Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

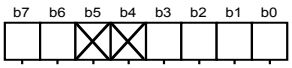
Details	
Product Status	Obsolete
Core Processor	M16C/60
Core Size	16-Bit
Speed	20MHz
Connectivity	I²C, IEBus, SIO, UART/USART
Peripherals	DMA, POR, PWM, Voltage Detect, WDT
Number of I/O	71
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	80-LQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/m30280fathp-u3a

Table 1.11 Pin Description (64-Pin and 80-Pin Packages) (Continued)

Classification	Symbol	I/O Type	Function
Timer S	$INPC1_0$ to $INPC1_7$	I	Input pins for the time measurement function
	$OUTC1_0$ to $OUTC1_7$	O	Output pins for the waveform generating function
I/O Ports	$P0_0$ to $P0_3$ $P1_5$ to $P1_7$ $P2_0$ to $P2_7$ $P3_0$ to $P3_3$ $P6_0$ to $P6_7$ $P7_0$ to $P7_7$ $P8_0$ to $P8_7$ $P9_0$ to $P9_3$ $P10_0$ to $P10_7$	I/O	CMOS I/O ports which have a direction register determines an individual pin is used as an input port or an output port. A pull-up resistor is selectable for every 4 input ports.

I : Input O : Output I/O : Input and output

DMA1 Request Cause Select Register



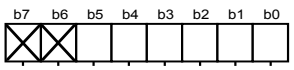
Symbol	Address	After Reset
DM1SL	03BA16	0016

Bit Symbol	Bit Name	Function	RW
DSEL0	DMA request cause select bit	Refer to note (1)	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
—— (b5-b4)	Nothing is assigned. If necessary, set to 0. When read, their contents are 0		——
DMS	DMA request cause expansion select bit	0: Basic cause of request 1: Extended cause of request	RW
DSR	Software DMA request bit	A DMA request is generated by setting this bit to 1 when the DMS bit is 0 (basic cause) and the DSEL3 to DSEL0 bits are 0001 2 (software trigger). The value of this bit when read is 0	RW

NOTES:
1. The causes of DMA1 requests can be selected by a combination of DMS bit and bits DSEL3 to DSEL0 in the manner described below.

DSEL3 to DSEL0	DMS=0(basic cause of request)	DMS=1(extended cause of request)
0 0 0 0 2	Falling edge of INT1 pin	IC/OC base timer
0 0 0 1 2	Software trigger	–
0 0 1 0 2	Timer A0	IC/OC channel 0
0 0 1 1 2	Timer A1	IC/OC channel 1
0 1 0 0 2	Timer A2	–
0 1 0 1 2	Timer A3	SI/O3
0 1 1 0 2	Timer A4	SI/O4
0 1 1 1 2	Timer B0	Two edges of INT1
1 0 0 0 2	Timer B1	–
1 0 0 1 2	Timer B2	–
1 0 1 0 2	UART0 transmit	IC/OC channel 2
1 0 1 1 2	UART0 receive	IC/OC channel 3
1 1 0 0 2	UART2 transmit	IC/OC channel 4
1 1 0 1 2	UART2 receive/ACK2	IC/OC channel 5
1 1 1 0 2	A/D conversion	IC/OC channel 6
1 1 1 1 2	UART1 receive	IC/OC channel 7

DMAi Control Register(i=0,1)



Symbol	Address	After Reset
DM0CON	002C16	00000X002
DM1CON	003C16	00000X002

Bit Symbol	Bit Name	Function	RW
DMBIT	Transfer unit bit select bit	0: 16 bits 1: 8 bits	RW
DMASL	Repeat transfer mode select bit	0: Single transfer 1: Repeat transfer	RW
DMAS	DMA request bit	0: DMA not requested 1: DMA requested	RW (1)
DMAE	DMA enable bit	0: Disabled 1: Enabled	RW
DSD	Source address direction select bit (2)	0: Fixed 1: Forward	RW
DAD	Destination address direction select bit (2)	0: Fixed 1: Forward	RW
—— (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, their contents are 0		——

NOTES:
1. The DMAS bit can be set to 0 by writing 0 by program (This bit remains unchanged even if 1 is written).
2. At least one of bits DAD and DSD must be set to 0 (address direction fixed).

Figure 11.3 DM1SL Register, DM0CON Register, and DM1CON Registers

Timer Ai Mode Register (i=2 to 4)
(When using two-phase pulse signal processing)

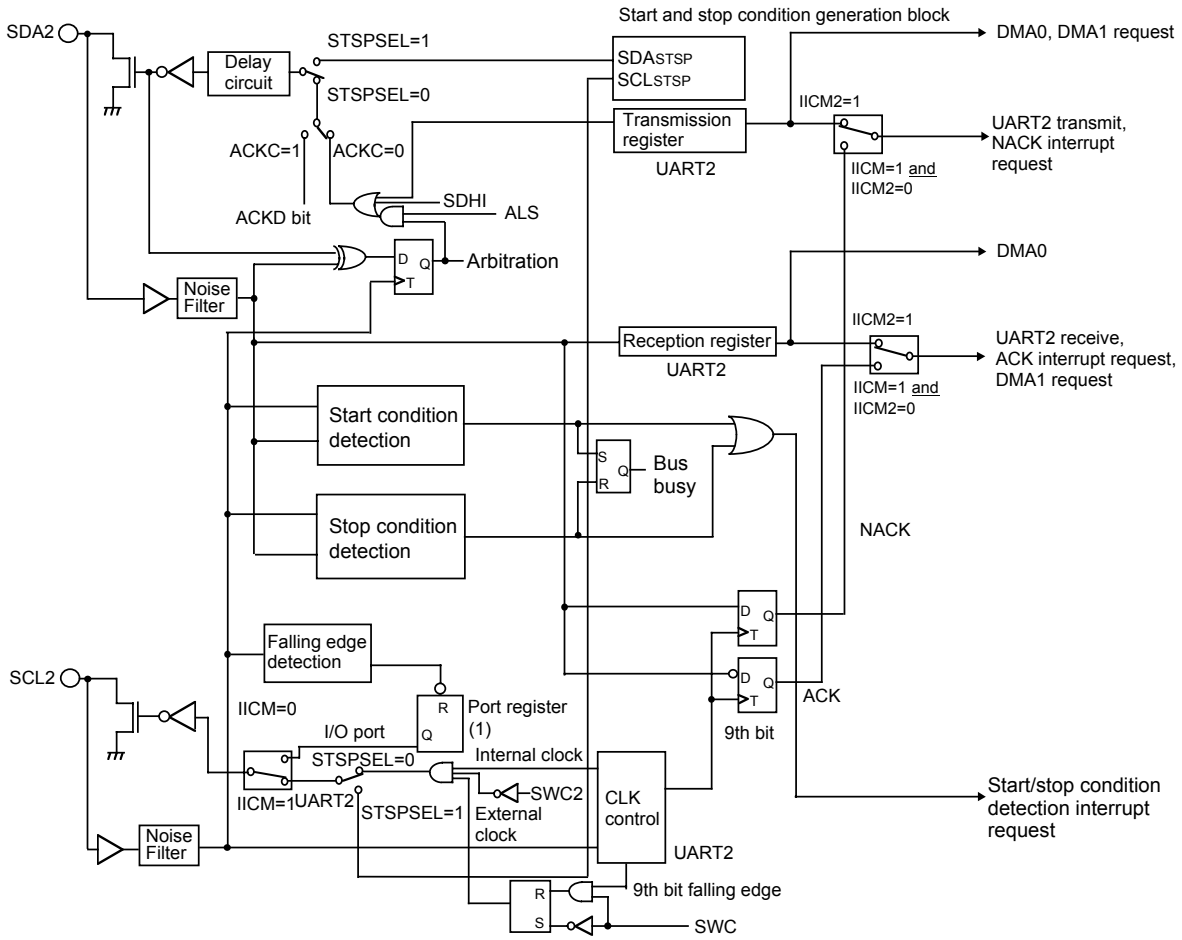
b6	b5	b4	b3	b2	b1	b0
		0	1	0	0	0	1

Symbol	Address	After Reset
TA2MR to TA4MR	0398_{16} to $039A_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1: Event counter mode	RW
TMOD1			RW
MR0	To use two-phase pulse signal processing, set this bit to 0		RW
MR1	To use two-phase pulse signal processing, set this bit to 0		RW
MR2	To use two-phase pulse signal processing, set this bit to 1		RW
MR3	To use two-phase pulse signal processing, set this bit to 0		RW
TCK0	Count operation type select bit	0: Reload type 1: Free-run type	RW
TCK1	Two-phase pulse signal processing operation select bit (1)(2)	0: Normal processing operation 1: Multiply-by-4 processing operation	RW

NOTES:
1. The TCK1 bit is valid for timer A3 mode register. No matter how this bit is set, timers A2 and A4 always operate in normal processing mode and x4 processing mode, respectively.
2. If two-phase pulse signal processing is desired, following register settings are required:
   - Set the TAiP bit in the UDF register to 1 (two-phase pulse signal processing function enabled).
   - Set bits TAiTGH and TAiTGL in the TRGSR register to 00_2 (TAiIN pin input).
   - Set the port direction bits for TAiIN and TAiOUT to 0 (input mode).

Figure 12.9 TA2MR to TA4MR Registers in Event Counter Mode (when using two-phase pulse signal processing with timer A2, A3 or A4)



This diagram applies to the case where bits SMD2 to SMD0 in the U2MR register is set to 010$_2$ and the IICM bit in the U2SMR register is set to 1.

IICM: Bit in the U2SMR register
IICM2, SWC, ALS, SWC2, SDHI: Bits in the U2SMR2 register
STSPSEL, ACKD, ACKC: Bits in the U2SMR4 register

NOTE:
1. If the IICM bit is set to 1, the pin can be read even when the PD7_1 bit is set to 1 (output mode).

Figure 14.22 I^{2}C bus Mode Block Diagram

14.1.3.4 Transfer Clock

Data is transmitted/received using a transfer clock like the one shown in **Figure 14.25**.

The CSC bit in the U2SMR2 register is used to synchronize the internally generated clock (internal SCL2) and an external clock supplied to the SCL_2 pin. In cases when the CSC bit is set to 1 (clock synchronization enabled), if a falling edge on the SCL_2 pin is detected while the internal SCL_2 is high, the internal SCL_2 goes low, at which time the U2BRG register value is reloaded with and starts counting in the low-level interval. If the internal SCL_2 changes state from low to high while the SCL_2 pin is low, counting stops, and when the SCL_2 pin goes high, counting restarts.

In this way, the UART2 transfer clock is comprised of the logical product of the internal SCL_2 and SCL_2 pin signal. The transfer clock works from a half period before the falling edge of the internal SCL_2 1st bit to the rising edge of the 9^{th} bit. To use this function, select an internal clock for the transfer clock.

The SWC bit in the U2SMR2 register allows to select whether the SCL_2 pin should be fixed to or freed from low-level output at the falling edge of the 9th clock pulse.

If the SCLHI bit in the U2SMR4 register is set to 1 (enabled), SCL_2 output is turned off (placed in the high-impedance state) when a stop condition is detected.

Setting the SWC2 bit in the U2SMR2 register is set to 1 (0 output) makes it possible to forcibly output a low-level signal from the SCL_2 pin even while sending or receiving data. Clearing the SWC2 bit to 0 (transfer clock) allows the transfer clock to be output from or supplied to the SCL_2 pin, instead of outputting a low-level signal.

If the SWC9 bit in the U2SMR4 register is set to 1 (SCL_2 hold low enabled) when the CKPH bit in the U2SMR3 register is set to 1, the SCL_2 pin is fixed to low-level output at the falling edge of the clock pulse next to the ninth. Setting the SWC9 bit to 0 (SCL_2 hold low disabled) frees the SCL_2 pin from low-level output.

14.1.3.5 SDA Output

The data written to the bit 7 to bit 0 (D_7 to D_0) in the U2TB register is sequentially output beginning with D_7. The ninth bit (D_8) is ACK or NACK.

The initial value of SDA_2 transmit output can only be set when IICM is set to 1 (I^2C bus mode) and bits SMD2 to SMD0 in the U2MR register is set to 000_2 (serial I/O disabled).

Bits DL2 to DL0 in the U2SMR3 register allow to add no delays or a delay of 2 to 8 U2BRG count source clock cycles to SDA_2 output.

Setting the SDHI bit in the U2SMR2 register to 1 (SDA_2 output disabled) forcibly places the SDA_2 pin in the high-impedance state. Do not write to the SDHI bit synchronously with the rising edge of the UART2 transfer clock. This is because the ABT bit may inadvertently be set to 1 (detected).

14.1.3.6 SDA Input

When the IICM2 bit is set to 0, the 1st to 8th bits (D_7 to D_0) in the received data are stored in bits 7 to 0 in the U2RB register. The 9th bit (D_8) is ACK or NACK.

When the IICM2 bit is set to 1, the 1st to 7th bits (D_7 to D_1) in the received data are stored in the bit 6 to bit 0 in the U2RB register and the 8th bit (D_0) is stored in the bit 8 in the U2RB register. Even when the IICM2 bit is set to 1, providing the CKPH bit is set to 1, the same data as when the IICM2 bit is set to 0 can be read out by reading the U2RB register after the rising edge of the corresponding clock pulse of 9th bit.

14.1.6.2 Format

• Direct Format

Set the PRY bit in the U2MR register to 1, the UFORM bit in U2C0 register to 0 and the U2LCH bit in U2C1 register to 0.

• Inverse Format

Set the PRY bit to 0, UFORM bit to 1 and U2LCH bit to 1.

Figure 14.34 shows the SIM interface format.

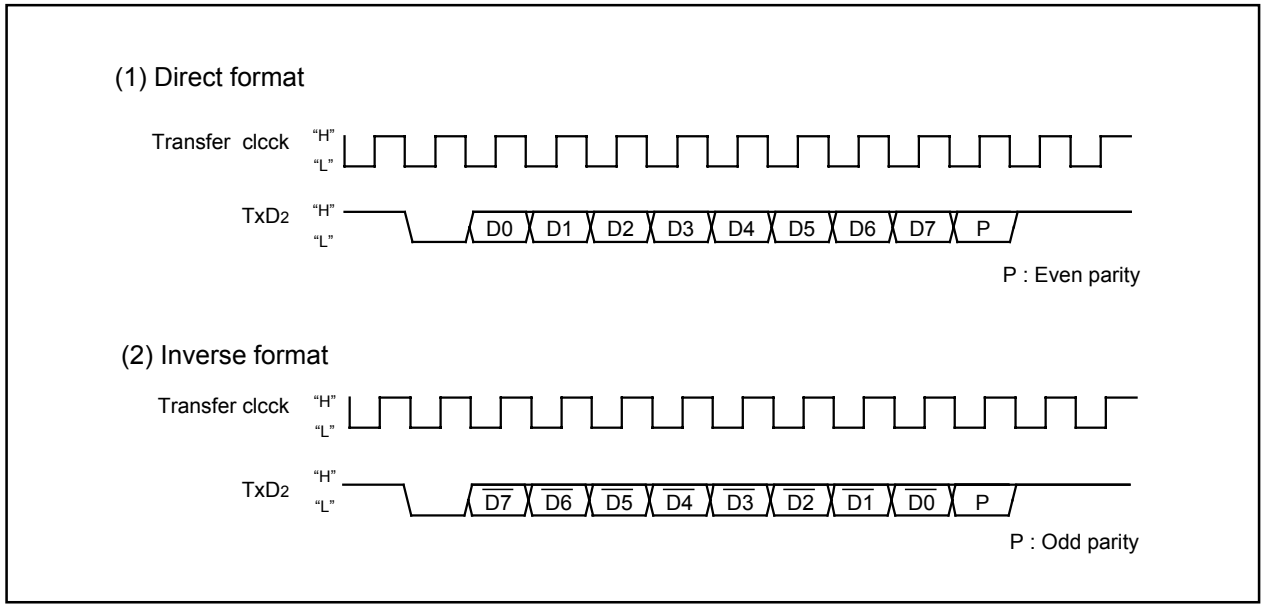


Figure 14.34 SIM Interface Format

Table 14.20 SI/O3 and SI/O4 Specifications

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• The SMi6 bit in the SiC (i=3, 4) register is set to 1 (internal clock) : fj/ (2(n+1)) fj = f1SIO, f2SIO, f8SIO, f32SIO. n=Setting value of SiBRG register 0016 to FF16. • SMi6 bit is set to 0 (external clock) : Input from CLKi pin (1)
Transmission/reception start condition	• Before transmission/reception can start, the following requirements must be met Write transmit data to the SiTRR register (2, 3)
Interrupt request generation timing	• When the SMi4 bit in the SiC register is set to 0 The rising edge of the last transfer clock pulse (4) • When SMi4 is set to 1 The falling edge of the last transfer clock pulse (4)
CLKi pin fucntion	I/O port, transfer clock input, transfer clock output
SOUTi pin function	I/O port, transmit data output, high-impedance
SINi pin function	I/O port, receive data input
Select function	• LSB first or MSB first selection Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected • Function for setting an SOUTi initial value set function When the SMi6 bit in the SiC register is set to 0 (external clock), the SOUTi pin output level while not tranmitting can be selected. • CLK polarity selection Whether transmit data is output/input timing at the rising edge or falling edge of transfer clock can be selected.

NOTE:

1. To set the SMi6 bit in the SiC register to 0 (external clock), follow the procedure described below.
   - If the SMi4 bit in the SiC register is set to 0, write transmit data to the SiTRR register while input on the CLKi pin is high. The same applies when rewriting the SMi7 bit in the SiC register.
   - If the SMi4 bit is set to 1, write transmit data to the SiTRR register while input on the CLKi pin is low. The same applies when rewriting the SMi7 bit.
   - Because shift operation continues as long as the transfer clock is supplied to the SI/Oi circuit, stop the transfer clock 2. Unlike UART0 to UART2, SI/Oi (i = 3 to 4) is not separated between the transfer register and buffer. Therefore, do not write the next transmit data to the SiTRR register during transmission.
3. When the SMi6 bit in the SiC register is set to 1 (internal clock), SOUTi retains the last data for a 1/2 transfer clock period after completion of transfer and, thereafter, goes to a high-impedance state. However, if transmit data is written to the SiTRR register during this period, SOUTi immediately goes to a high-impedance state, with the data hold time thereby reduced.
4. When the SMi6 bit in the SiC register is set to 1 (internal clock), the transfer clock stops in the high state if the SMi4 bit is set to 0, or stops in the low state if the SMi4 bit is set to 1.

15.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, analog voltages applied to the selected pins are repeatedly converted to a digital code. **Table 15.6** shows the repeat sweep mode 0 specifications. **Figure 15.12** shows the operation example in repeat sweep mode 0. **Figure 15.13** shows the ADCON0 to ADCON2 registers in repeat sweep mode 0.

Table 15.6 Repeat Sweep Mode 0 Specifications

Item	Specification
Function	Bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0 in the ADCON2 register select pins. Analog voltage applied to the selected pins is repeatedly converted to a digital code
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is 0 (software trigger) Set the ADST bit in the ADCON0 register to 1 (A/D conversion started) • When the TRG bit in the ADCON0 register is 1 (Hardware trigger) The $\overline{\text{ADTRG}}$ pin input changes state from "H" to "L" after setting the ADST bit to 1 (A/D conversion started)
A/D Conversion Stop Condition	Set the ADST bit to 0 (A/D conversion halted)
Interrupt Request Generation Timing	None generated
Analog Input Pin	Select from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), AN_0 to AN_7 (8 pins) (1)
Readout of A/D Conversion Result	Readout one of registers AD0 to AD7 that corresponds to the selected pin

NOTES:

1. $AN0_0$ to $AN0_7$, $AN2_0$ to $AN2_7$, and $AN3_0$ to $AN3_2$ can be used in the same way as AN_0 to AN_7. However, all input pins need to belong to the same group.

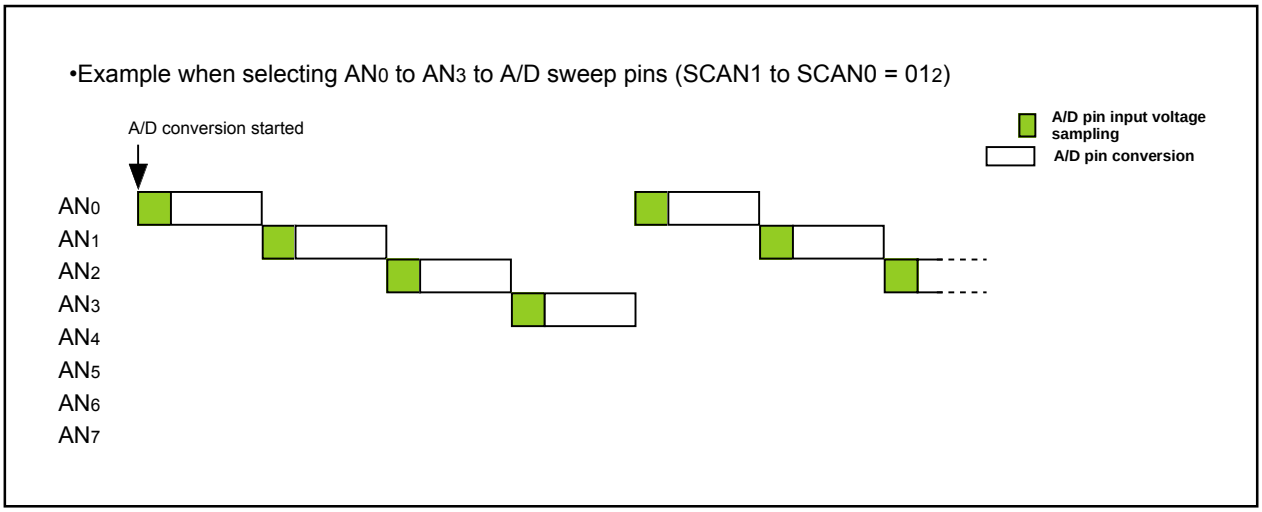


Figure 15.12 Operation Example in Repeat Sweep Mode 0

A/D Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
			1	1			

Symbol: ADCON0 Address: 03D6$_{16}$ After Reset: 00000XXX$_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit	Invalid in repeat sweep mode 1	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0	b4 b3 1 1: Repeat sweep mode 0 or repeat sweep mode 1	RW
MD1			RW
TRG	Trigger select bit	0: Software trigger 1: Hardware trigger ($\overline{\text{ADTRG}}$ trigger)	RW
ADST	A/D conversion start flag	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTE:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.

A/D Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	1			0		

Symbol: ADCON1 Address: 03D7$_{16}$ After Reset: 00$_{16}$

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit (2)	When repeat sweep mode 0 is selected, b1 b0 0 0: AN$_0$ (1 pin) 0 1: AN$_0$ to AN$_1$ (2 pins) 1 0: AN$_0$ to AN$_2$ (3 pins) 1 1: AN$_0$ to AN$_3$ (4 pins)	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	1: Repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect Bit (3)	1: Vref connected	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN00 to AN07, AN20 to AN27, and AN30 to AN32 can be used in the same way as AN$_0$ to AN$_7$. Use bits ADGSEL1 and ADGSEL 0 in the ADCON2 register to select the desired pin.
3. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 µs or more before starting A/D conversion.

A/D Control Register 2(1)

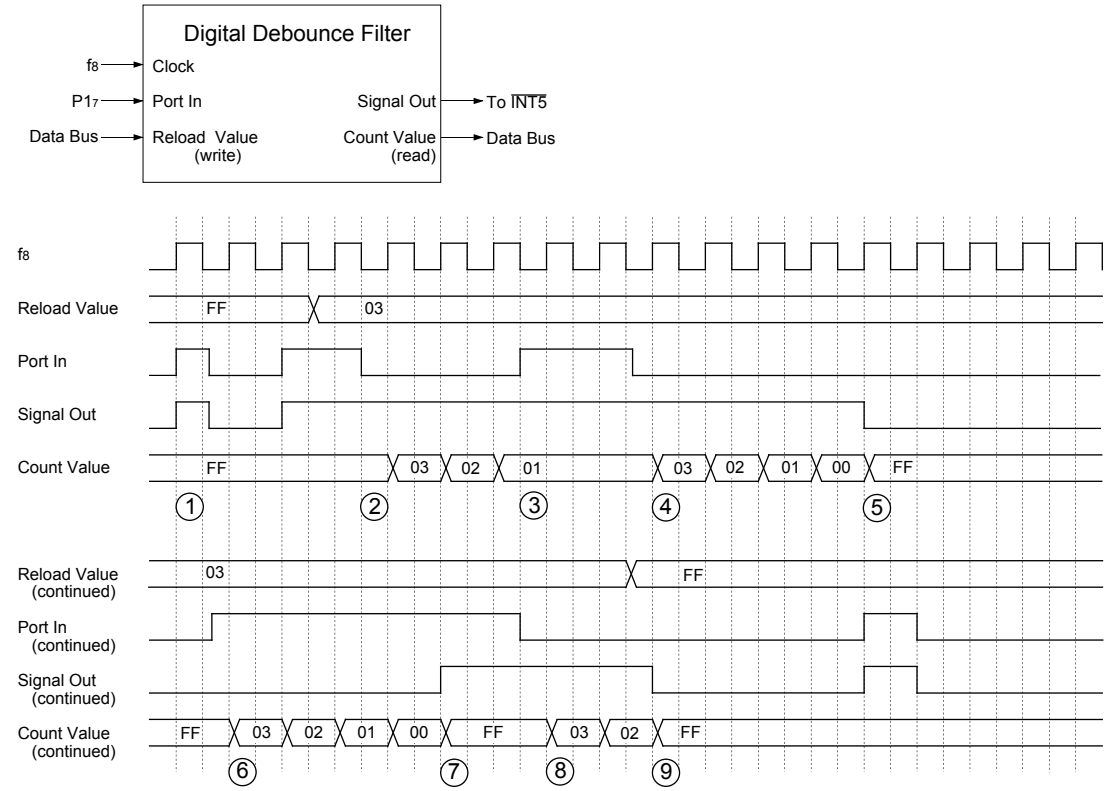
b7	b6	b5	b4	b3	b2	b1	b0
X	X	0		0			

Symbol: ADCON2 Address: 03D4$_{16}$ After Reset: 00$_{16}$

Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	0: Without sample and hold 1: With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
$\overline{\text{(b3)}}$	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	Set to 0 in repeat sweep mode 1	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

Figure 15.15 ADCON0 to ADCON2 Registers in Repeat Sweep Mode 1



1. (Condition after reset). P17DDR=FF16. Pin input signal will be output directly.
2. Set the P17DDR register to 03₁₆. The P17DDR register starts decrement along the f8 as a counter source, if the pin input level (e.g.,"L") and the signal output level (e.g.,"H") are not matched.
3. The P17DDR register will stops counting when the pin input level and the signal output level are matched (e.g., both levels are "H") while counting.
4. If the pin input level (e.g.,"L") and the signal output level (e.g.,"H") are not matched the P17DDR register will start decrement again after the setting value is reloaded.
5. When the P17DDR register is underflow, it stops counting and the signal output will output the same as pin input level (e.g."L").
6. If the pin input level (e.g.,"H") and the signal output level (e.g., "L") are not matched again, the P17DDR register will start decrement again after the setting value is reloaded.
7. When the P17DDR register is underflow, it stops counting and the signal output will output the same as pin input level (e.g."H").
8. If the pin input level (e.g.,"H") and the signal output level (e.g., "L") are not matched again, the P17DDR register will start decrement again after the setting value is reloaded.
9. Set the P17DDR register to FF16. The P17DDR register starts counting after the setting value is reloaded. Pin input signal will be output directly.

Figure 18.12 Functioning of Digital Debounce Filter

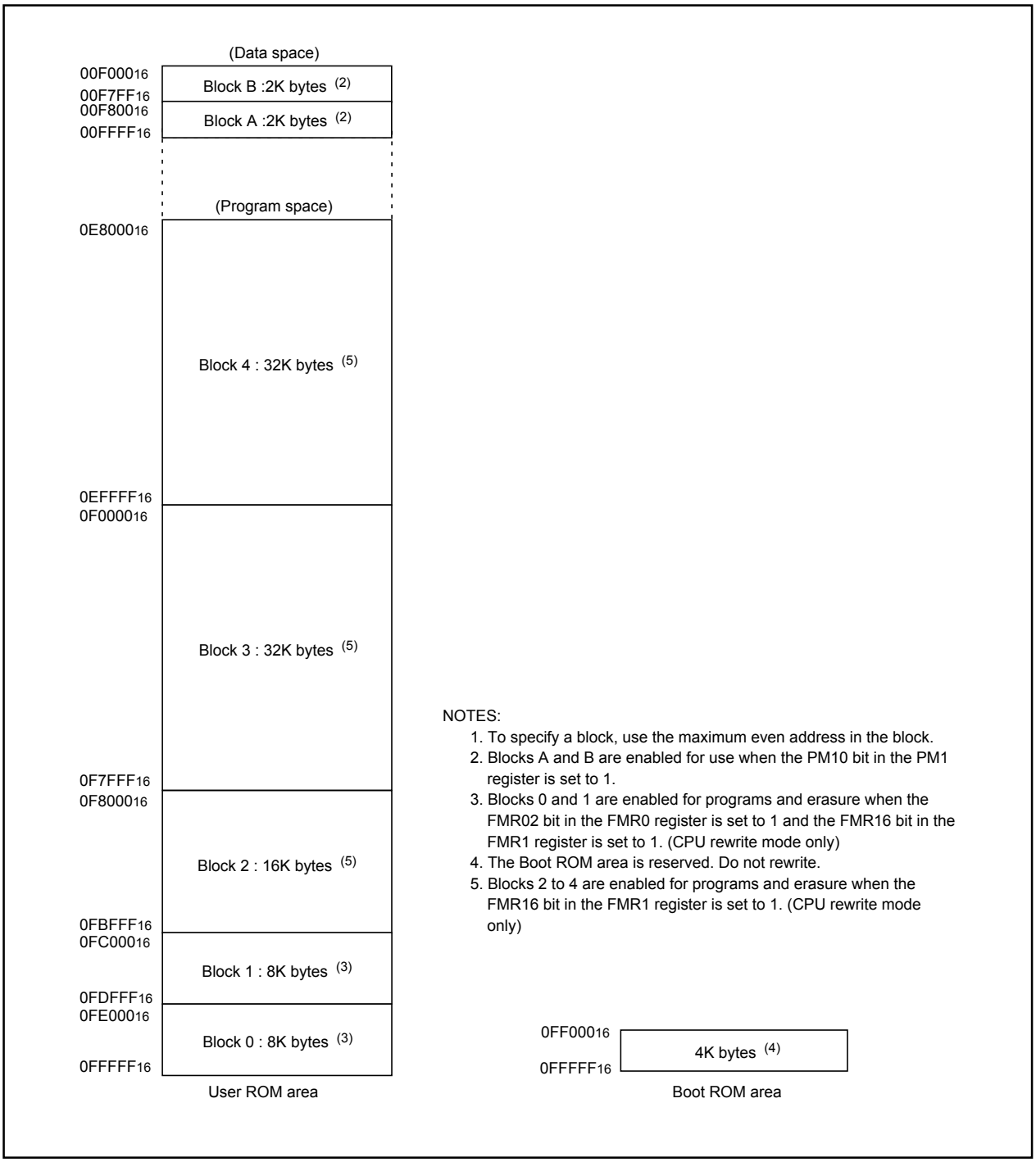


Figure 19.2 Flash Memory Block Diagram (ROM capacity 96 Kbytes)

19.5.2 Flash Memory Control Register 1 (FMR1)

•FMR11 Bit

EW mode 1 is entered by setting the FMR11 bit to 1 (EW mode 1). The FMR11 bit is valid only when the FMR01 bit is set to 1.

•FMR16 Bit

The combined setting of bits FMR02 and FMR16 enables program and erase in the user ROM area. To set the FMR16 bit to 1, first set it to 0 and then 1. The FMR16 bit is valid only when the FMR01 bit is set to 1 (CPU rewrite mode enable).

•FMR17 Bit

If the FMR17 bit is set to 1 (with wait state), 1 wait state is inserted when blocks A and B are accessed, regardless of the content of the PM17 bit in the PM1 register. The PM17 bit setting is reflected to access other blocks and internal RAM, regardless of the FMR17 bit setting.
Set the FMR17 bit to 1 (with wait state) to rewrite more than 100 times.

Table 19.4 Protection using FMR16 and FMR02

FMR16	FMR02	Block A, Block B	Block 0, Block 1	other user block
0	0	write enabled	write disabled	write disabled
0	1	write enabled	write disabled	write disabled
1	0	write enabled	write disabled	write enabled
1	1	write enabled	write enabled	write enabled

19.5.3 Flash Memory Control Register 4 (FMR4)

•FMR40 Bit

The erase-suspend function is enabled when the FMR40 bit is set to 1 (enabled).

•FMR41 Bit

When the FMR41 bit is set to 1 by program during auto-erasing in EW mode 0, erase-suspend mode is entered. In EW mode 1, the FMR41 bit is automatically set to 1 (suspend request) to enter erase-suspend mode when an enabled interrupt request is generated. Set the FMR41 bit to 0 (erase restart) to restart an auto-erasing operation.

•FMR46 Bit

The FMR46 bit is set to 0 during auto-erasing. It is set to 1 in erase-suspend mode.
Do not access to flash memory when the FMR46 bit is set to 0.

19.10 Parallel I/O Mode

In parallel input/output mode, the user ROM can be rewritten by a parallel programmer supporting the M16C/28 group (T-ver./V-ver.). Contact your parallel programmer manufacturer for more information on the parallel programmer. Refer to the user's manual included with your parallel programmer for instructions.

19.10.1 ROM Code Protect Function

The ROM code protect function prevents the flash memory from being read or rewritten. (Refer to **19.3 Functions To Prevent Flash Memory from Rewriting**).

V_{CC} = 5V

Table 20.7 Electrical Characteristics (1)

Symbol	Parameter			Condition	Standard			Unit
					Min.	Typ.	Max.	
V_{OH}	Output High ("H") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$		I_{OH}=-5mA	V_{CC}-2.0		V_{CC}	V
V_{OH}	Output High ("H") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$		I_{OH}=-200µA	V_{CC}-0.3		V_{CC}	V
V_{OH}	Output High ("H") Voltage	X_{OUT}	High Power	I_{OH}=-1mA	V_{CC}-2.0		V_{CC}	V
			Low Power	I_{OH}=-0.5mA	V_{CC}-2.0		V_{CC}	
	Output High ("H") Voltage	X_{COUT}	High Power	No load applied		2.5		V
			Low Power	No load applied		1.6		
V_{OL}	Output Low ("L") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$		I_{OL}=5mA			2.0	V
V_{OL}	Output Low ("L") Voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$		I_{OL}=200µA			0.45	V
V_{OL}	Output Low ("L") Voltage	X_{OUT}	High Power	I_{OL}=1mA			2.0	V
			Low Power	I_{OL}=0.5mA			2.0	
	Output Low ("L") Voltage	X_{COUT}	High Power	No load applied		0		V
			Low Power	No load applied		0		
V_{T+}-V_{T-}	Hysteresis	$TA0_{IN}$-$TA4_{IN}$, $TB0_{IN}$-$TB2_{IN}$, $\overline{INT0}$-$\overline{INT5}$, $\overline{NMI}$, $\overline{ADTRG}$, $\overline{CTS0}$-$\overline{CTS2}$, SCL, SDA, $CLK0$-$CLK2$, $TA2_{OUT}$-$TA4_{OUT}$, $\overline{KI0}$-$\overline{KI3}$, $RXD0$-$RXD2$, $SIN3$, $SIN4$			0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$			0.2		2.5	V
V_{T+}-V_{T-}	Hysteresis	X_{IN}			0.2		0.8	V
I_{IH}	Input High ("H") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ X_{IN}, $\overline{RESET}$, CNV_{SS}		V_I=5V			5.0	µA
I_{IL}	Input Low ("L") Current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$ X_{IN}, $\overline{RESET}$, CNV_{SS}		V_I=0V			-5.0	µA
R_{PULLUP}	Pull-up Resistance	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$, $P10_0$ to $P10_7$		V_I=0V	30	50	170	kΩ
Rf_{XIN}	Feedback Resistance	X_{IN}				1.5		MΩ
Rf_{XCIN}	Feedback Resistance	X_{CIN}				15		MΩ
V_{RAM}	RAM Standby Voltage			In stop mode	2.0			V

NOTE:

1. Referenced to V_{CC}=4.2 to 5.5V, V_{SS}=0V at Topr=-40 to 85 ° C, f(BCLK)=20MHz unless otherwise specified.

Vcc = 5V

Timing Requirements

(Vcc=5V, Vss=0V, at Topr=-40 to 125°C unless otherwise specified)

Table 20.48 Timer A Input (Counter Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN Input Cycle Time	100		ns
$t_{w(TAH)}$	TAiIN Input High ("H") Width	40		ns
$t_{w(TAL)}$	TAiIN Input Low ("L") Width	40		ns

Table 20.49 Timer A Input (Gating Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN Input Cycle Time	400		ns
$t_{w(TAH)}$	TAiIN Input High ("H") Width	200		ns
$t_{w(TAL)}$	TAiIN Input Low ("L") Width	200		ns

Table 20.50 Timer A Input (External Trigger Input in One-shot Timer Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN Input Cycle Time	200		ns
$t_{w(TAH)}$	TAiIN Input High ("H") Width	100		ns
$t_{w(TAL)}$	TAiIN Input Low ("L") Width	100		ns

Table 20.51 Timer A Input (External Trigger Input in Pulse Width Modulation Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN Input High ("H") Width	100		ns
$t_{w(TAL)}$	TAiIN Input Low ("L") Width	100		ns

Table 20.52 Timer A Input (Counter Increment/decrement Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT Input Cycle Time	2000		ns
$t_{w(UPH)}$	TAiOUT Input High ("H") Width	1000		ns
$t_{w(UPL)}$	TAiOUT Input Low ("L") Width	1000		ns
$t_{su(UP-TIN)}$	TAiOUT Input Setup Time	400		ns
$t_{h(TIN-UP)}$	TAiOUT Input Hold Time	400		ns

Table 20.53 Timer A Input (Two-phase Pulse Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN Input Cycle Time	800		ns
$t_{su(TAIN-TAOUT)}$	TAiOUT Input Setup Time	200		ns
$t_{su(TAOUT-TAIN)}$	TAiIN Input Setup Time	200		ns

21.2 Clock Generation Circuit

21.2.1 PLL Frequency Synthesizer

Stabilize supply voltage so that the standard of the power supply ripple is met.

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
$f_{(ripple)}$	Power supply ripple allowable frequency(V_{CC})				10	kHz
$V_{p-p(ripple)}$	Power supply ripple allowabled amplitude voltage	(V_{CC}=5V)			0.5	V
		(V_{CC}=3V)			0.3	V
$V_{CC(\|DV/DT\|)}$	Power supply ripple rising/falling gradient	(V_{CC}=5V)			0.3	V/ms
		(V_{CC}=3V)			0.3	V/ms

$f_{(ripple)}$
Power supply ripple allowable frequency (V_{CC})

$V_{p-p(ripple)}$
Power supply ripple allowable amplitude voltage

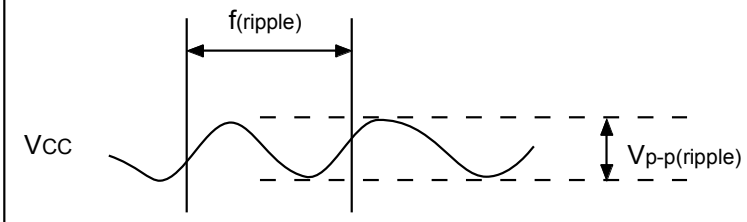


Figure 21.1 Voltage Fluctuation Timing

21.4.6 Rewrite the Interrupt Control Register

(1) The interrupt control register for any interrupt should be modified in places where no requests for that interrupt may occur. Otherwise, disable the interrupt before rewriting the interrupt control register.

(2) To rewrite the interrupt control register for any interrupt after disabling that interrupt, be careful with the instruction to be used.

Changing any bit other than the IR bit

If while executing an instruction, a request for an interrupt controlled by the register being modified occurs, the IR bit in the register may not be set to 1 (interrupt requested), with the result that the interrupt request is ignored. If such a situation presents a problem, use the instructions shown below to modify the register.

Usable instructions: AND, OR, BCLR, BSET

Changing the IR bit

Depending on the instruction used, the IR bit may not always be cleared to 0 (interrupt not requested). Therefore, be sure to use the MOV instruction to clear the IR bit.

(3) When using the I flag to disable an interrupt, refer to the sample program fragments shown below as you set the I flag. (Refer to (2) for details about rewrite the interrupt control registers in the sample program fragments.)

Examples 1 through 3 show how to prevent the I flag from being set to 1 (interrupts enabled) before the interrupt control register is rewrited, due to the internal bus and the instruction queue buffer.

Example 1: Using the NOP instruction to keep the program waiting until the interrupt control register is modified

```
INT_SWITCH1:
      FCLR    I              ; Disable interrupts
      AND.B   #00h, 0055h    ;Set the TA0IC register to 00₁₆
      NOP                    ;
      NOP
      FSET    I              ; Enable interrupts
```

The number of NOP instruction is as follows.
PM20 = 1 (1 wait) : 2, PM20 = 0 (2 waits): 3

Example 2:Using the dummy read to keep the FSET instruction waiting

```
INT_SWITCH2:
      FCLR    I              ; Disable interrupts
      AND.B   #00h, 0055h    ; Set the TA0IC register to 00₁₆
      MOV.W   MEM, R0        ; Dummy read
      FSET    I              ; Enable interrupts
```

Example 3:Using the POPC instruction to changing the I flag

```
INT_SWITCH3:
      PUSHC   FLG
      FCLR    I              ; Disable interrupts
      AND.B   #00h, 0055h    ; Set the TA0IC register to 00₁₆
      POPC    FLG            ; Enable interrupts
```

21.4.7 Watchdog Timer Interrupt

Initialize the watchdog timer after the watchdog timer interrupt occurs.

21.13 Mask ROM Version

21.13.1 Internal ROM Area

In the masked ROM version, do not write to internal ROM area. Writing to the area may increase power consumption.

21.13.2 Reserved Bit

The b3 to b0 in addresses $0FFFFF_{16}$ are reserved bits. Set these bits to 1111_2.

Rev.	Date	Description	
		Page	Summary
		365	• **21.4.6 Rewrite the Interrupt Control Register** Example 1 modified
		371	• **21.6.3 Three-phase Motor Control Timer Function** newly added
		372	• **21.7.1 Rewrite the G1IR Register** Description modified
		373	• **21.7.4 IC/OC Base Timer Interrupt** Section newly added
		381	• **21.13.1 Internal ROM Area** partially added
		383	• **21.14.9 Interrupts EW Mode 1** Description about watchdog timer interrupt deleted • **21.14.10 How to Access** partially deleted • **21.14.13 Regarding Programming/Erasure Times and Execution Time** Description partially modified
		-	**Functional Comparison** • **Difference between M16C/28 Group and M16C/29 Group (Normal-ver.)** is deleted
		388	• **Appendix 2.1 Difference between M16C/28 Group Normal-ver. and m16C/28 Group T-ver./V-ver.** flash memory added