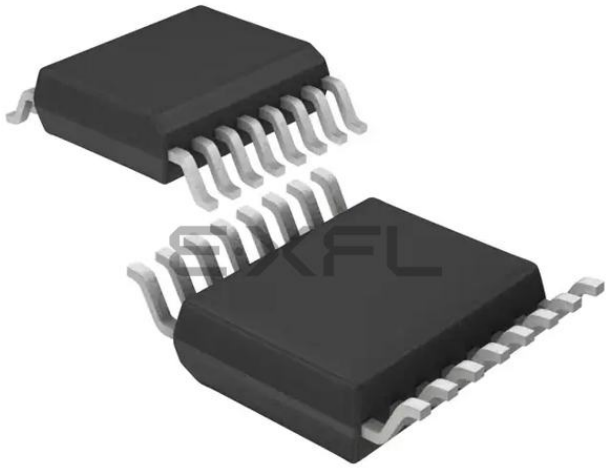
# Renesas Electronics America Inc - UPD78F0751MA-FAA-AX Datasheet



**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 | Active |
| Core Processor | 78K/0 |
| Core Size | 8-Bit |
| Speed | 10MHz |
| Connectivity | - |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 9 |
| Program Memory Size | 8KB (8K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | A/D 5x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 16-SSOP (0.173", 4.40mm Width) |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f0751ma-faa-ax |

**(a) $AV_{REF}$**

This is the A/D converter reference voltage input pin, and the positive power supply pin of A/D converter and ports 2 and 7.

When the A/D converter is not used, connect this pin directly to $V_{DD}$[Note].

**Note** Make the $AV_{REF}$ pin the same potential as the $V_{DD}$ pin when ports 2 and 7 are used as the digital ports.

**(b) $AV_{SS}$**

This is a ground potential pin of A/D converter and ports 2 and 7. Even when the A/D converter is not used, always use this pin with the same potential as the $V_{SS}$ pin.

**(c) $V_{DD}$**

$V_{DD}$ is a positive power supply pin.

**(d) $V_{SS}$**

$V_{SS}$ is a ground potential pin[Note].

**Note** In the 78K0/IY2 and 78K0/IA2, $V_{SS}$ functions alternately as the ground potential of the A/D converter. Be sure to connect $V_{SS}$ to a stabilized GND (= 0 V).

### 2.2.8 IC0

These are internally connected pins.

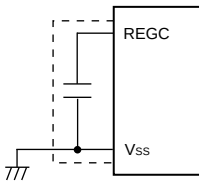
### 2.2.9 REGC

| 78K0/IY2 | 78K0/IA2 | 78K0/IB2 |
|---|---|---|
| 16 Pins | 20 Pins | 30 Pins/32 Pins |
| REGC | REGC | REGC |

**(a) REGC**

This is a pin for connecting regulator output (2.0 V/2.4 V) stabilization capacitance for internal operation. Connect this pin to $V_{SS}$ via a capacitor (0.47 to 1 $\mu$F). However, when using the STOP mode that has been entered since operation of the internal high-speed oscillation clock, 0.47 $\mu$F is recommended.

Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.




**Caution Keep the wiring length as short as possible for the broken-line part in the above figure.**

**(2) CALLT instruction table area**

The 64-byte area 0040H to 007FH can store the subroutine entry address of a 1-byte call instruction (CALLT).

**(3) Option byte area**

A 5-byte area of 0080H to 0084H and 1080H to 1084H can be used as an option byte area. Set the option byte at 0080H to 0084H when the boot swap is not used, and at 0080H to 0084H and 1080H to 1084H when the boot swap is used. For details, refer to **CHAPTER 24 OPTION BYTE**.

**(4) On-chip debug security ID setting area**

A 10-byte area of 0085H to 008EH and 1085H to 108EH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 0085H to 008EH when the boot swap is not used and at 0085H to 008EH and 1085H to 108EH when the boot swap is used. For details, refer to **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

**(5) CALLF instruction entry area**

The area 0800H to 0FFFH can perform a direct subroutine call with a 2-byte call instruction (CALLF).

### 3.1.2 Internal data memory space

78K0/Ix2 microcontrollers incorporate the following RAMs.

**(1) Internal high-speed RAM**

**Table 3-5. Internal High-Speed RAM Capacity**

| Product | | | Internal High-Speed RAM |
|---|---|---|---|
| 78K0/IY2 | 78K0/IA2 | 78K0/IB2 | |
| μPD78F0740, 78F0750 | – | – | 384 × 8 bits (FD80H to FEFFH) |
| μPD78F0741, 78F0751 | μPD78F0743, 78F0753 | μPD78F0745, 78F0755 | 512 × 8 bits (FD00H to FEFFH) |
| μPD78F0742, 78F0752 | μPD78F0744, 78F0754 | μPD78F0746, 78F0756 | 768 × 8 bits (FC00H to FEFFH) |

The 32-byte area FEE0H to FEFFH is assigned to four general-purpose register banks consisting of eight 8-bit registers per bank.

This area cannot be used as a program area in which instructions are written and executed.

The internal high-speed RAM can also be used as a stack memory.

### 3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (refer to **Tables 3-6 to 3-9 Special Function Register List** in **3.2.3 Special function registers (SFRs)**).

**Caution Do not access addresses to which SFRs are not assigned.**

**Table 3-7. Special Function Register List : 78K0/IA2 (6/7)**

| Address | Symbol | Bit No. | | | | | | | | R/W | Number of Bits Manipulated Simultaneously | | | After Reset | Reference page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | 1 | 8 | 16 | | |
| FFA5H | IICA | – | – | – | – | – | – | – | – | R/W | – | √ | – | 00H | 482 |
| FFA6H | SVA0 | – | – | – | – | – | – | – | 0 | R/W | – | √ | – | 00H | 482 |
| FFA7H | IICACTL0 | <IICE0> | <LREL0> | <WREL0> | <SPIE0> | <WTIM0> | <ACKE0> | <STT0> | <SPT0> | R/W | √ | √ | – | 00H | 484 |
| FFA8H | IICACTL1 | <WUP> | 0 | <CLD0> | <DAD0> | <SMC0> | <DFC0> | 0 | 0 | R/W | √ | √ | – | 00H | 493 |
| FFA9H | IICAF0 | <STCF> | <IICBSY> | 0 | 0 | 0 | 0 | <STCEN> | <IICRSV> | R/W | √ | √ | – | 00H | 491 |
| FFAAH | IICAS0 | <MSTS0> | <ALD0> | <EXC0> | <COI0> | <TRC0> | <ACKD0> | <STD0> | <SPD0> | R | √ | √ | – | 00H | 489 |
| FFABH | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| FFACH | RESF | 0 | 0 | 0 | WDTRF | 0 | 0 | 0 | LVIRF | R | – | √ | – | 00H[Note] | 628 |
| FFADH | IICWL | – | – | – | – | – | – | – | – | R/W | – | √ | – | FFH | 495 |
| FFAEH | IICWH | – | – | – | – | – | – | – | – | R/W | – | √ | – | FFH | 495 |
| FFAFH | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| FFB0H | TX1CR1 | – | – | – | – | – | – | – | – | R/W | – | – | √ | 0000H | 204 |
| FFB1H | | – | – | – | – | – | – | – | – | | | | | | |
| FFB2H | TX1CR2 | – | – | – | – | – | – | – | – | R/W | – | – | √ | 0000H | 204 |
| FFB3H | | – | – | – | – | – | – | – | – | | | | | | |
| FFB4H | TX1CR3 | – | – | – | – | – | – | – | – | R/W | – | – | √ | 0000H | 204 |
| FFB5H | | – | – | – | – | – | – | – | – | | | | | | |
| FFB6H | TX1CCR0 | – | – | – | – | – | – | – | – | R/W | – | – | √ | 0000H | 204 |
| FFB7H | | – | – | – | – | – | – | – | – | | | | | | |
| FFB8H | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| FFB9H | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| FFBAH | TMC00 | 0 | 0 | 0 | 0 | TMC003 | TMC002 | 0 | <OVF00> | R/W | √ | √ | – | 00H | 263 |
| FFBBH | PRM00 | 0 | 0 | ES010 | ES000 | 0 | 0 | PRM001 | PRM000 | R/W | √ | √ | – | 00H | 268 |
| FFBCH | CRC00 | 0 | 0 | 0 | 0 | 0 | CRC002 | CRC001 | CRC000 | R/W | √ | √ | – | 00H | 264 |

**Note** The reset value of RESF varies depending on the reset source.

**Remark** For a bit name enclosed in angle brackets (<>), the bit name is defined as a reserved word in the RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0.

**Table 4-3. Port Functions (78K0/IA2)**

| Function Name | I/O | Function | After Reset | Alternate Function |
|---|---|---|---|---|
| P00 | I/O | Port 0.<br>1-bit I/O port.<br>Input/output can be specified in 1-bit units.<br>Use of an on-chip pull-up resistor can be specified by a software setting. | Input port | TI000/INTP0/<TOH1>/<TI51> |
| P20 | I/O | Port 2.<br>6-bit I/O port.<br>Input/output can be specified in 1-bit units. | Analog input | ANI0/AMP-Note |
| P21 | | | | ANI1/AMPOUTNote/PGAINNote |
| P22 | | | | ANI2/AMP+Note |
| P23 | | | | ANI3/CMP2+ |
| P24 | | | | ANI4/CMP0+ |
| P25 | | | | ANI5/CMP1+ |
| P31 | I/O | Port 3.<br>4-bit I/O port.<br>Input/output can be specified in 1-bit units.<br>Use of an on-chip pull-up resistor can be specified by a software setting. | Input port | TOX00/INTP2/TOOLC1 |
| P32 | | | | TOX01/INTP3/TOOLD1 |
| P33 | | | | TOX10 |
| P34 | | | | TOX11/INTP4/<TOH1>/<TI51> |
| P60 | I/O | Port 6.<br>2-bit I/O port.<br>Input/output can be specified in 1-bit units.<br>Input can be set to SMBus input buffer in 1-bit units.<br>Output can be set to N-ch open-drain output ($V_{DD}$ tolerance).<br>Use of an on-chip pull-up resistor can be specified by a software setting. | Input port | SCLA0/TxD6 |
| P61 | | | | SDAA0/RxD6 |
| P121 | Input | Port 12.<br>3-bit input-only port.<br>For only P125, use of an on-chip pull-up resistor can be specified by a software setting. | Input port | X1/TOOLC0/<TI000>/<INTP0> |
| P122 | | | | X2/EXCLK/TOOLD0 |
| P125 | | | | $\overline{\text{RESET}}$ |

**Note** $\mu$PD78F0753, 78F0754 (products with operational amplifier) only

**Remark** Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

### 4.2.1 Port 0

| 78K0/IY2 | 78K0/IA2 | 78K0/IB2 | |
|---|---|---|---|
| 16 Pins | 20 Pins | 30 Pins | 32 Pins |
| – | P00/TI000/INTP0/<TOH1>/<TI51> | P00/TI000/INTP0 | P00/TI000/INTP0/<TOH1>/<TI51> |
| – | – | P01/TO00/TI010 | – |
| – | – | P02/$\overline{\text{SSI11}}$/INTP5 | P02/$\overline{\text{SSI11}}$/INTP5 |

**Remark** Functions in angle brackets < > can be assigned by setting the input switch control register (MUXSEL).

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P02 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

This port can also be used for timer I/O, external interrupt request input, and serial interface chip select input.

The timer input or timer output can be assigned to P00 of the 78K0/IA2 and 78K0/IB2 (32 Pins) by setting the port alternate switch control register (MUXSEL).

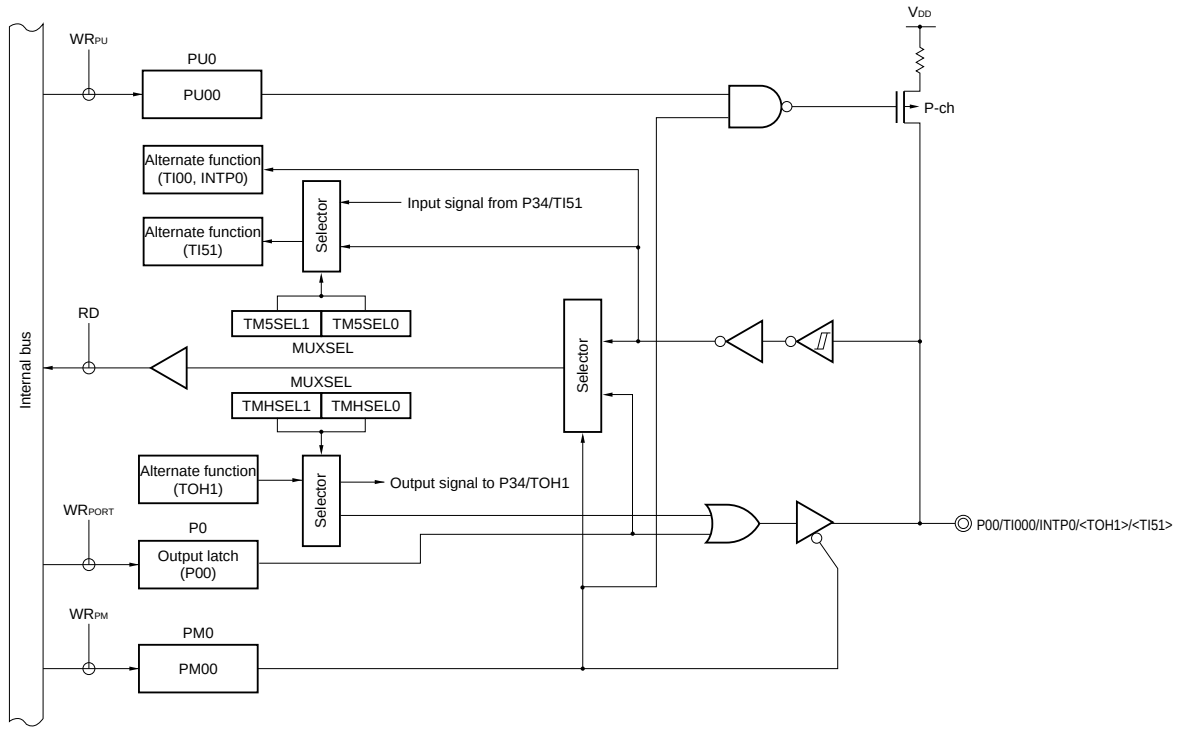
Reset signal generation sets port 0 to input mode.

Figures 4-1 to 4-3 show block diagrams of port 0.

**Caution When using the 78K0/IB2 (32 pins), be sure to clear bit 1 of port mode register 0 (PM0) to 0** after the reset status is released**.**

**Figure 4-1. Block Diagram of P00 (1/2)**

**(1) 78K0/IA2, 78K0/IB2 (32 Pins)**




P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
MUXSEL: Port alternate switch control register
RD: Read signal
WR××: Write signal

**(6) Oscillation stabilization time counter status register (OSTC)**

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter. When X1 clock oscillation starts with the internal high-speed oscillation clock used as the CPU clock, the X1 clock oscillation stabilization time can be checked.

OSTC can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset is released (reset by $\overline{\text{RESET}}$ input, POC, LVI, and WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

**Figure 5-7. Format of Oscillation Stabilization Time Counter Status Register (OSTC)**

Address: FFA3H After reset: 00H R

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTC | 0 | 0 | 0 | MOST11 | MOST13 | MOST14 | MOST15 | MOST16 |

| MOST11 | MOST13 | MOST14 | MOST15 | MOST16 | Oscillation stabilization time status | |
|---|---|---|---|---|---|---|
| | | | | | | $f_X$ = 10 MHz |
| 1 | 0 | 0 | 0 | 0 | $2^{11}/f_X$ min. | 204.8 μs min. |
| 1 | 1 | 0 | 0 | 0 | $2^{13}/f_X$ min. | 819.2 μs min. |
| 1 | 1 | 1 | 0 | 0 | $2^{14}/f_X$ min. | 1.64 ms min. |
| 1 | 1 | 1 | 1 | 0 | $2^{15}/f_X$ min. | 3.27 ms min. |
| 1 | 1 | 1 | 1 | 1 | $2^{16}/f_X$ min. | 6.55 ms min. |

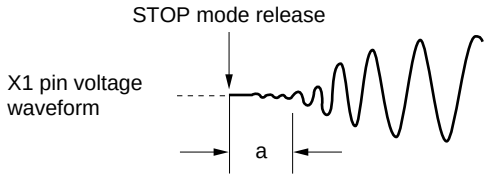
**Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.**

**2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.**

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

**Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.**

**3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).**




**Remark** $f_X$: X1 clock oscillation frequency

**(2) 16-bit timer Xn operation control register 1 (TXnCTL1)**

TXnCTL1 is a register that sets timer start via detection of INTP0 rising edge, output gate function by TOH1 output, the PWM output operation, and synchronous operation mode.

TXnCTL1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TXnCTL1 to 00H.

**Remark** n = 0, 1

**Figure 6-8. Format of 16-Bit Timer X0 Operation Control Register 1 (TX0CTL1)**

Address: FF7FH After reset: 00H R/W

| Symbol | <7> | 6 | <5> | <4> | <3> | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TX0CTL1 | TX0INTPST | 0 | TX0PWM CE | TX0PWM CINV | TX0PWM | 0 | 0 | 0 |

| TX0INTPST | Control of timer start operation via detection of INTP0 rising edge |
|---|---|
| 0 | Disables timer start operation via detection of INTP0 rising edge (starts timer via setting (1) of TX0TMC)[Note 1]. |
| 1 | Enables timer start operation via detection of INTP0 rising edge [Note 2]. |

| TX0PWM CE | Control of TOX0n output gate function by TOH1 output (n = 0, 1) |
|---|---|
| 0 | Does not use output gate function. |
| 1 | Use output gate function. |

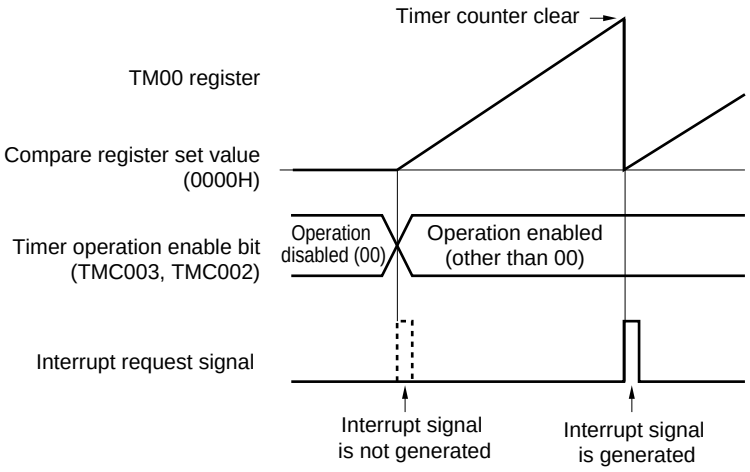
| TX0PWM CINV | Setting of TOXmn output by TOH1 output (mn = 00, 01, 10, 11) |
|---|---|
| 0 | • Performs PWM output from TOXmn while the TOH1 output is high level.<br>• Outputs a default level of TOXmn while the TOH1 output is low level. |
| 1 | • Performs PWM output from TOXmn while the TOH1 output is low level.<br>• Outputs a level that is the inverse of the default level of TOXmn while the TOH1 output is high level. |

| TX0PWM | TMX0 PWM output operation setting |
|---|---|
| 0 | • Single output (TOX00 pin only)<br>• INTTMX0 is generated upon match of counter and TX0CR1 register |
| 1 | • Dual output (TOX00 and TOX01 pins)<br>• INTTMX0 is generated upon match of counter and TX0CR3 register |

**Notes 1.** In TMX0 or TMX1 synchronous start mode, a timer start operation via detection of INTP0 rising edge cannot be performed, so set TX0INTPST to 0.

**2.** If 1 is set to TX0TMC after setting 1 to TX0INTPST, detection of INTP0 rising edge will be waited for. If the INTP0 rising edge is detected, 16-bit timer X0 will start counting up.

**Caution 1. During timer operation, setting the other bits of TX0CTL1 is prohibited. However, TX0CTL1 can be refreshed (the same value is written).**

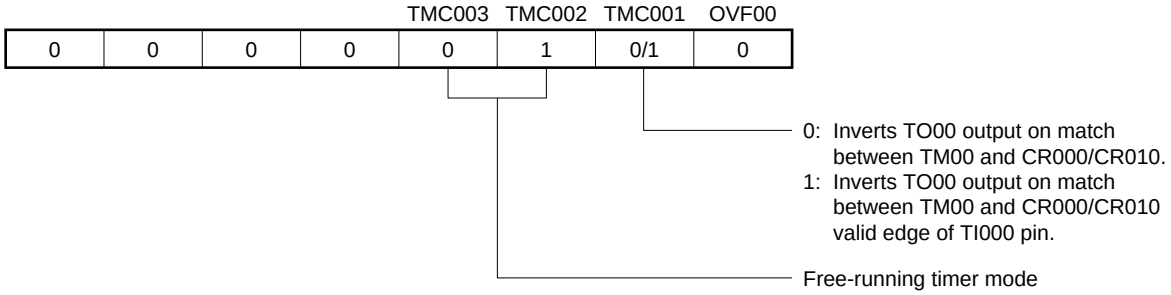



**Remarks 1.** N: CR000 register set value, M: CR010 register set value

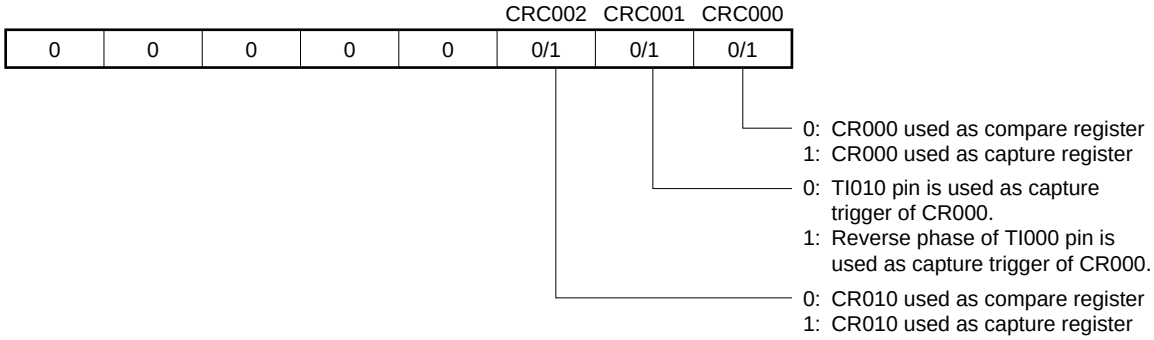
**2.** For details of the operation enable bits (bits 3 and 2 (TMC003 and TMC002)), refer to **7.3 (1) 16-bit timer mode control register 00 (TMC00)**.

**Figure 7-39. Example of Register Settings in Free-Running Timer Mode (1/2)**

**(a) 16-bit timer mode control register 00 (TMC00)**




**(b) Capture/compare control register 00 (CRC00)**




**(c) 16-bit timer output control register 00 (TOC00)**

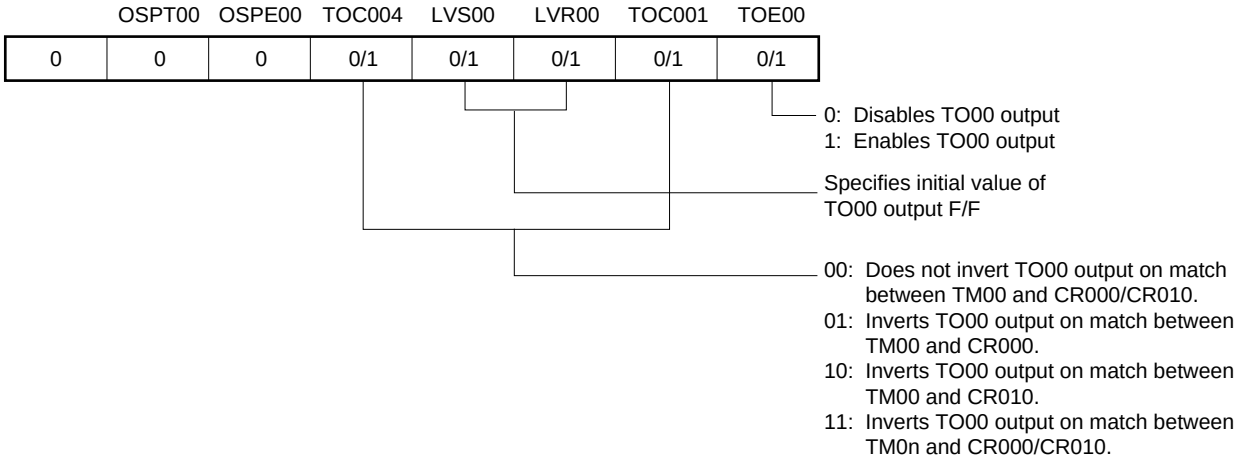
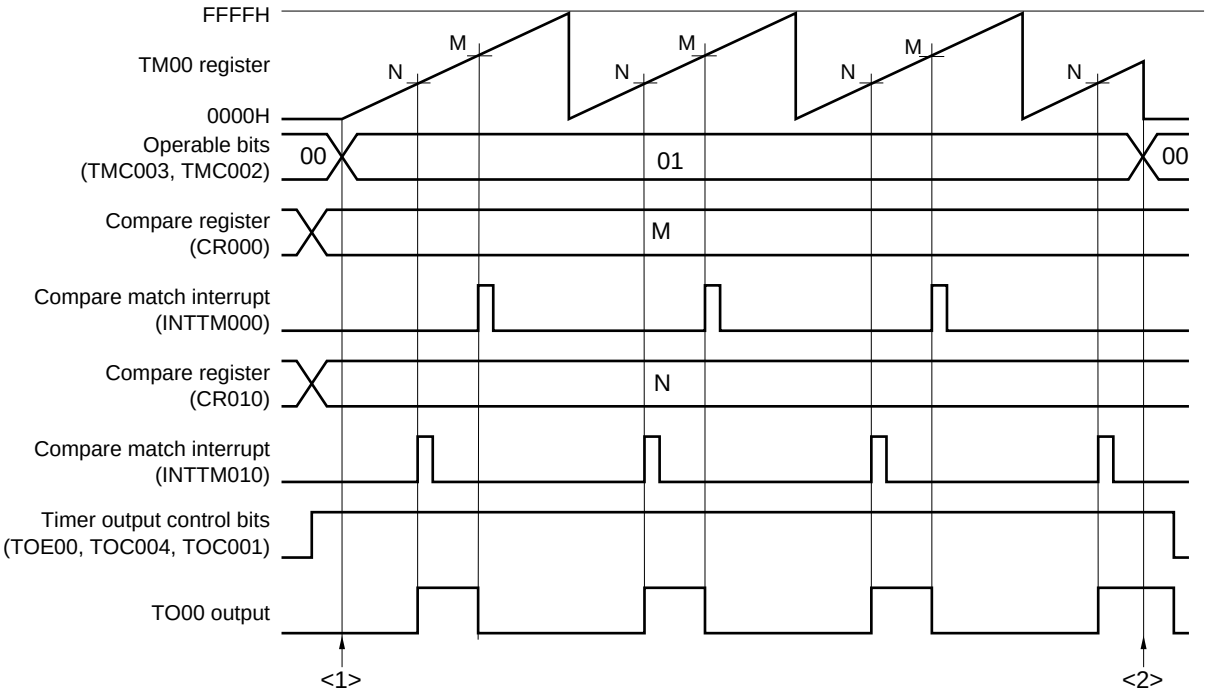
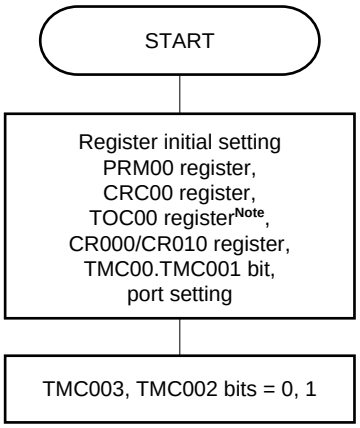


**Figure 7-40. Example of Software Processing in Free-Running Timer Mode**



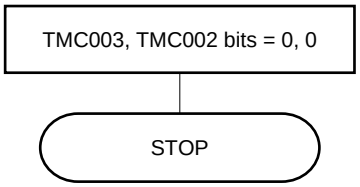

<1> Count operation start flow




Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 01.

Starts count operation

<2> Count operation stop flow




The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

**Note** Care must be exercised when setting TOC00. For details, refer to **7.3 (3) 16-bit timer output control register 00 (TOC00)**.

## 14.2 Configuration of Serial Interface UART6/DALI

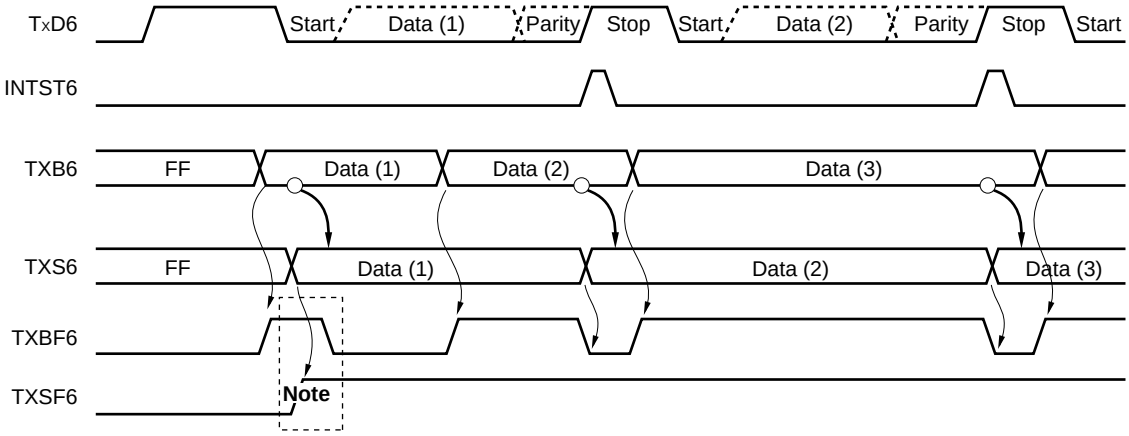
Serial interface UART6/DALI includes the following hardware.

**Table 14-1. Configuration of Serial Interface UART6/DALI**

| Item | Configuration |
|---|---|
| Registers | UART receive buffer register 6 (RXB6)<br>UART/DALI receive shift register 6 (RXS6)<br>DALI receive buffer register (RXBDL)<br>UART/DALI transmit buffer register 6 (TXB6)<br>UART/DALI transmit shift register 6 (TXS6) |
| Control registers | UART/DALI mode control register (UADLCTL)<br>UART/DALI operation mode register 6 (ASIM6)<br>UART/DALI reception error status register 6 (ASIS6)<br>UART transmission status register 6 (ASIF6)<br>Clock selection register 6 (CKSR6)<br>Baud rate generator control register 6 (BRGC6)<br>LIN operation control register 6 (ASICL6)<br>Input switch control register (ISC)<br>Port mode register 6 (PM6)<br>Port register 6 (P6)<br>Port output mode register 6 (POM6) |

Figure 14-19 shows the timing of starting continuous transmission, and Figure 14-20 shows the timing of ending continuous transmission.

**Figure 14-19. Timing of Starting Continuous Transmission**




**Note** When ASIF6 is read, there is a period in which TXBF6 and TXSF6 = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6 bit.

**Remark** TxD6: TxD6 pin (output)
INTST6: Interrupt request signal
TXB6: UART/DALI transmit buffer register 6
TXS6: Transmit shift register 6
ASIF6: UART transmission status register 6
TXBF6: Bit 1 of ASIF6
TXSF6: Bit 0 of ASIF6

**(2) IICA status register 0 (IICAS0)**

This register indicates the status of $I^2C$.

This register is read by a 1-bit or 8-bit memory manipulation instruction only when STT0 = 1 and during the wait period.

Reset signal generation clears this register to 00H.

**Caution Reading the IICAS0 register while the address match wakeup function is enabled (WUP = 1) in STOP mode is prohibited. When the WUP bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA0 interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIE0 = 1) the interrupt generated by detecting a stop condition and read the IICAS0 register after the interrupt has been detected.**

**Remark** STT0: Bit 1 of IICA control register 0 (IICACTL0)
WUP: Bit 7 of IICA control register 1 (IICACTL1)

**Figure 15-6. Format of IICA Status Register 0 (IICAS0) (1/3)**

Address: FFAAH After reset: 00H R

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IICAS0 | MSTS0 | ALD0 | EXC0 | COI0 | TRC0 | ACKD0 | STD0 | SPD0 |

| MSTS0 | Master status |
|---|---|
| 0 | Slave device status or communication standby status |
| 1 | Master device communication status |

| Condition for clearing (MSTS0 = 0) | Condition for setting (MSTS0 = 1) |
|---|---|
| • When a stop condition is detected<br>• When ALD0 = 1 (arbitration loss)<br>• Cleared by LREL0 = 1 (exit from communications)<br>• When the IICE0 bit changes from 1 to 0 (operation stop)<br>• Reset | • When a start condition is generated |

| ALD0 | Detection of arbitration loss |
|---|---|
| 0 | This status means either that there was no arbitration or that the arbitration result was a “win”. |
| 1 | This status indicates the arbitration result was a “loss”. The MSTS0 bit is cleared. |

| Condition for clearing (ALD0 = 0) | Condition for setting (ALD0 = 1) |
|---|---|
| • Automatically cleared after the IICAS0 register is read[Note]<br>• When the IICE0 bit changes from 1 to 0 (operation stop)<br>• Reset | • When the arbitration result is a “loss”. |

**Note** This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the ALD0 bit of the IICAS0 register. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

**Remark** LREL0: Bit 6 of IICA control register 0 (IICACTL0)
IICE0: Bit 7 of IICA control register 0 (IICACTL0)

### 15.5.7 Canceling wait

The $I^2C$ usually cancels a wait state by the following processing.

- Writing data to IICA shift register (IICA)
- Setting bit 5 (WREL0) of IICA control register 0 (IICACTL0) (canceling wait)
- Setting bit 1 (STT0) of IICACTL0 register (generating start condition)Note
- Setting bit 0 (SPT0) of IICACTL0 register (generating stop condition)Note

**Note** Master only

When the above wait canceling processing is executed, the $I^2C$ cancels the wait state and communication is resumed.

To cancel a wait state and transmit data (including addresses), write the data to the IICA register.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL0) of the IICA control register 0 (IICACTL0) to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT0) of the IICACTL0 register to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT0) of the IICACTL0 register to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to the IICA register after canceling a wait state by setting the WREL0 bit to 1, an incorrect value may be output to SDAA0 line because the timing for changing the SDAA0 line conflicts with the timing for writing the IICA register.

In addition to the above, communication is stopped if the IICE0 bit is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the $I^2C$ bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL0) of the IICACTL0 register, so that the wait state can be canceled.

**Caution If a processing to cancel a wait state executed when WUP (bit 7 of the IICA control register 1 (IICACTL1)) = 1, the wait state will not be canceled.**

**(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop**

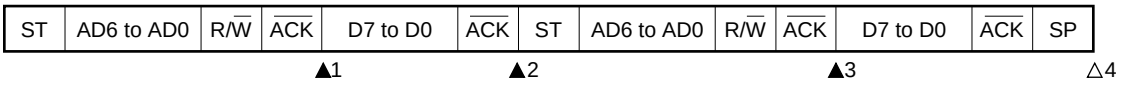
**(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))**


▲1 ▲2 ▲3 △4

▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001×000B
▲3: IICAS0 = 00000110B
△4: IICAS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))**




▲1 ▲2 ▲3 △4

▲1: IICAS0 = 0001×110B
▲2: IICAS0 = 0001××00B
▲3: IICAS0 = 00000110B
△4: IICAS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

### 16.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

In this mode, communication is executed by using three lines: the serial clock ($\overline{SCK11}$), serial output (SO11), and serial input (SI11) lines.

#### (1) Registers used

- Serial operation mode register 11 (CSIM11)
- Serial clock selection register 11 (CSIC11)
- Port mode register 0 and 3 (PM0, PM3)
- Port register 3 (P3)

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set the CSIC11 register (refer to **Figure 16-3**).
<2> Set bits 4 to 6 (DIR11, SSE11, and TRMD11) of the CSIM11 register (refer to **Figure 16-2**).
<3> Set bit 7 (CSIE11) of the CSIM11 register to 1. → Transmission/reception is enabled.
<4> Write data to transmit buffer register 11 (SOTB11). → Data transmission/reception is started.
Read data from serial I/O shift register 11 (SIO11). → Data reception is started.

**Caution Take relationship with the other party of communication when setting the port mode register and port register.**

# CHAPTER 24 OPTION BYTE

## 24.1 Functions of Option Bytes

The flash memory at 0080H to 0084H of the 78K0/Ix2 microcontrollers is an option byte area. When power is turned on or when the device is restarted from the reset status, the device automatically references the option bytes and sets specified functions. When using the product, be sure to set the following functions by using the option bytes.

When the boot swap operation is used during self-programming, 0080H to 0084H are switched to 1080H to 1084H. Therefore, set values that are the same as those of 0080H to 0084H to 1080H to 1084H in advance.

**(1) 0080H/1080H**

- Internal low-speed oscillator operation
  - Can be stopped by software
  - Cannot be stopped
- Watchdog timer interval time setting
- Watchdog timer counter operation
  - Enabled counter operation
  - Disabled counter operation
- Watchdog timer window open period setting

**Caution Set a value that is the same as that of 0080H to 1080H because 0080H and 1080H are switched during the boot swap operation.**

**(2) 0081H/1081H**

- LVI default start operation control
  - During LVI default start function enabled (LVISTART = 1)
    The device is in the reset state after reset release or upon power application and until the supply voltage reaches 1.91 V (TYP.). It is released from the reset state when the voltage exceeds 1.91 V (TYP.).
    If the supply voltage rises to 2.7 V after reset release or power application at a rate slower than 0.5 V/ms (MIN.), LVI default start function operation is recommended.
  - During LVI default start function stopped (LVISTART = 0)
    The device is in the reset state after reset release or upon power application and until the supply voltage reaches 1.61 V (TYP.). It is released from the reset state when the voltage exceeds 1.61 V (TYP.).

**Caution LVISTART can only be written by using a dedicated flash memory programmer. It cannot be set or change during self-programming or boot swap operation during self-programming.**

**Figure 24-1. Format of Option Byte (2/3)**

Address: 0081H/1081H[Notes 1, 2]

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | LVISTART |

| LVISTART | LVI default start operation control |
|---|---|
| 0 | LVI is OFF by default upon power application (LVI default start function stopped) |
| 1 | LVI is ON by default upon power application (LVI default start function enabled) |

**Notes 1.** LVISTART can only be written by using a dedicated flash memory programmer. It cannot be set during self-programming or boot swap operation during self-programming. However, because the value of 1081H is copied to 0081H during the boot swap operation, it is recommended to set a value that is the same as that of 0081H to 1081H when the boot swap function is used.

**2.** To change the setting for the LVI default start, set the value to 0081H again after batch erasure (chip erasure) of the flash memory. The setting cannot be changed after the memory of the specified block is erased.

**Caution Be sure to clear bits 7 to 1 to "0".**

Address: 0082H/1082H[Note]

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | R4M8MSEL |

| R4M8MSEL | Internal high-speed oscillation clock frequency selection |
|---|---|
| 0 | 8 MHz (TYP.) |
| 1 | 4 MHz (TYP.) |

**Note** Set a value that is the same as that of 0082H to 1082H because 0082H and 1082H are switched during the boot swap operation.

**Caution Be sure to clear bits 7 to 1 to "0".**

## 25.7 Processing Time for Each Command When PG-FP5 Is Used (Reference)

The following table shows the processing time for each command (reference) when the PG-FP5 is used as a dedicated flash memory programmer.

**Table 25-9. Processing Time for Each Command When PG-FP5 Is Used (Reference) (1/2)**

(1) Products with internal ROMs of the 4 KB: μPD78F0740, 78F0750

| Command of PG-FP5 | Port: UART-Internal-OSC (Internal high-speed oscillation clock ($f_{IH}$: 8 MHz (typ.)), Speed: 500,000 bps |
|---|---|
| Signature | 0.5 s (typ.) |
| Blankcheck | 0.5 s (typ.) |
| Erase | 0.5 s (typ.) |
| Program | 1 s (typ.) |
| Verify | 1 s (typ.) |
| E.P.V | 1 s (typ.) |
| Checksum | 0.5 s (typ.) |
| Security | 0.5 s (typ.) |

(2) Products with internal ROMs of the 8 KB: μPD78F0741, 78F0743, 78F0745, 78F0751, 78F0753, 78F0755

| Command of PG-FP5 | Port: UART-Internal-OSC (Internal high-speed oscillation clock ($f_{IH}$: 8 MHz (typ.)), Speed: 500,000 bps |
|---|---|
| Signature | 0.5 s (typ.) |
| Blankcheck | 0.5 s (typ.) |
| Erase | 1 s (typ.) |
| Program | 1.5 s (typ.) |
| Verify | 1 s (typ.) |
| E.P.V | 1.5 s (typ.) |
| Checksum | 0.5 s (typ.) |
| Security | 0.5 s (typ.) |

**Caution When executing boot swapping, do not use the E.P.V. command with the dedicated flash memory programmer.**

## C.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/7)

| Edition | Description | Chapter |
|---|---|---|
| 2nd Edition | Modification of related documents | **INTRODUCTION** |
| | Modification of **Figure 6-8 Format of 16-Bit Timer X0 Operation Control Register 1 (TX0CTL1)** | **CHAPTER 6 16-BIT TIMERS X0 AND X1** |
| | Modification of **Figure 6-9 Format of 16-Bit Timer X1 Operation Control Register 1 (TX1CTL1)** | |
| | Modification of **Table 6-2 Register Setting Bits Controlling Operation Mode and 16-bit Timers X0 and X1 (1/2)** | |
| | Modification of **Figure 6-18 Example of Register Settings for PWM Output Operation (Single Mode)** | |
| | Modification of **Figure 6-20 Example of Register Settings for PWM Output Operation (Dual Mode)** | |
| | Modification of **Figure 6-22 Example of Register Settings for PWM Output Operation (TMX0 and TMX1 synchronous start/clear mode, PWM output: TOX00, TOX01, TOX10, and TOX11 pins)** | |
| | Modification of **Figure 6-24 Example of Register Settings for PWM Output Operation (TMX0 and TMX1 Synchronous Start Mode, PWM Output: TOX00, TOX01, TOX10, and TOX11 Pins)** | |
| | Modification of **6.4 (5) PWM output operation (PWM output from TOX0n when TOH1 output is at high level)** | |
| | Modification of **6.4 (6) PWM output operation (PWM output from TOX0n when TOH1 output is at low level)** | |
| | Addition of **6.4 (7) PWM output operation (PWM output from TOX1n when TOH1 output is at high level)** | |
| | Addition of **6.4 (8) PWM output operation (PWM output from TOX00, TOX01, TOX10, and TOX11 when TOH1 output is at high level)** | |
| | Deletion of **Caution 2** in **11.4.1 Basic operations of A/D converter (software trigger mode)** in old edition | **CHAPTER 11 A/D CONVERTER** |
| | Deletion of **Caution 2** in **11.4.2 Basic operations of A/D converter (timer trigger mode)** in old edition | |
| | **11.4.5 A/D converter operation mode**<br>• Deletion of **Caution 2** in • **Software trigger mode** in old edition<br>• Deletion of **Caution 2** in • **Timer trigger mode** in old edition | |
| | Modification of and addition of **Remark** to **Figure 12-2 Format of Operational Amplifier 0 Control Register (AMP0M) (Products with Operational Amplifier Only)** | **CHAPTER 12 OPERATIONAL AMPLIFIERS** |
| | Modification of **25.4.2 TOOLD0 and TOOLD1 pins** | **CHAPTER 25 FLASH MEMORY** |
| | Modification of **Note 4** in **Figure A-1 Development Tool Configuration (1/2)** | **APPENDIX A DEVELOPMENT TOOLS** |
| | Addition of **Note 4** to and modification of **Note 6** in **Figure A-1 Development Tool Configuration (2/2)** | |
| | Modification of description of **System simulator** in **A.5 Debugging Tools (Software)** | |
| | Addition of chapter | **APPENDIX B REVISION HISTORY** |