FUJITSU SEMICONDUCTOR
DATA SHEET

DS07-13606-1E

16-bit Proprietary Microcontroller

CMOS

F^{2}MC-16L MB90620A Series

MB90622A/623A/P623A

■ DESCRIPTION

The MB90620A series is a line of general-purpose, 16-bit microcontrollers designed for those applications which require high-speed real-time processing, proving to be suitable for various industrial machines, camera and video devices, OA equipment, and for process control. The CPU used in this series is the F^{2}MC*-16L. The instruction set for the F^{2}MC-16L CPU core is designed to be optimized for controller applications while inheriting the AT architecture of the F^{2}MC-16/16H series, allowing a wide range of control tasks to be processed efficiently at high speed.

The peripheral resources integrated in the MB90620A series include: the UART (clock asynchronous/ synchronous transfer) × 1 channel, the extended serial I/O interface × 1 channel, the A/D converter (8/10-bit precision) × 4 channels, the 16-bit PPG timer (PWM/single-shot function) × 2 channels, the 16-bit reload timer × 3 channels, the 16-bit free-run timer (built-in compare register: 2 channels) × 2 channels, the external interrupt × 8 channels, the watch timer × 1 channel, LCD controller/driver 32 segments × 4 commons.

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

■ FEATURES

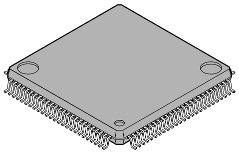
F^{2}MC-16L CPU

- Minimum execution time: 83.33 ns (at machine clock frequency of 12 MHz)
- Dual-clock control systems
- PLL clock control

(Continued)

■ PACKAGE

100-pin Plastic LQFP



(FPT-100P-M05)

(Continued)

- Instruction set optimized for controller applications
  Variety of data types: bit, byte, word, long-word
  Expanded addressing modes: 23 types
  High coding efficiency
  Improvement of high-precision arithmetic operations through use of 32-bit accumulator
- Instruction set supports high-level language (C language) and multitasking
  Inclusion of system stack pointer
  Enhanced pointer-indirect instructions
  Barrel shift instruction
- Improved execution speed: 4-byte instruction queue
- 8-level, 32-factor powerful interrupt service functions
- Automatic transfer function independent of CPU (EI^{2}OS)
- General-purpose ports: max. 59 channels
- 18-bit timebase timer/15-bit watch timer
- Watchdog timer function
- CPU intermittent operation function
- Various standby modes

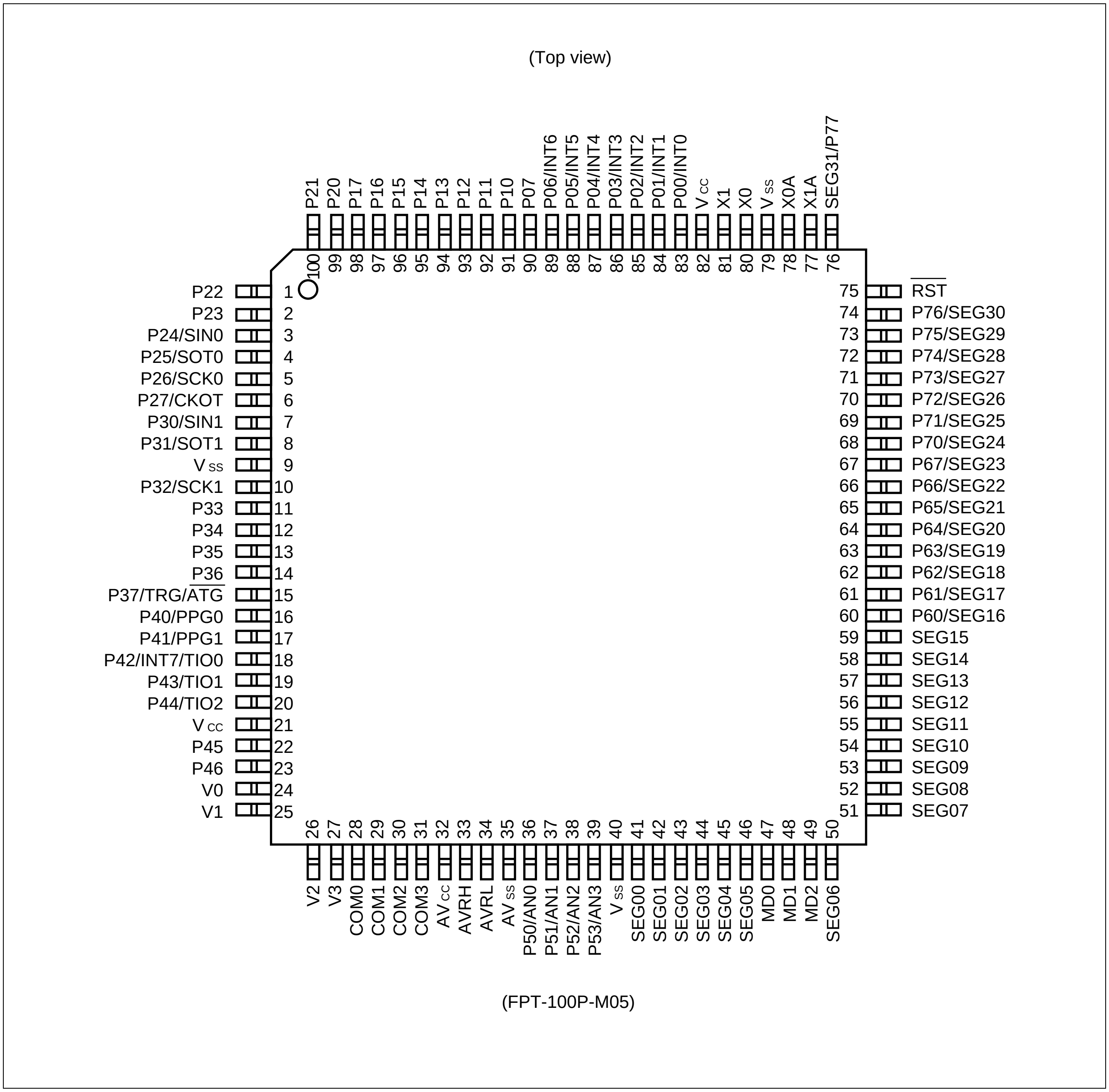
Peripheral blocks

- ROM: 32 Kbytes (MB90622A)
  48 Kbytes (MB90623A)
- One-time PROM: 48 Kbytes (MB90P623A)
- RAM: 1.64 Kbytes (MB90622A)
  2 Kbytes (MB90623A/P623A)
- General-purpose ports: max. 59 channels
- Dual-clock control system
- PLL clock multiplication control system
- UART: 1 channel
  Can be used for either asynchronous transfer or synchronous transfer with clock
- Extended serial I/O interface: 1 channel
  Can be used for 8-bit synchronous transfer
- A/D converter (8/10-bit resolution): 4 channels
- PPG (Programable pulse generator): 2 channels
- 16-bit reload timer: 3 channels
- 16-bit free-run timer: 2 channels
  With compare register 2 channels
- LCD controller/driver
  32 segments, 4 commons
- External interrupts: 8 channels
- 18-bit timebase timer
- 15-bit watch timer
- Watchdog timer function
- CPU intermittent operation function
- Standby mode
  Watch mode
  Sleep mode
  Stop mode

■ PRODUCT LINEUP

Part number Parameter	MB90622A	MB90623A	MB90P623A
Classification	Mass production products (Mask ROM products)		One-time model
ROM size	32 Kbytes	48 Kbytes	48 Kbytes
RAM size	1.64 Kbytes	2 Kbytes	2 Kbytes
CPU functions	Number of instructions: 340 Instruction bit length: 8 or 16 bits Instruction length: 1 to 7 bytes Data bit length: 1, 4, 8, 16, or 32 bits Minimum execution time: 83.33 ns at 12 MHz (internal)		
Oscillation circuit	Dual-clock system of main clock and sub clock		
Ports	Max. 59 channels (I/O ports (CMOS): 17 I/O ports (CMOS) with pull-up resistor available: 24 I/O ports (open drain): 18)		
UART	Number of channels: 1 Clock synchronous communication (1202 to 9615 bps, full-duplex double buffering) Clock asynchronous communication (62.5 K to 1 M bps, full-duplex double buffering) Supports multiprocessor mode		
Serial	Number of channels: 1 Internal or external clock mode Clock synchronous transfer (62.5 kHz to 1 MHz, “LSB first” or “MSB first” transfer)		
A/D converter	Resolution: 10 or 8 bits, Number of input channels: 4 Single-conversion mode (conversion for a specified input channel) Scan conversion mode (continuous conversion for specified consecutive channels) Continuous conversion mode (repeated conversion for a specified channel) Stop conversion mode (periodical conversion)		
Timer	Number of channels: 3 16-bit reload timer operation (operation clock: SUB/2, $\phi/2^3$, $\phi/2^5$, external)		
Free-run timer	Number of channels: 2 16-bit up-counter (four types of count clocks) 2 channels on each timer of the compare register (compare matching interrupt available)		
PPG timer	Number of channels: 2 PWM function, single-shot function With external trigger function		
LCD controller /driver	Common output: 4 channels, Segment output: 32 channels Direct driving of the LCD module 16 bytes of data memory for display Operation clock source (main clock/sub clock selective)		
Standby modes	Stop mode, sleep mode, and watch mode		
PLL functions	Main clock multiplication (×1, ×2, ×3 and ×4)		
Package	FPT-100P-M05		

■ PIN ASSIGNMENT



■ PIN DESCRIPTION

Pin no.	Pin name	Circuit type	Function
77 78	X1A X0A	A (Oscillation)	Crystal oscillator pins (32 kHz)
79	V_{SS}	Power supply	Digital circuit power supply (GND) pin
80 81	X0 X1	A (Oscillation)	Crystal/FAR oscillator pins (4 MHz)
82	V_{CC}	Power supply	Digital circuit power supply pin
83 to 89	P00 to P06	M (CMOS/H)	General-purpose I/O ports At this pin, a pull-up resistor is added in the input mode depending on the settings of the pull-up resistor setting register.
	INT0 to INT6		External interrupt request input pins When external interrupts are enabled, these inputs may be used at any time; therefore, it is necessary to stop output by other functions on these pins, except when using them for output deliberately.
90	P07	G (CMOS)	General-purpose I/O port At this pin, a pull-up resistor is added in the input mode depending on the settings of the pull-up resistor setting register.
91 to 98	P10 to P17	G (CMOS)	General-purpose I/O ports At this pin, a pull-up resistor is added in the input mode depending on the settings of the pull-up resistor setting register.
99, 100 1, 2	P20 to P23	G (CMOS)	General-purpose I/O ports At this pin, a pull-up resistor is added in the input mode depending on the settings of the pull-up resistor setting register.
3	P24	F (CMOS/H)	General-purpose I/O port At this pin, a pull-up resistor is added in the input mode depending on the settings of the pull-up resistor setting register.
	SIN0		UART serial data input pin During UART input operations, these inputs may be used at any time; therefore, it is necessary to stop output by other functions on these pins, except when using them for output deliberately.
4	P25	G (CMOS)	General-purpose I/O port At this pin, a pull-up resistor is added in the input mode depending on the settings of the pull-up resistor setting register.
	SOT0		UART serial data output pin This function is available when the UART serial data output is enabled.
5	P26	F (CMOS/H)	General-purpose I/O port At this pin, a pull-up resistor is added in the input mode depending on the settings of the pull-up resistor setting register.
	SCK0		UART serial data I/O pin This function is available when the UART clock output is enabled. During UART input operations, these inputs may be used at any time; therefore, it is necessary to stop output by other functions on these pins, except when using them for output deliberately.

(Continued)

Pin no.	Pin name	Circuit type	Function
6	P27	G (CMOS)	General-purpose I/O port At this pin, a pull-up resistor is added in the input mode depending on the settings of the pull-up resistor setting register.
	CKOT		Clock output pin This function is available when clock output is enabled.
7	P30	E (CMOS/H)	General-purpose I/O port
	SIN1		I/O extended serial data input pin This pin, as required, is used for input during input operation, and it is necessary to disable output for other functions from this pin unless such output is made intentionally.
8	P31	D (CMOS)	General-purpose I/O port
	SOT1		I/O extended serial data output pin This function is available when serial data data output is enabled.
9	V_{SS}	Power supply	Digital circuit power supply (GND) pin
10	P32	E (CMOS/H)	General-purpose I/O port
	SCK1		I/O extended serial clock I/O pins This function is available when clock input is enabled. This pin, as required, is used for input during input operation, and it is necessary to disable output for other functions from this pin unless such output is made intentionally.
11 to 14	P33 to P36	D (CMOS)	General-purpose I/O ports
15	P37	E (CMOS/H)	General-purpose I/O port
	TRG		PPG0 and PPG1 external trigger input pin
	$\overline{\text{ATG}}$		A/D converter trigger input pin During A/D converter input operations, these inputs may be used at any time; therefore, it is necessary to stop output by other functions on these pins, except when using them for output deliberately.
16	P40	D (CMOS)	General-purpose I/O port This function is available when PPG timer 0 output is disabled.
	PPG0		PPG timer 0 output pin This function is available when the PPG timer 0 waveform output is enabled.
17	P41	D (CMOS)	General-purpose I/O port This function is available when PPG timer 1 output is disabled.
	PPG1		PPG timer 1 output pin This function is available when the PPG timer 1 waveform output is enabled.

(Continued)

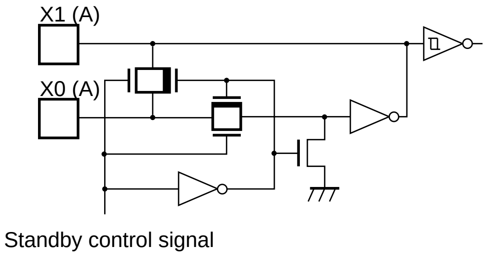
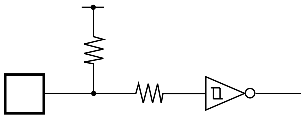
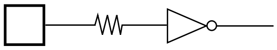
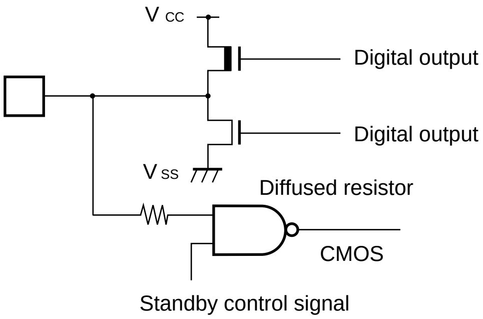
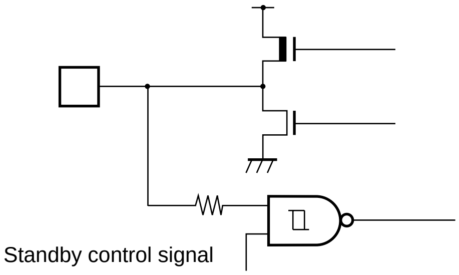
Pin no.	Pin name	Circuit type	Function
18	P42	L (CMOS/H)	General-purpose I/O port This function is available when the timer output from timer 0 is disabled.
	INT7		External interrupt request input pin When external interrupts are enabled, these inputs may be used at any time; therefore, it is necessary to stop output by other functions on these pins, except when using them for output deliberately.
	TIO0		Timer input pin The data on this pin is used as event count signal for timer 0. Timer output pin This function is available when the timer output from timer 0 is enabled.
19	P43	E (CMOS/H)	General-purpose I/O port This function is available when the timer output from timer 1 is disabled.
	TIO1		Timer input pin The data on this pin is used as event count signal for timer 1. Timer output pin This function is available when the timer output from timer 1 is enabled.
20	P44	E (CMOS/H)	General-purpose I/O port This function is available when the timer output from timer 2 is disabled.
	TIO2		Timer input pin The data on this pin is used as event count signal for timer 2. Timer output pin This function is available when the timer output from timer 2 is enabled.
21	V_{CC}	Power supply	Digital circuit power supply pin
22, 23	P45, P46	H (CMOS)	Open-drain I/O ports
24 to 27	V0 to V3	Power supply	LCDC reference power supply pins
28 to 31	COM0 to COM3	K	LCDC common pins
32	AV_{CC}	Power supply	Analog circuit power supply pin This power supply must only be turned on or off when electric potential of AV_{CC} or greater is applied to V_{CC}.
33	AVRH	Power supply	Analog circuit reference voltage input pin This pin must only be turned on or off when electric potential of AVRH or greater is applied to AV_{CC}.
34	AVRL	Power supply	Analog circuit reference voltage input pin
35	AV_{SS}	Power supply	Analog circuit power supply (GND) pin

(Continued)

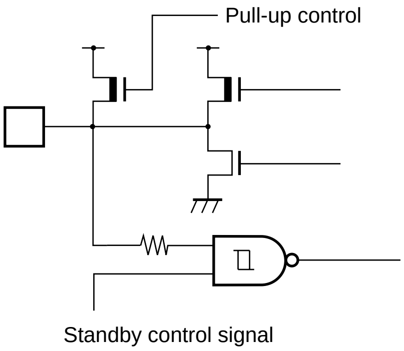
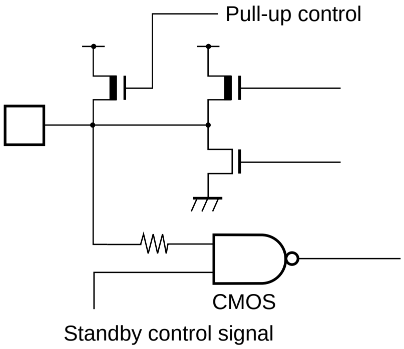
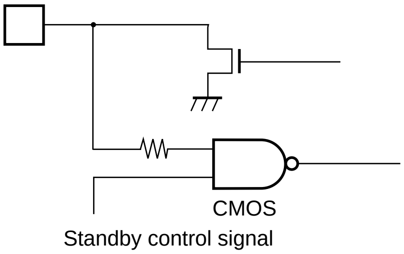
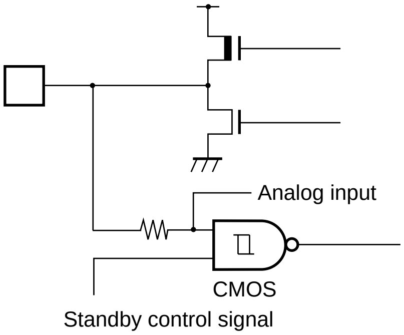
(Continued)

Pin no.	Pin name	Circuit type	Function
36 to 39	P50 to P53	I (AD)	General-purpose I/O ports This function is available when “port” is specified in the analog input enable register.
	AN0 to AN3		A/D converter analog input pins This function is available when the analog input enable register specification is “AD.”
40	V_{SS}	Power supply	Digital circuit power supply (GND) pin
41 to 46	SEG00 to SEG05	K	LCDC segment-only pins
47 to 49	MD0 to MD2	C (CMOS)	Operating mode selection input pins Connect directly to V_{CC} or V_{SS}.
50 to 59	SEG06 to SEG15	K	LCDC segment-only pins
60 to 67	P60 to P67	J	Open-drain I/O ports This is available when enabled by the LCR2.
	SEG16 to SEG23		LCDC segment pins
68 to 74	P70 to P76	J	Open-drain I/O ports This is available when enabled by the LCR2.
	SEG24 to SEG30		LCDC segment pins
75	$\overline{RST}$	B (CMOS/H)	External reset request input pin
76	P77	J	Open-drain I/O port This is available when enabled by the LCR2.
	SEG31		LCDC segment pin

■ I/O CIRCUIT TYPE

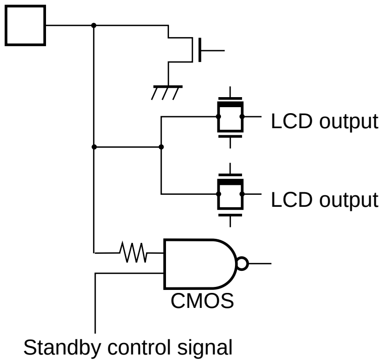
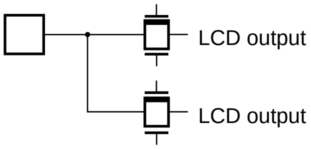
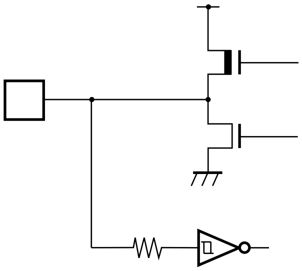
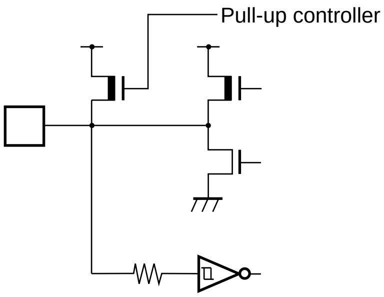
Type	Circuit	Remarks
A		• Oscillation feedback resistor: Approximately 1 MΩ
B		• Hysteresis input with pull-up resistor
C		• CMOS input port
D	 V CC Digital output Digital output V SS Diffused resistor CMOS Standby control signal	• CMOS level input/output
E		• CMOS level output • Hysteresis input

(Continued)

Type	Circuit	Remarks
F		• With input pull-up resistor control • CMOS level output • Hysteresis input
G		• With input pull-up resistor control • CMOS level input/output
H		• Open-drain type input/output
I		• CMOS level input/output • Analog input

(Continued)

(Continued)

Type	Circuit	Remarks
J		• Open-drain type output • CMOS level input • Combined with the LCD output
K		• LCD output pin
L		• CMOS level output • Hysteresis input
M		• With input pull-up resistor control • CMOS level output • Hysteresis input

■ HANDLING DEVICES

1. Preventing Latchup

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

2. Treatment of Unused Pins

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

3. External Reset Input

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

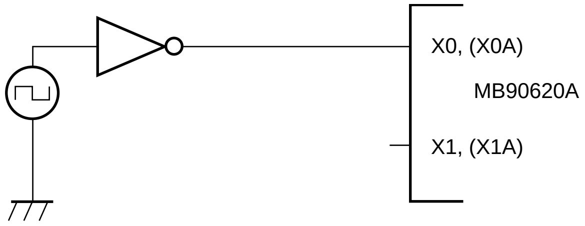
4. V_{CC} and V_{SS} Pins

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

5. Precautions when Using an External Clock

When an external clock is used, drive X0 pin.

- **Using of External Clock**



6. Sequence for Applying A/D Converter Power Supply and Analog Inputs

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

In addition, when the power is turned off, turn off the A/D converter power supply (AV_{CC}, AVRH, and AVRL) and the analog inputs (AN0 to AN15) first, and then turn off the digital power supply (AV_{CC}).

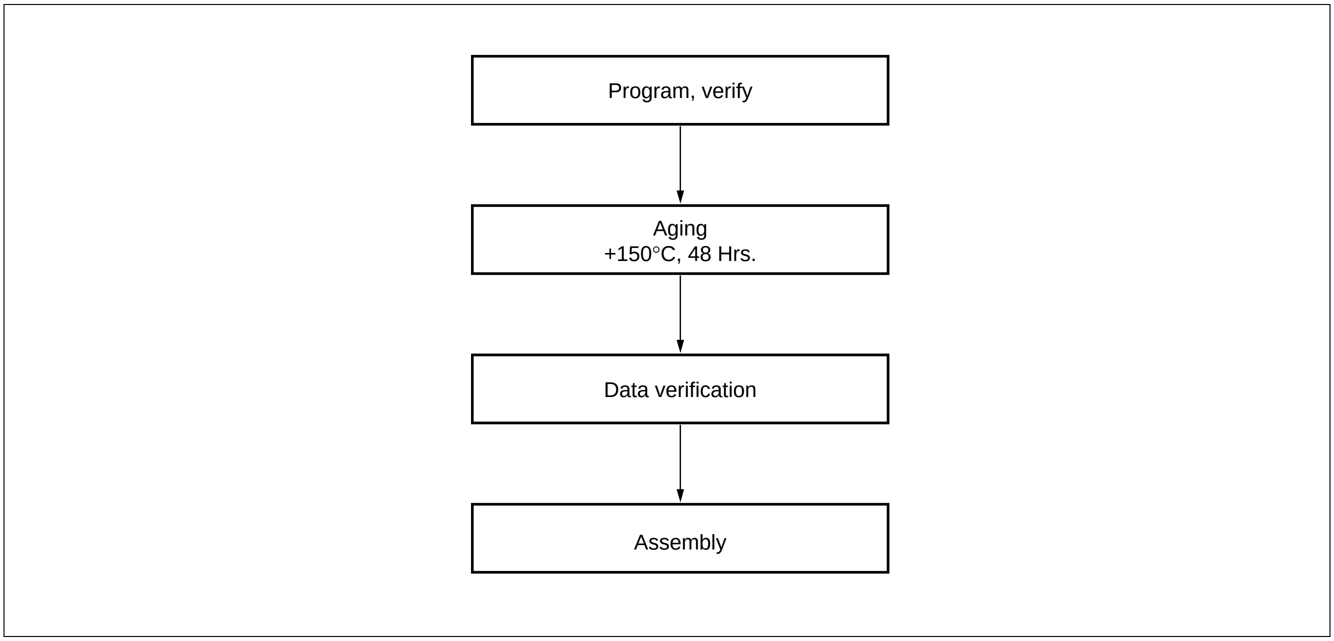
Whether applying or cutting off the power, be certain that AVRH does not exceed AV_{CC}.

7. Program Mode

In the MB90P623, all of the bits (48 K × 8 bits) are set to “1” when the IC is shipped from Fujitsu and after erasure. To input data, program the IC by selectively setting the desired bits to “0”. Bits cannot be set to “1” electrically.

8. Recommended Screening Conditions

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



9. Programming Yield

All bit 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.

■ PROGRAMMING TO THE EPROM ON THE MB90P623A

In EPROM mode, the MB90P623 EPROM functions equivalent to the MBM27C1000. This allows the PROM to be programmed with a general-purpose EPROM programmer by using the dedicated socket adapter.

1. EPROM Mode Pin Assignments

- **MBM27C1000 compatible pins**

MBM27C1000		MB90P623A	
Pin no.	**Pin name**	**Pin no.**	**Pin name**
1	V_{PP}	49	MD2 (V_{PP})
2	OE*	10	P32
3	A15	98	P17
4	A12	95	P14
5	A07	6	P27
6	A06	5	P26
7	A05	4	P25
8	A04	3	P24
9	A03	2	P23
10	A02	1	P22
11	A01	100	P21
12	A00	99	P20
13	D00	83	P00
14	D01	84	P01
15	D02	85	P02
16	GND*	—	—

MBM27C1000		MB90P623A	
Pin no.	**Pin name**	**Pin no.**	**Pin name**
32	V_{CC}	—	—
31	PGM	11	P33
30	N.C.	—	—
29	A14	97	P16
28	A13	96	P15
27	A08	91	P10
26	A09	92	P11
25	A11	94	P13
24	A16	7	P30
23	A10	93	P12
22	CE	8	P31
21	A07	90	P07
20	D06	89	P06
19	D05	88	P05
18	D04	87	P04
17	D03	86	P03

* : Connect a capacitance of 20 pF across OE (pin no.2) and GND (pin no.16) pins of the MBM27C1000.

- **Power supply, GND connection pins**

Classification	Pin no.	Pin name
Power supply	21 82	V_{CC} V_{CC}
GND	9 34 35 40 75 79 12 13 14	V_{SS} AVRL AV_{SS} V_{SS} $\overline{RST}$ V_{SS} P34 P35 P36

- **Non-MBM27C1000 compatible pins**

Pin no.	Pin name	Treatment
47	MD0	Connect a pull-up resistor of 4.7 kΩ
48	MD1	
80	X0	
78	X0A	
81	X1	OPEN
77	X1A	
28 to 31	COM0 to COM3	
41 to 46	SEG00 to SEG05	
50 to 59	SEG06 to SEG15	
15	P37	Connect a pull-up resistor of about 1 MΩ to each pin.
16 to 20	P40 to P44	
22	P45	
23	P46	
24 to 27	V0 to V3	
32	AV_{CC}	
33	AVRH	
36 to 39	P50 to P53	
60 to 74	P60 to p76	
76	P77	

2. EPROM Programmer Socket Adapter

Part no.	Package	Compatible socket adapter Sun Hayato Co., Ltd.
MB90P623APFV	SQFP-100	ROM-100SQF-32DP-16L

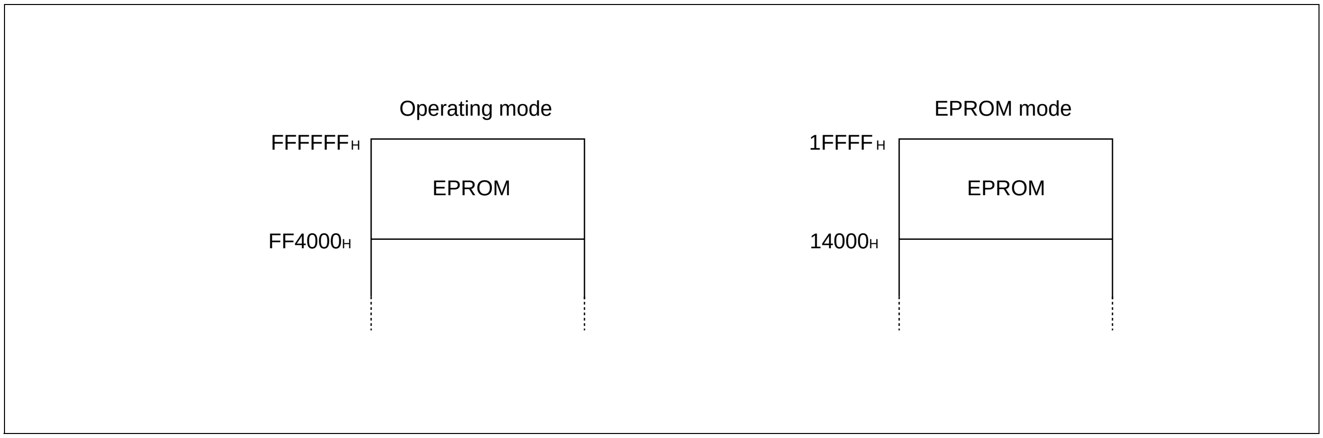
Inquiry: Sun Hayato Co., Ltd.: TEL (81)-3-3986-0403
FAX (81)-3-5396-9106

3. Programming Procedure

(1) Set the EPROM programmer to the MBM27C1000.

(2) Load the program data into the EPROM programmer at 14000_H to $1FFFF_H$.

The ROM addresses from $FF4000_H$ to $FFFFFF_H$ in operating mode of MB90P623A series correspond to 14000_H to $1FFFF_H$ in EPROM mode.



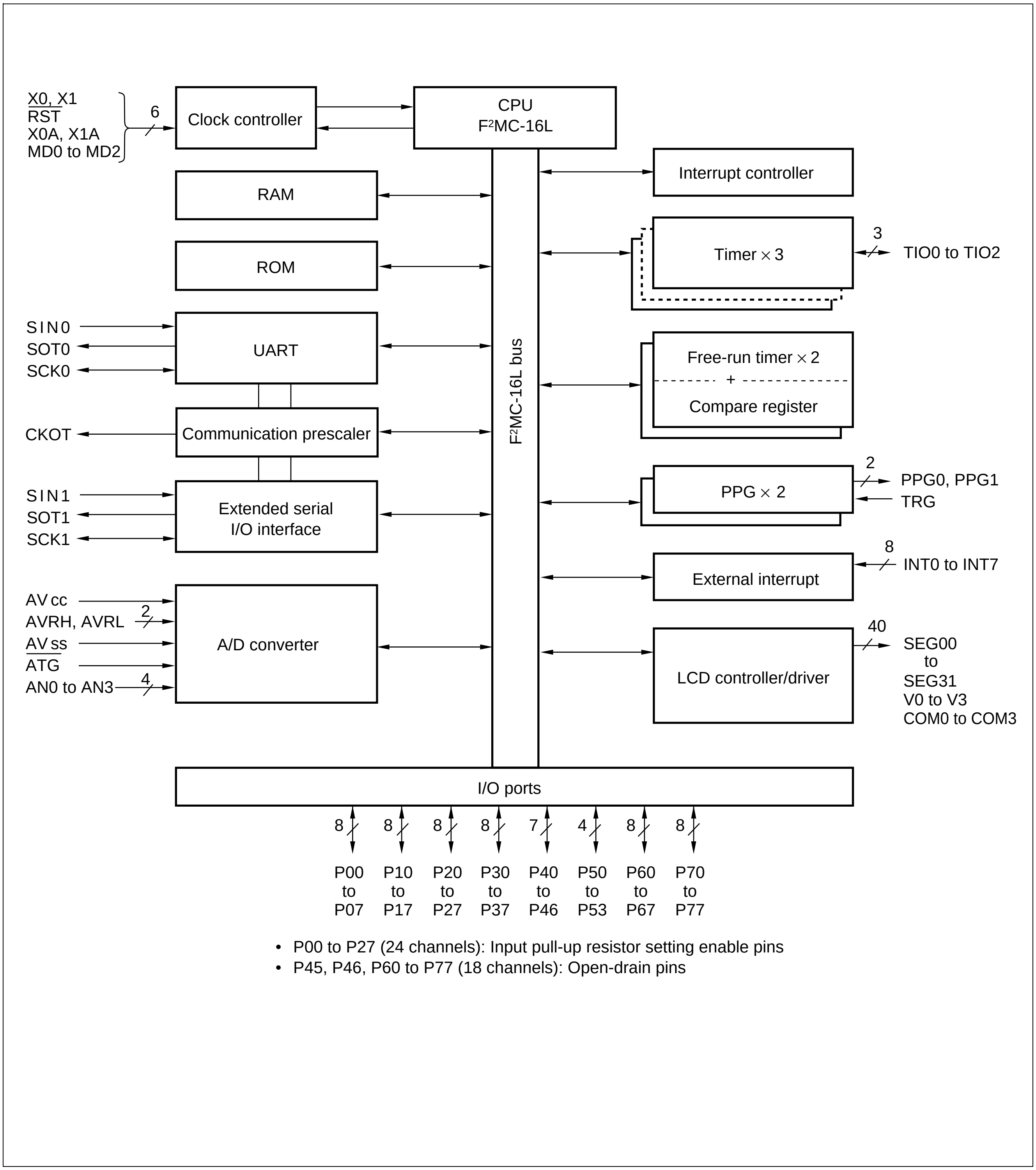
(3) Insert the MB90P623A in the socket adapter, and mount the socket adapter on the EPROM programmer. Pay attention to the orientation of the device and of the socket adapter when doing so.

(4) Activate the programming.

(5) If programming cannot be performed successfully, connect a 0.1 µF or similar capacitor between V_{CC} and GND and between V_{PP} and GND.

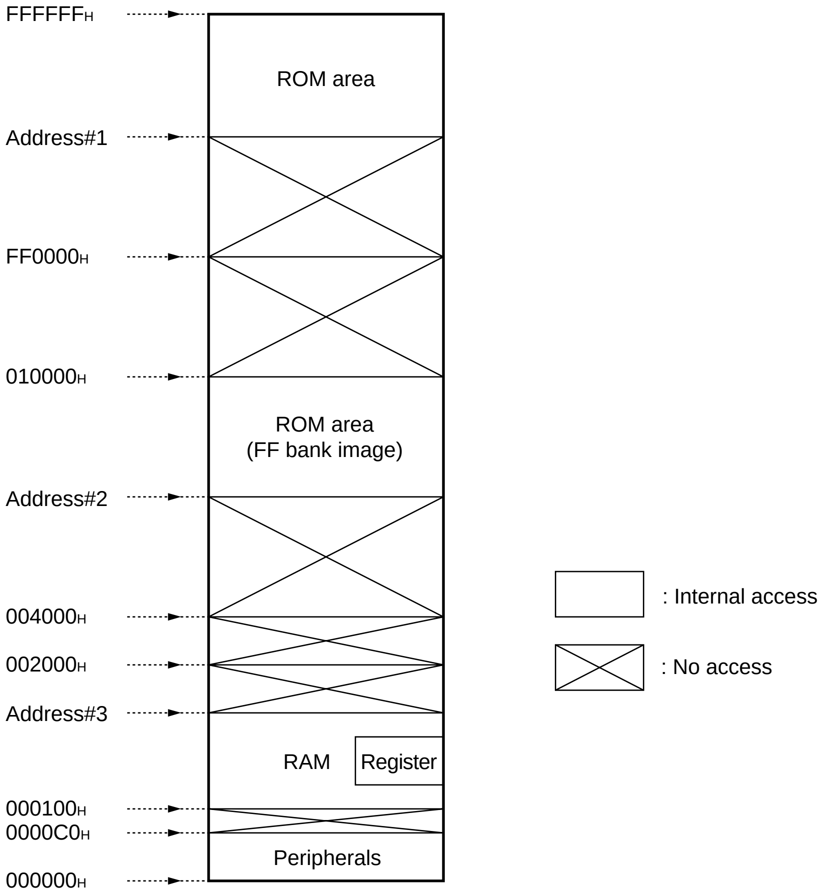
Note: Because the mask ROM products (MB90623A) do not have an EPROM mode, they cannot read data from the EPROM programmer.

■ BLOCK DIAGRAM



- P00 to P27 (24 channels): Input pull-up resistor setting enable pins
- P45, P46, P60 to P77 (18 channels): Open-drain pins

■ MEMORY MAP



Product	Address #1	Address #2	Address #3
MB90622A	FF8000H	008000H	000780H
MB90623A	FF4000H	004000H	000900H
MB90P623A	FF4000H	004000H	000900H

Note: While the ROM data image of bank FF can be seen in the upper portion of bank 00, this is done only to permit effective use of the C compiler’s small model. Because the lower 16 bits of bank FF address and the lower 16 bits of bank 00 are the same, it is possible to reference tables in ROM without declaring the “far” specification in the pointer.

■ I/O MAP

Address	Register	Register name	Access	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 X X X X
000004H	Port 4 data register	PDR4	R/W	Port 4	– X X X X X X X
000005H	Port 5 data register	PDR5	R/W	Port 5	– – – – X X X X
000006H	Port 6 data register	PDR6	R/W	Port 6	X X X X X X X X
000007H	Port 7 data register	PDR7	R/W	Port 7	– X X X X X X X
000008H to 0FH	Vacancy*				
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 0 0 0 0
000014H	Port 4 direction register	DDR4	R/W	Port 4	– 0 0 0 0 0 0 0
000015H	Port 5 direction register	DDR5	R/W	Port 5	– – – – 0 0 0 0
000016H	Port 6 direction register	DDR6	R/W	Port 6	0 0 0 0 0 0 0 0
000017H	Port 7 direction register	DDR7	R/W	Port 7	0 0 0 0 0 0 0 0
000018H to 19H	Vacancy*				
00001AH	Port 0 pull-up resistor setting register	RDR0	R/W	Port 0	0 0 0 0 0 0 0 0
00001BH	Port 1 pull-up resistor setting register	RDR1	R/W	Port 1	0 0 0 0 0 0 0 0
00001CH	Port 2 pull-up resistor setting register	RDR2	R/W	Port 2	0 0 0 0 0 0 0 0
00001DH	Analog input enable register	ADER	R/W	A/D	– – – – 1 1 1 1
00001EH	Clock output enable register	CKOT	R/W	Clock output (CKOT)	– – – – 0 0 0 0
00001FH	Vacancy*				
000020H	Serial mode register	SMR	R/W	UART	0 0 0 0 0 0 0 0
000021H	Serial control register	SCR	R/W		0 0 0 0 0 1 0 0
000022H	Serial input register/ Serial output register	SIDR/ SODR	R/W		X X X X X X X X
000023H	Serial status register	SSR	R/W		0 0 0 1 – – 0 0
000024H	Serial mode control status register	SMCS	R/W	Extended serial I/O interface	– – – 0 0 0 0 0
000025H					0 0 0 0 0 0 1 0
000026H	Serial data register	SDR	R/W		X X X X X X X X

(Continued)

Address	Register	Register name	Access	Resource name	Initial value
000027$_H$	Communication prescaler control register	CDCR	R/W	UART, I/O, serial	0 – – – 1 1 1 1
000028$_H$	DTP/Interrupt enable register	ENIR	R/W	DTP/external interrupt	0 0 0 0 0 0 0 0
000029$_H$	DTP/Interrupt source register	EIRR	R/W		0 0 0 0 0 0 0 0
00002A$_H$	Request level setting register	ELVR	R/W		0 0 0 0 0 0 0 0
00002B$_H$					0 0 0 0 0 0 0 0
00002C$_H$	A/D control status register	ADCS0	R/W	8/10-bit A/D converter	0 0 0 0 0 0 0 0
00002D$_H$		ADCS1			0 0 0 0 0 0 0 0
00002E$_H$	A/D data register	ADCR0	R/W		X X X X X X X X
00002F$_H$		ADCR1			0 0 0 0 0 0 X X
000030$_H$	PPG0 cycle setting register	PCSR0	W	16-bit PPG timer 0	X X X X X X X X
000031$_H$					X X X X X X X X
000032$_H$	PPG0 duty factor setting register	PDUT0	W		X X X X X X X X
000033$_H$					X X X X X X X X
000034$_H$	PPG0 control status register	PCNL0	R/W		0 0 0 0 0 0 0 0
000035$_H$		PCNH0			0 0 0 0 0 0 0 –
000036$_H$ to 37$_H$	Vacancy*				
000038$_H$	PPG1 cycle setting register	PCSR1	W	16-bit PPG timer 1	X X X X X X X X
000039$_H$					X X X X X X X X
00003A$_H$	PPG1 duty factor setting register	PDUT1	W		X X X X X X X X
00003B$_H$					X X X X X X X X
00003C$_H$	PPG1 control status register	PCNL1	R/W		0 0 0 0 0 0 0 0
00003D$_H$		PCNH1			0 0 0 0 0 0 0 –
00003E$_H$, 3F$_H$	Vacancy*				
000040$_H$	Timer control status register	TMCSR0	R/W	16-bit reload timer 0	0 0 0 0 0 0 0 0
000041$_H$					– – – – 0 0 0 0
000042$_H$	16-bit timer register	TMR0	R/W		X X X X X X X X
000043$_H$					X X X X X X X X
000044$_H$	16-bit reload register	TMRLR0	R/W		X X X X X X X X
000045$_H$					X X X X X X X X

(Continued)

Address	Register	Register name	Access	Resource name	Initial value
000046$_H$	Timer control status register 1	TMCSR1	R/W	16-bit reload timer 1	0 0 0 0 0 0 0 0
000047$_H$					– – – – 0 0 0 0
000048$_H$	16-bit timer register 1	TMR1	R/W		X X X X X X X X
000049$_H$					X X X X X X X X
00004A$_H$	16-bit reload register 1	TMRLR1	R/W		X X X X X X X X
00004B$_H$					X X X X X X X X
00004C$_H$ to 4F$_H$	Vacancy*				
000050$_H$	Timer control status register 2	TMCSR2	R/W	16-bit reload timer 2	0 0 0 0 0 0 0 0
000051$_H$					– – – – 0 0 0 0
000052$_H$	16-bit timer register 2	TMR2	R/W		X X X X X X X X
000053$_H$					X X X X X X X X
000054$_H$	16-bit reload register 2	TMRLR2	R/W		X X X X X X X X
000055$_H$					X X X X X X X X
000056$_H$	Timer data register 0	TCDT0	R	16-bit free-run timer 0	0 0 0 0 0 0 0 0
000057$_H$					0 0 0 0 0 0 0 0
000058$_H$	Timer control status register 0	TCS0	R/W		0 0 0 0 0 0 0 0
000059$_H$	Compare control status register 0	CCS0	R/W	Compare register block	0 0 0 0 – – 0 0
00005A$_H$	Timer 0 compare register 0	TCR00	R/W		X X X X X X X X
00005B$_H$					X X X X X X X X
00005C$_H$	Timer 0 compare register 1	TCR01	R/W		X X X X X X X X
00005D$_H$					X X X X X X X X
00005E$_H$, 5F$_H$	Vacancy*				
000060$_H$	Timer data register 1	TCDT1	R	16-bit free-run timer 1	0 0 0 0 0 0 0 0
000061$_H$					0 0 0 0 0 0 0 0
000062$_H$	Timer control status register 1	TCS1	R/W		0 0 0 0 0 0 0 0
000063$_H$	Compare control status register 1	CCS1	R/W	Compare register block	0 0 0 0 – – 0 0
000064$_H$	Timer 1 compare register 0	TCR10	R/W		X X X X X X X X
000065$_H$					X X X X X X X X
000066$_H$	Timer 1 compare register 1	TCR11	R/W		X X X X X X X X
000067$_H$					X X X X X X X X

(Continued)

Address	Register	Register name	Access	Resource name	Initial value
000068$_H$ to 6F$_H$	Vacancy*				
000070$_H$ to 7F$_H$	LCD display data RAM	VRAM	R/W	LCD controller/ driver	X X X X X X X X X X X X X X X X
000080$_H$	LCDC control register 0	LCR0	R/W		0 0 0 1 0 0 0 0
000081$_H$	LCDC control register 1	LCR1			0 – – 0 0 0 0 0
000082$_H$ to 8F$_H$	Vacancy*				
000090$_H$ to 9E$_H$	System reserved area*				
00009F$_H$	Delayed interrupt source generation/ release register	DIRR	R/W	Delayed interrupt generation module	– – – – – – – 0
0000A0$_H$	Low-power consumption mode control register	LPMCR	R/W	Low-power consumption	0 0 0 1 1 0 0 0
0000A1$_H$	Clock selection register	CKSCR	R/W		1 1 1 1 1 1 0 0
0000A2$_H$ to A7$_H$	Vacancy*				
0000A8$_H$	Watchdog timer control register	WDTC	R/W	Watchdog timer	X X X X X X X X
0000A9$_H$	Timebase timer control register	TBTC	R/W	Timebase timer	1 – – 0 0 0 0 0
0000AA$_H$	Watch timer control register	WTC	R/W	Watch timer	1 X – 0 0 0 0 0
0000AB$_H$ to AF$_H$	Vacancy*				
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
0000B8$_H$	Interrupt control register 08	ICR08	R/W		0 0 0 0 0 1 1 1
0000B9$_H$	Interrupt control register 09	ICR09	R/W		0 0 0 0 0 1 1 1
0000BA$_H$	Interrupt control register 10	ICR10	R/W		0 0 0 0 0 1 1 1
0000BB$_H$	Interrupt control register 11	ICR11	R/W		0 0 0 0 0 1 1 1
0000BC$_H$	Interrupt control register 12	ICR12	R/W		0 0 0 0 0 1 1 1
0000BD$_H$	Interrupt control register 13	ICR13	R/W		0 0 0 0 0 1 1 1

(Continued)

(Continued)

Address	Register	Register name	Access	Resource name	Initial value
0000BE$_H$	Interrupt control register 14	ICR14	R/W	Interrupt controller	0 0 0 0 0 1 1 1
0000BF$_H$	Interrupt control register 15	ICR15	R/W		0 0 0 0 0 1 1 1
0000C0$_H$ to FF$_H$	Vacancy*				

* : Access prohibited.

Explanation of initial values
- 0: The initial value of this bit is “0”.
- 1: The initial value of this bit is “1”.
- X: The initial value of this bit is undefined.
- –: This bit is not used. No initial value is defined.

■ INTERRUPT SOURCES AND THEIR INTERRUPT VECTORS AND INTERRUPT CONTROL REGISTERS

Interrupt source	I^{2}OS support	Interrupt vector			Interrupt control register	
		No.		Address	ICR	Address
Reset	×	#08	08$_H$	FFFFDC$_H$	—	—
INT9 instruction	×	#09	09$_H$	FFFFD8$_H$	—	—
Exception	×	#10	0A$_H$	FFFFD4$_H$	—	—
External interrupt #0	○	#11	0B$_H$	FFFFD0$_H$	ICR00	0000B0$_H$
External interrupt #1	○	#12	0C$_H$	FFFFCC$_H$		
External interrupt #2	○	#13	0D$_H$	FFFFC8$_H$	ICR01	0000B1$_H$
External interrupt #3	○	#14	0E$_H$	FFFFC4$_H$		
External interrupt #4	○	#15	0F$_H$	FFFFC0$_H$	ICR02	0000B2$_H$
External interrupt #5	○	#16	10$_H$	FFFFBC$_H$		
External interrupt #6	○	#17	11$_H$	FFFFB8$_H$	ICR03	0000B3$_H$
External interrupt #7	○	#18	12$_H$	FFFFB4$_H$		
Extended serial I/O interface	○	#19	13$_H$	FFFFB0$_H$	ICR04	0000B4$_H$
Free-run timer 0 overflow	○	#21	15$_H$	FFFFA8$_H$	ICR05	0000B5$_H$
Free-run timer 1 overflow	○	#22	16$_H$	FFFFA4$_H$		
Free-run timer 0 and compare register 0 matched	○	#23	17$_H$	FFFFA0$_H$	ICR06	0000B6$_H$
Free-run timer 0 and compare register 1 matched	○	#24	18$_H$	FFFF9C$_H$		
Free-run timer 1 and compare register 0 matched	○	#25	19$_H$	FFFF98$_H$	ICR07	0000B7$_H$
Free-run timer 1 and compare register 1 matched	○	#26	1A$_H$	FFFF94$_H$		
PPG timer #0	○	#27	1B$_H$	FFFF90$_H$	ICR08	0000B8$_H$
PPG timer #1	○	#28	1C$_H$	FFFF8C$_H$		
16-bit reload timer #0	○	#29	1D$_H$	FFFF88$_H$	ICR09	0000B9$_H$
16-bit reload timer #1	○	#30	1E$_H$	FFFF84$_H$		
16-bit reload timer #2	○	#31	1F$_H$	FFFF80$_H$	ICR10	0000BA$_H$
A/D converter measurement complete	○	#33	21$_H$	FFFF78$_H$	ICR11	0000BB$_H$
Watch prescaler	×	#35	23$_H$	FFFF70$_H$	ICR12	0000BC$_H$
Timebase timer interval interrupt	×	#36	24$_H$	FFFF6C$_H$		
UART 0 transmission complete	○	#37	25$_H$	FFFF68$_H$	ICR13	0000BD$_H$
UART 1 reception complete	◎	#39	27$_H$	FFFF60$_H$	ICR14	0000BE$_H$
Delayed interrupt generation module	×	#42	2A$_H$	FFFF54$_H$	ICR15	0000BF$_H$

○ : The request flag is cleared by the I^{2}OS interrupt clear signal (without stop requests).
◎ : The request flag is cleared by the I^{2}OS interrupt clear signal (with stop requests).
× : The request flag is not cleared by the I^{2}OS interrupt clear signal.

Note: Do not set I^{2}OS startup in an ICR$_{xx}$ that does not support I^{2}OS.

■ PERIPHERALS

1. Parallel Ports

The MB90620A series has 59 input/output pins.
In the twenty four input/output ports mapped on port 0 to 2, pull-up resistors are selectively added during input state operations depending on the settings in the resistor setting register.
P45, P46, port 6 and port 7 are open-drain ports.
Port 6 and port 7 are combined with the LCD segment pin function.

(1) Register configuration

Port data register

Address: PDR1 000001H
PDR3 000003H
PDR5 000005H
PDR7 000007H

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	PDRx

Port data register

Address: PDR0 000000H
PDR2 000002H
PDR4 000004H
PDR6 000006H

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	PDRx

Notes: Bit 7 of port 4 does not have a register bit.
Bit 4 to bit 7 of port 5 does not have a register bit.

Port direction register

Address: DDR1 000011H
DDR3 000013H
DDR5 000015H
DDR7 000017H

bit	15	14	13	12	11	10	9	8	
	DD×7	DD×6	DD×5	DD×4	DD×3	DD×2	DD×1	DD×0	DDRx

Port direction register

Address: DDR0 000010H
DDR2 000012H
DDR4 000014H
DDR6 000016H

bit	7	6	5	4	3	2	1	0	
	DD×7	DD×6	DD×5	DD×4	DD×3	DD×2	DD×1	DD×0	DDRx

Notes: Bit 7 of port 4 does not have a register bit.
Bit 4 to bit 7 of port 5 does not have a register bit.

Pull-up resistor setting register

Address: 00001BH

bit	15	14	13	12	11	10	9	8	
	RD×7	RD×6	RD×5	RD×4	RD×3	RD×2	RD×1	RD×0	RDR1

Pull-up resistor setting register

Address: 00001AH
00001CH

bit	7	6	5	4	3	2	1	0	
	RD×7	RD×6	RD×5	RD×4	RD×3	RD×2	RD×1	RD×0	RDR0, RDR2

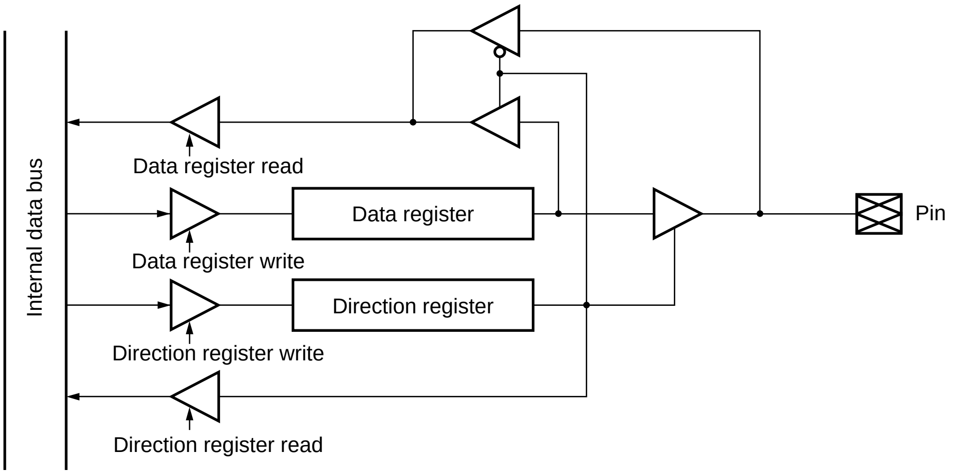
Analog input enable register

Address: 00001DH

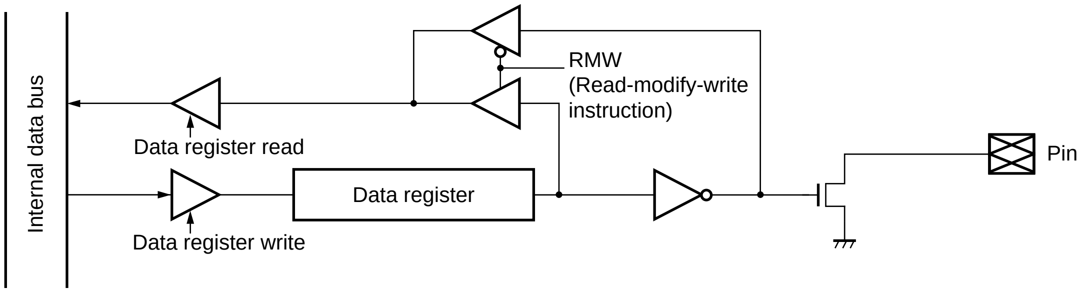
bit	15	14	13	12	11	10	9	8	
	—	—	—	—	ADE3	ADE2	ADE1	ADE0	ADER

(2) Block Diagram

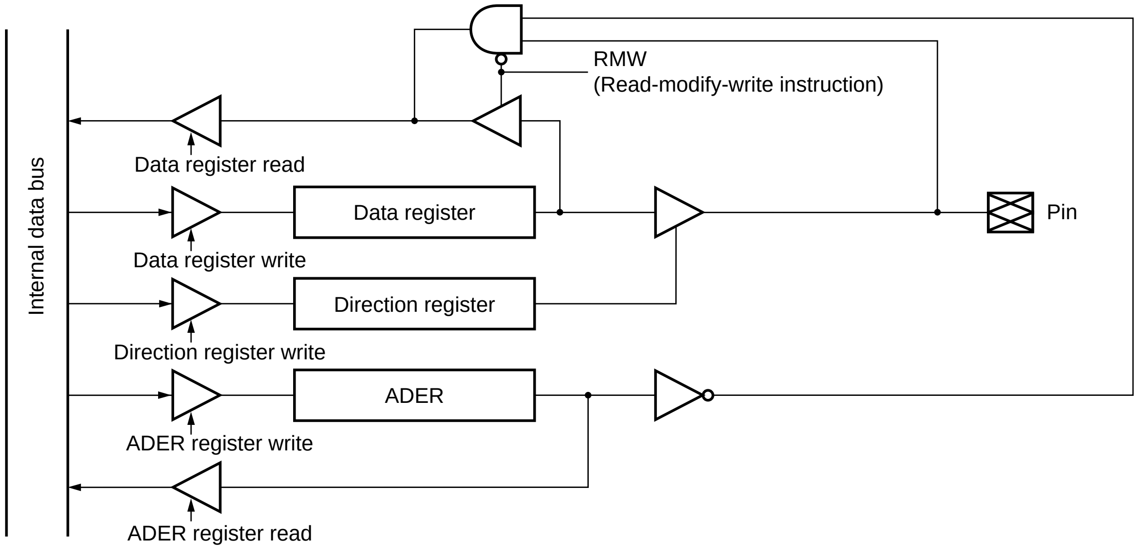
- **I/O Port**



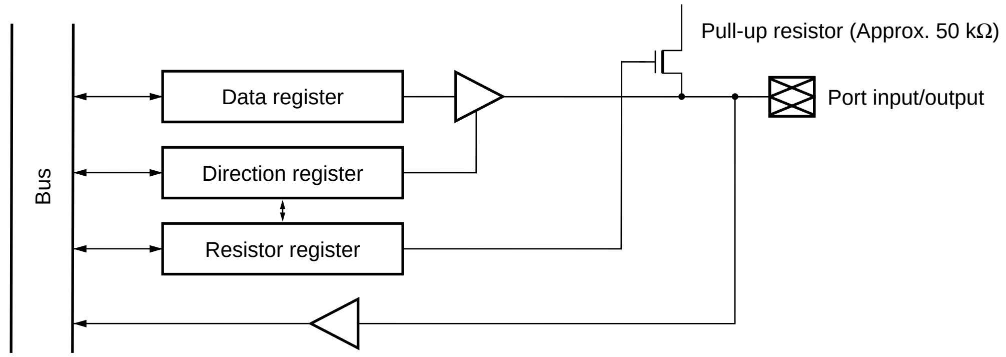
- **Open-drain Port**



- **Port combined with the A/D converter functions**



- **Port with a pull-up resistor option**



2. UART

The UART is a serial I/O port for CLK asynchronous (start-stop synchronization) communications or for CLK synchronous communications. The features of this module are described below:

- Full-duplex double buffer
- CLK asynchronous (start-stop synchronization) communications and CLK synchronous communications capable
- Supports multiprocessor mode
- Built-in dedicated baud rate generator

  CLK asynchronous: 9615, 31250, 4808, 2404, 1202 bps
  CLK synchronous: 1 M, 500K, 250K, 125K, 62.5K bps
  } For a 6, 8, 10, 12, or 16 MHz clock.
- Permits setting of any desired baud rate according to an external clock input
- Error detection function (parity errors, framing errors, and overrun errors)
- NRZ code as transfer signal
- Supports Intelligent I/O Service

(1) Register Configuration

	bit	7	6	5	4	3	2	1	0	
Address: 000020_H		MD1	MD0	CS2	CS1	CS0	Reserved	SCKE	SOE	Serial mode register (SMR)

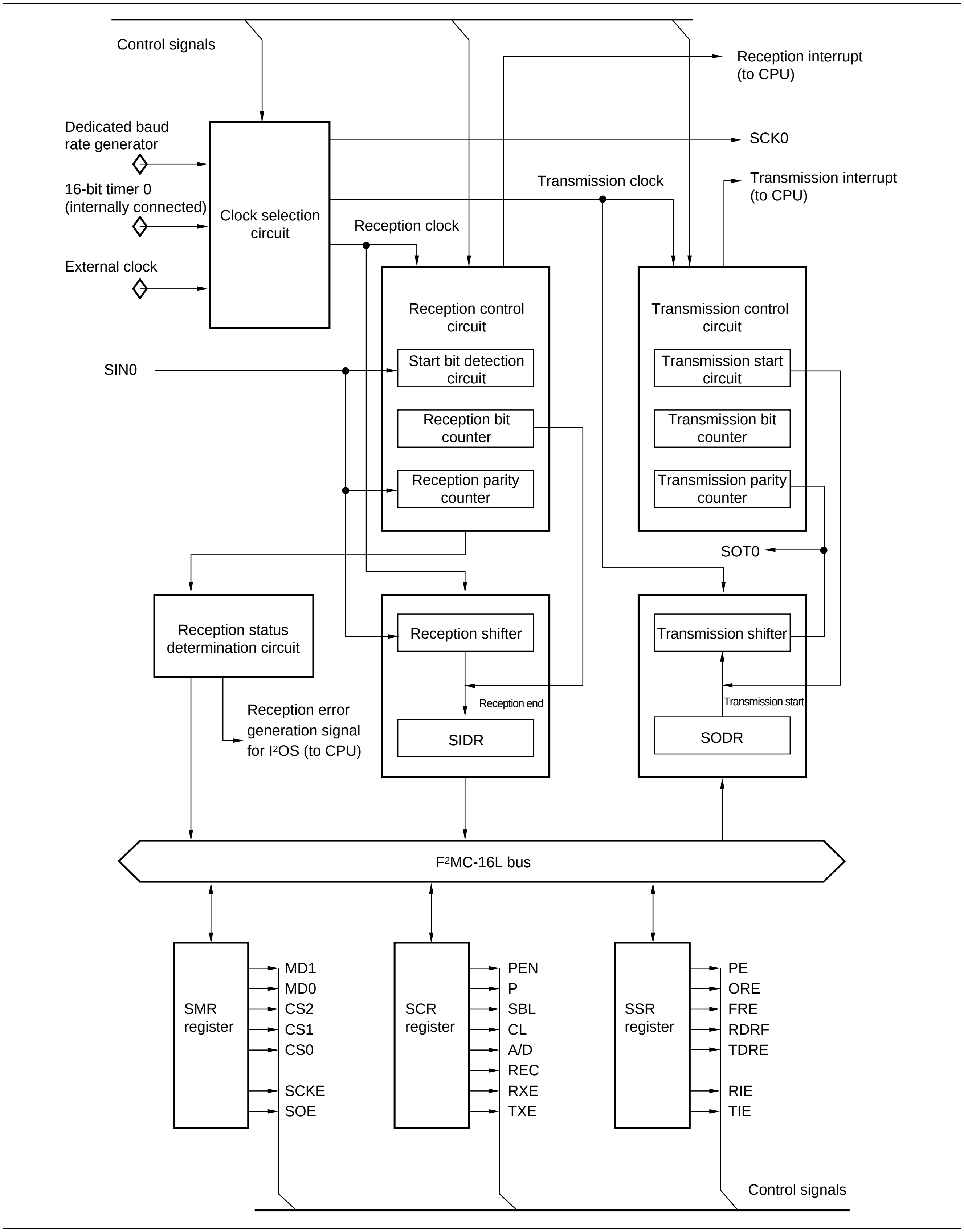
	bit	15	14	13	12	11	10	9	8	
Address: 000021_H		PEN	P	SBL	CL	A/D	REC	RXE	TXE	Serial control register (SCR)

	bit	7	6	5	4	3	2	1	0	
Address: 000022_H		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: 000023_H		PE	OPE	FRE	RDRF	TDRE	—	RIE	TIE	Serial status register (SSR)

	bit	15	14	13	12	11	10	9	8	
Address: 000027_H		MD	—	—	—	DIV3	DIV2	DIV1	DIV0	Communication prescaler control register (CDCR)

(2) Block Diagram



3. Extended Serial I/O Interface

This block consists of an 8-bit serial I/O interface that can perform clock synchronous data transfer. Either LSB-first or MSB-first data transfer can be selected. The serial I/O port to be used can also be selected.
The following two serial I/O operation modes are available.

Internal shift clock mode: Data transfer is synchronization with the internal clock.
External shift clock mode: Data transfer is synchronization with the clock input from the external pin (SCK1). By manipulating the general-purpose port that shares the external pin (SCK1), this mode also enables the data transfer operation to be driven by CPU instructions.

(1) Register Configuration

Address: 000025H — Serial mode control status register (SMCS)

bit	15	14	13	12	11	10	9	8
	SMD2	SMD1	SMD0	SIE	SIR	BUSY	STOP	STRT

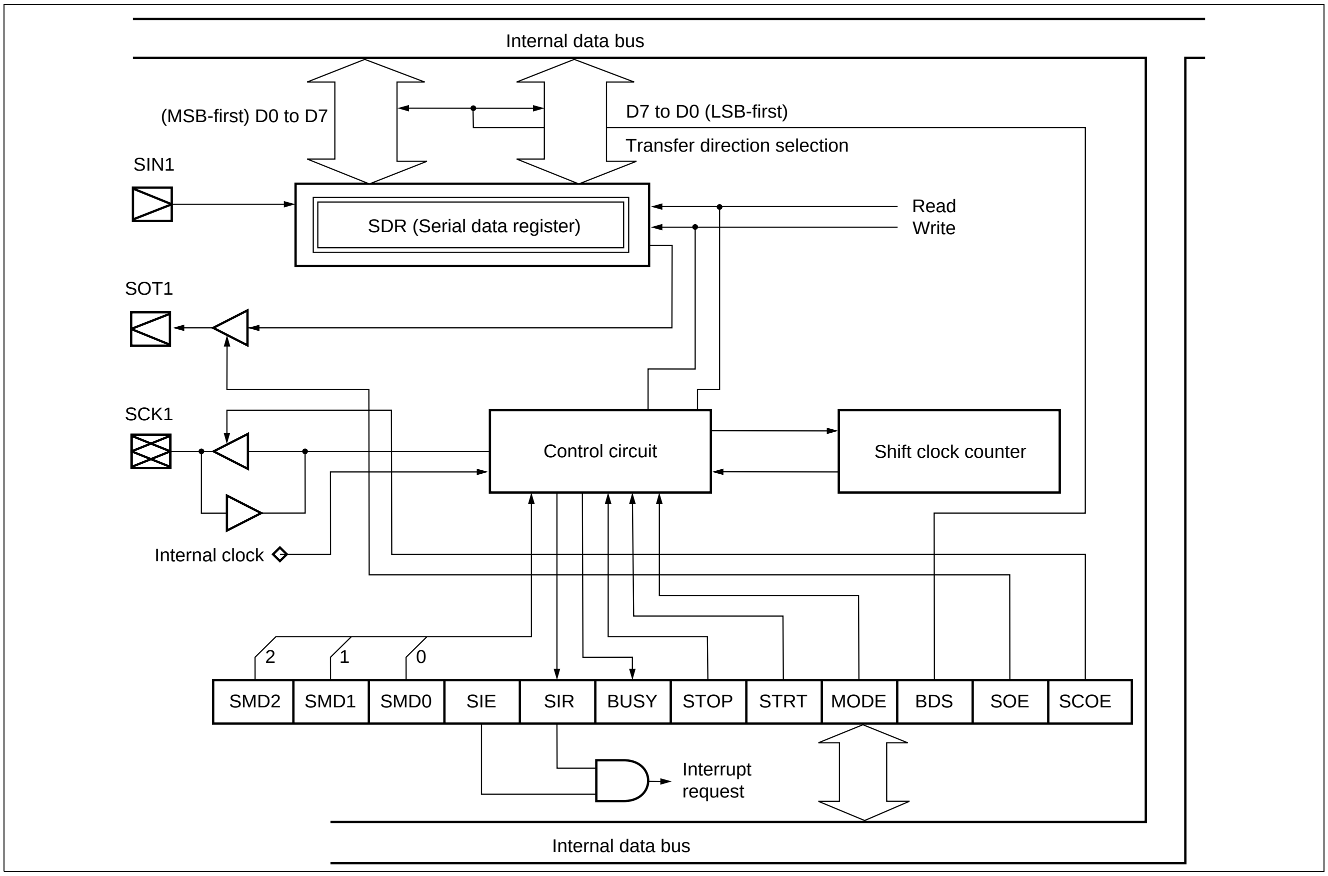
Address: 000024H

bit	7	6	5	4	3	2	1	0
	—	—	—	—	MODE	BDS	SOE	SCOE

Address: 000026H — Serial data register (SDR)

bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0

(2) Block Diagram



4. A/D Converter

The A/D converter converts the analog input voltage into a digital value. The features of this module are as follows:

- Conversion time: Minimum of 7 µs per channel (12 MHz machine clock)
- RC-type successive approximation conversion method with sample and hold circuit
- 8-bit/10-bit resolution
- Analog input is selectable by software from among 4 channels
- A/D conversion mode selectable from the following three:
  One-shot conversion mode: Converts a specified channel once.
  Continuous conversion mode: Converts a specified channel repeatedly.
  Stop conversion mode: Pauses after converting one channel and wait until the next activation (permits synchronization of start of conversion).
- Conversion mode:
  Single-conversion mode: Converts one channel (when the start and stop channels are the same).
  Scan conversion mode: Converts several consecutive channels (when the start and stop channels are different).
- When A/D conversion is completed, an “A/D conversion complete” interrupt request can be issued to the CPU. Because generating this interrupt can be used to activate the I^{2}OS and transfer the A/D conversion results to memory, this function is suitable for continuous processing.
- Activation sources can be selected from among software, an external trigger (falling edge), and timer (rising edge).

(1) Register Configuration

A/D converter control status register (ADCS1, ADCS0)

	bit	15	14	13	12	11	10	9	8
Address: 00002DH		BUSY	INT	INTE	PAUS	STS1	STS0	STRT	Reserved

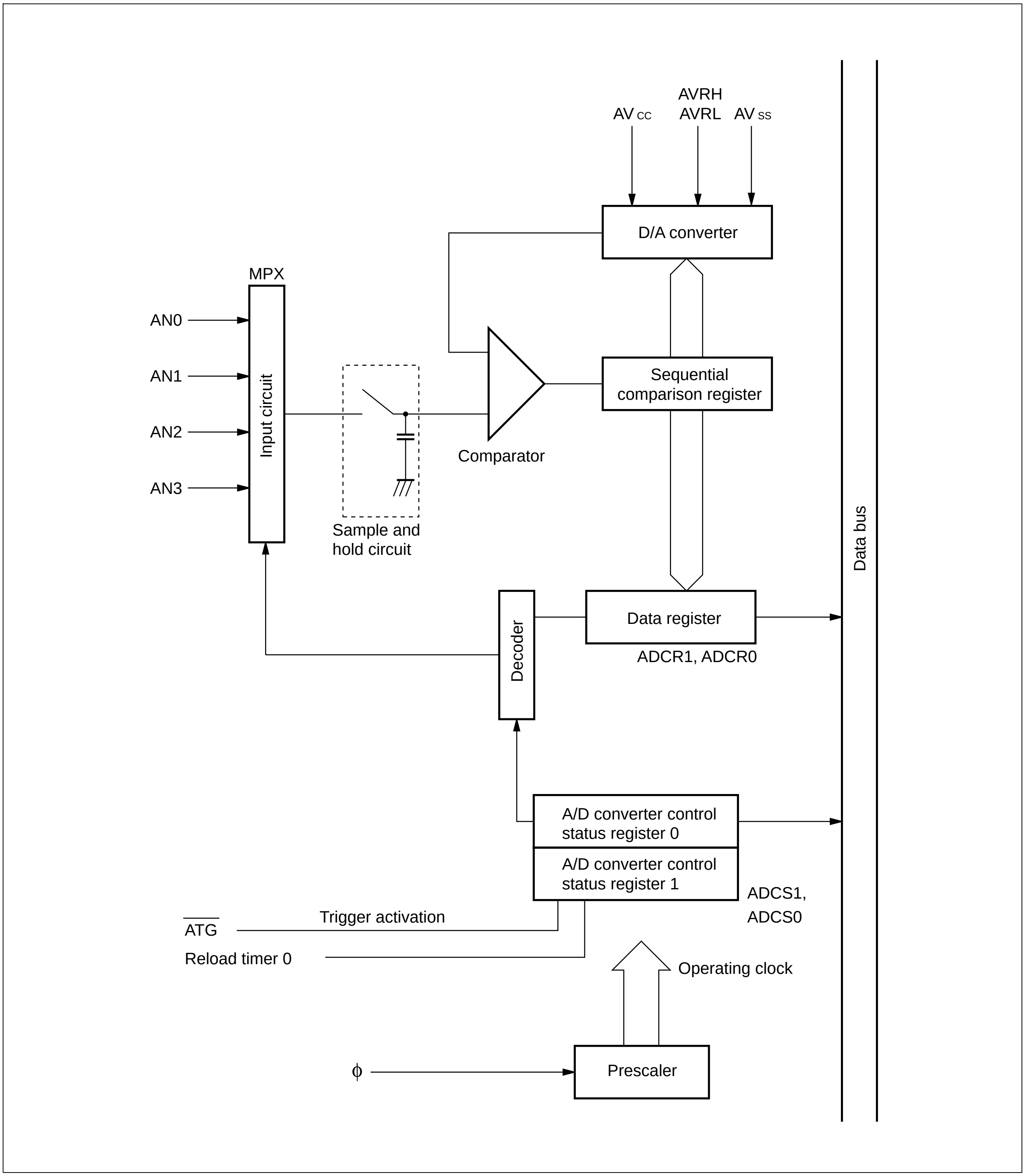
	bit	7	6	5	4	3	2	1	0
Address: 00002CH		MD1	MD0	Reserved	ANS1	ANS0	Reserved	ANE1	ANE0

A/D converter data register (ADCR1, ADCR0)

	bit	15	14	13	12	11	10	9	8
Address: 00002FH		0	0	0	0	0	0	D9	D8

	bit	7	6	5	4	3	2	1	0
Address: 00002EH		D7	D6	D5	D4	D3	D2	D1	D0

(2) Block Diagram



5. 16-bit Timer (with Event Count Function)

The 16-bit timer consists of a 16-bit down counter, a 16-bit reload register, one input and output pin (TIN$_X$,TOT$_X$), and a control register. Three internal clocks and an external clock can be selected for the input clock. When in reload mode, a toggled output waveform is output, while in one-shot mode a square wave indicating that the count is in progress is output pin (TOT$_X$). The input pin (TIN$_X$) serves as an event input in event count mode, and can be used for trigger input or gate input in internal clock mode.

(1) Register Configuration

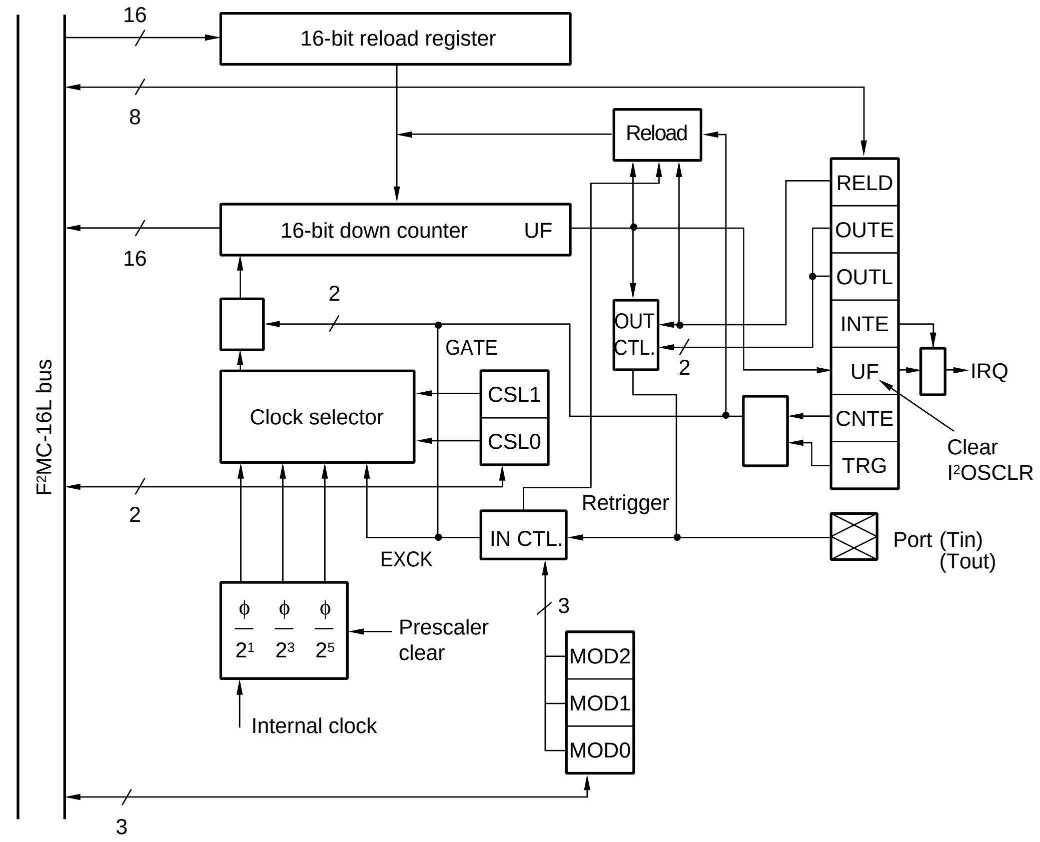
Address	bit	7	6	5	4	3	2	1	0	
Address: 000040$_H$: 000046$_H$: 000050$_H$		MOD0	OUTE	OUTL	RELD	INTE	UF	CNTE	TRG	

Address	bit	15	14	13	12	11	10	9	8	
Address: 000041$_H$: 000047$_H$: 000051$_H$		—	—	—	—	CSL1	CSL0	MOD2	MOD1	Timer control status register 0 to 2 (TMCSR$_0$ to TMCSR$_2$)

Address	bit	15 … 0	
Address: 000042$_H$: 000048$_H$: 000052$_H$			16-bit timer register 0 to 2 (TMR$_0$ to TMR$_2$)

Address	bit	15 … 0	
Address: 000044$_H$: 00004A$_H$: 000054$_H$			16-bit reload register 0 to 2 (TMRLR$_0$ to TMRLR$_2$)

(2) Block Diagram



6. 16-bit Free-run Timer

The 16-bit free-run timer consists of a 16-bit up counter, a control status register, and a compare register.

- Count clock is selectable from 4 types.
- A counter over flow interrupt can be generated.
- An interrupt can be generated on matching with the compare register value.
- Initialization of the counter on matching with compare register 0 value is enabled depending on the mode settings.

(1) Register Configuration

Address: 000056H
: 000060H

bit	15	14	13	12	11	10	9	8
	T15	T14	T13	T12	T11	T10	T09	T08

bit	7	6	5	4	3	2	1	0
	T07	T06	T05	T04	T03	T02	T01	T00

Timer data register 0, 1
(TCDT0, TCDT1)

Address: 000059H
: 000063H

bit	15	14	13	12	11	10	9	8
	ICP1	ICP0	ICE1	ICE0	—	—	CST1	CST0

Compare control status
0, 1 register
(CCS0, CCS1)

Address: 000058H
: 000062H

bit	7	6	5	4	3	2	1	0
	Reserved	IVF	IVFE	STOP	MODE	CLR	CLK1	CLK0

Timer control status 0, 1
register
(TCS0, TCS1)

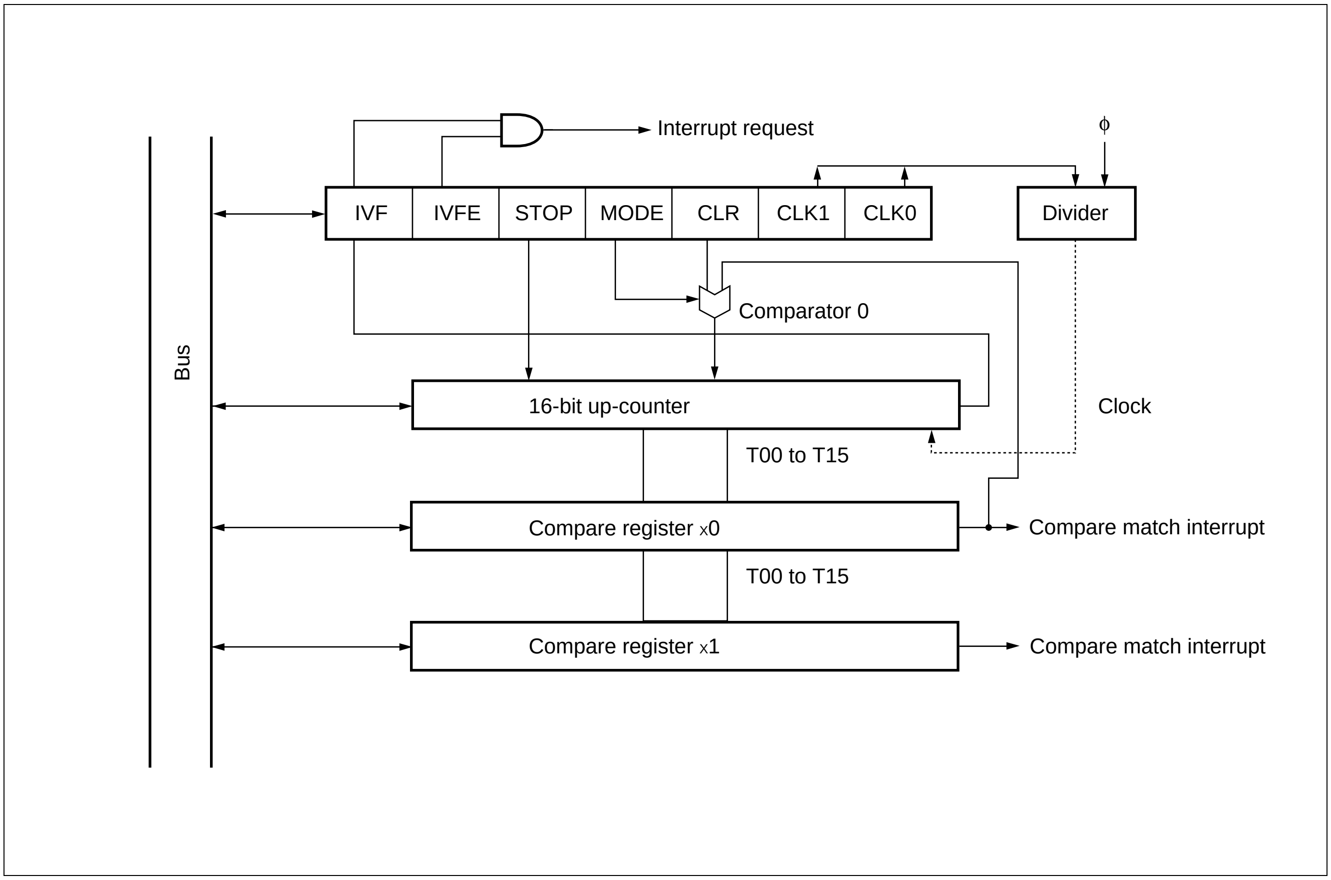
Address: 00005AH
: 00005CH
: 000064H
: 000066H

bit	15	14	13	12	11	10	9	8
	C15	C14	C13	C12	C11	C10	C09	C08

bit	7	6	5	4	3	2	1	0
	C07	C06	C05	C04	C03	C02	C01	C00

Timer 0, 1 compare register
(TCR00, TCR01/
TCR10, TCR11)

(2) Block Diagram



7. 16-bit PPG Timer

This module can output a pulse synchronized with an external trigger or a software trigger. In addition, the cycle and duty ratio of the output pulse can be changed as desired by overwriting the two 16-bit register values.

PWM function:	Synchronizes pulse with trigger, and permits programming of the pulse output by overwriting the register values mentioned above. This function permits use as a D/A converter with the addition of external circuits.
One-shot function:	Detects the edge of trigger input, and permits single-pulse output.

(1) Register Configuration

Address: 00035H
: 0003DH

bit	15	14	13	12	11	10	9	8
	CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	—

Address: 00034H
: 0003CH

bit	7	6	5	4	3	2	1	0
	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	POEN	OSEL

PPG0, 1 control status register (PCNH0, PCNH1)

PPG0, 1 control status register (PCNL0, PCNL1)

Address: 00031H
: 00039H

bit	15	14	13	12	11	10	9	8

Address: 00030H
: 00038H

bit	7	6	5	4	3	2	1	0

PPG0, 1 cycle setting register (PCSR0, PCSR1)

Address: 00033H
: 0003BH

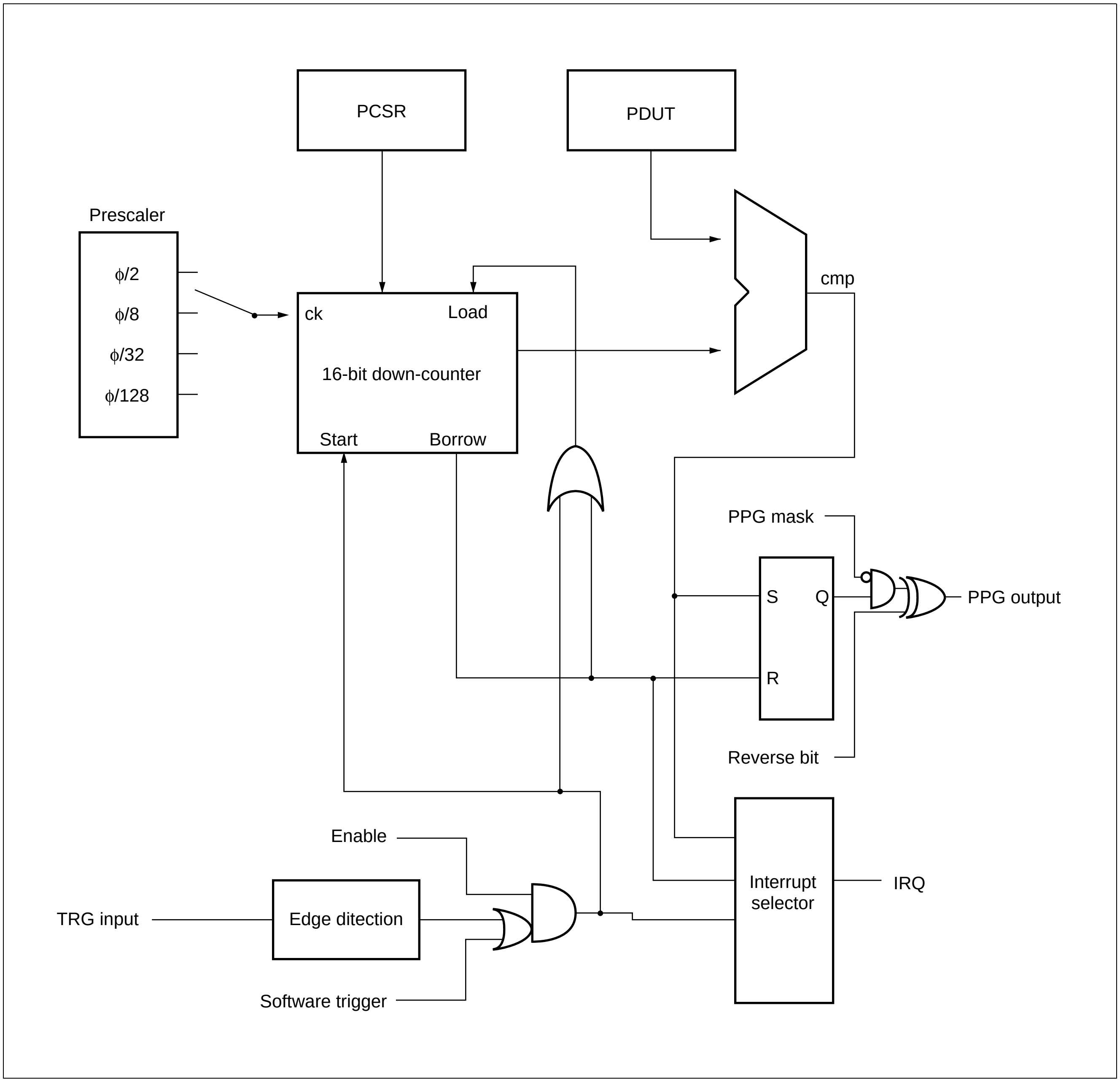
bit	15	14	13	12	11	10	9	8

Address: 00032H
: 0003AH

bit	7	6	5	4	3	2	1	0

PPG0, 1 duty setting register (PDUT0, PDUT1)

(2) Block Diagram



8. LCD Controller/driver

The LCD controller driver consists of the display controller for generating the segment signal and common signal according to data set in the display data memory, the segment driver and the common driver capable of directly driving the LCD panel (Liquid Crystal Display).
Primary functions are as follows;

- LCD direct drive function
- Common output 4 channels (COM0 to COM3), segment output 32 channels (SEG0 to SEG31)
- Built-in 16 bytes of data memory for display
- Duty ratio selective from 1/2, 1/3 and 1/4
- Driving clock source selective from the main clock (4 MHz) and the sub clock (32 kHz)
- SEG 16 to SEG 31 can be used as open-drain ports.

(1) Register Configuration

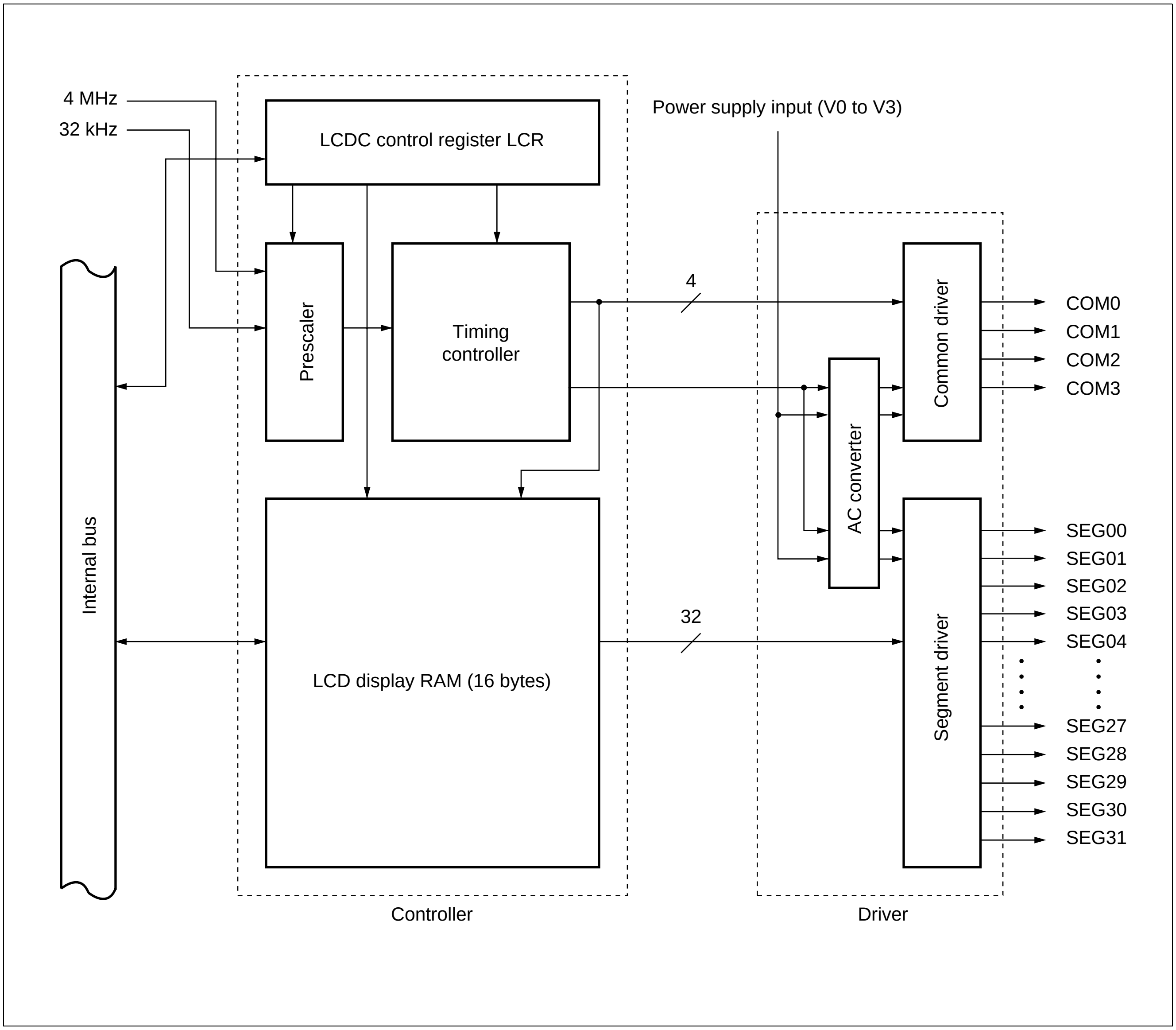
LCD control register

	bit 15	8 7	0	
Address: 000080$_H$: 000081$_H$	LCR1		LCR0	LCR0/LCR1

LCD display RAM

Address: 000080$_H$	b3	b2	b1	b0	SEG00
	b7	b6	b5	b4	SEG01
Address: 000080$_H$	b3	b2	b1	b0	SEG02
	b7	b6	b5	b4	SEG03
Address: 000080$_H$	b3	b2	b1	b0	SEG04
	b7	b6	b5	b4	SEG05
:	:	:	:	:	
Address: 000080$_H$	b3	b2	b1	b0	SEG16
	b7	b6	b5	b4	SEG17
Address: 000080$_H$	b3	b2	b1	b0	SEG18
	b7	b6	b5	b4	SEG19
:	:	:	:	:	
Address: 000080$_H$	b3	b2	b1	b0	SEG28
	b7	b6	b5	b4	SEG29
Address: 000080$_H$	b3	b2	b1	b0	SEG30
	b7	b6	b5	b4	SEG31
	COM3	COM2	COM1	COM0	

(2) Block Diagram



9. DTP/External Interrupt

The DTP (Data Transfer Peripheral) is a peripheral, positioned between peripherals external to the device and the F^{2}MC-16L CPU, that accepts DMA requests or interrupt requests generated by external peripherals and transfers them to the F^{2}MC-16L CPU to activate the Intelligent I/O Service or interrupt processing.

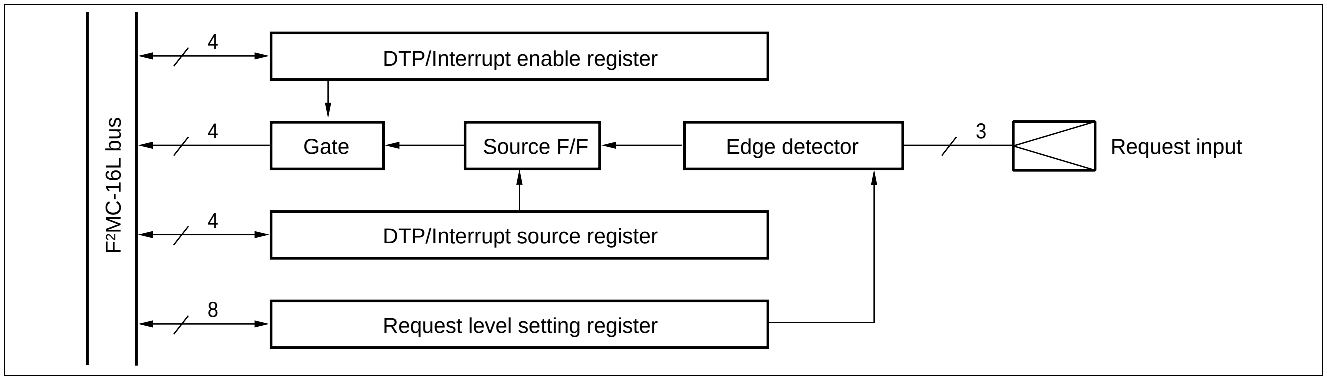
In the case of the Intelligent I/O Service, there are two request levels that can be selected: high and low; in the case of an external interrupt request, there are a total of four request levels that can be selected: high, low, rising edge and falling edge.

(1) Register Configuration

Address	bit 15	0	
000029$_H$: 000028$_H$	EIRR	ENIR	DTP/Interrupt enable register

Address	bit 15 … 0	
00002B$_H$: 00002A$_H$	ELVR	Request level setting register

(2) Block Diagram


F^{2}MC-16L bus
4
DTP/Interrupt enable register
4
Gate
Source F/F
Edge detector
3
Request input
4
DTP/Interrupt source register
8
Request level setting register

10. Watchdog Timer, Timebase Timer, and Watch Timer Functions

The watchdog timer consists of a 2-bit watchdog counter that uses the carry signal from the 18-bit timebase timer or the 15-bit watch timer as a clock source, a control register, and a watchdog reset controller.

The timebase timer consists of an 18-bit timer and a circuit that controls interval interrupts. Note that the timebase timer uses the main clock, regardless of the setting of the MCS bit and SCS bit in CKSCR.

The watch timer consists of a 15-bit timer and a circuit that controls interval interrupts. Note that the watch timer uses the sub clock, regardless of the setting of the MCS bit and SCS bit in CKSCR.

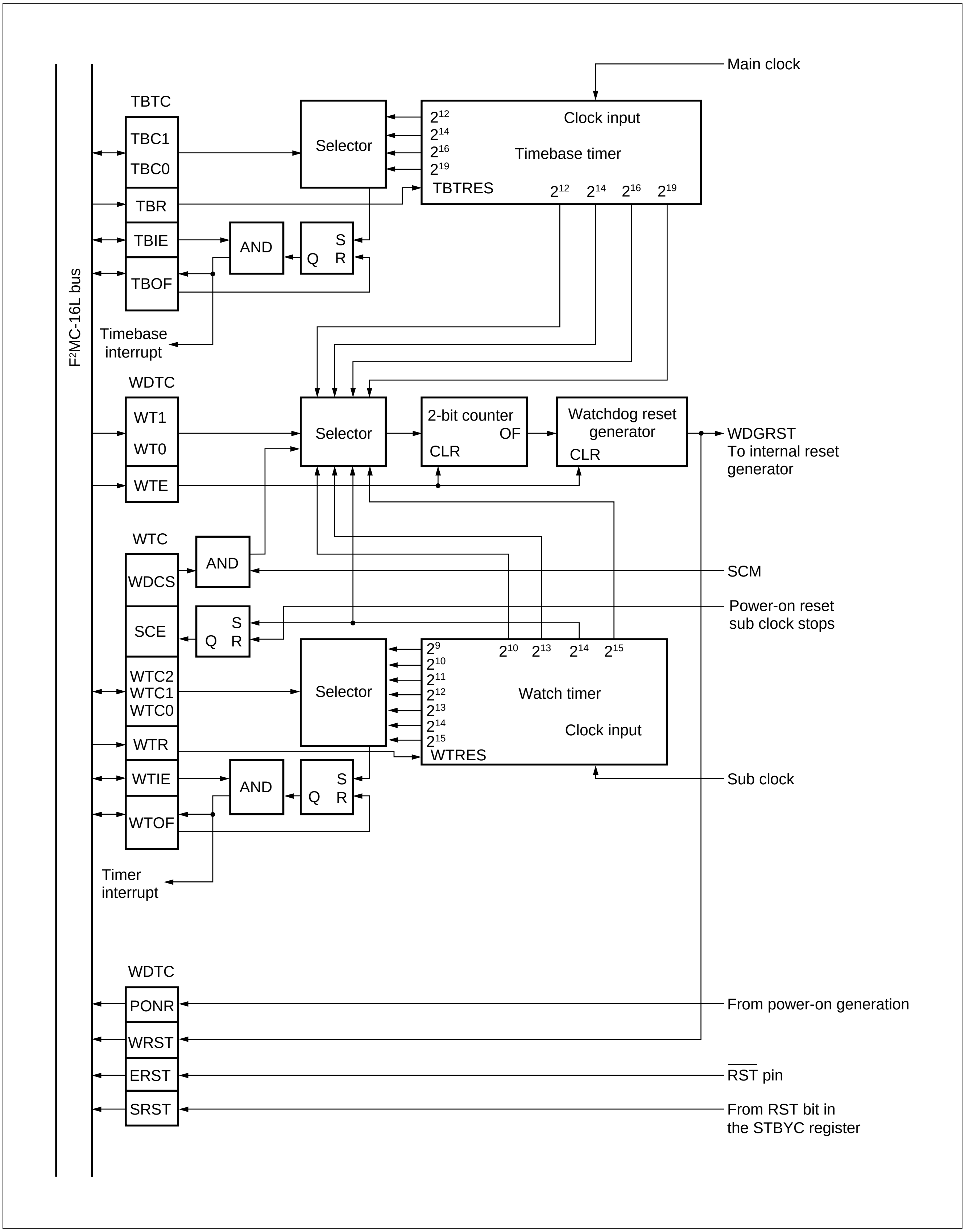
(1) Register Configuration

Address	bit	7	6	5	4	3	2	1	0	
0000A8$_H$		PONR	—	WRST	ERST	SRST	WTE	WT1	WT0	Watchdog timer control register (WDTC)

Address	bit	15	14	13	12	11	10	9	8	
0000A9$_H$		Reserved	—	—	TBIE	TBOF	TBR	TBC1	TBC0	Timebase timer control register (TBTC)

Address	bit	7	6	5	4	3	2	1	0	
0000AA$_H$		WDCS	SCE	WTIE	WTOF	WTR	WTC2	WTC1	WTC0	Watch timer control register (WTC)

(2) Block Diagram



11. Delayed Interrupt Generation Module

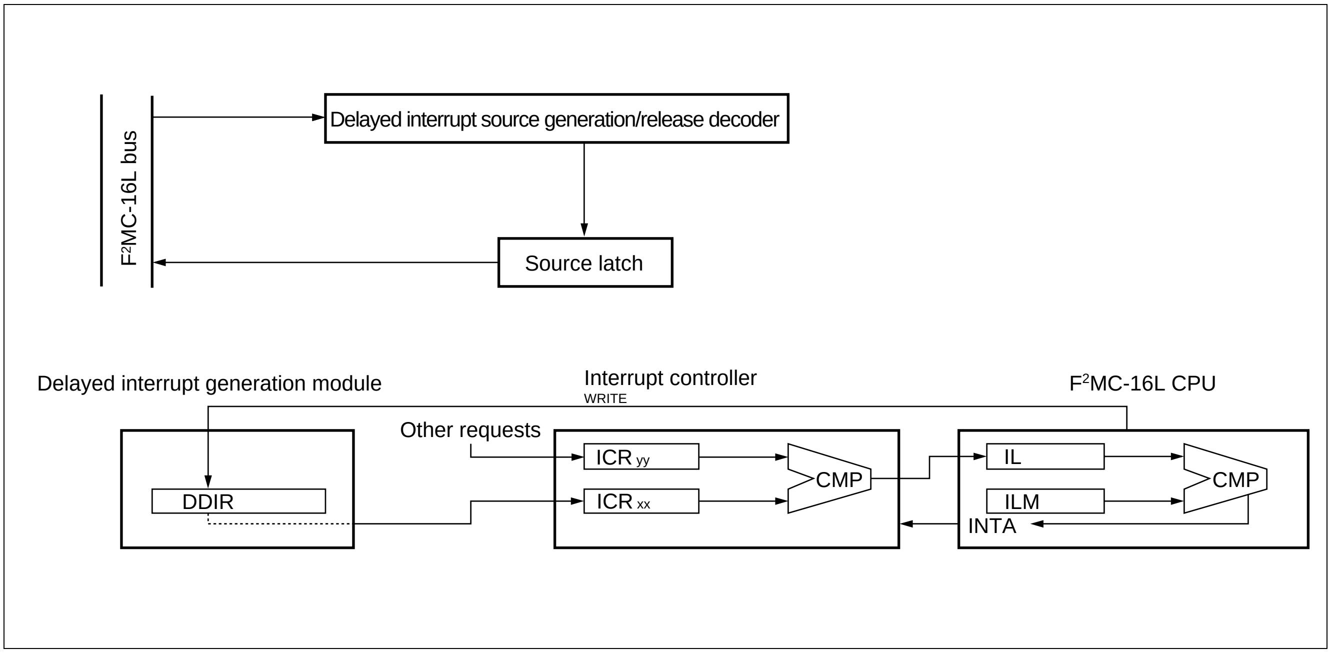
The delayed interrupt generation module generates task switching interrupts. This module can be used to generate/cancel interrupt requests to the F^{2}MC-16L CPU by software.

(1) Register Configuration

bit	15	14	13	12	11	10	9	8
Address: 00009F$_H$	—	—	—	—	—	—	—	R0

Delayed interrupt source generation/release register (DIRR)

(2) Block Diagram



12. Low-power Consumption Controller (CPU Intermittent Operation Function, Oscillation Stabilization Delay Time, Clock Multiplier Function)

The following are the operating modes: PLL clock mode, PLL sleep mode, PLL watch mode, Pseudo-watch mode, main clock mode, main sleep mode, main watch mode, main stop mode, sub clock mode, sub sleep mode, sub watch mode, sub stop mode, and hardware standby mode. Aside from the PLL clock mode, all of the other operating modes are low-power consumption modes.

In main clock mode and main sleep mode, the main clock (main OSC oscillation clock) and the sub clock (sub OSC oscillation clock) operate. In these modes, the main clock divided by 2 is used as the operation clock, the sub clock (sub OSC oscillation clock) is used as the timer clock, and the PLL clock (VCO oscillation clock) is stopped.

In sub clock mode and sub sleep mode, only the sub clock operates. In these modes, the sub clock is used as the operation clock, and the main clock and PLL clock are stopped.

In PLL sleep mode and main sleep mode, only the CPU's operation clock is stopped; all clocks other than the CPU clock operate.

In Pseudo-watch mode, only the watch timer and timebase timer operate.

In PLL watch mode, main watch mode, and sub watch mode, only the watch timer operates. In this mode, only the sub clock is used for operation, while the main clock and the PLL clock are stopped (the difference between the PLL watch mode, the main watch mode and the sub watch mode is that it resumes operation after an interrupt in the PLL clock mode, the main clock modes and the sub clock mode respectively, and there is no difference in the watch mode).

The main stop mode, sub stop mode, and hardware standby mode stop oscillation, making it possible to retain data while consuming the least amount of power. (The difference between the main stop mode and the sub stop mode is that it resumes operation in the main clock mode and the sub clock mode respectively, and there is no difference in the stop mode.)

The CPU intermittent operation function intermittently runs the clock supplied to the CPU when accessing registers, on-chip memory, on-chip resources, and the external bus. Processing is possible with lower power consumption by reducing the execution speed of the CPU while supplying a hig-speed clock and using on-chip resources.

The PLL clock multiplier can be selected as either 2, 4, 6, or 8 by setting the CS1 and CS0 bits. These clocks are divided by 2 to be used as a machine clock.

The WS1 and WS0 bits can be used to set the main clock oscillation stabilization delay time for when stop mode and hardware standby mode are woken up.

(1) Register Configuration

Address: 0000A0H

bit	7	6	5	4	3	2	1	0
	STP	SLP	SPL	RST	TMD	CG1	CG0	SSR

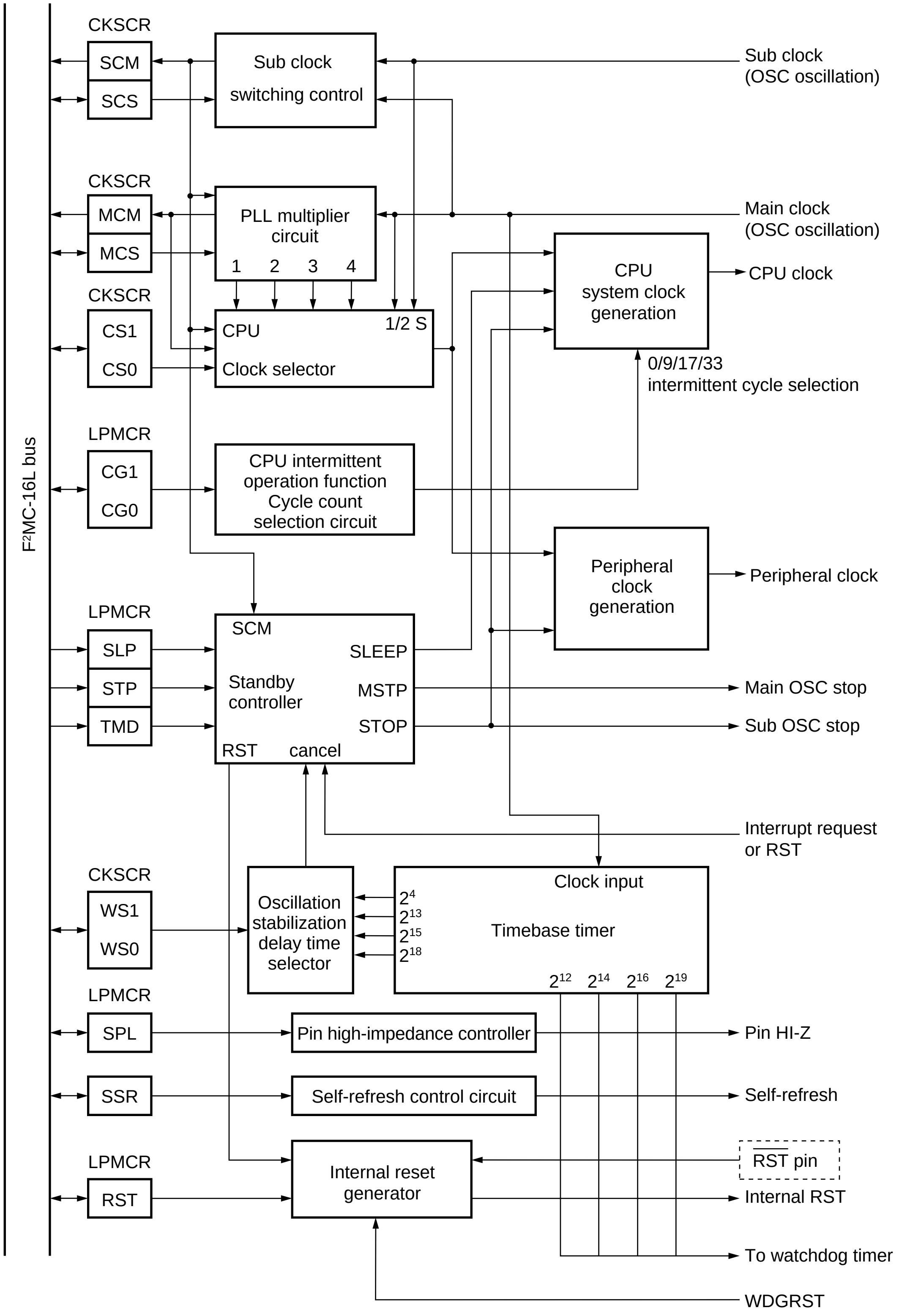
Low-power consumption mode control register (LPMCR)

Address: 0000A1H

bit	15	14	13	12	11	10	9	8
	SCM	MCM	WS1	WS0	SCS	MCS	CS1	CS0

Clock selection register (CKSCR)

(2) Block Diagram



■ ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Ratings

(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	
	AVRH*[1] AVRL	V_{SS} – 0.3	V_{SS} + 7.0	V	
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 output current	I_{OL}	—	15	mA	
"L" level total output current	ΣI_{OL}	—	50	mA	
"H" level output current	I_{OH}	—	–4	mA	
"H" level total output current	ΣI_{OH}	—	–48	mA	
Power consumption	P_d	—	+400	mW	
Operating temperature	T_A	–40	+85	°C	
Storage temperature	T_{STG}	–55	+150	°C	

*1: AV_{CC}, AVRH and AVRL must not exceed V_{CC}. In addition, AVRL must not exceed AVRH.
*2: V_I or V_O must not exceed V_{CC} + 0.3 V.

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	Value		Unit	Remarks
		Min.	Max.		
Power supply voltage	V_{CC}	4.0	5.5	V	Normal operation
		2.7	5.5	V	Maintaining the stop status
"H" level input voltage	V_{IH}	0.7 V_{CC}	V_{SS} + 0.3	V	Except V_{IHS}
	V_{IHS}	0.8 V_{CC}	V_{SS} + 0.3	V	Hysteresis inputs
"L" level input voltage	V_{IL}	V_{SS} – 0.3	0.8	V	Except V_{ILS}
	V_{ILS}	V_{SS} – 0.3	0.2 V_{CC}	V	Hysteresis inputs
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} = 4.0 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Condition	Value			Unit	Remarks
				Min.	Typ.	Max.		
“H” level output voltage	V_{OH}	—	V_{CC} = 4.5 V I_{OH} = –4.0 mA	V_{CC} – 0.5	—	—	V	
“L” level output voltage	V_{OL}	—	V_{CC} = 4.5 V I_{OH} = –4.0 mA	—	—	0.4	V	
Input leakage current	I_{IL}	—	V_{CC} = 5.5 V $<V_{SS} <V_I <V_{CC}$	–10	—	10	µA	
Pull-up resistor	R	—	—	22	—	110	kΩ	
Power supply current	I_{CC}	V_{CC}	—	—	40	80	mA	In 12 MHz operation
	I_{CC}			—	30	60	mA	In 8 MHz operation
	I_{CC}			—	15	40	mA	In 4 MHz operation
	I_{CCS}			—	10	40	mA	In 12 MHz sleep
	I_{CCL}			—	6	10	mA	In 32 KHz sub operation
	I_{CCT}			—	50	200	µA	In 32 KHz watch mode
	I_{CCH}			—	1	10	µA	In stop mode
LCD voltage division resistor	R_{LCD}	—	Between V_{CC} and V0, V_{CC} = 5.0 V	300	500	750	kΩ	
COM0 to COM3 output impedance	R_{VCOM}	—	V1 – V3 = 5.0 V	—	—	2.5	kΩ	
SEG 0 to SEG31 output impedance	R_{VSEG}	—	V1 – V3 = 5.0 V	—	—	15	kΩ	
LCD leakage current	I_{LCDL}	—	—	–10	—	10	µA	
Input capacitance	C_{IN}	Except V_{CC}, V_{SS}	—	—	10	—	pF	
Open-drain output leakage current	I_{leak}	Open-drain pin	—	—	0.1	10	µA	

4. AC Characteristics

(1) Clock Timing

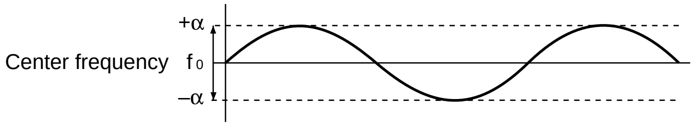
- **When V_{CC} = 4.0 V to 5.5 V**

(V_{CC} = 4.0 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Source oscillation frequency	F_C	X0, X1	—	3	24	MHz	
Source oscillation cycle time	t_C	X0, X1	—	41.66	333	ns	
Frequency fluctuation ratio*1 (when locked)	Δf	—	—	—	3	%	
Input clock pulse width	P_{WH}, P_{WL}	X0	—	12	—	ns	Use duty ratio of 30 to 70% as a guide
Input clock rising/falling time	t_{cr}, t_{cf}	X0	—	—	5	ns	
Internal operating clock frequency	f_{CP}	—	—	32 K*2	12 M	Hz	
Internal operating clock cycle time	t_{CP}	—	—	83.5	31250	ns	

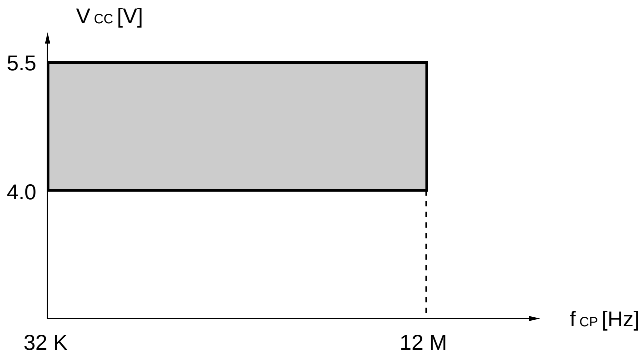
*1: The frequency fluctuation ratio indicates the maximum fluctuation ratio from the set center frequency while locked with multiply.

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

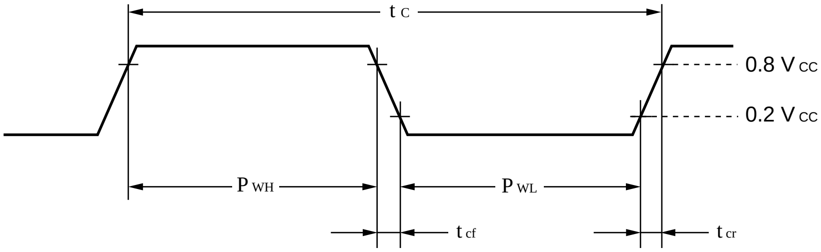


*2: 32 KHz operation means sub operation.

- **Relationship between Operating Clock Frequency and Power Supply Voltage**

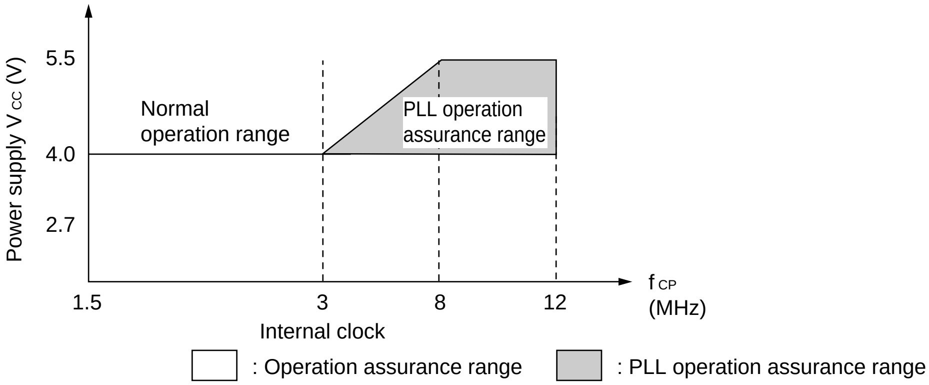


- **Clock Timing**

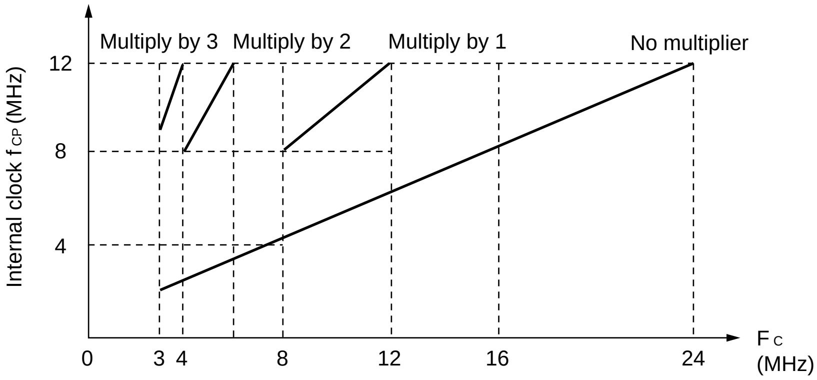


- **PLL Operation Assurance Range**

Relationship between internal operation clock frequency and power supply voltage



Relationship between source oscillation frequency, internal operating clock frequency

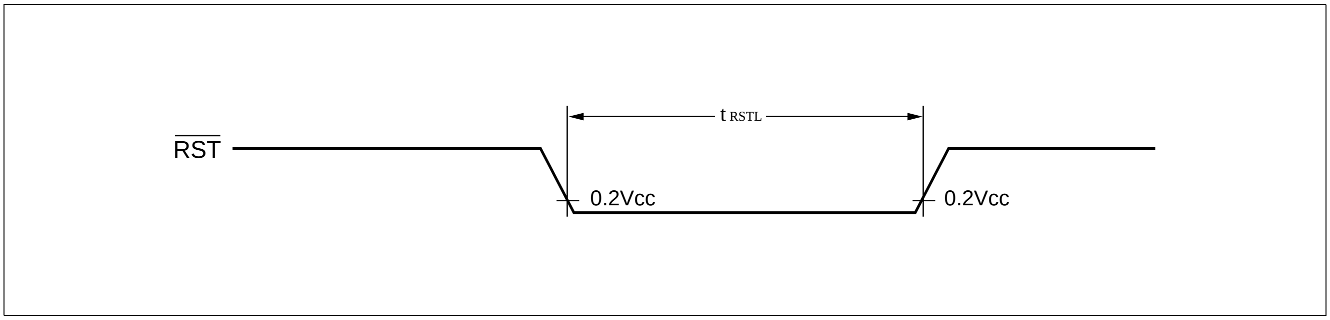


Source oscillation clock Fc (MHz)

(2) Reset Input Timing

(V_{CC} = 4.0 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

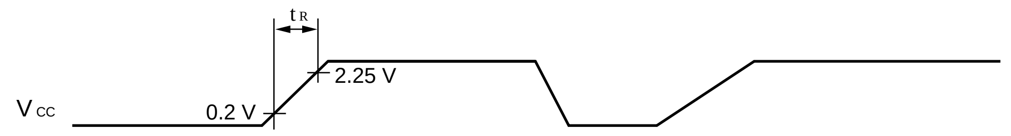
Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Reset input time	t_{RSTL}	$\overline{\text{RST}}$	—	4 t_C	—	ns	



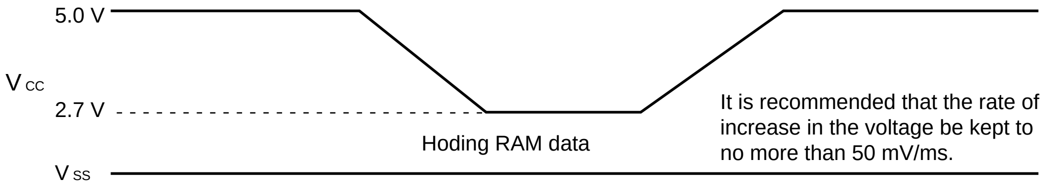
(3) Power-on Reset

(V_{CC} = 4.0 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Power supply rising time	t_R	V_{CC}	—	—	30	ms	
Power supply cut-off time	t_{OFF}	V_{CC}	—	1	—	ms	



If power supply voltage needs to be changed in the course of operation, a smooth voltage rise is recommended by suppressing the voltage variation as shown below. Also, do not use the PLL clock when varying the voltage.
However, the supply voltage can be changed when using the PLL clock if the voltage drops by less than 1 mV/s.



It is recommended that the rate of increase in the voltage be kept to no more than 50 mV/ms.

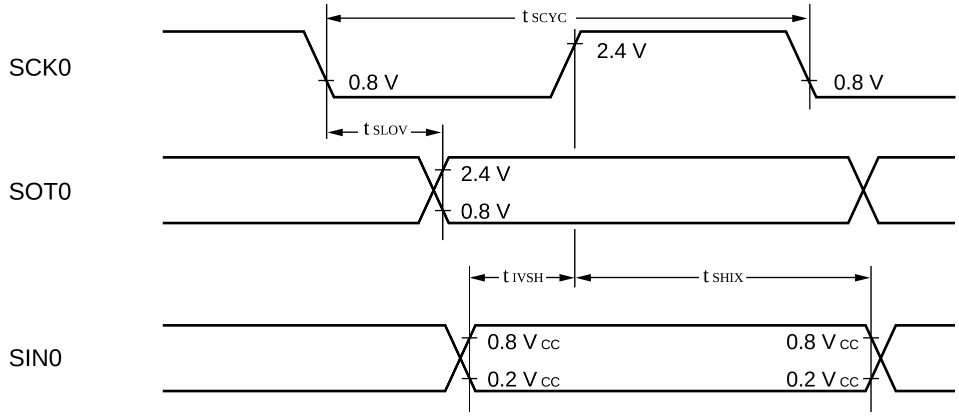
(4) UART Timing

(V_{CC} = 4.0 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

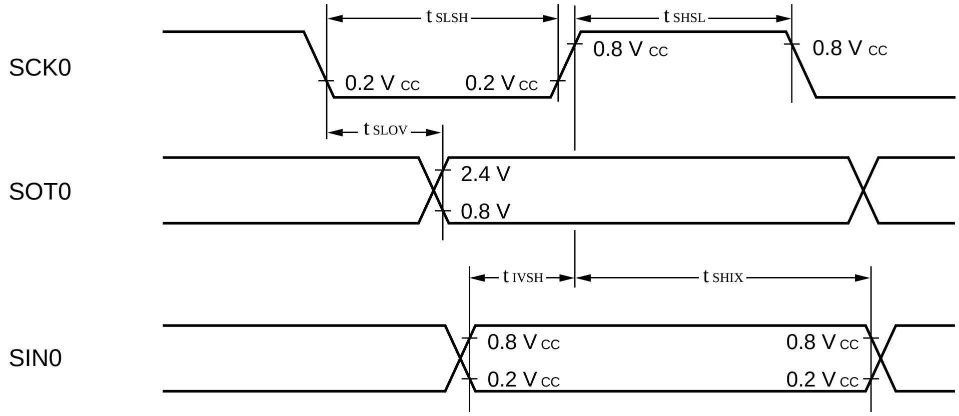
Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Serial clock cycle time	t_{SCYC}	—	For internal shift clock mode output pin, C_L = 80 pF+1 TTL	8 t_{CP}	—	ns	
SCK0 ↓ → SOT0 delay time	t_{SLOV}	—		–80	80	ns	
Valid SIN0 → SCK0 ↑	t_{IVSH}	—		100	—	ns	
SCK0 ↑ → Valid SIN0 hold time	t_{SHIX}	—		60	—	ns	
Serial clock "H" pulse width	t_{SHSL}	—	For external shift clock mode output pin, C_L = 80 pF+1 TTL	4 t_{CP}	—	ns	
Serial clock "L" pulse width	t_{SLSH}	—		4 t_{CP}	—	ns	
SCK0 ↓ → SOT0 delay time	t_{SLOV}	—		—	150	ns	
Valid SIN0 → SCK0 ↑	t_{IVSH}	—		60	—	ns	
SCK0 ↑ → Valid SIN0 hold time	t_{SHIX}	—		60	—	ns	

Notes:
- These are the AC characteristics for CLK synchronous mode.
- C_L is the load capacitance added to pins during testing.
- t_{CP} is the internal operating clock cycle time (unit: ns).
- The values in the table are target values.

- **Internal Shift Clock Mode**



- **External Shift Clock Mode**



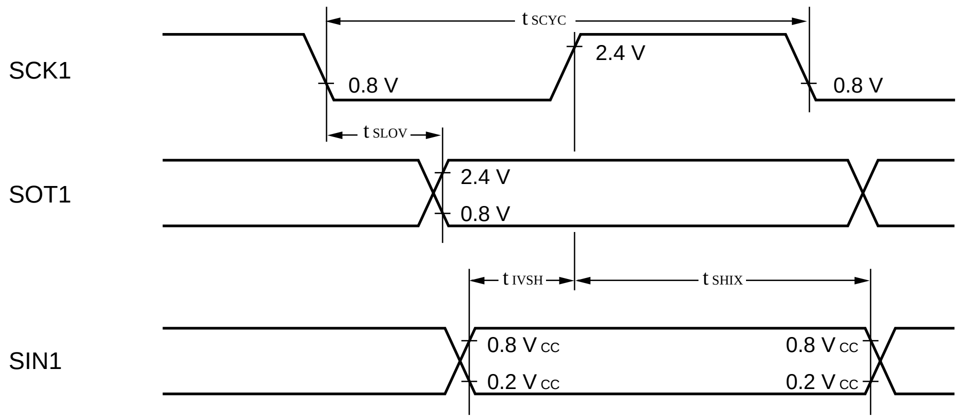
(5) Extended Serial I/O Timing

(V_{CC} = 4.0 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

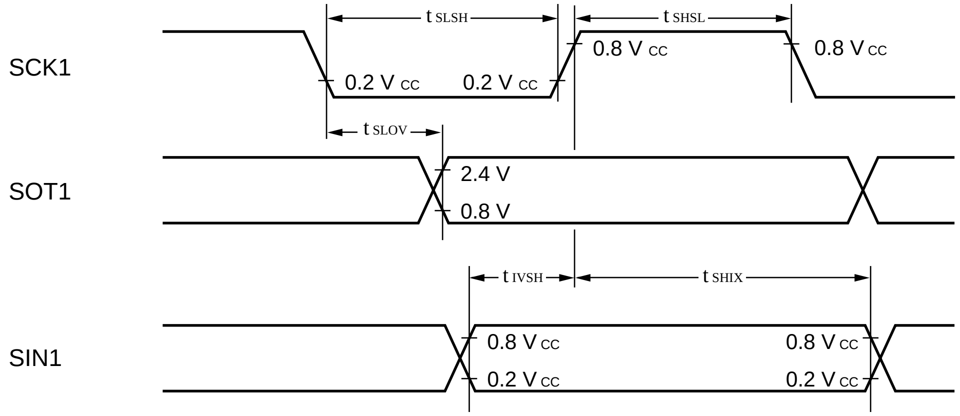
Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Serial clock cycle time	t_{SCYC}	—	—	8 t_{XMCYL}	—	ns	For internal shift clock mode output pin, C_L = 80 pF+1 TTL
SCK1 ↓ → SOT1 delay time	t_{SLOV}	—	V_{CC} = 5.0 V ±10%	—	80	ns	
Valid SIN1 → SCK1 ↑	t_{IVSH}	—	—	1 t_{XMCYL}	—	ns	
SCK1 ↑ → Valid SIN1 hold time	t_{SHIX}	—	—	1 t_{XMCYL}	—	ns	
Serial clock "H" pulse width	t_{SHSL}	—	V_{CC} = 5.0 V ±10%	230	—	ns	For external shift clock mode output pin, C_L = 80 pF Max. 2 MHz
Serial clock "L" pulse width	t_{SLSH}	—	V_{CC} = 5.0 V ±10%	230	—	ns	
SCK1 ↓ → SOT1 delay time	t_{SLOV}	—	—	2 t_{XMCYL}	—	ns	
Valid SIN1 → SCK1 ↑	t_{IVSH}	—	—	1 t_{XMCYL}	—	ns	
SCK1 ↑ → Valid SIN1 hold time	t_{SHIX}	—	—	1 t_{XMCYL}	—	ns	

Notes: • C_L is the load capacitance added to pins during testing.
• t_{XMCYL} is the internal operation clock cycle time (unit: ns).

- **Internal Shift Clock Mode**



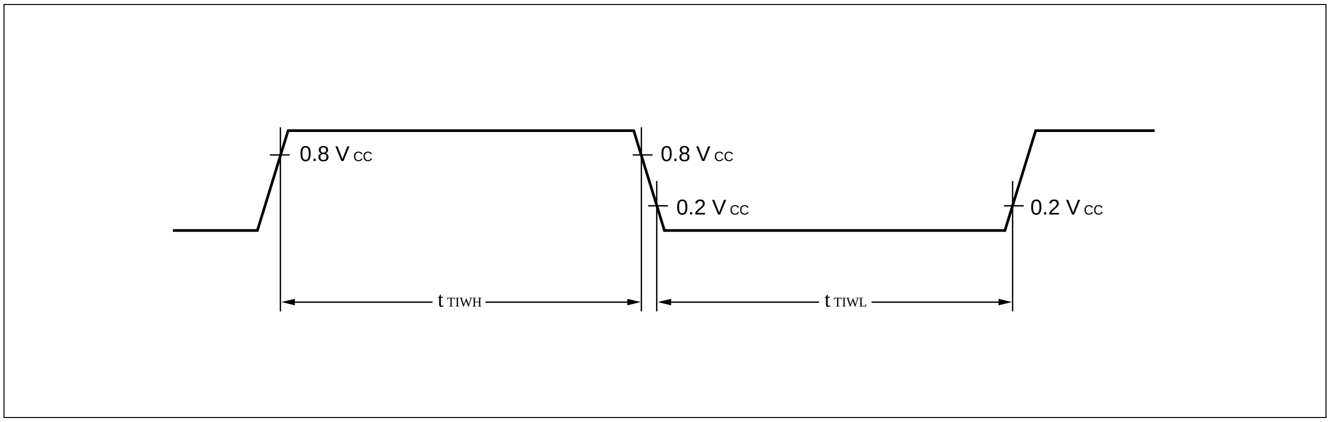
- **External Shift Clock Mode**



(6) Timer Input Timing

(V_{CC} = 4.0 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

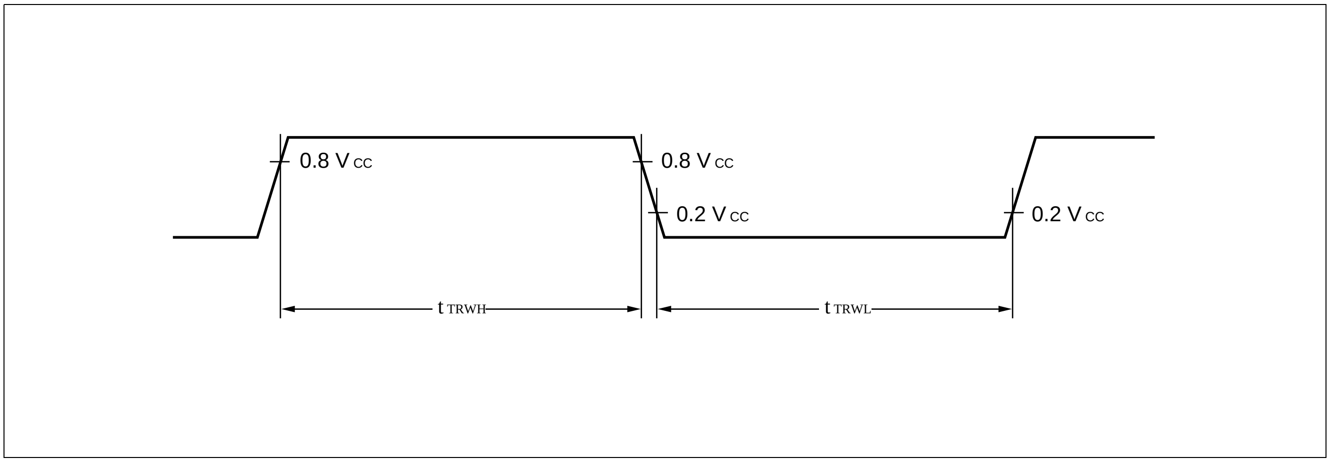
Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Input pulse width	t_{TIWH} t_{TIWL}	TIO0 to TIO2	—	4 t_{CP}	—	ns	



(7) Trigger Input Timing

(V_{CC} = 4.0 V to +5.5 V, V_{SS} = 0.0 V, T_A = –40°C to +85°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Trigger input width	t_{TRWH} t_{TRWL}	$\overline{\text{ADT}}$ TRG	—	4 t_{CP}	—	ns	A/D trigger



5. A/D Converter Electrical Characteristics

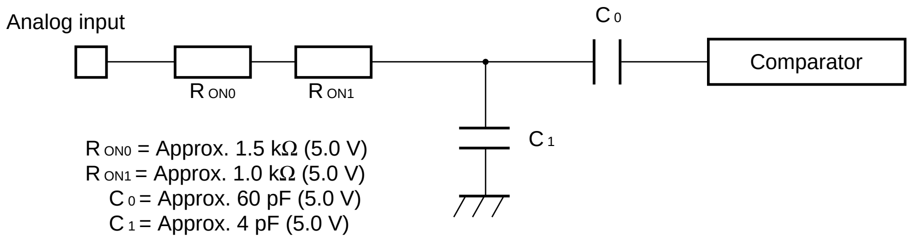
($AV_{CC} = V_{CC} = +2.7$ V to 5.5 V, $AV_{SS} = V_{SS} = 0.0$ V, +2.7 V ≤ AVRH – AVRL, $T_A = –40$°C to +85°C)

Parameter	Symbol	Pin name	Value			Unit
			Min.	Typ.	Max.	
Resolution	—	—	—	10	10	bit
Total error	—	—	—	—	±3.0	LSB
Linearity error	—	—	—	—	±1.5	LSB
Differential linearity error	—	—	—	—	±1.5	LSB
Zero transition voltage	V_{OT}	AN0 to AN3	–1.5	+0.5	+2.5	LSB
Full-scale transition voltage	V_{FST}	AN0 to AN3	AVRH – 3.5	AVRL – 1.5	AVRH + 0.5	LSB
Conversion time	—	—	8.16	—	—	µs
Analog port input current	I_{AIN}	AN0 to AN3	—	—	10	µA
Analog input voltage	V_{AIN}	AN0 to AN3	AVRL	—	AVRH	V
Reference voltage	—	AVRH	AVRL	—	AV_{CC}	V
	—	AVRL	—	—	AVRH	V
Power supply current	I_A	AV_{CC}	—	5	—	mA
	I_{AH}	AV_{CC}	—	—	5*	µA
Reference voltage supply current	I_R	AV_{CC}	—	200	—	µA
	I_{RH}	AV_{CC}	—	—	5*	µA
Interchannel disparity	—	AN0 to AN3	—	—	4	LSB

* : Current when the A/D converter is not operating and the CPU is stopped (when $V_{CC} = AV_{CC} =$ AVRH = +5.5 V)

Notes:
- The smaller | AVRH – AVRL |, the greater the error would become relatively.
- The output impedance of the external circuit for the analog input must satisfy the following conditions: The output impedance of the external circuit should be less than approximately 7 kΩ.
- If the output impedance of the external circuit is too high, an analog voltage sampling time might be insufficient (sampling time = 5 µs @ at a machine clock of 12 MHz).

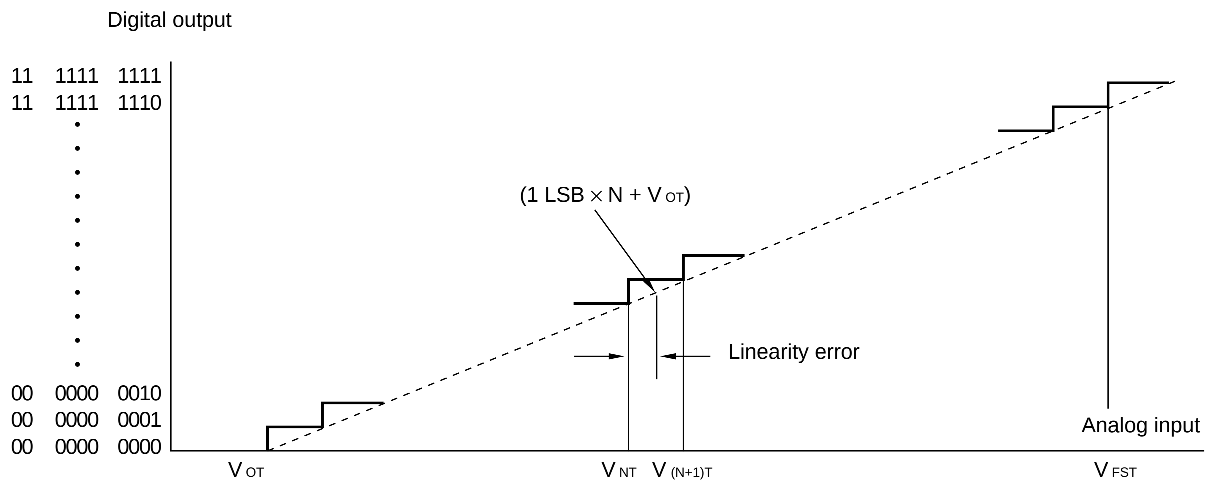
- **Analog Input Circuit Model Diagram**


Analog input
R_{ON0}
R_{ON1}
C_0
Comparator
C_1
R_{ON0} = Approx. 1.5 kΩ (5.0 V)
R_{ON1} = Approx. 1.0 kΩ (5.0 V)
C_0 = Approx. 60 pF (5.0 V)
C_1 = Approx. 4 pF (5.0 V)

Note: Use the values shown as guides only.

6. A/D Converter Glossary

- Resolution
  Analog changes that are identifiable with the A/D converter.
  If the resolution is 10 bits, the analog voltage can be resolved into 2^{10} = 1024 steps.
- Total error
  The deviation between the actual and logic value attributable to offset error, gain error, non-linearity error, and noise.
- Linearity error
  The deviation between the actual conversion characteristic of the device and the line linking the zero transition point ("00 0000 0000" ↔ "00 0000 0001") and the full scale transition point ("11 1111 1110" ↔ "11 1111 1111").
- Differential linearity error
  The deviation of input voltage needed to change the output code by 1 LSB from the theoretical value



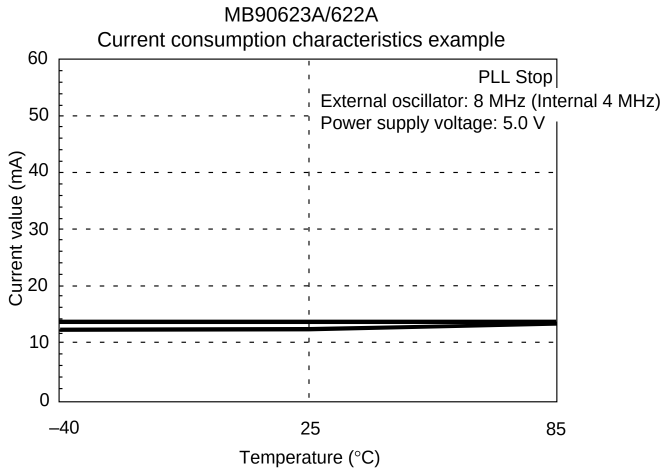
$$1\text{ LSB} = \frac{V_{FST} - V_{OT}}{1022}$$

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

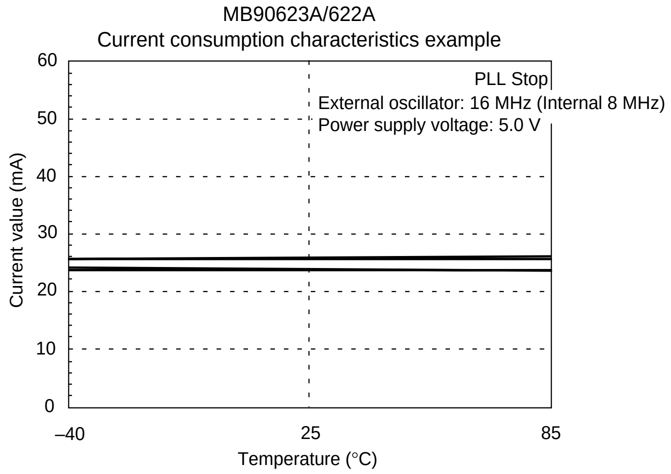
$$\text{Differential linearity error} = \frac{V_{(N+1)T} - V_{NT}}{1\text{ LSB}} - 1 \quad [\text{LSB}]$$

■ EXAMPLE CHARACTERISTICS

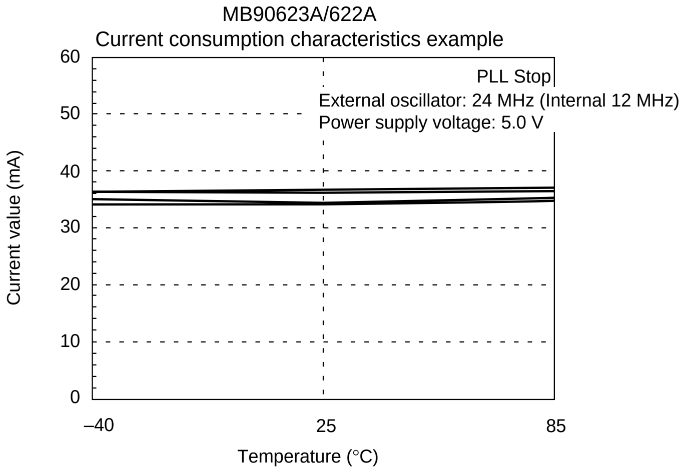
Power supply current vs temperature characteristics example



Power supply current vs temperature characteristics example

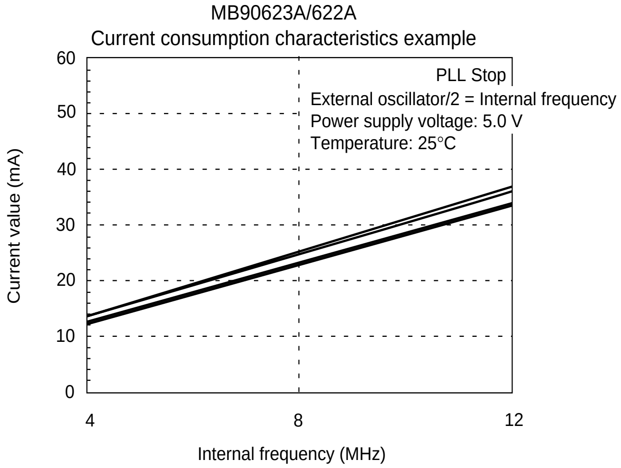


Power supply current vs temperature characteristics example

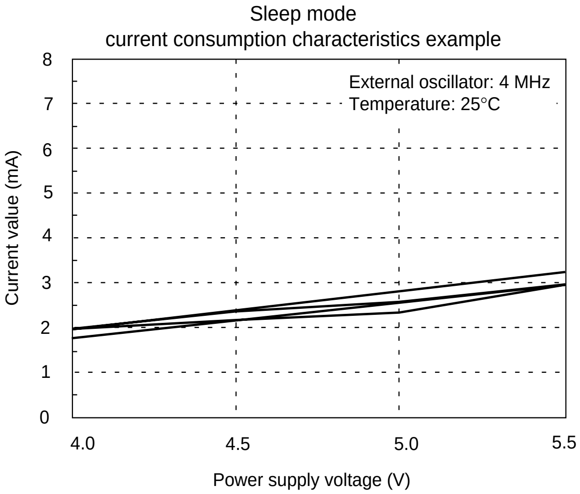


(Continued)

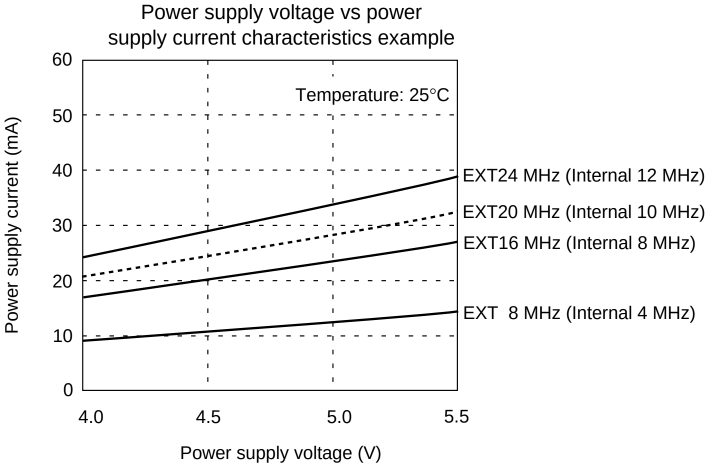
Operation frequency vs power supply current characteristics example



Sleep mode power supply current characteristics example

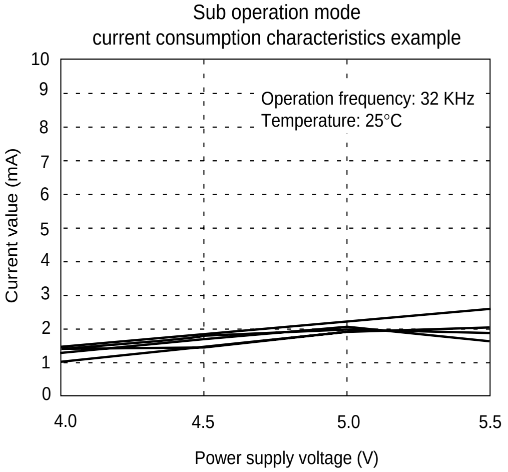


Power supply voltage vs power supply current characteristics example

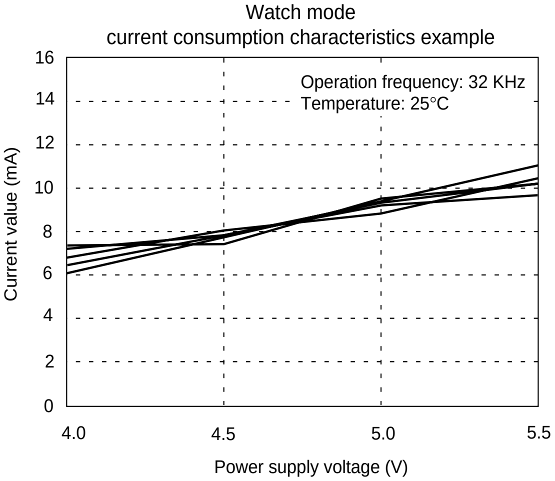


(Continued)

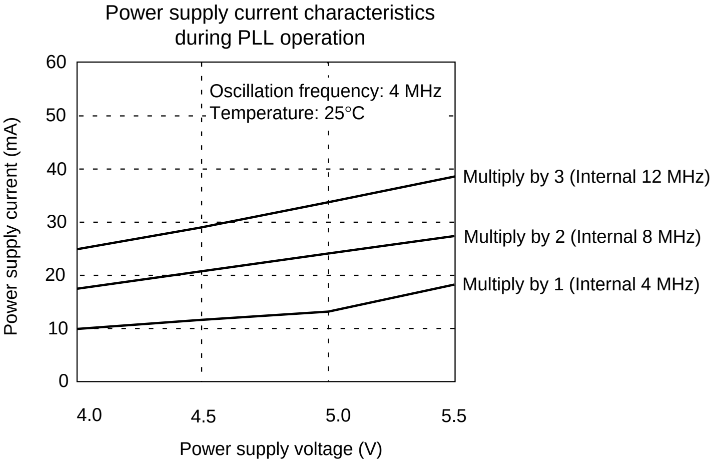
Sub operation mode power supply current characteristics example



Watch mode power supply current characteristics example



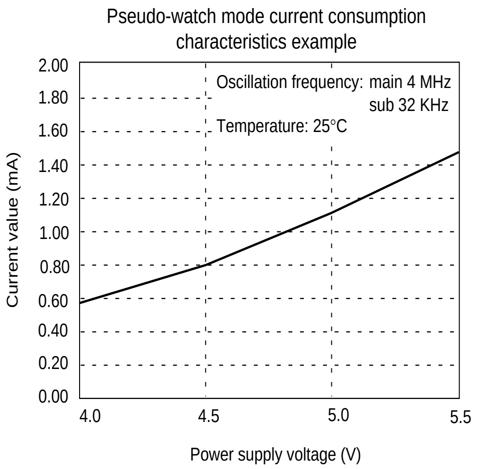
Power supply current characteristics during PLL operation



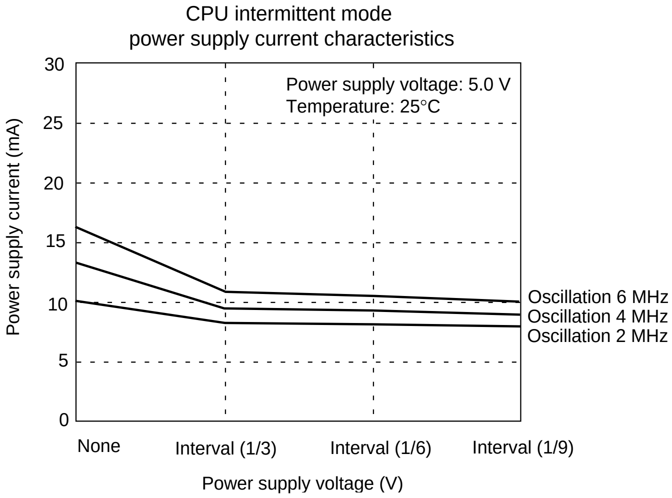
(Continued)

(Continued)

Pseudo-watch mode power supply current characteristics example



CPU intermittent mode power supply current characteristics



■ 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)

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	R0	RW0	RL0	Register direct “ea” corresponds to byte, word, and long-word types, starting from the left	—
01	R1	RW1	(RL0)		
02	R2	RW2	RL1		
03	R3	RW3	(RL1)		
04	R4	RW4	RL2		
05	R5	RW5	(RL2)		
06	R6	RW6	RL3		
07	R7	RW7	(RL3)		
08	@RW0			Register indirect	0
09	@RW1				
0A	@RW2				
0B	@RW3				
0C	@RW0 +			Register indirect with post-increment	0
0D	@RW1 +				
0E	@RW2 +				
0F	@RW3 +				
10	@RW0 + disp8			Register indirect with 8-bit displacement	1
11	@RW1 + disp8				
12	@RW2 + disp8				
13	@RW3 + disp8				
14	@RW4 + disp8				
15	@RW5 + disp8				
16	@RW6 + disp8				
17	@RW7 + disp8				
18	@RW0 + disp16			Register indirect with 16-bit displacement	2
19	@RW1 + disp16				
1A	@RW2 + disp16				
1B	@RW3 + disp16				
1C	@RW0 + RW7			Register indirect with index	0
1D	@RW1 + RW7			Register indirect with index	0
1E	@PC + disp16			PC indirect with 16-bit displacement	2
1F	addr16			Direct address	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 register accesses for each type of addressing
		Number of execution cycles 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) [28 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)

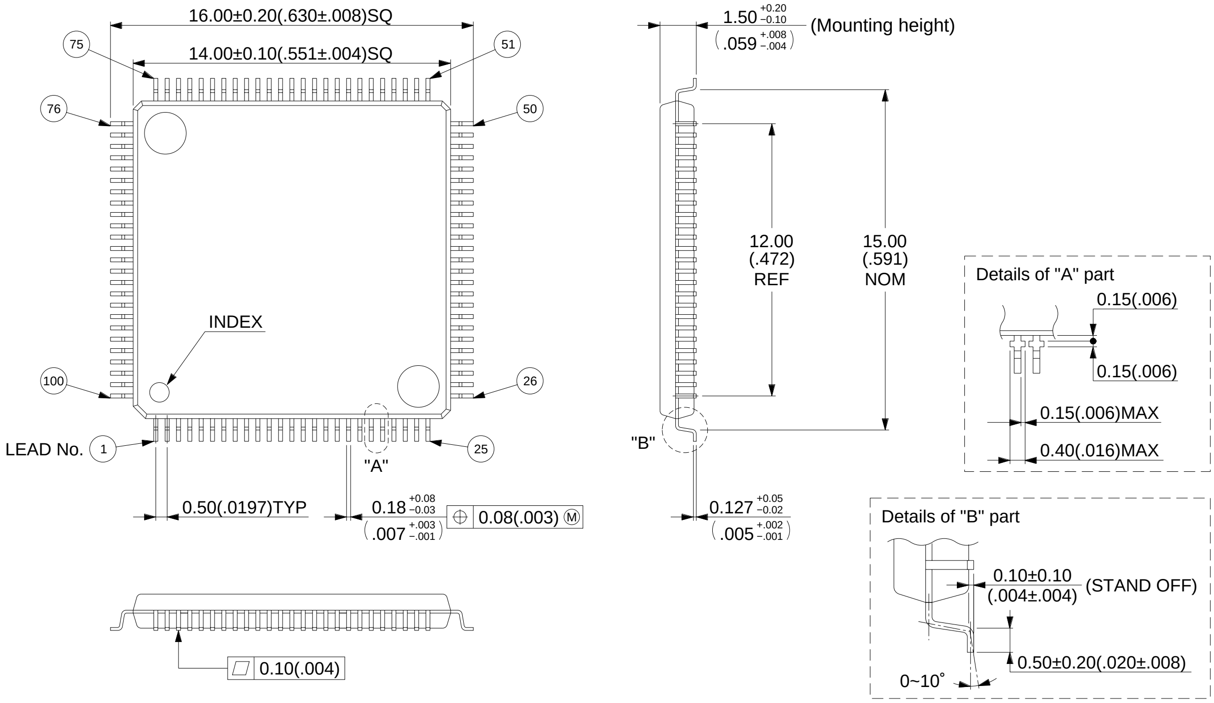
Note: For an explanation of "(a)" to "(d)", refer to Table 4, "Number of Execution Cycles for Each Type of Addressing," and Table 5, "Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles."

■ ORDERING INFORMATION

Model	Package	Remarks
MB90622PFV MB90623PFV MB90P623PFV	100-pin Plastic LQFP (FPT-100P-M05)	

■ PACKAGE DIMENSIONS

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



Dimension 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

F9704

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.