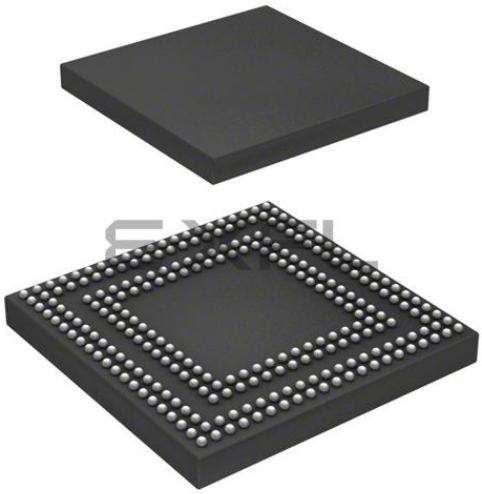


Welcome to E-XFL.COM

## Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

## Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | ARM® Cortex®-A9 |
| Number of Cores/Bus Width | 1 Core, 32-Bit |
| Speed | 400MHz |
| Co-Processors/DSP | Multimedia; NEON™ MPE |
| RAM Controllers | SDRAM, SRAM |
| Graphics Acceleration | Yes |
| Display & Interface Controllers | DVD, VDC |
| Ethernet | 10/100Mbps (1), 100Mbps (1) |
| SATA | - |
| USB | USB 2.0 (2) |
| Voltage - I/O | 1.2V, 3.3V |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Security Features | - |
| Package / Case | 256-LFBGA |
| Supplier Device Package | 256-LFBGA (11x11) |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/r7s721000vcbg-ac0 |

## NOTES FOR CMOS DEVICES

### ① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between $V_{IL}$ (MAX) and $V_{IH}$ (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between $V_{IL}$ (MAX) and $V_{IH}$ (MIN).

### ② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to $V_{DD}$ or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

### ③ PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

### ④ STATUS BEFORE INITIALIZATION

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

### ⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

### ⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

### 3.2.2 System register set

System registers control the status of the CPU and hold interrupt information.

Read from and write to system registers are performed by setting the system register numbers shown below with the system register load/store instructions (LDSR, STSR instructions).

**Table 3-2. System Register Numbers**

| System Register No. | System Register Name | Operand Specification Enabled | |
|---|---|---|---|
| | | LDSR Instruction | STSR Instruction |
| 0 | Interrupt status saving register (EIPC)[Note 1] | Yes | Yes |
| 1 | Interrupt status saving register (EIPSW)[Note 1] | Yes | Yes |
| 2 | NMI status saving register (FEPC) | Yes | Yes |
| 3 | NMI status saving register (FEPSW) | Yes | Yes |
| 4 | Interrupt source register (ECR) | No | Yes |
| 5 | Program status word (PSW) | Yes | Yes |
| 6 to 15 | Reserved numbers for future function expansion (The operation is not guaranteed if accessed.) | No | No |
| 16 | CALLT execution status saving register (CTPC) | Yes | Yes |
| 17 | CALLT execution status saving register (CTPSW) | Yes | Yes |
| 18 | Exception/debug trap status saving register (DBPC) | Yes[Note 2] | Yes[Note 2] |
| 19 | Exception/debug trap status saving register (DBPSW) | Yes[Note 2] | Yes[Note 2] |
| 20 | CALLT base pointer (CTBP) | Yes | Yes |
| 21 to 31 | Reserved numbers for future function expansion (The operation is not guaranteed if accessed.) | No | No |

**Notes 1.** Since only one set of these registers is available, the contents of this register must be saved by the program when multiple interrupt servicing is enabled.

**2.** Can be accessed only after the DBTRAP instruction or illegal opcode is executed and before the DBRET instruction is executed.

**Caution Even if bit 0 of EIPC, FEPC, or CTPC is set (1) by the LDSR instruction, bit 0 is ignored during return with the RETI instruction following interrupt servicing (because bit 0 of PC is fixed to 0). When setting a value to EIPC, FEPC, and CTPC, set an even number (bit 0 = 0).**

### (c) Port 2 mode control register (PMC2)

After reset: 00H R/W Address: FFFFF444H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PMC2 | PMC27 | PMC26 | PMC25 | PMC24 | PMC23 | PMC22 | PMC21 | PMC20 |

| PMC27 | Specification of operating mode of P27 pin |
|---|---|
| 0 | I/O port |
| 1 | TOP31 output |

| PMC26 | Specification of operating mode of P26 pin |
|---|---|
| 0 | I/O port |
| 1 | TOQ10 output |

| PMC25 | Specification of operating mode of P25 pin |
|---|---|
| 0 | I/O port |
| 1 | TOQ1B3 output |

| PMC24 | Specification of operating mode of P24 pin |
|---|---|
| 0 | I/O port |
| 1 | TOQ1T3 output |

| PMC23 | Specification of operating mode of P23 pin |
|---|---|
| 0 | I/O port |
| 1 | TOQ1B2 output |

| PMC22 | Specification of operating mode of P22 pin |
|---|---|
| 0 | I/O port |
| 1 | TOQ1T2 output |

| PMC21 | Specification of operating mode of P21 pin |
|---|---|
| 0 | I/O port |
| 1 | TOQ1B1 output |

| PMC20 | Specification of operating mode of P20 pin |
|---|---|
| 0 | I/O port |
| 1 | TOQ1T1 output |

# CHAPTER 6 16-BIT TIMER/EVENT COUNTER P (TMP)

Timer P (TMP) is a 16-bit timer/event counter.
The V850ES/IE2 incorporates TMP0 to TMP3.

## 6.1 Overview

The TMPn of channels are outlined below (n = 0 to 3).

**Table 6-1. TMPn Overview**

| Item | TMP0 | TMP1 | TMP2 | TMP3 |
|---|---|---|---|---|
| Clock selection | 8 ways | 8 ways | 8 ways | 8 ways |
| Capture trigger input pin | 2 | None | 2 | None |
| External event count input pin | 1 | None | 1 | None |
| External trigger input pin | 1 | None | 1 | None |
| Timer counter | 1 | 1 | 1 | 1 |
| Capture/compare register | 2 | 2 Note | 2 | 2 Note |
| Capture/compare match interrupt request signal | 2 | 2 Note | 2 | 2 Note |
| <R> Overflow interrupt request signal | 1 | 1 | 1 | 1 |
| Timer output pin | 2 | None | 1 | 1 |

**Note** Compare function only

## 6.2 Functions

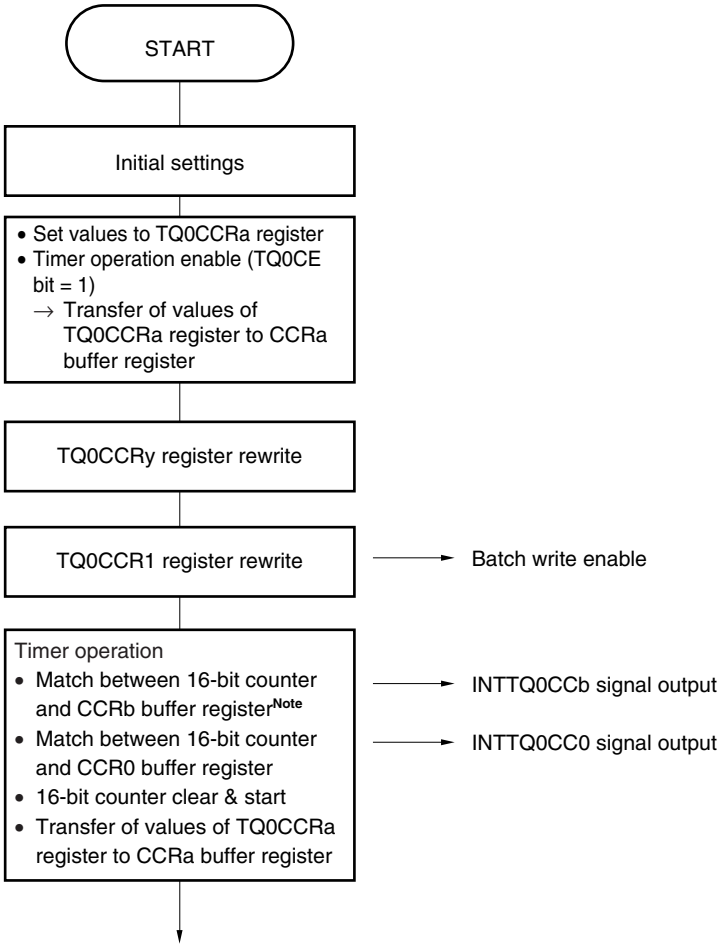
The functions of TMPn that can be realized differ from one channel to another, as shown in the table below (n = 0 to 3).

**Table 6-2. TMPn Functions**

| Function | TMP0 | TMP1 | TMP2 | TMP3 |
|---|---|---|---|---|
| Interval timer | √ | √ | √ | √ |
| External event counter | √ | × | √ | × |
| External trigger pulse output | √ | × | √ | √ Note |
| One-shot pulse output | √ | × | √ | √ Note |
| PWM output | √ | × | √ | √ |
| Free-running timer | √ | √ | √ | √ |
| Pulse width measurement | √ | × | √ | × |
| Timer tuning operation | × | √ (TMQ1) | × | × |

**Note** Realized by software trigger only. External trigger input cannot be used.

**Figure 7-6. Flowchart of Basic Operation for Batch Write**




**Note** The 16-bit counter is not cleared upon a match between the value of the 16-bit counter and the value of the CCRb buffer register. It is cleared upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.

**Caution Writing to the TQ0CCR1 register includes enabling of batch write. Thus, rewrite the TQ0CCR1 register after rewriting the TQ0CCR0, TQ0CCR2, and TQ0CCR3 registers.**

**Remarks 1.** The above flowchart illustrates an example of the operation in the PWM output mode.

**2.** a = 0 to 3
b = 1 to 3
y = 0, 2, 3

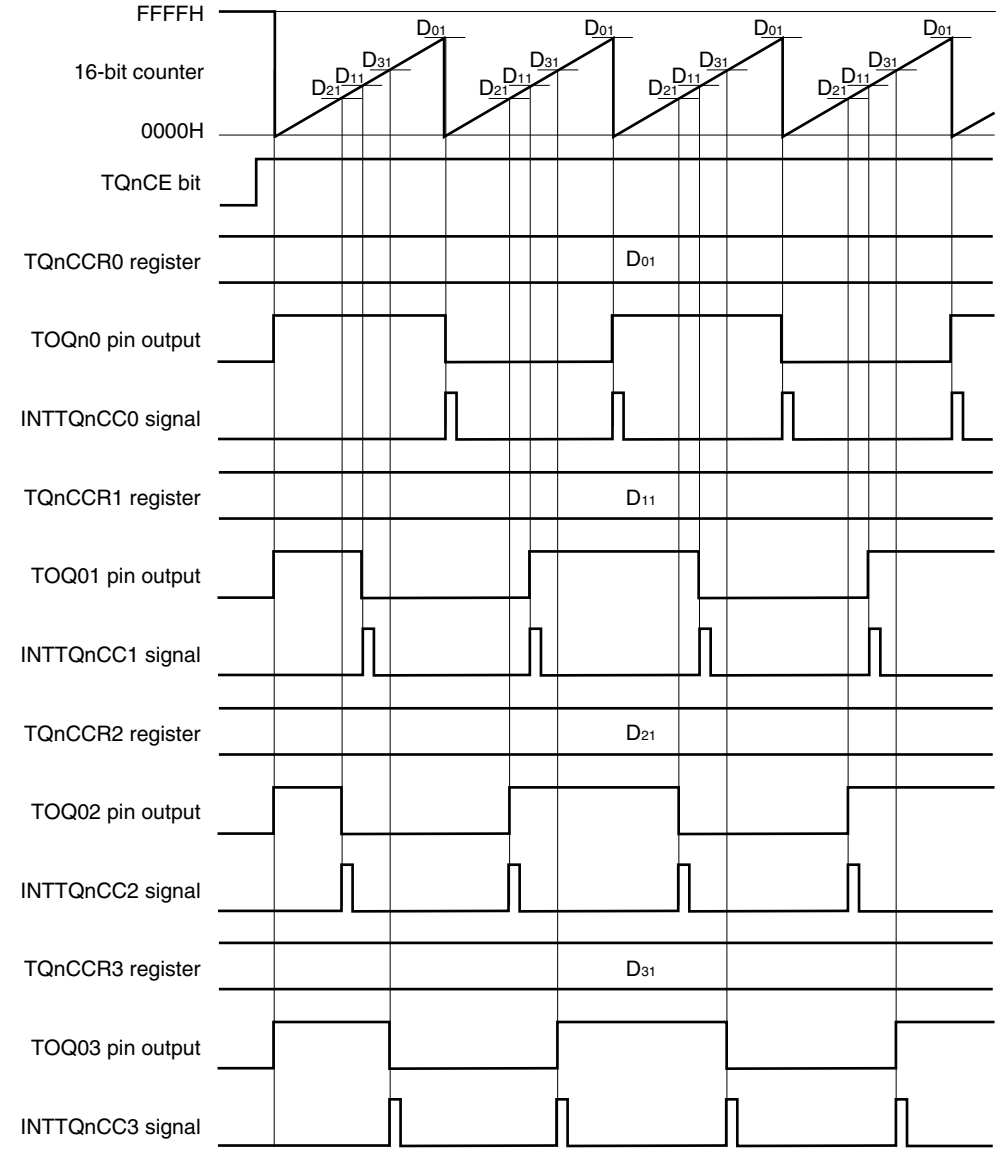
<R> When the TQnCCRb register is set to the same value as that of the TQnCCR0 register, the INTTQnCCb signal is generated at the same timing as the INTTQnCC0 signal is generated, and the TOQ0b pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOQ0b pin.

The following shows the operation when the TQnCCRb register is set to other than the value set in the TQnCCR0 register.

If the set value of the TQnCCRb register is less than the set value of the TQnCCR0 register, the INTTQnCCb signal is generated once per cycle. At the same time, the output of the TOQ0b pin is inverted.

<R> After outputting the short-width pulse first, the TOQ0b pin outputs a PWM waveform with a duty factor of 50%.

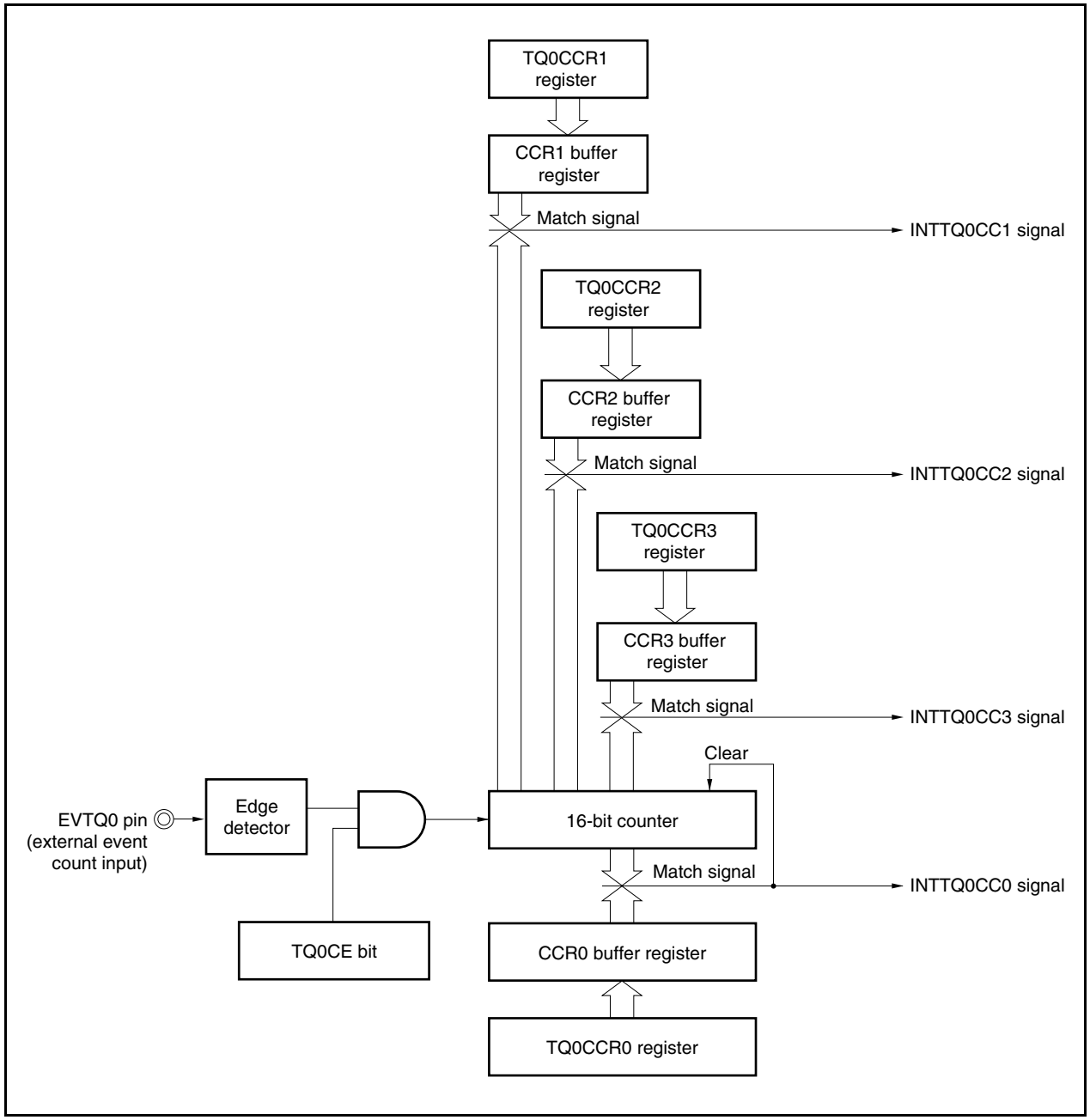
**Figure 7-13. Timing Chart When $D_{01} \geq D_{b1}$**




**Remark** n = 0, 1
b = 1 to 3

### (c) Operation of TQ0CCR1 to TQ0CCR3 registers

**Figure 7-19. Configuration of TQ0CCR1 to TQ0CCR3 Registers**



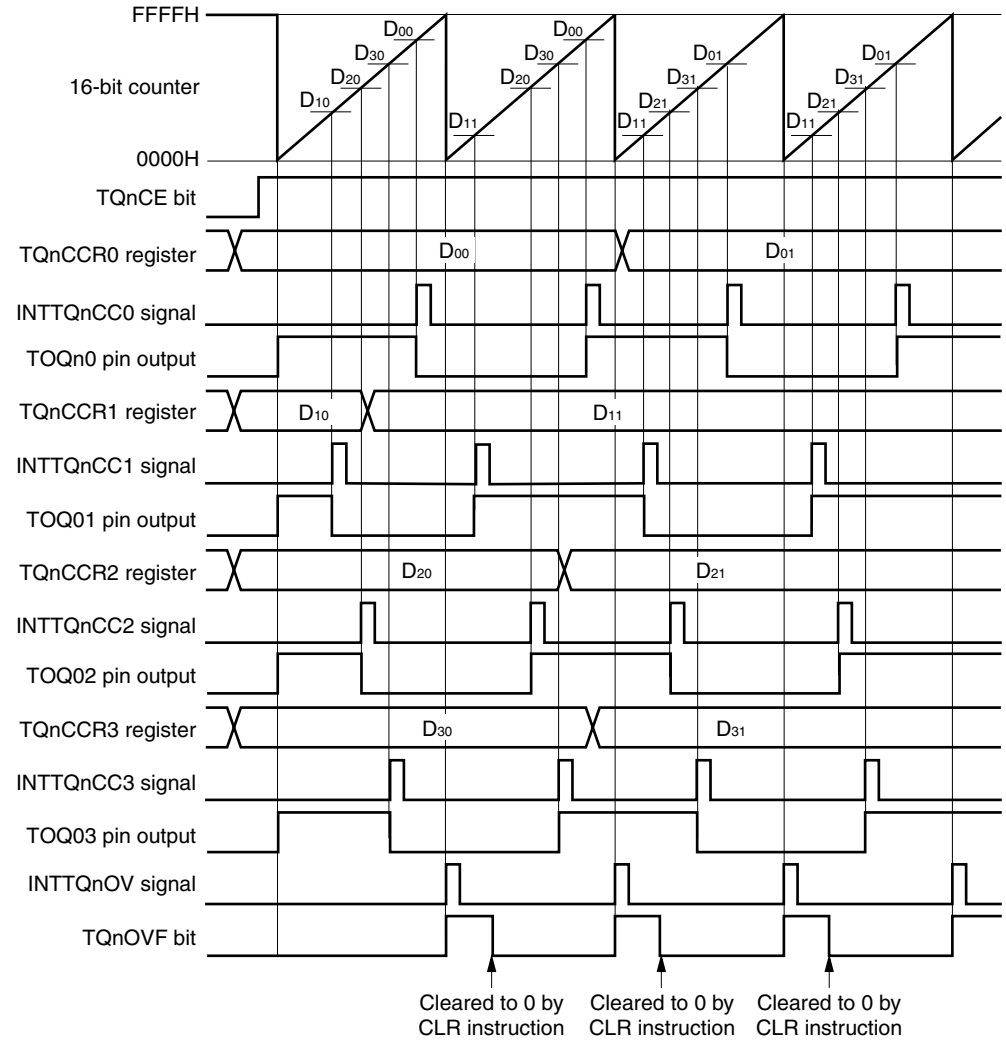
- Compare operation

  When the TQnCE bit is set to 1, 16-bit timer/event counter Q starts counting, and the output signals of the TOQ00 to TOQ03 and TOQ10 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TQnCCRa register, a compare match interrupt request signal (INTTQnCCa) is generated, and the output signals of the TOQ00 to TOQ03 and TOQ10 pins are inverted.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTQnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TQnOPT0.TQnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

  The TQnCCRa register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time by anytime write, and compared with the count value.

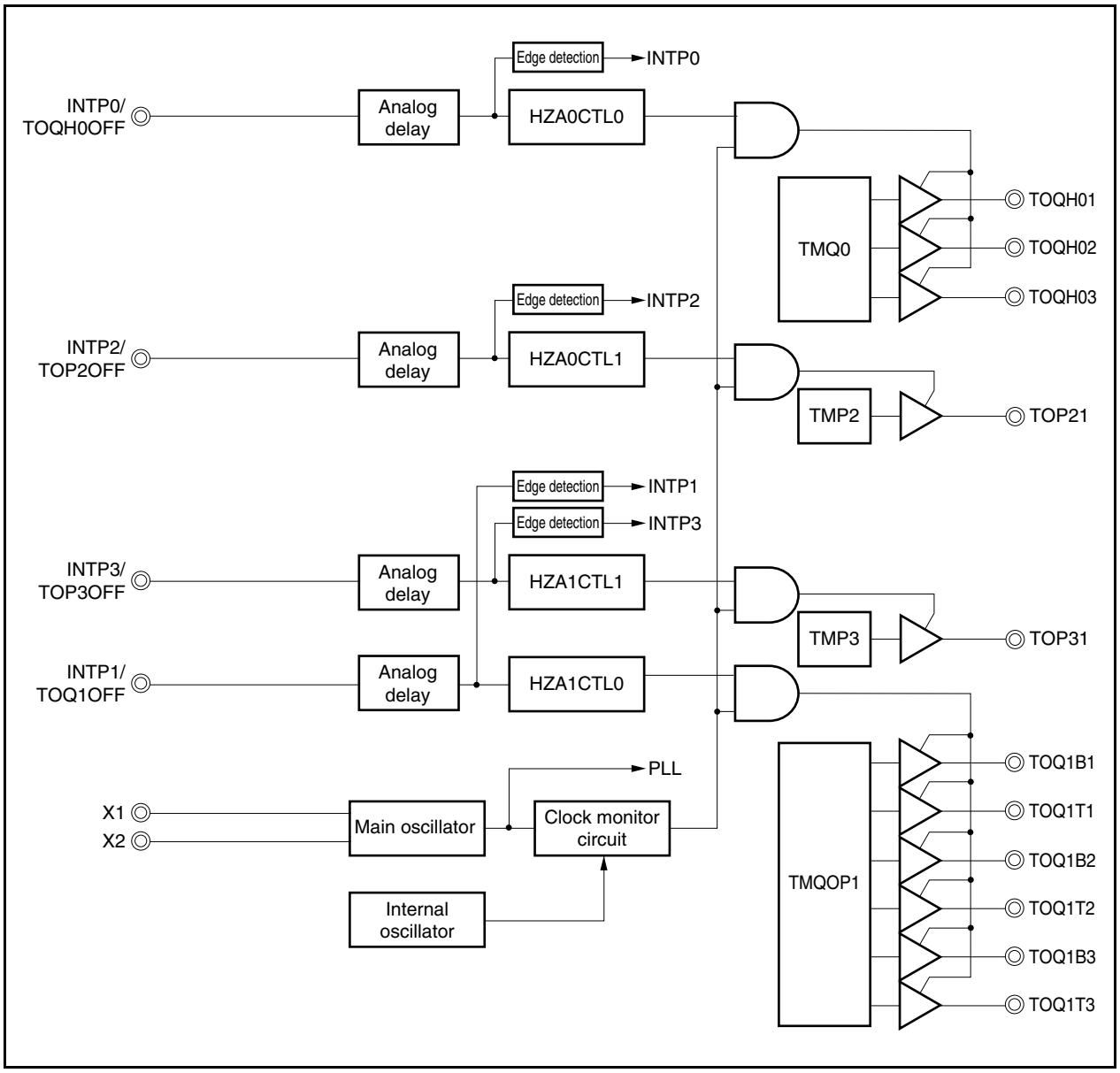
**Figure 7-35. Basic Timing in Free-Running Timer Mode (Compare Function)**




**Remark** n = 0, 1

a = 0 to 3

**Figure 9-4. High-Impedance Output Controller Configuration**



## 10.3 Control Registers

### (1) Watchdog timer mode register (WDTM)

The WDTM register sets the overflow time and operation clock of the watchdog timer.

This register can be read or written in 8-bit units. This register can be read any number of times, but can be written only once following reset release; it cannot then be written a second or subsequent time.

Reset sets this register to 67H.

After reset: 67H R/W Address: FFFFF6D0H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| WDTM | 0 | WDM1 | WDM0 | 0 | 0 | WDCS2 | WDCS1 | WDCS0 |

| WDM1 | WDM0 | Selection of operation mode of watchdog timer |
|---|---|---|
| 0 | 0 | Stop operation |
| 0 | 1 | Non-maskable interrupt request mode (generation of INTWDT signal) |
| 1 | × | Reset mode (generation of WDTRES signal) |

**Cautions 1. For details of the WDCS2 to WDCS0 bits, see Table 10-2 Overflow Time.**

**2. Be sure to clear bits 3, 4, and 7 to "0".**

**Table 10-2. Overflow Time**

| WDCS2 | WDCS1 | WDCS0 | Overflow Time | $f_{XX}$ = 20 MHz |
|---|---|---|---|---|
| 0 | 0 | 0 | $2^{18}/f_{XX}$ | 13.1 ms |
| 0 | 0 | 1 | $2^{19}/f_{XX}$ | 26.2 ms |
| 0 | 1 | 0 | $2^{20}/f_{XX}$ | 52.4 ms |
| 0 | 1 | 1 | $2^{21}/f_{XX}$ | 104.9 ms |
| 1 | 0 | 0 | $2^{22}/f_{XX}$ | 209.7 ms |
| 1 | 0 | 1 | $2^{23}/f_{XX}$ | 419.4 ms |
| 1 | 1 | 0 | $2^{24}/f_{XX}$ | 838.9 ms |
| 1 | 1 | 1 | $2^{25}/f_{XX}$ | 1677.7 ms |

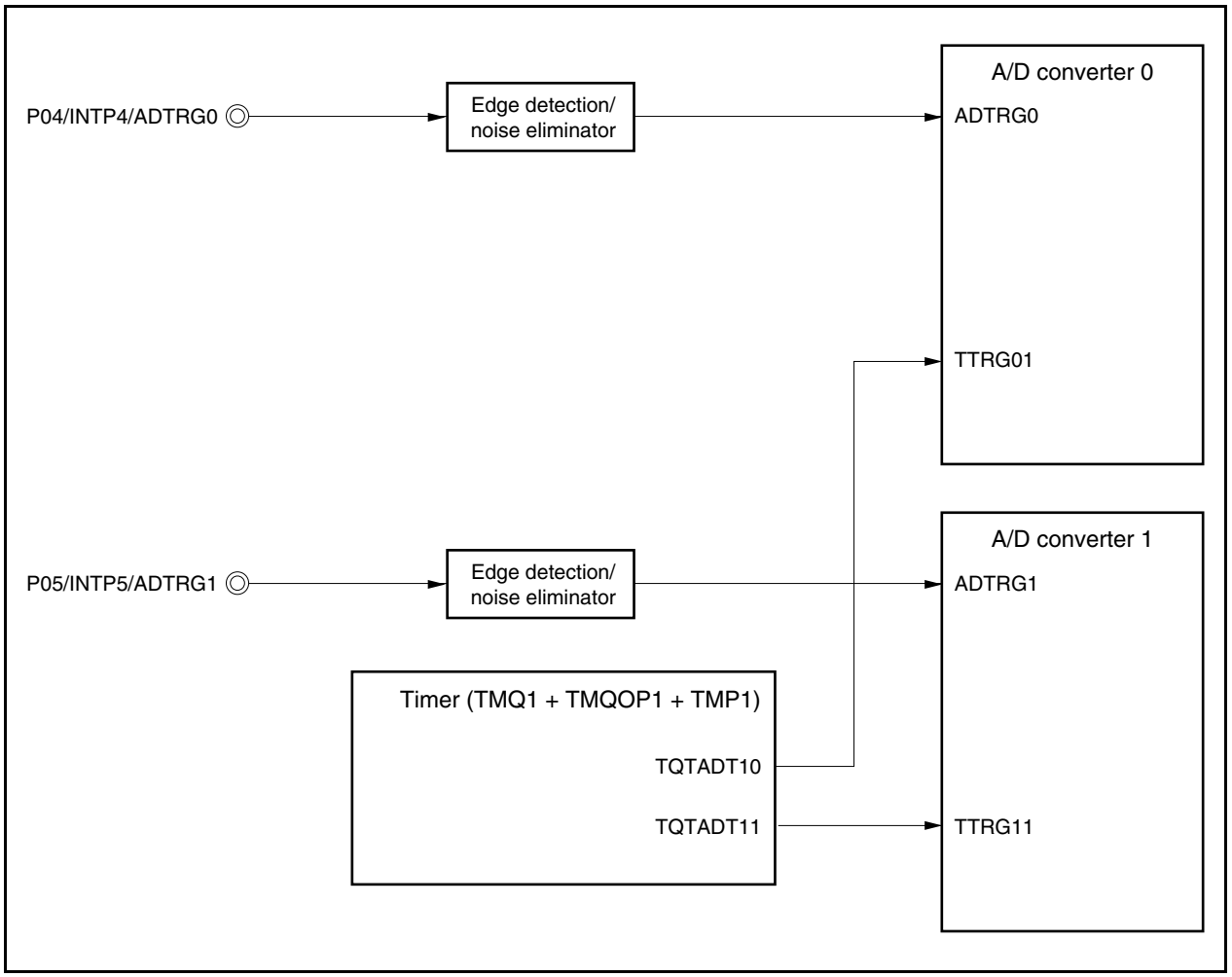
**Cautions 1. If there is noise at the analog input pins (ANIn0 to ANIn3) or at the A/D converter reference voltage input pin ($AV_{REFn}$), that noise may generate an illegal conversion result (n = 0, 1). Software processing will be needed to avoid a negative effect on the system from this illegal conversion result.**

**An example of this software processing is shown below.**

- **Take the average result of a number of A/D conversions and use that as the A/D conversion result.**
- **Execute a number of A/D conversions consecutively and use those results, omitting any exceptional results that may have been obtained.**
- **If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing malfunction processing.**

**2. Do not apply a voltage outside the $AV_{SSn}$ to $AV_{REFn}$ range to the pins that are used as input pins of A/D converters 0 and 1.**

**Figure 11-2. Block Diagram of Trigger Source Selector in Timer Trigger Mode**

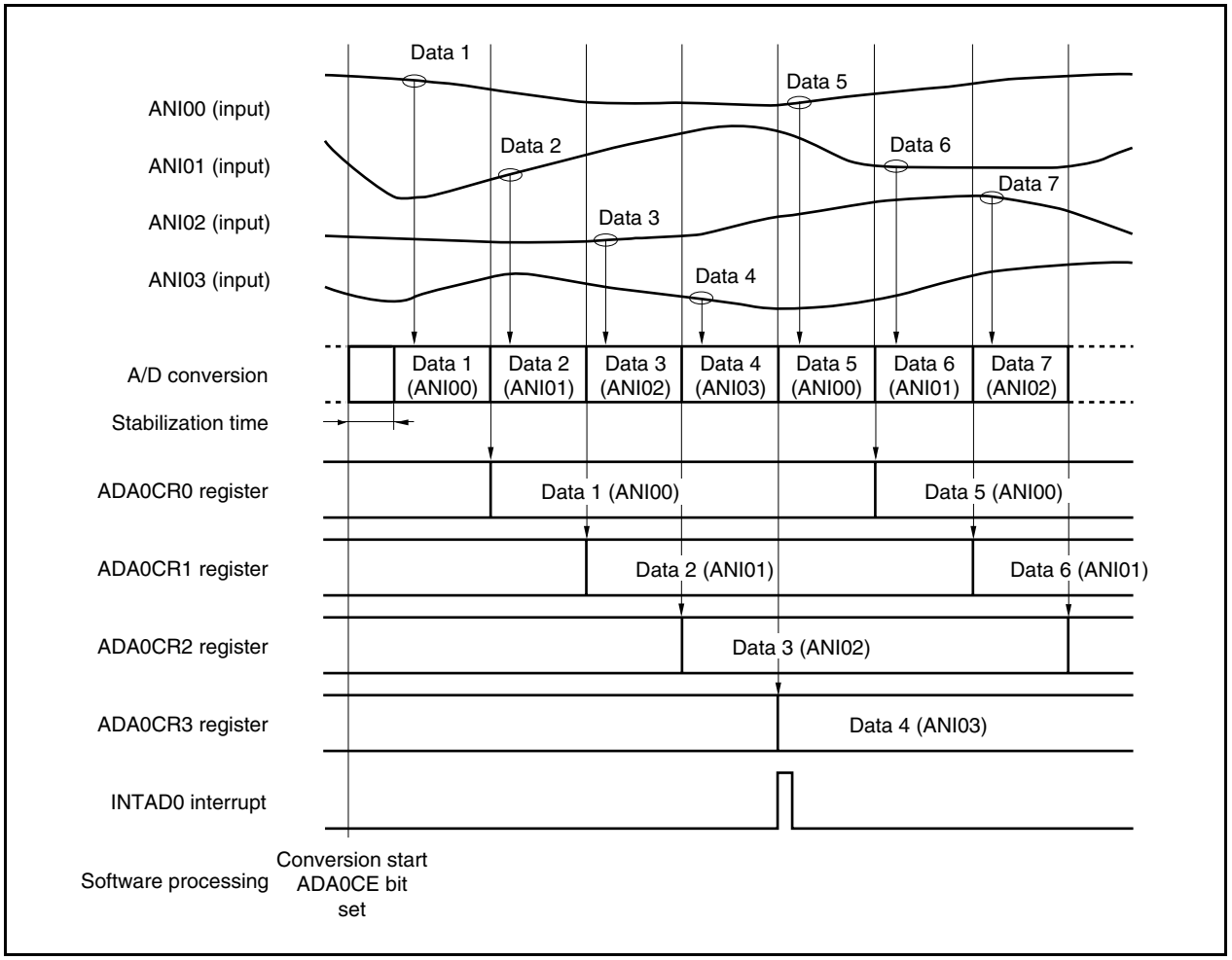


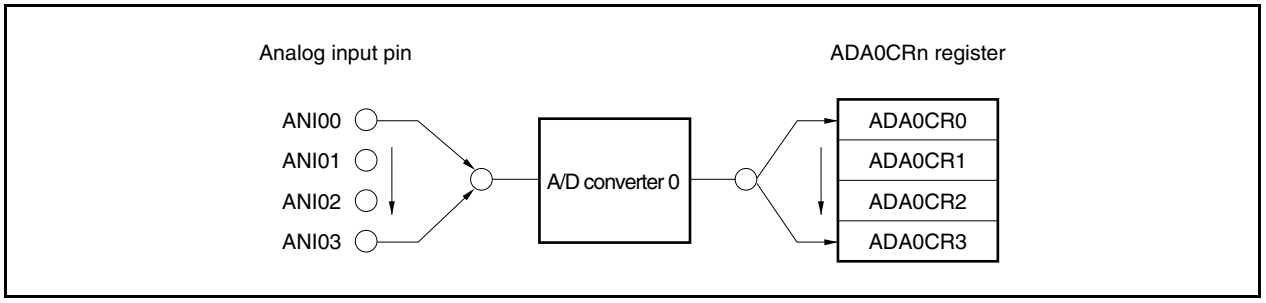
**(b) Continuous scan mode**

In this mode, the analog input pins (ANInm) specified by the ADAnS register are selected sequentially from the ANIn0 pin, and A/D conversion is executed continuously. The A/D conversion results are stored in the ADAnCRm register corresponding to the analog input pin. When conversion of all the specified analog input pin ends, the A/Dn conversion end interrupt request signal (INTADn) is generated. After the end of A/D conversion, the conversion is started again from the ANIn0 pin, unless the ADAnM0.ADAnCE bit is cleared to 0.

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

**Figure 11-6. Continuous Scan Mode Operation Timing**
**(When ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 Bits = 01, ADA0S.ADA0S1 and ADA0S.ADA0S0 Bits = 11)**



### 11.5.4 One-shot scan mode operations

In this mode, the analog input pins (ANInm) specified by the ADAnS register are selected sequentially from the ANIn0 pin, and A/D conversion is executed continuously. The A/D conversion results are stored in the ADAnCRm register corresponding to the analog input pin.

When conversion of all the specified analog input pin ends, the A/Dn conversion end interrupt request signal (INTADn) is generated. After the end of A/D conversion, the conversion operation is stopped.

If the ADAnM0.ADAnCE bit is set (1), A/D conversion can be restarted.

In the one-shot scan mode, only the 1-buffer mode is supported.

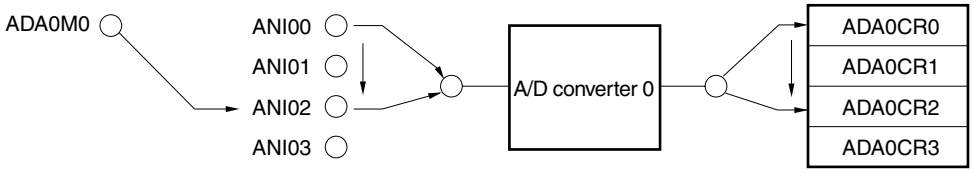
This mode is suitable for applications in which multiple analog inputs are constantly monitored.

| Analog Input Pin | A/D Conversion Result Register |
|---|---|
| ANIn0 | ADAnCR0 |
| ⋮ | ⋮ |
| ANInm[Note] | ADAnCRm |

**Note** Set by the ADAnS.ADAnS0 and ADAnS.ADAnS1 bits.

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

**Figure 11-15. Example of One-Shot Scan Mode Operation (Software Trigger One-Shot Scan)**




(1) The ADA0CE bit = 1 (enable)
(2) The ANI00 pin is A/D converted
(3) The conversion result is stored in the ADA0CR0 register
(4) The ANI01 pin is A/D converted
(5) The conversion result is stored in the ADA0CR1 register
(6) The ANI02 pin is A/D converted
(7) The conversion result is stored in the ADA0CR2 register
(8) The ADA0M0.ADA0EF bit = 0
(9) The INTAD0 interrupt request signal is generated

**Remark** This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 11, ADA0M0.ADA0TMD bit = 0,
ADA0M2.ADA0BS bit = 0, ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 10

## 11.9 Notes on Operation

### 11.9.1 Stopping conversion operation

When the ADAnM0.ADAnCE bit is cleared to 0 during a conversion operation, the conversion operation stops and the conversion results are not stored in A/Dn conversion result register m (ADAnCRm).

The ADAnCE bit is not cleared to 0 even after the A/Dn conversion end interrupt request signal (INTADn) has been generated in all modes.

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

### 11.9.2 Timer/external trigger interval

Make sure that the occurrence interval of the trigger in timer trigger mode or external trigger mode is longer than the total number of conversion clocks specified by the ADAnM1.ADAnFR1 and ADAnM1.ADAnFR0 bits (see **Table 11-2 Number of Conversion Clocks**).

**(1) When 0 < trigger occurrence interval < total number of A/D conversion clocks**

When the timer/external trigger is input during a conversion operation, the conversion operation is aborted and the conversion starts according to the last timer/external trigger input.

When conversion operations are aborted, the conversion results from the conversion operation immediately before are not stored in the ADAnCRm register. Note, therefore, that the generation of the INTADn signal and storing of the result in the ADAnCRm register are not guaranteed.

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

**(2) When trigger occurrence interval ≥ total number of A/D conversion clocks**

The INTADn signal is generated, and the value at the end of conversion is correctly stored in the ADAnCRm register. Design so that the trigger occurrence interval is equal or greater than the total number of A/D conversion clocks.

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

### 11.9.3 Operation in standby mode

**(1) HALT mode**

In this mode, A/D conversion continues.

**(2) IDLE mode, STOP mode**

As clock supply to A/D converters 0 and 1 is stopped, no conversion operations are performed.

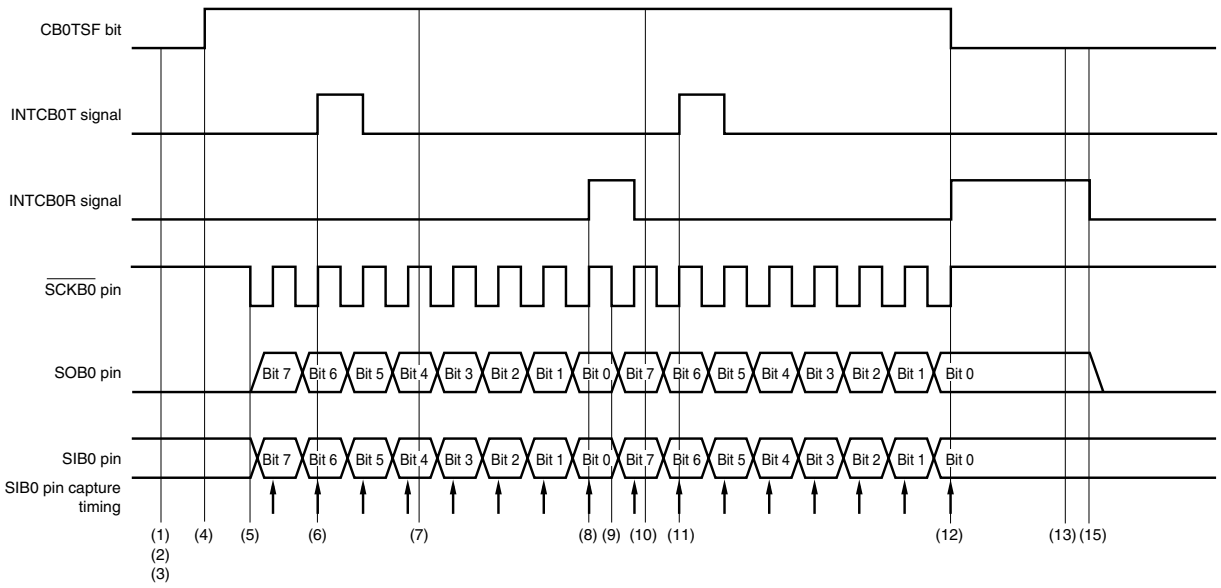
When these modes are released by the maskable interrupt request signal input pin[Note], the ADAnM0, ADAnM1, ADAnM2, and ADAnS registers and A/Dn conversion result register m (ADAnCRm) hold their values. However, when the IDLE or STOP mode is set during a conversion operation, the conversion operation is suspended. At this time, if the mode released by the maskable interrupt request signal input pin[Note], the conversion operation resumes. At this time, the A/Dn conversion end interrupt request signal (INTADn) may be generated, but the conversion result written to the ADAnCRm register will be undefined.

**Note** INTP0 to INTP5

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

### (2) Operation timing

(1/2)




(1) Write 00H to the CB0CTL1 register, and select communication type 1, communication clock ($f_{CCLK}$) = $f_{XX}$/2, and master mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CB0CTL0 register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock ($f_{CCLK}$).

(4) The CB0STR.CB0TSF bit is set to 1 by writing the transmit data to the CB0TX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the $\overline{\text{SCKB0}}$ pin, output the transmit data to the SOB0 pin in synchronization with the serial clock, and capture the receive data of the SIB0 pin.

(6) When transfer of the transmit data from the CB0TX register to the shift register is completed and writing to the CB0TX register is enabled, the transmission enable interrupt request signal (INTCB0T) is generated.

(7) To continue transmission/reception, write the transmit data to the CB0TX register again after the INTCB0T signal is generated.

(8) When one transmission/reception is completed, the reception end interrupt request signal (INTCB0R) is generated, and reading of the CB0RX register is enabled.

(9) When a new transmit data is written to the CB0TX register before communication completion, the next communication is started following communication completion.

(10) Read the CB0RX register.

### 13.4.11 Continuous transfer mode (slave mode, reception mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock ($f_{CCLK}$) = external clock ($\overline{SCKB0}$) (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 111), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

## 15.5 STOP Mode

### 15.5.1 Setting and operation status

The STOP mode is set by setting (1) the PSMR.PSM0 bit and setting (1) the PSC.STB bit in the normal operation mode.

In the STOP mode, the clock generator stops operation. Clock supply to the CPU and the on-chip peripheral functions is stopped.

As a result, program execution is stopped, and the contents of the internal RAM before the STOP mode was set <R> are retained. The CPU and other on-chip peripheral functions stop operating. The on-chip peripheral functions that can operate with an external clock continue operating.

Table 15-7 shows the operation status in the STOP mode.

Because the STOP mode stops operation of the clock generator, it reduces the power consumption to a level lower than the IDLE mode. The power consumption is therefore minimized with only leakage current flowing if the external clock is not used.

**Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode.**

### 15.5.2 Releasing STOP mode

The STOP mode is released by an unmasked external interrupt request signal (INTP0 to INTP5 pin input), unmasked internal interrupt request signal (INTLVI), unmasked internal interrupt request signal from the peripheral functions operable in the STOP mode (interrupt request signal related to CSIB in the slave mode), or reset signal ($\overline{\text{RESET}}$ pin input, reset signal generation by low-voltage detection (LVIRES), and reset signal generation by power-on clear (POCRES)).

After the STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

**(1) Releasing STOP mode by unmasked maskable interrupt request signal**

The STOP mode is released by an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the STOP mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

**Caution When the PSC.INTM bit is set to 1, the STOP mode cannot be released by an unmasked maskable interrupt request signal.**

(a) If an interrupt request signal with a priority lower than or same as the interrupt request signal currently being serviced is generated, the STOP mode is released, but the newly generated interrupt request signal is not acknowledged. The interrupt request signal itself is retained. Therefore, execution starts at the next instruction after the STOP instruction.

(b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued, the STOP mode is released and that interrupt request signal is acknowledged. Therefore, the execution branches to the handler address.

# CHAPTER 16 RESET FUNCTIONS

## 16.1 Overview

The following reset functions are available.

- Reset by $\overline{\text{RESET}}$ pin input
- Reset by watchdog timer overflow (WDTRES)
- System reset by low-voltage detector (LVI) (LVIRES)
- System reset by power-on-clear circuit (POC) (POCRES)

## 16.2 Registers to Check Reset Source

**(1) Reset source flag register (RESF)**

The RESF register is a special register that can be written only by a combination of specific sequences (see **3.4.7 Special registers**).

The RESF register indicates that a reset signal is generated by the watchdog timer (WDT).

The LVIRF and WDT2RF bits are cleared by reset via the $\overline{\text{RESET}}$ pin or by a bit manipulation instruction or store instruction (writing 0 to the LVIRF and WDT2RF bits).

This register is read or written in 8-bit units. However, bit 0 is write-only.

This register is cleared to 00H by $\overline{\text{RESET}}$ pin input and reset by the power-on-clear circuit (POC). The default value differs if the source of reset is other than these.

After reset: 00H[Note] R/W Address: FFFFF888H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| RESF | 0 | 0 | 0 | WDT2RF | 0 | 0 | 0 | LVIRF |

| WDT2RF | Reset signal from WDT |
|---|---|
| 0 | Not generated/cleared |
| 1 | Generated |

| LVIRF | Clear of RESF2.LVIRFS bit |
|---|---|
| 0 | Cleared |
| 1 | Write disabled |

**Note** The value of this register is cleared to 00H after a reset by $\overline{\text{RESET}}$ pin input or the power-on-clear circuit (POC). When a reset is executed by watchdog timer overflow, this register is set to 10H or 11H. When reset by the low-voltage detector (LVI), bit 4 retains the value before reset and bit 0 is undefined.

**Cautions 1. Only "0" can be written to each bit of this register. If writing "0" conflicts with setting the flag (occurrence of reset), setting the flag takes precedence.**

<R> **2. When writing to the RESF register, use command register PRCMD.**

**Notes 12.** In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

**13.** iiiii: Lower 5 bits of imm9.

IIII: Higher 4 bits of imm9.

**14.** Do not specify the same register for general-purpose registers reg1 and reg3.

**15.** sp/imm: specified by bits 19 and 20 of the sub-opcode.

**16.** ff = 00: Load sp in ep.

01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.

10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.

11: Load 32-bit immediate data (bits 63 to 32) in ep.

**17.** If imm = imm32, n + 3 clocks.

**18.** rrrrr: Other than 00000.

**19.** ddddddd: Higher 7 bits of disp8.

**20.** dddd: Higher 4 bits of disp5.

**21.** dddddd: Higher 6 bits of disp8.

**22.** Do not make a register combination that satisfies all the following conditions when executing the “MUL reg1, reg2, reg3” and “MULU reg1, reg2, reg3” instructions. If an instruction that satisfies these conditions is executed, the operation is not guaranteed.

- reg1 = reg3
- reg1 ≠ reg2
- reg1 ≠ r0
- reg3 ≠ r0