FUJITSU SEMICONDUCTOR
DATA SHEET

DS07-13711-1E

16-Bit Original Microcontroller

CMOS

F²MC-16LX MB90420G/5G (A) Series

MB90423G/423GA/F423G/F423GA/V420G
MB90427G/427GA/428G/428GA/F428G/F428GA

■ DESCRIPTIONS

The FUJITSU MB90420G/5G (A) Series is a 16-bit general purpose high-capacity microcontroller designed for vehicle meter control applications etc.

The instruction set retains the same AT architecture as the FUJITSU original F²MC-8L and F²MC-16L series, with further refinements including high-level language instructions, expanded addressing mode, enhanced (signed) multipler-divider computation and bit processing.

In addition, A 32-bit accumulator is built in to enable long word processing.

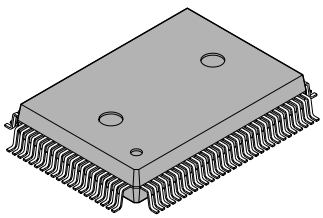
■ FEATURES

- 16-bit input capture (4 channels)
  Detects rising, falling, or both edges.
  16-bit capture register × 4
  Pin input edge detection latches the 16-bit free-run timer counter value, and generates an interrupt request.
- 16-bit reload timer (2 channels)
  16-bit reload timer operation (select toggle output or one-shot output)
  Event count function selection provided

(Continued)

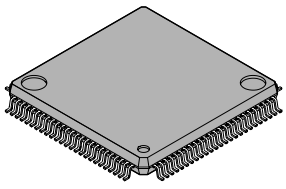
■ PACKAGES

Plastic QFP, 100-pin



(FPT-100P-M06)

Plastic LQFP, 100-pin



(FPT-100P-M05)

- Clock timer (main clock)
  Operates directly from oscillator clock.
  Compensates for oscillator deviation
  Read/write enabled second/minute/hour register
  Signal interrupt
- 16-bit PPG (3 channels)
  Output pins (3) , external trigger input pin (1)
  Output clock frequencies : f_{CP}, $f_{CP}/2^2$, $f_{CP}/2^4$, $f_{CP}/2^6$
- Delay interrupt
  Generates interrupt for task switching.
  Interruptions to CPU can be generated/deleted by software setting.
- External interrupts (8 channels)
  8-channel independent operation
  Interrupt source setting available : “L” to “H” edge/ “H” to “L” edge/ “L” level/ “H” level.
- A/D converter
  10-bit or 8-bit resolution × 8 channels (input multiplexed)
  Conversion time : 6.13 µs or less (at f_{CP} = 16 MHz)
  External trigger startup available (P50/INT0/ADTG)
  Internal timer startup available (16-bit reload timer 1)
- UART (2 channels)
  Full duplex double buffer type
  Supports asynchronous/synchronous transfer (with start/stop bits)
  Internal timer can be selected as clock (16-bit reload timer 0)
  Asynchronous : 4808 bps, 5208 bps, 9615 bps, 10417 bps, 19230 bps, 38460 bps, 62500 bps, 500000 bps
  Synchronous : 500 Kbps, 1Mbps, 2Mbps (at f_{CP} = 16 MHz)
- CAN interface *1
  Conforms to CAN specifications version 2.0 Part A and B.
  Automatic resend in case of error.
  Automatic transfer in response to remote frame.
  16 prioritized message buffers for data and messages for data and ID
  Multiple message support
  Receiving filter has flexible configuration : All bit compare/all bit mask/two partial bit masks
  Supports up to 1 Mbps
  CAN WAKEUP function (connects RX internally to INT0)
- LCD controller/driver (1 channel)
  Segment driver and command driver with direct LCD panel (display) drive capability
- Low voltage/Program Looping detect reset *2
  Automatic reset when low voltage is detected
  Program Looping detection function
- Stepping motor controller (4 channels)
  High current output for all channels × 4
  Synchronized 8/10-bit PWM for all channels × 2
- Sound generator
  8-bit PWM signal mixed with tone frequency from 8-bit reload counter.
  PWM frequencies : 62.5 kHz, 31.2 kHz, 15.6 kHz, 7.8 kHz (at f_{CP} = 16MHz)
  Tone frequencies : 1/2 PWM frequency, divided by (reload frequency +1)

(Continued)

(Continued)

- Input/output ports
  Push-pull output and Schmitt trigger input
  Programmable in bit units for input/output or peripheral signals.
- Flash memory
  Supports automatic programming, Embeded Algorithm™, write/erase/erase pause/erase resume instructions
  Flag indicates algorithm completion
  Minato Electronics flash writer
  Boot block configuration
  Erasable by blocks
  Block protection by external programming voltage

*1 : MB90420G (A) series has 2 channels built-in, MB90425G (A) series has 1 channel built-in

*2 : Built-in to MB90420GA/5GA series only. Not built-in to MB90420G/5G series.

■ PRODUCT LINEUP

• MB90420G (A) Series

Part number / Parameter	MB90V420G	MB90F423G *1	MB90F423GA *1	MB90423G *2	MB90423GA *2
Configuration	Evaluation model	Flash ROM model		Mask ROM model	
CPU	F²MC-16LX CPU				
System clock	On-chip PLL clock multiplier type (× 1, × 2, × 3, × 4, 1/2 when PLL stopped) Minimum instruction execution time 62.5 ns (with 4 MHz oscillator × 4)				
ROM	External	Flash ROM 128 KB		Mask ROM 128 KB	
RAM	6 KB	6 KB		6 KB	
CAN interface	2 channels				
Low voltage/ CPU operation detection reset	No	No	Yes	No	Yes
Packages	PGA-256	QFP100, LQFP100			
Emulator dedicated power supply*	No	—			

• MB90425G (A) Series

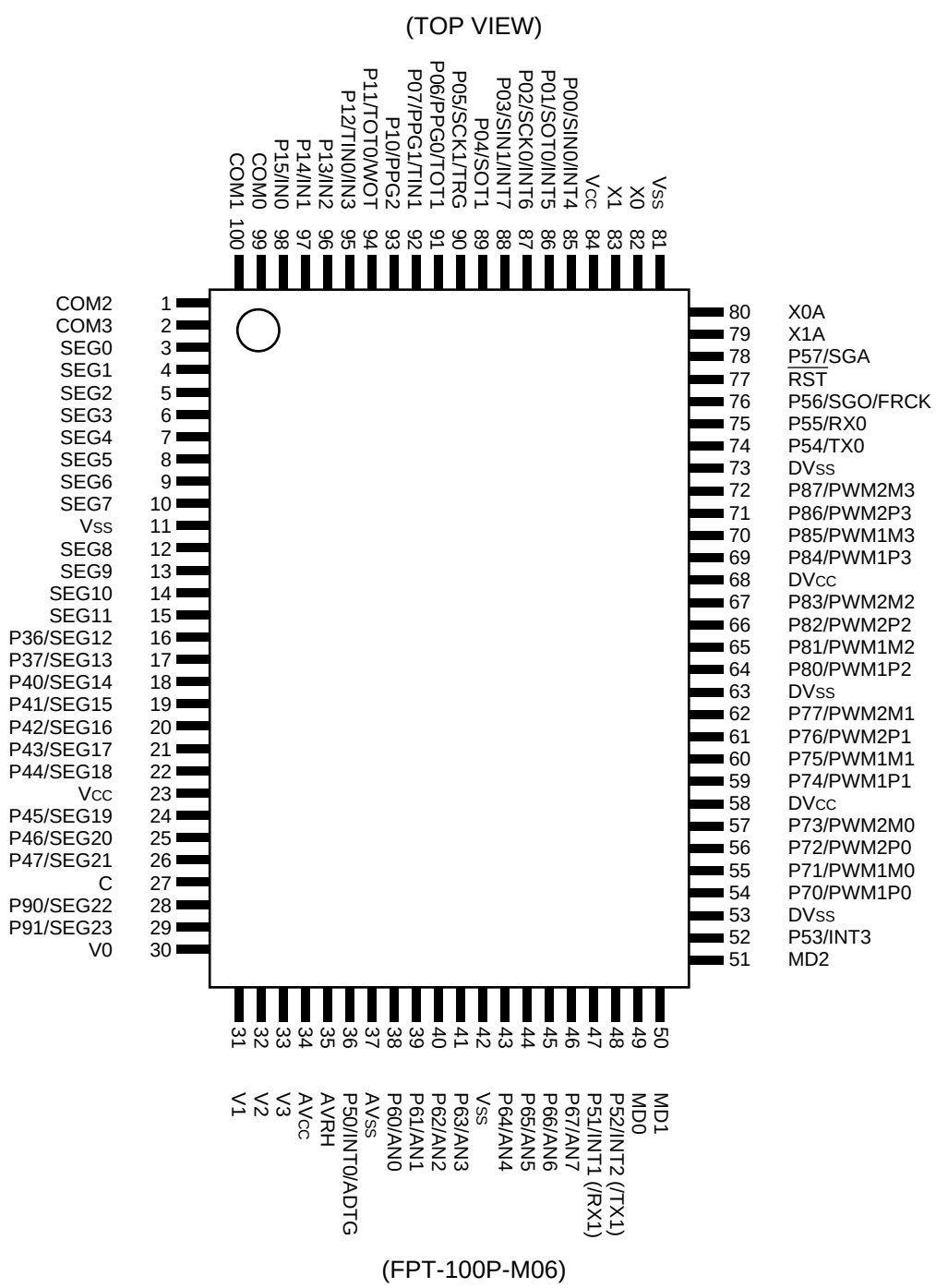
Part number / Parameter	MB90F428G	MB90F428GA	MB90427G*2	MB90427GA*2	MB90428G*1	MB90428GA*1
Configuration	Flash ROM model		Mask ROM model			
CPU	F²MC-16LX CPU					
System clock	On-chip PLL clock multiplier type (× 1, × 2, × 3, × 4, 1/2 when PLL stopped) Minimum instruction execution time 62.5 ns (with 4 MHz oscillator × 4)					
ROM	Flash ROM 128 KB		Mask ROM 64 KB		Mask ROM 128 KB	
RAM	6 KB		4 KB		6 KB	
CAN interface	1 channel					
Low voltage/ CPU operation detection reset	No	Yes	No	Yes	No	Yes
Packages	QFP100, LQFP100					
Emulator dedicated power supply*	—					

* : When used with evaluation pod MB2145-507, use DIP switch S2 setting. For details see the MB2145-507 Hardware Manual (2.7 “Emulator Dedicated Power Supply Pin”) .

*1 : Under development

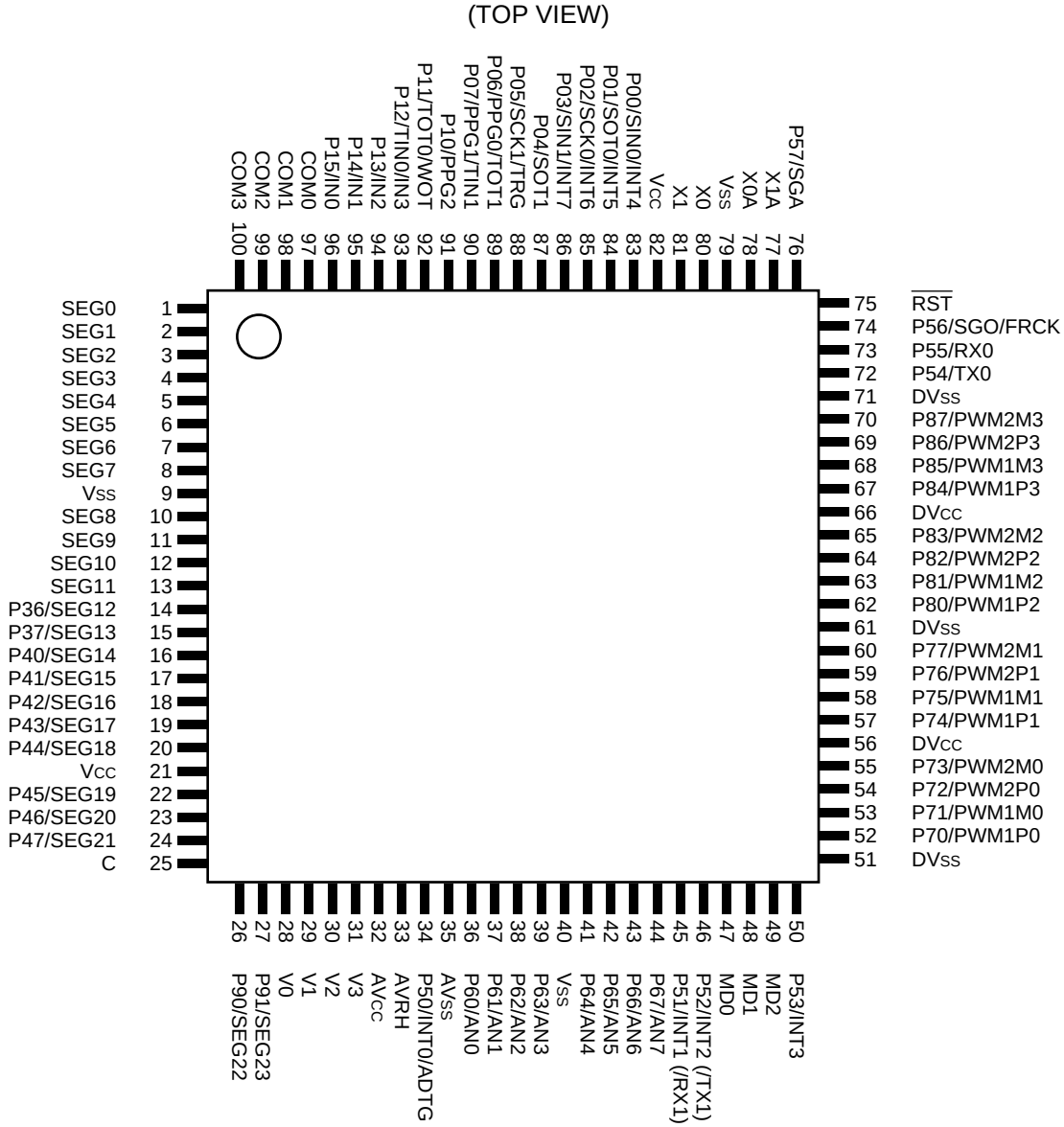
*2 : Planned

■ PIN ASSIGNMENTS



(Continued)

(Continued)



(FPT-100P-M05)

■ PIN DESCRIPTIONS

Pin no.		Symbol	Circuit type	Description
LQFP	QFP			
80	82	X0	A	High speed oscillator input pin.
81	83	X1		High speed oscillator output pin.
78	80	X0A	A	Low speed oscillator input pin. If no oscillator is connected, apply pull-down processing.
77	79	X1A		Low speed oscillator output pin. If no oscillator is connected, leave open.
75	77	$\overline{RST}$	B	Reset input pin.
83	85	P00	G	General purpose input/output port.
		SIN0		UART ch.0 serial data input pin.
		INT4		INT4 external interrupt input pin.
84	86	P01	G	General purpose input/output port.
		SOT0		UART ch.0 serial data output pin.
		INT5		INT5 external interrupt input pin.
85	87	P02	G	General purpose input/output port.
		SCK0		UART ch.0 serial clock input/output pin.
		INT6		INT6 external interrupt input pin.
86	88	P03	G	General purpose input/output port.
		SIN1		UART ch.1 serial data input pin.
		INT7		INT7 external interrupt input pin.
87	89	P04	G	General purpose input/output port.
		SOT1		UART ch.1 serial data output pin.
88	90	P05	G	General purpose input/output port.
		SCK1		UART ch.1 serial clock input/output pin.
		TRG		16-bit PPG ch.0-2 external trigger input pin.
89	91	P06	G	General purpose input/output port.
		PPG0		16-bit PPG ch.0 output pin.
		TOT1		16-bit reload timer ch.1 TOT output pin.
90	92	P07	G	General purpose input/output port.
		PPG1		16-bit PPG ch.1 output pin.
		TIN1		16-bit reload timer ch.1 TIN output pin.
91	93	P10	G	General purpose input/output port.
		PPG2		16-bit PPG ch.2 output pin.

(Continued)

Pin no.		Symbol	Circuit type	Description
LQFP	QFP			
92	94	P11	G	General purpose input/output port.
		TOT0		16-bit reload timer ch.0 TOT output pin.
		WOT		Real-time clock timer WOT output pin.
93	95	P12	G	General purpose input/output port.
		TIN0		16-bit reload timer ch.0 TIN output pin.
		IN3		Input capture ch.3 trigger input pin.
94 to 96	96 to 98	P13 to P15	G	General purpose input/output ports.
		IN2 to IN0		Input capture ch.0-2 trigger input pins.
97 to 100	99 to 100, 1 to 2	COM0 to COM3	I	LCD controller/driver common output pins.
1 to 8, 10 to 13	3 to 10, 12 to 15	SEG0 to SEG11	I	LCD controller/driver segment output pins.
14 to 15	16 to 17	P36 to P37	E	General purpose output ports.
		SEG12 to SEG13		LCD controller/driver segment output pins.
16 to 20, 22 to 24	18 to 22, 24 to 26	P40 to P47	E	General purpose input output ports.
		SEG14 to SEG21		LCD controller/driver segment output pins.
26 to 27	28 to 29	P90 to P91	E	General purpose input output ports.
		SEG22 to SEG23		LCD controller/driver segment output pins.
34	36	P50	G	General purpose input output ports.
		INT0		INT0 external interrupt input pin.
		ADTG		A/D converter external trigger input pin.
36 to 39, 41 to 44	38 to 41, 43 to 46	P60 to P67	F	General purpose input output ports.
		AN0 to AN7		A/D converter input pins.
45	47	P51	G	General purpose input output port.
		INT1		INT1 external interrupt input pin.
		(RX1 *)		CAN interface 1 RX intput pin.
46	48	P52	G	General purpose input output port.
		INT2		INT2 external interrupt input pin.
		(TX1 *)		CAN interface 1 TX output pin.
50	52	P53	G	General purpose input output port.
		INT3		INT3 external interrupt input pin.

* : MB90420G (A) series only.

(Continued)

Pin no.		Symbol	Circuit type	Description
LQFP	QFP			
52 to 55	54 to 57	P70 to P73	H	General purpose input output ports.
		PWM1P0 PWM1M0 PWM2P0 PWM2M0		Stepping motor controller ch.0 output pins.
57 to 60	59 to 62	P74 to P77	H	General purpose input output ports.
		PWM1P1 PWM1M1 PWM2P1 PWM2M1		Stepping motor controller ch.1 output pins.
62 to 65	64 to 67	P80 to P83	H	General purpose input output ports.
		PWM1P2 PWM1M2 PWM2P2 PWM2M2		Stepping motor controller ch.2 output pins.
67 to 70	69 to 72	P84 to P87	H	General purpose input output ports.
		PWM1P3 PWM1M3 PWM2P3 PWM2M3		Stepping motor controller ch.3 output pins.
72	74	P54	G	General purpose input output port.
		TX0		CAN interface 0 TX output pin.
73	75	P55	G	General purpose output port.
		RX0		CAN interface 0 RX input pin.
74	76	P56	G	General purpose input output port.
		SGO		Sound generator SG0 output pin.
		FRCK		Free-run timer clock input pin.
76	78	P57	G	General purpose input output port.
		SGA		Sound generator SGA output pin.
28 to 31	30 to 33	V0 to V3	—	LCD controller /driver reference power supply pins.
56, 66	58, 68	DV_{CC}	—	High current output buffer with dedicated power supply input pins (pin numbers 54-57, 59-62, 64-67, 69-72) .
51, 61, 71	53, 63, 73	DV_{SS}	—	High current output buffer with dedicated power supply GND pins (pin numbers 54-57, 59-62, 64-67, 69-72) .
32	34	AV_{CC}	—	A/D converter dedicated power supply input pin.
35	37	AV_{SS}	—	A/D converter dedicated GND supply pin.
33	35	AVRH	—	A/D converter Vref + input pin. Vref – AVss.

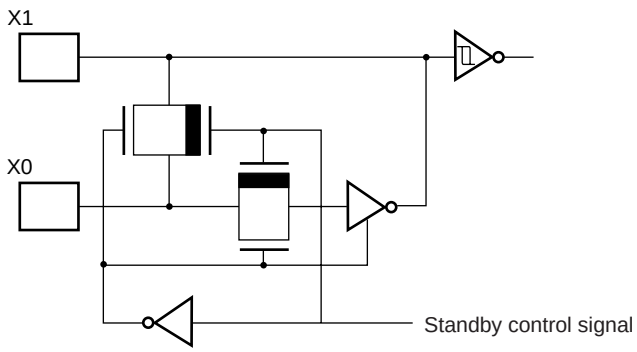
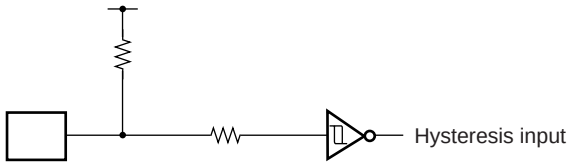
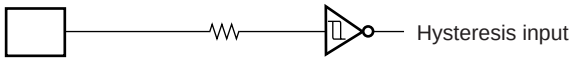
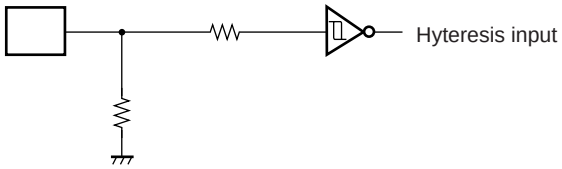
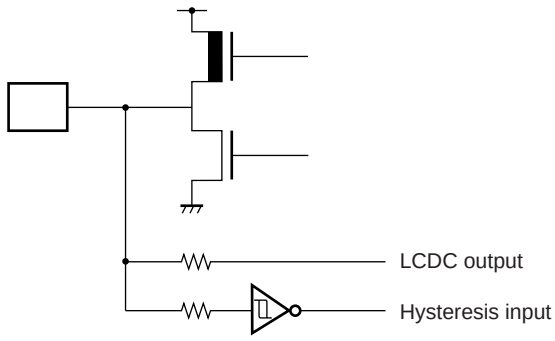
(Continued)

(Continued)

Pin no.		Symbol	Circuit type	Description
LQFP	QFP			
47 48	49 50	MD0 MD1	B *	Test mode input pins. Connect to V_{CC}.
49	51	MD2	D *	Text mode input pin. Connect to V_{SS}.
25	27	C	—	External capacitor pin. Connect an 0.1 μF capacitor between this pin and V_{SS}.
21, 82	23, 84	V_{CC}	—	Power supply input pins.
9, 40, 79	11, 42, 81	V_{SS}	—	GND power supply pins.

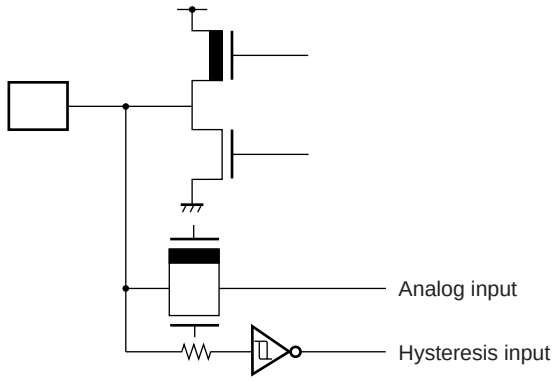
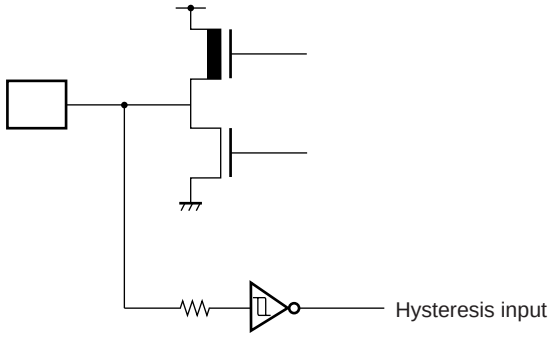
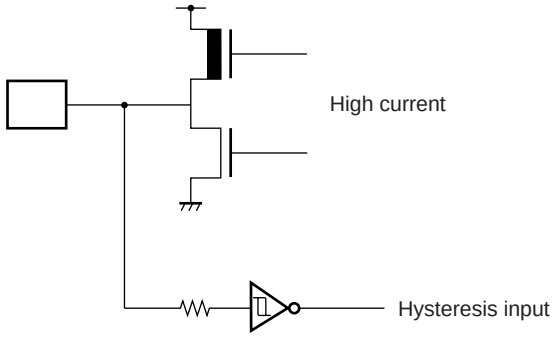
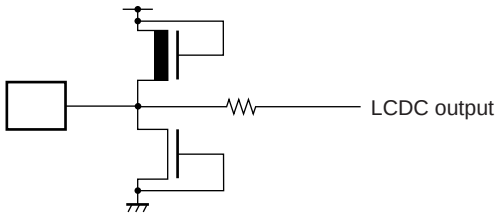
* : Type C in the flash ROM models.

■ I/O CIRCUIT TYPE

Type	Circuit	Remarks
A		• Oscillation feedback resistance : approx. 1 MΩ
B		• Pull-up resistance attached : approx. 50 kΩ, hysteresis input
C		• Hysteresis input
D		• Pull-down resistance attached : approx. 50 kΩ, hysteresis input • No pull-down resistance on flash models.
E		• CMOS output • LCDC output • Hysteresis input

(Continued)

(Continued)

Type	Circuit	Remarks
F		• CMOS output • Hysteresis input • Analog input
G		• CMOS output • Hysteresis input
H		• CMOS high current output • Hysteresis input
I		• LCDC output

■ HANDLING DEVICES

When handling semiconductor devices, care must be taken with regard to the following ten matters.

- Strictly observe maximum rated voltages (prevent latchup)
- Stable supply voltage
- Power-on procedures
- Treatment of unused input pins
- Treatment of A/D converter power supply pins
- Use of external clock signals
- Power supply pins
- Proper sequence of A/D converter power supply analog input
- Handling the power supply for high-current output buffer pins (DV_{CC}, DV_{SS})
- Pull-up/pull-down resistance
- Precautions when not using a sub clock signal.

Precautions for Handling Semiconductor Devices

- **Strictly observe maximum rated voltages (prevent latchup)**

When CMOS integrated circuit devices are subjected to applied voltages higher than V_{CC} at input and output pins other than medium- and high-withstand voltage pins, or to voltages lower than V_{SS}, or when voltages in excess of rated levels are applied between V_{CC} and V_{SS}, a phenomenon known as latchup can occur. In a latchup condition, supply current can increase dramatically and may destroy semiconductor elements. In using semiconductor devices, always take sufficient care to avoid exceeding maximum ratings.

Also care must be taken when power to analog systems is switched on or off, to ensure that the analog power supply (AV_{CC}, AVRH, DV_{CC}) and analog input do not exceed the digital power supply (V_{CC}) .

Once the digital power supply (V_{CC}) is switched on, the analog power (AV_{CC},AVRH,DV_{CC}) may be turned on in any sequence.

- **Stable supply voltage**

Even within the warranted operating range of V_{CC} supply voltage, sudden fluctuations in supply voltage can cause abnormal operation. The recommended stability for ripple fluctuations (P-P values) at commercial frequencies (50 to 60 Hz) should be within 10% of the standard V_{CC} value, and voltage fluctuations that occur during switching of power supplies etc. should be limited to transient fluctuation rates of 0.1 V/ms or less.

- **Power-on procedures**

In order to prevent abnormal operation of the internal built-in step-down circuits, voltage rise time during power-on should be attained within 50 μs (0.2 V to 2.7 V) .

- **Treatment of unused input pins**

If unused input pins are left open, they may cause abnormal operation or latchup which may lead to permanent damage to the semiconductor. Any such pins should be pulled up or pulled down through resistance of at least 2 kΩ.

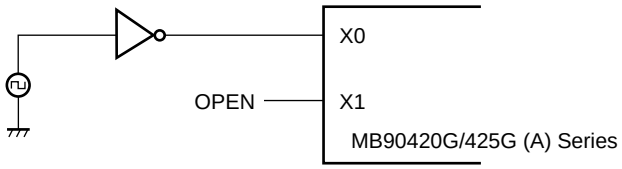
Also any unused input/output pins should be left open in output status, or if found set to input status, they should be treated in the same way as input pins.

- **Treatment of A/D converter power supply pins**

Even if the A/D converter is not used, pins should be connected so that $AV_{CC} = V_{CC}$, and $AV_{SS} = AVRH = V_{SS}$.

- **Use of external clock signals**

Even when an external clock is used, a stabilization period is required following a power-on reset or release from sub clock mode or stop mode. Also, when an external clock is used it should drive only the X0 pin and the X1 pin should be left open, as shown in Figure 3.

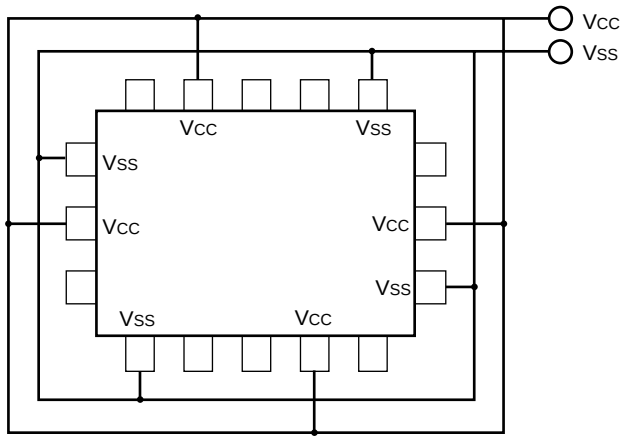


Sample external clock connection

- **Power supply pins**

Devices are designed to prevent problems such as latchup when multiple V_{CC} and V_{SS} supply pins are used, by providing internal connections between pins having the same potential. However, in order to reduce unwanted radiation, and to prevent abnormal operation of strobe signals due to rise in ground level, and to maintain total output current ratings, all such pins should always be connected externally to power supplies and ground.

As shown in Figure 4, all V_{CC} power supply pins must have the same potential. All V_{SS} power supply pins should be handled in the same way. If there are multiple V_{CC} or V_{SS} systems, the device will not operate properly even within the warranted operating range.



Power supply input pins (V_{CC}/V_{SS})

In addition, care must be given to connecting the V_{CC} and V_{SS} pins of this device to a current source with as little impedance as possible. It is recommended that a bypass capacitor of 1.0 µF be connected between V_{CC} and V_{SS} as close to the pins as possible.

- **Proper sequence of A/D converter power supply analog input**

A/D converter power (AV_{CC}, AVRH) and analog input (AN0-AN7) must be applied after the digital power supply (V_{CC}) is switched on. When power is shut off, the A/D converter power supply and analog input must be cut off before the digital power supply is switched on (V_{CC}) . In both power-on and shut-off, care should be taken that AVRH does not exceed AV_{CC}. Even when pins which double as analog input pins are used as input ports, be sure that the input voltage does not exceed AV_{CC}. (There is no problem if analog power supplies and digital power supplies are turned off and on at the same time.)

- **Handling the power supply for high-current output buffer pins (DV_{CC}, DV_{SS})**

Always apply power to high-current output buffer pins (DV_{CC}, DV_{SS}) after the digital power supply (V_{CC}) is turned on. Also when switching power off, always shut off the power supply to the high-current output buffer pins (DV_{CC}, DV_{SS}) before switching off the digital power supply (V_{CC}) . (There will be no problem if high-current output buffer pins and digital power supplies are turned off and on at the same time.)

Even when high-current output buffer pins are used as general purpose ports, the power for high current output buffer pins (DV_{CC}, DV_{SS}) should be applied to these pins.

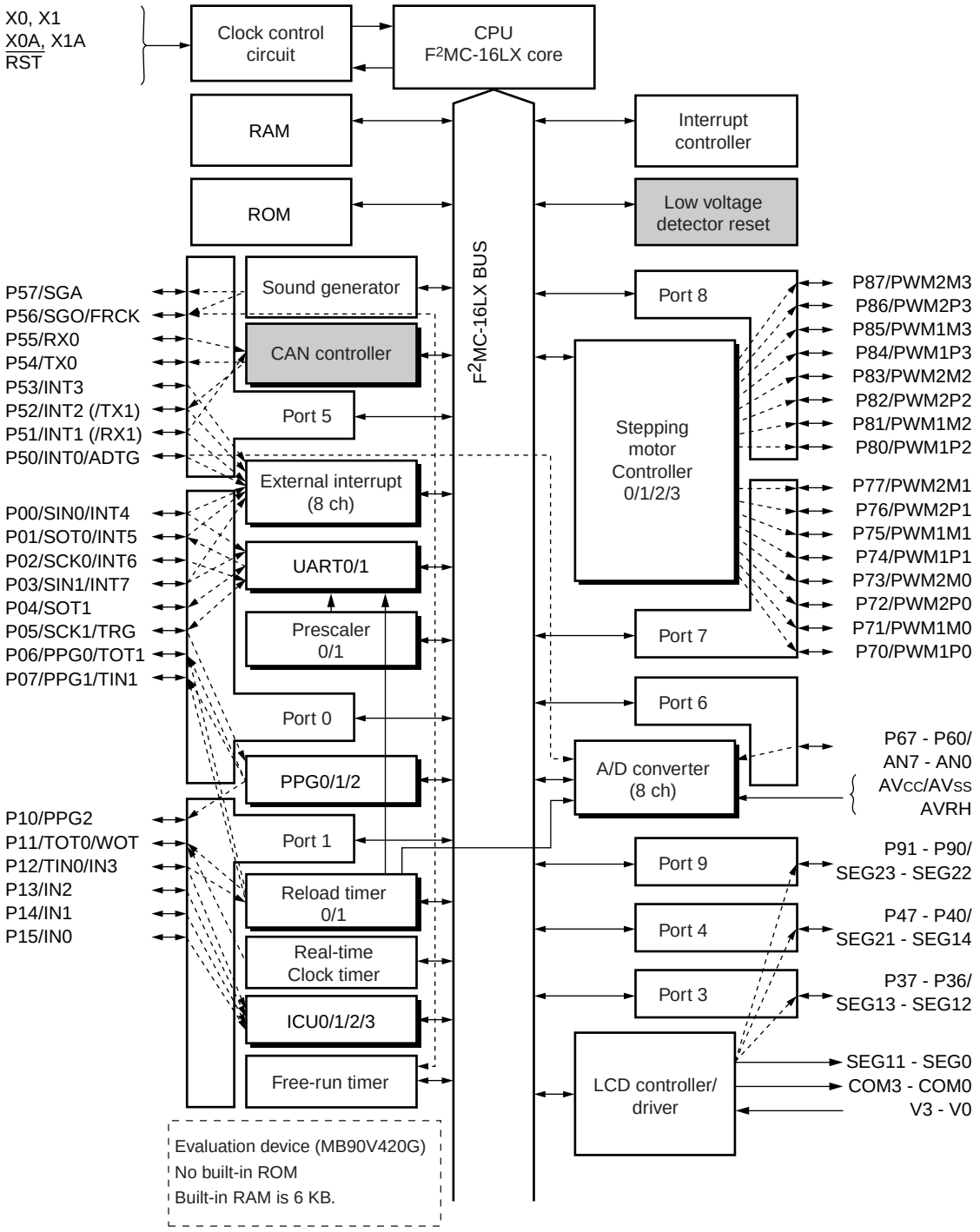
- **Pull-up/pull-down resistance**

The MB90420G/5G series does not support internal pull-up/pull-down resistance. If necessary, use external components.

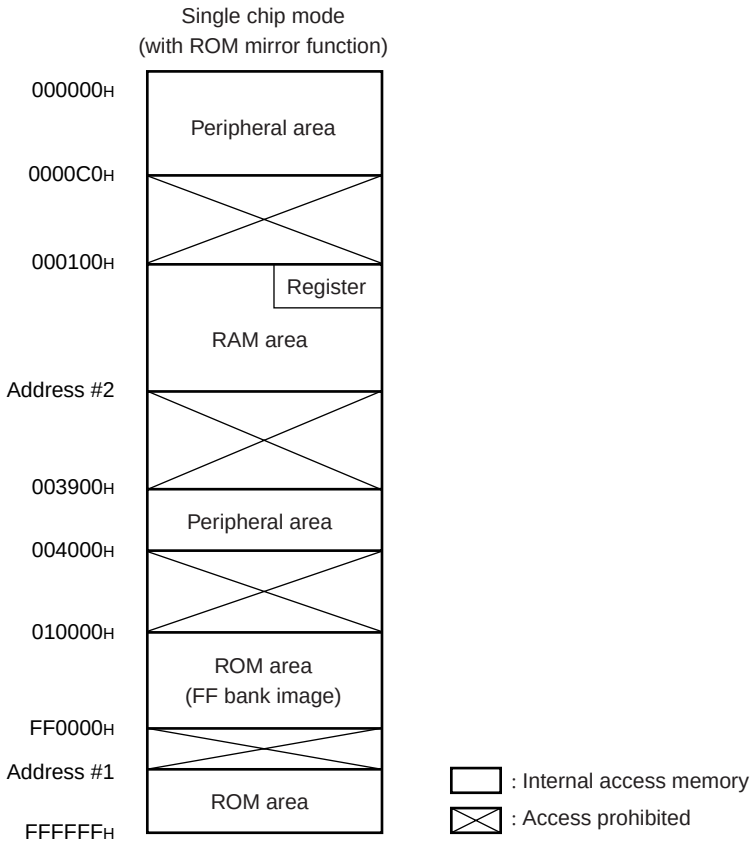
- **Precautions for when not using a sub clock signal.**

If the X0A and X1A pins are not connected to an oscillator, apply pull-down treatment to the X0A pin and leave the X1A pin open.

■ BLOCK DIAGRAM



■ MEMORY MAP



Parts No.	Address #1	Address #2
MB90423G (A)	FE0000H	001900H
MB90427G (A)	FF0000H	001100H
MB90428G (A)	FE0000H	001900H
MB90F423G (A)	FE0000H	001900H
MB90F428G (A)	FE0000H	001900H
MB90V420G	FE0000H *	001900H

* : MB90V420G has no built-in ROM. On the tool side this area may be considered a ROM decoder.

Note : To select models without the ROM mirror function, see the “ROM Mirror Function Selection Module.” The image of the ROM data in the FF bank appears at the top of the 00 bank, in order to enable efficient use of small C compiler models. The lower 16-bit address for the FF bank will be assigned to the same address, so that tables in ROM can be referenced without declaring a “far” indication with the pointer. For example when accessing the address 00C000H, the actual access is to address FFC000H in ROM. Here the FF bank ROM area exceeds 48 KB, so that it is not possible to see the entire area in the 00 bank image. Therefore because the ROM data from FF4000H to FFFFFFH will appear in the image from 004000H to 00FFFFH, it is recommended that the ROM data table be stored in the area from FF4000H to FFFFFFH.

■ I/O MAP

• Other than CAN Interface

Address	Register name	Symbol	Read/write	Peripheral function	Initial value
00H	Port 0 data register	PDR0	R/W	Port 0	XXXXXXXX
01H	Port 1 data register	PDR1	R/W	Port 1	- - XXXXXX
02H	(Disabled)				
03H	Port 3 data register	PDR3	R/W	Port 3	XX - - - - - -
04H	Port 4 data register	PDR4	R/W	Port 4	XXXXXXXX
05H	Port 5 data register	PDR5	R/W	Port 5	XXXXXXXX
06H	Port 6 data register	PDR6	R/W	Port 6	XXXXXXXX
07H	Port 7 data register	PDR7	R/W	Port 7	XXXXXXXX
08H	Port 8 data register	PDR8	R/W	Port 8	XXXXXXXX
09H	Port 9 data register	PDR9	R/W	Port 9	- - - - - -XX
0AH to 0FH	(Disabled)				
10H	Port 0 direction register	DDR0	R/W	Port 0	0 0 0 0 0 0 0 0
11H	Port 1 direction register	DDR1	R/W	Port 1	- - 0 0 0 0 0 0
12H	(Disabled)				
13H	Port 3 direction register	DDR3	R/W	Port 3	0 0 - - - - - -
14H	Port 4 direction register	DDR4	R/W	Port 4	0 0 0 0 0 0 0 0
15H	Port 5 direction register	DDR5	R/W	Port 5	0 0 0 0 0 0 0 0
16H	Port 6 direction register	DDR6	R/W	Port 6	0 0 0 0 0 0 0 0
17H	Port 7 direction register	DDR7	R/W	Port 7	0 0 0 0 0 0 0 0
18H	Port 8 direction register	DDR8	R/W	Port 8	0 0 0 0 0 0 0 0
19H	Port 9 direction register	DDR9	R/W	Port 9	- - - - - - 0 0
1AH	Analog input enable	ADER	R/W	Port 6, A/D	1 1 1 1 1 1 1 1
1BH to 1FH	(Disabled)				
20H	A/D control status register lower	ADCSL	R/W	A/D converter	0 0 0 0 0 0 0 0
21H	A/D control status register higher	ADCSH	R/W		0 0 0 0 0 0 0 0
22H	A/D data register lower	ADCRL	R		XXXXXXXX
23H	A/D data register higher	ADCRH	R/W		0 0 1 0 1 XXX
24H	Compare clear register	CPCLR	R/W	16-bit free-run timer	XXXXXXXX
25H			R/W		XXXXXXXX
26H	Timer data register	TCDT	R/W		0 0 0 0 0 0 0 0
27H			R/W		0 0 0 0 0 0 0 0
28H	Timer control status register lower	TCCSL	R/W		0 0 0 0 0 0 0 0
29H	Timer control status register higher	TCCSH	R/W		0 - - 0 0 0 0 0

(Continued)

Address	Register name	Symbol	Read/write	Peripheral function	Initial value
2A$_H$	PPG0 control status register lower	PCNTL0	R/W	16-bit PPG0	0 0 0 0 0 0 0 0
2B$_H$	PPG0 control status register higher	PCNTH0	R/W		0 0 0 0 0 0 0 -
2C$_H$	PPG1 control status register lower	PCNTL1	R/W	16-bit PPG1	0 0 0 0 0 0 0 0
2D$_H$	PPG1 control status register higher	PCNTH1	R/W		0 0 0 0 0 0 0 -
2E$_H$	PPG2 control status register lower	PCNTL2	R/W	16-bit PPG2	0 0 0 0 0 0 0 0
2F$_H$	PPG2 control status register higher	PCNTH2	R/W		0 0 0 0 0 0 0 -
30$_H$	External interrupt enable	ENIR	R/W	External interrupt	0 0 0 0 0 0 0 0
31$_H$	External interrupt request	EIRR	R/W		XXXXXXXX
32$_H$	External interrupt level lower	ELVRL	R/W		0 0 0 0 0 0 0 0
33$_H$	External interrupt level higher	ELVRH	R/W		0 0 0 0 0 0 0 0
34$_H$	Serial mode register 0	SMR0	R/W	UART 0	0 0 0 0 0 - 0 0
35$_H$	Serial control register 0	SCR0	R/W		0 0 0 0 0 1 0 0
36$_H$	Input data register 0/ Output data register 0	SIDR0/ SODR0	R/W		XXXXXXXX
37$_H$	Serial status register 0	SSR0	R/W		0 0 0 0 1 0 0 0
38$_H$	Serial mode register 1	SMR1	R/W	UART1	0 0 0 0 0 – 0 0
39$_H$	Serial control register 1	SCR1	R/W		0 0 0 0 0 1 0 0
3A$_H$	Input data register 1/ Output data register 1	SIDR1/ SODR1	R/W		XXXXXXXX
3B$_H$	Serial status register 1	SSR1	R/W		0 0 0 0 1 0 0 0
3C$_H$	(Disabled)				
3D$_H$	Clock division control register 0	CDCR0	R/W	Prescaler	0 - - - 0 0 0 0
3E$_H$	CAN wake-up control register	CWUCR	R/W	CAN	- - - - - - - 0
3F$_H$	Clock division control register 1	CDCR1	R/W	Prescaler	0 - - - 0 0 0 0
40$_H$ to 4F$_H$	Area reserved for CAN interface 0				
50$_H$	Timer control status register 0 lower	TMCSR0L	R/W	16-bit reload timer 0	0 0 0 0 0 0 0 0
51$_H$	Timer control status register 0 higher	TMCSR0H	R/W		- - - 0 0 0 0 0
52$_H$	Timer register 0/ Reload register 0	TMR0/ TMRLR0	R/W		XXXXXXXX
53$_H$					XXXXXXXX
54$_H$	Timer control status register 1 lower	TMCSR1L	R/W	16-bit reload timer 1	0 0 0 0 0 0 0 0
55$_H$	Timer control status register 1 higher	TMCSR1H	R/W		- - - 0 0 0 0 0
56$_H$	Timer register 1/ Reload register 1	TMR1/ TMRLR1	R/W		XXXXXXXX
57$_H$					XXXXXXXX
58$_H$	Clock timer control register lower	WTCRL	R/W	Real-time clock timer	0 0 0 - - 0 0 0
59$_H$	Clock timer control register higher	WTCRH	R/W		0 0 0 0 0 0 0 0

(Continued)

Address	Register name	Symbol	Read/write	Peripheral function	Initial value
5A$_H$	Sound control register lower	SGCRL	R/W	Sound generator	0 0 0 0 0 0 0 0
5B$_H$	Sound control register higher	SGCRH	R/W		0 - - - - - 0 0
5C$_H$	Frequency data register	SGFR	R/W		XXXXXXXX
5D$_H$	Amplitude data register	SGAR	R/W		0 0 0 0 0 0 0 0
5E$_H$	Decrement grade register	SGDR	R/W		XXXXXXXX
5F$_H$	Tone count register	SGTR	R/W		XXXXXXXX
60$_H$	Input capture register 0	IPCP0	R	Input capture 0/1	XXXXXXXX
61$_H$					XXXXXXXX
62$_H$	Input capture register 1	IPCP1	R		XXXXXXXX
63$_H$					XXXXXXXX
64$_H$	Input capture register 2	IPCP2	R	Input capture 2/3	XXXXXXXX
65$_H$					XXXXXXXX
66$_H$	Input capture register 3	IPCP3	R		XXXXXXXX
67$_H$					XXXXXXXX
68$_H$	Input capture control status 0/1	ICS01	R/W	Input capture 0/1	0 0 0 0 0 0 0 0
69$_H$	(Disabled)				
6A$_H$	Input capture control status 2/3	ICS23	R/W	Input capture 2/3	0 0 0 0 0 0 0 0
6B$_H$	(Disabled)				
6C$_H$	LCDC control register lower	LCRL	R/W	LCD controller/ driver	0 0 0 1 0 0 0 0
6D$_H$	LCDC control register higher	LCRH	R/W		0 0 0 0 0 0 0 0
6E$_H$	Low voltage detect reset control register	LVRC	R/W	Low voltage detect reset	1 0 1 1 1 0 0 0
6F$_H$	ROM mirror	ROMM	W	ROM mirror	XXXXXXX1
70$_H$ to 7F$_H$	Area reserved for CAN interface 1				
80$_H$	PWM control register 0	PWC0	R/W	Stepping motor controller0	0 0 0 0 0 - - 0
81$_H$	(Disabled)				
82$_H$	PWM control register 1	PWC1	R/W	Stepping motor controller1	0 0 0 0 0 - - 0
83$_H$	(Disabled)				
84$_H$	PWM control register 2	PWC2	R/W	Stepping motor controller2	0 0 0 0 0 - - 0
85$_H$	(Disabled)				
86$_H$	PWM control register 3	PWC3	R/W	Stepping motor controller3	0 0 0 0 0 - - 0
87$_H$ to 9D$_H$	(Disabled)				

(Continued)

(Continued)

Address	Register name	Symbol	Read/write	Peripheral function	Initial value
9E$_H$	ROM correction control register	PACSR	R/W	Address match detection function	- - - - - 0 - 0
9F$_H$	Delay interrupt/release	DIRR	R/W	Delayed interrupt	- - - - - - - 0
A0$_H$	Power saving mode	LPMCR	R/W	Power saving control circuit	0 0 0 1 1 0 0 0
A1$_H$	Clock select	CKSCR	R/W		1 1 1 1 1 1 0 0
A2$_H$ to A7$_H$	(Disabled)				
A8$_H$	Watchdog control	WDTC	R/W	Watchdog timer	XXXXX 1 1 1
A9$_H$	Time base timer control register	TBTC	R/W	Time base timer	1 - - 0 0 1 0 0
AA$_H$	Clock timer control register	WTC	R/W	Clock timer (sub clock)	1 X 0 0 0 0 0 0
AB$_H$ to AD$_H$	(Disabled)				
AE$_H$	Flash control register	FMCS	R/W	Flash interface	0 0 0 X 0 XX 0
AF$_H$	(Disabled)				
B0$_H$	Interrupt control register 00	ICR00	R/W	Interrupt controller	0 0 0 0 0 1 1 1
B1$_H$	Interrupt control register 01	ICR01	R/W		0 0 0 0 0 1 1 1
B2$_H$	Interrupt control register 02	ICR02	R/W		0 0 0 0 0 1 1 1
B3$_H$	Interrupt control register 03	ICR03	R/W		0 0 0 0 0 1 1 1
B4$_H$	Interrupt control register 04	ICR04	R/W		0 0 0 0 0 1 1 1
B5$_H$	Interrupt control register 05	ICR05	R/W		0 0 0 0 0 1 1 1
B6$_H$	Interrupt control register 06	ICR06	R/W		0 0 0 0 0 1 1 1
B7$_H$	Interrupt control register 07	ICR07	R/W		0 0 0 0 0 1 1 1
B8$_H$	Interrupt control register 08	ICR08	R/W		0 0 0 0 0 1 1 1
B9$_H$	Interrupt control register 09	ICR09	R/W		0 0 0 0 0 1 1 1
BA$_H$	Interrupt control register 10	ICR10	R/W		0 0 0 0 0 1 1 1
BB$_H$	Interrupt control register 11	ICR11	R/W		0 0 0 0 0 1 1 1
BC$_H$	Interrupt control register 12	ICR12	R/W		0 0 0 0 0 1 1 1
BD$_H$	Interrupt control register 13	ICR13	R/W		0 0 0 0 0 1 1 1
BE$_H$	Interrupt control register 14	ICR14	R/W		0 0 0 0 0 1 1 1
BF$_H$	Interrupt control register 15	ICR15	R/W		0 0 0 0 0 1 1 1
C0$_H$ to FF$_H$	(Disabled)				

Address	Register name	Symbol	Read/write	Peripheral function	Initial value
1FF0H	ROM correction address 0	PADR0	R/W	Address match detection function	XXXXXXXX
1FF1H	ROM correction address 1	PADR0	R/W		XXXXXXXX
1FF2H	ROM correction address 2	PADR0	R/W		XXXXXXXX
1FF3H	ROM correction address 3	PADR1	R/W		XXXXXXXX
1FF4H	ROM correction address 4	PADR1	R/W		XXXXXXXX
1FF5H	ROM correction address 5	PADR1	R/W		XXXXXXXX
3900H to 391FH	(Disabled)				
3920H	PPG0 down counter register	PDCR0	R	16-bit PPG 0	1 1 1 1 1 1 1 1
3921H					1 1 1 1 1 1 1 1
3922H	PPG0 cycle setting register	PCSR0	W		XXXXXXXX
3923H					XXXXXXXX
3924H	PPG0 duty setting register	PDUT0	W		XXXXXXXX
3925H					XXXXXXXX
3926H to 3927H	(Disabled)				
3928H	PPG1 down counter register	PDCR1	R	16-bit PPG 1	1 1 1 1 1 1 1 1
3929H					1 1 1 1 1 1 1 1
392AH	PPG1 cycle setting register	PCSR1	W		XXXXXXXX
392BH					XXXXXXXX
392CH	PPG1 duty setting register	PDUT1	W		XXXXXXXX
392DH					XXXXXXXX
392EH to 392FH	(Disabled)				
3930H	PPG2 down counter register	PDCR2	R	16 bit PPG 2	1 1 1 1 1 1 1 1
3931H					1 1 1 1 1 1 1 1
3932H	PPG2 cycle setting register	PCSR2	W		XXXXXXXX
3933H					XXXXXXXX
3934H	PPG2 duty setting register	PDUT2	W		XXXXXXXX
3935H					XXXXXXXX
3936H to 3959H	(Disabled)				

(Continued)

Address	Register name	Symbol	Read/write	Peripheral function	Initial value
395A$_H$	Sub second data register	WTBR	R/W	Real time clock timer	XXXXXXXX
395B$_H$					XXXXXXXX
395C$_H$					- - - XXXXX
395D$_H$	Second data register	WTSR	R/W		- - XXXXXX
395E$_H$	Minute data register	WTMR	R/W		- - XXXXXX
395F$_H$	Hour data register	WTHR	R/W		- - - XXXXX
3960$_H$ to 396F$_H$	LCD display RAM	VRAM	R/W	LCD controller/ driver	XXXXXXXX
3970$_H$ to 397F$_H$	(Disabled)				
3980$_H$	PWM1 compare register 0	PWC10	R/W	Stepping motor controller 0	XXXXXXXX
3981$_H$					- - - - - - XX
3982$_H$	PWM2 compare register 0	PWC20	R/W		XXXXXXXX
3983$_H$					- - - - - - XX
3984$_H$	PWM1 select register 0	PWS10	R/W		- - 0 0 0 0 0 0
3985$_H$	PWM2 select register 0	PWS20	R/W		- 0 0 0 0 0 0 0
3986$_H$ to 3987$_H$	(Disabled)				
3988$_H$	PWM1 compare register 1	PWC11	R/W	Stepping motor controller 1	XXXXXXXX
3989$_H$					- - - - - - XX
398A$_H$	PWM2 compare register 1	PWC21	R/W		XXXXXXXX
398B$_H$					- - - - - - XX
398C$_H$	PWM1 select register 1	PWS11	R/W		- - 0 0 0 0 0 0
398D$_H$	PWM2 select register 1	PWS21	R/W		- 0 0 0 0 0 0 0
398E$_H$ to 398F$_H$	(Disabled)				
3990$_H$	PWM1 compare register 2	PWC12	R/W	Stepping motor controller 2	XXXXXXXX
3991$_H$					- - - - - - XX
3992$_H$	PWM2 compare register 2	PWC22	R/W		XXXXXXXX
3993$_H$					- - - - - - XX
3994$_H$	PWM1 select register 2	PWS12	R/W		- - 0 0 0 0 0 0
3995$_H$	PWM2 select register 2	PWS22	R/W		- 0 0 0 0 0 0 0
3996$_H$ to 3997$_H$	(Disabled)				

(Continued)

(Continued)

Address	Register name	Symbol	Read/write	Peripheral function	Initial value
3998H	PWM1 compare register 3	PWC13	R/W	Stepping motor controller 3	XXXXXXXX
3999H					- - - - - - XX
399AH	PWM2 compare register 3	PWC23	R/W		XXXXXXXX
399BH					- - - - - - XX
399CH	PWM1 select register 3	PWS13	R/W		- - 0 0 0 0 0 0
399DH	PWM2 select register 3	PWS23	R/W		- 0 0 0 0 0 0 0
399EH to 39FFH	(Disabled)				
3A00H to 3AFFH	Area reserved for CAN interface 0				
3B00H to 3BFFH	Area reserved for CAN interface 1				
3C00H to 3CFFH	Area reserved for CAN interface 0				
3D00H to 3DFFH	Area reserved for CAN interface 1				
3E00H to 3EFFH	(Disabled)				

- Initial value symbols :
  “0” initial value 0.
  “1” initial value 1.
  “X” initial value undetermined
  “-” initial value undetermined (none)
- Write/read symbols :
  “R/W” read/write enabled
  “R” read only
  “W” write only
- Addresses in the area 0000H to 00FFH are reserved for the principal functions of the MCU. Read access attempts to reserved areas will result in an “X” value. Also, write access to reserved areas is prohibited.

• I/O Map for CAN Interface

Address		Register name	Symbol	Read/ write	Initial value
CAN0	CAN1				
000040H	000070H	Message buffer valid area	BVALR	(R/W)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
000041H	000071H				
000042H	000072H	Transmission request register	TREQR	(R/W)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
000043H	000073H				
000044H	000074H	Transmission cancel register	TCANR	(W)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
000045H	000075H				
000046H	000076H	Transmission completed register	TCR	(R/W)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
000047H	000077H				
000048H	000078H	Receiving completed register	RCR	(R/W)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
000049H	000079H				
00004AH	00007AH	Remote request receiving register	RRTRR	(R/W)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
00004BH	00007BH				
00004CH	00007CH	Receiving overrun register	ROVRR	(R/W)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
00004DH	00007DH				
00004EH	00007EH	Receiving interrupt enable register	RIER	(R/W)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
00004FH	00007FH				
003C00H	003D00H	Control status register	CSR	(R/W, R)	0 0 - - - 0 0 0 0 - - - - 0 - 1
003C01H	003D01H				
003C02H	003D02H	Last event indicator register	LEIR	(R/W)	- - - - - - - - 0 0 0 - 0 0 0 0
003C03H	003D03H				
003C04H	003D04H	RX/TX error counter	RTEC	(R)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
003C05H	003D05H				
003C06H	003D06H	Bit timing register	BTR	(R/W)	- 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
003C07H	003D07H				
003C08H	003D08H	IDE register	IDER	(R/W)	XXXXXXXX XXXXXXXX
003C09H	003D09H				
003C0AH	003D0AH	Transmission RTR register	TRTRR	(R/W)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
003C0BH	003D0BH				
003C0CH	003D0CH	Remote frame receiving wait register	RFWTR	(R/W)	XXXXXXXX XXXXXXXX
003C0DH	003D0DH				
003C0EH	003D0EH	Transmission interrupt enable register	TIER	(R/W)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
003C0FH	003D0FH				

(Continued)

Address		Register name	Symbol	Read/ write	Initial value
CAN0	CAN1				
003C10H	003D10H	Acceptance mask select register	AMSR	(R/W)	XXXXXXXX XXXXXXXX
003C11H	003D11H				
003C12H	003D12H				XXXXXXXX XXXXXXXX
003C13H	003D13H				
003C14H	003D14H	Acceptance mask register 0	AMR0	(R/W)	XXXXXXXX XXXXXXXX
003C15H	003D15H				
003C16H	003D16H				XXXXX- - - XXXXXXXX
003C17H	003D17H				
003C18H	003D18H	Acceptance mask register 1	AMR1	(R/W)	XXXXXXXX XXXXXXXX
003C19H	003D19H				
003C1AH	003D1AH				XXXXX- - - XXXXXXXX
003C1BH	003D1BH				
003A00H to 003A1FH	003B00H to 003B1FH	General purpose RAM	—	(R/W)	XXXXXXXX to XXXXXXXX
003A20H	003B20H	ID register 0	IDR0	(R/W)	XXXXXXXX XXXXXXXX
003A21H	003B21H				
003A22H	003B22H				XXXXX- - - XXXXXXXX
003A23H	003B23H				
003A24H	003B24H	ID register 1	IDR1	(R/W)	XXXXXXXX XXXXXXXX
003A25H	003B25H				
003A26H	003B26H				XXXXX- - - XXXXXXXX
003A27H	003B27H				
003A28H	003B28H	ID register 2	IDR2	(R/W)	XXXXXXXX XXXXXXXX
003A29H	003B29H				
003A2AH	003B2AH				XXXXX- - - XXXXXXXX
003A2BH	003B2BH				
003A2CH	003B2CH	ID register 3	IDR3	(R/W)	XXXXXXXX XXXXXXXX
003A2DH	003B2DH				
003A2EH	003B2EH				XXXXX- - - XXXXXXXX
003A2FH	003B2FH				
003A30H	003B30H	ID register 4	IDR4	(R/W)	XXXXXXXX XXXXXXXX
003A31H	003B31H				
003A32H	003B32H				XXXXX- - - XXXXXXXX
003A33H	003B33H				

(Continued)

Address		Register name	Symbol	Read/ write	Initial value
CAN0	CAN1				
003A34H	003B34H	ID register 5	IDR5	(R/W)	XXXXXXXX XXXXXXXX
003A35H	003B35H				
003A36H	003B36H				XXXXX- - - XXXXXXXX
003A37H	003B37H				
003A38H	003B38H	ID register 6	IDR6	(R/W)	XXXXXXXX XXXXXXXX
003A39H	003B39H				
003A3AH	003B3AH				XXXXX- - - XXXXXXXX
003A3BH	003B3BH				
003A3CH	003B3CH	ID register 7	IDR7	(R/W)	XXXXXXXX XXXXXXXX
003A3DH	003B3DH				
003A3EH	003B3EH				XXXXX- - - XXXXXXXX
003A3FH	003B3FH				
003A40H	003B40H	ID register 8	IDR8	(R/W)	XXXXXXXX XXXXXXXX
003A41H	003B41H				
003A42H	003B42H				XXXXX- - - XXXXXXXX
003A43H	003B43H				
003A44H	003B44H	ID register 9	IDR9	(R/W)	XXXXXXXX XXXXXXXX
003A45H	003B45H				
003A46H	003B46H				XXXXX- - - XXXXXXXX
003A47H	003B47H				
003A48H	003B48H	ID register 10	IDR10	(R/W)	XXXXXXXX XXXXXXXX
003A49H	003B49H				
003A4AH	003B4AH				XXXXX- - - XXXXXXXX
003A4BH	003B4BH				
003A4CH	003B4CH	ID register 11	IDR11	(R/W)	XXXXXXXX XXXXXXXX
003A4DH	003B4DH				
003A4EH	003B4EH				XXXXX- - - XXXXXXXX
003A4FH	003B4FH				
003A50H	003B50H	ID register 12	IDR12	(R/W)	XXXXXXXX XXXXXXXX
003A51H	003B51H				
003A52H	003B52H				XXXXX- - - XXXXXXXX
003A53H	003B53H				

(Continued)

Address		Register name	Symbol	Read/ write	Initial value	
CAN0	CAN1					
003A54H	003B54H	ID register 13	IDR13	(R/W)	XXXXXXXX	XXXXXXXX
003A55H	003B55H					
003A56H	003B56H				XXXXX- - -	XXXXXXXX
003A57H	003B57H					
003A58H	003B58H	ID register 14	IDR14	(R/W)	XXXXXXXX	XXXXXXXX
003A59H	003B59H					
003A5AH	003B5AH				XXXXX- - -	XXXXXXXX
003A5BH	003B5BH					
003A5CH	003B5CH	ID register 15	IDR15	(R/W)	XXXXXXXX	XXXXXXXX
003A5DH	003B5DH					
003A5EH	003B5EH				XXXXX- - -	XXXXXXXX
003A5FH	003B5FH					
003A60H	003B60H	DLC register 0	DLCR0	(R/W)	- - - -XXXX	- - - -XXXX
003A61H	003B61H					
003A62H	003B62H	DLC register 1	DLCR1	(R/W)	- - - -XXXX	- - - -XXXX
003A63H	003B63H					
003A64H	003B64H	DLC register 2	DLCR2	(R/W)	- - - -XXXX	- - - -XXXX
003A65H	003B65H					
003A66H	003B66H	DLC register 3	DLCR3	(R/W)	- - - -XXXX	- - - -XXXX
003A67H	003B67H					
003A68H	003B68H	DLC register 4	DLCR4	(R/W)	- - - -XXXX	- - - -XXXX
003A69H	003B69H					
003A6AH	003B6AH	DLC register 5	DLCR5	(R/W)	- - - -XXXX	- - - -XXXX
003A6BH	003B6BH					
003A6CH	003B6CH	DLC register 6	DLCR6	(R/W)	- - - -XXXX	- - - -XXXX
003A6DH	003B6DH					
003A6EH	003B6EH	DLC register 7	DLCR7	(R/W)	- - - -XXXX	- - - -XXXX
003A6FH	003B6FH					
003A70H	003B70H	DLC register 8	DLCR8	(R/W)	- - - -XXXX	- - - -XXXX
003A71H	003B71H					
003A72H	003B72H	DLC register 9	DLCR9	(R/W)	- - - -XXXX	- - - -XXXX
003A73H	003B73H					
003A74H	003B74H	DLC register 10	DLCR10	(R/W)	- - - -XXXX	- - - -XXXX
003A75H	003B75H					

(Continued)

Address		Register name	Symbol	Read/ write	Initial value
CAN0	CAN1				
003A76H	003B76H	DLC register 11	DLCR11	(R/W)	- - - -XXXX - - - -XXXX
003A77H	003B77H				
003A78H	003B78H	DLC register 12	DLCR12	(R/W)	- - - -XXXX - - - -XXXX
003A79H	003B79H				
003A7AH	003B7AH	DLC register 13	DLCR13	(R/W)	- - - -XXXX - - - -XXXX
003A7BH	003B7BH				
003A7CH	003B7CH	DLC register 14	DLCR14	(R/W)	- - - -XXXX - - - -XXXX
003A7DH	003B7DH				
003A7EH	003B7EH	DLC register 15	DLCR15	(R/W)	- - - -XXXX - - - -XXXX
003A7FH	003B7FH				
003A80H to 003A87H	003B80H to 003B87H	Data register 0 (8 bytes)	DTR0	(R/W)	XXXXXXXX to XXXXXXXX
003A88H to 003A8FH	003B88H to 003B8FH	Data register 1 (8 bytes)	DTR1	(R/W)	XXXXXXXX to XXXXXXXX
003A90H to 003A87H	003B90H to 003B97H	Data register 2 (8 bytes)	DTR2	(R/W)	XXXXXXXX to XXXXXXXX
003A98H to 003A9FH	003B98H to 003B9FH	Data register 3 (8 bytes)	DTR3	(R/W)	XXXXXXXX to XXXXXXXX
003AA0H to 003AA7H	003BA0H to 003BA7H	Data register 4 (8 bytes)	DTR4	(R/W)	XXXXXXXX to XXXXXXXX
003AA8H to 003AAFH	003BA8H to 003BAFH	Data register 5 (8 bytes)	DTR5	(R/W)	XXXXXXXX to XXXXXXXX
003AB0H to 003AB7H	003BB0H to 003BB7H	Data register 6 (8 bytes)	DTR6	(R/W)	XXXXXXXX to XXXXXXXX
003AB8H to 003ABFH	003BB8H to 003BBFH	Data register 7 (8 bytes)	DTR7	(R/W)	XXXXXXXX to XXXXXXXX
003AC0H to 003AC7H	003BC0H to 003BC7H	Data register 8 (8 bytes)	DTR8	(R/W)	XXXXXXXX to XXXXXXXX
003AC8H to 003ACFH	003BC8H to 003BCFH	Data register 9 (8 bytes)	DTR9	(R/W)	XXXXXXXX to XXXXXXXX

(Continued)

(Continued)

Address		Register name	Symbol	Read/ write	Initial value
CAN0	CAN1				
003AD0H to 003AD7H	003BD0H to 003BD7H	Data register 10 (8 bytes)	DTR10	(R/W)	XXXXXXXX to XXXXXXXX
003AD8H to 003ADFH	003BD8H to 003BDFH	Data register 11 (8 bytes)	DTR11	(R/W)	XXXXXXXX to XXXXXXXX
003AE0H to 003AE7H	003BE0H to 003BE7H	Data register 12 (8 bytes)	DTR12	(R/W)	XXXXXXXX to XXXXXXXX
003AE8H to 003AEFH	003BE8H to 003BEFH	Data register 13 (8 bytes)	DTR13	(R/W)	XXXXXXXX to XXXXXXXX
003AF0H to 003AF7H	003BF0H to 003BF7H	Data register 14 (8 bytes)	DTR14	(R/W)	XXXXXXXX to XXXXXXXX
003AF8H to 003AFFH	003BF8H to 003BFFH	Data register 15 (8 bytes)	DTR15	(R/W)	XXXXXXXX to XXXXXXXX

■ INTERRUPT SOURCES, INTERRUPT VECTORS, AND INTERRUPT CONTROL REGISTERS

Interrupt source	EI²OS compatible	Interrupt vector			Interrupt control register		Priority *2
		Number		Address	ICR	Address	
Reset	×	#08	08H	FFFFDCH	—	—	High
INT9 instruction	×	#09	09H	FFFFD8H	—	—	
Exception processing	×	#10	0AH	FFFFD4H	—	—	
CAN0 RX	×	#11	0BH	FFFFD0H	ICR00	0000B0H *1	
CAN0 TX/NS	×	#12	0CH	FFFFCCH			
CAN1 RX	×	#13	0DH	FFFFC8H	ICR01	0000B1H *1	
CAN1 TX/NS	×	#14	0EH	FFFFC4H			
Input capture 0	△	#15	0FH	FFFFC0H	ICR02	0000B2H *1	
DTP/external interrupt - ch 0 detected	△	#16	10H	FFFFBCH			
Reload timer 0	△	#17	11H	FFFFB8H	ICR03	0000B3H *1	
DTP/external interrupt - ch 1 detected	△	#18	12H	FFFFB4H			
Input capture 1	△	#19	13H	FFFFB0H	ICR04	0000B4H *1	
DTP/external interrupt - ch 2 detected	△	#20	14H	FFFFACH			
Input capture 2	△	#21	15H	FFFFA8H	ICR05	0000B5H *1	
DTP/external interrupt - ch 3 detected	△	#22	16H	FFFFA4H			
Input capture 3	△	#23	17H	FFFFA0H	ICR06	0000B6H *1	
DTP/external interrupt - ch 4/5 detected	△	#24	18H	FFFF9CH			
PPG timer 0	△	#25	19H	FFFF98H	ICR07	0000B7H *1	
DTP/external interrupt - ch 6/7 detected	△	#26	1AH	FFFF94H			
PPG timer 1	△	#27	1BH	FFFF90H	ICR08	0000B8H *1	
Reload timer 1	△	#28	1CH	FFFF8CH			
PPG timer 2	○	#29	1DH	FFFF88H	ICR09	0000B9H *1	
Real time clock timer	×	#30	1EH	FFFF84H			
Free-run timer over flow	×	#31	1FH	FFFF80H	ICR10	0000BAH *1	
A/D converter conversion end	○	#32	20H	FFFF7CH			
Free-run timer clear	×	#33	21H	FFFF78H	ICR11	0000BBH *1	
Sound generator	×	#34	22H	FFFF74H			
Time base timer	×	#35	23H	FFFF70H	ICR12	0000BCH *1	
Clock timer (sub clock)	×	#36	24H	FFFF6CH			
UART 1 RX	◎	#37	25H	FFFF68H	ICR13	0000BDH *1	
UART 1 TX	△	#38	26H	FFFF64H			
UART 0 RX	◎	#39	27H	FFFF60H	ICR14	0000BEH *1	
UART 0 TX	△	#40	28H	FFFF5CH			
Flash memory status	×	#41	29H	FFFF58H	ICR15	0000BFH *1	
Delayed interrupt generator module	×	#42	2AH	FFFF54H			Low

◎ : Compatible, with EI^{2}OS stop function
○ : Compatible
△ : Compatible when interrupt sources sharing ICR are not in use
× : Not compatible

*1 : • Peripheral functions sharing the ICR register have the same interrupt level.
• If peripheral functions sharing the ICR register are using expanded intelligent I/O services, one or the other cannot be used.
• When peripheral functions are sharing the ICR register and one specifies expanded intelligent I/O services, the interrupt from the other function cannot be used.

*2 : Priority applies when interrupts of the same level are generated.

■ PERIPHERAL FUNCTIONS

1. I/O Ports

The I/O ports function is to send data from the CPU to be output from I/O pins and load input signals at the I/O pins into the CPU, according to the port data register (PDR) . Port input/output at I/O pins can be controlled in bit units by the port direction register (DDR) as required. The following list shows each of the functions as well as the shared peripheral function for each port.

- Port 0 : General purpose I/O port, shared with peripheral functions (external interrupt/UART/PPG)
- Port 1 : General purpose I/O port, shared with peripheral functions (PPG/reload timer/clock timer/ICU)
- Port 3 : General purpose I/O port, shared with peripheral functions (LCD)
- Port 4 : General purpose I/O port, shared with peripheral functions (LCD)
- Port 5 : General purpose I/O port, shared with peripheral functions (External interrupt/CAN/SG)
- Port 6 : General purpose I/O port, shared with peripheral functions (A/D converter)
- Port 7 : General purpose I/O port, shared with peripheral functions (Stepping motor controller)
- Port 8 : General purpose I/O port, shared with peripheral functions (Stepping motor controller)
- Port 9 : General purpose I/O port, shared with peripheral functions (LCD)

(1) List of Functions

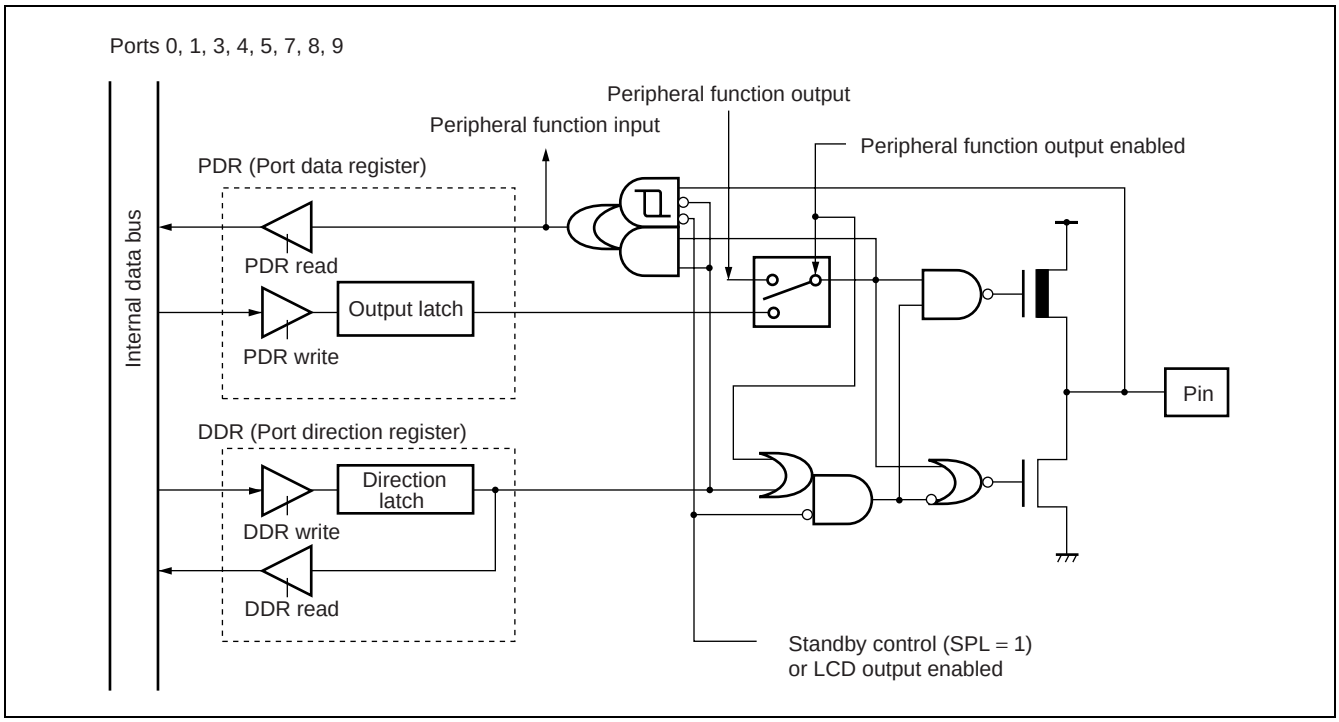
Port	Pin name	Input format	Output format	Function	bit15	bit14	bit13	bit12
Port 0	P00/SIN0/INT4 to P07/PPG1	CMOS (hysteresis)	CMOS	General purpose I/O port	—	—	—	—
				Peripheral function	—	—	—	—
					—	—	—	—
Port 1	P10/PPG2 to P15/IN0			General purpose I/O port	—	—	P15	P14
				Peripheral function	—	—	IN0	IN1
					—	—	—	—
Port 3	P36/SEG12 to P37/SEG13			General purpose I/O port	P37	P36	—	—
				Peripheral function	SEG13	SEG12	—	—
Port 4	P40/SEG14 to P47/SEG21			General purpose I/O port	—	—	—	—
				Peripheral function	—	—	—	—
Port 5	P50/INT0 to P57/SGA			General purpose I/O port	P57	P56	P55	P54
				Peripheral function	SGA	SGO	RX0	TX0
					—	FRCK	—	—
Port 6	P60/AN0 to P67/AN7	Analog CMOS (hysteresis)		General purpose I/O port	—	—	—	—
				Peripheral function	—	—	—	—
Port 7	P70/PWM1P0 to P77/PWM2M1	CMOS (hysteresis)		General purpose I/O port	P77	P76	P75	P74
				Peripheral function	PWM2M1	PWM2P1	PWM1M1	PWM1P1
Port 8	P80/PWM1P2 to P87/PWM2M3			General purpose I/O port	—	—	—	—
				Peripheral function	—	—	—	—
Port 9	P90/SEG22 to P91/SEG23			General purpose I/O port	—	—	—	—
				Peripheral function	—	—	—	—

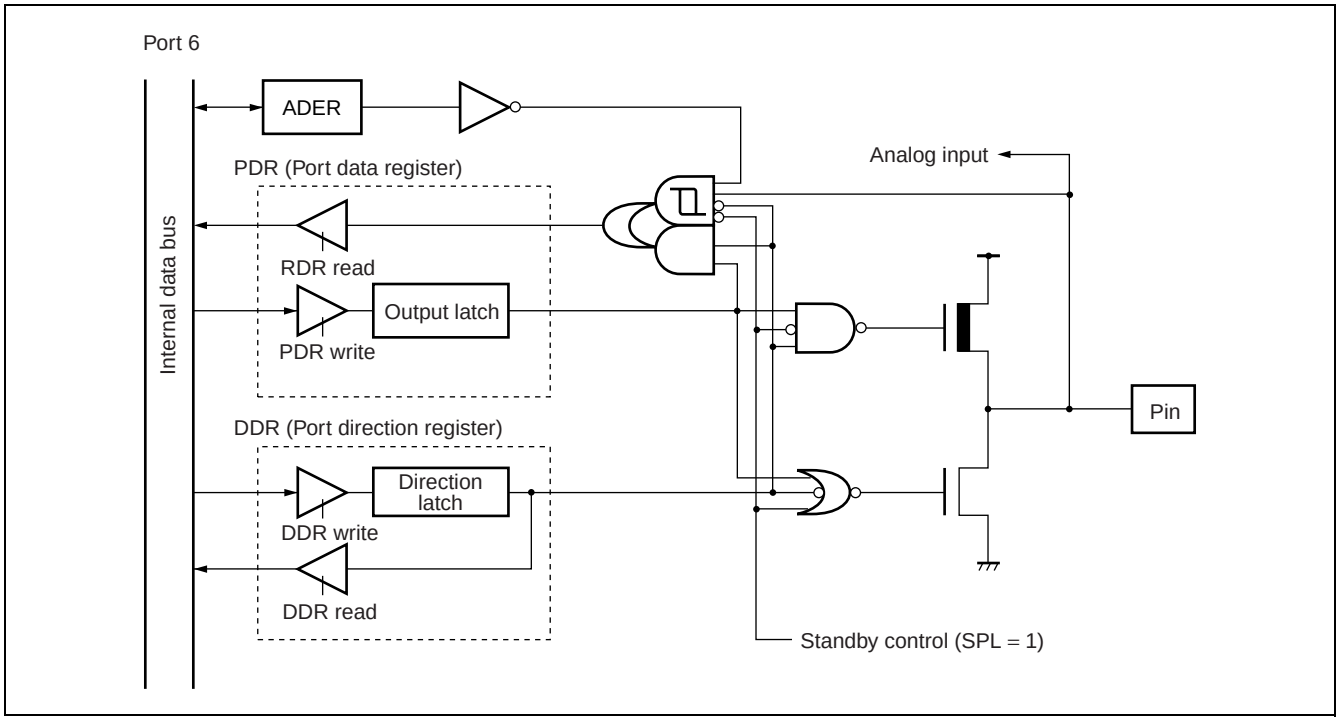
(Continued)

Port	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Port 0	—	—	—	—	P07	P06	P05	P04	P03	P02	P01	P00
	—	—	—	—	PPG1	PPG0	SCK1	SOT1	SIN1	SCK0	SOT0	SIN0
	—	—	—	—	TIN1	TOT1	—	—	INT7	INT6	INT5	INT4
Port 1	P13	P12	P11	P10	—	—	—	—	—	—	—	—
	IN2	IN3	WOT	PPG2	—	—	—	—	—	—	—	—
	—	TIN0	TOT0	—	—	—	—	—	—	—	—	—
Port 3	—	—	—	—	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—	—	—	—	—
Port 4	—	—	—	—	P47	P46	P45	P44	P43	P42	P41	P40
	—	—	—	—	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG15	SEG14
Port 5	P53	P52	P51	P50	—	—	—	—	—	—	—	—
	INT3	INT2	INT1	INT0	—	—	—	—	—	—	—	—
	—	TX1	RX1	—	—	—	—	—	—	—	—	—
Port 6	—	—	—	—	P67	P66	P65	P64	P63	P62	P61	P60
	—	—	—	—	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
Port 7	P73	P72	P71	P70	—	—	—	—	—	—	—	—
	PWM2M0	PWM2P0	PWM1M0	PWM1P0	—	—	—	—	—	—	—	—
Port 8	—	—	—	—	P87	P86	P85	P84	P83	P82	P81	P80
	—	—	—	—	PWM2M3	PWM2P3	PWM1M3	PWM1P3	PWM2M2	PWM2P2	PWM1M2	PWM1P2
Port 9	—	—	P91	P90	—	—	—	—	—	—	—	—
	—	—	SEG23	SEG22	—	—	—	—	—	—	—	—

Note : Port 6 also functions as an analog input pin. When using this port as a general purpose port, always write “0” to the corresponding analog input enable register (ADER) bit. The ADER bit is initialized to “1” at reset.

(2) Block Diagrams





2. Watchdog Timer/Time Base Timer/Clock Timer

The watchdog timer, timer base timer, and clock timer have the following circuit configuration.

- Watchdog timer : Watchdog counter, control register, watchdog reset circuit
- Time base timer : 18-bit timer, interval interrupt control circuit
- Clock timer : 15-bit timer, interval interrupt control circuit

(1) Watchdog timer function

The watchdog timer is composed of a 2-bit watchdog counter that uses the carry signal from the 18-bit time base timer or 15-bit clock timer as a clock source, plus a control register and watchdog reset control circuit.

After startup, this function will reset the CPU if not cleared within a given time.

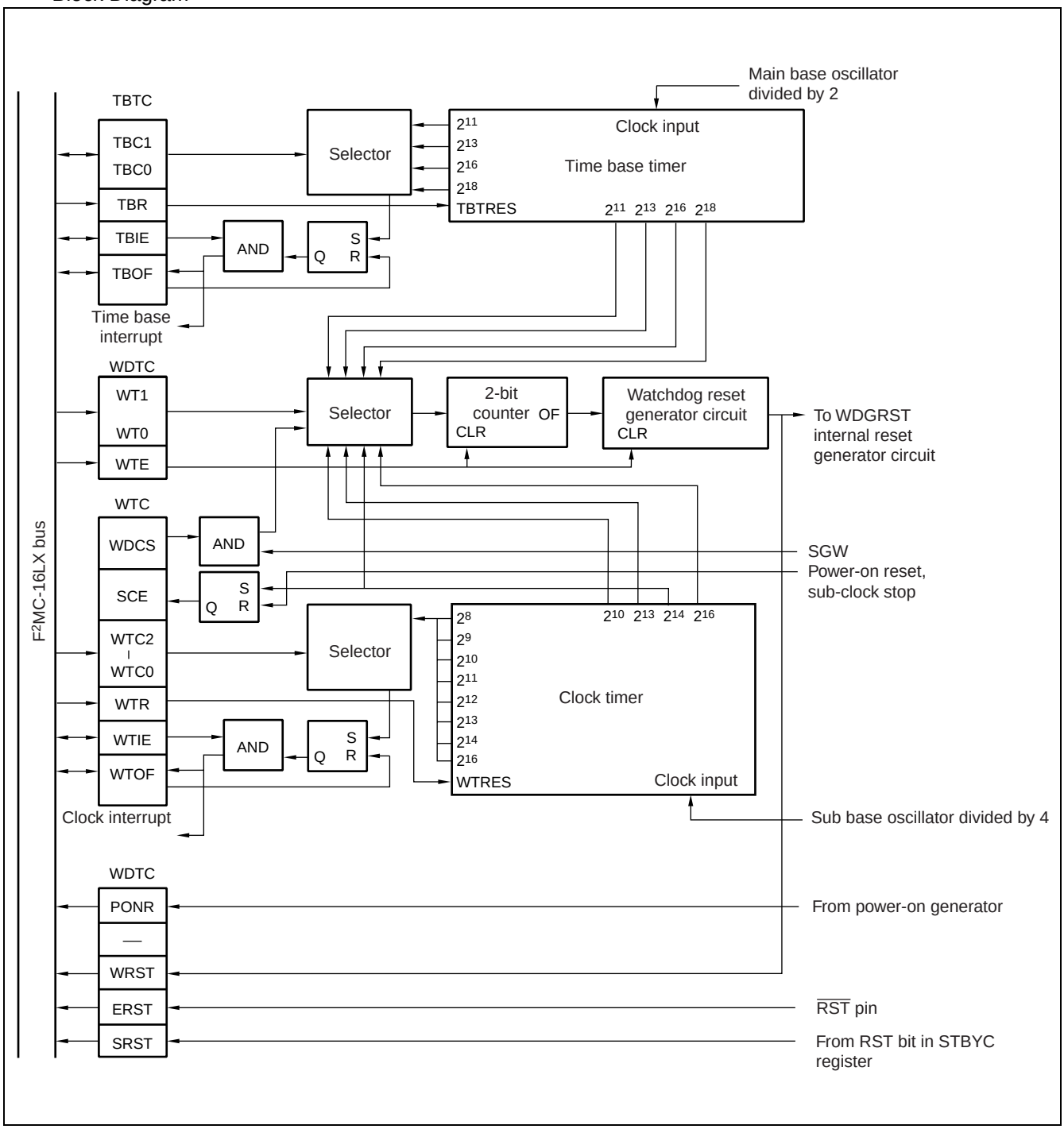
(2) Time base timer function

The time base timer is an 18-bit free-run counter (time base counter) synchronized with the internal count clock (base oscillator divided by 2) , with an interval timer function providing a selection of four interval times. Other functions include a timer output for an oscillator stabilization wait time and clock feed to the watchdog timer or other operating clocks. Note that the time base timer uses the main clock regardless of the setting of the MCS bit or SCS bit in the CKSCR register.

(3) Clock timer function

The clock timer provides functions including a clock source for the watchdog timer, a sub clock base oscillator stabilization wait timer, and an interval timer to generate an interrupt at fixed intervals. Note that the clock timer uses the sub clock regardless of the setting of the MCS bit or SCS bit in the CKSCR register.

- Block Diagram



3. Input Capture

This circuit is composed of a 16-bit free-run timer and four 16-bit input capture circuits.

(1) Input capture (× 4)

The input capture circuits consist of four independent external input pins and corresponding capture registers and control registers. When the specified edge of the external signal input (at the input pin) is detected, the value of the 16-bit free-run timer is saved in the capture register, and at the same time an interrupt can also be generated.

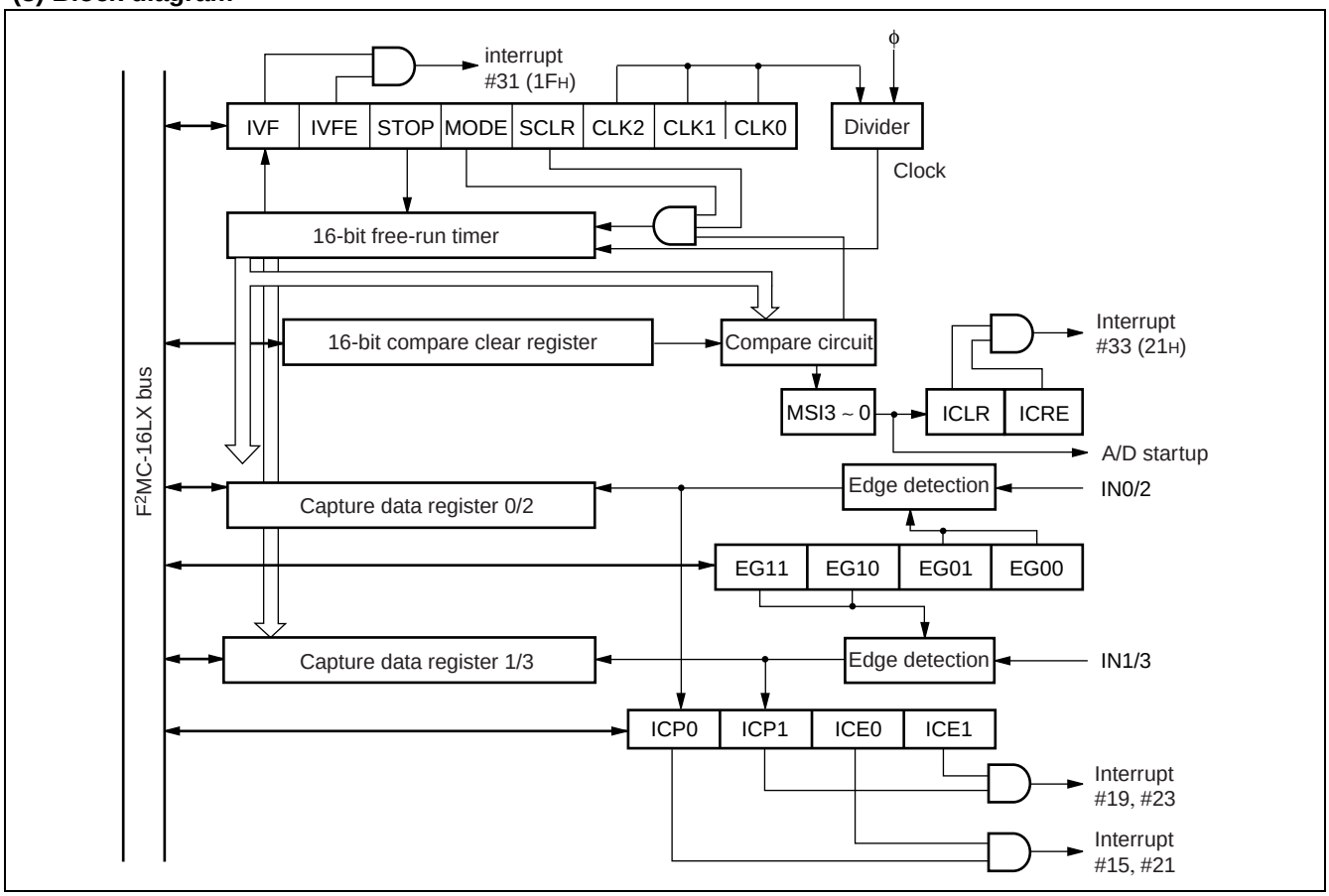
- The valid edge (rising edge, falling edge, both edges) of the external signal can be selected.
- The four input capture circuits can operate independently.
- The interrupt can be generated from the valid edge of the external input signal.

(2) 16-bit free-run timer (× 1)

The 16-bit free-run timer is composed of a 16-bit up-counter, control register, 16-bit compare register, and prescaler. The output values from this counter are used as the base time for the input capture circuits.

- The counter clock operation can be selected from 8 options. The eight internal clock settings are ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64, ϕ/128 where ϕ represents the machine clock cycle.
- Interrupts can be generated from overflow events, or from compare match events with the compare register. (Compare match operation requires a mode setting.)
- The counter value can be initialized to "0000H" by a reset, soft clear, or a compare match with the compare register.

(3) Block diagram



4. 16-bit Reload Timer

The 16-bit reload timer can either count down in synchronization with three types of internal clock signals in internal clock mode, or count down at the detection of the designated edge of an external signal. The user may select either function. This timer defines a transition from 0000_H to $FFFF_H$ as an underflow event. Thus an underflow occurs when counting from the value [Reload register setting + 1].

A selection of two counter operating modes are available. In reload mode, the counter is reset to the count value and continues counting after an underflow, and in one-shot mode the count stops after an underflow. The counter can generate an interrupt when an underflow occurs, and is compatible with the expanded intelligent I/O services (EI^2OS) .

(1) 16-bit Reload timer operating modes

Clock mode	Counter mode	16-bit reload timer operation
Internal clock mode	Reload mode	Soft trigger operation External trigger operation External gate input operation
	One-shot mode	
Event count mode (external clock mode)	Reload mode	Soft trigger operation
	One-shot mode	

(2) Internal clock mode

One of three input clocks is selected as the count clock, and can be used in one of the following operations.

- Soft trigger operation
  When “1” is written to the TRG bit in the timer control status register (TMCSR0/1) , the count operation starts.Trigger input at the TRG bit is normally valid with an external trigger input, as well as an external gate input.
- External trigger operation
  Count operation starts when a selected edge (rising, falling, both edges) is input at the TIN0/1 pin.
- External gate input operation
  Counting continues as long as the selected signal level (“L” or “H”) is input at the TIN0/1 pin.

(3) Event count mode (External clock mode)

In this mode a down count event occurs when a selected valid edge (rising, falling, both edges) is input at the TIN0/1 pin. This function can also be used as an interval timer when an external clock with a fixed period is used.

(4) Counter operation

- Reload mode

In down count operation, when an underflow event (transition from “0000_H” to “$FFFF_H$”) occurs, the set count value is reloaded and count operation continues. The function can be used as an interval timer by generating an interrupt request at each underflow event. Also, a toggle waveform that inverts at each underflow can be output from the TOT0/1 pin.

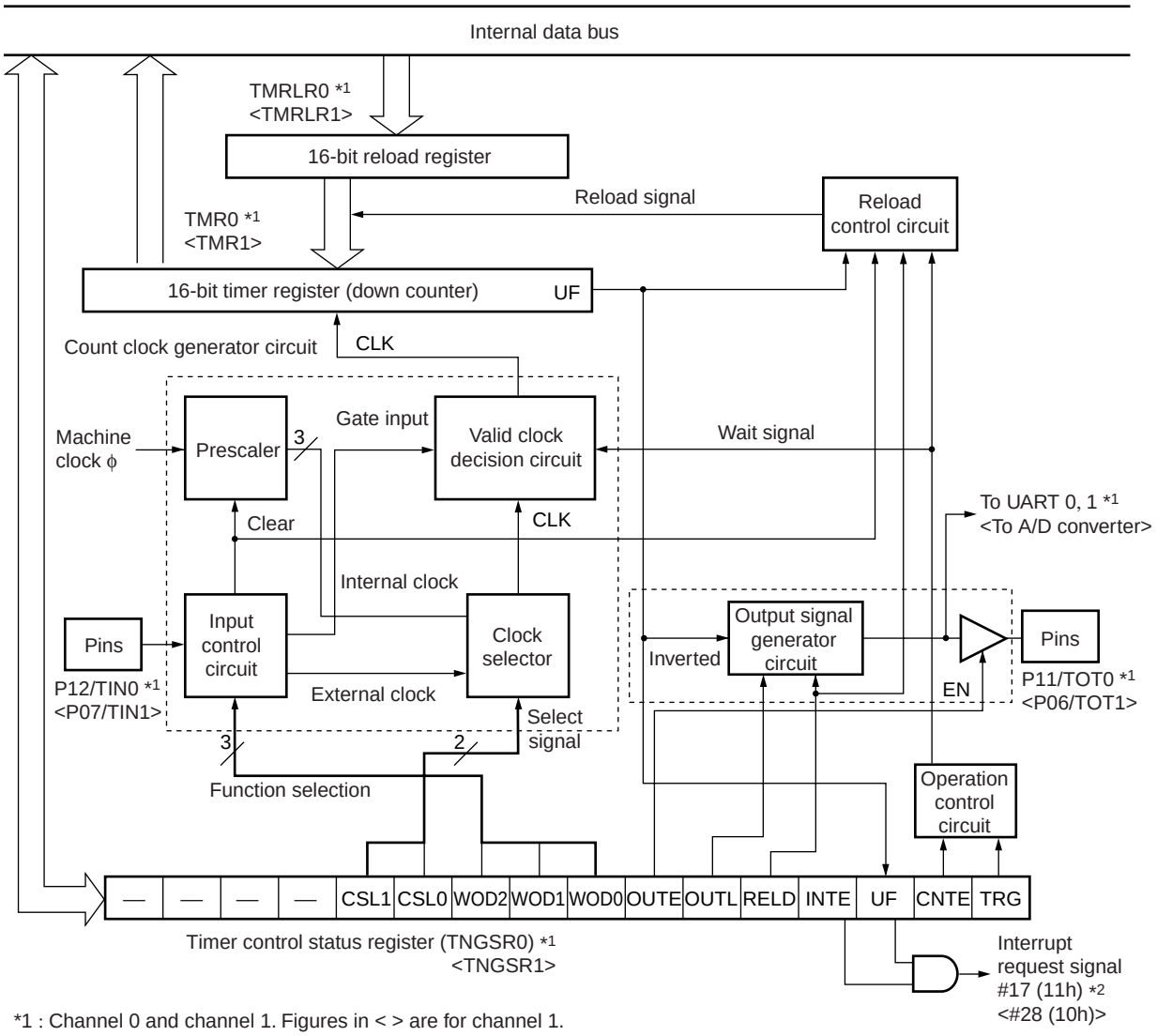
Counter clock	Counter clock period	Interval time
Internal clock	$2^1/\phi$ (0.125 µs)	0.125 µs to 8.192 ms
	$2^3/\phi$ (0.5 µs)	0.5 µs to 32.768 ms
	$2^5/\phi$ (2.0 µs)	2.0 µs to 131.1 ms
External clock	$2^3/\phi$ or greater (0.5 µs)	0.5 µs or greater

ϕ : Machine clock cycle. Figures in ( ) are values at machine clock frequency 16 MHz.

(5) One-shot mode

In down count operation, the count stops when an underflow event (transition from "0000H" to "FFFFH") occurs. This function can generate an interrupt at each underflow. While the counter is operating, a rectangular wave form indicating that the count is in progress can be output form the TOT0 and TOT1 pins.

(6) Block diagram

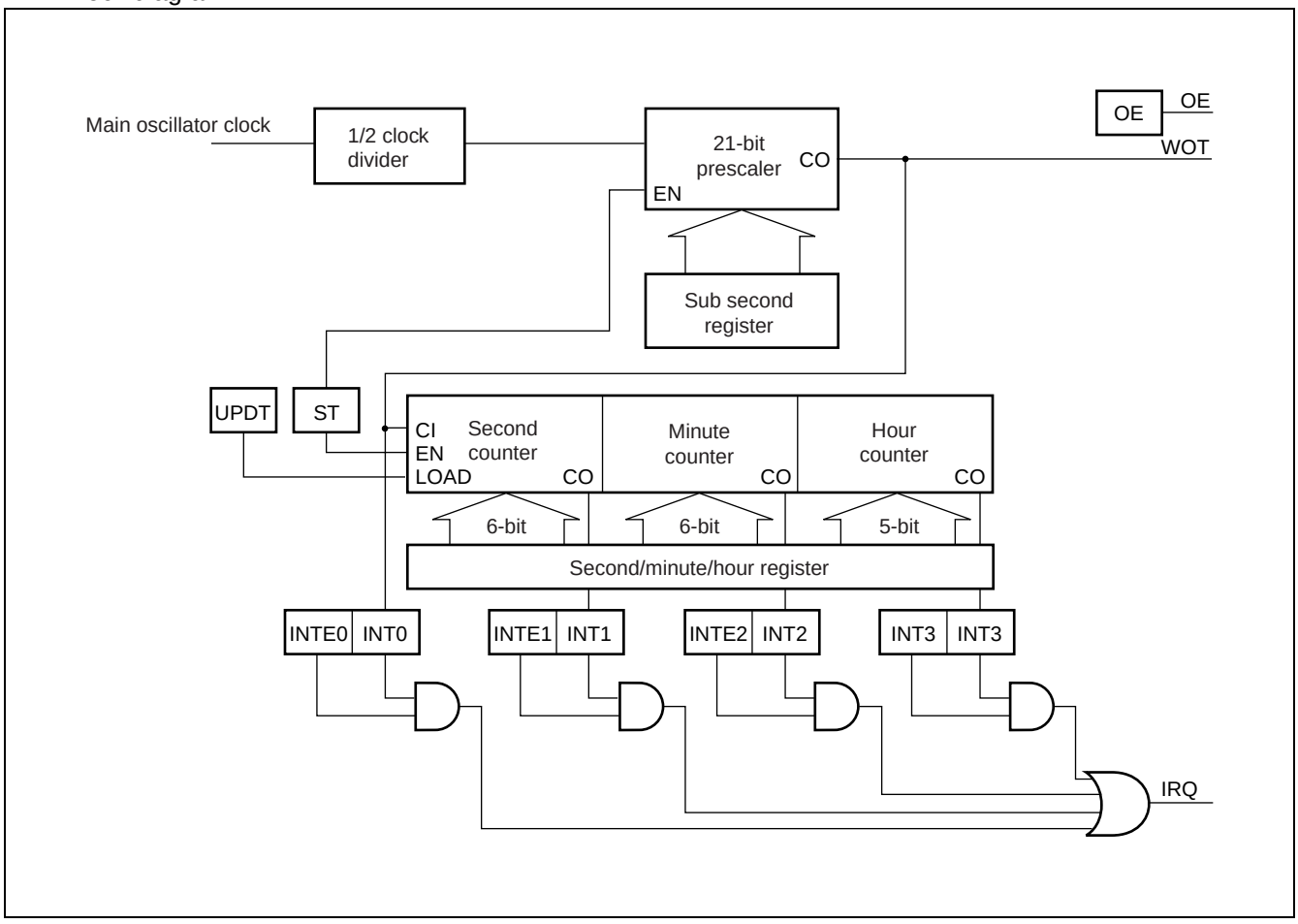


*1 : Channel 0 and channel 1. Figures in < > are for channel 1.
*2 : Interrupt number

5. Real Time Clock Timer

The real time clock timer is composed of a real time clock timer control register, sub second data register, second/minute/hour data registers, 1/2 clock divider, 21-bit prescaler and second/minute/hour counters. Because the MCU oscillation frequency operates on a given real time clock timer operation, a 4 MHz frequency is assumed. The real time clock timer operates as a real world timer and provides real world time information.

- Block diagram



6. PPG Timer

The PPG timer consists of a prescaler, one 16-bit down-counter, 16-bit data register with buffer for period setting, and 16-bit compare register with buffer for duty setting, plus pin control circuits.

The timer can output pulses synchronized with an externally input soft trigger. The period and duty of the output pulse can be adjusted by rewriting the values in the two 16-bit registers.

(1) PWM function

Programmable to output a pulse, synchronized with a trigger.

Can also be used as a D/A converter with an external circuit.

(2) One-shot function

Detects the edge of a trigger input, and outputs a single pulse.

(3) Pin control

- Set to “1” at a duty match (priority) .
- Reset to “0” at a counter borrow event
- Has a fixed output mode to output a simple all “L” (or “H”) signal.
- Polarity can be specified

(4) 16-bit down counter

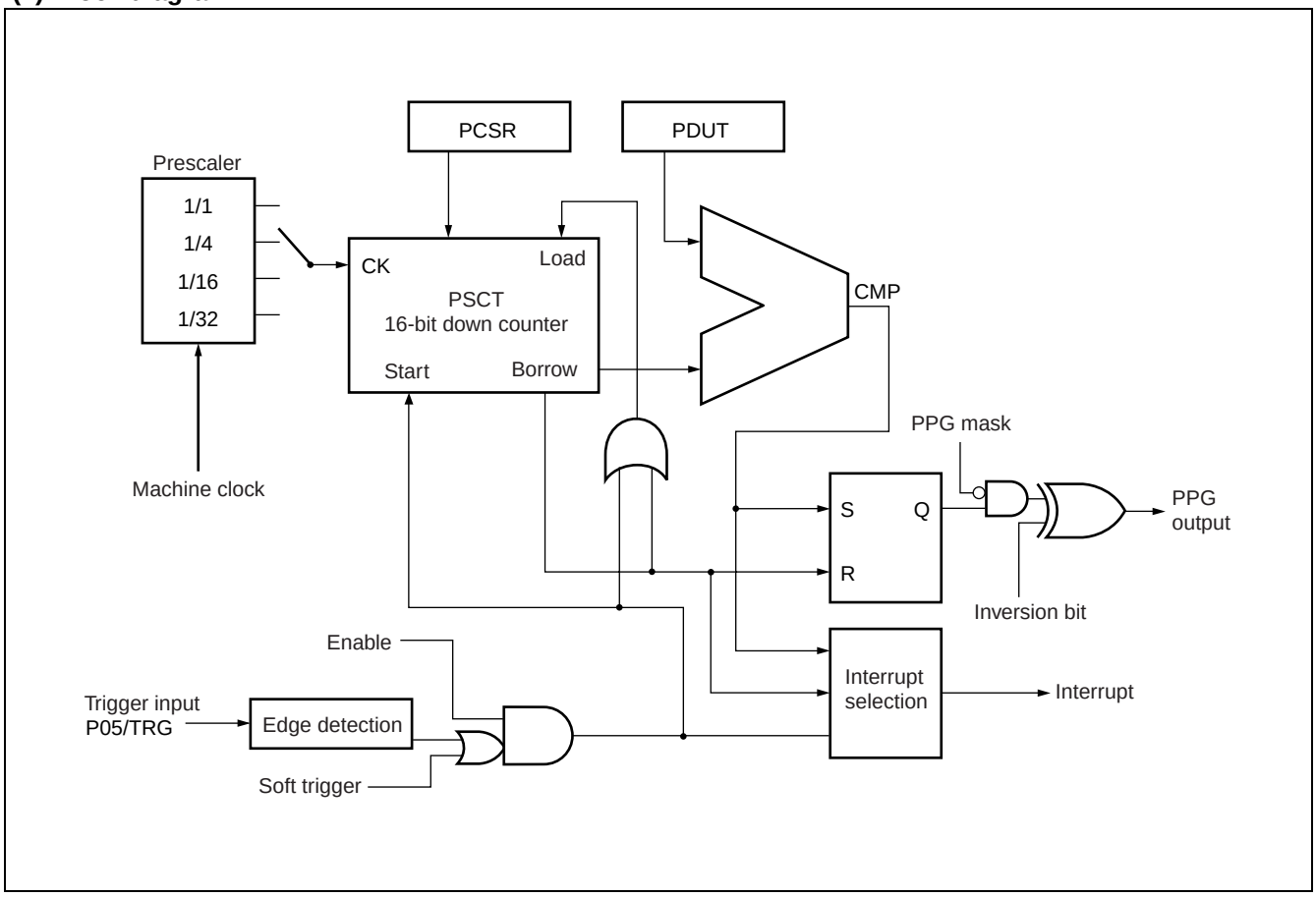
- Select from four types of counter operation clocks. Four internal clocks (ϕ, ϕ/4, ϕ/16, ϕ/64) ϕ : Machine clock cycles.
- The counter value can be initialized to “$FFFF_H$” at a reset or counter borrow event.

(5) Interrupt requests

- Timer startup
- Counter borrow event (period match)
- Duty match event
- Counter borrow event (period match) or duty match event

(6) Multiple channels can be set to start up at an external trigger, or to restart during operation.

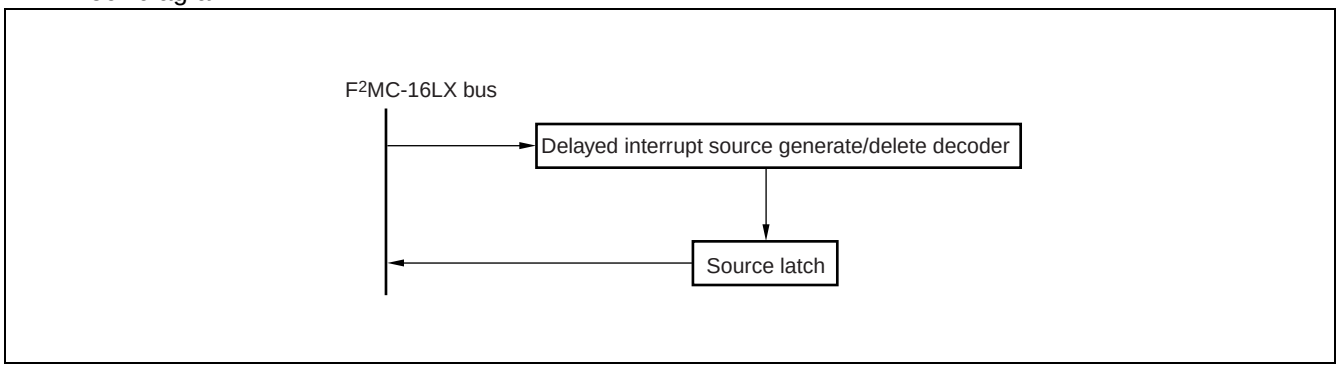
(7) Block diagram



7. Delayed Interrupt Generator Module

The delayed interrupt generator module is a module that generates interrupts for task switching. This module makes it possible to use software to generate/cancel interrupt requests to the F²MC-16LX CPU.

- Block diagram



8. DTP/External Interrupt Circuit

The DTP (Data transfer peripheral) /external interrupt circuit is located between an externally connected peripheral device and the F²MC-16LX CPU and sends interrupt requests or data transfer requests generated from the peripheral device to the CPU, thereby generating external interrupt requests or starting the expanded intelligent I/O services (EI²OS) .

(1) DTP/external interrupt function

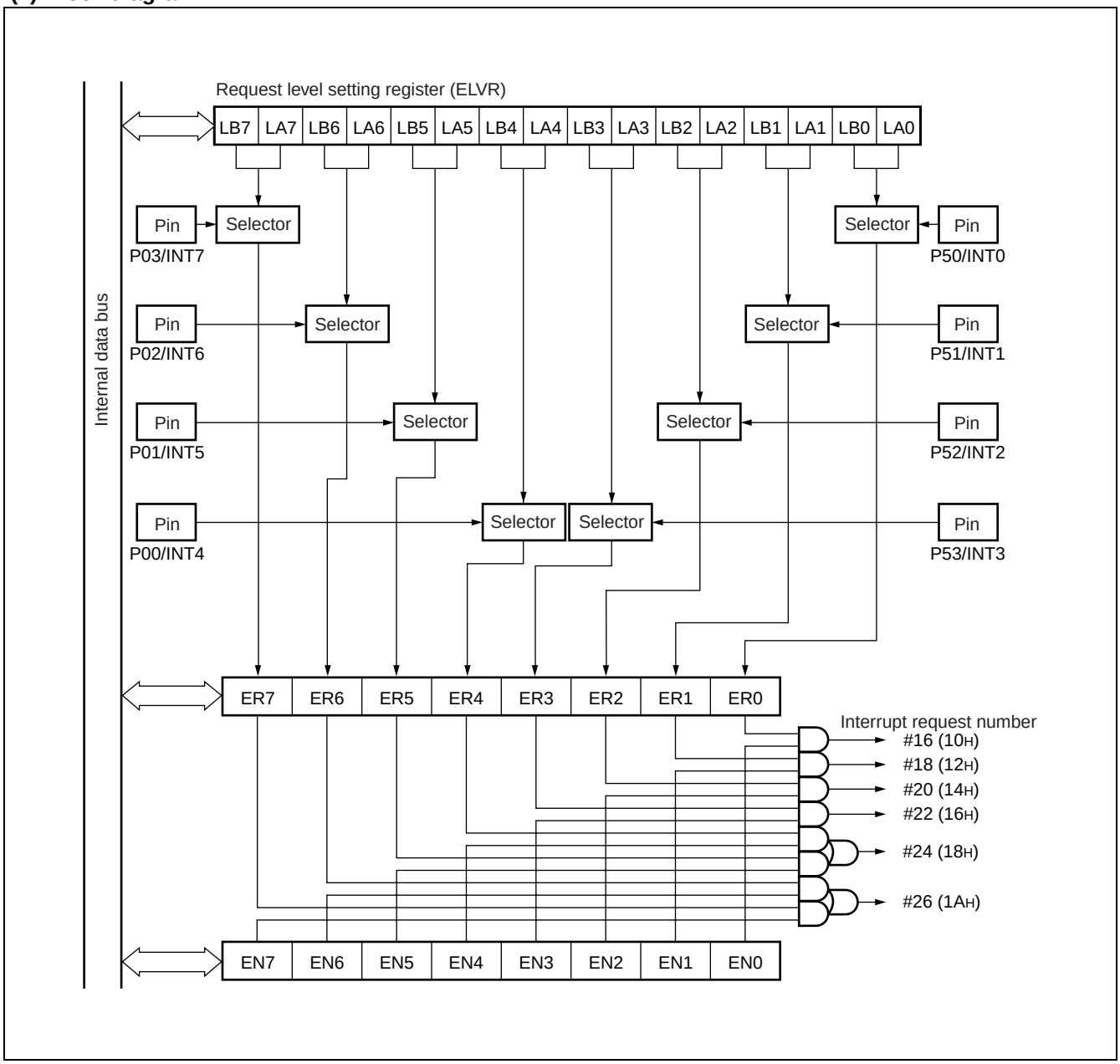
The DTP/external interrupt function uses a signal input from the DTP/external interrupt pin as a startup source. And it is accepted by the CPU by the same procedure as a normal hardware interrupt, and can generate an external interrupt or start the expanded intelligent I/O service (EI²OS) .

When the interrupt is accepted by the CPU, if the corresponding expanded intelligent I/O service (EI²OS) is prohibited the interrupt operates as an external interrupt function and branches to an interrupt routine. If the EI²OS is permitted the interrupt functions as a DTP function, using EI²OS for automatic data transfer, then branching to an interrupt routine after the completion of the specified number of data transfers.

	External interrupt	DTP function
Input pins	8 pins (P50/INT0 to P53/INT3, P00/INT4 to P03 INT7)	
Interrupt sources	Request level setting register (ELVR) sets the detection level, or selected edge for each pin	
	“H” level/ “L” level/ rising edge/falling edge input	“H” level/ “L” level input
Interrupt numbers	#16 (10_H) , #18 (12_H) , #20 (14_H) , #22 (16_H) , #24 (18_H) , #26 ($1A_H$)	
Interrupt control	DTP/interrupt enable register (ENIR) permits/prohibits interrupt request output	
Interrupt flags	DTP/interrupt enable register (EIRR) stores interrupt sources	
Process selection	When EI²OS prohibited (ICR : ISE = 0)	When EI²OS is enabled (ICR : ISE = 1)
Processing	Branch to external interrupt processing routine	EI²OS performs automatic data transfer, then after a specified number of cycles, branches to an interrupt routine

ICR : Interrupt control register

(2) Block diagram



9. 8/10-bit A/D Converter

The 8/10-bit A/D converter has functions for using RC sequential comparator conversion format to convert analog input voltage into 10-bit or 8-bit digital values. The input signal is selected from 8-channel analog input pins, and the conversion start can be selected from three types : by software, 16-bit reload timer 1 or a trigger input from an external signal pin.

(1) 8/10-bit A/D converter functions

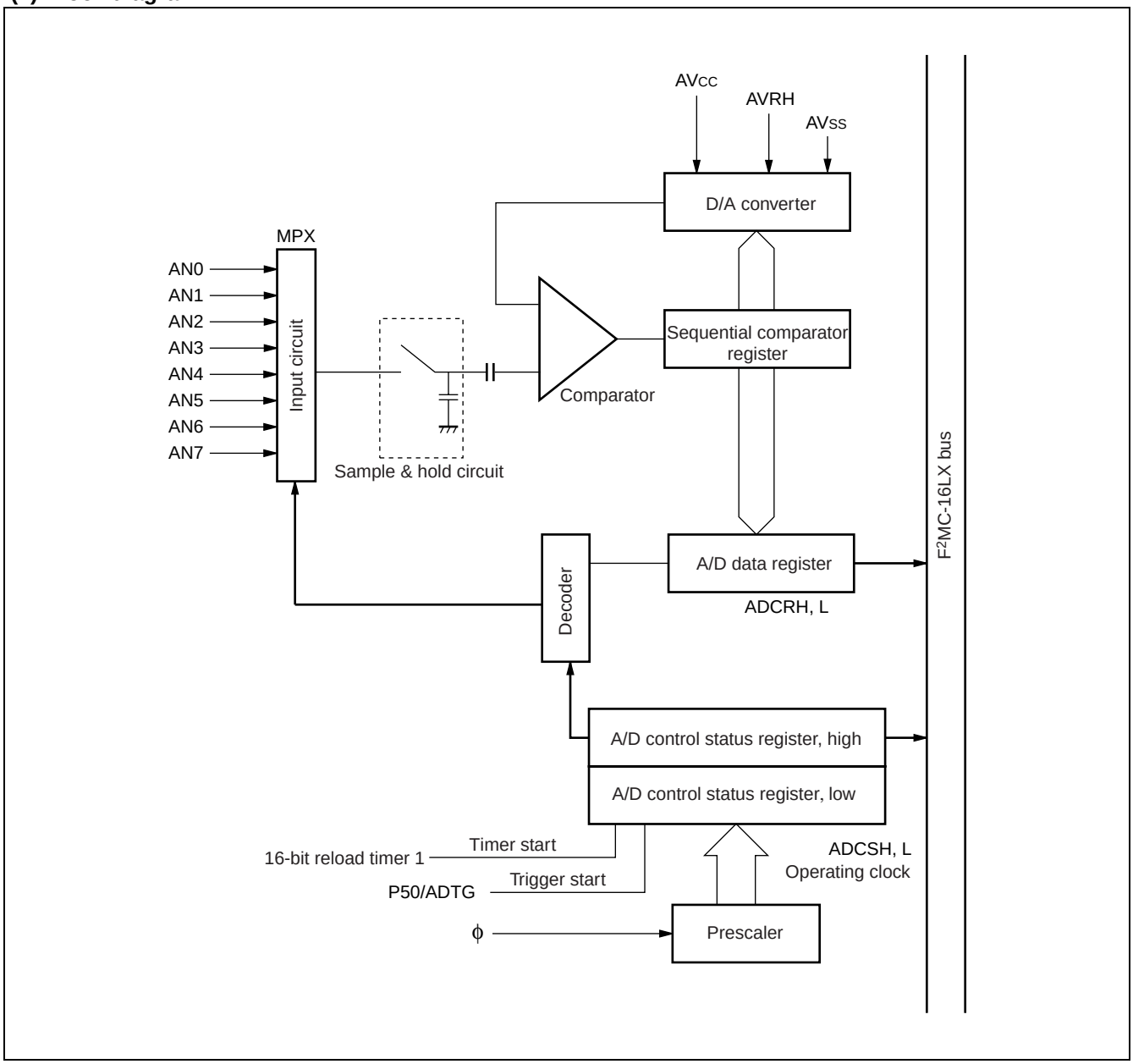
The A/D converter takes analog voltage signals (input voltage) input at analog input pins, and converts these to digital values, providing the following features.

- Minimum conversion time is 6.13 µs (at machine clock frequency of 16 MHz, including sampling time) .
- Minimum sampling time is 3.75 µs (at machine clock 16 MHz)
- The conversion method is an RC sequential conversion in comparison with a sample hold circuit.
- Either 10-bit or 8-bit resolution can be selected.
- The analog input pin can select from 8 channels by a program setting.
- At completion of A/D conversion, an interrupt request can be generated, or EI^{2}OS can be started.
- Because the conversion data protection function operates in an interrupt enabled state, no data is lost even in continuous conversion.
- The conversion start source may be selected from : software, 16-bit reload timer 1 (rising edge) , or external trigger input (falling edge) .

Three conversion modes are available

Conversion mode	Single conversion operation	Scan conversion operation
Single conversion mode	Converts the specified channel (1 channel only) one time, then stops.	Converts multiple consecutive channels (up to 8 channels may be specified) one time, then stops.
Continuous conversion mode	Converts the specified channel (1 channel only) repeatedly.	Converts multiple consecutive channels (up to 8 channels may be specified) repeatedly.
Stop conversion mode	Converts the specified channel (1 channel only) one time, then pauses, waits until the next start is applied.	Converts multiple consecutive channels (up to 8 channels may be specified) , however pauses after conversion of each channel, waits until the next start is applied.

(2) Block diagram



10. UART

The UART is a general purpose serial data communication interface for synchronous communication, or asynchronous (start-stop synchronized) communication with external devices. Functions include normal bi-directional functions, as well as master/slave type communication functions (multi-processor mode : master side only supported) .

(1) UART Functions

The UART is a general purpose serial data communication interface for sending and receiving of serial data with other CPU's or peripheral devices, and provides the following functions.

	Functions
Data buffer	Full duplex double buffer
Transfer modes	• Clock synchronous (no start/stop bits) • Clock asynchronous (start-stop synchronized)
Baud rate	• Exclusive baud rate generator provides a selection of 8 rates • External clock input enabled • Internal clock (can use internal clock feed from 16-bit reload timer)
Data length	• 7-bit (asynchronous normal mode only) • 8-bit
Signal type	NRZ (Non return to zero)
Receiving error detection	• Framing errors • Overrun errors • Parity errors (not enabled in multiprocessor mode)
Interrupt request	• Receiving interrupt (receiving completed, receiving error detection) • Sending interrupt (sending completed) • Sending/receiving both compatible with expanded intelligent I/O services (EI^2OS)
Master/slave type communication function (multi-processor mode)	1 (master) -to-n (slave) communication enabled (only master side supported) .

Note : The UART in clock synchronous transfer does not add start bits or stop bits, but transfers data only.

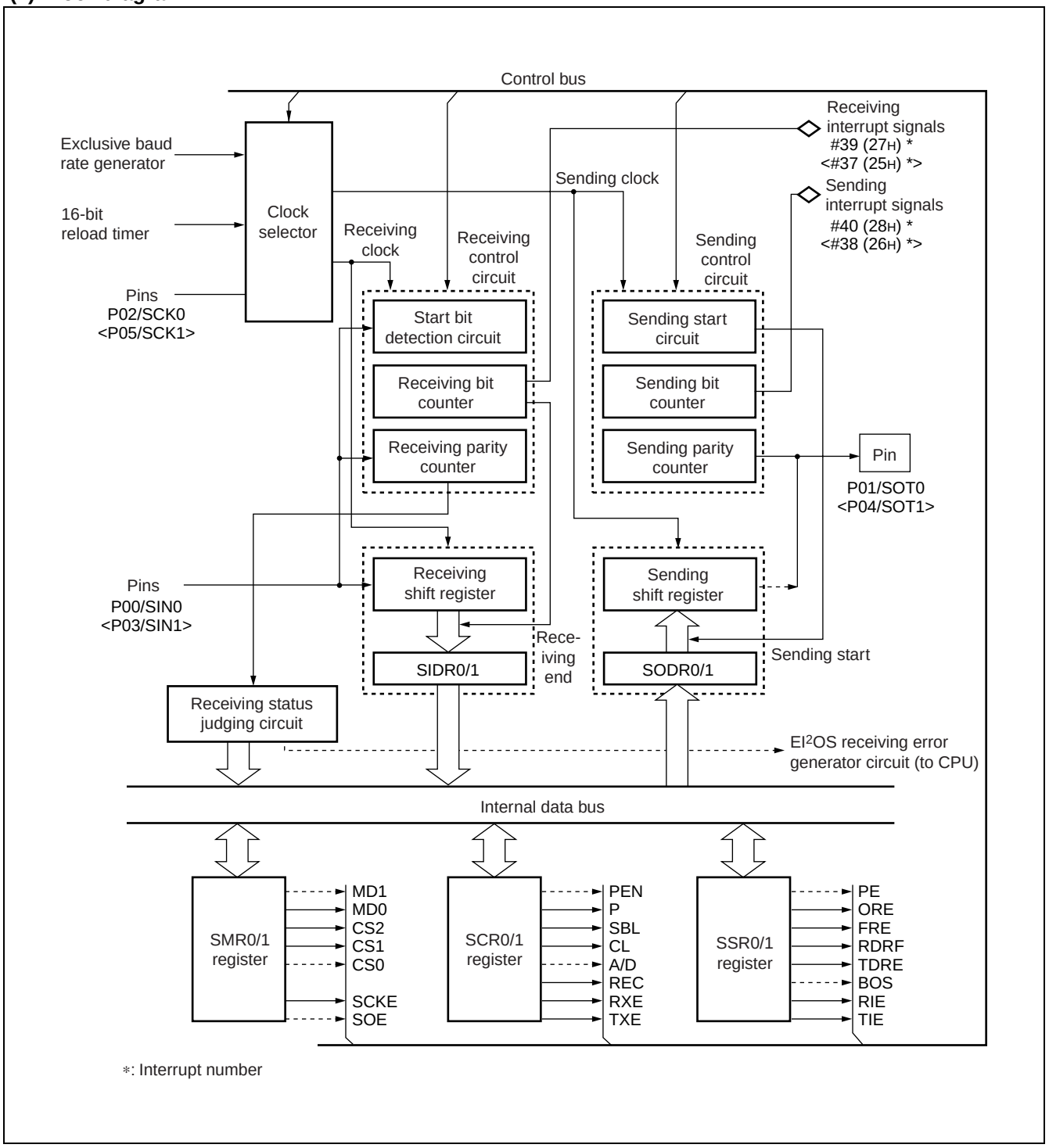
Operating mode		**Data length**		**Synchronization**	**Stop bit length**
		No parity	**Parity**		
0	Normal mode	7-bit or 8-bit		Asynchronous	1-bit or 2-bit *2
1	Multi-processor mode	8 + 1 *1	—	Asynchronous	
2	Normal mode	8	—	Synchronous	None

— : Setting not available

*1 : "+" indicates an address/data selection bit (A/D) for communication control.

*2 : In receiving only one stop bit is detected.

(2) Block diagram



11. CAN Controller

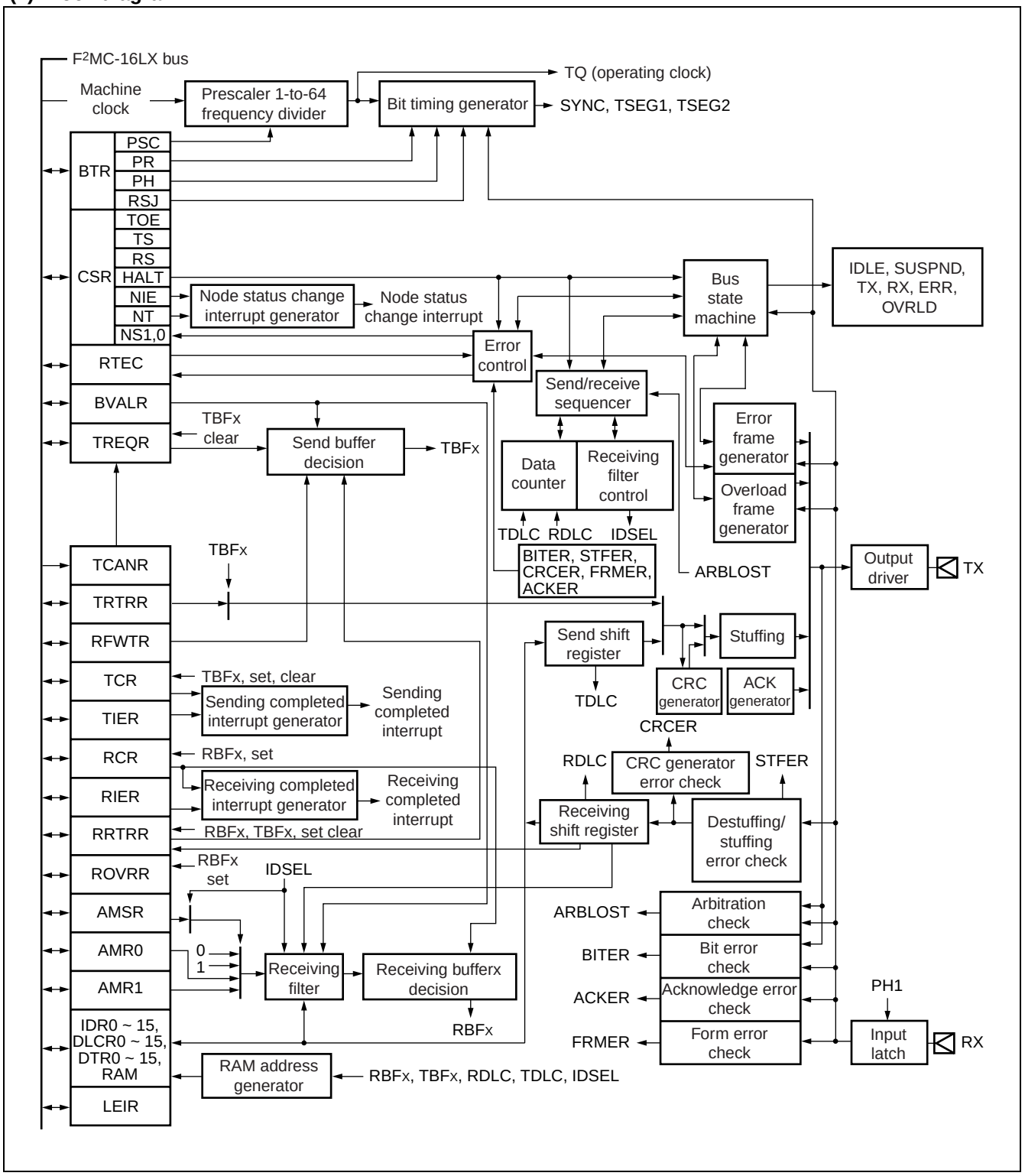
The CAN controller is a self-contained module within a 16-bit microcomputer (F^2MC-16LX) . The CAN (controller area network) controller is the standard protocol for serial transmissions among automotive controllers and is widely used in the industry.

(1) CAN controller features

The CAN controller has the following features.

- Conforms to CAN specifications version 2.0 A and B.
  Supports sending and receiving in standard frame and expanded frame format.
- Supports data frame sending by means of remote frame receiving.
- 16 sending/receiving message buffers
  29-bit ID and 8-byte data
  Multi-level message buffer configuration
- Supports full bit compare, full bit mask as well as partial bet mask filtering.
  Provides two receiving mask registers for either standard frame or expanded frame format.
- Bit speed programmable from 10 KB/s to 1 MB/s (at machine clock 16 MHz)
- CAN WAKE UP function
- The MB90420G (A) series has a two-channel built-in CAN controller. The MB90425G (A) series has a 1-channel built-in CAN controller.

(2) Block diagram



12. LCD Controller/Driver

The LCD controller/driver has a built-in 16 × 8-bit display data memory, and controls the LCD display by means of four common outputs and 24 segment outputs. A selection of three duty outputs are available. This block can drive an LCD (liquid crystal display) panel directly.

(1) LCD controller/driver functions

The LCD controller/driver provides functions for directly displaying the contents of display data memory (display RAM) on the LCD panel by means of segment output and common output.

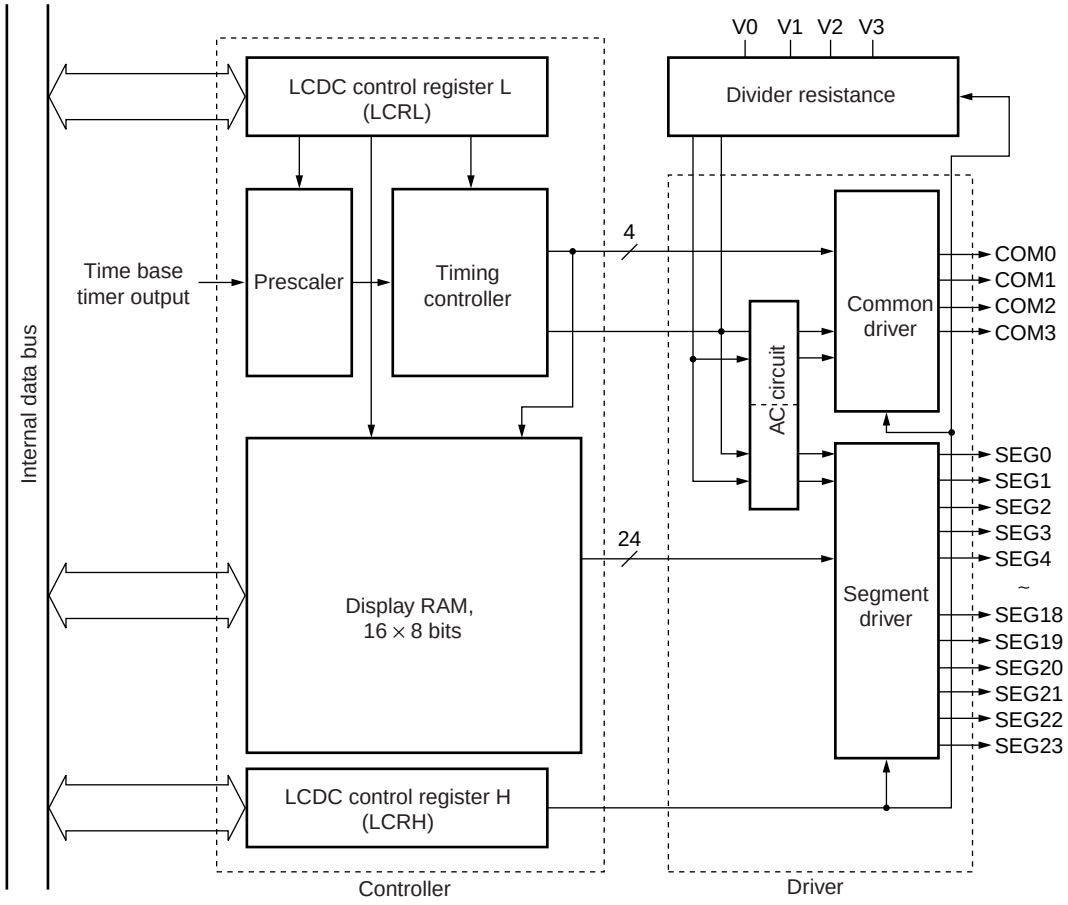
- LCD drive voltage divider resistance is built-in. External divider resistance can also be connected.
- Up to 4 common outputs (COM0 to COM3) and 24 segment outputs (SEG0 to SEG23) can be used.
- 16-byte display data memory (display RAM) is built-in.
- The duty can be selected at 1/2, 1/3, 1/4 (limited by bias setting) .
- Drives the LCD directly.

Bias	1/2 duty	1/3 duty	1/4 duty
1/2 bias	○	×	×
1/3 bias	×	○	○

○ : Recommended mode
× : Use prohibited

Note : When the SEG12 to SEG23 pins have been selected as general purpose ports by the LCRH setting, they cannot be used for segment output.

(2) Block diagram



13. Low voltage/Program Looping Detection Reset Circuit

The Low voltage detection reset circuit is a function that monitors power supply voltage in order to detect when a voltage drops below a given voltage level. When a low voltage condition is detected, an internal reset signal is generated.

The Program Looping detection reset circuit is a count clock with a 20-bit counter that generates an internal reset signal if not cleared within a given time after startup.

(1) Low voltage detection reset circuit

Detection voltage
4.0 V ± 0.3 V

When a low voltage condition is detected, the low voltage detection flag (LVRC : LVRF) is set to “1” and an internal reset signal is output.

Because the low voltage detection circuit continues to operate even in stop mode, detection of a low voltage condition generates an internal reset and releases stop mode.

During an internal RAM write cycle, an internal reset is generated after the completion of writing. During the output of this internal reset, the reset output from the low voltage detection circuit is suppressed.

(2) Program Looping detection reset circuit

The Program Looping detection reset circuit is a counter that prevents program looping. The counter starts automatically after a power-on reset, and must be continually cleared within a given time. If the given time interval elapses and the counter has not been cleared, a cause such as infinite program looping is assumed and an internal reset signal is generated. The internal reset generated form the Program Looping detection circuit has a width of 5 machine cycles.

Interval duration	Number of oscillation clock cycles
Approx. 262 ms *	2^{20} cycles

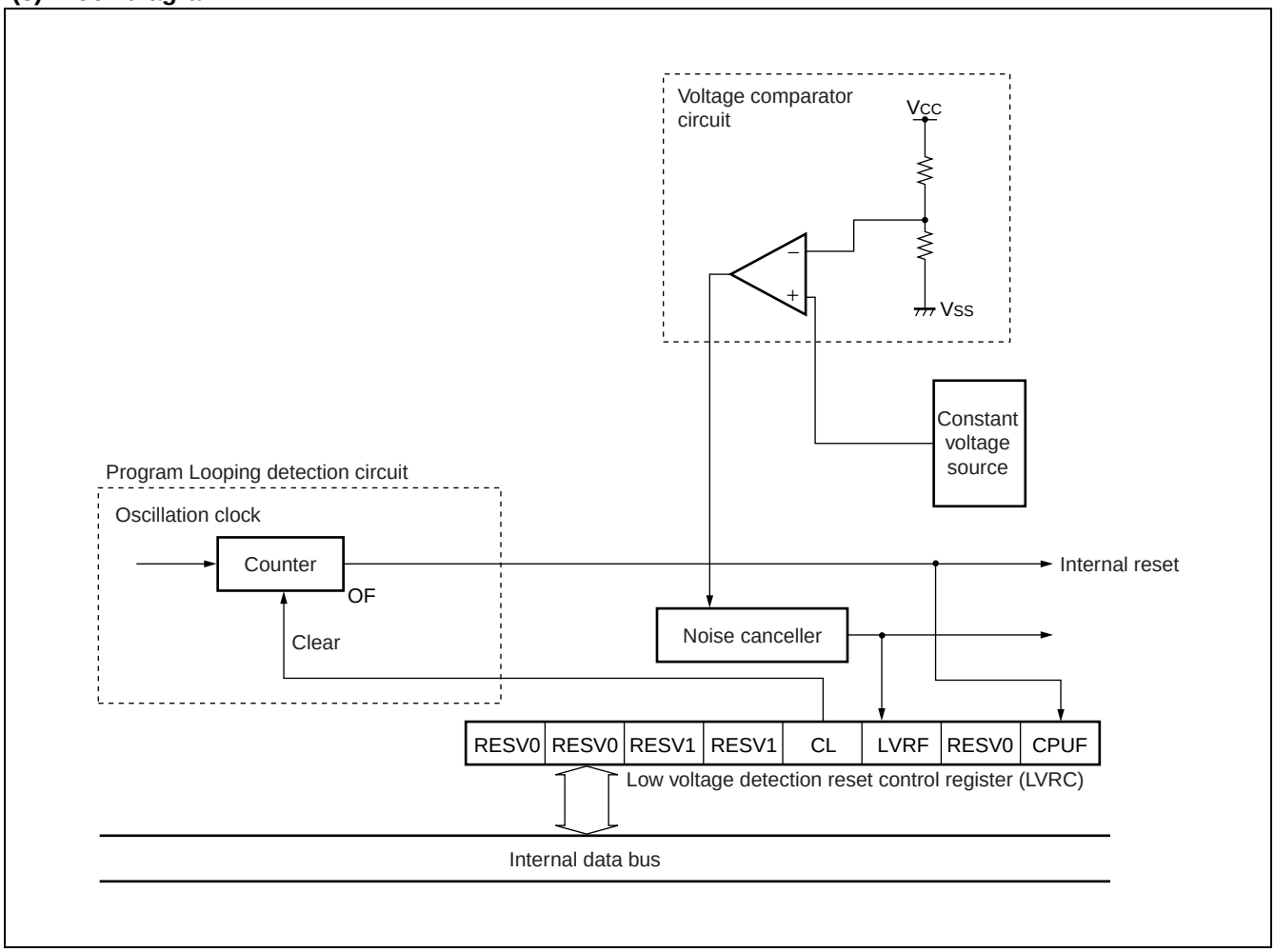
* : This value assumes an oscillation clock speed of 4 MHz.
During recovery from standby mode the detection period is the maximum interval plus 20 µs.

This circuit does not operate in modes where CPU operation is stopped.

The Program Looping detection reset circuit counter is cleared under any of the following conditions.

1. Writing “0” to the LVRC register CL bit
2. Internal reset
3. Main oscillation clock stop
4. Transition to sleep mode
5. Transition to time base timer mode or clock mode
6. Start of hold

(3) Block diagram

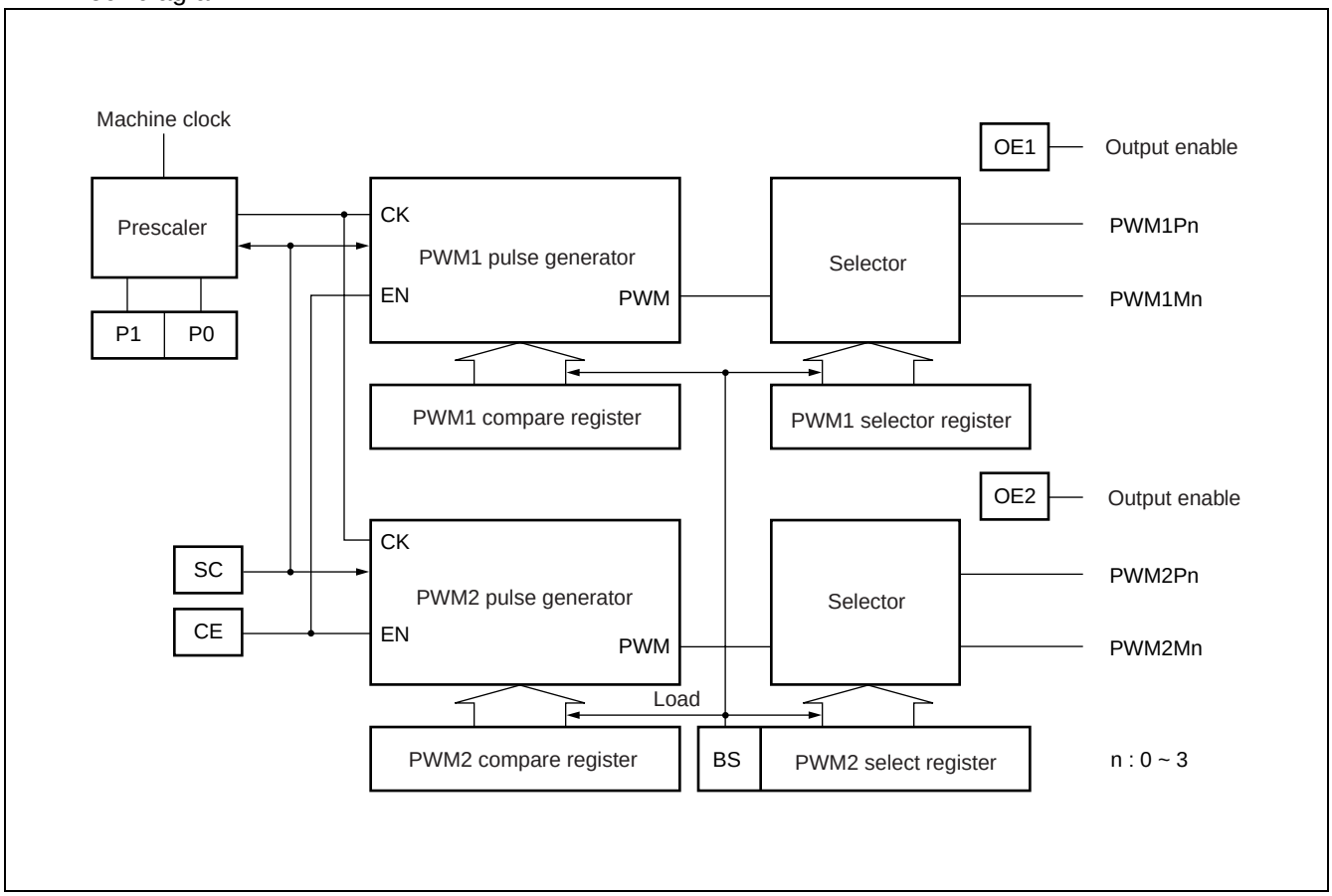


14. Stepping Motor Controller

The stepping motor controller is composed of two PWM pulse generators, four motor drivers and selector logic circuits.

The four motor drivers have a high output drive capacity and can be directly connected to the four ends of two motor coils. They are designed to operate together with the PWM pulse generators and selector logic circuits to control motor rotation. A synchronization mechanism assures synchronization of the two PWM pulse generators.

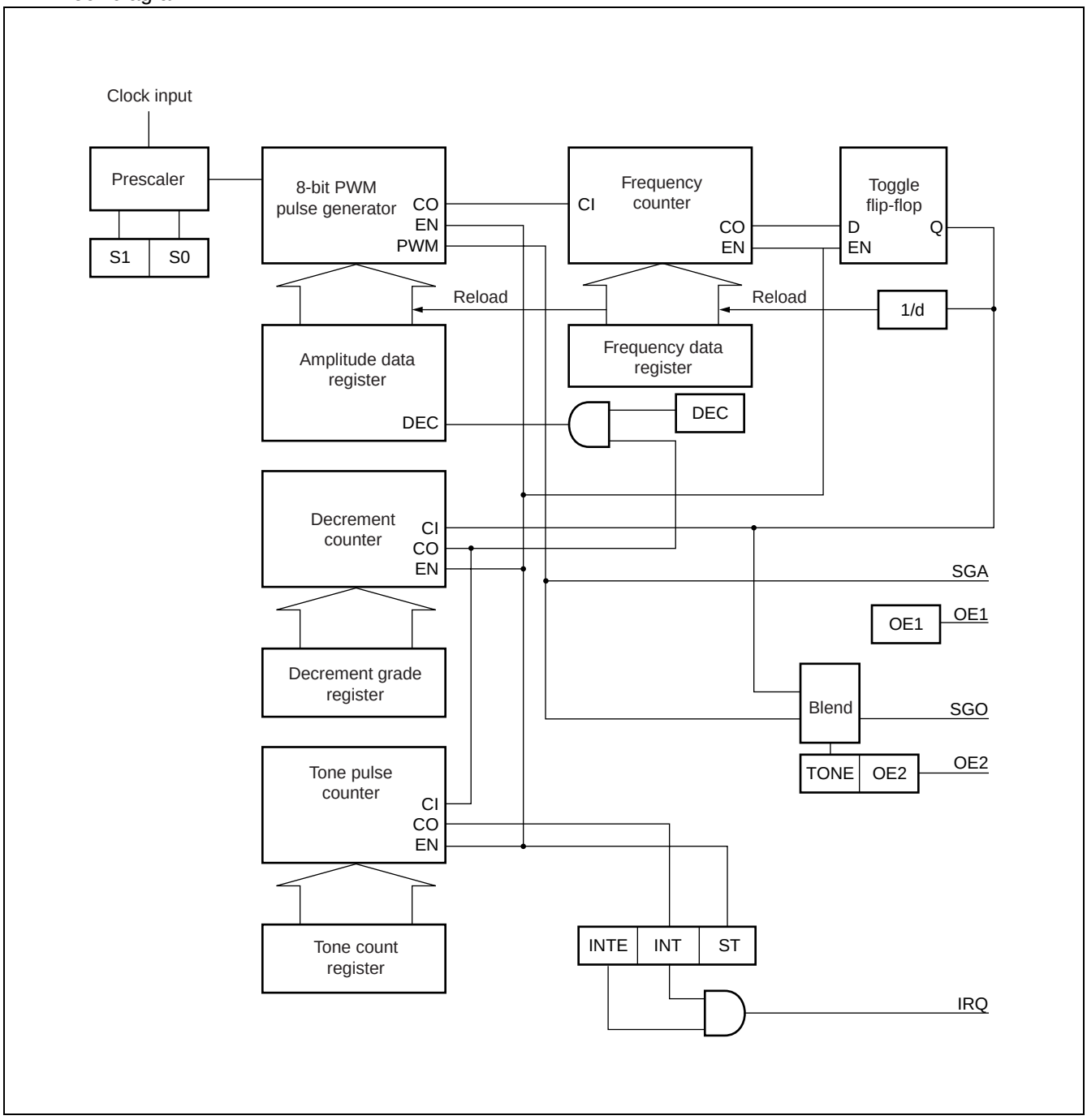
- Block diagram



15. Sound Generator

The sound generator is composed of a sound control register, frequency data register, amplitude data register, decrement grade register, tone count register, PWM pulse generator, frequency counter, decrement counter, and tone pulse counter.

- Block diagram

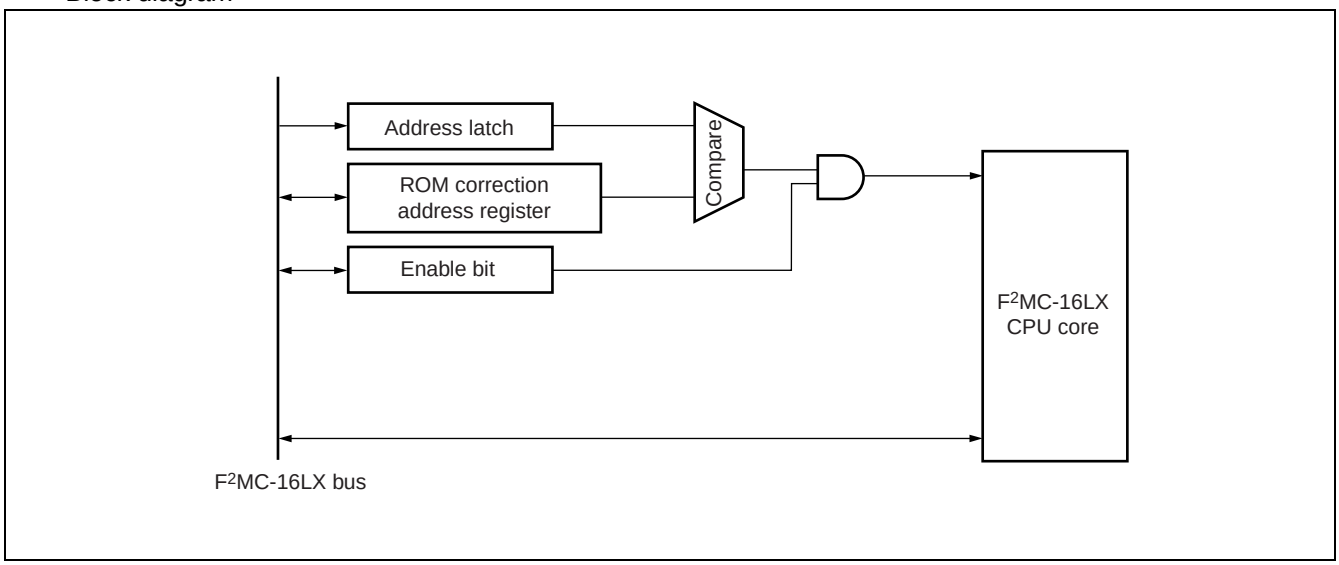


16. Address Match Detect Function

If the address setting is the same as the ROM correction address register, an INT9 instruction is executed. The ROM correction function can be implemented by processing the INT9 interrupt service routine.

Two address registers are used, each with its own compare enable bit. When there is a match between the address register and program counter, and the compare enable bit is set to “1” , the INT9 instruction is forcibly executed by the CPU.

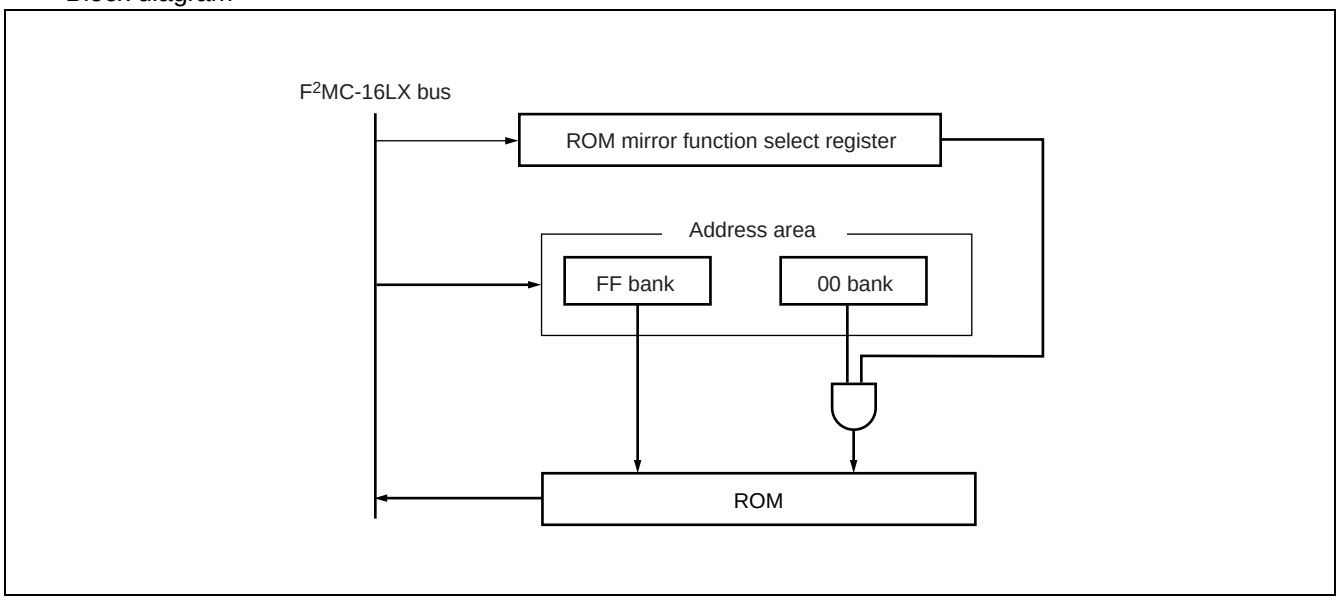
- Block diagram



17. ROM Mirror Function Select Module

The ROM mirror function select module uses a select register setting to enable the contents of ROM allocated to the FF bank to be viewed in the 00 bank.

- Block diagram



■ ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Ratings

(V_{SS} = AV_{SS} = DV_{SS} = 0 V)

Parameter	Symbol	Rating		Unit	Remarks
		Min.	Max.		
Power supply voltage	V_{CC}	V_{SS} – 0.3	V_{SS} + 6.0	V	
	AV_{CC}	V_{SS} – 0.3	V_{SS} + 6.0	V	AV_{CC} = V_{CC}*1
	V_{AVRH}	V_{SS} – 0.3	V_{SS} + 6.0	V	$AV_{CC} \geq V_{AVRH}$
	DV_{CC}	V_{SS} – 0.3	V_{SS} + 6.0	V	DV_{CC} = V_{CC}*1
Input voltage	V_I	V_{SS} – 0.3	V_{CC} + 0.3	V	
Output voltage	V_O	V_{SS} – 0.3	V_{CC} + 0.3	V	
Clamp current	I_{CLAMP}	–2.0	2.0	mA	
"L"level maximum output current*2	I_{OL1}	—	15	mA	Other than P70-P77, P80-P87
	I_{OL2}	—	40	mA	P70-77, P80-87
"L"level average output current*3	I_{OLAV1}	—	4	mA	Other than P70-P77, P80-P87
	I_{OLAV2}	—	30	mA	P70-77, P80-87
"L"level maximum total output current	ΣI_{OL1}	—	100	mA	Other than P70-P77, P80-P87
	ΣI_{OL2}	—	330	mA	P70-77, P80-87
"L"level average total output current	ΣI_{OLAV1}	—	50	mA	Other than P70-P77, P80-P87
	ΣI_{OLAV2}	—	250	mA	P70-77, P80-87
"H"level maximum output current	I_{OH1}*2	—	–15	mA	Other than P70-P77, P80-P87
	I_{OH2}*2	—	–40	mA	P70-77, P80-87
"H"level average output current	I_{OHAV1}*3	—	–4	mA	Other than P70-P77, P80-P87
	I_{OHAV2}*3	—	–30	mA	P70-77, P80-87
"H"level maximum total output current	ΣI_{OH1}	—	–100	mA	Other than P70-P77, P80-P87
	ΣI_{OH2}	—	–330	mA	P70-77, P80-87
"H"level average total output current	ΣI_{OHAV1}*4	—	–50	mA	Other than P70-P77, P80-P87
	ΣI_{OHAV2}*4	—	–250	mA	P70-77, P80-87
Power consumption	P_D	—	500	mW	
Operating temperature	T_A	–40	+105	°C	
Storage temperature	T_{STG}	–55	+150	°C	

*1 : Care must be taken to ensure that AV_{CC} and DV_{CC} do not exceed V_{CC} at power-on etc.

*2 : Maximum output current is defined as the peak value of the current of any one of the corresponding pins.

*3 : Average output current is defined as the value of the average current flowing over 100 ms at any one of the corresponding pins. The "average value" can be calculated from the formula of "operating current" times "operating factor".

*4 : Average total output current is defined as the value of the average current flowing over 100 ms at all of the corresponding pins. The "average value" can be calculated from the formula of "operating current" times " operating factor".

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.

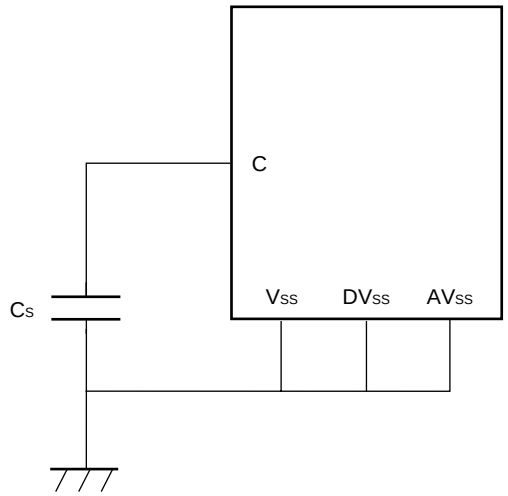
2. Recommended Operating Conditions

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

Parameter	Symbol	Value		Unit	Remarks
		Min.	Max.		
Power supply voltage	V_{CC} AV_{CC} DV_{CC}	4.5	5.5	V	In normal operation: (MB90F428G/F428GA, MB90428G/428GA, MB90427G/427GA)
		3.0	5.5	V	Holding stop operation status (MB90F428G, MB90428G, MB90427G)
		4.5	5.5	V	Holding stop operation status (MB90F428GA, MB90428GA, MB90427GA)
Smoothing capacitor*	C_S	0.1	1.0	µF	Use a ceramic capacitor or other capacitor of equivalent frequency characteristics. A smoothing capacitor on the V_{CC} pin should have a capacitance greater than Cs.
Operating temperature	T_A	–40	+105	°C	

* : For smoothing capacitor Cs connections, see the illustration below.

• C pin connection



WARNING: The recommended operating conditions are required in order to ensure the normal operation of the semiconductor device. All of the device's electrical characteristics are warranted when the device is operated within these ranges.

Always use semiconductor devices within their recommended operating condition ranges. 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 representatives beforehand.

3. DC Characteristics

(V_{CC} = 5.0 V±10%, V_{SS} = DV_{SS} = AV_{SS} = 0.0 V, T_A = –40 °C to +105 °C)

Parameter	Symbol	Pin name	Conditions	Value			Unit	Remarks
				Min.	Typ.	Max.		
"H"level input voltage	V_{IHS}	—	—	0.8 V_{CC}	—	V_{CC} + 0.3	V	CMOS hysteresis input pin*1
	V_{IHM}	—	—	V_{CC} – 0.3	—	V_{CC} + 0.3	V	MD pin*2
"L"level input voltage	V_{ILS}	—	—	V_{SS} – 0.3	—	0.6 V_{CC}	V	CMOS hysteresis input pin*1
	V_{ILM}	—	—	V_{SS} – 0.3	—	V_{SS} + 0.3	V	MD pin*2
Power supply current*3	I_{CC}	V_{CC}	Operating frequency F_{CP} = 16 MHz, normal operation	—	45	72	mA	MB90F428G/GA MB90F423G/GA
				—	38	61	mA	MB90428G/GA MB90427G/GA MB90423G/GA
	I_{CCS}		Operating frequency F_{CP} = 16 MHz, sleep mode	—	15	24	mA	MB90F428G/GA MB90F423G/GA
				—	13	21	mA	MB90428G/GA, MB90427G/GA MB90423G/GA
	I_{CTS}		Operating frequency F_{CP} = 2 MHz, time base timer mode	—	0.75	1.0	mA	
	I_{CCL}		Operating frequency F_{CP} = 8 kHz, T_A = 25 °C, subclock operation	—	0.35	0.7	mA	
	I_{CCLS}		Operating frequency F_{CP} = 8 kHz, T_A = 25 °C, sub sleep operation	—	40	100	µA	
	I_{CCT}		Operating frequency F_{CP} = 8 kHz, T_A = 25 °C, clock mode	—	40	100	µA	

*1 : All input pins except X0, X0A, MD0, MD1, MD2 pins.

*2 : MD0, MD1, MD2 pins.

*3 : Current values are provisional, and may be changed without prior notice for purposes of characteristic improve ment, etc. Supply current values assume external clock feed from the 1 pin and X1A pin. Users must be aware that supply current levels differ depending on whether an external clock or oscillator is useed.

(Continued)

(Continued)

(V_{CC} = 5.0 V±10%, V_{SS} = DV_{SS} = AV_{SS} = 0.0 V, T_A = –40 °C to +105 °C)

Parameter	Symbol	Pin name	Conditions	Value			Unit	Remarks
				Min.	Typ.	Max.		
Power supply current *3	I_{CCH}	V_{CC}	T_A = 25 °C, stop mode	—	5	20	µA	MB90F428G MB90F423G MB90428G MB90427G MB90423G
				—	40	100	µA	MB90F428GA MB90F423GA MB90428GA MB90427GA MB90423GA
Input leakage current	I_{IL}	All input pins	V_{CC} = DV_{CC} = AV_{CC} = 5.5 V V_{SS} < V_I < V_{CC}	–5	—	5	µA	
Input capacitance 1	C_{IN1}	Other than Vcc, Vss, DVcc, DVss, Avcc, Avss, C, P70 to P77, P80 to P87	—	—	5	15	pF	
Input capacitance 2	C_{IN2}	P70 to P77, P80 to P87	—	—	15	45	pF	
Pull-up resistance	R_{UP}	$\overline{RST}$, MD0, MD1	—	25	50	100	kΩ	
Pull-down resistance	R_{DOWN}	MD2	—	25	50	100	kΩ	
Output H voltage 1	V_{OH1}	Other than P70 to P77, P80 to P87	V_{CC} = 4.5 V I_{OH} = –4.0 mA	V_{CC} – 0.5	—	—	V	
Output H voltage 2	V_{OH2}	P70 to P77, P80 to P87	V_{CC} = 4.5 V I_{OH} = –30.0 mA	V_{CC} – 0.5	—	—	V	
Output L voltage 1	V_{OL1}	Other than P70 to P77, P80 to P87	V_{CC} = 4.5 V I_{OL} = 4.0 mA	—	—	0.4	V	
Output L voltage 2	V_{OL2}	P70 to P77, P80 to P87	V_{CC} = 4.5 V I_{OL} = 30.0 mA	—	—	0.5	V	

*3: Current values are provisional, and may be changed without prior notice for purposes of characteristic improve ment, etc. Supply current values assume external clock feed from the 1 pin and X1A pin. Users must be aware that supply current levels differ depending on whether an external clock or oscillator is useed.

(Continued)

(Continued)

Parameter	Symbol	Pin name	Conditions	Value			Unit	Remarks
				Min.	Typ.	Max.		
Large current output drive capacity variation 1	ΔV_{OH2}	PWM1Pn, PWM1Mn, PWM2Pn, PWM2Mn, n = 0 to 3	V_{CC} = 4.5 V I_{OH} = 30.0 mA V_{OH2} maximum variation	0	—	90	mV	*4
Large current output drive capacity variation 2	ΔV_{OL2}	PWM1Pn, PWM1Mn, PWM2Pn, PWM2Mn, n = 0 to 3	V_{CC} = 4.5 V I_{OH} = 30.0 mA V_{OL2} maximum variation	0	—	90	mV	*4
LCD divider resistance	R_{LCD}	V0 to V1, V1 to V2, V2 to V3	—	50	100	200	kΩ	
COM0 to COM3 output impedance	R_{VCOM}	COMn (n = 0 to 3)	—	—	—	2.5	kΩ	
SEG0 to SEG3 output impedance	R_{VSEG}	SEGn (n = 00 to 23)	—	—	—	15	kΩ	
LCD leakage current	I_{LCDC}	V0 to V3 COMm (m = 00 to 23) SEGn (n = 00 to 23)	—	–5.0	—	+5.0	kΩ	

*4 : Defined as maximum variation in V_{OH2}/V_{OL2} with all channel 0 PWM1P0/PWM1M0/PWM2P0/PWM2M0 simultaneously ON. Similarly for other channels.

4. AC Characteristics

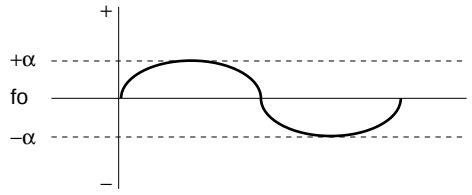
(1) Clock timing (**V_{CC}** = 5.0 V±10%, V_{SS} = DV_{SS} = AV_{SS} = 0.0 V, T_A = –40 °C to +105 °C)

Parameter	Symbol	Pin name	Condi-tions	Value			Unit	Remarks
				Min.	Typ.	Max.		
Base oscillation clock frequency	F_C	X0, X1		—	4	—	MHz	
	F_{LC}	X0A, X1A		—	32.768	—	kHz	
Base oscillation clock cycle time	t_{CYL}	X0, X1		—	250	—	ns	
	t_{LCYL}	X0A, X1A		—	30.5	—	µs	
Input clock pulse width	P_{WH}, P_{WL}	X0		10	—	—	ns	Use duty ratio of 40 to 60% as a guideline
	P_{WLH}, P_{WLL}	X0A		—	15.2	—	µs	
Input clock rise, fall time	tcr, tcf	X0, X0A	—	—	—	5	ns	With external clock signal
Input operating clock frequency	F_{CP}	—		2	—	16	MHz	Using main clock, PLL clock
	F_{LCP}	—		—	8.192	—	kHz	Using sub clock
Input operating clock cycle time	t_{CP}	—		62.5	—	500	ns	Using main clock, PLL clock
	t_{LCP}	—		—	122.1	—	µs	Using sub clock
Frequency variability ratio* (locked)	Δf	—		—	—	5	%	

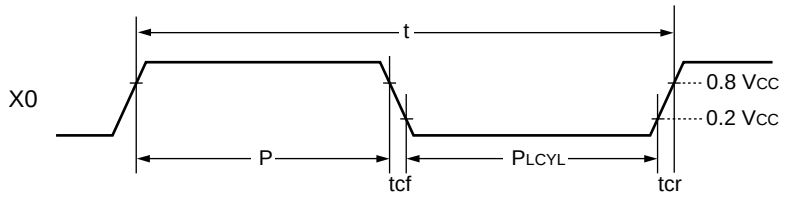
*: The frequency variability ratio is the maximum proportion of variation from the set central frequency using a multiplier in locked operation.

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

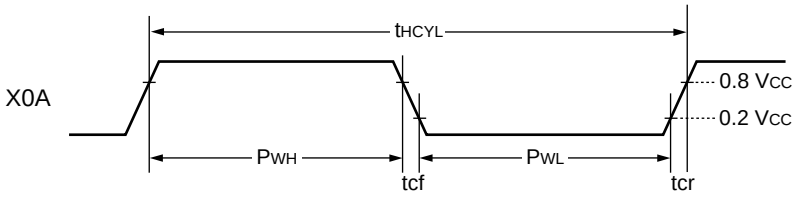
Central frequency



- X0, X1 clock timing

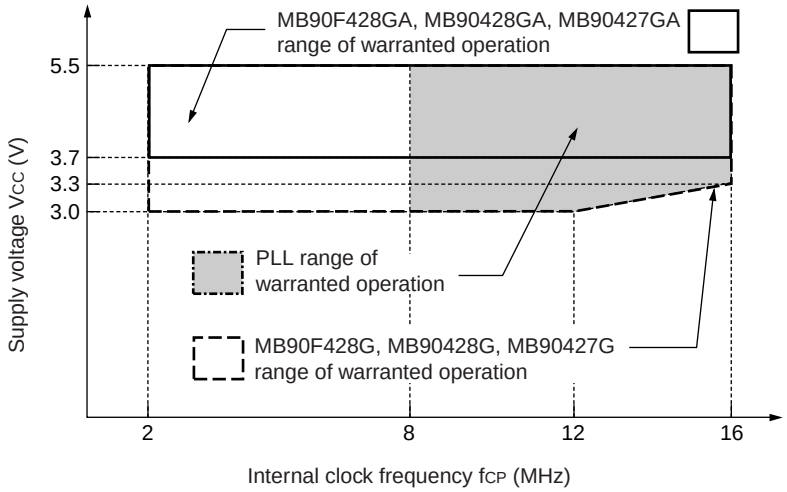


- X0A, X1A clock timing



- Range of warranted operation

Relation between internal operating clock frequency and supply voltage



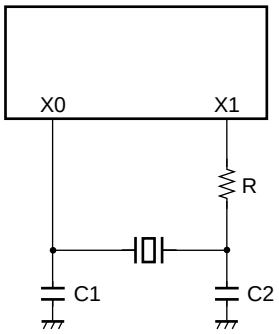
The MB90F428GA, MB90F423GA, MB90428GA, MB90427GA, and MB90423GA enter reset mode at supply voltage below 4 V ± 0.3 V.

Relation between oscillator clock frequency and internal operating clock frequency

		Internal operating clock frequency				
		Main clock	PLL clock			
			Multiplier × 1	Multiplier × 2	Multiplier × 3	Multiplier × 4
Oscillation clock frequency	4 MHz	2 MHz	—	8 MHz	12 MHz	16 MHz

- Sample oscillator circuit

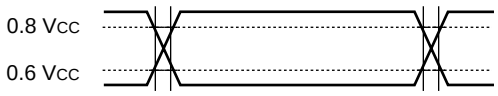
Oscillator element manufacturer	Oscillator	Frequency	C1	C2	R
TBD	TBD	4 MHz	TBD	TBD	TBD



AC ratings are defined for the following measurement reference voltage values:

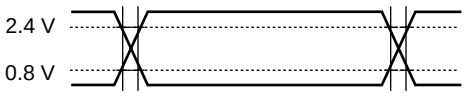
- Input signal waveform

Hysteresis input pin



- Output signal waveform

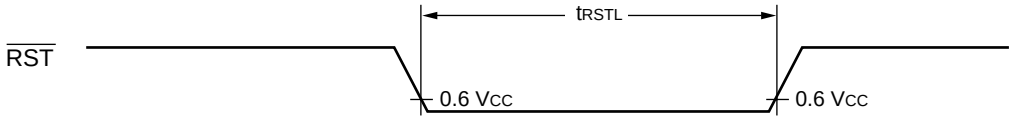
Output pin



(2) Reset input

(V_{CC} = 5.0 V±10%, V_{SS} = AV_{SS} = 0.0 V, T_A = –40 °C to +105 °C)

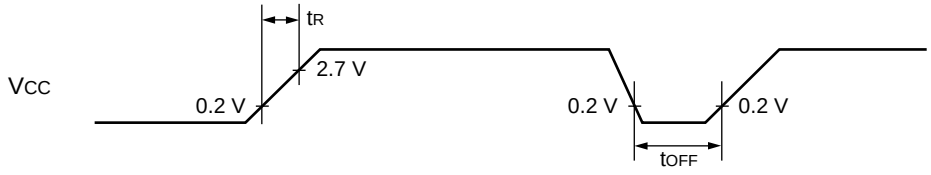
Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Reset input time	t_{RSTL}	RST	—	16 t_{CP}	—	ns	



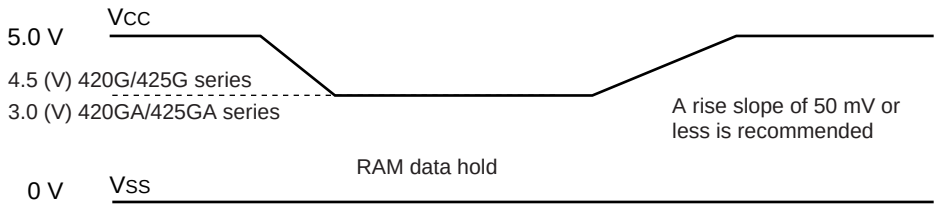
(3) Power-on reset, power on conditions

(V_{SS} = 0.0 V, T_A = –40 °C to +105 °C)

Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Power supply rise time	t_R	V_{CC}	—	0.05	30	ms	
Power supply start voltage	V_{OFF}			—	0.2	V	
Power supply attained voltage	V_{ON}			2.7	—	V	
Power supply cutoff time	t_{OFF}			50	—	ms	For repeat operation



Extreme variations in voltage supply may activate a power-on reset.
As the illustration below shows, when varying supply voltage during operation the use of a smooth voltage rise with suppressed fluctuation is recommended. Also in this situation, the PLL clock on the device should not be used, however it is permissible to use the PLL clock during a voltage drop of 1mV/s or less.



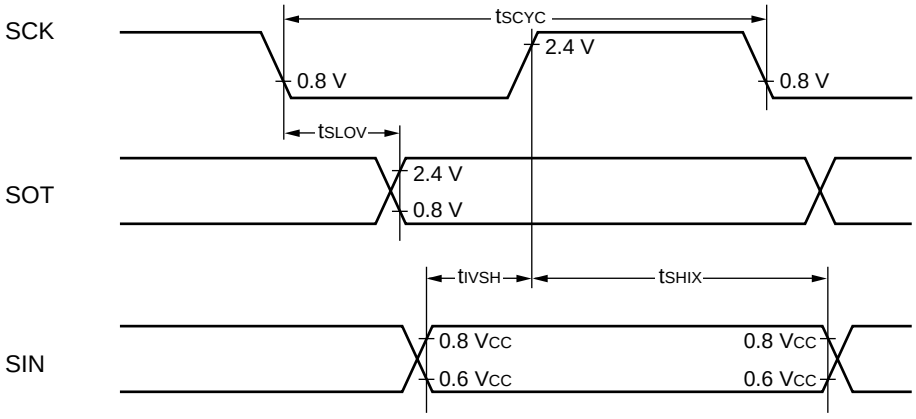
(4) UART0, UART1 timing

(V_{CC} = 5.0 V±10%, V_{SS} = AV_{SS} = 0.0 V, T_A = –40 °C to +105 °C)

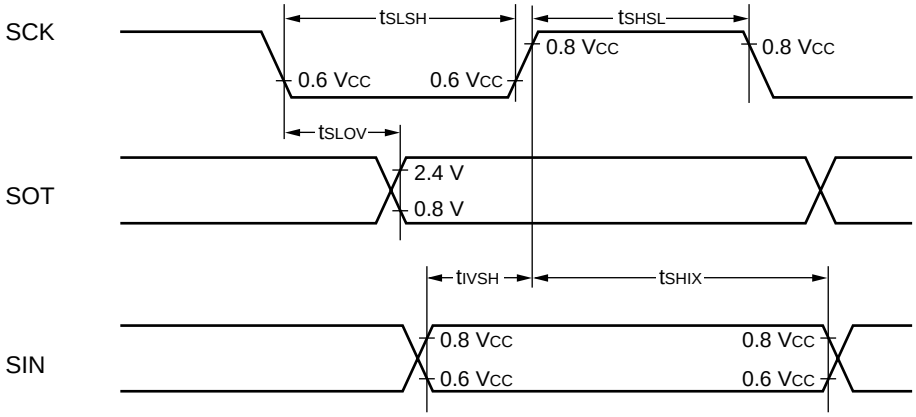
Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Serial clock cycle time	t_{SCYC}	SCK0, SCK1	—	8 t_{CP}	—	ns	Internal shift clock mode output pin C_L = 80 pF + 1•TTL
SCK fall to SOT delay time	t_{SLOV}	SCK0, SCK1 SOT0, SOT1		–80	80	ns	
Valid SIN to SCK rise	t_{IVSH}	SCK0, SCK1 SIN0, SIN1		100	—	ns	
SCK rise to valid SIN hold time	t_{SHIX}			60	—	ns	
Serial clock “H” pulse width	t_{SHSL}	SCK0, SCK1	—	4 t_{CP}	—	ns	External shift clock mode output pin C_L = 80 pF + 1•TTL
Serial clock “L” pulse width	t_{SLSH}			4 t_{CP}	—	ns	
SCK fall to SOT delay time	t_{SLOV}	SCK0, SCK1 SOT0, SOT1		—	150	ns	
Valid SIN to SCK rise	t_{IVSH}	SCK0, SCK1 SIN0, SIN1		60	—	ns	
SCK rise to valid SIN hold time	t_{SHIX}			60	—	ns	

Notes : • AC ratings are for CLK synchronous mode.
• C_L is load capacitance connected to pin during testing.

• Internal shift clock mode



• External shift clock mode

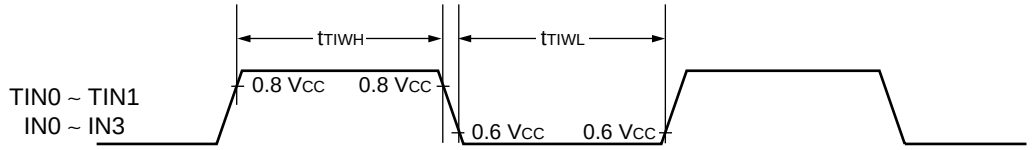


(5) Timer input timing

(V_{CC} = 5.0 V±10%, V_{SS} = AV_{SS} = 0.0 V, T_A = –40 °C to +105 °C)

Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Input pulse width	t_{TIWH} t_{TIWL}	TIN0, TIN1, IN0, IN1, IN2, IN3,	—	4 t_{CP}	—	ns	

- Timer input timing

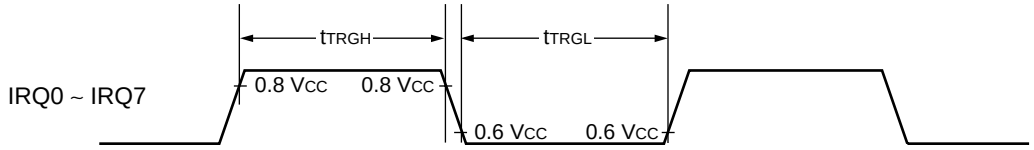


(6) Trigger input timing

(V_{CC} = 5.0 V±10%, V_{SS} = AV_{SS} = 0.0 V, T_A = –40 °C to +105 °C)

Parameter	Symbol	Pin name	Conditions	Value		Unit	Remarks
				Min.	Max.		
Input pulse width	t_{TRGL}	IRQ0 to IRQ7	—	5 t_{CP}	—	ns	

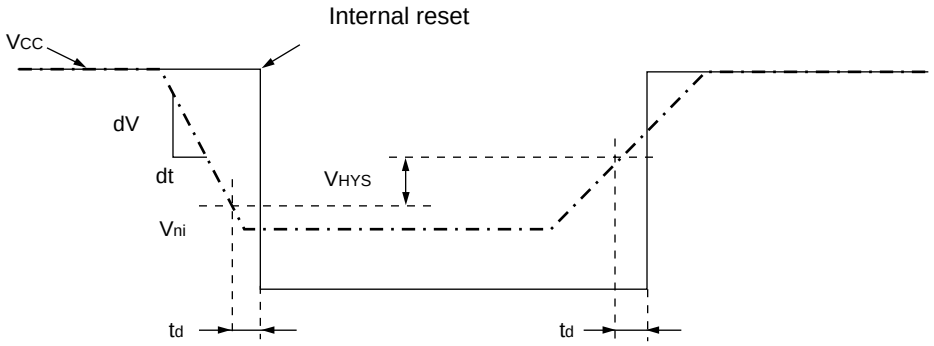
- Trigger input timing



(7) Low voltage detection

(V_{SS} = AV_{SS} = 0.0 V, T_A = –40 °C to +105 °C)

Parameter	Symbol	Pin name	Conditions	Value			Unit	Remarks
				Min.	Typ.	Max.		
Detection voltage	V_{DL}	V_{CC}	—	3.7	4.0	4.3	V	During voltage drop
Hysteresis width	V_{HYS}	V_{CC}		0.1	—	—	V	During voltage rise
Power supply voltage fluctuation ratio	dV/dt	V_{CC}		–0.1	—	0.02	V/µs	
Detection delay time	td	—		—	—	35	µs	



5. A/D Conversion Block

(1) Electrical Characteristics

(V_{CC} = AV_{CC} = 5.0 V±10%, V_{SS} = AV_{SS} = 0.0 V, T_A = –40 °C to +105 °C)

Parameter	Symbol	Pin name	Value			Unit	Remarks
			Min.	Typ.	Max.		
Resolution	—	—	—	—	10	bit	
Total error	—	—	—	—	±5.0	LSB	
Non-linear error	—	—	—	—	±2.5	LSB	
Differential linear error	—	—	—	—	±1.9	LSB	
Zero transition voltage	V_{OT}	AN0 to AN7	AV_{SS} – 3.5 LSB	AV_{SS} + 0.5 LSB	AV_{SS} + 4.5 LSB	V	1 LSB = (AVRH – AV_{SS}) / 1024
Full scale transition voltage	V_{FST}	AN0 to AN7	AVRH – 6.5 LSB	AVRH – 1.5 LSB	AVRH + 1.5 LSB	V	
Sampling time	t_{SMP}	—	2.000	—	—	µs	*1
Compare time	t_{CMP}	—	4.125	—	—	µs	*2
A/D conversion time	t_{CNV}	—	6.125	—	—	µs	*3
Analog port input current	I_{AIN}	AN0 to AN7	—	—	10	µA	V_{AVSS} = V_{AIN} = V_{AVCC}
Analog input current	V_{AIN}	AN0 to AN7	0	—	AVRH	V	
Reference voltage	AVR+	AVRH	3.0	—	AV_{CC}	V	
Power supply current	I_A	AV_{CC}	—	2.3	6.0	mA	
	I_{AH}		—	—	5	µA	*4
Reference voltage feed current	I_R	AVRH	200	400	600	µA	V_{AVRH} = 5.0 V
	I_{RH}	AVRH	—	—	5	µA	*4
Inter-channel variation	—	AN0 to AN7	—	—	4	LSB	

*1 : At F_{CP} = 16 MHz, t_{SMP} = 32 × t_{CP} = 2.000 (µs) .

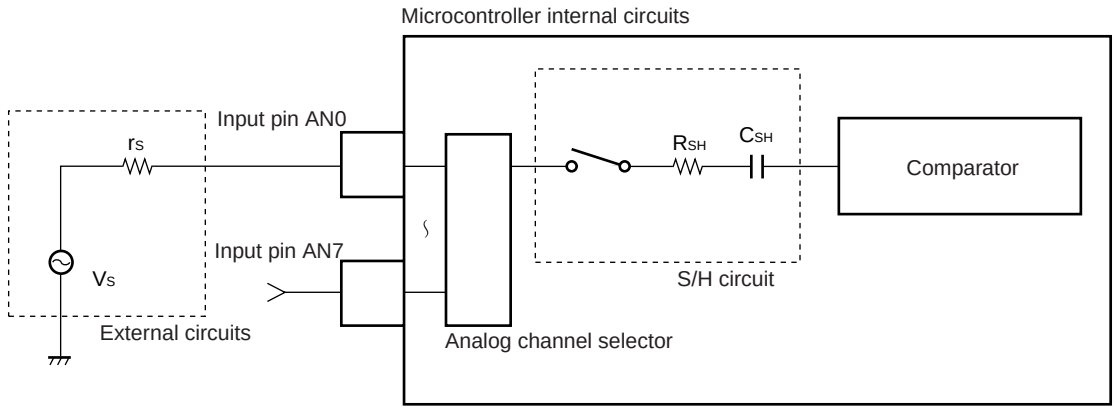
*2 : At F_{CP} = 16 MHz, t_{CMP} = 66 × t_{CP} = 4.125 (µs) .

*3 : Equivalent to conversion time per channel at F_{CP} = 16 MHz, and selection of t_{SMP} = 32 × t_{CP} and t_{CMP} = 32 × t_{CP}.

*4 : Defined as supply current (when V_{CC} = AV_{CC} = AVRH = 5.0 V) with A/D converter not operating, and CPU in stop mode.

Notes : •The relative error increases as AVRH is reduced.
- The output impedance (rs) on the external analog input circuit should be used as follows.
  External circuit output impedance rs = 5 kΩ max.
- If the output impedance on the external circuit is too great, the analog voltage sampling time may be insufficient.
- If DC inhibitor capacitance is placed between the external circuit and input pin, then a capacitance value several thousand times the value of the chip internal sampling capacitance (CSH) should be selected in order to suppress the effects of voltage division with CSH.

• Analog input equivalent circuit



<Recommended and guide values for element parameters>

rs = 5 kΩ or less

R_{SH} = approx. 3 kΩ

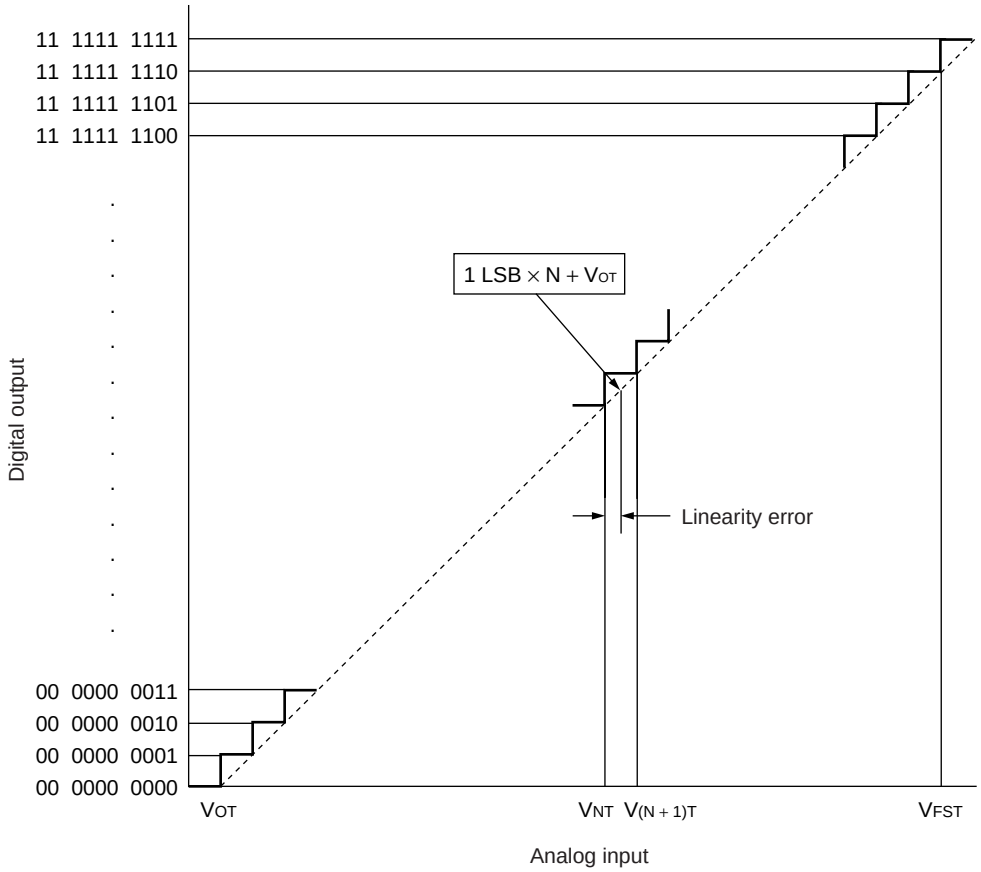
C_{SH} = approx. 25 pF

Note : These element parameters are intended as guidelines for reference, and are not warranted for actual use.

(2) Definition of terms

- Resolution
  Indicates the ability of the A/D converter to discriminate in analog conversion.
  10-bit resolution indicates that analog voltage can be resolved into $2^{10} = 1024$ levels.
- Total error
  Expresses the difference between actual and logical values. It is the total value of errors that can come from offset error, gain error, non-linearity error and noise.
- Linearity error
  Expresses the deviation between actual conversion characteristics and a straight line connecting the device's zero transition point (00 0000 0000 ←→ 00 0000 0001) and full scale transition point (11 1111 1110 ←→ 11 1111 1111) .
- Differential linearity error
  Expresses the deviation of the logical value of input voltage required to create a variation of 1 SLB in output code.

- 10-bit A/D converter conversion characteristics



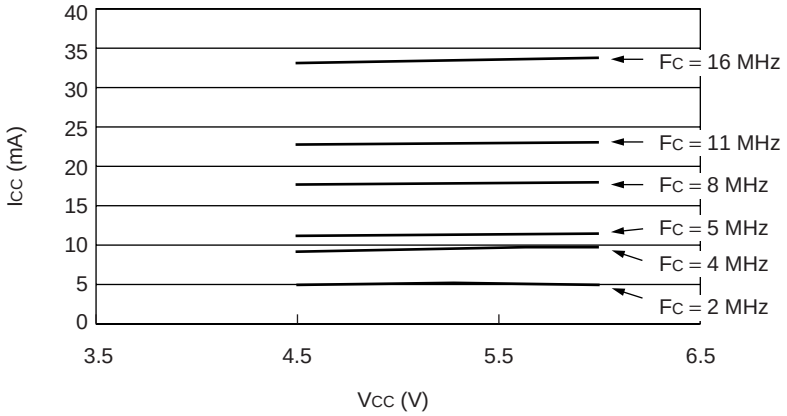
$$1\ \text{LSB} = \frac{V_{FST} - V_{OT}}{1022}$$

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

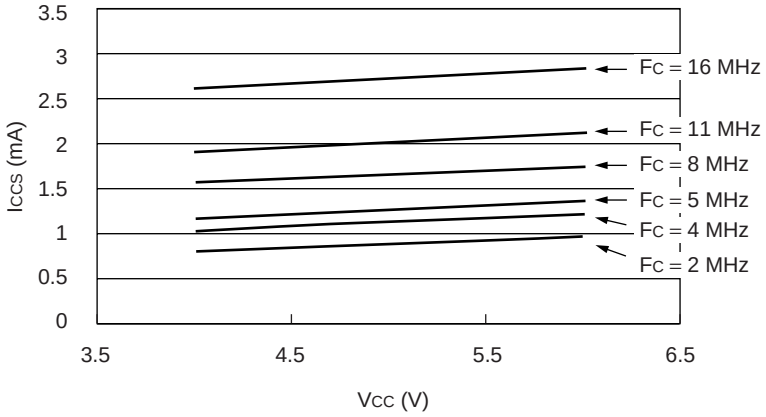
$$\text{Differential linearity error} = \frac{V_{(N+1)T} - V_{NT}}{1\ \text{LSB}} - 1\ [\text{LSB}]$$

■ EXAMPLE CHARACTERISTICS

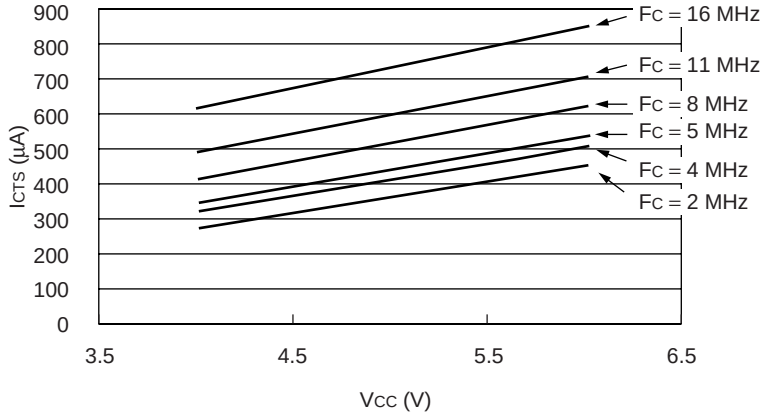
$I_{CC} - V_{CC}$ (T_A = +25 °C)



$I_{CCS} - V_{CC}$ (T_A = +25 °C)



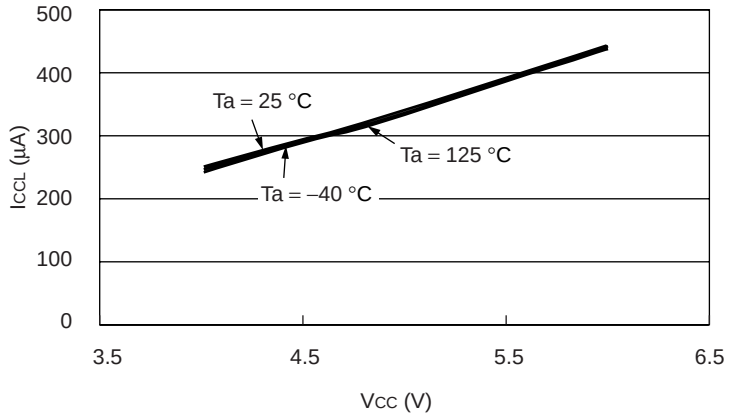
$I_{CTS} - V_{CC}$ (T_A = +25 °C)



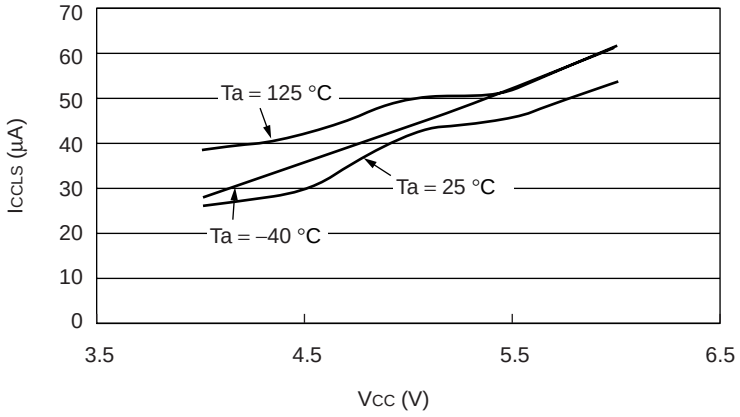
(Continued)

(Continued)

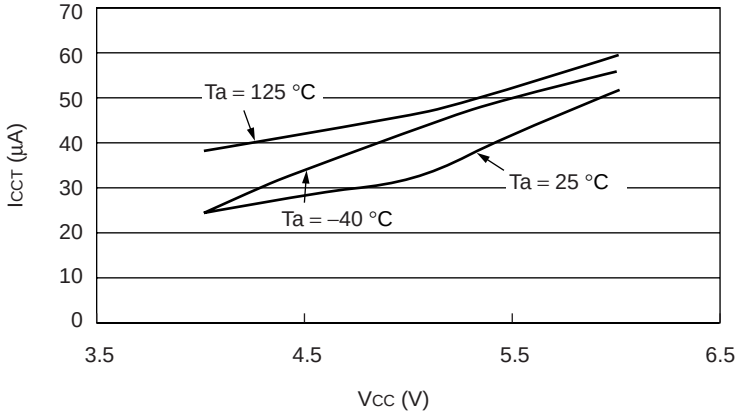
I_{CCL} – V_{CC} (F_C = 8 kHz)



I_{CCLS} – V_{CC} (F_C = 8 kHz)



I_{CCT} – V_{CC} (F_C = 8 kHz)



■ INSTRUCTIONS (351 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 code.
#	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 S T N Z V C	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.
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.

- **Number of execution cycles**

  The number of cycles required for instruction execution is acquired by adding the number of cycles for each instruction, a corrective value depending on the condition, and the number of cycles required for program fetch. Whenever the instruction being executed exceeds the two-byte (word) boundary, a program on an internal ROM connected to a 16-bit bus is fetched. If data access is interfered with, therefore, the number of execution cycles is increased.

  For each byte of the instruction being executed, a program on a memory connected to an 8-bit external data bus is fetched. If data access in interfered with, therefore, the number of execution cycles is increased. When a general-purpose register, an internal ROM, an internal RAM, an internal I/O device, or an external bus is accessed during intermittent CPU operation, the CPU clock is suspended by the number of cycles specified by the CG1/0 bit of the low-power consumption mode control register. When determining the number of cycles required for instruction execution during intermittent CPU operation, therefore, add the value of the number of times access is done × the number of cycles suspended as the corrective value to the number of ordinary execution cycles.

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 and 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
rel	PC relative addressing
ear eam	Effective addressing (codes 00 to 07) Effective addressing (codes 08 to 1F)
rlst	Register list

Table 3 Effective Address Fields

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

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

Table 4 Number of Execution Cycles for Each Type of Addressing

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

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

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

Operand	(b) byte		(c) word		(d) long	
	Cycles	Access	Cycles	Access	Cycles	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 Signed Multiplication and Division Instructions (Byte/Word/Long Word) [11 Instructions]

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
DIV	A	2	*1	0	0	word (AH) /byte (AL) Quotient → byte (AL) Remainder → byte (AH)	Z	–	–	–	–	–	–	*	*	–
DIV	A, ear	2	*2	1	0	word (A)/byte (ear) Quotient → byte (A) Remainder → byte (ear)	Z	–	–	–	–	–	–	*	*	–
DIV	A, eam	2 +	*3	0	*6	word (A)/byte (eam) Quotient → byte (A) Remainder → byte (eam)	Z	–	–	–	–	–	–	*	*	–
DIVW	A, ear	2	*4	1	0	long (A)/word (ear) Quotient → word (A) Remainder → word (ear)	–	–	–	–	–	–	–	*	*	–
DIVW	A, eam	2+	*5	0	*7	long (A)/word (eam) Quotient → word (A) Remainder → word (eam)	–	–	–	–	–	–	–	*	*	–
MULU	A	2	*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	2	*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: Set to 3 when the division-by-0, 8 or 18 for an overflow, and 18 for normal operation.
*2: Set to 3 when the division-by-0, 10 or 21 for an overflow, and 22 for normal operation.
*3: Set to 4 + (a) when the division-by-0, 11 + (a) or 22 + (a) for an overflow, and 23 + (a) for normal operation.
*4: Positive dividend: Set to 4 when the division-by-0, 10 or 29 for an overflow, and 30 for normal operation.
Negative dividend: Set to 4 when the division-by-0, 11 or 30 for an overflow and 31 for normal operation.
*5: Positive dividend: Set to 4 + (a) when the division-by-0, 11 + (a) or 30 + (a) for an overflow, and 31 + (a) for normal operation.
Negative dividend: Set to 4 + (a) when the division-by-0, 12 + (a) or 31 + (a) for an overflow, and 32 + (a) for normal operation.
*6: When the division-by-0, (b) for an overflow, and 2 × (b) for normal operation.
*7: When the division-by-0, (c) for an overflow, and 2 × (c) for normal operation.
*8: Set to 3 when byte (AH) is zero, 12 when the result is positive, and 13 when the result is negative.
*9: Set to 3 when byte (ear) is zero, 12 when the result is positive, and 13 when the result is negative.
*10: Set to 4 + (a) when byte (eam) is zero, 13 + (a) when the result is positive, and 14 + (a) when the result is negative.
*11: Set to 3 when word (AH) is zero, 12 when the result is positive, and 13 when the result is negative.
*12: Set to 3 when word (ear) is zero, 16 when the result is positive, and 19 when the result is negative.
*13: Set to 4 + (a) when word (eam) is zero, 17 + (a) when the result is positive, and 20 + (a) when the result is negative.

Notes:
- When overflow occurs during DIV or DIVW instruction execution, the number of execution cycles takes two values because of detection before and after an operation.
- When overflow occurs during DIV or DIVW instruction execution, the contents of AL are destroyed.
- For (a) to (d), refer to “Table 4 Number of Execution Cycles for Effective Address in Addressing Modes” and “Table 5 Correction Values for Number of Cycles for Calculating Actual Number of Cycles.”

Table 14 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 15 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	–	–

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 16 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)	–	–	–	–	–	*	*	*	*	*

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 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 18 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,	–	–	–	–	*	*	*	–	*	–
LSRW A, R0	2	*1	1	0	R0)	–	–	–	–	*	*	*	–	*	–
LSLW A, R0	2	*1	1	0	word (A) ← Logical right barrel shift (A, R0) 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 19 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 20 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*10	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*10	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	*7	Return from interrupt	–	–	*	*	*	*	*	*	*	–
LINK #imm8	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 *8	1	4	0	(c)	Return from subroutine	–	–	–	–	–	–	–	–	–	–
RETP *9	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: Set to 3 × (b) + 2 × (c) when an interrupt request occurs, and 6 × (c) for return.
*8: Retrieve (word) from stack
*9: Retrieve (long word) from stack
*10: 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 21 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 +3 × (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 22 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 23 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 A,T	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	*	–	–	–

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 24 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)

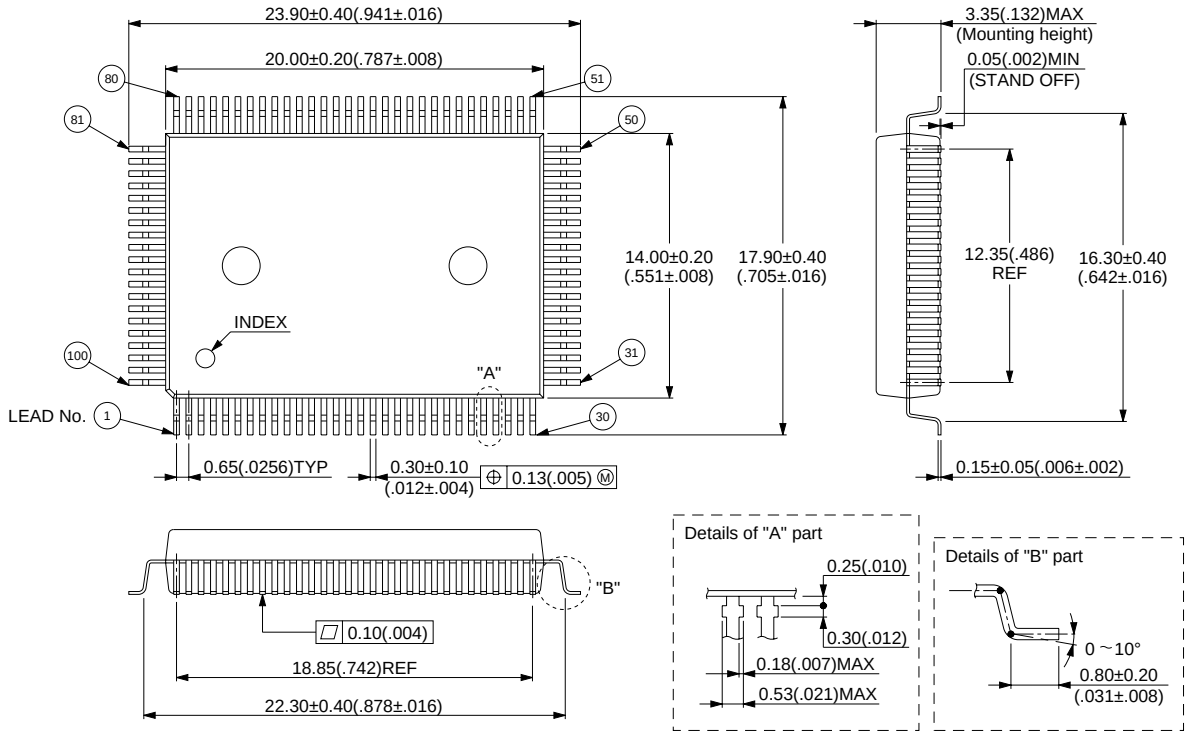
Note : For an explanation of “(a)” to “(d)”, refer to Table 4, “Number of Execution Cycles for Each Type of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

■ ORDERING INFORMATION

Part number	Package	Remarks
MB90F428GAPF MB90F423GAPF MB90428GAPF MB90427GAPF MB90423GAPF MB90F428GPF MB90F423GPF MB90428GPF MB90427GPF MB90423GPF	Plastic QFP, 100-pin (FPT-100P-M06)	
MB90F428GAPFV MB90F423GAPFV MB90428GAPFV MB90427GAPFV MB90423GAPFV MB90F428GPFV MB90F423GPFV MB90428GPFV MB90427GPFV MB90423GPFV	Plastic LQFP, 100-pin (FPT-100P-M05)	

■ PACKAGE DIMENSIONS

Plastic QFP, 100-pin
(FPT-100P-M06)

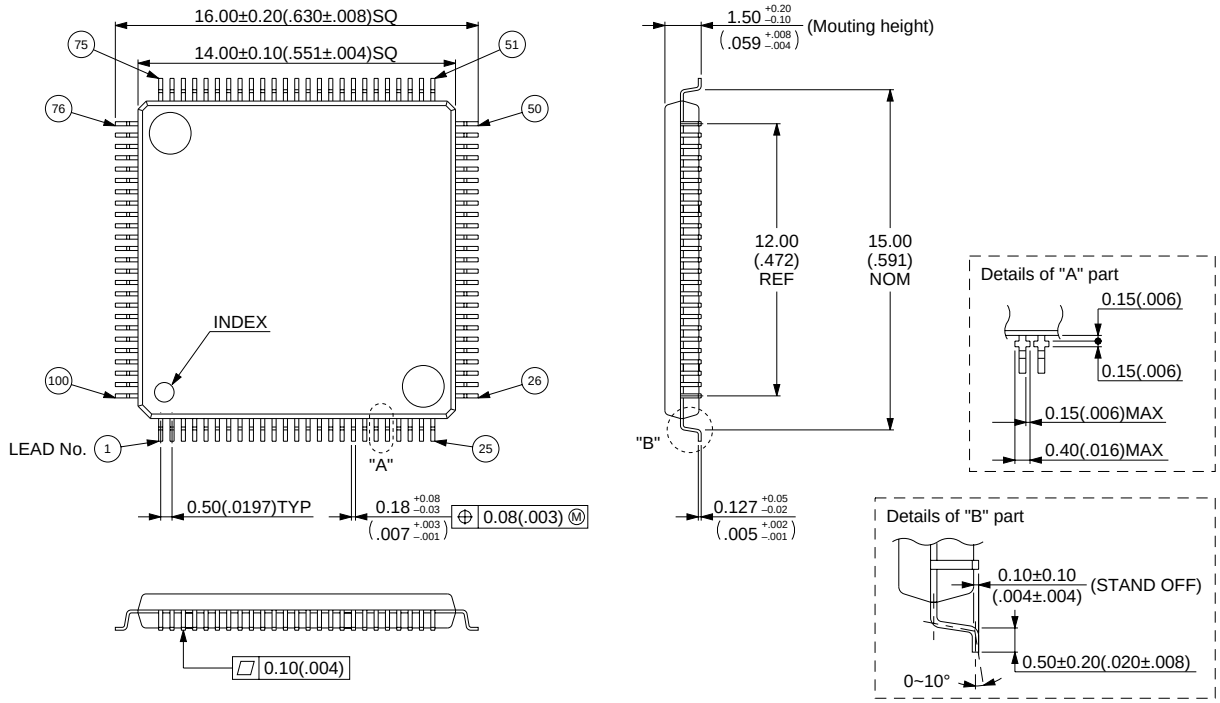


Dimensions in mm (inches)

(Continued)

(Continued)

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



Dimensions in mm (inches)

FUJITSU LIMITED

For further information please contact:

Japan

FUJITSU LIMITED
Corporate Global Business Support Division
Electronic Devices
Shinjuku Dai-Ichi Seimei Bldg. 7-1,
Nishishinjuku 2-chome, Shinjuku-ku,
Tokyo 163-0721, Japan
Tel: +81-3-5322-3347
Fax: +81-3-5322-3386

http://edevice.fujitsu.com/

North and South America

FUJITSU MICROELECTRONICS, INC.
3545 North First Street,
San Jose, CA 95134-1804, U.S.A.
Tel: +1-408-922-9000
Fax: +1-408-922-9179

Customer Response Center
Mon. - Fri.: 7 am - 5 pm (PST)
Tel: +1-800-866-8608
Fax: +1-408-922-9179

http://www.fujitsumicro.com/

Europe

FUJITSU MICROELECTRONICS EUROPE GmbH
Am Siebenstein 6-10,
D-63303 Dreieich-Buchschlag,
Germany
Tel: +49-6103-690-0
Fax: +49-6103-690-122

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

Asia Pacific

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

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

Korea

FUJITSU MICROELECTRONICS KOREA LTD.
1702 KOSMO TOWER, 1002 Daechi-Dong,
Kangnam-Gu,Seoul 135-280
Korea
Tel: +82-2-3484-7100
Fax: +82-2-3484-7111

F0012

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

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

The contents of this document may not be reproduced or copied without the permission of FUJITSU LIMITED.

FUJITSU semiconductor devices are intended for use in standard applications (computers, office automation and other office equipments, industrial, communications, and measurement equipments, personal or household devices, etc.).
CAUTION:
Customers considering the use of our products in special applications where failure or abnormal operation may directly affect human lives or cause physical injury or property damage, or where extremely high levels of reliability are demanded (such as aerospace systems, atomic energy controls, sea floor repeaters, vehicle operating controls, medical devices for life support, etc.) are requested to consult with FUJITSU sales representatives before such use. The company will not be responsible for damages arising from such use without prior approval.

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.