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	Not For New Designs
Core Processor	M16C/60
Core Size	16-Bit
Speed	20MHz
Connectivity	CANbus, I²C, IEBus, SIO, UART/USART
Peripherals	DMA, POR, PWM, Voltage Detect, WDT
Number of I/O	71
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 27x10b
Oscillator Type	Internal
Operating Temperature	-20°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/m30290fchp-u5a

Quick Reference to Pages Classified by Address

Address	Register	Symbol	Page
0100_{16} 0101_{16} 0102_{16} 0103_{16} 0104_{16} 0105_{16}	CAN0 message box 10: Identifer/DLC		289
0106_{16} 0107_{16} 0108_{16} 0109_{16} $010A_{16}$ $010B_{16}$ $010C_{16}$ $010D_{16}$	CAN0 message box 10: Data field		289
$010E_{16}$ $010F_{16}$	CAN0 message box 10: time stamp		289
0110_{16} 0111_{16} 0112_{16} 0113_{16} 0114_{16} 0115_{16}	CAN0 message box 11: Identifier/DLC		289
0116_{16} 0117_{16} 0118_{16} 0119_{16} $011A_{16}$ $011B_{16}$ $011C_{16}$ $011D_{16}$	CAN0 message box 11: Data field		289
$011E_{16}$ $011F_{16}$	CAN0 message box 11: time stamp		289
0120_{16} 0121_{16} 0122_{16} 0123_{16} 0124_{16} 0125_{16}	CAN0 message box 12: Identifier/DLC		289
0126_{16} 0127_{16} 0128_{16} 0129_{16} $012A_{16}$ $012B_{16}$ $012C_{16}$ $012D_{16}$	CAN0 message box 12: Data field		289
$012E_{16}$ $012F_{16}$	CAN0 message box 12: time stamp		289
0130_{16} 0131_{16} 0132_{16} 0133_{16} 0134_{16} 0135_{16}	CAN0 message box 13: Identifier/DLC		289
0136_{16} 0137_{16} 0138_{16} 0139_{16} $013A_{16}$ $013B_{16}$ $013C_{16}$ $013D_{16}$	CAN0 message box 13: Data field		289
$013E_{16}$ $013F_{16}$	CAN0 message box 13: time stamp		289

Address	Register	Symbol	Page
0140_{16} 0141_{16} 0142_{16} 0143_{16} 0144_{16} 0145_{16}	CAN0 message box 14: Identifier/DLC		289
0146_{16} 0147_{16} 0148_{16} 0149_{16} $014A_{16}$ $014B_{16}$ $014C_{16}$ $014D_{16}$	CAN0 message box 14: Data field		289
$014E_{16}$ $014F_{16}$	CAN0 message box 14: time stamp		289
0150_{16} 0151_{16} 0152_{16} 0153_{16} 0154_{16} 0155_{16}	CAN0 message box 15: Identifier/DLC		289
0156_{16} 0157_{16} 0158_{16} 0159_{16} $015A_{16}$ $015B_{16}$ $015C_{16}$ $015D_{16}$	CAN0 message box 15: Data field		289
$015E_{16}$ $015F_{16}$	CAN0 message box 15: time stamp		289
0160_{16} 0161_{16} 0162_{16} 0163_{16} 0164_{16} 0165_{16}	CAN0 global mask register	C0GMR	291
0166_{16} 0167_{16} 0168_{16} 0169_{16} $016A_{16}$ $016B_{16}$	CAN0 local mask A register	C0LMAR	291
$016C_{16}$ $016D_{16}$ $016E_{16}$ $016F_{16}$ 0170_{16} 0171_{16}	CAN0 local mask B register	C0LMBR	291
0172_{16}			
0173_{16}			
0174_{16}			
0175_{16}			
0176_{16}			
0177_{16}			
0178_{16}			
0179_{16}			
$017A_{16}$			
$017B_{16}$			
$017C_{16}$			
$017D_{16}$			
$017E_{16}$			
$017F_{16}$			

Note: The blank areas are reserved and cannot be accessed by users.

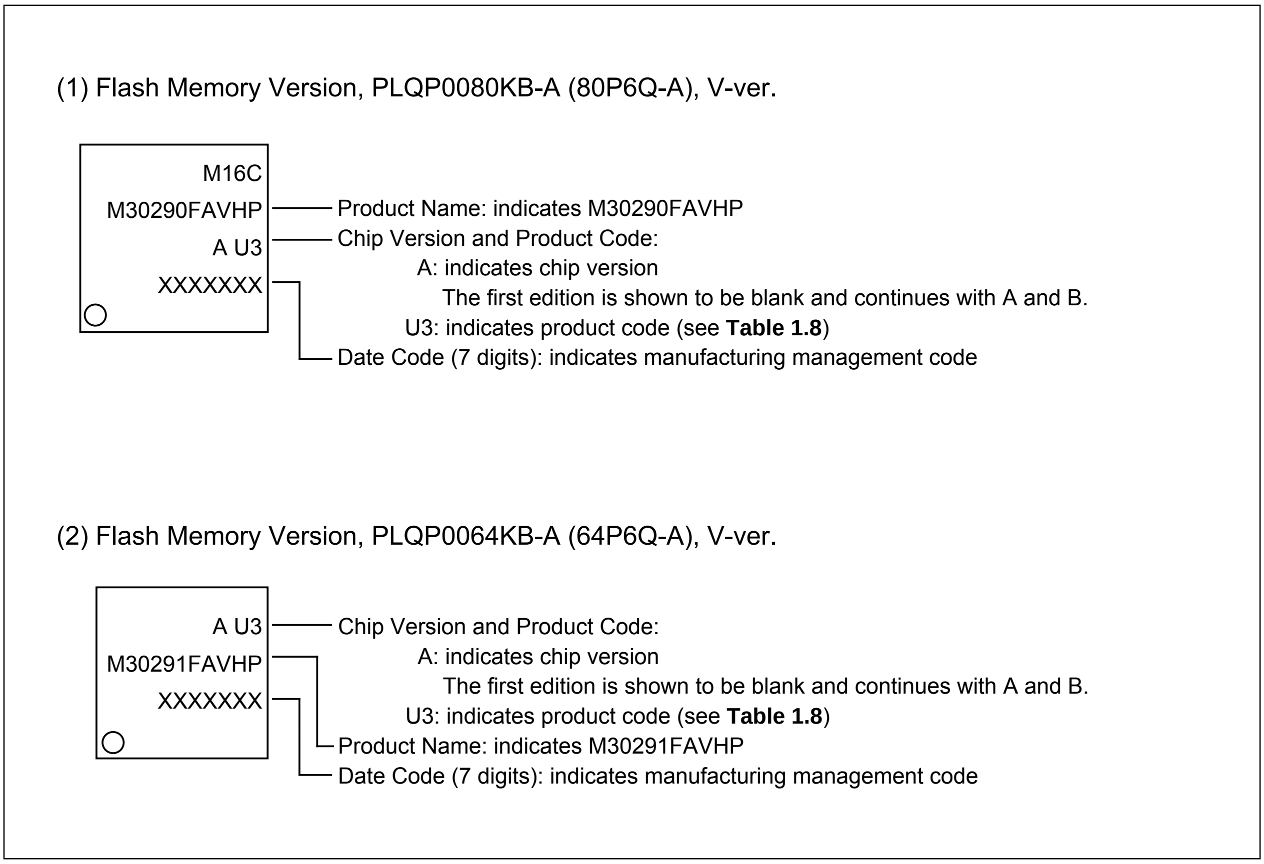


Figure 1.6 Marking Diagrams of Flash Memory Version - M16C/29 Group V-ver. (Top View)

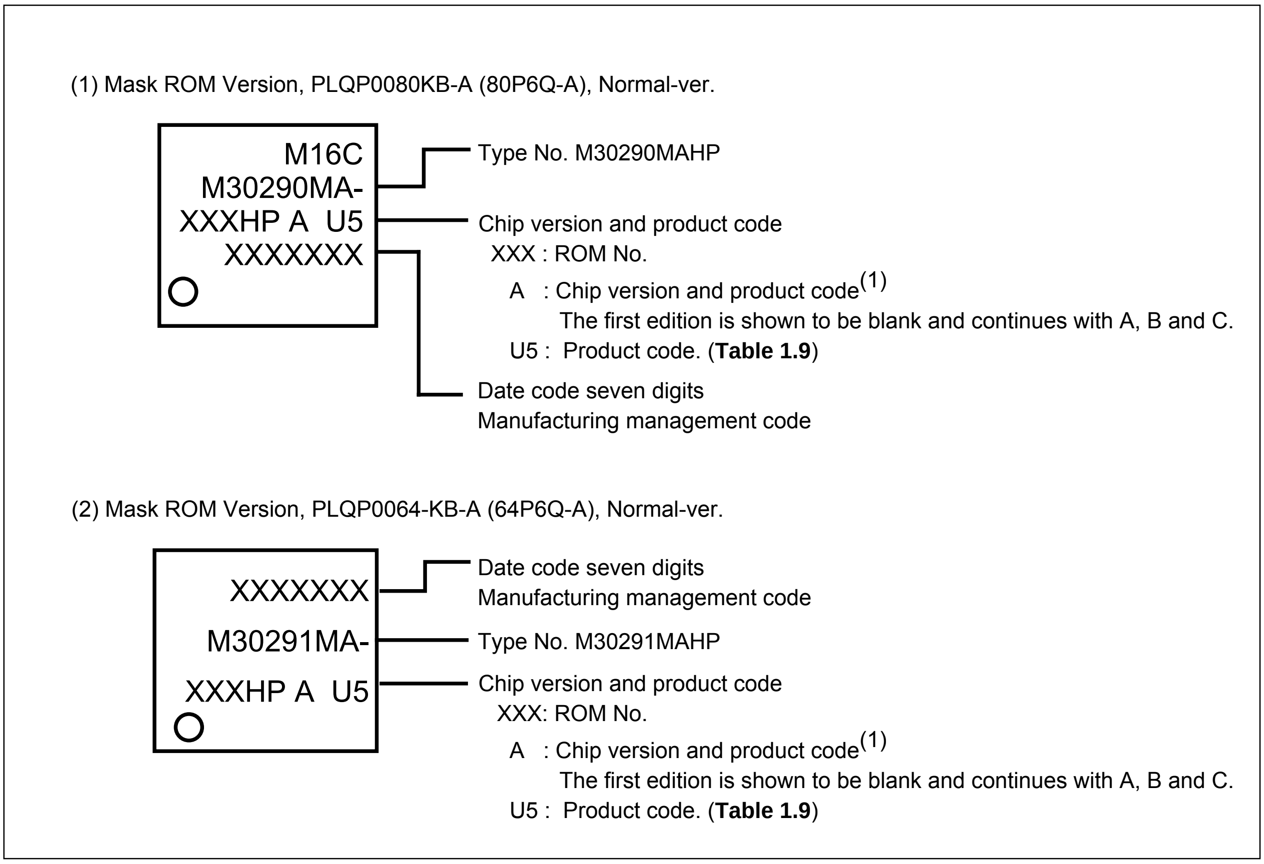


Figure 1.7 Marking Diagrams of Mask ROM Version - M16C/29 Group Normal-ver. (Top View)

9.1.1 Software Interrupts

A software interrupt occurs when executing certain instructions. Software interrupts are non-maskable interrupts.

9.1.1.1 Undefined Instruction Interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

9.1.1.2 Overflow Interrupt

An overflow interrupt occurs when executing the INTO instruction with the O flag set to 1 (the operation resulted in an overflow). The following are instructions whose O flag changes by arithmetic: ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

9.1.1.3 BRK Interrupt

A BRK interrupt occurs when executing the BRK instruction.

9.1.1.4 INT Instruction Interrupt

An INT instruction interrupt occurs when executing the INT instruction. Software interrupt Nos. 0 to 63 can be specified for the INT instruction. Because software interrupt Nos. 1 to 31 are assigned to peripheral function interrupts, the same interrupt routine as for peripheral function interrupts can be executed by executing the INT instruction.
In software interrupt Nos. 0 to 31, the U flag is saved to the stack during instruction execution and is cleared to 0 (ISP selected) before executing an interrupt sequence. The U flag is restored from the stack when returning from the interrupt routine. In software interrupt Nos. 32 to 63, the U flag does not change state during instruction execution, and the SP then selected is used.

10. Watchdog Timer

The watchdog timer is the function of detecting when the program is out of control. Therefore, we recommend using the watchdog timer to improve reliability of a system. The watchdog timer contains a 15-bit counter which counts down the clock derived by dividing the CPU clock using the prescaler. Whether to generate a watchdog timer interrupt request or apply a watchdog timer reset as an operation to be performed when the watchdog timer underflows after reaching the terminal count can be selected using the PM12 bit in the PM1 register. The PM12 bit can only be set to 1 (reset). Once this bit is set to 1, it cannot be set to 0 (watchdog timer interrupt) in a program. Refer to **5.3 Watchdog Timer Reset** for the details of watchdog timer reset.
When the main clock source is selected for CPU clock, on-chip oscillator clock, PLL clock, the WDC7 bit in the WDC register value for prescaler can be chosen to be 16 or 128. If a sub-clock is selected for CPU clock, the prescaler is always 2 no matter how the WDC7 bit is set. The period of watchdog timer can be calculated as given below. The period of watchdog timer is, however, subject to an error due to the prescaler.

With main clock source chosen for CPU clock, on-chip oscillator clock, PLL clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (16 or 128) X Watchdog timer count (32768)}}{\text{CPU clock}}$$

With sub-clock chosen for CPU clock

$$\text{Watchdog timer period} = \frac{\text{Prescaler dividing (2) X Watchdog timer count (32768)}}{\text{CPU clock}}$$

For example, when CPU clock is set to 16 MHz and the divide-by-N value for the prescale ris set to 16, the watchdog timer period is approx. 32.8 ms.
The watchdog timer is initialized by writing to the WDTS register. The prescaler is initialized after reset. Note that the watchdog timer and the prescaler both are inactive after reset, so that the watchdog timer is activated to start counting by writing to the WDTS register.
Write the WDTS register with shorter cycle than the watchdog timer cycle. Set the WDTS register also in the beginning of the watchdog timer interrupt routine.
In stop mode and wait mode, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.
Figure 10.1 shows the block diagram of the watchdog timer. Figure 10.2 shows the watchdog timer-related registers.

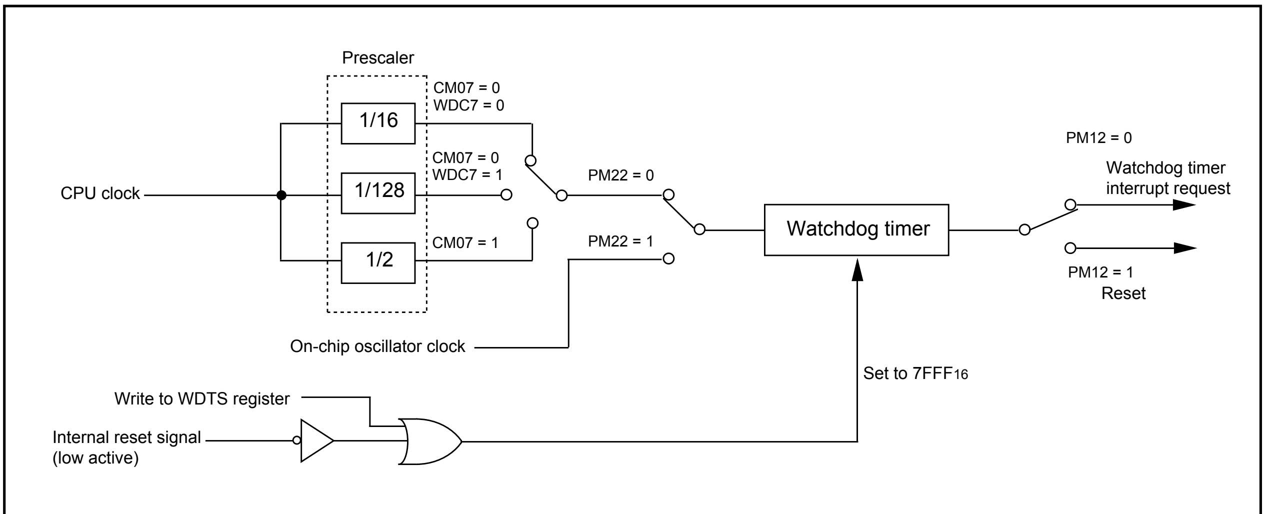
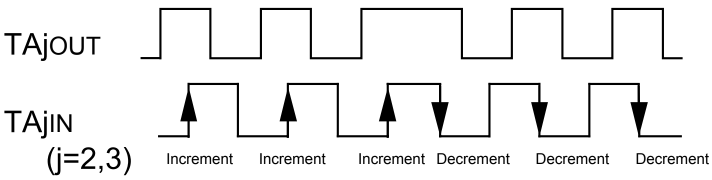
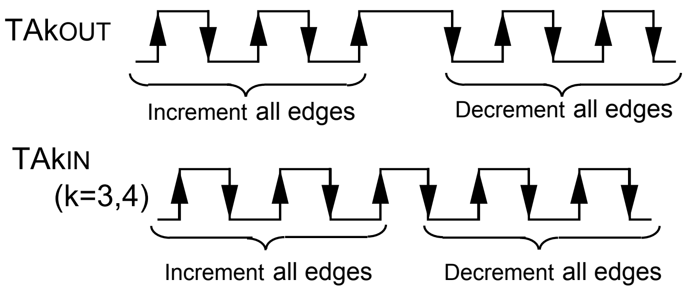


Figure 10.1 Watchdog Timer Block Diagram

Table 12.3 Specifications in Event Counter Mode
(when processing two-phase pulse signal with timers A2, A3 and A4)

Item	Specification
Count source	• Two-phase pulse signals input to TAiIN or TAiOUT pins (i = 2 to 4)
Count operation	• Increment or down-count can be selected by two-phase pulse signal • When the timer overflows or underflows, it reloads the reload register contents and continues counting. When operating in free-running mode, the timer continues counting without reloading.
Divide ratio	1/ ($FFFF_{16}$ - n + 1) for increment 1/ (n + 1) for down-count n : set value of TAi register 0000_{16} to $FFFF_{16}$
Count start condition	Set TAiS bit in the TABSR register to 1 (start counting)
Count stop condition	Set TAiS bit to 0 (stop counting)
Interrupt request generation timing	Timer overflow or underflow
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read by reading timer A2, A3 or A4 register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TAi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TAi register is written to reload register (Transferred to counter when reloaded next)
Select function (Note)	• Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on TAjIN pin when input signals on TAjOUT pin is “H”.  • Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that TAkIN(k=3, 4) pin goes “H” when the input signal on TAkOUT pin is “H”, the timer counts up rising and falling edges on TAkOUT and TAkIN pins. If the phase relationship is such that TAkIN pin goes “L” when the input signal on TAkOUT pin is “H”, the timer counts down rising and falling edges on TAkOUT and TAkIN pins.  • Counter initialization by Z-phase input (timer A3) The timer count value is initialized to 0 by Z-phase input.

NOTE:
1. Only timer A3 is selectable. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

13.6.1 INPC17 Alternate Input Pin Selection

The input capture pin for IC/OC channel 7 can be assigned to one of two package pins. The CH7INSEL bit in the G1BCR0 register selects IC/OC INPC1$_7$ from P2$_7$/OUTC1$_7$/INPC1$_7$ or P17/$\overline{\text{INT5}}$/INPC1$_7$/IDU.

13.6.2 Digital Debounce Function for Pin P17/$\overline{\text{INT5}}$/INPC17

The $\overline{\text{INT5}}$/INPC1$_7$ input from the P1$_7$/$\overline{\text{INT5}}$/INPC1$_7$/IDU pin has an effective digital debounce function against a noise rejection. Refer to **19.6 Digital Debounce function** for this detail.

Table 14.7 lists the functions of the input/output pins in UART mode. **Table 14.8** lists the P64 pin functions during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs an “H”. (If the N-channel open-drain output is selected, this pin is in a high-impedance state.)

Table 14.7 I/O Pin Functions in UART mode(1)

Pin Name	Function	Method of Selection
TxDi (i = 0 to 2) (P63, P67, P70)	Serial data output	(Outputs "H" when performing reception only)
RxDi (P62, P66, P71)	Serial data input	PD6_2 bit, PD6_6 bit in the PD6 register and the PD7_1 bit in the PD7 register (Can be used as an input port when performing transmission only)
CLKi (P61, P65, P72)	Input/output port	Set the CKDIR bit in the UiMR register to 0
	Transfer clock input	Set the CKDIR bit in the UiMR register to 1 Set the PD6_1 bit and PD6_5 bit in the PD6 register to 0, PD7_2 bit in the PD7 register to 0
$\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ (P60, P64, P73)	$\overline{\text{CTS}}$ input	Set the CRD bit in the UiC0 register to 0 Set the CRS bit in the UiC0 register to 0 Set the PD6_0 bit and PD6_4 bit in the PD6 register to 0, the PD7_3 bit in the PD7 register 0
	$\overline{\text{RTS}}$ output	Set the CRD bit in the UiC0 register to 0 Set the CRS bit in the UiC0 register to 1
	Input/output port	Set the CRD bit in the UiC0 register 1

NOTE:
1. When the U1MAP bit in PACR register is set to 1 (P73 to P70), UART1 pin is assgined to P73 to P70.

Table 14.8 P64 Pin Functions in UART mode (1)

Pin Function	Bit Set Value				
	U1C0 register		UCON register		PD6 register
	CRD	CRS	RCSP	CLKMD1	PD6_4
P64	1	—	0	0	Input: 0, Output: 1
$\overline{\text{CTS1}}$	0	0	0	0	0
$\overline{\text{RTS1}}$	0	1	0	0	—
$\overline{\text{CTS0}}$ (2)	0	0	1	0	0

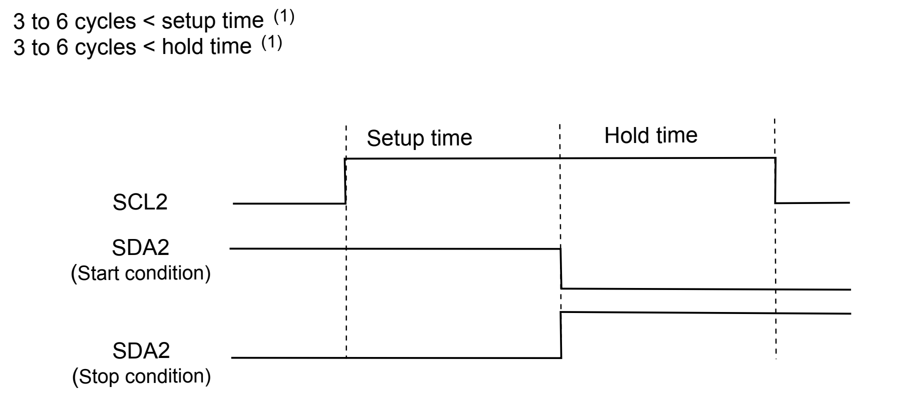
NOTES:
1. When the U1MAP bit in PACR register is 1 (P73 to P70), this table lists the P70 functions.
2. In addition to this, set the CRD bit in the U0C0 register to 0 ($\overline{\text{CTS0}}$/$\overline{\text{RTS0}}$ enabled) and the CRS bit in the U0C0 register to 1 ($\overline{\text{RTS0}}$ selected).

14.1.3.1 Detection of Start and Stop Condition

Whether a start or a stop condition has been detected is determined.

A start condition-detected interrupt request is generated when the SDA2 pin changes state from high to low while the SCL2 pin is in the high state. A stop condition-detected interrupt request is generated when the SDA2 pin changes state from low to high while the SCL2 pin is in the high state.

Because the start and stop condition-detected interrupts share the interrupt control register and vector, check the BBS bit in the U2SMR register to determine which interrupt source is requesting the interrupt.



NOTE:
1. When the PCLK1 bit in the PCLKR register is set to 1, the cycles indicates the f1SIO's generation frequency cycles; when PCLK1 bit is set to 0, the cycles indicated the f2SIO's generation frequency cycles.

Figure 14.24 Detection of Start and Stop Condition

14.1.3.2 Output of Start and Stop Condition

A start condition is generated by setting the STAREQ bit in the U2SMR4 register to 1 (start).

A restart condition is generated by setting the RSTAREQ bit in the U2SMR4 register to 1 (start).

A stop condition is generated by setting the STPREQ bit in the U2SMR4 register to 1 (start).

The output procedure is described below.

(1) Set the STAREQ bit, RSTAREQ bit or STPREQ bit to 1 (start).

(2) Set the STSPSEL bit in the U2SMR4 register to 1 (output).

Make sure that no interrupts or DMA transfers will occur between (1) and (2).

The function of the STSPSEL bit is shown in **Table 14.14** and **Figure 14.25**.

Table 14.14 STSPSEL Bit Functions

Function	STSPSEL = 0	STSPSEL = 1
Output of SCL2 and SDA2 pins	Output transfer clock and data/ Program with a port determines how the start condition or stop condition is output	The STAREQ, RSTAREQ and STPREQ bit determine how the start condition or stop condition is output
Start/stop condition interrupt request generation timing	Start/stop condition are detected	Start/stop condition generation are completed

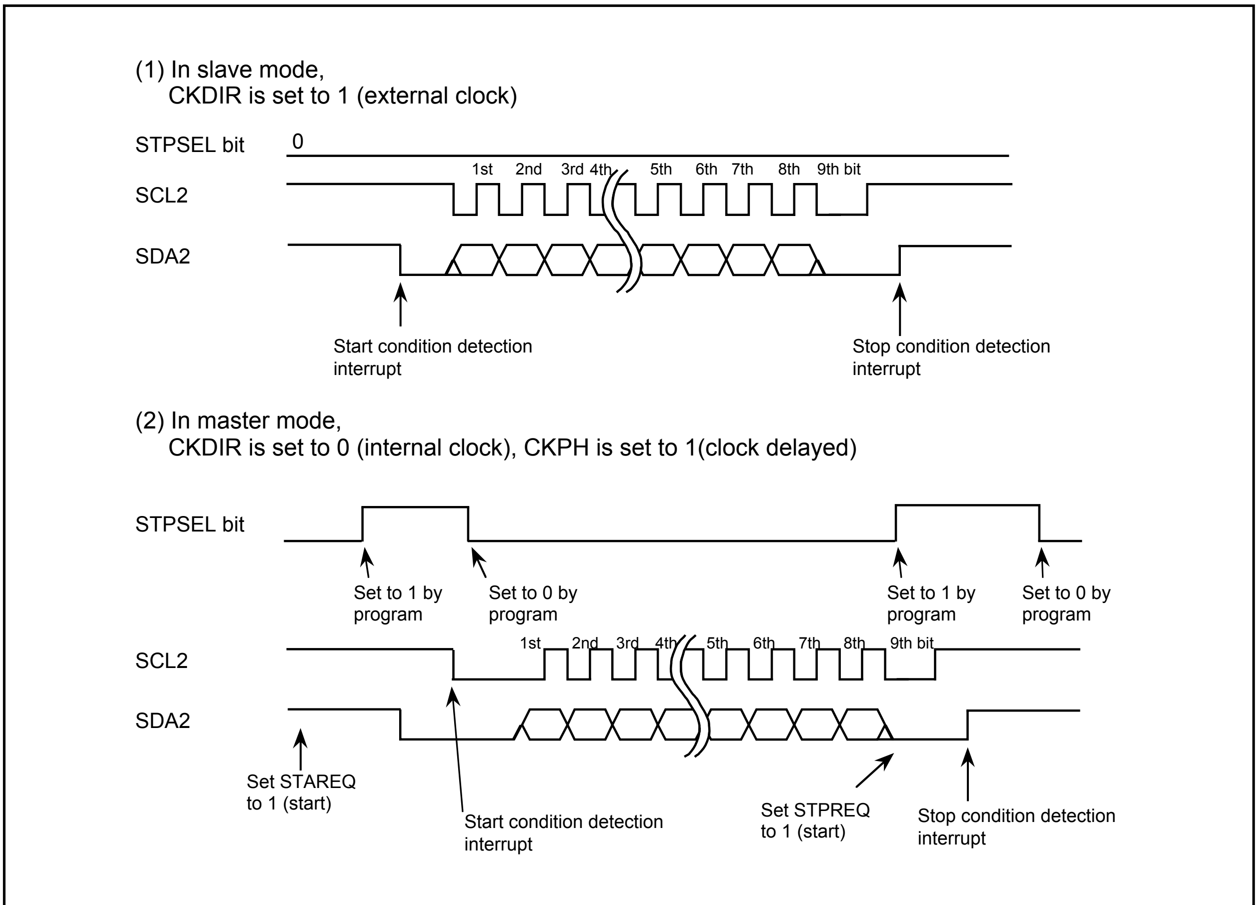


Figure 14.25 STSPSEL Bit Functions

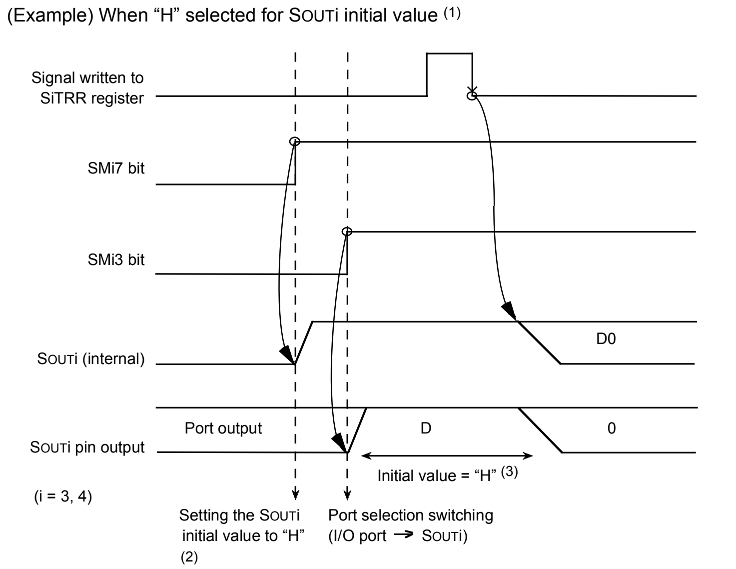
14.1.3.3 Arbitration

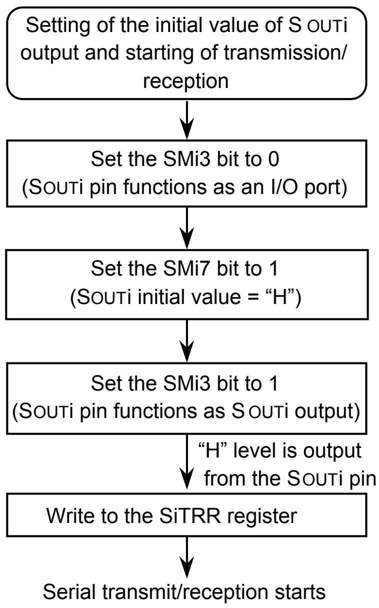
Unmatching of the transmit data and SDA2 pin input data is checked synchronously with the rising edge of SCL2. Use the ABC bit in the U2SMR register to select the timing at which the ABT bit in the U2RB register is updated. If the ABC bit is set to 0 (updated bitwise), the ABT bit is set to 1 at the same time unmatching is detected during check, and is cleared to 0 when not detected. In cases when the ABC bit is set to 1, if unmatching is detected even once during check, the ABT bit is set to 1 (unmatching detected) at the falling edge of the clock pulse of 9th bit. If the ABT bit needs to be updated bytewise, clear the ABT bit to 0 (undetected) after detecting acknowledge in the first byte, before transferring the next byte.

Setting the ALS bit in the U2SMR2 register to 1 (SDA2 output stop enabled) causes arbitration-lost to occur, in which case the SDA2 pin is placed in the high-impedance state at the same time the ABT bit is set to 1 (unmatching detected).

14.2.3 Functions for Setting an SOUTi Initial Value

If the SMi6 bit in SiC register is set to 0 (external clock), the SOUTi pin output level can be fixed high or low when not transferring data. However, when transmitting data consecutively, the last bit (bit 0) value of the last transmitted data is retained between the sccessive data transmissions. **Figure 14.39** shows the timing chart for setting an SOUTi initial value and how to set it.





NOTES:
1. This diagram applies to the case where the bits in the SiC register are set as follows:
   SMi2 = 0 (SOUTi output), SMi5 = 0 (LSB first) and SMi6 = 0 (external clock)
2. SOUTi can only be initialized when input on the CLKi pin is in the high state if the SMi4bit in the SiC register is set to 0 (transmit data output at the falling edge of the transfer clock) or in the low state if the SMi4 bit is set to 1 (transmit data output at the rising edge of the transfer clock).
3. If the SMi6 bit is set to 1 (internal clock) or if the SMi2 bit is set to 1 (SOUTi output disabled), this output goes to the high-impedance state.

Figure 14.39 SOUTi Initial Value Setting

15.5 Output Impedance of Sensor under A/D Conversion

To carry out A/D conversion properly, charging the internal capacitor C shown in **Figure 15.29** has to be completed within a specified period of time. T (sampling time) as the specified time. Let output impedance of sensor equivalent circuit be R0, MCU's internal resistance be R, precision (error) of the A/D converter be X, and the A/D converter's resolution be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

VC is generally $VC = VIN\{1-e^{-\frac{1}{c(R0+R)}t}\}$

And when t = T, $VC=VIN-\frac{X}{Y}VIN=VIN(1-\frac{X}{Y})$

$$e^{-\frac{1}{c(R0+R)}T} = \frac{X}{Y}$$

$$-\frac{1}{C(R0+R)}T = \ln\frac{X}{Y}$$

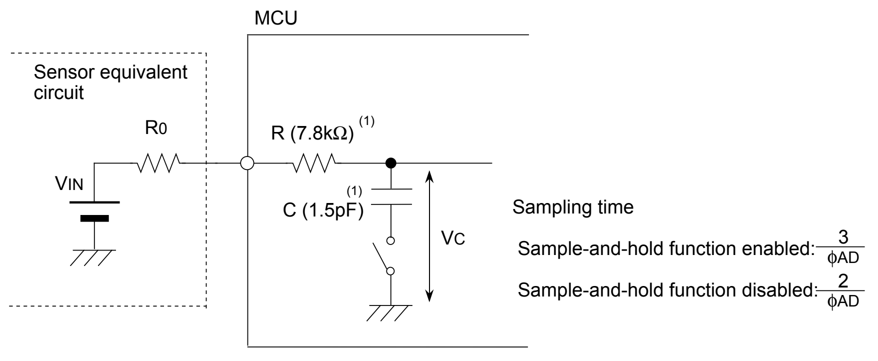
Hence, $R0 = -\frac{T}{C \bullet \ln\frac{X}{Y}} - R$

Figure 15.29 shows analog input pin and externalsensor equivalent circuit. When the difference between VIN and VC becomes 0.1 LSB, we find impedance R0 when voltage between pins. VC changes from 0 to VIN-(0.1/1024) VIN in timer T. (0.1/1024) means that A/D precision drop due to insufficient capacitor chage is held to 0.1LSB at time of A/D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB. When f(XIN) = 10MHz, T=0.3µs in the A/D conversion mode with sample & hold. Output inpedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.3µs, R = 7.8kΩ, C = 1.5pF, X = 0.1, and Y = 1024. Hence,

$$R0 = -\frac{0.3X10^{-6}}{1.5X10^{-12} \bullet \ln\frac{0.1}{1024}} - 7.8 \text{ X } 10^3 \cong 13.9 \text{ X } 10^3$$

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A/D converter turns out of be approximately 13.9kΩ.



NOTE:
1. Reference value

Figure 15.29 Analog Input Pin and External Sensor Equivalent Circuit

16.4 I2C0 Control Register 0 (S1D0)

The S1D0 register controls data communication format.

16.4.1 Bits 0 to 2: Bit Counter (BC0–BC2)

Bits BC2 to BC0 decide how many bits are in one byte data transferred next. After the selected numbers of bits are transferred successfully, I2C bus interface interrupt request is gnerated and bits BC2 to BC0 are reset to 000_2. At this time, if the ACK-CLK bit in the S20 register is set to 1 (with ACK clock), one bit for ACK clock is added to the numbers of bits selected by the BC2 to BC0 bits.

In addition, bits BC2 to BC0 become 000_2 even though the START condition is detected and the address data is transferred in 8 bits.

16.4.2 Bit 3: I2C Interface Enable Bit (ES0)

The ES0 bit enables to use the multi-master I2C bus interface. When the ES0 bit is set to 0, I2C bus interface is disabled and the SDA and SCL pins are placed in a high-h-impedance state. When the ES0 bit is set to 1, the interface is enabled.

When the ES0 bit is set to 0, the process is followed.

1)The bits in the S10 register are set as MST = 0, TRX = 0, PIN = 1, BB = 0, AL = 0, AAS = 0, ADR0 = 0

2)The S00 register cannot be written.

3)The TOF bit in the S4D0 register is set to 0 (time-out detection flag is not detected)

4)The I2C system clock (V_{IIC}) stops counting while the internal counter and flags are reset.

16.4.3 Bit 4: Data Format Select Bit (ALS)

The ALS bit determines whether the salve address is recognized. When the ALS bit is set to 0, an addressing format is selected and a address data is recognized. Only if the comparison is matched between the slave address stored into the S0D0 register and the received address data or if the general call is received, the data is transferred. When the ALS bit is set to 1, the free data format is selected and the slave address is not recognized.

16.4.4 Bit 6: I2C bus Interface Reset Bit (IHR)

The IHR bit is used to reset the I2C bus interface circuit when the error communication occurs.

When the ES0 bit in the S1D0 register is set to 1 (I2C bus interface is enabled), the hardware is reset by writing 1 to the IHR bit. Flags are processed as follows:

1)The bits in the S10 register are set as MST = 0, TRX = 0, PIN to 1, BB = 0, AL = 0, AAS = 0, and ADR0 = 0

2)The TOF bit in the S4D0 register is set to 0 (time-out detection flag is not detected)

3)The internal counter and flags are reset.

The I2C bus interface circuit is reset after 2.5 V_{IIC} cycles or less, and the IHR bit becomes 0 automatically by writing 1 to the IHR bit. **Figure 16.10** shows the reset timing.

17.10 Reception and Transmission

Configuration of CAN Reception and Transmission Mode

Table 17.3 shows configuration of CAN reception and transmission mode.

Table 17.3 Configuration of CAN Reception and Transmission Mode

TrmReq	RecReq	Remote	RspLock	Communication mode of the slot
0	0	–	–	Communication environment configuration mode: configure the communication mode of the slot.
0	1	0	0	Configured as a reception slot for a data frame.
1	0	1	0	Configured as a transmission slot for a remote frame. (At this time the RemActive = 1.) After completion of transmission, this functions as a reception slot for a data frame. (At this time the RemActive = 0.) However, when an ID that matches on the CAN bus is detected before remote frame transmission, this immediately functions as a reception slot for a data frame.
1	0	0	0	Configured as a transmission slot for a data frame.
0	1	1	1/0	Configured as a reception slot for a remote frame. (At this time the RemActive = 1.) After completion of reception, this functions as a transmission slot for a data frame. (At this time the RemActive = 0.) However, transmission does not start as long as RspLock bit remains 1; thus no automatic response. Response (transmission) starts when the RspLock bit is set to 0.

TrmReq, RecReq, Remote, RspLock, RemActive, RspLock: Bits in the C0MCTLj register (j = 0 to 15)

When configuring a slot as a reception slot, note the following points.

(1) Before configuring a slot as a reception slot, be sure to set the C0MCTLj register (j = 0 to 15) to 00_{16}.

(2) A received message is stored in a slot that matches the condition first according to the result of reception mode configuration and acceptance filtering operation. Upon deciding in which slot to store, the smaller the number of the slot is, the higher priority it has.

(3) In normal CAN operating mode, when a CAN module transmits a message of which ID matches, the CAN module never receives the transmitted data. In loop back mode, however, the CAN module receives back the transmitted data. In this case, the module does not return ACK.

When configuring a slot as a transmission slot, note the following points.

(1) Before configuring a slot as a transmission slot, be sure to set the C0MCTLj registers to 00_{16}.

(2) Set the TrmReq bit in the C0MCTLj register to 0 (not transmission slot) before rewriting a transmission slot.

(3) A transmission slot should not be rewritten when the TrmActive bit in the C0MCTLj register is 1 (transmitting).
If it is rewritten, an undefined data will be transmitted.

Pull-up Control Register 0 (1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
PUR0	$03FC_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
PU00	$P0_0$ to $P0_3$ pull-up	0: Not pulled up 1: Pulled up (1)	RW
PU01	$P0_4$ to $P0_7$ pull-up		RW
PU02	$P1_0$ to $P1_3$ pull-up		RW
PU03	$P1_4$ to $P1_7$ pull-up		RW
PU04	$P2_0$ to $P2_3$ pull-up		RW
PU05	$P2_4$ to $P2_7$ pull-up		RW
PU06	$P3_0$ to $P3_3$ pull-up		RW
PU07	$P3_4$ to $P3_7$ pull-up		RW

NOTE:
1. The pin for which this bit is 1 (pulled up) and the direction bit is 0 (input mode) is pulled up.

Pull-up Control Register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
PUR1	$03FD_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
—— (b3-b0)	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—
PU14	$P6_0$ to $P6_3$ pull-up	0: Not pulled high 1: Pulled high (1)	RW
PU15	$P6_4$ to $P6_7$ pull-up		RW
PU16	$P7_0$ to $P7_3$ pull-up		RW
PU17	$P7_4$ to $P7_7$ pull-up		RW

NOTE:
1. The pin for which this bit is 1 (pulled up) and the direction bit is 0 (input mode) is pulled up.

Pull-up Control Register 2

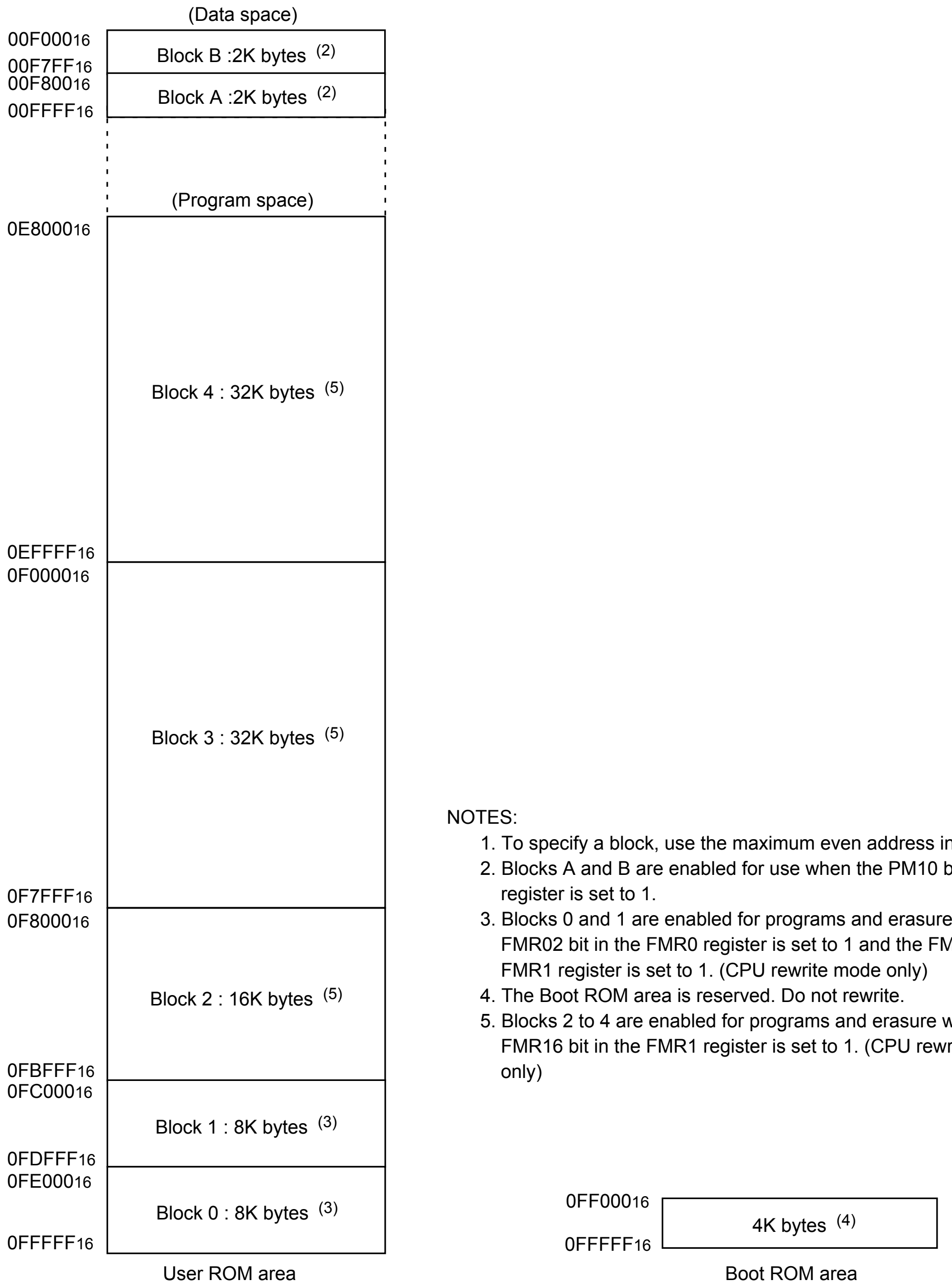
b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
PUR2	$03FE_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
PU20	$P8_0$ to $P8_3$ pull-up	0: Not pulled up 1: Pulled up (1)	RW
PU21	$P8_4$ to $P8_7$ pull-up		RW
PU22	$P9_0$ to $P9_3$ pull-up		RW
PU23	$P9_5$ to $P9_7$ pull-up		RW
PU24	$P10_0$ to $P10_3$ pull-up		RW
PU25	$P10_4$ to $P10_7$ pull-up		RW
—— (b7-b6)	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. The pin for which this bit is 1 (pulled up) and the direction bit is 0 (input mode) is pulled up.

Figure 19.8 PUR0 to PUR2 Registers



NOTES:
1. To specify a block, use the maximum even address in the block.
2. Blocks A and B are enabled for use when the PM10 bit in the PM1 register is set to 1.
3. Blocks 0 and 1 are enabled for programs and erasure when the FMR02 bit in the FMR0 register is set to 1 and the FMR16 bit in the FMR1 register is set to 1. (CPU rewrite mode only)
4. The Boot ROM area is reserved. Do not rewrite.
5. Blocks 2 to 4 are enabled for programs and erasure when the FMR16 bit in the FMR1 register is set to 1. (CPU rewrite mode only)

Figure 20.2 Flash Memory Block Diagram (ROM capacity 96 Kbytes)

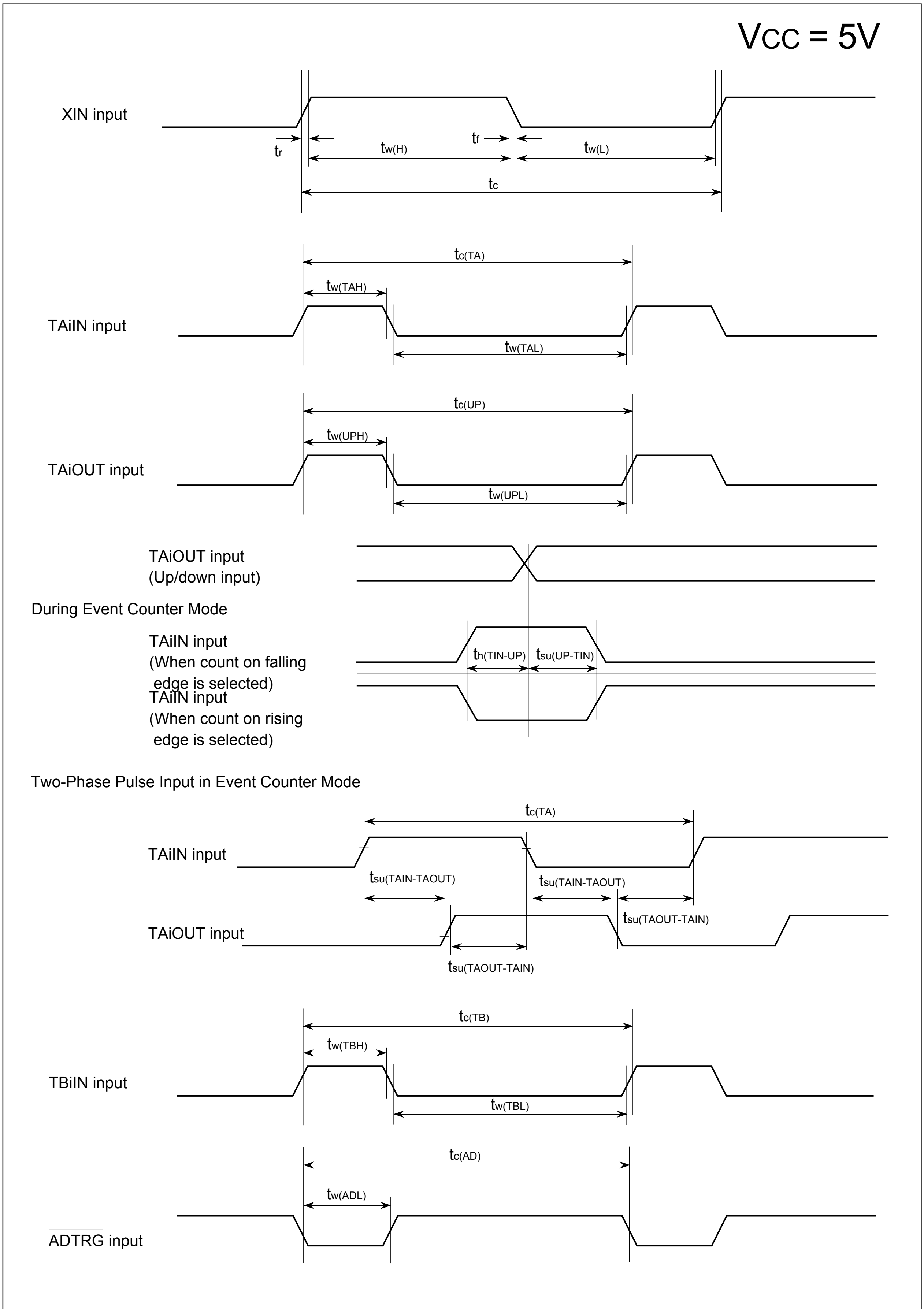


Figure 21.1 Timing Diagram (1)

$V_{CC} = 3V$

Timing Requirements

(V_{CC} = 3V, V_{SS} = 0V, at Topr = – 20 to 85°C / – 40 to 85°C unless otherwise specified)

Table 21.26 External Clock Input (X_{IN} input)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
t_r	External clock rise time		18	ns
t_f	External clock fall time		18	ns

Table 21.81 Flash Memory Version Electrical Characteristics [1] for 100/1000 E/W cycle products [Program Space and Data Space in U3; Program Space in U7]

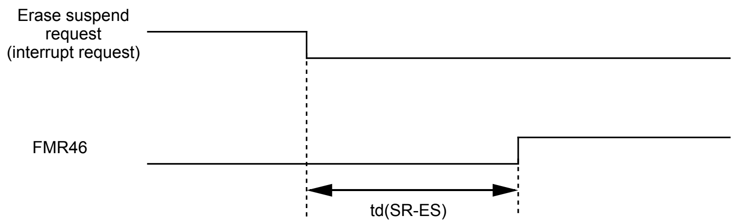
Symbol	Parameter		Standard Min.	Standard Typ.[2]	Standard Max.	Unit
-	Program and Erase Endurance[3]		100/1000[4, 11]			cycles
-	Word Program Time (V_{CC} = 5.0 V, Topr = 25° C)			75	600	µs
-	Block Erase Time (V_{CC} = 5.0 V, Topr = 25° C)	2-Kbyte Block		0.2	9	s
		8-Kbyte Block		0.4	9	s
		16-Kbyte Block		0.7	9	s
		32-Kbyte Block		1.2	9	s
td(SR-ES)	Duration between Suspend Request and Erase Suspend				8	ms
t_{PS}	Wait Time to Stabilize Flash Memory Circuit				15	µs
-	Data Hold Time [5]		20			years

Table 21.82 Flash Memory Version Electrical Characteristics [6] for 10000 E/W cycle products [Data Space in U7 [7]]

Symbol	Parameter	Standard Min.	Standard Typ.[2]	Standard Max.	Unit
-	Program and Erase Endurance[3, 8, 9]	10000[4, 10]			cycles
-	Word Program Time (V_{CC} = 5.0 V, Topr = 25° C)		100		µs
-	Block Erase Time (V_{CC} = 5.0V, Topr = 25° C) (2-Kbyte block)		0.3		s
td(SR-ES)	Duration between Suspend Request and Erase Suspend			8	ms
t_{PS}	Wait Time to Stabilize Flash Memory Circuit			15	µs
-	Data Hold Time [5]	20			years

NOTES:

1. Referenced to VCC = 4.2 to 5.5 V at Topr = 0 to 60° C (program space)/ Topr = -40 to 125° C(data space), unless otherwise specified.
2. VCC = 5.0 V; TOPR = 25° C
3. Program and erase endurance is defined as number of program-erase cycles per block. If program and erase endurance is *n* cycle (*n* = 100, 1000, 10000), each block can be erased and programmed *n* cycles. For example, if a 2-Kbyte block A is erased after programming one-word data to each address 1,024 times, this counts as one program and erase endurance. Data cannot be programmed to the same address more than once without erasing the block. (rewrite prohibited).
4. Number of E/W cycles for which operation is guranteed (1 to minimum value are guranteed).
5. Topr = 55° C
6. Referenced to VCC = 4.2 to 5.5 V at Topr = -40 to 125° C unless otherwise specified.
7. **Table 21.82** applies for data space in U7 when program and erase endurance is more than 1,000 cycles. Otherwise, use **Table 21.81**.
8. To reduce the number of program and erase endurance when working with systems requiring numerous rewrites, write to unused word addresses within the block instead of rewrite. Erase block only after all possible addresses are used. For example, an 8-word program can be written 128 times maximum before erase becomes necessary. Maintaining an equal number of times erasure between block A and block B will also improve efficiency. It is recommended to track the total number of erasure performed per block and to limit the number of erasure.
9. If an erase error is generated during block erase, execute the clear status register command and block erase command at least 3 times until an erase error is not generated.
10. When executing more than 100 times rewrites, set one wait state per block access by setting the FMR17 bit in the FMR1 register to 1 (wait state). When accessing to all other blocks and internal RAM, wait state can be set by the PM17 bit, regardless of the FMR17 bit setting value.
11. The program and erase endurance is 100 cycles for program space and data space in U3; 1,000 cycles for program space in U7.
12. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for further details on the E/W failure rate.



22.4 Interrupts

22.4.1 Reading Address 00000_{16}

Do not read the address 00000_{16} in a program. When a maskable interrupt request is accepted, the CPU reads interrupt information (interrupt number and interrupt request priority level) from the address 00000_{16} during the interrupt sequence. At this time, the IR bit for the accepted interrupt is cleared to 0.
If the address 00000_{16} is read in a program, the IR bit for the interrupt which has the highest priority among the enabled interrupts is cleared to 0. This causes a problem that the interrupt is canceled, or an unexpected interrupt request is generated.

22.4.2 Setting the SP

Set any value in the SP(USP, ISP) before accepting an interrupt. The SP(USP, ISP) is cleared to 0000_{16} after reset. Therefore, if an interrupt is accepted before setting any value in the SP(USP, ISP), the program may go out of control.

22.4.3 $\overline{\text{NMI}}$ Interrupt

1. The $\overline{\text{NMI}}$ interrupt is invalid after reset. The $\overline{\text{NMI}}$ interrupt becomes effective by setting the PM24 bit in the PM2 register to “1”. Set the PM24 bit to "1" when a high-level signal ("H") is applied to the $\overline{\text{NMI}}$ pin. If the PM24 bit is set to "1" when a low-level signal ("L") is applied, $\overline{\text{NMI}}$ interrupt is generated. Once $\overline{\text{NMI}}$ interrupt is enabled, it will not be disabled unless a reset is applied.
2. The input level of the $\overline{\text{NMI}}$ pin can be read by accessing the P8_5 bit in the P8 register.
3. When selecting $\overline{\text{NMI}}$ function, stop mode cannot be entered into while input on the $\overline{\text{NMI}}$ pin is low. This is because while input on the $\overline{\text{NMI}}$ pin is low the CM1 register’s CM10 bit is fixed to 0.
4. When selecting $\overline{\text{NMI}}$ function, do not go to wait mode while input on the $\overline{\text{NMI}}$ pin is low. This is because when input on the $\overline{\text{NMI}}$ pin goes low, the CPU stops but CPU clock remains active; therefore, the current consumption in the chip does not drop. In this case, normal condition is restored by an interrupt generated thereafter.
5. When selecting $\overline{\text{NMI}}$ function, the low and high level durations of the input signal to the $\overline{\text{NMI}}$ pin must each be 2 CPU clock cycles + 300 ns or more.
6. When using the $\overline{\text{NMI}}$ interrupt for exiting stop mode, set the NDDR register to FF_{16} (disable digital debounce filter) before entering stop mode.

22.4.4 Changing the Interrupt Generate Factor

If the interrupt generate factor is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to 1 (interrupt requested). If you changed the interrupt generate factor for an interrupt that needs to be used, be sure to clear the IR bit for that interrupt to 0 (interrupt not requested).
“Changing the interrupt generate factor” referred to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the generate factor, polarity or timing of an interrupt, be sure to clear the IR bit for that interrupt to 0 (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.
Figure 22.2 shows the procedure for changing the interrupt generate factor.

Rev.	Date	Description: Page	Description: Summary
		388	• **Table 21.41 Recommended Operating Conditions** VIH and VIL are modified
		389	• **Table 21.42 A/D Conversion Characeristics** tSAMP deleted, note 4 added
		390	• **Table 21.43 Flash Memory Version Electrical Characteristics:** Standard values of Program and Erase Endrance cycle modified, tps added
			• **Table 21.44 Flash Memory Version Electrical Characteristics:** tps added, data hold time added, note 1, 3, 8 modified, note 11 and 12 added
		391	• **Table 21.45 Power Supply Circuit from Timing CharacteristicsL** Note 2 & 3 are deleted, td(S-R) and td(E-A) are deleted, figure modified
		393	• **Table 21.47 Electrical Characteristics(2)** Note 4 is added
		401	• **Table 21.63 Electrical Characteristics(2)** Note 4 is added
			Precautions
		422	• **22.2.1 PLL Frequency Synthesizer** modified
		423	• **22.2.2 Power Control** Subsection sequence modified, 2., 3. and 4. information modified
		425	• **22.4.3 $\overline{\text{NMI}}$ Interrupt** 2. information partially deleted, 6. information added
		426	• **22.4.5 $\overline{\text{INT}}$ Interrupt** 3. information added
		427	• **22.4.6 Rewrite the Interrupt Control Register** Example 1 is modified
		431	• **22.6.1.3 Timer A (One-shot Timer Mode)** 6. information added
		434	• **22.6.3 Three-phase Motor Control Timer Function** newly added
		435	• **22.7.1 Rewrite the G1 IR Register description modified**
			• **Figure 22.3 IC/OC Interrupt Flow Chart** newly added
		436	• **22.7.2 Rewrite the ICOCiIC Register** newly added
			• **22.7.3 Waveform Generating Function** newly added
			• **22.7.4 IC/OC Base Timer Interrupt** newly added
		438	• **22.8.2.1 Special Mode (I^2C bus Mode)** added
			• **22.8.2.3 SI/O3, SI/O4** added
		441	• **22.10 Multi-master I^2C bus Interface** added
		445	• **22.12 Programmable I/O Ports** 2. and 3. information modified
		447	• **20.14 Mask ROM Version** is added
		448	• **22.15.1 Functions to Inhibit Rewriting Flash Memory Rewrite** modified
			• **22.15.2 Stop Mode** modified
			• **22.16.4 Low Power Disspation Mode, On-chip Oscillator Low Power Dissipation Mode** modified
			• **22.15.7 Operating Speed** modified
		449	• **22.15.9 Interrupts** modified
			• **22.15.13 Regarding Programming/Erasure Times and Execution Time** modified
		450	• **22.15.14 Definition of Programming/Erasure Times** added
			• **22.15.15 Flash Memory version Electrical Characteristics 10,000 E/W cycle**