FUJITSU SEMICONDUCTOR
MICROCONTROLLER MANUAL

CM25-10133-2E

F^2MC-8L FAMILY

8-BIT MICROCONTROLLER

MB89990 Series

HARDWARE MANUAL

FUJITSU

F²MC-8L FAMILY

8-BIT MICROCONTROLLER

MB89990 Series

HARDWARE MANUAL

FUJITSU LIMITED

PREFACE

Preface describes objectives and intended reader.

■ Objectives and Intended Reader

The MB89990 series of microcontrollers are mid-range of microcontroller. They are general-purpose and high-speed products in the F^2MC-8L Family series of 8-bit single-chip microcontrollers operating at low voltages. It has Timer, Remote-control transmission frequency generator.

This manual covers the functions and operations of the MB89990 series of microcontrollers. Refer to the ***F^2MC-8L Family Software Manual*** for instructions.

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.

©2000 FUJITSU LIMITED Printed in Japan

CONTENTS

FIGURES

TABLES

CHAPTER 1 GENARAL

The MB89990 series contains microcontrollers with a full range of resources such as timers, external interrupts, and remote-control function, as well as the F^2MC-8L CPU core for low-voltage and high-speed operation. This single-chip microcontroller is suitable for small devices such as remote controllers incorporating compact packages.

1.1 Features

This section describes the features.

■ Features

- Minimum instruction execution time: 0.95 μs at 4.2 MHz (V_{CC} = 3 V)
- CPU core common to F^2MC-8L CPU
  - Instruction set suitable for controller: - Multiply/subtraction instruction, - 16-bit operation, - Instruction test and branch instruction, - Bit operation instruction
- Two timers
  - 8/16-bit timer/counter
  - 20-bit time-base counter
- External interrupts
  - Edge detection: 3 pins (edge direction enabled)
  - Low-level interrupt: 8 pins (wake-up)
- Built-in remote-control carrier frequency generator
- Low-power consumption modes
  - Stop mode: Almost no power consumption because oscillation stopped
  - Sleep mode: 33% of normal power consumption because CPU stopped
- Package: SOP-28, SH-DIP-28 (mask ROM only)

1.2 Product Series

This section describes the product series.

■ Product Series

Table 1.2-1 "Types and Functions of MB89990 Series of Microcontrollers"lists the types and functions of the MB89990 series of microcontrollers.

Table 1.2-1 Types and Functions of MB89990 Series of Microcontrollers

Model Name	MB89997	MB89P195*	MB89PV190*
Classification	Mass-produced product (mask ROM product)	One-time product	For evaluation and development
ROM capacity	32 K × 8 bits (internal ROM)	16 K × 8 bits (internal ROM, write enable by general-purpose writer)	32 K × 8 bits (external ROM)
RAM capacity	128 × 8 bits	256 × 8 bits	
CPU functions	Number of basic instructions: 136 instructions Instruction bit length: 8 bits Instruction length: 1 to 3 bytes Data bit length: 1, 8, 16 bits Minimum instruction execution time: 0.95 µs/4.2 MHz Interrupt processing time: 8.6 µs/4.2 MHz		
Port	I/O port (N-ch open drain): 6 pins I/O port (CMOS): 16 pins (13 used as resource pins) Total: 22 pins		
Timer counter	2 channels 8-bit counter or 1 channel 16-bit counte		
External-interrupt 1	3 independent channels (edge selection, interrupt vector, interrupt source flag) Interrupt mode selectable from rising edge, falling edge, or both edges For releasing Stop/Sleep modes (edge detection in Stop mode enabled)		
External-interrupt 2 (Wake-up)	8 channels (Low-level interrupt enabled)		
Remote-control carrier frequency	Pulse width and cycle are programmable		
Standby mode	Sleep and Stop modes		
Process	CMOS		
Package	FPT-28P-M02, DIP-28P-M03	FPT-28P-M02	MQP-48C-P01
Operating voltage	2.2 to 6.0 V**	2.7V to 6.0 V	

* The MB89P195 microtroller is the one-time product for the MB89190 series which can be also be used for the MB89990 series.

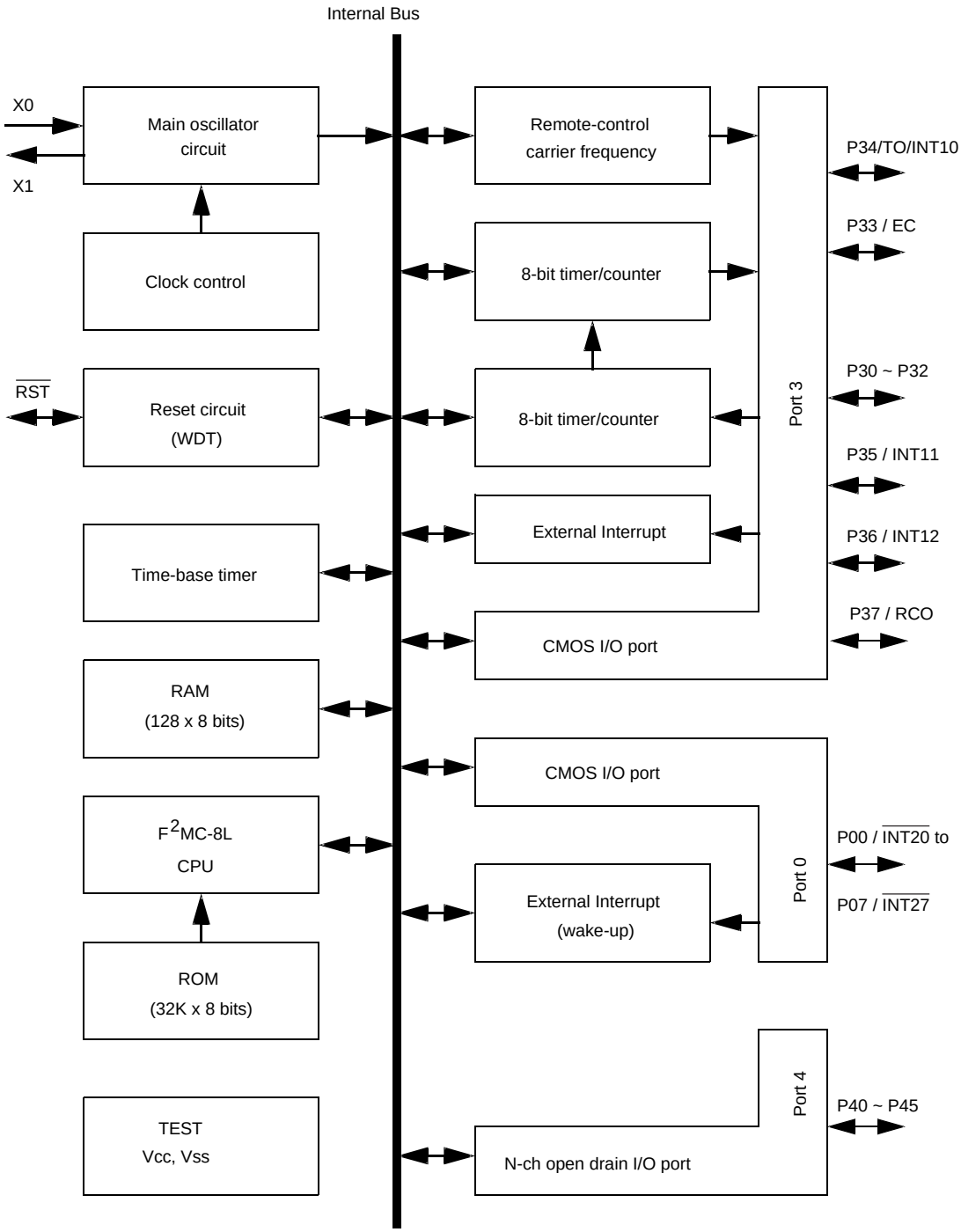
* The MB89PV190 microtroller is the evaluation and development product for the MB89190 series which can be also be used for the MB89990 series.

** Operating voltage varies with conditions such as frequency or others. See the data sheet for details.

1.3 Block Diagram

This section describes the block diagram.

■ Block Diagram



1.4 Pin Assignment

This section describes the pin assignment.

■ Pin Assignment

Figure 1.4-1 Pin Assignment (FPT-28P-M02, DIP-28P-M03)

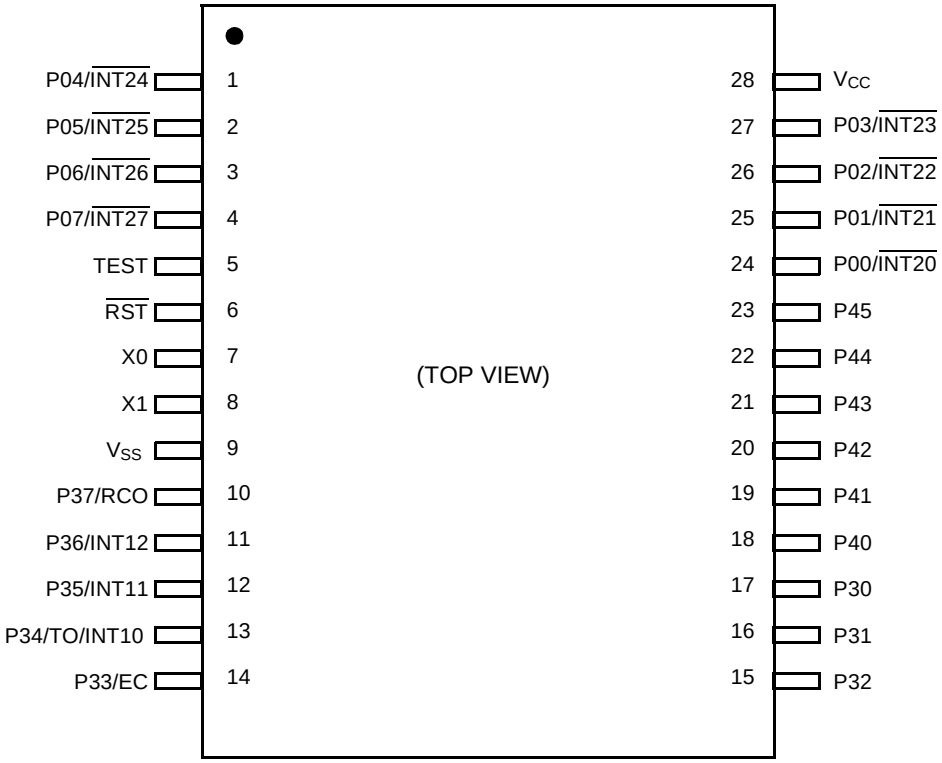
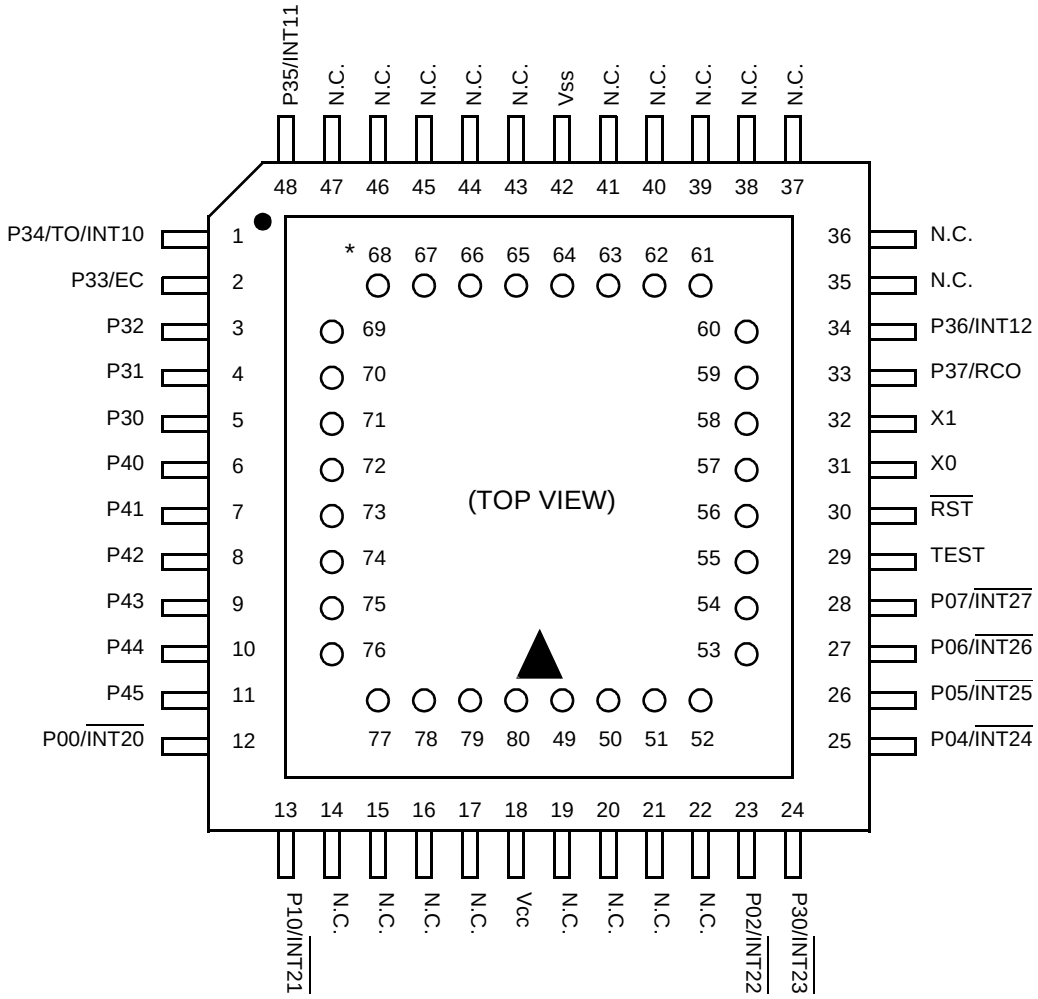


Figure 1.4-2 Pin Assignment (MQP-48C-P01)



* Pin assignment on package top (only for piggyback/evaluation product)

Pin No.	Symbol	Pin No.	Symbol	Pin No.	Symbol	Pin No.	Symbol
49	V_{PP}	57	NC	65	04	73	$\overline{OE}$
50	A12	58	A2	66	05	74	NC
51	A7	59	A1	67	06	75	A11
52	A6	60	A0	68	07	76	A9
53	A5	61	01	69	08	77	A8
54	A4	62	02	70	$\overline{CE}$	78	A13
55	A3	63	03	71	A10	79	A14
56	NC	64	GND	72	NC	80	V_{CC}

1.5 Pin Function Description

This section describes the pin functions.

■ Pin Function Description

Table 1.5-1 "Pin Function Description" and Table 1.5-2 "Pins for External ROM" list the pin function and Table 1.5-3 "Input/Output Circuit Configurations" shows the input/output circuit configurations.

Table 1.5-1 Pin Function Description

Pin No.	Pin Name	Circuit type	Function
7	X0	A	**Clock oscillator pins**
8	X1		
5	TEST	B	**Test input pin** These pins are connected directly to V_{SS}.
6	$\overline{RST}$	C	**Reset I/O pin** This pin consists of an N-ch open-drain output with a pull-up resistor and hysteresis input. A Low level is output from this pin by internal source. The internal circuit is initialized at input of a Low level.
24 to 27	P00/$\overline{INT20}$ to P03/$\overline{INT23}$	D	**General-purpose I/O port** These ports also serve as external interrupt input pin. The external interrupt input is hysteresis type.
1 to 4	P04/$\overline{INT24}$ to P07/$\overline{INT27}$	D	**General-purpose I/O port** These ports also serve as external interrupt input pin. The external interrupt input is hysteresis type.
17	P30	E	**General-purpose I/O port**
16	P31	E	**General-purpose I/O port**
15	P32	E	**General-purpose I/O port**
14	P33/EC	D	**General-purpose I/O port** This port also serves as an external clock input pin for the 8-bit timer/counter. The external clock input is hysteresis type with a built-in noise filter.
13	P34/T0/INT10	D	**General-purpose I/O port** This port also serves as an overflow output pin and an external interrupt input pin for the 8-bit timer/counter. The external interrupt input is hysteresis type with a built-in noise filter.
12	P35/INT11	D	**General-purpose I/O port** This port also serves as an external interrupt input pin. The external interrupt input is hysteresis type with a built-in noise filter.
11	P36/INT12		

Table 1.5-1 Pin Function Description (Continued)

Pin No.	Pin Name	Circuit type	Function
10	P37/RCO	E	**General-purpose I/O port** This port also serves as remote-control output pin.
18 to 23	P40 to P45	F	**N-ch open-drain type I/O port**
28	V_{CC}	—	**Power pin**
9	V_{SS}	—	**Power (GND) pin**

Table 1.5-2 Pins for External ROM

Pin No.	Pin Name	Circuit type	Functinon
49	V_{PP}	Output	**High-level output pin**
79 78 50 75 69 76 77 51 52 53 54 55 58 59 60	A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0	Output	**Address-output pins**
61 62 63 65 66 67 68 69	01 02 03 04 05 06 07 08	Intput	**Data-input pins**
70	$\overline{CE}$	Output	**Chip-enable pin for ROM** A High level is output in the standby mode.
73	$\overline{OE}$	Output	**Output-enable pin for ROM** A Low level is always output.
80	V_{CC}	Output	**Power pin for EPROM**
64	V_{SS}	Output	**Power (GND) pin**

Table 1.5-3 Input/Output Circuit Configurations

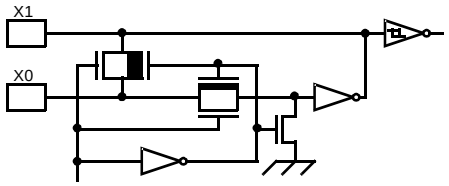
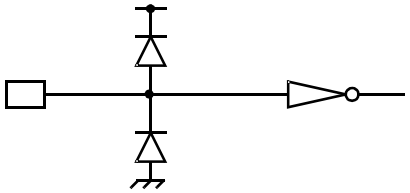
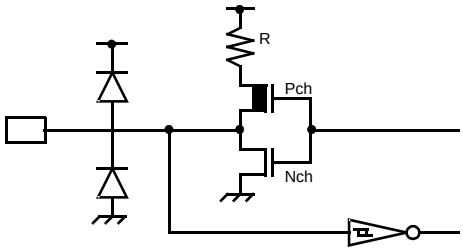
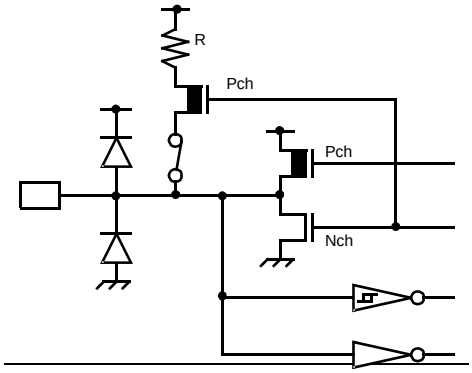
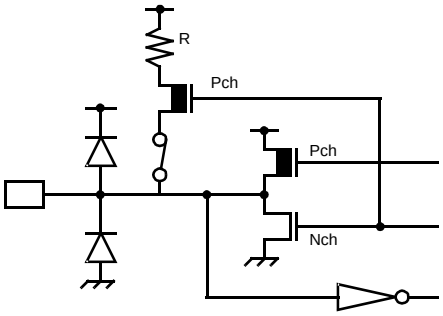
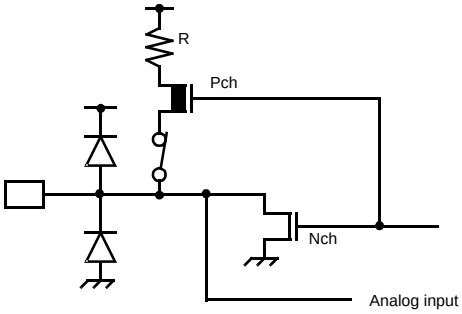
Classification	Circuit	Remarks
A	 Standby control signal	• Crystal oscillator • Feedback resistor: About 1 MΩ/5V (1 to 5MHz)
B		• CMOS input
C		• Output pull-up resistor (P-ch): About 50 kΩ (5 V) • Hysteresis input
D		• CMOS input/output • Hysteresis input (resorce input) • The pull-up resistor is available.

Table 1.5-3 Input/Output Circuit Configurations (Continued)

Classification	Circuit	Remarks
E		• CMOS input/output • The pull-up resistor is available.
F		• N-ch open-drain output • Analog input • The pull-up resistor is available. (MB89990 series only)

1.6 Handling Devices

This section describes handling devices.

■ Handling Devices

(1) Preventing latch-up

Latch-up may occur if a voltage higher than V_{CC} or lower than V_{SS} is applied to the input or output pins other than middle- and high-level-resistant pins, or if voltage exceeding the rated value is applied between V_{CC} and V_{SS}. When latch-up occurs, the supply current increases rapidly, sometimes resulting in overheating and destruction. Therefore, no voltage exceeding the maximum ratings should be used.

(2) Handling unused input pins

Leaving unused input pins open may cause a malfunction. Therefore, these pins should be set to pull-up or pull-down.

(3) Variations in supply voltage

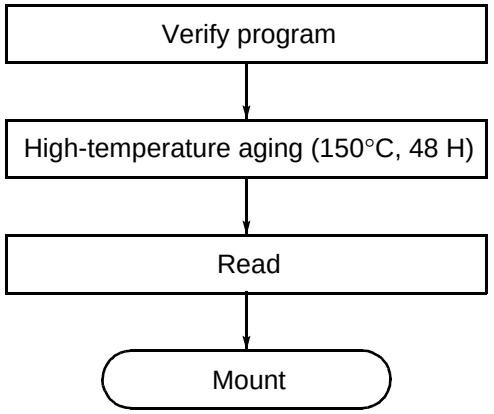
Although the specified V_{CC} supply voltage operating range is assured, a sudden change in the supply voltage within the specified range may cause a malfunction. Therefore, the voltage supply to the IC should be kept as constant as possible. The V_{CC} ripple (P-P value) at the supply frequency (50 to 60 Hz) should be less than 10% of the typical V_{CC} value, or the coefficient of excessive variation should be less than 0.1 V/ms instantaneous change when the power supply is switched.

(4) Precautions for external clocks

It takes some time for oscillation to stabilize after changing the mode from power-on reset (option selection) and stop mode. Consequently, an external clock must be input.

(5) Recommended screening conditions

The OPTROM product should be screened by high-temperature aging before mounting.



The programming test cannot be performed for all bits of the preprogrammed OPTROM product due to its characteristics. Consequently, 100% programming yielding cannot be ensured.

CHAPTER 2 HARDWARE CONFIGRATION

This chapter describes each block of the CPU hardware.

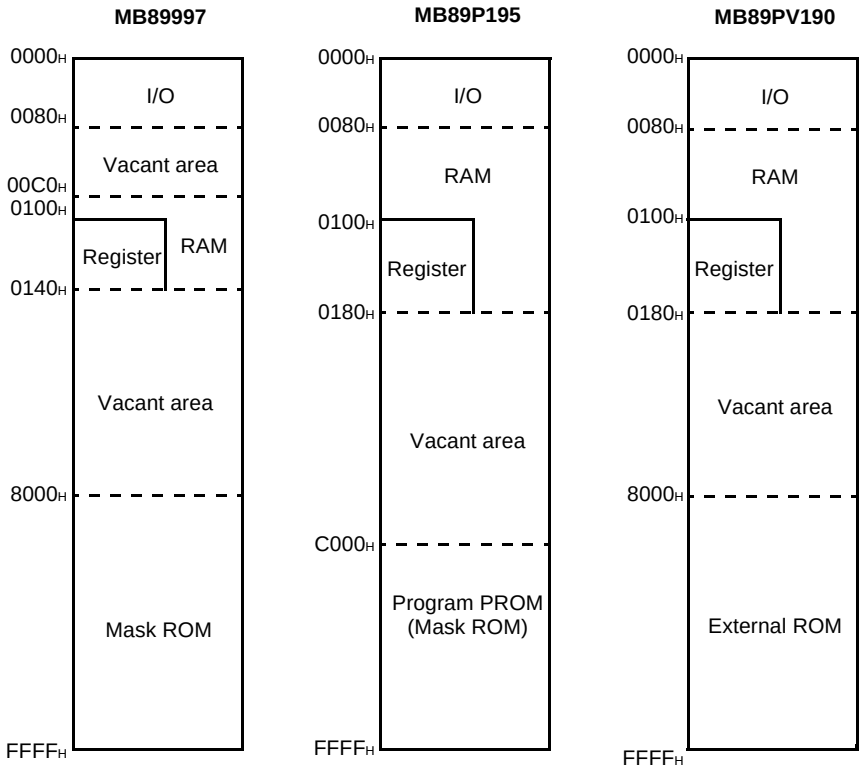
2.1 CPU

- **This section describes the memory space and register composing CPU hardware.**

■ Memory Space

The MB89990 series of microcontrollers have a memory area of 64K bytes. All I/O, data areas, and program areas are located in this space. The I/O area is at the lowest address and the data area is immediately above it. The data area may be divided into register, stack, and direct-address areas according to the applications. The program area is located near the highest address and the tables of interrupt and reset vectors and vector-call instructions are at the highest address. Figure 2.1-1 "Memory Space of MB89990 Series of Microcontrollers" shows the structure of the memory space for the MB89990 series of microcontrollers.

Figure 2.1-1 Memory Space of MB89990 Series of Microcontrollers



- I/O area
  - This area is where various resources such as control and data registers are located. The memory map for the I/O area is given in APPENDIX A .
- RAM area
  - This area is where the static RAM is located. Addresses from 0100_H to $017F_H$ (0100_H to $013F_H$ for the MB89997) are also used as the general-purpose register area.
- ROM area
  - This area is where the internal ROM is located. Addresses from $FFC0_H$ to $FFFF_H$ are also

used for the table of reset and vector-call instructions. Table 2.1-1 "Table of Reset and Interrupt Vectors " shows the correspondence between each interrupt number or reset and the table addresses to be referenced for the MB89990 series of microcontrollers.

Table 2.1-1 Table of Reset and Interrupt Vectors

	Table address	
	Upper data	**Lower data**
CALLV #0	$FFC0_H$	$FFC1_H$
CALLV #1	$FFC2_H$	$FFC3_H$
CALLV #2	$FFC4_H$	$FFC5_H$
CALLV #3	$FFC6_H$	$FFC7_H$
CALLV #4	$FFC8_H$	$FFC9_H$
CALLV #5	$FFCA_H$	$FFCB_H$
CALLV #6	$FFCC_H$	$FFCD_H$
CALLV #7	$FFCE_H$	$FFCF_H$

	Table address	
	Upper data	**Lower data**
Interrupt #11	$FFE4_H$	$FFE5_H$
Interrupt #10	$FFE6_H$	$FFE7_H$
Interrupt #9	$FFE8_H$	$FFE9_H$
Interrupt #8	$FFEA_H$	$FFEB_H$
Interrupt #7	$FFEC_H$	$FFED_H$
Interrupt #6	$FFFE_H$	$FFEF_H$
Interrupt #5	$FFF0_H$	$FFF1_H$
Interrupt #4	$FFF2_H$	$FFF3_H$
Interrupt #3	$FFF4_H$	$FFF5_H$
Interrupt #2	$FFF6_H$	$FFF7_H$
Interrupt #1	$FFF8_H$	$FFF9_H$
Interrupt #0	$FFFA_H$	$FFFB_H$
Reset mode	– – – –	$FFFD_H$
Reset vector	$FFFE_H$	$FFFF_H$

Note:

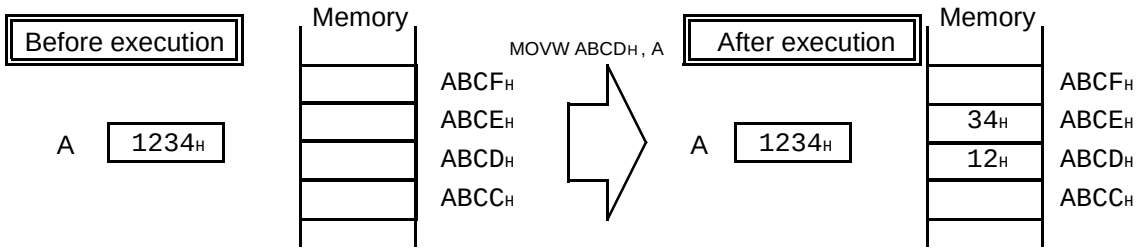
$FFFC_H$ is already reserved.

Set 00_H for $FFFD_H$ in the Reset mode.

■ Arrangement of 16-bit Data in Memory

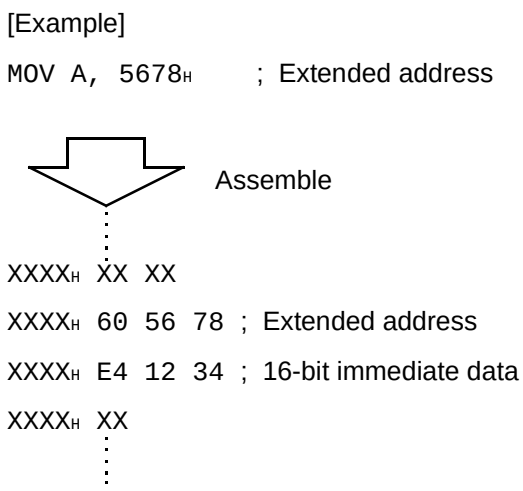
When the MB89990 series of microcontrollers handle 16-bit data, the data written at the lower address is treated as the upper data and that written at the next address is treated as the lower data as shown in Figure 2.1-2 "Arrangement of 16 bit Data in Memory".

Figure 2.1-2 Arrangement of 16 bit Data in Memory



This is the same as when 16 bits are specified by the operand during execution of an instruction. Bits closer to the OP code are treated as the upper byte and those next to it are treated as the lower byte. This is also the same when the memory address or 16-bit immediate data is specified by the operand.

Figure 2.1-3 Arrangement of 16-bit Data during Execution of Instruction



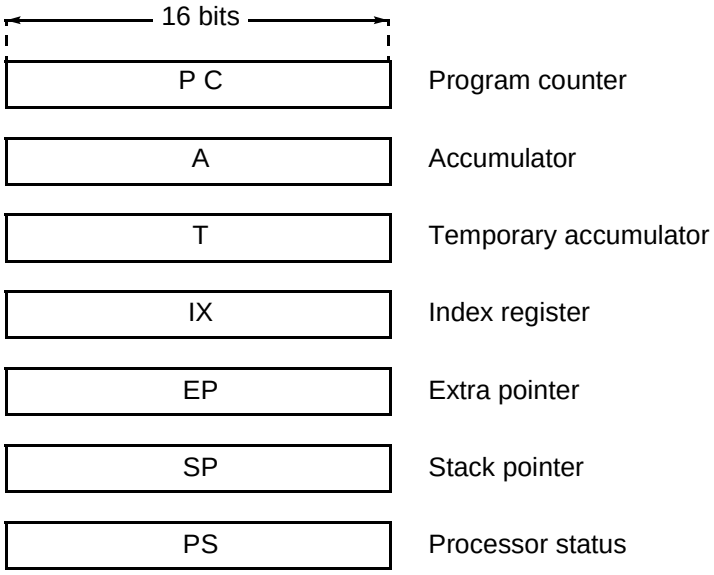
Data saved in the stack by an interrupt is also treated in the same manner.

■ Internal Registers in CPU

The MB89990 series of microcontrollers have dedicated registers specified applications in the CPU and general-purpose registers in memory.

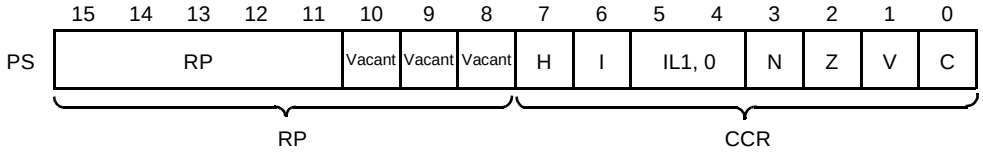
- Program counter (PC) — 16-bit long register indicating location where instructions stored
- Accumulator (A) — 16-bit long register where results of operations stored temporarily. The lower byte is used to execute 8-bit data processing instructions.
- Temporary accumulator (T) — 16-bit long register where the operations are performed between this register and the accumulator. The lower byte is used to execute 8-bit data processing instructions.
- Stack pointer (SP) — 16-bit long register indicating stack area

- Processor status (PS) — 16-bit long register where register pointers and condition codes stored
- Index register (IX) — 16-bit long register for index modification
- Extra pointer (EP) — 16-bit long register for memory addressing



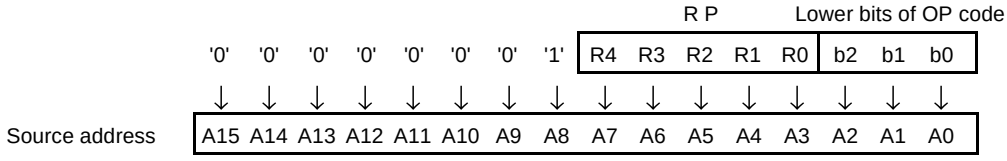
The 16 bits of the processor status (PS) can be divided into 8 upper bits for a register bank pointer (RP) and 8 lower bits for a condition code register (CCR). (See Figure 2.1-4 "Structure of Processor Status".)

Figure 2.1-4 Structure of Processor Status



The RP indicates the address of the current register bank. The contents of the RP and the real addresses are translated as shown in Figure 2.1-5 "Rule for Translating Real Addresses at General-purpose Register Area" .

Figure 2.1-5 Rule for Translating Real Addresses at General-purpose Register Area



The CCR has bits indicating the results of operations and transfer data contents, and bits controlling the CPU operation when an interrupt occurs.

- H-flag — H-flag is set when a carry or a borrow out of bit 3 into bit 4 is generated as a result of operations. It is cleared in other cases. This flag is used for decimal-correction instructions.

- I-flag — An interrupt is enabled when this flag is 1 and is disabled when it is 0. The I-flag is 0 at reset.

- IL1 and IL0 — These bits indicate the level of the currently-enabled interrupt. The Interrupt Processing executes interrupt processing only when an interrupt with a value smaller than the value indicated by this bit is requested.

IL1	IL0	Interrupt level	High and low
0	0	1	High ↕ Low = No interrupt
0	1		
1	0	2	
1	1	3	

- N-flag — The N-flag is set when the most significant bit is 1 as a result of operations. It is cleared when the MSB is 0.

- Z-flag — Z-flag is set when the bit is 0 as a result of operations. It is cleared in other cases.

- V-flag — V-flag is set when a twoís complement overflow occurs as a result of operations. It is reset when an overflow does not occur.

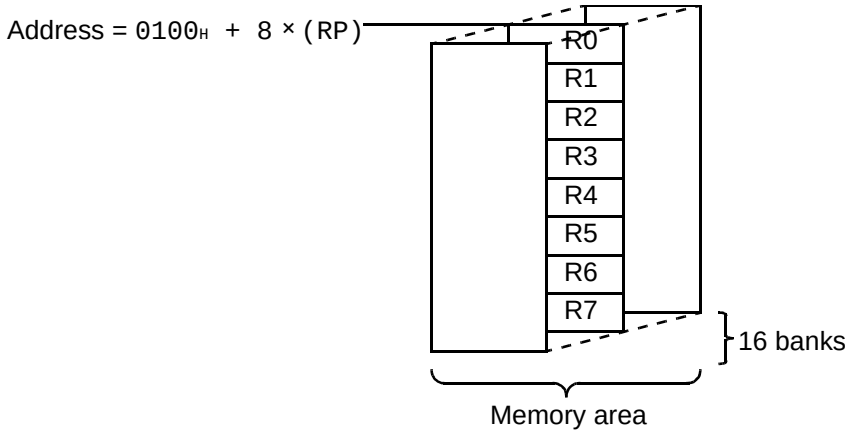
- C-flag — C-flag is set when a carry or a borrow out of bit 7 is generated as a result of operations. It is cleared in other cases. When the shift instruction is executed, the value of the C-flag is shifted out.

• General-purpose registers

General-purpose registers are 8-bit long registers for storing data.

The 8-bit long general-purpose registers are in the register banks in memory. One bank has eight registers and up to 16 banks are available for the MB89193 (8 banks for the MB89191). The register bank pointer (RP) indicates the currently-used bank.

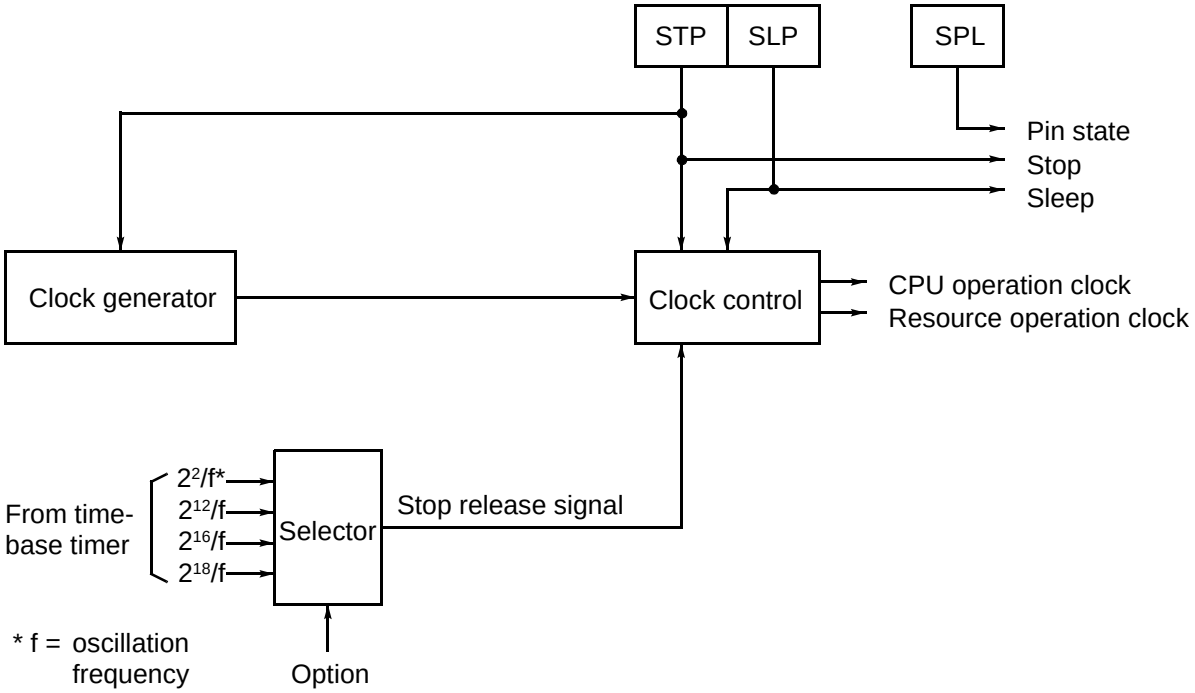
Figure 2.1-6 Register Bank Configuration



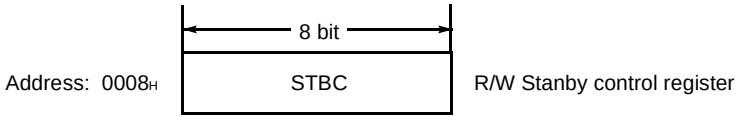
2.2 Lock Control Block

- **This block controls the standby operation and software reset.**

■ Machine Clock Control Block Diagram



■ Register List



■ Description of Registers

The detail of each register is described below.

- Standby-control register (STBC)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0008_H	STP	SLP	SPL	RST	—	—	—	—
	(W)	(W)	(R/W)	(W)				
				Intilial value $0001XXXX_B$				

[Bit 7] STP: Stop bit

This bit is used to specify switching CPU to the stop mode.

0	No operation
1	Stop mode

This bit is cleared at reset or stop cancellation.

0 is always read when this bit is read.

[Bit 6] SLP: Sleep bit

This bit is used to specify switching the CPU and resources to the sleep mode.

0	No operation
1	Sleep mode

This bit is cleared at reset, sleep or stop cancellation.

0 is always read when this bit is read.

[Bit 5] SPL: Pin state specifying bit

This bit is used to specify the external pin state in the stop mode.

0	Holds state and level immediately before stop mode
1	High impedance

This bit is cleared at resetting.

[Bit 4] RST: Software reset bit

This bit is used to specify the software reset.

0	Generates 4-cycle reset signal
1	No operation

1 is always read when this bit is read.

■ Description of Operation

Main/sub clock block has normal and low-power consumption mode. The low-power consumption mode are described below.

(1) Low-power consumption mode

This chip has three operation modes. The sleep mode and stop mode in the table below reduce the power consumption.

Table 2.2-1 Opeating State of Low-power Consumption Modes

Clock mode of CPU	Clock pulse	Each operating clock pulse (4 Mhz clock)			Wake-up source in each mode
		CPU	Time base timer	Each resource	
RUN	Oscillates	2.0 MHz	2.0 MHz	2.0 MHz	Various interrupt requests
Sleep		Stops			

- The SLEEP mode stops only the operating clock pulse of the CPU. Other operations are continued.
- The STOP mode stops the oscillation. Data can be held with the lowest power consumption in this mode.

(a) SLEEP state

- Switching to Sleep State
  - Writing 1 at the SLP bit (bit6) of the STBC register switches the mode to SLEEP state.
  - The SLEEP state is the mode to stop clock pulse operating the CPU. Only the CPU stops and the resources continue to operate.
  - If an interrupt is requested when 1 is written at the SLP bit (bit 6), instruction execution continues without switching to the SLEEP state.
  - In the SLEEP state, the values of registers and RAM immediately before entering the SLEEP state are held.
- Cancelling SLEEP state
  - The SLEEP state is cancelled by inputting the reset signal and requesting an interrupt.
  - When the reset signal is input during the SLEEP state, the CPU is switched to the reset state and the SLEEP state is cancelled.
  - When an interrupt level higher than 11 is requested from a resource during the SLEEP state, the SLEEP state is cancelled.
  - When the I flag and IL bit are enabled interrupt like an ordinary interrupt after cancelling, the CPU executes the interrupt processing. When they are disabled, the CPU executes the interrupt processing from the instruction next to the one before entering the SLEEP state.

(b) STOP state

- Switching to STOP state
  - Writing 1 at the STP bit (bit7) of the STBC register switches the mode to STOP state.
  - In the STOP state, the clock oscillation, CPU, and all resources are stopped.

- The input/output pins and output pins during the STOP state can be controlled by the SPL bit (bit5) of the STBC register so that they are held in the state immediately before entering the STOP state, or so that they enter in the high-impedance state.
- If an interrupt is requested when 1 is written at the STP bit (bit 7), instruction execution continues without switching to the STOP state.
- In the STOP state, the values of registers and RAM immediately before entering the STOP state are held.

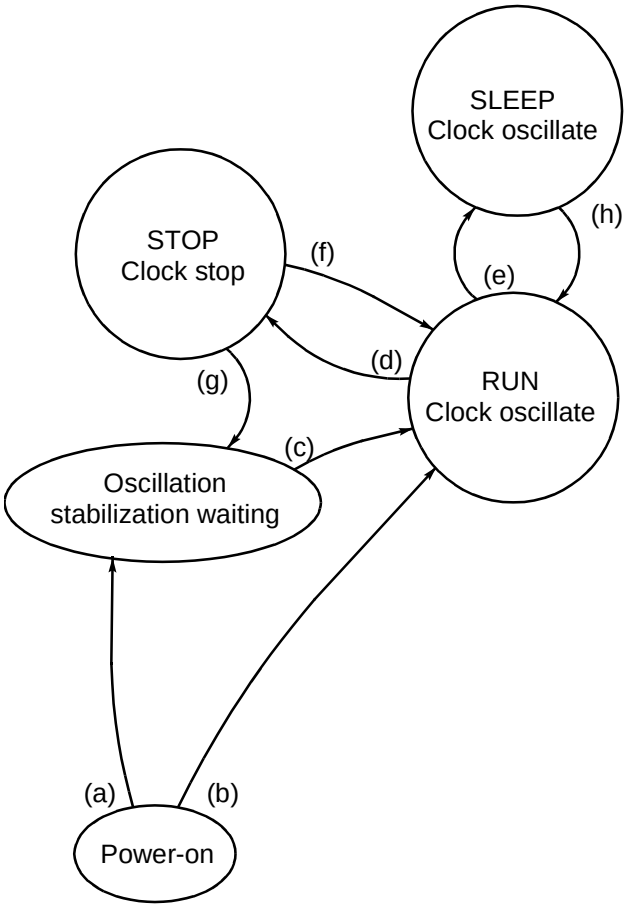
- Cancelling STOP state
  - The STOP state is cancelled either by inputting the reset signal or by requesting an interrupt.
  - When the reset signal is input during the STOP state, the CPU is switched to the reset state and the STOP state is cancelled.
  - When an interrupt higher than level 11 is requested from the external interrupt circuit during the STOP state, the STOP state is cancelled.
  - When the I flag and IL bit are enabled interrupt like an ordinary interrupt after cancelling, the CPU executes the interrupt processing. When they are disabled, the CPU executes the interrupt processing from the instruction next to the one before entering the STOP state.
  - The oscillation stabilization time can be selected by the option from any of the four types listed in Table 2.2-2 "Selection of Oscillation Stabilization Time".
  - If the STOP state is cancelled by inputting the reset signal, the CPU is switched to the oscillation stabilization wait state. Therefore, the reset sequence is not executed unless the oscillation stabilization time is elapsed. The oscillation stabilization time corresponds to the oscillation stabilization time of the main clock selected by the option. However, when Power-on Reset is not specified by the mask option, the CPU is not switched to the oscillation stabilization wait state even if the STOP state is cancelled by inputting the reset signal.

Table 2.2-2 Selection of Oscillation Stabilization Time

Ocillation stabilization time	Ocillation stabilization time with 4 MHz source clock
$2^{18}/f^*$	Approximate 65.5 ms
$2^{16}/f^*$	Approximate 16.4 ms
$2^{12}/f^*$	Approximate 1.2 ms
$2^{2}/f^*$	Approximate 0 ms

* f = source clock frequency

(2) State transition diagram at low power consumption mode



(a) When power-on reset option selected

(b) When power-on reset option not selected

(c) After oscillation stabilized

(d) Set STP bit to 1.

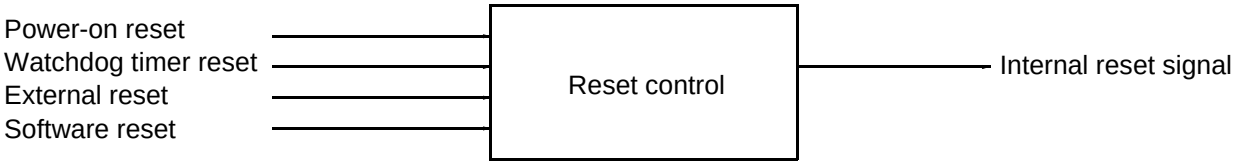
(e) Set SLP bit to 1.

(f) External reset when power-on reset option not selected

(g) External reset or interrupt when power-on reset option selected

(h) External reset or interrupt

■ Reset Control Section



- Reset
  - There are four types of resets as shown in Table 2.2-3 "Sources of Reset".

Table 2.2-3 Sources of Reset

Reset name	Description
Power-on reset	Turns power on
Watchdog reset	Overflows watchdog timer
External-pin reset	Sets external-reset pin to Low
Software reset	Writes 0 at RST bit (bit 4) of STBC

When the power-on reset and reset during the stop state are used, the oscillation stabilization time is needed after the oscillator operates because the oscillator stops. The time-base timer controls this stabilization time. Consequently, the operation does not start immediately even after cancelling the reset.

However, if the mask option without Power-on Reset is selected, no oscillation stabilization time is required in any state after external pins have been released from the reset.

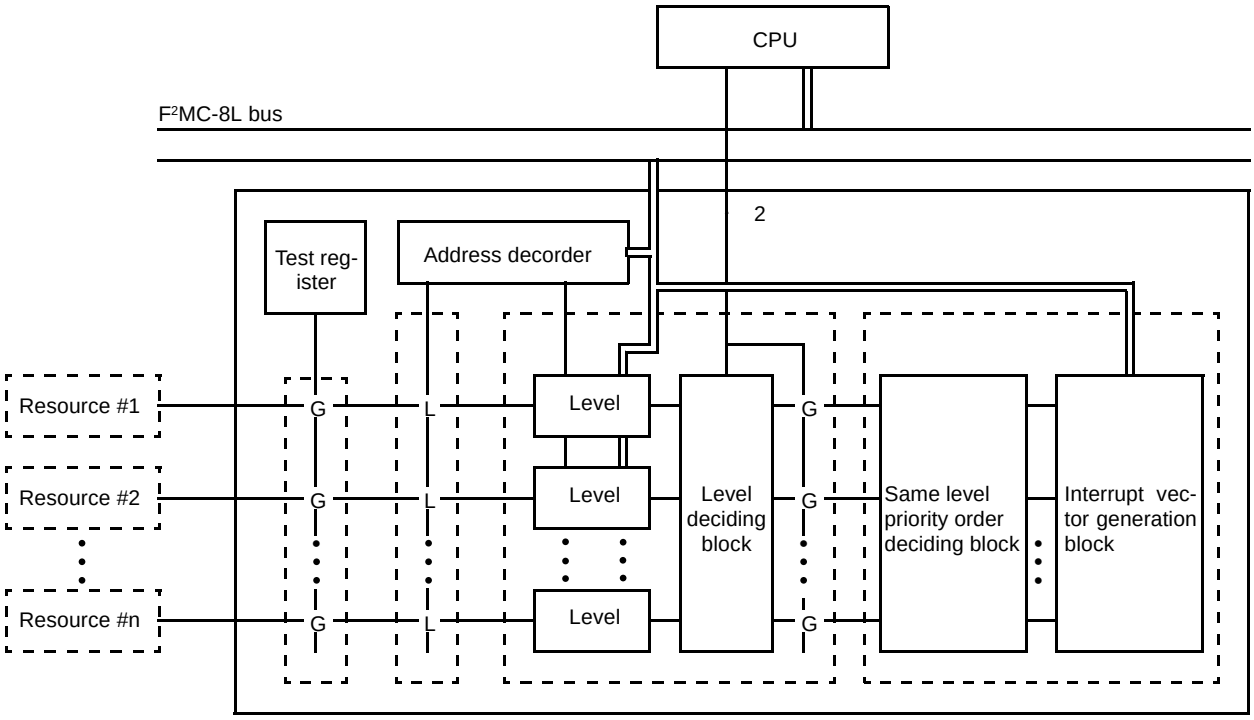
Note:

When resetting a product without the power-on reset function, set a longer time than the optional oscillation stabilization time. Otherwise, the reset timing matches the AC characteristics.

2.3 Interrupt Controller

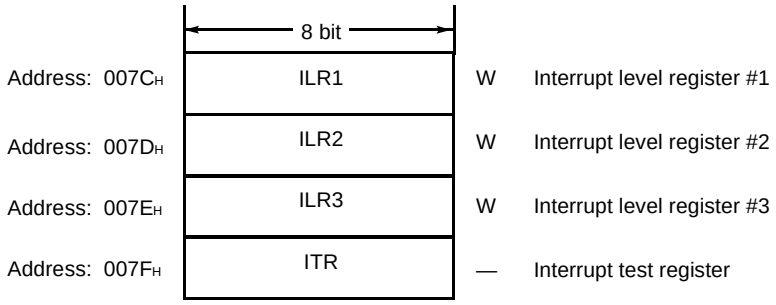
- **The interrupt controller for the F^{2}MC-8L family is located between the CPU and each resource. This controller receives interrupt requests from the resources, assigns priority to them, and transfers the priority to the CPU. It also decides the priority of same-level interrupts.**

■ Block Diagram



■ Register List

Interrupt controller consists of interrupt-level registers (ILR1, 2, and 3) and interrupt-test register (ITR).



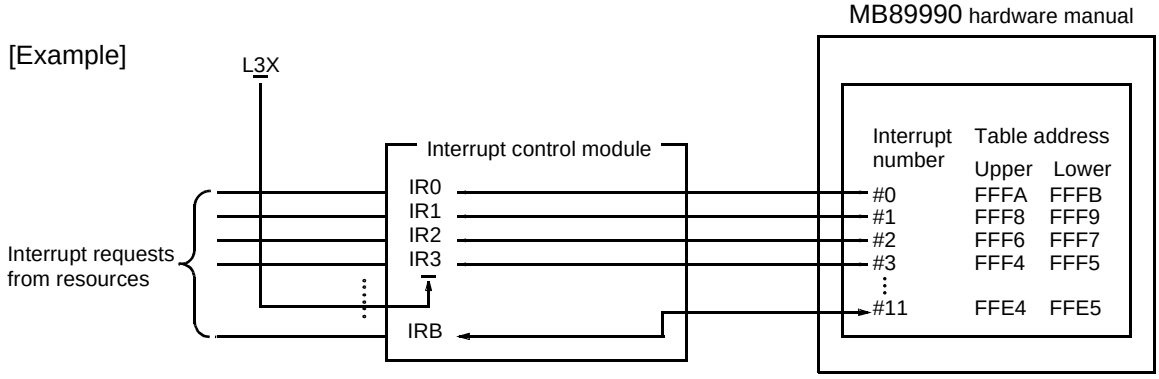
■ Description of Registers

The detail of each register is described below.

(1) Interrupt levei setting register (ILR1 to ILR3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: $007C_H$	L31	L30	L21	L20	L11	L10	L01	L00
Address: $007D_H$	L71	L70	L61	L60	L51	L50	L41	L40
Address: $007E_H$	LB1	LB0	LA1	LA0	L91	L90	L81	L80
	Intilial value 11111111							

The ILRX sets the interrupt level of each resource. The digits in the center of each bit correspond to the interrupt numbers.



When an interrupt is requested from each resource, the interrupt controller transfers the interrupt level based on the value set at the 2 bits of the ILRX corresponding to the interrupt to the CPU.

A relation between two bits of the ILRX and the interrupt level required is shown below.

Lx1	Lx0	Required interrupt level
0	X	1
1	0	2
1	1	3 (None)

(2) Interrupt test register (ITR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: $007F_H$	—	—	—	—	—	—	*	*

The ITR used for testing. Do not access it.

■ Description

The functions of interrupt controllers are described below.

- Interrupt functions
  - The MB89990 series of microcontrollers have 4 inputs for interrupt requests from each resource. The interrupt level can be set by 2-bit registers corresponding to each input. When an interrupt can be requested from a resource, the interrupt controller receives it and transfers the contents of the corresponding level register to the CPU. The interrupt to the device is processed as follows:

(a) An interrupt source is generated inside each resource.

(b) If an interrupt is enabled, an interrupt request is output from each resource to the interrupt controller by referring to the interrupt-enable bit inside each resource.

(c) After receiving this interrupt request, the interrupt controller determines the priority of simultaneously-requested interrupts and then transfers the interrupt level for the applicable interrupt to the CPU.

(d) The CPU compares the interrupt level requested from the interrupt controller with the IL bit in the processor status register.

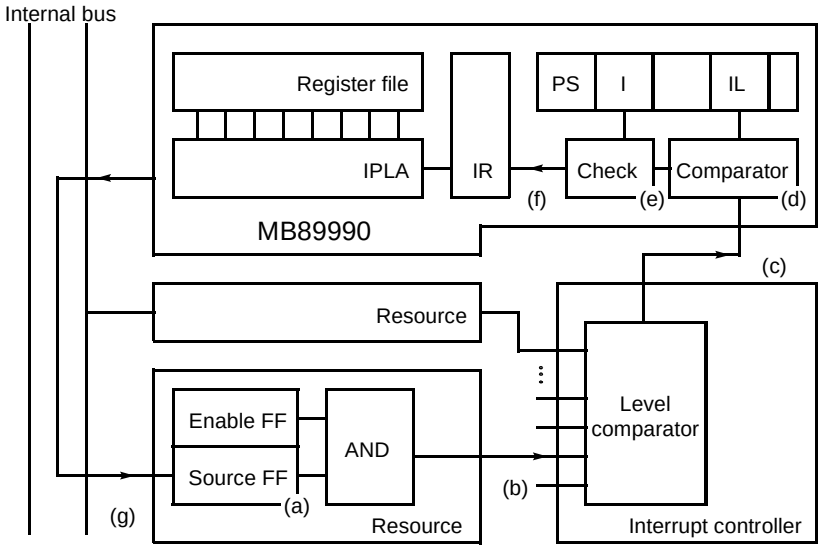
(e) As a result of the comparison, if the priority of the interrupt level is higher than that of the current interrupt processing level, the contents of the I-flag in the same processor status register are checked.

(f) As a result of the check in step (e), if the I-flag is enabled for an interrupt, the contents of the IL bit are set to the required level. As soon as the currently-executing instruction is terminated, the CPU performs the interrupt processing and transfers control to the interrupt-processing routine.

(g) When an interrupt source generated in step (a) is cleared by software in the userís interrupt processing routine, the CPU terminates the interrupt processing.

Figure 2.3-1 "Interrupt-processing Flowchart" outlines the interrupt operation for the MB89990 series of microcontrollers.

Figure 2.3-1 Interrupt-processing Flowchart



2.4 I/O Ports

- **he MB89990 series of microcontrollers have three parallel ports and 22 pins. P00 to P07 and P30 to P37 serve as 8-bit I/O ports, P40 to P45 serve as 6-bit I/O ports.**
- **Port0 and Port3 are also used as the I/O pin for the resource.**

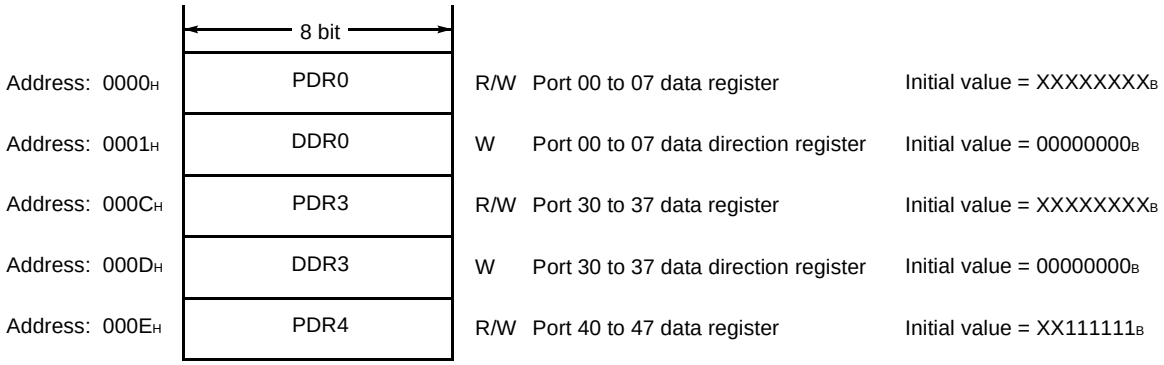
■ List of port functions

Table 2.4-1 List of Port Functions

Pin name	Input type	Output type	Function	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
P00 to P07	CMOS	CMOS push-pull	Parallel port 00 to 07	P07	P06	P05	P04	P03	P02	P01	P00
			External interrupt 2	$\overline{\text{INT27}}$	$\overline{\text{INT26}}$	$\overline{\text{INT25}}$	$\overline{\text{INT24}}$	$\overline{\text{INT23}}$	$\overline{\text{INT22}}$	$\overline{\text{INT21}}$	$\overline{\text{INT20}}$
P30 to P37	CMOS	Cmos push-pull	Parallel port 30 to 37	P37	P36	P35	P34	P3	P32	P31	P30
	Hysteresis		Timer, External interrupt 1	ROC	INT12	INT11	TO/ INT10	EC	—	—	—
P30 to P37	CMOS	N-ch open-drain	Parallel port 40 to 45	—	—	P45	P44	P43	P42	P41	P40

■ Register List

I/O port consists of the following registers.



■ Description of Functions

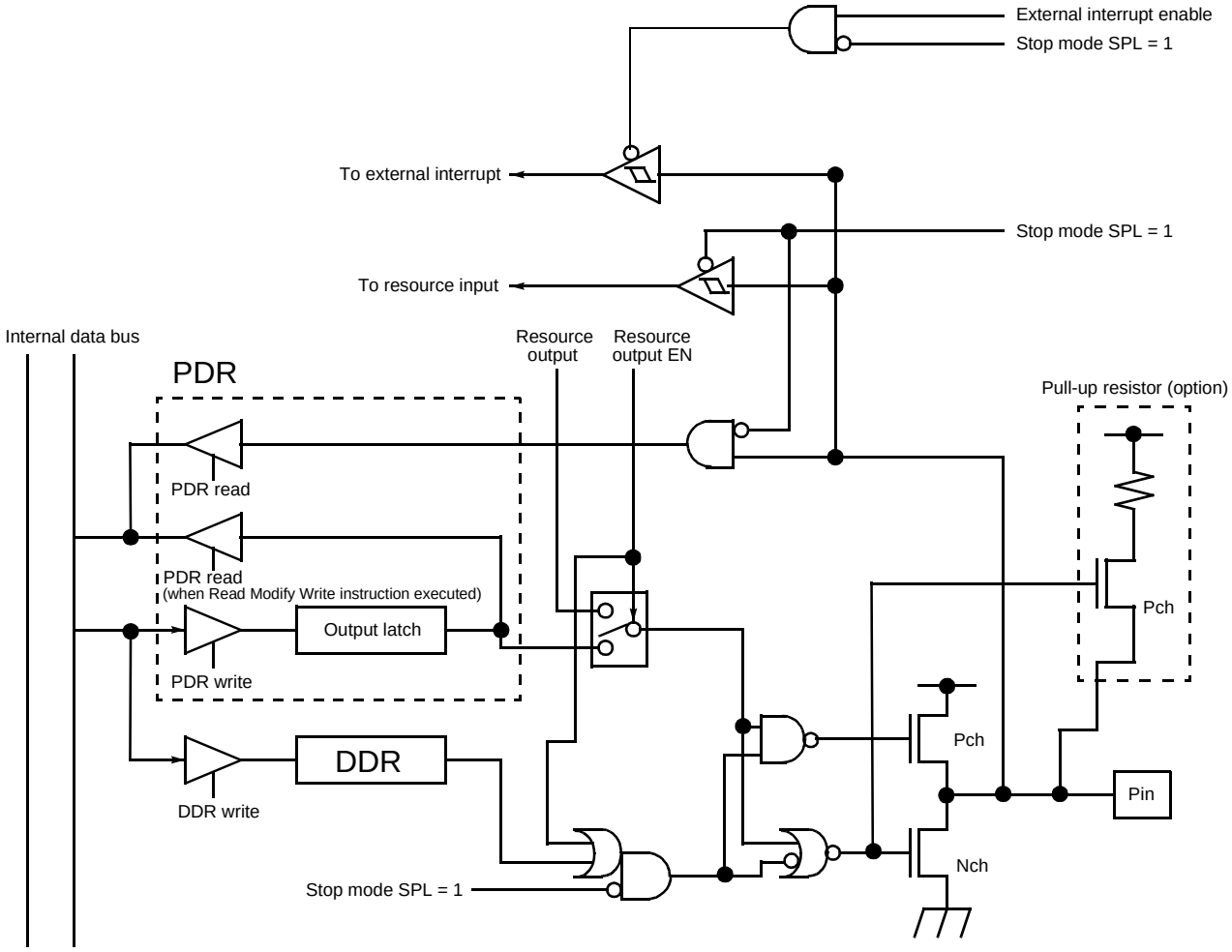
The function of each port is described below.

(1) P00 to P07: CMOS type I/O ports
(also used as resource input and output)

P30 to P37: CMOS type I/O ports
(also used as resource input and output)

- Switching input and output
  - This port has a data-direction register (DDR) and a port-data register (PDR) for each bit. Input and output can be set independently for each bit. The pin with the DDR set to 1 is set to output, and the pin with the DDR set to 0 is set to input. When the resource output bit is enabled, these ports are set to output irrespective of the DDR setting conditions.
- Operation for output port (DDR = 1)
  - The value written at the PDR is output to the pin when the DDR is set to 1. When the PDR is read, usually, the value of the pin is read instead of the contents of the output latch. However, when the Read Modify Write instruction is executed, the contents of the output latch are read irrespective of the DDR setting conditions. Therefore, the bit-processing instruction can be used even if input and output are mixed with each other. When data is written to the PDR, the written data is held in the output latch irrespective of the DDR setting conditions.
- Operation for input port (DDR = 0)
  - When used as the input port, the output impedance goes High. Therefore, when the PDR is read, the value of the pin is read.
- Resource output operation
  - When using as the resource output, setting is performed by the resource output enable bit. (See the description of each resource.) Since the resource output enable bit has priority in switching input and output, even if the DDR is set to 0, any bit is set as the resource output when output is enabled at each resource. Even if the output from each resource is enabled, the read parallel port is effective, so the resource output value can be checked.
- Resource input operation
  - The pin value at a port with the resource input function is always input for the resource input (irrespective of the setting of the DDR and resource). Set the DDR to input when using an external signal for the resource input.
- State when reset
  - When reset, the DDR and the output enable bit for each resource are initialized to 0 and the output impedance goes High at all bits. When reset, the PDR is not defined. Therefore, set the value of the PDR before setting the DDR to output.
- State when stop
  - With the SPL bit of the standby-control register set to 1, in the stop mode, the output impedance goes High irrespective of the value of the DDR.

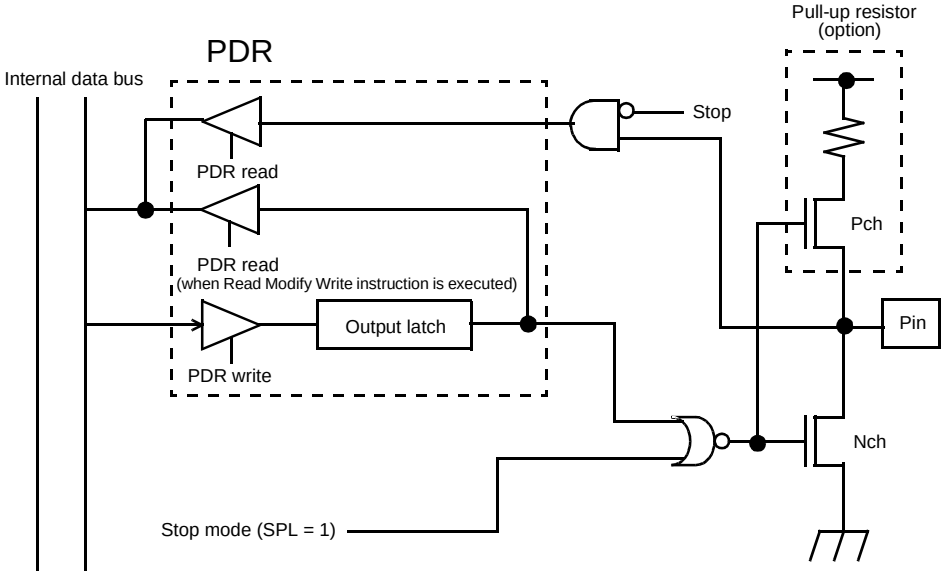
Figure 2.4-1 Ports 00 to 07 and 30 to 37



(2) P40 to P45: N-ch open-drain-type output ports
(also used as analog input)

- Operation for output port
  - The value written at the PDR is output to the pin. When the PDR is read in this port, the contents of the output latch is always read instead of the value of the pin.
- State when reset
  - The PDR is initialized to 1 at reset, so the output register is turned off at all bits.
- State in stop mode
  - When the SPL bit of the standby-control register is set to 1, in the stop mode, the output impedance goes High irrespective of the value of the PDR.

Figure 2.4-2 Ports 40 to 45

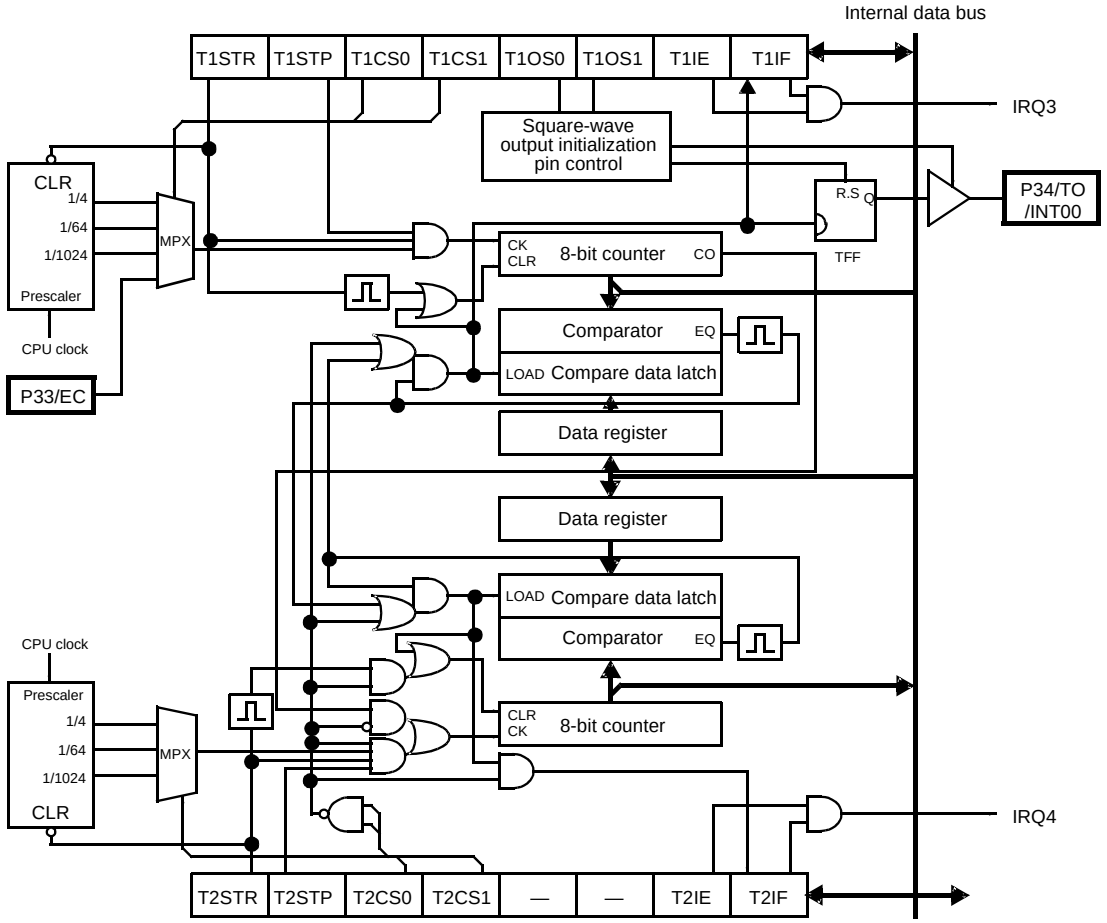


2.5 8/16-bit Timer (Timer 1 and Timer 2)

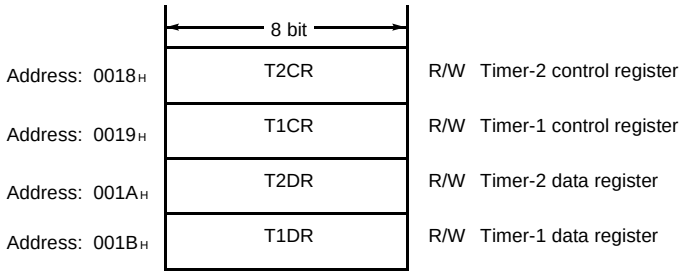
- **Three internal clock pulses and one external clock pulse can be selected.**
- **Operation in 8-bit 2-ch mode or 16-bit 1-ch mode can be selected.**
- **A square-wave output function is included.**

■ Block Diagram

Figure 2.5-1 8/16-bit Timer Block Diagram



■ Register List



■ Description of Register Details

The detail of each register is described below.

(1) Timer 1 control register (T1CR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0019_H	T1IF	T1IE	T1OS1	T1OS0	T1CS1	T1CS0	T1STP	T1STR
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
				Intilial value $X00000X0_B$				

[Bit 7] T1IF: Interrupt request flag

(When write)

0	Interrupt request flag clearing
1	No operation

(When read)

0	No interrupt request
1	Interval interrupt request

1 is always read when the Read Modify Write instruction is executed.

[Bit 6] T1IE: Interrupt-enable bit

0	Interrupt disabled
1	Interrupt enabled

[Bit 5 and 4] T1OS1, T1OS0: Square-wave output control bit

T1OS1	**T1OS0**	
0	0	Makes square-wave output port (P43) general-purpose port
0	1	Holds data setting square-wave output to Low level
1	0	Holds data setting square-wave output to High level
1	1	Sets square-wave output to held value

When the T1STR bit is 0, the square-wave output is set to the set value.

[Bit 3 and 2] T1CS1, T1CS0: Clock source select bit

T1CS1	T1CS0	Clock cycle time selected at 4 MHz	Clock cycle time
0	0	2.0 [μs]	× 2 instruction cycle
0	1	32.0 [μs]	× 32 instruction cycle
1	0	512 [μs]	× 512 instruction cycle
1	1	External clock	

Note:

When using Timer 1 in the 8-bit mode, the clock source selection bits (T1CS1 and T1CS0) of the Timer 2 control register (T2CR) must be set to other than the 16-bit mode.

[Bit 1] T1STP: Timer-stop bit

0	Counting continued without clearing counter
1	Counting suspended

[Bit 0] T1STR: Timer-start bit)

0	Terminates operation
1	Clears counter and starts operation

(2) Timer 2 control register (T2CR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0018_H	T21F	T21E	—	—	T2CS1	T2CS0	T2STP	T2STR
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
				Intilial value $X00000X0_B$				

[Bit 7] T2IF: Interrupt request flag bit

(When write)

0	Interrupt request flag clearing
1	No operation

(When read)

0	No interrupt request
1	Interval interrupt request

1 is always read when the Read Modify Write instruction is executed.

[Bit 6] T2IE: Interrupt-enable bit

0	Interrupt disabled
1	Interrupt enabled

[Bit 3 and 2]: T2CS1, T2CS0: Clock source select bit

T2CS1	T2CS0	Clock cycle time selected at 4 MHz	Clock cycle time
0	0	2.0 [μs]	× 2 instruction cycle
0	1	32.0 [μs]	× 32 instruction cycle
1	0	512 [μs]	× 512 instruction cycle
1	1	16 bit mode	

[Bit 1] T2STP: Timer stop bit

0	Operation continued without clearing counter
1	Count operation suspended

[Bit 0] T2STR: Timer start bit

0	Operation stopped
1	Operation started after clearing counter

(3) Timer 1 and 2 data registers (T1DR and T2DR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: $001B_H$								
Address: $001A_H$								
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
				Intilial value $XXXXXXXX_B$				

Write data is the set interval times and read data is the counted times.

■ Description of Operation

(1) 8-bit internal clock mode

In the 8-bit internal clock mode, three internal clock inputs can be selected by setting the clock source select bits (T1CS1 and T1CS0, T2CS1 and T2CS0) of the timer control registers (T1CR and T2CR). The timer data registers (T1DR and T2DR) serve as interval time setting registers.

To start the timer, set the interval time as the timer data registers, write 1 at the timer start bits (T1STR and T2STR) of the timer control registers to clear the counter to 00_H, and load the values of the timer data registers into the compare latch. Then, counting starts.

When the values of the counter agree with those of the timer data registers, the interval interrupt request flags (T1IF and T2IF) are set to 1. At this time, the counter is cleared to 00_H, the values of the timer data registers are reloaded into the compare latch, and counting is continued. If the interrupt enable bits (T1IE and T2IE) are set to 1, an interrupt request is output to the CPU. Assuming the set value of the timer data register is n and the selected clock is ϕ, the interval time (T) can be calculated as follows.

$T = \phi \times (n + 1)$ [μs]

Figure 2.5-2 Description Diagram for Internal Clock Mode Operation

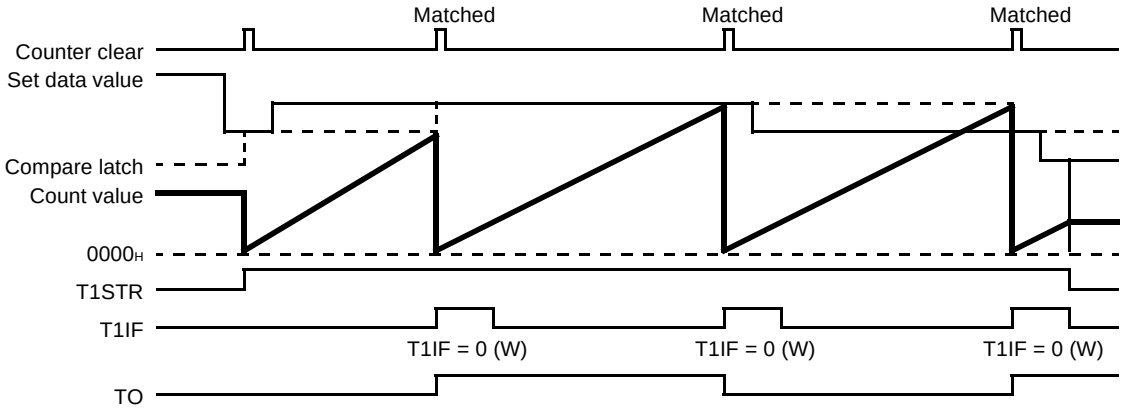
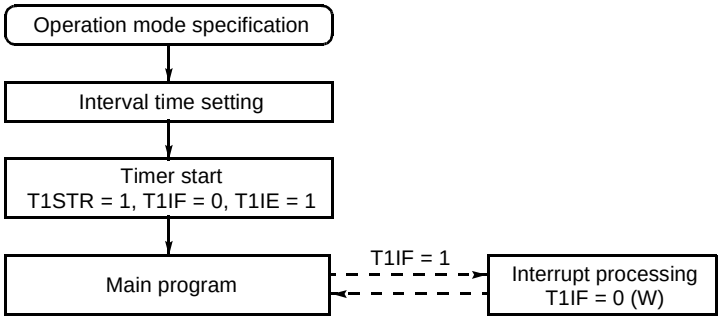


Figure 2.5-3 Flow Diagram for Timer Setting



(2) Initializing square-wave output

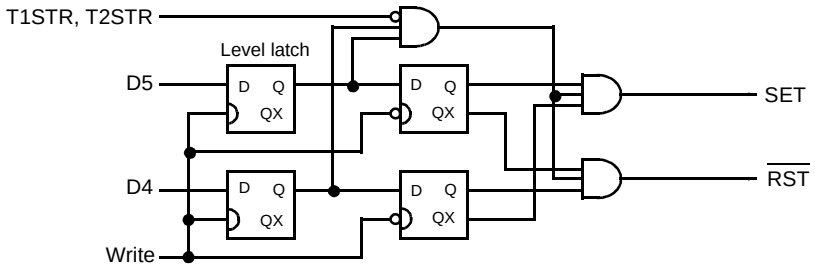
The square-wave output can be set to any value only when the timer stops (T1STR = 0 and T2STR = 0).

To set, proceed as follows:

(a) Write the set values (01 and 10) at the initialize bits (T1OS1 and T1OS0, T2OS1 and T2OS2, respectively) of the square wave output.

(b) Write 11 at the same bits. This initializes the square wave output to the set value. If the T1STR bit is set to 0, the square wave output of the pin is set to the set value during this write cycle.

Figure 2.5-4 Initialization of Equivalent Circuit



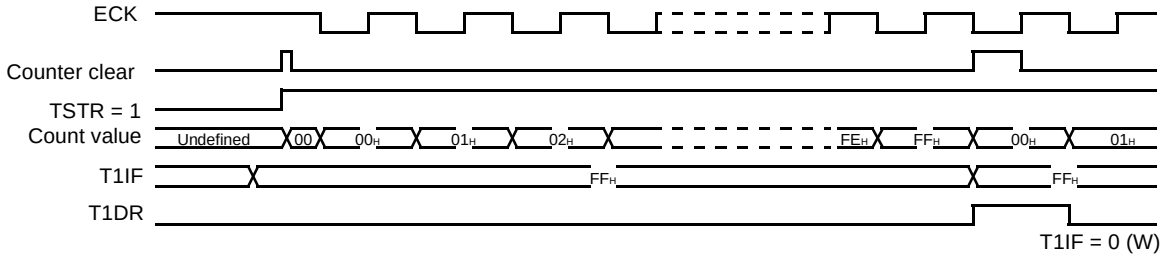
(3) 8-bit external clock mode

In the 8-bit external clock mode, the external clock input can be selected by setting the clock source select bits (T1CS1 and T1CS0) of the timer 1 control register (T1CR).

To start the timer, write 1 at the timer start bit (T1STR) of the T1CR to clear the counter. Then, counting starts.

When the value of the counter agrees with that of the timer data register setting, the interval interrupt request flag bit (T1IF) is set to 1. At this time, if an interrupt is enabled (T1IE = 1), an interrupt request is output to the CPU.

Figure 2.5-5 External Cock Mode Operation Description Diagram

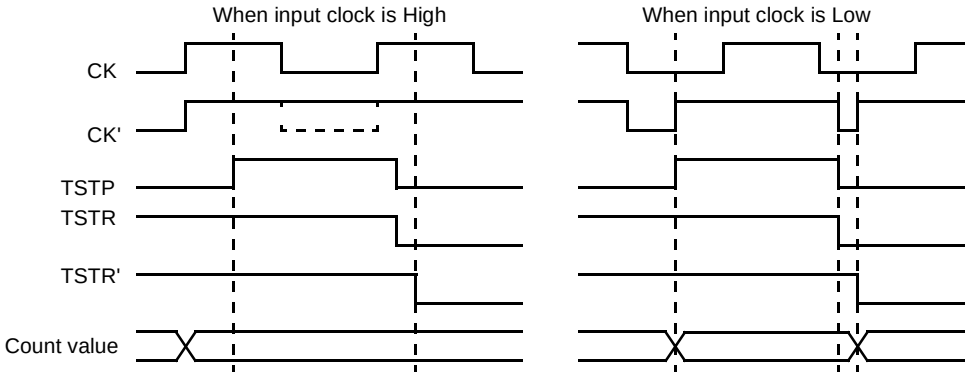


(4) Precautions for use of timer stop bit

Since an input clock pulse is fixed to High level when the timer is stopped by the timer start bits, the count value differs depending on the state of the input clock pulse.

When writing 00 at the timer stop and timer start bits simultaneously after stopping the timer with the timer stop bit, the count may be incremented by 1. Therefore, if the timer is stopped by the timer stop bit, read the counter and then write 00 at the timer start bits (See Figure 2.5-6 "Operation Diagram when Timer Stop Bit is Used".).

Figure 2.5-6 Operation Diagram when Timer Stop Bit is Used



(5) 16-bit mode

In the 16-bit mode, each bit of the timer control registers is as shown below.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0019_H	T1IF	T1IE	T1OS1	T1OS0	T1CS1	T1CS0	T1STP	T1STR
Address: 0018_H	T2IF	T2IF	—	—	TECS1	TECS0	T2STP	T2STR
	No operation		Set to 00		Set to 11		No operation	

In the 16-bit mode, write 11 at the T2CS1 and T2CS0 bits of the T2CR and set 00 at the T2OS1 and T2OS0 bits.

When in the 16-bit mode, the timer is controlled by the T1CR. The timer data registers T2DR and T1DR use the upper and lower bytes, respectively.

The clock source is selected by the T1CS1 and T1CS0 bits of the T1CR. To start the timer, write 1 at the T1STR bit of the T1CR to clear the counter.

If the value of the counter agrees with that of the timer data register, the T1IF bit is set to 1. At this time, an interrupt request is output to the CPU if the T1IE bit is 1.

Note:

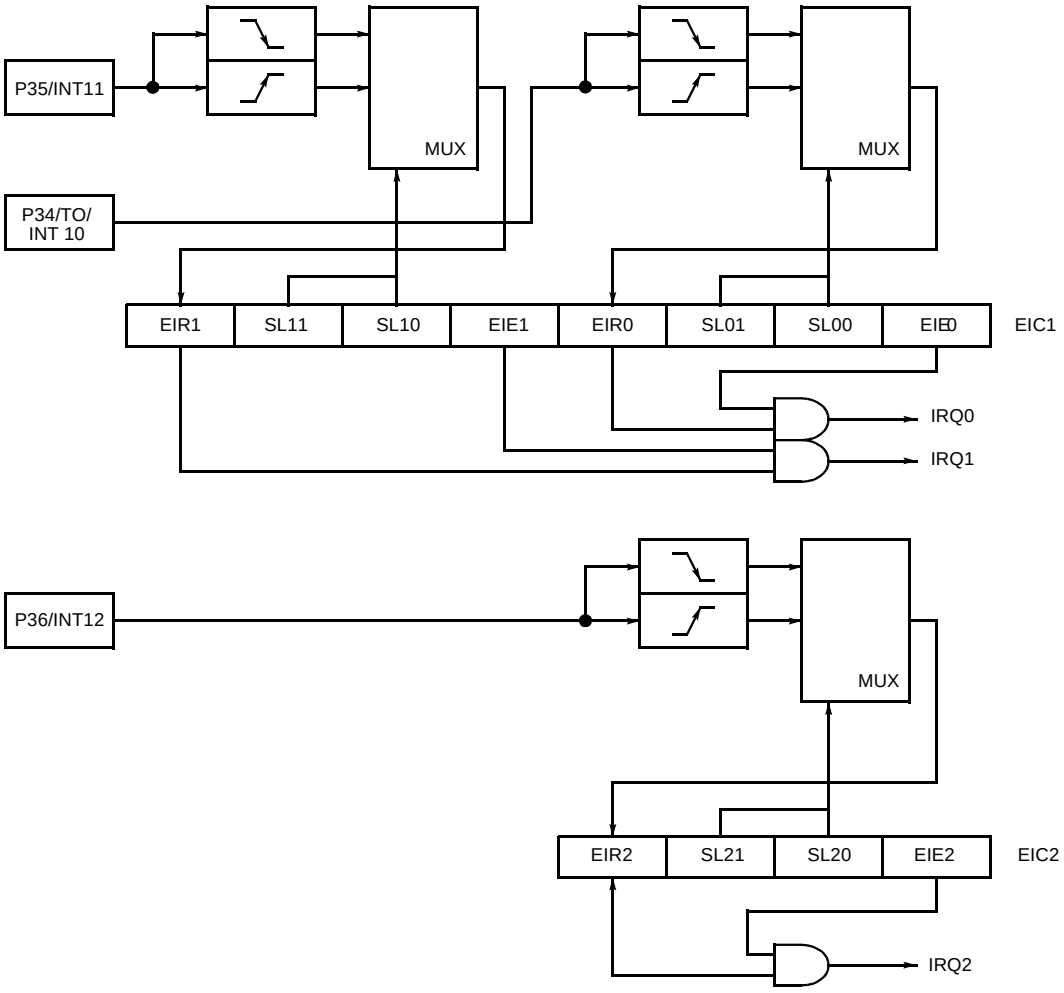
To read the value of the counter in the 16-bit mode, always read the value twice to check that it is valid, and then use the data.

See the 8-bit operation diagram for 16-bit mode operation.

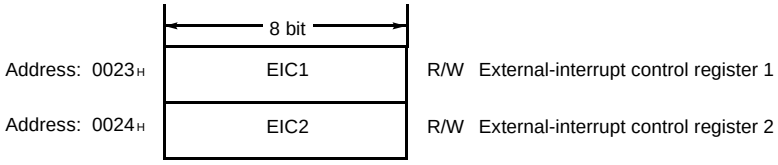
2.6 External Interrupt 1

- **The edges of three external-interrupt sources (INT10 to INT12) can be detected to set the corresponding flag.**
- **An interrupt can be generated at the same time the flag is set.**
- **The three interrupts can release the STOP or SLEEP mode.**

■ Block diagram



■ Registers



■ Description of Registers

(1) External-interrupt control register 1 (EIC1)

The EIC1 controls interrupts by the INT10 and INT11 pins.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0023_H	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
				Intilial value 00000000_B				

[Bit 7] EIR1: External-interrupt request flag

When the edge specified by the SL11 and SL10 bits is input to the INT11 pin, bit 7 is set to 1. When the EIE1 bit is 1, an interrupt request (IRQ1) is output if this bit is set.

The meaning of each bit to be read is as follows:

0	Specified edge not input to INT11 pin
1	Specified edge input to INT11 pin (IRQ1 is output.)

1 is always read when the Read Modify Write instruction is read.

The meaning of each bit to be written is as follows:

0	This bit is cleared.
1	This bit does not change nor affect other bits.

[Bit 6 and 5] SL11, SL10: Edge-polarity select bit

This bit is used to control the input edge polarity of the INT11 pin.

SL11	**SL10**	
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both-edge mode

[Bit 4] EIE1: Interrupt-enable bit

This bit is used to enable an external-interrupt request by the INT11 pin.

0	Interrupt request disabled
1	Interrupt request enabled by EIR1 setting

[Bit 3] EIR0: External-interrupt request flag

When the edge specified by the SL01 and SL00 bits is input to the INT10 pin, bit 3 is set to 1. When the EIE0 is 1, an interrupt request (IRQ0) is output if this bit is set.

The meaning of each bit to be read is as follows:

0	Specified edge not input to INT10 pin
1	Specified edge input to INT10 pin (IRQ0 is output.)

1 is always read when the Read Modify Write instruction is read.

The meaning of each bit to be written is as follows:

0	This bit is cleared.
1	This bit does not change nor affect other bits.

[Bit 2 and 1] SL01, SL00: Edge-polarity select bit

This bit is used to control the input edge polarity of the INT10 pin.

SL01	**SL00**	
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both-edge mode

[Bit 0] EIE0: Interrupt-enable bit

Bit 0 is used to enable an external-interrupt request by the INT10 pin.

0	Interrupt request disabled
1	Interrupt request enabled by EIR0 setting

(2) External-interrupt control register 2 (EIC2)

The EIC2 controls an interrupt by the INT12 pins.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0024_H	—	—	—	—	EIR2	SL21	SL20	EIE2
					(R/W)	(R/W)	(R/W)	(R/W)
					Intilial value			
					----0000_B			

[Bit 3] EIR2: External-interrupt request flag

When the edge specified by the SL21 and SL20 bit is input to the INT12 pin, bit 3 is set to 1. When the EIE2 bit is 1, an interrupt request (IRQ2) is output if this bit is set.

The meaning of each bit to be read is as follows:

0	Specified edge not input to INT12 pin
1	Specified edge input to INT12 pin (IRQ2 is output.)

1 is always read when the Read Modify Write instruction is read.

The meaning of each bit to be written is as follows:

0	This bit is cleared.
1	This bit does not change nor affect other bits.

[Bit 2 and 1] SL21, SL20: Edge-polarity select bit

This bit is used to control the input edge polarity of the INT12 pin.

SL21	SL20	
0	0	No edge detection
0	1	Rising edge
1	0	Falling edge
1	1	Both-edge mode

[Bit 0] EIE2: Interrupt-enable bit

This bit is used to enable an external-interrupt request by the INT12 pin.

0	Interrupt request disabled
1	Interrupt request enabled by setting of EIR2

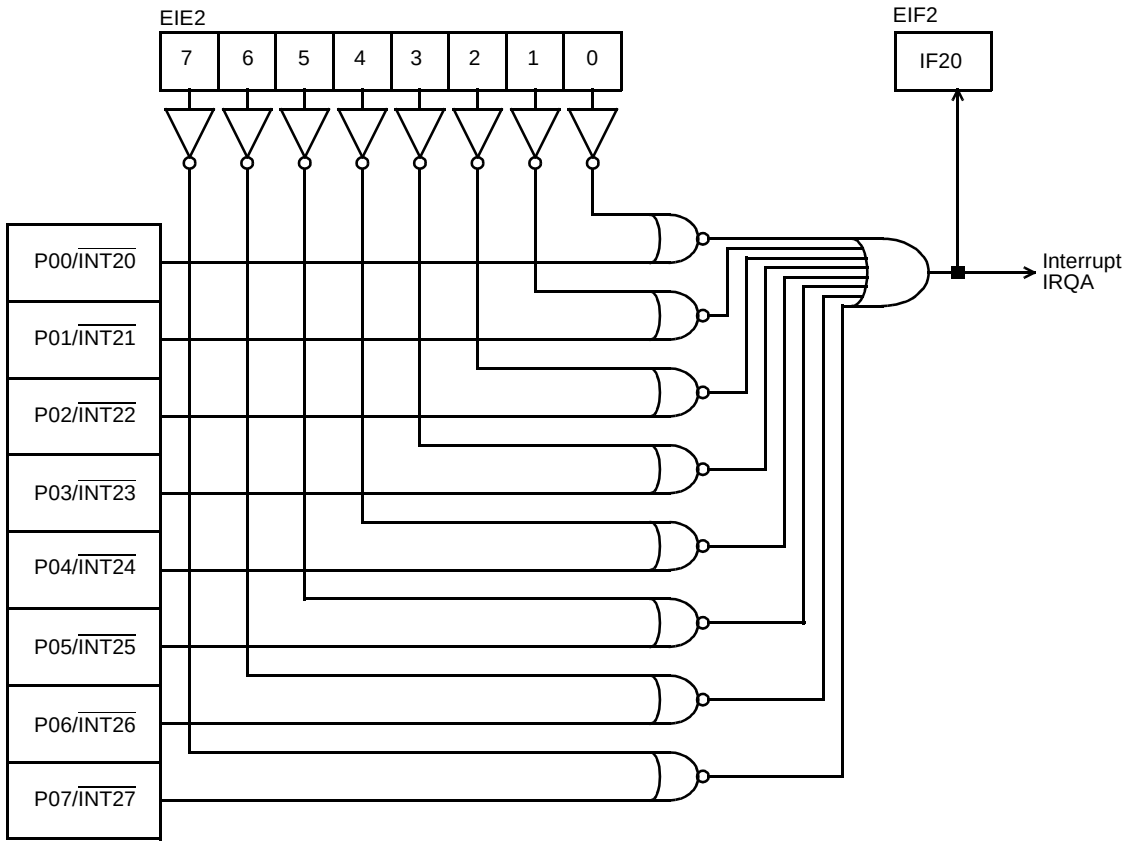
■ Precautions for External-interrupt Circuit

- When enabling an interrupt after clearing reset, always clear the interrupt flag simultaneously. An interrupt request is output immediately when the interrupt flags (EIR2, EIR1, EIR0) are set to 1.
- When no edge detection is specified by the edge-polarity select bit, the current input is held before the internal edge detection block. If an edge is specified in this state, edge detection may be erroneous . Therefore, always clear the flag after an edge is specified.

2.7 External Interrupt 2 (Wake Up)

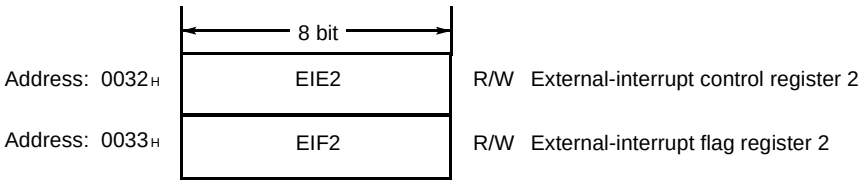
- **Eight external interrupt input pins**
- **An interrupt request is output by Low-level input signals.**
- **Also usable as wake-up input**

■ Block Diagram



■ Register List

This external interrupt 2 consists of external interrupt 2 control register (EIE2) and external interrupt 2 flag register (EIF2).



■ Description of Registers

The detail of each register is described below.

(1) External interrupt 2 control register (EIE2)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0032_H	IE27	IE26	IE25	IE24	IE23	IE22	IE21	IE20
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
				Intilial value 00000000_B				

[Bit 7 to 0] IE27 to IE20: Operation-enable bit

These bits are used to operation-enable external interrupt of $\overline{INT27}$ to $\overline{INT20}$.

0	External interrupt operation-disabled
1	External interrupt operation-enabled

(2) External interrupt 2 control register (EIF2)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0033_H	—	—	—	—	—	—	—	IF20
								(R/W)
				Intilial value $-------0_B$				

[Bit 0] IF20: Low-level detect flag bit

This bit is used to detect LOW level of $\overline{INT27}$ to $\overline{INT20}$.

(When write)

0	Clears flag for detecting LOW level
1	No operation

(When read)

0	No LOW level input
1	LOW level input detected

If any of the interrupt enable bits (IE27 to IE20) of the external interrupt 2 control register (EIE2) is 1, the Low-level detect flag bit (IF20) is set to 1 and an interrupt request is output to the CPU when a Low level is input to the port corresponding to this bit.

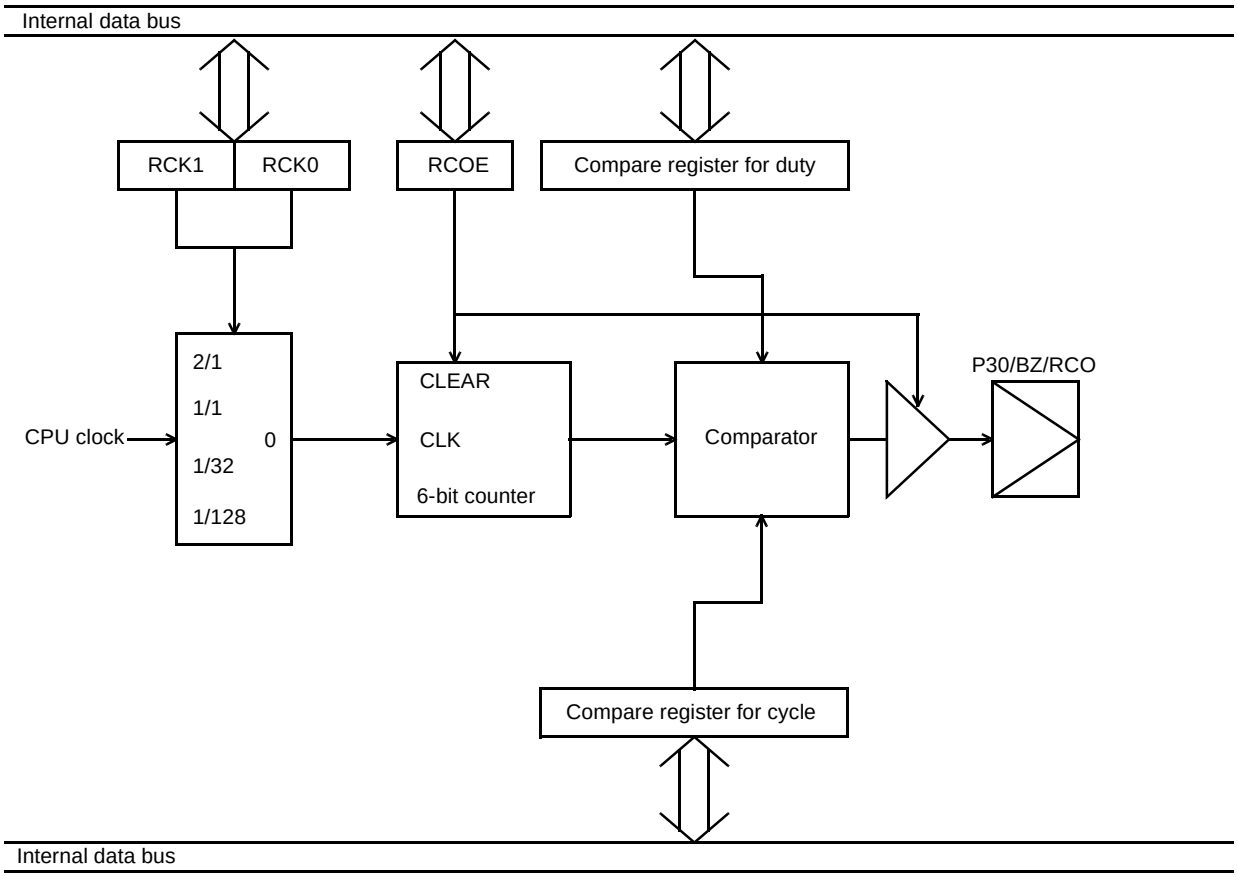
Note:

Unlike other resources, even if the external interrupt 2 circuit is disabled for an interrupt, it keeps generating interrupts until the interrupt source is cleared. Therefore, always clear the interrupt source (after disabling an interrupt).

2.8 Remote-control Carrier Frequency Generator

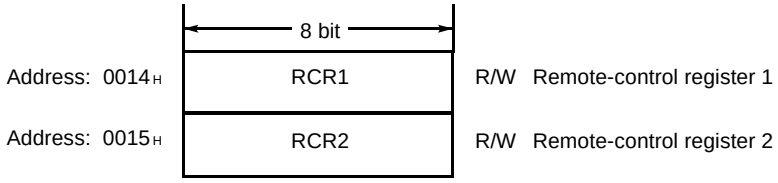
- **This generator is a remote-control circuit for generating remote-control carrier frequencies.**
- **The 6-bit binary counter is built in.**
- **Four internal clock pulses can be selected to set a duty (H width) and cycle.**

■ Block Diagram



CPU clock: Halved from source clock

■ Register List



■ Description of Registers

(1) Remote-control register 1 (RCR1)

This register is used to select the reference clock and set the duty of remote-control carrier frequency.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0014_H	RCK1	RCK0	HSC5	HSC4	HSC3	HSC2	HSC1	HSC0
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
	Intilial value 00000000_B							

[Bits 7 and 6] RCK1 and RCK0: Bits for selecting the reference clock for remote-control carrier frequency

These bits are used to select the reference clock for the remote-control carrier frequency.

RCR1	RCR0	Reference clock at 4 MHz
0	0	2/f (0.5 μs)
0	1	4/f (1.0 μs)
1	0	32/f (8.0 μs)
1	1	128/f (32.0 μs)

f = source clock frequency

[Bits 5 to 0] HSC5 to HSC0: Bits for setting duty of remote-control carrier frequency

These bits are used for the 6-bit compare register to set the duty of the remote-control carrier frequency.

To set the duty of the remote-control carrier frequency, set the value subtracted 1 from the value calculated from the clock in binary at these bits. For example, to set a duty of 26 ms, select resource clock = 4/f and set 011001 (1/26 oscillation) at these 6 bits. This enables the selection of any duty.

(2) Remote-control register 2 (RCR2)

This register is used to enable the output and set the cycle of remote-control carrier frequency.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0015_H	RCEN	—	SCL5	SCL4	SCL3	SCL2	SCL1	SCL0
	(R/W)		(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
	Intilial value 00000000_B							

[Bit 7] RCEN: Bit for enabling output of remote-control carrier frequency

This bit is used to enable the output of remote-control carrier frequency to the P37/BZ/RCO pin. Setting this bit to 0 enables clearing of the 6-bit counter.

[Bits 5 to 0] SCL5 to SCL0: Bits for setting cycle of remote-control carrier frequency

These bits are used for the 6-bit compare register to set the cycle of the remote-control carrier frequency. To set the cycle of the remote-control carrier frequency, set the value subtracted 1 from the value calculated from the clock source in binary at these bits. For example, to set a cycle of 60 ms, select reference clock = 4/f and set 111011 (1/60 oscillation) at these 6 bits. This enables selection of a cycle of 60 µs.

■ Description of Operation

Remote-control registers 1 and 2 (RCR1 and RCR2) control a 6-bit counter to output the remote-control carrier frequency to the P37/BZ/RCO pin.

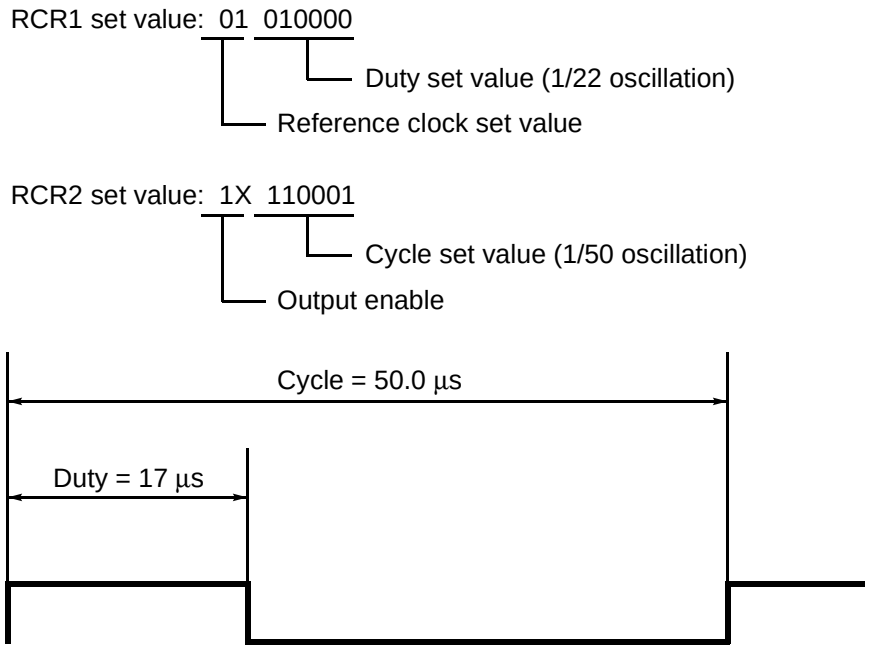
A usage example is given below.

<Example>

Cycle: about 20 kHz

Duty: 1/3

Reference clock: 4/f (f = source clock)



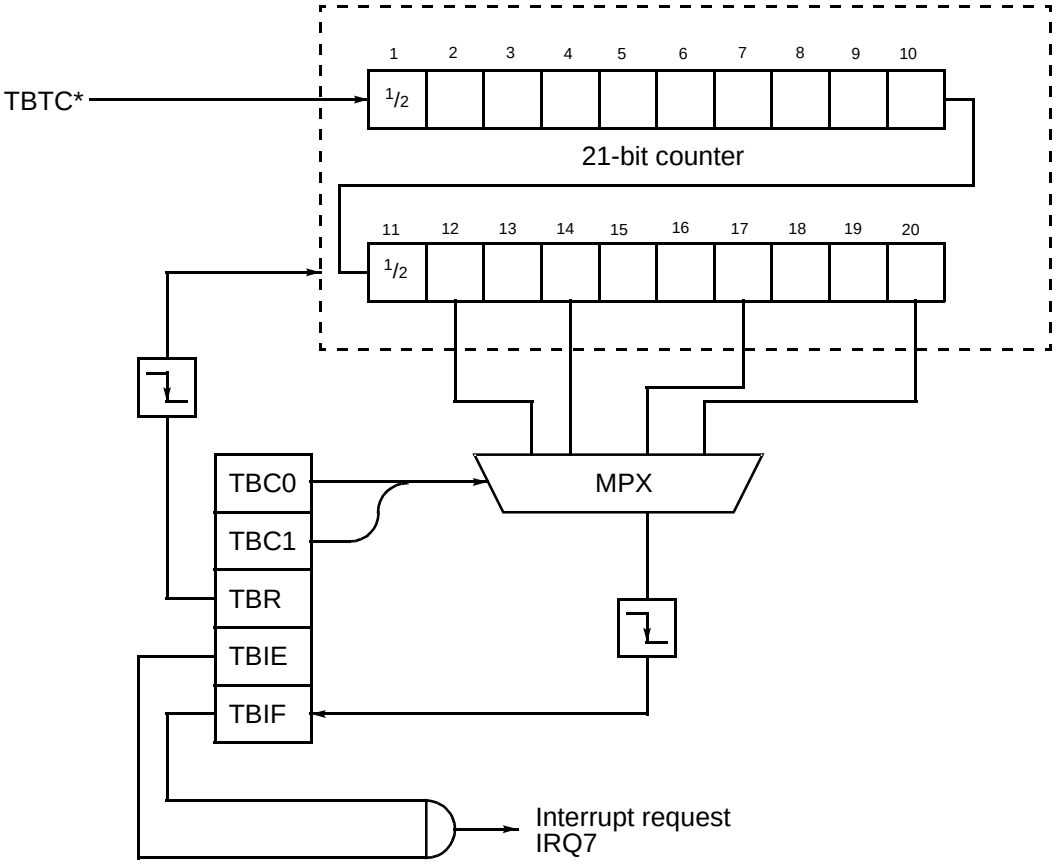
Note:

To set the duty and cycle, the cycle set value must always be greater than the set duty value.

2.9 Time-base Timer

- **This timer has a 20-bit binary counter and uses a clock pulse with 1/2 oscillation of the source clock.**
- **Four interval times can be selected.**
- **This function cannot be used in the STOP state.**

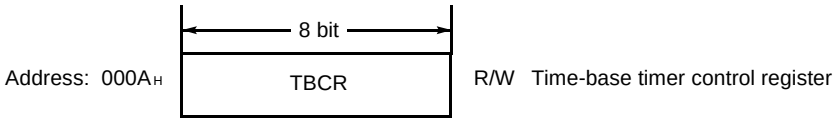
■ Block Diagram



*TBTC is a clock pulse with 1/2 oscillation of the souce clock.

■ Register List

The time-base timer has time-base timer control register (TBCR).



■ Description of Registers

The detail of time-base timer control register (TBCR) is described below.

(1) Timer-base timer control register (TBCR)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: $000A_H$	—	—	—	TBIE	TBOF	TBR	TBC1	TBC0
				(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
				Intilial value $XXX00000_B$				

[Bit 4] TBIE: Interval-timer interrupt enable bit

This bit is used to enable an interrupt by the interval timer.

0	Interval interrupt disabled
1	Interval interrupt enabled

[Bit 3] TBOF: Interval timer overflow bit

When writing, this bit is used to clear the interval timer overflow flag.

0	Interval timer overflow flag cleared
1	No operation

When reading, this bit indicates that an interval timer overflow has occurred.

0	Interval timer overflow not occurred
1	Interval timer overflow occurred

1 is read when the Read Modify Write instruction is read. If the TBIF bit is set to 1 when the TBIE bit is 1, an interrupt request is output. This bit is cleared upon reset.

[Bit 2] TBR: Time-base timer clear bit

This bit is used to clear time-base timer.

0	Time-base timer cleared
1	No operation

1 is always read when this bit is read.

[Bit 1 and 0] TBC1, TBC0: Interval time specification bit

These bits are used to specify interval timer cycle.

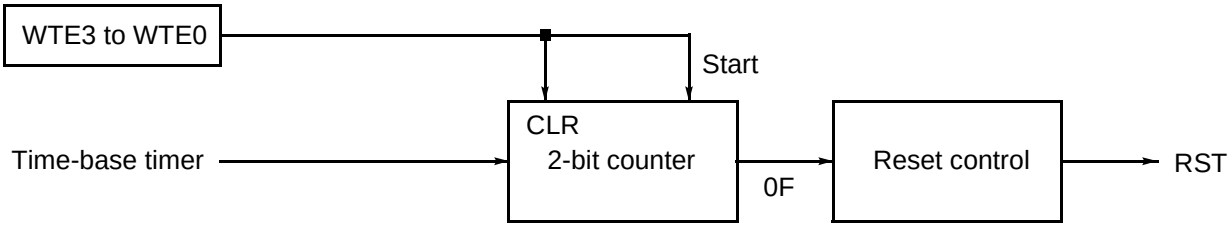
TBC1	TBC0	Interval time	Value at f = 4 MHz
0	0	$2^{13}/f$	2.05 [ms]
0	1	$2^{15}/f$	8.19 [ms]
1	0	$2^{18}/f$	65.54 [ms]
1	1	$2^{21}/f$	524.29 [ms]

f = clock frequency

2.10 Watchdog Timer Reset

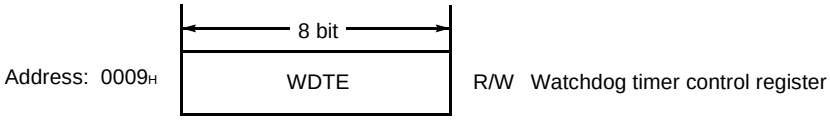
- **The watchdog timer is reset by using the time-base timer output as a clock.**

■ Block Diagram



■ Registers

The watchdog timer reset has watchdog timer control register (WDTE).



■ Description of Register

The detail of the watchdog timer control register (WDTE) is described below.

(1) Watchdog timer control register (WDTE)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address: 0009_H	—	—	—	—	WTE3	WTE2	WTE1	WTE0
				(W)	(W)	(W)	(W)	(W)
				Intilial value $XXXXXXXX_B$				

[Bits 3 to 0] WTE3 to WTE0: Watchdog timer control bit

These bits are used to control the watchdog timer.

First write after reset

0101	Watchdog timer started
Other than the above	No operation

Second and later write

0101	Watchdog timer counter cleared
Other than the above	No operation

The watchdog timer can be stopped only by reset. 1111 is read when these bit are read.

■ Description of Operation

(1) Starting watchdog timer

The watchdog timer starts when 0101 is written at the watchdog timer control bits.

(2) Clearing watchdog timer

When 0101 is written at the watchdog timer control bits after start, the watchdog timer is cleared. The counter of the watchdog timer is cleared when changing to the standby mode (STOP, SLEEP).

(3) Watchdog timer reset

If the watchdog timer is not cleared within the time given in the table below, a watchdog timer reset occurs to reset the chip internally.

	Time-base timer cycle $2^{21}/f$
Minimum time	Approx. 524 ms
Maximum time	Approx. 1049 ms

f : 4MHz

(4) Stopping watchdog timer

Once started, the watchdog timer will not stop until a reset occurs.

CHAPTER 3 OPERATION

The operation of MB89990 is described below.

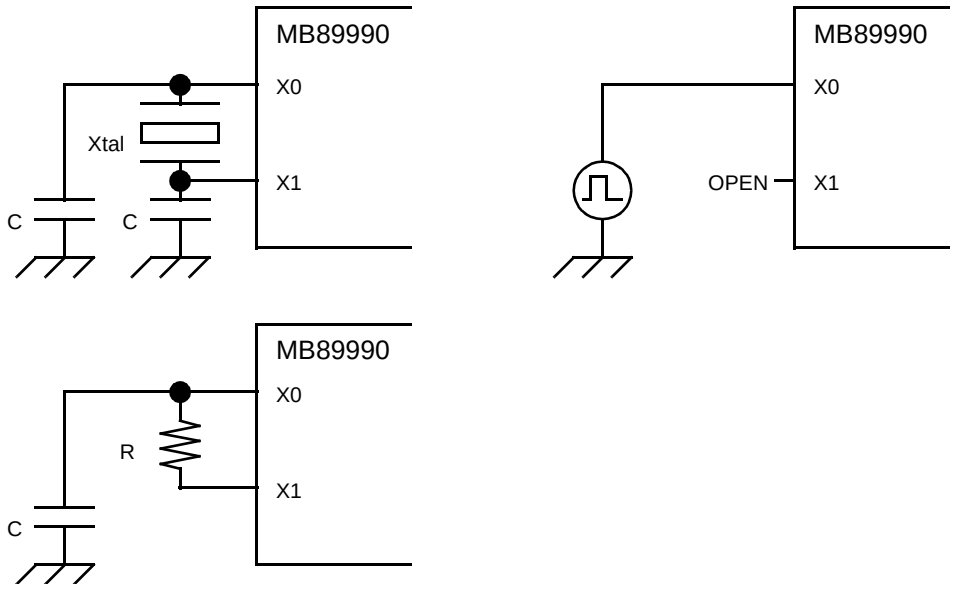
3.1 Clock Pulse Generator

This section describes the clock pulse generator.

■ Clock Pulse Generator

The MB89990 series of microcontrollers incorporate the system clock pulse generator. The ceramic or crystal oscillator, or CR is connected to the X0 and X1 pins to generate clock pulses. Clock pulses can also be supplied internally by inputting externally-generated clock pulses to the X0 pin. The X1 pin should be kept open.

Figure 3.1-1 Clock Pulse Generator



3.2 Reset

This section describes reset.

■ Reset

The detail of reset operation and reset sources are described below.

3.2.1 Reset Operation

The reset operation is described below.

■ Reset Operation

When reset conditions occur, the MB89990 series of microcontrollers suspend the currently-executing instruction to enter the reset state. The contents written at the RAM do not change before and after reset. However, if a reset occurs during writing of 16-bit long data, data is written to the upper bytes and may not be written to lower bytes. If a reset occurs around write timing, the contents of the addresses being written are not assured.

When the reset conditions are cleared, the MB89990 series of microcontrollers are released from the reset state and start operation after fetching the mode data from address $FFFD_H$, the upper bytes of the reset vectors from address $FFFE_H$, and the lower bytes from address $FFFF_H$, in that order. Figure 3.2-1 "Outline of Reset Operation" shows the flowchart for the reset operation.

Figure 3.2-1 Outline of Reset Operation

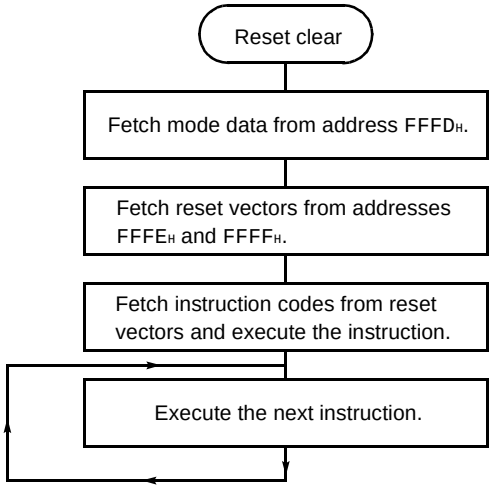
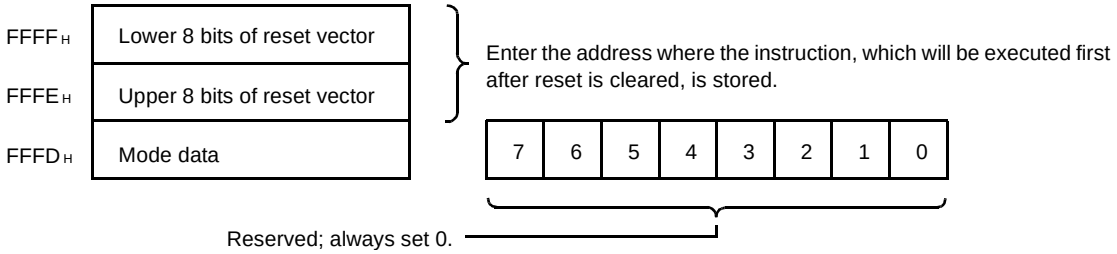


Figure 3.2-2 "Reset Vector Structure" indicates the structure of data to be stored in addresses FFFDH, FFFEH, and FFFFH.

Figure 3.2-2 Reset Vector Structure



3.2.2 Reset Source

The reset sources are described below.

■ **Reset Source**

The MB89990 series of microcontrollers have the following reset source.

(1) External pin

A Low level is input to the RST pin.

(2) Specification by software

0 is written at the $\overline{\text{RST}}$ bit of the standby-control register.

(3) Power-on

When the power is turned on when the power-on reset option is selected.

(4) Watchdog function

The watchdog function is enabled by the watchdog-control register and reaccess to this register is not obtained within the specified time.

When the stop mode is cleared or when the power-on reset (option selected) is operated, is started after elapse of the oscillation stabilization time.

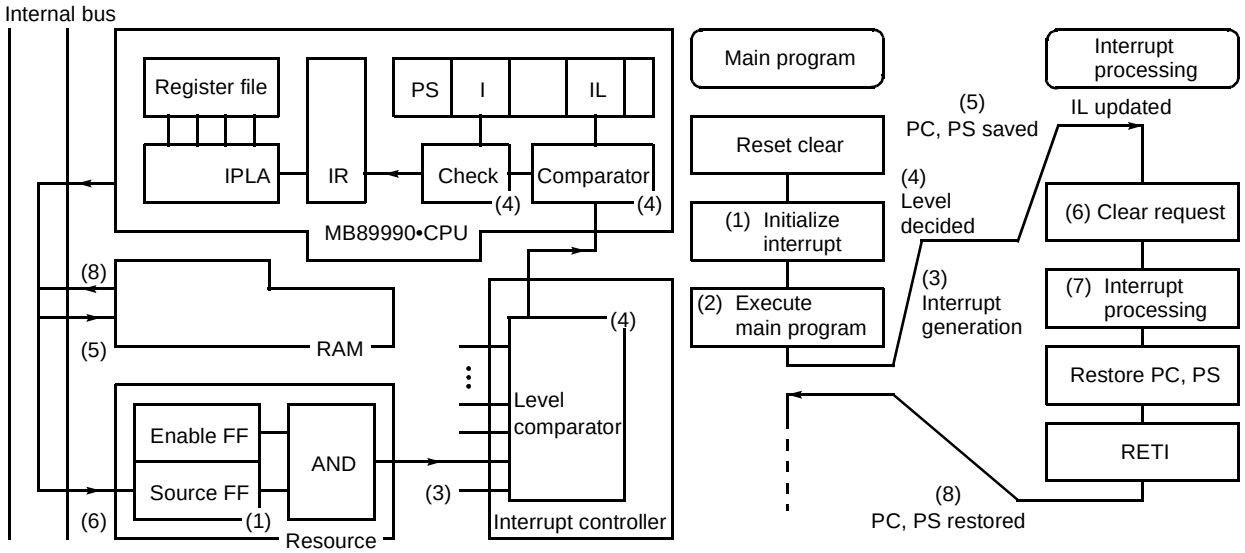
3.3 Interrupt

This section describes interrupt.

■ Interrupt

If the interrupt controller and CPU are ready to accept interrupts when an interrupt request is output from the internal resources or by an external-interrupt input, the CPU temporarily suspends the currently-executing instruction and executes the interrupt-processing program. Figure 3.3-1 "Interrupt-processing Flowchart" shows the interrupt-processing flowchart.

Figure 3.3-1 Interrupt-processing Flowchart



All interrupts are disabled after a reset is cleared. Therefore, initialize interrupts in the main program (1). Each resource generating interrupts and the interrupt-level-setting registers (ILR1 to ILR3) in the interrupt controller corresponding to these interrupts are to be initialized. The levels of all interrupts can be set by the interrupt-level-setting registers (ILR1 to ILR3) in the interrupt controller. The interrupt level can be set from 1 to 3, where 1 indicates the highest level, and 2 the second highest level. Level 3 indicates that no interrupt occurs. The interrupt request of level 3 cannot be accepted. After initializing the registers, the main program executes various controls (2). Interrupts are generated from the resources (3). The highest-priority interrupt requests are identified from those occurring at the same time by the interrupt controller and are transferred to the CPU. The CPU then checks the current interrupt level and the status of the I-flag (4), and starts the interrupt processing.

The CPU performs the interrupt processing to save the contents of the current PC and PS in the stack (5) and fetches the entry addresses of the interrupt program from the interrupt vectors. After updating the IL value in the PS to the required one, the CPU starts executing the interrupt-processing routine.

Clear the interrupt sources (6) and process the interrupts in the user's interrupt-processing routine. Finally, restore the PC and PS values saved by the RETI instruction in the stack (8) to return to the interrupted instruction.

Note:

Unlike the F^2MC-8 family, A and T are not saved in the stack at the interrupt time.

Table 3.3-1 "Interrupt Sources and Interrupt Vectors" lists the relationships between each interrupt source and interrupt vector.

Table 3.3-1 Interrupt Sources and Interrupt Vectors

Interrupt source	Upper vector address	Lower vector address
IRQ0 (External interrupt)	$FFFA_H$	$FFFB_H$
IRQ1 (External interrupt)	$FFF8_H$	$FFF9_H$
IRQ2 (External interrupt)	$FFF6_H$	$FFF7_H$
IRQ3 (8/16-bit timer counter timer 1)	$FFF4_H$	$FFF5_H$
IRQ4 (8/16-bit timer counter timer 2)	$FFF2_H$	$FFE3_H$
IRQ5 (Unused)	$FFF0_H$	$FFF1_H$
IRQ6 (Unused)	$FFEE_H$	$FFEF_H$
IRQ7 (Interval timer)	$FFEC_H$	$FFED_H$
IRQ8 (Unused)	$FFEA_H$	$FFEB_H$
IRQ9 (Unused)	$FFE8_H$	$FFE9_H$
IRQA (Wake-up)	$FFE6_H$	$FFE7_H$

3.4 Low-power Consumption Modes

This section describes low-power consumption modes.

■ Low-power Consumption Modes

The MB89990 series of microcontrollers have two standby modes: sleep and stop to reduce the power consumption. Writing to the standby control register (STBC) switches to these two standby modes. See 2.1.4 for setting and releasing each mode.

The MB89990 series of microcontrollers have a double clock module, and the low-power consumption modes vary with the main clock and subclock modes. Whether or not an oscillation stabilization period is required at release from each low-power consumption mode depends on the mask option of the power-on reset (See 2.1.4).

Table 3.4-1 Low-power Consumption Mode at Each Clock Mode

Function		Main mode		
		RUN	SLEEP	STOP
Clock oscillation		Operate	Operate	Stop
CPU	Instruction	Operate	Stop	Stop
	ROM	Operate	Hold	Hold
	RAM			
Resource	I/O	Operate	Hold	Hold
	Time-base timer	Operate	Operate	Stop
	16-bit timer	Operate	Operate	Stop
	8-bit SIO	Operate	Operate	Stop
	ADC	Operate	Operate	Stop
	External interrupt	Operate	Operate	Operate
	Buzzer output	Operate	Operate	Stop
	Watchdog timer	Operate	Stop	Stop

3.5 Pin States for Sleep, Stop and Reset

This section describes the pin states for sleep, stop, and reset.

■ Pin States for Sleep, Stop, and Reset

The state of each pin of the MB89990 series of microcontrollers at sleep, stop, and reset is as follows:

(1) Sleep

The pin state immediately before the sleep state is held.

(2) Stop

The pin state immediately before the stop state is held when the stop mode is started and bit 5 of the standby-control register (STBC) is set to 0; the impedance of the output and input/output pins goes High when the bit is set to 1.

(3) Reset

The impedance of all I/O and resource pins (excluding pins for pull-up option) goes High.

Table 3.5-1 Pin State of MB89990

Pin name	Normal	Sleep	Stop SPL = 0	Stop SPL = 1	Reset
P00/INT20 to P07/INT27	Port/resource I/O	Previous state	Previous state	High impedance*2, *3	High impedance
X0	Input for ocillation	Input for ocillation	High impedance	High impedance	Input for ocillation
X1	Output for ocillation	Output for ocillation	H output	H output	Output for ocillation
TEST	Test input	Test input	Test input	Test input	Test input
$\overline{RST}$	Reset input	Reset input	Reset input	Reset input	Reset input*1
P30 to P37/RCO	Port/resource I/O	Previous state	Previous state	High impedance*2, *3	High impedance
P40 to P45	Port	Previous state	Previous state	High impedance*3	High impedance

*1 Reset pin is output in some option setting.

*2 The internal input level is fixed for port and resource inputs to prevent leakage due to open input. However, when external interrupt is enabled, only input is allowed for P00 to P07 and P34 to P36.

*3 Pins with the pull-up option provided by selecting the mask option are the pull-up state.

CHAPTER 4 INSTRUCTIONS

This chapter describes instructions.

4.1 Transfer Instructions

This section describes the transfer instructions.

■ Transfer Instructions

Mnemonic	~	#	Operation	TL	TH	AH	N Z V C	OP code
MOV dir,A	3	2	(dir) ← (A)	–	–	–	– – – –	45
MOV @IX +off,A	4	2	((IX) +off) ← (A)	–	–	–	– – – –	46
MOV ext,A	4	3	(ext) ← (A)	–	–	–	– – – –	61
MOV @EP,A	3	1	((EP)) ← (A)	–	–	–	– – – –	47
MOV Ri,A	3	1	(Ri) ← (A)	–	–	–	– – – –	48 to 4F
MOV A,#d8	2	2	(A) ← d8	AL	–	–	+ + – –	04
MOV A,dir	3	2	(A) ← (dir)	AL	–	–	+ + – –	05
MOV A,@IX +off	4	2	(A) ← ((IX) +off)	AL	–	–	+ + – –	06
MOV A,ext	4	3	(A) ← (ext)	AL	–	–	+ + – –	60
MOV A,@A	3	1	(A) ← ((A))	AL	–	–	+ + – –	92
MOV A,@EP	3	1	(A) ← ((EP))	AL	–	–	+ + – –	07
MOV A,Ri	3	1	(A) ← (Ri)	AL	–	–	+ + – –	08 to 0F
MOV dir,#d8	4	3	(dir) ← d8	–	–	–	– – – –	85
MOV @IX +off,#d8	5	3	((IX) +off) ← d8	–	–	–	– – – –	86
MOV @EP,#d8	4	2	((EP)) ← d8	–	–	–	– – – –	87
MOV Ri,#d8	4	2	(Ri) ← d8	–	–	–	– – – –	88 to 8F
MOVW dir,A	4	2	(dir) ← (AH),(dir + 1) ← (AL)	–	–	–	– – – –	D5
MOVW @IX +off,A	5	2	((IX) +off) ← (AH), ((IX) +off + 1) ← (AL)	–	–	–	– – – –	D6
MOVW ext,A	5	3	(ext)←(AH),(ext+1)←(AL)	–	–	–	– – – –	D4
MOVW @EP,A	4	1	((EP))←(AH),((EP)+1)←(AL)	–	–	–	– – – –	D7
MOVW EP,A	2	1	(EP) ← (A)	–	–	–	– – – –	E3
MOVW A,#d16	3	3	(A) ← d16	AL	AH	dH	+ + – –	E4
MOVW A,dir	4	2	(AH) ← (dir), (AL) ← (dir + 1)	AL	AH	dH	+ + – –	C5
MOVW A,@IX +off	5	2	(AH) ← ((IX) +off), (AL) ← ((IX) +off + 1)	AL	AH	dH	+ + – –	C6
MOVW A,ext	5	3	(AH)←(ext),(AL)←(ext+1)	AL	AH	dH	+ + – –	C4
MOVW A,@A	4	1	(AH)←((A)),(AL)←((A))+1)	AL	AH	dH	+ + – –	93
MOVW A,@EP	4	1	(AH)←((EP)),(AL)←((EP)+1)	AL	AH	dH	+ + – –	C7
MOVW A,EP	2	1	(A) ← (EP)	–	–	dH	– – – –	F3
MOVW EP,#d16	3	3	(EP) ← d16	–	–	–	– – – –	E7
MOVW IX,A	2	1	(IX) ← (A)	–	–	–	– – – –	E2
MOVW A,IX	2	1	(A) ← (IX)	–	–	dH	– – – –	F2
MOVW SP,A	2	1	(SP) ← (A)	–	–	–	– – – –	E1
MOVW A,SP	2	1	(A) ← (SP)	–	–	dH	– – – –	F1
MOV @A,T	3	1	((A)) ← (T)	–	–	–	– – – –	82
MOVW @A,T	4	1	((A))←(TH),((A)+1)←(TL)	–	–	–	– – – –	83
MOVW IX,#d16	3	3	(IX) ← d16	–	–	–	– – – –	E6
MOVW A,PS	2	1	(A) ← (PS)	–	–	dH	– – – –	70
MOVW PS,A	2	1	(PS) ← (A)	–	–	–	+ + + +	71
MOVW SP,#d16	3	3	(SP) ← d16	–	–	–	– – – –	E5
SWAP	2	1	(AH) ↔ (AL)	–	–	AL	– – – –	10
SETB dir: b	4	2	(dir): b ← 1	–	–	–	– – – –	A8 to AF
CLRB dir: b	4	2	(dir): b ← 0	–	–	–	– – – –	A0 to A7
XCH A,T	2	1	(AL) ↔ (TL)	AL	–	–	– – – –	42
XCHW A,T	3	1	(A) ↔ (T)	AL	AH	dH	– – – –	43
XCHW A,EP	3	1	(A) ↔ (EP)	–	–	dH	– – – –	F7
XCHW A,IX	3	1	(A) ↔ (IX)	–	–	dH	– – – –	F6
XCHW A,SP	3	1	(A) ↔ (SP)	–	–	dH	– – – –	F5
MOVW A,PC	2	1	(A) ← (PC)	–	–	dH	– – – –	F0

Notes

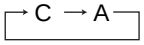
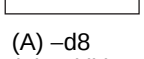
1. During byte transfer to A, T <-- A is restricted to low bytes.

2. Operands in more than one operand instruction must be stored in the order in which their mnemonics are written. (Reverse arrangement of F^2MC-8 family)

4.2 Operation Instruction

This section describes the operation instructions.

■ Operation Instructions

Mnemonic	~	#	Operation	TL	TH	AH	N Z V C	OP code
ADDC A,Ri	3	1	(A) ← (A) + (Ri) + C	–	–	–	+ + + +	28 to 2F
ADDC A,#d8	2	2	(A) ← (A) + d8 + C	–	–	–	+ + + +	24
ADDC A,dir	3	2	(A) ← (A) + (dir) + C	–	–	–	+ + + +	25
ADDC A,@IX +off	4	2	(A) ← (A) + ((IX) +off) + C	–	–	–	+ + + +	26
ADDC A,@EP	3	1	(A) ← (A) + ((EP)) + C	–	–	–	+ + + +	27
ADDCW A	3	1	(A) ← (A) + (T) + C	–	–	dH	+ + + +	23
ADDC A	2	1	(AL) ← (AL) + (TL) + C	–	–	–	+ + + +	22
SUBC A,Ri	3	1	(A) ← (A) – (Ri) – C	–	–	–	+ + + +	38 to 3F
SUBC A,#d8	2	2	(A) ← (A) – d8 – C	–	–	–	+ + + +	34
SUBC A,dir	3	2	(A) ← (A) – (dir) – C	–	–	–	+ + + +	35
SUBC A,@IX +off	4	2	(A) ← (A) – ((IX) +off) – C	–	–	–	+ + + +	36
SUBC A,@EP	3	1	(A) ← (A) – ((EP)) – C	–	–	–	+ + + +	37
SUBCW A	3	1	(A) ← (T) – (A) – C	–	–	dH	+ + + +	33
SUBC A	2	1	(AL) ← (TL) – (AL) – C	–	–	–	+ + + +	32
INC Ri	4	1	(Ri) ← (Ri) + 1	–	–	–	+ + + –	C8 to CF
INCW EP	3	1	(EP) ← (EP) + 1	–	–	–	– – – –	C3
INCW IX	3	1	(IX) ← (IX) + 1	–	–	–	– – – –	C2
INCW A	3	1	(A) ← (A) + 1	–	–	dH	+ + – –	C0
DEC Ri	4	1	(Ri) ← (Ri) – 1	–	–	–	+ + + –	D8 toDF
DECW EP	3	1	(EP) ← (EP) – 1	–	–	–	– – – –	D3
DECW IX	3	1	(IX) ← (IX) – 1	–	–	–	– – – –	D2
DECW A	3	1	(A) ← (A) – 1	–	–	dH	+ + – –	D0
MULU A	19	1	(A) ← (AL) × (TL)	–	–	dH	– – – –	01
DIVU A	21	1	(A) ← (T) / (AL),MOD (T)	dL	00	00	– – – –	11
ANDW A	3	1	(A) ← (A) ∧ (T)	–	–	dH	+ + R –	63
ORW A	3	1	(A) ← (A) ∨ (T)	–	–	dH	+ + R –	73
XORW A	3	1	(A) ← (A) ∀ (T)	–	–	dH	+ + R –	53
CMP A	2	1	(TL) – (AL)	–	–	–	+ + + +	12
CMPW A	3	1	(T) – (A)	–	–	–	+ + + +	13
RORC A	2	1		–	–	–	+ + – +	03
ROLC A	2	1		–	–	–	+ + – +	02
CMP A,#d8	2	2	(A) –d8	–	–	–	+ + + +	14
CMP A,dir	3	2	(A) – (dir)	–	–	–	+ + + +	15
CMP A,@EP	3	1	(A) – ((EP))	–	–	–	+ + + +	17
CMP A,@IX +off	4	2	(A) – ((IX) +off)	–	–	–	+ + + +	16
CMP A,Ri	3	1	(A) – (Ri)	–	–	–	+ + + +	18 to 1F
DAA	2	1	Decimal adjust foraddition	–	–	–	+ + + +	84
DAS	2	1	Decimal adjust forsubtraction	–	–	–	+ + + +	94
XOR A	2	1	(A) ← (AL) ∀ (TL)	–	–	–	+ + R –	52
XOR A,#d8	2	2	(A) ← (AL) ∀ d8	–	–	–	+ + R –	54
XOR A,dir	3	2	(A) ← (AL) ∀ (dir)	–	–	–	+ + R –	55
XOR A,@EP	3	1	(A) ← (AL) ∀ ((EP))	–	–	–	+ + R –	57
XOR A,@IX +off	4	2	(A) ← (AL) ∀ ((IX) +off)	–	–	–	+ + R –	56
XOR A,Ri	3	1	(A) ← (AL) ∀ (Ri)	–	–	–	+ + R –	58 to 5F

Mnemonic	~	#	Operation	TL	TH	AH	N Z V C	OP code
AND A	2	1	(A) ← (AL) ∧ (TL)	–	–	–	+ + R –	62
AND A,#d8	2	2	(A) ← (AL) ∧ d8	–	–	–	+ + R –	64
AND A,dir	3	2	(A) ← (AL) ∧ (dir)	–	–	–	+ + R –	65
AND A,@EP	3	1	(A) ← (AL) ∧ ((EP))	–	–	–	+ + R –	67
AND A,@IX +off	4	2	(A) ← (AL) ∧ ((IX) +off)	–	–	–	+ + R –	66
AND A,Ri	3	1	(A) ← (AL) ∧ (Ri)	–	–	–	+ + R –	68 to 6F
OR A	2	1	(A) ← (AL) ∨ (TL)	–	–	–	+ + R –	72
OR A,#d8	2	2	(A) ← (AL) ∨ d8	–	–	–	+ + R –	74
OR A,dir	3	2	(A) ← (AL) ∨ (dir)	–	–	–	+ + R –	75
OR A,@EP	3	1	(A) ← (AL) ∨ ((EP))	–	–	–	+ + R –	77
OR A,@IX +off	4	2	(A) ← (AL) ∨ ((IX) +off)	–	–	–	+ + R –	76
OR A,Ri	3	1	(A) ← (AL) ∨ (Ri)	–	–	–	+ + R –	78 to 7F
CMP dir,#d8	5	3	(dir) – d8	–	–	–	+ + + +	95
CMP @EP,#d8	4	2	((EP)) – d8	–	–	–	+ + + +	97
CMP @IX +off,#d8	5	3	((IX) +off) – d8	–	–	–	+ + + +	96
CMP Ri,#d8	4	2	(Ri) – d8	–	–	–	+ + + +	98 to 9F
INCW SP	3	1	(SP) ← (SP) + 1	–	–	–	– – – –	C1
DECW SP	3	1	(SP) ← (SP) – 1	–	–	–	– – – –	D1

4.3 Branch Instructions

This section describes the branch instructions.

■ Branch Instructions

Mnemonic	~	#	Operation	TL	TH	AH	N Z V C	OP code
BZ/BEQ rel	3	2	If Z = 1 then PC ← PC + rel	–	–	–	– – – –	FD
BNZ/BNE rel	3	2	If Z = 0 then PC ← PC + rel	–	–	–	– – – –	FC
BC/BLO rel	3	2	If C = 1 then PC ← PC + rel	–	–	–	– – – –	F9
BNC/BHS rel	3	2	If C = 0 then PC ← PC + rel	–	–	–	– – – –	F8
BN rel	3	2	If N = 1 then PC ← PC + rel	–	–	–	– – – –	FB
BP rel	3	2	If N = 0 then PC ← PC + rel	–	–	–	– – – –	FA
BLT rel	3	2	If V∀N=1 then PC←PC+rel	–	–	–	– – – –	FF
BGE rel	3	2	If V∀N=0 then PC←PC+reI	–	–	–	– – – –	FE
BBC dir: b,rel	5	3	If(dir:b)=0 then PC←PC+rel	–	–	–	– + – –	B0 to B7
BBS dir: b,rel	5	3	If(dir:b)=1 then PC←PC+rel	–	–	–	– + – –	B8 to BF
JMP @A	2	1	(PC) ← (A)	–	–	–	– – – –	E0
JMP ext	3	3	(PC) ← ext	–	–	–	– – – –	21
CALLV #vct	6	1	Vector call	–	–	–	– – – –	E8 to EF
CALL ext	6	3	Subroutine call	–	–	–	– – – –	31
XCHW A,PC	3	1	(PC) ← (A),(A) ← (PC) + 1	–	–	dH	– – – –	F4
RET	4	1	Return from subrountine	–	–	–	– – – –	20
RETI	6	1	Return form interrupt	–	–	–	Restore	30

4.4 Other Instructions

This section describes the other instructions.

■ Other Instructions

Mnemonic	~	#	Operation	TL	TH	AH	N Z V C	OP code
PUSHW A	4	1		–	–	–	– – – –	40
POPW A	4	1		–	–	dH	– – – –	50
PUSHW IX	4	1		–	–	–	– – – –	41
POPW IX	4	1		–	–	–	– – – –	51
NOP	1	1		–	–	–	– – – –	00
CLRC	1	1		–	–	–	– – – R	81
SETC	1	1		–	–	–	– – – S	91
CLRI	1	1		–	–	–	– – – –	80
SETI	1	1		–	–	–	– – – –	90

4.5 F²MC-8L Family Instruction Map

This section describes the F²MC-8L family instruction map.

■ F²MC-8L Family Instruction Map

H / L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SWAP	RET	RETI	PUSHW A	POPW A	MOV A,ext	MOVW A,PS	CLRI	SETI	CLRB dir:0	BBC dir:0,rel	INCW A	DECW A	JMP @A	MOVW A,PC
1	MULU A	DIVU A	JMP addr16	CALL addr16	PUSHW IX	POPW IX	MOV extA	MOVW PSA	CLRC	SETC	CLRB dir:1	BBC dir:1,rel	INCW SP	DECW SP	MOVW SPA	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 IXA	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 EPA	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 extA	MOVW A,#d16	XCHW A,PC
5	MOV A,dir	CMP A,dir	ADDC A,dir	SUBC A,dir	MOV dirA	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 dirA	MOVW SP,#d16	XCHW A,SP
6	MOV A,@IX +d	CMP A,@IX +d	ADDC A,@IX +d	SUBCA,@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 @EPA	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 @EPA	MOVW EP,#d16	XCHW A,EP
8	MOV A,R0	CMP A,R0	ADDC A,R0	SUBC A,R0	MOV R0A	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 R1A	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 R2A	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 R3A	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 R4A	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 R5A	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 R6A	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 R7A	XOR A,R7	AND A,R7	OR A,R7	MOV R7,#d8	CMP R7,#d8	SETB dir:7	BBS dir:7,rel	INC R7	DEC R7	CALLV #7	BLT rel

CHAPTER 5 MASK OPTIONS

This chapter describes mask options.

5.1 Mask Options

5.1 Mask Options

This section describes the mask options.

■ Mask Options

Table 5.1-1 Mask Options

No.	Part number		MB89997	MB89P195	MB89PV190
	Specifying procedure		Specify when ordering masking	Specify when ordering masking	Fixed
1	Port pull-up resistors	00 to P07 P30 to P37	Selectable by pin	Selectable by pin	Not available
		P00 to P03 P40 to P45	Selectable by pin	Selectable by pin	Not available
2	Power-on reset selection - Power-on reset provided - No power-on reset		Selectable	Enabled	Enabled
3	Selection of oscillation stabilization wait time (at 4.2 MHz)*1 - $2^{18}/F_C$ (approx. 62.4 ms) - $2^{16}/F_C$ (approx. 15.6 ms) - $2^{12}/F_C$ (approx. 0.98 ms) - $2^2/F_C$ (approx. 0 ms)		Selectable	Selectable	Fixed to $2^{16}/F_C$
4	Reset pin output - Reset output provided - No reset output		Selectable	Selectable	Output enabled
5	Oscillation type of clock - 1 ceramic oscillator - 2 crystal oscillator - 3 CR		Selectable	Selectable	"1" only

*1: The oscillation stabilization delay time is generated by dividing the original clock oscillation. The time described in this item should be used as a guideline since the oscillation cycle is unstable immediately after oscillation starts. "f" indicates the original oscillation frequency.

APPENDIX

The appendix describes I/O map and EPROM setting for MB89P195.

APPENDIX A I/ O Map

APPENDIX B EPROM Setting for MB89P195

APPENDIX A I/O Map

Appendix A describes the I/O map.

■ /O Map

Address	Read/write	Register name	Register description
00_H	(R/W)	PDR0	Port 0 data register
01_H	(W)	DDR0	Port 0 data direction register
02_H to 07_H	Vacancy		
08_H	(R/W)	STBC	Standby control register
09_H	(R/W)	WDTC	Watchdog control register
$0A_H$	(R/W)	TBTC	Time-base timer control register
$0B_H$	Vacancy		
$0C_H$	(R/W)	PDR3	Port 3 data register
$0D_H$	(W)	DDR3	Port 3 data direction register
$0E_H$	(R/W)	PDR4	Port 4 data register
$0F_H$ to 13_H	Vacancy		
14_H	(R/W)	RCR1	Remote-control register 1
15_H	(R/W)	RCR2	Remote-control register 2
16_H	Vacancy		
17_H	Vacancy		
18_H	(R/W)	T2CR	Timer 2 control register
19_H	(R/W)	T1CR	Timer 1 control register
$1A_H$	(R/W)	T2DR	Timer 2 data register
$1B_H$	(R/W)	T1DR	Timer 1 data register
$1C_H$ to 22_H	Vacancy		
23_H	(R/W)	EIC1	External interrupt control register 1
24_H	(R/W)	EIC2	External interrupt control register 2
25_H to 31_H	Vacancy		
32_H	(R/W)	EIE2	External interrupt 2 enable register
33_H	(R/W)	EIF2	External interrupt 2 flag register
34_H to $7B_H$	Vacancy		

Address	Read/write	Register name	Register description
$7C_H$	(W)	ILR1	Interrupt level setting register 1
$7D_H$	(W)	ILR2	Interrupt level setting register 2
$7E_H$	(W)	ILR3	Interrupt level setting register 3
$7F_H$	—	ITR	Interrupt test register

Note:

— indicate the vacant area, it is not used.

APPENDIX B EPROM Setting for MB89P195

Appendix B describes the EPROM setting for MB89P195.

■ EPROM Setting for MB89P195

MB89P195 is provided with the function corresponding to MBM27C256A by EPROM setting. The setting can be performed by writing program data with general-purpose EPROM writer through adaptor for exclusive use .

However, the electric signature mode is not supported.

❍ Setting

(1) Set the EPROM writer to MBM27C256A.

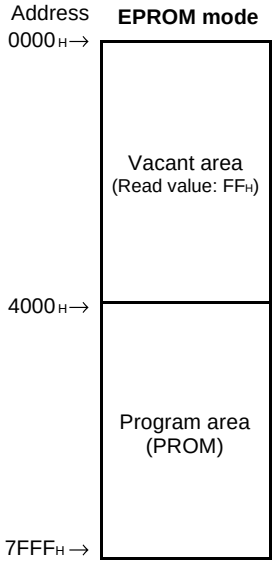
(2) Load the program data from address 4000_H to address $7FFF_H$ of EPROM writer.

The data is loaded from address $0C000_H$ to address $0FFFF_H$ in the operation mode, and from address 4000_H to address $7FFF_H$ in the EPROM mode.)

(3) Write the data from 0000_H with the EPROM writer.

(Writing to the correct address cannot be performed other than from 0000_H.)

The memory space in the EPROM mode is as follows:



❍ ROM writer adapter (Sun Hayato Co., Ltd.)

Package	Model No. of applicable adapter
FPT-28P-M02	ROM-28SOP-28DP-8L

INDEX

The index follows on the next page.
This is listed in alphabetic order.

Index

CM25-10133-2E

FUJITSU SEMICONDUCTOR • MICROCONTROLLER MANUAL

F^2MC-8L FAMILY

8-BIT MICROCONTROLLER

MB89990 Series

HARDWARE MANUAL

March 2000 the second edition

Published **FUJITSU LIMITED** Electronic Devices

Edited Technical Communication Dept.

FUJITSU

CM25-10133-2E

FUJITSU SEMICONDUCTOR F^{2}MC-8L FAMILY 8-BIT MICROCONTROLLER MB89990 Series HARDWARE MANUAL