FUJITSU SEMICONDUCTOR
DATA SHEET

DS07-13504-2E

16-bit Proprietary Microcontroller

CMOS

F²MC-16F MB90230 Series

MB90233/234/P234/W234

■ DESCRIPTION

The MB90230 series is a member of general-purpose, 16-bit microcontrollers designed for those applications which require high-speed realtimeprocessing, proving to be suitable for various industrial machines, camera and video devices, OA equipment, and for process control. The CPU used in this series is the F²MC*-16F. The instruction set for the F²MC-16F CPU core is designed to be optimized for controller applications while inheriting the AT architecture of the F²MC-16/16H series, allowing a wide range of control tasks to be processed efficiently at high speed.

The peripheral resources integrated in the MB90230 series include: the UART (clock asynchronous/synchronous transfer) × 1 channel, the extended serial I/O interface × 1 channel, the A/D converter (8/10-bit precision) × 8 channels, the D/A converter (8-bit precision) × 2 channels, the level comparator × 1 channel, the external interrupt input × 4 lines, the 8-bit PPG timer (PWM/single-shot function) × 1 channel, the 8-bit PWM controller × 6 channels, the 16-bit free run timer × 1 channel, the input capture unit × 4 channels, the output compare unit × 6 channels, and the serial E²PROM interface.

*: F²MC stands for FUJITSU Flexible Microcontroller.

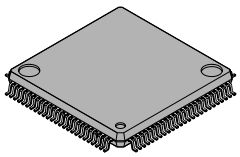
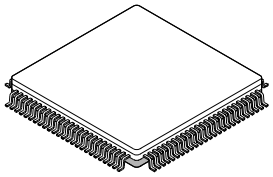
■ FEATURES

F²MC-16F CPU block

- Minimum execution time: 62.5 ns (at machine clock frequency of 16 MHz)
- Instruction set optimized for controllers
  Various data types supported (bit, byte, word, and long-word)
  Extended addressing modes: 23 types
  High coding efficiency
  Higher-precision operation enhanced by a 32-bit accumulator
  Signed multiplication and division instructions

(Continued)

■ PACKAGE

100-pin Plastic LQFP	100-pin Ceramic LQFP
	
(FPT-100P-M05)	(FPT-100C-C01)

(Continued)

- Enhanced instructions applicable to high-level language (C) and multitasking
  System stack pointer
  Enhanced pointer-indirect instructions
  Barrel shift instructions
- Increased execution speed: 8-byte instruction queue
- 8-level, 32-factor powerful interrupt service functions
- Automatic transfer function independent of the CPU (EI^2OS)
- General-purpose ports: Up to 84 lines
  Ports with input pull-up resistor available: 24 lines
  Ports with output open-drain available: 9 lines

Peripheral blocks

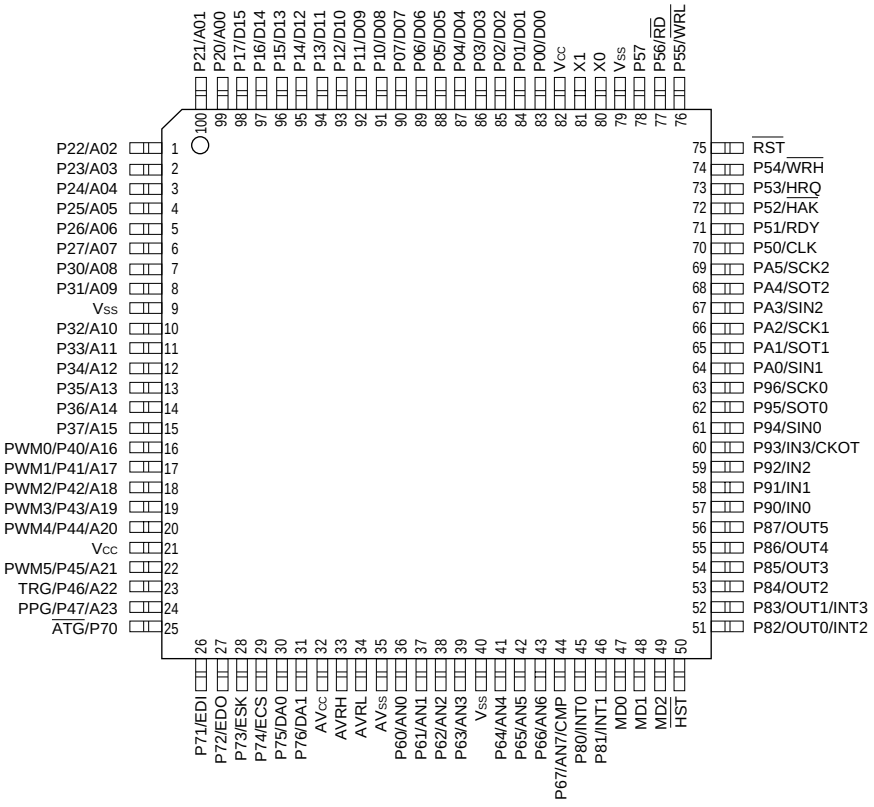
- ROM: 48 Kbytes (MB90233)
  96 Kbytes (MB90234)
  EPROM: 96 Kbytes (MB90W234)
  One-time PROM: 96 Kbytes (MB90P234)
- RAM: 2 Kbytes (MB90233)
  3 Kbytes (MB90234/W234/P234)
- PWM control circuit: (simple 8 bits): 6 channels
- Serial interface
  UART: 1 channel
  Extended serial I/O interface
  Switchable I/O port: 1 channel
  Communication prescaler (Source clock generator for the UART, serial I/O interface, CKOT, and level comparator): 1 channel
- Serial E^2PROM interface: 1 channel
- A/D converter with 8/10-bit resolution: input 8 channels
- Level comparator: 1 channel
  4-bit D/A converter integrated
- D/A converter with 8-bit resolution: 2 channels
  8-bit PPG timer: 1 channel
- Input/output timer
  16-bit free run timer: 1 channel
  16-bit output compare unit: 6 channels
  16-bit input capture unit: 4 channels
- 18-bit timebase timer
- Watchdog timer function
- Standby modes
  Sleep mode
  Stop mode

■ PRODUCT LINEUP

Parameter \ Part number	MB90233	NB90234	MB90P234	MB90W234	MB90V230
Classification	Mask ROM products		One-time PROM model	EPROM model	Evaluation model
ROM size	48 Kbytes	96 Kbytes	96 Kbytes	96 Kbytes	—
RAM size	2 Kbytes	3 Kbytes	3 Kbytes	3 Kbytes	4 Kbytes
CPU functions	Number of instructions: 420 Instruction bit length: 8 or 16 bits Instruction length: 1 to 7 bytes Data bit length: 1, 4, 8, 16, or 32 bits Minimum execution time: 62.5 ns at 16 MHz (internal)				
Ports	Up to 84 lines I/O ports (CMOS): 51 I/O ports (CMOS) with pull-up resistor available: 24 I/O ports (open-drain): 9				
UART	Number of channels: 1 (switchable I/O) Clock synchronous communication (2404 to 38460 bps, full-duplex double buffering) Clock asynchronous communication (500K to 5M bps, full-duplex double buffering)				
Serial interface	Number of channels: 1 Internal or external clock mode Clock synchronous transfer (62.5 kHz to 1 MHz, “LSB first” or “MSB first” transfer)				
A/D converter	Resolution: 10 or 8 bits, Number of input lines: 4 Single conversion mode (conversion for a specified input channel) Scan conversion mode (continuous conversion for specified consecutive channels) Continuous conversion mode (repeated conversion for a specified channel) Stop conversion mode (periodical conversion)				
D/A converter	Resolution: 8 bits, Number of output pins: 2				
Level comparator	Comparison to internal D/A converter (4-bit resolution)				
PWM	Number of channels: 6 8-bit PWM control circuit (operation of 1×ϕ, 2×ϕ, 16×ϕ, 32×ϕ)				
PPG timer	Number of channels: 1 channel with 8-bit resolution PWM function: Continuous output of pulse synchronous to trigger Single-shot function: Output of single pulse by trigger				
Serial E^2PROM interface	Number of channels: 1 Instruction code (NS type) Variable address length: 8 to 11 bits (with address increment function) Variable data length: 8 or 16 bits				
Timer	Number of channels: 6 16-bit reload timer operation (operation clock cycle of 0.25 µs to 1.05 s)				
Free run timer	Number of channels: 1 16-bit input capture unit: 4 channels 16-bit output compare unit: 6 channels				
External interrupt input	Number of input pins: 4				
Standby mode	Stop mode and sleep mode				
Package	FPT-100P-M05			FPT-100C-C01	PGA256-A02

■ PIN ASSIGNMENT

(TOP VIEW)



(FPT-100P-M05)
(FPT-100C-C01)

■ PIN DESCRIPTION

Pin no.	Pin name	Circuit type	Function
80	X0	A	Oscillator pins
81	X1		
82	V_{CC}	—	Power supply pin
83 to 90	P00 to P07	G	General-purpose I/O port An input pull-up resistor can be added to the port by setting the pull-up resistor setting register. These pins serve as D00 to D07 pins in bus modes other than the single-chip mode.
	D00 to D07		I/O pins for the lower eight bits of the external data bus. These pins are enabled in an external-bus enabled mode.
91 to 98	P10 to P17	G	General-purpose I/O port An input pull-up resistor can be added to the port by setting the pull-up resistor setting register. These pins are enabled in the single-chip mode with the external-bus enabled and the 8-bit data bus specified.
	D08 to D15		I/O pins for the upper eight bits of the external data bus These pins are enabled in an external-bus enabled mode with the 16-bit data bus specified.
99, 100 1 to 6	P20 to P27	G	General-purpose I/O port An input pull-up resistor can be added to the port by setting the pull-up resistor setting register. These pins are enabled in the single-chip mode.
	A00 to A07		I/O pins for the lower eight bits of the external data bus These pins are enabled in an external-bus enabled mode.
7, 8	P30, P31	E	General-purpose I/O port This port is enabled in the single-chip mode or when the middle address control register setting is “port.”
	A08, A09		I/O pins for the middle eight bits of the external data bus These pins are enabled in an external-bus enabled mode when the middle address control register setting is “address.”
9	V_{SS}	—	Power supply pin
10 to 15	P32 to P37	E	General-purpose I/O port This port is enabled in the single-chip mode or when the middle address control register setting is “port.”
	A10 to A15		I/O pins for the middle eight bits of the external data bus These pins are enabled in an external-bus enabled mode when the middle address control register setting is “address.”

(Continued)

Pin no.	Pin name	Circuit type	Function
16	P40	E	General-purpose I/O port This port is enabled in the single-chip mode or when the upper address control register setting is “port.”
	A16		Output pin for external address A16 This pin is enabled in the external-bus enabled mode with the upper address control register set to “address.”
	PWM0		This pin serves as the output pin for 8-bit PWM0 The pin is enabled for output by the control status register.
17	P41	E	General-purpose I/O port This port is enabled in the single-chip mode or when the upper address control register setting is “port.”
	A17		Output pin for external address A17 This pin is enabled in the external-bus enabled mode with the upper address control register set to “address.”
	PWM1		This pin serves as the output pin for 8-bit PWM1. The pin is enabled for output by the control status register.
18	P42	E	General-purpose I/O port This port is enabled in the single-chip mode or when the upper address control register setting is “port.”
	A18		Output pin for external address A18 This pin is enabled in the external-bus enabled mode with the upper address control register set to “address.”
	PWM2		This pin serves as the output pin for 8-bit PWM2. This pin is enabled for output by the control status register.
19	P43	E	General-purpose I/O port This port is enabled in the single-chip mode or when the upper address control register setting is “port.”
	A19		Output pin for external address A19 This pin is enabled in the external-bus enabled mode with the upper address control register set to “address.”
	PWM3		This pin serves as the output pin for 8-bit PWM3. This pin is enabled for output by the control status register.
20	P44	E	General-purpose I/O port This port is enabled in the single-chip mode or when the upper address control register setting is “port.”
	A20		Output pin for external address A20 This pin is enabled in the external-bus enabled mode with the upper address control register set to “address.”
	PWM4		This pin serves as the output pin for 8-bit PWM4. The pin is enabled for output by the control status register.
21	V_{CC}	—	Power supply pin

(Continued)

Pin no.	Pin name	Circuit type	Function
22	P45	E	General-purpose I/O port This port is enabled in the single-chip mode or when the upper address control register setting is “port.”
	A21		Output pin for external address A21 This pin is enabled in the external-bus enabled mode with the upper address control register set to “address.”
	PWM5		This pin serves as the output pin for 8-bit PWM5. The pin is enabled for output by the control status register.
23	P46	L*1	General-purpose I/O port This port is enabled in the single-chip mode or when the upper address control register setting is “port.”
	A22		Output pin for external address A22 This pin is enabled in the external-bus enabled mode with the upper address control register set to “address.”
	TRG		This pin serves as the external trigger pin for the 8-bit PPG timer The pin is enabled for triggering by the control status register.
24	P47	E	General-purpose I/O port This port is enabled in the single-chip mode or when the upper address control register setting is “port.”
	A23		Output pin for external address A23 This pin is enabled in the external-bus enabled mode with the upper address control register set to “address.”
	PPG		This pin serves as the output pin for the 8-bit PPG timer. The pin is enabled for output by the control status register.
25	P70	L*1	General-purpose I/O port
	$\overline{\text{ATG}}$		External trigger input pin for the A/D converter This pin functions when enabled by the control status register.
26	P71	F	General-purpose I/O port
	EDI		Data input pin for the serial EEPROM interface This pin functions when enabled by the control status register.
27	P72	E	General-purpose I/O port
	EDO		Data output pin for the serial EEPROM interface This pin functions when enabled by the control status register.
28	P73	E	General-purpose I/O port
	ESK		Clock output pin for the serial EEPROM interface This pin functions when enabled by the control status register.
29	P74	E	General-purpose I/O port
	ECS		Chip select signal output pin for the serial EEPROM interface This pin functions when enabled by the control status register.

(Continued)

Pin no.	Pin name	Circuit type	Function
30, 31	P75, P76	K	General-purpose I/O port
	DA0 DA1		This pin serves as the D/A converter output pin. The pin functions when enabled by the control status register.
32	AV_{CC}	—	A/D converter power supply pin
33	AV_{RH}	—	“H” reference power supply pin for the A/D converter
34	AV_{RL}	—	“L” reference power supply pin for the A/D converter
35	AV_{SS}	—	A/D converter power pin (GND)
36 to 39	P60 to P63	J	General-purpose I/O port This port is enabled when the analog input enable register setting is “port.”
	AN0 to AN3		A/D converter analog input pins These pins are enabled when the analog input enable register setting is “analog input.”
40	V_{SS}	—	Power pin (GND)
41 to 43	P64 to P66	J	General-purpose I/O port This port is enabled when the analog input enable register setting is “port.”
	AN4 to AN6		A/D converter analog input pins These pins are enabled when the analog input enable register setting is “analog input.”
44	P67	J	General-purpose I/O port This port is enabled when the analog input enable register setting is “port.”
	AN7		A/D converter analog input pin This pin is enabled when the analog input enable register setting is “analog input.”
	CMP		Comparator input pin
45	P80	L*2	General-purpose I/O port This port is always enabled.
	INT0		External interrupt request input 0 Since this pin serves for interrupt request as required when external interrupt is enabled, other outputs must be off unless used intentionally.
46	P81	L*2	General-purpose I/O port This port is always enabled.
	INT1		External interrupt request input 1 Since this pin serves for interrupt request as required when external interrupt is enabled, other outputs must be off unless used intentionally.
47	MD0	C	Mode pin This pin must be fixed to V_{CC} or V_{SS}.
48	MD1	C	Mode pin This pin must be fixed to V_{CC} or V_{SS}.

(Continued)

Pin no.	Pin name	Circuit type	Function
49	MD2	C	Mode pin This pin must be fixed to V_{SS}.
50	$\overline{HST}$	D	Hardware standby input pin
51, 52	P82, P83	L*2	General-purpose I/O port
	OUT0, OUT1		Output compare output pins These pins function when enabled by the control status register.
	INT2, INT3		External interrupt request inputs 2 and 3. Since these pins serve for interrupt request as required when external interrupt is enabled, other outputs must be off unless used intentionally.
53 to 56	P84 to P87	E	General-purpose I/O port This pin is always enabled.
	OUT2 to OUT5		Output compare output pins These pins function when enabled by the control status register.
57 to 59	P90 to P92	L*1	General-purpose I/O port This port is always enabled.
	IN0 to IN2		Input capture edge input pins These pins function when enabled by the control status register.
60	P93	L*1	General-purpose I/O port This port is always enabled.
	IN3		Input capture edge input pin This pin functions when enabled by the control status register.
	CKOT		Prescaler output pin This pin functions when enabled by the control status register.
61	P94	I	General-purpose I/O port This port is always enabled. The port serves as an open-drain output depending on the open-drain setting register.
	SIN0		Serial data input pin for the UART This pin functions when enabled by the control status register.
62	P95	H	General-purpose I/O port This port is always enabled. The port serves as an open-drain output depending on the open-drain setting register.
	SOT0		Serial data output pin for the UART This pin functions when enabled by the control status register.
63	P96	I	General-purpose I/O port This port is always enabled. The port serves as an open-drain output depending on the open-drain setting register.
	SCK0		UART clock output pin This pin functions when enabled by the control status register.

(Continued)

Pin no.	Pin name	Circuit type	Function
64	PA0	I	General-purpose I/O port This port is always enabled. The port serves as an open-drain output depending on the open-drain setting register.
	SIN1		Serial data input pin for the extended serial I/O interface This pin functions when enabled by the control status register and by the serial port switching register.
65	PA1	H	General-purpose I/O port This port is always enabled. The port serves as an open-drain output depending on the open-drain setting register.
	SOT1		Serial data output pin for the extended serial I/O interface This pin functions when enabled by the control status register and by the serial port switching register.
66	PA2	I	General-purpose I/O port This port is always enabled. The port serves as an open-drain output depending on the open-drain setting register.
	SCK1		Clock output pin for the extended serial I/O interface This pin functions when enabled by the control status register and by the serial port switching register.
67	PA3	I	General-purpose I/O port This port is always enabled. The port serves as an open-drain output depending on the open-drain setting register.
	SIN2		Serial data input pin for the extended serial I/O interface This pin functions when enabled by the control status register and by the serial port switching register.
68	PA4	H	General-purpose I/O port This port is always enabled. The port serves as an open-drain output depending on the open-drain setting register.
	SOT2		Serial data output pin for the extended serial I/O interface This pin functions when enabled by the control status register and by the serial port switching register.
69	PA5	I	General-purpose I/O port This port is always enabled. The port serves as an open-drain output depending on the open-drain setting register.
	SCK2		Clock output pin for the extended serial I/O interface This pin functions when enabled by the control status register and by the serial port switching register. The pin is a general-purpose I/O port.

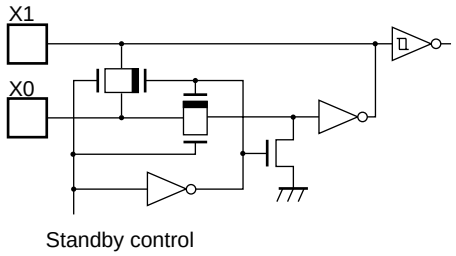
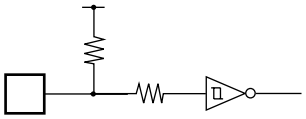
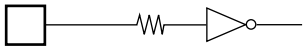
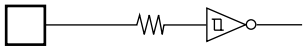
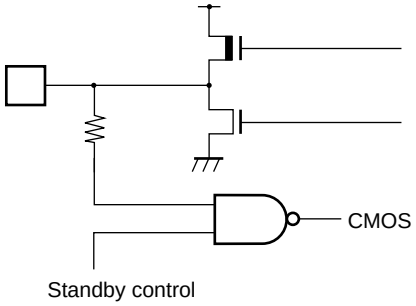
(Continued)

(Continued)

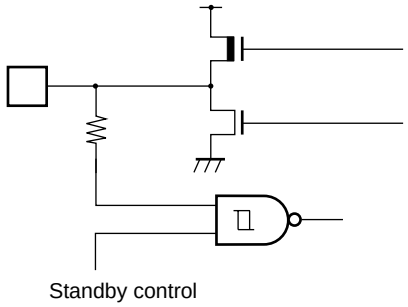
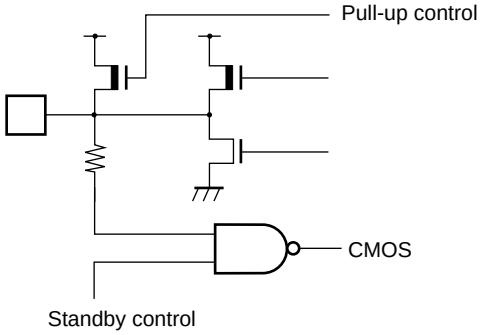
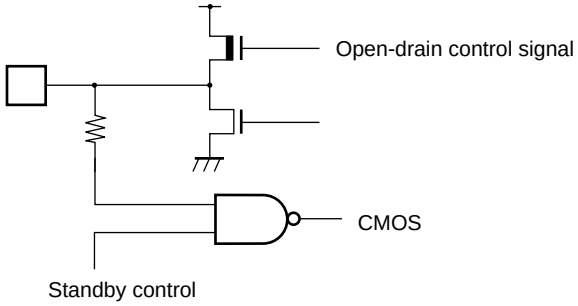
Pin no.	Pin name	Circuit type	Function
70	P50	H	This pin is enabled in the single-chip mode and when the CLK output is disabled.
	CLK		CLK output pin This pin is enabled in an external-bus enabled mode with the CLK output enabled.
71	P51	F	General-purpose I/O port This port is enabled in the single-chip mode.
	RDY		Ready signal input pin This pin is enabled in an external-bus enabled mode.
72	P52	E	General-purpose I/O port This port is enabled in the single-chip mode or when the hold function is disabled.
	$\overline{HAK}$		Hold acknowledge signal output pin This pin is enabled in the single-chip mode or when the hold function is enabled.
73	P53	E	General-purpose I/O port This port is enabled in the single-chip mode or when the hold function is disabled.
	HRQ		Hold acknowledge signal output pin This pin is enabled in the single-chip mode or when the hold function is enabled.
74	P54	E	General-purpose I/O port This port is enabled in the single-chip mode, in external-bus 8-bit mode, or when the WR pin output is disabled.
	$\overline{WRH}$		Write strobe output pin for the upper eight bits of the data bus This pin is enabled in an external-bus enabled mode and in external bus 16-bit mode with the WR pin output enabled.
75	$\overline{RST}$	B	Reset signal input pin
76	P55	E	This port is enabled in the single-chip mode, in external-bus 8-bit mode, or when the WR pin output is disabled
	$\overline{WRL}$		Write strobe output pin for the lower eight bits of the data bus This pin is enabled in an external-bus enabled mode and in external bus 16-bit mode with the WR pin output enabled. The pin is a general-purpose I/O port.
77	P56	E	This pin is enabled in the single-chip mode.
	$\overline{RD}$		Read strobe output pin for the data bus This pin is enabled in an external-bus enabled mode.
78	P57	E	General-purpose I/O port
79	V_{SS}	—	Power pin (GND)

*1: Enabled in any standby mode
*2: Enabled only in the hardware standby mode

■ I/O CIRCUIT TYPE

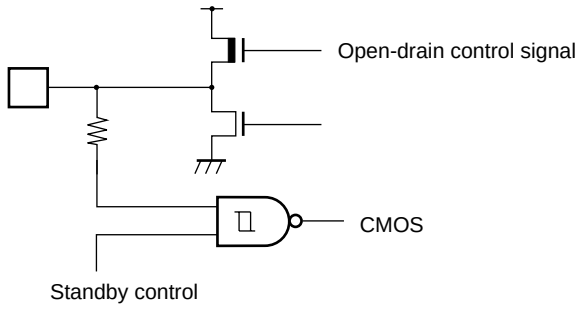
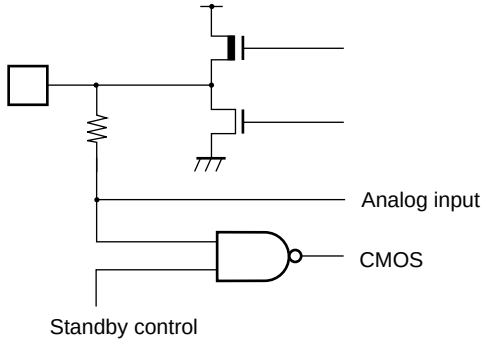
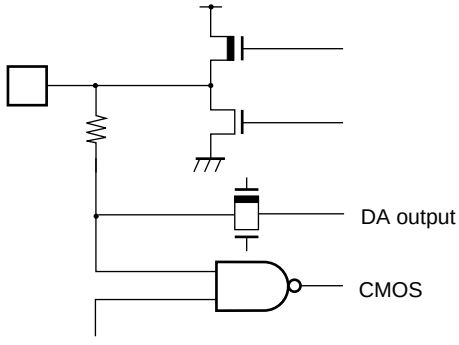
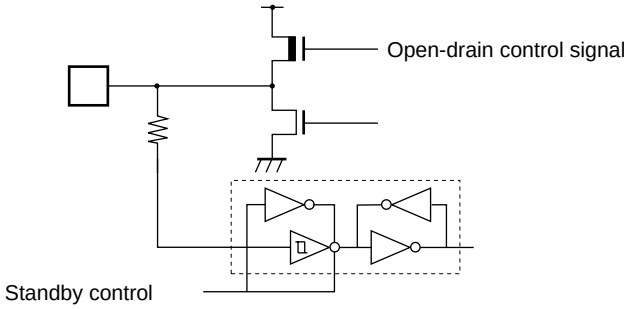
Type	Circuit	Remarks
A		• Oscillation feedback resistor: Approx. 1 MΩ
B		• Hysteresis input with pull-up resistor
C		• CMOS input port
D		• Hysteresis input port
E		• CMOS level output

(Continued)

Type	Circuit	Remarks
F		• CMOS level output • Hysteresis input
G		• Input pull-up resistor control provided • CMOS level input/output
H		• CMOS level input/output • Open-drain control provided

(Continued)

(Continued)

Type	Circuit	Remarks
I		• CMOS level output • Hysteresis input • Open-drain control provided
J		• CMOS level input/output • Analog input
K		• CMOS level input/output • Analog output • Also serving for D/A output
L		• CMOS level output • Hysteresis input • Open-drain control provided

■ HANDLING DEVICES

1. Preventing 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 higher than the voltage wihich shows on “1. Absolute Maximum Ratings” in section “■ Electrical Characteristics” 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.

2. Treatment of Unused Pins

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

3. External Reset Input

To reset the internal circuit by the Low-level input to the $\overline{RST}$ pin, the Low-level input to the $\overline{RST}$ pin must be maintained for at least five machine cycles. Pay attention to it if the chip uses external clock input.

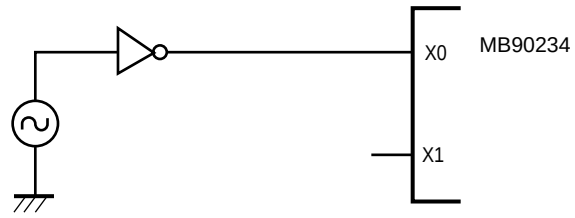
4. V_{CC} and V_{SS} Pins

Apply equal potential to the V_{CC} and V_{SS} pins.

5. Notes on Using an External Clock

When using an external clock, drive the X0 pin as illustrated below:

Use of External Clock



6. Power-on Sequence for A/D Converter Power Supplies and Analog Inputs

Be sure to turn on the digital power supply (V_{CC}) before applying voltage to the A/D converter power supplies (AV_{CC}, AVRH, and AVRL) and analog inputs (AN0 to AN15).

When turning power supplies off, turn off the A/D converter power supplies (AV_{CC}, AVRH, and AVRL) and analog inputs (AN0 to AN15) first, then the digital power supply (AV_{CC}).

When turning AVRH on or off, be careful not to let it exceed AV_{CC}.

7. Pin set when turning on power supplies

When turning on power supplies, set the hardware standby input pin ($\overline{HST}$) to “H”.

8. Program Mode

When shipped from Fujitsu, and after each erasure, all bits (96K × 8 bits) in the MB90W234 and MB90P234 are in the “1” state. Data is introduced by selectively programming “0’s” into the desired bit locations. Bits cannot be set to 1 electrically.

9. Erasure Procedure

Data written in the MB90W234 is erased (from 0 to 1) by exposing the chip to ultraviolet rays with a wavelength of 2,537Å through the translucent cover.

Recommended irradiation dosage for exposure is 10 Wsec/cm^2. This amount is reached in 15 to 20 minutes with a commercial ultraviolet lamp positioned 2 to 3 cm above the package (when the package surface illuminance is 1200 μW/cm^2).

If the ultraviolet lamp has a filter, remove the filter before exposure. Attaching a mirrored plate to the lamp increases the illuminance by a factor of 1.4 to 1.8, thus shortening the required erasure time. If the translucent part of the package is stained with oil or adhesive, transmission of ultraviolet rays is degraded, resulting in a longer erasure time. In that case, clean the translucent part using alcohol (or other solvent not affecting the package).

The above recommended dosage is a value which takes the guard band into consideration and is a multiple of the time in which all bits can be evaluated to have been erased. Observe the recommended dosage for erasure; the purpose of the guard band is to ensure erasure in all temperature and supply voltage ranges. In addition, check the lifespan of the lamp and control the illuminance appropriately.

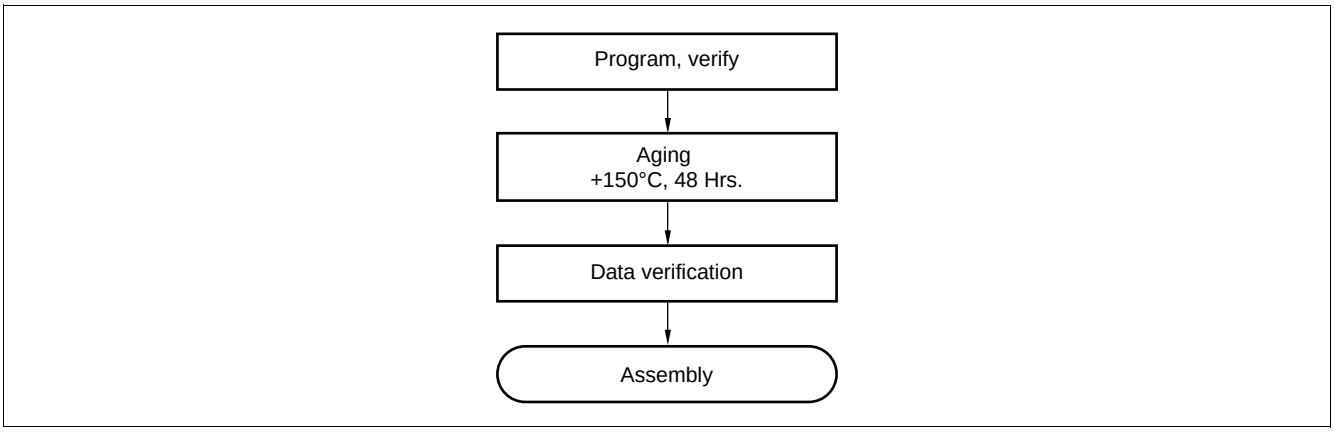
Data in the MB90W234 is erased by exposure to light with a wavelength of 4000Å or less.

Data in the device is also erased even by exposure to fluorescent lamp light or sunlight although the exposure results in a much lower erasure rate than exposure to 2537Å ultraviolet rays. Note that exposure to such lights for an extended period will therefore affect system reliability. If the chip is used where it is exposed to any light with a wavelength of 4000Å or less, cover the translucent part, for example, with a protective seal to prevent the chip from being exposed to the light.

Exposure to light with a wavelength of 4,000 to 5,000Å or more will not erase data in the device. If the light applied to the chip has a very high illuminance, however, the device may cause malfunction in the circuit for reasons of general semiconductor characteristics. Although the circuit will recover normal operation when exposure is stopped, the device requires proper countermeasures for use in a place exposed continuously to such light even though the wavelength is 4,000Å or more.

10. Recommended Screening Conditions

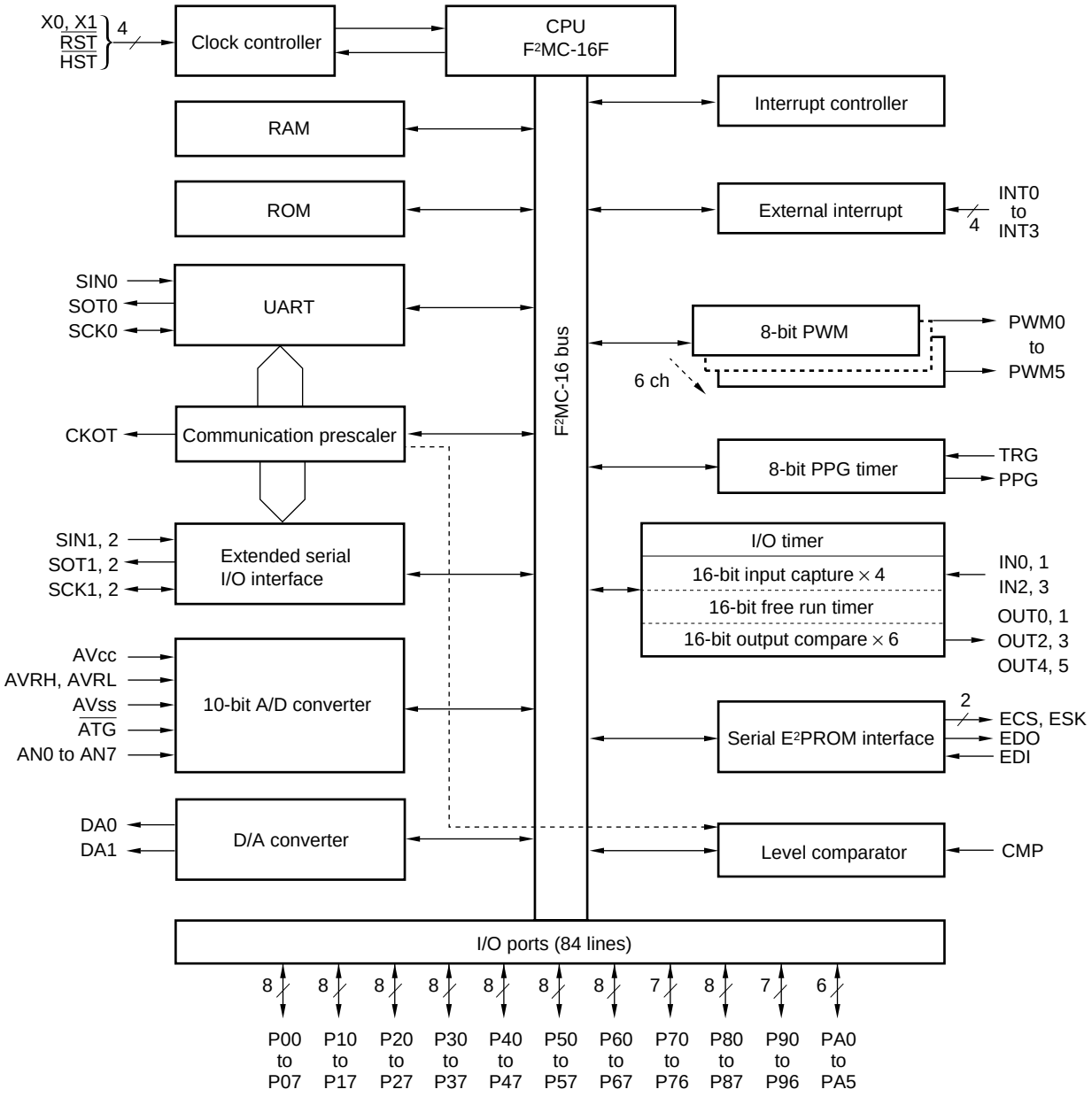
High-temperature aging is recommended for screening before packaging.



11. Write Yield

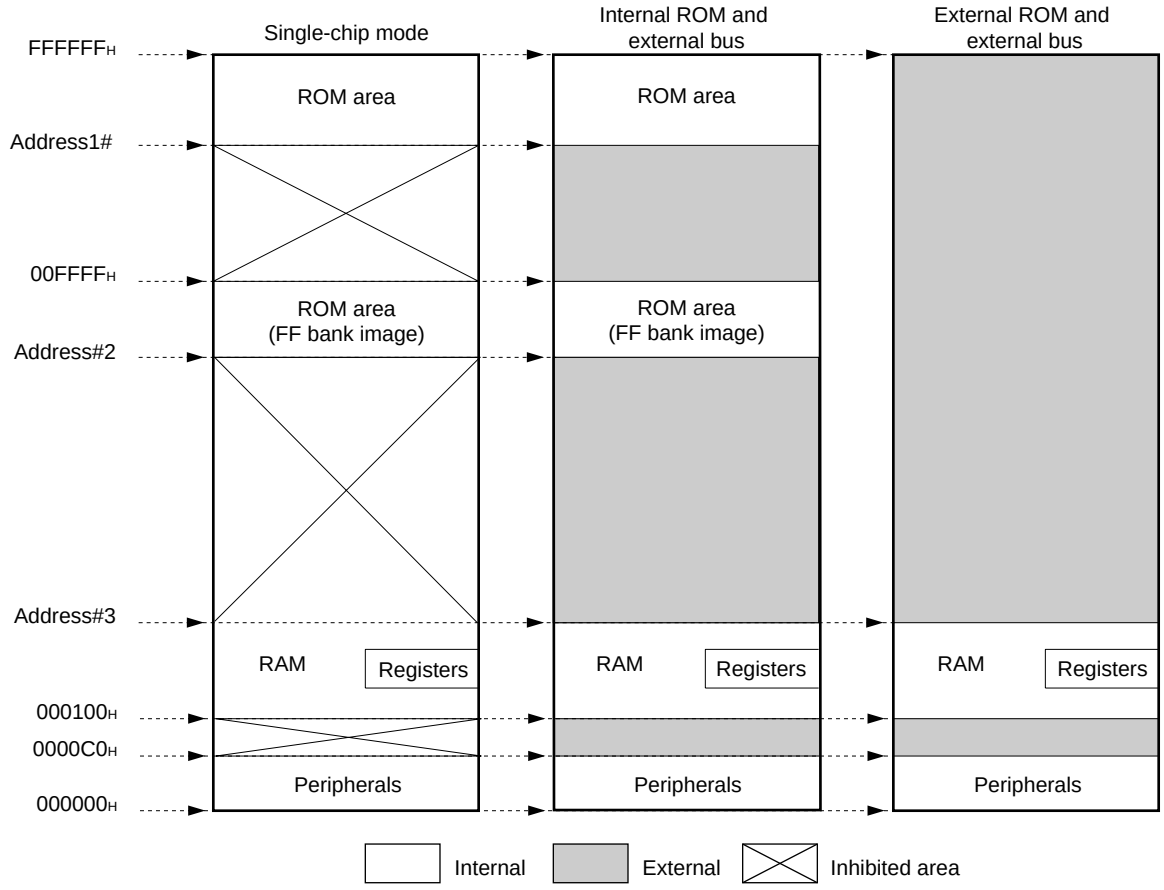
OTPROM products cannot be write-tested for all bits due to their nature. Therefore the write yield cannot always be guaranteed to be 100%.

■ BLOCK DIAGRAM



P00 to P27 (24 lines): Provided with input pull-up resistor setting registers
P94 to P96, PA0 to PA5 (9 lines): Provided with open-drain setting registers

■ MEMORY MAP



Note: 000000H to 000005H and 000010H to 000015H are allocated for external use when the external bus is enabled.

Product type	Address#1	Address#2	Address#3
MB90233	FF4000H	004000H	000900H
MB90234 MB90P234 MB90W234	 FE8000H FE8000H	 004000H 004000H	 000D00H 000D00H
MB90V230	(FE0000H)	(004000H)	(001100H)

The MB90230 series can access the 00 bank to read ROM data written to the upper 48-KB locations in the FF bank. An advantage of reading written to data addresses FFFFFFH-FF4000H from addresses 00FFFFH-004000H is that you can use the small model of a C compiler.
Note, however, that the products with more than 48KB ROM space (MB90V230, MB90P/W234, MB90234) cannot read data in addresses other than FFFFFFH to FF4000H from the 00 bank.

■ I/O MAP

Address	Register	Register name	Access	Resouce name	Initial value
00H	Port 0 data register	PDR0	R/W	Port 0	X X X X X X X X
01H	Port 1 data register	PDR1	R/W	Port 1	X X X X X X X X
02H	Port 2 data register	PDR2	R/W	Port 2	X X X X X X X X
03H	Port 3 data register	PDR3	R/W	Port 3	X X X X X X X X
04H	Port 4 data register	PDR4	R/W	Port 4	X X X X X X X X
05H	Port 5 data register	PDR5	R/W	Port 5	X X X X X X X X
06H	Port 6 data register	PDR6	R/W	Port 6	X X X X X X X X
07H	Port 7 data register	PDR7	R/W	Port 7	– X X X X X X X
08H	Port 8 data register	PDR8	R/W	Port 8	X X X X X X X X
09H	Port 9 data register	PDR9	R/W	Port 9	– X X X X X X X
0AH	Port A data register	PDRA	R/W	Port A	– – X X X X X X
10H	Port 0 direction register	DDR0	R/W	Port 0	0 0 0 0 0 0 0 0
11H	Port 1 direction register	DDR1	R/W	Port 1	0 0 0 0 0 0 0 0
12H	Port 2 direction register	DDR2	R/W	Port 2	0 0 0 0 0 0 0 0
13H	Port 3 direction register	DDR3	R/W	Port 3	0 0 0 0 0 0 0 0
14H	Port 4 direction register	DDR4	R/W	Port 4	0 0 0 0 0 0 0 0
15H	Port 5 direction register	DDR5	R/W	Port 5	0 0 0 0 0 0 0 0
16H	Port 6 direction register	DDR6	R/W	Port 6	0 0 0 0 0 0 0 0
17H	Port 7 direction register	DDR7	R/W	Port 7	– 0 0 0 0 0 0 0
18H	Port 8 direction register	DDR8	R/W	Port 8	0 0 0 0 0 0 0 0
19H	Port 9 direction register	DDR9	R/W	Port 9	– 0 0 0 0 0 0 0
1AH	Port A direction register	DDRA	R/W	Port A	– – 0 0 0 0 0 0
1BH	Port 0 resistor register	RDR0	R/W	Port 0	0 0 0 0 0 0 0 0
1CH	Port 1 resistor register	RDR1	R/W	Port 1	0 0 0 0 0 0 0 0
1DH	Port 2 resistor register	RDR2	R/W	Port 2	0 0 0 0 0 0 0 0
1EH	Port 9 pin register	ODR9	R/W	Port 9	– 0 0 0 – – – –
1FH	Port A pin register	ODRA	R/W	Port A	– – 0 0 0 0 0 0
20H	Mode control register	UMC	R/W	UART	0 0 0 0 0 1 0 0
21H	Status register	USR	R/W		0 0 0 1 0 0 0 0
22H	Serial input register /Serial output register	UIDR /UODR	R/W		X X X X X X X X
23H	Rate and data register	URD	R/W		0 0 0 0 – – 0 0
24H	Serial mode control status register	SMCS	R/W	Extended serial I/O interface	– – – 0 0 0 0 0
25H					0 0 0 0 0 0 1 0

(Continued)

Address	Register	Register name	Access	Resouce name	Initial value
26H	Serial data register	SDR	R/W	Extended serial I/O interface	XXXXXXXX
27H	Reserved area	—	—	—	—
28H	Cycle setting register	PCSR	W	8-bit PPG timer	XXXXXXXX
29H	Duty factor setting register	PDUT	W		XXXXXXXX
2AH	Control status register	PCNTL	R/W		00000000
2BH		PCNTH			0000000–
2CH	Reserved area	—	—	—	—
2DH	Communication prescaler	CDCR	R/W	UART, CKOT, I/O, serial IF	0–––1111
2EH	Clock control register	CLKR	R/W	CKOT output	–––––000
2FH	Level comparator	LVLC	R/W	Level comparator	XXXX0000
30H	Interrupt/DTP enable register	ENIR	R/W	DTP/external interrupt	––––0000
31H	Interrupt/DTP factor register	EIRR	R/W		––––0000
32H	Request level setting register	ELVR	R/W		00000000
33H	Reserved area	—	—	—	—
34H	Analog input enable register	ADER	R/W	10-bit A/D converter	11111111
35H	Reserved area	—	—		—
36H	Control status data register	ADCS0	R/W		00000000
37H		ADCS1			00000000
38H	Data register	ADCR0	R		XXXXXXXX
39H		ADCR1			000000XX
3AH	Reserved area	—	—	—	—
3BH	Reserved area	—	—	—	—
3CH	D/A converter data register 0	DAT0	R/W	8-bit D/A converter	XXXXXXXX
3DH	D/A converter data register 1	DAT1	R/W		00000000
3EH	D/A control register	DACR	R/W		––––––00
3FH	Reserved area	—	—	—	—
40H	PWM data register 0	PWD0	R/W	8-bit PWM0, 1	00000000
41H	PWM data register 1	PWD1	R/W		00000000
42H	Control status data register 0, 1	PWC01	R/W		00000000
43H	Reserved area	—	—	—	—
44H	PWM data register 2	PWD2	R/W	8-bit PWM2, 3	00000000
45H	PWM data register 3	PWD3	R/W		00000000
46H	Control status register 2, 3	PWC23	R/W		00000000

(Continued)

Address	Register	Register name	Access	Resouce name	Initial value
47H	Reserved area	—	—	—	—
48H	PWM data register 4	PWD4	R/W	8-bit PWM4, 5	0 0 0 0 0 0 0 0
49H	PWM data register 5	PWD5	R/W		0 0 0 0 0 0 0 0
4AH	Control status register 4, 5	PWC45	R/W		0 0 0 0 0 0 0 0
4BH	Reserved area	—	—	—	—
4CH	Data register	TCDT	R	16-bit free run timer	0 0 0 0 0 0 0 0
4DH					0 0 0 0 0 0 0 0
4EH	Control status register	TCCS	R/W		0 0 0 0 0 0 0 0
4FH	Reserved area	—	—	—	—
50H	Compare register 0	OCP0	R/W	Output compare 0, 1	X X X X X X X X
51H					X X X X X X X X
52H	Compare register 1	OCP1	R/W		X X X X X X X X
53H					X X X X X X X X
54H	Control status register 0, 1	CS00	R/W		0 0 0 0 – – 0 0
55H		CS01			– – – 0 0 0 0 0
56H	Reserved area	—	—	—	—
57H	Reserved area	—	—	—	—
58H	Compare register 2	OCP2	R/W	Output compare 2, 3	X X X X X X X X
59H					X X X X X X X X
5AH	Compare register 3	OCP3	R/W		X X X X X X X X
5BH					X X X X X X X X
5CH	Control status register 2, 3	CS10	R/W		0 0 0 0 – – 0 0
5DH		CS11			– – – 0 0 0 0 0
5EH	Reserved area	—	—	—	—
5FH	Reserved area	—	—	—	—
60H	Compare register 4	OCP4	R/W	Output compare 4, 5	X X X X X X X X
61H					X X X X X X X X
62H	Compare register 5	OCP5	R/W		X X X X X X X X
63H					X X X X X X X X
64H	Control status register 4, 5	CS20	R/W		0 0 0 0 – – 0 0
65H		CS21			– – – 0 0 0 0 0
66H	Reserved area	—	—	—	—
67H to 6FH	Reserved area	—	—	—	—

(Continued)

Address	Register	Register name	Access	Resouce name	Initial value
70H	Capture register 0	ICP0	R/W	Input capture 0, 1	XXXXXXXX
71H					XXXXXXXX
72H	Capture register 1	ICP1	R/W		XXXXXXXX
73H					XXXXXXXX
74H	Control status register 0, 1	ICS0	R/W		00000000
75H to 77H	Reserved area	—	—	—	—
78H	Capture register 2	ICP2	R/W	Input capture 2, 3	XXXXXXXX
79H					XXXXXXXX
7AH	Capture register 3	ICP3	R/W		XXXXXXXX
7BH					XXXXXXXX
7CH	Control status register 2, 3	ICS1	R/W		00000000
7DH to 7FH	Reserved area	—	—	—	—
80H	OP code register	EOPC	R/W	Serial E²PROM interface	----0000
81H	Format status register	ECTS	R/W		00000000
82H	Data register	EDAT	R/W		XXXXXXXX
83H					XXXXXXXX
84H	Address register	EADR	R/W		00000000
85H					00---000
86H to 8FH	Reserved area	—	—	—	—
90H to 9EH	System reserved area	—	*1	—	—
9FH	Delayed interrupt source generate/ release register	DIRR	R/W	Delayed interrupt generation module	-------0
A0H	Standby control register	STBYC	R/W	Low-power consumption mode	0001XXXX
A1H	Reserved area	—	—	—	—
A2H	Reserved area	—	—	—	—
A3H	Middle address control register	MACR	W	External pin	*2
A4H	Upper address control register	HACR	W	External pin	*2
A5H	External pin control register	EPCR	W	External pin	*2
A6H	Reserved area	—	—	—	—
A7H	Reserved area	—	—	—	—
A8H	Watchdog timer control register	TWC	R/W	Watchdog timer/ reset	XXXXXXXX

(Continued)

Address	Register	Register name	Access	Resouce name	Initial value
$A9_H$	Timebase timer control register	TBTC	R/W	Timebase timer	– – – 0 0 0 0 0
AA_H to AF_H	Reserved area	—	—	—	—
$B0_H$	Interrupt control register 00	ICR00	R/W	Interrupt controller	0 0 0 0 0 1 1 1
$B1_H$	Interrupt control register 01	ICR01	R/W		0 0 0 0 0 1 1 1
$B2_H$	Interrupt control register 02	ICR02	R/W		0 0 0 0 0 1 1 1
$B3_H$	Interrupt control register 03	ICR03	R/W		0 0 0 0 0 1 1 1
$B4_H$	Interrupt control register 04	ICR04	R/W		0 0 0 0 0 1 1 1
$B5_H$	Interrupt control register 05	ICR05	R/W		0 0 0 0 0 1 1 1
$B6_H$	Interrupt control register 06	ICR06	R/W		0 0 0 0 0 1 1 1
$B7_H$	Interrupt control register 07	ICR07	R/W		0 0 0 0 0 1 1 1
$B8_H$	Interrupt control register 08	ICR08	R/W		0 0 0 0 0 1 1 1
$B9_H$	Interrupt control register 09	ICR09	R/W		0 0 0 0 0 1 1 1
BA_H	Interrupt control register 10	ICR10	R/W		0 0 0 0 0 1 1 1
BB_H	Interrupt control register 11	ICR11	R/W		0 0 0 0 0 1 1 1
BC_H	Interrupt control register 12	ICR12	R/W		0 0 0 0 0 1 1 1
BD_H	Interrupt control register 13	ICR13	R/W		0 0 0 0 0 1 1 1
BE_H	Interrupt control register 14	ICR14	R/W		0 0 0 0 0 1 1 1
BF_H	Interrupt control register 15	ICR15	R/W		0 0 0 0 0 1 1 1
$C0_H$ to FF_H	External area	—	—	—	*3

Initial values
0: The initial value for the bit is “0.”
1: The initial value for the bit is “1.”
X: The initial value for the bit is undefined.
–: The bit is not used; the initial value is undefined.
*1: Access inhibited
*2: The initial value depends on each bus mode.
*3: Only this area can be used as the external access area in the area that follows address $0000FF_H$. Access to any address in reserved areas specified in the I/O map table is handled as access to an internal area. An access signal to the external bus is not generated.

■ INTERRUPT VECTORS AND INTERRUPT CONTROL REGISTERS FOR INTERRUPT SOURCES

Interrupt source	I²OS support	Interrupt vector			Interrupt control register	
		No.		Address	ICR	Address
Reset	×	#08	08H	FFFFDCH	—	—
INT9 instruction	×	#09	09H	FFFFD8H	—	—
Exceptional	×	#10	0AH	FFFFD4H	—	—
External interrupt (INT0) 0 ch	○	#11	0BH	FFFFD0H	ICR00	0000B0H
External interrupt (INT1) 1 ch	○	#12	0CH	FFFFCCH		
External interrupt (INT2) 2 ch	○	#13	0DH	FFFFC8H	ICR01	0000B1H
External interrupt (INT3) 3 ch	○	#14	0EH	FFFFC4H		
Extended serial I/O interface	○	#15	0FH	FFFFC0H	ICR02	0000B2H
Serial E²PROM interface	○	#17	11H	FFFFB8H	ICR03	0000B3H
Input capture channel 0	○	#19	13H	FFFFB0H	ICR04	0000B4H
Input capture channel 1	○	#21	15H	FFFFA8H	ICR05	0000B5H
Input capture channel 2	○	#23	17H	FFFFA0H	ICR06	0000B6H
Input capture channel 3	○	#24	18H	FFFF9CH		
Output compare channel 0	○	#25	19H	FFFF98H	ICR07	0000B7H
Output compare channel 1	○	#26	1AH	FFFF94H		
Output compare channel 2	○	#27	1BH	FFFF90H	ICR08	0000B8H
Output compare channel 3	○	#28	1CH	FFFF8CH		
Output compare channel 4	○	#29	1DH	FFFF88H	ICR09	0000B9H
Output compare channel 5	○	#30	1EH	FFFF84H		
16-bit free run timer overflow	○	#31	1FH	FFFF80H	ICR10	0000BAH
Timebase timer overflow	○	#32	20H	FFFF7CH		
8-bit PPG timer	○	#33	21H	FFFF78H	ICR11	0000BBH
Level comparator	○	#34	22H	FFFF74H		
UART reception	◎	#35	23H	FFFF70H	ICR12	0000BCH
UART transmission	◎	#37	25H	FFFF68H	ICR13	0000BDH
End of A/D conversion	○	#39	27H	FFFF60H	ICR14	0000BEH
Delayed interrupt	×	#42	2AH	FFFF54H	ICR15	0000BFH
Stack fault	×	#256	FFH	FFFC00H	—	—

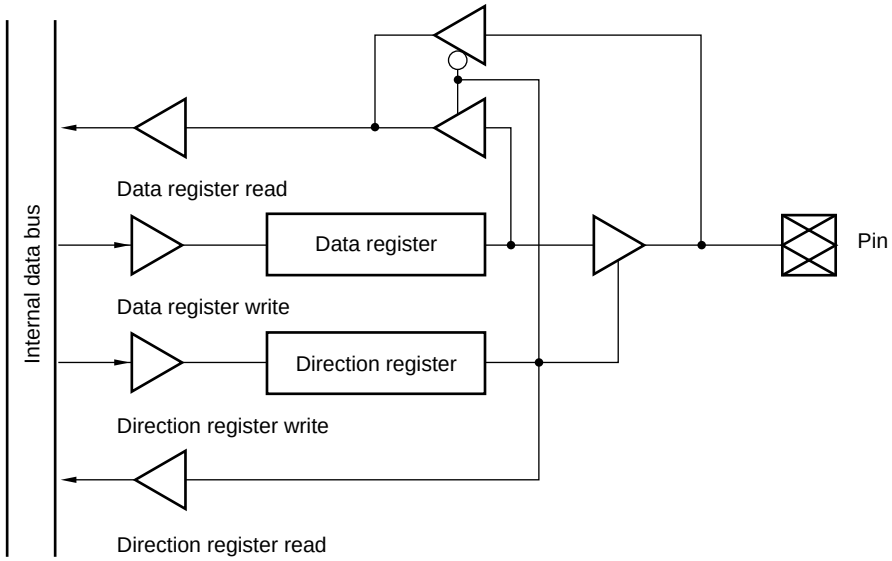
○: The request flag is cleared by the EI²OS interrupt clear signal.
◎: The request flag is cleared by the EI²OS interrupt clear signal. The stop request is available.
×: The request flag is not cleared by the EI²OS interrupt clear signal.

■ PERIPHERAL RESOURCES

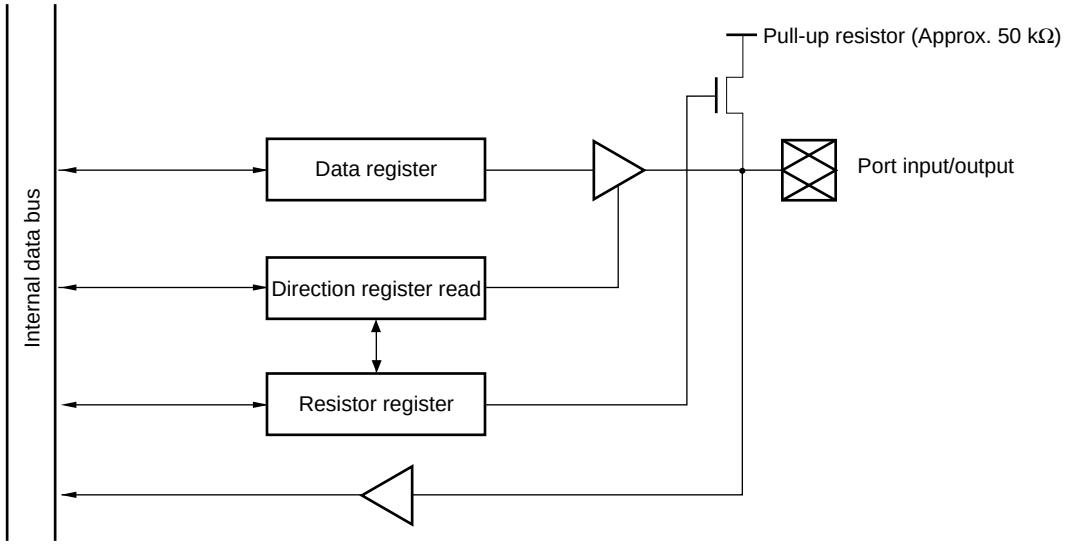
1. I/O Ports

Each pin in each port can be specified for input or output by setting the direction register when the corresponding peripheral resource is not set to use that pin. When the data register is read, the value depending on the pin level is read whenever the pin serves for input. When the data register is read with the pin serving for output, the latch value of the data register is read. This also applies to read operation by the read modify write instruction.

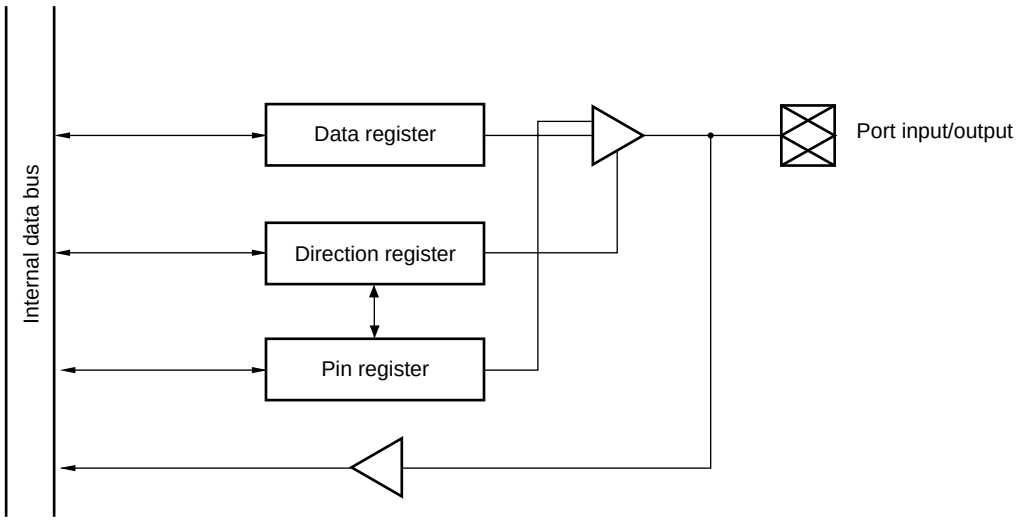
- **General-purpose I/O port**



- **Port with pull-up resistor setting register**



- **Port with open-drain setting register**



(1) Register Configuration

bit	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
Address: 000000H	P07	P06	P05	P04	P03	P02	P01	P00	Port 0 data register (PDR0)
Address: 000001H	P17	P16	P15	P14	P13	P12	P11	P10	Port 1 data register (PDR1)
Address: 000002H	P27	P26	P25	P24	P23	P22	P21	P20	Port 2 data register (PDR2)
Address: 000003H	P37	P36	P35	P34	P33	P32	P31	P30	Port 3 data register (PDR3)
Address: 000004H	P47	P46	P45	P44	P43	P42	P41	P40	Port 4 data register (PDR4)
Address: 000005H	P57	P56	P55	P54	P53	P52	P51	P50	Port 5 data register (PDR5)
Address: 000006H	P67	P66	P65	P64	P63	P62	P61	P60	Port 6 data register (PDR6)
Address: 000007H	—	P76	P75	P74	P73	P72	P71	P70	Port 7 data register (PDR7)
Address: 000008H	P87	P86	P85	P84	P83	P82	P81	P80	Port 8 data register (PDR8)
Address: 000009H	—	P96	P95	P94	P93	P92	P91	P90	Port 9 data register (PDR9)
Address: 00000AH	—	—	PA5	PA4	PA3	PA2	PA1	PA0	Port A data register (PDRA)

bit	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
Address: 000010H	P07	P06	P05	P04	P03	P02	P01	P00	Port 0 direction register (DDR0)
Address: 000011H	P17	P16	P15	P14	P13	P12	P11	P10	Port 1 direction register (DDR1)
Address: 000012H	P27	P26	P25	P24	P23	P22	P21	P20	Port 2 direction register (DDR2)
Address: 000013H	P37	P36	P35	P34	P33	P32	P31	P30	Port 3 direction register (DDR3)
Address: 000014H	P47	P46	P45	P44	P43	P42	P41	P40	Port 4 direction register (DDR4)
Address: 000015H	P57	P56	P55	P54	P53	P52	P51	P50	Port 5 direction register (DDR5)
Address: 000016H	P67	P66	P65	P64	P63	P62	P61	P60	Port 6 direction register (DDR6)
Address: 000017H	—	P76	P75	P74	P73	P72	P71	P70	Port 7 direction register (DDR7)
Address: 000018H	P87	P86	P85	P84	P83	P82	P81	P80	Port 8 direction register (DDR8)
Address: 000019H	—	P96	P95	P94	P93	P92	P91	P90	Port 9 direction register (DDR9)
Address: 00001AH	—	—	PA5	PA4	PA3	PA2	PA1	PA0	Port A direction register (DDRA)

bit	15	14	13	12	11	10	9	8	
Address: 000034H	ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0	Analog input enable register (ADER)

bit	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
Address: 00001BH	P07	P06	P05	P04	P03	P02	P01	P00	Port 0 resistor register (RDR0)
Address: 00001CH	P17	P16	P15	P14	P13	P12	P11	P10	Port 1 resistor register (RDR1)
Address: 00001DH	P27	P26	P25	P24	P23	P22	P21	P20	Port 2 resistor register (RDR2)

bit	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
Address: 00001EH	—	P96	P95	P94	—	—	—	—	Port 9 pin register (ODR9)
Address: 00001FH	—	—	PA5	PA4	PA3	PA2	PA1	PA0	Port A pin register (ODRA)

Ports 0 to 5 in the MB90230 series share the external bus and pins. Each pin function is selected depending on the bus mode and register settings.

Pin name	Function			
	Single-chip mode	External bus extended mode		EPROM write
		8 bits	16 bits	
P07 to P00	Port	D07 to D00		D07 to D00
P17 to P10		Port	D15 to D08	D15 to D08
P27 to P20		A07 to A00		A07 to A00
P37 to P30		A15 to A08*1		A15 to A08
P47 to P45		A23 to A16*1		A23 to A16
P44				
P43 to P40				
P50		CLK*2		Not used
P51		RDY*2		
P52		$\overline{\text{HAK}}$*2		
P53		HRQ*2		
P54		Port	$\overline{\text{WRH}}$*2	$\overline{\text{CE}}$
P55		$\overline{\text{WR}}$	$\overline{\text{WRL}}$*2	$\overline{\text{OE}}$
P56		$\overline{\text{RD}}$		$\overline{\text{PGM}}$
P57		Port		“0”

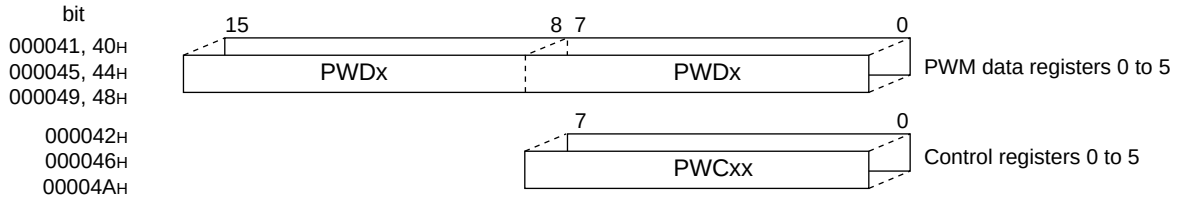
*1: The pin can be used as an I/O port by setting the upper and middle address control registers.
*2: The pin can be used as an I/O port by setting the external pin control register.

2. 8-bit PWM (with 6 channels in this series)

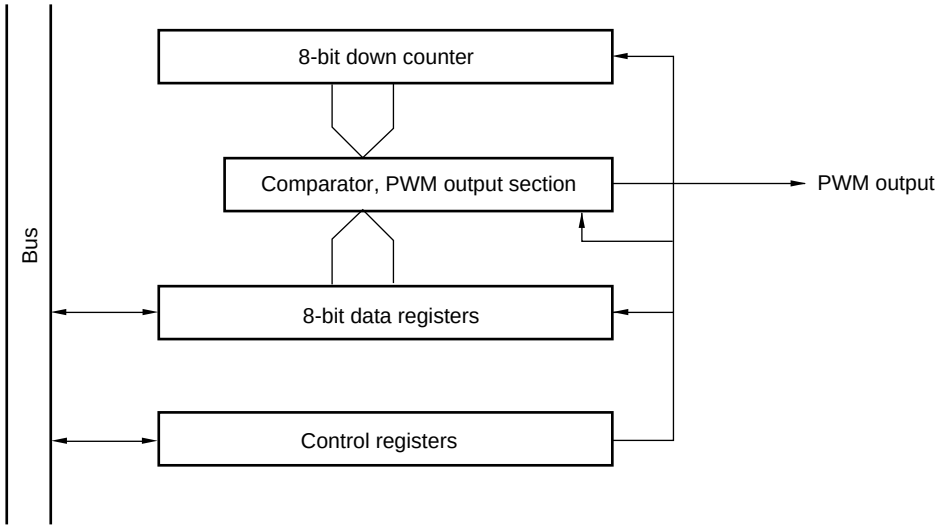
The PWM module consists of a pair of 8-bit PWM output circuits. The MB90230 series incorporates a set of three PWM modules. They can output a waveform continuously from the port at an arbitrary duty factor according to the register settings.

- 8-bit down counter
- 8-bit data registers
- Compare circuit
- Control registers

(1) Register Configuration



(2) Block Diagram

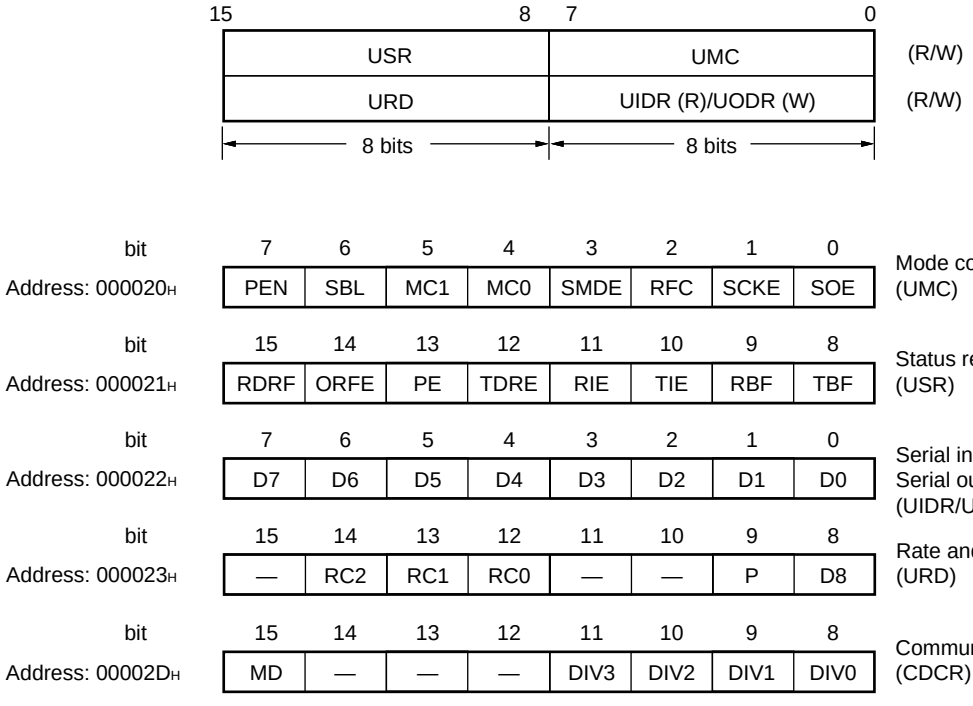


3. UART

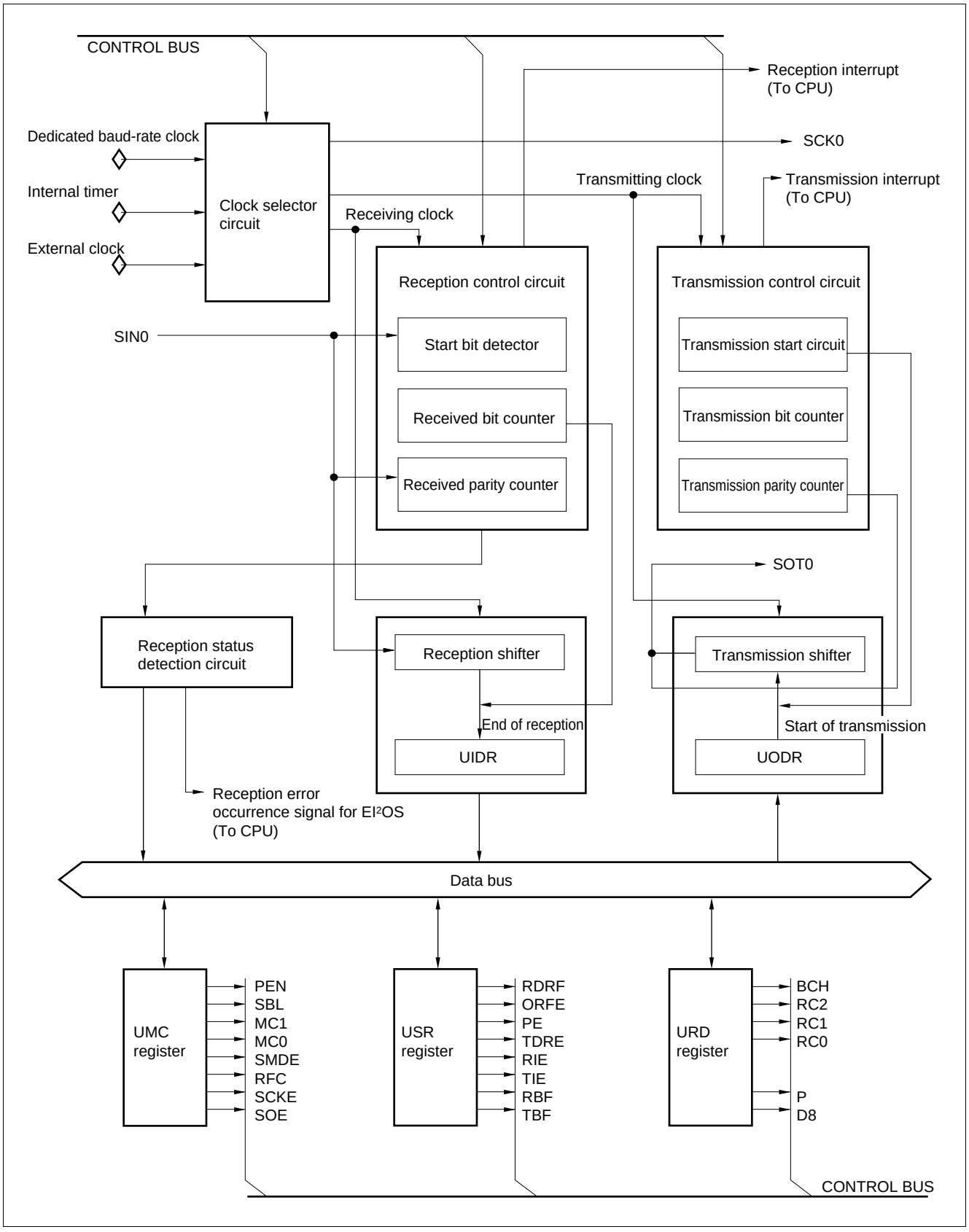
The UART is a serial I/O port for synchronous or asynchronous communication with external resources. It has the following features:

- Full-duplex double buffering
- Data transfer synchronous or asynchronous with clock pulses
- Multiprocessor mode support (Mode 2)
- Internal dedicated baud-rate generator
- Arbitrary baud-rate setting from external clock input or internal timer
- Variable data length (7 to 9 bits (without parity bit); 6 to 8 bits (with parity bit))
- Error detection function (Framing, overrun, parity)
- Interrupt function (Two sources for transmission and reception)
- Transfer in NRZ format

(1) Register Configuration



(2) Block Diagram



4. Extended Serial I/O Interface

This block is a serial I/O interface implemented on a single 8-bit channel that can transfer data in synchronization with clock pulses. It allows the “LSB first” or “MSB first” option to be selected for data transfer. The serial I/O port to be used can also be selected.

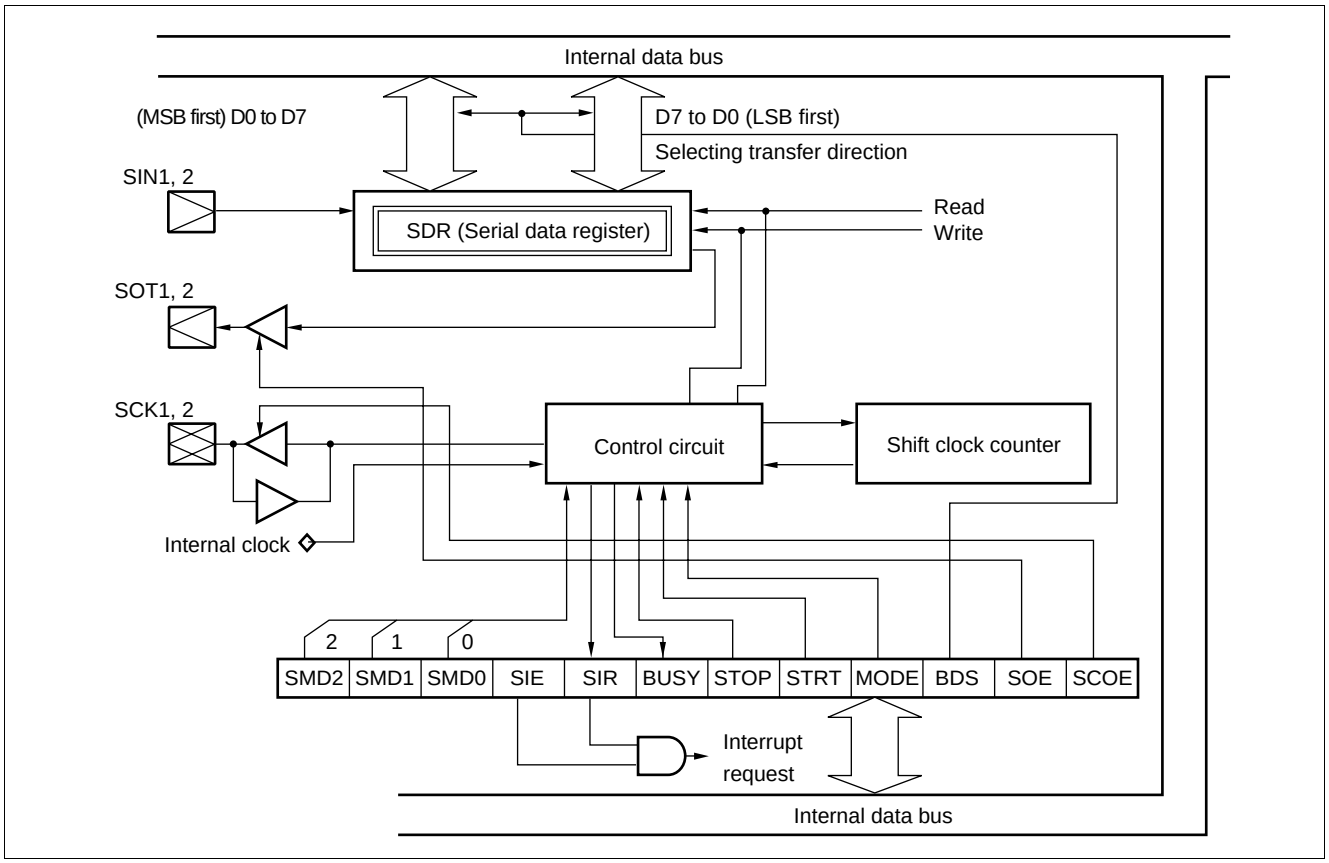
There are two serial I/O operation modes available:

- Internal shift clock mode: Transfers data in synchronization with internal clock pulses.
- External shift clock mode: Transfers data in synchronization with clock pulses entered from an external pin (SCKx). In this mode, data can be transferred by instructions from the CPU by operating the general-purpose port that shares the external pin (SCKx).

(1) Register Configuration

	bit	15	14	13	12	11	10	9	8	
Address: 000025H		SMD2	SMD1	SMD0	SIE	SIR	BUSY	STOP	STRT	
	bit	7	6	5	4	3	2	1	0	Serial mode control status register (SMCS)
Address: 000024H		—	—	—	OUTC	MODE	BDS	SOE	SCOE	
	bit	7	6	5	4	3	2	1	0	Serial data register (SDR)
Address: 000026H		D7	D6	D5	D4	D3	D2	D1	D0	

(2) Block Diagram

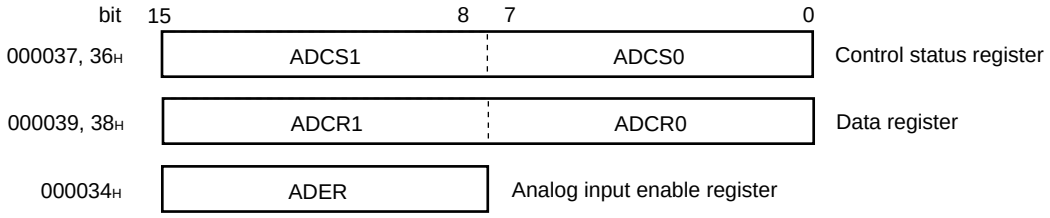


5. A/D Converter

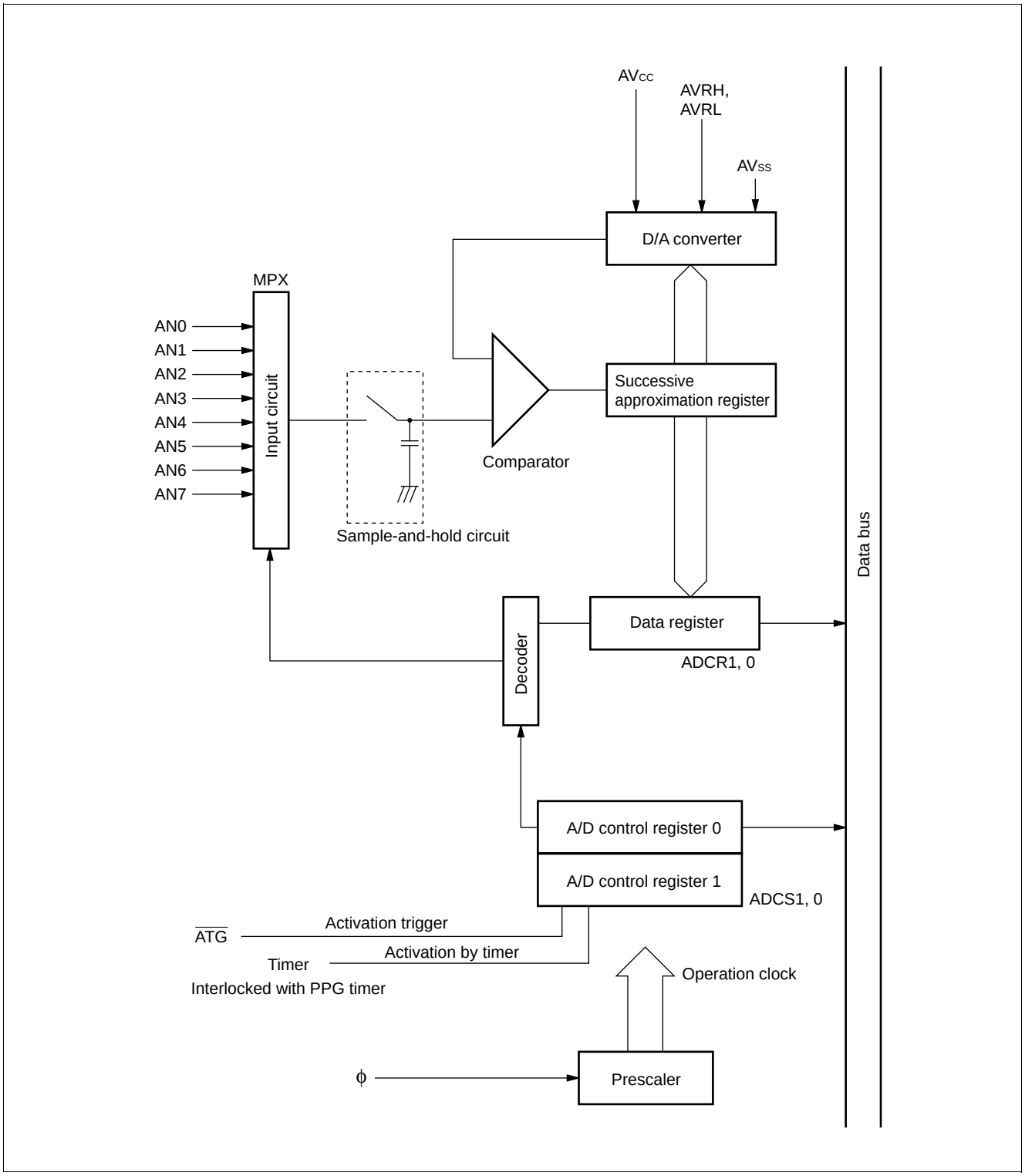
The A/D converter converts the analog input voltage to a digital value. It has the following features:

- Conversion time: 5 μs min. per channel (at 16 MHz machine clock)
- RC-type successive approximation with sample-and-hold circuit
- 8-bit or 10-bit resolution
- Eight analog input channels programmable for selection
- A/D conversion mode selectable from the following three:
  One-shot conversion mode: Converts a specified channel once.
  Consecutive conversion mode: Converts a specified channel repeatedly.
  Stop conversion mode: Converts one channel and suspends its own operation until the next activation (allowing synchronized conversion start).
- Conversion mode:
  Single conversion mode: Converts one channel (when the start and stop channels are the same).
  Scan conversion mode: Converts multiple consecutive channels (when the start and stop channels are different).
- On completion of A/D conversion, the converter can generate an interrupt request for termination of A/D conversion to the CPU. This interrupt generation can activate the EI^{2}OS to transfer the A/D conversion result to memory, making the converter suitable for continuous operation.
- Conversion can be activated by software, external trigger (falling edge), and/or timer (rising edge) as selected.

(1) Register Configuration



(2) Block Diagram



6. 16-bit I/O Timer

The 16-bit I/O timer consists of 16-bit free run timer, 6-line output compare, and 4-line input capture modules.

The 16-bit I/O timer can output six independent waveforms based on the 16-bit free run timer, allowing the input pulse width and external clock cycle to be measured.

(1) Outline of Functions

16-bit free run timer (× 1)

The 16-bit free run timer consists of a 16-bit up-count timer, a control register, and a prescaler. The value output from this timer/counter is used as the base time by the input capture and output compare modules.

- The counter operation clock cycle can be selected from the following four:
  Four internal clock cycles (ϕ/4, ϕ/16, ϕ/32, ϕ/64)
- The interrupt counter value can be generated by compare/match operation with the overflow register and compare register 0 (compare/match operation requires the mode setting).
- The counter value can be initialized to "0000_H" by compare/match operation with the reset register, software clear register, and compare register 0.

Output compare module (× 6)

The output compare module consists of six 16-bit compare registers, compare output latches, and control registers. When the compare value matches the 16-bit free run timer value, this module can generates an interrupt while inverting the output level.

- Six compare registers can operate independently, and have each output pin and interrupt flag.
- Two compare resisters can be used to control the same output pin.
- The initial value for each output pin can be set.
- The interrupt can be generated by compare/match operation.

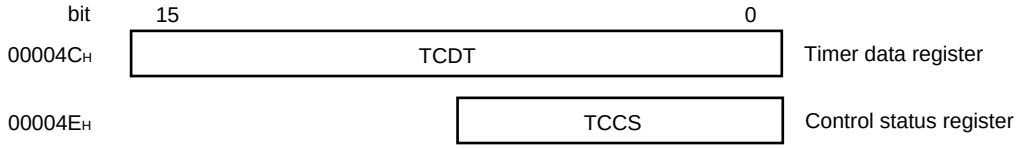
Input capture module (× 4)

The input capture module consists of four external input pins and associated capture and control registers. This module can detect an arbitrary edge of the signal input from each external input pin to generate an interrupt while holding the 16-bit free run timer value in the capture register.

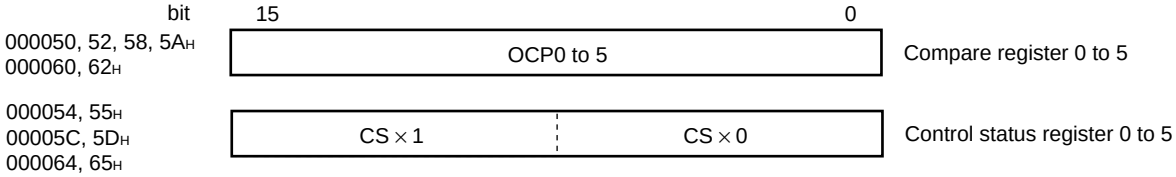
- The external input signal edge can be selected from the rising edge, failing edge or both edges.
- Four input capture lines can operate independently.
- The interrupts can be generated by a valid edge of external input signals. The extended intelligent I/O service (EI^2OS) can be activated.

(2) Register Configuration

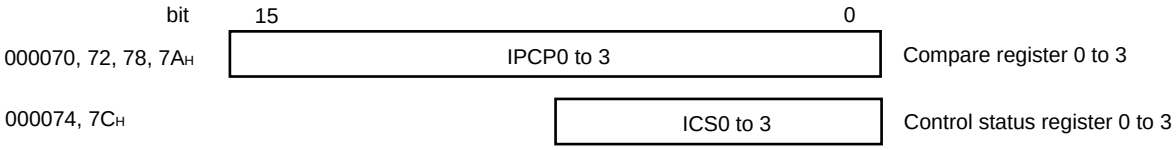
- **16-bit free run timer**



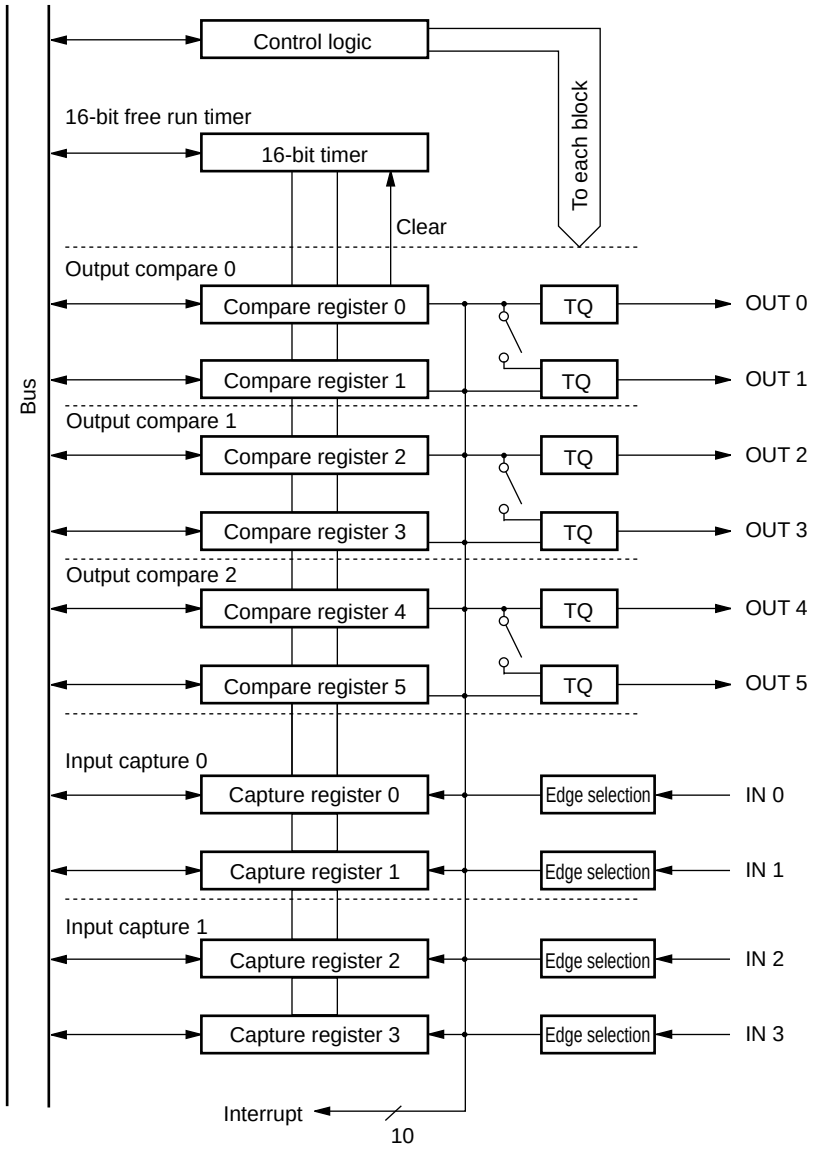
- **16-bit output compare module**



- **16-bit input capture module**



(3) Block Diagram



7. PPG Timer (Programmable Pulse Generator)

This module can output the pulse synchronized with an external or software trigger. The cycle and duty factor of the output pulse can be changed arbitrarily by changing the values in two 8-bit registers.

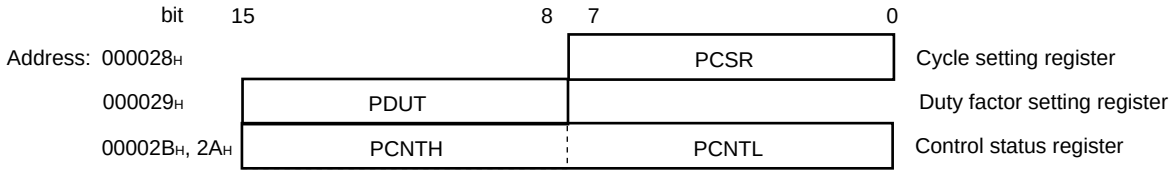
PWM function: Outputs a pulse in programmable mode while changing the values in the two registers in synchronization with the input trigger.
This module can also be used as a D/A converter using an external circuit.

Single-shot function: Detects the trigger input edge to output a single pulse.

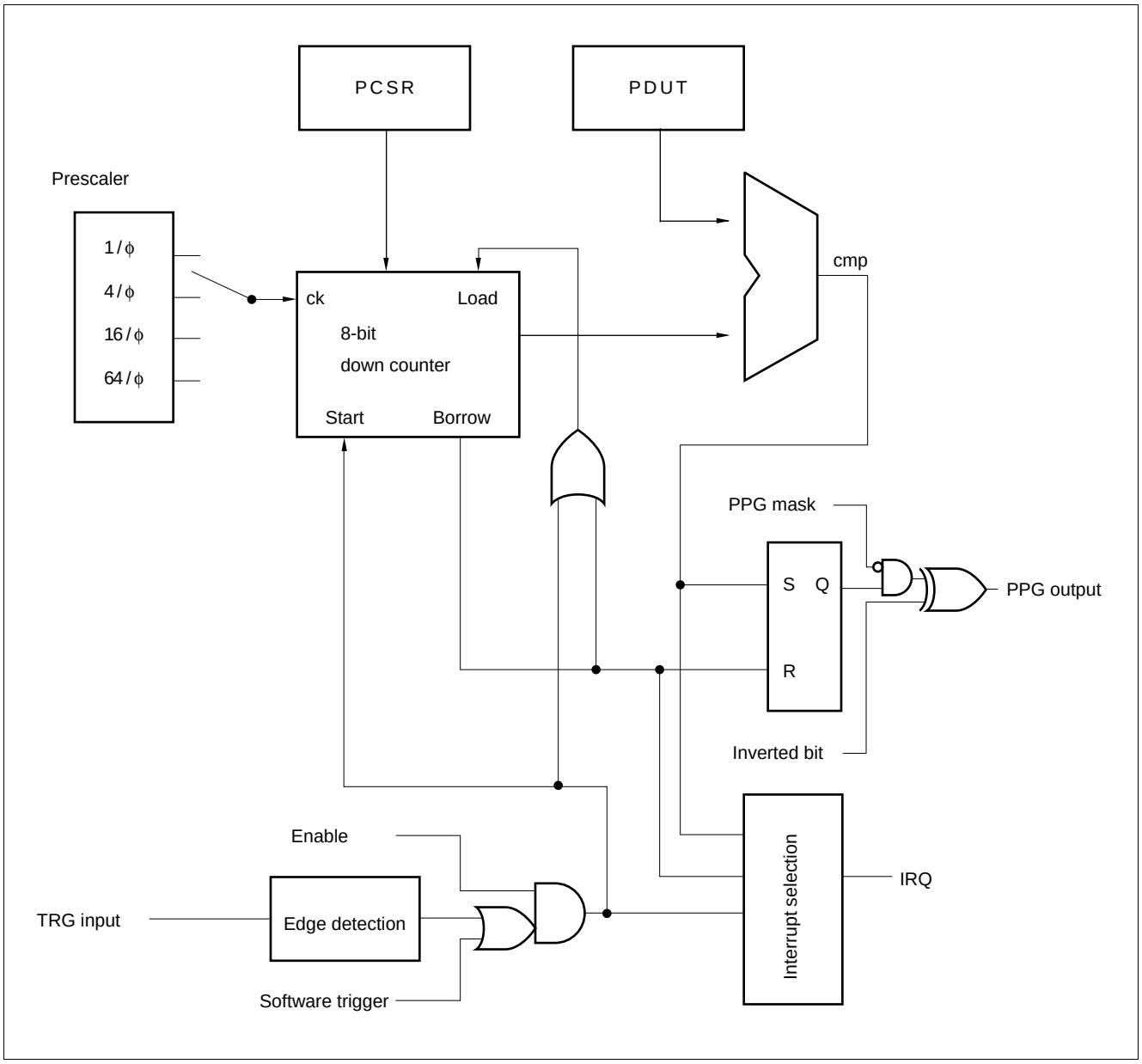
(1) Module Configuration

This module consists of an 8-bit down counter, prescaler, 8-bit cycle setting register, 8-bit duty factor setting register, 16-bit control register, external trigger input pin, and PPG output pin.

(2) Register Configuration



(3) Block Diagram



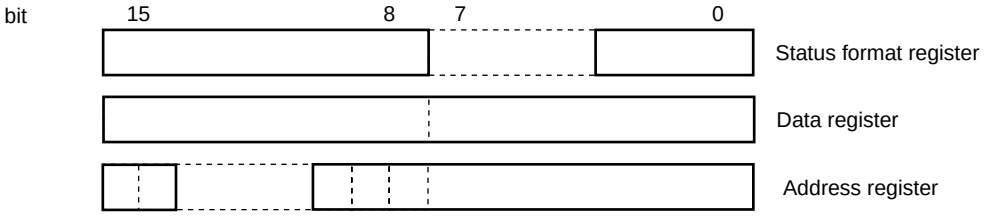
8. Serial E²PROM Interface

This module is the interface circuit dedicated to external bit-serial E²PROM.

(1) Features

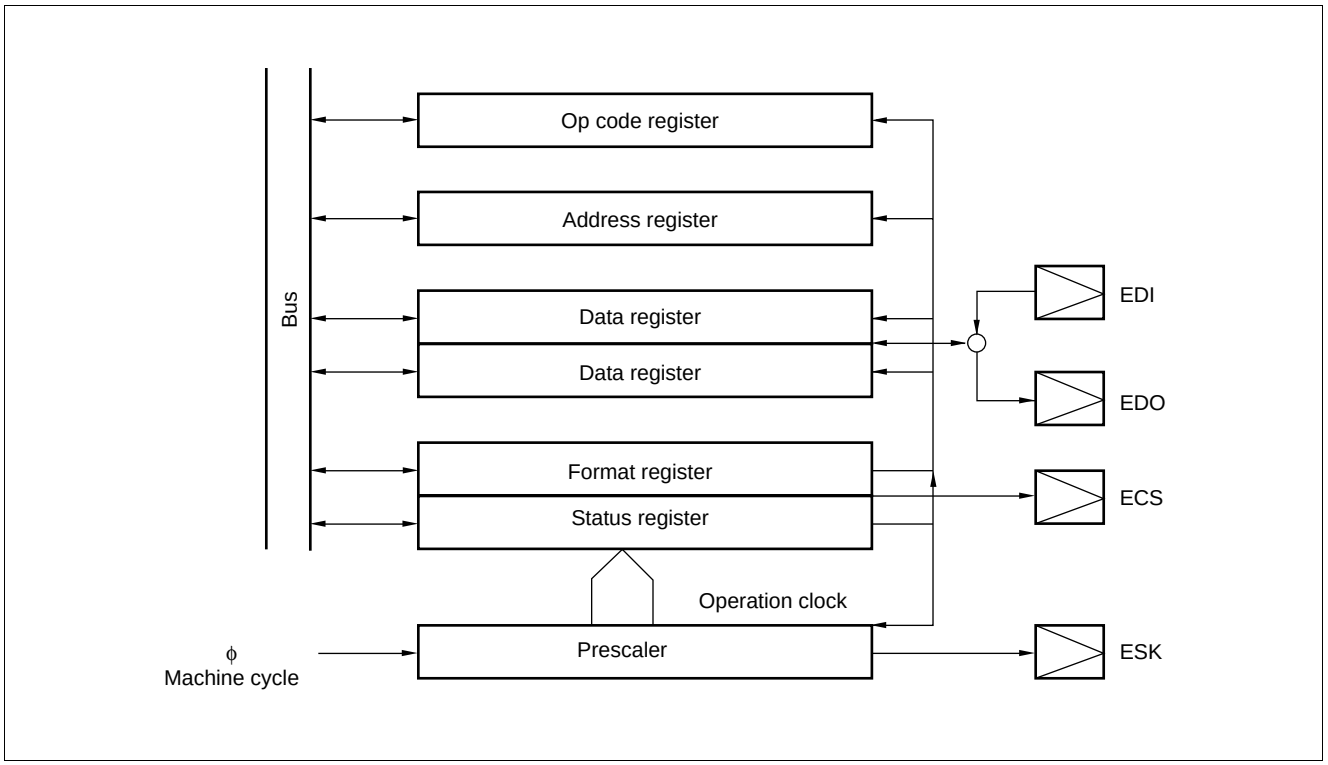
- Instruction code support (compatible with the MB8557).
- Selectable address length: 8 to 11 bits
- Selectable data length: 8 or 16 bits
- Automatic address increment function
- Transmit/receive data transfer enabled by EI²OS
- Up to 2048-by-16 bit access enabled (at an address length of 11 bits and a data length of 16 bits)

(2) Register Configuration



	bit	15	14	13	12	11	10	9	8	
Address: 000081H		IFEN	INT	INTE	BUSY	ADL1	ADL0	DTL	CON	Format status register (ECTS)
	bit	7	6	5	4	3	2	1	0	
Address: 000080H		—	—	—	—	OP3	OP2	OP1	OP0	Op code register (EOPC)
	bit	15	14	13	12	11	10	9	8	
Address: 000083H		D15	D14	D13	D12	D11	D10	D9	D8	Data register (EDAT)
	bit	7	6	5	4	3	2	1	0	
Address: 000082H		D7	D6	D5	D4	D3	D2	D1	D0	Data register (EDAT)
	bit	15	14	13	12	11	10	9	8	
Address: 000085H		CLK	FRQ	—	—	—	A10	A9	A8	Address register (EADR)
	bit	7	6	5	4	3	2	1	0	
Address: 000084H		A7	A6	A5	A4	A3	A2	A1	A0	Address register (EADR)

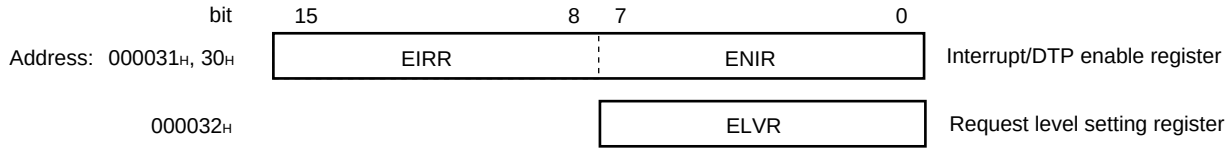
(3) Block Diagram



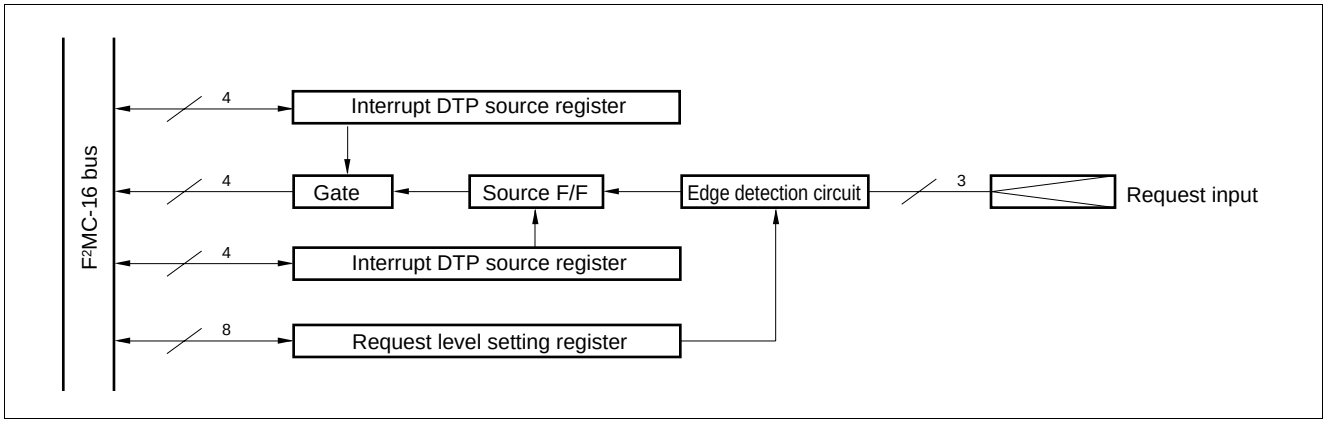
9. DTP/External Interrupt

The data transfer peripheral (DTP) is located between external peripherals and the F²MC-16F CPU. It receives a DMA request or interrupt request generated by the external peripherals and reports it to the F²MC-16F CPU to activate the extended intelligent I/O service or interrupt handler. The user can select two request levels of "H" and "L" for extended intelligent I/O service (EI²OS) or, four request levels of "H," "L," rising edge, and falling edge for external interrupt requests.

(1) Register Configuration



(2) Block Diagram



10. D/A Converter

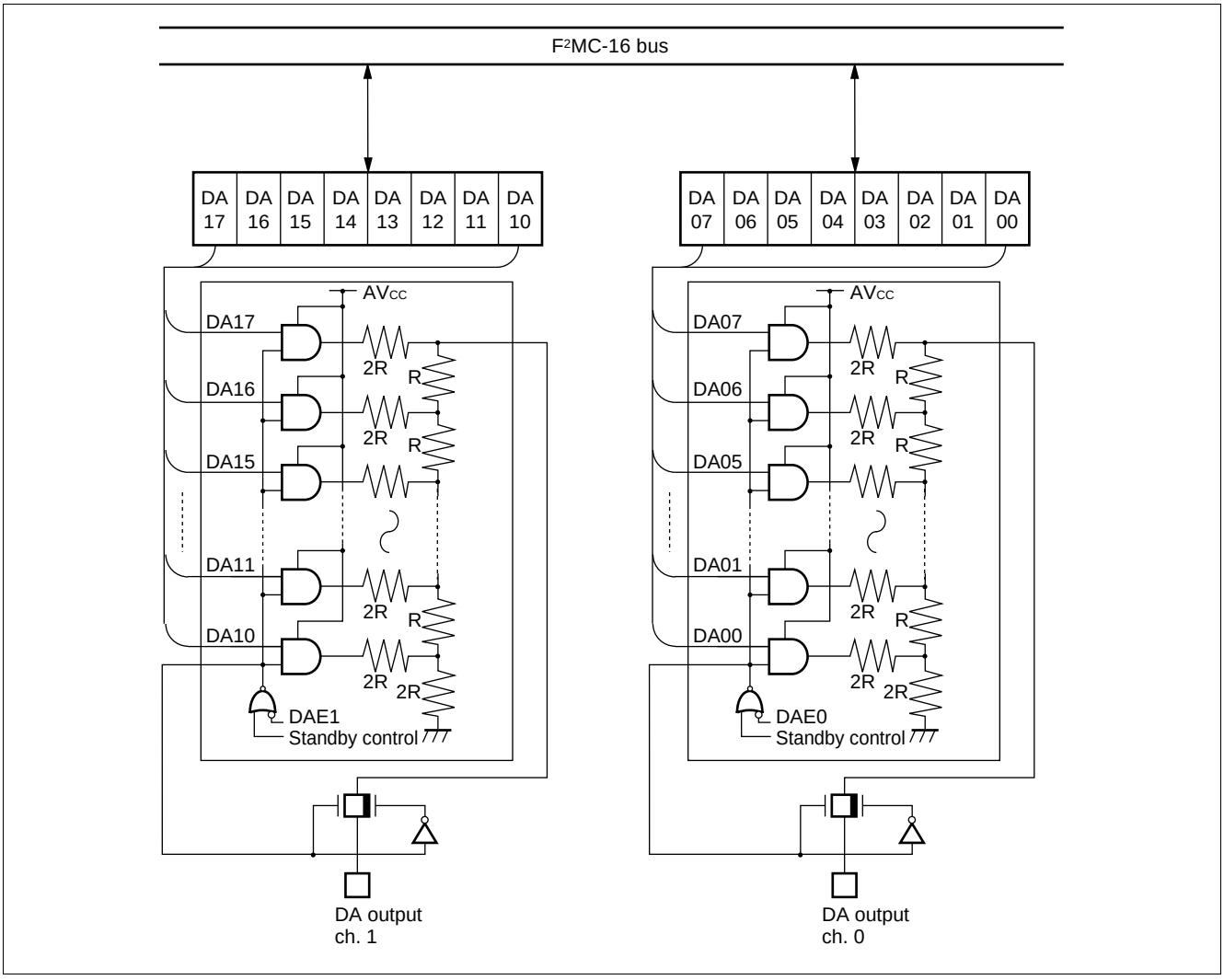
This block is an R-2R type D/A converter with 8-bit resolution.

The D/A converter incorporates two channels, each of which can be controlled for output independently by the D/A control register.

(1) Register Configuration

	bit	15	14	13	12	11	10	9	8	
DAT1 Address: 00003D$_H$		DA17	DA16	DA15	DA14	DA13	DA12	DA11	DA10	D/A converter data register 1
		7	6	5	4	3	2	1	0	
DAT0 Address: 00003C$_H$		DA07	DA06	DA05	DA04	DA03	DA02	DA01	DA00	D/A converter data register 2
		7	6	5	4	3	2	1	0	
DACR Address: 00003E$_H$		—	—	—	—	—	—	DAE1	DAE0	D/A control register

(2) Block Diagram

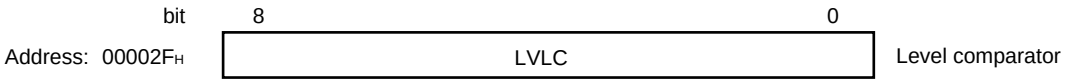


11. Level Comparator

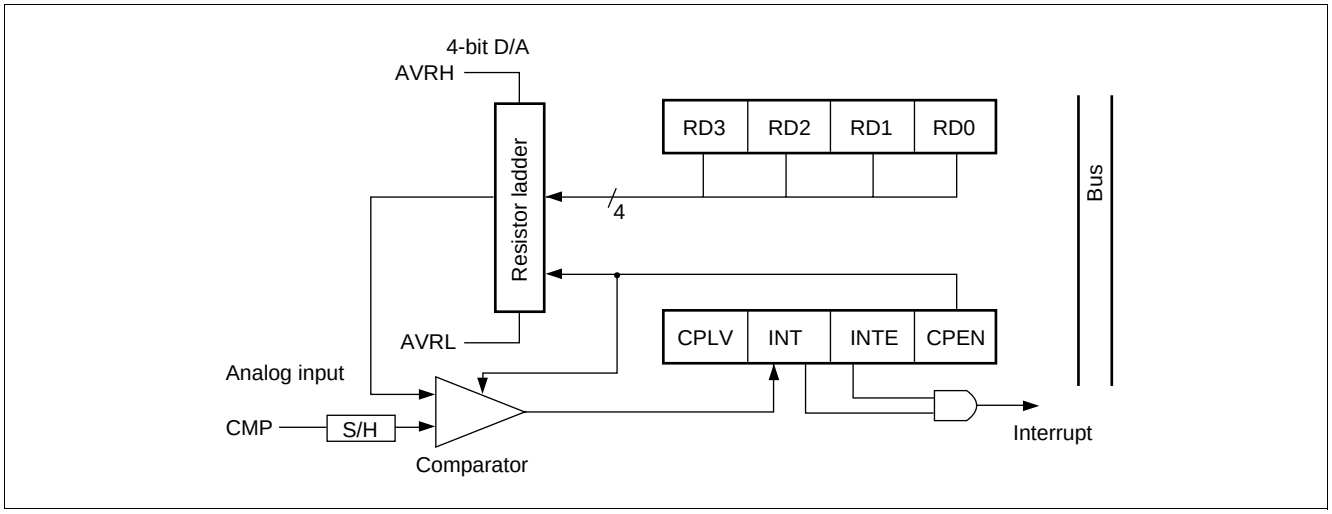
This module compares the input level (by checking whether it is high or low).
The module consists of a comparator, 4-bit resistor ladder, and control register.

- The external input can be compared to the internal 4-bit resistor ladder.

(1) Register Configuration



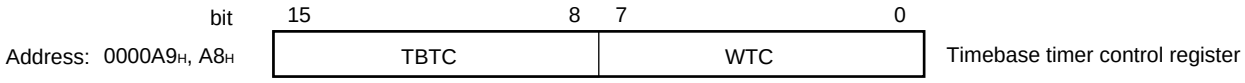
(2) Block Diagram



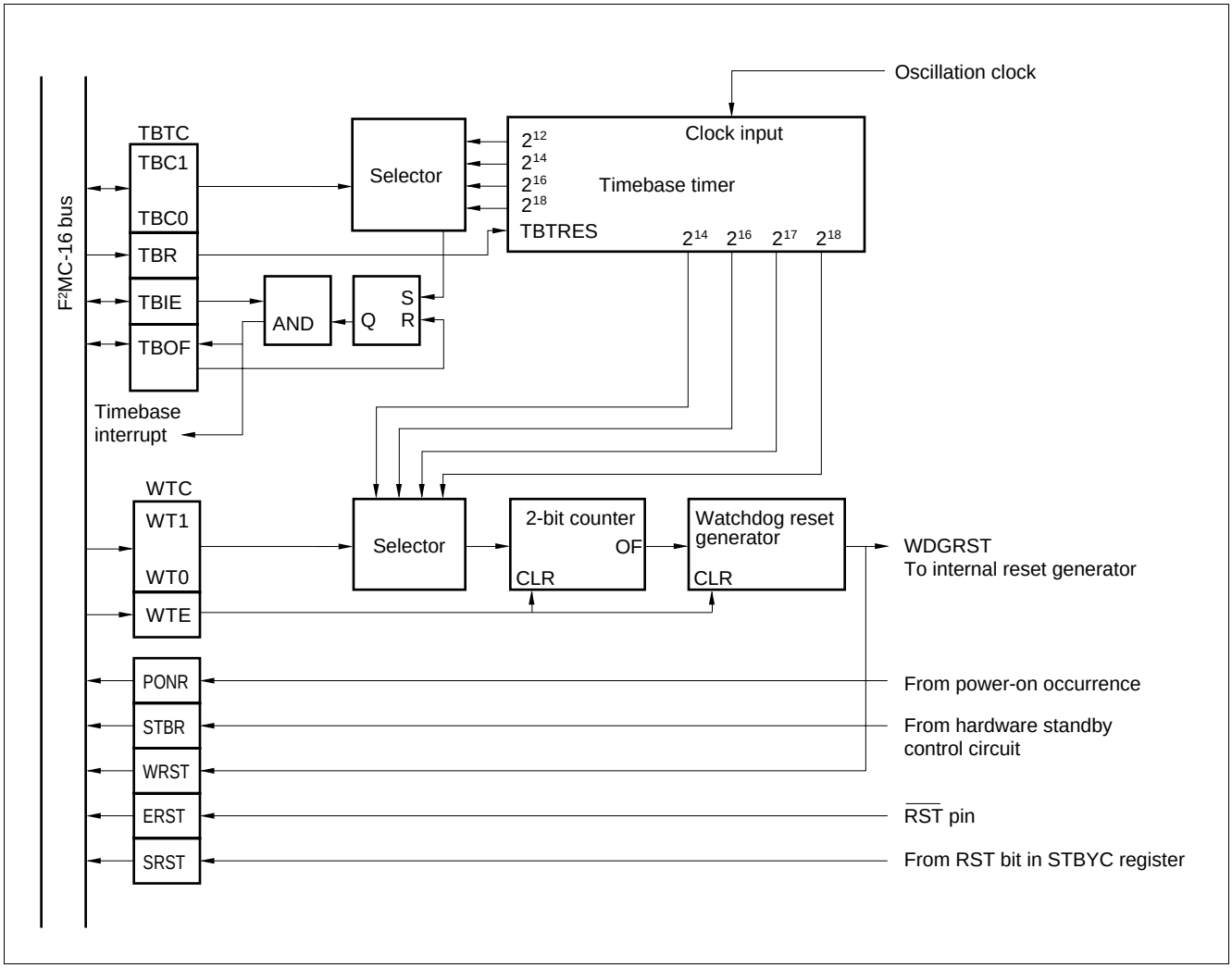
12.Watchdog Timer and Timebase Timer

The watchdog timer consists of a 2-bit watchdog counter using carry signals from an 18-bit timebase counter as the clock source, a control register, and a watchdog reset control section. The timebase timer consists of an 18-bit timer and an interval interrupt control circuit.

(1) Register Configuration



(2) Block Diagram



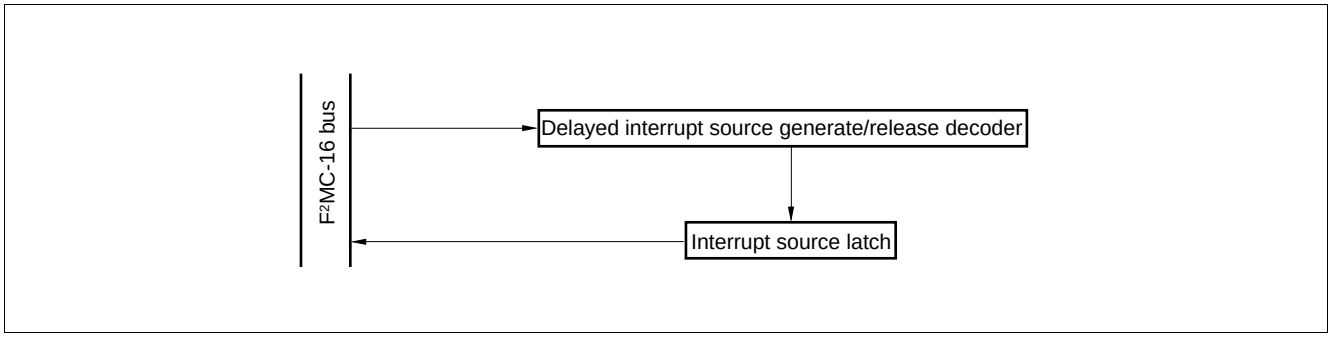
13. Delay Interruupt Generation Module

The delayed interrupt generation module is used to generate an interrupt for task switching. Using this module allows an interrupt request to the F²MC-16F CPU to be generated or canceled by software.

(1) Register Configuration

bit	15	14	13	12	11	10	9	8	
Delayed interrupt source generate/release register									
Address: 00009F$_H$	—	—	—	—	—	—	—	R0	DIRR
Read/write →	(—)	(—)	(—)	(—)	(—)	(—)	(—)	(R/W)	
Initial value →	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(0)	

(2) Block Diagram



14. Clock Output Control Register

The clock output control register outputs the output from the communication prescaler to the pin.

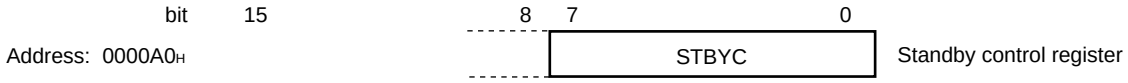
(1) Register Configuration

bit	15	14	13	12	11	10	9	8	
Clock control register									
Address: 00002E$_H$	—	—	—	—	—	CKEN	FRQ1	FRQ0	CLKR
Read/write →	(—)	(—)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	
Initial value →	(—)	(—)	(—)	(—)	(—)	(0)	(0)	(0)	

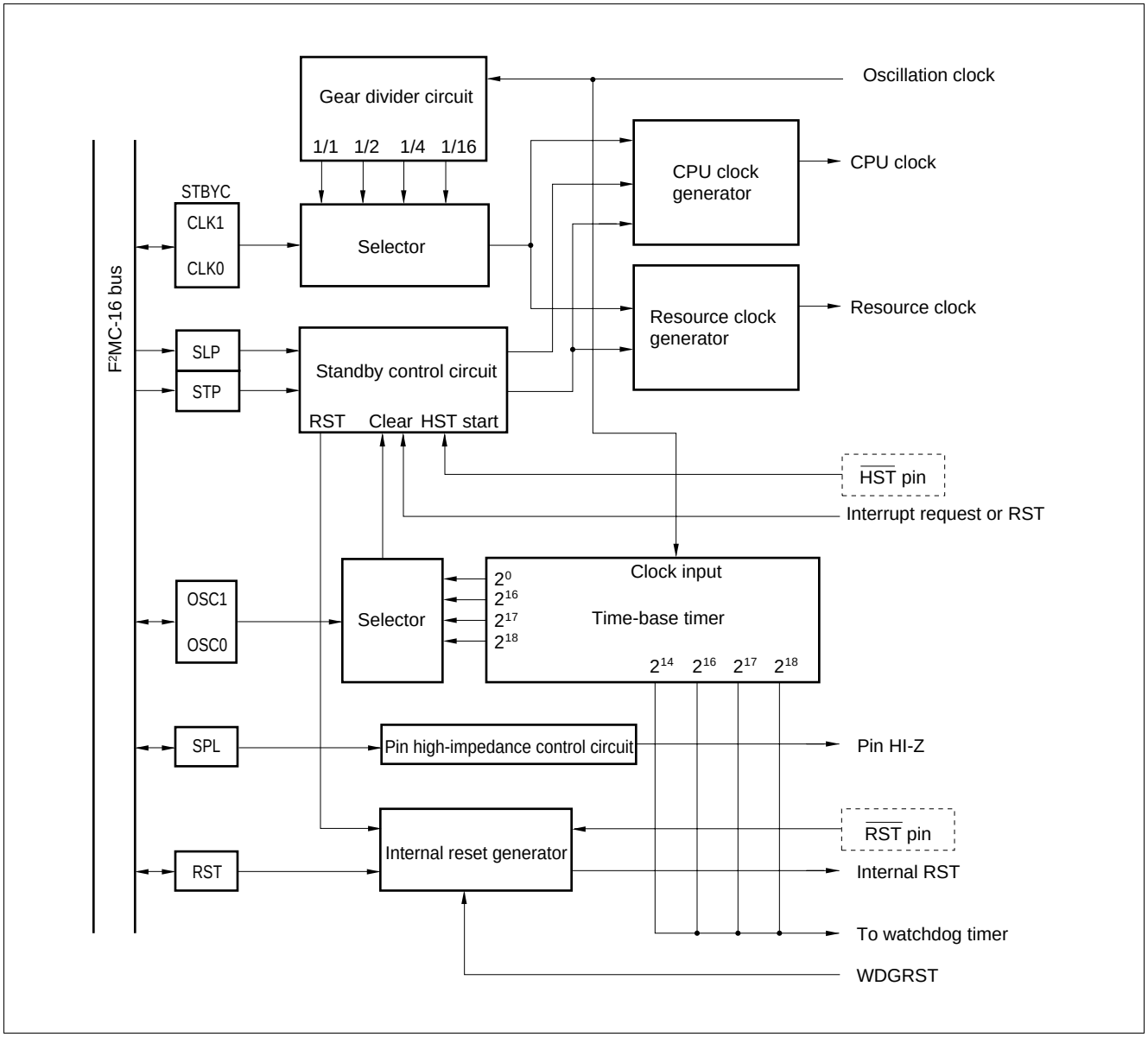
15.Low-power Consumption Control Circuit

The low-power consumption control circuit consists of a low-power consumption control register, clock generator, standby status control circuit, and gear divider circuit. These internal circuits implements the sleep, stop, and hardware standby modes as well as the clock gear function. The gear function allows the machine clock cycle to be selected as a division of the frequency of crystal oscillation or external clock input by 1, 2, 4, or 16.

(1) Register Configuration



(2) Block Diagram



■ ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Ratings

(V_{SS} = 0.0 V)

Parameter	Symbol	Value		Unit	Remarks
		Min.	Max.		
Power supply voltage	V_{CC}	V_{SS} – 0.3	V_{SS} + 7.0	V	
	AV_{CC}, AV_{SS} AVRH, AVRL	V_{CC} – 0.3*1	V_{SS} + 7.0	V	
Input voltage	V_I*2	V_{SS} – 0.3	V_{CC} + 0.3	V	
Output voltage	V_O*2	V_{SS} – 0.3	V_{CC} + 0.3	V	
“L” level output current	I_{OL}	—	20	mA	
“L” level average output current	I_{OLAV}	—	4	mA	
“L” level total output current	ΣI_{OL}	—	50	mA	
“H” level output current	I_{OH}	—	–10	mA	
“H” level average output current	I_{OHAV}	—	–4	mA	
“H” level total output current	ΣI_{OH}	—	–50	mA	
Power consumption	P_D	—	400	mW	
Operating temperature	T_A	–40	+70	°C	
Storage temperature	T_{STG}	–55	+150	°C	

*1: AVRH, AVRL, or AV_{CC} must not exceed V_{CC}.
AV_{SS} and AVRH must not exceed AVRH and AV_{CC}, respectively.
$V_{CC} \geq AV_{CC} \geq AVRH > AVRL \geq AV_{SS} \geq V_{SS}$

*2: V_I or V_O must not exceed “V_{CC} + 0.3 V.”

WARNING: Permanent device damage may occur if the above “Absolute Maximum Ratings” are exceeded. Functional operation should be restricted to the conditions as detailed in the operational sections of this data sheet. Exposure to absolute maximum rating conditions for externded periods may affect device reliability.

2. Recommended Operating Conditions

(V_{SS} = 0.0 V)

Parameter	Symbol	Value		Unit	Remarks
		Min.	Max.		
Power supply voltage	V_{CC}	4.75	5.25	V	During normal operation
		3.0	5.5	V	In stop mode
Operating temperature	T_A	–40	+70	°C	

3. DC Characteristics

(V_{CC} = 5.0 V±5%, V_{SS} = 0.0 V, T_A = –40°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value			Unit	Remarks
				Min.	Typ.	Max.		
"H" level input voltage	V_{IH}	*1	V_{CC} = 5.0 V±5%	0.7 V_{CC}	—	V_{CC} + 0.3	V	
	V_{IHS}	*2		0.8 V_{CC}	—	V_{CC} + 0.3	V	Hysteresis input
	V_{IHM}	*3		V_{CC} – 0.3	—	V_{CC} + 0.3	V	MD0 to 2
"L" level input voltage	V_{IL}	*1	V_{CC} = 5.0 V±5%	V_{SS} – 0.3	—	0.3 V_{CC}	V	
	V_{ILS}	*2		V_{SS} – 0.3	—	0.2 V_{CC}	V	Hysteresis input
	V_{ILM}	*3		V_{SS} – 0.3	—	V_{SS} + 0.3	V	MD0 to 2
"H" level output voltage	V_{OH}	*1, *2	V_{CC} = 4.75 V I_{OH} = –2.0 mA	2.4	—	—	V	
"L" level output voltage	V_{OL}	*1, *2	V_{CC} = 4.75 V I_{OL} = 1.8 mA	—	—	0.4	V	
Input leakage current	I_{IH}	*1, *2, *3	V_{SS} + 4.75 V $<V_I <V_{CC}$	–10	—	10	µA	
Power supply current	I_{CC}	V_{CC}	V_{CC} = 5.0 V±5% fc = 16 MHz	—	48	80	mA	
	I_{CCS}			—	15	25	mA	In sleep mode
	I_{CCH}			—	10	—	µA	In stop mode
Input capacity	C_{IN}	Other than V_{CC} and V_{SS}	—	—	10	—	pF	
Open-drain output leakage current (N-channel Tr OFF)	I_{LEAK}	*4	—	—	0.1	10	µA	
Pull-up current	I_{PULL}	*5	—	–250	—	–50	µA	

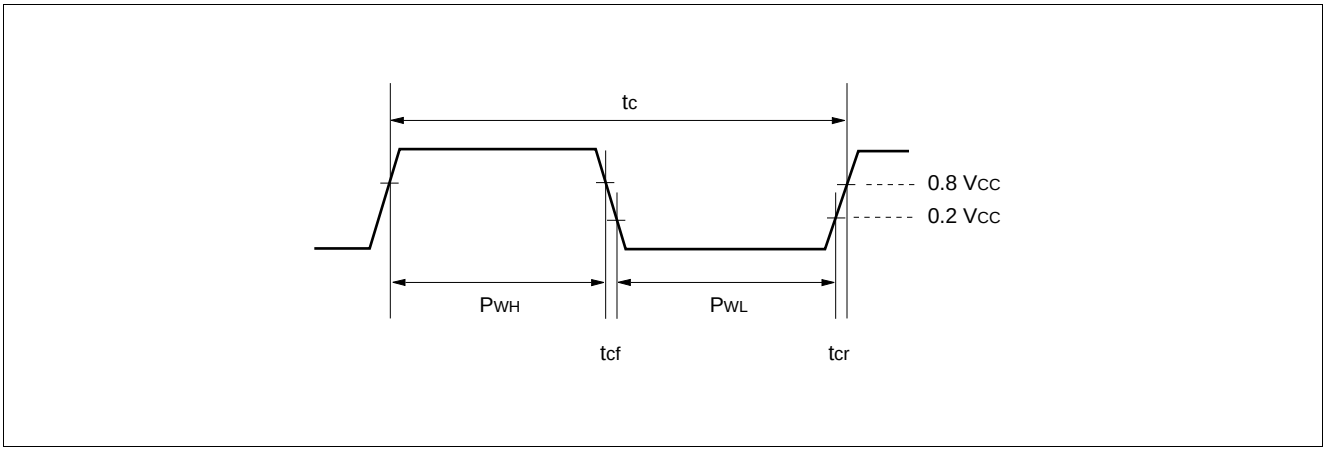
*1: CMOS I/O pin (Other than hysteresis pins)
*2: Hysteresis input pins: P46/TRG, P70/ATG, P71/ESI, P80/INT0, P81/INT1, P82/OUT0/INT2, P83/OUT1/INT3, P90/IN0, P91/IN1, P92/IN2, P93/IN3/CKOT, P94/SIN0, P96/SCK0, PA0/SIN1, PA2/SCK1, PA3/SIN2, PA5/SCK2
*3: Mode pins MD2 to MD0
*4: Open-drain pins P94 to P96 and PA0 to PA5: Set by registers
*5: Pins with pull-up resistor $\overline{RST}$ and P00 to P27: Set by registers

4. AC Characteristics

(1) Clock Timing Standards

(V_{CC} = +5.0 V±5%, V_{SS} = 0.0 V, T_A = –40°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Clock frequency	f_C	X0 X1	V_{CC} = 5.0 V ±5%	1	16	MHz	
Clock cycle time	t_C	X0 X1	V_{CC} = 5.0 V ±5%	62.5	—	ns	
Input clock pulse width	P_{WH} P_{WL}	X0	V_{CC} = 5.0 V ±5%	25.0	—	ns	Duty = 60%
Input clock rising/falling time	t_{cr} t_{cf}	X0	—	5	10	ns	



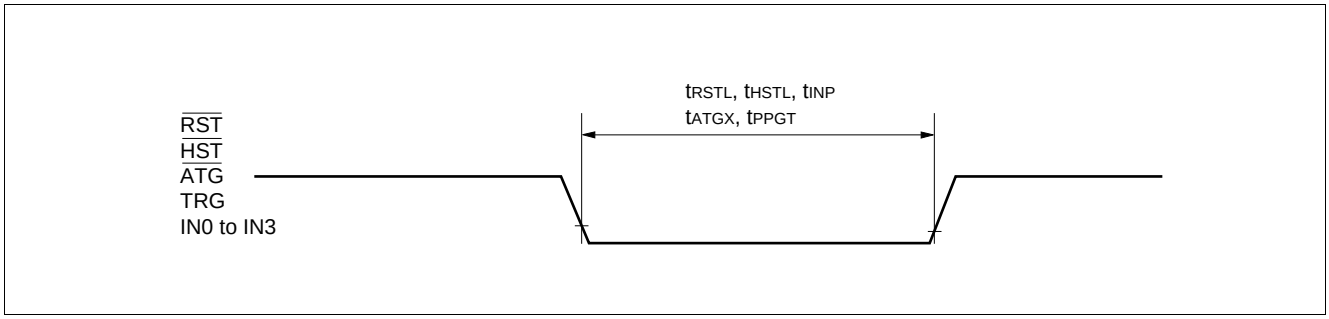
(2) Reset, Hardware Standby, and Trigger Input Standards

(V_{CC} = +5.0 V±5%, V_{SS} = 0.0 V, T_A = –40°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Reset input time	t_{RSTL}	$\overline{RST}$	—	5	—	Machine cycle*	
Hardware standby input time	t_{HSTL}	$\overline{HST}$	—	5	—	Machine cycle*	
A/D start trigger input time	t_{ATGX}	$\overline{ATG}$	—	5	—	Machine cycle*	
PPG start trigger input time	t_{PPGL}	TRG	—	5	—	Machine cycle*	
Input capture input trigger	t_{INP}	IN0 to IN3	—	5	—	Machine cycle*	

*Machine cycle: t_{CYC} = 1/machine clock = 1/(f_C ÷ N)
f_C: Oscillation frequency
N: Gear divide ratio (1, 2, 4, 16)

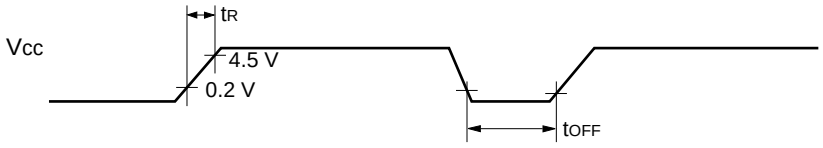
Note: Clock input is required during reset.
The machine cycle at hardware standby input is set to 1/32 divided oscillation.



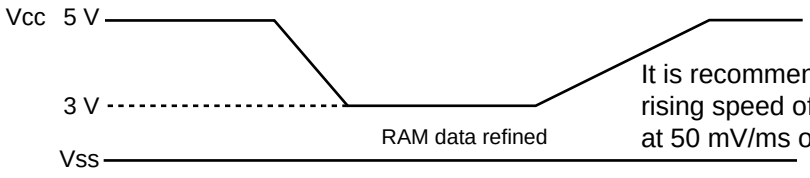
(3) Power-on Reset

(V_{CC} = +5.0 V ±5%, V_{SS} = 0.0 V, T_A = –40°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Power supply riseing time	t_R	V_{cc}	—	—	50	ms	
Power-off time	t_{OFF}			1	—	ms	



Keep in mind that abrupt changes in supply voltage may cause a power-on reset.



It is recommended to keep the rising speed of the supply voltage at 50 mV/ms or slower.

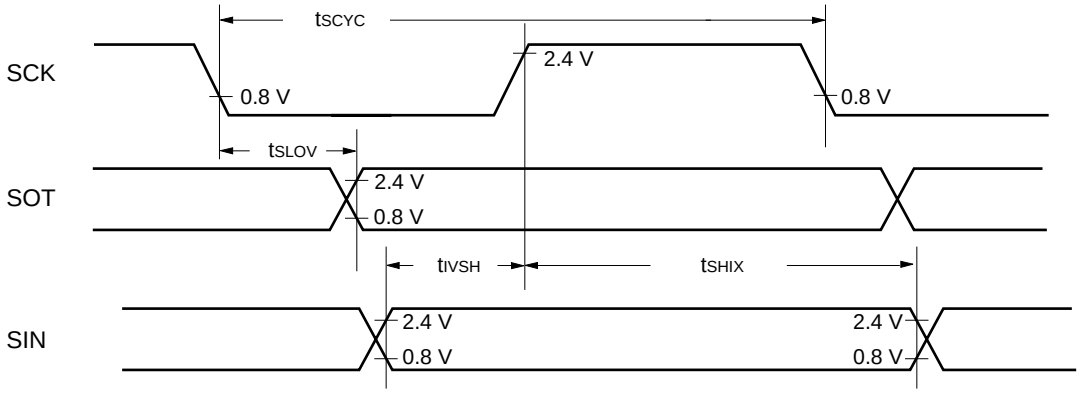
(4) UART Timing

(V_{CC} = +5.0 V±5%, V_{SS} = 0.0 V, T_A = –40°C to +70°C)

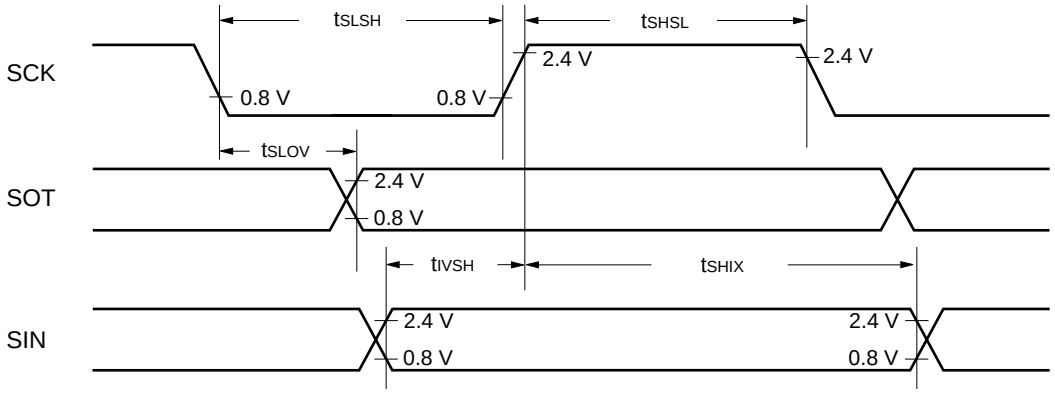
Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Serial clock cycle time	t_{SCYC}	—	Internal clock operation output pin: C_L = 80 pF	8 t_{CYC}	—	ns	
SCK ↓ → SOT delay time	t_{SLOV}	—		–80	80	ns	
Valid SIN → SCK ↑	t_{IVSH}	—		100	—	ns	
SCK ↑ → Valid SIN hold time	t_{SHIX}	—		60	—	ns	
Serial clock “H” pulse width	t_{SHSL}	—	External clock operation output pin: C_L = 80 pF	4 t_{CYC}	—	ns	
Serial clock “L” pulse width	t_{SLSH}	—		4 t_{CYC}	—	ns	
SCK ↓ → SOT delay time	t_{SLOV}	—		—	150	ns	
Valid SIN → SCK ↑	t_{IVSH}	—		60	—	ns	
SCK ↑ → Valid SIN hold time	t_{SHIX}	—		60	—	ns	

Notes: • These AC characteristics assume the CLK synchronous mode.
• C_L is the value for load capacity applied to the pin under testing.
• t_{CYC} is the machine cycle (in nanoseconds).

- **Internal shift clock mode**



- **External shift clock mode**



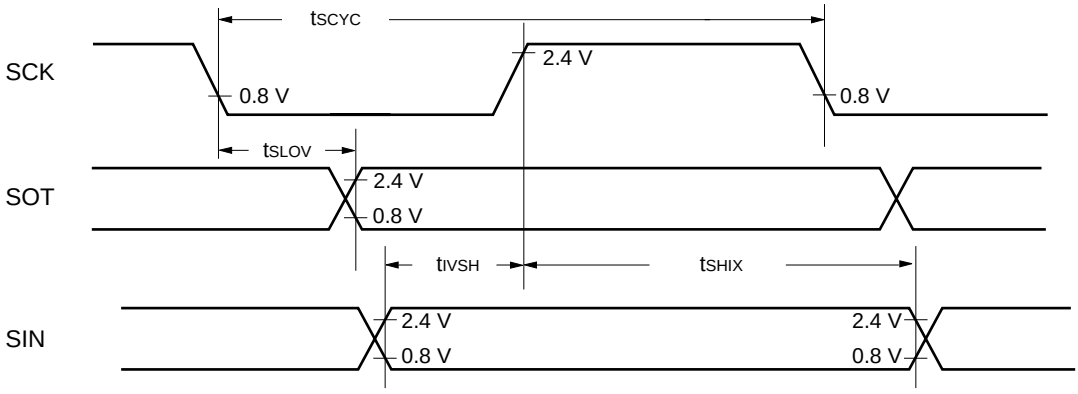
(5) Extended Serial I/O Timing

(V_{CC} = +5.0 V±5%, V_{SS} = 0.0 V, T_A = –40°C to +70°C)

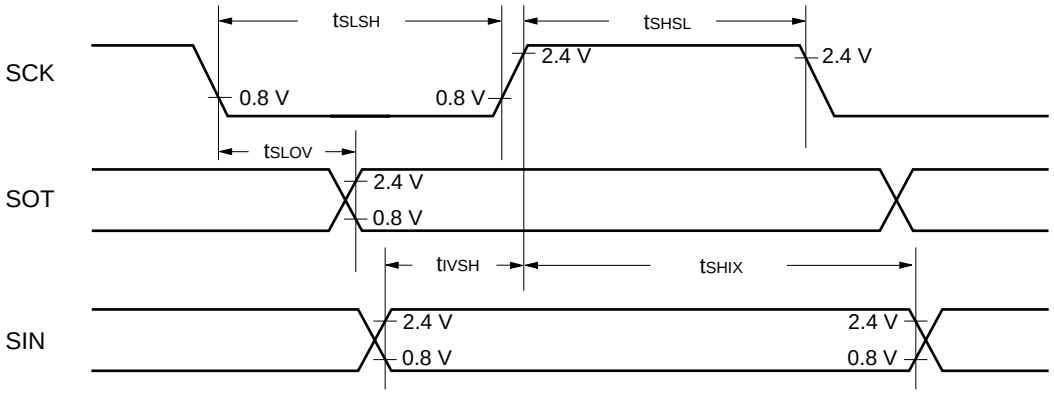
Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Serial clock cycle time	t_{SCYC}	—	Internal clock operation output pin: C_L = 80 pF	8 t_{CYC}	—	ns	
SCK ↓ → SOT delay time	t_{SLOV}	—		50	—	ns	
Valid SIN → SCK ↑	t_{IVSH}	—		1 t_{CYC}	—	ns	
SCK ↑ → Valid SIN hold time	t_{SHIX}	—		1 t_{CYC}	—	ns	
Serial clock “H” pulse width	t_{SHSL}	—	External clock operation output pin: C_L = 80 pF	250	—	ns	External clock: 2 MHz max.
Serial clock “L” pulse width	t_{SLSH}	—		250	—	ns	
SCK ↓ → SOT delay time	t_{SLOV}	—		2 t_{CYC}	—	ns	
Valid SIN → SCK ↑	t_{IVSH}	—		1 t_{CYC}	—	ns	
SCK ↑ → Valid SIN hold time	t_{SHIX}	—		2 t_{CYC}	—	ns	

Notes: • C_L is the value for load capacity applied to the pin under testing.
• t_{CYC} is the machine cycle (in nanoseconds).

- **Internal shift clock mode**



- **External shift clock mode**



5. A/D Converter Electrical Characteristics

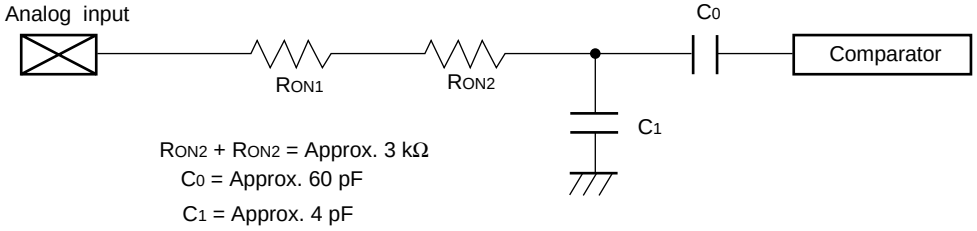
($AV_{CC} = V_{CC} = +5.0\ V \pm 5\%$, $AV_{SS} = V_{SS} = 0.0\ V$, $+3.0\ V \le AVRH - AVRL$, $T_A = -40°C$ to $+70°C$)

Parameter	Symbol	Pin name	Value			Unit
			Min.	Typ.	Max.	
Resolution	—	—	—	10	10	bit
Total error			—	—	±3.0	LSB
Linearity error			—	—	±2.0	LSB
Differential linearity error			—	—	±1.5	LSB
Zero transition voltage	V_{OT}	AN0 to AN7	–1.5	+0.5	+2.5	LSB
Full-scale transition voltage	V_{FST}		AVRH –4.5	AVRH –1.5	AVRH +0.5	LSB
Conversion time	—	f_C = 16 MHz	5.00	—	—	µs
Analog port input current	I_{AIN}	AN0 to AN7	—	—	10	µA
Analog input voltage	—		AVRL	—	AVRH	V
Reference voltage		AVRH	AVRL	—	AV_{CC}	V
		AVRL	0	—	AVRH	V
Power supply current	I_A	AV_{CC}	—	5	—	mA
	I_{AS}		—	—	5*	µA
Reference voltage supply current	I_R	AVRH	—	200	—	µA
	I_{RS}		—	—	5*	µA
Variation between channels	—	AN0 to AN7	—	—	4	LSB

* : Current applied in CPU stop mode with the A/D converter inactive ($V_{CC} = AV_{CC} = AVRH = 5.5\ V$).

Notes:
- The error becomes larger as |AVRH–AVRL| becomes smaller.
- Use the output impedance of the external circuit for analog input under the following conditions: External circuit output impedance < Approx. 7 kΩ
- If the output impedance the external circuit is too high, the analog voltage sampling time may be insufficient. (Sampling time = 3.0 µs at a machine clock frequency of 16 MHz)

- **Analog Input Circuit Mode**

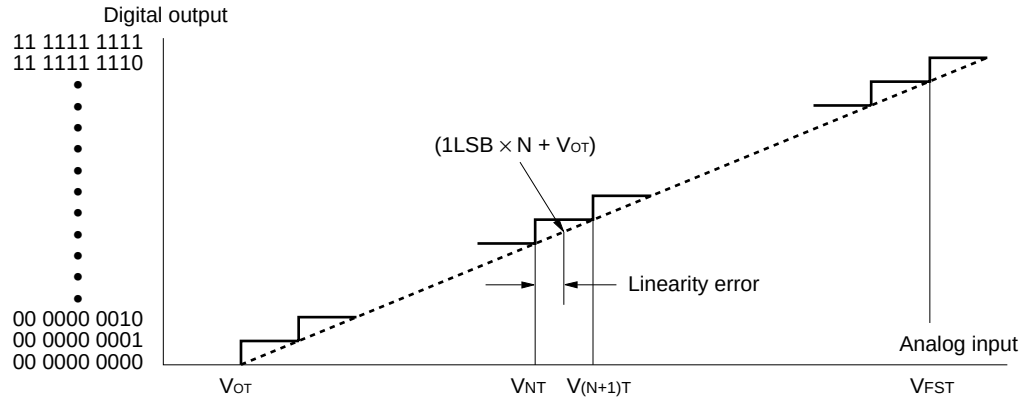


$R_{ON2} + R_{ON2}$ = Approx. 3 kΩ
C_0 = Approx. 60 pF
C_1 = Approx. 4 pF

Note: The values shown here are reference values.

6. A/D Glossary

- Resolution
  Analog changes that are identifiable with the A/D converter.
  When the number of bits is 10, analog voltage can be divided into 2^{10} = 1024
- Total error
  Difference between actual and logical values. This error is caused by a zero transition error, full-scale transition error, linearity error, differential linearity error, or by noise.
- Linearity error
  The deviation of the straight line connecting the zero transition point (“00 0000 0000” ↔ “00 0000 0001”) with the full-scale transition point (“11 1111 1111” ↔ “11 1111 1110”) from actual conversion characteristics
- Differential linearity error
  The deviation of input voltage needed to change the output code by 1 LSB from the theoretical value



$$1LSB = \frac{V_{FST} - V_{OT}}{1022}$$

$$\text{Linearity error} = \frac{V_{NT} - (1LSB \times N + V_{OT})}{1LSB} \quad (LSB)$$

$$\text{Differential linearity error} = \frac{V_{(N+1)T} - V_{NT}}{1LSB} - 1 \quad (LSB)$$

7. D/A Converter Electrical Characteristics

(AV_{CC} = V_{CC} = +5.0 V±5%, AV_{SS} = V_{SS} = 0.0 V, T_A = –40°C to +70°C)

Parameter	Symbol	Pin name	Value			Unit
			Min.	Typ.	Max.	
Resolution	—	—	—	8	8	bit
Differential linearity error	—	—	—	—	±0.9	LSB
Conversion time	—	—	—	10*	20*	µs
Analog output impedance	—	—	—	28	—	KΩ

*: A load capacity of 20 pF is assumed.

8. Serial E²PROM Interface Timing

(1) E²PROM interface at an operation clock frequency of 1 MHz

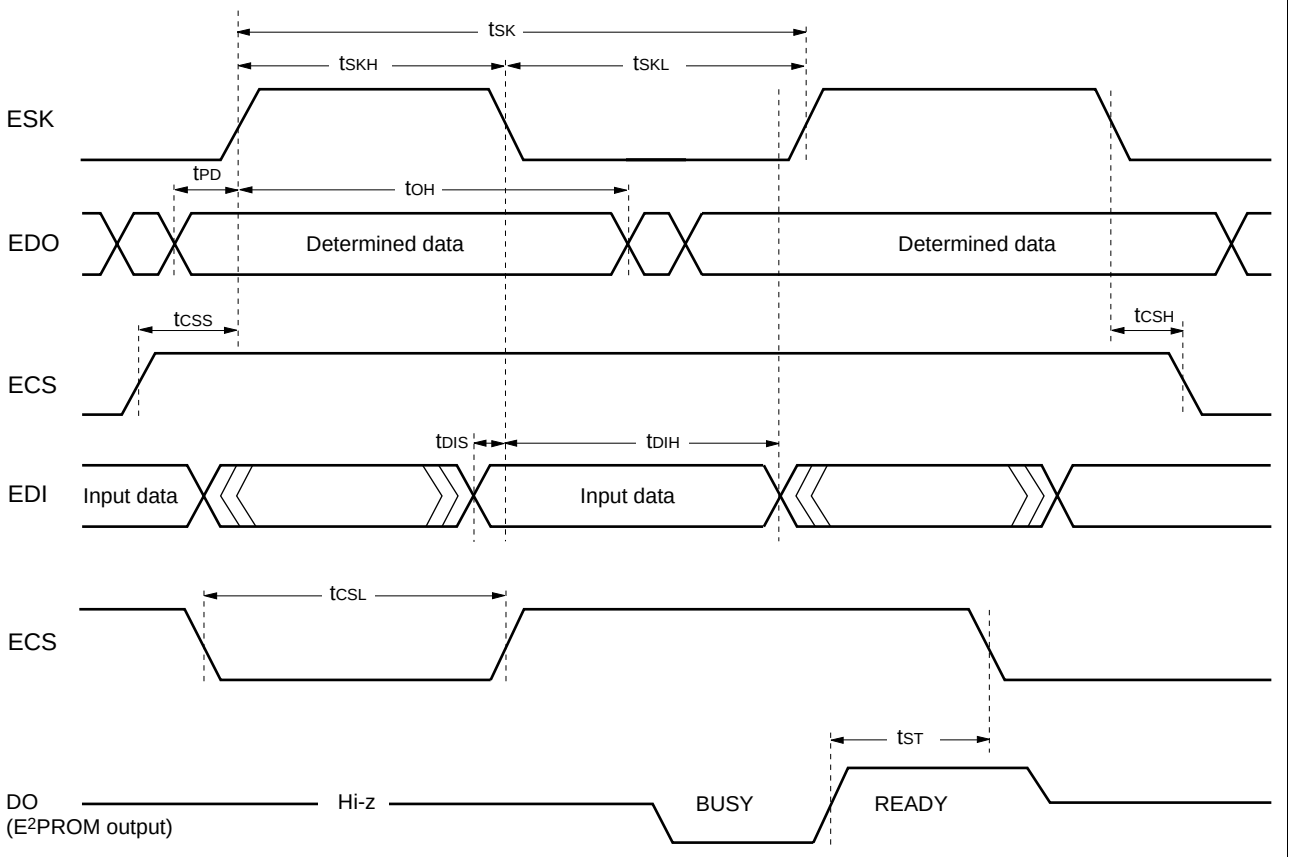
(V_{CC} = +5.0 V±5%, V_{SS} = 0.0 V, T_A = –40°C to +70°C)

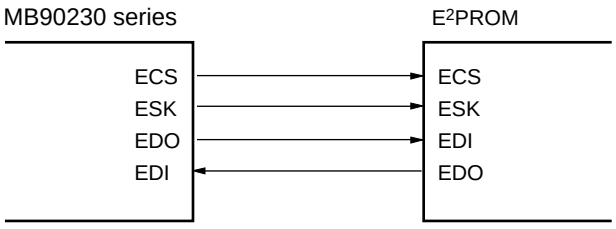
Parameter	Symbol	Value			Unit	Remarks
		Min.	Typ.	Max.		
Operation cycle	t_{SK}	1.0	—	—	µs	
Clock “H” time	t_{SKH}	0.4	0.5	—	µs	
Clock “L” time	t_{SKL}	0.4	0.5	—	µs	
ECS setup time	t_{CSS}	0.3	—	—	µs	
ECS hold time	t_{CSH}	0.0	—	—	µs	
EDO data decision time	t_{PD}	0.3	—	—	µs	
EDO output hold time	t_{OH}	0.5	—	—	µs	
EDI setup time	t_{DIS}	0.0	—	—	µs	
EDI hold time	t_{DIH}	0.4	—	—	µs	
READY ↑ → ECS ↓	t_{RCSH}	0.4	—	—	µs	
ECS “L” time	t_{CSL}	0.8	1.0	—	µs	

(2) E²PROM interface at an operation clock frequency of 2 MHz

(V_{CC} = +5.0 V±5%, V_{SS} = 0.0 V, T_A = –40°C to +70°C)

Parameter	Symbol	Value			Unit	Remarks
		Min.	Typ.	Max.		
Operation cycle	t_{SK}	0.5	—	—	µs	
Clock “H” time	t_{SKH}	0.2	0.25	—	µs	
Clock “L” time	t_{SKL}	0.2	0.25	—	µs	
ECS setup time	t_{CSS}	0.15	—	—	µs	
ECS hold time	t_{CSH}	0.0	—	—	µs	
EDO data decision time	t_{PD}	0.15	—	—	µs	
EDO output hold time	t_{OH}	0.25	—	—	µs	
EDI setup time	t_{DIS}	0.0	—	—	µs	
EDI hold time	t_{DIH}	0.2	—	—	µs	
READY ↑ → ECS ↓	t_{RCSH}	0.2	—	—	µs	
ECS “L” time	t_{CSL}	0.4	0.5	—	µs	


tSK
tSKH
tSKL
ESK
tPD
tOH
EDO
Determined data
Determined data
tCSS
tCSH
ECS
tDIS
tDIH
EDI
Input data
Input data
tCSL
ECS
tST
DO
(E2PROM output)
Hi-z
BUSY
READY


MB90230 series
E2PROM
ECS
ESK
EDO
EDI
ECS
ESK
EDI
EDO

■ INSTRUCTIONS (412 INSTRUCTIONS)

Table 1 Description of Instruction Table

Item	Description
Mnemonic	Upper-case letters and symbols: Described directry in assembly code Lower-case letters: Replaced when described in assembly code Numbers after lower-case letters: Indicates the bit width within the code
#	Indicates the number of bytes
~	Indicates the number of cycles See Table 4 for details about meanings of letters in items.
B	Indicates the compensation value for calculating the number of actual cycles during execution of instruction. The number of actual cycles during execution of instruction is summed with the value in the “cycles” column.
Operation	Indicates operation of instruction.
LH	Indicates special operations involving the bits 15 through 08 of the accumulator. Z: Transfers “0” X: Extends before transferring —: No transfer
AH	Indicates special operations involving the high-order 16 bits in the accumulator. *: Transfers from AL to AH —: No transfer Z: Transfers 00_H to AH. X: Transfers 00_H or FF_H to AH by extending AL
I	Indicates the status of each of the following flags: I (interrupt enable), S (stack), T (sticky bit), N (negative), Z (zero), V (overflow), and C (carry). *: Changes due to execution of instruction. —: No change. S: Set by execution of instruction. R: Reset by execution of instruction.
S	
T	
N	
Z	
V	
C	
RMW	Indicates whether the instruction is a read-modify-write instruction (a single instruction that reads data from memory, etc., processes the data, and then writes the result to memory.). *: Instruction is a read-modify-write instruction —: Instruction is not a read-modify-write instruction Note: Cannot be used for addresses that have different meanings depending on whether they are read or written.

Table 2 Explanation of Symbols in Table of Instructions

Symbol	Description
A	32-bit accumulator The number of bits used varies according to the instruction. Byte: Low order 8 bits of AL Word: 16 bits of AL Long: 32 bits of AL, AH
AH	High-order 16 bits of A
AL	Low-order 16 bits of A
SP	Stack pointer (USP or SSP)
PC	Program counter
SPCU	Stack pointer upper limit register
SPCL	Stack pointer lower limit register
PCB	Program bank register
DTB	Data bank register
ADB	Additional data bank register
SSB	System stack bank register
USB	User stack bank register
SPB	Current stack bank register (SSB or USB)
DPR	Direct page register
brg1	DTB, ADB, SSB, USB, DPR, PCB, SPB
brg2	DTB, ADB, SSB, USB, DPR, SPB
Ri	R0, R1, R2, R3, R4, R5, R6, R7
RWi	RW0, RW1, RW2, RW3, RW4, RW5, RW6, RW7
RWj	RW0, RW1, RW2, RW3
RLi	RL0, RL1, RL2, RL3
dir addr16 addr24 addr24 0 to 15 addr24 16 to 23	Compact direct addressing Direct addressing Physical direct addressing Bits 0 to 15 of addr24 Bits 16 to 23 of addr24
io	I/O area (000000_H to $0000FF_H$)
#imm4 #imm8 #imm16 #imm32 ext (imm8)	4-bit immediate data 8-bit immediate data 16-bit immediate data 32-bit immediate data 16-bit data signed and extended from 8-bit immediate data
disp8 disp16	8-bit displacement 16-bit displacement
bp	Bit offset value
vct4 vct8	Vector number (0 to 15) Vector number (0 to 255)
( )b	Bit address
rel ear eam	Branch specification relative to PC Effective addressing (codes 00 to 07) Effective addressing (codes 08 to 1F)
rlst	Register list

Table 3 Effective Address Fields

Code	Notation	Address format	Number of bytes in address extemsion*
00 01 02 03 04 05 06 07	R0 RW0 RL0 R1 RW1 (RL0) R2 RW2 RL1 R3 RW3 (RL1) R4 RW4 RL2 R5 RW5 (RL2) R6 RW6 RL3 R7 RW7 (RL3)	Register direct “ea” corresponds to byte, word, and long-word types, starting from the left	—
08 09 0A 0B	@RW0 @RW1 @RW2 @RW3	Register indirect	0
0C 0D 0E 0F	@RW0 + @RW1 + @RW2 + @RW3 +	Register indirect with post-increment	0
10 11 12 13 14 15 16 17	@RW0 + disp8 @RW1 + disp8 @RW2 + disp8 @RW3 + disp8 @RW4 + disp8 @RW5 + disp8 @RW6 + disp8 @RW7 + disp8	Register indirect with 8-bit displacement	1
18 19 1A 1B	@RW0 + disp16 @RW1 + disp16 @RW2 + disp16 @RW3 + disp16	Register indirect with 16-bit displacemen	2
1C 1D 1E 1F	@RW0 + RW7 @RW1 + RW7 @PC + dip16 addr16	Register indirect with index Register indirect with index PC indirect with 16-bit displacement Direct address	0 0 2 2

* : The number of bytes for address extension is indicated by the “+” symbol in the “#” (number of bytes) column in the Table of Instructions.

Table 4 Number of Execution Cycles for Each Form of Addressing

Code	Operand	(a)*
		Number of execution cycles for each from of addressing
00 to 07	Ri RWi RLi	Listed in Table of Instructions
08 to 0B	@RWj	1
0C to 0F	@RWj +	4
10 to 17	@RWi + disp8	1
18 to 1B	@RWj + disp16	1
1C 1D 1E 1F	@RW0 + RW7 @RW1 + RW7 @PC + dip16 @addr16	2 2 2 1

* : “(a)” is used in the “cycles” (number of cycles) column and column B (correction value) in the Table of Instructions.

Table 5 Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles

Operand	(b)*		(c)*		(d)*	
	byte		word		long	
Internal register	+	0	+	0	+	0
Internal RAM even address	+	0	+	0	+	0
Internal RAM odd address	+	0	+	1	+	2
Even address not in internal RAM	+	1	+	1	+	2
Odd address not in internal RAM	+	1	+	3	+	6
External data bus (8 bits)	+	1	+	3	+	6

* : “(b)”, “(c)”, and “(d)” are used in the “cycles” (number of cycles) column and column B (correction value) in the Table of Instructions.

Table 6 Transfer Instructions (Byte) [50 Instructions]

Mnemonic		#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOV	A, dir	2	2	(b)	byte (A) ← (dir)	Z	*	–	–	–	*	*	–	–	–
MOV	A, addr16	3	2	(b)	byte (A) ← (addr16)	Z	*	–	–	–	*	*	–	–	–
MOV	A, Ri	1	1	0	byte (A) ← (Ri)	Z	*	–	–	–	*	*	–	–	–
MOV	A, ear	2	1	0	byte (A) ← (ear)	Z	*	–	–	–	*	*	–	–	–
MOV	A, eam	2+	2+ (a)	(b)	byte (A) ← (eam)	Z	*	–	–	–	*	*	–	–	–
MOV	A, io	2	2	(b)	byte (A) ← (io)	Z	*	–	–	–	*	*	–	–	–
MOV	A, #imm8	2	2	0	byte (A) ← imm8	Z	*	–	–	–	*	*	–	–	–
MOV	A, @A	2	2	(b)	byte (A) ← ((A))	Z	–	–	–	–	*	*	–	–	–
MOV	A, @RLi+disp8	3	6	(b)	byte (A) ← ((RLi))+disp8)	Z	*	–	–	–	*	*	–	–	–
MOV	A, @SP+disp8	3	3	(b)	byte (A) ← ((SP)+disp8)	Z	*	–	–	–	*	*	–	–	–
MOVP	A, addr24	5	3	(b)	byte (A) ←(addr24)	Z	*	–	–	–	*	*	–	–	–
MOVP	A, @A	2	2	(b)	byte (A) ← ((A))	Z	–	–	–	–	*	*	–	–	–
MOVN	A, #imm4	1	1	0	byte (A) ← imm4	Z	*	–	–	–	R	*	–	–	–
MOVX	A, dir	2	2	(b)	byte (A) ← (dir)	X	*	–	–	–	*	*	–	–	–
MOVX	A, addr16	3	2	(b)	byte (A) ← (addr16)	X	*	–	–	–	*	*	–	–	–
MOVX	A, Ri	2	1	0	byte (A) ← (Ri)	X	*	–	–	–	*	*	–	–	–
MOVX	A, ear	2	1	0	byte (A) ← (ear)	X	*	–	–	–	*	*	–	–	–
MOVX	A, eam	2+	2+ (a)	(b)	byte (A) ← (eam)	X	*	–	–	–	*	*	–	–	–
MOVX	A, io	2	2	(b)	byte (A) ← (io)	X	*	–	–	–	*	*	–	–	–
MOVX	A, #imm8	2	2	0	byte (A) ← imm8	X	*	–	–	–	*	*	–	–	–
MOVX	A, @A	2	2	(b)	byte (A) ← ((A))	X	–	–	–	–	*	*	–	–	–
MOVX	A,@RWi+disp8	2	3	(b)	byte (A) ← ((RWi))+disp8)	X	*	–	–	–	*	*	–	–	–
MOVX	A, @RLi+disp8	3	6	(b)	byte (A) ← ((RLi))+disp8)	X	*	–	–	–	*	*	–	–	–
MOVX	A, @SP+disp8	3	3	(b)	byte (A) ← ((SP)+disp8)	X	*	–	–	–	*	*	–	–	–
MOVPX	A, addr24	5	3	(b)	byte (A) ←(addr24)	X	*	–	–	–	*	*	–	–	–
MOVPX	A, @A	2	2	(b)	byte (A) ← ((A))	X	–	–	–	–	*	*	–	–	–
MOV	dir, A	2	2	(b)	byte (dir) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV	addr16, A	3	2	(b)	byte (addr16) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV	Ri, A	1	1	0	byte (Ri) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV	ear, A	2	2	0	byte (ear) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV	eam, A	2+	2+ (a)	(b)	byte (eam) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV	io, A	2	2	(b)	byte (io) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV	@RLi+disp8, A	3	6	(b)	byte ((RLi)) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV	@SP+disp8, A	3	3	(b)	byte ((SP)+disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVP	addr24, A	5	3	(b)	byte (addr24) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV	Ri, ear	2	2	0	byte (Ri) ← (ear)	–	–	–	–	–	*	*	–	–	–
MOV	Ri, eam	2+	3+ (a)	(b)	byte (Ri) ← (eam)	–	–	–	–	–	*	*	–	–	–
MOVP	@A, Ri	2	3	(b)	byte ((A)) ← (Ri)	–	–	–	–	–	*	*	–	–	–
MOV	ear, Ri	2	3	0	byte (ear) ← (Ri)	–	–	–	–	–	*	*	–	–	–
MOV	eam, Ri	2+	3+ (a)	(b)	byte (eam) ← (Ri)	–	–	–	–	–	*	*	–	–	–
MOV	Ri, #imm8	2	2	0	byte (Ri) ← imm8	–	–	–	–	–	*	*	–	–	–
MOV	io, #imm8	3	3	(b)	byte (io) ← imm8	–	–	–	–	–	–	–	–	–	–
MOV	dir, #imm8	3	3	(b)	byte (dir) ← imm8	–	–	–	–	–	–	–	–	–	–
MOV	ear, #imm8	3	2	0	byte (ear) ← imm8	–	–	–	–	–	*	*	–	–	–
MOV	eam, #imm8	3+	2+ (a)	(b)	byte (eam) ← imm8	–	–	–	–	–	–	–	–	–	–
MOV	@AL, AH	2	2	(b)	byte ((A)) ← (AH)	–	–	–	–	–	*	*	–	–	–
XCH	A, ear	2	3	0	byte (A) ↔ (ear)	Z	–	–	–	–	–	–	–	–	–
XCH	A, eam	2+	3+ (a)	2× (b)	byte (A) ↔ (eam)	Z	–	–	–	–	–	–	–	–	–
XCH	Ri, ear	2	4	0	byte (Ri) ↔ (ear)	–	–	–	–	–	–	–	–	–	–
XCH	Ri, eam	2+	5+ (a)	2× (b)	byte (Ri) ↔ (eam)	–	–	–	–	–	–	–	–	–	–

For an explanation of “(a)” and “(b)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 7 Transfer Instructions (Word) [40 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVW A, dir	2	2	(c)	word (A) ← (dir)	–	*	–	–	–	*	*	–	–	–
MOVW A, addr16	3	2	(c)	word (A) ← (addr16)	–	*	–	–	–	*	*	–	–	–
MOVW A, SP	1	2	0	word (A) ← (SP)	–	*	–	–	–	*	*	–	–	–
MOVW A, RWi	1	1	0	word (A) ← (RWi)	–	*	–	–	–	*	*	–	–	–
MOVW A, ear	2	1	0	word (A) ← (ear)	–	*	–	–	–	*	*	–	–	–
MOVW A, eam	2+	2+ (a)	(c)	word (A) ← (eam)	–	*	–	–	–	*	*	–	–	–
MOVW A, io	2	2	(c)	word (A) ← (io)	–	*	–	–	–	*	*	–	–	–
MOVW A, @A	2	2	(c)	word (A) ← ((A))	–	–	–	–	–	*	*	–	–	–
MOVW A, #imm16	3	2	0	word (A) ← imm16	–	*	–	–	–	*	*	–	–	–
MOVW A, @RWi+disp8	2	3	(c)	word (A) ← ((RWi) +disp8)	–	*	–	–	–	*	*	–	–	–
MOVW A, @RLi+disp8	3	6	(c)	word (A) ← ((RLi) +disp8)	–	*	–	–	–	*	*	–	–	–
MOVW A, @SP+disp8	3	3	(c)	word (A) ← ((SP) +disp8	–	*	–	–	–	*	*	–	–	–
MOVPW A, addr24	5	3	(c)	word (A) ← (addr24)	–	*	–	–	–	*	*	–	–	–
MOVPW A, @A	2	2	(c)	word (A) ← ((A))	–	–	–	–	–	*	*	–	–	–
MOVW dir, A	2	2	(c)	word (dir) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW addr16, A	3	2	(c)	word (addr16) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW SP, # imm16	4	2	0	word (SP) ← imm16	–	–	–	–	–	*	*	–	–	–
MOVW SP, A	1	2	0	word (SP) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, A	1	1	0	word (RWi) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW ear, A	2	2	0	word (ear) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW eam, A	2+	2+ (a)	(c)	word (eam) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW io, A	2	2	(c)	word (io) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW @RWi+disp8, A	2	3	(c)	word ((RWi) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW @RLi+disp8, A	3	6	(c)	word ((RLi) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW @SP+disp8, A	3	3	(c)	word ((SP) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVPW addr24, A	5	3	(c)	word (addr24) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVPW @A, RWi	2	3	(c)	word ((A)) ← (RWi)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, ear	2	2	0	word (RWi) ← (ear)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, eam	2+	3+ (a)	(c)	word (RWi) ← (eam)	–	–	–	–	–	*	*	–	–	–
MOVW ear, RWi	2	3	0	word (ear) ← (RWi)	–	–	–	–	–	*	*	–	–	–
MOVW eam, RWi	2+	3+ (a)	(c)	word (eam) ← (RWi)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, #imm16	3	2	0	word (RWi) ← imm16	–	–	–	–	–	*	*	–	–	–
MOVW io, #imm16	4	3	(c)	word (io) ← imm16	–	–	–	–	–	–	–	–	–	–
MOVW ear, #imm16	4	2	0	word (ear) ← imm16	–	–	–	–	–	*	*	–	–	–
MOVW eam, #imm16	4+	2+ (a)	(c)	word (eam) ← imm16	–	–	–	–	–	–	–	–	–	–
MOVW @AL, AH	2	2	(c)	word ((A)) ← (AH)	–	–	–	–	–	*	*	–	–	–
XCHW A, ear	2	3	0	word (A) ↔ (ear)	–	–	–	–	–	–	–	–	–	–
XCHW A, eam	2+	3+ (a)	2× (c)	word (A) ↔ (eam)	–	–	–	–	–	–	–	–	–	–
XCHW RWi, ear	2	4	0	word (RWi) ↔ (ear)	–	–	–	–	–	–	–	–	–	–
XCHW RWi, eam	2+	5+ (a)	2× (c)	word (RWi) ↔ (eam)	–	–	–	–	–	–	–	–	–	–

Note: For an explanation of “(a)” and “(c)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 8 Transfer Instructions (Long Word) [11 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVL A, ear	2	1	0	long (A) ← (ear)	–	–	–	–	–	*	*	–	–	–
MOVL A, eam	2+	3+ (a)	(d)	long (A) ← (eam)	–	–	–	–	–	*	*	–	–	–
MOVL A, # imm32	5	3	0	long (A) ← imm32	–	–	–	–	–	*	*	–	–	–
MOVL A, @SP + disp8	3	4	(d)	long (A) ← ((SP) +disp8)	–	–	–	–	–	*	*	–	–	–
MOVPL A, addr24	5	4	(d)	long (A) ← (addr24)	–	–	–	–	–	*	*	–	–	–
MOVPL A, @A	2	3	(d)	long (A) ← ((A))	–	–	–	–	–	*	*	–	–	–
MOVPL @A, RLi	2	5	(d)	long ((A)) ← (RLi)	–	–	–	–	–	*	*	–	–	–
MOVL @SP + disp8, A	3	4	(d)	long ((SP) + disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVPL addr24, A	5	4	(d)	long (addr24) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVL ear, A	2	2	0	long (ear) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVL eam, A	2+	3+ (a)	(d)	long (eam) ← (A)	–	–	–	–	–	*	*	–	–	–

For an explanation of “(a)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 9 Addition and Subtraction Instructions (Byte/Word/Long Word) [42 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
ADD A, #imm8	2	2	0	byte (A) ← (A) + imm8	Z	–	–	–	–	*	*	*	*	–
ADD A, dir	2	3	(b)	byte (A) ← (A) + (dir)	Z	–	–	–	–	*	*	*	*	–
ADD A, ear	2	2	0	byte (A) ← (A) + (ear)	Z	–	–	–	–	*	*	*	*	–
ADD A, eam	2+	3+ (a)	(b)	byte (A) ← (A) + (eam)	Z	–	–	–	–	*	*	*	*	–
ADD ear, A	2	2	0	byte (ear) ← (ear) + (A)	–	–	–	–	–	*	*	*	*	*
ADD eam, A	2+	3+ (a)	2× (b)	byte (eam) ← (eam) + (A)	Z	–	–	–	–	*	*	*	*	*
ADDC A	1	2	0	byte (A) ← (AH) + (AL) + (C)	Z	–	–	–	–	*	*	*	*	–
ADDC A, ear	2	2	0	byte (A) ← (A) + (ear) + (C)	Z	–	–	–	–	*	*	*	*	–
ADDC A, eam	2+	3+ (a)	(b)	byte (A) ← (A) + (eam) + (C)	Z	–	–	–	–	*	*	*	*	–
ADDDC A	1	3	0	byte (A) ← (AH) + (AL) + (C) (Decimal)	Z	–	–	–	–	*	*	*	*	–
SUB A, #imm8	2	2	0	byte (A) ← (A) – imm8	Z	–	–	–	–	*	*	*	*	–
SUB A, dir	2	3	(b)	byte (A) ← (A) – (dir)	Z	–	–	–	–	*	*	*	*	–
SUB A, ear	2	2	0	byte (A) ← (A) – (ear)	Z	–	–	–	–	*	*	*	*	–
SUB A, eam	2+	3+ (a)	(b)	byte (A) ← (A) – (eam)	Z	–	–	–	–	*	*	*	*	–
SUB ear, A	2	2	0	byte (ear) ← (ear) – (A)	–	–	–	–	–	*	*	*	*	*
SUB eam, A	2+	3+ (a)	2× (b)	byte (eam) ← (eam) – (A)	–	–	–	–	–	*	*	*	*	*
SUBC A	1	2	0	byte (A) ← (AH) – (AL) – (C)	Z	–	–	–	–	*	*	*	*	–
SUBC A, ear	2	2	0	byte (A) ← (A) – (ear) – (C)	Z	–	–	–	–	*	*	*	*	–
SUBC A, eam	2+	3+ (a)	(b)	byte (A) ← (A) – (eam) – (C)	Z	–	–	–	–	*	*	*	*	–
SUBDC A	1	3	0	byte (A) ← (AH) – (AL) – (C) (Decimal)	Z	–	–	–	–	*	*	*	*	–
ADDW A	1	2	0	word (A) ← (AH) + (AL)	–	–	–	–	–	*	*	*	*	–
ADDW A, ear	2	2	0	word (A) ← (A) + (ear)	–	–	–	–	–	*	*	*	*	–
ADDW A, eam	2+	3+ (a)	(c)	word (A) ← (A) + (eam)	–	–	–	–	–	*	*	*	*	–
ADDW A, #imm16	3	2	0	word (A) ← (A) + imm16	–	–	–	–	–	*	*	*	*	–
ADDW ear, A	2	2	0	word (ear) ← (ear) + (A)	–	–	–	–	–	*	*	*	*	*
ADDW eam, A	2+	3+ (a)	2× (c)	word (eam) ← (eam) + (A)	–	–	–	–	–	*	*	*	*	*
ADDCW A, ear	2	2	0	word (A) ← (A) + (ear) + (C)	–	–	–	–	–	*	*	*	*	–
ADDCW A, eam	2+	3+ (a)	(c)	word (A) ← (A) + (eam) + (C)	–	–	–	–	–	*	*	*	*	–
SUBW A	1	2	0	word (A) ← (AH) – (AL)	–	–	–	–	–	*	*	*	*	–
SUBW A, ear	2	2	0	word (A) ← (A) – (ear)	–	–	–	–	–	*	*	*	*	–
SUBW A, eam	2+	3+ (a)	(c)	word (A) ← (A) – (eam)	–	–	–	–	–	*	*	*	*	–
SUBW A, #imm16	3	2	0	word (A) ← (A) – imm16	–	–	–	–	–	*	*	*	*	–
SUBW ear, A	2	2	0	word (ear) ← (ear) – (A)	–	–	–	–	–	*	*	*	*	*
SUBW eam, A	2+	3+ (a)	2× (c)	word (eam) ← (eam) – (A)	–	–	–	–	–	*	*	*	*	*
SUBCW A, ear	2	2	0	word (A) ← (A) – (ear) – (C)	–	–	–	–	–	*	*	*	*	–
SUBCW A, eam	2+	3+ (a)	(c)	word (A) ← (A) – (eam) – (C)	–	–	–	–	–	*	*	*	*	–
ADDL A, ear	2	5	0	long (A) ← (A) + (ear)	–	–	–	–	–	*	*	*	*	–
ADDL A, eam	2+	6+ (a)	(d)	long (A) ← (A) + (eam)	–	–	–	–	–	*	*	*	*	–
ADDL A, #imm32	5	4	0	long (A) ← (A) + imm32	–	–	–	–	–	*	*	*	*	–
SUBL A, ear	2	5	0	long (A) ← (A) – (ear)	–	–	–	–	–	*	*	*	*	–
SUBL A, eam	2+	6+ (a)	(d)	long (A) ← (A) – (eam)	–	–	–	–	–	*	*	*	*	–
SUBL A, #imm32	5	4	0	long (A) ← (A) – imm32	–	–	–	–	–	*	*	*	*	–

For an explanation of “(a)”, “(b)”, “(c)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 10 Increment and Decrement Instructions (Byte/Word/Long Word) [12 Instructions]

Mnemonic		#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
INC	ear	2	2	0	byte (ear) ← (ear) +1	–	–	–	–	–	*	*	*	–	*
INC	eam	2+	3+ (a)	2× (b)	byte (eam) ← (eam) +1	–	–	–	–	–	*	*	*	–	*
DEC	ear	2	2	0	byte (ear) ← (ear) –1	–	–	–	–	–	*	*	*	–	*
DEC	eam	2+	3+ (a)	2× (b)	byte (eam) ← (eam) –1	–	–	–	–	–	*	*	*	–	*
INCW	ear	2	2	0	word (ear) ← (ear) +1	–	–	–	–	–	*	*	*	–	*
INCW	eam	2+	3+ (a)	2× (c)	word (eam) ← (eam) +1	–	–	–	–	–	*	*	*	–	*
DECW	ear	2	2	0	word (ear) ← (ear) –1	–	–	–	–	–	*	*	*	–	*
DECW	eam	2+	3+ (a)	2× (c)	word (eam) ← (eam) –1	–	–	–	–	–	*	*	*	–	*
INCL	ear	2	4	0	long (ear) ← (ear) +1	–	–	–	–	–	*	*	*	–	*
INCL	eam	2+	5+ (a)	2× (d)	long (eam) ← (eam) +1	–	–	–	–	–	*	*	*	–	*
DECL	ear	2	4	0	long (ear) ← (ear) –1	–	–	–	–	–	*	*	*	–	*
DECL	eam	2+	5+ (a)	2× (d)	long (eam) ← (eam) –1	–	–	–	–	–	*	*	*	–	*

For an explanation of “(a)”, “(b)”, “(c)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 11 Compare Instructions (Byte/Word/Long Word) [11 Instructions]

Mnemonic		#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
CMP	A	1	2	0	byte (AH) – (AL)	–	–	–	–	–	*	*	*	*	–
CMP	A, ear	2	2	0	byte (A) – (ear)	–	–	–	–	–	*	*	*	*	–
CMP	A, eam	2+	2+ (a)	(b)	byte (A) – (eam)	–	–	–	–	–	*	*	*	*	–
CMP	A, #imm8	2	2	0	byte (A) – imm8	–	–	–	–	–	*	*	*	*	–
CMPW	A	1	2	0	word (AH) – (AL)	–	–	–	–	–	*	*	*	*	–
CMPW	A, ear	2	2	0	word (A) – (ear)	–	–	–	–	–	*	*	*	*	–
CMPW	A, eam	2+	2+ (a)	(c)	word (A) – (eam)	–	–	–	–	–	*	*	*	*	–
CMPW	A, #imm16	3	2	0	word (A) – imm16	–	–	–	–	–	*	*	*	*	–
CMPL	A, ear	2	3	0	long (A) – (ear)	–	–	–	–	–	*	*	*	*	–
CMPL	A, eam	2+	4+ (a)	(d)	long (A) – (eam)	–	–	–	–	–	*	*	*	*	–
CMPL	A, #imm32	5	3	0	long (A) – imm32	–	–	–	–	–	*	*	*	*	–

For an explanation of “(a)”, “(b)”, “(c)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 12 Unsigned Multiplication and Division Instructions (Word/Long Word) [11 Instructions]

Mnemonic		#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
DIVU	A	1	*1	0	word (AH) /byte (AL) Quotient → byte (AL) Remainder → byte (AH)	–	–	–	–	–	–	–	*	*	–
DIVU	A, ear	2	*2	0	word (A)/byte (ear) Quotient → byte (A) Remainder → byte (ear)	–	–	–	–	–	–	–	*	*	–
DIVU	A, eam	2+	*3	*6	word (A)/byte (eam) Quotient → byte (A) Remainder → byte (eam)	–	–	–	–	–	–	–	*	*	–
DIVUW	A, ear	2	*4	0	long (A)/word (ear) Quotient → word (A) Remainder → word (ear)	–	–	–	–	–	–	–	*	*	–
DIVUW	A, eam	2+	*5	*7	long (A)/word (eam) Quotient → word (A) Remainder → word (eam)	–	–	–	–	–	–	–	*	*	–
MULU	A	1	*8	0	byte (AH) × byte (AL) → word (A)	–	–	–	–	–	–	–	–	–	–
MULU	A, ear	2	*9	0	byte (A) × byte (ear) → word (A)	–	–	–	–	–	–	–	–	–	–
MULU	A, eam	2+	*10	(b)	byte (A) × byte (eam) → word (A)	–	–	–	–	–	–	–	–	–	–
MULUW	A	1	*11	0	word (AH) × word (AL) → long (A)	–	–	–	–	–	–	–	–	–	–
MULUW	A, ear	2	*12	0	word (A) × word (ear) → long (A)	–	–	–	–	–	–	–	–	–	–
MULUW	A, eam	2+	*13	(c)	word (A) × word (eam) → long (A)	–	–	–	–	–	–	–	–	–	–

For an explanation of “(b)” and “(c), refer to Table 5, “Correction Values for Number of Cycle Used to Calculate Number of Actual Cycles.”

*1: 3 when dividing into zero, 6 when an overflow occurs, and 14 normally.
*2: 3 when dividing into zero, 5 when an overflow occurs, and 13 normally.
*3: 5 + (a) when dividing into zero, 7 + (a) when an overflow occurs, and 17 + (a) normally.
*4: 3 when dividing into zero, 5 when an overflow occurs, and 21 normally.
*5: 4 + (a) when dividing into zero, 7 + (a) when an overflow occurs, and 25 + (a) normally.
*6: (b) when dividing into zero or when an overflow occurs, and 2 × (b) normally.
*7: (c) when dividing into zero or when an overflow occurs, and 2 × (c) normally.
*8: 3 when byte (AH) is zero, and 7 when byte (AH) is not 0.
*9: 3 when byte (ear) is zero, and 7 when byte (ear) is not 0.
*10: 4 + (a) when byte (eam) is zero, and 8 + (a) when byte (eam) is not 0.
*11: 3 when word (AH) is zero, and 11 when word (AH) is not 0.
*12: 3 when word (ear) is zero, and 11 when word (ear) is not 0.
*13: 4 + (a) when word (eam) is zero, and 12 + (a) when word (eam) is not 0.

Table 13 Signed Multiplication and Division Instructions (Word/Long Word) [11 Insturctions]

Mnemonic		#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
DIV	A	2	*1	0	word (AH) /byte (AL) Quotient → byte (AL) Remainder → byte (AH)	Z	–	–	–	–	–	–	*	*	–
DIV	A, ear	2	*2	0	word (A)/byte (ear) Quotient → byte (A) Remainder → byte (ear)	Z	–	–	–	–	–	–	*	*	–
DIV	A, eam	2+	*3	*6	word (A)/byte (eam) Quotient → byte (A) Remainder → byte (eam)	Z	–	–	–	–	–	–	*	*	–
DIVW	A, ear	2	*4	0	long (A)/word (ear) Quotient → word (A) Remainder → word (ear)	–	–	–	–	–	–	–	*	*	–
DIVW	A, eam	2+	*5	*7	long (A)/word (eam) Quotient → word (A) Remainder → word (eam)	–	–	–	–	–	–	–	*	*	–
MUL	A	2	*8	0	byte (AH) × byte (AL) → word (A)	–	–	–	–	–	–	–	–	–	–
MUL	A, ear	2	*9	0	byte (A) × byte (ear) → word (A)	–	–	–	–	–	–	–	–	–	–
MUL	A, eam	2+	*10	(b)	byte (A) × byte (eam) → word (A)	–	–	–	–	–	–	–	–	–	–
MULW	A	2	*11	0	word (AH) × word (AL) → long (A)	–	–	–	–	–	–	–	–	–	–
MULW	A, ear	2	*12	0	word (A) × word (ear) → long (A)	–	–	–	–	–	–	–	–	–	–
MULW	A, eam	2+	*13	(b)	word (A) × word (eam) → long (A)	–	–	–	–	–	–	–	–	–	–

For an explanation of “(b)” and “(c)”, refer to Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

*1: 3 when dividing into zero, 8 or 18 when an overflow occurs, and 18 normally.
*2: 3 when dividing into zero, 10 or 21 when an overflow occurs, and 22 normally.
*3: 4 + (a) when dividing into zero, 11 + (a) or 22 + (a) when an overflow occurs, and 23 + (a) normally.
*4: When the dividend is positive: 4 when dividing into zero, 10 or 29 when an overflow occurs, and 30 normally.
When the dividend is negative: 4 when dividing into zero, 11 or 30 when an overflow occurs, and 31 normally.
*5: When the dividend is positive: 4 + (a) when dividing into zero, 11 + (a) or 30 + (a) when an overflow occurs, and 31 + (a) normally.
When the dividend is negative: 4 + (a) when dividing into zero, 12 + (a) or 31 + (a) when an overflow occurs, and 32 + (a) normally.
*6: (b) when dividing into zero or when an overflow occurs, and 2 × (b) normally.
*7: (c) when dividing into zero or when an overflow occurs, and 2 × (c) normally.
*8: 3 when byte (AH) is zero, 12 when the result is positive, and 13 when the result is negative.
*9: 3 when byte (ear) is zero, 12 when the result is positive, and 13 when the result is negative.
*10: 4 + (a) when byte (eam) is zero, 13 + (a) when the result is positive, and 14 + (a) when the result is negative.
*11: 3 when word (AH) is zero, 12 when the result is positive, and 13 when the result is negative.
*12: 3 when word (ear) is zero, 16 when the result is positive, and 19 when the result is negative.
*13: 4 + (a) when word (eam) is zero, 17 + (a) when the result is positive, and 20 + (a) when the result is negative.

Note: Which of the two values given for the number of execution cycles applies when an overflow error occurs in a DIV or DIVW instruction depends on whether the overflow was detected before or after the operation.

Table 14 Logical 1 Instructions (Byte, Word) [39 Instructions]

Mnemonic		#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
AND	A, #imm8	2	2	0	byte (A) ← (A) and imm8	–	–	–	–	–	*	*	R	–	–
AND	A, ear	2	2	0	byte (A) ← (A) and (ear)	–	–	–	–	–	*	*	R	–	–
AND	A, eam	2+	3+ (a)	(b)	byte (A) ← (A) and (eam)	–	–	–	–	–	*	*	R	–	–
AND	ear, A	2	3	0	byte (ear) ← (ear) and (A)	–	–	–	–	–	*	*	R	–	*
AND	eam, A	2+	3+ (a)	2× (b)	byte (eam) ← (eam) and (A)	–	–	–	–	–	*	*	R	–	*
OR	A, #imm8	2	2	0	byte (A) ← (A) or imm8	–	–	–	–	–	*	*	R	–	–
OR	A, ear	2	2	0	byte (A) ← (A) or (ear)	–	–	–	–	–	*	*	R	–	–
OR	A, eam	2+	3+ (a)	(b)	byte (A) ← (A) or (eam)	–	–	–	–	–	*	*	R	–	–
OR	ear, A	2	3	0	byte (ear) ← (ear) or (A)	–	–	–	–	–	*	*	R	–	*
OR	eam, A	2+	3+ (a)	2× (b)	byte (eam) ← (eam) or (A)	–	–	–	–	–	*	*	R	–	*
XOR	A, #imm8	2	2	0	byte (A) ← (A) xor imm8	–	–	–	–	–	*	*	R	–	–
XOR	A, ear	2	2	0	byte (A) ← (A) xor (ear)	–	–	–	–	–	*	*	R	–	–
XOR	A, eam	2+	3+ (a)	(b)	byte (A) ← (A) xor (eam)	–	–	–	–	–	*	*	R	–	–
XOR	ear, A	2	3	0	byte (ear) ← (ear) xor (A)	–	–	–	–	–	*	*	R	–	*
XOR	eam, A	2+	3+ (a)	2× (b)	byte (eam) ← (eam) xor (A)	–	–	–	–	–	*	*	R	–	*
NOT	A	1	2	0	byte (A) ← not (A)	–	–	–	–	–	*	*	R	–	–
NOT	ear	2	2	0	byte (ear) ← not (ear)	–	–	–	–	–	*	*	R	–	*
NOT	eam	2+	3+ (a)	2× (b)	byte (eam) ← not (eam)	–	–	–	–	–	*	*	R	–	*
ANDW	A	1	2	0	word (A) ← (AH) and (A)	–	–	–	–	–	*	*	R	–	–
ANDW	A, #imm16	3	2	0	word (A) ← (A) and imm16	–	–	–	–	–	*	*	R	–	–
ANDW	A, ear	2	2	0	word (A) ← (A) and (ear)	–	–	–	–	–	*	*	R	–	–
ANDW	A, eam	2+	3+ (a)	(c)	word (A) ← (A) and (eam)	–	–	–	–	–	*	*	R	–	–
ANDW	ear, A	2	3	0	word (ear) ← (ear) and (A)	–	–	–	–	–	*	*	R	–	*
ANDW	eam, A	2+	3+ (a)	2× (c)	word (eam) ← (eam) and (A)	–	–	–	–	–	*	*	R	–	*
ORW	A	1	2	0	word (A) ← (AH) or (A)	–	–	–	–	–	*	*	R	–	–
ORW	A, #imm16	3	2	0	word (A) ← (A) or imm16	–	–	–	–	–	*	*	R	–	–
ORW	A, ear	2	2	0	word (A) ← (A) or (ear)	–	–	–	–	–	*	*	R	–	–
ORW	A, eam	2+	3+ (a)	(c)	word (A) ← (A) or (eam)	–	–	–	–	–	*	*	R	–	–
ORW	ear, A	2	3	0	word (ear) ← (ear) or (A)	–	–	–	–	–	*	*	R	–	*
ORW	eam, A	2+	3+ (a)	2× (c)	word (eam) ← (eam) or (A)	–	–	–	–	–	*	*	R	–	*
XORW	A	1	2	0	word (A) ← (AH) xor (A)	–	–	–	–	–	*	*	R	–	–
XORW	A, #imm16	3	2	0	word (A) ← (A) xor imm16	–	–	–	–	–	*	*	R	–	–
XORW	A, ear	2	2	0	word (A) ← (A) xor (ear)	–	–	–	–	–	*	*	R	–	–
XORW	A, eam	2+	3+ (a)	(c)	word (A) ← (A) xor (eam)	–	–	–	–	–	*	*	R	–	–
XORW	ear, A	2	3	0	word (ear) ← (ear) xor (A)	–	–	–	–	–	*	*	R	–	*
XORW	eam, A	2+	3+ (a)	2× (c)	word (eam) ← (eam) xor (A)	–	–	–	–	–	*	*	R	–	*
NOTW	A	1	2	0	word (A) ← not (A)	–	–	–	–	–	*	*	R	–	–
NOTW	ear	2	2	0	word (ear) ← not (ear)	–	–	–	–	–	*	*	R	–	*
NOTW	eam	2+	3+ (a)	2× (c)	word (eam) ← not (eam)	–	–	–	–	–	*	*	R	–	*

For an explanation of “(a)”, “(b)”, “(c)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 15 Logical 2 Instructions (Long Word) [6 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
ANDL A, ear	2	5	0	long (A) ← (A) and (ear)	–	–	–	–	–	*	*	R	–	–
ANDL A, eam	2+	6+ (a)	(d)	long (A) ← (A) and (eam)	–	–	–	–	–	*	*	R	–	–
ORL A, ear	2	5	0	long (A) ← (A) or (ear)	–	–	–	–	–	*	*	R	–	–
ORL A, eam	2+	6+ (a)	(d)	long (A) ← (A) or (eam)	–	–	–	–	–	*	*	R	–	–
XORL A, ear	2	5	0	long (A) ← (A) xor (ear)	–	–	–	–	–	*	*	R	–	–
XORL A, eam	2+	6+ (a)	(d)	long (A) ← (A) xor (eam)	–	–	–	–	–	*	*	R	–	–

For an explanation of “(a)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 16 Sign Inversion Instructions (Byte/Word) [6 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
NEG A	1	2	0	byte (A) ← 0 – (A)	X	–	–	–	–	*	*	*	*	–
NEG ear	2	2	0	byte (ear) ← 0 – (ear)	–	–	–	–	–	*	*	*	*	*
NEG eam	2+	3+ (a)	2× (b)	byte (eam) ← 0 – (eam)	–	–	–	–	–	*	*	*	*	*
NEGW A	1	2	0	word (A) ← 0 – (A)	–	–	–	–	–	*	*	*	*	–
NEGW ear	2	2	0	word (ear) ← 0 – (ear)	–	–	–	–	–	*	*	*	*	*
NEGW eam	2+	3+ (a)	2× (c)	word (eam) ← 0 – (eam)	–	–	–	–	–	*	*	*	*	*

For an explanation of “(a)”, “(b)” and “(c)” and refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 17 Absolute Value Instructions (Byte/Word/Long Word) [3 Insturctions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
ABS A	2	2	0	byte (A) ← absolute value (A)	Z	–	–	–	–	*	*	*	–	–
ABSW A	2	2	0	word (A) ← absolute value (A)	–	–	–	–	–	*	*	*	–	–
ABSL A	2	4	0	long (A) ← absolute value (A)	–	–	–	–	–	*	*	*	–	–

Table 18 Normalize Instructions (Long Word) [1 Instruction]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
NRML A, R0	2	*	0	long (A) ← Shifts to the position at which “1” was set first byte (R0) ← current shift count	–	–	–	–	*	–	–	–	–	–

* : 5 when the contents of the accumulator are all zeroes, 5 + (R0) in all other cases.

Table 19 Shift Instructions (Byte/Word/Long Word) [27 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
RORC A	2	2	0	byte (A) ← Right rotation with carry	–	–	–	–	–	*	*	–	*	–
ROLC A	2	2	0	byte (A) ← Left rotation with carry	–	–	–	–	–	*	*	–	*	–
RORC ear	2	2	0	byte (ear) ← Right rotation with carry	–	–	–	–	–	*	*	–	*	*
RORC eam	2+	3+ (a)	2× (b)	byte (eam) ← Right rotation with carry	–	–	–	–	–	*	*	–	*	*
ROLC ear	2	2	0	byte (ear) ← Left rotation with carry	–	–	–	–	–	*	*	–	*	*
ROLC eam	2+	3+ (a)	2× (b)	byte (eam) ← Left rotation with carry	–	–	–	–	–	*	*	–	*	*
ASR A, R0	2	*1	0	byte (A) ← Arithmetic right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSR A, R0	2	*1	0	byte (A) ← Logical right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSL A, R0	2	*1	0	byte (A) ← Logical left barrel shift (A, R0)	–	–	–	–	–	*	*	–	*	–
ASR A, #imm8	3	*3	0	byte (A) ← Arithmetic right barrel shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSR A, #imm8	3	*3	0	byte (A) ← Logical right barrel shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSL A, #imm8	3	*3	0	byte (A) ← Logical left barrel shift (A, imm8)	–	–	–	–	–	*	*	–	*	–
ASRW A	1	2	0	word (A) ← Arithmetic right shift (A, 1 bit)	–	–	–	–	*	*	*	–	*	–
LSRW A/SHRW A	1	2	0	word (A) ← Logical right shift (A, 1 bit)	–	–	–	–	*	R	*	–	*	–
LSLW A/SHLW A	1	2	0	word (A) ← Logical left shift (A, 1 bit)	–	–	–	–	–	*	*	–	*	–
ASRW A, R0	2	*1	0	word (A) ← Arithmetic right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSRW A, R0	2	*1	0	word (A) ← Logical right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSLW A, R0	2	*1	0	word (A) ← Logical left barrel shift (A, R0)	–	–	–	–	–	*	*	–	*	–
ASRW A, #imm8	3	*3	0	word (A) ← Arithmetic right barrel shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSRW A, #imm8	3	*3	0	word (A) ← Logical right barrel shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSLW A, #imm8	3	*3	0	word (A) ← Logical left barrel shift (A, imm8)	–	–	–	–	–	*	*	–	*	–
ASRL A, R0	2	*2	0	long (A) ← Arithmetic right shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSRL A, R0	2	*2	0	long (A) ← Logical right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSLL A, R0	2	*2	0	long (A) ← Logical left barrel shift (A, R0)	–	–	–	–	–	*	*	–	*	–
ASRL A, #imm8	3	*4	0	long (A) ← Arithmetic right shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSRL A, #imm8	3	*4	0	long (A) ← Logical right barrel shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSLL A, #imm8	3	*4	0	long (A) ← Logical left barrel shift (A, imm8)	–	–	–	–	–	*	*	–	*	–

For an explanation of “(a)” and “(b)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

*1: 3 when R0 is 0, 3 + (R0) in all other cases.
*2: 3 when R0 is 0, 4 + (R0) in all other cases.
*3: 3 when imm8 is 0, 3 + (imm8) in all other cases.
*4: 3 when imm8 is 0, 4 + (imm8) in all other cases.

Table 20 Branch 1 Instructions [31 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
BZ/BEQ rel	2	*1	0	Branch when (Z) = 1	–	–	–	–	–	–	–	–	–	–
BNZ/BNE rel	2	*1	0	Branch when (Z) = 0	–	–	–	–	–	–	–	–	–	–
BC/BLO rel	2	*1	0	Branch when (C) = 1	–	–	–	–	–	–	–	–	–	–
BNC/BHS rel	2	*1	0	Branch when (C) = 0	–	–	–	–	–	–	–	–	–	–
BN rel	2	*1	0	Branch when (N) = 1	–	–	–	–	–	–	–	–	–	–
BP rel	2	*1	0	Branch when (N) = 0	–	–	–	–	–	–	–	–	–	–
BV rel	2	*1	0	Branch when (V) = 1	–	–	–	–	–	–	–	–	–	–
BNV rel	2	*1	0	Branch when (V) = 0	–	–	–	–	–	–	–	–	–	–
BT rel	2	*1	0	Branch when (T) = 1	–	–	–	–	–	–	–	–	–	–
BNT rel	2	*1	0	Branch when (T) = 0	–	–	–	–	–	–	–	–	–	–
BLT rel	2	*1	0	Branch when (V) xor (N) = 1	–	–	–	–	–	–	–	–	–	–
BGE rel	2	*1	0	Branch when (V) xor (N) = 0	–	–	–	–	–	–	–	–	–	–
BLE rel	2	*1	0	((V) xor (N)) or (Z) = 1	–	–	–	–	–	–	–	–	–	–
BGT rel	2	*1	0	((V) xor (N)) or (Z) = 0	–	–	–	–	–	–	–	–	–	–
BLS rel	2	*1	0	Branch when (C) or (Z) = 1	–	–	–	–	–	–	–	–	–	–
BHI rel	2	*1	0	Branch when (C) or (Z) = 0	–	–	–	–	–	–	–	–	–	–
BRA rel	2	*1	0	Branch unconditionally	–	–	–	–	–	–	–	–	–	–
JMP @A	1	2	0	word (PC) ← (A)	–	–	–	–	–	–	–	–	–	–
JMP addr16	3	2	0	word (PC) ← addr16	–	–	–	–	–	–	–	–	–	–
JMP @ear	2	3	0	word (PC) ← (ear)	–	–	–	–	–	–	–	–	–	–
JMP @eam	2+	4+ (a)	(c)	word (PC) ← (eam)	–	–	–	–	–	–	–	–	–	–
JMPP @ear *3	2	3	0	word (PC) ← (ear), (PCB) ← (ear +2)	–	–	–	–	–	–	–	–	–	–
JMPP @eam *3	2+	4+ (a)	(d)	word (PC) ← (eam), (PCB) ← (eam +2)	–	–	–	–	–	–	–	–	–	–
JMPP addr24	4	3	0	word (PC) ← ad24 0 to 15 (PCB) ← ad24 16 to 23	–	–	–	–	–	–	–	–	–	–
CALL @ear *4	2	4	(c)	word (PC) ← (ear)	–	–	–	–	–	–	–	–	–	–
CALL @eam *4	2+	5+ (a)	2× (c)	word (PC) ← (eam)	–	–	–	–	–	–	–	–	–	–
CALL addr16 *5	3	5	(c)	word (PC) ← addr16	–	–	–	–	–	–	–	–	–	–
CALLV #vct4 *5	1	5	2× (c)	Vector call linstruction	–	–	–	–	–	–	–	–	–	–
CALLP @ear *6	2	7	2× (c)	word (PC) ← (ear) 0 to 15, (PCB) ← (ear) 16 to 23	–	–	–	–	–	–	–	–	–	–
CALLP @eam *6	2+	8+ (a)	*2	word (PC) ← (eam) 0 to 15, (PCB) ← (eam) 16 to 23	–	–	–	–	–	–	–	–	–	–
CALLP addr24 *7	4	7	2× (c)	word (PC) ← addr 0 to 15, (PCB) ← addr 16 to 23	–	–	–	–	–	–	–	–	–	–

For an explanation of “(a)”, “(c)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

*1: 3 when branching, 2 when not branching.
*2: 3 × (c) + (b)
*3: Read (word) branch address.
*4: W: Save (word) to stack; R: Read (word) branch address.
*5: Save (word) to stack.
*6: W: Save (long word) to W stack; R: Read (long word) branch address.
*7: Save (long word) to stack.

Table 21 Branch 2 Instructions [20 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
CBNE A, #imm8, rel	3	*1	0	Branch when byte (A) ≠ imm8	–	–	–	–	–	*	*	*	*	–
CWBNE A, #imm16, rel	4	*1	0	Branch when byte (A) ≠ imm16	–	–	–	–	–	*	*	*	*	–
CBNE ear, #imm8, rel	4	*1 *3	0	Branch when byte (ear) ≠ imm8	–	–	–	–	–	*	*	*	*	–
CBNE eam, #imm8, rel	4+	*1	(b)	Branch when byte (eam) ≠ imm8	–	–	–	–	–	*	*	*	*	–
CWBNE ear, #imm16, rel	5	*3	0	Branch when word (ear) ≠ imm16	–	–	–	–	–	*	*	*	*	–
CWBNE eam, #imm16, rel	5+	*2	(c)	Branch when word (eam) ≠ imm16	–	–	–	–	–	*	*	*	*	–
DBNZ ear, rel	3	*4	0	Branch when byte (ear) = (ear) – 1, and (ear) ≠ 0	–	–	–	–	–	*	*	*	–	–
DBNZ eam, rel	3+	*2	2× (b)	Branch when byte (ear) = (eam) – 1, and (eam) ≠ 0	–	–	–	–	–	*	*	*	–	*
DWBNZ ear, rel	3	*4	0	Branch when word (ear) = (ear) – 1, and (ear) ≠ 0	–	–	–	–	–	*	*	*	–	–
DWBNZ eam, rel	3+	14	2× (c)	Branch when word (eam) = (eam) – 1, and (eam) ≠ 0	–	–	–	–	–	*	*	*	–	*
INT #vct8	2	12 13	8× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
INT addr16	3	14	6× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
INTP addr24	4	9	6× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
INT9	1	11	8× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
RETI	1		6× (c)	Return from interrupt	–	–	*	*	*	*	*	*	*	–
RETIQ *6	2	6	*5	Return from interrupt	–	–	*	*	*	*	*	*	*	–
LINK #imm8	2	5	(c)	At constant entry, save old frame pointer to stack, set new frame pointer, and allocate local pointer area	–	–	–	–	–	–	–	–	–	–
UNLINK	1	4 5	(c)	At constant entry, retrieve old frame pointer from stack.	–	–	–	–	–	–	–	–	–	–
RET *7	1		(c)	Return from subroutine	–	–	–	–	–	–	–	–	–	–
RETP *8	1		(d)	Return from subroutine	–	–	–	–	–	–	–	–	–	–

For an explanation of “(b)”, “(c)” and “(d)”, refer to Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

*1: 4 when branching, 3 when not branching
*2: 5 when branching, 4 when not branching
*3: 5 + (a) when branching, 4 + (a) when not branching
*4: 6 + (a) when branching, 5 + (a) when not branching
*5: 3 × (b) + 2 × (c) when an interrupt request is generated, 6 × (c) when returning from the interrupt.
*6: High-speed interrupt return instruction. When an interrupt request is detected during this instruction, the instruction branches to the interrupt vector without performing stack operations when the interrupt is generated.
*7: Return from stack (word)
*8: Return from stack (long word)

Table 22 Other Control Instructions (Byte/Word/Long Word) [36 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
PUSHW A	1	3	(c)	word (SP) ← (SP) –2, ((SP)) ← (A)	–	–	–	–	–	–	–	–	–	–
PUSHW AH	1	3	(c)	word (SP) ← (SP) –2, ((SP)) ← (AH)	–	–	–	–	–	–	–	–	–	–
PUSHW PS	1	3	(c)	word (SP) ← (SP) –2, ((SP)) ← (PS)	–	–	–	–	–	–	–	–	–	–
PUSHW rlst	2	*3	*4	(SP) ← (SP) –2n, ((SP)) ← (rlst)	–	–	–	–	–	–	–	–	–	–
POPW A	1	3	(c)	word (A) ← ((SP)), (SP) ← (SP) +2	–	*	–	–	–	–	–	–	–	–
POPW AH	1	3	(c)	word (AH) ← ((SP)), (SP) ← (SP) +2	–	–	–	–	–	–	–	–	–	–
POPW PS	1	3	(c)	word (PS) ← ((SP)), (SP) ← (SP) +2	–	–	*	*	*	*	*	*	*	–
POPW rlst	2	*2	*4	(rlst) ← ((SP)) , (SP) ← (SP)	–	–	–	–	–	–	–	–	–	–
JCTX @A	1	9	6× (c)	Context switch instruction	–	–	*	*	*	*	*	*	*	–
AND CCR, #imm8	2	3	0	byte (CCR) ← (CCR) and imm8	–	–	*	*	*	*	*	*	*	–
OR CCR, #imm8	2	3	0	byte (CCR) ← (CCR) or imm8	–	–	*	*	*	*	*	*	*	–
MOV RP, #imm8	2	2	0	byte (RP) ← imm8	–	–	–	–	–	–	–	–	–	–
MOV ILM, #imm8	2	2	0	byte (ILM) ← imm8	–	–	–	–	–	–	–	–	–	–
MOVEA RWi, ear	2	3	0	word (RWi) ← ear	–	–	–	–	–	–	–	–	–	–
MOVEA RWi, eam	2+	2+ (a)	0	word (RWi) ← eam	–	–	–	–	–	–	–	–	–	–
MOVEA A, ear	2	2	0	word(A) ← ear	–	*	–	–	–	–	–	–	–	–
MOVEA A, eam	2+	1+ (a)	0	word (A) ← eam	–	*	–	–	–	–	–	–	–	–
ADDSP #imm8	2	3	0	word (SP) ← ext (imm8)	–	–	–	–	–	–	–	–	–	–
ADDSP #imm16	3	3	0	word (SP) ← imm16	–	–	–	–	–	–	–	–	–	–
MOV A, brgl	2	*1	0	byte (A) ← (brgl)	Z	*	–	–	–	*	*	–	–	–
MOV brg2, A	2	1	0	byte (brg2) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV brg2, #imm8	3	2	0	byte (brg2) ← imm8	–	–	–	–	–	*	*	–	–	–
NOP	1	1	0	No operation	–	–	–	–	–	–	–	–		–
ADB	1	1	0	Prefix code for AD space access	–	–	–	–	–	–	–	–	–	–
DTB	1	1	0	Prefix code for DT space access	–	–	–	–	–	–	–	–	–	–
PCB	1	1	0	Prefix code for PC space access	–	–	–	–	–	–	–	–	–	–
SPB	1	1	0	Prefix code for SP space access	–	–	–	–	–	–	–	–	–	–
NCC	1	1	0	Prefix code for no flag change	–	–	–	–	–	–	–	–	–	–
CMR	1	1	0	Prefix code for the common register bank	–	–	–	–	–	–	–	–	–	–
MOVW SPCU, #imm16	4	2	0	word (SPCU) ← (imm16)	–	–	–	–	–	–	–	–	–	–
MOVW SPCL, #imm16	4	2	0	word (SPCL) ← (imm16)	–	–	–	–	–	–	–	–	–	–
SETSPC	2	2	0	Stack check operation enable	–	–	–	–	–	–	–	–	–	–
CLRSPC	2	2	0	Stack check operation disable	–	–	–	–	–	–	–	–	–	–
BTSCN A	2	*5	0	byte (A) ← position of “1” bit in word (A)	Z	–	–	–	–	–	*	–	–	–
BTSCNS A	2	*6	0	byte (A) ← position of “1” bit in word (A) ×2	Z	–	–	–	–	–	*	–	–	–
BTSCND A	2	*7	0	byte (A) ← position of “1” bit in word (A) ×4	Z	–	–	–	–	–	*	–	–	–

For an explanation of “(a)” and “(c)”, refer to Tables 4 and 5.

*1: PCB, ADB, SSB, USB, and SPB: 1 cycle
DTB: 2 cycles
DPR: 3 cycles
*2: 3 + 4 × (pop count)
*3: 3 + 4 × (push count)
*4: Pop count × (c), or push count × (c)
*5: 3 when AL is 0, 5 when AL is not 0.
*6: 4 when AL is 0, 6 when AL is not 0.
*7: 5 when AL is 0, 7 when AL is not 0.

Table 23 Bit Manipulation Instructions [21 Instructions]

Mnemonic		#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVB	A, dir:bp	3	3	(b)	byte (A) ← (dir:bp) b	Z	*	–	–	–	*	*	–	–	–
MOVB	A, addr16:bp	4	3	(b)	byte (A) ← (addr16:bp) b	Z	*	–	–	–	*	*	–	–	–
MOVB	A, io:bp	3	3	(b)	byte (A) ← (io:bp) b	Z	*	–	–	–	*	*	–	–	–
MOVB	dir:bp, A	3	4	2× (b)	bit (dir:bp) b ← (A)	–	–	–	–	–	*	*	–	–	*
MOVB	addr16:bp, A	4	4	2× (b)	bit (addr16:bp) b ← (A)	–	–	–	–	–	*	*	–	–	*
MOVB	io:bp, A	3	4	2× (b)	bit (io:bp) b ← (A)	–	–	–	–	–	*	*	–	–	*
SETB	dir:bp	3	4	2× (b)	bit (dir:bp) b ← 1	–	–	–	–	–	–	–	–	–	*
SETB	addr16:bp	4	4	2× (b)	bit (addr16:bp) b ← 1	–	–	–	–	–	–	–	–	–	*
SETB	io:bp	3	4	2× (b)	bit (io:bp) b ← 1	–	–	–	–	–	–	–	–	–	*
CLRB	dir:bp	3	4	2× (b)	bit (dir:bp) b ← 0	–	–	–	–	–	–	–	–	–	*
CLRB	addr16:bp	4	4	2× (b)	bit (addr16:bp) b ← 0	–	–	–	–	–	–	–	–	–	*
CLRB	io:bp	3	4	2× (b)	bit (io:bp) b ← 0	–	–	–	–	–	–	–	–	–	*
BBC	dir:bp, rel	4	*1	(b)	Branch when (dir:bp) b = 0	–	–	–	–	–	–	*	–	–	–
BBC	addr16:bp, rel	5	*1	(b)	Branch when (addr16:bp) b = 0	–	–	–	–	–	–	*	–	–	–
BBC	io:bp, rel	4	*1	(b)	Branch when (io:bp) b = 0	–	–	–	–	–	–	*	–	–	–
BBS	dir:bp, rel	4	*1	(b)	Branch when (dir:bp) b = 1	–	–	–	–	–	–	*	–	–	–
BBS	addr16:bp, rel	5	*1	(b)	Branch when (addr16:bp) b = 1	–	–	–	–	–	–	*	–	–	–
BBS	io:bp, rel	4	*1	(b)	Branch when (io:bp) b = 1	–	–	–	–	–	–	*	–	–	–
SBBS	addr16:bp, rel	5	*2	2× (b)	Branch when (addr16:bp) b = 1, bit = 1	–	–	–	–	–	–	*	–	–	*
WBTS	io:bp	3	*3	*4	Wait until (io:bp) b = 1	–	–	–	–	–	–	–	–	–	–
WBTC	io:bp	3	*3	*4	Wait until (io:bp) b = 0	–	–	–	–	–	–	–	–	–	–

For an explanation of “(b)”, refer to Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

*1: 5 when branching, 4 when not branching
*2: 7 when condition is satisfied, 6 when not satisfied
*3: Undefined count
*4: Until condition is satisfied

Table 24 Accumulator Manipulation Instructions (Byte/Word) [6 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
SWAP	1	3	0	byte (A) 0 to 7 ← → (A) 8 to 15	–	–	–	–	–	–	–	–	–	–
SWAPW	1	2	0	word (AH) ← → (AL)	–	*	–	–	–	–	–	–	–	–
EXT	1	1	0	Byte code extension	X	–	–	–	–	*	*	–	–	–
EXTW	1	2	0	Word code extension	–	X	–	–	–	*	*	–	–	–
ZEXT	1	1	0	Byte zero extension	Z	–	–	–	–	R	*	–	–	–
ZEXTW	1	2	0	Word zero extension	–	Z	–	–	–	R	*	–	–	–

Table 25 String Instructions [10 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVS/MOVSI	2	*2	*3	Byte transfer @AH+ ← @AL+, counter = RW0	–	–	–	–	–	–	–	–	–	–
MOVSD	2	*2	*3	Byte transfer @AH– ← @AL–, counter = RW0	–	–	–	–	–	–	–	–	–	–
SCEQ/SCEQI	2	*1	*4	Byte retrieval @AH+ – AL, counter = RW0	–	–	–	–	–	*	*	*	*	–
SCEQD	2	*1	*4	Byte retrieval @AH– – AL, counter = RW0	–	–	–	–	–	*	*	*	*	–
FILS/FILSI	2	5m +3	*5	Byte filling @AH+ ← AL, counter = RW0	–	–	–	–	–	*	*	–	–	–
MOVSW/MOVSWI	2	*2	*6	Word transfer @AH+ ← @AL+, counter = RW0	–	–	–	–	–	–	–	–	–	–
MOVSWD	2	*2	*6	Word transfer @AH– ← @AL–, counter = RW0	–	–	–	–	–	–	–	–	–	–
SCWEQ/SCWEQI	2	*1	*7	Word retrieval @AH+ – AL, counter = RW0	–	–	–	–	–	*	*	*	*	–
SCWEQD	2	*1	*7	Word retrieval @AH– – AL, counter = RW0	–	–	–	–	–	*	*	*	*	–
FILSW/FILSWI	2	5m +3	*8	Word filling @AH+ ← AL, counter = RW0	–	–	–	–	–	*	*	–	–	–

m: RW0 value (counter value)

*1: 3 when RW0 is 0, 2 + 6 × (RW0) for count out, and 6n + 4 when match occurs
*2: 4 when RW0 is 0, 2 + 6 × (RW0) in any other case
*3: (b) × (RW0)
*4: (b) × n
*5: (b) × (RW0)
*6: (c) × (RW0)
*7: (c) × n
*8: (c) × (RW0)

Table 26 Multiple Data Transfer Instructions [18 Instructions]

Mnemonic	#	~	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVM @A, @RLi, #imm8	3	*1	*3	Multiple data trasfer byte ((A)) ← ((RLi))	–	–	–	–	–	–	–	–	–	–
MOVM @A, eam, #imm8	3+	*2	*3	Multiple data trasfer byte ((A)) ← (eam)	–	–	–	–	–	–	–	–	–	–
MOVM addr16, @RLi, #imm8	5	*1	*3	Multiple data trasfer byte (addr16) ← ((RLi))	–	–	–	–	–	–	–	–	–	–
MOVM addr16, eam, #imm8	5+	*2	*3	Multiple data trasfer byte (addr16) ← (eam)	–	–	–	–	–	–	–	–	–	–
MOVMW @A, @RLi, #imm8	3	*1	*4	Multiple data trasfer word ((A)) ← ((RLi))	–	–	–	–	–	–	–	–	–	–
MOVMW @A, eam, #imm8	3+	*2	*4	Multiple data trasfer word ((A)) ← (eam)	–	–	–	–	–	–	–	–	–	–
MOVMW addr16, @RLi, #imm8	5	*1	*4	Multiple data trasfer word (addr16) ← ((RLi))	–	–	–	–	–	–	–	–	–	–
MOVMW addr16, eam, #imm8	5+	*2	*4	Multiple data trasfer word (addr16) ← (eam)	–	–	–	–	–	–	–	–	–	–
MOVM @RLi, @A, #imm8	3	*1	*3	Multiple data trasfer byte ((RLi)) ← ((A))	–	–	–	–	–	–	–	–	–	–
MOVM eam, @A, #imm8	3+	*2	*3	Multiple data trasfer byte (eam) ← ((A))	–	–	–	–	–	–	–	–	–	–
MOVM @RLi, addr16, #imm8	5	*1	*3	Multiple data transfer byte ((RLi)) ← (addr16)	–	–	–	–	–	–	–	–	–	–
MOVM eam, addr16, #imm8	5+	*2	*3	Multiple data transfer byte (eam) ← (addr16)	–	–	–	–	–	–	–	–	–	–
MOVMW @RLi, @A, #imm8	3	*1	*4	Multiple data trasfer word ((RLi)) ← ((A))	–	–	–	–	–	–	–	–	–	–
MOVMW eam, @A, #imm8	3+	*2	*4	Multiple data trasfer word (eam) ← ((A))	–	–	–	–	–	–	–	–	–	–
MOVMW @RLi, addr16, #imm8	5	*1	*4	Multiple data transfer word ((RLi)) ← (addr16)	–	–	–	–	–	–	–	–	–	–
MOVMW eam, addr16, #imm8	5+	*2	*4	Multiple data transfer word (eam) ← (addr16)	–	–	–	–	–	–	–	–	–	–
MOVM bnk : addr16, *5 bnk : addr16, #imm8	7	*1	*3	Multiple data transfer byte (bnk:addr16) ← (bnk:addr16)	–	–	–	–	–	–	–	–	–	–
MOVMW bnk : addr16, *5 bnk : addr16, #imm8	7	*1	*4	Multiple data transfer word (bnk:addr16) ← (bnk:addr16)	–	–	–	–	–	–	–	–	–	–

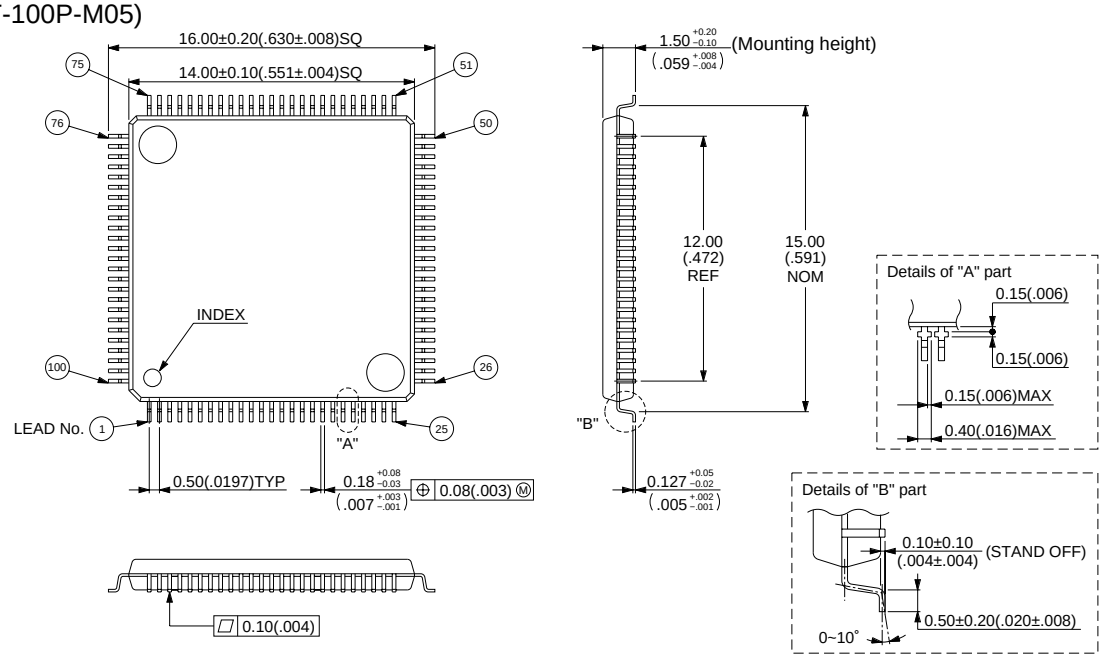
*1: 5 + imm8 × 5, 256 times when imm8 is zero.
*2: 5 + imm8 × 5 + (a), 256 times when imm8 is zero.
*3: Number of transfers × (b) × 2
*4: Number of transfers × (c) × 2
*5:The bank register specified by “bnk” is the same as for the MOVS instruction.

■ ORDERING INFORMATION

Model	Package	Remarks
MB90233PFV-XXX MB90234PFV-XXX	100-pin Plastic LQFP (FPT-100P-M05)	
MB90234PFV	100-pin Plastic LQFP (FPT-100P-M05)	Only ES
MB90W234ZFV	100-pin Ceramic SQFP (FPT-100C-C01)	Only ES

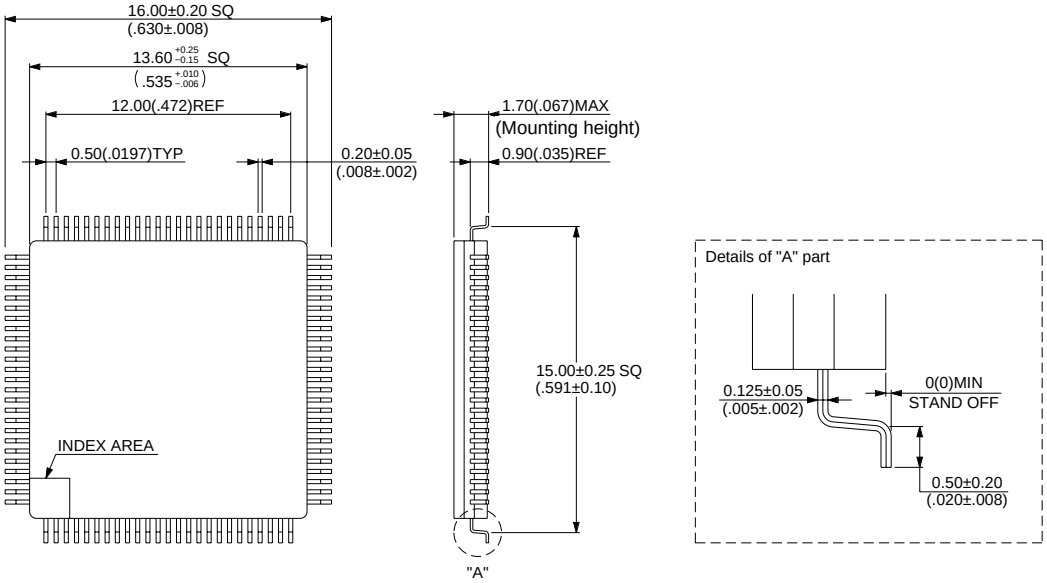
■ PACKAGE DIMENSIONS

100-pin Plastic LQFP
(FPT-100P-M05)



Dimensions in mm (inches)

100-pin Ceramic LQFP
(FPT-100C-C01)



Dimensions in mm (inches)

FUJITSU LIMITED

For further information please contact:

Japan
FUJITSU LIMITED
Corporate Global Business Support Division
Electronic Devices
KAWASAKI PLANT, 4-1-1, Kamikodanaka
Nakahara-ku, Kawasaki-shi
Kanagawa 211-8588, Japan
Tel: 81(44) 754-3763
Fax: 81(44) 754-3329

http://www.fujitsu.co.jp/

North and South America
FUJITSU MICROELECTRONICS, INC.
Semiconductor Division
3545 North First Street
San Jose, CA 95134-1804, USA
Tel: (408) 922-9000
Fax: (408) 922-9179

Customer Response Center
Mon. - Fri.: 7 am - 5 pm (PST)
Tel: (800) 866-8608
Fax: (408) 922-9179

http://www.fujitsumicro.com/

Europe
FUJITSU MIKROELEKTRONIK GmbH
Am Siebenstein 6-10
D-63303 Dreieich-Buchschlag
Germany
Tel: (06103) 690-0
Fax: (06103) 690-122

http://www.fujitsu-ede.com/

Asia Pacific
FUJITSU MICROELECTRONICS ASIA PTE LTD
#05-08, 151 Lorong Chuan
New Tech Park
Singapore 556741
Tel: (65) 281-0770
Fax: (65) 281-0220

http://www.fmap.com.sg/

F9901

The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering.

The information and circuit diagrams in this document are presented as examples of semiconductor device applications, and are not intended to be incorporated in devices for actual use. Also, FUJITSU is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of this information or circuit diagrams.

FUJITSU semiconductor devices are intended for use in standard applications (computers, office automation and other office equipment, industrial, communications, and measurement equipment, 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.

Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.

If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.