FUJITSU SEMICONDUCTOR
DATA SHEET

DS07-13604-2E

16-bit Proprietary Microcontroller

CMOS

F²MC-16L MB90660A Series

MB90662A/663A/P663A

■ DESCRIPTION

MB90660A series microcontrollers are 16-bit microcontrollers optimized for high speed realtime processing of consumer equipment and system control of air conditioner video cameras, VCRs, and copiers. Based on the F²MC*-16 CPU core, an F²MC-16L is used as the CPU. This CPU includes high-level language-support instructions and robust task switching instructions, and additional addressing modes.

Microcontrollers in this series have built-in peripheral resources including multi-function timers, 16-bit reload timer four channels, 8-bit PWM one channel, UART one channel, 10-bit A/D eight converter channels, and external interrupt eight channels.

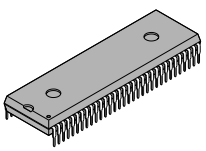
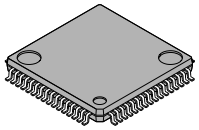
*: F²MC stands for FUJITSU Flexible Microcontroller.

■ FEATURES

- F²MC-16L CPU
- Minimum execution time: 62.5 ns/4 MHz oscillation (uses PLL multiplier): fastest speed at quadruple operation
- Instruction set optimized for controller applications
  Upward compatibility at object level with the F²MC-16(H)
  Various data types (bit, byte, word, long-word)
  Higher speed due to review of instruction cycle
  Expanded addressing modes: 23 types
  High coding efficiency
  Two access methods (bank system or linear pointer)
  Improved multiply-and-divide instructions (additional signed instructions)
  Improved high-precision operation with 32-bit accumulator
  Extended intelligent I/O services (access area extended by 64 Kbytes)
  Large memory space: 16 Mbytes

(Continued)

■ PACKAGE

64-pin Plastic SH-DIP	64-pin Plastic LQFP
	
(DIP-64P-M01)	(FPT-64P-M09)

(Continued)

- Improved instruction set applicable to high-level language (C) and multitasking
  System stack pointer
  Improved indirect instructions using various pointers
  Barrel shift instruction
  Stack check function
- Improved execution speed: 4-byte instruction queue
- Improved interrupt functions
- Automatic data transfer function independent of CPU

Peripheral Resources

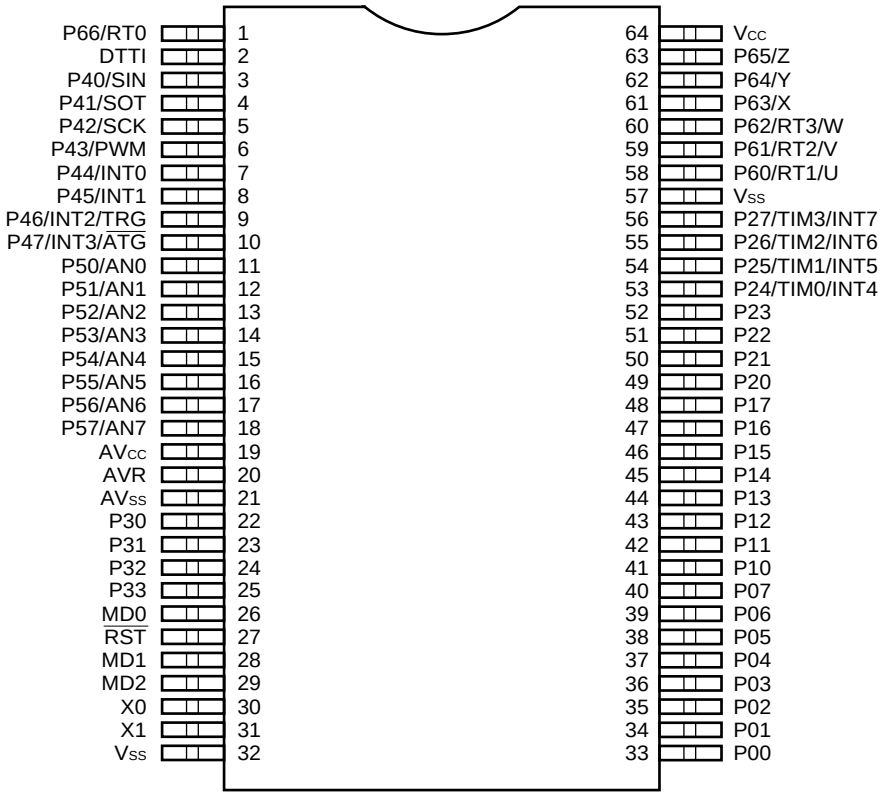
- ROM: 16 Kbytes (MB90661A)
  32 Kbytes (MB90662A)
  48 Kbytes (MB90663A)
  One-time PROM: 48 Kbytes (MB90P663A)
- RAM: 512 bytes (MB90661A)
  1.64 Kbytes (MB90662A)
  2 Kbytes (MB90663A/MB90P663A)
- General-purpose ports: Max. 51
- UART: 1 channel
  Can be used for both asynchronous transfer and clocked serial (I/O extended serial) communications
- A/D converter: 10-bit, 8 channels
  Includes 8-bit conversion mode
- 16-bit reload timer: 4 channels
- 8-bit PWM: 1 channel
- External interrupts: 8 channels
- 18-bit timebase timer with watchdog timer function
- PLL clock multiplier function
- CPU intermittent operation function
- Various standby modes
- Package: SH-DIP-64/LQFP-64 (0.65-mm pitch)
- CMOS technology

■ PRODUCT LINEUP

Parameter \ Part number	MB90P663A	MB90662A	MB90663A
Classification	OTPROM	MASK ROM	MASK ROM
ROM size	48 Kbytes	32 Kbytes	48 Kbytes
RAM size	2 Kbytes	1.64 Kbytes	2 Kbytes
CPU functions	Number of basic instructions : 340 Instruction bit length : 8/16 bits Instruction length : 1 to 7 bytes Data bit length : 1, 4, 8, 16, or 32 bits Minimum execution time : 62.5 ns/4 MHz (PLL 4 multiply) Interrupt processing time : 1000 ns/16 MHz (minimum)		
Ports	Input Ports : 4 I/O ports (CMOS) : 39 I/O ports (N channel open-drain) : 8 Total : 51		
Packages	DIP-64P-M01 FPT-64P-M09	DIP-64P-M01 FPT-64P-M09	DIP-64P-M01 FPT-64P-M09
Multi-Function Timer	14-bit up/down count timer × 1, buffered compare register × 4, buffered compare clear register, zero detect terminal control, 4 output channels, non-overlapped 3-phase waveform output, 3-phase independent dead time timer, 4-bit carrier counter		
UART	Full duplex double buffer Selectable clock synchronous/asynchronous operation Built-in dedicated baud rate generator (During asynchronous operation: 62500, 31250, 19230, 9615, 4808, 2404, 1202 bps)		
A/D Converter	10-bit precision × 8 channels A/D conversion time : 6.13 µs (98 machine cycles at 16 MHz machine clock, includes sample hold time) Startup trigger : Startup by software, external source, or multi-function timer output (RT0) can be selected Activiation : Single, scan (multiple channel continuous), continuous (1 channel continuous), stop (synchronized with conversion start in scan mode)		
16-Bit Reload Timer	16-bit reload timer operation (toggle output, one-shot output selectable) (Count clock can be selected from 0.125 µs, 0.5 µs, or 2.0 µs at 16 MHz machine cycle) Event count function selectable 4 channels built-in		
8-Bit PWM	8-bit resolution PWM operation (arbitrary cycle: duty ratio pulse output) (Count clock can be selected from 0.125 µs or 64.0 µs at 16 MHz machine cycle)		
External Interrupts	Number of inputs: 8 External interrupt mode (Interrupts can be generated by four types of request detect sources)		
PLL Function	1/2/3/4-time multiplier can be selected (Please set so as not to exceed guaranteed operation frequency)		
Miscellaneous Items	V_{PP} is shared with MD2 terminal (when writing the EPROM)	—	—

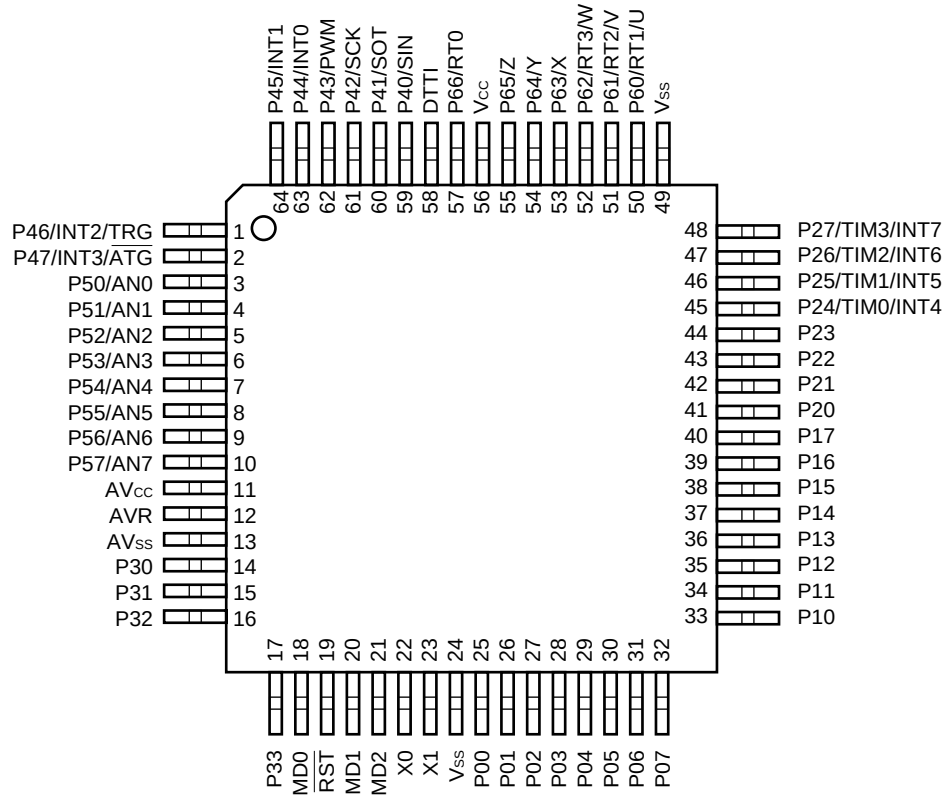
■ PIN ASSIGNMENT

(TOP VIEW)



(DIP-64P-M01)

(TOP VIEW)



(FPT-64P-M09)

■ PIN DESCRIPTION

Pin no.		Pin name	Circuit type	Function
SH-DIP*1	LQFP*2			
30	22	X0	A (Oscillator)	Crystal oscillator pin (32 MHz).
31	23	X1		
33 to 40	25 to 32	P00 to P07	B (CMOS)	General-purpose I/O ports.
41 to 48	33 to 40	P10 to P17	B (CMOS)	General-purpose I/O ports.
49 to 52	41 to 44	P20 to P23	B (CMOS)	General-purpose I/O ports.
53 to 56	45 to 48	P24 to P27	G (CMOS)	General-purpose I/O ports. This function is activated when the output specification of the reload timer is “disabled”.
		TIM0 to TIM3		I/O pins for reload timers 0 to 4. Input is used only as necessary while serving as input for the reload timer. It is therefore necessary to stop output beforehand using other functions unless intentionally used otherwise. Their function as output terminals for the reload timer is activated when the output specification is “enabled”.
		INT4 to INT7		External interrupt request input pins. Input is used only as necessary while external interrupts are enabled. It is therefore necessary to stop output beforehand using other functions unless intentionally used otherwise.
22 to 25	14 to 17	P30 to P33	B (CMOS)	General-purpose I/O ports.
3	59	P40	E (CMOS/H)	General-purpose I/O port. This function is always enabled.
		SIN		UART serial data input pin. Input is used only as necessary while serving as UART input. It is therefore necessary to stop output beforehand using other functions unless intentionally used otherwise.
4	60	P41	E (CMOS/H)	General-purpose I/O port. This function is activated when the serial data output specification of the UART is “disabled”.
		SOT		UART serial data output pin. This function is activated when the serial data output specification of the UART is “enabled”.

(Continued)

*1: DIP-64P-M01
*2: FPT-64P-M09

Pin no.		Pin name	Circuit type	Function
SH-DIP*1	LQFP*2			
5	61	P42	E (CMOS/H)	General-purpose I/O port. This function is activated when the clock output specification of the UART is “disabled”.
		SCK		UART clock I/O pin. This function is activated when the clock output specification of the UART is “enabled”. Input is used only as necessary while serving as UART input. It is therefore necessary to stop output beforehand using other functions unless intentionally used otherwise.
6	62	P43	E (CMOS/H)	General-purpose I/O port. This function is activated when the output specification of the PWM is “disabled”.
		PWM		PWM timer output pin. This function is activated when the waveform output specification of the PWM timer is “enabled”.
7 8	63 64	P44 to P45	D (CMOS/H)	General-purpose I/O ports. This function is always active.
		INT0 to INT1		External interrupt request input pins. Input is used only as necessary while external interrupts are enabled.
9	1	P46	D (CMOS/H)	General-purpose input port. This function is always active.
		INT2		External interrupt request input pin. Input is used only as necessary while external interrupts are enabled.
		TRG		Timer clear trigger input pin for multi-function timer. Input is used only as necessary while multi-function timer input is enabled.
10	2	P47	D (CMOS/H)	General-purpose input port. This function is always active.
		INT3		External interrupt request input pin. Input is used only as necessary while external interrupts are enabled.
		$\overline{\text{ATG}}$		Trigger input pin for the A/D converter. Input is used only as necessary while the A/D converter is performing input.
11 to 18	3 to 10	P50 to P57	C (AD)	Open-drain type I/O ports. This function is enabled when the analog input enable register specification is “port”.
		AN0 to AN7		Analog input pins for the A/D converter. This function is enabled when the analog input enable register specification is “AD”.

*1: DIP-64P-M01
*2: FPT-64P-M09

(Continued)

Pin no.		Pin name	Circuit type	Function
SH-DIP*1	LQFP*2			
58	50	P60	E (CMOS/H)	General-purpose I/O port. This function is enabled when the multi-function timer waveform output specification is “disabled” and the 3-phase waveform output specification is “disabled”.
		RT1		Multi-function timer waveform output pin. This function is enabled when the multi-function timer output specification is “enabled”.
		U		3-phase waveform output pin. This function is enabled when the 3-phase waveform output specification is “enabled”.
59	51	P61	E (CMOS/H)	General-purpose I/O port. This function is enabled when the multi-function timer waveform output specification is “disabled” and the 3-phase waveform output specification is “disabled”.
		RT2		Multi-function timer waveform output pin. This function is enabled when the multi-function timer output specification is “enabled”.
		V		3-phase waveform output pin. This function is enabled when the 3-phase waveform output specification is “enabled”.
60	52	P62	E (CMOS/H)	General-purpose I/O port. This function is enabled when the multi-function timer waveform output specification is “disabled” and the 3-phase waveform output specification is “disabled”.
		RT3		Multi-function timer waveform output pin. This function is enabled when the multi-function timer output specification is “enabled”.
		W		3-phase waveform output pin. This function is enabled when the 3-phase waveform output specification is “enabled”.
61	53	P63	E (CMOS/H)	General-purpose I/O port. This function is enabled when the 3-phase waveform output specification is “disabled”.
		X		3-phase waveform output pin. This function is enabled when the 3-phase waveform output specification is “enabled”.
62	54	P64	E (CMOS/H)	General-purpose I/O port. This function is enabled when the 3-phase waveform output specification is “disabled”.
		Y		3-phase waveform output pin. This function is enabled when the 3-phase waveform output specification is “enabled”.

(Continued)

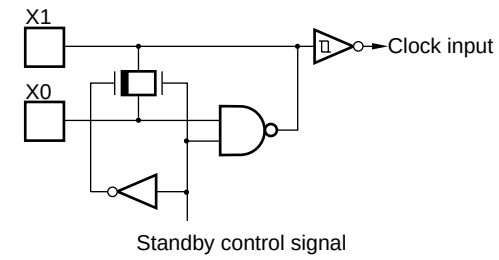
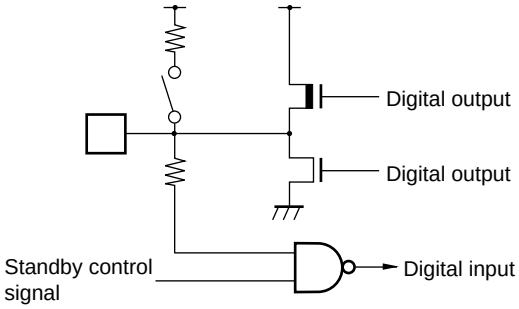
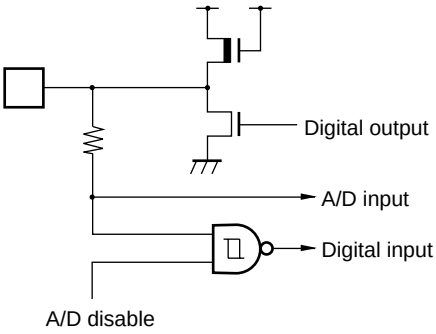
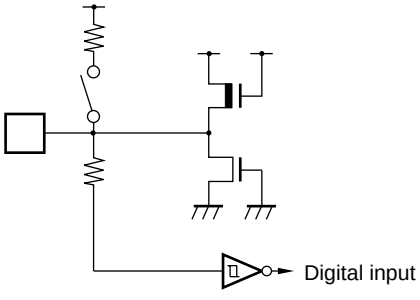
*1: DIP-64P-M01
*2: FPT-64P-M09

(Continued)

Pin no.		Pin name	Circuit type	Function
SH-DIP*1	LQFP*2			
63	55	P65	E (CMOS/H)	General-purpose I/O port. This function is enabled when the 3-phase waveform output specification is “disabled”.
		Z		3-phase waveform output pin. This function is enabled when the 3-phase waveform output specification is “enabled”.
1	57	P66	E (CMOS/H)	General-purpose I/O port. This function is enabled when the multi-function timer waveform output specification is “disabled”.
		RT0		Multi-function timer waveform output pin. This function is enabled when the multi-function timer output specification is “enabled”.
2	58	DTTI	D (CMOS/H)	3-phase waveform output disable input (DTTI) pin.
19	11	AV_{CC}	Power supply	Power supply for analog circuits. Turn this power supply on/off by applying a voltage level greater than AV_{CC} to V_{CC}.
20	12	AVR	Power supply	Reference power supply for analog circuits. Turn this pin on/off by applying a voltage level greater than AVR to AV_{CC}.
21	13	AV_{SS}	Power supply	Ground level for analog circuits.
26 28 29	18 20 21	MD0 to MD2	F (CMOS/H)	Input pins for specifying operation mode. Use these pins by directly connecting to V_{CC} or V_{SS}.
27	19	$\overline{RST}$	D (CMOS/H)	External reset request input pin.
64	56	V_{CC}	Power supply	Power supply for digital circuits.
32 57	24 49	V_{SS}	Power supply	Ground level for digital circuits.

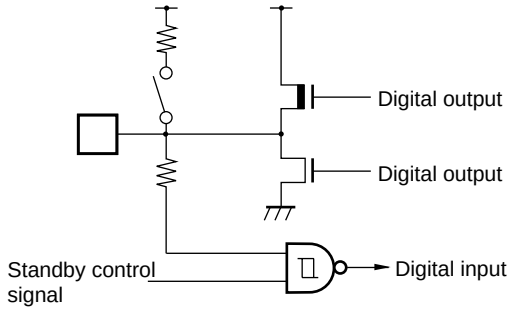
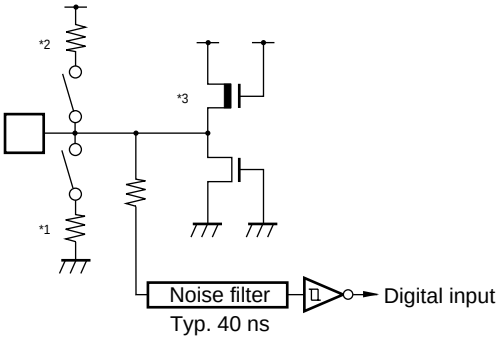
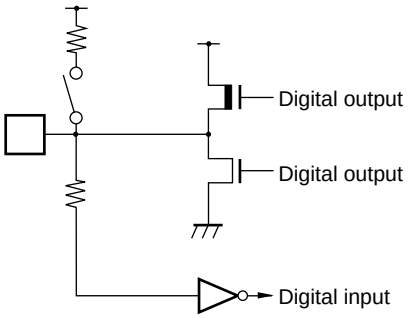
*1: DIP-64P-M01
*2: FPT-64P-M09

■ I/O CIRCUIT TYPE

Type	Circuit	Remarks
A		• 3 MHz to 32 MHz operation • Oscillation feedback resistor: Approx. 1 MΩ
B		• CMOS level input and output With standby control • Pull-up option can be selected With standby control
C		• N-channel open-drain output CMOS level hysteresis input With A/D control
D		• CMOS level hysteresis input Without standby control • Pull-up option can be selected Without standby control

(Continued)

(Continued)

Type	Circuit	Remarks
E		• CMOS level output • CMOS level hysteresis input With standby control • Pull-up option can be selected With standby control
F		• CMOS level input (Mask ROM version uses CMOS hysteresis input) Without standby control • Pull-up option can be selected for MD2 (*1) Pull-up option can be selected for MD1/0 (*2) Both without standby option • The MB90P663A does not include a noise filter. It also does not have a P channel protect Tr (*3) for the MD2 pin or pull-down.
G		• CMOS level input and output Without standby control • Pull-up option can be selected With standby control

■ HANDLING DEVICES

1. Preventing Latchup

Latchup may occur with 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 which 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.

To prevent the similar aftereffects, use also the utmost care not to allow the analog supply voltage to exceed the digital supply voltage.

2. Treatment of Unused Input Pins

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

3. External Reset Input

When resetting by inputting "L" level to the $\overline{RST}$ pin, the "L" level must be input for at least 5 machine cycles to ensure that internal reset has occurred. Be aware of this point when using external clock input.

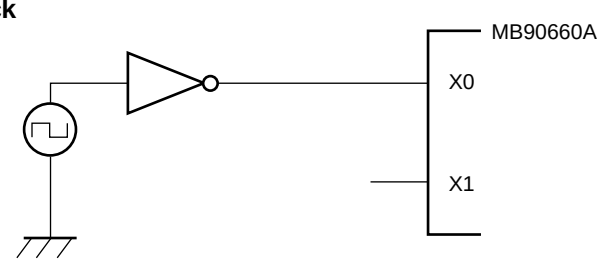
4. V_{CC}, V_{SS} Pin

Be sure that both V_{CC} and V_{SS} are at the same voltage.

5. Notes on Using an External Clock

Drive X0 when using an external clock.

- **Using an External Clock**



6. Order of Power-on to A/D Converter and Analog Inputs

Power-off (AV_{CC}, AVR) to the digital power supply (V_{CC}) must be performed only after the A/D converter and the analog inputs (AN0 to AN7) has been turned on.

Turning on or off should always be performed keeping AVR below AV_{CC}.

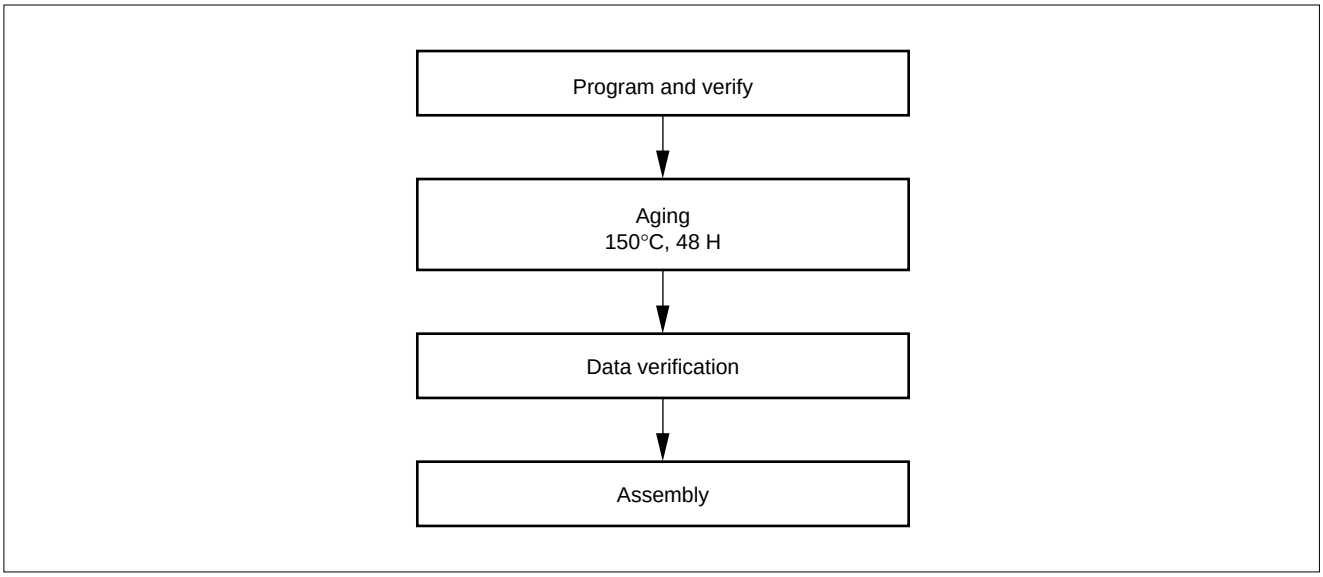
Use caution for the input voltage not to exceed AV_{CC} when the pin sharing the analog input for its function is used as an input port.

7. Programming Mode

When the MB90P663A is shipped from Fujitsu, all bits (48 K × 8 bits) are set to "1". Program by setting selected bits to "0" when you wish to set the data. Note that "1" cannot be programming electrically.

8. Recommended Screening Conditions

High-temperature aging is recommended as the pre-assembly screening procedure for a product with a blanked OTPROM microcomputer program.



9. Programming Yields

All bits cannot be programmed at Fujitsu shipping test to a blanked OTPROM microcomputer, due to its nature. For this reason, a programming yield of 100% cannot be assured at all times.

10. Fluctuations in Supply Voltage

Although the assured V_{CC} supply voltage operating range is as specified, sudden fluctuations even within this range may cause a malfunction. Therefore, the voltage supply to the IC should be kept as constant as possible. The V_{CC} ripple (P-P value) at the supply frequency (50 to 60 Hz) should be less than 10% of the typical V_{CC} value, or the coefficient of excessive variation should not be more than 0.1 V/ms instantaneous change when power is supplied.

■ PROGRAMMING THE MB90P663A EPROM

Since the MB90P663A is functionally equivalent to the MBM27C1000 when it is in EPROM mode, it is possible to program them with a general-purpose EPROM programmer by using a special adaptor socket.

However, the MB90660A does not support the electronic signature (device ID code) mode.

1. Pin Assignment in EPROM Mode

- **MBM27C1000-compatible pins**

MBM27C1000		MB90P663A		
Pin no.	Pin name	Pin no.		Pin name
		SH-DIP	LQFP	
1	V_{PP}	29	21	MD2 (V_{PP})
2	$\overline{OE}$	24	16	P32
3	A15	48	40	P17
4	A12	45	37	P14
5	A07	56	48	P27
6	A06	55	47	P26
7	A05	54	46	P25
8	A04	53	45	P24
9	A03	52	44	P23
10	A02	51	43	P22
11	A01	50	42	P21
12	A00	49	41	P20
13	D00	33	25	P00
14	D01	34	26	P01
15	D02	35	27	P02
16	GND	—	—	—

MBM27C1000		MB90P663A		
Pin no.	Pin name	Pin no.		Pin name
		SH-DIP	LQFP	
32	V_{CC}	64	56	V_{CC}
31	$\overline{PGM}$	25	17	P33
30	NC	—	—	—
29	A14	47	39	P16
28	A13	46	38	P15
27	A08	41	33	P10
26	A09	42	34	P11
25	A11	44	36	P13
24	A16	22	14	P30
23	A10	43	35	P12
22	$\overline{CE}$	23	15	P31
21	D07	40	32	P07
20	D06	39	31	P06
19	D05	38	30	P05
18	D04	37	29	P04
17	D03	36	28	P03

- **Power supply, GND connection pins**

Type	Pin no.		Pin name
	SH-DIP	LQFP	
Power	2	58	DTTI
	64	56	V_{CC}
GND	57	49	V_{SS}
	21	13	AV_{SS}
	27	19	$\overline{RST}$
	32	24	V_{SS}
	26	18	MD0
	3	59	P40
	4	60	P41
	5	61	P42

- **Pins other than MBM27C1000-compatible pins**

Pin no.		Pin name	Processing
SH-DIP	LQFP		
30 28	22 20	X0 MD1	Pull-up by 4.7 KΩ
31	23	X1	OPEN
9 10 11 to 18 19 20 58 to 63 1 6 to 8	1 2 3 to 10 11 12 50 to 55 57 62 to 64	P46 P47 P50 to P57 AVcc AVR P60 to P65 P66 P43 to P45	1 MΩ-level pull-up resistor connected to each pin

2. EPROM Programmer Socket Adapter and Recommended Programmer Manufacturer

Part no.	Package	Compatible socket adapter Sun Hayato Co., Ltd.	Recommended programmer manufacturer and programmer name		
			Minato Electronics Inc.	Data I/O Co., Ltd.	Advantest Corp.
MB90P663AP	SH-DIP-64	ROM-64SD-32DP-16L	Recommended	Recommended	Recommended
MB90P663APF	LQFP-64	ROM-64SF-32DP-16L	Recommended	Recommended	Recommended

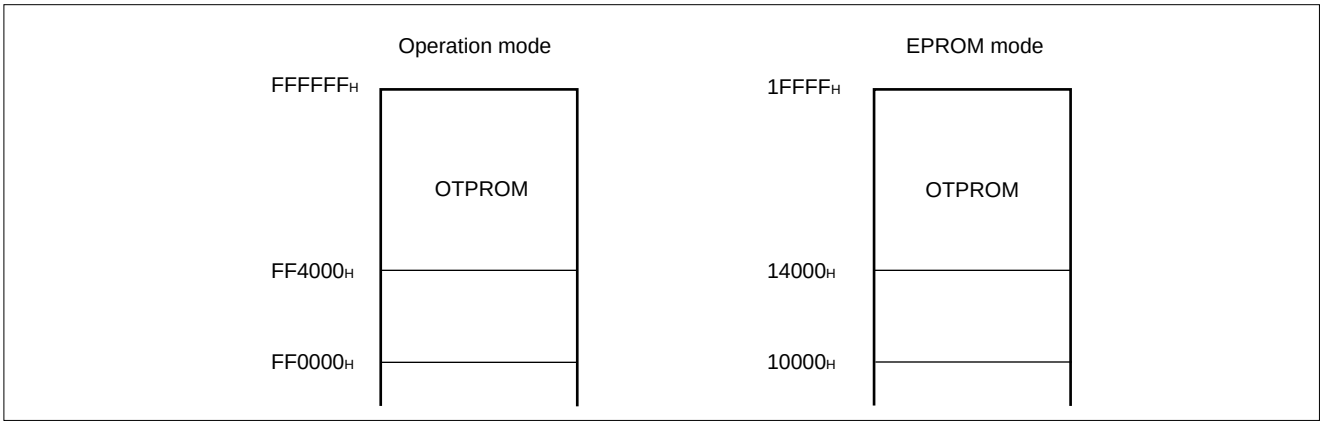
Inquiry: Sun Hayato Co., Ltd.: TEL (81)-3-3986-0403
FAX (81)-3-5396-9106
Minato Electronics Inc.: TEL: USA (1)-916-348-6066
JAPAN (81)-45-591-5611
Data I/O Co., Ltd.: TEL: USA/ASIA (1)-206-881-6444
EUROPE (49)-8-985-8580
Advantest Corp.: TEL: Except JAPAN (81)-3-3930-4111

3. Programming Data

(1) Adjust the EPROM programmer to settings for the MBM27C1000.

(2) Load program data from addresses 10000_H to $1FFFF_H$ in the EPROM programmer.

OTPROM addresses $FF4000_H$ to $FFFFFF_H$ of the MB90P663A in operation mode correspond to addresses 14000_H to $1FFFF_H$ in EPROM mode.



(3) Set the MB90P663A into the adaptor socket and install the adaptor socket into the EPROM programmer. Pay attention to the orientation of the device and the adaptor socket at this time.

(4) Programming data to the EPROM.

(5) If data cannot be programmed, try again with a 0.1 µF capacitor connected between V_{CC} and GND and V_{PP} and GND.

Note: Since Mask ROM products (MB90662A/663A) do not include an EPROM mode, data cannot be read-out using an EPROM programmer.

4. PROM Option Bitmap

The programming method is the same as a PROM, and can be set by programming values to addresses indicated in the memory map.

The following bit map shows the relation between bits and options.

- **PROM Option Bitmap**

Address \ Bit	7	6	5	4	3	2	1	0
00004H	P07 Pull-up 1: No 0: Yes	P06 Pull-up 1: No 0: Yes	P05 Pull-up 1: No 0: Yes	P04 Pull-up 1: No 0: Yes	P03 Pull-up 1: No 0: Yes	P02 Pull-up 1: No 0: Yes	P01 Pull-up 1: No 0: Yes	P00 Pull-up 1: No 0: Yes
00008H	P17 Pull-up 1: No 0: Yes	P16 Pull-up 1: No 0: Yes	P15 Pull-up 1: No 0: Yes	P14 Pull-up 1: No 0: Yes	P13 Pull-up 1: No 0: Yes	P12 Pull-up 1: No 0: Yes	P11 Pull-up 1: No 0: Yes	P10 Pull-up 1: No 0: Yes
0000CH	P27 Pull-up 1: No 0: Yes	P26 Pull-up 1: No 0: Yes	P25 Pull-up 1: No 0: Yes	P24 Pull-up 1: No 0: Yes	P23 Pull-up 1: No 0: Yes	P22 Pull-up 1: No 0: Yes	P21 Pull-up 1: No 0: Yes	P20 Pull-up 1: No 0: Yes
00010H	P43 Pull-up 1: No 0: Yes	P42 Pull-up 1: No 0: Yes	P41 Pull-up 1: No 0: Yes	P40 Pull-up 1: No 0: Yes	P33 Pull-up 1: No 0: Yes	P32 Pull-up 1: No 0: Yes	P31 Pull-up 1: No 0: Yes	P30 Pull-up 1: No 0: Yes
00014H *1	P47 Pull-up 1: No 0: Yes	P46 Pull-up 1: No 0: Yes	P45 Pull-up 1: No 0: Yes	P44 Pull-up 1: No 0: Yes	$\overline{\text{RST}}$ Pull-up 1: No 0: Yes	DTTI Pull-up 1: No 0: Yes	Accept asyn-chronous reset 1: Yes 0: No	MD1/MD0*2 Pull-up 1: No 0: Yes
00018H	Open	P66 Pull-up 1: No 0: Yes	P65 Pull-up 1: No 0: Yes	P64 Pull-up 1: No 0: Yes	P63 Pull-up 1: No 0: Yes	P62 Pull-up 1: No 0: Yes	P61 Pull-up 1: No 0: Yes	P60 Pull-up 1: No 0: Yes

Initially (value when blank), all bits are “1”.

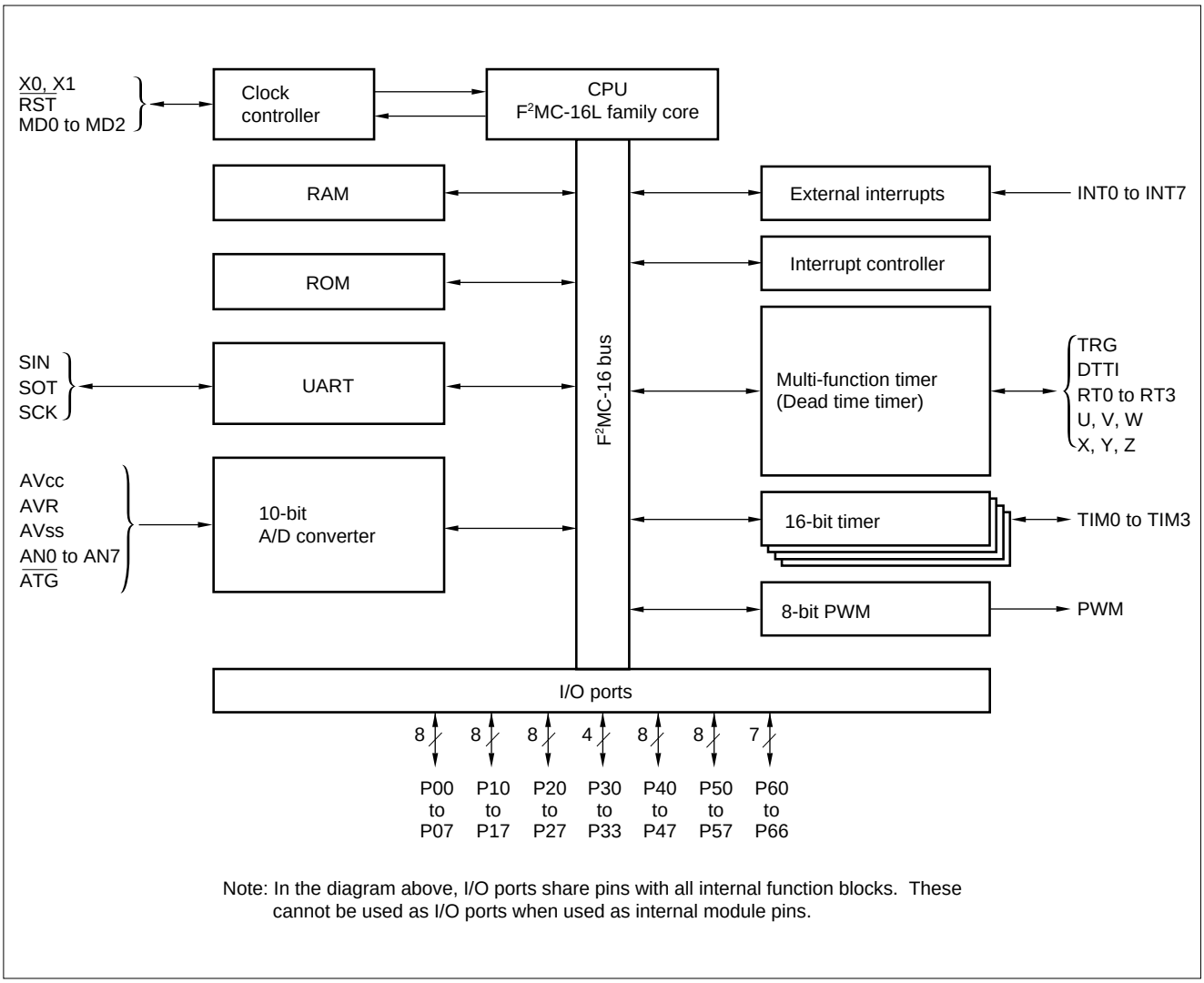
*1: Under this release, the pull-up resistor is cut-off during stop mode for pins for which the pull-up option was selected. (Pins for which the circuit type shown in the “■ Pin Description” is B or E.)
However, the pull-up resistor is not cut-off even in stop mode for P44 to 47, $\overline{\text{RST}}$, DTTI (pins for which the circuit type shown in the “■ Pin Description” is D or G), and MD1 and MD0.

*2: Whether or not a pull-up/pull-down resistor is present for MD2, MD1 and MD0 is determined as follows. If pull-up/pull-down resistor is selected, it is included with all 2 (or 3) pins. Presence or absence of the pull-up or pull-down resistors for the mode terminal cannot be selected for each pin.

Pin	MB90P663A	MB90663A/2A
MD2	No	Pull-down can be selected
MD1	With pull-up resistor	With pull-up resistor
MD0	With pull-up resistor	With pull-up resistor

Notes: • “FFH” must be set to addresses no defined in the table above.
• Since the option setting for the MB90P663A takes 8 machine cycles, the option setting is not made until a clock is provided after power-on. (This results in no pull-up for all pins, and asynchronous reset input is accepted.)

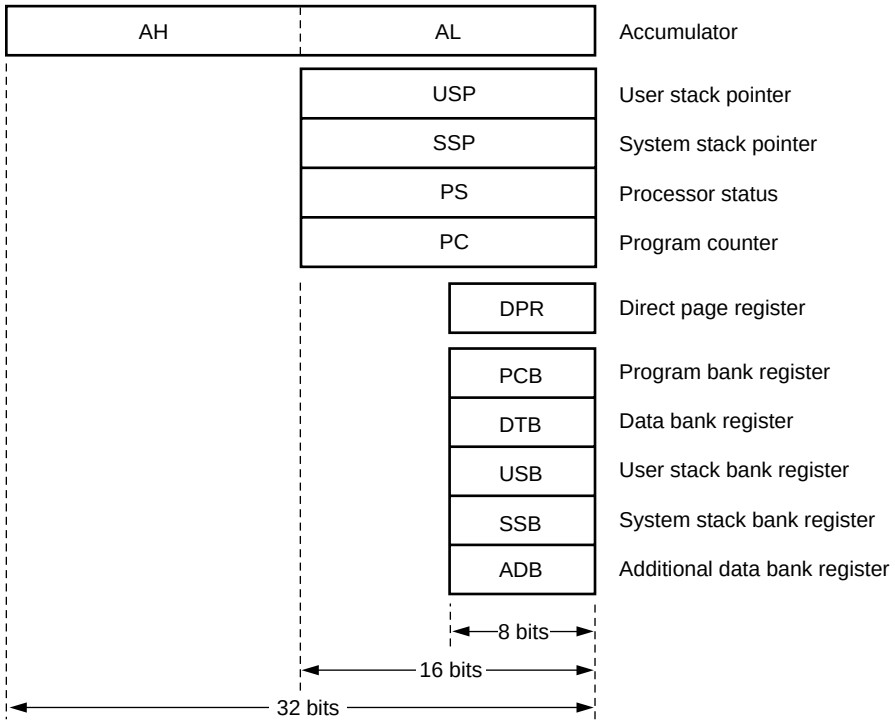
■ BLOCK DIAGRAM



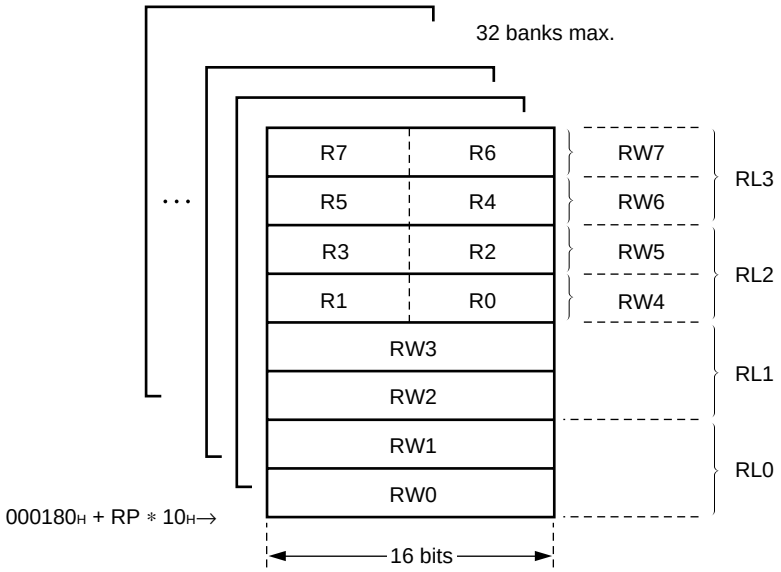
Note: In the diagram above, I/O ports share pins with all internal function blocks. These cannot be used as I/O ports when used as internal module pins.

■ F²MC-16L CPU PROGRAMMING MODEL

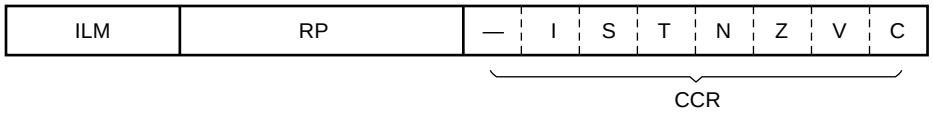
- **Dedicated Registers**



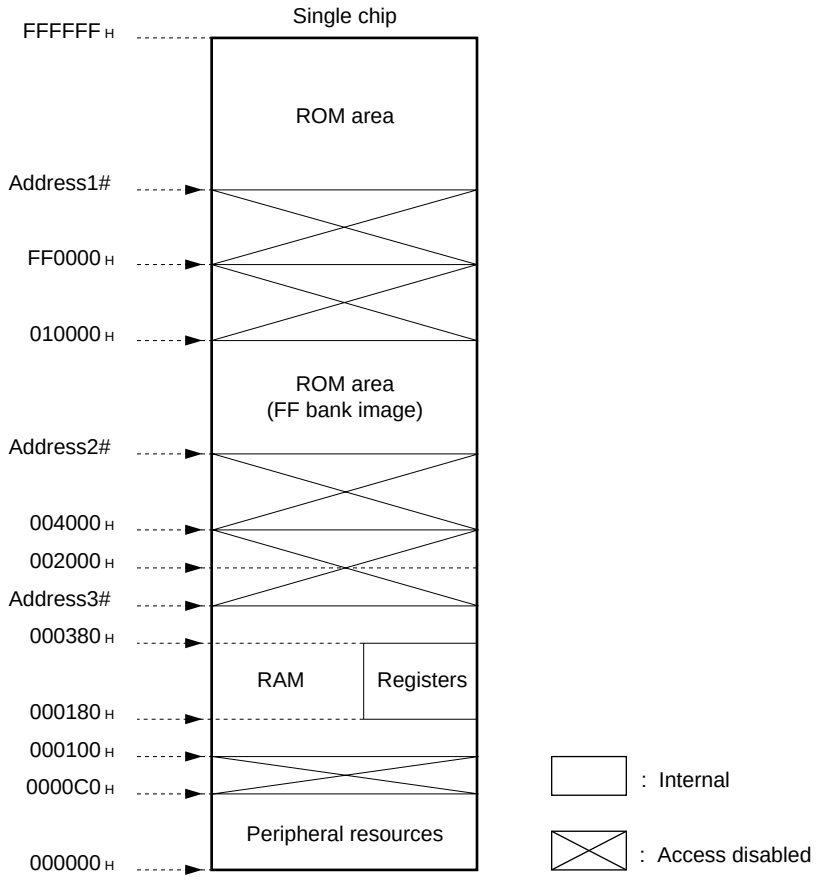
- **General-purpose Registers**



- **Processor States (PS)**



■ MEMORY MAP



Product Model	Address #1	Address #2	Address #3
MB90662A	FF8000H	008000H	000780H
MB90663A	FF4000H	004000H	000900H
MB90P663A	FF4000H	004000H	000900H

■ I/O MAP

Address	Register	Name	Access*2	Resource name	Initial value
000000H	Port 0 data register	PDR0	R/W*	Port 0	X X X X X X X X
000001H	Port 1 data register	PDR1	R/W*	Port 1	X X X X X X X X
000002H	Port 2 data register	PDR2	R/W*	Port 2	X X X X X X X X
000003H	Port 3 data register	PDR3	R/W*	Port 3	– – – – X X X X
000004H	Port 4 data register	PDR4	R/W!	Port 4	X X X X X X X X
000005H	Port 5 data register	PDR5	R/W*	Port 5	1 1 1 1 1 1 1 1
000006H	Port 6 data register/ Port data buffer register	PDR6/ PDBR	R/W*	Port 6	– X X X X X X X
000007H to 0FH	Vacancy	—	*1	—	—
000010H	Port 0 direction register	DDR0	R/W	Port 0	0 0 0 0 0 0 0 0
000011H	Port 1 direction register	DDR1	R/W	Port 1	0 0 0 0 0 0 0 0
000012H	Port 2 direction register	DDR2	R/W	Port 2	0 0 0 0 0 0 0 0
000013H	Port 3 direction register	DDR3	R/W	Port 3	– – – – 0 0 0 0
000014H	Port 4 direction register	DDR4	R/W	Port 4	– – – – 0 0 0 0
000015H	Analog input enable register	ADER	R/W	Port 5	1 1 1 1 1 1 1 1
000016H	Port 6 direction register	DDR6	R/W	Port 6	– 0 0 0 0 0 0 0
000017H to 1BH	Vacancy	—	*1	—	—
00001CH to 1FH	System reserved area	—	*1	—	—
000020H	PWM operation mode control register	PWMC	R/W	PWM	0 0 0 0 0 – – 1
000021H	Vacancy	—	*1		—
000022H	PWM reload register	PRLL	R/W		X X X X X X X X
000023H		PRLH	R/W		X X X X X X X X
000024H	Serial mode register	SMR	R/W!	UART	0 0 0 0 0 – 0 0
000025H	Serial control register	SCR	R/W!		0 0 0 0 0 1 0 0
000026H	Serial input data register/ Serial output data register	SIDR/ SODR	R/W		X X X X X X X X
000027H	Serial status register	SSR	R/W!		0 0 0 0 1 – 0 0
000028H	Interrupt enable register	ENIR	R/W	External interrupt	0 0 0 0 0 0 0 0
000029H	Interrupt source register	EIRR	R/W	External interrupt	X X X X X X X X
00002AH	Request level setting register	ELVR	R/W		0 0 0 0 0 0 0 0
00002BH					0 0 0 0 0 0 0 0
00002CH	A/D control status register	ADCS	R/W!	A/D converter	0 0 0 0 0 0 0 0
00002DH					0 0 0 0 0 0 0 0

(Continued)

(Continued)

Address	Register	Name	Access*2	Resource name	Initial value
00002E$_H$	A/D data register	ADCR	R/W!	A/D converter	X X X X X X X X
00002F$_H$					0 0 0 0 0 0 X X
000030$_H$	Control status register	TMCSR0	R/W	16-bit reload timer 0	0 0 0 0 0 0 0 0
000031$_H$					– – – – 0 0 0 0
000032$_H$	16-bit timer register/ 16-bit reload register	TMR0/ TMRLR0	R/W		X X X X X X X X
000033$_H$					X X X X X X X X
000034$_H$	Control status register	TMCSR1	R/W	16-bit reload timer 1	0 0 0 0 0 0 0 0
000035$_H$					– – – – 0 0 0 0
000036$_H$	16-bit timer register/ 16-bit reload register	TMR1/ TMRLR1	R/W		X X X X X X X X
000037$_H$					X X X X X X X X
000038$_H$	Control status register	TMCSR2	R/W	16-bit reload timer 2	0 0 0 0 0 0 0 0
000039$_H$					– – – – 0 0 0 0
00003A$_H$	16-bit timer register/ 16-bit reload register	TMR2/ TMRLR2	R/W		X X X X X X X X
00003B$_H$					X X X X X X X X
00003C$_H$	Control status register	TMCSR3	R/W	16-bit reload timer 3	0 0 0 0 0 0 0 0
00003D$_H$					– – – – 0 0 0 0
00003E$_H$	16-bit timer register/ 16-bit reload register	TMR3/ TMRLR3	R/W		X X X X X X X X
00003F$_H$					X X X X X X X X
000040$_H$	Timer control status register	TCSR	R/W!	Multi-function timer	1 0 0 0 0 0 0 0
000041$_H$	Compare interrupt control register	CICR	R/W		0 0 0 0 0 0 0 0
000042$_H$	Timer mode control register	TMCR	R/W!		0 0 1 – 0 0 0 0
000043$_H$	Compare/data select register	COER	R/W		– – – – 0 0 0 0
000044$_H$	Compare buffer mode control register	CMCR	R/W		– – – – 0 0 0 0
000045$_H$	Zero detect output control register	ZOCTR	W		– – – X 0 0 0 0
000046$_H$	Output control buffer register	OCTBR	R/W		1 1 1 1 1 1 1 1
000047$_H$	Zero detect interrupt control register	ZICR	R/W!		0 – – – X X X X
000048$_H$	Output compare buffer register 0	OCPBR0	W		X X X X X X X X
000049$_H$					– – X X X X X X
00004A$_H$	Output compare buffer register 1	OCPBR1	W		X X X X X X X X
00004B$_H$					– – X X X X X X
00004C$_H$	Output compare buffer register 2	OCPBR2	W		X X X X X X X X
00004D$_H$					– – X X X X X X

(Continued)

(Continued)

Address	Register	Name	Access*2	Resource name	Initial value
00004E$_H$	Output compare buffer register 3	OCPBR3	W	Multi-function timer	X X X X X X X X
00004F$_H$					– – X X X X X X
000050$_H$	Compare clear buffer register	CLRBR	W		0 0 0 0 0 0 0 0
000051$_H$					– – 0 0 0 0 0 0
000052$_H$	Dead time control register	DTCR	R/W!		0 0 0 0 0 0 0 0
000053$_H$	Dead time setting register	DTSR	W		X X X 0 X X X X
000054$_H$	Dead time compare register	DTCMR	W		X X X X X X X X
000055$_H$	Vacancy	—	*1	—	—
000056$_H$	Timer pin control register	TPCR	R/W	16-bit reload timer	– 0 0 1 – 0 0 0
000057$_H$					– 0 1 1 – 0 1 0
000058$_H$ to 5E$_H$	Vacancy	—	*1	—	—
00005F$_H$	Machine clock division control register	CDCR	W	UART	– – – – 1 1 1 1
000060$_H$ to 8F$_H$	Vacancy	—	*1	—	—
000090$_H$ to 9E$_H$	System reserved area	—	*1	—	—
00009F$_H$	Delayed interrupt source generate/ cancel register	DIRR	R/W	Delayed interrupt generator module	– – – – – – – 0
0000A0$_H$	Low power mode control register	LPMCR	R/W!	Low power	0 0 0 1 1 0 0 0
0000A1$_H$	Clock select register	CKSCR	R/W!		1 1 1 1 1 1 0 0
0000A2$_H$ to A7$_H$	System reserved area	—	*1	—	—
0000A8$_H$	Watchdog timer control register	WDTC	R/W!	Watchdog timer	X – X X X 1 1 1
0000A9$_H$	Timebase timer control register	TBTC	R/W!	Timebased timer	1 – – 0 0 1 0 0
0000AA$_H$ to AF$_H$	System reserved area	—	*1	—	—
0000B0$_H$	Interrupt control register 00	ICR00	R/W!	Interrupt controller	0 0 0 0 0 1 1 1
0000B1$_H$	Interrupt control register 01	ICR01	R/W!		0 0 0 0 0 1 1 1
0000B2$_H$	Interrupt control register 02	ICR02	R/W!		0 0 0 0 0 1 1 1
0000B3$_H$	Interrupt control register 03	ICR03	R/W!		0 0 0 0 0 1 1 1
0000B4$_H$	Interrupt control register 04	ICR04	R/W!		0 0 0 0 0 1 1 1
0000B5$_H$	Interrupt control register 05	ICR05	R/W!		0 0 0 0 0 1 1 1
0000B6$_H$	Interrupt control register 06	ICR06	R/W!		0 0 0 0 0 1 1 1
0000B7$_H$	Interrupt control register 07	ICR07	R/W!		0 0 0 0 0 1 1 1

(Continued)

(Continued)

Address	Register	Name	Access*2	Resource name	Initial value
0000B8H	Interrupt control register 08	ICR08	R/W!	Interrupt controller	0 0 0 0 0 1 1 1
0000B9H	Interrupt control register 09	ICR09	R/W!		0 0 0 0 0 1 1 1
0000BAH	Interrupt control register 10	ICR10	R/W!		0 0 0 0 0 1 1 1
0000BBH	Interrupt control register 11	ICR11	R/W!		0 0 0 0 0 1 1 1
0000BCH	Interrupt control register 12	ICR12	R/W!		0 0 0 0 0 1 1 1
0000BDH	Interrupt control register 13	ICR13	R/W!		0 0 0 0 0 1 1 1
0000BEH	Interrupt control register 14	ICR14	R/W!		0 0 0 0 0 1 1 1
0000BFH	Interrupt control register 15	ICR15	R/W!		0 0 0 0 0 1 1 1
0000C0H to FFH	System reserved area	—	*1	—	—

*1: Access prohibited

*2: Registers marked “R/W!” in the access column include some bits that can only be read or only be written. For details, see the register list for each resource.

* : When a register marked “R/W!”, “R/W*” or “W” in the access column is accessed by a read-modify-write instruction (such as a bit set instruction), the bit operated on by the instruction will be set to the specified value, but a malfunction will occur if there are any other bits which can only be written. Therefore, do not access these locations using these instructions.

Description of Initial Values

0: The initial value of this bit is “0”.

1: The initial value of this bit is “1”.

*: The initial value of this bit is “1” or “0”. (This is determined depending on the level of the MD0 to MD2 pins.)

X: The initial value of this bit is undefined.

–: This bit is not used. The initial value is undefined.

Note: The initial value results for bits which can only be written when initialized by a reset. Note that this is not the value when read.
Also, sometimes LPMCR, CKSCR and WDTC are initialized and sometimes they are not depending on the type of reset. If they are initialized, the initial value is used.

■ INTERRUPT SOURCES, INTERRUPT VECTORS AND INTERRUPT CONTROL REGISTERS

Interrupt source	I²OS support	Interrupt vector			Interrupt control register	
		Number		Address	ICR	Address
Reset	×	#08	08_H	$FFFFDC_H$	—	—
INT9 instruction	×	#09	09_H	$FFFFD8_H$	—	—
Exception	×	#10	$0A_H$	$FFFFD4_H$	—	—
Multi-function timer DTTI input	×	#12	$0C_H$	$FFFFCC_H$	ICR00	$0000B0_H$
External interrupt #0	○	#13	$0D_H$	$FFFFC8_H$	ICR01	$0000B1_H$
External interrupt #4	○	#14	$0E_H$	$FFFFC4_H$		
Multi-function timer trigger input or zero detect	○	#15	$0F_H$	$FFFFC0_H$	ICR02	$0000B2_H$
Multi-function timer zero detect	○	#17	11_H	$FFFFB8_H$	ICR03	$0000B3_H$
Multi-function timer overflow, compare clear or zero detect	○	#19	13_H	$FFFFB0_H$	ICR04	$0000B4_H$
External interrupt #1	○	#21	15_H	$FFFFA8_H$	ICR05	$0000B5_H$
Multi-function timer compare match	×	#22	16_H	$FFFFA4_H$		
External interrupt #5	○	#23	17_H	$FFFFA0_H$	ICR06	$0000B6_H$
PWM underflow	×	#24	18_H	$FFFF9C_H$		
External interrupt #2	○	#25	19_H	$FFFF98_H$	ICR07	$0000B7_H$
External interrupt #6	○	#26	$1A_H$	$FFFF94_H$		
16-bit reload timer #0	○	#27	$1B_H$	$FFFF90_H$	ICR08	$0000B8_H$
16-bit reload timer #1	○	#28	$1C_H$	$FFFF8C_H$		
16-bit reload timer #2	○	#29	$1D_H$	$FFFF88_H$	ICR09	$0000B9_H$
16-bit reload timer #3	○	#30	$1E_H$	$FFFF84_H$		
End of A/D converter conversion	○	#31	$1F_H$	$FFFF80_H$	ICR10	$0000BA_H$
Timebase timer interval interrupt	×	#34	22_H	$FFFF74_H$	ICR11	$0000BB_H$
UART send complete	○	#35	23_H	$FFFF70_H$	ICR12	$0000BC_H$
UART receive complete	◎	#37	25_H	$FFFF68_H$	ICR13	$0000BD_H$
External interrupt #3	○	#39	27_H	$FFFF60_H$	ICR14	$0000BE_H$
External interrupt #7	○	#40	28_H	$FFFF5C_H$		
Delayed interrupt generator module	×	#42	$2A_H$	$FFFF54_H$	ICR15	$0000BF_H$

○: indicates that the interrupt request flag is cleared by the I²OS interrupt clear signal (no stop request).
◎: indicates that the interrupt request flag is cleared by the I²OS interrupt clear signal (with stop request).
×: indicates that the interrupt request flag is not cleared by the I²OS interrupt clear signal.

Note: Do not specify I²OS activation in interrupt control registers that do not support I²OS.

■ PERIPHERAL RESOURCES

1. Parallel Port

The MB90660A includes 39 I/O pins, 4 input pins, and 8 open-drain output pins.

Port 0, 1, 2, 3 and 6 are I/O ports. They are used for input when the corresponding direction register value is “0”, and for output when the value is “1”.

Port 5 is an open-drain port. It is used as a port when the analog input enable register is “0”.

Ports 40 to 43 are I/O ports. They are used for input when the corresponding direction register value is “0”, and for output when the value is “1”. Ports 44 to 47 are input ports which can only be used for reading data.

(1) Register Configuration

Port Data Register
Address: PDR1 000001H
: PDR3 000003H

bit	15	14	13	12	11	10	9	8	
	PD×7	PD×6	PD×5	PD×4	PD×3	PD×2	PD×1	PD×0	PDR1, 3
Read/Write →	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value →	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

Port Data Register
Address: PDR0 000000H
: PDR2 000002H
: PDR6 000006H
(PDBR)

bit	7	6	5	4	3	2	1	0	
	PD×7	PD×6	PD×5	PD×4	PD×3	PD×2	PD×1	PD×0	PDR0, 2, 6
Read/Write →	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value →	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

Port Data Register
Address: 000005H

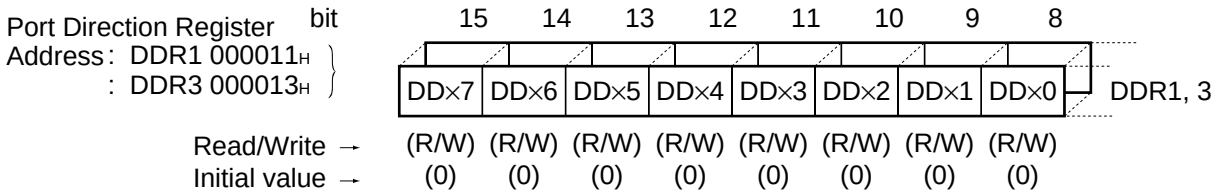
bit	15	14	13	12	11	10	9	8	
	PD57	PD56	PD55	PD54	PD53	PD52	PD51	PD50	PDR5
Read/Write →	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value →	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

Port Data Register
Address: 000004H

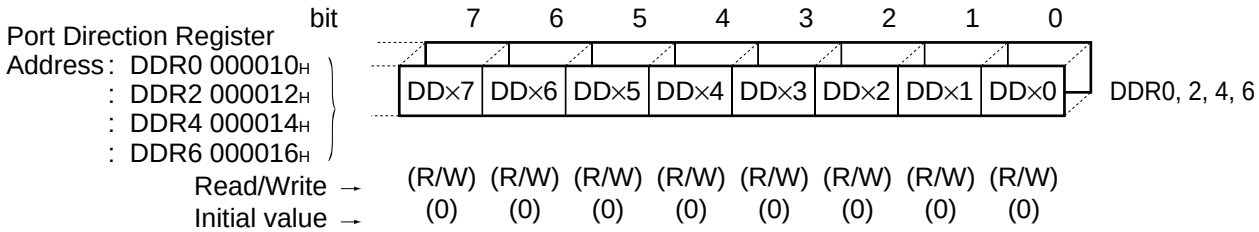
bit	7	6	5	4	3	2	1	0	
	PD47	PD46	PD45	PD44	PD43	PD42	PD41	PD40	PDR4
Read/Write →	(R)	(R)	(R)	(R)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value →	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

Notes: There are no register bits for bits 15 to 12 of Port 3.
There is no register bit for bit 7 of Port 6.
Bits 7 to 4 of Port 4 can only be used to read data.

Port Direction Register
Address : DDR1 000011H
: DDR3 000013H

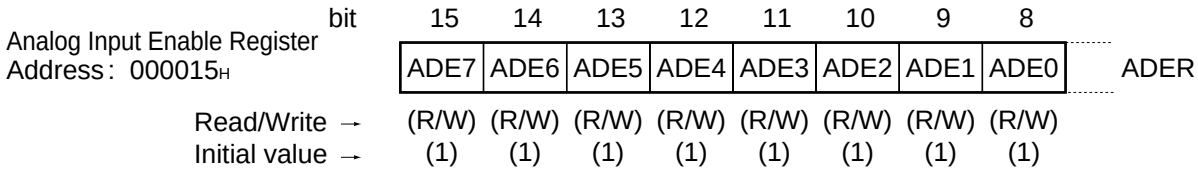


Port Direction Register
Address : DDR0 000010H
: DDR2 000012H
: DDR4 000014H
: DDR6 000016H



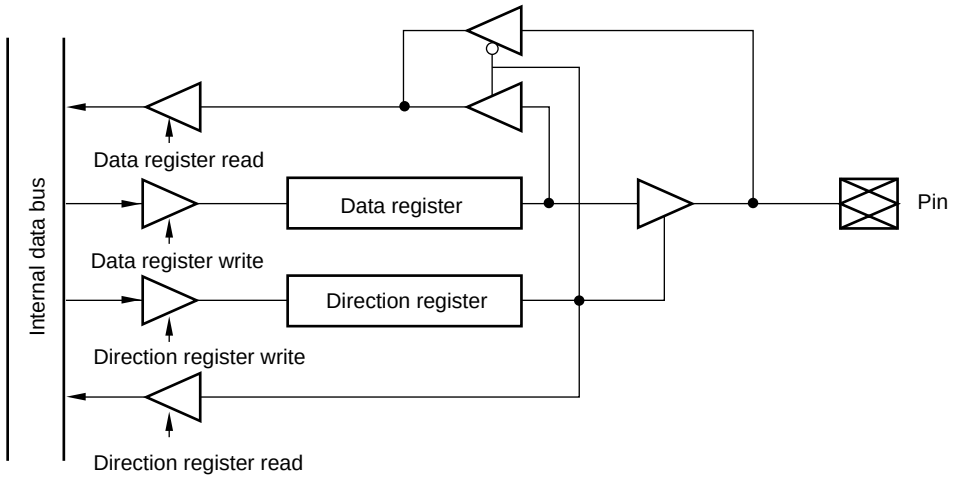
Notes: There are no register bits for bits 15 to 12 of Port 3.
There are not register bits for bits 7 to 4 of Port 4
There is no DDR for Port 5.
There is no register bit for bit 7 of Port 6.

Analog Input Enable Register
Address : 000015H

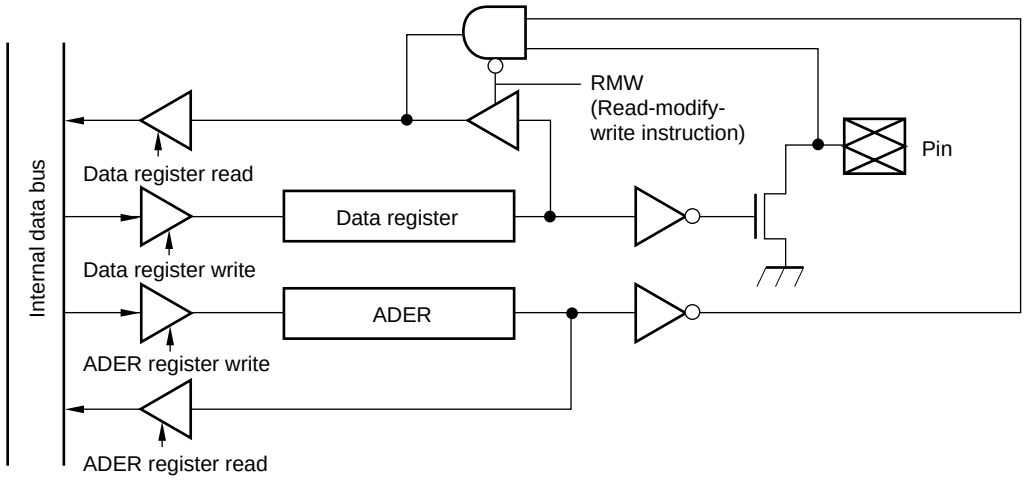


(2) Block Diagrams

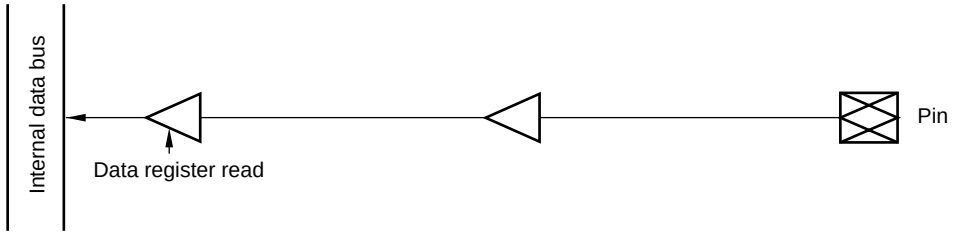
- **I/O Ports**



- **Open-drain Ports (Also Used for Analog Input)**



- **Input Ports**



2. Multi-function Timer

The multi-function timer controls up to 7 realtime output pins, and includes the following functions.

- Interval timer function
  It can output pulses or generate an interrupt at a fixed interval.
- PWM output function
  Can perform output for a fixed cycle pulse while changing the duty ratio (ratio between “L” output width and “H” output width) in realtime.
- 3-phase AC sine wave output (inverter control output) function
  Can perform 3-phase AC sine wave output using AC motor inverter control, etc. (using any setting for the non-overlap interval)

This timer also has the following characteristics.

- Pulse cycle control using 14-bit timer
  A machine cycle of 1, 2, 8 or 16 can be selected based on pre-scalars as the clock source (Minimum resolution of 62.5 ns at 16 MHz operation).

  Can use a carrier frequency up to 30 KHz at 8-bit stop when used for AC motor control.

  Up count only or up/down count can be selected using the count mode selection.

  Possessing a buffer, cycle can be changed in realtime by transferring data from buffer upon zero detect.
- Duty control using compare registers
  Possessing four compare registers, output pulse duty can be set for four separate channels.

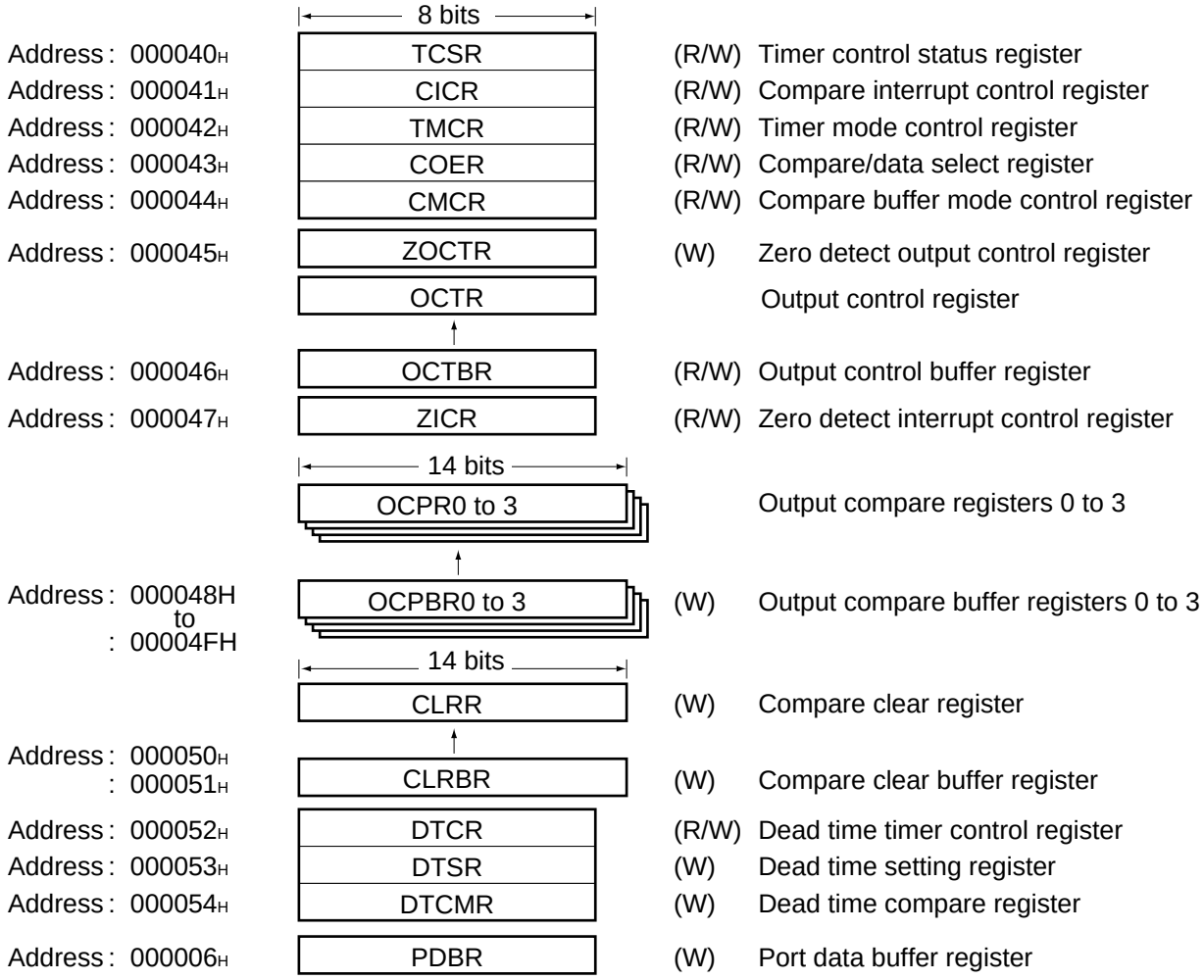
  Each possessing a separate buffer, duty can be changed in realtime by transferring data from buffer upon zero detect or comparison.
- Non-overlap control using dead time timer
  Dead time timer can be used to generate PWM output for three channels or even reversed signals with non-overlap, thus allowing an AC motor control wave (U, V, W, X, Y, Z) to be generated.

  A machine cycle of 1, 4, 8 or 32 can be selected based on pre-scalars as the clock source for the dead timer (Minimum resolution of 62.5 ns at 16 MHz operation)
- Forced stop control using DTTI pin input
  The forced pin output level can be fixed by DTTI pin input or software.

  Inactive control can be performed during AC motor control using DTTI pin input.

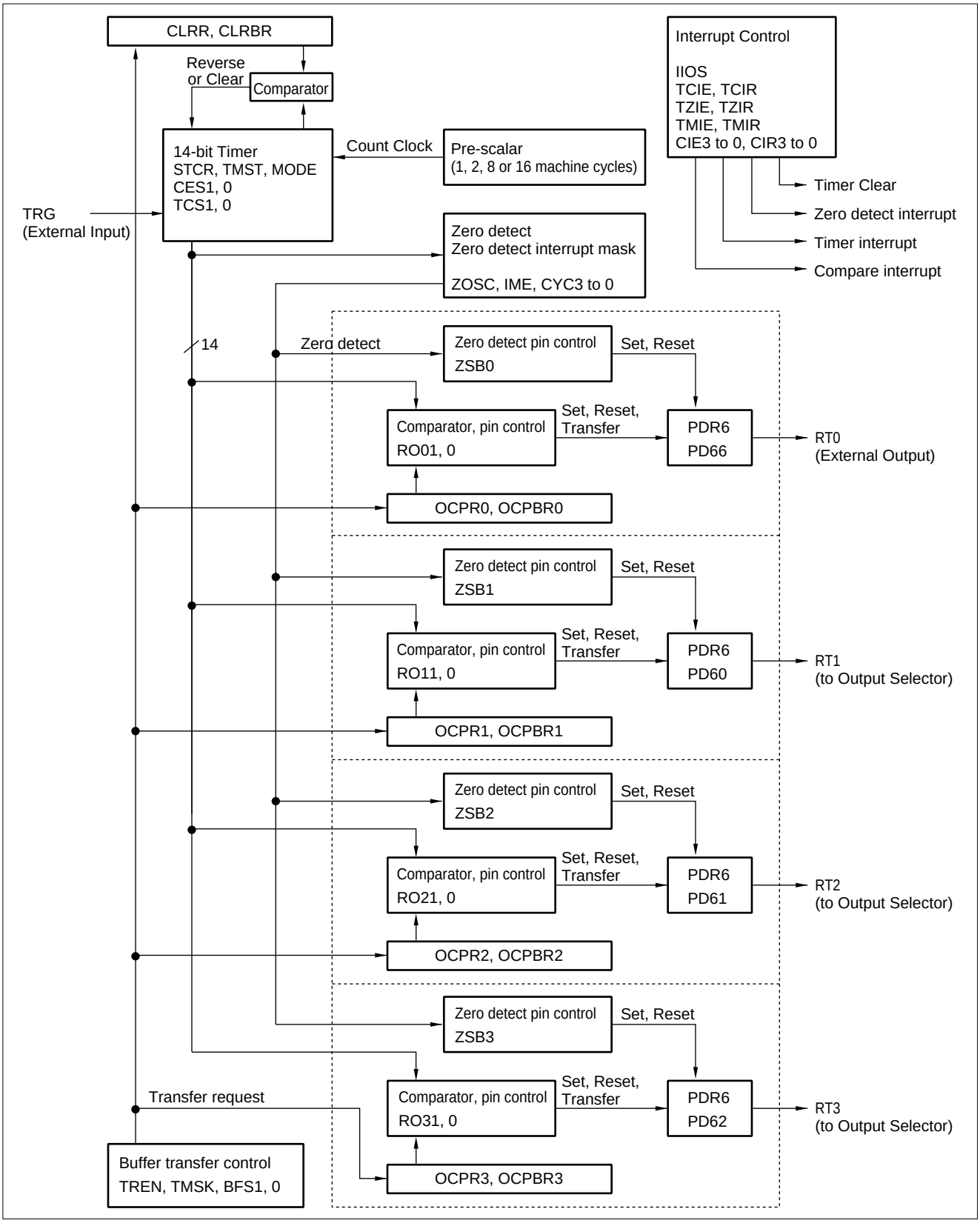
  External pin control even during vibration stop can be performed through clockless DTTI pin input.
- Event detection and interrupt generation using various flags
  Flags can be set and/or interrupts generated upon zero detect, overflow, detect of match with compare registers, or clear by TRG pin input, or any match of the compare registers for the four channels for the 14-bit timer (also possible to disable interrupt output).

(1) Register Configuration

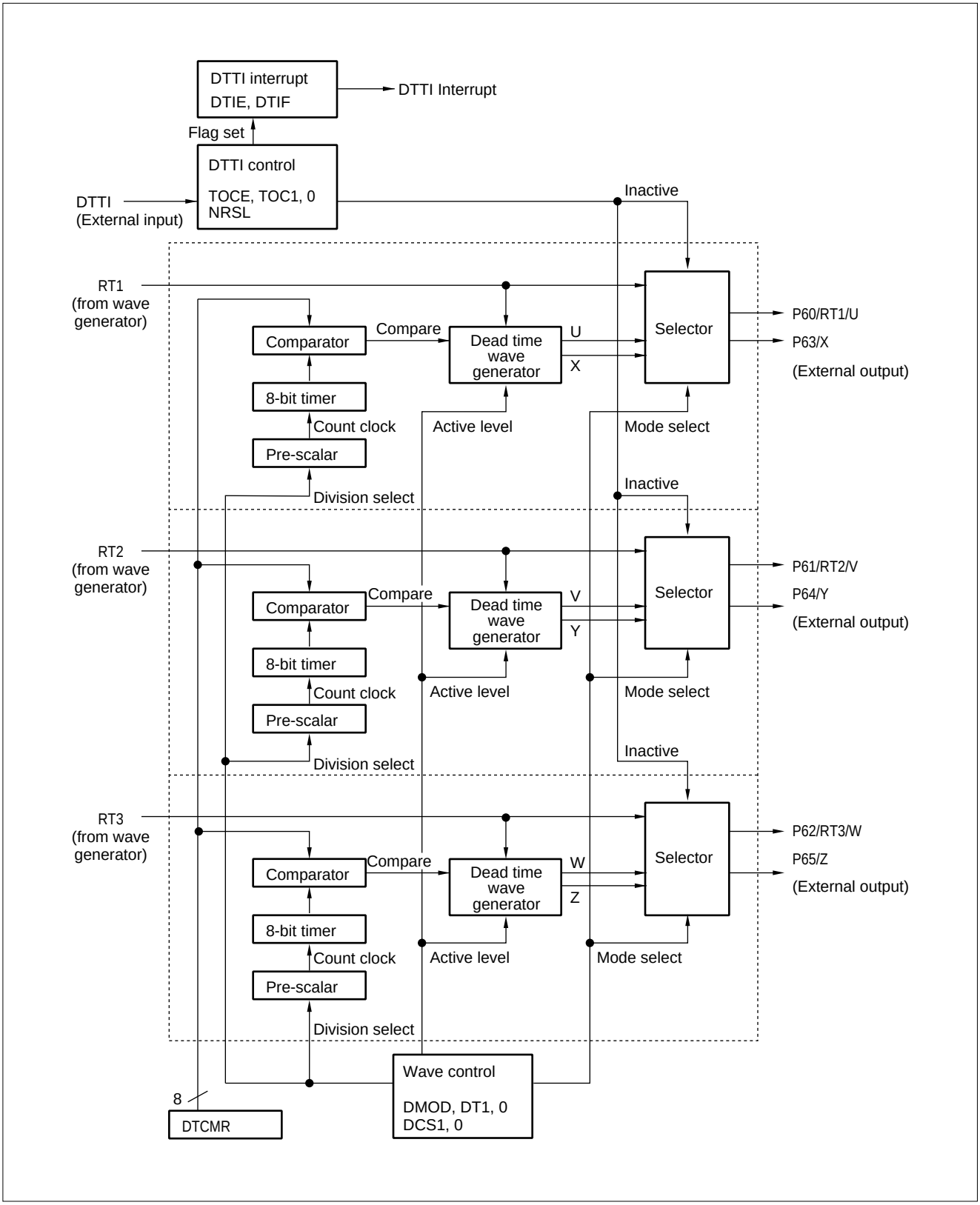


(2) Block Diagrams

- **Timer/wave generator block diagram**



- **Output selector/dead time generator block diagram**

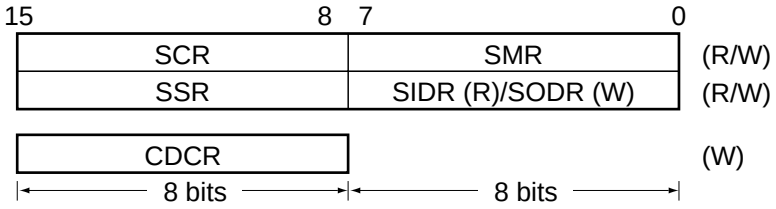


3. UART

The UART is a serial I/O port for asynchronous (start/stop) or CLK synchronous communications with external resources. It has the following characteristics:

- Full duplex double buffering
- Asynchronous (start/stop) or CLK synchronous communications
- Multiprocessor mode support
- Internal dedicated baud-rate generator
  Asynchronous : 19230/9615/31250/4808/2404/1202 bps
  CLK synchronous : 2 M/1 M/500 K/250 K bps
- Free baud-rate setting based on external clock
- Error detection functions (parity, framing and overrun)
- Use of NRZ coded transfer signal
- Supports intelligent I/O services

(1) Register Configuration



	bit	7	6	5	4	3	2	1	0	
Address : 000024H		MD1	MD0	CS2	CS1	CS0	–	SCKE	SOE	Serial mode register (SMR)

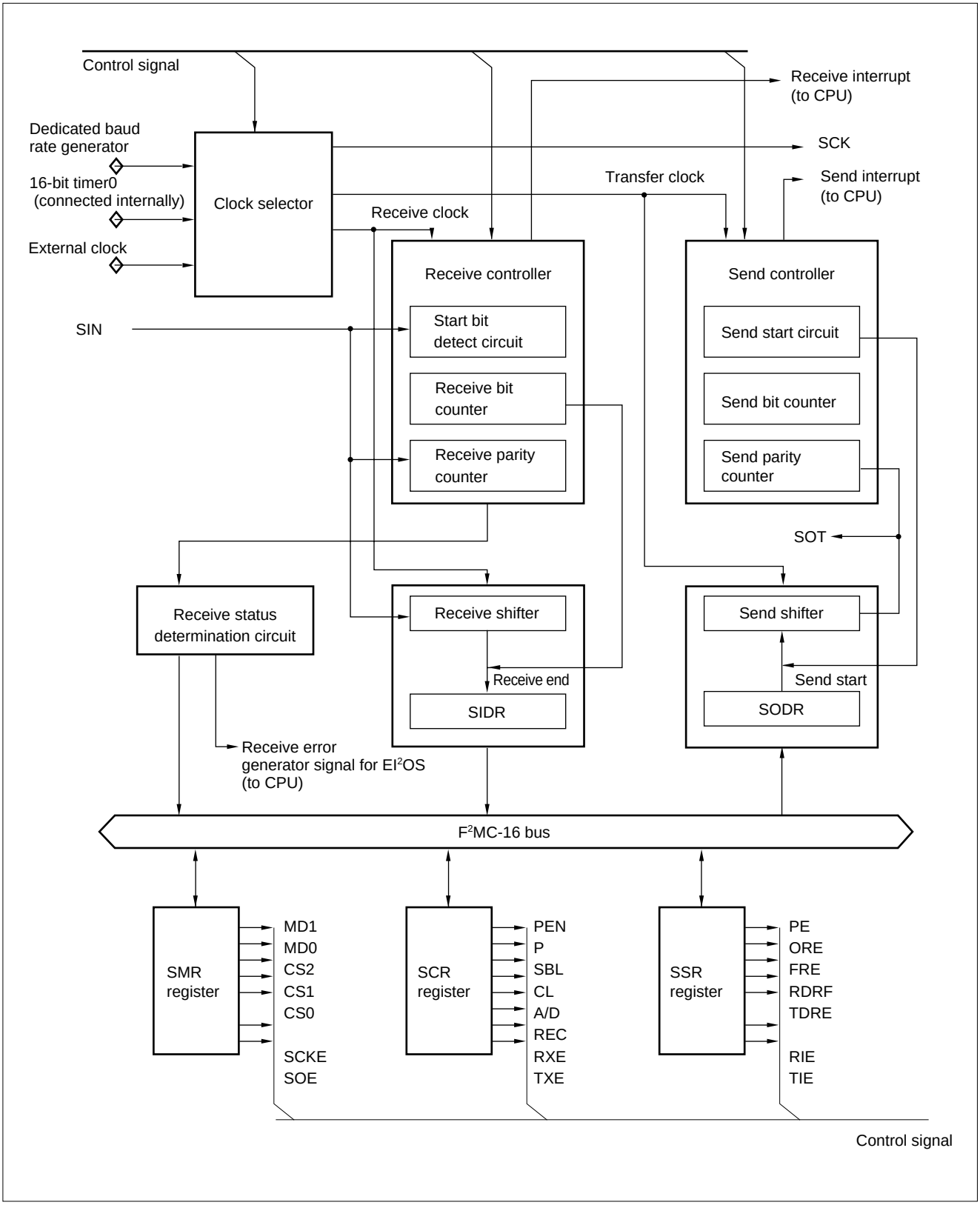
	bit	15	14	13	12	11	10	9	8	
Address : 000025H		PEN	P	SBL	CL	A/D	REC	RXE	TXE	Serial control register (SCR)

	bit	7	6	5	4	3	2	1	0	
Address : 000026H		D7	D6	D5	D4	D3	D2	D1	D0	Serial input register Serial output register (SIDR/SODR)

	bit	15	14	13	12	11	10	9	8	
Address : 000027H		PE	ORE	FRE	RDRF	TDRE	–	RIE	TIE	Serial status register (SSR)

	bit	15	14	13	12	11	10	9	8	
Address : 00005FH		–	–	–	–	DIV3	DIV2	DIV1	DIV0	Machine clock division control register (CDCR)

(2) Block Diagram



4. 10-bit, 8-channel A/D Converter (with 8-bit Resolution Mode)

This 10-bit, 8-channel A/D converter is used to convert analog input voltage to corresponding digital values. It has the following features.

- Conversion time: 6.13 µs per channel (includes sample and hold time at 98 machine cycles/machine clock of 16 MHz)
- Sample hold time: 3.75 µs per channel (60 machine cycles per machine clock of 16 MHz)
- RC-type sequential approximation conversion with sample and hold circuits
- 10-bit or 8-bit resolution
- Analog input can be selected from 8 channels

  Single conversion mode : One channel selected for conversion

  Scan conversion mode : Consecutive multiple channels converted (programmable with max. eight channels)

  Repetitive conversion mode : Data on the specified channel is converted repeatedly

  Stop conversion mode : Once one channel is converted, operations stop and the device waits until started again (conversion start can be synchronized)
- At the end of each A/D conversion, an interrupt request to the CPU can be generated. This interrupt can be used to activate I²OS or transfer A/D conversion results to memory, making it useful when continuous processing is desired.
- Conversion can be triggered by software, an external trigger (falling edge), and/or a timer (rising edge).

(1) Register Configuration

A/DControl status register (upper)
Address: 00002D$_H$

bit	15	14	13	12	11	10	9	8	
	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	Reserved	ADCS
Read/Write →	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(–)	
Initial value →	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

A/D Control status register (lower)
Address: 00002C$_H$

bit	7	6	5	4	3	2	1	0	
	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0	ADCS
Read/Write →	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value →	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

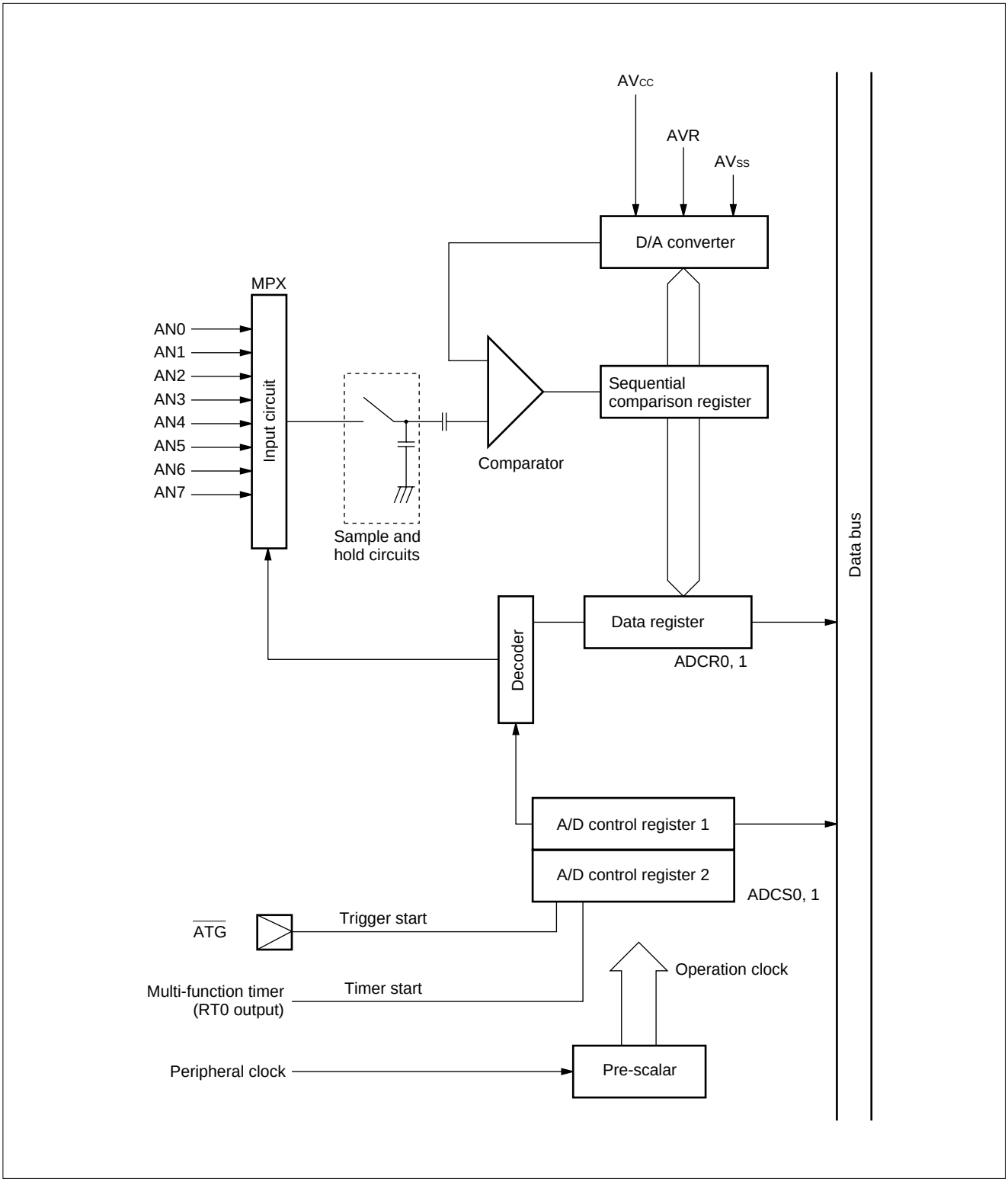
A/D Data register (upper)
Address: 00002F$_H$

bit	15	14	13	12	11	10	9	8	
	S10	–	–	–	–	–	D9	D8	ADCR
Read/Write →	(R/W)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Initial value →	(0)	(0)	(0)	(0)	(0)	(0)	(X)	(X)	

A/D Data register (lower)
Address: 00002E$_H$

bit	7	6	5	4	3	2	1	0	
	D7	D6	D5	D4	D3	D2	D1	D0	ADCR
Read/Write →	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Initial value →	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

(2) Block Diagram



5. PWM Timer

This block, which is an 8-bit reload timer module, outputs the pulse width modulation (PWM) using pulse output control corresponding to the timer operation.

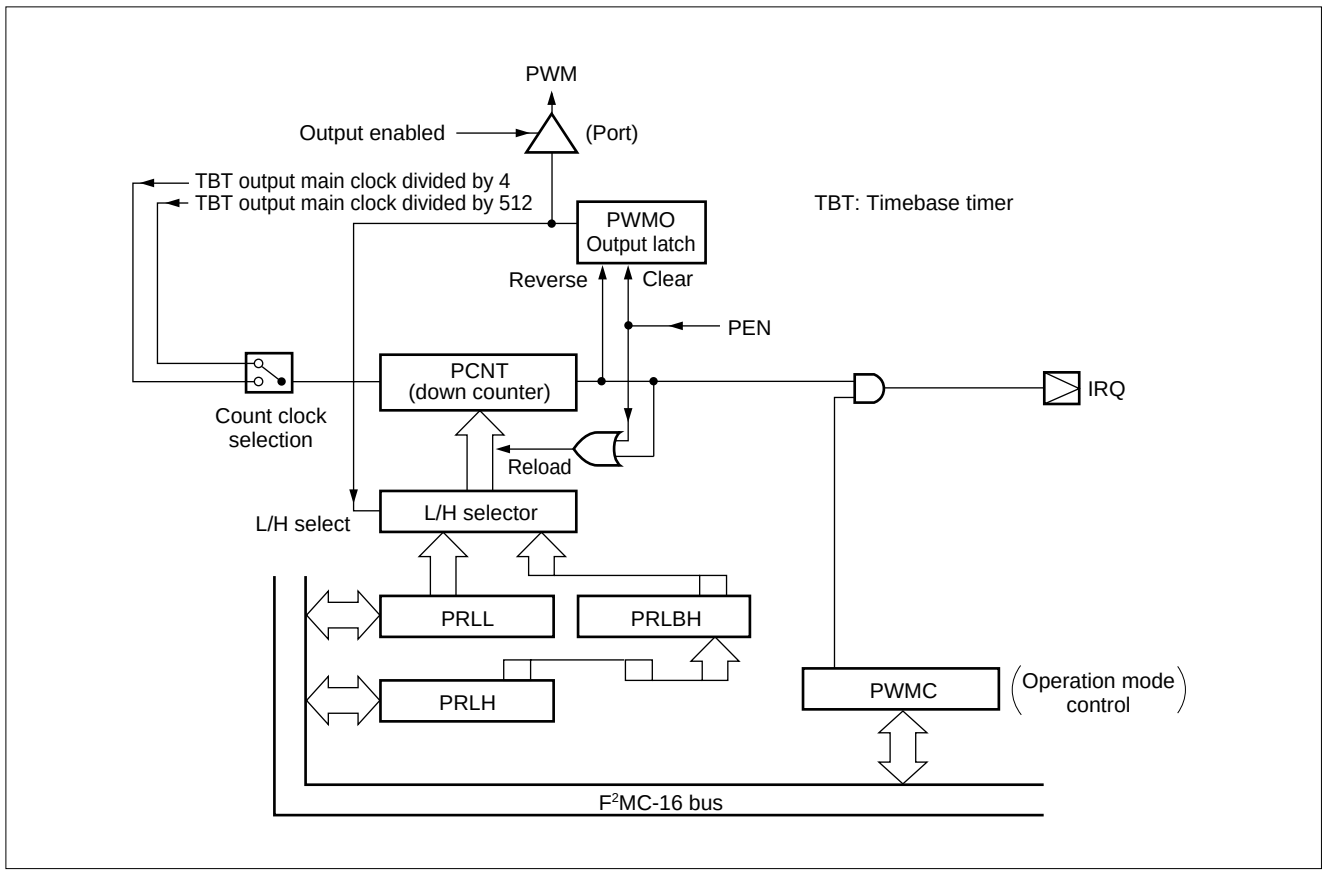
In terms of hardware, this block possesses an 8-bit down counter, two 8-bit reload registers for setting “L” width and “H” width, a control register, external pulse output pin, and interrupt output circuit to achieve the following functions.

- PWM output operation : Pulse waves of any period and duty factor are output.
  This block can also be used as a D/A converter with an external circuit.
  Interrupt requests can be output based on counter underflow.

(1) Register Configuration

	8 bits			(Functions)
PWM operation mode control register Address: 000020H		PWMC	(R/W)	Operation mode control
PWM reload register 000022H		PRLL	(R/W)	Hold “L” pulse width reload value
000023H		PRLH	(R/W)	Hold “H” pulse width reload value

(2) Block Diagram

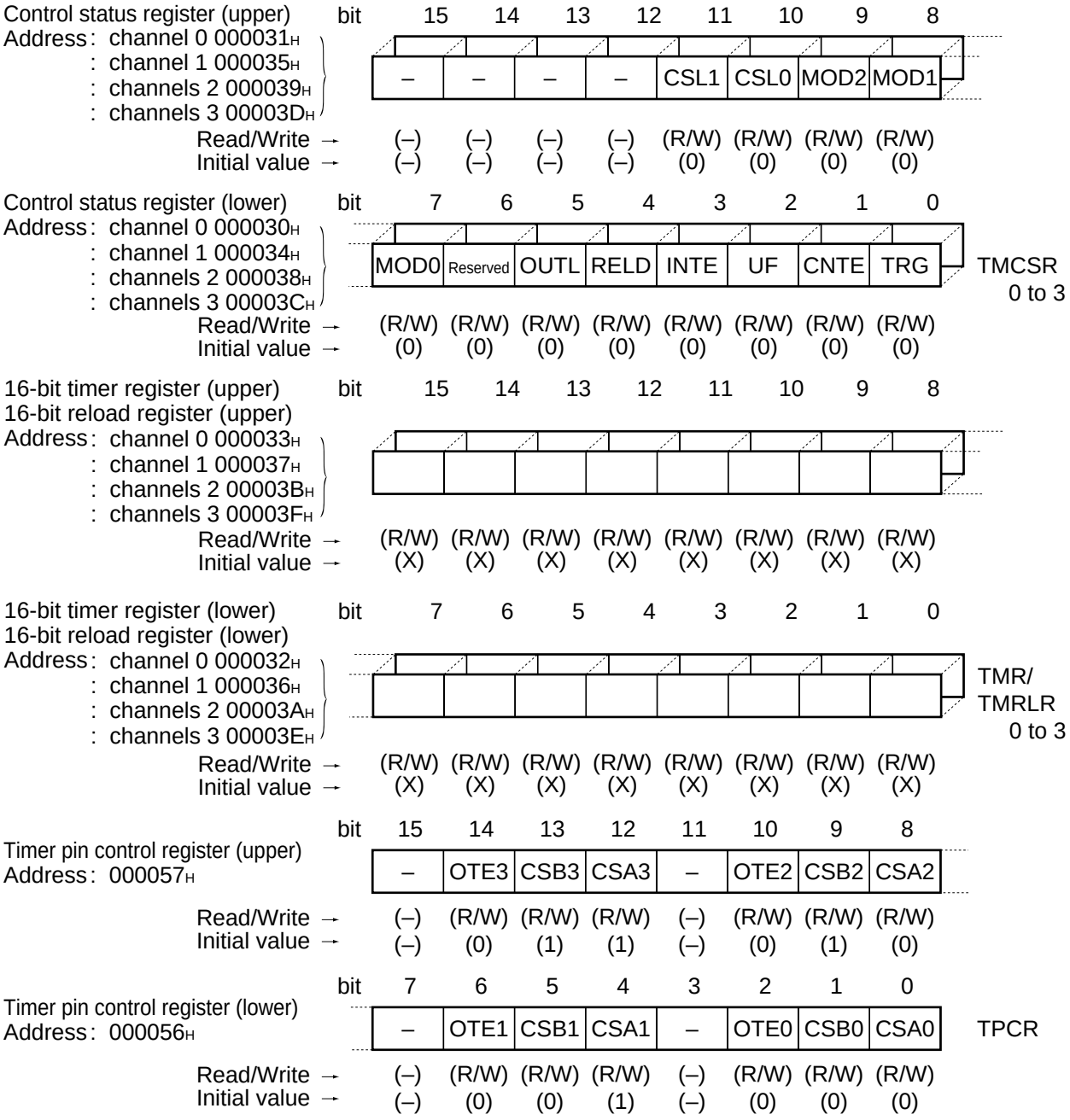


6. 16-bit Reload Timer (with Event Count Function)

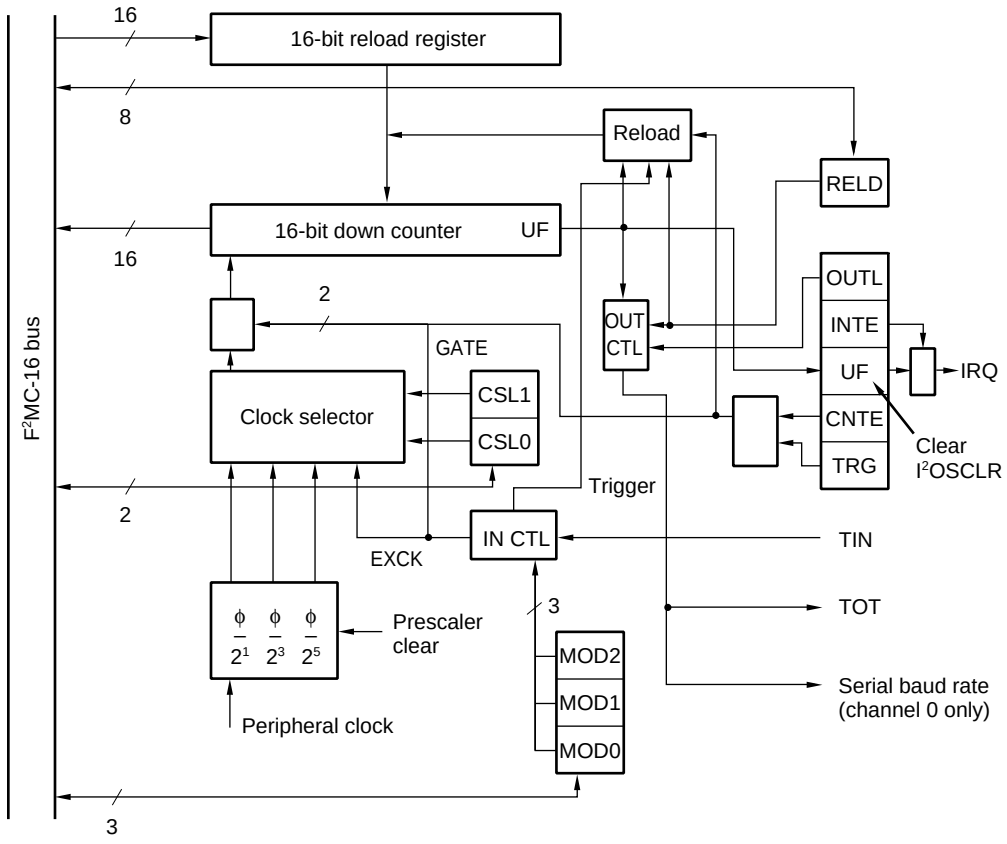
The 16-bit reload timer consists of a 16-bit down counter, a 16-bit reload register, control register, and 4 timer pins (I/O set by timer pin select register). Three internal clocks and an external clock can be selected as input clocks. A toggle output waveform is output at the output pin (TOT) in reload mode, while a square wave indicating that the timer is counting is output at the output pin in single-shot mode. The input pin (TIN) can be used for event input in even count mode, and for trigger input or gate input in internal clock mode.

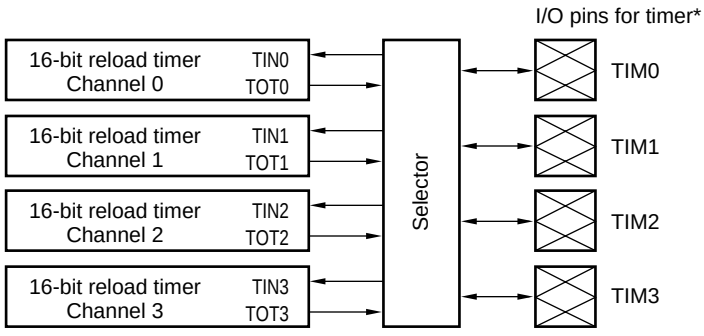
This product has this timer built into four channels.

(1) Register Configuration



(2) Block Diagram





* : Timer channel and direction (I/O) can be selected for each pin.

7. External Interrupts

In addition to “H” and “L”, rising and falling edge can be selected as the external interrupt level for a total of four interrupt level types.

(1) Register Configuration

Interrupt enable register
Address: 000028H

bit	7	6	5	4	3	2	1	0	
	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	ENIR
Read/Write →	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value →	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Interrupt source register
Address: 000029H

bit	15	14	13	12	11	10	9	8	
	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	EIRR
Read/Write →	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value →	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

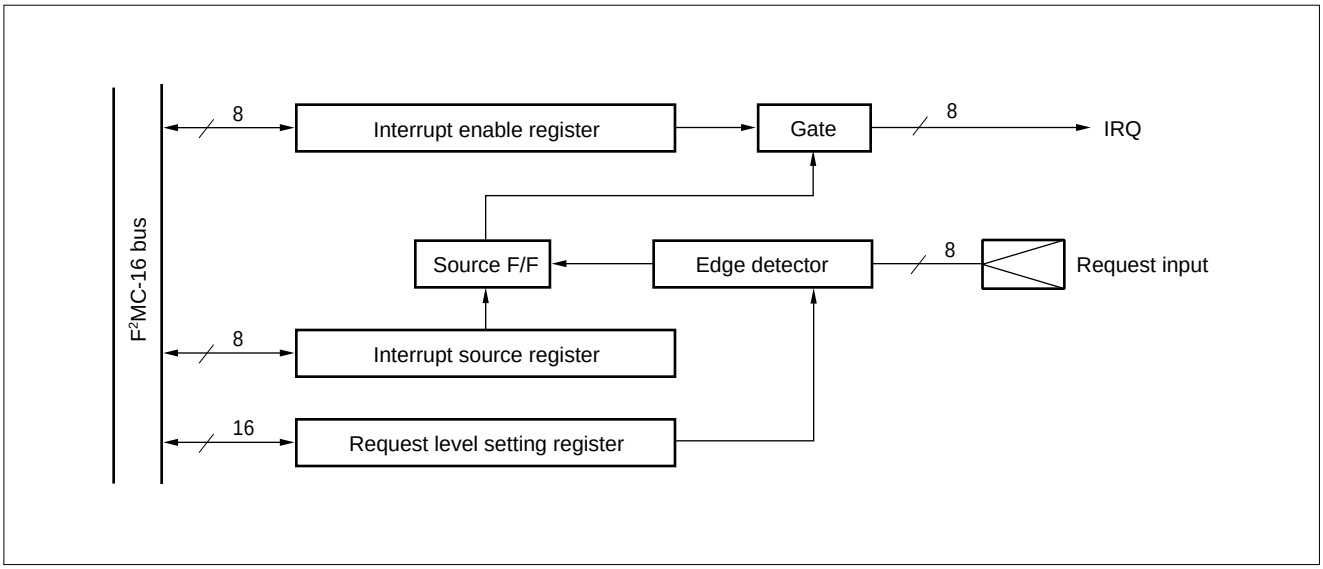
Request level setting register (upper)
Address: 00002BH

bit	15	14	13	12	11	10	9	8
	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4
Read/Write →	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value →	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Request level setting register (lower)
Address: 00002AH

bit	7	6	5	4	3	2	1	0	
	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	ELVR
Read/Write →	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value →	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

(2) Block Diagram



8. Delayed Interrupt Generation Module

The delayed interrupt generation module is used to generate an interrupt for task switching. If this module is used, an interrupt request to the F²MC-16L CPU can be generated or cancelled by software.

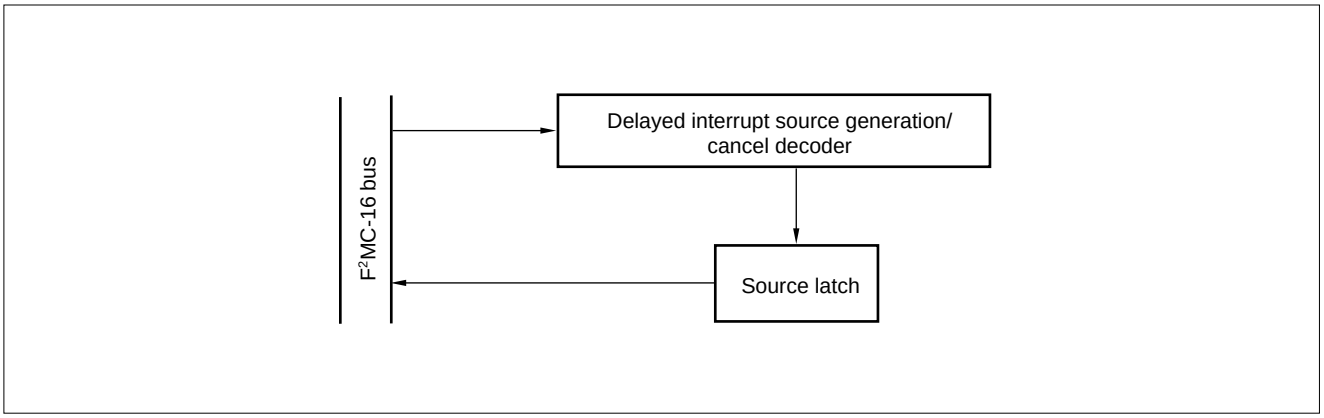
(1) Register Configuration

Delayed interrupt request generation/cancel register
Address : 000009H

bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	–	R0	DIRR
Read/Write →	(–)	(–)	(–)	(–)	(–)	(–)	(–)	(R/W)	
Initial value →	(–)	(–)	(–)	(–)	(–)	(–)	(–)	(0)	

The DIRR register controls the generation and cancellation of delayed interrupt requests. A delayed interrupt request is generated when “1” is written to this register, while a delayed interrupt request is cancelled when “0” is written here. Request cancel status results upon reset. Although either “0” or “1” may be written into reserved bits, we recommend using the set bit and clear bit instructions when accessing this register in consideration of possible future extensions.

(2) Block Diagram



9. Watchdog Timer and Timebase Timer Functions

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

In addition to an 18-bit timer, the timebase timer consists of a circuit for controlling interval interrupts. Note that the timebase timer uses the main clock regardless of the status of the MCS bit within the CKSCR register.

(1) Register Configuration

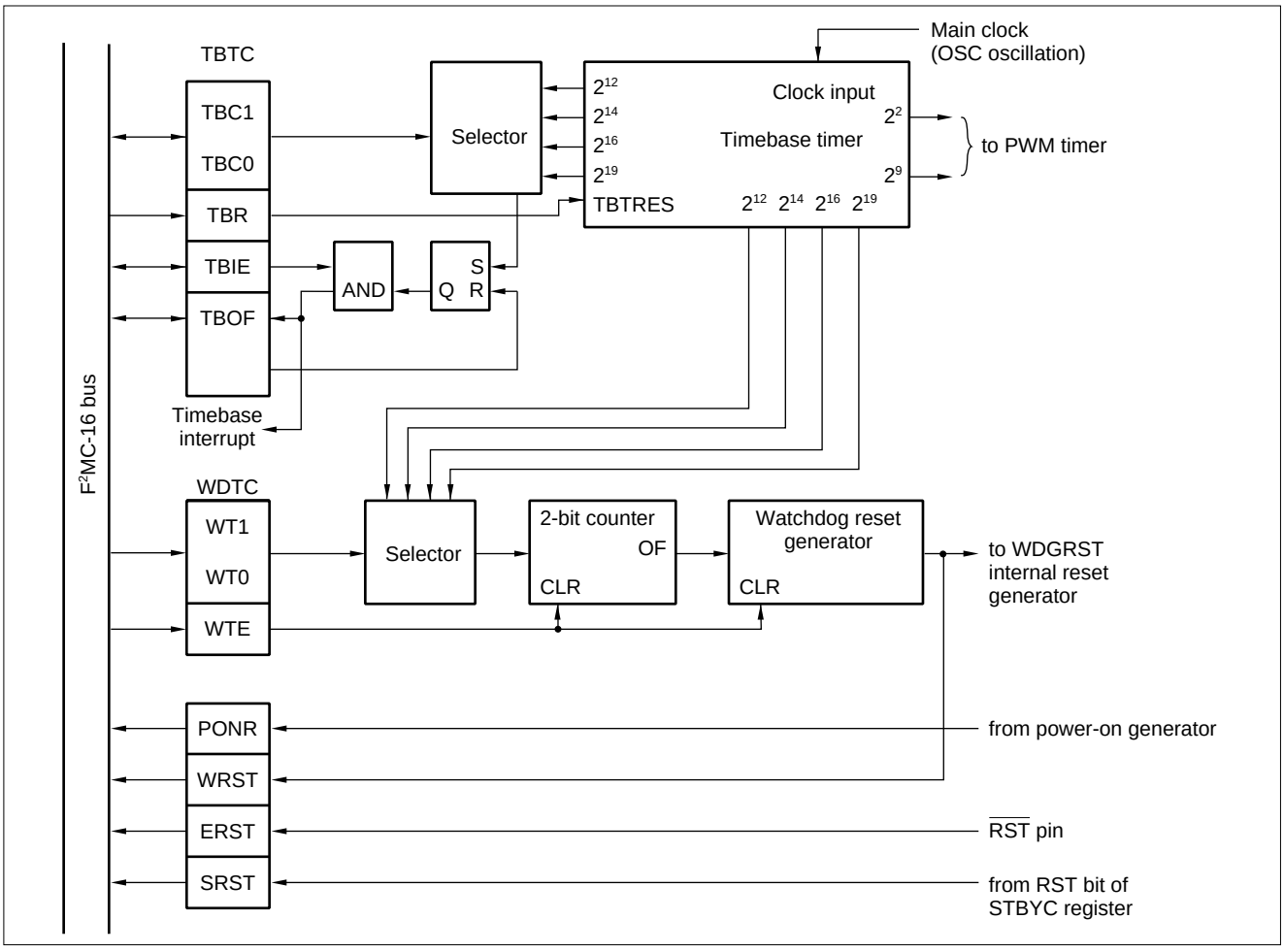
Watchdog timer control register
Address: 0000A8H

bit	7	6	5	4	3	2	1	0	
	PONR	–	WRST	ERST	SRST	WTE	WT1	WT0	WDTC
Read/Write →	(R)	(–)	(R)	(R)	(R)	(W)	(W)	(W)	
Initial value →	(X)	(–)	(X)	(X)	(X)	(1)	(1)	(1)	

Timebase timer control register
Address: 0000A9H

bit	15	14	13	12	11	10	9	8	
	Reserved	–	–	TBIE	TBOF	TBR	TBC1	TBC0	TBTC
Read/Write →	(–)	(–)	(–)	(R/W)	(R/W)	(W)	(R/W)	(R/W)	
Initial value →	(1)	(–)	(–)	(0)	(0)	(1)	(0)	(0)	

(2) Block Diagram



10. Low Power Consumption Controller (CPU intermittent operation function, stable oscillation wait time, and clock multiplier function)

The following operation modes are available: PLL clock mode, PLL sleep mode, clock mode, main clock mode, main sleep mode and stop mode. Operation modes other than PLL clock mode are classified as low power consumption modes.

Main clock mode and main sleep mode are modes where the microcontroller operates using the main clock (OSC oscillation clock) only. In these modes, the main clock divided by two is used as the operation clock and the PLL clock (VCO oscillation clock) is stopped.

In PLL sleep mode and main sleep mode, only the operation clock of the CPU is stopped, while operations besides the CPU clock continue.

In clock mode, only the timebase timer is allowed to operate.

In stop mode, oscillation is stopped, allowing data to be held at the lowest power consumption possible.

The CPU intermittent operation function causes the clock provided to the CPU to function intermittently when accessing registers, internal memory, internal resources and the external bus. This allows processing to be performed at lower power consumption by reducing the CPU execution speed while continuing to provide a high speed clock to internal resources.

The PLL clock multiplier can be selected as 1, 2, 3 or 4 using the CS1 and CS0 bits.

The stable oscillation wait time for the main clock when stop mode is cancelled can be set using the WS1 and WS0 bits.

(1) Register Configuration

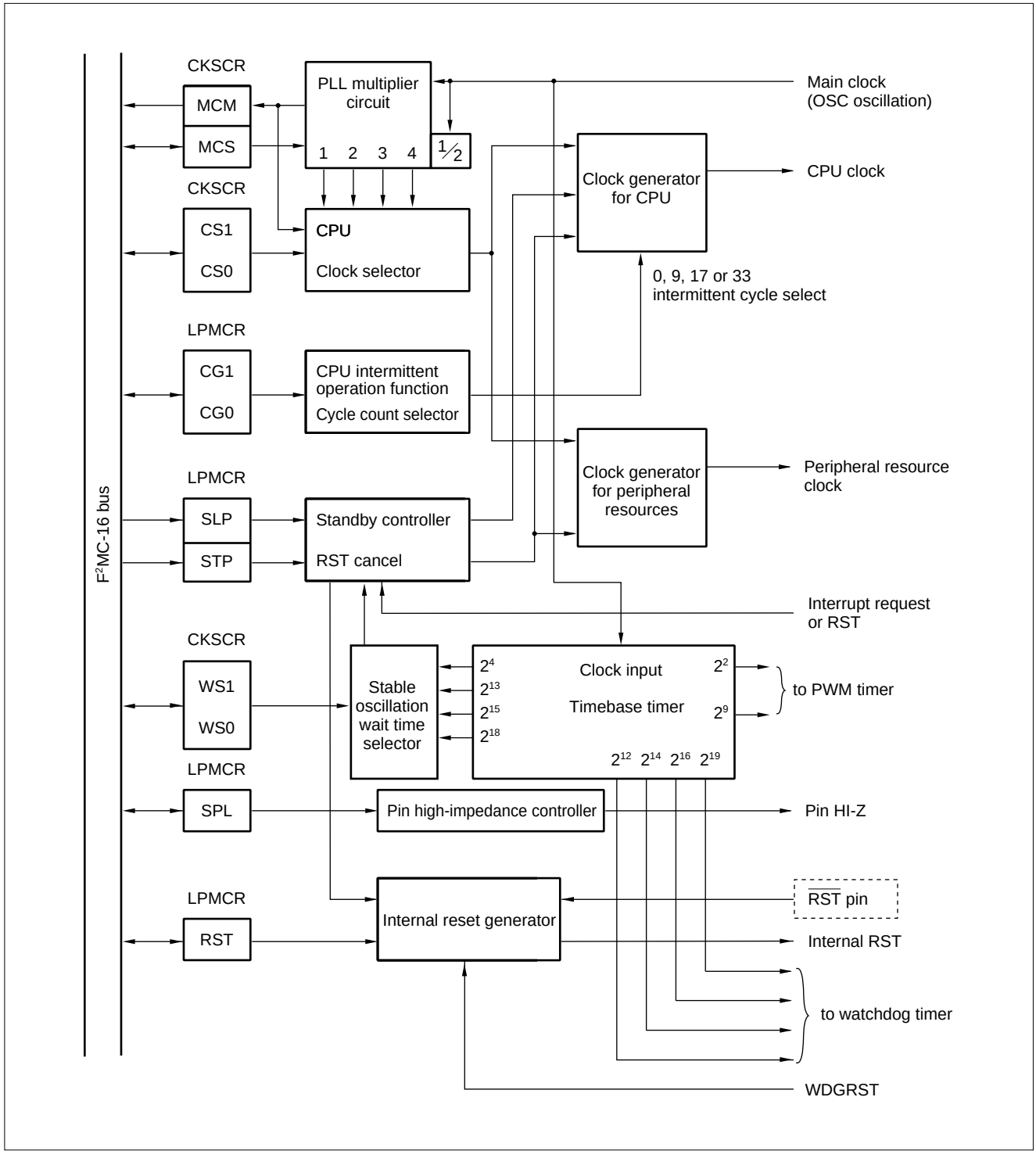
Low power consumption mode control register
Address: 0000A0H

bit	7	6	5	4	3	2	1	0	
	STP	SLP	SPL	RST	Reserved	CG1	CG0	Reserved	LPMCR
Read/Write →	(W)	(W)	(R/W)	(W)	(–)	(R/W)	(R/W)	(–)	
Initial value →	(0)	(0)	(0)	(1)	(1)	(0)	(0)	(0)	

Clock selection register
Address: 0000A1H

bit	15	14	13	12	11	10	9	8	
	Reserved	MCM	WS1	WS0	Reserved	MCS	CS1	CS0	CKSCR
Read/Write →	(–)	(R)	(R/W)	(R/W)	(–)	(R/W)	(R/W)	(R/W)	
Initial value →	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	

(2) Block Diagram

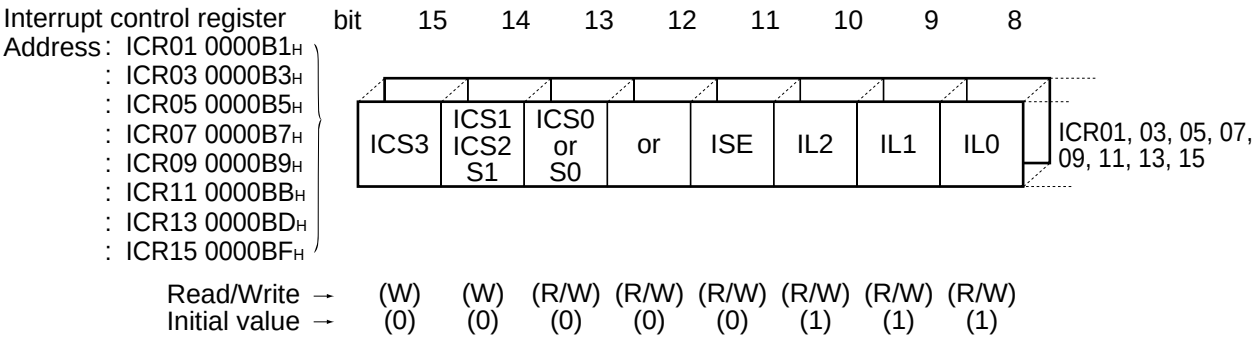


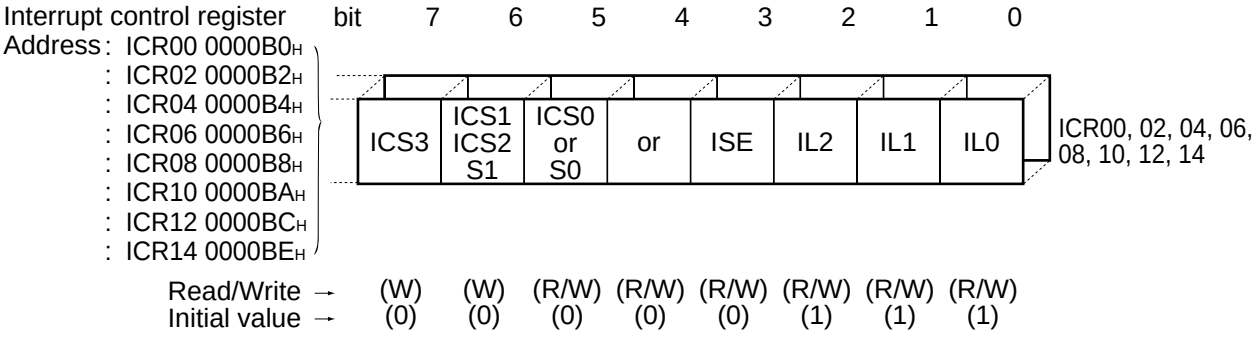
11. Interrupt Controller

The interrupt control register is located within the interrupt controller. Its status conforms to all I/O possessed by the interrupt function. This register includes the following three functions.

- Sets the interrupt level of the corresponding peripheral resource
- Selects whether to use conventional interrupts or extended intelligent I/O services for the interrupt of the corresponding peripheral resource
- Selects the channel for the extended intelligent I/O services

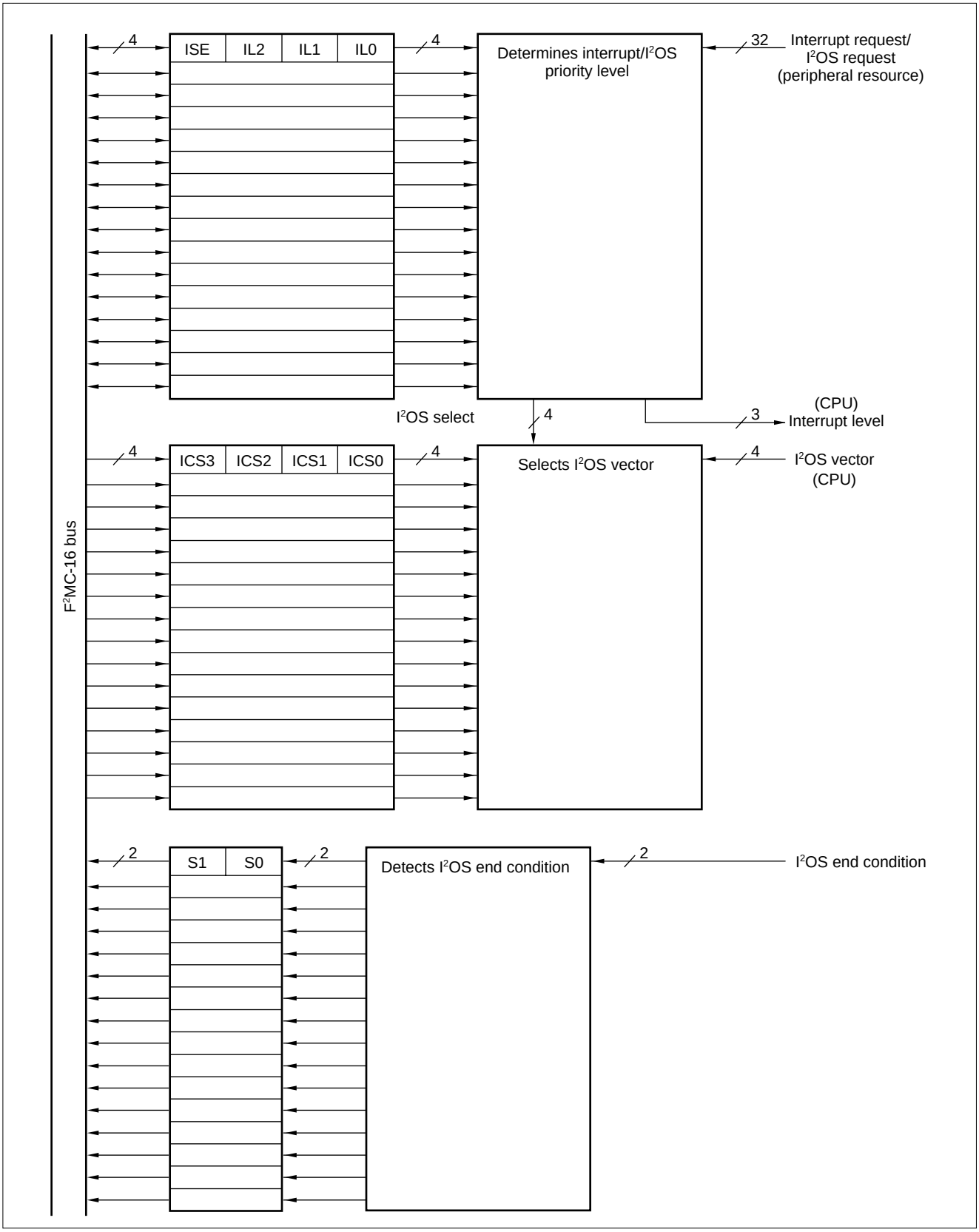
(1) Register Configuration





Note: Since read-modify-write type instructions can cause a malfunction, do not access using these instructions.

(2) Block Diagram



■ ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Rating

(V_{SS} = AV_{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}*1	V_{SS} – 0.3	V_{SS} + 7.0	V	
	V_{AVR}*1	V_{SS} – 0.3	V_{SS} + 7.0	V	
Programming voltage	V_{PP}	V_{SS} – 0.3	13.0	V	*6
Input voltage*2	V_I	V_{SS} – 0.3	V_{CC} + 0.3	V	
Output voltage*2	V_O	V_{SS} – 0.3	V_{CC} + 0.3	V	
“L” level maximum current*3	I_{OL1}	—	10	mA	*7
	I_{OL2}	—	30	mA	*8
“L” level average output current*4	I_{OLAV1}	—	4	mA	*7
	I_{OLAV2}	—	20	mA	*8
“L” level total average output current*5	$\sum I_{OLAV1}$	—	30	mA	*7
	$\sum I_{OLAV2}$	—	60	mA	*8
“H” level maximum output current*3	I_{OH}	—	–10	mA	
“H” level average output current*4	I_{OHAV}	—	–4	mA	
“H” level total average output current*5	$\sum I_{OHAV}$	—	–40	mA	
Power consumption	Pd	—	400	mW	
Operating temperature	T_A	–40	+85	°C	
Storage temperature	T_{stg}	–55	+150	°C	

*1: AV_{CC} and V_{AVR} must not exceed V_{CC}.
*2: V_I and V_O must not exceed V_{CC} + 0.3 V.
*3: Maximum output current specifies the peak value of one corresponding pin.
*4: Average output current specifies the average current within a 100 ms interval flowing through one corresponding pin.
*5: Average total output current specifies the average current within a 100 ms interval flowing through all corresponding pins.
*6: MD2 pin of MB90P663A
*7: Pins excluding P60/RT1/U, P61/RT2/V, P62/RT3/W, P63/X, P64/Y and P65/Z pins
*8: P60/RT1/U, P61/RT2/V, P62/RT3/W, P63/X, P64/Y and P65/Z pins

WARNING: Semiconductor devices can be permanently damaged by application of stress (voltage, current, temperature, etc.) in excess of absolute maximum ratings. Do not exceed these ratings.

2. Recommended Operating Conditions

(V_{SS} = AV_{SS} = 0.0 V)

Parameter	Symbol	Ratings		Unit	Remarks
		Min.	Max.		
Power supply voltage	V_{CC}	2.7	5.5	V	During normal operation
	V_{CC}	2.0	5.5		Stop operation status is held
Operating temperature	T_A	–40	+85	°C	

WARNING: Recommended operating conditions are normal operating ranges for the semiconductor device. All the device's electrical characteristics are warranted when operated within these ranges.

Always use semiconductor devices within the recommended operating conditions. Operation outside these ranges may adversely affect reliability and could result in device failure.

No warranty is made with respect to uses, operating conditions, or combinations not represented on the data sheet. Users considering application outside the listed conditions are advised to contact their FUJITSU representative beforehand.

3. DC Characteristics

(V_{CC} = +2.7 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Conditions	Value			Unit	Remarks
				Min.	Typ.	Max.		
"H" level output voltage	V_{OH}	Except P50 to P57	V_{CC} = 4.5 V I_{OH} = –4.0 mA	V_{CC} – 0.5	—	—	V	
			V_{CC} = 2.7 V I_{OH} = –1.6 mA	V_{CC} – 0.3	—	—	V	
"L" level output voltage	V_{OL1}	Except P60 to P65	V_{CC} = 4.5 V I_{OL} = 4.0 mA	—	—	0.4	V	
			V_{CC} = 2.7 V I_{OL} = 2.0 mA	—	—	0.4	V	
	V_{OL2}	P60 to P65	V_{CC} = 4.5 V I_{OL} = 15.0 mA	—	—	1.0	V	
			V_{CC} = 2.7 V I_{OL} = 2.0 mA	—	—	0.4	V	
"H" level input voltage	V_{IH}	Pins except V_{IHS}, V_{IHM}	—	0.7 V_{CC}	—	V_{CC} + 0.3	V	
	V_{IHS}	Hysteresis input pins	—	0.8 V_{CC}	—	V_{CC} + 0.3	V	*
	V_{IHM}	MD pin	—	V_{CC} – 0.3	—	V_{CC} + 0.3	V	
"L" level input voltage	V_{IL}	Pins except V_{ILS}, V_{ILM}	—	V_{SS} – 0.3	—	0.3 V_{CC}	V	
	V_{ILS}	Hysteresis input pins	—	V_{SS} – 0.3	—	0.2 V_{CC}	V	*
	V_{ILM}	MD pin	—	V_{SS} – 0.3	—	V_{SS} + 0.3	V	
Input leakage current	I_{IL}	Except P50 to P57	V_{CC} = 5.5 V V_{SS} < V_I < V_{CC}	–10	—	10	µA	
Pull-up resistor	R_{PUP}	Pins for which pull-up option is selected	When V_{CC} = 5.0 V	25	—	100	kΩ	
			When V_{CC} = 3.0 V	40	—	200	kΩ	
Pull-down resister	R_{PDN}	Pins for which pull-down options selected	When V_{CC} = 5.0 V	25	80	200	kΩ	
			When V_{CC} = 3.0 V	40	160	400	kΩ	
Supply current	I_{CC}	When V_{CC} = 5.0 V	Internal 16 MHz operation	—	50	70	mA	During normal operation
	I_{CCS}		Internal 16 MHz operation	—	25	30	mA	During sleep
	I_{CC}	When V_{CC} = 3.0 V	Internal 8 MHz operation	—	10	20	mA	During normal operation
	I_{CCS}		Internal 8 MHz operation	—	5	10	mA	During sleep
	I_{CCH}	—	T_A = 25°C	—	0.1	10	µA	During stop
Input capacitance	C_{IN}	Except AV_{CC}, AV_{SS}, V_{CC} and V_{SS}	—	—	10	—	pF	
Open-drain output leakage current	I_{leak}	P50 to P57	—	—	0.1	10	µA	N channel Tr off

* : Applies to pins P40 to P47, P50 to P57, P60 to P66, DTTI and $\overline{RST}$.

4. AC Characteristics

(1) Clock Timing Values

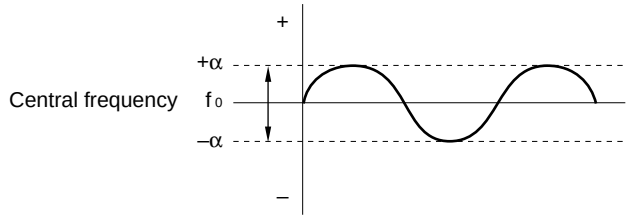
- **Used at V_{CC} = 5.0 V ±10%**

(V_{CC} = +4.5 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Oscillation frequency	F_C	X0, X1	—	3	32	MHz	
Oscillation cycle time	t_C	X0, X1	—	31.25	333	ns	
Frequency fluctuation ratio* (when locked)	Δf	—	—	—	3	%	
Input clock pulse width	P_{WH} P_{WL}	X0	—	10	—	ns	Use duty ratio of 30% to 70% as guideline
Input clock rising and falling times	t_{cr} t_{cf}	X0	—	—	5	ns	
Internal operating clock frequency	f_{CP}	—	—	1.5	16	MHz	
Internal operating clock cycle time	t_{CP}	—	—	62.5	666	ns	

* : The frequency fluctuation ratio represents the maximum fluctuation from the central frequency as a percentage when a multiplier is locked.

$$\Delta f = \frac{|\alpha|}{f_0} \times 100\ (\%)$$



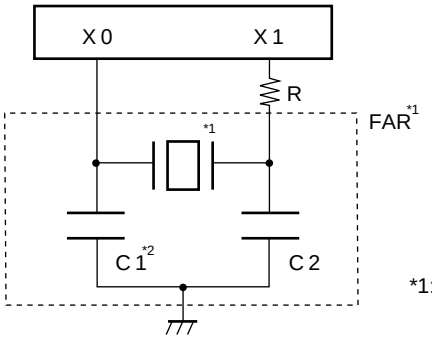
- **Used at V_{CC} = 2.7 V (minimum)**

(V_{CC} = +2.7 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Oscillation frequency	F_C	X0, X1	—	3	16	MHz	
Oscillation cycle time	t_C	X0, X1	—	62.5	333	ns	
Input clock pulse width	P_{WH} P_{WL}	X0	—	20	—	ns	Use duty ratio of 30% to 70% as guideline
Input clock rising and falling times	t_{cr} t_{cf}	X0	—	—	5	ns	
Internal operating clock frequency	f_{CP}	—	—	1.5	8	MHz	
Internal operating clock cycle time	t_{CP}	—	—	125	666	ns	

(2) Recommended Resonator Manufacturers

- **Sample Application of Piezoelectric Resonator (FAR Family)**

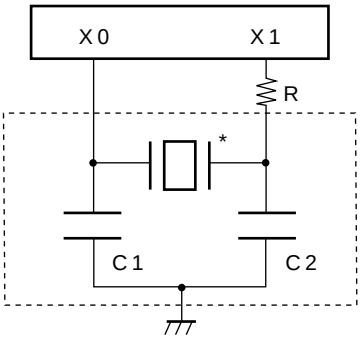


*1: Fujitsu Acoustic Resonator

FAR part number (built-in capacitor type)	Frequency (MHz)	Dumping resistor	Initial deviation of FAR frequency (T_A = +25°C)	Temperature characteristics of FAR frequency (T_A = –20°C to +60°C)	Loading*2 capacitors
FAR-C4CC-02000-L20	2.00	510 Ω	±0.5%	±0.5%	Built-in
FAR-C4SA-04000-M01	4.00	—	±0.5%	±0.5%	
FAR-C4CB-04000-M00		—	±0.5%	±0.5%	
FAR-C4CB-08000-M02	8.00	—	±0.5%	±0.5%	
FAR-C4CB-12000-M02	12.00	—	±0.5%	±0.5%	
FAR-C4CB-16000-M02	16.00	—	±0.5%	±0.5%	
FAR-C4CB-20000-L14B	19.80	—	±0.5%	±0.5%	
FAR-C4CB-24000-L14A	23.76	—	±0.5%	±0.5%	

Inquiry: FUJITSU LIMITED

- **Sample Application of Ceramic Resonator**



- **Mask Products**

Resonator manufacturer*	Resonator	Frequency (MHz)	C1 (pF)	C2 (pF)	R
Kyocera Corporation	KBR-2.0MS	2.00	150	150	—
	PBRC2.00A		150	150	—
	KBR-4.0MSA	4.00	33	33	680 Ω
	KBR-4.0MKS		Built-in	Built-in	680 Ω
	PBRC4.00A		33	33	680 Ω
	PBRC4.00B		Built-in	Built-in	680 Ω
	KBR-6.0MSA	6.00	33	33	—
	KBR-6.0MKS		Built-in	Built-in	—
	PBRC6.00A		33	33	—
	PBRC6.00B		Built-in	Built-in	—
	KBR-8.0M	8.00	33	33	560 Ω
	PBRC8.00A	8.00	33	33	—
	PBRC8.00B		Built-in	Built-in	—
	KBR-10.0M	10.00	33	33	330 Ω
	PBRC10.00B		Built-in	Built-in	680 Ω
	KBR-12.0M	12.00	33	33	330 Ω
	PBRC12.00B		Built-in	Built-in	680 Ω

(Continued)

(Continued)

Resonator manufacturer*	Resonator	Frequency (MHz)	C1 (pF)	C2 (pF)	R
Murata Mfg. Co., Ltd.	CSA2.00MG040	2.00	100	100	—
	CST2.00MG040		Built-in	Built-in	—
	CSA4.00MG040	4.00	100	100	—
	CST4.00MGW040		Built-in	Built-in	—
	CSA6.00MG	6.00	30	30	—
	CST6.00MGW		Built-in	Built-in	—
	CSA8.00MTZ	8.00	30	30	—
	CST8.00MTW		Built-in	Built-in	—
	CSA10.00MTZ	10.00	30	30	—
	CST10.00MTW		Built-in	Built-in	—
	CSA12.00MTZ	12.00	30	30	—
	CST12.00MTW		Built-in	Built-in	—
	CSA16.00MXZ040	16.00	15	15	—
	CST16.00MXW0C3		Built-in	Built-in	—
	CSA20.00MXZ040	20.00	10	10	—
	CSA24.00MXZ040	24.00	5	5	—
	CSA32.00MXZ040	32.00	5	5	—

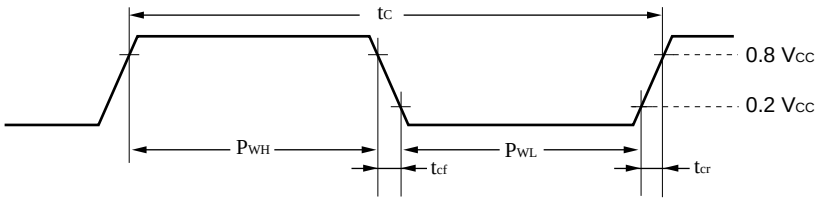
Inquiry: Kyocera Corporation

- AVX Corporation
  North American Sales Headquarters: TEL 1-803-448-9411
- AVX Limited
  European Sales Headquarters: TEL 44-1252-770000
- AVX/Kyocera H.K. Ltd.
  Asian Sales Headquarters: TEL 852-363-3303

Murata Mfg. Co., Ltd.

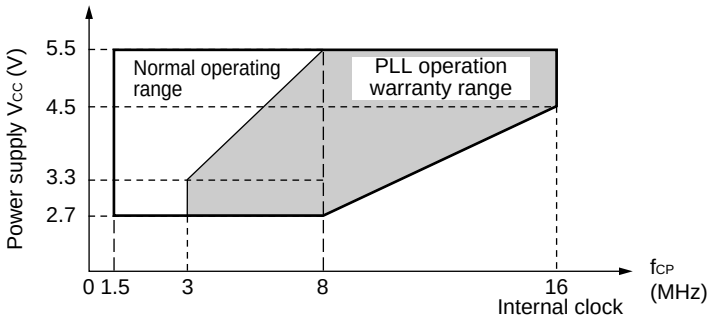
- Murata Electronics North America, Inc.: TEL 1-404-436-1300
- Murata Europe Management GmbH: TEL 49-911-66870
- Murata Electronics Singapore (Pte.) Ltd.: TEL 65-758-4233

- **Clock Timing**

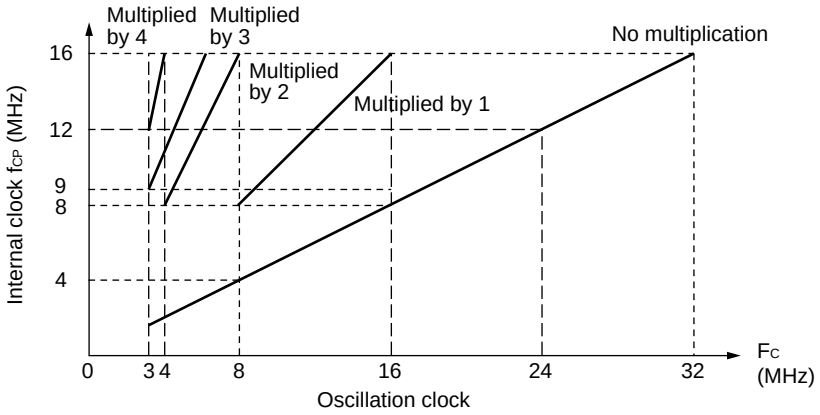


- **PLL Operation Warranty Range**

Relationship between clock frequency and supply voltage



Relationship between oscillator frequency and internal operating clock frequency



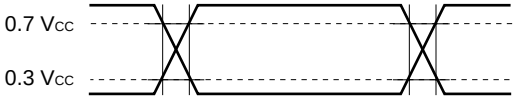
Note: Even in the case of evaluation tool, operation is assured down to 2.7 V.

AC specification values are specified for the measured reference voltages given below.

- **Input Signal Waveforms**

Hysteresis input pin

Pins except hysteresis input and MD input



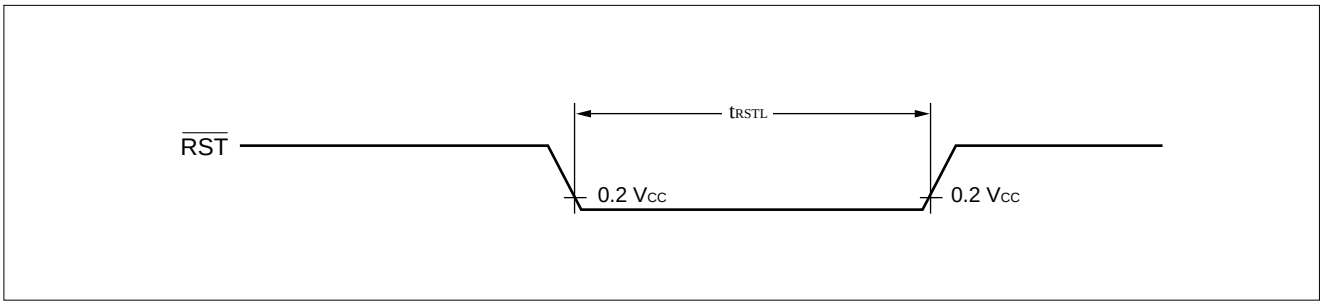
- **Output Signal Waveforms**

Output pin

(3) Reset Input Specifications

(V_{CC} = +2.7 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Reset input time	t_{RSTL}	$\overline{RST}$	—	16	—	Machine cycle	



(4) Power-On Reset

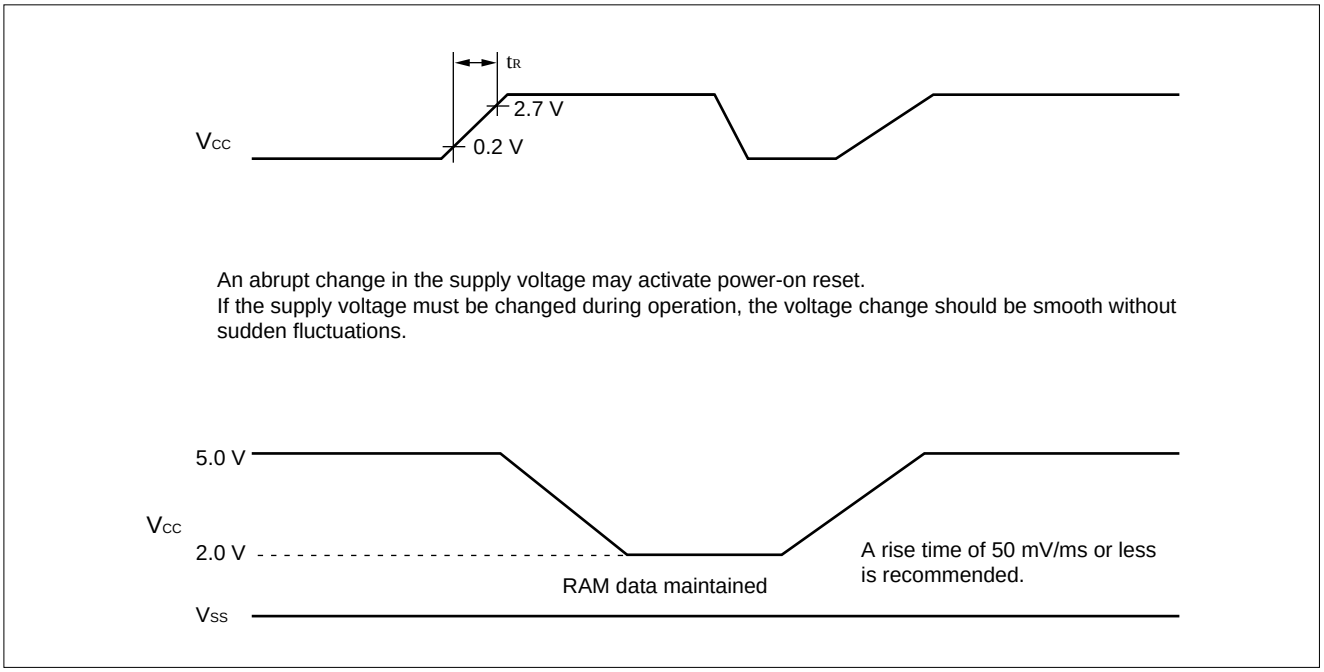
(V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Power supply rise time	t_R	V_{CC}	—	—	30	ms	*
Power supply cutoff time	t_{OFF}	V_{CC}		1	—	ms	Due to repeated operations

* : V_{CC} should be lower than 0.2 V before power supply rise.

Notes: • The above specifications are the numeric values needed for causing a power-on reset.

• There are built in resisters initialized only by power on reset in the device.
Turn on power supply according to the specification at the point of this initialization.



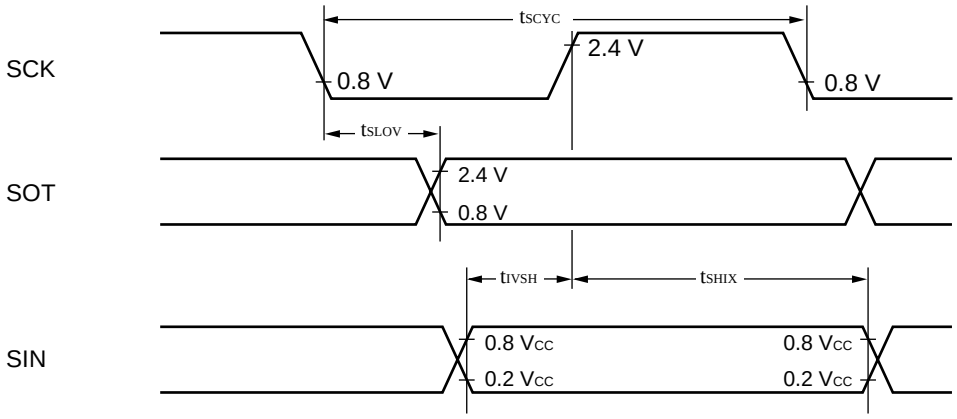
(5) UART timing

(V_{CC} = +2.7 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

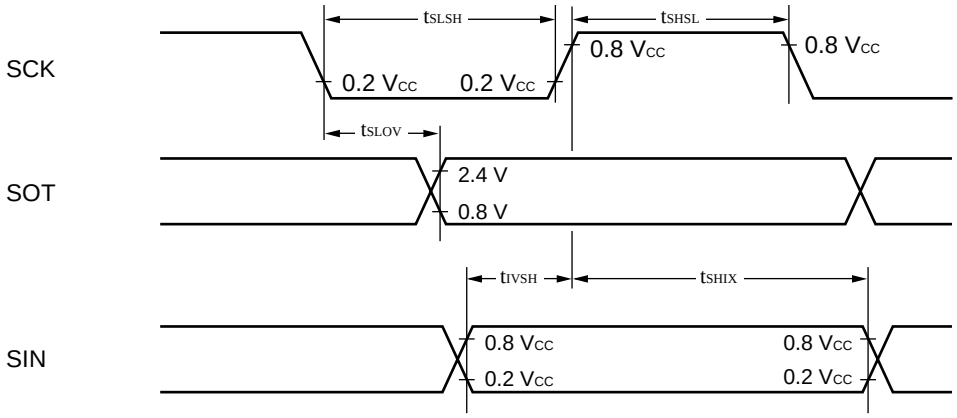
Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Serial clock cycle time	t_{SCYC}	SCK	—	8 t_{CP}	—	ns	C_L = 80 pF + 1 TTL for internal clock operation output pin
SCK ↓ → SOT delay time	t_{SLOV}	SCK SOT	V_{CC} = 5.0 V ±10%	–80	80	ns	
			V_{CC} = 3.0 V ±10%	–120	120	ns	
Valid SIN → SCK ↑	t_{IVSH}	SCK SIN	V_{CC} = 5.0 V ±10%	100	—	ns	
			V_{CC} = 3.0 V ±10%	200	—	ns	
SCK ↑ → valid SIN hold time	t_{SHIX}	SCK SIN	V_{CC} = 5.0 V ±10%	60	—	ns	
			V_{CC} = 3.0 V ±10%	120	—	ns	
Serial clock H pulse width	t_{SHSL}	SCK	—	4 t_{CP}	—	ns	C_L = 80 pF + 1 TTL for external clock operation output pin
Serial clock L pulse width	t_{SLSH}	SCK	—	4 t_{CP}	—	ns	
SCK ↓ → SOT delay time	t_{SLOV}	SCK SOT	V_{CC} = 5.0 V ±10%	—	150	ns	
			V_{CC} = 3.0 V ±10%	—	200	ns	
Valid SIN → SCK ↑	t_{IVSH}	SCK SIN	V_{CC} = 5.0 V ±10%	60	—	ns	
			V_{CC} = 3.0 V ±10%	120	—	ns	
SCK ↑ → valid SIN hold time	t_{SHIX}	SCK SIN	V_{CC} = 5.0 V ±10%	60	—	ns	
			V_{CC} = 3.0 V ±10%	120	—	ns	

Notes: • These are AC specification during CLK synchronous mode.
• C_L is the load capacity value assigned to the pin during testing.
• t_{CP} is the machine cycle time (unit: ns).

- **Internal Shift Clock Mode**



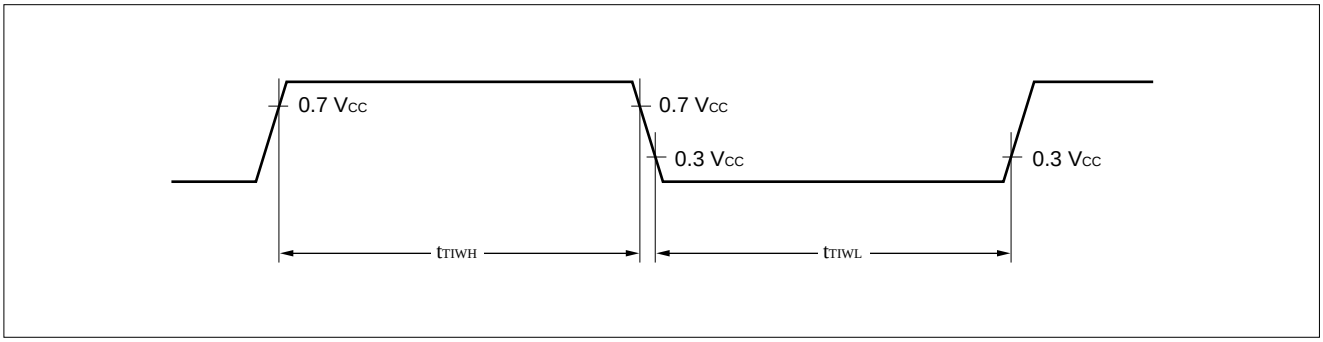
- **External Shift Clock Mode**



(6) Timer input timing

(V_{CC} = +2.7 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Input pulse width	t_{TIWH} t_{TIWL}	TIM0 to TIM3	—	4 t_{CP}	—	ns	

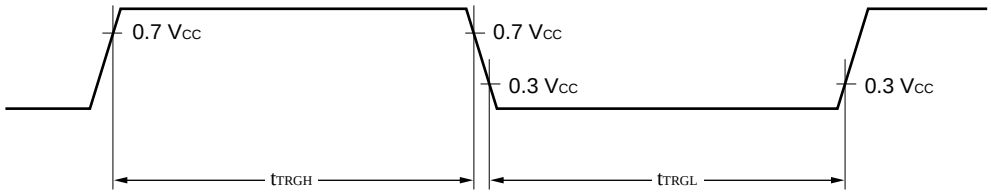


(7) Trigger input timing

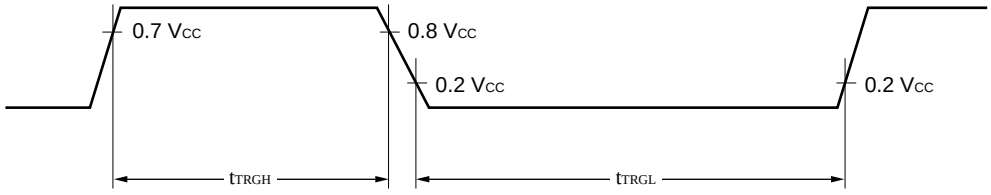
(V_{CC} = +2.7 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Input pulse width	t_{TRGH} t_{TRGL}	$\overline{ATG}$, DTTI, TRG, INT4 to INT7	—	5 t_{CP}	—	ns	
		$\overline{ATG}$, DTTI, TRG, INT0 to INT3		5 t_{CP}	—	ns	

- **INT4 to INT7**



- **INT0 to INT3**



5. Electrical Characteristics of A/D Converter

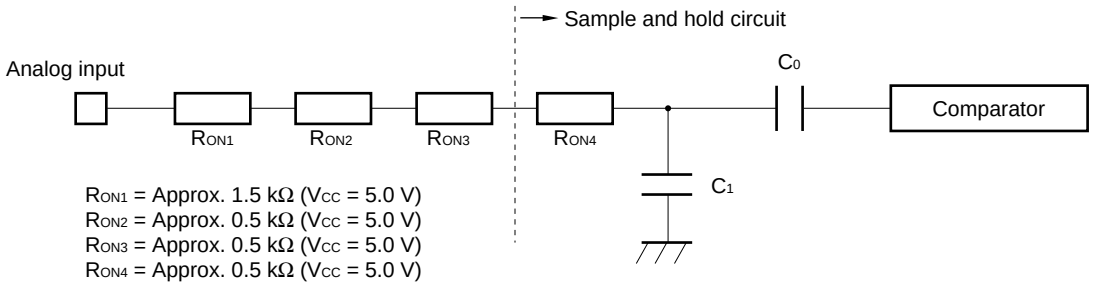
(AV_{CC} = V_{CC} = +2.7 V to +5.5 V, AV_{SS} = V_{SS} = 0.0 V, 2.7 V ≤ AVR, T_A = –40°C to +85°C)

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}	AN0 to AN7	AVR – 4.5	AVR – 1.5	AVR + 0.5	LSB
Conversion time	—	—	6.125*1	—	—	µs
			12.25*2	—	—	µs
Analog port input voltage	I_{AIN}	AN0 to AN7	—	0.1	10	µA
Analog input voltage	V_{AIN}	AN0 to AN7	0	—	AVR	V
Reference voltage	—	AVR	3.5	—	AV_{CC}	V
Supply current	I_A	AV_{CC}	—	3	—	mA
	I_{AH}	AV_{CC}	—	—	5*3	µA
Reference voltage supply current	I_R	AVR	—	200	—	µA
	I_{RH}	AVR	—	—	5*3	µA
Variation between channels	—	AN0 to AN7	—	—	4	LSB

*1: V_{CC} = 5.0 V ±10% at 16 MHz machine clock
*2: V_{CC} = 3.0 V ±10% at 8 MHz machine clock
*3: Current when CPU is stopped and A/D converter is not operating (when V_{CC} = AV_{CC} = AVR = 5.0 V)

Notes: • The relative error becomes larger as the reference voltage (AVR) becomes smaller.
- Be sure to use the A/D converter only when output impedance of the external analog input circuit meets the following conditions.
  External circuit output impedance < approx. 7 kΩ
- If the output impedance of the external circuit is too high, there may not be enough time to sample the analog voltage. (Sampling time = 3.75 µs @4 MHz (equivalent to internal 16 MHz when multiplying by 4))
- For an external capacitor to be provided outside the chip, its capacity should desirably be thousands times larger than of the capacity in the chip taking in consideration the influence of the capacity destribution of the external and internal capacitors.

• **Figure Model of Analog Input Circuit**

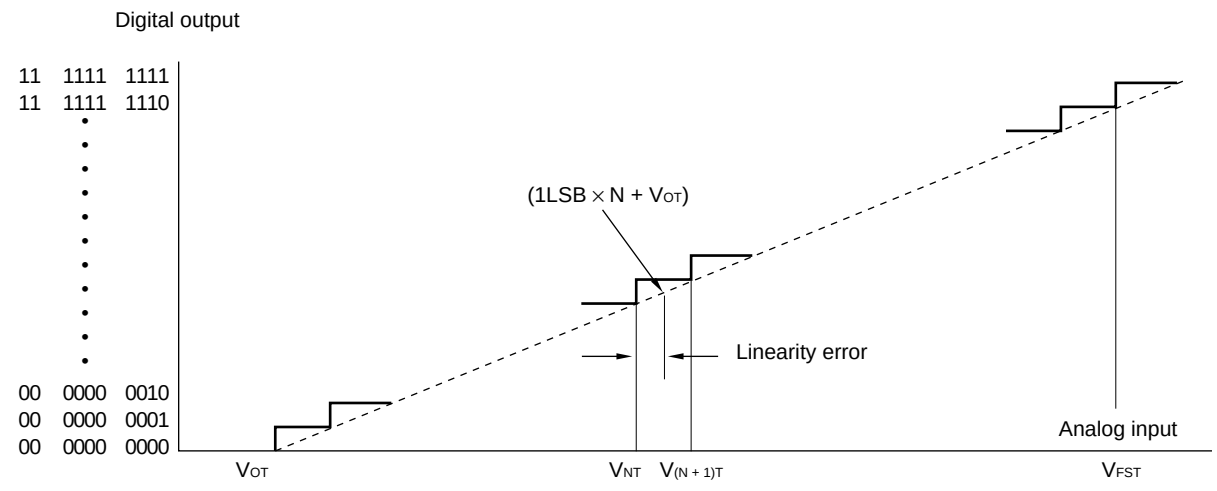


R_{ON1} = Approx. 1.5 kΩ (V_{CC} = 5.0 V)
R_{ON2} = Approx. 0.5 kΩ (V_{CC} = 5.0 V)
R_{ON3} = Approx. 0.5 kΩ (V_{CC} = 5.0 V)
R_{ON4} = Approx. 0.5 kΩ (V_{CC} = 5.0 V)
C_0 = Approx. 60 pF
C_1 = Approx. 4 pF

Note: Use the values shown here as guidelines.

6. Definitions of A/D Converter Terms

Resolution	:	Analog transition observed with an A/D converter. Analog voltage can be divided in 1024 = 2^{10} parts at 10-bit resolution.
Total error	:	This refers to the difference between actual and logical values. This error is caused by offset errors, gain errors, non-linearity errors and noise.
Linearity error	:	Deviation of the line drawn between the zero transition point (00 0000 0000 ↔ 00 0000 0001) and the full-scale transition point (11 1111 1110 ↔ 11 1111 1111) for the device from actual conversion characteristics.
Differential linearity error	:	Deviation from ideal input voltage required to shift output code by one LSB.



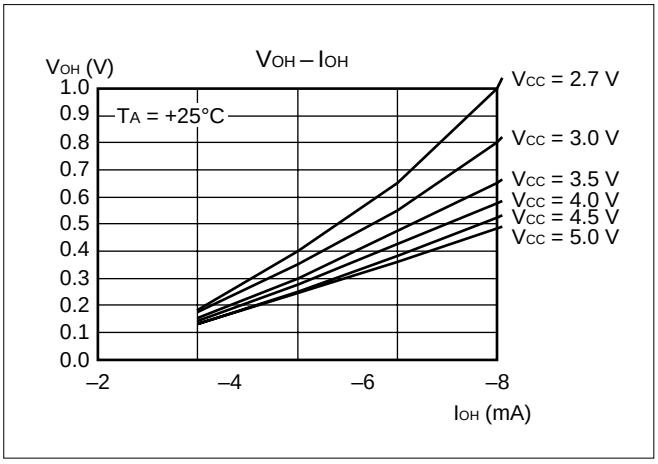
$$1LSB = \frac{V_{FST} - V_{OT}}{1022}$$

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

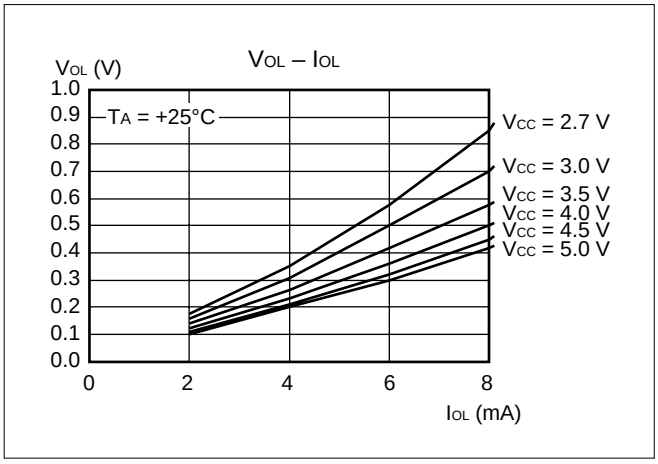
$$\text{Differential linearity error} = \frac{V_{(N+1)T} - V_{NT}}{1LSB} - 1 \text{ (LSB)}$$

■ EXAMPLES CHARACTERISTICS

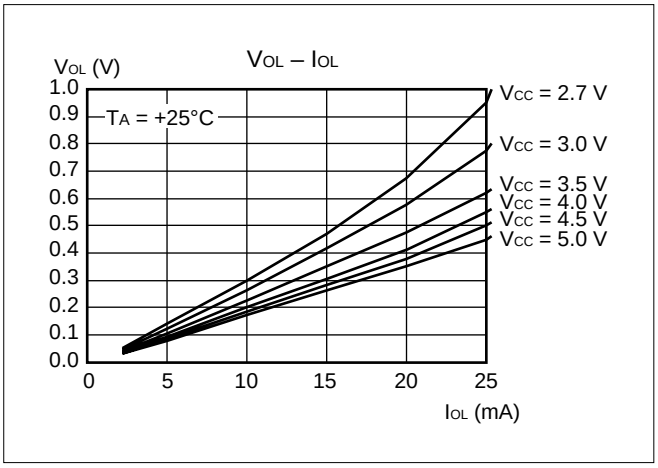
(1) "H" Level Output Voltage



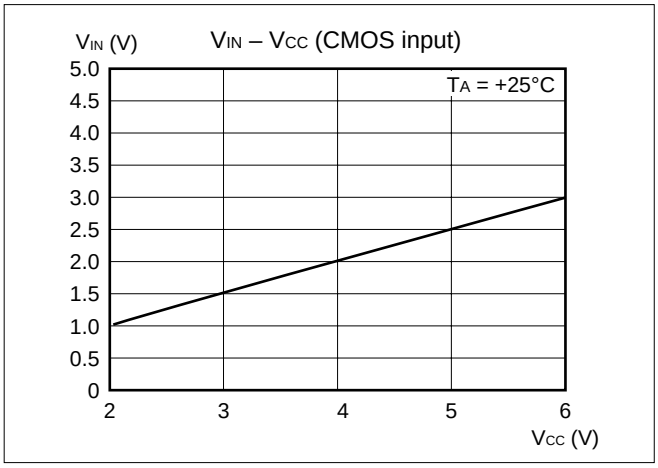
(2) "L" Level Output Voltage



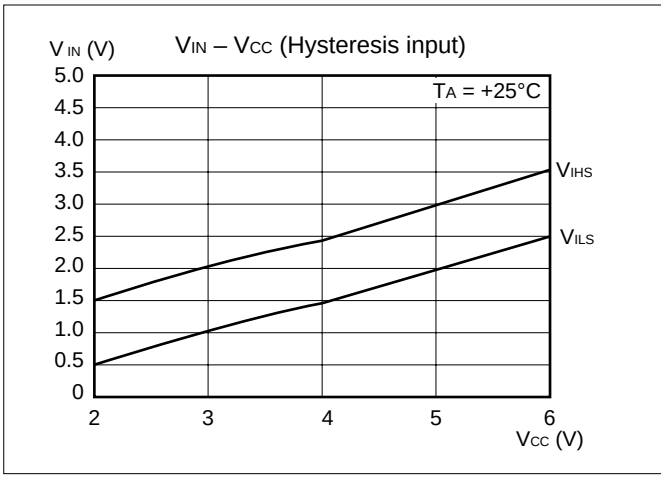
(3) "L" Level Output Voltage (P60 to P65)



(4) "H" Level Input Voltage/"L" Level Input Voltage



(5) "H" Level Input Voltage/"L" Level Input Voltage

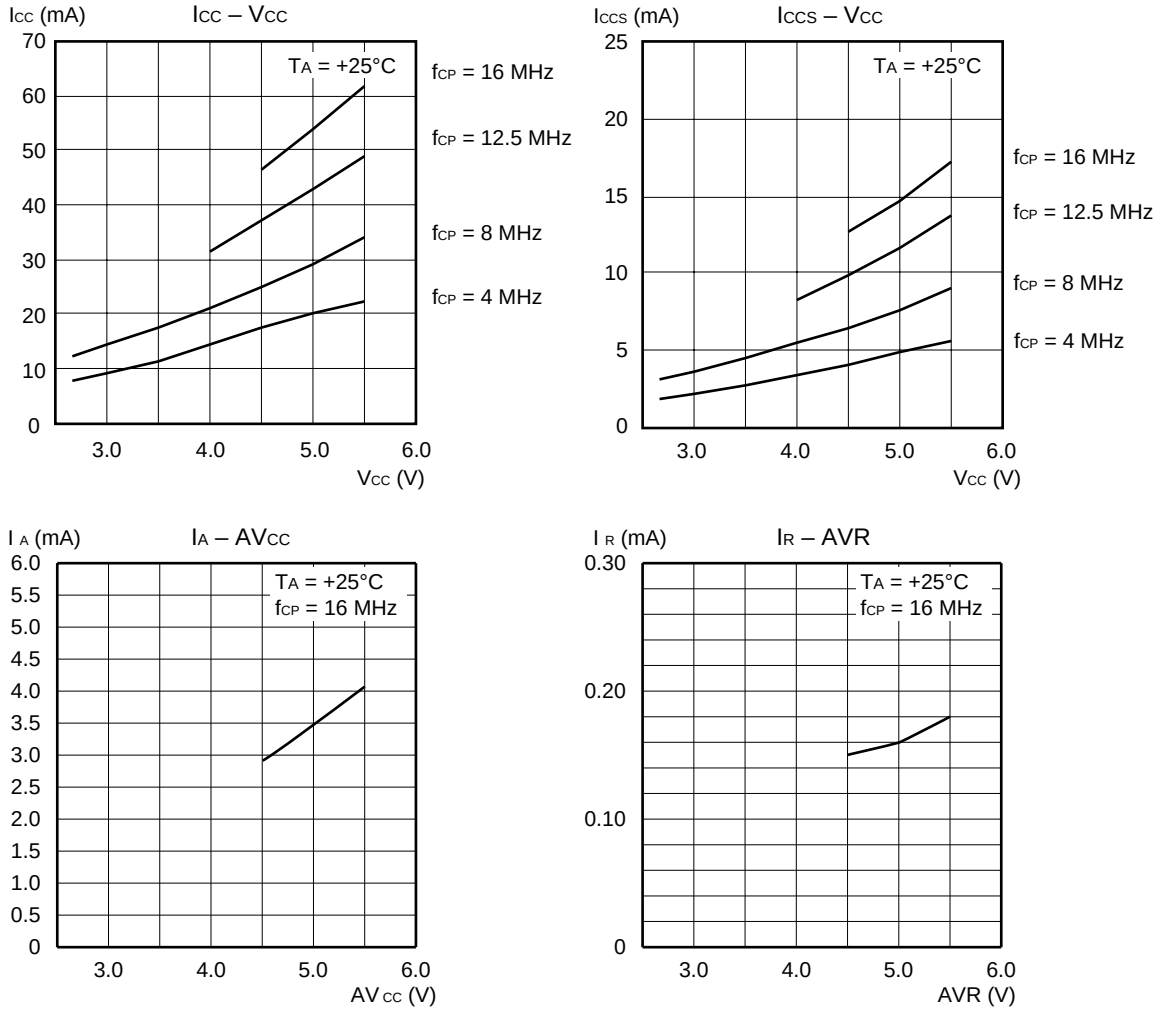


V_{IHS}: Threshold when input voltage in hysteresis characteristics is set to "H" level

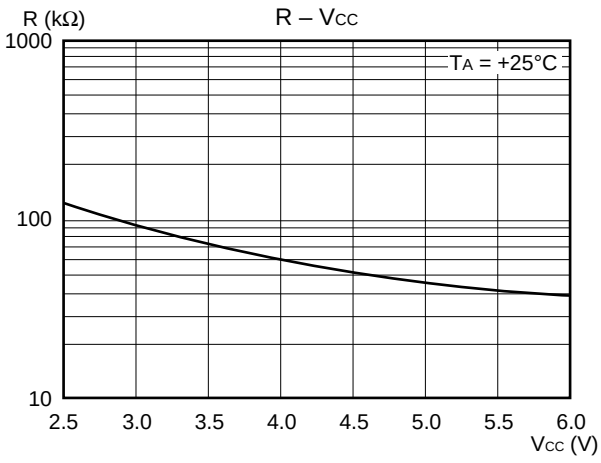
V_{ILS}: Threshold when input voltage in hysteresis characteristics is set to "L" level

(Continued)

(6) Power Supply Current (f_{cp} = Internal frequency)



(7) Pull-up Resistor



■ INSTRUCTIONS (340 INSTRUCTIONS)

Table 1 Explanation of Items in Tables of Instructions

Item	Meaning
Mnemonic	Upper-case letters and symbols: Represented as they appear in assembler. Lower-case letters: Replaced when described in assembler. Numbers after lower-case letters: Indicate the bit width within the instruction.
#	Indicates the number of bytes.
~	Indicates the number of cycles. m : When branching n : When not branching See Table 4 for details about meanings of other letters in items.
RG	Indicates the number of accesses to the register during execution of the instruction. It is used calculate a correction value for intermittent operation of CPU.
B	Indicates the correction value for calculating the number of actual cycles during execution of the instruction. (Table 5) The number of actual cycles during execution of the instruction is the correction value summed with the value in the “~” column.
Operation	Indicates the operation of instruction.
LH	Indicates special operations involving the upper 8 bits of the lower 16 bits of the accumulator. Z : Transfers “0”. X : Extends with a sign before transferring. – : Transfers nothing.
AH	Indicates special operations involving the upper 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 signing and 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: A read-modify-write instruction cannot be used on addresses that have different meanings depending on whether they are read or written.

Table 2 Explanation of Symbols in Tables of Instructions

Symbol	Meaning
A	32-bit accumulator The bit length varies according to the instruction. Byte : Lower 8 bits of AL Word : 16 bits of AL Long : 32 bits of AL:AH
AH AL	Upper 16 bits of A Lower 16 bits of A
SP	Stack pointer (USP or SSP)
PC	Program counter
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	Compact direct addressing
addr16 addr24 ad24 0 to 15 ad24 16 to 23	Direct addressing Physical direct addressing Bit 0 to bit 15 of addr24 Bit 16 to bit 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
vct4 vct8	Vector number (0 to 15) Vector number (0 to 255)
( )b	Bit address

(Continued)

(Continued)

Symbol	Meaning
rel	Branch specification relative to PC
ear eam	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 extension *
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 displacement	2
1C 1D 1E 1F	@RW0 + RW7 @RW1 + RW7 @PC + disp16 addr16	Register indirect with index Register indirect with index PC indirect with 16-bit displacement Direct address	0 0 2 2

Note: The number of bytes in the address extension is indicated by the “+” symbol in the “#” (number of bytes) column in the tables of instructions.

Table 4 Number of Execution Cycles for Each Type of Addressing

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

Note: “(a)” is used in the “~” (number of states) column and column B (correction value) in the tables of instructions.

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

Operand	(b) byte		(c) word		(d) long	
	Number of cycles	Number of access	Number of cycles	Number of access	Number of cycles	Number of access
Internal register	+0	1	+0	1	+0	2
Internal memory even address Internal memory odd address	+0 +0	1 1	+0 +2	1 2	+0 +4	2 4
Even address on external data bus (16 bits) Odd address on external data bus (16 bits)	+1 +1	1 1	+1 +4	1 2	+2 +8	2 4
External data bus (8 bits)	+1	1	+4	2	+8	4

Notes: • “(b)”, “(c)”, and “(d)” are used in the “~” (number of states) column and column B (correction value) in the tables of instructions.
• When the external data bus is used, it is necessary to add in the number of wait cycles used for ready input and automatic ready.

Table 6 Correction Values for Number of Cycles Used to Calculate Number of Program Fetch Cycles

Instruction	Byte boundary	Word boundary
Internal memory	—	+2
External data bus (16 bits)	—	+3
External data bus (8 bits)	+3	—

Notes: • When the external data bus is used, it is necessary to add in the number of wait cycles used for ready input and automatic ready.
• Because instruction execution is not slowed down by all program fetches in actuality, these correction values should be used for “worst case” calculations.

Table 7 Transfer Instructions (Byte) [41 Instructions]

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOV A, dir	2	3	0	(b)	byte (A) ← (dir)	Z	*	–	–	–	*	*	–	–	–
MOV A, addr16	3	4	0	(b)	byte (A) ← (addr16)	Z	*	–	–	–	*	*	–	–	–
MOV A, Ri	1	2	1	0	byte (A) ← (Ri)	Z	*	–	–	–	*	*	–	–	–
MOV A, ear	2	2	1	0	byte (A) ← (ear)	Z	*	–	–	–	*	*	–	–	–
MOV A, eam	2+	3+ (a)	0	(b)	byte (A) ← (eam)	Z	*	–	–	–	*	*	–	–	–
MOV A, io	2	3	0	(b)	byte (A) ← (io)	Z	*	–	–	–	*	*	–	–	–
MOV A, #imm8	2	2	0	0	byte (A) ← imm8	Z	*	–	–	–	*	*	–	–	–
MOV A, @A	2	3	0	(b)	byte (A) ← ((A))	Z	–	–	–	–	*	*	–	–	–
MOV A, @RLi+disp8	3	10	2	(b)	byte (A) ← ((RLi)+disp8)	Z	*	–	–	–	*	*	–	–	–
MOVN A, #imm4	1	1	0	0	byte (A) ← imm4	Z	*	–	–	–	R	*	–	–	–
MOVX A, dir	2	3	0	(b)	byte (A) ← (dir)	X	*	–	–	–	*	*	–	–	–
MOVX A, addr16	3	4	0	(b)	byte (A) ← (addr16)	X	*	–	–	–	*	*	–	–	–
MOVX A, Ri	2	2	1	0	byte (A) ← (Ri)	X	*	–	–	–	*	*	–	–	–
MOVX A, ear	2	2	1	0	byte (A) ← (ear)	X	*	–	–	–	*	*	–	–	–
MOVX A, eam	2+	3+ (a)	0	(b)	byte (A) ← (eam)	X	*	–	–	–	*	*	–	–	–
MOVX A, io	2	3	0	(b)	byte (A) ← (io)	X	*	–	–	–	*	*	–	–	–
MOVX A, #imm8	2	2	0	0	byte (A) ← imm8	X	*	–	–	–	*	*	–	–	–
MOVX A, @A	2	3	0	(b)	byte (A) ← ((A))	X	–	–	–	–	*	*	–	–	–
MOVX A,@RWi+disp8	2	5	1	(b)	byte (A) ← ((RWi)+disp8)	X	*	–	–	–	*	*	–	–	–
MOVX A, @RLi+disp8	3	10	2	(b)	byte (A) ← ((RLi)+disp8)	X	*	–	–	–	*	*	–	–	–
MOV dir, A	2	3	0	(b)	byte (dir) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV addr16, A	3	4	0	(b)	byte (addr16) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV Ri, A	1	2	1	0	byte (Ri) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV ear, A	2	2	1	0	byte (ear) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV eam, A	2+	3+ (a)	0	(b)	byte (eam) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV io, A	2	3	0	(b)	byte (io) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV @RLi+disp8, A	3	10	2	(b)	byte ((RLi) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV Ri, ear	2	3	2	0	byte (Ri) ← (ear)	–	–	–	–	–	*	*	–	–	–
MOV Ri, eam	2+	4+ (a)	1	(b)	byte (Ri) ← (eam)	–	–	–	–	–	*	*	–	–	–
MOV ear, Ri	2	4	2	0	byte (ear) ← (Ri)	–	–	–	–	–	*	*	–	–	–
MOV eam, Ri	2+	5+ (a)	1	(b)	byte (eam) ← (Ri)	–	–	–	–	–	*	*	–	–	–
MOV Ri, #imm8	2	2	1	0	byte (Ri) ← imm8	–	–	–	–	–	*	*	–	–	–
MOV io, #imm8	3	5	0	(b)	byte (io) ← imm8	–	–	–	–	–	–	–	–	–	–
MOV dir, #imm8	3	5	0	(b)	byte (dir) ← imm8	–	–	–	–	–	–	–	–	–	–
MOV ear, #imm8	3	2	1	0	byte (ear) ← imm8	–	–	–	–	–	*	*	–	–	–
MOV eam, #imm8	3+	4+ (a)	0	(b)	byte (eam) ← imm8	–	–	–	–	–	–	–	–	–	–
MOV @AL, AH /MOV @A, T	2	3	0	(b)	byte ((A)) ← (AH)	–	–	–	–	–	*	*	–	–	–
XCH A, ear	2	4	2	0	byte (A) ↔ (ear)	Z	–	–	–	–	–	–	–	–	–
XCH A, eam	2+	5+ (a)	0	2× (b)	byte (A) ↔ (eam)	Z	–	–	–	–	–	–	–	–	–
XCH Ri, ear	2	7	4	0	byte (Ri) ↔ (ear)	–	–	–	–	–	–	–	–	–	–
XCH Ri, eam	2+	9+ (a)	2	2× (b)	byte (Ri) ↔ (eam)	–	–	–	–	–	–	–	–	–	–

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

Table 8 Transfer Instructions (Word/Long Word) [38 Instructions]

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVW A, dir	2	3	0	(c)	word (A) ← (dir)	–	*	–	–	–	*	*	–	–	–
MOVW A, addr16	3	4	0	(c)	word (A) ← (addr16)	–	*	–	–	–	*	*	–	–	–
MOVW A, SP	1	1	0	0	word (A) ← (SP)	–	*	–	–	–	*	*	–	–	–
MOVW A, RWi	1	2	1	0	word (A) ← (RWi)	–	*	–	–	–	*	*	–	–	–
MOVW A, ear	2	2	1	0	word (A) ← (ear)	–	*	–	–	–	*	*	–	–	–
MOVW A, eam	2+	3+ (a)	0	(c)	word (A) ← (eam)	–	*	–	–	–	*	*	–	–	–
MOVW A, io	2	3	0	(c)	word (A) ← (io)	–	*	–	–	–	*	*	–	–	–
MOVW A, @A	2	3	0	(c)	word (A) ← ((A))	–	–	–	–	–	*	*	–	–	–
MOVW A, #imm16	3	2	0	0	word (A) ← imm16	–	*	–	–	–	*	*	–	–	–
MOVW A, @RWi+disp8	2	5	1	(c)	word (A) ← ((RWi) +disp8)	–	*	–	–	–	*	*	–	–	–
MOVW A, @RLi+disp8	3	10	2	(c)	word (A) ← ((RLi) +disp8)	–	*	–	–	–	*	*	–	–	–
MOVW dir, A	2	3	0	(c)	word (dir) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW addr16, A	3	4	0	(c)	word (addr16) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW SP, A	1	1	0	0	word (SP) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, A	1	2	1	0	word (RWi) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW ear, A	2	2	1	0	word (ear) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW eam, A	2+	3+ (a)	0	(c)	word (eam) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW io, A	2	3	0	(c)	word (io) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW @RWi+disp8, A	2	5	1	(c)	word ((RWi) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW @RLi+disp8, A	3	10	2	(c)	word ((RLi) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, ear	2	3	2	(0)	word (RWi) ← (ear)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, eam	2+	4+ (a)	1	(c)	word (RWi) ← (eam)	–	–	–	–	–	*	*	–	–	–
MOVW ear, RWi	2	4	2	0	word (ear) ← (RWi)	–	–	–	–	–	*	*	–	–	–
MOVW eam, RWi	2+	5+ (a)	1	(c)	word (eam) ← (RWi)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, #imm16	3	2	1	0	word (RWi) ← imm16	–	–	–	–	–	*	*	–	–	–
MOVW io, #imm16	4	5	0	(c)	word (io) ← imm16	–	–	–	–	–	–	–	–	–	–
MOVW ear, #imm16	4	2	1	0	word (ear) ← imm16	–	–	–	–	–	*	*	–	–	–
MOVW eam, #imm16	4+	4+ (a)	0	(c)	word (eam) ← imm16	–	–	–	–	–	–	–	–	–	–
MOVW AL, AH /MOVW @A, T	2	3	0	(c)	word ((A)) ← (AH)	–	–	–	–	–	*	*	–	–	–
XCHW A, ear	2	4	2	0	word (A) ↔ (ear)	–	–	–	–	–	–	–	–	–	–
XCHW A, eam	2+	5+ (a)	0	2× (c)	word (A) ↔ (eam)	–	–	–	–	–	–	–	–	–	–
XCHW RWi, ear	2	7	4	0	word (RWi) ↔ (ear)	–	–	–	–	–	–	–	–	–	–
XCHW RWi, eam	2+	9+ (a)	2	2× (c)	word (RWi) ↔ (eam)	–	–	–	–	–	–	–	–	–	–
MOVL A, ear	2	4	2	0	long (A) ← (ear)	–	–	–	–	–	*	*	–	–	–
MOVL A, eam	2+	5+ (a)	0	(d)	long (A) ← (eam)	–	–	–	–	–	*	*	–	–	–
MOVL A, #imm32	5	3	0	0	long (A) ← imm32	–	–	–	–	–	*	*	–	–	–
MOVL ear, A	2	4	2	0	long (ear) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVL eam, A	2+	5+ (a)	0	(d)	long (eam) ← (A)	–	–	–	–	–	*	*	–	–	–

Note: For an explanation of “(a)” to “(d)”, refer to Table 4, “Number of Execution Cycles for Each Type 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	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
ADD A,#imm8	2	2	0	0	byte (A) ← (A) +imm8	Z	–	–	–	–	*	*	*	*	–
ADD A, dir	2	5	0	(b)	byte (A) ← (A) +(dir)	Z	–	–	–	–	*	*	*	*	–
ADD A, ear	2	3	1	0	byte (A) ← (A) +(ear)	Z	–	–	–	–	*	*	*	*	–
ADD A, eam	2+	4+ (a)	0	(b)	byte (A) ← (A) +(eam)	Z	–	–	–	–	*	*	*	*	–
ADD ear, A	2	3	2	0	byte (ear) ← (ear) + (A)	–	–	–	–	–	*	*	*	*	–
ADD eam, A	2+	5+ (a)	0	2× (b)	byte (eam) ← (eam) + (A)	Z	–	–	–	–	*	*	*	*	*
ADDC A	1	2	0	0	byte (A) ← (AH) + (AL) + (C)	Z	–	–	–	–	*	*	*	*	–
ADDC A, ear	2	3	1	0	byte (A) ← (A) + (ear) + (C)	Z	–	–	–	–	*	*	*	*	–
ADDC A, eam	2+	4+ (a)	0	(b)	byte (A) ← (A) + (eam) + (C)	Z	–	–	–	–	*	*	*	*	–
ADDDC A	1	3	0	0	byte (A) ← (AH) + (AL) + (C) (decimal)	Z	–	–	–	–	*	*	*	*	–
SUB A, #imm8	2	2	0	0	byte (A) ← (A) –imm8	Z	–	–	–	–	*	*	*	*	–
SUB A, dir	2	5	0	(b)	byte (A) ← (A) – (dir)	Z	–	–	–	–	*	*	*	*	–
SUB A, ear	2	3	1	0	byte (A) ← (A) – (ear)	Z	–	–	–	–	*	*	*	*	–
SUB A, eam	2+	4+ (a)	0	(b)	byte (A) ← (A) – (eam)	Z	–	–	–	–	*	*	*	*	–
SUB ear, A	2	3	2	0	byte (ear) ← (ear) – (A)	–	–	–	–	–	*	*	*	*	–
SUB eam, A	2+	5+ (a)	0	2× (b)	byte (eam) ← (eam) – (A)	–	–	–	–	–	*	*	*	*	*
SUBC A	1	2	0	0	byte (A) ← (AH) – (AL) – (C)	Z	–	–	–	–	*	*	*	*	–
SUBC A, ear	2	3	1	0	byte (A) ← (A) – (ear) – (C)	Z	–	–	–	–	*	*	*	*	–
SUBC A, eam	2+	4+ (a)	0	(b)	byte (A) ← (A) – (eam) – (C)	Z	–	–	–	–	*	*	*	*	–
SUBDC A	1	3	0	0	byte (A) ← (AH) – (AL) – (C) (decimal)	Z	–	–	–	–	*	*	*	*	–
ADDW A	1	2	0	0	word (A) ← (AH) + (AL)	–	–	–	–	–	*	*	*	*	–
ADDW A, ear	2	3	1	0	word (A) ← (A) +(ear)	–	–	–	–	–	*	*	*	*	–
ADDW A, eam	2+	4+ (a)	0	(c)	word (A) ← (A) +(eam)	–	–	–	–	–	*	*	*	*	–
ADDW A, #imm16	3	2	0	0	word (A) ← (A) +imm16	–	–	–	–	–	*	*	*	*	–
ADDW ear, A	2	3	2	0	word (ear) ← (ear) + (A)	–	–	–	–	–	*	*	*	*	–
ADDW eam, A	2+	5+ (a)	0	2× (c)	word (eam) ← (eam) + (A)	–	–	–	–	–	*	*	*	*	*
ADDCW A, ear	2	3	1	0	word (A) ← (A) + (ear) + (C)	–	–	–	–	–	*	*	*	*	–
ADDCW A, eam	2+	4+ (a)	0	(c)	word (A) ← (A) + (eam) + (C)	–	–	–	–	–	*	*	*	*	–
SUBW A	1	2	0	0	word (A) ← (AH) – (AL)	–	–	–	–	–	*	*	*	*	–
SUBW A, ear	2	3	1	0	word (A) ← (A) – (ear)	–	–	–	–	–	*	*	*	*	–
SUBW A, eam	2+	4+ (a)	0	(c)	word (A) ← (A) – (eam)	–	–	–	–	–	*	*	*	*	–
SUBW A, #imm16	3	2	0	0	word (A) ← (A) –imm16	–	–	–	–	–	*	*	*	*	–
SUBW ear, A	2	3	2	0	word (ear) ← (ear) – (A)	–	–	–	–	–	*	*	*	*	–
SUBW eam, A	2+	5+ (a)	0	2× (c)	word (eam) ← (eam) – (A)	–	–	–	–	–	*	*	*	*	*
SUBCW A, ear	2	3	1	0	word (A) ← (A) – (ear) – (C)	–	–	–	–	–	*	*	*	*	–
SUBCW A, eam	2+	4+ (a)	0	(c)	word (A) ← (A) – (eam) – (C)	–	–	–	–	–	*	*	*	*	–
ADDL A, ear	2	6	2	0	long (A) ← (A) + (ear)	–	–	–	–	–	*	*	*	*	–
ADDL A, eam	2+	7+ (a)	0	(d)	long (A) ← (A) + (eam)	–	–	–	–	–	*	*	*	*	–
ADDL A, #imm32	5	4	0	0	long (A) ← (A) +imm32	–	–	–	–	–	*	*	*	*	–
SUBL A, ear	2	6	2	0	long (A) ← (A) – (ear)	–	–	–	–	–	*	*	*	*	–
SUBL A, eam	2+	7+ (a)	0	(d)	long (A) ← (A) – (eam)	–	–	–	–	–	*	*	*	*	–
SUBL A, #imm32	5	4	0	0	long (A) ← (A) –imm32	–	–	–	–	–	*	*	*	*	–

Note: For an explanation of "(a)" to "(d)", refer to Table 4, "Number of Execution Cycles for Each Type 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		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
INC	ear	2	2	2	0	byte (ear) ← (ear) +1	–	–	–	–	–	*	*	*	–	–
INC	eam	2+	5+ (a)	0	2× (b)	byte (eam) ← (eam) +1	–	–	–	–	–	*	*	*	–	*
DEC	ear	2	3	2	0	byte (ear) ← (ear) –1	–	–	–	–	–	*	*	*	–	–
DEC	eam	2+	5+ (a)	0	2× (b)	byte (eam) ← (eam) –1	–	–	–	–	–	*	*	*	–	*
INCW	ear	2	3	2	0	word (ear) ← (ear) +1	–	–	–	–	–	*	*	*	–	–
INCW	eam	2+	5+ (a)	0	2× (c)	word (eam) ← (eam) +1	–	–	–	–	–	*	*	*	–	*
DECW	ear	2	3	2	0	word (ear) ← (ear) –1	–	–	–	–	–	*	*	*	–	–
DECW	eam	2+	5+ (a)	0	2× (c)	word (eam) ← (eam) –1	–	–	–	–	–	*	*	*	–	*
INCL	ear	2	7	4	0	long (ear) ← (ear) +1	–	–	–	–	–	*	*	*	–	–
INCL	eam	2+	9+ (a)	0	2× (d)	long (eam) ← (eam) +1	–	–	–	–	–	*	*	*	–	*
DECL	ear	2	7	4	0	long (ear) ← (ear) –1	–	–	–	–	–	*	*	*	–	–
DECL	eam	2+	9+ (a)	0	2× (d)	long (eam) ← (eam) –1	–	–	–	–	–	*	*	*	–	*

Note: For an explanation of “(a)” to “(d)”, refer to Table 4, “Number of Execution Cycles for Each Type 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		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
CMP	A	1	1	0	0	byte (AH) – (AL)	–	–	–	–	–	*	*	*	*	–
CMP	A, ear	2	2	1	0	byte (A) ← (ear)	–	–	–	–	–	*	*	*	*	–
CMP	A, eam	2+	3+ (a)	0	(b)	byte (A) ← (eam)	–	–	–	–	–	*	*	*	*	–
CMP	A, #imm8	2	2	0	0	byte (A) ← imm8	–	–	–	–	–	*	*	*	*	–
CMPW	A	1	1	0	0	word (AH) – (AL)	–	–	–	–	–	*	*	*	*	–
CMPW	A, ear	2	2	1	0	word (A) ← (ear)	–	–	–	–	–	*	*	*	*	–
CMPW	A, eam	2+	3+ (a)	0	(c)	word (A) ← (eam)	–	–	–	–	–	*	*	*	*	–
CMPW	A, #imm16	3	2	0	0	word (A) ← imm16	–	–	–	–	–	*	*	*	*	–
CMPL	A, ear	2	6	2	0	word (A) ← (ear)	–	–	–	–	–	*	*	*	*	–
CMPL	A, eam	2+	7+ (a)	0	(d)	word (A) ← (eam)	–	–	–	–	–	*	*	*	*	–
CMPL	A, #imm32	5	3	0	0	word (A) ← imm32	–	–	–	–	–	*	*	*	*	–

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

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

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

*1: 3 when the result is zero, 7 when an overflow occurs, and 15 normally.
*2: 4 when the result is zero, 8 when an overflow occurs, and 16 normally.
*3: 6 + (a) when the result is zero, 9 + (a) when an overflow occurs, and 19 + (a) normally.
*4: 4 when the result is zero, 7 when an overflow occurs, and 22 normally.
*5: 6 + (a) when the result is zero, 8 + (a) when an overflow occurs, and 26 + (a) normally.
*6: (b) when the result is zero or when an overflow occurs, and 2 × (b) normally.
*7: (c) when the result is zero or when an overflow occurs, and 2 × (c) normally.
*8: 3 when byte (AH) is zero, and 7 when byte (AH) is not zero.
*9: 4 when byte (ear) is zero, and 8 when byte (ear) is not zero.
*10: 5 + (a) when byte (eam) is zero, and 9 + (a) when byte (eam) is not 0.
*11: 3 when word (AH) is zero, and 11 when word (AH) is not zero.
*12: 4 when word (ear) is zero, and 12 when word (ear) is not zero.
*13: 5 + (a) when word (eam) is zero, and 13 + (a) when word (eam) is not zero.

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

Table 13 Logical 1 Instructions (Byte/Word) [39 Instructions]

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
AND	A, #imm8	2	2	0	0	byte (A) ← (A) and imm8	–	–	–	–	–	*	*	R	–	–
AND	A, ear	2	3	1	0	byte (A) ← (A) and (ear)	–	–	–	–	–	*	*	R	–	–
AND	A, eam	2+	4+ (a)	0	(b)	byte (A) ← (A) and (eam)	–	–	–	–	–	*	*	R	–	–
AND	ear, A	2	3	2	0	byte (ear) ← (ear) and (A)	–	–	–	–	–	*	*	R	–	–
AND	eam, A	2+	5+ (a)	0	2× (b)	byte (eam) ← (eam) and (A)	–	–	–	–	–	*	*	R	–	*
OR	A, #imm8	2	2	0	0	byte (A) ← (A) or imm8	–	–	–	–	–	*	*	R	–	–
OR	A, ear	2	3	1	0	byte (A) ← (A) or (ear)	–	–	–	–	–	*	*	R	–	–
OR	A, eam	2+	4+ (a)	0	(b)	byte (A) ← (A) or (eam)	–	–	–	–	–	*	*	R	–	–
OR	ear, A	2	3	2	0	byte (ear) ← (ear) or (A)	–	–	–	–	–	*	*	R	–	–
OR	eam, A	2+	5+ (a)	0	2× (b)	byte (eam) ← (eam) or (A)	–	–	–	–	–	*	*	R	–	*
XOR	A, #imm8	2	2	0	0	byte (A) ← (A) xor imm8	–	–	–	–	–	*	*	R	–	–
XOR	A, ear	2	3	1	0	byte (A) ← (A) xor (ear)	–	–	–	–	–	*	*	R	–	–
XOR	A, eam	2+	4+ (a)	0	(b)	byte (A) ← (A) xor (eam)	–	–	–	–	–	*	*	R	–	–
XOR	ear, A	2	3	2	0	byte (ear) ← (ear) xor (A)	–	–	–	–	–	*	*	R	–	–
XOR	eam, A	2+	5+ (a)	0	2× (b)	byte (eam) ← (eam) xor (A)	–	–	–	–	–	*	*	R	–	*
NOT	A	1	2	0	0	byte (A) ← not (A)	–	–	–	–	–	*	*	R	–	–
NOT	ear	2	3	2	0	byte (ear) ← not (ear)	–	–	–	–	–	*	*	R	–	–
NOT	eam	2+	5+ (a)	0	2× (b)	byte (eam) ← not (eam)	–	–	–	–	–	*	*	R	–	*
ANDW	A	1	2	0	0	word (A) ← (AH) and (A)	–	–	–	–	–	*	*	R	–	–
ANDW	A, #imm16	3	2	0	0	word (A) ← (A) and imm16	–	–	–	–	–	*	*	R	–	–
ANDW	A, ear	2	3	1	0	word (A) ← (A) and (ear)	–	–	–	–	–	*	*	R	–	–
ANDW	A, eam	2+	4+ (a)	0	(c)	word (A) ← (A) and (eam)	–	–	–	–	–	*	*	R	–	–
ANDW	ear, A	2	3	2	0	word (ear) ← (ear) and (A)	–	–	–	–	–	*	*	R	–	–
ANDW	eam, A	2+	5+ (a)	0	2× (c)	word (eam) ← (eam) and (A)	–	–	–	–	–	*	*	R	–	*
ORW	A	1	2	0	0	word (A) ← (AH) or (A)	–	–	–	–	–	*	*	R	–	–
ORW	A, #imm16	3	2	0	0	word (A) ← (A) or imm16	–	–	–	–	–	*	*	R	–	–
ORW	A, ear	2	3	1	0	word (A) ← (A) or (ear)	–	–	–	–	–	*	*	R	–	–
ORW	A, eam	2+	4+ (a)	0	(c)	word (A) ← (A) or (eam)	–	–	–	–	–	*	*	R	–	–
ORW	ear, A	2	3	2	0	word (ear) ← (ear) or (A)	–	–	–	–	–	*	*	R	–	–
ORW	eam, A	2+	5+ (a)	0	2× (c)	word (eam) ← (eam) or (A)	–	–	–	–	–	*	*	R	–	*
XORW	A	1	2	0	0	word (A) ← (AH) xor (A)	–	–	–	–	–	*	*	R	–	–
XORW	A, #imm16	3	2	0	0	word (A) ← (A) xor imm16	–	–	–	–	–	*	*	R	–	–
XORW	A, ear	2	3	1	0	word (A) ← (A) xor (ear)	–	–	–	–	–	*	*	R	–	–
XORW	A, eam	2+	4+ (a)	0	(c)	word (A) ← (A) xor (eam)	–	–	–	–	–	*	*	R	–	–
XORW	ear, A	2	3	2	0	word (ear) ← (ear) xor (A)	–	–	–	–	–	*	*	R	–	–
XORW	eam, A	2+	5+ (a)	0	2× (c)	word (eam) ← (eam) xor (A)	–	–	–	–	–	*	*	R	–	*
NOTW	A	1	2	0	0	word (A) ← not (A)	–	–	–	–	–	*	*	R	–	–
NOTW	ear	2	3	2	0	word (ear) ← not (ear)	–	–	–	–	–	*	*	R	–	–
NOTW	eam	2+	5+ (a)	0	2× (c)	word (eam) ← not (eam)	–	–	–	–	–	*	*	R	–	*

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

Table 14 Logical 2 Instructions (Long Word) [6 Instructions]

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
ANDL A, ear	2	6	2	0	long (A) ← (A) and (ear)	–	–	–	–	–	*	*	R	–	–
ANDL A, eam	2+	7+ (a)	0	(d)	long (A) ← (A) and (eam)	–	–	–	–	–	*	*	R	–	–
ORL A, ear	2	6	2	0	long (A) ← (A) or (ear)	–	–	–	–	–	*	*	R	–	–
ORL A, eam	2+	7+ (a)	0	(d)	long (A) ← (A) or (eam)	–	–	–	–	–	*	*	R	–	–
XORL A, ea	2	6	2	0	long (A) ← (A) xor (ear)	–	–	–	–	–	*	*	R	–	–
XORL A, eam	2+	7+ (a)	0	(d)	long (A) ← (A) xor (eam)	–	–	–	–	–	*	*	R	–	–

Table 15 Sign Inversion Instructions (Byte/Word) [6 Instructions]

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
NEG A	1	2	0	0	byte (A) ← 0 – (A)	X	–	–	–	–	*	*	*	*	–
NEG ear	2	3	2	0	byte (ear) ← 0 – (ear)	–	–	–	–	–	*	*	*	*	–
NEG eam	2+	5+ (a)	0	2× (b)	byte (eam) ← 0 – (eam)	–	–	–	–	–	*	*	*	*	*
NEGW A	1	2	0	0	word (A) ← 0 – (A)	–	–	–	–	–	*	*	*	*	–
NEGW ear	2	3	2	0	word (ear) ← 0 – (ear)	–	–	–	–	–	*	*	*	*	–
NEGW eam	2+	5+ (a)	0	2× (c)	word (eam) ← 0 – (eam)	–	–	–	–	–	*	*	*	*	*

Table 16 Normalize Instruction (Long Word) [1 Instruction]

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
NRML A, R0	2	*1	1	0	long (A) ← Shift until first digit is "1" byte (R0) ← Current shift count	–	–	–	–	–	–	*	–	–	–

*1: 4 when the contents of the accumulator are all zeroes, 6 + (R0) in all other cases (shift count).

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

Table 17 Shift Instructions (Byte/Word/Long Word) [18 Instructions]

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
RORC A	2	2	0	0	byte (A) ← Right rotation with carry	–	–	–	–	–	*	*	–	*	–
ROLC A	2	2	0	0	byte (A) ← Left rotation with carry	–	–	–	–	–	*	*	–	*	–
RORC ear	2	3	2	0	byte (ear) ← Right rotation with carry	–	–	–	–	–	*	*	–	*	–
RORC eam	2+	5+ (a)	0	2× (b)	byte (eam) ← Right rotation with carry	–	–	–	–	–	*	*	–	*	*
ROLC ear	2	3	2	0	byte (ear) ← Left rotation with carry	–	–	–	–	–	*	*	–	*	–
ROLC eam	2+	5+ (a)	0	2× (b)	byte (eam) ← Left rotation with carry	–	–	–	–	–	*	*	–	*	*
ASR A, R0	2	*1	1	0	byte (A) ← Arithmetic right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSR A, R0	2	*1	1	0	byte (A) ← Logical right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSL A, R0	2	*1	1	0	byte (A) ← Logical left barrel shift (A, R0)	–	–	–	–	–	*	*	–	*	–
ASRW A	1	2	0	0	word (A) ← Arithmetic right shift (A, 1 bit)	–	–	–	–	*	*	*	–	*	–
LSRW A/SHRW A	1	2	0	0	word (A) ← Logical right shift (A, 1 bit)	–	–	–	–	*	R	*	–	*	–
LSLW A/SHLW A	1	2	0	0	word (A) ← Logical left shift (A, 1 bit)	–	–	–	–	–	*	*	–	*	–
ASRW A, R0	2	*1	1	0	word (A) ← Arithmetic right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSRW A, R0	2	*1	1	0	word (A) ← Logical right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSLW A, R0	2	*1	1	0	word (A) ← Logical left barrel shift (A, R0)	–	–	–	–	–	*	*	–	*	–
ASRL A, R0	2	*2	1	0	long (A) ← Arithmetic right shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSRL A, R0	2	*2	1	0	long (A) ← Logical right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSLL A, R0	2	*2	1	0	long (A) ← Logical left barrel shift (A, R0)	–	–	–	–	–	*	*	–	*	–

*1: 6 when R0 is 0, 5 + (R0) in all other cases.
*2: 6 when R0 is 0, 6 + (R0) in all other cases.

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

Table 18 Branch 1 Instructions [31 Instructions]

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
BZ/BEQ rel	2	*1	0	0	Branch when (Z) = 1	–	–	–	–	–	–	–	–	–	–
BNZ/BNE rel	2	*1	0	0	Branch when (Z) = 0	–	–	–	–	–	–	–	–	–	–
BC/BLO rel	2	*1	0	0	Branch when (C) = 1	–	–	–	–	–	–	–	–	–	–
BNC/BHS rel	2	*1	0	0	Branch when (C) = 0	–	–	–	–	–	–	–	–	–	–
BN rel	2	*1	0	0	Branch when (N) = 1	–	–	–	–	–	–	–	–	–	–
BP rel	2	*1	0	0	Branch when (N) = 0	–	–	–	–	–	–	–	–	–	–
BV rel	2	*1	0	0	Branch when (V) = 1	–	–	–	–	–	–	–	–	–	–
BNV rel	2	*1	0	0	Branch when (V) = 0	–	–	–	–	–	–	–	–	–	–
BT rel	2	*1	0	0	Branch when (T) = 1	–	–	–	–	–	–	–	–	–	–
BNT rel	2	*1	0	0	Branch when (T) = 0	–	–	–	–	–	–	–	–	–	–
BLT rel	2	*1	0	0	Branch when (V) xor (N) = 1	–	–	–	–	–	–	–	–	–	–
BGE rel	2	*1	0	0	Branch when (V) xor (N) = 0	–	–	–	–	–	–	–	–	–	–
BLE rel	2	*1	0	0	Branch when ((V) xor (N)) or (Z) = 1	–	–	–	–	–	–	–	–	–	–
BGT rel	2	*1	0	0	Branch when ((V) xor (N)) or (Z) = 0	–	–	–	–	–	–	–	–	–	–
BLS rel	2	*1	0	0	Branch when (C) or (Z) = 1	–	–	–	–	–	–	–	–	–	–
BHI rel	2	*1	0	0	Branch when (C) or (Z) = 0	–	–	–	–	–	–	–	–	–	–
BRA rel	2	*1	0	0	Branch unconditionally	–	–	–	–	–	–	–	–	–	–
JMP @A	1	2	0	0	word (PC) ← (A)	–	–	–	–	–	–	–	–	–	–
JMP addr16	3	3	0	0	word (PC) ← addr16	–	–	–	–	–	–	–	–	–	–
JMP @ear	2	3	1	0	word (PC) ← (ear)	–	–	–	–	–	–	–	–	–	–
JMP @eam	2+	4+ (a)	0	(c)	word (PC) ← (eam)	–	–	–	–	–	–	–	–	–	–
JMPP @ear *3	2	5	2	0	word (PC) ← (ear), (PCB) ← (ear +2)	–	–	–	–	–	–	–	–	–	–
JMPP @eam *3	2+	6+ (a)	0	(d)	word (PC) ← (eam), (PCB) ← (eam +2)	–	–	–	–	–	–	–	–	–	–
JMPP addr24	4	4	0	0	word (PC) ← ad24 0 to 15, (PCB) ← ad24 16 to 23	–	–	–	–	–	–	–	–	–	–
CALL @ear *4	2	6	1	(c)	word (PC) ← (ear)	–	–	–	–	–	–	–	–	–	–
CALL @eam *4	2+	7+ (a)	0	2× (c)	word (PC) ← (eam)	–	–	–	–	–	–	–	–	–	–
CALL addr16 *5	3	6	0	(c)	word (PC) ← addr16	–	–	–	–	–	–	–	–	–	–
CALLV #vct4 *5	1	7	0	2× (c)	Vector call instruction	–	–	–	–	–	–	–	–	–	–
CALLP @ear *6	2	10	2	2× (c)	word (PC) ← (ear) 0 to 15 (PCB) ← (ear) 16 to 23	–	–	–	–	–	–	–	–	–	–
CALLP @eam *6	2+	11+ (a)	0	*2	word (PC) ← (eam) 0 to 15 (PCB) ← (eam) 16 to 23	–	–	–	–	–	–	–	–	–	–
CALLP addr24 *7	4	10	0	2× (c)	word (PC) ← addr0 to 15, (PCB) ← addr16 to 23	–	–	–	–	–	–	–	–	–	–

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

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

Table 19 Branch 2 Instructions [19 Instructions]

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
CBNE A, #imm8, rel	3	*1	0	0	Branch when byte (A) ≠ imm8	–	–	–	–	–	*	*	*	*	–
CWBNE A, #imm16, rel	4	*1	0	0	Branch when word (A) ≠ imm16	–	–	–	–	–	*	*	*	*	–
CBNE ear, #imm8, rel	4	*2	1	0	Branch when byte (ear) ≠ imm8	–	–	–	–	–	*	*	*	*	–
CBNE eam, #imm8, rel*9	4+	*3	0	(b)	Branch when byte (eam) ≠ imm8	–	–	–	–	–	*	*	*	*	–
CWBNE ear, #imm16, rel	5	*4	1	0	Branch when word (ear) ≠ imm16	–	–	–	–	–	*	*	*	*	–
CWBNE eam, #imm16, rel*9	5+	*3	0	(c)	Branch when word (eam) ≠ imm16	–	–	–	–	–	*	*	*	*	–
DBNZ ear, rel	3	*5	2	0	Branch when byte (ear) = (ear) – 1, and (ear) ≠ 0	–	–	–	–	–	*	*	*	–	–
DBNZ eam, rel	3+	*6	2	2× (b)	Branch when byte (eam) = (eam) – 1, and (eam) ≠ 0	–	–	–	–	–	*	*	*	–	*
DWBNZ ear, rel	3	*5	2	0	Branch when word (ear) = (ear) – 1, and (ear) ≠ 0	–	–	–	–	–	*	*	*	–	–
DWBNZ eam, rel	3+	*6	2	2× (c)	Branch when word (eam) = (eam) – 1, and (eam) ≠ 0	–	–	–	–	–	*	*	*	–	*
INT #vct8	2	20	0	8× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
INT addr16	3	16	0	6× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
INTP addr24	4	17	0	6× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
INT9	1	20	0	8× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
RETI	1	15	0	6× (c)	Return from interrupt	–	–	*	*	*	*	*	*	*	–
LINK #local8	2	6	0	(c)	At constant entry, save old frame pointer to stack, set new frame pointer, and allocate local pointer area	–	–	–	–	–	–	–	–	–	–
UNLINK	1	5	0	(c)	At constant entry, retrieve old frame pointer from stack.	–	–	–	–	–	–	–	–	–	–
RET *7	1	4	0	(c)	Return from subroutine	–	–	–	–	–	–	–	–	–	–
RETP *8	1	6	0	(d)	Return from subroutine	–	–	–	–	–	–	–	–	–	–

*1: 5 when branching, 4 when not branching
*2: 13 when branching, 12 when not branching
*3: 7 + (a) when branching, 6 + (a) when not branching
*4: 8 when branching, 7 when not branching
*5: 7 when branching, 6 when not branching
*6: 8 + (a) when branching, 7 + (a) when not branching
*7: Retrieve (word) from stack
*8: Retrieve (long word) from stack
*9: In the CBNE/CWBNE instruction, do not use the RWj+ addressing mode.

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

Table 20 Other Control Instructions (Byte/Word/Long Word) [36 Instructions]

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
PUSHW A	1	4	0	(c)	word (SP) ← (SP) –2, ((SP)) ← (A)	–	–	–	–	–	–	–	–	–	–
PUSHW AH	1	4	0	(c)	word (SP) ← (SP) –2, ((SP)) ← (AH)	–	–	–	–	–	–	–	–	–	–
PUSHW PS	1	4	0	(c)	word (SP) ← (SP) –2, ((SP)) ← (PS)	–	–	–	–	–	–	–	–	–	–
PUSHW rlst	2	*3	*5	*4	(SP) ← (SP) –2n, ((SP)) ← (rlst)	–	–	–	–	–	–	–	–	–	–
POPW A	1	3	0	(c)	word (A) ← ((SP)), (SP) ← (SP) +2	–	*	–	–	–	–	–	–	–	–
POPW AH	1	3	0	(c)	word (AH) ← ((SP)), (SP) ← (SP) +2	–	–	–	–	–	–	–	–	–	–
POPW PS	1	4	0	(c)	word (PS) ← ((SP)), (SP) ← (SP) +2	–	–	*	*	*	*	*	*	*	–
POPW rlst	2	*2	*5	*4	(rlst) ← ((SP)), (SP) ← (SP) +2n	–	–	–	–	–	–	–	–	–	–
JCTX @A	1	14	0	6× (c)	Context switch instruction	–	–	*	*	*	*	*	*	*	–
AND CCR, #imm8	2	3	0	0	byte (CCR) ← (CCR) and imm8	–	–	*	*	*	*	*	*	*	–
OR CCR, #imm8	2	3	0	0	byte (CCR) ← (CCR) or imm8	–	–	*	*	*	*	*	*	*	–
MOV RP, #imm8	2	2	0	0	byte (RP) ←imm8	–	–	–	–	–	–	–	–	–	–
MOV ILM, #imm8	2	2	0	0	byte (ILM) ←imm8	–	–	–	–	–	–	–	–	–	–
MOVEA RWi, ear	2	3	1	0	word (RWi) ←ear	–	–	–	–	–	–	–	–	–	–
MOVEA RWi, eam	2+	2+ (a)	1	0	word (RWi) ←eam	–	–	–	–	–	–	–	–	–	–
MOVEA A, ear	2	1	0	0	word(A) ←ear	–	*	–	–	–	–	–	–	–	–
MOVEA A, eam	2+	1+ (a)	0	0	word (A) ←eam	–	*	–	–	–	–	–	–	–	–
ADDSP #imm8	2	3	0	0	word (SP) ← (SP) +ext (imm8)	–	–	–	–	–	–	–	–	–	–
ADDSP #imm16	3	3	0	0	word (SP) ← (SP) +imm16	–	–	–	–	–	–	–	–	–	–
MOV A, brgl	2	*1	0	0	byte (A) ← (brgl)	Z	*	–	–	–	*	*	–	–	–
MOV brg2, A	2	1	0	0	byte (brg2) ← (A)	–	–	–	–	–	*	*	–	–	–
NOP	1	1	0	0	No operation	–	–	–	–	–	–	–	–	–	–
ADB	1	1	0	0	Prefix code for accessing AD space	–	–	–	–	–	–	–	–	–	–
DTB	1	1	0	0	Prefix code for accessing DT space	–	–	–	–	–	–	–	–	–	–
PCB	1	1	0	0	Prefix code for accessing PC space	–	–	–	–	–	–	–	–	–	–
SPB	1	1	0	0	Prefix code for accessing SP space	–	–	–	–	–	–	–	–	–	–
NCC	1	1	0	0	Prefix code for no flag change	–	–	–	–	–	–	–	–	–	–
CMR	1	1	0	0	Prefix code for common register bank	–	–	–	–	–	–	–	–	–	–

*1: PCB, ADB, SSB, USB, and SPB : 1 state
DTB, DPR : 2 states

*2: 7 + 3 × (pop count) + 2 × (last register number to be popped), 7 when rlst = 0 (no transfer register)

*3: 29 + (push count) – 3 × (last register number to be pushed), 8 when rlst = 0 (no transfer register)

*4: Pop count × (c), or push count × (c)

*5: Pop count or push count.

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

Table 21 Bit Manipulation Instructions [21 Instructions]

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVB	A, dir:bp	3	5	0	(b)	byte (A) ← (dir:bp) b	Z	*	–	–	–	*	*	–	–	–
MOVB	A, addr16:bp	4	5	0	(b)	byte (A) ← (addr16:bp) b	Z	*	–	–	–	*	*	–	–	–
MOVB	A, io:bp	3	4	0	(b)	byte (A) ← (io:bp) b	Z	*	–	–	–	*	*	–	–	–
MOVB	dir:bp, A	3	7	0	2× (b)	bit (dir:bp) b ← (A)	–	–	–	–	–	*	*	–	–	*
MOVB	addr16:bp, A	4	7	0	2× (b)	bit (addr16:bp) b ← (A)	–	–	–	–	–	*	*	–	–	*
MOVB	io:bp, A	3	6	0	2× (b)	bit (io:bp) b ← (A)	–	–	–	–	–	*	*	–	–	*
SETB	dir:bp	3	7	0	2× (b)	bit (dir:bp) b ← 1	–	–	–	–	–	–	–	–	–	*
SETB	addr16:bp	4	7	0	2× (b)	bit (addr16:bp) b ← 1	–	–	–	–	–	–	–	–	–	*
SETB	io:bp	3	7	0	2× (b)	bit (io:bp) b ← 1	–	–	–	–	–	–	–	–	–	*
CLRB	dir:bp	3	7	0	2× (b)	bit (dir:bp) b ← 0	–	–	–	–	–	–	–	–	–	*
CLRB	addr16:bp	4	7	0	2× (b)	bit (addr16:bp) b ← 0	–	–	–	–	–	–	–	–	–	*
CLRB	io:bp	3	7	0	2× (b)	bit (io:bp) b ← 0	–	–	–	–	–	–	–	–	–	*
BBC	dir:bp, rel	4	*1	0	(b)	Branch when (dir:bp) b = 0	–	–	–	–	–	–	*	–	–	–
BBC	addr16:bp, rel	5	*1	0	(b)	Branch when (addr16:bp) b = 0	–	–	–	–	–	–	*	–	–	–
BBC	io:bp, rel	4	*2	0	(b)	Branch when (io:bp) b = 0	–	–	–	–	–	–	*	–	–	–
BBS	dir:bp, rel	4	*1	0	(b)	Branch when (dir:bp) b = 1	–	–	–	–	–	–	*	–	–	–
BBS	addr16:bp, rel	5	*1	0	(b)	Branch when (addr16:bp) b = 1	–	–	–	–	–	–	*	–	–	–
BBS	io:bp, rel	4	*2	0	(b)	Branch when (io:bp) b = 1	–	–	–	–	–	–	*	–	–	–
SBBS	addr16:bp, rel	5	*3	0	2× (b)	Branch when (addr16:bp) b = 1, bit = 1	–	–	–	–	–	–	*	–	–	*
WBTS	io:bp	3	*4	0	*5	Wait until (io:bp) b = 1	–	–	–	–	–	–	–	–	–	–
WBTC	io:bp	3	*4	0	*5	Wait until (io:bp) b = 0	–	–	–	–	–	–	–	–	–	–

*1: 8 when branching, 7 when not branching
*2: 7 when branching, 6 when not branching
*3: 10 when condition is satisfied, 9 when not satisfied
*4: Undefined count
*5: Until condition is satisfied

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

Table 22 Accumulator Manipulation Instructions (Byte/Word) [6 Instructions]

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
SWAP	1	3	0	0	byte (A) 0 to 7 ↔ (A) 8 to 15	–	–	–	–	–	–	–	–	–	–
SWAPW/XCHW AL, AH	1	2	0	0	word (AH) ↔ (AL)	–	*	–	–	–	–	–	–	–	–
EXT	1	1	0	0	byte sign extension	X	–	–	–	–	*	*	–	–	–
EXTW	1	2	0	0	word sign extension	–	X	–	–	–	*	*	–	–	–
ZEXT	1	1	0	0	byte zero extension	Z	–	–	–	–	R	*	–	–	–
ZEXTW	1	1	0	0	word zero extension	–	Z	–	–	–	R	*	–	–	–

Table 23 String Instructions [10 Instructions]

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

m: RW0 value (counter value)
n: Loop count

*1: 5 when RW0 is 0, 4 + 7 × (RW0) for count out, and 7 × n + 5 when match occurs
*2: 5 when RW0 is 0, 4 + 8 × (RW0) in any other case
*3: (b) × (RW0) + (b) × (RW0) when accessing different areas for the source and destination, calculate (b) separately for each.
*4: (b) × n
*5: 2 × (RW0)
*6: (c) × (RW0) + (c) × (RW0) when accessing different areas for the source and destination, calculate (c) separately for each.
*7: (c) × n
*8: 2 × (RW0)

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

■ MASK OPTION LIST

<table>
<tr><th rowspan="2">No.</th><th>Part number</th><th colspan="2">MB60662A
MB90663A</th><th colspan="2">MB90P663A</th></tr>
<tr><th>Specifying procedure</th><th colspan="2">Specify when ordering masking</th><th colspan="2">Set with EPROM programmer</th></tr>
<tr><td>1</td><td>P00 to P07
P10 to P17
P20 to P27
P30 to P33
P40 to P47
P60 to P66
$\overline{RST}$
DTTI</td><td colspan="2">Pull-up resistor can be selected for each pin</td><td colspan="2">Pull-up resistor can be selected for each pin</td></tr>
<tr><td rowspan="3">2</td><td>MD2</td><td>Pull-down resistor</td><td rowspan="3">Can be selected all at once</td><td colspan="2">Cannot be selected; pull-down resistor not provided</td></tr>
<tr><td>MD1</td><td>Pull-up resistor</td><td>Pull-up resistor</td><td rowspan="2">Can be selected all at once</td></tr>
<tr><td>MD0</td><td>Pull-up resistor</td><td>Pull-up resistor</td></tr>
<tr><td>3</td><td>Accept asynchronous reset input

Accepted
Not accepted</td><td colspan="2">Can be selected</td><td colspan="2">Can be selected</td></tr>
</table>

Notes: • A specification of “yes” for accept asynchronous reset input refers to a function whereby reset input is accepted when oscillation for output ports (including peripheral resource output) is stopped and port output (including peripheral resource output) is forced Hi-z. Note, however, that since internal reset (reset of the CPU and peripheral resources) is synchronized with the clock, the CPU and peripheral resources are not initialized when the clock is stopped.

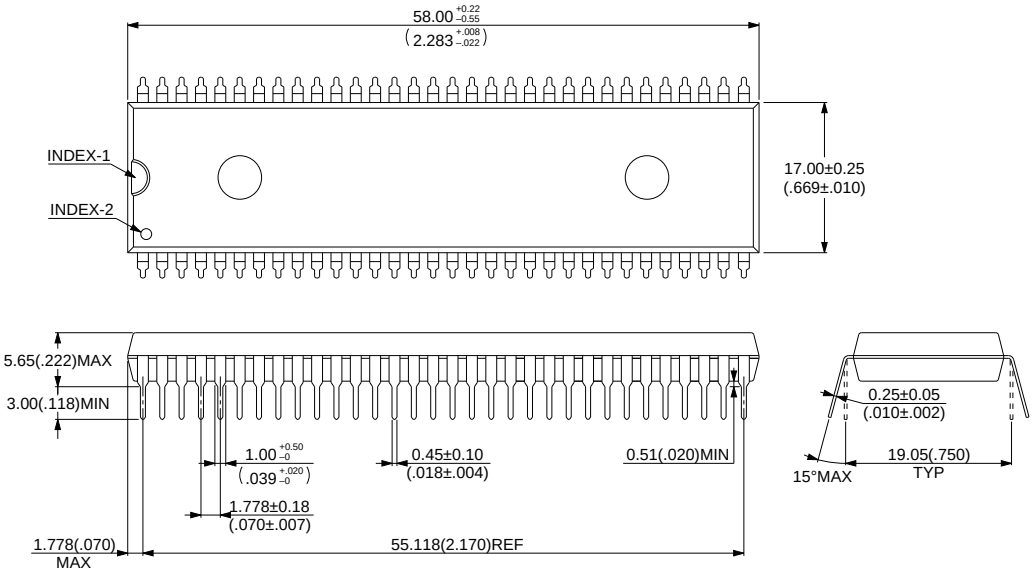
- For details on writing to the MB90P663A, see Chapter 6, “■ PROGRAMMING THE MB90P663A EPROM”.
- Use of a pull-up/pull-down resistors for the mode pins (MD2 to MD0) can be selected separately for each pin. If “yes” is selected, a pull-up is attached to MD0 and MD1 and a pull-down to MD2 for mask ROM versions. A pull-up is attached to MD0 and MD1, but a pull-down is not attached to MD2 for OTP versions.
- Since it takes eight machine cycles to make option settings for the MB90P663A, options cannot be set between when power is first turned on and the clock is supplied. (This results in a setting of no pull-up for all pins and accept asynchronous reset input.)

■ ORDERING INFORMATION

Part number	Package	Remarks
MB90662AP-SH MB90663AP-SH MB90P663AP-SH	64-pin plastic SH-DIP (DIP-64P-M01)	
MB90662APFM MB90663APFM MB90P663APFM	64-pin plastic LQFP (FTP-64P-M09)	

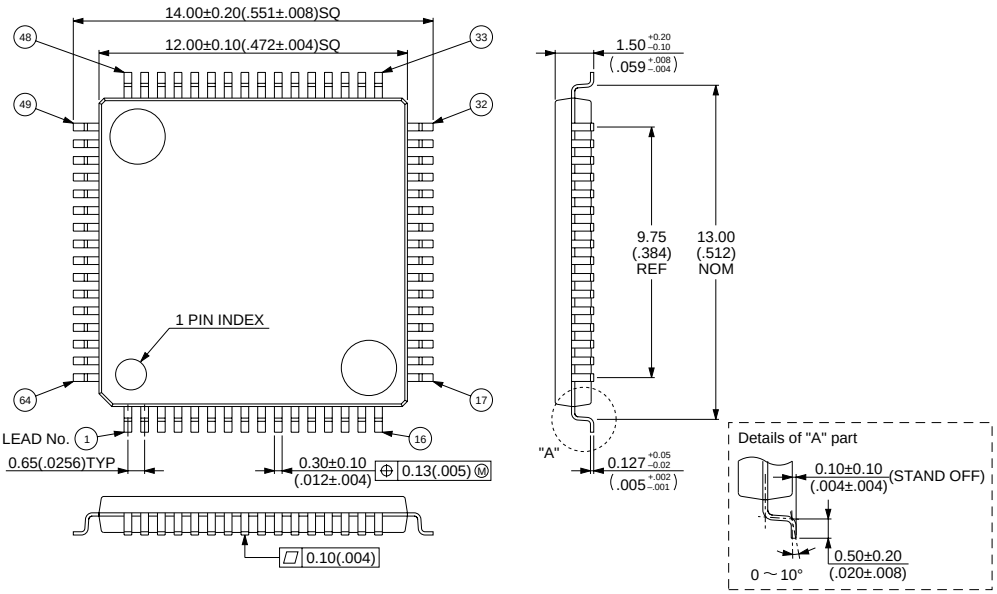
■ PACKAGE DIMENSIONS

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



Dimensions in mm (inches)

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



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 211-88, Japan
Tel: (044) 754-3763
Fax: (044) 754-3329

North and South America
FUJITSU MICROELECTRONICS, INC.
Semiconductor Division
3545 North First Street
San Jose, CA 95134-1804, U.S.A.
Tel: (408) 922-9000
Fax: (408) 432-9044/9045

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

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

F9703

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 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 inherently a certain rate of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.

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