FUJITSU SEMICONDUCTOR
CONTROLLER MANUAL

CM25-10153-2E

F²MC-8L

8-BIT MICROCONTROLLER

MB89202/F202RA Series HARDWARE MANUAL

FUJITSU

F²MC-8L

8-BIT MICROCONTROLLER

MB89202/F202RA Series

HARDWARE MANUAL

Be sure to refer to the “Check Sheet” for the latest cautions on development.

“Check Sheet” is seen at the following support page

URL:http://www.fujitsu.com/global/services/microelectronics/product/micom/support/index.html

“Check Sheet” lists the minimal requirement items to be checked to prevent problems beforehand in system development.

FUJITSU LIMITED

PREFACE

■ Purpose of This Manual and Intended Reader

The MB89202/F202RA series was developed as one of the general-purpose products of the F^2MC-8L family, which contains original 8-bit one-chip microcontrollers for use with ASICs (application specific ICs). The MB89202/F202RA series can be used in a wide range of products from consumer products to industrial products.

This manual explains the functions and operations of the MB89202/F202RA series for product development.

The F^2MC-8L Programming Manual contains details of the programming instructions.

Note: F^2MC, an abbreviation for FUJITSU Flexible Microcontroller, is a registered trademark of FUJITSU LIMITED.

■ Trademark

■ Structure of This Manual

This manual consists of the following 17 chapters and appendix.

CHAPTER 1 OVERVIEW

This chapter describes the features and basic specification of the MB89202/F202RA series.

CHAPTER 2 HANDLING DEVICES

This chapter describes the precautions to be taken when handling the MB89202/F202RA series.

CHAPTER 3 CPU

This chapter describes the functions and operation of the CPU.

CHAPTER 4 I/O PORTS

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

CHAPTER 5 TIME-BASE TIMER

This chapter describes the functions and operation of the time-base timer.

CHAPTER 6 WATCHDOG TIMER

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

CHAPTER 7 8-BIT PWM TIMER

This chapter describes the functions and operation of the 8-bit PWM timer.

CHAPTER 8 8/16-BIT CAPTURE TIMER/COUNTER

This chapter describes the functions and operation of the 8/16-bit capture timer/counter.

CHAPTER 9 12-BIT PPG TIMER

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

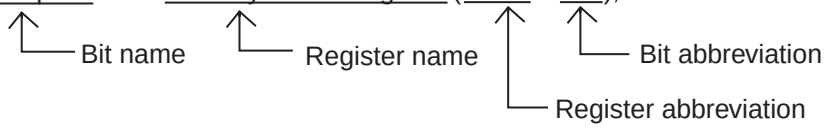
- The contents of this document are subject to change without notice.
  Customers are advised to consult with sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of FUJITSU semiconductor device; FUJITSU does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. FUJITSU assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of FUJITSU or any third party or does FUJITSU warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- 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.
- Exportation/release of any products described in this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.
- The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

READING THIS MANUAL

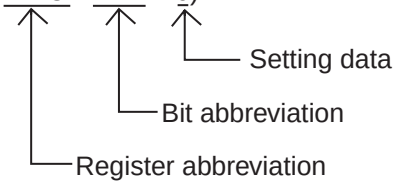
■ Example Notation of Register Names and Pin Names

❍ Example notation of register names and bit names

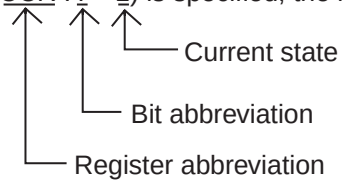
By writing 1 into the sleep bit of the standby control register (STBC : SLP), ...



Prohibit the output of interrupt request of the time-base timer (TBTC : TBIE = 0).



If interrupt enabled (CCR : I = 1) is specified, the interrupt is accepted.



❍ Example notation of multi-use pins

P33/EC pin

Some pins can switch functions according to a setting made by a program or other method. These pins are called multi-use pins. For multi-use pins, the names corresponding to functions are listed and divided by /.

CONTENTS

Main changes in this edition

Page	Changes (For details, refer to main body.)	
-	-	The followings product name is changed. (MB89202 →MB89202/F202RA)
		The followings term is changed. (source oscillation →oscillation frequency)
6	1.3 Differences between Models	"Notes:" is changed. (The followings sentence is deleted. "• At turning on the power, when the device is used without inputting the external reset, select "reset output supported" and "power-on reset supported" by mask option.")
		The followings package is changed in Table 1.3-1. (FPT-34P-M03 →FPT-32P-M03)
12	1.7 Pin Functions	$\overline{RST}$ pin in Table 1.7-1 is changed.
19	2.1 Precautions on Handling Devices	"● External pull-up for the External Reset Pin ($\overline{RST}$) of MB89F202/F202RA" is changed.
24	3.1.1 Specific-purpose Areas	The summary is changed.
		"■ General-purpose Register Area (address: 0100_H to $01FF_H$)" is changed.
		"■ Vector Table Area (Address: $FFC0_H$ to $FFFF_H$)" is changed.
44	3.5 Reset	"● Power-on reset" is changed.
		"Note:" is deleted.
56	3.6.3 System Clock Control Register (SYCC)	Figure 3.6-5 is changed.
57		Table 3.6-1 is changed.
130	6.3 Watchdog Control Register (WDTC)	Figure 6.3-1 is changed.
186	8.6 Explanation of Operations of Interval Timer Functions	"● 8-bit mode"is changed. (The followings sentence is deleted. "The initial value of the square wave output is "L" level. The square wave output is initialized by writing "0" to the TSTR bit of the timer control register (TCR).")
264	12.3 Pins of A/D Converter	"■ Block Diagram of the Pins Related to the A/D Converter" is changed. ("Note:" is deleted.)
308	13.6.2 Reception Operations (Operating Mode 0, 1, or 3)	"■ Reception Operations (Operating Mode 0, 1, or 3)" is changed. ("Note:" is changed.)

Page	Changes (For details, refer to main body.)	
310	13.6.3 Reception Operations (Operating Mode 2 Only)	"■ Reception Operations (Operating Mode 2 Only)" is changed. ("Note:" is changed.)
358	17.1 Overview of Flash Memory	"■ High voltage supply on $\overline{\text{RST}}$ pin (applicable to MB89F202RA only)" is added.
370	17.5.2 Writing Data	Figure 17.5-1 is changed. (F555 →F554)
394	B.4 F^2MC-8L Instructions List	Table B.4-2 is changed. ("No.22 DECW A" is changed.)

The vertical lines marked in the left side of the page show the changes.

CHAPTER 1
OVERVIEW

This chapter describes the features and basic specification of the MB89202/F202RA series.

1.1 Features of MB89202/F202RA Series

The MB89202/F202RA series contains general-purpose single-chip microcontrollers that incorporate a full range of peripheral functions such as A/D converter, UART, PWM timer, PPG, capture timer/counter and external interrupts as well as a compact instruction set.

■ Features of MB89202/F202RA Series

- F^2MC-8L CPU core
  - Instruction set most suitable for controllers
  - Multiplication and division instruction
  - 16-bit operation
  - Branch instruction by bit test
  - Bit operation instruction, and others
- 4-system timers
  - 8/16-bit capture timer/counter (8-bit capture timer/counter + 8-bit timer or 16-bit capture timer/counter)
  - 8-bit PWM timer (also available as an interval timer)
  - 21-bit time-base timer
  - Watchdog timer
- 10-bit A/D converter
  - 10-bit A/D × 8 channels
  - Activation by 8/16-bit capture timer/counter output is possible.
- Programmable pulse generator (PPG)
  - Pulse width and cycle are software selectable (12-bit PPG).
- UART
  - 6, 7, or 8 transfer data length
- 8-bit serial I/O
  - Available when switched from UART
  - LSB first/MSB first selectability
- External interrupts
  - External interrupt 1 (edge detection × 3 pins) has three independent inputs and can be used for wake-up from low-power consumption mode. (The edge detection can be selected from rising-edge, falling-edge, and both-edge modes.)

- External interrupt 2 (level detection × 8 pins, 1 channel) has eight independent inputs and can be used for wake-up from low-power consumption mode. (L level detection function is supported.)

● Low-power consumption modes (standby modes)

- Stop mode (The oscillation is stopped so that current consumption is minimal.)
- Sleep mode (The CPU is stopped so that the current consumption is reduced by one-third of normal consumption.)

● Up to 26 pins of I/O ports

- General-purpose I/O ports (CMOS): 26 pins (4 of which can be used as N-ch open-drain I/O ports.)

● Wild registers

- 2-byte data at two addresses are available.
- When a specific address or data is used on a wild register, the data in the ROM area is changed.

● 16 KB Flash with read protection

- Once the protection code is written in the specified address, the FLASH content cannot be read by parallel/serial programmer.

1.2 MB89202/F202RA Series Product Lineup

Four MB89202 series models are available. Table 1.2-1 shows the models and Table 1.2-2 shows the CPU and peripheral functions.

■ MB89202/F202RA Series Models

Table 1.2-1 MB89202/F202RA Series Models

	MB89201	MB89F202/F202RA	MB89V201
Classification	Evaluation product (for development)	Flash memory product (read protection)	Mask ROM product
ROM size	32K × 8 bits (External EPROM*2)	16K × 8 bits (Internal Flash)	16K × 8 bits (Internal mask ROM)
RAM size	512 × 8 bits		
Low-power consumption (standby mode)	Sleep mode and stop mode		
Process	CMOS		
Operating voltage*1	2.7V to 5.5V	3.5V to 5.5V	2.2V to 5.5V

*1: The minimum operating voltage varies with conditions such as operating frequency, functions, and connecting ICE.

*2: MBM27C256A is used as the external ROM.

Table 1.2-2 CPU and Peripheral Functions of MB89202/F202RA Series

Item		Specification
CPU function		Number of basic instructions: 136 instructions Instruction bit length: 8 bits Instruction length: 1 to 3 bytes Data bit length: 1, 8, or 16 bits Minimum instruction execution time: 0.32 to 5.1 μs (at 12.5 MHz) Interrupt processing time: 2.88 to 46.1 μs (at 12.5 MHz)
Peripheral function	Port	General-purpose I/O port: 26 pins (Also serve as peripherals. 4 of which can be used as N-ch open-drain I/O ports.)
	21-bit time-base timer	21 bits Interrupt cycle: 0.66 ms, 2.64 ms, 21 ms, or 335.5 ms with 12.5MHz main clock
	Watchdog timer	Reset occurrence cycle: When the main clock is at 12.5 MHz (minimum 335.5 ms)
	8-bit PWM timer	8-bit interval timer operation (Square wave output is supported. Operating clock cycle: 1 t_{INST}, 16 t_{INST}, 64 t_{INST}, and 8/16-bit capture timer/counter output) 8-bit resolution PWM operation (Conversion cycle: 256 t_{INST}, 4096 t_{INST}, 16384 t_{INST} and 256 times 8/16-bit capture timer/counter output)
	8/16-bit capture timer/ counter	8-bit capture timer/counter × 1 channel + 8-bit timer or 16-bit capture timer/counter × 1 channel When timer 0 or a 16-bit counter is operating, event-counting operation by external clock input and square wave output are supported.
	UART	Transfer data length: 6, 7, or 8 bits
	8-bit serial I/O	8 bits length, LSB first/MSB first selectability One clock selectable from four operation clocks (one external shift clock, three internal shift clocks: 2 t_{INST}, 8 t_{INST}, 32 t_{INST})
	12-bit PPG timer	Output frequency: Pulse width and cycle are selectable.
	External interrupt 1 (wake-up)	3 channels (interrupt vector, request flag, and request output enable) Edge selectability (selectable from rising edge, falling edge, and both-edge modes) Also available for wake-up from stop or sleep (Edge detection is also available in stop mode.)
	External interrupt 2 (wake-up)	8 inputs 1 channel (L level interrupt and input enable are independent.) Also available for wake-up from stop or sleep (Level detection is also available in stop mode.)
	10-bit A/D converter	10-bit resolution × 8 channels A/D conversion function (Conversion time: 38 t_{INST}) Continuous activation by 8/16-bit capture timer/counter output or time-base timer output.
	Wild register	8-bit × 2

Note:

The oscillation is 12.5 MHz unless another condition such as the main clock maximum speed, the clock cycle value, or conversion time is stated.

1.3 Differences between Models

This section describes the precautions to be taken when selecting a MB89202/F202RA series model.

■ Precautions when Selecting a Model

Table 1.3-1 Differences between Models

Package	MB89201	MB89F202/F202RA	MB89V202
DIP-32P-M06	×	○	○
FPT-32P-M03	×	○	○
FPT-64P-M03	○	×	×

- Current consumption
  - When operated at a low speed, the current consumption of a model with a flash is greater than that of a model with a mask ROM, though the current consumption in sleep or stop mode is the same.

Notes:

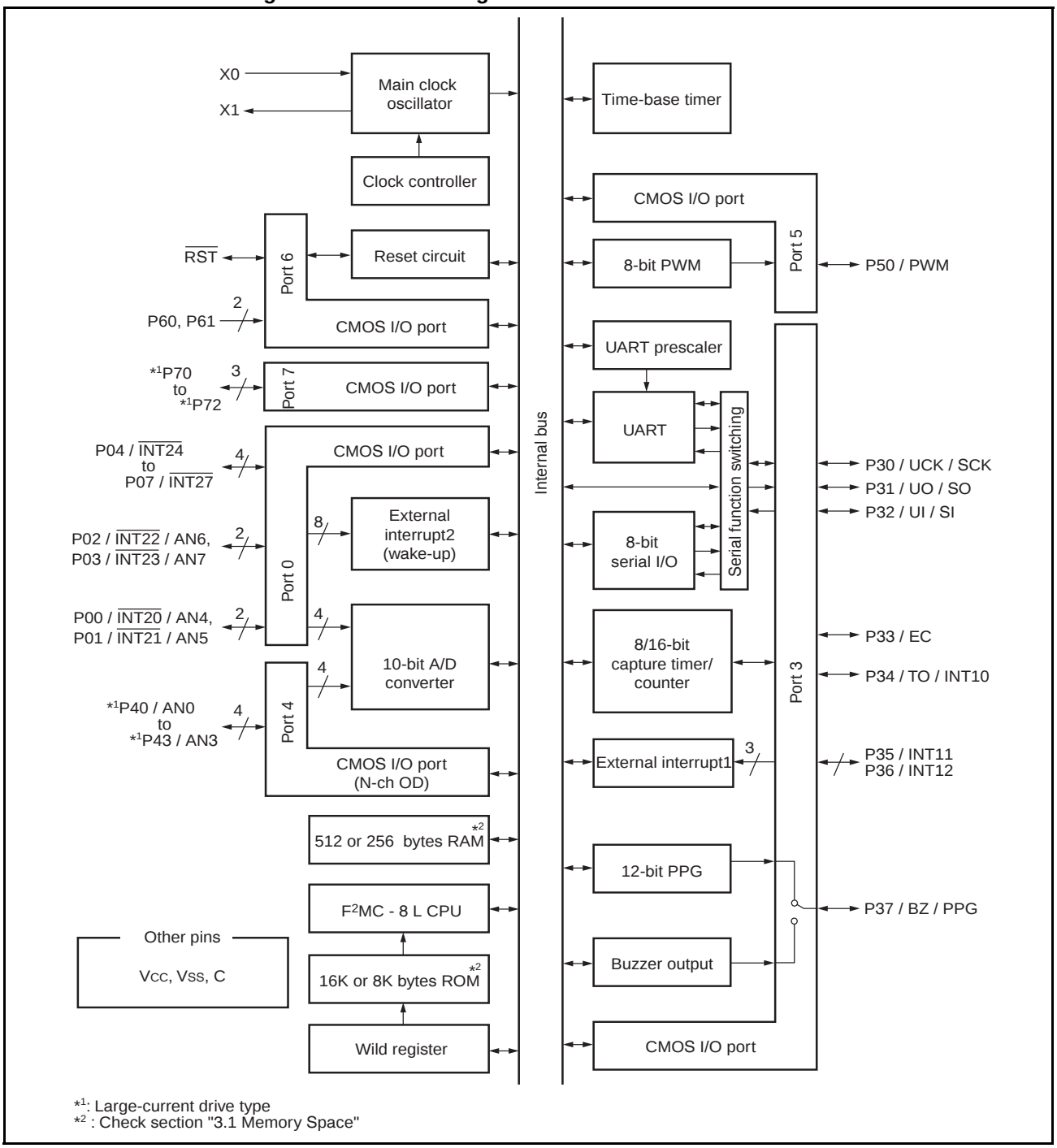
- For details on each package, see Section "1.6 Package Dimensions ".
- For details on current consumption and electrical characteristics of A/D converter, see the electrical characteristics in the Data Sheet.

1.4 Block Diagram of MB89202/F202RA Series

Figure 1.4-1 shows the block diagram of the MB89202/F202RA series.

■ Block Diagram of MB89202/F202RA Series

Figure 1.4-1 Block Diagram of MB89202/F202RA Series



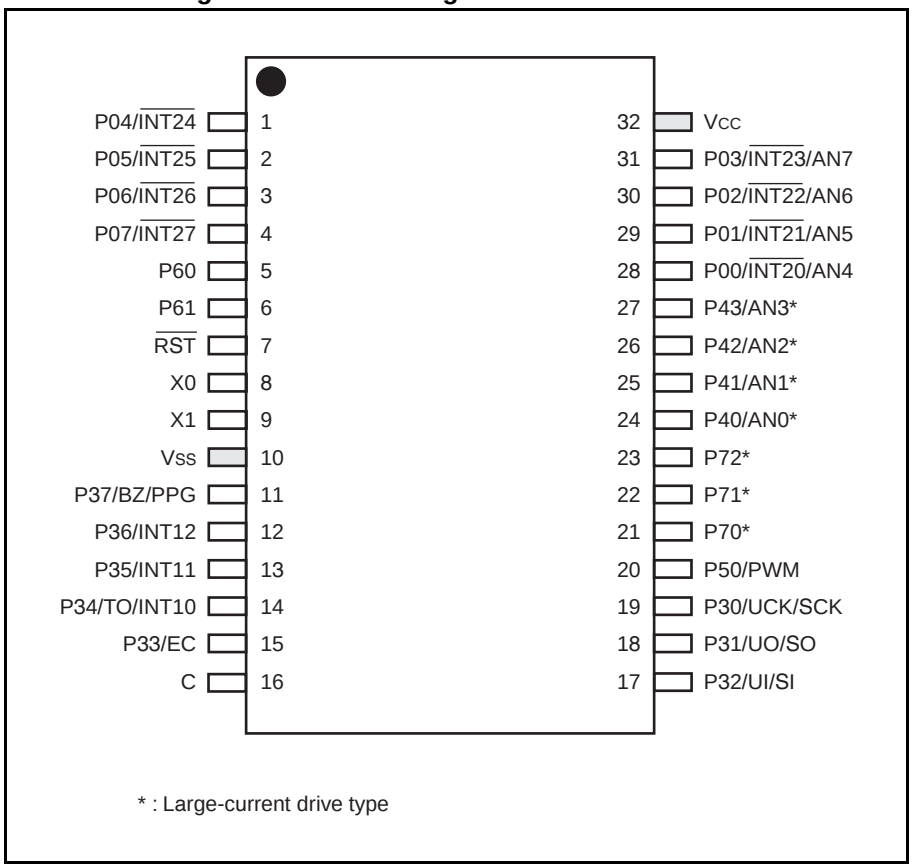
*1: Large-current drive type
*2 : Check section "3.1 Memory Space"

1.5 Pin Assignment

Figure 1.5-1 and Figure 1.5-2 show the pin assignment of the MB89202/F202RA series.

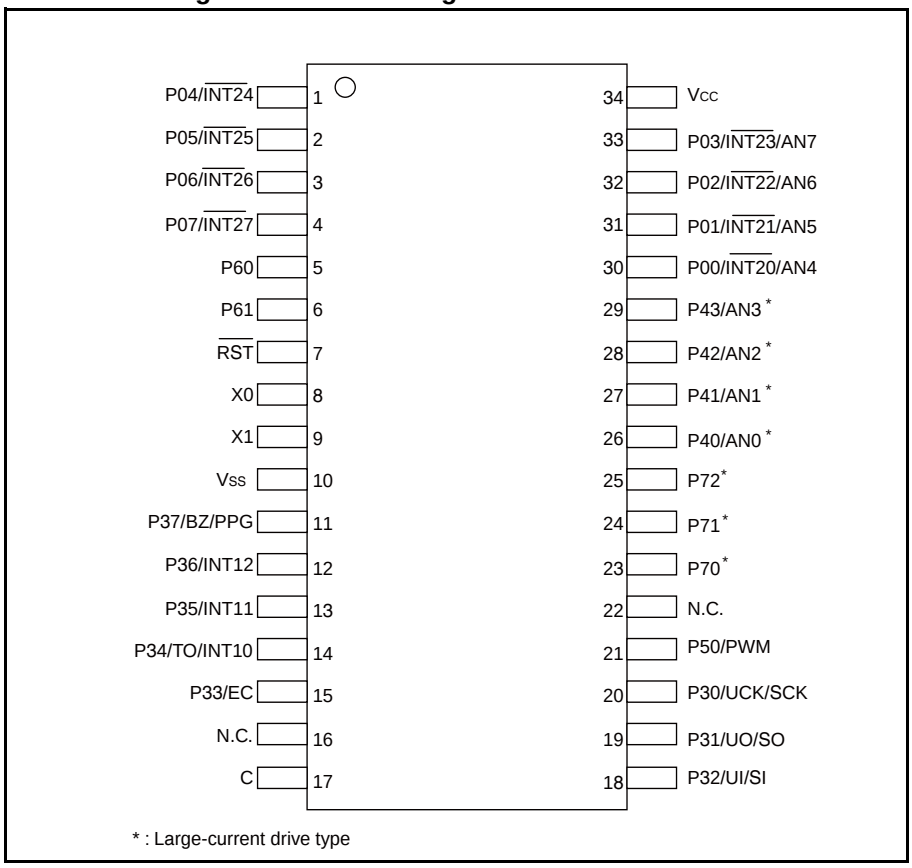
■ Pin Assignment of DIP-32P-M06

Figure 1.5-1 Pin Assignment of DIP-32P-M06



■ Pin Assignment of FPT-34P-M03

Figure 1.5-2 Pin Assignment of FPT-34P-M03



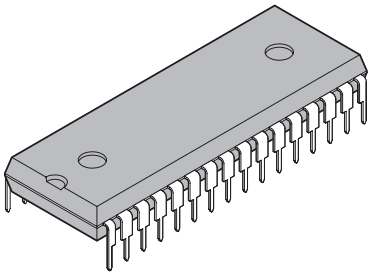
Note: N.C.: Do not use because it is connected internally.

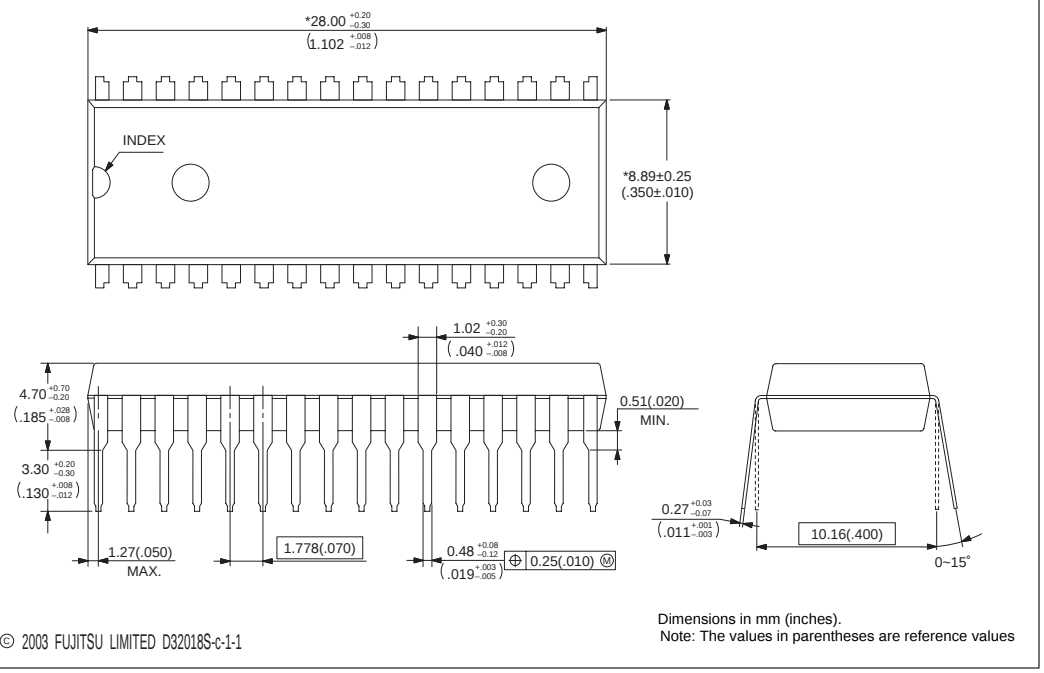
1.6 Package Dimensions

Two different packages are available for MB89202/F202RA series.
Figure 1.6-1 and Figure 1.6-2 show package dimensions.

■ Package Dimension of DIP-32P-M06

Figure 1.6-1 Package Dimension of DIP-32P-M06

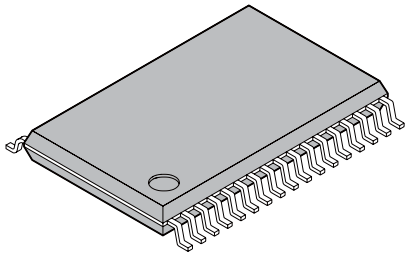
32-pin plastic SH-DIP  (DIP-32P-M06)	Lead pitch	1.778 mm
	Low space	10.16 mm
	Sealing method	Plastic mold

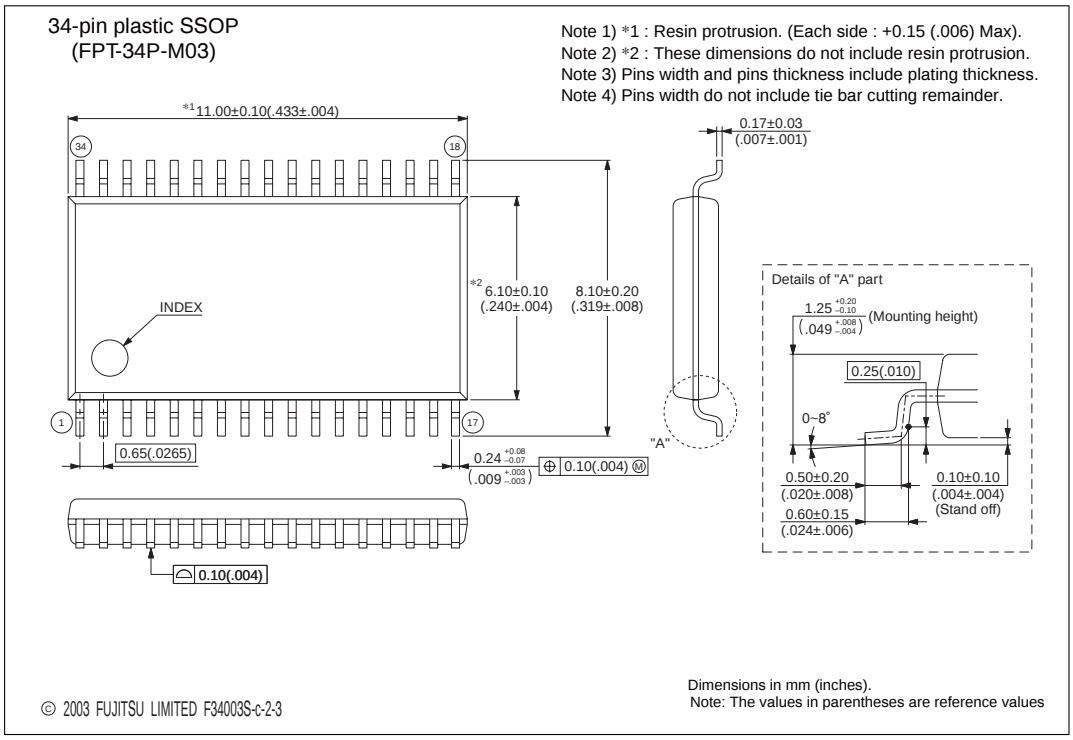


Please confirm the latest Package dimension by following URL.
http://edevice.fujitsu.com/fj/DATASHEET/ef-ovpklv.html

■ Package Dimension of FPT-34P-M03

Figure 1.6-2 Package Dimension of FPT-34P-M03

34-pin plastic SSOP  (FPT-34P-M03)	Lead pitch	0.65 mm
	Package width × package length	6.10 × 11.00 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.45 mm MAX
	Code (Reference)	P-SSOP34-6.1×11-0.65



Please confirm the latest Package dimension by following URL.
http://edevice.fujitsu.com/fj/DATASHEET/ef-ovpklv.html

1.7 Pin Functions Description

Table 1.7-1 describes the I/O pins and functions.
The letters in the circuit type column shown in Table 1.7-1 correspond to the letters in the Circuit Type column shown in Table 1.8-1 .

■ Pin Functions Description

Table 1.7-1 Pin Functions Description (1/2)

Pin No.		Pin name	Circuit type	Function
SHDIP32*1	SSOP34*2			
8	8	X0	A	Pins for connecting the crystal for the main clock. To use an external clock, input the signal to X0 and leave X1 open.
9	9	X1		
5, 6	5, 6	P60, P61	H / E	General-purpose CMOS input port.
7	7	$\overline{RST}$	C	Reset I/O pin. This pin serves as an N-ch open-drain reset output and a reset input as well. The reset is a hysteresis input. It outputs the "L" signal in response to an internal reset request. Also, it initializes the internal circuit upon input of the "L" signal.
28, 29	30, 31	P00/ $\overline{INT20}$/ AN4, P01/ $\overline{INT21}$/ AN5	G	General-purpose CMOS I/O ports. These pins also serve as an input (wake-up input) of external interrupt 2 or as an 10-bit A/D converter analog input. The input of external interrupt 2 is a hysteresis input.
30, 31	32, 33	P02/ $\overline{INT22}$/ AN6, P03/ $\overline{INT23}$/ AN7	G	General-purpose CMOS I/O ports. These pins also serve as an input (wake-up input) of external interrupt 2 or as an 10-bit A/D converter analog input. The input of external interrupt 2 is a hysteresis input.
1 to 4	1 to 4	P04/ $\overline{INT24}$ to P07/ $\overline{INT27}$	D	General-purpose CMOS I/O ports. These pins also serve as an input (wake-up input) of external interrupt 2. The input of external interrupt 2 is a hysteresis input.
19	20	P30/ UCK/ SCK	B	General-purpose CMOS I/O ports. This pin also serves as the clock I/O pin for the UART or 8-bit serial I/O. The resource is a hysteresis input.
18	19	P31/ UO/SO	E	General-purpose CMOS I/O ports. This pin also serves as the data output pin for the UART or 8-bit serial I/O.

Table 1.7-1 Pin Functions Description (2/2)

Pin No.		Pin name	Circuit type	Function
SHDIP32*1	SSOP34*2			
17	18	P32/UI/ SI	B	General-purpose CMOS I/O ports. This pin also serves as the data input pin for the UART or 8-bit serial I/O. The resource is a hysteresis input.
15	15	P33/EC	B	General-purpose CMOS I/O ports. This pin also serves as the external clock input pin for the 8/16-bit capture timer/counter. The resource is a hysteresis input.
14	14	P34/ TO/ INT10	B	General-purpose CMOS I/O ports. This pin also serves as the output pin for the 8/16-bit capture timer/ counter or as the input pin for external interrupt 1. The resource is a hysteresis input.
13	13	P35/ INT11	B	General-purpose CMOS I/O ports. These pins also serve as the input pin for external interrupt 1. The resource is a hysteresis input.
12	12	P36/ INT12	B	General-purpose CMOS I/O ports. These pins also serve as the input pin for external interrupt 1. The resource is a hysteresis input.
11	11	P37/ BZ/ PPG	E	General-purpose CMOS I/O ports. This pin also serves as the buzzer output pin or the 12-bit PPG output pin.
20	21	P50/ PWM	E	General-purpose CMOS I/O ports. This pin also serves as the 8-bit PWM timer output pin.
24 to 27	26 to 29	P40/ AN0 to P43/ AN3	F	General-purpose CMOS I/O ports. These pins can also be used as N-ch open-drain ports. These pins also serve as 10-bit A/D converter analog input pins.
21 to 23	23 to 25	P70 to P72	E	General-purpose CMOS I/O ports.
32	34	V_{CC}	--	Power supply pin
10	10	V_{SS}	--	Power (GND) pin
16	17	C	--	MB89F202/F202RA: Capacitance pin for regulating the power supply. Connect an external ceramic capacitor of about 0.1µF. MB89202: This pin is not internally connected. It is unnecessary to connect a capacitor.
--	16, 22	N.C.	--	Internally connected pins Be sure to leave it open.

*1 : DIP-32P-M06
*2 : FPT-34P-M03

1.8 I/O Circuit Types

Table 1.8-1 describes the I/O circuit types.
The letters in the circuit column shown in Table 1.8-1 correspond to the letters in the circuit type column shown in Table 1.7-1 .

■ I/O Circuit Types

Table 1.8-1 I/O Circuit Types (1/2)

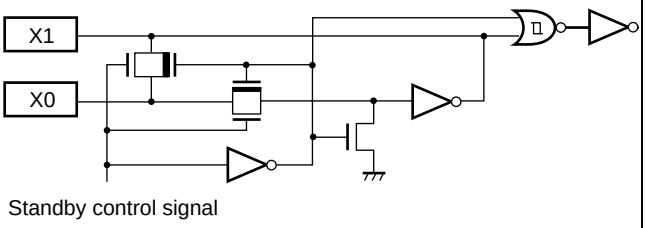
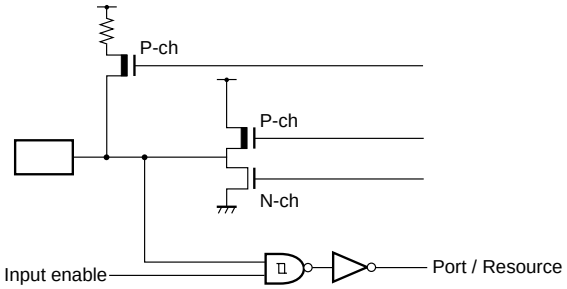
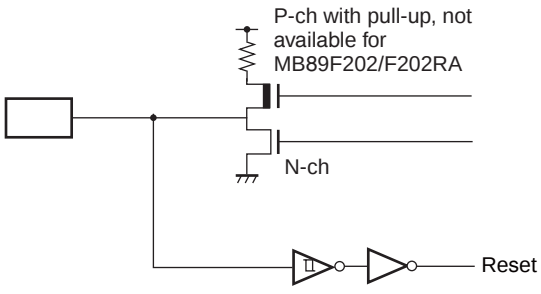
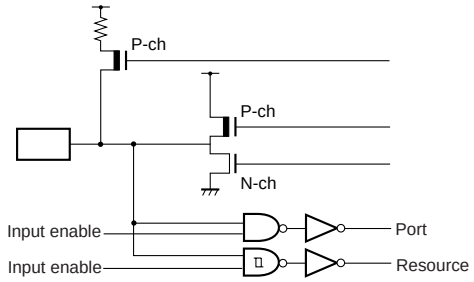
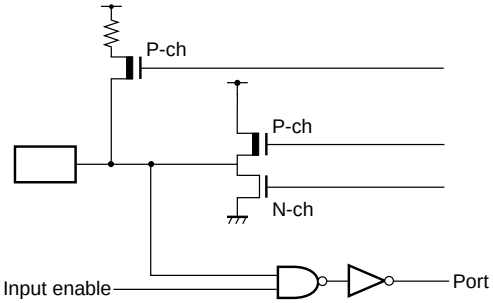
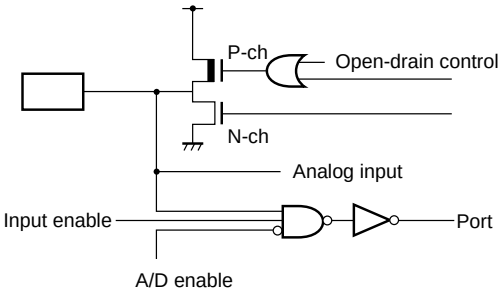
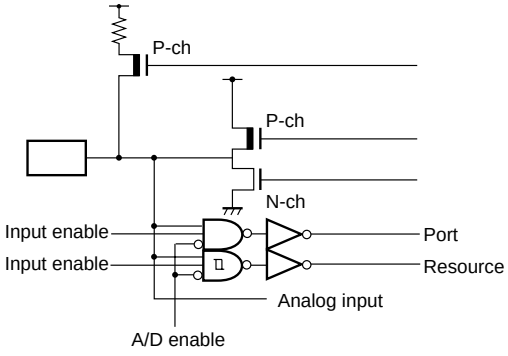
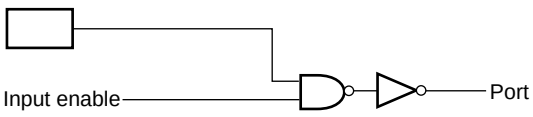
Types	Circuit	Remarks
A		At an oscillation feedback resistance of approximately 500 kΩ
B		CMOS output Hysteresis input Pull-up resistor optional
C		At an output pull-up resistor (P-ch) of approximately 50 kΩ/5.0 V (not available for MB89F202/F202RA) N-ch open-drain reset output Hysteresis input High voltage input tolerable in MB90F202RA

Table 1.8-1 I/O Circuit Types (2/2)

Types	Circuit	Remarks
D		CMOS output CMOS input Hysteresis input (Resource input) Pull-up resistor optional
E		CMOS output CMOS input Pull-up resistor optional P70 to P72 are large current drive type
F		CMOS output CMOS input Analog input N-ch open-drain output available P40 to P43 are large current drive type
G		CMOS output CMOS input Hysteresis input (Resource input) Analog input
H		CMOS input

CHAPTER 2

HANDLING DEVICES

This chapter describes the precautions to be taken when handling general-purpose one-chip microcontrollers.

2.1 Precautions on Handling Devices

2.1 Precautions on Handling Devices

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

■ Precautions on Handling Devices

- Ensure that the voltage does not exceed the maximum ratings. (Preventing latch-up)

  A latch-up may occur if a voltage higher than Vcc or lower than Vss is applied to input or output pins other than middle- or high-level resistant pins, or if voltage exceeding the rated value is applied between Vcc and Vss.

  When a latch-up occurs, the supply current increases rapidly, occasionally resulting in overheating. Therefore, ensure that the voltage does not exceed the maximum ratings when using the microcontrollers.

- Stabilize the supply voltage as much as possible

  Although the specified Vcc supply voltage operating range is assured, a sudden change in the supply voltage within the specified range may result in a malfunction.

  The following stabilization guidelines are recommended: The Vcc ripple (P-P value) at the supply frequency (50 Hz to 60 Hz) should be less than 10% of the typical Vcc value, and the transient fluctuation rate should be less than 0.1 V/ms at the time of momentary fluctuation when switching the power supply.

- Handling unused input pins

  Leaving unused input pins open may result in a malfunction or equipment damage due to a latch-up. Therefore, set these pins to pull-up or pull-down via resistors of 2 kΩ or higher.

- Handling the N.C. pins

  Ensure that the N.C. (internally connected) pins are opened before using.

- Precautions on using an external clock

  When an external clock is used, the oscillation stabilization wait time is also provided for power-on reset and stop mode release.

- Wild register function

  Because wild registers cannot be debugged on MB89V201, check operation on an actual MB89F202/F202RA.

- Program execution on RAM

  When MB89V201 is used, a program cannot be executed on RAM.

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

  If the reset pulse applied to the external reset pin ($\overline{RST}$) 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 ($\overline{RST}$).

- External pull-up for the External Reset Pin ($\overline{RST}$) of MB89F202/F202RA

  Internal pull-up control for $\overline{RST}$ is not available for MB89F202/F202RA. To ensure proper external reset control in MB89F202/F202RA, an external pull-up (recommend 100 kΩ) for $\overline{RST}$ pin must be required.

  For MB89F202RA only, high voltage must be applied to $\overline{RST}$ during flash memory program / erase. The typical high voltage is 10 V.

- Step-down circuit stabilization time

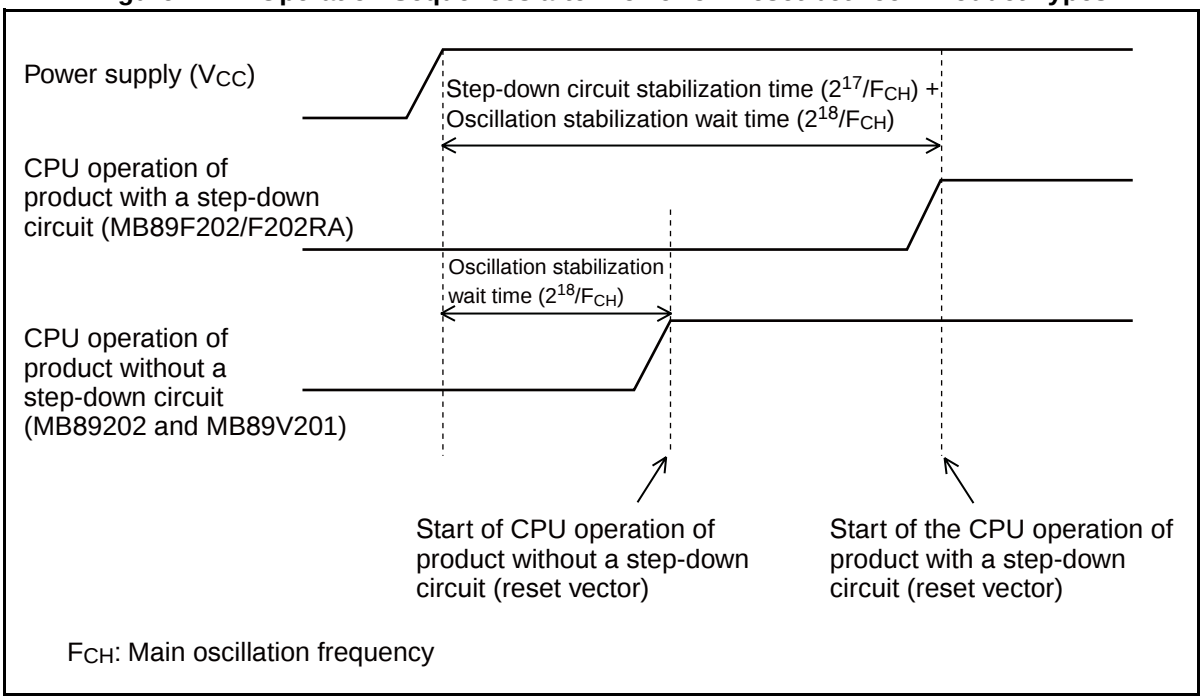
  The MB89202/F202RA series consists of the products listed in Table 2.1-1 "Pin Processing for the Products with and without a Step-down Circuit". The operation characteristic depends on whether a product contains a step-down circuit.

Table 2.1-1 Pin Processing for the Products with and without a Step-down Circuit

Product name	Operating voltage	Step-down circuit
MB89V201	2.7 V to 5.5 V	Not contained
MB89202	2.2 V to 5.5 V	Not contained
MB89F202/F202RA	3.5 V to 5.5 V	Contained

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 shows the sequence of operations after the power-on reset for each model.

Figure 2.1-1 Operation Sequences after Power-on Reset between Product Types



As shown in Figure 2.1-1 , the start of CPU operation of a product with a step-down circuit is slower than that of the product without a step-down circuit. This is because time is required for the step-down circuit to stabilize prior to normal operation of the step-down circuit.

CHAPTER 3

CPU

This chapter describes the functions and operations of the CPU.

3.1 Memory Space

The MB89202/F202RA series has 64-KB memory space that consists of the I/O area, RAM area, ROM area, and external area. Part of the memory space is applied for specific use such as general-purpose registers or a vector table.

■ Configuration of Memory Space

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

  The control registers and data registers for built-in peripheral functions are assigned.

  The I/O area is assigned as part of the memory space, thus access to the I/O area can be obtained in the same manner as access to memory. Also, direct addressing provides high-speed access.

- RAM area

  Static RAM is equipped as the internal data area.

  The size of internal RAM depends on the model.

  Direct addressing allows high-speed access to an area from 80_H to FF_H. (Some models restrict the usable range of the area.)

  100_H to $1FF_H$ can be used as the general-purpose register area.

  If a reset occurs while data is being written into RAM, the data being written cannot be guaranteed.

- ROM area

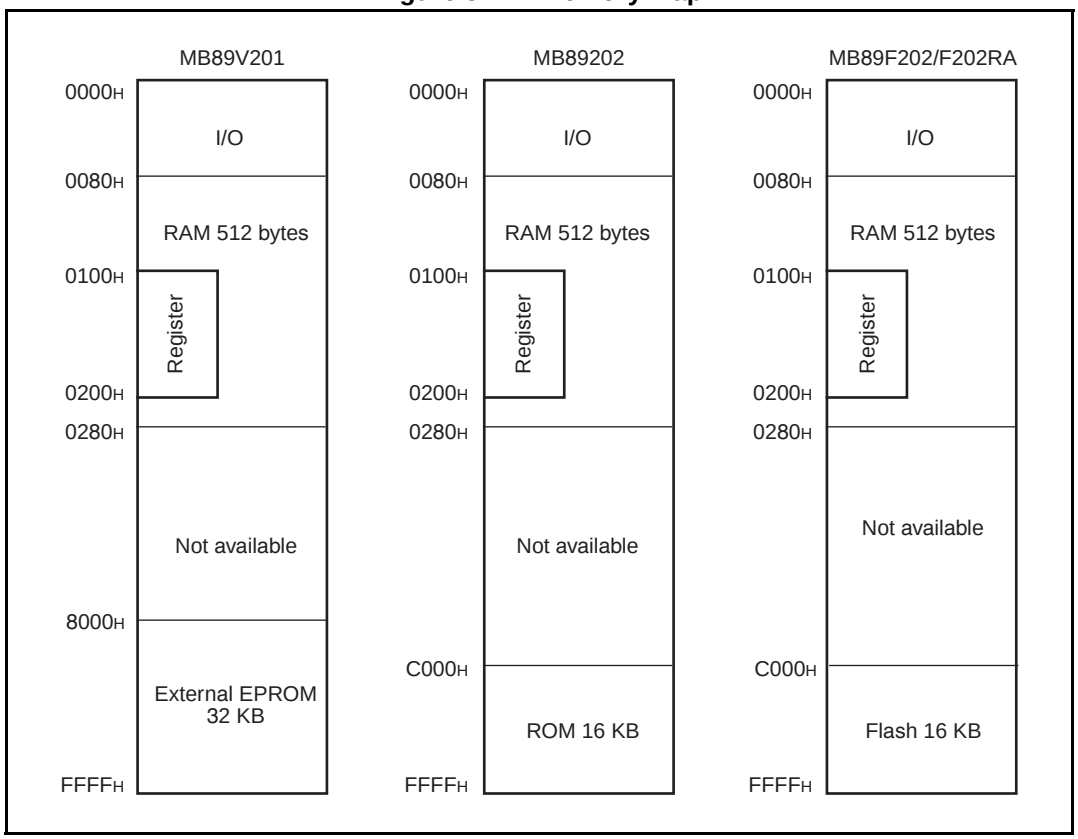
  ROM is equipped as the internal program area.

  The size of internal ROM depends on the model.

  $FFC0_H$ to $FFFF_H$ are usable as a vector table or another feature.

■ Memory Map

Figure 3.1-1 Memory Map



3.1.1 Specific-purpose 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 to $01FF_H$)

- 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: $FFC0_H$ to $FFFF_H$)

- 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 provides the reference addresses in the vector table that correspond to the vector instructions, interrupts, and reset.

For details, see Section "3.4 Interrupts ", Section "3.5 Reset ", and "CALLV #vct" in APPENDIX "B.2 Special Instructions ".

Table 3.1-1 Vector Table (1/2)

Vector call instruction	Address in the vector table	
	Upper digits	Lower digits
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	$FFCE_H$	$FFCF_H$
IRQF	$FFDC_H$	$FFDD_H$
IRQE	$FFDE_H$	$FFDF_H$
IRQD	$FFE0_H$	$FFE1_H$
IRQC	$FFE2_H$	$FFE3_H$

Table 3.1-1 Vector Table (2/2)

Vector call instruction	Address in the vector table	
	Upper digits	Lower digits
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	- *	$FFFD_H$
Reset vector	$FFFE_H$	$FFFF_H$

*: For MB89202 / MB89V201, $FFFC_H$ is prohibited. (Use "FF_H".)
For MB89F202/F202RA, write "01_H" to $FFFC_H$ to activate read protection, otherwise write "FF_H".

3.1.2 Location of 16-bit Data on Memory

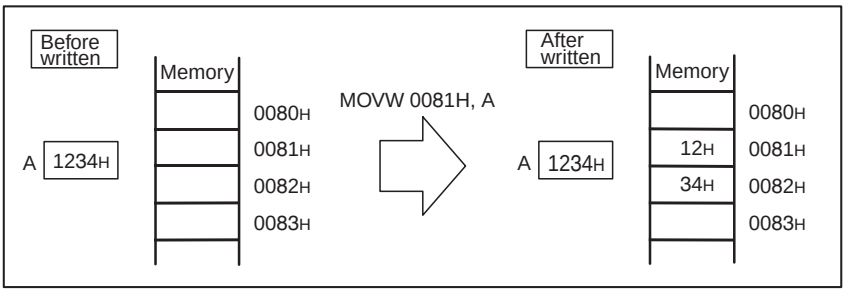
Upper digits of 16-bit data and stack data are stored in lower addresses on memory.

■ 16-bit Data Storage State on RAM

When 16-bit data is written into RAM, the upper byte of the data is stored with a lower address and the lower byte of the data is stored with the next address. 16-bit data is read in the same manner.

Figure 3.1-2 shows the location of 16-bit data on RAM.

Figure 3.1-2 Location of 16-bit Data on RAM



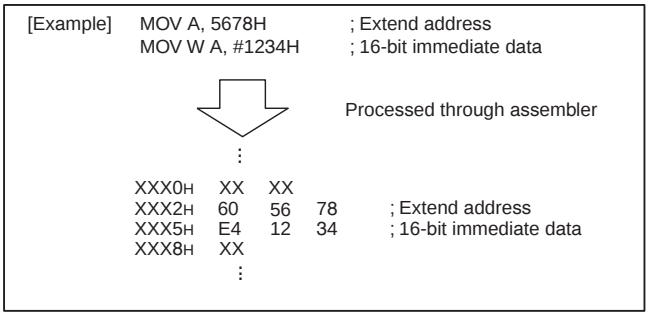
■ 16-bit Operand Storage State

When 16 bits are specified for operands in instructions, upper bytes are also stored in addresses close to operation codes (instructions) and lower bytes are stored in the following addresses.

Operands that indicate memory addresses and 16-bit immediate data are handled in the same manner as stated above.

Figure 3.1-3 shows the locations of 16-bit data in instructions.

Figure 3.1-3 Location of 16-bit Data in Instructions



■ 16-bit Data Storage State in Stack

The upper byte of data for a 16-bit register put in the stack due to an interrupt is also stored with a lower address.

3.2 Dedicated Register

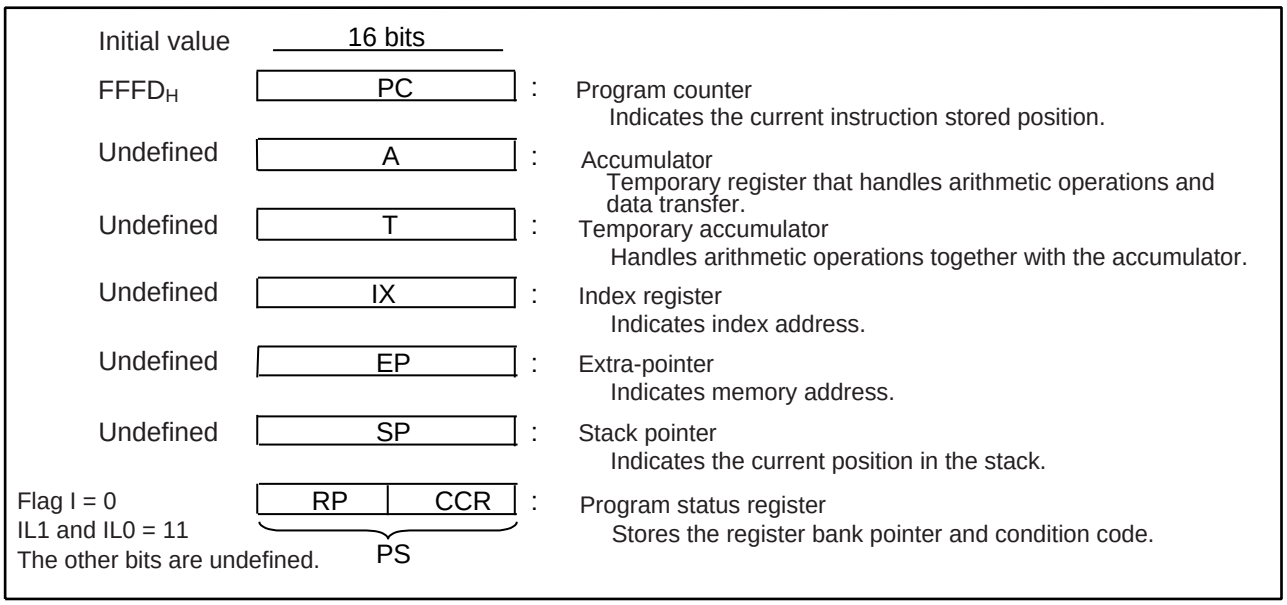
The dedicated register in the CPU consists of a program counter (PC), two arithmetic operation registers (A and T), three address pointers (IX, EP, and SP), and program status (PS) register. The size of each register is 16 bits.

■ Dedicated Register Configuration

The dedicated register in the CPU consists of seven 16-bit registers. Some registers allow only the lower 8 bits to be used.

Figure 3.2-1 shows the configuration of the dedicated register.

Figure 3.2-1 Configuration of Dedicated Register



■ Functions of the Dedicated Register

- Program counter (PC)

  The size of the program counter is 16 bits. It indicates the memory address at which the CPU is currently handling an instruction. The program counter is updated with an instruction executed, interrupt, or reset. The initial value specified after the reset operation is the mode data read address (FFFD$_H$).

- Accumulator (A)

  The accumulator is a 16-bit arithmetic operation register. It handles arithmetic operations or data transfer using data on memory or data in another register such as temporary accumulator (T). The accumulator allows data in it to be used as a word (16 bits) or bytes (8 bits). When arithmetic operations or data transfer is handled in the unit of a byte, only the lower 8 bits (AL) of the accumulator are used; the upper 8 bits (AH) remain unchanged. The initial value specified after the reset operation is undefined.

- Temporary Accumulator (T)

  The temporary accumulator is an auxiliary 16-bit arithmetic operation register. It handles arithmetic operations using data in the accumulator (A). When arithmetic operations in the accumulator (A) are handled in word units (16 bits), data in the temporary accumulator is handled in word units. Otherwise, it is handled in byte units (8 bits). When arithmetic operations are handled in byte units, only the lower 8 bits (TL) in the temporary accumulator are used; the upper 8 bits (TH) are not used.

  When an MOV instruction is used to transfer data into the accumulator (A), data stored in the accumulator is automatically transferred to the temporary accumulator before it is transferred. For data transfer in byte units, the upper 8 bits of the temporary accumulator (TH) does not change. The initial value of the temporary accumulator specified after the reset operation is undefined.

- Index register (IX)

  The index register is a 16-bit register that stores an index address. The index register is used together with a 1-byte offset (-128 to +127). It generates a memory address for accessing data by adding a sign-extended offset to the index address. The initial value of the index register specified after the reset operation is undefined.

- Extra-pointer (EP)

  The extra-pointer is a 16-bit register. Data in the extra-pointer is handled as the memory address for accessing data. The initial value of the extra-pointer specified after the reset operation is undefined.

- Stack pointer (SP)

  The stack pointer is a 16-bit register that stores an address that is used to call an interrupt or subroutine, or to which a stack/recovery instruction makes a reference. While a program is being executed, the value of the stack pointer indicates the address of the latest data put in the stack. The initial value of the stack pointer specified after the reset operation is undefined.

- Program status (PS) register

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

  The lower 8 bits are the condition code register (CCR) that composes flags for indicating the CPU status. Because these 8-bit registers comprise the program status register, they cannot be accessed. (Only instructions MOVW A, PS and MOVW PS, A access the program status register.)

  Note:

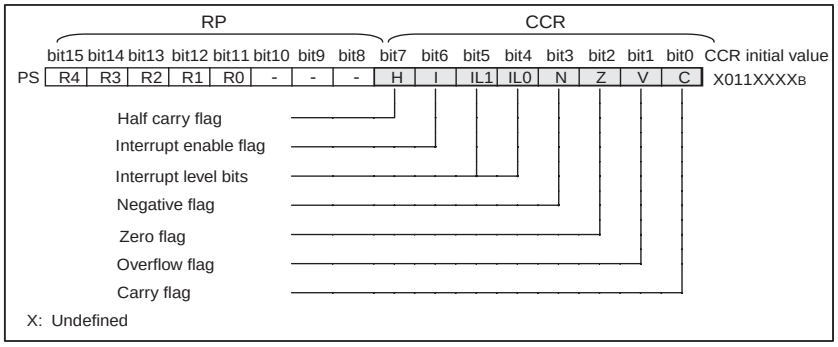
  For details on how to use the dedicated register, see the F^2MC-8L MB89600 Series Programming Manual.

3.2.1 Condition Code Register (CCR)

The condition code register (CCR) is the lower 8 bits of the program status register (PS). The condition code register consists of bits (C, V, Z, N, and H) for indicating the results of arithmetic operations or data to be transferred and control bits (I, IL1, and IL0) for controlling the acceptance of interrupt requests.

■ Configuration of the Condition Code Register (CCR)

Figure 3.2-2 Configuration of Condition Code Register



■ Bits for Indicating Arithmetic Operation Results

- Half carry flag (H)

  When a carry from bit3 to bit4 or a borrow from bit4 to bit3 occurs as a result of an arithmetic operation, the half carry flag is set to "1". Otherwise, the half carry flag is cleared with "0". The half carry flag is intended only for decimal adjustment instructions, and thus should not be used for operations other than addition or subtraction.

- Negative flag (N)

  When the highest bit becomes "1" as a result of an arithmetic operation, the negative flag is set to "1". When it becomes "0", it is cleared with "0".

- Zero flag (Z)

  When the result of an arithmetic operation is "0", the zero flag is set to "1". Otherwise, the zero flag is cleared with "0".

- Overflow flag (V)

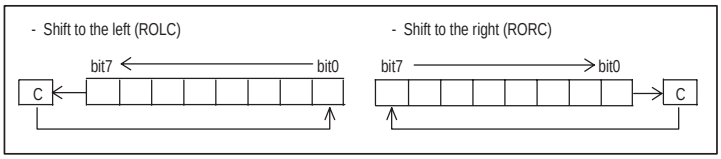
  When a complement on 2 overflow occurs as a result of an arithmetic operation, the overflow flag is set to "1". Otherwise, the overflow flag is cleared with "0".

- Carry flag (C)

  When a carry from bit7 or a borrow to bit7 occurs as a result of an arithmetic operation, the carry flag is set to "1". Otherwise, the carry flag is cleared with "0". The shift instruction causes the value to be shifted out.

Figure 3.2-3 shows how the shift commands change the carry flag.

Figure 3.2-3 Change of the Carrier Flag by the Shift Commands



Note:

The condition code register is part of the program status register (PS), and thus is not allowed to access only the condition code register.

It is uncommon to fetch and use only some of the flag bits directly. Normally, branch instructions (such as BNZ) or decimal adjustment instructions (such as DAA and DAS) use them indirectly. The initial values of these flags specified after the reset operation are undefined.

■ Bits for Controlling Acceptance of Interrupts

● Interrupt enable flag (I)

When this flag is "1", interrupts are allowed and the CPU accepts interrupts.

When this flag is "0", interrupts are prohibited and the CPU does not accept interrupts.

The initial value of the interrupt enable flag after the reset operation is "0".

Normally, the SETI instruction sets the interrupt enable flag to "1", and the CLRI instruction sets it to "0" to clear.

● Interrupt level bits (IL1 and IL0)

These bits indicate the level of an interrupt the CPU is accepting, then it is compared with the values in the interrupt level setting registers (ILR1 to 4) which is specified as the level of interrupt requests of peripheral functions (IRQ0 to IRQF).

When the interrupt enable flag is turned on (I = 1), and if an interrupt is requested with an interrupt level value lower than that of these bits, the CPU accepts the interrupt. Table 3.2-1 provides interrupt level intensities. The initial value of the interrupt level specified after the reset operation is 11_B.

Table 3.2-1 Interrupt Levels

IL1	IL0	Interrupt level	Intensity
0	0	1	High ↕ Low (no interrupts allowed)
0	1		
1	0	2	
1	1	3	

Note:

When the CPU is not handling an interrupt (handling the main program), the interrupt level bits (IL1 and IL0) are normally set to 11_B.

For details on interrupts, see Section "3.4 Interrupts ".

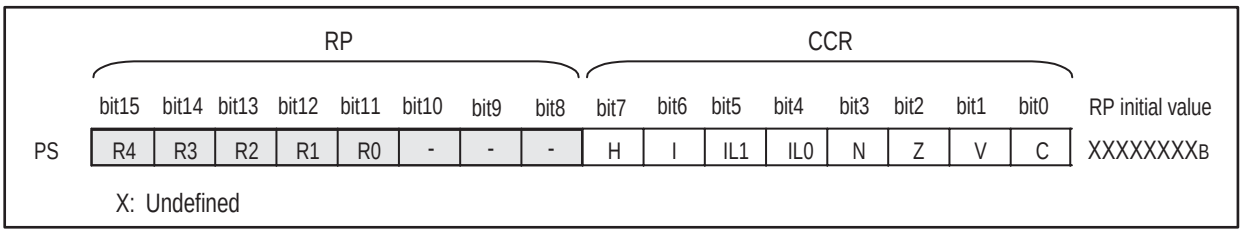
3.2.2 Register Bank Pointer (RP)

The register bank pointer (RP) is the upper 8 bits of the program status register (PS). The register bank pointer indicates the general-purpose register bank address being used, and the address is converted to the actual address in general-purpose register addressing.

■ Configuration of the Register Bank Pointer (RP)

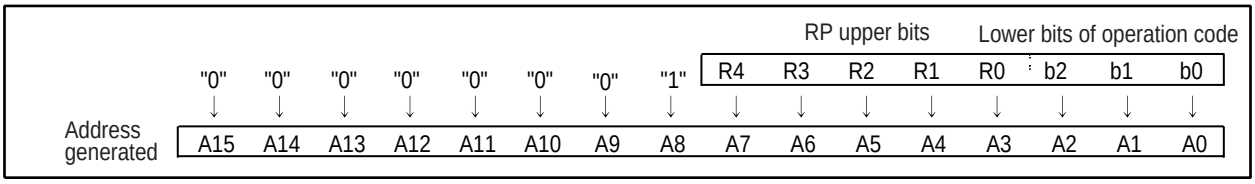
Figure 3.2-4 shows the configuration of the register bank pointer.

Figure 3.2-4 Configuration of Register Bank Pointer



The register bank pointer indicates the address of the register bank being used. Figure 3.2-5 shows the rule of conversion from the register bank pointer bits to the actual address.

Figure 3.2-5 Rule of Conversion from the RP Bits to the Actual Address



The register bank pointer specifies a memory block (register bank) used as a general-purpose register in the RAM area. There are 32 register banks. Setting a value (from 0 to 31) in the upper five bits of the register bank pointer specifies a register bank. One register bank contains eight 8-bit general-purpose registers that are selected with the lower 3 bits of an operation code.

The register bank pointer allows a range of 0100_H to $01FF_H$ (maximum) to be used as the general-purpose register area. However, some models restrict the usable range when only internal RAM is used. The initial value of the register bank pointer specified after the reset operation is undefined.

Note:

Be sure to set up the register bank pointer (RP) before using general-purpose registers.

The register bank pointer is part of the program status register (PS), and thus is not allowed to access only the register bank pointer.

3.3 General-Purpose Registers

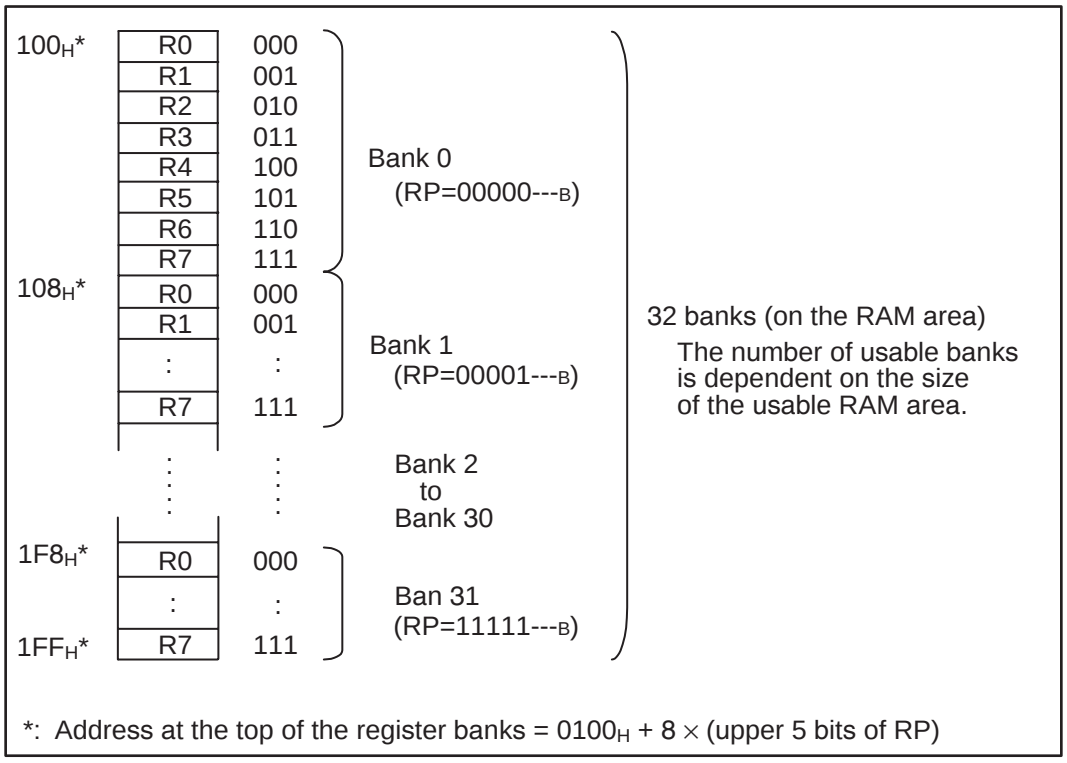
The general-purpose registers are memory blocks. Eight 8-bits comprise a bank.
The register bank pointer (RP) specifies a register bank.
Although up to 32 banks can be used, some banks can be expanded onto external RAM if the capacity of internal RAM is not sufficient for all 32 banks.
The general-purpose registers are effective for processing interrupts, vector calls, or subroutine calls.

■ Configuration of the General-purpose Registers

- Each general-purpose register consists of 8 bits. The general-purpose registers are placed in the register banks at the general-purpose register area (on RAM).
- One bank contains eight registers (R0 to R7), and up to 32 banks can be used. However, some models restrict the number of usable banks when only internal RAM is used.
- The register bank pointer (RP) specifies the register bank being used. The lower three bits of an operation code indicate general-purpose register 0 (R0) to general-purpose register 7 (R7).

Figure 3.3-1 shows the configuration of the register banks.

Figure 3.3-1 Configuration of Register Bank



For details on the general-purpose register area on each model, see Section "3.1.1 Specific-purpose Areas ".

■ Features of the General-purpose Registers

The general-purpose registers have the following features:

- High-speed access with short instructions (general-purpose register addressing)
- Register banks (in blocks) that allow data to be easily conserved and partitioned in the unit of function

The general-purpose registers allow specific register banks to be statically assigned with the interrupt processing routine or vector call (CALLV #0 to #7) processing routine. For example, it can be used such that the fourth register bank is always used for the second interrupt.

For interrupts, unless data in a specific register bank that corresponds to an interrupt processing is incorrectly overwritten by another routine, simply specifying the specific register bank at the beginning of the interrupt processing routine stores the data contained in the general-purpose registers before interruption. This feature allows data in general-purpose registers to avoid being put in the stack and allows interrupts to be handled efficiently at high speed.

For subroutine calls, in addition to conservation of data in general-purpose registers, the register banks can implement re-entrant programs (reloadable programs with variable addresses unfixed) that are usually created using the index register (IX) or another function.

Note:

A program must be created so that the values of the interrupt level bits in the condition code register (CCR: IL1 and IL0) do not change when the register bank pointer (RP) is rewritten to specify a register bank in the interrupt processing routine.

3.4 Interrupts

The MB89202/F202RA series supports 12 interrupt request inputs corresponding to peripheral functions and allows an interrupt level to be assigned to each of the inputs. The interrupt controller compares levels of interrupts generated by peripheral functions when output of interrupt requests is allowed for peripheral functions. The CPU performs the interrupt operation according to its interrupt acceptance settings. The CPU cancels standby mode on reception of an interrupt request, then returns to the interrupt operation or normal operation.

■ Interrupt Requests from Peripheral Functions

Table 3.4-1 lists the interrupt requests that correspond to peripheral functions. When the CPU accepts an interrupt, the CPU takes a branch to the interrupt processing routine using the address in the interrupt vector table corresponding to the interrupt request as the branch address.

The interrupt level setting registers (ILR1, 2, 3, and 4) allow one of four interrupt processing intensities to be assigned to each interrupt request.

Interrupt requests with levels equal to or less than that of an interrupt request being handled in the interrupt processing routine are usually handled after the current interrupt processing routine ends. If interrupt requests with the same assigned level are generated simultaneously, IRQ0 has priority.

Table 3.4-1 Interrupt Requests and Interrupt Vectors (1/2)

Interrupt request	Address in the vector table		Names of bits in the interrupt level setting registers	Priority at identical level (at simultaneous occurrence)
	Upper digits	Lower digits		
IRQ0 (External interrupt INT10)	$FFFA_H$	$FFFB_H$	L01, L00	High ↓ Low
IRQ1 (External interrupt INT11)	$FFF8_H$	$FFF9_H$	L11, L10	
IRQ2 (External interrupt INT12)	$FFF6_H$	$FFF7_H$	L21, L20	
IRQ3 (8/16-bit capture timer/counter's timer)	$FFF4_H$	$FFF5_H$	L31, L30	
IRQ4 (8/16-bit capture timer/counter's capture)	$FFF2_H$	$FFF3_H$	L41, L40	
IRQ5 (Transmission with UART)	$FFF0_H$	$FFF1_H$	L51, L50	
IRQ6 (Reception with UART)	$FFEE_H$	$FFEF_H$	L61, L60	
IRQ7 (Time-base timer)	$FFEC_H$	$FFED_H$	L71, L70	
IRQ8 (A/D converter)	$FFEA_H$	$FFEB_H$	L81, L80	
IRQ9 (8-bit PWM)	$FFE8_H$	$FFE9_H$	L91, L90	
IRQA (External interrupt 2)	$FFE6_H$	$FFE7_H$	LA1, LA0	

Table 3.4-1 Interrupt Requests and Interrupt Vectors (2/2)

Interrupt request	Address in the vector table		Names of bits in the interrupt level setting registers	Priority at identical level (at simultaneous occurrence)
	Upper digits	Lower digits		
IRQB (Flash interface)	$FFE4_H$	$FFE5_H$	LB1, LB0	High
IRQC (8-bit serial I/O)	$FFE2_H$	$FFE3_H$	LC1, LC0	↕
IRQD (Unused)	$FFE0_H$	$FFE1_H$	LD1, LD0	
IRQE (Unused)	$FFDE_H$	$FFDF_H$	LE1, LE0	
IRQF (Unused)	$FFDC_H$	$FFDD_H$	LF1, LF0	Low

3.4.1 Interrupt Level Setting Registers (ILR1 to ILR4)

For the interrupt level setting registers (ILR1, 2, 3, and 4), 16 two-bit data items corresponding to interrupt requests sent from peripheral functions are assigned. Interrupt levels can be specified in these 2-bits (interrupt level setting bits).

■ Configuration of the Interrupt Level Setting Registers (ILR1 to ILR4)

Figure 3.4-1 Configuration of Interrupt Level Setting Register

Register	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	(Initial value)
ILR1	007BH	L31	L30	L21	L20	L11	L10	L01	L00	1111 1111B
		(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	
ILR2	007CH	L71	L70	L61	L60	L51	L50	L41	L40	1111 1111B
		(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	
ILR3	007DH	LB1	LB0	LA1	LA0	L91	L90	L81	L80	1111 1111B
		(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	
ILR4	007EH	LF1	LF0	LE1	LE0	LD1	LD0	LC1	LC0	1111 1111B
		(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	

W: Write only

For each interrupt request, 2 bits of the interrupt level setting registers are assigned. The values specified in the interrupt level setting registers are the intensities for processing the interrupts (interrupt levels 1 to 3).

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

When interrupt level 3 is specified, the CPU does not accept interrupt requests.

Table 3.4-2 provides the relationship between interrupt level setting bits and interrupt levels.

Table 3.4-2 Relationship between Interrupt Level Setting Bits and Interrupt Levels

L01 to LF1	L00 to LF0	Requested interrupt level	Priority
0	0	1	High ↕ Low (no interrupt)
0	1		
1	0	2	
1	1	3	

Notes:

- When the main program is being executed, the interrupt level bits in the condition code register (CCR: IL1 and IL0) are normally set to 11_B.
- The ILR1 to ILR4 registers are write-only enabled, and thus the bit manipulation instructions (SETB and CLRB) cannot be used.

3.4.2 Steps in the Interrupt Operation

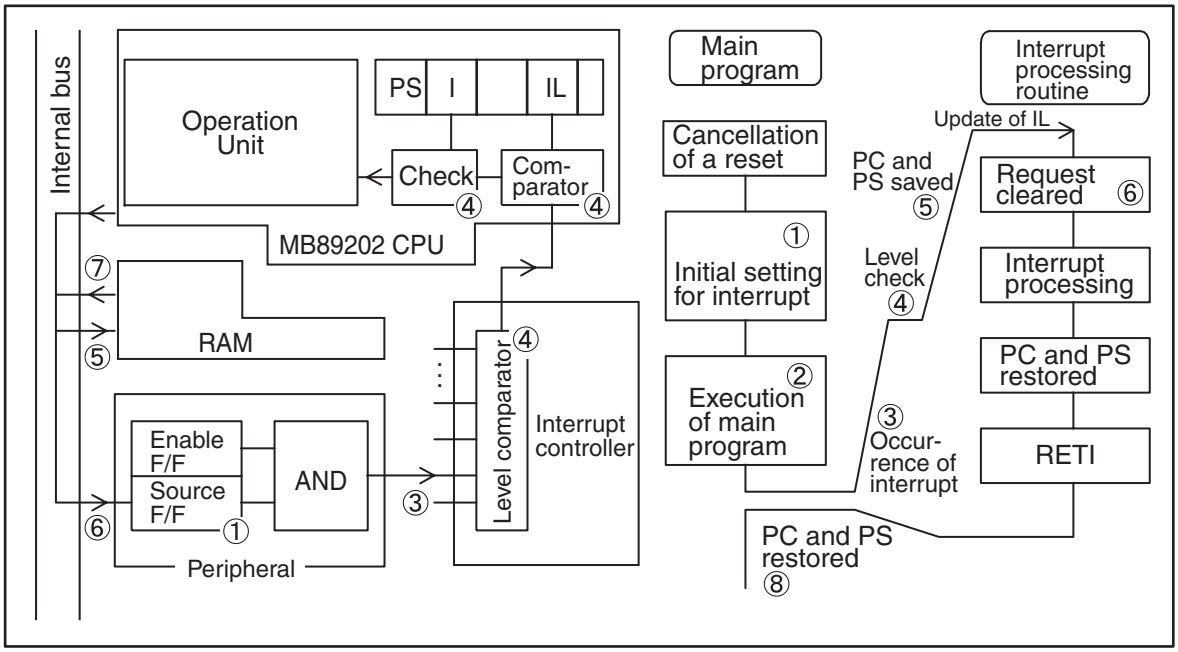
When an interrupt request is generated in a peripheral function, the interrupt controller notifies the CPU of its interrupt level. If the CPU can accept an interrupt, the CPU temporarily stops the program that is handling and starts the interrupt processing routine.

■ Steps in the Interrupt Operation

The steps for processing an interrupt are: occurrence of a source of an interrupt in a peripheral function, designation of the interrupt request flag bit (request F/F), check on the interrupt request enable bit (enable F/F), check on the interrupt level (ILR1, 2, 3, or 4, and CCR: IL1 and IL0), check on another request with the same level, and check on the interrupt enable flag (CCR: I).

Figure 3.4-2 shows the steps in the interrupt operation.

Figure 3.4-2 Steps in the Interrupt Operation



① After a reset, all interrupt requests are prohibited.

Initialize the peripheral functions that generate interrupts using a initialization program for peripheral functions, specify interrupt levels in the interrupt level setting registers (ILR1 to ILR4) concerned, then start up the peripheral functions.

Interrupt levels 1, 2, and 3 can be specified. Level 1 is the highest level, and level 2 is the second highest level. Level 3 prohibits interrupts from the peripheral functions to which it is assigned.

② Run the main program. (For a multiple-interrupt, run the interrupt processing routine.)

③ When a peripheral function generates a source of an interrupt, the interrupt request flag bit for peripheral function (request F/F) is set to "1". If the interrupt request enable bit for a peripheral function is turned on (enable F/F = 1) at that time, an interrupt request is output to the interrupt controller.

④ The interrupt controller is always monitoring interrupt requests from peripheral functions. The interrupt controller notifies the CPU of the highest interrupt level interrupt among levels corresponding to interrupt requests currently generated. If different requests are made with the same interrupt level, the interrupt controller also determines their priorities.

⑤ The CPU checks the value in the interrupt enable flag (CCR: I) when the priority of the interrupt level that is received is higher (the level value is lower) than the level specified in the interrupt level bits in the condition code register (CCR: IL1 and IL0). The CPU then accepts the interrupt when the enable flag is turned on (CCR: I = 1).

⑥ Put the values in the program counter (PC) and program status (PS) in the stack, fetch the start address of the interrupt processing routine from the interrupt vector table concerned, change the value of the interrupt level bits in the condition code register (CCR: IL1 and IL0) to the value of the interrupt level accepted, and then start the interrupt processing routine.

⑦ Finally, restore the values of the program counter (PC) and program status (PS) put into the stack with the RETI instruction, then execute an instruction following the instruction executed immediately before the interruption.

Standby mode (low-power consumption mode) is cancelled by an interrupt. For details, see Section "3.7 Standby Mode (Low-Power Consumption Mode) ".

Notes:

- An interrupt request flag bit for a peripheral function is not automatically cleared even if the interrupt request is accepted. Therefore, it is necessary to clear the bit using a program in the interrupt processing routine (by writing "0" into the interrupt request flag bit normally).
- Clearing an interrupt request flag bit at the beginning of the interrupt processing routine allows the peripheral function that generated the interrupt to re-generate an interrupt (set an interrupt request flag bit again) while the interrupt processing routine is being executed. However, the re-generated interrupt is normally accepted after the interrupt processing routine ends its current cycle.

3.4.3 Multiple Interrupts

Multiple interrupts are allowed by setting different levels into the interrupt level setting registers (ILR1 to ILR4) for multiple interrupt requests from peripheral functions.

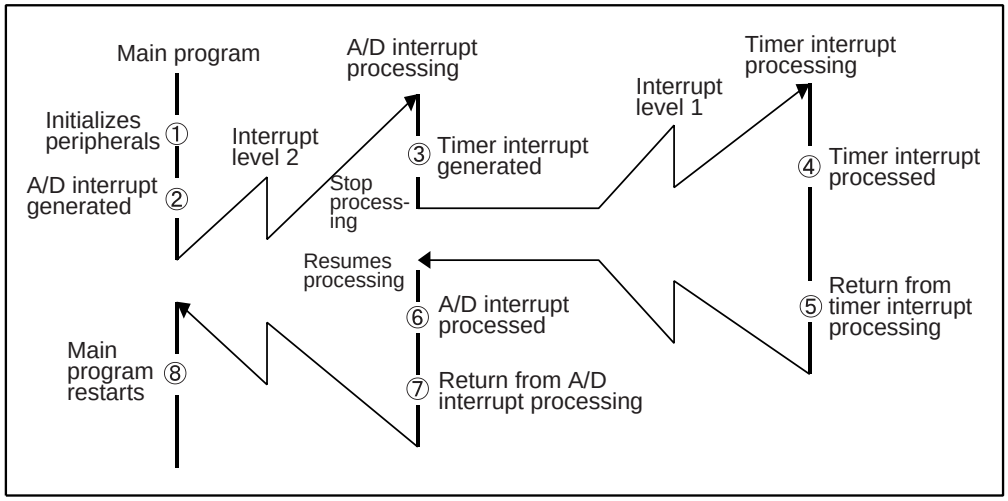
■ Multiple Interrupts

When an interrupt request with a higher interrupt level is generated while the interrupt processing routine is operating, the current interrupt processing cycle is stopped to accept the higher-level interrupt request. Interrupt levels 1, 2, and 3 can be specified. Level 3 prohibits the CPU from accepting interrupts.

● Example of multiple interrupts

As an example of multiple-interrupt processing, suppose a case in which a timer interrupt has precedence using the A/D interrupt, and the A/D interrupt level is set to level 2 and the timer interrupt level is set to level 1. Figure 3.4-3 shows the sequence performed when an external interrupt is generated while an A/D interrupt is being processed.

Figure 3.4-3 Example of Multiple Interrupts



- In the A/D interrupt processing, the interrupt level bits in the condition code register (CCR: IL1 and IL0) are set to the same value as the value in the interrupt level setting register corresponding to the A/D interrupt (ILR1, 2, 3, or 4) (i.e., 2 in this example). If an interrupt request with a higher interrupt level specified is generated (1 in this example), processing for the higher interrupt level is effected first.
- To temporarily prohibit multiple interrupts in the A/D interrupt processing, turn off the interrupt enable flag (CCR: I = 0) in the condition code register, or set 00_B to the interrupt level bits (IL1 and IL0).
- Executing the return instruction (RETI) after interrupt processing restores the values of the program counter (PC) and program status (PS) and ensures resumption of the interrupted program.
- The value in the condition code register (CCR) is returned to the value used before interruption when the program status (PS) value is restored.

3.4.4 Interrupt Processing Time

From when an interrupt request is generated to when control is transferred to the interrupt processing routine, both the time to quit the instruction being executed and the time to manage the interrupt (required to prepare interrupt processing) are required. The total time must be within 30 instruction cycles.

■ Interrupt Processing Time

From when an interrupt request is generated and accepted to when the interrupt processing routine starts, sufficient time is required to wait for an interrupt request sample and to manage the interrupt.

- Interrupt request sample wait time

  Generation of an interrupt request is checked by sampling an interrupt request at the last cycle of each instruction. Therefore, the CPU cannot identify an interrupt request while it is executing an instruction. The wait time becomes maximum when an interrupt request is generated immediately after the CPU executes the DIVU instruction (21 instruction cycles) with the longest instruction cycle.

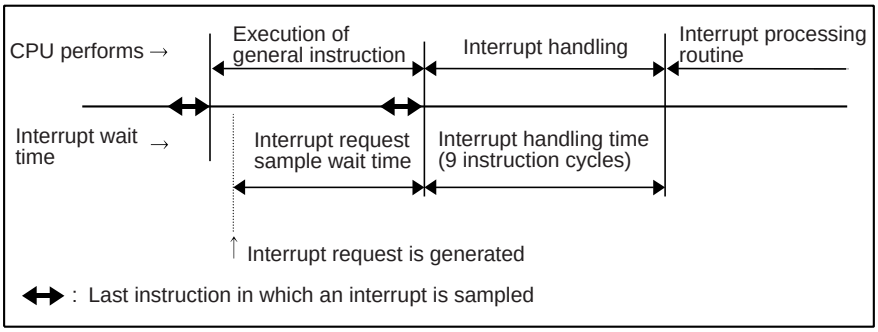
- Interrupt handling time

  After accepting an interrupt, the CPU needs 9 instruction cycles for interrupt processing preparation to:

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

  Figure 3.4-4 shows the interrupt processing time.

Figure 3.4-4 Interrupt Processing Time



When an interrupt request is generated immediately after the DIVU instruction having the longest instruction cycle (21 instruction cycles), 30 instruction cycles (21 instructions + 9 instructions) are required for the interrupt processing time. However, if the DIVU instruction and MULU instruction are not used in the program, a maximum of 15 (6 instructions + 9 instructions) instructions are required for the instruction processing time.

An instruction cycle is changed by clock speed switching (gears). For details, see Section "3.6 Clock ".

3.4.5 Stack Operation at Interrupt Processing

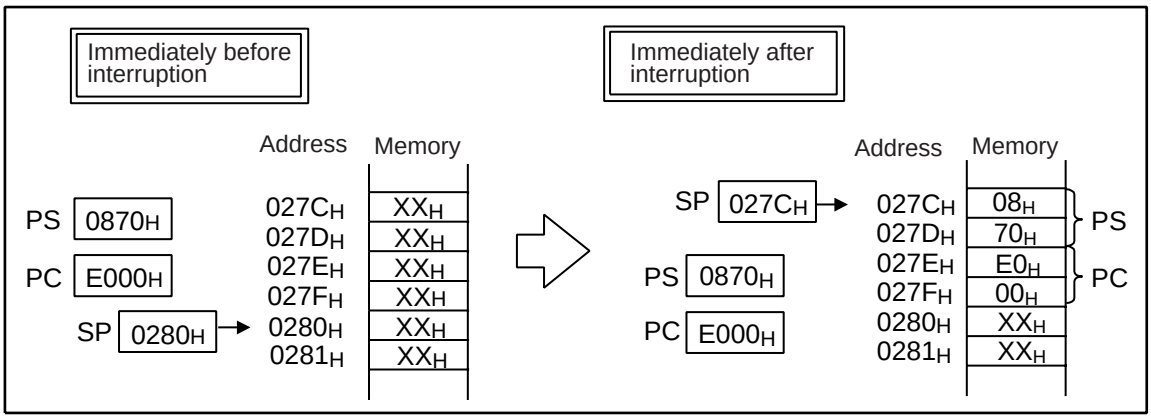
This section describes how values in registers are saved and restored at interrupt processing.

■ Stack Operation at the Beginning of Interrupt Processing

After accepting an interrupt, the CPU automatically saves the values in the program counter (PC) and program status (PS) in the stack.

Figure 3.4-5 shows the stack operation at the beginning of interrupt processing.

Figure 3.4-5 Stack Operation at the Beginning of Interrupt Processing



■ Stack Operation at the End of Interrupt Processing

When the return instruction (RETI) is executed at the end of interrupt processing, the values in the program status (PS) and the program counter (PC) are restored from the stack in that order (which is opposite to that at the beginning of interrupt processing). This operation restores the values in the PS and PC to those values used before interruption.

Note:

Values in the accumulator (A) and temporary accumulator (T) are not automatically saved in the stack. Therefore, save and restore the values using the PUSHW and POPW instructions.

3.4.6 Stack Area for Interrupt Processing

A stack area on RAM is used for interrupt processing. The value in the stack pointer (SP) is used as the start address of the stack area.

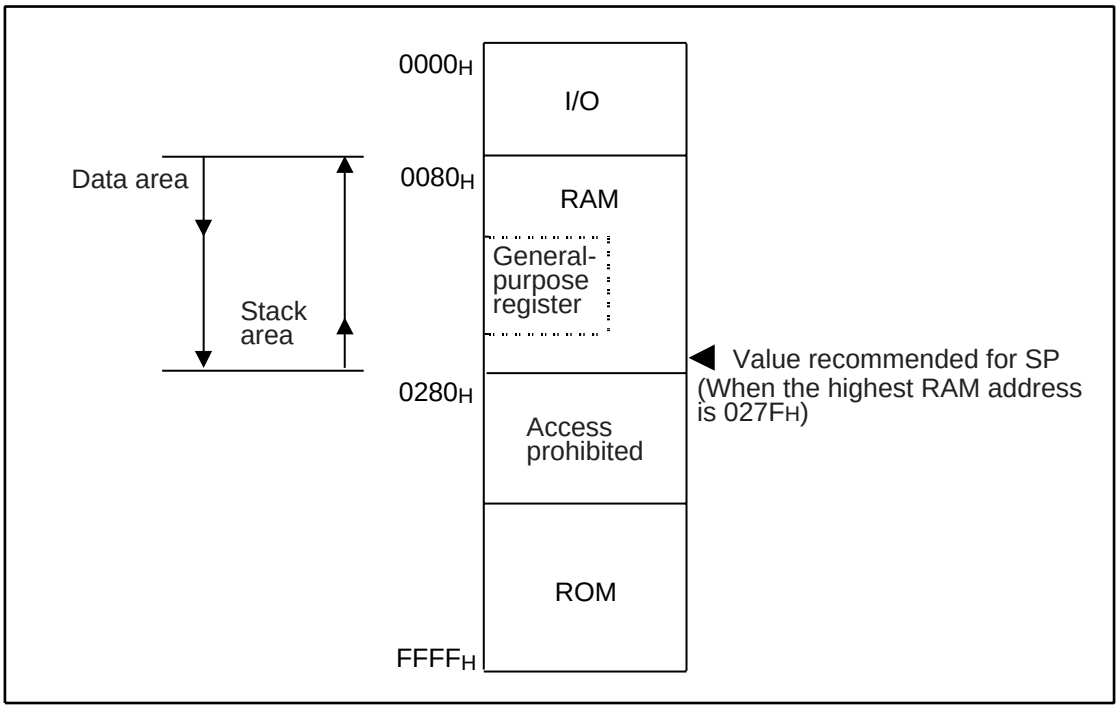
■ Stack Area for Interrupt Processing

The stack area is used to save/restore the value in the program counter (PC) when executing the subroutine call instruction (CALL) or vector call instruction (CALLV) or temporarily save and restore values in registers or other storage with the PUSHW and POPW instruction.

- Locate the stack area on RAM together with the data area.
- It is recommended that the initial settings be specified such that the stack pointer (SP) indicates the highest address of RAM and that the data area be set up from the lowest address of RAM.

Figure 3.4-6 is an example showing the stack area.

Figure 3.4-6 Stack Area for Interrupt Processing



Note:

For the stack area, interrupts, subroutine calls, or PUSHW instruction use addresses in descending order, and the return instructions (RETI and RET) or the POPW instruction releases addresses in the stack area in ascending order. When a lower address is used in the stack area due to multiple interrupts or subroutine calls, make arrangements so that the stack area does not overlap with the data area and general-purpose register area containing other data.

3.5 Reset

There are four sources of reset:

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

Oscillation stabilization wait time is not applied in some operating modes when a reset occurs or in some option settings.

■ Reset Sources

Table 3.5-1 Reset Sources

Reset source	Reset condition
External reset	The external reset pin is "L" level.
Software reset	"0" is written into the software reset bit in the standby control register (STBC: RST).
Watchdog reset	The watchdog timer overflows.
Power-on reset	Power is turned on.

● External reset

External reset occurs when "L" level is input to the external reset pin ($\overline{RST}$). When the reset pin becomes "H" level, the external reset is cancelled.

For external reset when power is turned on or in stop mode, the reset operation is performed after oscillation stabilization wait time is up or the external reset is cancelled.

The external reset pin functions as the reset output pin in accordance with option settings.

● Software reset

Software reset generates a 4-instruction cycle reset by writing "0" into the software reset bit in the standby control register (STBC: RST). Software reset does not wait until oscillation stabilization wait time has expired.

● Watchdog reset

Watchdog reset generates a 4-instruction cycle reset when no data is written into the watchdog control register (WDTC) within a specified time after the watchdog timer is activated. Watchdog reset does not wait until oscillation stabilization wait time is up.

- Power-on reset

  Power-on reset occurs when power is turned on. Power-on reset occurs after oscillation stabilization wait time has expired.

  Power-on reset requires an external reset circuit.

■ Reset Sources and Oscillation Stabilization Wait Time

Operations in oscillation stabilization wait time depend on the operating mode used when a reset occurs.

After a reset, active mode is set regardless of the operating mode applied before the reset (standby mode) and reset source. Therefore, if a reset occurs while oscillation is being stopped or within the oscillation stabilization wait time, the oscillation stabilization wait reset mode is set.

Software reset and watchdog reset do not apply oscillation stabilization wait time.

Table 3.5-2 shows the relationship between reset sources, oscillation stabilization wait time, and the reset operation (mode fetch).

Table 3.5-2 Relationship between the Reset Sources and Oscillation Stabilization Wait Time

Reset source	Operating mode	Reset operation and oscillation stabilization wait time
External reset*	When power is turned on or stop mode	The reset operation is performed when external reset is cancelled after oscillation stabilization wait time has expired.
Software reset and watchdog reset	Active mode	The reset operation is performed following the generation of a 4-instruction cycle reset.
Power-on reset	When power is turned on	The reset operation is performed after power is turned on and oscillation stabilization wait time has expired.

* External reset in active mode does not apply oscillation stabilization wait time. The reset operation is performed after cancellation of external reset.

3.5.1 Reset Flag Register (RSFR)

The reset flag register (RSFR) allows confirmation of the source for a generated reset.

■ Configuration of the Reset Flag Register (RSFR)

Figure 3.5-1 Configuration of Reset Flag Register (RSFR)

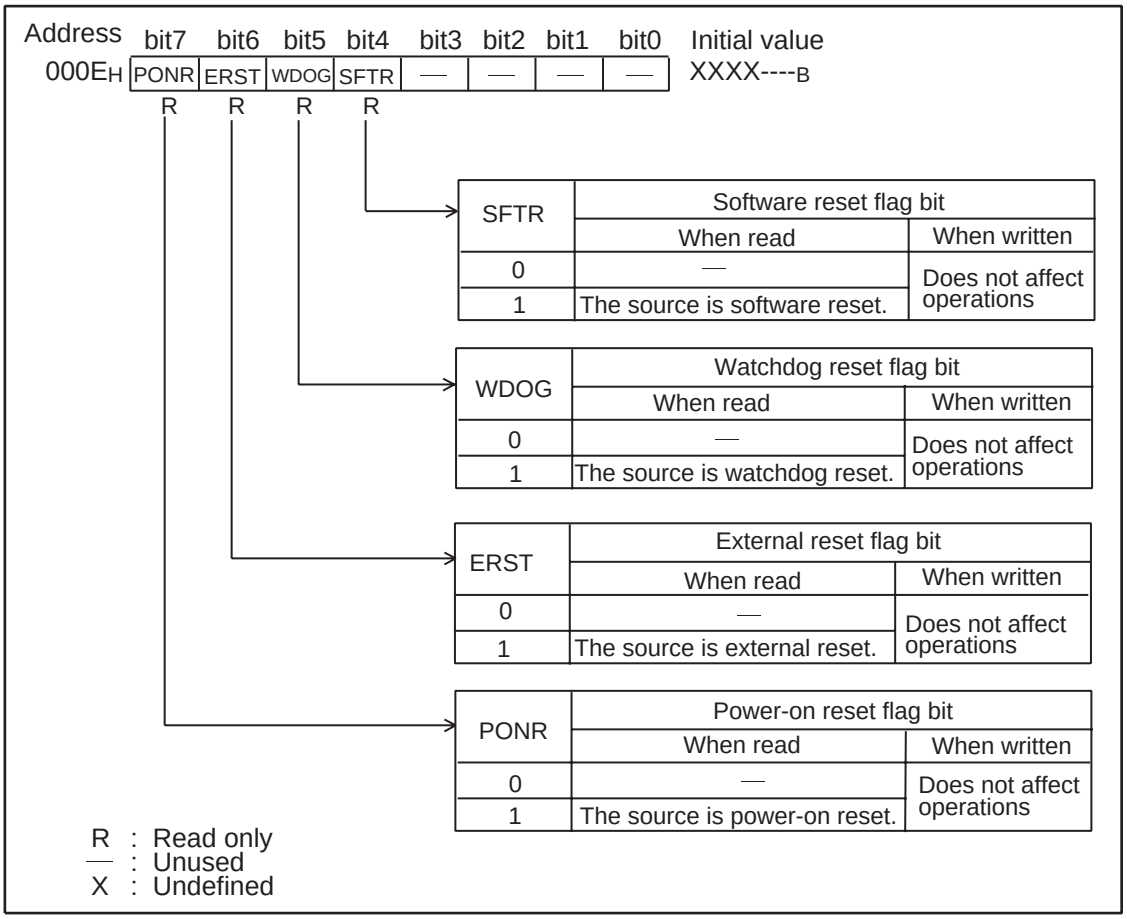


Table 3.5-3 Explanation of Functions of Each Bit in the Reset Flag Register (RSFR)

Bit name		Description
bit7	PONR: Power-on reset flag bit	"1" is set to this bit when power-on reset occurs. "1" is set to this bit after power is turned on. This bit is cleared with "0" after being read. Writing a value to this bit has no significance.
bit6	ERST: External reset flag bit	"1" is set to this bit when external reset occurs. "1" is set to this bit while other reset flags are maintained when all other reset flags have been set before the external reset flag is set. This bit is cleared with "0" after being read. Writing a value to this bit has no significance.
bit5	WDOG: Watchdog reset flag bit	"1" is set to this bit when watchdog reset occurs. "1" is set to this bit while other reset flags are maintained when all other reset flags have been set before the watchdog reset flag is set. This bit is cleared with "0" after being read. Writing a value to this bit has no significance.
bit4	SFTR: Software reset flag bit	"1" is set to this bit when software reset occurs. "1" is set to this bit while other reset flags are maintained when all other reset flags have been set before the software reset flag is set. This bit is cleared with "0" after being read. Writing a value to this bit has no significance.
bit3 to bit0	Unused bits	The values read out are undefined. Writing data to these bits does not affect operations.

Note:

A reset source flag is set when a reset source is generated. When the reset source flag register is read, all bits in the reset source flag register are cleared. Therefore, to determine the source of a reset, read this register using the initial value setting routine after the reset.

3.5.2 External Reset Pin

The external reset pin generates a reset by "L" level input. When an option setting for enabling reset output is selected, the "L" level signal is output depending on the internal reset source.

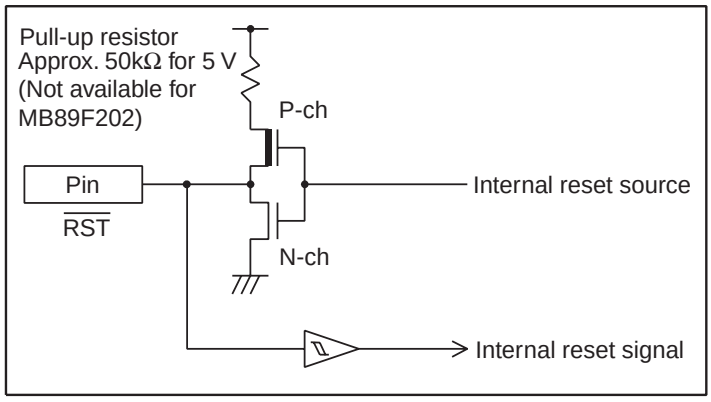
■ Block Diagram of External Reset Pin

The external reset pin ($\overline{RST}$) on models with supported reset output has hysteresis input and pull-up N-ch open drain output.

The external reset pin on models without supported reset output is used only as the pin dedicated to reset input.

Figure 3.5-2 is a block diagram of the external reset pin.

Figure 3.5-2 Block Diagram of External Reset Pin



■ Function of the External Reset Pin

The external reset pin ($\overline{RST}$) generates an internal reset signal by making use of "L" level input.

The $\overline{RST}$ outputs the "L" level signal according to the internal reset source and oscillation stabilization wait time applied following a reset. The internal reset source may be software reset, watchdog reset, or power-on reset.

Note:

External reset input is accepted asynchronously regardless of the internal clock.

Initialization of the internal circuits requires a clock. In particular, for operations with an external clock, the clock must be input when a reset signal is input.

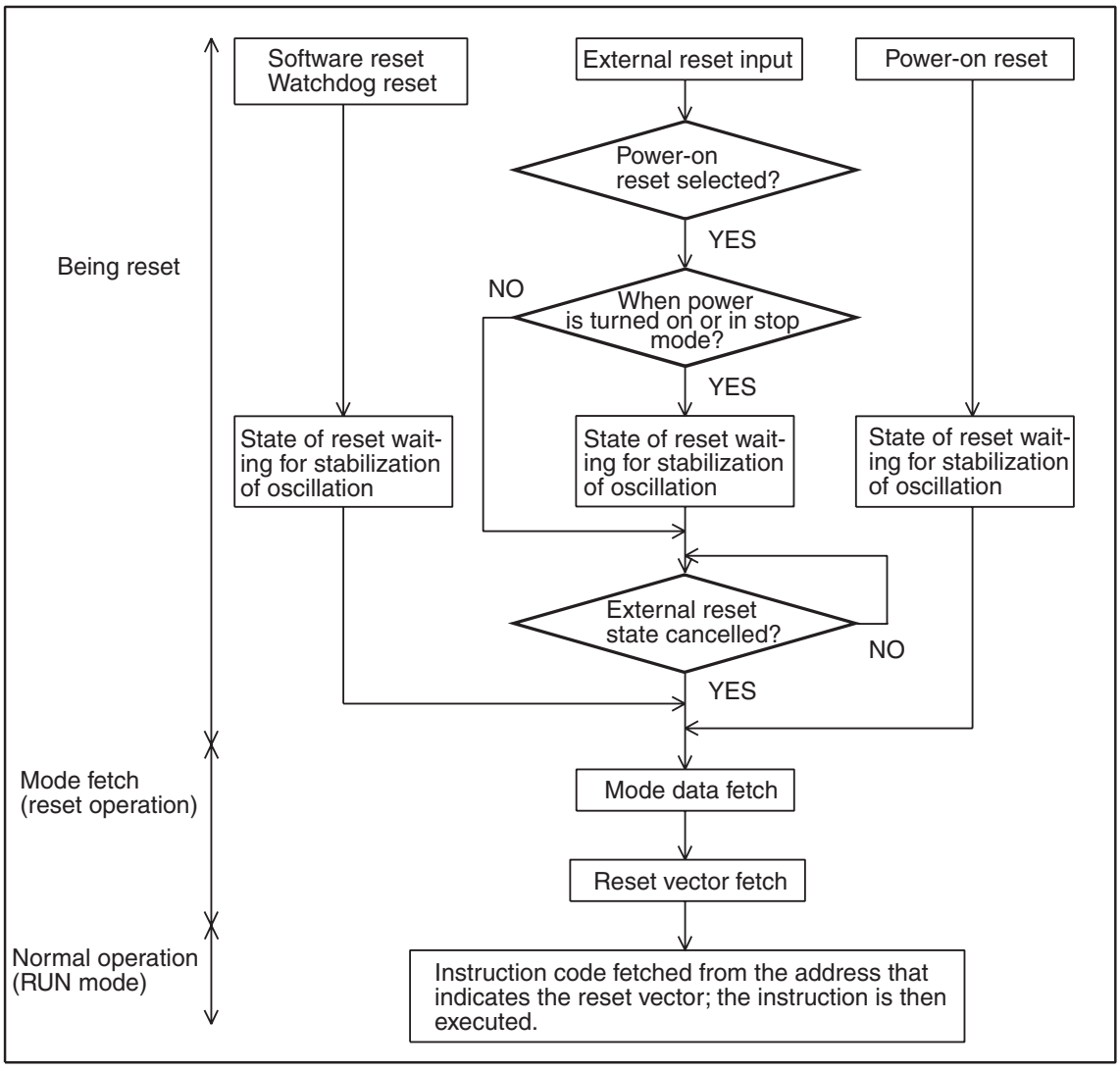
Internal pull-up control for $\overline{RST}$ is not available for MB89F202/F202RA. To ensure proper external reset control in MB89F202/F202RA, an external pull-up (recommend 100 kΩ) for $\overline{RST}$ pin must be required.

3.5.3 Reset Operation

The CPU reads the mode data (mode fetch) and reset vector from internal ROM according to the mode pin settings following the cancellation of a reset. For a return triggered by a reset when power is turned on and in stop mode, the CPU fetches the mode after oscillation stabilization wait time has expired. When a reset occurs, the contents in RAM cannot be guaranteed.

■ Overview of the Reset Operation

Figure 3.5-3 Reset Operation Flow



■ Mode Fetch

The CPU reads the mode data and reset vector from internal ROM following the cancellation of the reset.

- Mode data (address: $FFFD_H$)

  Set single-chip mode (00_H) to the mode data.

- Reset vector (address: $FFFE_H$ (highest)/$FFFF_H$ (lowest))

  Specify the address at which execution is to be started after the reset operation is completed. The CPU starts executing instructions from the specified address.

■ State of Reset Waiting for Stabilization of Oscillation

The CPU performs a reset operation for a reset when power is turned on or an external reset in stop mode when the oscillation stabilization wait time specified with option settings has expired. In this case, if the external reset input is not cancelled, the CPU performs the reset operation following cancellation of the external reset.

When an external clock is used, oscillation stabilization wait time is applied, and thus input of an external clock is required at a reset.

The time-base timer generates oscillation stabilization wait time.

■ Influence from a Reset of Contents in RAM

When reset conditions occur, the CPU stops handling the current instruction, then enters the reset state. The contents in RAM does not change even after a reset. However, if a reset occurs while 16-bit data is being written, the upper byte (only) is written; the lower byte may be unwritten. If a reset occurs immediately after, immediately before, or while data is written, the contents in the address to which data is written at that time is not guaranteed.

3.5.4 State of Each Pin at Reset

The state of each pin is initialized by a reset.

■ States of Pins during Reset

When a reset occurs, most I/O pins (resource pins) become Hi-Z, and the CPU reads the mode data from internal ROM.

■ States of Pins after the CPU Reads the Mode Data

Most of the I/O pins remain Hi-Z immediately after the CPU reads the mode data.

For pin states established by something other than a reset, see "APPENDIX E Pin State of the MB89202/F202RA Series " for details.

Note:

For pins that are Hi-Z when a reset source is generated, set up the devices connected with the pins such that they do not malfunction.

3.6 Clock

The clock generator includes the oscillation circuit. A high-speed clock is generated by connecting an external resonator for oscillation frequency. Alternatively, when the clock is supplied from an external source, a clock signal can be connected to the clock input pin.
The clock controller manages the speed and supply of the clock in active mode and standby mode.

■ Clock Supply Map

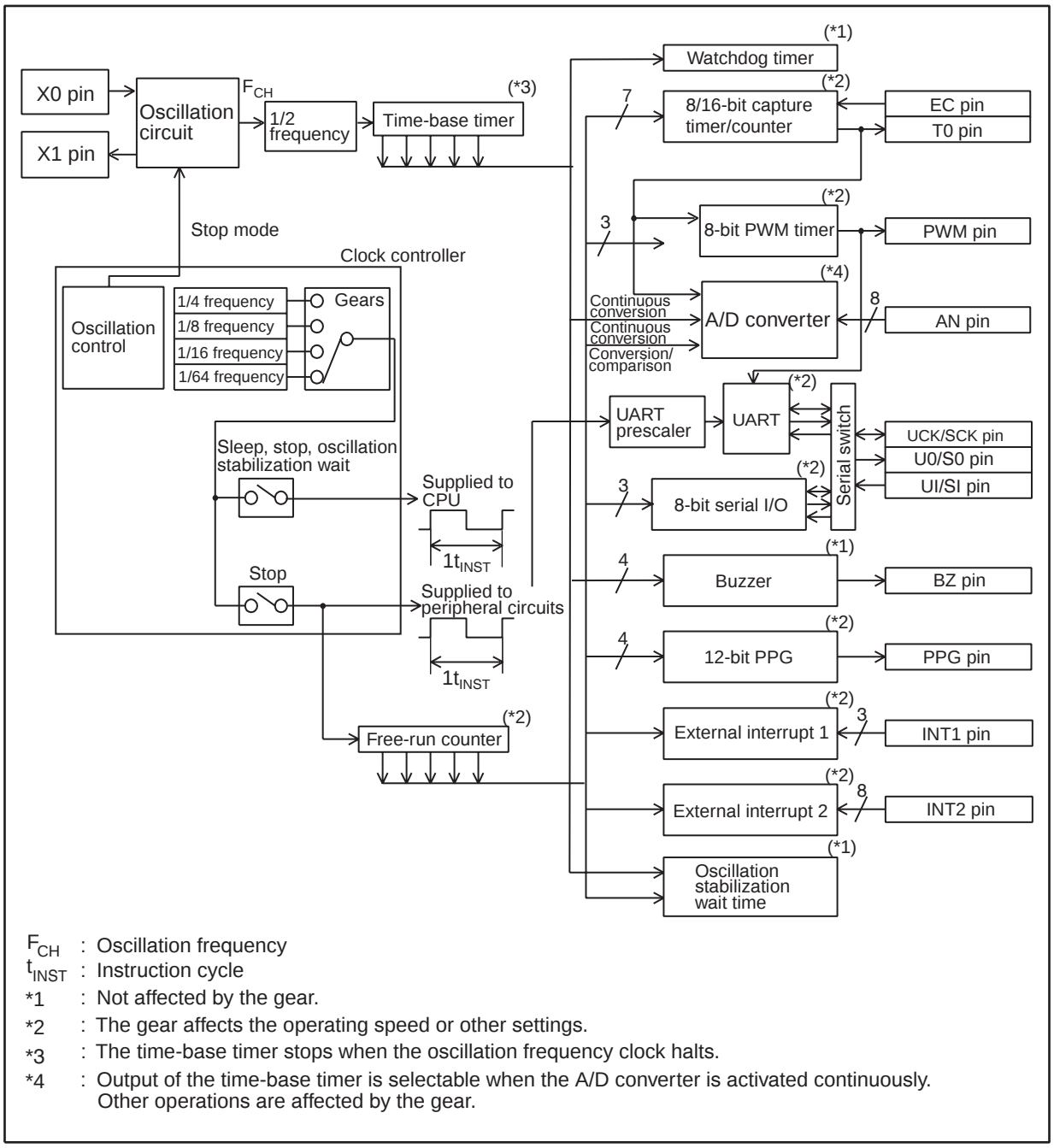
The clock controller manages oscillation of the clock and provision of the clock to the CPU and peripheral circuits (peripheral functions). Thus, the operating clock for the CPU or peripheral circuits is affected by clock speed switching (gears) and setting in standby mode (sleep/stop).

To peripheral functions, a divided frequency output of the free-run counter operating with the clock for peripheral circuits is provided.

However, the divided frequency output of the time-base timer operating with 1/2 frequency of the oscillation frequency is not affected by the gear.

Figure 3.6-1 shows the clock supply map.

Figure 3.6-1 Clock Supply Map



F_{CH} : Oscillation frequency
t_{INST} : Instruction cycle
*1 : Not affected by the gear.
*2 : The gear affects the operating speed or other settings.
*3 : The time-base timer stops when the oscillation frequency clock halts.
*4 : Output of the time-base timer is selectable when the A/D converter is activated continuously. Other operations are affected by the gear.

3.6.1 Clock Generator

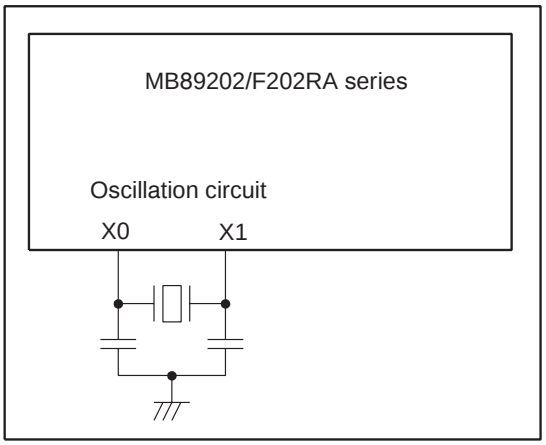
The clock generator enables oscillation in active mode and disables oscillation in stop mode.

■ Clock Generator

- For a crystal resonator or ceramic resonator

  Connect it as shown in Figure 3.6-2 .

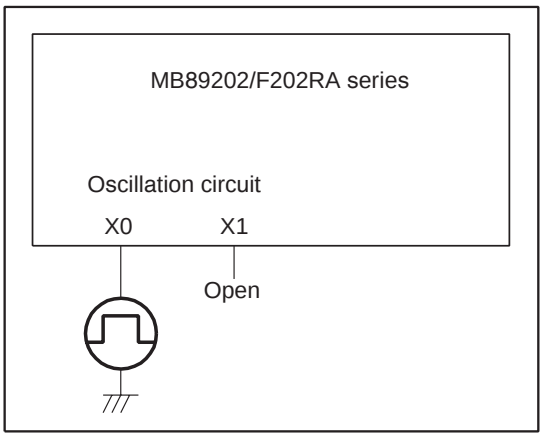
Figure 3.6-2 Example of Connecting a Crystal Resonator or Ceramic Resonator



- For an external clock

  Connect it to the X0 pin and open the X1 pin as shown in Figure 3.6-3 .

Figure 3.6-3 Example of Connecting an External Clock



3.6.2 Clock Controller

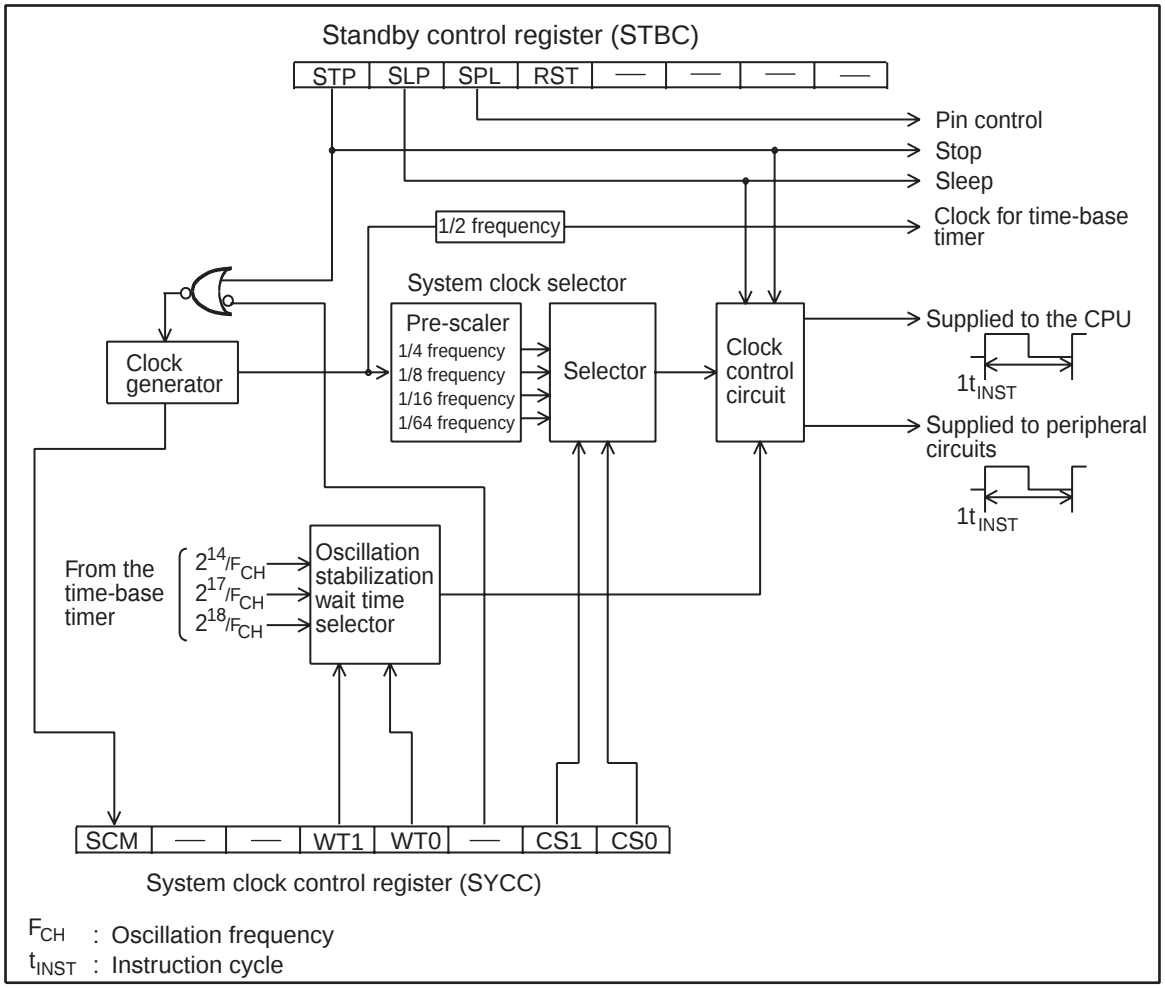
The clock controller consists of the following six blocks:

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

■ Block Diagram of Clock Controller

Figure 3.6-4 is a block diagram of the clock controller.

Figure 3.6-4 Block Diagram of Clock Controller



- Oscillator

  Oscillation circuit that halts oscillation in stop mode.

- System clock selector

  Selects one of four frequency-divided source clocks to be supplied to the clock control circuit.

- Clock controller

  Controls the operating clock supplied to the CPU and peripheral circuits according to the active (RUN) mode and standby mode (sleep, stop).

  It also stops supply of the clock to the CPU until the clock supply stop signal for the oscillation stabilization wait time selector is cancelled.

- Oscillation stabilization wait time selector

  Selects one of three oscillation stabilization wait time periods generated by the time-base timer according to the standby mode or a reset, then outputs the clock supply stop signal to the CPU by using the selected time period.

- System clock control register (SYCC)

  Selects the clock speed and oscillation stabilization wait time setting, then checks the clock state.

- Standby control register (STBC)

  Controls transition from active (RUN) mode to standby mode, pin state settings at stop mode, and software reset.

3.6.3 System Clock Control Register (SYCC)

The system clock control register (SYCC) manages clock settings such as selection of the clock speed and oscillation stabilization wait time.

■ Configuration of the System Clock Control Register (SYCC)

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

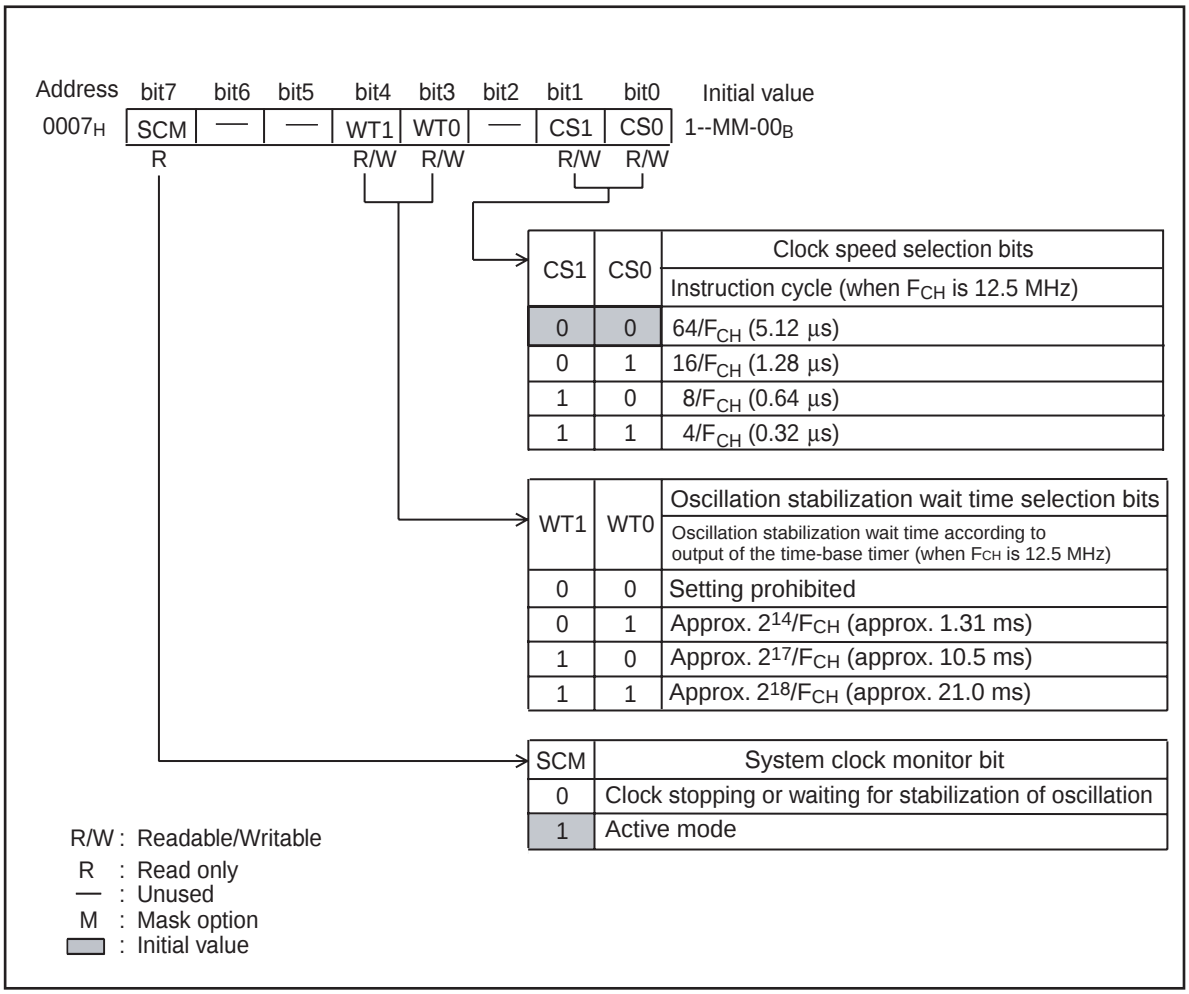


Table 3.6-1 Explanation of Functions of Each Bit in the System Clock Control Register (SYCC)

Bit name		Description
bit7	SCM: System clock monitor bit	Used to check the current clock mode. When this bit is 0, the clock is stopping or waiting for stabilization of oscillation. When this bit is 1, operations are performed in active mode. **Note:** This bit is read-only enabled. Writing a value to this bit does not affect operation.
bit6, bit5	Unused bits	Values in these bits are undefined when read. Writing values into these bits does not affect operation.
bit4, bit3	WT1, WT0: Oscillation stabilization wait time selection bits	Used to select an oscillation stabilization wait time setting. When external interrupt causes a return from stop mode to active mode, the oscillation stabilization wait time setting selected by these bits is applied. The initial values of these bits are determined by options. Therefore, when an oscillation stabilization wait time setting is to be applied for a reset, it is selected by options. **Note:** Change values in these bits after confirming that the clock is not waiting for stabilization of oscillation using the SCM bit.
bit2	Unused bit	This bit is always "1" when read. **Note:** Specify "1".
bit1, bit0	CS1, CS0: Clock speed selection bits	Used to select the clock speed in active mode. One of four operating clock speeds (gears) can be specified for the CPU and peripheral functions. However, these bits do not affect the operating clock for the time-base timer.

■ Instruction Cycle (t_{INST})

For instruction cycles (minimum instruction run time), a 1/4, 1/8, 1/16, or 1/64 frequency can be selected using the clock speed selection bits (CS1 and CS0).

In active mode, when the oscillation frequency (F_{CH}) is 12.5 MHz, the instruction cycle for the maximum speed (SYCC: CS1 and CS0 = 11_B) is $4/F_{CH}$ (= about 0.32 µs).

3.6.4 Clock Mode

The clock speed is switched by selecting one of four frequency-divided source clocks (gears).

■ Operations in Each Clock Mode

Table 3.6-2 Operations in Each Clock Mode

Clock speed		Standby mode	Clock	Operating clock in each block			Cause that cancels standby mode (excepting reset)
SYCC register (SYCC: CS1 and CS0)				CPU	time-base timer	Peripheral function	
(1, 1)	High speed	RUN	Generated	$F_{CH}/4$	$F_{CH}/2$	$F_{CH}/4$	Interrupt request
		Sleep		Stopped			
		Stop	Stopped		Stopped	Stopped	External interrupt
(1, 0)		RUN	Generated	$F_{CH}/8$	$F_{CH}/2$	$F_{CH}/8$	Interrupt request
		Sleep		Stopped			
		Stop	Stopped		Stopped	Stopped	External interrupt
(0, 1)		RUN	Generated	$F_{CH}/16$	$F_{CH}/2$	$F_{CH}/16$	Interrupt request
		Sleep		Stopped			
		Stop	Stopped		Stopped	Stopped	External interrupt
(0, 0)	Low speed	RUN	Generated	$F_{CH}/64$	$F_{CH}/2$	$F_{CH}/64$	Interrupt request
		Sleep		Stopped			
		Stop	Stopped		Stopped	Stopped	External interrupt

Each clock mode allows transition to a corresponding standby (sleep/stop) mode. For details of standby mode, see Section "3.7 Standby Mode (Low-Power Consumption Mode) ".

■ Gears (Clock Speed Switching Function)

Writing one of 00_B to 11_B into the clock speed selection bits (SYCC: CS1 and CS0) in the system clock control register selects one of four clock speeds.

The CPU and peripheral circuits operate using the clock speed selected. However, the gear does not affect the time-base timer.

Power consumption can be reduced by lowering the clock speed.

■ Operations in Active Mode

In active (RUN) mode, the oscillator is generating a clock. The CPU, time-base timer, and other peripheral circuits operate using the clock.

In active mode, all clock speeds except the time-base timer clock speed can be changed (using gears). In active mode, specifying standby mode results in a transition to sleep mode or stop mode.

Operations always start in RUN mode after a reset (any type). (Operating modes are cancelled by a reset.)

Note:

Do not rewrite the values in the oscillation stabilization wait time selection bits (SYCC: WT1 and WT0) while the clock is waiting for stabilization of oscillation. Using the system clock monitor bits, change the values in these bits after checking that SYCC: SCM is "1".

3.6.5 Oscillation Stabilization Wait Time

Oscillation stabilization wait time is to be applied when power is turned on to start the clock in RUN mode while the clock is stopped in stop mode.

■ Oscillation Stabilization Wait Time

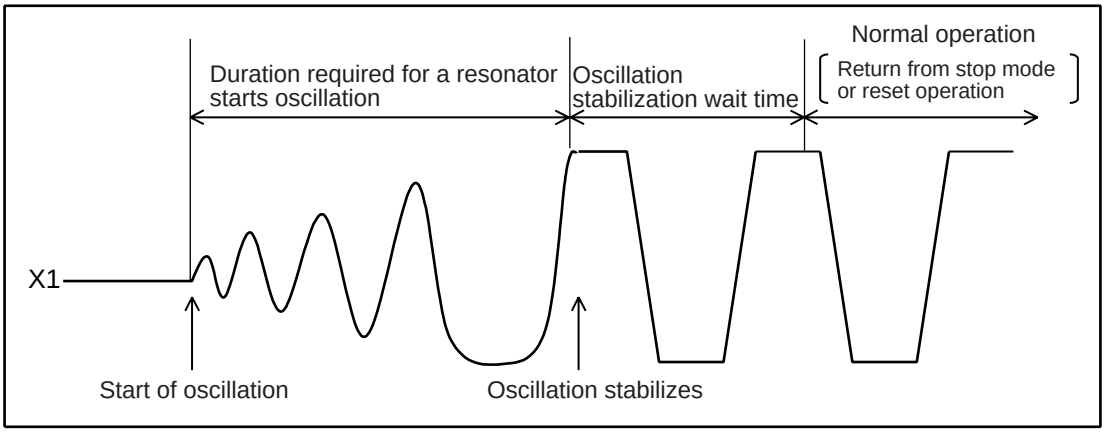
A ceramic or crystal resonator normally requires several or several tens of milli-seconds from oscillation start to oscillation stabilization at a specific cycle (oscillation frequency).

Thus, CPU operation must be prohibited immediately after the start of oscillation, and the clock is to be supplied to the CPU when oscillation is stable following the expiration of oscillation stabilization wait time.

The period during which oscillation becomes stable is dependent on the type of oscillator (such as crystal or ceramic) connected to the oscillation circuit (clock generator). Therefore, an oscillation stabilization wait time setting appropriate to the oscillator used must be selected.

Figure 3.6-6 shows changes in a frequency generated by an resonator from generation to stabilization.

Figure 3.6-6 Changes of a Frequency after Generation



■ Oscillation Stabilization Wait Time

Oscillation stabilization wait time is to be applied to start the clock in active mode while the clock is stopped.

Oscillation stabilization wait time is the duration from when the counter of the time-base timer is cleared to when the specified bits overflow.

- Oscillation stabilization wait time during operation

For oscillation stabilization wait time applied for a return from stop mode to active (RUN) mode due to external interrupt, one of three oscillation stabilization wait time settings can be selected using the oscillation stabilization wait time selection bits in the system clock control register (SYCC: WT1 and WT0).

- Oscillation stabilization wait time at a reset

  Option settings specify oscillation stabilization wait time at a reset (initial values of WT1 and WT0).

  Cancellation of stop mode by external reset also applies oscillation stabilization wait time.

  Table 3.6-3 shows the relationship between the active mode operation start conditions and oscillation stabilization wait time.

Table 3.6-3 Active Mode Operation Start Conditions and Oscillation Stabilization Wait Time

Active mode operation start condition	When power is turned on	Cancellation of stop mode	
		External reset	External interrupt
Selection of oscillation stabilization wait time	Option settings		SYCC: WT1, WT0

3.7 Standby Mode (Low-Power Consumption Mode)

The MB89202/F202RA series supports sleep mode and stop mode in standby mode. Transition to standby mode is controlled by the standby control register (STBC) settings.
In active mode, transition to sleep mode or stop mode is allowed.
In standby mode, operation of the CPU and peripheral functions is stopped to reduce power consumption.
This section describes the relationship between standby mode and clock mode and explains block operations in standby mode.

■ Standby Mode

In active mode, power consumption is reduced by lowering the speed of the operating clock for the CPU and peripheral circuits using clock speed switching (gears). However, in standby mode, the clock controller stops supply of the clock to the CPU (sleep mode) or stops oscillation of the source (stop mode) to reduce power consumption.

- Sleep mode

  In sleep mode, the CPU and watchdog timer are stopped. Peripheral functions operate using the normal clock.

- Stop mode

  In stop mode, the CPU and peripheral functions are stopped, and the clock does not oscillate. All the functions except for external interrupt halt.

3.7.1 Operations in Standby Mode

This section describes CPU and peripheral function operation in standby mode.

■ Operations in Standby Mode

Table 3.7-1 Operations of the CPU and Peripheral Functions in Standby Mode

Function		RUN	Sleep	Stop (SPL=0)	Stop (SPL=1)
Clock		Active	Active	Stopped	Stopped
CPU	Instruction	Active	Stopped	Stopped	Stopped
	ROM	Active	Holding	Holding	Holding
	RAM				
Peripheral function	I/O port	Active	Holding	Holding	Hi-Z
	Time-base timer	Active	Active	Stopped	Stopped
	Watchdog timer	Active	Stopped	Stopped	Stopped
	8-bit PWM timer/counter	Active	Active	Stopped	Stopped
	8/16-bit capture timer/counter	Active	Active	Stopped	Stopped
	UART	Active	Active	Stopped	Stopped
	8-bit serial I/O	Active	Active	Stopped	Stopped
	12-bit PPG	Active	Active	Stopped	Stopped
	Buzzer	Active	Active	Stopped	Stopped
	External interrupt 1 and 2	Active	Active	Active	Active
	A/D converter	Active	Active	Stopped	Stopped

- State of pins in standby mode

The state of most I/O pins can remain the same as those set immediately before transition to stop mode or set to Hi-Z using the pin state setting bit in the standby control register (STBC: SPL), regardless of clock mode.

Note:

For details on pin states in standby mode, see "APPENDIX E Pin State of the MB89202/F202RA Series ".

3.7.2 Sleep Mode

This section describes sleep mode.

■ Operations Relating to Sleep Mode

- Transition to sleep mode

  In sleep mode, the operating clock for CPU is stopped. Although the CPU stops storing data in the registers and RAM used immediately before transition to sleep mode, peripheral functions, excepting the watchdog timer, continue to operate.

  Writing "1" to the sleep bit in the standby control register (STBC: SLP) results in a transition to sleep mode. Any attempt to write "1" into the SLP bit while an interrupt request is being generated fails, transition to sleep mode cannot be made, and instructions are processed continuously. (Even after the interrupt is processed completely, transition to sleep mode is not possible.)

- Cancellation of sleep mode

  Sleep mode is cancelled by a reset or interrupt from a peripheral function.

  Pin states are initialized by the reset operation.

  When an interrupt request with an interrupt level higher than 11_B is generated in a peripheral function or external interrupt circuit in sleep mode, sleep mode is cancelled regardless of the CPU interrupt enable flag (CCR: I) or interrupt level bits (CCR: IL1 and IL0).

  When sleep mode is cancelled, a normal interrupt operation is performed, and if interrupts are acceptable, interrupt processing is performed. Otherwise, if interrupts are unacceptable, the processing resumes starting from an instruction next to the instruction which was issued immediately before transition to sleep mode.

3.7.3 Stop Mode

This section describes the stop mode.

■ Operations Relating to Stop Mode

● Transition to stop mode

In stop mode, the oscillation frequency is stopped. Most functions stop storing data in the registers and RAM used immediately before transition to stop mode.

The clock circuit stops oscillating, the peripheral functions and CPU stop operating, but the external interrupt circuit continues to operate.

Writing "1" to the stop bit in the standby control register (STBC: STP) causes a transition to stop mode. At that time, if the pin state setting bit (STBC: SPL) is "0", the states of the external pins are maintained. If the pin state setting bit is "1", the states of the external pins are set to Hi-Z (the states of pins for which a pull-up resistor is specified in the pull-up setting resistor are set to level "H").

An attempt to write "1" into the STP bit while an interrupt request is being generated fails, transition to stop mode cannot made, and instructions are processed continuously. (Even after the interrupt is processed completely, transition to stop mode is not made.)

For a transition to stop mode, prohibit the time-base timer interrupt request output (TBTC: TBIE = 0) when necessary.

● Cancellation of stop mode

Stop mode is cancelled by a reset or external interrupt.

When a reset occurs in stop mode, the reset operation is performed after oscillation stabilization wait time.

pin states are initialized by the reset operation.

When an interrupt request with an interrupt level higher than 11_B is generated in an external interrupt circuit in stop mode, stop mode is cancelled regardless of the CPU interrupt enable flag (CCR: I) or interrupt level bits (CCR: IL1 and IL0).

When stop mode is cancelled and oscillation stabilization wait time has expired, a normal interrupt operation is performed. Then, if interrupts are acceptable, interrupt processing is performed. Otherwise, an instruction following the instruction immediately before transition to stop mode is managed.

When an external interrupt cancels stop mode, part of the peripheral functions are restarted with data stored before the beginning of sleep mode. Therefore, the initial interval of the interval timer and other similar settings are rendered unknown. The peripheral functions must be initialized after returning from stop mode.

Note:

Among interrupts, only an interrupt request from the external interrupt circuit cancels the stop mode.

3.7.4 Standby Control Register (STBC)

The standby control register (STBC) controls transition to sleep /stop modes, pin state settings in stop mode, and software reset.

■ Standby Control Register (STBC)

Figure 3.7-1 Standby Control Register (STBC)

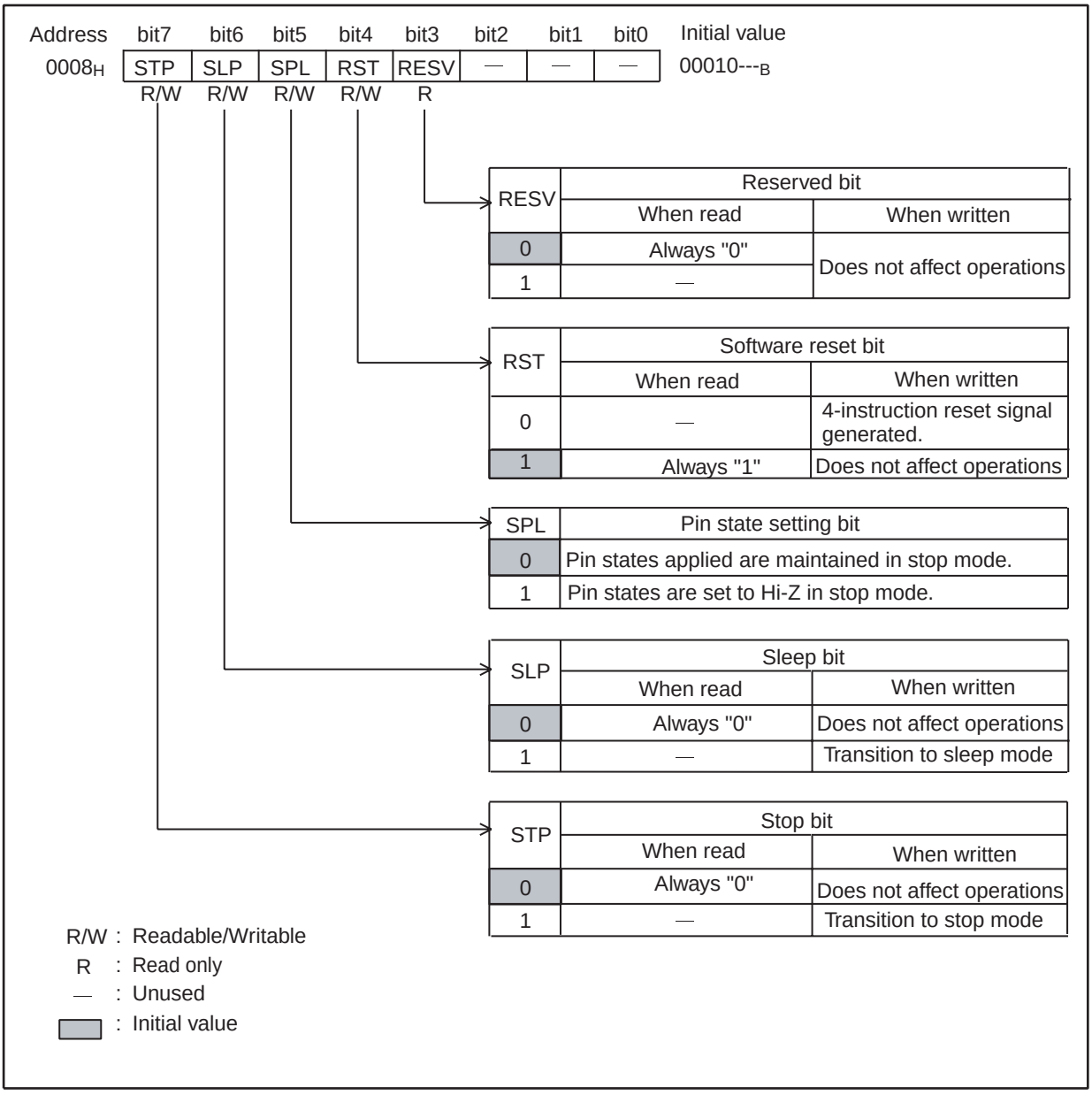


Table 3.7-2 Explanation of Functions of Each Bit in the Standby Control Register (STBC)

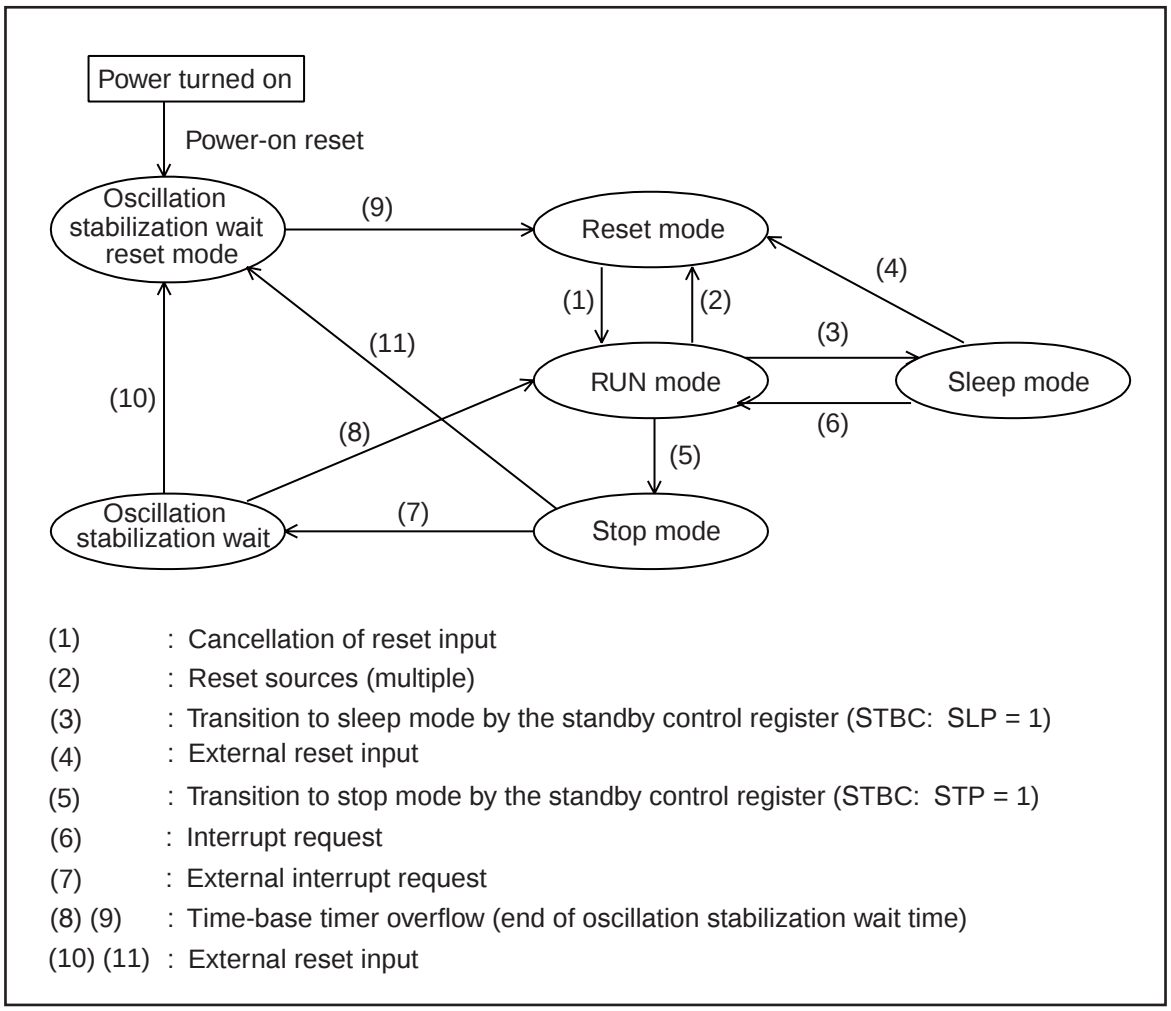
Bit name		Description
bit7	STP: Stop bit	This bit specifies transition to stop mode. Writing "1" into this bit allows transition to stop mode. Writing "0" into this bit does not affect operations. This bit is always read with the value of "0".
bit6	SLP: Sleep bit	This bit specifies transition to sleep mode. Writing "1" into this bit allows transition to sleep mode. Writing "0" into this bit does not affect operations. This bit is always read with the value of "0".
bit5	SPL: pin state setting bit	This bit specifies external pin states in stop mode. Writing "0" into this bit maintains states (levels) of the external pins at transition to stop mode. Writing "1" into this bit sets states of the external pins to Hi-Z (states of pins for which a pull-up resistor is specified are set to level "H"). This bit becomes "0" after a reset.
bit4	RST: Software reset bit	This bit specifies software reset. Writing "0" into this bit generates a source of 4-instruction cycle internal reset. Writing "1" into this bit does not affect operations. This bit is always read with the value of "1".
bit3	RESV: Reserved bit	This bit is always read with the value of "0". Writing a value into this bit does not affect operations.
bit2 to bit0	Unused bits	Values read out of these bits are undefined. Writing values into these bits does not affect operations.

3.7.5 Diagram for State Transition in Standby Mode

Figure 3.7-2 shows the state transition diagram in standby mode.

■ Diagram for State Transition in Standby Mode

Figure 3.7-2 State Transition Diagram



(1) : Cancellation of reset input
(2) : Reset sources (multiple)
(3) : Transition to sleep mode by the standby control register (STBC: SLP = 1)
(4) : External reset input
(5) : Transition to stop mode by the standby control register (STBC: STP = 1)
(6) : Interrupt request
(7) : External interrupt request
(8) (9) : Time-base timer overflow (end of oscillation stabilization wait time)
(10) (11) : External reset input

- Transition to and cancellation of clock mode (non-standby mode)

Table 3.7-3 Transition to and Cancellation of Clock Mode

State transition	Transition conditions
Transition to active mode after power-on reset	(9) End of oscillation stabilization wait time (output of time-base timer) (1) Cancellation of reset input
Reset in RUN mode	(2) External reset, software reset, or watchdog reset

- Transition to and cancellation of standby mode

Table 3.7-4 Transition to and Cancellation of Standby Mode

State transition	Transition conditions
Transition to sleep mode	(3) STBC: SLP=1
Cancellation of sleep mode	(6) Interrupt (each type) (4) External reset
Transition to stop mode	(5) STBC: STP=1
Cancellation of stop mode	(7) External interrupt (8) End of oscillation stabilization wait time (output of the time-base timer) (10) External reset (11) External reset (during oscillation stabilization wait)

Note:

In standby mode, the CPU and watchdog timer stop. Thus, software and watchdog resets do not occur.

3.7.6 Notes on Standby Mode

Even if the standby control register (STBC) sets standby mode, transition to standby mode is not allowed when a peripheral function generates an interrupt request. When an interrupt causes a return from standby mode to active mode, subsequent operations depend on whether interrupt requests are acceptable.

■ Transition to Standby Mode and Interrupt

When an interrupt request with an interrupt level higher than 11_B is generated in a peripheral function to the CPU, an attempt to write "1" into the stop bit (STBC: STP) or sleep bit (SLP) in the standby control register is ignored. Therefore, any attempt at transition to standby mode fails. (Even after the interrupt is processed, transition to standby mode is not allowed.)

This type of rejection does not depend on whether the CPU can accept interrupts.

Even if the CPU is processing an interrupt, transition to standby mode is allowed when the request flag bit for the interrupt has been cleared and there are no other interrupt requests to be processed.

■ Cancellation of Standby Mode by an Interrupt

When an interrupt request with an interrupt level higher than 11_B is generated in a peripheral function or another component in sleep mode or stop mode, standby mode is cancelled. This operation does not depend on whether the CPU can accept interrupts.

After cancellation of standby mode, the CPU normally takes a branch to the interrupt processing routine if the priority of the interrupt level setting register (ILR1 to ILR4) corresponding to the interrupt request is higher than the level specified in the interrupt level bits (CCR: IL1 and IL0) in the condition code register and if the interrupt enable flag is turned on (CCR: I = 1). Otherwise, an instruction is managed following the instruction causing standby mode to be set.

To prohibit a branch to the interrupt processing routine immediately after return, interrupts must be prohibited before standby mode is set.

■ Notes on Setting Standby Mode

For setting standby mode using the standby control register (STBC), use the settings specified in Table 3.7-5 . When 1 is set to both bits at the same time, stop mode has precedence over sleep mode. However, it is recommended that "1" not be set to the bits at the same time.

Table 3.7-5 Low-power Consumption Mode Established using the Standby Control Register (STBC)

STBC register		Mode
STP(bit7)	SLP(bit6)	
0	0	Active
0	1	Sleep
1	0	Stop

■ Oscillation Stabilization Wait Time

The oscillator for oscillation frequency stops in stop mode, thus oscillation stabilization wait time must be applied after the oscillator is activated.

Use one of three clock oscillation stabilization wait time settings generated by the time-base timer.

If the interval selected for the time-base timer is shorter than the oscillation stabilization wait time, an interval timer interrupt request is generated during oscillation stabilization wait time. To prevent this from occurring, disable output of time-base timer interrupt requests (TBTC: TBIE = 0) before transition to stop mode when necessary.

3.8 Memory Access Mode

The MB89202/F202RA series supports only single-chip mode for access to memory.

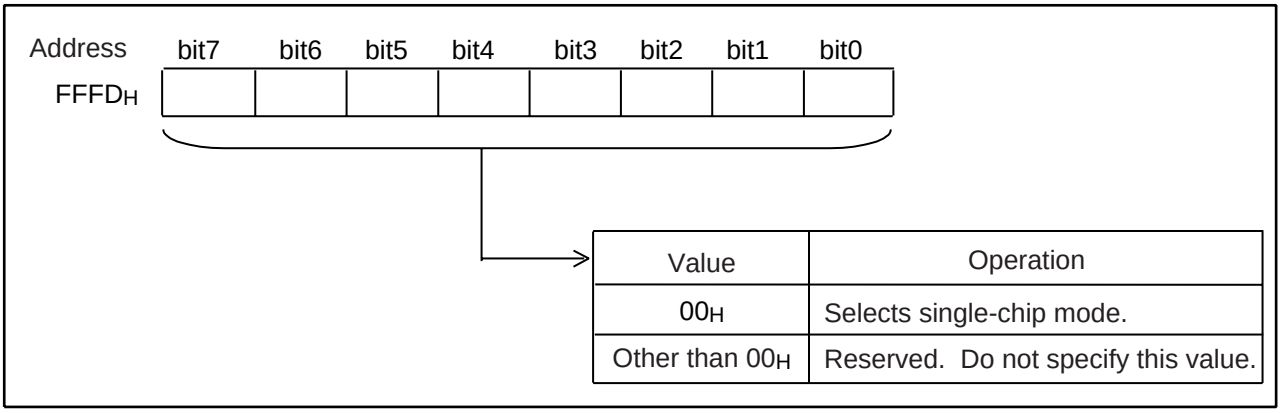
■ Single-chip Mode

In single-chip mode, only internal RAM and ROM are used. The CPU can access only the internal I/O area, RAM area, and ROM area.

■ Mode Data

Set 00_H into the mode data in internal ROM to select single-chip mode.

Figure 3.8-1 Configuration of Mode Data



■ Operations for Selecting Memory Access Mode

Only single-chip mode is selectable.

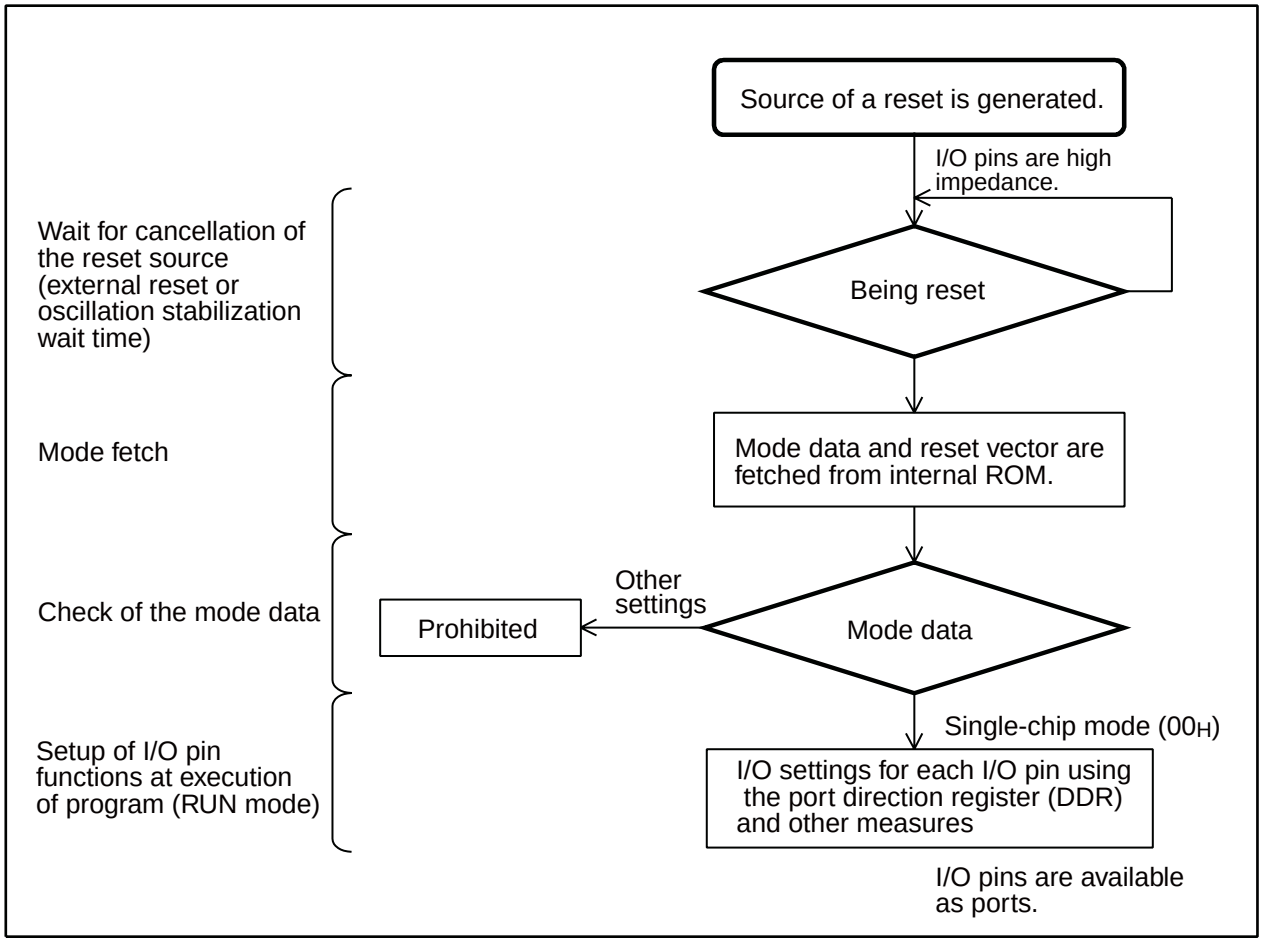
Table 3.8-1 provides the settings for the mode pins and mode data.

Table 3.8-1 Settings for Mode Data

Memory access mode	Mode data
Single-chip mode	00_H
Other modes	Prohibited

Figure 3.8-2 shows the operations for selecting memory access.

Figure 3.8-2 Operations for Selecting Memory Access



CHAPTER 4

I/O PORTS

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

4.1 Overview of I/O Ports

Six I/O ports (comprising 26 pins) are available as general-purpose I/O ports (parallel I/O ports).
These ports also serve peripherals (as I/O pins for specific peripheral functions).

■ Functions of I/O Ports

The I/O ports function to output data from the CPU to I/O pins via their port data register (PDR) and send signals input to I/O pins to the CPU. For some ports, the I/O direction of I/O pins can be set by optionally setting the bits of the port data direction register (DDR), with the bits corresponding to the pins.

The functions of the ports and peripherals for which the ports may serve are summarized below.

- Port 0: General-purpose I/O port may also serve peripherals (external interrupt 2 and analog input pins)
- Port 3: General-purpose I/O port may also serve peripherals (12-bit PPG, external interrupt 1, UART, 8-bit serial I/O, 8/16-bit timers, and buzzer output pin)
- Port 4: General-purpose I/O port of a type switched between CMOS push-pull and N-ch open-drain may also serve peripherals (analog input pins)
- Port 5: General-purpose I/O port may also serve peripherals (8-bit PWM pin)
- Port 6: General-purpose I/O port (for MB89F202/F202RA, P61, P60 are input port)
- Port 7: General-purpose I/O port

Table 4.1-1 lists the functions of the ports, and Table 4.1-2 lists the register of ports.

Table 4.1-1 Functions of Ports

Port name	Pin name	Input form	Output form	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Port 0	P00/$\overline{\text{INT20}}$/ AN4 to P07/$\overline{\text{INT27}}$	Hysteresis CMOS analog	CMOS push-pull	P07	P06	P05	P04	P03	P02	P01	P00
				$\overline{\text{INT27}}$	$\overline{\text{INT26}}$	$\overline{\text{INT25}}$	$\overline{\text{INT24}}$	AN7	AN6	AN5	AN4
								$\overline{\text{INT23}}$	$\overline{\text{INT22}}$	$\overline{\text{INT21}}$	$\overline{\text{INT20}}$
Port 3	P30/UCK/SCK to P37/BZ/PPG	CMOS hysteresis		P37	P36	P35	P34	P33	P32	P31	P30
				BZ/ PPG	INT12	INT11	TO/ INT10	EC	UI/SI	UO/SO	UCK/ SCK
Port 4	P40/AN0 to P43/AN3	CMOS analog	CMOS push-pull or N-ch open-drain	-	-	-	-	P43	P42	P41	P40
								AN3	AN2	AN1	AN0
Port 5	P50/PWM	CMOS hysteresis	CMOS push-pull	-	-	-	-	-	-	-	P50
											PWM
Port 6	P60, P61	CMOS hysteresis	CMOS push-pull or N-ch open-drain	-	-	-	-	-	-	P61	P60
										-	-
Port 7	P70 to P72	CMOS	CMOS	-	-	-	-	-	P72	P71	P70

Table 4.1-2 Registers of Ports

Register name		Read/Write	Address	Initial value
Port 0 data register	(PDR0)	R/W	0000_H	$XXXXXXXX_B$
Port 0 data direction register	(DDR0)	W *1	0001_H	00000000_B
Port 0 pull-up setting register	(PUL0)	R/W	0070_H	00000000_B
Port 3 data register	(PDR3)	R/W	$000C_H$	$XXXXXXXX_B$
Port 3 data direction register	(DDR3)	W *1	$000D_H$	00000000_B
Port 3 pull-up setting register	(PUL3)	R/W	0071_H	00000000_B
Port 4 data register	(PDR4)	R/W	$000F_H$	$----XXXX_B$
Port 4 data direction register	(DDR4)	R/W	0010_H	$----0000_B$
Port 4 output form setting register	(OUT4)	R/W	0011_H	$----0000_B$
Port 5 data register	(PDR5)	R/W	0012_H	$-------X_B$
Port 5 data direction register	(DDR5)	R/W	0013_H	$-------0_B$
Port 5 pull-up setting register	(PUL5)	R/W	0072_H	$-------0_B$
Port 6 data register	(PDR6)	R/W	0060_H	$------XX_B$
Port 6 data direction register *2	(DDR6)	R/W	0061_H	$------00_B$
Port 6 pull-up setting register	(PUL6)	R/W	0062_H	$------00_B$
Port 7 data register	(PDR7)	R/W	0063_H	$-----XXX_B$
Port 7 data direction register	(DDR7)	R/W	0064_H	$-----000_B$
Port 7 pull-up setting register	(PUL7)	R/W	0065_H	$-----000_B$

R/W : Readable and Writable
W : Write only
X : Undefined
*1 : DDR0 and DDR3 cannot be used for bit manipulation instructions.
*2 : DDR6 is not used in MB89F202/F202RA.

4.2 Port 0

Port 0 is a general-purpose I/O port and may also serve as peripheral inputs. The pins of this port can be used for peripherals or normal port function that can be selected according to the setting of a bit corresponding to the pin on a specific register.
This section mainly explains the general-purpose I/O function of the port.
This section also describes the structure, pins, and associated registers of port 0 and provides a block diagram of pins.

■ Structure of Port 0

Port 0 comprises the following four elements:

- General-purpose I/O pins, external interrupt 2, and analog input pins (P00/$\overline{\text{INT20}}$/AN4 to P07/$\overline{\text{INT27}}$)
- Port 0 data register (PDR0)
- Port 0 data direction register (DDR0)
- Port 0 pull-up setting register (PUL0)

■ Pins of Port 0

Port 0 has eight general-purpose I/O pins. When used as input pins at the same time, these pins can be also used as external interrupt input pins.

Table 4.2-1 lists the pins of port 0.

Table 4.2-1 Pins of Port 0

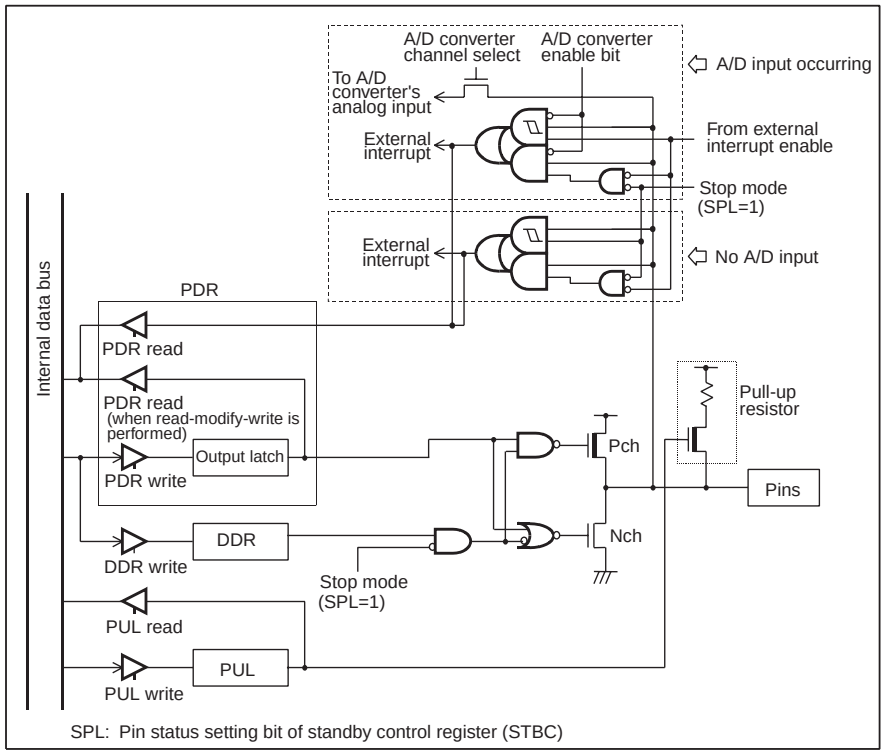
Port name	Pin name	Function	Peripherals for which a pin may serve	Input and output form		Circuit type
				Input	Output	
Port 0	P00/$\overline{\text{INT20}}$/AN4	P00 general-purpose I/O	$\overline{\text{INT20}}$: external interrupt input 20 AN4 : analog input 4	Analog CMOS hysteresis	CMOS	G
	P01/$\overline{\text{INT21}}$/AN5	P01 general-purpose I/O	$\overline{\text{INT21}}$: external interrupt input 21 AN5 : analog input 5			
	P02/$\overline{\text{INT22}}$/AN6	P02 general-purpose I/O	$\overline{\text{INT22}}$: external interrupt input 22 AN6 : analog input 6			
	P03/$\overline{\text{INT23}}$/AN7	P03 general-purpose I/O	$\overline{\text{INT23}}$: external interrupt input 23 AN7 : analog input 7			
	P04/$\overline{\text{INT24}}$	P04 general-purpose I/O	$\overline{\text{INT24}}$: external interrupt input 24	CMOS hysteresis		D
	P05/$\overline{\text{INT25}}$	P05 general-purpose I/O	$\overline{\text{INT25}}$: external interrupt input 25			
	P06/$\overline{\text{INT26}}$	P06 general-purpose I/O	$\overline{\text{INT26}}$: external interrupt input 26			
	P07/$\overline{\text{INT27}}$	P07 general-purpose I/O	$\overline{\text{INT27}}$: external interrupt input 27			

For circuit type, see Section "1.7 Pin Functions Description " and "1.8 I/O Circuit Types ".

For pin operation when used as analog input, see "CHAPTER 12 A/D CONVERTER ".

■ Block Diagram of Port 0

Figure 4.2-1 Block Diagram of Port 0



Note:

When the A/D converter is used, deselect pull-up action for pins P03/$\overline{\text{INT23}}$/AN7 to P00/$\overline{\text{INT20}}$/AN4.

Pins set to be used as analog input pins must not be used as an output port.

■ Registers PDR0, DDR0, and PUL0 of Port 0

Registers PDR0, DDR0, and PUL0 are associated with port 0.

The bits of these registers correspond to the pins of port 0 in one-to-one correspondence.

Table 4.2-2 tabulates the correspondence between the pins and the bits of the port 0 registers.

Table 4.2-2 Correspondence between the Pins and the Bits of the Port 0 Registers

Port name	Bits of associated registers and corresponding pins								
Port 0	PDR0, DDR0, PUL0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Pin corresponding to bit	P07	P06	P05	P04	P03	P02	P01	P00

4.2.1 Registers of Port 0 (PDR0, DDR0, and PUL0)

This section describes the registers associated with port 0.

■ Functions of Port 0 Registers

- Port 0 data register (PDR0)

  The PDR0 register indicates the state of the output latch. For a pin set to function as an output port, the same value ("0" or "1") as the value state of the output pin can be read from this register. If the pin is set to function as an input port, however, its output latch value cannot be read from the register.

  Note:

  When a bit manipulation instruction (SETB, CLRB) is executed, the output latch values, not the value states of the pins, are read; thus, output latch values, other than those for bits to be manipulated, do not change.

- Port 0 data direction register (DDR0)

  The DDR0 register sets the I/O direction of each pin per bit.

  When a bit of the DDR0 register corresponding to a pin of port 0 is set to "1", the pin functions as an output port. When the bit is set to "0", the pin functions as an input port.

  Note:

  Because the DDR0 register is write only, bit manipulation instructions (SETB, CLRB) do not apply.

- Setting a port pin to serve external interrupt inputs

  If a pin of port 0 is used as an external interrupt input pin, enable the external interrupt circuit operation and set the pin to function as an input port. When the pin is set in this mode, its output latch value has no significance.

- Setting a port pin to serve analog inputs

  If a pin of port 0 is used as an analog input pin, write "0" for the bit corresponding to the pin on the DDR0 register. The output transistor is then set to OFF and the pin is set in the Hi-Z state.

  Set the bit of the ADEN register of the A/D converter to "1", the bit corresponding to the analog input pin in use.

- Setting the input to a peripheral enable

  If a peripheral with an input pin is used, set the pin of port 0 for the input to the peripheral to function as an input port. In this mode, the corresponding output latch value has no significance.

Table 4.2-3 lists the functions of the port 0 registers.

Table 4.2-3 Functions of Port 0 Registers

Register name	Data	When being read	When being written	Read/ Write	Address	Initial value
Port 0 data register (PDR0)	0	Pin state is "L" level.	Output latch of "0" is set and "L" level is output to the pin in output port mode.	R/W	0000_H	$XXXXXXXX_B$
	1	Pin state is "H" level.	Output latch of "1" is set and "H" level is output to the pin in output port mode.			
Port 0 data direction register (DDR0)	0	Read prohibited (write only)	Output transistor operation is disabled and the pin is set to serve as an input pin.	W	0001_H	00000000_B
	1		Output transistor operation is enabled and the pin is set to serve as output pin.			

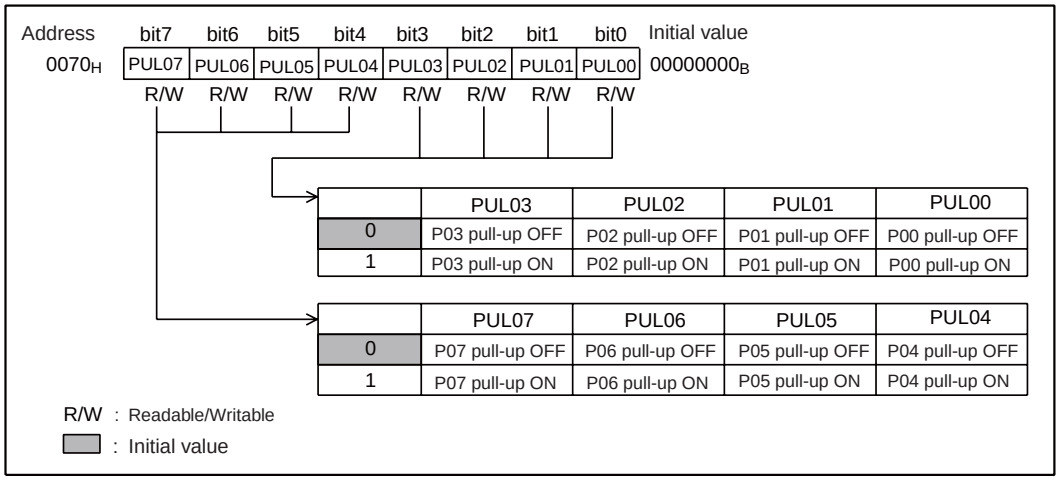
R/W : Readable/Writable
W : Write only
X : Undefined

● Port 0 pull-up setting register (PUL0)

The bits of the pull-up setting register correspond to the pins of port 0 in one-to-one correspondence. When the pull-up resistor is selected by using the pull-up setting register, the pin will be at "H" level (pull-up state) instead of Hi-Z during stop (SPL = 1). During a reset, however, the pull-up is invalid and the pin remains at Hi-Z.

Figure 4.2-2 shows the pull-up resistor settings assigned to the values of the bits of the port 0 pull-up register.

Figure 4.2-2 Pull-up Resistor Settings (PUL0)



4.2.2 Operations of Port 0 Functions

This section describes the operation of port 0.

■ Operation of Port 0

- Operation in output port mode

  When "1" is written to a bit of the DDR0 register, the bit corresponding to a pin of port 0, the pin functions as an output port.

  In output port mode, the output transistor operation is enabled and the output latch data is output to the pin.

  Once data has been written into the PDR0 register, the written data is held in the output latch and output to the pin as it is.

  The value state of the pin can be read by reading the PDR0 register.

- Operation in input port mode

  When "0" is written to a bit of the DDR0 register, the bit corresponding to a pin of port 0, the pin functions as an input port.

  In input port mode, the output transistor is OFF and the pin status is Hi-Z.

  Once data has been written into the PDR0 register, the written data is held in the output latch but is not output to the pin.

  The value state of the pin can be read by reading the PDR0 register.

- Operation in external interrupt input mode

  Set a bit of the DDR0 register to "0", the bit corresponding to a pin of port 0 that is to serve as an external interrupt input pin, to set the pin to function as an input port.

  The value state of the pin can be read by reading the PDR0 register regardless of whether external interrupt inputs or interrupt request outputs are enabled or disabled.

- Operation in analog input mode

  To use a pin of port 0 as analog input and to inhibit output transistor operation, set the bit corresponding to the analog input pin to "0" on the DDR0 register. The value state of the pin can be read by reading the PDR0 register.

  Set the bit of the ADEN register of the A/D converter to "1", the bit corresponding to the analog input pin in use.

- Operation when a reset is performed

  When the CPU is reset, the bits of the DDR0 register are initialized to "0". Thus, all output transistors become OFF and the pins become Hi-Z.

  However, CPU resets do not initialize the PDR0 register. If a pin is used as an output port after the reset, reinitialize the PDR0 register to contain new output data in the bit position corresponding to the pin and then set the corresponding bit of the DDR0 register so that the pin will function as an output port.

- Operation in stop mode

When the pin state setting bit of the standby control register (STBC: SPL) is "1" and when the stop mode is entered, the output transistor is turned OFF and the pin becomes Hi-Z because the output transistor is forcibly turned OFF without respect to the value existing on the DDR0 register in the bit position corresponding to the pin.

Input remains fixed to prevent leaks by input open.

Table 4.2-4 summarizes the operating modes of the pins of port 0.

Table 4.2-4 Operating Modes of Pins of Port 0

Pin name	Normal operation, sleep, stop (SPL = 0)	Stop (SPL = 1)	At a reset
P00/$\overline{\text{INT20}}$/AN4 to P03/$\overline{\text{INT23}}$/AN7	General-purpose I/O port may also serve external interrupt inputs or analog inputs	Hi-Z (External interrupt input)	Hi-Z
P04/$\overline{\text{INT24}}$ to P07/$\overline{\text{INT27}}$	General-purpose I/O port may also serve external interrupt inputs		

SPL : Pin state setting bit of standby control register (STBC: SPL)
Hi-Z: High impedance

Note:

When the pull-up resistor is selected by using the pull-up setting register, the pin state will be "H" level instead of Hi-Z in stop mode (SPL = 1). During a reset, however, the pull-up is invalid with the pin remaining at Hi-Z.

4.3 Port 3

Port 3 is a general-purpose I/O port and may also serve as input pins for external interrupts as well as input and output pins for peripherals.
This section mainly explains the general-purpose I/O function of the port.
This section also describes port 3 concerning to the structure, pins, a block diagram of pins, and associated registers.

■ Structure of Port 3

Port 3 comprises the following four elements:

- General-purpose I/O pins, external interrupt 1 input pins, and input/output pins for peripherals (P30/UCK/SCK to P37/BZ/PPG)
- Port 3 data register (PDR3)
- Port 3 data direction register (DDR3)
- Port 3 pull-up setting register (PUL3)

■ Pins of Port 3

Port 3 has eight CMOS I/O pins. These pins can be used as both input pins and external interrupt input pins at the same time. These pins cannot be used as a general-purpose I/O port when being used for peripherals.

Table 4.3-1 lists the pins of port 3.

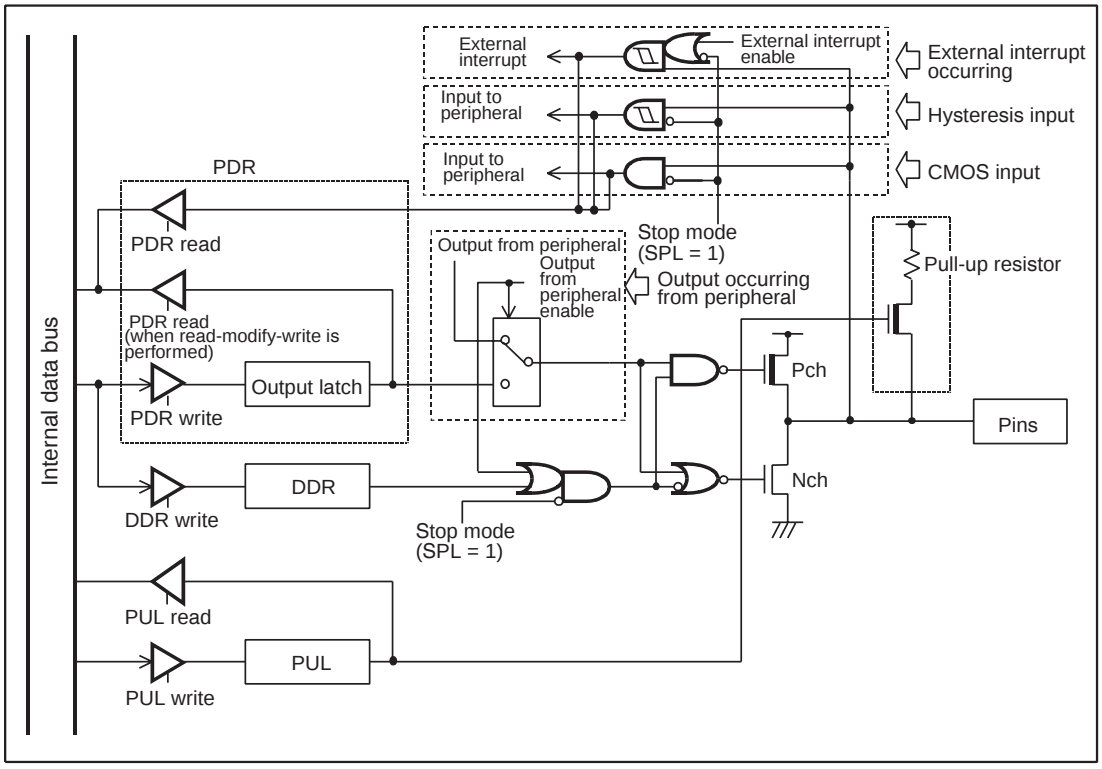
Table 4.3-1 Pins of Port 3

Port name	Pin name	Function	Peripherals for which the pin may serve		Input and output form		Circuit type
					Input	Output	
Port 3	P30/UCK/SCK	P30 general-purpose I/O	UCK SCK	UART clock I/O 8-bit serial I/O clock I/O	CMOS hysteresis	CMOS	B
	P31/UO/SO	P31 general-purpose I/O	UO SO	UART data output 8-bit serial I/O data output	CMOS		E
	P32/UI/SI	P32 general-purpose I/O	UI SI	UART data input 8-bit serial I/O data input	CMOS hysteresis		B
	P33/EC	P33 general-purpose I/O	EC	8/16-bit timer and counter clock inputs	CMOS hysteresis		
	P34/TO/INT10	P34 general-purpose I/O	TO INT10	8/16-bit timer and counter timer outputs External interrupt input 10			
	P35/INT11	P35 general-purpose I/O	INT11	External interrupt input 11			
	P36/INT12	P36 general-purpose I/O	INT12	External interrupt input 12			
	P37/BZ/PPG	P37 general-purpose I/O	BZ PPG	Buzzer output 12-bit PPG output	CMOS		E

For circuit type, see "1.7 Pin Functions Description ".

■ Block Diagram of Port 3

Figure 4.3-1 Block Diagram of Port 3



Note:

Because the value states of the pins are always input to the external interrupt circuit, when a pin is used as a normal I/O port, the operation of the external interrupt circuit corresponding to the pin must be inhibited. See "CHAPTER 10 EXTERNAL INTERRUPT CIRCUIT 1 (EDGE) ".

■ Registers PDR3, DDR3, and PUL3 of Port 3

The registers PDR3, DDR3, and PUL3 are associated with port 3.

The bits of these registers correspond to the pins of port 3 in one-to-one correspondence.

Table 4.3-2 tabulates the correspondence between the pins and the bits of port 3 registers.

Table 4.3-2 Correspondence between the Pins and the Bits of Port 3 Registers

Port name	Bits of associated registers and corresponding pins								
Port 3	PDR3, DDR3, PUL3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Pin corresponding to bit	P37	P36	P35	P34	P33	P32	P31	P30

4.3.1 Registers of Port 3 (PDR3, DDR3, PUL3)

This section describes the registers associated with port 3.

■ Functions of Port 3 Registers

- Port 3 data register (PDR3)

  The PDR3 register indicates the state of the pins. For a pin set to function as an output port, the same value ("0" or "1") as held by the output latch can be read from this register. If the pin is set to function as an input port, however, its output latch value cannot be read from the register.

  Note:

  When a bit manipulation instruction (SETB, CLRB) is executed, the output latch values, not the value states of the pins, are read; thus, output latch values, excepting those for bits to be manipulated, do not change.

- Port 3 data direction register (DDR3)

  The DDR3 register sets the I/O direction of each pin per bit.

  When a bit of the DDR3 corresponding to a pin of port 3 is set to "1", the pin functions as an output port. When the bit is set to "0", the pin functions as an input port.

  Note:

  Because the DDR3 register is write only, bit manipulation instructions (SETB, CLRB) do not apply.

- Setting a port pin to serve external interrupts

  If a pin of port 3 is used as an external interrupt input pin, enable the external interrupt circuit operation and set the pin to function as an input port.

  When the pin is set in this mode, its output latch value has no significance.

- Setting the output from a peripheral enable

  If a peripheral with an output pin is used, set the output enable bit of the peripheral enable.

  Because the output from the peripheral has priority, the values set on the PDR3 and DDR3 registers in the bit position corresponding to the output pin for the peripheral have no significance, regardless of the value output from the peripheral and the output enabled.

- Setting the input to a peripheral enable

  If a peripheral with an input pin is used, set the pin of port 3 for the input to the peripheral to function as an input port. In this mode, the corresponding output latch value has no significance.

Table 4.3-3 lists the functions of port 3 registers.

Table 4.3-3 Functions of Port 3 Registers

Register name	Data	When being read	When being written	Read/ Write	Address	Initial value
Port 3 data register (PDR3)	0	Pin state is "L" level.	Output latch of "0" is set and "L" level is output to the pin in output port mode.	R/W	$000C_H$	$XXXXXXXX_B$
	1	Pin state is "H" level.	Output latch of "1" is set and "H" level is output to the pin in output port mode.			
Port 3 data direction register (DDR3)	0	Read prohibited (write only)	Output transistor operation is disabled and the pin is set to serve as an input pin.	W	$000D_H$	00000000_B
	1		Output transistor operation is enabled and the pin is set to serve as an output pin.			

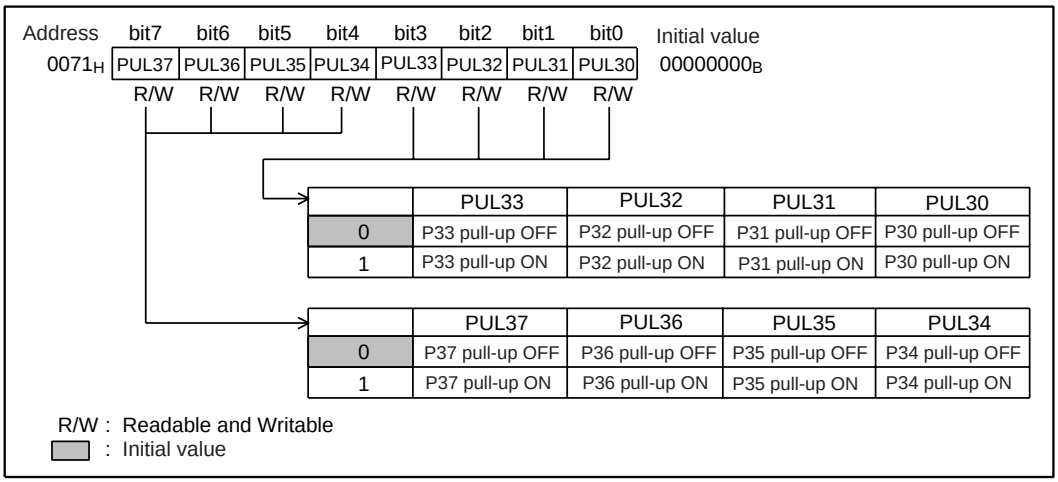
R/W : Readable and Writable
W : Write only
X : Undefined

● Port 3 pull-up setting register (PUL3)

The bits of the pull-up setting register correspond to the pins of port 3 in one-to-one correspondence. When the pull-up resistor is selected by using the pull-up setting register, the pin state will be "H" level (pull-up state) instead of Hi-Z during stop (SPL = 1). During a reset, however, the pull-up is invalid and the pin remains at Hi-Z.

Figure 4.3-2 shows the pull-up resistor settings assigned to the values of the bits of the port 3 pull-up register.

Figure 4.3-2 Pull-up Setting Register (PUL3)



4.3.2 Operations of Port 3 Functions

This section describes the operation of port 3.

■ Operation of Port 3

- Operation in output port mode

  When "1" is written for a bit of the DDR3 register, the bit corresponding to a pin of port 3, the pin functions as an output port.

  In output port mode, output transistor operation is enabled and output latch data is output to the pin.

  Once data has been written into the PDR3 register, the written data is held in the output latch and output to the pin as it is.

  The value state of the pin can be read by reading the PDR3 register.

- Operation in input port mode

  When "0" is written for a bit of the DDR3 register, the bit corresponding to a pin of port 3, the pin functions as an input port.

  In input port mode, the output transistor is OFF and the pin state is Hi-Z.

  Once data has been written into the PDR3 register, the written data is held in the output latch but is not output to the pin.

  The value state of the pin can be read by reading the PDR3 register.

- Operation in external interrupt input mode

  Set a bit of the DDR3 register to "0", the bit corresponding to a pin of port 3 that is to serve as an external interrupt input pin, to set the pin to function as an input port.

  The value state of the pin can be read by reading the PDR3 register regardless of whether or not the external interrupt inputs or interrupt request outputs are enabled.

- Operation in mode enabling the output from a peripheral

  When the output enable bit for a peripheral is set to enable, the corresponding pin is set to serve the output from the peripheral.

  Because the value state of the pin can be read from the PDR3 register even when the output from the peripheral is enabled, the value output from the peripheral can be read.

- Operation in mode enabling the input to a peripheral

  Set a bit of the DDR3 register to "0", the bit corresponding to the pin of port 3 assigned for the input to the desired peripheral, for the pin to function as an input port.

  The value state of the pin is always input to the peripheral (except during stop mode).

  The value state of the pin can be read by reading the PDR3 register regardless of whether or not the peripheral is using the input pin.

- Operation when a reset is performed

  When the CPU is reset, the bits of the DDR3 register are initialized to "0", at which time the output transistors become OFF (input port mode) and the pins become Hi-Z.

  However, CPU resets do not initialize the PDR3 register. If a pin is used as an output port after the reset, reinitialize the PDR3 register to contain new output data in the bit position corresponding to the pin and then set the corresponding bit of the DDR3 register so that the pin will function as an output port.

- Operation in stop mode

  When the pin state setting bit of the standby control register (STBC: SPL) is set to "1" and when the stop mode is entered, the pin becomes Hi-Z because the output transistor is turned OFF regardless of the value existing on the DDR3 register in the bit position corresponding to the pin.

  Table 4.3-4 summarizes the operating modes of the pins of port 3.

Table 4.3-4 Operating Modes of Pins of Port 3

Pin name	Normal operation, sleep, stop (SPL = 0)	Stop (SPL = 1)	At a reset
P30/UCK/SCK to P33/EC, P37/BZ/PPG	General-purpose I/O port may also serve I/O for peripherals	Hi-Z	Hi-Z
P34/TO/INT10 to P36/INT12	General-purpose I/O port may also serve outputs from peripherals and external interrupt inputs	Hi-Z (External interrupt input)	

SPL : Pin state setting bit of standby control register (STBC: SPL)
Hi-Z: High impedance

Note:

When the pull-up resistor is selected by using the pull-up setting register, the pin state will be "H" level instead of Hi-Z in stop mode (SPL = 1). During a reset, however, the pull-up is invalid with the pin remaining at Hi-Z.

4.4 Port 4

Port 4 is a type of I/O port that is switched between CMOS push-pull and N-ch open-drain and may also serve analog inputs. Each pin of this port can be used for peripherals or normal port function that can be selected according to the setting of the bit corresponding to the pin on a specific register.
This section explains the I/O port function of CMOS push-pull/ N-ch open-drain type. This section also describes port 4 concerning to the structure, pins, a block diagram of pins, and associated registers.

■ Structure of Port 4

Port 4 comprises the following four elements:

- Type of I/O pins that are switched between CMOS push-pull and N-ch open-drain and analog input pins (P40/AN0 to P43/AN3)
- Port 4 data register (PDR4)
- Port 4 data direction register (DDR4)
- Port 4 output format setting register (OUT4)

■ Pins of Port 4

Port 4 has four I/O pins of CMOS push-pull/N-ch open-drain.

These pins can also be used as analog input pins.

Those pins that are used for analog inputs cannot be used as a general-purpose I/O port. Table 4.4-1 lists the pins of port 4.

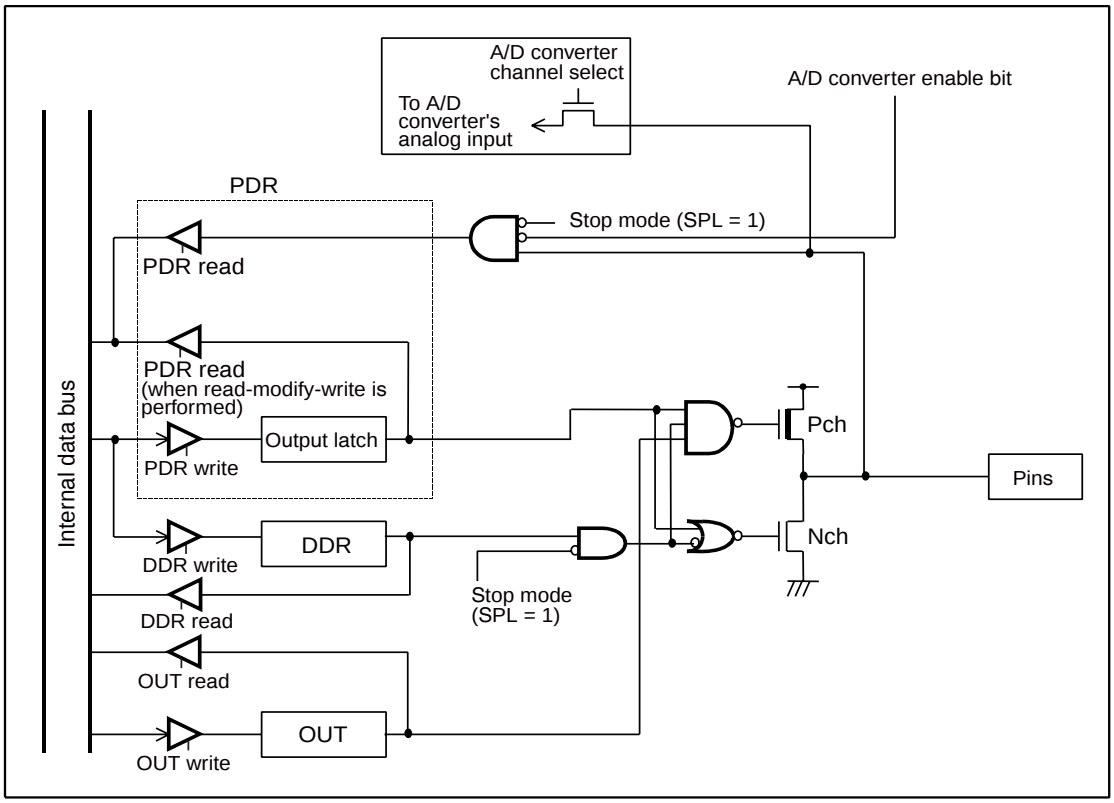
Table 4.4-1 Pins of Port 4

Port name	Pin name	Function	Peripherals for which the pin may serve	Input and output form		Circuit type
				Input	Output	
Port 4	P40/AN0	P40 I/O	AN0 analog input 0	CMOS analog	CMOS push-pull/N-ch open-drain	F
	P41/AN1	P41 I/O	AN1 analog input 1			
	P42/AN2	P42 I/O	AN2 analog input 2			
	P43/AN3	P43 I/O	AN3 analog input 3			

For circuit type, see Section "1.7 Pin Functions Description " and "1.8 I/O Circuit Types ".

■ Block Diagram of Port 4

Figure 4.4-1 Block Diagram of Port 4



■ Registers of Port 4

The registers PDR4, DDR4, and OUT4 are associated with port 4.

The bits of these registers correspond to the pins of port 4 in one-to-one correspondence.

Table 4.4-2 tabulates the correspondence between the pins and the bits of the port 4 registers.

Table 4.4-2 Correspondence between the Pins and the Bits of the Port 4 Register

Port name	Bits of associated registers and corresponding pins								
Port 4	PDR4, DDR4, OUT4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Pin corresponding to bit	-	-	-	-	P43	P42	P41	P40

4.4.1 Registers of Port 4 (PDR4)

This section describes the registers associated with port 4.

■ Functions of Port 4 Registers

● Port 4 data register (PDR4)

The PDR4 register indicates the state of the pins. For a pin set to function as an output port, the same value ("0"or "1") as held by the output latch can be read from this register. If the pin is set to function as an input port, however, its output latch value cannot be read from the register.

Note:

When a bit manipulation instruction (SETB, CLRB) is executed, the output latch values, not the value states of the pins, are read; thus, output latch values, excepting those for bits to be manipulated, do not change.

Table 4.4-3 lists the functions of the port 4 registers.

Table 4.4-3 Functions of Port 4 Registers

Register name	Data	When being read		When being written	Read/ Write	Address	Initial value
Port 4 data register (PDR4)	0	Pin state is "L" level.		Output latch of "0" is set and "L" level is output to the pin in output port mode.	R/W	$000F_H$	----$XXXX_B$
	1	Pin state is "H" level.	N-ch open-drain type	Output latch of "1" is set and the pin in output port mode is set at Hi-Z.			
			CMOS push-pull type	Output latch of "1" is set and "H" level is output to the pin in output port mode.			
Port 4 data direction register (DDR4)	0	Input port pin		The pin is set to function as input pin with output transistor operation disabled.	R/W	0010_H	----0000_B
	1	Output port pin		The pin is set to function as output pin with output transistor operation enabled.			
Port 4 output format setting register (OUT4)	0	N-ch open-drain type		Output format of the pin is set to N-ch open-drain type.	R/W	0011_H	----0000_B
	1	CMOS push-pull type		Output format of the pin is set to CMOS push-pull type.			

R/W : Readable and Writable
X : Undefined

4.4.2 Operations of Port 4 Functions

This section describes the operation of port 4.

■ Operation of Port 4

- Operation in output port mode

  When "1" is written for a bit of the DDR4 register, the bit corresponding to a pin of port 4, the pin functions as an output port.

  In output port mode, the output transistor operation is enabled and output latch data is output to the pin.

  By setting the bit corresponding to the pin on the OUT4 register, N-ch open-drain or CMOS push-pull type can be selected as the output format of the pin.

  Once data has been written into the PDR4 register, the written data is held in the output latch and output to the pin as it is.

  The value state of the pin can be read by reading the PDR4 register.

- Analog input mode setting

  Set a bit of the DDR4 register to "0", the bit corresponding to a pin of port 4 assigned for desired analog input, so that its output transistor is set to OFF and the pin is set at Hi-Z.

  Its output latch value can be read by reading the PDR4 register.

  Set the bit of the ADEN register of the A/D converter to "1", the bit corresponding to the analog input pin in use.

- Operation when a reset is performed

  When the CPU is reset, the bits of the PDR4 register are initialized to "1". Thus, the output transistors become OFF (input port mode) and the pins become Hi-Z.

- Operation in stop mode

  When the pin state setting bit of the standby control register (STBC: SPL) is set to "1" and when the stop mode is entered, the pin becomes Hi-Z because the output transistor is turned OFF regardless of the value existing on the DDR4 register in the bit position corresponding to the pin. Input remains fixed to prevent leaks by input open.

  Table 4.4-4 summarizes the operating modes of the pins of port 4.

Table 4.4-4 Operating Modes of Pins of Port 4

Pin name	Normal operation, sleep, stop (SPL = 0)	Stop (SPL = 1)	At a reset
P40/AN0 to P43/AN3	General-purpose I/O port may also serve I/O for peripherals	Hi-Z	Hi-Z

SPL : Pin state setting bit of standby control register (STBC: SPL)
Hi-Z: High impedance

4.5 Port 5

Port 5 is a general-purpose I/O port and may also serve the input/output for peripherals. The pins of this port can be used for peripherals or normal port function that can be selected according to the setting of the bit corresponding to the pin on a specific register.
This section explains the general-purpose I/O function of the port.
This section also describes port 5 concerning to the structure, pins, a block diagram of pins, and associated registers.

■ Structure of Port 5

Port 5 comprises the following four elements:

- General-purpose I/O pins (P50/PWM)
- Port 5 data register (PDR5)
- Port 5 data direction register (DDR5)
- Port 5 pull-up setting register (PUL5)

■ Pins of Port 5

Port 5 has one CMOS I/O pin.

Table 4.5-1 provides information on the pin of port 5.

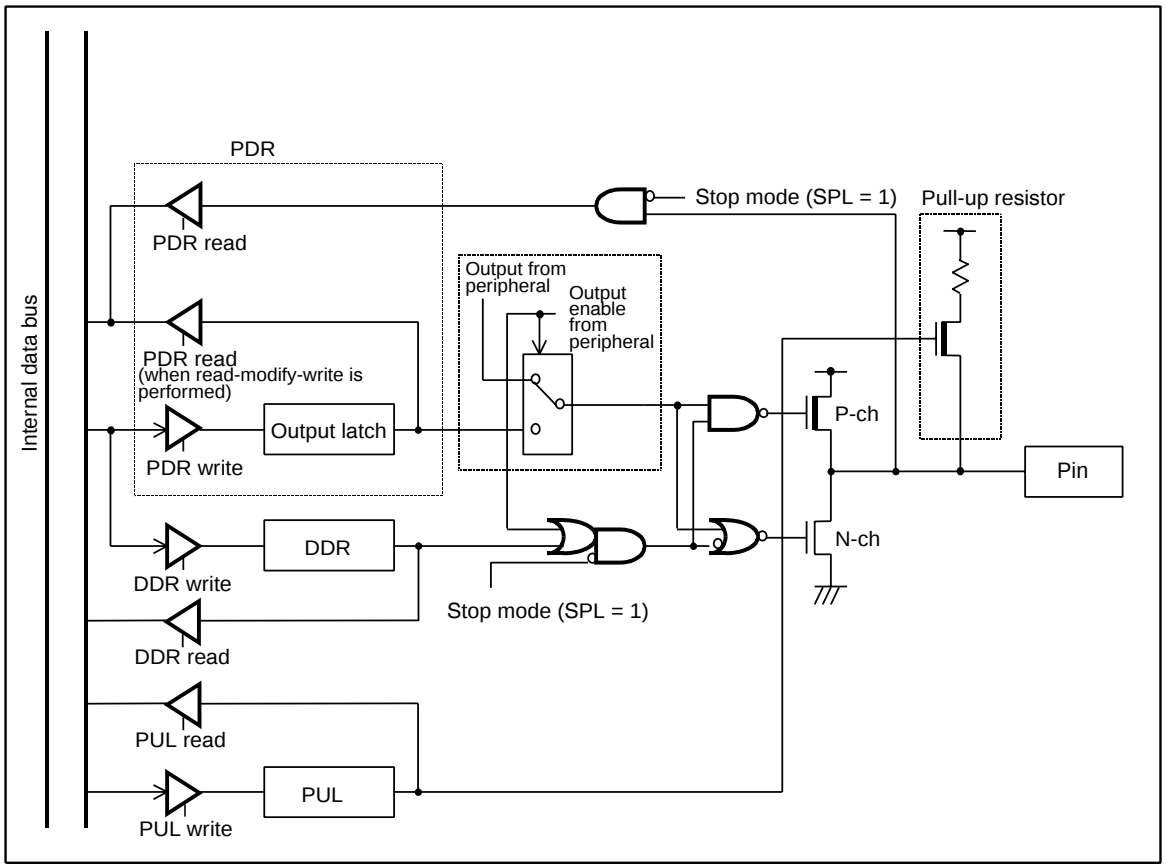
Table 4.5-1 Pin of Port 5

Port name	Pin name	Function	Peripherals for which the pin may serve	Input and output form		Circuit type
				Input	Output	
Port 5	P50/PWM	P50 general-purpose I/O	PWM (8-bit PWM/timer output)	CMOS	CMOS	E

For circuit type, see Section "1.7 Pin Functions Description " and "1.8 I/O Circuit Types ".

■ Block Diagram of Port 5

Figure 4.5-1 Block Diagram of Port 5



■ Registers of Port 5

The registers PDR5, DDR5, and PUL5 are associated with port 5.

One of the bits of these registers corresponds to one pin of port 5.

Table 4.5-2 tabulates the correspondence between the pin and a bit of the port 5 registers.

Table 4.5-2 Correspondence between the Pin and a Bit of the Port 5 Registers

Port name	Bits of associated registers and corresponding pins								
Port 5	PDR5, DDR5, PUL5	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Pin corresponding to bit	-	-	-	-	-	-	-	P50

4.5.1 Registers of Port 5 (PDR5, DDR5, PUL5)

This section describes the registers associated with port 5.

■ Functions of Port 5 Registers

- Port 5 data register (PDR5)

  The PDR5 register indicates the state of pins. For a pin set to function as an output port, the same value ("0" or "1") as held by the output latch can be read from this register. If the pin is set to function as an input port, however, its output latch value cannot be read from the register.

  Note:

  When a bit manipulation instruction (SETB, CLRB) is executed, the output latch values, not the value states of the pins, are read; thus, output latch values, excepting those for bits to be manipulated, do not change.

- Port 5 data direction register (DDR5)

  A bit of the DDR5 register sets the I/O direction of the pin corresponding to the bit.

  When the bit of the DDR5 register is set to "1", the pin functions as an output port. When the bit is set to "0", the pin functions as an input port.

- Setting the output from a peripheral enable

  If a peripheral with an output pin is used, set the output enable bit of the peripheral enable.

  As it is apparent from the block diagram, the pin in this mode serves output from the peripheral, thereby superseding its general-purpose port function.

  Because the output from the peripheral has priority, the values set on the PDR5 and DDR5 registers for the output pin used for the peripheral have no significance, regardless of the value output from the peripheral and the output enabled.

  Table 4.5-3 lists the functions of the port 5 registers.

Table 4.5-3 Functions of Port 5 Registers

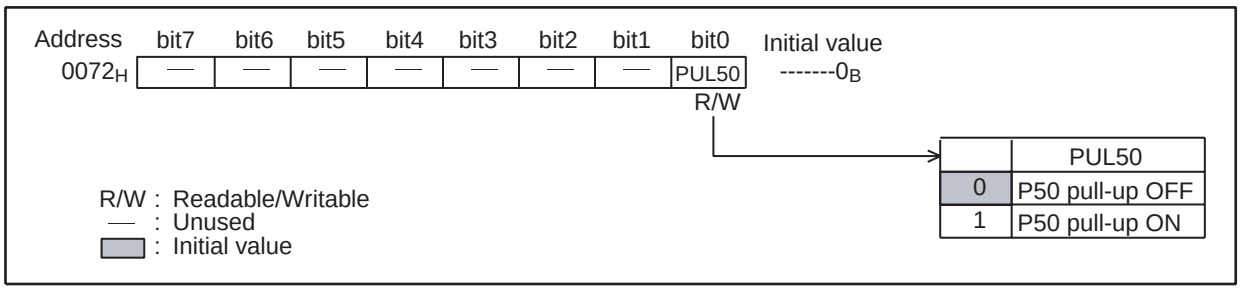
Register name	Data	When being read	When being written	Read/Write	Address	Initial value
Port 5 data register (PDR5)	0	Pin state is "L" level.	Output latch of "0" is set and "L" level is output to the pin in output port mode.	R/W	00012_H	-------X_B
	1	Pin state is "H" level.	Output latch of "1" is set and the pin in output port mode is set at Hi-Z.			
Port 5 data direction register (DDR5)	0	Input port pin	The pin is set to function as an input pin with output transistor operation disabled.	R/W	0013_H	-------0_B
	1	Output port pin	The pin is set to function as an output pin with output transistor operation enabled.			

R/W : Readable/Writable
X : Undefined

- Port 5 pull-up setting register (PUL5)

When the ON setting of the pull-up resistor is selected by using the pull-up setting register, the pin state will be "H" level (pull-up state) instead of Hi-Z during stop (SPL = 1). During a reset, however, the pull-up is invalid and the pin remains at Hi-Z.

Figure 4.5-2 Pull-up Setting Register (PUL5)



4.5.2 Operations of Port 5 Functions

This section describes the operation of port 5.

■ Operation of Port 5

- Operation in output port mode

  When "1" is written for a bit of the DDR5 register, the bit corresponding to the pin of port 5, the pin functions as an output port.

  In output port mode, the output transistor operation is enabled and the output latch data is output to the pin.

  Once data has been written into the PDR5 register, the written data is held in the output latch and output to the pin as it is.

  The value state of the pin can be read by reading the PDR5 register.

- Operation in input port mode

  When "0" is written for a bit of the DDR5 register, the bit corresponding to the pin of port 5, the pin functions as an input port.

  In input port mode, the output transistor is OFF and the pin state is Hi-Z.

  Once data has been written into the DDR5 register, the written data is held in the output latch but is not output to the pin.

  The value state of the pin can be read by reading the PDR5 register.

- Operation in mode enabling the output from a peripheral

  When the output enable bit for a peripheral is set enable, the corresponding pin is set to serve the output from the peripheral.

  Because the value state of the pin can be read from the PDR5 register even when the output from the peripheral is enabled, the value output from the peripheral can be read.

- Operation when a reset is performed

  When the CPU is reset, the bits of the DDR5 register are initialized to "0". Thus, the output transistor becomes OFF (input port mode) and the pin becomes Hi-Z.

  However, CPU resets do not initialize the PDR5 register. If the pin is used as an output port after the reset, reinitialize the PDR5 register to contain new output data in the bit position corresponding to the pin and then set the corresponding bit of the DDR5 register so that the pin will function as an output port.

- Operation in stop mode

  When the pin state setting bit of the standby control register (STBC: SPL) is set to "1" and when the stop mode is entered, the pin becomes Hi-Z because the output transistor is turned OFF regardless of the value existing on the DDR5 register in the bit position corresponding to the pin. Input remains fixed to prevent leaks by input open.

Table 4.5-4 summarizes the operating modes of the pin of port 5.

Table 4.5-4 Operating Modes of Pin of Port 5

Pin name	Normal operation, sleep, stop (SPL = 0)	Stop (SPL = 1)	At a reset
P50/PWM	General-purpose I/O port further may serve I/O for peripherals	Hi-Z	Hi-Z

SPL : Pin state setting bit of standby control register (STBC: SPL)
Hi-Z: High impedance

Note:

If the pull-up resistor is selected by using the pull-up setting register, the pin state will be "H" level (pull-up state) instead of Hi-Z in stop mode (SPL = 1). During a reset, however, the pull-up is invalid with the pin remaining at Hi-Z.

4.6 Port 6

Port 6 is a general-purpose I/O port.
This section describes the port function when operating as general-purpose I/O port.
This section also describes the structure, pins, the block diagram of pins, and associated registers of port 6.

■ Structure of Port 6

Port 6 comprises the following four elements:

- General-purpose I/O pins (input pins P61-P60 for MB89F202/F202RA)
- Port 6 data register (PDR6)
- Port 6 data direction register (DDR6, not used in MB89F202/F202RA)
- Port 6 pull-up setting register (PUL6)

■ Pins of Port 6

Port 6 has 2 I/O pins of CMOS input type (they are input only pins for MB89F202/F202RA).

Table 4.6-1 lists the pins of port 6.

Table 4.6-1 Pins of Port 6

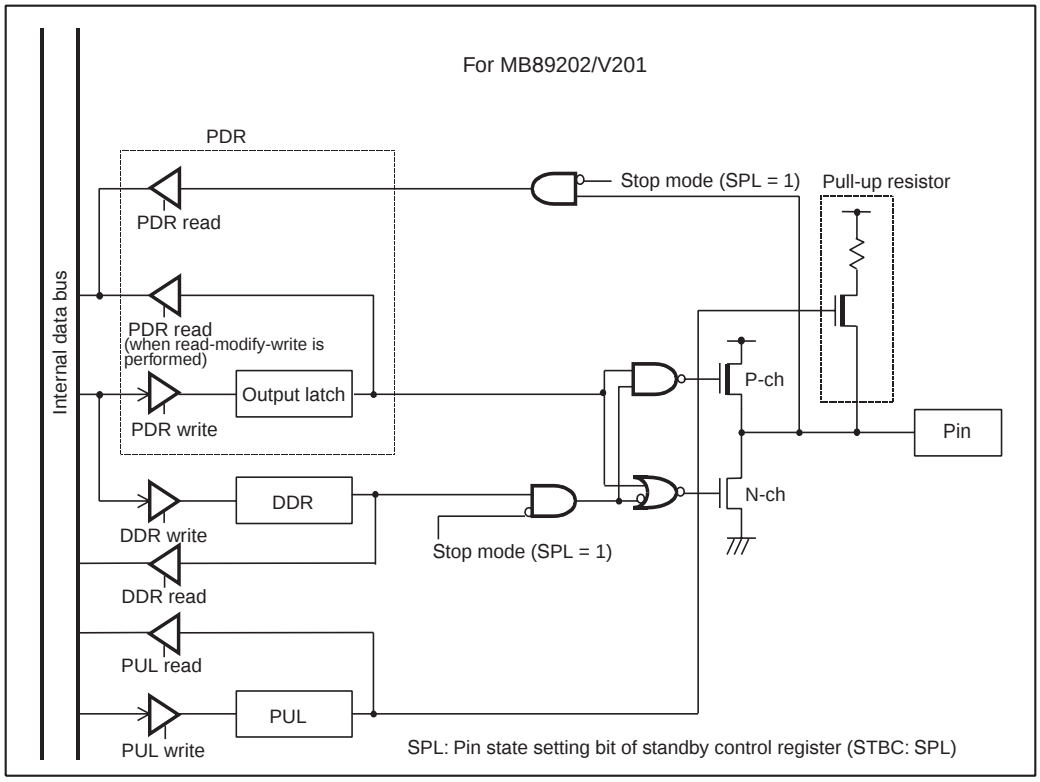
Port name	Pin name	Function	Peripherals for which a pin may serve	Input and output form		Circuit type
				Input	Output	
Port 6	P60	P60 general-purpose I/O*	-	CMOS	-	H / E
	P61	P61 general-purpose I/O*	-	CMOS	-	H / E

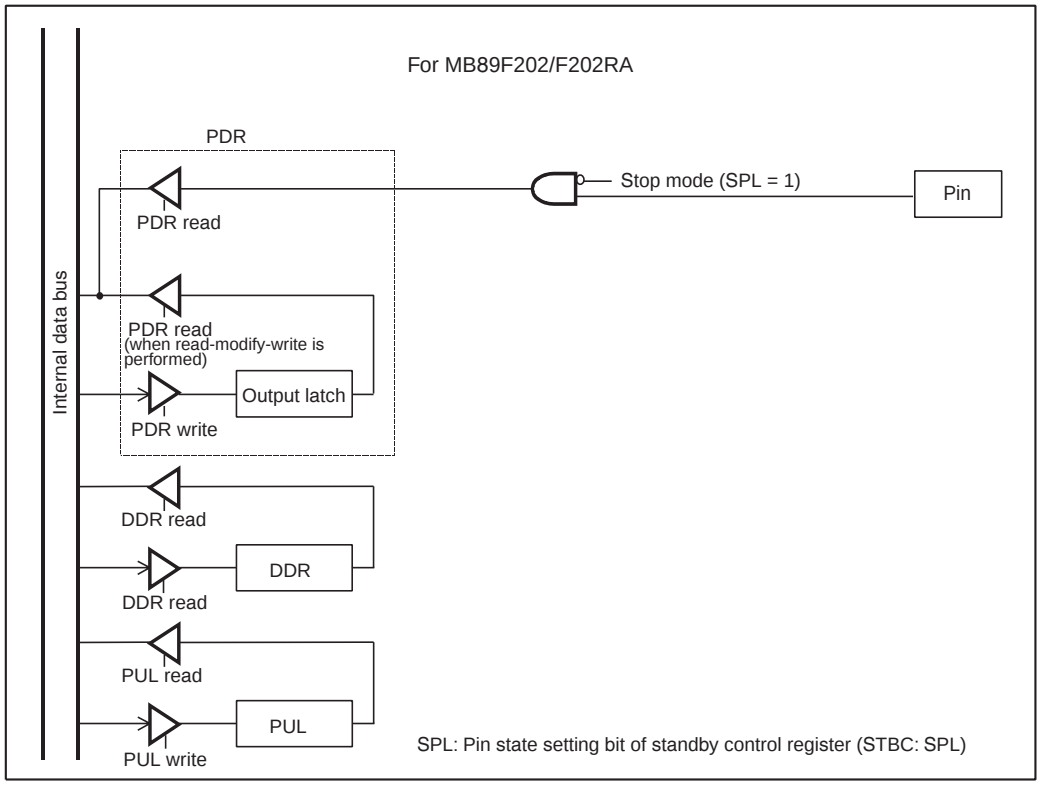
*: P61 and P60 are general-purpose input port for MB89F202/F202RA.

For circuit type, see Section "1.7 Pin Functions Description " and "1.8 I/O Circuit Types ".

■ Block Diagram of Port 6

Figure 4.6-1 Block Diagram of Port6





■ Registers PDR6, DDR6, and PUL6 of Port 6

Registers PDR6, DDR6, and PUL6 are associated with port 6.

The bits of these registers correspond to the pins of port 6 in one-to-one correspondence.

Table 4.6-2 tabulates the correspondence between the pins and the bits of the port 6 registers.

Table 4.6-2 Correspondence between the Pins and the Bits of Port 6 Registers

Port name	Bits of associated registers and corresponding pins								
Port 6	PDR6, DDR6, PUL6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Pin corresponding to bit	-	-	-	-	-	-	P61*	P60*

*: DDR control is not used for this bit in MB89F202/F202RA.

4.6.1 Registers of Port 6 (PDR6, DDR6, PUL6)

This section describes the registers associated with port 6.

■ Functions of Port 6 Registers

- Port 6 data register

The PDR6 register indicates the state of the output latch. For a pin set to function as an output port, the same value ("0" or "1") as the value state of the output pin can be read from this register. If the pin is set to function as an input port, however, its output latch value cannot be read from the register.

Note:

When a bit manipulation instruction (SETB, CLRB) is executed, the output latch values, not the value states of the pins, are read; thus, output latch values, other than those for bits to be manipulated, do not change.

- Port 6 data direction register (DDR6 for P60, P61)

The DDR6 register sets the I/O direction of each pin per bit.

When a bit of the DDR6 corresponding to a pin of port 6 is set to "1", the pin functions as an output port. When the bit is set to "0", the pin functions as an input port.

Table 4.6-3 lists the functions of the port 6 registers.

Table 4.6-3 Functions of Port 6 Registers

Register name	Data	When being read	When being written	Read/ Write	Address	Initial value
Port 6 data register (PDR6)	0	Pin state is "L" level.	Output latch of "0" is set and "L" level is output to the pin in output port mode.	R/W	0060_H	------XX_B
	1	Pin state is "H" level.	Output latch of "1" is set and "H" level is output to the pin in output port mode.			
Port 6 data direction register (DDR6)	0	Input port	Output transistor operation is disabled and the pin is set to serve as an input pin.	R/W	0061_H	------00_B
	1	Output port	Output transistor operation is enabled and the pin is set to serve as output pin.			

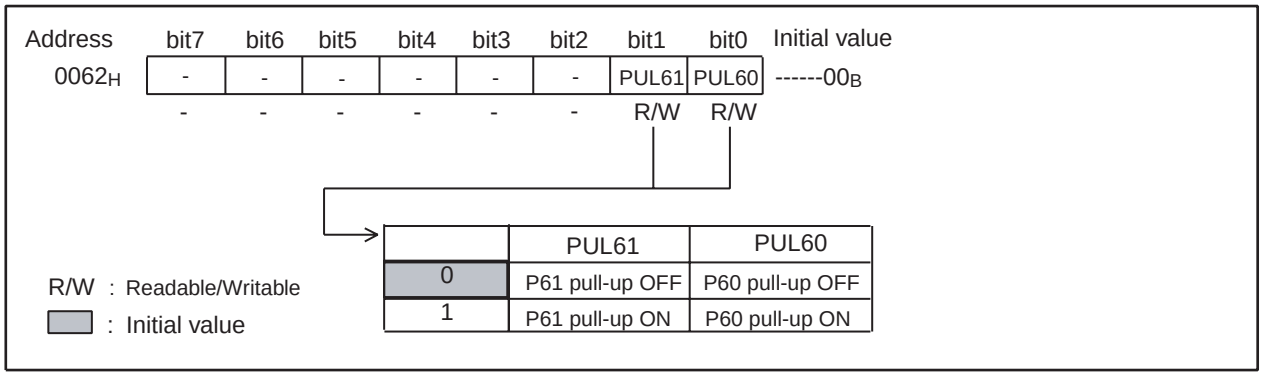
R/W : Readable/Writable
X : Undefined

● Port 6 pull-up setting register (PUL6)

The bits of the pull-up setting register correspond to the pins of port 6 in one-to-one correspondence. When the pull-up resistor is selected by using the pull-up setting register, the pin will be at "H" level (pull-up state) instead of Hi-Z during stop (SPL = 1). During a reset, however, the pull-up is invalid and the pin remains at Hi-Z.

Figure 4.6-2 shows the pull-up resistor settings assigned to the values of the bits of the port 6 pull-up setting register.

Figure 4.6-2 Pull-up Resistor Settings (PUL6)



4.6.2 Operations of Port 6 Functions

This section describes the operation of port 6.

■ Operation of Port 6

- Operation in output port mode

  When "1" is written for a bit of the DDR6 register, the bit corresponding to a pin of port 6, the pin functions as an output port.

  In output port mode, the output transistor operation is enabled and the output latch data is output to the pin.

  Once data has been written into the PDR6 register, the written data is held in the output latch and output to the pin as it is.

  The value state of the pin can be read by reading the PDR6 register.

- Operation in input port mode

  When "0" is written for a bit of the DDR6 register, the bit corresponding to a pin of port 6, the pin functions as an input port.

  In input port mode, the output transistor is OFF and the pin status is Hi-Z.

  Once data has been written into the PDR6 register, the written data is held in the output latch but is not output to the pin.

  The value state of the pin can be read by reading the PDR6 register.

- Operation when a reset is performed

  When the CPU is reset, the bits of the DDR6 register are initialized to "0". Thus, all output transistors become OFF and the pins become Hi-Z.

  However, CPU resets do not initialize the PDR6 register. If a pin is used as an output port after the reset, reinitialize the PDR6 register to contain new output data in the bit position corresponding to the pin and then set the corresponding bit of the DDR6 register so that the pin will function as an output port.

- Operation in stop mode

  When the pin state setting bit of the standby control register (STBC: SPL) is "1" and when the stop mode is entered, the output transistor is turned OFF and the pin becomes Hi-Z because the output transistor is forcibly turned OFF without respect to the value existing on the DDR6 register in the bit position corresponding to the pin.

  Input remains fixed to prevent leaks by input open.

  Table 4.6-4 summarizes the operating modes of the pins of port 6.

Table 4.6-4 Operating Modes of Pins of Port 6

Pin name	Normal operation, sleep, stop (SPL = 0)	Stop (SPL = 1)	At a reset
P60, P61	General-purpose I/O port	Hi-Z	Hi-Z

Note:

When the pull-up resistor is selected by using the pull-up setting register, the pin state will be "H" level instead of Hi-Z in stop mode (SPL = 1). During a reset, however, the pull-up is invalid with the pin remaining at Hi-Z.

4.7 Port 7

Port 7 is a general-purpose I/O port.
This section describes the port function when operating as general-purpose I/O port.
This section also describes the port structure, pins, the pin block diagram associated registers of port 7.

■ Structure of Port 7

Port 7 comprises the following four elements:

- General-purpose I/O pin (P70 to P72)
- Port 7 data register (PDR7)
- Port 7 data direction register (DDR7)
- Port 7 pull-up setting register (PUL7)

■ Pins of Port 7

Port 7 has 3 CMOS I/O pin.

Table 4.7-1 lists the pins of port 7.

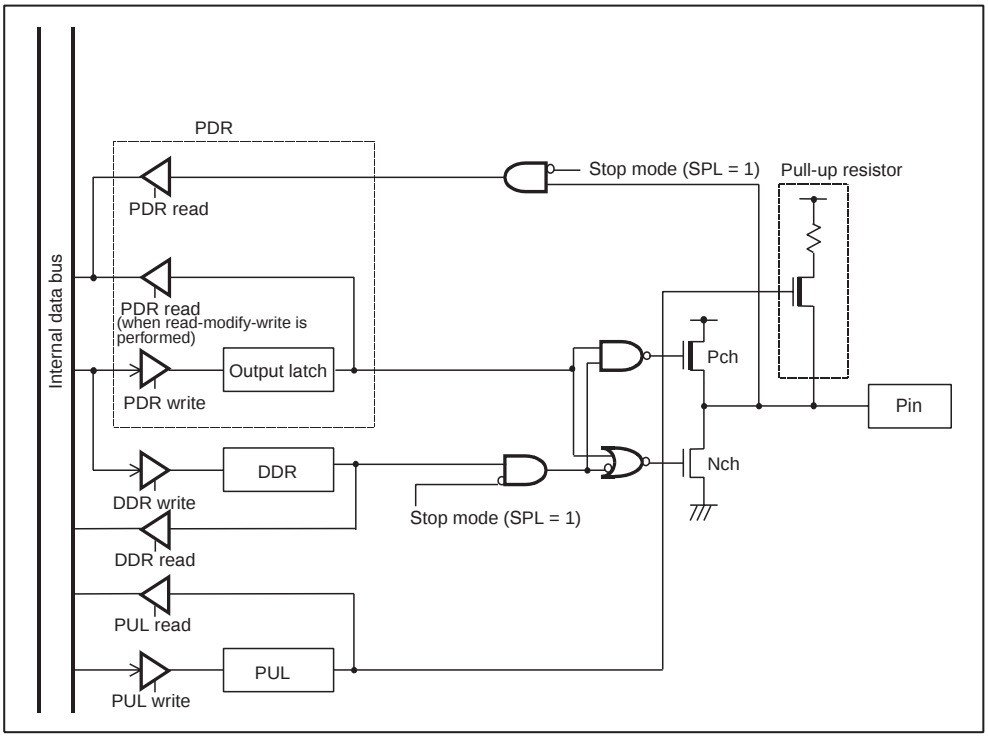
Table 4.7-1 Pins of Port 7

Port name	Pin name	Function	Input and output form		Circuit type
			Input	Output	
Port 7	P70 to P72	General-purpose I/O	CMOS	CMOS	E

For circuit type, see Section "1.7 Pin Functions Description " and "1.8 I/O Circuit Types ".

■ Block Diagram of Port 7

Figure 4.7-1 Block Diagram of Port 7



■ Registers PDR7, DDR7, and PUL7 of Port 7

Registers PDR7, DDR7, and PUL7 are associated with port 7.

The bits of these registers correspond to the pins of port 7 in one-to-one correspondence.

Table 4.7-2 tabulates the correspondence between the pins and the bits of the port 7 registers.

Table 4.7-2 Correspondence between the Pins and the Bits of the Port 7 Registers

Port name	Bits of associated registers and corresponding pins								
Port 7	PDR7, DDR7, PUL7	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	Pin corresponding to bit	-	-	-	-	-	P72	P71	P70

4.7.1 Registers of Port 7 (PDR7, DDR7, PUL7)

This section describes the registers associated with port 7.

■ Functions of Port 7 Registers

- Port 7 data register (PDR7)

  The PDR7 register indicates the state of the output latch. For a pin set to function as an output port, the same value ("0" or "1") as the value state of the output pin can be read from this register. If the pin is set to function as an input port, however, its output latch value cannot be read from the register.

  Note:

  When a bit manipulation instruction (SETB, CLRB) is executed, the output latch values, not the value states of the pins, are read; thus, output latch values, other than those for bits to be manipulated, do not change.

- Port 7 data direction register (DDR7)

  The DDR7 register sets the I/O direction of each pin per bit.

  When a bit of the DDR7 corresponding to a pin of port 7 is set to "1", the pin functions as an output port. When the bit is set to "0", the pin functions as an input port.

  Table 4.7-3 lists the functions of the port 7 registers.

Table 4.7-3 Functions of Port 7 Registers

Register name	Data	When being read	When being written	Read/ Write	Address	Initial value
Port 7 data register (PDR7)	0	Pin state is "L" level.	Output latch of "0" is set and "L" level is output to the pin in output port mode.	R/W	0063_H	-----XXX_B
	1	Pin state is "H" level.	Output latch of "1" is set and "H" level is output to the pin in output port mode.			
Port 7 data direction register (DDR7)	0	Input port	Output transistor operation is disabled and the pin is set to serve as an input pin.	R/W	0064_H	-----000_B
	1	Output port	Output transistor operation is enabled and the pin is set to serve as output pin.			

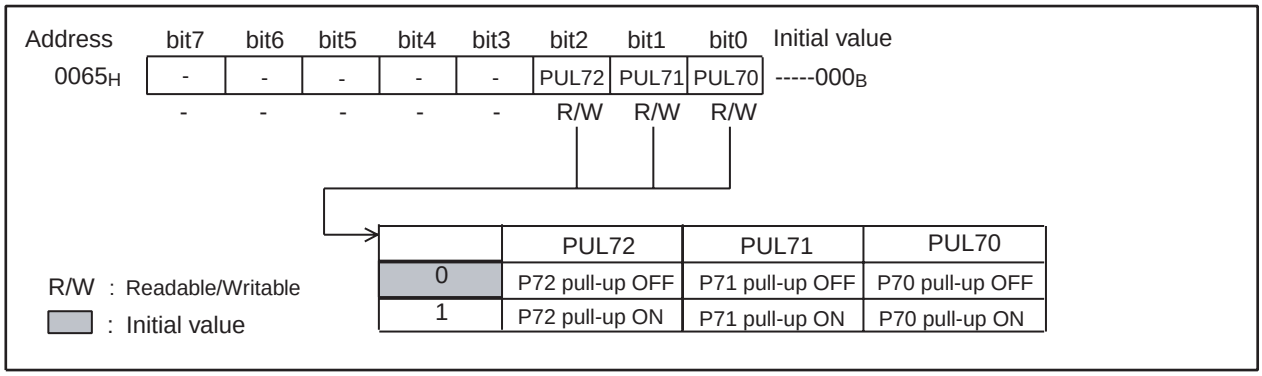
R/W : Readable and Writable
X : Undefined

● Port 7 pull-up setting register (PUL7)

The bits of the pull-up setting register correspond to the pins of port 7 in one-to-one correspondence. When the pull-up resistor is selected by using the pull-up setting register, the pin will be at "H" level (pull-up state) instead of Hi-Z during stop (SPL = 1). During a reset, however, the pull-up is invalid and the pin remains at Hi-Z.

Figure 4.7-2 shows the pull-up resistor settings assigned to the values of the bits of the port 7 pull-up register.

Figure 4.7-2 Pull-up Resistor Settings (PUL7)



4.7.2 Operations of Port 7 Functions

This section describes the operation of port 7.

■ Operation of Port 7

- Operation in output port mode

  When "1" is written for a bit of the DDR7 register, the bit corresponding to a pin of port 7, the pin functions as an output port.

  In output port mode, the output transistor operation is enabled and the output latch data is output to the pin.

  Once data has been written into the PDR7 register, the written data is held in the output latch and output to the pin as it is.

  The value state of the pin can be read by reading the PDR7 register.

- Operation in input port mode

  When "0" is written for a bit of the DDR7 register, the bit corresponding to a pin of port 7, the pin functions as an input port.

  In input port mode, the output transistor is OFF and the pin status is Hi-Z.

  Once data has been written into the PDR7 register, the written data is held in the output latch but is not output to the pin.

  The value state of the pin can be read by reading the PDR7 register.

- Operation when a reset is performed

  When the CPU is reset, the bits of the DDR7 register are initialized to "0". Thus, all output transistors become OFF and the pins become Hi-Z.

  However, CPU resets do not initialize the PDR7 register. If a pin is used as an output port after the reset, reinitialize the PDR7 register to contain new output data in the bit position corresponding to the pin and then set the corresponding bit of the DDR7 register so that the pin will function as an output port.

- Operation in stop mode

  When the pin state setting bit of the standby control register (STBC: SPL) is "1" and when the stop mode is entered, the output transistor is turned OFF and the pin becomes Hi-Z because the output transistor is forcibly turned OFF without respect to the value existing on the DDR7 register in the bit position corresponding to the pin.

  Input remains fixed to prevent leaks by input open.

Table 4.7-4 summarizes the operating modes of the pins of port 7.

Table 4.7-4 Operating Modes of Pins of Port 7

Pin name	Normal operation, sleep, stop (SPL = 0)	Stop (SPL = 1)	At a reset
P70 to P72	General-purpose I/O port	Hi-Z	Hi-Z

Note:

When the pull-up resistor is selected by using the pull-up setting register, the pin state will be "H" level instead of Hi-Z in stop mode (SPL = 1). During a reset, however, the pull-up is invalid with the pin remaining at Hi-Z.

4.8 Programming Example of I/O Port

This section provides an example of programming with I/O ports.

■ I/O Port Programming Example

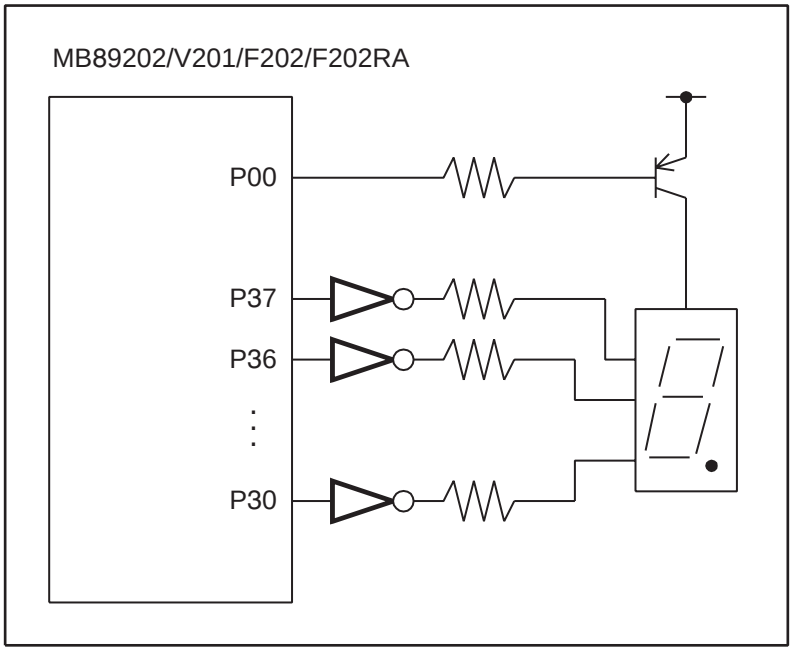
- Processing specification

  Ports 0 and 3 are used to light all seven segments of LED (eight segments if the decimal point is included).

  Pin P00 is connected to the anode common pin of LED and pins P30 to P37 are connected to the pins of the segments.

  Figure 4.8-1 provides an example of the pins and the 8-segment LED connected.

Figure 4.8-1 Example of the Pins and the 8-segment LED Connected



- Coding example

```
PDR0   EQU   0000H                ;  Address of port 0 data register
DDR0   EQU   0001H                ;  Address of port 0 data direction register
PDR3   EQU   000CH                ;  Address of port 3 data register
DDR3   EQU   000DH                ;  Address of port 3 data direction register
;-----------------------------Main program-----------------------------------------------------------
       CSEG                       ;  [CODE SEGMENT]
        :
       CLRB  PDR0:0               ;  Set P00 at "L" level.
       MOV   PDR3,#11111111B      ;  Set all pins of port 3 at "H" level.
       MOV   DDR0,#11111111B      ;  Set P00 to function as an output port by coding
                                     #XXXXXXX1B.
       MOV   DDR3,#11111111B      ;  Set all bits of DDR3 such that all pins of port 3 function
                                     as an output port.
        :
       ENDS
;-----------------------------------------------------------------------------------------------------
       END
```

CHAPTER 5

TIME-BASE TIMER

This chapter describes the functions and operations of the time-base timer.

5.1 Overview of Time-base Timer

The time-base timer functions as an interval timer. The time-base timer is a 21-bit free-run counter that counts up in synchronization with the internal count clock (at the oscillation frequency divided by 2). The timer also has an interval timer function to select one of four time intervals. In addition, it provides timer output for oscillation stabilization time and an operation clock for the watchdog timer.
The time-base timer stops operating in modes in which oscillation stops.

■ Interval Timer Function

The interval timer function is used to continuously generate interrupts at specified intervals.

- An interrupt occurs when the interval timer bits of the time-base timer counter overflow.
- One of four time intervals can be selected by setting the interval timer bits.

Table 5.1-1 lists the time intervals for the time-base timer.

Table 5.1-1 Time Intervals for Time-base Timer

Internal count clock cycle	Time interval
$2/F_{CH}$ (0.16 μs)	$2^{13}/F_{CH}$ (Approximately 0.66 ms)
	$2^{15}/F_{CH}$ (Approximately 2.6 ms)
	$2^{18}/F_{CH}$ (Approximately 21.0 ms)
	$2^{22}/F_{CH}$ (Approximately 335.5 ms)

F_{CH}: Oscillation frequency
The values enclosed in parentheses are time intervals when the oscillation frequency is 12.5 MHz.

■ Clock Supply Function

The clock supply function is used to provide one of three timer outputs for the oscillation stabilization wait time and the operation clock for the resource function.

Table 5.1-2 lists cycles of the clock that the time-base timer supplies to peripherals.

Table 5.1-2 Clock Cycles Supplied by Time-base Timer (1/2)

Clock supplied to	Clock cycle	Remarks
Oscillation stabilization time	$2^{14}/F_{CH}$ (Approximately 1.31 ms)	Selected by the oscillation stabilization time selection bits (SYCC: WT1, WT0) of the clock control section system clock control register.
	$2^{17}/F_{CH}$ (Approximately 10.49 ms)	
	$2^{18}/F_{CH}$ (Approximately 20.97 ms)	

Table 5.1-2 Clock Cycles Supplied by Time-base Timer (2/2)

Clock supplied to	Clock cycle	Remarks
Watchdog timer	$2^{22}/F_{CH}$ (Approximately 335.5 ms)	Watchdog timer count up clock
A/D converter	$2^{8}/F_{CH}$ (Approximately 20.5 μs)	Continuous activation clock

F_{CH}: Oscillation frequency
The values enclosed in parentheses are time intervals when the oscillation frequency is 12.5 MHz.

Note:

Because oscillation cycles vary immediately after oscillation starts, the oscillation stabilization wait time is listed for reference.

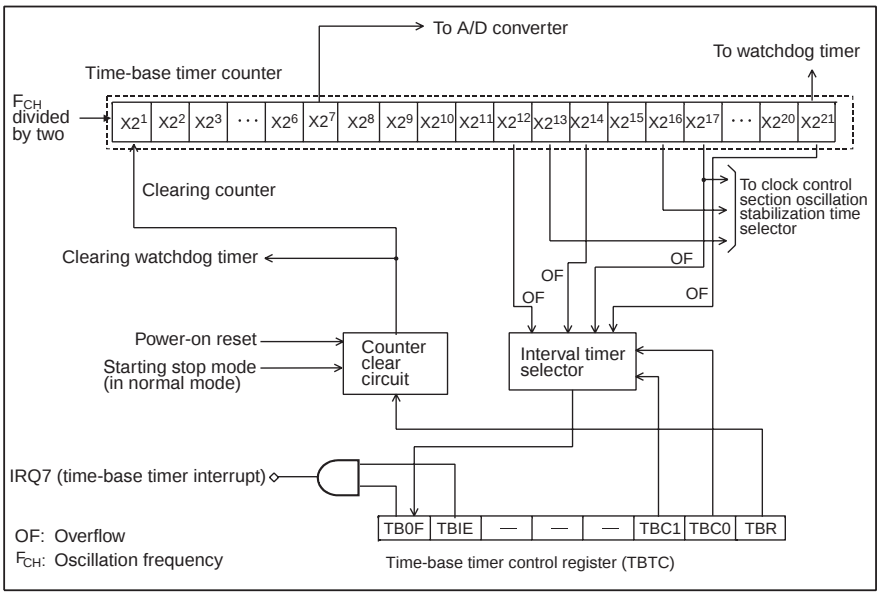
5.2 Configuration of Time-base Timer

The time-base timer consists of the following four function blocks.

- **Time-base counter**
- **Counter clear circuit**
- **Interval timer selector**
- **Time-base timer control register (TBTC)**

■ Block Diagram of Time-base Timer

Figure 5.2-1 Block Diagram of Time-base Timer



● Time-base timer counter

A 21-bit up counter that accepts the oscillation frequency divided by two as the count clock and stops operating when oscillation stops.

● Counter clear circuit

Clears the counter when the TBTC register is set (TBR = 0), stop mode is entered (STBC: STP = 1), or a power-on reset occurs.

● Interval timer selector

Selects 1 bit for the interval timer from four bits in the time-base counter. When the specified bit overflows, an interrupt occurs.

● Time-base timer control register (TBTC)

Selects a time interval, clears the counter, controls interrupts, or checks the status.

5.3 Time-base Timer Control Register (TBTC)

The time-base timer control register (TBTC) selects a time interval, clears the counter, controls interrupts, or checks the status.

■ Time-base Timer Control Register (TBTC)

Figure 5.3-1 Time-base Timer Control Register (TBTC)

Address: 000A$_H$

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TBOF	TBIE	—	—	—	TBC1	TBC0	TBR
R/W	R/W				R/W	R/W	R/W

Initial value: 00---000$_B$

TBR	Time-base timer initialization bit	
	Read	Write
0	—	The time-base timer counter is cleared.
1	"1" is always read.	Nothing is changed and affected.

TBC1	TBC0	Time interval selection 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}: Oscillation frequency

TBIE	Interrupt request enable bit
0	The interrupt request output is disabled.
1	The interrupt request output is enabled.

TBOF	Overflow interrupt request flag bit	
	Read	Write
0	The specified bit has not overflowed.	This bit is cleared.
1	The specified bit has overflowed.	Nothing is changed and affected.

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

Table 5.3-1 Explanation of Functions of Each Bit in Time-base Timer Control Register (TBTC)

Bit name		Description
bit7	TBOF: Overflow interrupt request flag bit	• This bit is set to "1" when the specified bit of the time-base timer counter overflows. • An interrupt request is sent when this bit and the interrupt request enable bit (TBIE) are both "1". • While this bit is written, it is cleared when "0" is specified, and nothing is changed and affected when "1" is specified.
bit6	TBIE: Interrupt request enable bit	This bit enables or disables an interrupt request to be output to the CPU. An interrupt request is output when this bit and the overflow interrupt request flag bit (TBOF) are both "1".
bit5 to bit3	Unused bits	• These bits are undefined when they are read. • Nothing is affected when they are written.
bit2, bit1	TBC1, TBC0: Time interval selection bits	• These bits specify a time interval for the interval timer. • The interval timer bits of the time-base timer counter are specified. • One of four time intervals can be selected.
bit0	TBR: Time-base timer initialization bit	• This bit clears the time-base timer counter. • The counter is cleared to 000000_H when "0" is written to this bit, nothing is changed and affected when "1" is written. **Note:** This bit is always "1" at the beginning of reading.

5.4 Interrupt of Time-base Timer

The time-base timer counter generates an interrupt when the specified bit of the counter overflows (interval timer function).

■ Interrupts when the Interval Timer Function is Enabled

The counter counts up with the internal count clock. When the specified interval timer bit overflows, the overflow interrupt request flag bit (TBTC: TBOF) is set to "1". Then if the interrupt request enable bit is enabled (TBTC: TBIE = 1), an interrupt request (IRQ7) is sent to the CPU. When this occurs, use the interrupt handling routine and set the TBOF bit to "0" to clear the interrupt request. The TBOF bit is set to "1" when the specified bit overflows regardless of the value of the TBIE bit.

Note:

When the interrupt request is allowed to be output (TBIE = 1) after a reset is released, clear the TBOF bit (TBOF = 0) at the same time.

Note:

- An interrupt request is generated immediately after the TBIE bit is set from 0 (disable) to 1 (enable) if the TBOF bit is "1".
- When the counter is cleared (TBTC: TBR = 0) and the specified bit overflows at the same time, the TBOF bit is not set.

■ Oscillation Stabilization Time and Time-base Timer Interrupts

If a time interval is set the time shorter than the oscillation stabilization time, the interval interrupt request (TBTC: TBOF = 1) is generated from the time-base timer upon the start of normal mode. In this case, interrupts from the time-base timer must be disabled (TBTC: TBIE = 0) when switching to stop mode in which an oscillation is stopped.

■ Register and Vector Table Related to Interrupts from Time-base Timer

Table 5.4-1 Register and Vector Table Related to Time-base Timer Interrupts

Interrupt name	Interrupt level setting register			Address of vector table	
	Register	Bits to be set		High-order	Low-order
IRQ7	ILR2 ($007C_H$)	L71 (bit7)	L70 (bit6)	$FFEC_H$	$FFED_H$

See Section "3.4.2 Steps in the Interrupt Operation " for details on interrupt operations.

5.5 Operations of Time-base Timer Functions

The time-base timer functions as an interval timer or supplies clocks to some peripherals.

■ Operations of Interval Timer Function (Time-base Timer)

To use as an interval timer, the settings shown below must be made.

Figure 5.5-1 Setting Interval Timer Function

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TBTC	TBOF	TBIE	—	—	—	TBC1	TBC0	TBR
	0	1				◎	◎	0

◎ : Used bit
1 : Set to "1"
0 : Set to "0"

The counter in the time-base timer continues to count up in synchronization with the internal count clock (at the oscillation frequency divided by two) as long as the clock oscillates.

The counter counts from "0" upon being cleared (TBR = 0). When the interval timer bit overflows, the overflow interrupt request flag bit (TBOF) is set to "1". In other words, interrupts are generated at specified intervals, starting from when the counter is cleared.

■ Operations of Clock Supply Function

The time-base timer is often used to make oscillation stabilization wait time. The oscillation stabilization time is measured from when the time-base timer counter is cleared to when the oscillation stabilization bit overflows. One of three oscillation stabilization time can be selected by the oscillation stabilization time selection bits of the system clock control register (SYCC: WT1, WT0).

The time-base timer supplies clocks to the watchdog timer and A/D converter. Clearing the time-base timer counter affects the operation of continuous activation cycles. In addition, when the time-base timer is cleared, the counter in the watchdog timer is also cleared.

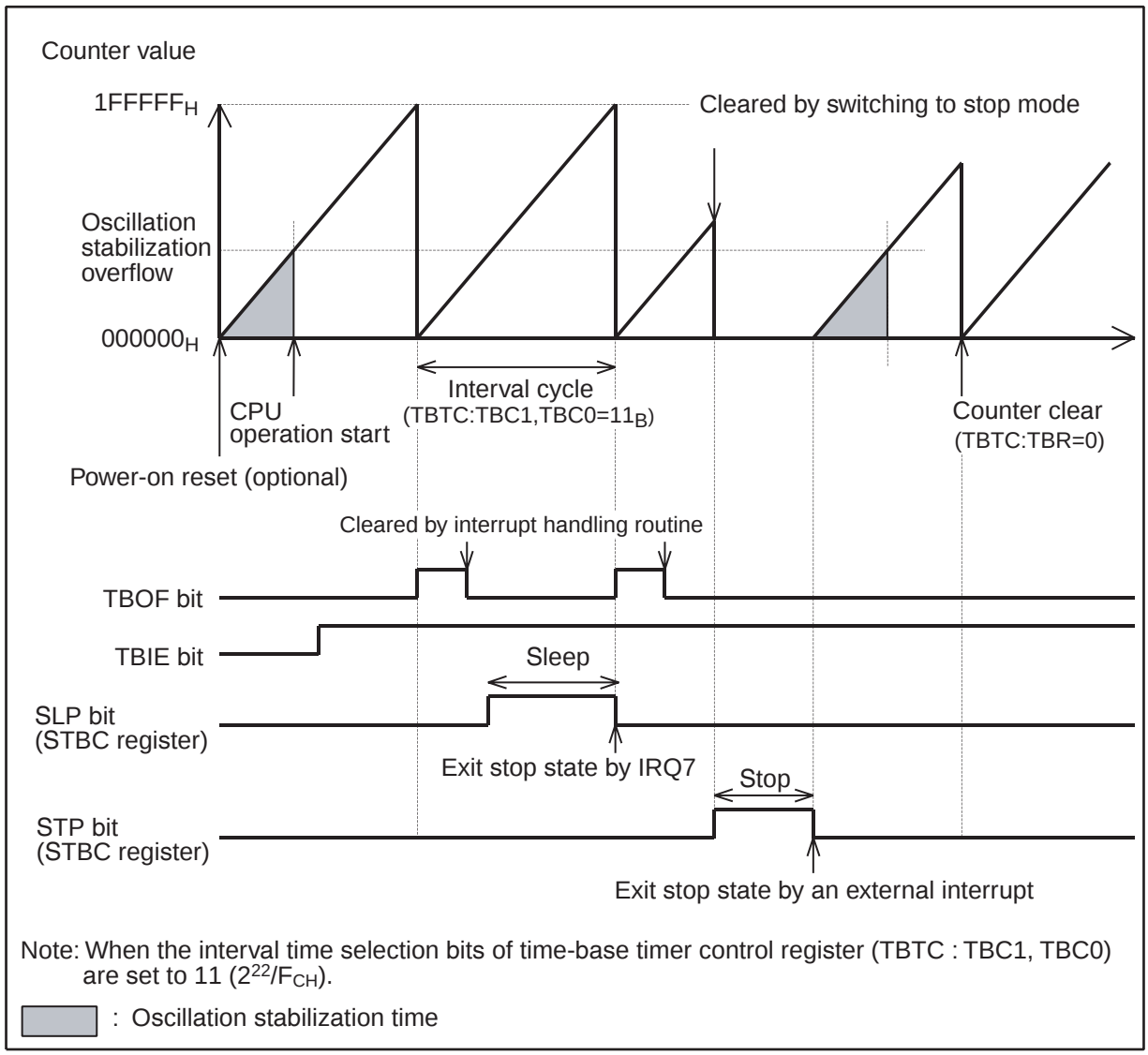
■ Operations of Time-base Timer

Figure 5.5-2 shows the operation of the time-base timer when:

- power-on reset occurs.
- sleep mode is entered while the interval timer function is being performed in normal mode.
- stop mode is entered.
- a counter clear request is generated.

In stop mode, the time-base timer is cleared and stops operating. When returning from stop mode, the time-base timer counts the oscillation stabilization time.

Figure 5.5-2 Operations of Time-base Timer



Note: When the interval time selection bits of time-base timer control register (TBTC : TBC1, TBC0) are set to 11 ($2^{22}/F_{CH}$).

: Oscillation stabilization time

5.6 Notes on Using Time-base Timer

Notes on using the time-base timer are shown below.

■ Notes on Using Time-base Timer

- Notes on using programs to set time-base timer

  When the interrupt request flag bit (TBTC: TBOF) is "1" and the interrupt request enable bit is enabled (TBTC: TBIE = 1), a return from interrupt handling is not possible. The TBOF bit must be cleared.

- Clearing time-base timer

  The time-base timer is cleared when the time-base timer initialization bit is set to 0 (TBTC: TBR = 0) or when the oscillation stabilization time is required. Because the time-base timer is used as the count clock for the watchdog timer, clearing the time-base timer also clears the watchdog timer.

- Using time-base timer as oscillation stabilization time timer

  Oscillation has not yet started in stop mode or when the power is turned on. Therefore, the time-base timer makes oscillation stabilization wait time after the resonator starts operating.

  The appropriate oscillation stabilization time must be selected according to the type of resonator connected to the resonator (clock generator).

  See Section "3.6.1 Clock Generator ".

- Notes on peripheral functions the time-base timer supplies to the clock

  When entering the modes in which oscillation stops, the counter is cleared and the time-base timer stops operating. The clock from the time-base timer may have a shorter "H" level period or longer "L" level period (up to half the clock cycle) when the counter of the time-base timer is cleared because the clock starts operating from the initial state. The clock for the watchdog timer also starts operating from the initial state, but the watchdog timer operates at a normal cycle because the watchdog timer counter is cleared at the same time.

5.7 Program Example for Time-base Timer

Programming examples for the time-base timer are shown below.

■ Programming Examples for Time-base Timer

- Processing specification

  Repeatedly generate an interval timer interrupt at intervals of $2^{18}/F_{CH}$ (F_{CH}: oscillation frequency). The time interval is approximately 21.0 ms (operating at 12.5 MHz).

- Coding examples

```
TBTC    EQU     0000AH              ;  Address of time-base timer control register

TBOF    EQU     TBTC:7              ;  Definition of 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 interrupt vector
INT_V   ENDS
;-----------------------Main program-----------------------------------------------------------------------
        CSEG                        ;  [CODE SEGMENT]
                                    ;  Stack pointer (SP) or other registers are assumed to
                                       have been initialized.
         :
        CLRI                        ;  Interrupt disable
        MOV     ILR2,#01111111B     ;  Setting interrupt level (level 1)
        MOV     TBTC,#01000100B     ;  Clearing interrupt request flag, enabling interrupt
                                       request output, selecting 2^18 /F_CH, and clearing
                                       time-base timer
        SETI                        ;  Interrupt enable
          :
;-----------------------Interrupt processing routine-------------------------------------------------------
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 Watchdog Timer

The watchdog timer is a 1-bit counter that uses output from the time-base timer, based on oscillation frequency, as the count clock. The watchdog timer resets the CPU when not cleared within a specified period after activation.

■ Watchdog Timer Function

The watchdog timer is a counter for preventing programs from hanging up. The timer must be cleared at specified intervals after being activated. If the timer is not cleared within a specified period of time because, for example, a program goes into an endless loop, the timer sends to the CPU a watchdog reset having a period of four instruction cycles.

The watchdog timer uses the output from the time-base timer as the count clock.

The time intervals for the watchdog timer are listed in Table 6.1-1 . When the watchdog timer is not cleared, a watchdog reset occurs following the time between the minimum time interval and the maximum time interval. The counter must be cleared before the time of the minimum time interval.

Table 6.1-1 Watchdog Timer Time Intervals

	Count clock
	Time-base timer output (Oscillation frequency: 12.5 MHz)
Minimum time interval	Approximately 335.5 ms *
Maximum time interval	Approximately 671.0 ms

*: (number of counts of time-base timer (2^{22})) × (oscillation frequency (F_{CH}) divided by 2)

See Section "6.4 Operations of Watchdog Timer Functions " for details on the maximum and minimum time intervals of the watchdog timer.

Notes:

- The watchdog timer counter is cleared when the time-base counter is cleared (TBTC : TBR = 0) while output from the time-base timer is selected. Therefore, if the time-base timer counter supplying the count clock is cleared repeatedly within the time interval of the watchdog timer, the watchdog timer does not function correctly.
- When switching to sleep or stop mode, the watchdog timer counter is cleared and stops operating until returning to normal operation (RUN state).

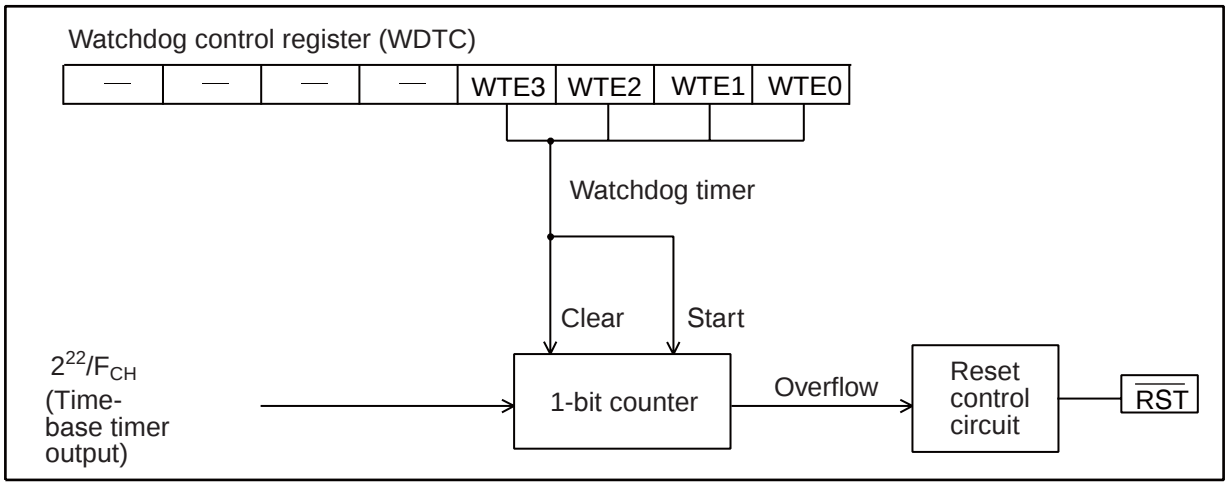
6.2 Configuration of Watchdog Timer

The watchdog timer consists of the following four function blocks.

- **Watchdog timer counter**
- **Reset control circuit**
- **Counter clear control circuit**
- **Watchdog control register (WDTC)**

■ Block Diagram of Watchdog Timer

Figure 6.2-1 Block Diagram of Watchdog Timer



● Watchdog timer counter (1-bit counter)

A 1-bit counter that operates by accepting output from the time-base timer as the count clock.

● Reset control circuit

Sends the reset signal to the CPU when the watchdog timer counter overflows.

● Counter clear control circuit

Controls the clearing and stopping of the watchdog timer counter.

● Watchdog control register (WDTC)

Activates and clears the watchdog timer counter. Because this register is write-only, bit manipulation instructions cannot be used.

6.3 Watchdog Control Register (WDTC)

The watchdog control register (WDTC) activates and clears the watchdog timer.

■ Watchdog Control Register (WDTC)

Figure 6.3-1 Watchdog Control Register (WDTC)

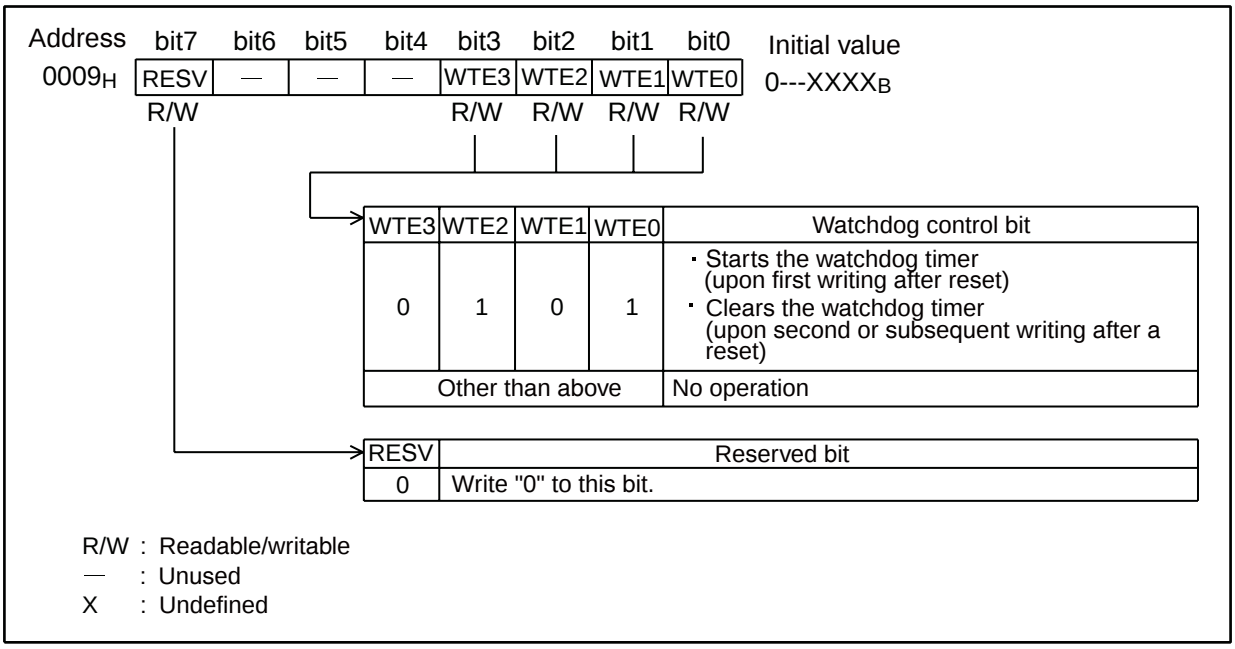


Table 6.3-1 Explanation of Functions of Each Bit in Watchdog Control Register (WDTC)

Bit name		Description
bit7	RESV: Reserved bit	• Write "0" to this bit.
bit6 to bit4	Unused bits	• Undefined when it is read. • Writing values does not affect operation.
bit3 to bit0	WTE3, WTE2, WTE1, WTE0: Watchdog control bits	• Writing "0101_B" activates (for first writing) or clears (for second or subsequent writing) the watchdog timer. • Writing other than "0101_B" does not affect operation. **Note:** These bits indicate "1111_B" when read. Bit manipulation instructions cannot be used.

6.4 Operations of Watchdog Timer Functions

The watchdog timer generates a watchdog reset when the watchdog timer counter overflows.

■ Operations of Watchdog Timer

- Activating watchdog timer

  The watchdog timer is activated when the first time "0101_B" is written to the watchdog control bits (WDTC: WTE3 to WTE0) of the watchdog control register.

  The watchdog timer cannot be stopped without accepting a reset upon activation.

- Clearing watchdog timer

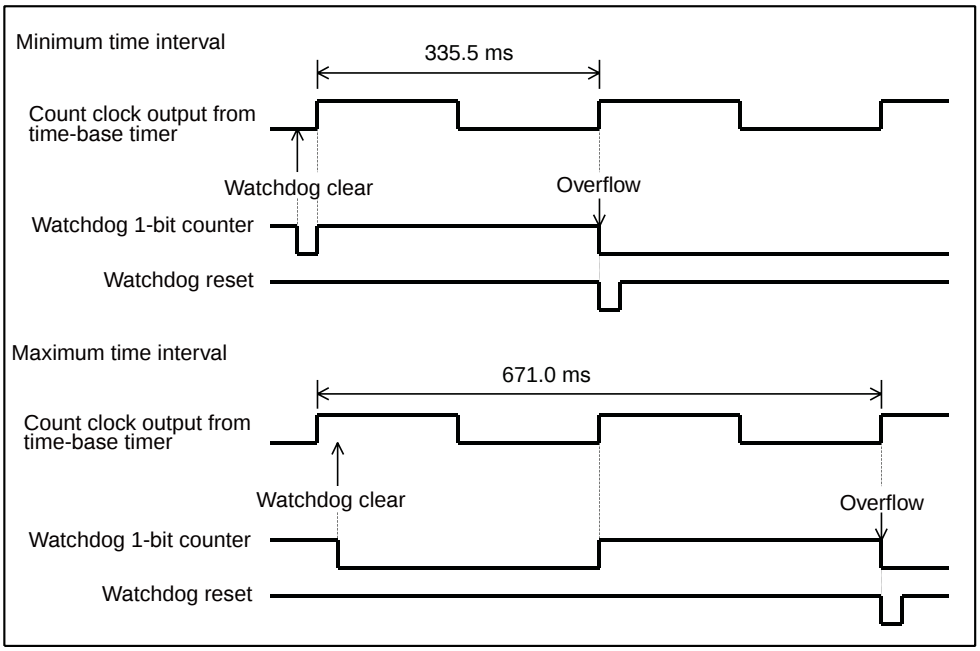
  The watchdog timer counter is cleared the second or subsequent time "0101_B" is written to the watchdog control bits (WDTC: WTE3 to WTE0) of the watchdog control register.

  When the counter is not cleared within the time interval of the watchdog timer, the counter overflows and the timer generates the internal reset signal having a period of four instruction cycles.

- Time intervals of watchdog timer

  The time interval varies depending on the timing at which the watchdog timer is cleared. Figure 6.4-1 shows the relationship between the clear timings and time intervals of the watchdog timer when output from the time-base timer is used as the count clock (oscillation frequency: 12.5 MHz).

Figure 6.4-1 Clearing Watchdog Timer and Time Interval



6.5 Notes on Using Watchdog Timer

Notes on using the watchdog timer are provided below.

■ Notes on Using Watchdog Timer

- Stopping watchdog timer

  The watchdog timer cannot be stopped without accepting a reset upon activation.

- Clearing watchdog timer

  Clearing the time-base timer counter that supplies the count clock to the watchdog timer also clears the watchdog timer counter at the same time.

  Switching to sleep or stop mode clears the watchdog timer counter.

- Notes on creating programs

  When creating a program that repeatedly clears the watchdog timer in the main loop, ensure that the time necessary for main loop processing, including interrupt handling, is shorter than the minimum time interval of the watchdog timer.

6.6 Program Example for Watchdog Timer

Programming examples for the watchdog timer are provided below.

■ Programming Examples of Watchdog Timer

● Processing specification

- Activate the watchdog timer immediately after the program starts.
- Clear the watchdog timer whenever the loop of the main program is run.
- Ensure that the time necessary for running the main loop once, including interrupt handling, is shorter than the minimum time interval (approximately 335.5 ms: operating at 12.5 MHz) of the watchdog timer.

● Coding example

```
WDTC   EQU     0009H               ;   Address of watchdog control register
WDT_CLR EQU    00000101B

VECT   DSEG    ABS                 ;   [DATA SEGMENT]
       ORG     0FFFEH
RST_V  DW      PROG                ;   Setting reset vector
VECT   ENDS
;----------------------Main program-----------------------------------------------------------------------------
       CSEG                        ;   [CODE SEGMENT]
PROG                               ;   Initialization routine upon 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

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

8-BIT PWM TIMER

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

7.1 Overview of 8-bit PWM Timer

An 8-bit PWM timer has the interval timer functions and the PWM timer functions of an 8-bit resolution. A counter is incremented using interval timer functions in synchronization with three types of internal count clocks or the output of 8/16-bit capture timer/counter. The user can select one of these functions. Therefore, the 8-bit interval timer can be set and the square wave of any frequency can be output using the set output. In addition, if a low-pass filter is connected to the PWM output, the D/A converter can be used.

■ Interval Timer Functions (Functions to Output the Square Wave)

Interrupts are generated repeatedly at any interval by the interval timer functions.

Because the output level of the pin (P50/PWM pin) can be inverted for each interrupt, the square wave of any frequency can also be output.

- An interval timer operation from the cycle of the count clock to 2^8-times cycle is possible.
- The count clock can be selected from four types.

Table 7.1-1 shows the range of intervals and square wave output.

Table 7.1-1 Range of Intervals and Square Wave Output

	Count clock cycle		Interval	Square wave output (Hz)
1	Internal count clock	$1t_{INST}$	$1t_{INST}$ to 2^8t_{INST}	$1/(2t_{INST})$ to $1/(2^9t_{INST})$
2		$16t_{INST}$	2^4t_{INST} to $2^{12}t_{INST}$	$1/(2^5t_{INST})$ to $1/(2^{13}t_{INST})$
3		$64t_{INST}$	2^6t_{INST} to $2^{14}t_{INST}$	$1/(2^7t_{INST})$ to $1/(2^{15}t_{INST})$
4	8/16-bit timer count clock	$2t_{INST}$	$2t_{INST}$ to $2^{17}t_{INST}$	$1/(2^2t_{INST})$ to $1/(2^{10}t_{INST})$
		$32t_{INST}$	2^5t_{INST} to $2^{21}t_{INST}$	$1/(2^6t_{INST})$ to $1/(2^{14}t_{INST})$
		$512t_{INST}$	2^9t_{INST} to $2^{25}t_{INST}$	$1/(2^{10}t_{INST})$ to $1/(2^{18}t_{INST})$
		$1t_{EXT}$	$1t_{EXT}$ to $2^{16}t_{EXT}$	$1/(2t_{EXT})$ to $1/(2^9t_{EXT})$

t_{INST}: Instruction cycle (Affected by the clock mode and others.)

t_{EXT} : Output cycle of an 8/16-bit capture timer

Note:

Calculation example of intervals and square wave frequency

The following expression is the interval when the count clock cycle is set to 1 t_{INST} and when an oscillation frequency (F_{CH}) of 12.5 MHz and a PWM compare register (COMR) value of DD_H (221) are set. Another expression is the frequency of the square wave output from the PWM pin that is operated continuously without changing the COMR register value.

However, the values are true when the maximum speed clock of the normal mode is selected (CS1, CS0 = 11_B, 1 instruction cycle = $4/F_{CH}$) with the system clock control register (SYCC).

Interval = (1 4/F_{CH}) (COMR register value + 1)
= (4/12.5 MHz) (221 + 1)
= 71.0 s

Output frequency = F_{CH} / (1 8 (COMR register value + 1))
= 12.5 MHz/ (8 (221 + 1))
≒ 7.04 kHz

■ PWM Timer Functions

The PWM timer functions have an 8-bit resolution and can control the "H" level width and "L" level width of one cycle.

Because the resolution is 1/256, a pulse can be output at a duty ratio of 0 to 99.6%.

The frequency of the PWM wave can be selected from four types.

The low-pass filter can be connected to the output and used as the D/A converter.

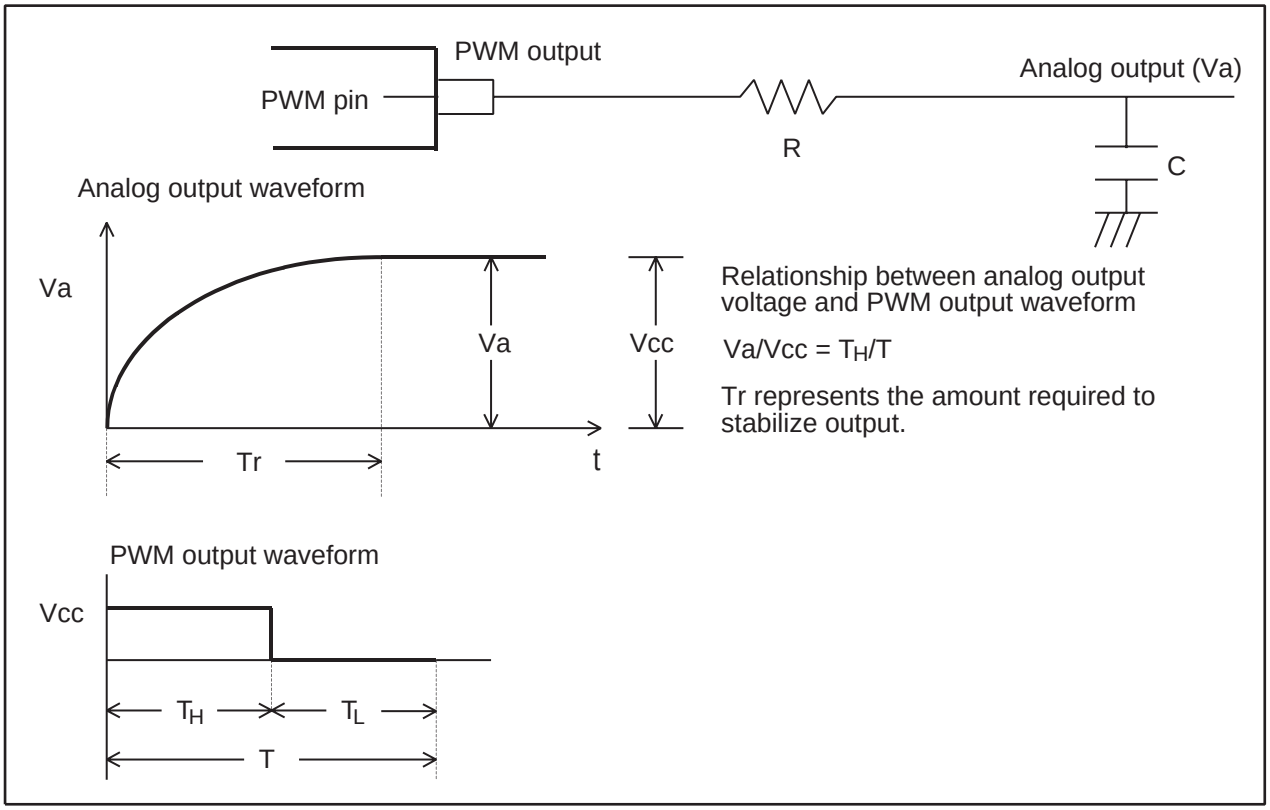
Table 7.1-2 shows the frequency of the PWM wave that can be set by PWM timer functions. Figure 7.1-1 is a configuration example of the D/A converter.

Table 7.1-2 Frequencies of the PWM Wave that can be Set by the PWM Timer Functions

	1	2	3	4			
	Internal clock			Output of an 8/16-bit capture timer/counter			
Count clock cycle	$1t_{INST}$	$16t_{INST}$	$64t_{INST}$	2^2t_{INST} to $2^{10}t_{INST}$	2^6t_{INST} to $2^{14}t_{INST}$	$2^{10}t_{INST}$ to $2^{18}t_{INST}$	$1t_{EXT}$ to 2^8t_{EXT}
PWM wave cycle	2^8t_{INST}	$2^{12}t_{INST}$	$2^{14}t_{INST}$	$2^{10}t_{INST}$ to $2^{18}t_{INST}$	$2^{14}t_{INST}$ to $2^{22}t_{INST}$	$2^{18}t_{INST}$ to $2^{26}t_{INST}$	2^8t_{EXT} to $2^{16}t_{EXT}$

t_{INST}: Instruction cycle (Affected by the clock mode and others.)
t_{EXT}: Output frequency of an 8/16-bit capture timer

Figure 7.1-1 Configuration Example of the D/A Converter with the PWM Output and a Low-Pass Filter



Note:

While PWM timer functions are enabled, no interrupt request occurs.

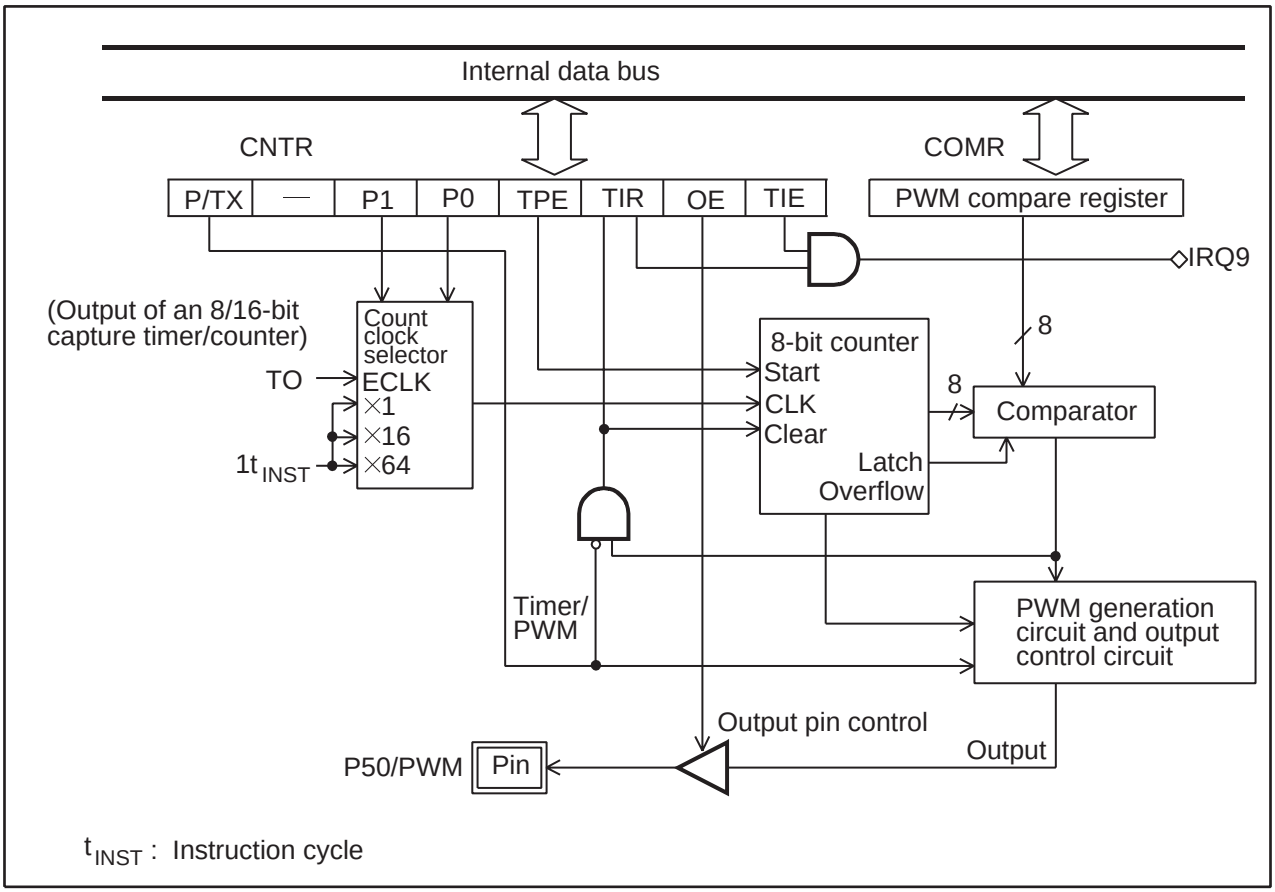
7.2 Configuration of 8-bit PWM Timer

An 8-bit PWM timer consists of the following six blocks.

- **Count clock selector**
- **8-bit counter**
- **Comparator**
- **PWM generation and output control circuit**
- **PWM compare register (COMR)**
- **PWM control register (CNTR)**

■ Block Diagram of an 8-bit PWM Timer

Figure 7.2-1 Block Diagram of an 8-bit PWM Timer



- Count clock selector

  The count clock selector selects one of three types of internal counter clock. The selector also selects an 8/16-bit capture timer or counter and uses it to increment the count of the 8-bit counter.

- 8-bit counter

  This counter is incremented by the count clock selected by the count clock selector.

- Comparator

  A latch in the comparator holds the COMR register value when the value of the 8-bit counter is 00_H and then compares the 8-bit counter with the COMR register value latched and detects a match.

- PWM generation circuit and PWM output control circuit

  During the interval timer operation, once a match is detected, an interrupt request occurs. And when the bit to control the output pin (CNTR: OE) is "1", the output level of the P50/PWM pin is inverted by the output control circuit, at which time the 8-bit counter is cleared.

  During the PWM timer operation, once a match is detected, the output level of the P50/PWM pin is changed from "H" level to "L" level by the PWM generation circuit. Thereafter, when the 8-bit counter overflows, the output level is returned to "H" level.

- COMR register

  This register is used to set a value for comparison with the counter value of the 8-bit counter.

- CNTR register

  This register is used to select the operation mode, enable and disable operations, set the count clock, control interrupts, and check status.

  When the operation mode is the PWM timer mode (P/TX = 0), the 8-bit counter cannot be cleared (by the match detection signal from the comparator) and the interrupt request (IRQ9) is disabled.

7.3 Pin of 8-bit PWM Timer

This section describes the pin and provides a block diagram of the pin related to the 8-bit PWM timer.

■ Pin Related to the 8-bit PWM Timer

The pin related to the 8-bit PWM timer is the P50/PWM pin.

- P50/PWM pin

This pin can be used as a general-purpose I/O port (P50) and for output of the interval timer or PWM timer (PWM).

PWM:

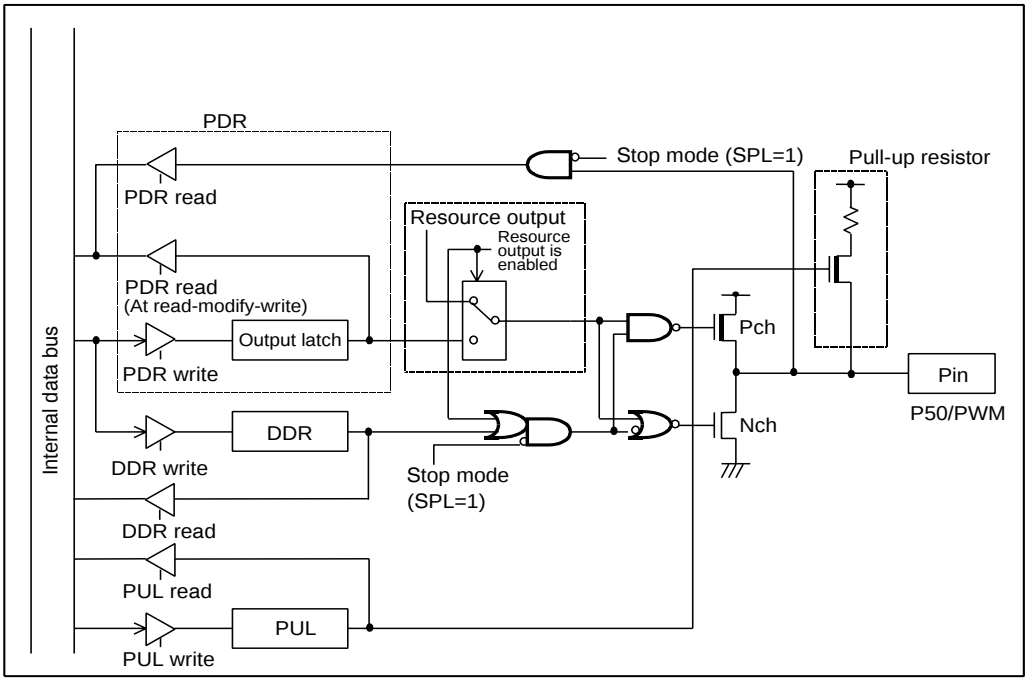
While the pin functions as the interval timer, the square wave is output to the pin.

While the pin functions as the PWM timer, the PWM wave is output to the pin.

When the bit to control the output pin is set to the dedicated pin (CNTR: OE = 1), the P50/PWM pin automatically functions as an output pin, regardless of the value of the port 5 data direction register (DDR5: bit0), and as the PWM pin.

■ Block Diagram of the Pin Related to the 8-bit PWM Timer

Figure 7.3-1 Block Diagram of the Pin Related to the 8-bit PWM Timer



7.4 Registers of 8-bit PWM Timer

This section describes the registers related to the 8-bit PWM timer.

■ Registers Related to the 8-bit PWM Timer

Figure 7.4-1 Registers Related to the 8-bit PWM Timer

CNTR (PWM control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0022_H	P/TX	—	P1	P0	TPE	TIR	OE	TIE	$0\text{-}000000_B$
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

COMR (PWM compare register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0023_H									$XXXXXXXX_B$
	W	W	W	W	W	W	W	W	

R/W : Readable/Writable
W : Write only
— : Unused
X : Undefined

Note:

Because the PWM compare register (COMR) is a write-only register, an instruction to operate bits cannot be used.

7.4.1 PWM Control Register (CNTR)

The PWM control register (CNTR) is used to select the operation mode (interval timer operation or PWM timer operation) of the 8-bit PWM timer, switch the resolution of the PWM timer functions, and select the count clock.

■ PWM Control Register (CNTR)

Figure 7.4-2 PWM Control Register (CNTR)

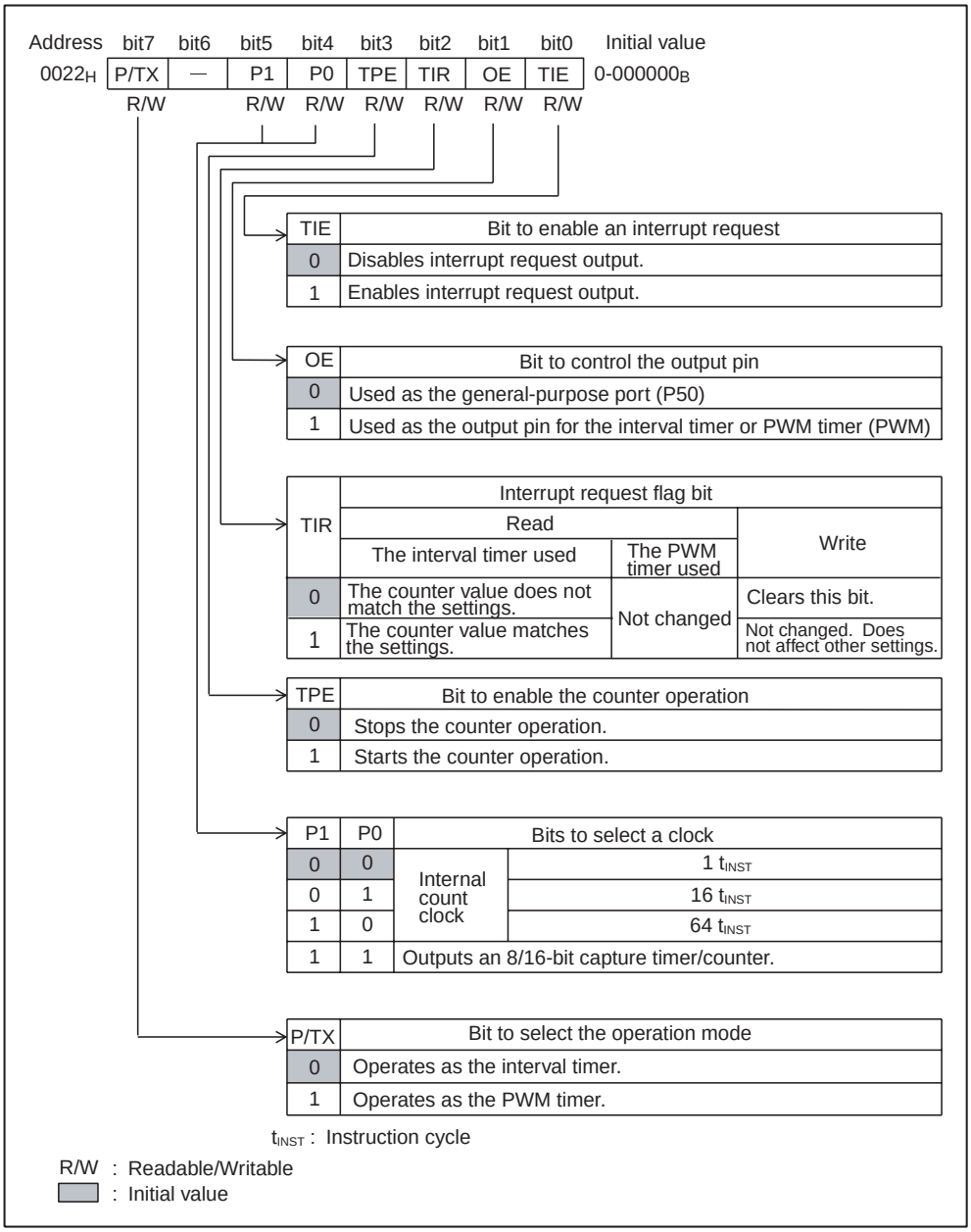


Table 7.4-1 Explanation of the Functions of Each Bit in the PWM Control Register (CNTR)

Bit name		Function
bit7	P/TX: Bit to select the operation mode	This bit is used to select the interval timer operation (P/TX = 0) or PWM timer operation (P/TX = 1). **Note:** Before writing into this bit, stop the counter operation (TPE = 0), disable an interrupt (TIE = 0), and clear the interrupt request flag bit (TIR = 0).
bit6	Unused bit	The value during a read is undetermined. A write does not affect operations.
bit5, bit4	P1, P0: Bits to select the clock	This bit is used to select the count clock of the interval timer functions or PWM timer functions. One of three types of internal count clock or the output of the 8/16-bit capture timer or counter can be selected. **Note:** When the counter is operating (TPE = 1), do not switch P1 and P0.
bit3	TPE: Bit to enable the counter operation	This bit is used to start and stop the interval timer functions or PWM timer functions. To start the count operation, write "1" to this bit. When "0" is written to this bit, the counter is cleared (setting 00_H) and then stopped.
bit2	TIR: Interrupt request flag bit	While the internal timer functions are enabled: When the counter value matches the PWM compare register (COMR) value, "1" is set to this bit. When this bit and the bit to enable an interrupt request (TIE) are "1", an interrupt request to the CPU is output. While the PWM timer functions are enabled, an interrupt request does not occur. When this bit is written, it is cleared (setting "0"). Writing "1" does not affect this bit in any way.
bit1	OE: Bit to control the output pin	When this bit is "0", the P50/PWM pin is used as a general-purpose port (P50). When the bit is "1", it is used as a dedicated pin (PWM). When the interval timer functions are enabled, the square wave is output to the PWM pin. When the PWM timer functions are enabled, the PWM wave is output to the PWM pin.
bit0	TIE: Bit to enable an interrupt request	This bit is used to enable and disable the output of an interrupt request to the CPU. When this bit and the interrupt request flag bit (TIR) are both "1", an interrupt request is output.

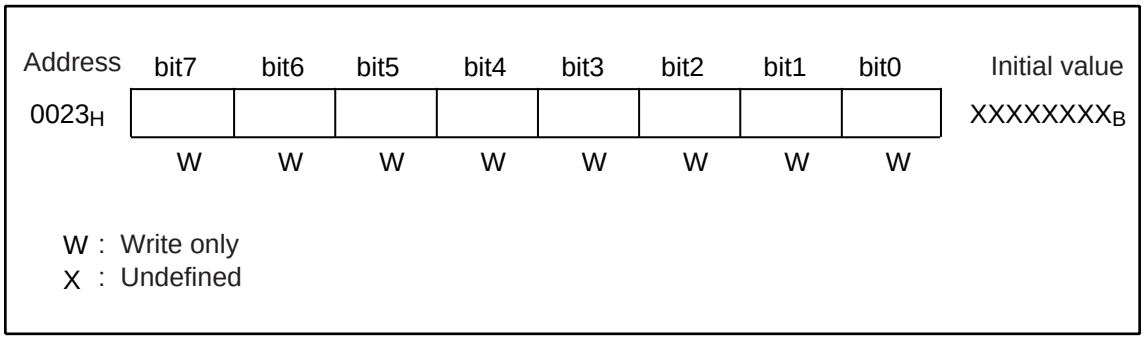
7.4.2 PWM Compare Register (COMR)

The PWM compare register (COMR) is used to set an interval while the internal timer functions are enabled. In addition, the register becomes the "H" level width of a pulse while the PWM timer functions are enabled.

■ PWM Compare Register (COMR)

Figure 7.4-3 shows the bit configuration of a PWM compare register. Because this register is a write-only register, an instruction to operate bits cannot be used.

Figure 7.4-3 PWM Compare Register (COMR)



- While the interval timer is operating:

  Specify an interval in the register to which the value compared with the counter value is to be set.

  When the settings written to this register match the counter value, the counter is cleared and "1" is set to the interrupt request flag bit (CNTR: TIR = 1).

  If a value is written to the COMR register while the counter is operating, the value takes effect at the next cycle (after detection of a match).

  Note:

  The settings of the COMR register, while the interval timer is operating, can be calculated using the following formula. The gear function, however, affects the instruction cycle.

  COMR register value = interval/(count clock cycle × instruction cycle) - 1

- While the PWM timer is operating:

  Specify the "H" level width of a pulse in the register to which the value that is compared with the counter value is to be set.

  Until the settings written to this register match the counter value, "H" is output from the PWM pin. When a match is found, "L" is output until the counter value overflows.

  If a value is written to the COMR register while the counter is operating, the value takes effect at the next cycle (after overflow).

  Note:

  The settings and cycle of the COMR register, while the PWM timer is operating, can be calculated using the following formula. The gear function, however, affects the instruction cycle.

  COMR register value = duty ratio (%) × 256

  PWM wave cycle = count clock cycle × instruction cycle × 256

7.5 Interrupt of 8-bit PWM Timer

An interrupt factor of an 8-bit PWM timer can be a match between the counter value and the PWM compare register value while interval timer functions are operating. While the PWM timer functions are enabled, an interrupt request does not occur.

■ Interrupts while Interval Timer Functions are Enabled

When the counter value is incremented from 00_H using the selected count clock and matches the PWM compare register (COMR) value, "1" is set to the corresponding interrupt request flag bit (CNTR: TIR).

At this time, if the bit to enable an interrupt request is enabled (CNTR: TIE = 1), an interrupt request (IRQ9) to the CPU occurs. Write "0" to the TIR bit using the interrupt handling routine to clear the interrupt request.

The TIR bit is set to "1" when the counter value matches the settings regardless of the value of the TIE bit.

Note:

When a match is found between the counter value and the COMR register value concurrently with the stop of the counter (CNTR: TPE = 0), the TIR bit is not set.

When the TIR bit is "1", if the TIE bit is changed from disabled to enabled (changed from "0" to "1"), an interrupt request occurs immediately.

■ Register and Vector Table Related to the Interrupts of an 8-bit PWM Timer

Table 7.5-1 Register and Vector Table Related to the Interrupts of an 8-bit PWM Timer

Interrupt name	Interrupt level setting register			Address of vector table	
	Register	Bits to be set		High-order	Low-order
IRQ9	ILR3($007D_H$)	L91(bit3)	L90(bit2)	$FFE8_H$	$FFE9_H$

See Section "3.4.2 Steps in the Interrupt Operation " for interrupt operations.

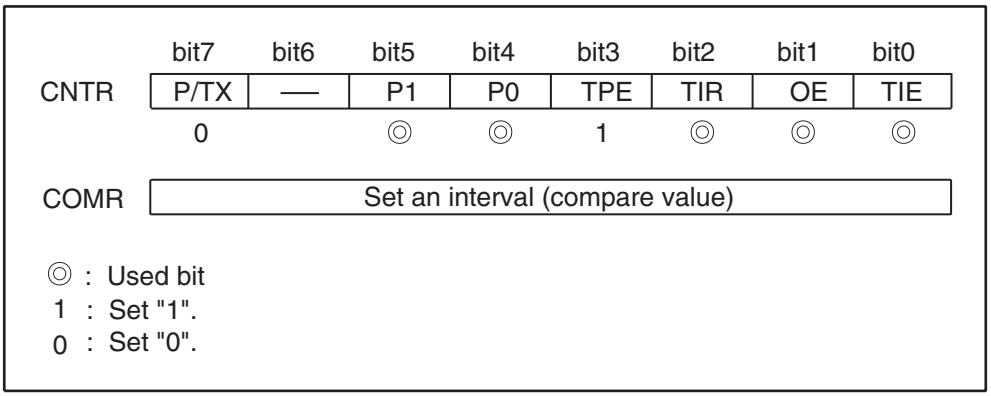
7.6 Operations of the Interval Timer Functions

This section describes the operations of the interval timer functions of an 8-bit PWM timer.

■ Operations of the Interval Timer Functions

To make an 8-bit PWM timer operate as an interval timer, set registers as shown in Figure 7.6-1 .

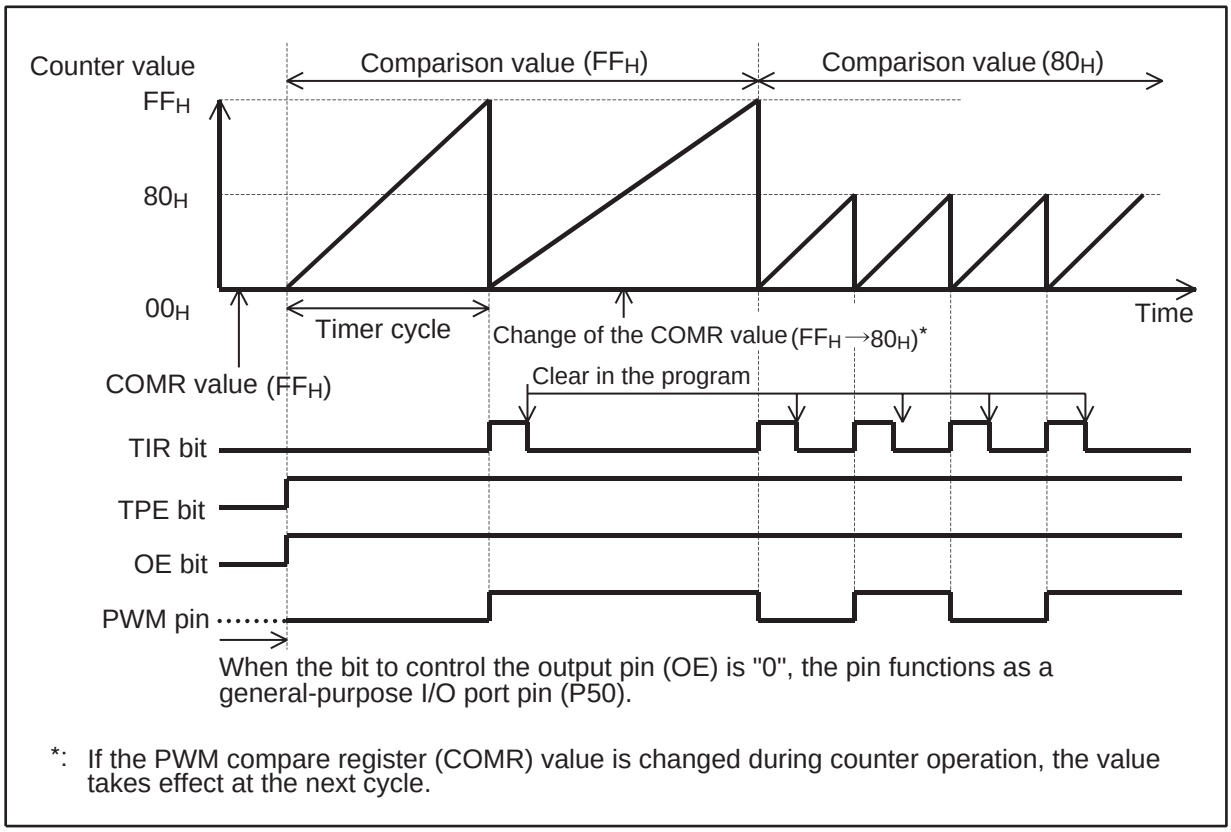
Figure 7.6-1 Setting Interval Timer Functions



When the counter is activated, the counter is incremented from 00_H at the start-up of the selected count clock. When the counter value matches the value set in the COMR register (comparison value), the timer inverts the level of the PWM pin, clears the counter, sets the interrupt request flag bit (CNTR: TIR = 1), and starts incrementing again from 00_H at the next start-up of the count clock.

Figure 7.6-2 shows the operations of an 8-bit PWM timer.

Figure 7.6-2 Operations of an 8-bit PWM Timer



Notes:

- While interval timer functions are enabled (CNTR: TPE = 1), do not change the count clock cycle (CNTR: P1, P0).
- When 00_H is set to the COMR register, the output of the PWM pin is inverted in the cycle of the count clock.
  While interval timer functions are enabled, the output level of the PWM pin in the counter stop state (CNTR: TPE = 0) is at "L" level.

7.7 Operations of the 8-bit PWM Timer Functions

This section describes the operations of the 8-bit PWM timer functions.

■ Operations of the 8-bit PWM Timer Functions

To enable 8-bit PWM timer functions, set registers as shown in Figure 7.7-1 .

Figure 7.7-1 Setting 8-bit PWM Timer Functions

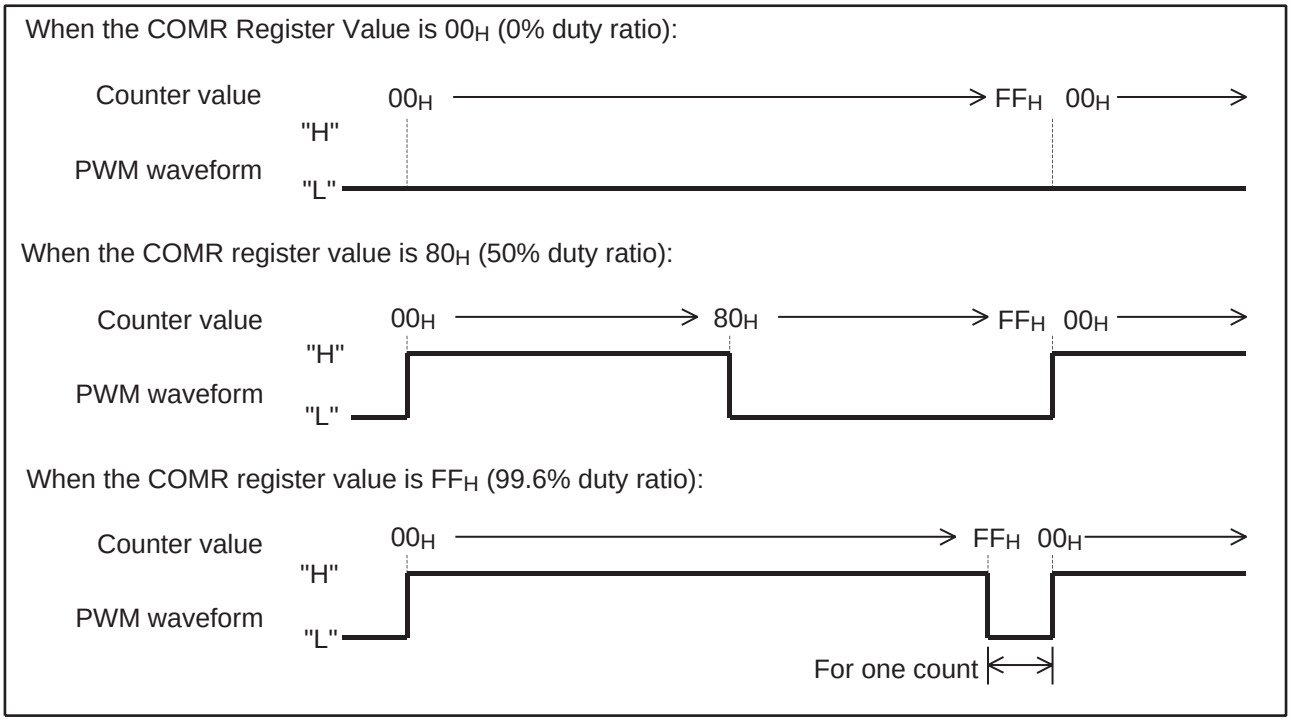
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR	P/TX	——	P1	P0	TPE	TIR	OE	TIE
	1		◎	◎	1	×	1	×
COMR	Set an H-level pulse width (compare value).							

◎ : Used bit
× : Unused bit
1 : Set "1".

When the counter is activated, the counter is incremented from 00_H at the start-up of the selected count clock. The output (PWM waveform) of the PWM pin is "H" until a match between the counter value and the value set in the COMR register is found. Once a match is found, the output is "L" until the counter value overflows ($FF_H \rightarrow 00_H$).

Figure 7.7-2 shows the PWM waveform output to the PWM pin.

Figure 7.7-2 Output Example of the PWM Waveform of 8-bit PWM Timer Functions



Notes:

- While PWM timer functions are enabled (CNTR: TPE = 1), do not change the count clock cycle (CNTR: P1, P0).
- While PWM timer functions are enabled, the level immediately before the stop is held as the output level of the PWM pin in the counter stop state (CNTR: TPE = 0).

7.8 States in Each Mode During Operation

This section describes the operations for a move to the sleep mode, a move to the stop mode, and the occurrence of a suspend request during the operation of an 8-bit PWM timer.

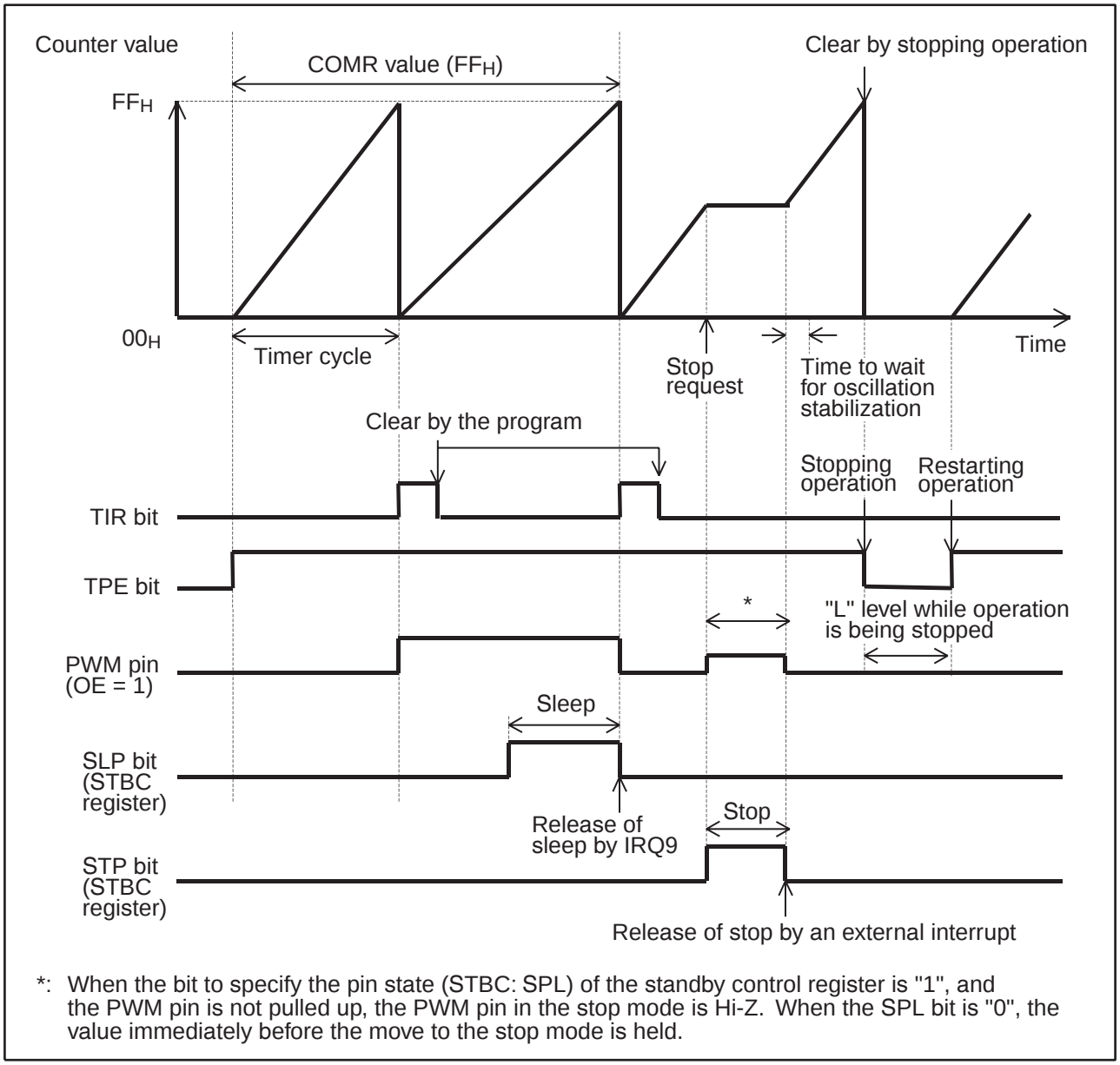
■ Operations in the Standby Mode and at a Suspension

When the mode is moved to sleep and stop modes, and when a suspend request occurs, the counter value status in which interval timer functions are enabled is shown in the Figure 7.8-1 , and the counter value status in which PWM timer functions are enabled is shown in the Figure 7.8-2 .

When switched to the stop mode, the counter holds a value and stops. When the stop mode is released by an external interrupt, the counter starts operation from the held value. Therefore, the first interval and the first cycle of the PWM waveform are not the values that are set. After the release of the stop mode, initialize the 8-bit PWM timer.

- While interval timer functions are enabled:

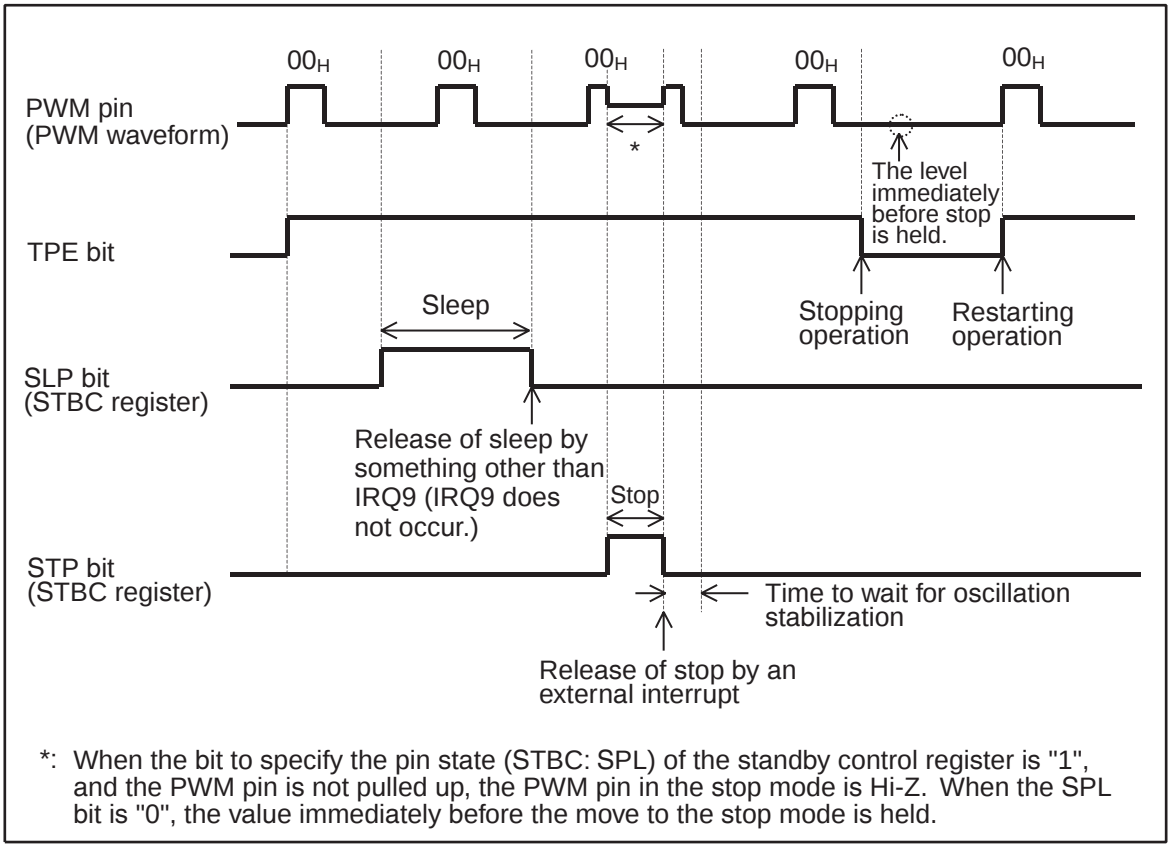
Figure 7.8-1 Operation of the Counter in the Standby Mode and during Suspension (while Interval Functions are Enabled)



*: When the bit to specify the pin state (STBC: SPL) of the standby control register is "1", and the PWM pin is not pulled up, the PWM pin in the stop mode is Hi-Z. When the SPL bit is "0", the value immediately before the move to the stop mode is held.

- While PWM timer functions are enabled:

Figure 7.8-2 Operation in the Standby Mode and during Suspension (while PWM Timer Functions are Enabled)



*: When the bit to specify the pin state (STBC: SPL) of the standby control register is "1", and the PWM pin is not pulled up, the PWM pin in the stop mode is Hi-Z. When the SPL bit is "0", the value immediately before the move to the stop mode is held.

7.9 Notes on Using 8-bit PWM Timer

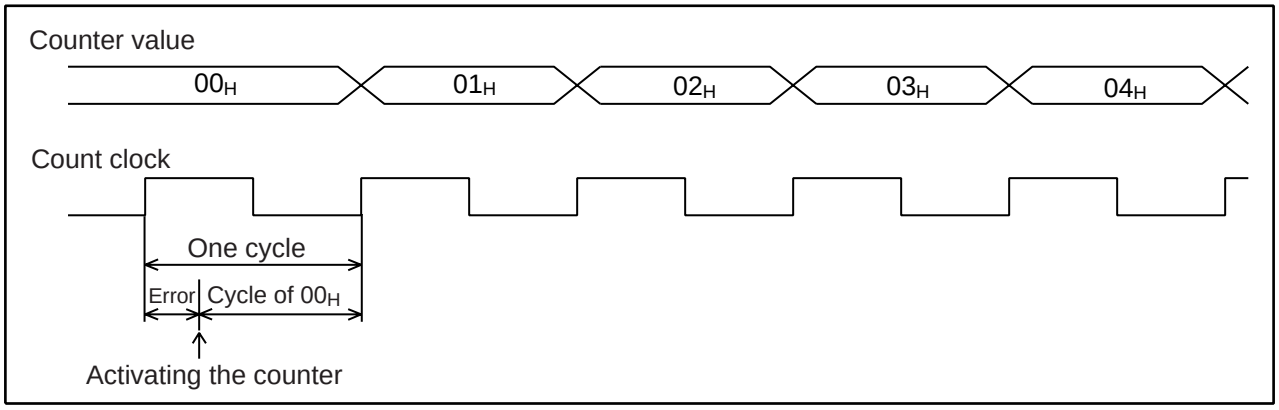
This section provides notes on using 8-bit PWM timer.

■ Notes on Using 8-bit PWM Timer

● Error

The activation of the counter by a program does not synchronize the start of an increment by the selected count clock. Therefore, as an error until a match between the counter value and the PWM compare register (COMR) value is detected, the time may be shortened by up to one cycle of the count clock cycle. Figure 7.9-1 shows an error until the count operation is started.

Figure 7.9-1 Error until the Count Operation is Started

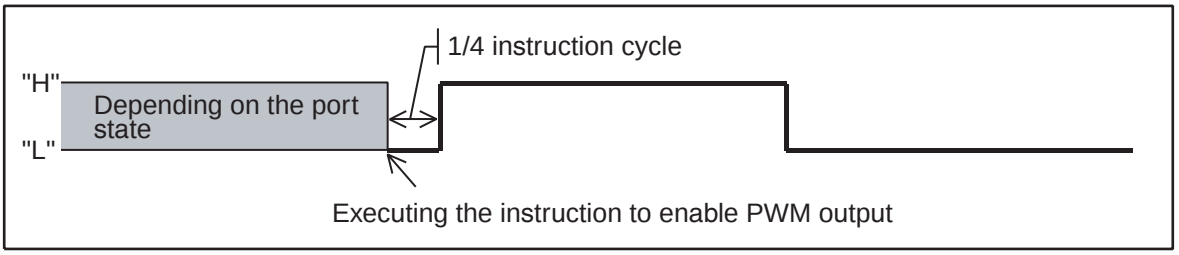


● Notes on setting by a program

- While interval timer functions or PWM timer functions are enabled (CNTR: TPE = 1), do not change the count clock cycle (CNTR: P1, P0).
- If the user wants to switch between the interval timer function and the PWM timer function (CNTR: P/TX), proceed when the counter is stopped (CNTR: TPE = 0), interrupts are disabled (CNTR: TIE = 0), and interrupt requests are cleared (CNTR: TIR = 0).
- When the interrupt request flag bit (CNTR: TIR) is "1" and the bit to enable an interrupt request is enabled (CNTR: TIE = 1), recovery from interrupt handling is no longer possible. The TIR bit must be cleared.
- When the counter value matches the COMR register value concurrently with the counter stop (CNTR: TPE = 0), the TIR bit is not set.
- Depending on how to set TPE, P/TX, and OE, the PWM output waveform varies as shown below. Be careful when using a program to set TPE, P/TX, and OE.

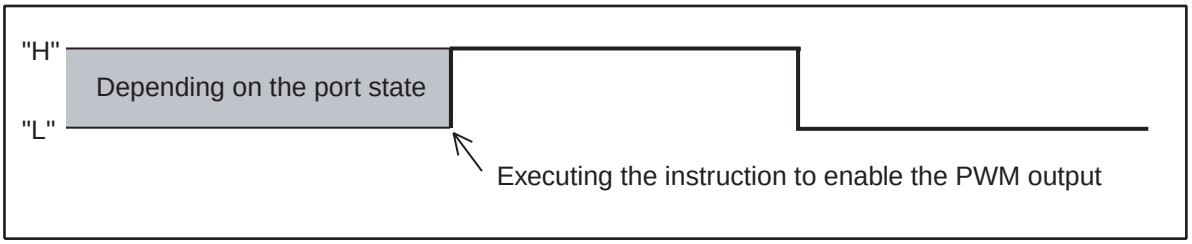
  (1) When TPE, P/TX, and OE are set at the same time:

```
MOV CNTR, #11001010B ; Starts PWM operations, internal clocks, and count operations.
                     ; Enables the PWM output.
```



(2) When OE is set after TPE and P/TX are set:

```
MOV CNTR, #11001000B ;  Starts PWM operations, internal clocks, and count operations.
                  ↑  ;  Uses the general-purpose port.
                Check
MOV CNTR, #11001010B ; Enables PWM output.
                  ↑
                Check
```



7.10 Program Example for PWM Timer

This section describes program examples of an 8-bit PWM timer.

■ Program Example of Interval Timer Functions

● Processing specifications

- 5 ms interval timer interrupts occur repeatedly.
- The square waveform that inverts at an interval is output to the P50/PWM pin.
- The following expression yields the COMR register value for which the interval is about 5 ms when the top speed of the gear (one instruction cycle = $4/F_{CH}$) is obtained at an oscillation frequency of 12.5 MHz. The count clock is 64 t_{INST} of the internal count clock.

  COMR register value = 5 ms/(64 × 4/12.5 MHz) - 1 = 244.1 ($0F4_H$)

● Coding example

```
CNTR    EQU     0022H               ;  Address of the PWM control register
COMR    EQU     0023H               ;  Address of the PWM compare register
TPE     EQU     CNTR:3              ;  Defining the bit to enable the counter operation
TIR     EQU     CNTR:2              ;  Defining the interrupt request flag bit
ILR3    EQU     007D                ;  Address of the register to set the interrupt level
INT_V   DSEG    ABS                 ;  [DATA SEGMENT]
        ORG     0FFF8H
IRQ9    DW      WARI1               ;  Setting the interrupt vector
INT_V   ENDS
;----------------------Main program-----------------------------------------------------------------------
        CSEG                        ;  [CODE SEGMENT]
                                    ;  The stack pointer (SP) and others are assumed to have
                                       been initialized.
         :
        CLRI                        ;  Disabling interrupts
        CLRB    TPE                 ;  Stopping the counter operation
        MOV     ILR3,#11110111B     ;  Setting the interrupt level (level 1)
        MOV     COMR,#0F4H          ;  Comparison value with the counter value (interval)
        MOV     CNTR,#00101011B     ;  Enabling the output of the PWM
                                    ;  Interval timer operation, selection of 64 tINST
                                    ;  Starting the counter  operation and enabling the output
                                       of interrupt requests
        SETI                        ;  Enabling interrupts
         :
;----------------------Interrupt program------------------------------------------------------------------
WARI1   CLRB    TIR                 ;  Clearing the interrupt request flag
        PUSHW   A
        XCHW    A,T                 ;  Saving A and T
        PUSHW   A
```

```
        :
        User processing
        :
        POPW    A
        XCHW    A,T             ;  Restoring A and T
        POPW    A
        RETI
        ENDS
; ---------------------------------------------------------------------------------------------------------------
```

■ Program Example of PWM Timer Functions

● Processing specifications

- A PWM wave with a duty ratio of 50% is generated. The duty ratio is then changed to 25%.
- No interrupt occurs.
- When the count clock is 16 t_{INST} of an internal count clock, the cycle of the PWM wave is 16 × 4/12.5 MHz × 256 = 1.3107 ms, which occurs when the top speed of the gear (one instruction cycle = $4/F_{CH}$) is obtained at an oscillation frequency of 12.5 MHz.
- The COMR register value with a duty ratio of 50% is shown below.
  COMR register value = 50/100 × 256 = 128 (080_H)

● Coding example

```
CNTR   EQU     0022H              ;  Address of the PWM control register
COMR   EQU     0023H              ;  Address of the PWM compare register
TPE    EQU     CNTR:3             ;  Defining the bit to enable the counter operation
;-----------------------Main program-----------------------------------------------------------------------------
       CSEG                       ;  [CODE SEGMENT]
         :
       CLRB    TPE                ;  Stopping the counter operation
       MOV     COMR,#80H          ;  Specification of the H-level width of a pulse, 50% duty
                                  ratio
       MOV     CNTR,#10011010B    ;  PWM timer operation, selection of 16 t_INST
                                  ;  Starting the counter operation, clearing the interrupt
                                  request flag
                                  ;  Enabling the output of the PWM pin, disabling the output
                                  of interrupt requests
         :
         :
       MOV     COMR,#40H          ;  Changing the duty ratio to 25% (Takes effect at the next
                                  cycle of the PWM wave.)
         :
       ENDS
;-----------------------------------------------------------------------------------------------------------------
       END
```

CHAPTER 8

8/16-BIT CAPTURE TIMER/ COUNTER

This chapter describes the functions and operation of the 8/16-bit capture timer/counter.

8.1 Overview of 8/16-bit Capture Timer/Counter

The 8/16-bit capture timer/counter consists of two 8-bit counters (timer 0 and timer 1). These counters can be used separately (8-bit mode) or in combination (16-bit mode). Timer 0 provides seven internal count clocks. This timer can select the interval timer function or counter function. The interval timer function increments the counter value in synchronization with one of the seven internal clocks. The counter function increments the counter value according to the clock to be input to an external pin. Timer 0 can output square waves of any frequency according to outputs from the interval timer and counter.
Timer 1 provides seven internal count clocks. This timer can output square waves of any frequency but can use only the interval timer function that increments the timer value in synchronization with one of the seven internal counter clocks. For the 16-bit mode, timer 0 and timer 1 are connected in series to serve as a 16-bit timer.

■ Interval Timer Function

The interval timer function generates interrupt requests repeatedly at any time interval.

This function can also invert the output level of P34/TO/INT10 pin per time interval and output square waves of any frequency.

- In the 8-bit mode, the interval timer function operates as two independent timers: timer 0 (8-bit capture timer/counter) and timer 1 (8-bit timer). Interval timer operation from each count clock cycle to a 2^8 times cycle is possible.
- The interval timer function can select and output square waves to the TO pin according to the timer 0 or 1 output.
- In the 16-bit mode, the interval timer function operates as a 16-bit capture timer/counter in which timer 0 is concatenated as the lower counter and timer 1 is concatenated as the upper counter. Interval timer operation from the count clock cycle to the 2^{16} times cycle is possible.
- The count clock can be selected from the seven internal clock cycles (if timer 0 selects an external clock, the interval timer function operates as the capture/counter function).
- The timer 0 output cycle can be used as the clock for starting A/D converters continuously or as the 8-bit PWM timer count clock.

Table 8.1-1 to Table 8.1-3 show the interval time and square wave output range in each operation mode.

Table 8.1-1 Timer 0 Interval Time and Square Wave Output Range in 8-bit Mode

Count clock cycle		Interval time	Square wave output range (Hz)
Internal count clock	$2t_{INST}$	$2t_{INST}$ to 2^9t_{INST}	$1/(2^2t_{INST})$ to $1/(2^{10}t_{INST})$
	$4t_{INST}$	2^2t_{INST} to $2^{10}t_{INST}$	$1/(2^3t_{INST})$ to $1/(2^{11}t_{INST})$
	$16t_{INST}$	2^4t_{INST} to $2^{12}t_{INST}$	$1/(2^5t_{INST})$ to $1/(2^{13}t_{INST})$
	$64t_{INST}$	2^6t_{INST} to $2^{14}t_{INST}$	$1/(2^7t_{INST})$ to $1/(2^{15}t_{INST})$
	$128t_{INST}$	2^7t_{INST} to $2^{15}t_{INST}$	$1/(2^8t_{INST})$ to $1/(2^{16}t_{INST})$
	$256t_{INST}$	2^8t_{INST} to $2^{16}t_{INST}$	$1/(2^9t_{INST})$ to $1/(2^{17}t_{INST})$
	$512t_{INST}$	2^9t_{INST} to $2^{17}t_{INST}$	$1/(2^{10}t_{INST})$ to $1/(2^{18}t_{INST})$
External clock	$1t_{ext}$	$1t_{ext}$ to 2^8t_{ext}	$1/(2t_{ext})$ to $1/(2^9t_{ext})$

Table 8.1-2 Timer 1 Interval Time and Square Wave Output Range in 8-bit Mode

Count clock cycle		Interval time	Square wave output range (Hz)
Internal count clock	$2t_{INST}$	$2t_{INST}$ to 2^9t_{INST}	$1/(2^2t_{INST})$ to $1/(2^{10}t_{INST})$
	$4t_{INST}$	2^2t_{INST} to $2^{10}t_{INST}$	$1/(2^3t_{INST})$ to $1/(2^{11}t_{INST})$
	$16t_{INST}$	2^4t_{INST} to $2^{12}t_{INST}$	$1/(2^5t_{INST})$ to $1/(2^{13}t_{INST})$
	$64t_{INST}$	2^6t_{INST} to $2^{14}t_{INST}$	$1/(2^7t_{INST})$ to $1/(2^{15}t_{INST})$
	$128t_{INST}$	2^7t_{INST} to $2^{15}t_{INST}$	$1/(2^8t_{INST})$ to $1/(2^{16}t_{INST})$
	$256t_{INST}$	2^8t_{INST} to $2^{16}t_{INST}$	$1/(2^9t_{INST})$ to $1/(2^{17}t_{INST})$
	$512t_{INST}$	2^9t_{INST} to $2^{17}t_{INST}$	$1/(2^{10}t_{INST})$ to $1/(2^{18}t_{INST})$

Table 8.1-3 Interval Time and Square Wave Output Range in 16-bit Mode

Count clock cycle		Interval time	Square wave output range (Hz)
Internal count clock	$2t_{INST}$	$2t_{INST}$ to $2^{17}t_{INST}$	$1/(2^2t_{INST})$ to $1/(2^{18}t_{INST})$
	$4t_{INST}$	2^2t_{INST} to $2^{18}t_{INST}$	$1/(2^3t_{INST})$ to $1/(2^{19}t_{INST})$
	$16t_{INST}$	2^4t_{INST} to $2^{20}t_{INST}$	$1/(2^5t_{INST})$ to $1/(2^{21}t_{INST})$
	$64t_{INST}$	2^6t_{INST} to $2^{22}t_{INST}$	$1/(2^7t_{INST})$ to $1/(2^{23}t_{INST})$
	$128t_{INST}$	2^7t_{INST} to $2^{23}t_{INST}$	$1/(2^8t_{INST})$ to $1/(2^{24}t_{INST})$
	$256t_{INST}$	2^8t_{INST} to $2^{24}t_{INST}$	$1/(2^9t_{INST})$ to $1/(2^{25}t_{INST})$
	$512t_{INST}$	2^9t_{INST} to $2^{25}t_{INST}$	$1/(2^{10}t_{INST})$ to $1/(2^{26}t_{INST})$
External clock	$1t_{EXT}$	$1t_{EXT}$ to $2^{16}t_{EXT}$	$1/(2t_{EXT})$ to $1/(2^{17}t_{EXT})$

t_{INST} : Instruction cycle (this cycle is affected by the clock mode, etc.)
t_{EXT} : External clock cycle ($1t_{EXT}$ greater than or equal to $4t_{INST}$)

Note:

Example of calculating interval time and square wave frequency

If the oscillation (F_{CH}) is set to 12.5 MHz, the timer 0 data register (TDR0) value is set to DD_H(221), and the count clock cycle is set to $2t_{INST}$ 8-bit mode operation, the interval time of timer 0 and the square wave frequency output from the TO pin when the interval timer function is operated continuously without modifying the TDR0 register value are calculated from the following expressions:

However, the values calculated from these expressions are valid when the highest speed clock of the normal mode (CS1, CS0 = 11_B, 1 instruction cycle = $4/F_{CH}$) is selected according to the system clock control register (SYCC).

Interval time = (2 4/F_{CH}) (TDR0#1 register value + 1)
= (8/12.5 MHz) (221 + 1)
≒ 142.1 s

Output frequency = F_{CH}/(2 8 (TDR0#1 register value + 1))
= 12.5 MHz/(16 (221 + 1))
≒ 3.53 kHz

■ Counter Function

The counter function counts the falling edges of the external clocks input to the P33/EC external pin. The 8/16-bit capture timer/counter can operate independently because the EC pin acts as an external clock input pin. Only timer 0 can select the external clock. The counter function operates using timer 0 with the 8-bit mode or with the 16-bit mode.

- The counter function counts the number of edges of the external clocks selected by the count clock selection bit (CINV) of the timer 0 control register (TCR0). When the number of edges equals the setting value, the counter function generates an interrupt request and inverts the output level of the square wave output pin.
- In timer 0 for the 8-bit mode, a count operation up to 2^8 is possible.
- In the 16-bit mode, a count operation up to 2^{16} is possible.
- Inputting an external clock whose cycle is constant enables the counter to be used as a device whose function is similar to an interval timer.

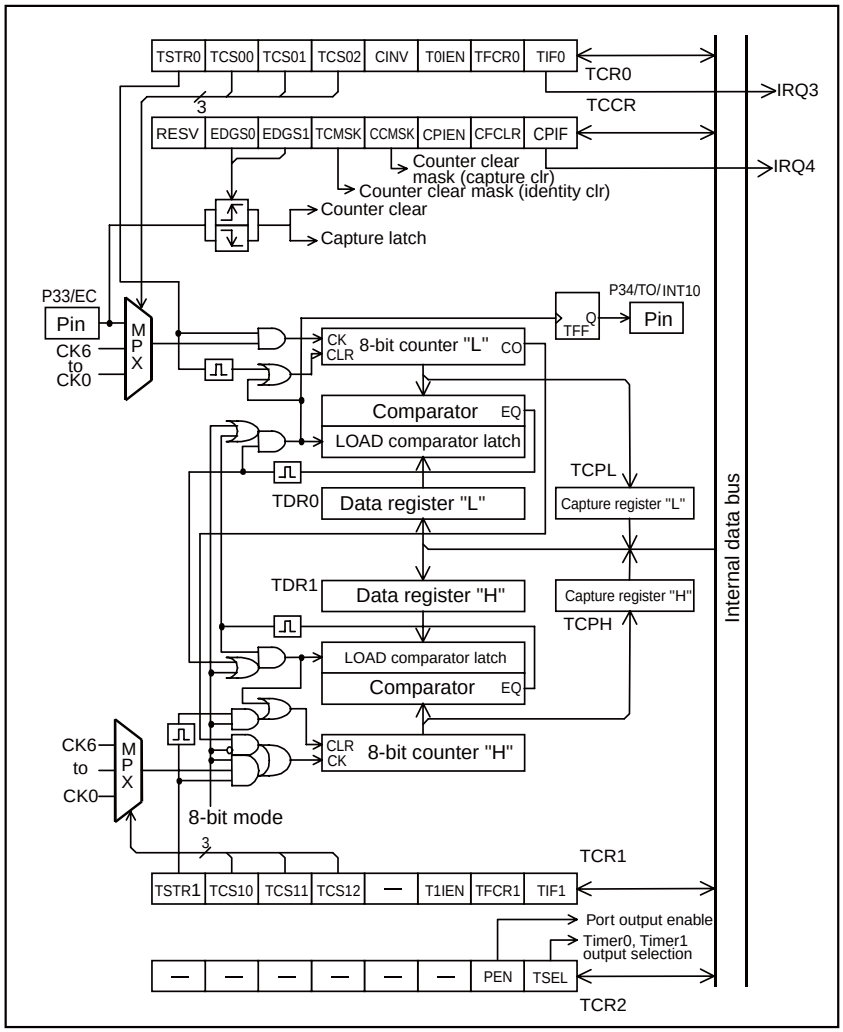
8.2 Configuration of 8/16-bit Capture Timer/Counter

The 8/16-bit capture timer/counter consists of the following seven blocks:

- **Count clock selectors 0/1**
- **Counter circuits 0/1**
- **Square wave output control circuit**
- **Timer 0/1 data registers (TDR0, TDR1)**
- **Timer 0/1 control registers (TCR0, TCR1)**
- **Capture data registers (TCPL, TCPH)**
- **Timer output control register (TCR2)**

■ Block Diagram of 8/16-bit Capture Timer/Counter

Figure 8.2-1 Block Diagram of 8/16-bit Capture Timer/Counter



- Count clock selectors 0/1

  Circuits that select input clocks. In timer 0 for the 8-bit mode or in the 16-bit mode, count clock selector 0/1 can select seven internal clocks and one external clock. In timer 1 for the 8-bit mode, the selector can select only seven internal clocks.

- Counter circuits 0/1

  Counter circuit 0 and counter circuit 1 each consist of an 8-bit counter, a comparator, a comparator data latch, and data registers (TDR0, TDR1).

  The 8-bit counter is incremented according to the selected count clock and clock edge (rising/falling). The comparator compares the counter value with the comparator data latch value. When these values match, the counter is cleared and the data register value is set in (loaded to) the comparator data latch.

  In the 8-bit mode, counter circuits 0 and 1 operate independently as timer 0 and timer 1, respectively. In the 16-bit mode, counter circuits 0 and 1 operate as the 16-bit counter in which counter circuit 0 is concatenated as lower 8 bits and counter circuit 1 is concatenated as higher 8 bits.

- Square wave output control circuit

  When the comparator detects that the counter value matches the comparator data latch value in the 8- or 16-bit mode, an interrupt request is generated. In this case, if square wave output is allowed, the corresponding output control circuit inverts the output of the square wave output pin.

- Timer 0/1 data registers (TDR0, TDR1)

  TDR0 and TDR1 are used to set the data to be compared with each 8-bit counter value at write.

- Timer 0/1 control registers (TCR0, TCR1)

  TCR0 and TCR1 are used to select functions, allow and prohibit operations, control interrupts, and check interrupt states.

- 8/16-bit capture timer/counter interrupt

  IRQ3: If the interrupt request output is allowed when the counter value equals the value set in the data register in the interval timer or counter function, an IRQ3 interrupt request is generated. (In timer 0 for the 8-bit mode or in the 16-bit mode, the interrupt request output is allowed when TCR0: T0IEN=1. In timer 1 for the 8-bit mode, the interrupt request output is allowed when TCR1: T1IEN =1.)

- 8/16-bit capture counter interrupt

  IRQ4: If the interrupt request output is allowed each time a capture input edge is detected, an IRQ4 interrupt request is generated. (In timer 0 for the 8-bit mode or in the 16-bit mode, the interrupt request output is allowed when TCCR: TCEN=1.)

- Capture data registers (TCPL, TCPH)

  TCPL and TCPH store the number of events detected in the capture mode.

  When capture data is read in the timer mode, the counter value is also read.

- Timer output control register (TCR2)

  TCR2 is used to allow and prohibit square wave output and select timer 0 output/timer 1 output.

8.3 Pins of 8/16-bit Capture Timer/Counter

This section provides pins of 8/16-bit capture timer/counter and a block diagram for these pins.

■ Pins of 8/16-bit Capture Timer/Counter

8/16-bit capture timer/counter pins include P33/EC and P34/TO/INT10.

● P33/EC pin

The P33/EC pin shares functions of the general-purpose I/O port (P33) and the external clock for the timer or capture input pin (EC).

EC:

When external clock input is selected (TCR0: TCS02, TCS01, TCS00=111_B) in timer 0 for the 8-bit mode or in the 16-bit mode, the clocks input to this pin are counted. In the capture function, this pin is also used as an input pin. When using this pin as the EC pin, set 0 in the port data 3-direction register (DDR3: bit3) and set the output transistor to OFF to enable the EC pin to be used as an input port.

● P34/TO/INT10 pin

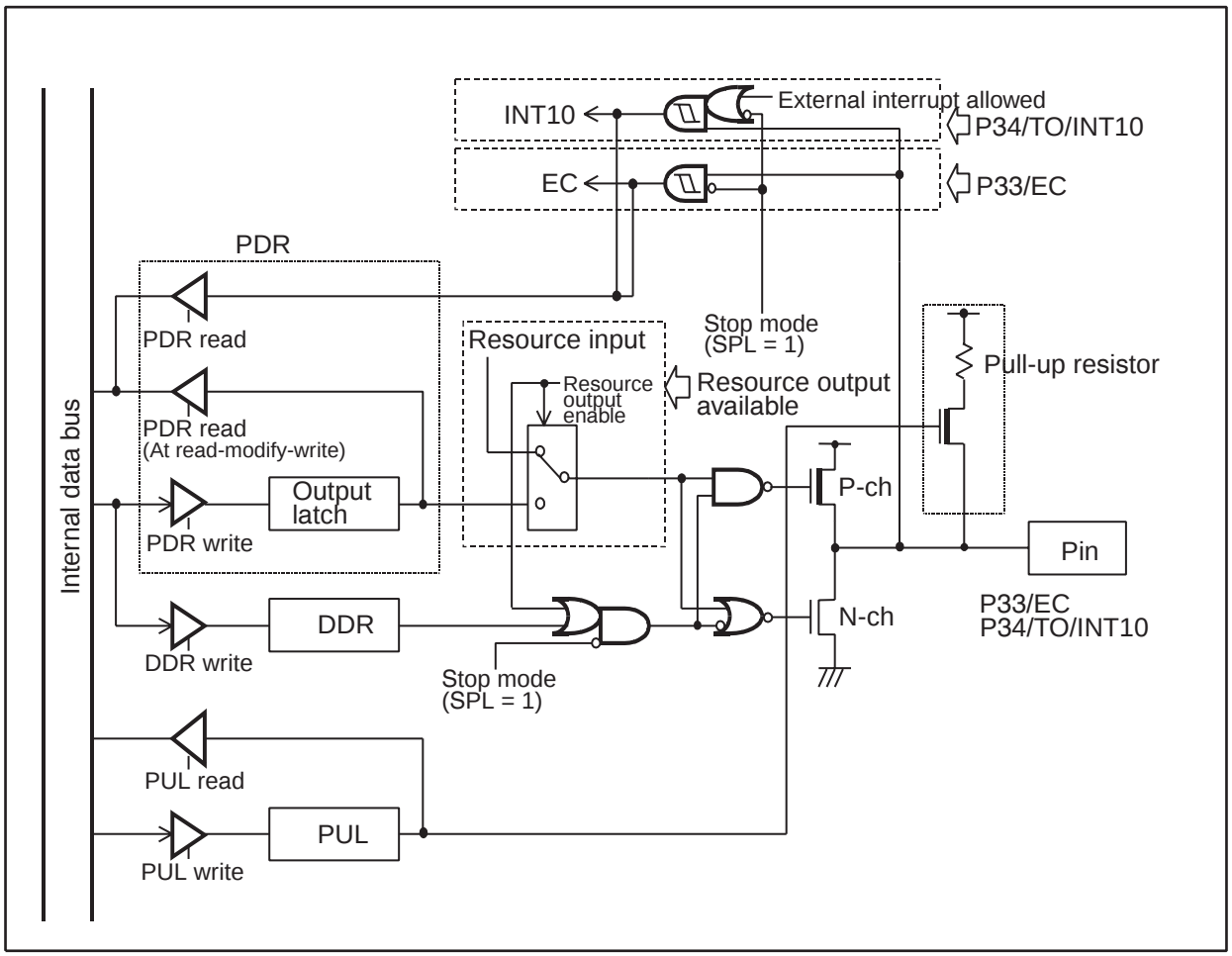
The P34/TO/INT10 pin shares functions of the general-purpose I/O port (P34) and the square wave output pin for the timer (TO). It also shares a function of the input pin for external interrupt 1 (INT10).

TO:

In timer 0 or 1 (switching allowed) for the 8-bit mode or in the 16-bit mode, a square wave is output from this pin. If square wave output is enabled (TCR2: PEN = 1), the P34/TO/INT10 pin automatically functions as an output pin without reference to the port 3-direction register (DDR3: bit4); it functions as the TO pin. The TCR2: TSEL is used to select whether timer 0 output or timer 1 output is to be used.

■ Block Diagram for 8/16-bit Capture Timer/Counter Pins

Figure 8.3-1 Block Diagram for 8/16-bit Capture Timer/Counter Pins



Note:

When "pull-up resistor available" is selected in the pull-up setting register, the pin state in the stop mode (SPL = 1) becomes high (pull-up state), not Hi-Z. During the reset, however, pull-up becomes ineffective and the pin state becomes Hi-Z.

8.4 Registers of 8/16-bit Capture Timer/Counter

This section shows registers of 8/16-bit capture timer/counter.

■ Registers of 8/16-bit Capture Timer/Counter

Figure 8.4-1 Registers of 8/16-bit Capture Timer/Counter

TCCR (capture control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0019_H	CPIF	CFCLR	CPIEN	CCMSK	TCMSK	EDGS1	EDGS0	RESV	00000000_B
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

TCR1 (timer 1 control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$001A_H$	TIF1	TFCR1	T1IEN	—	TCS12	TCS11	TCS10	TSTR1	$000\text{-}0000_B$
	R	R/W	R/W		R/W	R/W	R/W	R/W	

TCR0 (timer 0 control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$001B_H$	TIF0	TFCR0	T0IEN	CINV	TCS02	TCS01	TCS00	TSTR0	00000000_B
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

TDR1 (timer 1 data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$001C_H$									$XXXXXXXX_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

TDR0 (timer 0 data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$001D_H$									$XXXXXXXX_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

TCPH (capture data register H)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$001E_H$									$XXXXXXXX_B$
	R	R	R	R	R	R	R	R	

TCPL (capture data register L)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$001F_H$									$XXXXXXXX_B$
	R	R	R	R	R	R	R	R	

TCR2 (timer output control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0020_H	—	—	—	—	—	—	PEN	TSEL	$------00_B$
							R/W	R/W	

R/W : Readable/Writable
R : Read only
X : Undefined
— : Unused

8.4.1 Capture Control Register (TCCR)

The capture control register (TCCR) is used to select functions and detection edges, control interrupts, and check interrupt states in timer 0 for the 8-bit mode of the 8/16 bit capture timer/counter or in capture mode (16-bit mode).

■ Capture Control Register (TCCR)

Figure 8.4-2 Capture Control Register (TCCR)

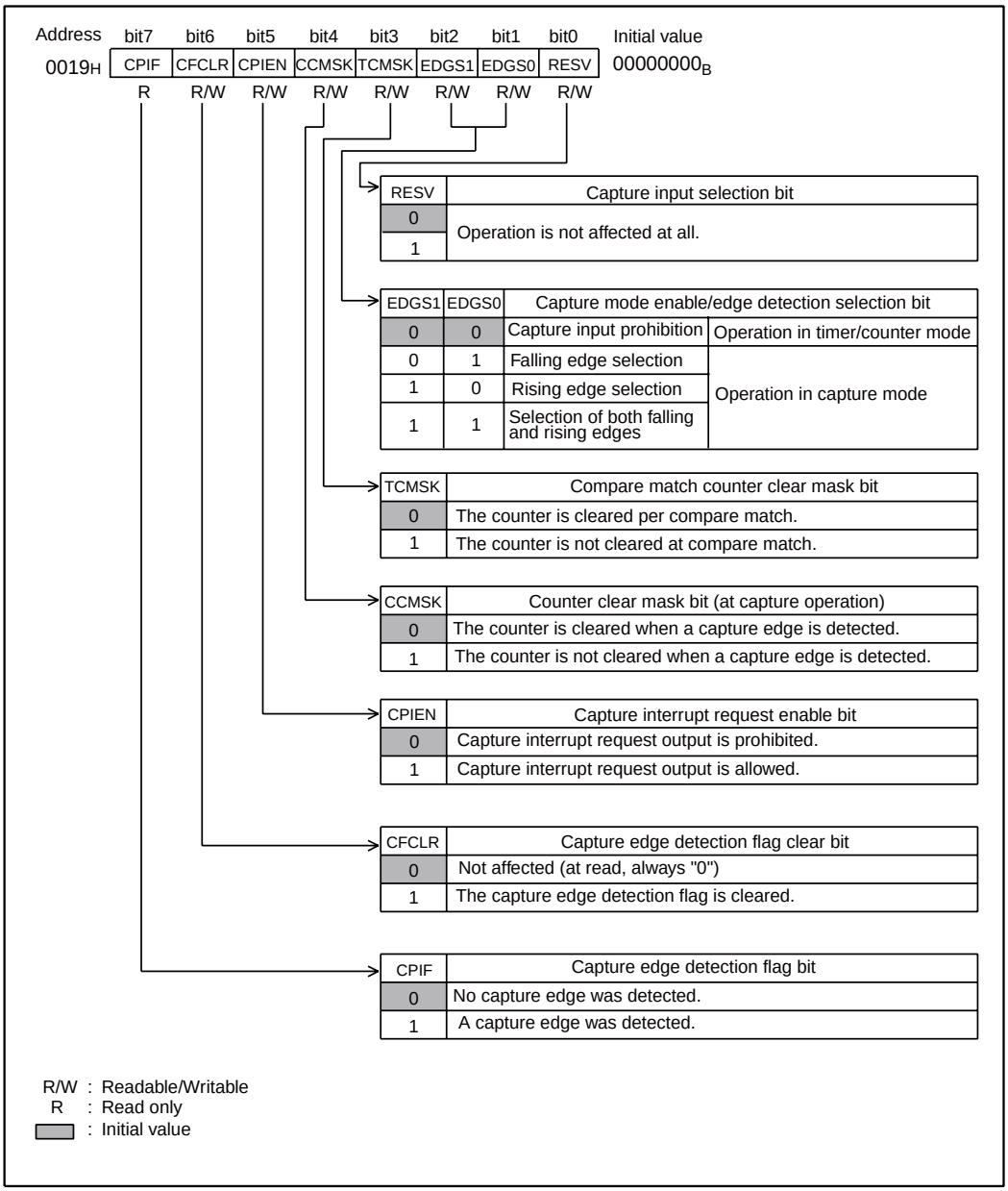


Table 8.4-1 Explanation of Functions of Each Bit in Capture Control Register (TCCR)

Bit name		Function
bit7	CPIF: Capture edge detection flag bit	• This bit is set to "1" when the edge specified by EDGS1 and EDGS0 is detected. An interrupt request is output when this bit and the capture interrupt request enable bit (CPIEN) are "1".
bit6	CFCLR: Capture edge detection flag clear bit	• This bit is used to clear the capture edge detection flag. • When this bit is "1" at write, the capture edge detection flag is cleared. When "0", the capture edge detection flag is not affected (remains unchanged).
bit5	CPIEN: Capture interrupt request enable bit	• This bit is used to allow and prohibit interrupt request output to the CPU. • An interrupt request is output when this bit and the capture edge detection flag bit (CPIF) are "1".
bit4	CCMSK: Counter clear mask bit (at capture operation)	• The counter state when a capture match is detected is set. • When this bit is "0", the counter is cleared. When this bit is "1", the counter is not cleared.
bit3	TCMSK: Compare match counter clear mask bit	• The counter state when a compare edge is detected is set. • When this bit is "0", the counter is cleared. When this bit is "1", the counter is not cleared.
bit2, bit1	EDGS1 and EDGS0: Capture mode enable/ edge detection selection bits	• These bits are used to allow and prohibit the capture function and select capture edges. • When using the 8/16-bit capture timer/counter in the capture mode, set these bits to a value other than "00_B". • When the edge set by these bits is input, the capture edge detection flag bit (CPIF) is set to "1".
bit0	RESV: Reserved bit	• Even if this bit is set to "0" or "1", the operation is not affected. The value previously written becomes the read value.

8.4.2 Timer 0 Control Register (TCR0)

The timer 0 control register (TCR0) is used to select functions, allow and prohibit operation, control interrupts, and check interrupt states in timer 0 for the 8-bit mode of the 8/16-bit capture timer/counter or in the 16-bit mode. Even if only timer 0 is used in the 8-bit mode, the timer 1 control register (TCR1) must be initialized.

■ Timer 0 Control Register (TCR0)

Figure 8.4-3 Timer 0 Control Register (TCR0)

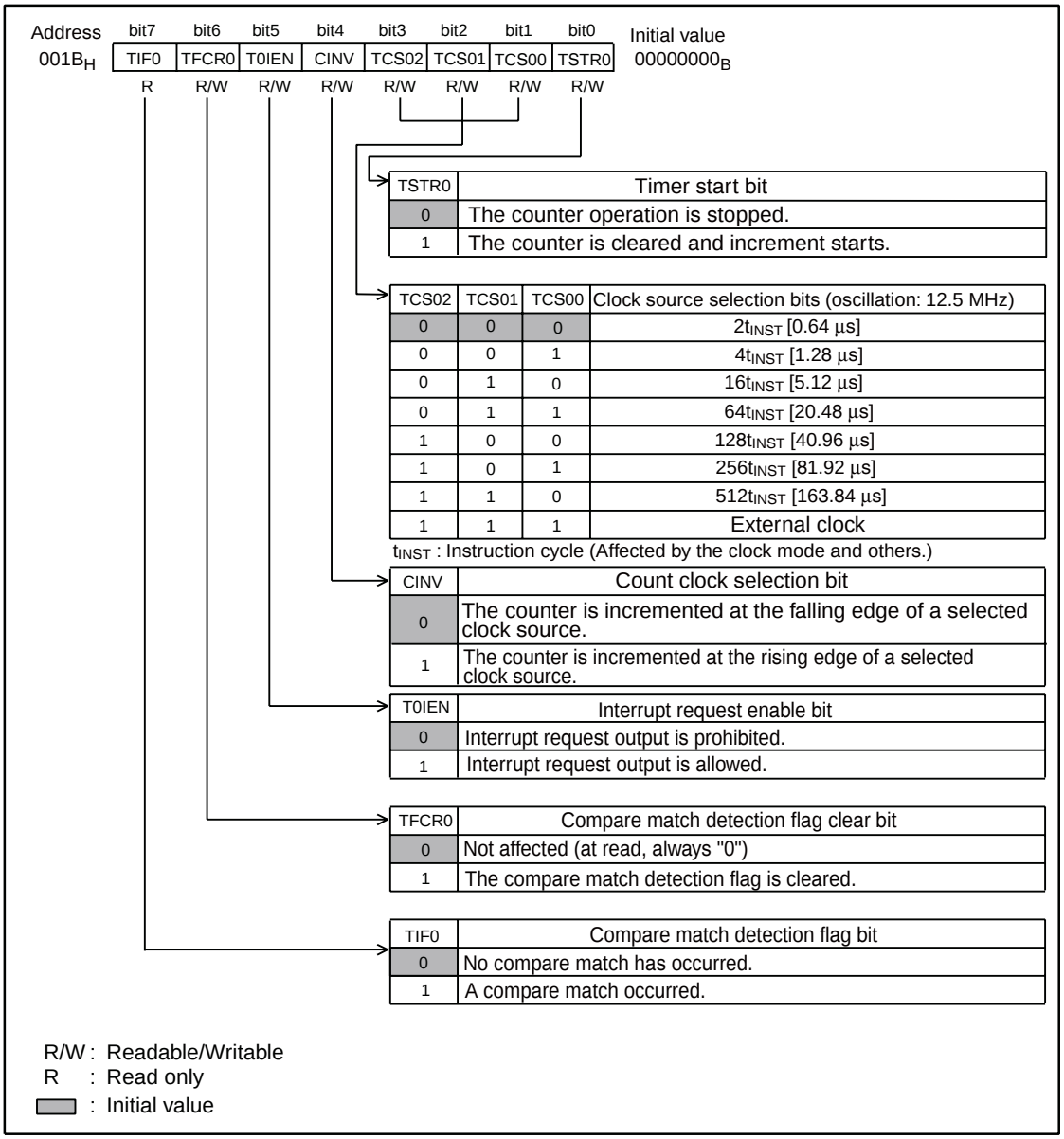


Table 8.4-2 Explanation of Functions of Each Bit in Timer 0 Control Register (TCR0)

Bit name		Function
bit7	TIF0: Compare match detection flag bit	• 8-bit mode When the counter value of timer 0 matches the value (comparator data latch) set in the timer 0 data register (TDR0), this bit is set to "1". • 16-bit mode When the counter value of timer 0 matches the value set in TDR0 and the counter value of timer 1 matches the value set in TDR1, this bit is set to "1". • An interrupt request is output when this bit and the interrupt request enable bit (T0IEN) are "1".
bit6	TFCR0: Compare match detection flag clear bit	• This bit is used to clear the compare match detection flag bit (TIF0). When this bit is set to "1", the compare match detection flag is cleared. The flag is not affected even if this bit is set to "0".
bit5	T0IEN: Interrupt request enable bit	• This bit is used to allow and prohibit interrupt request output to the CPU. • An interrupt request is output when this bit and the interrupt request enable bit (T0IEN) are "1".
bit4	CINV: Count clock selection bit	• This bit is used to select whether to increment the counter at the rising or falling edge of a clock. When this bit is "0", the counter is incremented at the falling edge of the clock. When "1", the counter is incremented at the rising edge.
bit3 to bit1	TCS02, TCS01, TCS00: Clock source selection bits	• These bits are used to select the count clocks to be supplied to the counter. • Select one clock from the seven internal clocks and one external clock. • When these bits are 111_B, the external clock is input. In this case, timer 0 can operate as the counter function. **Note:** When external clock input is selected (TCS02, TCS01, TCS00 = 111_B), the P33/EC pin must be set in the input port.
bit0	TSTR0: Timer start bit	• This bit is used to start and stop the counter. • When this bit is set to "1", the counter is cleared and incremented according to the selected count clock. When this bit is set to "0", the counter stops its operation. • When the timer is started (TSTR0 = 0 →1) in the 16-bit mode, the counters of both timer 0 and timer 1 are cleared.

Note:

When using only timer 0 of the 8/16-bit capture timer/counter in the 8-bit mode, set a value other than 111_B in the count clock selection bits (TCS12, TCS11, TCS10) of the timer 1 control register (TCR1). Using timer 0 with setting value TCS12, TCS11, TCS10 = 111_B results in a malfunction.

8.4.3 Timer 1 Control Register (TCR1)

The timer 1 control register (TCR1) is used to select functions, allow and prohibit operation, control interrupts, and check interrupt states in timer 1 for the 8-bit mode of the 8/16-bit capture timer/counter. When used in the 16-bit mode, TCR1 is controlled by the timer 0 control register (TCR0), but TCR1 setting is required.

■ Timer 1 Control Register (TCR1)

Figure 8.4-4 Timer 1 Control Register (TCR1)

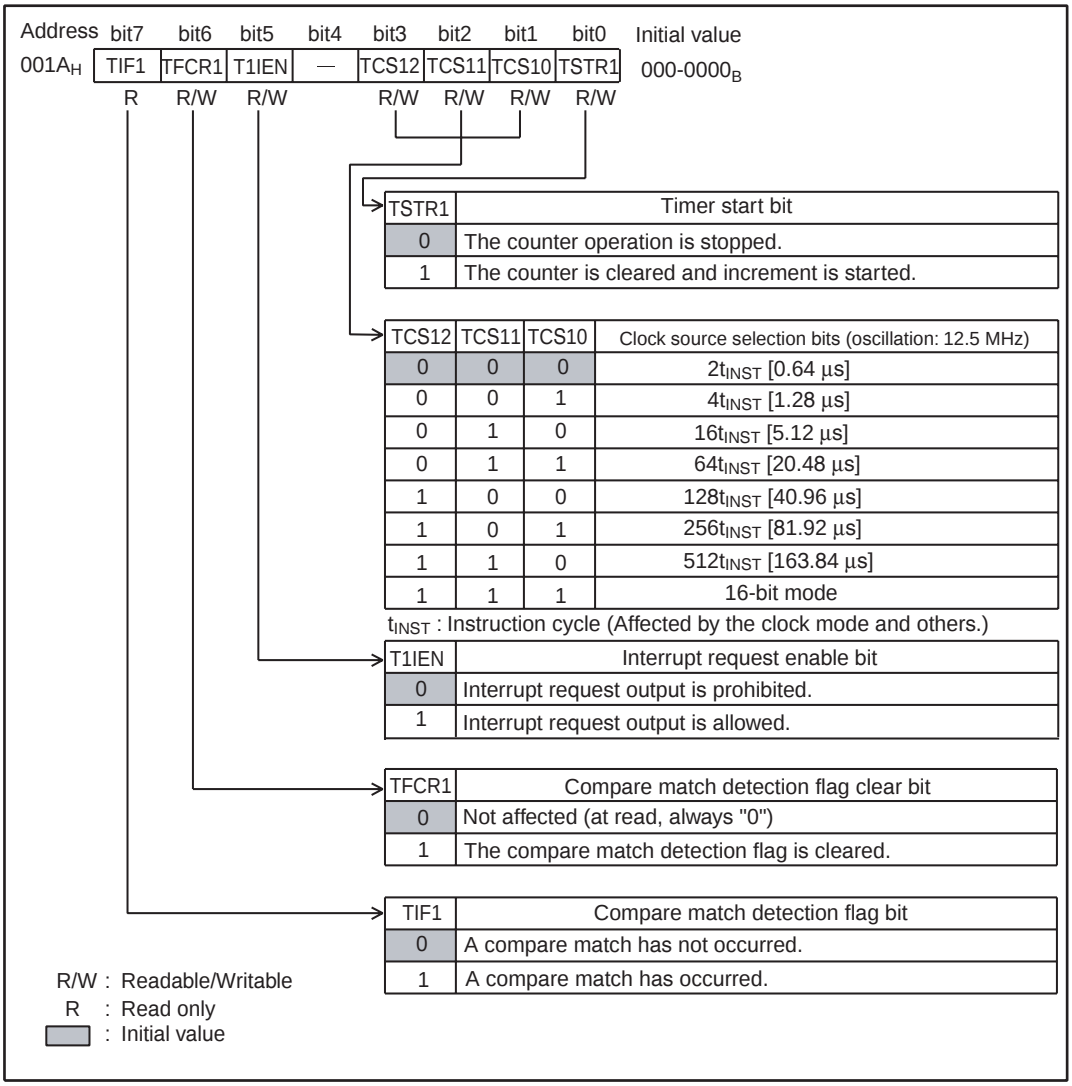


Table 8.4-3 Explanation of Functions of Each Bit in Timer 1 Control Register (TCR1)

Bit name		Function
bit7	TIF1: Compare match detection flag bit	• This bit is set to "1" when the counter value of timer 1 matches the value (comparator data latch) set in the timer 1 data register (TDR1). • An interrupt request is output when this bit and the interrupt request enable bit (T1IEN) are "1". **Note:** In the 16-bit mode, the TIF0 bit of TCR0 is valid. The TIF1 bit is unrelated to operation.
bit6	TFCR1: Compare match detection flag clear bit	• This bit is used to clear the compare match detection flag bit (TIF1). When this bit is set to "1", the compare match detection flag is cleared. The flag is not affected even if this bit is set to "0".
bit5	T1IEN: Interrupt request enable bit	• This bit is used to allow and prohibit interrupt request output to the CPU. • An interrupt request is output when this bit and the interrupt request enable bit (T0IEN) are "1".
bit4	Not used	• This bit is undefined at read. • At write, this bit does not affect operation.
bit3 to bit1	TCS12, TCS11, TCS10: Clock source selection bits	• These bits are used to select the count clocks to be supplied to the counter. • Of seven internal clocks, select one. • When 111_B is written to these bits, timer 1 operates as the 16-bit mode. **Note:** In the 16-bit mode, the TCS02, TCS01, and TCS00 bits are valid. The TCS12, TCS11, and TCS10 bits are used to select the 16-bit mode only.
bit0	TSTR1: Timer start bit	• This bit is used to start and stop the counter. • When this bit is set to "1", the counter is cleared and incremented according to the selected count clock. When this bit is set to "0", the counter stops its operation. • In the 16-bit mode, only the TSTR0 bit can be used to start the timer. The TSTR1 bit is unrelated to operation.

Note:

When using timer 1 in the 16-bit mode, write 111_B to the TCS12, TCS11, and TCS10 bits and then control timer 1 with TCR0.

8.4.4 Timer Output Control Register (TCR2)

The timer output control register (TCR2) is used to allow and prohibit the square wave output of the 8/16-bit capture timer/counter and select timer 0 output and timer 1 output.

■ Timer Output Control Register (TCR2)

Figure 8.4-5 Timer Output Control Register (TCR2)

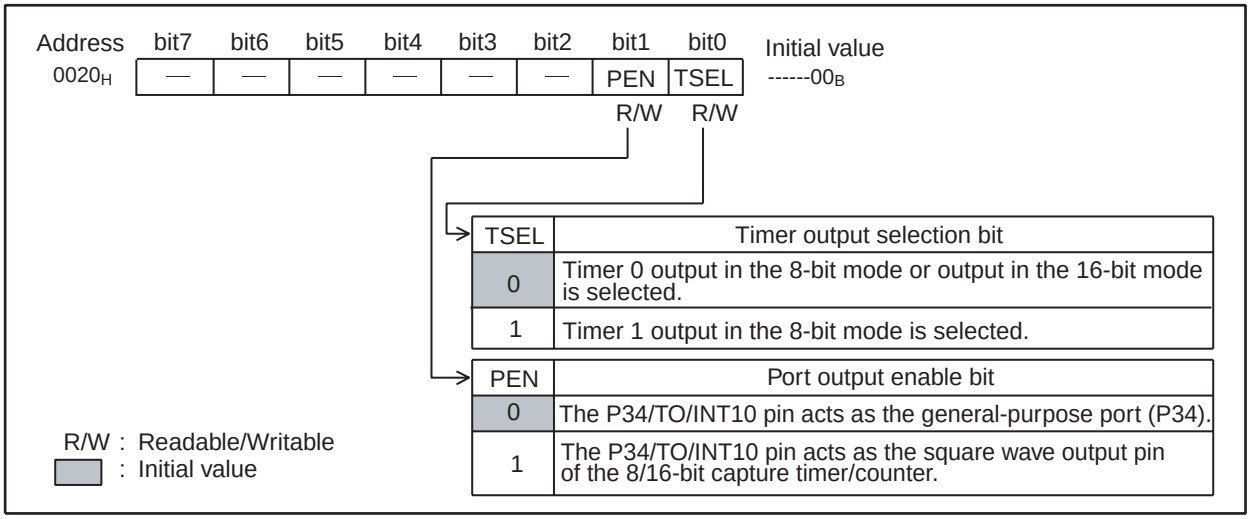


Table 8.4-4 Explanation of Functions of Each Bit in Timer Output Control Register (TCR2)

Bit name		Function
bit7 to bit2	Unused bits	• At read, the values of these bits are undefined. • At write, these bits do not affect operation.
bit1	PEN: Port output enable bit	• When this bit is "0", the P34/TO/INT10 pin acts as the general-purpose port (P34). When "1", it acts as the square wave output pin (TO).
bit0	TSEL: Timer output selection bit	• When this bit is "0", timer 0 output in the 8-bit mode or output in the 16-bit mode is selected. In this case, data is output from the TO pin. When "1", timer 1 output in the 8-bit mode is selected. In this case, data is also output from the TO pin.

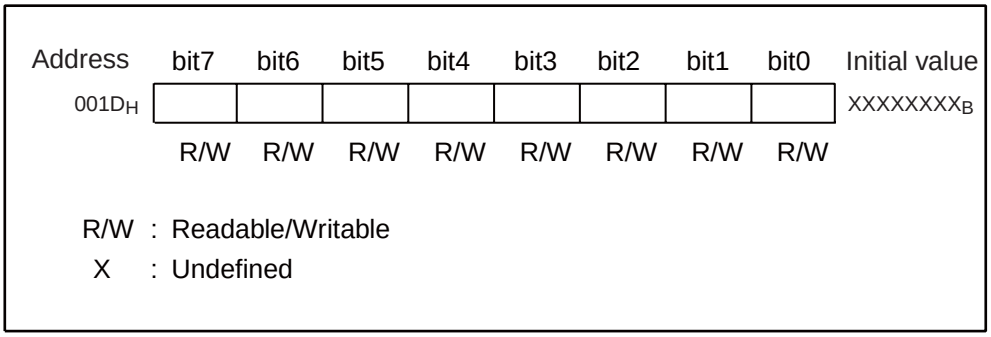
8.4.5 Timer 0 Data Register (TDR0)

The timer 0 data register (TDR0) is used to set the timer 0 value in the 8-bit mode of the 8/16-bit capture timer/counter or the interval timer value (interval timer function) or counter value (counter function) of the lower 8 bits in 16-bit mode.

■ Timer 0 Data Register (TDR0)

The values set in this register are compared with those set in the counter. Figure 8.4-6 shows the bit structure of timer 0 data register (TDR0).

Figure 8.4-6 Timer 0 Data Register (TDR0)



● 8-bit mode (timer 0)

The values set in this register are compared with those set in the counter. When the interval timer function is used, an interval timer value is set. When the counter function is used, the count value to be detected is set. When the count operation is allowed (TCR0: TSTR0 = 0 →1), the value in TDR0 is set in (loaded to) the comparator data latch and the counter is incremented.

When the values in the comparator data latch match those in the counter as a result of the increment, the values in the TDR0 are reset in the comparator data latch, the counter is cleared, and the count operation is continued.

The comparator data latch is reset when a match is detected, thus, the values written to the TDR0 when the counter is in operation become valid from the next cycle (after match detection).

Note:

The values set in TDR0 when the interval timer is in operation can be calculated from the expression shown below. However, the instruction cycle is affected by the clock mode and gear function.

Values set in TDR0 = interval time/(count clock cycle × instruction cycle) - 1

- 16-bit mode

  The values in TDR0 are compared with the counter values in the lower 8 bits of the 16-bit timer.

  When the interval timer function is used, the lower 8 bits of the interval time are set. When the counter function is used, the lower 8 bits of the count value to be detected are set. The values in TDR0 are loaded to the lower 8 bits of the comparator data latch when matching the counter values of the 16-bit timer or when the count operation is started. The values written to TDR0 when the 16-bit counter is in operation become valid after match detection.

  For the values set in TDR1 when the interval timer function is used, see Section "8.4.6 Timer 1 Data Register (TDR1) ".

8.4.6 Timer 1 Data Register (TDR1)

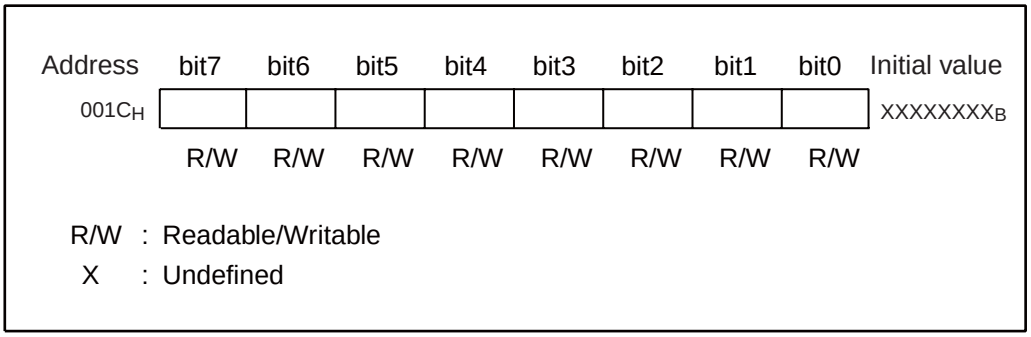
The timer 1 data register (TDR1) is used to set the timer 1 value in the 8-bit mode of the 8/16-bit capture timer/counter or the interval timer value (interval timer function) or counter value (counter function) of the higher 8 bits in the 16-bit mode.

■ Timer 1 Data Register (TDR1)

The values set in this register are compared with those set in the counter.

Figure 8.4-7 shows the bit structure of timer 1 data register (TDR1).

Figure 8.4-7 Timer 1 Data Register (TDR1)



● 8-bit mode (timer 1)

The values set in this register are compared with those set in the counter. When the interval timer function is used, an interval timer value is set. When the counter function is used, the count value to be detected is set. The values in TDR1 are reset in (loaded to) the comparator data latch when they match the values in the counter or when the count operation is started.

The values written to TDR1 when the counter is operating become valid from the next cycle (after match detection).

Note:

The values set in TDR1 when the interval timer is operating can be calculated from the expression shown below. However, the instruction cycle is affected by the clock mode and gear function.

Values set in TDR1 = interval time/(count clock cycle × instruction cycle) - 1

- 16-bit mode

  The values in TDR1 are compared with the counter values in the higher 8 bits of the 16-bit timer.

  When the interval timer function is used, the higher 8 bits of the interval time are set. When the counter function is used, the higher 8 bits of the count value to be detected are set. The values in TDR1 are loaded to the higher 8 bits of the comparator data latch when matching the counter values of the 16-bit timer or when the count operation is started. The values written to TDR1 when the 16-bit counter is operating become valid after match detection. In the 16-bit mode, the count operation is controlled by the timer 0 control register (TCR0).

  Note:

  The values set in TDR0 and TDR1 when the interval function is used can be calculated from the expression shown below. However, the instruction cycle is affected by the clock mode and gear function.

  16-bit data value = interval time/(count clock cycle × instruction cycle) - 1

  The higher 8 bits of the 16-bit data value are set in TDR1 and the lower 8 bits are set in TDR0.

8.4.7 Capture Data Registers H and L (TCPH and TCPL)

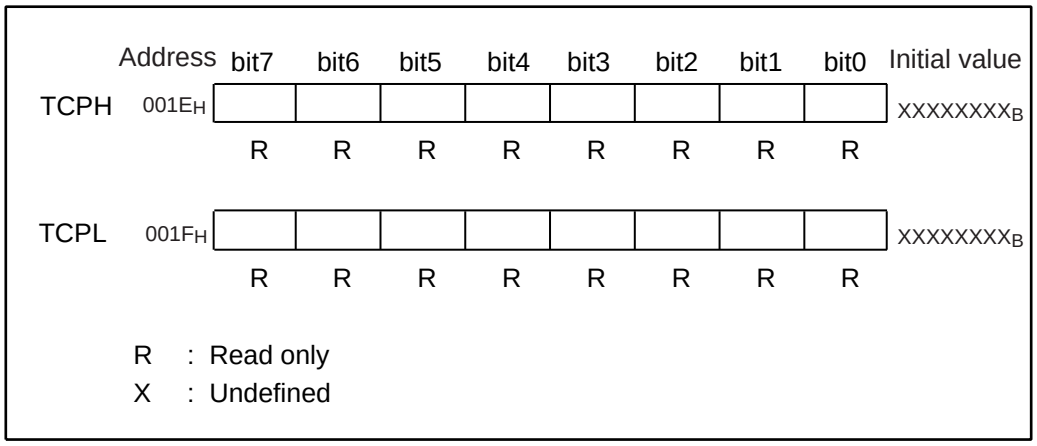
The capture data register H (TCPH) stores the number of events of the higher 8 bits in the 16-bit capture mode of the 8/16-bit capture timer/counter.
The capture data register L (TCPL) stores the number of events in the 8-bit capture mode of the 8/16-bit capture timer/counter or the number of events of the lower 8 bits in the 16-bit capture mode.
In the read operation in the timer/counter mode, counter values are read.

■ Capture Data Registers H and L (TCPH and TCPL)

The number of events detected in the capture mode is stored in TCPH and TCPL. Data cannot be written to these registers because the registers are read-only.

Figure 8.4-8 shows the bit structures of capture data registers H and L.

Figure 8.4-8 Bit Structures of Capture Data Registers H and L (TCPH and TCPL)



8.5 8/16-bit Capture Timer/Counter of Interrupts

The 8/16-bit capture timer/counter generates an interrupt if the values set in a data register match those set in the counter when the interval timer or counter is operating. The interrupt level is IRQ3 when generated by the 8/16-bit capture timer/counter. When the capture is in operation and a capture edge is detected, IRQ4 is generated.

■ 8/16-bit Capture Timer/Counter of Interrupts

Table 8.5-1 shows the interrupt request flag bit, interrupt flag clear bit, interrupt request enable bit, and the cause of the 8/16-bit capture timer/counter interruption.

Table 8.5-1 Interrupt Control Bits and the Cause of the 8/16-bit Capture Timer/Counter Interrupt

	8-bit mode		16-bit mode	Capture mode
	Timer 0	Timer 1	Timer 0 + timer 1	Timer 0 or timer 0 + timer 1
Interrupt request flag bit	TCR0 : TIF0	TCR1 : TIF1	TCR0 : TIF0	TCCR : CPIF
Interrupt flag clear bit	TCR0 : TFCR0	TCR1 : TFCR1	TCR0 : TFCR0	TCCR : CFCLR
Interrupt request enable bit	TCR0 : T0IEN	TCR1 : T1IEN	TCR0 : T0IEN	TCCR : CPIEN
Interrupt cause	The values in TDR0 match those in the 8-bit counter.	The values in TDR1 match those in the 8-bit counter.	The values in TDR0 and TDR1 match those in the 16-bit counter.	A capture edge is detected.

In the 8-bit mode, timer 0 and timer 1 independently generats the interrupt request for 8/16-bit capture timer/counter. In the 16-bit mode, timer 0 generates the interrupt request. All basic operations are the same. Timer 0 interrupt operation in the 8-bit mode is explained here.

- Timer 0 interrupt operation in the 8-bit mode

The counter value is incremented according to the selected count clock, starting at 00_H. When the counter value matches the value set in the comparator data latch (timer 0 data register (TDR0)) corresponding to the timer 0 data register (TDR0), the compare match detection flag bit (TCR0: TIF0) is set to "1".

In this case, when the interrupt request flag bit is allowed (when TCR0: T0IEN = 1), timer 0 generates an interrupt request (IRQ3) to the CPU. Set the TFCR0 bit to "1" and clear the interrupt request with the interrupt processing routine.

When the counter value matches the value set in the comparator data latch, the TIF0 bit is set to "1" regardless of the TFCR0 bit value.

In the 8-bit mode, timer 0 and timer 1 operate independently, and because they generate the same interrupt request (IRQ3), determination of the interrupt request flag by software may be required.

Notes:

- When the counter value matches the TDR0 value and at the same time the counter stops (TCR0: TSTR0 = 0), the TIF0 bit is not set.
  If the T0IEN bit is set to "1" (enable) when the TIF0 bit is "1", an interrupt request is generated immediately.
- If the compare register value is 0000_H or 00_H, the 8/16-bit capture timer/counter cannot generate an interrupt. Therefore, when using interrupts, set a value greater than or equal to 0001_H or 01_H. The 8/16-bit capture timer/counter also cannot generate an interrupt if the counter function detects the 0000_H or 00_H width.

■ Register and Vector Table Related to 8/16-bit Capture Timer/Counter of Interrupts

Table 8.5-2 Register and Vector Table Related to 8/16-bit Capture Timer/counter of Interrupts

	Interrupt name	Interrupt level setting register			Vector table address	
		Register	Setting bit		Higher	Lower
Timer/counter function	IRQ3	ILR1 ($007B_H$)	L31 (bit7)	L30 (bit6)	$FFF4_H$	$FFF5_H$
Capture function	IRQ4	ILR2 ($007C_H$)	L41 (bit1)	L40 (bit0)	$FFF2_H$	$FFF3_H$

For interrupt operation, see Section "3.4.2 Steps in the Interrupt Operation ".

8.6 Explanation of Operations of Interval Timer Functions

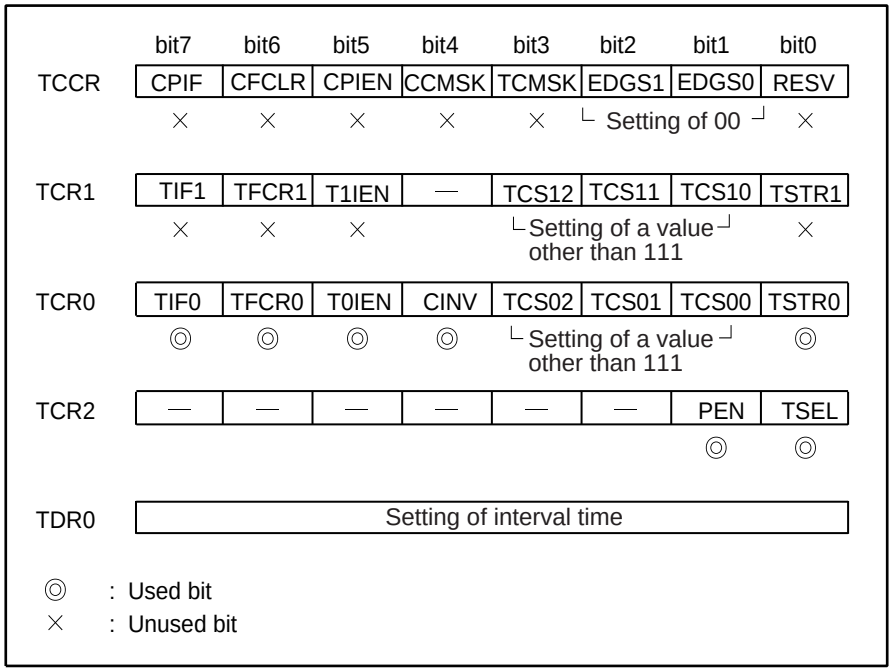
This section describes the interval timer function operation of the 8/16-bit capture timer/counter.

■ Interval Timer Function Operation

● 8-bit mode

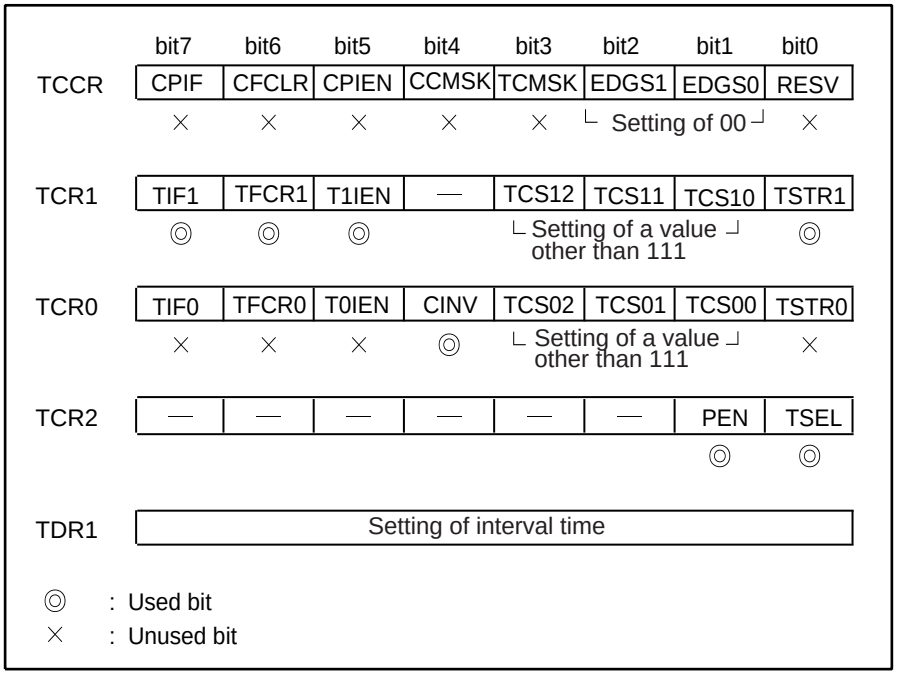
To operate timer 0 as the interval timer function in the 8-bit mode, the function must be set as shown in Figure 8.6-1 .

Figure 8.6-1 Setting of Interval Timer Function (Timer 0)



To operate timer 1 as the interval timer function in the 8-bit mode, the function must be set as shown in Figure 8.6-2 .

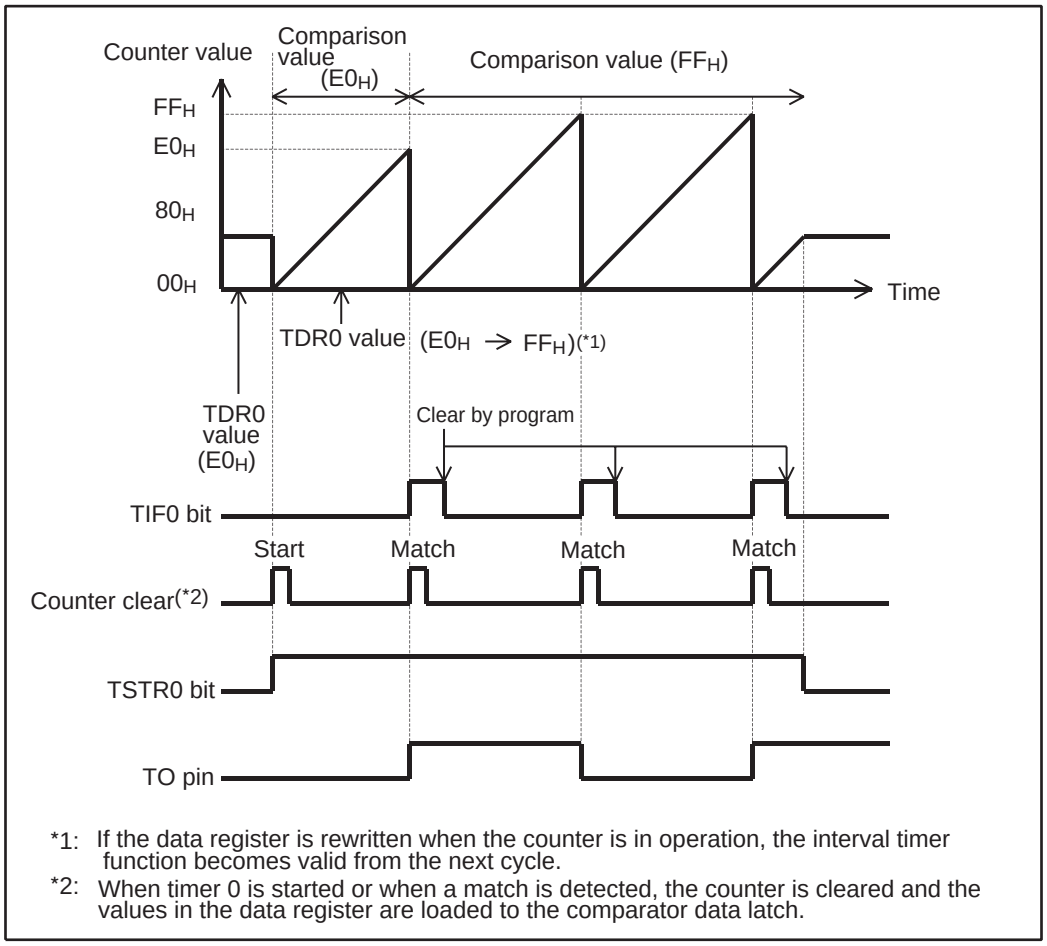
Figure 8.6-2 Setting of Interval Timer Function (Timer 1)



When the counter is activated in the 8-bit mode, increment begins at the rising or falling edge of the selected clock, starting at 00_H. When the counter value matches the value set in the data register (comparator data latch), the interrupt request bit (TCR0: TIF0 or TCR1: TIF1) of the timer 0 control register is set to "1" and the count operation is started at 00_H. If the counter value matches the value set in the data register when timer 0 is being used, the output of the square wave output control circuit toggles. When square wave output is allowed (TCR2: PEN) and timer 0 is set to output selection (TCR2: TSEL = 0), a square wave is output from the timer output pin (TO). If the counter value matches the value set in the data register when timer 1 is being used, the output of the square wave output control circuit toggles. When square wave output is allowed (TCR2: PEN) and timer 1 is set to output selection (TCR2: TSEL = 1), a square wave is output from the timer output pin (TO).

Figure 8.6-3 shows interval timer function operation in the 8-bit mode.

Figure 8.6-3 Interval Timer Function Operation in 8-bit Mode (Timer 0)



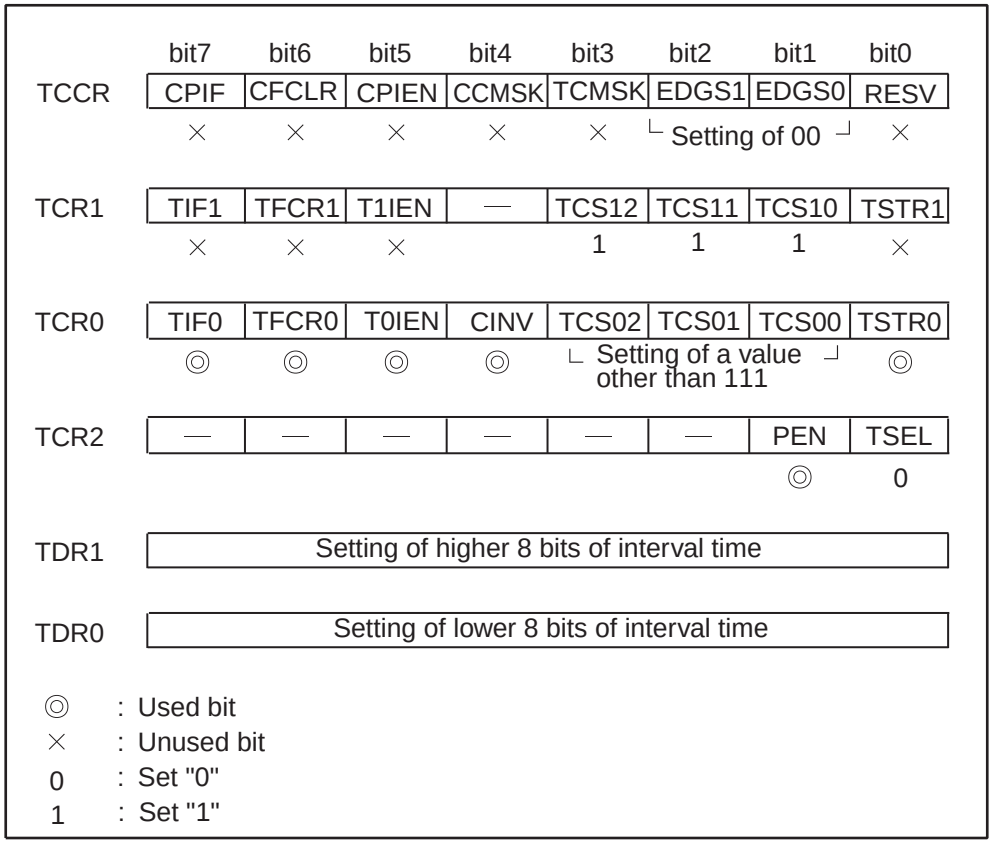
*1: If the data register is rewritten when the counter is in operation, the interval timer function becomes valid from the next cycle.

*2: When timer 0 is started or when a match is detected, the counter is cleared and the values in the data register are loaded to the comparator data latch.

● 16-bit mode

To operate timer 0 as the interval timer function in the 16-bit mode, the function must be set as shown in Figure 8.6-4 .

Figure 8.6-4 Setting of Interval Timer Function in 16-bit Mode



In the 16-bit mode, timers are controlled by the timer 0 control register (TCR0), but the timer 1 control register (TCR1) must be initialized. The values to be set in the data register are the higher 8 bits of TDR1 and the lower 8 bits of TDR0 (16 bits in total). The values are compared with the 16-bit counter value. The 16 bits of the counter are cleared at the same time. Other operations in the 16-bit mode are the same as timer 0 operation in the 8-bit mode.

8.7 Operation of Counter Functions

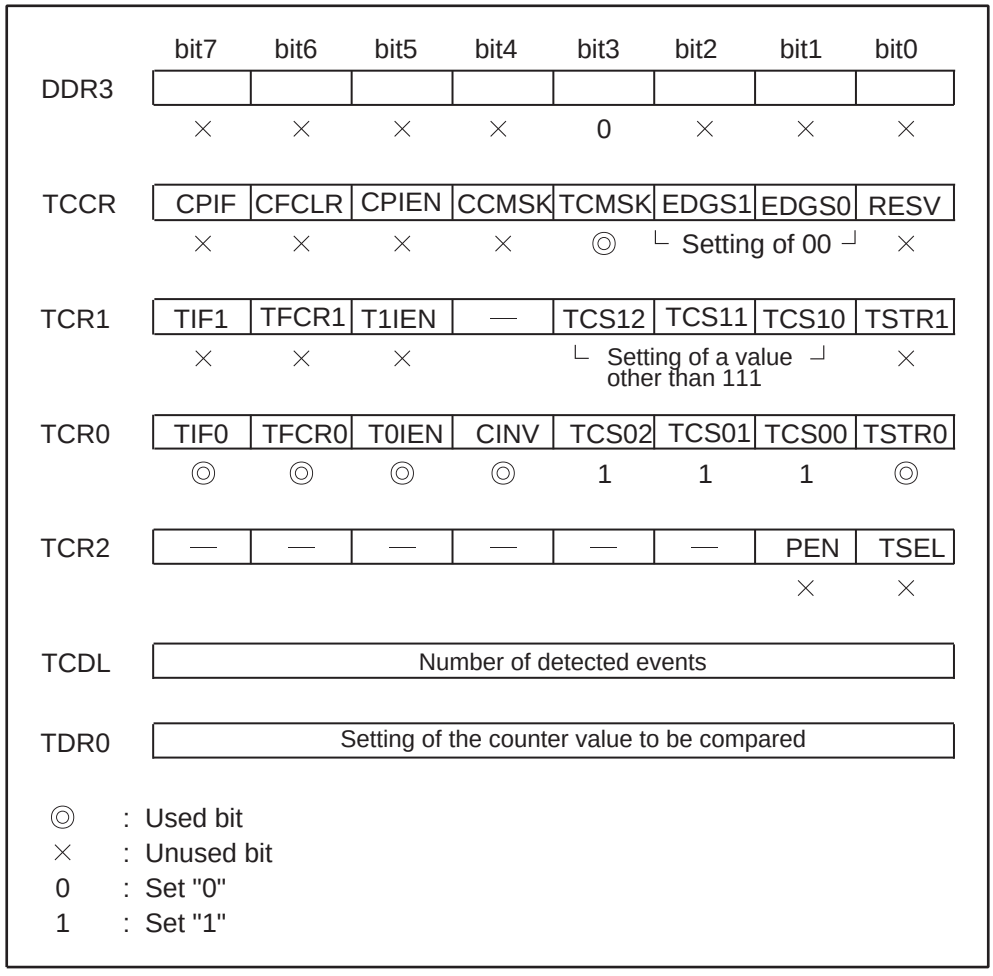
This section describes the operation of the 8/16-bit capture timer/counter function.

■ Counter Function Operation

● 8-bit mode

To operate timer 0 as the counter function in the 8-bit mode, the function must be set as shown in Figure 8.7-1 .

Figure 8.7-1 Setting of Counter Function in 8-bit Mode



Counter function operation in the 8-bit mode is the same as interval timer function (timer 0 in 8-bit mode) operation except that the external clock is used instead of the internal clock.

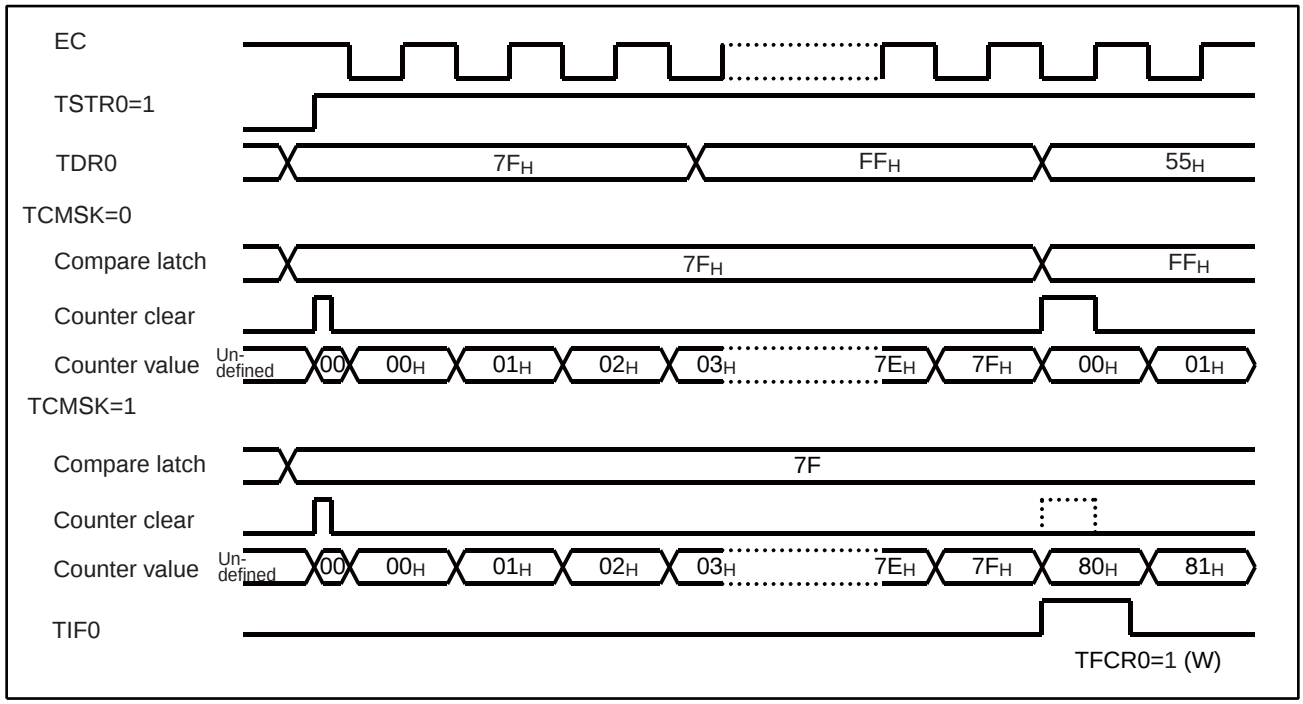
The number of events can be known by reading the capture data register (TCPL). A specific number of events can be known by the event count detection function.

- Detection of the number of events

In the external clock mode, counter clear can be prohibited by the compare match counter clear mask bit (TCMSK) of the capture control register (TCCR) when a match is detected. Setting the compare match counter clear mask bit to "1" enables the event count detection function to be used. In this case, a compare match does not cause data to be re-loaded to the compare latch. To update the compare latch value, stop and restart the timer.

Figure 8.7-2 shows counter function operation in the external clock mode in which TCMSK is used.

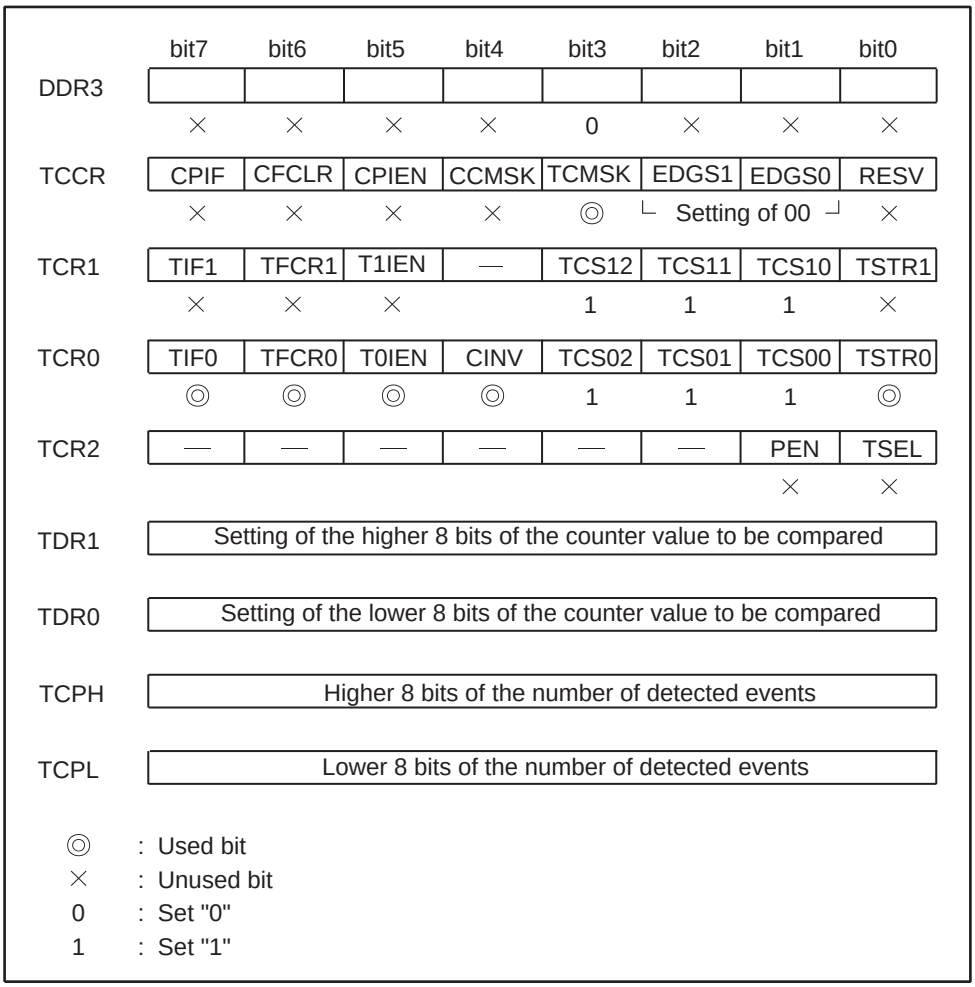
Figure 8.7-2 Counter Function Operation in External Clock Mode



● 16-bit mode

To operate timer 0 as the counter function in the 16 bit mode, the function must be set as shown in Figure 8.7-3 .

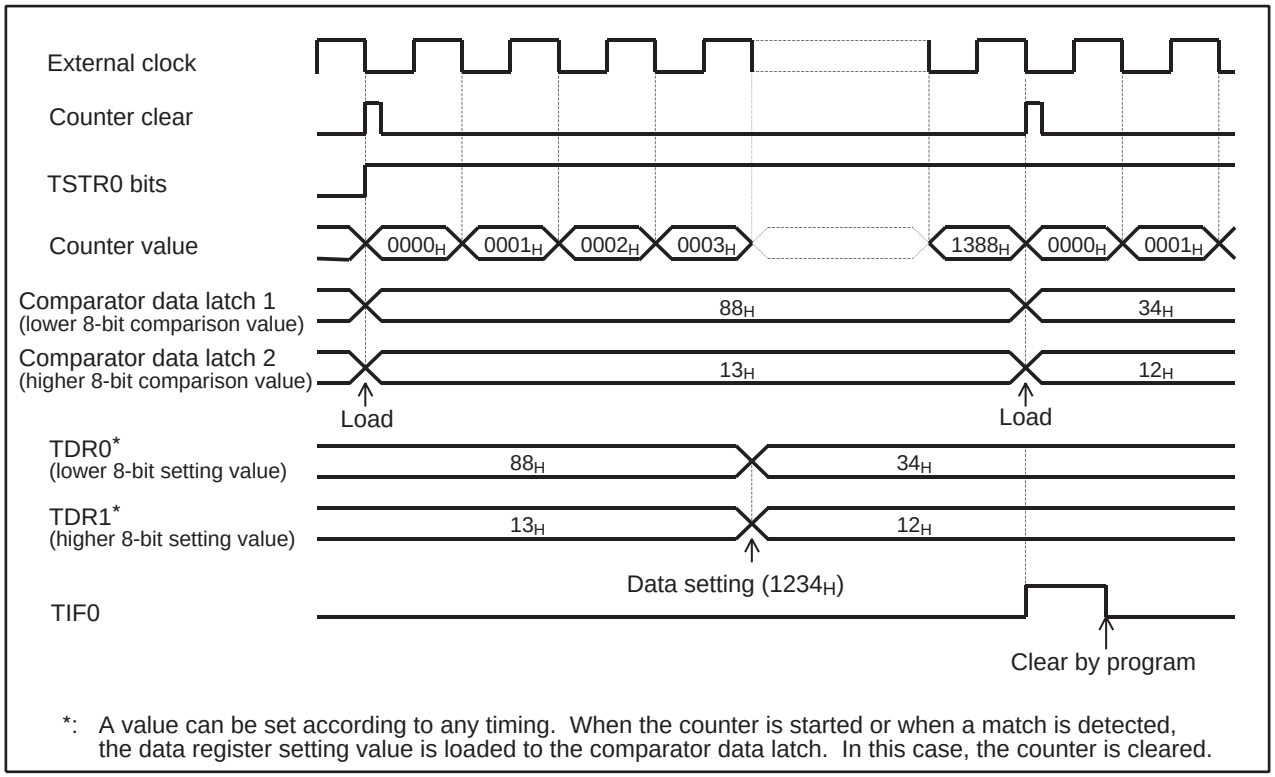
Figure 8.7-3 Setting of Counter Function in 16-bit Mode



Counter function operation in the 16-bit mode is the same as interval timer function operation in the 16-bit mode, except that the external clock is used instead of the internal clock.

Figure 8.7-4 shows counter function operation in 16-bit mode.

Figure 8.7-4 Counter Function Operation in 16-bit Mode



Note:

Confirm the validity of the values set in the counter operating in 16-bit mode.

8.8 Functions of Operations of Capture Functions

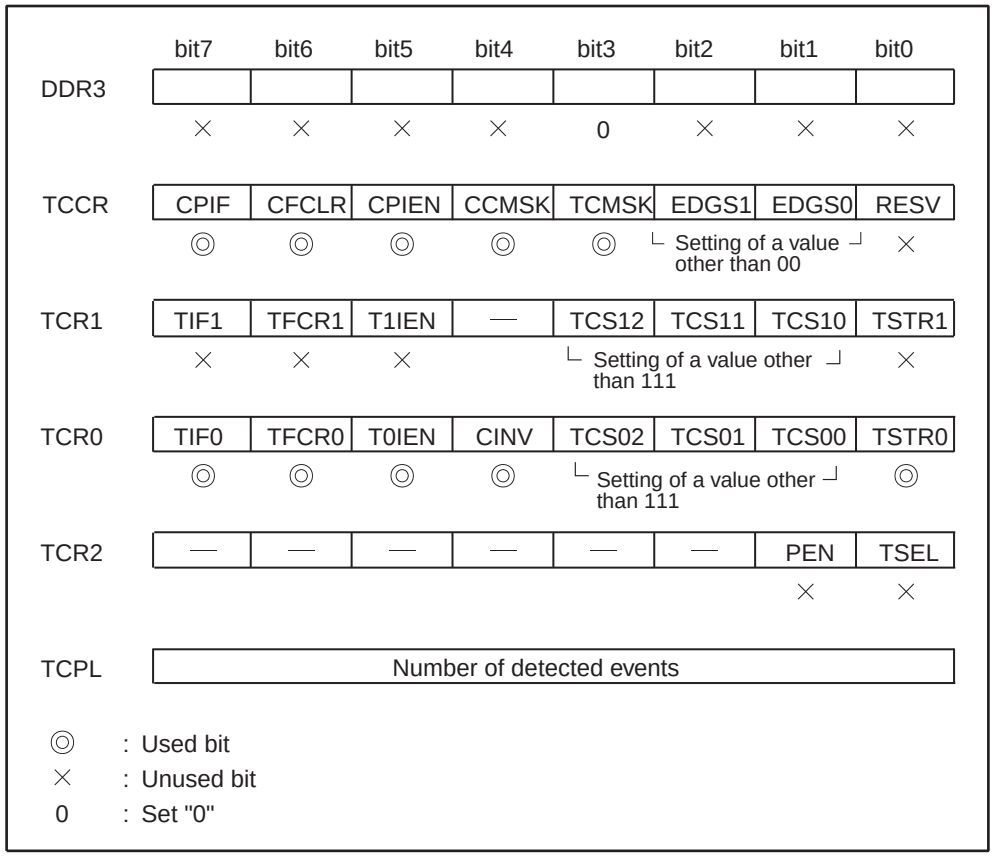
This section describes the capture function operation of the 8/16-bit capture timer/counter.

■ Capture Function Operation

● 8-bit mode

To operate the capture function in the 8-bit mode, the function must be set as shown in Figure 8.8-1 .

Figure 8.8-1 Setting of Capture Function in 8-bit Mode



The 8-bit capture mode is allowed by the capture mode enable/edge detection selection bits (EDGS1 and EDGS0) of the capture control register (TCCR). "1" is written to the timer start bit (TSTR0) after the clock source selection bits (TCS02, TCS01, and TCS00) of the timer 0 control register (TCR0) have been set.

In the capture mode, the count value is captured to the capture data register (TCPL) each time a capture input edge is detected and the capture edge detection flag (CPIF) is set to "1". In this case, if the capture interrupt enable bit (CPIEN) is already set to "1", an interrupt request is output to the CPU.

The capture mode is divided into free-run mode and clear mode.

- Free-run mode

  Setting the clear mask bits (CCMSK and TCMSK) of TCCR to 11_B enables the capture function to operate as the free-run timer.

- Clear mode

  Setting the clear mask bits (CCMSK and TCMSK) of TCCR to a value other than 11_B enables the capture function to operate as a clear mode.

  The clear mode enables the measurement of signal pulse widths and cycles. In this case, using the clear mode with the compare match detection function also enables the determination of signal availability.

  Note:

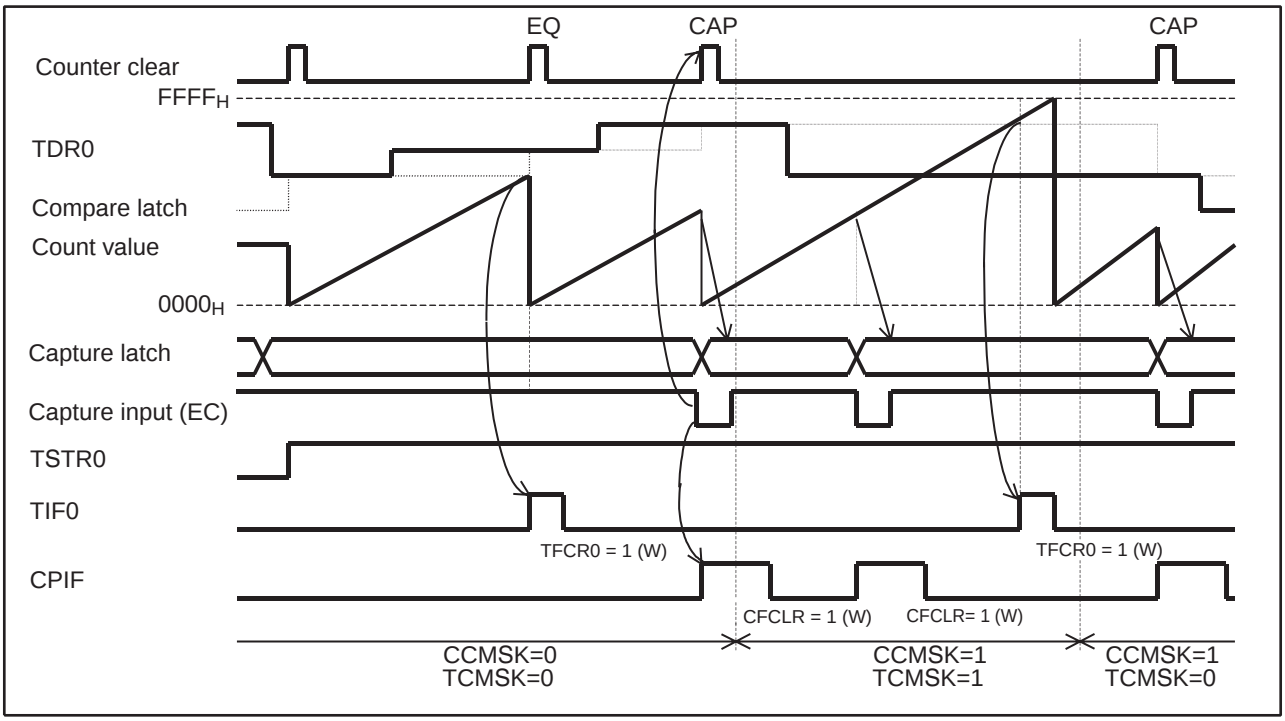
  The capture input pin also serves as the external clock input pin. The external clock mode cannot be used in the capture mode.

  Table 8.8-1 shows the relationship between the counter mode and the compare latch operation according to the clear mask bit value.

Table 8.8-1 Relationship between Counter Mode and Compare Latch Operation

CCMSK	TCMSK	Counter mode	Data load to compare latch and counter clear (provided/not provided)			
			At capture edge detection		At compare match	
			Data load	Counter clear	Data load	Counter clear
0	0	Clear mode	Provided	Provided	Provided	Provided
0	1		Provided	Provided	Not provided	Not provided
1	0		Not provided	Not provided	Provided	Provided
1	1	Free-run mode	Not provided	Not provided	Not provided	Not provided

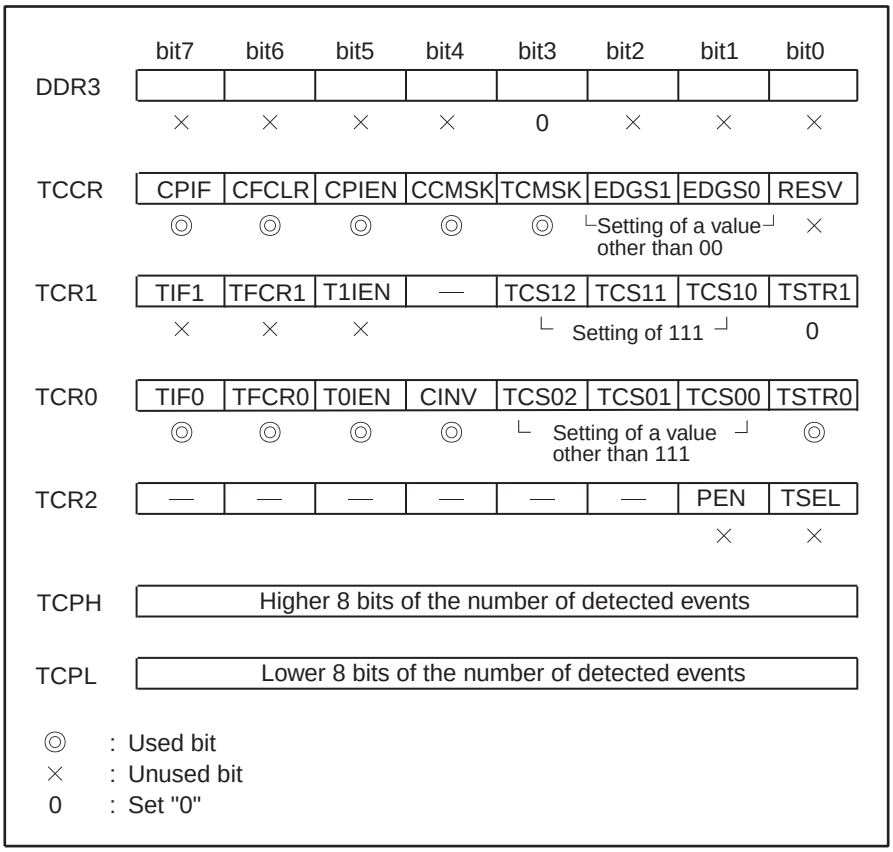
Figure 8.8-2 Capture Mode Operation



● 16-bit mode

To operate the capture function in the 16-bit mode, the function must be set as shown in Figure 8.8-3 .

Figure 8.8-3 Setting of Capture Function in 16-bit Mode



To set the 16-bit capture mode, set the TCS12, TCS11, and TCS10 bits of the timer 1 control register (TCR1) to 111_B.

In the 16-bit mode, timers are controlled by the timer 0 control register (TCR0). The higher 8 bits of the number of detected events are stored in the capture data register H (TCPH), and the lower 8 bits are stored in the capture data register L (TCPL).

For operation in the 16-bit mode, see operation in the 8-bit mode.

8.9 8/16-bit Capture Timer/Counter Operation in Each Mode

This section describes the operation of the 8/16-bit capture timer/counter when it switches to the sleep or stop mode or when a halfway stop request is issued during the operation of the interval timer or counter function.

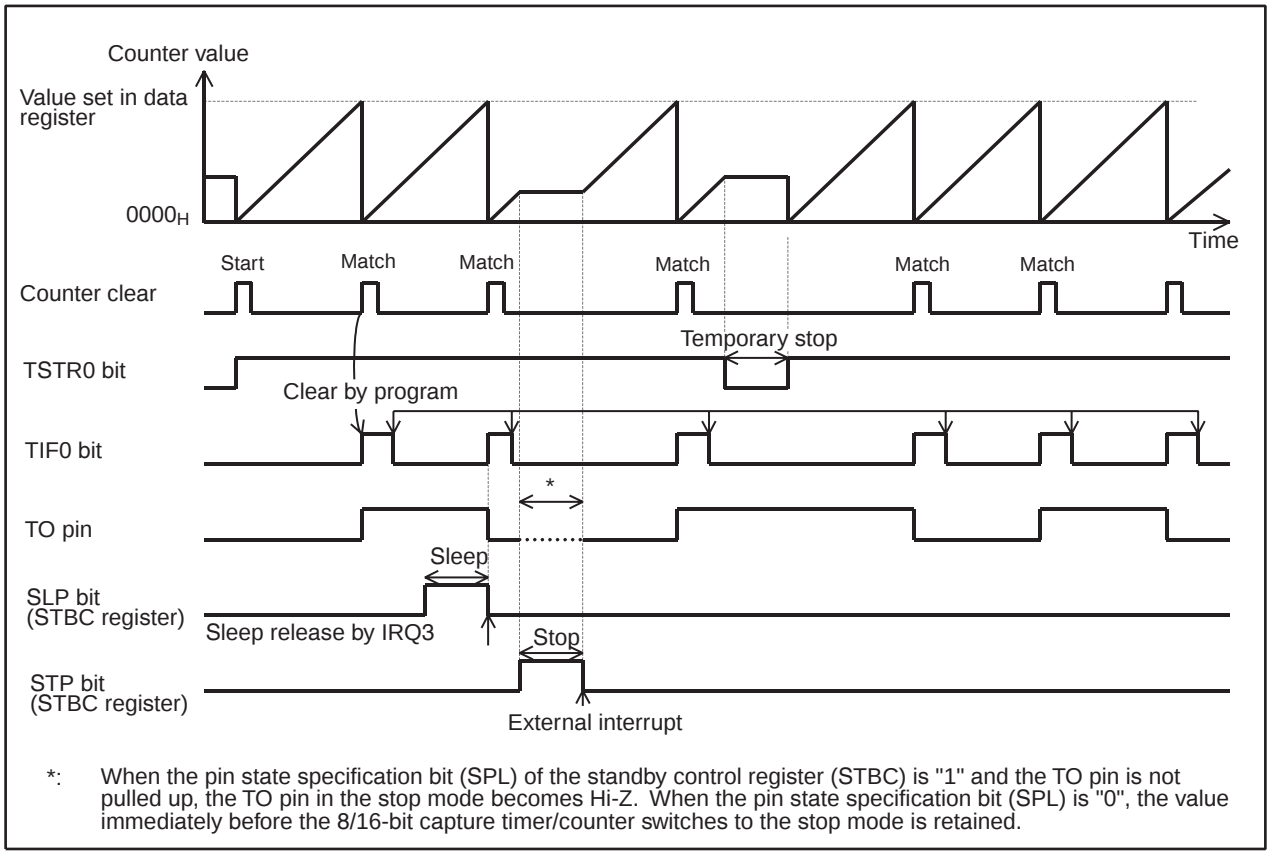
■ Operation in Standby Mode and at Halfway Stop

Figure 8.9-1 shows the counter value states if the 8/16-bit capture timer/counter switches to the sleep or stop mode or when a halfway stop request is issued when the interval timer or counter function is in operation (at timer 0 operation).

When the counter switches to the stop mode, it retains the value and stops. If the stop mode is released by an external interrupt, the counter starts its operation at the retained value, and so the first interval time and external clock count are incorrect. When the stop mode is released, the 8/16-bit capture timer/counter must be initialized.

When the counter is temporarily stopped (TSTR0 = 0), it retains its value and stops. If the subsequent operation is continued (TSTR0 = 1), the count value is cleared and the counter is restarted.

Figure 8.9-1 Counter Operation in Standby Mode and at Halfway Stop



*: When the pin state specification bit (SPL) of the standby control register (STBC) is "1" and the TO pin is not pulled up, the TO pin in the stop mode becomes Hi-Z. When the pin state specification bit (SPL) is "0", the value immediately before the 8/16-bit capture timer/counter switches to the stop mode is retained.

8.10 Notes on Using 8/16-bit Capture Timer/Counter

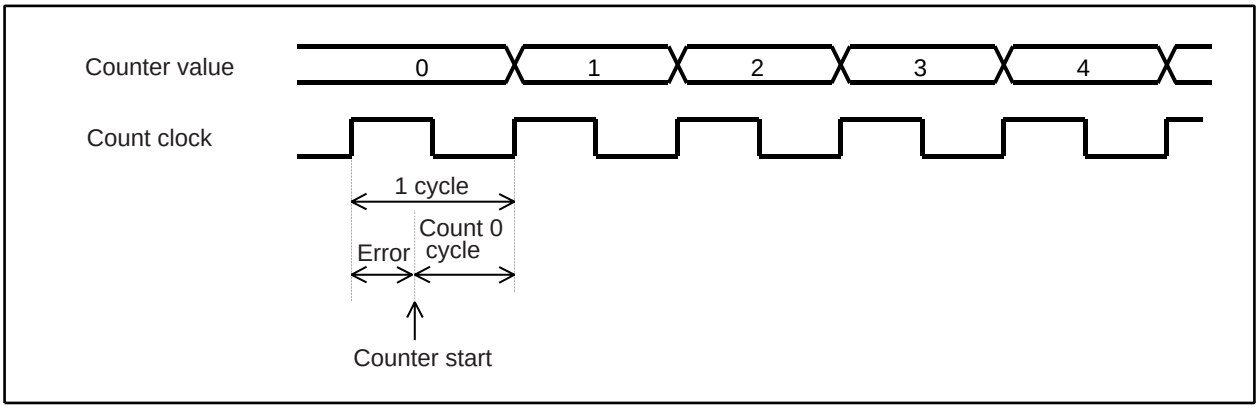
This section provides notes on using the 8/16-bit capture timer/counter.

■ Notes on Using the 8/16-bit Capture Timer/Counter

- Error

The start of the 8/16-bit capture timer/counter by a program is asynchronous with the start of the counter incremented by the selected count clock, and therefore, the error (a time difference) continues until the counter value matches the set data. Such a time difference may shorten the total count duration by a maximum of one count clock cycle. Figure 8.10-1 shows the error (a time difference) that prolongs the actual start of count operation.

Figure 8.10-1 Error Occurring until the Count Operation is Started



- Using only timer 0 in 8-bit mode

When using only timer 0 of the 8/16-bit capture timer/counter in the 8-bit mode, set a value other than 111_B in the count clock selection bits (TCS12, TCS11, TCS10) of the timer 1 control register (TCR1). Using timer 0 without setting 111_B results in a malfunction.

- Note on setting by program

When using the 8/16-bit capture timer/counter in the 16-bit mode, set the count clock selection bits (TCS12, TCS11, TCS10) of TCR1 to 111_B.

Before using the counter values when the counter is in operation with 16-bit mode, be sure to read the counter values twice and confirm that the values are valid.

Even if square wave output is initialized when the timer is in operation (TCR0: TSTR0 = 1), the output value is not modified. The output state is initialized when the timer operation stops.

When the interrupt request flag bits (TCCR: CPIF, TCR0: TIF0, TCR1: TIF1) are "1" and the interrupt request enable bits are allowed (TCCR: CPIEN, TCR0: T0IEN, and TCR1: T1IEN = 1), return from an interrupt is impossible. In this case, clear the interrupt request flag bits (TCCR: CFCLR = 1, TCR0: TFCR0 = 1, and TCR1: TFCR1 = 1).

When the counter operation stops according to the timer start bits (TCR0: TSTR0 = 0 and TCR1: TSTR1 = 0) and the interrupt source occurs at the same time, the interrupt request flag bits (TCR0: TIF0 and TCR1: TIF1) are not set.

In the capture mode, no external clock can be selected; set the count clock bits (TCS12,TCS11, and TCR1: TCS10) to a value other than 111_B.

- Note on using interrupts

If the compare register value is 0000_H or 00_H, the 8/16-bit capture timer/counter cannot generate interrupts. For this reason, when using interrupts, set a value greater than or equal to 0001_H or 01_H. The 8/16-bit capture timer/counter cannot generate interrupts if the capture counter function detects the 0000_H or 00_H width.

8.11 Program Example for 8/16-bit Capture Timer/Counter

This section provides program examples of the 8/16-bit capture timer/counter.

■ Program Example of Interval Timer Function

● Processing specifications

- In the 8-bit mode, only timer 0 is used to generate a 20 ms interval timer interrupt.
- When the interval time has elapsed, the square wave to be inverted is output to the TO pin.
- At 12.5-MHz oscillation (F_{CH}), the TDR0 value whose interval time becomes 20 ms at the maximum gear speed (1 instruction cycle = $4/F_{CH}$) is shown below. The count clock is $256t_{INST}$ of the internal count clock.

  TDR0 value = 20 ms/(256 × 4/12.5 MHz) - 1 = 244 ($F4_H$)

● Coding example

```
TCCR    EQU     0019H               ;  Address of capture control register
TCR1    EQU     001AH               ;  Address of timer 1 control register
TCR0    EQU     001BH               ;  Address of timer 0 control register
TCR2    EQU     0020H               ;  Address of timer output control register
TDR1    EQU     001CH               ;  Address of timer 1 data register
TDR0    EQU     001DH               ;  Address of timer 0 data register

TIF0    EQU     TCR0:7              ;  Defines the timer 0 interrupt request flag bit.
ILR1    EQU     007BH               ;  Address of interrupt request setting register
INT_V   DSEG    ABS                 ;  [DATA SEGMENT]
        ORG     0FFF0H
IRQD    DW      WARI                ;  Sets an interrupt vector.
        ENDS
;----------------------Main program----------------------------------------------------------------------------
        CSEG                        ;  [CODE SEGMENT]
                                    ;  The stack pointer (SP), etc., is already initialized.
          :
        CLRI                        ;  Disables the interrupt.
        MOV     ILR1,#10111111B     ;  Sets the interrupt level to 2.
        MOV     TCR0,#01001010B     ;  Clears the timer 0 interrupt request flag, increments the
                                    ;  counter at a rising edge, selects 256tINST, and stops the
                                    ;  operation.
        MOV     TCR1,#01000010B     ;  Clears the timer 1 interrupt request flag, prohibits
                                    ;  interrupt request output, sets a mode other than the 16-
                                    ;  bit mode, and stops the operation.
        MOV     TDR0,#F4H           ;  Sets the value (interval time) to be compared with the
                                    ;  counter value.
```

```
        MOV     TCR2,#00000010B     ;  Outputs a square wave (TO) from the P34 pin.
        MOV     TCR0,#10101011B     ;  Allows timer 0 interrupt request output, clears the
                                    ;  counter, and starts the timer.
        SETI                        ;  Enables the CPU interrupt.
          :
;-----------------------Interrupt program-----------------------------------------------------------------------
WARI    CLRB    TIF0                ;  Clears the interrupt request flag.
        PUSHW   A
        XCHW    A,T
        PUSHW   A
          :
        User processing
        POPW    A
        XCHW    A,T
        POPW    A
        RETI
        ENDS
; ---------------------------------------------------------------------------------------------------------------
        END
```

■ Program Example of Counter Function

● Processing specifications

- In the 16-bit mode, timer 0 and timer 1 are used to generate an interrupt whenever the external clock to be input to the EC pin is counted 5,000 times (1388_H).
- The sample program for reading the 16-bit counter value when the counter is in operation is shown below (READ16).

● Coding example

```
DDR3    EQU     000DH               ; Address of port 3-direction register
TCCR    EQU     0019H               ; Address of capture control register
TCR1    EQU     001AH               ; Address of timer 1 control register
TCR0    EQU     001BH               ; Address of timer 0 control register
TDR1    EQU     001CH               ; Address of timer 1 data register
TDR0    EQU     001DH               ; Address of timer 0 data register
TIF0    EQU     TCR0:7              ; Defines the timer 0 interrupt request flag bit.
ILR1    EQU     007CH               ; Address of interrupt level setting register 2
INT_V   DSEG    ABS                 ; [DATA SEGMENT]
        ORG     0FFF0H
IRQD    DW      WARI                ; Sets the interrupt vector.
        ENDS
;-----------------------Main program-----------------------------------------------------------------------
        CSEG                        ; [CODE SEGMENT]
                                    ; The stack pointer (SP), etc., is already initialized.
         :
        MOV     DDR3,#00000000B     ; Sets the EC pin to input.
        CLRI                        ; Disables the interrupt.
        MOV     ILR1,#10111111B     ; Sets the interrupt level to 2.
        MOV     TDR0,#088H          ; Sets the counter value and the lower 8 bits of the
                                      compare value.
        MOV     TDR1,#013H          ; Sets the counter value and the higher 8 bits of the
                                      compare value.
        MOV     TCR1,#00001110B     ; Sets timer 1 to 16-bit mode.
        MOV     TCR0,#01101111B     ; Clears the timer 0 interrupt request flag, allows interrupt
                                      request output, selects an external clock, clears the
                                      counter, starts the operation, and increments the counter
                                      at a rising edge.
        SETI                        ; Enables a CPU interrupt.
;-----------------------Data read subroutine---------------------------------------------------------------
READ16
        MOVW    A,TDR1              ; Reads 16 bits from TDR1 and TDR0.
        MOVW    A,TDR1              ; Reads 16 bits from TDR1 and TDR0 and stores the old
                                      value in the T register.
        CMPW    A                   ; Executes double read check and compares A with T.
        BEQ     RET16               ; Match and return
```

```
        XCHW    A,T
        INCW    A               ; Old value + 1
        CMPW    A
        BNE     READ16          ; Jumps to re-read when a mismatch is detected.
RET16   RET                     ; Restarts the count operation and begins counting 10,000
                                  pulses.
;-----------------------Interrupt program-----------------------------------------------------------------------
WARI    CLRB    TIF0            ; Clears the interrupt request flag.
        PUSHW   A
        XCHW    A,T
        PUSHW   A
          :
        User processing
          :
        POPW    A
        XCHW    A,T
        POPW    A
        RETI
        ENDS
; ---------------------------------------------------------------------------------------------------------------
        END
```

CHAPTER 9

12-BIT PPG TIMER

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

9.1 Overview of 12-bit PPG Timer

The 12-bit PPG timer is a 12-bit binary counter, enabling the selection of one of four types of internal count clocks. The timer is capable of setting a cycle period and "H" width of output pulse waveforms and can also be used as a remote control transmission frequency generator or 12-bit PPG.

■ Functions of 12-bit PPG Timer

- The timer generates a frequency for remote control and outputs signals to a PPG pin.
- The timer is capable of setting a cycle period and "H" width of output pulse waveforms separately.
- The timer enables the selection of a count clock from four types of internal clocks.
- The timer can generate a frequency in a range from twice to 2^{12}-1 times as fast as the counter clock.

Table 9.1-1 lists the ranges in which the output pulse cycle period and "H" width are variable.

Table 9.1-1 Ranges in which the Output Pulse Cycle Period and "H" Width are Variable

Internal count clock cycle period	Output pulse cycle period	Output pulse "H" width
$2t_{INST}$	$4t_{INST}$ to $8190t_{INST}$	$2t_{INST}$ to $8188t_{INST}$
$4t_{INST}$	$8t_{INST}$ to $16380t_{INST}$	$4t_{INST}$ to $16376t_{INST}$
$16t_{INST}$	$32t_{INST}$ to $65520t_{INST}$	$16t_{INST}$ to $65504t_{INST}$
$256t_{INST}$	$512t_{INST}$ to $1048.32kt_{INST}$	$256t_{INST}$ to $1048.064kt_{INST}$

t_{INST}: Instruction cycle (to be affected by a gear function)

Note:

An example of calculating the output pulse cycle period and "H" width as executed by a 12-bit PPG function is given below.

When an oscillation (F_{CH}) of 12.5 MHz and a count clock cycle period of 2 t_{INST} are set, and if:

Compare value for cycle period = 011110_B (30-clock period)

Compare value for "H" width = 001010_B (10-clock width)

Then, "H" width and the cycle period of output pulse waveforms are calculated as given below. These calculations are obtained provided the system clock control register (SYCC) selects the fastest clock (CS1, CS0 = 11_B with one instruction cycle = $4/F_{CH}$).

Cycle period = Compare value for cycle period × Count clock cycle period

= 011110_H (30-clock period) × 2 × 4/F_{CH}

= 30 × 2 × 0.32 µs = 19.2 µs

"H" width = Compare value for "H" width × Count clock cycle period

= 001010_B (10-clock width) × 2 × 4/F_{CH}

=10 × 2 × 0.32 µs

=6.4 µs

If the set "H" width is equal to or greater than the set cycle period, "H" level outputs occur.

■ 12-bit PPG Function

The timer's programmable pulse output generator function can be used as a 12-bit PPG because it can set a cycle period and "H" pulse width of output pulse waveforms separately. A range of controllable duty cycles is 0.02% to 100%. However, the smaller the compare value for the cycle period, the lower the resolution (the greater a minimum-step duty cycle).

If the compare value for the cycle period is "2", a comparative setting of "H" pulse width is 1 or 2 (a duty cycle of 50% or 100%) and the resolution is 1/2.

An output frequency and a duty cycle can be calculated using the following equations:

Output pulse cycle period = Compare value for cycle period × Count clock cycle period

Duty cycle = Compare value for "H" width/Compare value × 100(%)

Table 9.1-2 lists available resolution values, minimum-step duty cycles, and output pulse cycle periods.

Table 9.1-2 Resolutions and Output Pulse Cycle Periods Supported when the Timer is Used as a 12-bit PPG (1/2)

Compare value for cycle period	Range of available compare values for "H" width	Output pulse cycle period				Resolution	Minimum-step duty cycle
		Count clock = 2 t_{INST}	Count clock = 4 t_{INST}	Count clock = 16 t_{INST}	Count clock = 256 t_{INST}		
0	-	Unavailable					
1	-						
2	1, 2	4 t_{INST}	8 t_{INST}	32 t_{INST}	512 t_{INST}	1/2	50.0%
3	1 to 3	6 t_{INST}	12 t_{INST}	48 t_{INST}	768 t_{INST}	1/3	33.3%
4	1 to 4	8 t_{INST}	16 t_{INST}	64 t_{INST}	1024 t_{INST}	1/4	25.0%
5	1 to 5	10 t_{INST}	20 t_{INST}	80 t_{INST}	1280 t_{INST}	1/5	20.0%
6	1 to 6	12 t_{INST}	24 t_{INST}	96 t_{INST}	1536 t_{INST}	1/6	16.7%
7	1 to 7	14 t_{INST}	28 t_{INST}	112 t_{INST}	1792 t_{INST}	1/7	14.3%
8	1 to 8	16 t_{INST}	32 t_{INST}	128 t_{INST}	2048 t_{INST}	1/8	12.5%

Table 9.1-2 Resolutions and Output Pulse Cycle Periods Supported when the Timer is Used as a 12-bit PPG (2/2)

Compare value for cycle period	Range of available compare values for "H" width	Output pulse cycle period				Resolution	Minimum-step duty cycle
		Count clock = 2 t_{INST}	Count clock = 4 t_{INST}	Count clock = 16 t_{INST}	Count clock = 256 t_{INST}		
9	1 to 9	18 t_{INST}	36 t_{INST}	144 t_{INST}	2304 t_{INST}	1/9	11.1%
10	1 to 10	20 t_{INST}	40 t_{INST}	160 t_{INST}	2560 t_{INST}	1/10	10.0%
		:	:	:	:		
20	1 to 20	40 t_{INST}	80 t_{INST}	320 t_{INST}	5120 t_{INST}	1/20	5.0%
		:	:	:	:		
100	1 to 100	200 t_{INST}	400 t_{INST}	1600 t_{INST}	25600 t_{INST}	1/100	1.0%
		:	:	:	:		
500	1 to 500	1000 t_{INST}	2000 t_{INST}	8000 t_{INST}	128000 t_{INST}	1/500	0.2%
		:	:	:	:		
1000	1 to 1000	2000 t_{INST}	4000 t_{INST}	16000 t_{INST}	256000 t_{INST}	1/1000	0.1%
		:	:	:	:		
2000	1 to 2000	4000 t_{INST}	8000 t_{INST}	32000 t_{INST}	512000 t_{INST}	1/2000	0.05%
		:	:	:	:		
3000	1 to 3000	6000 t_{INST}	12000 t_{INST}	48000 t_{INST}	768000 t_{INST}	1/3000	0.03%
		:	:	:	:		
4095	1 to 4095	8190 t_{INST}	16380 t_{INST}	65520 t_{INST}	1048320 t_{INST}	1/4095	0.02%

t_{INST}: Instruction cycle

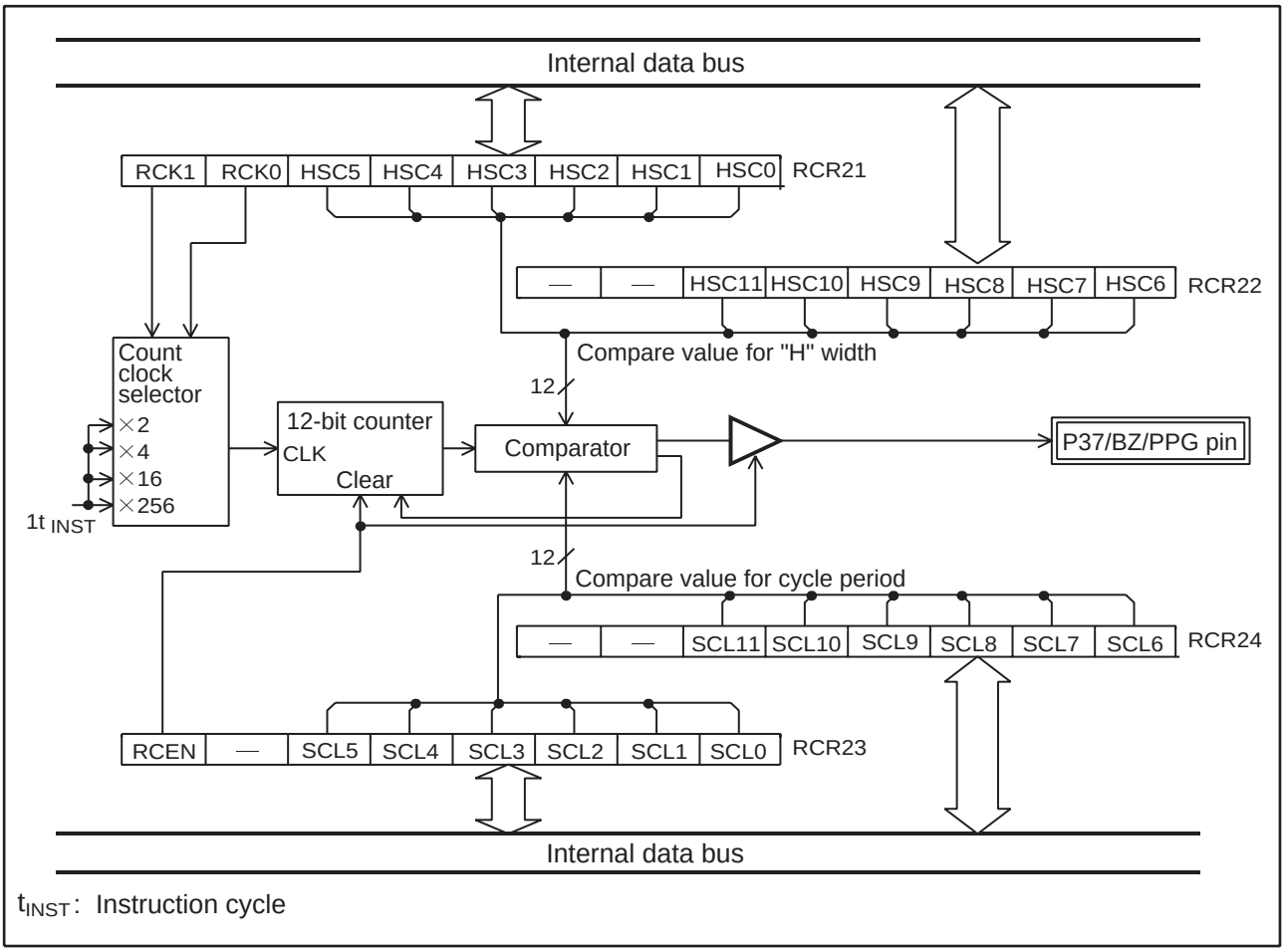
9.2 Configuration of 12-bit PPG Timer Circuit

The 12-bit PPG timer comprises the following seven blocks:

- **Count clock selector**
- **12-bit counter**
- **Comparator**
- **12-bit PPG control register 1 (RCR21)**
- **12-bit PPG control register 2 (RCR22)**
- **12-bit PPG control register 3 (RCR23)**
- **12-bit PPG control register 4 (RCR24)**

■ Block Diagram of 12-bit PPG Timer

Figure 9.2-1 Block Diagram of 12-bit PPG Timer



- Count clock selector

  This selector circuit selects one of four types of internal count clocks as the count-up clock for a 12-bit counter.

- 12-bit counter

  The 12-bit counter executes a count-up operation based on the count clock selected by the count clock selector.

  This counter may be cleared according to the value of the output enable bit of the RCR23 register (RCR23:RCEN=0).

- Comparator

  The comparator maintains outputs at "H" until a count by the 12-bit counter has been synchronized with the value of the register containing the compare value for "H" width. The comparator then maintains outputs at "L" until a count by the counter is synchronized with the value of the register containing the set cycle period. At this time, the 12-bit counter is cleared and restarts to count from "00_H".

- 12-bit PPG control registers 1 (RCR21) and 2 (RCR22)

  These registers comprise bits for count clock selection and bits for setting a compare value for the "H" width.

- 12-bit PPG control registers 3 (RCR23) and 4 (RCR24)

  These registers comprise a bit for specifying whether to enable or disable the 12-bit PPG output and bits for setting a compare value for the cycle period.

9.3 Pin of 12-bit PPG Timer

This section describes the pin associated with the 12-bit PPG timer and illustrates a block diagram of circuitry terminating at the pin.

■ Pin Associated with the 12-bit PPG Timer

The pin associated with the 12-bit PPG timer is P37/BZ/PPG pin.

- P37/BZ/PPG pin

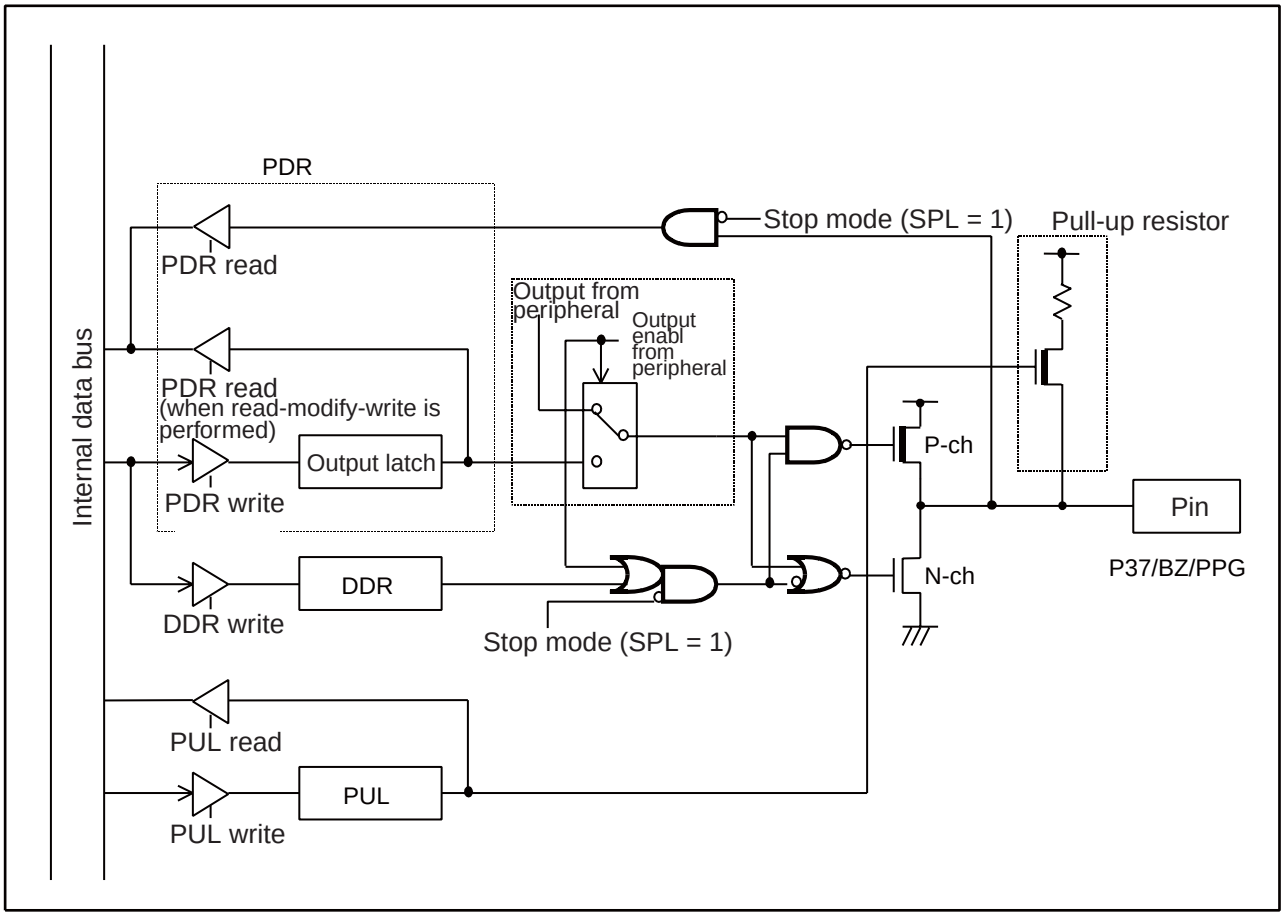
This pin functions as a CMOS type (P37) general-purpose I/O port, further functioning as 12-bit PPG timer output (PPG).

PPG:

By setting the output enable bit of the appropriate 12-bit PPG control register (RCR23:RCEN) to "1", the pin functions as the PPG output pin through which the set cycle period and "H" width of PPG pulse waveforms are output.

■ Block Diagram of Circuitry Terminating at the Pin Associated with the 12-bit PPG Timer

Figure 9.3-1 Block Diagram of Circuitry Terminating at the P37/BZ/PPG Pin



Notes:

- If the ON setting of the pull-up resistor is selected by the pull-up setting register, the pin state will be the "H" level (pull-up state) in stop mode (SPL = 1).
- Because buzzer outputs to the P37/BZ/PPG pin precede 12-bit PPG outputs to this pin, if the pin is used as the PPG pin, turn the buzzer outputs off and set the RCEN bit such that PPG outputs are enabled.

9.4 Registers of 12-bit PPG Timer

This section describes the registers associated with the 12-bit PPG timer.

■ Registers Associated with 12-bit PPG Timer

Figure 9.4-1 Registers Associated with 12-bit PPG Timer

RCR21 (12-bit PPG control register 1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0014_H	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

RCR22 (12-bit PPG control register 2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0015_H	—	—	HSC11	HSC10	HSC9	HSC8	HSC7	HSC6	$--000000_B$
			R/W	R/W	R/W	R/W	R/W	R/W	

RCR23 (12-bit PPG control register 3)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0016_H	RCEN	—	SCL5	SCL4	SCL3	SCL2	SCL1	SCL0	$0-000000_B$
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

RCR24 (12-bit PPG control register 4)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0017_H	—	—	SCL11	SCL10	SCL9	SCL8	SCL7	SCL6	$--000000_B$
			R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable and Writable
— : Unused

9.4.1 12-bit PPG Control Register 1 (RCR21)

The 12-bit PPG control register 1 comprises bits for count clock selection of the 12-bit PPG timer and bits for setting the "H" width.

■ 12-bit PPG Control Register 1 (RCR21)

Figure 9.4-2 12-bit PPG Control Register 1 (RCR21)

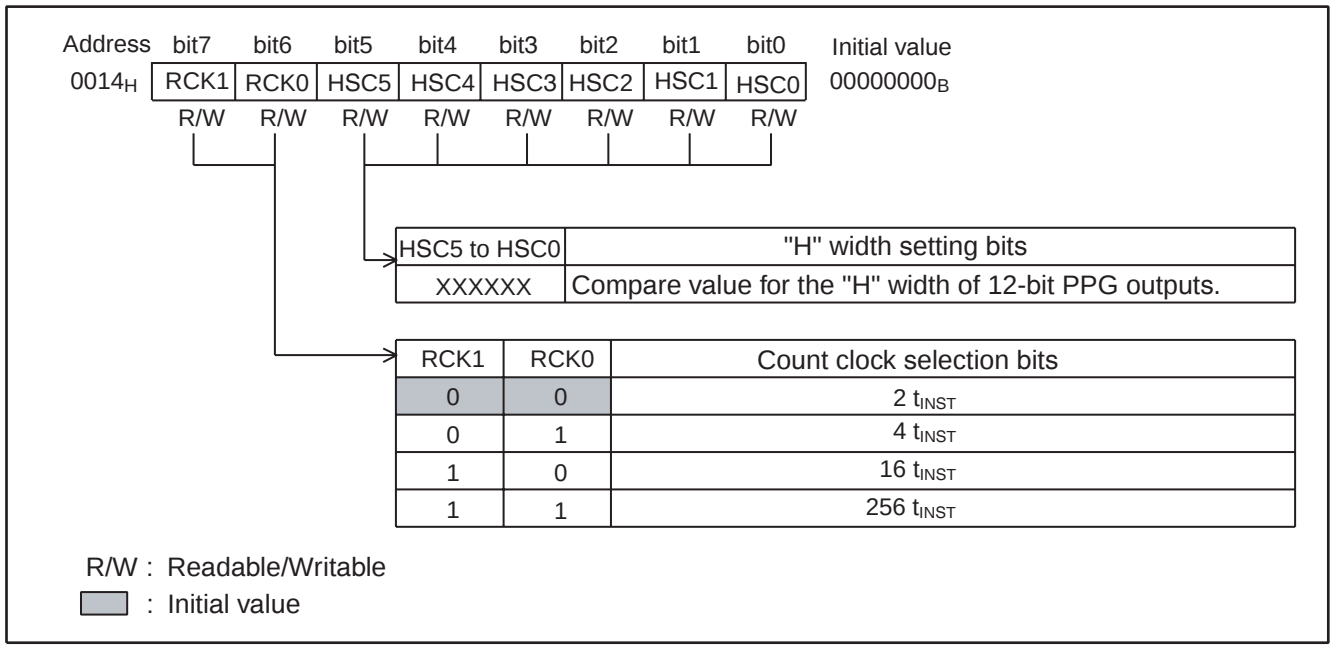


Table 9.4-1 Explanation of Functions of Each Bit in 12-bit PPG Control Register 1 (RCR21)

Bit name		Function
bit7, bit6	RCK1, RCK0: Count clock selection bits	These bits are used to select a count clock of the 12-bit PPG timer from four types of internal count clocks.
bit5 to bit0	HSC5 to HSC0: "H" width setting bits	These bits are used to set the number of counts corresponding to the "H" width of 12-bit PPG timer outputs (the compare value for the "H" width), and the contents of these bits and the HSC6 to HSC11 bits of the RCR22 register are compared with a count by the counter.

9.4.2 12-bit PPG Control Register 2 (RCR22)

The 12-bit PPG control register 2 comprises bits for setting the "H" width of 12-bit PPG pulse waveforms.

■ 12-bit PPG Control Register 2 (RCR22)

Figure 9.4-3 12-bit PPG Control Register 2 (RCR22)

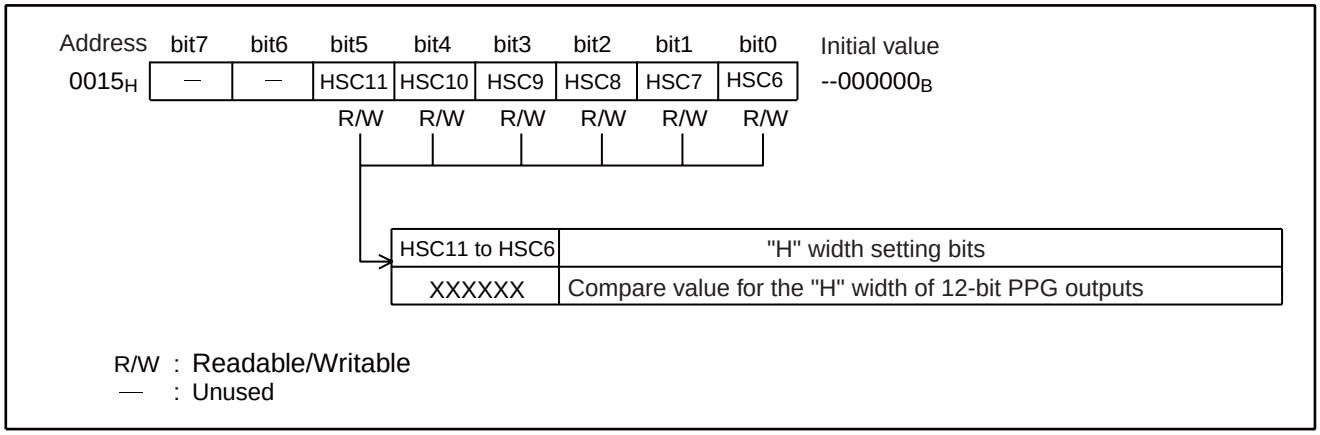


Table 9.4-2 Explanation of Functions of Each Bit in 12-bit PPG Control Register 2 (RCR22)

Bit name		Function
bit7, bit6	Unused bits	• Bit value is undefined when being read. • Written value does not affect other operations.
bit5 to bit0	HSC11 to HSC6: "H" width setting bits	These bits are used to set the number of counts corresponding to the "H" width of 12-bit PPG timer outputs (the compare value for the "H" width), and the contents of these bits and the HSC0 to HSC5 bits of the RCR21 register are compared with a count by the counter.

9.4.3 12-bit PPG Control Register 3 (RCR23)

The 12-bit PPG control register 3 comprises a bit for enabling 12-bit PPG waveform outputs and bits for setting a cycle period of outputs.

■ 12-bit PPG Control Register 3 (RCR23)

Figure 9.4-4 12-bit PPG Control Register 3 (RCR23)

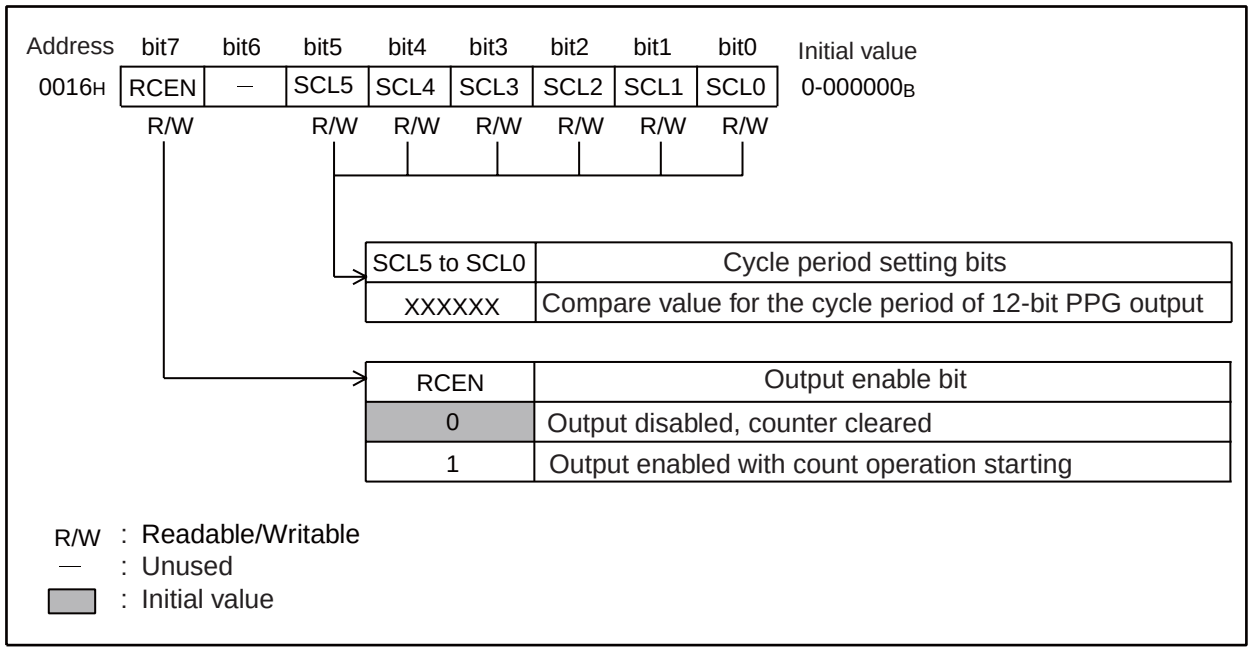


Table 9.4-3 Explanation of Functions of Each Bit in 12-bit PPG Control Register 3 (RCR23)

Bit name		Function
bit7	RCEN: Output enable bit	When this bit is "0", the P37/BZ/PPG pin functions as a general-purpose port (P37); when the bit is "1", the pin functions as a 12-bit PPG output pin (PPG). When "0" is written for this bit, the counter is cleared and its operation stops; when "1" is written, the count operation starts. Even if PPG outputs are enabled by this bit setting, buzzer outputs, if enabled, have priority.
bit6	Unused bit	• Bit value is undefined when being read. • Written value does not affect other operations.
bit5 to bit0	SCL5 to SCL0: Cycle period setting bits	These bits are used to set the number of counts corresponding to the cycle period of 12-bit PPG waveform outputs (the compare value for the cycle period), and the contents of these bits and the SCL6 to SCL11 bits of RCR24 are compared with a count by the counter. Note: Set a value that falls within the range of "000000000010_B" to "111111111111_B" (002_H to FFF_H).

Note:

Because buzzer outputs to the P37/BZ/PPG pin precede 12-bit PPG outputs to this pin, if the pin is used as the PPG pin, turn the buzzer outputs off and set the RCEN bit such that PPG outputs are enabled.

9.4.4 12-bit PPG Control Register 4 (RCR24)

The 12-bit PPG control register 4 comprises bits for setting a cycle period of 12-bit PPG waveform outputs.

■ 12-bit PPG Control Register 4 (RCR24)

Figure 9.4-5 12-bit PPG Control Register 4 (RCR24)

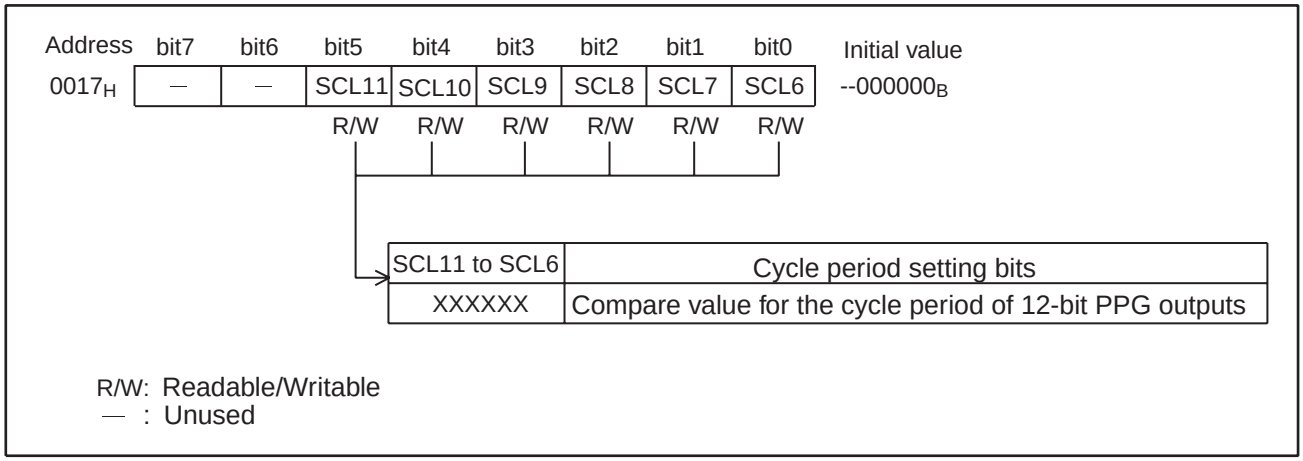


Table 9.4-4 Explanation of Functions of Each Bit in 12-bit PPG Control Register 4 (RCR24)

Bit name		Function
bit7, bit6	Unused bits	• Bit value is undefined when being read. • Written value does not affect other operations.
bit5 to bit0	SCL11 to SCL6: Cycle period setting bits	These bits are used to set the number of counts corresponding to the cycle period of 12-bit PPG waveform outputs (the compare value for cycle period), and the contents of these bits and the SCL0 to SCL5 bits of RCR23 are compared with a count by the counter. Note: Set a value that falls within the range of "000000000010_B" to "111111111111_B" (002_H to FFF_H).

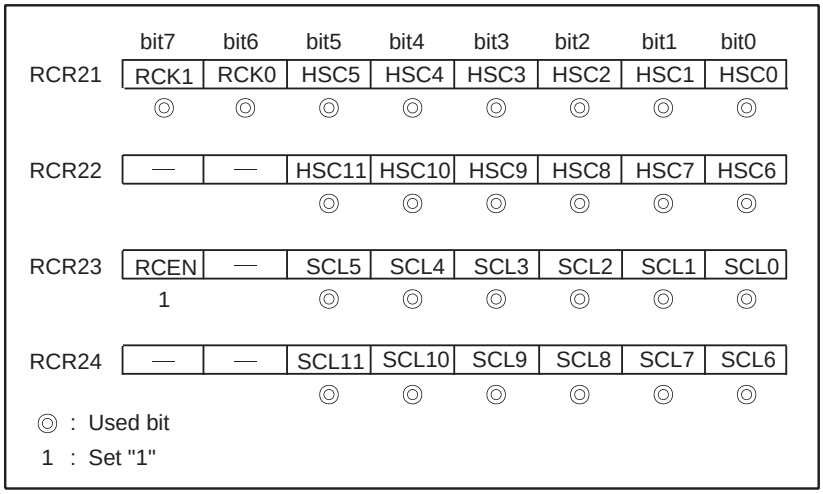
9.5 Operations of 12-bit PPG Timer Functions

The 12-bit PPG timer can be used as a 12-bit PPG because the output pulse cycle period and "H" pulse width can be set separately.

■ Example of Operations of 12-bit PPG Timer Functions

To operate the 12-bit PPG timer, the bits of the registers must be set as shown in Figure 9.5-1 .

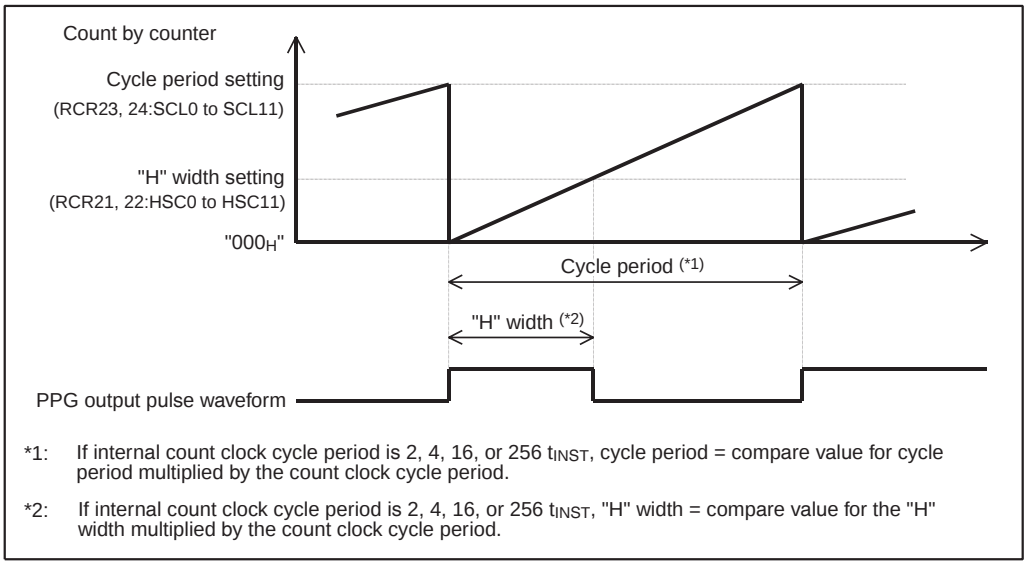
Figure 9.5-1 Setting 12-bit PPG Timer



When 12-bit PPG outputs are enabled, the 12-bit counter starts counting from" $000_{H"}$ in synchronization with the selected count clock and the PPG pin is maintained at "H" level until a count by the counter is synchronized with the compare value for the "H" width. The PPG pin is then maintained at "L" level until a count by the counter is synchronized with the compare value for the cycle period. At this time, the 12-bit counter is cleared and restarts counting from "$000_{H"}$. Because the "H" width and cycle period can be set separately, the timer can be used as a 12-bit PPG.

Figure 9.5-2 illustrates the operation of the 12-bit PPG timer.

Figure 9.5-2 Operation of 12-bit PPG Timer



*1: If internal count clock cycle period is 2, 4, 16, or 256 t_{INST}, cycle period = compare value for cycle period multiplied by the count clock cycle period.

*2: If internal count clock cycle period is 2, 4, 16, or 256 t_{INST}, "H" width = compare value for the "H" width multiplied by the count clock cycle period.

9.6 Notes on Using 12-bit PPG Timer

This section provides notes on using the 12-bit PPG timer.

■ Notes on Using 12-bit PPG Timer

- Output pin changeover

  The P37/BZ/PPG pin shares functions of a general-purpose port and a 12-bit PPG output. Because its buzzer output (BZ) function precedes the 12-bit PPG output function, if buzzer outputs are enabled, it functions as the buzzer output (BZ) pin even if PPG outputs are enabled by the RCR23 (RCEN bit). To use it as the 12-bit PPG output (PPG) pin, turn the buzzer outputs OFF.

- Limitation of "H" width setting

  Using the "H" width setting bits of the 12-bit PPG control registers 1 and 2 (RCR21:HSC5 to HSC0 and RCR22:HSC11 to HSC6), set a value that falls within the range of "000000000001_B" to "111111111111_B" (001_H to FFF_H). If "000_H" is set, "H" level outputs are delivered through the PPG pin. Furthermore, set the value of the "H" width so as to be smaller than the value given by the cycle period setting bits of 12-bit PPG control registers 3 and 4 (RCR23:SCL5 to SCL0 and RCR24:SCL11 to SCL6). If the "H" width is equal to or greater than the cycle period, "H" level outputs are always delivered through the PPG pin.

- Resolution

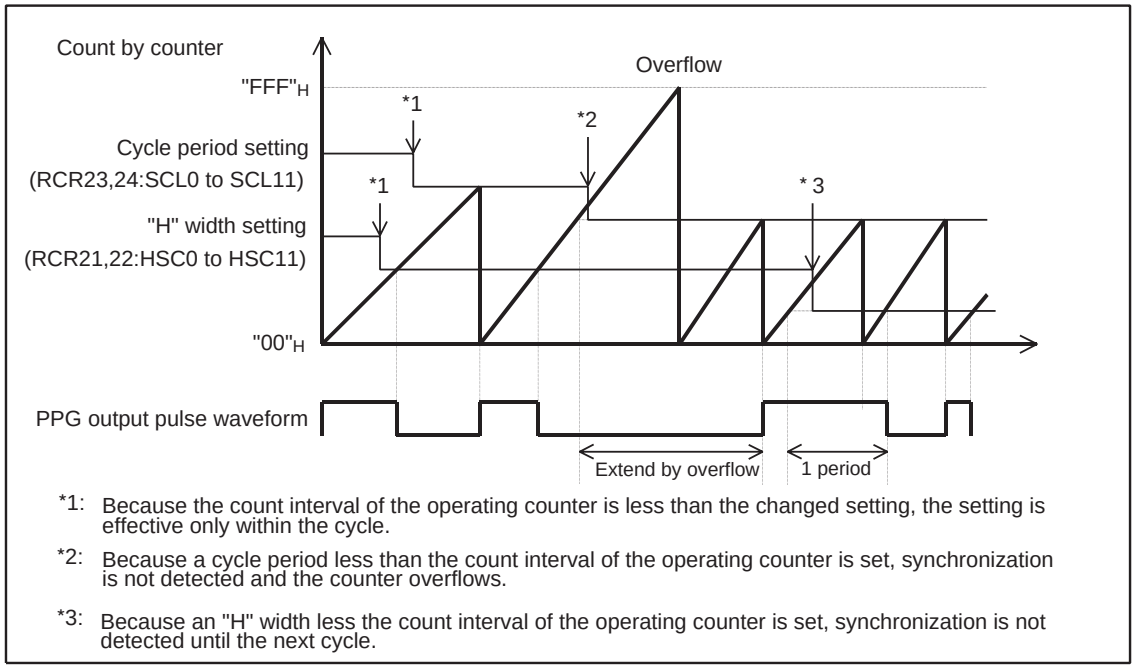
  When the cycle period is set at "111111111111_B" (FFF_H), a maximum "H" width resolution of 1/4095 is obtained. This resolution is reduced as the cycle period setting becomes smaller and limited to a minimum of 1/2 when the cycle period is set at "000000000010_B" (002_H).

- Setting change during operation

  The "H" width setting bits (RCR21:HSC5 to HSC0 and RCR22:HSC11 to HSC6) and the cycle period setting bits (RCR23:SCL5 to SCL0 and RCR24:SCL11 to SCL6) are compared with the 12-bit counter for generating a frequency of 12-bit PPG waveforms. If the set values given by these bits are changed to smaller values during the operation of the counter, a counter overflow occurs, which may extend the cycle period until synchronization with a count by the counter is detected again. Similarly, this may extend the "H" width until synchronization with a count by the counter is detected in the next cycle (cycle period).

  Figure 9.6-1 illustrates setting change during the operation of the 12-bit PPG timer.

Figure 9.6-1 Setting Change during 12-bit PPG Timer Operation



*1: Because the count interval of the operating counter is less than the changed setting, the setting is effective only within the cycle.

*2: Because a cycle period less than the count interval of the operating counter is set, synchronization is not detected and the counter overflows.

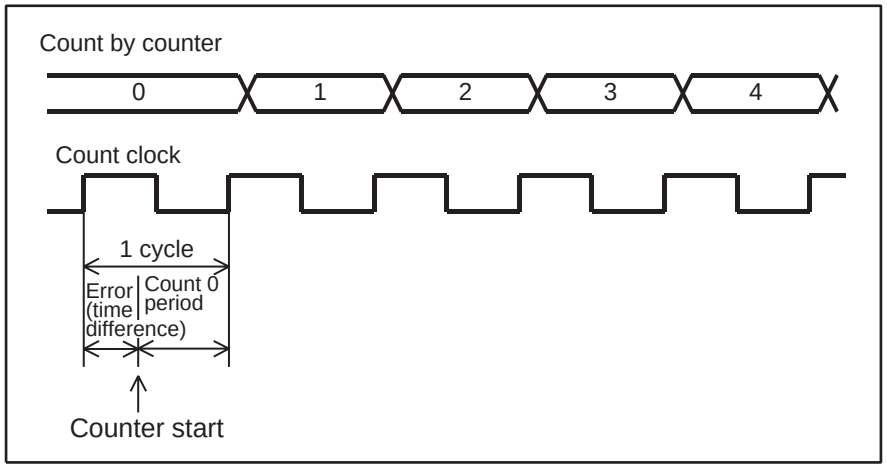
*3: Because an "H" width less the count interval of the operating counter is set, synchronization is not detected until the next cycle.

- Error

Because the counter start by program is asynchronous with the count-up start by the selected count clock, an error (a time difference) may occur until detection of synchronization of compare values for the "H" width and for the cycle period with a count by the counter. A major error may shorten the time before the above synchronization to one count clock cycle.

Figure 9.6-2 illustrates an error (a time difference) before the count operation start.

Figure 9.6-2 Error before Count Operation Start



9.7 Program Example for 12-bit PPG Timer

An example of 12-bit PPG timer programming is given below.

■ Program Example for 12-bit PPG Timer

● Processing specification

- A remote control transmission frequency with a period of about 38 μs and a duty cycle of approx. 33% is generated.
- The compare value for the PPG output pulse cycle period giving the above period of about 38 μs at the maximum gear speed with oscillation of 12.5 MHz (F_{CH}) is determined as below. The count clock is assumed to be 4 t_{INST}.

  Compare value for cycle period (RCR23:SCL5 to SCL0 and RCR24:SCL11 to SCL6) = 38 μs/ (4 × 4/ 10MHz) = 30
- The compare value for the "H" width of the PPG output pulse giving the duty cycle of approx. 33% is determined as below. At this time, the "H" width is about 3 μs.

  Compare value for the "H" width (RCR21:HSC5 to HSC0 and RCR22:HSC11 to HSC6) =33/100 × Compare value for the cycle period = 0.33 × 30 = 10

● Coding example

```
RCR21 EQU     0014H                ;   Address of 12-bit PPG control register 1
RCR22 EQU     0015H                ;   Address of 12-bit PPG control register 2
RCR23 EQU     0016H                ;   Address of 12-bit PPG control register 3
RCR24 EQU     0017H                ;   Address of 12-bit PPG control register 4
;----------------------Main program-----------------------------------------------------------------------------
        CSEG                       ;   [CODE SEGMENT]
          :
        MOV     RCR21,#01001010B   ;   Select count clock of 4 tINST and set the above compare
                                       value for "H" width.
        MOV     RCR22,#00H         ;

        MOV     RCR23,#10011110B   ;   Specify outputs enabled and counter operation start and
                                       set the above compare value for cycle period.
        MOV     RCR24,#00H         ;
          :
        ENDS
;-------------------------------------------------------------------------------------------------------------------
        END
```

CHAPTER 10

EXTERNAL INTERRUPT CIRCUIT 1 (EDGE)

This chapter describes the function and operation of external interrupt circuit 1 (edge).

10.1 Overview of External Interrupt Circuit 1

External interrupt circuit 1 detects a predetermined edge or edges of a signal input to any of three external interrupt pins and then generates and issue an interrupt request to the CPU.

■ Functions of External Interrupt Circuit 1

The external interrupt circuit 1 functions to detect an optionally selected edge or edges of a signal input to any of the external interrupt pins and then generate and issue an interrupt request to the CPU. This interrupt ensures recovery from standby mode and enables transition to a normal operating state (main clock operation mode).

- External interrupt pins: Three pins (P34/TO/INT10 to P36/INT12)
- External interrupt triggering: Input of a signal with an optionally selected edge or edges (rising and/or falling edges) to one of the above external interrupt pins triggers an external interrupt.
- Interrupt control: Interrupt request outputs are enabled or disabled according to the content of an interrupt request enable bit of external interrupt 1 control registers 1 and 2 (EIC1, EIC2).
- Interrupt flag: Detection of specified edge or edges is indicated by an external interrupt request flag bit of external interrupt 1 control registers 1 and 2 (EIC1, EIC2).
- Interrupt request: An interrupt request is generated according to the pin at which the input of the signal triggering an external interrupt is detected (IRQ0, IRQ1, IRQ2).

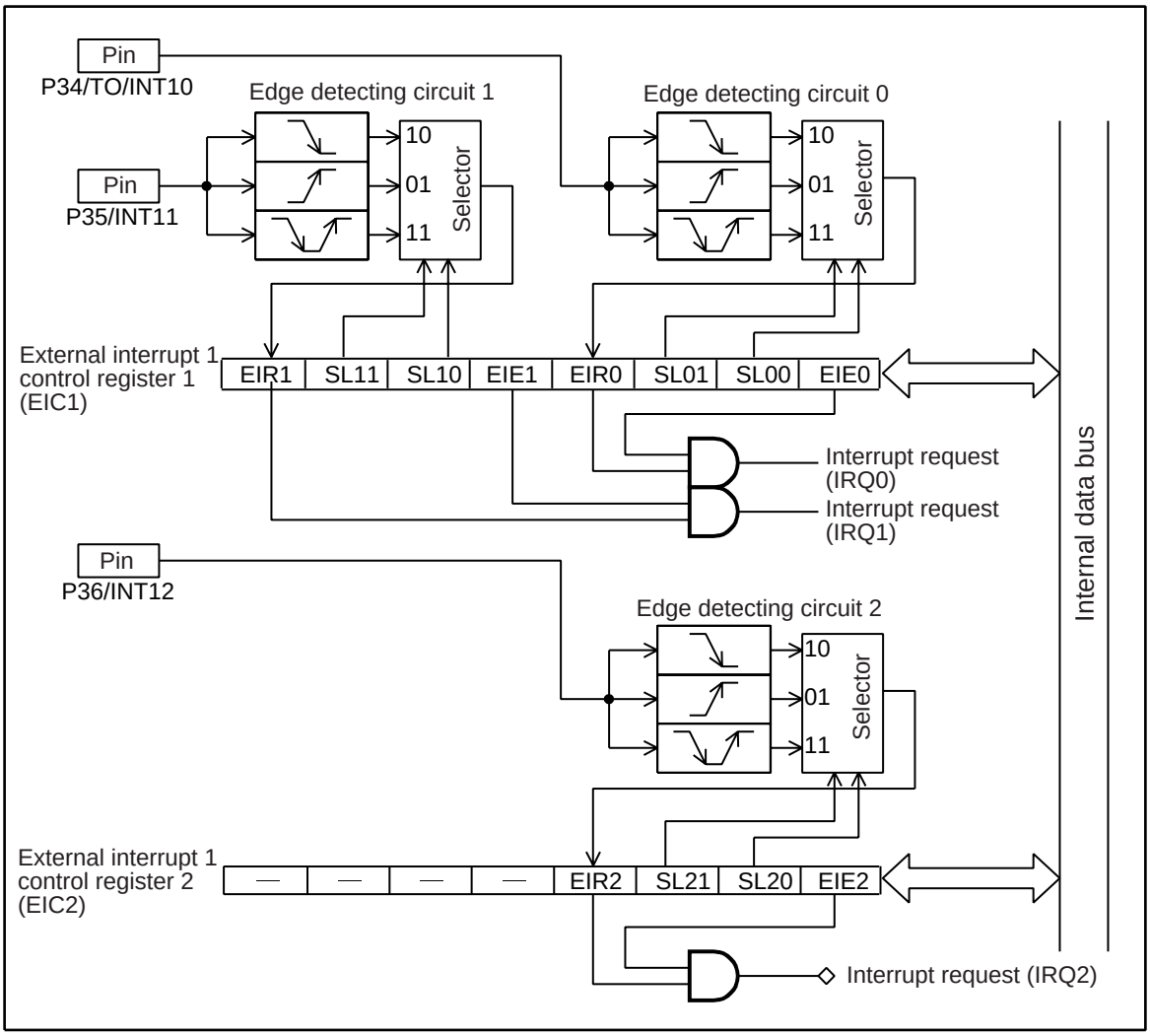
10.2 Configuration of External Interrupt Circuit 1

External interrupt circuit 1 comprises the following two blocks:

- **Edge detecting circuits (0 to 2)**
- **External interrupt control 1 registers 1, 2 (EIC1, EIC2)**

■ Block Diagram of External Interrupt Circuit 1

Figure 10.2-1 Block Diagram of External Interrupt Circuit 1 (EIC1, EIC2)



- Edge detecting circuits

  When the edge polarity of a signal input to one of the pins (INT10 to INT12) for external interrupt circuit 1 matches the selected edge polarity for the pin, stored in either the EIC1 or EIC2 registers in the appropriate bit position (SL00 to SL21), one of the external interrupt request flag bits (EIR0 to EIR2) corresponding to the pin is set to "1".

- External interrupt 1 control registers (EIC1, EIC2)

  The EIC1 and EIC2 registers comprise bits for edge selection, for enabling or disabling interrupt requests, and for confirming an interrupt request.

- Triggers that cause external interrupt circuit 1 to generate an interrupt request

  - IRQ0: When a signal with an edge or edges corresponding to the selected edge polarity is input to the INT10 pin for external interrupt circuit 1, if interrupt request outputs are enabled (EIC1:EIE0=1), external interrupt circuit 1 generates an IRQ0 interrupt request.
  - IRQ1: When a signal with an edge or edges corresponding to the selected edge polarity is input to the INT11 pin for external interrupt circuit 1, if interrupt request outputs are enabled (EIC1:EIE1=1), external interrupt circuit 1 generates an IRQ1 interrupt request.
  - IRQ2: When a signal with an edge or edges corresponding to the selected edge polarity is input to the INT12 pin for external interrupt circuit 1, if interrupt request outputs are enabled (EIC2:EIE2=1), external interrupt circuit 1 generates an IRQ2 interrupt request.

10.3 Pins of External Interrupt Circuit 1

This section describes the pins associated with external interrupt circuit 1 and illustrates a block diagram of circuitry terminating at the pins with reference to the registers and external interrupt triggering.

■ Pins Associated with External Interrupt Circuit 1

The pins associated with external interrupt circuit 1 are the P34/TO/INT10 to P36/INT12 pins.

- P34/TO/INT10 pin

  This pin functions as a general-purpose I/O dedicated port and may also serve 8/16-bit capture timer outputs (TO) and external interrupt inputs (hysteresis inputs) (INT10).

  If the timer 1 control register (TCR0) disables 8/16-bit capture timer outputs and, by the port data direction register (DDR3), the pin is set to function as an input port only. The pin can also function as an external interrupt input pin (INT10). When external interrupt 1 control register 1 (EIC1) sets edge detection to OFF, however, no external interrupt requests are generated, and when interrupt request outputs are disabled, no interrupt requests are output. The pin state can be read directly from the port data register (PDR3) at any time.

- P35/INT11 and P36/INT12 pins

  These pins function as a general-purpose I/O dedicated port (P35, P36) and may also serve external interrupt inputs (hysteresis inputs) (INT11, INT12).

  If, by the port data direction register (DDR3), these pins are set to function as an input port only, they also function as external interrupt input pins (INT11, INT12). When external interrupt 1 control registers 1 and 2 (EIC1, EIC2) set edge detection to OFF, however, no external interrupt requests are generated, and when interrupt request outputs are disabled, no interrupt requests are output. The pin state can be read directly from the port data register (PDR3) at any time.

  Table 10.3-1 lists the pins associated with external interrupt circuit 1.

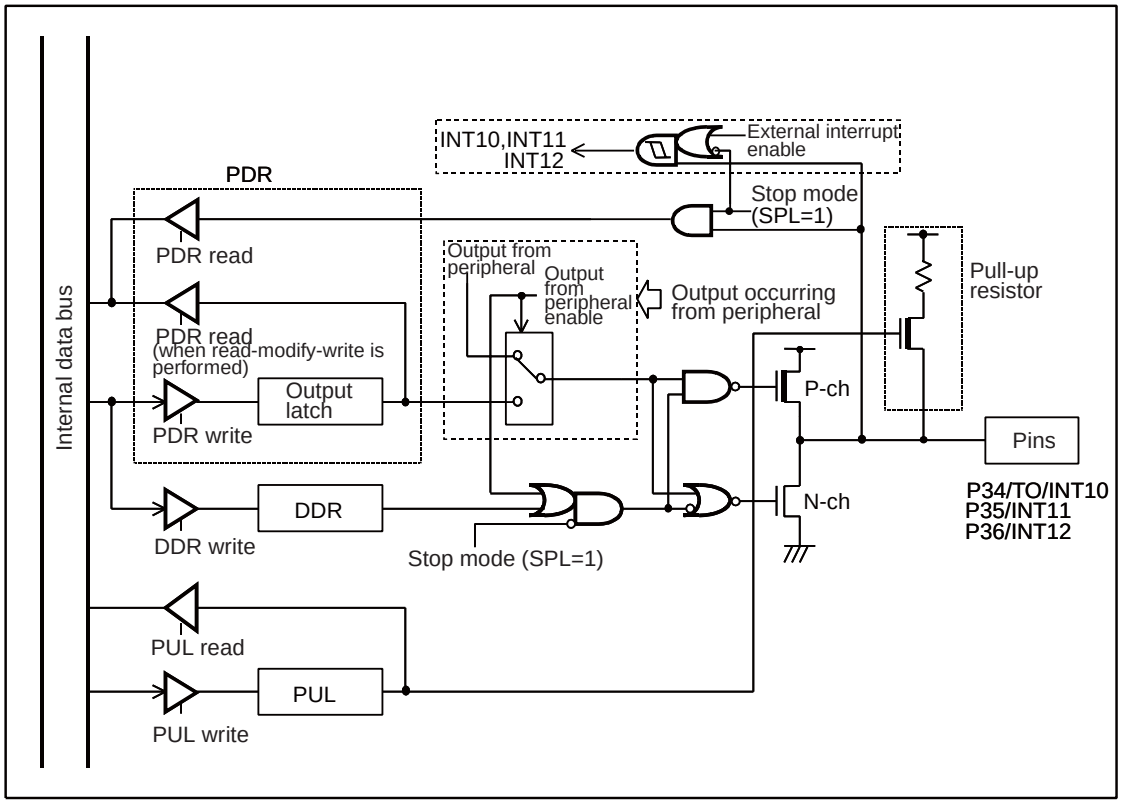
Table 10.3-1 Pins Associated with External Interrupt Circuit 1

External interrupt pin	Use for external interrupt input (Interrupt request output enabled)	Use for input port only (Interrupt request output or edge detection disabled)
P34/TO/INT10	INT10 (EIC1:EIE0=1, DDR3:bit4=0, TCR2:PEN=0)	P34(EIC1:EIE0=0 or SL01, SL00=00_B)
P35/INT11	INT11 (EIC1:EIE1=1, DDR3:bit5=0)	P35(EIC1:EIE1=0 or SL11, SL10=00_B)
P36/INT12	INT12 (EIC2:EIE2=1, DDR3:bit6=0)	P36(EIC2:EIE2=0 or SL21, SL20=00_B)

INT10 to INT12: When a signal with an edge or edges corresponding to the selected edge polarity is input to these pins, an interrupt corresponding to the pin is generated.

■ Block Diagram of Circuitry Terminating at the Pins Associated with External Interrupt Circuit 1

Figure 10.3-1 Block Diagram of Circuitry Terminating at the Pins Associated with External Interrupt Circuit 1



Note:

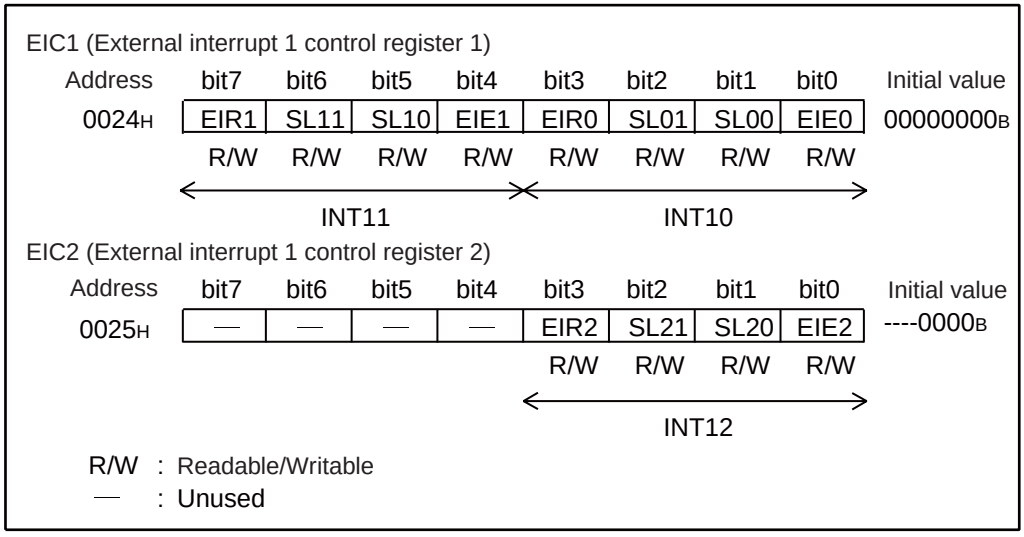
When the ON setting of the pull-up resistor is selected by the pull-up setting register, the pin state will be "H" level (pull-up state) rather than Hi-Z during stop mode (SPL = 1). During a reset, however, the pull-up is invalid and the pin remains at Hi-Z.

10.4 Registers of External Interrupt Circuit 1

This section describes the registers associated with external interrupt circuit 1.

■ Registers Associated with External Interrupt Circuit 1

Figure 10.4-1 Registers Associated with External Interrupt Circuit 1



10.4.1 External Interrupt Control Register 1 (EIC1)

External interrupt control register 1 (EIC1) comprises bits for edge polarity selection and interrupt control for the INT10 and INT11 external interrupt pins.

■ External Interrupt Control Register 1 (EIC1)

Figure 10.4-2 External Interrupt Control Register 1 (EIC1)

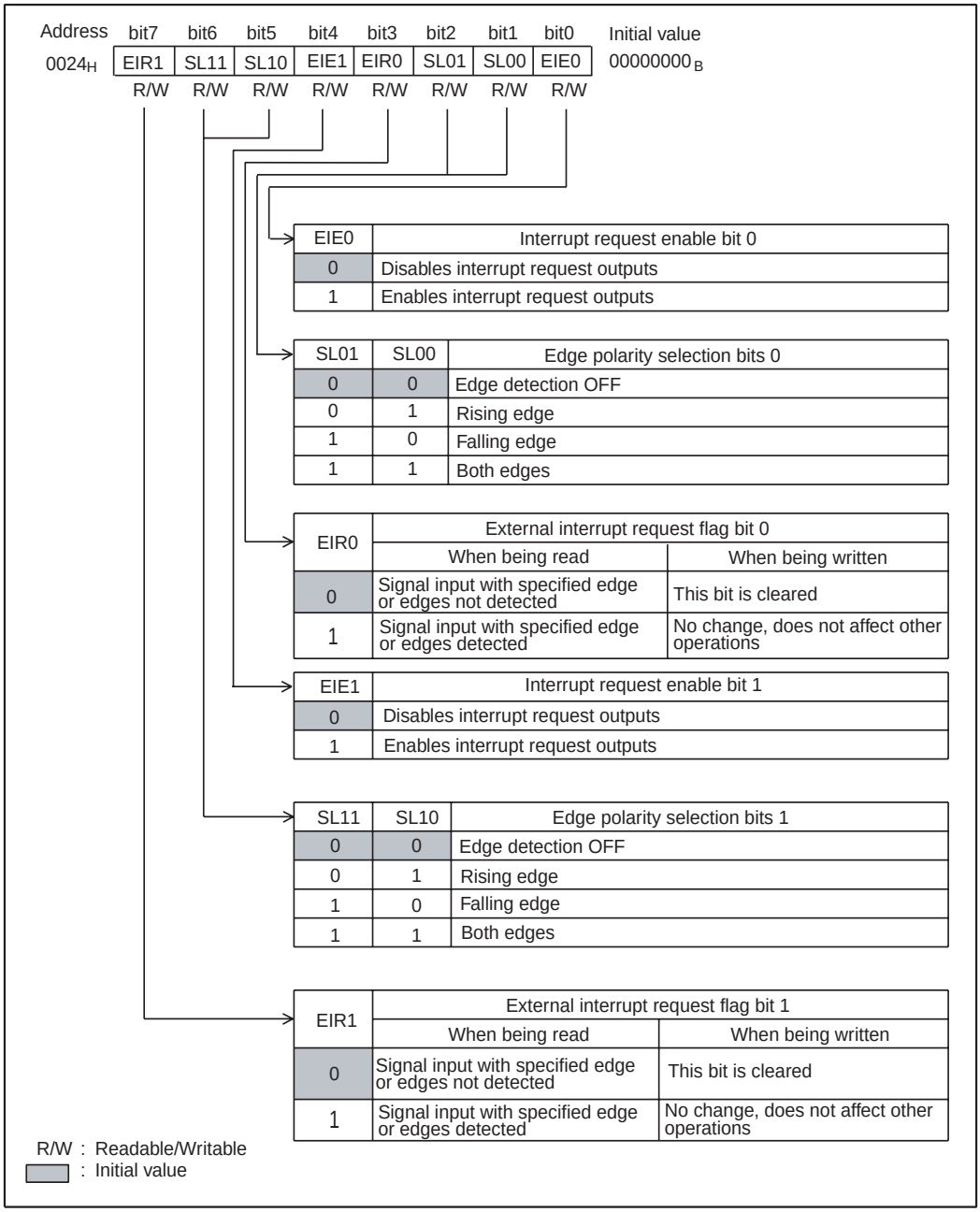


Table 10.4-1 Explanation of Functions of Each Bit in External Interrupt Control Register 1 (EIC1) (1/2)

Bit name		Function
bit7	EIR1: External interrupt request flag bit1	• When a signal with an edge or edges corresponding to edge polarity selected by edge polarity selection bits (SL11, SL10) is input to INT11 external interrupt pin, this bit is set to "1". • When this bit and interrupt request enable bit 1 (EIE1) are "1", the interrupt request is output. • Writing "0" clears this bit; writing "1" does not affect this bit (no change).
bit6, bit5	SL11, SL10: Edge polarity selection bits 1	• These bits are used to select the polarity of an edge or edges of a signal pulse that triggers an interrupt when the signal is input to INT11 external interrupt pin. • When these bits provide a value of "00_B", edge detection is not performed and interrupt requests are not generated. • These bits may specify "01_B", indicating a rising edge, "10_B", a falling edge, or "11_B", both edges to be detected. **Note:** If an edge is selected while edge detection is OFF, edge detection may be performed unconditionally. Always clear the EIR0 bit after selecting an edge.
bit4	EIE1: Interrupt request enable bit 1	This bit enables or disables interrupt request outputs to CPU. When this bit and external interrupt request flag bit 1 (EIR1) are "1", the interrupt request is output. **Notes:** • When using the external interrupt pin, write "0" for bit5 of the port data direction register (DDR3) so that the pin serves inputs only. • Regardless of the interrupt request enable bit state, the state of the external interrupt pin can be read directly from the port data register (PDR3).
bit3	EIR0: External interrupt request flag bit 0	• When a signal with an edge or edges corresponding to edge polarity selected by edge polarity selection bits (SL01, SL00) is input to INT10 external interrupt pin, this bit is set to "1". • When this bit and interrupt request enable bit 0 (EIE0) are "1", the interrupt request is output. • Writing "0" clears this bit, and writing "1" does not affect this bit.
bit2, bit1	SL01, SL00: Edge polarity selection bits 0	• These bits are used to select the polarity of an edge or edges of a signal pulse that triggers an interrupt when the signal is input to the INT10 external interrupt pin. • When these bits provide a value of "00_B", edge detection is not performed and interrupt requests are not generated. • These bits may specify "01_B", indicating a rising edge, "10_B", a falling edge, or "11_B", both edges to be detected. **Note:** If edge is selected when edge detection is OFF, edge detection may be performed unconditionally. Always clear the EIR0 bit after selecting an edge.

Table 10.4-1 Explanation of Functions of Each Bit in External Interrupt Control Register 1 (EIC1) (2/2)

Bit name		Function
bit0	EIE0: Interrupt request enable bit 0	This bit enables or disables interrupt request outputs to the CPU. When this bit and external interrupt request flag bit 0 (EIR0) are "1", the interrupt request is output. **Notes:** • When using the external interrupt pin, write "0" for bit4 of the port data direction register (DDR3) so that the pin serves inputs only. Write "0" for bit1 of the timer output control register (TCR2) for the 8/16-bit capture timer/counter to set the port input function on. • Regardless of the interrupt request enable bit state, the state of the external interrupt pin can be read directly from the port data register (PDR3).

10.4.2 External Interrupt Control Register 2 (EIC2)

As with external interrupt control register 1 (EIC1), external interrupt control register 2 (EIC2) comprises bits for edge polarity selection and interrupt control for the INT12 external interrupt pin.

■ External Interrupt Control Register 2 (EIC2)

Figure 10.4-3 External Interrupt Control Register 2 (EIC2)

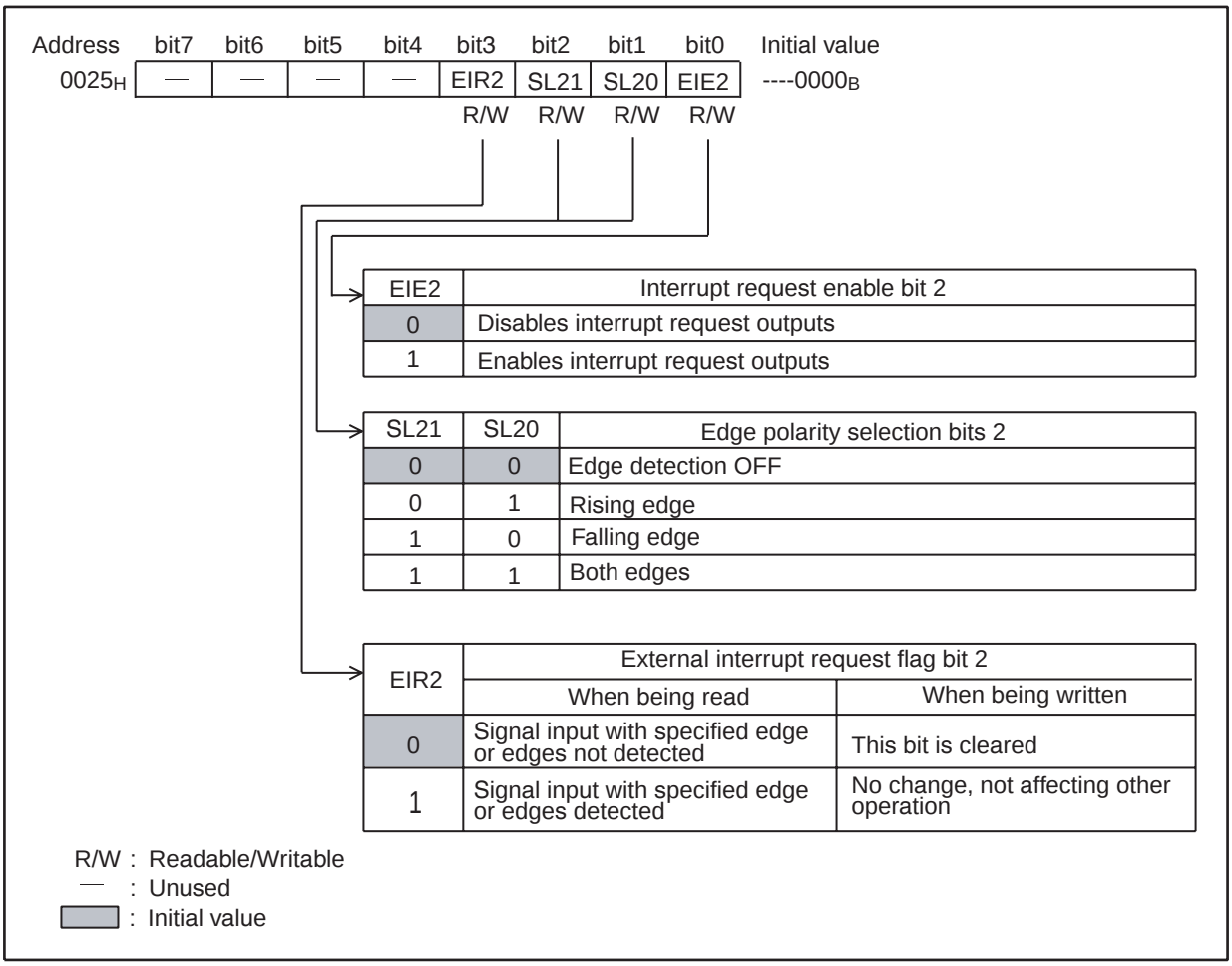


Table 10.4-2 Explanation of Functions of Each Bit in External Interrupt Control Register 2 (EIC2)

Bit name		Function
bit7 to bit4	Unused bits	• Bit value is undefined when being read. • Written value does not affect other operations.
bit3	EIR2: External interrupt request flag bit 2	• When a signal with an edge or edges corresponding to edge polarity selected by edge polarity selection bits 2 (SL21, SL20) is input to the INT12 external interrupt pin, this bit is set to "1". • When this bit and interrupt request enable bit 2 (EIE2) are "1", the interrupt request is output. • Writing "0" clears this bit, and writing "1" does not affect this bit (no change).
bit2 bit1	SL21, SL20: Edge polarity selection bits 2	• These bits are used to select the polarity of an edge or edges of a signal pulse that triggers an interrupt when the signal is input to the INT12 external interrupt pin. • When these bits provide a value of "00_B", edge detection is not performed and interrupt requests are not generated. • These bits may specify "01_B", indicating a rising edge, "10_B", a falling edge, or "11_B", both edges to be detected. **Note:** If edge is selected while edge detection is OFF, edge detection may be performed unconditionally. Always clear EIR2 bit after selecting an edge.
bit0	EIE2: Interrupt request enable bit 2	This bit enables or disables interrupt request outputs to the CPU. When this bit and external interrupt request flag bit 2 (EIR2) are "1", the interrupt request is output. **Notes:** • When using the external interrupt pin, write "0" for bit6 of the port data direction register (DDR3) so that the pin serves inputs only. • Regardless of the interrupt request enable bit state, the state of external interrupt pin can be read directly from the port data register (PDR3).

10.5 Interrupt of External Interrupt Circuit 1

The detection of a signal with the specified edge or edges, input to any of the external interrupt pins, triggers external interrupt circuit 1 to generate an interrupt request.

■ Interrupt during the Operation of External Interrupt Circuit 1

When external interrupt circuit 1 detects the specified edge or edges of external interrupt input at a pin, an external interrupt request flag bit (EIC1, EIC2:EIR0 to EIR2) corresponding to the pin is set to "1". At this time, if the interrupt request enable bit corresponding to the pin contains the value indicating the enabled state (EIC1, EIC2:EIE0 to EIE2=1), the external interrupt circuit 1 generates and then issues the appropriate interrupt request (IRQ0, IRQ1, IRQ2) to the CPU.

Write "0" for the external interrupt request flag bit within the interrupt processing routine for the interrupt request, thus clearing the interrupt request.

If external interrupts are not used for recovery from stop mode, set the edge polarity selection bits to "00_B" and the interrupt enable bits to "0".

Notes:

- When edge detection OFF is selected and set with edge polarity selection bits, the occurring input is held as is before entry to the internal edge detecting circuit. If edge is selected during the edge detection OFF state, edge detection may be performed unconditionally with the external interrupt request flag bit set to "1".
- When interrupts are set enabled (EIC1, EIC2:EIE0 to EIE2=1) after the release from the reset state, clear the appropriate external interrupt request flag bit (EIR0 to EIR2=0) at the same time.

  If the external interrupt request flag bit is "1" with the interrupt request enable bit containing a value indicating enable state, a return from the interrupt processing is not possible. Always clear the external interrupt request flag bit within the interrupt processing routine.
- For edge selection during the edge detection OFF state, specify an edge or edges when interrupt request outputs are disabled and then clear the external interrupt request flag bit.

  Regardless of the value of the appropriate interrupt request enable bit (EIE0 to EIE2), the external interrupt request flag bit is set to "1" whenever edge polarity matching is detected.

  Only external interrupt circuits 1 and 2 can execute a release from stop mode by an interrupt.

  With the external interrupt request flag bit being set to "1", when the interrupt request enable bit setting changes from disable to enable ($0 \rightarrow 1$), an interrupt request is generated immediately.

■ Register Associated with Interrupt Generation by External Interrupt Circuit 1 and Vector Table

Table 10.5-1 Register Associated with Interrupt Generation by External Interrupt Circuit 1 and Vector Table

Interrupt designation	Interrupt level setting register			Vector table address	
	Register	Bit for setting level		Upper	Lower
IRQ0	ILR1 ($007B_H$)	L01 (bit1)	L00 (bit0)	$FFFA_H$	$FFFB_H$
IRQ1		L11 (bit3)	L10 (bit2)	$FFF8_H$	$FFF9_H$
IRQ2		L21 (bit5)	L20 (bit4)	$FFF6_H$	$FFF7_H$

For interrupt operation, see Section "3.4.2 Steps in the Interrupt Operation ".

■ Exercise Caution when Changing Edge Polarity Selection

When changing edge polarity for INT10 to INT12, always write "0" for the appropriate EIR bit to prevent unintended interrupt generation.

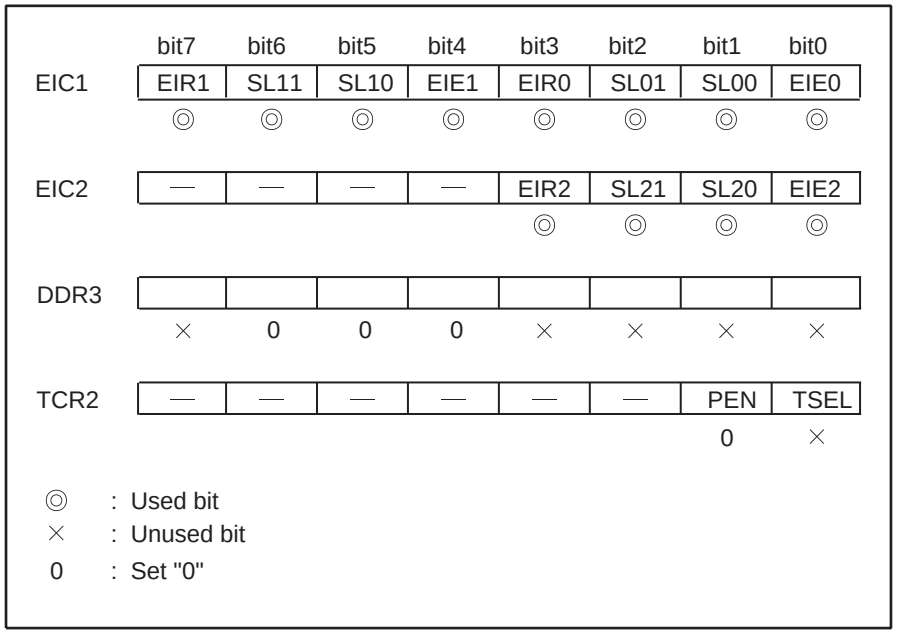
10.6 Operations of External Interrupt Circuit 1

The external interrupt circuit 1 can detect a specified edge or edges of a signal input to any of the external interrupt pins.

■ Operation of External Interrupt Circuit 1

To operate external interrupt circuit 1, the bits of the registers must be set as shown in Figure 10.6-1 .

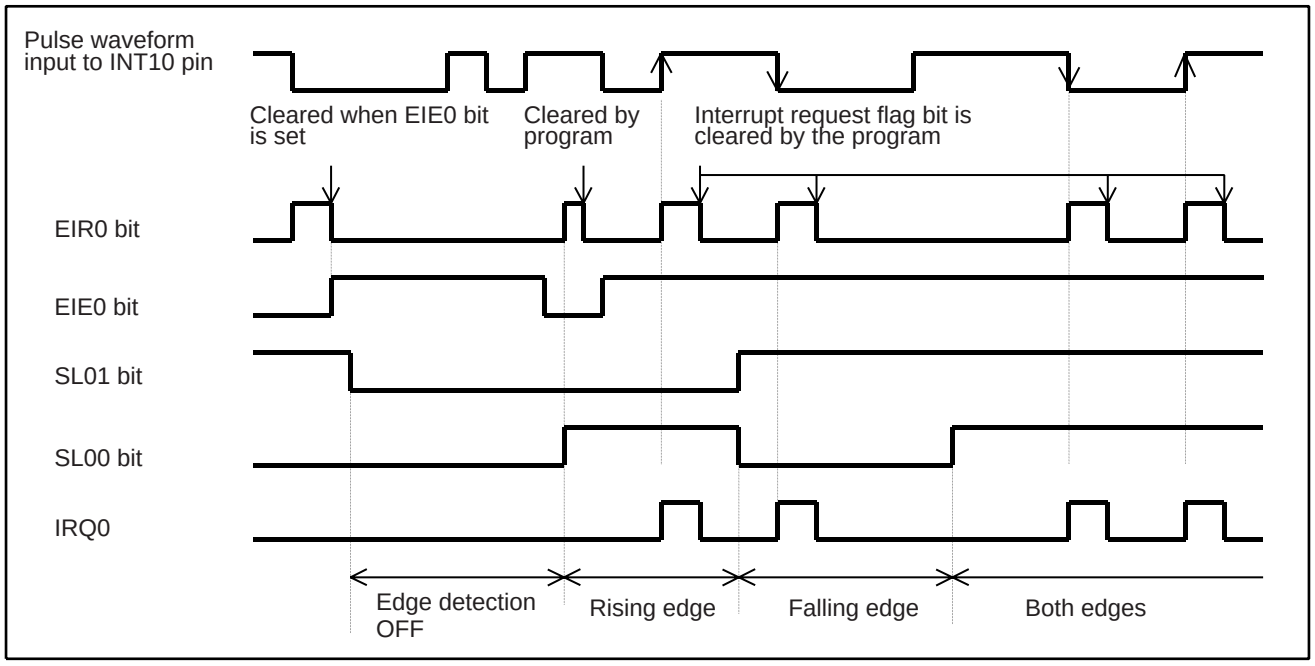
Figure 10.6-1 Setting External Interrupt Circuit 1



When the polarity of an edge or edges of a signal input from one of the external interrupt pins 1 (INT10 to INT12) matches the selected edge polarity for the pin stored in the appropriate external interrupt control register (EIC1, EIC2:SL00 to SL21), one of the external interrupt request flag bits (EIC1, EIC2:EIR0 to EIR2) corresponding to the pin is set to "1".

Figure 10.6-2 shows the operation when an external interrupt is input to the INT10 pin.

Figure 10.6-2 Operation of External Interrupt 1 (INT10)



Note:

Even when the pin is used as an external interrupt input pin, the pin state can be read directly from the port data register (PDR3).

10.7 Program Example for External Interrupt Circuit 1

An example of programming external interrupt circuit 1 is given below.

■ External Interrupt Circuit 1 Programming Example

● Processing specification

External interrupt circuit 1 detects the rising edge of a pulse input to the INT10 pin and generates an interrupt.

● Coding example

```
DDR3    EQU     000DH           ;  Address of port data direction register (DDR)
EIC1    EQU     0024H           ;  External interrupt control register 1
TCR2    EQU     0020H           ;  Address of 8/16-bit capture timer output control
                                   register
ILR1    EQU     007BH           ;  Setting of interrupt level setting register 1

EIR0    EQU     EIC1:3          ;  Definition of external interrupt request flag bit
SL01    EQU     EIC1:2          ;  Definition of edge polarity selection bits
SL00    EQU     EIC1:1          ;  Definition of edge polarity selection bits
EIE0    EQU     EIC1:0          ;  Definition of interrupt request enable bit
INT_V   DSEG    ABS             ;  [DATA SEGMENT]
        ORG     0FFFAH
IRQ1    DW      WARI            ;  Interrupt vector (INT1) setting
INT_V   ENDS
;-----------------------Main program-----------------------------------------------------------------------
        CSEG                    ;  [CODE SEGMENT]
                                ;  Stack pointer (SP) is assumed to have been initialized.
         :                      ;
        CLRI                    ;  Disable interrupts.
        CLRB    EIR1            ;  Clear external interrupt request flag.
        MOV     TCR2,#00000000B ;  Set pin P34/TO/INT10 to serve port inputs.
        MOV     DDR3,#00000000B ;  Set P34 to serve inputs only.
        MOV     ILR1,#11111110B ;  Set interrupt level at 2.
        CLRB    SL01            ;  Select rising edge.
        SETB    SL00            ;
        CLRB    EIR0            ;  Clear external interrupt request flag.
        SETB    EIE0            ;  Enable interrupt request outputs.
        SETI                    ;  Enable interrupts.
         :
;-----------------------Interrupt processing routing-------------------------------------------------------
WARI    CLRB    EIE0            ;  Clear external interrupt request flag (INT0).
        PUSHW   A
        XCHW    A,T
```

```
                PUSHW   A
                 :
                User processing
                 :
                POPW    A
                XCHW    A,T
                POPW    A
                RETI
                ENDS
;---------------------------------------------------------------------------------------------------------------
                END
```

CHAPTER 11

EXTERNAL INTERRUPT CIRCUIT 2 (LEVEL)

This chapter describes the function and operation of an external interrupt circuit 2 (level).

11.1 Overview of External Interrupt Circuit 2

External interrupt circuit 2 detects the predetermined level of a signal input to any of the eight external interrupt pins and generates and issues an interrupt request to the CPU.

■ Functions of External Interrupt Circuit 2 (Level Detection)

External interrupt circuit 2 functions to detect an "L" level signal input to any of the external interrupt pins and generate and issue an interrupt request to the CPU, thereby enabling recovery from standby mode and a transition to normal operating state (main clock operation mode).

- External interrupt pins: Eight pins (P00/$\overline{\text{INT20}}$/AN4 to P03/$\overline{\text{INT23}}$/AN7, P04/$\overline{\text{INT24}}$ to P07/$\overline{\text{INT27}}$)
- External interrupt triggering: Input of an "L" level signal to one of the above external interrupt pins triggers an external interrupt.
- Interrupt control: An external interrupt 2 control register (EIE2) enables or disables external interrupt inputs.
- Interrupt flag: Detection of the "L" level is indicated by an external interrupt request flag bit of the external interrupt 2 flag register (EIF2).
- Interrupt request: An interrupt request is generated if the state of one of the above external interrupt pins is "L" (IRQA).

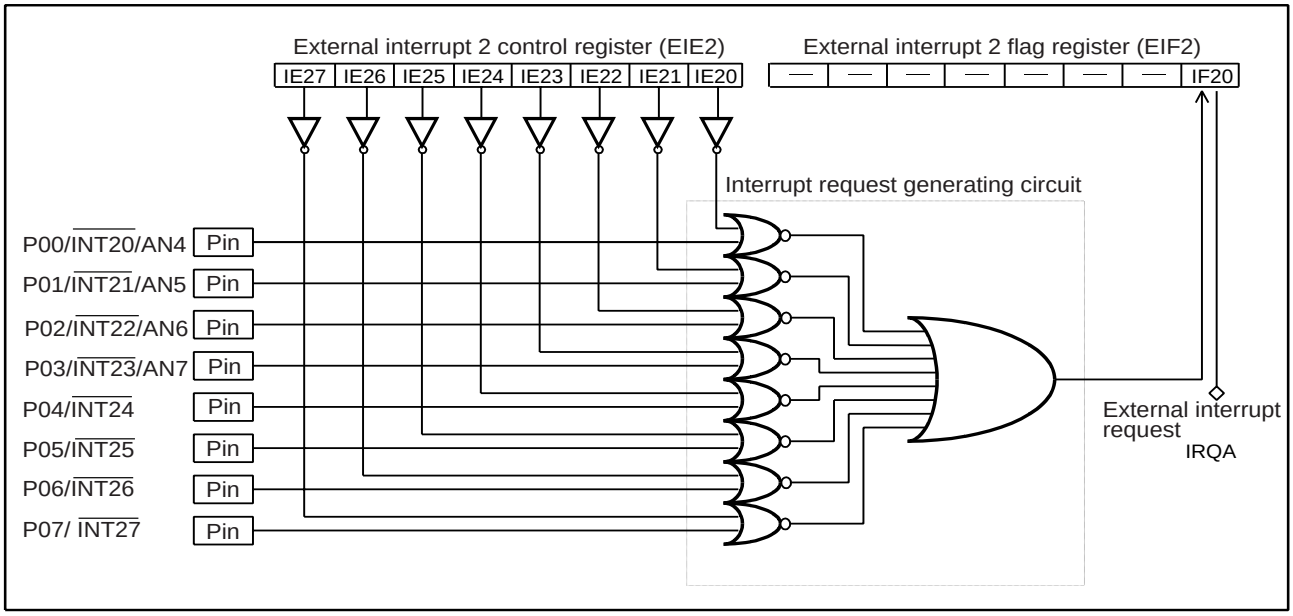
11.2 Configuration of External Interrupt Circuit 2

The external interrupt circuit 2 comprises the following three blocks:

- **Interrupt request generating circuit**
- **External interrupt 2 control register (EIE2)**
- **External interrupt 2 flag register (EIF2)**

■ Block Diagram of External Interrupt Circuit 2

Figure 11.2-1 Block Diagram of External Interrupt Circuit 2



● Interrupt request generating circuit

The interrupt request generating circuit generates an interrupt request signal in accordance with the signal input to one of the external interrupt pins ($\overline{INT20}$ to $\overline{INT27}$) and the contents of an external interrupt input enable bit.

● External interrupt 2 control register (EIE2)

The external interrupt input enable bits (IE20 to IE27) enable or disable "L" level input from the external interrupt pins, with each bit corresponding to a pin.

● External interrupt 2 flag register (EIF2)

The external interrupt request flag bit (IF20) is used to hold or clear an interrupt request signal.

● Trigger causing external interrupt circuit 2 to generate an interrupt

- IRQA: When an "L" level signal is input to any of the external interrupt pins $\overline{INT20}$ to $\overline{INT27}$ and the external interrupt input enable bit corresponding to the pin is "1", external interrupt circuit 2 generates an interrupt request.

11.3 Pins of External Interrupt Circuit 2

This section describes the pins associated with external interrupt circuit 2 and illustrates a block diagram of circuitry terminating at the pins with reference to the registers and interrupt triggering.

■ Pins Associated with External Interrupt Circuit 2

The pins associated with external interrupt circuit 2 are eight external interrupt pins.

- P00/$\overline{\text{INT20}}$/AN4 to P03/$\overline{\text{INT23}}$/AN7

  These external interrupt pins function as external interrupt input pins (hysteresis input) and as the pins of the general-purpose I/O port and analog inputs.

  The P00/$\overline{\text{INT20}}$/AN4 to P03/$\overline{\text{INT23}}$/AN7 pins function as external interrupt input pins ($\overline{\text{INT20}}$ to $\overline{\text{INT23}}$) if set to function as an input port by the corresponding bits of the port 0 data direction register (DDR0), if set to be enabled for external interrupt inputs (ADEN=0) by the corresponding bits of the A/D enable register (ADEN), and if external interrupt inputs are enabled (EIE2:IE20 to IE27=1) by the external interrupt 2 control register (EIE2). When set to function as an input port by the DDR0 register, the state of these pins can be read from the port 0 data register (PDR0) at any time.

- P04/$\overline{\text{INT24}}$ to P07/$\overline{\text{INT27}}$

  These external interrupt pins function as external interrupt input pins (hysteresis input) and also serving as the pins of the general-purpose I/O port.

  The P04/$\overline{\text{INT24}}$ to P07/$\overline{\text{INT27}}$ pins function as external interrupt input pins ($\overline{\text{INT24}}$ to $\overline{\text{INT27}}$) if set to function as an input port by the corresponding bits of the port 0 data direction register (DDR0) and if external interrupt inputs are enabled by the external interrupt 2 control register (EIE2). When set to function as an input port, the state of these pins can be read from the port 0 data register (PDR0) at any time.

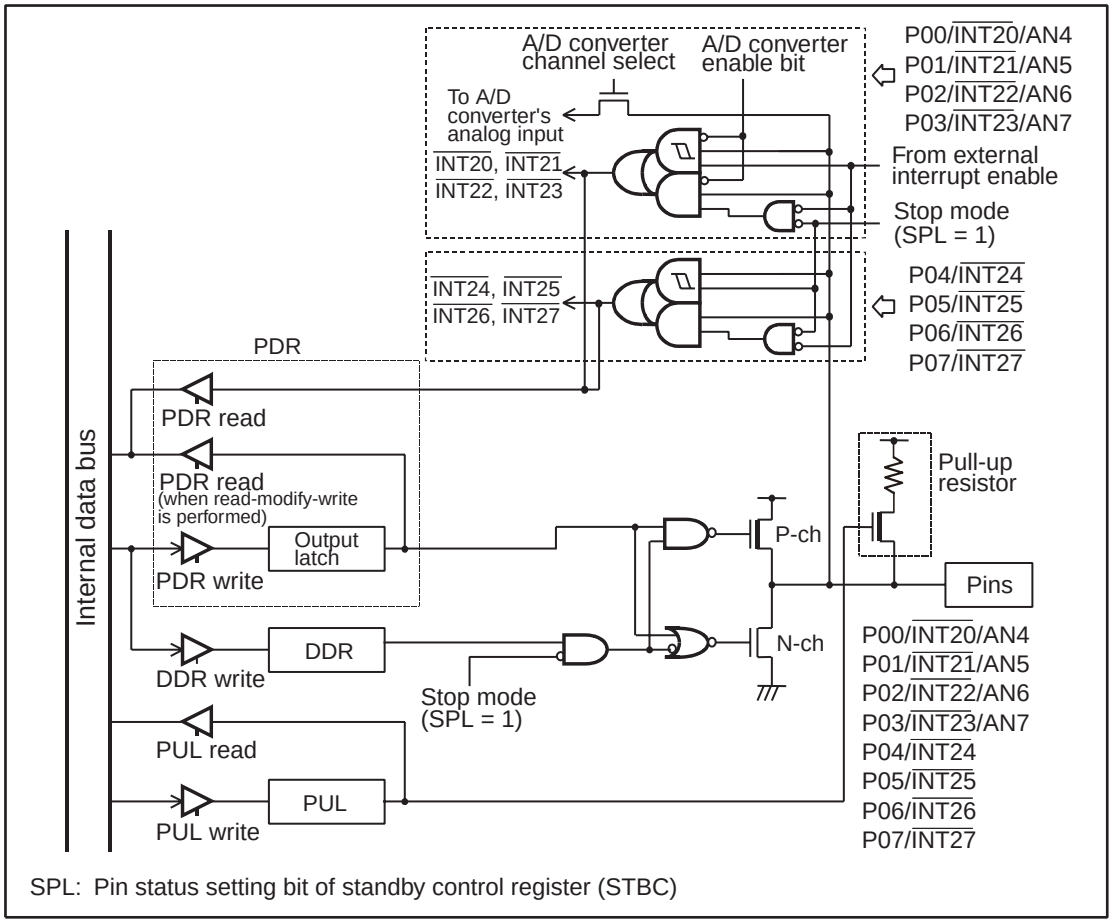
  Table 11.3-1 lists the pins associated with external interrupt circuit 2.

Table 11.3-1 Pins Associated with External Interrupt Circuit 2

External interrupt pin	Use for external interrupt input (Interrupt input enabled)	Use for general-purpose I/O port (Interrupt input disabled)
P00/$\overline{\text{INT20}}$/AN4	$\overline{\text{INT20}}$ (EIE2:IE20=1,DDR0:bit0=0,ADEN:ADE4=0)	P00 (EIE2:IE20=0)
P01/$\overline{\text{INT21}}$/AN5	$\overline{\text{INT21}}$ (EIE2:IE21=1,DDR0:bit1=0,ADEN:ADE5=0)	P01 (EIE2:IE21=0)
P02/$\overline{\text{INT22}}$/AN6	$\overline{\text{INT22}}$ (EIE2:IE22=1,DDR0:bit2=0,ADEN:ADE6=0)	P02 (EIE2:IE22=0)
P03/$\overline{\text{INT23}}$/AN7	$\overline{\text{INT23}}$ (EIE2:IE23=1,DDR0:bit3=0,ADEN:ADE7=0)	P03 (EIE2:IE23=0)
P04/$\overline{\text{INT24}}$	$\overline{\text{INT24}}$ (EIE2:IE24=1,DDR0:bit4=0)	P04 (EIE2:IE24=0)
P05/$\overline{\text{INT25}}$	$\overline{\text{INT25}}$ (EIE2:IE25=1,DDR0:bit5=0)	P05 (EIE2:IE25=0)
P06/$\overline{\text{INT26}}$	$\overline{\text{INT26}}$ (EIE2:IE26=1,DDR0:bit6=0)	P06 (EIE2:IE26=0)
P07/$\overline{\text{INT27}}$	$\overline{\text{INT27}}$ (EIE2:IE27=1,DDR0:bit7=0)	P07 (EIE2:IE27=0)

■ Block Diagram of Circuitry Terminating at the Pins Associated with External Interrupt Circuit 2

Figure 11.3-1 Block Diagram of Circuitry Terminating at the Pins Associated with External Interrupt Circuit 2



Note:

When the ON setting of the pull-up resistor is selected by the pull-up setting register, the pin state will be "H" level (pull-up state) rather than Hi-Z during stop mode (SPL = 1). During a reset, however, the pull-up is invalid and the pin remains at Hi-Z.

■ Association between the Interrupt Enable Bits for External Interrupt Circuit 2 and the External Interrupt Pins

The interrupt enable bits are associated with the external interrupt pins as listed in Table 11.3-2 .

Table 11.3-2 Correspondence between the External Interrupt Enable Bits and the External Interrupt Pins

Register	Bit name		External interrupt pin
EIE2	bit7	IE27	$\overline{\text{INT27}}$
	bit6	IE26	$\overline{\text{INT26}}$
	bit5	IE25	$\overline{\text{INT25}}$
	bit4	IE24	$\overline{\text{INT24}}$
	bit3	IE23	$\overline{\text{INT23}}$
	bit2	IE22	$\overline{\text{INT22}}$
	bit1	IE21	$\overline{\text{INT21}}$
	bit0	IE20	$\overline{\text{INT20}}$

11.4 Registers of External Interrupt Circuit 2

The external interrupt 2 control register (EIE2) is used to enable or disable the external interrupt pins.

■ Registers Associated with External Interrupt Circuit 2

Figure 11.4-1 Registers Associated with External Interrupt Circuit 2

EIE2 (External interrupt 2 control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0036H	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
0037H	—	—	—	—	—	—	—	IF20	-------0B
								R/W	

R/W : Readable/Writable
— : Unused

11.4.1 External Interrupt 2 Control Register (EIE2)

The external interrupt circuit 2 control register (EIE2) enables or disables the interrupt inputs to the external interrupt pins $\overline{\text{INT20}}$ to $\overline{\text{INT27}}$.

■ External Interrupt Circuit 2 Control Register (EIE2)

Figure 11.4-2 External Interrupt Circuit 2 Control Register (EIE2)

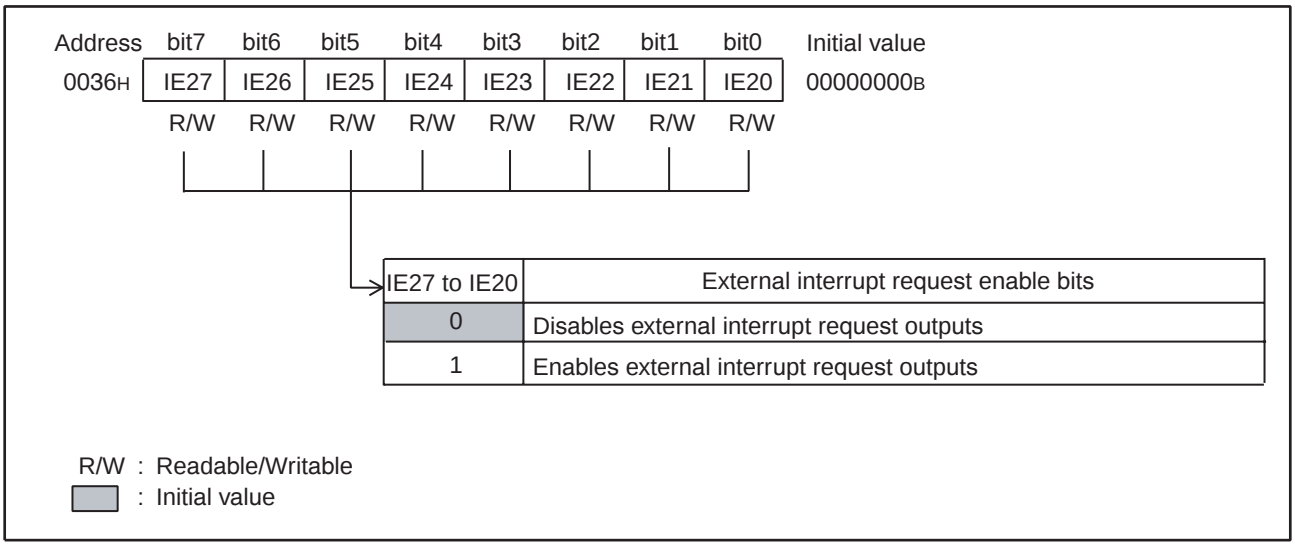


Table 11.4-1 Correspondence between the Bits of the External Interrupt 2 Control Register (EIE2) and the External Interrupt Pins

Register	Bit name		External interrupt pin
EIE2	bit7	IE27	$\overline{\text{INT27}}$
	bit6	IE26	$\overline{\text{INT26}}$
	bit5	IE25	$\overline{\text{INT25}}$
	bit4	IE24	$\overline{\text{INT24}}$
	bit3	IE23	$\overline{\text{INT23}}$
	bit2	IE22	$\overline{\text{INT22}}$
	bit1	IE21	$\overline{\text{INT21}}$
	bit0	IE20	$\overline{\text{INT20}}$

Table 11.4-2 Functions of the Bits of the External Interrupt 2 Control Register (EIE2)

Bit name		Function
bit7 to bit0	IE27 to IE20: External interrupt input enable bits	• These bits enable or disable the interrupt inputs to external interrupt pins $\overline{INT20}$ to $\overline{INT27}$. • When one of these bits is set to "1", the corresponding external interrupt pin functions as an external interrupt input pin and accepts external interrupt inputs. • When the bit is set to "0", the corresponding pin functions as a general-purpose port, but does not accept external interrupt inputs. **Notes:** • When using a pin as an external interrupt pin, write "0" in the port 0 data direction register (DDR0) in the bit corresponding to the pin so that the pin serves inputs only. • Regardless of the external interrupt input enable bit state, the state of the external interrupt pin can be read directly from the port 0 data register (PDR0). For pins $\overline{INT20}$ to $\overline{INT23}$, furthermore, write "0" into A/D enable register (ADEN) in the bits corresponding to the pins to use the pins for external interrupt inputs.

11.4.2 External Interrupt 2 Flag Register (EIF2)

The external interrupt 2 flag register (EIF2) is used to hold the interrupt state by flagging an interrupt request flag when a level interrupt is detected and then clearing the flag.

■ External Interrupt 2 Flag Register (EIF2)

Figure 11.4-3 External Interrupt 2 Flag Register (EIF2)

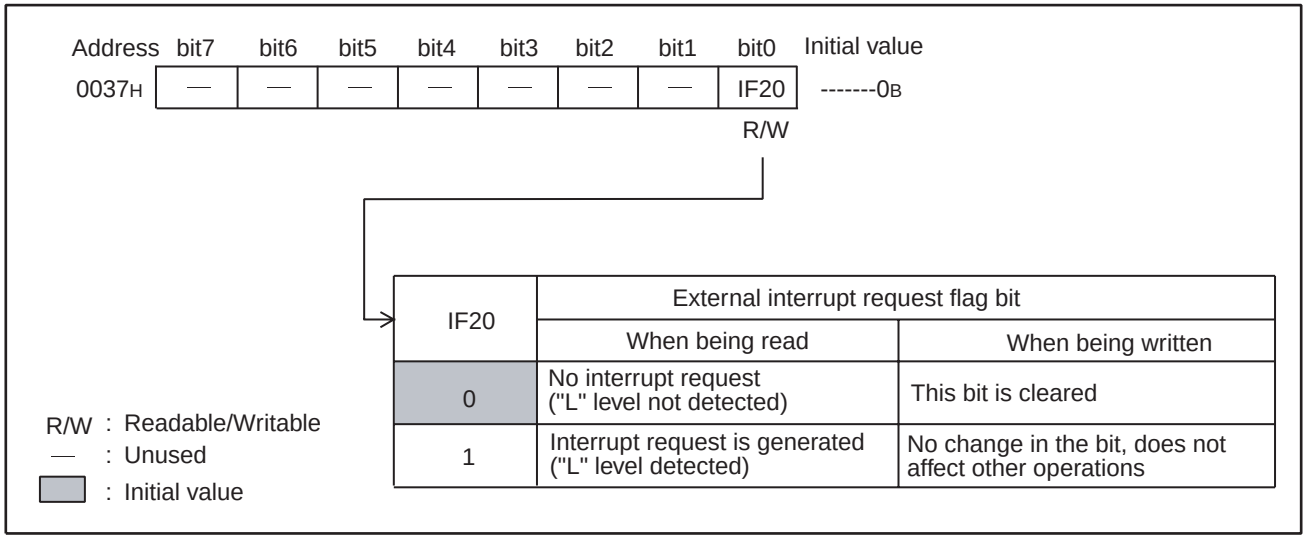


Table 11.4-3 Explanation of Functions of Each Bit in External Interrupt 2 Flag Register (EIF2)

Bit name		Function
bit7 to bit1	Unused bits	• Bit value is undefined when being read. • The written value does not affect other operations.
bit0	IF20: External interrupt request flag bit	• When an "L" level signal is input to one of the external interrupt pins($\overline{INT20}$ to $\overline{INT27}$) for which external interrupt inputs are enabled, this bit is set to "1". • Writing "0" clears this bit, and writing "1" does not change this bit state and does not affect other operations. **Note:** The external interrupt enable bits of the external interrupt 2 control register (EIE2:IE20 to IE27) may disable external interrupt inputs. Interrupt requests continue to be generated and issued to the CPU until the IF20 bit is cleared to "0".

11.5 Interrupt of External Interrupt Circuit 2

An "L" level input signal input to one of the external interrupt pins triggers external interrupt circuit 2 to generate an interrupt.

■ Interrupt during the Operation of External Interrupt Circuit 2

When an "L" level signal is input to one of the external interrupt pins for which interrupt inputs are enabled, the external interrupt request flag bit (EIF2:IF20) is set to "1" and external interrupt circuit 2 generates and issues an interrupt request (IRQA) to the CPU. Write "0" for the IF20 bit within the interrupt processing routine, thus clearing the interrupt request.

When the external interrupt request flag bit (IF20) is set to "1", external interrupt circuit 2 generates the interrupt request, even if external interrupt inputs to the pin are set to disabled by the bit corresponding to the pin among the interrupt enable bits (IE20 to IE27) of the external interrupt 2 control register (EIE2), until the IF20 bit is cleared. Therefore, the IF20 bit must always be cleared.

If the "L" level input to the external interrupt pin continues as it is, even if the IF20 bit is cleared with external interrupt inputs to the pin remaining enabled, the IF20 bit is immediately set to "1" again. Disable external interrupt inputs to the pin or remove the cause of the external interrupt as required.

Notes:

- When enabling interrupts to the CPU following a release from the reset state, clear the IF20 bit in advance.
- "L" level inputs to external interrupt pins ($\overline{INT20}$ to $\overline{INT27}$) trigger external interrupt circuit 2 to generate the same interrupt request (IRQA). Thus, when an external interrupt input is detected, it is necessary to identify the pin at which the input occurs by reading the port 0 data register (PDR0) before the input changes to "H" level.

  Only external interrupt circuits 1 and 2 can execute a release from the stop mode by an interrupt.

■ Register Associated with Interrupt Generation by External Interrupt Circuit 2 and Vector Table

Table 11.5-1 Register Associated with Interrupt Generation by External Interrupt Circuit 2 and Vector Table

Interrupt designation	Interrupt level setting register			Vector table address	
	Register	Bit for setting level		Upper	Lower
IRQA	ILR3 ($007D_H$)	LA1 (bit5)	LA0 (bit4)	$FFE6_H$	$FFE7_H$

For interrupt operation, see Section "3.4.2 Steps in the Interrupt Operation ".

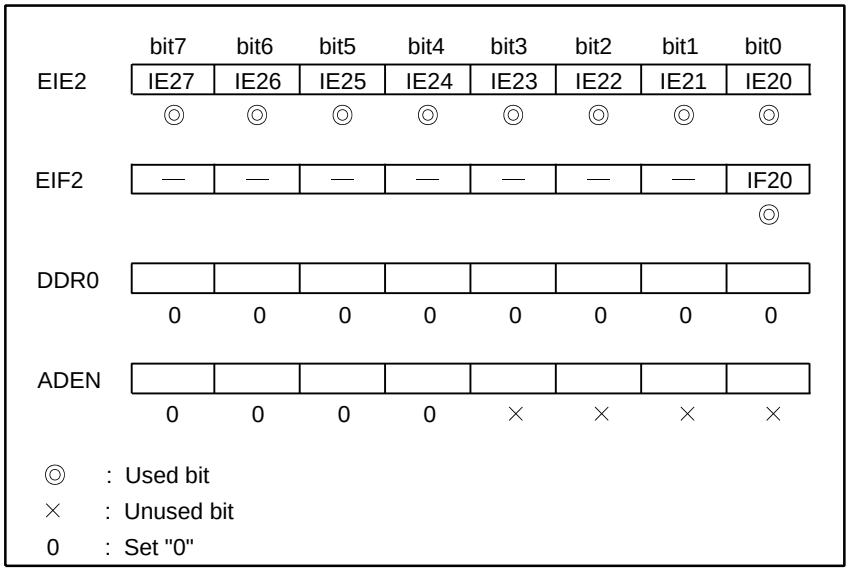
11.6 Operations of External Interrupt Circuit 2

External interrupt circuit 2 detects "L" level at any of the external interrupt pins, then generates and issues an interrupt request to the CPU.

■ Operation of External Interrupt Circuit 2

To operate the external interrupt circuit 2, the bits of the registers must be set as shown in Figure 11.6-1 .

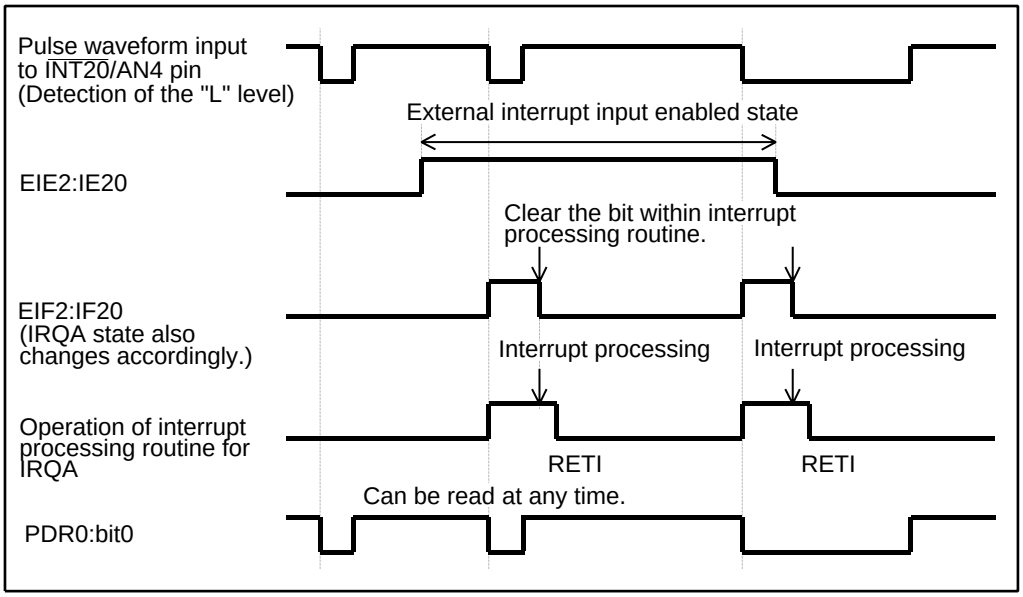
Figure 11.6-1 Setting External Interrupt Circuit 2



When an "L" level signal is input to an external interrupt pin among the pins $\overline{INT20}$ to $\overline{INT27}$ with external interrupt inputs being enabled by one of the IE20 to IE27 bits corresponding to the pin, external interrupt circuit 2 generates and issues an IRQA interrupt request to the CPU.

Figure 11.6-2 shows the operation of external interrupt circuit 2 (when the $\overline{INT20}$/AN4 pin is used).

Figure 11.6-2 Operation of External Interrupt 2 ($\overline{\text{INT20}}$)



Note:

Even when the pin is used as an external interrupt input pin, the pin state can be read directly from the port 0 data register (PDR0).

11.7 Program Example for External Interrupt Circuit 2

An example of programming external interrupt circuit 2 is given below.

■ Program Example for External Interrupt Circuit 2

● Processing specification

The external interrupt circuit 2 detects an "L" level signal input to the P00/$\overline{\text{INT20}}$/AN4 pin and generates an interrupt.

● Coding example

```
DDR0    EQU     0001H           ;  Address of the port data direction register
ADEN    EQU     0034H           ;  Address of the A/D enable register
EIE2    EQU     0036H           ;  Address of the external interrupt 2 control register
EIF2    EQU     0037H           ;  Address of the external interrupt 2 flag register

IF20    EQU     EIF2:0          ;  Definition of the external interrupt request flag bit

ILR3    EQU     007DH           ;  Address of the interrupt level setting register

INT_V   DSEG    ABS             ;  [DATA SEGMENT]
        ORG     0FFE6H
IRQA    DW      WARI            ;  Interrupt vector setting
INT_V   ENDS

;-----------------------Main program-----------------------------------------------------------------------
CSEG                            ;  [CODE SEGMENT]
                                ;  Stack pointer (SP) is assumed to have been initialized.
          :                     ;
CLRI                            ;  Disable interrupts.
CLRB    IF20                    ;  Clear external interrupt request flag.
MOV     ILR2,#11111110B         ;  Set interrupt level to 2.

MOV     DDR0,#00000000B         ;  Set pin INT20/AN4 to serve inputs only.

MOV     ADEN,#00000000B         ;  Set pin INT20/AN4 to enable external interrupt inputs.

MOV     EIE2,#00000001B         ;  Enable external interrupt inputs to pin INT20/AN4.
SETI                            ;  Enable interrupts.
          :
;-----------------------Interrupt processing routine-------------------------------------------------------

WARI    MOV     EIE2,#00000000B ;  Disable external interrupt inputs to pin INT20/AN4.
        CLRB    IF20            ;  Clear external interrupt request flag.
```

```
        PUSHW   A
        XCHW    A,T
        PUSHW   A
         :
        User processing
         :
        POPW    A
        XCHW    A,T
        POPW    A
        RETI
        ENDS
;-------------------------------------------------------------------------------------------------------------------
        END
```

CHAPTER 12

A/D CONVERTER

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

12.1 Overview of A/D Converter

An A/D converter, which is of a 10-bit successive approximation type, selects an input signal from eight channel analog inputs. The A/D converter can be activated with software, an internal clock, or the output of an 8/16-bit capture timer/counter (16-bit mode).

■ A/D Conversion Functions

These functions convert the analog voltage (input voltage) input from an analog input to 10-bit digital values.

- An analog input can be selected from eight channels.
- The conversion speed is 38 instruction cycles (when the main clock oscillation frequency is 12.5 MHz, the speed is 12.2 μs).
- When A/D conversion is completed, an interrupt occurs.
- Software can determine that the conversion has been completed.

To activate A/D conversion functions, follow one of the methods given below.

- Activation with a software program.
- Continuous activation through the output of a time-base timer (main clock oscillation frequency divided by 2^8).
- Continuous activation through the output of an 8/16-bit capture timer/counter (16-bit mode).

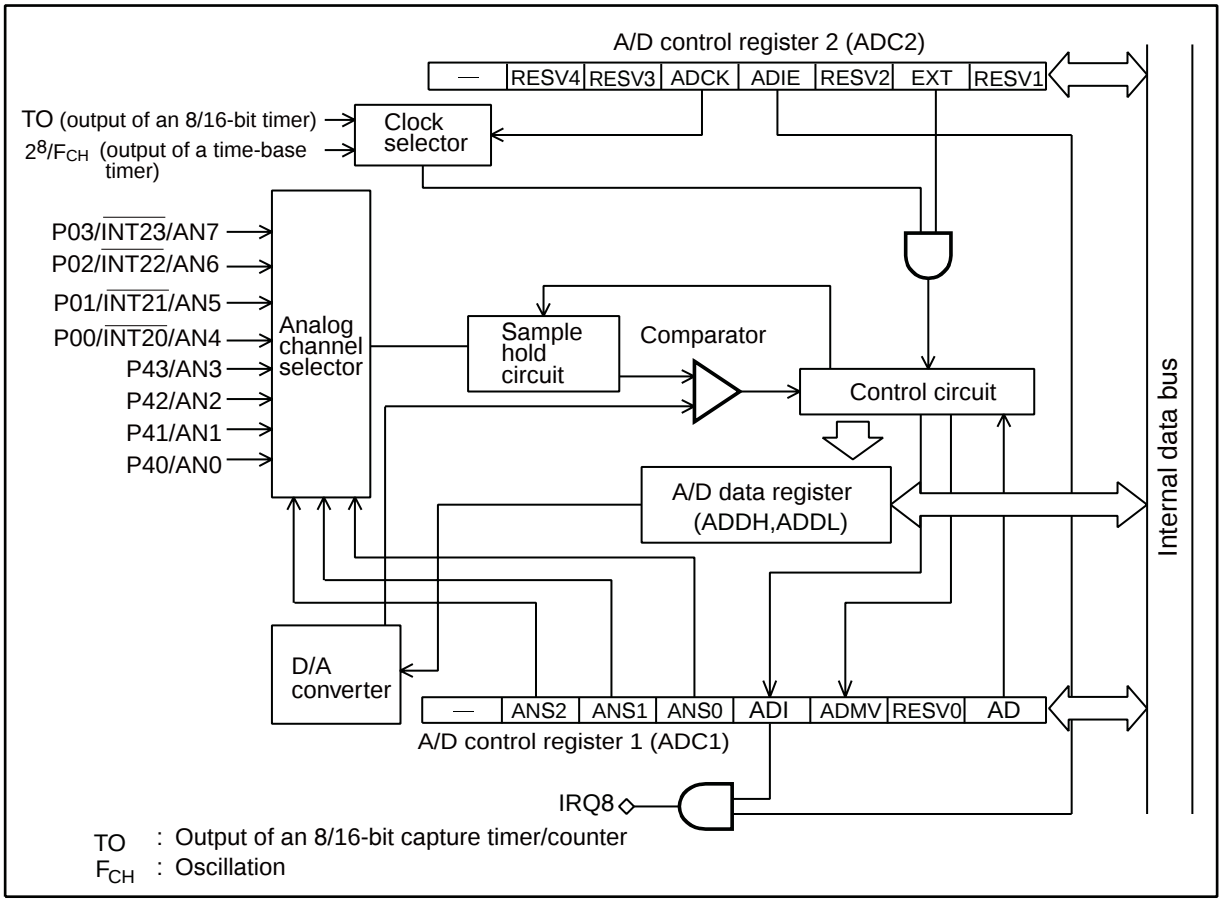
12.2 Configuration of A/D Converter

The A/D converter consists of the following nine blocks.

- **Clock selector (input clock selector for activation of A/D conversion)**
- **Analog channel selector**
- **Sample hold circuit**
- **D/A converter**
- **Comparator**
- **Control circuit**
- **A/D data register (ADDH and ADDL)**
- **A/D control register 1 (ADC1)**
- **A/D control register 2 (ADC2)**

■ Block Diagram of the A/D Converter

Figure 12.2-1 Block Diagram of the A/D Converter



- Clock selector

  The clock selector selects the clock to be used to activate A/D conversion while continuous activation is enabled (ADC2: EXT = 1).

- Analog channel selector

  This circuit selects one out of the eight analog inputs.

- Sample hold circuit

  This circuit holds the input voltage selected by the analog channel selector. By performing the sample hold of the voltage input immediately after the activation of A/D conversion, A/D conversion can be performed without the variance of the input voltage affecting it during A/D conversion (during comparison).

- D/A converter

  This generates the voltage that corresponds to the values set in the ADDH and ADDL registers.

- Comparator

  This compares the input voltage for which sample hold is performed, with the output voltage of the D/A converter to determine which is the greater of the two.

- Control circuit

  The control circuit has the following function.

  • For A/D conversion functions, this circuit determines the values in turn from the MSB in the 10-bit A/D data register toward the LSB based on the large and small signals from the comparator. When the conversion is completed, it sets the interrupt request flag bit (ADC1: ADI).

- A/D data register (ADDH and ADDL)

  The high-order 2 bits of 10-bit A/D data are stored in the ADDH register. The low-order 8 bits of 10-bit A/D data are stored in the ADDL register.

  The ADDH and ADDL registers have the following function.

  • For A/D conversion function, these registers store the results of A/D conversion.

- A/D control register 1 (ADC1)

  This register is used to enable and disable functions, select an analog input, check statuses, and control interrupts.

- A/D control register 2 (ADC2)

  This register is used to select an input clock, enable and disable interrupts, select functions, and perform other activities.

- Interrupts of the A/D converter

  When the set conditions are satisfied at the completion of A/D conversion for IRQ8, if an interrupt request output is enabled (ADC2: ADIE = 1), an interrupt request occurs.

12.3 Pins of A/D Converter

This section describes the pins related to the A/D converter and shows a block diagram of the pins related to the A/D converter.

■ Pins Related to the A/D Converter

The pins related to the A/D converter are P03/$\overline{\text{INT23}}$/AN7 to P00/$\overline{\text{INT20}}$/AN4, and P43/AN3 to P40/AN0 pins.

● P03/$\overline{\text{INT23}}$/AN7 to P00/$\overline{\text{INT20}}$/AN4 and P43/AN3 to P40/AN0

P03/$\overline{\text{INT23}}$/AN7 to P00/$\overline{\text{INT20}}$/AN4, and P43/AN3 to P40/AN0 pins can be used as general-purpose I/O ports (P03 to P00, and P43 to P40) and as analog inputs (AN7 to AN0).

[AN7 to AN0]

When A/D conversion functions are used, input the analog voltage to be converted to these pins. To enable a pin as the analog input, set "1" to the bit that corresponds to the A/D enable register (ADEN), set "0" to the bit that corresponds to the port data direction register (DDR0), and switch the output transistor to OFF, and select one using the bit for selecting an analog input channel (ADC1: ANS0 to ANS2). Even when the A/D converter is used, pins not used as analog inputs can be used as general-purpose I/O ports.

■ Block Diagram of the Pins Related to the A/D Converter

Figure 12.3-1 Block Diagram of P03/$\overline{\text{INT23}}$/AN7 to P00/$\overline{\text{INT20}}$/AN4 Pins

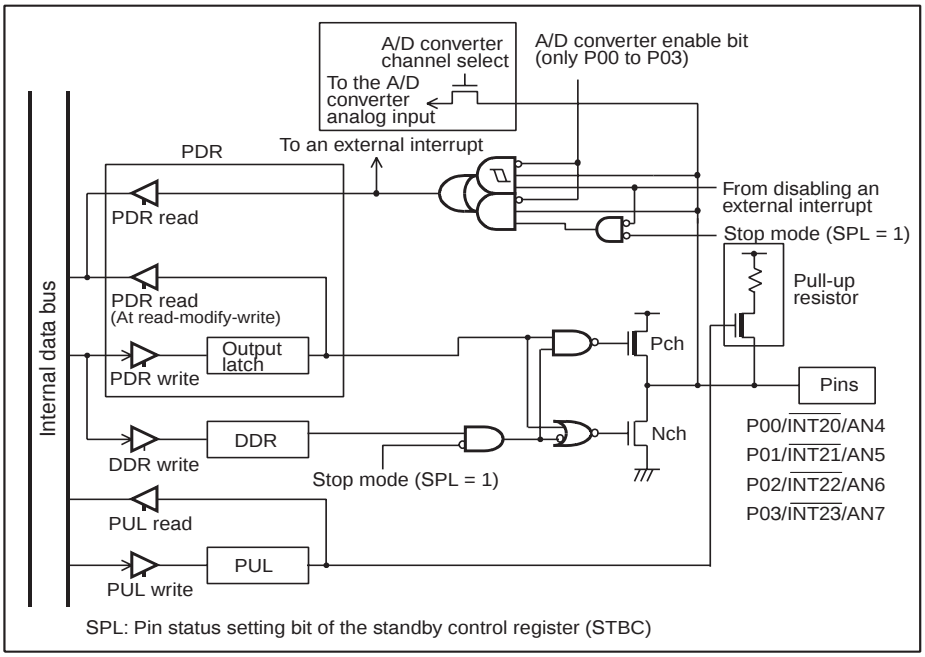
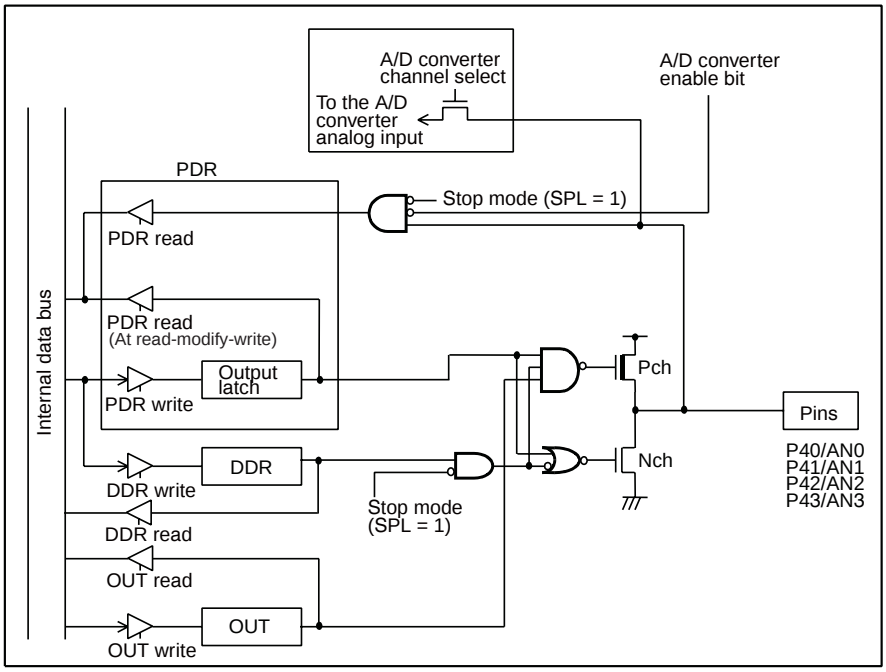


Figure 12.3-2 Block Diagram of P43/AN3 to P40/AN0 Pins

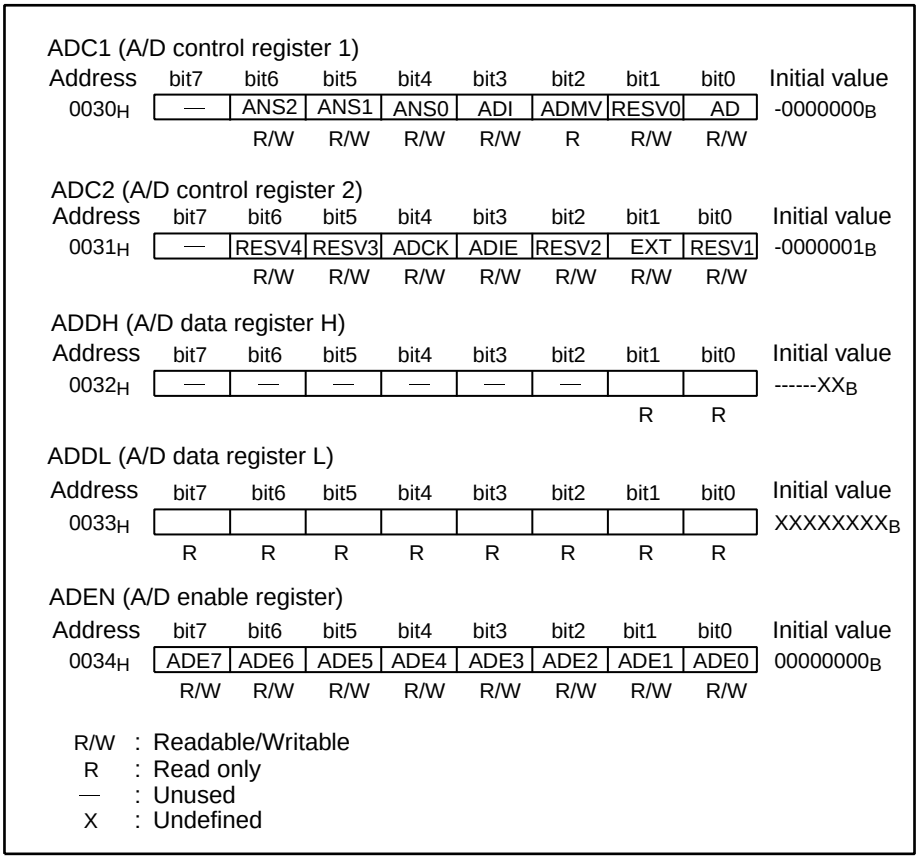


12.4 Registers of A/D Converter

Figure 12.4-1 shows the registers related to the A/D converter.

■ Registers Related to the A/D Converter

Figure 12.4-1 Registers Related to the A/D Converter



12.4.1 A/D Control Register 1 (ADC1)

A/D control register 1 (ADC1) is used to set the enabling and disabling functions of the A/D converter, select an analog input, and check the status.

■ A/D Control Register 1 (ADC1)

Figure 12.4-2 A/D Control Register 1 (ADC1)

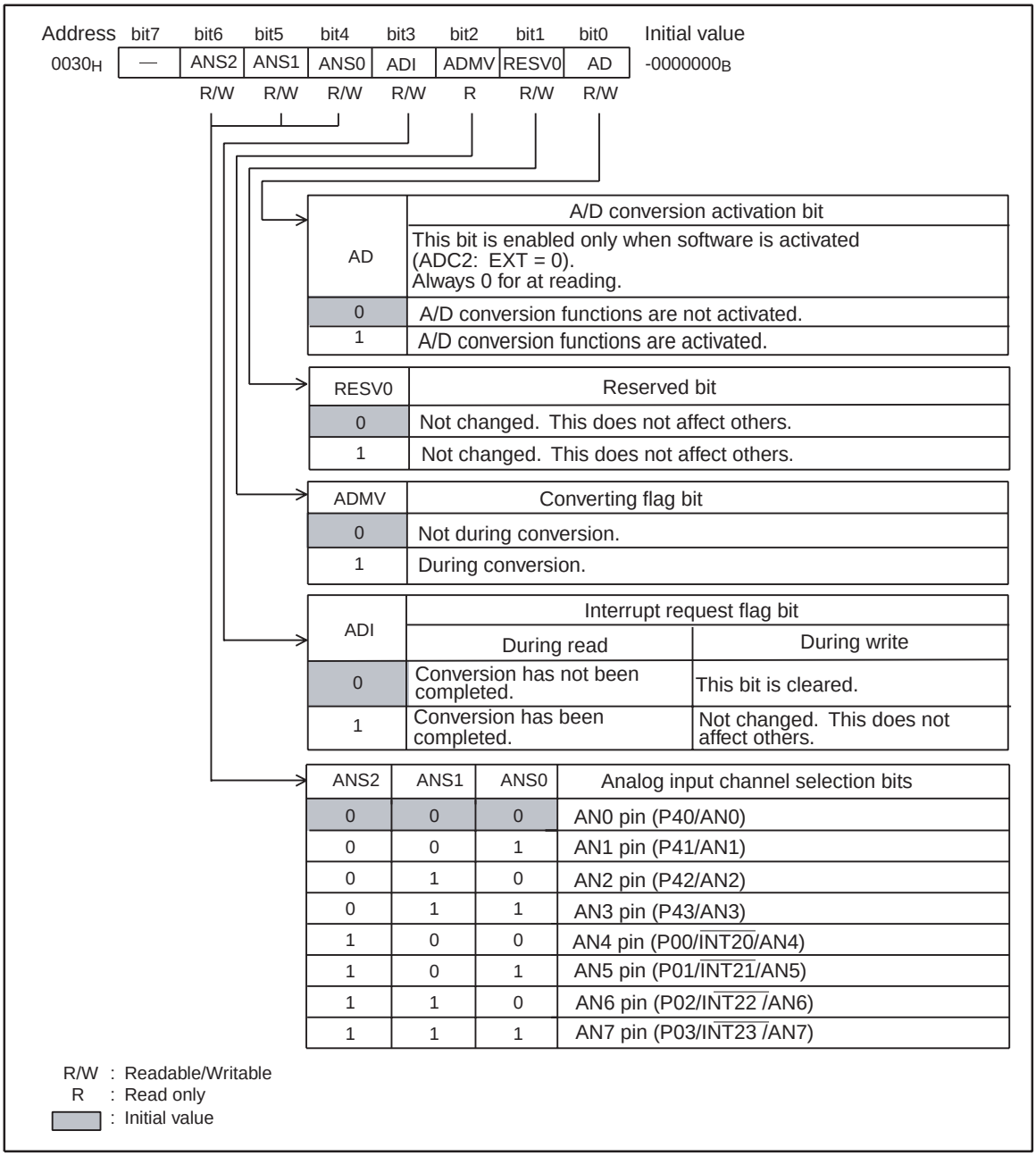


Table 12.4-1 Explanation of Functions of Each Bit in the Bits in A/D Control Register 1 (ADC1)

Bit name		Function
bit7	Unused bit	• The value during read is not determined. • Write does not affect operations.
bit6 to bit4	ANS2, ANS1, ANS0: Analog input channel selection bits	This bit is used to select which pin to be used as an analog input from AN0 to AN7. When software is activated (ADC2: EXT = 0), this bit can be rewritten concurrently with the activation of A/D conversion (AD = 1). **Note:** Pins not used as analog inputs can be used as general-purpose ports.
bit3	ADI: Interrupt request flag bit	• When A/D conversion functions are enabled: • When A/D conversion is completed, "1" is set to this bit. • When this bit and the bit for enabling an interrupt request (ADC2: ADIE) are "1", an interrupt request is output. • At write, this bit is cleared with "0". When "1" is set to this bit, nothing is changed or affected by this.
bit2	ADMV: Convertion flag bit	This bit indicates that A/D conversion is being performed, when A/D conversion functions are enabled. During conversion (comparison), this bit is set to "1". **Note:** This bit is read-only. The written value is ignored and nothing is affected by the value.
bit1	RESV0: Reserved bit	• The value during read is not determined. • Write does not affect operations.
bit0	AD: A/D conversion activation bit	• This bit is used to activate A/D conversion functions with software. • In the state where continuous activation is not performed (ADC2: EXT = 0), when "1" is set to this bit, A/D conversion functions are activated. **Notes:** • Even if "0" is written to this bit, the operation of A/D conversion functions cannot be stopped. The read value is always "0". • During continuous activation, this bit is ignored.

12.4.2 A/D Control Register 2 (ADC2)

A/D control register 2 (ADC2) is used to select an input clock, enable and disable an interrupt and continuous activation.

■ A/D Control Register 2 (ADC2)

Figure 12.4-3 A/D Control Register 2 (ADC2)

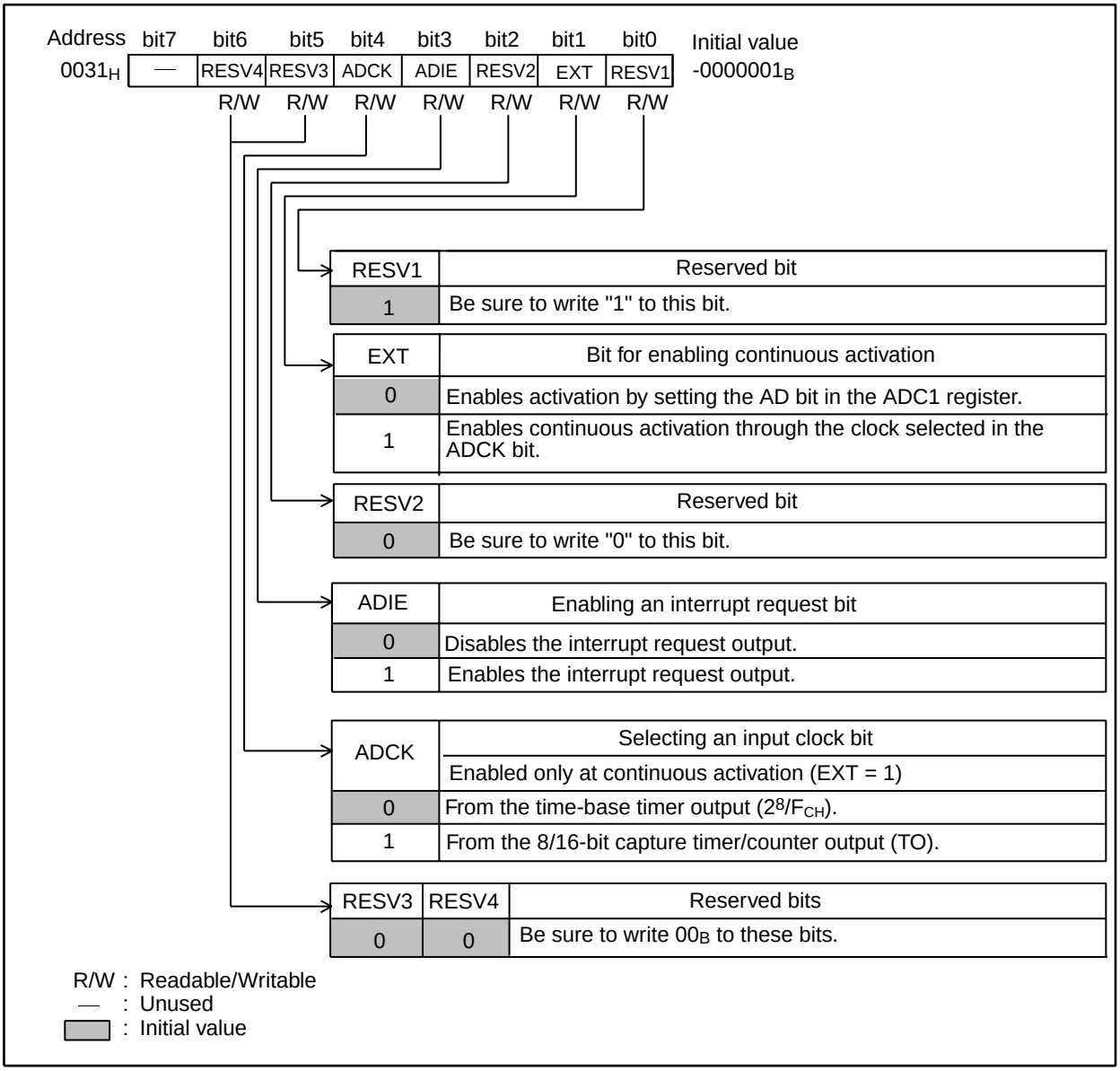


Table 12.4-2 Explanation of Functions of Each Bit in A/D Control Register 2 (ADC2)

Bit name		Function
bit7	Unused bit	• The value during read is not determined. • Write does not affect operations.
bit6, bit5	RESV4,RESV3: Reserved bits	• This bit is a reserved bit. • Be sure to write 00_B to these bits.
bit4	ADCK: Selecting an input clock bit	This bit is used to select an input clock for activation of A/D conversion functions in the state where continuous activation is performed (EXT = 1). When this bit is "0", the internal clock with an oscillation frequency (selected using the output of a time-base timer) divided by 2^8 is selected. When "1", the output of an 8/16-bit capture timer/counter (TO: 16-bit mode) is selected.
bit3	ADIE: Enabling an interrupt request bit	This bit is used to enable and disable the output of an interrupt to the CPU. When this bit and the interrupt request flag bit (ADC1: ADI) are "1", an interrupt request is output.
bit2	RESV2: Reserved bit	• This bit is a reserved bit. • Be sure to write "0" to this bit.
bit1	EXT: Bit for enabling continuous activation	This bit is used to select whether the A/D conversion functions are to be activated with software or activated continuously in synchronization with an input clock. When this bit is "0", software activation with the bit for activating A/D conversion (ADC1: AD) is enabled. When "1", continuous activation on the rising edge of the clock selected using the bit for selecting an input clock (ADC2: ADCK) is enabled.
bit0	RESV1: Reserved bit	• This bit is a reserved bit. • Be sure to write "1" to this bit.

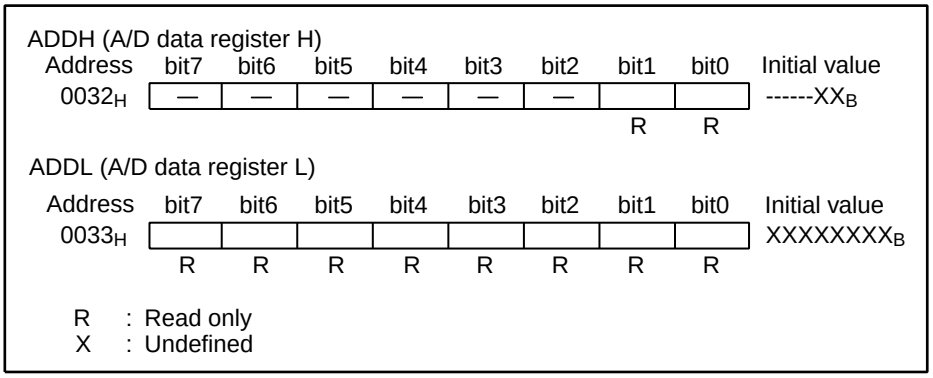
12.4.3 A/D Data Register (ADDH and ADDL)

A/D data register (ADDH and ADDL) stores the results of A/D conversion at 10-bit A/D conversion.
The high-order 2 bits of 10-bit data correspond to the ADDH register. The low-order 8 bits correspond to the ADDL register.

■ A/D Data Register (ADDH and ADDL)

Figure 12.4-4 shows the bit configuration of the A/D data registers.

Figure 12.4-4 A/D Data Registers (ADDH and ADDL)



Of the 10-bit A/D data, the high-order 2 bits correspond to bits 1 and 0 in the ADDH register. The low-order 8 bits correspond to bits 7 to 0 in the ADDL register.

- When A/D conversion functions are enabled

  When A/D conversion is activated, after about 38 instruction cycles, the data on the conversion results are fixed and stored to these registers. Therefore, after A/D conversion, read these registers (conversion results), write "0" to the ADI bit (bit3) in the ADC1 register until the next A/D conversion is completed, and clear the flags after A/D conversion. During A/D conversion, the values in these registers are not determined. When A/D conversion functions are enabled, these registers function as read-only registers.

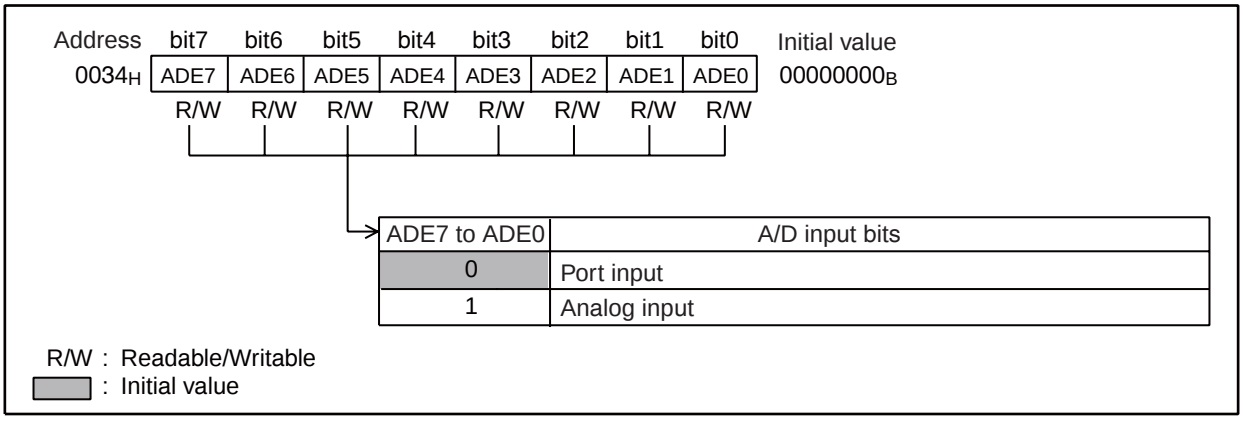
12.4.4 A/D Enable Register (ADEN)

The ADEN register is used to select the analog input port that corresponds to different pins. Writing "1" to an appropriate ADEN register bit enables analog input.

■ A/D Enable Register (ADEN)

Figure 12.4-5 shows the bit configuration of the A/D enable register.

Figure 12.4-5 A/D Enable Register (ADEN)



An A/D input port can be used as a general-purpose I/O port.

The ADEN register is used to select the port that corresponds to the analog input.

Set "1" to the corresponding bit in the ADEN register for the port to be used for analog input. This prevents the DC pass when the middle level voltage is applied to the A/D input port.

When this register is to be used as the A/D input port, do not select the bit that indicates use of a pull-up resistor from the pull-up setting register.

12.5 Interrupt of A/D Converter

A factor for an interrupt of the A/D converter is the following.

- **Completion of conversion when A/D conversion functions are enabled**

■ Interrupt when A/D Conversion Functions are Enabled

When A/D conversion is completed, the interrupt request flag bit (ADC1: ADI) is set to "1". At this time, if the bit for enabling an interrupt request is enabled (ADC2: ADIE = 1), an interrupt request to the CPU (IRQ8) occurs. Write "0" to the ADI bit using the routine for interrupt handling to clear the interrupt request.

The ADI bit is set when A/D conversion is completed, irrespective of the value of the ADIE bit.

Note:

When the ADI bit is "1", if the ADIE bit is enabled (changed from "0" to "1"), an interrupt request occurs immediately.

■ Register and Vector Table Related to the Interrupt of the A/D Converter

Table 12.5-1 Register and Vector Table Related to the Interrupt of the A/D Converter

Interrupt name	Register to set the interrupt level			Address of the vector table	
	Register	Bit to be set		High order	Low order
IRQ8	ILR3 ($007D_H$)	L81 (bit1)	L80 (bit0)	$FFEA_H$	$FFEB_H$

See Section "3.4.2 Steps in the Interrupt Operation " for the interrupt operation.

12.6 Operations of A/D Converter Functions

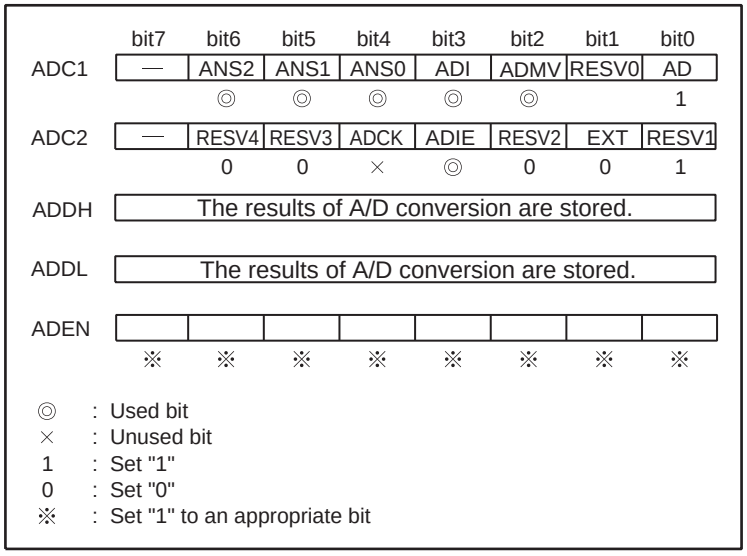
The A/D converter can be activated with software or activated continuously.

■ Activating the A/D Converter Functions

● Software activation

To activate A/D conversion functions with software, set registers as shown in Figure 12.6-1 .

Figure 12.6-1 Setting A/D Conversion Functions (at Software Activation)

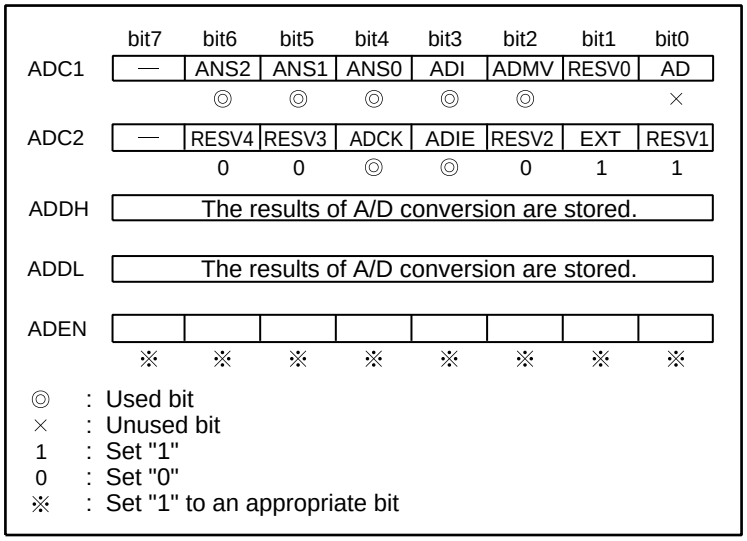


When A/D conversion is activated, the operations of A/D conversion functions are started. In addition, even during conversion, A/D conversion functions can be reactivated.

● Continuous activation

To activate A/D conversion functions continuously, set registers as shown in Figure 12.6-2 .

Figure 12.6-2 Setting A/D Conversion Functions (at Continuous Activation)



When continuous activation is enabled, A/D conversion is activated on a rising edge of the selected input clock and the operations of A/D conversion functions are started. When continuous activation is disabled (ADC2: EXT = 0), continuous activation is stopped and activation with software is possible.

■ Operations of A/D Conversion Functions

The operations of the A/D converter are described here. It takes about 38 instruction cycles from activating A/D conversion to completing it.

1. When A/D conversion is activated, the conversion in-progress flag bit is set (ADC1: ADMV = 1), and the analog input set is connected to the sample hold circuit.
2. During about 16 instruction cycles, the voltage of the analog input is captured and held in the capacitor for internal sample hold. This voltage is held until A/D conversion has been completed.
3. The comparator compares the voltage captured and held in the capacitor for sample hold with the reference voltage for A/D conversion from the MSB to the LSB. The results are transferred to the ADDH and ADDL registers in turn.
4. When all the results have been transferred to the ADDH and ADDL registers, the conversion in-progress flag bit is cleared (ADC1: ADMV = 0), and the interrupt request flag bit is set (ADC1: ADI = 1).

12.7 Notes on Using A/D Converter

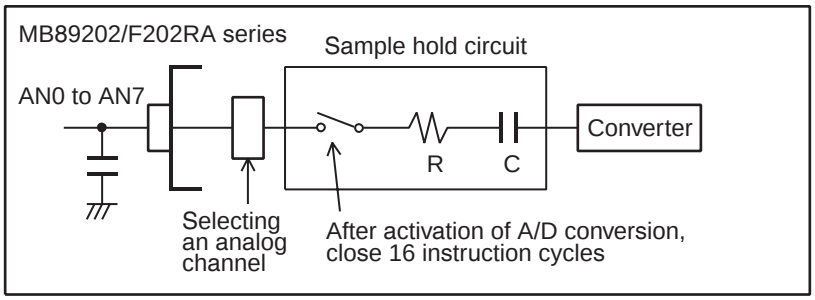
This section describes notes on using the A/D converter.

■ Notes on Using the A/D Converter

● Input impedance of the analog input

The A/D converter contains the sample hold circuit as shown in Figure 12.7-1 , captures the voltage of the analog input, and holds it in the capacitor for sample hold in about 16 instruction cycles, after activation of A/D conversion. Accordingly, when the output impedance of the external circuit of the analog input is high, the analog input voltage may not be stabilized during analog input sampling period. Therefore, set the output impedance of the external circuit to a sufficiently low level (lower than about 4 kΩ). If the output impedance of the external circuit cannot be set low, it is recommended that a capacitor with about 0.1 μF be added externally to the analog input.

Figure 12.7-1 Equivalent Circuit of Analog Input



● Notes on setting using a program

- When A/D conversion functions are enabled, the values in the ADDH and ADDL registers are held without being changed until the activation of A/D conversion. However, once A/D conversion is activated, the values in the ADDH and ADDL registers become undefined immediately.
- When A/D conversion functions are enabled, do not reselect an analog input channel (ADC1: ANS3 to ANS0). Especially, during continuous activation, disable continuous activation (ADC2: EXT = 0), and wait for the conversion in-progress flag bit (ADC1: ADMV) to be "0" for reselection.
- The A/D converter is stopped via a reset and activation of the stop mode, and all registers are initialized.
- When the interrupt request flag bit (ADC1: ADI) is "1" and an interrupt request is enabled (ADC2: ADIE = 1), recovery from interrupt handling is no longer possible. Be sure to clear the ADI bit.

Note:

When A/D conversion is completed, if the next conversion is reactivated, the interrupt request flag bit (ADC1: ADI) is not set.

- 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.

- Conversion time

  Changing the oscillation frequency or clock speed (gear functions) affects the conversion speed of A/D conversion functions.

- Input clock of continuous activation

  The output of an 8/16-bit capture timer/counter is affected by gear functions. The output of a time-base timer is not affected by gear functions. Clearing a time-base timer affects cycles.

  Since the output of an 8/16-bit capture timer/counter is the output of the 16-bit mode, the 8-bit mode cannot be used.

12.8 Program Example for A/D Converter

This section shows a program example of the 10-bit A/D converter.

■ Program Example of the A/D Conversion Functions

- Processing specifications

  The analog voltage to be applied to the AN0 pin is converted to digital voltage through software activation.

  In this example, completion of conversion is detected in a loop in the program without using interrupts.

- Coding example

```
PDR4    EQU     000FH           ;  Address of port 4 data register 4
ADC1    EQU     0030H           ;  Address of A/D control register 1
ADC2    EQU     0031H           ;  Address of A/D control register 2
ADDH    EQU     0032H           ;  Address of A/D data register H
ADDL    EQU     0033H           ;  Address of A/D data register L
ADEN    EQU     0034H           ;  Enables the A/D input pin.
AN0     EQU     PDR4:0          ;  Defines the AN0 analog input.
ADE0    EQU     ADEN:0          ;  Enables the AN0 analog input.
ADI     EQU     ADC1:3          ;  Defines the interrupt request flag bit.
ADMV    EQU     ADC1:2          ;  Defines the conversion in-progress flag bit.
AD      EQU     ADC1:0          ;  Defines the bit for activating A/D conversion (software
                                   activation).
EXT     EQU     ADC2:1          ;  Defines the bit for enabling continuous activation.
;-----------------------Main program-----------------------------------------------------------
        CSEG                    ;  [CODE SEGMENT]
         :
        SETB    AN0             ;  Sets the P40/AN0 pin to the analog input.
        CLRI                    ;  Disables interrupts.
        SETB    ADE0            ;  Enables the AN0 pin.
        CLRB    EXT             ;  Disables continuous activation.
AD_WAIT
        BBS     ADMV,AD_WAIT    ;  Loop for verifying that the A/D converter is stopped.
        MOV     ADC1,#00000000B ;  Selects analog input channel 0 (AN0), clears the
                                   interrupt request flag, does not perform software
                                   activation.
        MOV     ADC2,#00000001B ;  Disables the interrupt request output, selects A/D
                                   conversion functions, and selects software activation.
        SETI                    ;  Enables interrupts.
         :
        SETB    AD              ;  Activates software.
AD_CONV
        BBS     ADMV,AD_CONV    ;  Loop for waiting for completion of A/D conversion
                                   (at about 12.2 µs/12.5 MHz)
        CLRB    ADI             ;  Clears the interrupt request flag.
```

```
        MOV     A,ADDL          ;  Reads A/D conversion data (low-order 8 bits).
        MOV     A,ADDH          ;  Reads A/D conversion data (high-order 2 bits).
         :
         :
        ENDS
;-------------------------------------------------------------------------------------------------------------------
        END
```

CHAPTER 13

UART

This chapter describes the functions and operations of UART.

13.1 Overview of UART

UART is a general-purpose communication interface for serial data. UART allows variable-length serial data to be transferred synchronously or asynchronously with a clock. The transfer format is NRZ. The dedicated baud rate generator, external clock, or internal timer (8-bit PWM timer) settings determine the data transfer format.

■ Functions of UART

UART supports (serial I/O) functions for sending serial data to, or receiving serial data from a CPU or peripheral functions.

The full-duplex double-buffer enables bi-directional full-duplex communication.

- Synchronous data transfer mode or asynchronous data transfer mode can be selected.
- The internal baud rate generator allows one of 14 baud rates to be selected. Also, external clock input and 8-bit PWM timer output allow user-defined baud rates to be specified.
- The length of data is variable. When no parity is used, 7 bits to 9 bits are available. When parity is used, 6 bits to 8 bits are available (Table 13.1-1).
- The data transfer format is NRZ (Non Return to Zero).

Table 13.1-2 provides the transfer rates of the dedicated baud rate generator, and Table 13.1-3 provides the transfer rates of the external clock.

Table 13.1-1 UART Operating Modes

Operating mode	Data length		Synchronization mode	Stop bit length
	Parity not used	Parity used		
0	7 bits	6 bits	Synchronous/asynchronous	1 bit or 2 bits *
1	8 bits	7 bits	Synchronous/asynchronous	1 bit or 2 bits *
2	8+1 bits	-	Synchronous/asynchronous	1 bit or 2 bits *
3	9 bits	8 bits	Synchronous/asynchronous	1 bit or 2 bits *

*: Only one bit is allowed for the stop bit length when data is received. The second bit is ignored even if it is received.

■ Serial Switch

UART and 8-bit serial I/O use the same pins, thus they cannot be simultaneously used. The serial switch circuit needs be used to select either of them.

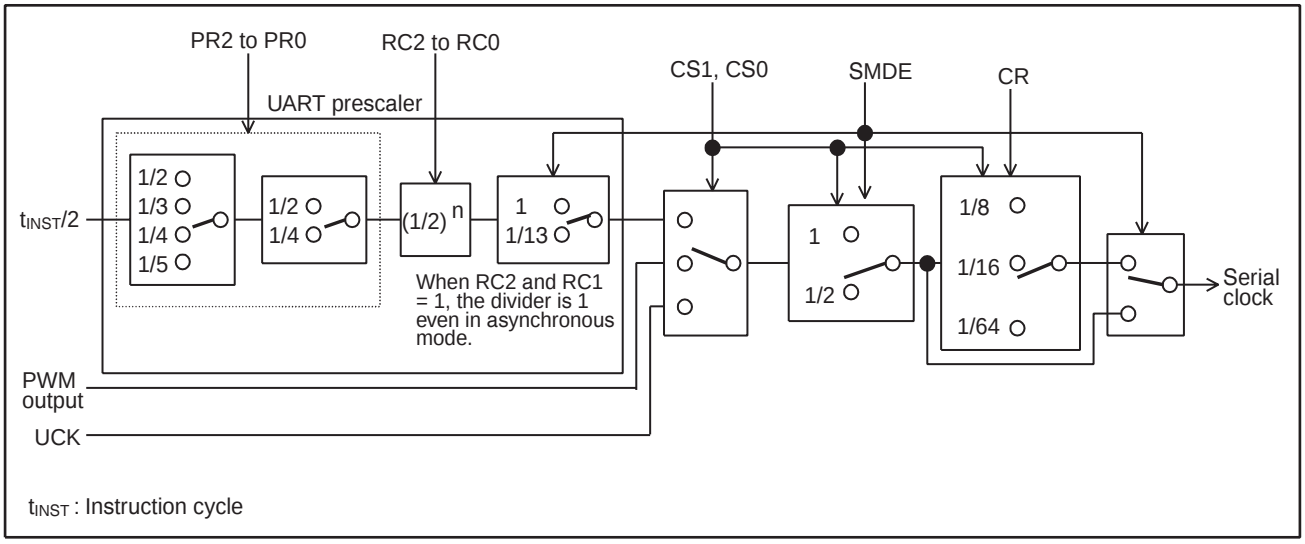
When UART is selected using the serial switch, P30/UCK/SCK is used as the UART serial clock I/O pin (UCK), P31/UO/SO is used as the UART data output pin (UO), and P32/UI/SI is used as the UART data input pin (UI).

Note:

In this chapter, the pin function switch and register functions are explained on the presupposition that UART is selected using the serial switch circuit.

■ Choice of the Transfer Clock Rate

Figure 13.1-1 Baud Rate Generator and Serial Clock Generator



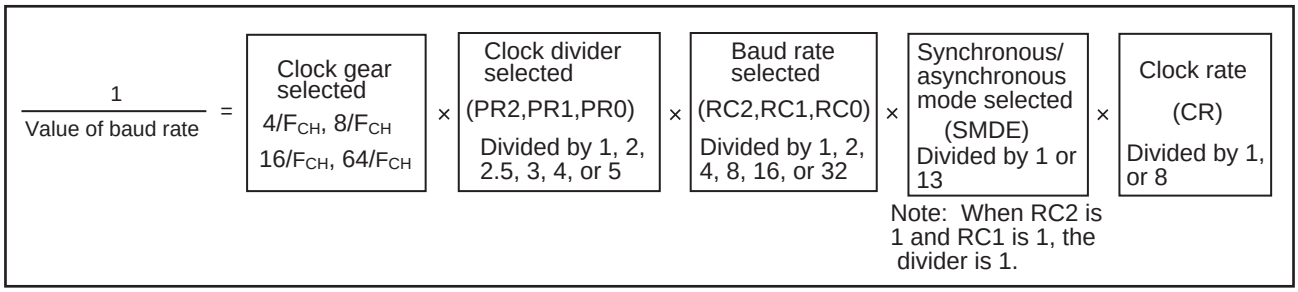
● Example of the baud rates selectable when the dedicated baud rate generator is used

Table 13.1-2 lists the baud rates selectable when the dedicated baud rate generator is used.

Table 13.1-2 Transfer Cycles and Transfer Rates Available for the Dedicated Baud Rate Generator (when F_{CH} = 12.5 MHz)

				Transfer rate (μs/baud)			
				Clock divider = 2.5 (PR2=0, PR1=1, PR0=0)		Clock divider = 1 (PR2=0, PR1=0, PR0=0)	
RC2	RC1	RC0	Divider for baud rate (multiplier = n)	Asynchronous transfer	Synchronous transfer	Asynchronous transfer	Synchronous transfer
0	0	0	1(0)	83/12019	0.8/1.25M	33/30048	0.32/3.1M
0	0	1	2(1)	166/6010	1.6/625k	67/15024	0.64/1.6M
0	1	0	4(2)	333/3005	3.2/313k	133/7512	1.28/781k
0	1	1	8(3)	666/1503	6.4/156k	266/3756	2.56/391k
1	0	0	16(4)	1331/751	12.8/78k	532/1878	5.12/195k
1	0	1	32(5)	2662/375	25.6/39k	1065/939	10.24/98k
1	1	0	2(1)	13/78125	1.6/625k	5.12/195k	0.64/1.6M
1	1	1	16(4)	102/9766	12.8/78k	41/24k	5.12/195k

Figure 13.1-2 Example of Calculating the Baud Rate



Notes:

The baud rate is specified using the clock gear register (CS1 and CS0), clock divider registers (PR2, PR1, and PR0), or baud rate selection registers (RC2, RC1, and RC0). For the example of calculating the baud rate, see Table 13.1-2 .

- Asynchronous transfer mode

  $1/12019\text{bps} = 0.8\ \mu\text{s}\{4/F_{CH} \times 2.5(PR2=0,PR1=1,PR0=0)\} \times 1(CS1=CS0=1) \times 8(\text{asynchronous}) \times 1(RC2=RC1=RC0=0) \times 13(\text{asynchronous})$

- Synchronous transfer mode

  $1/1.25\text{Mbps} = 0.8\ \mu\text{s}\{4/F_{CH} \times 2.5(PR2=0,PR1=1,PR0=0)\} \times 1(CS1=CS0=1) \times 1(\text{synchronous}) \times 1(RC2=RC1=RC0=0) \times 1(\text{synchronous})$

Table 13.1-3 provides an example of the baud rates selectable when an external clock is used.

Table 13.1-3 Transfer Cycles and Transfer Rates Selectable for an External Clock

Asynchronous transfer mode				Synchronous transfer mode		
Divider for baud rate		Transfer cycle	Transfer rate (bps) *	Divider for baud rate	Transfer cycle	Transfer rate (bps) *
CR=0	16	$256/F_{CH}$ or more	48828 or less	1	$16/F_{CH}$ or more	781 k or less
CR=1	64	$1024/F_{CH}$ or more	12207 or less			

*: The minimum value of F_{CH} specified for 12.5 MHz is external clock cycle $16/F_{CH} = 1.28\ \mu\text{s}$.
F_{CH}: Oscillation frequency

Figure 13.1-3 Example of Calculating the Baud Rate (when an External Clock is Selected)

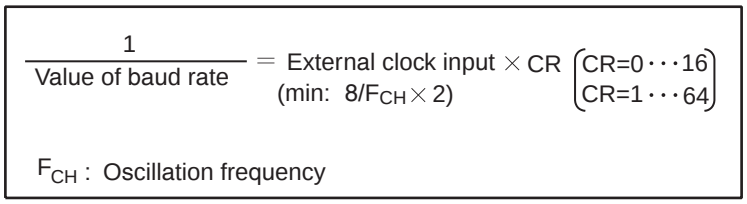


Table 13.1-4 provides an example of the baud rates selectable when the 8-bit PWM timer is used.

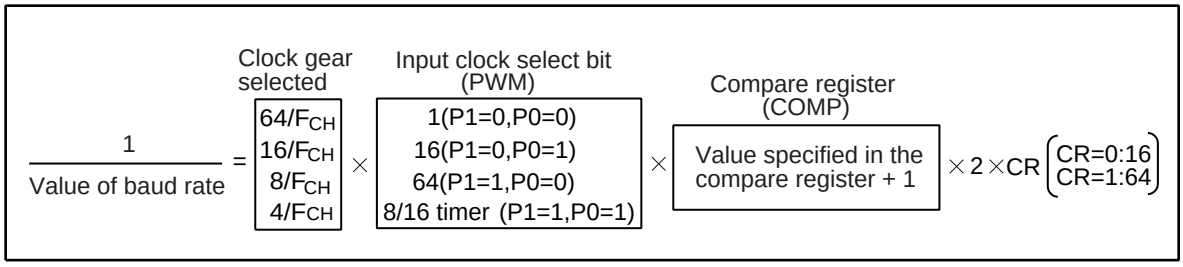
Table 13.1-4 Transfer Cycles and Transfer Rates Selectable for the 8-bit PWM Timer

PWM timer count clock cycle	Asynchronous transfer mode			Synchronous transfer mode	
	Divider for clock		Transfer rate (bps)	Divider for clock	Transfer rate (bps)
$1t_{INST}$	CR=0	16	97656 to 763	2	781k to 6.1k
	CR=1	64	24414 to 191		
$16t_{INST}$	CR=0	16	6103 to 47.8	2	48828 to 381.5
	CR=1	64	1526 to 11.9		
$64t_{INST}$	CR=0	16	1526 to 11.9	2	12207 to 95.4
	CR=1	64	381.5 to 3		
8/16-bit capture timer/counter	CR=0	16	48828 to 381.5	2	391k to 3k
	CR=1	64	12207 to 95.4		

t_{INST}: Instruction cycle
The system clock control register (SYCC) selects the maximum clock speed (CS1 and CS0 = 11_B, 1 instruction cycle = $4/F_{CH}$) in active mode.

Figure 13.1-4 shows an example of calculating the baud rate when the PWM timer is selected.

Figure 13.1-4 Example of Calculating the Baud Rate (when the PWM Timer is Selected)



The value of the baud rate is determined by the clock input specified in the clock dividing rate register (CS1 and CS0). The clock input is determined with an external clock (PWM timer). For calculation, see Table 13.1-3 and Table 13.1-4 .

- When an external clock is selected (F_{CH} = 12.5 MHz)

  1/49kbps=1.28 µs (min.) × 16 (CR=0)

- When the PWM timer is selected (F_{CH} = 12.5 MHz)

  1/98kbps=0.32 µs ($4/F_{CH}$) × 1 (P1=0,P0=0) × 1 (COMR=0) × 2 × 16 (CR=0)

  1/24414 bps=0.32 µs ($4/F_{CH}$) × 1 (P1=0,P0=0) × 1 (COMR=0) × 2 × 64 (CR=1)

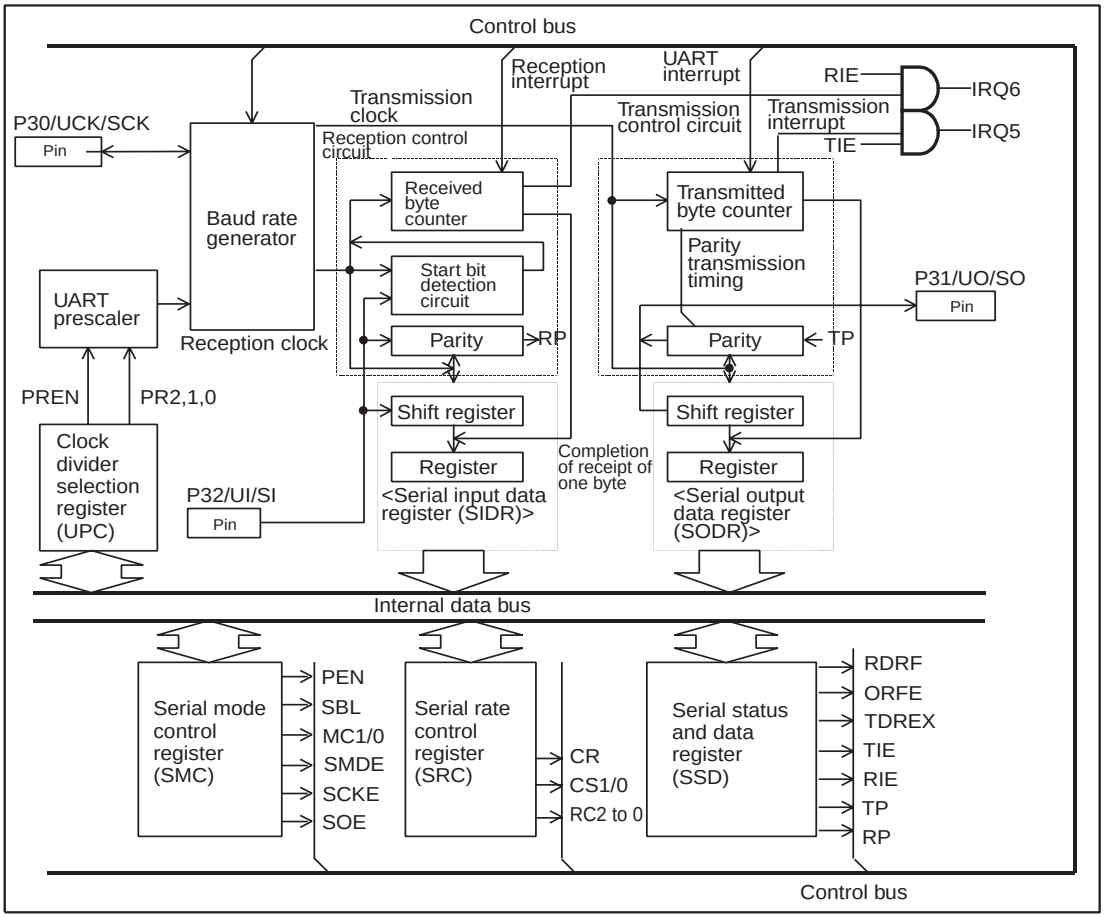
13.2 Configuration of UART

UART consists of the following ten registers and components:

- **Serial mode control register (SMC)**
- **Serial rate control register (SRC)**
- **Serial status and data register (SSD)**
- **Serial input data register (SIDR)**
- **Serial output data register (SODR)**
- **Baud rate generator**
- **Reception control circuit**
- **Transmission control circuit**
- **Clock divider selection register (UPC)**
- **UART prescaler**

■ Block Diagram of UART

Figure 13.2-1 Block Diagram of UART



- Serial mode control register (SMC)

  The SMC register controls UART operating mode. This register specifies the parity setting, stop bit length, operating mode (data length), and synchronous/asynchronous mode, and enables/disables UART serial clock output and serial data output.

- Serial rate control register (SRC)

  The SRC register controls the UART data transfer speed (baud rate). This register selects the input clock and specifies the transfer rate to be applied when the baud rate generator is used.

- Serial status and data register (SSD)

  The SSD register indicates UART transmitting/receiving status, status in an error, parity received, or data received at bit8. This register also enables/disables interrupts or specifies and confirms parity transmitted or data transmitted with bit8.

- Serial input data register (SIDR)

  The SIDR register stores received data. Serial input is converted, then stored into this register. However, the most significant bit of 9-bit data is stored in the SSD RD8/RP bit.

- Serial output data register (SODR)

  The SODR register specifies data to be transmitted. Data written into this register is converted to serial format, then output. The most significant bit of 9-bit data is set in the SSD TD8/TP bit.

- Clock generator

  The clock generator generates the transmit/receive clock in accordance with the dedicated baud rate generator, external clock, and 8-bit PWM timer output.

- Reception control circuit

  The reception control circuit consists of the received byte counter, start bit detection circuit, and received parity handling circuit.

  The received byte counter takes count of received data. When a unit of data that corresponds to the specified data length is fully received, an interrupt request is generated.

  The start bit detection circuit detects start bits in serial input signals. When the start bit detection circuit detects a start bit, it writes data into the SIDR with shifts in accordance with the transfer rate.

  When parity is used, the received parity handling circuit stores the parity bit in the data received. It also stores the most significant bit of 9-bit data received.

- Transmission control circuit

  The transmission control circuit consists of the transmitted byte counter and transmitted parity handling circuit.

  The transmitted byte counter takes count of data to be transmitted. When a unit of data that corresponds to the specified data length is fully transmitted, an interrupt request is generated.

  When parity is used, the transmitted parity handling circuit generates a parity bit for the data to be transmitted. It sets the most significant bit for data transmitted when it is made up of 9 bits.

- UART interrupt sources

[Reception]

When data with the specified length is correctly received or when the overrun error or framing error occurs while data is being received, the reception interrupt request (IRQ6) is generated if the reception interrupt request is enabled (SSD: RIE = 1).

[Transmission]

When data to be transmitted is written into the SODR register, sent to the internal shift register, and the next data then becomes writable, the transmission interrupt request (IRQ5) is generated if the transmission interrupt request is allowed (SSD: TIE = 1).

- UART prescaler, baud rate generator, clock divider selection register

The clock input to the baud rate generator is changeable by switching the rate of division using the clock divider selection registers.

13.3 Pins of UART

Pins relating to UART are the clock I/O pin (P30/UCK/SCK), serial data output pin (P31/UO/SO), and serial data input pin (P32/UI/SI).

■ UART Relating Pins

- P30/UCK/SCK

  This pin functions as the general-purpose I/O port (P30), UART clock I/O pin (UCK), or 8-bit serial clock I/O pin (SCK). When clock output is enabled (SMC: SCKE = 1), this pin functions as the UART clock output pin (UCK) regardless of the value in the corresponding port direction register. When this pin functions as the UART clock, do not use any external clock (SRC: CS1 and CS0 must be other than 00_B). When using this pin as the UART clock input pin, disable clock output (SMC: SCKE = 0) and set it as the input port using the corresponding port direction register (DDR3: bit0 = 0). In this case, be sure to select the external clock (SRC: CS1 and CS0 = 00_B).

- P31/UO/SO

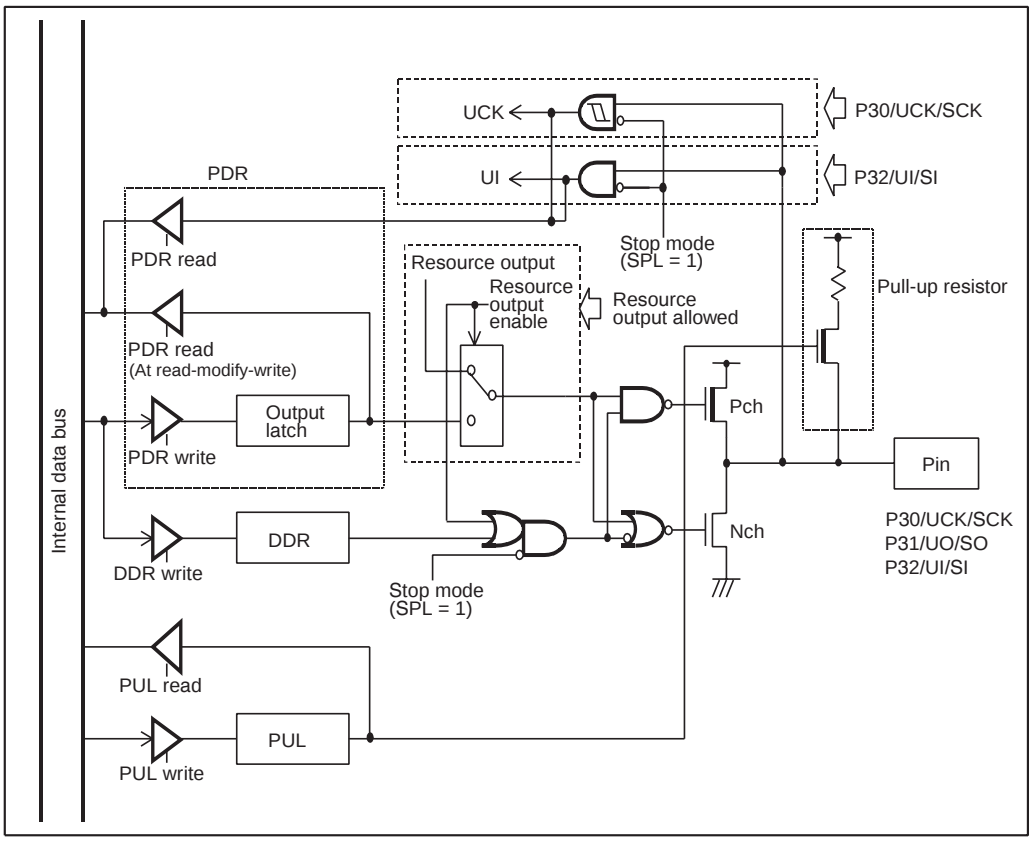
  This port functions as the general-purpose I/O port (P31), UART serial data output pin (UO), or 8-bit serial data output pin (SO). When serial data output is enabled (SMC: SOE = 1), this pin functions as the UART serial data output pin (UO) regardless of the value in the corresponding port direction register.

- P32/UI/SI

  This port functions as the general-purpose I/O port (P32), UART serial data input pin (UI), or 8-bit serial data input pin (SI). When using this pin as the UART serial data input pin, set this pin as the input port by using the corresponding port direction register (DDR3: bit2 = 0).

■ Block Diagram of the UART-relating Pins

Figure 13.3-1 Block Diagram of UART-relating Pins



When use of the pull-up resistor is selected in the pull-up setting register, the pin status does not become Hi-Z but "H" level (pull-up state) in stop mode (SPL = 1). However, the pull-up resistor is not applied during reset; accordingly, the pin status becomes Hi-Z.

13.4 Registers of UART

Figure 13.4-1 shows the UART-relating registers.

■ UART-relating Registers

Figure 13.4-1 UART-relating Registers

SMC (serial mode control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0028_H	PEN	SBL	MC1	MC0	SMDE	—	SCKE	SOE	$00000\text{-}00_B$
	R/W	R/W	R/W	R/W	R/W		R/W	R/W	

SRC (serial rate control register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0029_H	—	—	CR	CS1	CS0	RC2	RC1	RC0	$\text{--}011000_B$
			R/W	R/W	R/W	R/W	R/W	R/W	

SSD (serial status and data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$002A_H$	RDRF	ORFE	TDRE	TIE	RIE	—	TD8/TP	RD8/RP	$00100\text{-}1X_B$
	R	R	R/W	R/W	R/W		R/W	R	

SIDR (serial input data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$002B_H$									$XXXXXXXX_B$
	R	R	R	R	R	R	R	R	

SODR (serial output data register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$002B_H$									$XXXXXXXX_B$
	W	W	W	W	W	W	W	W	

UPC (clock divider selection register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
$002C_H$	—	—	—	—	PREN	PR2	PR1	PR0	$\text{----}0010_B$
					R/W	R/W	R/W	R/W	

SSEL (serial switch register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
$003B_H$	—	—	—	—	—	—	—	SSEL	$\text{-------}0_B$
								R/W	

R/W : Readable/Writable
R : Read only
W : Write only
— : Unused
X : Undefined

13.4.1 Serial Mode Control Register (SMC)

The serial mode control register (SMC) specifies the parity setting, stop bit length, operating mode (data length), and synchronous/asynchronous mode, and enables/disables UART serial clock output and serial data output.

■ Serial Mode Control Register (SMC)

Figure 13.4-2 Serial Mode Control Register (SMC)

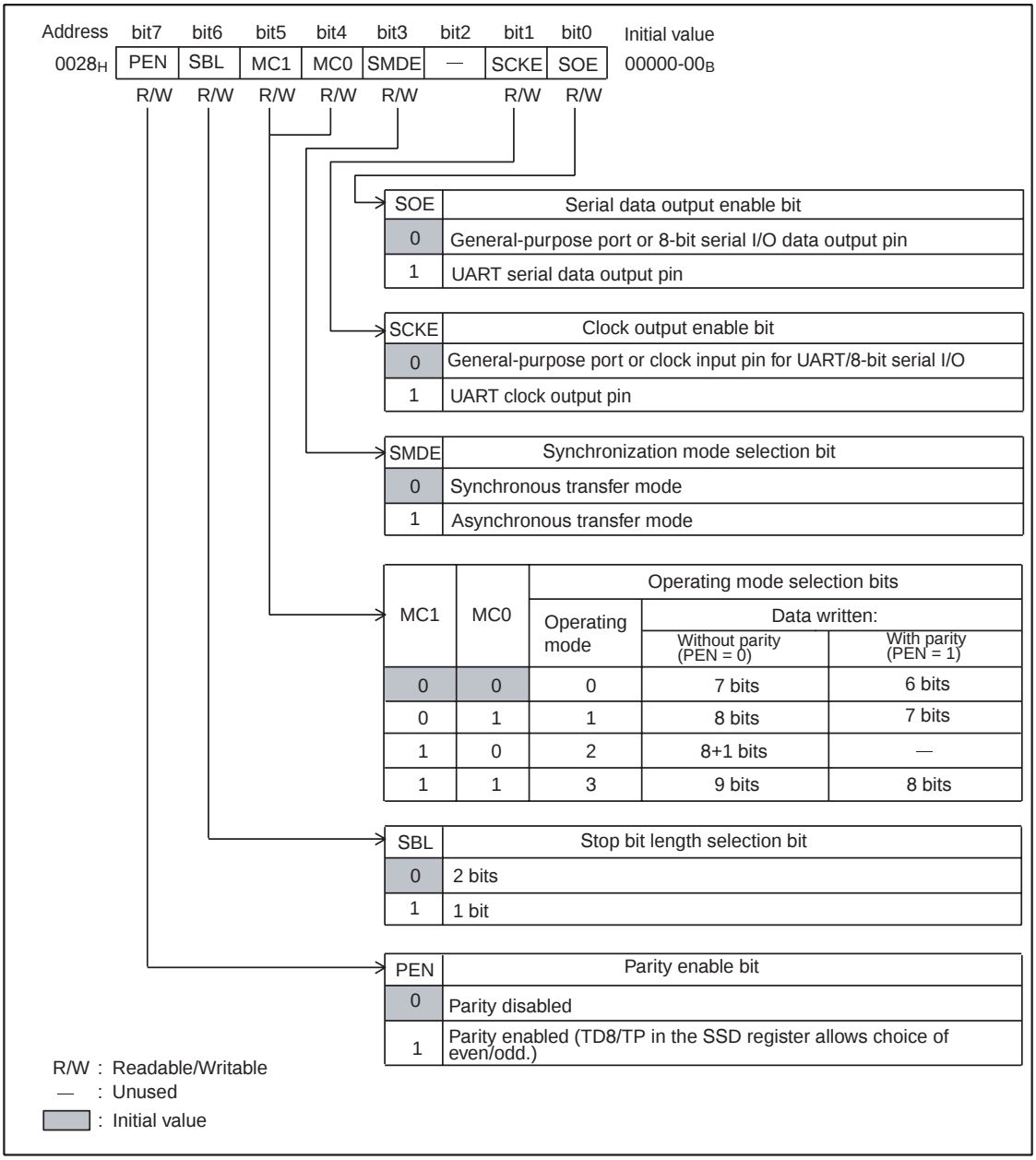


Table 13.4-1 Explanation of Functions of Each Bit in the Serial Mode Control Register (SMC)

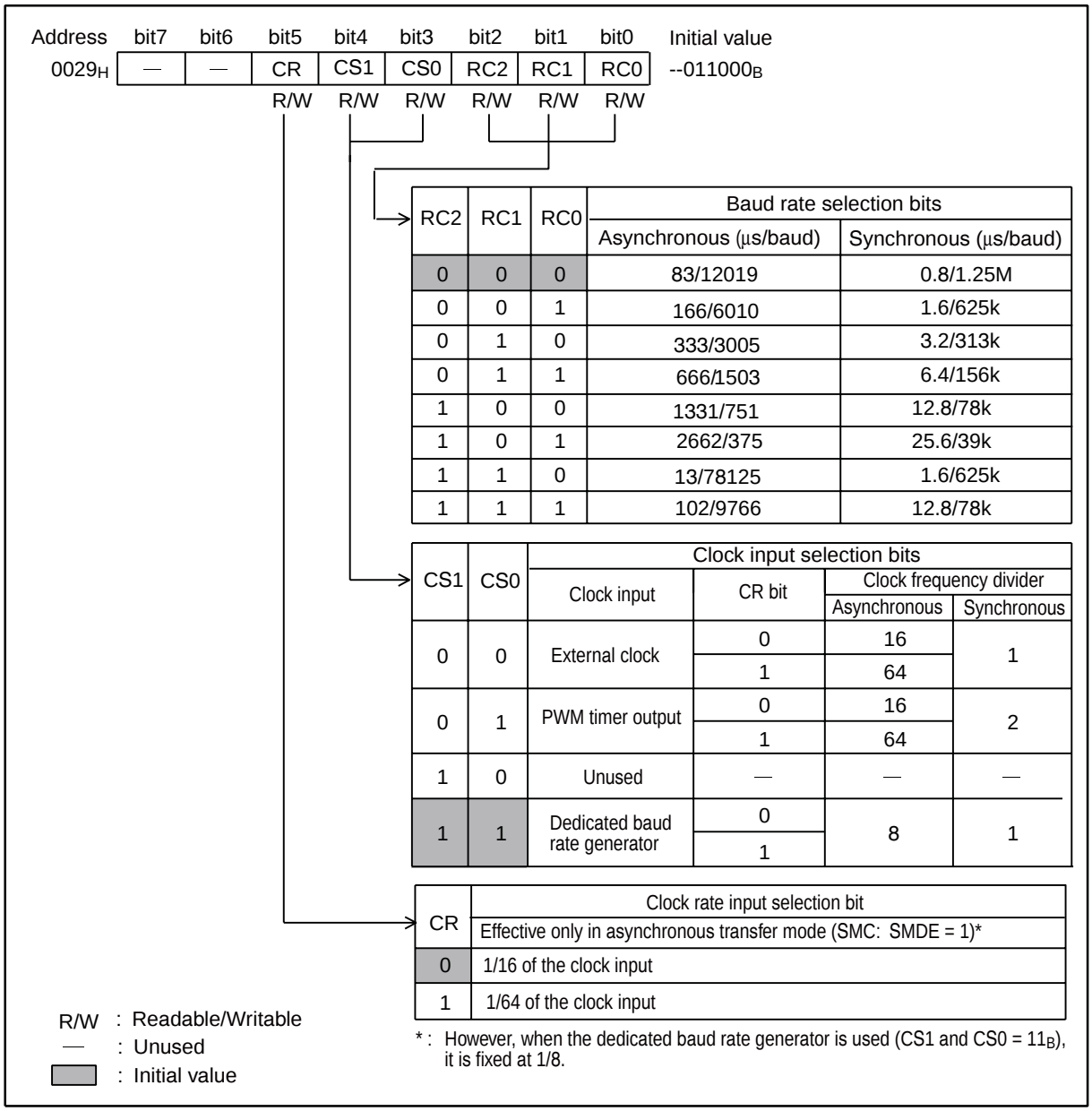
Bit name		Description
bit7	PEN: Parity enable bit	This bit selects whether the parity bit is to be added (at transmission) and detected (at reception) when serial data is input/output.
bit6	SBL: Stop bit length selection bit	This bit selects the stop bit length for data to be transmitted. **Note:** When data is received, only the first bit of stop bits is detected and the second and later bits are ignored.
bit5, bit4	MC1, MC0: Operating mode selection bits	• These bits specify operating mode (data length). • There are 7 types of data length selectable in combination with a parity bit.
bit3	SMDE: Synchronization mode selection bit	• This bit specifies synchronous transfer or asynchronous transfer mode. • When this bit is "0", synchronous transfer mode is set. When this is "1", asynchronous transfer mode is set.
bit2	Unused bit	• The value read out from this bit is undefined. • Writing a value into this bit does not affect any operations.
bit1	SCKE: Clock output enable bit	• This bit controls I/O of the serial clock. • When this bit is "0", P30/UCK/SCK pin functions as the serial clock input pin. When this bit is "1", it functions as the serial clock output pin. **Notes:** • When the UCK pin functions as the serial clock input pin (SCKE = 0), set the P30/UCK/SCK pin as the input port. Also, select the external clock using the clock input selection bit (SRC: CS1 and CS0 = 00_B). • When the UCK pin is set as the serial clock output pin (SCKE = 1), select a clock other than the external clock (SRC: CS1 and CS0 must not be 00_B). **Note:** When the UCK pin is specified as the serial clock output (SCKE = 1), it functions as the UCK output pin regardless of the state of the general-purpose port (P30).
bit0	SOE: Serial data output enable bit	When this bit is "0", the P31/UO/SO pin functions as a general-purpose port (P31). When this bit is "1", it functions as the serial data output pin (UO). **Note:** When serial data output is enabled (SOE = 1), the pin functions as the UO pin regardless of the state of the general-purpose port (P31).

13.4.2 Serial Rate Control Register (SRC)

The serial rate control register (SRC) controls the data transfer rate (baud rate) in asynchronous transfer mode. The SRC selects the input clock and sets the transfer rate for the dedicated baud rate generator.

■ Serial Rate Control Register (SRC)

Figure 13.4-3 Serial Rate Control Register (SRC)



RC2	RC1	RC0	Baud rate selection bits	
			Asynchronous (µs/baud)	Synchronous (µs/baud)
0	0	0	83/12019	0.8/1.25M
0	0	1	166/6010	1.6/625k
0	1	0	333/3005	3.2/313k
0	1	1	666/1503	6.4/156k
1	0	0	1331/751	12.8/78k
1	0	1	2662/375	25.6/39k
1	1	0	13/78125	1.6/625k
1	1	1	102/9766	12.8/78k

CS1	CS0	Clock input selection bits			
		Clock input	CR bit	Clock frequency divider	
				Asynchronous	Synchronous
0	0	External clock	0	16	1
			1	64	
0	1	PWM timer output	0	16	2
			1	64	
1	0	Unused	—	—	—
1	1	Dedicated baud rate generator	0	8	1
			1		

CR	Clock rate input selection bit
	Effective only in asynchronous transfer mode (SMC: SMDE = 1)*
0	1/16 of the clock input
1	1/64 of the clock input

* : However, when the dedicated baud rate generator is used (CS1 and CS0 = 11_B), it is fixed at 1/8.

Table 13.4-2 Explanation of Functions of Each Bit in the Serial Rate Control Register (SRC)

Bit name		Description
bit7, bit6	Unused bits	• The values read out from these bits are undefined. • Writing values to these bits does not affect any operations.
bit5	CR: Clock rate input selection bit	• This bit selects the clock rate in asynchronous transfer mode. However, when the dedicated baud rate generator is used (CS1 and CS0 = 11_B), it is fixed at 1/8 regardless of the value in the CR bit. Specifying an external clock or 8-bit PWM timer output as the clock input, the baud rate is set to 1/16 or 1/64 of the corresponding clock frequency, depending on the CR value. • This bit is not significant in synchronous transfer mode.
bit4, bit3	CS1,CS0: Clock input selection bits	• These bits select the clock input. • The clock input can be an external clock (UCK pin), 8-bit PWM timer, or dedicated baud rate generator.
bit2 to bit0	RC2,RC1,RC0: Baud rate selection bits	• There are 8 types of baud rate in asynchronous transfer mode and 6 types of baud rate in synchronous transfer mode: 14 types of baud rate are selectable in total. • These bits are effective only when the dedicated baud rate generator is used for the clock input. These bits are not significant when an external clock or 8-bit PWM timer output is used.

13.4.3 Serial Status and Data Register (SSD)

The serial status and data register (SSD) controls data transmission/reception of UART and status in an error, enables/disables interrupts, and specifies and checks settings for parity or bit-8 transmitting data.

■ Serial Status and Data Register (SSD)

Figure 13.4-4 Serial Status and Data Register (SSD)

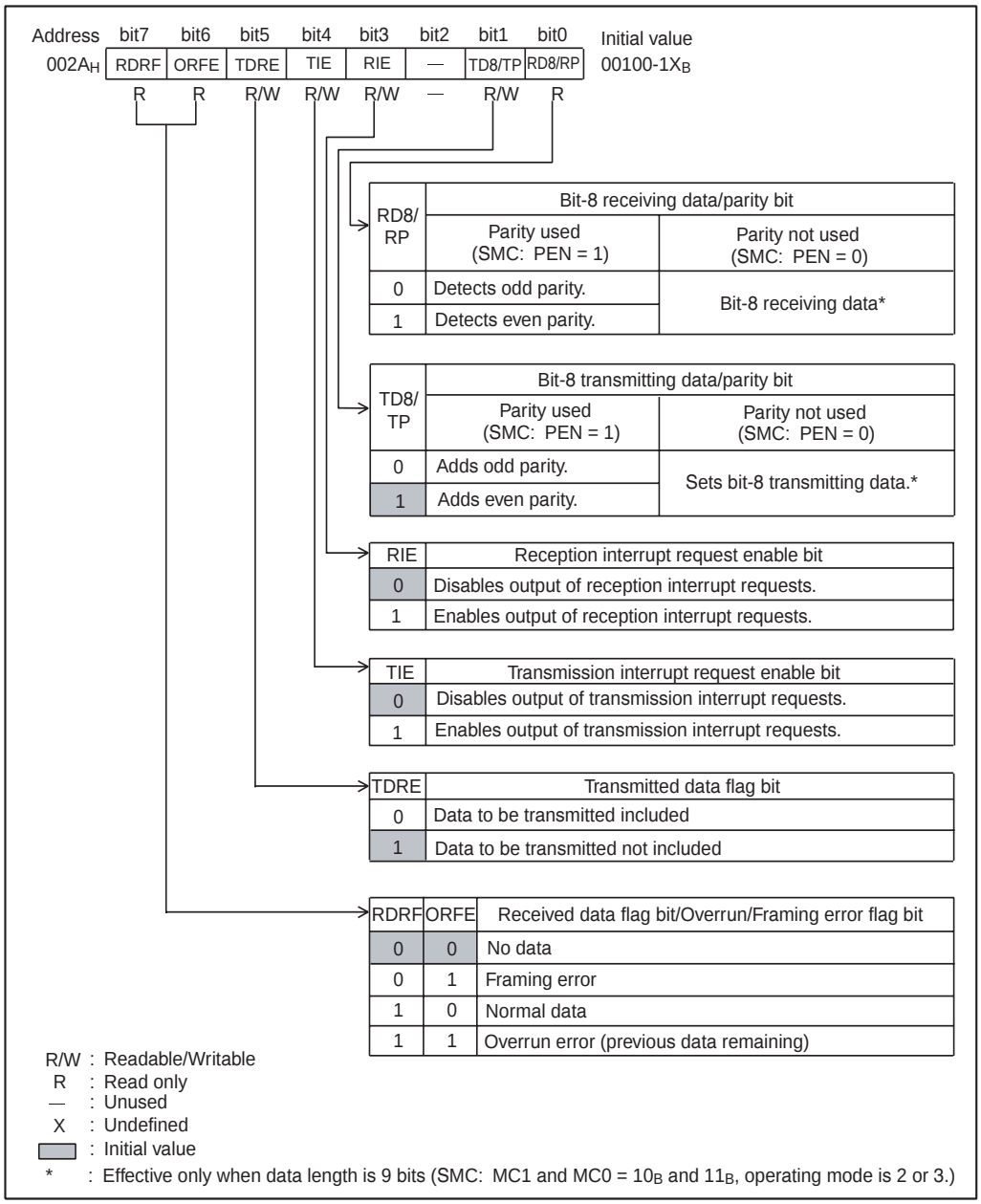


Table 13.4-3 Explanation of Functions of Each Bit in the Serial Status and Data Register (SSD)

Bit name		Description
bit7	RDRF: Received data flag bit	• This bit indicates the state of serial input data register (SIDR). • When this bit is "1", reading the SSD register, then the SIDR register clears RDRF. • When this bit and reception interrupt request enable bit (RIE) are "1", the reception interrupt request is output. • This bit is intended only for read. Writing a value into this bit has no significance and does not affect any operation.
bit6	ORFE: Overrun/Framing error flag bit	• This bit indicates that the overrun or framing error occurs. • When an error occurs (ORFE = 1), no data is transferred from the reception shift register to the SIDR register. Therefore, when an error occurs, the RDRF bit is not set. When this bit is "1", reading the SSD register then SIDR register clears the ORFE bit with "0". • When this bit and reception interrupt request enable bit (RIE) are "1", the reception interrupt request is output. • This bit is intended only for read. Writing a value into this bit has no significance and does not affect any operation.
bit5	TDRE: Transmitted data flag bit	• This bit indicates the state of serial output data register (SODR). • When this bit is "1", reading the SSD register and writing data into the SODR register output the data to the serial data output pin (UO). • When this bit and transmission interrupt request enable bit (TIE) are "1", the transmission interrupt request is output.
bit4	TIE: Transmission interrupt request enable bit	• This bit enables or disables the transmission interrupt request to the CPU. • When this bit and transmission data flag bit (TDRE) are "1", the transmission interrupt request is output.
bit3	RIE: Reception interrupt request enable bit	• This bit enables or disables the reception interrupt request to the CPU. • When this bit and reception data flag bit (RDRF) are "1", the reception interrupt request is output. • When this bit and error flag bit (ORFE) are "1", the reception interrupt request for an error is output.
bit2	Unused bit	• The value read out from this bit is undefined. • Writing a value into this bit does not affect any operations.
bit1	TD8/TP: Bit-8 transmitting data/ parity bit	• When parity is not used and operating mode is 2 or 3 (the length of data to be transmitted/received is 9), this bit is handled as bit8 in the SODR register. When operating mode is not 2 or 3 and parity is not used, this bit is not significant. • When parity is used, this bit selects even parity or odd parity for the transmitted data.
bit0	RD8/RP: Bit-8 receiving data/ parity bit	• When parity is not used and operating mode is 2 or 3 (the length of data to be transmitted/received is 9), this bit is handled as bit8 in the SIDR register. When operating mode is not 2 or 3 and parity is not used, this bit is not significant. • When parity is used, this bit indicates the parity of received data.

■ Receiving Status

Figure 13.4-5 shows the states (receiving status) of serial input data obtained from the received data flag bit (RDRF) and error flag bit (ORFE).

Figure 13.4-5 Receiving Status

RDRF	ORFE	Received data flag bit/Overrun/Framing error flag bit
0	0	No data
0	1	Framing error
1	0	Normal data
1	1	Overrun error (previous data remaining)

: Initial value

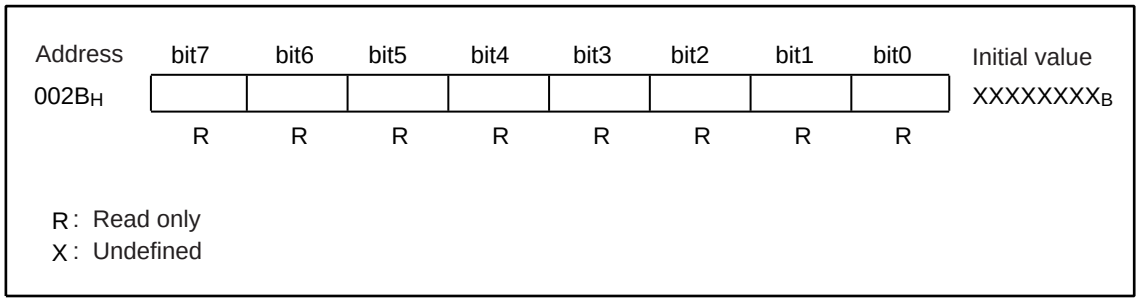
13.4.4 Serial Input Data Register (SIDR)

The serial input data register (SIDR) is for inputting (receiving) serial data.

■ Serial Input Data Register (SIDR)

Figure 13.4-6 shows the configuration of the serial input data register bits.

Figure 13.4-6 Serial Input Data Register (SIDR)



The SIDR stores received data. The serial data input pin (UI pin) receives serial data signals, the shift register converts them, then this register stores them.

- When operating mode is 0, 1, or 3

  For both the RDRF (Received data flag bit) and ORFE (Overrun/framing error flag bit), these flags go on and an interrupt request to the CPU is generated when data is fully transmitted or received, then the stop bit at the end is detected. When the RDRF is active, the data received is transmitted to the SIDR.

  When the received data is correctly stored in this register, "1" is set for the received data flag bit (RDRF). If the reception interrupt request is allowed, the reception interrupt is generated. When the RDRF bit has been checked in interrupt processing or the program and the received data has been stored into this register, read the contents in this register after reading the SSD register, then clear the RDRF flag.

- When operating mode is 2

  For both RDRF and ORFE, these flags go on when data is fully transmitted or received with the final data bit (D8) set to "1" and the stop bit at the end is detected. However, when the framing error occurs, the flag goes on regardless of the final data bit. An interrupt request to the CPU is generated when the flag goes on and the interrupt request is allowed.

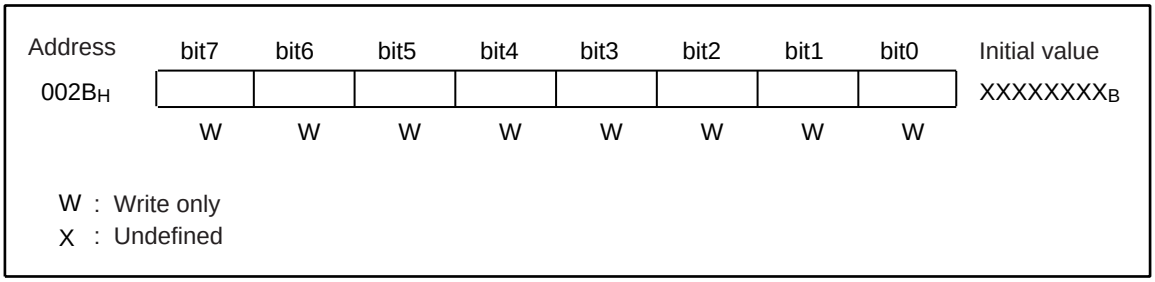
13.4.5 Serial Output Data Register (SODR)

The serial output data register (SODR) sends out (transmits) serial data.

■ Serial Output Data Register (SODR)

Figure 13.4-7 shows the configuration of the serial output data register bits.

Figure 13.4-7 Serial Output Data Register (SODR)



When transmission is enabled, writing data to be transmitted into this register after reading the SSD register sends the data to be transmitted to the transmission shift register, converts it into the serial format, then outputs it from the serial data output pin (UO pin).

When the transmitted data is written into the SODR register, the transmitted data flag bit is cleared with "0". After the transmitted data is sent to the transmission shift register, the transmitted data flag bit is set to "1", the data transmitted next then becomes writable. At this time, if the transmission interrupt request is enabled, an interrupt is generated. Write the data transmitted next when a transmission interrupt occurs or while the transmitted data flag bit is "1".

13.4.6 Clock Divider Selection Register (UPC)

The clock divider selection register is used to generate the UART reference clock by dividing the oscillation frequency. It also enables/disables operation of the prescaler for creating the reference clock.

■ Clock Divider Selection Register (UPC)

Figure 13.4-8 Clock Divider Selection Register (UPC)

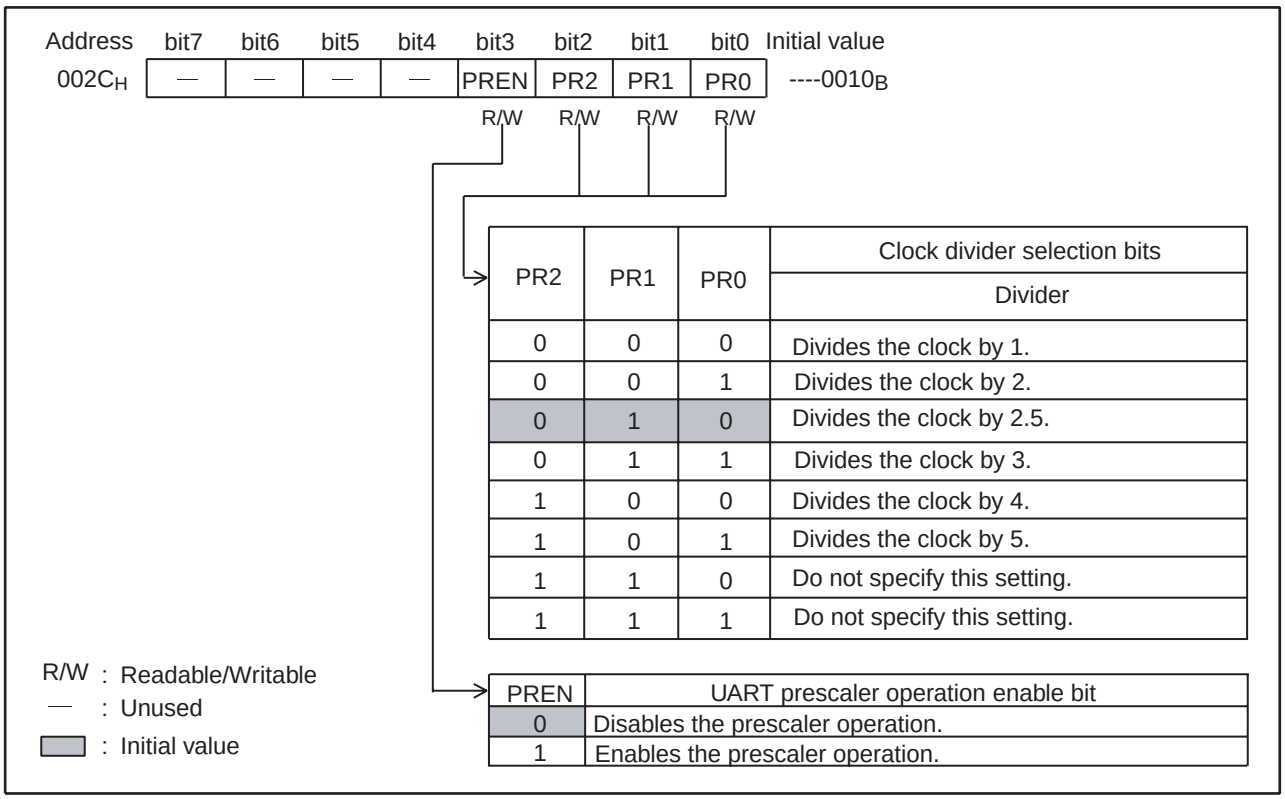


Table 13.4-4 Explanation of Functions of Each Bit in the Clock Divider Selection Register (UPC)

Bit name		Description
bit7 to bit4	Unused bits	• The values read out from these bits are undefined. • Writing values into these bits does not affect any operation.
bit3	PREN: UART prescaler operation enable bit	• Enables/disables operation of the prescaler that creates the UART reference clock by dividing the oscillation frequency. • When this bit is "1", the UART prescaler supplies the reference clock that corresponds to the frequency selected using the oscillation frequency selection bit to the baud rate generator. When this bit is "0", the prescaler does not operate, thus the internal baud rate generator cannot be used for data transfer or data receive purposes.
bit2 to bit0	PR2, PR1, PR0: Clock divider selection bits	• The UART prescaler supplies to the internal baud rate generator, the reference clock that corresponds to the divider selected using these bits.

Note:

The inside of UART is initialized when it is in synchronization transfer mode, asynchronous transfer mode, external clock mode, or internal clock mode with the clock from the prescaler. Therefore, turn on the PREN bit (PREN = 1) to enable operation of the prescaler before using the UART functions.

13.4.7 Serial Switch Register (SSEL)

The serial switch register (SSEL) switches the P30/UCK/SCK, P31/UO/SO, and P32/UI/SI pins between UART and 8-bit serial I/O.

■ Serial Switch Register (SSEL)

Figure 13.4-9 Serial Switch Register (SSEL)

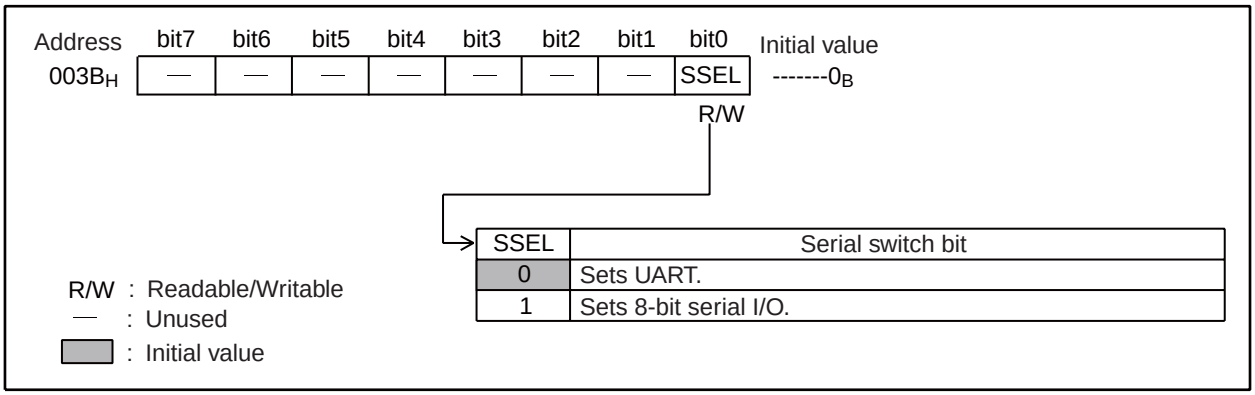
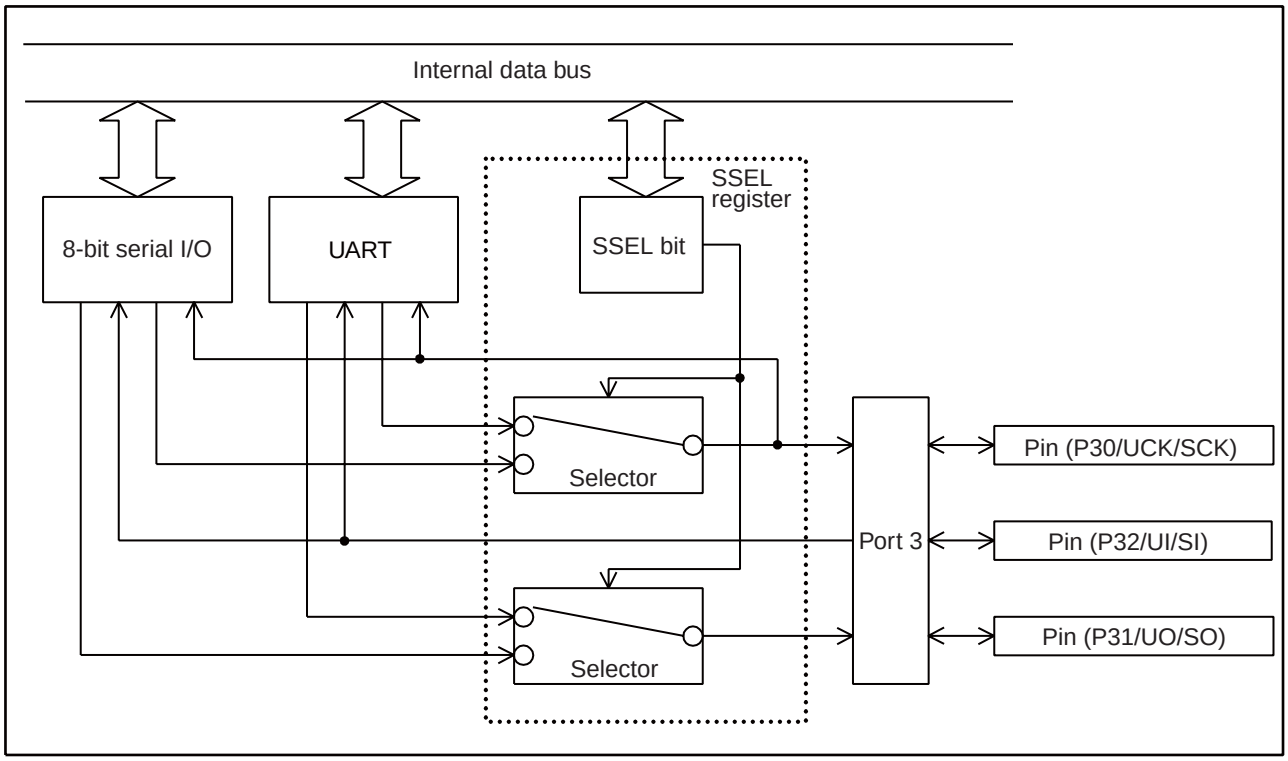


Table 13.4-5 Description of the Serial Switch Register (SSEL) Bits

Bit name		Description
bit7 to bit1	Unused bits	• The values read out from these bits are undefined. • Writing values to these bits does not affect any operations.
bit0	SSEL: Serial switch bit	• When this bit is "0", UART is used. • When this bit is "1", 8-bit serial I/O is used.

Figure 13.4-10 Block Diagram of Serial Switch Register



13.5 Interrupt of UART

UART supports the interrupt-related error flag bit (ORFE), received data flag bit (RDRF), and transmitted data flag bit (TDRE), and the following interrupt sources:

- **When received data is sent from the reception shift register to the serial input data register (SIDR). (Reception interrupt)**
- **When transmitted data is sent from the serial output data register (SODR) to the transmission shift register. (Transmission interrupt)**

■ Transmission Interrupt

When the SSD register is read and the output data is written into the SODR register, the data written into the SODR register is transferred to the internal transmission shift register. When next data becomes writable, the TDRE bit is set to "1", then an interrupt request to the CPU (IRQ5) is generated if the transmission interrupt is enabled (SSD: TIE = 1).

■ Reception Interrupt

● When operating mode is 0, 1, or 3

When data is correctly 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 a stop bit is detected. If the reception interrupt is enabled (SSD: RIE = 1), an interrupt request to the CPU (IRQ6) is generated.

● When operating mode is 2

For both RDRF and ORFE, data is received or transmitted with the final data bit (D8) set to "1", these flags go on when the stop bit at the end is detected. However, when the framing error occurs, the flag goes on regardless of the final data bit. An interrupt request to the CPU is generated when the flag goes on and the input data becomes "1".

■ UART Interrupt Related Registers and Vector Table Addresses

Table 13.5-1 provides the registers relating to the UART interrupts and vector table addresses. For details of the interrupt operation, see Section "3.4.2 Steps in the Interrupt Operation ".

Table 13.5-1 UART Interrupt Related Registers and Vector Table Addresses

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Bit		Upper digits	Lower digits
IRQ5	ILR2 ($007C_H$)	L51 (bit3)	L50 (bit2)	$FFF0_H$	$FFF1_H$
IRQ6	ILR2 ($007C_H$)	L61 (bit5)	L60 (bit4)	$FFEE_H$	$FFEF_H$

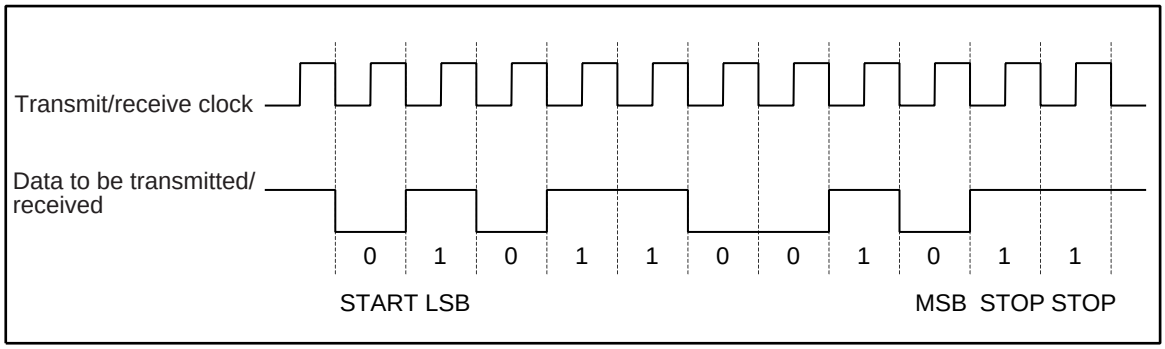
13.6 Operations of UART Functions

UART supports four types of operating mode. Mode 0, mode 1, and mode 3 are general serial transfer mode in which any data length can be selected in the range of 6 bits with parity used, to 9 bits without parity used. (See Table 13.1-1 .)

■ Transferred Data Format

UART can handle data in the NRZ (Non Return to Zero) format only. Data to be transferred always begins with the start bit ("L" level), specified number of data bits are transferred with LSB first, then data transfer is ended with the stop bit ("H" level). Figure 13.6-1 shows the relationship between the transmit/receive clock and transferred/received data when operating mode 0 without parity used, two stop bits, synchronous transfer mode, and transferred data 01001101_B (8 bits) are specified. Note that Figure 13.6-1 does not apply to the relationship between the serial clock and serial I/O signal in asynchronous transfer mode.

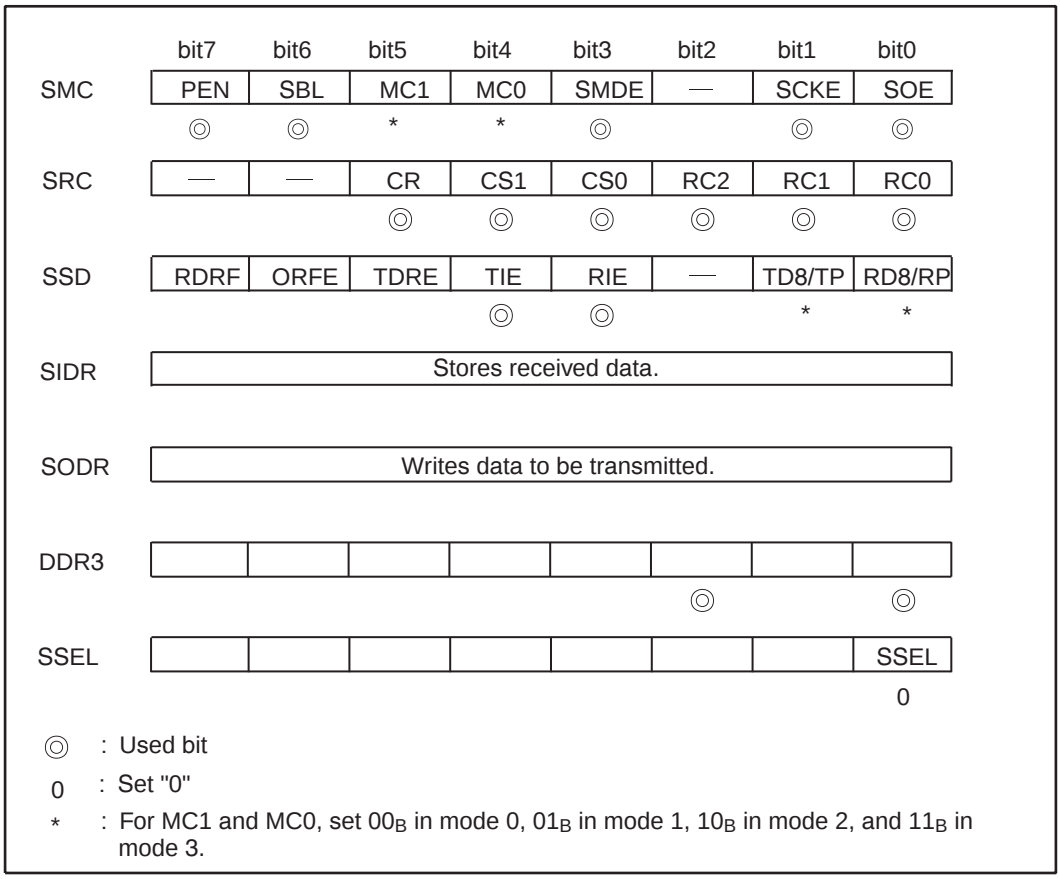
Figure 13.6-1 Transferred Data Format



■ Theory of Operation for Operating Mode 0, 1, 2, and 3

In operating mode 0, 1, 2, or 3, UART operates as a general serial communication function. Figure 13.6-2 shows the settings required in UART operating mode 0, 1, 2, or 3.

Figure 13.6-2 Operating Mode 0, 1, 2, or 3



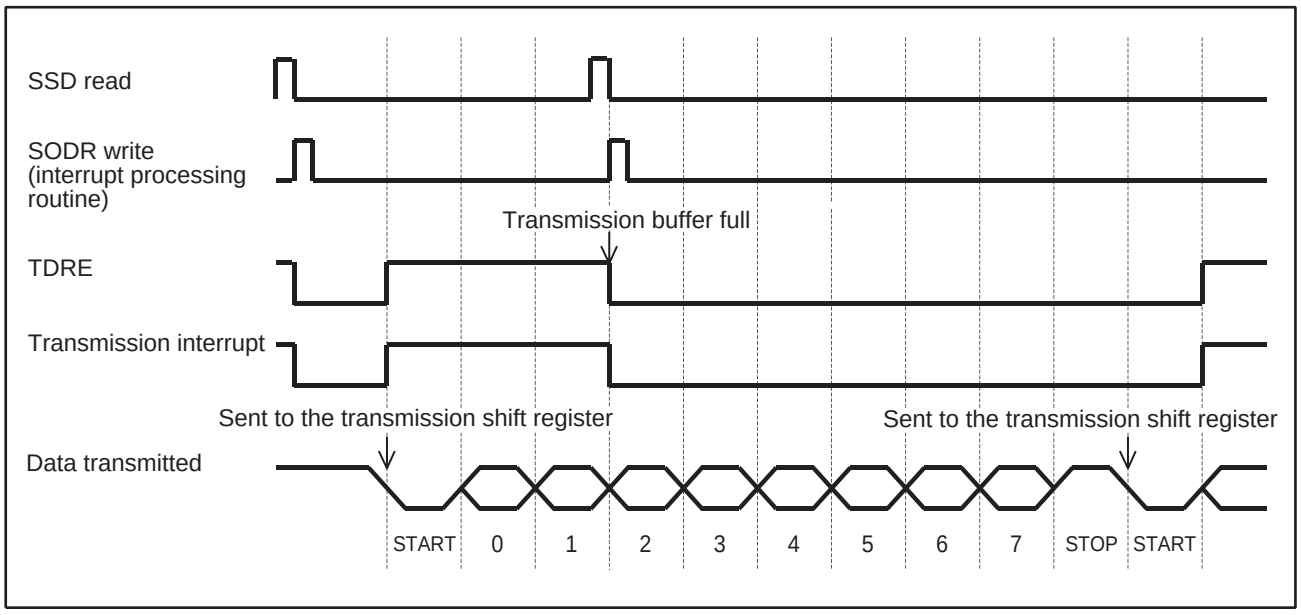
13.6.1 Transmission Operations (Operating Mode 0, 1, 2, and 3)

When writing data to be transmitted into the SODR register after reading the SSD register sends the data written into the SODR register to the transmission shift register, parallel-serial conversion then starts. The data converted is output at the serial data output pin from the lowest bit in sequence (with LSB first). When the next data becomes writable, "1" is set to the TDRE bit, then an interrupt request to the CPU is generated if the transmission interrupt is allowed (SSD: TIE = 1).

■ Transmission Operations in Operating Mode is 0, 1, 2, or 3

Figure 13.6-3 shows the transmission operations when operating mode is 1, parity is not used, and the number of stop bits is "1".

Figure 13.6-3 Transmission Operations in Operating Mode 0, 1, 2, or 3



13.6.2 Reception Operations (Operating Mode 0, 1, or 3)

When data is received at the serial data input pin, the internal reception shift register converts it from serial to parallel. If the data is correctly transmitted up to the stop bit(s), data in the internal shift register is transferred to the SIDR register, then "1" is set to the RDRF bit.

■ Reception Operations (Operating Mode 0, 1, or 3)

If an overrun error or framing error occurs, the received data is not transmitted to the SIDR register, but the ORFE bit is set to "1".

Either of the RDRF bit and ORFE bit goes on when the final stop bit is detected after data is fully received. If the reception interrupt is enabled (SSD: RIE = 1), an interrupt request to the CPU (IRQ6) is generated. When the RDRF bit goes on, the received data has been transmitted to the SIDR register.

In operating mode 2, when the RIE bit is "1", RDRF bit or ORFE bit is "1", and reception interrupt pin is "1", the mode 2 UART reception interrupt request is output to the CPU.

Note:

In operating mode 1, the parity bit is read as data in the 7th bit. Set up the program so that the 7th bit is not read.

Figure 13.6-4 , Figure 13.6-5 , and Figure 13.6-6 show the reception operations when parity is not used and the number of stop bits is "1" in operating mode 0, 1, or 3.

Figure 13.6-4 Reception Operations in Operating Mode 0, 1, or 3

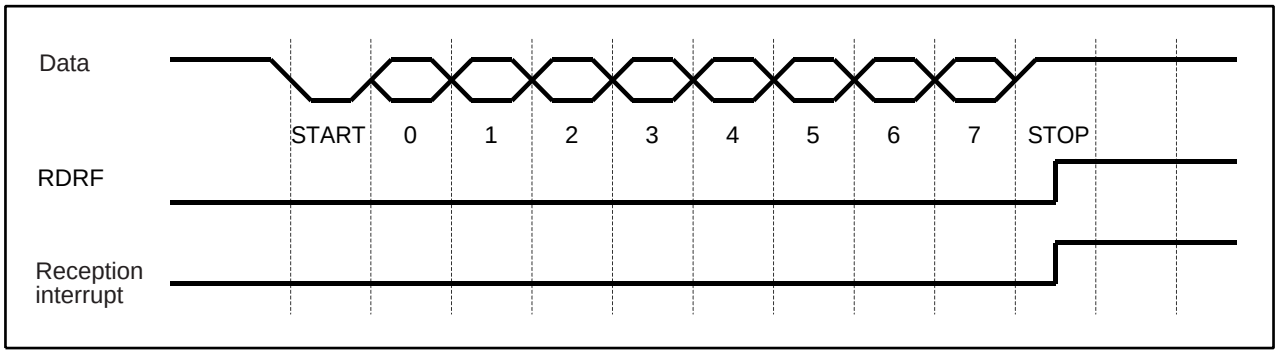


Figure 13.6-5 Operations in Operating Mode 0, 1, or 3 when the Overrun Error Occurs

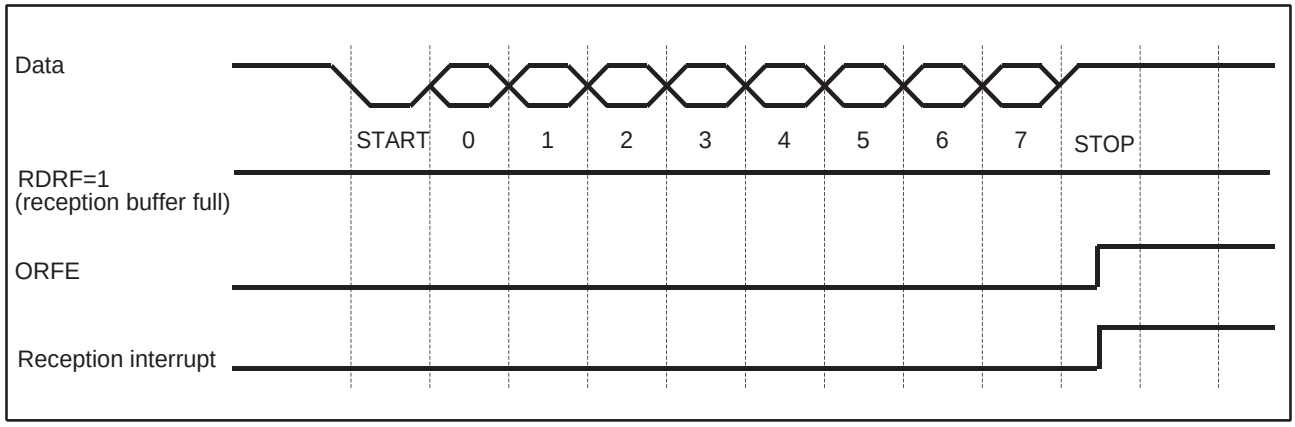
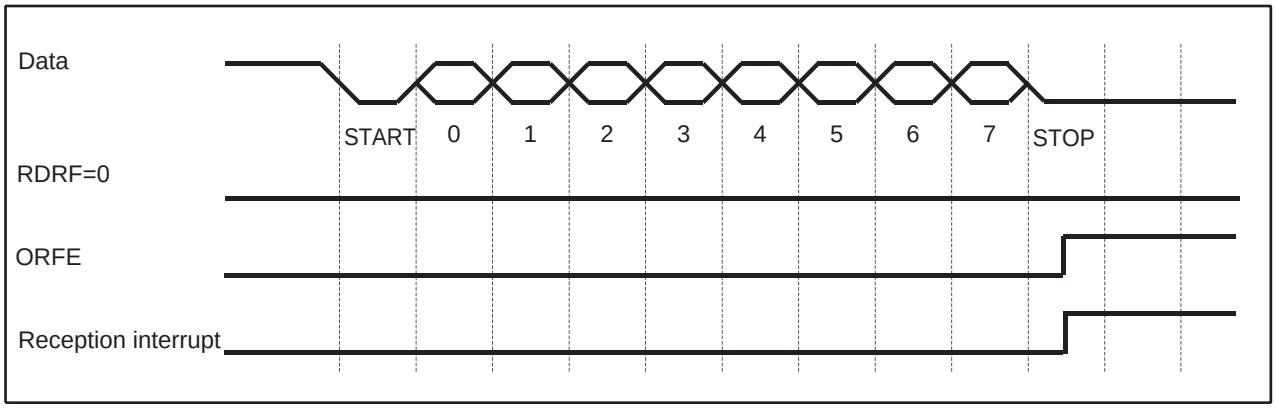


Figure 13.6-6 Operations in Operating Mode 0, 1, or 3 when the Framing Error Occurs



Note:

After initialization is cancelled due to a reset, time for 11 shift-clock cycles is required to initialize the internal controller. Therefore, be sure to enable the UART prescaler operation (PREN = 1) using the oscillation frequency register after a reset.

13.6.3 Reception Operations (Operating Mode 2 Only)

When data is received at the serial data input pin, the internal reception shift register converts it from serial to parallel. If the data is correctly transmitted up to the stop bit(s), data in the internal shift register is transferred to the SIDR register, then "1" is set to the RDRF bit.

■ Reception Operations (Operating Mode 2 Only)

If an overrun error or framing error occurs, the received data is not transmitted to the SIDR register, but the ORFE bit is set to "1".

For both RDRF and ORFE, data is fully received/transmitted with the final data bit (D8) set to "1", these flags go on when the stop bit at the end is detected. However, when the framing error occurs, the flag goes on regardless of the final data bit. An interrupt request to the CPU is generated when the flag goes on and interrupt request is enabled.

If the reception interrupt is enabled (SSD: RIE = 1), an interrupt request to the CPU (IRQ5) is generated. When the RDRF bit goes on, the received data is transmitted to the SIDR register.

Figure 13.6-7 to Figure 13.6-9 show the reception operations when parity is not used and the number of stop bits is "1" in operating mode 2.

Figure 13.6-7 Reception Operations in Operating Mode 2

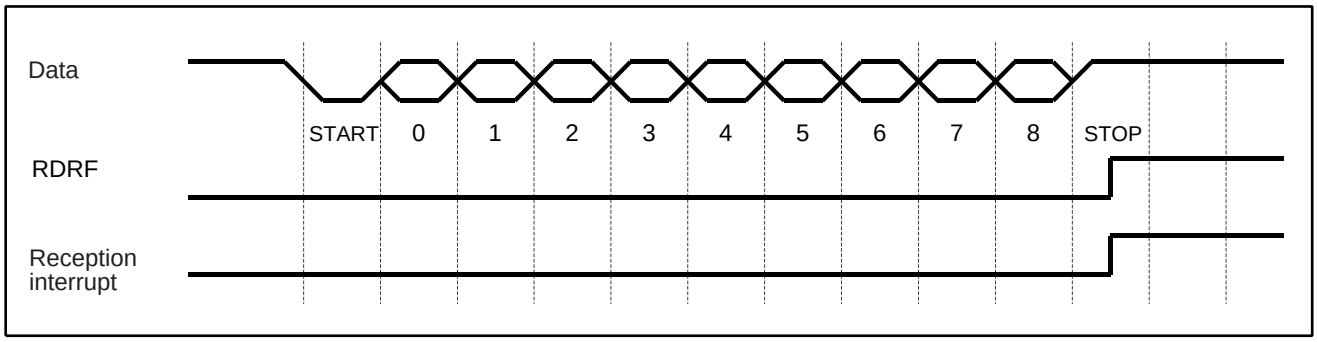


Figure 13.6-8 Operations in Operating Mode 2 when the Overrun Error Occurs

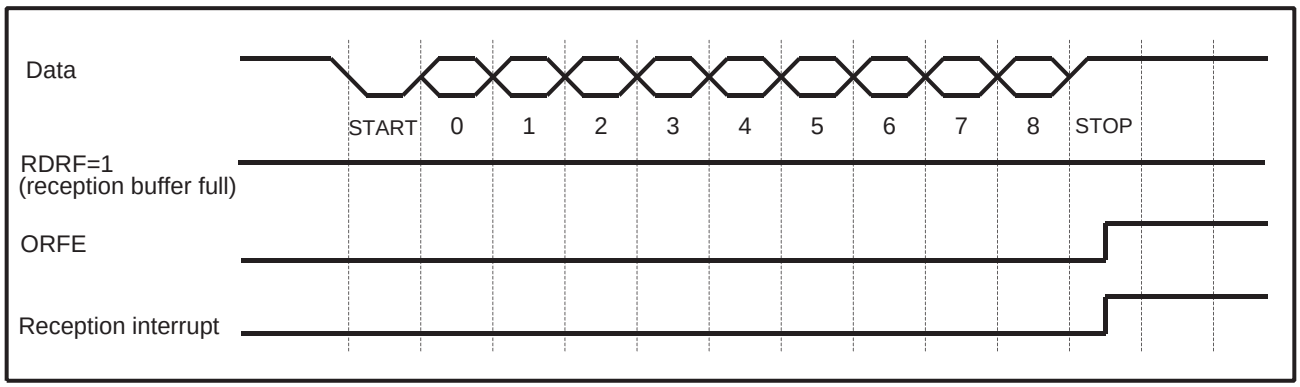
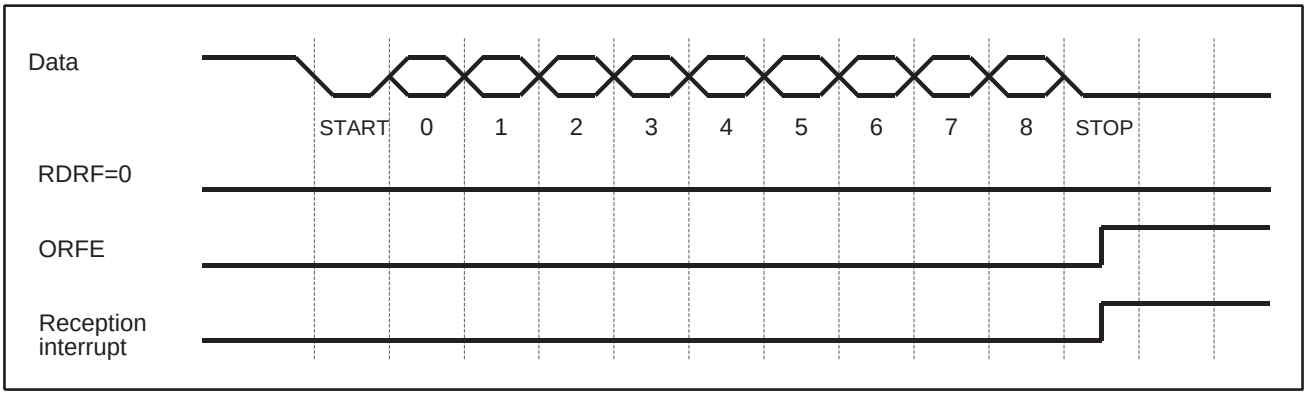


Figure 13.6-9 Operations in Operating Mode 2 when the Framing Error Occurs



Note:

After initialization is cancelled due to a reset, time for 11 shift-clock cycles is required to initialize the internal controller. Therefore, be sure to enable the UART prescaler operation (PREN = 1) using the oscillation frequency register after a reset.

13.7 Program Example for UART

This section provides program example for UART.

■ Program Example for UART

● Program specifications

- Serial data transfer is implemented using the UART communication functions.
- The P30/UCK/SCK, P31/UO/SO, and P32/UI/SI pins are used for communication.
- The transfer rate is set to 300 bps using the internal baud rate generator.
- 13_H is transmitted from the UO pin, and data is received by interrupts.
- The baud rate is the oscillation frequency (F_{CH} = 12.5 MHz) at the maximum gear speed (1 instruction cycle = $4/F_{CH}$). The clock divider is 2.5. (1/375 bps = 8320 t_{INST})

● Coding example

```
PDR3    EQU     000CH           ;  Port data register address
DDR3    EQU     000DH           ;  Port direction register address
SSEL    EQU     003BH           ;  Serial selection register address
SMC     EQU     0028H           ;  Serial mode control register address
SRC     EQU     0029H           ;  Serial rate control register address
SSD     EQU     002AH           ;  Serial status and data register address
SIDR    EQU     002BH           ;  Serial input data register address
SODR    EQU     002BH           ;  Serial output data register address
UPC     EQU     002CH           ;  Clock divider selection register address
ILR2    EQU     007CH           ;  Interrupt level setting register address
INT_V   DSEG    ABS             ;  [DATA SEGMENT]
        ORG     0FFEEH
IRQ6    DW      WARI2           ;  Reception interrupt vector setting
IRQ5    DW      WARI1           ;  Transmission interrupt vector setting
INT_V   ENDS
;--------------------Main program-----------------------------------------------------------------
        CSEG                    ;  [CODE SEGMENT]
                                ;  The stack pointer (SP) and related components have to be
                                   initialized.
          :
        CLRI                    ;  Disable interrupts.
        MOV     ILR2,#11101011B ;  Set an interrupt level (level 1).
        MOV     UPC,#11111010B  ;  Allow operation with the clock whose frequency is divided
                                   by 2.5.
        MOV     SSEL,#00000000B ;  Select UART.
        MOV     DDR3,#00000000B ;  Set the UI pin as the input pin.
        MOV     SMC,#01011011B  ;  Set non-parity, the number of stop bits 1, and operating
```

```
                                        mode 1. Set asynchronous mode, enable clock output
                                        and serial data output.
        MOV     SRC,#00011101B  ;  Select the dedicated baud rate generator, and set the baud
                                        rate 375 bps.
        MOV     SSD,#00001000B  ;  Disable the transmission interrupt request and enable the
                                        reception interrupt request.
        MOV     A,SSD           ;  Required before transmission
                                        (TDRE = 1 enables transmission)
        MOV     A,SIDR          ;  Clear error flags.
        MOV     SODR,#13H       ;  Write the data to be transmitted (13H).
        SETI                    ;  Enable instruction.
          :
;--------------------Interrupt processing routine-----------------------------------------------------------
WARI    PUSHW   A               ;  Save A and T.
        XCHW    A,T
        PUSHW   A
        MOV     A,SSD           ;  Read the data to be transmitted, then clears the input data
                                        flag.
        MOV     A,SIDR
          :
        User-defined process
          :
        POPW    A               ;  Restore A and T.
        XCHW    A,T
        POPW    A
        RETI
        ENDS
; ------------------------------------------------------------------------------------------------------------
        END
```

CHAPTER 14

8-BIT SERIAL I/O

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

14.1 Overview of 8-Bit Serial I/O

The 8-bit serial I/O has a function that serially transfers 8-bit data in synchronization with a shift clock. It can select one shift clock from three internal shift clocks and one external shift clock. It can also select LSB first or MSB first as the data shift direction.

■ Serial I/O Function

The 8-bit serial I/O function serially inputs and outputs 8-bit data in synchronization with a shift clock.

- Converts 8-bit parallel data to 8-bit serial data and outputs it. Also inputs 8-bit serial data, converts the data to 8-bit parallel data, and stores it.
- Can select one shift clock from three internal shift clocks and one external shift clock.
- Can control shift clock input/output and output internal shift clocks.
- Can select LSB first or MSB first as the data shift direction.

Table 14.1-1 Shift Clock Cycle and Transfer Rate

Shift clock	Clock cycle	Frequency (Hz)	Transfer rate (F_{CH}=12.5MHz, At maximum speed*)
Internal shift clock (output)	$2t_{INST}$	$1/(2t_{INST})$	1562.5 kbps
	$8t_{INST}$	$1/(8t_{INST})$	390.6 kbps
	$32t_{INST}$	$1/(32t_{INST})$	97.66 kbps
External shift clock (input)	$2t_{INST}$ or lower	$1/(2t_{INST})$ or lower	DC to 1562.5 kbps

F_{CH} : Oscillation frequency

t_{INST}: Instruction cycle

* : When the highest speed clock of a general mode is selected with the system clock control register (SYCC) (CS1 and CS0 bits of SYCC = 11_B, 1 instruction cycle = $4/F_{CH}$)

■ Serial Function Switching

The 8-bit serial I/O and UART cannot be used simultaneously because they use the same pin. For this reason, the serial function switching circuit must be used to switch the 8-bit serial I/O and UART. For more information on the serial function switching circuit, see Section "13.4.7 Serial Switch Register (SSEL) ".

Selecting the 8-bit serial I/O with this serial function switching circuit enables P30/UCK/SCK to be used as the serial clock I/O pin (SCK) of the serial I/O, and P31/UO/SO to be used as the data output pin (SO). This selection also enables P32/UI/SI to be used as the data input pin (SI).

Note:

This chapter describes pin function switching and the register function, etc., on the assumption that the 8-bit serial I/O is selected with the serial function switching circuit.

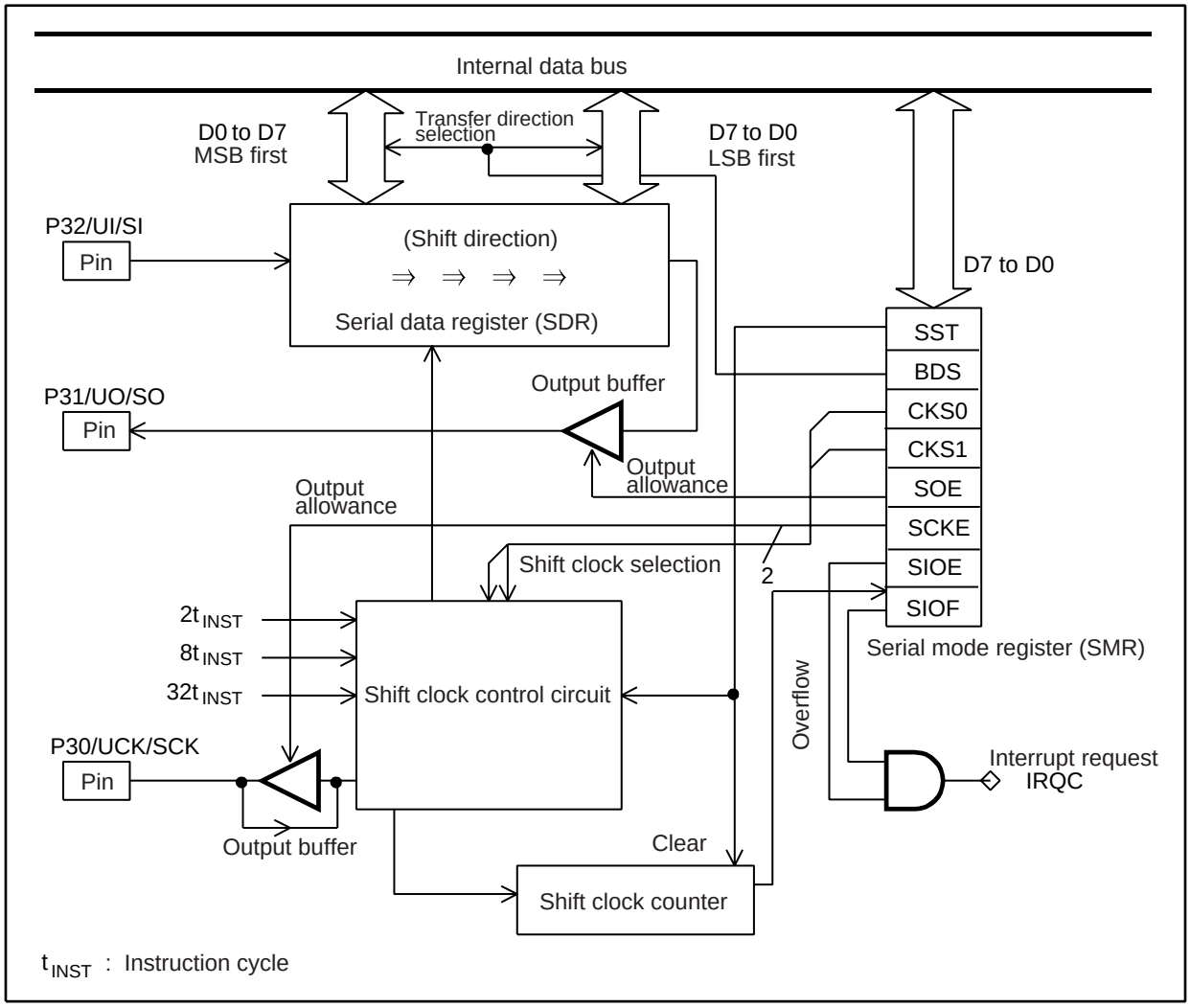
14.2 Configuration of 8-Bit Serial I/O

Each 8-bit serial I/O channel 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 14.2-1 Block Diagram of 8-bit Serial I/O



- Shift clock control circuit

  As a shift clock of the shift clock control circuit, one of three internal clocks and one external clock is selected.

  Selecting an internal clock enables the shift clock to be output to the SCK pin. Selecting an external clock enables the clock to be input from the SCK pin to act as the shift clock. The shift clock control circuit shifts the SDR in accordance with this shift clock and outputs the shifted-out value to the SO pin. It also captures the data input from the SI pin while shifting it to the SDR.

- Shift clock counter

  The shift clock counter counts the number of times the SDR was shifted using the shift clock. When 8-bit shift is completed, the counter overflows.

  When the counter overflows, the serial I/O transfer start bit of the SMR (SST = 0) is cleared and the interrupt request flag bit (SIOF = 1) is set. When serial transfer stops (SST = 0), the counter stops its count. It is cleared when serial transfer is started (SST = 1).

- Serial data register (SDR)

  The SDR retains transfer data. The data written to the SDR is converted to serial data and output. Serial input is converted to parallel data and stored.

- Serial mode register (SMR)

  The SMR is a serial I/O control register. It is used to allow and prohibit serial I/O operation, select shift clocks, and set a transfer (shift) direction. It is also used to control interrupts and check interrupt states.

- 8-bit serial I/O interrupt

  - IRQC: If the interrupt request output is allowed (SMR: SIOE = 1) when the I/O function of the 8-bit serial I/O inputs or outputs 8-bit serial data, a interrupt request (IRQC) is generated.

14.3 Pins of 8-Bit Serial I/O

8-bit serial I/O pins include P32/UI/SI, P31/UO/SO, and P30/UCK/SCK pins.

■ Pins of 8-bit Serial I/O

- P32/UI/SI pin

  The P32/UI/SI pin functions as the general-purpose I/O port (P32). It also functions as the serial data input pin (SI) of the 8-bit serial I/O or as the serial data input pin (UI) of the UART.

  When using the P32/SI pin as the SI pin, set the P32/UI/SI pin to "input port" with the port direction register (DDR3: bit2 = 0).

- P31/UO/SO pin

  The P31/UO/SO pin functions as the general-purpose I/O port (P31). It also functions as the serial data output pin (SO) function of the 8-bit serial I/O or as the serial data output pin (UO) function of the UART.

  When the serial data output is allowed (SMR: SOE = 1), the P31/UO/SO pin automatically becomes an output pin irrespective of the values in the port direction register (bit1 of DDR3) and functions as the SO pin.

- P30/UCK/SCK pin

  The P30/UCK/SCK pin functions as the general-purpose I/O port (P30). It also functions as the shift clock I/O pin (SCK) of the 8-bit serial I/O or as the shift clock I/O pin (UCK) of the UART.

  When using the P30/UCK/SCK pin as the shift clock input pin

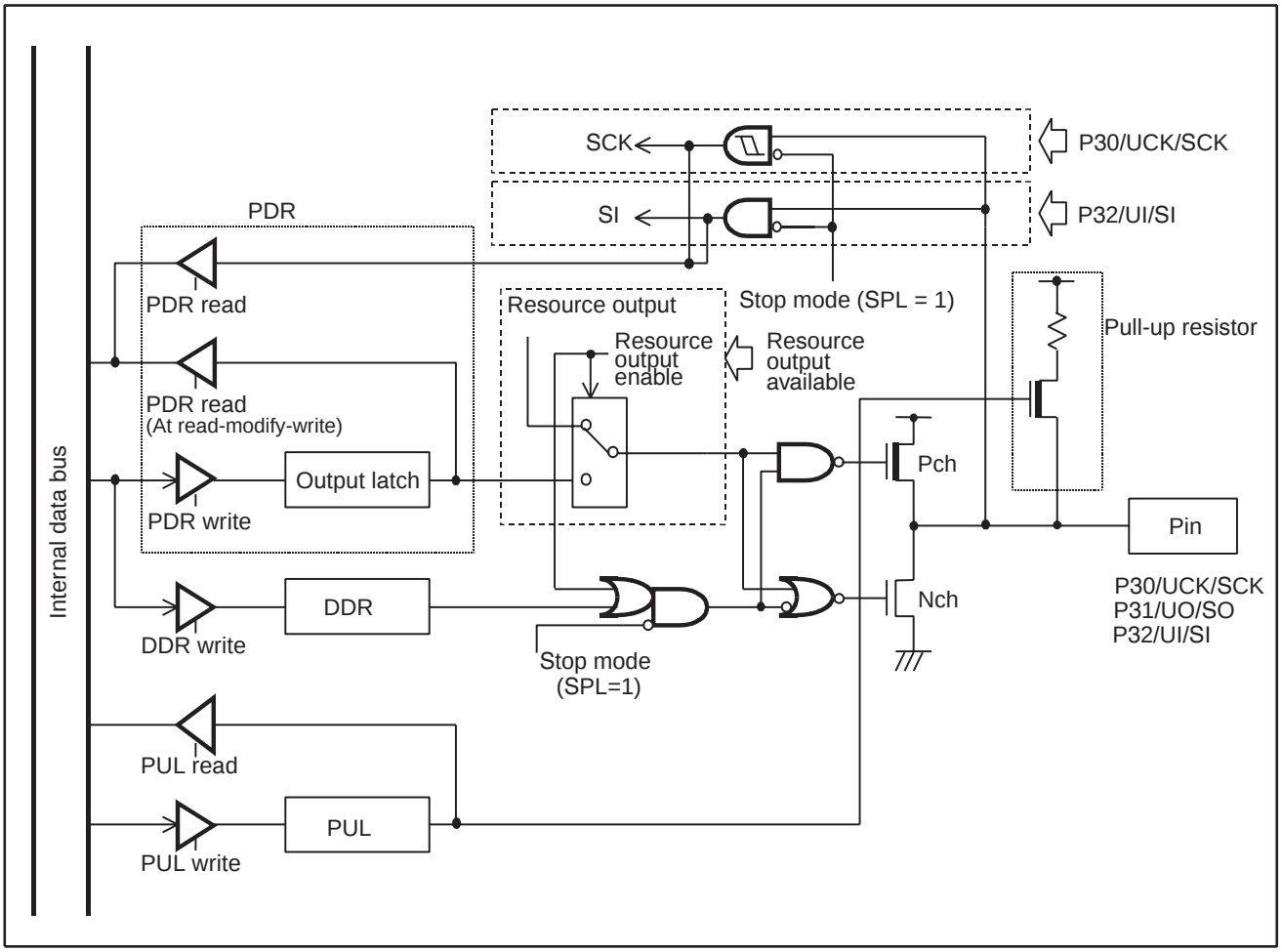
  When using the SCK pin as the shift clock input pin, set it to "input port" with the port direction register (DDR3: bit3 = 0) and prohibit shift clock output (SMR: SCKE = 0). In this case, be sure to select the external shift clock (SMR: CKS1, CKS0 = 11_B).

  When using the P30/UCK/SCK pin as the shift clock output pin

  When the shift clock output is allowed (SMR: SCKE = 1), the P30/UCK/SCK pin automatically becomes an output pin irrespective of the values in the port direction register (DDR3: bit0) and functions as the SCK output pin. In this case, be sure to select an internal shift clock (when SMR: CKS1, CKS0 are not 11_B).

■ Block Diagram for 8-bit Serial I/O Pins

Figure 14.3-1 Block Diagram for 8-bit Serial I/O Pins



Note:

When "pull-up resistor available" is selected in the pull-up setting register, the pin state in stop mode (SPL = 1) becomes high (pull-up state), not Hi-Z. During the reset, however, pull-up becomes invalid and the pin state becomes Hi-Z.

14.4 Registers of 8-Bit Serial I/O

Figure 14.4-1 shows 8-bit serial I/O registers.

■ Registers of 8-bit Serial I/O

Figure 14.4-1 8-bit Serial I/O Registers

SMR (serial mode register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0039_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
$003A_H$									$XXXXXXXX_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable and Writable
X : Undefined

Note:

When using a bit manipulation instruction, make sure that the SST bit is "0".

14.4.1 Serial Mode Register (SMR)

The serial mode register (SMR) is used to allow and prohibit 8-bit serial I/O operation, select a shift clock, set a transfer direction, control interrupts, and check interrupt states.

■ Serial Mode Register (SMR)

Figure 14.4-2 Serial Mode Register (SMR)

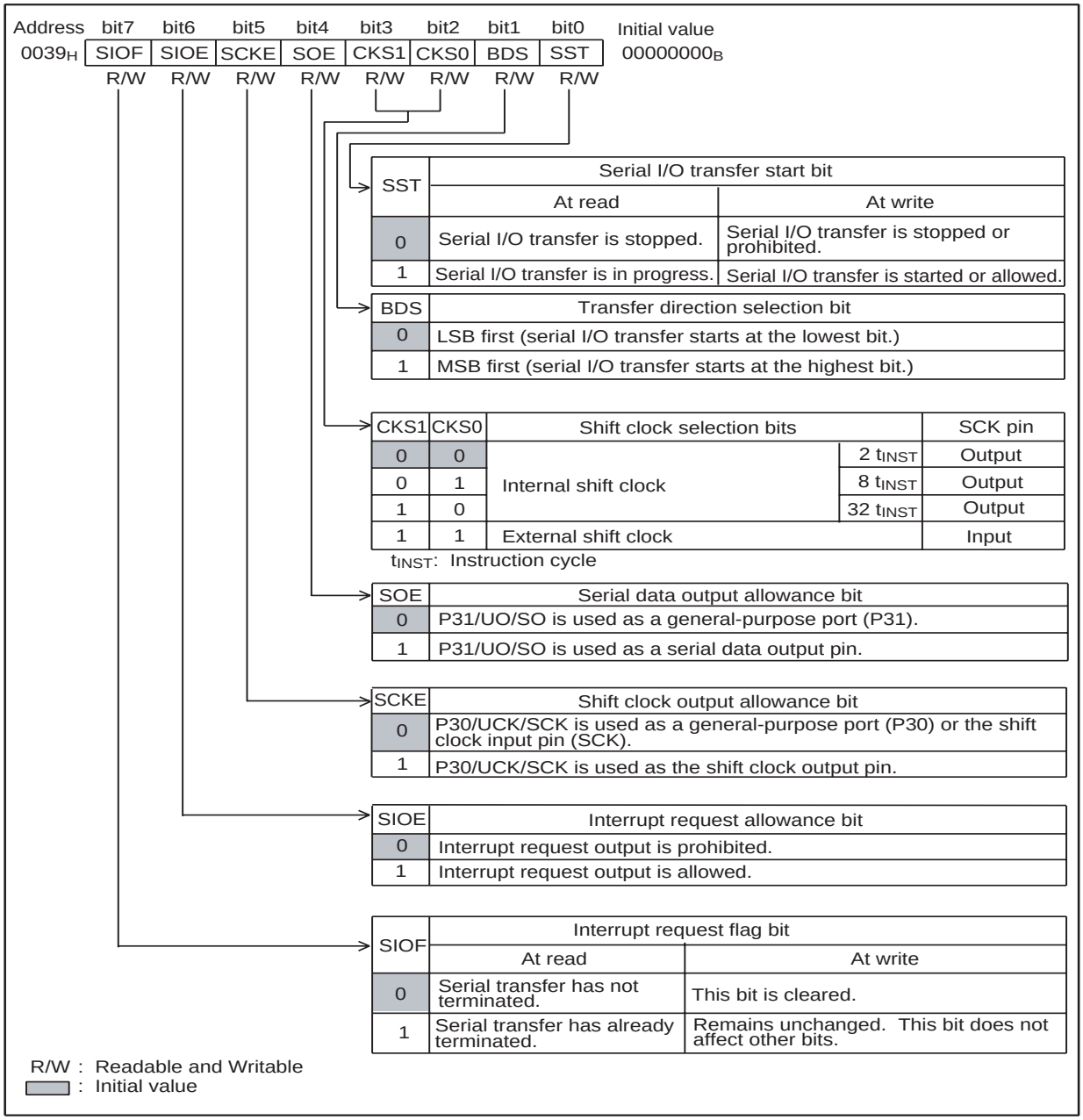


Table 14.4-1 Explanation of Functions of Each Bit in Serial Mode Register (SMR) (1/2)

Bit name		Function
bit7	SIOF: Interrupt request flag bit	• When 8-bit serial data is input or output during serial I/O operation, this bit is set to "1". When this bit and the interrupt request allowance bit (SIOE) are "1", an interrupt request is output. • Setting this bit to "0" clears it, while setting it to "1" does not affect this bit or implement any changes.
bit6	SIOE: Interrupt request allowance bit	This bit is used to allow and prohibit interrupt request output to the CPU. When this bit and the interrupt request allowance bit (SIOF) are "1", an interrupt request is output.
bit5	SCKE: Shift clock output allowance bit	• This bit is used to control shift clock I/O. • When this bit is "0", the P30/UCK/SCK pin functions as the shift clock input pin. When "1", it functions as the shift clock output pin. **Notes:** • To use the P30/UCK/SCK pin as the shift clock input pin, it must be set as an input port. Also select the external shift clock with the shift clock selection bits (Set the CKS1 and CKS0 bits to 11_B). • For shift clock output (SCKE bit = 1), select an internal shift clock (do not set the CKS1 and CKS0 bits to 11_B). **Notes:** • When shift clock output is allowed (when this bit is "1"), the P30/UCK/SCK pin functions as the UCK/SCK output pin irrespective of the general-purpose port (P30) state. • When using the P30/UCK/SCK pin as a general-purpose port (P30), set its pin as the shift clock input pin (set this bit to "0").
bit4	SOE: Serial data output allowance bit	When this bit is "0" the P31/UO/SO pin functions as a general-purpose port (P31). When "1", the P31/UO/SO pin functions as the serial data output pin (UO/SO). **Note:** When serial data output is allowed (when this bit is set to "1"), the P31/UO/SO pin functions as the UO/SO pin irrespective of the general-purpose port (P31) state.
bit3, bit2	CKS1, CKS0: Shift clock selection bits	• These bits are used to select three internal shift clocks or one external shift clock. • When these bits are not 11_B, an internal shift clock is selected. When the shift clock output allowance bit (SCKE) is "1", a shift clock is output from the UCK/SCK pin. • When these bits are 11_B, the external clock is selected. When the P30/UCK/SCK pin is set as the shift clock input pin, a shift clock is input from the UCK/SCK pin (when the SCKE bit and bit0 of the DDR3 are "0").

Table 14.4-1 Explanation of Functions of Each Bit in Serial Mode Register (SMR) (2/2)

Bit name		Function
bit1	BDS: Transfer direction selection bit	This bit is used to select whether to transfer serial data, starting at the lowest bit (LSB first, BDS = 0) or the highest bit (MSB first, BDS = 1). When this bit is set to "0", serial data is transferred, starting at the lowest bit. When it is set to "1", serial data is transferred, starting at the highest bit. **Note:** If this bit is rewritten after serial data has been written to the SDR to replace higher data with lower data, the data in the SDR becomes invalid.
bit0	SST: Serial I/O transfer start bit	• This bit is used to control serial I/O transfer start and allowance. It can also be used to judge whether serial I/O transfer terminated. • If this bit is set to "1" when the internal shift clock is selected (when the CKS1 and CKS0 bits are not 11_B), the shift clock counter is cleared and serial I/O transfer is started. • If this bit is set to "1" when the external shift clock is selected (when the CKS1 and CKS0 bits are 11_B), serial I/O transfer is allowed and the shift clock counter is cleared. The transfer side enters the external shift clock input wait state. • When serial I/O transfer terminates, this bit is set (cleared) to "0" and the SIOF bit is set to "1". • If this bit is set to "0" during serial I/O transfer (SST = 1), serial I/O transfer is suspended. When serial I/O transfer is suspended, it is necessary to reset the SDR of the data output side and restart data input side transfer (clear the shift clock counter).

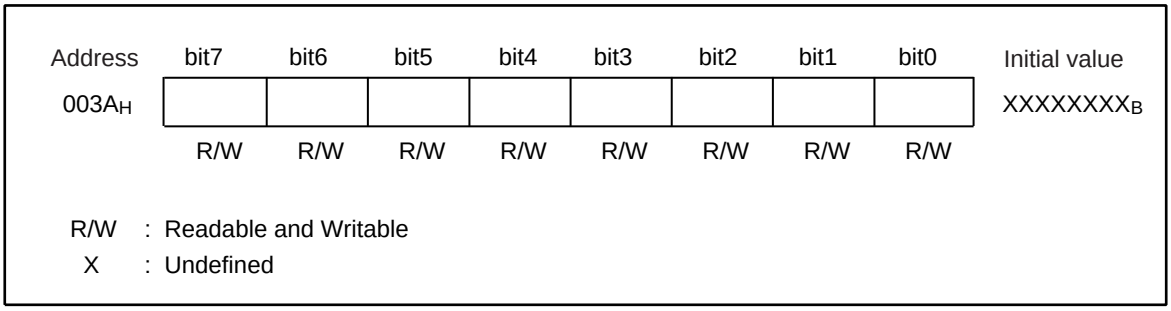
14.4.2 Serial Data Register (SDR)

The serial data register (SDR) retains 8-bit serial I/O transfer data. The SDR functions as a transmission data register at serial output operation. It functions as a reception data register at serial input operation.

■ Serial Data Register (SDR)

Figure 14.4-3 shows the bit structure of the SDR.

Figure 14.4-3 Serial Data Register (SDR)



- At serial output operation

  The SDR functions as a transmission data register at serial output operation. When serial I/O transfer is started (SMR: SST = 1), the 8-bit serial data written to the SDR is transferred. Transmission data does not remain in the SDR because it is shifted out via serial I/O transfer.

- At serial input operation

  The SDR functions as a reception data register. When serial I/O transfer is started (SMR: SST = 1), the serially transferred reception data is stored in the SDR.

- When the serial I/O is in transfer operation

  When the serial I/O is in transfer operation, do not write data to the SDR. Moreover, note that the read values have no significance.

  When serial output and serial input are allowed at the same time, serial I/O operation is performed.

14.5 Interrupt of 8-Bit Serial I/O

An 8-bit serial I/O interrupt is caused by completion of 8-bit serial data I/O.

■ Interrupt at Serial I/O Operation

In the 8-bit serial I/O, serial output operation and serial input operation are performed at the same time. When serial I/O transfer is started, the values in the serial data register (SDR) are input and output on a per bit basis in synchronization with the set shift clock cycle. When the shift clock of the 8th bit rises, the interrupt request flag bit (SMR: SIOF) is set to "1".

In this case, when the interrupt request output allowance bit is allowed (SMR: SIOE = 1), the interrupt request (IRQC) for CPU interrupt occurs.

Write "0" to the SIOF bit with the interrupt processing routine and clear the interrupt request. When 8-bit serial output is completed, the SIOF bit is set irrespective of the SIOE bit value.

If serial I/O transfer stop (SMR: SST = 0) and serial data transfer termination take place at the same time during serial I/O operation, the interrupt request flag bit (SMR: SIOF = 1). If the SIOE bit is allowed (0 → 1), however, the interrupt request occurs immediately.

■ 8-bit Serial I/O Interrupt Register and Vector Table

Table 14.5-1 8-bit Serial I/O Interrupt Register and Vector Table

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Setting bit		Higher	Lower
IRQC	ILR4 ($007E_H$)	LC1 (bit1)	LC0 (bit0)	$FFE2_H$	$FFE3_H$

For interrupt operation, see Section "3.4.2 Steps in the Interrupt Operation ".

14.6 Operations of Serial Output Functions

In the 8-bit serial I/O, 8-bit serial output operation synchronized with a shift clock is possible.

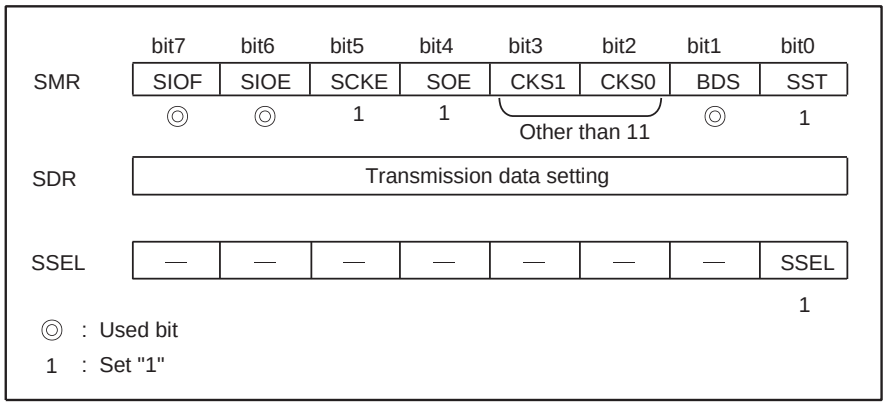
■ Serial Output Operation

Serial output operation is divided into serial output operation using an internal shift clock and serial output operation by using the external shift clock. When serial I/O operation is allowed, serial data is input in the SDR and, at the same time, the contents of the SDR are output to the serial data output pin (SO).

● Serial output operation via internal shift clock

Serial output operation using the internal shift clock requires the settings shown in Figure 14.6-1 .

Figure 14.6-1 Settings Required for Serial Output Operation using Internal Shift Clock

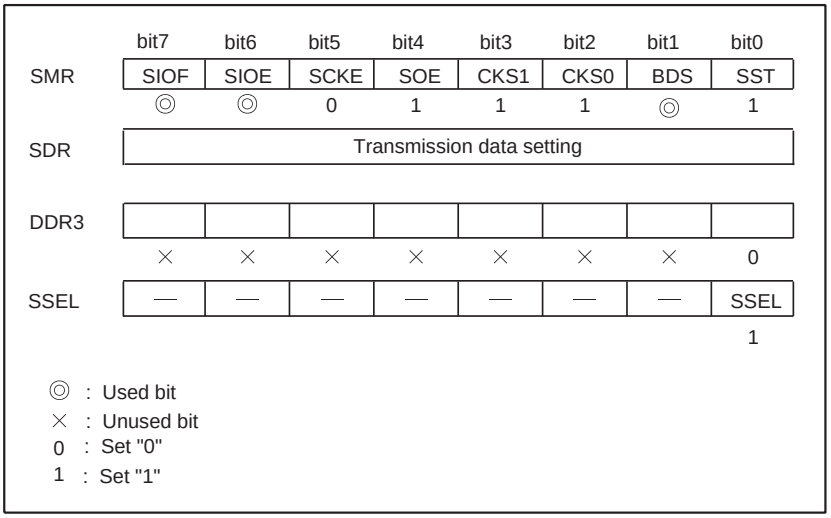


When serial output operation is started, the contents of the SDR are output to the SO pin in synchronization with the falling edge of the selected internal shift clock. In this case, the transfer destination (serial input side) must be in the external shift clock input wait state.

● Serial output operation using external shift clock

Serial output operation with the external shift clock requires the settings shown in Figure 14.6-2 .

Figure 14.6-2 Settings Required for Serial Output Operation using External Shift Clock

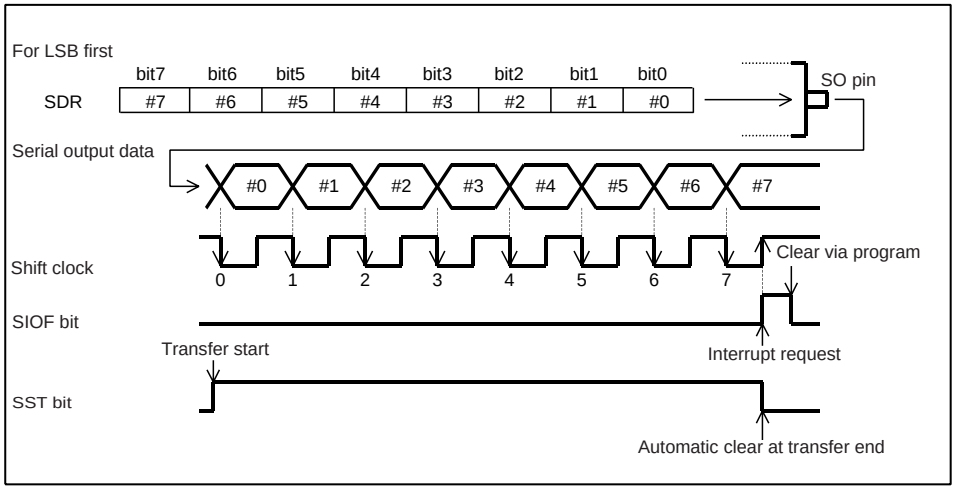


When serial output operation is allowed, the contents of the SDR are output to the SO pin in synchronization with the falling edge of the external shift clock. When serial operation is completed, immediately reset the SDR, set it again, then allow serial output operation (SMR: SST = 1) to prepare for the output of the next data.

When the remote serial input operation (rising edge) is completed and the 8-bit serial I/O enters the idle state (state in which it waits for the output of the next data), set the external shift clock to a high level.

Figure 14.6-3 shows 8-bit serial output operation.

Figure 14.6-3 8-bit Serial Output Operation



■ Operation at Serial Output Completion

At the rising edge of the shift clock for serial data of the 8th bit, the interrupt request flag bit (SMR: SIOF) is set to "1" and the serial I/O start bit (SMR: SST) is set (cleared) to "0".

14.7 Operations of Serial Input Functions

In the 8-bit serial I/O, 8-bit serial input operation synchronized with a shift clock is possible.

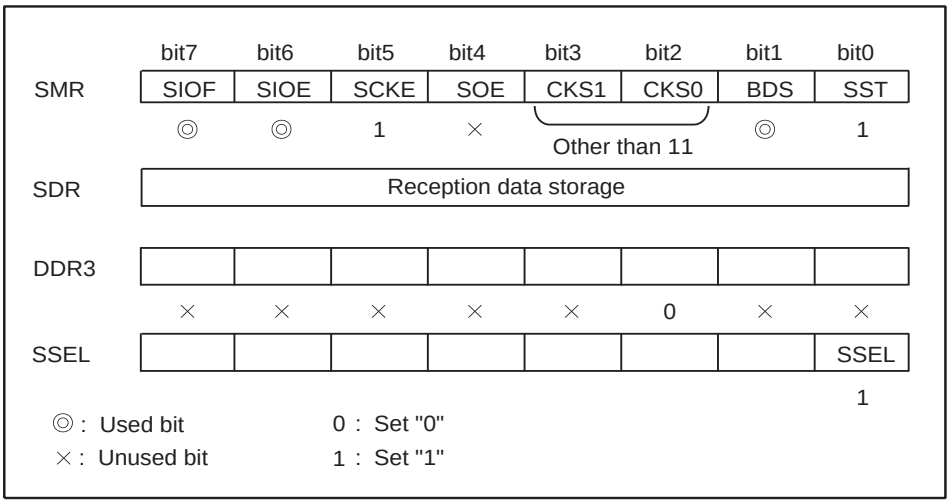
■ Serial Input Operation

Serial input operation is divided into serial input operation with an internal shift clock and serial input operation with an external shift clock. When serial I/O operation is allowed, serial data is input in the SDR and, at the same time, the contents of the SDR are output to the serial data output pin (SO).

● Serial input operation using internal shift clock

Serial input operation with the internal shift clock requires the settings shown in Figure 14.7-1 .

Figure 14.7-1 Settings Required for Serial Input Operation using Internal Shift Clock

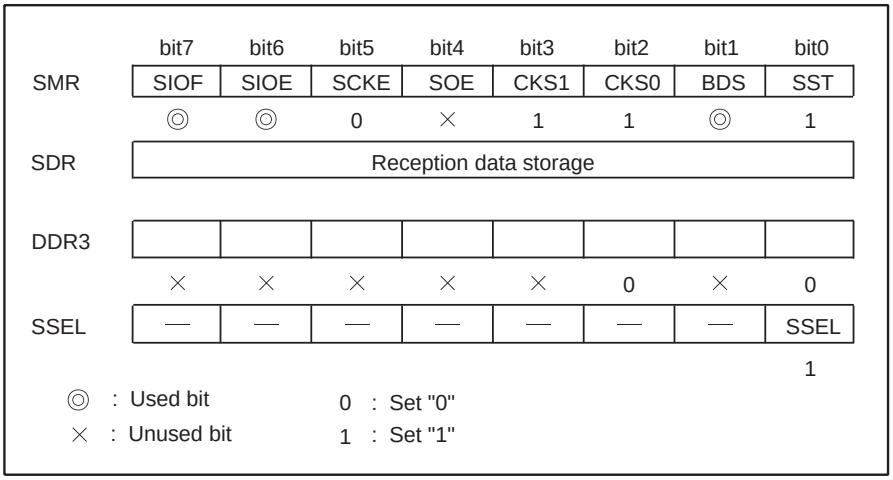


When serial input operation is started, the value of the serial data input pin (SI) is captured and held in the SDR in synchronization with the rising edge of the selected internal shift clock. In this case, the SDR of the transfer destination (serial output side) must already be set and the transfer destination must be in the external shift clock input wait state.

● Serial input operation using external shift clock

Serial input operation with the external shift clock requires the settings shown in Figure 14.7-2 .

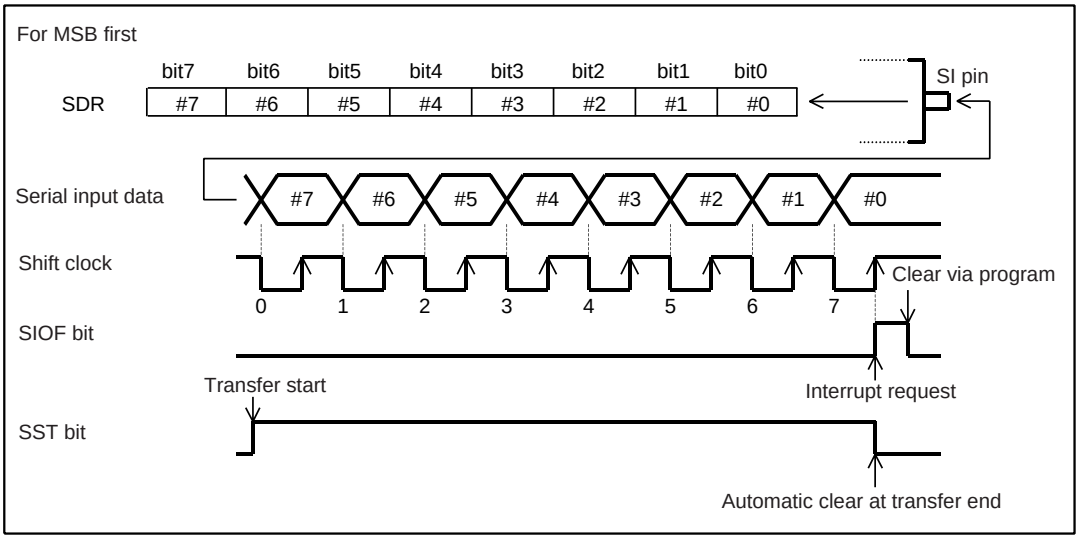
Figure 14.7-2 Settings Required for Serial Input Operation using External Shift Clock



When serial input operation is allowed, the value of the SI pin is captured and held in the SDR in synchronization with the rising edge of the external shift clock. When serial input is completed, immediately read the SDR and allow serial input operation (SMR: SST = 1) to prepare for the input of the next data. In this case, when the 8-bit serial I/O is idle (state in which it is waiting for the output of the next data), keep the external shift clock at a "H" level.

Figure 14.7-3 shows 8-bit serial input operation.

Figure 14.7-3 8-bit Serial Input Operation



■ Operation at Serial Input Completion

At the rising edge of the shift clock for the serial data of the 8th bit, the interrupt request flag bit (SMR: SIOF) is set to "1" and the serial I/O start bit (SMR: SST) is set (cleared) to "0".

14.8 8-Bit Serial I/O Operation in Each Mode

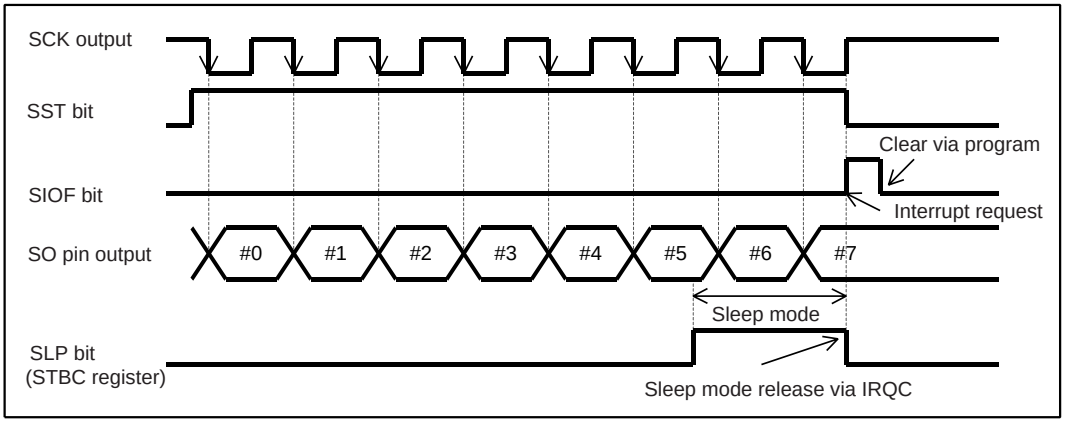
This section describes the operation of the 8-bit serial I/O if the 8-bit serial I/O switches to sleep or stop mode or a stop request is issued when it is in operation.

■ When the Internal Shift Clock is Used

● 8-bit serial I/O operation in sleep mode

In sleep mode, as shown in Figure 14.8-1 , the 8-bit serial I/O continues data transfer without stopping the serial I/O operation.

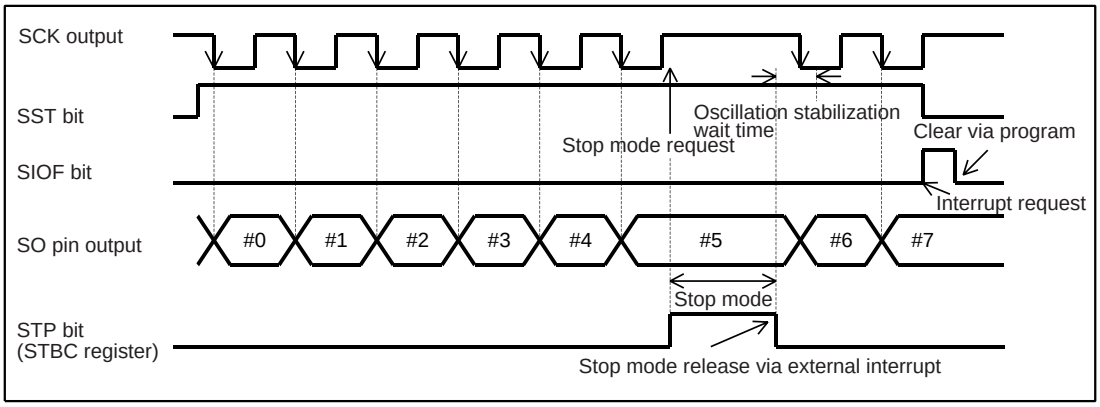
Figure 14.8-1 8-bit Serial I/O Operation in Sleep Mode (Internal Shift Clock)



● 8-bit serial I/O operation in stop mode

In stop mode, as shown in Figure 14.8-2 , the 8-bit serial I/O stops the serial I/O operation and suspends data transfer. After stop mode has been released, reinitialize the 8-bit serial I/O because operation is resumed halfway.

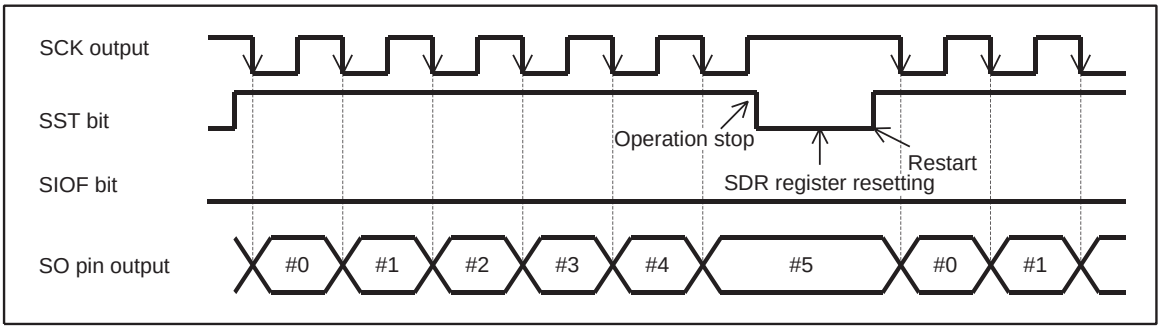
Figure 14.8-2 8-bit Serial I/O Operation in Stop Mode (Internal Shift Clock)



- 8-bit serial I/O operation at issuance of stop request during operation

  As shown in Figure 14.8-3 , if operation is stopped (SMR: SST = 0) during data transfer, the 8-bit serial I/O stops data transfer and clears the shift clock counter. For this reason, the transfer destination must also be initialized. If serial output is in operation, set the SDR again before restarting the 8-bit serial I/O.

Figure 14.8-3 8-bit Serial I/O Operation at Issuance of Stop Request during Operation(Internal Shift Clock)

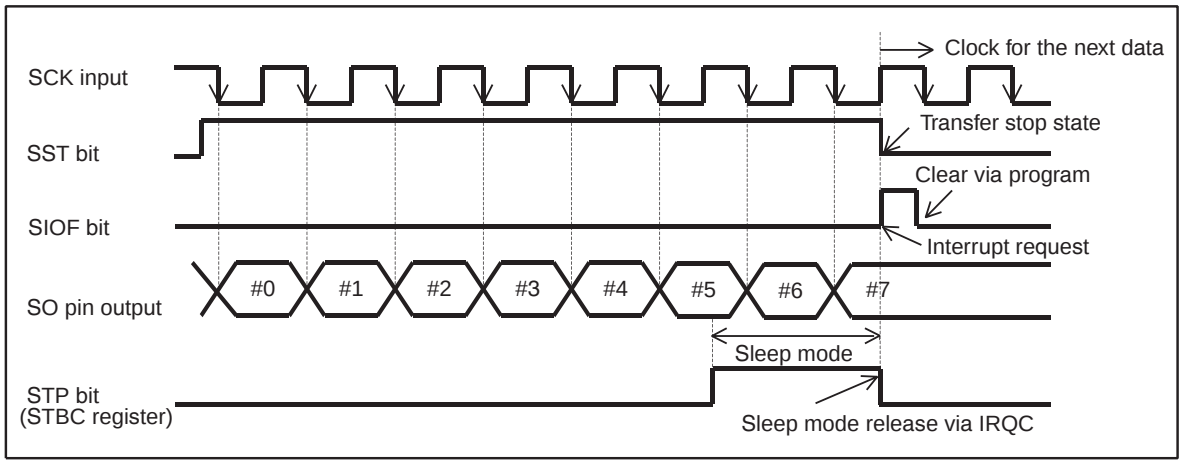


■ When the External Shift Clock is Used

● 8-bit serial I/O operation in sleep mode

In sleep mode, as shown in Figure 14.8-4 , the 8-bit serial I/O continues data transfer without stopping the serial I/O operation.

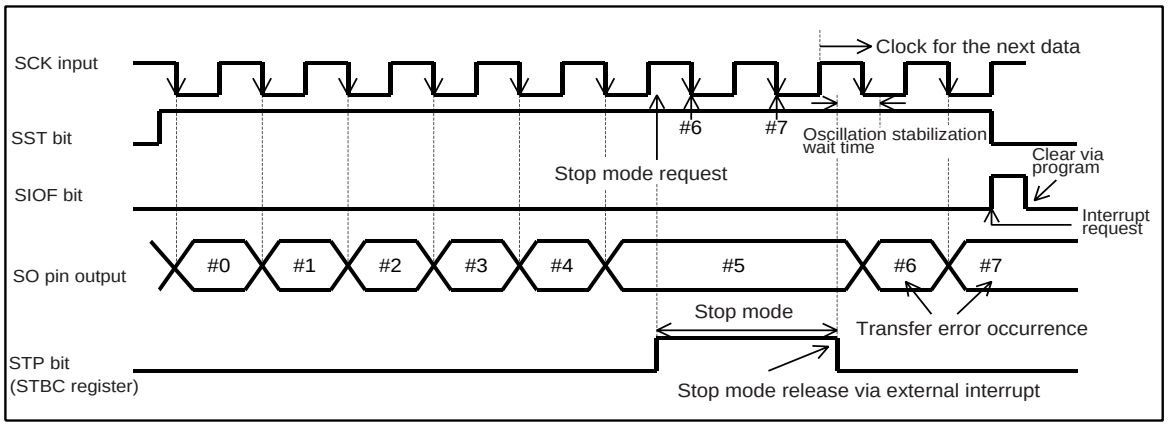
Figure 14.8-4 8-bit Serial I/O Operation in Sleep Mode (External Shift Clock)



● 8-bit serial I/O operation in stop mode

In stop mode, as shown in Figure 14.8-5 , the 8-bit serial I/O stops the serial I/O operation and suspends data transfer. After stop mode has been released, a transfer destination error occurs because operation is resumed halfway. In this case, initialize the 8-bit serial I/O.

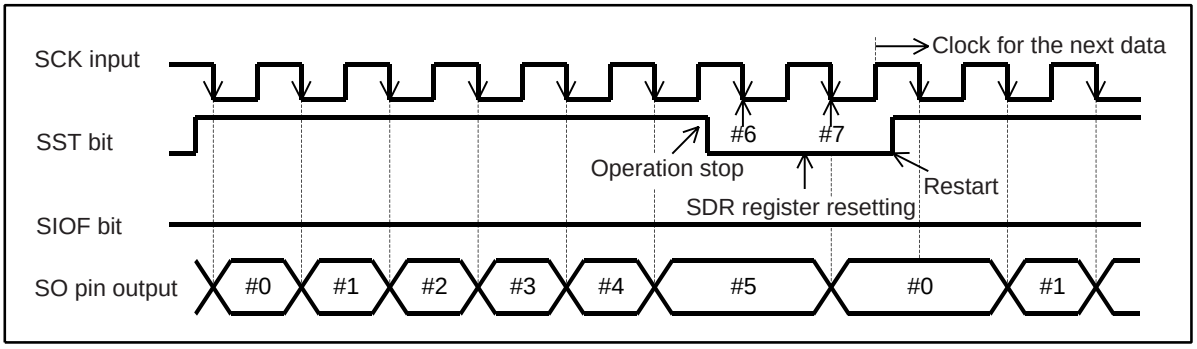
Figure 14.8-5 8-bit Serial I/O Operation in Stop Mode (External Shift Clock)



- 8-bit serial I/O operation at issuance of stop request during operation

  As shown in Figure 14.8-6 , if operation is stopped (SMR: SST = 0) during data transfer, the 8-bit serial I/O stops data transfer and clears the shift clock counter. For this reason, the transfer destination must also be initialized. If serial output is in operation, set the SDR again before restarting the 8-bit serial I/O. In this case, when the external clock is input, the SO pin output changes.

Figure 14.8-6 8-bit Serial I/O Operation at Issuance of Stop Request during Operation (External Shift Clock)



14.9 Notes on Using 8-Bit Serial I/O

This section provides notes on using the 8-bit serial I/O.

■ Notes on Using 8-bit Serial I/O

- Error at serial transfer start

  The time at which serial I/O transfer is started with a serial transfer program (SMR: SST = 1) is asynchronous with the time when the falling edge (output) or rising (input) edge of a shift clock occurs. For this reason, the time that lasts until the first serial data is input or output is delayed by a maximum of one cycle of the set shift clock.

- Malfunction due to noise

  If external noise causes an extra pulse (pulse exceeding the hysteresis width) to be placed on a shift clock during serial data transfer, the 8-bit serial I/O may malfunction.

- Notes on setting via program

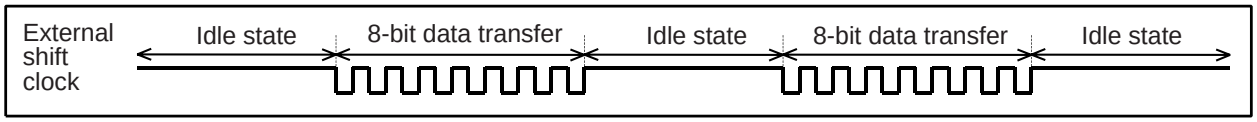
  - Write data to the serial mode register (SMR) and serial data register (SDR) only when the 8-bit serial I/O is stopped (SMR: SST = 0).
  - When starting or allowing serial I/O transfer (SMR: SST=1), do not change other bits of the SMR.
  - If MSB first is set when a shift clock is used in external shift clock input, the highest bit level is output as the SO pin output level. If LSB first is set, the lowest bit level is output as the SO pin output level. MSB first and LSB first are set when the external shift clock is input. In this case, however, serial data output must be allowed (SMR: SOE = 1) even if serial I/O transfer is stopped (SMR: SST = 0).
  - If serial I/O transfer stop (SMR: SST = 0) and serial data transfer termination take place at the same time during serial I/O operation, the interrupt request flag bit (SMR: SIOF) is not set to "1".
  - If the SIOF bit is set to "1" and the interrupt request output allowance bit is enabled (SMR: SIOE = 1), control cannot return from interrupt processing. Be sure to clear the SIOF bit.

- Shift clock idle state

  The external shift clock must maintain the "H" level during the wait time between one 8-bit data transfer and another (idle state). When the internal shift clock is selected (SMR: CKS1, CKS0 = not 11_B) and the P30/UCK/SCK pin is used as the shift clock output pin (SMR: SCKE = 1), data is output at the "H" level in the idle state.

  Figure 14.9-1 shows the shift clock idle state.

Figure 14.9-1 Shift Clock Idle State



14.10 Example of 8-Bit Serial I/O Connection

This section provides an example of mutual connection between 8-bit serial I/Os of MB89202/F202RA series for bidirectional serial I/O operation.

■ When Bidirectional Serial I/O Operation is Performed

Figure 14.10-1 Example of 8-bit Serial I/O Connection (Interface between MB89202/F202RA Series)

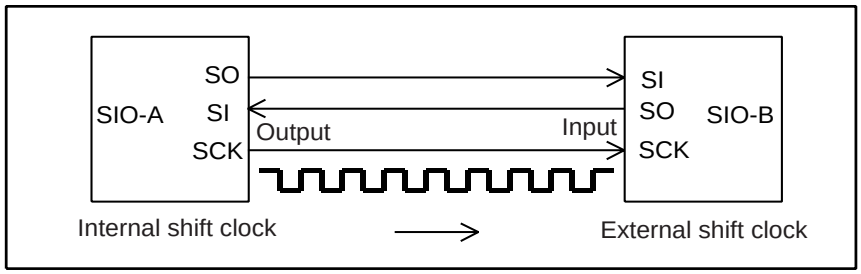
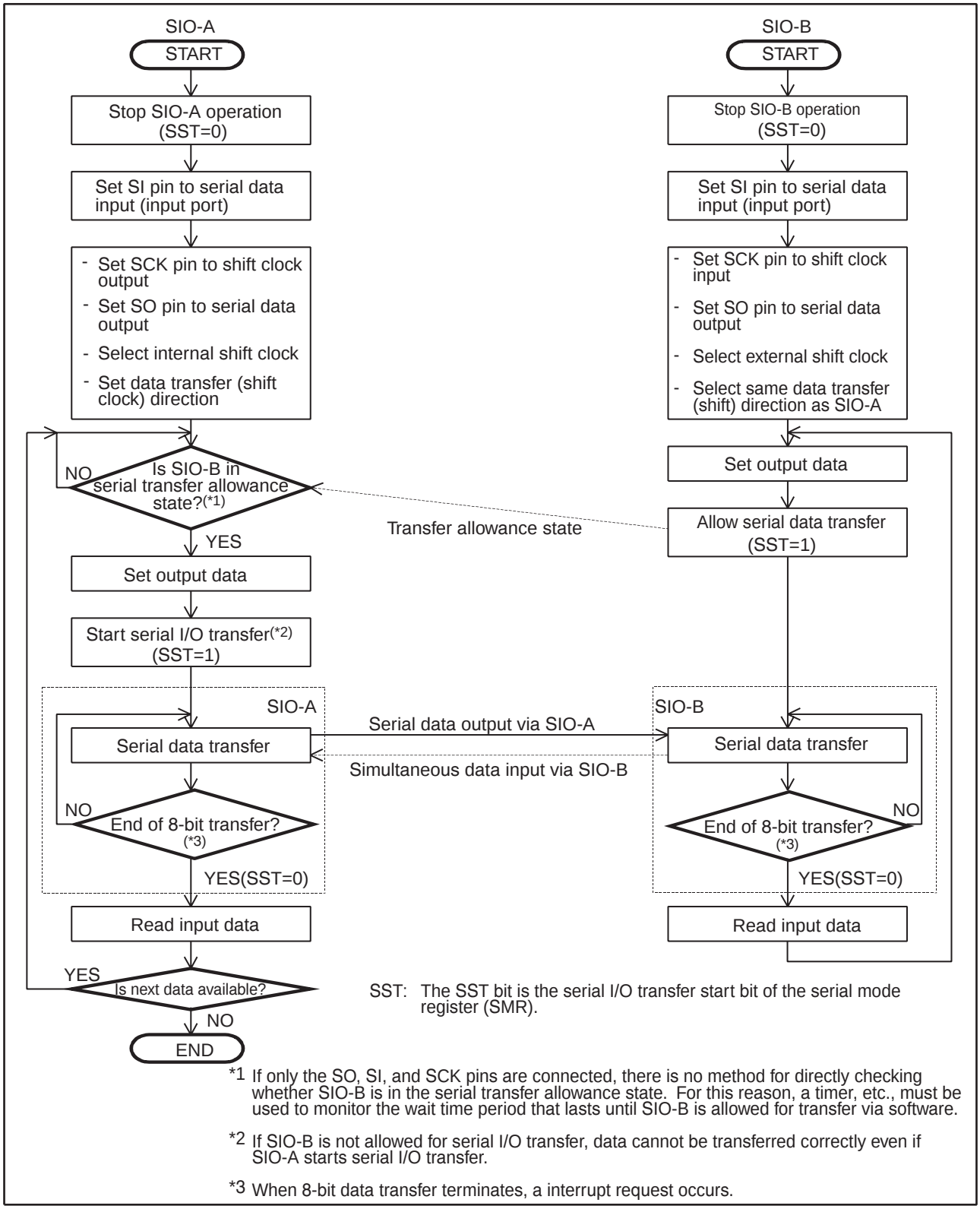


Figure 14.10-2 Bidirectional Serial I/O Operation



14.11 Program Example for 8-Bit Serial I/O

This section provides program example for 8-bit serial I/O.

■ Program Example for 8-bit Serial Output

● Processing Specifications

- The 8-bit serial output program outputs 8-bit serial data (55_H) from the SO pin of the 8-bit serial I/O. When serial I/O transfer terminates, an interrupt occurs.
- The program resets transfer data with the interrupt processing routine and outputs it continuously.
- The program operates in accordance with the internal shift clock. This clock is output from the SCK pin.
- If the shift clock is $32t_{INST}$ when the maximum gear speed (1 instruction cycle = $4/F_{CH}$) at the 12.5-MHz oscillation (F_{CH}), the transfer rate and interrupt cycle are as follows:

  Transfer rate = 12.5 MHz/4/32 = 97.7 kbps, interrupt cycle = 8 × 32 × 4/10 MHz = 81.92 μs

● Coding example

```
SMR     EQU     0039H               ;  Address of serial mode register
SDR     EQU     003AH               ;  Address of serial data register
SSEL    EQU     003BH               ;  Address of serial/UART selection register
SIOF    EQU     SMR:7               ;  Defines the interrupt request flag bit.
SST     EQU     SMR:0               ;  Defines the serial I/O transfer start bit.
ILR4    EQU     007EH               ;  Address of interrupt request setting register 4
INT_V   DSEG    ABS                 ;  [DATA SEGMENT]
        ORG     0FFE2H
IRQC    DW      WARI                ;  Sets an interrupt vector.
INT_V   ENDS
;--------------------Main program-------------------------------------------------------------------
        CSEG                        ;  [CODE SEGMENT]
                                    ;  The stack pointer (SP), etc., are already initialized.
          :
        CLRI                        ;  Disables interrupts.
        CLRB    SST                 ;  Stops serial I/O transfer.
        MOV     ILR4,#11111101B     ;  Sets the interrupt level to 1.
        MOV     SDR,#55H            ;  Sets transfer data (55H).
        MOV     SMR,#01111000B      ;  Clears the interrupt request flag, allows the interrupt
                                       request output, shift clock output (SCK), and serial data
                                       output (SO), selects 32tINST, and sets LSB first.
        MOV     SSEL,#00000001B     ;  Selects the 8-bit serial I/O.
        SETB    SST                 ;  Starts serial I/O transfer.
        SETI                        ;  Enables interrupts.
          :
```

```
;--------------------Interrupt processing routine-----------------------------------------------------------
WARI    CLRB    SIOF            ; Clears the interrupt request flag.
        PUSHW   A
        XCHW    A,T             ; Saves A and T.
        PUSHW   A
        MOV     SDR,#55H        ; Resets transfer data (55H).
        SETB    SST             ; Starts serial I/O transfer.
          :
        User processing
          :
        POPW    A
        XCHW    A,T             ; Returns A and T.
        POPW    A
        RETI
        ENDS
; ---------------------------------------------------------------------------------------------------------------
        END
```

■ Program Example for 8-bit Serial Input

● Processing specifications

- The 8-bit serial input program inputs 8-bit serial data from the SI pin of the 8-bit serial I/O. When serial I/O transfer terminates, an interrupt occurs.
- The program reads transfer data with the interrupt processing routine and inputs it continuously.
- The program uses the external shift clock to be input from the SCK pin.

● Coding example

```
DDR3    EQU     000DH           ; Address of data direction register 3
SMR     EQU     0039H           ; Address of serial mode register
SDR     EQU     003AH           ; Address of serial data register
SSEL    EQU     003BH           ; Address of serial/UART selection register
SIOF    EQU     SMR:7           ; Defines the interrupt request flag bit.
SST     EQU     SMR:0           ; Defines the serial I/O transfer start bit.
ILR4    EQU     007EH           ; Address of interrupt request setting register 4
INT_V   DSEG    ABS             ; [DATA SEGMENT]
        ORG     0FFE2H
IRQC    DW      WARI            ; Sets an interrupt vector.
INT_V   ENDS
;--------------------Main program---------------------------------------------------------------------------
        CSEG                    ; [CODE SEGMENT]
                                ; The stack pointer (SP), etc., is already initialized.
          :
        MOV     DDR3,#00000000B ; Sets the P30/SCK and P32/SI pins to input.
        CLRI                    ; Disables interrupts.
        CLRB    SST             ; Stops serial I/O transfer.
        MOV     ILR4,#11111101B ; Sets the interrupt level to 1.
```

```
        MOV     SMR,#01001100B  ;  Clears the interrupt request flag, allows the interrupt
                                   request output, sets shift clock input (SCK), prohibits
                                   serial data output (SO), selects the external shift clock,
                                   and sets LSB first.
        MOV     SSEL,#00000001B ;  Selects the 8-bit serial I/O.
        SETB    SST             ;  Allows serial I/O transfer.
        SETI                    ;  Enables interrupts.
          :
;-------------------Interrupt processing routine-----------------------------------------------------------
WARI    CLRB    SIOF            ;  Clears the interrupt request flag.
        PUSHW   A
        XCHW    A,T
        PUSHW   A
        MOV     A,SDR           ;  Reads transfer data.
        SETB    SST             ;  Allows serial I/O transfer.
          :
        User processing
          :
        POPW    A
        XCHW    A,T
        POPW    A
        RETI
        ENDS
; ---------------------------------------------------------------------------------------------------------
        END
```

CHAPTER 15

BUZZER OUTPUT

This chapter describes the functions and operation of the buzzer output.

15.1 Overview of the Buzzer Output

For the buzzer output, four kinds of output frequencies (square waves) can be selected. The buzzer output may be used for the confirmation tone of key input and other tones.

■ Buzzer Output Function

The buzzer output function is a function for outputting a signal (square wave) used for tones such as a confirmation tone.

For the buzzer output, it is selectable whether to output one of four output frequencies or to disable the output.

As the buzzer output, four kinds of divided-frequency outputs are supplied from the time-base timer.

Note:

The time-base timer supplies clock for the buzzer output. Therefore, buzzer output will be affected when time-base timer is cleared.

Table 15.1-1 lists the four kinds of output frequencies (square waves) specifiable for the buzzer output.

Table 15.1-1 Output Frequencies

Clock supplier	Buzzer output	Square wave output (at 12.5 MHz)
time-base timer	$2^{13}/F_{CH}$	$F_{CH}/2^{13}$ (1.526 kHz)
	$2^{12}/F_{CH}$	$F_{CH}/2^{12}$ (3.052 kHz)
	$2^{11}/F_{CH}$	$F_{CH}/2^{11}$ (6.104 kHz)
	$2^{10}/F_{CH}$	$F_{CH}/2^{10}$ (12.21 kHz)

F_{CH}: Oscillation frequency

Note:

Calculation example of an output frequency

If time-base timer output $F_{CH}/2^{10}$ is selected in the buzzer register (BZCR) (BZ2=1, BZ1=0, and BZ0=0) and the oscillation (F_{CH}) is 12.5 MHz, the output frequency being output from the BZ pin is calculated as follows:

$$\begin{aligned}\text{Output frequency} &= F_{CH}/2^{10}\\ &= 12.5\ \text{MHz}/1024\\ &\fallingdotseq 12.21\ \text{kHz}\end{aligned}$$

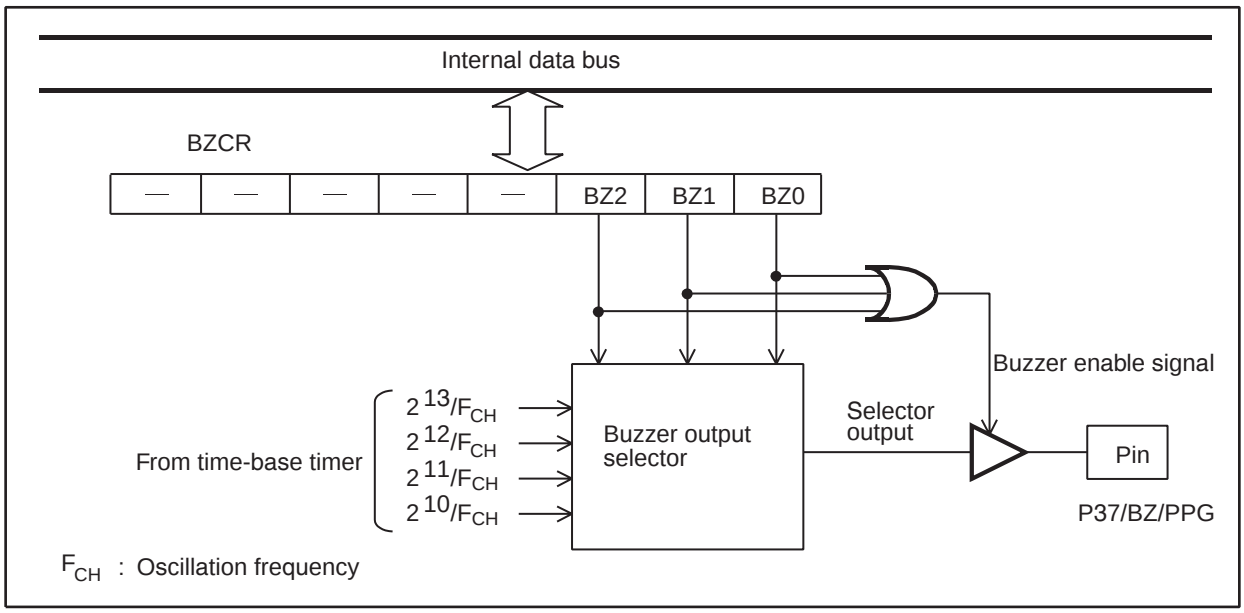
15.2 Configuration of the Buzzer Output

The buzzer output consists of the following two blocks:

- **Buzzer output selector**
- **Buzzer register (BZCR)**

■ Block Diagram of the Buzzer Output

Figure 15.2-1 Block Diagram of Buzzer Output



● Buzzer output selector

The buzzer output selector is a circuit for selecting one of the four frequencies (square waves) output from the time-base timer. The buzzer register (BZCR) sets it.

● Buzzer register (BZCR)

The buzzer register (BZCR) is a register for setting the buzzer output frequency and enable the buzzer output. When the BZCR register sets an output frequency (other than 000_B), the buzzer output is enabled so that the P37/BZ/PPG pin automatically becomes the buzzer output (BZ) pin. Even if the PPG pin has been enabled, the BZ pin has higher priority.

15.3 Pin of the Buzzer Output

The pin related to the buzzer output is P37/BZ/PPG.

■ P37/BZ/PPG Pin

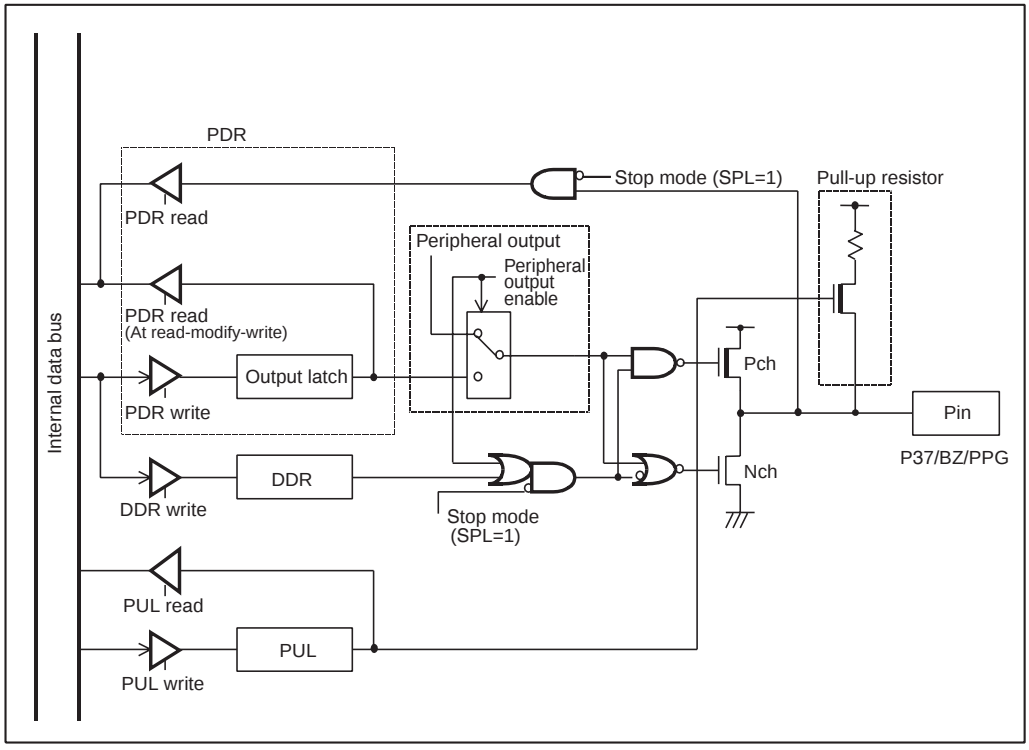
The P37/BZ/PPG pin works as a general-purpose I/O (P37) pin, output pin for the buzzer output (BZ), or output pin for the 12-bit PPG (PPG).

- BZ pin

The BZ pin outputs the square wave for the buzzer of the frequency having been specified for the BZ pin. When a buzzer output frequency is specified (other than BZCR:BZ2,BZ1,BZ0=000_B), the P37/BZ/PPG pin automatically works as the BZ pin regardless of the value of output latch. Even if the PPG output has been enabled, it works as the BZ pin that has higher priority.

■ Block Diagram of the Pin Related to the Buzzer Output

Figure 15.3-1 Block Diagram of Pin Related to Buzzer Output



Note:

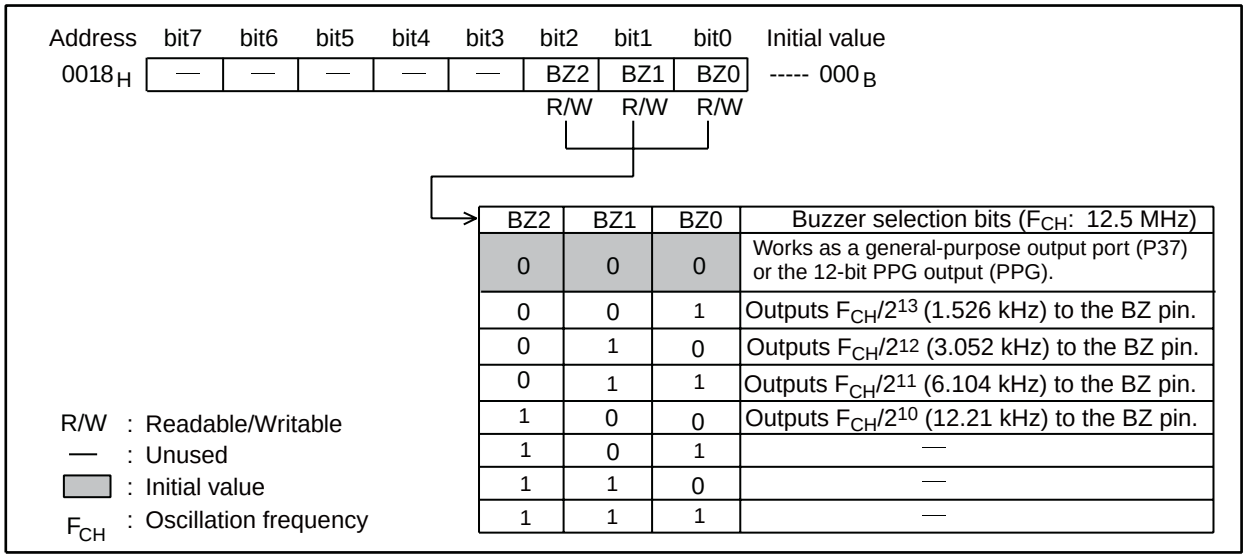
If pull-up resistor supported is specified by the pull-up setting register, the state of the pin in stop mode (SPL=1) is not Hi-Z but "H" level (pull-up state). During a reset, however, the pull-up is disabled and the state is Hi-Z.

15.4 Buzzer Register (BZCR)

The buzzer register (BZCR) is used to select an output frequency of the buzzer and also serves as the buzzer output enable.

■ Buzzer Register (BZCR)

Figure 15.4-1 Buzzer Register (BZCR)



BZ2	BZ1	BZ0	Buzzer selection bits (F_{CH}: 12.5 MHz)
0	0	0	Works as a general-purpose output port (P37) or the 12-bit PPG output (PPG).
0	0	1	Outputs $F_{CH}/2^{13}$ (1.526 kHz) to the BZ pin.
0	1	0	Outputs $F_{CH}/2^{12}$ (3.052 kHz) to the BZ pin.
0	1	1	Outputs $F_{CH}/2^{11}$ (6.104 kHz) to the BZ pin.
1	0	0	Outputs $F_{CH}/2^{10}$ (12.21 kHz) to the BZ pin.
1	0	1	—
1	1	0	—
1	1	1	—

Table 15.4-1 Functions of Each Bit in Buzzer Register (BZCR)

Bit name		Function
bit7 to bit3	Unused bits	• Undefined at read • No effect on the operation at write
bit2 to bit0	BZ2, BZ1, and BZ0: Buzzer selection bits	• Select a buzzer output and enable the output. • If 000_B is set to these bits, the buzzer output is disabled and the pin works as a general-purpose port (P37) or as the 12-bit PPG output (PPG). With the exception of 000_B, the pin becomes the buzzer pin and outputs a square wave. Even if the pin has been functioning as the 12-bit PPG output, setting a value other than 000_B causes the pin to work as the BZ pin prior to its operation as the PPG pin. For the buzzer output, four kinds of time-base timer divided cycle outputs are supplied.

15.5 Program Example for Buzzer Output

This section shows an program example for buzzer output.

■ Program Example for Buzzer Output

- Processing specification

  Suppose that the buzzer output of 3.052 kHz is output to the BZ pin and then the buzzer output is cut off.

  If $2^{12}/F_{CH}$ is selected when the oscillation (F_{CH}) is 12.5 MHz, the buzzer output frequency is calculated as follows:

  Buzzer output frequency: 12.5 MHz/2^{12} = 12.5 MHz/4096 = 3.052 kHz

- Coding example

```
BZCR    EQU     0018H               ;   Address of the buzzer register
;-------------------Main program-------------------------------------------------------------------
        CSEG                        ;   [CODE SEGMENT]
          :
        MOV     BZCR,#00000010      ;   Buzzer output on (3.052 kHz / Oscillation of 12.5 MHz)
          :
          :
          :
        MOV     BZCR,#00000000      ;   Buzzer output off (I/O port or PPG output)
          :
          :
        ENDS
; ------------------------------------------------------------------------------------------------------
        END
```

CHAPTER 16

WILD REGISTER FUNCTION

This chapter describes the functions and operation of the wild registers.

16.1 Overview of the Wild Register Function

The wild register function is a function for patching the faulty part of a program by setting the address and the correct data in the incorporated registers. Up to two bytes of data correction is possible.

■ Wild Register Function

The wild register function assigns an address in the ROM area of the microcontroller and replaces the existing data corresponding to the address, with new data. For example, if an error exists in a program, setting the address of the faulty part and correction data to the register can correct the faulty data.

■ Wild Register Applicable Addresses

The address area where the wild register function can apply varies slightly with the models. Table 16.1-1 shows the wild register applicable addresses for each model.

Table 16.1-1 Wild Register Applicable Addresses

Model name	ROM area
MB89V201	8000_H to $FFFF_H$
MB89202/F202	$C000_H$ to $FFFF_H$

Note:

The wild register function cannot be debugged with a tool. Perform the operation check of the wild register on the actual microcontroller, MB89202/F202/F202RA.

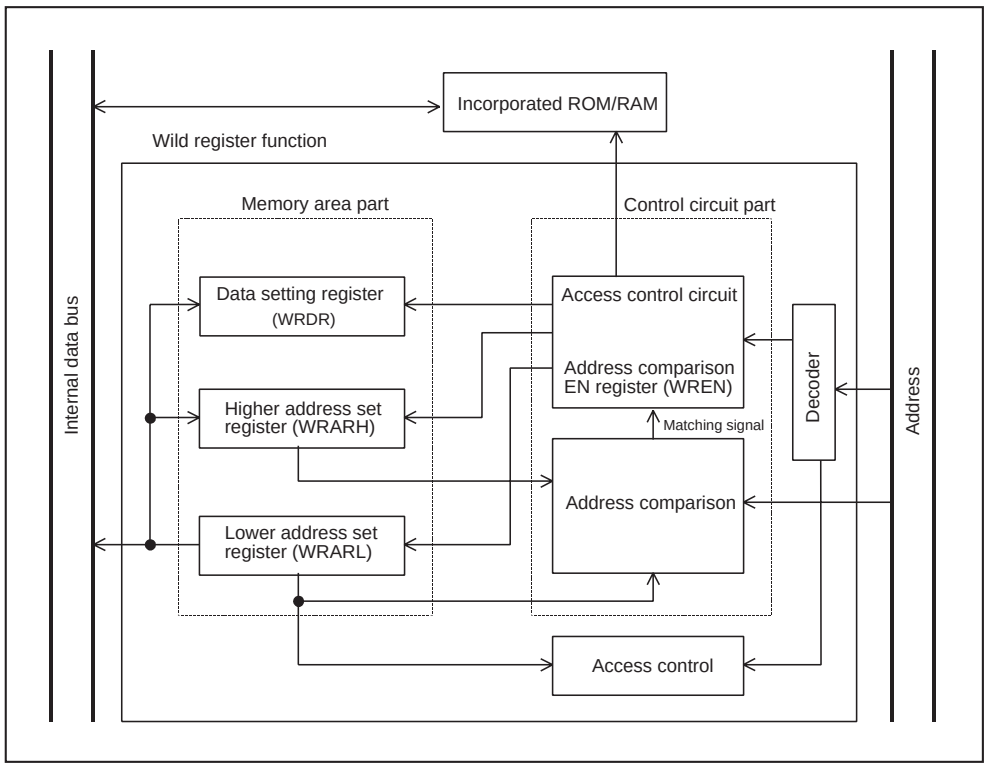
16.2 Configuration of the Wild Register Function

The wild register function consists of the following two blocks:
Memory area part
- **Data setting register (WRDR)**
- **Higher address set register (WRARH)**
- **Lower address set register (WRARL)**

Control circuit part

■ Block Diagram of the Wild Register Function

Figure 16.2-1 Block Diagram of Wild Register Function



● Memory area part

This part consists of the data setting register, higher address set register ("H" address), and lower address set register ("L" address). Set the address and data to be replaced by the wild register. The MB89202/F202RA series incorporates two bytes for each register.

● Control circuit part

This part compares the data held in the address set registers and the actual data on the address bus. If it detects a match, it sets the data in the data setting register to the data bus. The control circuit part can control the operation by the address comparison EN register.

16.3 Registers of the Wild Register Function

Figure 16.3-1 shows the registers related to the wild register function.

■ Registers Related to the Wild Register Function

Figure 16.3-1 Registers Related to Wild Register Function

WRDR0,WRDR1 (Data setting register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0042_H	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	$XXXXXXXX_B$
0045_H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARH0,WRARH1 (Higher address set register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0040_H	RA15	RA14	RA13	RA12	RA11	RA10	RA09	RA08	$XXXXXXXX_B$
0043_H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WRARL0,WRARL1 (Lower address set register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0041_H	RA07	RA06	RA05	RA04	RA03	RA02	RA01	RA00	$XXXXXXXX_B$
0044_H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WREN (Address comparison EN register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0046_H	—	—	—	—	—	—	EN01	EN00	$------00_B$
							R/W	R/W	

WROR (Data test set register)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0047_H	—	—	—	—	—	—	RESV1	RESV0	$------00_B$
							R/W	R/W	

R/W : Readable and Writable
— : Unused
X : Undefined

16.3.1 Data Setting Registers (WRDR0 and WRDR1)

The data setting registers (WRDR0 and WRDR1) are registers where the correct data used by the wild register function is set.

■ Data Setting Register (WRDR)

Figure 16.3-2 Data Setting Register (WRDR)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WRDR0	0042_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	
WRDR1	0045_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 : Readable and Writable
X : Undefined

Table 16.3-1 Functions of Data Setting Register (WRDR)

Wild register number	Register name	Function
0	WRDR0	1-byte registers that store the data at the address assigned by WRARL and WRARH. The data will be effective at the addresses (WRARL and WRARH) corresponding to the individual wild register numbers.
1	WRDR1	

Note:

The WRDR register is readable, only when the WREN register (address comparison EN register) is set.

16.3.2 Higher Address Set Registers (WRARH0 and WRARH1)

The higher address set registers (WRARH0 and WRARH1) are registers where the higher byte of addresses to be corrected by the wild register function are set.

■ Higher Address Set Register (WRARH)

Figure 16.3-3 Higher Address Set Register (WRARH)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WRARH0	0040 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	
WRARH1	0043 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	

R/W : Readable and Writable
X : Undefined

Table 16.3-2 Functions of Higher Address Set Register (WRARH)

Wild register number	Register name	Function
0	WRARH0	1-byte registers that specify the higher addresses of memory being assigned. They specify the addresses corresponding to the individual wild register numbers.
1	WRARH1	

16.3.3 Lower Address Set Registers (WRARL0 and WRARL1)

The lower address set registers (WRARL0 and WRARL1) are registers where the lower byte of addresses to be corrected by the wild register function are set.

■ Lower Address Set Register (WRARL)

Figure 16.3-4 Lower Address Set Register (WRARL)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WRARL0	0041_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	
WRARL1	0044_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	

R/W : Readable and Writable
X : Undefined

Table 16.3-3 Functions of Lower Address Set Register (WRARL)

Wild register number	Register name	Function
0	WRARL0	1-byte registers that specify the lower addresses of memory being assigned. They specify the addresses corresponding to the individual wild register numbers.
1	WRARL1	

16.3.4 Address Comparison EN Register (WREN)

The address comparison EN register (WREN) is a register that enables the operation of wild register function for the individual wild register numbers.

■ Address Comparison EN Register (WREN)

Figure 16.3-5 Address Comparison EN Register (WREN)

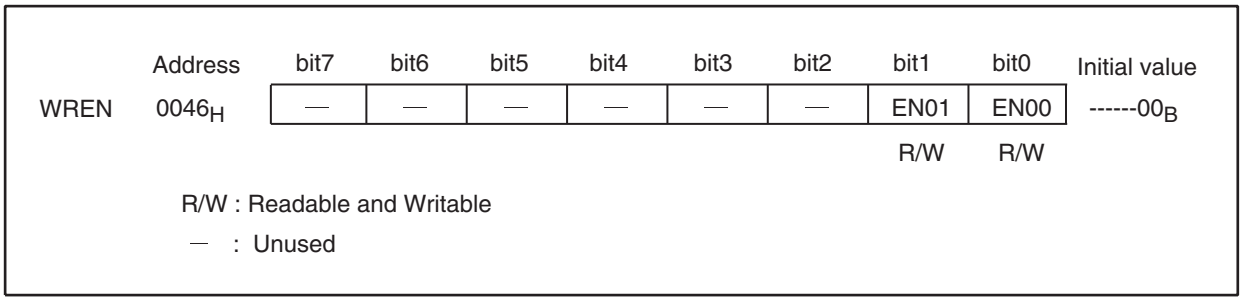


Table 16.3-4 Explanation of Functions of Each Bit in Address Comparison EN Register (WREN)

Bit name		Function
bit7 to bit2	Unused bits	Undefined at read No effect to the operation at write
bit1	EN01	When this bit is "0", the corresponding wild register function does not work. When this bit is "1", the wild register function is enabled. If there is a match with the address held in WRARH1 and WRARL1, the value of WRDR1, instead of ROM, is output to the internal bus.
bit0	EN00	When this bit is "0", the corresponding wild register function does not work. When this bit is "1", the wild register function is enabled. If there is a match with the address held in WRARH0 and WRARL0, the value of WRDR0, instead of ROM, is output to the internal bus.

16.3.5 Data Test Set Register (WROR)

A test register. Do not access this register.

16.4 Operations of the Wild Register Functions

This section describes the operation order of the wild register.

■ Operation Order of the Wild Register Function

Table 16.4-1 describes the operation order of the wild register. In the operation example column, it corrects data at address $FC36_H$, from FF_H to $B5_H$.

Table 16.4-1 Operation Order of Wild Register

	Operation	Operation example
1	Set an address of the wild register correspondence area to the address set register.	Address: $FC36_H$/data: FF_H WRARL0=36_H WRARH0=FC_H
2	Set the correction data to the data setting register.	WRDR0=$B5_H$
3	Set "1" to the address comparison EN00 bit.	WREN=01_H
4	The wild register works at the time of address matching.	When address = $FC36_H$ is accessed ↓ Data = $B5_H$

■ Wild Register Addresses List

Table 16.4-2 lists the addresses corresponding to the wild register numbers.

Table 16.4-2 Wild Register Addresses List

	Higher address		Lower address		Data	
	Register name	Address	Register name	Address	Register name	Address
1	WRARH0	040_H	WRARL0	041_H	WRDR0	042_H
2	WRARH1	043_H	WRARL1	044_H	WRDR1	045_H

CHAPTER 17

FLASH MEMORY

This chapter describes the functions and operation of the 128K-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".

Note: A user must create a serial programmer for writing.

17.1 Overview of Flash Memory

The 128K-bit flash memory is mapped to the $C000_H$ 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

- 16 Kbyte × 8-bit configuration
- Use of automatic program algorithm (Embedded Algorthm)
- 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
- Minimum of 10000 write / erase operations (MB89F202/F202RA)

■ High voltage supply on $\overline{RST}$ pin (applicable to MB89F202RA only)

During writing data to or erasing all data in flash memory, a typical +10V D.C. voltage should be applied at the RST pin. After applying the high voltage, wait for 10ms before writing data or erasing all data in flash memory. And this applied voltage should be kept at the $\overline{RST}$ pin until data writing or erasing has been completed.

■ 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
0079_H	INTE	RDYINT	WE	RDY	-	-	-	-
Read/write	(R/W)	(R/W)	(R/W)	(R)	(-)	(-)	(-)	(-)
Initial value	(0)	(0)	(0)	(X)	(-)	(-)	(-)	(-)

17.2 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 17.2-1 Flash Memory Control Status Register (FMCS)

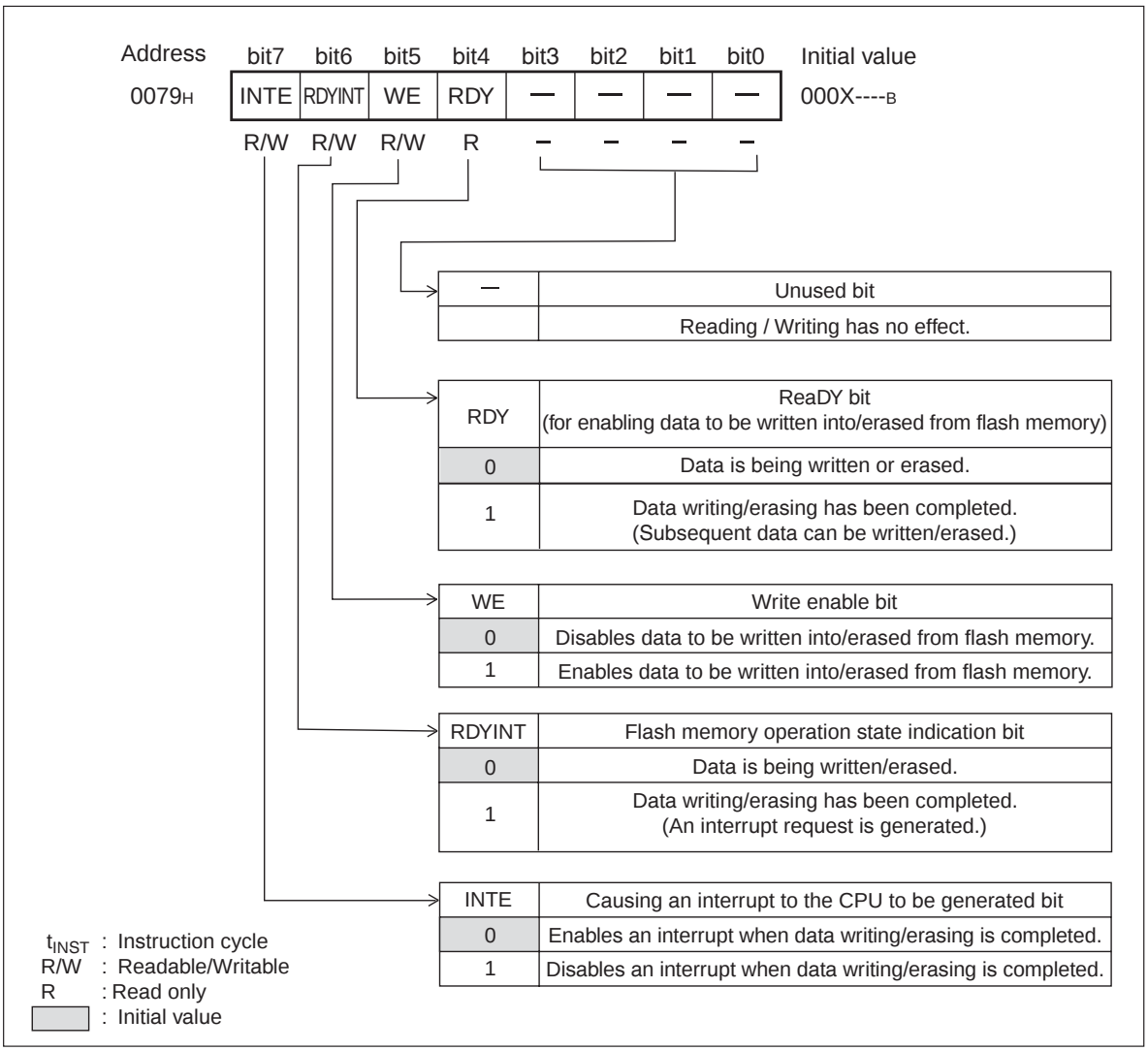
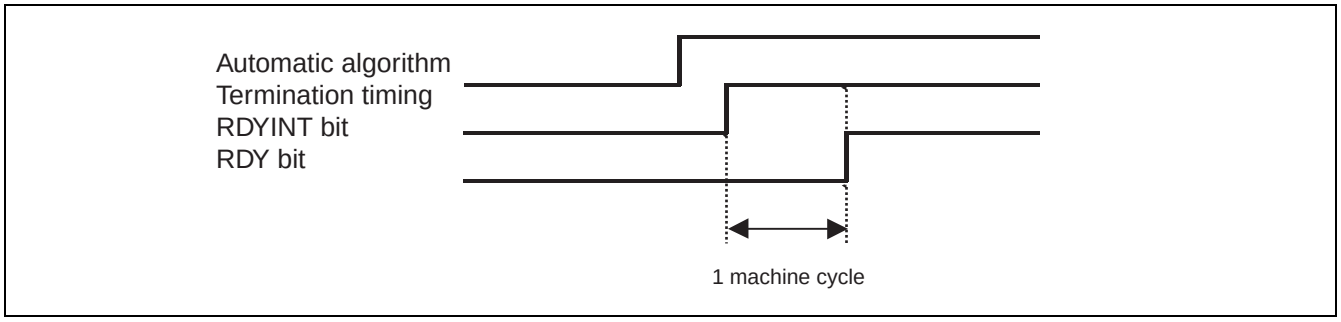


Table 17.2-1 Explanation of Functions of Each Bit in the Flash Memory Control Status Register (FMCS)

Bit name		Description
bit7	INTE: Causing an interrupt to the CPU to be generated bit	Bit causing an interrupt (IRQB) to the CPU to be generated when writing into or erasing from flash memory is completed. An interrupt (IRQB) 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: Flash memory operation state indication bit	Bit for indicating operation status of flash memory. This bit is set to "1" when writing into or erasing from flash memory is completed. After data has been written into or erased from flash memory and this bit has been set to "1", subsequent 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 "17.3 Starting the Flash Memory Automatic Algorithm ". The read modifier write (RMW) command always reads "1" from this bit.
bit5	WE: Write enable bit	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 $C000_H$ to $FFFF_H$ (see Section "17.3 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	Bit for status checking 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 "0". However, a read command, reset command, and suspend commands such as the sector erase suspend command can be accepted while this bit is "0".
bit3 to bit0	Unused bits	Reading / Writing for these bits have no effect.

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.



17.3 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.

■ Command Sequence Table

Table 17.3-1 lists the commands used for flash memory write/erase.

Table 17.3-1 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	XXXX	F0	-	-	-	-	-	-	-	-	-	-
	4	FAAA	AA	F554	55	FAAA	F0	RA	RD	-	-	-	-
Write program	4	FAAA	AA	F554	55	FAAA	A0	PA	PD	-	-	-	-
Chip Erase	6	FAAA	AA	F554	55	FAAA	80	FAAA	AA	F554	55	FAAA	10

*: Both of the two types of Read/Reset commands can reset the flash memory to read mode.

Notes:

- 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.
  RD: Read data
  PD: Write data.
- The flash memory can only accept the command sequences mentioned on the above table (Read/Reset, Write program, Chip Erase), other command sequences are strictly prohibited to be sent to the flash memory or else the flash memory may become malfunction.

17.4 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 sequence flags.

■ Hardware Sequence Flags

The hardware sequence flags are configured from the five-bit output of DQ7, DQ6, DQ5, and DQ2. The functions of these bits are those of the data polling flag (DQ7), toggle bit flag (DQ6), timing limit exceeded flag (DQ5), and toggle bit2 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 17.3-1 in Section "17.3 Starting the Flash Memory Automatic Algorithm "). Table 17.4-1 lists the bit assignments of the hardware sequence flags.

Table 17.4-1 Bit Assignments of Hardware Sequence Flags

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Hardware sequence flag	DQ7	DQ6	DQ5	-	-	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 status 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 17.4-2 lists the functions of the hardware sequence flags.

Table 17.4-2 Hardware Sequence Flag Functions

State		DQ7	DQ6	DQ5	DQ2
Executing	Automatic writing operation	$\overline{DQ7}$	Toggle	0	1
	Automatic erasing operation	0	Toggle	0	Toggle
Exceeding the time limit	Automatic writing operation	$\overline{DQ7}$	Toggle	1	1
	Automatic erasing operation	0	Toggle	1	Toggle

17.4.1 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 bit7 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 bit7 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.

Note:

When the automatic algorithm comes to the end of its operation, bit7 (data polling) changes its state asynchronously during a read operation. This means that flash memory sends data about the operation state to bit7 and will then send out fixed data. When flash memory ends the automatic algorithm or even if bit7 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.

17.4.2 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 bit6 toggle and outputs the value of bit6 (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.

17.4.3 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

Bit5 indicates that execution of the automatic algorithm exceeded the time (internal pulse count) specified in flash memory. For an excess, bit5 outputs 1. Thus, if this bit outputs 1 while the automatic algorithm is operating, data writing or data erasing failed.

Bit5 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 bit7 (data polling) and bit6 (toggle bit) remains unchanged (toggled). If the time limit is exceeded while there is a failure, "1" is set in bit5. In this case, note that the setting of bit5 to "1" does not indicate a flash memory failure but the incorrect use of flash memory. If bit5 is set to "1" as described above, execute a reset command.

17.4.4 Toggle Bit-2 Flag (DQ2)

The toggle bit-2 flag (DQ2) is used to detect that flash memory is performing an automatic erase operation, together with the toggle bit.

■ Automatic Write/Erase

Making successive read accesses while the automatic 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 while the automatic writing algorithm is being performed toggles flash memory and makes it output 1 regardless of the specified address.

Making successive read accesses when the automatic writing/erasing algorithm ends makes flash memory to stop bit2 toggle and outputs the value of bit2 (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.

17.5 Detailed Explanation of Writing to Erasing Flash Memory

This section describes each operation procedure of flash memory Read/Reset, Write, Chip Erase, 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 17.3-1 in Section "17.3 Starting the Flash Memory Automatic Algorithm "for a write cycle to the bus to perform Read/Reset, Write, Chip Erase 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:

17.5.1 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 17.3-1 in Section "17.3 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.

17.5.2 Writing Data

This section describes the procedure for issuing the Write command to write data to the flash memory. Figure 17.5-1 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 17.3-1 in Section "17.3 Starting the Flash Memory Automatic Algorithm ") continuously to 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

Writing can be done in any order of addresses. 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

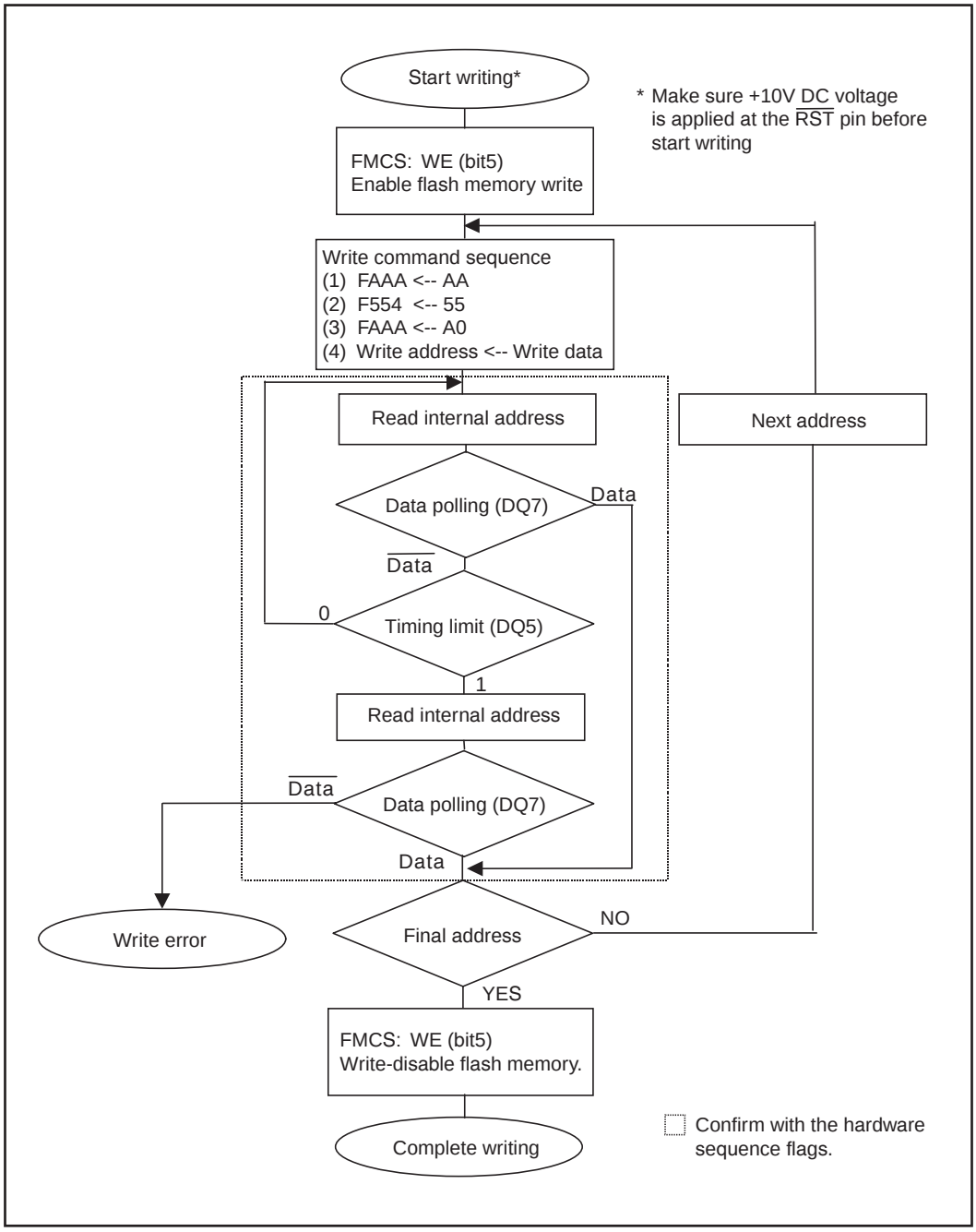
Figure 17.5-1 is an example of the procedure for writing to the flash memory. The hardware sequence flags (see Section "17.4 Confirming the Automatic Algorithm Execution State ") can be used to determine the state of the automatic algorithm in the flash memory. Here, the data polling flag (DQ7) is used 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 17.5-1 Example of the Flash Memory Write Procedure



17.5.3 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 17.3-1 in Section "17.3 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.

17.6 Flash Security Feature

Flash security feature provides possibilities to protect the content of the flash memory from being read from external.

■ Abstract

By writing the protection code of "01_H" to the predefined flash security address of the flash memory, access to the flash memory is restricted. Once the flash memory is protected, unlock the security function can only be done by performing the chip erase operation. Otherwise, read/write access of the flash memory from the external pins is not possible. This function is suitable for applications requiring security of self-containing data stored in the flash memory.

Table 17.6-1 Flash Security Address

Product	Flash Memory Size	Flash Security Address
MB89F202/F202RA	16 Kbyte	$FFFC_H$

■ How to enable the Flash Security Feature

After writing the code "01_H" to the flash security address, the subsequent external reset or power on enables the flash security feature.

■ How to disable the Flash Security Feature

Perform the chip erase operation.

■ Behavior under the Flash Security Feature

Read operation: invalid data read

Write operation: ignored

■ Others

- For the configuration of the standard parallel programmer, please follow the specification of parallel programmer.
- In order to prevent the device form enabling the flash security feature accidentally, writing the protection code at the last of flash memory programming is recommended.

Note:

The security byte is allocated inside the flash memory. After writing the code "01_H" to the flash security address, the subsequent external reset or power on enables the flash security feature. Therefore, if the flash security feature is not required, do not write "01_H" to the security byte address.

Once the flash security feature is enabled, all the flash memory failure analysis cannot be performed.

17.7 Notes on using Flash Memory

This section provides notes on using the MB89F202, especially for flash memory.

■ Input of a Hardware Reset ($\overline{RST}$)

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 1650 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 1650 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 or shut-down 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 cancelled, 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 the flash 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.

■ Flash Content Protection

Flash content can be read using parallel / serial programmer if the flash content protection mechanism is not activated.

One predefined area of the flash ($FFFC_H$) is assigned to be used for preventing the read access of flash content. If the protection code "01_H" is written in this address ($FFFC_H$), the flash content cannot be read by any parallel / serial programmer.

Note : The program written into the flash cannot be verified once the flash protection code is written ("01_H" in $FFFC_H$). It is advised to write the flash protection code at last.

APPENDIX

This appendix shows the I/O map, the overview of the instructions, mask options in MB89202/F202RA series, and the pin states.

APPENDIX A I/O Map

For the registers of peripheral functions incorporated in the MB89202/F202RA series, the addresses shown in Table A-1 are assigned.

■ I/O Map

Table A-1 I/O Map (1 / 4)

Address	Register abbreviation	Register name	Read/write	Initial value
0000_H	PDR0	Port 0 data register	R/W	XXXXXXXX
0001_H	DDR0	Port 0 data direction register	W	00000000
0002_H to 0006_H	Vacancy			
0007_H	SYCC	System clock control register	R/W	1--11100
0008_H	STBC	Standby control register	R/W	00010---
0009_H	WDTC	Watchdog control register	R/W	0---XXXX
$000A_H$	TBTC	Time-base timer control register	R/W	00---000
$000B_H$	Vacancy			
$000C_H$	PDR3	Port 3 data register	R/W	XXXXXXXX
$000D_H$	DDR3	Port 3 data direction register	W	00000000
$000E_H$	RSFR	Reset flag register	R	XXXX----
$000F_H$	PDR4	Port 4 data register	R/W	----XXXX
0010_H	DDR4	Port 4 data direction register	R/W	----0000
0011_H	OUT4	Port 4 output format register	R/W	----0000
0012_H	PDR5	Port 5 data register	R/W	-------X
0013_H	DDR5	Port 5 data direction register	R/W	-------0
0014_H	RCR21	12-bit PPG control register 1	R/W	00000000
0015_H	RCR22	12-bit PPG control register 2	R/W	--000000
0016_H	RCR23	12-bit PPG control register 3	R/W	0-000000
0017_H	RCR24	12-bit PPG control register 4	R/W	--000000

Table A-1 I/O Map (2 / 4)

Address	Register abbreviation	Register name	Read/write	Initial value
0018_H	BZCR	Buzzer register	R/W	-----000
0019_H	TCCR	Capture control register	R/W	00000000
$001A_H$	TCR1	Timer 1 control register	R/W	000-0000
$001B_H$	TCR0	Timer 0 control register	R/W	00000000
$001C_H$	TDR1	Timer 1 data register	R/W	XXXXXXXX
$001D_H$	TDR0	Timer 0 data register	R/W	XXXXXXXX
$001E_H$	TCPH	Capture data register H	R	XXXXXXXX
$001F_H$	TCPL	Capture data register L	R	XXXXXXXX
0020_H	TCR2	Timer output control register	R/W	------00
0021_H	Vacancy			
0022_H	CNTR	PWM control register	R/W	0-000000
0023_H	COMR	PWM compare register	W	XXXXXXXX
0024_H	EIC1	External interrupt 1 control register 1	R/W	00000000
0025_H	EIC2	External interrupt 1 control register 2	R/W	----0000
0026_H	Vacancy			
0027_H				
0028_H	SMC	Serial mode control register	R/W	00000-00
0029_H	SRC	Serial rate control register	R/W	--011000
$002A_H$	SSD	Serial status and data register	R/W	00100-1X
$002B_H$	SIDR	Serial input data register	R	XXXXXXXX
	SODR	Serial output data register	W	XXXXXXXX
$002C_H$	UPC	Clock divided cycle selection register	R/W	----0010
$002D_H$ to $002F_H$	Vacancy			
0030_H	ADC1	A/D control register 1	R/W	-0000000
0031_H	ADC2	A/D control register 2	R/W	-0000001
0032_H	ADDH	A/D data register H	R	------XX
0033_H	ADDL	A/D data register L	R	XXXXXXXX

Table A-1 I/O Map (3 / 4)

Address	Register abbreviation	Register name	Read/write	Initial value
0034_H	ADEN	A/D enable register	R/W	00000000
0035_H	Vacancy			
0036_H	EIE2	External interrupt 2 control register 1	R/W	00000000
0037_H	EIF2	External interrupt 2 control register 2	R/W	-------0
0038_H	Vacancy			
0039_H	SMR	Serial mode register	R/W	00000000
$003A_H$	SDR	Serial data register	R/W	XXXXXXXX
$003B_H$	SSEL	Serial function switching register	R/W	-------0
$003C_H$ to $003F_H$	Vacancy			
0040_H	WRARH0	Higher address set register 0	R/W	XXXXXXXX
0041_H	WRARL0	Lower address set register 0	R/W	XXXXXXXX
0042_H	WRDR0	Data setting register 0	R/W	XXXXXXXX
0043_H	WRARH1	Higher address set register 1	R/W	XXXXXXXX
0044_H	WRARL1	Lower address set register 1	R/W	XXXXXXXX
0045_H	WRDR1	Data setting register 1	R/W	XXXXXXXX
0046_H	WREN	Address comparison EN register	R/W	------00
0047_H	WROR	Wild register data test register	R/W	------00
0048_H to $005F_H$	Vacancy			
0060_H	PDR6	Port 6 data register	R/W	------XX
0061_H	DDR6	Port 6 data direction register*	R/W	------00
0062_H	PUL6	Port 6 pull-up set register	R/W	------00
0063_H	PDR7	Port 7 data register	R/W	-----XXX
0064_H	DDR7	Port 7 data direction register	R/W	-----000
0065_H	PUL7	Port 7 pull-up set register	R/W	-----000

Table A-1 I/O Map (4 / 4)

Address	Register abbreviation	Register name	Read/write	Initial value
0066_H to $006F_H$	Vacancy			
0070_H	PUL0	Port 0 pull-up set register	R/W	00000000
0071_H	PUL3	Port 3 pull-up set register	R/W	00000000
0072_H	PUL5	Port 5 pull-up set register	R/W	-------0
0073_H to 0078_H	Prohibited area			
0079_H	FMCS	Flash memory control status register	R/W, R	000X----
$007A_H$	Prohibited area			
$007B_H$	ILR1	Interrupt level set register 1	W	11111111
$007C_H$	ILR2	Interrupt level set register 2	W	11111111
$007D_H$	ILR3	Interrupt level set register 3	W	11111111
$007E_H$	ILR4	Interrupt level set register 4	W	11111111
$007F_H$	ITR	Interrupt test register	Inhibited	------00

- Explanation on read/write

  R/W: Readable and Writable

  R: Read only

  W: Write only

- Explanation on initial value

  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.

  *: No used in MB89F202/F202RA.

  Note:

  Do not use the prohibited areas.

APPENDIX B Overview of the Instructions

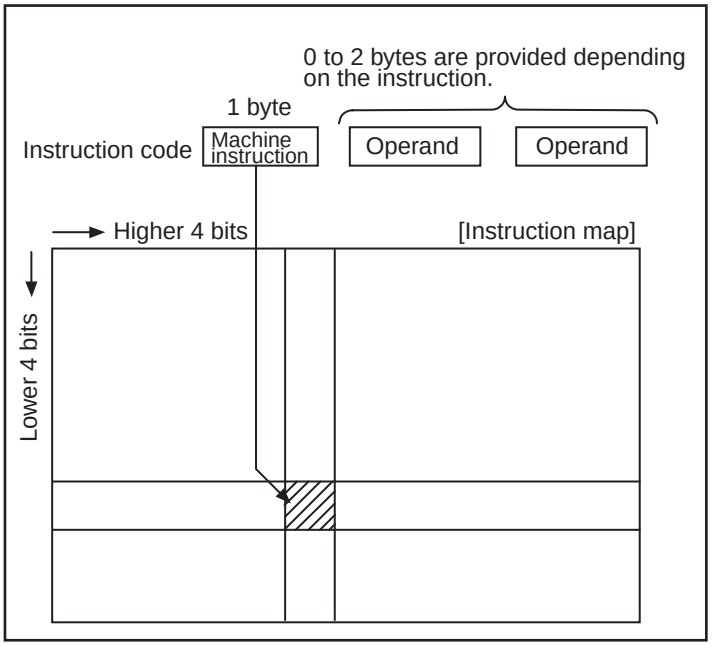
This section describes the instructions used for the F^{2}MC-8L.

■ Overview of the Instructions of the F^{2}MC-8L

The F^{2}MC-8L has 140 kinds of 1-byte machine instructions (actually, the map is 256 bytes). An instruction and succeeding operands make an instruction code.

Figure B-1 shows the correspondence between the instruction codes and instruction map.

Figure B-1 Correspondence between Instruction Codes and Instruction Map



- The instructions are classified into four groups including transfer instructions and branch instructions.
- Various methods for addressing are supported. Depending on the selection of an instruction and specification of operands, 10 kinds of addressing can be selected.
- Bit manipulation instructions are supported, so read-modify-write operation is possible.
- Instructions directing special operations are supported.

■ Explanation on the Codes Representing Instructions

Table B-1 describes the codes used to explain the instruction codes in Appendix B.

Table B-1 Explanation on Codes on Instructions' List

Code	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:b	Bit direct address (8:3 bits)
rel	Branch relative address (8 bits)
@	Register indirect (Example: @A, @IX, and @EP)
A	Accumulator (8 bits or 16 bits, determined on basis of instruction to be used)
AH	Higher 8 bits of the accumulator (8 bits)
AL	Lower 8 bits of the accumulator (8 bits)
T	Temporary accumulator (8 bits or 16 bits, determined on basis of instruction to be 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	Indicates that X itself is the immediate data. (8 bits or 16 bits, determined on basis of instruction to be used)
(X)	Indicates that the contents of X is the accessing subject. (8 bits or 16 bits, determined on basis of instruction to be used)
((X))	Indicates that the address specified in X is the accessing subject. (8 bits or 16 bits, determined on basis of instruction to be used)

■ Explanation on the Items of Instructions' List

Table B-2 Explanation on Items of Instructions' List

Item	Description
MNEMONIC	Represents the instruction coded in the assembler.
~	Indicates the number of cycles of the instruction (number of instruction cycles).
#	Indicates the number of bytes of the instruction.
Operation	Indicates the operation of the instruction.
TL,TH,AH	Indicates how the contents of TL, TH, and AH change (automatic transfer from A to T) when the instruction is executed. The codes in this column indicate the following: • - indicates no change. • dH indicates the higher 8 bits of the data coded for the operation. • AL and AH indicate the contents of AL and AH just before the execution of the instruction. • 00 indicates that it becomes 00.
N,Z,V,C	Indicates whether the instruction changes the corresponding flags. If (is shown in this column, the instruction changes the corresponding flags.
OP CODE	Indicates the instruction code. If the appropriate instruction occupies multiple codes, they are listed under the following rule: Example: 48 to 4F means the serial numbers from 48 to 4F.

B.1 Addressing

For the F^{2}MC-8L, the following 10 kinds of addressing modes are supported:

- **Direct addressing**
- **Extended addressing**
- **Bit direct addressing**
- **Index addressing**
- **Pointer addressing**
- **General-purpose register addressing**
- **Immediate addressing**
- **Vector addressing**
- **Relative addressing**
- **Inherent addressing**

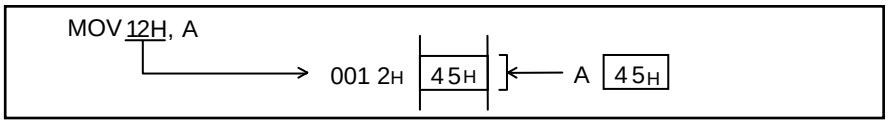
■ Explanation on Addressing

● Direct addressing

The addressing, which is indicated by dir in the instructions list, is used for accessing the area from 0000_H to $00FF_H$. In this addressing, the higher one byte of the address is 00_H. Specify the lower one byte with the operand.

Figure B.1-1 shows an example.

Figure B.1-1 Example of Direct Addressing

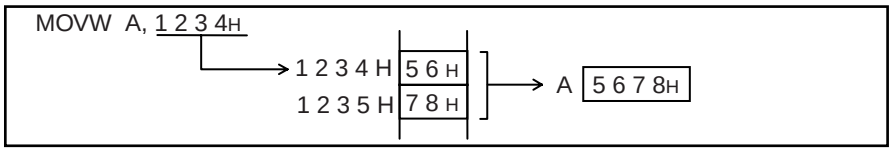


● Extended Addressing

The addressing, which is indicated by ext in the instructions list, is used for accessing the entire area of 64 KB. In this addressing, specify the higher one byte of the address with the first operand and the lower one byte with the second operand.

Figure B.1-2 shows an example.

Figure B.1-2 Example of Extended Addressing

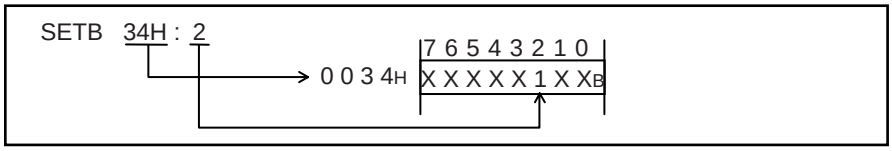


● Bit Direct Addressing

The addressing, which is indicated by dir:b in the instructions list, is used for accessing the area from 0000_H to $00FF_H$ on a per bit basis. In this addressing, the higher one byte of the address is 00_H. Specify the lower one byte with the operand; and the bit position in the specified address with the lower three bits of the operation code.

Figure B.1-3 shows an example.

Figure B.1-3 Example of Bit Direct Addressing

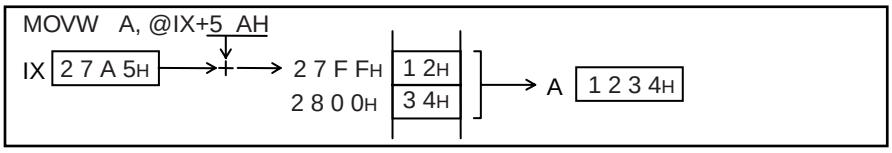


● Index addressing

The addressing, which is indicated by @IX(off in the instructions list, is used for accessing the entire area of 64 KB. In this addressing, the contents of the first operand are signed and added to IX (index register). Then the results are used as the address.

Figure B.1-4 shows an example.

Figure B.1-4 Example of Index Addressing

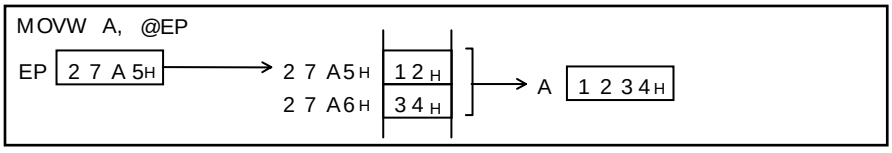


● Pointer Addressing

The addressing, which is indicated by @EP in the instructions list, is used for accessing the entire area of 64 KB. In this addressing, the contents of EP (extra pointer) are used as the address.

Figure B.1-5 shows an example.

Figure B.1-5 Example of Pointer Addressing

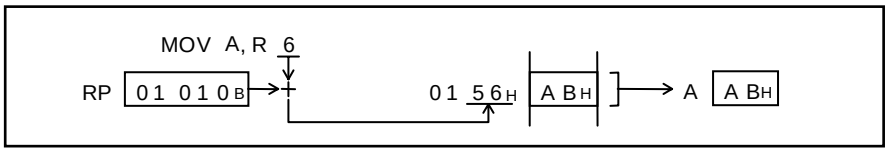


● General-purpose Register Addressing

The addressing, which is indicated by Ri in the instructions list, is used for accessing the register bank of the general-purpose register area. In this addressing, the higher one byte of the address is fixed to 01. The lower one byte is generated from the contents of RP (register bank pointer) and the lower three bits of the operation code. The address is then accessed.

Figure B.1-6 shows an example.

Figure B.1-6 Example of General-purpose Register Addressing

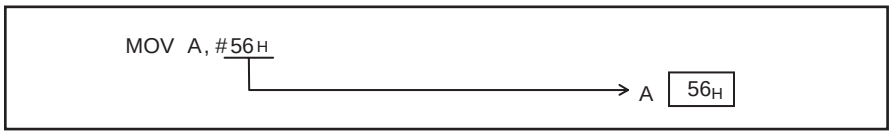


● Immediate Addressing

The addressing, which is indicated by #d8 in the instructions list, is used when immediate data is required. In this addressing, the operand directly becomes the immediate data. The specification of byte/word is determined using the operation code.

Figure B.1-7 shows an example.

Figure B.1-7 Example of Immediate Addressing



● Vector Addressing

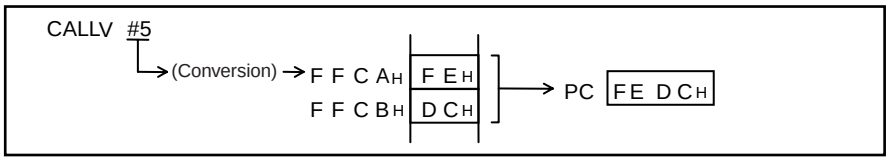
The addressing, which is indicated by vct in the instructions list, is used for branching to a subroutine registered in the table. In this addressing, the operation code includes the vct information, with the addresses generated on the basis of the correspondence with the contents of Table B.1-1 .

Table B.1-1 Vector Table Address Corresponding to vct

#vct	Vector table address (Jump destination, higher address:lower address)
0	$FFCO_H$: $FFC1_H$
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.1-8 shows an example.

Figure B.1-8 Example of Vector Addressing

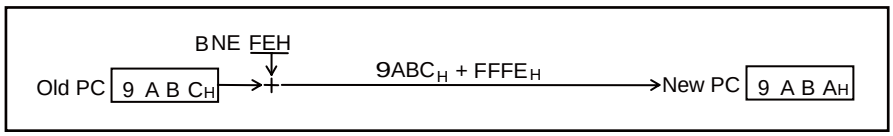


- Relative Addressing

The addressing, which is indicated by rel in the instructions list, is used for branching to the area of 128 bytes before or after the PC (program counter). In this addressing, the contents of the operand with a sign are added to the PC. The results are then stored in the PC.

Figure B.1-9 shows an example.

Figure B.1-9 Example of Relative Addressing



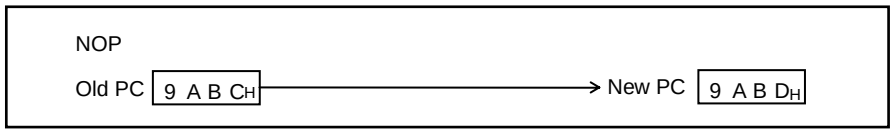
In this example, the control jumps to the address holding the operation code of BNE, causing an endless loop.

- Inherent addressing

This addressing, which has no operand in the instructions list, is used for performing an operation determined on the basis of the operation code. In this addressing, the operations differ depending on the instructions.

Figure B.1-10 shows an example.

Figure B.1-10 Example of Inherent Addressing



B.2 Special Instructions

This section describes the special instructions other than addressing.

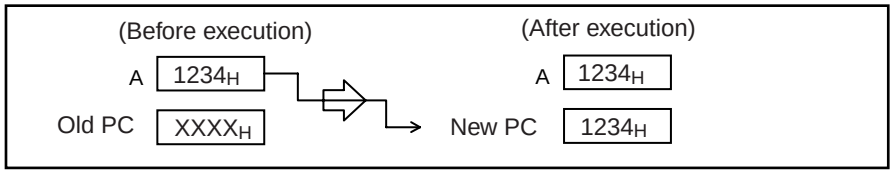
■ Special Instructions

● JMP @A

By this instruction, the control branches to PC (program counter) using the contents of A (accumulator) as the address. N items of jump destinations have been listed on the table, one of which is selected and transferred to A. Executing this instruction can achieve N kinds of branch processing.

Figure B.2-1 shows an overview.

Figure B.2-1 JMP @A

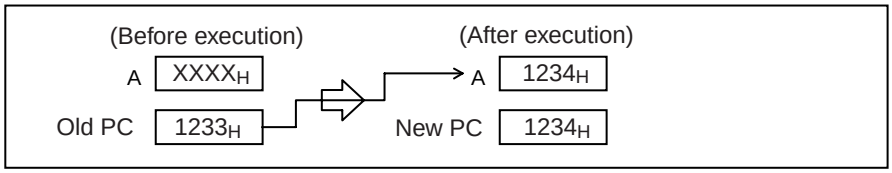


● MOVW A, PC

This instruction performs the opposite operation of JMP @A. In other words, the contents of the PC are stored in A. When this instruction has been executed in the main routine and a specific subroutine is to be called, it is possible to verify that the contents of A are the predetermined value in the subroutine. It is also possible to verify that the branch was not from an unexpected part, so it is useful in judging that a runaway has occurred.

Figure B.2-2 shows an overview.

Figure B.2-2 MOVW A, PC



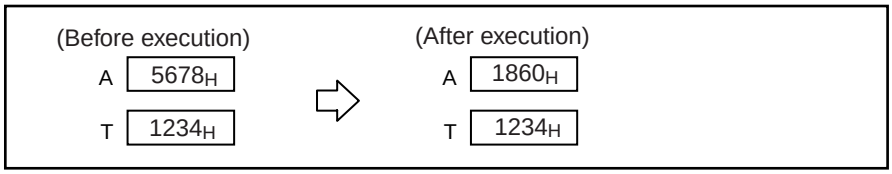
When this instruction is executed, the contents of A are not the address holding the operation code of this instruction but the same value as the address holding the next instruction. In Figure B.2-2 , therefore, the value stored in A, 1234_H, is the same as the address holding the operation code next to the MOVW A, PC.

- MULU A

This instruction multiplies AL (the lower eight bits of accumulator) by TL (the lower eight bits of the temporary accumulator) without a sign and stores the results in 16 bits length to A. The contents of T (temporary accumulator) remain as they are. For the operation, the contents of AH (the higher eight bits of accumulator) and TH (the higher eight bits of temporary accumulator) before the execution are not used. Take care when using a branch based on the result of multiplication because the flags were not changed.

Figure B.2-3 shows an overview.

Figure B.2-3 MULU A



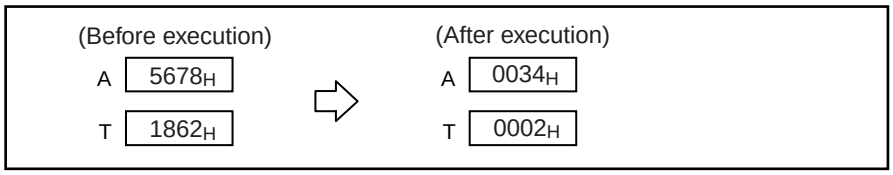
- DIVU A

This instruction divides T of 16 bits by AL of 8 bits without a sign, stores the results in 8 bits to AL, and stores the remainder of 8 bits to TL. Both AH and TH become 0. For the operation, the contents of AH before execution are not used. If the results exceed 8 bits, they are not guaranteed. Also, the fact that the results exceeded 8 bits is not indicated. So when using data units that may cause this type of situation, judge them in advance.

Take care when using a branch based on the result of division, because the flags were not changed.

Figure B.2-4 shows an overview.

Figure B.2-4 DIVU A

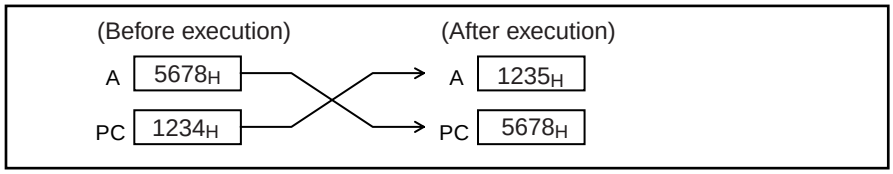


● XCHW A, PC

This instruction replaces the contents of A and the contents of PC, resulting in a branch to the address indicated by the contents of A before execution. The contents of A after execution become the value of the address next to the address holding the operation code, XCHW A, PC. This instruction is useful especially when a table is specified in the main routine and a subroutine uses it.

Figure B.2-5 shows an overview.

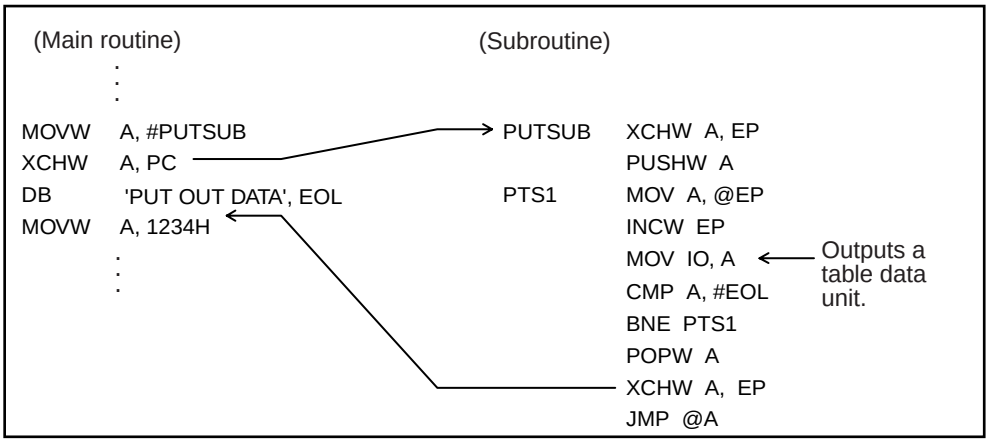
Figure B.2-5 XCHW A, PC



After execution of this instruction, the contents of A do not become the address holding the operation code of this instruction. Instead, they are the same as the address holding the next instruction. In Figure B.2-5 , therefore, the value stored in A is 1235_H, agreeing with the address holding the operation code next to XCHW A, PC. Note that it is not 1234_H but 1235_H.

Figure B.2-6 shows an example of assembler coding.

Figure B.2-6 Usage Example of XCHW A, PC

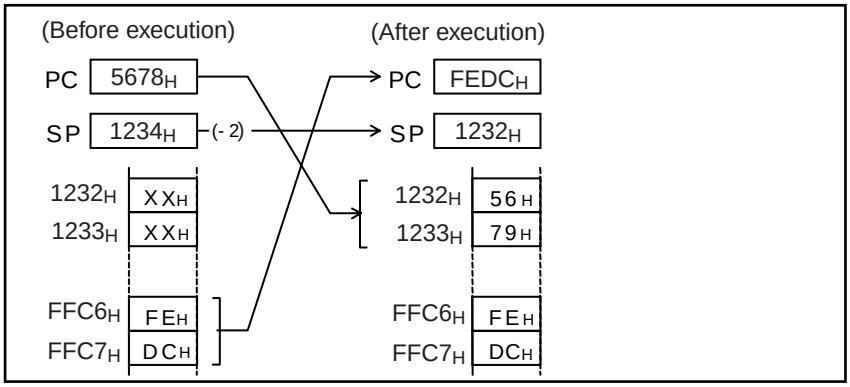


● CALLV #vct

This addressing is used for branching to one of the subroutine addresses registered in the table. After the return address (the contents of PC) is saved to the address indexed by SP (stack pointer), the control is branched to the address listed in the vector table via the vector addressing. This instruction is one byte, so using it for the frequently used subroutines enables the entire program size to be smaller.

Figure B.2-7 shows the overview.

Figure B.2-7 Executing Example of CALLV #3



When this instruction is executed, the contents of the PC to be saved in the stack area are not the address holding the operation code of this instruction. Instead, they comprise the address holding the next instruction. In Figure B.2-7 , therefore, the value saved in the stack (1232_H and 1233_H) is the same as the address holding the operation code next to CALLV #vct (return address), i.e. 5679_H.

B.3 Bit Manipulation Instructions (SETB and CLRB)

Some registers of peripheral functions have bits that perform a read operation different from ordinary read for a bit manipulation instruction.

■ Read-modify-write Operation

The bit manipulation instructions can set "1" (SETB) to the specified bit in a register or RAM or clear it to "0" (CLRB). Because the CPU handles the data in 8 bits, however, it actually reads the 8-bit data, modifies the specified bit, and then writes it back to the original address. This series of operations is called read-modify-write operation.

Table B.3-1 shows the bus operation at bit manipulation instructions.

Table B.3-1 Bus Operation at Bit Manipulation Instructions

CODE	MNEMONIC	~	Cycle	Address bus	Data bus	$\overline{RD}$	$\overline{WR}$	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 operation	0	1	0

■ Read Destination at Execution of a Bit Manipulation Instruction

For some I/O ports and interrupt request flag bits, the read destination for read-modify-write is different from that for ordinary read.

- I/O port (at bit manipulation)

  For some I/O ports, the value of the I/O pin is read at ordinary read; meanwhile, the value of output latch is read at bit manipulation. This is to prevent the other bits of the output latch from being accidentally changed regardless of the I/O direction and pin state.

- Interrupt request flag bit (at bit manipulation)

  The interrupt request flag bits work as flag bits for confirming an interrupt request at ordinary read; meanwhile, "1" is always read at bit manipulation. This is to prevent the interrupt request flag bits from being written as "0", and accidentally clearing the flags at bit manipulation for another bit.

B.4 F²MC-8L Instructions List

Table B.4-1 to Table B.4-4 list the instructions used by the F²MC-8L.

■ Transfer Instructions

Table B.4-1 List of Transfer Instructions (1 / 2)

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	-	-	-	-	-	-	-	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

Table B.4-1 List of Transfer Instructions (2 / 2)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
23	MOVW A, dir	4	2	(AH) ←(dir), (AL) ←(dir+1)	AL	AH	dH	+	+	-	-	C5
24	MOVW A, @IX+off	5	2	(AH) ←((IX) +off), (AL) ←((IX)	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:

At byte transfer operation to A, the automatic transfer to T is represented by TL ←AL.

The operands in a multiple-operand instruction are stored in the order in which they are indicated in MNEMONIC.

■ Operation Instructions

Table B.4-2 List of Operation Instructions (1 / 4)

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

Table B.4-2 List of Operation Instructions (2 / 4)

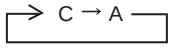
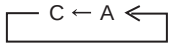
No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
25	ANDW A	3	1	(A) ← (A) ∧ (T)	-	-	dH	+	+	R	-	63
26	ORW A	3	1	(A) ← (A) ∨ (T)	-	-	dH	+	+	R	-	73
27	XORW A	3	1	(A) ← (A) ∀ (T)	-	-	dH	+	+	R	-	53
28	CMP A	2	1	(TL) - (AL)	-	-	-	+	+	+	+	12
29	CMPW A	3	1	(T)- (A)	-	-	-	+	+	+	+	13
30	RORC A	2	1		-	-	-	+	+	-	+	03
31	ROLC A	2	1		-	-	-	+	+	-	+	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

Table B.4-2 List of Operation Instructions (3 / 4)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
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) ∨ (Ri)	-	-	-	+	+	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

Table B.4-2 List of Operation Instructions (4 / 4)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
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.4-3 List of 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.4-4 List of Other Instructions

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
1	PUSHW A	4	1	((SP)) ←(A), (SP) ←(SP)-2	-	-	-	-	-	-	-	40
2	POPW A	4	1	(A) ←((SP)), (SP) ←(SP)+2	-	-	dH	-	-	-	-	50
3	PUSHW IX	4	1	((SP)) ←(IX), (SP) ←(SP)-2	-	-	-	-	-	-	-	41
4	POPW IX	4	1	(IX) ←((SP)), (SP) ←(SP)+2	-	-	-	-	-	-	-	51
5	NOP	1	1	No operation	-	-	-	-	-	-	-	00
6	CLRC	1	1	(C) ←0	-	-	-	-	-	-	R	81
7	SETC	1	1	(C) ←1	-	-	-	-	-	-	S	91
8	CLRI	1	1	(I) ←0	-	-	-	-	-	-	-	80
9	SETI	1	1	(I) ←1	-	-	-	-	-	-	-	90

B.5 Instruction Map

Table B.5-1 shows the instruction map of the F^{2}MC-8L.

■ Instruction Map

Table B.5-1 Instruction Map of the F^{2}MC-8L

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 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 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 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 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 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 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 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 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 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 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 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 dir:3	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 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 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 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 dir:7, rel	INC R7	DEC R7	CALLV #7	BLT rel

APPENDIX C Mask Options

Table C-1 lists the mask options of the MB89202/F202RA series.

■ Mask Options

Table C-1 Mask Options

No.	Part number	MB89202	MB89F202/F202RA	MB89V201
	Specifying procedure	Specify when ordering masking	Specify by part number	
1	Selection of initial value of main clock oscillation settling time* (with F_{CH} = 12.5 MHz) 01 : $2^{14}/F_{CH}$ (Approx.1.31 ms) 10 : $2^{17}/F_{CH}$ (Approx.10.5 ms) 11 : $2^{18}/F_{CH}$ (Approx.21.0 ms)	Selectable	Fixed to $2^{18}/F_{CH}$	Fixed to $2^{18}/F_{CH}$
2	Reset pin output With reset output Without reset output	Selectable	With reset output	With reset output
3	Power on reset selection With power-on reset Without power-on reset	Selectable	With power-on reset	With power-on reset

F_{CH} : Main clock scillation frequency

* : Initial value to which the oscillation settling time bit (SYCC : WT1, WT0) in the system clock control register is set

APPENDIX D Programming EPROM with Evaluation Chip

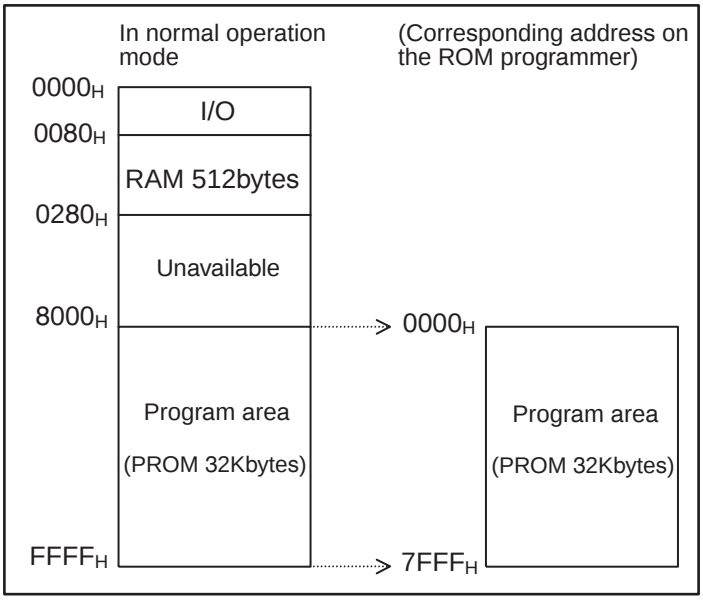
This section describes how to program EPROM with evaluation chip.

■ Programming EPROM with Evaluation Chip

- EPROM for use

  32 Kbyte EPROM (equivalent to MBM27C256A DIP-28)

Figure D-1 Memory Map of the Evaluation Chip



- Programming EPROM

  1. Make the MBM27C256-equivalent setting for the EPROM programmer
  2. Load the program data to the area from 0000_H to $7FFF_H$ of the EPROM programmer.
  3. Program the area from 0000_H to $7FFF_H$ with the EPROM programmer.

APPENDIX E Pin State of the MB89202/F202RA Series

Table E-1 describes the pin states in each operation mode of the MB89202/F202RA series.

■ Pin States in Each Operation Mode

Table E-1 Pin States in Each Operation Mode

Pin name	In normal operation mode	In sleep mode	In stop mode (SPL = 0)	In stop mode (SPL = 1)	During a reset
X0	Oscillation input	Oscillation input	Hi-Z	Hi-Z	Oscillation input
X1	Oscillation output	Oscillation output	"H" output	"H" output	Oscillation output
P00/INT20/AN4 to P07/INT27	Port I/O or resource I/O	Hold	Hold	Hi-Z *1,*2	Hi-Z
P30/UCK/SCK to P37/BZ/PPG	Port I/O or resource I/O	Hold	Hold	Hi-Z *1,*2	Hi-Z
P40/AN0 to P43/AN3	Port I/O or resource I/O	Hold	Hold	Hi-Z *2	Hi-Z
P50/PWM	Port I/O or resource I/O	Hold	Hold	Hi-Z *2	Hi-Z
P60, P61	Port I/O	Hold	Hold	Hi-Z *2	Hi-Z
P70 to P72	Port I/O	Hold	Hold	Hi-Z *2	Hi-Z

*1: For port input and peripheral input, the internal input level is fixed to prevent them from generating a leak via the input open. However, if external interrupts are allowed for P00 to P07 and P34 to P36, only the external interrupts are available as their inputs.
*2: The pins, for which pull-up is selected by the option setting, enter the pull-up state.
Hi-Z: Indicates high impedance.
Hold: The pins, for which output is set, maintain the pin state (level) just before the mode transition.
SPL: Pin state spacification bit of the standby control register (STBC)

INDEX

The index follows on the next page.
This is listed in alphabetic order.

Index

A

B

C

D

E

F

G

H

L

M

P

R

S

T

U

V

W

CM25-10153-2E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

F^2MC-8L
8-BIT MICROCONTROLLER
MB89202/F202RA Series
HARDWARE MANUAL

February 2008 the second edition

Published **FUJITSU LIMITED** Electronic Devices

Edited Strategic Business Development Dept.