSD2 (Serial status/data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 31 H	PER	OVE	FER	RDRF	TDRE	-	-	-	00001---B
	R	R	R	R	R				

SIDR2 (Serial input data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 2 H									XXXXXXXXB
	R	R	R	R	R	R	R	R	

SODR2 (Serial output data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 2 H									XXXXXXXXB
	W	W	W	W	W	W	W	W	

SRC2 (Baud rate generator reload register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 3 H									XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

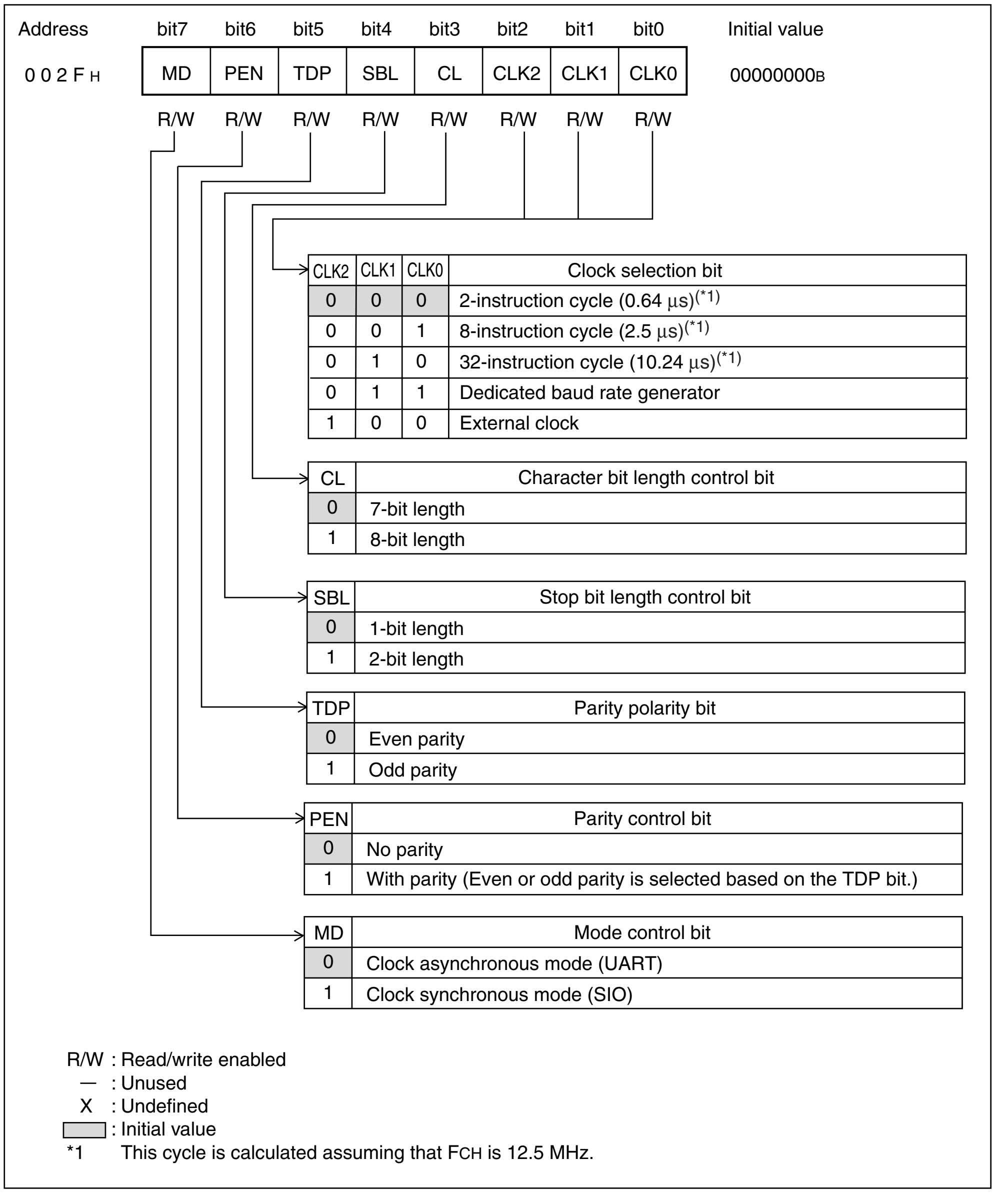
R/W	: Read/write enabled
R	: Read only
W	: Write only
—	: Not used
X	: Undefined

16.4.1 Serial Mode Control Register 1 (SMC21)

The serial mode control register 1 (SMC21) controls the operation mode of the UART/SIO. This register sets the presence/absence of parity, stop bit length, operation mode (data length), synchronous/asynchronous mode, and serial clock.

■ Serial Mode Control Register 1 (SMC21)

Figure 16.4-2 Serial Mode Control Register 1 (SMC1)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
002FH	MD	PEN	TDP	SBL	CL	CLK2	CLK1	CLK0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

CLK2	CLK1	CLK0	Clock selection bit
0	0	0	2-instruction cycle (0.64 µs)(*1)
0	0	1	8-instruction cycle (2.5 µs)(*1)
0	1	0	32-instruction cycle (10.24 µs)(*1)
0	1	1	Dedicated baud rate generator
1	0	0	External clock

CL	Character bit length control bit
0	7-bit length
1	8-bit length

SBL	Stop bit length control bit
0	1-bit length
1	2-bit length

TDP	Parity polarity bit
0	Even parity
1	Odd parity

PEN	Parity control bit
0	No parity
1	With parity (Even or odd parity is selected based on the TDP bit.)

MD	Mode control bit
0	Clock asynchronous mode (UART)
1	Clock synchronous mode (SIO)

R/W : Read/write enabled
— : Unused
X : Undefined
▭ : Initial value
*1 This cycle is calculated assuming that FCH is 12.5 MHz.

Table 16.4-1 Functions of Each Bit in Serial Mode Control Register 1 (SMC21)

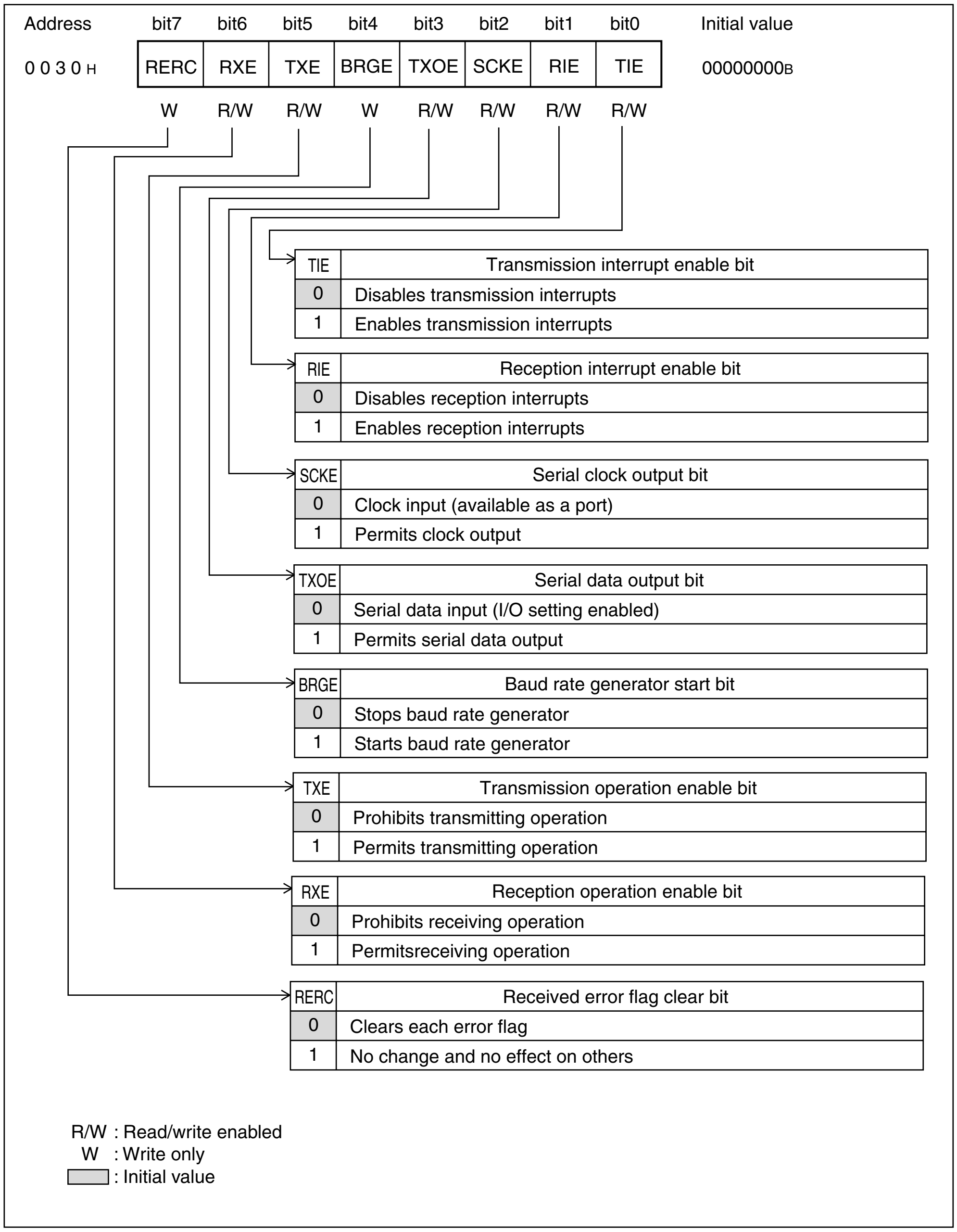
Bit name		Function
Bit 7	MD: Mode control bit	• This bit specifies the operation mode of the UART/SIO. In asynchronous mode, the UART/SIO operates with the serial clock divided by 8. In clock synchronous mode, the UART/SIO operates with the selected serial clock.
Bit 6	PEN: Parity control bit	• This bit specifies whether there is parity in clock asynchronous mode.
Bit 5	TDP: Parity polarity bit	• This bit specifies the parity data attached at the time of serial transmission in clock asynchronous mode. At the time of serial reception, this bit checks the parity data.
Bit 4	SBL: Stop bit length control bit	• This bit specifies the stop bit length in clock asynchronous mode. At the time of serial transmission, this bit attaches a stop bit of the specified bit length. At the time of serial reception, this bit evaluates the stop bit with one bit length irrespective of the set value.
Bit 3	CL: Character bit length control bit	• This bit specifies the character bit length in clock asynchronous mode. • In clock synchronous mode, set this bit to "1".
Bit 2 Bit 1 Bit 0	CLK2 CLK1 CLK0: Clock selection bits	• These bits select a serial clock.

16.4.2 Serial Mode Control Register 2 (SMC22)

The serial mode control register 2 (SMC22) controls the operation mode of the UART/SIO. This register also enables or disables serial clock output, serial data output, interrupts during data transfer, and the baud rate generator.

■ Serial Mode Control Register 2 (SMC22)

Figure 16.4-3 Serial Mode Control Register 2 (SMC22)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0030H	RERC	RXE	TXE	BRGE	TXOE	SCKE	RIE	TIE	00000000B
	W	R/W	R/W	W	R/W	R/W	R/W	R/W	

TIE	Transmission interrupt enable bit
0	Disables transmission interrupts
1	Enables transmission interrupts

RIE	Reception interrupt enable bit
0	Disables reception interrupts
1	Enables reception interrupts

SCKE	Serial clock output bit
0	Clock input (available as a port)
1	Permits clock output

TXOE	Serial data output bit
0	Serial data input (I/O setting enabled)
1	Permits serial data output

BRGE	Baud rate generator start bit
0	Stops baud rate generator
1	Starts baud rate generator

TXE	Transmission operation enable bit
0	Prohibits transmitting operation
1	Permits transmitting operation

RXE	Reception operation enable bit
0	Prohibits receiving operation
1	Permitsreceiving operation

RERC	Received error flag clear bit
0	Clears each error flag
1	No change and no effect on others

R/W : Read/write enabled
W : Write only
▨ : Initial value

Table 16.4-2 Functions of Each Bit in Serial Mode Control Register 2 (SMC2)

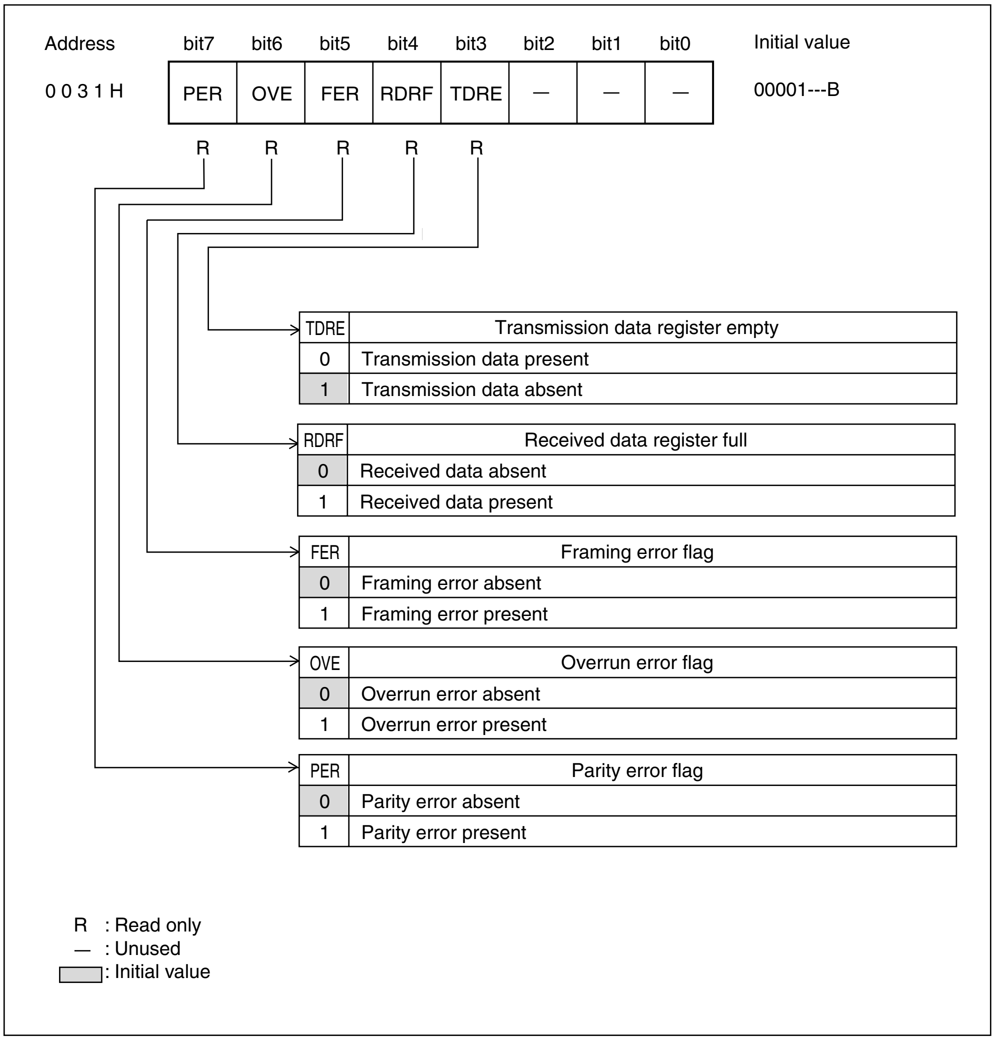
Bit name		Function
Bit 7	RERC: Received error flag clear bit	• When "0" is written to this bit, each error flag (PER/OVR/FER) in the SSD2 register is cleared. In read cycle, the value is always "1". • Writing "1" has no effect on operation.
Bit 6	RXE: Receiving operation enable bit	• This bit permits the reception of serial data. When "0" is written to this bit during a receiving operation, the operation stops after data reception is completed and the receiving operation is prohibited.
Bit 5	TXE: Transmitting operation enable bit	• This bit permits the transmission of serial data. When "0" is written to this bit during a transmitting operation, the operation stops after data transmission is completed and the transmitting operation is prohibited.
Bit 4	BRGE: Baud rate generator start bit	• This bit starts the baud rate generator.
Bit 3	TXOE: Serial data output bit	• This bit controls the permission/prohibition of serial data output.
Bit 2	SCKE: Serial clock output bit	• This bit controls the I/O of the serial clock in clock synchronous mode. To input an external clock to the P31/SCK1(UCK1)/LMC0 pin, set the port 3 data direction register to input (bit 1 of DDR3 = 0).
Bit 1	RIE: Reception interrupt enable bit	• This bit enables reception interrupts. If reception interrupts are enabled when the RDRF bit is "1" or when each error flag is "1", a reception interrupt occurs immediately.
Bit 0	TIE: Transmission interrupt enable bit	• This bit enables transmission interrupts. If transmission interrupts are enabled when the TDRE bit is "1", a transmission interrupt occurs immediately.

16.4.3 Serial Status/Data Register (SSD2)

The serial status/data register (SSD2) indicate the state of transmission/reception of the UART/SIO and errors.

■ Serial Status/Data Register (SSD2)

Figure 16.4-4 Serial Status/Data Register (SSD2)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 1 H	PER	OVE	FER	RDRF	TDRE	—	—	—	00001---B
	R	R	R	R	R				

TDRE	Transmission data register empty
0	Transmission data present
1	Transmission data absent

RDRF	Received data register full
0	Received data absent
1	Received data present

FER	Framing error flag
0	Framing error absent
1	Framing error present

OVE	Overrun error flag
0	Overrun error absent
1	Overrun error present

PER	Parity error flag
0	Parity error absent
1	Parity error present

R : Read only
— : Unused
▨ : Initial value

Table 16.4-3 Functions of Each Bit in Serial Status/Data Register (SSD2)

Bit name		Function
Bit 7	PER: Parity error flag	• This bit is set if a parity error occurs during reception and is cleared when "0" is written to the RERC bit in the SMC22 register. When this flag is set, the data in SIDR2 becomes invalid. If the PER bit is set when the RIE bit is set to "1", an interrupt occurs.
Bit 6	OVE: Overrun error flag	• This bit is set if an overrun error occurs during reception and is cleared when "0" is written to the RERC bit in the SMC22 register. When this flag is set, the data in SIDR2 becomes invalid. If the OVE bit is set when the RIE bit is set to "1", an interrupt occurs.
Bit 5	FER: Framing error flag	• This bit is set if an framing error occurs during reception and is cleared when "0" is written to the RERC bit in the SMC22 register. When this flag is set, the data in SIDR2 becomes invalid. If the FER bit is set when the RIE bit is set to "1", an interrupt occurs.
Bit 4	RDRF: Received data register full	• This bit is a flag indicating the state of the received data register (SIDR2). This bit is set when the received data is loaded to the SIDR register and is cleared when the SIDR2 register is read. If the RDRF bit is set when the RIE bit is set to "1", an interrupt occurs.
Bit 3	TDRE: Transmission data register empty	• This bit is a flag indicating the state of the serial transmission data register (SODR2). This bit is cleared when the transmission data is written to the SODR register and is set when the data is loaded to the shifter for transmission and transmission of the data starts. If the TDRE bit is set when the TIE bit is set to "1", an interrupt occurs.
Bit2 Bit1 Bit0	Unused bits	• The read value is undefined. • Writing has no effect on operation.

16.4.4 Serial Input Data Register (SIDR2)

The serial input data register (SIDR2) is a register for inputting (receiving) serial data.

■ Serial Input Data Register (SIDR2)

Figure 16.4-5 "Serial Input Data Register (SIDR2)" shows the bit configuration of the serial input data register.

Figure 16.4-5 Serial Input Data Register (SIDR2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0032H									XXXXXXXXB
	R	R	R	R	R	R	R	R	

R : Read only
X : Undefined

The SIDR is a register for storing the received data. The serial data signal sent to the serial input pin (SI1(UI1) pin) is converted in the shift register and stored in this register.

❍ Operation in mode 0 and mode 1

When the received data is set to this register successfully, the received data flag bit (SSD2: RDRF) is set to "1". If the reception interrupt request is enabled, an interrupt occurs. When the received data is stored in this register in an interrupt or when checking the RDRF bit with the program, the RDRF flag is cleared by reading the description in this register.

16.4.5 Serial Output Data Register (SODR2)

The serial output data register (SODR2) is a register for outputting (transmitting) serial data.

■ Serial Output Data Register (SODR2)

Figure 16.4-6 "Serial Output Data Register (SODR2)" shows the bit configuration of the serial output data register.

Figure 16.4-6 Serial Output Data Register (SODR2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0032$_H$									XXXXXXXX$_B$
	W	W	W	W	W	W	W	W	

R : Write only
X : Undefined

If the SSD2 register is read and the data to be transmitted is written to this register when transmission is permitted, transmission data is transferred to the sift register for transmission and converted to serial data. The converted data is then transmitted from the serial data output pin (SO1(UO1) pin).

When the transmission data is written to the SODR2 register, the transmission data flag bit is set to "0". After the transmission data is transferred to the shift register for transmission, the transmission data flag bit is set to "1" so that the next transmission data can be written in the register. If the interrupt request is enabled at this time, an interrupt occurs. The next transmission data can be written by generating an interrupt or when the transmission data flag bit is set to "1".

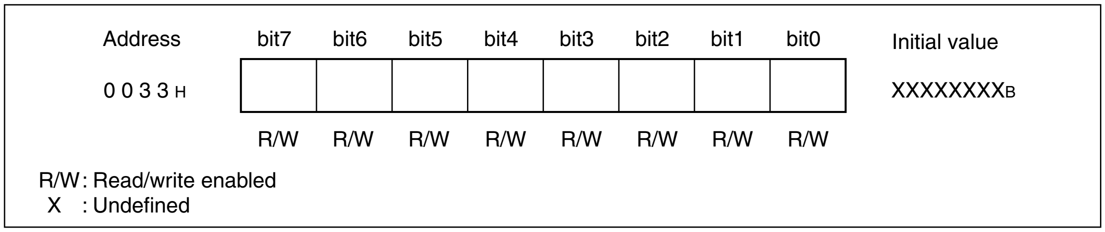
16.4.6 Baud Rate Generator Reload Register (SRC2)

The baud rate generator reload register (SRC2) controls the UART/SIO data transmission speed (baud rate).

■ Baud Rate Generator Reload Register (SRC2)

Figure 16.4-7 "Baud Rate Generator Reload Register (SRC2)" shows the bit configuration of the baud rate generator reload register (SRC2).

Figure 16.4-7 Baud Rate Generator Reload Register (SRC2)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 3 H									XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Read/write enabled
X : Undefined

When CLK2 to CLK0 (clock selection bits) are "011$_B$", the dedicated baud rate generator is selected as a serial clock. This register can be used to set an optional baud rate. Write data to this register when UART operation is stopped.

16.5 UART/SIO Interrupt

The UART/SIO has three flags related to interrupts, the error flag bits (PER, OVE, FER), the received data flag bit (RDRF), and the transmission data flag bit (TDRE), as well as the following two interrupt sources.

- **When the received data is transferred from the shift register for reception to the serial input data register (SIDR2) (reception interrupt)**
- **When the transmission data is transferred from the serial output data register (SODR2) to the shift register for transmission. (transmission interrupt)**

■ Transmission Interrupt

Output data is written to the SODR2 register and then transferred to the shift register for internal transmission. When the register is ready to accept the next data, the TDRE bit is set to "1". If the transmission interrupt is enabled (SMC22: TIE = 1), an interrupt request to the CPU (IRQA) occurs.

■ Reception Interrupt

After data is input up to the stop bit successfully, the RDRF bit is set to "1". If an overrun, parity, or error framing error has occurred, the bit of the corresponding error flag is set to "1".

These bits are set when the stop bit is detected. If the reception interrupt is enabled (SMC22: RIE = 1), an interrupt request to the CPU (IRQA) occurs.

■ Register and Vector Table Address Related to Interrupt of UART/SIO

Table 16.5-1 Register and Vector Table Address Related to Interrupt of UART/SIO

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Bit to be set		Upper	Lower
IRQA	ILR3 (007D$_H$)	LA1 (bit 5)	LA0 (bit 4)	FFE6$_H$	FFE7$_H$

For interrupt operation, see Section 3.4.2 "Interrupt Processing".

16.6 Operation of the UART/SIO

This section describes the operation of the UART/SIO.
The UART/SIO has ordinary serial communication functions (operation modes 0 and 1).

■ Operation of UART/SIO

❍ Operation modes

The UART/SIO has two operation modes: clock synchronous mode (SIO) and clock asynchronous mode (UART). (See Table 16.1-1 "Operation Mode of UART/SIO".)

16.7 Operation of the Operation Mode 0

Operation mode 0 operates in clock asynchronous mode.

■ Explanation of Operation Mode 0 of UART/SIO

The serial clock is selected, with bits CLK21 to CLK0 in the SMC1 register, from among three types of internal clocks, an external clock, and a baud rate generator output. When the external clock is selected, the clock must be entered.

In CLK asynchronous mode, the shift clock selected with bits CLK2 to CLK0 is divided by 8 and data can be transferred in the range between -2% and +2% of the selected baud rate. The baud rate calculation expressions for the internal and external clocks and the baud rate generator are shown in the following.

Figure 16.7-1 Baud Rate Calculation Expression when Dedicated Baud Rate Generator is Used

$$\text{Baud rate value} = \frac{1}{8 \times 2 \times \underset{\text{Clock gear selection}}{\boxed{\begin{array}{r} 64/F_{CH} \\ 16/F_{CH} \\ 8/F_{CH} \\ 4/F_{CH} \end{array}}} \times \boxed{\begin{array}{c} \text{SRC2 register value} \\ \text{(SRC2)} \end{array}}} \quad \text{[bps]}$$

F_{CH}: Frequency used

Figure 16.7-2 Expression for Baud Rate Calculation with Internal and External Clocks

$$\text{Baud rate value} = \frac{1}{8 \times \boxed{\begin{array}{c} \text{Clock selected according} \\ \text{to CLK2 to CLK0} \end{array}}} \quad \text{[bps]}$$

Table 16.7-1 Example of the Asynchronous Transfer Rate with the Baud Rate Generator

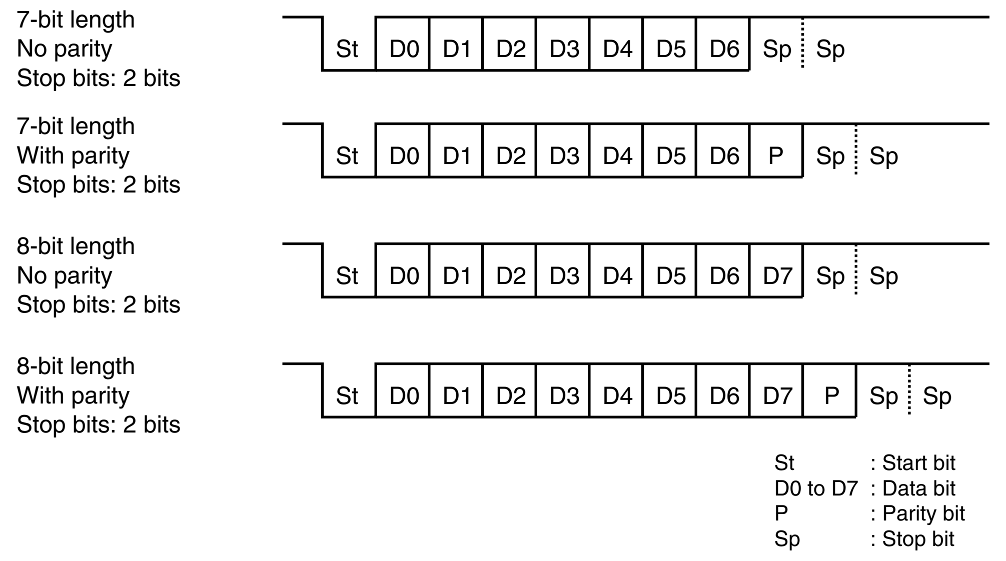
Operating frequency	10 MHz	8 MHz	7.3728 MHz	4.9152 MHz
Instruction cycle	$4/F_{CH}$ 0.4 µs	$4/F_{CH}$ 0.5 µs	$4/F_{CH}$ 0.54 µs	$4/F_{CH}$ 0.81 µs
Baud rate Values in parentheses indicate SRC2 register set values	78125(n=2)	62500(n=2)	--	76800(n=1)
	39062(n=4)	31250(n=4)	38400(n=3)	38400(n=2)
	19531(n=8)	17857(n=7)	19200(n=6)	19200(n=4)
	9765(n=16)	9615(n=13)	9600(n=12)	9600(n=8)
	4882(n=32)	4807(n=26)	4800(n=24)	4800(n=16)
	2403(n=65)	2403(n=52)	2400(n=48)	2400(n=32)
	1201(n=130)	1201(n=104)	1200(n=96)	1200(n=64)
	--	600(n=208)	600(n=192)	600(n=128)
	--	--	--	300(n=0)

■ Transfer Data Format

The UART/SIO can only use the NRZ (Non Return to Zero) format data. Figure 16.7-3 "Transfer Data Format" shows the data format. In the following example, the stop bit length is two bits.

As shown in Figure 16.7-3 "Transfer Data Format", data transfer always starts with the start bit ("L" level), followed by the data bit length specified as the LSB first, and ends with the stop bit ("H" level). In an idle state, it is at the "H" level.

Figure 16.7-3 Transfer Data Format

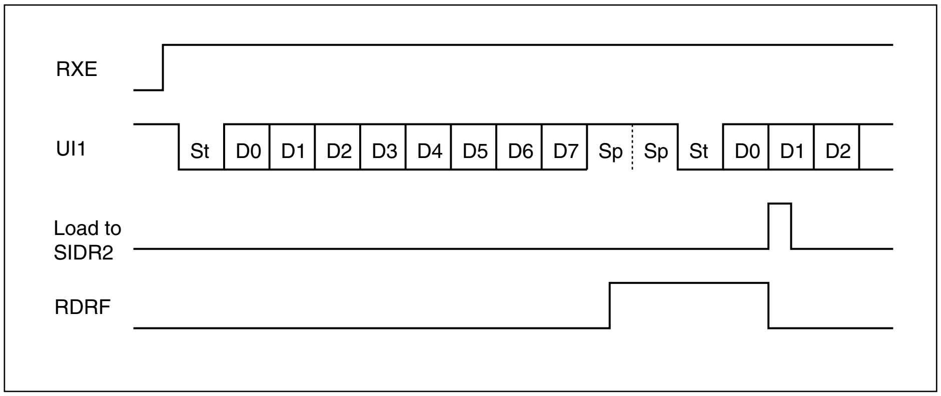


■ Receiving Operation in CLK Asynchronous Mode

Select the baud rate clock with bits CLK2 to CLK0 in the SMC21 register. For the baud rate clock, see the section about clock selection. In a receiving operation, reception is permitted when the RXE bit in the SMC1 register is "1" and the receiving operation starts at the first falling edge of the input data (detection of the start bit). When the receiving operation is completed, the RDRF bit in the SSD register is set to "1" and the received data is loaded to the SIDR register. If the RDRF bit is set to "1" when the RIE bit is "1", a reception interrupt to the CPU is generated. If any of the three errors (PER/OVE/FER) is detected when reception is completed, the RDRF bit is not set to "1" and the received data is not loaded to the SIDR register. Therefore, the value in the SIDR register is the previously received data. Unless the RXE bit is set to "0", the receiving operation is continued whenever a start bit is detected even if an error flag is set.

If "0" is written to the RXE bit of the SMC22 register during a receiving operation, the receiving operation is prohibited after data reception is completed.

Figure 16.7-4 Receiving Operation of CLK Asynchronous Mode

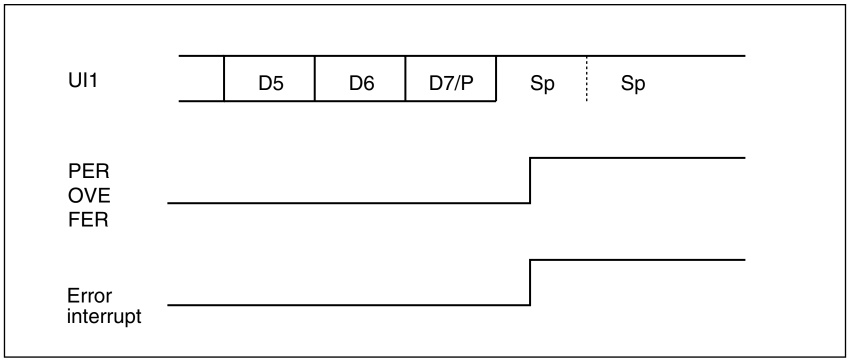


■ Reception Error in CLK Asynchronous Mode

In CLK asynchronous mode, three types of errors are detected. When a parity error, overrun error, or framing error is detected, the PER, OVE, or FER bit in the SSD2 register is set to "1", respectively.

The detection of these errors are performed at the end of reception as shown in the following. When any of these errors is detected, RDRF is not set and the received data is not loaded to the SIDR2 register. Therefore, the value in the SIDR2 register is the previously received data. By writing "0" to the RERC bit in the SCM22 register, all of the three error flags are cleared.

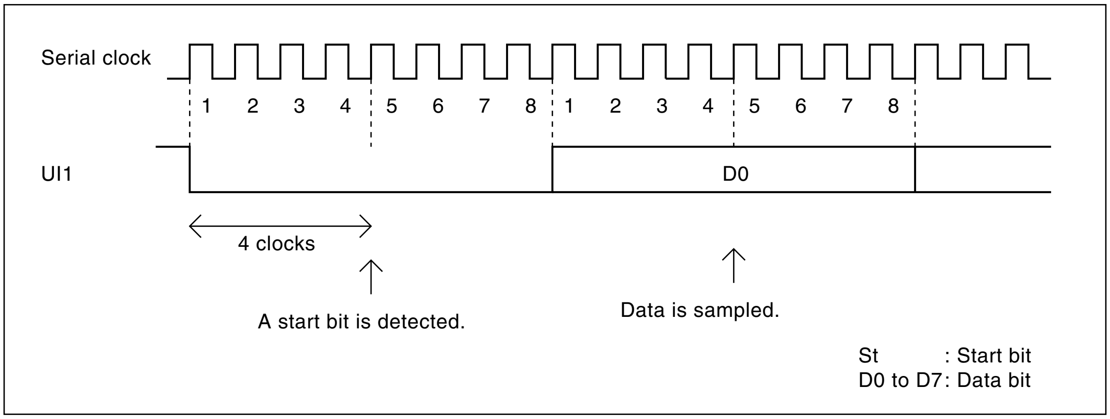
Figure 16.7-5 Reception Error Setting Timing



■ Detecting the Start Bit At Receiving Operation

When the "L" level remains for four clocks with the selected serial clock (generator output, etc.) after the first falling edge of the input data, the UART/SIO regards it as a start bit. After the start bit is detected, data is sampled at the rising edge of the fifth clock of the serial clock after the start bit is detected.

Figure 16.7-6 Detecting a Start Bit

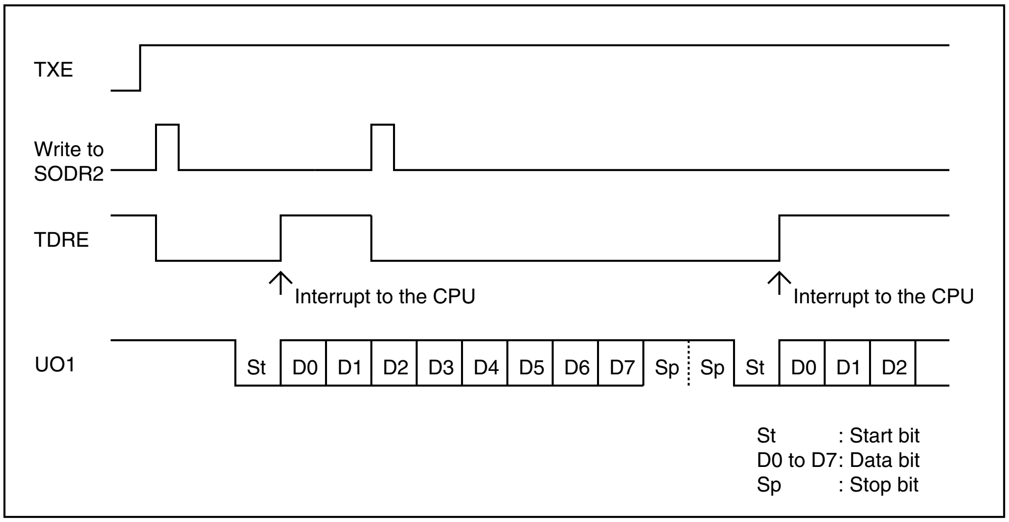


■ Transmitting Operation in CLK Asynchronous Mode

If transmission data is written to the SODR2 register when the TXE bit in the SMC22 register is "1", the TDRE bit in the SSD2 register is cleared and a transmitting operation starts. When the data in the SODR2 register is loaded to the shifter and the output of transmission data starts, the TDRE bit in the SSD2 register is set. If data is written to the SODR register when data is being transmitted (when the TDRE bit is set to "1"), the TDRE bit is cleared and data is transmitted continuously following the transmission of the specified bit length data.

If "0" is written to the TXE bit in the SMC22 register during a transmitting operation, the transmitting operation is prohibited following the transmission of the specified bit length data when the SODR2 register is vacant (when the TDRE bit is set to "1"). When there is data in the SODR2 register (when the TDRE bit is set to "1"), the transmitting operation is prohibited after the data in the SODR register is transmitted.

Figure 16.7-7 Transmission in CLK Asynchronous Mode



16.8 Operation of the Operation Mode 1

Operation mode 1 operates in clock synchronous mode.

■ Explanation of UART/SIO Operation Mode

In CLK synchronous mode, the clock is selected, with bits CLK2 to CLK0 in the SMC21 register, from among three types of internal clocks, an external clock, and a baud rate generator output. Shift operation is performed with the selected clock as a shift clock. When the external clock is entered, set the SCKE bit to "0".

When the internal clock or the output of the baud rate generator is output as a shift clock, set the SCKE bit to "1". The baud rate calculation expressions for the internal and external clocks and the baud rate generator are shown in the following

Figure 16.8-1 Baud Rate Calculation Expression when the Dedicated Baud Rate Generator is Used (CLK2, CLK1, CLK0 = 011B)

$$\text{Baud rate value} = \frac{1}{2 \times \underset{\text{Clock gear selection}}{\boxed{\begin{matrix}64/F_{CH}\\16/F_{CH}\\8/F_{CH}\\4/F_{CH}\end{matrix}}} \times \boxed{\begin{matrix}\text{SRC2 register value}\\\text{(SRC2)}\end{matrix}}}\quad[\text{bps}]$$

F_{CH}: Main clock oscillation frequency

Figure 16.8-2 Baud Rate Calculation Expression for the Internal and External Clocks (Other than CLK2, CLK1, CLK0 = 011B)

$$\text{Baud rate value} = \frac{1}{\boxed{\begin{matrix}\text{Selected clock for}\\\text{CLK2 to CLK0}\end{matrix}}}\quad[\text{bps}]$$

■ 8-bit Receiving Operation

Figure 16.8-3 Registers Used at Reception in Operation Mode 1

Serial mode control register 1 (SMC21)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MD	PEN	TDP	SBL	CL	CLK2	CLK1	CLK0
1	0	0	0	1	◎	◎	◎

Serial mode control register 2 (SMC22)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RERC	RXE	TXE	BRGE	TXOE	SCKE	RIE	TIE
◎	◎	◎	×	0	◎	◎	×

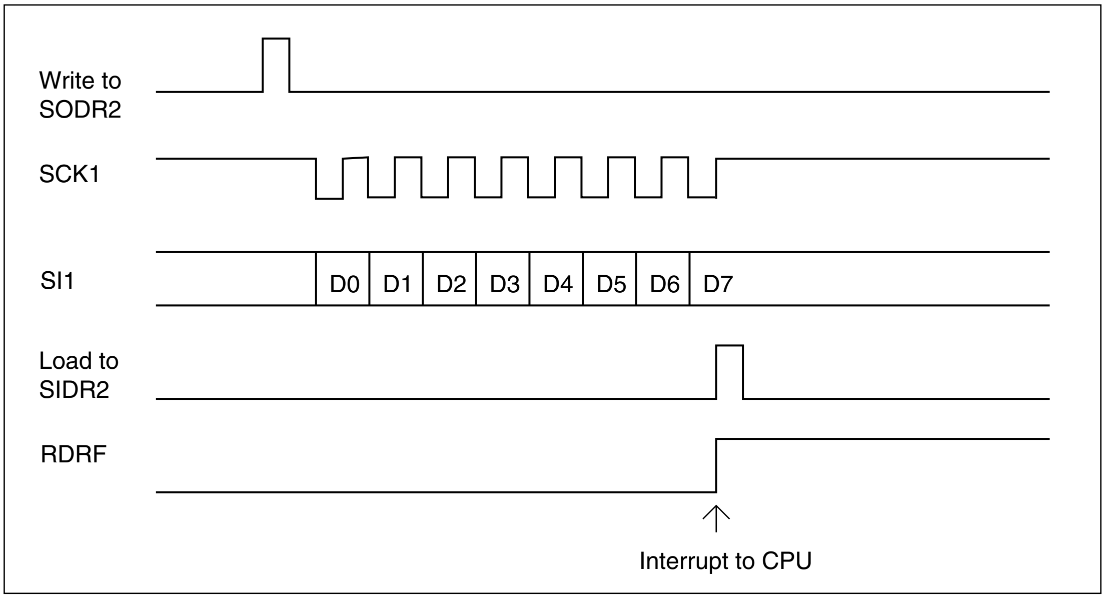
Serial status and data register (SSD2)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PER	OVE	FER	RDRF	TDRE	—	—	—
×	◎	×	◎	×			

◎ : Used bit
× : Unused bit
1 : Bit with 1
0 : Bit with 0

Receiving operation is permitted by setting the TXE and RXE bits of serial mode control register 2 (SMC22) to "1" and then started when data is written to the SODR2 register. Reception is performed in synchronization with the rising edge of the shift clock. Upon the completion of 8-bit data reception, shifter data is loaded to the SIDR2 register and the RDRF flag is then set to "1". If the RIE bit is set to "1" at this time, an interrupt request to the CPU is generated. If an overrun error occurs when the receiving operation finishes, data is not loaded into the SIDR2 register. Writing "0" to the RXE bit during a receiving operation stops the receiving operation after the 8-bit data has been received. When serial operation is stopped, always keep the serial clock input at the "H" level (without referencing the value of the RXE bit).

Figure 16.8-4 8-Bit Data Receiving Operation in CLK Synchronous Mode

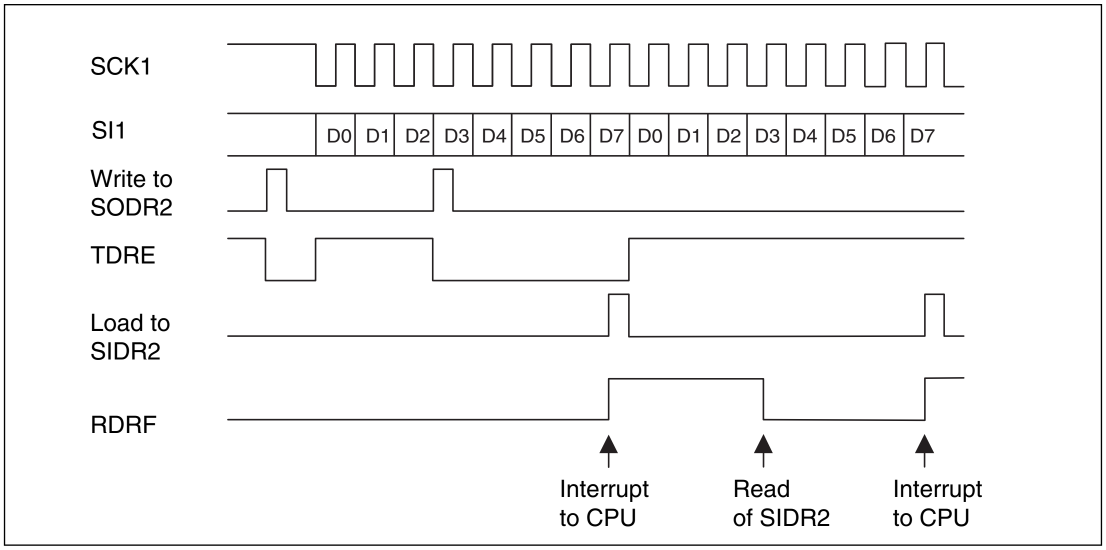


■ Continuous Receiving Operation

In CLK synchronous mode, not only 8-bit data reception but also continuous reception can be performed. In continuous reception, the TIE bit of the SMC22 register and the TDRE bit of the SSD2 register are used in addition to the bits used for 8-bit data reception. Reception is enabled by setting the TXE and RXE bits of the serial mode control register 2 (SMC22) to "1" and starts when data is written to the SODR2 register. This is performed in synchronization with the rising edge of the shift clock. When a shift operation is started, the TDRE bit of SSD2 is set to "1". If the TIE bit of SMC22 is set to "1" at this time, a CPU interrupt is generated. Writing data to the SODR2 register before 8-bit shift operation finishes permits the next shift operation and continuous reception even after the reception of 8-bit data. When 8-bit data reception finishes, shifter data is loaded to the SIDR2 register and the RDRF flag of SSD2 is set to "1". If the RIE bit of SMC22 is "1" at this time, a CPU interrupt request is generated.

If an overrun error occurs upon the completion of 8-bit data reception, data is not loaded to the SIDR2 register. In this case, the contents of the SIDR2 register are those of the previously received data. Reading the SIDR2 register clears the reception interrupt (RDRF flag of SSD2). Continuous reception stops when "0" is written to the RXE bit of SMC22. If "0" is written to the RXE bit of SMC22 when 8-bit data is being received, continuous reception stops after 8-bit data has been received.

Figure 16.8-5 Continuous Receiving Operation in CLK Synchronous Mode



■ 8-bit Transmitting Operation

Figure 16.8-6 Registers Used at Transmission in Operation Mode 1

Serial mode control register 1 (SMC21)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MD	PEN	TDP	SBL	CL	CLK2	CLK1	CLK0
1	0	0	0	1	◎	◎	◎

Serial mode control register 2 (SMC22)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RERC	RXE	TXE	BRGE	TXOE	SCKE	RIE	TIE
◎	◎	◎	×	1	◎	◎	◎

Serial status and data register (SSD2)

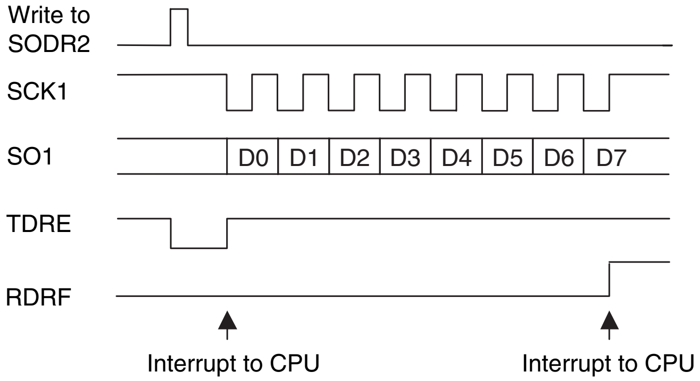
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PER	OVE	FER	RDRF	TDRE	—	—	—
×	×	×	◎	◎			

◎ : Used bit
× : Unused bit
1 : Bit with 1
0 : Bit with 0

Transmission is started by setting the TXE and RXE bits of the serial mode control register 2 (SMC22) to "1" and writing data to the SODR2 register. When transmission is started, the data written to the SODR2 register is loaded to the shifter and then shift operation is performed. When data is loaded from the SODR2 register to the shifter, the TDRE flag of SSD2 is set to "1". If the TIE bit of SMC22 is "1" at this time, a CPU interrupt request is generated. Serial data output is permitted by setting the TXOE bit of SMC22 to "1". Serial data is output in synchronization with the falling edge of the shift clock.

If "0" is written to the TXE bit of SMC22 when 8-bit data is being transmitted, transmission stops after 8-bit data has been transmitted. After 8-bit data has been transmitted, the RDRF bit of SSD2 is set to "1". If the RIE bit of SMC22 is "1" at this time, a CPU interrupt request is generated. Data transmission starts with bit 0 and ends with bit 7. When serial operation is stopped, keep the serial clock input at the "H" level (without referencing the value of the TXE bit of SMC22).

Figure 16.8-7 8-Bit Transmitting Operation in CLK Synchronous Mode

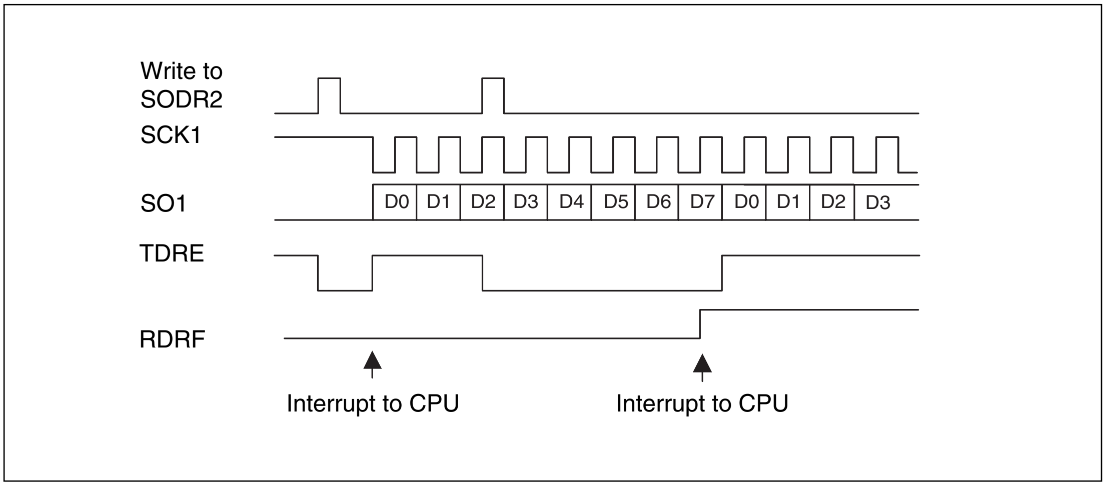


■ Continuous Transmission at Operation Mode 1

In CLK synchronous mode, not only 8-bit data transmission but also continuous transmission can be performed. Transmission is started by setting the TXE and RXE bits of serial mode control register 2 (SMC22) to "1" and writing data to the SODR2 register. When transmission is started, the data written to the SODR2 register is loaded to the shifter and then shift operation is performed. When data is loaded from the SODR2 register to the shifter, the TDRE flag of SSD2 is set to "1". If the TIE bit of SMC22 is "1" at this time, a CPU interrupt request is generated.

Continuous transmission is performed by writing the next transmission data to the SODR2 register during transmission when the TDRE bit is "1" (the SODR2 register is vacant). When data is written to the SODR2 register, the TDRE bit of SSD2 is cleared. After 8-bit data has been transmitted, the data written to the SODR2 register is loaded to the shifter to continue transmission. Transmission stops when "0" is written to the TXE bit of SMC22. If the SODR2 register is vacant (TDRE bit of SSD2 = "1") when "0" is written to the TXE bit during transmission, transmission stops after 8-bit data has been transmitted. If data exists in the SODR2 register (TDRE bit of SSD2 = "0"), transmission stops after data in the SODR2 register has been transmitted. When 8-bit data transmission finishes, the RDRF bit of SSD2 is set to "1". If the RIE bit is "1" at this time, a CPU interrupt request is generated.

Figure 16.8-8 Continuous Receiving Operation in CLK Synchronous Mode



CHAPTER 17 HIGH-SPEED UART

This chapter describes the functions and operation of the high-speed UART.

17.1 Overview of the High-Speed UART

The high-speed UART is a general-purpose serial data communication interface used to transfer variable-length data in clock synchronous or clock asynchronous mode. The transfer format is the non-return-to-zero (NRZ) transfer format. The transfer rate can be set with a dedicated baud rate generator, external clock, or internal timer (2-channel 8-bit PWM timer 1). Either a single-clock or a dual-clock system can be used.

■ Function of the High Speed UART

The high-speed UART transmits serial data to and receives serial data from other CPUs and peripheral devices (input and output of serial data).

- Its full-duplex double buffer allows bidirectional transmission in full-duplex mode.
- Synchronous transfer mode or asynchronous transfer mode can be selected.
- With the built-in baud rate generator, one of 14 baud rates can be selected. Moreover, an arbitrary baud rate can be specified using the externally input clock and 2-channel 8-bit PWM timer 1 output.
- Variable-length data is supported. Without a parity bit, a data length of 5-bits to 9-bits can be selected. With a parity bit, 4-bits to 8-bits can be specified (see Table 17.1-1 "Operation Modes of the High-Speed UART".
- The non-return-to-zero (NRZ) transfer format is used for data transfer.
- Use of either the single-clock system or dual-clock system can be specified by software.

Table 17.1-2 "Transfer Cycles and Transfer Rates When Baud Rate Generator Used", Table 17.1-3 "Transfer Cycles and Transfer Rates When External Clock Used", and Table 17.1-4 "Transfer Cycles and Transfer Rates When 2-channel 8-Bit PWM Timer 1 Used" list the transfer rates provided by the dedicated baud rate generator, external clock, and 2-channel 8-bit PWM timer 1, respectively.

Table 17.1-1 Operation Modes of the High-Speed UART

Operation mode	Data length		Synchronous mode	Stop bit count
	No parity	With parity		
0	5	4	Asynchronous/synchronous	1-bit or 2-bits (*1)
1	7	6	Asynchronous/synchronous	1-bit or 2-bits (*1)
2	8	7	Asynchronous/synchronous	1-bit or 2-bits (*1)
3	8+1	-	Asynchronous/synchronous	1-bit or 2-bits (*1)
4	9	8	Asynchronous/synchronous	1-bit or 2-bits (*1)

*1: However, only one stop bit can be identified during reception. The second stop bit is ignored if transmitted.

■ Transfer Clock Selection

Figure 17.1-1 Baud Rate Generator and Serial Clock Generator

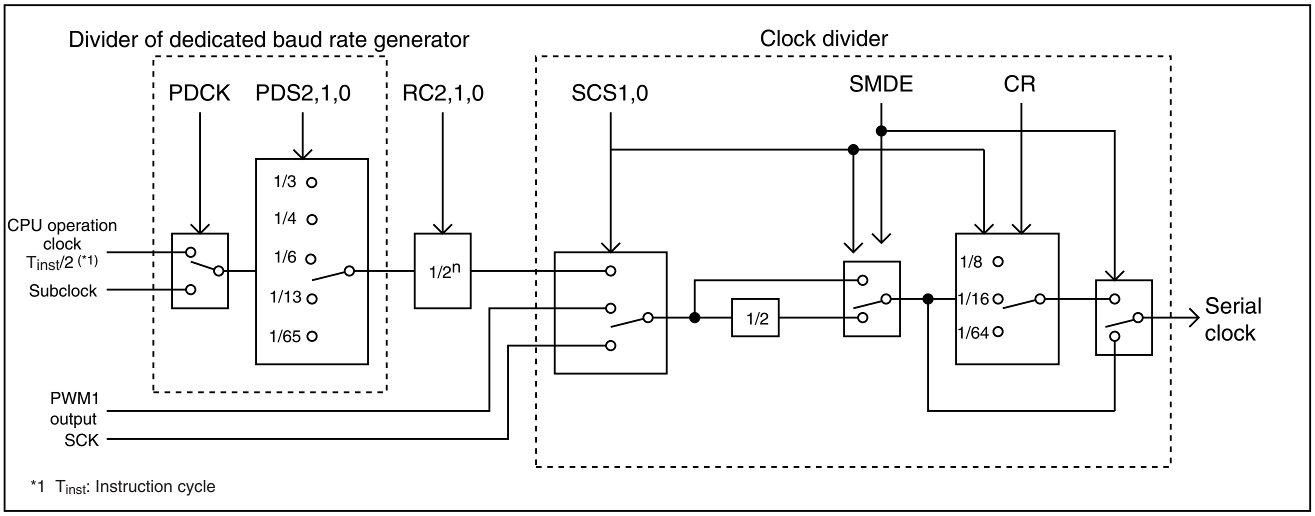


Table 17.1-2 Transfer Cycles and Transfer Rates When Baud Rate Generator Used

				In asynchronous transfer mode		In synchronous transfer mode	
				9.216MHz	9.984MHz	10MHz	Clock frequency
				1/3	1/65	1/3	PDS division
				1/8	1/8	1/1	SCS, CR division
RC2	RC1	RC0	Baud rate selection division ratio	Transfer rate (*1) (µs/baud)	Transfer rate (*1) (µs/baud)	Transfer rate (*1) (µs/baud)	
0	0	0	1	5.2/192K	104/9600	0.6/1.67M	
0	0	1	2	10.4/96K	208/4800	1.2/833.3K	
0	1	0	4	20.8/48K	416/2400	2.4/416.7K	
0	1	1	8	41.7/24K	832/1200	4.8/208.3K	
1	0	0	16	83.3/12K	1664/600	9.6/104.2K	
1	0	1	32	166.7/6K	3328/300	19.2/52.08K	
1	1	0	64	333.3/3K	-	38.4/26.04K	
1	1	1	128	666.7/1.5K	-	76.8/13.02K	

*1: When the maximum main clock rate is selected

Figure 17.1-2 Example for Baud Rate Calculation

$$\frac{1}{\text{Baud rate}} = \overset{\text{Clock gear selection}}{\boxed{\begin{array}{r}64/F_{CH}\\16/F_{CH}\\8/F_{CH}\\4/F_{CH}\end{array}}} \times 1/2 \times \boxed{\begin{array}{c}\text{Clock division ratio}\\\text{(SCS1, SCS0, CR)}\end{array}} \times \boxed{\begin{array}{c}\text{Baud rate selection}\\\text{(PDS2, PDS1, PDS0)}\end{array}} \times \boxed{\begin{array}{c}\text{Baud rate selection}\\\text{(RC2, RC1, RC0)}\end{array}}$$

F_{CH}: Main clock oscillation frequency

Reference:

A dedicated baud rate is set by the clock gear register/clock division ratio register (SCS1, SCS0) and the baud rate selection register. See Table 17.1-1 "Operation Modes of the High-Speed UART" for examples of baud rate calculation.

1/208333 baud = 0.4 µs ($4/F_{CH}$) x 1/2 x 8 (asynchronous mode) x 3 (RC2 = RC1 = RC0 = 0) (F_{CH} = 10 MHz)

Table 17.1-3 Transfer Cycles and Transfer Rates When External Clock Used

Asynchronous transfer mode				Synchronous transfer mode		
Baud rate division ratio		Transfer cycle	Transfer rate (*1) (baud)	Baud rate division ratio	Transfer cycle	Transfer rate (*1) (baud)
CR=0	16	$96/F_{CH}$ or more	104.2 k or less	1	$16/F_{CH}$ or more	625 k or less
CR=1	64	$384/F_{CH}$ or more	26041 or less			

F_{CH}: Main clock oscillation frequency
*1: The minimum external clock cycle when F_{CH} is set to 10 MHz ($6/F_{CH}$ = 0.6 µs)

Figure 17.1-3 Example of Baud Rate Calculation (When External Clock is Selected)

$$\frac{1}{\text{Baud rate}} = \begin{array}{c}\text{External clock input} \times \text{CR}\\(\text{min. : } 8/F_{CH} \times 2)\end{array} \left(\begin{array}{l}16 \text{ for CR} = 0\\64 \text{ for CR} = 1\end{array}\right)$$

F_{CH}: Main clock oscillation frequency

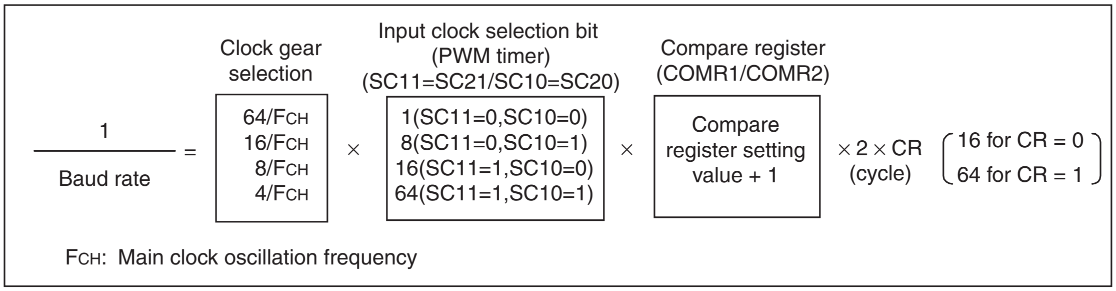
Table 17.1-4 Transfer Cycles and Transfer Rates When 2-channel 8-Bit PWM Timer 1 Used

PWM timer counter clock cycle	Asynchronous transfer mode			Synchronous transfer mode	
	Clock division ratio		Transfer rate (baud)	Clock division ratio	Transfer rate (baud)
1 t_{inst}	CR=0	16	78125 to 610.4	2	625k to 4.88k
	CR=1	64	19531.3 to 152.6		
8 t_{inst}	CR=0	16	9765.6 to 76.3	2	78125 to 610.4
	CR=1	64	2441.4 to 19.1		
16 t_{inst}	CR=0	16	4882.8 to 38.2	2	39062.5 to 305.2
	CR=1	64	1220.7 to 9.5		
64 t_{inst}	CR=0	16	1220.7 to 9.5	2	9765.6 to 76.3
	CR=1	64	305.2 to 2.4		

T_{inst}: Instruction cycle (affected by the clock mode)

* In main clock mode (SCS = 1), the maximum clock rate (CS1, CS0 = 11_B, 1 instruction cycle = 4/Fch) is selected by the system clock control register (SYCC).

Figure 17.1-4 Example for Baud Rate Calculation (When PWM Timer 1 is Selected)



Reference:

The baud rate is determined by the input clock specified by the clock division ratio register (SCS1, SCS0). For the input clock, the external clock or PWM timer 1 is selected. See Table 17.1-2 "Transfer Cycles and Transfer Rates When Baud Rate Generator Used" and Table 17.1-3 "Transfer Cycles and Transfer Rates When External Clock Used" for the calculation.

When the external clock is selected

1/39k baud = 1.6 µs (min.) × 16 (CR = 0)   (FCH = 10 MHz)

When PWM timer 1 is selected

1/78k baud = 0.4 µs (4/FCH) × 1 (SC11 = 0, SC10 = 0) × 1 (COMR1 = 0) × 2 × 16 (CR = 0)
(FCH = 10 MHz)
1/19531 baud = 0.4 µs (4/FCH) × 1 (SC11 = 0, SC10 = 0) × 1 (COMR1 = 3) × 2 × 16(CR = 0)
(FCH = 10 MHz)

See CHAPTER 8 "2-CHANNEL 8-Bit PWM TIMERS", for the PWM timer count clock cycle, PWM compare register setting value, and PWM timer output cycle.

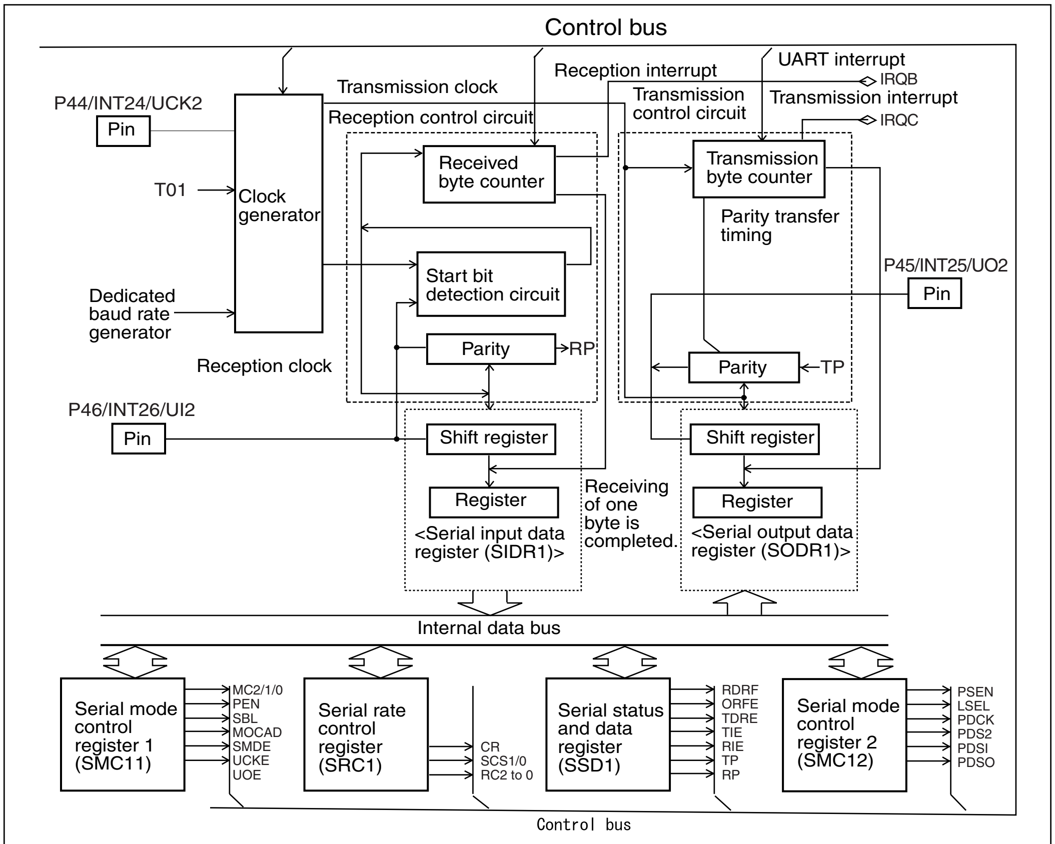
17.2 Configuration of the High-Speed UART

The high-speed UART consists of the following nine blocks:

- **Serial mode control register 1 (SMC11)**
- **Serial mode control register 2 (SMC12)**
- **Serial rate control register (SRC1)**
- **Serial status/data register (SSD1)**
- **Serial input data register (SIDR1)**
- **Serial output data register (SODR1)**
- **Clock generator**
- **Reception control circuit**
- **Transmission control circuit**

■ Block Diagram of High-Speed UART

Figure 17.2-1 Block Diagram of High-Speed UART



TO1: PWM timer 1 output of the 2-channel 8-bit PWM timer

- **Serial mode control register 1 (SMC11)**

  One of the registers that controls the operation mode of the high-speed UART. This register specifies whether parity is used, the stop bit count, the operation mode (data length), and whether synchronous or asynchronous mode is used, and enables serial clock output of the high-speed UART and serial data output.

- **Serial mode control register 2 (SMC12)**

  One of the registers that controls the data transfer rate (baud rate) of the high-speed UART. This register selects the input clock and sets the transfer rate for the baud rate generator.

- **Serial rate control register (SRC1)**

  SRC1 controls the data transfer rate (baud rate) of the high-speed UART. This register selects the input clock and sets the transfer rate for the baud rate generator.

- **Serial status/data register (SSD1)**

  SSD1 indicates the operational state of transmission or reception, errors occurring in the high-speed UART, and whether data transfer with parity bit or with bit 8 used for transmission data has been specified. This register also enables interrupts and can be used to set and check whether transmission is to be performed with a parity bit or with bit 8 used for transmission data.

- **Serial input data register (SIDR1)**

  Register for storing received data. Serial input is converted and stored in this register. For a data length of 9-bits, the leading bit is stored in the SSD1: RD8/RP bit.

- **Serial output data register (SODR1)**

  Register that sets transmission data. Data written to this register is converted to serial data and output. For a data length of 9-bits, the leading bit is stored in the SSD1: TD8/TP bit.

- **Clock generator**

  This circuit generates transmission or reception clock pulses using the dedicated baud rate generator, the external clock, and two-channel 8-bit PWM timer output.

- **Reception control circuit**

  This circuit consists of a received byte counter, start bit detection circuit, and received parity circuit.

  The received byte counter counts the bits of data received. It generates an interrupt request whenever a data item of the specified length has been received.

  The start bit detection circuit detects a start bit in the serial input signal. Upon detection of the start bit, this circuit writes data to the SIDR, applying shifts according to the specified transfer rate.

  The received parity circuit stores a parity bit in received data if transfer with parity is specified. However, if a data length of 9-bits is specified, the circuit stores the leading bit of the received data.

- **Transmission control circuit**

The transmission control circuit consists of a transmission byte counter and transmission parity circuit.

The transmission byte counter counts the bits of transmission data. It generates an interrupt request whenever a data item of the specified length has been transferred.

The transmission parity circuit generates a parity bit for data to be transferred if transmission with parity is specified. For a data length of 9-bits, the leading bit of the transmission data is stored.

- **Interrupt sources related to the high-speed UART**

Reception: IRQB

If interrupt requests are enabled (SSD1: RIE = 1), a reception interrupt request (IRQB) is generated when a data item of the specified length has been received correctly or when an overrun error or a framing error occurs during reception.

Transmission: IRQC

If transmission request interrupts are enabled (SSD1: TIE = 1), a transmission interrupt request (IRQC) is generated when data written to the SODR1 register has been transferred to the internal shift register and writing of the next item of data is allowed.

17.3 Pins of the High-Speed UART

This section describes the pins related to the high-speed UART and provides block diagrams of the pins.

■ Pins Related to the High-Speed UART

The pins related to the high-speed UART are the clock I/O pin (P44/INT24/UCK2), serial data output pin (P45/INT25/UO2), and serial data input pin (P46/INT26/UI2).

P44/INT24/UCK2:

The P44/INT24/UCK2 pin functions as a general-purpose I/O pin (P44), external interrupt pin (INT24), or clock I/O pin (hysteresis input) for the high-speed UART (UCK2). When clock output is enabled (SMC1: UCKE = 1), this pin functions as the clock I/O pin (UCK2) of the high-speed UART regardless of the value of the corresponding port direction register. At this time, do not select the external clock (SRC1: SCS1, SCS0 = other than 00_B). When using this pin as the clock input pin of the high-speed UART, disable clock output (SMC1: UCKE = 0) and set used input port as the use of this pin in the corresponding port direction register (DDR4: bit 4 = 0). Also select the external clock (SRC1: SCS1, SCS0 = 00_B).

P45/INT25/UO2:

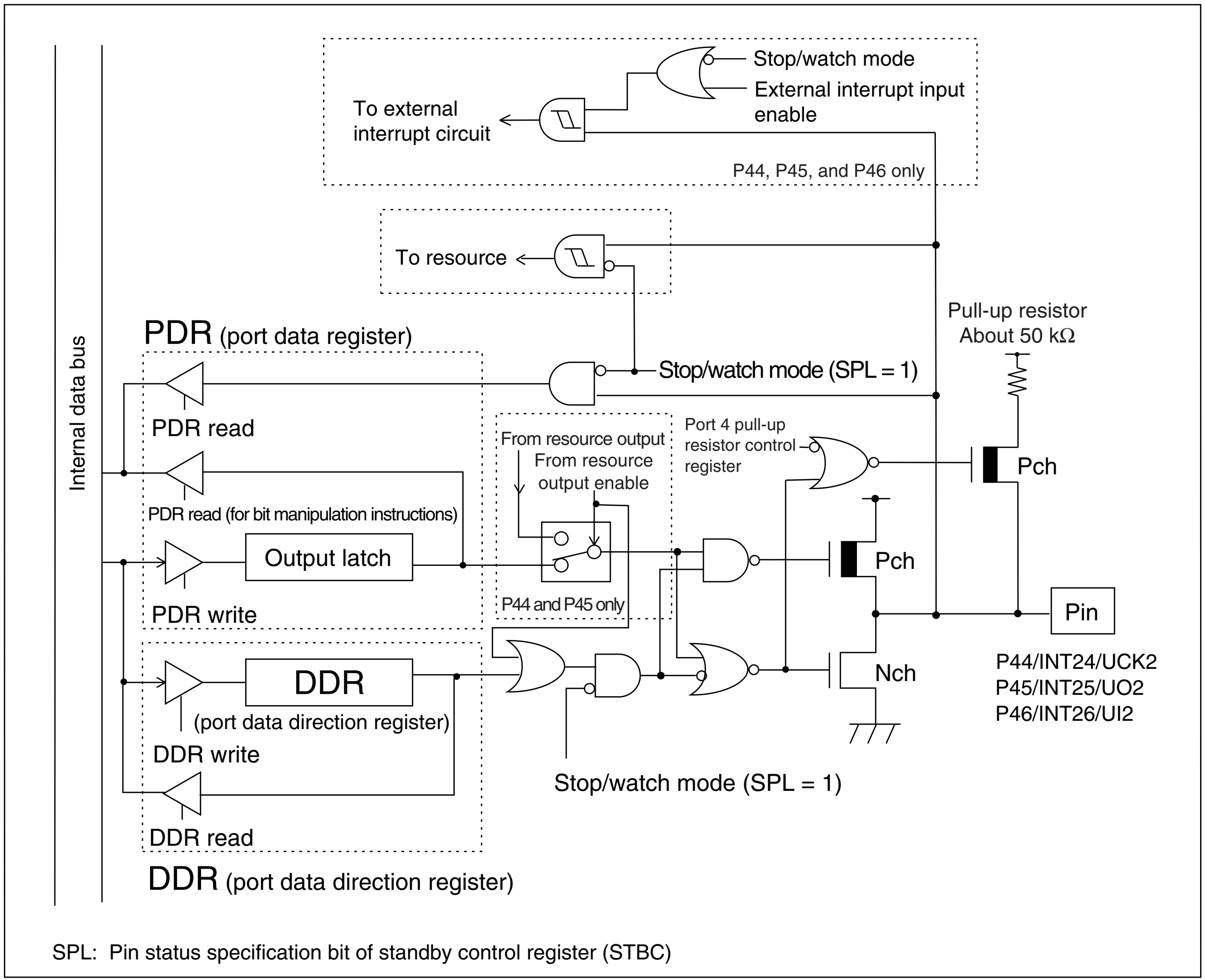
The P45/INT25/UO2 pin functions as a general-purpose I/O pin (P45), external interrupt pin (INT25), or serial data output pin for the high-speed UART (UO2). When serial data output is enabled (SMC11: UOE = 1), this pin functions as the serial data output pin (UO2) of the high-speed UART regardless of the value of the corresponding port direction register.

P46/INT26/UI2:

The P46/INT26/UI2 pin functions as a general-purpose I/O pin (P46), external interrupt pin (INT26), or serial data input pin (hysteresis input) for the high-speed UART (UI2). To use this pin as the serial data input pin for the high-speed UART, set the corresponding port data register for the input port (DDR4:bit 6 = 0).

■ Block Diagrams of Pins Related to the High-Speed UART

Figure 17.3-1 Block Diagram of Pins Related to the High-Speed UART



Stop/watch mode
External interrupt input enable
To external interrupt circuit
P44, P45, and P46 only
To resource
Pull-up resistor
About 50 kΩ
PDR (port data register)
Internal data bus
Stop/watch mode (SPL = 1)
PDR read
From resource output
From resource output enable
Port 4 pull-up resistor control register
Pch
PDR read (for bit manipulation instructions)
Output latch
Pch
PDR write
P44 and P45 only
Pin
DDR
Nch
P44/INT24/UCK2
P45/INT25/UO2
P46/INT26/UI2
(port data direction register)
DDR write
Stop/watch mode (SPL = 1)
DDR read
DDR (port data direction register)

SPL: Pin status specification bit of standby control register (STBC)

Reference:

When "pull-up resistor available" is selected using the port 4 pull-up resistor control register the pins are set to "H" (pulled up) level, not the high impedance state, in stop mode or watch mode (STBC:SPL = 1). During a reset, however, pull-up is disabled and the pins are set to Hi-z.

17.4 Registers of the High-Speed UART

This section describes the registers related to the high-speed UART.

■ High-Speed UART Registers

Figure 17.4-1 High-Speed UART Registers

SMC11 (serial mode control register 1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 2H	PEN	SBL	MC2	MC1	MC0	SMDE	UCKE	UOE	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

SMC12 (serial mode control register 2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 6H	–	–	PSEN	LSEL	PDCK	PDS2	PDS1	PDS0	--100001B
			R/W	R/W	R/W	R/W	R/W	R/W	

SRC1 (serial rate control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 3H	–	–	CR	SCS1	SCS0	RC2	RC1	RC0	--011000B
			R/W	R/W	R/W	R/W	R/W	R/W	

SSD1 (serial status/data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 4H	RDRF	ORFE	TDRE	TIE	RIE	–	TD8/TP	RD8/RP	00100-1XB
	R	R	R/W	R/W	R/W		R/W	R	

SIDR1 (serial input data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 5H									XXXXXXXXB
	R	R	R	R	R	R	R	R	

SODR1 (serial output data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 5H									XXXXXXXXB
	W	W	W	W	W	W	W	W	

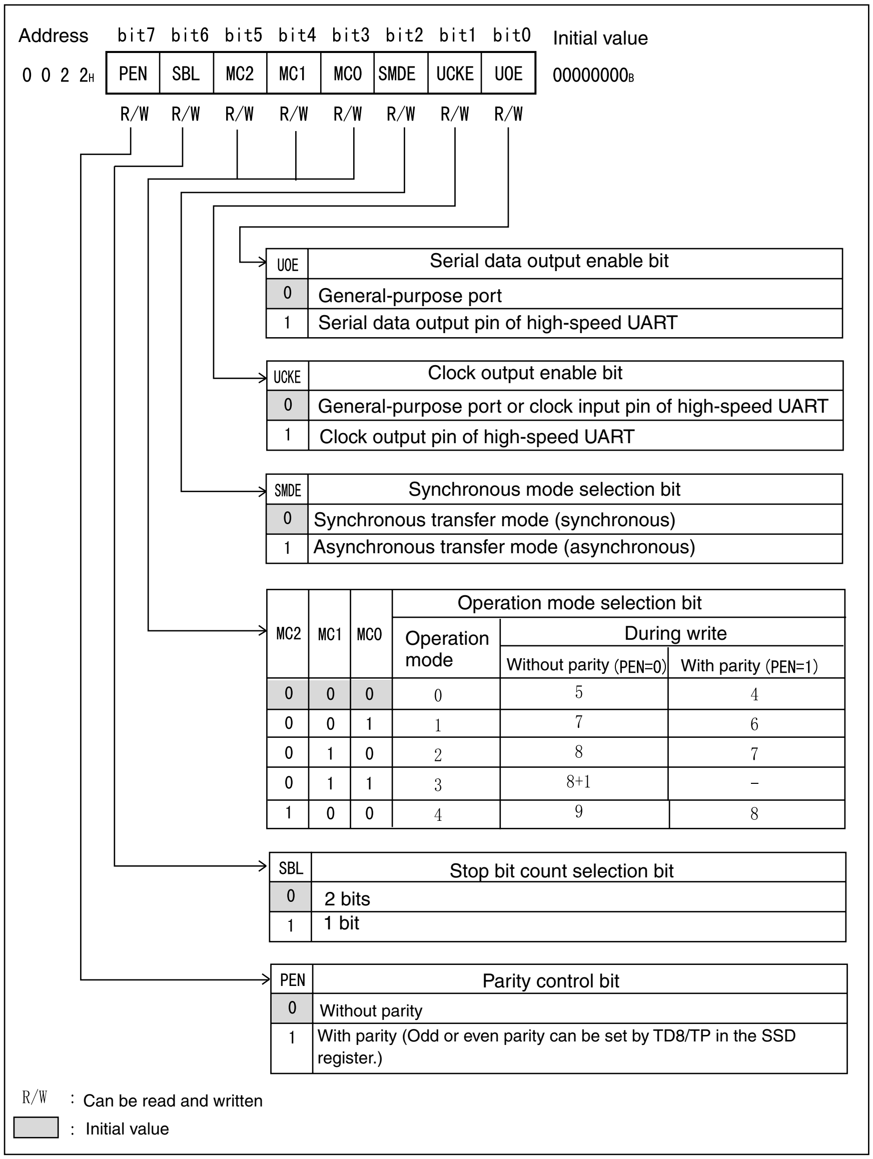
R/W	: Read/Write enabled
R	: Read only
W	: Write only
—	: Unused
X	: Undefined

17.4.1 Serial Mode Control Register 1 (SMC11)

Serial mode control register 1 (SMC11) specifies whether parity is used, defines the stop bit length, sets the operation mode (data length), sets synchronous or asynchronous mode, and enables/disables serial clock output and serial data output of the high-speed UART.

■ Serial Mode Control Register 1 (SMC11)

Figure 17.4-2 Serial Mode Control Register 1 (SMC11)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 2H	PEN	SBL	MC2	MC1	MC0	SMDE	UCKE	UOE	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

UOE	Serial data output enable bit
0	General-purpose port
1	Serial data output pin of high-speed UART

UCKE	Clock output enable bit
0	General-purpose port or clock input pin of high-speed UART
1	Clock output pin of high-speed UART

SMDE	Synchronous mode selection bit
0	Synchronous transfer mode (synchronous)
1	Asynchronous transfer mode (asynchronous)

MC2	MC1	MC0	Operation mode selection bit		
			Operation mode	During write	
				Without parity (PEN=0)	With parity (PEN=1)
0	0	0	0	5	4
0	0	1	1	7	6
0	1	0	2	8	7
0	1	1	3	8+1	–
1	0	0	4	9	8

SBL	Stop bit count selection bit
0	2 bits
1	1 bit

PEN	Parity control bit
0	Without parity
1	With parity (Odd or even parity can be set by TD8/TP in the SSD register.)

R/W : Can be read and written

▨ : Initial value

Table 17.4-1 Functions of the Serial Mode Control Register 1 (SMC11) Bits

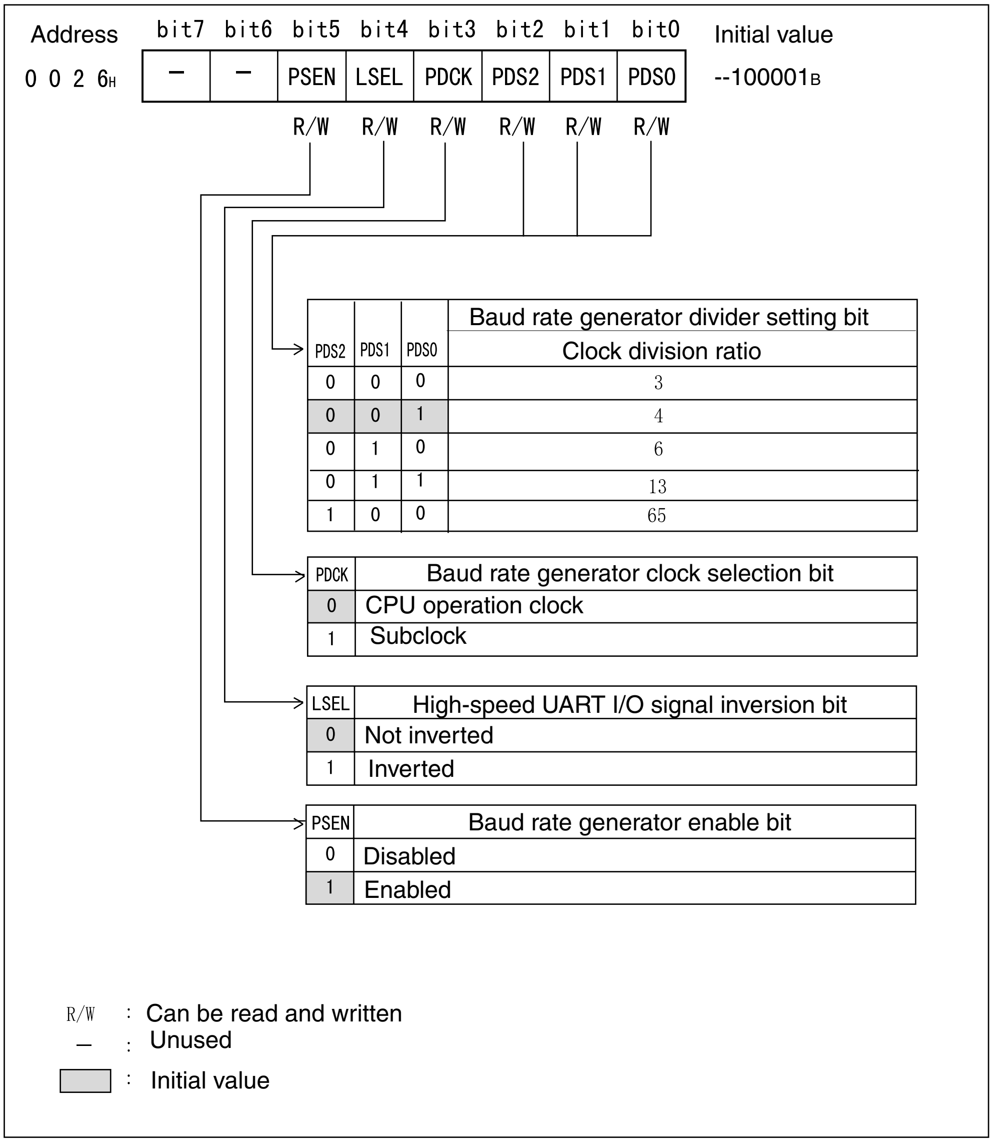
Bit name		Function
bit7	PEN: Parity enable bit	• For serial data I/O, this bit specifies whether a parity bit is added during transmission and whether parity bits are detected during reception.
bit6	SBL: Stop bit count selection bit	• This bit specifies the stop bit count of transmission data. **Note:** During reception, only the first stop bit is detected; subsequent stop bits are ignored.
bit5 bit4 bit3	MC2, MC1, MC0: Operation mode selection bits	• These bits select the operation mode (data length). • Six types of data length can be selected in combination with a parity bit.
bit2	SMDE: Synchronous mode selection bit	• This bit selects synchronous or asynchronous transfer. • When this bit is "0", synchronous transfer mode is selected. When this bit is "1", asynchronous transfer mode is selected.
bit1	UCKE: Clock output enable bit	• This bit controls serial clock I/O. • When this bit is "0", the P44/INT24/UCK2 pin functions as the serial clock input pin. When this bit is "1", the pin functions as the serial clock output pin. **Note:** • When the UCK2 pin is set to function as serial clock input (UCKE = 0), set the P44/INT24/UCK2 pin to be an input port. Alternatively, select use of the external clock with the clock input selection bit (SRC1: SCS1, SCS0 = 00_B). • When the UCKE pin is set to function as a serial clock output (UCKE = 0), select a clock other than the external clock (SRC1:SCS1, SCS0 = other than 00_B). **Reference:** When the UCK2 pin is set to function as a serial clock output (UCKE = 1), the pin functions as the UCK2 output pin regardless of the state of the general-purpose port (P44).
bit0	UOE: Serial data output enable bit	• When this bit is "0", the P45/INT25/UO2 pin functions as a general-purpose port (P45). When this bit is "1", it functions as a serial data output pin (UO). **Reference:** When serial clock output is specified (UOE = 1), the pin functions as the UO2 pin regardless of the state of the general-purpose port (P45).

17.4.2 Serial Mode Control Register 2 (SMC12)

Serial mode control register 2 (SMC12) enables/disables the baud rate generator, specifies whether the output of the high-speed UART I/O signal pins is to be inverted, and sets the input clock division ratio of the baud rate generator.

■ Serial Mode Control Register 2 (SMC12)

Figure 17.4-3 Serial Mode Control Register 2 (SMC12)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 6H	–	–	PSEN	LSEL	PDCK	PDS2	PDS1	PDS0	--100001B
			R/W	R/W	R/W	R/W	R/W	R/W	

			Baud rate generator divider setting bit
PDS2	PDS1	PDS0	Clock division ratio
0	0	0	3
0	0	1	4
0	1	0	6
0	1	1	13
1	0	0	65

PDCK	Baud rate generator clock selection bit
0	CPU operation clock
1	Subclock

LSEL	High-speed UART I/O signal inversion bit
0	Not inverted
1	Inverted

PSEN	Baud rate generator enable bit
0	Disabled
1	Enabled

R/W : Can be read and written
– : Unused
▨ : Initial value

Table 17.4-2 Functions of the Serial Mode Control Register 2 (SMC12) Bits

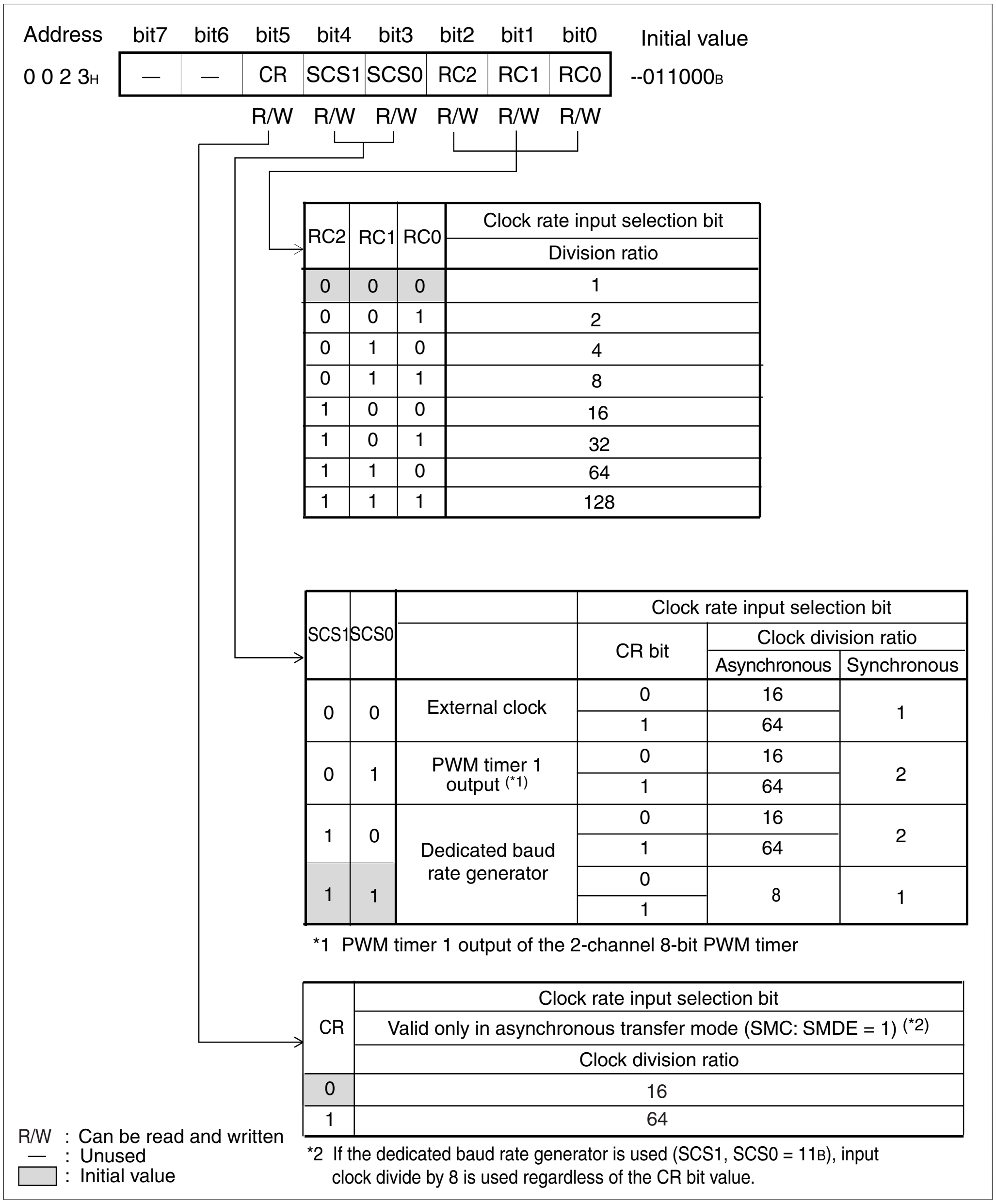
Bit name		Function
bit7 bit6	Unused	• Value read is undefined. • Writing has no effect on the operation.
bit5	PSEN: Baud rate generator enable bit	• This bit enables or disables the baud rate generator.
bit4	LSEL: High-speed UART I/O signal inversion bit	• This bit specifies whether to invert the high-speed UART I/O data.
bit3	PDCK: Clock selection bit	Baud rate generator enable bit • This bit selects a division ratio of the divider, which is located in the stage preceding the baud rate generator. **Note:** For the CPU operation clock, "F_{CH}/2" is used in main watch mode (SYCC:SCS = 1), and "F_{CL}" is used in subwatch mode (SYCC:SCS = 0).
bit2 bit1 bit0	PDS2, PDS1, PDS0: Baud rate generator divider setting bits	• Bits 2, 1, and 0 select the division ratio of the divider, which is located in the stage preceding the baud rate generator. • In clock synchronous mode, any divide-by-three setting is prohibited. • These bits enable/disable the baud rate generator.

17.4.3 Serial Rate Control Register (SRC1)

The serial rate control register (SRC1) controls the data transfer rate (baud rate) in asynchronous mode. This register selects an input clock and sets the transfer rate for the baud rate generator.

■ Serial Rate Control Register (SRC1)

Figure 17.4-4 Serial Rate Control Register (SRC1)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 3$_H$	—	—	CR	SCS1	SCS0	RC2	RC1	RC0	--011000$_B$
			R/W	R/W	R/W	R/W	R/W	R/W	

RC2	RC1	RC0	Clock rate input selection bit Division ratio
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

SCS1	SCS0		Clock rate input selection bit		
			CR bit	Clock division ratio	
				Asynchronous	Synchronous
0	0	External clock	0	16	1
			1	64	
0	1	PWM timer 1 output (*1)	0	16	2
			1	64	
1	0	Dedicated baud rate generator	0	16	2
			1	64	
1	1		0	8	1
			1		

*1 PWM timer 1 output of the 2-channel 8-bit PWM timer

CR	Clock rate input selection bit Valid only in asynchronous transfer mode (SMC: SMDE = 1) (*2) Clock division ratio
0	16
1	64

R/W : Can be read and written
— : Unused
▨ : Initial value

*2 If the dedicated baud rate generator is used (SCS1, SCS0 = 11$_B$), input clock divide by 8 is used regardless of the CR bit value.

Table 17.4-3 Functions of the Serial Rate Control Register (SRC1) Bits

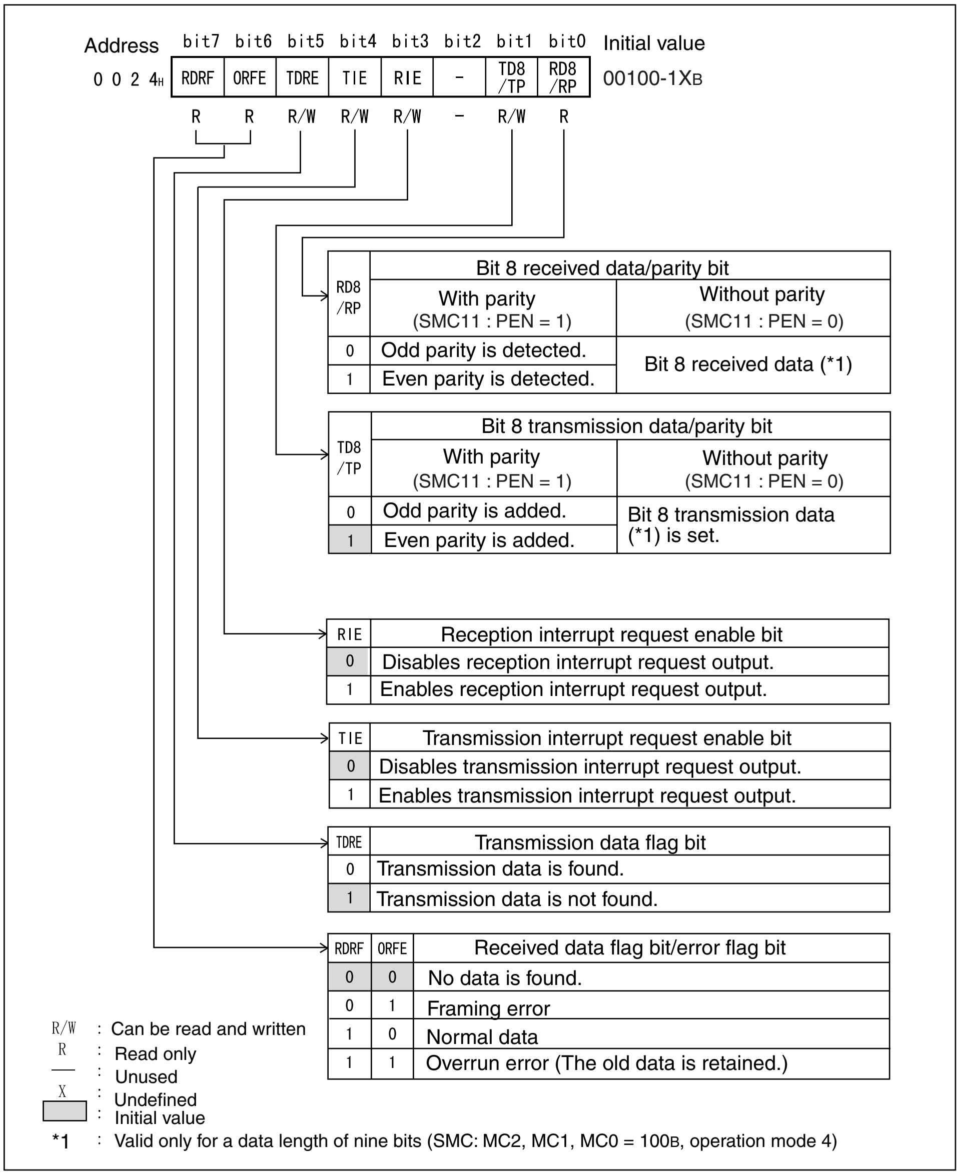
Bit name		Function
bit7 bit6	Unused	• The read value is undefined. • Writing has no effect on operation.
bit5	CR: Clock rate selection bit	• This bit selects the clock rate in asynchronous transfer mode. If the dedicated baud rate generator is used (SCS1, SCS0 = 11_B), divide by 8 is used regardless of the CR bit value. • When the external clock or two-channel 8-bit PWM timer output is specified as the clock input, the baud rate becomes 1/16 or 1/64 of the clock frequency, depending on the CR value. • This bit has no meaning in synchronous transfer mode.
bit4 bit3	SCS1, SCS0: Clock input selection bits	• These bits select clock input. • The clock input can be selected from the external clock (UCK pin), two-channel 8-bit PWM timer (PWM timer 1 output), and dedicated baud rate generator.
bit2 bit1 bit0	RC2, RC1, RC0: Baud rate selection bits	• In asynchronous transfer mode, eight baud rates can be selected. In synchronous transfer mode, six baud rates can be selected. • These bits are valid only when the dedicated baud rate generator is used for clock input. These bits are invalid when the external clock or two-channel 8-bit PWM timer 1 output is used.

17.4.4 Serial Status/Data Register (SSD1)

The serial status/data register (SSD1) indicates the transmission/reception status, error status, and received parity data or bit 8 received data in the high-speed UART. SSD1 also enables/disables interrupts, and can be used to set and check transmission parity data or bit 8 transmission data.

■ Serial Status/Data Register (SSD1)

Figure 17.4-5 Serial Status/Data Register (SSD1)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 4H	RDRF	ORFE	TDRE	TIE	RIE	-	TD8/TP	RD8/RP	00100-1XB
	R	R	R/W	R/W	R/W	-	R/W	R	

RD8/RP	Bit 8 received data/parity bit	
	With parity (SMC11 : PEN = 1)	Without parity (SMC11 : PEN = 0)
0	Odd parity is detected.	Bit 8 received data (*1)
1	Even parity is detected.	

TD8/TP	Bit 8 transmission data/parity bit	
	With parity (SMC11 : PEN = 1)	Without parity (SMC11 : PEN = 0)
0	Odd parity is added.	Bit 8 transmission data (*1) is set.
1	Even parity is added.	

RIE	Reception interrupt request enable bit
0	Disables reception interrupt request output.
1	Enables reception interrupt request output.

TIE	Transmission interrupt request enable bit
0	Disables transmission interrupt request output.
1	Enables transmission interrupt request output.

TDRE	Transmission data flag bit
0	Transmission data is found.
1	Transmission data is not found.

RDRF	ORFE	Received data flag bit/error flag bit
0	0	No data is found.
0	1	Framing error
1	0	Normal data
1	1	Overrun error (The old data is retained.)

R/W : Can be read and written
R : Read only
— : Unused
X : Undefined
▨ : Initial value
*1 : Valid only for a data length of nine bits (SMC: MC2, MC1, MC0 = 100B, operation mode 4)

Table 17.4-4 Functions of the Serial Status/Data Register (SSD1) Bits

Bit name		Function
bit7	RDRF: Received data flag bit	• This bit indicates the state of the serial input data register (SIDR1). • Setting this bit to 1 start to read the SSD1 register and then reading the SIDR1 register clears the received data flag bit (RDRF). • When this bit and the reception interrupt request enable bit (RIE) are both "1", a reception interrupt request is output. • This bit is a read-only bit. Writing does not change the value of this bit and has no effect on operation.
bit6	ORFE: Overrun error flag bit	• This bit is a flag for indicating whether an overrun or a framing error has occurred. • If an error occurs (ORFE = 1), no data is transferred from the reception shift register to the SIDR1 register. Consequently, the RDRF bit is not set in this case. • When this bit is "1", reading the SSD1 register and SIDR1 register, in this sequence, clears this bit to "0". • When this bit and the reception interrupt request enable bit (RIE) are both "1", a reception interrupt request is output. • This bit is a read-only bit. Writing does not change the value of this bit and has no effect on operation.
bit5	TDRE: Transmission data flag bit	• This bit is a flag indicating the state of the serial output data register (SODR1). • When this bit is "1", reading the SSD1 register then writing data in the SODR1 register causes the data to be output to the serial data output pin (U02). • When "1" is written to this bit and the transmission interrupt request enable bit (TIE), a transmission interrupt request is output.
bit4	TIE: Transmission interrupt request enable bit	• This bit enables transmission interrupt requests to the CPU. • When both this bit and the transmission data flag bit (TDRE) are "1", a transmission interrupt request is output.
bit3	RIE: Reception interrupt request enable bit	• This bit enables reception interrupt requests to the CPU. • When both this bit and the received data flag bit (RDRF) are "1", a reception interrupt request is output. • When both this bit and the overrun error flag bit (ORFE) are "1", a reception interrupt request is output when an error occurs.
bit2	Unused	• The read value of this bit is undefined. • Writing to this bit has no effect on operation.
bit1	TD8/TP: bit 8 transmission data/parity bit	• This bit is treated as bit 8 of the SODR1 register in operation mode 3 without parity (data length for transmission or reception: 9-bits). This bit has no meaning in operation modes other than mode 3 without parity. • This bit specifies which parity, even or odd, is selected for data transmission when a parity bit is attached.
bit0	RD8/RP: bit 8 received data/ parity bit	• This bit is treated as bit 8 of the SIDR1 register in operation mode 3 without parity (data length for transmission or reception: 9-bits). This bit has no meaning in operation modes other than mode 3 without parity. • This bit indicates detection of parity in received data if a parity bit is attached.

17.4.5 Serial Input Data Register (SIDR1)

The serial input data register (SIDR1) is used for inputting (receiving) serial data.

■ Serial Input Data Register (SIDR1)

Figure 17.4-6 "Serial Input Data Register (SIDR1)" shows the bit configuration of the serial input data register.

Figure 17.4-6 Serial Input Data Register (SIDR1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0025H									XXXXXXXXB
	R	R	R	R	R	R	R	R	

R : Read only
X : Undefined

The SIDR1 is used for storing received data. The serial data signal sent to the serial data input pin (UI pin) is first converted in the shift register and then stored in this register.

❍ In operation mode 0, 1, 2 or 4

When the received data has been successfully stored in SIDR1, the receive data flag bit (SSD1: RDRF) is set to "1". If reception interrupt requests are enabled, a reception interrupt will occur. If a check of the SSD1: RDRF bit during interrupt processing or by the program indicates that the received data is stored in SIDR1, read SSD1 and SIDR1, then clear the SSD1: RDRF flag.

❍ In operation mode 3

Both RDRF and ORFE are set when data transfer ends, the last data bit (D8) is "1", and the stop bit for the last transfer has been detected. If a framing error is detected, however, the flags are set irrespective of the last data bit.

An interrupt request to the CPU occurs when these flags are set and the data input bit is "1".

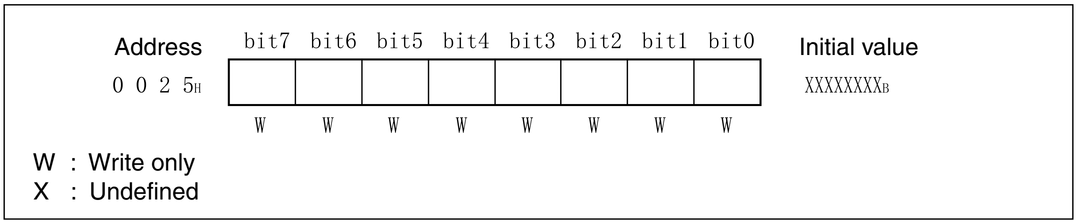
17.4.6 Serial Output Data Register (SODR1)

The serial output data register (SODR1) is used for outputting (transmitting) serial data.

■ Serial Output Data Register (SODR1)

Figure 17.4-7 "Serial Output Data Register (SODR1)" shows the bit configuration of the serial output data register.

Figure 17.4-7 Serial Output Data Register (SODR1)



When data to be transmitted is written to this register after the SSD1 register has been read in the transmission-enabled state, the data is transferred to the shift register for transmission, converted to serial data, and transmitted from the serial data output pin (UO pin).

When the transmission data is written to SODR1, the transmission data flag bit is set to "0". After the transmission data is transferred to the shift register for transmission, the transmission data flag bit is set to "1" so that the next transmission data item can be written to the register. If interrupt requests are enabled at this time, an interrupt occurs. The transmission data next item can be written when a transmission interrupt is generated or when the transmission data flag bit is set to "1".

17.5 High-Speed UART Interrupts

The high-speed UART has three flags related to interrupts: the error flag bit (SSD1: ORFE), received data flag bit (SSD1: RDRF), and transmission data flag bit (SSD1: TDRE). These flags are used for the following two interrupt sources:

- **Received data is transferred from the shift register for reception to the serial input data register (SIDR1) (reception interrupt)**
- **Transmission data is transferred from the serial output data register (SODR1) to the shift register for transmission (transmission interrupt)**

■ Transmission Interrupt

When output data is written to SODR1 after SSD1 is read, the data written to SODR1 is transferred to the shift register for internal transmission. When the register is ready to accept the next data item, the TDRE bit is set to "1". If transmission interrupts are enabled (SSD1: TIE = 1), an interrupt request is issued to the CPU (IRQC).

■ Reception Interrupt

❍ In operation mode 0, 1, 2, or 4

After data has been input up to the stop bit, the RDRF bit is set to "1". If an overrun error or framing error occurs, the ORFE bit is set to "1".

These bits are set when the stop bit is detected. If reception interrupts are enabled (SSD1: RIE = 1), an interrupt request is issued to the CPU (IRQB).

❍ In operation mode 3

Both the SSD1: RDRF and ORFE flags are set when the last data bit (D8) is "1" because data transfer has been completed and the stop bit for the last transfer has been detected. If a framing error is detected, however, these flags are set regardless of the last data bit.

An interrupt request to the CPU is generated when these flags are set and the data input bit is "1".

■ Register and Vector Table Address for High-Speed UART Interrupts

Table 17.5-1 Register and Vector Table Address for High-Speed UART Interrupts

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Setting bits		Upper part of the address	Lower part of the address
IRQB	ILR3 (007D$_H$)	LB1 (bit7)	LB0 (bit6)	FFE4$_H$	FFE5$_H$
IRQC	ILR4 (007E$_H$)	LC1 (bit1)	LC0 (bit0)	FFE2$_H$	FFE3$_H$

For an explanation of the operation of interrupts, see 3.4.2 "Interrupt Processing "

17.6 Operation of the High-Speed UART

This section describes the operation of the high-speed UART.
The high-speed UART has standard serial communication functions (in operation modes 0, 1, 2, 3, and 4).

■ Operation of the High-Speed UART

❍ Operation modes

The high-speed UART has five operation modes. Modes 0, 1, 2, and 4 support ordinary serial transfer. Whether a parity bit is to be attached can be selected and the data length for transfer can be selected from four bits to 9-bits (see Table 17.1-1 "Operation Modes of the High-Speed UART").

Operation mode 3 supports serial transfer of eight-bit data and one slave bit. This mode can be used for connection of two or more slave CPUs to one host CPU.

❍ Transfer data format

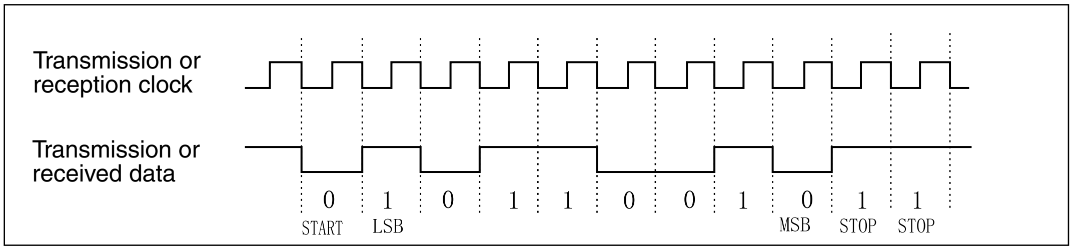
The high-speed UART can only handle data in non-return-to-zero (NRZ) format.

Data transfer starts with the start bit ("L" level), then data of the data length specified in bits is transferred using the LSB-first method, and data transfer ends with the stop bit ("H" level).

During asynchronous transfer, the relationship between the serial clock and serial I/O signal differs from that shown in Figure 17.6-1 "Transfer Data Format".

Figure 17.6-1 "Transfer Data Format" shows the relationship between the transmission or reception clock and data when no parity is attached; operation mode 1, two stop bits, and synchronous transfer are selected; and the transfer data is 01001101_B (8-bits).

Figure 17.6-1 Transfer Data Format

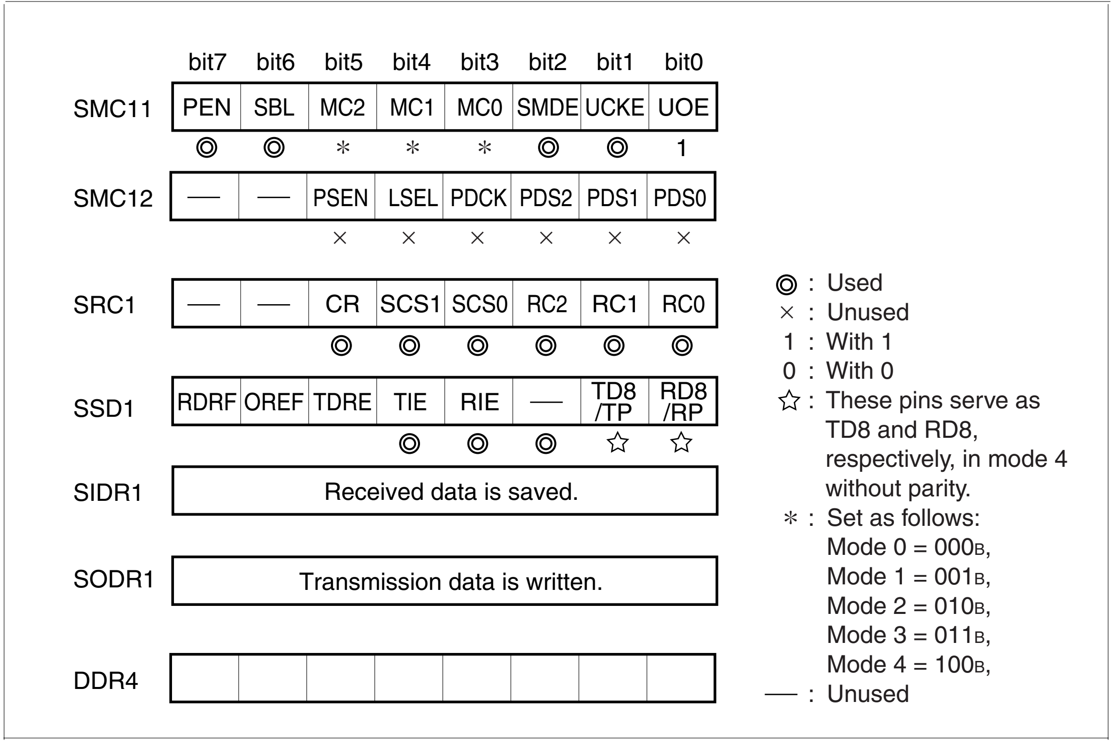

Transmission or reception clock
Transmission or received data
0
1
0
1
1
0
0
1
0
1
1
START
LSB
MSB
STOP
STOP

17.7 Operation of Operation Modes 0, 1, 2, and 4

Operation modes 0, 1, 2, 3, and 4 support the standard serial communication function.

■ Explanation of Operation Modes 0, 1, 2, and 4

Figure 17.7-1 Operation Modes 0, 1, 2, 3, and 4



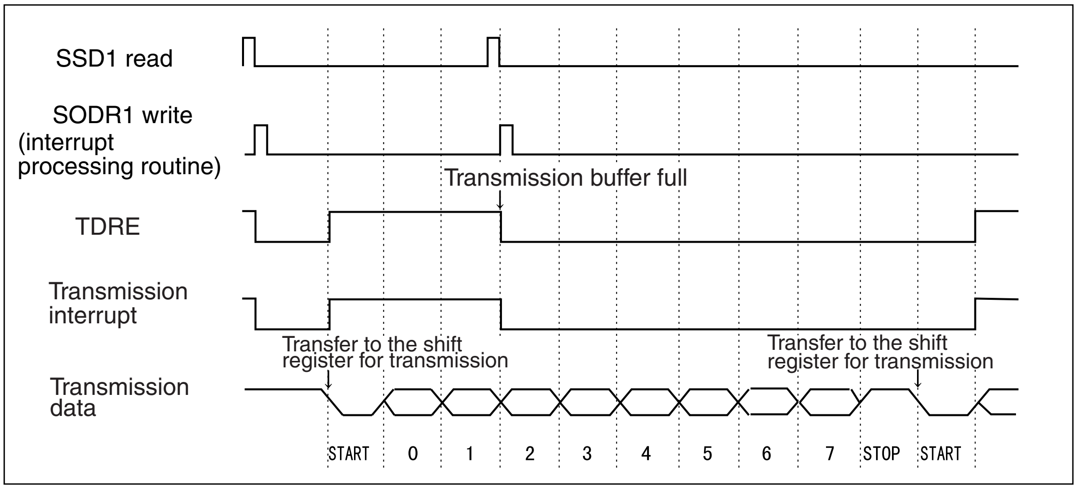
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SMC11	PEN	SBL	MC2	MC1	MC0	SMDE	UCKE	UOE
	◎	◎	*	*	*	◎	◎	1
SMC12	—	—	PSEN	LSEL	PDCK	PDS2	PDS1	PDS0
			×	×	×	×	×	×
SRC1	—	—	CR	SCS1	SCS0	RC2	RC1	RC0
			◎	◎	◎	◎	◎	◎
SSD1	RDRF	OREF	TDRE	TIE	RIE	—	TD8 /TP	RD8 /RP
				◎	◎	◎	☆	☆
SIDR1	Received data is saved.							
SODR1	Transmission data is written.							
DDR4								

◎ : Used
× : Unused
1 : With 1
0 : With 0
☆ : These pins serve as TD8 and RD8, respectively, in mode 4 without parity.
* : Set as follows:
Mode 0 = 000_B,
Mode 1 = 001_B,
Mode 2 = 010_B,
Mode 3 = 011_B,
Mode 4 = 100_B,
— : Unused

❍ Transmission operation

When transmission data is written to SODR1 after SSD1 is read, the data is transferred to the shift register for transmission and parallel-serial conversion starts. The converted transmission data is output from the serial data output pin beginning with the LSB bit (LSB first). When writing of the next item of data is enabled, the TDRE bit is set to "1". If transmission interrupts are enabled (SSD1: TIE = 1), an interrupt request to the CPU is issued. Figure 17.7-2 "Transmission Operation in Operation Modes 0, 1, 2, 3, and 4" shows the transmission operation performed when operation mode 2 is set, no parity is added, and one stop bit is used.

Figure 17.7-2 Transmission Operation in Operation Modes 0, 1, 2, 3, and 4


SSD1 read
SODR1 write (interrupt processing routine)
Transmission buffer full
TDRE
Transmission interrupt
Transfer to the shift register for transmission
Transfer to the shift register for transmission
Transmission data
START
0
1
2
3
4
5
6
7
STOP
START

❍ Reception operation (modes 0, 1, 2, and 4)

When received data is sent to the serial data input pin, the data is converted from serial to parallel format using the internal shift register for reception. When data has been sent successfully up to the stop bit, the data in the internal shift register is transferred to the SIDR1 register and the SSD1: RDRF bit is set to "1".

If an overrun or framing error occurs, the received data is not transferred to the SIDR1 register and the SSD1: ORFE bit is set to "1".

The SSD1: RDRF and ORFE bits are set when data reception is completed and the last stop bit is detected. If reception interrupts are enabled (SSD1: RIE = 1), an interrupt request (IRQB) to the CPU is generated. When the RDRF bit is set, the received data has been transferred to the SIDR1 register.

Figure 17.7-3 "Reception Operation in Modes 0, 1, 2 and 4" to Figure 17.7-5 "Operation When a Framing Error Occurs in Operation Modes 0, 1, 2, and 4" show the reception operation performed when operation mode 0, 1, 2, and 4 are set, no parity is attached, and one stop bit is set.

Figure 17.7-3 Reception Operation in Modes 0, 1, 2 and 4

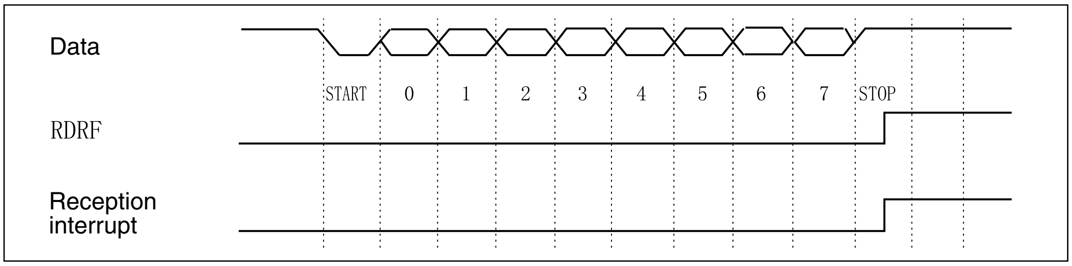


Figure 17.7-4 Operation When an Overrun Error Occurs in Operation Modes 0, 1, 2, and 4

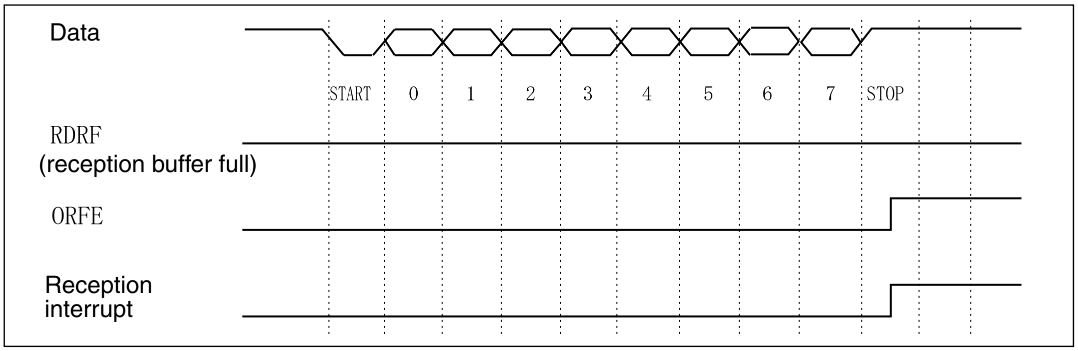
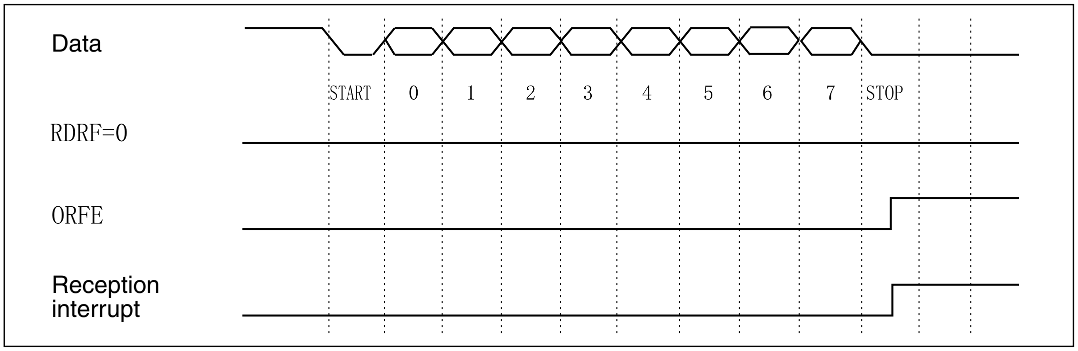


Figure 17.7-5 Operation When a Framing Error Occurs in Operation Modes 0, 1, 2, and 4



Note:

After initialization has been canceled by a reset, a period of 11 shift clocks is required for initialization of the internal control section. To perform the initialization, the microprocessor sends dummy data "FF_H" at the UART setting baud rate. If UO2 pin output enable is set in the SMC11 register during initialization, the start bit of the dummy data is output. Thus, UO2 pin output enable must be set after 12 shift clocks of the initial baud rate have elapsed after a reset.

The dummy data is transmitted only once at UART initialization after a reset. No dummy data is sent unless a reset is performed again. The initial value of the SSD1 register described in the manual is the value set after the completion of UART initialization.

17.8 Operation of Operation Mode 3

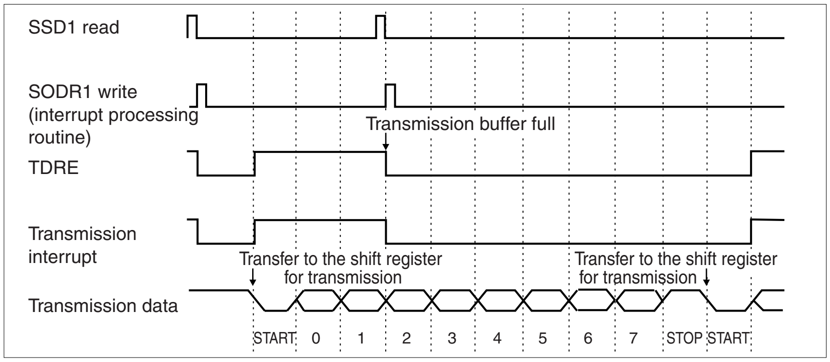
Operation mode 3 applies to cases where one host CPU connects to multiple slave CPUs

■ Description of high-speed UART operation mode 3

❍ Transmission operation

After data is read from the SSD1 register and written into the SODR1 register, the data is transferred to the shift register for transmission and parallel-serial conversion starts. The converted transmission data is output from the serial data output pin beginning with the LSB bit (LSB first). When writing of the next item of data is enabled, the TDRE bit is set to "1". If the transmission interrupts are enabled (SSD1:TIE = 1), an interrupt request is issued to the CPU. Figure 17.8-1 "Transmission Operation in Operation Mode 3" shows the transmission operation that is performed when operation mode 0 is set, no parity is added, and two stop bits are used.

Figure 17.8-1 Transmission Operation in Operation Mode 3



❍ Reception operation

When data is received at the serial data input pin, the data is converted from serial to parallel using the internal shift register for reception. Once the data has been received successfully up to the stop bit, the data in the internal shift register is transferred to the SIDR1 register and the SSD1:RDRF bit is then set to "1".

If an overrun or framing error occurs, the received data is not transferred to the SIDR1 register and the SSD1:ORFE bit is set to "1".

The flags of the SSD1:RDRF and ORFE bits are set when received data is successfully transferred with the last data bit set to "1" and the last stop bit detected. If, however, a framing error occurs, the flags are set regardless of the last data bit. A CPU interrupt request is generated when the flags are set and the input data becomes "1".

If reception interrupts have been enabled (SSD1:RIE = 1), a CPU interrupt request (IRQB) is generated. When the RDRF bit has been set, the received data is transferred to the SIDR1 register. Figure 17.8-2 "Reception in Operation Mode 3" to Figure 17.8-4 "Operation for a Framing Error in Operation Mode 3" illustrate reception in operation mode 3.

Figure 17.8-2 Reception in Operation Mode 3

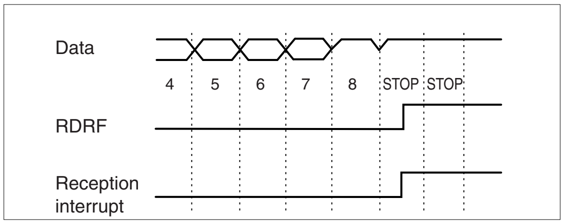


Figure 17.8-3 Operation for an Overrun Error in Operation Mode 3

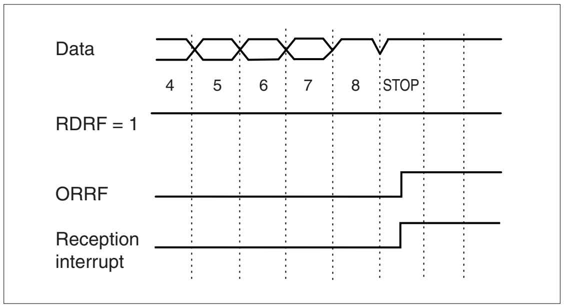
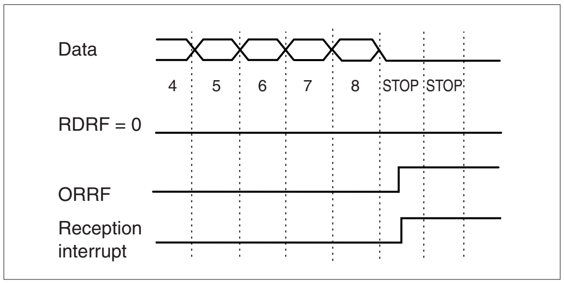


Figure 17.8-4 Operation for a Framing Error in Operation Mode 3



Note:

Operation of UART initialization

This UART requires a period of 11 shift clocks (after supply of the clock for generating a baud rate is started) to initialize. To perform initialization, the microprocessor sends dummy data "FF_H" at the UART setting baud rate. Setting UO2 output enable in the SMC11 register during the initialization, the start bit of the dummy data is output. Thus, UO2 output enable must be set after 12 shift clocks of a specified baud rate have elapsed after the supply of the clock for generating a baud rate to the UART is started.

The dummy data is transmitted only once upon UART initialization after a reset. No dummy data is sent unless another reset is issued. The initial value of the SSD1 register described in the manual is the value that is set after the completion of UART initialization.

- Timing at which the supply of the clock for generating a baud rate is started:
  - Case in which a dedicated baud rate generator is being used: When operation of the baud rate generator is allowed (When "1" is written into the PSEN bit of the SMC12 register)
  - Case in which the internal timer is being used: When the PWM timer 1 counter starts operating
  - Case in which an external clock is being used: When input of the external clock is started

17.9 Program Example of the UART

This section shows a program example for the UART.

■ Program Example of the UART

❍ Processing specifications

- Serial transfer, transmission, and reception operations are performed using the communication function of the UART.
- P44/INT24/SCK2, P45/INT25/SO2, and P46/INT26/UI2 are used for communication.
- The transfer rate is set to 300 baud by the internal baud rate generator.
- 13_H is transmitted from the UO pin to trigger the operation with an interrupt.
- The baud rate indicated is that used when the main clock oscillation frequency (F_{CH}) is set to 10 MHz and the main clock is set to the fastest (instruction cycle = $4/F_{CH}$) (1/300 baud = $8320t_{inst}$).

❍ Coding example

```
PDR4    EQU     0010H             ;Address of port data register
DDR4    EQU     0011H             ;Address of port data direction register
SMC11   EQU     0022H             ;Address of serial mode control register
SRC1    EQU     0023H             ;Address of serial rate control register
SSD1    EQU     0024H             ;Address of serial status/data register
SIDR1   EQU     0025H             ;Address of serial input data register
SODR1   EQU     0025H             ;Address of serial output data register
ILR3    EQU     007DH             ;Address of interrupt level setting register
INT_V   DSEG    ABS               ;[DATA SEGMENT]
        ORG     0FFE4H
IRQB    DW      WARI              ;Set interrupt vector.
INT_V   ENDS
---------Main program------------------------------------------------------------
        CSEG                      ;[CODE SEGMENT]
        :
        CLRI                      ;Disable interrupts.
        MOV     ILR3,#01111111B   ;Set the interrupt level (level 1).
        MOV     DDR4,#00000000B   ;Set the UI2 pin as an input pin.
        MOV     SMC11,#01001111B  ;Attach no parity;  set one stop bit, operation
                                   mode 1, and asynchronous mode, clock output
                                   enable, and serial data output enable.
                                   (Set them after the 11 shift clock period
                                   needed for UART initialization after a reset
                                   has elapsed.)
        MOV     SRC1,#00011101B   ;Select the dedicated baud rate generator
                                   and set the baud rate to 300 baud.
        MOV     SSD1,#00001000B   ;Disable transmission interrupt requests
                                   and enable reception interrupt requests.
        MOV     A,SSD1            ;Perform this before transmission.
                                   (Setting TDRE = 1 enables transmission.)
        MOV     A,SIDR1           ;Clear an error flag.
        MOV     SODR1,#13H        ;Write transmission data (13$_H$).
        SETI                      ;Enable interrupts.
        :
---------Interrupt processing routine--------------------------------------------
WARI    PUSHW   A                 ;Save A and T.
        XCHW    A,T
        PUSHW   A
        MOV     A,SSD1            ;Read transfer data and clear the input
                                   data flag.
        MOV     A,SIDR1
        :
        User processing
        :
        POPW    A                 ;Restore A and T.
        XCHW    A,T
        POPW    A
        RETI
        ENDS
----------------------------------
        END
```

CHAPTER 18   8-BIT SERIAL I/O

This chapter describes the functions and operation of the 8-bit serial I/O.

18.1 Overview of the 8-Bit Serial I/O

The 8-bit serial I/O transfers 8-bit serial data synchronized by the shift clock. The shift clock can be selected from three internal clocks and an external clock. Either LSB-first or MSB-first can be selected as the data shift direction.

■ Serial I/O Function

The 8-bit serial I/O transfers 8-bit serial data synchronized by the shift clock.

- Converts 8-bit parallel data to serial data and outputs. Converts serial data into parallel data and stores it.
- The shift clock can be selected from three internal clocks and an external clock.
- Shift clock input and output can be controlled and the internal shift clock can be output.
- Either LSB-first or MSB-first can be selected as the data shift direction (transfer direction).

Table 18.1-1 Shift Clock Period and Transfer Speed

Shift clock	Clock period	Frequency (Hz)	Transfer speed (F_{CH} = 10 MHz, maximum clock speed (*1))
Internal shift clock (output)	$2t_{inst}$	$1/(2t_{inst})$	1250 kbps
	$8t_{inst}$	$1/(8t_{inst})$	312.5 kbps
	$32t_{inst}$	$1/(32t_{inst})$	28.125 kbps
External shift clock (input)	$2t_{inst}$ or more	$1/(2t_{inst})$ or less	DC to 1250 kbps

F_{CH}: Main clock oscillation
t_{inst}: Instruction cycle (depends on clock mode, etc.)
*1: For the case of main clock mode (SCS = 1) with the maximum clock speed (CS1, CS0 = 11_B, 1 instruction cycle = $4/F_{CH}$) selected in the system clock control register (SYCC).

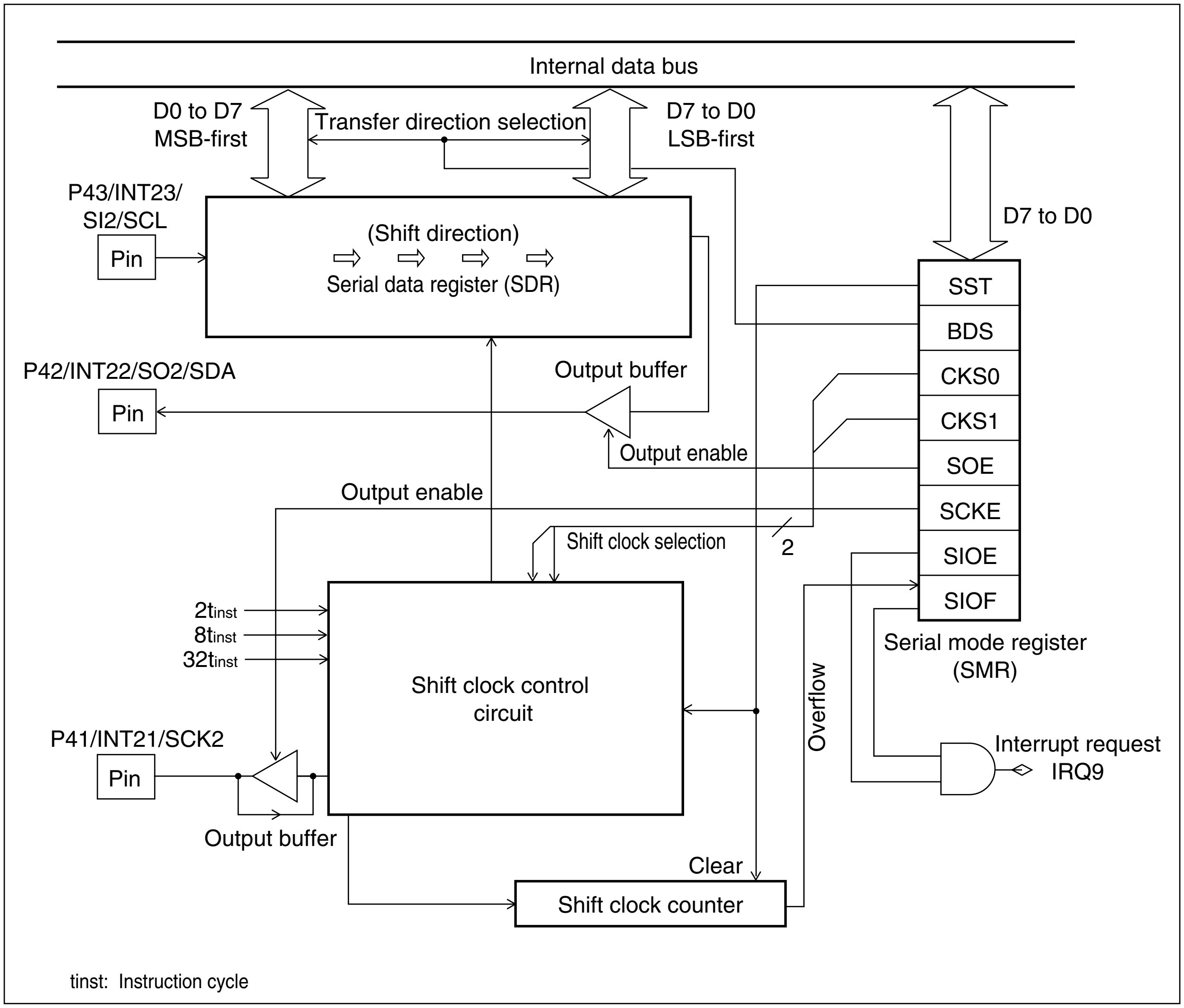
18.2 Configuration of the 8-Bit Serial I/O

Each channel of the 8-bit serial I/O consists of the following four blocks:

- **Shift clock control circuit**
- **Shift clock counter**
- **Serial data register (SDR)**
- **Serial mode register (SMR)**

■ Block Diagram of 8-Bit Serial I/O

Figure 18.2-1 Block Diagram of 8-Bit Serial I/O



❍ **Shift clock control circuit**

The shift clock can be selected from three internal clocks and an external clock.

If an internal clock is selected, the shift clock can be output to the SCK2 pin. Selecting the external clock uses the clock input from the SCK2 pin as the shift clock. The SDR register shift operation is driven by this shift clock and the shifted-out values are output from the SO2 pin. Similarly, the SI2 pin input is shifted into the SDR register.

❍ **Shift clock counter**

This counter counts the number of SDR register shifts driven by the shift clock and overflows after the 8-bit shift is complete.

When the counter overflows, the serial I/O transfer start bit of the SMR register is cleared (SMR: SST = 0) and the interrupt request flag is set (SMR: SIOF = 1). Halting serial transfer (SMR: SST = 0) halts the count on the shift clock counter and the counter is cleared by the next start (SMR: SST = 1).

❍ **Serial data register (SDR)**

This register stores the transfer data. The data written to this register is converted to serial data and is output. At the same time, the serial data is converted to parallel data and stored.

❍ **Serial mode register (SMR)**

This is the control register for the serial I/O. The register functions include enabling and disabling serial I/O operation, selecting the shift clock, setting the transfer (shift) direction, controlling interrupts, and checking the status.

❍ **Cause of an interrupt related to 8-bit serial I/O**

IRQ9:

When the I/O function completes 8-bit serial data I/O, 8-bit serial I/O generates an interrupt request (IRQ9) if output of interrupt requests has been enabled (SMR:SIOE = 1).

18.3 Pins of the 8-Bit Serial I/O

This section describes the 8-bit serial I/O pins, provides block diagrams of the pins, and explains the causes of 8-bit serial I/O interrupts.

■ 8-bit Serial I/O Pins

The 8-bit serial I/O pins are P43/INT23/SI2/SCL, P42/INT22/SO2/SDA, P41/INT21/SCK2.

❍ P43/INT23/SI2/SCL pin

The P43/INT23/SI2/SCL pin can function as an N-ch open-drain I/O port (P43), external interrupt pin (INT23), serial data input (hysteresis input) for 8-bit serial I/O (SI2), or I^{2}C serial clock I/O pin (SCL).

❍ P42/INT22/SO2/SDA pin

The P42/INT22/SO2/SDA pin can function as a general-purpose I/O port (P42), external interrupt pin (INT22), serial data output pin (N-ch open-drain) for 8-bit serial I/O (SO), or I^{2}C data I/O pin (SDA).

Enabling serial data output (SMR:SOE = 1) automatically sets the pin for output (SO2).

❍ P41/INT21/SCK2 pin

The P41/INT21/SCK2 pin can function either as a general-purpose I/O port (P41), external interrupt pin (INT21), or shift clock I/O pin (hysteresis input) for serial I/O (SCK2).

- When used as the shift clock input pin:

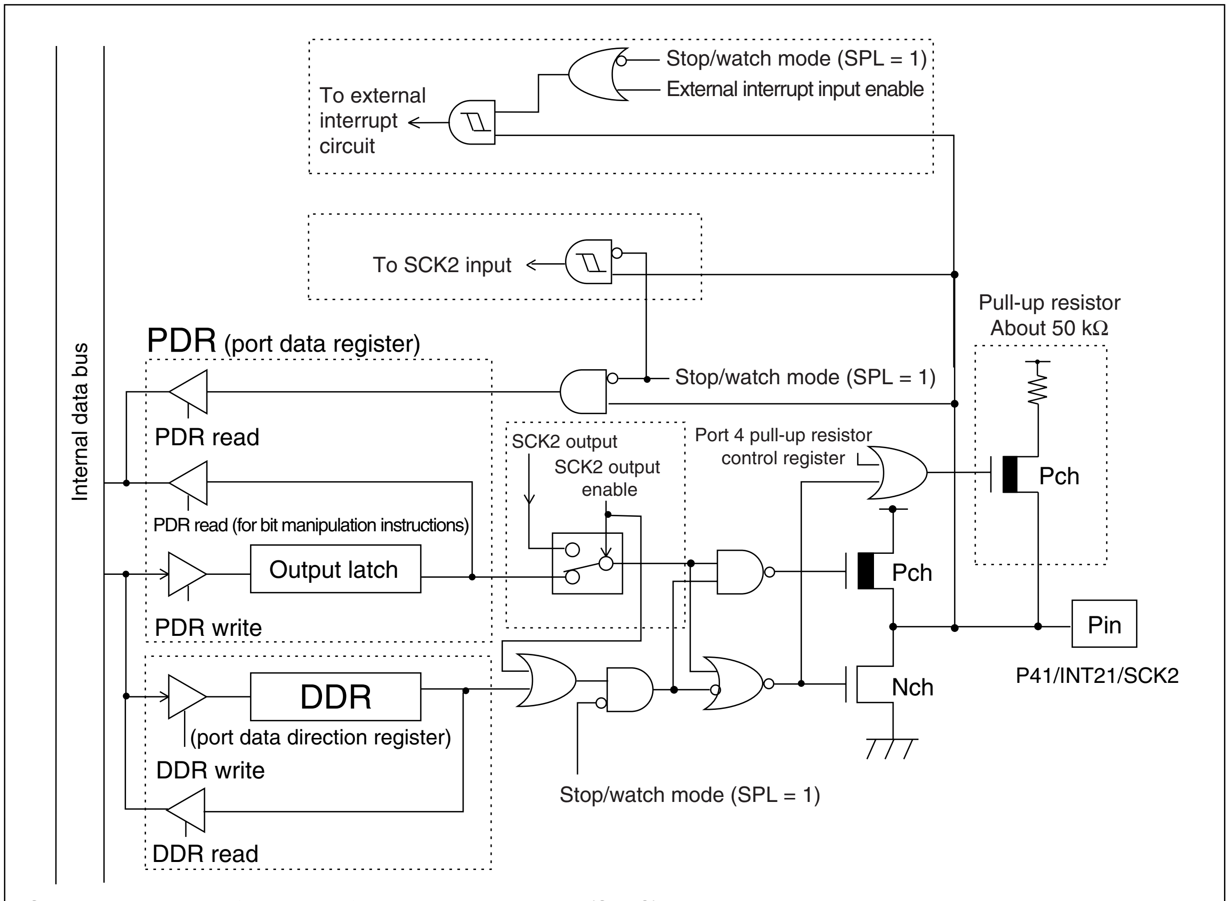
  To use the SCK pin as an input, set as an input port in the port data direction register (DDR4: bit 1 = 0) and disable shift clock output (SMR:SCKE = 0). In this case, select the external shift clock (SMR:CKS1, CKS0 = 11_B).

- When used as the shift clock output pin:

  Enabling shift clock output (SMR:SCKE = 1) automatically sets the P41/INT21/SCK2 pin as an output pin regardless of the value of the port data direction register (DDR4: bit 1) and sets the pin to function as the SCK2 output pin. In this case, select an internal shift clock (SMR:CKS1, CKS0 = other than 11_B).

■ Block Diagram of 8-Bit Serial I/O Pins

Figure 18.3-1 Block Diagram of SCK2 Pin



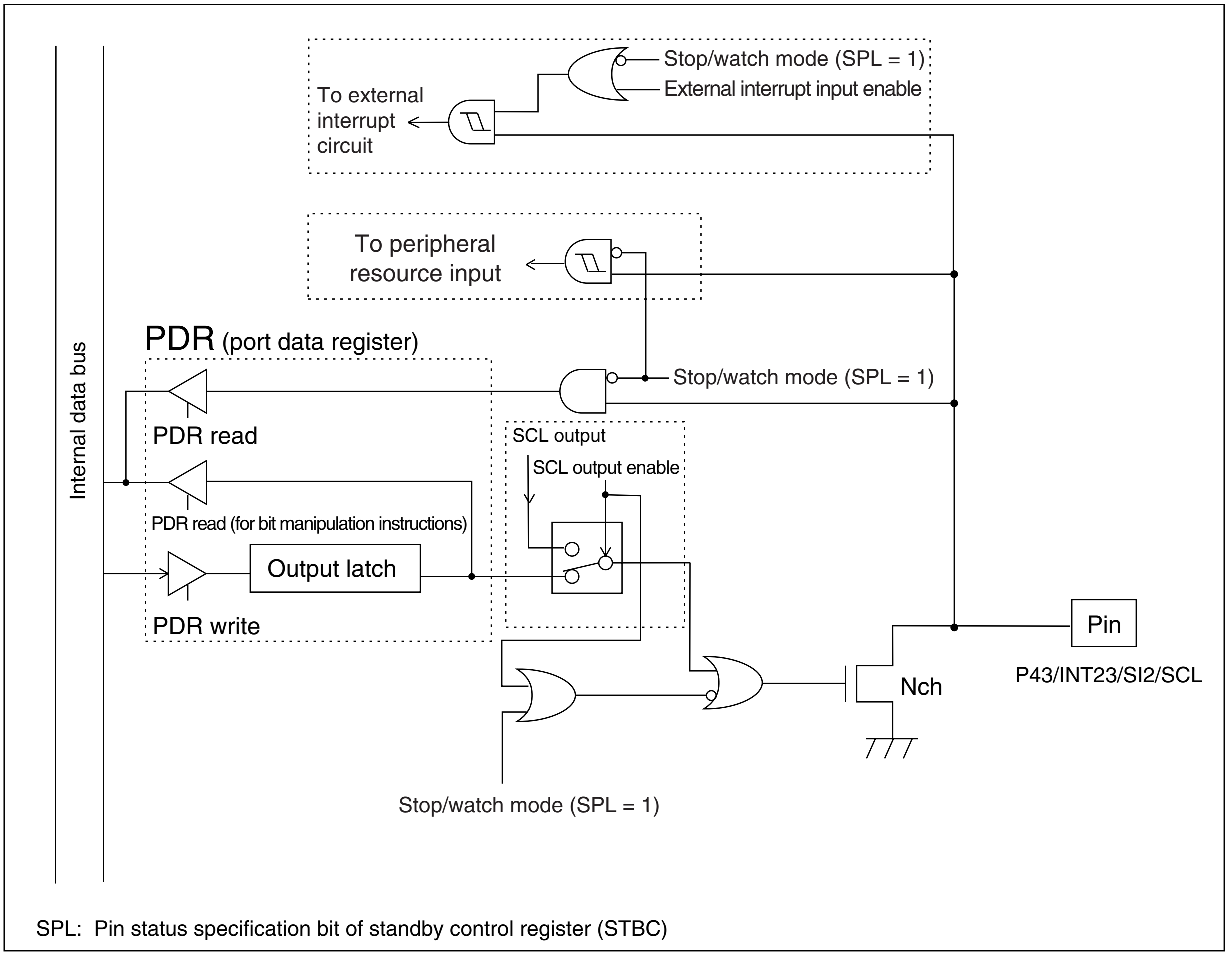
SPL: Pin status specification bit of standby control register (STBC)

Reference:

If "pull-up resistor available" is selected using the port 4 pull-up resistor control register, the pins are set to the "H" (pulled up) level, not the high impedance state, in stop mode or watch mode (STBC:SPL = 1) .

During a reset, however, the pull-up is disabled and the pins are set to the Hi-z level.

Figure 18.3-2 Block Diagram of SI2 Pin

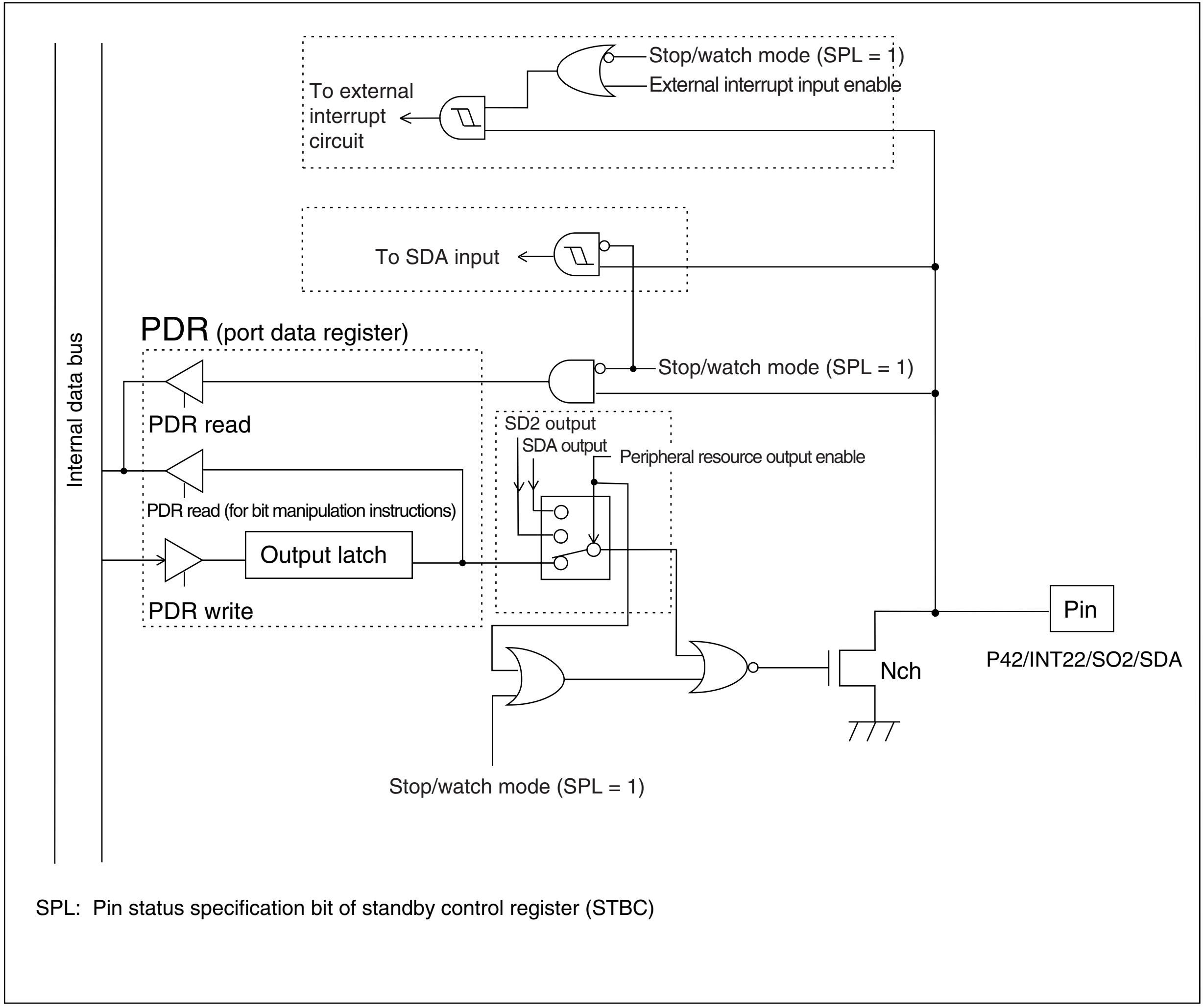

Stop/watch mode (SPL = 1)
External interrupt input enable
To external interrupt circuit
To peripheral resource input
Internal data bus
PDR (port data register)
Stop/watch mode (SPL = 1)
PDR read
SCL output
SCL output enable
PDR read (for bit manipulation instructions)
Output latch
PDR write
Pin
P43/INT23/SI2/SCL
Nch
Stop/watch mode (SPL = 1)
SPL: Pin status specification bit of standby control register (STBC)

Reference:

For the P43/INT23/SI2/SCL pin, the port 4 pull-up control register cannot be used to select whether pull-up resister is available or unavailable.

To use the P43/INT23/SI2/SCL pin as an output pin, attach a pull-up resistor to the external pin.

Figure 18.3-3 Block Diagram of SO2 Pin


Stop/watch mode (SPL = 1)
External interrupt input enable
To external interrupt circuit
To SDA input
Internal data bus
PDR (port data register)
Stop/watch mode (SPL = 1)
PDR read
SD2 output
SDA output
Peripheral resource output enable
PDR read (for bit manipulation instructions)
Output latch
PDR write
Pin
Nch
P42/INT22/SO2/SDA
Stop/watch mode (SPL = 1)
SPL: Pin status specification bit of standby control register (STBC)

Reference:

For the P42/INT22/SO2/SDA pin, the port 4 pull-up control register cannot be used to select whether the pull-up resistor is available or unavailable.

To use the P42/INT22/SO2/SDA pin as an output pin, attach a pull-up resistor to the external pin.

18.4 Registers of the 8-Bit Serial I/O

This section describes the 8-bit serial I/O registers.

■ 8-Bit Serial I/O Registers

Figure 18.4-1 8-Bit Serial I/O Registers

SMR (Serial mode register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 7 0$_H$	SIOF	SIOE	SCKE	SOE	CKS1	CKS0	BDS	SST	00000000$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

SDR (Serial data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 7 1$_H$									XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

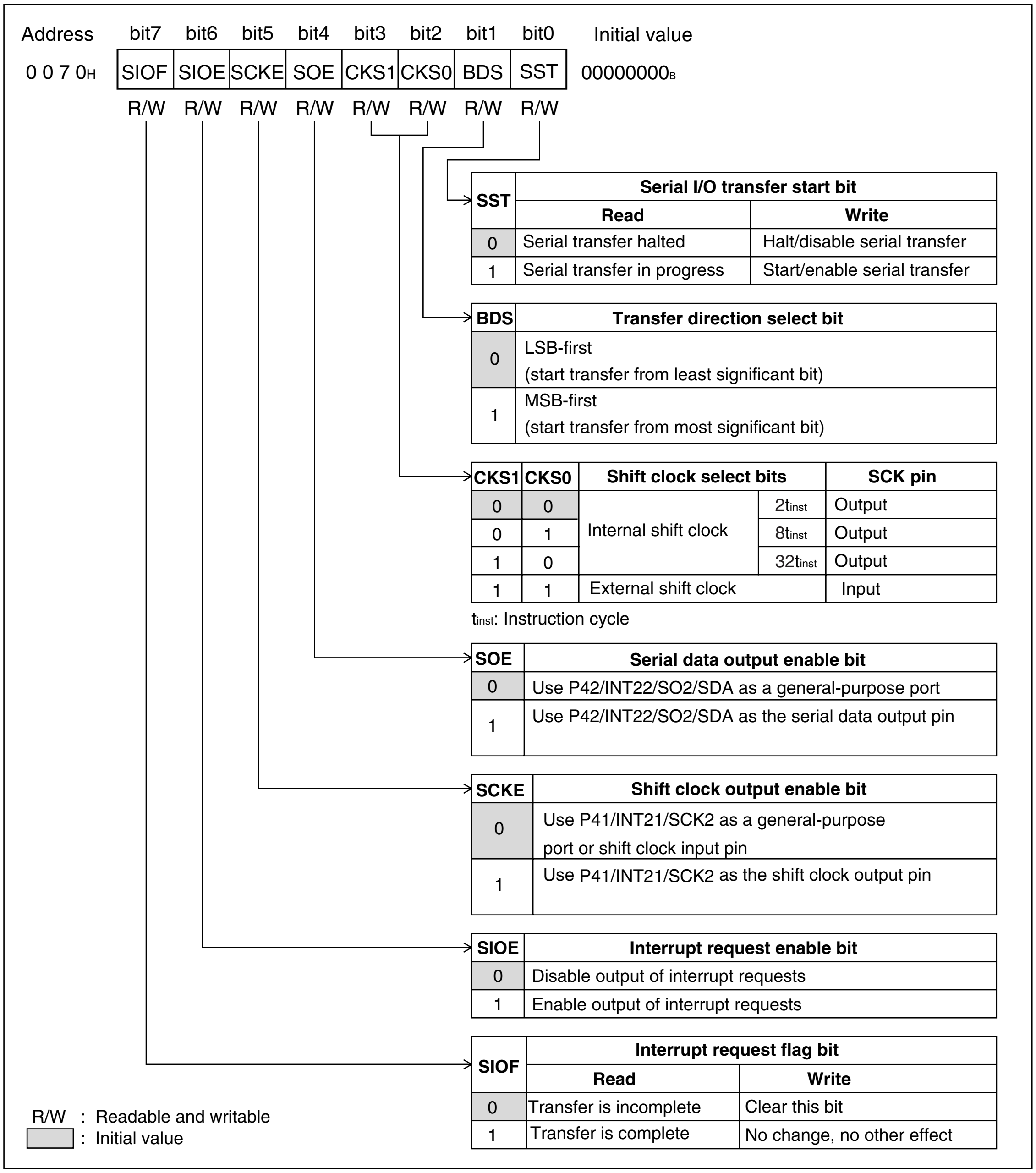
R/W : Readable and writable
X : Undefined

18.4.1 Serial Mode Register (SMR)

The serial mode register (SMR) is used to enable and disable 8-bit serial I/O operation, select the shift clock, set the transfer direction, control interrupts, and check the status.

■ Serial Mode Register (SMR)

Figure 18.4-2 Serial Mode Register (SMR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 7 0H	SIOF	SIOE	SCKE	SOE	CKS1	CKS0	BDS	SST	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

SST	Serial I/O transfer start bit	
	Read	Write
0	Serial transfer halted	Halt/disable serial transfer
1	Serial transfer in progress	Start/enable serial transfer

BDS	Transfer direction select bit
0	LSB-first (start transfer from least significant bit)
1	MSB-first (start transfer from most significant bit)

CKS1	CKS0	Shift clock select bits		SCK pin
0	0	Internal shift clock	$2t_{inst}$	Output
0	1		$8t_{inst}$	Output
1	0		$32t_{inst}$	Output
1	1	External shift clock		Input

t_{inst}: Instruction cycle

SOE	Serial data output enable bit
0	Use P42/INT22/SO2/SDA as a general-purpose port
1	Use P42/INT22/SO2/SDA as the serial data output pin

SCKE	Shift clock output enable bit
0	Use P41/INT21/SCK2 as a general-purpose port or shift clock input pin
1	Use P41/INT21/SCK2 as the shift clock output pin

SIOE	Interrupt request enable bit
0	Disable output of interrupt requests
1	Enable output of interrupt requests

SIOF	Interrupt request flag bit	
	Read	Write
0	Transfer is incomplete	Clear this bit
1	Transfer is complete	No change, no other effect

R/W : Readable and writable
▨ : Initial value

Table 18.4-1 Function of Each Serial Mode Register (SMR) Bit

Bit		Description
bit7	SIOF: Interrupt request flag bit	• Set to "1" after the serial I/O operation has input and output 8 bits of serial data. An interrupt request is output if this bit and the interrupt request enable bit (SIOE) are "1". • Writing "0" clears the bit. Writing "1" has no effect and does not change the bit value.
bit6	SIOE: Interrupt request enable bit	This bit enables or disables output of interrupt requests to the CPU. An interrupt request is output if this bit and the interrupt request flag bit (SIOF) are "1".
bit5	SCKE: Shift clock output enable bit	• This bit controls shift clock input and output. • The P41/INT21/SCK2 pin becomes the shift clock input pin when this bit is "0" and the shift clock output pin when the bit is "1". **Note:** • When used as the shift clock input, the P41/INT21/SCK2 pin must be set as an input port. Also, select the external shift clock in the shift clock select bits (CKS1, CKS0 = 11_B). • Select an internal shift clock (CKS1, CKS0 = other than 11_B) when the pin is the shift clock output (SCKE = 1). **Reference:** • Enabling the shift clock output (SCKE = 1) causes the P41/INT21/SCK2 pin to function as the SCK output pin regardless of the state of the general-purpose port (P41). • When using the P41/INT21/SCK2 pin as a general-purpose port (P41), set as a shift clock input (SCKE = 0).
bit4	SOE: Serial data output enable bit	The P42/INT22/SO2/SDA pin becomes a general-purpose port (P42) when this bit is "0" and the serial data output pin (SO2) when the bit is "1". **Reference:** Enabling serial data output (SOE = 1) causes the P42/INT22/SO2/SDA pin to function as the SO2 pin regardless of the state of the general-purpose port (P42).
bit3 bit2	CKS1, CKS0: Shift clock select bits	• These bits select the external shift clock or one of the three internal shift clocks. • When these bits are other than "11_B", an external shift clock is selected and, if the shift clock output enable bit (SCKE) is "1", the shift clock is output from the SCK2 output pin. • When these bits are "11_B", the external shift clock is selected and, if set as the shift clock input, the shift clock is input from the SCK2 output pin (SCKE=0, DDR2:bit5=0).
bit1	BDS: Transfer direction select bit	This bit selects whether to transfer the serial data starting from the least significant bit (LSB-first, BDS = 0) or the most significant bit (MSB-first, BDS = 1). **Note:** As the bit order is set when the data is read or written to the serial data register (SDR), changing the value of this bit after data has been written to the SDR register invalidates the data.
bit0	SST: Serial I/O transfer start bit	• This bit controls starting and enabling of serial I/O transfer. The bit can also be used to test whether transfer is complete. • When using an internal shift clock (CKS1, CKS0 = other than 11_B), writing "1" to this bit clears the shift clock counter and starts transfer. • When using an external shift clock (CKS1, CKS0 = 11_B), writing "1" to this bit enables transfer, clears the shift clock counter, and waits for input of the external shift clock. • When transfer is completed, the bit is cleared to "0" and the SIOF bit is set to "1". • Writing "0" to this bit during transfer (SST = 1) halts the transfer. Once a transfer has been halted, the output SDR register must be written to again and the transfer restarted for data input (to clear the shift clock counter).

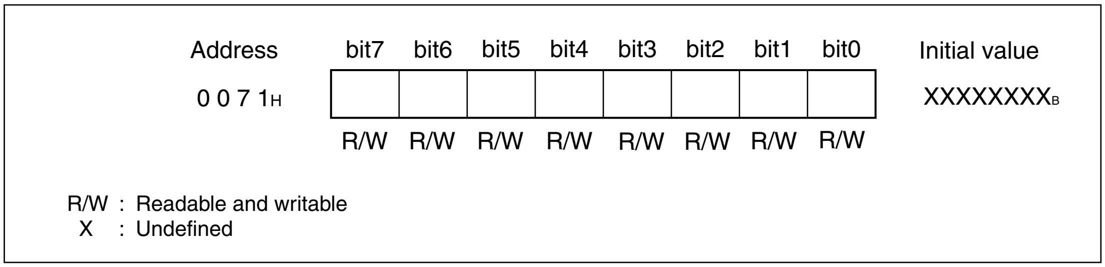
18.4.2 Serial Data Register (SDR)

The serial data register (SDR) stores the 8-bit serial I/O transfer data.
For serial output operation, the register functions as the transmission data register.
For serial input operation, the register functions as the reception data register.

■ Serial Data Register (SDR)

Figure 18.4-3 "Serial Data Register (SDR)" shows the bit structure of the serial data register.

Figure 18.4-3 Serial Data Register (SDR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 7 1H									XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable and writable
X : Undefined

❍ Serial output operation

The register functions as the transmission data register. Starting serial I/O transfer (SMR:SST = 1) performs serial transfer of the data written to this register.

As the transmission data is shifted out by the transfer operation, the data does not remain in the SDR register.

❍ Serial input operation

The register functions as the reception data register. Starting serial I/O transfer (SMR:SST = 1) stores the received serial transfer data in this register.

❍ During serial I/O transfer

Do not write data to the SDR register while a serial I/O transfer operation is in progress. Similarly, values read from the register at this time have no meaning.

If serial output and serial input are enabled at the same time, both serial input and output operations are performed.

18.5 8-Bit Serial I/O Interrupts

Completion of an 8-bit serial I/O operation generates an interrupt request from the 8-bit data serial I/O.

■ Interrupts during Serial I/O Operation

The 8-bit serial I/O performs serial input and serial output simultaneously. When serial transfer starts, the contents of the serial data register (SDR) are input and output one bit at a time synchronized with the period of the specified shift clock. The interrupt request flag bit (SMR:SIOF) is set to "1" on the leading edge of the eighth shift clock pulse.

An interrupt request (IRQ9) is output to the CPU if the interrupt request output enable bit is enabled (SMR:SIOE = 1) at this time.

Write "0" to the SIOF bit in the interrupt processing routine to clear the interrupt request. The SIOF bit is always set when the output of 8 bits of serial is completed, regardless of the SIOE bit value.

Reference:

During serial I/O operation, setting the interrupt request flag bit (SMR:SIOF = 1) is not performed if serial transfer is halted (SMR:SST = 0) at the same time that serial data transfer is completed. An interrupt request is generated immediately if the SIOF bit is "1" when the SIOE bit is switched from disabled to enabled (0 to 1).

■ Register and Vector Table for the 8-Bit Serial I/O Interrupt

Table 18.5-1 Register and Vector Table for the 8-Bit Serial I/O Interrupt

Interrupt	Interrupt level setting register			Vector table address	
	Register	Setting bits		Upper	Lower
IRQ9	ILR3 (007D$_H$)	L91 (bit3)	L90 (bit2)	FFE8$_H$	FFE9$_H$

For more information on interrupt operation, see Section 3.4.2 "Interrupt Processing".

18.6 Operation of the Serial Output

The 8-bit serial I/O can output 8-bit serial data synchronized with a shift clock.

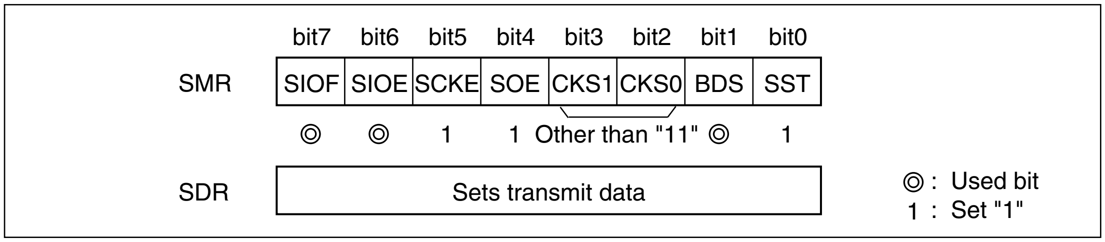
■ Serial Output Operation

Serial output can operate using either an internal or external shift clock. When serial I/O operation is enabled, the contents of the SDR register are output from the serial data output pin (SO2) at the same time that serial input is performed.

❍ When using an internal shift clock

Figure 18.6-1 "Serial Output Settings (for an Internal Shift Clock)" shows the settings required for operating serial output using an internal shift clock.

Figure 18.6-1 Serial Output Settings (for an Internal Shift Clock)



	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SMR	SIOF	SIOE	SCKE	SOE	CKS1	CKS0	BDS	SST
	◎	◎	1	1	Other than "11"		◎	1
SDR	Sets transmit data							

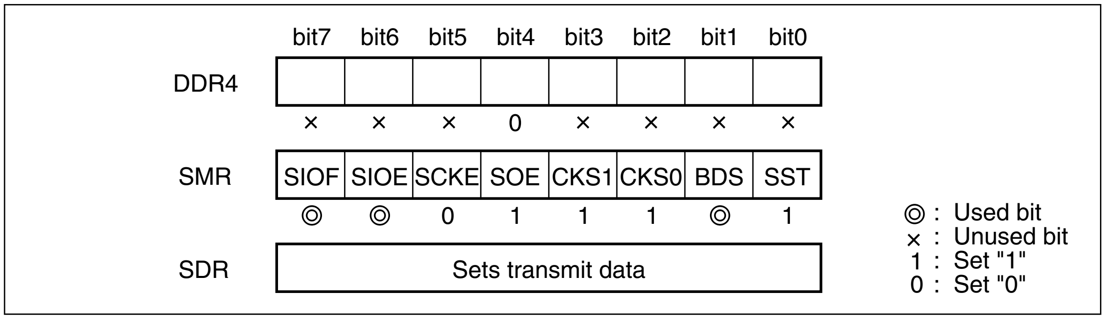
◎ : Used bit
1 : Set "1"

Activating the serial output operation outputs the contents of the SDR register from the serial data output pin (SO2). Output is synchronized with the trailing edge of the selected internal shift clock. At this time, the device being communicated with (serial input device) must be waiting for the input of an external shift clock.

❍ When using an external shift clock

Figure 18.6-2 "Serial Output Settings (for an External Shift Clock)" shows the settings required for operating serial output using an external shift clock.

Figure 18.6-2 Serial Output Settings (for an External Shift Clock)



	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR4								
	×	×	×	0	×	×	×	×
SMR	SIOF	SIOE	SCKE	SOE	CKS1	CKS0	BDS	SST
	◎	◎	0	1	1	1	◎	1
SDR	Sets transmit data							

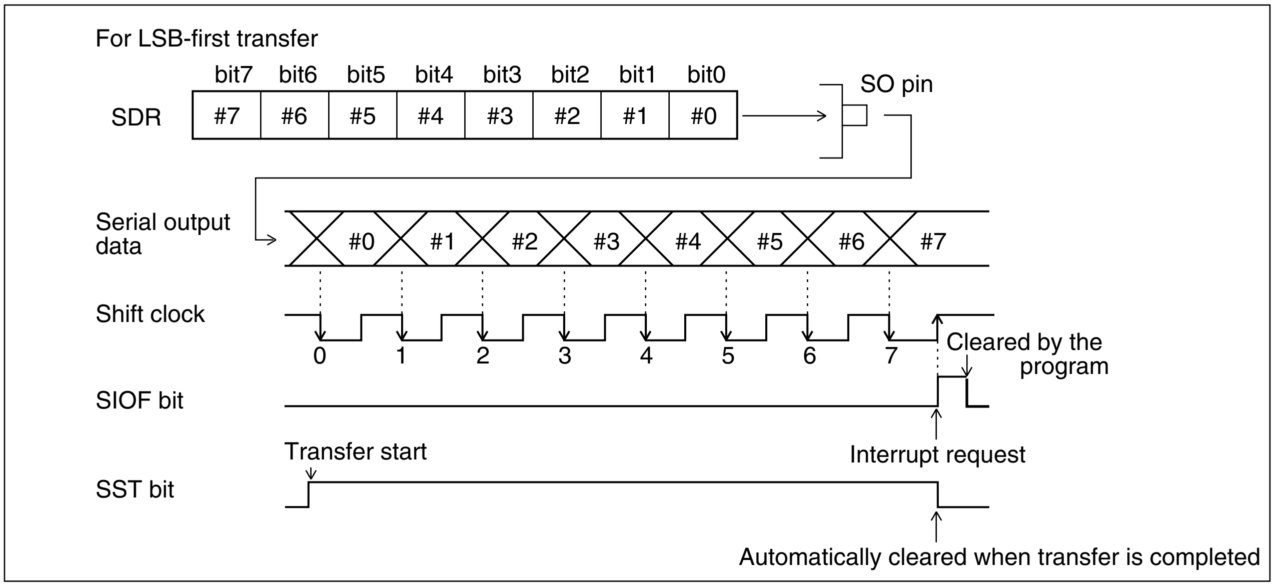
◎ : Used bit
× : Unused bit
1 : Set "1"
0 : Set "0"

Enabling the serial output operation outputs the contents of the SDR register from the serial data output pin (SO2). Output is synchronized with the trailing edge of the external shift clock. When serial output is completed, the SDR register must immediately be set again and operation enabled (SMR:SST = 1) so as to be ready to output the next data.

When the serial input operation is completed (leading edge) at the receiving device, set the external shift clock to the "H" level while waiting for output of the next data (idle state).

Figure 18.6-3 "Operation of 8-Bit Serial Output" shows the operation of 8-bit serial output.

Figure 18.6-3 Operation of 8-Bit Serial Output



■ Operation when serial output is completed

The interrupt request flag bit is set (SMR:SIOF = 1) and the serial I/O start bit cleared (SMR:SST = 0) on the leading edge of the shift clock after inputting and outputting the eighth bit of serial data.

18.7 Operation of the Serial Input

The 8-bit serial I/O can input 8-bit serial data synchronized with a shift clock.

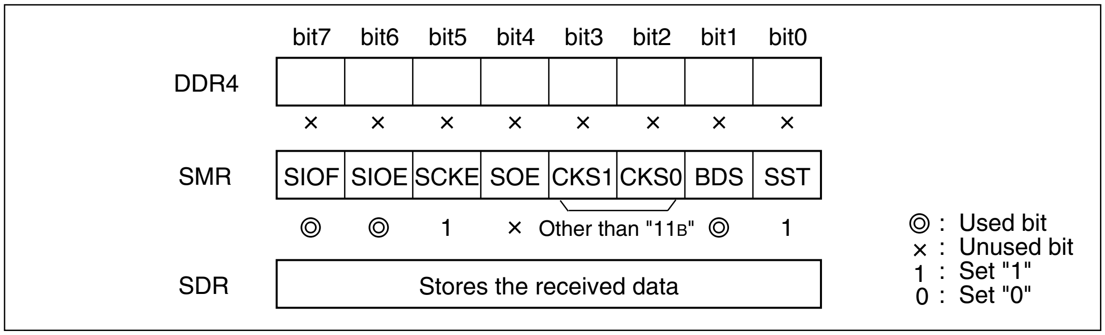
■ Serial Input Operation

Serial input can operate using either an internal or external shift clock. When serial I/O operation is enabled, the contents of the SDR register are output from the serial data output pin (SO2) at the same time that serial input is performed.

❍ When using an internal shift clock

Figure 18.7-1 "Serial Input Settings (for an Internal Shift Clock)" shows the settings required for operating serial input using an internal shift clock.

Figure 18.7-1 Serial Input Settings (for an Internal Shift Clock)



	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR4								
	×	×	×	×	×	×	×	×
SMR	SIOF	SIOE	SCKE	SOE	CKS1	CKS0	BDS	SST
	◎	◎	1	×	Other than "11_B"		◎	1
SDR	Stores the received data							

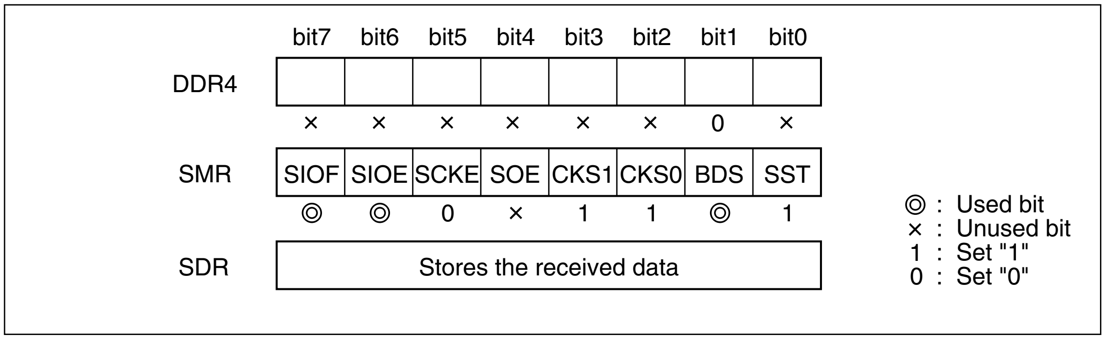
◎ : Used bit
× : Unused bit
1 : Set "1"
0 : Set "0"

Activating the serial input operation inputs the value of the serial data input pin (SI2) to the SDR register. Input is synchronized with the leading edge of the selected internal shift clock. At this time, the device being communicated with (serial output device) must have set a value in the SDR register and be waiting for input of an external shift clock.

❍ When using an external shift clock

Figure 18.7-2 "Serial Input Settings (for an External Shift Clock)" shows the settings required for operating serial input using an external shift clock.

Figure 18.7-2 Serial Input Settings (for an External Shift Clock)



	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR4								
	×	×	×	×	×	×	0	×
SMR	SIOF	SIOE	SCKE	SOE	CKS1	CKS0	BDS	SST
	◎	◎	0	×	1	1	◎	1
SDR	Stores the received data							

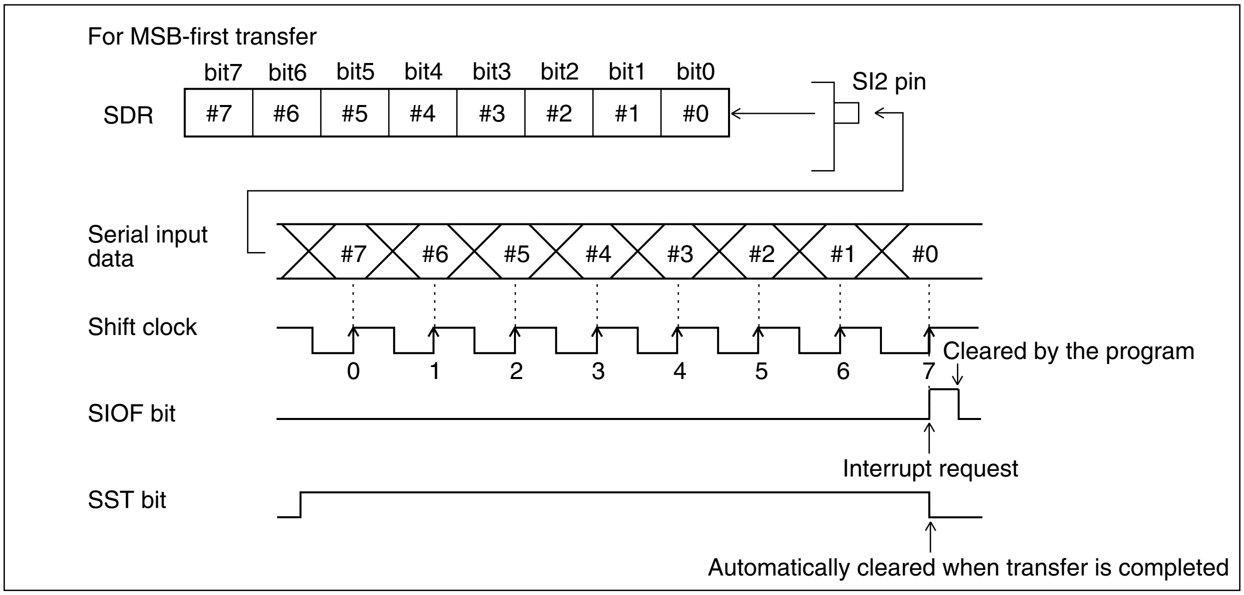
◎ : Used bit
× : Unused bit
1 : Set "1"
0 : Set "0"

Enabling the serial input operation inputs the value of the serial data input pin (SI1) to the SDR register. Input is synchronized with the leading edge of the external shift clock. When serial input is completed, the SDR register must immediately be read and operation enabled (SMR:SST = 1) so as to be ready to input the next data.

While waiting for output of the next data (idle state), set the external shift clock to the "H" level.

Figure 18.7-3 "Operation of 8-Bit Serial Input" shows the operation of 8-bit serial input.

Figure 18.7-3 Operation of 8-Bit Serial Input



■ Operation When Serial Input is Completed

The interrupt request flag bit is set (SMR:SIOF = 1) and the serial I/O start bit is cleared (SMR:SST = 0) on the leading edge of the shift clock after inputting and outputting the eighth bit of serial data.

18.8 States in Each Mode of 8-Bit Serial I/O Operation

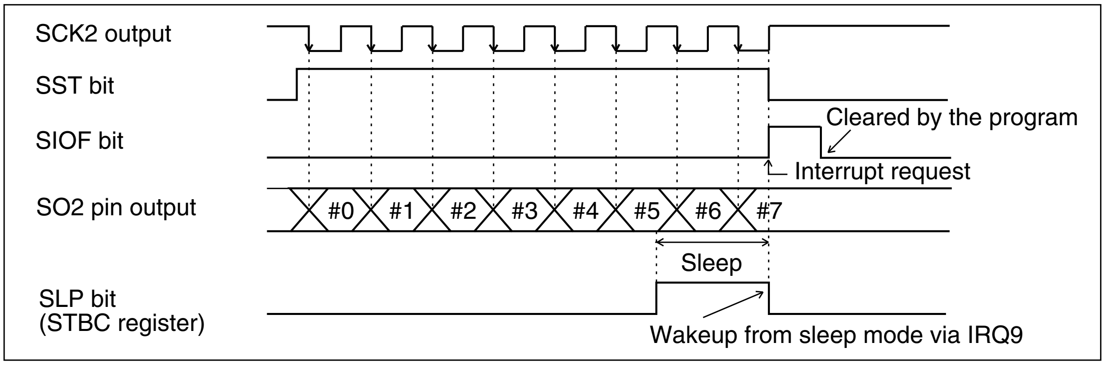
This section describes the operation when the device enters sleep mode, the device enters stop or watch mode, or a halt request occurs during operation of the 8-bit serial I/O.

■ When Using an Internal Shift Clock

❍ Operation in sleep mode

Figure 18.8-1 "Operation in Sleep Mode (Internal Shift Clock)" shows how serial I/O operation does not halt and transfer continues when the device enters sleep mode.

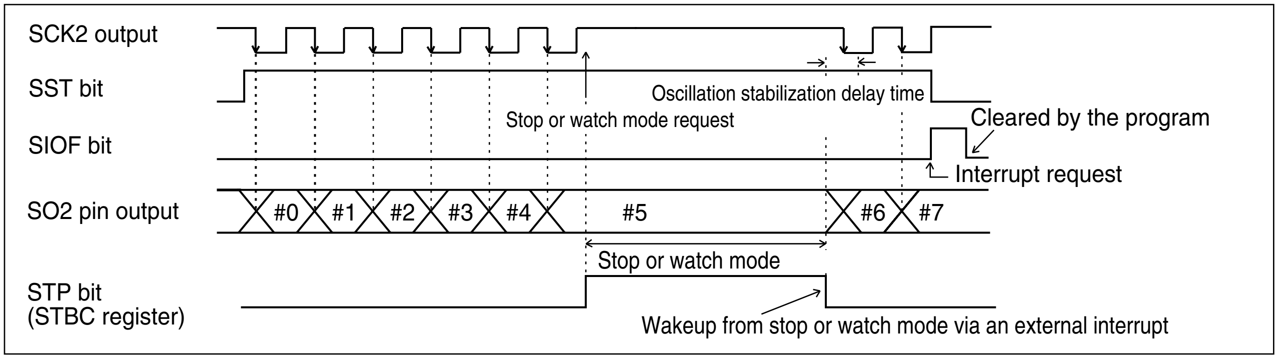
Figure 18.8-1 Operation in Sleep Mode (Internal Shift Clock)



❍ Operation in stop or watch mode

Figure 18.8-2 "Operation in Stop or Watch Mode (Internal Shift Clock)" shows how serial I/O operation halts and transfer is interrupted when the device enters stop or watch mode. When the device wakes up from stop or watch mode, operation restarts from the point where it halted. Therefore, initialize the serial I/O in accordance with the state of the device with which you are communicating.

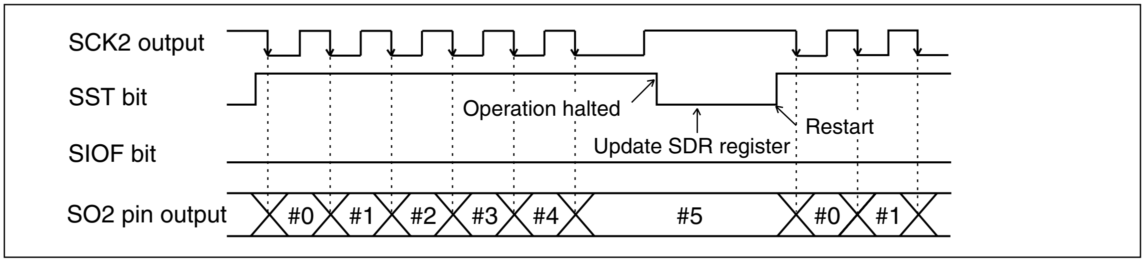
Figure 18.8-2 Operation in Stop or Watch Mode (Internal Shift Clock)



❍ Operation during a halt

Figure 18.8-3 "Operation during a Halt (Internal Shift Clock)" shows how transfer halts and the shift clock counter is cleared when operation is halted midway through a transfer (SMR:SST = 0). Accordingly, the device with which you are communicating must also be initialized. When performing serial output, update the SDR register before restarting operation.

Figure 18.8-3 Operation during a Halt (Internal Shift Clock)

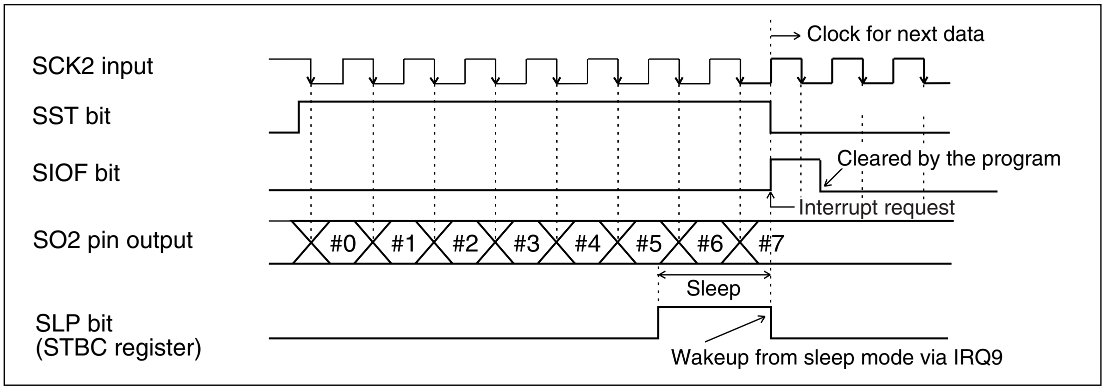


■ When Using an External Shift Clock

❍ Operation in sleep mode

Figure 18.8-4 "Operation in Sleep Mode (External Shift Clock)" shows how serial I/O operation does not halt and transfer continues when the device enters sleep mode.

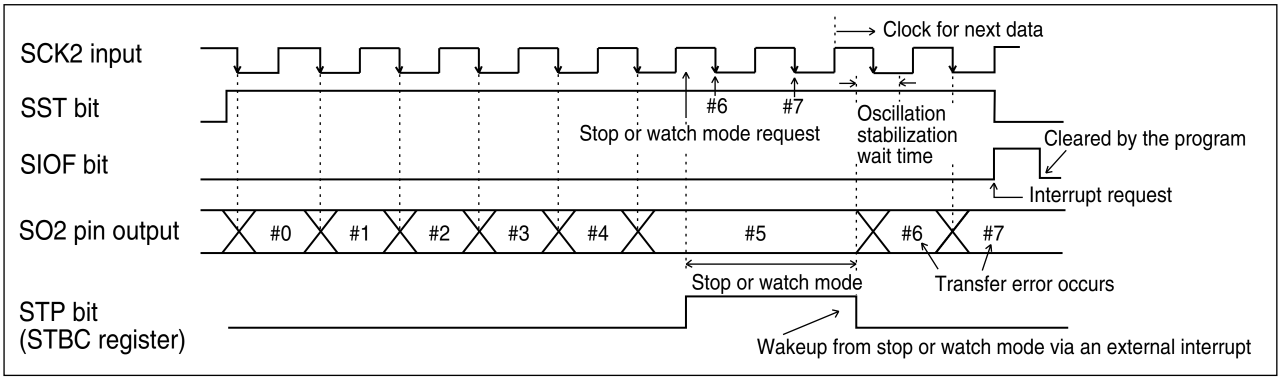
Figure 18.8-4 Operation in Sleep Mode (External Shift Clock)



❍ Operation in stop or watch mode

Figure 18.8-5 "Operation in Stop or Watch Mode (External Shift Clock)" shows how serial I/O operation halts and transfer is interrupted when the device enters stop or watch mode. When the device wakes up from stop or watch mode, operation restarts from the point where it halted. Therefore, a transfer error occurs. You must reinitialize the serial I/O.

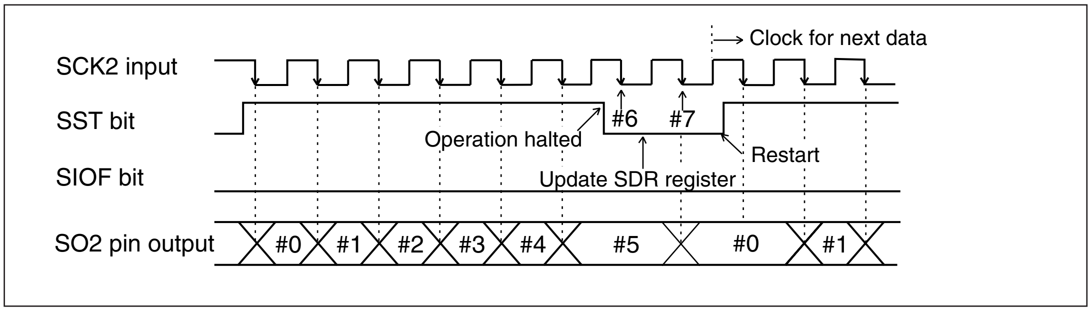
Figure 18.8-5 Operation in Stop or Watch Mode (External Shift Clock)



❍ Operation during a halt

Figure 18.8-6 "Operation during a Halt (External Shift Clock)" shows how transfer halts and the shift clock counter is cleared when operation is halted midway through a transfer (SMR:SST = 0). Accordingly, the device with which you are communicating must also be initialized. When performing serial output, update the SDR register before restarting operation. At this time, the SO2 pin output changes when an external clock is input.

Figure 18.8-6 Operation during a Halt (External Shift Clock)



18.9 Notes on Using the 8-Bit Serial I/O

This section describes points to note when using the 8-bit serial I/O.

■ Notes on Using 8-Bit Serial I/O

❍ Error in serial transfer start timing

As the timing at which serial transfer is activated by the program (SMR:SST = 1) is asynchronous to the trailing (output) or leading (input) edge of the shift clock, the timing of the first serial data input or output may be delayed by up to the period of the specified shift clock.

❍ Malfunction due to noise

Malfunction may occur on the serial I/O if an unwanted pulse (a pulse which exceeds the hysteresis width) is present on the shift clock due to external noise during serial data transfer.

❍ Notes on setting the serial I/O by using the program

- Write to the serial mode register (SMR) and serial data register (SDR) when the serial I/O is halted (SMR:SST = 0).
- Do not change the values of other SMR register bits when starting (enabling) serial I/O transfer (SMR:SST = 1).
- When inputting an external shift clock and when serial data output is enabled (SMR:SOE = 1), the output level of the SO1 pin enters the level of the most significant bit (for MSB-first transfer) or least significant bit (for LSB-first transfer) when the external shift clock is input, even if serial I/O transfer is halted (SMR:SST = 0).
- If serial I/O transfer is halted (SMR:SST = 0) at the same time that a serial data transfer is completed, the interrupt request flag bit (SMR:SIOF) is not set.
- Interrupt processing cannot return if the SIOF bit is "1" and interrupt request output is enabled (SIOE = 1). Always clear the SIOF bit.

❍ Serial I/O transfer speed

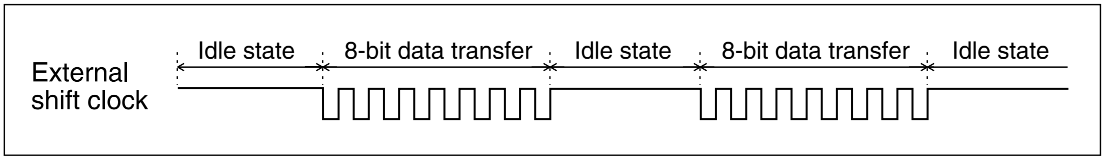
The serial data output pin (SO2) for serial I/O cannot be used for high-speed transfer because of N-ch open-drain output. Note this point when high-speed shift clocks are used.

❍ Shift clock idle state

During the delay between 8-bit data transfers (idle state), set the external shift clock to the "H" level. When using an internal shift clock (SMR2:CKS1, CKS0 = other than 11_B) to provide the shift clock output (SMR:SCKE = 1), the output enters the "H" level when idling.

Figure 18.9-1 "Shift Clock Idle State" shows the idle state of the shift clock.

Figure 18.9-1 Shift Clock Idle State



18.10 8-Bit Serial I/O Connection Example

This section shows an example of connecting together the 8-bit serial I/O of two MB89530/530H/530A series devices to perform bi-directional serial I/O.

■ Bi-Directional Serial I/O

Figure 18.10-1 8-Bit Serial I/O Connection Example (Interfacing Two MB89530/530H/530A Devices)

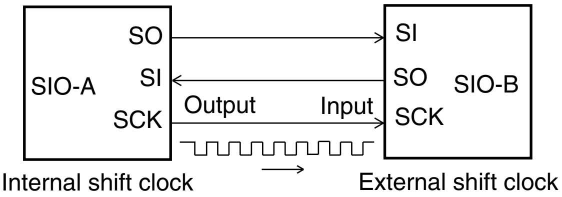
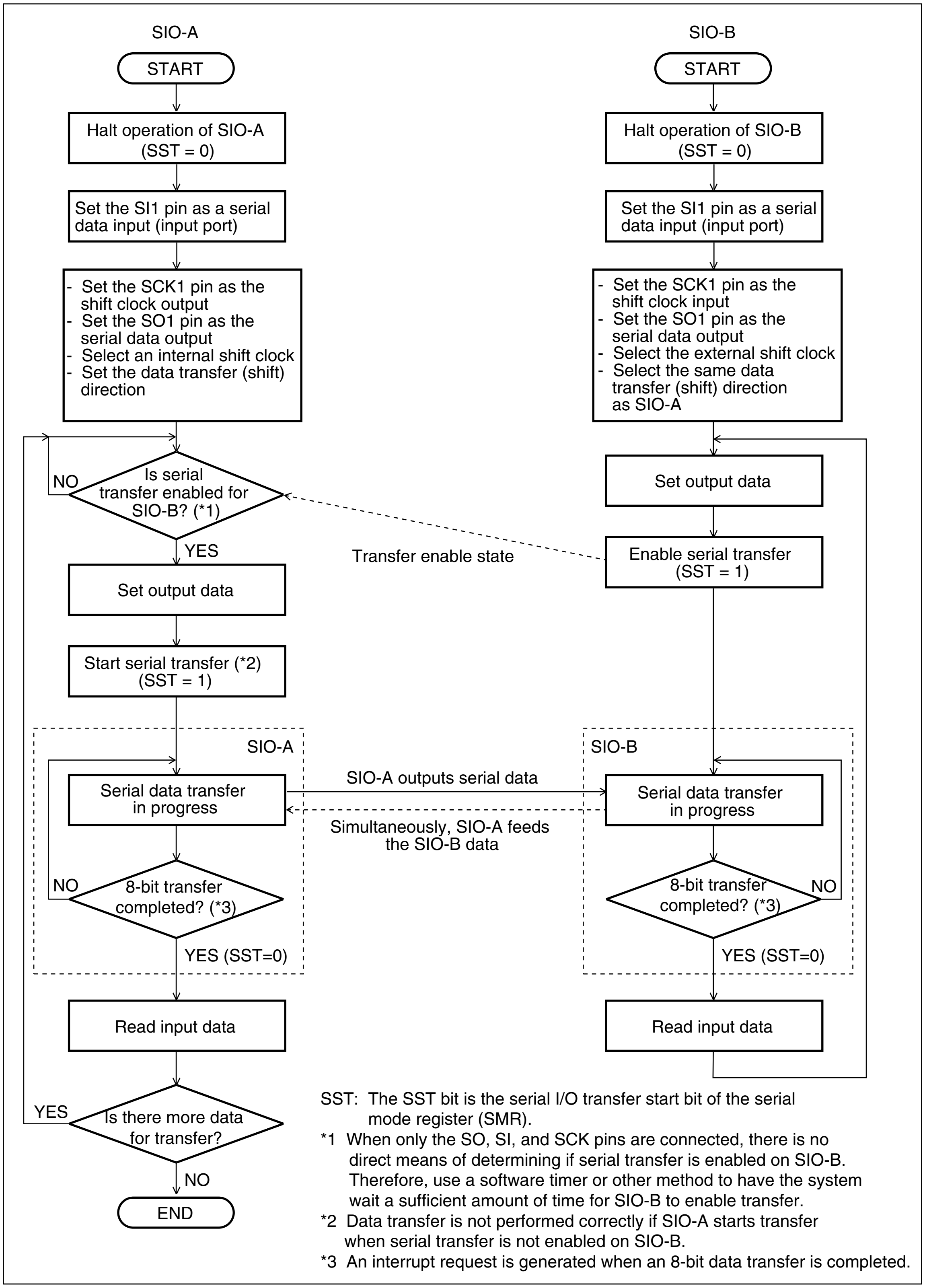


Figure 18.10-2 Operation of Bidirectional Serial I/O


SIO-A
START
Halt operation of SIO-A (SST = 0)
Set the SI1 pin as a serial data input (input port)
- Set the SCK1 pin as the shift clock output
- Set the SO1 pin as the serial data output
- Select an internal shift clock
- Set the data transfer (shift) direction
NO
Is serial transfer enabled for SIO-B? (*1)
YES
Set output data
Start serial transfer (*2) (SST = 1)
SIO-A
Serial data transfer in progress
NO
8-bit transfer completed? (*3)
YES (SST=0)
Read input data
YES
Is there more data for transfer?
NO
END
SIO-B
START
Halt operation of SIO-B (SST = 0)
Set the SI1 pin as a serial data input (input port)
- Set the SCK1 pin as the shift clock input
- Set the SO1 pin as the serial data output
- Select the external shift clock
- Select the same data transfer (shift) direction as SIO-A
Set output data
Transfer enable state
Enable serial transfer (SST = 1)
SIO-B
SIO-A outputs serial data
Serial data transfer in progress
Simultaneously, SIO-A feeds the SIO-B data
8-bit transfer completed? (*3)
NO
YES (SST=0)
Read input data

SST: The SST bit is the serial I/O transfer start bit of the serial mode register (SMR).

*1 When only the SO, SI, and SCK pins are connected, there is no direct means of determining if serial transfer is enabled on SIO-B. Therefore, use a software timer or other method to have the system wait a sufficient amount of time for SIO-B to enable transfer.

*2 Data transfer is not performed correctly if SIO-A starts transfer when serial transfer is not enabled on SIO-B.

*3 An interrupt request is generated when an 8-bit data transfer is completed.

18.11 Program Examples of the 8-Bit Serial I/O

This section describes example programs using the 8-bit serial I/O.

■ Program Example for Serial Output

❍ **Program specifications**

- Output 8 bits of serial data (55_H) from the SO2 pin of the serial I/O and generate an interrupt when transfer is completed.
- In the interrupt processing routine, set the next data to be transferred and restart output.
- Operate using an internal shift clock and output the shift clock from the SCK pin.
- The transfer speed and time between interrupts is as follows for a main clock oscillation (F_{CH}) of 10 MHz, the main clock speed (gear) set to maximum speed (1 instruction cycle = 4/ F_{CH}), and $32t_{inst}$ shift clock.
  - Transfer speed = 10 MHz/4/32 = 78.1 kbps
  - Interrupt period = 8 x 32 x 4/10 MHz = 102.4 µs

❍ Coding example

```
SMR     EQU     0070H           ;Address of serial mode register
SDR     EQU     0071H           ;Address of serial data register
SIOF    EQU     SMR:7           ;Interrupt request flag bit definition
SST     EQU     SMR:0           ;Serial I/O transfer start bit definition
ILR3    EQU     007DH           ;Address of the interrupt level set register
INT_V   DSEG    ABS             ;[DATA SEGMENT]
        ORG     0FFE8H
IRQ9    DW      WARI            ;Interrupt vector setting
INT_V   ENDS
;------Main program--------------------------------------------------------------
        CSEG                    ;[CODE SEGMENT]
                                ;Assume stack pointer (SP), etc.,
                                ;have been already initialized.
           :
        CLRI                    ;Disable interrupts.
        CLRB    SST             ;Halt serial I/O transfer.
        MOV     ILR3,#11110111B ;Set interrupt level (level 1).
        MOV     SDR,#55H        ;Set transfer data (55_H).
        MOV     SMR,#01111000B  ;Clear interrupt request flag,
                                 enable output of interrupt requests, enable shift
                                 clock output (SCK2), enable serial data output (SO2),
                                 and select 32t_inst, LSB-first.
        SETB    SST             ;Start serial I/O transfer.
        SETI                    ;Enable interrupts.
           :
;------Interrupt processing routine----------------------------------------------
WARI    CLRB    SIOF            ;Clear interrupt request flag.
        PUSHW   A
        XCHW    A,T             ;Save A and T.
        PUSHW   A
        MOV     SDR,#55H        ;Update transfer data (55_H).
        SETB    SST             ;Start serial I/O transfer.
         :
        User processing
         :
        POPW    A
        XCHW    A,T             ;Restore A and T.
        POPW    A
        RETI
        ENDS
;--------------------------------------------------------------------------------
        END
```

■ Program Example for Serial Input

❍ Program specifications

- Input 8 bits of serial data from the SI1 pin of the serial I/O and generate an interrupt when transfer is completed.
- In the interrupt processing routine, read the transfer data and re-enable input.
- Operate using the external shift clock and input the shift clock from the SCK pin.

❍ Coding example

```
DDR4   EQU    0011H           ;Address of port direction register
SMR    EQU    0070H           ;Address of serial mode register
SDR    EQU    0071H           ;Address of serial data register
SIOF   EQU    SMR:7           ;Interrupt request flag bit definition
SST    EQU    SMR:0           ;Serial I/O transfer start bit definition
ILR3   EQU    007DH           ;Address of the interrupt level set register
INT_V  DSEG   ABS             ;[DATA SEGMENT]
       ORG    0FFE8H
IRQ9   DW     WARI            ;Interrupt vector setting
INT_V  ENDS
;------Main program---------------------------------------------------------------
       CSEG                   ;[CODE SEGMENT]
                              ;Assume stack pointer (SP), etc.,
                               have been already initialized.
          :
       MOV    DDR4,#00000000B ;Set P41/INT21/SCK2 and P43/INT23/SI2/SCL as inputs.
       CLRI                   ;Disable interrupts.
       CLRB   SST             ;Halt serial I/O transfer.
       MOV    ILR3,#11110111B ;Set interrupt level (level 1).
       MOV    SMR,#01001100B  ;Clear interrupt request flag bit,
                               enable output of interrupt requests,
                               set shift clock input (SCK2),
                               disable serial data output (SO2),
                               and select the external clock,
                               LSB-first.
       SETB   SST             ;Enable serial I/O transfer.
       SETI                   ;Enable interrupts.
          :
;------Interrupt processing routine-----------------------------------------------
WARI   CLRB   SIOF            ;Clear interrupt request flag.
       PUSHW  A
       XCHW   A,T
       PUSHW  A
       MOV    A,SDR           ;Read transfer data.
       SETB   SST             ;Enable serial I/O transfer.
          :
       User processing
        :
       POPW   A
       XCHW   A,T
       POPW   A
       RETI
       ENDS
;---------------------------------------------------------------------------------
       END
```

CHAPTER 19 I^{2}C INTERFACE

This chapter describes the functions and operations of the I^{2}C interface.

19.1 Overview of the I^{2}C Interface

The I^{2}C interface that supports Philips's I^{2}C bus specification and Intel's SM bus specification provides master/slave transmission and reception, arbitration lost detection, slave address/general call address detection, generation and detection of start/stop conditions, and buss error detection.

■ I^{2}C interface Functions (for MB89PV530, MB89P538, MB89F538/F538L, MB89537C/538C, MB89537HC/538HC, MB89537AC/538AC only)

The I^{2}C bus interface is a simple structure bidirectional bus consisting of two wires: a serial data line (SDA) and a serial clock line (SCL).

All devices connected to the serial bus must support open-drain or open-collector output and to connect a pull-up resistor with each bus line to operate.

Each device connected to the bus has a unique address, which can be specified using software. Among these devices, there is always a master-slave relation. The master functions as a master transmission device.

It is a full-fledged multi-master bus equipped with a collision detection function and communication adjustment procedure that prevent data corruption if two or more masters attempt to start data transfer simultaneously. Eight-bit bidirectional serial data can be transferred at a transfer rate of up to 100 kbps. As many ICs as required can be connected to one bus provided the upper limit of the bus capacitance (400 pF) is not exceeded.

The I^{2}C interface, which supports the Philips’s I^{2}C bus and Intel’s system management bus, has the following functions:

- Master and slave transmission and reception
- Automatic change from master to slave when arbitration lost detected
- Address comparison between slave and a general call
- Detection of data transfer direction
- Generation and detection of start/stop condition
- Generation and detection of continuous start condition
- Bus error detection
- Selection from 32 types of shift clock frequency by software
- Selection of acknowledge bit by software
- Generation and detection of acknowledge bit
- Byte-data transfer
- Noise canceller for input spikes up to 20 ns
- Selection of input buffer of I^{2}C interface and SM bus interface

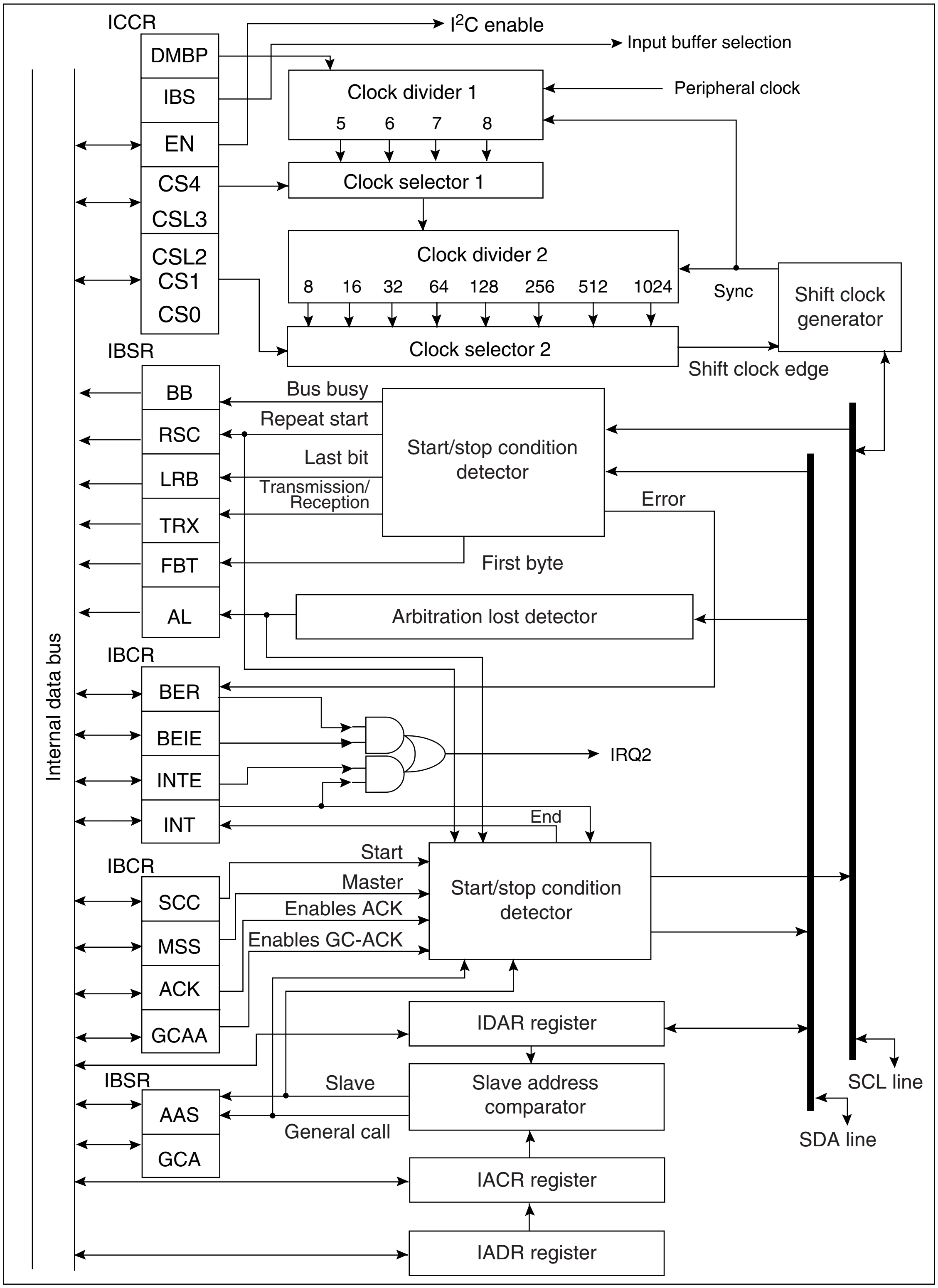
19.2 Configuration of the I^{2}C Interface

The I^{2}C interface consists of the following 10 blocks.

- **Clock controller (clock selector, clock divider, shift clock generator)**
- **Start/stop condition generator**
- **Start/stop condition detector**
- **Arbitration lost detector**
- **Slave address comparator**
- **I^{2}C bus status register (IBSR)**
- **I^{2}C bus control register (IBCR)**
- **I^{2}C clock control register (ICCR)**
- **I^{2}C address control register (IACR)**
- **I^{2}C address register (IADR)**
- **I^{2}C data register (IDAR)**

■ I^{2}C Bus Interface Block Diagram

Figure 19.2-1 Block Diagram of the I^{2}C Bus Interface



❍ **Clock controller (clock selector, clock divider, shift clock generator)**

This circuit selects and generates a shift clock of the I^2C bus based on the internal clock.

❍ **Start/stop condition generator**

When the bus is released (when the SCL and SDA lines are at a "H" level), transmitting a start condition causes the master to start communication. When the SDA line is changed from "H" to "L" when SCL = H, a start condition is generated. When a stop condition is generated, the master can stop communication. The stop condition is generated when the SDA line is changed from "L" to "H" when SCL = H.

❍ **Start/stop condition detector**

This circuit detects the start/stop condition for data transfer.

❍ **Arbitration lost detector**

This interface circuit supports the multi-master system. If two or more masters transmit data simultaneously, arbitration lost is generated. When logic level "1" is transmitted when the SDA line is at level "L", this state is regarded as arbitration lost. At this time, IBSR:AL is set to "1" and the master is changed into a slave.

❍ **Slave address comparator**

After a start condition is posted, a slave address is transmitted. This address is seven-bit data, followed by a data direction bit (R/W) as bit 8. ACK is returned only to the slave whose address matches the transmitted address.

❍ **I^2C bus status register (IBSR)**

The IBSR register indicates the status of the I^2C interface. This register is read-only.

❍ **I^2C bus control register (IBCR)**

The IBCR register is used to select the operating mode, enable/disable interrupts, enable/disable acknowledgement, and enable/disable general call acknowledgement.

❍ **I^2C clock control register (ICCR)**

The ICCR register is used to permit the operation of the I^2C interface and select the shift clock frequency.

❍ **I^2C address control register (IACR)**

The first three bits of the IACR register are effective. These bits are readable, but writable only when I^2C is not operating. When 1 is written into these bits, the corresponding IADR bits are not compared.

❍ **I^2C address control register (IADR)**

The IADR register is used to set the slave address.

❍ **I^2C data register (IDAR)**

The IDAR register is used to transfer serial data from the MSB. While data is being received (IBSR:TRX = 0), the value of the data output is "1".

■ I^{2}C interface interrupt source

IRQ2:

An interrupt request is generated by the I^{2}C interface when the bus error interrupt request bit is enabled (IBCR: BEIE = "1") and a bus error has occurred or when the transfer end interrupt enable bit is enabled (IBCR: INTE = "1") and data transfer is completed.

19.3 Configuration of the I^{2}C Bus Interface

This section describes the I^{2}C bus interface pins, their block diagrams, registers, and interrupt function and provides a block diagram of the pins.

■ Pins Related to the I^{2}C Bus Interface

The pins related to the I^{2}C bus interface are P42/INT22/SO2/SDA and P43/INT23/SI2/SCL.

❍ P42/INT22/SO2/SDA pin

The P42/INT22/SO2/SDA pin can function as an N-ch open drain output port (P42) or data I/O pin (SDA). Enabling I^{2}C automatically sets the P42/INT22/SO2/SDA pin as a data I/O pin.

❍ P43/INT23/SI2/SCL pin

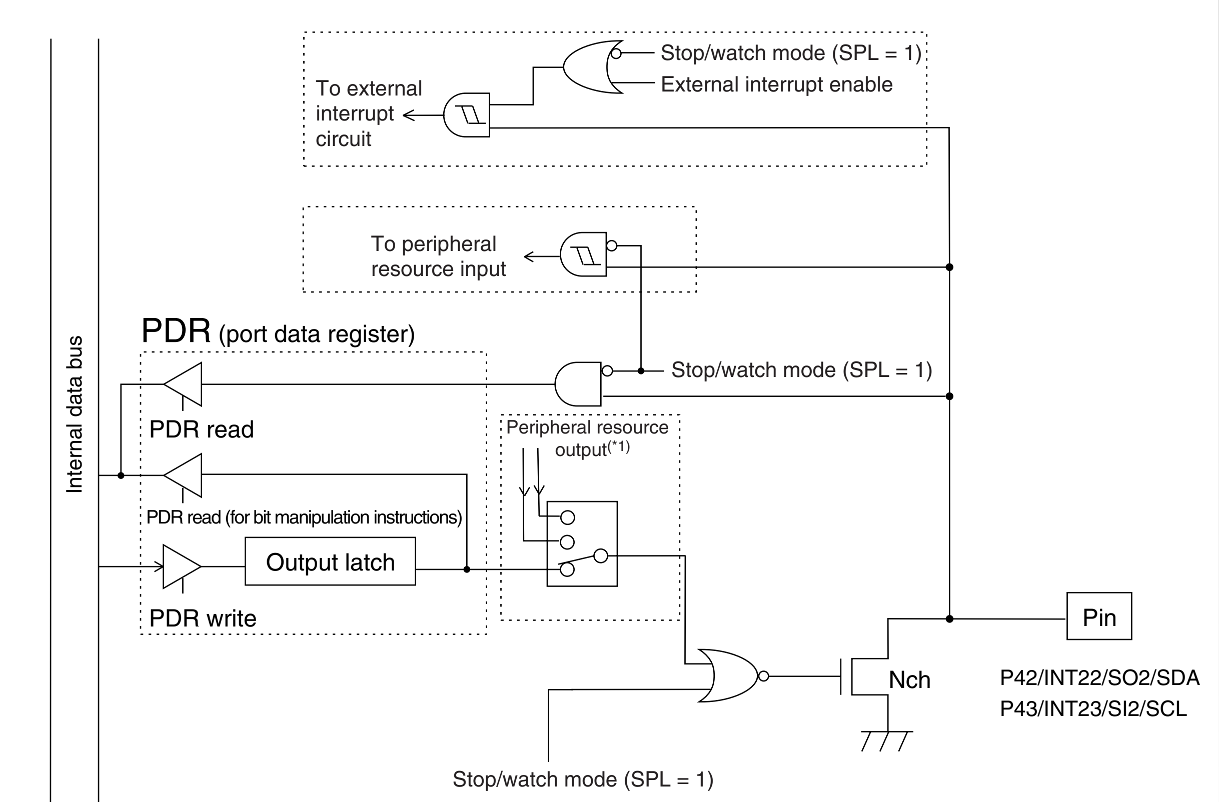
The P43/INT23/SI2/SCL pin can function as an N-ch open-drain output port (P43) or shift clock I/O pin (SCL). Enabling I^{2}C automatically sets the P43/INT23/SI2/SCL pin as a shift clock I/O pin.

■ Noise canceller on P42/INT22/SO2/SDA and P43/INT23/SI2/SCL

Noise cancellation is applied to the external signals of the P42/INT22/SO2/SDA and P43/INT23/SI2/SCL pins before the internal interface circuit. Spikes of 20 ns or less are canceled. The noise canceller cannot be used while the DMBP bit of the ICCR is enabled (only for MB89PV530).

■ Block diagram of pins related to the I2C bus interface

Figure 19.3-1 Block Diagram of Pins Related to I2C Bus Interface


Stop/watch mode (SPL = 1)
External interrupt enable
To external interrupt circuit
To peripheral resource input
PDR (port data register)
Stop/watch mode (SPL = 1)
Internal data bus
PDR read
Peripheral resource output(*1)
PDR read (for bit manipulation instructions)
Output latch
PDR write
Pin
Nch
P42/INT22/SO2/SDA
P43/INT23/SI2/SCL
Stop/watch mode (SPL = 1)

SPL : Pin status specification bit of standby control register (STBC)

*1 P42 is connected to two peripheral resource outputs.
P43 is connected to one peripheral resource output.

19.4 Registers of the I^{2}C Bus Interface

This section shows the registers related to the I^{2}C bus interface.

■ Registers Related to I^{2}C Bus Interface

Figure 19.4-1 Registers Related to I^{2}C Bus Interface

IACR (I^{2}C address control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 50 H	–	–	–	–	–	SAC2	SAC1	SAC0	-----000B
						R/W	R/W	R/W	

IBSR (I^{2}C bus status register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 51 H	BB	RSC	AL	LRB	TRX	AAS	GCA	FBT	00000000B
	R	R	R	R	R	R	R	R	

IBCR (I^{2}C bus control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 52 H	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ICCR (I^{2}C clock control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 53 H	DMBP	IBS	EN	CS4	CS3	CS2	CS1	CS0	000XXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

IADR (I^{2}C address register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 54 H	–	A6	A5	A4	A3	A2	A1	A0	-XXXXXXXB
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	

IDAR (I^{2}C data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 55 H	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

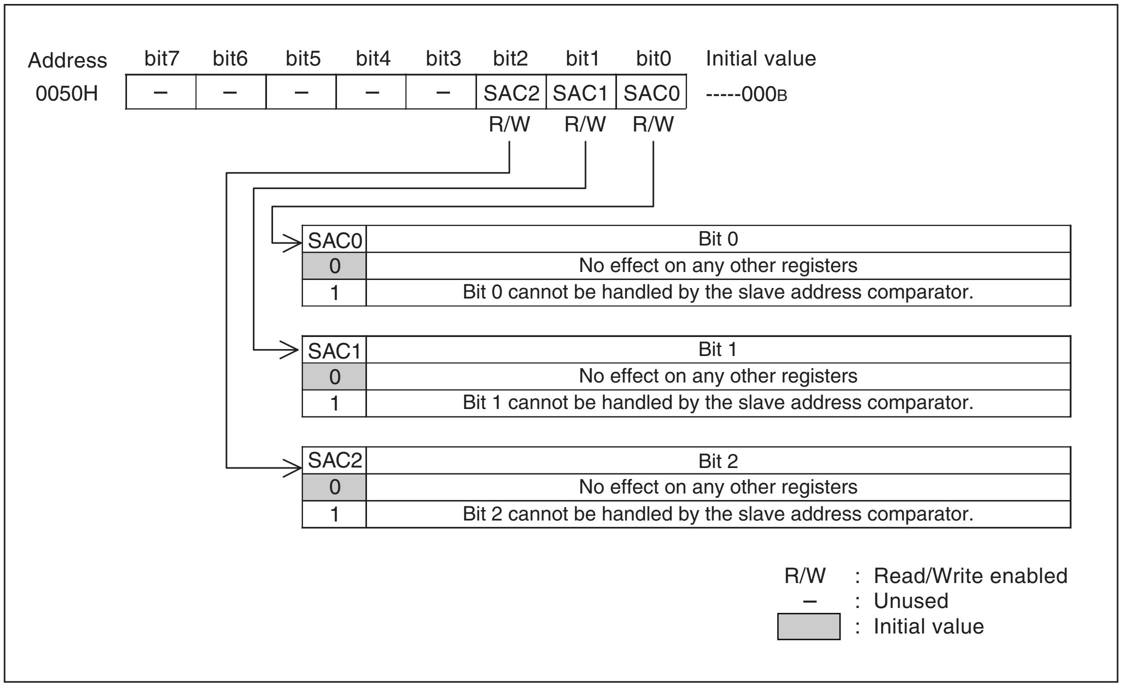
R/W : Read/write enabled
R : Read only
– : Not used
X : Undefined

19.4.1 I²C Address Control Register (IACR)

The I²C address control register (IACR) indicates the state of the interface.

■ I²C address control register (IACR)

Figure 19.4-2 I²C Address Control Register (IACR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0050H	–	–	–	–	–	SAC2	SAC1	SAC0	-----000B
						R/W	R/W	R/W	

SAC0	Bit 0
▨ 0	No effect on any other registers
1	Bit 0 cannot be handled by the slave address comparator.

SAC1	Bit 1
▨ 0	No effect on any other registers
1	Bit 1 cannot be handled by the slave address comparator.

SAC2	Bit 2
▨ 0	No effect on any other registers
1	Bit 2 cannot be handled by the slave address comparator.

R/W	:	Read/Write enabled
–	:	Unused
▨	:	Initial value

Table 19.4-1 Description of the I^{2}C Address Control Register (IACR) Bits

Bit		Description
bit7 bit6 bit5 bit4 bit3	Unused	• Values read are undefined. • Writing has no effect on the operation.
bit2	SAC2	• Writing 0 into this bit has no effect on the operation. • Setting 1 to this bit makes the sleep address comparator ignore bit 2 of the I^{2}C address register (IADR). When the other IADR bits contain an address code which is the same as that sent from the master chip, it is assumed that the address codes match.
bit1	SAC1	• Writing 0 into this bit has no effect on the operation. • Setting 1 to this bit makes the sleep address comparator ignore bit 1 of the I^{2}C address register (IADR). When the other IADR bits contain an address code which is the same as that sent from the master chip, it is assumed that the address codes match.
bit0	SAC0	• Writing 0 into this bit has no effect on the operation. • Setting 1 to this bit makes the sleep address comparator ignore bit 0 of the I^{2}C address register (IADR). When the other IADR bits contain an address code which is the same as that sent from the master chip, it is assumed that the address codes match.

19.4.2 I2C Bus Status Register (IBSR)

The IBSR register indicates the status of the interface.

■ I2C Bus Status Register (IBSR)

Figure 19.4-3 I2C Bus Status Register (IBSR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 5 1 H	BB	RSC	AL	LRB	TRX	AAS	GCA	FBT	00000000B
	R	R	R	R	R	R	R	R	

FBT	First byte detection bit
0	The received data is a byte other than the first byte when data is received
1	The received data is the first byte (address data) when data is received

GCA	General call address detection bit
0	In slave mode, the general call address is not received
1	In slave mode, the general call address is received

AAS	Addressing detection bit
0	The specific address calling address did not match in slave mode.
1	The specific address calling address matched in slave mode.

TRX	Data transfer state bit
0	Reception mode
1	Transmission mode

LRB	Last received bit
0	An acknowledge was detected in the ninth clock.
1	An acknowledge has not been detected in the ninth clock.

AL	Arbitration lost bit
0	Arbitration lost is not detected
1	Arbitration lost is generated while the master is transmitting data or "1" is written to the IBCR: MSS bit when another system is using the bus

RSC	Repeated start condition detection bit
0	Repeated start condition is not detected
1	Start condition is detected again when the bus is in use

BB	Bus busy bit
0	Stop condition is detected
1	Start condition is detected (The bus is in use.)

R/W : Read only

▨ : Initial value

Table 19.4-2 Functions of Each Bit in I^2C Bus Status Register (IBSR)

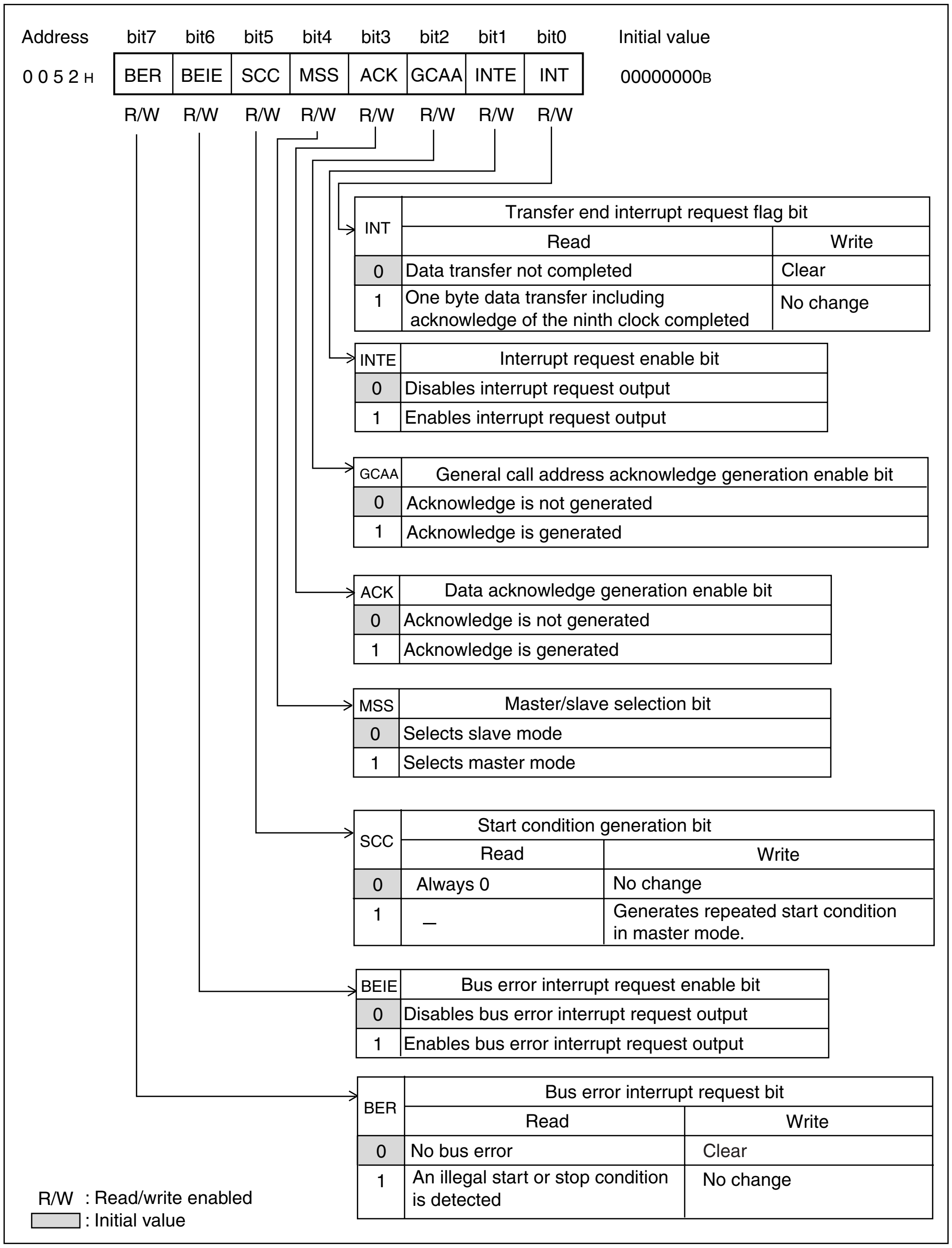
Bit name		Function
Bit 7	BB: Bus busy bit	• This bit indicates the state of the bus. • This bit is cleared when a stop condition is detected and set when a start condition is detected.
Bit 6	RSC: Repeated start condition detection bit	• This bit detects the repeated start condition. (RSC=1) This bit is set when a start condition is detected and cleared in the following state. 1. "0" is written to the IBCR: INT bit 2. This bit is not addressed in slave mode. 3. A start condition is detected during bus stop 4. A stop condition is detected
Bit 5	AL: Arbitration lost bit	• This bit detects arbitration lost. This bit is set in the following states. 1. Arbitration lost is detected when the master is transmitting data 2. "1" is written to the IBCR: MSS bit when another system is using the bus • This bit is also cleared when "0" is written to the IBCR: INT bit
Bit 4	LRB: Acknowledge storage bit	• This bit stores the SDA line value of the 9th clock when the data byte is transferred. • Cleared when an acknowledge bit is detected. (SDA = L) • Set when an acknowledge bit is not detected. (SDA = H) • Cleared with "0" when a start or stop condition is detected.
Bit 3	TRX: Data transfer state bit	• This bit indicates whether the data transfer is performed in the transmission mode or the reception mode.
Bit 2	AAS: Addressing detection bit	• This bit indicates addressing is performed in slave mode. • This bit is set when addressing is performed in slave mode and cleared when a start or stop condition is detected.
Bit 1	GCA: General call address detection bit	• This bit detects a general call address. • If this bit is set to "1" in slave mode, the general call address (00_H) is received. • This bit is cleared when a start or stop condition is detected.
Bit 0	FBT: First byte detection bit	• This bit detects the first byte • This bit is always set to "1" in the start condition. • This bit is set to "1" when a start condition is detected and cleared when "0" is written to the IBCR:INT bit or when this bit is not addressed in slave mode.

19.4.3 I^{2}C Bus Control Register (IBCR)

The IBCR register is used to select the operating mode, enables/disables interrupts, enables/disables acknowledge, and enables/disables general call acknowledge.

■ I^{2}C Bus Control Register (IBCR)

Figure 19.4-4 I^{2}C Bus Control Register (IBCR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0052H	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

INT	Transfer end interrupt request flag bit	
	Read	Write
0	Data transfer not completed	Clear
1	One byte data transfer including acknowledge of the ninth clock completed	No change

INTE	Interrupt request enable bit
0	Disables interrupt request output
1	Enables interrupt request output

GCAA	General call address acknowledge generation enable bit
0	Acknowledge is not generated
1	Acknowledge is generated

ACK	Data acknowledge generation enable bit
0	Acknowledge is not generated
1	Acknowledge is generated

MSS	Master/slave selection bit
0	Selects slave mode
1	Selects master mode

SCC	Start condition generation bit	
	Read	Write
0	Always 0	No change
1	–	Generates repeated start condition in master mode.

BEIE	Bus error interrupt request enable bit
0	Disables bus error interrupt request output
1	Enables bus error interrupt request output

BER	Bus error interrupt request bit	
	Read	Write
0	No bus error	Clear
1	An illegal start or stop condition is detected	No change

R/W : Read/write enabled

▭ : Initial value

Table 19.4-3 Functions of Each Bit in I^{2}C Bus Control Register (IBCR)

Bit name		Function
Bit 7	BER: Bus error interrupt request flag bit	• This bit clears a bus error interrupt and detects a bus error. • When a bus error is detected, "0" is written and the bus • error interrupt is cleared. • When "1" is written, there is no change and no effect on others. • When an illegal start or stop condition is detected during data transfer, this bit is set. • When this bit is set, the I^{2}C bus interface enable bit in the ICCR register (ICCR:EN) is cleared, the I^{2}C bus interface enters stop mode, and data transfer is terminated.
Bit 6	BEIE: Bus error interrupt request enable bit	• This bit enables (BEIE = 1) or disables (BEIE = 0) the generation of a bus error interrupt request. • When this bit is set and BER = 1, an interrupt request is sent to the CPU.
Bit 5	SCC: Start condition generation bit	• When this bit is set, a repeated start condition in master mode is generated. (SCC = 1) • No change when "0" is written. • The read value of this bit is always "0". **Note:** • Do not write SCC = 1 and MSS = 0 simultaneously. • If "1" is written to SCC when INT = 0, setting the SCC bit to "1" has the higher priority, and a start condition is generated.
Bit 4	MSS: Master/slave selection bit	• This bit selects the slave mode (MSS = 0) or the master mode (MSS = 1). • When this bit is cleared to "0", a stop condition is generated and the master mode is switched to the slave mode after transfer is completed. • When this bit is set to "1", the slave mode is switched to the master mode, a start condition is generated, and transfer is started. • If arbitration lost is generated when the master is transmitting data, this bit is cleared and the master mode is switched to the slave mode. **Note:** • Do not write SCC = 1 and MSS = 0 simultaneously. • If "0" is written to MSS when INT = 0, "0" in the MSS bit has a higher priority and a stop condition is generated.
Bit 3	ACK: Data acknowledge generation enable bit	• This bit enables or disables the output of the acknowledge bit in the 9th clock at data reception. • This bit is disabled while address data is received in slave mode. Moreover, acknowledge is sent during addressing.
Bit 2	GCAA: General call address acknowledge generation enable bit	• This bit permits output of the general call address acknowledge bit during reception in slave mode.

Table 19.4-3 Functions of Each Bit in I^{2}C Bus Control Register (IBCR) (Continued)

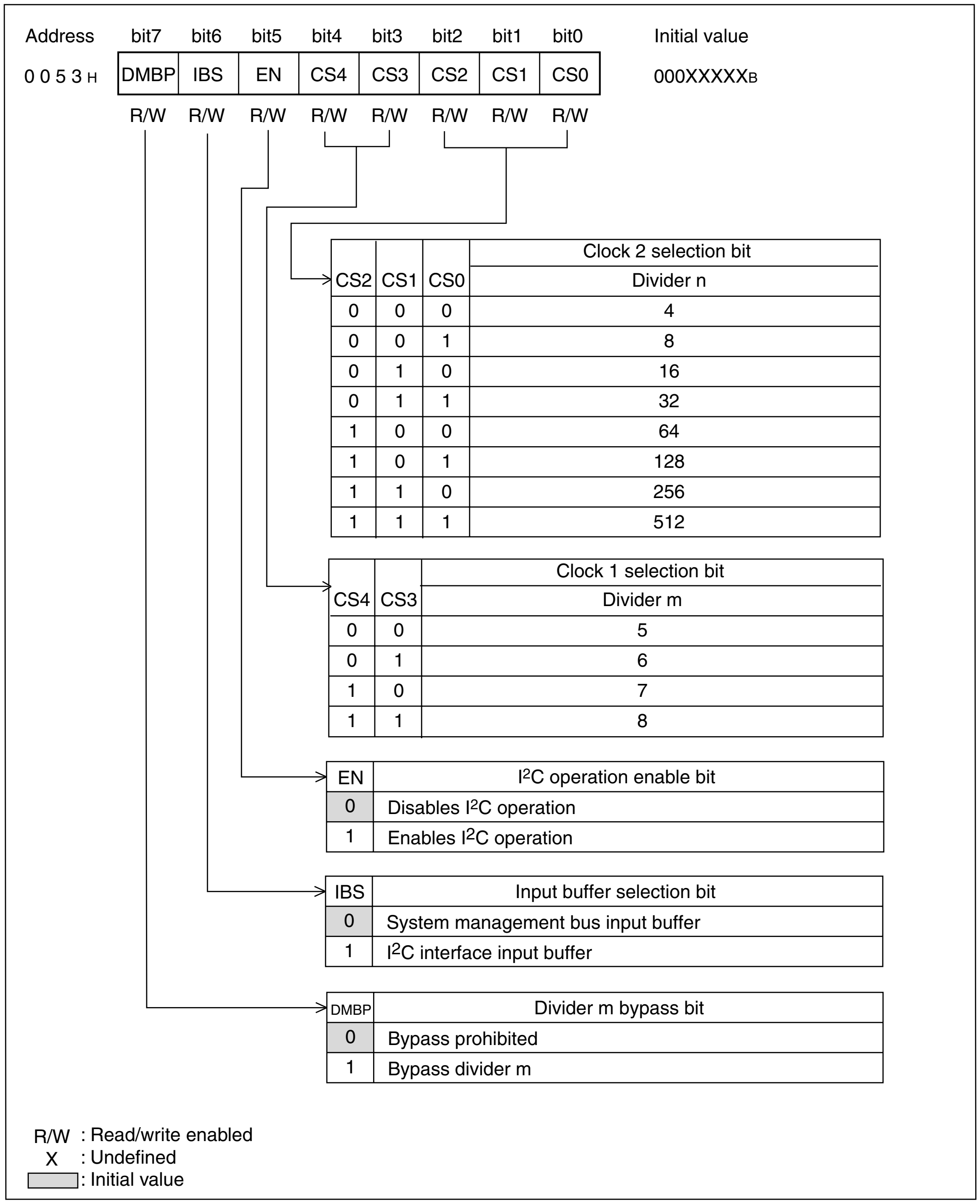
Bit name		Function
Bit 1	INTE: Transfer end interrupt request enable bit	• This bit selects whether an interrupt at the end of transfer is enabled (INTE = 1) or disabled (INTE = 0). • When this bit is set and INT is set to "1", a transfer end interrupt request is sent to the CPU.
Bit 0	INT: Transfer end interrupt request flag bit	• With this bit, the data transfer end interrupt request flag can be cleared. In addition, it can be determined whether the interrupt is detected. • When "0" is written, the transfer end interrupt request flag is cleared. When "1" is written, no change occurs. • If any of the following five conditions is met when one byte transfer including the acknowledge bit is completed (including the acknowledge bit in the 9th clock), this bit is set to "1". 1. Bus master mode 2. The calling address matches the slave address 3. A general call address is received 4. Arbitration lost is generated 5. An attempt was made to generate a start condition while another system was using the bus • When this bit is set to "1", the SCL line is kept at the "L" level. This bit is cleared when "0" is written to this bit. At this time, this macro releases the SCL line and transfers the next byte. • This bit is also cleared to "0" when a start or stop condition is generated in master mode. **Note:** • If "1" is written to SCC when INT = 0, "1" in the SCC bit has a higher priority and a start condition is generated. • If "0" is written to MSS when INT = 0, "0" in the MSS bit has a higher priority and the stop condition is generated.

19.4.4 I^2C Clock Control Register (ICCR)

The ICCR register is used to permit the operation of the I^2C and select the shift clock frequency.

■ I^2C Clock Control Register (ICCR)

Figure 19.4-5 I^2C Clock Control Register (ICCR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0053H	DMBP	IBS	EN	CS4	CS3	CS2	CS1	CS0	000XXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

			Clock 2 selection bit
CS2	CS1	CS0	Divider n
0	0	0	4
0	0	1	8
0	1	0	16
0	1	1	32
1	0	0	64
1	0	1	128
1	1	0	256
1	1	1	512

		Clock 1 selection bit
CS4	CS3	Divider m
0	0	5
0	1	6
1	0	7
1	1	8

EN	I^2C operation enable bit
0	Disables I^2C operation
1	Enables I^2C operation

IBS	Input buffer selection bit
0	System management bus input buffer
1	I^2C interface input buffer

DMBP	Divider m bypass bit
0	Bypass prohibited
1	Bypass divider m

R/W : Read/write enabled
X : Undefined
▨ : Initial value

Table 19.4-4 Functions of Each Bit in I^{2}C Clock Control Register (ICCR)

Bit name		Function
Bit 7	DMBP: Divider m bypass bit	• This bit is used to bypass the m divider for generating a shift clock frequency. • When "0" is written, the value set in CS3 and CS4 becomes the value of the m divider. • When "1" is written, the m divider is bypassed. This is equivalent to m = 1. • In read cycle, the present set value can be read. • When n = 8 (CS2 = CS1 = CS0 = 0), do not set this bit.
Bit 6	IBS: Input buffer selection bit	• This bit is used to select the characteristics of the input buffer. • Writing 0 selects the system management bus input buffer. • Writing 1 selects the I^{2}C input buffer. • The current value is read from this bit.
Bit 5	EN: Multi-address I^{2}C operation permission bit	• This bit permits the operation of the multi-address I^{2}C interface (EN = "1"). • When the bit is "0", each bit of the IBSR and IBCR registers (excluding BER and BEIE bits) is cleared to "0". • When the IBCR:BER bit is set, the bit is cleared. • This bit must be enabled to write all the I^{2}C registers.
Bit 4 Bit 3	CS4, CS3: Clock 1 selection bit	• This bit sets shift clock frequency. • Shift clock frequency Fsck is determined by the following formula. $Fsck = \frac{2F_{inst}}{m \times n + 4}$ Where, t_{inst} is an instruction cycle (the clock in the SYCC selected by the CS bit). When DMBP is "0", m is selected by CS4 and CS3. When DMBP is "1," m is "1." n is selected by CS2, CS1, and CS0.
Bit 2 Bit 1 Bit 0	CS2, CS1, CS0: Clock 2 selection bit	

Note:

The I^{2}C interface described in this section is an I^{2}C bus of the standard mode. Therefore, a shift clock frequency of up to 100 KHz can be set.

19.4.5 I^{2}C Address Register (IADR)

The IADR register is used to set the slave address.

■ I^{2}C Address Register (IADR)

Figure 19.4-6 I^{2}C Address Register (IADR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0054H	-	A6	A5	A4	A3	A2	A1	A0	-XXXXXXXB
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Read/write enabled
X : Undefined

In slave mode only, this register specifies a valid slave address. The address consists of 7 bits. The master sends the address using 8 bits by adding a R/W bit to its end.

In slave mode, after the reception of an address sent from the master, values in the lower 7 bits of the IDAR register are compared with those of this register to judge the addressing.

19.4.6 I^{2}C Data Register (IDAR)

The IDAR register is used to set transmission data and to store received data.

■ I^{2}C Data Register (IDAR)

Figure 19.4-7 I^{2}C Data Register (IDAR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 5 5 $_H$	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Read/write enabled
X : Undefined

In master mode, the data written in the register is shifted to the SDA line bit by bit from the MSB bit.

The write side in this register is made up of a double buffer. When the bus is in use (IBSR: BB = 1), written data is loaded to the eight-bit shift register when the transfer of the present byte is completed. The data in the shift register is shifted and output to the SDA line bit by bit. The value written to this register has no effect on the present data transfer. Also in slave mode, the same function can be used after the address is determined.

When IBCR: INT = 1 at data reception (IBSR: TRX = 0), the received data can be read from this register. In other cases, "FF$_H$" is always read from this register. During data transfer (IBSR:TRX = 1), however, the value of the shift register can be read from the IDAR register.

19.5 I^2C Interface Interrupts

The I^2C interface may generate an interrupt request when the data transfer is completed, a bus error has occurred, or a timeout is detected.

■ Interrupt at Bus Error

When the following conditions are met, a bus error is assumed to have occurred and the I^2C interface is stopped.

- When a violation of an I^2C bus standard definition is detected during data transfer (including transfer of the ACK bit).
- When a violation of an I^2C bus standard definition is detected while the bus is idle.
- When the SCL line is set to the "L" level at the time a start condition is generated.
- When a retransmission start condition is detected (IBSR:RSC = 1) in bus master mode.

If the bus error interrupt request enable bit is enabled (IBCR:BEIE = 1) at this time, an interrupt request is output to the CPU. Clear the interrupt request by writing "0" to the IBCR:BER bit in the interrupt processing routine.

If a bus error has occurred in spite of the IBCR:BEIE bit value, the BER bit is set to "1".

■ Interrupt at Data Transfer Completion

The I^2C interface is used to transmit data to and receive data from the SDA line one bit at a time. One data byte is defined to always consist of eight bits. Data can be changed only while the SCL is set to "L." While the SCL is set to "H," the data must be stabilized. One clock pulse transfers 1 bit of data, starting with the MSB. A byte of data must arrive after the SDA line is pulled down at the 9th clock and an acknowledge signal has been received from the receiving device. That is, nine clock pulses are required to completely transfer one byte of data.

When data transfer is completed and the transfer end interrupt request enable bit is enabled (IBCR: INTE = 1), an interrupt request (IRQ2) is output to the CPU. Clear the interrupt request by writing "0" to the INT bit in the interrupt processing routine.

If data transfer is completed in spite of the INTE bit value, the INT bit is set to "1".

■ Register and Vector Table Address Related to Interrupt of I^2C Interface

Table 19.5-1 "Register and Vector Table Address Related to Interrupt of I^2C Interface" shows the register and vector table for I^2C interface interrupts.

Table 19.5-1 Register and Vector Table Address Related to Interrupt of I^2C Interface

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Bit to be set		Upper	Lower
IRQ2	ILR1 ($007B_H$)	L21 (bit 5)	L20 (bit 4)	$FFF6_H$	$FFF7_H$

For interrupt operation, see Section 3.4.2 "Interrupt Processing".

19.6 Operation of the I^{2}C Interface

The I^{2}C interface is a serial data base of 8-bit data synchronized with the shift clock.

■ I^{2}C Bus System

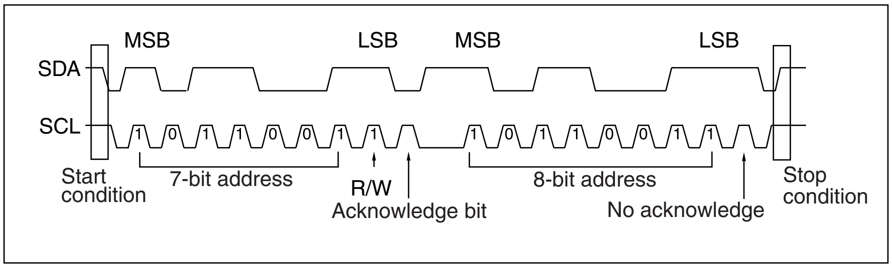
The I^{2}C bus system uses a serial data line (SDA) and a serial clock line (SCL) to transfer data. All connected devices require an open-drain or open collector output. The logic function is used by connecting a pull-up resistor.

Each device connected to the bus has a unique address and can be set by software. Among the devices, simple master/slave relations are established and master devices function as master transmitters or master receivers. The I^{2}C interface is a full-fledged multi-master bus equipped with collision detection and arbitration functions so that data destruction can be prevented even if two or more masters attempt to start data transfer simultaneously.

■ I^{2}C Bus Protocol

Figure 19.6-1 "Data Transfer Example" shows the format required for data transfer.

Figure 19.6-1 Data Transfer Example



After a start condition (S) is generated, a slave address is transmitted. This address is a seven-bit address followed by a data direction bit (R/W) as bit 8. Data transfer is always ended with the master stop condition (P). It is also possible to address to another slave without generating a stop condition by generating a repeated start condition (Sr).

■ Start Condition

While the bus is released (both the SCL and SDA lines are "H"), the master generates a start condition to start transmission. As indicated in Figure 19.6-1, changing the SDA line state from "H" to "L" while SCL is "H" generates a start condition, then a start of transmission (hereinafter referred to as "bus busy") is posted to devices connected to the bus. The two methods for generating a start condition are shown as follows.

- Writing "1" to the IBCR: MSS bit in states where the I^{2}C bus is not used (IBCR: MSS = 0, IBSR: BB = 0, IBCR: INT = 0, IBSR: AL = 0). Thereafter, IBSR: BB is set to "1" to indicate bus busy.
- Writing "1" to the IBCR: SCC bit in interrupt states in bus master mode (IBCR: MSS = 1, IBSR: BB = 1, IBCR: INT = 1, IBSR: AL = 0) and generates a repeated start condition.

Even if "1" is written to the IBCR:MSS bit or "1" is written to the IBCR:SCC bit under conditions other than the above, it is ignored. If "1" is written to the IBCR:MSS bit while another system is using the bus (in the idle state), the IBSR:AL bit is set to "1".

■ Addressing

❍ Addressing in master mode

In master mode, after the start condition is generated, "1" is set to IBSR:BB and "1" is set to IBSR:TRX. Then, the address data in the IDAR register (IDAR:D7 to D1) is output from the MSB. This address data consists of 8 bits: 7 bits for the slave address and 1 R/W bit (IDAR:DO) for indicating the direction of data transmission.

After the address data is transmitted, the master receives an acknowledgement from the slave (the SDA line is set to "L" by the 9th clock, the master receives the acknowledgement bit from the receiving device), and then reverses bit 0 of the received data (IDAR:DO after transmission) and stores it in the IBSR:TRX bit.

❍ Addressing in slave mode

In slave mode, the BB and TRX bits in the IBSR register are set to "1" and "0," respectively, after a start condition is detected and data from the master is received by the IDAR register. After receiving the address data, the IDAR and IADR registers are compared. If the values match, IBSR:AAS is set to "1" and an acknowledgement is sent to the master. Thereafter, bit 0 of the received data (IDAR:DO after reception) is stored in the IBSR:TRX bit.

■ Data Transfer

After addressing of a slave has been completed, data can be transmitted and received in byte units in the direction determined by the R/W bit sent by the master.

Each byte output to the SDA line is fixed to 8 bits. As shown in Figure 19.6-1 "Data Transfer Example", data (SDA line) is changeable only when the SCL line is at Level L. While the SCL line is at H, the state must be stable. With the MSB at the head, each bit of data is transmitted with one clock pulse. Each byte has an acknowledgement bit (the SDA line is set to "L" by the 9th clock and the master receives the acknowledgement from the receiving device). Therefore, 9 clock pulses are required to transfer one complete data byte.

■ Acknowledge

Acknowledge is transmitted from the receiving end for the 9th clock of data byte transfer from the transmitting end.

When data is received, the IBCR:ACK bit is used to enable or disable acknowledgement. When data is transmitted, acknowledgement from a receiving end is stored into the IBSR:LRB bit.

Upon transmission from a slave, if no acknowledgement is received from the master, 0 is set to IBSR:TRX, and then the slave enters reception mode. The master can generate a stop condition when the slave releases the SCL line or repeatedly generates a start condition.

■ Stop Condition

By generating a stop condition, the master can release the bus to terminate communication. A stop condition can be generated by changing the SDA line from "L" to "H" when the SCL line is at the "H" level. The master can generate start conditions continuously without generating a stop condition. This is called the repeated start condition.

In bus master mode, a stop condition is generated by writing "0" to the IBCR: MSS bit in the interrupt state (IBCR: MSS = 1, IBSR: BB = 1, IBCR: INT = 1, IBSR: AL = 0) and the master mode is switched to the slave mode.

Even if "0" is written to the IBCR: MSS bit in other the above, it is ignored.

■ Arbitration

This interface circuit is a full-fledged multi-master bus that can connect two or more masters. If a master transfers data and another master transfers data simultaneously, an arbitration is generated.

An arbitration occurs in the SDA line when the SCL line is at the "H" level. The master recognizes the occurrence of an arbitration lost when its transmission data is "1" and data on the SDA line is at the "L" level, and then it sets data output to off and sets the IBSR: AL bit to "1". When the IBSR: AL is set to "1", "0" is written to IBCR: MSS and IBSR: TRX. As a result, the TRX is cleared and the master mode is switched to the slave reception mode.

Note:

When the bus is used as described above, an attempt to generate a start condition sets IBSR:AL to 1. However, IBCR:MSS is set to 1.

19.7 Notes on Using the I^{2}C Bus Interface

This section describes precautions to take when using the I^{2}C bus interface.

■ Notes on Using the I^{2}C Bus Interface

❍ Precaution in Setting the I^{2}C Bus Interface Register

- Before writing to the bus control register (IBCR), the I^{2}C interface must be enabled (ICCR: EN).
- When the master slave selection bit (IBCR: MSS) is set, transfer starts.

❍ Precaution in Setting the Shift Clock Frequency

- To calculate the shift clock frequency using the F_{sck} expression (1) in Table 19.4-4 "Functions of Each Bit in I^{2}C Clock Control Register (ICCR)", it is necessary to know the values of m, n, and DMBP.
- When m is 5 (ICCR:CS4 = CS3 = 0) and n is 8 (ICCR:CS2 = CS1 = CS0 = 0), the DMBP value cannot be selected. Other combinations do not present a problem.

❍ Precautions Related to Priority when Contention Occurs during Transfer of the Next Byte, Occurrence of a Start Condition, and Occurrence of a Stop Condition

- Contention of the next byte transfer and stop condition

  When "0" is written to IBCR: MSS in states where IBCR: INT is cleared, the MSS bit has a higher priority and a stop condition is generated.
- Contention of the next byte transfer and start condition

  When "1" is written to IBCR: SCC in states where IBCR: INT is cleared, the SCC bit has a higher priority and a start condition is generated.

❍ Precaution on Setting with Software

- Do not select the repeated start condition (IBCR: EN = 0) and the slave mode (IBCR: MSS = 0) at the same time.
- In those states in which the interrupt request flag bits (BER and INT in the IBCR register) are set to "11_B" and the interrupt request enable bits are enabled (BEIE and INTE in the IBCR register are set to "11_B"), recovery from interrupt processing is not possible. Clear the BER and INT bits in the IBCR register.
- When I^{2}C operation is not permitted (ICCR:EN = 0), all the bits of the bus status register IBSR and the bus control register IBCR (excluding the bus error BER bit and the bus error enable BEIE bit) are cleared.

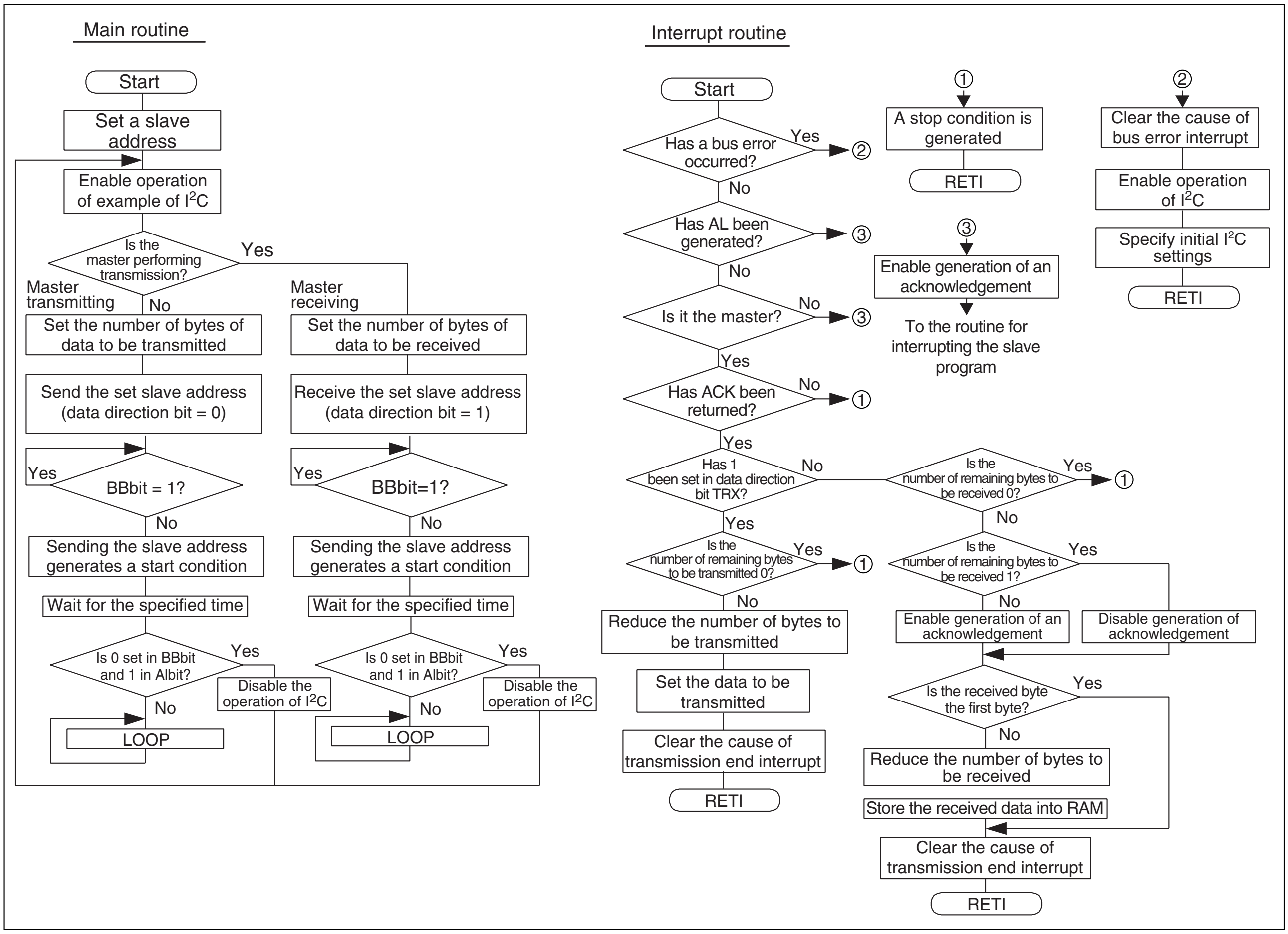
❍ Input buffer selection (for other than MB89567A)

This I^{2}C interface can support the I^{2}C bus and SM bus. Therefore, select an appropriate input buffer characteristics depending on the bus interface system to be used. The characteristics of this input buffer can be selected by using the IBS bit of the ICCR register. Specifying 0 for the bit sets the SM bus, while specifying 1 sets the I^{2}C bus.

19.8 I^{2}C Bus Interface Flowcharts

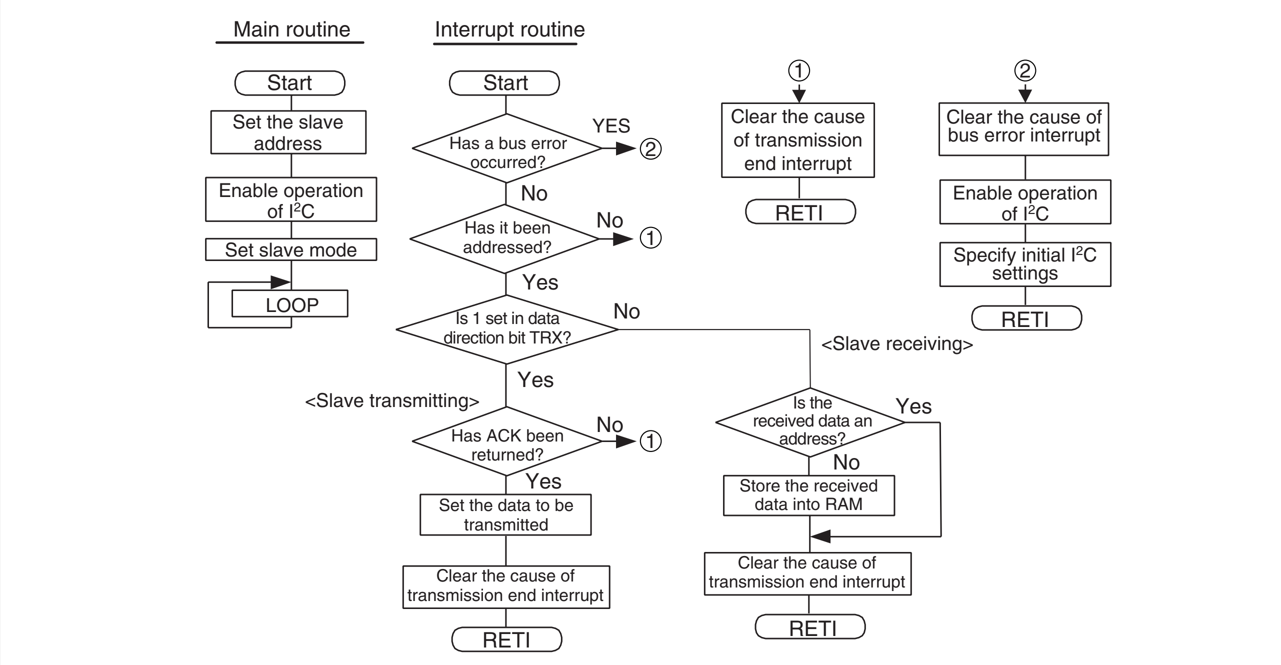
This section provides sample operation flowcharts for the I^{2}C bus interface.

■ Sample flowchart of an I^{2}C master transmission/reception program



Main routine
Start
Set a slave address
Enable operation of example of I²C
Is the master performing transmission?
Yes
No
Master transmitting
Master receiving
Set the number of bytes of data to be transmitted
Set the number of bytes of data to be received
Send the set slave address (data direction bit = 0)
Receive the set slave address (data direction bit = 1)
Yes
BBbit = 1?
No
Yes
BBbit=1?
No
Sending the slave address generates a start condition
Sending the slave address generates a start condition
Wait for the specified time
Wait for the specified time
Is 0 set in BBbit and 1 in Albit?
Yes
No
Disable the operation of I²C
LOOP
Is 0 set in BBbit and 1 in Albit?
Yes
No
Disable the operation of I²C
LOOP
Interrupt routine
Start
Has a bus error occurred?
Yes
②
No
Has AL been generated?
③
No
Is it the master?
No
③
Yes
Has ACK been returned?
No
①
Yes
Has 1 been set in data direction bit TRX?
No
Yes
Is the number of remaining bytes to be transmitted 0?
Yes
①
No
Reduce the number of bytes to be transmitted
Set the data to be transmitted
Clear the cause of transmission end interrupt
RETI
Is the number of remaining bytes to be received 0?
Yes
①
No
Is the number of remaining bytes to be received 1?
Yes
No
Enable generation of an acknowledgement
Disable generation of acknowledgement
Is the received byte the first byte?
Yes
No
Reduce the number of bytes to be received
Store the received data into RAM
Clear the cause of transmission end interrupt
RETI
①
A stop condition is generated
RETI
③
Enable generation of an acknowledgement
To the routine for interrupting the slave program
②
Clear the cause of bus error interrupt
Enable operation of I²C
Specify initial I²C settings
RETI

■ Sample flowchart of an I2C slave transmission/reception program


Main routine
Interrupt routine
Start
Set the slave address
Enable operation of I2C
Set slave mode
LOOP
Start
Has a bus error occurred?
YES
②
No
Has it been addressed?
No
①
Yes
Is 1 set in data direction bit TRX?
No
<Slave receiving>
Yes
<Slave transmitting>
Has ACK been returned?
No
①
Yes
Set the data to be transmitted
Clear the cause of transmission end interrupt
RETI
Is the received data an address?
Yes
No
Store the received data into RAM
Clear the cause of transmission end interrupt
RETI
①
Clear the cause of transmission end interrupt
RETI
②
Clear the cause of bus error interrupt
Enable operation of I2C
Specify initial I2C settings
RETI

19.9 Program Example of the I^{2}C Bus Interface

This section shows a sample program for the I^{2}C bus interface.

■ Program Example for Master Transfer Mode

❍ Processing specifications

- Transfer rate: 100kbps
- Use of a buffer with input features that support the SM bus
- Setting in master transfer mode
- Data (64_H) transmission to the slave at address 19_H
- The following equation shows the m and n values of the ICCR register required to set the serial transfer rate (Fsck) at about 100 kbps when the oscillation frequency (Fc) is 10MHz.

$$F_{sck} = \frac{2F_{inst}}{m \times n + 4} = \frac{2(10\ MHz/2/2)}{6 \times 8 + 4} = 100\ kbps$$

$F_{inst} = 1/t_{inst} \times 10$ MHz/2/2=2.5 MHz
m is selected by CS4 and CS3 of the ICCR register.
n is selected by CS2, CS1, and CS0 of the ICCR register.
m × n = 5 MHz/100 KHz - 4 = 46 (app. 6 × 8)
The ICCR register value (bits 4 to 0) is 01000_B and
ICCR:DMBP (bit 7) is set to 0_B.

❍ Coding example

```
IBSR    EQU     0051H             ;Address of the I2C bus status register
IBCR    EQU     0052H             ;Address of the I2C bus control register
ICCR    EQU     0053H             ;Address of the I2C clock control register
IADR    EQU     0054H             ;I2C address register
IDAR    EQU     0055H             ;I2C data register

EN      EQU     ICCR:5            ;Define the I2C bus interface enable bit.
BER     EQU     IBCR:7            ;Definition of the bus error interrupt request
                                   flag bit
INT     EQU     IBCR:0            ;Definition of the transfer end  interrupt
                                   request flag bit
ILR1    EQU     007BH             ;Address of the interrupt level setting  register
INT_V   DSEG    ABS               ;[DATA SEGMENT]
        ORG     0FFF6H
IRQ2    DW      WARI              ;Interrupt vector setting
INT_V   ENDS
;---------Main program------------------------------------------------------------
        CSEG                      ;[CODE SEGMENT]
                                  ;Assume that the stack pointer (SP) and other
                                   necessary elements have already been
                                   initialized.
          :
        CLRI                      ;Disables interrupts.
        SETB    EN                ;Enable I2C operation.
        MOV     ILR1,#11011111B   ;Set the interrupt level (level 1).
        MOV     ICCR,#028H        ;Enable I2C operation and select 100 kHz for
                                   the shift clock frequency.
        SETI                      ;Enables interrupts.
        MOV     IADR,#10110010B   ;Set the local address to B2H.
        MOV     IDAR,#00110010B   ;Transfer the calling address as 19H and SET R/W
                                   permission to write.
        MOV     IBCR,#01011110B   ;Enable the interrupt request flag, set master
                                   mode, and enable acknowledge.
STA_CON MOV     A,#01110110B      ;Wait for generation of a start condition and end
                                  transfer of the first byte.
        OR      A,IBSR            ;
        CMP     A,#11111111B      ;
        BNZ     STA_CON           ;
        MOV     IDAR,#01100100B   ;Transfer 64H as data.
         :
;---------Interrupt program-------------------------------------------------------
WARI    BBS     IBCR:7,BE_INT     ;Check the bus error interrupt request flag.
        BBS     IBCR:0,DTC_INT    ;Transfer end interrupt request flag
BE_INT  CLRB    BER               ;Clear the bus error interrupt request flag.
        PUSHW   A
        XCHW    A,T               ;Save A and T.
        PUSHW   A
           :
        User processing
           :
        JMP WARI_F
DTC_INT CLRB    INT               ;Clear the data transfer end interrupt request
                                   flag.
        PUSHW   A
        XCHW    A,T               ;Save A and T.
        PUSHW   A
           :
WARI_F  POPW    A
        XCHW    A,AT              ;Restore A and T.
        POPW    A
        RETI
        ENDS
;---------------------------------------------------------------------------------
        END
```

CHAPTER 20 WILD REGISTER FUNCTION

This chapter describes the functions and operations of the wild register function.

20.1 Overview of the Wild Register Function

The wild register includes a combination of a 6-byte data register, a 6-byte upper address register, a 6-byte lower address register, a 1-byte data enable register, and a 1-byte data test register. The wild register has many functions, but its main function is to replace ROM codes in the ROM space. This section describes the functions of the wild register.

■ Wild Register Function

The wild register consists of a combination of a 6-byte data register, a 6-byte upper address register, a 6-byte lower address register, a 1-byte data enable register, and a 1-byte data test register. Data in the ROM space can be replaced with arbitrary data in these registers by specifying the data and address for replacement. Data of up to six bytes can be allocated to the six data registers. Up to six addresses can be allocated to the six upper address registers and six lower address registers.

Using the wild register function, you can debug a program and apply a patch to a faulty location after masking. You can also replace any ROM code in the ROM space and search tables according to settings by other applications. The settings of the wild register are made employing a specialized communication method via a device.

Note:

The functions of the wild register cannot be debugged by a tool. Use the MB89P568 to check wild register operation on an actual machine.

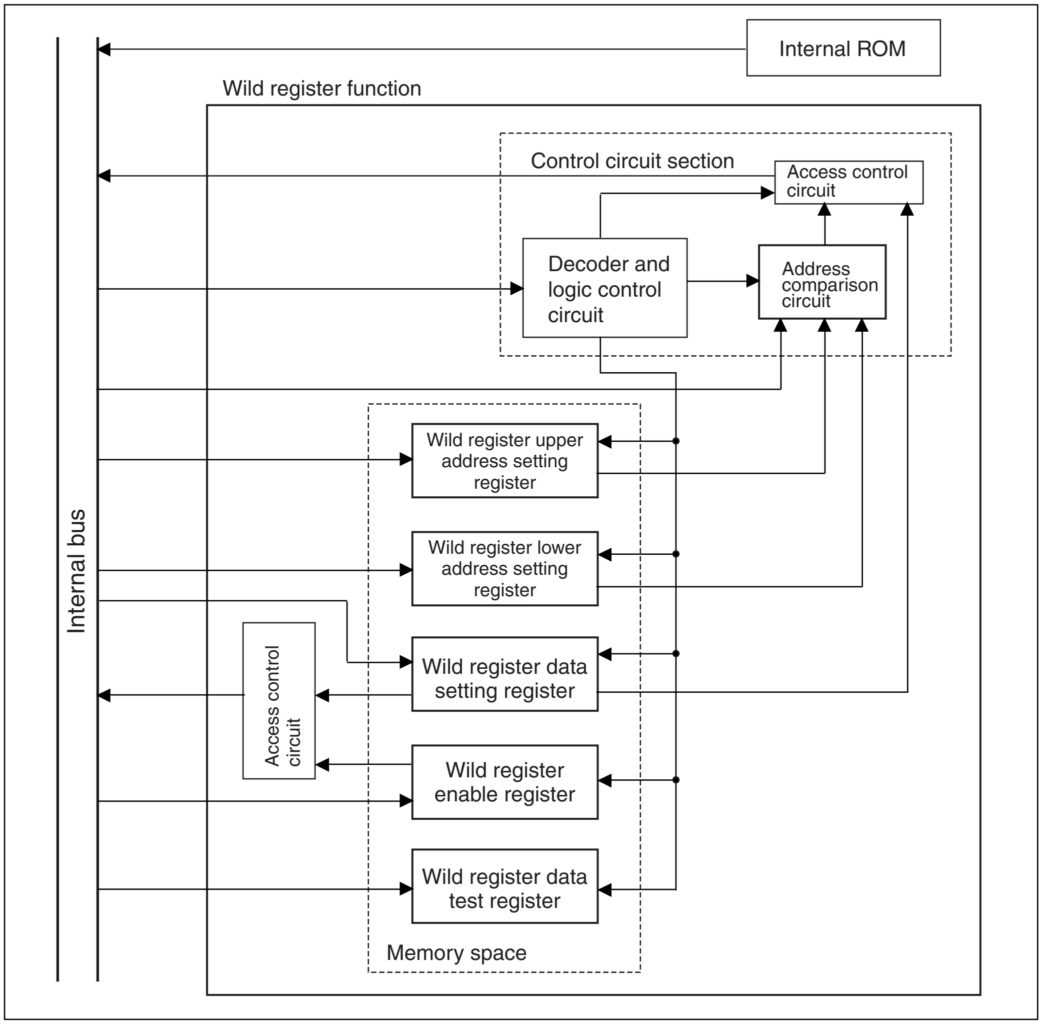
20.2 Configuration of the Wild Register Function

The wild register function consists of the following two blocks:

- **Wild register data setting registers (WRDR1 to WRDR6)**
- **Wild register upper address setting registers (WRARH1 to WRARH6)**
- **Wild register lower address setting registers (WRARL1 to WRARL6)**
- **Wild register enable register (WREN)**
- **Wild register data test register (WROR)**
- **Wild register comparison circuit and logic control circuit**

■ Block Diagram of the Wild Register Function

Figure 20.2-1 Block Diagram of the Wild Register Function


Internal ROM
Wild register function
Control circuit section
Access control circuit
Decoder and logic control circuit
Address comparison circuit
Wild register upper address setting register
Wild register lower address setting register
Wild register data setting register
Access control circuit
Wild register enable register
Wild register data test register
Memory space
Internal bus

20.3 Registers of the Wild Register Function

This section describes the registers related to the wild register function.

■ Registers Related to the Wild Register Function

Figure 20.3-1 Registers Related to the Wild Register Function

Wild register data setting register (WRDR1 to WRDR6)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C82$_H$ 0C85$_H$ 0C88$_H$ 0C8B$_H$ 0C8E$_H$ 0C91$_H$	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Wild register upper address setting register (WRARH1 to WRARH6)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C80$_H$ 0C83$_H$ 0C86$_H$ 0C89$_H$ 0C8C$_H$ 0C8F$_H$	RA15	RA14	RA13	RA12	RA11	RA10	RA09	RA08	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Wild register lower address setting register (WRARL1 to WRARL6)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C81$_H$ 0C84$_H$ 0C87$_H$ 0C8A$_H$ 0C8D$_H$ 0C90$_H$	RA07	RA06	RA05	RA04	RA03	RA02	RA01	RA00	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Wild register enable register (WREN)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0077$_H$	—	—	EN05	EN04	EN03	EN02	EN01	EN00	--000000$_B$
			R/W	R/W	R/W	R/W	R/W	R/W	

Wild register data test register (WROR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0078$_H$	—	—	DRR5	DRR4	DRR3	DRR2	DRR1	DRR0	--000000$_B$
			R/W	R/W	R/W	R/W	R/W	R/W	

R/W :	Read/write enabled
R :	Read-only enabled
— :	Not used
X :	Undefined

20.3.1 Data Setting Registers (WRDR1 to WRDR6)

The data setting registers (WRDR1 to WRDR6) contain correction data to be set using the wild register function.

■ Data Setting Registers (WRDR1 to WRDR6)

Figure 20.3-2 Setting Data Registers (WRDR1 to WRDR6)

WRDR1

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C82$_H$	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRDR2

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C85$_H$	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRDR3

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C88$_H$	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRDR4

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C8B$_H$	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRDR5

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C8E$_H$	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRDR6

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C91$_H$	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	XXXXXXXX$_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Read/write enabled
X : Undefined

Figure 20.3-3 Data Settings for Wild Registers

WRDR1 to WRDR6

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	XXXXXXXXB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit name	Description
RD07 RD06 RD05 RD04 RD03 RD02 RD01 RD00	A 1-byte register that saves data for the address assigned by WRARL and WRARH. Data is valid at the address (WRARL/WRARH) corresponding to each wild register number.

R/W : Read/write enabled
X : Undefined

Note:

Six data setting registers (WRDR1 to WRDR6) are provided. Data setting registers correspond to address setting registers (WRARH1 to WRARH6, WRARL1 to WRARL6), respectively.

20.3.2 Upper Address Setting Registers (WRARH1 to WRARH6)

The upper address setting registers (WRARH1 to WRARH6) contain the upper part of an address where data is corrected.

■ Upper Address Setting Registers (WRARH1 to WRARH6)

Figure 20.3-4 Upper Address Setting Registers (WRARH1 to WRARH6)

WRARH1

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C80H	RA15	RA14	RA13	RA12	RA11	RA10	RA09	RA08	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARH2

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C83H	RA15	RA14	RA13	RA12	RA11	RA10	RA09	RA08	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARH3

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C86H	RA15	RA14	RA13	RA12	RA11	RA10	RA09	RA08	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARH4

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C89H	RA15	RA14	RA13	RA12	RA11	RA10	RA09	RA08	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARH5

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C8CH	RA15	RA14	RA13	RA12	RA11	RA10	RA09	RA08	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARH6

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C8FH	RA15	RA14	RA13	RA12	RA11	RA10	RA09	RA08	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Read/write enabled
X : Undefined

Figure 20.3-5 Upper Address Registers (WRARH1 to WRARH6) Settings

WRARH1 to WRARH6

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
RA15	RA14	RA13	RA12	RA11	RA10	RA09	RA08	XXXXXXXX$_B$
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit name	Description
RA15 RA14 RA13 RA12 RA11 RA10 RA09 RA08	1-byte register that specifies an upper address allocated in memory. Specify an address for each upper address register.

R/W : Read/write enabled
X : Undefined

20.3.3 Lower Address Setting Registers (WRARL1 to WRARL6)

The lower address setting registers (WRARL1 to WRARL6) contain the lower part of an address where data is corrected.

■ Lower Address Setting Registers (WRARL1 to WRARL6)

Figure 20.3-6 Lower Address Setting Registers (WRARL1 to WRARL6)

WRARL1

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C81H	RA07	RA06	RA05	RA04	RA03	RA02	RA01	RA00	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARL2

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C84H	RA07	RA06	RA05	RA04	RA03	RA02	RA01	RA00	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARL3

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C87H	RA07	RA06	RA05	RA04	RA03	RA02	RA01	RA00	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARL4

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C8AH	RA07	RA06	RA05	RA04	RA03	RA02	RA01	RA00	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARL5

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C8DH	RA07	RA06	RA05	RA04	RA03	RA02	RA01	RA00	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARL6

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0C90H	RA07	RA06	RA05	RA04	RA03	RA02	RA01	RA00	XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Read/write enabled
X : Undefined

Figure 20.3-7 Lower Address Registers (WRARL1 to WRARL6) Settings

WRARL1 to WRARL6

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
RA07	RA06	RA05	RA04	RA03	RA02	RA01	RA00	$XXXXXXXX_B$
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit name	Description
RA07 RA06 RA05 RA04 RA03 RA02 RA01 RA00	1-byte register that specifies an upper address allocated in memory. Specify an address for each upper address register.

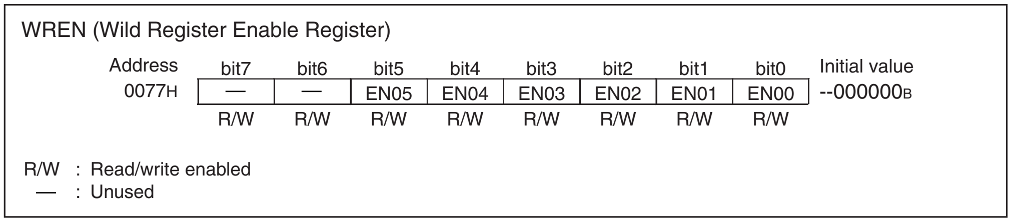
R/W : Read/write enabled
X : Undefined

20.3.4 Wild Register Enable Register (WREN)

The WREN register enables or disables operation of the wild register function for each wild register number.

■ Wild Register Enable Register (WREN)

Figure 20.3-8 Wild Register Enable Register (WREN)



WREN (Wild Register Enable Register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0077H	—	—	EN05	EN04	EN03	EN02	EN01	EN00	--000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Read/write enabled
— : Unused

Table 20.3-1 Functions of the Wild Register Enable Register (WREN)

Bit name		Function
Bit 7 Bit 6	Unused bits	The read value is undefined. Writing this bit has no effect on operation.
Bit 5	EN05	Setting this bit to 0 disables the wild register function. Setting this bit to 1 enables the wild register function. If a match is then found for the address set in WRARH6 and WRARL6, the value in the WRDR6 is output to the internal bus instead of the value in ROM.
Bit 4	EN04	Setting this bit to 0 disables the wild register function. Setting this bit to 1 enables the wild register function. If a match is then found for the address set in WRARH5 and WRARL5, the value in the WRDR5 is output to the internal bus instead of the value in ROM.
Bit 3	EN03	Setting this bit to 0 disables the wild register function. Setting this bit to 1 enables the wild register function. If a match is then found for the address set in WRARH4 and WRARL4, the value in WRDR4 is output to the internal bus instead of the value in ROM.
Bit 2	EN02	Setting this bit to 0 disables the wild register function. Setting this bit to 1 enables the wild register function. If a match is then found for the address set in WRARH3 and WRARL3, the value in the WRDR3 is output to the internal bus instead of the value in ROM.

Table 20.3-1 Functions of the Wild Register Enable Register (WREN) (Continued)

Bit name		Function
Bit 1	EN01	Setting this bit to 0 disables the wild register function. Setting this bit to 1 enables the wild register function. If a match is then found for the address set in WRARH2 and WRARL2, the value in the WRDR2 is output to the internal bus instead of the value in ROM.
Bit 0	EN00	Setting this bit to 0 disables the wild register function. Setting this bit to 1 enables the wild register function. If a match is then found for the address set in WRARH1 and WRARL1, the value in the WRDR1 is output to the internal bus instead of the value in ROM.

20.3.5 Wild Register Data Test Register (WROR)

This section describes the wild register test register and its relationship to the wild register.

■ Wild register data test register (WROR) and its relationship to the wild register

Figure 20.3-9 Wild Register Data Test Register (WROR) and Its Relationship to the Wild Register

WROR

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0078H	—	—	DRR5	DRR4	DRR3	DRR2	DRR1	DRR0	--000000B
			R/W	R/W	R/W	R/W	R/W	R/W	

Bit name	Description
---- ---- DRR5 DRR4 DRR3 DRR2 DRR1 DRR0	Bits that enable the ordinary read function for the corresponding data registers (WRDR1 to WRDR6)

R/W : Read/write enabled
— : Undefined

20.4 Operation of the Wild Register Function

This section describes the sequence of wild register function operations.

■ Wild Register Operations

A special program defining access between external memory (for example, EEPROM) and the wild register must be placed before ordinary user programs.

The procedure for setting the wild register is described below. This section does not discuss the communication method between external memory and the device.

- Write addresses at which internal ROM codes are to be modified in the upper address registers (WRARH1 to WRARH6) and lower address registers (WRARL1 to WRARL6).
- Write new code to the corresponding data registers (WRDR1 to WRDR6).
- Set the corresponding enable bits in the enable register (WREN) by writing to enable the wild register function.
- In the normal state, the data registers (WRDR1 to WRDR6) can be written to only via the address bus or data bus. These registers cannot be read directly via the address bus or data bus. To read these registers directly via the address bus or data bus, the corresponding bits in the data test register (WROR) must be set correctly. The main function of the data test register is to serve as a dedicated test function register.

Table 20.4-1 "Sequence of Wild Register Function Operations" shows the sequence setting the registers with the wild register function.

Table 20.4-1 Sequence of Wild Register Function Operations

Order	Operation	Operation example
0	Set data for replacement in the internal ROM space using the appropriate communication method.	Write the address of an internal ROM code to be modified at address $F011_H$, and write the address of the replacement data to $A6_H$ to $B5_H$. Up to six internal ROM codes can be changed.
1	Write to the first upper address register (WRARH1 to WRARH6).	Set WRARH1 = $F0_H$, and set WRARH2 to WRARH6 accordingly.
2	Write to the first lower address register (WRARL1 to WRARL6).	Set WRARL1 = 11_H, and set WRARL2 to WRARL6 accordingly.
3	Write the first new ROM code, which will replace the existing internal ROM code, to the first data register (WRDR1 to WRDR6).	Set WRDR1 = $B5_H$, and set WRDR2 to WRDR6 accordingly.
4	To replace more than one internal ROM code, repeat steps 1 to 3. Up to six addresses and data items can be set. When no more setting is required, go to the next step.	Repeat steps 1 to 3.
5	Enable the corresponding bits in the enable register (WREN).	Setting bit 0 of the WREN to "1" enables the WRDR1 register. The data in this register can be accessed if the address is set accordingly. To replace more than one ROM code, set additional addresses and WRDR registers.

Table 20.4-2 List of Wild Register Addresses

	Operation	Write	Read	RMW
WRARH1 to WRARH6	Enables reading from and writing to the addresses assigned to the registers.	Ordinary	Ordinary	Ordinary
WRARL1 to WRARL6	Enables reading from and writing to the addresses assigned to the registers.	Ordinary	Ordinary	Ordinary
WRDR1 to WRDR6	Enables writing to the addresses assigned to the registers. Reading is controlled by WROR.	Ordinary	Depends on WROR	Depends on WROR
WREN	Enables reading from and writing to the addresses assigned to the register.	Ordinary	Ordinary	Ordinary
WROR	Enables reading from and writing to the addresses assigned to the register.	Ordinary	Ordinary	Ordinary

Reference:

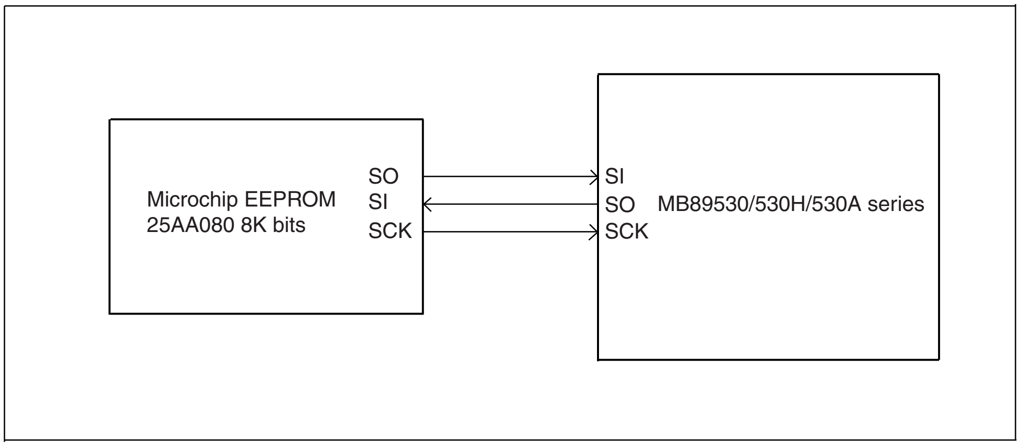
When these registers are used for the wild register function, writing to them may cause an error. Therefore, when modifying data in these registers, set the WREN value to "00_H" before modifying the data.

20.5 General Hardware Connections

This section describes the general connections between hardware units when the wild register function is used.

■ Hardware Connections

Figure 20.5-1 General Hardware Connections



When serial I/O is used as the communication interface with external EEPROM, special precautions are required. For serial I/O of the MB89530/530H/530A series, the SP-compatible method is selected as the communication method for this series. When an external memory device is selected, an SPI-compatible device needs to be selected for correct communication with the device.

CHAPTER 21 CLOCK OUTPUT

This section describes the functions and operation of clock output.

21.1 Overview of Clock Output

Clock output supplies divide by 2 waveforms of the high-speed clock and waveforms of the low-speed clock from a port. This section describes the main clock output and subclock output.

■ Clock output

The following describes the pins used for clock output:

- P30/PPG03/MCO: A general-purpose I/O port for outputting the main clock divided by 2 (P30), 6-bit PPG, and main clock. (For instance, when the main clock is 10 MHz, MCO output is set to 5 MHz.)
- P31/SCK1 (UCK1)/LMCO: A general-purpose port (P31), SIO/UART clock I/O, subclock output.

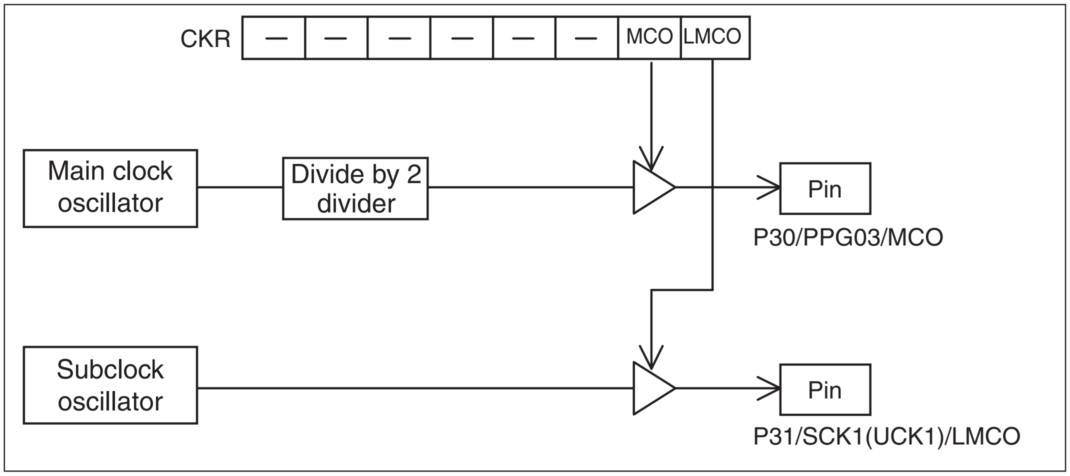
21.2 Clock Output Components

The clock output block consists of the following three components:

- **Clock output control register (CKR)**
- **Main clock oscillator**
- **Subclock oscillator**

■ Block diagram of clock output

Figure 21.2-1 Clock Output Block Diagram



❍ **Main clock oscillator**

Supplies the system main clock (10 MHz) in main watch mode.

❍ **Subclock oscillator**

Supplies the system subclock (32.768 kHz) in subwatch mode.

❍ **Clock output control register (CKR)**

Enables output of the subclock square waves and divide by 2 main clock square waves supplied from the main clock oscillator.

21.3 Clock Output Pins

This section provides a description and block diagram of the pins related to clock output.

■ Pins for the main clock output

P30/PPG03/MCO:

A general-purpose I/O port for outputting the main clock divided by 2 (P30), 6-bit PPG, and main clock. (For instance, when the main clock is 10 MHz, MCO output is set to 5 MHz.)

- MCO: Outputs divide by 2 main clock square waves.

■ Pins for the subclock output

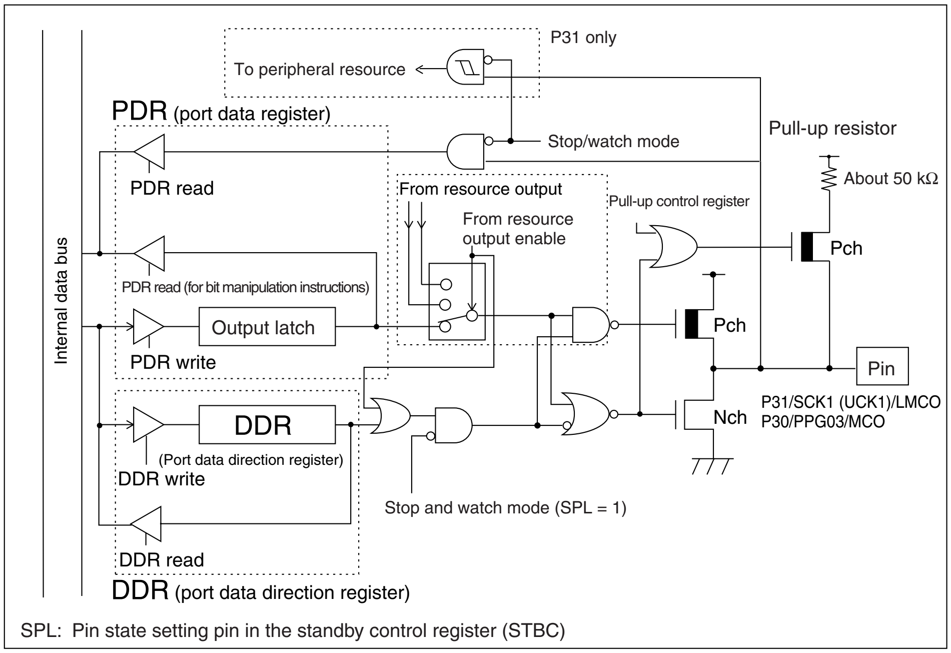
P31/SCK1 (UCK1)/LMCO:

A general-purpose I/O port (P31), SIO/UART clock I/O, subclock output.

- LMCO: Outputs square waves whose frequency is the same as that of the subclock.

■ Block diagram of P30/PPG03/MCO and P31/SCK1 (UCK1)/LMCO

Figure 21.3-1 Block Diagram of P30/PPG03/MCO and P31/SCK1(UCK1)/LMCO Pins for Clock Output



Reference:

When "pull-up resistor available" is selected using the pull-up option setting register, the pins are set to the "H" (pulled up) level, not the high impedance state, in stop mode or watch mode (SPL = 1). During a reset, however, the pull-up is disabled and the pins are set to Hi-z.

21.4 Registers for Clock Output

This section describes the registers for clock output.

■ Registers for clock output

Figure 21.4-1 Registers for Clock Output

Clock output control register (CKR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$005A_H$	—	—	—	—	—	—	MCO	LMCO	$------00_B$
							R/W	R/W	

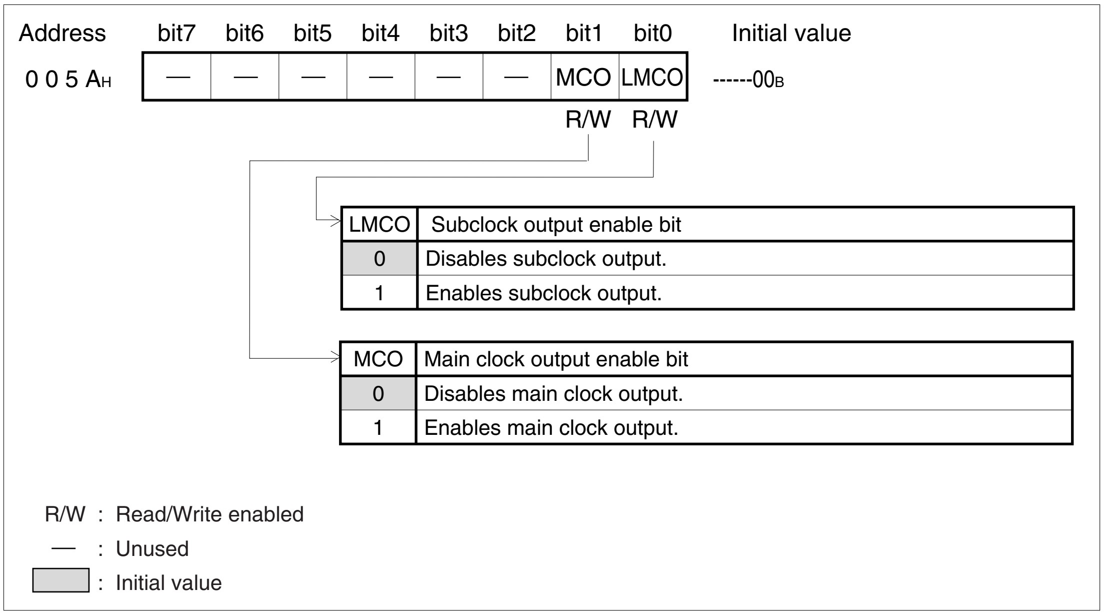
R/W : Read/Write enabled
— : Unused

21.4.1 Clock Output Control Register (CKR)

The clock output control register (CKR) is used to enable clock output.

■ Clock output control register (CKR)

Figure 21.4-2 Clock Output Control Register (CKR)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 5 AH	—	—	—	—	—	—	MCO	LMCO	------00B
							R/W	R/W	

LMCO	Subclock output enable bit
0	Disables subclock output.
1	Enables subclock output.

MCO	Main clock output enable bit
0	Disables main clock output.
1	Enables main clock output.

R/W : Read/Write enabled

— : Unused

▨ : Initial value

Table 21.4-1 Description of Clock Output Control Register (CKR) Bits

Bit		Description
bit7 bit6 bit5 bit4 bit3 bit2	Unused	• Value read are undefined. • Writing has no effect on operation.
bit1	MCO: Main clock output enable bit	• This bit enables output of waveforms of the main clock divided by 2 for P30/PPG03/MCO.
bit0	LMCO: Subclock output enable bit	• This bit enables output of the subclock waveform for P31/SCK1/(UCK1)/LMCO.

21.5 Description of Clock Output Operation

This section describes the clock output operation.

■ Description of clock output operation

Figure 21.5-1 Setting for Clock Output

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CKR	–	–	–	–	–	–	MCO	LMCO
							◎	◎

◎ : Bit used

Setting bit 0 of the clock output control register to 1 makes P31/SCK1(UCK1)/LMCO output square waves with a frequency that is the same as that of the subclock oscillator.

Setting bit 1 of the clock output control register to 1 makes P30/PPG03/MCO output divide by 2 square waves of the main clock oscillator frequency.

21.6 Notes on Use of Clock Output

This section provides notes on using the clock output.

■ Notes on use of clock output

For P30/PPG03/MCO, MCO cannot provide a clock output when PPG03 output is enabled, even if the corresponding clock control register bit is enabled. To use MCO output, disable PPG03.

For P31/SCK1(UCK1)/LMCO, LMCO cannot provide clock output when SCK1(UCK1) output is enabled, even if the clock control register bit corresponding to it is enabled. To use LMCO output, disable SCK1(UCK1).

21.7 Sample Clock Output Program

This section provides a sample program for clock output.

■ Sample clock output program

❍ Processing specification

Outputting divide by 2 square waves of the main clock oscillator frequency

❍ Coding sample

```
CKR     EQU       005AH           ;Clock control register
;----------Main program-------------------------------------------------------
        CSEG                      ;[CODE SEGMENT]
         :
        MOV       CKR,#00000010B  ;Enabling main clock output
         :
        ENDS
;-----------------------------------------------------------------------------
        END
```

CHAPTER 22 FLASH MEMORY

This chapter explains the functions and operation of the 1M-bit flash memory. The following three methods are available for writing data to and erasing data from the flash memory:

1. Parallel programmer
2. Writing/erasing data using a serial programmer
3. Executing programs to write/erase data
This chapter explains "Executing programs to write/erase data".
***: A user must create a serial programmer for writing.**

22.1 Outline of Flash Memory

The 1M-bit flash memory is mapped to the 4000_H to $FFFF_H$ bank in the CPU memory map. The functions of the flash memory interface circuit enable read-access and program-access from the CPU in the same way as mask ROM. Instructions from the CPU can be used via the flash memory interface circuit to write data to and erase data from the flash memory. Internal CPU control therefore enables rewriting of the flash memory while it is mounted. As a result, improvements in programs and data can be performed efficiently.

■ Flash Memory Features

- Sector configuration: 48K bytes x 8 bits (16 K + 8 K + 8 K + 16 K)
- Use of automatic program algorithm (Embedded Algorithm: Equivalent to MBM29LV200)
- Erase pause/restart functions provided
- Detection of completion of writing/erasing using data polling or toggle bit functions
- Detection of completion of writing/erasing using CPU interrupts
- Compatible with JEDEC standard commands
- Sector erase function (any combination of sectors)
- Minimum of 10,000 write/erase operations

Embedded Algorithm is a trademark of Advanced Micro Device, Inc.

■ Writing to/Erasing Flash Memory

The flash memory cannot be written to and read at the same time. That is, when data is written to or erased data from the flash memory, the program in the flash memory must first be copied to RAM. The entire process is then executed in RAM so that data is simply written to the flash memory. This eliminates the need for the program to access the flash memory from the flash memory itself.

■ Flash Memory Register

Compatible with JEDEC standard commands

Bit No.	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$007A_H$	INTE	RDYINT	WE	RDY	Reserved	Reserved	-	Reserved
Read/write	(R/W)	(R/W)	(R/W)	(R)	(R/W)	(R/W)	(-)	(R/W)
Initial value	(0)	(0)	(0)	(X)	(0)	(0)	(-)	(0)

22.2 Sector Configuration of the Flash Memory

Figure 22.2-1 "Flash Memory Sector Configuration" shows the sector configuration of the flash memory.

■ Sector Configuration

Figure 22.2-1 "Flash Memory Sector Configuration" shows the addresses of each sector used when access is made from the CPU and when a flash memory writing programmer is used.

Figure 22.2-1 Flash Memory Sector Configuration

Flash memory	CPU address	Programmer address(*1)
16K bytes	FFFFH	1FFFFH
	to	to
	C000H	1C000H
8K bytes	BFFFH	1BFFFH
	to	to
	A000H	1A000H
8K bytes	9FFFH	19FFFH
	to	to
	8000H	18000H
16K bytes	7FFFH	17FFFH
	to	to
	4000H	14000H

*** Programmer address**

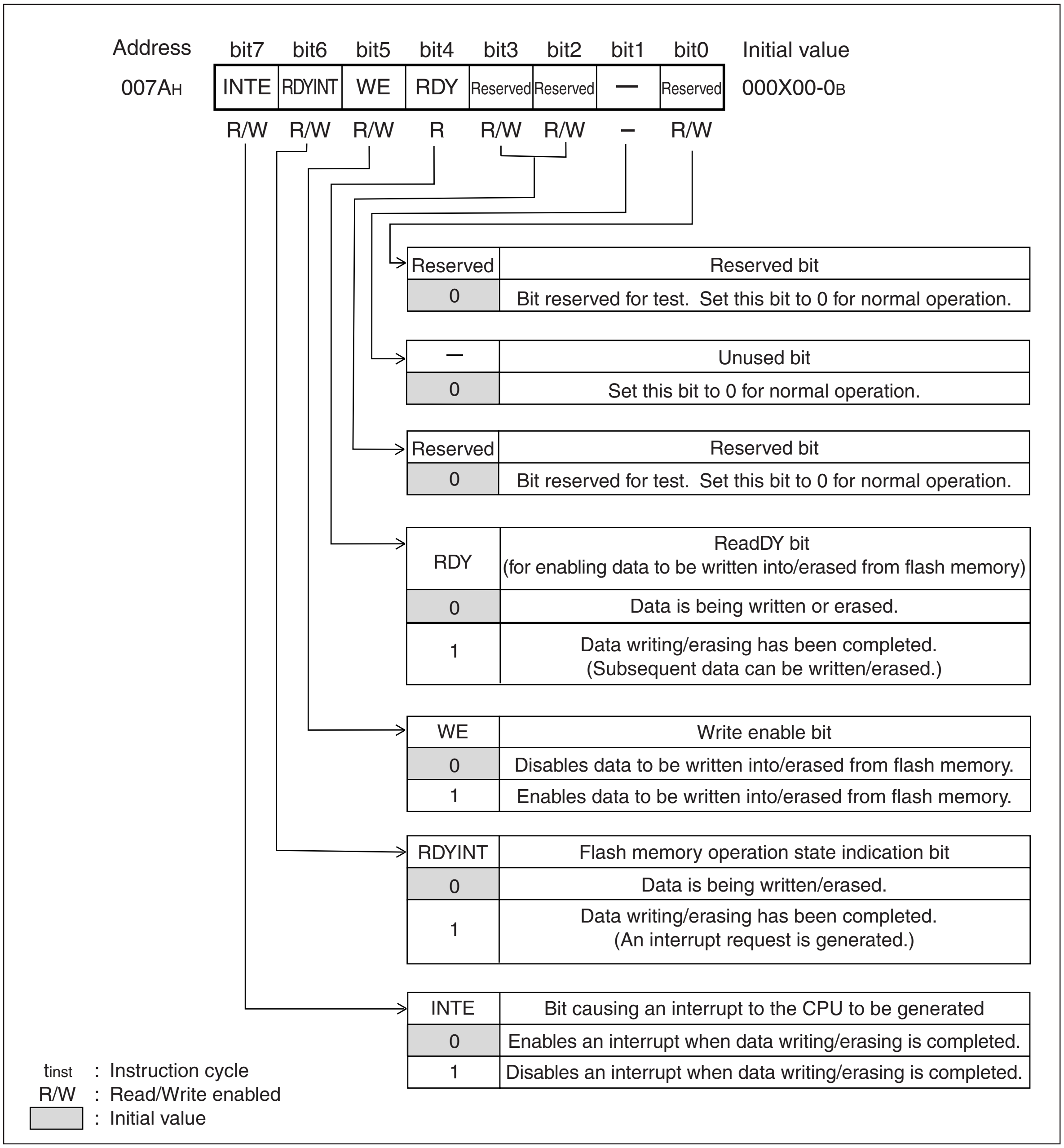
The programmer address is equivalent to the CPU address when data is written to the flash memory using a parallel programmer. When a general programmer is used for writing/erasing, this address is used for writing/erasing.

22.3 Flash Memory Control Status Register (FMCS)

The flash memory control status register (FMCS), together with the flash memory interface circuit, is used to write data to and erase data from the flash memory.

■ Flash Memory Control Status Register (FMCS)

Figure 22.3-1 Control Status Register (FMCS)



Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$007A_H$	INTE	RDYINT	WE	RDY	Reserved	Reserved	—	Reserved	$000X00\text{-}0_B$
	R/W	R/W	R/W	R	R/W	R/W	–	R/W	

Reserved	Reserved bit
0	Bit reserved for test. Set this bit to 0 for normal operation.

—	Unused bit
0	Set this bit to 0 for normal operation.

Reserved	Reserved bit
0	Bit reserved for test. Set this bit to 0 for normal operation.

RDY	ReadDY bit (for enabling data to be written into/erased from flash memory)
0	Data is being written or erased.
1	Data writing/erasing has been completed. (Subsequent data can be written/erased.)

WE	Write enable bit
0	Disables data to be written into/erased from flash memory.
1	Enables data to be written into/erased from flash memory.

RDYINT	Flash memory operation state indication bit
0	Data is being written/erased.
1	Data writing/erasing has been completed. (An interrupt request is generated.)

INTE	Bit causing an interrupt to the CPU to be generated
0	Enables an interrupt when data writing/erasing is completed.
1	Disables an interrupt when data writing/erasing is completed.

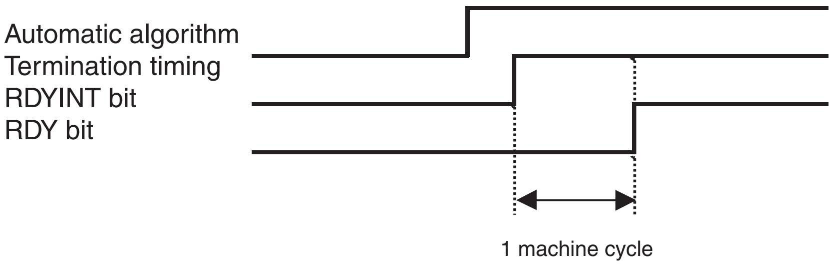
t_{inst} : Instruction cycle
R/W : Read/Write enabled
▨ : Initial value

Table 22.3-1 Description of the Control Status Register (FMCS) bits

Bit		Description
bit7	INTE: INTERRPUT Enable	Bit causing an interrupt to the CPU to be generated when writing into or erasing from flash memory is completed. An interrupt to the CPU is generated when both the INTE bit and RDYINT bit are 1. If the INTE bit is 0, no interrupt is generated.
bit6	RDYINT: READY INTERRUPT	Bit for indicating operation status of flash memory. This bit is set to 1 when writing into or erasing from flash memory is completed. While this bit is set to 0 after writing into or erasing from flash memory has been completed, data cannot be written into or erased from flash memory. After data has been written into or erased from flash memory and this bit has been set to 1, data can be written into or erased from flash memory. Writing 0 clears this bit with 0, while if 1 is written into this bit, it is ignored. This bit is set to 1 upon the termination of the flash memory automatic algorithm (see Section 22.4 "Starting the Flash Memory Automatic Algorithm".). The read modifier write (RMW) command always reads 1 from this bit.
bit5	WE: WRITE ENABLE	Bit for write-enabling flash memory areas. When this bit is set to 1, a write instruction performed after a command sequence for a section from 1000_H to $FFFF_H$ (see Section 22.4 "Starting the Flash Memory Automatic Algorithm") is issued writes data into a flash memory area. When this bit is set to 0, no write/erase signals are generated. This bit is used to start a command for writing data into or erasing data from flash memory. It is recommended that this bit be set to 0 to prevent data from being incorrectly written into flash memory, whenever there is no data to be written or erased.
bit4	RDY: READY	Bit for writing data into or erasing data from flash memory. No data can be written into or erased from flash memory while this bit is set to 0. However, a read command, reset command, and suspend commands such as the sector erase suspend command can be accepted while this bit is set to 0.
bit3 bit2	Reserved bit	Bit reserved for test. Specify 0 for this bit for normal operation.
bit1	Unused bit	Specify 0 to this bit for normal operation.
bit0	Reserved bit	Bit reserved for test. Specify 0 for this bit for normal operation.

Note:

The RDYINT and RDY bits cannot be changed at the same time. Create a program so that decisions are made using one or the other of these bits.



22.4 Starting the Flash Memory Automatic Algorithm

Four types of commands are available for starting the flash memory automatic algorithm: Read/Reset, Write, and Chip Erase. Control of suspend and restart is enabled for sector erase.

■ Command Sequence Table

Table 22.4-1 "Command Sequence Table" lists the commands used for flash memory write/ erase.

Table 22.4-1 Command Sequence Table

Command sequence	Bus write access	1st bus write cycle		2nd bus write cycle		3rd bus write cycle		4th bus read/ write cycle		5th bus write cycle		6th bus write cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Read/Reset (*1)	1	XXXX	F0	-	-	-	-	-	-	-	-	-	-
Read/ Reset (*1)	3	AAAA	AA	5554	55	AAAA	F0	RA	RD	-	-	-	-
Write program	3	AAAA	AA	5554	55	AAAA	A0	PA	PD	-	-	-	-
Chip Erase	6	AAAA	AA	5554	55	AAAA	80	AAAA	AA	5554	55	AAAA	10
Sector Erase	6	AAAA	AA	5554	55	AAAA	80	AAAA	AA	5554	55	SA	30
Sector Erase Suspend		Entering address "XXXXX" and data (B0$_{H}$) suspends erasing during sector erase.											
Sector Erase Restart		Entering address "XXXXX" and data (30$_{H}$) restarts erasing after erasing has been suspended during sector erase.											

Note:

The addresses shown in the table are those on the CPU memory map. All addresses and data are represented in hexadecimal notation. The letter X indicates an appropriate value.
RA: Read address
PA: Write address.
SA: Sector address. See Section 22.2 "Sector Configuration of the Flash Memory".
RD: Read data
PD: Write data.

*1: Both of the two types of Read/Reset commands can reset the flash memory to read mode.

22.5 Confirming the Automatic Algorithm Execution State

Because the write/erase flow of the flash memory is controlled using the automatic algorithm, the flash memory has hardware for posting its internal operating state and completion of operation. This automatic algorithm enables confirmation of the operating state of the built-in flash memory using the following hardware sequences.

■ Hardware Sequence Flags

The hardware sequence flags are configured from the five-bit output of DQ7, DQ6, DQ5, DQ3 and DQ2. The functions of these bits are those of the data polling flag (DQ7), toggle bit flag (DQ6), timing limit exceeded flag (DQ5), sector erase timer flag (DQ3) and toggle bit 2 flag (DQ2). The hardware sequence flags can therefore be used to confirm that writing or chip sector erase has been completed or that erase code write is valid.

The hardware sequence flags can be accessed by read-accessing the addresses of the target sectors in the flash memory after setting of the command sequence (see Table 22.4-1 "Command Sequence Table" in Section 22.4 "Starting the Flash Memory Automatic Algorithm". Table 22.5-1 "Bit Assignments of Hardware Sequence Flags" lists the bit assignments of the hardware sequence flags.

Table 22.5-1 Bit Assignments of Hardware Sequence Flags

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Hardware sequence flag	DQ7	DQ6	DQ5	-	DQ3	DQ2	-	-

To determine whether automatic writing or chip sector erase is being executed, the hardware sequence flags can be checked or the status can be determined from the RDY bit of the flash memory control register (FMCS) that indicates whether writing has been completed. After writing/erasing has terminated, the state returns to the read/reset state. When creating a program, use one of the flags to confirm that automatic writing/erasing has terminated. Then, perform the next processing operation, such as data read. In addition, the hardware sequence flags can be used to confirm whether the second or subsequent sector erase code write is valid. The following sections describe each hardware sequence flag separately. Table 22.5-2 "Hardware Sequence Flag Functions" lists the functions of the hardware sequence flags.

Table 22.5-2 Hardware Sequence Flag Functions

State			DQ7	DQ6	DQ5	DQ3	DQ2
Executing	Automatic writing operation		$\overline{\text{DQ7}}$	Toggle	0	0	1
	Write/erase operation during automatic erasing		0	Toggle	0	1	Toggle
	Erasing	Read (sector in which data is being erased)	1	1	0	0	Toggle
	Suspending	Read (sector in which no data is erased)	DATA	DATA	DATA	DATA	DATA
	Stopping	Write (sector in which no data is erased)	$\overline{\text{DQ7}}$	Toggle	0	0	1(*1)
Exceeding the time limit	Automatic writing operation		$\overline{\text{DQ7}}$	Toggle	1	0	1
	Write/erase operation during automatic erasing		0	Toggle	1	1	(*2)

*1 While an erase suspend write is being performed, DQ2 outputs logical 1 to the address of the sector into which data is being written. However, DQ2 toggles to successive reads from the erase-suspended sector.

*2 When 1 is set to DQ5 (excess of the time limit), DQ2 toggles to successive reads from a sector in which data is being written or erased, and never toggles to any read to other sectors.

22.5.1 Data Polling Flag (DQ7)

The data polling flag uses the data polling function to post that the automatic algorithm is being executed or has terminated

■ Write

Read-access during execution of the automatic write algorithm causes the flash memory to output the opposite data of bit 7 last written, regardless of the value at the address specified by the address signal. Read-access at the end of the automatic write algorithm causes the flash memory to output bit 7 of the read value of the address specified by the address signal.

■ Automatic erasing

Read-access during execution of the automatic erasing algorithm causes the flash memory to output 0, regardless of the value at the address specified by the address signal. After the automatic erasing algorithm is executed, 1 is output.

■ Sector erase suspend

Read-access during a sector erase suspend causes the flash memory to output 1 if the address specified by the address signal belongs to the sector being erased. The flash memory outputs bit 7 (DATA: 7) of the read value at the address specified by the address signal if the address specified by the address signal does not belong to the sector being erased. Referencing this flag together with the toggle bit flag (DQ6) enables a decision to be made on whether the flash memory is in the erase suspended state and which sector is being erased.

Note:

When the automatic algorithm comes to the end of its operation, bit 7 (data polling) changes its state asynchronously during a read operation. This means that flash memory sends data about the operation state to bit 7 and will then send out fixed data. When flash memory ends the automatic algorithm or even if bit 7 is outputting fixed data, the values of the other bits are still undetermined. Fixed data in the other bits can be read by successively executing read operations.

22.5.2 Toggle Bit Flag (DQ6)

Like the data polling flag, the toggle bit flag uses the toggle bit function to post that the automatic algorithm is being executed or has terminated.

■ Automatic write/erase

Making successive read accesses while the automatic writing/erasing algorithm is being performed toggles flash memory and makes it output 1 and then 0, in turn, regardless of the specified address. Making successive read accesses when the automatic writing/erasing algorithm ends makes flash memory to stop bit 6 toggle and outputs the value of bit 6 (DATA:6) corresponding to the value read from the specified address. The toggle bit becomes effective after the last write cycle in each command sequence.

■ Sector erase suspend

Read-access during a sector erase suspend causes the flash memory to output 1 if the address specified by the address signal belongs to the sector being erased. The flash memory outputs bit 6 (DATA: 6) of the read value at the address specified by the address signal if the address specified by the address signal does not belong to the sector being erased.

Note:

For a write, if the sector where data is to be written is rewrite-protected, the toggle bit terminates the toggle operation after approximately 2µs without any data being rewritten. For an erase, if all of the selected sectors are write-protected, the toggle bit performs toggling for approximately 100µs and then returns to the read/reset state without any data being rewritten.

22.5.3 Timing Limit Exceeded Flag (DQ5)

The timing limit exceeded flag is used to post that execution of the automatic algorithm has exceeded the time (internal pulse count) prescribed in the flash memory.

■ Automatic write/erase

Bit 5 indicates that execution of the automatic algorithm exceeded the time (internal pulse count) specified in flash memory. For an excess, bit 5 outputs 1. Thus, if this bit outputs 1 while the automatic algorithm is operating, data writing or data erasing failed.

Bit 5 indicates a failure when an attempt is made to write data into a non-blank area without erasing any data. In the case of such a failure, fixed data cannot be read from bit 7 (data polling) and bit 6 (toggle bit) remains unchanged (toggled). If the time limit is exceeded while there is a failure, 1 is set in bit 5. In this case, note that the setting of bit 5 to 1 does not indicate a flash memory failure but the incorrect use of flash memory. If bit 5 is set to 1 as described above, execute a reset command.

22.5.4 Sector Erase Timer Flag (DQ3)

The sector erase timer flag is used to post whether the automatic algorithm is being executed during the sector erase wait period after the Sector Erase command has been started.

■ Sector erase

After the first sector erase command sequence is executed, the sector erase wait period begins. Bit 3 outputs 0 in the sector erase wait period and, if the period has ended, it outputs 1. Data polling and the toggle bit become effective after the first sector erase command sequence is executed.

If the data polling function or toggle bit function indicates that the erase algorithm is being executed, internally controlled erase has already started if this flag is 1. Continuous write of the sector erase codes or commands other than the Sector Erase Suspend command will be ignored until erase is terminated. (Only erase suspend code input is allowed.)

If this flag is 0, the flash memory will accept write of additional sector erase codes. To confirm this, it is recommended that the state of this flag be checked before continuing to write sector erase codes. If this flag is 1 after the second state check, it is possible that additional sector erase codes may not be accepted.

Read-access during execution of sector erase suspend causes the flash memory to output 1 if the address specified by the address signal belongs to the sector being erased. The flash memory outputs bit 3 of the read value of the address specified by the address signal if the address specified by the address signal does not belong to the sector being erased.

22.5.5 Toggle Bit-2 Flag (DQ2)

The toggle bit-2 flag (DQ2) is used to detect that flash memory is performing an automatic erase operation or erase suspend operation, together with the toggle bit.

■ Sector erase

This toggle bit is used to detect that flash memory is performing an automatic erase operation or erase suspend operation, together with the toggle bit, bit 6. Executing successive reads to a sector in which an automatic erase operation is being performed toggles bit 2. When the flash memory is in erase suspend read mode, executing successive reads to an erase-suspended sector toggles bit 2.

When the flash memory is in erase suspend write mode, successively reading addresses from a sector that is not erase-suspended provides 1 to bit 2. Unlike bit 2, bit 6 toggles during normal write, erase, or erase suspend write.

For example, bit 2 is used together with bit 6 to detect erase suspend read mode (bit 2 toggles but bit 6 does not toggle).

In addition, bit 2 is used to detect sectors in which data is being deleted. While flash memory is erasing data, bit 2 toggles to a read from a sector from which data is being erased.

22.6 Detailed Explanation of Writing to and Erasing Flash Memory

This section describes each operation procedure of flash memory Read/Reset, Write, Chip Erase, Sector Erase, Sector Erase Suspend, and Sector Erase Restart when a command that starts the automatic algorithm is issued.

■ Detailed Explanation of Flash Memory Write/Erase

The flash memory executes the automatic algorithm by issuing a command sequence (see Table 22.4-1 "Command Sequence Table" in Section 22.4 "Starting the Flash Memory Automatic Algorithm") for a write cycle to the bus to perform Read/Reset, Write, Chip Erase, Sector Erase, Sector Erase Suspend, or Sector Erase Restart operations. Each bus write cycle must be performed continuously. In addition, whether the automatic algorithm has terminated can be determined using the data polling or other function. At normal termination, the flash memory is returned to the read/reset state.

Each operation of the flash memory is described in the following order:

22.6.1 "Setting the read/reset state"

22.6.2 "Writing data"

22.6.3 "Erasing all data (erasing chips)"

22.6.4 "Erasing data (erasing sectors)"

22.6.5 "Suspending sector erase"

22.6.6 "Restarting sector erase"

22.6.1 Setting The Read/Reset State

This section describes the procedure for issuing the Read/Reset command to set the flash memory to the read/reset state.

■ Setting the Read/Reset State

The flash memory can be set to the read/reset state by sending the Read/Reset command in the command sequence table (see Table 22.4-1 "Command Sequence Table" in Section 22.4 "Starting the Flash Memory Automatic Algorithm") continuously to the target sector in the flash memory.

The Read/Reset command has two types of command sequences that execute the first and third bus operations. However, there are no essential differences between these command sequences.

The read/reset state is the initial state of the flash memory. When the power is turned on and when a command terminates normally, the flash memory is set to the read/reset state. In the read/reset state, other commands wait for input.

In the read/reset state, data is read by regular read-access. As with the mask ROM, program access from the CPU is enabled. The Read/Reset command is not required to read data by a regular read. The Read/Reset command is mainly used to initialize the automatic algorithm in such cases as when a command does not terminate normally.

22.6.2 Writing Data

This section describes the procedure for issuing the Write command to write data to the flash memory. Figure 22.6-1 "Example of the Flash Memory Write Procedure" shows an example of the flash memory write procedure.

■ Writing Data

The data write automatic algorithm of the flash memory can be started by sending the Write command in the command sequence table (see Table 22.4-1 "Command Sequence Table" in Section 22.4 "Starting the Flash Memory Automatic Algorithm") continuously to the target sector in the flash memory. When data write to the target address is completed in the fourth cycle, the automatic algorithm and automatic write are started.

■ Specifying addresses

Only even addresses can be specified in bytes for the write addresses specified in a write data cycle.

Writing can be done in any order of addresses or even if the sector boundary is exceeded. However, the Write command writes only data of one byte for each execution.

■ Notes on writing data

Writing cannot return data 0 to data 1. When data 1 is written to data 0, the data polling algorithm (DQ7) or toggle operation (DQ6) does not terminate and the flash memory elements are determined to be faulty. If the time prescribed for writing is thus exceeded, the timing limit exceeded flag (DQ5) is determined to be an error. Otherwise, the data is viewed as if dummy data 1 had been written. However, when data is read in the read/reset state, the data remains 0. Data 0 can be set to data 1 only by erase operations.

All commands are ignored during execution of the automatic write algorithm. If a hardware reset is started during writing, the data of the written addresses will be unpredictable.

■ Writing to the Flash Memory

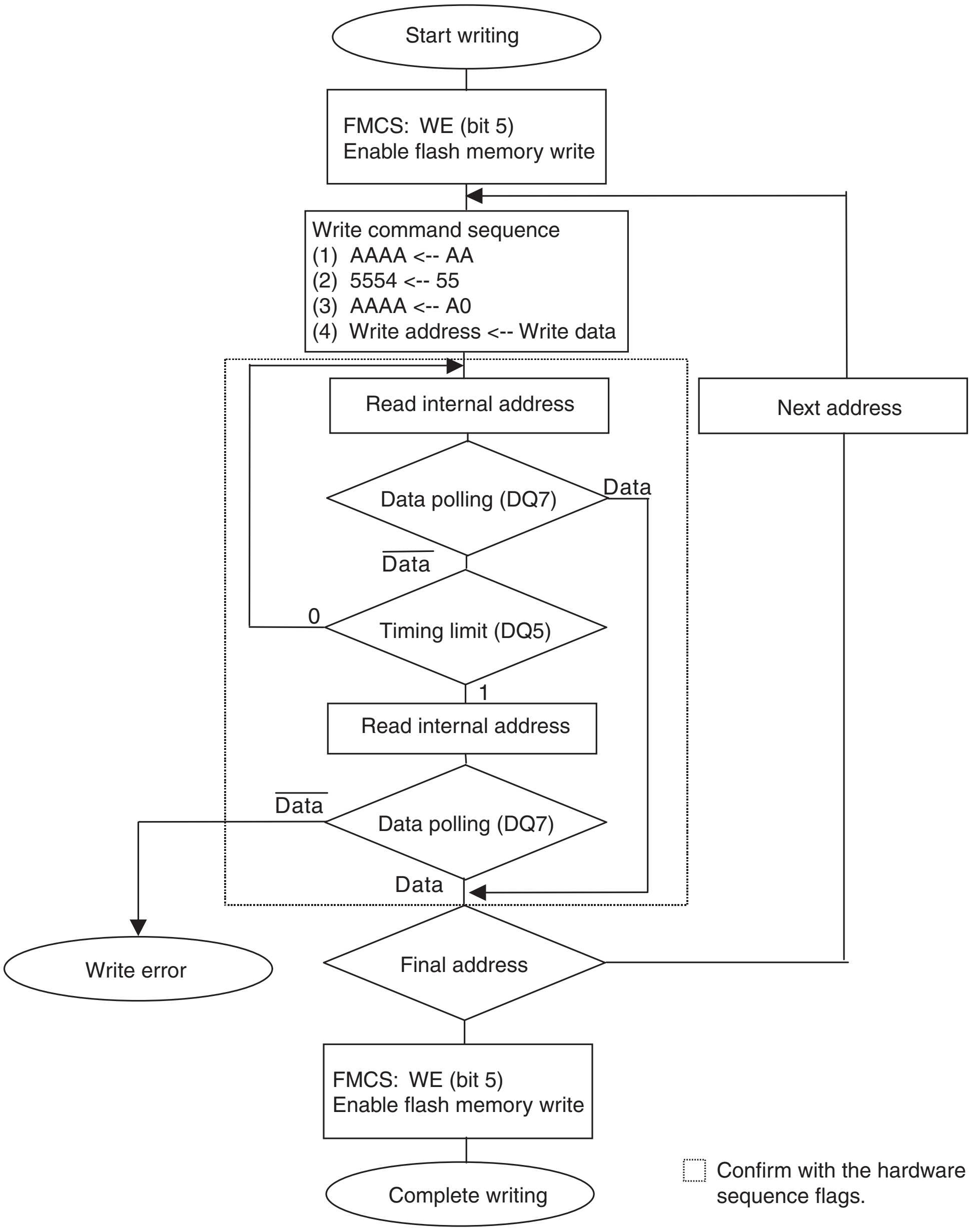
Figure 22.6-1 "Example of the Flash Memory Write Procedure" is an example of the procedure for writing to the flash memory. The hardware sequence flags (see Section 22.5 "Confirming the Automatic Algorithm Execution State") can be used to determine the state of the automatic algorithm in the flash memory. Here, the data polling flag (DQ7) is used to confirm that writing has terminated.

The data read to check the flag is read from the address written to last.

The data polling flag (DQ7) changes at the same time that the timing limit exceeded flag (DQ5) changes. For example, even if the timing limit exceeded flag (DQ5) is 1, the data polling flag bit (DQ7) must be rechecked.

Also for the toggle bit flag (DQ6), the toggle operation stops at the same time that the timing limit exceeded flag bit (DQ5) changes to 1. The toggle bit flag (DQ6) must therefore be rechecked.

Figure 22.6-1 Example of the Flash Memory Write Procedure



22.6.3 Erasing All Data (Erasing Chips)

This section describes the procedure for issuing the Chip Erase command to erase all data in the flash memory.

■ Erasing all Data (Erasing Chips)

All data can be erased from the flash memory by sending the Chip Erase command in the command sequence table (see Table 22.4-1 "Command Sequence Table" in Section 22.4 "Starting the Flash Memory Automatic Algorithm") continuously to the target sector in the flash memory.

The Chip Erase command is executed in six bus operations. When writing of the sixth cycle is completed, the chip erase operation is started. For chip erase, the user need not write to the flash memory before erasing. During execution of the automatic erase algorithm, the flash memory writes 0 for verification before all of the cells are erased automatically.

22.6.4 Erasing Data (Erasing Sectors)

This section describes the procedure for issuing the Sector Erase command to erase optional data (erase sector) in the flash memory. Individual sectors can be erased. Multiple sectors can also be specified at one time.

■ Erasing Data (Erasing Sectors)

Optional sectors in the flash memory can be erased by sending the Sector Erase command in the command sequence table (see Table 22.4-1 "Command Sequence Table" in Section 22.4 "Starting the Flash Memory Automatic Algorithm") continuously to the target sector in the flash memory.

■ Specifying sectors

The Sector Erase command is executed in six bus operations. Sector erase wait of 50µs is started by writing the sector erase code (30h) to an accessible even-numbered address in the target sector in the sixth cycle. To erase multiple sectors, write the erase code (30h) to the addresses in the target sectors after the above processing operation.

■ Notes on specifying multiple sectors

Erase is started when the sector erase wait period of 50µs terminates after the final sector erase code has been written. That is, to erase multiple sectors at one time, an erase code (sixth cycle of the command sequence) must be written within 50µs of writing of the address of a sector and the address of the next sector must be written within 50µs of writing of the previous erase code. Otherwise, the address and erase code may not be accepted. The sector erase timer (hardware sequence flag DQ3) can be used to check whether writing of the subsequent sector erase code is valid. At this time, specify so that the address used for reading the sector erase timer indicates the sector to be erased.

■ Erasing Sectors

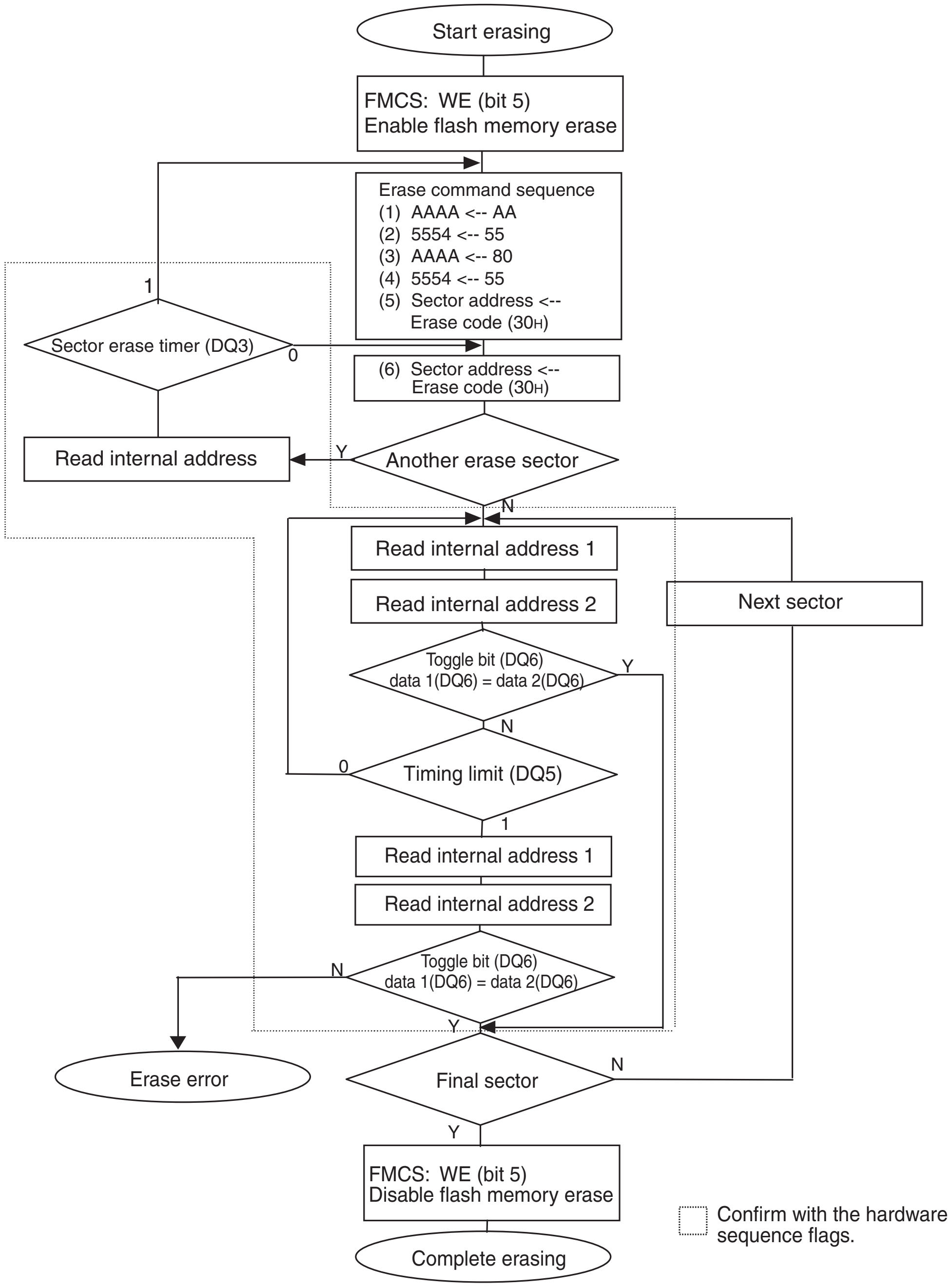
The hardware sequence flags (see Section 22.5 "Confirming the Automatic Algorithm Execution State") can be used to determine the state of the automatic algorithm in the flash memory. Figure 22.6-2 "Example of the Sector Erase Procedure" is an example of the procedure for erasing sectors in the flash memory. Here, the toggle bit flag (DQ6) is used to confirm that erasing has terminated.

The data that is read to check the flag is read from the sector to be erased.

The toggle bit flag (DQ6) stops the toggle operation at the same time that the timing limit exceeded flag (DQ5) is changed to 1. For example, even if the timing limit exceeded flag (DQ5) is 1, the toggle bit flag (DQ6) must be rechecked.

The data polling flag (DQ7) also changes at the same time that the timing limit exceeded flag bit (DQ5) changes. As a result, the data polling flag (DQ7) must be rechecked.

Figure 22.6-2 Example of the Sector Erase Procedure



22.6.5 Suspending Sector Erase

This section describes the procedure for issuing the Sector Erase Suspend command to suspend erasing of flash memory sectors. Data can be read from sectors that are not being erased.

■ Suspending Erasing of Sectors

Erasing of flash memory sectors can be suspended by sending the Sector Erase Suspend command in the command sequence table (see Table 22.4-1 "Command Sequence Table" in Section 22.4 "Starting the Flash Memory Automatic Algorithm") continuously to the target sector in the flash memory.

The Sector Erase Suspend command suspends the sector erase operation being executed and enables data to be read from sectors that are not being erased. In this state, only reading is enabled; data cannot be written. This command is valid only during sector erase operations that include the erase wait time. The command will be ignored during chip erase or write operations.

This command is implemented by writing the erase suspend code (B0h). At this time, specify an optional address in the flash memory for the address. An Erase Suspend command issued again during erasing of sectors will be ignored.

Entering the Sector Erase Suspend command during the sector erase wait period will immediately terminate sector erase wait, cancel the erase operation, and set the erase stop state. Entering the Erase Suspend command during the erase operation after the sector erase wait period has terminated will set the erase suspend state after a maximum period of 15µs has elapsed.

22.6.6 Restarting Sector Erase

This section describes the procedure for issuing the Sector Erase Restart command to restart suspended erasing of flash memory sectors.

■ Restarting Erasing of Sectors

Suspended erasing of flash memory sectors can be restarted by sending the Sector Erase Restart command in the command sequence table (see Table 22.4-1 "Command Sequence Table" in Section 22.4 "Starting the Flash Memory Automatic Algorithm") continuously to the target sector in the flash memory.

The Sector Erase Restart command is used to restart erasing of sectors from the sector erase suspend state set using the Sector Erase Suspend command. The Sector Erase Restart command is implemented by writing the erase restart code (30h). At this time, specify an optional address in the flash memory area for the address.

If a Sector Erase Restart command is issued during sector erase, the command will be ignored.

22.7 Notes on using Flash Memory

This section provides notes on using the MB89F538/F538L, especially for flash memory.

■ Input of a hardware reset ($\overline{\text{RST}}$)

To input a hardware reset when reading is in progress, i.e., when the automatic algorithm has not been started, secure a minimum low-level width of 500 ns.

To input a hardware reset while a write or erase is in progress, i.e., while the automatic algorithm is being started, secure a minimum low-level width of 500 ns. In this case, 20 µs are required until the data becomes readable after the operation being performed terminates and the flash memory is fully initialized.

Performing a hardware reset during a write operation makes the data being written undetermined. Also note that performing a hardware reset during an erase operation may make the sector from which data is being erased unusable.

■ Software reset, watchdog timer reset

When write/erase of flash memory is set up for normal mode and CPU memory access mode is internal ROM mode, and if a reset cause occurs while the automatic algorithm of flash memory is being activated, the CPU may run out of control.

The cause of a reset does not initialize the flash memory and keeps the automatic algorithm operating. Thus, when the CPU starts a sequence after the reset is canceled, the flash memory may not have been in a read state. Prevent a cause of a reset from occurring while the flash memory is writing or erasing.

■ Program access to flash memory

While the automatic algorithm is being activated, any read access to the flash memory is disabled. When CPU memory access mode is set to internal ROM mode, move program areas into another area such as RAM, and then start a write or erase.

In this case, when sectors containing interrupt vectors are erased, the writing or erasing of interrupt processing cannot be executed.

For the same reason, other interrupt processing shall be disabled while the automatic algorithm is being activated.

CHAPTER 23 MB89F538/F538L SERIAL PROGRAMMING

This chapter describes the example of the serial programming when the flash MCU programmer by Yokogawa Digital Computer Corporation.

23.1 Basic Configuration of MB89F538/F538L Serial Programming Connection

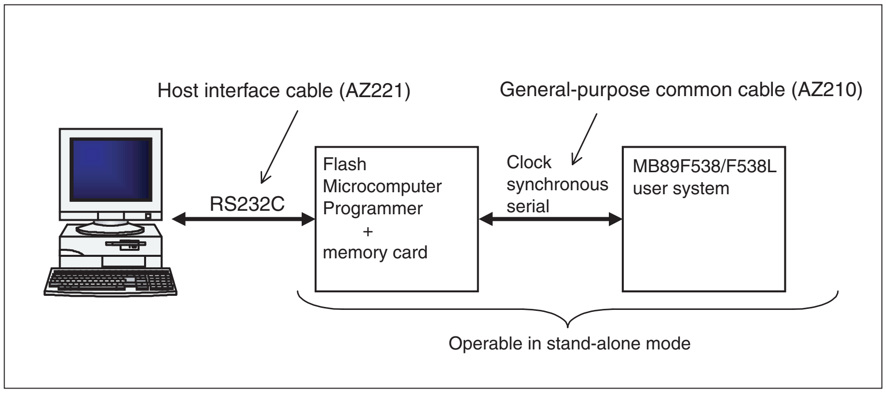
In MB89F538/F538L, the serial on-board programming of flash ROM (Fujitsu standard) is supported. The specification is explained as follows.

■ Basic Configuration of MB89F538/F538L Serial Programming Connection

Based on Fujitsu standard serial on-board programming, the flash MCU programmer by Yokogawa Digital Computer Corporation can be used.

Figure 23.1-1 "Basic Configuration of MB89F538/F538L Serial Programming Connection" shows the basic configuration of the MB89F538/F538L serial programming connection.

Figure 23.1-1 Basic Configuration of MB89F538/F538L Serial Programming Connection


Host interface cable (AZ221)
General-purpose common cable (AZ210)
RS232C
Flash Microcomputer Programmer + memory card
Clock synchronous serial
MB89F538/F538L user system
Operable in stand-alone mode

Note:

Please inquire to Yokogawa Digital Computer Corporation for more information about the operation method of flash MCU programmer (AF220/AF210/AF120/AF110), general-purpose common cable for connection (AZ210) and the connector.

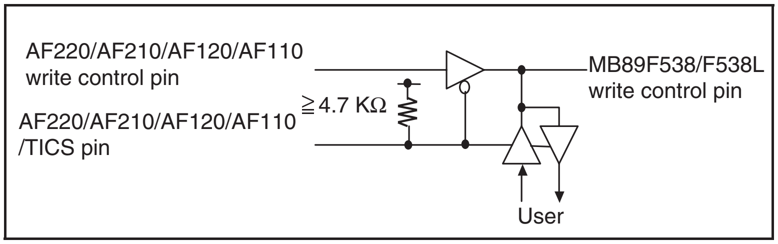
❍ Pins for Fujitsu standard serial on-board programming

Table 23.1-1 Pins for Fujitsu Standard Serial On-board Programming

Pin	Function	Note
MOD2, MOD1 MOD0, P22, P23, P10	Mode pins	Set MOD2 = 0, MOD1 = 1, MOD0 = 1, P22 = 1, P23 = 0, P10 = 1 to change to serial programming mode.
X0, X1	Oscillation pins	The internal operation of CPU clock at the serial programming mode is divided by four of the oscillation frequency. Moreover , it is necessary to input the oscillation frequency with 3 MHz or more at the serial programming.
$\overline{\text{RST}}$	Reset pin	-
SI2	Serial data input pin	Use 8-bit serial I/O.
SO2	Serial data output pin	
SCK2	Serial clock input pin	
V_{CC}	Power voltage supply pin	When the user system supplies the programming voltage, the connection with the flash MCU programmer is unnecessary.
V_{SS}	GND pin	It is common with flash MCU programmer's GND.

Moreover, if SI, SO, SCK pins are also used by the user system, the control circuit is necessary (see Figure 23.1-2 "Control Circuit"). (During the serial programming, the user circuit can be cut off by the flash MCU programmer's /TICS signal. See the connection example.)

Figure 23.1-2 Control Circuit



❍ About oscillation clock frequency and serial input clock frequency

For MB89F538/F538L, the serial clock frequency that is able to input is calculate from the following formula. Therefore, please change the serial clock input frequency in the flash MCU programmer's setting by the using oscillation clock frequency.

Serial clock frequency (Max.) = 0.125 x oscillation clock frequency

Example:

Oscillation clock frequency	Max. Serial Clock Frequency provided by MCU	Max. Serial Clock Frequency provided by AF220/AF210/AF120/AF110	Max. Clock Serial Frequency provided by AF200
4 MHz	500 kHz	500 kHz	500 kHz
8 MHz	1 MHz	850 kHz	500 kHz
10 MHz	1.25 MHz	1.25 MHz	500 kHz

❍ Configuration of flash microcomputer programmer system (made by Yokogawa Digital Computer Corporation)

Table 23.1-2 Configuration of Flash Microcomputer Programmer System (Made by Yokogawa Digital Computer Corporation)

Model		Function
Mainframe	AF220/AC4P	Model with Ethernet interface built in / 100 to 220 V power adaptor
	AF210/AC4P	Standard model / 100V to 220 V power adaptor
	AF120/AC4P	Single-key Ethernet interface model / 100V to 220 V power adaptor
	AF110/AC4P	Single-key model/100V to 220 V power adaptor
AZ221		PC/AT RS232C cable only for Programmer
AZ210		Standard target probe (a), length: 1 m
FF201		Fujitsu F^{2}MC-16LX flash microcomputer control model
AZ290		Remote controller
/P2		2 MB PC card (option) for flash memory sizes of up to 128 KB
/P4		4 MB PC card (option) for flash memory sizes of up to 512 KB

Inquiries: Yokogawa Digital Computer Corporation

Telephone number: (81)-42-333-6224

Note:

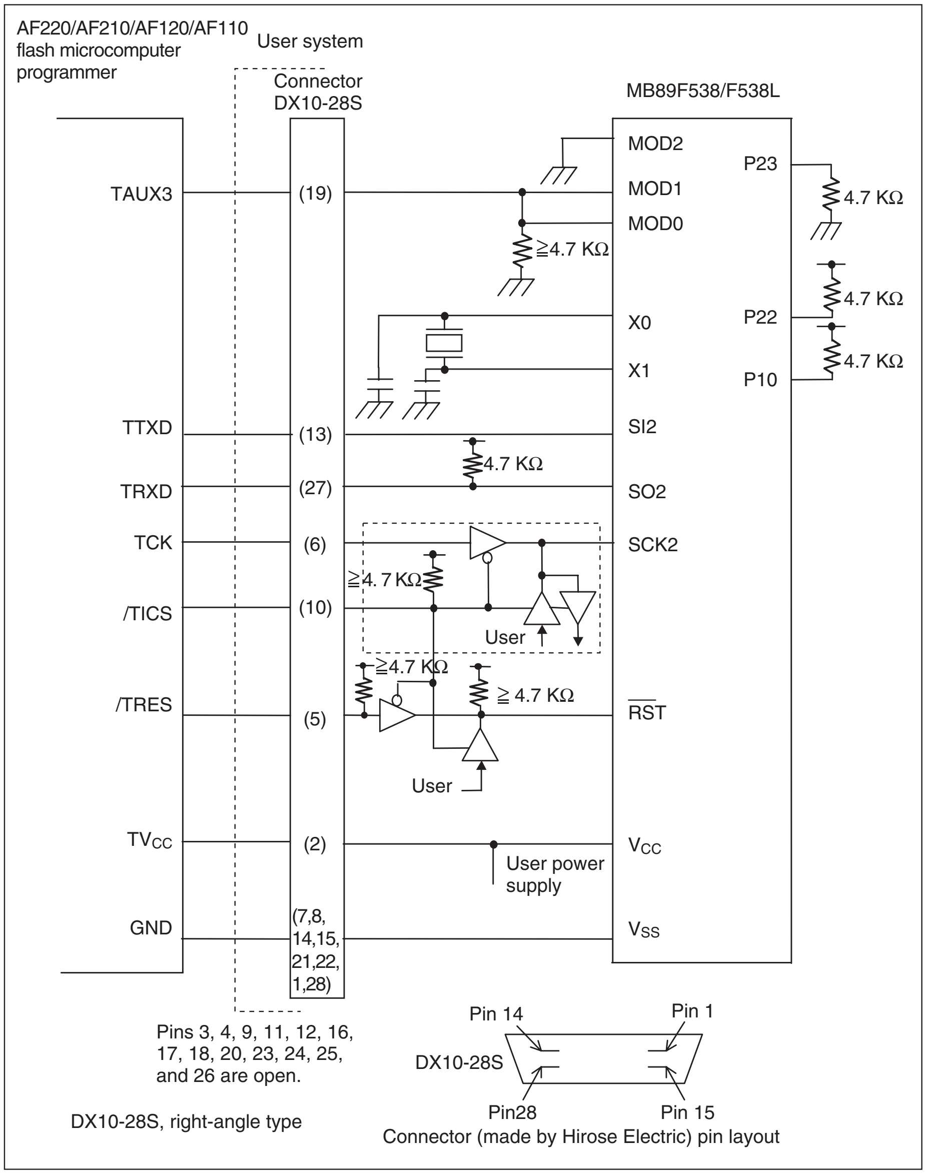
Though AF200 flash microcomputer programmer is not available, it is possible to correspond by using control module FF201. Example of serial programming connection is possible to correspond by the connection example that is shown in the next page.

23.2 Connection Example of Serial Programming (when User Power Supply is Used)

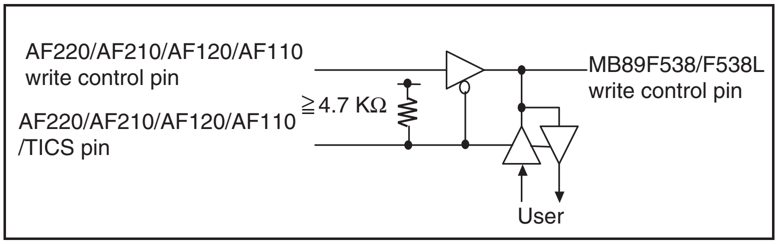
In the user system, to change to serial programming mode, input MOD1, MOD0 = 11 to the mode pin that is set to MOD2, MOD1, MOD0 = 000 from TAUX3 of AF220/AF210/AF120/AF110. (Serial programming mode: MOD2, MOD1, MOD0 = 011)

■ Example of Connection for Serial Programming (when Power Supplied by User)

Figure 23.2-1 Example of Connection for Serial Programming in MB89F538/538L (when Power Supplied by User)



- Moreover, if SI2, SO2 pins are also used by the user system, SCK2 pin and the following control circuit is necessary. (During the serial programming, the user circuit can be cut off by the flash MCU programmer's /TICS signal.)



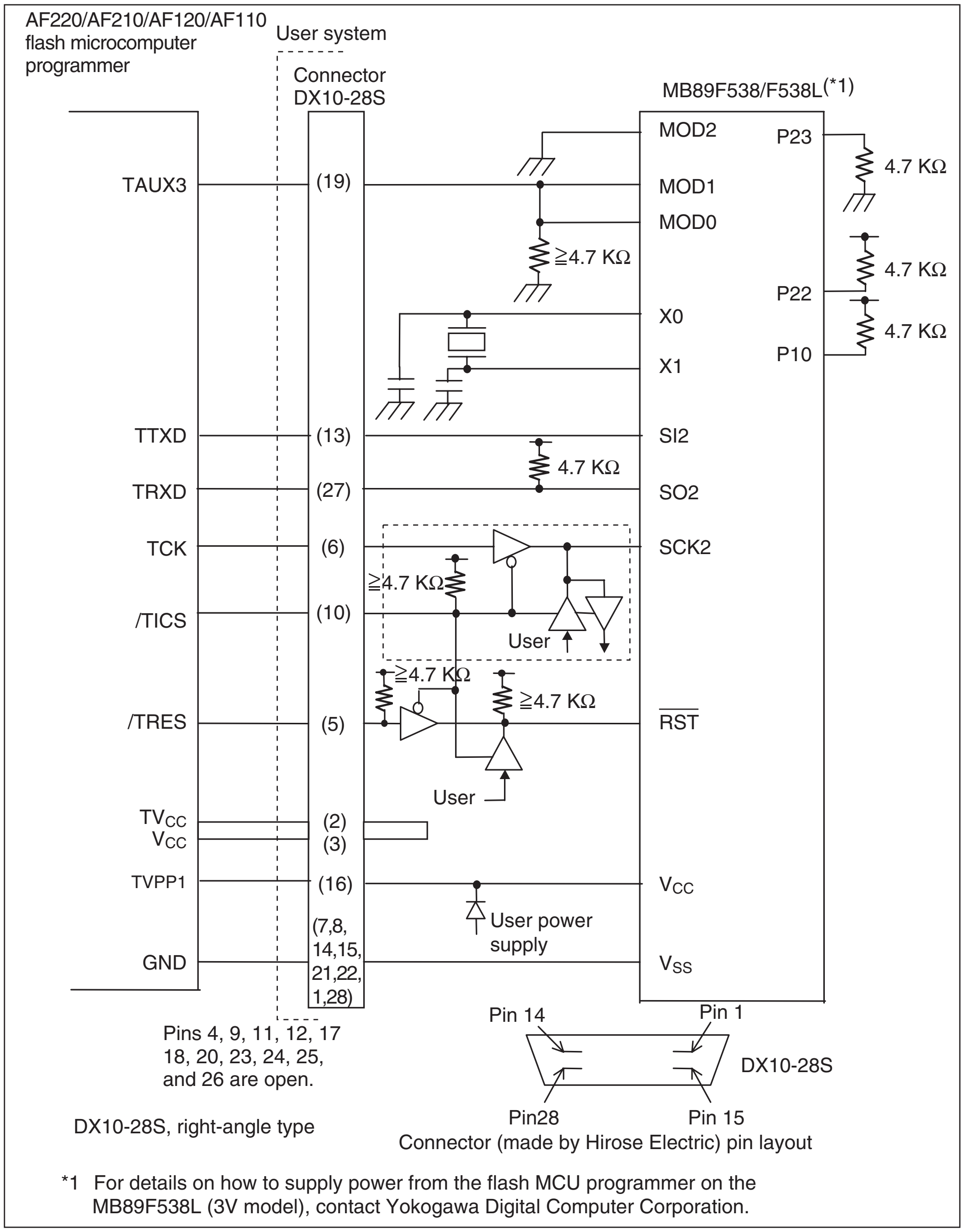
- Connect AF220/AF210/AF120/AF110 during user power supply is "OFF".
- In Figure 23.2-1 "Example of Connection for Serial Programming in MB89F538/F538L (when Power Supplied by User)," the values indicated for the pull-up and pull-down resistors are samples, which can be changed depending on your system requirements. When noise may cause the input levels at the MOD0 or MOD1 pin to fluctuate, it is recommended that anti-noise measures be taken by connecting capacitors or other devices.

23.3 Connection Example of Serial Programming (when Power Supply is Supplied from Flash MCU Programmer)

In the user system, to change to serial programming mode, input MOD1, MOD0 = 11 to the mode pin that is set to MOD2, MOD1, MOD0 = 000 from TAUX3 of AF220/AF210/AF120/AF110. (Serial programming mode: MOD2, MOD1, MOD0 = 011).

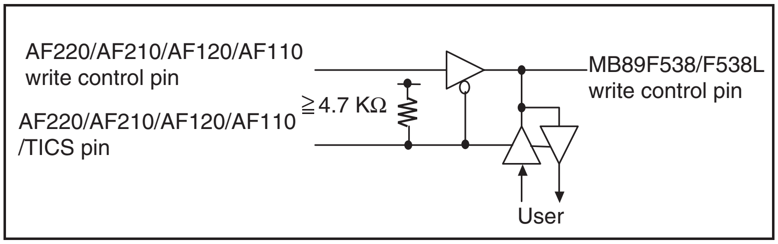
■ Example of Connection for Serial Programming (when Power Supplied by Flash MCU Programmer)

Figure 23.3-1 Example of Connection for Serial Programming in MB89F538/538L (when Power Supplied by Flash MCU Programmer)



*1 For details on how to supply power from the flash MCU programmer on the MB89F538L (3V model), contact Yokogawa Digital Computer Corporation.

- Moreover, if SI2, SO2 pins are also used by the user system, SCK2 pin and the following control circuit is necessary. (During the serial programming, the user circuit can be cut off by the flash MCU programmer's /TICS signal.)



- Connect AF220/AF210/AF120/AF110 during user power supply is "OFF".

  No short-circuit with the user power supply is allowed when AF220/AF210/AF120/AF110 supplies the programming power supply.

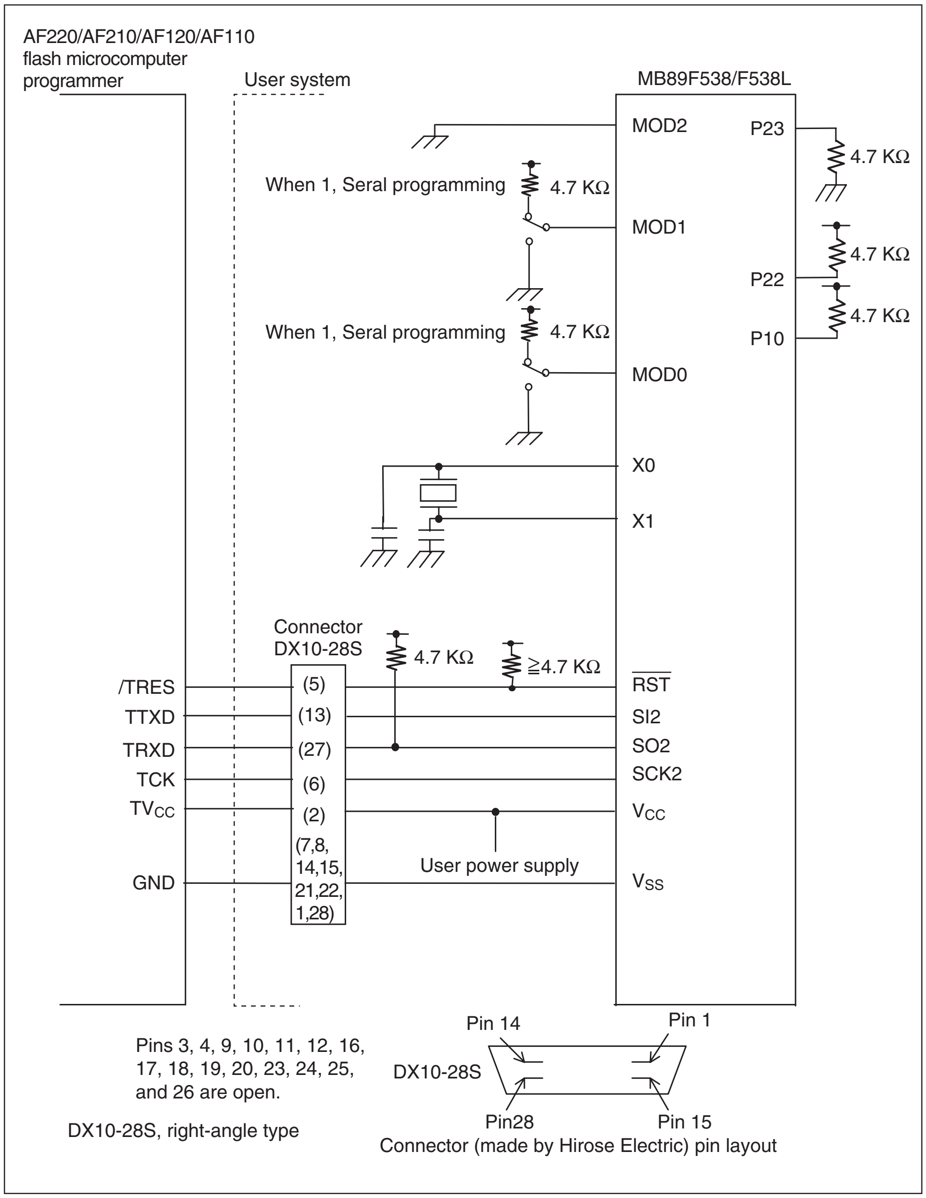
- In Figure 23.3-1 "Example of Connection for Serial Programming in MB89F538/F538L (when Power Supplied by Flash MCU Programmer)," the values indicated for the pull-up and pull-down resistors are samples, which can be changed depending on your system requirements. When noise may cause the input levels at the MOD0 or MOD1 pin to fluctuate, it is recommended that anti-noise measures be taken by connecting capacitors or other devices.

23.4 Minimum Connection Example with Flash MCU Programmer (when User Power Supply is Used)

If each pin is set as shown in Figure 23.4-1 "Example of Minimum Connection for Flash MCU Programmer in MB89F538/538L (when Power Supplied by User)", MOD1 and MOD0 do not need to be connected with the flash MCU programmer.

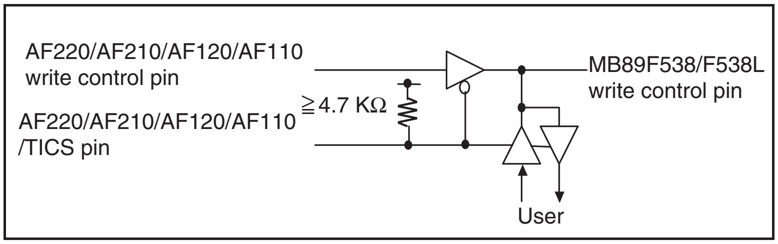
■ Example of Minimum Connection for Flash MCU Programmer (when Power Supplied by User)

Figure 23.4-1 Example of Minimum Connection for Flash MCU Programmer in MB89F538/538L (when Power Supplied by User)



AF220/AF210/AF120/AF110 flash microcomputer programmer
User system
MB89F538/F538L
MOD2
P23
4.7 KΩ
When 1, Seral programming
4.7 KΩ
MOD1
4.7 KΩ
P22
4.7 KΩ
When 1, Seral programming
4.7 KΩ
P10
MOD0
X0
X1
Connector DX10-28S
4.7 KΩ
≧4.7 KΩ
/TRES
TTXD
TRXD
TCK
TVCC
GND
(5)
(13)
(27)
(6)
(2)
(7,8, 14,15, 21,22, 1,28)
RST
SI2
SO2
SCK2
VCC
User power supply
VSS
Pins 3, 4, 9, 10, 11, 12, 16, 17, 18, 19, 20, 23, 24, 25, and 26 are open.
DX10-28S, right-angle type
Pin 14
Pin 1
DX10-28S
Pin28
Pin 15
Connector (made by Hirose Electric) pin layout

- Moreover, if SI2, SO2, SCK2 pins are also used by the user system, the following control circuit is necessary. (During the serial programming, the user circuit can be cut off by the flash MCU programmer's /TICS signal.)



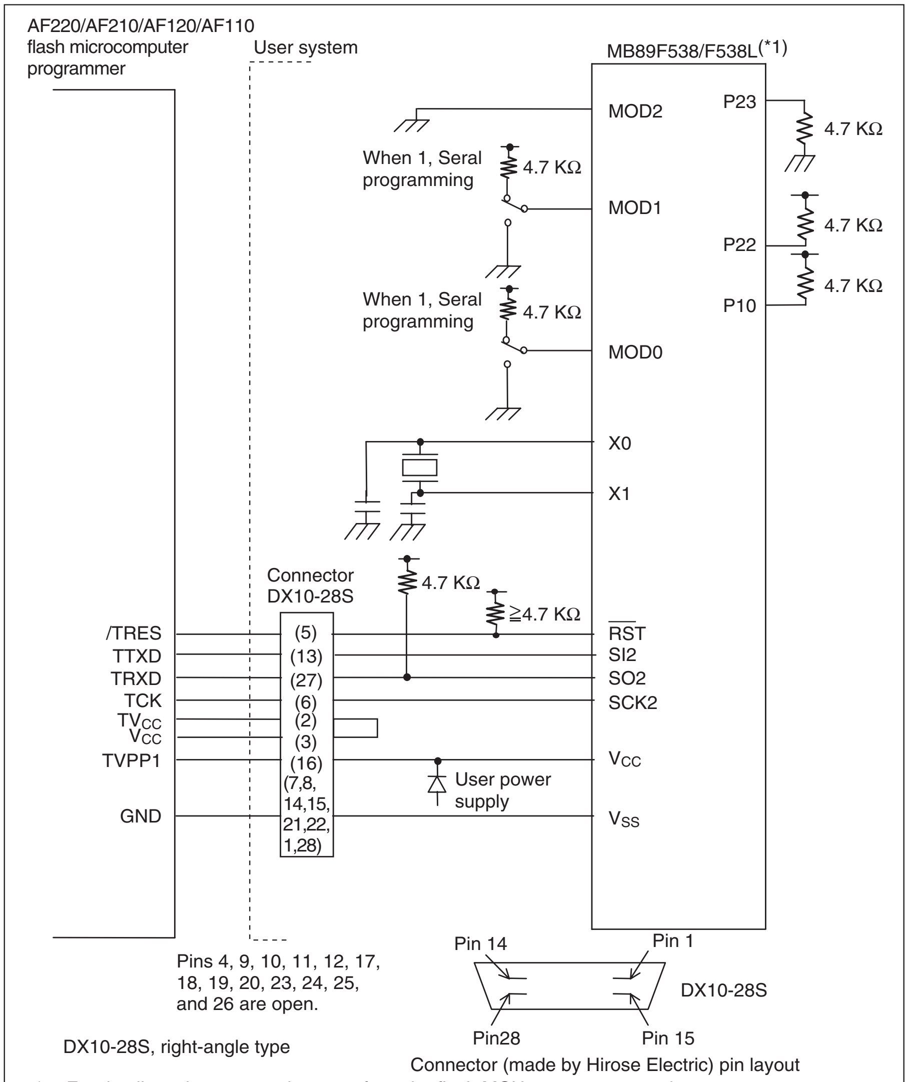
- Connect AF220/AF210/AF120/AF110 during user power supply is "OFF".
- In Figure 23.4-1 "Example of Minimum Connection for Flash MCU Programmer in MB89F538/F538L (when Power Supplied by User)," the values indicated for the pull-up and pull-down resistors are samples, which can be changed depending on your system requirements. When noise may cause the input levels at the MOD0 or MOD1 pin to fluctuate, it is recommended that anti-noise measures be taken by connecting capacitors or other devices.

23.5 Minimum Connection Example with Flash Microcomputer Programmer (when Power Supply is Supplied from Flash MCU Programmer)

If each pin is set as shown in Figure 23.5-1 "Example of Minimum Connection for Flash MCU Programmer in MB89F538/538L (when Power Supplied by Flash MCU Programmer)" during the serial programming, the connection with MOD1 and MOD2 and flash MCU programmer is unnecessary.

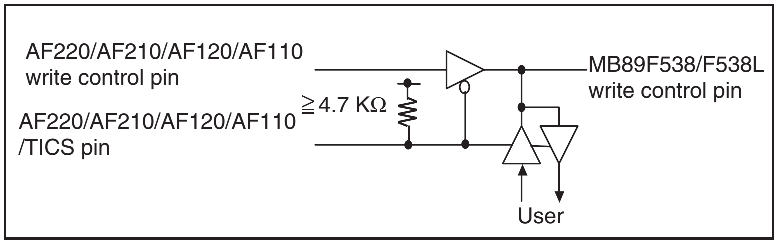
■ Minimum Connection Example with Flash Microcomputer Programmer (when Power Supply is Supplied from Flash MCU Programmer)

Figure 23.5-1 Example of Minimum Connection for Flash MCU Programmer in MB89F538/538L (when Power Supplied by Flash MCU Programmer)



*1 For details on how to supply power from the flash MCU programmer on the MB89F538L (3V model), contact Yokogawa Digital Computer Corporation.

- Moreover, if SI2, SO2, SCK2 pins are also used by the user system, the following control circuit is necessary. (During the serial programming, the user circuit can be cut off by the flash MCU programmer's /TICS signal.)



- Connect AF220/AF210/AF120/AF110 during user power supply is "OFF".

  No short-circuit with the user power supply is allowed when AF220/AF210/AF120/AF110 supplies the programming power supply.

- In Figure 23.5-1 "Example of Minimum Connection for Flash MCU Programmer in MB89F538/F538L (when Power Supplied by Flash MCU Programmer)," the values indicated for the pull-up and pull-down resistors are samples, which can be changed depending on your system requirements. When noise may cause the input levels at the MOD0 or MOD1 pin to fluctuate, it is recommended that anti-noise measures be taken by connecting capacitors or other devices.

APPENDIX

This appendix lists the I/O map and instructions.

APPENDIX A I/O Maps

The registers of the peripheral functions contained in MB89530/530H/530A are assigned addresses as listed in Table Table A-1 "I/O Map."

■ I/O Maps

Table A-1 I/O Map

Address	Register abbreviation	Register name	Read/Write	Initial value
00$_H$	PDR0	Port 0 data register	R/W	XXXXXXXX$_B$
01$_H$	DDR0	Port 0 direction register	W	00000000$_B$
02$_H$	PDR1	Port 1 data register	R/W	XXXXXXXX$_B$
03$_H$	DDR1	Port 1 direction register	W	00000000$_B$
04$_H$ to 06$_H$	(Vacancy)			
07$_H$	SYCC	System clock area control register	R/W	X-1MM100$_B$
08$_H$	STBC	Standby control register	R/W	00010---$_B$
09$_H$	WDTC	Watchdog control register	R/W	0---XXXX$_B$
0A$_H$	TBTC	Timebase timer control register	R/W	00---000$_B$
0B$_H$	WPCR	Watch prescaler control register	R/W	00--0000$_B$
0C$_H$	PDR2	Port 2 data register	R/W	XXXXXXXX$_B$
0D$_H$	DDR2	Port 2 direction register	R/W	00000000$_B$
0E$_H$	PDR3	Port 3 data register	R/W	XXXXXXXX$_B$
0F$_H$	DDR3	Port 3 direction data register	R/W	00000000$_B$
10$_H$	PDR4	Port 4 data register	R/W	XXXX11XX$_B$
11$_H$	DDR4	Port 4 direction register	R/W	0000--00$_B$
12$_H$	PDR5	Port 5 data register	R/W	11111111$_B$
13$_H$	PDR6	Port 6 data register	R	XXXXXXXX$_B$
14$_H$ to 21$_H$	(Vacancy)			
22$_H$	SMC11	Serial mode control register 1 (UART)	R/W	00000000$_B$
23$_H$	SRC1	Serial rate control register (UART)	R/W	--011000$_B$
24$_H$	SSD1	Serial status at data register (UART)	R/W	00100-1X$_B$
25$_H$	SIDR1/SODR1	Serial input/output data register (UART)	R/W	XXXXXXXX$_B$
26$_H$	SMC12	Serial mode control register 2 (UART)	R/W	--100001$_B$
27$_H$	CNTR1	PWM control register 1	R/W	00000000$_B$

Table A-1 I/O Map (Continued)

Address	Register abbreviation	Register name	Read/Write	Initial value
28$_H$	CNTR2	PWM control register 2	R/W	000-0000$_B$
29$_H$	CNTR3	PWM control register 3	R/W	-000----$_B$
2A$_H$	COMR1	PWM compare register 1	W	XXXXXXXX$_B$
2B$_H$	COMR2	PWM compare register 2	W	XXXXXXXX$_B$
2C$_H$	PCR1	PWC pulse width control register 1	R/W	000--000$_B$
2D$_H$	PCR2	PWC pulse width control register 2	R/W	00000000$_B$
2E$_H$	RLBR	PWC reload buffer register	R/W	XXXXXXXX$_B$
2F$_H$	SMC21	Serial mode control register 1 (UART/SIO)	R/W	00000000$_B$
30$_H$	SMC22	Serial mode control register 2 (UART/SIO)	R/W	00000000$_B$
31$_H$	SSD2	Serial status and data register (UART/SIO)	R/W	00001---$_B$
32$_H$	SIDR2/SODR2	Serial data register (UART/SIO)	R/W	XXXXXXXX$_B$
33$_H$	SRC2	Baud rate generator load register	R/W	XXXXXXXX$_B$
34$_H$	ADC1	A/D control register 1	R/W	X00000X0$_B$
35$_H$	ADC2	A/D control register 2	R/W	-0000001$_B$
36$_H$	ADDL	Lower A/D data register	R/W	XXXXXXXX$_B$
37$_H$	ADDH	Upper A/D data register	R/W	------00$_B$
38$_H$	PPGC2	PPG2 control register (12-bit PPG)	R/W	00000000$_B$
39$_H$	PRL22	PPG2 reload register 2 (12-bit PPG)	R/W	0X000000$_B$
3A$_H$	PRL21	PPG2 reload register 1 (12-bit PPG)	R/W	XX000000$_B$
3B$_H$	PRL23	PPG2 reload register 3 (12-bit PPG)	R/W	XX000000$_B$
3C$_H$	TMCR	16-bit timer control register	R/W	--000000$_B$
3D$_H$	TCHR	Upper 16-bit timer count register	R/W	00000000$_B$
3E$_H$	TCLR	Lower 16-bit timer count register	R/W	00000000$_B$
3F$_H$	EIC1	External interrupt 1 control register 1	R/W	00000000$_B$
40$_H$	EIC2	External interrupt 1 control register 2	R/W	00000000$_B$
41$_H$ to 48$_H$	(Vacancy)			
49$_H$	DDCR	DDC selection register	R/W	-------0$_B$
4A$_H$ to 4B$_H$	(Vacancy)			
4C$_H$	PPGC1	PPG1 control register (12-bit PPG)	R/W	00000000$_B$
4D$_H$	PRL12	PPG1 reload register 2 (12-bit PPG)	R/W	0X000000$_B$
4E$_H$	PRL11	PPG1 reload register 1 (12-bit PPG)	R/W	XX000000$_B$
4F$_H$	PRL13	PPG1 reload register 3 (12-bit PPG)	R/W	XX000000$_B$
50$_H$	IACR	I^{2}C address control register	R/W	-----000$_B$

Table A-1 I/O Map (Continued)

Address	Register abbreviation	Register name	Read/Write	Initial value
51$_H$	IBSR	I^2C bus status register	R	00000000$_B$
52$_H$	IBCR	I^2C bus control register	R/W	00000000$_B$
53$_H$	ICCR	I^2C clock control register	R/W	000XXXXX$_B$
54$_H$	IADR	I^2C address register	R/W	-XXXXXXX$_B$
55$_H$	IDAR	I^2C data register	R/W	XXXXXXXX$_B$
56$_H$	EIE2	External interrupt 2 control register	R/W	00000000$_B$
57$_H$	EIF2	External interrupt 2 flag register	R/W	-------0$_B$
58$_H$	RCR1	6-bit PPG control register 1	R/W	00000000$_B$
59$_H$	RCR2	6-bit PPG control register 2	R/W	0X000000$_B$
5A$_H$	CKR	Clock output control register	R/W	------00$_B$
5B$_H$ to 6F$_H$	(Vacancy)			
70$_H$	SMR	Serial mode register (SIO)	R/W	00000000$_B$
71$_H$	SDR	Serial data register (SIO)	R/W	XXXXXXXX$_B$
72$_H$	PURR0	Port 0 pull-up resistor register	R/W	11111111$_B$
73$_H$	PURR1	Port 1 pull-up resistor register	R/W	11111111$_B$
74$_H$	PURR2	Port 2 pull-up resistor register	R/W	11111111$_B$
75$_H$	PURR3	Port 3 pull-up resistor register	R/W	11111111$_B$
76$_H$	PURR4	Port 4 pull-up resistor register	R/W	1111--11$_B$
77$_H$	WREN	Wild register enable register	R/W	--000000$_B$
78$_H$	WROR	Wild register data test register	R/W	--000000$_B$
79$_H$	PURR6	Port 6 pull-up resistor register	R/W	---11111$_B$
7A$_H$	(Vacancy)			
7B$_H$	ILR1	Interrupt level set register 1	W	11111111$_B$
7C$_H$	ILR2	Interrupt level set register 2	W	11111111$_B$
7D$_H$	ILR3	Interrupt level set register 3	W	11111111$_B$
7E$_H$	ILR4	Interrupt level set register 4	W	11111111$_B$
7F$_H$	ITR	Interrupt test register	Access prohibited	XXXXX00$_B$

Table A-2 Extended I/O area

Address	Register abbreviation	Register name	Read/Write	Initial value
$C80_H$	WRARH1	Upper address set register 1	R/W	$XXXXXXXX_B$
$C81_H$	WRARL1	Lower address set register 1	R/W	$XXXXXXXX_B$
$C82_H$	WRDR1	Data set register 1	R/W	$XXXXXXXX_B$
$C83_H$	WRARH2	Upper address set register 2	R/W	$XXXXXXXX_B$
$C84_H$	WRARL2	Lower address set register 2	R/W	$XXXXXXXX_B$
$C85_H$	WRDR2	Data set register 2	R/W	$XXXXXXXX_B$
$C86_H$	WRARH3	Upper address set register 3	R/W	$XXXXXXXX_B$
$C87_H$	WRARL3	Lower address set register 3	R/W	$XXXXXXXX_B$
$C88_H$	WRDR3	Data set register 3	R/W	$XXXXXXXX_B$
$C89_H$	WRARH4	Upper address set register 4	R/W	$XXXXXXXX_B$
$C8A_H$	WRARL4	Lower address set register 4	R/W	$XXXXXXXX_B$
$C8B_H$	WRDR4	Data set register 4	R/W	$XXXXXXXX_B$
$C8C_H$	WRARH5	Upper address set register 5	R/W	$XXXXXXXX_B$
$C8D_H$	WRARL5	Lower address set register 5	R/W	$XXXXXXXX_B$
$C8E_H$	WRDR5	Data set register 5	R/W	$XXXXXXXX_B$
$C8F_H$	WRARH6	Upper address set register 6	R/W	$XXXXXXXX_B$
$C90_H$	WRARL6	Lower address set register 6	R/W	$XXXXXXXX_B$
$C91_H$	WRDR6	Data set register 6	R/W	$XXXXXXXX_B$

❍ **Explanation of read/write**

- R/W: Read and write possible
- R: Read only
- W: Write only

❍ **Explanation of initial values**

- 0: The initial value of this bit is "0."
- 1: The initial value of this bit is "1."
- X: The initial value of this bit is undefined.
- M: The initial value of this bit is the mask option.
- -: This bit is unused.

Note:

Do not use the vacancy.

APPENDIX B Overview of Instructions

Appendix B describes the instructions used by the F^2MC-8L.

B.1 Overview of F^{2}MC-8L Instructions

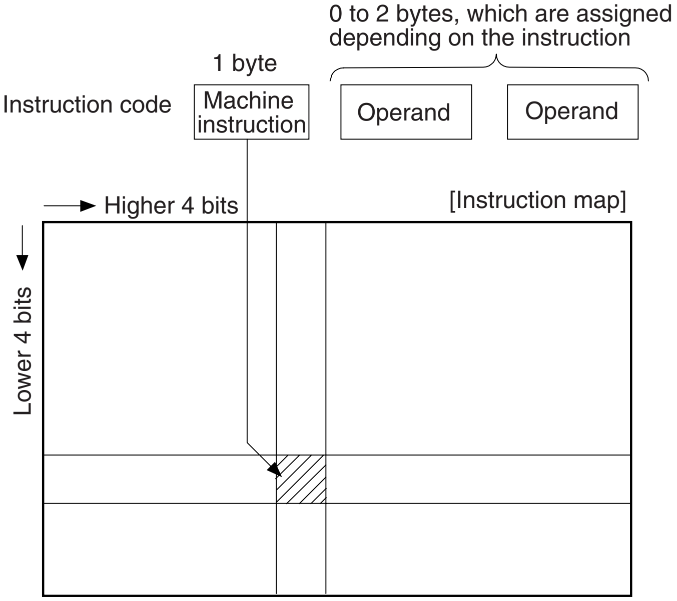
The F^{2}MC-8L supports 140 types of instructions.

■ Overview of F^{2}MC-8L Instructions

The F^{2}MC-8L has 140 1-byte machine instructions (256-byte instruction map). An instruction code consists of an instruction and zero or more operands that follow.

Figure B.1-1 "Relationship between the Instruction Codes and the Instruction Map" shows the relationship between the instruction codes and the instruction map.

Figure B.1-1 Relationship between the Instruction Codes and the Instruction Map



- The instructions are classified into four types: transfer, arithmetic, branch, and other.
- A variety of addressing methods is available. One of ten addressing modes can be selected depending on the selected instruction and specified operand(s).
- Bit manipulation instructions are provided. They can be used for read-modify-write operations.
- Some instructions are used for special operations.

■ Symbols used with Instructions

Table B.1-1 "Symbols in the Instruction List" lists the symbols used in the instruction code descriptions in Appendix B.

Table B.1-1 Symbols in the Instruction List

Symbol	Meaning
dir	Direct address (8 bits)
off	Offset (8 bits)
ext	Extended address (16 bits)
#vct	Vector table number (3 bits)
#d8	Immediate data (8 bits)
#d16	Immediate data (16 bits)
dir:16	Bit direct address (8 bits:3 bits)
rel	Branch relative address (8 bits)
@	Register indirect addressing (examples: @A, @IX, @EP)
A	Accumulator (8 or 16 bits, which are determined depending on the instruction being used)
AH	Higher 8 bits of the accumulator (8 bits)
AL	Lower 8 bits of the accumulator (8 bits)
T	Temporary accumulator (8 or 16 bits, which are determined depending on the instruction being used)
TH	Higher 8 bits of the temporary accumulator (8 bits)
TL	Lower 8 bits of the temporary accumulator (8 bits)
IX	Index register (16 bits)
EP	Extra pointer (16 bits)
PC	Program counter (16 bits)
SP	Stack pointer (16 bits)
PS	Program status (16 bits)
dr	Either accumulator or index register (16 bits)
CCR	Condition code register (8 bits)
RP	Register bank pointer (5 bits)
Ri	General-purpose register (8 bits, i = 0 to 7)
X	X is immediate data (8 or 16 bits, which are determined depending on the instruction being used).
(X)	The content of X is to be accessed (8 or 16 bits, which are determined depending on the instruction being used).
((X))	The address indicated by the X is to be accessed (8 or 16 bits, which are determined depending on the instruction being used).

B.2 Addressing

The F^2MC-8L has the following ten addressing modes:

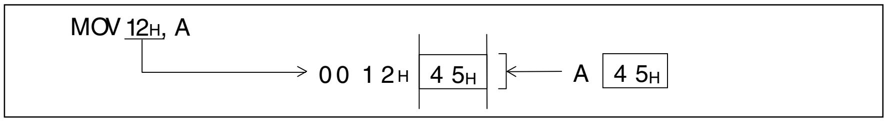
- **Direct addressing**
- **Extended addressing**
- **Bit direct addressing**
- **Index addressing**
- **Pointer addressing**
- **General-purpose register addressing**
- **Immediate addressing**
- **Vector addressing**
- **Relative addressing**
- **Inherent addressing**

■ Explanation of Addressing

❍ Direct addressing

Direct addressing is indicated by dir in the instruction list. This addressing is used to access the area between 0000_H and $00FF_H$. In this addressing mode, the higher byte of the address is 00_H and the lower byte is specified by the operand. Figure B.2-1 "Example of Direct Addressing" shows an example.

Figure B.2-1 Example of Direct Addressing

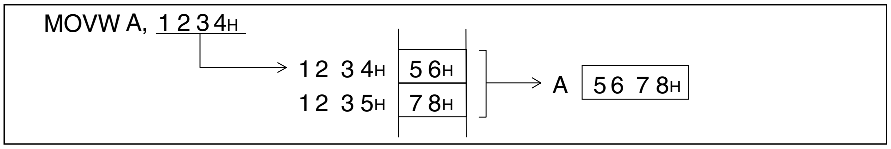


❍ Extended addressing

Extended addressing is indicated by ext in the instruction list. This addressing is used to access the entire 64-KB area. In this addressing mode, the first operand specifies the higher byte of the address, and the second operand specifies the lower byte.

Figure B.2-2 "Example of Extended Addressing" shows an example.

Figure B.2-2 Example of Extended Addressing

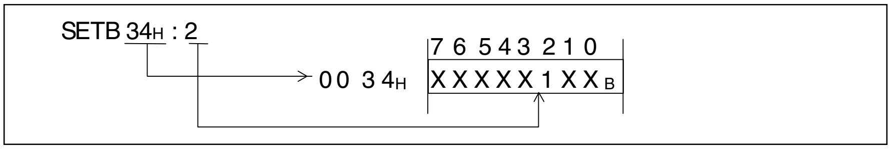


❍ Bit direct addressing

Bit direct addressing is indicated by dir:b in the instruction list. This addressing is used to access a particular bit in the area between 0000_H and $00FF_H$. In this addressing mode, the higher byte of the address is 00_H and the lower byte is specified by the operand. The bit position at the address is specified by the lower three bits of the operation code.

Figure B.2-3 "Example of Bit Direct Addressing" shows an example.

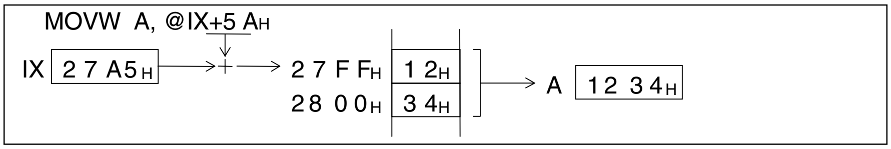
Figure B.2-3 Example of Bit Direct Addressing



❍ Index addressing

Index addressing is indicated by @IX+off in the instruction list. This addressing is used to access the entire 64-KB area. In this addressing mode, the address is the value resulting from sign-extending the contents of the first operand and adding them to IX (index register). Figure B.2-4 "Example of Index Addressing" shows an example.

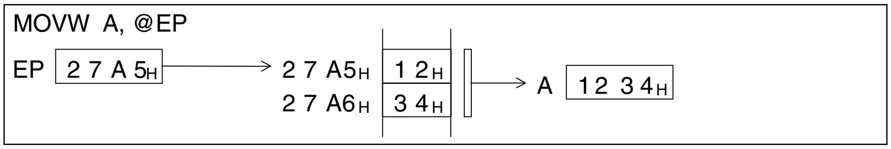
Figure B.2-4 Example of Index Addressing



❍ Pointer addressing

Pointer addressing is indicated by @EP in the instruction list. This addressing is used to access the entire 64-KB area. In this addressing mode, the address is contained in EP (extra pointer). Figure B.2-5 "Example of Pointer Addressing" shows an example.

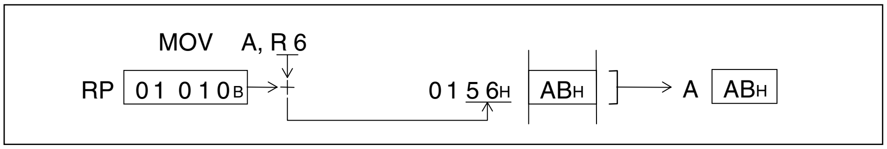
Figure B.2-5 Example of Pointer Addressing



❍ General-purpose register addressing

General-purpose register addressing is indicated by Ri in the instruction list. This addressing is used to access a register bank in the general-purpose register area. In this addressing mode, the higher byte of the address is always 01 and the lower byte is specified based on the contents of RP (register bank pointer) and the lower three bits of the operation code. Figure B.2-6 "Example of general-purpose register addressing" shows an example.

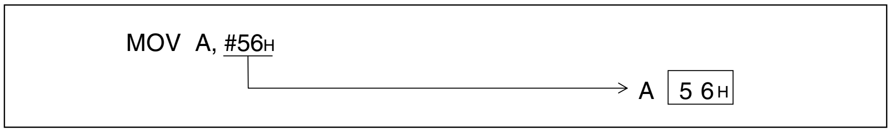
Figure B.2-6 Example of General-purpose Register Addressing



❍ Immediate addressing

Immediate addressing is indicated by #d8 in the instruction list. This addressing is used when immediate data is required. In this addressing mode, the operand is used as immediate data. Whether the data is specified in bytes or words is determined by the operation code. Figure B.2-7 "Example of Immediate Addressing" shows an example.

Figure B.2-7 Example of Immediate Addressing



❍ Vector addressing

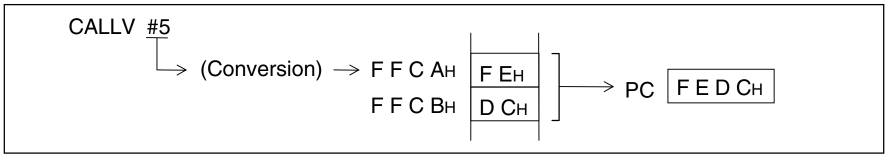
Vector addressing is indicated by vct in the instruction list. This addressing is used to branch to a subroutine address stored in the vector table. In this addressing mode, vct information is contained in the operation codes, and the corresponding table addresses are created as shown in Table B.2-1 "Vector Table Addresses Corresponding to vct".

Table B.2-1 Vector Table Addresses Corresponding to vct

#vct	Vector table address (higher address:lower address of branch destination)
0	$FFC0_H$: $FFC1_H$
1	$FFC2_H$: $FFC3_H$
2	$FFC4_H$: $FFC5_H$
3	$FFC6_H$: $FFC7_H$
4	$FFC8_H$: $FFC9_H$
5	$FFCA_H$: $FFCB_H$
6	$FFCC_H$: $FFCD_H$
7	$FFCE_H$: $FFCF_H$

Figure B.2-8 "Example of Vector Addressing" shows an example.

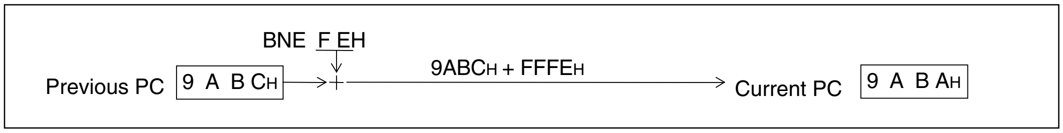
Figure B.2-8 Example of Vector Addressing



❍ Relative addressing

Relative addressing is indicated by rel in the instruction list. This addressing is used to branch to within the area between the address 128 bytes higher and that 128 bytes lower relative to the address contained in the PC (program counter). In this addressing mode, the result of a signed addition of the contents of the operand to the PC is stored in the PC. Figure B.2-9 "Example of Relative Addressing" shows an example.

Figure B.2-9 Example of Relative Addressing

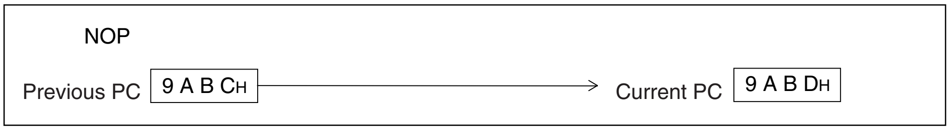


In this example, a branch to the address of the BNE operation code occurs, thus resulting in an infinite loop.

❍ Inherent addressing

Inherent addressing is indicated as the addressing without operands in the instruction list. This addressing is used to perform the operation determined by the operation code. In this addressing mode, different operations are performed via different instructions. Figure B.2-10 "Example of Inherent Addressing" shows an example.

Figure B.2-10 Example of Inherent Addressing



B.3 Special Instructions

This section describes the special instructions used for other than addressing.

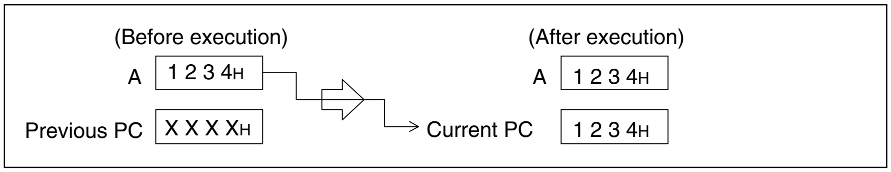
■ Special Instructions

❍ JMP @A

This instruction sets the contents of A (accumulator) to PC (program counter) as the address, and causes a branch to that address. One of the N branch destination addresses is selected from a table, and then transferred to A. The instruction can be executed to perform N-branch processing.

Figure B.3-1 "JMP @A" shows a summary of the instruction.

Figure B.3-1 JMP @A

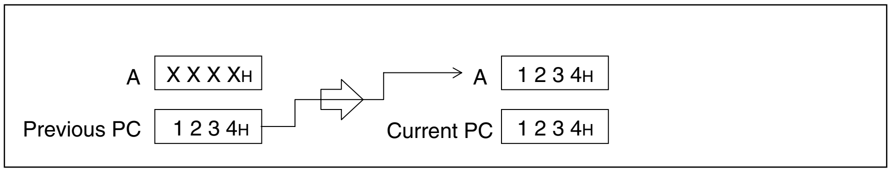


❍ MOVW A, PC

This instruction performs the operation which is the reverse of that performed by JMP @A. That is, the instruction stores the contents of PC in A. When the instruction is executed in the main routine, so that a specific subroutine is called, whether A contains a predetermined value can be checked by the subroutine. This can be used to determine that the branch source is not any unexpected section of the program and to check for program runaway.

Figure B.3-2 "MOVW A, PC" shows a summary of the instruction.

Figure B.3-2 MOVW A, PC



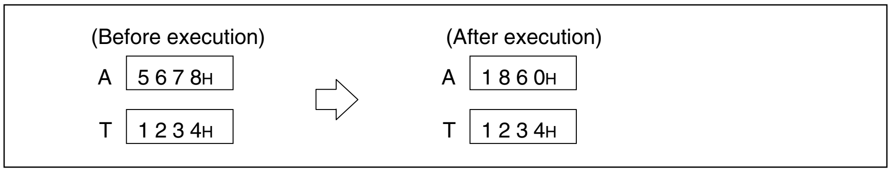
After the MOVW A, PC instruction is executed, A contains the address of the operation code of the next instruction, rather than the address of the operation code of MOVW A, PC. Accordingly, Figure B.3-2 "MOVW A, PC" shows that A contains 1234_H, which is the address of the operation code of the instruction that follows MOVW A, PC.

❍ MULU A

This instruction performs an unsigned multiplication of AL (lower eight bits of the accumulator) and TL (lower eight bits of the temporary accumulator), and stores the 16-bit result in A. The contents of T (temporary accumulator) do not change. The contents of AH (higher eight bits of the accumulator) and TH (higher eight bits of the temporary accumulator) before execution of the instruction are not used for the operation. The instruction does not change the flags, and therefore care must be taken when a branch may occur depending on the result of a multiplication.

Figure B.3-3 "MULU" shows a summary of the instruction.

Figure B.3-3 MULU



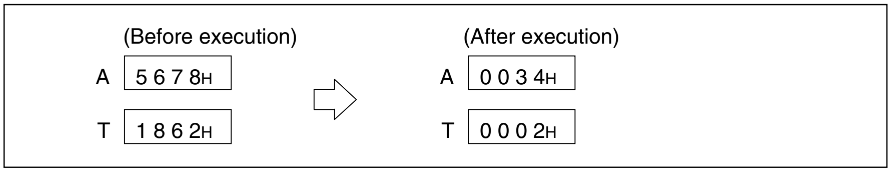
❍ DIVU A

This instruction divides the 16-bit value in T by the unsigned 8-bit value in AL, and stores the 8-bit result and the 8-bit remainder in AL and TL, respectively. A value of 0 is set to both AH and TH. The contents of AH before execution of the instruction are not used for the operation. An unpredictable result is produced from data that results in more than eight bits. In addition, there is no indication of the result having more than eight bits. Therefore, if it is likely that data will cause a result of more than eight bits, the data must be checked to ensure that the result will not have more than eight bits before it is used.

The instruction does not change the flags, and therefore care must be taken when a branch may occur depending on the result of a division.

Figure B.3-4 "DIVU A" shows a summary of the instruction.

Figure B.3-4 DIVU A

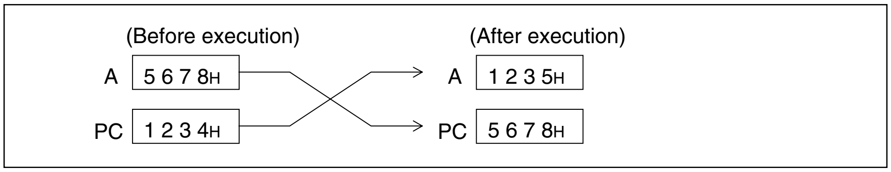


❍ XCHW A, PC

This instruction swaps the contents of A and PC, resulting in a branch to the address contained in A before execution of the instruction. After the instruction is executed, A contains the address that follows the address of the operation code of MOVW A, PC. This instruction is effective especially when it is used in the main routine to specify a table for use in a subroutine.

Figure B.3-5 "XCHW A, PC" shows a summary of the instruction.

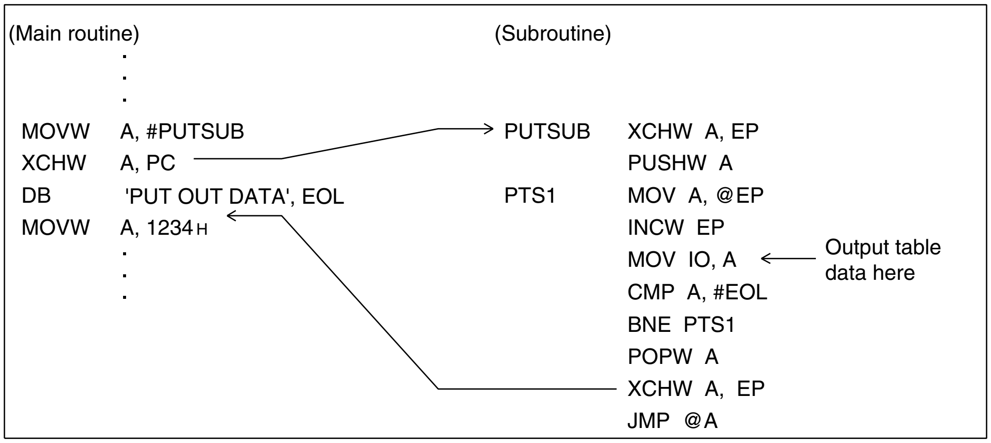
Figure B.3-5 XCHW A, PC



After the XCHW A, PC instruction is executed, A contains the address of the operation code of the next instruction, rather than the address of the operation code of XCHW A, PC. Accordingly, Figure B.3-5 "XCHW A, PC" shows that A contains 1235_H, which is the address of the operation code of the instruction that follows XCHW A, PC. This is why 1235_H is stored instead of $1234_{H.}$

Figure B.3-6 "Example of Using XCHW A, PC" shows an assembly language example.

Figure B.3-6 Example of Using XCHW A, PC



```
(Main routine)                              (Subroutine)
            .
            .
            .
 MOVW   A, #PUTSUB                           PUTSUB   XCHW  A, EP
 XCHW   A, PC                                         PUSHW  A
 DB     'PUT OUT DATA', EOL                  PTS1     MOV  A, @EP
 MOVW   A, 1234H                                      INCW  EP
            .                                         MOV  IO, A
            .                                         CMP  A, #EOL
            .                                         BNE  PTS1
                                                      POPW  A
                                                      XCHW  A,  EP
                                                      JMP  @A
```

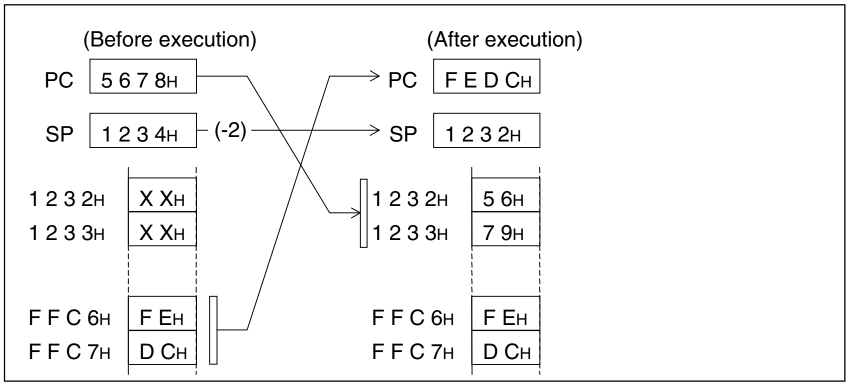
Output table data here

❍ CALLV #vct

This instruction is used to branch to a subroutine address stored in the vector table. The instruction saves the return address (contents of PC) in the location at the address contained in SP (stack pointer), and uses vector addressing to cause a branch to the address stored in the vector table. Because CALLV #vct is a 1-byte instruction, the use of this instruction for frequently used subroutines can reduce the entire program size.

Figure B.3-7 "Example of Executing CALLV #3" shows a summary of the instruction.

Figure B.3-7 Example of Executing CALLV #3



After the CALLV #vct instruction is executed, the contents of PC saved on the stack area are the address of the operation code of the next instruction, rather than the address of the operation code of CALLV #vct. Accordingly, Figure B.3-7 "Example of Executing CALLV #3" shows that the value saved in the stack (1232_H and 1233_H) is 5679_H, which is the address of the operation code of the instruction that follows CALLV #vct (return address).

B.4 Bit Manipulation Instructions (SETB, CLRB)

Some bits of peripheral function registers include bits that are read by a bit manipulation instruction differently than usual.

■ Read-modify-write Operation

By using these bit manipulation instructions, only the specified bit in a register or RAM location can be set to 1 (SETB) or cleared to 0 (CLRB). However, as the CPU operates on data in 8-bit units, the actual operation (read-modify-write operation) involves a sequence of steps: 8-bit data is read, the specified bit is changed, and the data is written back to the location at the original address.

Table B.4-1 "Bus Operation for Bit Manipulation Instructions" shows bus operation for bit manipulation instructions.

Table B.4-1 Bus operation for Bit Manipulation Instructions

CODE	MNEMONIC	TO	Cycle	Address bus	Data bus	$\overline{\text{RD}}$	$\overline{\text{WR}}$	RMW
A0 to A7	CLRB dir:b	4	1	N+1	dir	0	1	0
			2	dir address	Data	0	1	1
A8 to AF	SETB dir:b		3	dir address	Data	1	0	0
			4	N+2	Next instruction	0	1	0

■ Read Operation Upon the Execution of Bit Manipulation Instructions

For some I/O ports and for the interrupt request flag bits, the value to be read differs between a normal read operation and a read-modify-write operation.

❍ I/O ports (during a bit manipulation)

From some I/O ports, an I/O pin value is read during a normal read operation, while an output latch value is read during a bit manipulation. This prevents the other output latch bits from being changed accidentally, regardless of the I/O directions and states of the pins.

❍ Interrupt request flag bits (during a bit manipulation)

An interrupt request flag bit functions as a flag bit indicating whether an interrupt request exists during a normal read operation. However, 1 is always read from this bit during a bit manipulation. This prevents the flag from being cleared accidentally by a value of 0 which would otherwise be written to the interrupt request flag bit when another bit is manipulated.

B.5 F^{2}MC-8L Instructions

Table B.5-1 "Transfer Instructions" to Table B.5-4 "Other Instructions" list the instructions used with the F^{2}MC-8L.

■ Transfer Instructions

Table B.5-1 Transfer Instructions

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
1	MOV dir, A	3	2	(dir)<--(A)	-	-	-	-	-	-	-	45
2	MOV @IX+off, A	4	2	((IX)+off)<--(A)	-	-	-	-	-	-	-	46
3	MOV ext, A	4	3	(ext)<--(A)	-	-	-	-	-	-	-	61
4	MOV @EP, A	3	1	((EP))<--(A)	-	-	-	-	-	-	-	47
5	MOV Ri, A	3	1	(Ri)<--(A)	-	-	-	-	-	-	-	48 to 4F
6	MOV A, #d8	2	2	(A)<--d8	AL	-	-	+	+	-	-	04
7	MOV A, dir	3	2	(A)<--(dir)	AL	-	-	+	+	-	-	05
8	MOV A, @IX+off	4	2	(A)<--((IX)+off)	AL	-	-	+	+	-	-	06
9	MOV A, ext	4	3	(A)<--(ext)	AL	-	-	+	+	-	-	60
10	MOV A, @A	3	1	(A)<--((A))	AL	-	-	+	+	-	-	92
11	MOV A, @EP	3	1	(A)<--((EP))	AL	-	-	+	+	-	-	07
12	MOV A, Ri	3	1	(A)<--(Ri)	AL	-	-	+	+	-	-	08 to 0F
13	MOV dir, #d8	4	3	(dir)<--d8	-	-	-	-	-	-	-	85
14	MOV @IX+off, #d8	5	3	((IX)+off)<--d8	-	-	-	-	-	-	-	86
15	MOV @EP, #d8	4	2	((EP))<--d8	-	-	-	-	-	-	-	87
16	MOV Ri, #d8	4	2	(Ri)<--d8	-	-	-	-	-	-	-	88 to 8F
17	MOVW dir, A	4	2	(dir)<--(AH), (dir+1)<--(AL)	-	-	-	-	-	-	-	D5
18	MOVW @IX+off, A	5	2	((IX)+off)<--(AH), ((IX)+off+1)<--(AL)	-	-	-	-	-	-	-	D6
19	MOVW ext, A	5	3	(ext)<--(AH), (ext+1)<--(AL)	-	-	-	-	-	-	-	D4
20	MOVW @EP, A	4	1	((EP))<--(AH), ((EP)+1)<--(AL)	-	-	-	-	-	-	-	D7
21	MOVW EP, A	2	1	(EP)<--(A)	-	-	-	-	-	-	-	E3
22	MOVW A, #d16	3	3	(A)<--d16	AL	AH	dH	+	+	-	-	E4
23	MOVW A, dir	4	2	(AH)<--(dir), (AL)<--(dir+1)	AL	AH	dH	+	+	-	-	C5

Table B.5-1 Transfer Instructions (Continued)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
24	MOVW A, @IX+off	5	2	(AH)<--((IX)+off), (AL)<--((IX)+off+1)	AL	AH	dH	+	+	-	-	C6
25	MOVW A, ext	5	3	(AH)<--(ext), (AL)<--(ext+1)	AL	AH	dH	+	+	-	-	C4
26	MOVW A, @A	4	1	(AH)<--((A)), (AL)<--((A)+1)	AL	AH	dH	+	+	-	-	93
27	MOVW A, @EP	4	1	(AH)<--((EP)), (AL)<--((EP)+1)	AL	AH	dH	+	+	-	-	C7
28	MOVW A, EP	2	1	(A)<--(EP)	-	-	dH	-	-	-	-	F3
29	MOVW EP, #d16	3	3	(EP)<--d16	-	-	-	-	-	-	-	E7
30	MOVW IX, A	2	1	(IX)<--(A)	-	-	-	-	-	-	-	E2
31	MOVW A, IX	2	1	(A)<--(IX)	-	-	dH	-	-	-	-	F2
32	MOVW SP, A	2	1	(SP)<--(A)	-	-	-	-	-	-	-	E1
33	MOVW A, SP	2	1	(A)<--(SP)	-	-	dH	-	-	-	-	F1
34	MOV @A, T	3	1	((A))<--(T)	-	-	-	-	-	-	-	82
35	MOVW @A, T	4	1	((A))<--(TH), ((A)+1)<--(TL)	-	-	-	-	-	-	-	83
36	MOVW IX, #d16	3	3	(IX)<--d16	-	-	-	-	-	-	-	E6
37	MOVW A, PS	2	1	(A)<--(PS)	-	-	dH	-	-	-	-	70
38	MOVW PS, A	2	1	(PS)<--(A)	-	-	-	+	+	+	+	71
39	MOVW SP, #d16	3	3	(SP)<--d16	-	-	-	-	-	-	-	E5
40	SWAP	2	1	(AH)<-- -->(AL)	-	-	AL	-	-	-	-	10
41	SETB dir:b	4	2	(dir):b <--1	-	-	-	-	-	-	-	A8 to AF
42	CLRB dir:b	4	2	(dir):b <--0	-	-	-	-	-	-	-	A0 to A7
43	XCH A, T	2	1	(AL)<-- -->(TL)	AL	-	-	-	-	-	-	42
44	XCHW A, T	3	1	(A)<-- -->(T)	AL	AH	dH	-	-	-	-	43
45	XCHW A, EP	3	1	(A)<-- -->(EP)	-	-	dH	-	-	-	-	F7
46	XCHW A, IX	3	1	(A)<-- -->(IX)	-	-	dH	-	-	-	-	F6
47	XCHW A, SP	3	1	(A)<-- -->(SP)	-	-	dH	-	-	-	-	F5
48	MOVW A, PC	2	1	(A)<--(PC)	-	-	dH	-	-	-	-	F0

Note:

In automatic transfer to T during byte transfer to A, AL is transferred to TL.

If an instruction has two or more operands, they are assumed to be saved in the order indicated by MNEMONIC.

■ Arithmetic Instructions

Table B.5-2 Arithmetic Operation Instructions

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
1	ADDC A, Ri	3	1	(A)<--(A)+(Ri)+C	-	-	-	+	+	+	+	28 to 2F
2	ADDC A, #d8	2	2	(A)<--(A)+d8+C	-	-	-	+	+	+	+	24
3	ADDC A, dir	3	2	(A)<--(A)+(dir)+C	-	-	-	+	+	+	+	25
4	ADDC A, @IX+off	4	2	(A)<--(A)+((IX)+off)+C	-	-	-	+	+	+	+	26
5	ADDC A, @EP	3	1	(A)<--(A)+((EP))+C	-	-	-	+	+	+	+	27
6	ADDCW A	3	1	(A)<--(A)+(T)+C	-	-	dH	+	+	+	+	23
7	ADDC A	2	1	(AL)<--(AL)+(TL)+C	-	-	-	+	+	+	+	22
8	SUBC A, Ri	3	1	(A)<--(A)-(Ri)-C	-	-	-	+	+	+	+	38 to 3F
9	SUBC A, #d8	2	2	(A)<--(A)-d8-C	-	-	-	+	+	+	+	34
10	SUBC A, dir	3	2	(A)<--(A)-(dir)-C	-	-	-	+	+	+	+	35
11	SUBC A, @IX+off	4	2	(A)<--(A)-((IX)+off)-C	-	-	-	+	+	+	+	36
12	SUBC A, @EP	3	1	(A)<--(A)-((EP))-C	-	-	-	+	+	+	+	37
13	SUBCW A	3	1	(A)<--(T)-(A)-C	-	-	dH	+	+	+	+	33
14	SUBC A	2	1	(AL)<--(TL)-(AL)-C	-	-	-	+	+	+	+	32
15	INC Ri	4	1	(Ri)<--(Ri)+1	-	-	-	+	+	+	-	C8 to CF
16	INCW EP	3	1	(EP)<--(EP)+1	-	-	-	-	-	-	-	C3
17	INCW IX	3	1	(IX)<--(IX)+1	-	-	-	-	-	-	-	C2
18	INCW A	3	1	(A)<--(A)+1	-	-	dH	+	+	-	-	C0
19	DEC Ri	4	1	(Ri)<--(Ri)-1	-	-	-	+	+	+	-	D8 to DF
20	DECW EP	3	1	(EP)<--(EP)-1	-	-	-	-	-	-	-	D3
21	DECW IX	3	1	(IX)<--(IX)-1	-	-	-	-	-	-	-	D2
22	DECW A	3	1	(A)<--(A)-1	-	-	dH	+	+	-	-	D0
23	MULU A	19	1	(A)<--(AL)x(TL)	-	-	dH	-	-	-	-	01
24	DIVU A	21	1	(A)<--(T)/(AL), MOD -->(T)	dL	00	00	-	-	-	-	11
25	ANDW A	3	1	(A)<--(A) ∧ (T)	-	-	dH	+	+	R	-	63
26	ORW A	3	1	(A)<--(A) ∨ (T)	-	-	dH	+	+	R	-	73
27	XORW A	3	1	(A)<--(A) ∀ (T)	-	-	dH	+	+	R	-	53
28	CMP A	2	1	(TL)-(AL)	-	-	-	+	+	+	+	12
29	CMPW A	3	1	(T)-(A)	-	-	-	+	+	+	+	13
30	RORC A	2	1	→ C --> A ┐	-	-	-	+	+	-	+	03

Table B.5-2 Arithmetic Operation Instructions (Continued)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
31	ROLC A	2	1	C <-- A	-	-	-	+	+	-	+	02
32	CMP A, #d8	2	2	(A)-d8	-	-	-	+	+	+	+	14
33	CMP A, dir	3	2	(A)-(dir)	-	-	-	+	+	+	+	15
34	CMP A, @EP	3	1	(A)-((EP))	-	-	-	+	+	+	+	17
35	CMP A, @IX+off	4	2	(A)-((IX)+off)	-	-	-	+	+	+	+	16
36	CMP A, Ri	3	1	(A)-(Ri)	-	-	-	+	+	+	+	18 to 1F
37	DAA	2	1	decimal adjust for addition	-	-	-	+	+	+	+	84
38	DAS	2	1	decimal adjust for subtraction	-	-	-	+	+	+	+	94
39	XOR A	2	1	(A)<--(AL) ∀ (TL)	-	-	-	+	+	R	-	52
40	XOR A, #d8	2	2	(A)<--(AL) ∀ d8	-	-	-	+	+	R	-	54
41	XOR A, dir	3	2	(A)<--(AL) ∀ (dir)	-	-	-	+	+	R	-	55
42	XOR A, @EP	3	1	(A)<--(AL) ∀ ((EP))	-	-	-	+	+	R	-	57
43	XOR A, @IX+off	4	2	(A)<--(AL) ∀ ((IX)+off)	-	-	-	+	+	R	-	56
44	XOR A, Ri	3	1	(A)<--(AL) ∀ (Ri)	-	-	-	+	+	R	-	58 to 5F
45	AND A	2	1	(A)<--(AL) ∧ (TL)	-	-	-	+	+	R	-	62
46	AND A, #d8	2	2	(A)<--(AL) ∧ d8	-	-	-	+	+	R	-	64
47	AND A, dir	3	2	(A)<--(AL) ∧ (dir)	-	-	-	+	+	R	-	65
48	AND A, @EP	3	1	(A)<--(AL) ∧ ((EP))	-	-	-	+	+	R	-	67
49	AND A, @IX+off	4	2	(A)<--(AL) ∧ ((IX)+off)	-	-	-	+	+	R	-	66
50	AND A, Ri	3	1	(A)<--(AL) ∧ (Ri)	-	-	-	+	+	R	-	68 to 6F
51	OR A	2	1	(A)<--(AL) ∨ (TL)	-	-	-	+	+	R	-	72
52	OR A, #d8	2	2	(A)<--(AL) ∨ d8	-	-	-	+	+	R	-	74
53	OR A, dir	3	2	(A)<--(AL) ∨ (dir)	-	-	-	+	+	R	-	75
54	OR A, @EP	3	1	(A)<--(AL) ∨ ((EP))	-	-	-	+	+	R	-	77
55	OR A, @IX+off	4	2	(A)<--(AL) ∨ ((IX)+off)	-	-	-	+	+	R	-	76
56	OR A, Ri	3	1	(A)<--(AL) ∨ (Ri)	-	-	-	+	+	R	-	78 to 7F
57	CMP dir, #d8	5	3	(dir)-d8	-	-	-	+	+	+	+	95
58	CMP @EP, #d8	4	2	((EP))-d8	-	-	-	+	+	+	+	97
59	CMP @IX+off, #d8	5	3	((IX)+off)-d8	-	-	-	+	+	+	+	96
60	CMP Ri, #d8	4	2	(Ri)-d8	-	-	-	+	+	+	+	98 to 9F
61	INCW SP	3	1	(SP)<--(SP)+1	-	-	-	-	-	-	-	C1
62	DECW SP	3	1	(SP)<--(SP)-1	-	-	-	-	-	-	-	D1

■ Branch Instructions

Table B.5-3 Branch Instructions

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
1	BZ/BEQ rel	3	2	if Z=1 then PC<--PC+rel	-	-	-	-	-	-	-	FD
2	BNZ/BNE rel	3	2	if Z=0 then PC<--PC+rel	-	-	-	-	-	-	-	FC
3	BC/BLO rel	3	2	if C=1 then PC<--PC+rel	-	-	-	-	-	-	-	F9
4	BNC/BHS rel	3	2	if C=0 then PC<--PC+rel	-	-	-	-	-	-	-	F8
5	BN rel	3	2	if N=1 then PC<--PC+rel	-	-	-	-	-	-	-	FB
6	BP rel	3	2	if N=0 then PC<--PC+rel	-	-	-	-	-	-	-	FA
7	BLT rel	3	2	if V ∀ N=1 then PC<--PC+rel	-	-	-	-	-	-	-	FF
8	BGE rel	3	2	if V ∀ N=0 then PC<--PC+rel	-	-	-	-	-	-	-	FE
9	BBC dir:b, rel	5	3	if (dir:b)=0 then PC<--PC+rel	-	-	-	-	+	-	-	B0 to B7
10	BBS dir:b, rel	5	3	if (dir:b)=1 then PC<--PC+rel	-	-	-	-	+	-	-	B8 to BF
11	JMP @A	2	1	(PC)<--(A)	-	-	-	-	-	-	-	E0
12	JMP ext	3	3	(PC)<--ext	-	-	-	-	-	-	-	21
13	CALLV #vct	6	1	vector call	-	-	-	-	-	-	-	E8 to EF
14	CALL ext	6	3	subroutine call	-	-	-	-	-	-	-	31
15	XCHW A, PC	3	1	(PC)<--(A), (A)<--(PC)+1	-	-	dH	-	-	-	-	F4
16	RET	4	1	return from subroutine	-	-	-	-	-	-	-	20
17	RETI	6	1	return from interrupt	-	-	-	restore				30

■ Other Instructions

Table B.5-4 Other Instructions

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
1	PUSHW A	4	1		-	-	-	-	-	-	-	40
2	POPW A	4	1		-	-	dH	-	-	-	-	50
3	PUSHW IX	4	1		-	-	-	-	-	-	-	41
4	POPW IX	4	1		-	-	-	-	-	-	-	51
5	NOP	1	1		-	-	-	-	-	-	-	00
6	CLRC	1	1		-	-	-	-	-	-	R	81
7	SETC	1	1		-	-	-	-	-	-	S	91
8	CLRI	1	1		-	-	-	-	-	-	-	80
9	SETI	1	1		-	-	-	-	-	-	-	90

B.6 Instruction Map

Table B.6-1 "F^{2}MC-8L Instruction Map" shows the F^{2}MC-8L instruction map.

■ Instruction map

Table B.6-1 F^{2}MC-8L Instruction Map

H L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SWAP	RET	RETI	PUSHW A	POPW A	MOV A, ext	MOVW A, PS	CLRI	SETI	CLRB dir : 0	BBC dir : 0, rel	INCW A	DECW A	JMP @A	MOVW A, PC
1	MULU A	DIVU A	JMP addr16	CALL addr16	PUSHW IX	POPW IX	MOV ext, A	MOVW PS, A	CLRC	SETC	CLRB dir : 1	BBC dir : 1, rel	INCW SP	DECW SP	MOVW SP, A	MOVW A, SP
2	ROLC A	CMP A	ADDC A	SUBC A	XCH A, T	XOR A	AND A	OR A	MOV @A, T	MOV A, @A	CLRB dir : 2	BBC dir : 2, rel	INCW IX	DECW IX	MOVW IX, A	MOVW A, IX
3	RORC A	CMPW A	ADDCW A	SUBCW A	XCHW A, T	XORW A	ANDW A	ORW A	MOVW @A, T	MOVW A, @A	CLRB dir : 3	BBC dir : 3, rel	INCW EP	DECW EP	MOVW EP, A	MOVW A, EP
4	MOV A, #d8	CMP A, #d8	ADDC A, #d8	SUBC A, #d8		XOR A, #d8	AND A, #d8	OR A, #d8	DAA	DAS	CLRB dir : 4	BBC dir : 4, rel	MOVW A, ext	MOVW ext, A	MOVW A, #d16	XCHW A, PC
5	MOV A, dir	CMP A, dir	ADDC A, dir	SUBC A, dir	MOV dir, A	XOR A, dir	AND A, dir	OR A, dir	MOV dir, #d8	CMP dir, #d8	CLRB dir : 5	BBC dir : 5, rel	MOVW A, dir	MOVW dir, A	MOVW SP, #d16	XCHW A, SP
6	MOV A, @IX+d	CMP A, @IX+d	ADDC A, @IX+d	SUBC A, @IX+d	MOV @IX+d, A	XOR A, @IX+d	AND A, @IX+d	OR A, @IX+d	MOV @IX+d,#d8	CMP @IX+d,#d8	CLRB dir : 6	BBC dir : 6, rel	MOVW A, @IX+d	MOVW @IX+d, A	MOVW IX, #d16	XCHW A, IX
7	MOV A, @EP	CMP A, @EP	ADDC A, @EP	SUBC A, @EP	MOV @EP, A	XOR A, @EP	AND A, @EP	OR A, @EP	MOV @EP#, d8	CMP @EP#, d8	CLRB dir : 7	BBC dir : 7, rel	MOVW A, @EP	MOVW @EP, A	MOVW EP, #d16	XCHW A, EP
8	MOV A, R0	CMP A, R0	ADDC A, R0	SUBC A, R0	MOV R0, A	XOR A, R0	AND A, R0	OR A, R0	MOV R0, #d8	CMP R0, #d8	SETB dir : 0	BBS dir : 0, rel	INC R0	DEC R0	CALLV #0	BNC rel
9	MOV A, R1	CMP A, R1	ADDC A, R1	SUBC A, R1	MOV R1, A	XOR A, R1	AND A, R1	OR A, R1	MOV R1, #d8	CMP R1, #d8	SETB dir : 1	BBS dir : 1, rel	INC R1	DEC R1	CALLV #1	BC rel
A	MOV A, R2	CMP A, R2	ADDC A, R2	SUBC A, R2	MOV R2, A	XOR A, R2	AND A, R2	OR A, R2	MOV R2, #d8	CMP R2, #d8	SETB dir : 2	BBS dir : 2, rel	INC R2	DEC R2	CALLV #2	BP rel
B	MOV A, R3	CMP A, R3	ADDC A, R3	SUBC A, R3	MOV R3, A	XOR A, R3	AND A, R3	OR A, R3	MOV R3, #d8	CMP R3, #d8	SETB dir : 3	BBS dir : 3, rel	INC R3	DEC R3	CALLV #3	BN rel
C	MOV A, R4	CMP A, R4	ADDC A, R4	SUBC A, R4	MOV R4, A	XOR A, R4	AND A, R4	OR A, R4	MOV R4, #d8	CMP R4, #d8	SETB dir : 4	BBS dir : 4, rel	INC R4	DEC R4	CALLV #4	BNZ rel
D	MOV A, R5	CMP A, R5	ADDC A, R5	SUBC A, R5	MOV R5, A	XOR A, R5	AND A, R5	OR A, R5	MOV R5, #d8	CMP R5, #d8	SETB dir : 5	BBS dir : 5, rel	INC R5	DEC R5	CALLV #5	BZ rel
E	MOV A, R6	CMP A, R6	ADDC A, R6	SUBC A, R6	MOV R6, A	XOR A, R6	AND A, R6	OR A, R6	MOV R6, #d8	CMP R6, #d8	SETB dir : 6	BBS dir : 6, rel	INC R6	DEC R6	CALLV #6	BGE rel
F	MOV A, R7	CMP A, R7	ADDC A, R7	SUBC A, R7	MOV R7, A	XOR A, R7	AND A, R7	OR A, R7	MOV R7, #d8	CMP R7, #d8	SETB dir : 7	BBS dir : 7, rel	INC R7	DEC R7	CALLV #7	BLT rel

APPENDIX C Mask Options

Table C-1 "Mask options" lists the mask options of the MB89530/530H/530A.

■ Mask options

Table C-1 Mask options

No.	Model	MB89535A/535AC MB89537/537H/537A MB89537C/537HC/537AC MB89538/538H/538A MB89538C/538HC/538AC	MB89P538-101 MB89P538-201	MB89PV530-101 MB89PV530-201	MB89F538L-101 MB89F538L-201 MB89F538-101 MB89F538-201
	Specification method	**Specified at ordering masking**	**Setting not possible**	**Setting not possible**	**Setting not possible**
1	Selection of main clock oscillation stabilization wait time (for F_{CH} = 10 MHz) Approx. $2^{14}/F_{CH}$ (approx. 1.6 ms) Approx. $2^{17}/F_{CH}$ (approx. 13.1 ms) Approx. $2^{18}/F_{CH}$ (approx. 26.2 ms)	Selectable	$2^{18}/F_{CH}$ (Approx. 26.2 ms)	$2^{18}/F_{CH}$ (Approx. 26.2 ms)	$2^{18}/F_{CH}$ (Approx. 26.2 ms)
2	Clock mode selection • Dual-clock system mode • Single-clock system mode	Selectable	-101: Single-clock system mode -201: Dual-clock system mode		

F_{CH}: Main clock frequency

APPENDIX D Write Specifications for the One-Time PROM and EPROM Microcomputer

MB89P538 has PROM mode that becomes a function equivalent to MBM27C1001. In this mode, writing is possible with the general-purpose ROM writer using the dedicated adapter. However, note that the electronic signature mode cannot be used.

■ ROM writer adapter

Some ROM writers can write data more stably when a capacitor of about 0.1 μF is inserted between V_{CC} and V_{SS}.

Table D-1 "ROM writer adapters" lists the ROM writer adapters.

Table D-1 ROM writer adapters

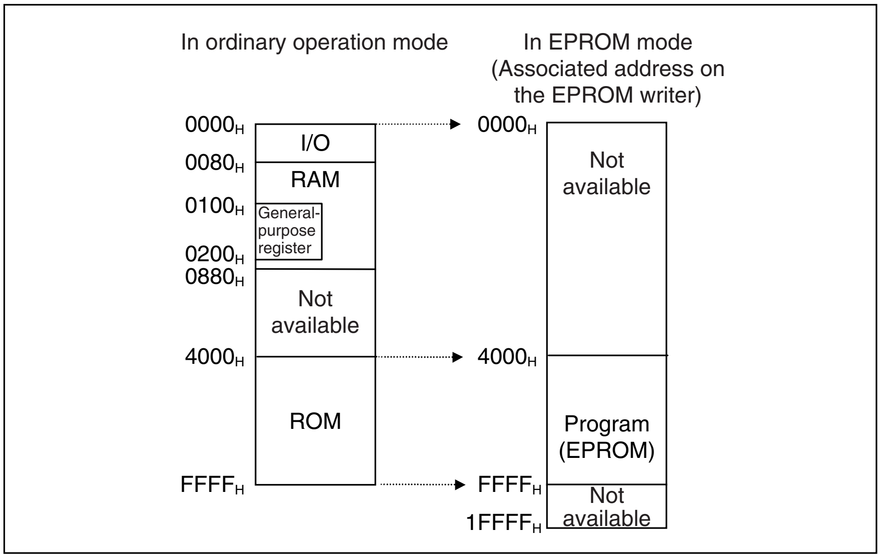
Product name	Package	Compatible adapter
MB89P538-101PF-G-BND MB89P538-201PF-G-BND	FTP-64P-M06	ROM-64QF-32DP-8LA2(*1)
MB89P538-101PFM-G-BND MB89P538-201PFM-G-BND	FTP-64P-M09	ROM-64QF2-32DP-8LA
MB89P538-101P-G-SH MB89P538-201P-G-SH	DIP-64P-M01	ROM-64SD-32DP-8LA2(*1)

Contact: Sunhayato Corporation
Tel. 03-3986-0403
*1 Use the adapters of version 3 or later.

■ Memory map in EPROM mode

Figure D-1 "Memory map in EPROM mode" shows the memory map in EPROM mode. The PROM option is not available.

Figure D-1 Memory map in EPROM mode

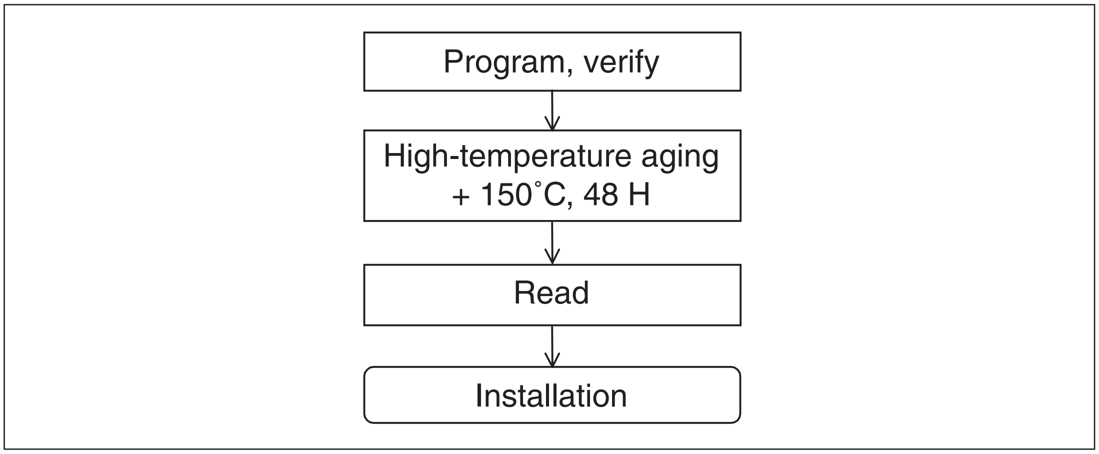


■ Recommended screening conditions

For a product with a blank one-time PROM microcomputer program, Fujitsu recommends using high-temperature aging as the screening method before installation.

Figure D-2 "Flow of screening" shows the flow of screening.

Figure D-2 Flow of screening



■ Programming yield

The all-bit programming at Fujitsu shipping test cannot be performed for a product with a blank one-time PROM microcomputer program. Therefore, a programming yield of 100% may not always be guaranteed.

APPENDIX E EPROM with Piggyback/Evaluation Chip

This section describes how to write EPROM to be mounted on the piggyback/ evaluation chip.

■ Usable EPROM

MBM27C512-20TV

■ Programming socket adapter

To write data into EPROM with the ROM writer, use the programming socket adapter (manufactured by Sunhayato Corporation) shown in Table E-1 "Programming socket adapter."

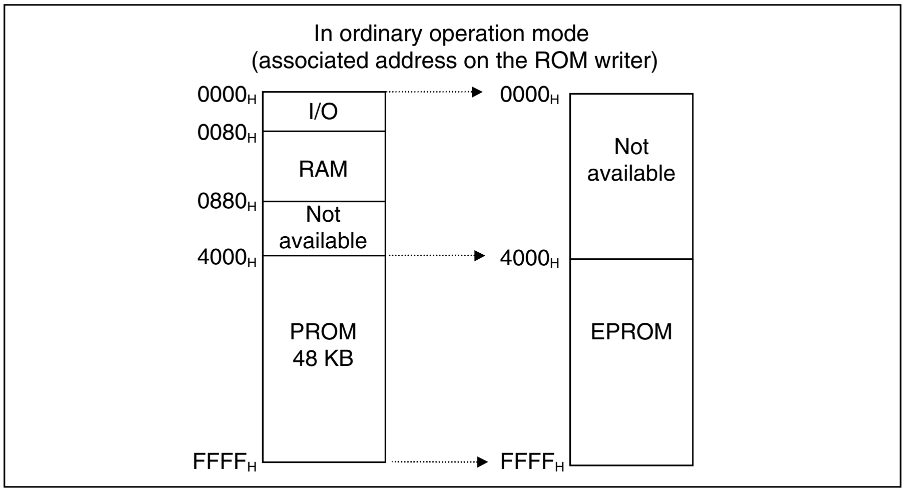
Table E-1 Programming socket adapter

Package	Adapter socket model
LCC-32 (rectangle)	ROM-32LC-28DP-YG

Contact: Sunhayato Corporation
Tel. 03-3986-0403

■ Memory space

Figure E-1 Memory map of the piggyback/evaluation chip



■ Method of writing to EPROM

1. Set the EPROM writer to MBM27C512.
2. Load program data to 4000_H to $FFFF_H$ of the EPROM writer.
3. Write 4000_H to $FFFF_H$ with the EPROM writer.

APPENDIX F Pin Statuses of the MB89530/530H/530A Series

Table F-1 "Pin status in each mode" lists the pin statuses of the MB89530/530H/530A series.

■ Pin statuses in each mode

Table F-1 Pin status in each mode

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		Reset ongoing
			SPL = "0"	SPL = "1"	SPL = "0"	SPL = "1"	
X0,X1	Oscillation circuit input	Oscillation circuit input	Hi-z	Hi-z	Hi-z	Hi-z	Oscillation circuit input
MOD0 MOD1	Mode input	Mode input	Mode input	Mode input	Mode input	Mode input	Mode input
$\overline{\text{RST}}$	Reset input	Reset input	Reset input	Reset input	Reset input	Reset input	Reset input
P00 to P17	Port/peripheral equipment I/O	Retention/ peripheral equipment I/O	Retention	Hi-z	Retention	Hi-z	Hi-z
P20/PWCK	Port/peripheral equipment I/O	Retention/ peripheral equipment I/O	Retention	"H" (for pull-up) Hi-z (for other than pull-up)	Retention	"H" (for pull-up) Hi-z (for other than pull-up)	Hi-z
P21/PPG01 to P22/ PPG02							
P23 to P27							
P30/PPG03/ MCO							
P31/ SCK1(UCK1) /LMCO							
P32/ SO1(UO1)							
P33/SI1(UI1)							
P34/PTO2							
P35/PWC							
P36/WTO							
P37/PTO1							

Table F-1 Pin status in each mode (Continued)

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		Reset ongoing
			SPL = "0"	SPL = "1"	SPL = "0"	SPL = "1"	
P40/INT20/ EC	Port/external interrupt 2 input/ peripheral equipment I/O	Retention/ external interrupt 2 input/ peripheral equipment I/O	Retention/ external interrupt 2 input	"H" (for pull-up) Hi-z (for other than pull-up)/ external interrupt 2 input	Retention/ external interrupt 2 input	"H" (for pull-up) Hi-z (for other than pull-up)/ external interrupt 2 input	Hi-z
P41/INT21/ SCK2							
P42/INT22/ SO2/SDA							
P43/INT23/ SI2/SCL							
P44/INT24/ UCK2							
P45/INT25/ UO2							
P46/INT26/ UI2							
P47/INT27/ ADST							
P50/AN0 to P57/AN7	Port/AD input	Retention/AD input	Retention	"H" (for pull-up) Hi-z (for other than pull-up)	Retention	"H" (for pull-up) Hi-z (for other than pull-up)	Hi-z
P60/INT10 to P62/INT12	Port/external interrupt 1 input	Retention/ external interrupt 1 input	Retention/ external interrupt 1 input	"H" (for pull-up) Hi-z (for other than pull-up)/ external interrupt 1 input	Retention/ external interrupt 1 input	"H" (for pull-up) Hi-z (for other than pull-up)/ external interrupt 1 input	Hi-z
P63/INT13/ X0A	Port/external interrupt 1 input/ subclock input	Retention/ external interrupt 1 input/ subclock input	Retention/ external interrupt 1 input	"H" (for pull-up) Hi-z (for other than pull-up)/ external interrupt 1 input	Subclock input	Subclock input	Oscillation circuit input
P64/X1A	Port/subclock output	Retention/ subclock output	Retention	"H" (for pull-up) Hi-z (for other than pull-up)	Subclock output	Subclock output	Oscillation circuit output

APPENDIX G   Troubleshooting

If a fault occurs, take appropriate action in accordance with the checklist shown below.
Alternatively, software may have caused the fault. Therefore, read the manual of the software in use together.

■ Checklist

Table G-1 Checklist

Symptom	Cause	Action	Check mark entry
The microcomputer operation is not normal.	Power supply (V_{CC}, GND) is not connected.	Connect the power supply (V_{CC}, GND).	
	The electrical characteristics of the input signal for each pin do not meet the specifications.	Make sure that the electrical characteristics of the input signal for each pin meet the specifications.	
	The MOD0 and MOD1 pins are not processed in the operating mode to be used.	Perform the MOD0 and MOD1 pin processing.	
	The oscillator has been connected. However, no oscillation is made at power-on and reset.	When oscillation is ongoing, check it to see whether the frequency of the oscillator being connected is used.	
		If oscillation is not ongoing, the status may have been changed to stop mode of the standby mode. Check the program.	
The microcomputer was operating during evaluation made by the tool. However, it does not operate for the one-time ROM product.	Each register and RAM are not initialized yet.	If operation is unstable even if power is turned on repeatedly, the initial value may not be set yet.	

■ Items to be confirmed for inquiry

For an inquiry, confirm the following and consult a person in charge of sales or a special agent:

1. Confirm the differences in operation between the normal and abnormal statuses, at the pin level (waveforms).
2. Confirm the frequency of generation of the problem, conditions, the number of times the problem is generated, and to what degree the problem generation depends on the voltage, temperature, and oscillator frequency.
3. Confirm the operation for the one-time product, mask ROM product, and piggyback/ evaluation product.

Index

CM25-10135-5E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

F^{2}MC-8L

8-BIT MICROCONTROLLER

MB89530/530H/530A Series

HARDWARE MANUAL

October 2002 the fifth edition

Published **FUJITSU LIMITED** Electronic Devices

Edited Standardization Promoting Dept.