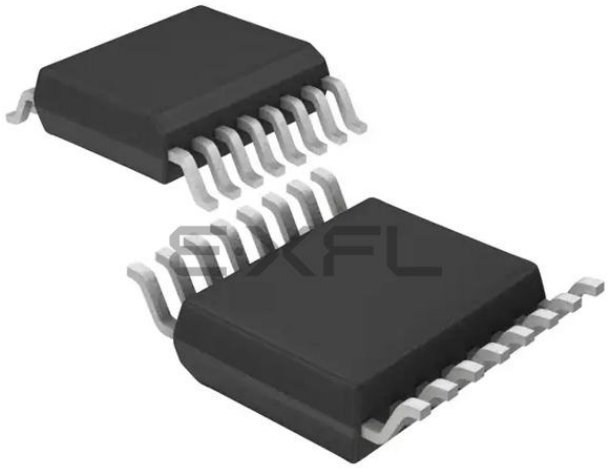
# Renesas Electronics America Inc - UPD78F0742MA-FAA-AX Datasheet



**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 | 16KB (16K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 768 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/upd78f0742ma-faa-ax |

**Documents Related to Development Tools (Software)**

| Document Name | | Document No. |
|---|---|---|
| RA78K0 Ver.3.80 Assembler Package User's Manual[Note 1] | Operation | U17199E |
| | Language | U17198E |
| | Structured Assembly Language | U17197E |
| 78K0 Assembler Package RA78K0 Ver.4.01 Operating Precautions (Notification Document)[Note 1] | | ZUD-CD-07-0181-E |
| CC78K0 Ver.3.70 C Compiler User's Manual[Note 2] | Operation | U17201E |
| | Language | U17200E |
| 78K0 C Compiler CC78K0 Ver. 4.00 Operating Precautions (Notification Document)[Note 2] | | ZUD-CD-07-0103-E |
| SM+ System Simulator User's Manual | Operation | U18601E |
| | User Open Interface | U18212E |
| ID78K0-QB Ver.2.94 Integrated Debugger User's Manual | Operation | U18330E |
| ID78K0-QB Ver.3.00 Integrated Debugger User's Manual | Operation | U18492E |
| PM plus Ver.5.20[Note 3] User's Manual | | U16934E |
| PM+ Ver.6.30[Note 4] User's Manual | | U18416E |

**Notes 1.** This document is installed into the PC together with the tool when installing RA78K0 Ver. 4.01. For descriptions not included in "78K0 Assembler Package RA78K0 Ver. 4.01 Operating Precautions", refer to the user's manual of RA78K0 Ver. 3.80.

**2.** This document is installed into the PC together with the tool when installing CC78K0 Ver. 4.00. For descriptions not included in "78K0 C Compiler CC78K0 Ver. 4.00 Operating Precautions", refer to the user's manual of CC78K0 Ver. 3.70.

**3.** PM plus Ver. 5.20 is the integrated development environment included with RA78K0 Ver. 3.80.

**4.** PM+ Ver. 6.30 is the integrated development environment included with RA78K0 Ver. 4.01. Software tool (assembler, C compiler, debugger, and simulator) products of different versions can be managed.

**Other Documents**

| Document Name | Document No. |
|---|---|
| RENESAS MICROCOMPUTER GENERAL CATALOG | R01CS00001E |
| Semiconductor Package Mount Manual | **Note** |
| Quality Grades on NEC Semiconductor Devices | C11531E |
| NEC Semiconductor Device Reliability/Quality Control System | C10983E |
| Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD) | C11892E |

**Note** See the "Semiconductor Package Mount Manual" website
<R> (http://www.renesas.com/products/package/manual/index.jsp).

**Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.**



Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

**(1) Vector table area**

The 64-byte area 0000H to 003FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area.

Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

**Table 3-4. Vector Table**

| Vector Table Address | Interrupt Source | 78K0/IY2 | 78K0/IA2 | 78K0/IB2 | |
|---|---|---|---|---|---|
| | | 16 Pins | 20 Pins | 30 Pins | 32 Pins |
| 0000H | $\overline{\text{RESET}}$ input, POC, LVI, WDT | √ | √ | √ | √ |
| 0004H | INTLVI | √ | √ | √ | √ |
| 0006H | INTP0 | √ | √ | √ | √ |
| 0008H | INTP1 | – | – | √ | – |
| 000AH | INTP2 | √ | √ | √ | √ |
| 000CH | INTP3 | √ | √ | √ | √ |
| 000EH | INTP4 | √ | √ | √ | √ |
| 0010H | INTP5 | – | – | √ | √ |
| 0012H | INTSRE6 | – | √ | √ | √ |
| 0014H | INTSR6 | – | √ | √ | √ |
| 0016H | INTST6 | – | √ | √ | √ |
| 0018H | INTCSI11 | – | – | √ | √ |
| 001AH | INTTMH1 | √ | √ | √ | √ |
| 001CH | INTTMX0 | √ | √ | √ | √ |
| 001EH | INTTMX1 | √ | √ | √ | √ |
| 0020H | INTTM000 | √ | √ | √ | √ |
| 0022H | INTTM010 | √ | √ | √ | √ |
| 0024H | INTAD | √ | √ | √ | √ |
| 002AH | INTTM51 | √ | √ | √ | √ |
| 002CH | INTCMP0 | √ | √ | √ | √ |
| 002EH | INTCMP1 | √ | √ | √ | √ |
| 0030H | INTCMP2 | √ | √ | √ | √ |
| 0034H | INTIICA0 | – | √ | √ | √ |
| 003EH | BRK | √ | √ | √ | √ |

**Remark** √: Mounted, –: Not mounted

## 3.2 Processor Registers

The 78K0/Ix2 microcontrollers incorporate the following processor registers.

### 3.2.1 Control registers

The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

**(1) Program counter (PC)**

The program counter is a 16-bit register that holds the address information of the next program to be executed.

In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched.

When a branch instruction is executed, immediate data and register contents are set.

Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the program counter.

**Figure 3-7. Format of Program Counter**

| | 15 | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PC | PC15 | PC14 | PC13 | PC12 | PC11 | PC10 | PC9 | PC8 | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 |

**(2) Program status word (PSW)**

The program status word is an 8-bit register consisting of various flags set/reset by instruction execution.

Program status word contents are stored in the stack area upon vectored interrupt request acknowledge or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets PSW to 02H.

**Figure 3-8. Format of Program Status Word**

| | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|
| PSW | IE | Z | RBS1 | AC | RBS0 | 0 | ISP | CY |

**(a) Interrupt enable flag (IE)**

This flag controls the interrupt request acknowledge operations of the CPU.

When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgment is controlled with an in-service priority flag (ISP), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

**(b) Zero flag (Z)**

When the operation result is zero, this flag is set (1). It is reset (0) in all other cases.

**(c) Register bank select flags (RBS0 and RBS1)**

These are 2-bit flags to select one of the four register banks.

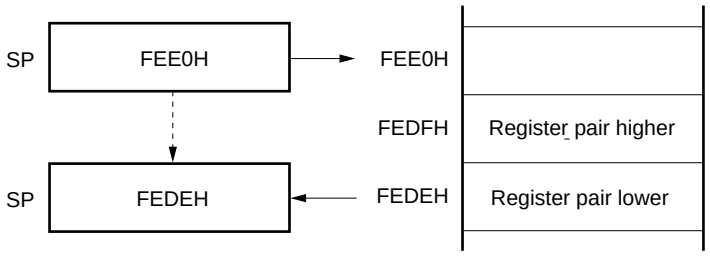
In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

**(d) Auxiliary carry flag (AC)**

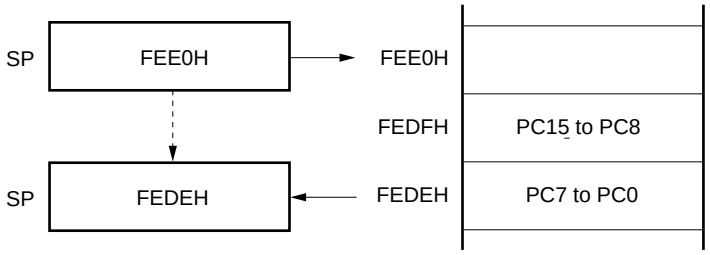
If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (1). It is reset (0) in all other cases.

**Figure 3-10. Data to Be Saved to Stack Memory**

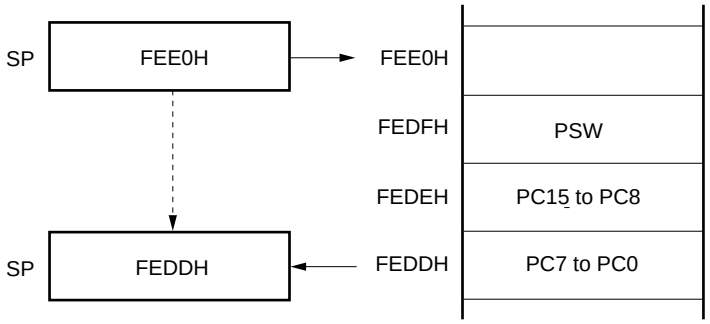
**(a) PUSH rp instruction (when SP = FEE0H)**




**(b) CALL, CALLF, CALLT instructions (when SP = FEE0H)**




**(c) Interrupt, BRK instructions (when SP = FEE0H)**



**(2) A/D conversion start timing signal output**

If bit 1 (TXnADEN) of the 16-bit timer Xn operation control register 2 (TXnCTL2) is set to 1, the generation of the A/D conversion synchronization trigger is enabled. If bit 7 (TXnTMC) of the 16-bit timer Xn operation control register 0 (TXnCTL0) is set to 1, the count operation is started in synchronization with the count clock.

When the value of the 16-bit timer counter Xn (TMXn) later matches the value of TXnCCR0, the A/D conversion synchronization trigger is generated. When the value of TMXn matches the value of TXnCRm, TMXn is cleared to 0000H.

To output the A/D conversion start timing signal, satisfy the relationship between TXnCCR0 and TXnCRm as follows.

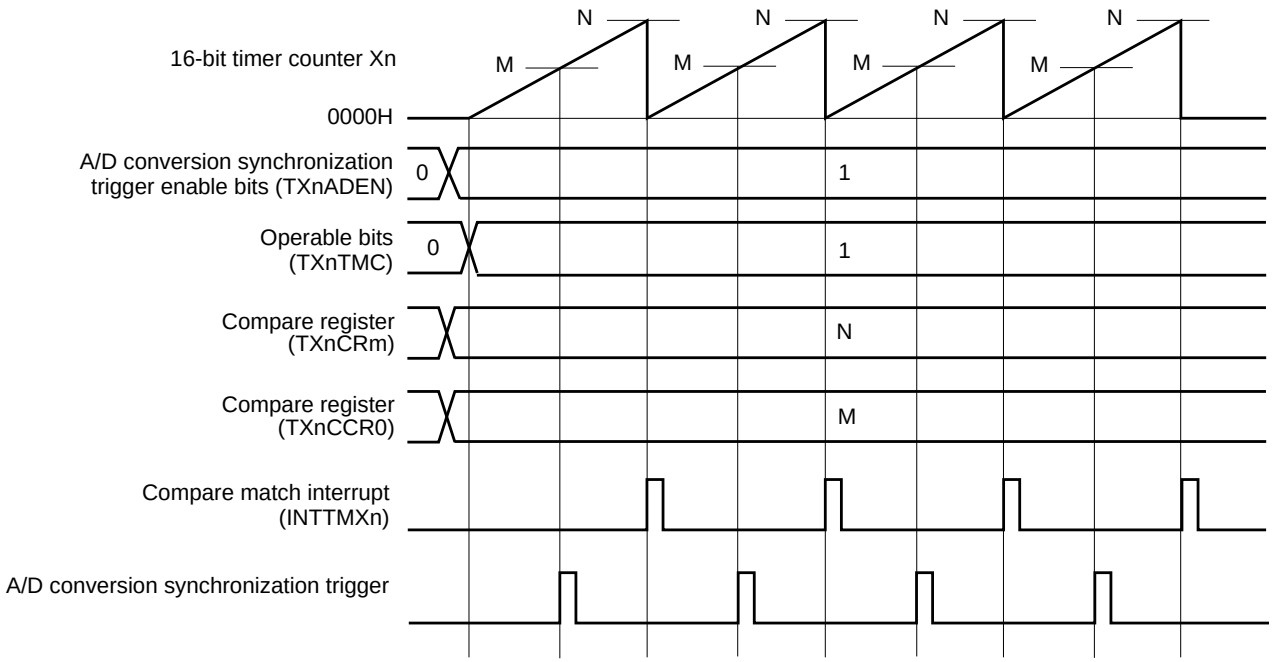
- TXnCCR0 < TXnCRm

If this relationship is not satisfied, the A/D conversion trigger is not generated.

**Remarks 1.** For details of the A/D conversion in combination with 16 bit timer X0 or X1, refer to **11.4.2 Basic operation of A/D converter (timer trigger mode)**.

**2.** m = 1, 3
n = 0, 1

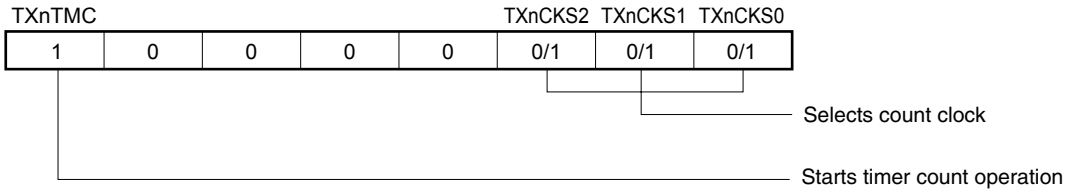
**Figure 6-21. Basic Timing Example of A/D Conversion Start Timing Signal Output**



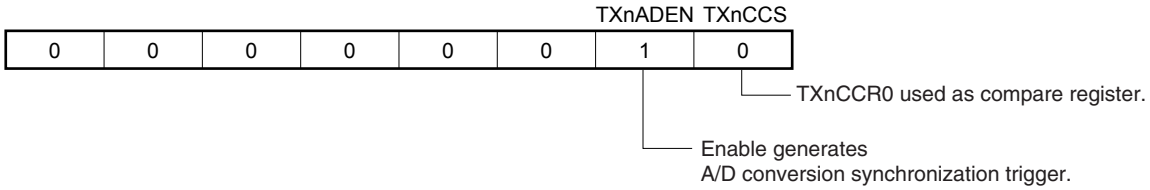

**Remark** m = 1, 3
n = 0, 1

**Figure 6-22. Example of Register Settings for A/D Conversion Start Timing Signal Output**

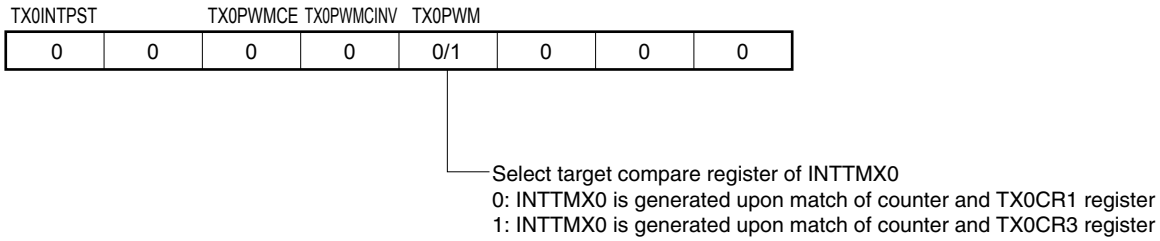
**(a) 16-bit timer Xn operation control register 0 (TXnCTL0)**



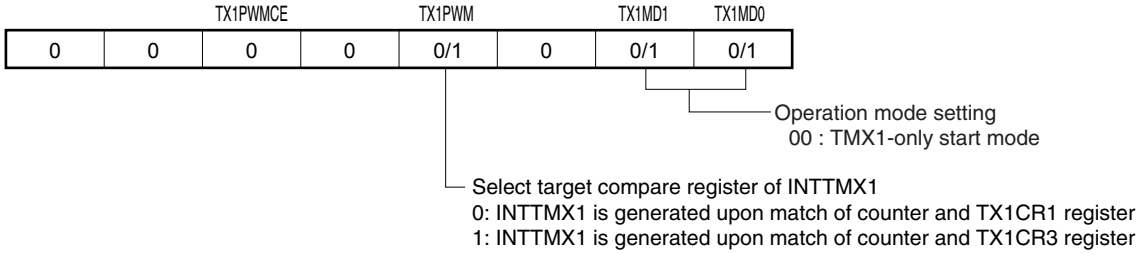

**(b) 16-bit timer Xn operation control register 2 (TXnCTL2)**




**(c) 16-bit timer X0 operation control register 1 (TX0CTL1)**




**(d) 16-bit timer X1 operation control register 1 (TX1CTL1)**




**(e) 16-bit timer Xn compare register m (TXnCRm)**

If N is set to TXnCRm, the A/D conversion synchronization trigger generation period is as follows.

- The A/D conversion synchronization trigger generation period = (N + 1) × Count clock cycle

Setting TXnCRm to 0000H is prohibited.

**(f) 16-bit timer Xn capture/compare register 0 (TXnCCR0)**

If M is set to TXnCCR0, the A/D conversion synchronization trigger is generated at a time later than counting 0000H only for M.

**Remarks 1.** m = 1, 3
n = 0, 1

**2.** For details of A/D conversion in combination with 16 bit timer X0 or X1, refer to **11.4.2 Basic operation of A/D converter (timer trigger mode)**.

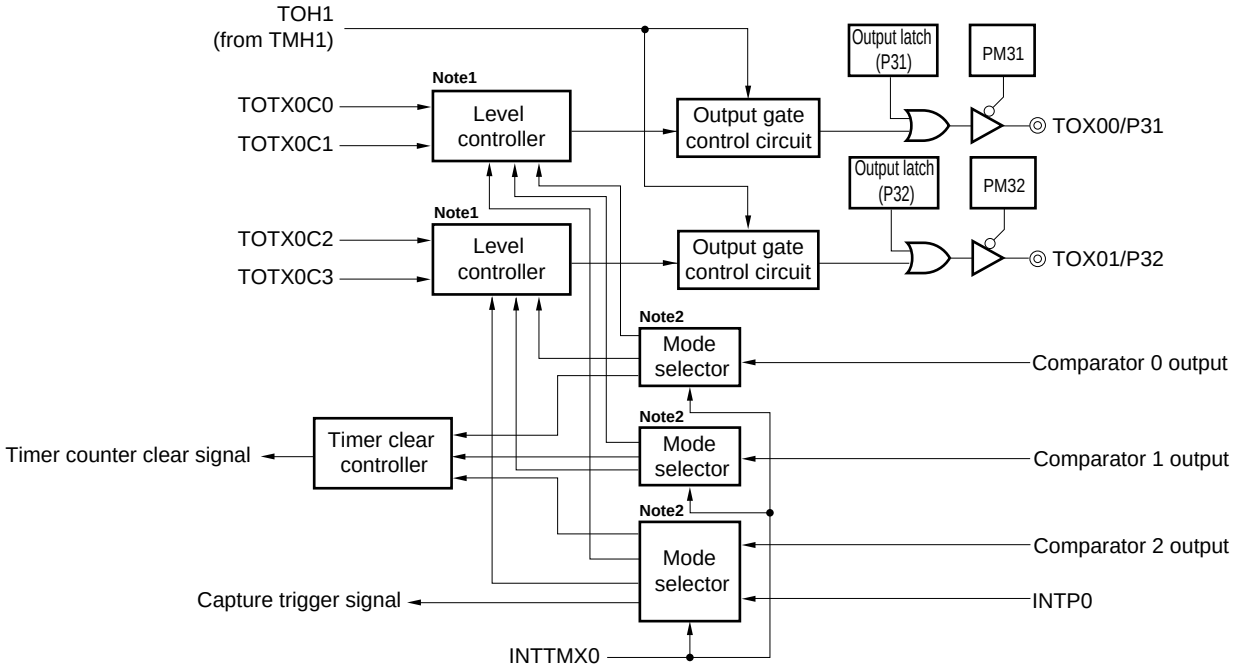
## 6.6 Interlocking Function with Comparator or INTP0

16-bit timers X0 and X1 can control PWM waveforms by interlocking with the output of comparators 0 to 2 or the INTP0 input signal, without involving the CPU.

TMX0 and TMX1, comparators 0 to 2, and INTP0 can be combined as follows.

- 16-bit timer X0 (TMX0-Only operation mode, synchronous start mode): CMP0, CMP1, CMP2, INTP0
- 16-bit timer X1 (TMX1-Only operation mode, synchronous start mode): CMP0, CMP1
- 16-bit timers X0 and X1 (synchronous start/clear mode): CMP2, INTP0

**Figure 6-43. Block Diagram of 16-Bit Timer X0 Output Configuration**

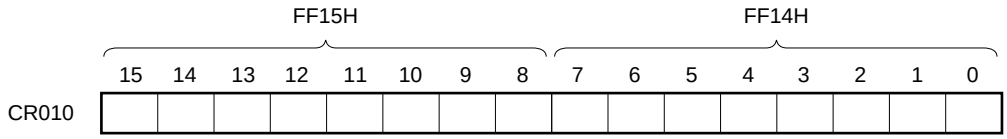



**Notes 1.** Timer output is controlled by the level controller according to the value of the compare register and the mode selector output.

**2.** Resetting timer output (interlocking modes 1 and 3) and clearing the timer counter (interlocking modes 1 and 2) are controlled by the mode selector according to the comparator output or INTP0 input.

**Figure 7-4. Format of 16-Bit Timer Capture/Compare Register 010 (CR010)**

Address: FF14H, FF15H After reset: 0000H R/W




**(i) When CR010 is used as a compare register**

The value set in CR010 is constantly compared with the TM00 count value, and an interrupt request signal (INTTM010) is generated if they match.

**Caution CR010 does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.**

**(ii) When CR010 is used as a capture register**

The count value of TM00 is captured to CR010 when a capture trigger is input.

It is possible to select the valid edge of the TI000 pin as the capture trigger. The TI000 pin valid edge is set by PRM00.

**(iii) Setting range when CR000 or CR010 is used as a compare register**

When CR000 or CR010 is used as a compare register, set it as shown below.

| Operation | CR000 Register Setting Range | CR010 Register Setting Range |
|---|---|---|
| Operation as interval timer | 0000H < N ≤ FFFFH | 0000H[Note 2] ≤ M ≤ FFFFH<br>Normally, this setting is not used. Mask the match interrupt signal (INTTM010). |
| Operation as square-wave output[Note 1] | | |
| Operation as external event counter | | |
| Operation in the clear & start mode entered by TI000 pin valid edge input | 0000H[Note 2] ≤ N ≤ FFFFH | 0000H[Note 2] ≤ M ≤ FFFFH |
| Operation as free-running timer | | |
| Operation as PPG output[Note 1] | M < N ≤ FFFFH | 0000H[Note 2] ≤ M < N |
| Operation as one-shot pulse output[Note 1] | 0000H[Note 2] ≤ N ≤ FFFFH (N ≠ M) | 0000H[Note 2] ≤ M ≤ FFFFH (M ≠ N) |

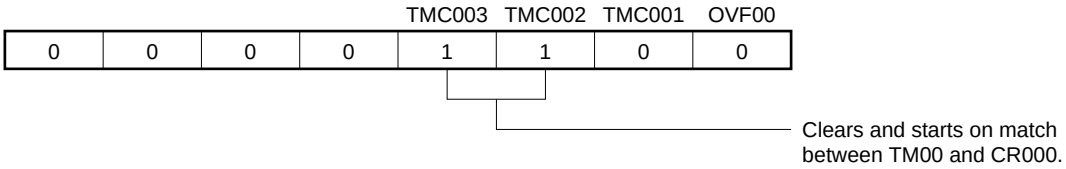
**Notes 1.** 78K0/IB2 (30 pins) only

**2.** When 0000H is set, a match interrupt immediately after the timer operation does not occur and timer output is not changed, and the first match timing is as follows. A match interrupt occurs at the timing when the timer counter (TM00 register) is changed from 0000H to 0001H.

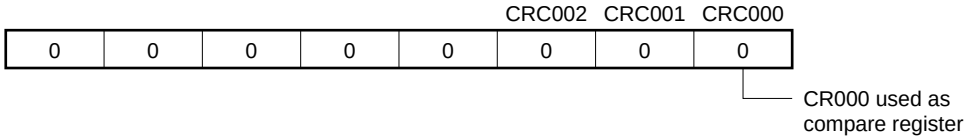
- When the timer counter is cleared due to overflow
- When the timer counter is cleared due to TI000 pin valid edge (when clear & start mode is entered by TI000 pin valid edge input)
- When the timer counter is cleared due to compare match (when clear & start mode is entered by match between TM00 and CR000 (CR000 = other than 0000H, CR010 = 0000H))

**Figure 7-21. Example of Register Settings in External Event Counter Mode (1/2)**

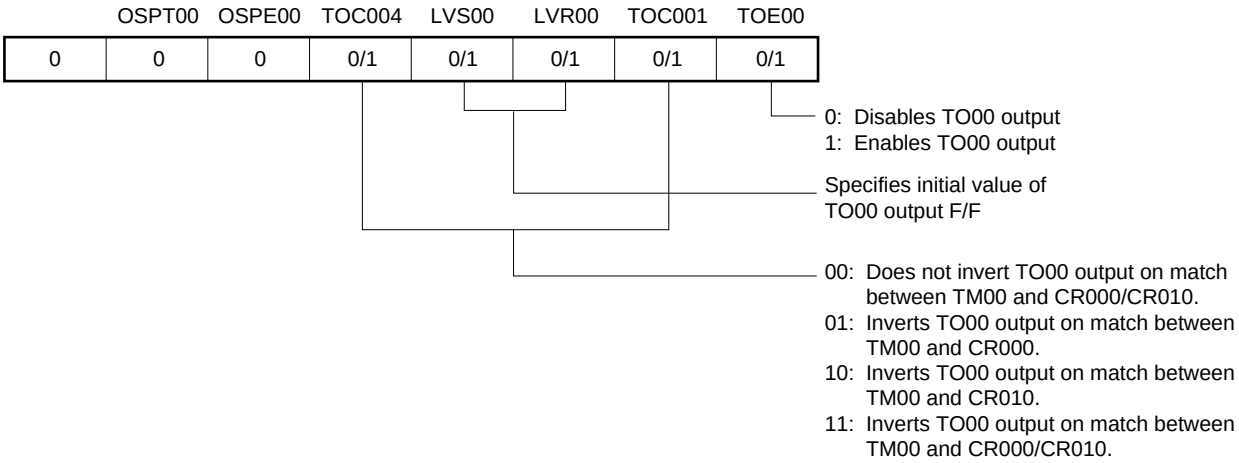
**(a) 16-bit timer mode control register 00 (TMC00)**




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




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




**(d) Prescaler mode register 00 (PRM00)**

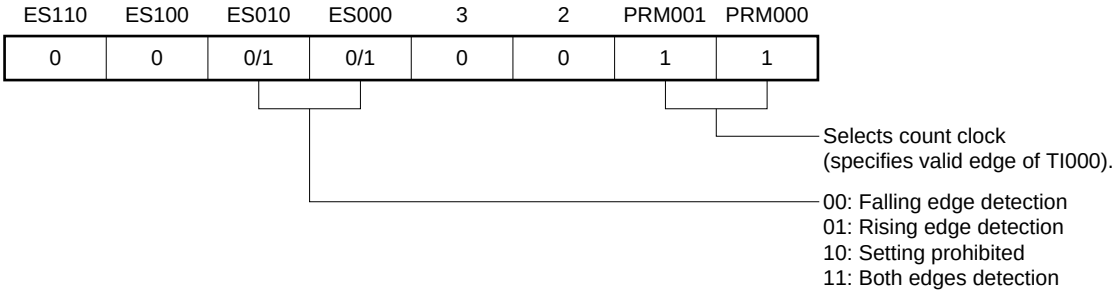


**Figure 7-52. Example of Register Settings for Pulse Width Measurement (1/2)**

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

| | | | | TMC003 | TMC002 | TMC001 | OVF00 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0/1 | 0/1 | 0 | 0 |

01: Free running timer mode
10: Clear and start mode entered by valid edge of TI000 pin.

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

| | | | | | CRC002 | CRC001 | CRC000 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0/1 | 1 |

1: CR000 used as capture register

0: TI010 pin is used as capture trigger of CR000.
1: Reverse phase of TI000 pin is used as capture trigger of CR000.

1: CR010 used as capture register

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

| | OSPT00 | OSPE00 | TOC004 | LVS00 | LVR00 | TOC001 | TOE00 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(d) Prescaler mode register 00 (PRM00)**

| ES110 | ES100 | ES010 | ES000 | 3 | 2 | PRM001 | PRM000 |
|---|---|---|---|---|---|---|---|
| 0/1 | 0/1 | 0/1 | 0/1 | 0 | 0 | 0/1 | 0/1 |

Selects count clock
(setting valid edge of TI000 is prohibited)

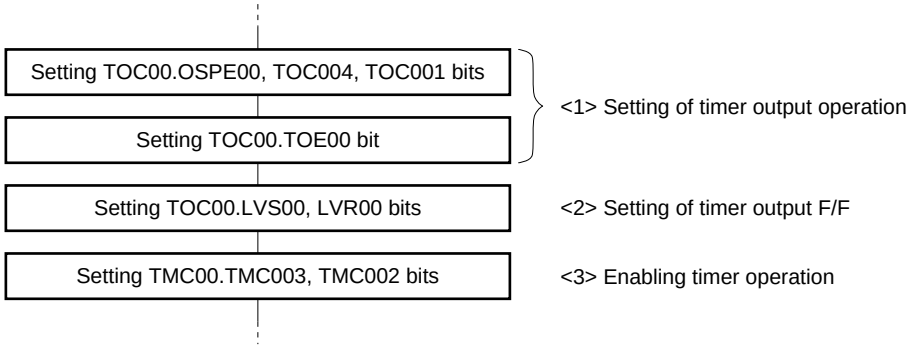
00: Falling edge detection
01: Rising edge detection
10: Setting prohibited
11: Both edges detection
(setting when CRC001 = 1 is prohibited)

00: Falling edge detection
01: Rising edge detection
10: Setting prohibited
11: Both edges detection

**(2) Setting LVS00 and LVR00**

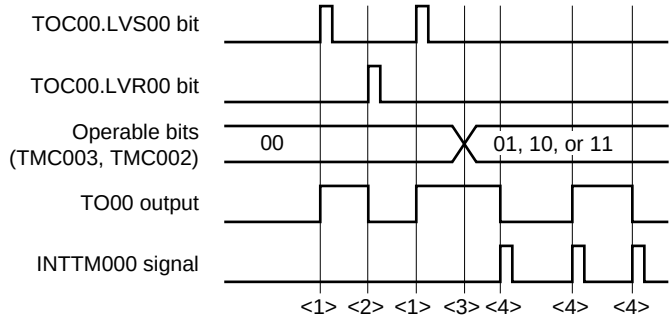
Set LVS00 and LVR00 using the following procedure.

**Figure 7-54. Example of Flow for Setting LVS00 and LVR00 Bits**




**Caution Be sure to set LVS00 and LVR00 following steps <1>, <2>, and <3> above.**
**Step <2> can be performed after <1> and before <3>.**

**Figure 7-55. Timing Example of LVR00 and LVS00**




<1> The TO00 output goes high when LVS00 and LVR00 = 10.

<2> The TO00 output goes low when LVS00 and LVR00 = 01 (the pin output remains unchanged from the high level even if LVS00 and LVR00 are cleared to 00).

<3> The timer starts operating when TMC003 and TMC002 are set to 01, 10, or 11. Because LVS00 and LVR00 were set to 10 before the operation was started, the TO00 output starts from the high level. After the timer starts operating, setting LVS00 and LVR00 is prohibited until TMC003 and TMC002 = 00 (disabling the timer operation).

<4> The TO00 output level is inverted each time an interrupt signal (INTTM000) is generated.

# CHAPTER 8 8-BIT TIMER/EVENT COUNTER 51

## 8.1 Functions of 8-Bit Timer/Event Counter 51

8-bit timer/event counter 51 is mounted onto all 78K0/Ix2 microcontroller products.
8-bit timer/event counter 51 has the following functions.

- Interval timer
- External event counter

## 8.2 Configuration of 8-Bit Timer/Event Counter 51

8-bit timer/event counter 51 includes the following hardware.

**Table 8-1. Configuration of 8-Bit Timer/Event Counter 51**

| Item | Configuration |
|---|---|
| Timer register | 8-bit timer counter 51 (TM51) |
| Timer input | TI51 |
| Register | 8-bit timer compare register 51 (CR51) |
| Control registers | Timer clock selection register 51 (TCL51)<br>8-bit timer mode control register 51 (TMC51)<br>Port alternate switch control register (MUXSEL)<br>Port mode register 0 (PM0) or port mode register 3 (PM3)Note<br>Port register 0 (P0) or port register 3 (P3)Note |

**Note** 78K0/IY2, 78K0/IB2 (30 pins): PM3
78K0/IA2, 78K0/IB2 (32 pins): PM0 or PM3

Figure 8-1 shows the block diagram of 8-bit timer/event counter 51.

## 10.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

**Table 10-1. Configuration of Watchdog Timer**

| Item | Configuration |
|---|---|
| Control register | Watchdog timer enable register (WDTE) |

How the counter operation is controlled, overflow time, and window open period are set by the option byte.

**Table 10-2. Setting of Option Bytes and Watchdog Timer**

| Setting of Watchdog Timer | Option Byte (0080H) |
|---|---|
| Window open period | Bits 6 and 5 (WINDOW1, WINDOW0) |
| Controlling counter operation of watchdog timer | Bit 4 (WDTON) |
| Overflow time of watchdog timer | Bits 3 to 1 (WDCS2 to WDCS0) |

**Remark** For the option byte, refer to **CHAPTER 24 OPTION BYTE**.

**Figure 10-1. Block Diagram of Watchdog Timer**

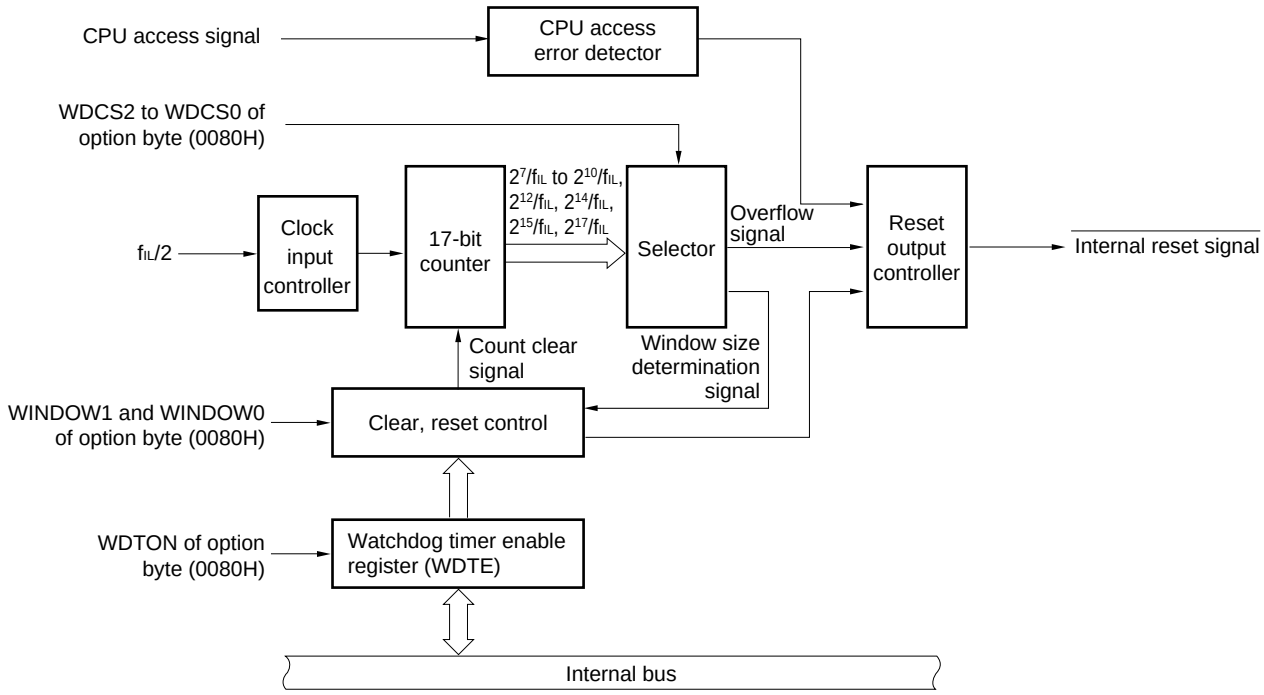


**Figure 11-9. Format of 8-bit A/D conversion result register L for TMXn synchronization (ADCRXnL)**

Address: FF16H (ADCRX0L), FF18H (ADCRX1L) After reset: 00H R

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADCRXnL (n = 0, 1) | | | | | | | | |

**Cautions 1. When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration registers 0, 1 (ADPC0, ADPC1), the contents of ADCRXnL may become undefined. Read the conversion result following conversion completion before writing to ADM0, ADS, ADPC0, and ADPC1. Using timing other than the above may cause an incorrect conversion result to be read.**

**2. If data is read from ADCRXnL, a wait cycle is generated. Do not read data from ADCRXnL when the peripheral hardware clock ($f_{PRS}$) is stopped. For details, refer to CHAPTER 31 CAUTIONS FOR WAIT.**

**Remark** n = 0, 1

### (7) Analog input channel specification register (ADS)

ADS specifies the input channel of the analog voltage to be A/D converted and sets the A/D conversion start method.

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

Reset signal generation clears ADS to 00H.

**Remark** A/D converter analog input pins differ depending on products.

- 78K0/IY2: ANI0, ANI1, ANI3 to ANI5
- 78K0/IA2: ANI0 to ANI5
- 78K0/IB2: ANI0 to ANI8

**Figure 11-10. Format of Analog Input Channel Specification Register (ADS) (1/2)**

#### (1) 78K0/IY2, 78K0/IA2

Address: FF0EH After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | 3 | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| ADS | V12SEL | ADOAS | ADTRG1 | ADTRG0 | 0 | ADS2 | ADS1 | ADS0 |

#### (2) 78K0/IB2

Address: FF0EH After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| ADS | V12SEL | ADOAS | ADTRG1 | ADTRG0 | ADS3 | ADS2 | ADS1 | ADS0 |

**Cautions 3. Set POWER6 = 1 and then set TXE6 = 1 (transmission) or RXE6 = 1 (reception) to start communication.**

**4. TXE6 and RXE6 are synchronized by the base clock ($f_{XCLK6}$) set by CKSR6. To enable transmission or reception again, set TXE6 or RXE6 to 1 at least two clocks of the base clock after TXE6 or RXE6 has been cleared to 0. If TXE6 or RXE6 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.**

**5. Set transmit data to TXB6 at least one base clock ($f_{XCLK6}$) after setting TXE6 = 1.**

**6. If data is continuously transmitted, the communication timing from the stop bit to the next start bit is extended two operating clocks of the macro. However, this does not affect the result of communication because the reception side initializes the timing when it has detected a start bit. Do not use the continuous transmission function if the interface is used in LIN communication operation.**

**Remark** LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

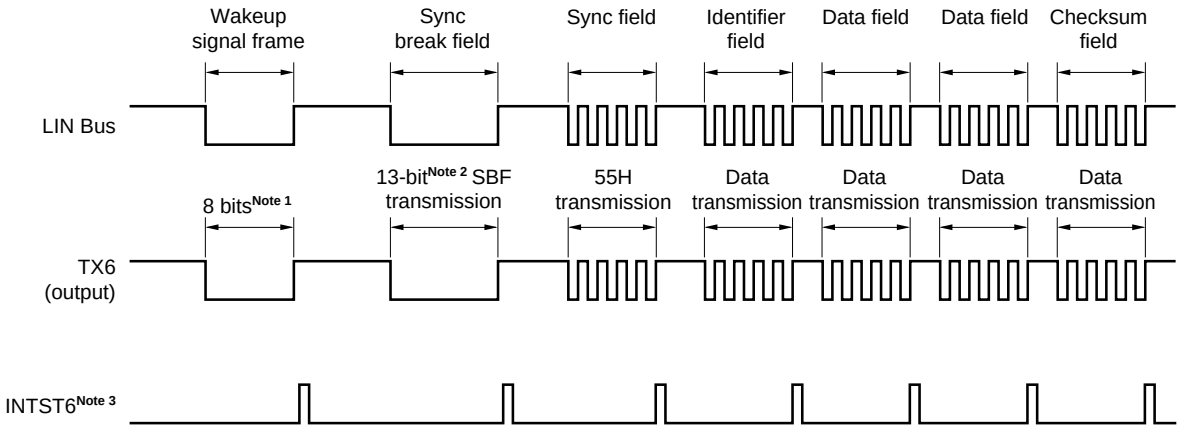
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±15% or less.

Figures 14-1 and 14-2 outline the transmission and reception operations of LIN.

**Figure 14-1. LIN Transmission Operation**




**Notes 1.** The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.

**2.** The sync break field is output by hardware. The output width is the bit length set by bits 4 to 2 (SBL62 to SBL60) of asynchronous serial interface control register 6 (ASICL6) (refer to **14.4.2 (2) (h) SBF transmission**).

**3.** INTST6 is output on completion of each transmission. It is also output when SBF is transmitted.

**Remark** The interval between each field is controlled by software.

**(5) IICA low-level width setting register (IICWL)**

This register is used to set the low-level width of the SCLA0 pin signal that is output by serial interface IICA being in master mode.

This register can be set by an 8-bit memory manipulation instruction.

Set this register while operation of $I^2C$ is disabled (bit 7 (IICE0) of the IICA control register 0 (IICACTL0) is 0).

Reset signal generation sets this register to FFH.

**Figure 15-9. Format of IICA Low-Level Width Setting Register (IICWL)**

Address: FFADH After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IICWL | | | | | | | | |

**(6) IICA high-level width setting register (IICWH)**

This register is used to set the high-level width of the SCLA0 pin signal that is output by serial interface IICA being in master mode.

This register can be set by an 8-bit memory manipulation instruction.

Set this register while operation of $I^2C$ is disabled (bit 7 (IICE0) of the IICA control register 0 (IICACTL0) is 0).

Reset signal generation sets this register to FFH.

**Figure 15-10. Format of IICA High-Level Width Setting Register (IICWH)**

Address: FFAEH After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IICWH | | | | | | | | |

**Remark** For how to set the transfer clock by using the IICWL and IICWH registers, see **15.4.2 Setting transfer clock by using IICWL and IICWH registers**.

**(7) Port input mode register 6 (PIM6)**

This register sets the input buffer of P60 and P61 in 1-bit units. When using an input compliant with the SMBus specifications in $I^2C$ communication, set PIM60 and PIM61 to 1.

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

Reset signal generation clears this register to 00H.

**Figure 15-11. Format of Port Input Mode Register 6 (PIM6)**

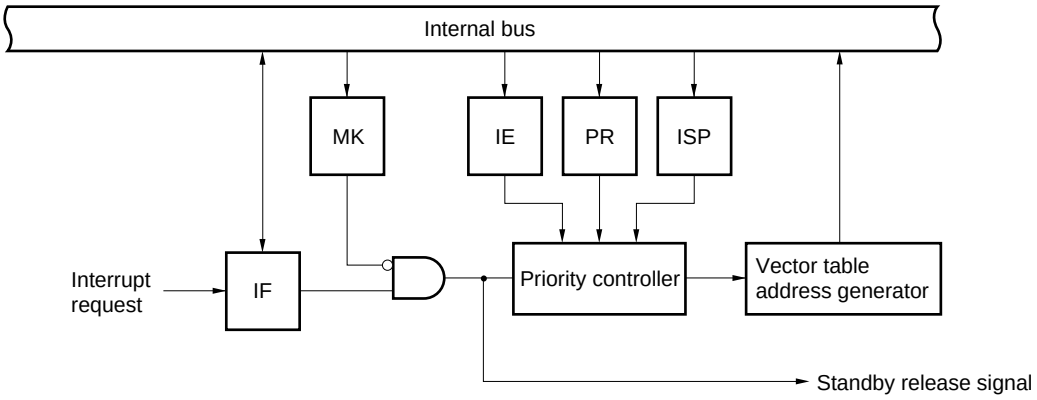
Address: FF3EH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIM6 | 0 | 0 | 0 | 0 | 0 | 0 | PIM61 | PIM60 |

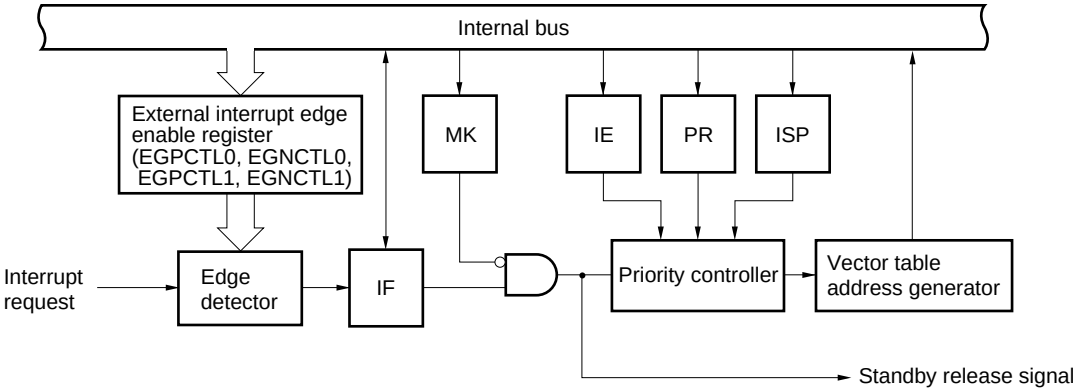
| PIM6n | P6n pin input buffer selection (n = 0, 1) |
|---|---|
| 0 | Normal input (Schmitt) buffer |
| 1 | SMBus input buffer |

**Figure 18-1. Basic Configuration of Interrupt Function (1/2)**

**(A) Internal maskable interrupt**




**(B) External maskable interrupt (INTPm, INTCMP0 to INTCMP2)**




**Remark** m = 0, 2 to 4: 78K0/IY2, 78K0/IA2
m = 0 to 5: 78K0/IB2 (30 pins)
m = 0, 2 to 5: 78K0/IB2 (32 pins)

IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

**(1) Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H)**

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

IF0L, IF0H, IF1L, and IF1H are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, and IF1L and IF1H are combined to form 16-bit registers IF0 and IF1, they are set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Cautions 1. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.**

**2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).**

**If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.**

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

**In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.**

**Caution The pins mounted depend on the product. Refer to Caution 2 at the beginning of this chapter.**

**($T_A$ = –40 to +105°C, 2.7 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Output voltage, low | $V_{OL1}$ | P00 to P02, P30 to P37 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL1}$ = 8.5 mA | | | 0.7 | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OL1}$ = 5.0 mA | | | 0.7 | V |
| | $V_{OL2}$ | P20 to P27, P70 | $AV_{REF}$ = $V_{DD}$, $I_{OL2}$ = 0.4 mA | | | 0.4 | V |
| | $V_{OL3}$ | P60, P61 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL1}$ = 15.0 mA | | | 2.0 | V |
| | | | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL1}$ = 5.0 mA | | | 0.4 | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OL1}$ = 5.0 mA | | | 0.6 | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OL1}$ = 3.0 mA | | | 0.4 | V |
| Input leakage current, high | $I_{LIH1}$ | P00 to P02, P30 to P37, P60, P61, P125/$\overline{\text{RESET}}$ | $V_I$ = $V_{DD}$ | | | 3 | μA |
| | $I_{LIH2}$ | P20 to P27, P70 | $V_I$ = $AV_{REF}$ = $V_{DD}$ | | | 3 | μA |
| | $I_{LIH3}$ | P121, P122 | $V_I$ = $V_{DD}$ I/O port mode | | | 3 | μA |
| | | X1, X2 | OSC mode | | | 20 | μA |
| Input leakage current, low | $I_{LIL1}$ | P00 to P02, P30 to P37, P60, P61, P125/$\overline{\text{RESET}}$ | $V_I$ = $V_{SS}$ | | | –3 | μA |
| | $I_{LIL2}$ | P20 to P27, P70 | $V_I$ = $V_{SS}$, $AV_{REF}$ = $V_{DD}$ | | | –3 | μA |
| | $I_{LIL3}$ | P121, P122 | $V_I$ = $V_{SS}$ I/O port mode | | | –3 | μA |
| | | X1, X2 | OSC mode | | | –20 | μA |
| Pull-up resistor | $R_{PLU1}$ | P00 to P02, P30 to P37, P60, P61 | $V_I$ = $V_{SS}$ | 10 | 20 | 100 | kΩ |
| | $R_{PLU2}$ | P125/$\overline{\text{RESET}}$ | | 75 | 150 | 300 | kΩ |

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

## 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** |