# Renesas Electronics America Inc - M30291FATHP#U3AAE9 Datasheet

**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 | - |
| Core Size | - |
| Speed | - |
| Connectivity | - |
| Peripherals | - |
| Number of I/O | - |
| Program Memory Size | - |
| Program Memory Type | - |
| EEPROM Size | - |
| RAM Size | - |
| Voltage - Supply (Vcc/Vdd) | - |
| Data Converters | - |
| Oscillator Type | - |
| Operating Temperature | - |
| Mounting Type | - |
| Package / Case | - |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/m30291fathp-u3aae9 |

**Table 1.14 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 |
| CAN | CRX | I | Input pin for the CAN communication function |
| | CTX | O | Output pin for the CAN communication function |
| I/O Ports | $P0_0$ to $P0_3$<br>$P1_5$ to $P1_7$<br>$P2_0$ to $P2_7$<br>$P3_0$ to $P3_3$<br>$P6_0$ to $P6_7$<br>$P7_0$ to $P7_7$<br>$P8_0$ to $P8_7$<br>$P9_0$ to $P9_3$<br>$P10_0$ 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

## 7.7 System Clock Protective Function

When the main clock is selected for the CPU clock source, this function protects the clock from modifications in order to prevent the CPU clock from becoming halted by run-away.

If the PM21 bit in the PM2 register is set to 1 (clock modification disabled), the following bits are protected against writes:

- Bits CM02, CM05, and CM07 in CM0 register
- Bits CM10 and CM11 in CM1 register
- CM20 bit in CM2 register
- All bits in the PLC0 register

Before the system clock protective function can be used, the following register settings must be made while the CM05 bit in the CM0 register is 0 (main clock oscillating) and CM07 bit is 0 (main clock selected for the CPU clock source):

(1) Set the PRC1 bit in the PRCR register to 1 (enable writes to PM2 register).

(2) Set the PM21 bit in the PM2 register to 1 (disable clock modification).

(3) Set the PRC1 bit in the PRCR register to 0 (disable writes to PM2 register).

Do not execute the WAIT instruction when the PM21 bit is 1.

## 7.8 Oscillation Stop and Re-oscillation Detect Function

The oscillation stop and re-oscillation detect function detects the re-oscillation after stop of main clock oscillation circuit. When the oscillation stop and re-oscillation detection occurs, the oscillation stop detect function is reset or oscillation stop and re-oscillation detection interrupt is generated, depending on the CM27 bit set in the CM2 register. The oscillation stop detect function is enabled or disabled by the CM20 bit in the CM2 register. **Table 7.8** lists a specification overview of the oscillation stop and re-oscillation detect function.

**Table 7.8 Specification Overview of Oscillation Stop and Re-oscillation Detect Function**

| Item | Specification |
|---|---|
| Oscillation stop detectable clock and frequency bandwidth | $f(X_{IN}) \geq 2$ MHz |
| Enabling condition for oscillation stop, re-oscillation detection function | Set CM20 bit to 1(enable) |
| Operation at oscillation stop, re-oscillation detection | •Reset occurs (when CM27 bit =0)<br>•Oscillation stop, re-oscillation detection interrupt occurs(when CM27 bit =1) |

## 9.2 Interrupts and Interrupt Vector

One interrupt vector consists of 4 bytes. Set the start address of each interrupt routine in the respective interrupt vectors. When an interrupt request is accepted, the CPU branches to the address set in the corresponding interrupt vector. **Figure 9.2** shows the interrupt vector.

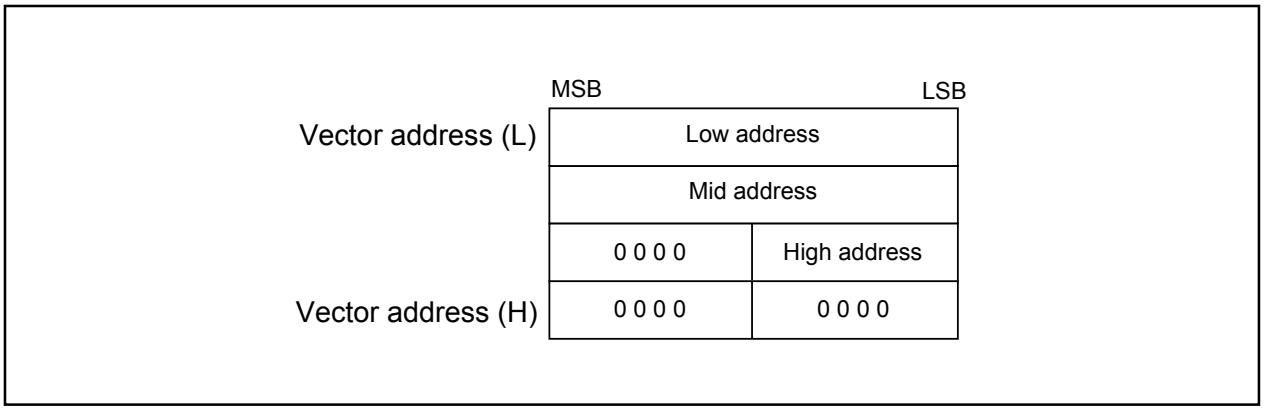



**Figure 9.2 Interrupt Vector**

### 9.2.1 Fixed Vector Tables

The fixed vector tables are allocated to the addresses from FFFDC16 to FFFFF16. **Table 9.1** lists the fixed vector tables. In the flash memory version of MCU, the vector addresses (H) of fixed vectors are used by the ID code check function. For details, refer to the section "flash memory rewrite disabling function".

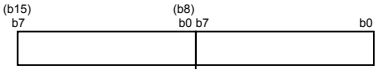
**Table 9.1 Fixed Vector Tables**

| Interrupt source | Vector table addresses Address (L) to address (H) | Remarks | Reference |
|---|---|---|---|
| Undefined instruction | FFFDC16 to FFFDF16 | Interrupt on UND instruction | M16C/60, M16C/20 serise software maual |
| Overflow | FFFE016 to FFFE316 | Interrupt on INTO instruction | |
| BRK instruction | FFFE416 to FFFE716 | If the contents of address FFFE716 is FF16, program execution starts from the address shown by the vector in the relocatable vector table | |
| Address match | FFFE816 to FFFEB16 | | Address match interrupt |
| Single step (1) | FFFEC16 to FFFEF16 | | |
| Watchdog timer Oscillation stop and re-oscillation detection, low voltage detection | FFFF016 to FFFF316 | | Watchdog timer, clock generating circuit, voltage detection circuit |
| $\overline{\text{DBC}}$ (1) | FFFF416 to FFFF716 | | |
| $\overline{\text{NMI}}$ | FFFF816 to FFFFB16 | | $\overline{\text{NMI}}$ interrupt |
| Reset(2) | FFFFC16 to FFFFF16 | | Reset |

NOTE:

1. Do not normally use this interrupt because it is provided exclusively for use by development tools.

Timer Ai Register (i= 0 to 4) (1)

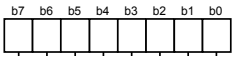



| Symbol | Address | After Reset |
|---|---|---|
| TA0 | 038716, 038616 | Undefined |
| TA1 | 038916, 038816 | Undefined |
| TA2 | 038B16, 038A16 | Undefined |
| TA3 | 038D16, 038C16 | Undefined |
| TA4 | 038F16, 038E16 | Undefined |

| Mode | Function | Setting Range | RW |
|---|---|---|---|
| Timer mode | Divide the count source by n + 1 where n = set value | 000016 to FFFF16 | RW |
| Event counter mode | Divide the count source by FFFF16 – n + 1 where n = set value when counting up or by n + 1 when counting down(5) | 000016 to FFFF16 | RW |
| One-shot timer mode | Divide the count source by n where n = set value and cause the timer to stop | 000016 to FFFF16 (2, 4) | WO |
| Pulse width modulation mode (16-bit PWM) | Modify the pulse width as follows:<br>PWM period: (216 – 1) / fj<br>High level PWM pulse width: n / fj where n = set value, fj = count source frequency | 000016 to FFFE16 (3, 4) | WO |
| Pulse width modulation mode (8-bit PWM) | Modify the pulse width as follows:<br>PWM period: (28 – 1) x (m + 1)/ fj<br>High level PWM pulse width: (m + 1)n / fj where n = high-order address set value, m = low-order address set value, fj = count source frequency | 0016 to FE16 (High-order address)<br>0016 to FF16 (Low-order address) (3, 4) | WO |

NOTES:
1. The register must be accessed in 16 bit units.
2. If the TAi register is set to 000016, the counter does not work and timer Ai interrupt requests are not generated either. Furthermore, if “pulse output” is selected, no pulses are output from the TAiOUT pin.
3. If the TAi register is set to 000016, the pulse width modulator does not work, the output level on the TAiOUT pin remains low, and timer Ai interrupt requests are not generated either. The same applies when the 8 high-order bits of the timer TAi register are set to 000016 while operating as an 8-bit pulse width modulator.
4. Use the MOV instruction to write to the TAi register.
5. The timer counts pulses from an external device or overflows or underflows in other timers.

Count Start Flag




| Symbol | Address | After Reset |
|---|---|---|
| TABSR | 038016 | 0016 |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TA0S | Timer A0 count start flag | 0 : Stops counting<br>1 : Starts counting | RW |
| TA1S | Timer A1 count start flag | | RW |
| TA2S | Timer A2 count start flag | | RW |
| TA3S | Timer A3 count start flag | | RW |
| TA4S | Timer A4 count start flag | | RW |
| TB0S | Timer B0 count start flag | | RW |
| TB1S | Timer B1 count start flag | | RW |
| TB2S | Timer B2 count start flag | | RW |

Up/Down Flag (1)


| Symbol | Address | After Reset |
|---|---|---|
| UDF | 038416 | 0016 |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TA0UD | Timer A0 up/down flag | 0: Down count<br>1: Up count<br>Enabled by setting the MR2 bit in the TAiMR register to 0 (= switching source in UDF register) during event counter mode | RW |
| TA1UD | Timer A1 up/down flag | | RW |
| TA2UD | Timer A2 up/down flag | | RW |
| TA3UD | Timer A3 up/down flag | | RW |
| TA4UD | Timer A4 up/down flag | | RW |
| TA2P | Timer A2 two-phase pulse signal processing select bit | 0: two-phase pulse signal processing disabled<br>1: two-phase pulse signal processing enabled (2, 3) | WO |
| TA3P | Timer A3 two-phase pulse signal processing select bit | | WO |
| TA4P | Timer A4 two-phase pulse signal processing select bit | | WO |

NOTES:
1. Use MOV instruction to write to this register.
2. Make sure the port direction bits for the TA2IN to TA4IN and TA2OUT to TA4OUT pins are set to 0 input mode.
3. When the two-phase pulse signal processing function is not used, set the corresponding bit to 0.

**Figure 12.5 TA0 to TA4 Registers, TABSR Register, and UDF Register**

Timer Ai Mode Register (i=0 to 4)
(When not using two-phase pulse signal processing)

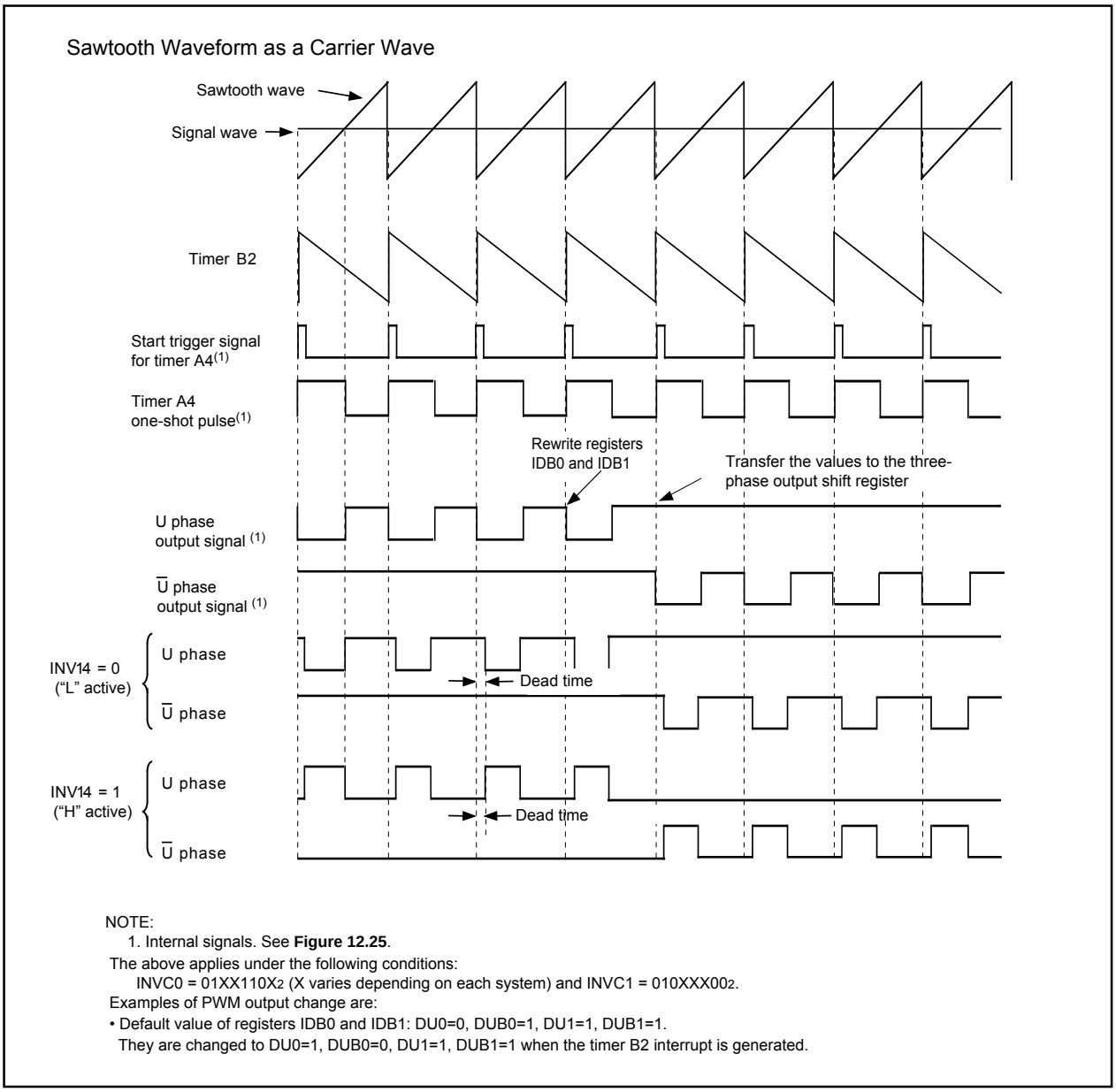
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|
| | | 0 | | | | 0 | 1 |

| Symbol | Address | After Reset |
|---|---|---|
| TA0MR to TA4MR | 039616 to 039A16 | 0016 |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| TMOD0 | Operation mode select bit | b1 b0<br>0 1 : Event counter mode (1) | RW |
| TMOD1 | | | RW |
| MR0 | Pulse output function select bit | 0: Pulse is not output (TAiOUT pin functions as I/O port)<br>1: Pulse is output (TAiOUT pin functions as pulse output pin) | RW |
| MR1 | Count polarityselect bit (2) | 0: Counts external signal's falling edge<br>1: Counts external signal's rising edge | RW |
| MR2 | Up/down switching cause select bit | 0: UDF register<br>1: Input signal to TAiOUT pin(3) | RW |
| MR3 | Set to 0 in event counter mode | | RW |
| TCK0 | Count operation type select bit | 0: Reload type<br>1: Free-run type | RW |
| TCK1 | Can be 0 or 1 when not using two-phase pulse signal processing | | RW |

NOTES:
1. During event counter mode, the count source can be selected using registers ONSF and TRGSR.
2. Effective when bits TAiTGH and TAiTGL in the ONSF or TRGSR register are 002 (TAiIN pin input).
3. Decrement when input on TAiOUT pin is low or increment when input on that pin is high. The port direction bit for TAiOUT pin must be set to 0 (input mode).

**Figure 12.8 TAiMR Register in Event Counter Mode (when not using two-phase pulse signal processing)**

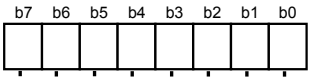



NOTE:
1. Internal signals. See **Figure 12.25**.

The above applies under the following conditions:
INVC0 = 01XX110X2 (X varies depending on each system) and INVC1 = 010XXX002.
Examples of PWM output change are:
• Default value of registers IDB0 and IDB1: DU0=0, DUB0=1, DU1=1, DUB1=1.
They are changed to DU0=1, DUB0=0, DU1=1, DUB1=1 when the timer B2 interrupt is generated.

**Figure 12.34 Sawtooth Wave Modulation Operation**

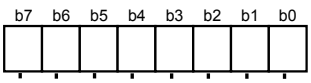
## Interrupt Enable Register 0




| Symbol | Address | After Reset |
|---|---|---|
| G1IE0 | $0331_{16}$ | $00_{16}$ |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| G1IE00 | Interrupt enable 0, CH0 | 0 : IC/OC interrupt 0 request disable<br>1 : IC/OC interrupt 0 request enable | RW |
| G1IE01 | Interrupt enable 0, CH1 | | RW |
| G1IE02 | Interrupt enable 0, CH2 | | RW |
| G1IE03 | Interrupt enable 0, CH3 | | RW |
| G1IE04 | Interrupt enable 0, CH4 | | RW |
| G1IE05 | Interrupt enable 0, CH5 | | RW |
| G1IE06 | Interrupt enable 0, CH6 | | RW |
| G1IE07 | Interrupt enable 0, CH7 | | RW |

## Interrupt Enable Register 1



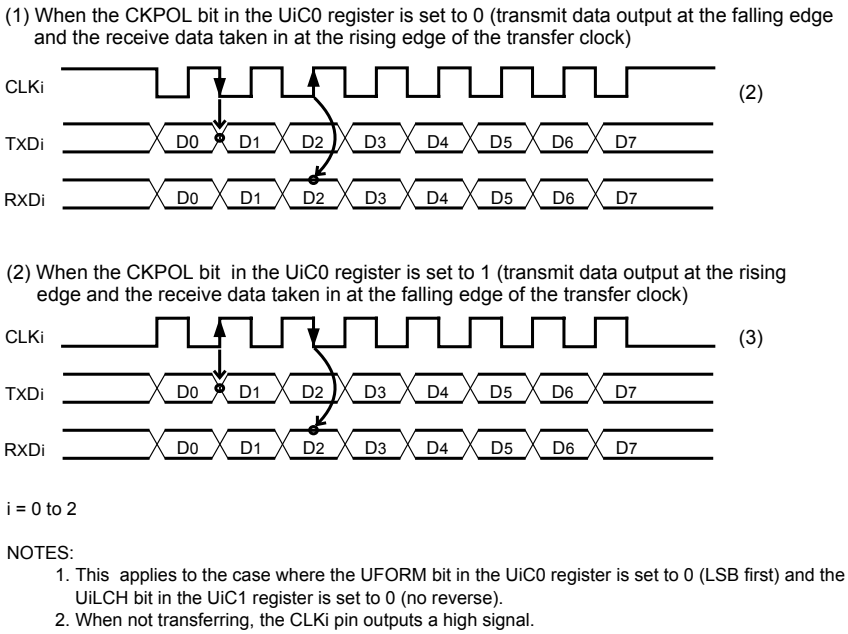

| Symbol | Address | After Reset |
|---|---|---|
| G1IE1 | $0332_{16}$ | $00_{16}$ |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| G1IE10 | Interrupt enable 1, CH0 | 0 : IC/OC interrupt 1 request disable<br>1 : IC/OC interrupt 1 request enable | RW |
| G1IE11 | Interrupt enable 1, CH1 | | RW |
| G1IE12 | Interrupt enable 1, CH2 | | RW |
| G1IE13 | Interrupt enable 1, CH3 | | RW |
| G1IE14 | Interrupt enable 1, CH4 | | RW |
| G1IE15 | Interrupt enable 1, CH5 | | RW |
| G1IE16 | Interrupt enable 1, CH6 | | RW |
| G1IE17 | Interrupt enable 1, CH7 | | RW |

**Figure 13.10 G1IE0 and G1IE1 Registers**

#### 14.1.1.2 CLK Polarity Select Function

Use the CKPOL bit in the UiC0 register (i=0 to 2) to select the transfer clock polarity. **Figure 14.11** shows the polarity of the transfer clock.



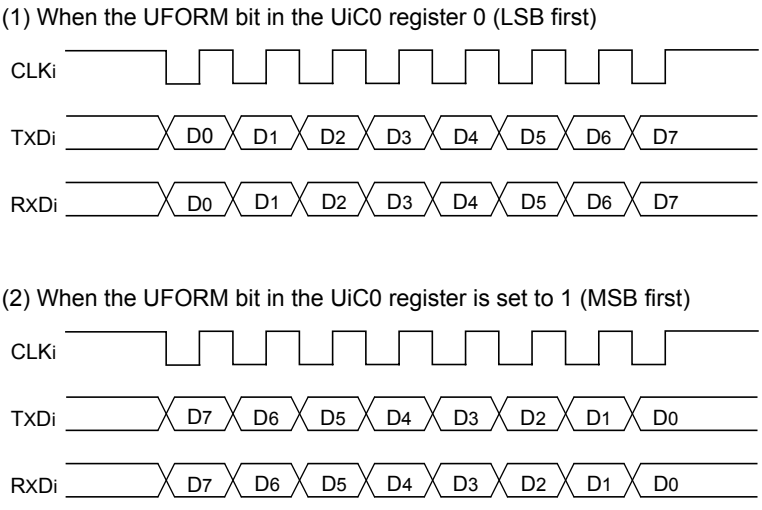

i = 0 to 2

NOTES:
1. This applies to the case where the UFORM bit in the UiC0 register is set to 0 (LSB first) and the UiLCH bit in the UiC1 register is set to 0 (no reverse).
2. When not transferring, the CLKi pin outputs a high signal.

**Figure 14.11 Polarity of transfer clock**

#### 14.1.1.3 LSB First/MSB First Select Function

Use the UFORM bit in the UiC0 register (i=0 to 2) to select the transfer format. **Figure 14.12** shows the transfer format.




i = 0 to 2

NOTE:
1. This applies to the case where the CKPOL bit in the UiC0 register is set to 0 (transmit data output at the falling edge and the receive data taken in at the rising edge of the transfer clock) and the UiLCH bit in the UiC1 register 0 (no reverse).

**Figure 14.12 Transfer format**

#### 14.1.1.4 Continuous receive mode

When the UiRRM bit (i=0 to 2) is set to 1 (continuous receive mode), the TI bit in the UiC1 register is set to 0 (data present in the UiTB register) by reading the UiRB register. In this case, i.e., UiRRM bit is set to 1, do not write dummy data to the UiTB register in a program. The U0RRM and U1RRM bits are the bit 2 and bit 3 in the UCON register, respectively, and the U2RRM bit is the bit 5 in the U2C1 register.

#### 14.1.1.5 Serial data logic switch function (UART2)

When the U2LCH bit in the U2C1 register is set to 1 (reverse), the data written to the U2TB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the U2RB register. **Figure 14.13** shows serial data logic.

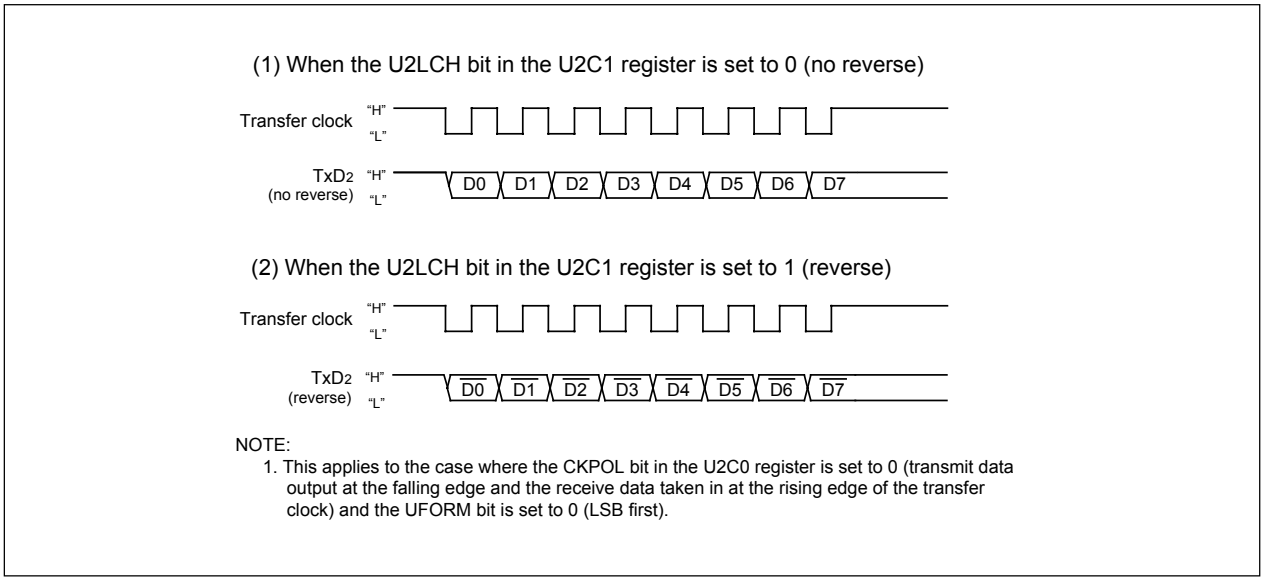



**Figure 14.13 Serial data logic switch timing**

#### 14.1.1.6 Transfer clock output from multiple pins function (UART1)

The CLKMD1 to CLKMD0 bits in the UCON register can choose one from two transfer clock output pins. (See **Figure 14.14**) This function is valid when the internal clock is selected for UART1.

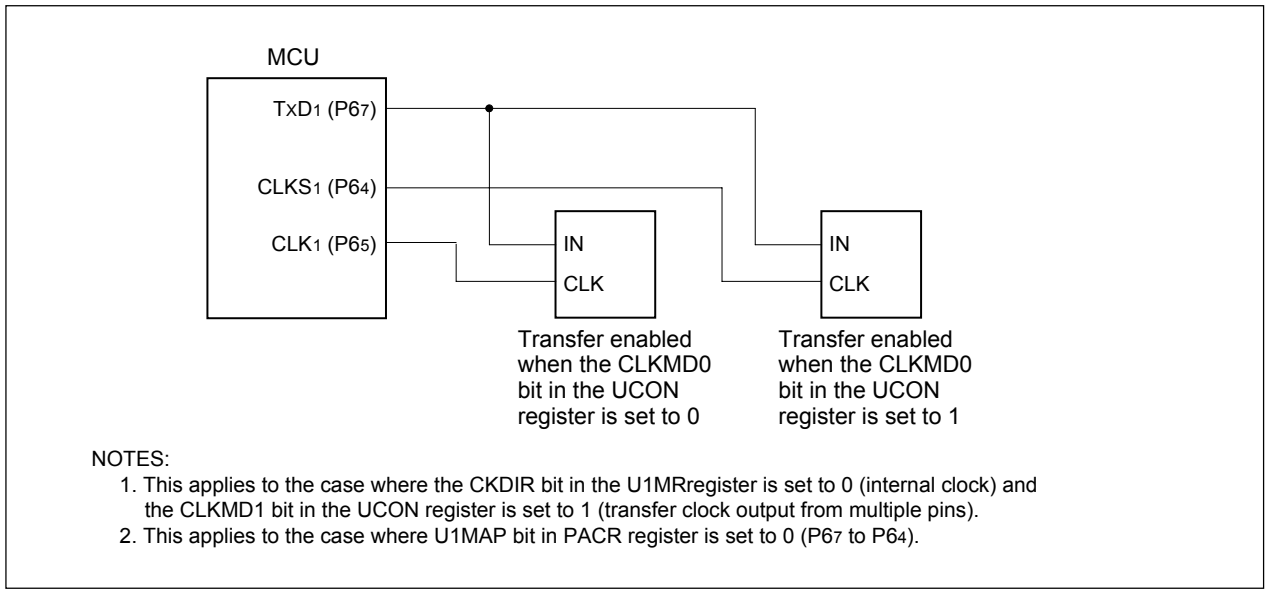



**Figure 14.14 Transfer Clock Output From Multiple Pins**

#### 14.1.3.7 ACK and NACK

If the STSPSEL bit in the U2SMR4 register is set to 0 (start and stop conditions not generated) and the ACKC bit in the U2SMR4 register is set to 1 (ACK data output), the value of the ACKD bit in the U2SMR4 register is output from the $SDA_2$ pin.

If the IICM2 bit is set to 0, a NACK interrupt request is generated if the $SDA_2$ pin remains high at the rising edge of the 9th bit of transmit clock pulse. An ACK interrupt request is generated if the $SDA_2$ pin is low at the rising edge of the 9th bit of transmit clock pulse.

If ACK2 is selected for the cause of DMA1 request, a DMA transfer can be activated by detection of an acknowledge.

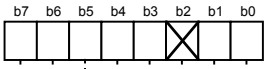
#### 14.1.3.8 Initialization of Transmission/Reception

If a start condition is detected while the STAC bit is set to 1 (UART2 initialization enabled), the serial I/O operates as described below.

- The transmit shift register is initialized, and the content of the U2TB register is transferred to the transmit shift register. In this way, the serial I/O starts sending data synchronously with the next clock pulse applied. However, the UART2 output value does not change state and remains the same as when a start condition was detected until the first bit in the data is output synchronously with the input clock.
- The receive shift register is initialized, and the serial I/O starts receiving data synchronously with the next clock pulse applied.
- The SWC bit is set to 1 ($SCL_2$ wait output enabled). Consequently, the $SCL_2$ pin is pulled low at the falling edge of the ninth clock pulse.

Note that when UART2 transmission/reception is started using this function, the TI does not change state. Note also that when using this function, the selected transfer clock should be an external clock.

A/D Conversion Status Register 0 (1)



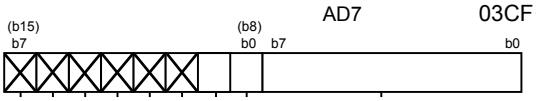

| Symbol | Address | After reset |
|---|---|---|
| ADSTAT0 | 03D316 | 0016 |

| Bit Symbol | Bit Name | Function | RW |
|---|---|---|---|
| ADERR0 | AN1 trigger status flag | 0: AN1 trigger did not occur during AN0 conversion<br>1: AN1 trigger occured during AN0 conversion | RW |
| ADERR1 | Conversion termination flag | 0: Conversion not terminated<br>1: Conversion terminated by Timer B0 underflow | RW |
| (b2) | Nothing is assigned. If necessary, set to 0. When read, its content is 0 | | — |
| ADTCSF | Delayed trigger sweep status flag | 0: Sweep not in progress<br>1: Sweep in progress | RO |
| ADSTT0 | AN0 conversion status flag | 0: AN0 conversion not in progress<br>1: AN0 conversion in progress | RO |
| ADSTT1 | AN1 conversion status flag | 0: AN1 conversion not in progress<br>1: AN1 conversion in progress | RO |
| ADSTRT0 | AN0 conversion completion status flag | 0: AN0 conversion not completed<br>1: AN0 conversion completed | RW |
| ADSTRT1 | AN1 conversion completion status flag | 0: AN1 conversion not completed<br>1: AN1 conversion completed | RW |

NOTE:
1. ADSTAT0 register is valid only when the DTE bit in the ADTRGCON register is set to 1.

A/D Register i (i=0 to 7)

| Symbol | Address | After Reset |
|---|---|---|
| AD0 | 03C116 to 03C016 | Undefined |
| AD1 | 03C316 to 03C216 | Undefined |
| AD2 | 03C516 to 03C416 | Undefined |
| AD3 | 03C716 to 03C616 | Undefined |
| AD4 | 03C916 to 03C816 | Undefined |
| AD5 | 03CB16 to 03CA16 | Undefined |
| AD6 | 03CD16 to 03CC16 | Undefined |
| AD7 | 03CF16 to 03CE16 | Undefined |




| Function | | RW |
|---|---|---|
| When the BITS bit in the ADCON1 register is 1 (10-bit mode) | When the BITS bit in the ADCON1 register is 0 (8-bit mode) | RW |
| Eight low-order bits of A/D conversion result | A/D conversion result | RO |
| Two high-order bits of A/D conversion result | When read, its content is undefined | RO |
| Nothing is assigned. If necessary, set to 0. When read, its content is 0 | | — |

**Figure 15.4 ADSTAT0 Register and AD0 to AD7 Registers**

## 16.1 I2C0 Data Shift Register (S00 register)

The S00 register is an 8-bit data shift register to store a received data and to write a transmit data. When a transmit data is written to the S00 register, the transmit data is synchronized with a SCL clock and the data is transferred from bit 7. Then, every one bit of the data is transmitted, the register's content is shifted for one bit to the left. When the SCL clock and the data is imported into the S00 register from bit 0. Every one bit of the data is imported, the register's content is shifted for one bit to the left. **Figure 16.9** shows the timing to store the receive data to the S00 register.

The S00 register can be written when the ES0 bit in the S1D0 register is set to 1 (I2C0 bus interface enabled). If the S00 register is written when the ES0 bit is set to 1 and the MST bit in the S10 register is set to 1 (master mode), the bit counter is reset and the SCL clock is output. Write to the S00 register when the START condition is generatedor when an "L" signal is applied to the SCL pin. The S00 register can be read anytime regardless of the ES0 bit value.

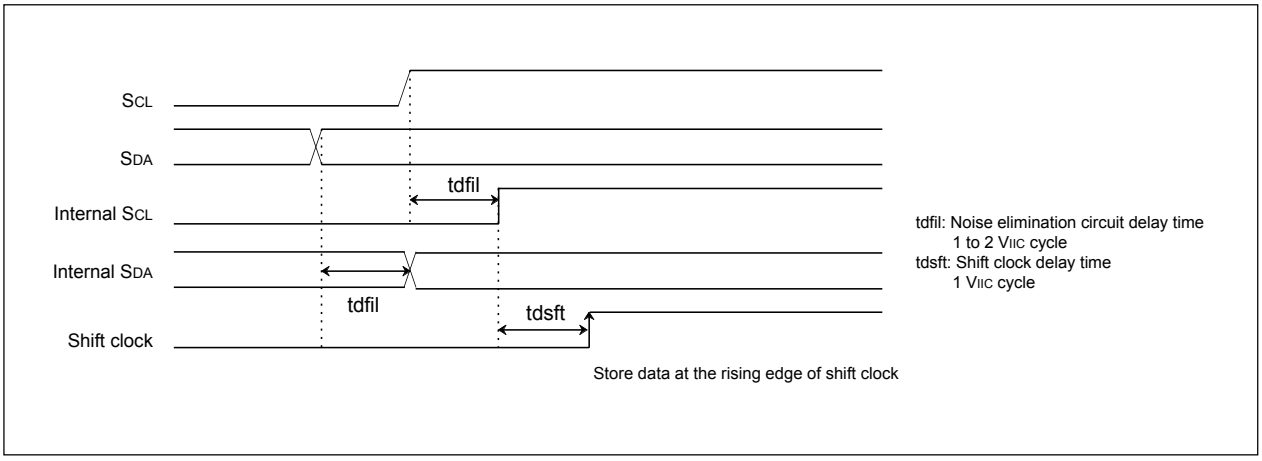



**Figure 16.9 The Receive Data Storing Timing of S00 Register**

## 16.2 I2C0 Address Register (S0D0 register)

The S0D0 register consists of bits SAD6 to SAD0, total of 7. At the addressing is formatted, slave address is detected automatically and the 7-bit received address data is compared with the contents of bits SAD6 to SAD0.

### 16.5.5 Bit 4: I2C bus Interface Interrupt Request Bit (PIN)

The PIN bit generates an I2C bus interface interrupt request signal. Every one byte data is ransferred, the PIN bit is changed from 1 to 0. At the same time, an I2C bus interface interrupt request is generated. The PIN bit is synchronized with the last clock of the internal transfer clock (when ACK-CLK=1, the last clock is the ACK clock: when the ACK-CLK=0, the last clock is the 8th clock) and it becomes 0. The interrupt request is generated on the falling edge of the PIN bit. When the PIN bit is set to 0, the clock applied to SCL maintains "L" and further clock generation is disabled. When the ACK-CLK bit is set to 1 and the WIT bit in the S3D0 register is set to 1 (enable the I2C bus interface interrupt of data receive completion). The PIN bit is synchronized with the last clock and the falling edge of the ACK clock. Then, the PIN bit is set to 0 and I2C bus interface interrupt request is generated. **Figure 16.11** shows the timing of the I2C bus interface interrupt request generation.

The PIN bit is set to 1 in one of the following conditions:

- •When data is written to the S00 register
- •When data is written to the S20 register (when the WIT bit is set to 1 and the internal WAIT flag is set to 1)
- •When the ES0 bit in the S1D0 register is set to 0 (I2C bus interface disabled)
- •When the IHR bit in the S1D0 register is set to 1(reset)

The PIN bit is set to 0 in one of the following conditions:

- •With completion of 1-byte data transmit (including a case when arbitration lost is detected)
- •With completion of 1-byte data receive
- •When the ALS bit in the S1D0 register is set to 0 (addressing format) and slave address is matched or general call address is received successfully in slave receive mode
- •When the ALS bit is set to 1 (free format) and the address data is received successfully in slave receive mode

### 16.5.6 Bit 5: Bus Busy Flag (BB)

The BB flag indicates the operating conditions of the bus system. When the BB flag is set to 0, a bus system is not in use and a START condition can be generated. The BB flag is set and reset based on an input signal of the SCL and SDA pins either in master mode or in slave mode. When the START condition is detected, the BB flag is set to 1. On the other hand, when the STOP condition is detected, the BB flag is set to 0. Bits SSC4 to SSC0 in the S2D0 register decide to detect between the START condition and the STOP condition. When the ES0 bit in the S1D0 register is set to 0 (I2C bus interface disabled) or when the IHR bit in the S1D0 register is set to 1 (reset), the BB flag is set to 0. Refer to **16.9 START Condition Generation Method and 16.11 STOP Condition Generation Method**.

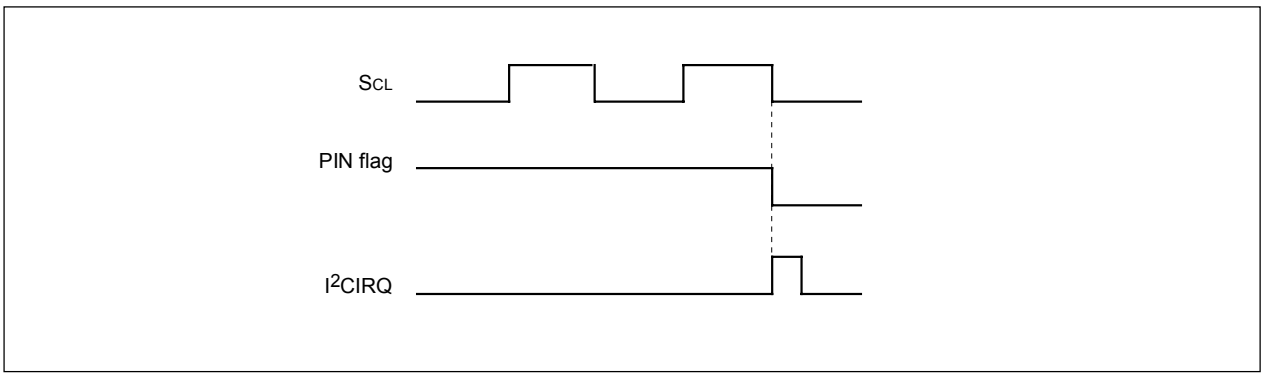
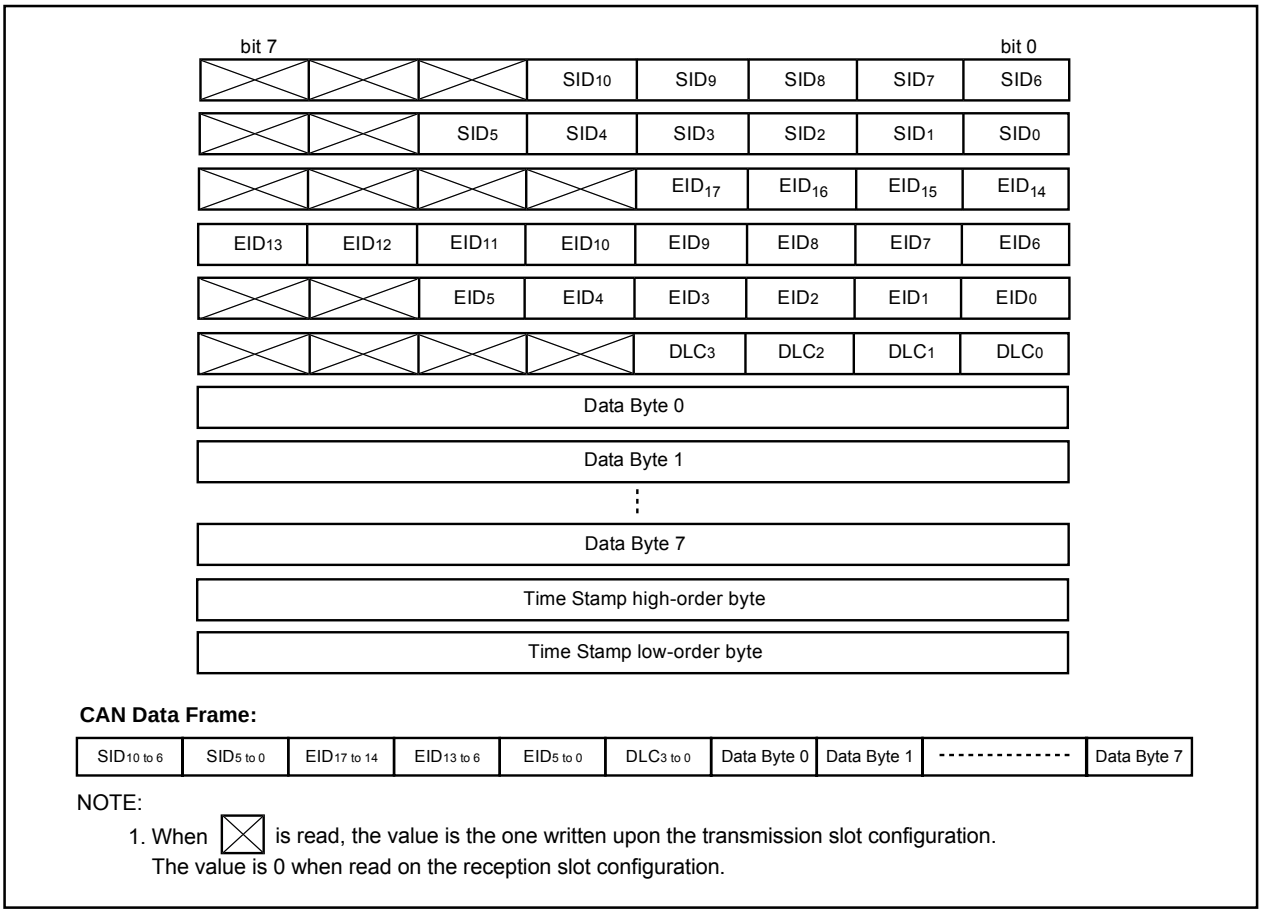


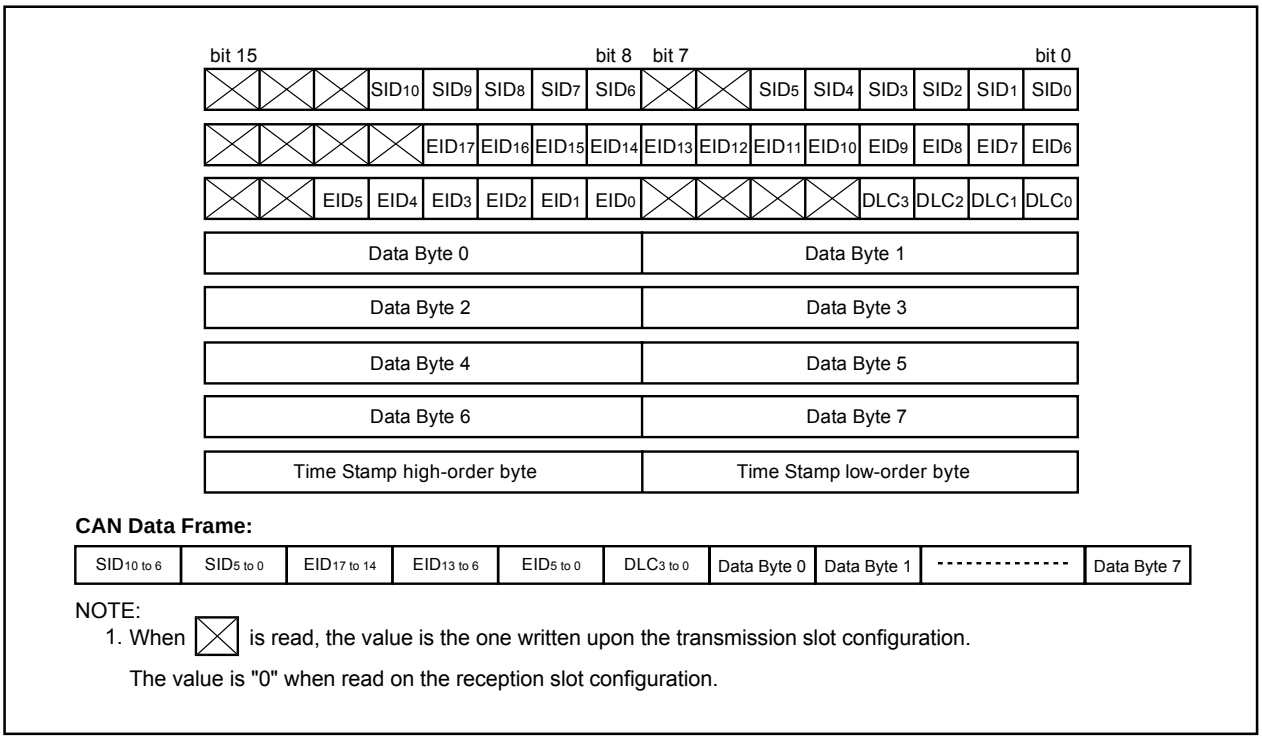

**Figure 16.11 Interrupt request signal generation timing**

**Figures 17.2** and **17.3** show the bit mapping in each slot in byte access and word access. The content of each slot remains unchanged unless transmission or reception of a new message is performed.



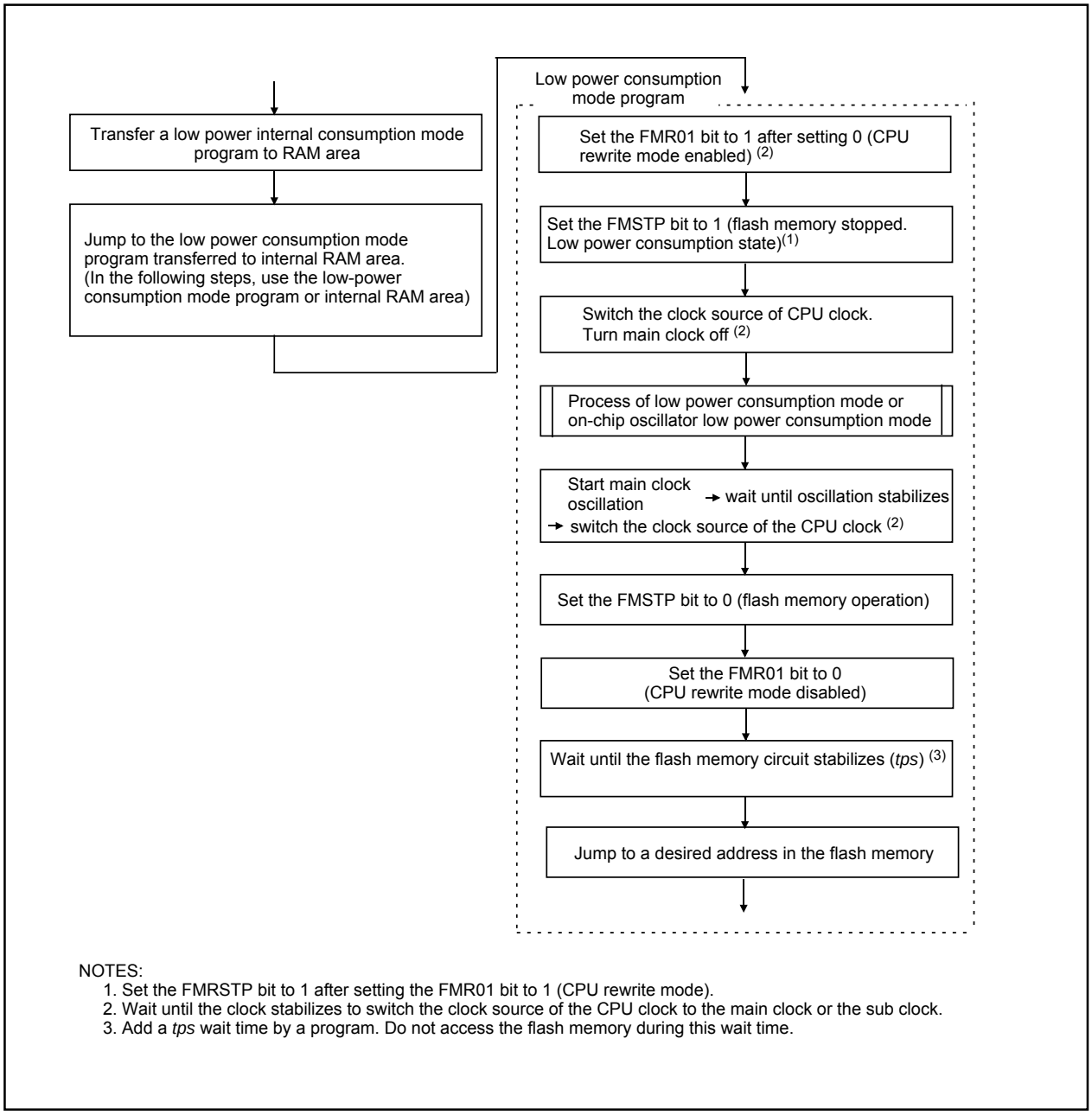

NOTE:
1. When ☒ is read, the value is the one written upon the transmission slot configuration.
The value is 0 when read on the reception slot configuration.

**Figure 17.2 Bit Mapping in Byte Access**




NOTE:
1. When ☒ is read, the value is the one written upon the transmission slot configuration.
The value is "0" when read on the reception slot configuration.

**Figure 17.3 Bit Mapping in Word Access**




NOTES:
1. Set the FMRSTP bit to 1 after setting the FMR01 bit to 1 (CPU rewrite mode).
2. Wait until the clock stabilizes to switch the clock source of the CPU clock to the main clock or the sub clock.
3. Add a *tps* wait time by a program. Do not access the flash memory during this wait time.

**Figure 20.10 Processing Before and After Low Power Dissipation Mode**

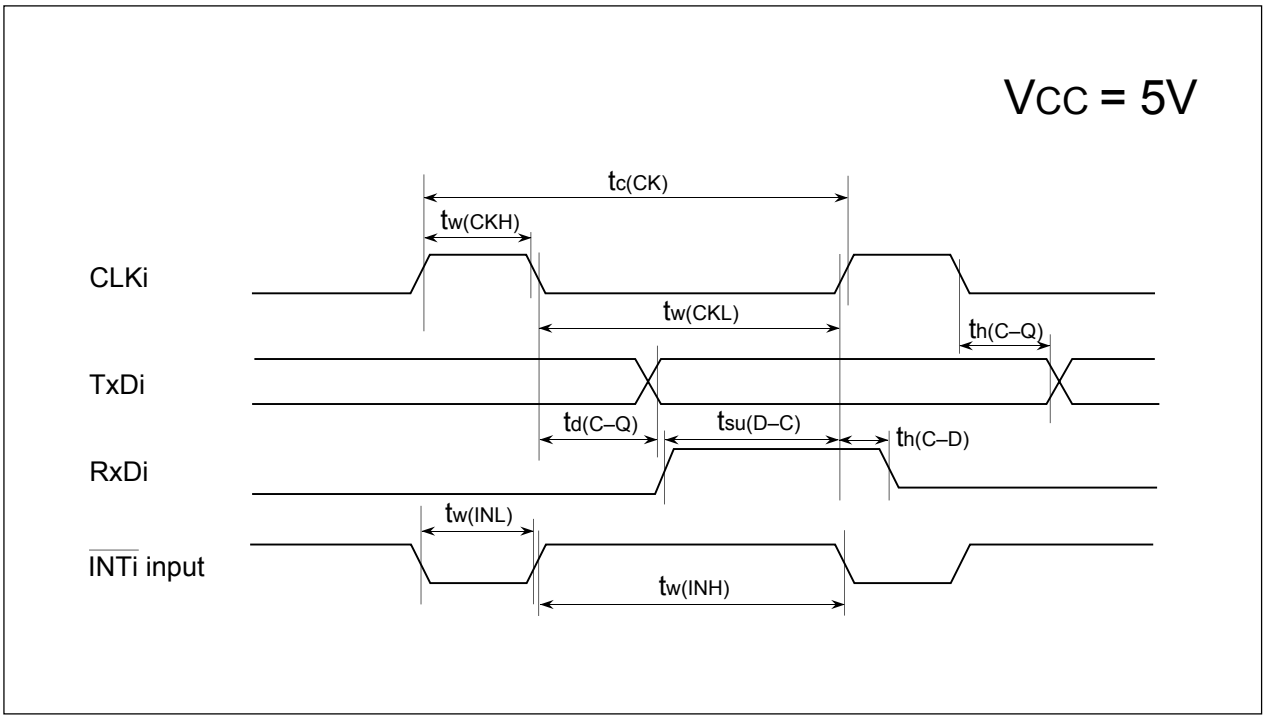



**Figure 21.2 Timing Diagram (2)**

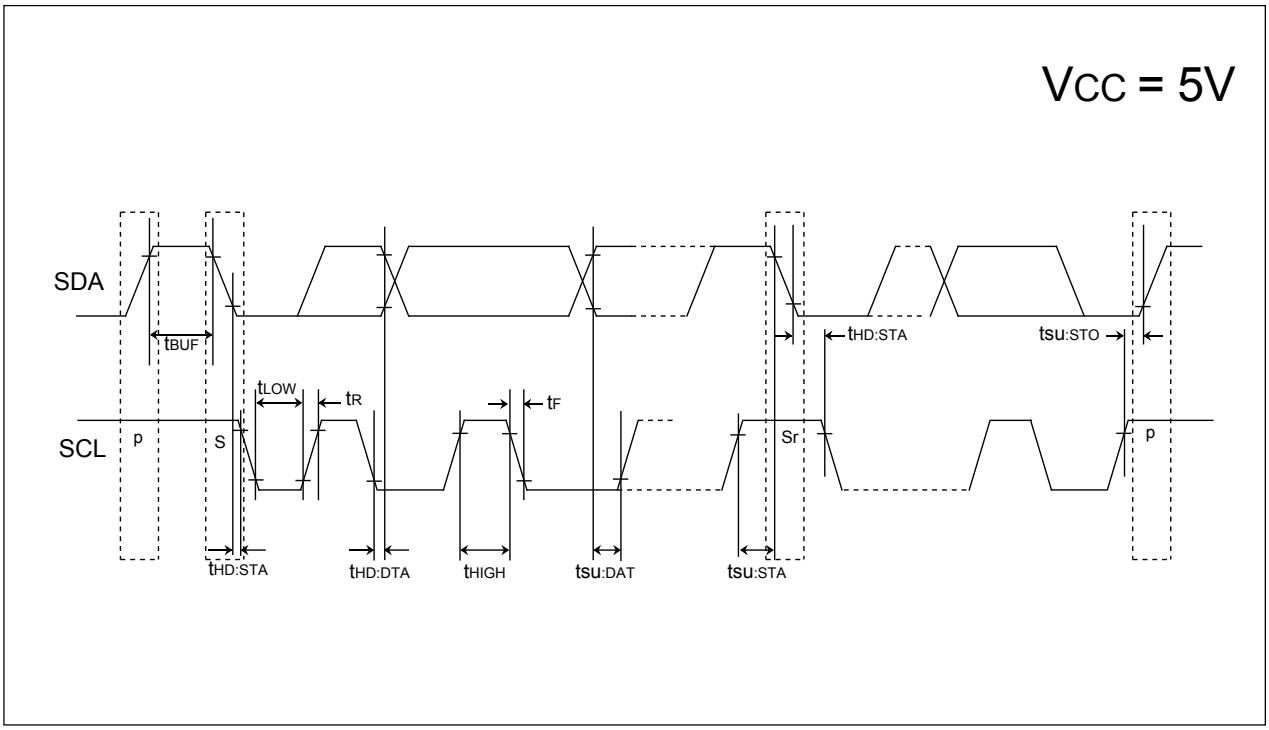



**Figure 21.3 Timing Diagram (3)**

$V_{CC}$ = 5V

### Timing Requirements

**($V_{CC}$ = 5V, $V_{SS}$ = 0V, at Topr = – 40 to 85°C unless otherwise specified)**

**Table 21.61 Multi-master $I^2C$ bus Line**

| Symbol | Parameter | Standard clock mode | | High-speed clock mode | | Unit |
|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | |
| tBUF | Bus free time | 4.7 | | 1.3 | | µs |
| tHD;STA | The hold time in start condition | 4.0 | | 0.6 | | µs |
| tLOW | The hold time in SCL clock 0 status | 4.7 | | 1.3 | | µs |
| tR | SCL, SDA signals' rising time | | 1000 | 20+0.1Cb | 300 | ns |
| tHD;DAT | Data hold time | 0 | | 0 | 0.9 | µs |
| tHIGH | The hold time in SCL clock 1 status | 4.0 | | 0.6 | | µs |
| tF | SCL, SDA signals' falling time | | 300 | 20+0.1Cb | 300 | ns |
| tsu;DAT | Data setup time | 250 | | 100 | | ns |
| tsu;STA | The setup time in restart condition | 4.7 | | 0.6 | | µs |
| tsu;STO | Stop condition setup time | 4.0 | | 0.6 | | µs |

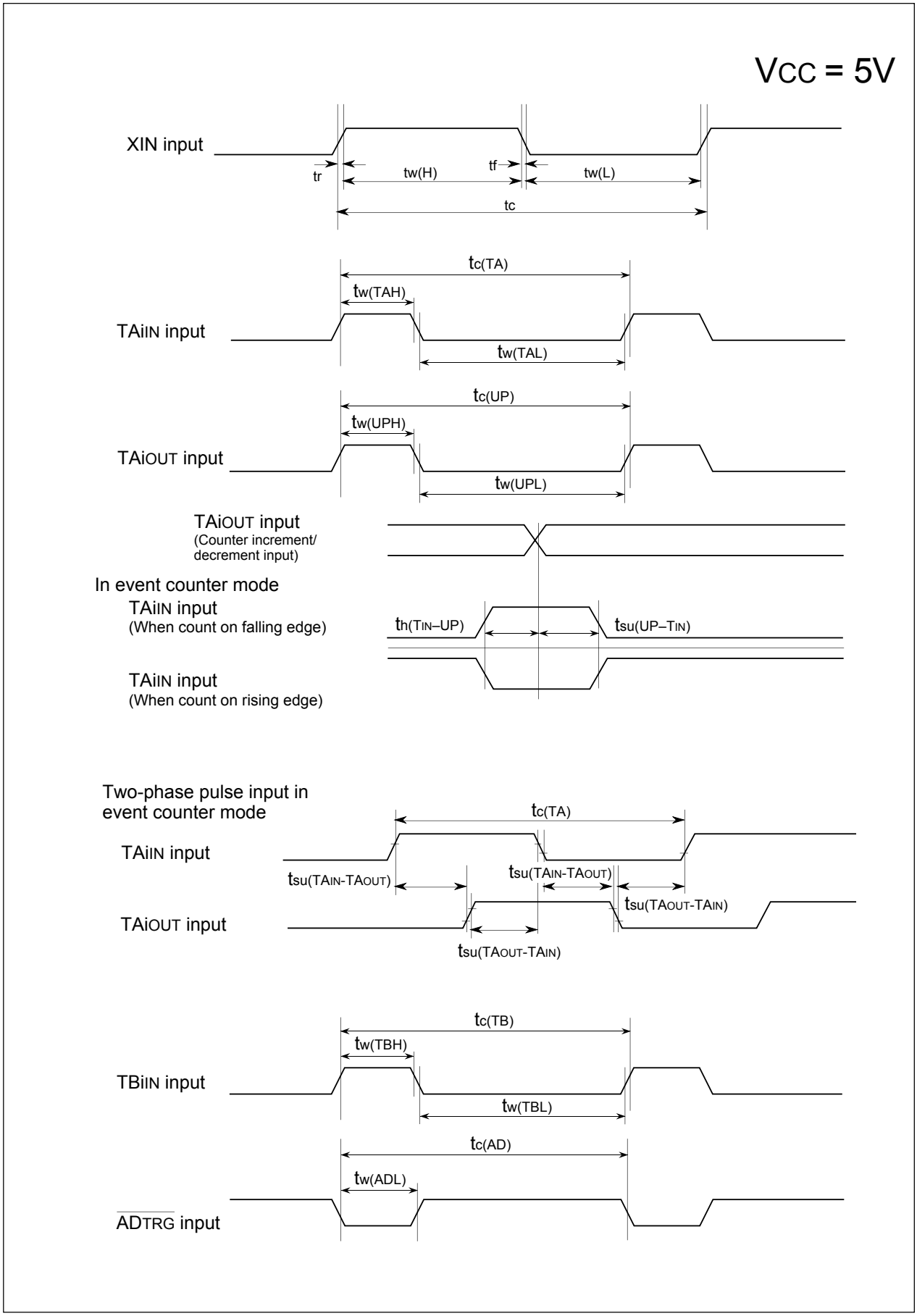



**Figure 21.13 Timing Diagram (1)**

# 22.6 Timers

## 22.6.1 Timer A

### 22.6.1.1 Timer A (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register and the TAi register before setting the TAiS bit in the TABSR register to 1 (count starts).
   Always make sure the TAiMR register is modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the TAi register is read at the same time the counter is reloaded, the read value is always FFFF16. If the TAi register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.

3. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

### 22.6.1.2 Timer A (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the UDF register, bits TAZIE, TA0TGL, and TA0TGH in the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to 1 (count starts).
   Always make sure bits TAZIE, TA0TGL, and TA0TGH in the TAiMR register, the UDF register, the ONSF register, and the TRGSR register are modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the TAi register is read at the same time the counter is reloaded, the read value is always FFFF16 when the timer counter underflows and 000016 when the timer counter overflows. If the TAi register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.

3. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| | | 63 | • **Figure 7.11 State Transition to Stop Mode and Wait Mode** modified, Note 7 is added |
| | | 64 | • **Figure 7.12 State Transition in Normal Mode** modified, note 5 deleted, note 6 and 7 are simplified |
| | | 65 | • **Table 7.7 Allowed Transition and Setting** note 2 partially modified, table con tents are partially modified |
| | | 68 | • **Figure 7.13 Procedure to Switch Clock Source From On-chip Oscillator Clock to Main Clock** is modified |
| | | | **Interrupt** |
| | | 70 | • Note is newly added |
| | | 73 | • **Table 9.1 Fixed Vector Tables** Note 2 is added |
| | | | **Watchdog Timer** |
| | | 89 | • Additional information of the WDTS register is inserted |
| | | 90 | • **Figure 10.1 Watchdog Timer Block Diagram** modified |
| | | | • **Figure 10.2 WDC Register and WDTS Register** All notes are deleted |
| | | - | • **10.2 Cold Start/Warm Start** Section is deleted |
| | | | **DMAC** |
| | | 96 | • Note is added |
| | | | **Timer** |
| | | 105 | • **Figure 12.6 TRGSR Register** Note 2 added |
| | | 117 | • **12.2 Timer B** Description of A/D trigger mode modified |
| | | | • **Figure 12.15 Timer B Block Diagram** “A/D trigger mode” is added |
| | | 123 | • **12.2.4 A/D Trigger Mode** Description modified |
| | | 129 | • **Figure 12.28 IDB0 Register, IDB1 Register, DTT Register, and ICTB2 Register** Information of bit 7 and 6 modified |
| | | 131 | • **Figure 12.30 TB2SC Register** Note 4 added, contents modified |
| | | 133 | • **Figure 12.32 TA1MR Register, TA2MR Register, TA4MR Register** MR0 bit is modified |
| | | 134 | • **Figure 12.33 Triangular Wave Modulation Operation** Description modified |
| | | 135 | • **Figure 12.34 Sawtooth Wave Modulation Operation** Description modified |
| | | 139 | • **Figure 12.38 TPRC Register** Bit map is modified |
| | | | **Timer S** |
| | | 142 | • **Figure 13.2 G1BT and G1BCR0 Registers** Function of G1BT register modified, note 3 is added, function of bits 5 to 3 modified, description patially modified |
| | | 143 | • **Figure 13.3 G1BCR1 Register** Note 1 is partially added |
| | | 146 | • **Figure 13.6 G1TM0 to G1TM7 Registers** Note 3 and 4 are added |
| | | 151-166 | • **Table 13.2, 13.5, 13,8, 13.9 and 13.10** Output wave form and Selectable function are modified |
| | | 155 | • **Figure 13.15 Base Timer Reset Operation by Base Timer Reset Register** Base timer overflow request line is added, base timer interrupt line is modified, |