FUJITSU SEMICONDUCTOR
CONTROLLER MANUAL

CM25-10141-1E

F²MC-8L

8-BIT MICROCONTROLLER

MB89980 Series

HARDWARE MANUAL

FUJITSU

F^2MC-8L

32-BIT MICROCONTROLLER

MB89980 Series

HARDWARE MANUAL

FUJITSU LIMITED

PREFACE

■ Objective and Intended Readership of This Manual

The MB89980 series has been developed as a general-purpose version of the F^2MC*-8L family consisting of proprietary 8-bit, single-chip microcontrollers. The MB89980 series is applicable to a wide range of applications from welfare products to industrial equipment, including portable devices.

This manual describes the functions and operation of the MB89980 series and is aimed at engineers using the MB89980 series of microcontrollers to develop actual products. See the F^2MC-8L MB89600 Series Programming Manual for details on the MB89980 instruction set.

*: F^2MC stands for FUJITSU Flexible Microcontroller.

■ Configuration of This Manual

This manual consists of the following 15 chapters:

CHAPTER 1 "OVERVIEW"

Provides an overview of the features and functions of the MB89980 series.

CHAPTER 2 "HANDLING DEVICES"

Describes points to note when using the MB89980 series.

CHAPTER 3 "CPU"

Describes the functions of the MB89980 series CPU.

CHAPTER 4 "I/O PORTS"

Describes the function and operation of the MB89980 series I/O ports.

CHAPTER 5 "TIMEBASE TIMER"

Describes the function and operation of the MB89980 series timebase timer.

CHAPTER 6 "WATCHDOG TIMER"

Describes the functions and operation of the MB89980 series watchdog timer.

CHAPTER 7 "8-BIT PWM TIMER (PWM TIMER 1, PWM TIMER 2)"

Describes the functions and operation of the MB89980 series 8-bit PWM timer.

CHAPTER 8 "8/16-BIT TIMER/COUNTER"

Describes the functions and operation of the MB89980 series 8/16-bit timer/counter.

CHAPTER 9 "EXTERNAL INTERRUPT CIRCUIT 1 (EDGE-TRIGGERED)"

Describes the functions and operation of the MB89980 series external interrupt circuit 1 (edge-triggered interrupt).

CHAPTER 10 "EXTERNAL INTERRUPT CIRCUIT 2 (LEVEL-TRIGGERED)"

Describes the functions and operation of the MB89980 series external interrupt circuit 2 (level-triggered interrupt).

CHAPTER 11 "A/D CONVERTER"

Describes the functions and operation of the MB89980 series A/D converter.

CHAPTER 12 "WATCH PRESCALER"

Describes the functions and operation of the MB89980 series watch prescaler.

CHAPTER 13 "REMOTE CONTROL GENERATOR (6-BIT PPG)"

Describes the functions and operation of the MB89980 series remote control transmission output.

CHAPTER 14 "LCD CONTROLLER-DRIVER"

Describes the functions and operation of the MB89980 series liquid crystal display (LCD) controller-driver circuit.

CHAPTER 15 "BUZZER OUTPUT"

Describes the function and operation of the MB89980 series audible alarm output.

APPENDIX

The appendices include the I/O map, mask options, instruction summary, instruction list, and instruction map.

1. The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering.
2. The information and circuit diagrams in this document are presented as examples of semiconductor device applications, and are not intended to be incorporated in devices for actual use. Also, FUJITSU is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of this information or circuit diagrams.
3. The contents of this document may not be reproduced or copied without the permission of FUJITSU LIMITED.
4. FUJITSU semiconductor devices are intended for use in standard applications (computers, office automation and other office equipments, industrial, communications, and measurement equipments, personal or household devices, etc.).
   CAUTION:
   Customers considering the use of our products in special applications where failure or abnormal operation may directly affect human lives or cause physical injury or property damage, or where extremely high levels of reliability are demanded (such as aerospace systems, atomic energy controls, sea floor repeaters, vehicle operating controls, medical devices for life support, etc.) are requested to consult with FUJITSU sales representatives before such use. The company will not be responsible for damages arising from such use without prior approval.
5. Any semiconductor devices have inherently a certain rate 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.
6. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Control Law of Japan, the prior authorization by Japanese government should be required for export of those products from Japan.

READING THIS MANUAL

■ Page Layout

In this manual, an entire section is presented on a single page or spread whenever possible. The reader can thus view a section without having to flip pages.

The content of each section is summarized immediately below the title. You can obtain a rough overview of this product by reading through these summaries.

Also, higher level section headings are given in lower sections so that you can know to which section the text you are currently reading belongs.

■ Finding Information

In addition to the standard table of contents and index, the following methods are available to find information in a particular section when required.

❍ Register index

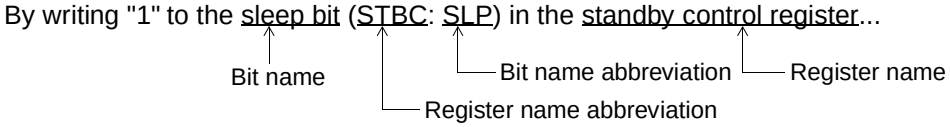
Information can be looked up in the register index by register name, by bit name, and by their respective abbreviations.

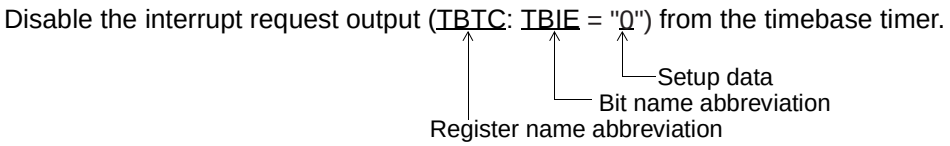
❍ Subheading index

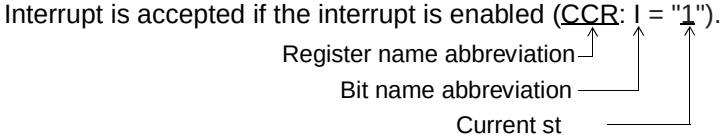
The sub-headings in each section (lines that start with n) are collected together in the subheading index. The subheading index provides a means of looking up information at a finer level of detail than the table of contents.

■ Naming Conventions for Register Name and Pin Name

❍ Example for description of register name and bit name







❍ **Notations for shared pins**

Pin P25/SCK

Many of the pins in the devices of this series are multi-function pins. (They can be switched between two or more functions under program control.) The multiple names of these pins (indicating their multiple functions) are separated by slant bars (/).

■ Development Tools and Other Resources Required for Development

The following items are required for developing using the MB89980 series.

Contact FUJITSU sales staff for the required development tools and other resources.

❍ **Manuals required for development**

Checklist

- ❑ • *F²MC-8L MB89980 Series Data Sheet*
  (Provides electrical characteristics and various characteristic examples for the device.)
- ❑ • *F²MC-8L MB89980 Series Hardware manual*
- ❑ • *F²MC-8L MB89600 Series Programming Manual*
  (Describes the F²MC-8L series instruction set.)
- ∗ • *F²MC-8L MB89600 Series C Compiler Manual*
  (Only required when developing in C.)
  (Describes program development in C and how to run the compiler.)
- ∗ • *F²MC-8L MB89600 Series Assembler Manual*
  (Describes program development in assembly language.)
- ∗ • *F²MC-8L MB89600 Series Support System Manual*
  (Describes how to run the macro assembler, linker, and library manager.)
- ∗ • *F²MC-8L MB89600 Series Software Simulator Manual*
  (Only required when performing evaluation using the simulator.)
  (Describes how to operate the software simulator.)

Manuals marked with * are provided with the products.

In addition, manuals for products such as development tools are provided with the product.

❍ **Software required for development**

Checklist

- ❑ • C compiler (Only required when developing in C.)
- ❑ • Assembler, linker, librarian
- ❑ • Software simulator (Only required when performing evaluation using the simulator.)
- ❑ • Emulator/debugger (Only required when performing evaluation using the MB2140A series.)

The model number for each software package differs depending on the operating system.

See the F²MC development tools catalog or product guide for details.

❍ **Items required for evaluation using one-time PROM or EPROM microcontrollers (when performing your own PROM or EPROM programming)**

Checklist

❑ • One of: MB89P985
❑ • ROM programmer (a programmer able to program an MBM27C256A)
See the data sheet for details of recommended programmers.
❑ • Package conversion adaptor for writing (available from Sun Hayato Co., Ltd.)

Package	Socket Model
FPT-64P-M03 (0.5 mm pitch)	ROM-64SQF-28DP-8L3
FPT-64P-M09 (0.65 mm pitch)	ROM-64QF2-28DP-8L4

❍ **Development tools**

Checklist

❑ • MB89PV980 (Piggyback/evaluation device)
❑ • Evaluation tools

(Main unit) (Pod) (Probe)
MB2141A + MB2144-505+MB2144-20

Check with the supplier when using a third party development environment.

❍ **Reference material**

- F^2MC development tools catalog
- Microcomputer product guide

CONTENTS

FIGURES

TABLES

CHAPTER 1 OVERVIEW

This chapter describes the main features and basic specifications of the MB89980 series.

1.1 MB89980 Series Features

The MB89980 series is a line of the general-purpose, single-chip microcontrollers. In addition to a compact instruction set, the microcontrollers contain a variety of peripheral functions such as an LCD controller/driver, an A/D converter, timers, PWM timers, remote control output, buzzer output and external interrupts.

■ MB89980 Series Features

❍ **Various package options**

- QFP package (0.65-mm lead pitch)
- LQFP package (0.5-mm lead pitch)

❍ **High speed processing at low voltage**

Minimum execution time: 0.95µs/4.2 MHz

❍ **F^{2}MC-8L family CPU core**

Instruction set optimized for controllers

- Multiplication and division instructions
- 16-bit arithmetic operations
- Test and branch instructions
- Bit manipulation instructions, etc.

❍ **Dual-clock control system**

- Main clock: max. 4.2 MHz (Four selectable speeds; oscillation stopped in subclock mode.)
- Subclock: 32.768 kHz (Operating clock used in subclock mode)

❍ **Six types of timers**

- 8-bit PWM timer 1 (also usable as an interval timer)
- 8-bit PWM timer 2 (also usable as an interval timer)
- 8/16-bit timer/counter (8 bits × 2 channels or 16 bits × 1 channel)
- 21-bit timebase timer
- Watch prescaler (15 bits)

❍ **A/D converter**

- Sense function enabling voltage comparison at 11.4 µs (at 4.2 MHz).
- Activation by an 8/16-bit timer/counter output capable

❍ **LCD controller/driver**

- 14 segments × 4 commons (max. 56 pixels).

❍ **Buzzer output**

- Output frequency: 1025 Hz, 2051 Hz, 4102 Hz, 8203 Hz/4.2 MHz for main clock
- 1024 Hz, 2048 Hz, 4096 Hz/32.768 kHz for subclock

❍ **Remote control transmission output**

- Program-selectable pulse width and period.

❍ **External interrupts (wake-up function)**

- External Interrupt 1 (4 channels)
  Four channels are independent and capable of wake-up from low-power consumption modes (with an edge detection function).
- External Interrupt 2 (8 inputs, 1 channel)
  Eight channels are independent and capable of wake-up from low-power consumption modes (with a "L" level detection function).

❍ **Standby modes (low power modes)**

- Stop mode (If in sub mode, oscillation stops to minimize the current consumption.)
- Sleep mode (The CPU stops to reduce the current consumption to approx. 1/3 of normal.)
- Watch mode (Everything except the watch prescaler stops to reduce the power consumption to an extremely low level.)

❍ **I/O ports: max. 47 channels**

- General-purpose I/O ports (N-ch open-drain): 8
- Output-only ports (N-ch open-drain): 20
- General-purpose I/O ports (CMOS): 16
- Input-only ports (CMOS): 2
- Output-only ports (CMOS): 1

1.2 MB89980 Series

The MB89980 series contains 3 type of products. Table 1.2-1 "MB89980 Series Product Lineup" lists the product lineup and Table 1.2-2 "MB89980 Series CPU and Peripheral Functions" lists the CPU and peripheral functions.

■ MB89980 Series Product Lineup

Table 1.2-1 MB89980 Series Product Lineup

Parameter	Part number		
	MB89983	**MB89P985**	**MB89PV980**
Classification	Mask ROM	OTP	Piggyback
ROM size	8K × 8 bits (Internal ROM)	16K × 8 bits (Internal ROM)	32K × 8 bits (External ROM)
RAM size	256 × 8 bits	512 × 8 bits	512 × 8 bits
Low-power consumption (stand by modes)	Sleep mode, stop mode, and watch mode		
Process	CMOS		
Operating voltage	2.7 V to 6.0 V		
	3.5 V to 6.0 V when using the A/D converter		

Table 1.2-2 MB89980 Series CPU and Peripheral Functions

Parameter		Specification
CPU funtions		Number of instructions: 136 Instruction bit length: 8 bits Instruction length: 1 to 3 bytes Data bit length: 1, 8, 16 bits Minimum execution time: 0.95 µs to 15.2 µs/4.2 MHz, 61.0 µs/32.768 kHz Interrupt processing time: 8.6 µs to 137.1 µs/4.2 MHz, 549.3 µs/32.768 kHz
Peripheral Functions	Ports	General-purpose I/O ports (N-ch open-drai) 8 (4 ports also serve as peripherals, 3 ports are heavy-current drive type.) Output-only ports (N-ch open-drain) 20 (4 ports also serve as A/D, 14 ports serve as segment pins and 2 ports serve as common pins, 10 ports are heavy-current drive) General-purpose I/O ports (CMOS) 16 (12 ports also serve as an external interrupt,) Input-only ports (CMOS) 2 (serve with sub-clock pins) Output-only ports (CMOS) 1 (serves as peripherials Total 47 (max.)
	21-bit timebase timer	21 bits Interrupt cycle: 1.95 ms, 7.80 ms, 62.41 ms, 998.64 ms/4.2 MHz for main clock
	Watchdog timer	Reset generate cycle: min. 998.6 ms/4.2 MHz for main clock min. 500 ms/32.768 kHz for subclock
	8-bit PWM timer 1, PWM timer 2	8-bit interval timer operation (square wave output capable, operating clock cycle: 0.95 µs to 249.7 ms) 8-bit resolution PWM operation (conversion cycle: 243.8 µs to 63.9 s) 8/16-bit timer/counter output for counter clock selectability
	8/16-bit timer/counter	Can be operated either as a 2-channel 8-bit timer/counter (Timer 1 and Timer 2, each with its own independent operating clock cycle), or as one 16-bit timer/counter (operating clock cycle: 1.90 µs to 487.6 µs) In Timer 1 or 16-bit timer/counter operation, event counter operation (external clock-triggered) and square wave output capable
	External interrupt 1 (wake-up function)	4 independent channels (interrupt vector, request flag, request output enable) Edge selectability (rising/falling) Used also for wake-up from stop/sleep mode. (Low-level detection is also permitted in stop mode.)
	External interrupt 2 (wake-up function)	8 inputs, one channel ("L" level interrupts, independent input enable). Used also for wake-up from stop/sleep mode. (Low-level detection is also permitted in stop mode.)
	8-bit A/D converter	8-bit resolution × 4 channels A/D conversion function (conversion time: 41.9 ∝s) Sense function (comparison time: 11.4 ∝s) Continuous activation by an 8/16-bit timer/counter output or a timebase timer output capable. Reference voltage input (AVR)
	Watch prescaler	15 bits Interrupt cycle: 31.25 ms, 0.25 s, 0.50 s, 1.00 s/32.768 kHz for subclock

Table 1.2-2 MB89980 Series CPU and Peripheral Functions

Parameter		Specification
	LCD controller/driver	Common output: 4 (max.) *1 Segment output: 14 (max.) *1 LCD driving power (bias) pins: 4 LCD display RAM size: 7 bytes (14 × 4 bits, max. 56 pixels)
	Buzzer output	Output frequency: 1025 Hz, 2051 Hz, 4102 Hz, 8203 Hz/4.2 MHz for main clock 1024 Hz, 2048 Hz, 4096 Hz/32.768 kHz for subclock
	Remote control transmit output	Internal 6-bit counter Pulse width ("H" level pulse width of 0 μs to 1920 μs) and cycle (0.95 μs to 1920 μs) are program selectable Can also be used as 6-bit PPG.

Note: Unless otherwise specified, values given for clock cycle, conversion times, etc. are for 4.2 MHz operation, with main clock maximum clock speed selected.

*1: The number of ports used for common/segment outputs is selected by mask option for MB89983 and is selected by software for MB89P985 and MB89PV980

1.3 Differences Between Products

This section describes the differences between products in the MB89980 series and lists points to note in product selection.

■ Differences among Products and Points to Note for Product Selection

Table 1.3-1 Package and Corresponding Products

Package	Part number		
	MB89983	MB89P985	MB89PV980
FPT-64P-M03	O	O	X
FPT-64P-M09	O	O	X
MQP-64C-P01	X	X	O

O: Available

X: Not available

❍ Memory size

Before evaluating using the piggyback product, verify its differences from the product that will actually be used. Take particular care on the following points. (See section 3.1 "Memory Space."):

- On the MB89983, each general-purpose register area is limited to 0100_H to $017F_H$ (16 banks).
- The stack area, etc., is set at the upper limit of the RAM.

❍ Current consumption

- In the case of the Piggyback add the current consumed by the EPROM which is connected to the top socket.
- When operated at low speed, the product with a one-time PROM (OTPROM) or an EPROM will consume more current than the product with mask ROM. However the current consumption in sleep/stop modes, is the same.

Reference:

For more information about the package, see Section 1.6 "Package Dimensions."

For more information about the current consumption, see the electrical characteristics in the Data Sheet.

❍ Mask options

Functions that can be selected as options and how to designate these options vary by the product. Before using, check appendix C "Mask Options."

❍ **Pull-up resistor**

- Pull-up resisitor of MB89P985 and MB89PV980 are selected by pull-up control register (Port 0, 1, 5), but there are no pull-up resistor for Port 2, 4 and 6 in MB89P985 and MB89PV980.
- Pull-up resistor of MB89983 are selected by mask option (Port 0, 1, 2, 4, 5, 6)

❍ **Segment/Common port**

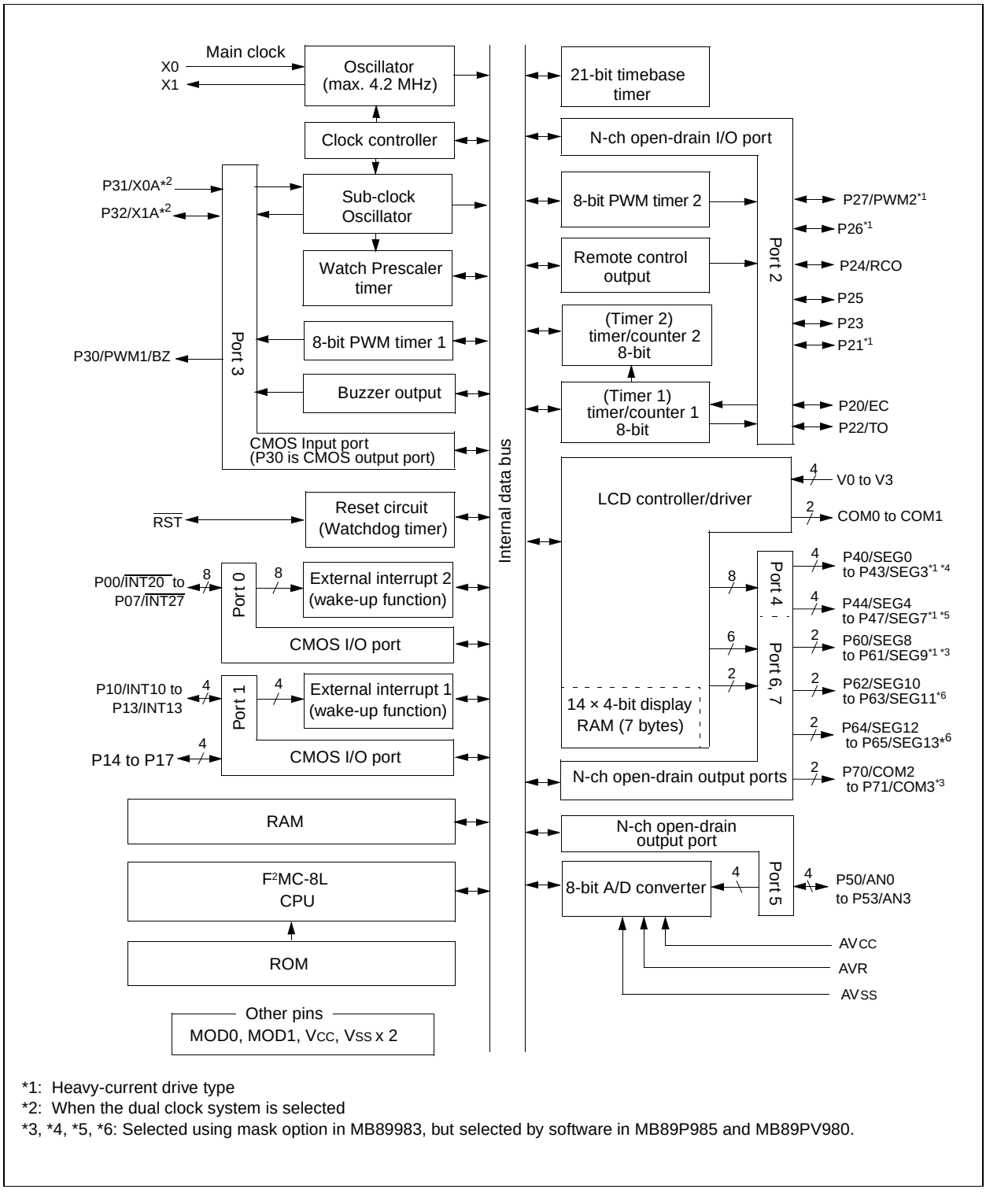
- The Segment/Port, Common/Port output in MB89P985 and MB89PV980 are selected by LCD control register, LCR2.
- The Segment/Port , Common/Port output in MB89983 are selected by mask option.

1.4 Block Diagram of MB89980 Series

Figure 1.4-1 "MB89980 Series Overall Block Diagram" shows the block diagram of the MB89980 series.

■ MB89980 Series Block Diagram

Figure 1.4-1 MB89980 Series Overall Block Diagram



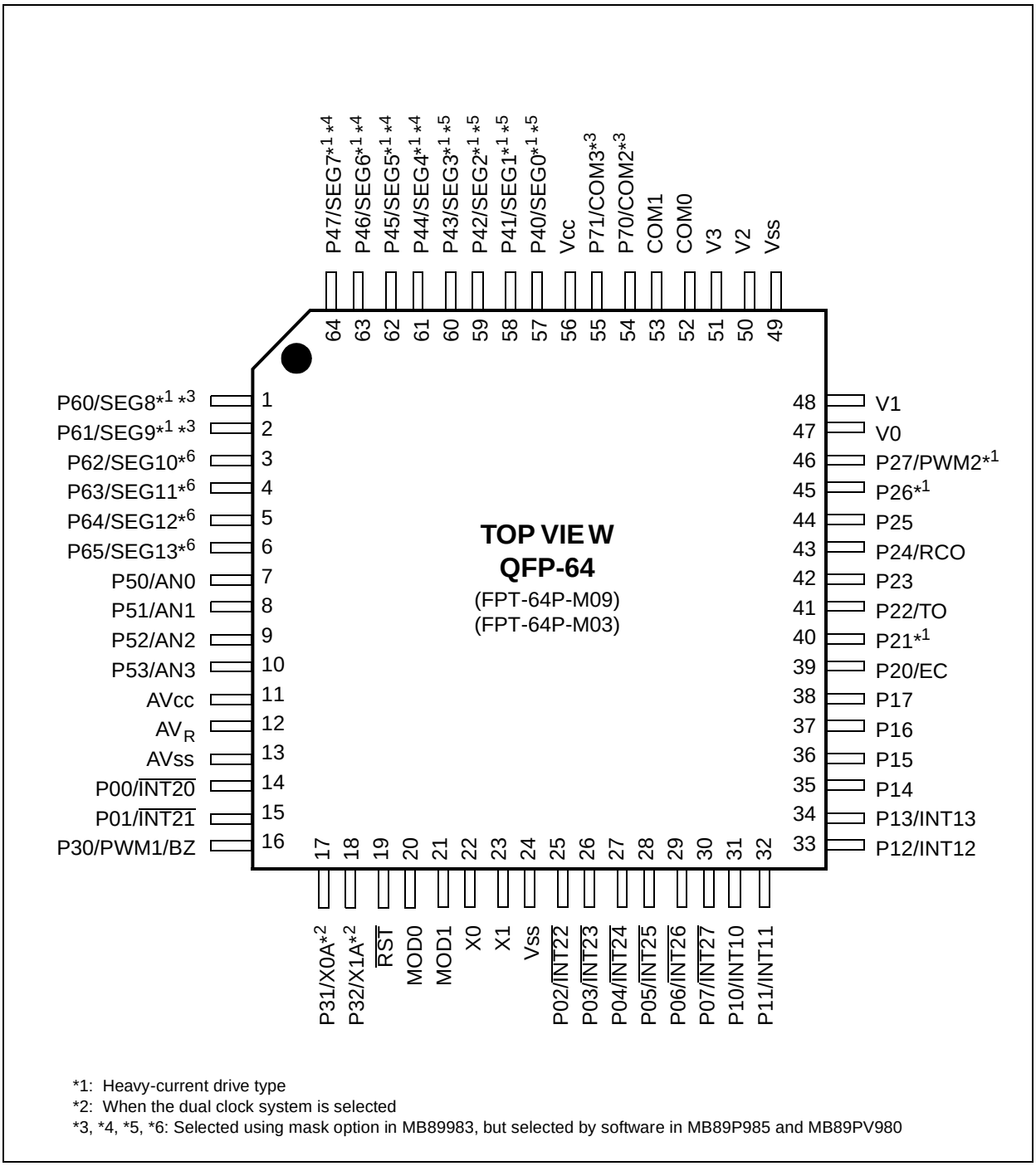
*1: Heavy-current drive type
*2: When the dual clock system is selected
*3, *4, *5, *6: Selected using mask option in MB89983, but selected by software in MB89P985 and MB89PV980.

1.5 Pin Assignment

Figure 1.5-1 "FPT-64P-M03 and FPT-64P-M09 Pin Assignment" and 1.5-2 "MQP-64C-P01 Pin Assignment" show the pin assignment diagrams for the MB89980 series.

■ FPT-64P-M03 and FPT-64P-M09 Pin Assignment

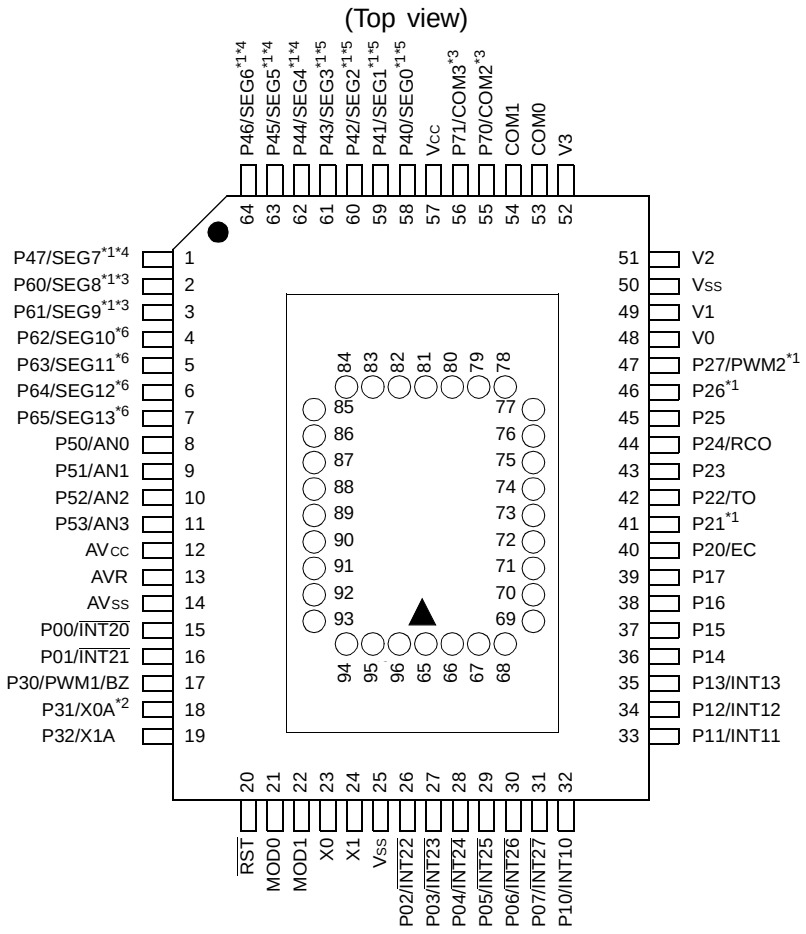
Figure 1.5-1 FPT-64P-M03 and FPT-64P-M09 Pin Assignment



*1: Heavy-current drive type
*2: When the dual clock system is selected
*3, *4, *5, *6: Selected using mask option in MB89983, but selected by software in MB89P985 and MB89PV980

■ MQP-64C-P01 Pin Assignment

Figure 1.5-2 MQP-64C-P01 Pin Assignment



*1: Heavy-current drive type
*2: When the dual clock system is selected
*3, *4,*5, *6: Selected using mask option in MB89983, but selected by software in MB89P985 and MB89PV980.

- Pin assignment on package top (MB89PV980 only)

Pin no.	Pin name	Pin no.	Pin name	Pin no.	Pin name	Pin no.	Pin name
65	N.C.	73	A2	81	N.C.	89	$\overline{OE}$
66	VPP	74	A1	82	O4	90	N.C.
67	A12	75	A0	83	O5	91	A11
68	A7	76	N.C.	84	O6	92	A9
69	A6	77	O1	85	O7	93	A8
70	A5	78	O2	86	O8	94	A13
71	A4	79	O3	87	$\overline{CE}$	95	A14
72	A3	80	VSS	88	A10	96	VCC

N.C.: Internally connected. Do not use.

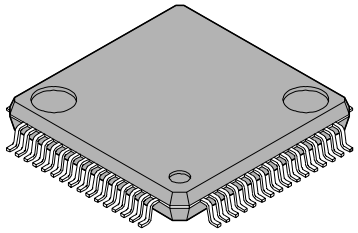
1.6 Package Dimensions

Three types of package are available for MB89980 series.
Figure 1.6-1 "FPT-64P-M03 Package Dimensions" and Figure 1.6-2 "FPT-64P-M09 Package Dimensions" shows the package dimensions.

■ **FPT-64P-M03 Package Dimensions**

Figure 1.6-1 FPT-64P-M03 Package Dimensions

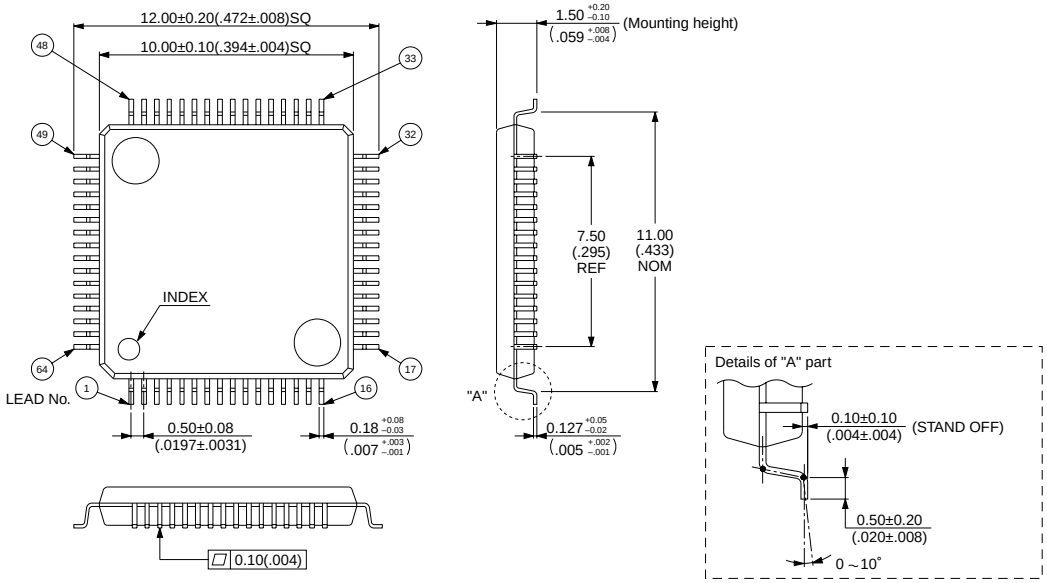
64-pin Plastic



(FPT-64P-M03)

Lead pitch	0.5 mm
Package width × length	10 × 10 mm
Lead shape	Gull-wing
Sealing method	Plastic mold

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

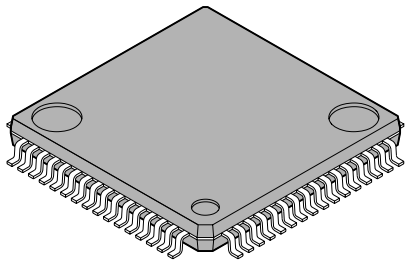


Dimension in mm (inches)

■ FPT-64P-M09 Package Dimensions

Figure 1.6-2 FPT-64P-M09 Package Dimensions

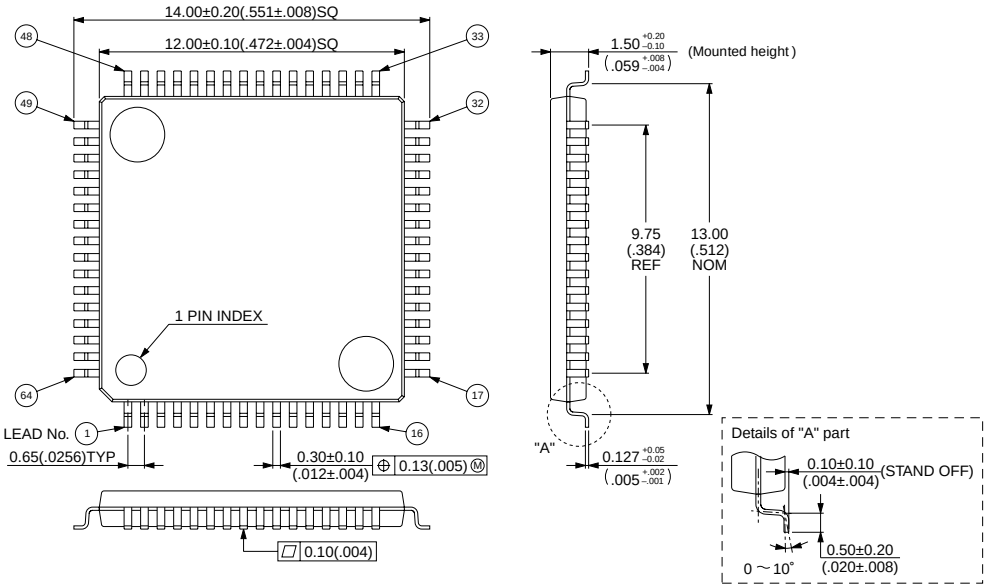
64-pin Plastic QFP



(FPT-64P-M09)

Lead pitch	0.65 mm
Package width × length	12 × 12 mm
Lead shape	Gull-wing
Sealing method	Plastic mold

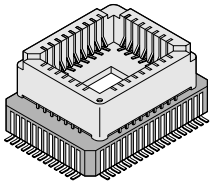
64-pin Plastic QFP
(FPT-64P-M09)

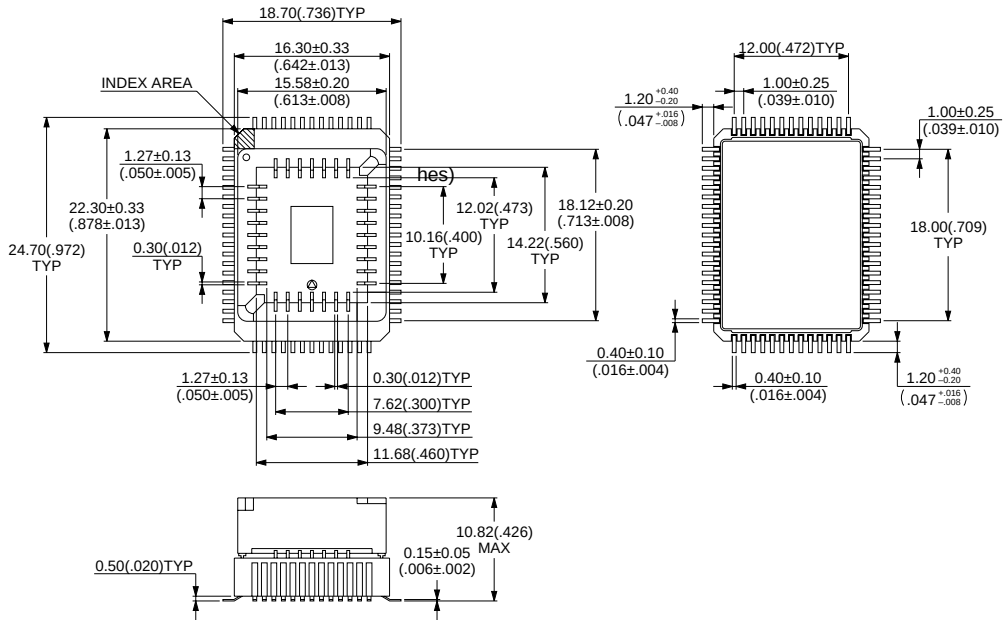


Dimensions in mm (inches)

■ MQP-64C-P01 Package Dimension

Figure 1.6-3 MQP-64C-P01 Package Dimensions

64-pin Ceramic MQFP  (MQP-64C-P01)	Lead pitch	1.00 mm
	Lead shape	Straight
	Motherboard material	Ceramic
	Mounted socket material	Plastic



Dimension in mm (inches)

1.7 I/O Pins and Pin Functions

Table 1.7-1 "Pin Description" lists the MB89980 series I/O pins and their functions.
Table 1.7-2 "I/O Circuit Type" lists the I/O circuit types.
The letter in the "I/O circuit type" column in Table 1.7-1 "Pin Description" refers to the letter in the "Type" column Table 1.7-2 "I/O Circuit Type".

■ I/O Pins and Pin Functions

Table 1.7-1 Pin Description

Pin no.		Pin name	I/O circuit type			Function
LQFP*1 QFP*2	MQFP*3		MB89983	MB89P985	MB89PV980	
22	23	X0	A			Crystal or other resonator connector pins for the main clock. The external clock can be connected to X0. When this is done, be sure to leave X1 open. CR oscillation selectability in model with a mask ROM only.
23	24	X1				
20	21	MOD0	C	R	C	A hysteresis input type Memory access mode setting pins Connect directly to VSS.
21	22	MOD1				
19	20	$\overline{\text{RST}}$	D			Reset I/O pin This pin is an N-ch open-drain output type with a pull-up resistor, and a hysteresis input type. "L" is output from this pin by an internal reset request (optional). The internal circuit is initialized by the input of "L".
14 to 15	15 to 16	P00/$\overline{\text{INT20}}$ to P01/$\overline{\text{INT21}}$	E	F		General-purpose I/O ports Also serve as an external interrupt 2 input (wake-up function). External interrupt 2 input is hysteresis input.
25 to 30	26 to 31	P02/$\overline{\text{INT22}}$ to P07/$\overline{\text{INT27}}$	E	F		General-purpose I/O ports Also serve as an external interrupt 2 input (wake-up function). External interrupt 2 input is hysteresis input.

Table 1.7-1 Pin Description

Pin no.		Pin name	I/O circuit type			Function
LQFP*1 QFP*2	MQFP*3		MB89983	MB89P985	MB89PV980	
31 to 34	32 to 35	P10/ INT10 to P13/ INT13	E	F		General-purpose I/O ports Also serve as input for external interrupt 1 input (wake-up function). External interrupt 1 input is hysteresis input.
35 to 38	36 to 39	P14 to P17	G	H		General-purpose I/O ports
39	40	P20/EC	J	K		N-ch open-drain general-purpose I/O port Also serve as the external clock input for the 8/16-bit timer/ counter. The peripheral is a hysteresis input.
40	41	P21	L	M		N-ch open-drain general-purpose I/O port
41	42	P22/TO	L	M		N-ch open-drain general-purpose I/O port Also serves as an 8/16-bit timer/ counter output.
42	43	P23	L	M		N-ch open-drain general-purpose I/O port
43	44	P24/RCO	L	M		N-ch open-drain general-purpose I/O port Also serves as Remote control output.
44 to 45	45 to 45	P25 to P26	L	M		N-ch open-drain general-purpose I/O port
46	47	P27/ PWM2	L	M		N-ch open-drain general-purpose I/O port Also serves as the square wave or PWM wave output for the 8-bit PWM timer 2.
16	17	P30/ PWM1/ BZ	I			General-purpose CMOS Output port Also serves as the square wave or PWM wave output for the 8-bit PWM timer 1, or buzzer output.

Table 1.7-1 Pin Description

Pin no.		Pin name	I/O circuit type			Function
LQFP*1 QFP*2	MQFP*3		MB89983	MB89P985	MB89PV980	
17	18	P31	R			General-purpose CMOS Input port (Hysteresis input type)
		X0A	B			Crystal or other resonator connector pins for the subclock (Subclock: 32.768 kHz) The external clock can be connected to X0A. When this is done, Be sure to leave X1A open.
18	19	P32	R			General-purpose CMOS Input port (Hysteresis input type)
		X1A	B			Crystal or other resonator connector pins for the subclock (Subclock: 32.768 kHz) The external clock can be connected to X0A. When this is done, Be sure to leave X1A open.
7 to 10	8 to 11	P50/AN0 to P53/ AN3	P	Q		N-ch open-drain general-purpose output ports Also serve as the analog input for the A/D converter.
57 to 64	58 to 64 and 1	P40/ SEG0 to P47/ SEG7	N/O	S		N-ch open-drain general-purpose output ports (High current type) Also serve as an LCD controller/ driver segment output. Switching between port and segment output is done by the mask option for MB89983 and by register for MB89P985/PV980.
1 to 2	2 to 3	P60/ SEG8 to P61/ SEG9	N/O	S		N-ch open-drain general-purpose output ports (High-current type) Also serve as an LCD controller/ driver segment output. Switching between port and segment output is done by the mask option for MB89983 and by register for MB89P9 85/PV980.

Table 1.7-1 Pin Description

Pin no.		Pin name	I/O circuit type			Function
LQFP*1 QFP*2	MQFP*3		MB89983	MB89P985	MB89PV980	
3 to 6	4 to 7	P62/ SEG10 to P65/ SEG13	N/O	S		N-ch open-drain general-purpose output ports Also serve as an LCD controller/ driver segment output. Switching between port and segment output is done by the mask option for MB89983 and by register for MB89P985/PV980.
54, 55	55, 56	P70/ COM2, P71/ COM3	T/O	S		N-ch open-drain general-purpose output ports Also serve as an LCD controller/ driver common output. Switching between port and common output is done by the mask option for MB89983 and by register for MB89P985/PV980.
52, 53	53, 54	COM0, COM1	O			LCD controller/driver common output
47, 48, 50, 51	48, 49, 51, 52	V0 to V3	-	-		LCD driving power supply pins.
56	57	Vcc	-	-		Power supply pin
24, 29	25, 50	Vss	-	-		Power supply (GND) pin
11	12	AVcc	-	-		A/D converter power supply pin
12	13	AVR	-	-		A/D converter reference voltage input pin
13	14	AVss	-	-		A/D converter power supply pin Use this pin at the same voltage as VSS.

*1 : FPT-64P-M03

*2 : FPT-64P-M09

*3 : MQP-64C-P01

Table 1.7-2 I/O Circuit Type

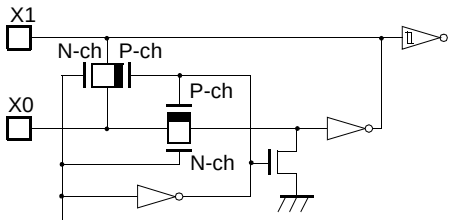
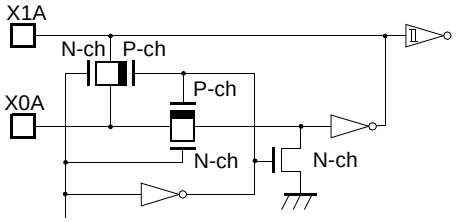
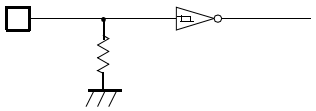
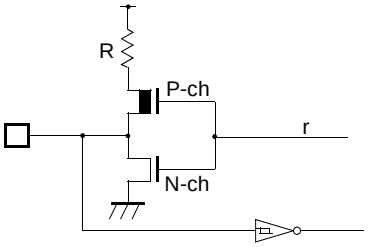
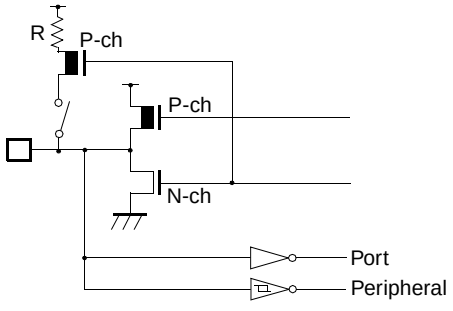
Type	Circuit	Remarks
A		Main clock (main clock crystal oscillator) • At an oscillation feedback resistor of approximately 1 MΩ/5.0 V • CR oscillation is selectable for MB89983 only
B		Subclock (subclock crystal oscillator) • At an oscillation feedback resistor of approximately 4.5 MΩ/5.0 V
C		• Hysteresis input • At a pull-down resistor (P-ch) of approximately 50 kΩ/5.0 V
D		• At an output pull-up resistor (P-ch) of approximately 50 kΩ/5.0 V • Hysteresis input
E		• CMOS output • CMOS input • The peripheral is a hysteresis input type. • Pull-up resistor is approximately 50 kΩ/ 5.0 V • Pull-up resistor is selected by mask option.

Table 1.7-2 I/O Circuit Type

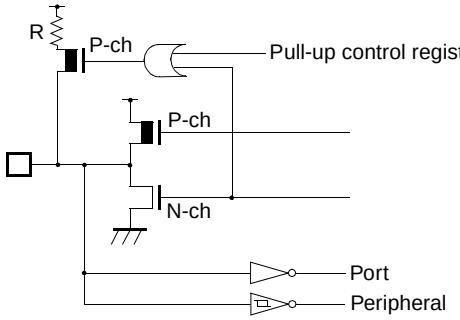
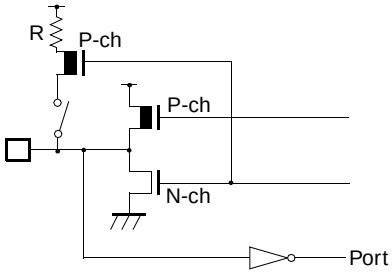
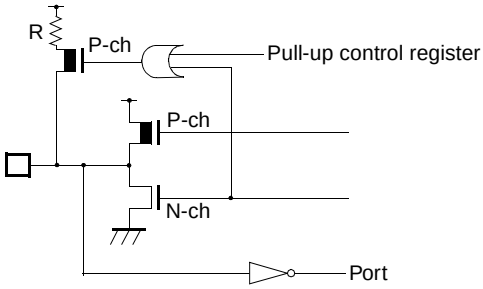
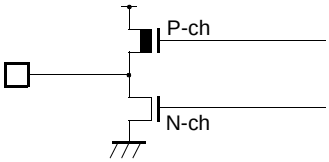
Type	Circuit	Remarks
F		• CMOS output • CMOS input • The peripheral is a hysteresis input type. • Pull-up resistor is approximately 50 kΩ/ 5.0 V • Pull-up resisitor is selected by pull-up control register
G		• CMOS output • CMOS input • Pull-up resistor is approximately 50 kΩ/ 5.0 V • Pull-up resistor is selected by mask option.
H		• CMOS output • CMOS input • Pull-up resistor is approximately 50 kΩ/ 5.0 V • Pull-up resisitor is selected by pull-up control register
I		• CMOS output

Table 1.7-2 I/O Circuit Type

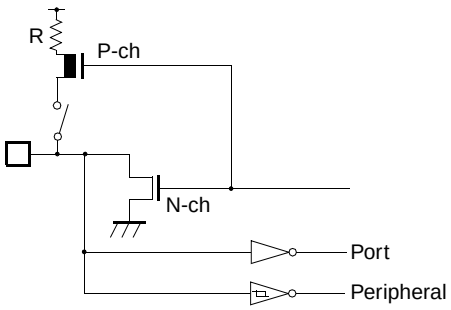
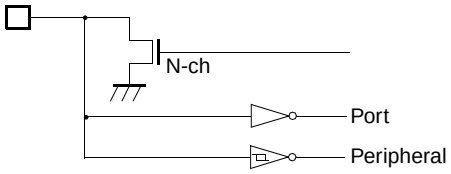
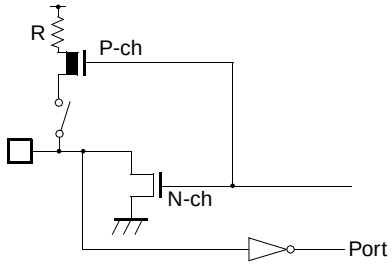
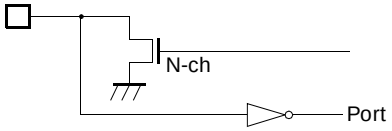
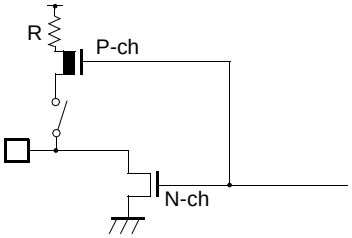
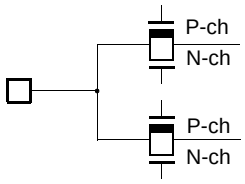
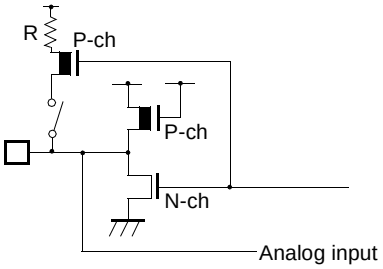
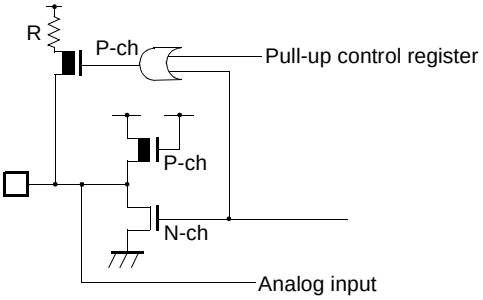
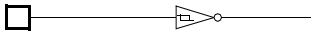
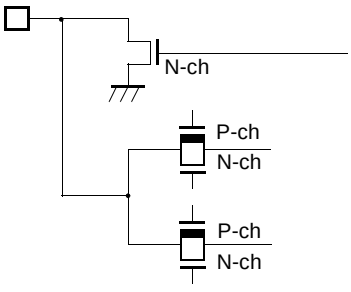
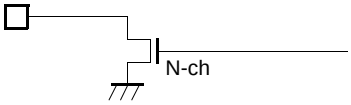
Type	Circuit	Remarks
J		• N-ch open-drain output • CMOS input • The peripheral is a hysteresis input type. • Pull-up resistor is approximately 50 kΩ/ 5.0 V • Pull-up resistor is selected by mask option.
K		• N-ch open-drain output • CMOS input • The peripheral is a hysteresis input type.
L		• N-ch open-drain output • CMOS input • P21, P26, and P27 are a heavy-current drive type. • Pull-up resistor is approximately 50 kΩ/ 5.0 V • Pull-up resistor is selected by mask option.
M		• N-ch open-drain output • CMOS input • P21, P26, and P27 are a heavy-current drive type.
N		• N-ch open-drain output • Pull-up resistor is approximately 50 kΩ/ 5.0 V • Pull-up resistor is selected by mask option.

Table 1.7-2 I/O Circuit Type

Type	Circuit	Remarks
O		• LCD controller/driver common/segment output
P		• N-ch open-drain output • Analog input (A/D converter) • Pull-up resistor is approximately 50 kΩ/ 5.0 V • Pull-up resistor is selected by mask option.
Q		• N-ch open-drain output • Analog input (A/D converter) • Pull-up resistor is approximately 50 kΩ/ 5.0 V • Pull-up resisitor is selected by pull-up control register
R		• Hysteresis input
S		• N-ch open-drain output • LCD controller/driver segment output
T		• N-ch open-drain output

CHAPTER 2 HANDLING DEVICES

This chapter describes points to note when using the general-purpose single-chip microcontroller.

2.1 Notes on Handling Devices

This section lists points to note regarding the power supply voltage, pins, and other device handling aspects.

■ Notes on Handling Devices

❍ Take great care not to exceed the maximum rated voltage (prevent latchup).

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

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

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

❍ Stabilizing supply voltage is important.

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

❍ Treatment of unused input pins

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

❍ Treatment of N.C. pins

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

❍ Treatment of power supply pins on microcontroller with A/D or D/A converters

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

❍ Precautions when using an external clock

Even when an external clock is used, oscillation stabilization delay time is required for power-on reset (optional) and wake-up from stop mode.

❍ Treatment of two Vss pins

Two Vss pins should be connected together externally.

❍ **Treatment of input port pins in standby mode**

To avoid current leakage, it is recommended to remain a known logic level of input port pins during the standby mode.

CHAPTER 3 CPU

This chapter describes the functions and operation of the CPU.

3.1 Memory Space

The microcontrollers of the MB89980 series offer a memory space of 64 Kbytes. The memory space contains the I/O area, RAM area, ROM area, and external area. The memory space contains areas used for special purposes such as the general-purpose registers and vector table.

■ Memory Space Structure

❍ I/O area (addresses: 0000_H to $007F_H$)

- Control registers and data registers for the internal peripheral functions are located in this area.
- As the I/O area is allocated within the memory space, I/O can be accessed in the same way as memory. High-speed access using direct addressing is available.

❍ RAM area

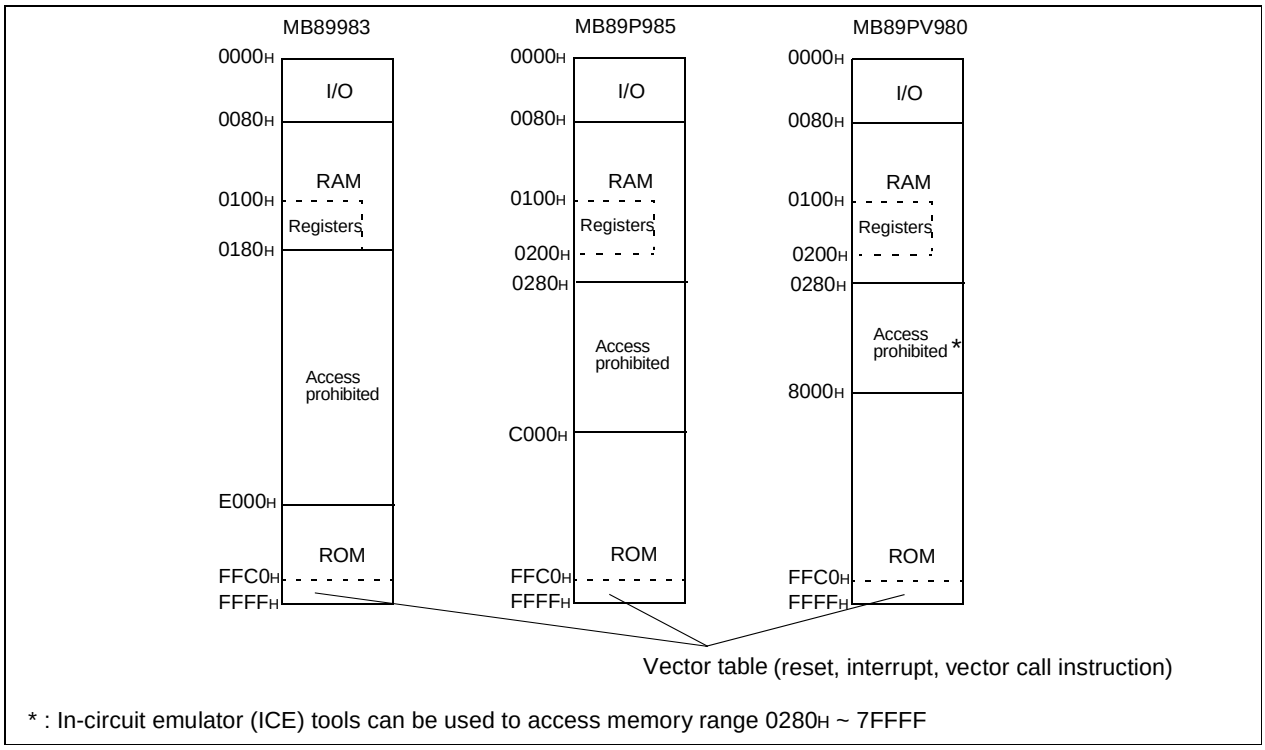
- Internal static RAM is provided as an internal data area.
- The internal RAM size differs between products.
- Addresses between 80_H and FF_H support high-speed access using direct addressing.
- Addresses between 100_H and $1FF_H$ can be used as the general-purpose register area (restrictions apply for some products).
- The contents of RAM is indeterminate after a reset.

❍ ROM area

- Internal ROM is provided as an internal program area.
- The internal ROM size differs between products. Setting the memory access mode to external ROM mode enables internal ROM to be disconnected and set as an external area.
- Addresses between $FFC0_H$ and $FFFF_H$ are used for the vector table, etc.

■ Memory Map

Figure 3.1-1 Memory Map



* : In-circuit emulator (ICE) tools can be used to access memory range 0280H ~ 7FFFF

3.1.1 Special Areas

In addition to the I/O area, the special purpose areas in the memory space include the general-purpose register area and the vector table area.

■ General-purpose Register Areas (Addresses: 0100_H to $01FF_H$)

- Provides auxiliary registers for 8-bit arithmetic operation and transfer instructions.
- Allocated to a region of the RAM area. Can also be used as normal RAM.
- Using the area as general-purpose registers enables high-speed access by general-purpose register addressing using short instructions.

Table 3.1-1 "General-purpose Register Areas" lists the areas in each device that can be used for general-purpose registers.

Table 3.1-1 General-purpose Register Areas

CPU	MB89983	MB89P985	MB89PV980
Number of Banks	16	32	32
Address Range	0100_H to $017F_H$	0100_H to $01FF_H$	0100_H to $01FF_H$

Reference:

See section 3.2.2 "Register Bank Pointer (RP)" and section 3.3 "General-purpose Registers" for details.

■ Vector Table Area (Addresses: $FFC0_H$ to $FFFF_H$)

- Used as the vector table for the vector call instruction, interrupts, and resets.
- The vector table is allocated at the top of the ROM area. The start address of the corresponding processing routine is set as data at each vector table address.

Table 3.1-2 "Vector Table" lists the vector table addresses referenced by the vector call instruction, interrupts, and resets.

Reference:

See Section 3.4 "Interrupts" Section 3.5 "Resets" and "(6) CALLV #vct" in Appendix B.3 "Special Instructions" for details.

Table 3.1-2 Vector Table

Vector call instruction	Vector table address	
	Upper	Lower
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	$FFCE_H$	$FFCF_H$

Interrupts	Vector table address	
	Upper	Lower
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	$FFFA_H$	$FFFB_H$
More data	-*	$FFFD_H$
Reset vector	$FFFE_H$	$FFFF_H$

$FFFC_H$ is not available. (Set FF_H.)

3.1.2 Storing 16-bit Data in Memory

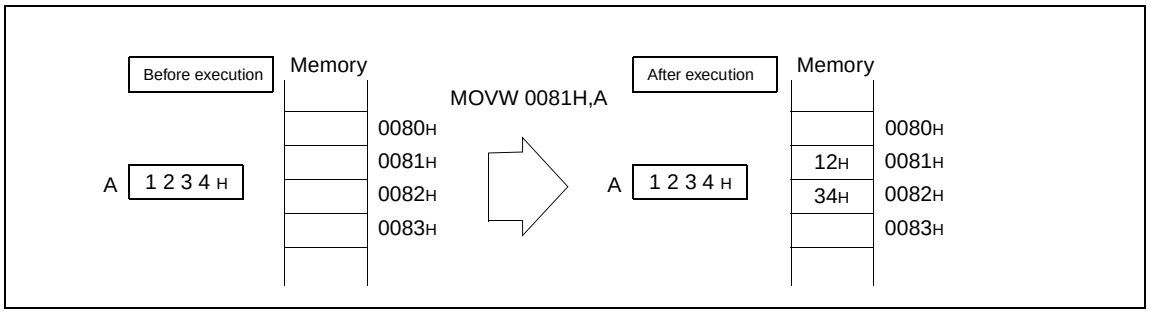
For 16-bit data and the stack, store the upper data in the lower memory address value.

■ Storing 16-bit Data in RAM

When writing 16-bit data to memory, store the upper byte at the lower address and the lower byte at the next address. Handle reading of 16-bit data in the same way.

Figure 3.1-2 "Storing 16-bit Data in Memory" shows how 16-bit data is stored in memory.

Figure 3.1-2 Storing 16-bit Data in Memory



■ Storing 16-bit Operands

The byte order applies when specifying a 16-bit operand in an instruction. Store the upper byte at the address following the operation code (instruction) and the lower byte at the next address.

The byte ordering applies to both 16-bit immediate data and operands that specify a memory address.

Figure 3.1-3 "Byte Order of 16-bit Data in an Instruction" shows how 16-bit data is stored in an instruction.

Figure 3.1-3 Byte Order of 16-bit Data in an Instruction

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

                  After assembly
                  .
                  .
                  .
            X X X 0 H   XX XX
            X X X 2 H   60 56 78   ; Extended address
            X X X 5 H   E4 12 34   ; 16-bit immediate data
            X X X 8 H   XX
                  .
                  .
                  .
```

■ Storing 16-bit Data on Stack

The same byte order applies when saving 16-bit register data on the stack during an interrupt or similar. The upper byte is stored in the lower address.

3.2 Dedicated Registers

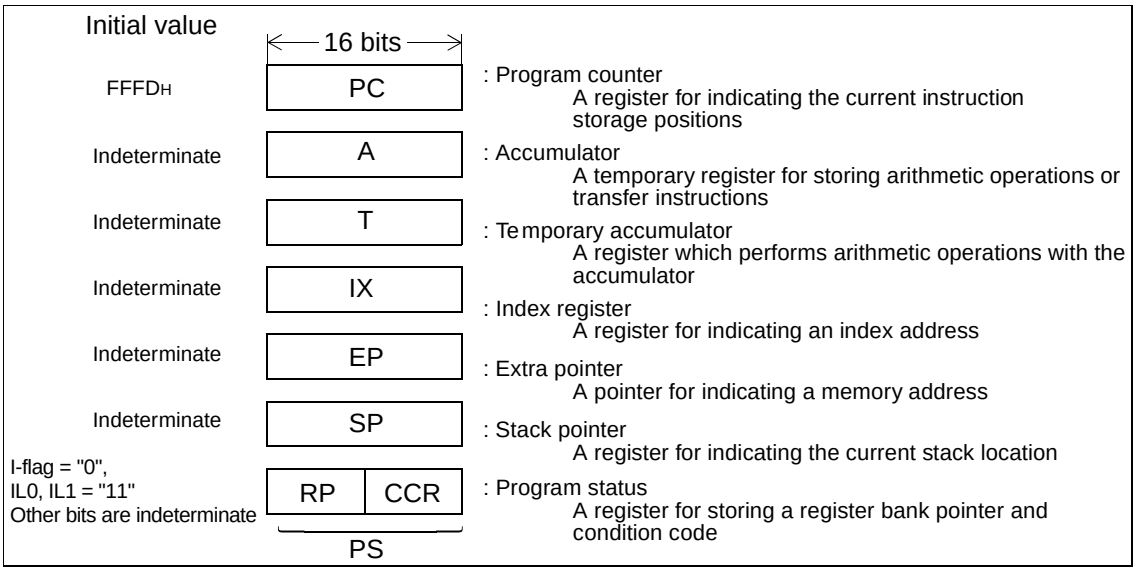
The dedicated registers in the CPU consist of the program counter (PC), two arithmetic operation registers (A and T), three address pointers (IX, EP, and SP), and the program status (PS). All registers are 16 bits.

■ Dedicated Register Configuration

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

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

Figure 3.2-1 Dedicated Register Configuration



■ Dedicated Register Functions

❍ Program counter (PC)

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

❍ Accumulator (A)

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

❍ **Temporary accumulator (T)**

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

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

❍ **Index register (IX)**

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

❍ **Extra pointer (EP)**

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

❍ **Stack pointer (SP)**

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

❍ **Program status (PS)**

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

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

Reference:

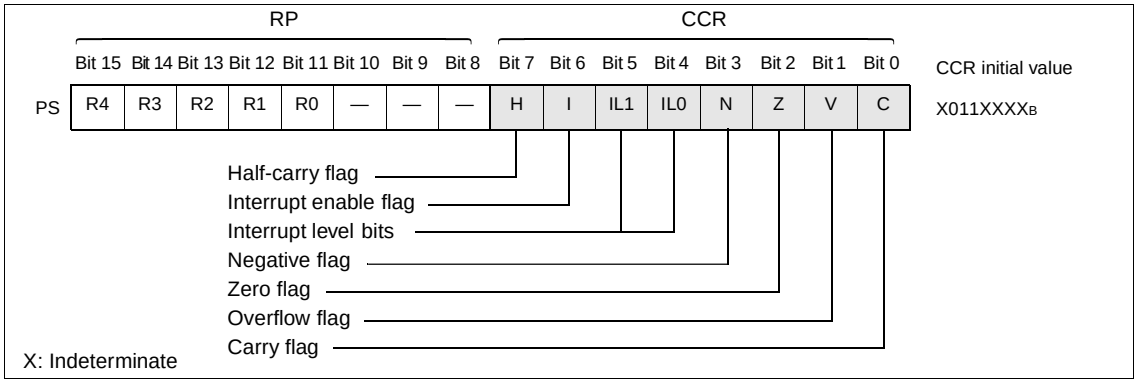
Refer to the F2MC-8L MB89600 series Programming Manual for details on using the dedicated registers.

3.2.1 Condition Code Register (CCR)

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

■ Structure of Condition Code Register (CCR)

Figure 3.2-2 Structure of Condition Code Register



■ Arithmetic Operation Result Bits

❍ **Half-carry flag (H)**

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

❍ **Negative flag (N)**

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

❍ **Zero flag (Z)**

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

❍ **Overflow flag (V)**

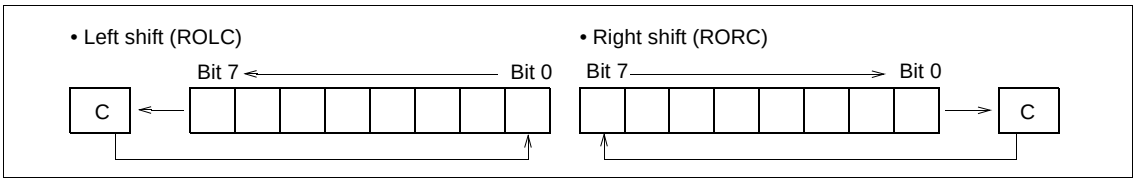
Set if the complement on 2 overflows as a result of an arithmetic operation. Reset if the overflow does not occur.

❍ **Carry flag (C)**

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

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

Figure 3.2-3 Change of Carry Flag by Shift Instruction



Check:

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

Note:

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

■ Interrupt Acceptance Control Bit

❍ **Interrupt enable flag (I)**

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

The initial value after a reset is "0".

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

❍ **Interrupt level bits (IL1, IL0)**

These bits indicate the level of the interrupt currently being accepted by the CPU. The value is compared with the interrupt level setting registers (ILR1 to ILR3) which have a setting for each peripheral function interrupt request (IRQ0 to IRQB).

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

Table 3.2-1 Interrupt Level

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

Note:

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

Reference:

See Section 3.4 "Interrupts" for details on interrupts.

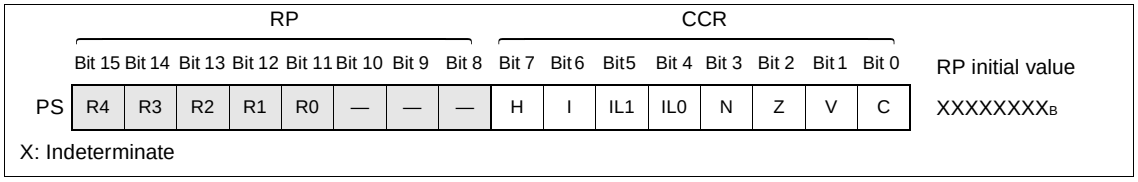
3.2.2 Register Bank Pointer (RP)

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

■ Structure of Register Bank Pointer (RP)

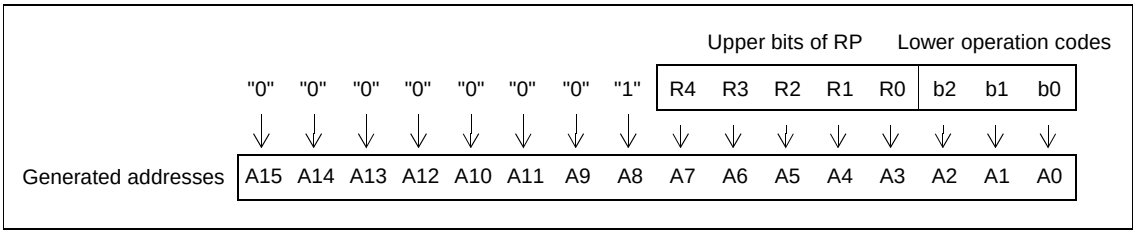
Figure 3.2-4 "Structure of Register Bank Pointer" shows the structure of the register bank pointer.

Figure 3.2-4 Structure of Register Bank Pointer



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

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



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

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

Check:

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

3.3 General-purpose Registers

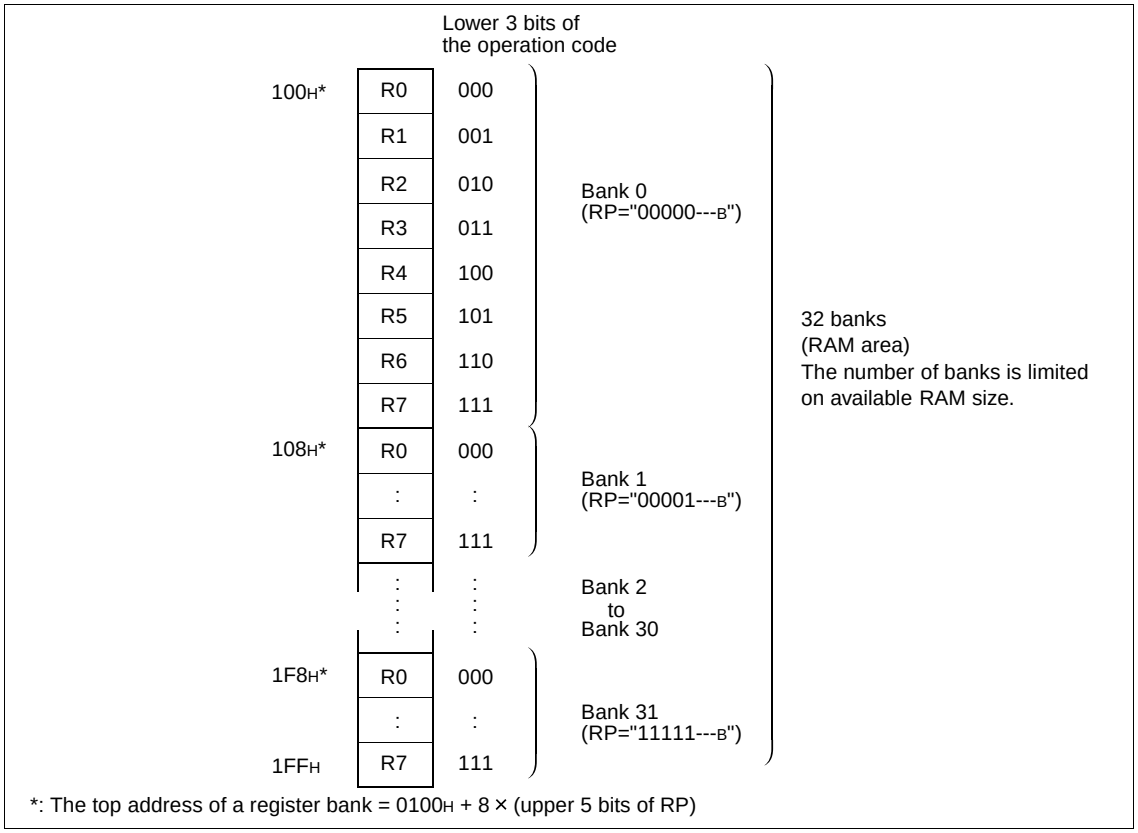
The general-purpose registers are a memory block made up of banks, with 8 × 8-bit registers per bank.
The register bank pointer (RP) is used to specify the register bank.
The function permits the use of up to 32 banks, but the number of banks that can actually be used depends on how much RAM the device has.
Register banks are valid for interrupt processing, vector call processing, and subroutine calls.

■ Structure of General-purpose Registers

- The general-purpose registers are 8 bits and located in the register banks of the general-purpose register area (in RAM).
- One bank contains eight registers (R0 to R7) and up to a total of 32 banks. However, the number of banks available for general-purpose registers is limited on some products if internal RAM only is used.
- The register bank currently in use is specified by the register bank pointer (RP). The lower three bits of the operation code specify general-purpose register 0 (R0) to general-purpose register 7 (R7).

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

Figure 3.3-1 Register Bank Structure



Reference:

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

■ Features of General-purpose Registers

General-purpose registers have the following features:

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

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

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

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

Check:

If an interrupt processing routine changes the register bank pointer (RP), ensure that the program does not also change the interrupt level bits in the condition code register (CCR: IL1, IL0) when specifying the register bank.

3.4 Interrupts

The MB89980 series has 12 interrupt request input corresponding to peripheral functions. An interrupt level can be set independently.
If an interrupt request output is enabled in the peripheral function, an interrupt request from a peripheral function is compared with the interrupt level in the interrupt controller. The CPU performs interrupt operation according to how the interrupt is accepted. The CPU wakes up from standby modes, and returns to the interrupt or normal operation.

■ Interrupt Requests from Peripheral Functions

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

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

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

Table 3.4-1 Interrupt Request and Interrupt Vector

Interrupt request	Vector table address		Bit names of the interrupt level setting register	Simultaneously-generated same-level IRQ priority
	Upper	Lower		
IRQ0 (External interrupt 1-1)	$FFFA_H$	$FFFB_H$	L01, L00	High
IRQ1 (External interrupt 1-2)	$FFF8_H$	$FFF9_H$	L11, L10	
IRQ2 (External interrupt 1-3)	$FFF6_H$	$FFF7_H$	L21, L20	
IRQ3 (External interrupt 1-4)	$FFF4_H$	$FFF5_H$	L31, L30	
IRQ4 (External interrupt 2)	$FFF2_H$	$FFF3_H$	L41, L40	
IRQ5 (8/16-bit timer/counter)	$FFF0_H$	$FFF1_H$	L51, L50	
IRQ6 (Vacancy)	$FFEE_H$	$FFEF_H$	L61, L60	
IRQ7 (Timebase timer)	$FFEC_H$	$FFED_H$	L71, L70	
IRQ8 (Watch prescaler)	$FFEA_H$	$FFEB_H$	L81, L80	
IRQ9 (PWM timer 1)	$FFE8_H$	$FFE9_H$	L91, L90	
IRQA (PWM timer 2)	$FFE6_H$	$FFE7_H$	LA1,LA0	
IRQB (A/D converter)	$FFE4_H$	$FFE5_H$	LB1, LB0	Low

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

The interrupt level setting registers (ILR1, ILR2, ILR3) together contain 12 blocks of 2-bit data, with each data corresponding to an interrupt request from a peripheral function. The interrupt level for each interrupt is set in that interrupt's corresponding 2-bit data (interrupt level setting bits).

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

Figure 3.4-1 Structure of Interrupt Level Setting Registers

Register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
ILR1	007CH	L31	L30	L21	L20	L11	L10	L01	L00	11111111B
		W	W	W	W	W	W	W	W	
ILR2	007DH	L71	L70	L61	L60	L51	L50	L41	L40	11111111B
		W	W	W	W	W	W	W	W	
ILR3	007EH	LB1	LB0	LA1	LA0	L91	L90	L81	L80	11111111B
		W	W	W	W	W	W	W	W	

W: Write-only

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

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

The CPU does not accept interrupt requests set to interrupt level 3.

Table 3.4-2 "Interrupt Level Setting Bit and Interrupt Level" shows the relationship between the interrupt level setting bits and the interrupt levels.

Table 3.4-2 Interrupt Level Setting Bit and Interrupt Level

L01 to LB1	L00 to LB0	Request interrupt level	High-low
0	0	1	High
0	1		
1	0	2	↕
1	1	3	Low (no interrupt)

Note:

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

Check:

As the IRL1, ILR2, and ILR3 registers are write-only, the bit manipulation instructions cannot be used.

3.4.2 Interrupt Processing

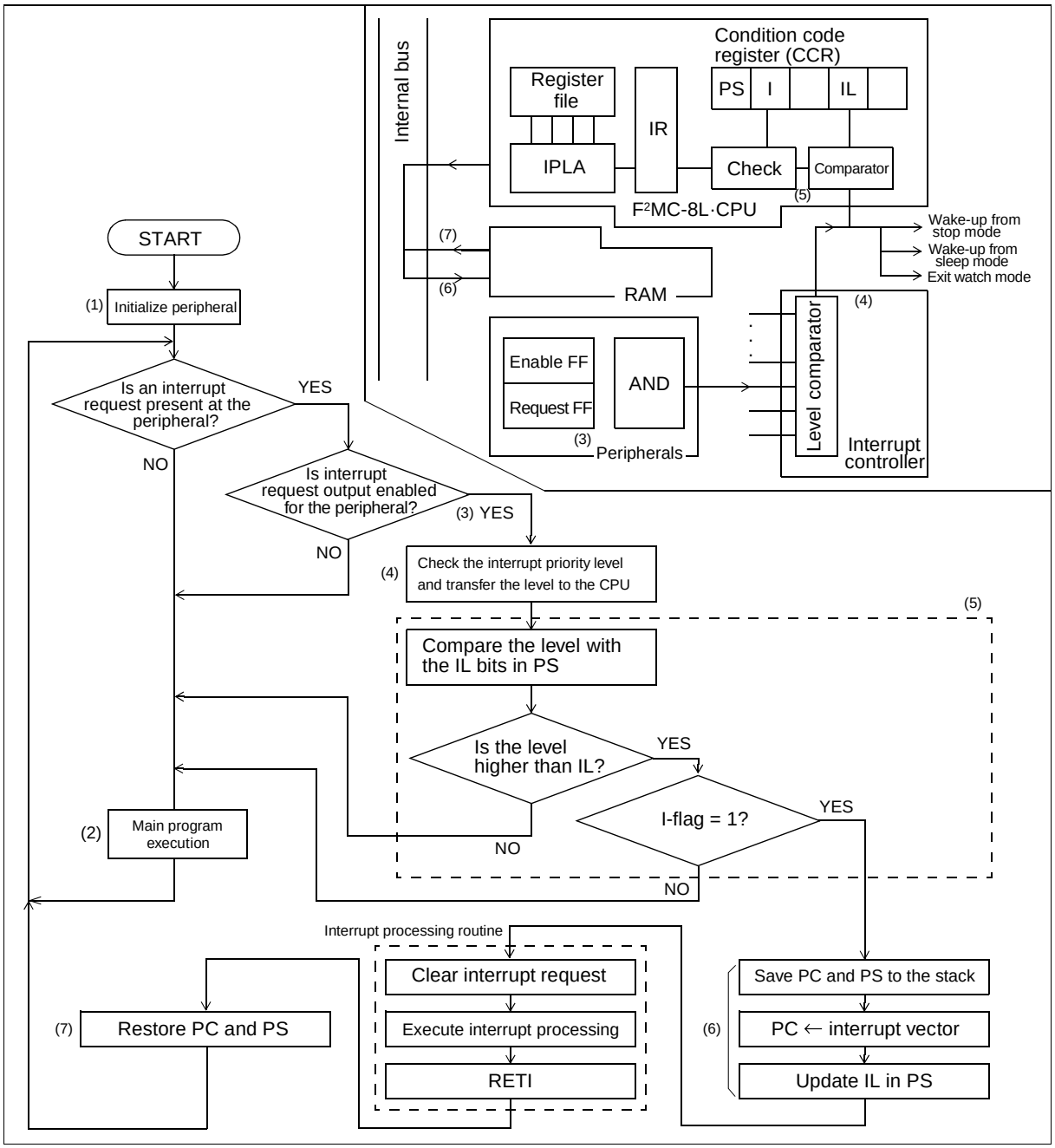
The interrupt controller transmits the interrupt level to the CPU when an interrupt request is generated by a peripheral function. If the CPU is able to receive the interrupt, the CPU temporarily halts the currently executing program and executes the interrupt processing routine.

■ Interrupt Processing

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

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

Figure 3.4-2 Interrupt Processing



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

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

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

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

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

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

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

Check:

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

Reference:

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

Note:

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

3.4.3 Multiple Interrupts

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

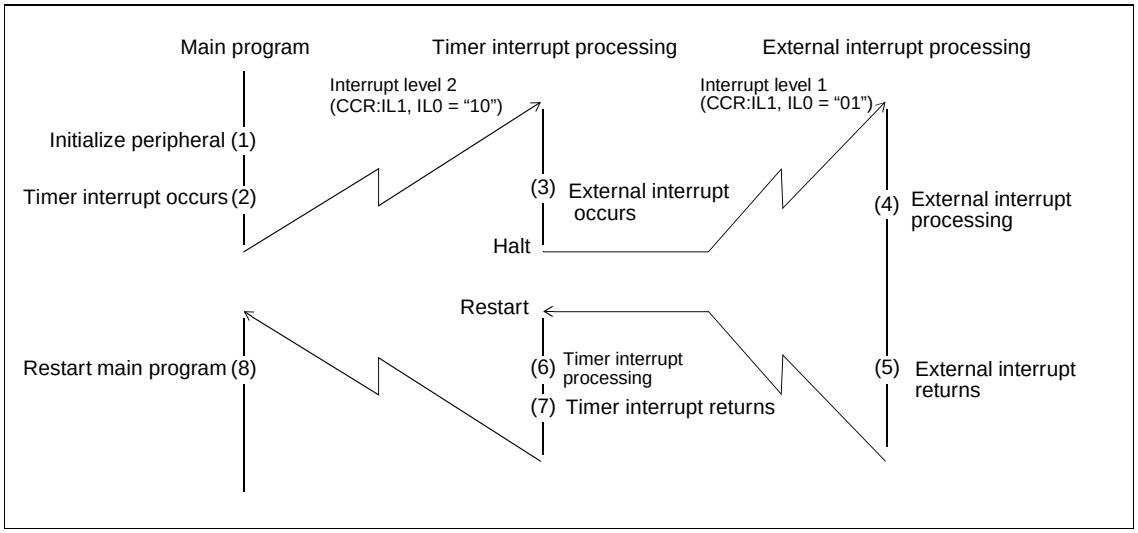
■ Multiple Interrupts

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

❍ Example of multiple interrupts

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

Figure 3.4-3 Example of Multiple Interrupts



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

3.4.4 Interrupt Processing Time

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

■ Interrupt Processing Time

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

❍ Interrupt request sampling time

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

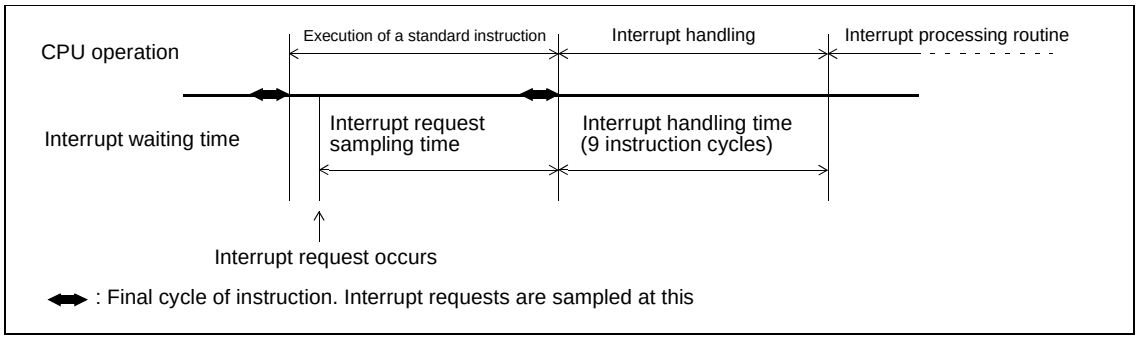
❍ Interrupt handling time

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

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

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

Figure 3.4-4 Interrupt Processing Time



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

Reference:

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

3.4.5 Stack Operation during Interrupt Processing

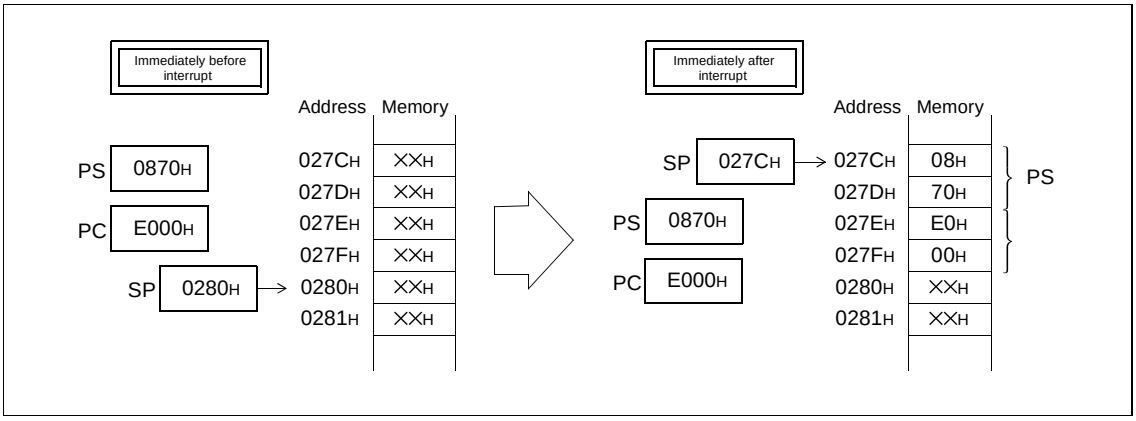
This section describes the saving of the register contents to the stack and restore operation during interrupt processing.

■ Stack Operation at Start of Interrupt Processing

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

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

Figure 3.4-5 Stack Operation at Start of Interrupt Processing



■ Stack Operation at Interrupt Return

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

Check:

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

3.4.6 Stack Area for Interrupt Processing

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

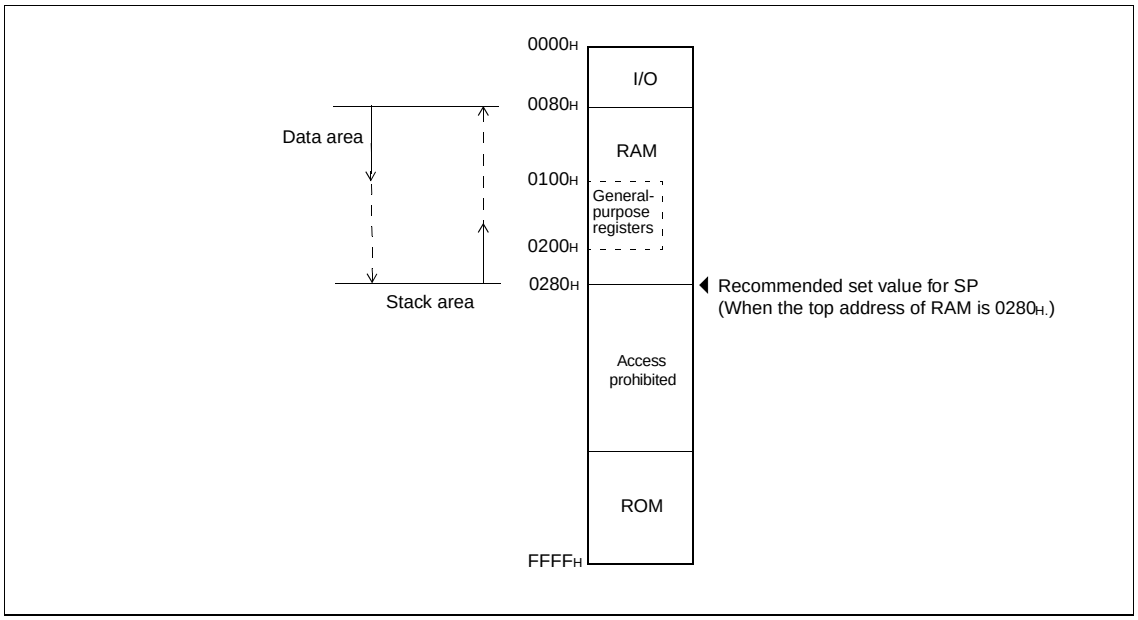
■ Stack Area for Interrupt Processing

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

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

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

Figure 3.4-6 Stack Area for Interrupt Processing



Note:

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

3.5 Resets

The MB89980 series supports the following four types of reset source:

- **External reset**
- **Software reset**
- **Watchdog reset**
- **Power-on reset (optional)**

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

■ Reset Source

Table 3.5-1 Reset Source

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

❍ External reset

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

When power is turned on to products with power-on reset or for external resets in stop mode, the reset operation is performed after the oscillation stabilization delay time has passed and the CPU wakes up from the external reset. External resets on products without power-on reset do not wait for the oscillation stabilization delay time.

The external reset pin can also function as a reset output pin (optional).

❍ Software reset

Writing "0" to the software reset bit in the standby control register (STBC: RST) generates a four-instruction cycle reset. The software reset does not wait for the oscillation stabilization delay time.

❍ Watchdog reset

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

❍ **Power-on reset**

Products can be set to with or without power-on reset (optional). On products with power-on reset, turning on the power generates a reset. The reset operation is performed after the oscillation stabilization delay time has passed.

On products without power-on reset, an external reset circuit is required to generate a reset when the power is turned on.

■ Main Clock Oscillation Stabilization Delay Time and the Reset Source

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

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

In software or watchdog reset, if the reset occurs while the device is in main clock mode, no stabilization time is provided. If it occurs in the subclock mode, however, a stabilization time is provided since the main clock oscillation is stopped.

Table 3.5-2 "Reset Source and Oscillation Stabilization Delay Time" shows the relationships between the reset sources and the main clock oscillation stabilization delay time, and reset mode (mode fetch) operations.

Table 3.5-2 Reset Source and Oscillation Stabilization Delay Time

Reset source	Operating state	Reset operation and main clock oscillation stabilization delay time	
		With power-on reset	Without power-on reset
External reset*1	At power on, during stop mode, or subclock mode	After the main clock oscillation stabilization delay time, if the external reset is waked up, reset is operated.*2	Reset state is held until external reset is waked up; then the reset is operated.
Software and watchdog reset	Main clock mode	After 4-instruction-cycle reset occurs, reset is operated.*3	
	Subclock mode	Reset is operated after the main clock oscillation stabilization delay time.*2	
Power-on reset		Device enters main clock oscillation stabilization delay time at power on. Reset is operated after delay time ends.*2	An external circuit must be provided to hold external reset asserted at power on until main clock has had time to stabilize.

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

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

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

3.5.1 External Reset Pin

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

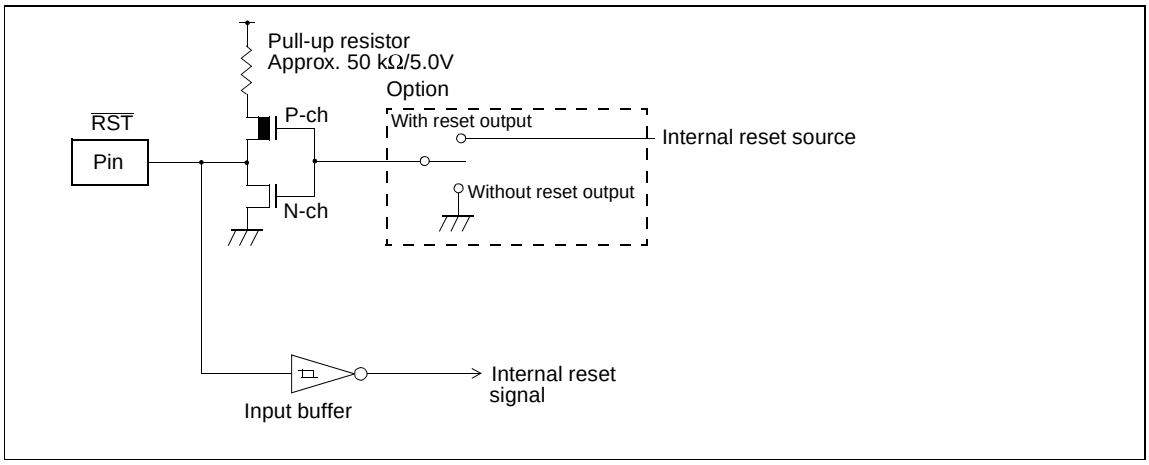
■ Block Diagram of External Reset Pin

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

The external reset pin on products without a reset output option is only for the reset input.

Figure 3.5-1 "Block Diagram of External Reset Pin" shows the block diagram of the external reset pin.

Figure 3.5-1 Block Diagram of External Reset Pin



■ External Reset Pin Functions

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

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

Check:

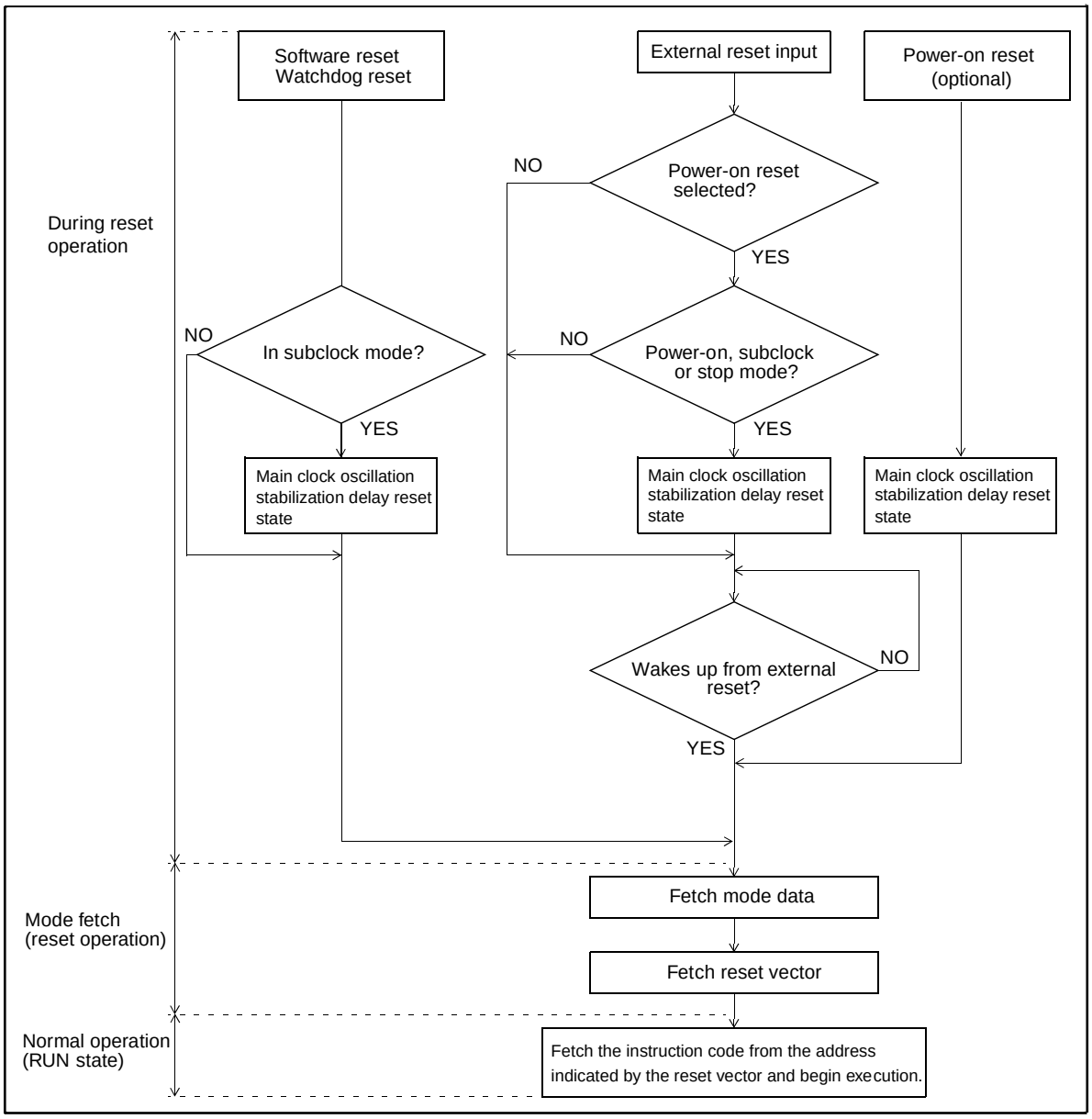
The external reset input accepts asynchronous with the internal clock. Therefore, initialization of the internal circuit requires a clock. Especially when an external clock is used, a clock is needed to be input at the reset.

3.5.2 Reset Operation

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

■ Overview of Reset Operation

Figure 3.5-2 Reset Operation Flow Diagram



■ Mode Pins

The MB89980 series devices are single-chip mode devices. The mode pins (MOD1 and MOD0) must be tied to V_{SS}. The mode pin settings determine whether the mode data and reset vector are read from internal ROM.

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

■ Mode Fetch

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

❍ Mode data (address: $FFFD_H$)

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

❍ Reset vector (address: $FFFE_H$ (upper), $FFFF_H$ (lower))

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

■ Oscillation Stabilization Delay Reset State

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

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

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

On products without power-on reset, the oscillation stabilization delay reset state is not used. Therefore, for such products, hold the external reset pin ($\overline{RST}$) at the "L" level to disable the CPU operation until the source oscillation stabilizes.

■ Effect of Reset on RAM Contents

The contents of RAM are unchanged before and after a reset other than power-on reset. If an external reset is input close to a write timing, however, the contents of the write address cannot be assured. For this reason, all RAM locations being used should be initialized following reset.

3.5.3 Pin States during Reset

Reset initialized the pin states.

■ Pin States during Reset

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

■ Pin States after Reading Mode Data

With a few exceptions, the I/O pins remain in the high-impedance state immediately after reading the mode data. (Pins with a pull-up resistor (optional) go to the "H" level.)

Check:

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

Reference:

See Appendix E "MB89980 Series Pin States" for pin states at times other than a reset.

3.6 Clocks

The clock generator is provided with two oscillators. By connecting with external resonators, the two circuits generate the high speed main clock and low speed subclock source oscillators. Alternatively, externally generated clock inputs can be used.
Clock controller controls the speed and supply of the dual-clock signals according to the clock and standby modes.
As an option, a one-clock system can also be selected.

■ Clock Supply Map

Oscillation of a clock and its supply to the CPU and peripheral circuit (peripheral functions) are controlled by the clock controller. As shown in the map, operating clocks fed to the CPU and peripheral circuits are affected by main clock/subclock switching (clock mode), main clock speed switching (speed-shift function), and standby modes (sleep/stop/watch).

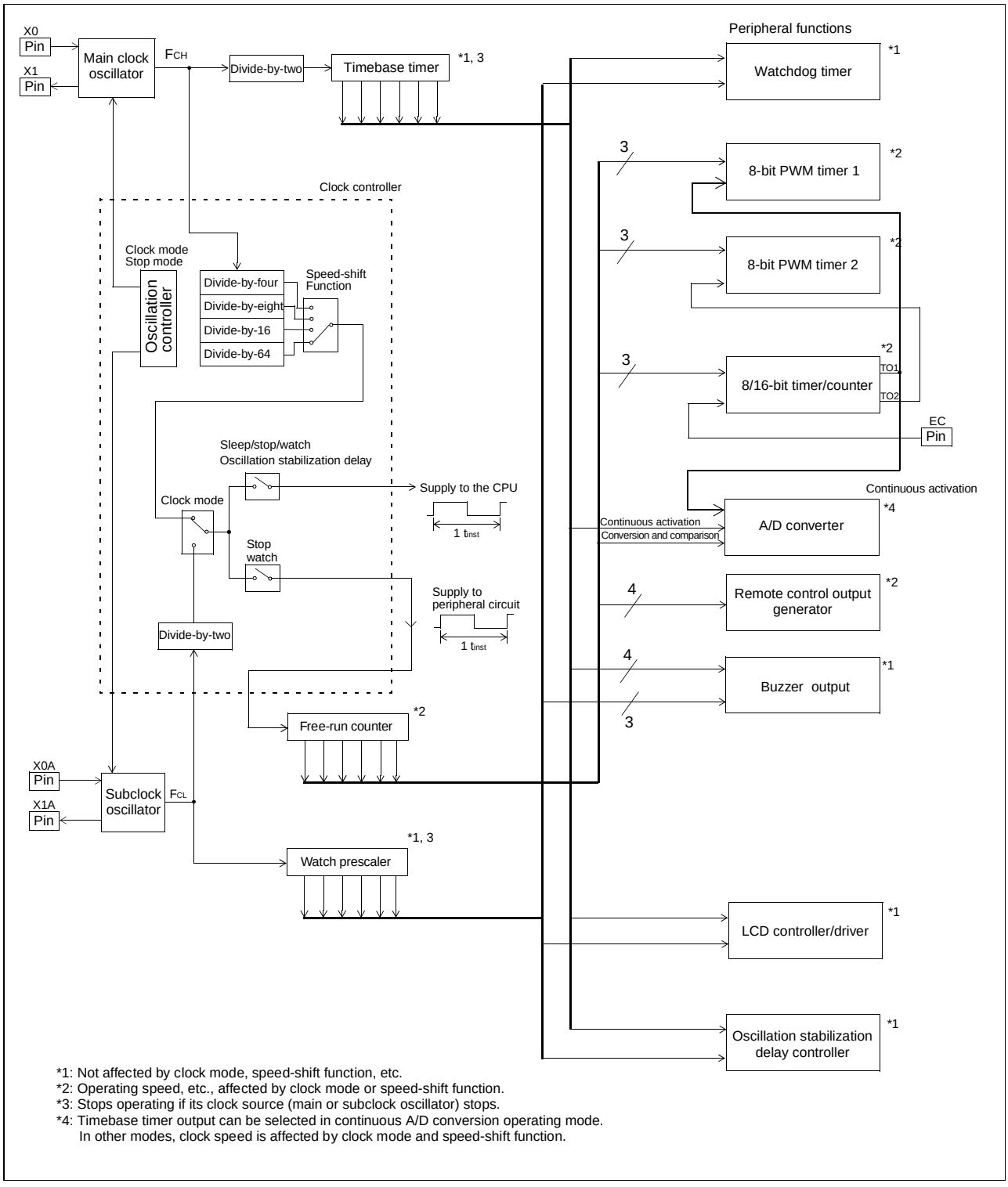
Divide-by-n output derived from the free-run counter clocked by the peripheral circuit clock is supplied to the peripheral functions.

Divide-by-n outputs from the timebase timer and watch prescaler are also supplied to the peripheral functions.

These clocks, however, are not affected by the speed-shift function, etc. The timebase timer is clocked by the output of the main clock source oscillator after it is fed through a divide-by-n circuit, and the watch prescaler is clocked directly by the subclock oscillator.

Figure 3.6-1 "Clock Supply Map" shows the clock supply map.

Figure 3.6-1 Clock Supply Map



*1: Not affected by clock mode, speed-shift function, etc.
*2: Operating speed, etc., affected by clock mode or speed-shift function.
*3: Stops operating if its clock source (main or subclock oscillator) stops.
*4: Timebase timer output can be selected in continuous A/D conversion operating mode.
In other modes, clock speed is affected by clock mode and speed-shift function.

3.6.1 Clock Generator

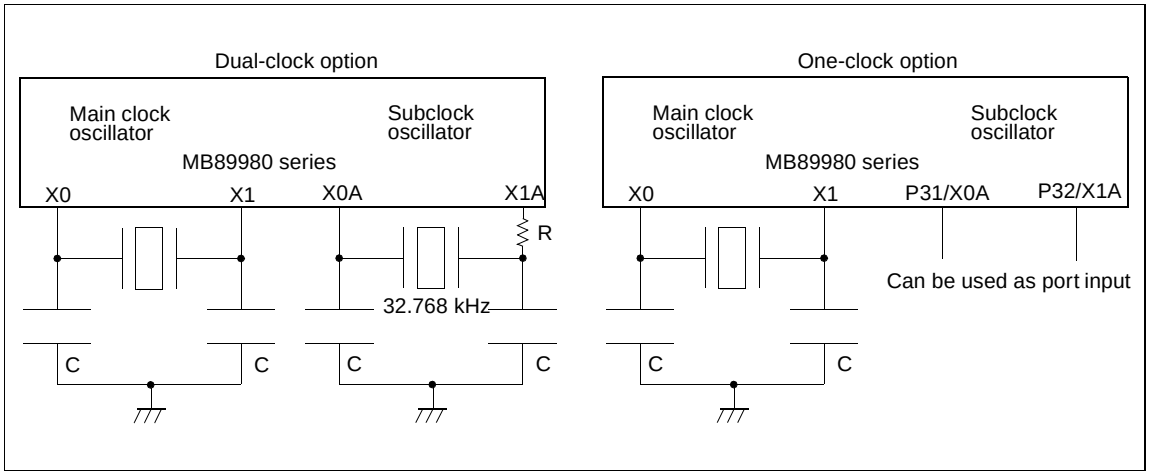
Enable and stop of the main clock and subclock oscillations are controlled by clock and stop modes.

■ Clock Generator

❍ Crystal or ceramic resonator

Connect as shown in Figure 3.6-2 "Connection Example for a Crystal or Ceramic Resonator".

Figure 3.6-2 Connection Example for a Crystal or Ceramic Resonator



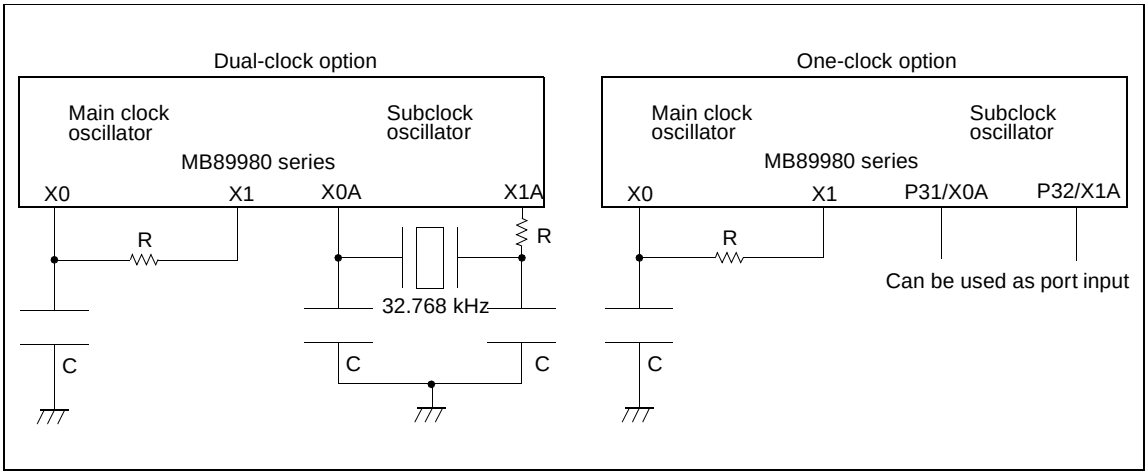
Note:

A piezoelectric resonator (FAR series) that contains the external capacitors can also be used.
See Data Sheet for details.

❍ CR (main clock only)

Connect as shown in Figure 3.6-3 "Connection Example for CR". External resistors and capacitors can only be used on devices with mask ROM.

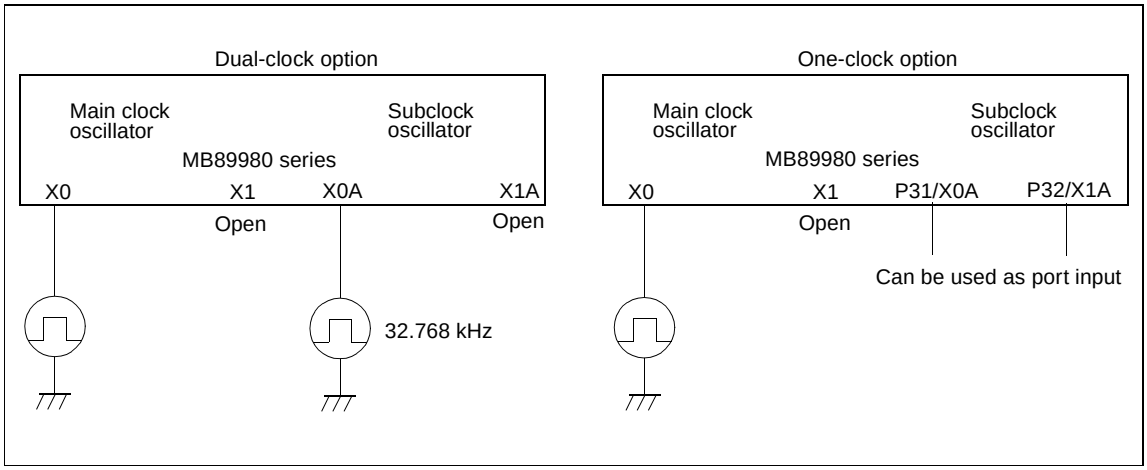
Figure 3.6-3 Connection Example for CR



❍ **External clock**

Connect the external clock to the X0 pin and leave X1 pin open, as shown in Figure 3.6-4 "Connection Example for External Clock" To use an external subclock source, connect the external clock to the X0A pin and leave the X1A pin open.

Figure 3.6-4 Connection Example for External Clock



Check:

In the MB89980 series, you can select a single clock system as an option. If only the main clock were to be used without the single clock option, there would be no way to recover once the system goes into subclock mode. Therefore, the single clock option must be selected in order to operate with one clock.

To use CR instead of a main clock oscillator, the device you are using must have mask ROM. For information on selecting mask ROM, See the Appendix C "Mask Options".

3.6.2 Clock Controller

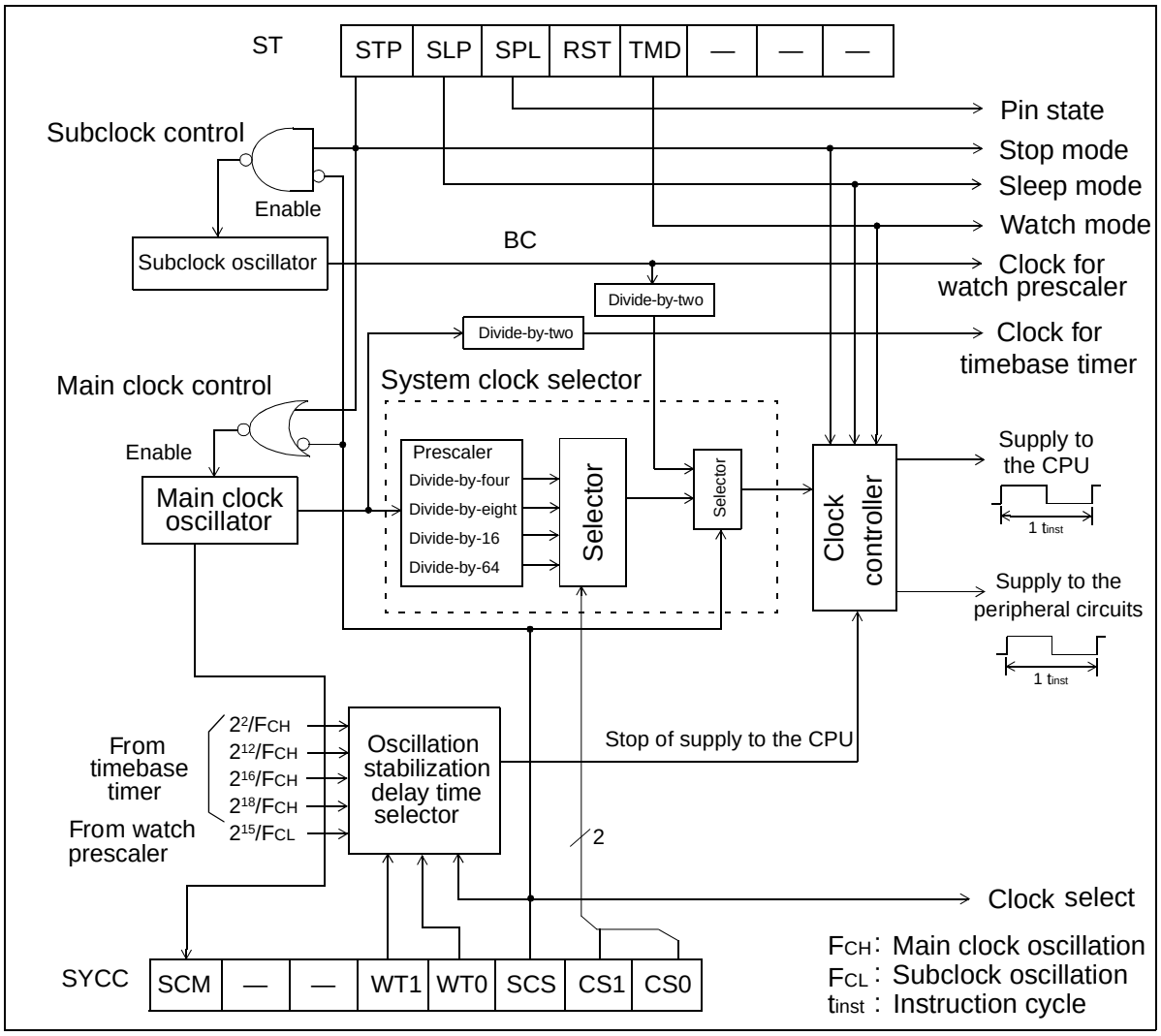
The clock controller contains the following seven blocks:

- **Main clock oscillator**
- **Subclock oscillator**
- **System clock selector**
- **Clock controller**
- **Oscillation stabilization delay time selector**
- **System clock control register (SYCC)**
- **Standby control register (STBC)**

■ Block Diagram of Clock Controller

Figure 3.6-5 "Block Diagram of Clock Controller" shows the block diagram of the clock controller.

Figure 3.6-5 Block Diagram of Clock Controller



- **Main clock oscillator**

  The main clock oscillator is stopped in main-stop and subclock modes.

- **Subclock oscillator**

  The subclock oscillator is normally running except in sub-stop mode. It does not operate in "one-clock" option devices.

- **System clock selector**

  The system clock selector selects one of five clocks: the subclock, or one of four divided clocks derived from the main clock master clock oscillator.

- **Clock controller**

  This circuit controls the supply of operating clocks to the CPU and peripheral circuits, selecting the clock based on the active mode: normal (RUN), or standby (sleep/stop/watch) mode.

  Supply of the clock to the CPU is stopped until the clock supply stop signal in the oscillation stabilization delay time selector is released.

- **Oscillation stabilization delay time selector**

  This register selector selects a delay time from among four main clock oscillation stabilization times timed by the timebase timer and a subclock oscillation stabilization time timed by the watch prescaler, and outputs the time as the clock supply stop signal to the CPU based on the clock mode, standby mode and reset.

- **SYCC register**

  The SYCC register is used to select the clock mode, the speed of the main clock, and the main clock oscillation stabilization delay time, and to check the status of these selections.

- **STBC register**

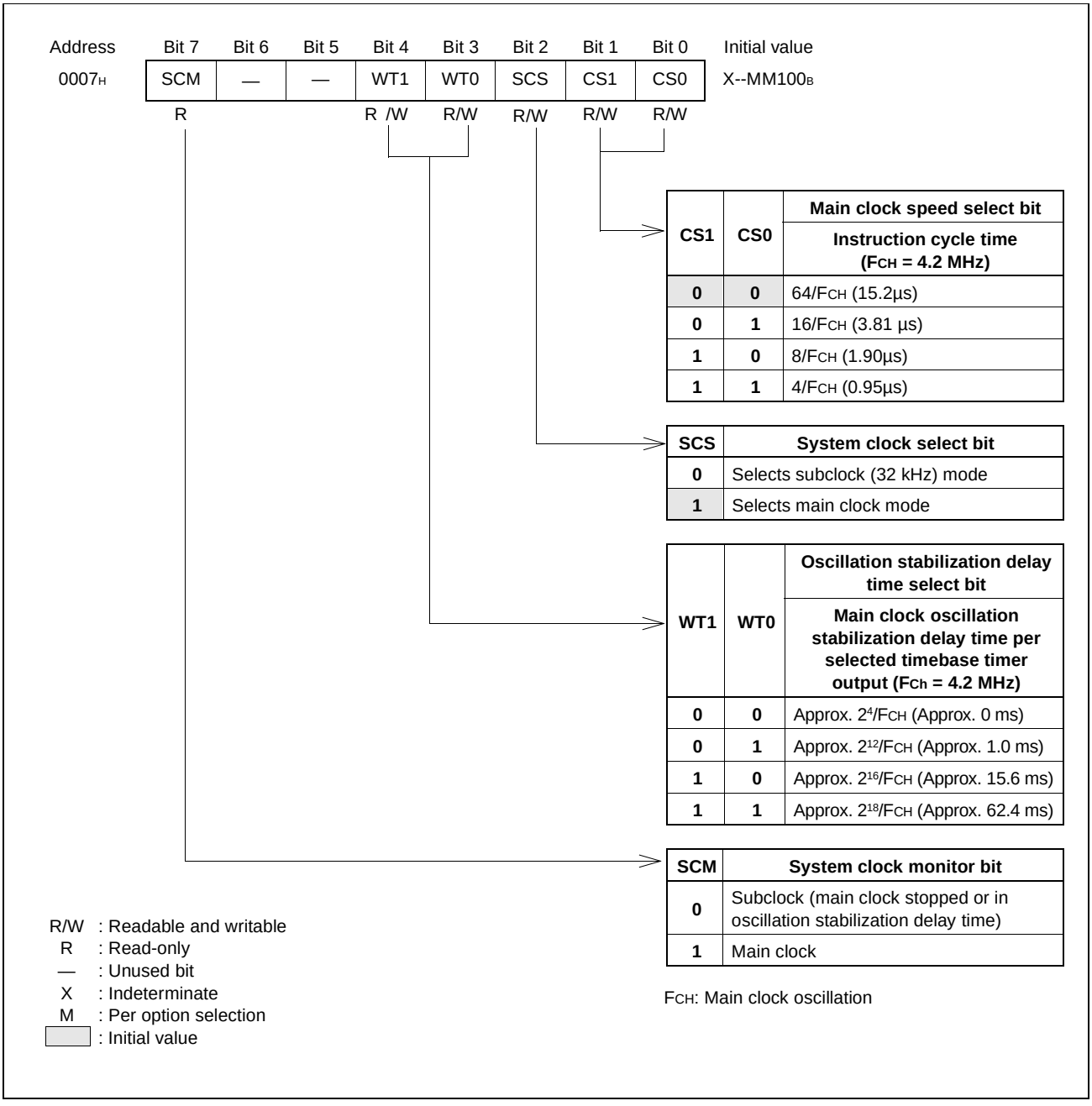
  This register controls from normal operation (RUN) to the standby modes, sets the pin states in the stop or watch mode, and initiates software reset.

3.6.3 System Clock Control Register (SYCC)

The system clock control register (SYCC) controls main clock/subclock switching, main clock speed selection, and oscillation stabilization delay time selection.

■ Structure of System Clock Control Register (SYCC)

Figure 3.6-6 Structure of System Clock Control Register (SYCC)



CS1	CS0	Main clock speed select bit Instruction cycle time (F_{CH} = 4.2 MHz)
0	0	64/F_{CH} (15.2µs)
0	1	16/F_{CH} (3.81 µs)
1	0	8/F_{CH} (1.90µs)
1	1	4/F_{CH} (0.95µs)

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

WT1	WT0	Oscillation stabilization delay time select bit Main clock oscillation stabilization delay time per selected timebase timer output (F_{Ch} = 4.2 MHz)
0	0	Approx. 2^4/F_{CH} (Approx. 0 ms)
0	1	Approx. 2^{12}/F_{CH} (Approx. 1.0 ms)
1	0	Approx. 2^{16}/F_{CH} (Approx. 15.6 ms)
1	1	Approx. 2^{18}/F_{CH} (Approx. 62.4 ms)

SCM	System clock monitor bit
0	Subclock (main clock stopped or in oscillation stabilization delay time)
1	Main clock

F_{CH}: Main clock oscillation

Table 3.6-1 System Clock Control Register (SYCC) Bits

Bit		Function
Bit 7	SCM: System clock monitor bit	• Indicates the current clock mode (operating clock). • "0" indicates subclock mode (main clock is stopped or in the oscillation stabilization delay time to go to main clock mode). • "0" indicates main clock mode. **Note:** This is a read-only bit. Writing to it has no effect.
Bit 6 Bit 5	Unused bits	• The read value is indeterminate. • Writing to these bits has no effect on operation.
Bit 4 Bit 3	WT1, WT0: Oscillation stabilization delay time select bits	• Select main clock oscillation stabilization delay time. • Selected wait time applies when going from subclock to main clock mode, or if external interrupt causes "wakeup" from main-stop mode (transition to normal (run) mode). • Initial value of these bits is an option selection. Accordingly, when an oscillation stabilization delay time is provided at reset, the delay time will be as selected by the option. **Note:** These bits should not be changed at the same time switching from subclock to main clock (SCS = 1 --> 0). Before changing the bits, first check the SCM bit to verify that the device is not currently in the stabilization delay time.
Bit 2	SCS: System clock select bit	• Specifies the clock mode. • Writing "0" to this bit sets the CPU changing from main clock to subclock mode. • Writing "1" to this bit causes the device to go from subclock to main clock mode after the oscillation stabilization delay time set by WT1 and WT0 bits. **Note:** If the single clock option is selected, this bit has no function. It should be set to "1".
Bit 1 Bit 0	CS1, CS0: Main clock speed select bits	• These bits select the clock speed for the main clock mode. • Four different speeds can be set for CPU and peripheral function operating clocks (speed-shift function). The clocks that clock the timebase timer and watch prescaler are not affected by these bits.

■ Instruction Cycle (tinst)

Instruction cycle (minimum execution time) can be selected as

1/4, 1/8, 1/16, or 1/64 of the main clock, or 1/2 of the subclock (32.768 kHz clock) period. The selection is made by the system clock select bit (SCS) and main clock speed select bits (CS1 and CS0) of the SYCC register.

With main clock mode, and the highest clock speed selected (SYCC: SCS = 1, CS1 = 11_B, CS0 = 11_B), and with a main clock source oscillation (F_{CH}) of 4.2 MHz, the instruction cycle is $4/F_{CH}$ = approximately 0.95 μs.

With subclock mode selected (SCS = 0), and with a subclock source oscillation (F_{CL}) of 32.768 kHz, the instruction cycle is $2/F_{CL}$= approximately 61.0 μs.

3.6.4 Clock Modes

The clock modes consists of main clock mode and subclock mode.
In the main clock mode, the primary operating clock is provided by the main clock oscillator. The speed of the operating clock is selected by switching between one of four clocks obtained by dividing the output of the main clock oscillator (speed-shift function).
In the subclock mode, the main clock oscillator is stopped, and operating clocks are provided solely by the subclock.

■ Clock Mode Operating States

Table 3.6-2 Clock Mode Operating States

Clock mode	Main clock speed SYCC register (SYCC: CS1, CS0)		Stand-by mode	Clock Oscillator		Operating clocks for various sections				Non-reset event triggering exit from stanby
				Main	Sub	CPU	Time-base timer	Peri-pherals	Watch prescaler	
Main clock mode	(1.1)	Fast	RUN	Oscillate	Oscillate	$F_{CH}/4$	$F_{CH}/2$	$F_{CH}/4$	F_{CL}	IRQ
			Sleep			Stop				
			Stop	Stop			Stop	Stop		External interrupt
	(1.0)		RUN	Oscillate	Oscillate	$F_{CH}/8$	$F_{CH}/2$	$F_{CH}/8$	F_{CL}	IRQ
			Sleep			Stop				
			Stop	Stop			Stop	Stop		External interrupt
	(0.1)		RUN	Oscillate	Oscillate	$F_{CH}/16$	$F_{CH}/2$	$F_{CH}/16$	F_{CL}	IRQ
			Sleep			Stop				
			Stop	Stop			Stop	Stop		External interrupt
	(0.0)		RUN	Oscillate	Oscillate	$F_{CH}/64$	$F_{CH}/2$	$F_{CH}/64$	F_{CL}	IRQ
			Sleep			Stop				
		Slow	Stop	Stop			Stop	Stop		External interrupt
Sub-clock mode	-		RUN	Stop	Stop	F_{CL}	Stop*	F_{CL}	F_{CL}	IRQ
			Sleep			Stop				
			Stop		Stop			Stop	Stop	External interrupt
			Watch Mode	Stop	Oscillate	Stop	Stop*	Stop	F_{CL}	External or watch interrupt

F_{CH} Main clock source oscillation

F_{CL} Subclock source oscillation

*: Since the timebase timer is derived from the main clock, it stops in subclock mode.

Reference:

See Section 3.7 "Standby Modes" for a description of the standby modes."

■ Speed-Shift (Main Clock Speed-switching) Function

One of four main clock frequencies can be selected by writing the appropriate values between "00B"and "11B" to main clock speed select bits of the system clock control register (SYCC: CS1, CS0).

The switch-selected clock signal provides the operating clock for the CPU and peripheral circuits. The timebase timer and watch prescaler, however, are not affected by the speed-shift (gear) function.

A slower main clock speed reduces power consumption.

■ Operation of Main Clock Mode

The main clock and the subclock oscillators both run in the "main-run" mode (the normal main clock operating mode). The watch prescaler runs on the subclock, but the CPU, timebase timer, and other peripheral circuits all use the main clock.

When operating in main clock mode, the speed-shift function can be used to select a main clock speed. This selection affects all circuits that are clocked by the main clock except for the timebase timer. By specifying a standby mode, you can also go to "main-sleep,"or "main-stop" mode.

When the device is reset, the system always starts out in "main-run" mode regardless of how the reset was initiated. (Each operating mode exited by reset.)

❍ Changing from main clock mode to subclock mode

Writing "0" to the system clock select bit in the system clock control register (SYCC:SCS) changes the CPU from the main clock to subclock mode. You can determine which clock is currently being used by checking the system clock monitor bit of the same register (SYCC: SCM).

Check:

If you go to subclock mode immediately after power on, write the software so as to provide a longer subclock oscillation stabilization delay time than that defined by the watch prescaler.

■ Operation of Subclock Mode

In the normal subclock operating mode ("sub-run" mode), the system runs on the subclock only. The main clock oscillator is stopped. Using the low speed subclock reduces power consumption.

Other than the timebase timer, all functions operate the same in subclock mode as they do in main clock mode. If standby mode is specified while operating in subclock mode, the device goes to "sub-sleep," "sub-stop," or "watch" mode.

❍ Returning to main clock mode from subclock mode

Writing "1" to the system clock select bit in the system clock control register (SYCC:SCS) returns to main clock mode from subclock mode.

Operation from the main clock, however, will not start until after the main clock oscillation stabilization delay time has passed. One of four wait times can be selected by setting the

oscillator stabilization delay time select bits of the system clock control register (SYCC: WT1, WT0).

Check:

Do not change the oscillation stabilization delay time select bits (SYCC: WT1, WT0) at the same time you switch from subclock to main clock mode (SYCC: SCS = 1), or during the oscillator stabilization delay time. Always check the system clock monitor bit to verify that the main clock is the active operating clock (SYCC: SCM = 1) before changing these bits.

If the device has the power-on reset option it always enters the oscillator stabilization delay time when the system is returned from subclock mode to main clock mode by reset. If the device does not have power-on reset option, there will be no delay time unless the reset was a software or watchdog reset.

3.6.5 Oscillation Stabilization Delay Time

When the system transitions to main-run mode from a state in which the main clock is stopped (such as at power-on, and in main-stop and subclock modes, etc.), a delay time is required for oscillation to stabilize before operation starts.
Similarly, a subclock oscillation stabilization wait time is required when exiting the sub-stop mode, because the subclock oscillator is stopped in that mode.

■ Oscillation Stabilization Delay Time

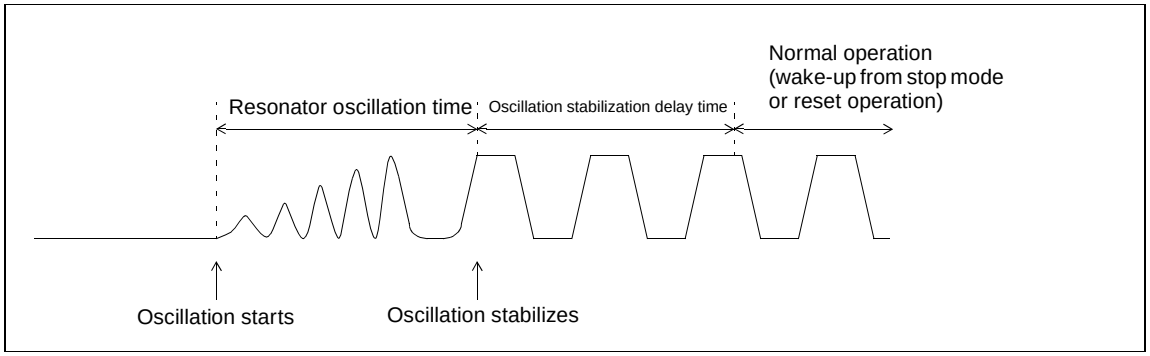
After starting, ceramic, crystal, and other resonators typically require the time between several milliseconds and several tens of milliseconds to stabilize at their fixed oscillation frequency.

Therefore, operation of the CPU and other functions is disabled when oscillation first starts and no clock signal is supplied to the CPU and peripheral functions until the oscillation stabilization delay time has passed and the oscillation has sufficiently stabilized.

The time required for oscillation to stabilize depends on the resonator type (crystal, ceramic, etc.) connected to the clock generator. Consequently, it is necessary to select an oscillation stabilization delay time that matches the type of oscillator being used.

Figure 3.6-7 "Operation of Oscillator after Starting Oscillation" shows the operation of an oscillator after starting oscillation.

Figure 3.6-7 Operation of Oscillator after Starting Oscillation



■ Main Clock Oscillation Stabilization Delay Time

When first starting operation in main clock mode after a state in which the main clock oscillator is stopped, a delay time is required for oscillation to stabilize. This delay time starts when the timebase timer starts counting up from its cleared state, and ends when the count overflows at the specified bit.

❍ Oscillation stabilization delay time during operation

A time length must be selected for the oscillation stabilization delay time when an external interrupt takes the system from main-stop mode back to main-run mode, or when going from subclock to main clock mode. One of four possible delay times can be selected, using the oscillator stabilization delay time select bits of the system clock control register (SYCC: WT1, WT0).

❍ **Oscillation stabilization delay time at reset**

The oscillation stabilization delay time at reset (the initial values of WT1 and WT0) is selected as an option setting.

Products with power-on reset require an oscillation stabilization delay time when exit from stop mode is triggered by resets in subclock mode (multiple), power-on reset, or external reset.

Products without power-on reset only require an oscillation stabilization delay time for watchdog reset or software reset during subclock mode.

Table 3.6-3 "Main Clock Startup Conditions vs. Oscillation Stabilization Delay Time" shows the relationships between the conditions in which main clock mode operation is started and oscillation stabilization delay time.

Table 3.6-3 Main Clock Startup Conditions vs. Oscillation Stabilization Delay Time

Main clock mode startup conditions	At power-on	During subclock mode		Exit from main-stop		Transition from subclock to main clock mode (SYCC: SCS*1=1)
		External reset	Software reset and watchdog reset	External reset	External interrupt	
Oscillation stabilization delay time selection	Option setting				SYCC: WT1, WT0 *2	
With power-on reset	O	O	O	O	O	O
No power-on reset	X	X	O	X	O	O

O: Oscillation stabilization delay time provided

X: Oscillation stabilization delay time not provided

*1 System clock select bit of system clock control register

*2 Oscillation stabilization delay time select bits of system clock control register

■ **Subclock Oscillation Stabilization Delay Time**

When an external interrupt returns the system from sub-stop (subclock oscillator stopped) to sub-run mode (thus starting the subclock oscillator), a set subclock oscillation stabilization delay time is provided. (This set delay time is equal to $2^{15}/F_{CL}$, where F_{CL} is the subclock oscillator frequency.)

The subclock oscillation stabilization delay time is also entered at power-on. Therefore, if you go to subclock mode after power on, you should insert a software delay, to provide a longer delay time before starting this transition than the subclock oscillation stabilization delay time alone.

The subclock oscillation stabilization delay time starts when the watch prescaler starts counting up from the cleared state, and ends when it overflows.

3.7 Standby Modes (Low-Power Consumption)

**The standby modes consist of sleep mode, stop mode, and watch mode.
From both main and subclock clock modes, standby modes are changed to sleep mode, stop mode, or watch mode by setting the standby control register (STBC).
From main clock mode, you can go only to sleep or stop mode, but from subclock mode you can go any of the three standby modes. Standby mode reduces the power consumption by stopping the operation of the CPU and peripheral functions.
This section describes the relationship between standby mode and clock mode, and the operation of various sections during standby.**

■ Standby Modes

Watch mode reduce the power consumption by lowering the frequencies of the clocks that run the CPU and peripheral circuits. You can do this by switching from main clock to subclock mode, or by using the speed-shift function to select a lower main clock frequency. Standby mode reduce the power consumption, however, by stopping the clock signal supply to the CPU via clock controller (sleep mode), by stopping the clock signal supply to the CPU and peripheral circuits (watch mode), or by stopping the source oscillator itself (stop mode).

❍ Main-sleep Mode

Main-sleep mode stops the CPU and watchdog timer, but operate the peripheral functions except watch prescaler by the main clock. (Certain functions can still run on the subclock.)

❍ Sub-sleep Mode

Sub-sleep mode stops the main clock oscillator, CPU, watchdog timer, and timebase timer, but operate the peripheral functions on the subclock.

❍ Main-stop Mode

Main-stop mode stops the CPU and peripheral functions. The main clock oscillator is stopped, but the subclock oscillator keeps running. Everything is shut down except external interrupt servicing, watch prescaler counter operation, and some functions that operate from the subclock.

❍ Sub-stop Mode

Sub-stop mode stops all chip functions except the external interrupt, and stops the main clock and subclock oscillations.

❍ Watch Mode

You can only go to watch mode from the subclock clock mode. All functions are shut down except the watch prescaler (watch interrupt), external interrupts, and some functions that operate from the subclock.

Even when the main clock is stopped, as it is in the main-stop a

Even when the main clock is stopped, as it is in the main-stop and watch modes, as long as the subclock oscillation is still operating, LCD controller/drivers will still operate. See CHAPTER 14 "LCD CONTROLLER/DRIVER" and the clock supply map provided in Section 3.6 "Clocks" for details.

3.7.1 Operating States in Standby Modes

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

■ Operating States during Standby Modes

Table 3.7-1 Operating States of the CPU and Peripheral Functions in Standby Modes

Function		Main clock mode			Subclock mode			
		RUN	Sleep	Stop	RUN	Sleep	Stop	Watch
Main clock		Operating	Operating	Stop	Stop	Stop	Stop	Stop
Subclock		Operating	Operating	Operating	Operating	Operating	Stop	Operating
CPU	Instructions	Operating	Stop	Stop	Operating	Stop	Stop	Stop
	ROM	Operating	Hold	Hold	Operating	Hold	Hold	Hold
	RAM							
Peripheral Functions	I/O ports	Operating	Hold	Hold	Operating	Hold	Hold	Hold
	Watch prescaler	Operating	Operating	Operating*1	Operating	Operating	Stop	Operating
	Timebase timer	Operating	Operating	Stop	Stop	Stop	Stop	Stop
	8/16-bit timer/ counter	Operating	Operating	Stop	Operating	Operating	Stop	Stop
	Remote control output frequency generator	Operating	Operating	Stop	Operating	Operating	Stop	Stop
	LCD controller-driver	Operating	Operating	Operating*2	Operating*2	Operating*2	Stop	Operating*2
	8-bit PWM timers	Operating	Operating	Stop	Operating	Operating	Stop	Stop
	A/D converter	Operating	Operating	Stop	Operating	Operating	Stop	Stop
	External interrupts 1 and 2	Operating	Operating	Operating	Operating	Operating	Operating	Operating
	Buzzer output	Operating	Operating	Operating*3	Operating*3	Operating*3	Stop	Operating*3
	Watchdog timer	Operating	Stop	Stop	Operating*3	Stop	Stop	Stop

*1: Watch prescaler counts but does not generate watch interrupts.

*2: If the subclock is selected as the operating clock, in watch mode, LCD controller/driver operation must be enabled.

*3: Can be operated if watch prescaler output is selected as its operating clock.

❍ Pin States in Standby Mode

Almost all I/O pins will either keep the state they were placed in by the pin state control bit of the standby control register (STBC: SPL) just prior to going to the stop or watch mode, or will go to the high impedance state. This is true regardless of the clock mode.

Reference:

See Appendix E "MB89980 Series Pin States." for pin states in a standby mode.

3.7.2 Sleep Mode

This section describes the operations of sleep mode.

■ Operation of Sleep Mode

❍ Changing to sleep mode

Sleep mode stops the CPU operating clock. The CPU stops while maintaining all register contents, RAM contents, and pin states at their values immediately prior to entering sleep mode. However, peripheral functions except the watchdog timer continue to operate.

If the system is in subclock mode, however, the main clock oscillation is stopped; and because the timebase timer operates on a divide-by-two version of the main clock source oscillation, it also stops operating.

Writing "1" to the sleep bit in the standby control register (STBC: SLP) changes the CPU to sleep mode. If an interrupt request is generated when "1" is written to the SLP bit, the write to the bit is ignored, and the CPU continues the instruction execution without change to sleep mode. (The CPU does not change to sleep mode even after completion of the interrupt processing.)

❍ Wake-up from sleep mode

A reset or an interrupt from a peripheral function wakes up the CPU from sleep mode.

If a reset occurs during sub-sleep mode on a product with power-on reset, the reset operation starts after the main clock oscillation stabilization delay time.

In products without power-on reset, or if the reset occurs in main-sleep mode, there is no oscillation stabilization delay period.

The reset operation also initializes the pin states.

If an interrupt request with an interrupt level higher than "11" occurs from a peripheral function or an external interrupt circuit during sleep mode, the CPU wakes up from sleep mode, regardless of the interrupt enable flag (CCR: I) and interrupt level bits (CCR: IL1 and IL0) in the CPU.

The normal interrupt operation is performed after wake-up from sleep mode. If the interrupt request is accepted, the CPU executes interrupt processing. If the interrupt request is not accepted, the CPU continues execution from the subsequent instruction following the instruction executed immediately before changing to sleep mode.

3.7.3 Stop Mode

This section describes the operations of stop mode.

■ Operation of Stop Mode

❍ Changing to stop mode

Stop mode the source oscillation. Almost functions stop while maintaining all register and RAM contents at their value immediately before changing to stop mode.

If the system is in main clock mode, the main clock oscillation stops, but the subclock oscillation continues to run. This means that the watch prescaler can still count and some functions that run on the subclock can still function. Except the external interrupt circuit, however, the CPU and other peripheral functions stop operating.

If the system is in subclock mode, both the main and subclock oscillations are stopped. All chip functions other than external interrupt circuits stop. Accordingly, data can be held with minimum power consumption.

Writing "1" to the stop bit in the standby control register (STBC: STP) changes the CPU to stop mode. At this time, external pin states are held if the pin state specification bit (STBC: SPL) is "0" If SPL is "1" external pins go to the high-impedance state. (Pins with the pull-up resistor (optional) go to the "H" level.)

If an interrupt request is generated when "1" is written to the STP bit, the write to the bit is ignored, and the CPU continues the instruction execution without change to stop mode. (The CPU does not assume stop mode even after completion of the interrupt processing.)

Prohibit interrupt request out from the timebase timer (TBTC: TBIE = "0") before changing to stop mode in main clock mode as necessary. Similarly, prohibita timeclock interrupt request output from the watch prescaler (WPCR: WIE = "0") before changing to stop mode in subclock mode.

❍ Wake-up from stop mode

A reset or an external interrupt wakes up the CPU from stop mode.

If reset occurs during stop mode on a product with power-on reset, the reset operation starts after the main clock oscillation stabilization delay time. Products without power-on reset do not require for the oscillation stabilization delay time after a reset in stop mode. The reset initializes pin states.

If an interrupt request with an interrupt level higher than "11" occurs from an external interrupt circuit during stop mode, the CPU wakes up from stop mode, regardless of the interrupt enable flag (CCR: I) and interrupt level bits (CCR: IL1, IL0) in the CPU. Only external interrupt requests can occur during stop mode because peripheral functions are stopped. In main-stop mode, the watch prescaler operates, but it does not generate watch interrupts.

After wake-up from stop mode, the normal interrupt operation is performed after the oscillation stabilization delay time has passed. If the interrupt request is accepted, the CPU executes interrupt processing. If the interrupt request is not accepted, the CPU continues execution from the subsequent instruction following the instruction executed immediately before entering stop mode.

Some peripheral functions restart from mid-operation when the CPU wakes up from stop mode by an external interrupt. The first interval time from the interval timer function, for example, is indeterminate. Therefore, initialize all peripheral functions after wake-up from stop mode.

Check:

Only interrupt requests from external interrupt circuits can be used to wake up from stop mode by an interrupt.

3.7.4 Watch Mode

This section describes the operations of watch mode.

■ Operation of watch Mode

❍ Changing to watch mode

Watch mode stops the clocks that clock the CPU and the main peripheral functions. You can go to watch mode only from subclock mode (in which the main clock oscillation is stopped).

Prior to going to watch mode, registers are saved and the contents of RAM are held. All chip functions other than watch prescaler (timeclock interrupt), external interrupt circuit, and certain functions that run off of the subclock stop. Accordingly, data can be held with extremely small power consumption.

Writing "1" to the timeclock bit in the standby control register (STBC: TMD) changes the CPU to watch mode.

This can be done, however, only when the system clock select bit of the system clock control register (SYCC: SCS) is "0" (subclock mode active).

When you go to watch mode, external pin states are held if the pin state specification bit in the standby control register (STBC: SPL) is "0" If SPL is "1" external pins go the high-impedance state. (Pins with a pull-up resistor (optional) go to the "H" level)

If an interrupt request is generated when "1" is written to the TMD bit, the write to the bit is ignored, and the CPU continues the instruction execution without change to watch mode. (The CPU does not assume watch mode even after completion of the interrupt processing.)

❍ Wake-up from watch mode

A reset, a timeclock interrupt or an external interrupt wakes up CPU from watch mode.

If a reset occurs during watch mode on a product with power-on reset, the reset operation starts after the main clock oscillation stabilization delay time.

Products without power-on reset do not require for the oscillation stabilization delay time after a reset in watch mode.

The reset initializes pin states.

If an interrupt request with an interrupt level higher than "11" occurs from a watch prescaler or an external interrupt circuit during watch mode, the CPU wakes up from watch mode, regardless of the interrupt enable flag (CCR:I) and interrupt level bits ((CCR: IL1, IL0) in the CPU. Only timeclock or external interrupt requests can occur during watch mode because most of the peripheral functions except watch prescaler are stopped.

After wake-up from stop mode, the normal interrupt operation is performed. If the interrupt request is accepted, the CPU executes interrupt processing. If the interrupt request is not accepted, the CPU continues execution from the subsequent instruction following the instruction executed immediately before entering watch mode.

Some peripheral functions restart from mid-operation when the CPU wakes up from watch mode. The first interval time from the interval timer function, for example, is indeterminate. Therefore,initialize all peripheral functions after wake-up from watch mode.

3.7.5 Standby Control Register (STBC)

The standby control register (STBC) controls the changing to sleep mode, stop mode, or watch mode, sets the pin states in stop mode and watch mode, and initiates software resets.

■ Standby Control Register (STBC)

Figure 3.7-1 Standby Control Register (STBC)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0008H	STP	SLP	SPL	RST	TMD	—	—	—	00010---B
	W	W	R/W	W	W				

TMD	Watch bit	
	Valid only in subclock mode (SYCC: SCS = 0)	
	Read	Write
0	Reading always returns "0".	No effect on operation
1	—	Changing to watch mode

RST	Software reset bit	
	Read	Write
0	—	Generates a reset signal for four instruction cycles.
1	Reading always returns "1".	No effect on operation

SPL	Pin state specification bit
0	External pins hold their states prior to entering stop mode or watch mode.
1	External pins go to high-impedance state on entering stop mode or watch mode.

SLP	Sleep bit	
	Read	Write
0	Reading always returns "0".	No effect on operation
1	—	Change to sleep mode.

STP	Stop bit	
	Read	Write
0	Reading always returns "0".	No effect on operation
1	—	Change to stop mode.

R/W : Readable and writable
W : Write-only
— : Unused
X : Indeterminate
: Initial value

Table 3.7-2 Standby Control Register (STBC) Bits

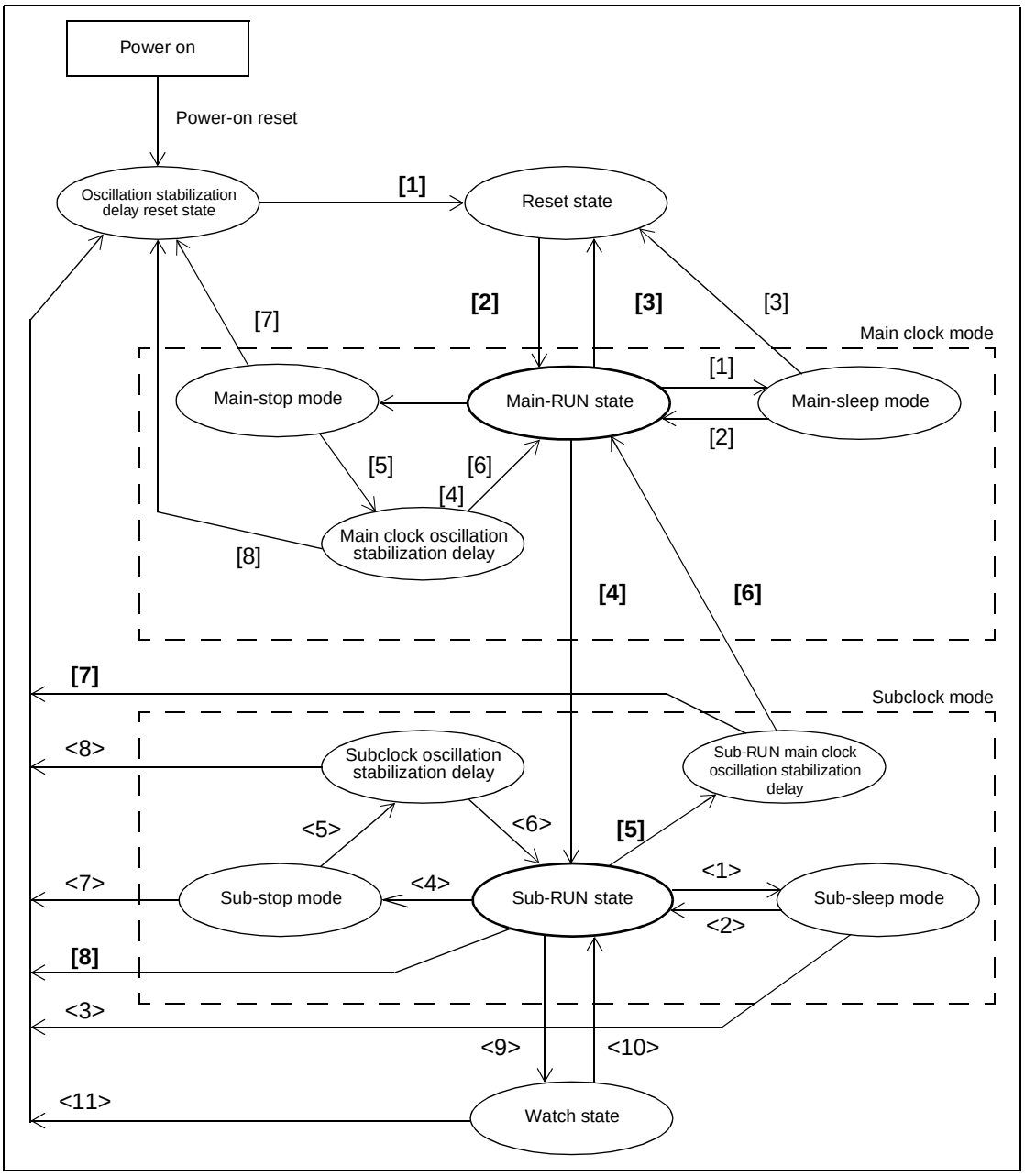
Bit		Function
Bit 7	STP: Stop bit	• Sets the CPU changing to stop mode. • Writing "1" to this bit sets the CPU changing to stop mode. • Writing "0" to this bit has no effect on operation. • Reading this bit always returns "0".
Bit 6	SLP: Sleep bit	• Sets the CPU changing to sleep mode. • Writing "1" to this bit sets the CPU changing to sleep mode. • Writing "0" to this bit has no effect on operation. • Reading this bit always returns "0".
Bit 5	SPL: Pin state specification bit	• Specifies the states of the external pins during stop mode and watch mode. • Writing "0" to this bit specifies that external pin hold their states (levels) on changing to stop mode or watch mode. • Writing "1" to this bit specifies that external pins to go to high impedance state on entering stop mode or watch mode. (Pin with a pull-up resistor (optional) go to "H" level.) • Initialized to "0" by a reset.
Bit 4	RST: Software reset bit	• Specifies a software reset. • Writing "0" to this bit generates an internal reset source for four instruction cycles. • Writing "1" to this bit has no effect on operation. • Reading this bit always returns "1". **Note:** When the software reset is applied in subclock mode, operation will start up in main clock mode after an oscillation stabilization delay time. For this reason, if the reset output option is selected, the $\overline{\text{RST}}$ signal will be output during the oscillation stabilization delay time.
Bit 3	TMD: Watch bit	• Sets the CPU changing to watch mode. • A write to this bit is valid only in subclock mode (SYCC: SCS = 0). • Writing "1" to this bit sets the CPU changing to watch mode. • Writing "0" to this bit has no effect on operation. • Reading this bit always returns "0".
Bit 2 Bit 1 Bit 0	Unused bits	• The read value is indeterminate. • Writing to these bits has no effect on operation.

3.7.6 State Transition Diagram 1 (Options: Power-on Reset, Two Clocks)

This section shows the state transition diagram for products with power-on reset and dualclock options.

■ State Transition Diagram 1 (Options: Power-on Reset, Two Clocks)

Figure 3.7-2 State Transition Diagram 1 (Options: Power-on Reset, Two Clocks)



❍ Changing to/wake-up from clock modes (non-standby modes)

Table 3.7-3 Changing to/wake-up from Clock Modes (Options: Power-on Reset, Two Clocks)

State transition	Conditions/events required to transition
Changing to main-RUN state (normal main clock mode) after power-on reset	[1] Main clock oscillation stabilization delay time complete. (Timebase timer output) [2] Wake up from reset input.
Reset in main-RUN state	[3] Have external, software, or watchdog reset.
Changing from main-RUN state to sub-RUN state	[4] SYCC: SCS=0*
Changing from sub-RUN state back to main-RUN state	[5] SYCC: SCS=1 [6] Main clock oscillation stabilization delay time complete. (Can be checked by looking at SYCC: SCM) [7] Have external, software, or watchdog reset.
Reset in sub-RUN state	[8] Have external, software, or watchdog reset.

SYCC: System clock control register

*: Changing to sub-RUN state at power-on occurs after the subclock oscillation stabilization delay time complete.

❍ **Changing to/wake-up from standby modes**

Table 3.7-4 Changing to/wake-up from Standby Modes (Options: Power-on Reset, Two Clocks)

State transition	Conditions/events required to transition	
	Main clock mode	**Subclock mode**
Changing to sleep mode	[1] STBC: SLP = 1	<1> STBC: SLP = 1
Wake-up from sleep mode	[2] Interrupt (any) [3] External reset	<2> Interrupt (any) <3> External reset
Changing to stop mode	[4] STBC: STP = 1	<4> STBC: STP = 1
Wake-up from stop mode	[5] External interrupt [6] Main clock oscillation stabilization delay time complete. (Have timebase timer output.) [7] External reset [8] External reset (during oscillation stabilization delay time)	<5> External interrupt <6> Subclock oscillation stabilization delay time ends.(Have watch prescaler output.) <7> External reset <8> External reset (during oscillation stabilization delay time)
Changing to watch mode	-	<9> STBC: TMD = 1*
Wake-up from watch mode	-	<10> External or watch interrupt <11> External reset

STBC: Standby control register

*: Changing to watch mode is possible only from sub-RUN state (SYCC: SCS = 0).

Note:

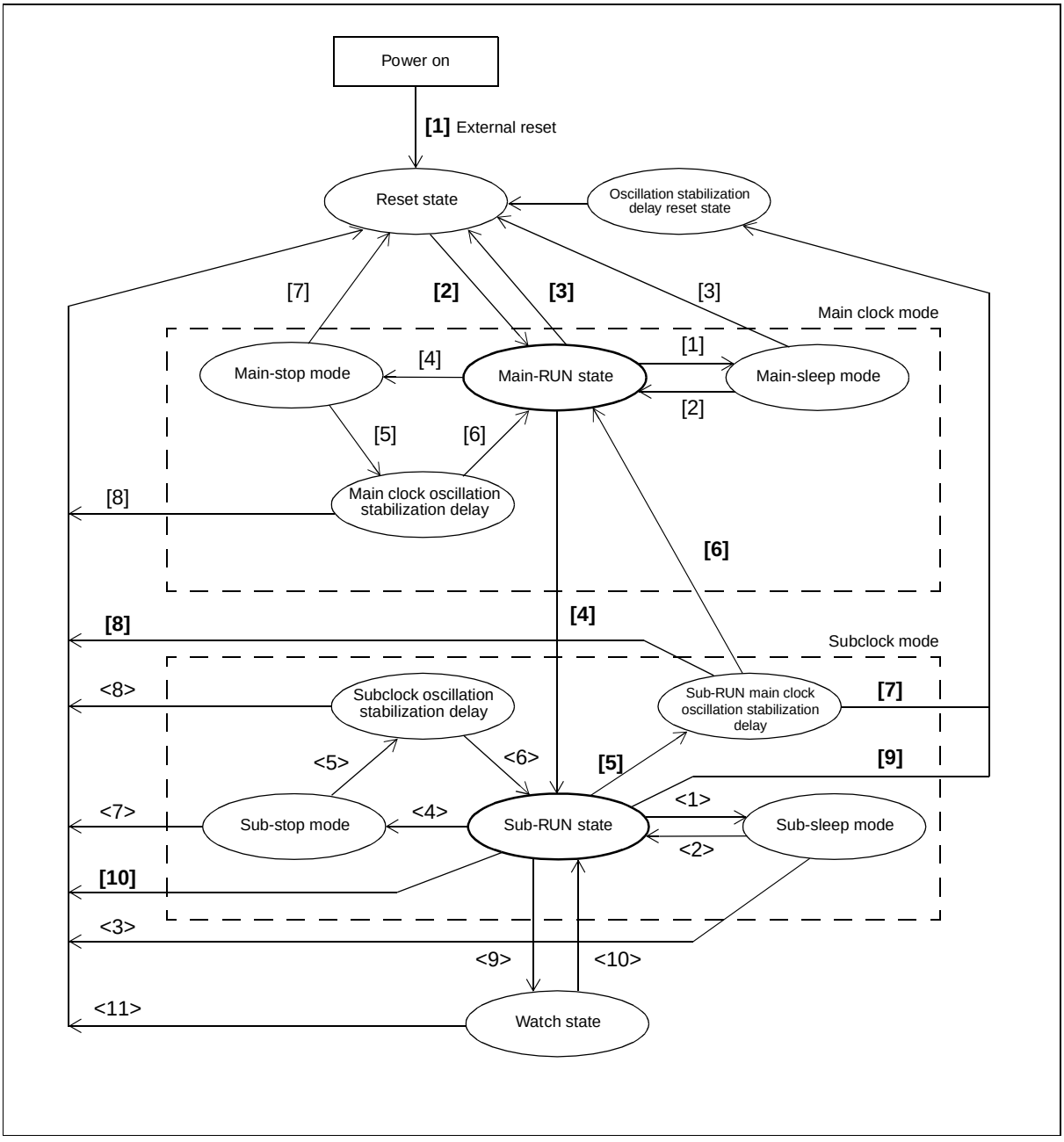
Neither software nor watchdog resets can occur during standby because the CPU and watchdog timer are both stopped.

3.7.7 State Transition Diagram 2 (Options: No Power-on Reset, Two Clocks)

This section shows the state transition diagram for products without power-on reset and dualclock options.

■ State Transition Diagram 2 (Options: Without Power-on Reset, Two Clocks)

Figure 3.7-3 State Transition Diagram 2 (Options: Without Power-on Reset, Two Clocks)



❍ **Changing to/wake-up from clock modes (non-standby modes)**

Table 3.7-5 Changing to/wake-up from Clock Modes (Options: Without Power-on Reset, Two Clocks)

State transition	Conditions/events required to transition	
Changing to main-RUN state (normal main clock mode) after external reset	[1]	External reset input must be held asserted until main clock oscillation has had time to stabilize.
	[2]	Wake-up from Reset input.
Reset in main-RUN state	[3]	Have external, software, or watchdog reset.
Changing from main-RUN state to sub-RUN state	[4]	SYCC: SCS=0*
Changing from sub-RUN state back to main-RUN state	[5]	SYCC: SCS=1
	[6]	Main clock oscillation stabilization delay time complete. (Can be checked by looking at SYCC: SCM)
	[7]	Have software or watchdog reset.
	[8]	Have external reset.
Reset in sub-RUN state	[9]	Have software or watchdog reset
	[10]	Have external reset

SYCC: System clock control register

*: Changing to sub-RUN state at power-on occurs after the subclock oscillation stabilization delay time complete.

❍ **Changing to/wake-up from standby modes**

Table 3.7-6 Changing to/Wake-up from Standby Modes (Options: Without Power-on Reset, Two Clocks)

State transition	Conditions/events required to transition	
	Main clock mode	**Subclock mode**
Changing to sleep mode	[1] STBC: SLP = 1	<1> STBC: SLP = 1
Wake-up from sleep mode	[2] Interrupt (any) [3] External reset	<2> Interrupt (any) <3> External reset
Changing to stop mode	[4] STBC: STP = 1	<4> STBC: STP = 1
Wake-up from stop mode	[5] External interrupt [6] Main clock oscillation stabilization delay time complete. (Have timebase timer output.) [7] External reset [8] External reset (during oscillation stabilization delay time)	<5> External interrupt <6> Subclock oscillation stabilization delay time complete. (watch prescaler output.) <7> External reset <8> External reset (during oscillation stabilization delay time)
Changing to watch mode	-	<9> STBC: TMD = 1*
Wake-up from watch mode	-	<10> External or watch interrupt <11> External reset

STBC: Standby control register

*: Changing to watch mode is possible only from sub-RUN state (SYCC: SCS = 0).

Check:

In all states except the main clock mode normal run (main-RUN) and sleep states, the external reset input must be held asserted until main clock oscillation has had time to stabilize.

3.7.8 State Transition Diagram 3 (One-clock Option)

This section shows two state transition diagrams for one-clock option products: one diagram for "with power-on reset" option products and one for "without power-on reset" products. There are no subclock or watch modes when one clock is used.

■ State Transition Diagram 3 (One-clock Option)

Figure 3.7-4 State Transition Diagram 3 (Products with Power-on Reset)

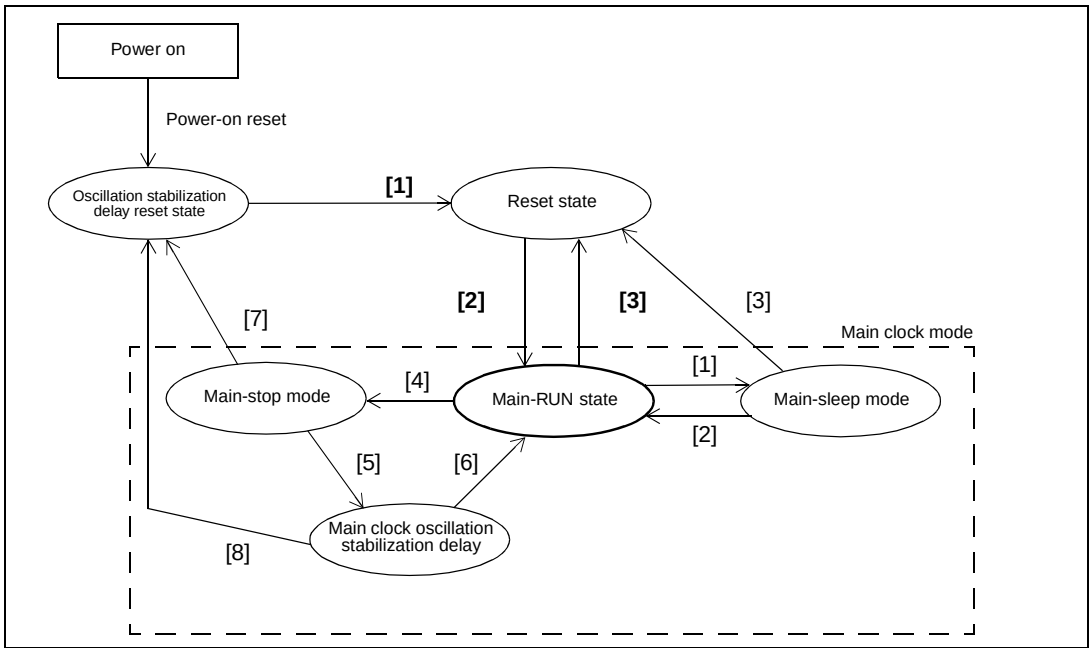
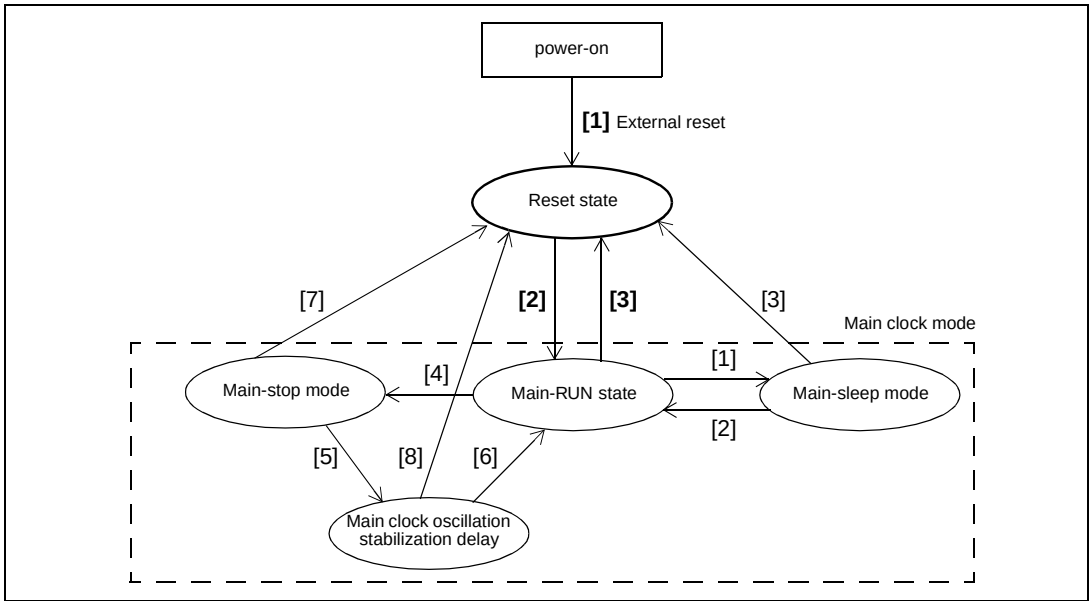


Figure 3.7-5 State Transition Diagram 3 (Products without Power-on Reset)



❍ Changing to normal state (RUN) and reset

Table 3.7-7 Changing to Main Clock Mode Run State and Reset (One-Clock Option)

State transition	Conditions/events required to transition	
	Products with power-on reset (Figure 3.7-4 "State Transition Diagram 3 (Products with Power-on Reset)")	**Products without power-on reset (Figure 3.7-5 "State Transition Diagram 3 (Products without Power-on Reset)")**
Changing to normal state (RUN) after power-on	[1] Main clock oscillation stabilization delay time complete. (Timebase timer ouptput.) [2] Wake-up from Reset input.	[1] External reset input must be held asserted until main clock oscillation has had time to stabilize. [2]
Reset in RUN state	[3] Have external, software, or watchdog reset.	[3] Have external, software, or watchdog reset.

❍ Changing to/wake-up from standby mode

Table 3.7-8 Changing to/wake-up from Standby Modes (Options: Power-on Reset, Two Clocks)

State transition	Conditions/events required to transition	
	Products with power-on reset (Figure 3.7-4 "State Transition Diagram 3 (Products with Power-on Reset)")	**Products without power-on reset (Figure 3.7-5 "State Transition Diagram 3 (Products without Power-on Reset)")**
Changing to sleep mode	[1] STBC: SLP = 1	[1] STBC: SLP = 1
Wake-up from sleep mode	[2] Interrupt [3] External reset	[2] Interrupt [3] External reset
Changing to stop mode	[4] STBC: STP = 1	[4] STBC: STP = 1
Wake-up from stop mode	[5] External interrupt [6] Main clock oscillation stabilization delay time complete. (Timebase timer output.) [7] External reset [8] External reset (during oscillation stabilization delay time)	[5] External interrupt [6] Main clock oscillation stabilization delay time complete. (Timebase timer [7] output.) [8] External reset External reset (during oscillation stabilization delay time)

STBC: Standby control register

3.7.9 Notes on Using Standby Modes

The CPU does not change to a standby mode if an interrupt request occurs from a peripheral function when a standby mode is set in the standby control register. (STBC) Also, if an interrupt is used to wake up from a standby mode to the normal operating state, the operation after wake-up differs depending on whether or not the interrupt request is accepted.

■ Changing to a Standby Mode and Interrupts

If an interrupt request with an interrupt level higher than "11" occurs from a peripheral function to the CPU, writing "1" to the stop bit (STP), sleep bit (SLP), or watch bit (TMD) in the standby control register (STBC) is ignored. Therefore, the CPU does not change to a standby mode. (The CPU also does not change to the standby mode after completing interrupt processing.)

This does not depend on whether or not the CPU accepts the interrupt.

Even if the CPU is currently performing interrupt processing, the interrupt request flag bit is cleared and, if no other interrupt request is present, the device can change to the standby mode.

■ Wake-up from Standby Mode by Interrupt

If an interrupt request with an interrupt level higher than "11" occurs from a peripheral function or others during sleep or stop mode, the CPU wakes up from a standby mode. This does not depend on whether or not the CPU accepts the interrupt.

After wake-up from a standby mode, the CPU performs the normal interrupt operations. If the level set in the interrupt level setting register (ILR1 to ILR3) corresponding to the interrupt request is higher than the interrupt level bits in the condition code register (CCR: IL1, IL0), and if the interrupt enable flag is enabled (CCR: I = "1"), the CPU branches to the interrupt processing routine. If the interrupt is not accepted, operation restarts from the instruction following the instruction that activated a standby mode.

To prevent control from branching to an interrupt processing routine after wake-up, take measures such as disabling interrupts before setting a standby mode.

■ Notes on Setting Standby Mode

When setting the standby control register (STBC) to go to a standby mode, make the settings in accordance with Table 3.7-9 "Standby Control Register (STBC) Low-Power Consumption Mode Settings". The order of precedence as to which mode will be activated if more than one bit is set to "1" is "stop" mode, "watch" mode, and "sleep" mode. Other factors being equal, it is best to set "1" for just one bit.

Also avoid going to stop, sleep, or watch mode immediately after switching from subclock to main clock mode (SYCC: SCS=0 --> 1). First verify that the clock monitor bit (SYCC: SCM) of the system control register is "1", then make the standby mode change.

Note that you cannot go to the watch standby mode when operating in main clock mode. (A write to the TMD bit will be ignored.)

Table 3.7-9 Standby Control Register (STBC) Low-Power Consumption Mode Settings

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

■ Oscillation Stabilization Delay Time

As the oscillator that provides the source oscillation is stopped during stop mode in both main clock mode and subclock mode, a delay time is required for oscillation to stabilize after the oscillator restarts operation.

In main clock mode, the main clock oscillation stabilization delay time is selected from one of four possible delay times defined by the timebase timer. In subclock mode, the subclock oscillation stabilization delay time is defined by the watch prescaler.

In main clock mode, if the interval time set for the timebase timer is less than the oscillation stabilization delay time, the timebase timer generates an interval timer interrupt request before the end of the oscillation stabilization delay time. To prevent this, disable the interrupt request output for the timebase timer (TBTC: TBIE = "0") before changing to stop mode in main clock mode as necessary.

Selection of a watch prescaler interrupt interval shorter than the oscillation stabilization delay time will similarly cause the watch interrupt request to be generated during the oscillation stabilization delay time.To prevent this, disable the watch interrupt request output for the watch prescaler (WPCR: WIE =0) before changing to stop mode in subclock mode as necessary.

3.8 Memory Access Mode

In the MB89980 series, the only memory access mode is the single-chip mode.

■ Single-chip Mode

In single-chip mode, the device uses internal RAM and ROM only. Therefore, the CPU can access no areas other than the internal I/O area, RAM area, and ROM area (internal access).

■ Mode Pins (MOD1, MOD0)

Always set the mode pins, MOD1 and MOD0, for V_{SS}

At reset, reads the mode data and reset vector from internal ROM.

Do not change the mode pin settings, even after completion of the reset (i.e. during normal operation).

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

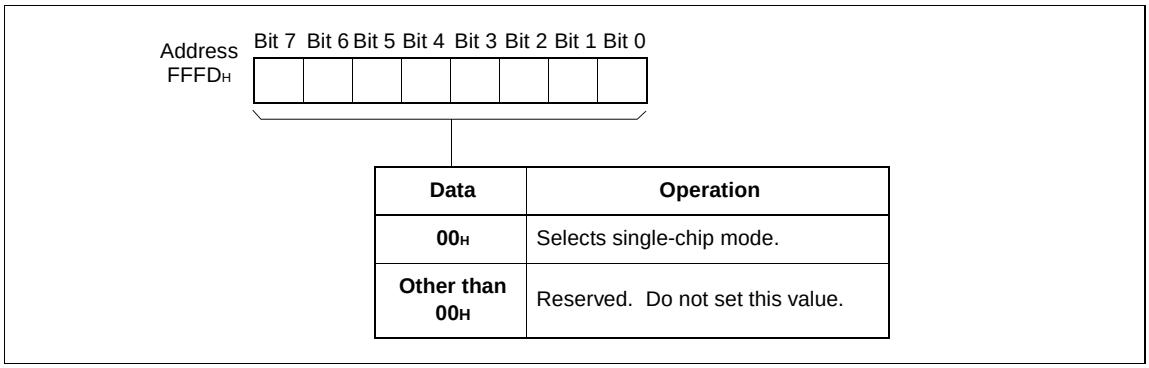
Table 3.8-1 Mode Pin Setting

Pin state		Description
MOD1	MOD0	
V_{SS}	V_{SS}	Reads the mode data and reset vector from internal ROM
V_{SS}	V_{CC}	Prohibited settings
V_{CC}	V_{SS}	
V_{CC}	V_{CC}	

■ Mode Data

Always set the mode data in internal ROM to "00_H" to select single-chip mode.

Figure 3.8-1 Mode Data Structure



■ Memory Access Mode Selection Operation

Only the single-chip mode can be selected.

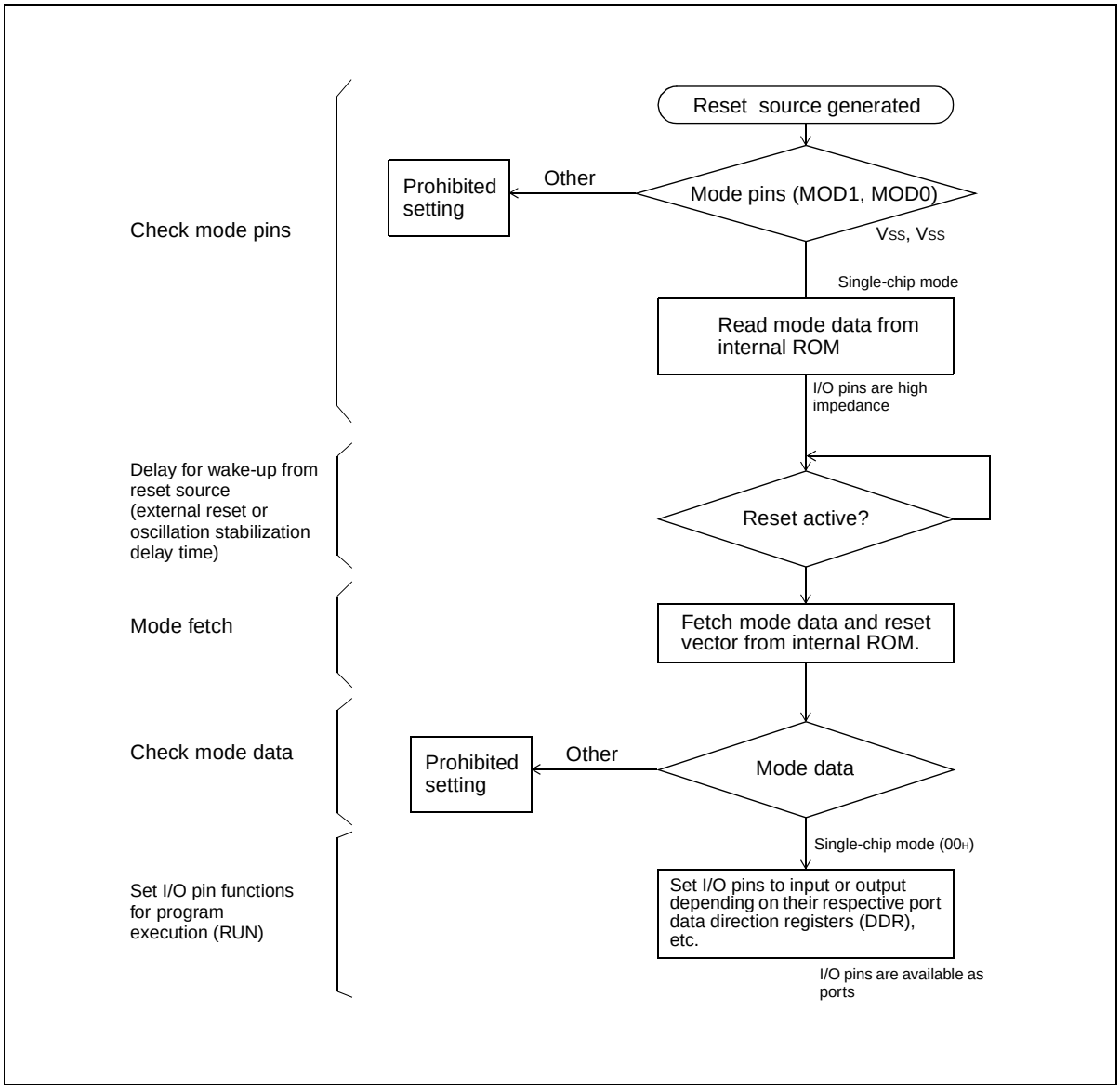
Table 3.8-2 ''Mode Pins and Mode Data' lists the mode pin and mode data options.

Table 3.8-2 Mode Pins and Mode Data

Memory access mode	Mode pins (MOD1, MOD0)	Mode data
Single-chip mode	V_{SS}, V_{SS}	00_H
Other modes	Prohibited settings	Prohibited settings

Figure 3.8-2 "Memory Access Selection Operation" shows the operation for memory access mode selection.

Figure 3.8-2 Memory Access Selection Operation



CHAPTER 4 I/O PORTS

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

4.1 Overview of I/O Ports

The I/O ports consist of eight ports (47 pins) including input-only, output-only and general-purpose I/O ports (parallel I/O ports).
The ports also serve as peripherals (I/O pins of peripheral functions).

■ I/O Port Functions

The functions of the I/O ports are to output data from the CPU via the I/O pins and to fetch signals input to the I/O pins into the CPU. Input and output are performed via the port data registers (PDR). Also, for certain ports the direction of each I/O pin can be individually set to either input or output for each bit by the port data direction register (DDR).

The following lists the functions of each port and the peripheral with which the ports also serve as.

- Port 0: General-purpose I/O port. Also serves as peripherals (external interrupt 2 pins).
- Port 1: General-purpose I/O port. Also serves as peripherals (external interrupt 1 pins).
- Port 2: General-purpose I/O port. Also serves as peripherals (timer, PWM 2, remote control pins).
- Port 3: Port 3: Output-only port (P30) and input-only port (P31 and P32). Also serves as peripherals (PWM 1, buzzer output and sub-clock pins).
- Port 4: Output-only port. Also serves as peripherals (LCDC segment output pins).
- Port 5: Output-only port. Also serves as peripherals (analog input pins).
- Port 6: Port 6: Output-only port. Also serves as peripherals (LCDC segment output pins).
- Port 7: Output-only port. Also serves as peripherals (LCDC common output pins).

Table 4.1-1 "Port Function" lists the functions of each port and Table 4.1-2 "Port Registers" lists the registers for each port.

Table 4.1-1 Port Function

Port	Pin name	Input type	Output type	Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Port0	P00/$\overline{INT20}$ to P07/$\overline{INT27}$	CMOS (resource: hysteresis)	CMOS	General-purpose I/O port	P07	P06	P05	P04	P03	P02	P01	P00
				External interrupts 2	$\overline{INT27}$	$\overline{INT26}$	$\overline{INT25}$	$\overline{INT24}$	$\overline{INT23}$	$\overline{INT22}$	$\overline{INT21}$	$\overline{INT20}$
Port1	P10/INT10 to P13/INT13 P14 to P17			General-purpose I/O port	P17	P16	P15	P14	P13	P12	P11	P10
				External interrupts 1	——		——		INT13	INT12	INT11	INT10
Port2	P20/EC to P27/PWM2		N-ch open-drain*3	General-purpose I/O port	P27	P26	P25	P24	P23	P22	P21	P20
				Peripherals	PWM2	—	—	RCO	—	TO	—	EC
Port3	P30/PWM1/ BZ to P32/ X1A*2	—	CMOS	General-purpose I/O*2	——		——		—	P32	P31	P30
				Peripherals	—	—	—	—	—	X1A	X0A	PWM1/ BZ
Port4	P40/SEG0 to P47/SEG7		N-ch open-drain*1, *3	Output-only port	P47	P46	P45	P44	P43	P42	P41	P40
				LCDC segment output	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
Port5	P50/AN0 to P53/AN3	Analog channel selector	N-ch open-drain	Output-only port	—	—	—	—	P53	P52	P51	P50
				Analog input	—	—	—	—	AN3	AN2	AN1	AN0
Port6	P60/SEG8 to P65/SEG13	—	N-ch open-drain*1	Output-only port	—	—	P65	P64	P63	P62	P61	P60
				LCDC segment output	—	—	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8
Port7	P70/COM2, P71/COM3			Output-only port	——		——		——		P71	P70
				LCDC common output	—	—	—	—	—	—	COM3	COM2

*1: Ports 4, 6, and 7 are output ports only when the ports are selected.

*2 P30 is CMOS output and P31 and P32 are CMOS input only.

*3: PinsP40 to P47, P21, P26, P27, P60 and P61have high current drive-type output circuits.

Table 4.1-2 Port Registers

Register	Read/Write	Address	Initial value
Port 0 data register (PDR0)	R/W	0000_H	$XXXXXXXX_B$
Port 0 data direction register (DDR0)	W*	0001_H	00000000_B
Port 0 pull-up control register (PURR0)	R/W	0040_H	11111111_B
Port 1 data register (PDR1)	R/W	0002_H	$XXXXXXXX_B$
Port 1 data direction register (DDR1)	W*	0003_H	00000000_B
Port 1 pull-up control register (PURR1)	R/W	0041_H	11111111_B
Port 2 data register (PDR2)	R/W	0004_H	$XXXXXXXX_B$
Port 2 data direction register (DDR2)	W*	0005_H	00000000_B
Port 3 data register (PDR3)	R/W	$000C_H$	$XXXXXXX1_B$
Port 4 data register (PDR4)	R/W	$000E_H$	11111111_B
Port 5 data register (PDR5)	R/W	$000F_H$	$XXXX1111_B$
Port 5 pull-up control register (PURR5)	R/W	0042_H	$XXXX1111_B$
Port 6 data register (PDR6)	R/W	0012_H	$XX111111_B$
Port 7 data register (PDR7)	R/W	0013_H	$XXXXXX11_B$

* Bit manipulation instructions cannot be used on DDR0, DDR1, and DDR2.

R/W: Readable and writable

R: Read-only

W: Write-only

X: Indeterminate

4.2 Port 0 and Port 1

Port 0 and port 1 are general-purpose I/O ports that also serve as input pins for external interrupts.
This section principally describes the port functions when operating as general-purpose I/O ports.
The section describes the port structure and pins, the pin block diagram, and the registers for port 0 and port 1.

■ Structure of Port 0 and Port 1

Port 0 and port 1 consist of four components respectively.

❍ Port 0

- General-purpose I/O pins/external interrupt 2 input pins (P00/$\overline{\text{INT20}}$ to P07/$\overline{\text{INT27}}$)
- Port 0 data register (PDR0)
- Port 0 data direction register (DDR0)
- Port 0 pull-up control register (PURR0) (only available in MB89P985 and MB89PV980)

❍ Port 1

- General-purpose I/O pins/external interrupt 1 input pins (P10/INT10 to P13/INT13) and general-purpose I/O pins (P14 to P17)
- Port 1 data register (PDR1)
- Port 1 data direction register (DDR1)
- Port 1 pull-up control register (PURR1) (only available in MB89P985 and MB89PV980)

■ Port-0 and Port-1 Pins

Port 0 and port 1 both consist of eight I/O pins of a CMOS input and CMOS output type respectively.

When P00/$\overline{\text{INT20}}$ to P07/$\overline{\text{INT27}}$ (Port 0) and P10/INT10 to P13/INT13 (Port 1) are used as input pins, they can also be used as external interrupt input pins.

Table 4.2-1 Port-0 and Port-1 Pins

Port	Pin name	Function	Shared peripheral	I/O type		Circuit type	
				Input	Output	MB89983	MB89P985 MB89PV980
Port 0	P00/$\overline{\text{INT20}}$ to P07/$\overline{\text{INT27}}$	General-purpose I/O	External interrupt 2	CMOS*	CMOS	E	F
Port 1	P10/INT10 to P13/INT13		External interrupt 1				
	P14 to P17		-	CMOS		G	H

* External interrupt inputs are hysteresis inputs.

Reference:

See Section 1.7 "I/O Pins and Pin Functions" for a description of the circuit type.

■ Block Diagram of Port-0 and Port-1 Pin

Figure 4.2-1 Block Diagram of Port-0 and Port-1 Pin for MB89983

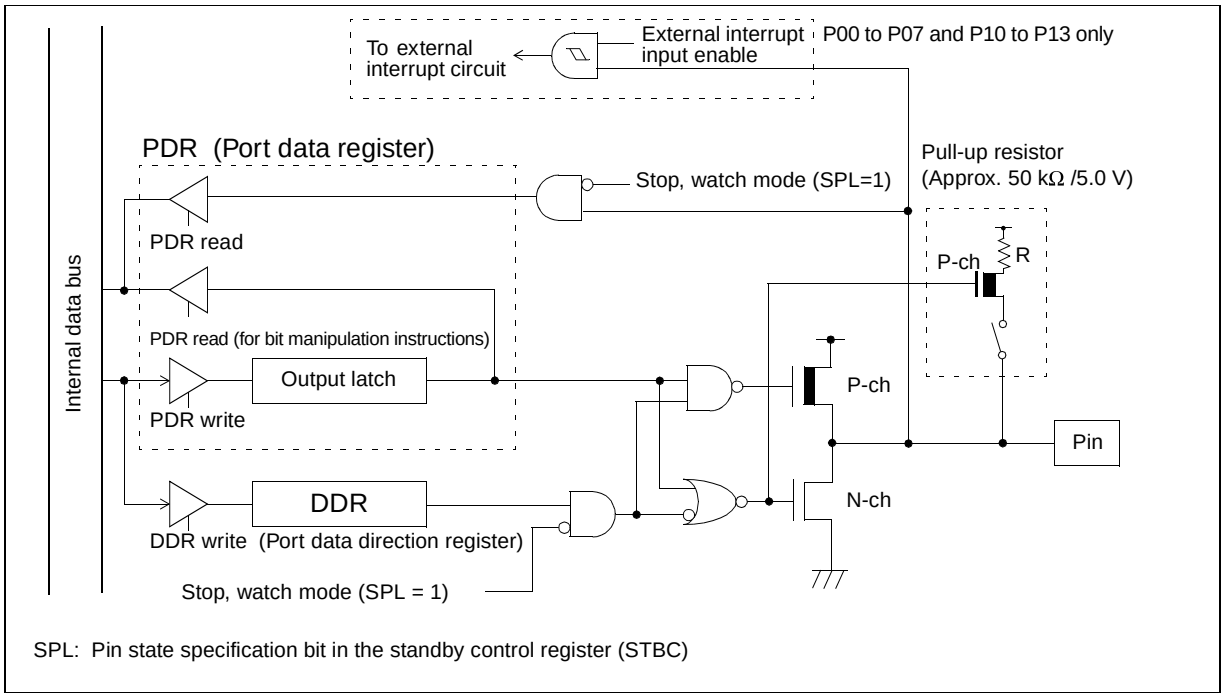
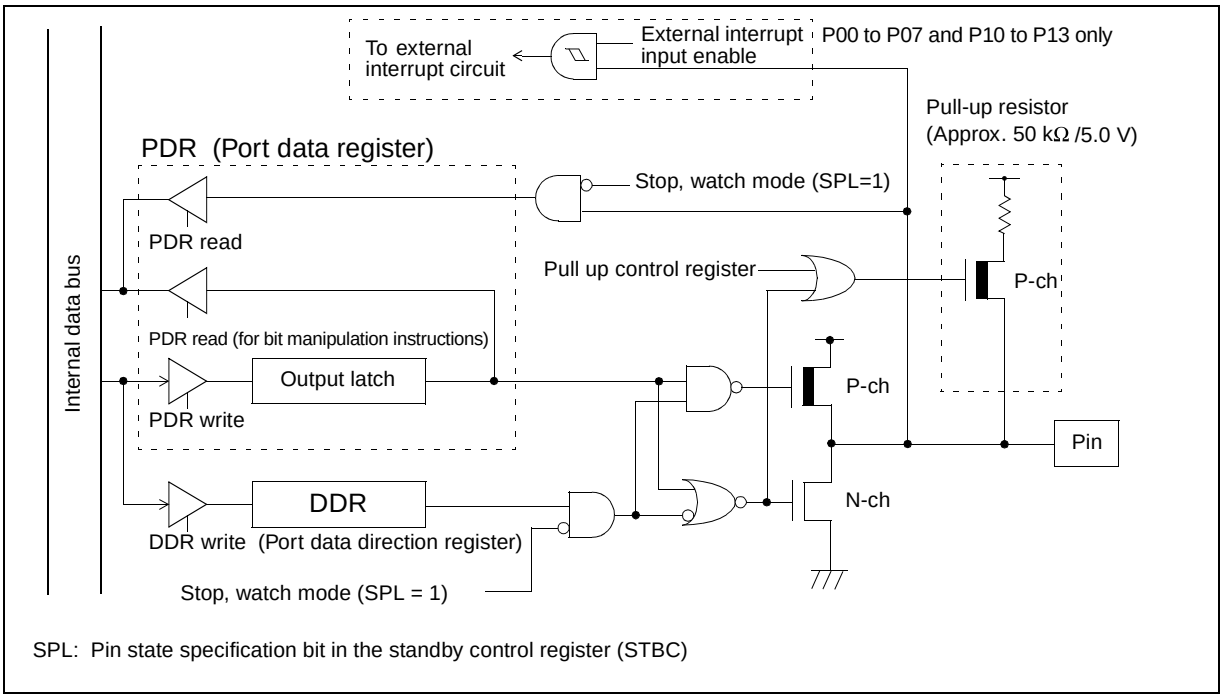


Figure 4.2-2 Block Diagram of Port-0 and Port-1 Pin for MB89P985 and MB89PV980



Check:

When a port is used as a normal input port, external interrupt circuit operation that use the same pins must be disabled. See CHAPTER 9 "EXTERNAL INTERRUPT CIRCUIT 1" and CHAPTER 10 "EXTERNAL INTERRUPT CIRCUIT 2" for details.

Check:

MB89983 can be set with a pull-up resistor by mask option

Check:

MB89P985 and MB89PV980 can be set with a pull-up resistor by software (pull-up control register)

■ Port-0 and Port-1 Registers

The port-0 registers consist of PDR0, DDR0 and PURR0. The port-1 registers consist of PDR1, DDR1 and PURR1. Each bit in these registers has a one-to-one relationship with port-0 and port-1 pin respectively. Table 4.2-2 "Correspondence between Pin and Register for Port-0 and Port-1" shows the correspondence between pins and registers for port-0 and port-1.

Table 4.2-2 Correspondence between Pin and Register for Port-0 and Port-1

Port	Correspondence between register bit and pin								
Port 0	PDR0,DDR0,PURR0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Corresponding pin	P07	P06	P05	P04	P03	P02	P01	P00
Port 1	PDR1,DDR1,PURR1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Corresponding pin	P17	P16	P15	P14	P13	P12	P11	P10

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

This section describes the port-0 and port-1 registers.

■ Port-0 and Port-1 Register Functions

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

The PDR0 and PDR1 registers hold the pin states. Therefore, a bit corresponding to a pin set as an output port can be read as the same state ("0" or "1" as the output latch, but when it is an input port, it cannot be read as the output latch state.

Note:

For SETB and CLRB bit operation instructions, since the state of output latch (not the pin) is read, the output latch states of bits other than those being operated on are not changed.

❍ **Port 0, 1 data direction registers (DDR0, DDR1)**

The DDR0 and DDR1 registers set the direction (input or output) for each pin (bit).

Setting "1" to the bit corresponding to a port (pin) sets the pin as an output port. Setting "0" sets the pin as an input port.

Check:

As the DDR0 and DDR1 registers are write-only, the bit manipulation instructions (SETB and CLRB) cannot be used.

❍ **Settings when pins are used as external interrupt inputs**

When port pins are used as external interrupt input pins, in addition to enabling the interrupt circuit (external interrupt 1 or 2), the corresponding pins must also be set as inputs. (The corresponding output latch data has no significance in this case.)

Table 4.2-3 "Port-0 and Port-1 Register Function" lists the functions of the port-0 and port-1 registers.

Table 4.2-3 Port-0 and Port-1 Register Function

Register	Data	Read	Write	Read/ Write	Address	Initial value
Port 0 data register (PDR0)	0	Pin state is the "L" level.	Sets "0" to the output latch. Outputs an "L" level to the pin if the pin functions as an output port.	R/W	0000_H	$XXXXXXXX_B$
	1	Pin state is the "H" level.	Sets "1" to the output latch. Outputs an "H" level to the pin if the pin functions as an output port.			

Table 4.2-3 Port-0 and Port-1 Register Function

Register	Data	Read	Write	Read/ Write	Address	Initial value
Port 0 data direction register (DDR0)	0	Reading is not permitted (write-only).	Disables output transistor and sets the pin as an input pin.	W	0001_H	00000000_B
	1		Enables output transistor and sets the pin as output pin.			
Port 1 data register (PDR1)	0	Pin state is the "L" level.	Sets "0" to the output latch. Outputs an "L" level to the pin if the pin functions as an output port.	R/W	0002_H	$XXXXXXXX_B$
	1	Pin state is the "H" level.	Sets "1" to the output latch. Outputs an "H" level to the pin if the pin functions as an output port.			
Port 1 data direction register (DDR1)	0	Reading is not permitted. (write-only.)	Disables output transistor and sets the pin as an input pin.	W	0003_H	00000000_B
	1		Enables output transistor and sets the pin as output pin.			

R/W: Readable and writable

W: Write-only

X: Indeterminate

❍ **Port 0, 1 pull-up control registers (PURR0, PURR1)**

By using pull-up resistor option in each pin for port 0, 1, setting is possible bit by bit when writing to pull-up control register for MB89P985 and MB89PV980 only. The pull-up resistor for Mb89983 is selected by mask option.

When pull-up resistor is selected in pull-up register in stop and clock mode (SPL=1). The state of these pin are in "H" level (pull up state) rather than high impedance. However, during reset, pull up is unavailable and will be in high impedance state.

Figure 4.2-3 "Pull up control registers setting (PURR0)" and Figure 4.2-4 "Pull up control registers setting (PURR0, PURR1)" are list of the pull-up resistor option setting of PURR0 and PURR1

Figure 4.2-3 Pull up control registers setting (PURR0)

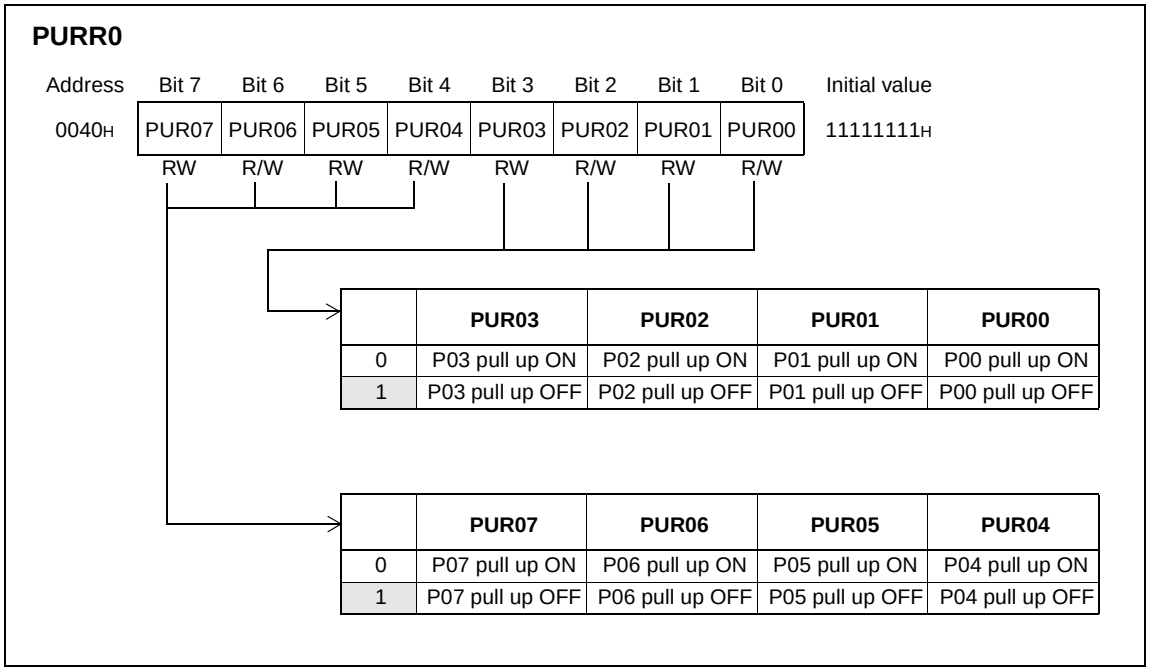
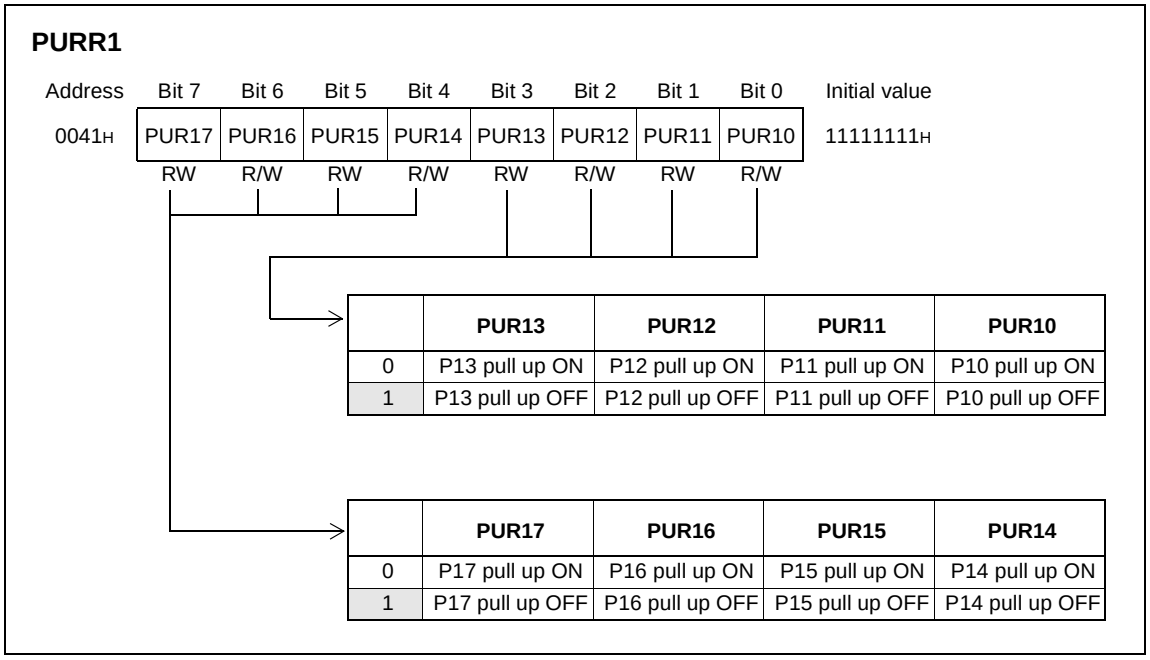


Figure 4.2-4 Pull up control registers setting (PURR0, PURR1)



Note:

For MB89P985 and MB89PV980, there will be current leakage through the pull-up resistor in stop mode when the pull-up resistor is enabled and these ports are input "0". To prevent the current leakage, the pull-up resistor should be disabled before going into stop mode.

4.2.2 Operation of Port 0 and Port 1

This section describes the operations of the port 0 and port 1.

■ Operation of Port 0 and Port 1

❍ **Operation as an output port**

- Setting the corresponding DDR0 or DDR1 register bit to "1" sets a pin as an output port.
- When a pin is set as an output port, its output transistor is enabled and the pin outputs the data stored in the output latch.
- Writing data to the PDR0 and PDR1 registers stores the data in the output latch and outputs the data directly to the pin.
- Reading the PDR0 or PDR1 register returns the pin value.

❍ **Operation as an input port**

- Setting the corresponding DDR0 or DDR1 register bit to "0" sets a pin as an input port.
- When a pin is set as an input port, the output transistor is "OFF" and the pin goes to the high-impedance state.
- Writing data to the PDR0 and PDR1 registers stores the data in the output latch but does not output the data to the pin.
- Reading the PDR0 or PDR1 register returns the pin value.

❍ **Operation as an external interrupt input**

- When a port is an external interrupt input, the port is made an input by setting the corresponding DDR0 or DDR1 register bits to "0".
- Reading the PDRO or PDR1 register returns the pin value, regardless of whether external interrupt inputs or interrupt request outputs are enabled/disabled.

❍ **Operation at reset**

- Resetting the CPU initializes the DDR0 and DDR1 register values to "0". This sets the output transistors "OFF" (all pins become input ports) and sets the pins to the high-impedance state.
- The PDR0 and PDR1 registers are not initialized by a reset. Therefore, to use as output ports, the output data must be set in the PDR0 and PDR1 registers before setting the corresponding DDR0 or DDR1 register bits to output mode.

❍ **Operation in stop and watch modes**

The pins go to the high-impedance state if the pin state specification bit in the standby control register (STBC: SPL) is "1" when the device changes to stop or watch mode. This is achieved by forcibly setting the output transistor "OFF" regardless of the DDR0 and DDR1 register values.

To avoid current leakage, it is recommended to remain a known logic level of the input port pins during the standby mode.

Table 4.2-4 "Port-0 and Port-1 Pin State" lists the port-0 and port-1 pin states.

Table 4.2-4 Port-0 and Port-1 Pin State

Pin name	Normal operation main-sleep mode main-stop mode (SPL=0) sub-sleep mode sub-stop mode (SPL=0) watch mode (SPL=0)	Main-stop mode (SPL = 1) sub-stop mode (SPL = 1) watch mode (SPL = 1)	Reset
P00/$\overline{\text{INT20}}$ to P07/ $\overline{\text{INT27}}$	General-purpose I/O ports/ external interrupt input	Hi-z (external interrupt input)	Hi-z
P10/INT10 to P13/ INT13 P14 to P17			

SPL: Pin state specification bit in the standby control register (STBC)

Hi-z: High impedance

Note:

Pins with a pull-up resistor go to the "H" level (pull-up state) rather than to the high-impedance state when the output transistors are all "OFF".

4.3 Port 2

Port 2 is a general-purpose I/O port that also serves as the resource signal I/O pins. Individual pins can be switched in units of a bit between the port and resource functions. This section principally describes the port functions when operating as a general-purpose I/O port.
The section describes the port structure and pins, the pin block diagram, and the port registers for port 2.

■ Structure of Port 2

Port 2 consists of the following three components:

- General-purpose I/O port/resource I/O pins (P20/EC to P27/PWM2)
- Port 2 data register (PDR2)
- Port 2 data direction register (DDR2)

■ Port-2 Pins

Port 2 consists of eight I/O pins of a CMOS input, an N-ch open-drain output types. Three of these pins are also used as signal I/O pins for various resources. While they are being used by the resource, these pins cannot be used as the general-purpose I/O port. Some pin outputs are high current drive-type outputs.

Table 4.3-1 "Port-2 Pin" lists the port-2 pins.

Table 4.3-1 Port-2 Pin

Port	Pin name	Function	Shared peripheral	I/O type		Circuit type	
				Input	Output	MB89983	MB89P985 MB89PV980
Port 2	P20/EC	P20 General-purpose I/O	EC 8/16-bit timer, pulse input	CMOS*1	N-ch open-drain*2	J	K
	P21	P21 General-purpose I/O	-	CMOS		L	M
	P22/TO	P22 General-purpose I/O	TO 8/16-bit timer, timer output				
	P23	P23 General-purpose I/O	-	CMOS			
	P24/ RCO	P24 General-purpose I/O	RCO Remote control output	CMOS			
	P25	P25 General-purpose I/O	-	CMOS			
	P26	P26 General-purpose I/O	-	CMOS			
	P27/ PWM2	P27 General-purpose I/O	PWM 2 8-bit PWM timer 2, timer output				

*1: Peripheral inputs are hysteresis inputs.

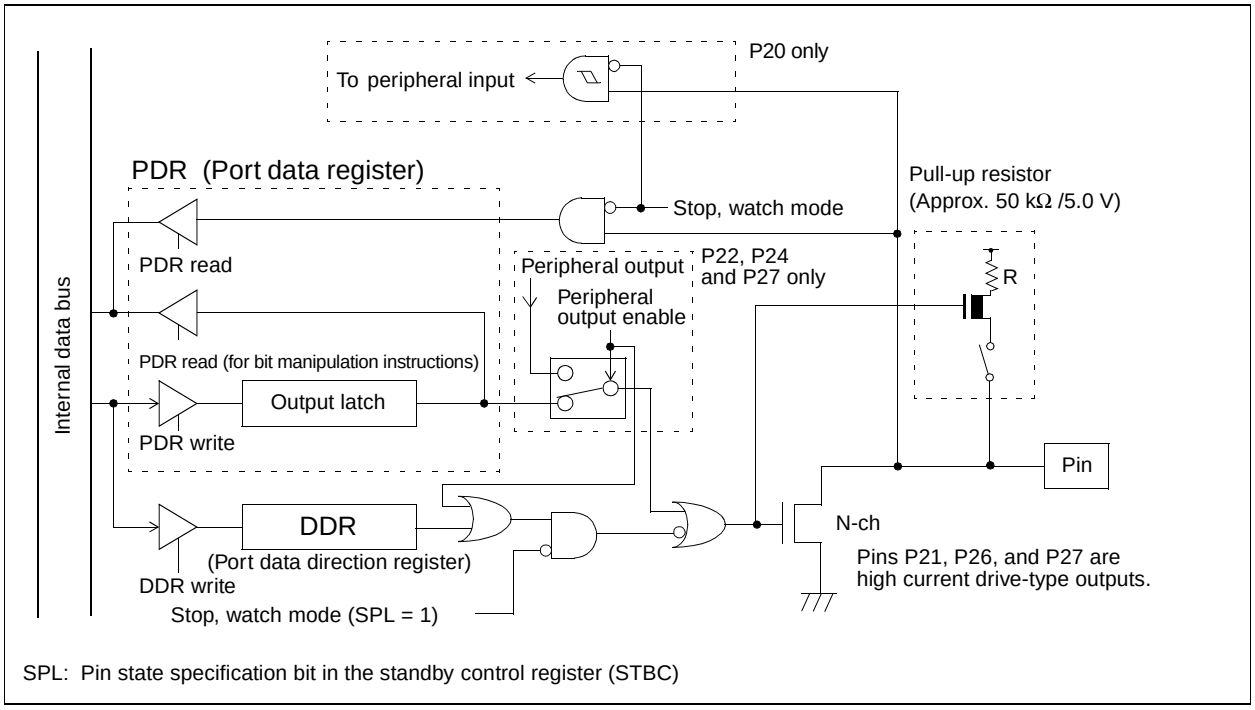
*2: P21, P26, and P27 have high current drive-type outputs.

Reference:

See Section 1.7 "I/O Pins and Pin Functions" for a description of the circuit type.

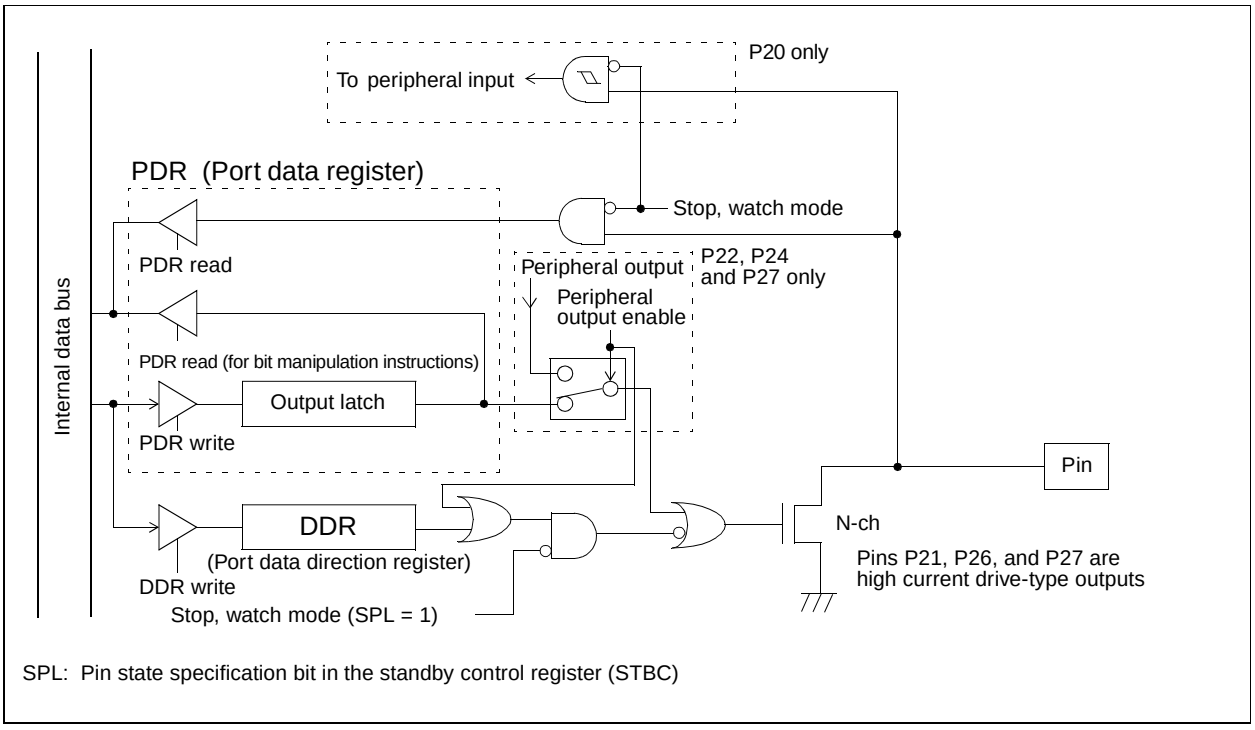
■ Block Diagram of Port-2 Pin

Figure 4.3-1 Block Diagram of Port-2 Pin for MB89983



SPL: Pin state specification bit in the standby control register (STBC)

Figure 4.3-2 Block Diagram of Port-2 Pin for MB89P985 and MB89PV980



SPL: Pin state specification bit in the standby control register (STBC)

Note:

Peripheral inputs continuously are input the pin value (except during stop and watch modes).

Check:

MB89983 can be set with a pull-up resistor by mask option

■ Port-2 Registers

The port-2 registers consist of PDR2 and DDR2.

Each bit in these registers has a one-to-one relationship with a port 2 bit and port 2 pin.

Table 4.3-2 "Port-2 Register Function" shows the correspondence between pins and registers for port 2.

Table 4.3-2 Correspondence between Pin and Register for Port 2

Port	Correspondence between register bit and pin								
Port 2	PDR2, DDR2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Corresponding pin	P27	P26	P25	P24	P23	P22	P21	P20

4.3.1 Port-2 Registers (PDR2, DDR2)

This section describes the port-2 registers.

■ Port-2 Register Functions

❍ Port 2 data register (PDR2)

The PDR2 register holds the pin states. Therefore, when used as an output port that is not a peripheral output, it reads out as the same state ("0" or "1") as that of the output data latch; and when it is an input port, the output latch state cannot be read out.

Note:

As the bit manipulation instructions (SETB and CLRB) read the output latch data rather than the pin level, the instructions do not change the output latch values for bits other than the bit being set or cleared.

❍ Port 2 data direction register (DDR2)

The DDR2 register sets the direction (input or output) for each pin (bit).

Setting "1" to the bit corresponding to a port (pin) sets the pin as an output port. Setting "0" sets the pin as an input port.

Check:

As the DDR2 register is write-only, the bit manipulation instructions (SETB and CLRB) cannot be used.

❍ Settings as a peripheral output

To use a peripheral that has an output pin, set the peripheral output enable bit for that pin to the "enable" state. As can be seen in the block diagram, the peripheral has precedence over the general-purpose port for use of the output pin. Once the peripheral output is enabled, the states set in the PDR2 and DDR2 registers are no longer valid, and do not affect the data output by the peripheral, or the enabling of the output.

❍ Settings as a peripheral input

To use a peripheral that has a Port 2 pin as an input pin, set that pin as an input port. The output latch data for that pin will no longer be valid.

Table 4.3-3 "Port-2 Register Function" lists the functions of the port-2 registers.

Table 4.3-3 Port-2 Register Function

Register	Data	Read	Write	Read/ Write	Address	Initial value
Port 2 data register (PDR2)	0	Pin state is the "L" level.	Outputs an "L" level to the pin if the pin functions as an output port. (Sets "0" to the output latch and turns the output transistor "ON")	R/W	0004_H	$XXXXXXXX_B$
	1	Pin state is the "H" level.	Sets the pin to the high-impedance state if the pin functions as an output port*. (Sets "1" to the output latch and turns the output transistor "OFF")			
Port 2 data direction register (DDR2)	0	Reading is not permitted (write-only).	Disables the output transistor and sets the pin as an input pin.	W	0005_H	00000000_B
	1		Enables the output transistor and sets the pin as an output pin.			

R/W: Readable and writable

W: Write-only

X: Indeterminate

*: Pins with a pull-up resistor, go to the pull-up state.

4.3.2 Operation of Port 2

This section describes the operations of the port 2.

■ Operation of Port 2

❍ Operation as an output port

- Setting the corresponding DDR2 register bit to "1" sets a pin as an output port.
- When a pin is as an output port, the output transistor is enabled. When the output latch value is "0", the output transistor turns "ON" and an "L" level is output from the pin. When the output latch value is "1" the transistor turns "OFF" and the pin goes to the high-impedance state. If a pull-up is set to the output pin, the pin goes to the pull-up state when the output latch value is "1".
- Writing data to the PDR2 register stores the data in the output latch and outputs the data to the pin.
- Reading the PDR2 register returns the pin value.

❍ Operation as an input port

- Setting the corresponding DDR2 register bit to "0" sets a pin as an input port.
- When a pin is set as an input port, the output transistor is "OFF" and the pin goes to the high-impedance state.
- Writing data to the PDP2 register stores the data in the output latch but does not output the data to the pin.
- Reading the PDR2 register returns the pin value.

❍ Operation as a peripheral output

- If a peripheral output enable bit is set to "enable", the corresponding pin becomes a peripheral output.
- As the pin value can be read even if the peripheral output is enabled, the peripheral output value can be read via the PDR2 register.

❍ Operation as a peripheral input

- A port pin is set as a peripheral input by setting the corresponding DDR2 register bit to "0".
- Reading the PDR2 register returns the pin value, regardless of whether or not the peripheral is using the input pin.

❍ Operation at reset

- Resetting the CPU initializes the DDR2 register value to "0". This sets output transistors "OFF" (pins become input ports) and sets the pins to the high-impedance state.
- The PDR2 register is not initialized by a reset. Therefore, to use as output ports, the output data must be set in the PDR2 register before setting the corresponding DDR2 register bit to output mode.
- Pin state of P27 is undetermined until the internal clock starts operation.

❍ **Operation in stop and watch modes**

The pins go to the high-impedance state, if the pin state specification bit in the standby control register (STBC: SPL) is "1" when the device changes to stop or watch mode. This is achieved by forcibly setting the output transistor "OFF" regardless of the DDR2 register value.

Table 4.3-4 "Port-0 and Port-1 Pin State" lists the port-2 pin states.

Table 4.3-4 Port-0 and Port-1 Pin State

Pin name	Normal operation main-sleep mode main-stop mode (SPL=0) sub-sleep mode sub-stop mode (SPL=0) watch mode (SPL=0)	Main-stop mode (SPL = 1) sub-stop mode (SPL = 1) watch mode (SPL = 1)	Reset
P20/EC to P27/ PWM2	General-purpose I/O port/ peripheral I/O	Hi-z	Hi-z

SPL: Pin state specification bit in the standby control register (STBC)

Hi-z: High impedance

Note:

Pins with a pull-up resistor go to the "H" level (pull-up state) rather than to the high-impedance state when the output transistor is turned "OFF".

*: Pin state of P27 is undetermined until the internal clock starts operation.

4.4 Port 3

P30 is an output-only port and P31 and P32 are input-only ports that also serves as peripheral output. Each pin can be switched between peripheral and port operation in bit units.
The section describes the port structure and pins, the pin block diagram, and the port registers for port 3.

■ Structure of Port 3

Port 3 consist of the following two components:

- CMOS I/O pins/peripheral output pins (P30/PWM1/BZ to P32/X1A)
- Port 3 data register (PDR3)

■ Port-3 Pins

Port 3 has three output-only pins: two Hysteresis inputs and one CMOS output.

Table 4.4-1 "Port-3 Pin" lists the port-3 pins.

Table 4.4-1 Port-3 Pin

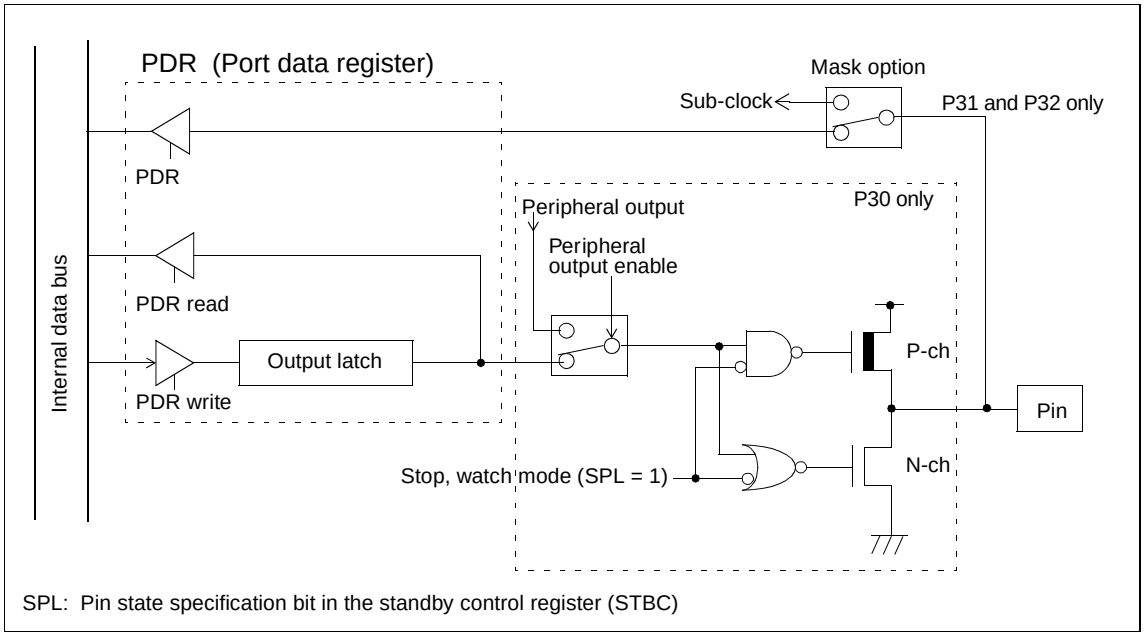
Port	Pin name	Function	Shared peripheral	I/O type		Circuit type
				Input	Output	
Port 3	P30/PWM1/BZ	P30 CMOS Output	PWM 1 output for the 8-bit PWM timer/ counter or Buzzer output	-	CMOS	I
	P31/X0A	P31 Hysteresis input	Sub-clock X0A	Hystersis	-	R/B
	P32/X1A	P32 Hysteresis input	Sub-clock X1A	Hystersis	-	R/B

Reference:

See Section 1.7 "I/O Pins and Pin Functions" for a description of the circuit type.

■ Block Diagram of Port-3 Pin

Figure 4.4-1 Block Diagram of Port-3 Pin



SPL: Pin state specification bit in the standby control register (STBC)

■ Port-3 Registers

The port-3 registers consist of PDR3.

Each bit in PDR3 registers has a one-to-one relationship with a port-3 pin. Table 4.4-2 "Correspondence between Pin and Register for Port 3" shows the correspondence between pins and registers for port 3.

Table 4.4-2 Correspondence between Pin and Register for Port 3

Port	Correspondence between register bit and pin								
Port 3	PDR3, DDR3	-	-	-	-	-	Bit 2	Bit 1	Bit 0
	Corresponding pin	-	-	-	-	-	P32	P31	P30

4.4.1 Port-3 Register (PDR3)

This section describes the Port-3 register.

■ Port-3 Register Functions

❍ **Port 3 data register (PDR3)**

The PDR3 register holds the output latch states. Therefore, it does not read out as the pin states or peripheral output data.

❍ **Settings as a peripheral output**

When using peripherals that have output pins, set the peripheral's output enable bit for that pin to the "enable" state.

Since the peripheral has precedence over the port for use of the output pin, once the peripheral output is enabled, the data in the PDR3 register has no significance, and has no affect on the data output by the peripheral, or the enabling of the output.

Table 4.4-3 "Port-3 Register Function" lists the functions of the port-3 registers.

Table 4.4-3 Port-3 Register Function

Register	Data	Read	Write	Read/ Write	Address	Initial value
Port 3 data register (PDR3)	0	Output latch value is "0".	Sets "0" to the output latch. Outputs an "L" level to the pin.	R/W	$000C_H$	$XXXXXXX1_B$
	1	Output latch value is "1".	Sets "1" to the output latch. Outputs an "H" level to the pin.			

R/W: Readable and writable

X: Indeterminate

4.4.2 Operation of Port 3

This section describes the operations of the port 3.

■ Operation of Port 3

❍ Operation as an output port (P30/PWM1/BZ only)

- Writing data to the PDR3 register stores the data in the output latch. The pin outputs the data stored in the output latch at P30.
- Reading the PDR3 register always returns the output latch value.

❍ Operation as a peripheral output (P30/PWM1/BZ only)

- Setting the output enable bit of the peripheral to "enable" makes the corresponding pin a peripheral output.
- You cannot read the peripheral output value by reading PDR3. (PDR3 contains the output latch value.)

❍ Operation as an input port (P31/X0A and P32/X1A only)

- Writing data to the PDR3 registers stores the data in the output latch but does not output the data to the pin.
- Reading the PDR3 register returns the pin value.
- For single clock product, if P31 and P32 are not used as input pins, P31 and P32 should be connected to pull-up or pull-down resistor.

❍ Operation at reset

- Resetting the CPU initializes the PDR3 register values to "1". This outputs "H" level at P30. P31 and P32 are input pins.
- Pin state of P30 is undetermined until the internal clock starts operation.

❍ Operation in stop and watch modes

- The output transistors are forcibly turned "OFF" and the pins go to the high-impedance state if the pin state specification bit in the standby control register (STBC:SPL) is "1" when the device changes to stop or watch mode.
- To avoid current leakage, it is recommended to remain a known logic level of the input port pins during the standby mode.
- Table 4.4-4 "Port-3 Pin State" lists the port-3 pin states.

Table 4.4-4 Port-3 Pin State

Pin name	Normal operation main-sleep mode main-stop mode (SPL=0) sub-sleep mode sub-stop mode (SPL=0) watch mode (SPL=0)	Main-stop mode (SPL = 1) sub-stop mode (SPL = 1) watch mode (SPL = 1)	Reset
P30/PWM1/BZ	CMOS output /peripheral output	Hi-z	"H"*
P31/X0A P32/X1A	General-purpose Input/peripheral I/O	input	input

SPL: Pin state specification bit in the standby control register (STBC)

Hi-z: High impedance

*: Pin state of P30 is undetermined until the internal clock starts operation.

4.5 Ports 4, 6 and 7

Ports 4, 6, and 7 are output-only ports that also serve as the LCDC common and segment outputs. The output ports and LCDC segment (and common) outputs are selected as mask options. This section principally describes the port functions when operating as an output-only port. The section describes the port structure and pins, the pin block diagram, and the port register for port 4, 6, and 7.

■ Structure of Port 4, 6, and 7

Ports 4 and 6 are each made up of three elements. Ports 7 is made up of two elements.

Port 4:

- Output-only pins/LCDC segment output pins (P40/SEG0 to P47/SEG7)
- Port 4 data register (PDR4)

Port 6:

- Output-only pins/LCDC segment output pins (P60/SEG8 to P65/SEG13)
- Port 6 data register (PDR6)

Port 7:

- Output-only pins/LCDC common output pins (P70/COM2 to P71/COM3)
- Port 7 data register (PDR7)

■ Port-4, 6, and 7 Pins

Ports 4 consists of eight output-only pins of an N-ch open-drain output type, Port 6 consists of six and Port 7 consists of two.

The pin functions are selected by mask options (MB89983) or by control register (MB89P985 and MB89PV980). When the LCDC common and segment outputs are selected. Do not use these pins as output-only ports. Port 4 option can be selected in eight-bit group and Port 6 option can be selected in two-bit groups. Table 4.5-1 "Port-4, 6, and 7 Pin" lists the port-4, 6,

and 7 pins.

Table 4.5-1 Port-4, 6, and 7 Pin

Port	Pin name	Function	Shared peripheral	I/O type		Circuit type	
				Input	Output	MB89983	MB89P985 MB89PV980
Port 4	P40/SEG0 to P47/SEG7*1	P40 to P47 Output-only	SEG0 to SEG7 LCDC segment output	-	N-ch open-drain	N/O*	S
Port 6	P60/SEG8*1, P61/SEG9*1, P62/SEG10 to P65/SEG13	P60 to P65 Output-only	SEG8 to SEG13 LCDC segment output				
Port 7	P70/COM2, P71/COM3	P70, P71 Output-only	COM2, COM3 LCDC common output			T/O*	

*: The circuit type is "O" when LCDC segment output pin is selected.

*1: Hign current type

Reference:

See Section 1.7 "I/O Pins and Pin Functions" for a description of the circuit type.

See CHAPTER 14 "LCD CONTROLLER-DRIVER" for details of pin operation when used as an LCDC common and segment outputs.

■ Block Diagram of Port-4, 6 and 7 Pin

Figure 4.5-1 Block Diagram of Port-4, 6 and 7 for MB89983

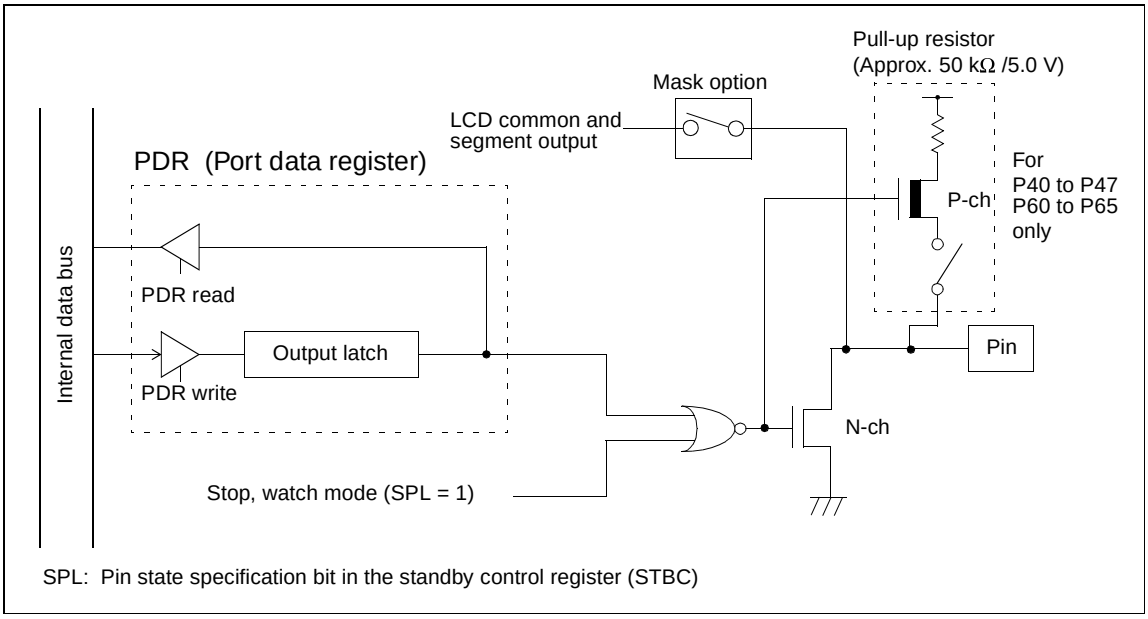
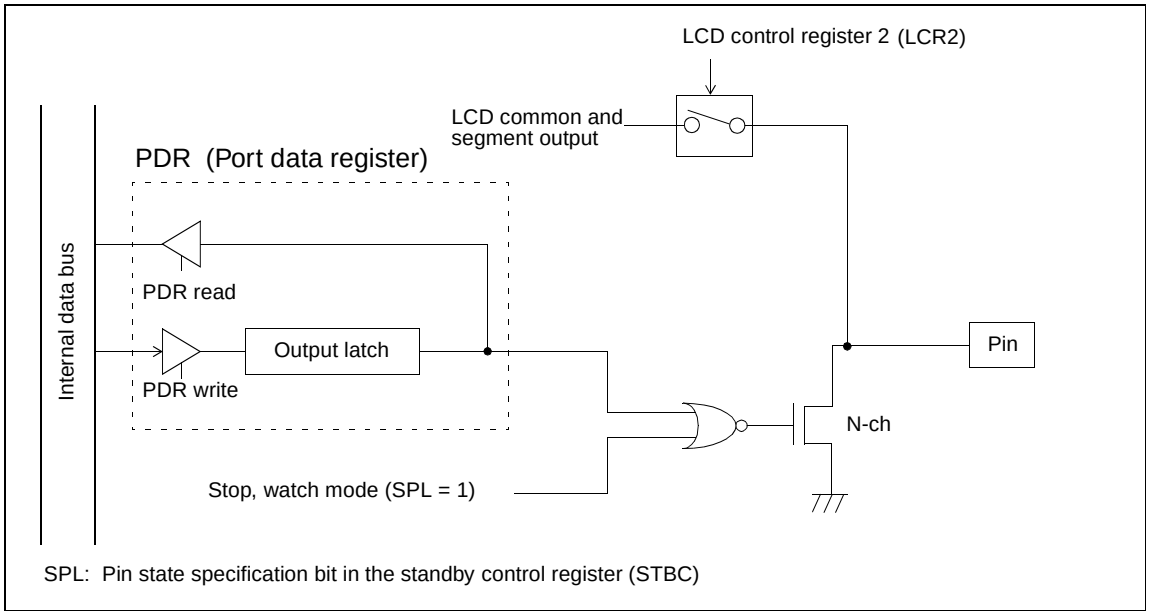


Figure 4.5-2 Block Diagram of Port-4, 6 and 7 for MB89P985 and MB89PV980



Check:

If you use Port 4, 6 pins as the LCD common and segment output pins, do not set a pull-up resistor for those pins, and do not use the pins as output ports.

■ Port-4, 6, and 7 Registers

Port 4, 6, and 7 registers consist of PDR4, PDR6, and PDR7. Each bit in these registers has a one-to-one relationship with a port 4,6, and 7 bit and pin respectively.

Table 4.5-2 "Correspondence between Pin and Register for Port 4, 6, and 7" shows the correspondence between the pins and register for port-4, 6, and 7.

Table 4.5-2 Correspondence between Pin and Register for Port 4, 6, and 7

Port	Correspondence between register bit and pin								
Port 4	PDR4	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Corresponding pin	P47	P46	P45	P44	P43	P42	P41	P40
Port 6	PDR6	-	-	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Corresponding pin	-	-	P65	P64	P63	P62	P61	P60
Port 7	PDR7	-	-	-	-	-	-	Bit 1	Bit 0
	Corresponding pin	-	-	-	-	-	-	P71	P70

4.5.1 Port-4, Port-6, and Port-7 Registers (PDR4, PDR6, and PDR7)

This section describes the port-4, port-6, and port-7 registers.

■ Port-4, port-6, and port-7 Register Functions

❍ Port 4, port 6, and port 7 data registers (PDR4, PDR6, and PDR7)

The PDR4, 6, and 7 registers hold the states of their respective output data latches. Therefore, you cannot determine the pin states or LCDC common and segment output states by reading the registers.

❍ Settings as an LCDC common and segment output

To use pins as LCDC common and segment outputs, that function must be selected by a mask option (MB89983) or by control register (MB89P985 and MB89PV980). The register bits for pins used for this purpose should be set to turn the output transistor "OFF" to prevent it from interfering with the LCDC common and segment output.

Table 4.5-3 "Port-4, Port-6, and Port-7 Register Function" lists the functions of Port-4, port-6, and port-7 register.

Table 4.5-3 Port-4, Port-6, and Port-7 Register Function

Register	Data	Read	Write	Read/ Write	Address	Initial value
Port 4 data register (PDR4)	0	Output latch value is "0".	Sets "0" to the output latch. Outputs an "L" level to the pin.	R/W	$000E_H$	11111111_B
	1	Output latch value is "1".	Sets "1" to the output latch. Outputs an "H" level to the pin.			
Port 4 data register (PDR4)	0	Output latch value is "0".	Sets "0" to the output latch. Outputs an "L" level to the pin.	R/W	0012_H	$XX111111_B$
	1	Output latch value is "1".	Sets "1" to the output latch. Outputs an "H" level to the pin.			
Port 4 data register (PDR4)	0	Output latch value is "0".	Sets "0" to the output latch. Outputs an "L" level to the pin.	R/W	0013_H	$XXXXXX11_B$
	1	Output latch value is "1".	Sets "1" to the output latch. Outputs an "H" level to the pin.			

R/W: Readable and writable

X: Indeterminate

*1: Sets "0" to the output latch and turn the output transistor "ON".

*2: Sets "1" to the output latch and turn the output transistor "OFF". Pins with a pull-up resistor go to the pull-up state.

4.5.2 Operation of Port 4, Port 6 and Port 7

This section describes the operations of the port 4, port 6 and port 7.

■ Operation of Port 4, 6, and 7

❍ Operation as an output port

- When used as an output-only port (mask option), the pins cannot be used for LCDC common and segment outputs.
- Writing data to the PDR4, 6, and 7 register stores the data in the output latches. When the output latch value is "0" the output transistor turns "ON" and an "L" level is output from the pin. When the output latch value is "1", the transistor turns "OFF" and the pin goes to the high-impedance state. For port 4 and 6 (mask option), if a pull-up is set to the output pin, the pin goes to the pull-up state when the output latch value is "1".
- Reading the PDR4, 6, and 7 register always returns the output latch data.

❍ Operation as an LCDC common and segment output

- When the LCDC output option is selected, set the PDR4, 6, and 7-register bits corresponding to the LCDC common and segment output pins to "1" to turn the output transistor "OFF".
- You cannot read the LCDC output data by reading PDR4, 6 or 7. (If you read the PDR registers you will get the output latch data, not the LCDC output data.)

❍ Operation at reset

Resetting the CPU initializes the PDR4, 6, and 7 register values to "1". This turns "OFF" the output transistor for all pins and sets the pins to the high-impedance state.

❍ Operation in stop and watch modes

The output transistors are forcibly turned "OFF" and the pins go to the high-impedance state if the pin state specification bit in the standby control register (STBC: SPL) is "1" when the device changes to stop or watch mode.

Table 4.5-4 "Port-4, 6, and 7 Pin State" lists the Port-4, 6, and 7 pin states.

Table 4.5-4 Port-4, 6, and 7 Pin State

Pin name	Normal operation main-sleep mode main-stop mode (SPL=0) sub-sleep mode sub-stop mode (SPL=0) watch mode (SPL=0)	Main-stop mode (SPL = 1) sub-stop mode (SPL = 1) watch mode (SPL = 1)	Reset
P40/SEG0 to P47/SEG7	Output-only port/LCDC segment output	Hi-z*1	Hi-z*2
P60/SEG8 to P65/SEG13	Output-only port/LCDC segment output		
P70/COM2, P71/COM3	Output-only port/LCDC common output		

*1: Does not go to the high-impedance state when the pin is used as an LCDC common and segment output.

*2: Goes "L" when the pin is used as an LCDC common and segment output.

SPL: Pin state specification bit in the standby control register (STBC)

Hi-Z: High impedance

Note:

Pins with a pull-up resistor go to the "H" level (pull-up state) rather than to the high-impedance state when the output transistor is turned "OFF".

4.6 Port 5

Port 5 is an output-only port that also serves as an analog input. Each pin can be switched between analog input and port operation in bit units. This section principally describes the port functions when operating as an output-only port.
The section describes the port structure and pins, the pin block diagram, and the port register for port 5.

■ Structure of Port 5

Port 5 consists of the following three components:

- Output-only pins/analog input pins (P50/AN0 to P53/AN3)
- Port-5 data register (PDR5)
- Port 5 pull-up control register (PURR5) (only available in MB89P985 and MB89PV980)

■ Port-5 Pins

Port 5 consists of four output pins of an N-ch open-drain output type. Do not use these pins as output-only ports when the pins are used as the analog input for the A/D converter.

Table 4.6-1 "Port-5 Pin" lists the port-5 pins.

Table 4.6-1 Port-5 Pin

Port	Pin name	Function	Shared peripheral	I/O type		Circuit type	
				Input	Output	MB89983	MB89P985 MB89PV980
Port 5	P50/AN0	P50 Output-only	AN0 Analog input 0	Analog	N-ch open-drain	P	Q
	P51/AN1	P51 Output-only	AN1 Analog input 1				
	P52/AN2	P52 Output-only	AN2 Analog input 2				
	P53/AN3	P53 Output-only	AN3 Analog input 3				

Reference:

See Section 1.7 "I/O Pins and Pin Functions" for a description of the circuit type.

See CHAPTER 11 "A/D CONVERTER" for details of pin operation when used as an analog input.

■ Block Diagram of Port-5 Pin

Figure 4.6-1 Block Diagram of Port-5 for MB89983

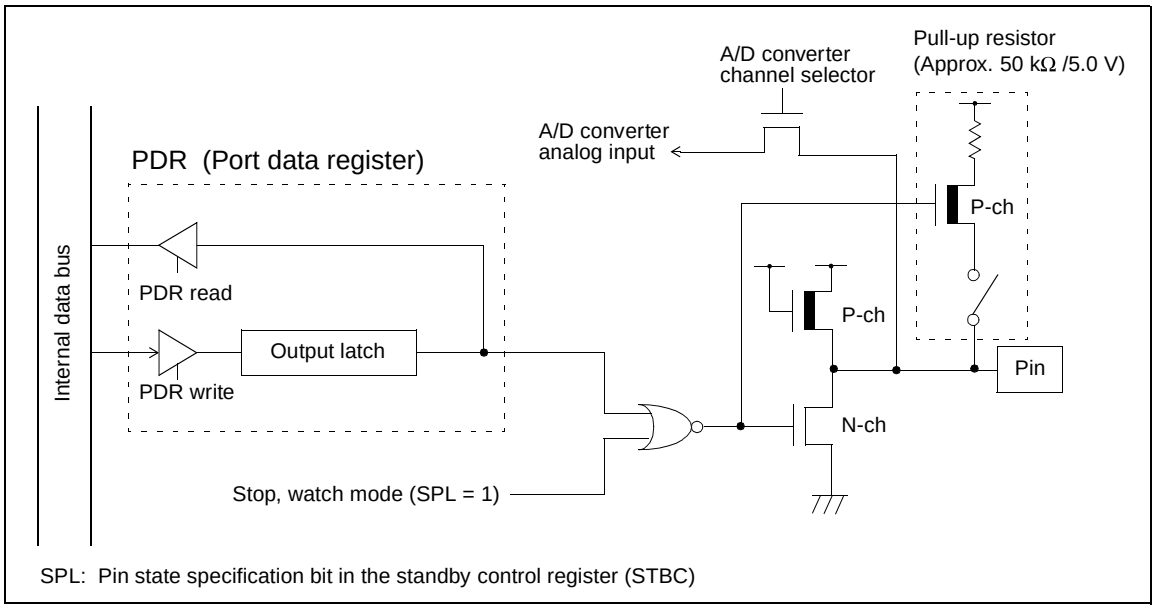
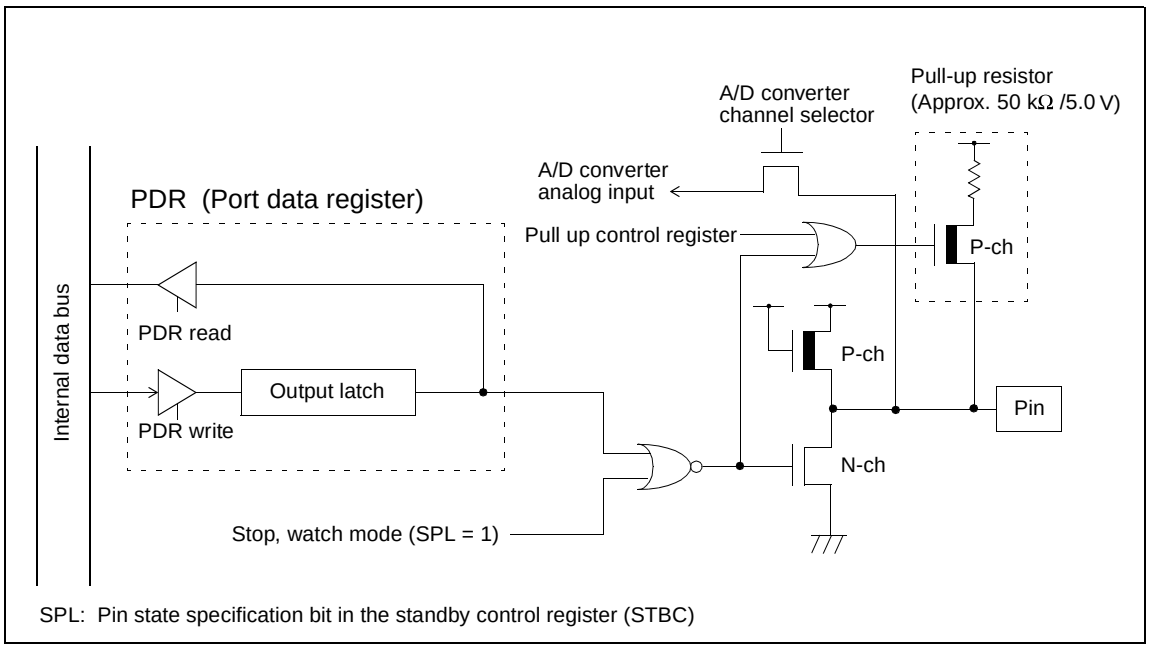


Figure 4.6-2 Block Diagram of Port-5 for MB89P985 and MB89PV980



Check:

If using the A/D converter, do not set a pull-up resistor for any of P53/AN3 to P50/AN0.

Check:

Do not use the pins as output ports if using as an analog input.

■ Port-5 Registers

The port-5 register consists of PDR5. Each bit in the register has a one-to-one relationship with a port 5 pin.

Table 4.6-2 "Correspondence between Pin and Register for Port 5" shows the correspondence between the pins and register for port-5.

Table 4.6-2 Correspondence between Pin and Register for Port 5

Port	Correspondence between register bit and pin								
Port 5	PDR5, PURR5	-	-	-	-	Bit 3	Bit 2	Bit 1	Bit 0
	Corresponding pin	-	-	-	-	P53	P52	P51	P50

4.6.1 Port-5 Register (PDR5)

This section describes the port-5 register.

■ Port-5 Register Functions

❍ Port 5 data register (PDR5)

The PDR5 register holds the output latch states. Therefore, pin states cannot be checked by reading this register.

❍ Settings as an analog input

When Port 5 pins are used as analog signal inputs, write "1" to the corresponding bits of PDR5 to turn the output transistors "OFF". Setting the pins to the high-impedance state.

Table 4.6-3 "Port-4, Port-6, and Port-7 Register Function" lists the functions of the port-5 register.

Table 4.6-3 Port-4, Port-6, and Port-7 Register Function

Register	Data	Read	Write	Read/ Write	Address	Initial value
Port 5 data register (PDR5)	0	Output latch value is "0".	Outputs an "L" level to the pin. (Sets "0" to the output latch and turn the output transistor "ON")	R/W	$000F_H$	$XXXX1111_B$
	1	Output latch value is "1".	Sets the pin to the high-impedance state*. (Sets "1" to the output latch and turn the output transistor "OFF.)			

R/W: Readable and writable

*: Pins with a pull-up resistor go to the pull-up state.

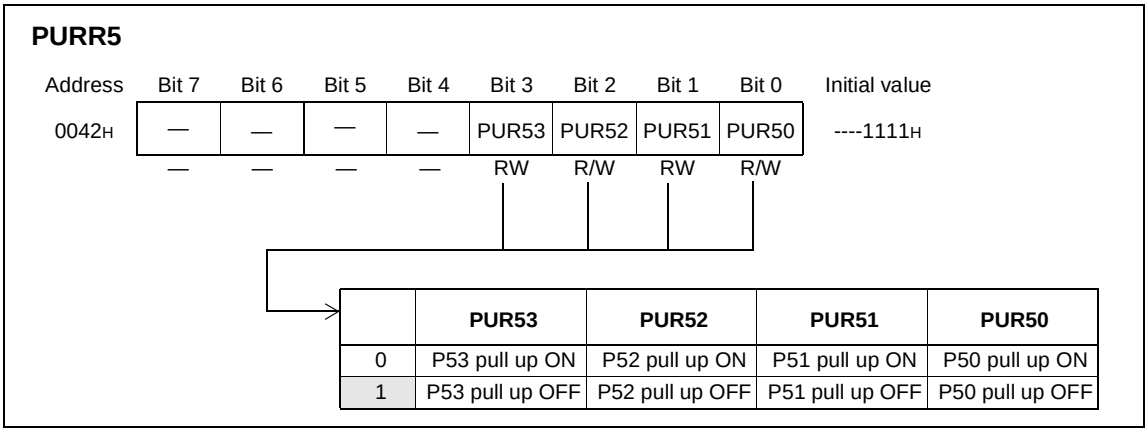
❍ Port 5 pull-up control registers (PURR5)

By using pull-up resistor option in each pin for port 5, setting is possible bit by bit when writing to pull-up control register for MB89P985 and MB89PV980 only. The pull-up resistor for MB89983 is selected by mask option.

When pull-up resistor is selected in pull-up register in stop and clock mode (SPL=1). The state of these pin are in "H" level (pull up state) rather than high impedance. However, during reset, pull up is unavailable and will be in high impedance state.

Figure 4.6-3 "Pull up control registers setting (PURR5)" is list of the pull-up resistor option setting of PURR5

Figure 4.6-3 Pull up control registers setting (PURR5)



	PUR53	PUR52	PUR51	PUR50
0	P53 pull up ON	P52 pull up ON	P51 pull up ON	P50 pull up ON
1	P53 pull up OFF	P52 pull up OFF	P51 pull up OFF	P50 pull up OFF

Note:

For MB89P985 and MB89PV980, do not enabled the pull-up resistor when P50/AN0 to P53/AN3 are used the A/D converter input.

4.6.2 Operation of Port 5

This section describes the operations of the port 5.

■ Operation of Port 5

❍ Operation as an output port

- Writing data to the PDR5 register stores the data in the output latch. When the output latch value is "0", the output transistor turns "ON" and an "L" level is output from the pin. When the output latch value is "1", the transistor turns "OFF" and the pin goes to the high-impedance state. If a pull-up is provided the output pin the pin goes to the pull-up state when the output latch value is "1".
- Reading the PDR5 register always returns the output latch value.

❍ Operation as an analog input

- Set the PDR5 bit that corresponds to the analog input pin to "1" to turn the output transistor "OFF".
- Reading the PDR5 register always returns the output latch value.

❍ Operation at reset

Resetting the CPU initializes the PDR5 register value to "1". This turns all the output transistors "OFF" and sets the pins to the high-impedance state.

❍ Operation in stop and watch modes

The output transistors are forcibly turned "OFF" and the pins go to the high-impedance state if the pin state specification bit in the standby control register (STBC: SPL) is "1" when the device changes to stop or watch mode.

Table 4.6-4 "Port-5 Pin State" lists the port-5 pin states.

Table 4.6-4 Port-5 Pin State

Pin name	Normal operation main-sleep mode main-stop mode (SPL=0) sub-sleep mode sub-stop mode (SPL=0) watch mode (SPL=0)	Main-stop mode (SPL = 1) sub-stop mode (SPL = 1) watch mode (SPL = 1)	Reset
P50/AN0 to P53/AN3	Output-only ports/analog input	Hi-z	Hi-z

SPL: Pin state specification bit in the standby control register (STBC)

Hi-z: High impedance

Note:

Pins with a pull-up resistor go to the "H" level (pull-up state) rather than to the high-impedance state when the output transistor is turned "OFF".

4.7 Program Example for I/O Ports

This section gives an example program using the I/O ports.

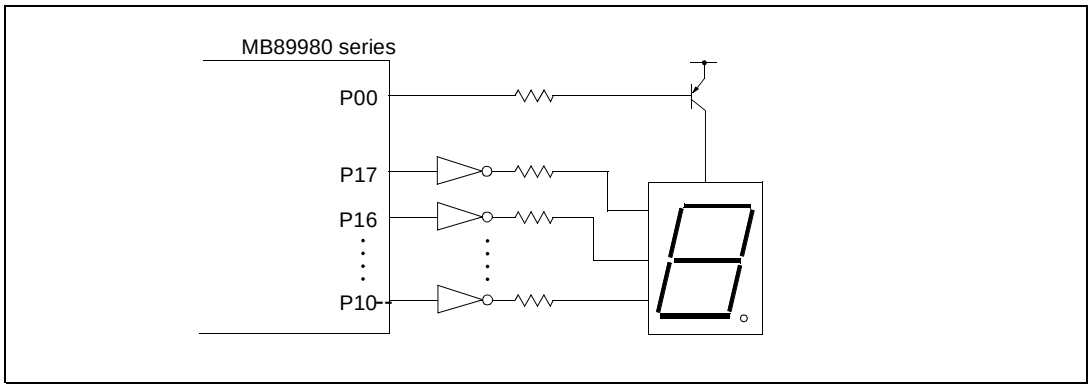
■ Program Example for I/O Ports

❍ Processing description

- Port 0 and port 1 are used to illuminate all elements of seven segment LED (eight segments if the decimal point is included).
- The P00 pin is used for the anode common pin of the LED and the P10 to P17 pins operate as the segment pins.

Figure 4.7-1 "Connection Example for an Eight Segment LED" shows the connection example for an eight segment LED.

Figure 4.7-1 Connection Example for an Eight Segment LED



❍ Coding example

```
PDR0    EQU     0000H           ; Address of the Port 0 data register
DDR0    EQU     0001H           ; Address of the Port 0 data direction register
PDR1    EQU     0002H           ; Address of the Port 1 data register
DDR1    EQU     0003H           ; Address of the Port 1 data direction register
;-------- Main program ----------------------------------------------------------------------------------------
CSEG;  [CODE SEGMENT]
            :
            CLRB    PDR0:0              ; Set P00 to the "L" level.
            MOV     PDR1, #11111111B    ; Set all port-1 pins to the "H" level.
            MOV     PDR1, #11111111B    ; Set P00 as an output (#XXXXXXX1B).
```

```
                MOV         PDR1, #11111111B      ; Set all port-1 pins as outputs.
                :
                ENDS
;-------- Main program ------------------------------------------------------------------------------------------
                END
```

CHAPTER 5 TIMEBASE TIMER

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

5.1 Overview of Timebase Timer

The timebase timer provides interval timer functions. Four different interval times can be selected. The timebase timer uses a 21-bit free-run counter which counts-up in sync with the internal count clock (divide-by-two the main clock source oscillation). The timebase timer also provides the timer output for the oscillation stabilization delay time and the operating clock for the watchdog and timers.
The timebase timer stops operating in modes in which the main clock master oscillator is stopped.

■ Interval Timer Function

The interval timer function generates repeated interrupts at fixed time intervals.

- The timer generates an interrupt each time the interval timer bit overflows on the timebase timer counter.
- The interval timer bit (interval time) can be selected from four different settings.

Table 5.1-1 "Timebase timer Interval time" lists the available interval for the timebase timer.

Table 5.1-1 Timebase timer Interval time

Internal count clock cycle	Interval time
2/FCH (0.48 µs)	$2^{13}/F_{CH}$ (approx. 1.95 ms)
	$2^{15}/F_{CH}$ (approx. 7.80 ms)
	$2^{18}/F_{CH}$ (approx. 62.4 ms)
	$2^{22}/F_{CH}$ (approx. 998.6 ms)

F_{CH}: Main clock source oscillation

The values enclosed in parentheses ( ) are for a 4.2 MHz main clock source oscillation.

■ Clock Supply Function

The clock supply function provides the timer output used for the main clock oscillation stabilization delay time (four values), and operation clock for some peripheral functions.

Table 5.1-2 "Clocks Supplied by Timebase Timer" lists the cycles of the clocks that the timebase timer supplies to various peripherals.

Table 5.1-2 Clocks Supplied by Timebase Timer

Clock destination	Clock cycle	Remarks
Main clock oscillation stabilization delay time	$2^4/F_{CH}$ (approx. 0.0 ms)	Selected by the oscillator stabilization wait time select bit of the system clock control register (SYCC: WT1, WT0), which is in the clock control section.
	$2^{12}/F_{CH}$ (approx. 0.98 ms)	
	$2^{16}/F_{CH}$ (approx. 15.6 ms)	
	$2^{18}/F_{CH}$ (approx. 62.4 ms)	
Watchdog timer	$2^{22}/F_{CH}$ (approx. 998.6 ms)	Count-up clock for the watchdog timer
A/D converter	$2^8/F_{CH}$ (approx. 61.0 µs)	Clock for continuous activation
LCD controller/driver	$2^7/F_{CH}$ (approx. 30.5 µs)	Frame cycle clock

F_{CH}: Main clock source oscillation

The values enclosed in parentheses ( ) are for a 4.2 MHz main clock source oscillation.

Note:

The oscillation stabilization delay time should be used as a guide line since the oscillation cycle is unstable immediately after oscillation starts.

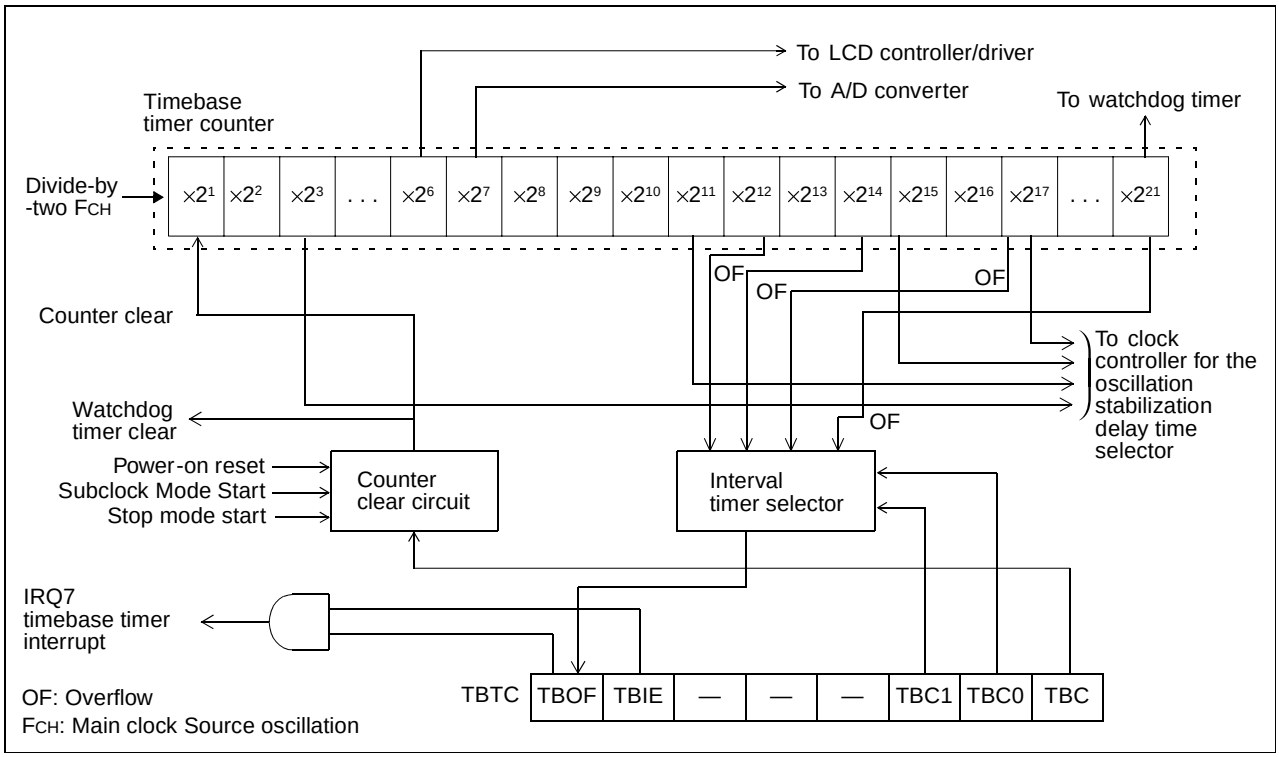
5.2 Block Diagram of Timebase Timer

The timebase timer consists of the following four blocks:

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

■ Block Diagram of Timebase Timer

Figure 5.2-1 Block Diagram of Timebase Timer



❍ Timebase timer counter

A 21-bit up-counter that uses the divide-by-two main clock source oscillation as a count clock. The counter stops when the main clock oscillator is stopped.

❍ Counter clear circuit

In addition to being cleared by setting the TBTC register (TBR="0"), the counter is cleared when device changes to main stop (STBC: STP ="1") and subclock (SYCC: SCS="0") mode and by power-on reset (optional).

❍ **Interval timer selector**

Selects one of four operating timebase timer counter bits as the interval timer bit. An overflow on the selected bit triggers an interrupt.

❍ **TBTC register**

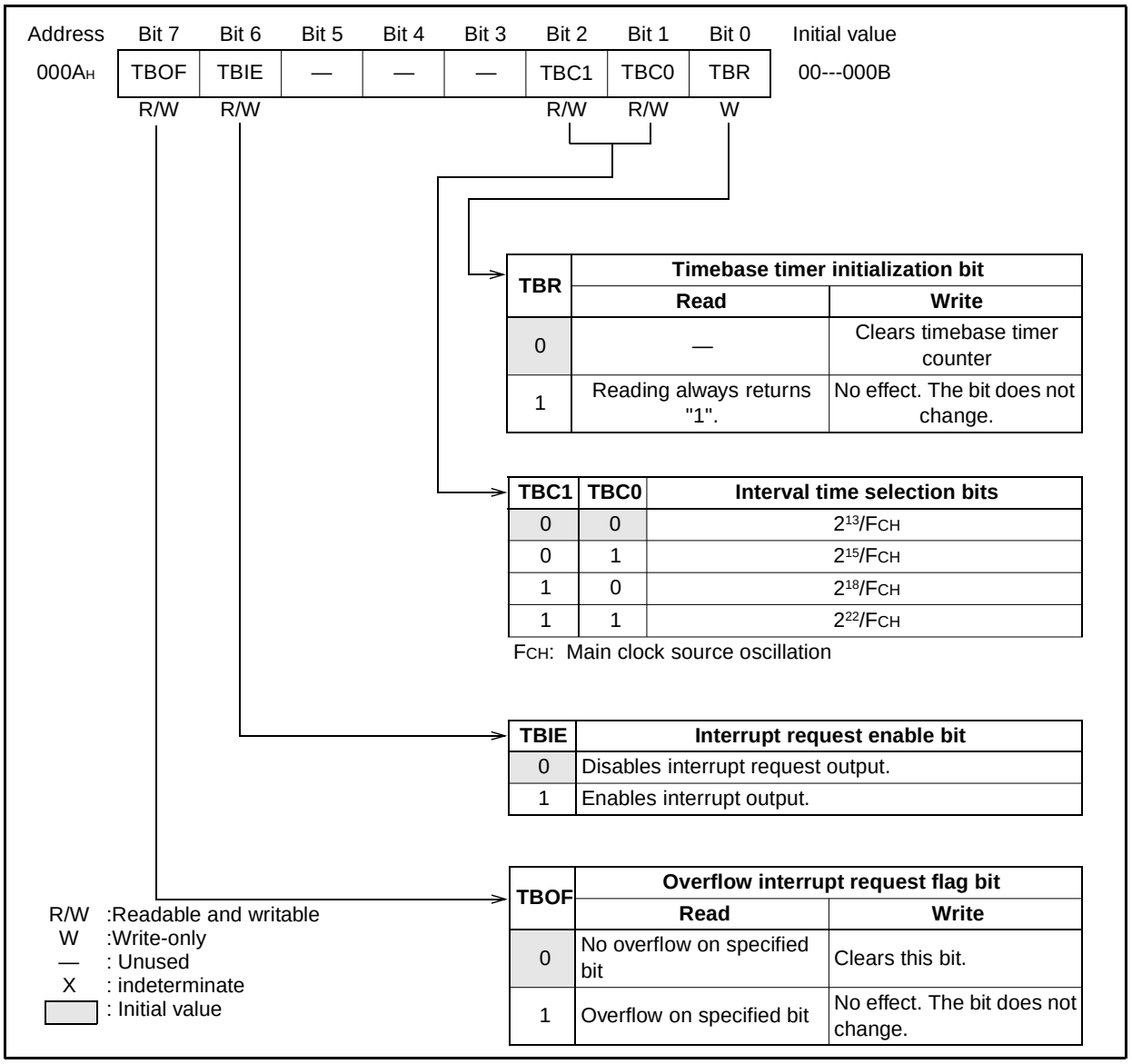
The TBTC register is used to select the interval timer bit, clear the counter, control interrupts, and check the state of the timebase timer.

5.3 Timebase Timer Control Register (TBTC)

The timebase timer control register (TBTC) is used to select the interval times bit, clear the counter, control interrupts, and check the state of the timebase timer.

■ Timebase Timer Control Register (TBTC)

Figure 5.3-1 Timebase Timer Control Register (TBTC)



TBR	Timebase timer initialization bit	
	Read	Write
0	—	Clears timebase timer counter
1	Reading always returns "1".	No effect. The bit does not change.

TBC1	TBC0	Interval time 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}: Main clock source oscillation

TBIE	Interrupt request enable bit
0	Disables interrupt request output.
1	Enables interrupt output.

TBOF	Overflow interrupt request flag bit	
	Read	Write
0	No overflow on specified bit	Clears this bit.
1	Overflow on specified bit	No effect. The bit does not change.

Table 5.3-1 Timebase Timer Control Register (TBTC) Bits

Bit		Function
Bit 7	TBOF: Overflow interrupt request flag bit	• This bit is set to "1" when count an overflow occurs on the specified bit of the timebase timer counter. • An interrupt request is output when both this bit and the interrupt request enable bit (TBIE) are "1". • Writing "0" clears this bit. Writing "1" has no effect and does not change the bit value.
Bit 6	TBIE: interrupt request	This bit enables or disables an interrupt request output to the CPU. An interrupt request is output when both this bit and the overflow interrupt request flag bit (TBOF) are "1".
Bit 5 Bit 4 Bit 3	Unused bits	• The read value is indeterminate. • Writing to these bits has no effect on the operation.
Bit 2 Bit 1	TBC1, TBC0: Interval time selection bits	These bits select the cycle of the interval timer. These bits select which bit of the timebase timer counter to use as the interval timer bit. Four different interval times can be selected.
Bit 0	TBR: Timebase timer initialization bit	This bit clears the timebase timer counter. writing "0" to this bit clears the counter to "000000_H". Writing "1" has no effect and does not change the bit value. **Note:** The read value is always "1".

5.4 Timebase Timer Interrupt

The timebase timer can generate an interrupt request when an overflow occurs on the specified bit of the timebase counter (for the interval timer function).

■ Interrupts for Interval Timer Function

The counter counts-up on the internal count clock. When an overflow occurs on the selected interval timer bit, the overflow interrupt request flag bit (TBTC: TBOF) is set to "1". At this time, an interrupt request (IRQ7) to the CPU is generated if the interrupt request enable bit is enabled (TBTC: TBIE ="1"). Write "0" to the TBOF bit in the interrupt processing routine to clear the interrupt request. The TBOF bit is set when at the specified counter bit overflows, regardless of the TBIE bit value.

Check:

When enabling an interrupt request output (TBIE = "1") after wake-up from a reset, always clear the TBOF bit (TBOF = "0") at the same time.

Notes:

An interrupt request is generated immediately if the TBOF bit is "1" when the TBIE bit is changed from disabled to enabled ("0" --> "1").

The TBOF bit is not set if the counter is cleared (TBTC: TBR = "0") at the same time as an overflow on the specified bit occurs.

■ Oscillation Stabilization Delay Time and Timebase Timer Interrupt

If the interval time is set shorter than the main clock oscillation stabilization delay time, an interval interrupt request from the timebase timer (TBTC: TBOF = "1") is generated at the time when the main clock mode starts operation. In this case, disable the timebase timer interrupt when changing to a mode in which the main clock oscillation is stopped (main stop mode and subclock mode).

■ Register and Vector Table for Timebase Timer Interrupts

Table 5.4-1 Register and Vector Table for Timebase Timer Interrupt

Interrupt	Interrupt level settings register			Vector table address	
	Register	Set bit		Upper	Lower
IRQ7	ILR2($007D_H$)	L71 (Bit 7)	L70 (Bit 6)	$FFEC_H$	$FFED_H$

Reference:

See Section 3.4.2 "Interrupt Processing" for details on the operation of interrupt.

5.5 Operation of Timebase Timer

The timebase timer has the interval timer function and the clock supply function for some peripherals.

■ Operation of Interval Timer Function (Timebase Timer)

Figure 5.5-1 "Interval Timer Function Settings" shows the settings required to operate the interval timer function.

Figure 5.5-1 Interval Timer Function Settings

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TBTC	TBOF	TBIE	—	—	—	TBC1	TBC0	TBR
	0	1				◎	◎	0

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

Provided the main clock is oscillating, the timebase timer counter continues to count-up in sync with the internal count clock (divide-by-two main clock source oscillation).

After being cleared (TBR = "0"), the counter restarts counting-up from zero. The timebase timer sets the overflow interrupt request flag bit (TBOF) to "1" when an overflow occurs on the interval timer bit. Consequently, the timebase timer generates interrupt requests at fixed intervals (the selected interval time), based on the time that the counter is cleared.

■ Operation of Clock Supply Function

The timebase timer is also used as a timer to generate the main clock oscillation stabilization delay time. The time from when the timebase timer counter is cleared and starts counting-up until an overflow occurs on the oscillation stabilization delay time bit is the oscillation stabilization delay time. One of four possible delay times is selected by the oscillation stabilization delay time bits of the system clock control register (SYCC: WT1, WT0).

The timebase timer also provides the clock for the watchdog timer and buzzer output, A/D converter, and LCD controller/driver. Clearing the timebase timer counter affects the operation of the continuous activation cycle of the A/D converter, the frame cycle of the LCD controller/driver, and the buzzer output. If the timebase timer output is selected by the watchdog timer counter (WDTC: CS=0), it will also be cleared simultaneously.

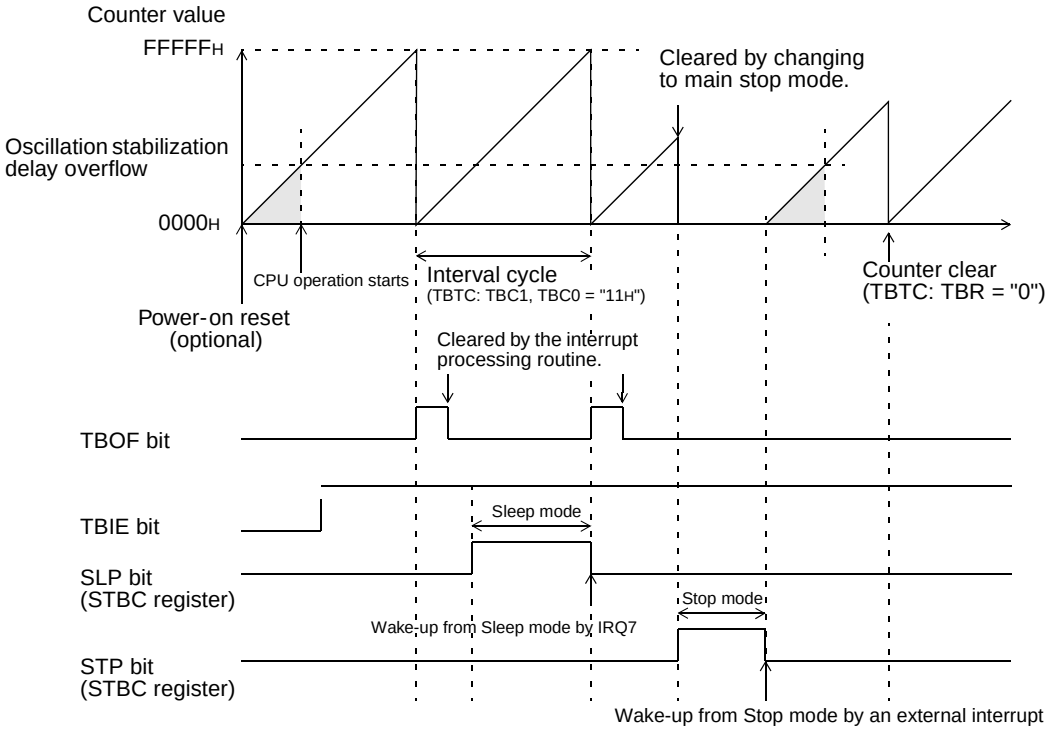
■ Operation of Timebase Timer

The state of following operations are shown in Figure 5.5-2 "Operation of Timebase Timer".

- A power-on reset occurs.
- Changes to sleep mode during operation of the interval timer function in the main clock mode.
- Changes to main stop mode.
- A counter clear request occurs.

The timebase timer is cleared by changing to subclock and main stop modes, and stops operation. The timebase timer counts the oscillation stabilization delay time after wake-up from subclock and main stop modes.

Figure 5.5-2 Operation of Timebase Timer



For the case when the interval time selection bits in the timebase timer control register (TBTC: TBC1, TBC0) are set to "11" (2^{21}/FC).

: Indicates the oscillation stabilization delay time.

5.6 Notes on Using Timebase Timer

This section lists points to note when using the timebase timer.

■ Notes on Using Timebase Timer

❍ Notes on setting bits by program

The system cannot recover from interrupt processing if the overflow interrupt request flag bit (TBTC: TBOC) is "1" and the interrupt request enable bit is enabled (TBTC: TBIE = "1"). Always clear the TBOF bit.

❍ Clearing timebase timer

In addition to being cleared by the timebase timer initialization bit (TBTC: TBR = "0"), the timer is cleared whenever the main clock oscillation stabilization delay time is required. When the timebase timer is selected as a count clock of the watchdog timer, clearing the timebase timer also clears the watchdog timer.

❍ Using as timer for oscillation stabilization delay time

As the main clock source oscillation is stopped when the power is turned on during main-stop mode, and during subclock mode, the timebase timer provides the oscillation stabilization delay time after the oscillator starts.

An appropriate oscillation stabilization delay time must be selected for the type of resonator connected to the main clock oscillator (clock generator).

Reference:

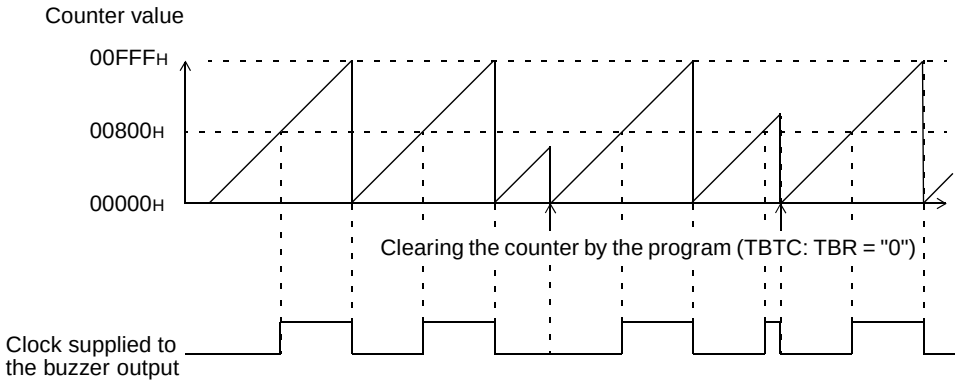
See Section 3.6.1 "Clock Generator Section".

❍ Notes on peripheral functions that provided a clock supply from timebase timer

In modes in which the main clock source oscillation is stopped, the timebase timer also stops, and the counter is cleared. As the clock derived from the timebase timer restarts output from its initial state when the timebase timer counter is cleared, the "H" level may be shorter or the "L" level longer by a maximum of half cycle. The clock of the watchdog timer also restarts output from its initial state. However, as the watchdog timer counter is cleared at the same time, the watchdog timer operates in normal cycle.

Figure 5.6-1 "Effect on Buzzer Output of Clearing Timebase Timer" shows the effect on the buzzer output of clearing the timebase timer.

Figure 5.6-1 Effect on Buzzer Output of Clearing Timebase Timer



For the case when the buzzer selection bits in the buzzer register (BZCR: BZ1, BZ0) are set to "01B".
(Divide-by-8192 source oscillation, approximately 1.221 kHz output at 10 MHz operation)

5.7 Program Example for Timebase Timer

This section gives a program example for the timebase timer.

■ Program Example for Timebase Timer

❍ Processing description

- Generates repeated interval timer interrupts at $2^{18}/F_{CH}$ (F_{CH}: the main clock source oscillation) intervals. At this time, the interval time is approximately 52.4 ms (at 4.2 MHz operation).

❍ Coding example

```
TBTC    EQU     0000AH          ; Address of the timebase timer control register
TBOF    EQU     TBTC:3          ; Define the interrupt request flag bit.
ILR2    EQU     007EH           ; Address of the interrupt level setting register 2
INT_V   DSEG    ABS             ; [DATA SEGMENT]
        ORG     0FFECH
INQ7    DW      WARI            ; Set interrupt vector.
INT_V   ENDS
;-------- Main program ------------------------------------------------------------------------------------
        CSEG                    ;  [CODE SEGMENT]
                                ; Stack pointer (SP) etc. are already initialized.
        :
        CLRI                    ; Disable interrupts.
        MOV     ILR2,#01111111B ; Set interrupt level (level 1).
        MOV     TBTC,#00000100B ; Clear interrupt request flag enable interrupt
                                  request output, select 218/FC, and clear
                                  timebase timer.
        SETI                    ; Enable interrupts.
        :
;-------- Interrupt program-------------------------------------------------------------------------------
WARI    CLRB    TBOF            ; Clear 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 operation of the watchdog timer.

6.1 Overview of Watchdog Timer

The watchdog timer is a 1-bit counter that uses, as its count clock source, either the timebase timer derived from the main clock, or the watch prescaler derived from the subclock. The watchdog timer resets the CPU if not cleared within a fixed time after activation.

■ Watchdog Timer Function

The watchdog timer is a counter provided to guard against program runaway. Once activated, the counter must be repeatedly cleared within a fixed time interval. If the program becomes trapped in an endless loop or similar and does not clear the counter within the fixed time, the watchdog timer generates a four-instruction cycle watchdog reset to the CPU.

Either the timebase timer output or the watch prescaler output can be selected as the watchdog timer count clock.

Table 6.1-1 "Watchdog Timer Interval Time" lists the watchdog timer interval times. If not cleared, the watchdog timer generates a watchdog reset at a time between the minimum and maximum times listed. Clear the counter within the minimum time given in the .

Table 6.1-1 Watchdog Timer Interval Time

	Count clock	
	Watch timer output (main clock oscillator frequency at 4.2 MHz)	**Watch prescaler output (subclock oscillator frequency at 32.768 kHz)**
Minimum time	Approx. 998.6 ms*1	500 ms*2
Maximum time	Approx. 1997.3 ms	1000 ms

*1: Divide-by-two the main clock source oscillation (F_{CH})× timebase timer count value (2^{21}).

*2: The time of a clock cycle at the subclock oscillator frequency (F_{CL})× timeclock prescaler count (2^{14}).

Reference:

See Section 6.4 "Watchdog Timer Operation" for the details on the minimum and maximum time of the watchdog timer interval times.

Check:

When the timebase timer output is selected as the watchdog timer counter count clock, the watchdog timer will be cleared at the same time (and each time) the timebase timer counter is cleared (TBTC: TBR = 0). When the timeclock prescaler is selected as the count clock, the watchdog timer will be cleared at the same time as the timeclock prescaler is cleared (WPCR: WCLR = 0). Therefore, if the counter selected as the count clock (either the timebase timer or the timeclock prescaler) is repeatedly cleared before the end of the watchdog timer interval, the watchdog timer will fail to perform its intended function.

Note:

The watchdog timer counter is cleared whenever the device changes to sleep or stop watch mode. Operation halts until the device returns to normal operation (RUN state).

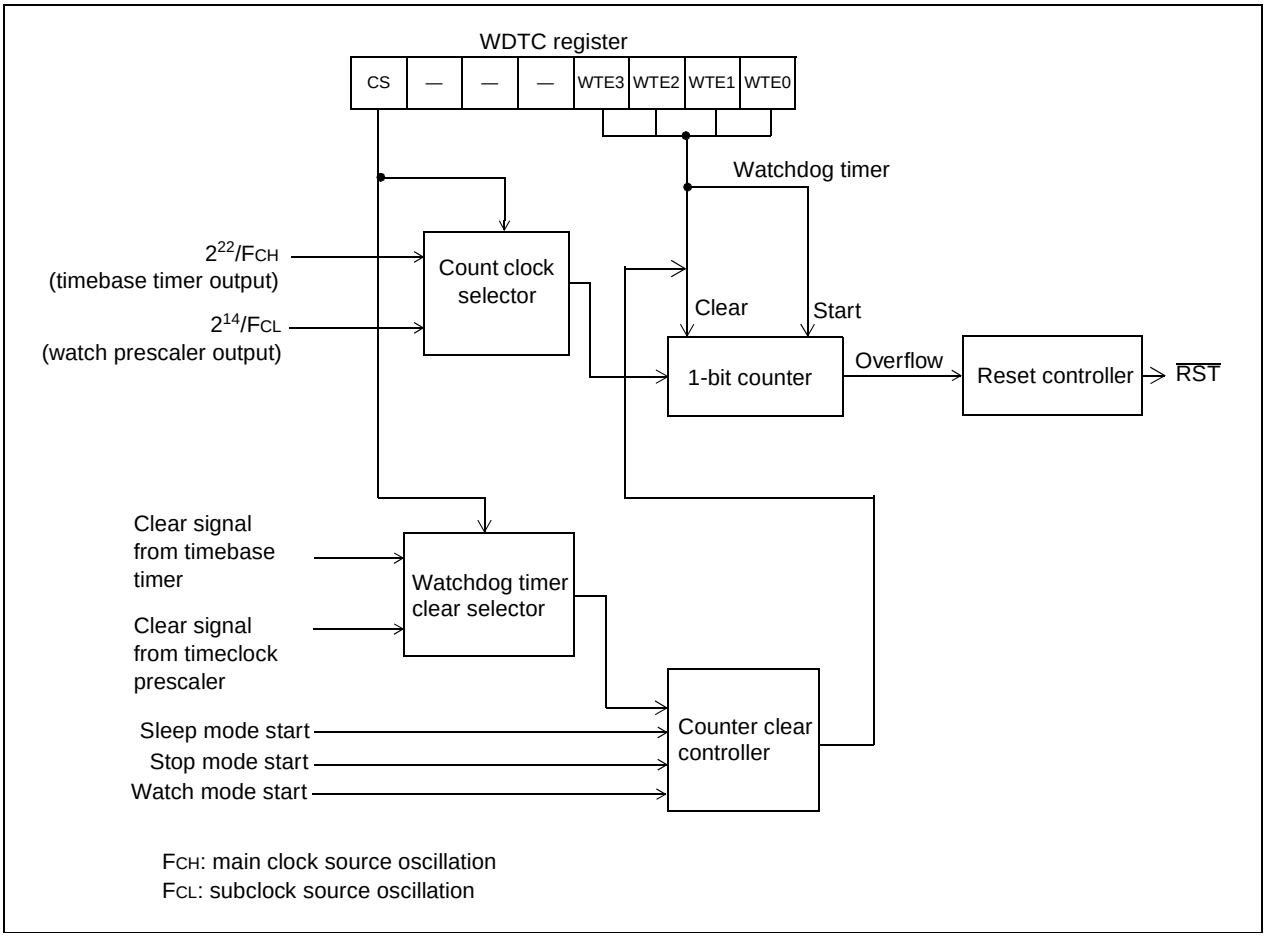
6.2 Block Diagram of Watchdog Timer

The watchdog timer consists of the following six blocks:

- **Count clock selector**
- **Watchdog timer counter**
- **Reset controller**
- **Watchdog timer clear selector**
- **Counter clear controller**
- **Watchdog timer control register (WDTC)**

■ Block Diagram of Watchdog Timer

Figure 6.2-1 Block Diagram of Watchdog Timer



- **Count clock selector**

  The count clock selector selects the count clock for the watchdog timer counter. Either the timebase timer counter output or the timeclock prescaler output can be selected as the count clock.

- **Watchdog timer counter (1-bit counter)**

  A 1-bit counter that uses the timebase timer output or the watch prescaler output as a count clock

- **Reset controller**

  Generates a reset signal to the CPU when an overflow occurs on the watchdog timer counter.

- **Watchdog timer clear selector**

  The clear selector selects a watchdog timer clear signal from either the timebase timer or timeclock prescaler at the same time as the count clock selector selects a clock. (It selects the clear signal from the selected clock source.)

- **Counter clear controller**

  Controls clearing and halting the operation of watchdog timer counter.

- **WDTC register**

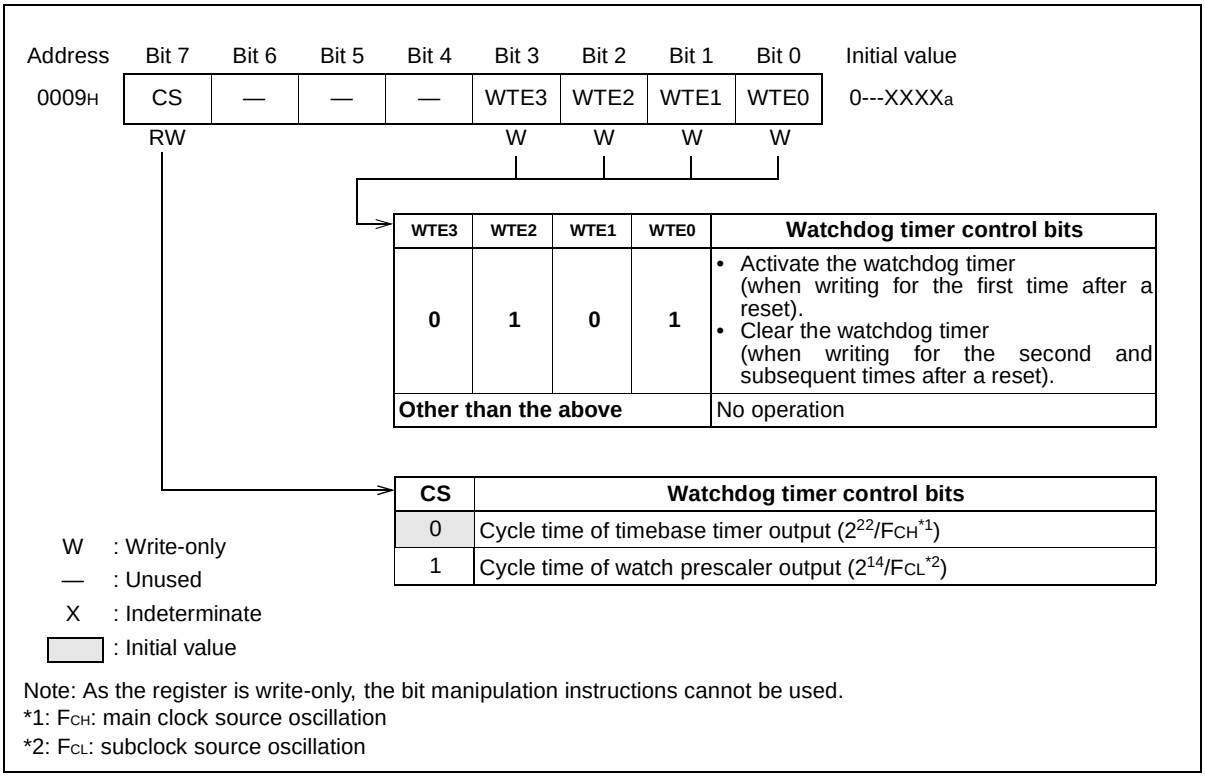
  The WDTC register is used to select the count clock, and to activate or clear the watchdog timer counter. As the register is write-only, the bit manipulation instructions cannot be used.

6.3 Watchdog Timer Control Register (WDTC)

The watchdog timer control register (WDTC) is used to activate or clear the watchdog timer.

■ Watchdog Timer Control Register (WDTC)

Figure 6.3-1 Watchdog timer Control Register (WDTC)



WTE3	WTE2	WTE1	WTE0	Watchdog timer control bits
0	1	0	1	• Activate the watchdog timer (when writing for the first time after a reset). • Clear the watchdog timer (when writing for the second and subsequent times after a reset).
Other than the above				No operation

CS	Watchdog timer control bits
0	Cycle time of timebase timer output ($2^{22}/F_{CH}$*1)
1	Cycle time of watch prescaler output ($2^{14}/F_{CL}$*2)

Note: As the register is write-only, the bit manipulation instructions cannot be used.
*1: F_{CH}: main clock source oscillation
*2: F_{CL}: subclock source oscillation

Table 6.3-1 Watchdog Timer Control Register (WDTC) Bits

Bit		Function
Bit 7	CS: Count clock select bit	• At watchdog timer startup, selects the watchdog timer count clock. Selects either the timebase timer output or the timeclock prescaler output as the count clock. **Note:** When using the subclock mode, always select the timeclock prescaler output. Make the count clock selection when the watchdog timer is started. Once the timer is started, do not change the count clock. Bit operation instructions cannot be used.
Bit 6 Bit 5 Bit 4	Unused bits	• The read value is indeterminate. • Writing to these bits has no effect on operation.
Bit 3 Bit 2 Bit 1 Bit 0	WTE3, WTE0: Watchdog timer control bits	• Writing "0101_B" to these bits activates (when writing for the first time after a reset) or clears (when writing for the second and subsequent times after a reset) the watchdog timer. • Writing a value other than "0101_B" has no effect on the operation. **Check:** The read value is "1111_B" The bit manipulation instructions cannot be used.

6.4 Operation of Watchdog Timer

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

■ Operation of Watchdog Timer

❍ Activating watchdog timer

- The watchdog timer is activated by writing "0101_B" to the watchdog control bits in the watchdog control register (WDTC: WTE3 to WTE0) for the first time after a reset.
  The count clock select bit (WDTC: CS) is written to the desired state in the same write operation.
- Once activated, the watchdog timer cannot be stopped other than by a reset.

❍ Clearing watchdog timer

- The watchdog timer counter is cleared by writing "0101_B"to the watchdog control bits in the watchdog control register (WDTC: WTE3 to WTE0) for the second or subsequent times after a reset.
- If the counter is not cleared within the interval time of the watchdog timer, the counter overflows and the watchdog timer generates an internal reset signal for four-instruction cycles.

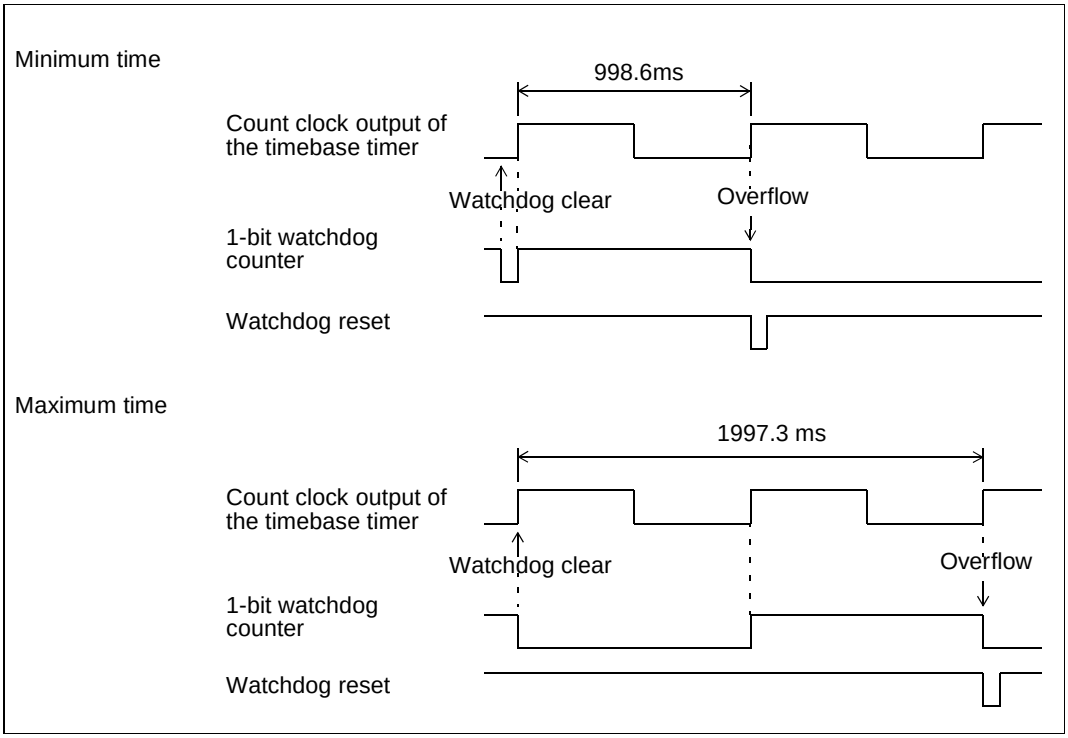
❍ Interval time of watchdog timer

The interval time changes depending on when the watchdog timer is cleared.

Figure 6.4-1 "Watchdog Timer Clear and Interval Time" shows the relationship between the watchdog timer clear timing and the interval time.

The indicated times apply if the timebase timer output is selected as the count clock, and the main clock source oscillation is 4.2 MHz.

Figure 6.4-1 Watchdog Timer Clear and Interval Time



6.5 Notes on Using Watchdog Timer

This section lists points to note when using the watchdog timer.

■ Notes on Using Watchdog Timer

❍ Stopping watchdog timer

Once activated, the watchdog timer can not stop until a reset generates.

❍ Count clock selection

The only time the count clock select bit (WDTC: CS) can be changed is at watchdog timer activation. You can change it by writing the desired state to the count clock select bit (WDTC: CS) at the same time as you write "0101_B" to the watchdog control bits (WDTC: WTE3 to WTE0) to activate the watchdog timer. Therefore, the CS bit cannot be changed by a bit operation instruction. Do not change the CS bit after activating the timer.

In the subclock mode, the main clock source oscillation is stopped, which means that the timebase timer also stops.

For the watchdog timer to operate in subclock mode, then, the watch prescaler must have been selected in advance as the count clock (WDTC: CS = 1).

❍ Clearing watchdog timer

- Clearing the counter being used as a count clock of the watchdog timer (timebase timer or watch prescaler) also simultaneously clears the watchdog timer counter.
- The watchdog timer counter is cleared on changing to sleep, stop or watch mode.

❍ Notes on programming

When writing a program in which the watchdog timer is repeatedly cleared in the main loop, including interrupt processing, is less than the minimum watchdog timer interval time.

❍ Operation in subclock mode

If the watchdog reset signal is generated in subclock mode, operation will start in main clock mode after an oscillation stabilization delay time. Therefore, if the device has the reset signal output option, a reset signal will be output during the oscillation stabilization delay time.

6.6 Program Example for Watchdog Timer

This section gives a program example for the watchdog timer.

■ Program Example for Watchdog Timer

❍ Processing description

- Selects the watch prescaler as the count clock and activates the watchdog timer immediately after the program.
- Clears the watchdog timer in each loop of the main program.
- The processing time for the main loop, including interrupt processing, must be less than the minimum interval time of the watchdog timer (approximately 209.7 ms at 10 MHz operation).

❍ Coding example

```
WDTC      EQU   00009H              ; Address of the watchdog timer control register
WDT_CLR   EQU   10000101B

VECT      DSEG  ABS                 ; [DATA SEGMENT]
          ORG   0FFFEH
RST_V     DW    PROG                ; Set reset vector.
VECT      ENDS

;-------- Main program ------------------------------------------------------------------------------------

          CSEG                      ; [CODE SEGMENT]
PROG                                ; Initialization routine after a reset
          MOV                       ; Set initial value of (for interrupt processing).
          W
          :

          Initialization of peripheral functions (interrupts), etc.

          :     WDTC,#WDT_CLR       ; Activate the watchdog timer.
INIT      MOV

                                    ; Select the watch prescaler as the count clock.
MAIN      MOV   WDTC,#WDT_CLR       ; Clear the watchdog timer.
          :

          User processing (interrupt processing may occur during this cycle)

          :
          JMP   MAIN                ; The loop must be executed in less than the
                                      minimum interval  time of the watchdog timer.

          ENDS

;-------- Main program ------------------------------------------------------------------------------------

          END
```

CHAPTER 7 8-BIT PWM TIMER

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

7.1 Overview of 8-bit PWM Timer

The 8-bit PWM timer can be selected to function as either an interval timer or PWM timer with 8-bit resolution. The interval timer function count-up in sync with either the 8/16-bit timer/counter (timer 1, timer 2) or one of three internal count clocks. Therefore, an 8-bit interval timer time can be set and the output can be used to generate variable frequency square waves. Also, the 8-bit PWM timer can be used as a D/A converter by connecting the PWM output to low pass filter.
There are two 8-bit PWM timer "channels", that perform the same function: 8-Bit PWM Timer 1 and 8-Bit PWM Timer 2.

■ **Interval Timer Function (Square Wave Output Function)**

The interval timer function generates repeated interrupts at variable time intervals.

Also, as the 8-bit PWM timer can invert the output level of the pin (PWM1, PWM2) each time an interrupt is generated, the 8-bit PWM timer can output a variable frequency square waves.

Each 8-bit PWM timer (1 and 2) can operate independently of the other.

- The interval timer can operate with a cycle among 1 and 28 times the count clock cycle.
- The count clock can be selected from four different clocks.

Table 7.1-1 "Interval Time and Square Wave Output Range" lists the range for the interval time and square wave output.

Table 7.1-1 Interval Time and Square Wave Output Range

	Count clock cycle		Interval time	Square wave output (Hz)
1	Internal count clock	1 t_{inst}	1 t_{inst} to 2^8 t_{inst}	$1/(2\ t_{inst})$ to $1/(2^9\ t_{inst})$
2		16 t_{inst}	$2^4\ t_{inst}$ to $2^{12}\ t_{inst}$	$1/(2^5\ t_{inst})$ to $1/(2^{13}\ t_{inst})$
3		64 t_{inst}	$2^6\ t_{inst}$ to $2^{14}\ t_{inst}$	$1/(2^7\ t_{inst}$ to $1/(2^{15}\ t_{inst})$
4	8-bit timer output cycle	$2^2\ t_{inst}$ to $2^{10}\ t_{inst}$	$2^2\ t_{inst}$ to $2^{18}\ t_{inst}$	$1/(2^3\ t_{inst})$ to $1/(2^{19}\ t_{inst})$
		$2^6\ t_{inst}$ to $2^{14}\ t_{inst}$	$2^6\ t_{inst}$ to $2^{22}\ t_{inst}$	$1/(2^7 t_{inst})$ to $1/(2^{23}\ t_{inst})$
		$2^{10}\ t_{inst}$ to $2^{18}\ t_{inst}$	$2^{10}\ t_{inst}$ to $2^{26}\ t_{inst}$	$1/(2^{11}\ t_{inst})$ to $1/(2^{27}\ t_{inst})$
		$1t_{ext}$ to $2^8\ t_{ext}$	1 t_{ext} to $2^{16}\ t_{ext}$	$1/(2\ t_{ext})$ to $1/(2^9\ t_{ext})$

t_{inst}: Instruction cycle (affected by clock mode, etc.)

t_{ext}: 8/16-bit timer/counter Timer 1 external clock period (8-bit PWM Timer 1 only)

8-bit timer output cycle: 8/16-bit timer/counter 8-bit timer output (Timer 1, Timer 2)

Note:

Calculation example for the interval time and square wave frequency

In this example, the main clock source oscillation (F_{CH}) is 4.2 MHz, the PWM compare register (COMR) value is set to "DD_H (221)" and the count clock cycle is set to 1 tinst. In this

case, the interval time and the frequency of the square wave output from the PWM1 or PWM2 pin (where the PWM timer operates continuously and the value of the COMR register is constant) are calculated as follows.

Assume that the main clock mode (SCS = 1) and its highest clock speed has been selected via the system clock control register (STCC: SCS = 1, CS = 11_B, CS0 = 11_B, 1 instruction cycle = $4/F_{CH}$).

Interval time = $(1\times 4/F_C) \times$ (COMR register value + 1)
= (4/4.2 MHz) × (221 + 1)
= 211 µs

Output frequency = $F_{CH}/(1\times 8 \times$ (COMR register value +1))
= 4.2 MHz/(8 × (221 + 1))
= 2.4 kHz

■ PWM Timer Function

The PWM timer function has 8-bit resolution and can control the "H" and "L" widths of one cycle.

8-Bit PWM Timers 1 and 2 can be operated independently of each other.

- As the resolution is 1/256, pulses can be output with duty ratios of between 0 and 99.6%.
- The cycle of the PWM wave can be selected from four types.
- The PWM timer can be used as a D/A converter by connecting the output to a low pass filter.

Table 7.1-2 "Available PWM Wave Cycle for PWM Timer Function" lists the available PWM wave cycles for the PWM timer function. Figure 7.1-1 "Example D/A Converter Configuration Using PWM Output and Low Pass Filter" shows an example D/A converter configuration.

Table 7.1-2 Available PWM Wave Cycle for PWM Timer Function

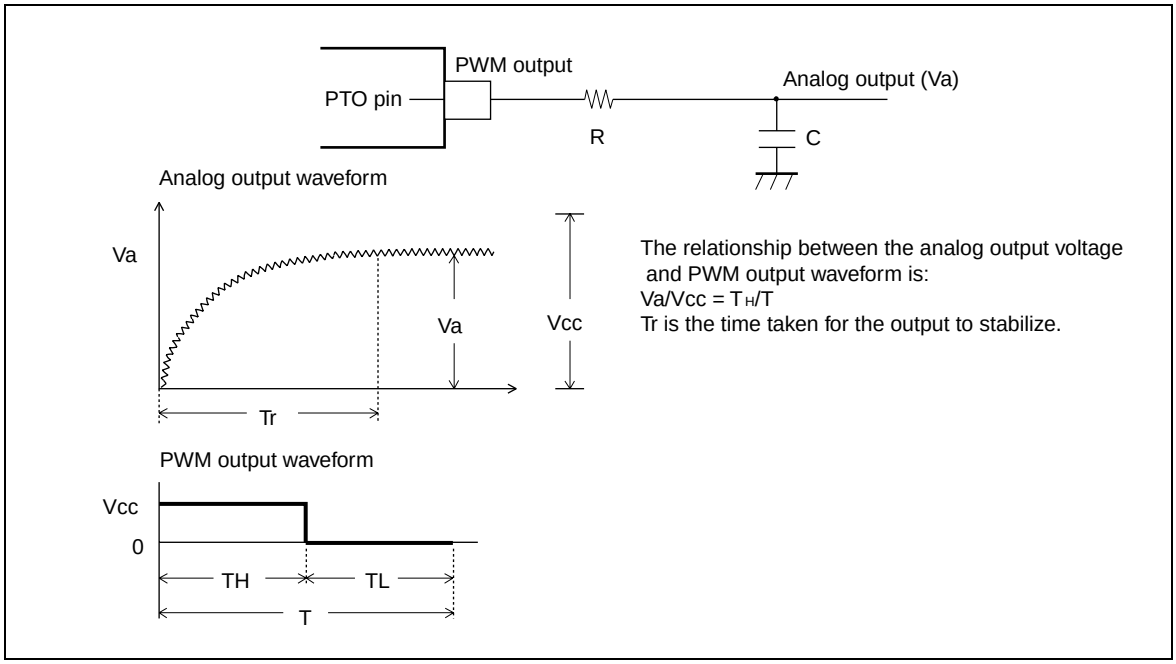
	1	2	3	4			
	Internal count clock			8-bit Timer Output Cycle Times			
Count clock cycle	1 t_{inst}	16 t_{inst}	64 t_{inst}	2^2 t_{inst} to 2^{10} t_{inst}	2^6 t_{inst} to 2^{14} t_{inst}	2^{10} t_{inst} to 2^{18} t_{inst}	1 t_{lext} to 2^8 t_{ext}
PWM wave cycle	2^8 t_{inst}	2^{12} t_{inst}	2^{14} t_{inst}	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}	2^8 t_{lext} to 2^{16} t_{ext}

tinst: Instruction cycle (affected by clock mode, etc.)

tlext: 8/16-bit timer/counter Timer 1 external clock period (8-bit PWM Timer 1 only)

8-bit timer output cycle: 8/16-bit timer/counter 8-bit timer output (Timer 1, Timer 2)

Figure 7.1-1 Example D/A Converter Configuration Using PWM Output and Low Pass Filter



Note:

Interrupt requests are not generated during operation of the PWM function.

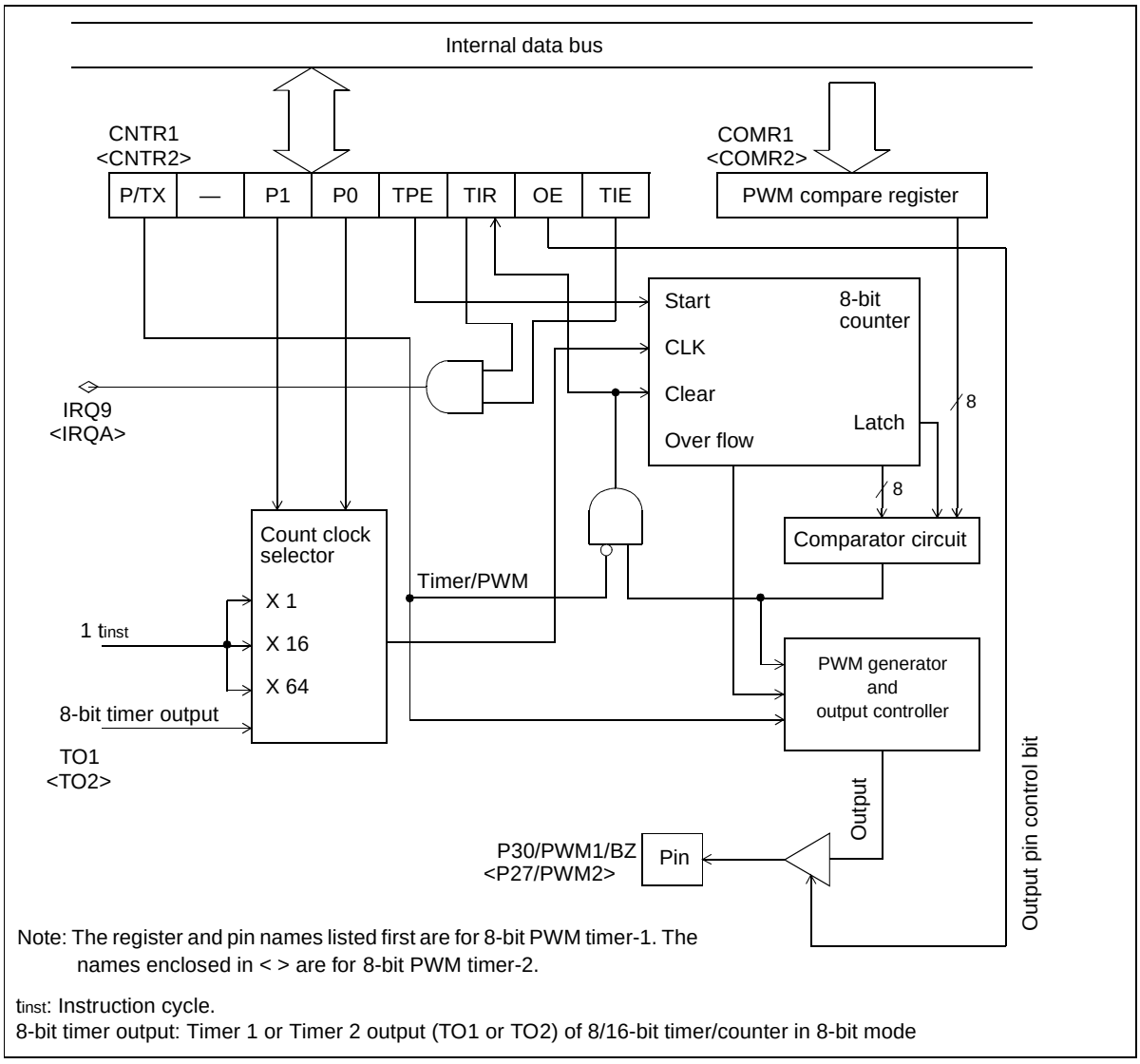
7.2 Block Diagram of 8-bit PWM Timer

The 8-bit PWM timer consists of the following six blocks:

- **Count clock selector**
- **8-bit counter**
- **Comparator circuit**
- **PWM generator and output controller**
- **PWM compare register (COMR)**
- **PWM control register (CNTR)**
- **8-bit PWM Timer-1 and 8-bit PWM Timer-2 have the same functions.**

■ Block Diagram of 8-bit PWM Timer

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



Note: The register and pin names listed first are for 8-bit PWM timer-1. The names enclosed in < > are for 8-bit PWM timer-2.

t_{inst}: Instruction cycle.
8-bit timer output: Timer 1 or Timer 2 output (TO1 or TO2) of 8/16-bit timer/counter in 8-bit mode

❍ **Count clock selector**

Selects a count-up clock for the 8-bit counter from the three internal count clocks and the 8-bit timer output cycle of the 8/16-bit timer/counter.

❍ **8-bit counter**

The 8-bit counter counts-up on the count clock selected by the count clock selector.

❍ **Comparator circuit**

The comparator circuit has a latch to hold the COMR register value. The circuit latches the COMR register value when the 8-bit counter value is "00H". The comparator circuit compares the 8-bit counter value with the latched COMR register value, and detects when a match occurs.

❍ **PWM generator and output controller**

When a match is detected during interval timer operation, an interrupt request is generated and, if the output pin control bit (CNTR: OE) is "1", the output controller inverts the output level of the PWM1(PWM2) pin. At the same time, the 8-bit counter is cleared.

When a match is detected during PWM timer operation, the PWM generator changes the output level of the PWM1(PWM2) pin from "H" to "L". The pin is set back to the "H" level when the next overflow occurs on the 8-bit counter.

❍ **COMR register**

The COMR register is used to set the value that is compared with the value of the 8-bit counter.

❍ **CNTR register**

The CNTR register is used to select the operating mode, enable or disable operation, set the count clock, control interrupts, and check the PWM status.

Setting the operation to PWM timer mode (P/TX = "0" disables clearing of the 8-bit counter and generation of interrupt requests (IRQ9, IRQA) when the comparator circuit detects a match.

7.3 Structure of 8-bit PWM Timer 1

This section describes the pin, pin block diagram, register source, and interrupts of the 8-bit PWM timer.

■ 8-bit PWM Timer 1 Pin

The 8-bit PWM timer uses the P30/PWM1/BZ pin. This pin can function as a output-only port (P30), or Buzzer output/BZ, or as the interval timer or PWM timer output (PWM1).

PWM1:

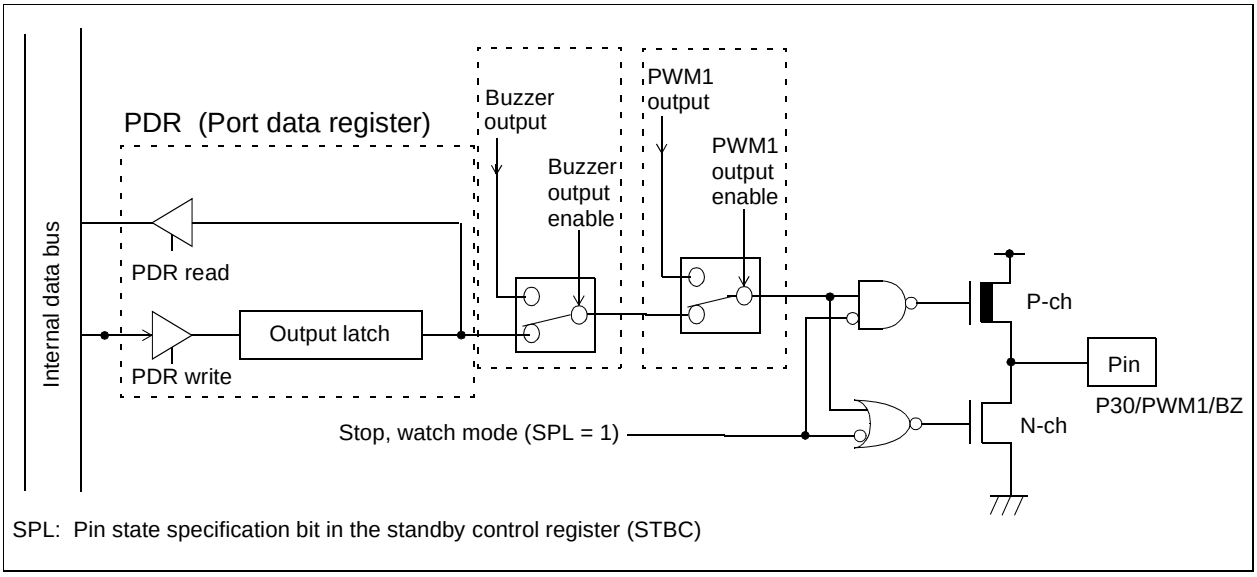
When the interval timer function is selected, the square waves are output to this pin.

When the PWM timer function is selected, the pin outputs the PWM wave.

Setting the output pin control bit (CNTR1: OE) to "1" makes Pin P30/PWM1/BZ the output-only pin for 8-Bit PWM Timer 1. Once this has been done, the pin performs its PWM1 function regardless of the state of the port data register output latch data (PDR3: Bit 0) and Buzzer output.

■ Block Diagram of 8-bit PWM Timer Pin

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



■ 8-bit PWM Timer 1 Registers

Figure 7.3-2 8-bit PWM1 Timer Registers

CNTR1 (PWM1 control register)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
001EH	P/TX	—	P1	P0	TPE	TIR	OE	TIE	0-000000B
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

COMR1 (PWM1 compare register)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
001FH									XXXXXXXXB
	W	W	W	W	W	W	W	W	

R/W: Readable and writable
W : Write-only
— : Unused
X : Indeterminate

Check:

As the PWM1 compare register (COMR1) is write-only, the bit manipulation instructions can be used.

■ 8-bit PWM Timer 1 Interrupt Source

IRQ9:

For the interval timer function, the 8-bit PWM timer generates an interrupt request if interrupt request output is enabled (CNTR: TIE = "1") when the counter value matches the value set in the COMR register. (No interrupt requests are generated when the PWM function is operating.)

7.3.1 PWM1 Control Register (CNTR1)

The PWM1 control register (CNTR1) is used to select the operating mode of the 8-bit PWM timer (interval timer operation or PWM timer operation), enable or disable operation, select the count clock, control interrupts, and check the state of the 8-bit PWM timer.

■ PWM1 Control Register (CNTR1)

Figure 7.3-3 PWM1 Control Register (CNTR 1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
001EH	P/TX	—	P1	P0	TPE	TIR	OE	TIE	0-000000B
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

TIE	Interrupt request enable bit
0	Disables interrupt request output.
1	Enables interrupt request output.

OE	Output pin control bit
0	Functions as a general-purpose port (P30) or Buzzer output.
1	Functions as the interval timer/PWM timer output pin (PWM1).

TIR	Interrupt request flag bit		
	Read		Write
	Interval timer function	PWM timer function	
0	Counter value and set value do not match.	No change	Clears this bit.
1	Counter value and set value match.		No effect. The bit does not change.

TPE	Counter operation enable bit
0	Stops count operation.
1	Starts count operation.

P1	P0	Clock selection bits		
0	0	Internal count clock	1 t_{inst}	*1
0	1		16 t_{inst}	*1
1	0		64 t_{inst}	*1
1	1	Timer 1 output		*2

P/TX	Operating mode selection bit
0	Operates as an interval timer.
1	Operates as a PWM timer.

*1: t_{inst} : Instruction cycle
*2: The “Timer 1 output” referred to here is the timer 1 output cycle when the 8/16-bit timer/counter is used in the 8-bit mode

R/W : Readable and writable
R : Read-only
W : Write-only
— : Unused
X : Indeterminate
□ : Initial value

Table 7.3-1 PWM 1 Control Register (CNTR1) bit

Bit		Function
Bit 7	P/TX: Operating mode selection bit	• This bit switches between the interval timer function (P/TX = "0") and PWM timer function (P/TX = "1"). **Check:** Write to this bit when the counter operation is stopped (TPE = "0"), interrupts are disabled (TIE = "0"), and the interrupt request flag bit is cleared (TIR = "0").
Bit 6	Unused bit	• The read value is indeterminate. • Writing to this bit has no effect on the operation.
Bit 5 Bit 4	P1, P0: Clock selection bit	• These bits select the count clock for the interval timer function and PWM timer function. • These bits can select the count clock from three internal count clocks or the output cycle of the timer 1. • If the timer 1 output is selected, operate the 8/16-bit timer/counter in 8-bit mode. **Check:** Do not change P1 and P0 when the counter is operating (TPE = "1").
Bit 3	TPE: Counter operation enable bit	• This bit activates or stops operation of the PWM timer function and interval timer function. • Writing "1" to this bit starts the count operation. Writing "0" to this bit stops the count and clears the counter to "00_H".
Bit 2	TIR: Interrupt request flag bit	• For the interval timer function: This bit is set to "1" when the counter and PWM 1 compare register (COMR1) values match. An interrupt request is output to the CPU when both this bit and the interrupt request enable bit (TIE) are "1". • For the PWM timer function: Interrupt requests are not generated. • Writing "0" clears this bit. Writing "1" has no effect and does not change the bit value.
Bit 1	OE: Output pin control bit	• The P30/PWM1/BZ pin functions as a general-purpose port (P30) or Buzzer output when this bit is set to "0" and a dedicated pin (PWM1) when this bit is set to "1". • The PWM1 pin outputs a square wave when the interval timer function is selected and a PWM waveform when the PWM timer function is selected.
Bit 0	TIE: Interrupt request enable bit	This bit enables or disables and an interrupt request output to the CPU. An interrupt request is output when both this bit and the interrupt request flag bit (TIR) are "1".

7.3.2 PWM 1 Compare Register (COMR1)

The PWM 1 compare register (COMR1) sets the interval time for the interval timer function. The register value sets the "H" width of the pulse for the PWM timer function.

■ PWM 1 Compare Register (COMR1)

Figure 7.3-4 "PWM 1 Compare Register (COMR1)" shows the bit structure of the PWM1 compare register.

As the register is write-only, bit manipulation instructions cannot be used.

Figure 7.3-4 PWM 1 Compare Register (COMR1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
001F$_H$									XXXXXXXX$_B$
	W	W	W	W	W	W	W	W	

W: Write-only
X: Indeterminate

❍ **Interval timer operation**

This register is used to set the value to be compared with the counter value. The register specifies the interval time.

The counter is cleared when the counter value matches the value set in this register, and the interrupt request flag bit is set to "1" (CNTR 1: TIR = "1").

If data is written to the COMR1 register during counter operation, the new value applies from the next cycle (after the next match is detected).

Note:

The COMR1 setting for interval timer operation can be calculated using the following formula.

The instruction cycle time is affected by the clock mode, and the speed-shift selection.

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

❍ **PWM timer operation**

This register is used to set the value to be compared with the counter value. The register therefore sets the "H" width of the pulse.

The PWM1 pin outputs an "H" level until the counter value matches the value set in this register. From the match until the counter value overflows, the PWM1 pin outputs an "L" level.

If data is written to the COMR 1 register during counter operation, the new value applies from the next cycle (after the next overflow).

Note:

In PWM timer operation, the COMR1 setting and the PWM cycle time can be calculated using the following formulas. (The instruction cycle time is affected by the clock mode, and the speed-shift selection.)

COMR1 register value = duty ratio (%) × 256

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

7.4 Structure of 8-bit PWM Timer 2

This section describes the pin, pin block diagram, register source, and interrupts of the 8-bit PWM timer.

■ 8-bit PWM Timer 2 Pin

The 8-bit PWM timer 2 uses the P27/PWM2 pin. This pin can function either as a general-purpose I/O port (P27) or as the interval timer or PWM timer output (N-ch open-drain output)(PWM2).

PWM2:

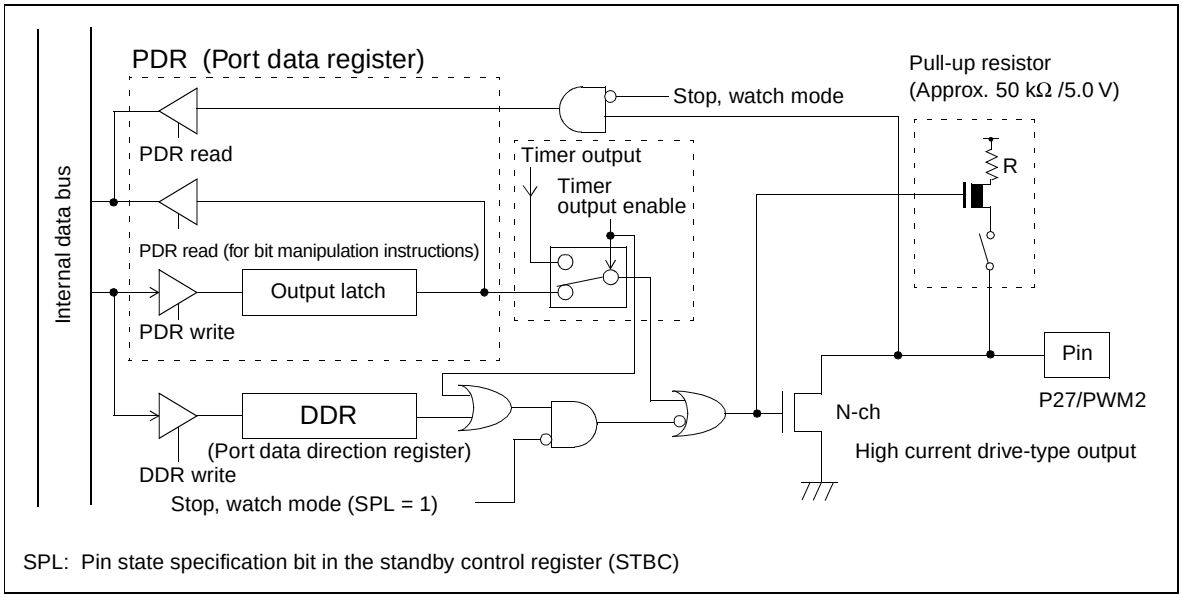
When the interval timer function is selected, the square waves are output to this pin.

When the PWM timer function is selected, the pin outputs the PWM wave.

Setting the P27/PWM2 pin as a dedicated pin in the output pin control bit (CNTR 2: OE = "1") automatically sets the pin as an output pin, regardless of the port data direction register (DDR2: bit 7) value, and sets the pin to function as the PWM2 pin.

■ Block Diagram of 8-bit PWM Timer 2 Pin

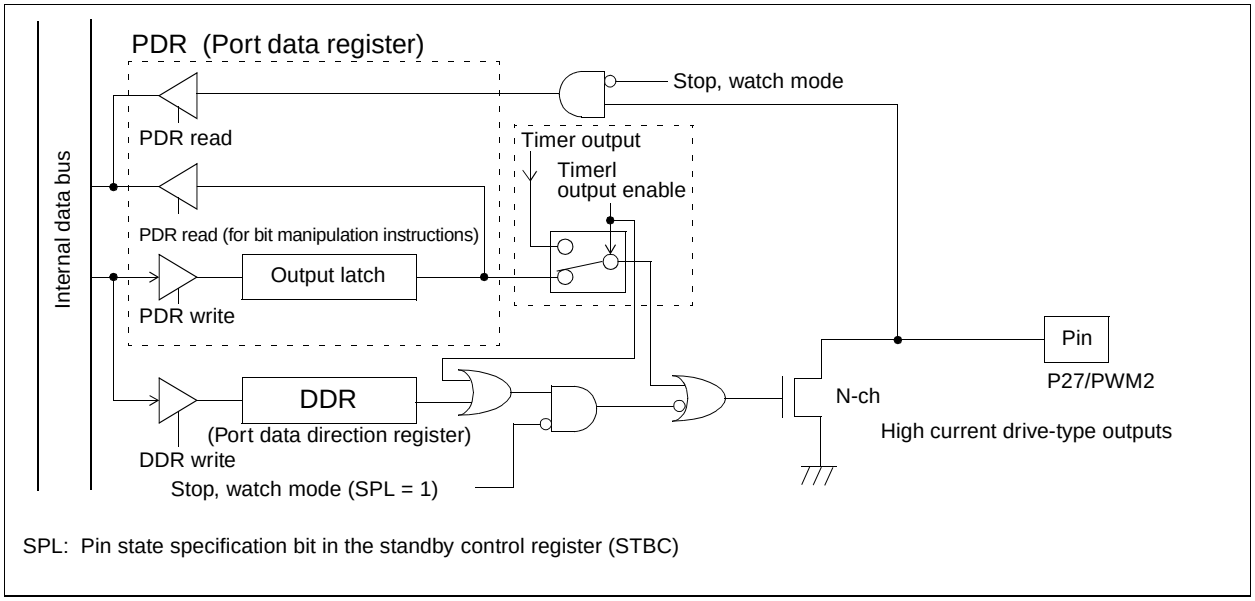
Figure 7.4-1 Block Diagram of 8-bit PWM Timer 2 Pin for MB89983



Note:

Pins with a pull-up resistor (optional) go to the "H" level during a reset or in stop and watch modes.

Figure 7.4-2 Block Diagram of 8-bit PWM Timer 2 Pin for MB89P985 and MB89PV980



■ 8-bit PWM Timer 2 Registers

Figure 7.4-3 8-bit PWM Timer 2 Registers

CNTR2 (PWM2 control register)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0020H	P/TX	—	P1	P0	TPE	TIR	OE	TIE	0-000000B
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

COMR2 (PWM2 compare register)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0021H									XXXXXXXXB
	W	W	W	W	W	W	W	W	

R/W: Readable and writable
W : Write-only
— : Unused
X : Indeterminate

Check:

As the PWM2 compare register (COMR2) is write-only, the bit manipulation instructions cannot be used.

■ 8-bit PWM Timer 2 Interrupt Source

IRQA:

For the interval timer function, the 8-bit PWM timer generates an interrupt request if interrupt request output is enabled (CNTR 2: TIE = "1") when the counter value matches the value set in the COMR2 register. (No interrupt requests are generated when the PWM function is operating.)

7.4.1 PWM 2 Control Register (CNTR2)

The PWM 2 control register (CNTR2) is used to select the operating mode of the 8-bit PWM timer (interval timer operation or PWM timer operation), enable or disable operation, select the count clock, control interrupts, and check the state of the 8-bit PWM timer.

■ PWM 2 Control Register (CNTR2)

Figure 7.4-4 PWM 2 Control Register (CNTR2)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0020H	P/TX	—	P1	P0	TPE	TIR	OE	TIE	0-000000B
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

TIE	Interrupt request enable bit
0	Disables interrupt request output.
1	Enables interrupt request output.

OE	Output pin control bit
0	Function as a general-purpose port (P27).
1	Function as the interval timer/PWM timer output pin (PWM2).

TIR	Interrupt request flag bit		
	Read		Write
	Interval timer function	PWM timer function	
0	Counter value and set value do not match.	No change	Clears this bit.
1	Counter value and set value match.		No effect. The bit does not change.

TPE	Counter operation enable bit
0	Stops count operation.
1	Starts count operation.

P1	P0	Clock selection bits		
0	0	Internal count clock	1 t_{inst}	*1
0	1		16 t_{inst}	*1
1	0		64 t_{inst}	*1
1	1	Timer 2 output		*2

P/TX	Operating mode selection bit
0	Operates as an interval timer.
1	Operates as a PWM timer.

*1: t_{inst} : Instruction cycle
*2: The “Timer 1 output” referred to here is the timer 1 output cycle when the 8/16-bit timer/counter is used in the 8-bit mode

R/W : Readable and writable
R : Read-only
W : Write-only
— : Unused
X : Indeterminate
: Initial value

Table 7.4-1 PWM 2 Control Register (CNTR2) Bits

Bit		Function
Bit 7	P/TX: Operating mode selection bit	• This bit switches between the interval timer function (P/TX = "0") and PWM timer function (P/TX = "1") **Check:** Write to this bit when the counter operation is stopped (TPE = "0"), interrupts are disabled (TIE = "0"), and the interrupt request flag bit is cleared (TIR = "0").
Bit 6	Unused bit	• The read value is indeterminate. • Writing to this bit has no effect on the operation.
Bit 5 Bit 4	P1, P0: Clock selection bit	• These bits select the count clock for the interval timer function and PWM timer function. • Selects either one of three internal count clock, or the Timer 2 output cycle. • When you selects the Timer 2 output, operate the 8/16-bit timer/counter in its 8-bit mode. **Check:** Do not change P1 and P0 when the counter is operating (TPE = "1").
Bit 3	TPE: Counter operation enable bit	• This bit activates or stops operation of the PWM timer function and interval timer function. • Writing "1" to this bit starts the count operation. Writing "0" to this bit stops the count and clears the counter to "00_H".
Bit 2	TIR: Interrupt request flag bit	• For the interval timer function: This bit is set to "1" when the counter and PWM 2 compare register (COMR2) values match. An interrupt request is output to the CPU when both this bit and the interrupt request enable bit (TIE) are "1". • For the PWM timer function: Interrupt requests are not generated. • Writing "0" clears this bit. Writing "1" has no effect and does not change the bit value.
Bit 1	OE: Output pin control bit	• The P27/PWM2 pin functions as a general-purpose port (P27) when this bit is set to "0" and a dedicated pin (PWM2) when this bit is set to "1". • The PWM2 pin outputs a square wave when the interval timer function is selected and a PWM waveform when the PWM timer function is selected.
Bit 0	TIE: Interrupt request enable bit	This bit enables or disables and an interrupt request output to the CPU. An interrupt request is output when both this bit and the interrupt request flag bit (TIR) are "1".

7.4.2 PWM 2 Compare Register (COMR2)

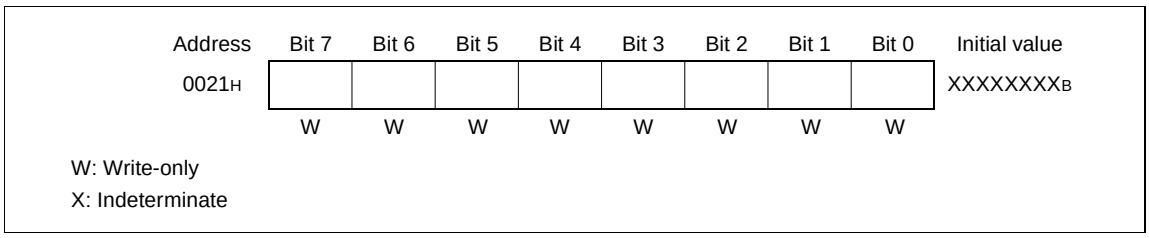
The PWM 2 compare register (COMR2) sets the interval time for the interval timer function. The register value sets the "H" width of the pulse for the PWM timer function.

■ PWM 2 Compare Register (COMR2)

Figure 7.4-5 "PWM2 Compare Register (COMR2)" shows the bit structure of the PWM 2 compare register.

As the register is write-only, bit manipulation instructions cannot be used.

Figure 7.4-5 PWM2 Compare Register (COMR2)



❍ Interval timer operation

This register is used to set the value to be compared with the counter value. The register specifies the interval time.

The counter is cleared when the counter value matches the value set in this register, and the interrupt request flag bit is set to "1" (CNTR2: TIR = "1").

If data is written to the COMR2 register during counter operation, the new value applies from the next cycle (after the next match is detected).

Note:

The COMR2 setting for interval timer operation can be calculated using the following formula.
(The instruction cycle time is affected by the clock mode, and the speed-shift selection.)

COMR2 register value = interval time/(count clock cycle × instruction cycle) - 1

❍ PWM timer operation

This register is used to set the value to be compared with the counter value. The register therefore sets the "H" width of the pulse.

The PWM2 pin outputs an "H" level until the counter value matches the value set in this register. From the match until the counter value overflows, the PWM2 pin outputs an "L" level.

If data is written to the COMR2 register during counter operation, the new value applies from the next cycle (after the next overflow).

Note:

In PWM timer operation, the COMR2 setting and the PWM cycle time can be calculated using the following formulas. (The instruction cycle time is affected by the clock mode, and the speed-shift selection.)

COMR2 register value = duty ratio (%) × 256

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

7.5 8-bit PWM Timer Interrupts

The 8-bit PWM timer can generate an interrupt request when a match is detected between the counter value and PWM compare register value for the interval timer function. Interrupt requests are not generated for the PWM timer function.
8-bit PWM timer-1 generates the IRQ9 as an interrupt request and 8-bit PWM timer-2 generates the IRQA as an interrupt request.

■ Interrupts for Interval Timer Function

The counter value is counted-up from "00_H" on the selected count clock. When the counter value matches the PWM compare register (COMR) value, the interrupt request flag bit (CNTR: TIR) is set to "1".

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

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

Note:

The TIR bit is not set if the counter is stopped (CNTR: TPE = "0") at the same time as the counter value matches the COMR register value.

An interrupt request is generated immediately if the TIR bit is "1" when the TIE bit is changed from disabled to enabled ("0" --> "1").

■ Registers and Vector Tables for 8-bit PWM Timer Interrupts

Table 7.5-1 Registers and Vector Tables for 8-bit PWM Timer Interrupts

	Interrupt	Interrupt level setting register			Vector table address	
		Register	Setting bits		Upper	Lower
8-bit PWM Timer 1	IRQ9	ILR3 ($007E_H$)	L91 (Bit3)	L90 (Bit2)	$FFF8_H$	$FFF9_H$
8-bit PWM Timer 2	IRQA	ILR3 ($007E_H$)	LA1 (Bit5)	LA0 (Bit4)	$FFF6_H$	$FFF7_H$

Reference:

See Section 3.4.2 "Interrupt Processing" for details on the interrupt operation.

7.6 Operation of Interval Timer Function

This section describes the operation of the timer interval timer function of the 8-bit PWM timer.

■ Operation of Interval Timer Function

Figure 7.6-1 "Interval Timer Function Settings" shows the settings required to operate as an interval timer function.

Figure 7.6-1 Interval Timer Function Settings

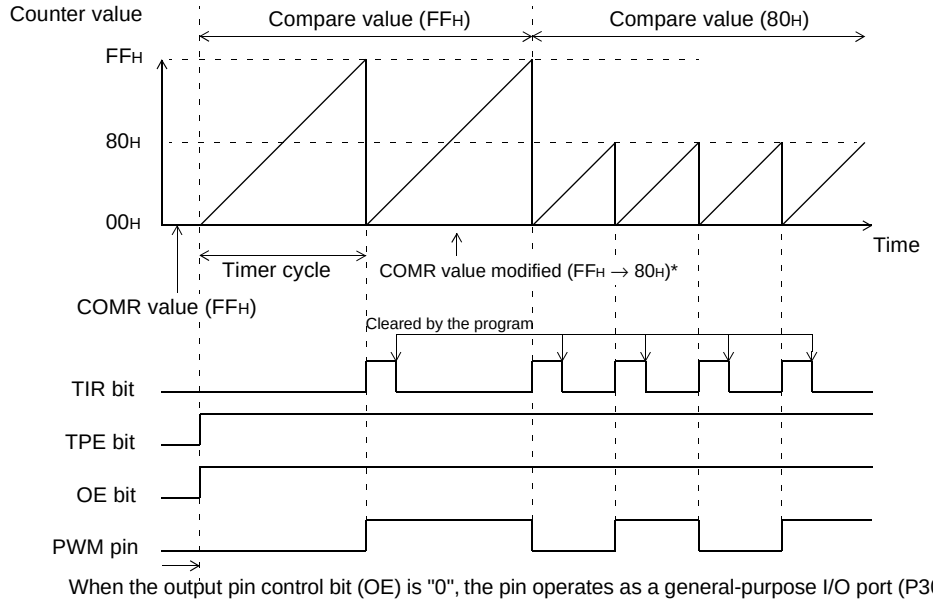
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CNTR	P/TX	—	P1	P0	TPE	TIR	OE	TIE
	0		◎	◎	1	◎	◎	◎
COMR	Sets interval time (compare value).							

◎: Used bit
1 : Set "1".
0 : Set "0".

On activation, the counter starts counting-up from "00_H" on the rising edge of the selected count clock. When the counter value matches the value set in the COMR register (compare value), the PWM timer inverts the level of the output pin (PWM1, PWM2) on the next rising edge of the count clock, clears the counter, sets the interrupt request flag bit (CNTR: TIR = "1"), and restarts counting from "00_H".

Figure 7.6-2 "Operation of 8-bit PWM Timer" shows the operation of the 8-bit PWM timer.

Figure 7.6-2 Operation of 8-bit PWM Timer



When the output pin control bit (OE) is "0", the pin operates as a general-purpose I/O port (P30 or P27).

*: If the PWM compare register (COMR) value is modified during counter operation, the new value is used from the next cycle.

Check:

Do not change the count clock cycle (CNTR: P1, P0) during operation of the interval timer function (CNTR: TPE = "1").

Notes:

Setting the COMR register value to "00_H" causes the PWM pin output to be inverted with the cycle of the selected count clock.

When the counter is stopped (CNTR: TPE = "0") while the interval timer function is selected, the PWM pin outputs an "L" level.

7.7 Operation of PWM Timer Function

This section describes the operation of the PWM timer function of the 8-bit PWM timer.

■ Operation of PWM Timer Function

Figure 7.7-1 "PWM Timer Function Settings" shows the settings required to operate as the PWM timer function.

Figure 7.7-1 PWM Timer Function Settings

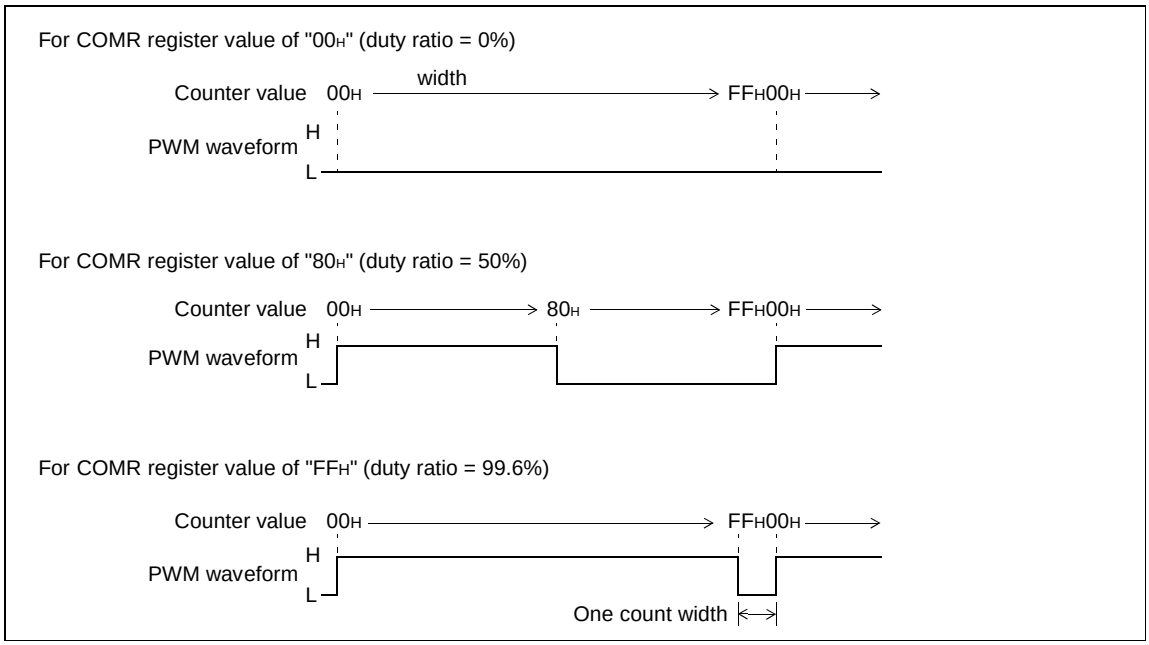
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CNTR	P/TX	—	P1	P0	TPE	TIR	OE	TIE
	1		◎	◎	1	×	1	×
COMR	Sets “H” width of pulse (compare value).							

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

On activation, the counter starts counting-up from "00_H" on the rising edge of the selected count clock. The PWM pin (PWM1, PWM2) outputs (PWM waveform) an "H" level until the counter value matches the value set in the COMR register. From the match until the counter value overflows (FF_H --> 00_H), the PTO pin outputs an"L" level.

Figure 7.7-2 "Example of PWM Waveform Output (PTO Pin)" shows the PWM waveforms output from the PTO pin.

Figure 7.7-2 Example of PWM Waveform Output (PTO Pin)



Check:

Do not change the count clock cycle (CNTR: P1, P0) during operation of the PWM timer function (CNTR: TPE = "1").

Note:

When the PWM timer function is selected, the PWM pin maintains its existing level when the counter is stopped (CNTR: TPE = "0").

7.8 States in Each Mode during 8-bit PWM Timer Operation

This section describes the operation of the 8-bit PWM timer when the device changes to sleep or stop mode or an operation halt request occurs during operation.

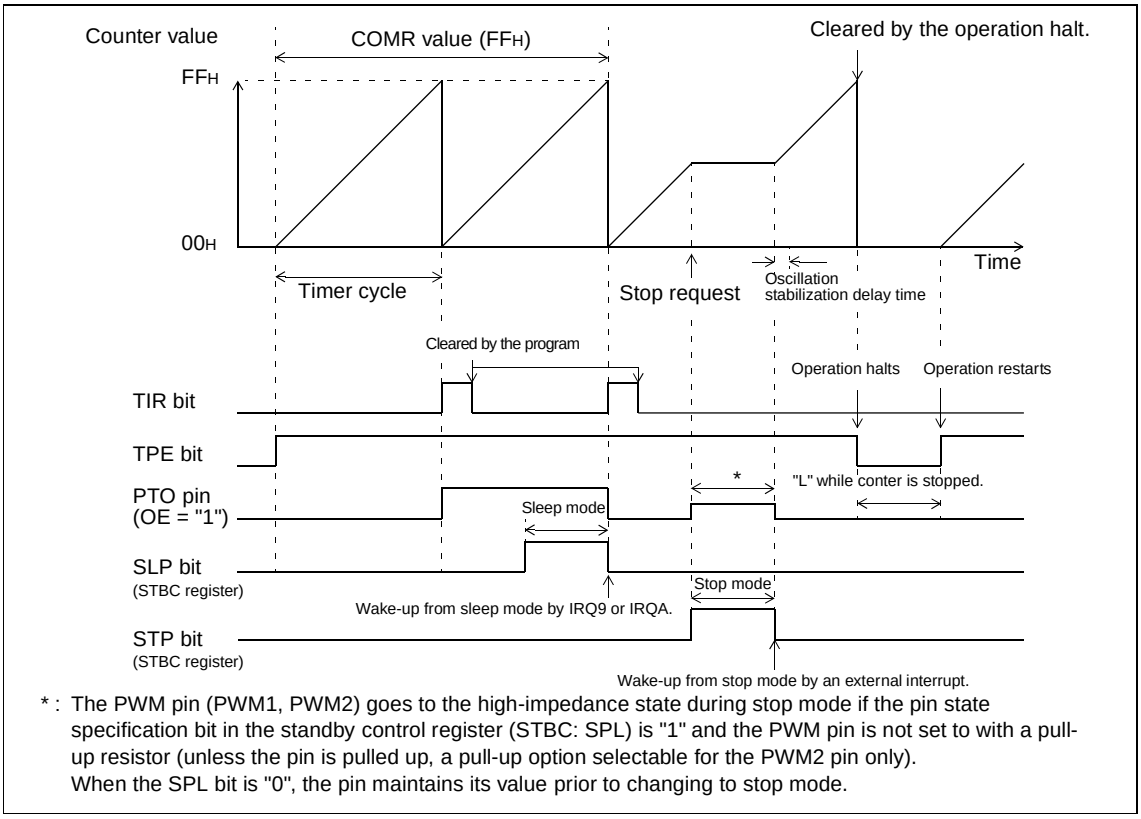
■ Operation during Standby Mode or Operation Halt

Figure 7.8-1 "Counter Operation during Standby Modes or Operation Halt (For Interval Timer Function)" and 7.8-2 "Operation during Standby Modes or Operation Halt (For PWM Timer Function)" show the counter value states when the device changes to sleep or stop mode, or an operation halt request occurs, during operation of the interval timer function or PWM timer function.

The counter halts and maintains its current value when the device changes to stop mode. Operation starts again from the stored counter value after wake-up from stop mode by an external interrupt. Therefore, the first interval time or PWM wave cycle does not match the set value. Always initialize the 8-bit PWM timer after wake-up from stop mode.

❍ For interval timer function

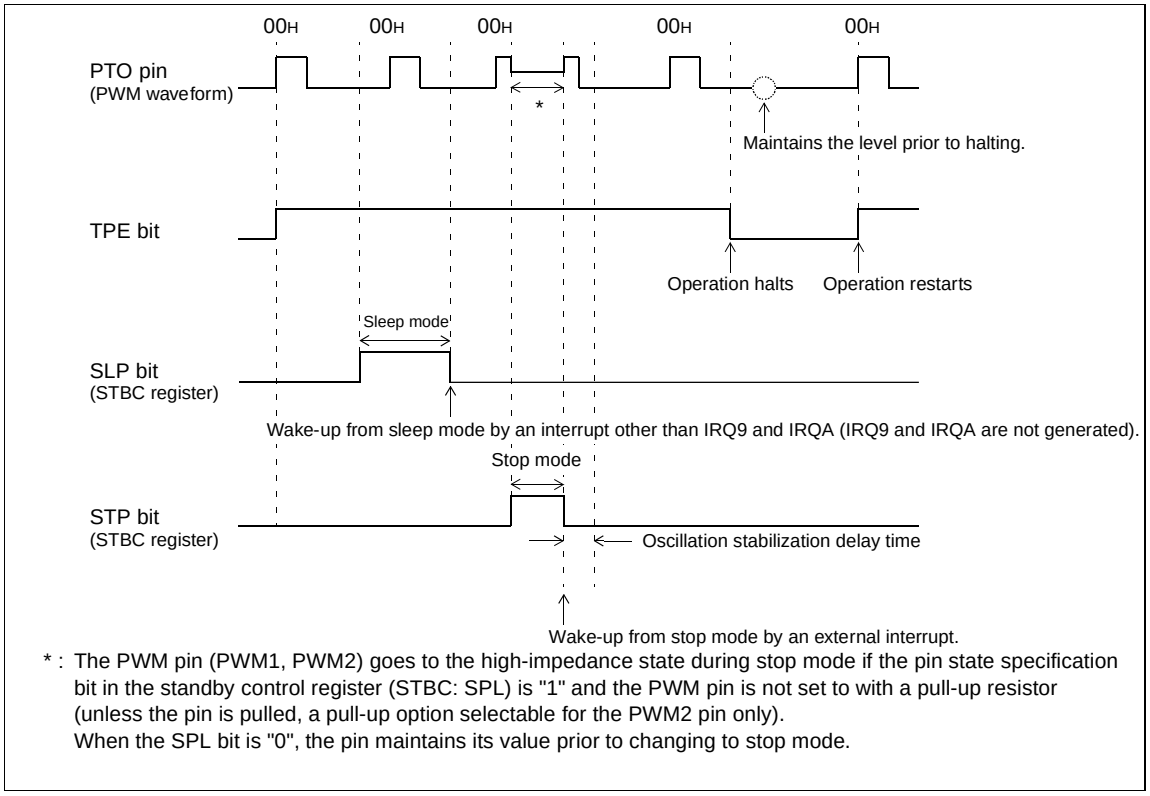
Figure 7.8-1 Counter Operation during Standby Modes or Operation Halt (For Interval Timer Function)



* : The PWM pin (PWM1, PWM2) goes to the high-impedance state during stop mode if the pin state specification bit in the standby control register (STBC: SPL) is "1" and the PWM pin is not set to with a pull-up resistor (unless the pin is pulled up, a pull-up option selectable for the PWM2 pin only).
When the SPL bit is "0", the pin maintains its value prior to changing to stop mode.

❍ **For PWM timer function**

Figure 7.8-2 Operation during Standby Modes or Operation Halt (For PWM Timer Function)



* : The PWM pin (PWM1, PWM2) goes to the high-impedance state during stop mode if the pin state specification bit in the standby control register (STBC: SPL) is "1" and the PWM pin is not set to with a pull-up resistor (unless the pin is pulled, a pull-up option selectable for the PWM2 pin only).
When the SPL bit is "0", the pin maintains its value prior to changing to stop mode.

7.9 Notes on Using 8-bit PWM Timer

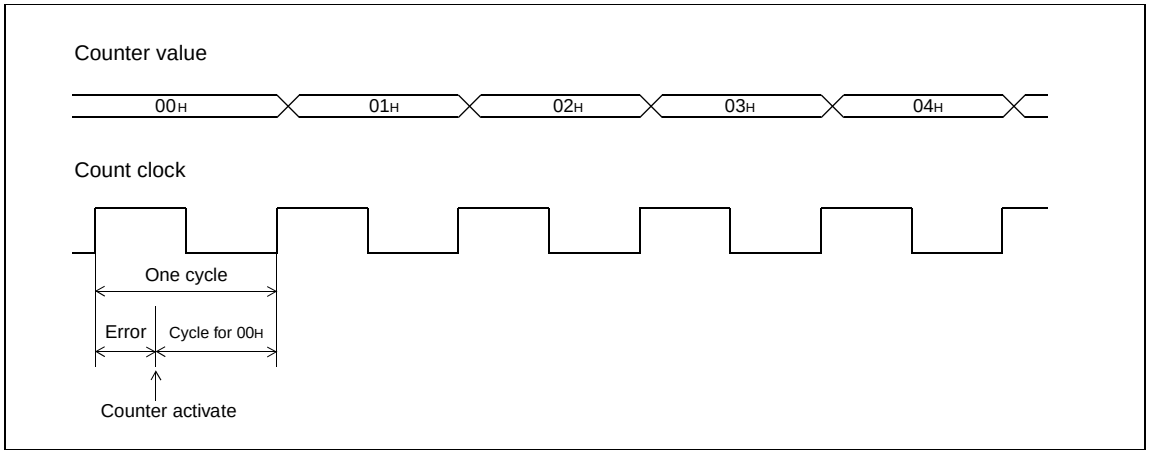
This section lists points to note when using the 8-bit PWM timer.

■ Notes on Using 8-bit PWM Timer

❍ Error

Activating the counter by program is not synchronized with the start of counting-up using the selected count clock. Therefore, the time from activating the counter until a match with the PWM compare register (COMR) is detected may be shorter than the theoretical time by a maximum of one cycle of the count clock. Figure 7.9-1 "Error on Starting Counter Operation" shows the error that occurs on starting counter operation.

Figure 7.9-1 Error on Starting Counter Operation



❍ Notes on setting by program

- Do not change the count clock cycle (CNTR: P1, P0) when the interval timer function or PWM timer function is operating (CNTR: TPE = "1").
- Stop the counter (CNTR: TPE = "0"), disable interrupts (TIE = "0"), and clear the interrupt request flag (TIR = "0") before switching between the interval timer function and PWM timer function (CNTR: P/TX).
- Interrupt processing cannot return if the interrupt request flag bit (CNTR: TIR) is "1" and the interrupt request enable bit is enabled (CNTR: TIE = "1"). Always clear the TIR bit.
- The TIR bit is not set if the counter is disabled (TPE = "0") at the same time as the counter and COMR register values match.

❍ Notes when using 8/16-bit timer/counter output

- When selecting timer 1 output for 8-bit PWM timer 1 and operating the 8/16-bit timer/counter in 16-bit ode, always select internal count clock for 8-bit PWM timer 2. Timer 2 output cannot be selected.

7.10 Program Example for 8-bit PWM Timer

This section gives program examples for the 8-bit PWM timer.

■ Program Example for Interval Timer Function

❍ Processing description

- Generates repeated interval timer interrupts at 5 ms intervals.
- Outputs a square wave to the PWM1 pin that inverts after each interval time.
- With a main clock master oscillation F_{CH} of 4.2 MHz, and the highest speed clock selected by the speed-shift function (1 instruction cycle time = $4/F_{CH}$), the COMR register is set for an interval time of approximately 5 ms. (AN internal clock period of 64 tinst is selected as the count clock.) The COMR register setting is calculated as follows:

  COMR register value = 5 ms/(64 × 4/4.2 MHz) - 1 = 81.0 (051H)

❍ Coding example

```
CNTR1   EQU     001EH           ; Address of the PWM 1 control register
COMR1   EQU     001FH           ; Address of the PWM 1 compare register

TPE     EQU     CNTR1:3         ; Define the counter operation enable bit.
TIR     EQU     CNTR1:2         ; Define the interrupt request flag bit.

ILR3    EQU     007EH           ; Address of the interrupt level setting
                                  register 3

INT_V   DSEG    ABS             ; [DATA SEGMENT]
        ORG     0FFE8H
IRQ9    DW      WARI            ; Set reset vector.
INT_V   ENDS

;-------- Main program ------------------------------------------------------------------------------

        CSEG                    ; [CODE SEGMENT]
                                ; Stack pointer (SP) etc. are already
                                  initialized.
        :
        CLRI                    ; Disable interrupts.
        CLRB    TPE             ; Stop counter operation.
        MOV     ILR3,#11110111B ; Set interrupt level (level 1).
        MOV     COMR1,#051H     ; Value compared with the counter value
                                  (interval time)
        MOV     CNTR1,#00101011B ; Operate interval timer, select 64 tinst, start
                                  counter operation, clear interrupt request
                                  flag, enable TO pin output, enable interrupt
                                  request output.
        SETI                    ; Enable interrupts.
        :
```

```
;-------- Interrupt program ------------------------------------------------------------------------------------
--
WARI    CLRB    TIR                 ; Clear interrupt request flag.
        PUSHW   A
        XCHW    A,T                 ; Save A and T.
        PUSHW   A
        :
        User processing
        :
        POPW    A
        XCHW    A, T                ; Restore A and T.
        POPW    A
        RETI
        ENDS
;-----------------------------------------------------------------------------------------------------------------

        END
```

■ Program Example for PWM Timer Function

❍ Processing description

- Generates a PWM wave with a duty ratio of 50%. Then, changes the duty ratio to 25%.
- Does not generate interrupts.
- For a 4.2 MHz source oscillation, (F_{CH}), and the highest speed clock selected by the speedshift function (1 instruction cycle time = $4/F_{CH}$), selecting the interval 16 t_{inst} count clock gives a PWM wave cycle of $16 \times 4/4.2\ \text{MHz} \times 256 \fallingdotseq 3.901$ ms.
- The following shows the COMR register value required for a duty ratio of 50%:

  COMR register value = $50/100 \times 256 = 128\ (080_H)$

❍ Coding example

```
CNTR1   EQU     001EH               ; Address of the PWM 1 control register
COMR1   EQU     001FH               ; Address of the PWM 1 compare register

TPE     EQU     CNTR1:3             ; Define the counter operation enable bit.

;-------- Main program ----------------------------------------------------------------------------------------

        CSEG                        ; [CODE SEGMENT]
        :
        CLRB    TPE                 ; Stop counter operation.
        MOV     COMR1,#80H          ; Set "H" width of pulse. Duty ratio = 50%
        MOV     CNTR1,#100110101B   ; Operate PWM timer, select 16 tinst, start
                                      counter operation, clear interrupt request
                                      flag, enable TO pin output, and disable
                                      interrupt request output.
```

```
        :
        :
        MOV     COMR1,#40H      ; Change the duty ratio to 25% (effective
                                  from the next PWM wave cycle).
        :
        ENDS
;-----------------------------------------------------------------------------------------------------------------------

        END
```

CHAPTER 8 8/16-BIT TIMER/COUNTER

This chapter describes the functions and operation of the 8/16-bit Timer/Counter.

8.1 Overview of 8/16 -bit Timer/Counter

The 8/16-bit timer/counter is made up of two 8-bit timers (Timer 1 and Timer 2) that can be used separately (8-bit mode) or connected in cascade to form one counter (16-bit mode).
Timer 1 can be selected to function as either an interval timer or a counter. The interval timer function counts up in sync with one of three interval count clocks. The counter function counts up by a clock input to the external pin.The output can be used to generate variable frequency square wave output.
Timer 2 functions as an interval timer clocked by one of three internal count clocks. In the 16 bit mode, it is connected in series with Timer 1.

■ Interval Timer Function

The interval timer function generates repeated interrupts at variable intervals. Also, as the 8/16-bit timer/counter can invert the output level of the pin (TO pin) each time an interval time is generated, the 8/16-bit timer/counter can output variable frequency square waves (Timer 1 in 8 bit mode, or 16 bit mode).

- In 8-bit mode, Timer 1 and Timer 2 operate as two independent interval timers, each of which can count time intervals ranging from the clock period (the time of one clock cycle) to 2^8 times the clock period.
- In 16-bit mode, the two counters form a single 16-bit timer, with Timer 1 containing the LSBs and Timer 2 the MSBs. The interval timer can operate with a cycle among 1 and 2^{16} times the internal count clock cycle.
- The count clock can be selected from three different internal clocks. (An external clock can be selected for Timer 1, but it will then function as a counter).
- Timer 1 and Timer 2 outputs can be used as the count clocks for 8-Bit PWM Timer 1 and 8-Bit PWM Timer 2, respectively.
- The Timer 1 output can be used as the clock for the A/D converter in continuous operation mode.

Table 8.1-1 "Timer 1 Interval Times and Square Wave Frequencies in 8-bit Mode" to 8.1-3 "Interval Times and Square Wave Frequencies 16-bit Mode" list the interval time and square wave output ranges for the various modes.

Table 8.1-1 Timer 1 Interval Times and Square Wave Frequencies in 8-bit Mode

Count clock cycle		Interval time	Square wave output range (Hz)
Internal count clock	2 t_{inst}	2 t_{inst} to 2^9 t_{inst}	$1/(2^2\ t_{inst})$ to $1/(2^{10}\ t_{inst})$
	32 t_{inst}	2^5 t_{inst} to 2^{13} t_{inst}	$1/(2^6\ t_{inst})$ to $1/(2^{14}\ t_{inst})$
	512 t_{inst}	2^9 t_{inst} to 2^{17} t_{inst}	$1/(2^{10}\ t_{inst})$ to $1/(2^{18}\ t_{inst})$
External clock	1 t_{ext}	1 t_{ext} to 2^8 t_{ext}	$1/(2\ t_{ext})$ to $1/(2^9\ t_{ext})$

Table 8.1-2 Timer 2 Interval Times and Square Wave Frequencies In 8-bit Mode

Count clock cycle		Interval time	Square wave output range (Hz)
Internal count clock	2 t_{inst}	2 t_{inst} to 2^9 t_{inst}	$1/(2^2\ t_{inst})$ to $1/(2^{10}\ t_{inst})$
	32 t_{inst}	2^5 t_{inst} to 2^{13} t_{inst}	$1/(2^6\ t_{inst})$ to $1/(2^{14}\ t_{inst})$
	512 t_{inst}	2^9 t_{inst} to 2^{17} t_{inst}	$1/(2^{10}\ t_{inst})$ to $1/(2^{18}\ t_{inst})$
External clock	1 t_{ext}	1 t_{ext} to 2^8 t_{ext}	$1/(2\ t_{ext})$ to $1/(2^9\ t_{ext})$

Table 8.1-3 Interval Times and Square Wave Frequencies 16-bit Mode

Count clock cycle		Interval time	Square wave output range (Hz)
Internal count clock	2 t_{inst}	2 t_{inst} to 2^{17} t_{inst}	$1/(2^2\ t_{inst})$ to $1/(2^{18}\ t_{inst})$
	32 t_{inst}	2^5 t_{inst} to 2^{21} t_{inst}	$1/(2^6\ t_{inst})$ to $1/(2^{22}\ t_{inst})$
	512 t_{inst}	2^9 t_{inst} to 2^{25} t_{inst}	$1/(2^{10}\ t_{inst})$ to $1/(2^{26}\ t_{inst})$
External clock	1 t_{ext}	1 t_{ext} to 2^{16} t_{ext}	$1/(2\ t_{ext})$ to $1/(2^{17}\ t_{ext})$

t_{inst}: Instruction cycle (affected by clock mode, etc.)

t_{ext}: External clock period

Note:

Calculation example for the interval time and square wave frequency:

In this example, the main clock source oscillation (F_{CH}) is 4.2 MHz, the timer 1 data register (T1DR) value is set to "DD_H(221)" and the count clock cycle is set to the 8-bit mode operation at 2 tinst. In this case, the timer 1 interval time and frequency of square wave output from the TO pin (where the PWM timer operates continuously and the value of the T1DR register is constant) are calculated as follows.

Assume that the main clock mode (SCS = 1) and the highest clock speed (CS1/CS0 = 11_B) has been selected via the system clock control register (SYCC: SCS = 1, CS1 = 11_B, CS0 = 11_B) (1 instruction cycle = $4/F_{CH}$).

Interval time = $(2 \times 4/F_{CH}) \times$ (T1DR register value + 1)
= (8/4.2 MHz) × (221 + 1)
≅ 422.9 µs

Output frequency = $F_{CH}/(2 \times 8 \times$ (T1DR register value + 1))
= 4.2 MHz/ (16 × (221 + 1))
≅ 1.18 kHz

■ Counter Function

The counter function counts rising edges of an external count clock applied to the external pin (EC pin). Since the external clock can be selected only for Timer 1, the counter function operates in either the 8-bit Timer 1 or 16 bit mode.

- The counter counts up, clocked by external clocks. When the count equals the set value, it generates an interrupt request and inverts the level being output at the TO pin.
- In the 8 bit mode, Timer 1 can count as high as 2^8.
- In the 16 bit mode, the function counts as high as 2^{16}.
- By injecting an external clock having a set period, the counter function can be used the same way as the interval timer function.

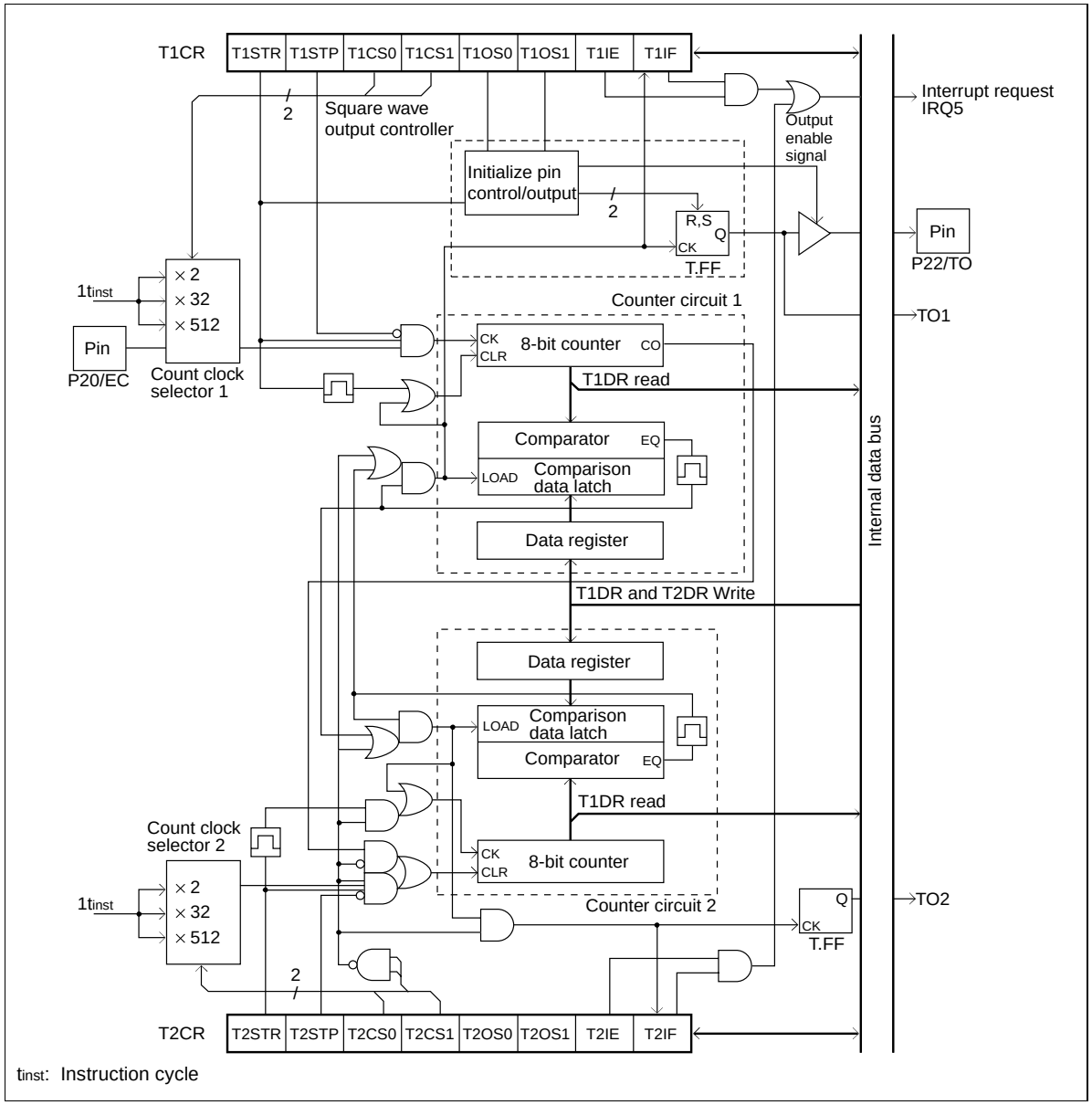
8.2 Block Diagram of 8/16-bit Timer/Counter

The 8/16-bit timer/counter consists of the following five blocks:

- **Count clock selectors 1 and 2**
- **Counter circuits 1 and 2**
- **Square wave output controller**
- **Timer data registers (T1DR and T2DR)**
- **Timer control registers (T1CR and T2CR)**

■ Block diagram of 8/16-bit Timer/Counter

Figure 8.2-1 Block Diagram 8/16-bit Timer/Counter



- **Count clock selectors 1, 2**

  This circuit selects an input clock. In the 8-bit timer 1 and 16 bit modes, count clock selector 1 selects one of four clocks: three internal clocks, and an external clock. In the 8-bit mode, count Clock Selector 2 selects one of three internal clocks only.

- **Counter circuit 1, 2**

  Counter circuit 1 and counter circuit 2 are each made up of an 8-bit counter, a comparator, a comparison data latch, and a data register (T1DR or T2DR).

  In each counter circuit, the 8-bit counter is an up-counter clocked by the selected count clock. The comparator compares the count in the counter with the value in the comparison data latch. When it detects a match, it clears the counter, and loads the contents of the data register into the comparison data latch.

  In the 8 bit-mode, the two counter circuits operate independently as timer 1 and timer 2. In the 16-bit mode, the two circuits are connected in series to form a single 16-bit counter with counter circuit 1 forming the low (8 LSBs) end of the counter, and counter circuit 2 at the high (8 MSBs) end.

- **Square wave output control circuit**

  An interrupt request is generated when the comparator detects a match in the 8-bit timer 1 mode or the 16-bit mode. At this time, if the square wave output is enabled, the output control circuit inverts the level output at the TO pin.

  The circuit can also initialize the output level to have the output square wave start out in a specific state ("H" or "L").

- **T1DR and T2DR registers**

  The value to be compared with the count in the counter is set by writing the desired value into these registers. They can be read to determine the current counter values.

- **T1CR and T2CR registers**

  The T1CR and T2CR registers are used to select the function, to enable or disable operation, control interrupts, and check the timer/counter status.

8.3 Structure of 8/16-bit Timer/Counter

This section describes the pins, pin block diagram, registers, and interrupt source of the 8/16-bit Timer/Counter.

■ 8/16-bit Timer/Counter Pins

The 8/16-bit timer/counter uses the P20/EC and P22/TO pins. The P20/EC pin can function either as a general-purpose I/O port (P20), as the external clock input pin of timer (EC). The P22/TO pin can function either as general-purpose I/O port (P22), as the square wave output pin of time (TO).

EC:

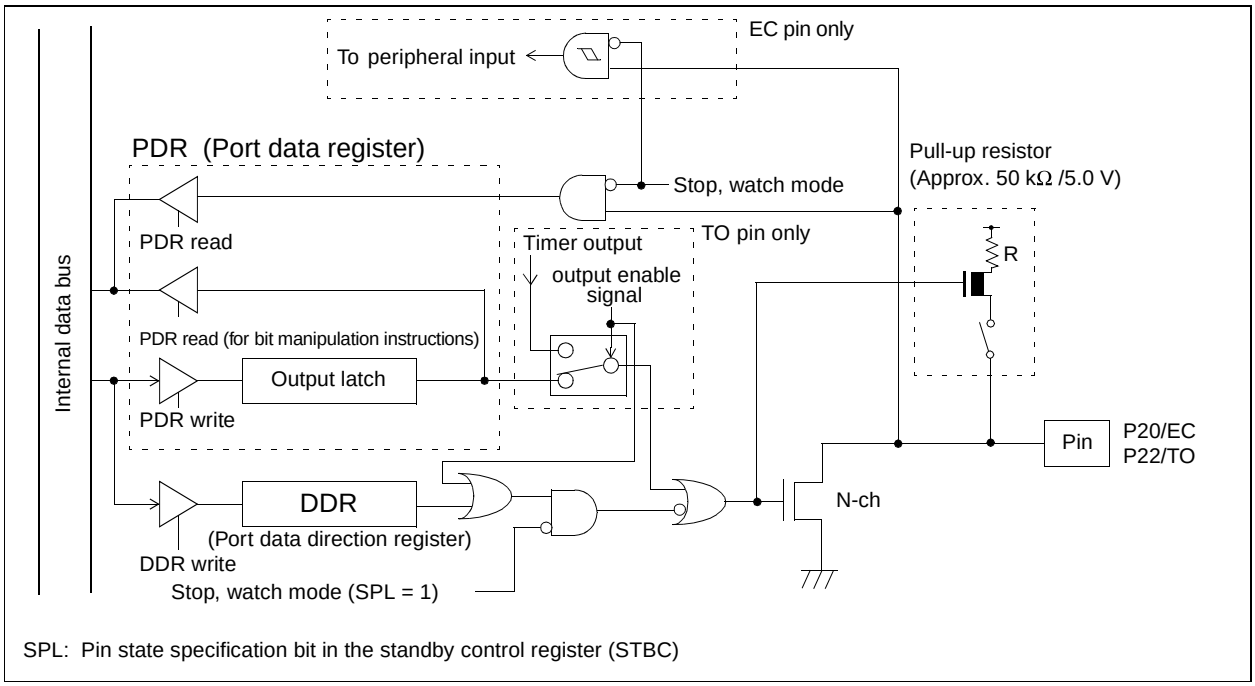
In the 8-bit timer 1 or 16-bit mode, if external clock input (counter function) is selected (T1CR: T1CS1 = 11_B, T1CS0 = 11_B), the counter counts the external clocks applied to this pin. The P20/EC pin sets the pin as an input port in the port data direction register DDR2: bit 0 = "0" when using as the EC pin.

TO:

In the 8-bit timer 1 or 16-bit mode, a square wave is output at this pin. Enabling square wave output (T1CR: T1OS1, T1OS0 = expect 00_B) automatically sets the P22/TO pin as an output pin, regardless of the port data direction register (DDR2: bit 2) value, and sets the pin to function as the TO pin.

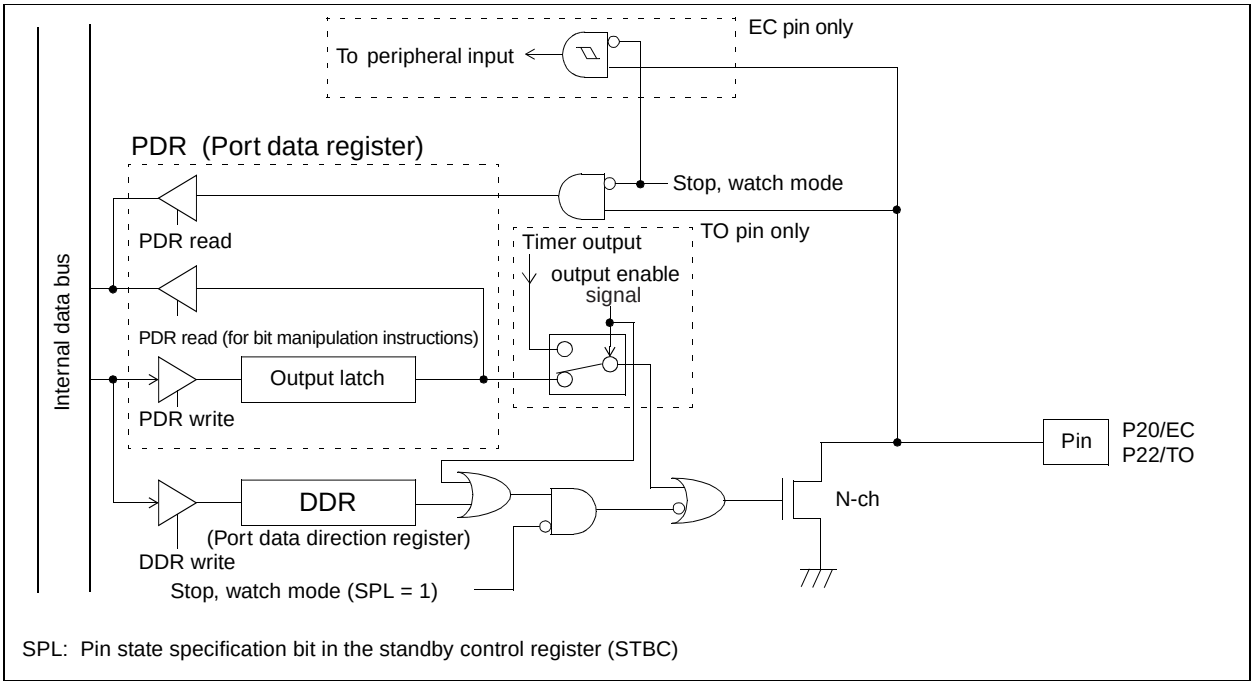
■ Block Diagram of 8/16-bit Timer Counter Pins

Figure 8.3-1 Block Diagram of 8/16-Bit Timer/Counter Pins for MB89983



SPL: Pin state specification bit in the standby control register (STBC)

Figure 8.3-2 Block Diagram of 8/16-Bit Timer/Counter Pins for MB89P985 and MB89PV980



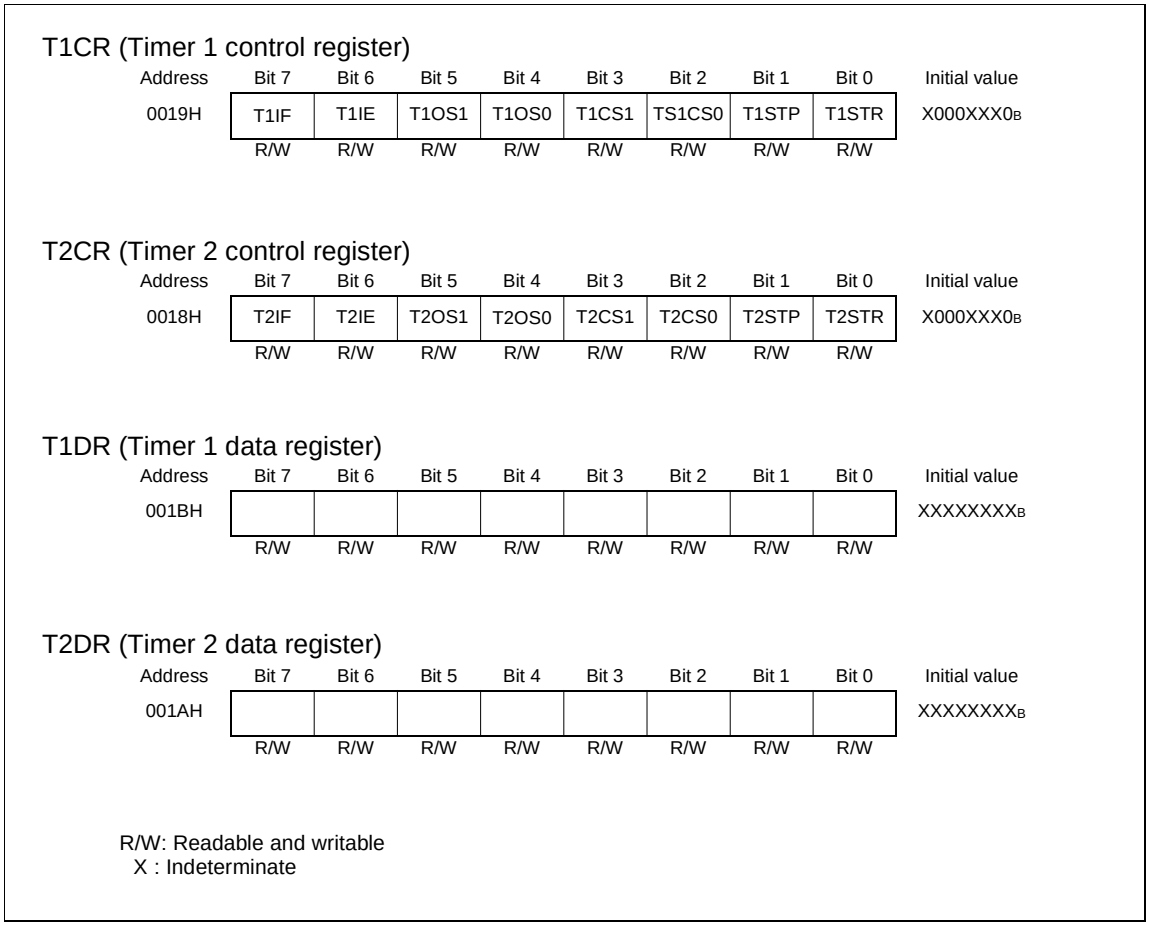
SPL: Pin state specification bit in the standby control register (STBC)

Note:

Pins with a pull-up resistor (optional) go to the "H" level during a reset or in stop mode and watch mode (SPL = "1").

■ 8/16-bit Timer/Counter Registers

Figure 8.3-3 8/16-bit Timer/Counter Registers



■ 8/16-bit Timer/Counter Interrupt Source

IRQ5:

In the interval timer and counter functions, if the interrupt request output is enabled, an IRQ5 interrupt request will be generated when the count in the counter equals the value set in the data register. Interrupt request outputs are enabled by setting the proper timer control register bit (T1CR: T1IE = 1 in the 8-bit Timer 1 or 16-bit mode; or T2CR: T2IE = 1 in the 8-bit Timer 2 mode).

8.3.1 Timer 1 Control Register (T1CR)

In the 8-bit Timer 1 and in the 16-bit mode, the Timer 1 Control Register (T1CR) is used to select functions, to enable/disable operation, to control interrupts, and to check status. In the 8-bit mode, the Timer 2 control register (T2CR) must still be initialized, even when only Timer 1 is used.

■ **Timer 1 Control Register (T1CR)**

Figure 8.3-4 PWC Pulse Width Control Register 1 (T1CR)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0019H	T1IF	T1IE	T1OS1	T1OS0	T1CS1	T1CS0	T1STP	T1STR	X000XXX0B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

T1STR	Timer activation bit
0	Stops counter
1	Clears counter, then starts it

T1STP	Timer stop bits
0	Causes counter to resume counting without clearing
1	Temporarily stops counter

T1CS1	T1CS0	Count clock selection Bits
0	0	2 t_{inst}
0	1	32 t_{inst}
1	0	512 t_{inst}
1	1	External clock

t_{inst}: instruction cycle

T1OS1	T1OS0	Square wave output control bits
0	0	Use pin for general-purpose port (P22)
0	1	Set the initial value of output square wave as "L" state
1	0	Set the initial value of output square wave as "H" state
1	1	Output set level at square wave output pin (TO).*

T1IE	interrupt request enable bit
0	Disable interrupt request output
1	Enable interrupt request output

T1IF	interrupt request Flag Bit	
	Read	Write
0	No counter match	clears this bit
1	Have counter match	No effect: The bit does not change.

R/W : Readable and writable
W : Write only
— : Unused bit
X : Indeterminate
□ : Initial value

*: The square wave output pin will go to the level corresponding to data set when T1STR is “0.”

Table 8.3-1 Timer 1 Control Register (T1CR) Bits

Bit		Function
Bit 7	T1IF: Interrupt request flag bit	• 8 bit-mode: • Set to "1" when the count in the timer 1 counter matches the value set in the T1DR, the timer 1 data register (comparison data latch). • 16-bit mode: • Set to "1" when the counts in the timer 1 and timer 2 counters match the values set in the T1DR and T2DR registers, respectively. • An interrupt request is output when both this bit and the interrupt request enable bit (T1IE) are "1". • Writing "0" clears this bit. writing "1" has no effect and does not change the bit value.
Bit 6	T1IE: Interrupt request enable bit	• This bit enables or disables an interrupt request output to the CPU. • An interrupt request is output when both this bit and the interrupt request flag bit (T1IF) are "1".
Bit 5 Bit 4	T1OS1 and T1OS0: Square wave output control bits	• P22/TO is a general-purpose I/O port pin (P22) if both of these bits are "00B". If either bit is "1", it is the square wave output pin (TO). • If written to "01B", or "10B", the initialize data will be set in the square wave output controller, but the corresponding level will not be output to the TO pin. • If both bits are "11B", and the function is in the stop timer state (T1STR= 0), the TO pin is set to a level corresponding to the initialize data.
Bit 3 Bit 2	T1CS1 and T1CS0: Clock source selection bits	• Selects the count clock to be supplied to the counter. • Selects one of three internal clocks, or an external clock. • When both bits are "11B", Timer 1 operates as a counter with the external clock is selected as the count clock. **Check:** If external clock input is selected (T1CS1, T1CS0 = 11B), P20/EC must be set as an input port.
Bit 1	T1STP: Timer stop bit	• This bit is used to temporarily stop the counter. • Writing this bit to "1" temporarily stops the counter. Writing it to "0" when the timer in startup state (T1STR = 1), restarts the counter where it left off.
Bit 0	1STR: Timer activation bit	• Starts and stops timer. • Changing this bit from "0" to "1" clears the counter. At this time, if the timer is in the continuous operation mode (T1STP = 0), the counter starts (counts up, clocked by the selected count clock). Writing this bit to "0" stops the counter. • In the 16 bit mode, both Timer 1 and Timer 2 are cleared at timer start (T1STP = 0 --> 1).

Check:

Before using 8/16-bit timer/counter Timer 1 only in 8 bit mode, first set the timer count clock selection bits of the Timer 2 control register (T2CR: T2CS1, T2CS0) to some state other than "11_B". Operating in this mode without making this register setting could result in faulty operation.

Remark:

The square wave output pin, TO1 will be unknown at the time from setting the initial value of square wave output by T1OS0, T1OS1 ("01_B" or "10_B" to setting output to TO1(T1OS0, T1OS1= "11_B").

8.3.2 Timer 2 Control Register (T2CR)

In the 8 bit mode, the Timer 2 Control Register (T2CR) is used to select functions, to enable/disable operation, to control interrupts, and to check states. In the 16 bit mode, although the function is controlled by the Timer 1 control register (T1CR), the Timer 2 control register (T2CR) must still be set.

■ **Timer 2 Control Register (T2CR)**

Figure 8.3-5 Timer 2 Control Register (T2CR)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0018H	T2IF	T2IE	T2OS1	T2OS0	T2CS1	T2CS0	T2STP	T2STR	X000XXX0B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

T2STR	Timer activation bit
0	Stops counter
1	Clears counter, then starts it

T2STP	Timer stop bit
0	Causes counter to resume counting without clearing
1	Temporarily stops counter

T2CS1	T2CS0	Count clock selection bits
0	0	2 t_{inst}
0	1	32 t_{inst}
1	0	512 t_{inst}
1	1	16-bit mode

T2OS1	T2OS0	Unused bits
0	0	Always write to "00".
0	1	Prohibited setting
1	0	Prohibited setting
1	1	Prohibited setting

T2IE	interrupt request enable bit
0	Disables interrupt request output
1	Enables interrupt request output

T2IF	interrupt request flag bit	
	Read	Write
0	No counter match	Clears this file
1	Have counter match	No effect: The bit does not change.

R/W : Readable and writable
W : Write-only
— : Unused
X : Indeterminate
□ : Initial value

Table 8.3-2 Timer 2 Control Register (T2CR) Bits

Bit		Function
Bit 7	T2IF: Interrupt request flag bit	• Set to "1" when the count in the timer 2 counter matches the value set in the T2DR, the Timer 2 data register (comparison data latch). • An interrupt request is output when both this bit and the interrupt request enable bit (T2IE) are "1". • Writing "0" clears this bit. Writing "1"has no effect and does not change the bit value. **Check:** In the 16-bit mode, the T1IF bit is the valid interrupt request flag, and the T2IF bit has no effect.
Bit 6	T2IE: Interrupt request enable bit	• This bit enables or disables an interrupt request output to the CPU. • An interrupt request is output when both this bit and the interrupt request flag bit (T2IF) are "1". **Check:** In the 16 bit mode, the T1IE bit is the valid interrupt request enable bit, and the T2IE bit has no effect.
Bit 5 Bit 4	T2OS1 and T2OS0: Unused bits	These bits are not used in timer 2. They should always be written to "00".
Bit 3 Bit 2	T2CS1 and T2CS0: Clock Source Selection Bits	• TSelects the count clock to be supplied to the counter. • Selects one of three internal clocks. • Setting to "11B" selects the 16-bit mode. **Check:** In 16-bit mode, T1CS1 and T1CS0 select the clock. T2CS1 and T2CS0 serve only to select the 16 bit-mode.
Bit 1	T2STP: Timer stop Bit	• This bit is used to temporarily stop the counter. • Writing this bit to "1" temporarily stops the counter. Writing it to "0" when the timer start bit (T2STR) is "1", restarts the counter where it left off. **Check:** In 16-bit mode, T1STP is the stop bit, and T2STP has no effect.
Bit 0	T2STR: Timer activation bit	• Starts and stops timer. • Changing this bit from "0" to "1" clears the counter. At this time, if the T2STP bit is "0", the counter starts (counts up, clocked by selected count clock). Writing this bit to "0" stops the counter. **Check:** In 16-bit mode, T1STR is the start bit, and T2STR has no effect.

Check:

When using timer 2 in the 16-bit mode, set T2CS1 and T2CS0 to "11B"; then use the T1CR register to control the circuit.

8.3.3 Timer 1 Data Register (T1DR)

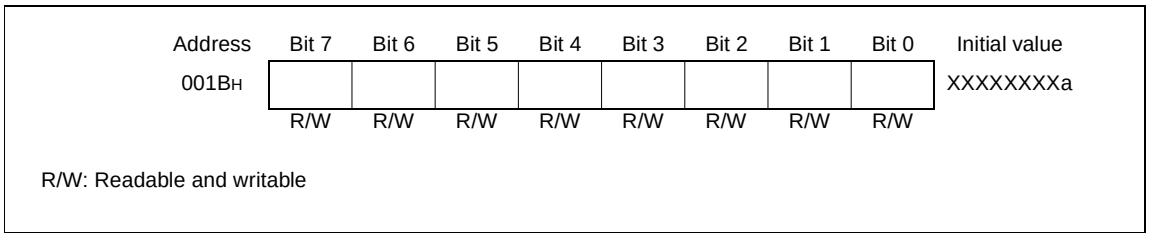
The Timer 1 data register (T1DR) is used to set all or part of the interval time or counter value, and to read out all or part of the counter value, depending on the mode and function being used. In 8-bit mode, it sets the Timer 1 interval time (interval timer function) or counter value (counter function), and reads out the counter value. In 16-bit mode, it sets the 8 LSBs of the 16-bit timer interval (interval timer function) or counter value (counter function), and reads out the counter value.

■ Timer 1 Data Register (T1DR)

The value set into this register is compared with the counter value (count). If you read the register, you get the current counter value. The register setting cannot be read out.

Figure 8.3-6 "Timer 1 Data Register (T1DR)" shows the bit structure of the Timer 1 data register.

Figure 8.3-6 Timer 1 Data Register (T1DR)



❍ 8-bit mode (Timer 1)

The value in this register is compared with the count in the timer 1 counter. For the interval timer function it sets the interval time, and for the counter function, it sets the count to be detected. When count operation enabled (T1CR: T1STR = 0 --> 1, T1STP = 0), the value in the T1DR register is loaded into the comparison data latch, and the counter starts counting up.

When the counter counts up to where it matches the value in the comparison data latch, the value in the T1DR register is re-loaded into the comparison data latch, and the counter is cleared and continues to count.

Since the comparison data latch is reloaded when a match is detected, if a new value is loaded into the T1DR register while the counter is counting, the new value will not take effect until the next count cycle (after a match is detected in the current cycle).

Note:

The T1DR setting for interval timer operation can be calculated using the following formula. (The instruction cycle time is affected by the clock mode, and the speed-shift selection.)

T1DR register value = interval time/(count clock cycle × instruction cycle) -1

❍ **16-bit mode**

The value in this register is compared with the counter value for the lower 8 bits .

(LSBs) of the 16-bit timer. In the interval timer function, this sets the lower 8 bits of the interval time setting, and in the counter function, the lower 8 bits of the count to be detected. The contents of the T1DR register are loaded into the lower 8 bits of the comparison data latch when the counter first starts operating and when a match is detected in the 16-bit count. Therefore, if a new value is loaded into the T1DR register while the 16-bit counter is counting, the new value will not take effect until after the next match is detected.

Reference:

For information on T1DR settings in the interval timer mode, refer to Section 8.3.4 "Timer 2 Data Register (T2DR)".

8.3.4 Timer 2 Data Register (T1DR)

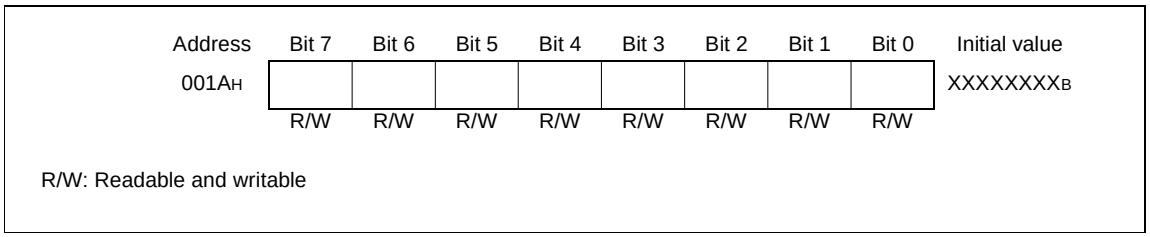
The Timer 2 data register (T2DR) is used to set all or part of the interval time or counter value, and to read out all or part of the counter value, depending on the mode and function being used. In 8-bit mode, it sets the Timer 2 interval time (interval timer function) or counter value (counter function), and reads out the counter value. In 16-bit mode, it sets the 8 MSBs of the 16-bit timer interval (interval timer function) or counter value (counter function), and reads out the counter value.

■ Timer 2 Data Register (T2DR)

The value set into this register is compared with the counter value (count). If you read the register, you get the current counter value. The register setting cannot be read out.

Figure 8.3-7 "Timer 2 Data Register (T2DR)" shows the bit structure of the Timer 2 data register.

Figure 8.3-7 Timer 2 Data Register (T2DR)



❍ 8-bit mode (Timer 2)

The value in this register is compared with the count in the timer 2 counter. For the interval timer function, it sets the interval time, and for the counter function, it sets the count to be detected. The value in the T2DR register is reloaded into the comparison data latch when counter operation starts, and when a match is detected.

If a new value is loaded into the T2DR register while the counter is counting, the new value will not take effect until the next count cycle (after a match is detected in the current cycle).

Note:

The T2DR setting for interval timer operation can be calculated using the following formula. (The instruction cycle time is affected by the clock mode, and the speed-shift selection.)

T2DR register value = interval time/(count clock cycle × instruction cycle time) -1

❍ 16-bit mode

The value in this register is compared with upper 8 bits (MSBs) of the 16-bit timer. In the interval timer function, this sets the upper 8 bits of the interval time setting, and in the counter function, the upper 8 bits of the count to be detected. The contents of the T2DR register are loaded into the upper 8 bits of the comparison data latch when the counter first starts operating and when a match is detected in the 16-bit count. Therefore, if a new value is loaded into the T2DR register while the 16-bit counter is counting, the new value will not take effect until after the next match is detected. In the 16 bit mode, the operation of the counter is controlled by the Timer 1 control register (T1CR).

Note:

In the interval timer function, the T1DR and T2DR register settings can be calculated from the following formula. (The instruction cycle time is affected by the clock mode, and the speed-shift selection.)

16-bit data value = interval time/(count clock cycle × instruction cycle) -1

The 8 MSBs of the 16-bit data value are the T2DR setting, and the 8 LSBs are the T1DR setting.

8.4 8/16-bit Timer/Counter Interrupt

In the 8/16-bit timer/counter, interrupt conditions are satisfied (if interrupts are enabled) when the counter matches the data register. This is true for both the interval timer and counter functions.

■ 8/16-bit Timer/Counter Interrupt

Table 8.4-1 "8/16-bit Timer/Counter Interrupt Control Bits and Interrupts" lists the 8/16-bit timer/counter interrupts, interrupt request flags and IRQ output enable bits.

Table 8.4-1 8/16-bit Timer/Counter Interrupt Control Bits and Interrupts

	8-bit mode		16-bit mode
	Timer 1	Timer 2	Timer 1+Timer2
Interrupt request flag bit	T1CR:T1IF	T2CR:T2IF	T1CR:T1IF
interrupt request enable bit	T1CR:T1IE	T2CR:T2IE	T1CR:T1IE
Interrupt source	8-bit counter matches T1DR	8-bit counter matches T2DR	16-bit counter matches T1DR+T2DR

In 8-bit mode, 8/16-bit timer/counter interrupt requests are generated independently for Timer 1 and Timer 2. In 16-bit mode, the interrupt request is generated only for Timer 1, but basic operation is the same. Interrupt operation will therefore be described only for Timer 1 in 8-bit mode.

❍ 8-bit mode timer 1 interrupt operation

The counter counts up from "00H" clocked by the selected count clock. When the count in the counter matches the value in the comparison data latch (corresponding to the value in timer data register T1DR), the interrupt request flag bit is set to "1" (T1CR: T1IF).

At this time, an interrupt request (IRQ5) to the CPU is generated if the interrupt request enable bit is enabled (T1CR: T1IF="1"). Write "0" to the TCEF bit in the interrupt processing routine to clear the interrupt request.

The T1IF bit is set to "1" when the counter value matches the set value, regardless of the value of the T1IE bit.

In 8 bit mode, although Timer 1 and Timer 2 operate independently of each other, they both generate IRQ5. When processing IRQ5, then, the software may have to check the interrupt request flag bits to determine which timer generated the interrupt.

Note:

The T1IF bit is not set if the counter is stopped (T1CR: T1STR = "0" at the same time as the counter value matches the T1DR register.

An interrupt request is generated immediately if the T1IF bit is "1" when the T1IE bit is changed from disabled to enabled ("0" --> "1").

■ Registers and Vector Table for 8/16-bit Timer/Counter Interrupt

Table 8.4-2 Registers and Vector Table for 8/16-bit Timer/Counter Interrupt

Interrupt	Interrupt level settings register			Vector table address	
	Register	Setting bits		Upper	Lower
IRQ5	ILR2 (007DH)	L51 (Bit 3)	L50 (Bit 2)	$FFF0_H$	$FFF1_H$

Reference:

See section 3.4.2 "Interrupt Processing" for details on the interrupt operation.

8.5 Operation of Interval Timer Function

This section describes the operation of the interval timer function of the 8/16-bit timer/counter.

■ Operation of Interval Timer Function

❍ 8-bit mode

Figure 8.5-1 "Interval Timer Function (Timer 1) Settings" shows the settings required to operate Timer 1 as the interval timer function in the 8-bit mode.

Figure 8.5-1 Interval Timer Function (Timer 1) Settings

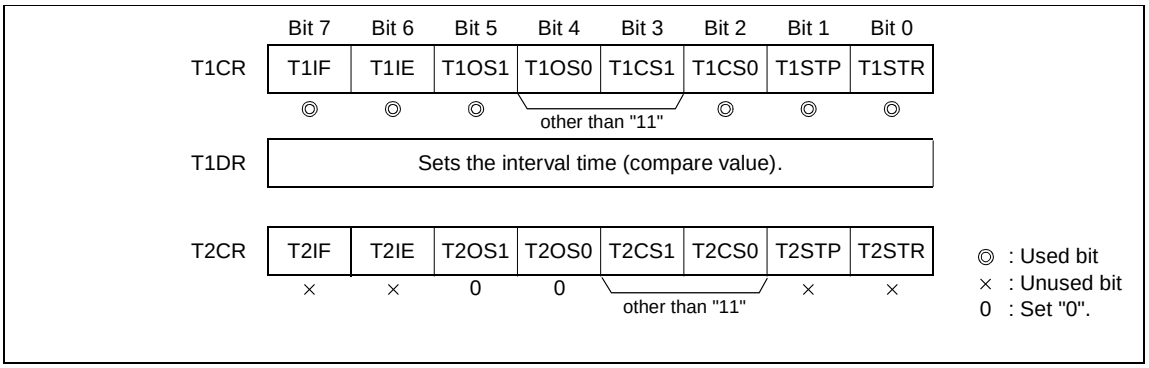
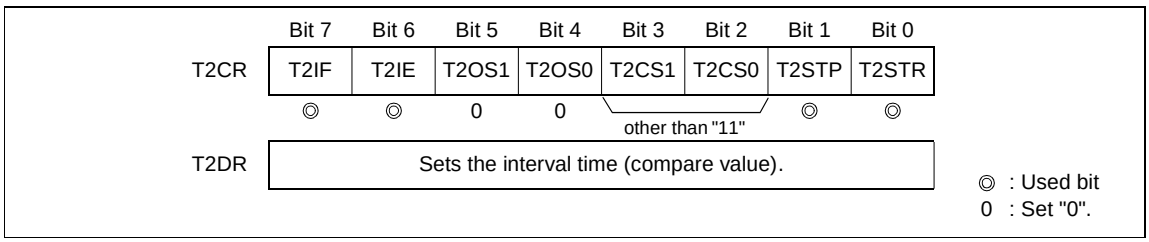


Figure 8.5-2 "Interval Timer Function (Timer 2) Settings" shows the settings required to operate Timer 2 as the interval timer function in the 8-bit mode.

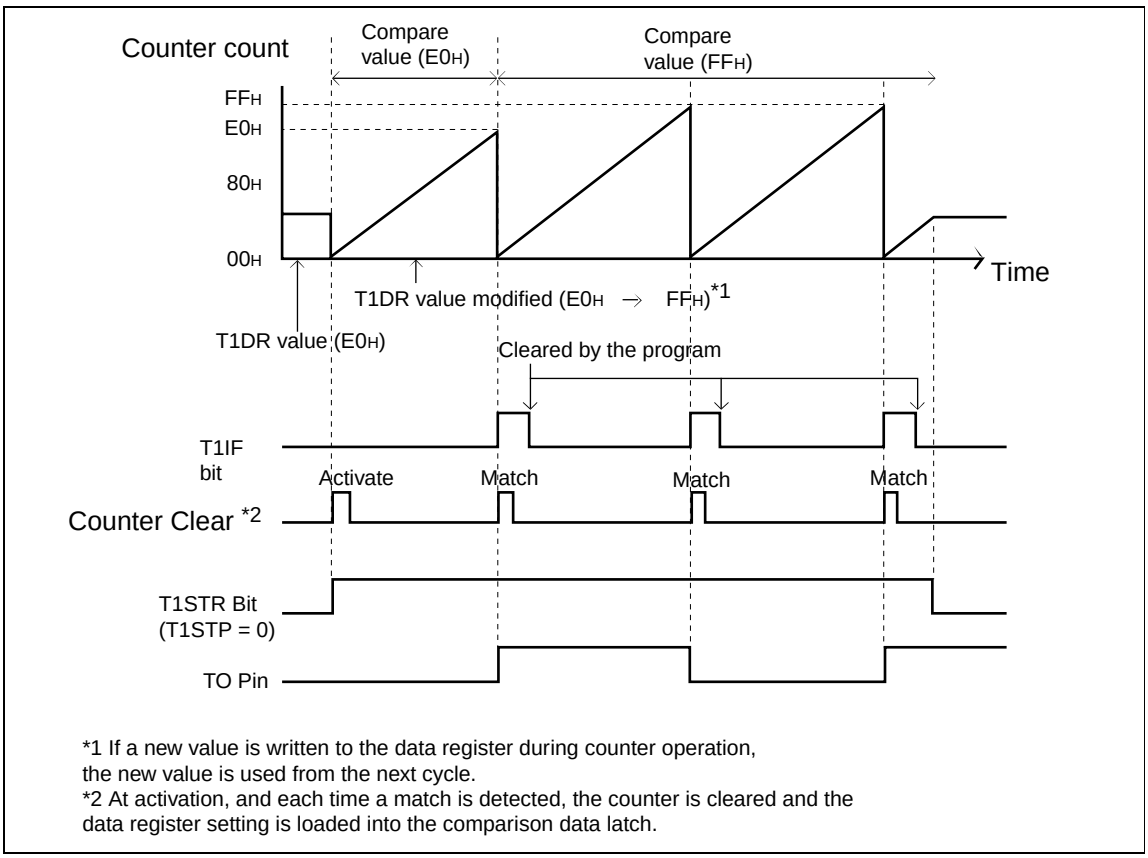
Figure 8.5-2 Interval Timer Function (Timer 2) Settings



On activation in 8-bit mode, the counter starts counting-up from "00_H" on the rising edge of the selected count clock. Eventually, the count in the counter will match the value set in the data register (comparison data latch). When this occurs, the timer control register interrupt request flag bit (T1CR: T1IF) is set to "1" and the counter starts counting-up again from "00_H". If using timer 1, the output of the square wave output control circuit is inverted when a match is detected; and if square wave output is enabled (T1CR: T1OS1, T1OS0 = values other than "00_B", a square wave is output at the TO pin.

Figure 8.5-3 "Operation of Interval Timer (Timer 1)" shows the interval timer function operation in the 8 bit mode.

Figure 8.5-3 Operation of Interval Timer (Timer 1)



*1 If a new value is written to the data register during counter operation, the new value is used from the next cycle.
*2 At activation, and each time a match is detected, the counter is cleared and the data register setting is loaded into the comparison data latch.

❍ 16-bit mode

Figure 8.5-4 "Interval Timer Function Settings (16-bit Mode)" shows the settings required to operate the interval timer function in the 16-bit mode.

Figure 8.5-4 Interval Timer Function Settings (16-bit Mode)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CR	T1IF	T1IE	T1OS1	T1OS0	T1CS1	T1CS0	T1STP	T1STR
	◎	◎	◎	◎	Other than "11"		◎	◎
T2CR	T2IF	T2IE	T2OS1	T2OS0	T2CS1	T2CS0	T2STP	T2STR
	×	×	0	0	1	1	×	×
T1DR	Sets the interval time (lower8 bits).							
T2DR	Sets the interval time (upper8 bits).							

◎ : Used bit
× : Unused bit
1 : Set "1".
0 : Set "0".

In 16-bit mode, the timer 1 control register (T1CR) controls the timer. The timer 2 control register (T2CR) must, however, still be initialized. The data to be compared with the 16-bit counter is set in both data registers: the upper 8 bits in T2DR and the lower 8 bits in T1DR. All 16 bits of the counter are cleared simultaneously. All other operation in the 16-bit mode is the same as timer 1 operation in 8-bit mode.

8.6 Operation of Counter Function

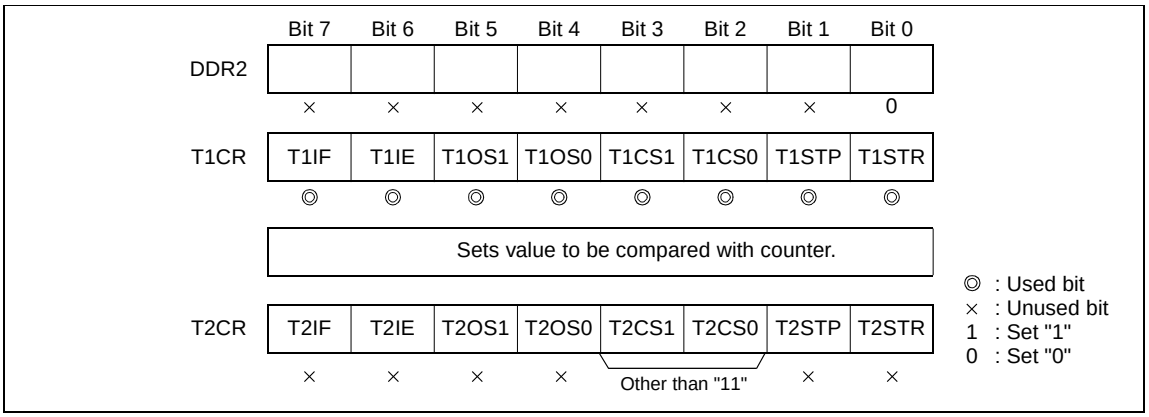
This section describes the operations of the counter function of the 8/16-bit timer/ counter.

■ Operation of Counter Function

❍ 8-bit mode

Figure 8.6-1 "Counter Function Settings (8-Bit Mode)" shows the settings required to operate the timer 1 as the counter function in the 8-bit mode.

Figure 8.6-1 Counter Function Settings (8-Bit Mode)

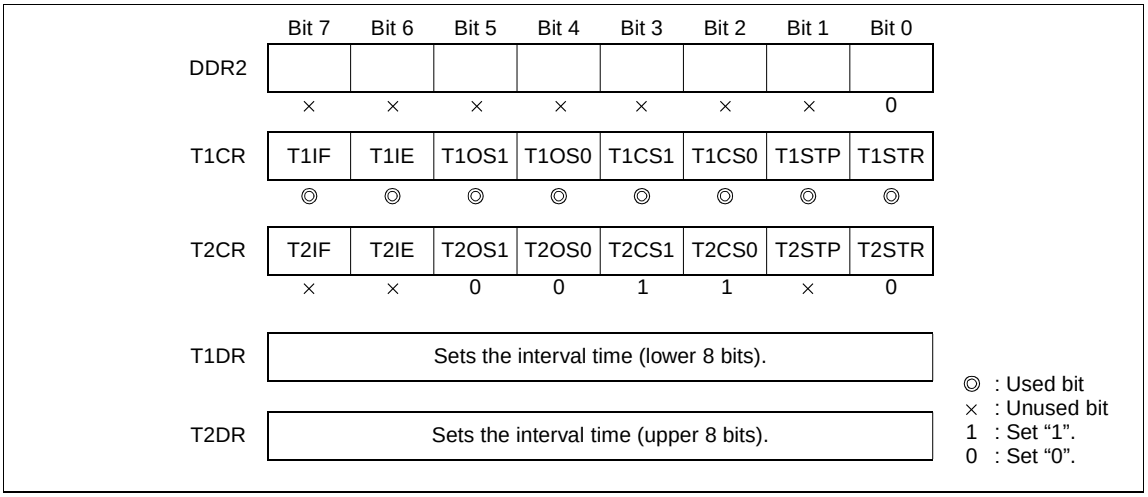


Counter operation in the 8-bit mode is the same as interval timer operation of timer 1 in 8-bit mode, except that an external clock is used in lieu of the internal clock.

❍ 16-bit mode

Figure 8.6-2 "Counter Function Settings (16-bit Mode)" shows the settings required to operate the counter function in the 16-bit mode.

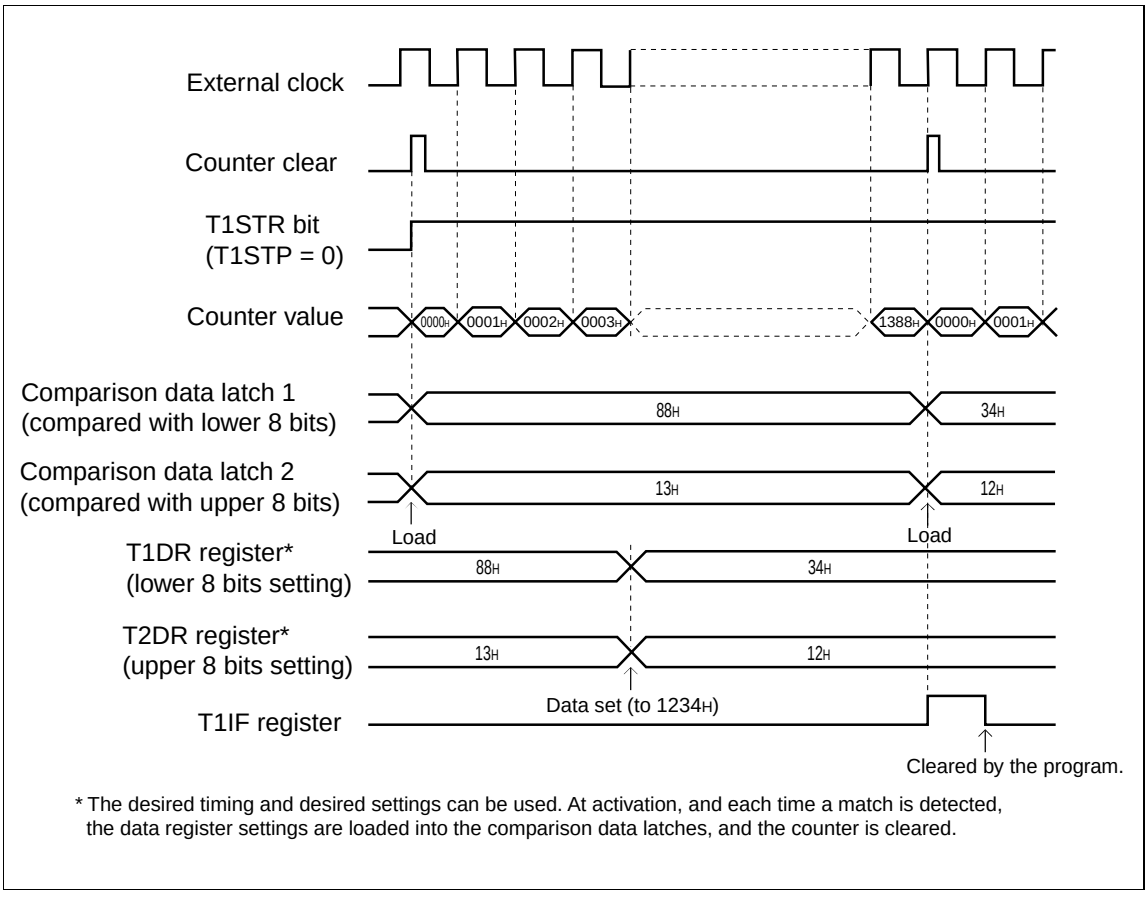
Figure 8.6-2 Counter Function Settings (16-bit Mode)



Counter operation in the 16-bit mode is the same as interval timer operation in 16-bit mode, except that an external clock is used in lieu of the internal clock.

Figure 8.6-3 "Operation of Counter Function in 16-bit Mode" shows the counter function operation in the 16-bit mode.

Figure 8.6-3 Operation of Counter Function in 16-bit Mode



* The desired timing and desired settings can be used. At activation, and each time a match is detected, the data register settings are loaded into the comparison data latches, and the counter is cleared.

Check:

When the counter value during operation is read out in 16-bit mode, always read it twice, and verify that a proper value is got before using it.

8.7 Operation of the Square Wave Output Initial Setting Function

The Square wave output can be set to the desired initial value using the timer 1 control register (T1CR).

■ Operation of Square Wave Output Initial Setting Function

The square wave output can be set to the desired initial value by the program, but this can be done only when the timer operation is stopped (T1CR: T1STR = 0).

Figure 8.7-1 "Square Wave Output Initial Setting Equivalent Circuit" shows an equivalent circuit for the square wave output control circuit initial setting. To perform the initial setting, follow the procedure in Table 8.7-1 "Square Wave Output Initial Setting Procedure (T1CR Register)". The operation of the square wave output when this is done is as shown in Figure 8.7-2 "Square Wave Output Initial Setting Operation".

Figure 8.7-1 Square Wave Output Initial Setting Equivalent Circuit

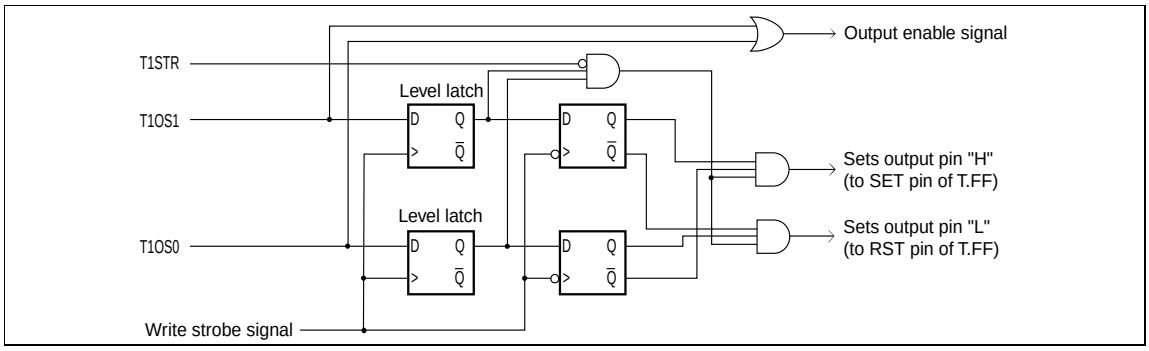
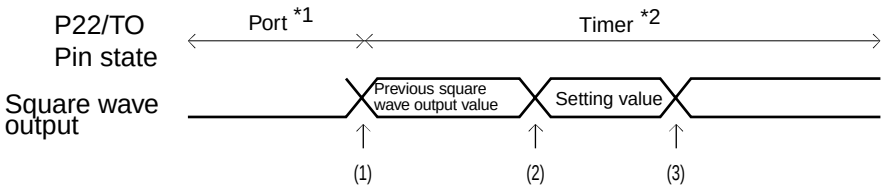


Table 8.7-1 Square Wave Output Initial Setting Procedure (T1CR Register)

Step	Settings and Operation
(1)	To set the square wave output pin (TO) "L", set the square wave output control bits (T1CR: T1OS1, T1OS0) first to "01B", then to "11B". To set the TO pin "H", set the bits to "10B" then "11B". **Note:** Until the bits are written to "11B", the circuit simply holds the latched value, and the TO pin level remains in its current or previous state.
(2)	If the square wave output control bits (T1OS1, T1OS0) are written to "11B" and the timer operation stopped (T1STR = 0), the TO pin will output the level corresponding to the level latch value (initial value). This can also be accomplished by setting T1OS1, T1OS0, and T1STR simultaneously. If the timer activation bit is set (T1STR= 1), the counter will start.
(3)	The square wave output is inverted each time the counter value matches the data register settings.

Figure 8.7-2 Square Wave Output Initial Setting Operation



*1: When the T1OS1 and T1OS0 bits of the T1CR register are both "00 B", the P22/TO pin is a general-purpose port pin (P22).
*2: If either T1OS1 or T1OS2 bit is "1", the P22/TO pin is a square wave output pin (TO).

8.8 Operation of 8/16-bit Timer/Counter Stop and Restart

This section describes the operation of stop and restart operation functions of the 8/16-bit timer/counter.

■ Timer Stop and Restart

Operation is described for timer 1 only. timer 2, however, operates the same way.

Timer 1 is stopped and restarted using the timer 1 control register stop and start bits (T1CR: T1STP and T1STR).

- To start the counter after clearing it, with "0" in T1STR bit

Set T1STP, T1STR bit to "01_B". On the T1STR bit rising edge, the counter will be cleared and start counting.

- To temporarily stop the counter and then resume counting (without clearing the counter).

First stop the counter by setting T1STP, T1STR to "11_B", then set T1STP, T1STR to "01_B" to resume counting where you left off.

Table 8.8-1 "Timer Stop and Restart" lists the timer states for each T1STP, T1STR bit, and operation when the timer is activated from that state (T1STP, T1STR = 01B).

Table 8.8-1 Timer Stop and Restart

T1STP (T2STP)	T1STR (T2STR)	Timer State	Timer operation when counter is activated (T1STP, T1STR="01B") from the state shown at the left
0	0	Counter stopped	Counter cleared and starts counting
0	1	Counter operating	Counter keeps on operating as-is
1	0	Counter stopped	Counter cleared and starts counting
1	1	Counter temporarily stopped	Counter resumes counting without being cleared

8.9 States in Each Mode during 8/16-bit Timer/Counter Operation

This section describes the operation of the 8/16-bit timer/counter when the device changes to sleep or stop mode or an operation halt request occurs during operation.

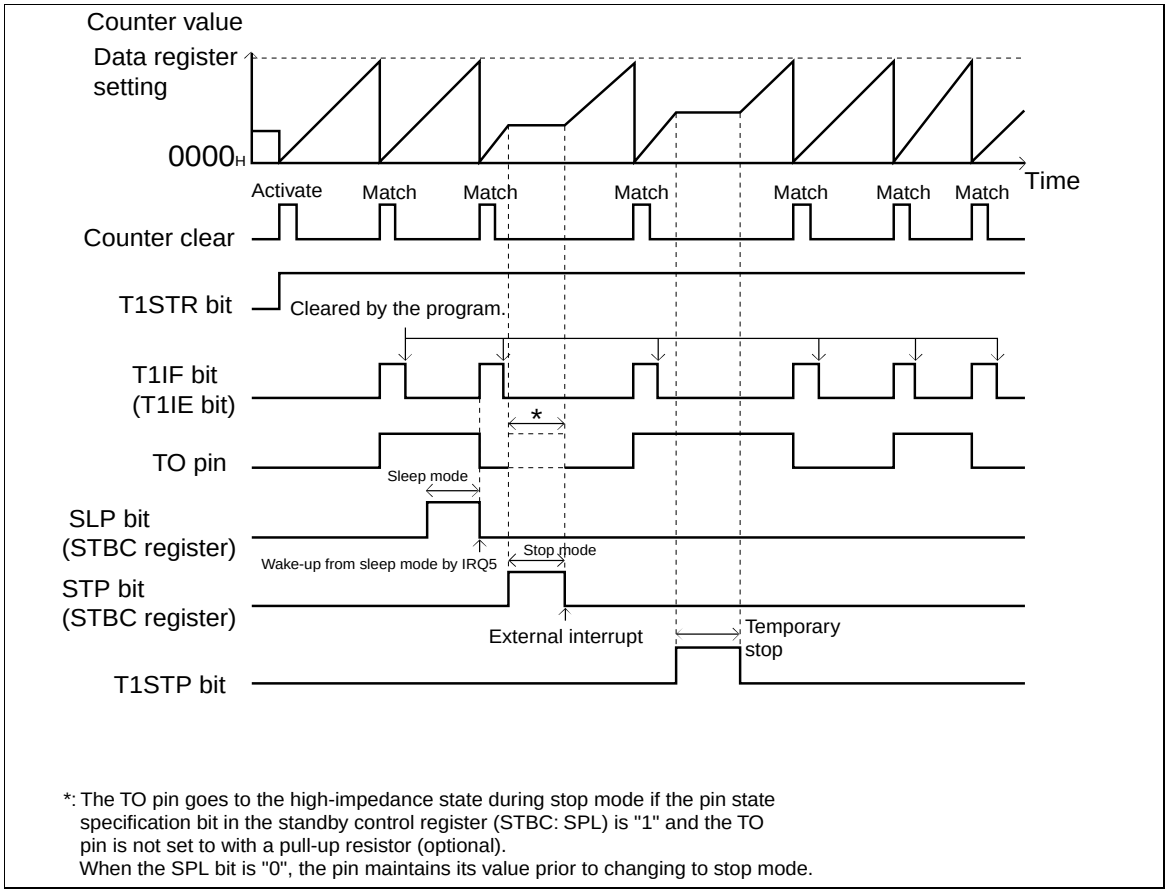
■ Operation during Subclock Mode, Standby Mode, or Operation Halt

Figure 8.9-1 "Counter Operation during Subclock Mode, Standby Mode, or Operation Halt" shows the counter value state when the device changes to sleep or stop mode, or an operation halt request occurs, during operation of the interval timer function or counter function (for timer 1).

The counter halts an maintains its current value when the device changes to stop mode. Operation starts again from the stored counter value after wake-up from stop mode by an external interrupt. therefore, the first interval time or external clock count is not correct value. Always initialize the 8/16-bit timer/counter after wake-up from stop mode.

Operation when entering or exiting timeclock mode (STBC: TMD = 1) is the same as when entering or exiting stop mode. Timeclock mode is cleared by a timeclock interrupt or external interrupt. When the counter is stopped temporarily (T1STP = 1), it holds the count it had when it was stopped. When it is restarted (T1STP= 0), it resumes counting from the count at which it was stopped.

Figure 8.9-1 Counter Operation during Subclock Mode, Standby Mode, or Operation Halt



*: The TO pin goes to the high-impedance state during stop mode if the pin state specification bit in the standby control register (STBC: SPL) is "1" and the TO pin is not set to with a pull-up resistor (optional).
When the SPL bit is "0", the pin maintains its value prior to changing to stop mode.

8.10 Notes on Using 8/16-bit Timer/Counter

This section lists points to note when using the 8/16-bit timer/counter.

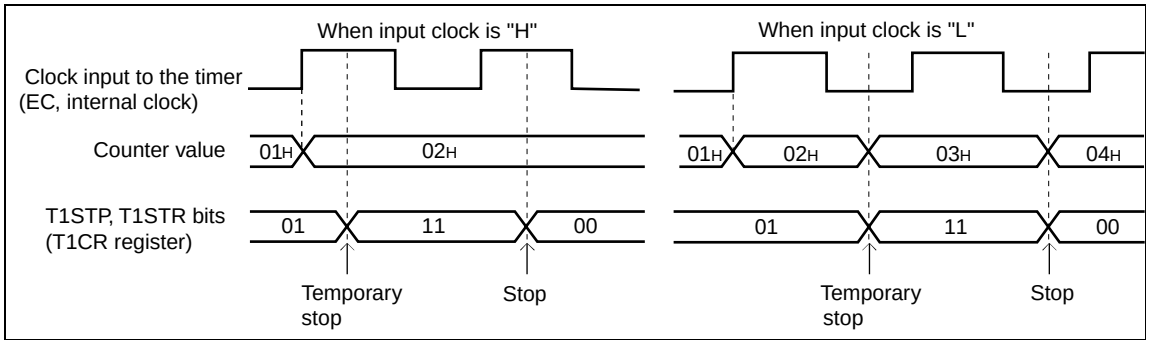
■ Notes on Using 8/16-bit Timer/Counter

❍ Notes when counter is stopped

This information is described for timer 1, but the same information applies to timer 2.

As shown in Figure 8.10-1 "Operation when Timer Stop Bit is Used", if the clock is "L" when T1STP temporarily stops the timer, the count will be incremented by 1. This may also occur if the input clock is "L" after a temporary stop, and the T1STP and T1STR bits are both written to "00B" simultaneously. When using the T1STP bit to temporarily stop the counter, first read out the counter value; then write T1STR bit to "0".

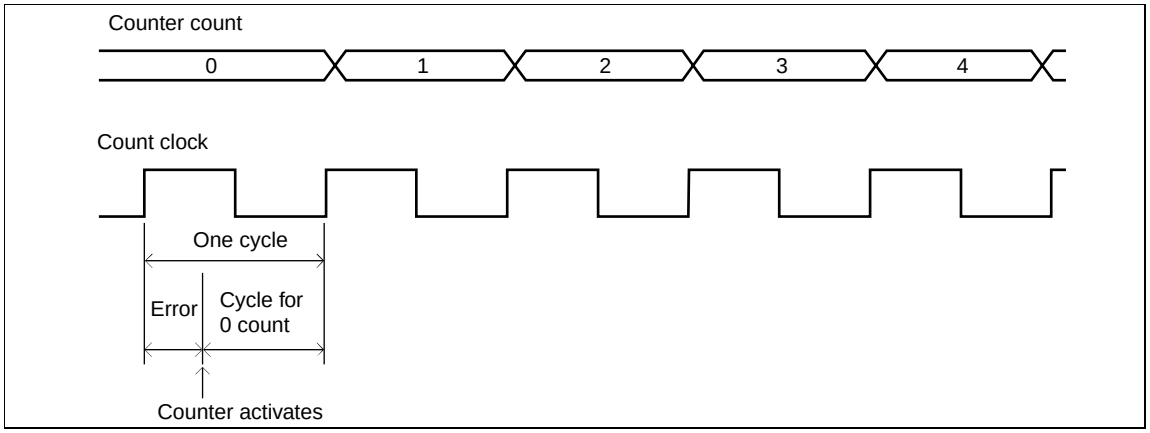
Figure 8.10-1 Operation when Timer Stop Bit is Used



❍ Error

Activating of by program 8/16-bit timer/counter is not synchronized with the start of counting-up using the selected count clock. Therefore, the time from activating the counter to match the register setting may be shorter than the theoretical time by a maximum of one cycle of the count clock. Figure 8.10-2 "Error on Starting Counter Operation" shows the error that occurs on standing counter operation.

Figure 8.10-2 Error on Starting Counter Operation



❍ **Using one 8-bit channel**

When 8/16-bit timer/counter timer 1 only is used in the 8-bit mode, before doing so, first set the timer count clock select bits of the timer 2 control register (T2CR: T2CS1, T2CS0) to some state other than "11_B". Failure to do so may result in faulty operation.

❍ **Notes on setting by program**

- When the 8/16-bit timer/counter Timer 1 only is used in the 16-bit mode, the timer 2 control register count clock select bits (T2CR: T2CS1, T2CS0) should always be set to "11_B" and bits 5 and 4, the unused bits (T2CR: TSOS1, TSOS0) to "00_B".
- In 16-bit mode, when the counter value is read out during operation, always read it twice and verify that a proper value is got before using it.
- While the timer is operating (T1CR: T1STR = 1), performing the initial state setting will not immediately cause the square wave output level to change. The output state will be initialized when the timer stops.
- Interrupt processing cannot return if the interrupt request flag bit (T1CR: T1IF, T2CR: T2IF) is "1" and the interrupt request enable bit is enabled (T1CR: T1IE= "1", T2CR: T2IE= "1"). Always clear the interrupt request flag bit.
- The interrupt request flag bit (TLCR: T1IF or T2CR: T2IF) is not set if the counter is disabled by the timer start bit (T1CR: T1STR=0 or T2CR: T2CR =0) at the same time as an interrupt source is generated.

8.11 Program Examples for 8/16-bit Timer/Counter

This section gives a program examples for 8/16-bit timer/counter.

■ Program Example for Interval Timer Function

❍ Processing description

- Using timer 1 only, in 8-bit mode, generates repeated interval timer interrupts at 20 ms intervals.
- Outputs a square wave to the TO pin that inverts after each interval time.
- With a main clock master oscillation F_{CH} of 4.2 MHz, and the highest speed main clock selected by the speed-shift function (1 instruction cycle time = $4/F_{CH}$), and with an internal clock period of 512 t_{inst} selected as the count clock, the T1DR setting for an interval of approximately 20 ms is calculated as follows:

  T1DR register value = 20 ms/(512 × 4/4.2 MHz) - 1 = 40.0 (28_H)

❍ Coding example

```
T2CR    EQU     0018H           ; Address of the Timer 2 control register
T1CR    EQU     0019H           ; Address of the Timer 1 control register
T2DR    EQU     001AH           ; Address of the Timer 2 data register
T1DR    EQU     001BH           ; Address of the Timer 1 data register
T1IF    EQU     T1CR:7          ; Define the timer 1 interrupt request flag bit.
ILR2    EQU     007DH           ; Address of the interrupt level setting register 2
INT_V   DSEG    ABS
        ORG     0FFF0H
IRQ5    DW      WARI            ; Set interrupt vector.
        ENDS

;---------- Main program ----------------------------------------------------------------------------------------

        CSEG                    ; [CODE SEGMENT]
                                ; Stack pointer (SP) etc. are already initialized.
```

```
        :
        CLRI                        ; Disable interrupts.
        MOV     ILR2,#11111011B     ; Set interrupt priority to level 2.
        MOV     T2CR,#00000010B     ; Clear timer 2 interrupt request flag, disable
                                      interrupt request
                                    ; output, set other  than 16-bit mode, stop
                                      operation.
        MOV     T1CR,#00011000B     ; Clear timer 1 interrupt request flag, initialize
                                      square wave
                                    ; output "L", select 512 tinst, and stop operation.
        MOV     T1DR,#28H           ; Set value compared with the counter value
                                      (interval time).
        MOV     T1CR,#00111000B     ; Output "L" at square wave output pin (TO).
        MOV     T1CR,#11111001B     ; timer 1 interrupt request, clear counter, and start
                                      timer
        SETI                        ; Enable CPU interrupts.

        :

;---------- Interrupt Program ----------------------------------------------------------------------

WARI    CLRB    T1IF                ; Clear interrupt request flag.
        PUSH    A
        W       A, T
        XCHW    A
        PUSH
        W
        :

        User processing

        :
        POPW    A
        XCHW    A, T
        POPW    A
        RETI
        ENDS

;-------------------------------------------------------------------------------------------------------------

        END
```

■ Program Example for Pulse Counter Function

❍ Processing description

- Using timer 1 and timer 2 in 16-bit mode, count external clocks input to the EC pin, and generate an interrupt once for each 5000 clocks (1388_H).
- Shows a sample program (READ16) for reading out the count in the 16-bit counter, while the counter is counting.

❍ **Coding example**

```
DDR2    EQU     000DH           ; Address of the Port 2 data direction register
T2CR    EQU     0018H           ; Address of the Timer 2 control register
T1CR    EQU     0019H           ; Address of the Timer 1 control register
T2DR    EQU     001AH           ; Address of the Timer 2 data register
T1DR    EQU     001BH           ; Address of the Timer 1 data register
T1IF    EQU     T1CR:7          ; Define the timer 1 interrupt request flag bit.
ILR2    EQU     007DH           ; Address of the interrupt level setting
NT_V    DSEG    ABS               register 2
        ORG     0FFF0H
IRQ5    DW      WARI            ; Set interrupt vector.
        ENDS

;---------- Main program ----------------------------------------------------------------------------------

        CSEG                    ; [CODE SEGMENT]
                                ; Stack pointer (SP) etc. are already
                                  initialized.

        :

        MOV     DDR2,#00000000B ; Set P20/EC pin as an input.
        CLRI                    ; Disable interrupts.
        MOV     ILR2,#11111011B ; Set interrupt level 2.
        MOV     T1DR,#088H      ; Set lower 8 bits of counter comparison value.
        MOV     T2DR,#013H      ; Set upper 8 bits of counter comparison value.
        MOV     T2CR,#00001100B ; Set timer 2 to 16-bit mode.
        MOV     T1CR,#01001101B ; Clear timer 1 interrupt request flag, enable
                                  interrupt
                                ; request output, set P22/TO
                                ; as general-purpose port (P22), select
                                  external clock,
                                ; clear counter, and start operation.
        SETI                    ; Enable CPU interrupts.

        :

;---------- Data read subroutine --------------------------------------------------------------

        :

READ16  MOVW    A,T2DR          ; 16-bit read, T1DR + T2DR.
        MOVW    A,T2DR          ; 16-bit read, T1DR + T2DR, save old value in
                                  T register.
        CMPW    A               ; Check first and second reads, compare A
                                  and T registers.
        BEQ     RET16           ; If match, return.
        XCHW    A,T
        CMPW    A               ; Old value + 1
        BNE     A
        RET     READ16          ; If mismatch, read again.

        :

;---------- Interrupt program -----------------------------------------------------------------
```

```
WARI    CLRB    T1IF            ; Clear interrupt request flag.
        PUSHW   A
        XCHW    A,T
        PUSHW   A

        :
        User process
        :

        POPW    A
        XCHW    A,T
        POPW    A
        RETI
        ENDS

;-----------------------------------------------------------------------------------------------------------

        END
```

CHAPTER 9 EXTERNAL INTERRUPT CIRCUIT 1 (EDGE)

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

9.1 Overview of External Interrupt Circuit 1

The external interrupt circuit 1 detects edges on the signals input to the four external interrupt pins and generates the corresponding interrupt requests to the CPU.

■ External Interrupt Circuit 1 Function (Edge Detection)

The external interrupt circuit 1 function detects specified edges on signals input to the external interrupt pins and to generate interrupt requests to the CPU. These interrupts can wake up the CPU from standby mode and change the device to the normal operating state (main-run or sub-run mode).

External interrupt pins	:	4 pins (P10/INT10 to P13/INT13)
External interrupt source	:	Inputs a specified edge (rising edge or falling edge) on the signal input to an external interrupt pin.
Interrupt control	:	Enable or disable to input external interrupts and to output in interrupt requests, by the external interrupt 1 control register (EIE1).
Interrupt flags	:	Detects specified edges by the external interrupt request flag bits in the external interrupt 1 flag register (EIF1).
Interrupt requests	:	Separate interrupt requests are generated for each external interrupt source (IRQ0, IRQ1, IRQ2, and IRQ3).

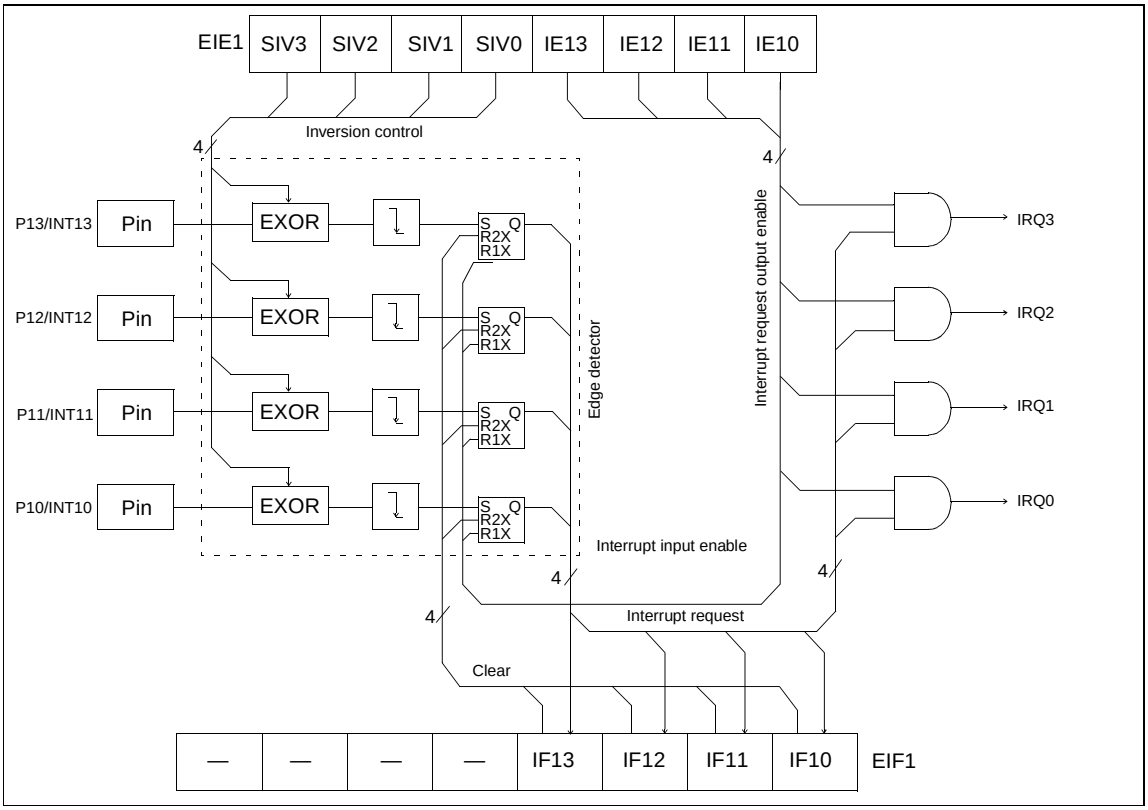
9.2 Block Diagram of External Interrupt Circuit 1

The external interrupt circuit 1 consists of the following three blocks:

- **Edge detector**
- **External interrupt 1 control register (EIE1)**
- **External interrupt 1 flag register (EIF1)**

■ Block Diagram of External Interrupt Circuit 1

Figure 9.2-1 Block Diagram of External Interrupt Circuit 1



❍ Edge detector circuit

The signals received at the external interrupt pins (INT10 to INT13) are either inverted or not inverted, based on the state of the corresponding signal inversion (SIV) bit of the EIE1 register. If external interrupt inputs are enabled, a falling edge is detected in one of the inverted or non-inverted signals, and the interrupt request flag bit for that INT pin (IF10 to IF13) is set to "1".

❍ EIE1 register

The states of the signal inversion bits (SIV0 to SIV3) of this register determine whether the corresponding external interrupt input signal will be inverted. The external interrupt enable bits (IE10 to IE13) simultaneously enable or disable both the interrupt inputs and the interrupt request outputs.

❍ **EIF1 register**

The external interrupt request flag bits (IF10 to IF13) of this register are used to check interrupt request status and clear the interrupt requests.

9.3 Structure of External Interrupt Circuit 1

This section describes the pins, pin block diagram, registers, and interrupt sources of the external interrupt circuit 1.

■ External Interrupt Circuit 1 Pins

The external interrupt circuit 1 uses four external interrupt pins.

The external interrupt pins can function either as external interrupt inputs (hysteresis inputs) or as general-purpose I/O ports.

When P10/INT10 to P13/INT13 Pins are set as inputs in their port 1 data direction register (DDR1), and the corresponding external interrupt inputs are enabled in the external interrupt 1 control register (EIE1) they operate as external interrupt input pins (INT10 to INT13). When they are being used as the port, the pin states, can be read from the port data register (PDR1) at any time

Table 9.3-1 "External Interrupt Circuit 1 Pins" lists the pins associated with external interrupt circuit 1.

Table 9.3-1 External Interrupt Circuit 1 Pins

External interrupt pin	When used as external interrupt input (interrupt input enabled)	When used as general-purpose I/O port (interrupt input disabled)
P10/INT10	INT10(EIE1:IE10=1,DDR1:bit0=0)	P10 (EIE1:IE10=0)
P11/INT11	INT11(EIE1:IE11=1,DDR1:bit1=0)	P11 (EIE1:IE11=0)
P12/INT12	INT12(EIE1:IE12=1,DDR1:bit2=0)	P12 (EIE1:IE12=0)
P13/INT13	INT13(EIE1:IE13=1,DDR1:bit3=0)	P13 (EIE1:IE13=0)

■ Block Diagram of External Interrupt Circuit 1 Pins

Figure 9.3-1 Block Diagram of External Interrupt Circuit 1 Pins for MB89983

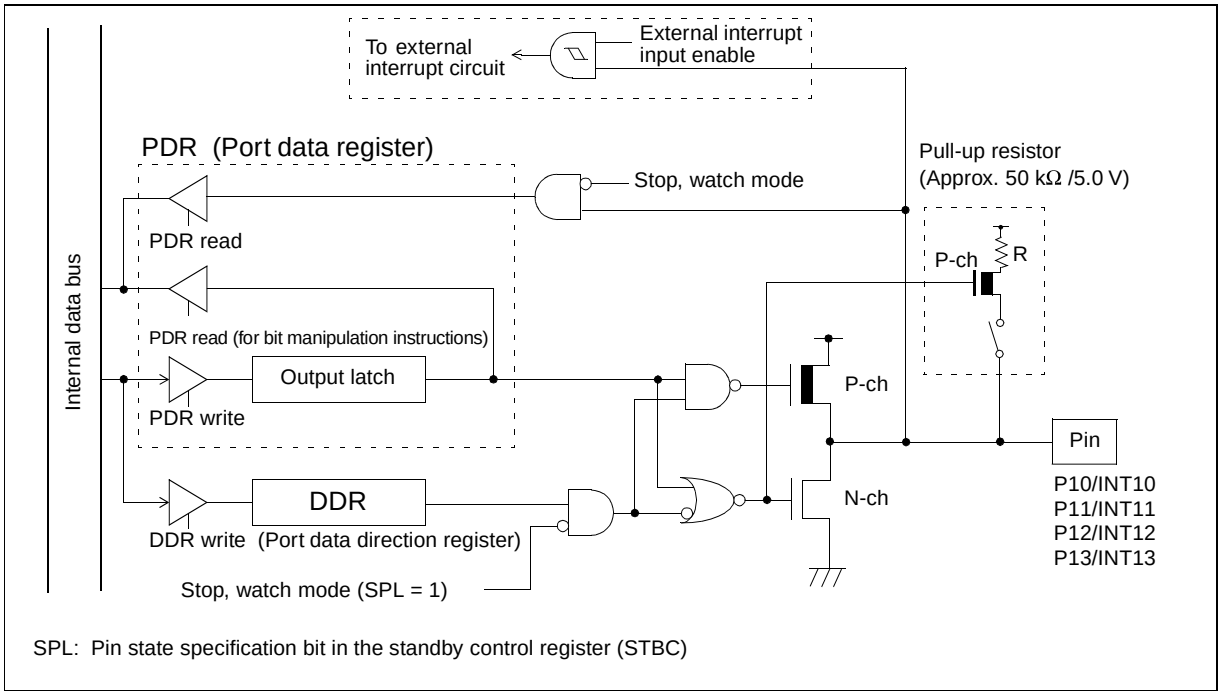
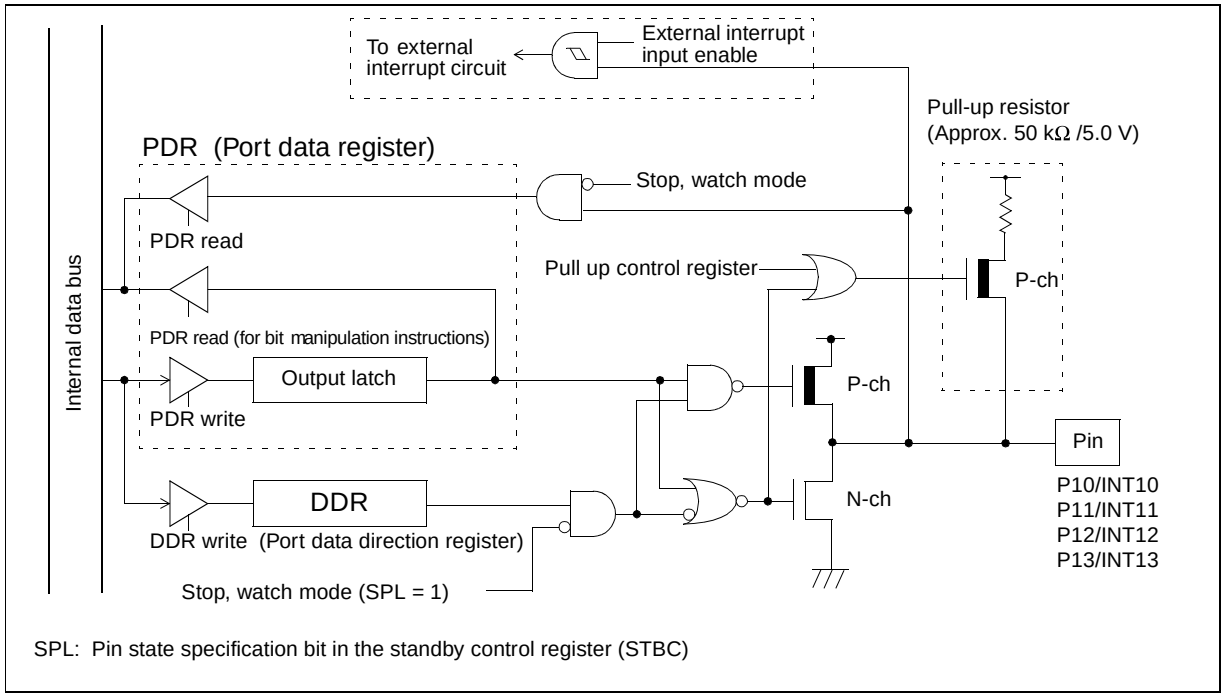


Figure 9.3-2 Block Diagram of External Interrupt Circuit 1 Pins for MB89P985 and MB89PV980



Note:

Pins with a pull-up register (optional) go to the "H" level during a reset or in stop and watch mode (SPL = "1").

■ External Interrupt Circuit 1 Registers

Figure 9.3-3 External Interrupt Circuit 1 Registers

EIE1 (External interrupt 1 control register)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0030H	SIV3	SIV2	SIV1	SIV0	IE13	IE12	IE11	IE10	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

EIF1 (External interrupt 1 flag register)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0031H	—	—	—	—	IF13	IF12	IF11	IF10	----0000B
					R/W	R/W	R/W	R/W	

R/W : Readable and writable
— : Unused
X : Indeterminate

■ External Interrupt Circuit 1 Interrupt Sources

IRQ0: External interrupt circuit generates an interrupt request (IRQ0) if an edge of the selected polarity is input to the external interrupt pin (INT0) when external interrupt is enabled (EIC1: EIE0 = "1").

IRQ1: External interrupt circuit generates an interrupt request (IRQ1) if an edge of the selected polarity is input to the external interrupt pin (INT1) when external interrupt is enabled (EIC1: EIE1 = "1").

IRQ2: External interrupt circuit generates an interrupt request (IRQ2) if an edge of the selected polarity is input to the external interrupt pin (INT2) when external interrupt is enabled (EIC2: EIE2 = "1").

IRQ3: External interrupt circuit generates an interrupt request (IRQ3) if an edge of the selected polarity is input to the external interrupt pin (INT3) when external interrupt is enabled (EIC2: EIE3 = "1").

9.3.1 External Interrupt 1 Control Register (EIE1)

The external interrupt 1 control register (EIE1) is used to select the inversion or non-inversion of interrupt input signal and to select enable or disable interrupts for external interrupt pins (INT10 to INT13).

■ **External Interrupt 1 Control Register (EIE1)**

Figure 9.3-4 External Interrupt 1 Control Register (EIE1)

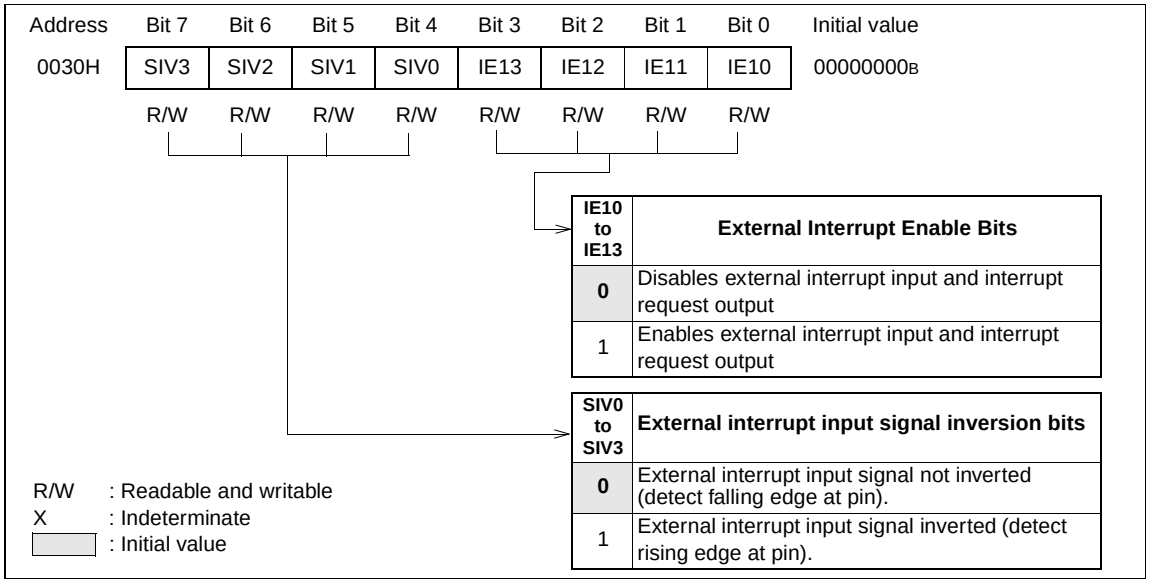


Table 9.3-2 External Interrupt 1 Control Register (EIE1) Bits vs. Interrupts Pins

	Bit	External interrupt pin	Interrupt
Bit 7 Bit 3	SIV3 IE13	INT13	IRQ3
Bit 6 Bit 2	SIV2 IE12	INT12	IRQ2
Bit 5 Bit 1	SIV1 IE11	INT11	IRQ1
Bit 4 Bit 0	SIV0 IE10	INT10	IRQ0

Note:

For MB89P985 and MB89PV980, there will be current leakage through the pull-up resistor in stop mode when the pull-up resistor is enabled and any enabled external interrupt 1 pin is input "0". To prevent the current leakage, the pull-up resistor should be disabled for each

enabled interrupt pin before going into stop mode.

Table 9.3-3 External Interrupt 1 Control Register (EIE1) Bits

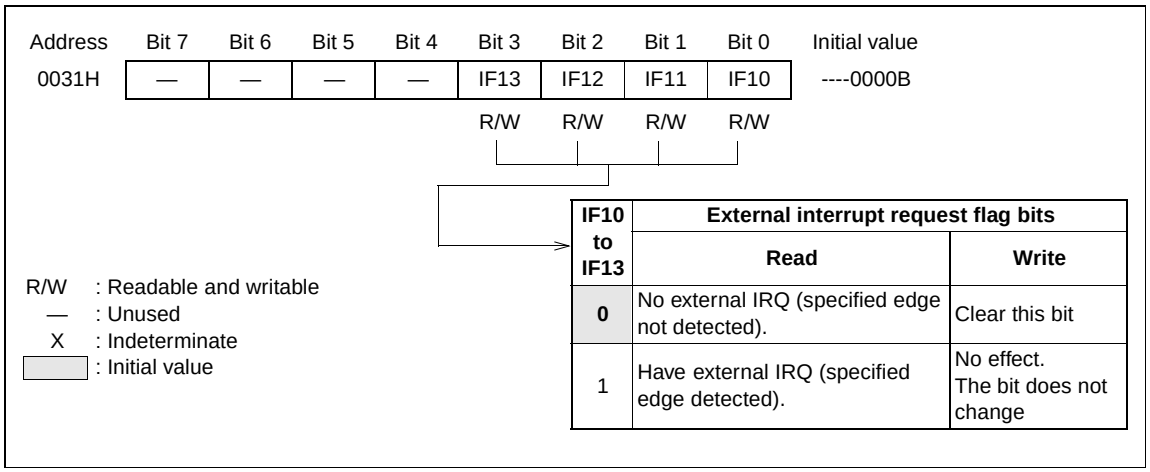
Bit		Function
Bit 7 Bit 6 Bit 5 Bit 4	SIV3 to SIV0: External Interrupt Input Signal Inversion Bits	• Select inversion/non-inversion of signal input to external interrupt pin. • A value of "0" has the effect of selecting falling edge detection for that pin, and a "1" selects rising edge detection.
Bit 3 Bit 2 Bit 1 Bit 0	IE13 to IE10: External Interrupt Enable Bits	Enables or disables both the external interrupt input, and the interrupt request output to the CPU. An interrupt request output when both this bit and the corresponding external interrupt request flag bit (IF13 to IF10) are "1". **Note:** When using an external interrupt pin, set it as an input by writing its bit in the port 1 data direction register (DDR1) to "0". The state of the external interrupt pin can always be read directly out of the port 1 data register (PDR1) regardless of the state of this interrupt enable bit.

9.3.2 External Interrupt 1 Flag Register (EIF1)

External Interrupt 1 flag register (EIF1) is used to hold the IRQ state when an interrupt edge has been detected, and to clear the interrupt.

■ External Interrupt 1 Flag Register (EIF1)

Figure 9.3-5 External Interrupt 1 Flag Register (EIF1)



IF10 to IF13	External interrupt request flag bits	
	Read	Write
0	No external IRQ (specified edge not detected).	Clear this bit
1	Have external IRQ (specified edge detected).	No effect. The bit does not change

Table 9.3-4 External Interrupt 1 Flag Register (EIF1) Bits

	Bit	External interrupt pin	Interrupt
Bit 3	IF13	INT13	IRQ3
Bit 2	IF12	INT12	IRQ2
Bit 1	IF11	INT11	IRQ1
Bit 0	IF10	INT10	IRQ0

Table 9.3-5 External Interrupt 1 Control Register (EIE1) Bits

Bit		Function
Bit 7 Bit 6 Bit 5 Bit 4	Unused bits	• This read value is indeterminate. • Writing to this bit has no effect on the operation.
Bit 3 Bit 2 Bit 1 Bit 0	IF13 to IF10: External interrupt request flag bits	• When a rising or falling edge, as specified by the state of the external interrupt in put signal inversion bit (SIV3 to SIV0), the corresponding IRQ flag bit is set to "1". • When both this bit and the corresponding interrupt enable bit (EIE1: IE13 to IE10) are set to "1", that IRQ is sent to the CPU. • Writing "0" clears this bit. Writing "1" has no effect and does not change the bit value. **Note:** When the external interrupt enable bits (EIE1: IE13 to IE10) are cleared to "0", the external interrupt inputs are disabled at the same time, which means that in this state, these bits will not be changed even when the specified edge is input.

9.4 External Interrupt Circuit 1 Interrupts

The external interrupt circuit 1 can generate interrupt requests when it detects a specified edge on the signal input to an external interrupt pin.

■ Interrupts for External Interrupt Circuit 1 Operation

If external interrupts are enabled (EIE1: IE10 to IE30 = 1) and the specified edge is detected at the external interrupt input, the corresponding external IRQ flag bit (EIF1: IF10 to IF13) is set to "1" and the corresponding IRQ (IRQ0 to IRQ3) sent to the CPU. Write "0" to the corresponding external interrupt request flag bit in the interrupt processing routine to clear the interrupt request.

Check:

When enabling interrupts (EIE1: IE10 to IE13 = "1") after wake-up from a reset, always clear the corresponding external interrupt request flag bit (EIE1: IF10 to IF13 = "0") in advance.

Also, interrupt processing cannot return if the external interrupt request flag bit is "1" and the interrupt request enable bit is enabled. In the interrupt processing routine, always clear the external interrupt request flag bit.

Notes:

Changing a signal inversion bit from the "non-invert" to the "invert" state while the INT pin is "H", or from "invert" to "non-invert" while the pin is LOW, will cause the external interrupt request flag bit (EIF1: IF10 to IF13) to be set immediately. Changing an external interrupt bit from "disable" to "enable" (EIE1: IE10 to IE13: 0 --> 1) may also set the external IRQ flag bit. For this reason, you should make the inversion or enable bit changes with interrupts in the disabled state, then clear IRQ flags before enabling interrupts again.

An interrupt request is generated immediately if the external interrupt request flag bit is "1" when the external interrupt enable bit is changed from disabled to enabled ("0" --> "1").

Wake-up from stop mode by an interrupt is possible using only the external interrupt circuit 1 and 2.

Perform with interrupts disabled, then clear external IRQ flags before enabling interrupts.

■ Register and Vector Table for External Interrupt Circuit 1 Interrupts

Table 9.4-1 Register and Vector Table for External Interrupt Circuit 1 Interrupts

IRQ	Interrupt level setting register			Vector table address	
	Register	Setting bits		Upper	Lower
IRQ0	ILR1 (007CH)	L01 (bit1)	L00 (bit0)	$FFFA_H$	$FFFB_H$
IRQ1		L11 (bit3)	L10 (bit2)	$FFF8_H$	$FFF9_H$
IRQ2		L21 (bit5)	L20 (bit4)	$FFF6_H$	$FFF7_H$
IRQ3		L31 (bit7)	L30 (bit6)	$FFF4_H$	$FFF5_H$

Reference:

See section 3.4.2 "Interrupt Processing" for details on the interrupt operation.

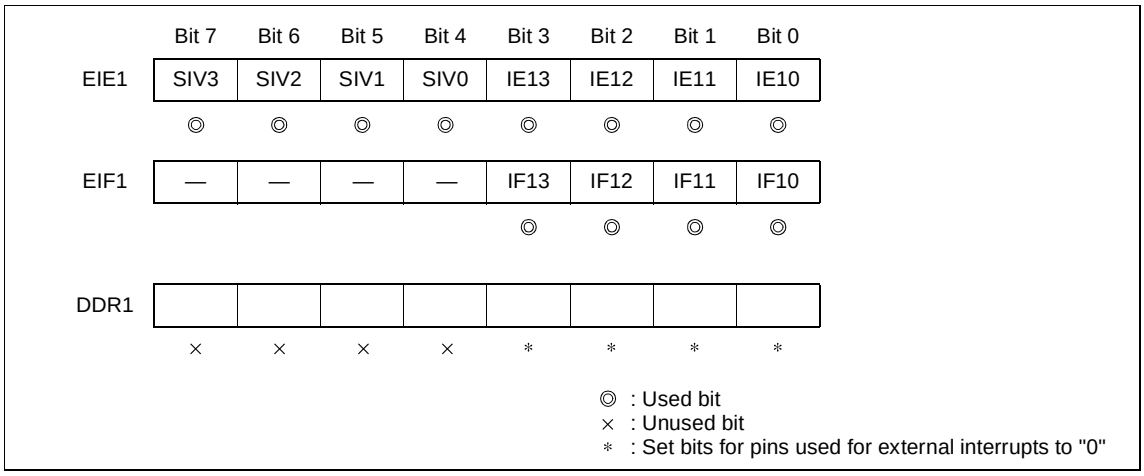
9.5 Operation of External Interrupt Circuit 1

The external interrupt circuit 1 sends an interrupt request to the CPU when it detects a specified edge at one of its external interrupt pins.

■ Operation of External Interrupt Circuit 1

Figure 9.5-1 "External Interrupt Circuit 1 Settings" Settings shows the settings required to operate the external interrupt circuit 1.

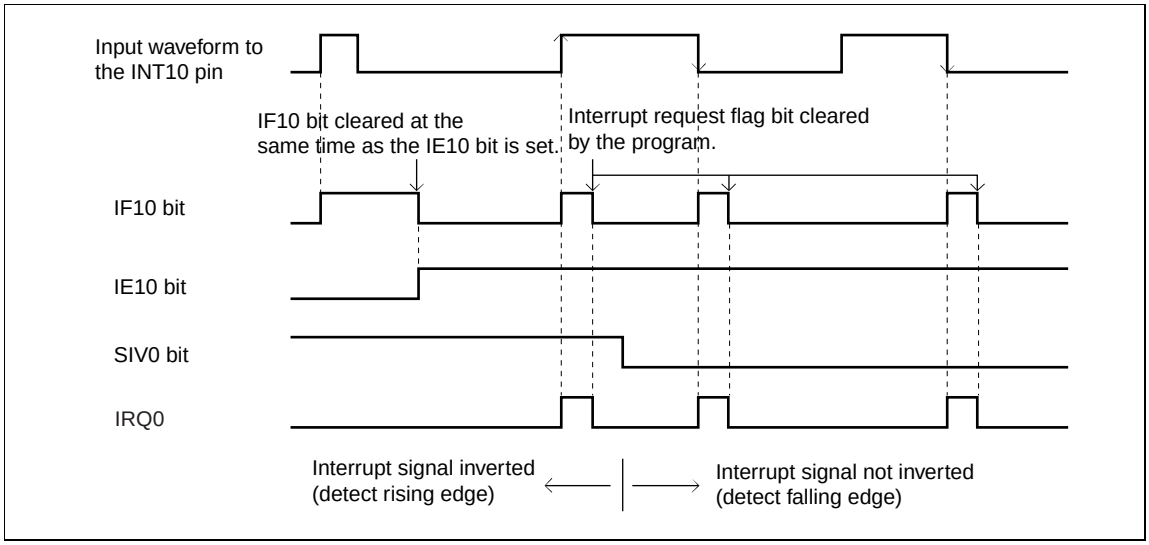
Figure 9.5-1 External Interrupt Circuit 1 Settings



The input signals from the external interrupt pins (INT10 to INT13) are either inverted or not, depending on the state of the applicable external interrupt signal inversion bit (EIE1: SIV0 to SIV3). If the external interrupt enable bit (EIE1: IE10 to IE13) is "1", the corresponding external interrupt request flag bit (EIF1: IF10 to IF13) will be set to "1" when a falling edge is detected in the inverted/non-inverted signal.

Figure 9.5-2 "Operation of External Interrupt 1 (INT10)" shows the external interrupt 1 operation (for signals received at pin INT10).

Figure 9.5-2 Operation of External Interrupt 1 (INT10)



Note:

The pin state can be read directly from the port data register (PDR6), even when used as an external interrupt pins.

9.6 Program Example for External Interrupt Circuit 1

This section gives a program example for the external interrupt circuit 1.

■ Program Example for External Interrupt Circuit 1

❍ Processing description

- Generates interrupts on detecting a falling edge input to the INT10 pin.

❍ Coding Example

```
DDR     EQU     0003H           ; Address of the port 1 data direction register
EIE1    EQU     0030H           ; Address of the external interrupt 1 control
                                  register
EIF1    EQU     0031H           ; Address of the external interrupt 1 flag register

IE10    EQU     EIE1:0          ; Define the external interrupt enable bit.
SIV0    EQU     EIE1:4          ; Define the external interrupt signal inversion
                                  bit.
IF10    EQU     EIE1:0          ; Define the external interrupt request flag bit.

ILR1    EQU     007CH           ; Address of the set interrupt level settings
                                  register

INT_V   DSEG    ABS             ; [DATA SEGMENT]
        ORG     0FFFAH
IRQ0    DW      WARI            ; Set interrupt vector.
INT_V   ENDS

;---------- Main program ----------------------------------------------------------------------------

        CSEG                    ; [CODE SEGMENT]
        :                       ; Stack pointer (SP) etc. are already initialized.
        CLRI                    ; Disable interrupts.
        MOV     ILR1,#11111110B ; Set interrupt priority to level 2.
        MOV     DDR0,#00000000B ; Set P10/INT10 pin as input.
        CLRB    SIV0            ; Select falling edge.
        SETB    IE10            ; Enable INT10 interrupt input.
        CLRB    IF10            ; Clear external interrupt request flag.
        SETI                    ; Enable interrupts.
        :

;---------- Interrupt processing routine ------------------------------------------------------------

WARI    CLRB    IF10            ; 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 10 EXTERNAL INTERRUPT CIRCUIT 2 (LEVEL)

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

10.1 Overview of External Interrupt Circuit 2

The external interrupt circuit 2 detects the level of the signals input to the eight external interrupt pins and generates the interrupt requests to the CPU.

■ External Interrupt Circuit 2 Function (Level Detection)

The external interrupt circuit 2 function detects the signals of the "L" levels input to the external interrupt pins and to generate interrupt request to the CPU. These interrupts can wake up the CPU from standby mode and change the device to the normal operating state (main-run or sub-run mode).

External interrupt pins:	8 pins (P00/$\overline{\text{INT20}}$ to P07/$\overline{\text{INT27}}$)
External interrupt sources:	"L" level signal input to an external interrupt pin.
Interrupt control:	Enables or disables to input external interrupt controlled by external interrupt 2 control register (EIE2)
Interrupt flag:	IRQ flag bit of external interrupt 2 flag register (EIF2). Flag set when there is an IRQ.
Interrupt request:	IRQ4 is generated if any enabled external interrupt pin goes LOW.

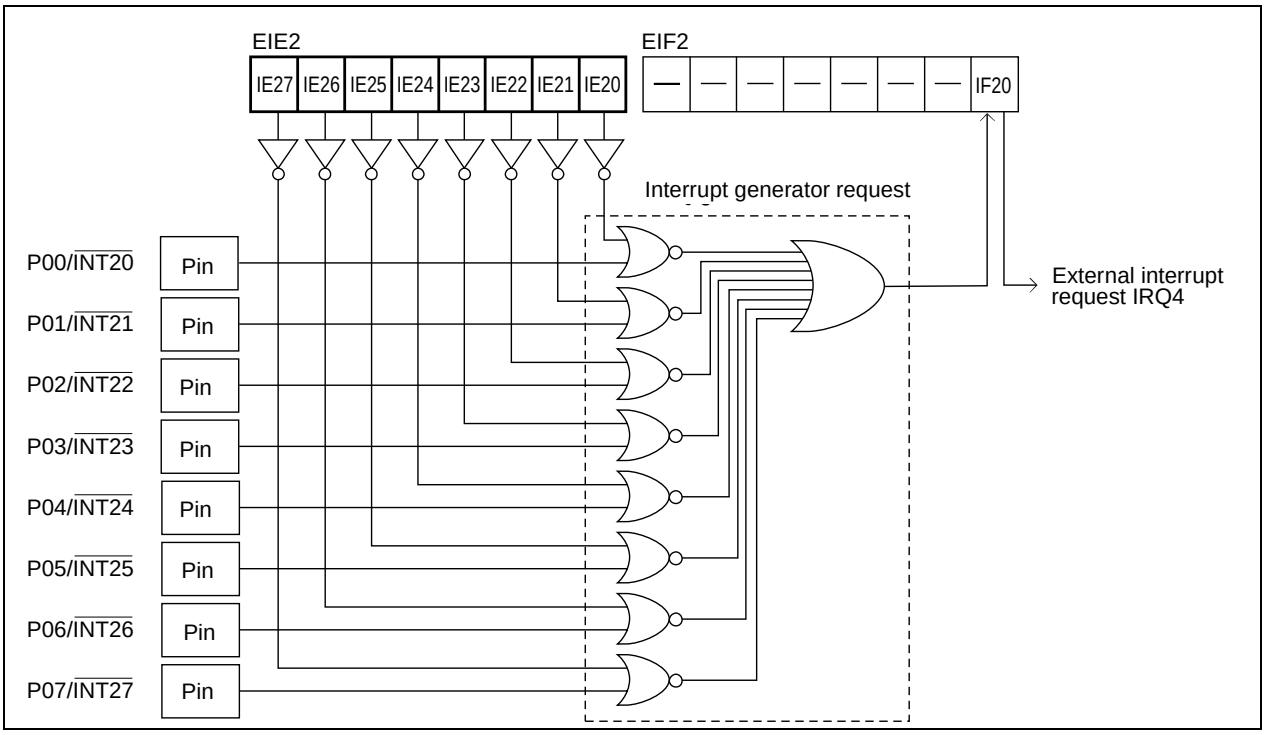
10.2 Block Diagram of External Interrupt Circuit 2

The external interrupt circuit 2 consists of the following three blocks:

- **Interrupt request generator**
- **External interrupt 2 control register (EIE2)**
- **External interrupt 2 flag register (EIF2)**

■ Block Diagram of External Interrupt Circuit 2

Figure 10.2-1 Block Diagram of External Interrupt Circuit 2



❍ Interrupt request generator

The interrupt request generator generates CPU interrupt requests based on signals input at external interrupt pins ($\overline{INT20}$ to $\overline{INT27}$) and the external interrupt enable bits.

❍ EIE2 register

External interrupt input enable bits (IE20 to IE27) enable/disable "L" level signals input at the corresponding external interrupt input pins.

❍ EIF2 register

The interrupt request flag bit of this register (IF20) is used to hold (and clear) interrupt request signals.

10.3 Structure of External Interrupt Circuit 2

This section describes the pins, pin block diagram, registers, and interrupt sources of the external interrupt circuit 2.

■ External Interrupt Circuit 2 Pins

The external interrupt circuit 2 uses eight external interrupt pins.

The external interrupt pins can function either as external interrupt inputs (hysteresis inputs) or as general-purpose I/O ports.

When P00/$\overline{INT20}$ to P07/$\overline{INT27}$ pins are set as inputs in the port 0 data direction register (DDR0), and the corresponding external interrupt inputs are enabled in the external interrupt 2 control register (EIE2) they operate as external interrupt input pins ($\overline{INT20}$ to $\overline{INT27}$). When they are being used as the input port, the pin states can be read from the port data register (PDR0) at any time.

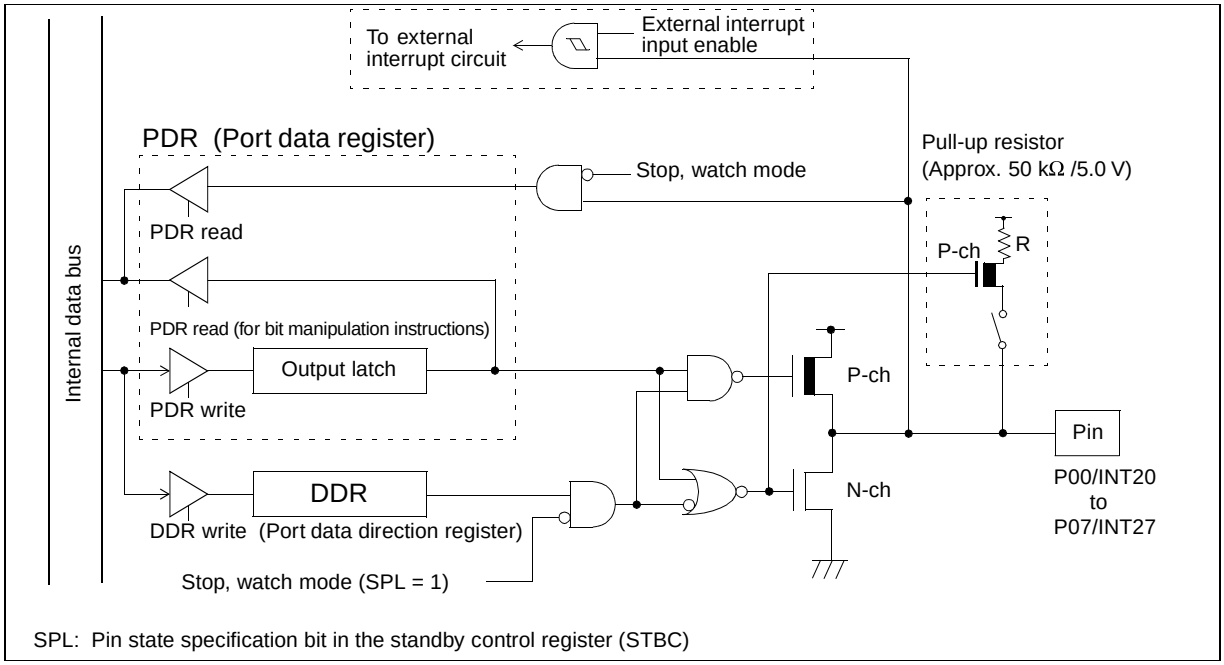
Table 10.3-1 "External Interrupt Circuit 2 Pins" lists the external interrupt circuit 2 pins.

Table 10.3-1 External Interrupt Circuit 2 Pins

External interrupt pin	When used as external interrupt input (Interrupt Input Enabled)	When used as General-purpose I/O port (Interrupt Input Disabled)
P00/$\overline{INT20}$	$\overline{INT20}$ (EIE2:IE20=1, DDR0:bit0=0)	P00 (EIE2:IE20=0)
P01/$\overline{INT21}$	$\overline{INT21}$ (EIE2:IE21=1, DDR1:bit1=0)	P01 (EIE2:IE21=0)
P02/$\overline{INT22}$	$\overline{INT22}$ (EIE2:IE22=1, DDR2:bit2=0)	P02 (EIE2:IE22=0)
P03/$\overline{INT23}$	$\overline{INT23}$ (EIE2:IE23=1, DDR3:bit3=0)	P03 (EIE2:IE23=0)
P04/$\overline{INT24}$	$\overline{INT24}$ (EIE2:IE24=1, DDR4:bit4=0)	P04 (EIE2:IE24=0)
P05/$\overline{INT25}$	$\overline{INT25}$ (EIE2:IE25=1, DDR5:bit5=0)	P05 (EIE2:IE25=0)
P06/$\overline{INT26}$	$\overline{INT26}$ (EIE2:IE26=1, DDR6:bit6=0)	P06 (EIE2:IE26=0)
P07/$\overline{INT27}$	$\overline{INT27}$ (EIE2:IE27=1, DDR7:bit7=0)	P07 (EIE2:IE27=0)

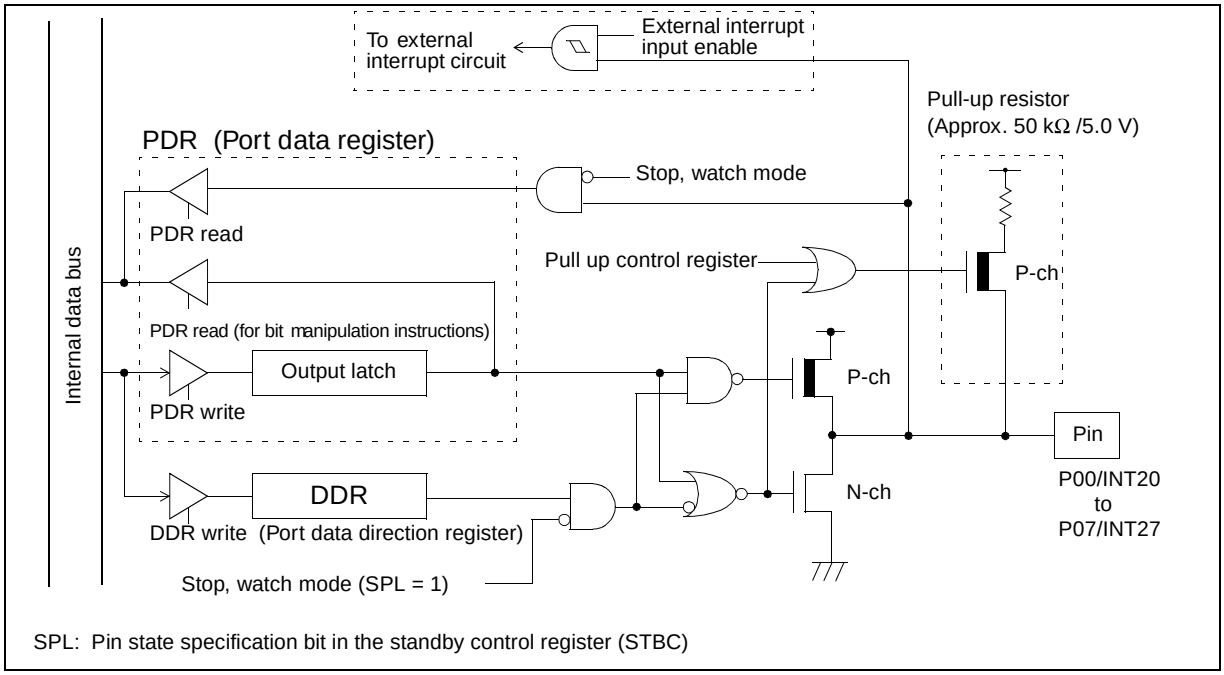
■ Block Diagram of External Interrupt Circuit 2 Pins

Figure 10.3-1 Block Diagram of External Interrupt Circuit 1 Pins for MB89983



SPL: Pin state specification bit in the standby control register (STBC)

Figure 10.3-2 Block Diagram of External Interrupt Circuit 2 Pins for MB89P985 and MB89PV980



SPL: Pin state specification bit in the standby control register (STBC)

Note:

Pins with a pull-up register (optional) go to the "H" level during a reset or in stop and watch mode (SPL = "1").

■ External Interrupt Circuit 2 Registers

Figure 10.3-3 External Interrupt Circuit 2 Registers

EIE2 (External interrupt 2 control register)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0032H	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	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0033H	—	—	—	—	—	—	—	IF20	-------0B
								R/W	

R/W: Readable and writable
— : Unused
X : Indeterminate

■ External Interrupt Circuit 2 Interrupt Sources

IRQ4: IRQ4 is generated if any one of external interrupt pins $\overline{INT20}$ to $\overline{INT27}$ goes to "L" with a "1" in the external interrupt input enable bit for that pin.

10.3.1 External Interrupt 2 Control Register (EIE2)

The external interrupt 2 control register (EIE2) is used to enable/disable input of external interrupt pins ($\overline{INT20}$ to $\overline{INT27}$).

■ **External Interrupt 2 Control Register (EIE2)**

Figure 10.3-4 External Interrupt 2 Control Register (EIE2)

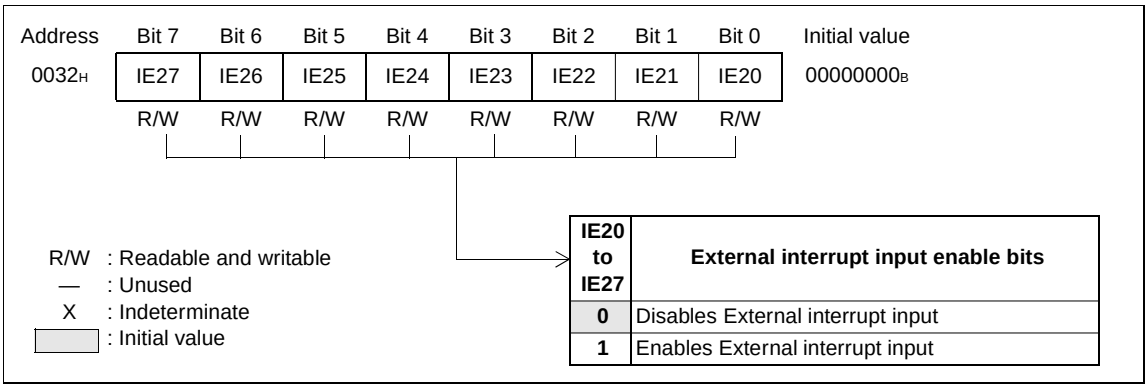


Table 10.3-2 External Interrupt 2 Control Register (EIE2) Bits vs. Pins

	Bit	Pin
Bit 7	IE27	$\overline{INT27}$
Bit 6	IE26	$\overline{INT26}$
Bit 5	IE25	$\overline{INT25}$
Bit 4	IE24	$\overline{INT24}$
Bit 3	IE23	$\overline{INT23}$
Bit 2	IE22	$\overline{INT22}$
Bit 1	IE21	$\overline{INT21}$
Bit 0	IE20	$\overline{INT20}$

Note:

For MB89P985 and MB89PV980, there will be current leakage through the pull-up resistor in stop mode when the pull-up resistor is enabled and any enabled external interrupt 2 pin is input "0". To prevent the current leakage, the pull-up resistor should be disabled for each enabled interrupt pin before going into stop mode.

Table 10.3-3 External Interrupt 1 Control Register (EIE1) Bits

Bit		Function
Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0	IE27 to IE20: External Interrupt Enable Bits	• These bits enable/disable input of external interrupts at external interrupt pins $\overline{INT20}$ to $\overline{INT27}$. • Setting these bits to "1" puts the corresponding pin into its external interrupt input mode, and enables input of external interrupts at the pin. • Conversely, a "0" in the bit allows the pin to function in its general-purpose port mode and inhibits input of interrupts at the pin. **Note:** When using a pin for external interrupts, set it as an input by writing its bit in the port 0 data direction register (DDR0) to "0". The state of the pin can always be read directly out of the port 0 data register (PDR0) regardless of the sate of this external interrupt enable bit.

10.3.2 External Interrupt 2 Flag Register (EIF2)

External Interrupt 2 flag register (EIF2) is used to hold the IRQ state when a level interrupt has been detected, and clear the interrupt.

■ External Interrupt 2 Flag Register (EIF2)

Figure 10.3-5 External Interrupt 2 Flag Register (EIF2)

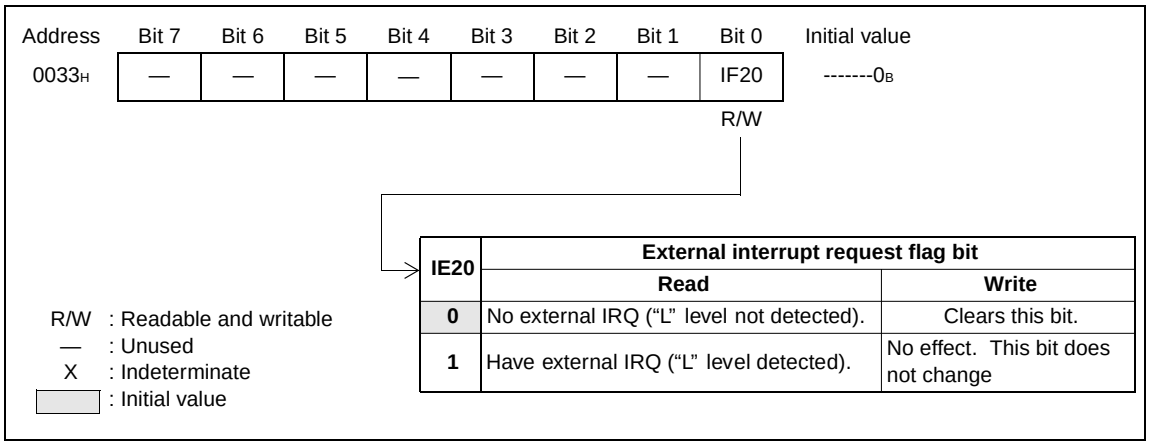


Table 10.3-4 External Interrupt 1 Control Register (EIE1) Bits

Bit		Function
Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1	Unused Bits	• The read value is indeterminate. • Writing to this bit has no effect on the option.
Bit 0	IF20: External interrupt request flag bit	• This bit is set to "1" when a "L" is detected at an enabled external input pin ($\overline{INT20}$ to $\overline{INT27}$). • Writing "0" clears this bit. Writing "1" has no effect and does not change the bit value. **Note:** Writing "0" to the external interrupt enable bits of the external interrupt 2 control register (EIE2: IE20 to IE27) simply disables the corresponding external interrupt input; it does not clear the interrupt request. IRQ4 will continue to be sent to the CPU until it is cleared by writing "0" to the IF20 bit.

10.4 External Interrupt Circuit 2 Interrupt

The external interrupt circuit 2 interrupt trigger event is the detection of a "L" level at the external interrupt pin.

■ Interrupts for External Interrupt Circuit 2 Operation

If a "L" is detected at an enabled external interrupt pin, the external interrupt request flag bit (EIF2: IF20) is set to "1", and an interrupt request (IRQ4) to the CPU is generated. Write "0" to the IF20 bit in the interrupt processing routine to clear the interrupt request.

Once the external interrupt request flag bit (IF20) is set to "1", IRQ4 continues to be asserted as long as the flag set. Disabling the interrupt input by writing the IE bit (IE20 to IE27) of the EIE2 register to "0" will not clear the interrupt request. Always clear the IF20 bit.

Also, if the external interrupt pin stays "L", writing "0" to the IF20 bit without disabling the external interrupt input will not clear the interrupt either, because IF20 will immediately be set again by the "L" pin. After an interrupt request is generated, then, either the input must be disabled, or the external IRQ signal de-asserted.

Check:

When enabling interrupts of CPU after wake-up from a reset, clear the IF20 bit in advance.

Note:

Wake-up from stop mode by an interrupt is possible using only the external interrupt circuit 1 and 2.

■ Register and Vector Table for External Interrupt Circuit 2 Interrupts

Table 10.4-1 Registers and Vector Table for External Interrupt Circuit 2 Interrupts

Interrupt	Interrupt level setting register			Vector table address	
	Register	Setting bits		Upper	Lower
IRQ4	ILR2 ($007D_H$)	L41 (Bit 1)	L40 (Bit 0)	$FFF2_H$	$FFF3_H$

Reference:

See Section 3.4.2 "Interrupt Processing" for details on the interrupts operation.

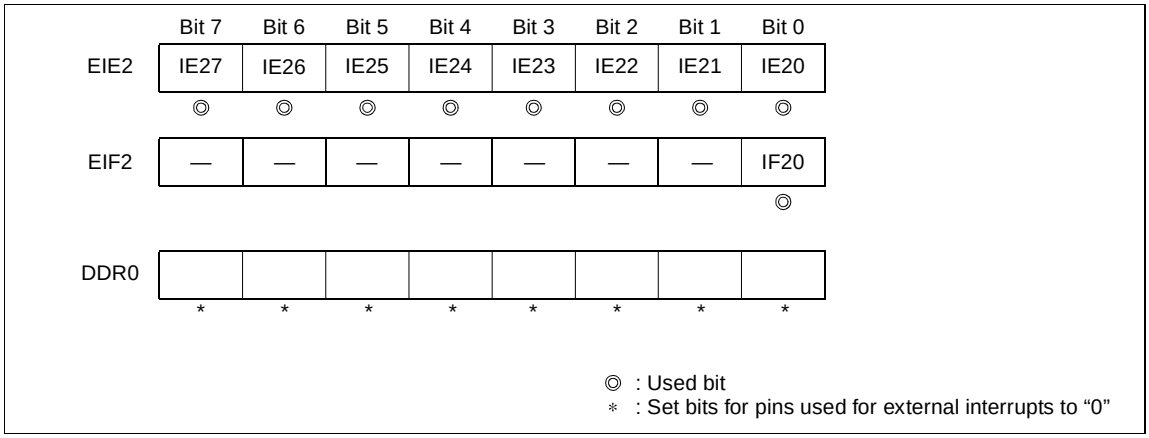
10.5 Operation of External Interrupt Circuit 2

The external interrupt circuit 2 sends an interrupt request to the CPU when it detects a "L" at one of its external interrupt pins.

■ Operation of External Interrupt Circuit 2

Figure 10.5-1 "External Interrupt Circuit 2 Settings" shows the settings required to operate the external interrupt circuit 2.

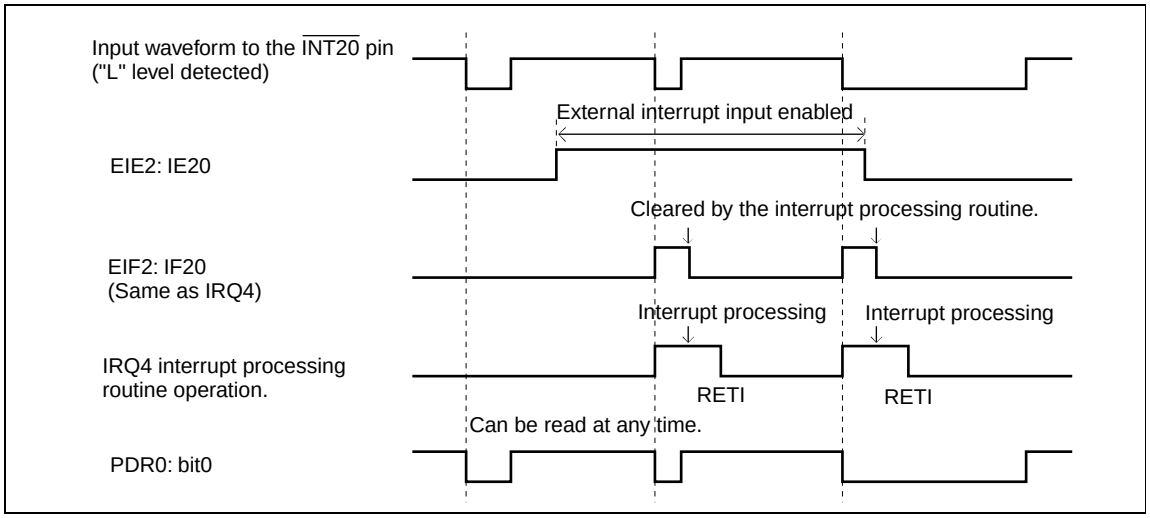
Figure 10.5-1 External Interrupt Circuit 2 Settings



If a "L" is applied to one of the $\overline{\text{INT20}}$ to $\overline{\text{INT27}}$ pins with the corresponding external interrupt input enable bit (IE20 to IE27) in the "enable" state, the circuit sends an IRQ4 interrupt request to the CPU.

Figure 10.5-2 "Operation of External Interrupt 2 (INT20)" shows external interrupt circuit 2 operation (for a signal received at INT20).

Figure 10.5-2 Operation of External Interrupt 2 ($\overline{\text{INT20}}$)



Note:

The pin state can be read directly from the port data register (PDR0) even when the pin is being used as an external interrupt input.

10.6 Program Example for External Interrupt Circuit 2

This section gives a program example for the external interrupt circuit 2.

■ Program Example for External Interrupt Circuit 2

❍ Processing description

- Generates interrupts on detecting a "L" input the $\overline{\text{INT20}}$ pin.

❍ Coding example

```
DDR0    EQU     0001H           ; Address of port 0 data direction register
EIE2    EQU     0032H           ; Address of external interrupt 2 control register
EIF2    EQU     0033H           ; Address of external interrupt 2 flag register

IF20    EQU     EIF2:0          ; Define the external interrupt request flag bit.

ILR2    EQU     007DH           ; Address of the set interrupt level setting
                                  register 2

INT_V   DSEG    ABS             ; [DATA SEGMENT]
        ORG     0FFF2H
IRQ4    DW      WARI            ; Set interrupt vector.
INT_V   ENDS

;---------- Main program ------------------------------------------------------

CSEG                            ; [CODE SEGMENT]
                                ; Stack pointer (SP) etc. are already initialized.
        :                       ;
        CLRI                    ; Disable interrupts.
        CLRB    IF20            ; Clear external interrupt request flag.
        MOV     ILR2,#11111110B ; Set interrupt priority (level 2).
        MOV     DDR0,#00000000B ; Set P00/INT20 pin as input.
        MOV     EIE2,#00000001B ; Enable external interrupt input at INT20 pin
        SETI                    ; Enable interrupts.
        :

;---------- Interrupt processing routine --------------------------------------

WARI    MOV     EIE2,#00000000B ; Disable external interrupt input at INT20 pin.
        CLRB    IF20            ; Clear external interrupt request flag.
        PUSHW   A
        XCHW    A,T
        PUSHW

        :
        User processing
        :
```

```
        POPW    A
        XCHW    A,T
        POPW    A
        RETI
        ENDS

;-----------------------------------------------------------------------------------------------------------------------

        END
```

CHAPTER 11 A/D CONVERTER

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

11.1 Overview of A/D Converter

The A/D converter can be selected to function either as an 8-bit successive approximation type A/D conversion or as a sense function. The sense function performs a high-speed comparison between the input voltage and a set voltage. Both functions select one input signal from the eight analog input pin channels and can be activated either by software, by an internal clock, or by an 8/16-bit timer output.

■ A/D Conversion Function

The A/D conversion function converts the analog voltage (input voltage) input to an analog input pin to an 8-bit digital value.

- Selects one input from eight analog input pins.
- Conversion speed is 44 instruction cycles (41.9 µs with highest main clock and 4.2 MHz source oscillation).
- Generates an interrupt when A/D conversion completes.
- Conversion completion can also be determined by software.

The following methods are available to activate A/D conversion:

- Activation by software
- Continuous activation by a timebase timer output (divide-by-2^8 main clock source oscillation)
- Continuous activation by an 8/16-bit timer/counter output.

■ Sense Function

The sense function compares the analog voltage (input voltage) input to an analog input pin with the voltage (compare voltage) corresponding to the value set in the A/D data register (ADCD), and determines which voltage is higher or lower.

- Selects one input from eight analog input pins.
- Compare speed is 12 instruction cycles (11.4 µs in main clock mode, with the highest clock speed selected, and a clock oscillator frequency of 4.2 MHz).
- Generates an interrupt when the comparison condition is satisfied.

The following methods are available to activate the sense function:

- Activation by software
- Continuous activation by a timebase timer output (divide-by-2^8 main clock source oscillation)
- Continuous activation by an 8/16-bit timer/counter output.

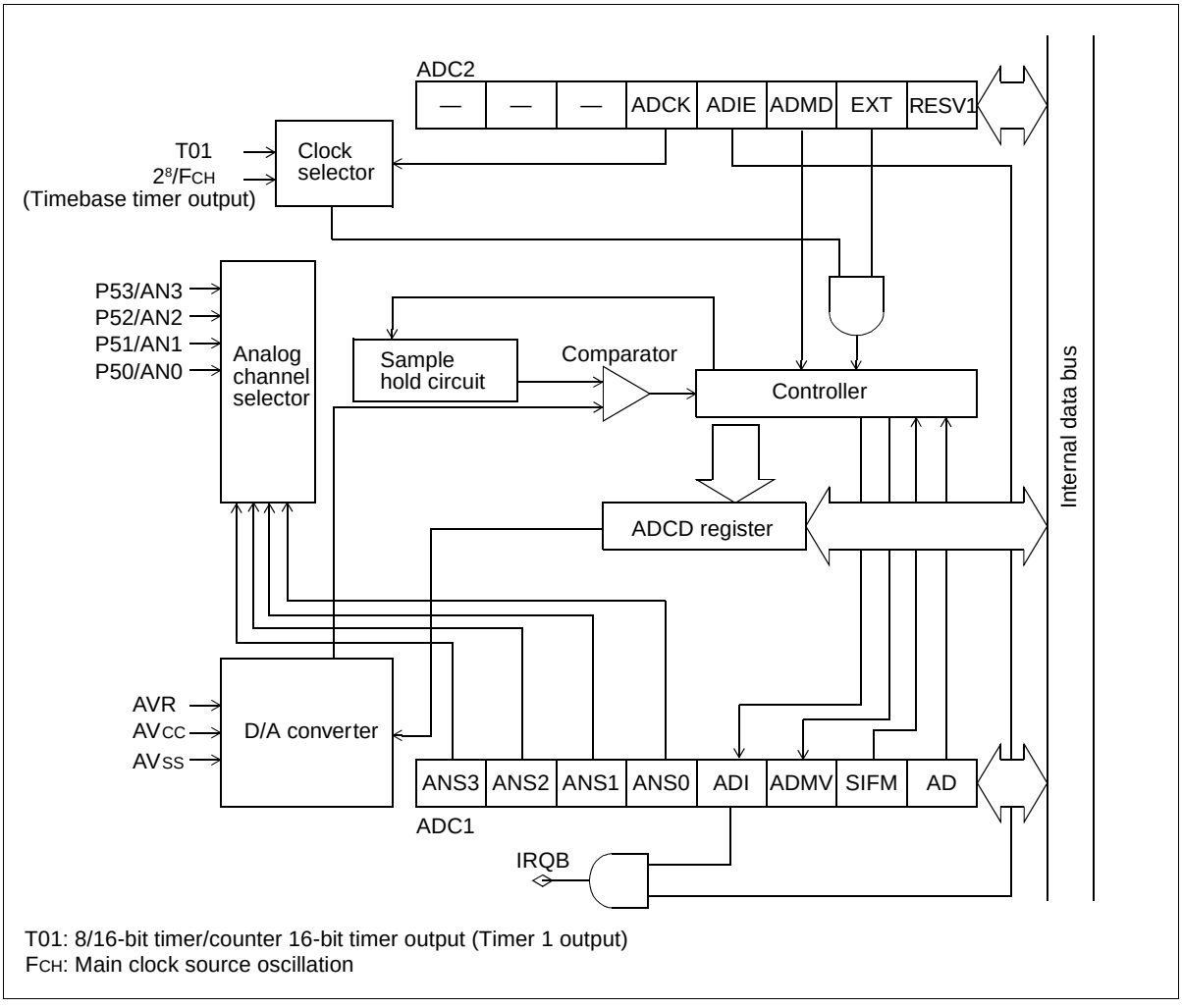
11.2 Block Diagram of A/D Converter

The A/D converter consists of the following nine blocks:

- **Clock selector (input clock selector for A/D converter activation)**
- **Analog channel selector**
- **Sample hold circuit**
- **D/A converter**
- **Comparator**
- **Controller**
- **A/D data register (ADCD)**
- **A/D control register 1 (ADC1)**
- **A/D control register 2 (ADC2)**

■ Block Diagram of A/D Converter

Figure 11.2-1 Block Diagram of A/D Converter



T01: 8/16-bit timer/counter 16-bit timer output (Timer 1 output)
FCH: Main clock source oscillation

❍ **Clock selector**

Selects the clock used to activate the A/D conversion or sense function when continuous activation is enabled (ADC2: EXT = "1").

❍ **Analog channel selector**

Selects one of the eight analog input channels.

❍ **Sample hold circuit**

Holds the input voltage selected by the analog channel selector. The circuit samples and holds the input voltage immediately after the A/D conversion or sense function is activated. This allows A/D conversion (or comparison) to proceed without being affected by input voltage fluctuation.

❍ **D/A converter**

Generates the voltage corresponding to the value set in the ADCD register.

❍ **Comparator**

Compares the sampled and held input voltage with the output voltage of the D/A converter, and determines which voltage is higher or lower.

❍ **Controller**

The controller has two functions:

- For the A/D conversion function, the controller successively determines the value of each bit of the ADCD register, starting from the most significant bit and proceeding to the least significant bit, based on the greater-than/less-than signal from the comparator. When conversion is complete, the circuit sets the interrupt request flag bit (ADC1: ADI).
- For the sense function, the controller sets the interrupt request flag bit (ADI) if the greater-than/less-than signal from the comparator matches the compare condition setting bit (SIFM) in the ADC1 register.

❍ **ADCD register**

The ADCD register has two functions:

- Stores the A/D conversion result for the A/D conversion function.
- For the sense function, the data for the voltage that is compared with the input voltage is written to this register.

❍ **ADC1 register**

The ADC1 register is used to enable or disable each function, select the analog input pin, check statuses, and control interrupts.

❍ **ADC2 register**

The ADC2 register is used to select the input clock, enable or disable interrupts, and select functions.

■ A/D Converter Power Supply Voltage

❍ AV_{CC}

The A/D converter power supply pin. Use at the same voltage as V_{CC}. When high A/D conversion resolution is required, take measures to ensure that the noise on V_{CC} is not present on AV_{CC}, or use a separate power supply. Connect this pin to the power supply, even if the A/D converter is not used.

❍ AV_{SS}

The A/D converter ground pin. Use at the same voltage as V_{SS}. When high A/D conversion accuracy is required, take measures to ensure that the noise on V_{SS} is not present on AV_{SS}. Connect this pin to ground (GND), even if the A/D converter is not used.

❍ AVR

Reference voltage input pin for the A/D converter. The A/D converter performs 8-bit A/D conversion between AVR and AV_{SS}.

Connect to AV_{SS} if the A/D converter is not used.

11.3 Structure of A/D Converter

This section describes the pins, pin block diagrams, registers, and an interrupt source for the A/D converter.

■ A/D Converter Pins

The A/D converter function uses the P50/AN0 to P53/AN3 pins. These pins can function as either output-only ports of the N-ch open-drain outputs (P50 to P53), as the analog input pins (AN0 to AN3).

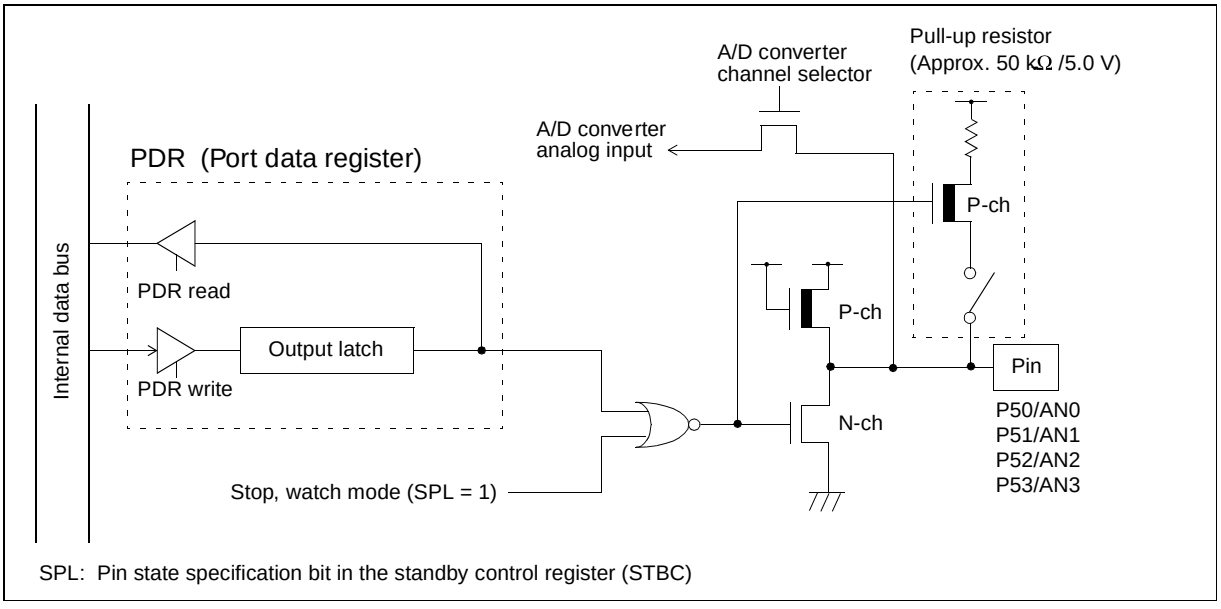
AN0 to AN3:

The analog voltages to be converted (A/D conversion function) or compared (sense function) are applied to these pins.

To select the analog input function for one of these pins, you set the corresponding bit of the port data register (PDR5) to "1", to turn off the port output transistor, then set the analog input channel select bits (ADC1: ANS0 to ANS3) to select the pin as the analog input channel. Pins that are not needed for analog inputs can still be used as output port pins, even while the A/D converter is being used.

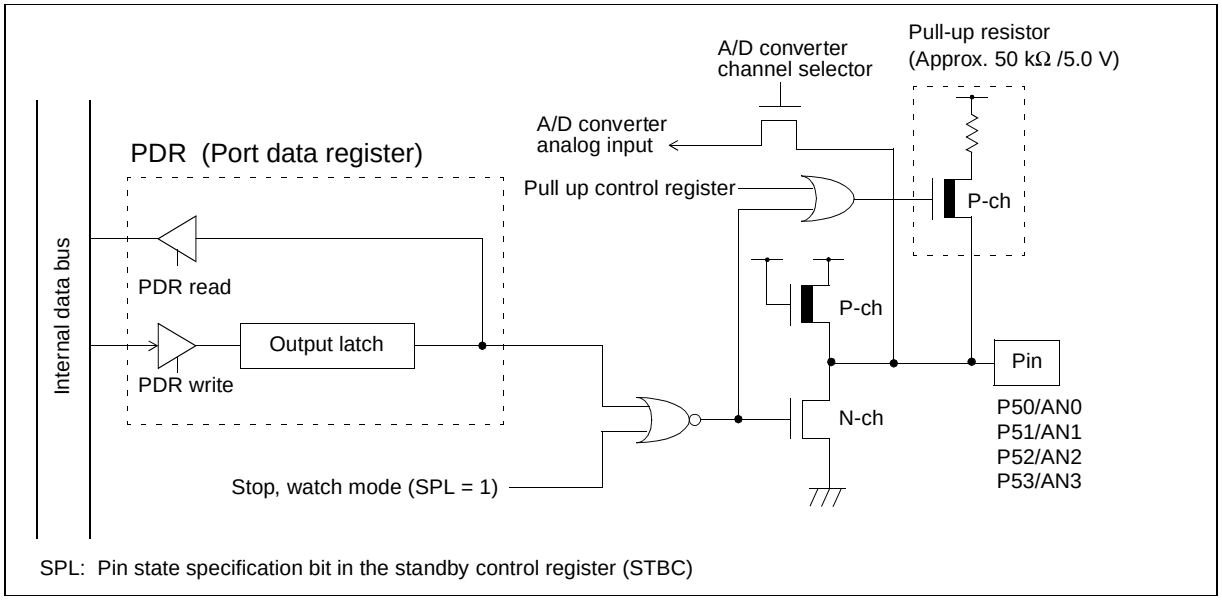
■ Block Diagram of A/D Converter Pin

Figure 11.3-1 Block Diagram of P53/AN3 to P50/AN0 Pins for MB89983



SPL: Pin state specification bit in the standby control register (STBC)

Figure 11.3-2 Block Diagram of P53/AN3 to P50/AN0 Pins for MB89P985 and MB89PV980



SPL: Pin state specification bit in the standby control register (STBC)

Check:

If using the A/D converter, do not set a pull-up resistor for any of P53/AN3 to P50/AN0.

Check:

Do not use the pins as output ports if using as an analog input.

■ A/D Converter Registers

Figure 11.3-3 A/D Converter Registers

ADC1 (A/D control register 1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
002DH	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	SIFM	AD	00000000B
	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	

ADC2 (A/D control register 2)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
002EH	—	—	—	ADCK	ADIE	ADMD	EXT	RESV1	---00001B
				R/W	R/W	R/W	R/W	R/W	

ADCD (A/D data register)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
002FH									XXXXXXXXB
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable and writable
R : Read-only
— : Unused
X : Indeterminate

■ A/D Converter Interrupt Source

IRQB: The A/D converter generates an interrupt request if an interrupt request output is enabled (ADC2: ADIE = "1") when A/D conversion completes or the sense function detects the specified condition.

11.3.1 A/D Control Register 1 (ADC1)

A/D control register 1 (ADC1) is used to enable or disable the functions, select the analog input pin, and check the state of the A/D converter.

■ A/D Control Register 1 (ADC1)

Figure 11.3-4 A/D Control Register 1 (ADC1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
002DH	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	SIFM	AD	00000000B
	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	

AD	A/D converter activation bit
	Only applies when software activation is specified (ADC2: EXT = "0").
0	Does not activate the A/D conversion or sense function.
1	Activates the A/D conversion or sense function.

SIFM	Compare condition setting bit
	Only applies when the sense function is selected (ADC2: ADMD = "1")
0	Sets the interrupt request flag bit when the input voltage is less than the compare voltage.
1	Sets the interrupt request flag bit when the input voltage is greater than the compare voltage.

ADMV	Conversion-in-progress flag bit
0	Conversion or comparison not currently in progress.
1	Conversion or comparison in progress.

ADI	Interrupt request flag bit		
	Read		Write
	For the A/D conversion function	For the sense function	
0	Conversion not complete.	Specified condition has not occurred.	Clears this bit.
1	Conversion completes.	Specified condition occurred.	No effect. The bit does not change.

ANS3	ANS2	ANS1	ANS0	Analog input channel selection bits
0	0	0	0	AN0 pin
0	0	0	1	AN1 pin
0	0	1	0	AN2 pin
0	0	1	1	AN3 pin
0	1	0	0	Not available
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	×	×	×	

R/W : Readable and writable
R : Read-only
□ : Initial value

Table 11.3-1 A/D Control Register 1 (ADC1) Bitss

Bit		Function
Bit 7 Bit 6 Bit 5 Bit 4	ANS3 to ANS0: Analog input channel selection bits	These bits select which of the AN0 to AN3 pins to use as the analog input pin. When using software activation (ADC2: EXT = "0"), these bits can be modified to at the same time as activating the A/D conversion or sense function (AD = "1"). **Check:** Always set ANS3 to "0". If ANS3 is "1", no pin is selected as the analog input pin. Also, disable general-purpose port output corresponding to the analog input pin. Do not modify these bits when the ADMV bit is set to "1". **Note:** Pins not used as analog inputs can be used as general-purpose ports.
Bit 3	ADI: Interrupt request flag bit	• For the A/D conversion function: This bit is set to "1" when the A/D conversion is completed. • For the sense function: This bit is set to "1" when the input voltage satisfies the condition set in the compare condition setting bit (SIFM). • An interrupt request is output for either function when both this bit and the interrupt request enable bit (ADC2: ADIE) are "1". • Writing "0" clears this bit. Writing "1" has no effect and does not change the bit value.
Bit 2	ADMV: Conversion-in-progress flag bit	This bit indicates whether or not the A/D conversion function is currently performing a conversion or the sense function is currently performing a voltage comparison. The bit is set to "1" when a conversion or comparison is in progress. **Note:** This bit is read-only. The write value has no meaning and has no effect on the operation.
Bit 1	SIFM: Compare condition setting bit	• This bit has no meaning for the A/D conversion function. • For the sense function: This bit sets the comparison condition for the input voltage and compare voltage that generates an interrupt source. An interrupt request is generated (ADI = "1") when the input voltage is less than the compare voltage if the bit is "0", and when the input voltage is greater than the compare voltage if the bit is "1". No interrupt request is generated if the input voltage and compare voltage are equal. • When using software activation (ADC2: EXT = "0"), this bit can be modified to at the same time as starting the sense function (AD = "1"). **Check:** Do not modify these bits when the ADMV bit is set to "1".

Table 11.3-1 A/D Control Register 1 (ADC1) Bitss

Bit		Function
Bit 0	IE13 to IE10: External Interrupt Enable Bits	• This bit activates the A/D conversion or sense function by software. • Writing "1" to this bit activates the A/D conversion or sense function when continuous activation is not specified (ADC2: EXT = "0"). **Check:** Writing "0" to this bit does not stop the A/D conversion or sense function. The read value is always "0". This bit has no meaning when continuous activation is specified.

11.3.2 A/D Control Register 2 (ADC2)

A/D control register 2 (ADC2) is used to select the A/D converter functions, select the input clock, enable or disable interrupts and continuous activation, and check the state of the A/D converter.

■ A/D Control Register 2 (ADC2)

Figure 11.3-5 A/D Control Register 2 (ADC2)

Address	Bit 7	Bit 6	Bit5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
002EH	—	—	—	ADCK	ADIE	ADMD	EXT	RESV	---00001B
				R/W	R/W	R/W	R/W	R/W	

RESV1	Reserved bit
	Always write "1" to this bit.

EXT	Continuous activation enable bit
0	Activates by the AD bit in the ADC1 register.
1	Activates continuously by the clock selected in the ADCK bit.

ADMD	Function selection bit
0	A/D conversion function
1	Sense function

ADIE	Interrupt request enable bit
0	Disables interrupt request output.
1	Enables interrupt request output.

ADCK	Input clock selection bit
	Only applies when continuous activation is enabled (EXT = "1").
0	Timebase timer output (divide-by-2^8 source oscillation)
1	8/16-bit timer 16-bit time output

R/W : Readable and wrtiable
— : Unused
X : Indeterminate
□ : Initial value

Table 11.3-2 A/D Control Register 1 (ADC1) Bitss

Bit		Function
Bit 7 Bit 6 Bit 5	Unused bits	• The read value is indeterminate. • Writing to these bits has no effect on the operation.
Bit 4	ADCK: Input clock selection bit	• This bit selects the input clock used to activate the A/D conversion or sense function when continuous activation is specified (EXT = "1"). Setting this bit to "0" selects the timebase timer output (divide=by=2^8 main clock source oscillation). Setting this bit to "1" selects the 16-bit timer output (TO1) in 8/16 timer/counter. **Check:** In the subclock mode, the main clock oscillator is stopped, which means that the timebase timer output cannot be used to trigger continuous mode conversions/comparisons.
Bit 3	ADIE: Interrupt request enable bit	• This bit enables or disables an interrupt request output to the CPU. • An interrupt request is output when both this bit and the interrupt request flag bit (ADC1: ADI) are "1".
Bit 2	ADMD: Function selection bit	• This bit switches between the A/D conversion function and sense function. • The A/D converter operates as the A/D conversion function when this bit is set to "0" and as the sense function when this bit is set to "1". **Check:** Do not modify this bit when the conversion-in-progress bit (ADC1: ADMV) is set to "1". Also, clear the interrupt request flag bit (ADC1: ADI = "0") at switching functions.
Bit 1	EXT: Continuous activation enable bit	• This bit selects whether to activate the A/D conversion and sense functions by software or to operate continuously synchronized with an input clock. • Setting this bit to "0" enables software activation by the A/D converter activation bit (ADC1: AD). Setting this bit to "1" enables continuous activation on the rising edge of the clock selected in the input clock selection bit (ADC2: ADCK).
Bit 0	RESV1: Reserved bit	**Check:** Always write "1" to this bit. The read value is always "1".

11.3.3 A/D Data Register (ADCD)

The A/D data register stores the A/D conversion result for the A/D conversion function.

■ A/D Data Register (ADCD)

Figure 11.3-6 A/D Data Register (ADCD)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
002FH									XXXXXXXXB
	R	R	R	R	R	R	R	R	For the A/D conversion function
	W	W	W	W	W	W	W	W	For the sense function

R: Read-only
W: Write-only
—: Unused
X: Indeterminate

❍ **For A/D conversion function**

The conversion result is decided approximately 44 instruction cycles after A/D conversion is activated. The data of conversion is stored in this register. The value of the register is indeterminate while A/D conversion is in progress. The register is read-only for the A/D conversion function.

❍ **For sense function**

Before activating the sense function, set the data corresponding to the voltage to be compared (compare voltage).

As the register is write-only when the sense function is selected, bit manipulation instructions cannot be used. Confirm operation stopped (ADC2: EXT = "0", ADC1: ADMV = "0") before writing to this register.

■ Example of ADCD Register Setting for Sense Function

Table 11.3-3 Example of ADCD Register Setting for Sense Function

	Compare voltage (V)					
	5.0	**4.0**	**3.0**	**2.0**	**1.0**	**0.0**
ACDC register set value	FF_H	CD_H	$9A_H$	66_H	33_H	00_H

Condition: AV_{CC} = AVR = 5.0 V, AV_{SS} = 0.0 V

11.4 A/D Converter Interrupts

The A/D converter has the following two interrupts:

- **Conversion completion for the A/D conversion function**
- **Match of the input voltage and the comparison condition**

■ **Interrupt for A/D Conversion Function**

When A/D conversion completes, the interrupt request flag bit (ADC1: ADI) is set to "1". At this time, an interrupt request (IRQB) to the CPU is generated if the interrupt request enable bit is enabled (ADC2: ADIE = "1"). Write "0" to the ADI bit in the interrupt processing routine to clear the interrupt request.

The ADI bit is set after completion of A/D conversion, regardless of the ADIE bit value.

Note:

An interrupt request is generated immediately if the ADI bit is "1" when the ADIE bit is changed from disabled to enabled ("0" --> "1").

■ **Interrupt for Sense Function**

When the specified comparison condition is satisfied after completion of comparison of the input voltage and compare voltage, the interrupt request flag bit (ADC1: ADI) is set to "1".

At this time, an interrupt request (IRQB) to the CPU is generated if the interrupt request enable bit is enabled (ADC2: ADIE = "1"). Write "0" to the ADI bit in the interrupt processing routine to clear the interrupt request.

The ADI bit is set when the comparison condition is satisfied, regardless of the ADIE bit value.

Note:

An interrupt request is generated immediately if the ADI bit is "1" when the ADIE bit is changed from disabled to enabled ("0" --> "1").

■ **Register and Vector Table for A/D Converter Interrupt**

Table 11.4-1 Register and Vector Table for A/D Converter Interrupt

Interrupt	Interrupt level settings register			Vector table address	
	Register	Setting bits		Upper	Lower
IRQB	ILR3 ($007E_H$)	LB1 (Bit 7)	LB0 (Bit 6)	$FFE4_H$	$FFE5_H$

Reference:

See Section 3.4.2 "Interrupt Processing" for details on the operation of interrupt.

11.5 Operation of A/D Converter

The A/D conversion and sense functions of the A/D converter can be activated by software or can be activated continuously.

■ Activating A/D Conversion Function

❍ Software activation

Figure 11.5-1 "A/D Conversion Function (Software Activation) Settings" shows the settings required for software activation of the A/D conversion function.

Figure 11.5-1 A/D Conversion Function (Software Activation) Settings

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADC1	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	SIFM	AD
	0	◎	◎	◎	◎	◎	×	1
ADC2	—	—	—	ADCK	ADIE	ADMD	EXT	RESV
				×	◎	0	0	1
ADCD	Stores the A/D conversion result.							

◎ : Used bit
× : Unused bit
1 : Set "1".
0 : Set "0".

On activation, the A/D converter starts the operation of the A/D conversion function. The A/D conversion function can be reactivated while conversion is in progress.

❍ Continuous activation

Figure 11.5-2 "A/D Conversion Function (Continuous Activation) Settings" shows the settings required for continuous activation of the A/D conversion function.

Figure 11.5-2 A/D Conversion Function (Continuous Activation) Settings

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADC1	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	SIFM	AD
	0	◎	◎	◎	◎	◎	×	×
ADC2	—	—	—	ADCK	ADIE	ADMD	EXT	RESV
				◎	◎	0	1	1
ADCD	Stores the A/D conversion value.							

◎ : Used bit
× : Unused bit
1 : Set "1".
0 : Set "0".

When continuous activation is enabled, the rising edge of the selected input clock activates the A/D conversion, starting operation of the A/D conversion function. When continuous activation is disabled (ADC2: EXT = "0"), continuous activation halts but software activation is available.

■ Operation of A/D Conversion Function

The following describes the operation of the A/D converter. From activation to completion of A/D conversion requires approximately 44 instruction cycles.

1. On activation, A/D conversion sets the conversion-in-progress flag bit (ADC1: ADMV = "1") and connects the sample hold circuit to the specified analog input pin.
2. The internal sample hold capacitor captures the voltage at the analog input pin for approximately 8 instruction cycles. The capacitor holds the voltage until the A/D conversion completes.
3. The comparator compares the voltage captured by the sample hold capacitor with the A/D converter reference voltage starting from the most significant bit (MSB) and ending with the least significant bit (LSB), and transfers each bit sequentially to the ADCD register.
4. When the complete result has been transferred to the ADCD register, the conversion-in-progress flag bit is cleared (ADC1: ADMV = "0") and the interrupt request flag bit is set (ADC1: ADI = "1").

■ Activating Sense Function

❍ Software activation

Figure 11.5-3 "Sense Function (Software Activation) Settings" shows the settings required for software activation of the sense function.

Figure 11.5-3 Sense Function (Software Activation) Settings

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADC1	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	SIFM	AD
	0	◎	◎	◎	◎	◎	◎	1
ADC2	—	—	—	ADCK	ADIE	ADMD	EXT	RESV
				×	◎	1	0	1
ADCD	Sets the compare voltage.							

◎: Used bit
×: Unused bit
1 : Set "1".
0 : Set "0".

On activation the sense function starts the operation of the sense function.

❍ Continuous activation

Figure 11.5-4 "Sense Function (Continuous Activation) Settings" shows the settings required for continuous activation of the sense function.

Figure 11.5-4 Sense Function (Continuous Activation) Settings

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADC1	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	SIFM	AD
	0	◎	◎	◎	◎	◎	◎	×
ADC2	—	—	—	ADCK	ADIE	ADMD	EXT	RESV
				◎	◎	1	1	1
ADCD	Sets the compare voltage.							

◎: Used bit
×: Unused bit
1 : Set "1".
0 : Set "0".

When continuous activation is enabled, the rising edge of the selected input clock activates the

sense function, starting operation of the sense function. When continuous activation is disabled (ADC2: EXT = "0"), continuous activation stops but software activation is available.

■ Operation of Sense Function

The following describes the operation of the sense function. From activation to completion of the sense function requires approximately 12 instruction cycles.

1. On activation, the sense function sets the conversion-in-progress flag bit (ADC1: ADMV = "1") and connects the sample hold circuit to the specified analog input pin.
2. The internal sample hold capacitor captures the voltage at the analog input pin for approximately 8 instruction cycles. The capacitor holds the voltage until the comparison completes.
3. The comparator compares the voltage captured by the sample hold capacitor with the voltage corresponding to the value set in the ADCD register.
4. When voltage comparison completes, the interrupt request flag bit is set (ADC1: ADI = "1") if the input voltage matches the condition specified by the compare condition setting bit (ADC1: SIFM). The ADI bit does not change if the input voltage does not match the specified condition or if the input voltage and set voltage are equal.

Note:

For the sense function, an interrupt request is not generated when comparison completes if the comparison condition is not matched. Whether or not comparison has completed can be determined by checking whether the conversion-in-progress flag bit (ADC1: ADMV) is "0".

11.6 Notes on Using A/D Converter

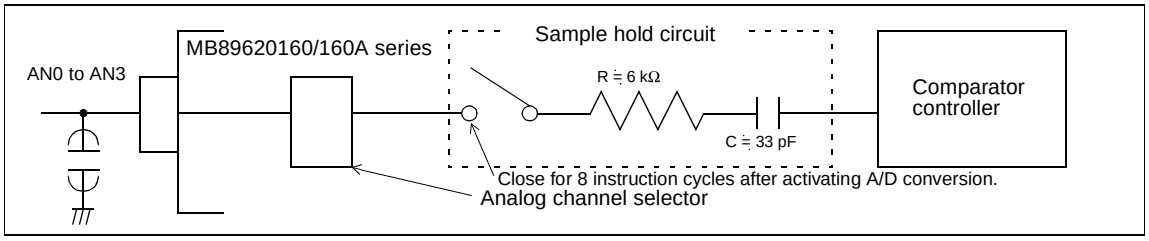
This section lists points to note when using the A/D converter.

■ Notes on Using A/D Converter

❍ Input impedance of analog input pins

The A/D converter contains a sample hold circuit as shown in Figure 11.6-1 "Analog Input Equivalent Circuit" to fetch analog input voltage into the sample hold capacitor for eight instruction cycles after activating A/D conversion (or the sense function). For this reason, if the output impedance of the external circuit for the analog input is high, analog input voltage might not stabilize within the analog input sampling period. Therefore, it is recommended to keep the output impedance of the external circuit low (below 10 kΩ). Note that if the impedance cannot be kept low, it is recommended to connect an external capacitor of about 0.1µF for the analog input pin.

Figure 11.6-1 Analog Input Equivalent Circuit



❍ Notes on setting by program

- For the A/D conversion function, the ADCD register maintains previous value until the next A/D conversion is activated. However, the content of the ADCD register becomes indeterminate immediately after activating A/D conversion.
- Do not re-select the analog input channel (ADC1: ANS3 to ANS0) or do not switch between the A/D conversion and sense functions (ADC2: ADMD) while the A/D conversion or sense function is operating. Particularly, when continuous activation is enabled, only perform such operations after disabling continuous activation (ADC2: EXT = "0") and waiting for the conversion-in-progress flag bit (ADC1: ADMV) to go to "0". Stop operation before modifying the compare condition setting bit (ADC1: SIFM) in the same way when the sense function is operating.
- When using the sense function, stop operation before writing to the ADCD register.
- Clear the interrupt request flag bit (ADC1: ADI = "0") before switching between the A/D conversion and sense functions.
- A reset or activation of stop mode stops the A/D converter and initializes all registers.
- Interrupt processing cannot return if the interrupt request flag bit (ADC1: ADI) is "1" and the interrupt request enable bit is enabled (ADC2: ADIE = "1"). Always clear the ADI bit.

❍ **Note on interrupt requests**

The interrupt request flag bit (ADC1: ADI) is not set if A/D conversion is reactivated (ADC1: AD = "1") at the same time as the previous A/D conversion completes, or if the sense function is reactivated (ADC1: AD = "1") at the same time as the comparison condition is satisfied.

❍ **Error**

The smaller the AVR - AVSS, the greater the error would become relatively.

❍ **Turn-on sequence for A/D converter power supply and analog inputs**

Always apply the A/D converter power supply (AV_{CC}, AV_{SS}) and analog inputs (AN0 to AN3) at the same time or after turning on the digital power supply (V_{CC}).

Similarly, when power supply is turned off, always turn off the A/D converter power supply (AV_{CC}, AV_{SS}) and analog inputs (AN0 to AN3) at the same time or before turning off the digital power supply (V_{CC}).

Take care that AV_{CC}, AV_{SS}, and the analog inputs do not exceed the digital power supply voltage when turning the A/D converter power supply on or off.

❍ **Conversion time**

A/D conversion function conversion time and sense function comparison time are affected by the clock mode, oscillator frequency, and main clock speed (speed shift function).

❍ **Continuous activation input clock**

The 8/16-bit timer/counter output, which can be selected for continuous activation (ADC2: EXT = 1), is affected by the clock mode and speed shift function. The timebase timer output, which can also be selected, is not affected by the speed shift function, but the timebase timer output cannot be used in subclock mode because the main clock (which drives the timebase timer) is stopped in that mode. Note also that the cycle time is affected (for one cycle) when the timebase timer is cleared.

11.7 Program Example for A/D Converter

This section gives program examples for the A/D conversion and sense functions of the 8-bit A/D converter.

■ Program Example for A/D Conversion Function

❍ Processing description

- Performs software-activated A/D conversion of the analog voltage input to the AN0 pin. The example does not use interrupts and detects conversion completion within the program loop.

❍ Coding example

```
PDR5        EQU     000FH           ; Port 5 data register
ADC1        EQU     002DH           ; A/D control register 1
ADC2        EQU     002EH           ; A/D control register 2
ADCD        EQU     002FH           ; A/D data register

AN0         EQU     PDR5:0          ; Define the AN0 analog input pin.
ADI         EQU     ADC1:3          ; Define the interrupt request flag bit.
ADMV        EQU     ADC1:2          ; Define the conversion-in-progress flag bit.
AD          EQU     ADC1:0          ; A/D converter activation bit (software
                                      activation)
EXT         EQU     ADC2:1          ; Define the continuous activation enable bit.

;---------- Main program ----------------------------------------------------------------------------------

                    CSEG            ; [CODE SEGMENT]

                    :
                    SETB    AN0     ; Set P50/AN0 pin as an analog input pin
                                      (AN0).
                    CLRI            ; Disable interrupts.
                    CLRB    EXT     ; Disable continuous activation.

AD_WAIT
                    BBS     ADMV,AD_WAIT    ; Loop to check that the A/D converter is
                                              stopped.
                    MOV     ADC1,#00000000B ; Select analog input channel 0 (AN0),  clear
                                              interrupt request flag, and do not activate by
                                              software.
                    MOV     ADC2,#00000001B ; Disable interrupt request output, select the
                                              A/D conversion function, and select software
                                              activation by the AD bit.
                    SETI            ; Enable interrupts.
                    :
                    SETB    AD      ; Activate by software
```

```
AD_CONV
            BBS     ADMV,AD_CONV    ; Loop to delay until A/D conversion completes
                                      (approx. 41.9 µs /4.2 MHz).
            CLRB    ADI             ; Clear interrupt request flag.
            MOV     A,ADCD          ; Read A/D conversion data.
             :
             :
            ENDS

;-----------------------------------------------------------------------------------------------

            END
```

■ Program Example for Sense Function

❍ Processing description

- Generate an interrupt if the analog voltage input to the AN0 pin is less than 3.0 V.
- Perform continuous activation of the sense function synchronized with pulses (Timebase timer output (from divided by $F_{CH}2^8$).
- For analog power supply voltage (AV_{CC}) = reference voltage (AVR) = 5.0 V, an ADCD register value of 0A9H gives a compare voltage of 3.0 V. With a main clock oscillator frequency of 4.2 MHz, the continuous activation cycle time would be 2^8/4.2 MHz = approx. 61.0 µs.

❍ Coding example

```
PDR5        EQU     000FH           ; Port 5 data register
ADC1        EQU     002DH           ; A/D control register 1
ADC2        EQU     002EH           ; A/D control register 2
ADCD        EQU     002FH           ; A/D data register

AN0         EQU     PDR5:0          ; Define the AN0 analog input pin.
ADI         EQU     ADC1:3          ; Define the interrupt request flag bit.
ADMV        EQU     ADC1:2          ; Define the conversion-in-progress flag bit.
AD          EQU     ADC1:0          ; A/D converter activation bit (software
                                      activation)
EXT         EQU     ADC2:1          ; Define the continuous activation enable bit.

ILR3        EQU     007EH           ; Set interrupt level setting register.

INT_V       DSEG    ABS             ; [DATA SEGMENT]
            ORG     0FFE8H
IRQB        DW      WARI
INT_V       ENDS

;---------- Main program ----------------------------------------------------------------------

            CSEG                    ; [CODE SEGMENT]
                                    ; Stack pointer (SP) etc. are already initialized.
```

```
                :
                SETB    AN0             ; Set P50/AN0 pin as an analog input pin.
                CLRI                    ; Disable interrupts.
                MOV     ILR3,#01111111B ; Set interrupt level (level 1).
                CLRB    EXT             ; Disable continuous activation.

AD_WAIT
                BBS     ADMV,AD_WAIT    ; Loop to check that the A/D converter is
                                          halted.
                MOV     ADCD,#9AH       ; Set compare voltage data (3.0 V).
                MOV     ADC1,#00000000B ; Select analog input channel 0 (AN0), clear
                                          interrupt request flag, set compare condition
                                          (interrupt if the input voltage is lower), and do
                                          not activate by software.
                MOV     ADC2,#00001111B ; Select timebase timer output as A/D clock,
                                          enable interrupt request output, select the
                                          sense function, and enable continuous
                                          activation.
                SETI                    ; Enable interrupts.

;---------- Interrupt processing routine ----------------------------------------------------------------------

WARI            CLRB    ADI             ; Clear interrupt request flag.
                PUSHW   A
                XCHW    A,T
                PUSHW   A

                :
                User processing
                :

                POPW    A
                XCHW    A,T
                POPW    A
                RETI
                ENDS

;-------------------------------------------------------------------------------------------------------------------

                END
```

CHAPTER 12 WATCH PRESCALER

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

12.1 Overview of Watch Prescaler

The watch prescaler provides interval timer functions. Four different interval times can be selected. The watch rescaler uses a 15-bit free-run counter which counts-up in sync with a subclock generated by the clock generator.

■ Interval Timer Function (Watch Interrupt)

- The interval timer function generates repeated interrupts at fixed intervals with the subclock used as the count clock.
- Interrupts are generated by watch prescaler interval timer divided clock outputs.
- The interval timer divided clock output (interval time) can be selected from different settings.
- The watch prescaler counter can be cleared.

Table 12.1-1 "Watch Prescaler Interval Time" lists the available interval times for the watch prescaler.

Table 12.1-1 Watch Prescaler Interval Time

Subclock Cycle Time	Interval time
$1/F_{CL}$ (approx. 30.5 µs)	$2^{10}/F_{CL}$ (31.25 ms)
	$2^{13}/F_{CL}$ (0.25 s)
	$2^{14}/F_{CL}$ (0.50 s)
	$2^{15}/F_{CL}$ (1.00 s)

F_{CL}: Subclock source oscillation

The values enclosed in parentheses ( ) are for a 32.768 kHz subclock source oscillation.

Check:

The watch prescaler cannot be used in devices in which a single clock option has been selected.

■ Clock Supply Function

The watch prescaler has the following clock supply functions:

- The timer output used for the subclock oscillation stabilization delay time (one value)
- The clock used for the watchdog timer (one value)
- The clock used for the buzzer output (three values)

Table 12.1-2 "Clocks Supplied by Watch Prescaler" lists the cycles of the clocks that the watch prescaler supplies to various peripherals.

Table 12.1-2 Clocks Supplied by Watch Prescaler

Subclock destination	Subclock cycle	Remarks
Subclock oscillation stabilization delay time	$2^{15}/F_{CL}$ (1.00 s)	Do not switch to the subclock mode during the oscillator stabilization wait time.
Watchdog timer	$2^{14}/F_{CL}$ (0.50 s)	Count-up clock for the watchdog timer
Buzzer output	$2^{3}/F_{CL}$ to $2^{5}/F_{CL}$ (approx. 0.24 to 0.98 ms)	See CHAPTER 15 "BUZZER OUTPUT".

F_{CL}: Subclock source oscillation

The values enclosed in parentheses ( ) are for a 32.768 kHz subclock source oscillation.

Note:

The oscillation stabilization delay time should be used as a guideline since the oscillation cycle is unstable immediately after oscillation starts.

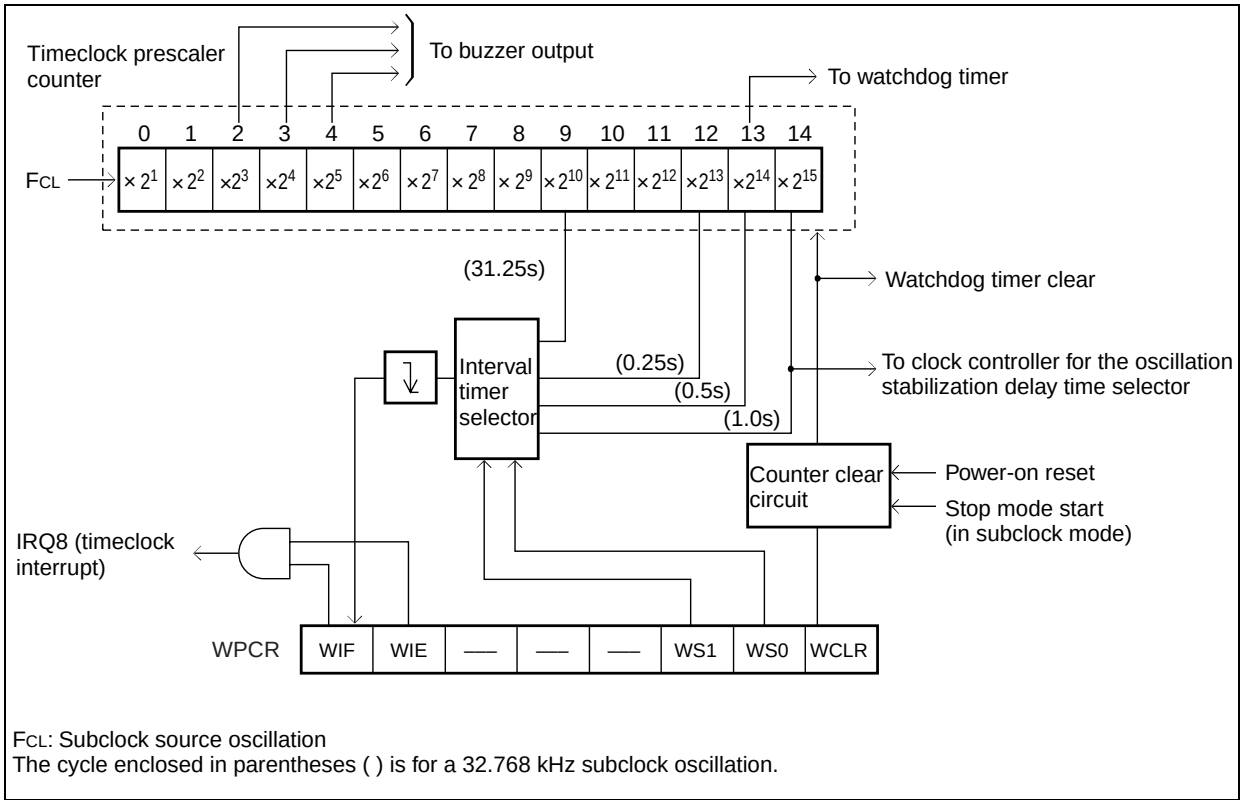
12.2 Block Diagram of Watch Prescaler

The watch prescaler consists of the following four blocks:

- **Watch prescaler counter**
- **Counter clear circuit**
- **Interval timer selector**
- **Watch prescaler control register (WPCR)**

■ Block Diagram of Watch Prescaler

Figure 12.2-1 Block Diagram of Watch Prescaler



FCL: Subclock source oscillation
The cycle enclosed in parentheses ( ) is for a 32.768 kHz subclock oscillation.

❍ Watch prescaler counter

A 15-bit up-counter that uses the subclock source oscillation clock as its count clock.

❍ Counter clear circuit

In addition to being cleared by setting the WPCR register (WCLR = "0"), the counter is cleared when the device changes to sub-stop mode (STBC : STP = "1") and by power-on reset (optional).

❍ **Interval timer selector**

This circuit selects one of four divided clock outputs of the watch prescaler counter as the interval timer output. The falling edge of the selected output is the event that triggers the watch interrupt.

❍ **WPCR register**

The WPCR register is used to select the interval time bit, clear the counter, control interrupts, and check the state of the watch prescaler.

12.3 Watch Prescaler Control Register (WPCR)

The watch prescaler control register (WPCR) is used to select the interval timer bit, clear the counter, control interrupts, and check state of the watch prescaler.

■ Watch Prescaler Control Register (WPCR)

Figure 12.3-1 Watch Prescaler Control Register (WPCR)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
000B$_H$	WIF	WIE	—	—	—	WS1	WS0	WCLR	00---000$_B$
	R/W	R/W				R/W	R/W	R/W	

WCLR	Watch prescaler clear bit	
	Read	Write
0	—	Clears the watch prescaler
1	Reading always return "1".	No effect. The bit does not change.

WS1	WS0	Watch interrupt interval time select bits
0	0	$2^{10}/F_{CL}$
0	1	$2^{13}/F_{CL}$
1	0	$2^{14}/F_{CL}$
1	1	$2^{15}/F_{CL}$

WIE	Interrupt Request Enable Bit
0	Disables interrupt request output
1	Enables interrupt request output

WIF	Watch interrupt request flag bit	
	Read	Write
0	Have no interval interrupt	Clears this bit.
1	Have interval interrupt	No effect. The bit does not change.

R/W : Readable and writable
W : Write-only
— : Unused
X : Indeterminate
□ : Initial value

Table 12.3-1 Watch Prescaler Control Register (WPCR) Bits

Bit		Function
Bit 7	WIF: Watch interrupt request flag bit	• Set to "1" by the falling edge of the selected interval timer divided output. • An interrupt request is output when both this bit and the interrupt request enable bit (WIE) are "1". • Writing "0" clears this bit. Writing "1" has no effect and does not change the bit value.
Bit 6	WIE: Interrupt request enable bit	• This bit enables or disables an interrupt request output to the CPU. An interrupt request is output when both this bit and the watch interrupt request flag bit (WIF) are "1".
Bit 5 Bit 4 Bit 3	Unused bits	• The read value is indeterminate. • Writing to these bits has no effect on the operation.
Bit 2 Bit 1	WS1, WS0: Watch interrupt interval time selection bits	• Select interval timer cycle. • Specify which bit of the watch prescaler counter (or which divided output) will be used for the interval timer. • One of four interval times may be selected.
Bit 0	WCLR: Watch prescaler clear bit	• Bit used to clear the watch prescaler counter. • Writing "0" to this bit clears the counter to 0000_H. Writing "1" has no effect and does not change the bit value. **Note:** The read value is always "1".

12.4 Watch Prescaler Interrupt

The watch prescaler generates an interrupt request at the falling edge of the specific divided output (interval timer function).

■ Interrupts for Interval Timer Function (Watch Interrupt)

The watch prescaler counter counts up, clocked by the subclock source oscillation. Unless the system is in main-stop mode, the watch interrupt request flag is set to "1" (WPCR: WIF = 1) at the end of the selected time interval. At this time, an interrupt request (IRQ8) to the CPU is generated if the interrupt request enable bit is enabled (WPCR: WIE = "1"). Write "0" to the WIF bit in the interrupt processing routine to clear the interrupt request. The WIF bit is set when the specified divide output falls, regardless of the WIE bit value.

Check:

When enabling an interrupt request output (WE = "1") after wake-up from a reset, always clear the WIF bit (WIF = "0") at the same time.

Notes:

An interrupt request is generated immediately if the WIF bit is "1" when the WIF bit is changed from disabled to enabled ("0" --> "1").

The WIF bit is not set if the counter cleared (WPCR: WCLR = "0") at the same time as an overflow on the specified bit occurs.

■ Oscillation Stabilization Delay Time and Watch Interrupt

If the interval time is set shorter than the subclock oscillation stabilization delay time, an watch interrupt request from the watch prescaler (WPCR: WIF ="1") is generated at the time when CPU wakes up from sub-stop mode by an external interrupt. In this case, disable the watch prescaler interrupt (WPCR: WIE = "0") when changing to sub-stop mode.

■ Register and Vector Table for Watch Prescaler Interrupt

Table 12.4-1 "Register and Vector for Watch Prescaler Interrupt" lists the register and vector table for watch prescaler interrupt.

Table 12.4-1 Register and Vector for Watch Prescaler Interrupt.

Interrupt	Interrupt level settings register			Vector table address	
	Register	Setting bits		Upper	Lower
IRQ8	ILR3 ($007E_H$)	L81 (bit 1)	L80 (bit 0)	$FFEA_H$	$FFEB_H$

Reference:

See Section 3.4.2 "Interrupt Processing" for details on the interrupt operations.

12.5 Operation of Watch Prescaler

The watch prescaler has the interval timer function and the clock supply function.

■ Operation of Interval Timer Function (Watch Prescaler)

Figure 12.5-1 "Interval Timer Function Settings" shows the settings required to operate the interval timer function.

Figure 12.5-1 Interval Timer Function Settings

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WPCR	WIF	WIE	—	—	—	WS1	WS0	WCLR
	0	1				◎	◎	0

◎ : Used bit
1 : Set "1"
0 : Set "0"

Provided the subclock is oscillating, the watch prescaler 15-bit counter (continues to count-up using the subclock as its count clock).

After being cleared (WCLR = "0"), the counter restarts counting-up from "0000_H". When the counter reaches a full count of "$7FFF_H$", the next count takes it "0000_H" and it continues to count-up. As the count proceeds, a falling edge will eventually occur at the selected divided clock output. At this time, unless the system is in main clock stop mode, the watch prescaler sets the watch interrupt request flag bit (WIF) to "1". Consequently, the watch prescaler generates interrupt requests at fixed intervals (the selected interval time), based on the time that the counter is cleared.

■ Operation of Clock Supply Function

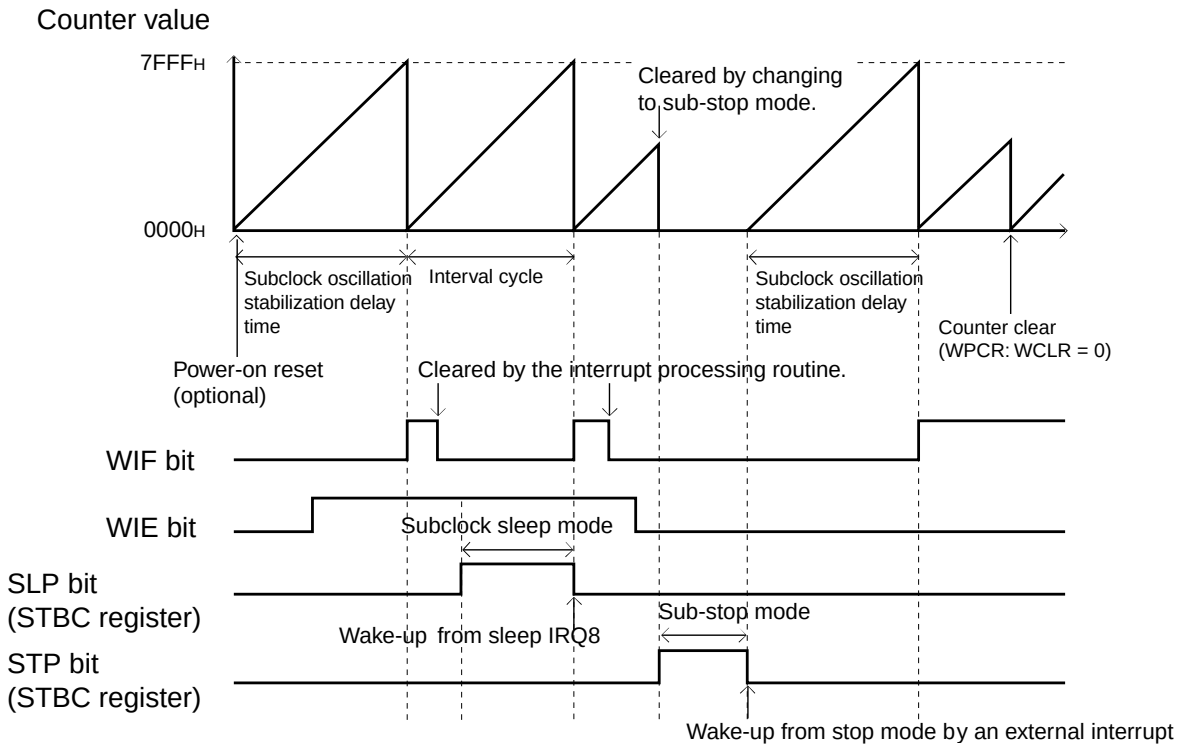
The watch prescaler is also used as a timer to generate the subclock oscillation stabilization delay time. The time between the counter cleared state and the falling edge of the MSB output is used the subclock oscillation stabilization delay time ($2^{15}/F_{CL}$, where F_{CL} is subclock source oscillation).

The watch prescaler also provides the clock for the watchdog timer. When the watch prescaler is selected as the clock source for the watchdog timer (WDTC: CS = 1) both counters are cleared simultaneously.

■ Operation of Watch Prescaler

Figure 12.5-2 "Operation of Watch Prescaler" shows counter states when the interval timer is operating in subclock mode and the system goes into the sleep and stop modes, and when there is a counter clear request.

Figure 12.5-2 Operation of Watch Prescaler



For the case when the interrupt interval time selection bits in the watch prescaler control register (WPCR: WS1, WS0) are "set to 11" ($2^{15}/F_{CL}$).

12.6 Notes on Using Watch Prescaler

This section lists points to note when using the watch prescaler. The watch prescaler cannot be used in devices in which the single-clock option has been selected.

■ Notes on Using Watch Prescaler

❍ Notes on setting bits by program

The system cannot recover from interrupt processing if the interrupt request flag bit (WPCR : WIE) is "1" and the interrupt request enable bit is enabled (WPCR: WIE = "1"). Always clear the WIF bit.

❍ Clearing Watch prescaler

In addition to being cleared by the watch prescaler clear bit (WPCR: WCLR = "0"), the watch prescaler is cleared wherever the subclock oscillation stabilization delay time is required.

When the watch prescaler is selected as a count clock of the watchdog timer (WDTC: CS = "1"), clearing the watch prescaler also clears the watchdog timer.

❍ Using as timer for oscillator stabilization delay time

As the subclock source oscillation is stopped when the power is turned on and during sub-stop mode, the watch prescaler provides the oscillation stabilization delay time after the oscillator starts. Do not switch clock modes from main clock to subclock during this delay time (immediately after power on, etc.)

The subclock oscillation stabilization delay time is fixed.

Reference:

See Section 3.6.5 "Oscillation Stabilization Delay Time" for details.

❍ Notes on watch interrupt

In main-stop mode, the watch prescaler counter operates, but no interrupt request interrupt requests are generated.

❍ Notes on peripheral functions that provides a clock supply from watch prescaler

As the clock derived from the watch prescaler restarts output from the its initial state when the watch prescaler counter is cleared, the "H" level may be shorter or the "L" level longer by a maximum of half cycle. the clock of the watchdog timer also restarts output from its initial state. However, as the watchdog timer counter is cleared at the same time, the watchdog timer operates in normal cycle.

12.7 Program Example for Watch Prescaler

This section gives program example for the watch prescaler.

■ Program Example for the Watch Prescaler

❍ Processing description

Generates repeated watch interrupts at $2^{15}/F_{CL}$ (F_{CL} = subclock source oscillation) intervals. At this time, the interval time is 1 second (at 32,768 kHz operation).

❍ Coding example

```
WPCR    EQU     000BH           ; Address of watch prescaler control register
WIF     EQU     WPCR:7          ; Define the watch interrupt request flag bit.
ILR3    EQU     007EH           ; Address of the interrupt level setting
                                  register 2
INT_V   DSEG    ABS             ;  [DATA SEGMENT]
        ORG     0FFEAH
IRQ8    DW      WARI            ; Set interrupt vector.
INT_V   ENDS

;---------- Main program ----------------------------------------------------------------------------------

        CSEG                    ; [CODE SEGMENT]
                                ; Stack pointer (SP) etc. are already initialized.

        :
        CLRI                    ; Disable interrupts.
        MOV     ILR3,#11111110B ; Set interrupt priority (level 2).
        MOV     WPCR,#01000110B ; Clear interrupt request flag, enable interrupt
                                  request output, select 215/FCL, and clear
                                  watch prescaler.
        SETI                    ; Enable interrupts.
        :

;---------- Interrupt program -----------------------------------------------------------------------------

WARI    CLRB    WIF             ; Clear interrupt request flag.
        PUSHW   A
        XCHW    A,T
        PUSHW   A

        :
        User processing
        :
```

```
        POPW    A
        XCHW    A,T
        POPW    A
        RETI
        ENDS
;---------------------------------------------------------------------------------------------------------------------

        END
```

CHAPTER 13 REMOTE CONTROL GENERATOR (6-BIT PPG)

This chapter describes the functions and operation of the remote control generator.

13.1 Overview of Remote Control Generator

The remote control generator is a 6-bit binary counter that can select one of four clocks as its count clock. Both the cycle of the output waveform and its "H" state pulse width can be set, which allows the circuit to be used as a 6-bit PPG. The circuit uses the same output pin as the buzzer output.

■ Remote Control Generation Function

- Generates frequencies for use by a remote control unit, and outputs the signal at the RCO pin.
- The cycle and "H" state pulse width of the output waveform can be set separately.
- The count clock can be selected from four different internal clocks.
- The frequencies can generate with a cycle among 2 and 2^6 times the count clock cycle.

Table 13.1-1 "Output Cycles and "H" Pulse Width Ranges" lists the available range of "H" state pulse widths.

Table 13.1-1 Output Cycles and "H" Pulse Width Ranges

Internal count clock cycle	Output cycle	Output "H" pulse width*
0.5 t_{inst}	1 t_{inst} to 32 t_{inst}	0.5 t_{inst} to 31.5 t_{inst}
1 t_{inst}	2 t_{inst} to 63 t_{inst}	1 t_{inst} to 62 t_{inst}
8 t_{inst}	16 t_{inst} to 504 t_{inst}	8 t_{inst} to 496 t_{inst}
32 t_{inst}	64 t_{inst} to 2016 t_{inst}	32 t_{inst} to 1984 t_{inst}

t_{inst}: instruction cycle (affected by clock mode, etc.)

*: Can also output a steady "H" state (100% duty cycle).

❍ Calculation example for the remote control generator cycle an "H" width (when a 0.5 tinst clock is selected for count clock cycle)

Assume a main clock source oscillation (F_{CH}) of 4.2 MHz, and a 0.5 t_{inst} clock selected for count clock cycle. Also assume main clock mode, and the highest clock speed selected from the system clock control register (SYCC: SCS = CS1 = CS0 = 1). (This makes the instruction cycle time $4/F_{CH}$.) Then, for the indicated comparison values, the output waveform cycle and "H" state pulse width can be calculated as follows:

Cycle comparison value = 011110_B (30 clock cycles)
Pulse width comparison value = 001010_B (10 clock cycles)

Cycle = (cycle comparison value + 1) × count clock cycle
= "011110_B" (30+1 clock cycles) × 0.5 × $4/F_{CH}$
= 31 × 0.475 µs
= 14.725 µs

"H" pulse width = ("H" pulse width comparison value + 1) × count clock cycle
= "001010_B" (10 + 1 clock cycles) × 0.5 × $4/F_{CH}$
= 11 × 0.475 µs
= 5.225 µs

If the "H" pulse width setting is equal to or greater than the cycle setting, the output will be a steady "H" state.

❍ **Calculation example for the remote control generator cycle an "H" width (when a 1/8/32 tinst clock is selected for count clock cycle)**

Assume a main clock source oscillation (F_{CH}) of 4.2 MHz, and a 1 t_{inst} clock selected for count clock cycle. Also assume main clock mode, and the highest clock speed selected from the system clock control register (SYCC: SCS = CS1 = CS0 = 1). (This makes the instruction cycle time $4/F_{CH}$.) Then, for the indicated comparison values, the output waveform cycle and "H" state pulse width can be calculated as follows:

Cycle comparison value = 011110_B (30 clock cycles)
Pulse width comparison value = 001010_B (10 clock cycles)

Cycle = cycle comparison value × count clock cycle
= "011110_B" (30 clock cycles) × 1 × $4/F_{CH}$
= 30 × 0.95 µs
= 28.6 µs

"H" pulse width = "H" pulse width comparison value × count clock cycle
= "001010_B" (10 clock cycles) × 1 × $4/F_{CH}$
= 11 × 0.95 µs
= 9.5 µs

If the "H" pulse width setting is equal to or greater than the cycle setting, the output will be a steady "H" state.

■ **6-bit PPG Function (when a 0.5 tinst clock selected for count clock cycle)**

Because the cycle and "H" pulse width of its output waveform can be set separately, the remote control generator can be used as a 6-bit PPG. The duty ratio is from 1.56% to 100%. The valid range of "H" pulse width comparison settings, however, is from "0" to the cycle comparison setting. This means that the lower the cycle comparison setting (the shorter the cycle of the output waveform), the lower the resolution (the larger the minimum duty ratio step size).

For a cycle comparison setting of "1", for example, the possible "H" pulse width comparison settings would be "0" and "1" which would result in a resolution of 1/2. The duty ratios for these settings would be 50% and 100%, or a minimum duty ratio step of 50%.

The output cycle and duty ratio are calculated as follows:

Output cycle = (cycle comparison value +1) × 0.5 tinst.

Duty ratio (%) = ("H" pulse width compare value +1)/(cycle compare value +1) × 100

Table 13.1-2 "6-Bit PPG Resolution and Output Cycles (0.5 tinst count clock)" shows the available output cycle, resolution and the minimum steps for duty ratio.

Table 13.1-2 6-Bit PPG Resolution and Output Cycles (0.5 tinst count clock)

Cycle comparison value	"H" pulse width comparison value setting range	Output cycle Count clock = 0.5 t_{inst}	Resolution	Duty ratio Minimum step
0	0	-	Output "H"	
1	0 to 1	1 t_{inst}	1/2	50.0%*
2	0 to 2	1.5 t_{inst}	1/3	33.3%*
3	0 to 3	2.0 t_{inst}	1/4	25.0%*
4	0 to 4	2.5 t_{inst}	1/5	20.0%*
5	0 to 5	3.0 t_{inst}	1/6	16.7%*
6	0 to 6	3.5 t_{inst}	1/7	14.3%*
7	0 to 7	4.0 t_{inst}	1/8	12.5%*
8	0 to 8	4.5 t_{inst}	1/9	11.1%*
9	0 to 9	5.0 t_{inst}	1/10	10.0%*
10	0 to 10	5.5 t_{inst}	1/11	9.09%*
		:		
15	0 to 15	8.0 t_{inst}	1/16	6.25%*
		:		
20	0 to 20	10.5 t_{inst}	1/21	4.76%*
		:		
25	0 to 25	13.0 t_{inst}	1/26	3.85%*
		:		
30	0 to 30	15.5 t_{inst}	1/31	3.23%*
		:		
40	0 to 40	20.5 t_{inst}	1/41	2.44%*
		:		
50	0 to 50	25.5 t_{inst}	1/51	1.96%*
		:		
60	0 to 60	30.5 t_{inst}	1/61	1.64%*
		:		
63	0 to 63	32 t_{inst}	1/64	1.56%*

t_{inst}: instruction cycle time

*: Steady "H" output when "H" pulse width comparison value is equal to period comparsion value.

■ 6-bit PPG Function (when a 1/8/32 tinst clock selected for count clock cycle)

Because the cycle and "H" pulse width of its output waveform can be set separately, the remote control generator can be used as a 6-bit PPG. The duty ratio is from 1.6% to 100%. The valid range of "H" pulse width comparison settings, however, is from "1" to the cycle comparison setting. This means that the lower the cycle comparison setting (the shorter the cycle of the output waveform), the lower the resolution (the larger the minimum duty ratio step size).

For a cycle comparison setting of "2", for example, the possible "H" pulse width comparison settings would be "1", and "2", which would result in a resolution of 1/2. The duty ratios for these settings would be 50%, and 100%, or a minimum duty ratio step of 50%.

Output cycle = cycle comparison value × selected count clock cycle

Duty ratio (%) = ("H" pulse width compare value/cycle compare value) × 100

Table 13.1-3 "6-Bit PPG Resolution and Output Cycles (0.5 tinst count clock)" shows the available output cycle, resolution and the minimum steps for duty ratio.

Table 13.1-3 6-Bit PPG Resolution and Output Cycles (0.5 tinst count clock)

Cycle comparison value	"H" pulse width comparison value setting range	Output cycle (Count clock)			Resolution	Duty ratio Minimum step
		1 t_{inst}	8 t_{inst}	32 t_{inst}		
0	0	Prohibited setting			Output "H"	
1	1					
2	1 to 2	2 t_{inst}	16 t_{inst}	64 t_{inst}	1/2	50.0%*
3	1 to 3	3 t_{inst}	24 t_{inst}	96 t_{inst}	1/3	33.3%*
4	1 to 4	4 t_{inst}	32 t_{inst}	128 t_{inst}	1/4	25.0%*
5	1 to 5	5 t_{inst}	40 t_{inst}	160 t_{inst}	1/5	20.2%*
6	1 to 6	6 t_{inst}	48 t_{inst}	192 t_{inst}	1/6	16.7%*
7	1 to 7	7 t_{inst}	56 t_{inst}	224 t_{inst}	1/7	14.3%*
8	1 to 8	8 t_{inst}	64 t_{inst}	256 t_{inst}	1/8	12.5%*
9	1 to 9	9 t_{inst}	72 t_{inst}	288 t_{inst}	1/9	11.1%*
10	1 to 10	10 t_{inst}	80 t_{inst}	320 t_{inst}	1/10	10.0%*
				:		
15	1 to 15	15 t_{inst}	120 t_{inst}	480 t_{inst}	1/15	6.7%*
				:		
20	1 to 20	20 t_{inst}	160 t_{inst}	640 t_{inst}	1/20	5.0%*
				:		
25	1 to 25	25 t_{inst}	200 t_{inst}	800 t_{inst}	1/25	4.0%*

Table 13.1-3 6-Bit PPG Resolution and Output Cycles (0.5 tinst count clock)

Cycle comparison value	"H" pulse width comparison value setting range	Output cycle (Count clock)			Resolution	Duty ratio Minimum step
		1 t_{inst}	8 t_{inst}	32 t_{inst}		
				:		
30	1 to 30	30 t_{inst}	240 t_{inst}	960 t_{inst}	1/30	3.3%*
				:		
40	1 to 40	40 t_{inst}	320 t_{inst}	1280 t_{inst}	1/40	2.5%*
				:		
50	1 to 50	50 t_{inst}	400 t_{inst}	1600 t_{inst}	1/50	2.0%*
				:		
60	1 to 60	60 t_{inst}	480 t_{inst}	1920 t_{inst}	1/60	1.7%*
				:		
63	1 to 63	63 t_{inst}	504 t_{inst}	2016 t_{inst}	1/63	1.6%*

t_{inst}: instruction cycle time

*: If "H" pulse width comparison setting is "00_H", a 0.5 tinst long "H" pulse will be outputted. Steady "H" output when "H" pulse width comparison value is equal to period comparsion value.

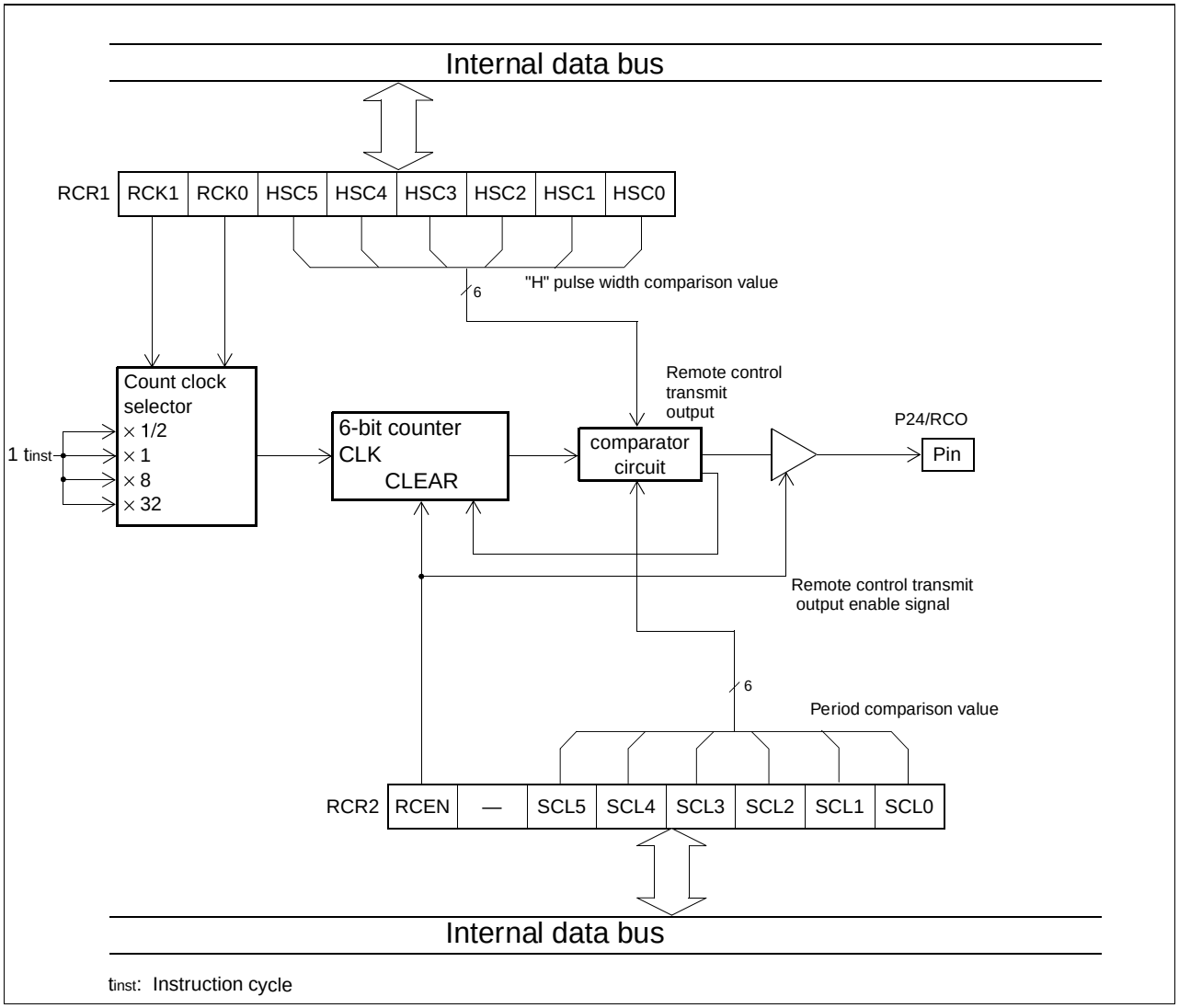
13.2 Block Diagram of Remote Control Generator

The remote control generator consists of the following five blocks:

- **Count clock selector**
- **6-bit counter**
- **Comparator circuit**
- **Remote control register 1 (RCR1)**
- **Remote control register 2 (RCR2)**

■ Block Diagram of Remote Control Generator

Figure 13.2-1 Block Diagram of Remote Control Generator



- **Count clock selector**

  Selects a count-up clock for the 6-bit counter from the four available internal count clock.

- **6-bit counter**

  The 6-bit counter counts-up, on the count clock selected by the count clock selector. The counter can be cleared by clearing the output enable bit of the RCR2 register (RCR2: RCEN = 0).

- **Comparison circuit**

  The comparison circuit holds a "H" state until the count in the 6-bit counter matches the setting in the "H" pulse width compare register. Then it holds the "L" state until the counter count matches the setting in the cycle compare register, at which time the counter is cleared to all zeros and continues counting.

- **Remote control register 1 (RCR1)**

  RCR1 is used to select the counter clock for remote control transmit output, and set the output "H" pulse width comparison value.

- **Remote control register 2 (RCR2)**

  RCR2 is used to enable/disable outputs for remote control transmit output, and set the output cycle comparison value.

13.3 Structure of Remote Control Generator

The section describes the pin, pin block diagram and register of the remote control generator.

■ Remote Control Generator Pin

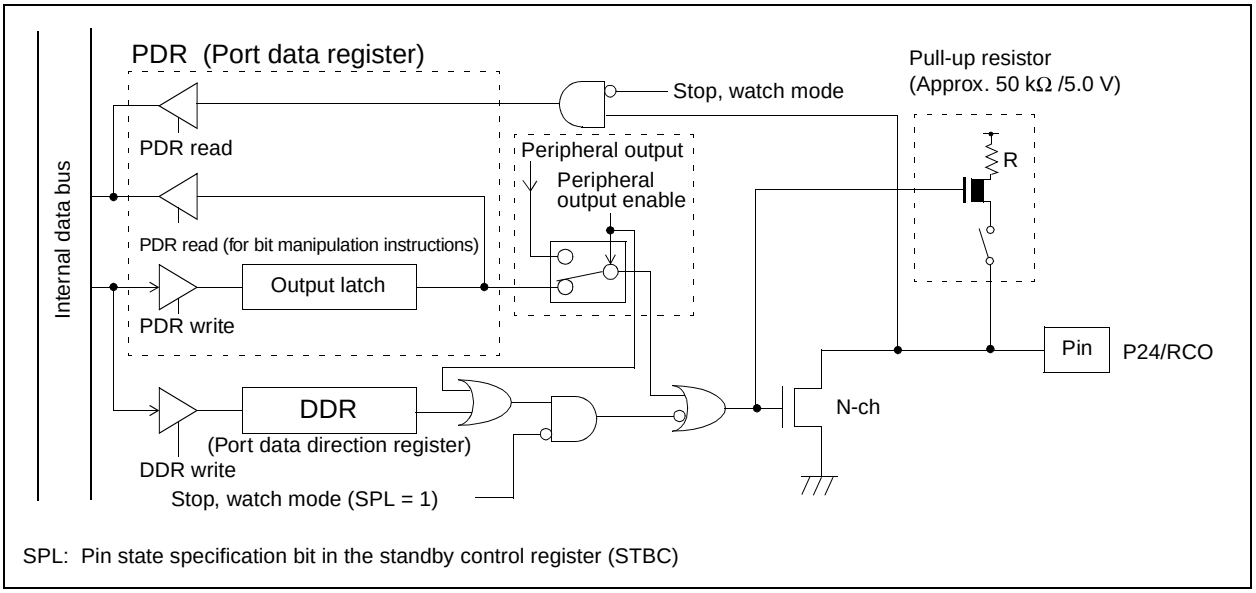
The remote control generator uses the P24/RCO pin. The pin can function as an I/O port (P24), or as the remote control output (RCO).

RCO:

When the remote control transmit output enable bit is set to "1" (RCR2: RCEN = 1), this pin functions as the remote control transmit output pin, outputting a waveform having a "H" state pulse width and cycle as set.

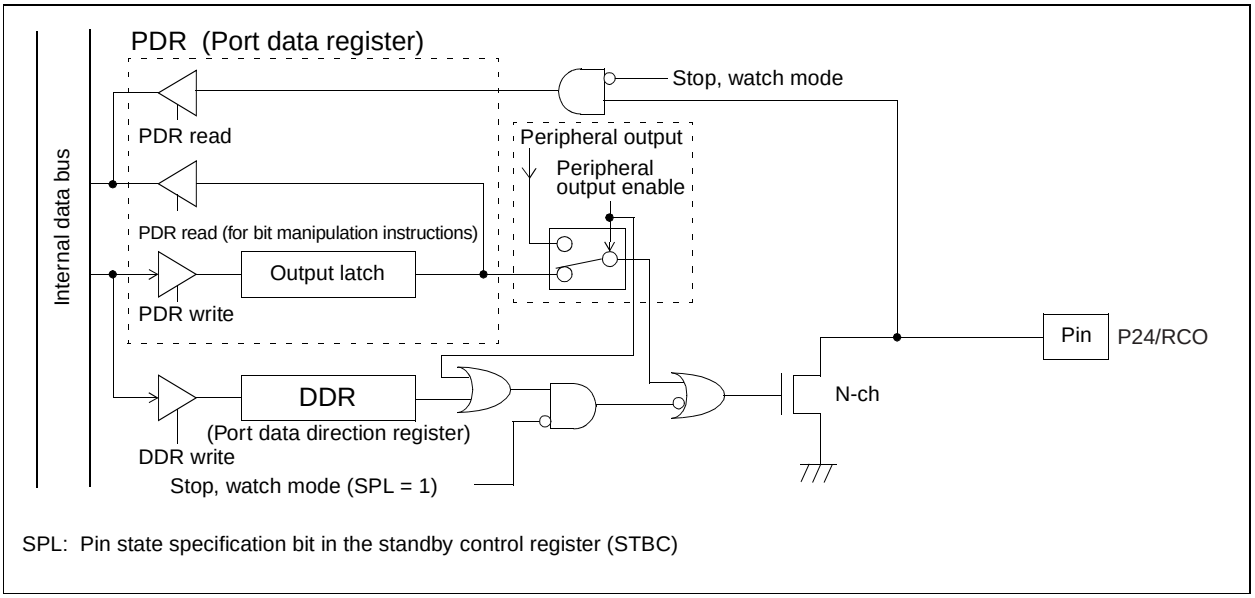
■ Block Diagram of Remote Control Generator Pin

Figure 13.3-1 Block Diagram of P24/RCO Pin for MB89983



SPL: Pin state specification bit in the standby control register (STBC)

Figure 13.3-2 Block Diagram of P24/RCO Pin for MB89P985 and MB89PV980



SPL: Pin state specification bit in the standby control register (STBC)

■ Remote Control Generator Registers

Figure 13.3-3 Remote Control Transmit Generator Registers

RCR1 (Remote control register 1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0014H	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

RCR2 (Remote control register 2)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0015H	RCEN	—	SCL5	SCL4	SCL3	SCL2	SCL1	SCL0	0-000000B
	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

R.W: Readable and writable
— : Unused

13.3.1 Remote Control Register 1 (RCR1)

Remote control register 1 is used to select the counter clock, and set the "H" pulse width.

■ **Remote Control Register 1 (RCR1)**

Figure 13.3-4 Remote Control Register 1 (RCR1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0014H	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0	00000000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

HSC5 to HSC0	"H" pulse width setting bits
XXXXXX	"H" pulse width comparison value

RCK1	RCK0	Count clock selection bits
0	0	1/2 t_{inst}
0	1	1 t_{inst}
1	0	8 t_{inst}
1	1	32 t_{inst}

R/W : Readable and writable
t_{inst} : Instruction cycle

Table 13.3-1 Remote Control Register 1 (RCR1) Bits

Bit		Function
Bit 7 Bit 6	RCK2, RCK1: Count clock selection bits	Select one of four internal clocks as the remote control transmit frequency generator count clock.
Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0	HSC5 to HSC0: "H" pulse width setting bits	These bits set the number of counts for which the remote control output generator output is to remain "H". (The "H" pulse width comparison value to be matched by the count in the counter.) **Note:** When the count clock is 0.5 tinst The setting value between "000000" and "111110" (00 to $3E_H$) is always set a value less than the cycle comparison setting. If they are set equal to or greater than the setting, a steady "H" is output. When the count clock is 1/8/32 tinst The setting value between "000001" and "111110" (01 to $3E_H$) is always set a value less than the cycle comparison value. If the value is set to "000000", a 0.5 tinst long "H" pulse will be outputted. If they are set equal to or greater than the setting, a steady "H" is output.

13.3.2 Remote Control Register 2 (RCR2)

Remote control register 2 is used to enable/disable outputs, and set the output cycle period.

■ Remote Control Register 2 (RCR2)

Figure 13.3-5 Remote Control Register 2 (RCR2)

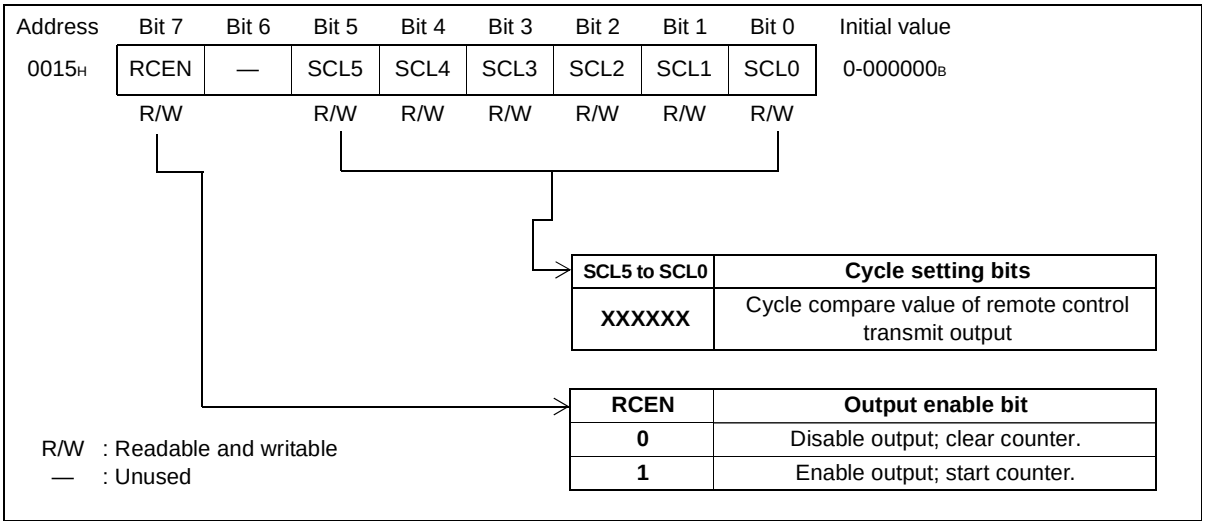


Table 13.3-2 Remote Control Register 2 (RCR2) Bits

Bit		Function
Bit 7	RCEN: Output enable bit	When this bit is "0", the P24/RCO pin functions as a N-ch open-drain port pin (P24), and when it is "1", the pin functions as the remote control transmit output pin (RCO). Setting this bit to "0" clears and stops the counter; setting it to "1" starts the counter.
Bit 6	Unused bits	• The read value is indeterminate. • Writing to this bit has no effect on the operation.
Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0	SCL5 to SCL0: Cycle setting bits	These bits set the length of the output cycle in terms of counter counts. (The cycle comparison value to be matched by the count in the counter.) **Note:** When the count clock is 0.5 tinst The setting value is between "000001" and "111111" (01 to $3F_H$). The RCO output will remain the previous state until the cycle comparison value is matched by the count value in the counter. Then the RCO output will output "H" state. When the count clock is 1/8/32 tinst The setting value is between "000010" and "111111" (02 to $3F_H$). The RCO output will remain the previous state until the cycle comparison value is matched by the count value in the counter. Then the RCO output will output "H" state. If this setting value is set to "01_H" and the "H" pulse comparison value is "00_H", RCO will output a 0.5 tinst long "H" pulse.

13.4 Operation of Remote Control Generator

The remote control generator (6-bit PPG) generates a remote control transmit output in which the cycle and "H" state pulse width of the output can be set separately.

■ Operation of Remote Control Generator

Figure 13.4-1 "Remote Control Generator Settings" shows the settings required to operate the remote control generator.

Figure 13.4-1 Remote Control Generator Settings

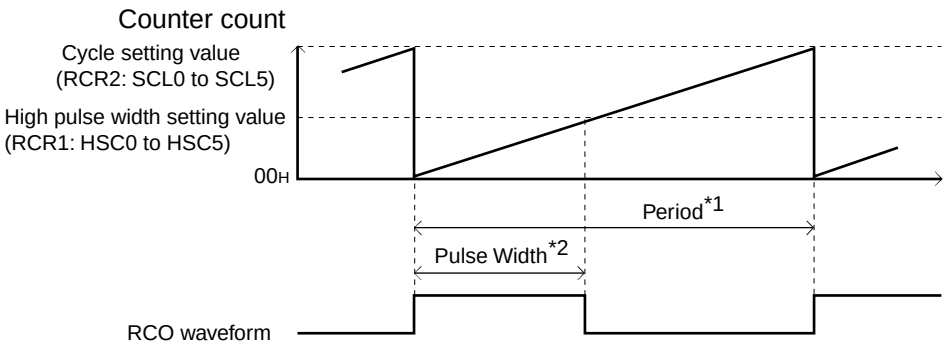
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCR1	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0
	◎	◎	◎	◎	◎	◎	◎	◎
RCR2	RCEN	—	SCL5	SCL4	SCL3	SCL2	SCL1	SCL0
	1		◎	◎	◎	◎	◎	◎

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

When the remote control generator output is enabled, its 6-bit counter starts counting up from "zero" in synchronization with the selected count clock. During the first cycle, the RCO pin output will remain in the previous state value until the count in the counter matches the "H" state pulse width comparison value. When this happens, the RCO pin goes "L" and stays there until the count matches the cycle period comparison value, at which time the RCO pin goes "H" and the 6-bit counter is cleared to zero and continues to count. The fact that the pulse width and period can be set separately enables the circuit to be used as a 6-bit PPG.

Figure 13.4-2 "Operation of Remote Control Generator" shows the operation of the remote control generator after the first RCO output cycle.

Figure 13.4-2 Operation of Remote Control Generator



*1 : When count clock is 0.5 t_{inst}
Period = 0.5 t_{inst} x (Period comparison value + 1)
When count clock is 1/8/32 t_{inst}
Period = Count clock period x Period comparison value

*2: When count clock is 0.5 t_{inst}
Pulse width = 0.5 t_{inst} x (High pulse width comparison value + 1)
When count clock is 1/8/32 t_{inst}
Pulse width = Count clock period x High pulse width comparison value

13.5 Notes on Using Remote Control Generator

This section lists points to note when using the remote control generator.

■ Notes on Using Remote Control Generator

❍ "H" pulse width restrictions

Generally, the "H" pulse width setting bits of remote control register 1 (RCR1: HSC5 to 0) must always be set less than the cycle setting bits of remote control register 2(RCR2: SCL5 to 0).

For any clock, when SCL5 to 0 is "00_H" HSC5 to 0 is equal to SCL5:0, or HSC5 to 0 is larger than SCL5 to 0, the RCO pin will remain a steady "H" level.

For any clock, when HSC5 to 0 is "00_H" and SCL5 to 0 is not equal to "00_H", the RCO pin will start to output "L" level and then output a 0.5 tinst long "H" pulse after every cycle match is detected.

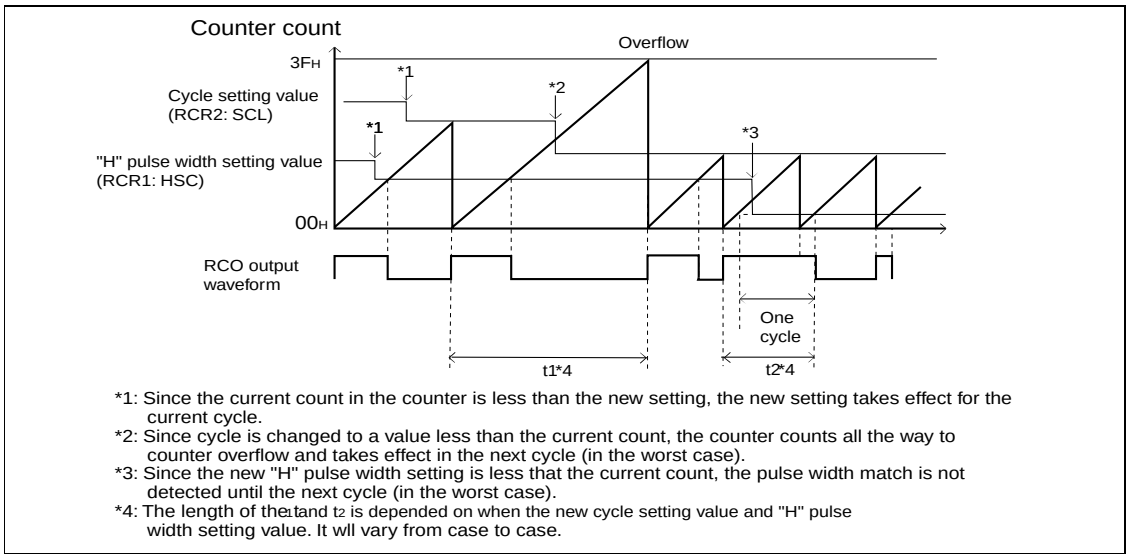
❍ Resolution

When count clock is 0.5 t_{inst}, the maximum "H" pulse width resolution is 1/64 of the cycle (the resolution when the cycle setting is "111111" ($3F_H$). Reducing the time of the cycle reduces the resolution, with the minimum resolution of 1/2 occurring with a cycle setting of "000001" (01_H). When count clock is 1/8/32 t_{inst}, the maximum "H" pulse width resolution is 1/63 of the cycle (the resolution when the cycle setting is "111111" ($3F_H$). Reducing the time of the cycle reduces the resolution, with the minimum resolution of 1/2 occurring with a cycle setting of "000010" (02_H).

❍ Changing settings during operation

Figure 13.5-1 "Changing Settings during Operation (Remote Control)" illustrates what happens when settings are changed during remote control generator operation.

Figure 13.5-1 Changing Settings during Operation (Remote Control)



*1: Since the current count in the counter is less than the new setting, the new setting takes effect for the current cycle.
*2: Since cycle is changed to a value less than the current count, the counter counts all the way to counter overflow and takes effect in the next cycle (in the worst case).
*3: Since the new "H" pulse width setting is less that the current count, the pulse width match is not detected until the next cycle (in the worst case).
*4: The length of the t_1 and t_2 is depended on when the new cycle setting value and "H" pulse width setting value. It wll vary from case to case.

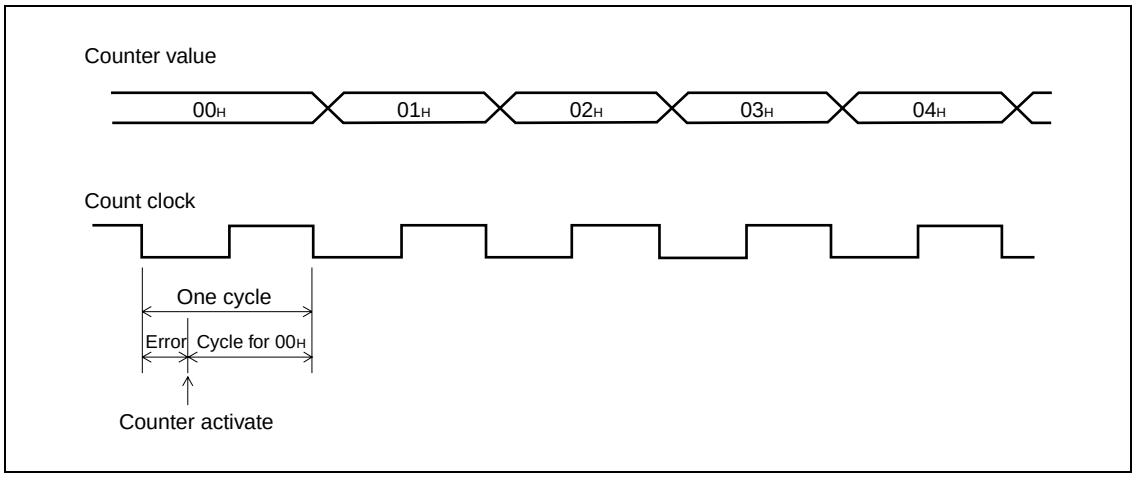
Direct comparisons are performed between the 6-bit counter of the remote control generator and the "H" pulse width setting bits (RCR1: HSC5 to HSC0), and between the counter and the cycle setting bits (RCR2: SCL5 to SCL0). Therefore, if a setting is reduced in mid-count, the cycle time (t_1) may be long until the counter overflows in the worst case and the takes effect in the next cycle. Similarly, the "H" state pulse width (t_2) may be long in the worst case until the next end-of-cycle match is detected.

❍ Errors

Activating the counter by program is not synchronized with the start of counting-up using the selected count clock. Therefore, the time from activating the counter until a match with the "H" pulse comparison value and Cycle comparison value are detected may be shorter than the theoretical time by a maximum of one cycle of the count clock.

Figure 13.5-2 "Error during activating Operation (Remote Control)" shows the error that occurs on starting counter operation.

Figure 13.5-2 Error during activating Operation (Remote Control)



13.6 Program Example for Remote Control Generator

This section gives a program example for the remote control generator.

■ Programming Example for Remote Control generator

❍ Processing description

- Generate the remote control transmit Output at a cycle of approximately 28.6 μs and a 33% duty ratio.
- With a main clock frequency (F_{CH}) of 4.2 MHz, the highest clock speed (speed shift function), and the 1 tinst clock selected (1 instruction cycle time = $4/F_{CH}$), the comparison value for a cycle of approximately 28.6 μs is as found as follows:

  Cycle comparison value (RCR2: SCL5 to SCL0) = 28.6 μs/(1 × 4/4.2 MHz) = 30
- The comparison value for a "H" state pulse width to provide a 33% duty ratio is found as follows:

  "H" pulse width comparison value (RCR1: HSC5 to HSC0) = 33/100 × cycle comparison value = 0.33 × 30 = 10

(This is an approximately 9.5 μs "H" pulse width.)

❍ Coding example

```
RCR1    EQU     0014H               ; Remote control register 1
RCR2    EQU     0015H               ; Remote control register 2

;---------- Main program ------------------------------------------------------------------------

        CSEG                        ; [CODE SEGMENT]
        :
        MOV     RCR1,#01001010B     ; Select 1 tinst count clock, set "H" pulse width
                                      comparison
                                    ; value
        MOV     RCR2,#10011110B     ; Enable output and start counter, set cycle
                                      period,
                                    ; comparison value.
        :
        ENDS

;-------------------------------------------------------------------------------------------------

        END
```

CHAPTER 14 LCD CONTROLLER/DRIVER

This chapter describes the functions and operation of the LCD controller/driver.

14.1 Overview of LCD Controller/Driver

The LCD controller/driver includes 7 bytes of on-chip display data in memory, the contents of which control an LCD display via 14 segment and 4 common outputs. The function can drive an LCD display panel directly, using one of three selectable duty ratios.

■ LCD Controller/Driver Function

The LCD controller/driver function displays the contents of a display data memory directly to the LCD panel (liquid crystal display) by segment and common outputs.

- Up to 14 segment outputs (SEG0 to SEG13) and four common outputs (COM0 to COM3) may be used.
- Built-in display RAM: 7 bytes (14 × 4 bits)
- Three selectable duty ratios (1/2, 1/3, and 1/4). Not all duty ratios are available with all bias settings, however.
- Either the main or subclock can be selected as the drive clock.
- LCD can be driven directly.

Table 14.1-1 "Bias and Duty Ratio Combinations" shows the duty ratios available with each bias setting.

Table 14.1-1 Bias and Duty Ratio Combinations

Part number	Bias	1/2 duty ratio	1/3 duty ratio	1/4 duty ratio
MB89980 series	1/2 bias	O	X	X
	1/3 bias	X	O	O

O: Recommended mode

X: Do not use

Check:

The 1/2 bias mode cannot be used with devices that have internal voltage boosters (MB89980 series) because it requires an external divider resistor.

If P40/SEG0 to P47/SEG7, P60/SEG8 to P65/SEG13, P70/COM2, and P71/COM3 are set as output-only port pins (general-purpose outputs) as a mask option, they cannot be used as LCD segment and common outputs.

When P70/COM2 and P71/COM3 are used as output-only port pins, the 1/3 and 1/4 duty ratio output modes cannot be used.

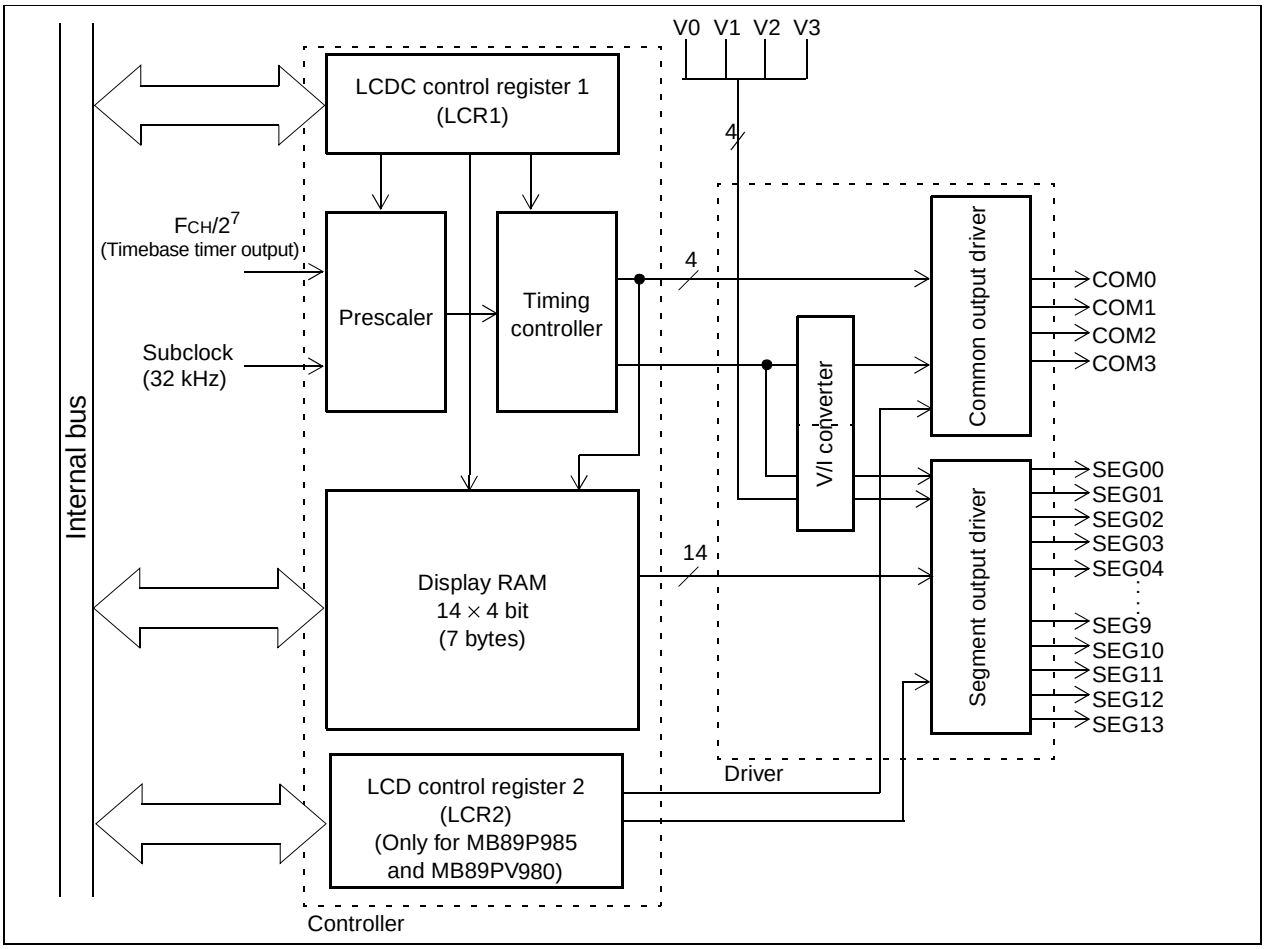
14.2 Block Diagram of LCD Controller/Driver

The LCD controller/driver is made up of the eight blocks listed below. Functionally, the circuit can be broken into two major sections: the controller section, which generates LCD segment and common signals based on the current contents of display RAM, and the driver section, which develops sufficient drive to operate the display.

- **LCD control register 1 (LCR1)**
- **LCD control register 2 (LCR2)**
- **Display RAM**
- **Prescaler**
- **Timing controller**
- **V/I converter**
- **Common output driver**
- **Segment output driver**

■ Block Diagram of LCD Controller/Driver

Figure 14.2-1 Block Diagram of LCD Controller/Driver



- **LCDC control register 1 (LCR1)**

  This register is used to select the frame clock (the clock used to generate the frame cycle), enable/disable operation in watch mode, control the LCD drive supply voltage, select display blanking/non-blanking, select the display mode, and select the LCD clock cycle.

- **LCDC control register 2 (LCR2) (only for MB89P985 and MB89PV980)**

  This register is used to control port/segment and port/common selection. It is used for MB89P985 and MB89PV980 only.

- **Display RAM**

  This 14 × 4-bit block of RAM controls the segment output signals. Its contents are automatically read out to the segment outputs in sync with the timing of the selected common signal.

- **Prescaler**

  The prescaler selects settings from 2 clocks and 4 frequencies to generate the frame frequency.

- **Timing controller**

  This block controls the segment and common signals based on the frame frequency and LCR1 register settings.

- **V/I converter**

  This circuit generates alternating current waveforms from the voltage signals it receives from the timing controller to drive the LCD.

- **Common output driver**

  Contains the drivers for the LCD common pins.

- **Segment output driver**

  Contains the drivers for the LCD segment pins.

- **LCD Controller/Driver Supply Voltage**

  The LCD driver supply voltage is taken from a voltage divider. The divider can be made up of internal or external resistors connected to the V0 to V3 pins.

14.2.1 LCD Controller/Driver Internal Divider Resistors

In devices that have internal divider resistors, the LCD driver supply voltage is taken from an internal voltage divider. (External divider resistors may also be used.)

■ Internal Divider Resistors

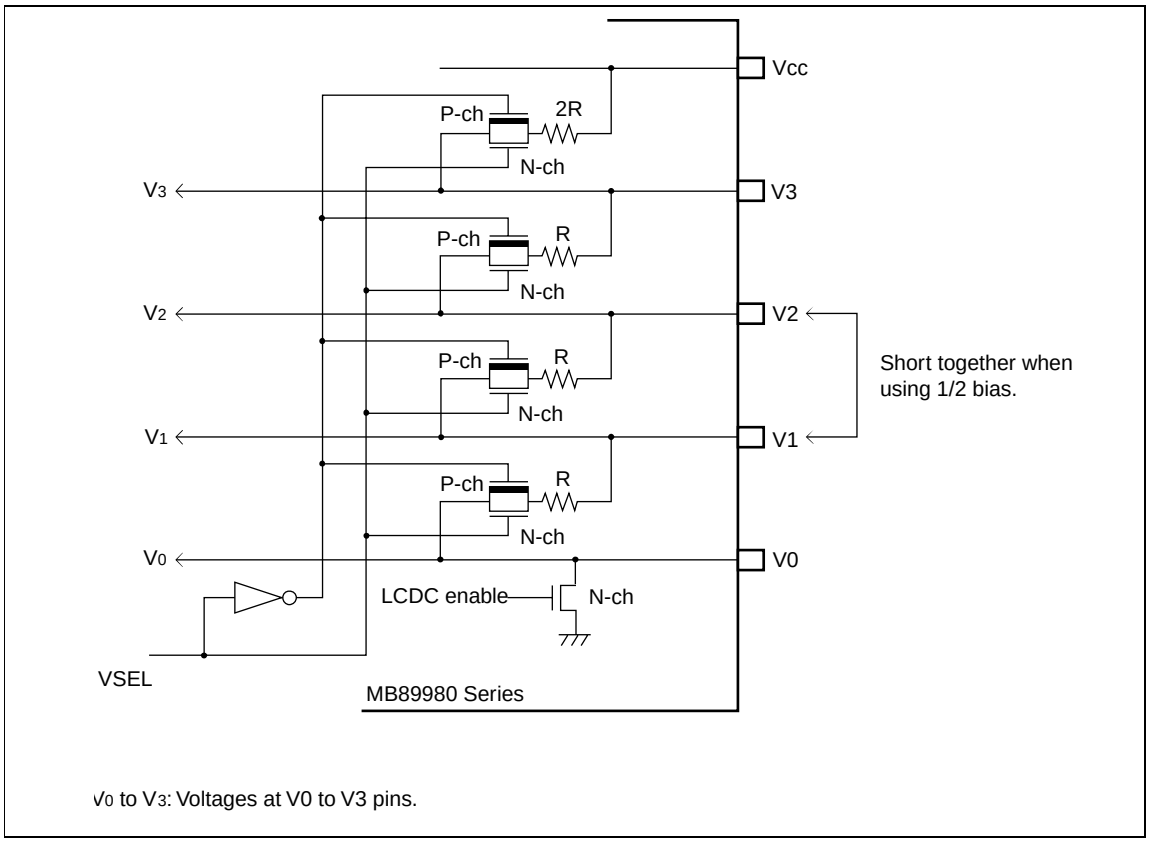
Devices have internal divider resistors. In these devices, external divider resistors may also be connected at pins V0 through V3.

The selection of internal or external resistors is made by the drive supply voltage control bit of LCDC control register 1 (LCR1: VSEL). VSEL = 1 connects the internal resistors. Set VSEL to "1" when you want to use the internal resistors only (when no external resistors are connected).

The LCDC enable is inactive when LCD operation is stopped (LCR1: MS1 = MS0 = 00_B), and when operation is stopped (LCR1: LCEN = 0) in watch mode (STBC: TMD = 1). Pin V2 and V1 should be shorted together when using the 1/2 bias setting.

Figure 14.2-2 "Internal Voltage Divider Equivalent Circuit" shows an equivalent circuit of the internal voltage divider.

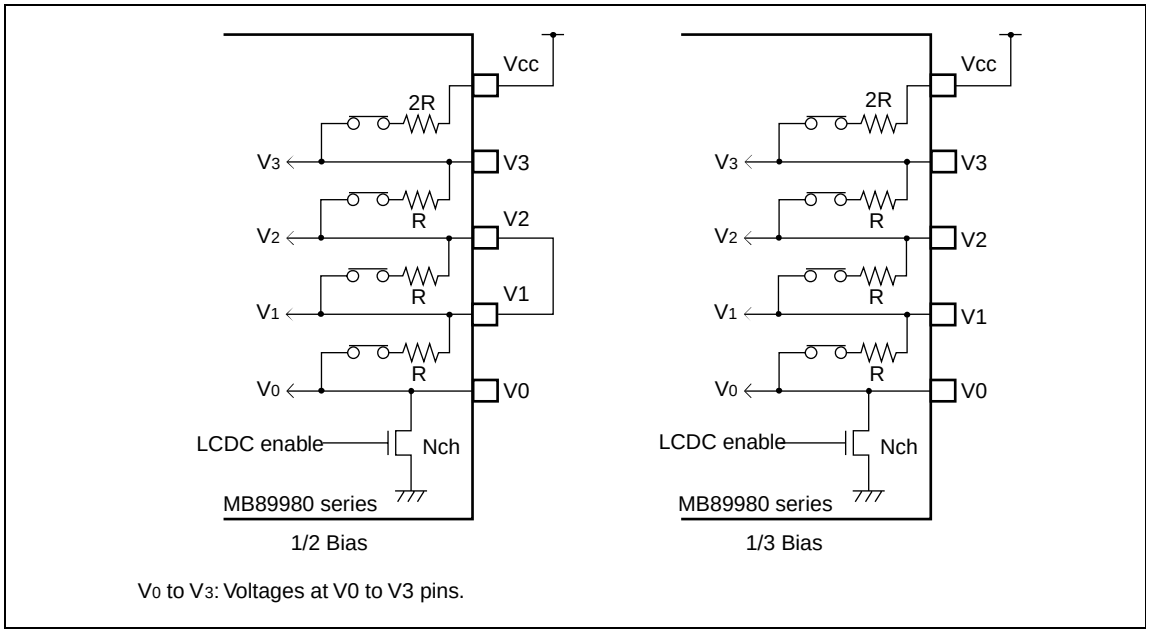
Figure 14.2-2 Internal Voltage Divider Equivalent Circuit



■ Use of Internal Voltage Divider Resistors

Figure 14.2-3 "Use of Internal Voltage Divider Resistors" shows the voltage divider circuits for 1/2 and 1/3 bias. As shown in this figure, in the 1/2 bias mode (with LCDC enabled) V2 and V1 will be 1/2 of V3 (V3 is the LCD operating voltage, which is VCC/2 in this configuration). In the 1/3 bias mode, V1 is 1/3 of V3, and V2 is 2/3 of V3.

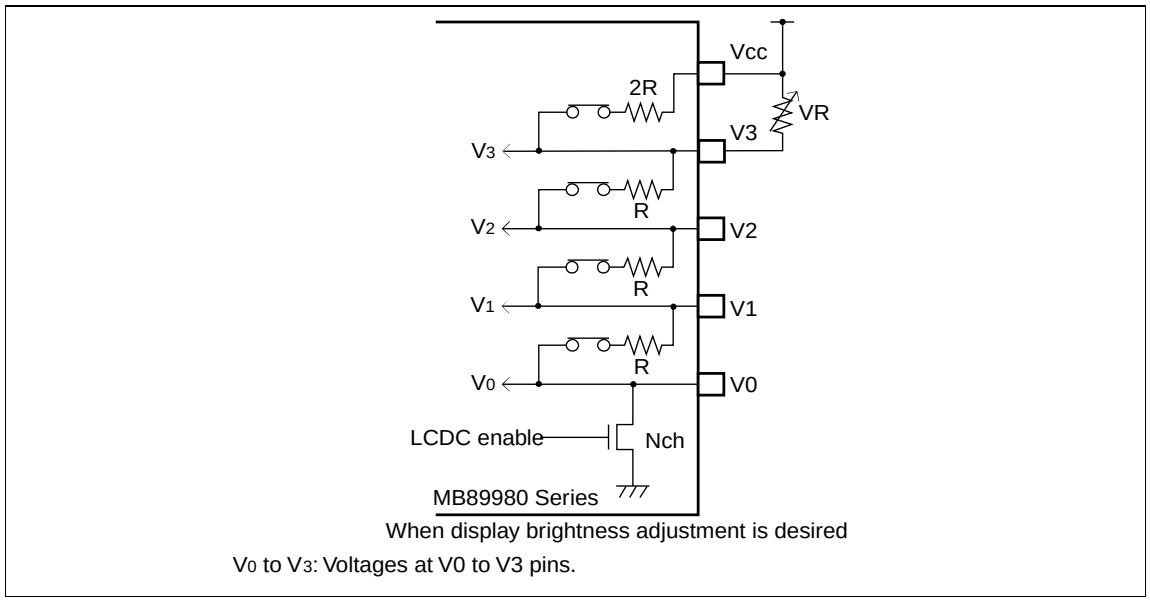
Figure 14.2-3 Use of Internal Voltage Divider Resistors



■ Display Brightness Adjustment when Internal Divider Resistors Are Used

When internal divider resistors do not provide sufficient LCD display brightness, connect an external brightness adjust variable resistor between VCC and V3 as shown in Figure 14.2-4 "Use of Internal Voltage Divider Resistors with Brightness Adjustment".

Figure 14.2-4 Use of Internal Voltage Divider Resistors with Brightness Adjustment



Note:

During LCD operation, the 2R internal resistance will be in the divider circuit, and VR will be in parallel with this resistor.

14.2.2 LCD Controller/Driver External Divider Resistors

External voltage divider resistors can also be used with devices that have internal divider resistors. Display brightness can be adjusted by a variable resistor connected between the V_{CC} and V3 pins.

■ External Divider Resistors

When you are using a device without a voltage booster, but do not wish to use the internal divider resistors, external voltage divider resistors can be connected at the LCD drive voltage supply pins (V0 to V3). Figure 14.2-5 "External Voltage Divider Resistor Connections" shows connections for external divider resistors for the two biasing modes, and Table 14.2-1 "LCD Drive Voltages and Biasing Modes" lists the corresponding LCD drive voltages.

Figure 14.2-5 External Voltage Divider Resistor Connections

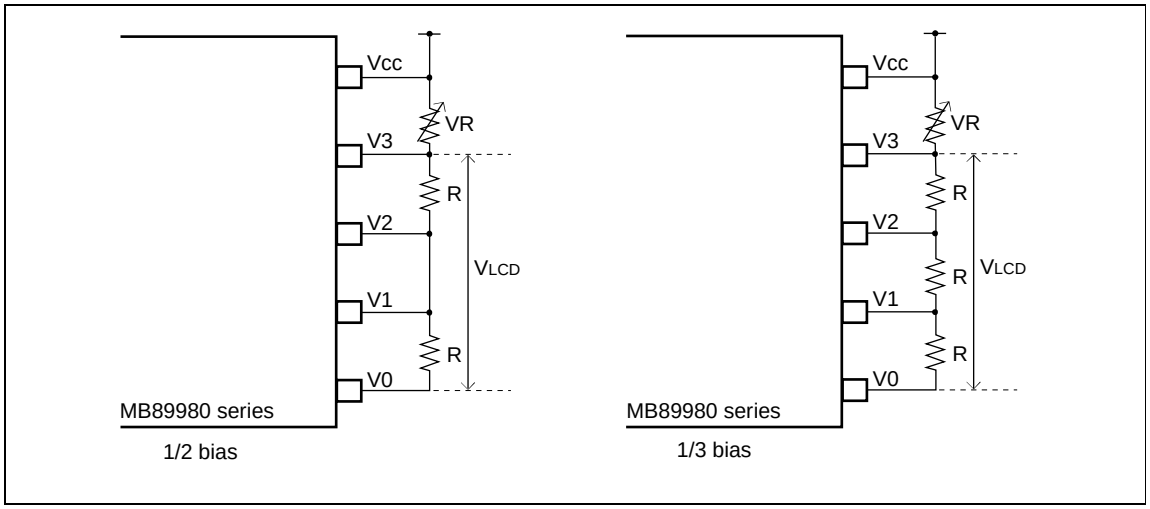


Table 14.2-1 LCD Drive Voltages and Biasing Modes

	V3	V2	V1	V0
1/2 bias	V_{LCD}	$1/2V_{LCD}$	$1/2V_{LCD}$	GND
1/3 bias	V_{LCD}	$2/3V_{LCD}$	$1/3V_{LCD}$	GND

V0 to V3: Voltages at pins V0 to V3.

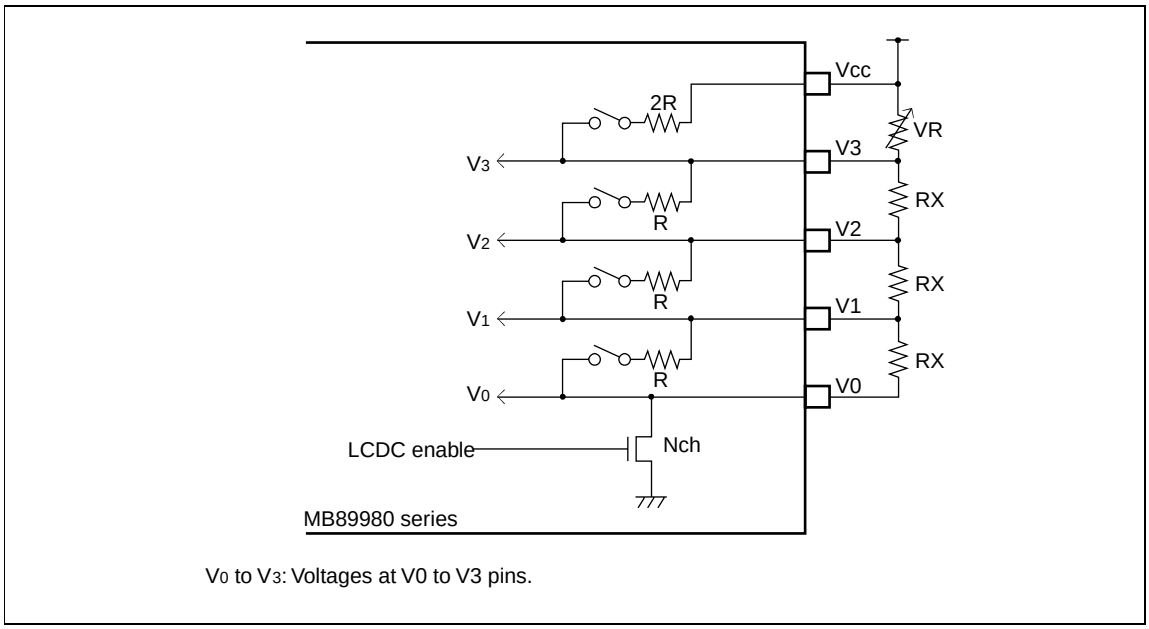
V_{LCD}: LCD operating voltage

■ Using External Divider Resistors

Internally, the V0 pin is connected through a transistor to V_{SS} (GND). Therefore, when external voltage divider resistors are used, the current flow to the external resistors with the LCD controller off can be cut off by connecting the V_{SS} end of the divider to the V0 pin only. Figure 14.2-6 "External Voltage Divider Resistor Connections" shows an external voltage divider

resistor connection.

Figure 14.2-6 External Voltage Divider Resistor Connections



V_0 to V_3: Voltages at V0 to V3 pins.

1. To preclude the external voltage divider from being affected by the internal divider resistors, the LCDC control register drive voltage control bit (LCR1: VSEL) must be written to "0" to isolate it from the entire internal voltage divider.
2. With the internal voltage divider thus isolated, writing the display mode select bits (MS1 and MS0) of the LCR1 register to any state other than "00_B" will turn on the LCDC enable transistor (Q1), causing current to flow in the external divider resistors.
3. Writing the display mode select bits (MS1 and MS0) to "00_B" will turn off the LCDC enable transistor (Q1), and current will stop flowing in the external resistors.

Note:

The resistance of RX in the external divider depends on the LCD used. Select an appropriate value.

14.3 Structure of LCD Controller/Driver

This section describes the pins, pin block diagrams, register, and display RAM of the LCD controller/driver.

■ LCD Controller/Driver Pins

The LCD controller/driver uses 4 common output pins (COM0 to COM3), 14 segment output pin (SEG0 to SEG13) and 4 LCD driving power supply pins (V0 to V3).

❍ COM0, COM1, P70/COM2, and P71/COM3 Pins

P70/COM2 and P71/COM3 pins can function either as output-only ports (P70 and P71) and LCD common output pins (COM2 and COM3). The selection, however is made by mask option in MB89983 and by LCD control register 2 (LCR2) in MB89P985 and MB89PV980.

Check:

When the pins are used as LCD common outputs, the corresponding port data register bits (PDR7: bits 0 and 1) should be set to "1" to turn the output transistor "off". (COM0 and COM1 are dedicated LCD common output pins.)

❍ P40/SEG0 to P47/SEG7 and P60/SEG8 to P65/SEG13

P40/SEG0 to P47/SEG7 and P60/SEG8 to P65/SEG13 can function either as output-only ports (P40 to P47 and P60 to P65) and LCD segment output pins (SEG0 to SEG7 and SEG8 to SEG13). The selection, however is made by mask option in MB89983 and by LCD control register 2 (LCR2) in MB89P985 and MB89PV980.

Check:

When these pins are used as LCD segment outputs, the corresponding port data registers (PDR4 and PDR6) should be set to all "1s" to turn the output transistors off.

❍ V0 to V3

These pins are the LCD driving power supply pins.

■ Block Diagrams of LCD Controller/Driver Pin

Figure 14.3-1 Block Diagram of LCD Controller/Driver Pin (Dedicated Common Output Pins)

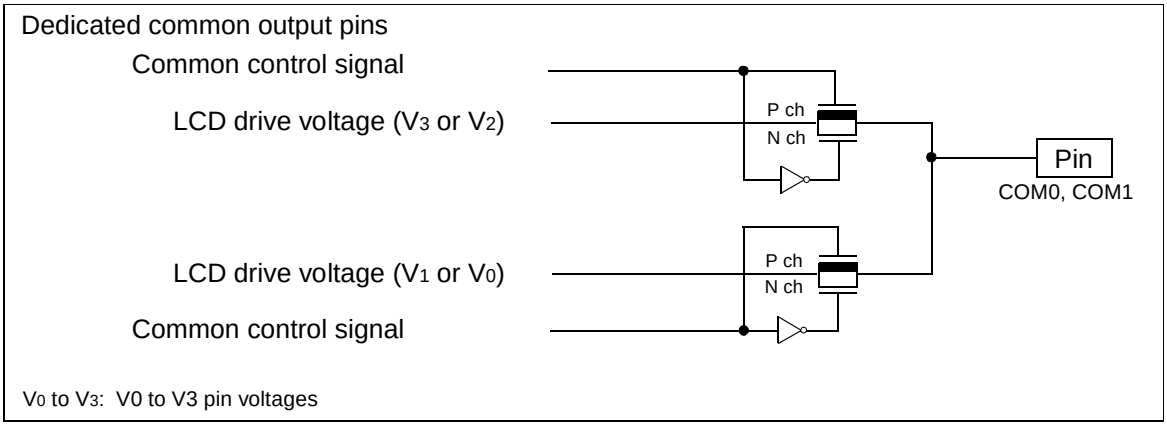


Figure 14.3-2 Block Diagram of LCD Controller-Driver Pin for MB89983 (Dual Function Common/ Segment Output Pins)

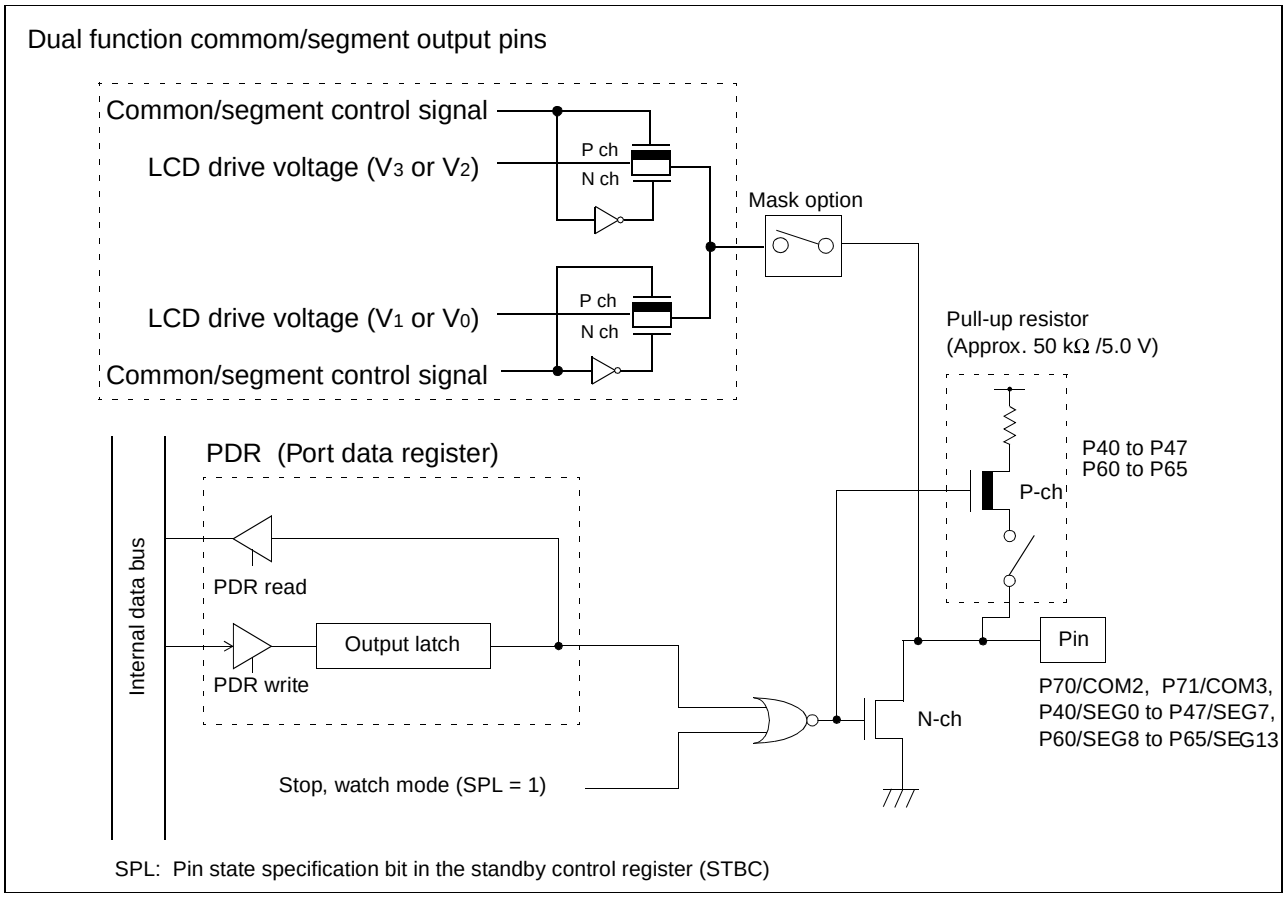
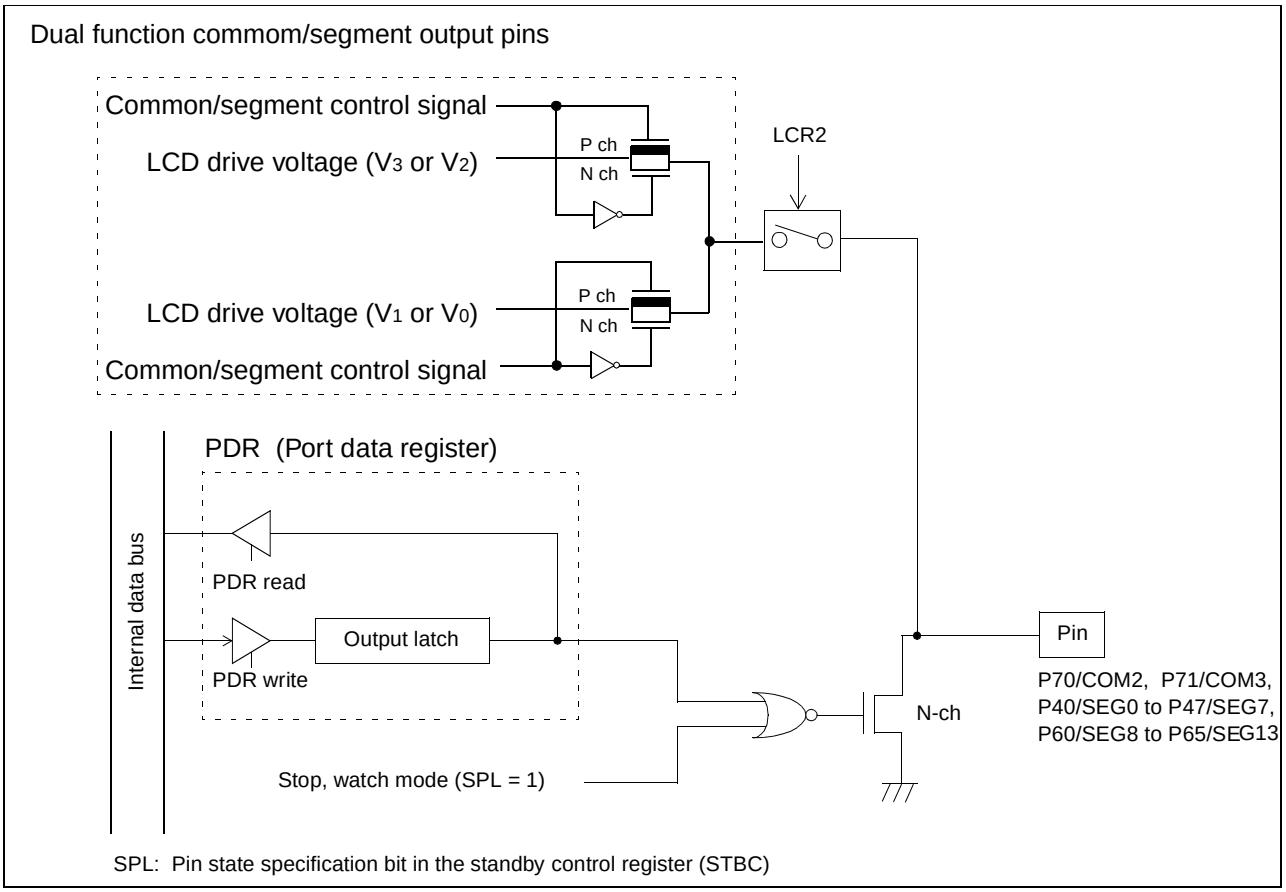


Figure 14.3-3 Block Diagram of LCD Controller-Driver Pin for MB89P985 and MB89PV980 (Dual Function Common/Segment Output Pins)



Check:

Do not select the pull-up resistor option on pins used for common or segment outputs or capacitor connection pins.

■ LCD Controller/Driver Register

Figure 14.3-4 LCD Controller/Driver Register

LCR1 (LCD Control Register 1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0072H	CSS	LCEN	VSEL	BK	MS1	MS0	FP1	FP0	00010000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

LCR2 (LCD Control Register 2)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0073H	SC3	—	—	LS4	LS3	LS2	LS1	—	0--0000-B
	R/W	—	—	R/W	R/W	R/W	R/W	—	

R/W : Readable and writable
X : Indeterminate

■ LCD Controller/Driver RAM

LCD controller/driver has 14 × 4-bit of internal display RAM in which the data used to generate the segment output signals is stored.

14.3.1 LCD Control Register (LCR1)

LCD Control Register 1 (LCR1) is used to select the frame cycle, enable/disable operation in watch mode, control the LCD drive supply voltage, select display blanking/non-blanking, and select the display mode.

■ LCDC Control Register 1 (LCR1)

Figure 14.3-5 LCDC Control Register 1 (LCR1)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0072H	CSS	LCEN	VSEL	BK	MS1	MS0	FP1	FP0	00010000B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

FP1	FP0	Frame cycle Selection bits	
		Main clock (CSS=0)	Subclock (CSS=1)
0	0	$F_{CH}/(2^{12} \times N)$ (256 Hz)	$F_{CL}/(2^{5} \times N)$ (256 Hz)
0	1	$F_{CH}/(2^{13} \times N)$ (128 Hz)	$F_{CL}/(2^{6} \times N)$ (128 Hz)
1	0	$F_{CH}/(2^{14} \times N)$ (64 Hz)	$F_{CL}/(2^{7} \times N)$ (64 Hz)
1	1	$F_{CH}/(2^{15} \times N)$ (32 Hz)	$F_{CL}/(2^{8} \times N)$ (32 Hz)

( ) : Values for F_{CH}=4.2 MHz, F_{CL}=32.768 kHz, and N=4
N : Number of time divisions
F_{CH} : Main clock source oscillation
F_{CL} : Subclock source oscillaion

MS1	MS0	Display mode selection bits
0	0	Stop LCD operation
0	1	1/2 duty ratio output mode (time division N = 2)
1	0	1/3 duty ratio output mode (time division N = 3)
1	1	1/4 duty ratio output mode (time division N = 4)

BK	Display blanking selection bit
0	Display unblanked
1	Display blanked

VSEL	Drive supply voltage control bit
0	External divider resistors used (internal divider resistors isolated)
1	Internal divider rediditors used.

LCEN	Watch mode operation enable bit
0	Stop in watch mode
1	Run in watch mode (also)

CSS	Frame cycle generate clock selection bit
0	Main clock
1	Subclock

R/W : Readable and writable
W : Write only
□ : Initial value

Table 14.3-1 LCDC Control Register (LCR1) Bit Functions

Bit		Function
Bit 7	CSS: Frame cycle generation clock selection bit	Selects the frame clock, which generates the frame cycle for LCD display. "0" selects the output of the timebase timer derived from the main clock divided by $F_{CL}2^7$; "1" selects the subclock as the frame clock. **Check:** The timebase timer output may not be selected as the frame clock in the main-stop and subclock modes because the main clock oscillator is stopped in those modes. **Note:** When the timebase timer output is selected, the frame clock is not affected when clock speed is changed via the speed shift function. (The timebase timer's count clock is not supplied through the speed shift function.)
Bit 6	LCEN: Watch mode operation enable bit	Determines whether the LCD controller/driver will operate in watch mode. If this bit is "1", the LCD display will continues to operate after the system goes to watch mode; if it is "0", the LCD will cease operation. **Check:** To use the display in watch mode, the subclock must be selected as the frame clock (CSS = 1).
Bit 5	VSEL: LCD drive supply voltage control bit	• In devices that have an internal divider resistor, the VSEL bit controls the divider current path continuity. A "1" in this bit completes the divider current path; a "0" opens it. This bit must be "0" when external divider resistors are used.
Bit 4	BK: Display blanking selection bit	Blanks/unblanks the LCD. Setting this bit to "1" (blank) outputs a "deselect" waveform to the LCD segments (which blanks the display).
Bit 3 Bit 2	MS1, MS0: Display mode selection bits	Select one of three output waveform duty ratio modes. The mode selected affects the common pins used. Setting both bits to "0" turns "off" the display (stops LCD controller/driver display operation). **Check:** Before going to a mode in which the selected frame cycle generate clock oscillator is stopped (stop mode, etc.), these bits should be written to "00_B" to turn off the display.
Bit 1 Bit 0	FP1, FP0: Frame cycle selection bits	These bits select one of four LCD display frame cycles. **Check:** To determine this register setting, calculate the optimum frame frequency for the LCD module you are using. Note that the frame cycle is a function of main clock frequency.

14.3.2 LCD Control Register (LCR2)

LCD Control Register 2 (LCR2) is used to make the port/segment and port/common selection in MB89P985 and MB89PV980 only. In MB89983, the port/segment and port/common selection is made by mask option, so that it is no need to use this register.

■ LCD Control Register 2 (LCR2)

Figure 14.3-6 LCD Control Register 2 (LCR2)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0073H	SC3	—	—	LS4	LS3	LS2	LS1	—	0--0000-B
	R/W	—	—	R/W	R/W	R/W	R/W	—	

LS1	Port/Segment Selection bit	
	Port	Segment
0	P40 to P43	—
1	—	SEG0 to SEG03

LS2	Port/Segment Selection bit	
	Port	Segment
0	P44 to P47	—
1	—	SEG04 to SEG07

LS3	Port/Segment Selection bit	
	Port	Segment
0	P60 to P61	—
1	—	SEG08 to SEG09

LS4	Port/Segment Selection bit	
	Port	Segment
0	P62 to P65	—
1	—	SEG10 to SEG13

SC3	Port/Common Selection bits	
	Port	Common
0	P70, P71	—
1	—	COM2, COM3

R/W : Readable and writable
— : Not used
□ : Initial value

Table 14.3-2 LCD Control Register 2 (LCR2) Bit Functions

Bit		Function
Bit 7	SC3: Port/Common selection bit	• These bits are selected for an either general output port or commmon output pin for LCD.
Bit 6 Bit 5	Unused bits	• The read value is indeterminate. • Writing to this bit has no effect on the operation.
Bit 4 Bit 3 Bit 2 Bit 1	LS4, LS3, LS2, LS1: Port/Segment selection bits	• These bits are selected for an either general output port or segment output pin for LCD.
Bit 0	Unused bit	• The read value is indeterminate. • Writing to this bit has no effect on the operation.

14.3.3 Display RAM

Display RAM consists of 14 × 4-bit (7 bytes) of display data memory used to generate the segment output signals.

■ Display RAM and Output Pins

The contents of display RAM are automatically read out and output via the segment outputs in sync with the selected common signal timing. A "1" bit is converted to a "select" (display on) voltage and a "0" to a "deselect" (display off) voltage. Since the operation of the LCD is not directly related to the operation of the CPU, display RAM read/write timing can be set by the user. The SEG8 to SEG23 pins that are not made dedicated segment outputs by mask option selection may be used as general-purpose output-only port pins, and the RAM that goes with those pins may be used as regular RAM. (See Table 14.3-3 "Segment Outputs, Display RAM Locations, and Sharing Port Pins".)

Table 14.3-4 "Common Outputs and Display RAM Bits Used in Each Duty Ratio Mode" shows the relationship between duty ratio mode, common outputs, and display RAM.

Figure 14.3-7 "Segment/Common Output Pins and Corresponding Display RAM" shows which display RAM bits are associated with each segment and common output pin.

Figure 14.3-7 Segment/Common Output Pins and Corresponding Display RAM

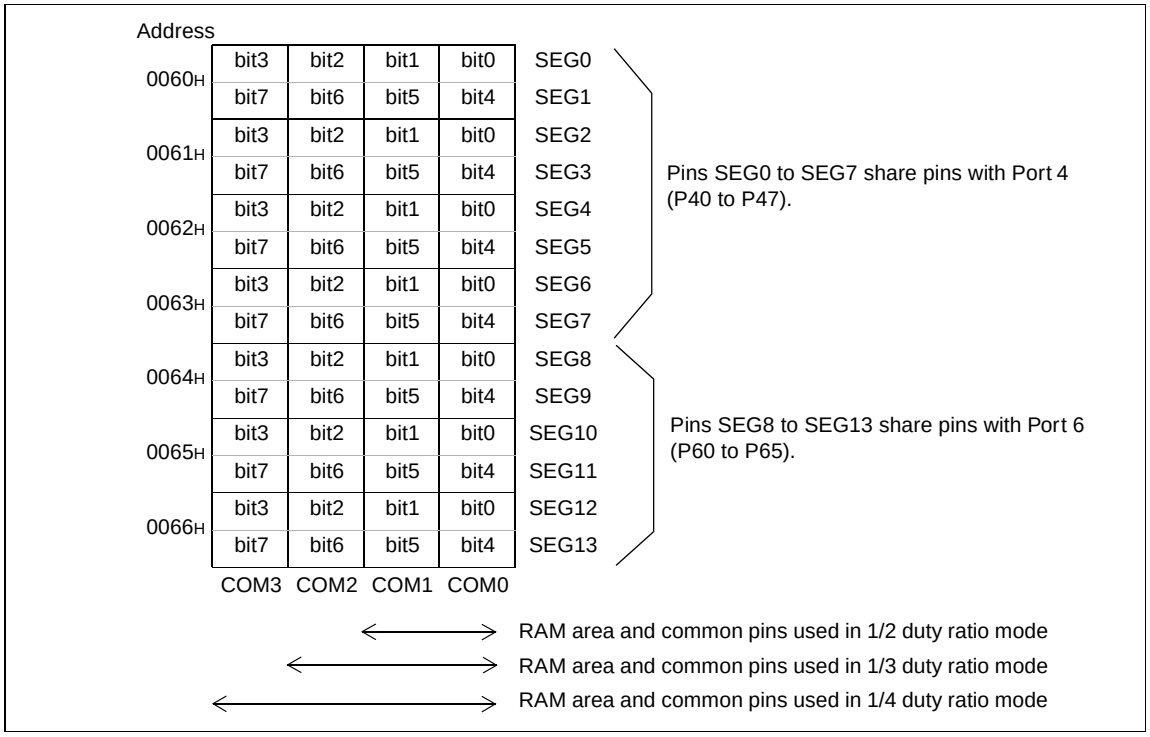


Table 14.3-3 Segment Outputs, Display RAM Locations, and Sharing Port Pins

Segment/Common Output Pins Used (Mask Option)	Corresponding Display RAM Area	General-Purpose Ports Sharing Same Pins
SEG10 to SEG13 (4 pins) COM0, COM1	65_H to 66_H	P40 to P47, P60, P61, P70, P71 (10 + 2 pins)
SEG8 to SEG13 (6 pins) COM0, COM1	64_H to 66_H	P40 to P47, P70, P71 (8 + 2 pins)
SEG4 to SEG13 (10 pins) COM0, COM1	62_H to 66_H	P40 to P43, P70, P71 (4 + 2 pins)
SEG0 to SEG13 (14 pins) COM0, COM1	60_H to 66_H	P70, P71 (2 pins)
SEG10 to SEG13 (4 pins) COM0 to COM3	65_H to 66_H	P40 to P47, P60, P61 (10 pins)
SEG8 to SEG13 (6 pins) COM0 to COM3	64_H to 66_H	P40 to P47 (8 pins)
SEG4 to SEG13 (10 pins) COM0 to COM3	62_H to 66_H	P40 to P43 (4 pins)
SEG0 to SEG13 (14 pins) COM0 to COM3	60_H to 66_H	None

Note:

Locations in the display RAM area that are not required for display data can be used as regular RAM.

Table 14.3-4 Common Outputs and Display RAM Bits Used in Each Duty Ratio Mode

Duty Ratio Setting	Common Outputs Used	Display Data Bit Used							
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
1/2	COM0 to COM1 (2 pins)	-	-	O	O	-	-	O	O
1/3*	COM0 to COM2 (3 pins)	-	O	O	O	-	O	O	O
1/4*	COM0 to COM3 (4 pins)	O	O	O	O	O	O	O	O

O: Used

-: Not used

*: The LCD common output option must be selected for pins COM2 and COM3 (mask option).

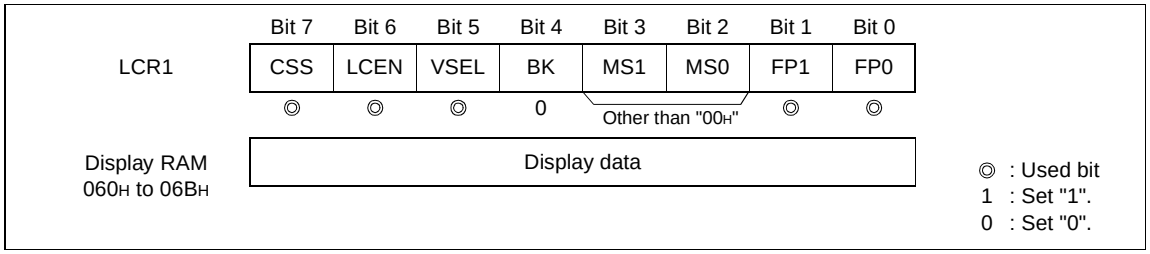
14.4 Operation of LCD Controller/Driver

The LCD controller/driver provides the necessary control and drive for an LCD display.

■ Operation of LCD Controller/Driver

Figure 14.4-1 "LCD Controller/Driver Settings" shows the settings required to operate the LCD display.

Figure 14.4-1 LCD Controller/Driver Settings



Once the above settings have been made, if the selected clock for frame cycle generation is running, LCD panel driving waveforms reflecting the contents of display RAM will be output at the segment and common output pins (COM0 to COM3 and SEG0 to SEG13).

Although the clock for frame period generation can be switched even while the LCD is displaying data, the display may flicker when the switching occurs. This can be avoided by temporarily blanking the display (LCR1: BK = 1), etc. while switching.

The display driving output is a two-frame a.c. waveform for which the bias level and display duty cycle is selected by settings.

When the P70/COM2 and P71/COM3 pins are set as COM outputs, deselection levels are output in the waveforms at the COM2 and COM3 outputs in 1/2 duty ratio operation, and at the COM3 output in 1/3 duty ratio operation.

When LCD display operation is stopped (LCR1: MS1 = MS0 = 00B), and during reset, all COM and SEG output pins are taken "L".

Check:

If the selected frame cycle generate clock were to stop while the LCD display is operating, the circuit that converts the waveform from d.c. to a.c. would also stop, causing a d.c. voltage to be applied to the liquid crystal cells. The LCD display must be therefore be stopped before the clock is stopped. The conditions under which the main clock (timebase timer) and subclock are stopped are a function of the clock mode and standby mode. Also note that when the timebase timer is selected as the frame clock source (LCR1: CSS = 0), clearing the timebase timer will affect the frame cycle.

■ LCD Driving Waveforms

It is characteristic of LCDs that applying d.c. drive to the panel can cause electrochemical degradation of the material used in the LCD cells. For this reason, the LCD controller/driver includes a circuit to convert the original driving waveform to a two-frame a.c. output waveform (zero d.c. bias) to drive the LCD. There are three types of output waveform:

- 1/2 bias, 1/2 duty ratio output waveform (only devices without voltage boosters)
- 1/3 bias, 1/3 duty ratio output waveform
- 1/3 bias, 1/4 duty ratio output waveform

14.4.1 Output Waveforms during LCD Controller/Driver Operation (1/2 Duty Ratio)

The display drive output is a multiplex drive-type two-frame a.c. waveform. In the 1/2 duty ratio mode, the only common outputs are COM0 and COM1. (COM2 and COM3 are not used.)

■ Output Waveforms during LCD Controller/Driver Operation (1/2 Duty Ratio)

❍ 1/2 Bias, 1/2 duty output waveform

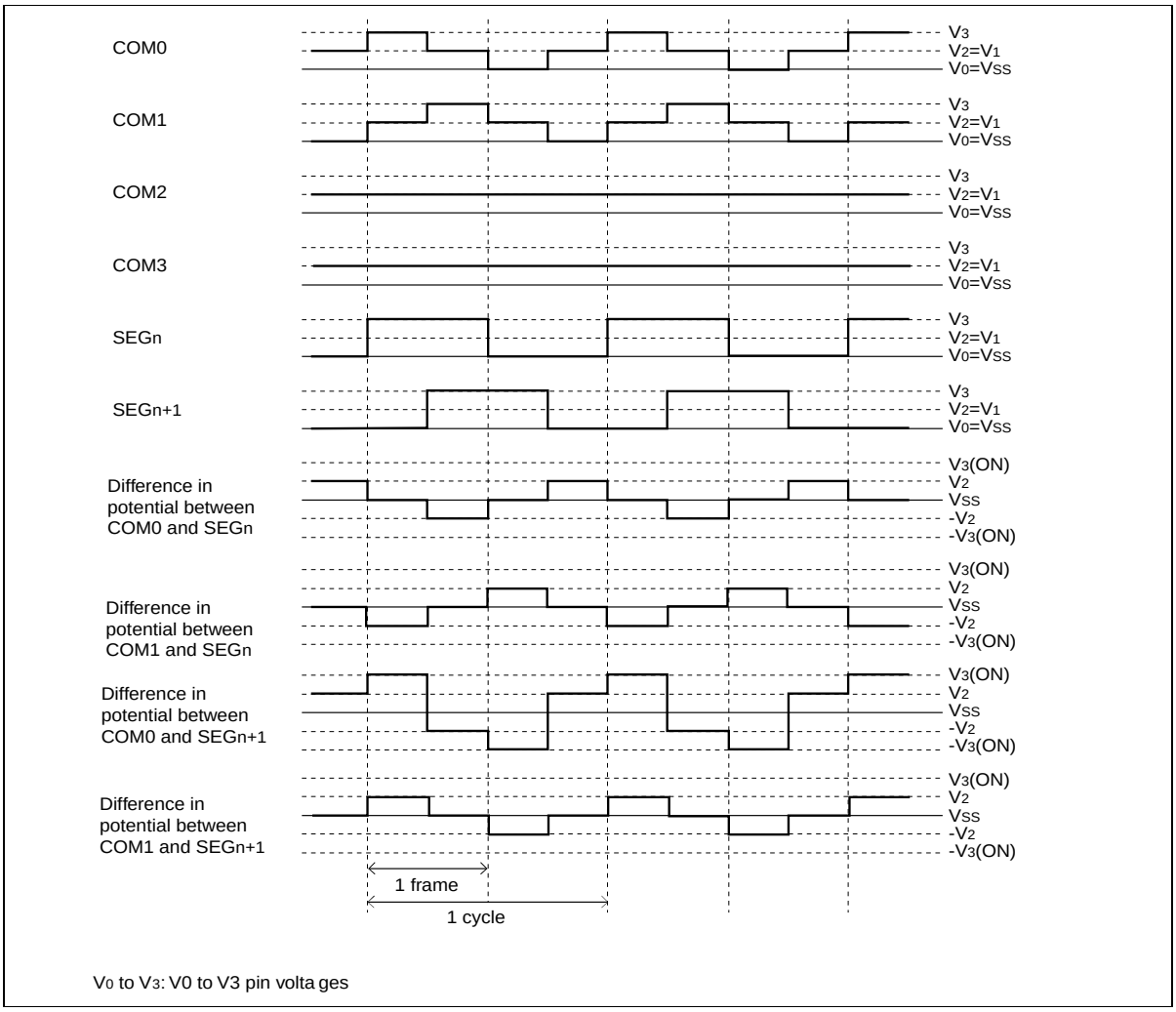
The maximum potential difference exists between a segment output and the corresponding common output when the segment (LCD cell) is "turned on". Figure 14.4-2 "Output Waveforms, 1/2 Bias and 1/2 Duty Ratio Example" shows the output waveforms for the display RAM contents listed in Table 14.4-1 "Display RAM Contents Example".

Table 14.4-1 Display RAM Contents Example

Segment	Display RAM contents			
	COM3	COM2	COM1	COM0
SEGn	-	-	0	0
SEGn+1	-	-	0	1

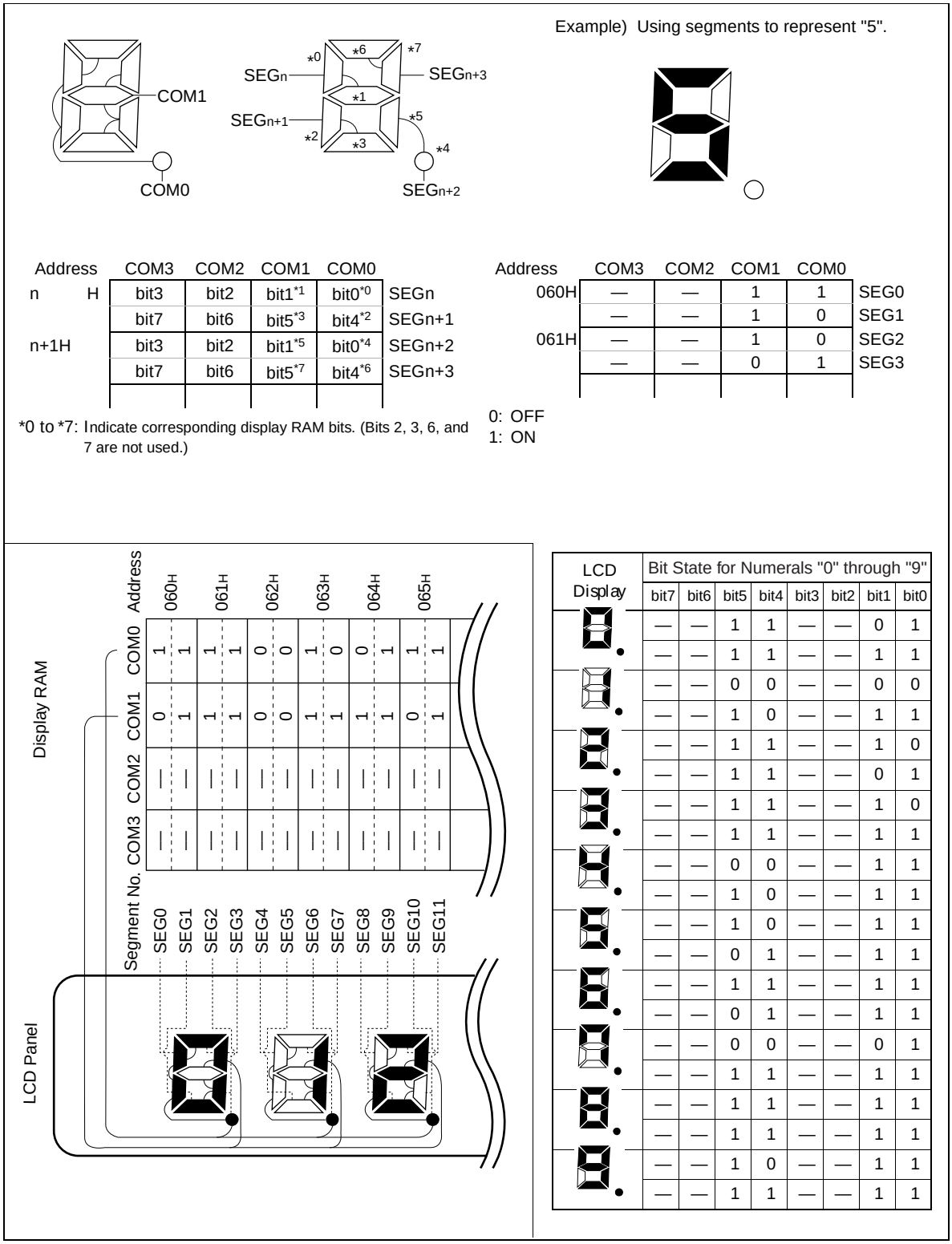
-: Not used

Figure 14.4-2 Output Waveforms, 1/2 Bias and 1/2 Duty Ratio Example



❍ LCD panel connections and display data example (1/2 duty ratio drive mode)

Figure 14.4-3 Segment/Common Connections, Data States and Corresponding Display



Address		COM3	COM2	COM1	COM0	
n	H	bit3	bit2	bit1*1	bit0*0	SEGn
		bit7	bit6	bit5*3	bit4*2	SEGn+1
n+1H		bit3	bit2	bit1*5	bit0*4	SEGn+2
		bit7	bit6	bit5*7	bit4*6	SEGn+3

*0 to *7: Indicate corresponding display RAM bits. (Bits 2, 3, 6, and 7 are not used.)

Address	COM3	COM2	COM1	COM0	
060H	—	—	1	1	SEG0
	—	—	1	0	SEG1
061H	—	—	1	0	SEG2
	—	—	0	1	SEG3

0: OFF
1: ON

LCD Display	Bit State for Numerals "0" through "9"							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	—	—	1	1	—	—	0	1
	—	—	1	1	—	—	1	1
1	—	—	0	0	—	—	0	0
	—	—	1	0	—	—	1	1
2	—	—	1	1	—	—	1	0
	—	—	1	1	—	—	0	1
3	—	—	1	1	—	—	1	0
	—	—	1	1	—	—	1	1
4	—	—	0	0	—	—	1	1
	—	—	1	0	—	—	1	1
5	—	—	1	0	—	—	1	1
	—	—	0	1	—	—	1	1
6	—	—	1	1	—	—	1	1
	—	—	0	1	—	—	1	1
7	—	—	0	0	—	—	0	1
	—	—	1	1	—	—	1	1
8	—	—	1	1	—	—	1	1
	—	—	1	1	—	—	1	1
9	—	—	1	0	—	—	1	1
	—	—	1	1	—	—	1	1

14.4.2 Output Waveforms during LCD Controller/Driver Operation (1/3 Duty Ratio)

In the 1/3 duty ratio mode, the COM0, COM1 and COM2 outputs are used by the display. COM3 is not used.

■ Output Waveforms during LCD Controller/Driver Operation (1/3 Duty Ratio)

❍ 1/3 bias, 1/3 duty output waveform

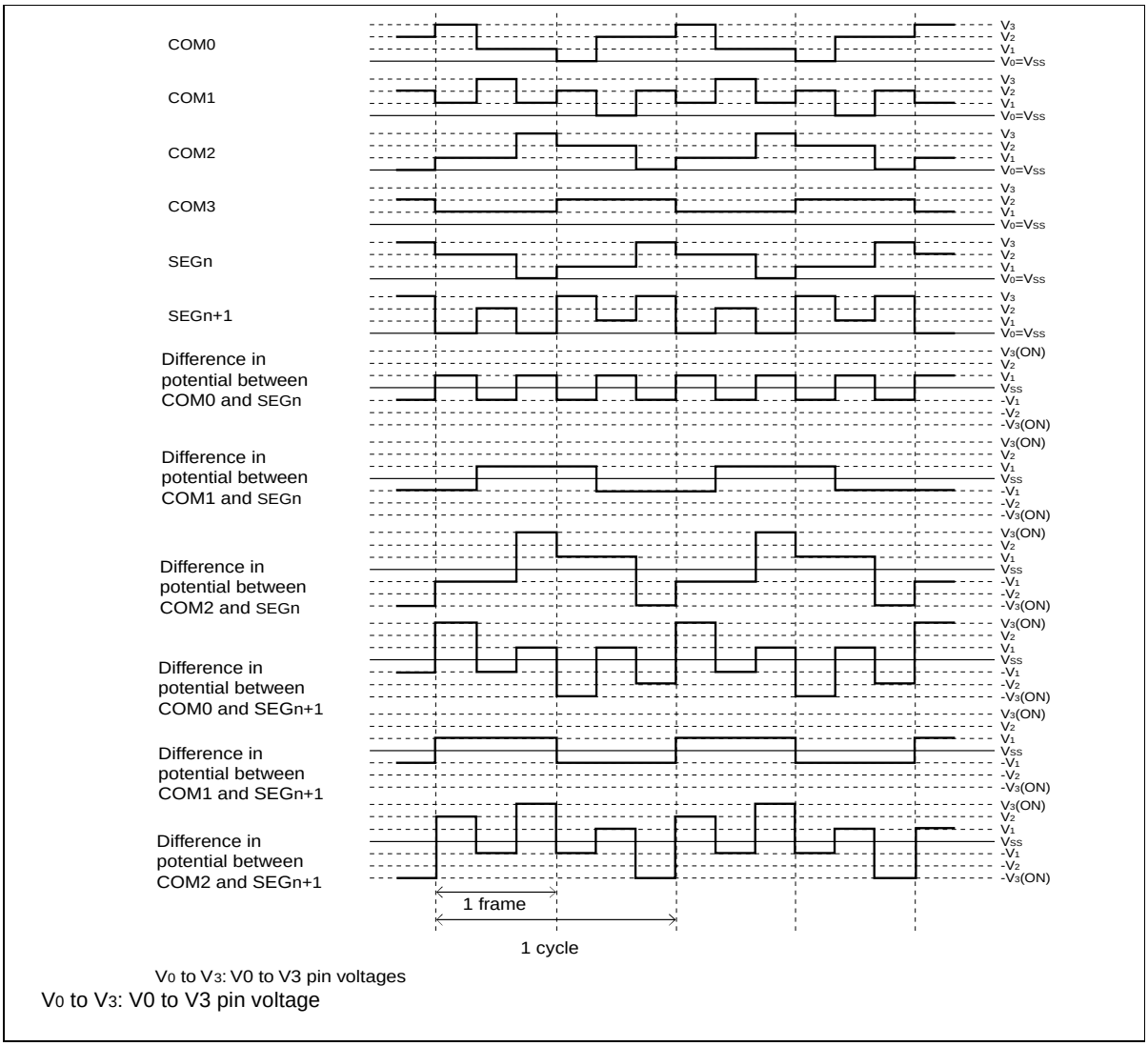
The maximum potential difference exists between a segment output and the corresponding common output when the segment (LCD cell) is "turned on". Figure 14.4-4 "Output Waveforms, 1/3 Bias and 1/3 Duty Ratio Example" shows the output waveforms for the display RAM contents listed in Table 14.4-2 "Display RAM Contents Example".

Table 14.4-2 Display RAM Contents Example

Segment	Display RAM contents			
	COM3	COM2	COM1	COM0
SEGn	-	1	0	0
SEGn+1	-	1	0	1

-: Not used

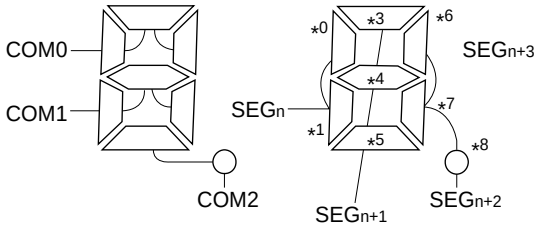
Figure 14.4-4 Output Waveforms, 1/3 Bias and 1/3 Duty Ratio Example



V0 to V3: V0 to V3 pin voltage

❍ LCD panel connections and display data example (1/3 duty ratio drive mode)

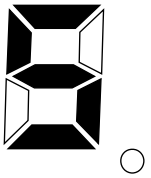
Figure 14.4-5 Segment/Common Connections, Data States and Corresponding Display



Address		COM3	COM2	COM1	COM0	
n	H	bit3	bit2*2	bit1*1	bit0*0	SEGn
		bit7	bit6*5	bit5*4	bit4*3	SEGn+1
n+1H		bit3	bit2*8	bit1*7	bit0*6	SEGn+2

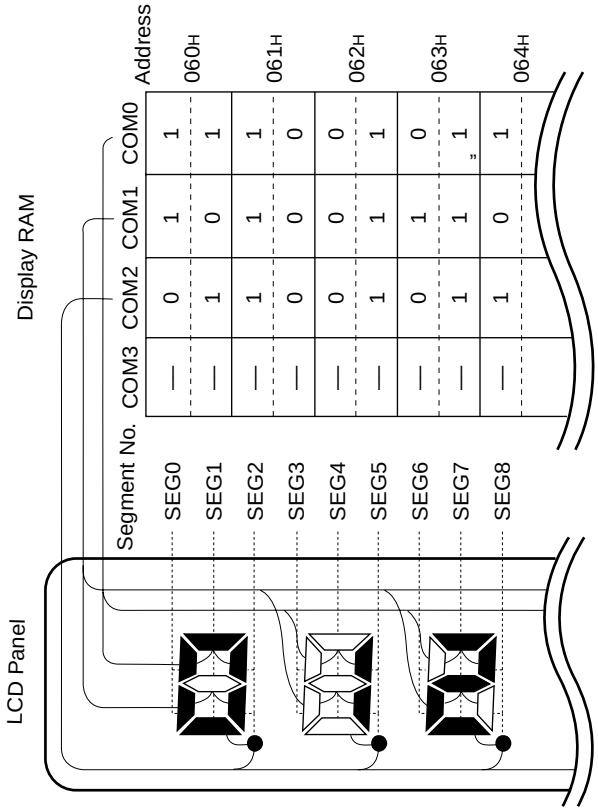
*0 to *8: Indicate corresponding display RAM bits. (Bit3 and 7 and *2 are not used.)

Example) Using segments to represent "5".



Address	COM3	COM2	COM1	COM0	
060H	—	0	0	1	SEG0
	—	1	1	1	SEG1
061H	—	0	1	0	SEG2
	—	0	0	1	SEG3
062H	—	1	1	1	SEG4
	—	0	1	0	SEG5

0: OFF
1: ON



LCD Display	Bit States for Numerals "0" through "9"							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0.	—	1	0	1	—	0	1	1
	—	0	1	1	—	1	1	1
	—	1	1	1	—	1	0	1
1.	—	0	0	0	—	0	0	0
	—	0	0	0	—	1	1	1
	—	1	1	1	—	0	0	0
2.	—	1	1	1	—	0	1	0
	—	0	1	0	—	1	0	1
	—	1	0	1	—	1	1	1
3.	—	1	1	1	—	0	0	0
	—	0	0	0	—	1	1	1
	—	1	1	1	—	1	1	1
4.	—	0	1	0	—	0	0	1
	—	0	0	1	—	1	1	1
	—	1	1	1	—	0	1	0
5.	—	1	1	1	—	0	0	1
	—	0	0	1	—	1	1	0
	—	1	1	0	—	1	1	1
6.	—	1	1	1	—	0	1	1
	—	0	1	1	—	1	1	0
	—	1	1	0	—	1	1	1
7.	—	0	0	1	—	0	0	1
	—	0	0	1	—	1	1	1
	—	1	1	1	—	0	0	1
8.	—	1	1	1	—	0	1	1
	—	0	1	1	—	1	1	1
	—	1	1	1	—	1	1	1
9.	—	1	1	1	—	0	0	1
	—	0	0	1	—	1	1	1
	—	1	1	1	—	1	1	1

:Data in unit starting at bit 4
:Data in unit starting at bit 0

In 1/3 duty ratio operation, to be able to define two digits in three bytes, the data stored in two bytes, with the first byte starting at bit 0, and second byte starting at bit 4.

14.4.3 Output Waveforms during LCD Controller/Driver Operation (1/4 Duty Ratio)

In the 1/4 duty ratio mode, all four common outputs, COM0, COM1, COM2, and COM3 are used.

■ Output Waveforms during LCD Controller/Driver Operation (1/4 Duty Ratio)

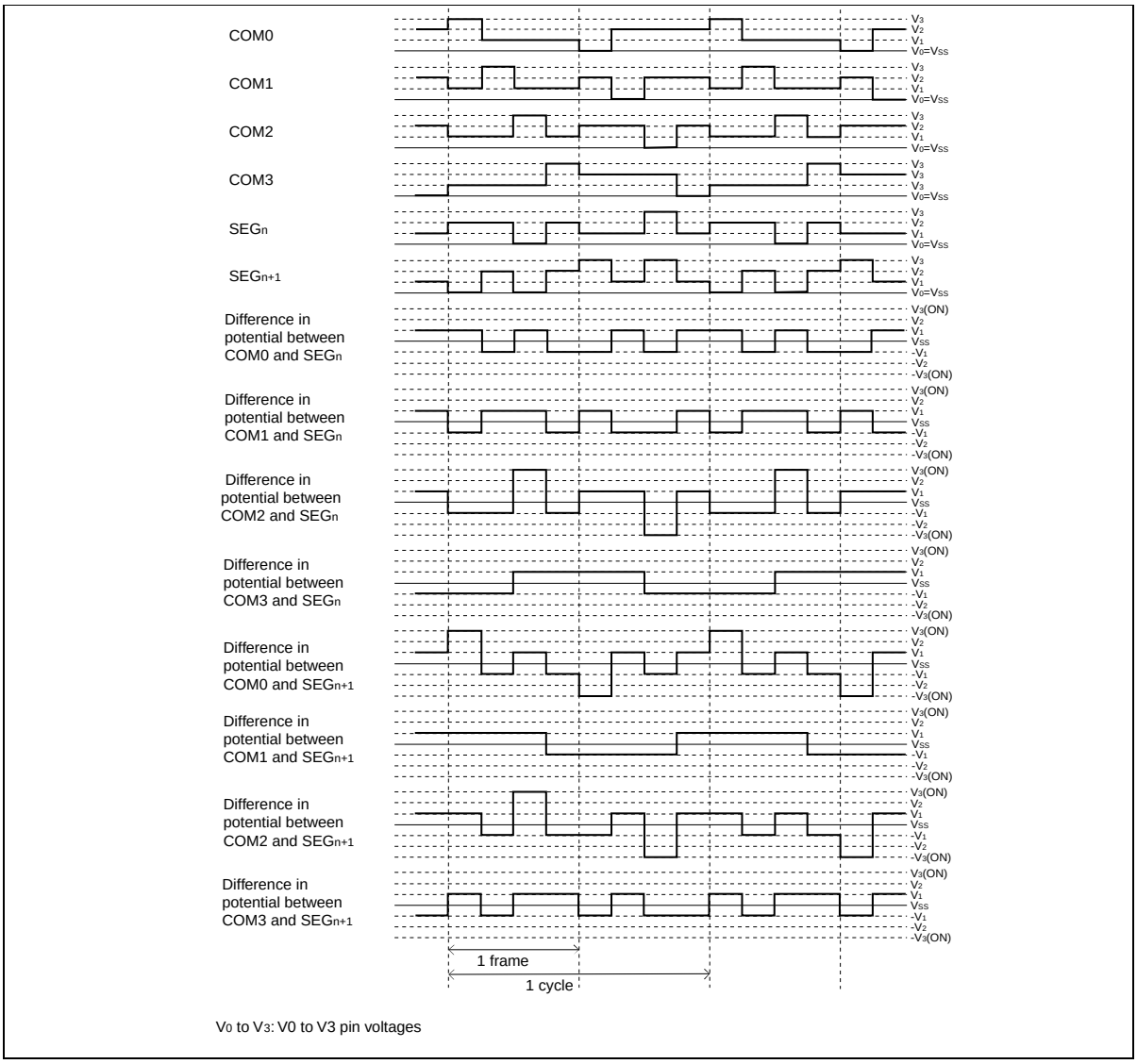
❍ 1/3 bias, 1/4 duty output waveforms

The maximum potential difference exists between a segment output and the corresponding common output when the segment (LCD cell) is "turned on". Figure 14.4-6 "Output Waveforms, 1/3 Bias and 1/4 Duty Ratio Example" shows the output waveforms for the display RAM contents listed in Table 14.4-3 "Display RAM Contents Example".

Table 14.4-3 Display RAM Contents Example

Segment	Display RAM contents			
	COM3	COM2	COM1	COM0
SEGn	0	1	0	0
SEGn+1	0	1	0	1

Figure 14.4-6 Output Waveforms, 1/3 Bias and 1/4 Duty Ratio Example

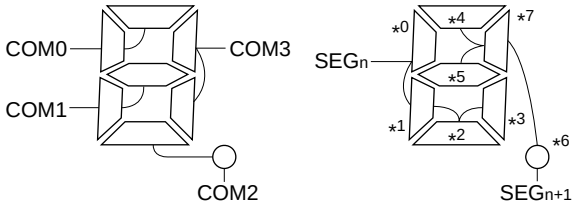


V0 to V3: V0 to V3 pin voltages

❍ 8-segment LCD panel connections and Display data (1/4 duty ratio drive mode)

Figure 14.4-7 Segment/Common Connections, Data States and Corresponding Display

Example) Using segments to represent "5".

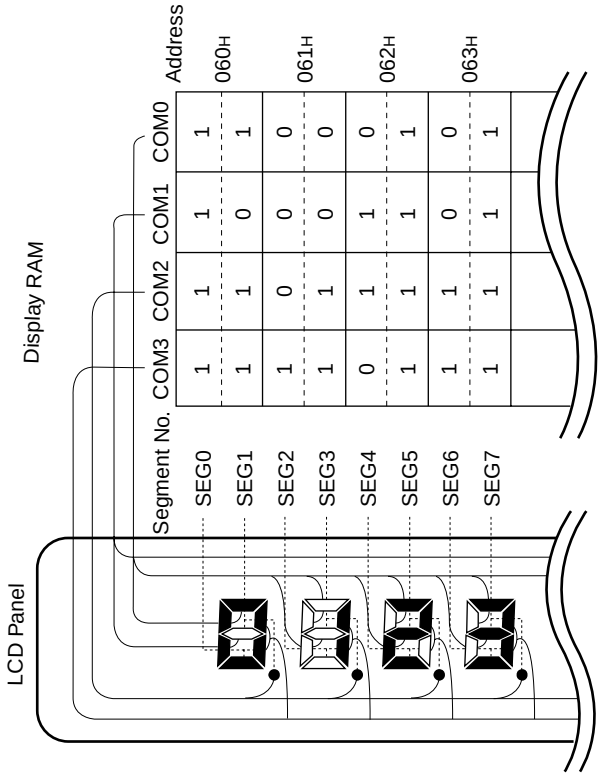


Address n H	COM3	COM2	COM1	COM0	
	bit3*3	bit2*2	bit1*1	bit0*0	SEGn
	bit7*7	bit6*6	bit5*5	bit4*4	SEGn+1

*0 to *7: Indicate corresponding display RAM bits.

Address	COM3	COM2	COM1	COM0	
060H	1	1	0	1	SEG0
	0	0	1	1	SEG1

0: OFF
1: ON



LCD Display	Bit States for Numerals "0" through "9"							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0.	1	1	0	1	1	1	1	1
1.	1	1	0	0	1	0	0	0
2.	1	1	1	1	0	1	1	0
3.	1	1	1	1	1	1	0	0
4.	1	1	1	0	1	0	0	1
5.	0	1	1	1	1	1	0	1
6.	0	1	1	1	1	1	1	1
7.	1	1	0	1	1	0	0	1
8.	1	1	1	1	1	1	1	1
9.	1	1	1	1	1	1	0	1

14.5 Program Example for LCD Controller/Driver

This section gives a program example for LCD controller/driver.

■ Program Example for LCD Controller/Driver Function

❍ Processing description

The process writes LCD display data to display RAM. The data is that required to display the numbers "0" through "9" in an LCD panel connected as shown in Figure 14.4-7 "Segment/ Common Connections, Data States and Corresponding Display". The settings are as follows:

- Internal voltage divider resistors are selected in a device with no voltage booster (LCR1: VSEL = 1)
- 1/3 bias and 1/4 duty ratio are used.
- The subclock (LCR1: CSS = 1) is selected as the clock for frame cycle generation.
- The frame frequency is set at 32 Hz (LCR1: FP1, FP0 = 11_B)
- Operation is stopped in watch mode.

❍ Coding example

```
LCRAM     EQU     0060H           ;Starting address of LCD display RAM
LCR1      EQU     0072H           ; Address of LCDC control register 1 (LCR1)

LCDSEG    CSEG                    ; 8-segment LCD display data
LCDDATA   DB      11011111B       ; "0"
          DB      11001000B       ; "1"
          DB      11110110B       ; "2"
          DB      11111100B       ; "3"
          DB      11101001B       ; "4"
          DB      01111101B       ; "5"
          DB      01111111B       ; "6"
          DB      11011001B       ; "7"
          DB      11111111B       ; "8"
          DB      11111101B       ; "9"
          DB      00000000B       ;END
LCDSEG    ENDS

;---------- Main Program ----------------------------------------------------------------------
```

```
                CSEG                            ; [CODE SEGMENT]
                :
                MOVW    EP,#LCRAM               ; Set LCD display RAM address.
                MOVW    IX,#LCDDATA             ; Set LCD display data table address.
LCDSET          MOV     A,@IX+00H
                MOV     @EP,A
                INCW    EP
                INCW    IX
                BNZ     LCDSET
                MOV     LCR1,#10101111B         ; Continue until data end (00H) is detected.
                :                               ; Set LCR1 and turn LCD display on.
                ENDS

;-----------------------------------------------------------------------------------------------------------

                END
```

CHAPTER 15 BUZZER OUTPUT

This chapter describes the functions and operation of the buzzer output.

15.1 Overview of Buzzer Output

The buzzer output can select from seven different output frequencies (square waves) and can be used for applications such as sounding a buzzer to confirm key input. The function uses the same output pin as the remote control transmit output.

■ Buzzer Output Function

The buzzer output function outputs a signal (square wave) suitable for applications such as sounding a buzzer to confirm an operation.

- For buzzer output, one of seven output frequencies can be selected, or the output disabled.
- Four divide-by-n outputs are supplied from the timebase timer and three from the watch prescaler, for selection as the buzzer output signal.

Note:

Since divided outputs of the timebase timer and timeclock prescaler are fed as the buzzer output signal, the buzzer output will be affected when the signal source selected for it (timebase timer or watch prescaler) is cleared.

Check:

Since the timebase timer stops when the main clock oscillator stops (during subclock mode), do not select the divided output of the timebase timer as the buzzer output when subclock mode is used.

Check:

Similarly, do not select the watch prescaler as the buzzer source in a chip in which the single clock option is selected.

Table 15.1-1 "Output Frequency" lists the seven output frequencies (square waves) that can be selected for the buzzer output function.

Table 15.1-1 Output Frequency

Clock supply source	Buzzer output cycle	Square wave output (Hz)
Timebase Timer	$2^{12}/F_{CH}$	$F_{CH}/2^{12}$ (1.025 kHz)
	$2^{11}/F_{CH}$	$F_{CH}/2^{11}$ (2.051 kHz)
	$2^{10}/F_{CH}$	$F_{CH}/2^{10}$ (4.102 kHz)
	$2^{9}/F_{CH}$	$F_{CH}/2^{9}$ (8.203 kHz)
Watch Prescaler	$2^{5}/F_{CL}$	$F_{CL}/2^{5}$ (1.024 kHz)
	$2^{4}/F_{CL}$	$F_{CL}/2^{4}$ (2.048 kHz)
	$2^{3}/F_{CL}$	$F_{CL}/2^{3}$ (4.096 kHz)

F_{CH}: Main clock oscillation frequency

F_{CL}: Subclock oscillation frequency

The frequencies enclosed in parentheses ( ) are for F_{CH} = 4.2 MHz, and F_{CL} = 32.768 kHz.

Calculation example for output frequency

Note:

For a 4.2 MHz main clock source oscillation and if the buzzer register (BZCR) selects a timebase timer divided output of $F_{CH}/2^{10}$ (Bz2, Bz1, Bz0 = 011B), the output frequency of the BZ pin is calculated as follows:

$$\begin{aligned} \text{Output frequency} &= F_{CH}/2^{10} \\ &= 4.2\ \text{MHz}/1024 \\ &\fallingdotseq 4.102\ \text{kHz} \end{aligned}$$

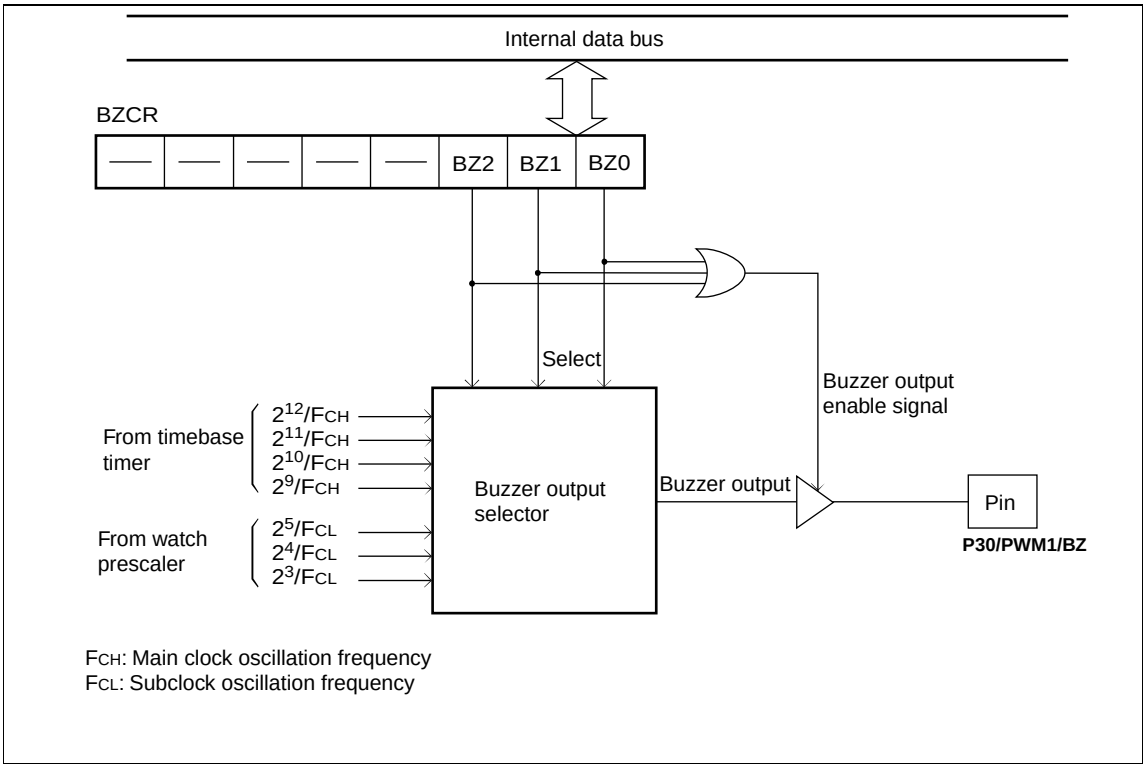
15.2 Block Diagram of Buzzer Output

The buzzer output consists of the following two blocks:

- **Buzzer output selector**
- **Buzzer register (BZCR)**

■ Block Diagram of Buzzer Output

Figure 15.2-1 Block Diagram of Buzzer Output



❍ **Buzzer output selector**

Selects one of the four frequencies output from the timebase timer or three frequencies output from the watch prescaler.

❍ **BZCR register**

The BZCR register to set the buzzer output frequency and enable buzzer output.

Buzzer output is enabled if an output frequency is specified (other than "000_B") in the BZCR register.

15.3 Structure of Buzzer Output

This section describes the pin, pin block diagram, and register of the buzzer output.

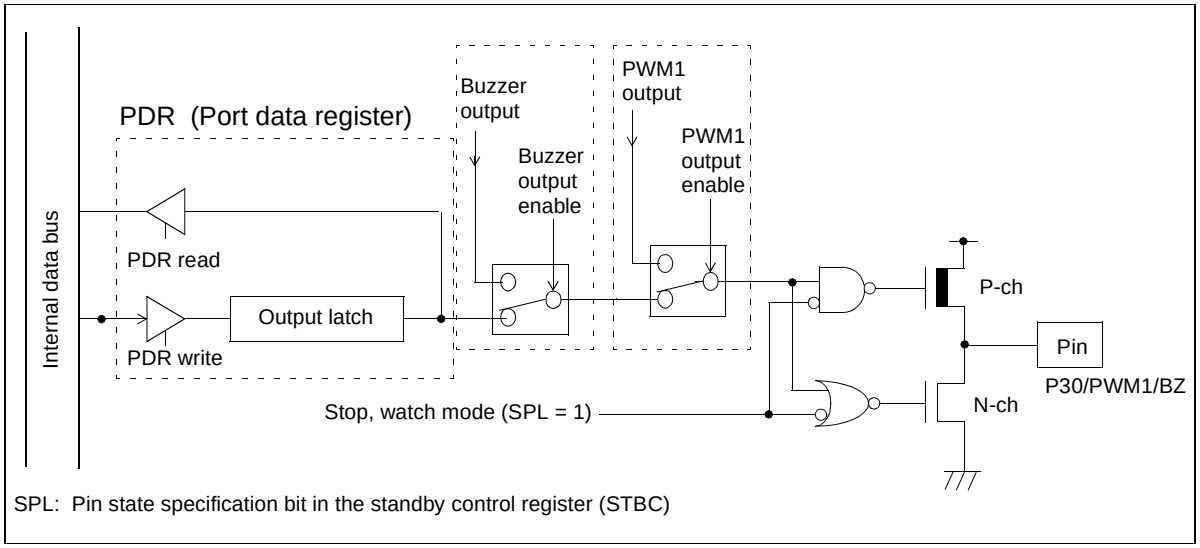
■ Buzzer Output Pin

The buzzer output uses the P30/PWM1/BZ pin. This pin can function as a CMOS output port (P30), or PWM 1 output (PWM1) or the buzzer output pin (BZ).

BZ: This pin outputs a buzzer square wave with the specified frequency.
Setting a buzzer output frequency in the buzzer output register (BZCR: BZ1, BZ0 = other than "00B") automatically sets the P30PWM1/BZ pin as the BZ pin regardless of the output latch value when PWM1 is disbled.

■ Block Diagram of Buzzer Output Pin

Figure 15.3-1 Block Diagram of P30/PWM1/BZ Pin



■ Buzzer Output Register

Figure 15.3-2 Buzzer Output Register

BZCR(Buzzer Register)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0010_H	—	—	—	—	—	BZ2	BZ1	BZ0	$-----000_B$
						R/W	R/W	R/W	

R/W : Readable and writable
— : Unused
X : Indeterminate

15.4 Buzzer Register (BZCR)

The buzzer register (BZCR) is used to select the buzzer output frequency and also enables buzzer output.

■ Buzzer Register (BZCR)

Figure 15.4-1 Buzzer Register (BZCR)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Initial value
0010H	—	—	—	—	—	BZ2	BZ1	BZ0	-----000B
						R/W	R/W	R/W	

BZ2	BZ1	BZ0	Buzzer selection bits		
0	0	0	Disables buzzer output. Makes pin available for use either as a general-purpose port (P30) or PWM1 output (PWM1).		
0	0	1	Functions as a buzzer output pin (BZ).	Timebase timer outputs	FCH/2^12
0	1	0			FCH/2^11
0	1	1			FCH/2^10
1	0	0			FCH/2^9
1	0	1		Watch prescaler outputs	FCL/2^5
1	1	0			FCL/2^4
1	1	1			FCL/2^3

R/W : Readable and writable
— : Unused
X : Indeterminate
Initial value
FCH : Main clock oscillation frequency
FCL : Subclock oscillation frequen

Table 15.4-1 Buzzer Register (BZCR) Bits

Bit		Function
Bit 7 Bit 6 Bit 5 Bit 4	Unused bits	• The read value is indeterminate. • Writing to these bits has no effect on the operation.
Bit 2 Bit 1 Bit 0	BZ2, BZ1, BZ0: Buzzer selection bits	• Setting "000B" disables the buzzer output and sets the pin as a general-purpose port (P30) or PWM 1 output (PWM1). • Setting other than "000B" sets the pin the as the buzzer output (BZ) pin and outputs a square wave of the selected frequency. • Selects one of four divided outputs from the timebase timer or three from the timeclock prescaler. **Note:** Do not select a timebase timer division output in subclock mode. **Note:** The subclock oscillator operates in the main-stop mode. Therefore, if the pin state specification bit (STBC: SPL) is "0", the buzzer output can be used even in main-stop mode by selecting one of the watch prescaler divided-by-n outputs (BZ2, BZ1, BZ0 = 101B to 111B).

15.5 Program Example for Buzzer Output

This section gives a program example for the buzzer output.

■ Program Example for Buzzer Output

❍ Processing description

- Output a buzzer output of approximately 1.025 kHz to the BZ pin, then turn the buzzer output "OFF".
- For a 4.2 MHz main clock source oscillation and selecting $2^{12}/F_C$ (F_C: main clock oscillation), the buzzer output frequency is as follows:

  Buzzer output frequency = 4.2 MHz/2^{12}
  = 4.2 MHz/4096
  = 1.025 kHz

❍ Coding example

```
BZCR        EQU         0010H                   ; Buzzer register
;---------- Main Program ----------------------------------------------------------------------
            CSEG                                ; [CODE SEGMENT]
            :
BUZON       MOV         BZCR,#00000001B         ; Buzzer output "ON".
            :
            :
            :
BUZOFF      MOV         BZCR,#00000000B         ; Buzzer output "OFF".
            :
            ENDS
;-------------------------------------------------------------------------------------------------------------
            END
```

APPENDIX

The appendices are include an I/O map and the instruction list.

A "I/O Map"

B "Instructions"

C "Mask Options"

D "Programming Specifications for One-time PROM and EPROM Microcontroller"

E "MB89980 Series Pin States"

APPENDIX A I/O Map

Table A-1 "I/O Map" lists the addresses of the registers of used by the internal peripheral functions of the MB89980 series.

■ I/O Map

Table A-1 I/O Map

Address	Register name	Register description	Read/Write	Initial value
00_H	PDR0	Port 0 data register	R/W	$XXXXXXXX_B$
01_H	DDR0	Port 0 data direction register	W	00000000_B
02_H	PDR1	Port 1 data register	R/W	$XXXXXXXX_B$
03_H	DDR1	Port 1 data direction register	W	00000000_B
04_H	PDR2	Port 2 data register	R/W	$XXXXXXXX_B$
05_H	DDR2	Port 2 data direction register	W	00000000_B
06_H	(Vacancy)			$XXXXXXXX_B$
07_H	SYCC	System clock control register	R/W	$---MM100_B$
08_H	STBC	Standby control register	R/W	$00010---_B$
09_H	WDTC	Watchdog timer control register	R/W	$0---XXXX_B$
$0A_H$	TBTC	Timebase timer control register	R/W	$00---000_B$
$0B_H$	WPCR	Watch prescaler control register	R/W	$00---000_B$
$0C_H$	PDR3	Port 3 data register	R/W	$XXXXXXX1_B$
$0D_H$	(Vacancy)			$XXXXXXXX_B$
$0E_H$	PDR4	Port 4 data register	R/W	11111111_B
$0F_H$	PDR5	Port 5 data register	R/W	$XXXX1111_B$
10_H	BZCR	Buzzer register	R/W	$-----000_B$
11_H	(Vacancy)			$XXXXXXXX_B$
12_H	PDR6	Port 6 data register	R/W	$XX111111_B$
13_H	PDR7	Port 7 data register	R/W	$XXXXXX11_B$
14_H	RCR1	Remote control transmission register 1	R/W	00000000_B
15_H	RCR2	Remote control transmission register 2	R/W	$0-000000_B$
16_H to 17_H	(Vacancy)			$XXXXXXXX_B$
18_H	T2CR	Timer 2 control register	R/W	$X000XXX0_B$

Table A-1 I/O Map

Address	Register name	Register description	Read/Write	Initial value
19_H	T1CR	Timer 1 control register	R/W	$X000XXX0_B$
$1A_H$	T2DR	Timer 2 data register	R/W	$XXXXXXXX_B$
$1B_H$	T1DR	Timer 1 data register	R/W	$XXXXXXXX_B$
$1C_H$ - $1D_H$	(Vacancy)			$XXXXXXXX_B$
$1E_H$	CNTR1	PWM 1 control register	R/W	$0\text{-}000000_B$
$1F_H$	COMR1	PWM 1 compare register	W	$XXXXXXXX_B$
20_H	CNTR2	PWM 2 control register	R/W	$0\text{-}000000_B$
21_H	COMR2	PWM 2 compare register	W	$XXXXXXXX_B$
22_H to $2C_H$	(Vacancy)			$XXXXXXXX_B$
$2D_H$	ADC1	A/D control register 1	R/W	00000000_B
$2E_H$	ADC2	A/D control register 2	R/W	$---00001_B$
$2F_H$	ADCD	A/D data register	R/W	$XXXXXXXX_B$
30_H	EIE1	External interrupt 1 control register	R/W	00000000_B
31_H	EIF1	External interrupt 1 flag register	R/W	$----0000_B$
32_H	EIE2	External interrupt 2 control register	R/W	00000000_B
33_H	EIF2	External interrupt 2 flag register	R/W	$-------0_B$
34_H to $3F_H$	(Vacancy)			$XXXXXXXX_B$
40_H	PURR0	Pull-up control register 0 (For MB89P985/PV980 only)	R/W	11111111_B
41_H	PURR1	Pull-up control register 1 (For MB89P985/PV980 only)	R/W	11111111_B
42_H	PURR5	Pull-up control register 5 (For MB89P985/PV980 only)	R/W	$----1111_B$
43_H to $5F_H$	(Vacancy)			$XXXXXXXX_B$
60_H to 66_H	VRAM	Display RAM	R/W	$XXXXXXXX_B$
67_H to 71_H	(Vacancy)			$XXXXXXXX_B$
72_H	LCR1	LCD control register 1	R/W	00010000_B
73_H	LCR2	LCD control register 2 (For MB89P985/PV980 only)	R/W	$0\text{--}0000\text{-}_B$
74_H to $7B_H$	(Vacancy)			$XXXXXXXX_B$
$7C_H$	ILR1	Interrupt level setting register 1	W	11111111_B
$7D_H$	ILR2	Interrupt level setting register 2	W	11111111_B
$7E_H$	ILR3	Interrupt level setting register 3	W	11111111_B

Table A-1 I/O Map

Address	Register name	Register description	Read/Write	Initial value
$7F_H$	ITR	Interrupt test register	Access prohibited	$XXXXXX00_B$

❍ **Read/write access symbols**

R/W: Readable and writable

R: Read-only

W: Write-only

❍ **Initial value symbols**

0: The initial value of this bit is "0".

1: The initial value of this bit is "1".

X: The initial value of this bit is undefined.

M: The initial value of this bit is determined by mask option.

Check:

Do not use vacancies.

APPENDIX B Instructions

This appendix describes the F^{2}MC-8L instruction set.

B.1 Instruction List Symbols

Table B.1-1 "Instruction List Symbols"lists the meaning of the symbols and Table B.1-2 "Instruction List Columns"lists the meaning of the columns used in section B.4 "F^{2}MC-8L Instruction List".

■ Instruction List Symbols

Table B.1-1 Instruction List Symbols

Symbol	Meaning
dir	Direct address (8 bits)
off	Offset (8 bits)
ext	Extended address (16 bits)
#vct	Vector table number (3 bits)
#d8	Immediate data (8 bits)
#d16	Immediate data (16 bits)
dir: b	Bit direct address (8:3 bits)
rel	Branch relative address (8 bits)
@	Register indirect (Example: @A, @IX, @EP)
A	Accumulator A (Whether its length is 8 or 16 bits is determined by the instruction in use.)
AH	Upper 8 bits of the accumulator A (8 bits)
AL	Lower 8 bits of the accumulator A (8 bits)
T	Temporary accumulator T (Whether its length is 8 or 16 bits is determined by the instruction in use.)
TH	Upper 8 bits of the temporary accumulator T (8 bits)
TL	Lower 8 bits of the temporary accumulator T (8 bits)
IX	Index register IX (16 bits)
EP	Extra pointer EP (16 bits)
PC	Program counter PC (16 bits)
SP	Stack pointer SP (16 bits)
PS	Program status PS (16 bits)
dr	Accumulator A or index register IX (16 bits)
CCR	Condition code register CCR (8 bits)

Table B.1-1 Instruction List Symbols

Symbol	Meaning
RP	Register bank pointer RP (5 bits)
Ri	General-purpose register Ri (8 bits, i = 0 to 7)
¥	Indicates that the very × is the immediate data. (Whether its length is 8 or 16 bits is determined by the instruction in use.)
(×)	Indicates that the contents of × is the target of accessing. (Whether its length is 8 or 16 bits is determined by the instruction in use.)
((×))	The address indicated by the contents of × is the target of accessing. (Whether its length is 8 or 16 bits is determined by the instruction in use.)

Table B.1-2 Instruction List Columns

Column	Description
Mnemonic	Assembler notation of an instruction
~	Number of instructions
#	Number of bytes
Operation	Operation of an instruction
TL, TH, AH	A content change when each of the TL, TH, and AH instructions is executed. Symbols in the column indicate the following: • "-" indicate no change. • dH is the 8 upper bits of operation description data. • AL and AH must become the contents of AL and AH immediately before the instruction is executed. • 00 becomes 00.
N, Z, V, C	An instruction of which the corresponding flag will change. If + is written in this column, the relevant instruction will change its corresponding flag.
OP Code	Code of an instruction. If an instruction is more than one code, it is written according to the following rule: Example: "48 to 4F" <-- This indicates 48, 49 ... 4F.

B.2 Addressing

The F^{2}MC-8L supports the following ten addressing modes:

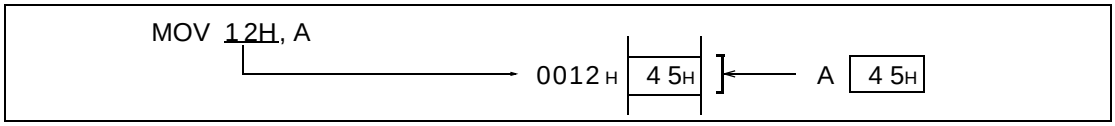
- **Direct addressing**
- **Extended addressing**
- **Bit direct addressing**
- **Index addressing**
- **Pointer addressing**
- **General-purpose register addressing**
- **Immediate addressing**
- **Vector addressing**
- **Relative addressing**
- **Inherent addressing**

■ Addressing Modes

❍ Direct addressing

Indicated by "dir" in the instruction list. Used to access the area between "0000H" and "00FFH". For direct addressing, the upper one byte of the address is "00H" and the operand specifies the lower one byte. Figure B.2-1 "Direct Addressing" shows an example.

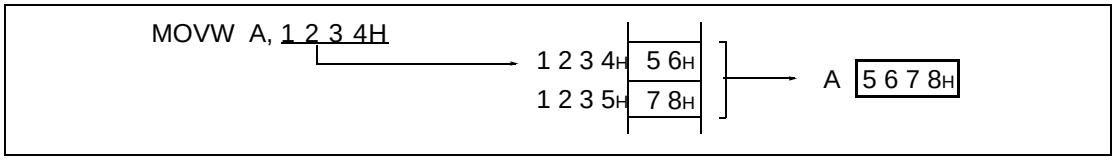
Figure B.2-1 Direct Addressing



❍ Extended addressing

Indicated by "ext" in the instruction list. Used to access the entire 64-Kbyte area. For extended addressing, the first operand specifies the upper one byte of the address and the second operand specifies the lower one byte. Figure B.2-2 "Extended Addressing" shows an example.

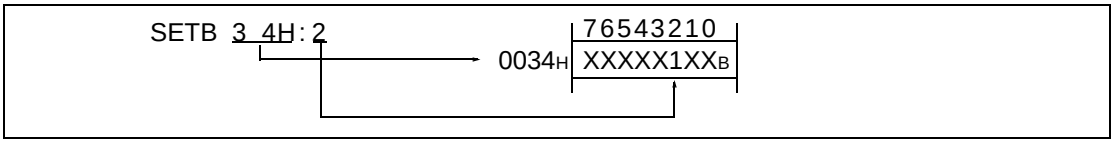
Figure B.2-2 Extended Addressing



❍ **Bit direct addressing**

Indicated by "dir: b" in the instruction list. Used to access the area between "0000H" and "00FFH" in bit units. For bit direct addressing, the upper one byte of the address is "00H" the operand specifies the lower one byte of the address, and the lower three bits of the operation code specify the bit position. Figure B.2-3 "Bit Direct Addressing" shows an example.

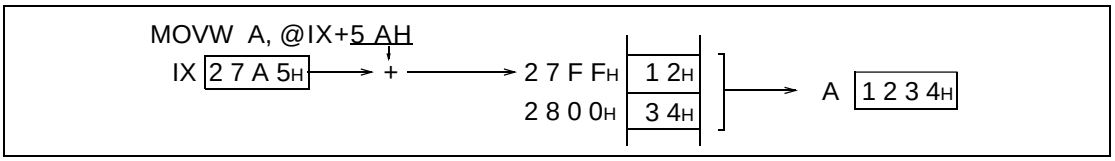
Figure B.2-3 Bit Direct Addressing



❍ **Index addressing**

Indicated by "@IX+off" in the instruction list. Used to access the entire 64-Kbyte area. Index addressing generates the address is obtained by adding the sign-extended contents of the first operand to the index register (IX). Figure B.2-4 "Index Addressing" shows an example.

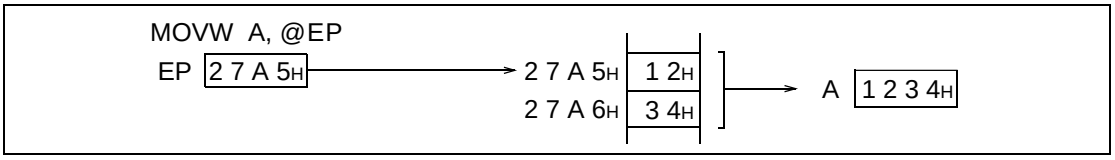
Figure B.2-4 Index Addressing



❍ **Pointer addressing**

Indicated by "@EP" in the instruction list. Used to access the entire 64-Kbyte area. Pointer addressing uses the extra pointer (EP) the address. Figure B.2-5 "Pointer Addressing" shows an example.

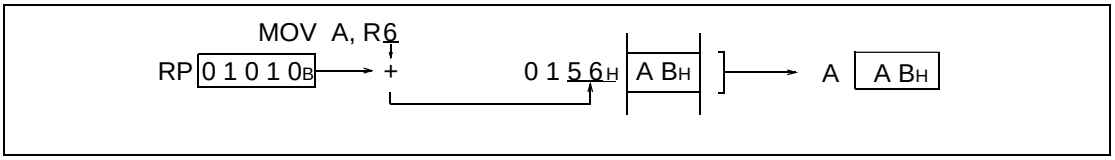
Figure B.2-5 Pointer Addressing



❍ **General-purpose register addressing**

Indicated by "Ri" in the instruction list. Used to access the general-purpose register area register bank. For general-purpose register addressing, the upper one byte of the address is fixed at "01" and the lower byte is generated from the register bank pointer (RP) and the lower three bits of the operation code. The CPU accesses the resulting address. Figure B.2-6 "General-purpose Register Addressing" shows an example.

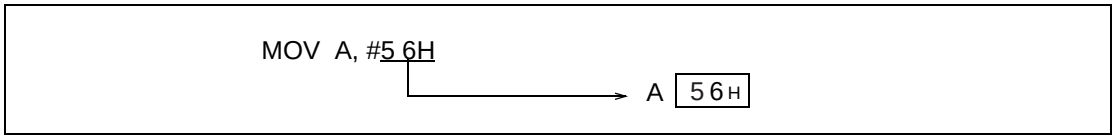
Figure B.2-6 General-purpose Register Addressing



❍ Immediate addressing

Indicated by "#d8" in the instruction list. Used when immediate data is required. In immediate addressing, the operand is used directly as immediate data. The operation code determines whether the data is byte or word. Figure B.2-7 "Immediate Addressing" shows an example.

Figure B.2-7 Immediate Addressing



❍ Vector addressing

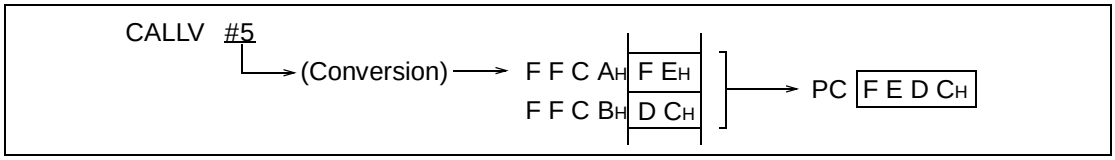
Indicated by "vct" in the instruction list. Used to branch to a subroutine address stored in the vector table. For vector addressing, the "vct" data is contained in the operation code. Table B.2-1 "Vector Table Address Corresponding to "vct""lists the correspondence between "vct" and the resulting address.

Table B.2-1 Vector Table Address Corresponding to "vct"

#vct	Vector table address (branch destination upper address: lower address)
0	$FFC0_H$: $FFC1_H$
1	$FFC2_H$: $FFC3_H$
2	$FFC4_H$: $FFC5_H$
3	$FFC6_H$: $FFC7_H$
4	$FFC8_H$: $FFC9_H$
5	$FFCA_H$: $FFCB_H$
6	$FFCC_H$: $FFCD_H$
7	$FFCE_H$: $FFCF_H$

Figure B.2-8 "Vector Addressing" shows an example.

Figure B.2-8 Vector Addressing



❍ Relative addressing

Indicated by "rel" in the instruction list. Used to branch to a destination in the area 128 bytes above or below the program counter (PC). Relative addressing adds the sign-extended contents of the first operand to the PC and stores the result in the PC. Figure B.2-9 "Relative Addressing" shows an example.

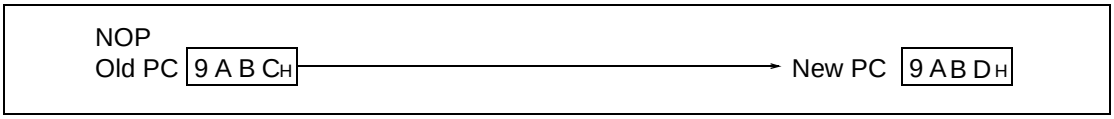
Figure B.2-9 Relative Addressing

This example branches to the address containing the BNE operation code and therefore results in an endless loop.

❍ **Inherent addressing**

Inherent addressing is used for instructions in the instruction list that do not have operands and for which the operation code determines the operation. The operation of inherent addressing depends on the instruction. Figure B.2-10 "Inherent Addressing" shows an example.

Figure B.2-10 Inherent Addressing



B.3 Special Instructions

This section describes special instructions, other than addressing.

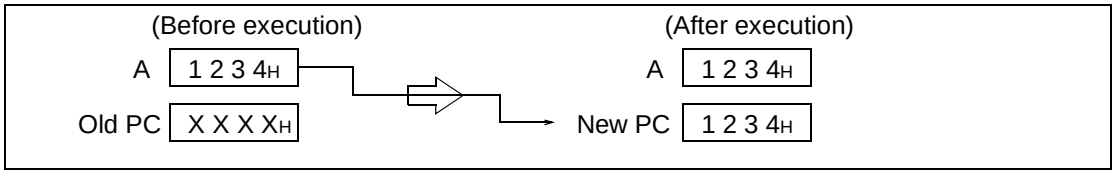
■ Special Instructions

❍ JMP @A

This instruction moves the address contained in the accumulator (A) to the program counter (PC) and branches to the new address. This instruction can be used to perform an N option branch by placing N branch destination addresses in a table and moving the desired address to the accumulator.

Figure B.3-1 "JMP @A" shows an outline of the instruction operation.

Figure B.3-1 JMP @A

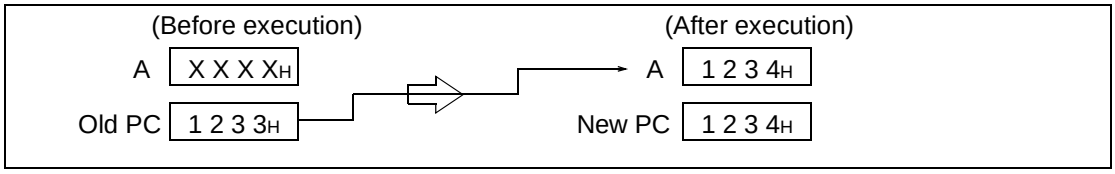


❍ MOVW A,PC

This instruction stores the PC contents in the accumulator A. This performs the opposite operation to "JMP @A". By executing this instruction in the main routine and calling a particular subroutine, the subroutine can determine whether the contents of A match a predetermined value. The subroutine can check whether program runaway has occurred by checking whether or not execution has branched from an expected location.

Figure B.3-2 "MOVW A,PC" shows an outline of the instruction operation.

Figure B.3-2 MOVW A,PC



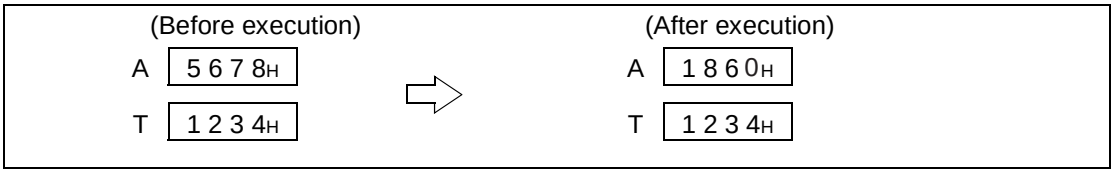
The content of A after executing this instruction is the address of the next instruction (not the address containing the operation code of this instruction. Accordingly, the value "1234_H" stored in A in the example shown in Figure B.3-2 "MOVW A,PC" is the address of the next operation code after "MOVW A, PC".

❍ **MULU A**

This instruction performs an unsigned multiplication of AL (lower 8 bits of the accumulator) and TL (lower 8 bits of the temporary accumulator) and stores the 16-bit result in A. The contents of T (temporary accumulator) does not change. The arithmetic operation does not use the pre-execution contents of AH (upper 8 bits of the accumulator) and TH (upper 8 bits of the temporary accumulator). Since the flags remain unchanged, use care when branching is required based on the result of multiplication.

Figure B.3-3 "MULU A" shows an outline of the instruction operation.

Figure B.3-3 MULU A

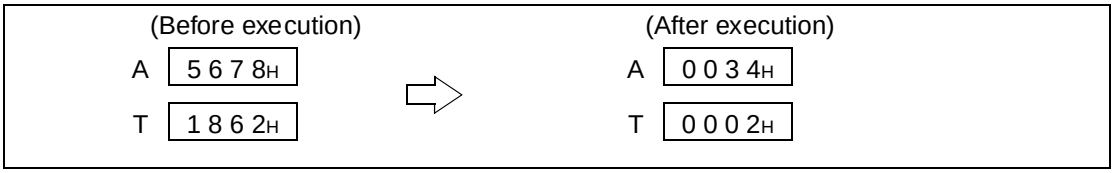


❍ **DIVU A**

This instruction divides the 16 bits of T by the 8 bits of AL, treating the data as unsigned. The instruction stores the result in AL and the remainder in TL, both as 8 bit data. AH and TH are both set to "zero". The arithmetic operation does not use the value of AH prior to instruction execution. The result is not assured for data that produces a result that exceeds 8 bits. As there is no indication that the result exceeded 8 bits, check the data before performing. Since the flags remain unchanged, use care when branching is required based on the result of the division.

Figure B.3-4 "DIVU A" shows an outline of the instruction operation.

Figure B.3-4 DIVU A

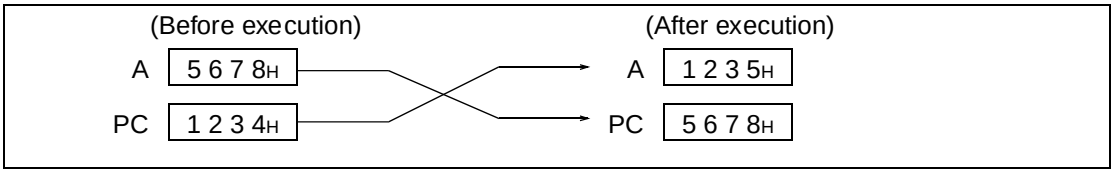


❍ **XCHW A,PC**

This instruction exchanges the contents of A and PC, and as a result branches to the address corresponding to contents of A before execution. The contents of A after execution assume the address next to the address where the operation code of the "XCHW A,PC" is stored. The instruction can be used to specify a table in the main routine which is used in a subroutine.

Figure B.3-5 "XCHW A,PC" shows an outline of the instruction operation.

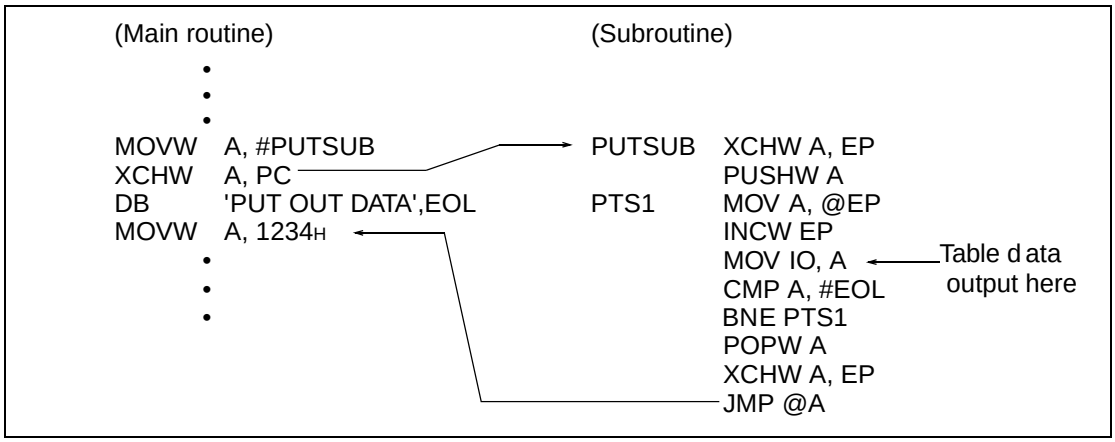
Figure B.3-5 XCHW A,PC



The content of A after executing this instruction is the address of the next instruction (not the address containing the operation code of this instruction). Accordingly, the value "1235_H" stored in A in the example shown in Figure B.3-5 "XCHW A,PC" is the address of the next operation code after "XCHW A,PC". Therefore, the value of A is "1235_H" not "1234_H".

Figure B.3-6 "Example Using XCHW A, PC" shows an assembly language example.

Figure B.3-6 Example Using XCHW A, PC

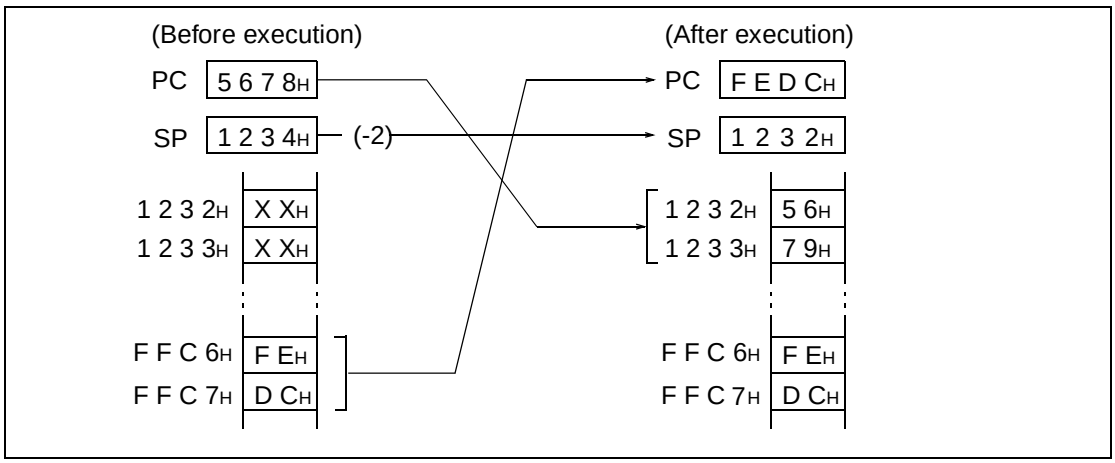


❍ **CALLV #vct**

This instruction is used to branch to a subroutine address in the vector table. The instruction saves the return address (contents of the PC) to the address corresponding to the SP (stack pointer) branches to the address stored in the vector table using vector addressing. As "CALLV #vct" is a single-byte instruction, using this instruction for commonly used subroutines reduces the overall program size.

Figure B.3-7 "Execution example of CALLV #3" shows an outline of the instruction operation.

Figure B.3-7 Execution example of CALLV #3



The content of PC saved to stack area after executing this instruction is the address of next instruction (not the address containing the operation code of this instruction). Accordingly, the value "5679_H" saved to stack (1232_H, 1233_H) in the example shown in Figure B.3-7 "Execution example of CALLV #3" is the address (return address) of the next operation code after "MOVW A,PC".

B.4 F²MC-8L Instructions

Table B.4-1 "Transfer Instructions"to B.4-4 "Other Instructions"list the F²MC-8L instructions.

■ Transfer Instructions

Table B.4-1 Transfer Instructions

No.	Mnemonic	~	#	Operation	TL	TH	AH	NZVC	OP code
1	MOV dir, A	3	2	(dir) <-- (A)	-	-	-	----	45
2	MOV @IX+off, A	4	2	((IX)+off) <-- (A)	-	-	-	----	46
3	MOV ext, A	4	3	(ext) <-- (A)	-	-	-	----	61
4	MOV @EP, A	3	1	((EP)) <-- (A)	-	-	-	----	47
5	MOV Ri, A	3	1	(Ri) <-- (A)	-	-	-	----	48 to 4F
6	MOV A, #d8	2	2	(A) <-- d8	AL	-	-	++--	04
7	MOV A, dir	3	2	(A) <-- (dir)	AL	-	-	++--	05
8	MOV A, @IX+off	4	2	(A) <-- ((IX)+off)	AL	-	-	++--	06
9	MOV A, ext	4	3	(A) <-- (ext)	AL	-	-	++--	60
10	MOV A, @A	3	1	(A) <-- ((A))	AL	-	-	++--	92
11	MOV A, @EP	3	1	(A) <-- ((EP))	AL	-	-	++--	07
12	MOV A, Ri	3	1	(A) <-- (Ri)	AL	-	-	++--	08 to 0F
13	MOV dir, #d8	4	3	(dir) <-- d8	-	-	-	----	85
14	MOV @IX+off, #d8	5	3	((IX)+off) <-- d8	-	-	-	----	86
15	MOV @EP, #d8	4	2	((EP)) <-- d8	-	-	-	----	87
16	MOV Ri, #d8	4	2	(Ri) <-- d8	-	-	-	----	88 to 8F
17	MOVW dir, A	4	2	(dir) <-- (AH), (dir+1) <-- (AL)	-	-	-	----	D5
18	MOVW @IX+off, A	5	2	((IX)+off) <-- (AH), ((IX)+off+1) <-- (AL)	-	-	-	----	D6
19	MOVW ext, A	5	3	(ext) <-- (AH), (ext+1) <-- (AL)	-	-	-	----	D4
20	MOVW @EP, A	4	1	((EP)) <-- (AH), ((EP)+1) <-- (AL)	-	-	-	----	D7
21	MOVW EP, A	2	1	(EP) <-- (A)	-	-	-	----	E3
22	MOVW A, #d16	3	3	(A) <-- d16	AL	AH	dH	++--	E4
23	MOVW A, dir	4	2	(AH) <-- (dir), (AL) <-- (dir+1)	AL	AH	dH	++--	C5
24	MOVW A, @IX+off	5	2	(AH) <-- ((IX)+off), (AL) <-- ((IX)+off+1)	AL	AH	dH	++--	C6
25	MOVW A, ext	5	3	(AH) <-- (ext), (AL) <-- (ext+1)	AL	AH	dH	++--	C4
26	MOVW A, @A	4	1	(AH) <-- ((A)), (AL) <-- ((A)+1)	AL	AH	dH	++--	93
27	MOVW A, @EP	4	1	(AH) <-- ((EP)), (AL) <-- ((EP)+1)	AL	AH	dH	++--	C7
28	MOVW A, EP	2	1	(A) <-- (EP)	-	-	dH	----	F3
29	MOVW EP, #d16	3	3	(EP) <-- d16	-	-	-	----	E7
30	MOVW IX, A	2	1	(IX) <-- (A)	-	-	-	----	E2

Table B.4-1 Transfer Instructions

No.	Mnemonic	~	#	Operation	TL	TH	AH	NZVC	OP code
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

Check:

The automatic transfer to the T register is TL <-- AL for instructions that perform a byte transfer to A.

Operands in more than one operand instruction must be stored in the order in which their mnemonics are written.

■ Arithmetic Operation Instructions

Table B.4-2 Arithmetic Opeation Instructions

No.	Mnemonic	~	#	Operation	TL	TH	AH	NZVC	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

Table B.4-2 Arithmetic Opeation Instructions

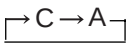
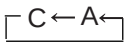
No.	Mnemonic	~	#	Operation	TL	TH	AH	NZVC	OP code
16	INCW EP	3	1	(EP) <-- (EP)+1	-	-	-	----	C3
17	INCW IX	3	1	(IX) <-- (IX)+1	-	-	-	----	C2
18	INCW A	3	1	(A) <-- (A)+1	-	-	dH	++--	C0
19	DEC Ri	4	1	(Ri) <-- (Ri)-1	-	-	-	+++-	D8 to DF
20	DECW EP	3	1	(EP) <-- (EP)-1	-	-	-	----	D3
21	DECW IX	3	1	(IX) <-- (IX)-1	-	-	-	----	D2
22	DECW A	3	1	(A) <-- (A)-1	-	-	dH	++--	D0
23	MULU A	19	1	(A) <-- (AL)*(TL)	-	-	dH	----	01
24	DIVU A	21	1	(A) <-- (T)/(AL), MOD --> (T)	dL	00	00	----	11
25	ANDW A	3	1	(A) <-- (A) ^ (T)	-	-	dH	++R-	63
26	ORW A	3	1	(A) <-- (A) v (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
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) v (TL)	-	-	-	++R-	72
52	OR A, #d8	2	2	(A) <-- (AL) v d8	-	-	-	++R-	74
53	OR A, dir	3	2	(A) <-- (AL) v (dir)	-	-	-	++R-	75
54	OR A, @EP	3	1	(A) <-- (AL) v ((EP))	-	-	-	++R-	77
55	OR A, @IX+off	4	2	(A) <-- (AL) v ((IX)+off)	-	-	-	++R-	76

Table B.4-2 Arithmetic Opeation Instructions

No.	Mnemonic	~	#	Operation	TL	TH	AH	NZVC	OP code
56	OR A, Ri	3	1	(A) <-- (AL) v (Ri)	-	-	-	++R-	78 to 7F
57	CMP dir, #d8	5	3	(dir)-d8	-	-	-	++++	95
58	CMP @EP, #d8	4	2	((EP))-d8	-	-	-	++++	97
59	CMP @IX+off, #d8	5	3	((IX)+off)-d8	-	-	-	++++	96
60	CMP Ri, #d8	4	2	(Ri)-d8	-	-	-	++++	98 to 9F
61	INCW SP	3	1	(SP) <-- (SP)+1	-	-	-	----	C1
62	DECW SP	3	1	(SP) <-- (SP)-1	-	-	-	----	D1

■ Branch Instructions

Table B.4-3 Branch Instructions

No.	Mnemonic	~	#	Operation	TL	TH	AH	NZVC	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 form interrupt	-	-	-	restore	30

■ Other Instructions

Table B.4-4 Other Instructions

No.	Mnemonic	~	#	Operation	TL	TH	AH	NZVC	OP code
1	PUSHW A	4	1		-	-	-	----	40
2	POPW A	4	1		-	-	dH	----	50
3	PUSHW IX	4	1		-	-	-	----	41
4	POPW IX	4	1		-	-	-	----	51
5	NOP	1	1		-	-	-	----	00
6	CLRC	1	1		-	-	-	---R	81
7	SETC	1	1		-	-	-	---S	91
8	CLRI	1	1		-	-	-	----	80
9	SETI	1	1		-	-	-	----	90

B.5 Instruction Map

Table B.5-1 "F^{2}MC-8L Instruction Map" lists the F^{2}MC-8L instruction map.

■ Instruction Map

Table B.5-1 F^{2}MC-8L Instruction Map

L \ H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SWAP	RET	RETI	PUSHW A	POPW A	MOV A,ext	MOVW A,PS	CLRI	SETI	CLRB dir:0	BBC dir:0,rel	INCW A	DECW A	JMP @A	MOVW A,PC
1	MULU A	DIVU A	JMP addr16	CALL addr16	PUSHW IX	POPW IX	MOV ext,A	MOVW PS,A	CLRC	SETC	CLRB dir:1	BBC dir:1,rel	INCW SP	DECW SP	MOVW SP,A	MOVW A,SP
2	ROLC A	CMP A	ADDC A	SUBC A	XCH A,T	XOR A	AND A	OR A	MOV @A,T	MOV A,@A	CLRB dir:2	BBC dir:2,rel	INCW IX	DECW IX	MOVW IX,A	MOVW A,IX
3	RORC A	CMPW A	ADDCW A	SUBCW A	XCHW A,T	XORW A	ANDW A	ORW A	MOVW @A,T	MOVW A,@A	CLRB dir:3	BBC dir:3,rel	INCW EP	DECW EP	MOVW EP,A	MOVW A,EP
4	MOV A,#d8	CMP A,#d8	ADDC A,#d8	SUBC A,#d8		XOR A,#d8	AND A,#d8	OR A,#d8	DAA	DAS	CLRB dir:4	BBC dir:4,rel	MOVW A,ext	MOVW ext,A	MOVW A,#d16	XCHW A,PC
5	MOV A,dir	CMP A,dir	ADDC A,dir	SUBC A,dir	MOV dir,A	XOR A,dir	AND A,dir	OR A,dir	MOV dir,#d8	CMP dir,#d8	CLRB dir:5	BBC dir:5,rel	MOVW A,dir	MOVW dir,A	MOVW SP,#d16	XCHW A,SP
6	MOV A,@IX+d	CMP A,@IX+d	ADDC A,@IX+d	SUBC A,@IX+d	MOV @IX+d,A	XOR A,@IX+d	AND A,@IX+d	OR A,@IX+d	MOV @IX+d,#d8	CMP @IX+d,#d8	CLRB dir:6	BBC dir:6,rel	MOVW A,@IX+d	MOVW @IX+d,A	MOVW IX,#d16	XCHW A,IX
7	MOV A,@EP	CMP A,@EP	ADDC A,@EP	SUBC A,@EP	MOV @EP, A	XOR A,@EP	AND A,@EP	OR A,@EP	MOV @EP, #d8	CMP @EP, #d8	CLRB dir:7	BBC dir:7,rel	MOVW A,@EP	MOVW @EP,A	MOVW EP,#d16	XCHW A,EP
8	MOV A,R0	CMP A,R0	ADDC A,R0	SUBC A,R0	MOV R0,A	XOR A,R0	AND A,R0	OR A,R0	MOV R0,#d8	CMP R0,#d8	SETB dir:0	BBS dir:0,rel	INC R0	DEC R0	CALLV #0	BNC rel
9	MOV A,R1	CMP A,R1	ADDC A,R1	SUBC A,R1	MOV R1,A	XOR A,R1	AND A,R1	OR A,R1	MOV R1,#d8	CMP R1,#d8	SETB dir:1	BBS dir:1,rel	INC R1	DEC R1	CALLV #1	BC rel
A	MOV A,R2	CMP A,R2	ADDC A,R2	SUBC A,R2	MOV R2,A	XOR A,R2	AND A,R2	OR A,R2	MOV R2,#d8	CMP R2,#d8	SETB dir:2	BBS dir:2,rel	INC R2	DEC R2	CALLV #2	BP rel
B	MOV A,R3	CMP A,R3	ADDC A,R3	SUBC A,R3	MOV R3,A	XOR A,R3	AND A,R3	OR A,R3	MOV R3,#d8	CMP R3,#d8	SETB dir:3	BBS dir:3,rel	INC R3	DEC R3	CALLV #3	BN rel
C	MOV A,R4	CMP A,R4	ADDC A,R4	SUBC A,R4	MOV R4,A	XOR A,R4	AND A,R4	OR A,R4	MOV R4,#d8	CMP R4,#d8	SETB dir:4	BBS dir:4,rel	INC R4	DEC R4	CALLV #4	BNZ rel
D	MOV A,R5	CMP A,R5	ADDC A,R5	SUBC A,R5	MOV R5,A	XOR A,R5	AND A,R5	OR A,R5	MOV R5,#d8	CMP R5,#d8	SETB dir:5	BBS dir:5,rel	INC R5	DEC R5	CALLV #5	BZ rel
E	MOV A,R6	CMP A,R6	ADDC A,R6	SUBC A,R6	MOV R6,A	XOR A,R6	AND A,R6	OR A,R6	MOV R6,#d8	CMP R6,#d8	SETB dir:6	BBS dir:6,rel	INC R6	DEC R6	CALLV #6	BGE rel
F	MOV A,R7	CMP A,R7	ADDC A,R7	SUBC A,R7	MOV R7,A	XOR A,R7	AND A,R7	OR A,R7	MOV R7,#d8	CMP R7,#d8	SETB dir:7	BBS I dir:7,rel	INC R7	DEC R7	CALLV #7	BLT rel

B.6 Bit Manipulation Instructions (SETB, CLRB)

The bit manipulation instructions use a different read operation to the normal operation for some bits of peripheral function registers.

■ Read-modify-write Operation

Bit manipulation instructions set to "1" (SETB) or clear to "0" (CLRB) the specified bit only of a register or RAM location. However, as the CPU handles data in 8-bit units, the actual operation consists of reading the 8-bit data, modifying the specified bit, then writing the result back to the same address. This is called a read-modify-write operation.

Table B.6-1 "Bus Operation for Bit Manipulation Instructions" shows the bus operation for bit manipulation instructions.

Table B.6-1 Bus Operation for Bit Manipulation Instructions

Code	Mnemonic	~	Cycle	Address bus	Data bus	$\overline{RD}$	$\overline{WR}$	RMW
A0 to A7 A8 to AF	CLRB dir:b SETB dir:b	4	1 2 3 4	N+1 dir address dir address N+2	dir Data Data Next instruction	0 0 1 0	1 1 0 1	0 1 0 0

■ Read Source When Executing Bit Manipulation Instructions

The read source for a read-modify-write of some I/O ports and interrupt request flag bits is different than for a standard read.

❍ I/O ports (Bit manipulation instructions)

For some I/O ports, a standard read reads the I/O pin values whereas a bit manipulation instruction reads, the output latch value. This is to prevent unintentionally modifying other output latch bit values and is independent of the pin input/output direction or pin state.

❍ Interrupt request flag bits (Bit manipulation instructions)

For interrupt request flag bits, a standard read reads the flag bit to determine whether an interrupt has occurred. Bit manipulation instructions, however, always read interrupt request flag bits as "1". This is to prevent unintentionally clearing the flag by writing "0" to the interrupt request flag bit when performing bit manipulation of a different bit.

APPENDIX C Mask Options

This appendix lists the mask options for the MB89980 series.

■ Mask Options

Table C-1 Mask Options

No.	Part number	MB89983	MB89P985	MB89PV980
	Specifying procedure	Specify when ordering masking	Setting with software	Setting with software
1	Pull-up resistors P00 to P07, P10 to P17, P20 to P27, P40 to P47, P50 to P53, P60 to P65	Selectable per pin. Pull-ups for pins P40 to P47 and P60 to P65 can only be specified, however, when the LCD segment output option is not selected. Also, P50 to P53 must be set to without a pull-up resistor when an A/D converter is used.	Selectable per pin by pull-up control registers. Pull-up resistors are not available for P20 to P27, P40 to P47 and P60 to P65. Furthermore, P50 to P53 must be set to without a pull-up resistor when an A/D converter is used.	Selectable per pin by pull-up control registers. Pull-up resistors are not available for P20 to P27, P40 to P47 and P60 to P65. Furthermore, P50 to P53 must be set to without a pull-up resistor when an A/D converter is used.
2	Power-on reset • With power-on reset • Without power-on reset	Selectable	Fixed with power-on reset	Fixed with power-on reset
3	Main clock oscillation stabilization delay time initial value* selection (F_{CH} = 4.2 MHz) • 00: $2^4/F_{CH}$ (Approx. 0 ms) • 01: $2^{12}/F_{CH}$ (Approx. 1.0 ms) • 10: $2^{16}/F_{CH}$ (Approx. 15.6 ms) • 11: $2^{18}/F_{CH}$ (Approx. 62.4 ms)	Selectable	Fixed to oscillation stabilization time of $2^{18}/F_{CH}$(Approx. 62.4 ms).	Fixed to oscillation stabilization time of $2^{18}/F_{CH}$ (Approx. 62.4 ms).
4	Main clock frequency determining device: • Crystal/ceramic resonator • CR	Selectable	Crystal or ceramic resonator only	Crystal or ceramic resonator only
5	Reset pin output • With reset output • Without reset output	Selectable	Fixed with reset output	Fixed with reset output

Table C-1 Mask Options

No.	Part number	MB89983	MB89P985	MB89PV980
	Specifying procedure	Specify when ordering masking	Setting with software	Setting with software
6	Clock mode selection • Dual-clock mode • Single-clock mode	Selectable	Selection by version number 101 : Single clock 201 : Dual clock	Selection by version number 101 : Single clock 201 : Dual clock

F_{CH}: Main clock source oscillation frequency

*: This option selects the state to which the oscillator stabilization "wait time" bits of the system control register (SYCC: WT1, WT0) are initialized at reset.

Table C-2 Mask Options (Segment Options)

Port pin names corresponding to LCD controller-driver common and segment outputs					Part number	MB89983
P40 to P43	P44 to P47	P60 to P61	P62 to P65	P70 P71	Specifying procedure	Specify when ordering masking
O	O	O	O	O	SEG0 to SEG13 (14 pins) COM0 to COM3 (4 pins)	Specify as SEG=3
X	O	O	O	O	SEG4 to SEG13 (10 pins) COM0 to COM3 (4 pins)	Specify as SEG=2
X	X	O	O	O	SEG8 to SEG13 (6 pins) COM0 to COM3 (4 pins)	Specify as SEG=1
X	X	X	X	X	No Segment output COM0, COM1 (2 pins)	Specify as SEG=0

O: Used as common/segment output pins. (Pull-up resistors may not be selected for these pins.)

X: Used as output-only port pins. (Except for pins P70 and P71, pull-up resistors may be selected.)

Table C-3 Version number

Version			Features
Mass production product	One-time PROM product	Piggyback product	Clock mode
MB89983	MB89P985-101	MB89PV980-101	Single clock
MB89983	MB89P985-201	MB89PV980-201	Dual clock

Table C-4 Ordering Information

Part number	Package	Remarks
MB89983-xxx-PFV	64-pin Plastic LQFP (FPT-64P-M03)	
MB89983-xxx-PFM	64-pin Plastic QFP (FPT-64P-M09)	
MB89P985PFV-101	64-pin Plastic LQFP (FPT-64P-M03)	Single clock
MB89P985PFM-101	64-pin Plastic QFP (FPT-64P-M09)	
MB89P985PFV-201	64-pin Plastic LQFP (FPT-64P-M03)	Dual clock
MB89P985PFM-201	64-pin Plastic QFP (FPT-64P-M09)	
MB89PV980-101	64-pin Ceramic MQFP (MQP-64C-P01)	Single clock
MB89PV980-201	64-pin Ceramic MQFP (MQP-64C-P01)	Dual clock

APPENDIX D Programming Specifications for One-Time PROM and EPROM Microcontrollers

In EPROM mode, the MB89P985 function equivalent to the MBM27C256A. This allows the PROM to be programmed with a general-purpose EPROM programmer by using the dedicated adaptor. Note that the electronic signature mode cannot be used.

D.1 Programming to the One-Time PROM Microcontroller

■ EPROM Programmer Socket Adaptor

Connect the jumper pin on the adapter to VSS.

Depending on the EPROM programmer, inserting a capacitor of about 0.1 µF between Vpp and Vss or Vcc and Vss can stabilize programming operations.

Table D.1-1 "EPROM Programmer Socket Adaptor" lists the EPROM programmer socket adaptors.

Table D.1-1 EPROM Programmer Socket Adaptor

Package	Compatible socket adaptor
FPT-64P-M03	ROM-64SQF-28DP-8L3
FPT-64P-M09	ROM-64QF2-28DP-8L4

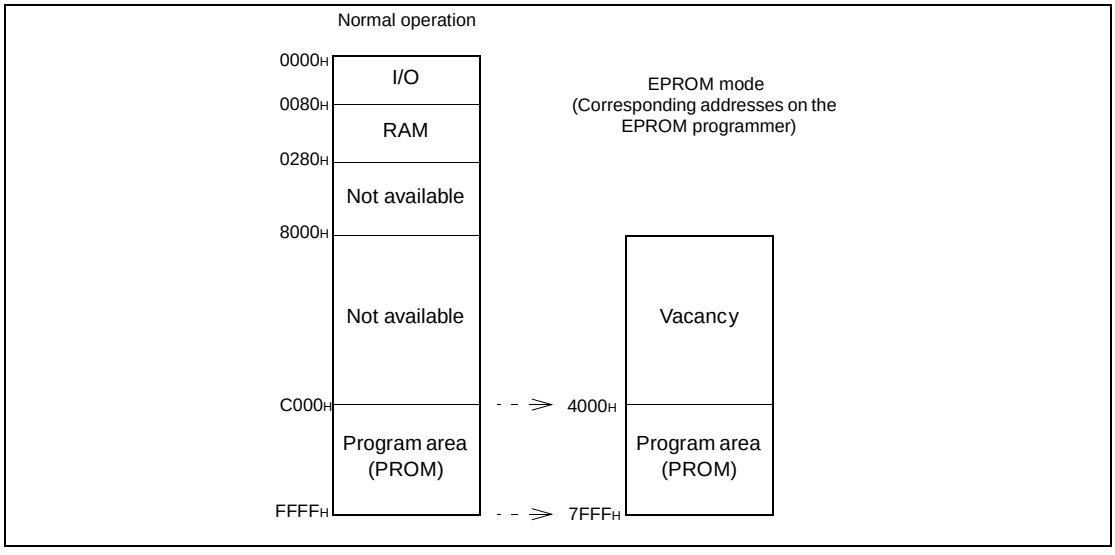
Inquiry**:**

Sun Hayato Co., Ltd.: TEL. 81-3-3802-5760

■ Memory Map in EPROM Mode

Figure D.1-1 'Memory Map in EPROM Mode" shows the memory map in EPROM mode. Write the option data in the option setting area after consulting the "OTPROM Option Bit Map".

Figure D.1-1 Memory Map in EPROM Mode

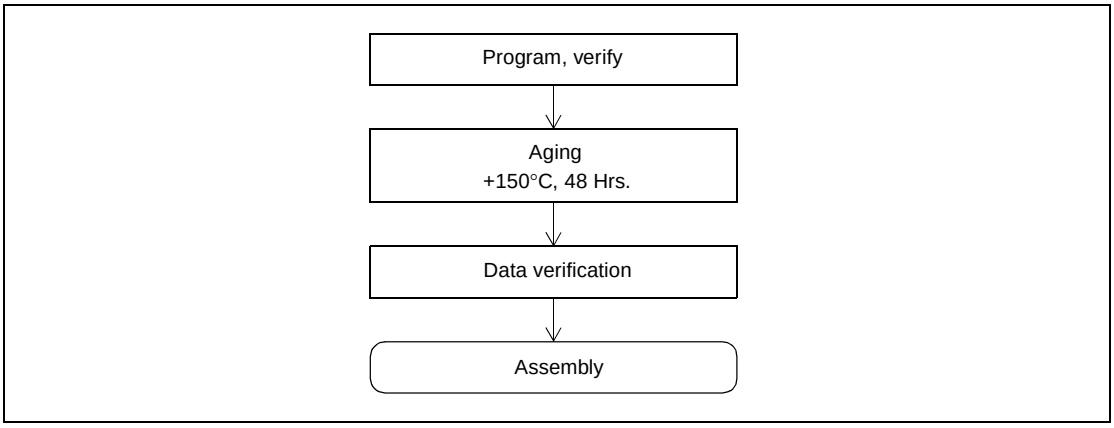


■ Recommended Screening Conditions

High-temperature aging is recommended as the pre-assembly screening procedure for a product with a blanked OTPROM microcomputer program.

Figure D.1-2 "Screening Procedure" shows the screening procedure.

Figure D.1-2 Screening Procedure



D.2 Programming Yield and Erasure

This section describes the programming yield and the data erasure on EPROM microcomputer.

■ Programming Yield

All bits cannot be programmed at Fujitsu shipping test to a blanked OTPROM microcomputer, due to its nature. For this reason, a programming yield of 100% cannot be assured at all times.

■ Notes on Using and Data Erasure on EPROM Microcomputer

❍ Erasure

In order to clear all locations of their programmed contents, it is necessary to expose the internal EPROM to an ultraviolet light source. A dosage of 10 W-seconds/cm2 is required to completely erase an internal EPROM. This dosage can be obtained by exposure to an ultraviolet lamp (wavelength of 2537 Angstroms (Å)) with intensity of 12000 mW/cm2 for 15 to 21 minutes. The internal EPROM should be about one inch from the source and all filters should be removed from the UV light source prior to erasure.

It is important to note that the internal EPROM and similar devices, will erase with light sources having wavelengths shorter than 4000 Å . Although erasure time will be much longer than with UV source at 2537 Å, nevertheless the exposure to fluorescent light and sunlight will eventually erase the internal EPROM, and exposure to them should be prevented to realize maximum system reliability. If used in such an environment, the package windows should be covered by an opaque label or substance.

D.3 Programming to the EPROM with Piggyback/Evaluation Device

This section describes the programming to the EPROM with piggyback/evaluation device.

■ EPROM for Use

MBM27C256A-20TV

■ Programming Socket Adaptor

To program to the PROM using an EPROM programmer, use the socket adaptor (manufacturer: Sun Hayato Co., Ltd.) listed below.

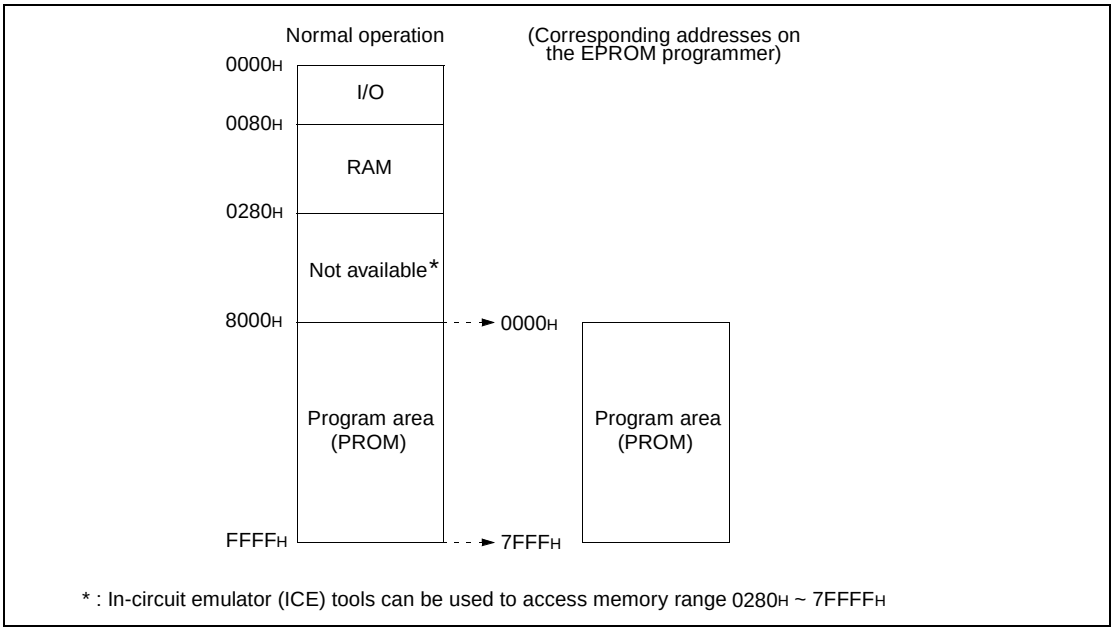
Table D.3-1 Programming Socket Adaptor

Package	Adaptor socket part number
LCC-32 (Rectangle)	ROM-32LC-28DP-YG

Inquiry: Sun Hayato Co., Ltd.: TEL 81-3-5396-9106

■ Memory Space

Figure D.3-1 Memory Map of Piggyback/Evaluation Device



■ Programming to EPROM

1. Set the EPROM programmer to the MBM27C256A.
2. Load program data into the EPROM programmer at 0000_H to $7FFF_H$.
3. Program to 0000H to $7FFF_H$ with the EPROM programmer.

APPENDIX E MB89980 Series Pin States

This section describes the pin states of the MB89980 series in each mode.

■ Pin States in Each Mode

Table E-1 Pin States in Each Mode

Pin name	Normal operation	Sleep mode	Stop mode SPL="0"	Stop mode SPL="1"	During a reset
P00/INT20 to P07/INT27	Port I/O/external interrupt 2 input	Hold/external interrupt 2 input	Hold/external interrupt 2 input	Hi-z/external interrupt 2 input	Hi-z
P10/INT10 to P13/INT13	Port I/O/external interrupt 1 input	Hold/external interrupt 1 input	Hold/external interrupt 1 input	Hi-z/external interrupt 1 input	
P14 to P17	Port I/O	Hold	Hold	Hi-z	
X0, X0A	Oscillator input	Oscillator input	Hi-z	Hi-z	Oscillator input
X1, X1A	Oscillator output	Oscillator output	"H" output	"H" output	Oscillator output
MOD0 MOD1	Mode input	Mode input	Mode input	Mode input	Mode input
RST	Reset input	Reset input	Reset input	Reset input	Reset input*1
P20/EC	Port or peripheral I/O	Hold/peripheral I/O	Hold	Hi-z	Hi-z
P21					
P22/TO					
P23					
P24/RCO					
P25					
P26					
P27/PWM2					Hi-z *6
P30/PWM1/BZ					"H" output *6
P31					Input
P32					

Table E-1 Pin States in Each Mode

Pin name	Normal operation	Sleep mode	Stop mode SPL="0"	Stop mode SPL="1"	During a reset
P40 to P47*2 P60 to P65*2 P70,P71*3	Port or peripheral I/O	Hold/peripheral I/O	Hold	Hi-z	Hi-z
P50/AN0 to P53/AN3					
COM0 to COM3*4	LCD common output	LCD common output	Hold*5	Hold*5	"L" output
SEG0 to SEG13*4	LCD segment output	LCD segment output			

*1: The reset pin can function as an output depending on an option setting.

*2: If segment output is selected, the states of these pins are as indicated for pins SEG0 to SEG13.

*3: If common output is selected, the states of these pins are as indicated for COM2 and COM3.

*4: Pins COM2, COM3 and SEG0 to SEG13 are also used as general-purpose output ports (selected by mask option in MB89983 and by software in MB89P985 and MB89PV980).

*5: Operate as common/segment output if an LCD controller-driver operating clock is supplied.

*6: Pin state of P27 and P30 are undetermined until the internal clock starts operation.

Hi-Z: High impedance. Pin with a pull-up resistor being selected will go to the pull-up state.

SPL: Pin state specification bit in the standby control register (STBC)

SPL: The pin set as output holds its state (level) before changing to each mode.

INDEX

The index follows on the next page.
This is listed in alphabetic order.

Index

F

G

H

I

L

M

O

P

R

S

T

V

W

CM25-10141-1E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

F^2MC-8L
8-BIT MICROCONTROLLER
MB89980 Series
HARDWARE MANUAL

March 2000 the first edition

Published **FUJITSU LIMITED** Electronic Devices

Edited Technical Communication Dept.

FUJITSU

CM25-10141-1E

FUJITSU SEMICONDUCTOR F²MC-8L 8-Bit MICROCONTROLLER MB89980 Series HARDWARE MANUAL