

TOSHIBA

8 Bit Microcontroller
TLCS-870/C Series

TMP86PS44UG

Not Recommended
for New Design

TOSHIBA CORPORATION

The information contained herein is subject to change without notice. 021023_D

TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress.

It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.

In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications.

Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc. 021023_A

The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.).

These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer's own risk. 021023_B

The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations. 060106_Q

The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patents or other rights of TOSHIBA or the third parties. 070122_C

The products described in this document are subject to foreign exchange and foreign trade control laws. 060925_E

For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions. 030619_S

Revision History

| Date | Revision | |
|-----------|----------|------------------|
| 2007/8/28 | 1 | First Release |
| 2008/8/29 | 2 | Contents Revised |

Not Recommended
for New Design

Caution in Setting the UART Noise Rejection Time

When UART is used, settings of RXDNC are limited depending on the transfer clock specified by BRG. The combination "O" is available but please do not select the combination "--".

The transfer clock generated by timer/counter interrupt is calculated by the following equation :

$$\text{Transfer clock [Hz]} = \text{Timer/counter source clock [Hz]} \div \text{TTREG set value}$$

| BRG setting | Transfer clock [Hz] | RXDNC setting | | | |
|--|---------------------|----------------------------|--|--|---|
| | | 00 (No noise rejection) | 01 (Reject pulses shorter than $31/fc[s]$ as noise) | 10 (Reject pulses shorter than $63/fc[s]$ as noise) | 11 (Reject pulses shorter than $127/fc[s]$ as noise) |
| 000 | $fc/13$ | O | O | O | - |
| 110 (When the transfer clock generated by timer/counter interrupt is the same as the right side column) | $fc/8$ | O | - | - | - |
| | $fc/16$ | O | O | - | - |
| | $fc/32$ | O | O | O | - |
| The setting except the above | | O | O | O | O |

Not Recommended for New Design

Table of Contents

TMP86PS44UG

| | | |
|-----|-------------------------|---|
| 1.1 | Features | 1 |
| 1.2 | Pin Assignment | 3 |
| 1.3 | Block Diagram | 4 |
| 1.4 | Pin Names and Functions | 5 |

2. Operational Description

| | | |
|---------|--|----|
| 2.1 | CPU Core Functions | 7 |
| 2.1.1 | Memory Address Map | 7 |
| 2.1.2 | Program Memory (OTP) | 7 |
| 2.1.3 | Data Memory (RAM) | 7 |
| 2.2 | System Clock Controller | 8 |
| 2.2.1 | Clock Generator | 8 |
| 2.2.2 | Timing Generator | 10 |
| 2.2.2.1 | Configuration of timing generator | |
| 2.2.2.2 | Machine cycle | |
| 2.2.3 | Operation Mode Control Circuit | 11 |
| 2.2.3.1 | Single-clock mode | |
| 2.2.3.2 | Dual-clock mode | |
| 2.2.3.3 | STOP mode | |
| 2.2.4 | Operating Mode Control | 16 |
| 2.2.4.1 | STOP mode | |
| 2.2.4.2 | IDLE1/2 mode and SLEEP1/2 mode | |
| 2.2.4.3 | IDLE0 and SLEEP0 modes (IDLE0, SLEEP0) | |
| 2.2.4.4 | SLOW mode | |
| 2.3 | Reset Circuit | 29 |
| 2.3.1 | External Reset Input | 29 |
| 2.3.2 | Address trap reset | 30 |
| 2.3.3 | Watchdog timer reset | 30 |
| 2.3.4 | System clock reset | 30 |

3. Interrupt Control Circuit

| | | |
|---------|--|----|
| 3.1 | Interrupt latches (IL15 to IL2) | 33 |
| 3.2 | Interrupt enable register (EIR) | 34 |
| 3.2.1 | Interrupt master enable flag (IMF) | 34 |
| 3.2.2 | Individual interrupt enable flags (EF15 to EF4) | 34 |
| 3.3 | Interrupt Source Selector (INTSEL) | 37 |
| 3.4 | Interrupt Sequence | 37 |
| 3.4.1 | Interrupt acceptance processing is packaged as follows | 37 |
| 3.4.2 | Saving/restoring general-purpose registers | 38 |
| 3.4.2.1 | Using PUSH and POP instructions | |
| 3.4.2.2 | Using data transfer instructions | |
| 3.4.3 | Interrupt return | 40 |
| 3.5 | Software Interrupt (INTSW) | 41 |
| 3.5.1 | Address error detection | 41 |
| 3.5.2 | Debugging | 41 |

| | | |
|-----|--|----|
| 3.6 | Undefined Instruction Interrupt (INTUNDEF) | 41 |
| 3.7 | Address Trap Interrupt (INTATRAP) | 41 |
| 3.8 | External Interrupts | 41 |

4. Special Function Register (SFR)

| | | |
|-----|-----------|----|
| 4.1 | SFR | 45 |
|-----|-----------|----|

5. I/O Ports

| | | |
|-----|----------------------------|----|
| 5.1 | Port P0 (P07 to P00) | 48 |
| 5.2 | Port P1 (P17 to P10) | 49 |
| 5.3 | Port P2 (P22 to P20) | 50 |
| 5.4 | Port P3 (P37 to P30) | 51 |
| 5.5 | Port P4 (P47 to P40) | 52 |

6. Time Base Timer (TBT)

| | | |
|-------|----------------------------|----|
| 6.1 | Time Base Timer | 53 |
| 6.1.1 | Configuration | 53 |
| 6.1.2 | Control | 53 |
| 6.1.3 | Function | 54 |
| 6.2 | Divider Output (DVO) | 55 |
| 6.2.1 | Configuration | 55 |
| 6.2.2 | Control | 55 |

7. Watchdog Timer (WDT)

| | | |
|-------|--|----|
| 7.1 | Watchdog Timer Configuration | 57 |
| 7.2 | Watchdog Timer Control | 58 |
| 7.2.1 | Malfunction Detection Methods Using the Watchdog Timer | 58 |
| 7.2.2 | Watchdog Timer Enable | 59 |
| 7.2.3 | Watchdog Timer Disable | 60 |
| 7.2.4 | Watchdog Timer Interrupt (INTWDT) | 60 |
| 7.2.5 | Watchdog Timer Reset | 61 |
| 7.3 | Address Trap | 62 |
| 7.3.1 | Selection of Address Trap in Internal RAM (ATAS) | 62 |
| 7.3.2 | Selection of Operation at Address Trap (ATOUT) | 62 |
| 7.3.3 | Address Trap Interrupt (INTATRAP) | 62 |
| 7.3.4 | Address Trap Reset | 63 |

8. 16-Bit TimerCounter 1 (TC1)

| | | |
|-------|-----------------------------------|----|
| 8.1 | Configuration | 65 |
| 8.2 | TimerCounter Control | 66 |
| 8.3 | Function | 68 |
| 8.3.1 | Timer mode | 68 |
| 8.3.2 | External Trigger Timer Mode | 70 |
| 8.3.3 | Event Counter Mode | 72 |
| 8.3.4 | Window Mode | 73 |

| | | |
|-------|---|----|
| 8.3.5 | Pulse Width Measurement Mode | 74 |
| 8.3.6 | Programmable Pulse Generate (PPG) Output Mode | 77 |

9. 16-Bit Timer/Counter2 (TC2)

| | | |
|-------|-------------------------|----|
| 9.1 | Configuration | 81 |
| 9.2 | Control | 82 |
| 9.3 | Function | 83 |
| 9.3.1 | Timer mode..... | 83 |
| 9.3.2 | Event counter mode..... | 85 |
| 9.3.3 | Window mode | 85 |

10. 8-Bit TimerCounter (TC3, TC4)

| | | |
|----------|--|-----|
| 10.1 | Configuration | 87 |
| 10.2 | TimerCounter Control | 88 |
| 10.3 | Function | 93 |
| 10.3.1 | 8-Bit Timer Mode (TC3 and 4) | 93 |
| 10.3.2 | 8-Bit Event Counter Mode (TC3, 4) | 94 |
| 10.3.3 | 8-Bit Programmable Divider Output (PDO) Mode (TC3, 4)..... | 94 |
| 10.3.4 | 8-Bit Pulse Width Modulation (PWM) Output Mode (TC3, 4)..... | 97 |
| 10.3.5 | 16-Bit Timer Mode (TC3 and 4) | 99 |
| 10.3.6 | 16-Bit Event Counter Mode (TC3 and 4) | 100 |
| 10.3.7 | 16-Bit Pulse Width Modulation (PWM) Output Mode (TC3 and 4)..... | 100 |
| 10.3.8 | 16-Bit Programmable Pulse Generate (PPG) Output Mode (TC3 and 4)..... | 103 |
| 10.3.9 | Warm-Up Counter Mode..... | 105 |
| 10.3.9.1 | Low-Frequency Warm-up Counter Mode (NORMAL1 → NORMAL2 → SLOW2 → SLOW1) | |
| 10.3.9.2 | High-Frequency Warm-Up Counter Mode (SLOW1 → SLOW2 → NORMAL2 → NORMAL1) | |

11. Synchronous Serial Interface (SIO)

| | | |
|----------|------------------------------|-----|
| 11.1 | Configuration | 107 |
| 11.2 | Control | 108 |
| 11.3 | Function | 110 |
| 11.3.1 | Serial clock | 110 |
| 11.3.1.1 | Clock source | |
| 11.3.1.2 | Shift edge | |
| 11.3.2 | Transfer bit direction | 112 |
| 11.3.2.1 | Transmit mode | |
| 11.3.2.2 | Receive mode | |
| 11.3.2.3 | Transmit/receive mode | |
| 11.3.3 | Transfer modes..... | 113 |
| 11.3.3.1 | Transmit mode | |
| 11.3.3.2 | Receive mode | |
| 11.3.3.3 | Transmit/receive mode | |

12. Asynchronous Serial interface (UART)

| | | |
|------|----------------------------|-----|
| 12.1 | Configuration | 125 |
| 12.2 | Control | 126 |
| 12.3 | Transfer Data Format | 128 |
| 12.4 | Transfer Rate | 129 |
| 12.5 | Data Sampling Method | 129 |

| | | |
|--------|----------------------------------|-----|
| 12.6 | STOP Bit Length | 130 |
| 12.7 | Parity | 130 |
| 12.8 | Transmit/Receive Operation | 130 |
| 12.8.1 | Data Transmit Operation | 130 |
| 12.8.2 | Data Receive Operation | 130 |
| 12.9 | Status Flag | 131 |
| 12.9.1 | Parity Error | 131 |
| 12.9.2 | Framing Error | 131 |
| 12.9.3 | Overrun Error | 131 |
| 12.9.4 | Receive Data Buffer Full | 132 |
| 12.9.5 | Transmit Data Buffer Empty | 132 |
| 12.9.6 | Transmit End Flag | 133 |

13. 10-bit AD Converter (ADC)

| | | |
|--------|---|-----|
| 13.1 | Configuration | 135 |
| 13.2 | Register configuration | 136 |
| 13.3 | Function | 139 |
| 13.3.1 | Software Start Mode | 139 |
| 13.3.2 | Repeat Mode | 139 |
| 13.3.3 | Register Setting | 140 |
| 13.4 | STOP/SLOW Modes during AD Conversion | 141 |
| 13.5 | Analog Input Voltage and AD Conversion Result | 142 |
| 13.6 | Precautions about AD Converter | 143 |
| 13.6.1 | Analog input pin voltage range | 143 |
| 13.6.2 | Analog input shared pins | 143 |
| 13.6.3 | Noise Countermeasure | 143 |

14. 8-Bit DA Converter (DAC)

| | | |
|--------|------------------------------|-----|
| 14.1 | Register Configuration | 145 |
| 14.1.1 | DA Converter Operation | 146 |

15. Key-on Wakeup (KWU)

| | | |
|------|---------------------|-----|
| 15.1 | Configuration | 147 |
| 15.2 | Control | 147 |
| 15.3 | Function | 147 |

16. OTP operation

| | | |
|----------|---|-----|
| 16.1 | Operating mode | 149 |
| 16.1.1 | MCU mode | 149 |
| 16.1.1.1 | Program Memory | |
| 16.1.1.2 | Data Memory | |
| 16.1.1.3 | Input/Output Circuitry | |
| 16.1.2 | PROM mode | 151 |
| 16.1.2.1 | Programming Flowchart (High-speed program writing) | |
| 16.1.2.2 | Program Writing using a General-purpose PROM Programmer | |

17. Input/Output Circuitry

| | | |
|------|--------------------------|-----|
| 17.1 | Control Pins | 155 |
| 17.2 | Input/Output Ports | 156 |

18. Electrical Characteristics

| | | |
|--------|---|-----|
| 18.1 | Absolute Maximum Ratings | 157 |
| 18.2 | Operating Condition | 158 |
| 18.3 | DC Characteristics | 159 |
| 18.4 | AD Conversion Characteristics | 160 |
| 18.5 | DA Conversion Characteristics | 161 |
| 18.6 | AC Characteristics | 162 |
| 18.7 | DC Characteristics, AC Characteristics (PROM mode) | 163 |
| 18.7.1 | Read operation in PROM mode | 163 |
| 18.7.2 | Program operation (High-speed) ($T_{opr} = 25 \pm 5^{\circ}\text{C}$) | 164 |
| 18.8 | Recommended Oscillating Conditions | 165 |
| 18.9 | Handling Precaution | 165 |

19. Package Dimensions

This is a technical document that describes the operating functions and electrical specifications of the 8-bit microcontroller series TLCS-870/C (LSI).

Not Recommended
for New Design



Not Recommended
for New Design

CMOS 8-Bit Microcontroller
TMP86PS44UG

The TMP86PS44UG is a single-chip 8-bit high-speed and high-functionality microcomputer incorporating 61440 bytes of One-Time PROM. It is pin-compatible with the TMP86CS44UG (Mask ROM version). The TMP86PS44UG can realize operations equivalent to those of the TMP86CS44UG by programming the on-chip PROM.

| Product No. | ROM (EPROM) | RAM | Package | MaskROM MCU | Emulation Chip |
|-------------|-------------|------------|---------------------|-------------|----------------|
| TMP86PS44UG | 61440 bytes | 1024 bytes | LQFP44-P-1010-0.80A | TMP86CS44UG | TMP86C944XB |

1.1 Features

1. 8-bit single chip microcomputer TLCS-870/C series
 - Instruction execution time :
 - 0.25 μ s (at 16 MHz)
 - 122 μ s (at 32.768 kHz)
 - 132 types & 731 basic instructions
2. 19interrupt sources (External : 6 Internal : 13)
3. Input / Output ports (35 pins)
 - Large current output: 19pins (Typ. 20mA), LED direct drive
4. Prescaler
 - Time base timer
 - Divider output function
5. Watchdog Timer
6. 16-bit timer counter: 1 ch
 - Timer, External trigger, Window, Pulse width measurement, Event counter, Programmable pulse generate (PPG) modes
7. 16-bit timer counter: 1 ch
 - Timer, Event counter, Window modes

• The information contained herein is subject to change without notice. 021023_D

• TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc. 021023_A

• The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer's own risk. 021023_B

• The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations. 060106_Q

• The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patents or other rights of TOSHIBA or the third parties. 070122_C

• The products described in this document are subject to foreign exchange and foreign trade control laws. 060925_E

• For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions. 030619_S

8. 8-bit timer counter : 2 ch
 - Timer, Event counter, Programmable divider output (PDO),
Pulse width modulation (PWM) output,
Programmable pulse generation (PPG) modes
9. High-Speed SIO: 1ch
10. 8-bit UART : 1 ch
11. 10-bit successive approximation type AD converter
 - Analog input: 8 ch
12. 8-Bit DA Converter
 - Analog output : 1 channel
13. Key-on wakeup : 4 ch
14. Clock operation
 - Single clock mode
 - Dual clock mode
15. Low power consumption operation
 - STOP mode: Oscillation stops. (Battery/Capacitor back-up.)
 - SLOW1 mode: Low power consumption operation using low-frequency clock.(High-frequency clock stop.)
 - SLOW2 mode: Low power consumption operation using low-frequency clock.(High-frequency clock oscillate.)
 - IDLE0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using high frequency clock. Release by falling edge of the source clock which is set by TBTCCR<TBTCK>.
 - IDLE1 mode: CPU stops and peripherals operate using high frequency clock. Release by interrupts(CPU restarts).
 - IDLE2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interrupts. (CPU restarts).
 - SLEEP0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using low frequency clock.Release by falling edge of the source clock which is set by TBTCCR<TBTCK>.
 - SLEEP1 mode: CPU stops, and peripherals operate using low frequency clock. Release by interrupt.(CPU restarts).
 - SLEEP2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interrupt.
16. Wide operation voltage:
 - 4.5 V to 5.5 V at 16MHz /32.768 kHz
 - 2.7 V to 5.5 V at 8 MHz/32.768 kHz

1.2 Pin Assignment

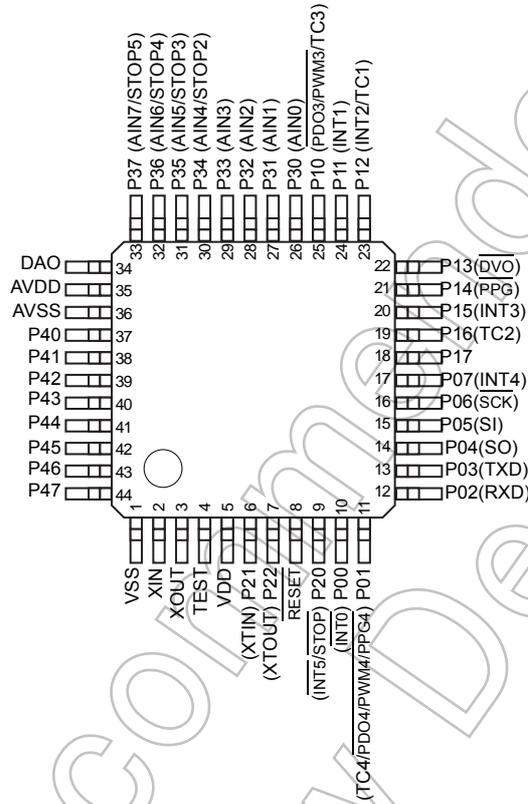


Figure 1-1 Pin Assignment

Not Recommended for New Design

1.3 Block Diagram

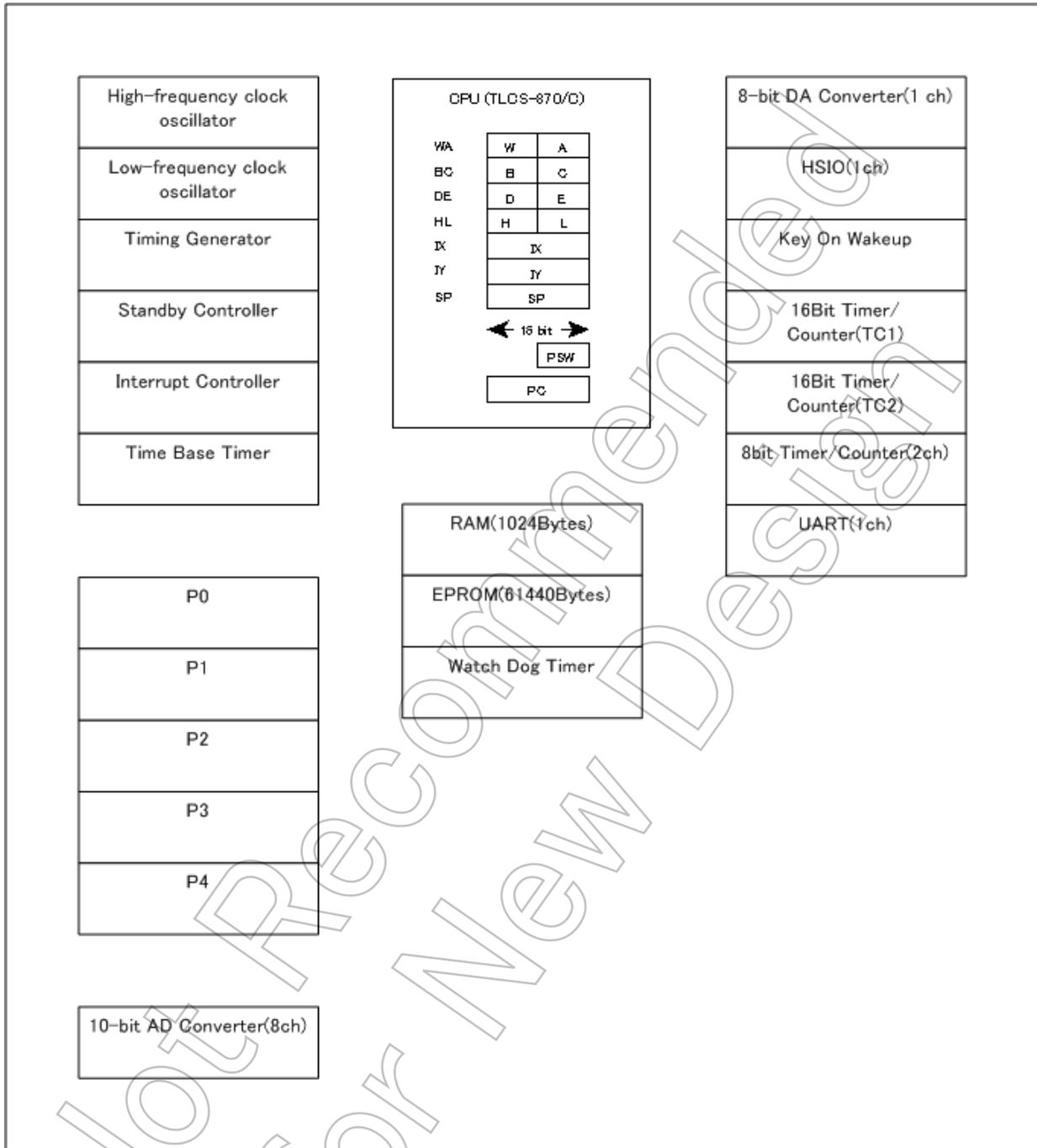


Figure 1-2 Block Diagram

1.4 Pin Names and Functions

The TMP86PS44UG has MCU mode and PROM mode. Table 1-1 shows the pin functions in MCU mode. The PROM mode is explained later in a separate chapter.

Table 1-1 Pin Names and Functions(1/2)

| Pin Name | Pin Number | Input/Output | Functions |
|------------------------------|------------|--------------|---|
| P07 INT4 | 17 | IO I | PORT07 External interrupt 4 input |
| P06 SCK | 16 | IO IO | PORT06 Serial clock input/output |
| P05 SI | 15 | IO I | PORT05 Serial data input |
| P04 SO | 14 | IO O | PORT04 Serial data output |
| P03 TXD | 13 | IO O | PORT03 UART data output |
| P02 RXD | 12 | IO I | PORT02 UART data input |
| P01 PDO4/PWM4/PPG4 TC4 | 11 | IO O I | PORT01 PDO4/PWM4/PPG4 output TC4 input |
| P00 INT0 | 10 | IO I | PORT00 External interrupt 0 input |
| P17 | 18 | IO | PORT17 |
| P16 TC2 | 19 | IO I | PORT16 TC2 input |
| P15 INT3 | 20 | IO I | PORT15 External interrupt 3 input |
| P14 PPG | 21 | IO O | PORT14 PPG output |
| P13 DVO | 22 | IO O | PORT13 Divider Output |
| P12 INT2 TC1 | 23 | IO I I | PORT12 External interrupt 2 input TC1 input |
| P11 INT1 | 24 | IO I | PORT11 External interrupt 1 input |
| P10 PDO3/PWM3 TC3 | 25 | IO O I | PORT10 PDO3/PWM3 output TC3 input |
| P22 XTOUT | 7 | IO O | PORT22 Resonator connecting pins(32.768kHz) for inputting external clock |
| P21 XTIN | 6 | IO I | PORT21 Resonator connecting pins(32.768kHz) for inputting external clock |
| P20 STOP INT5 | 9 | IO I I | PORT20 STOP mode release signal input External interrupt 5 input |

Table 1-1 Pin Names and Functions(2/2)

| Pin Name | Pin Number | Input/Output | Functions |
|----------------------|------------|--------------|---|
| P37 AIN7 STOP5 | 33 | IO I I | PORT37 Analog Input7 STOP5 input |
| P36 AIN6 STOP4 | 32 | IO I I | PORT36 Analog Input6 STOP4 input |
| P35 AIN5 STOP3 | 31 | IO I I | PORT35 Analog Input5 STOP3 input |
| P34 AIN4 STOP2 | 30 | IO I I | PORT34 Analog Input4 STOP2 input |
| P33 AIN3 | 29 | IO I | PORT33 Analog Input3 |
| P32 AIN2 | 28 | IO I | PORT32 Analog Input2 |
| P31 AIN1 | 27 | IO I | PORT31 Analog Input1 |
| P30 AIN0 | 26 | IO I | PORT30 Analog Input0 |
| P47 | 44 | IO | PORT47 |
| P46 | 43 | IO | PORT46 |
| P45 | 42 | IO | PORT45 |
| P44 | 41 | IO | PORT44 |
| P43 | 40 | IO | PORT43 |
| P42 | 39 | IO | PORT42 |
| P41 | 38 | IO | PORT41 |
| P40 | 37 | IO | PORT40 |
| XIN | 2 | I | Resonator connecting pins for high-frequency clock |
| XOUT | 3 | O | Resonator connecting pins for high-frequency clock |
| RESET | 8 | IO | Reset signal |
| TEST | 4 | I | Test pin for out-going test. Normally, be fixed to low. |
| AVDD | 35 | I | Power supply |
| AVSS | 36 | I | Power supply |
| VDD | 5 | I | Power Supply |
| VSS | 1 | I | 0(GND) |

2. Operational Description

2.1 CPU Core Functions

The CPU core consists of a CPU, a system clock controller, and an interrupt controller.

This section provides a description of the CPU core, the program memory, the data memory, and the reset circuit.

2.1.1 Memory Address Map

The TMP86PS44UG memory is composed OTP, RAM and SFR(Special function register). They are all mapped in 64-Kbyte address space. Figure 2-1 shows the TMP86PS44UG memory address map.

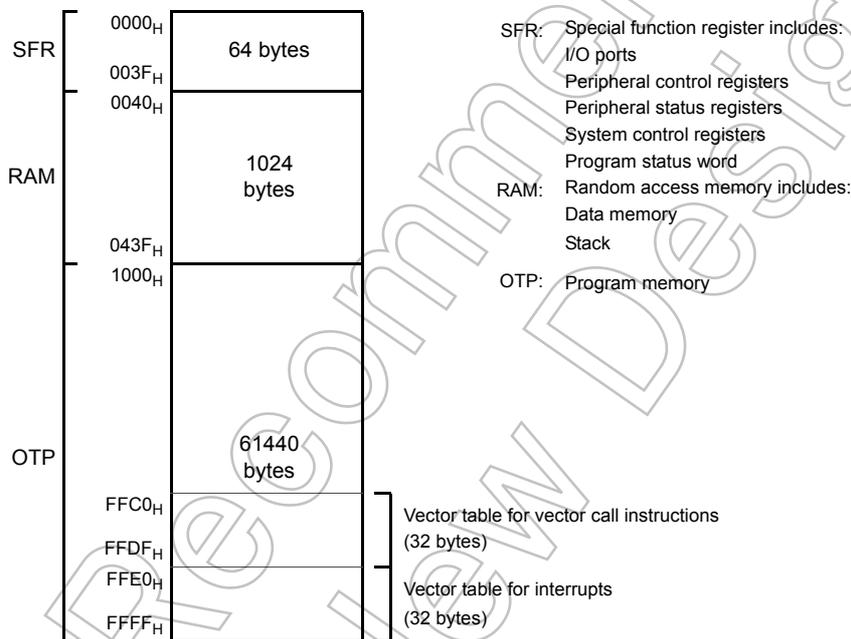


Figure 2-1 Memory Address Map

2.1.2 Program Memory (OTP)

The TMP86PS44UG has a 61440 bytes (Address 1000H to FFFFH) of program memory (OTP).

2.1.3 Data Memory (RAM)

The TMP86PS44UG has 1024bytes (Address 0040H to 043FH) of internal RAM. The first 192 bytes (0040H to 00FFH) of the internal RAM are located in the direct area; instructions with shorten operations are available against such an area.

The data memory contents become unstable when the power supply is turned on; therefore, the data memory should be initialized by an initialization routine.

Example :Clears RAM to “00H”. (TMP86PS44UG)

```

LD      HL, 0040H      ; Start address setup
LD      A, H          ; Initial value (00H) setup
LD      BC, 03FFH
SRAMCLR: LD      (HL), A
INC     HL
DEC     BC
JRS    F, SRAMCLR
    
```

2.2 System Clock Controller

The system clock controller consists of a clock generator, a timing generator, and a standby controller.

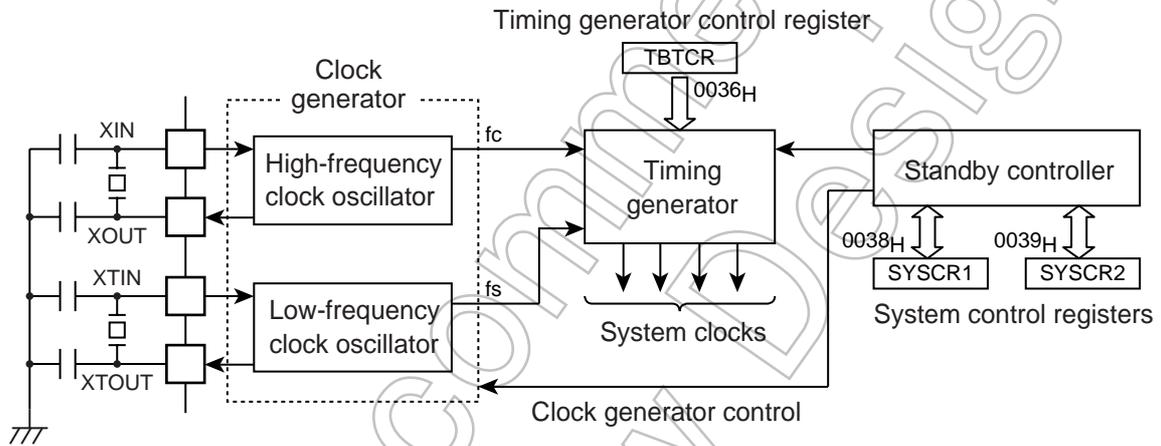


Figure 2-2 System Clock Control

2.2.1 Clock Generator

The clock generator generates the basic clock which provides the system clocks supplied to the CPU core and peripheral hardware. It contains two oscillation circuits: One for the high-frequency clock and one for the low-frequency clock. Power consumption can be reduced by switching of the standby controller to low-power operation based on the low-frequency clock.

The high-frequency (fc) clock and low-frequency (fs) clock can easily be obtained by connecting a resonator between the XIN/XOUT and XTIN/XTOUT pins respectively. Clock input from an external oscillator is also possible. In this case, external clock is applied to XIN/XTIN pin with XOUT/XTOUT pin not connected.

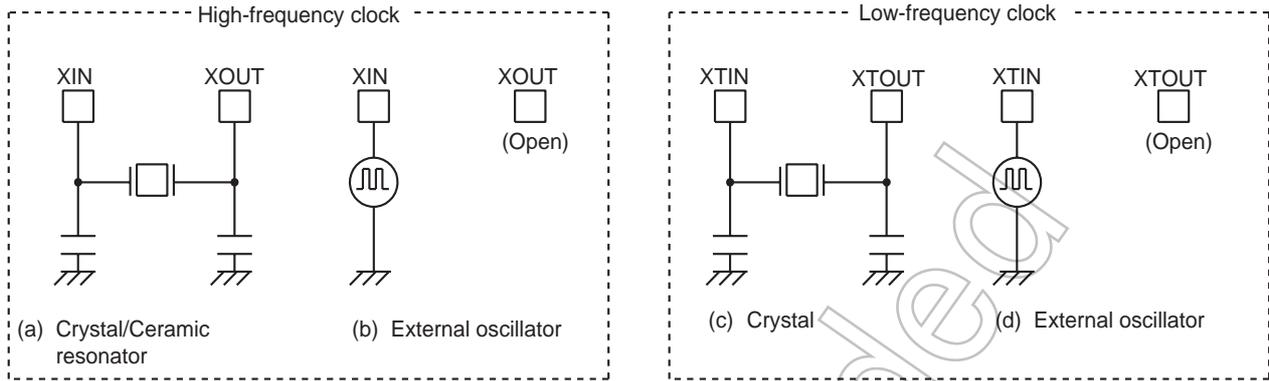


Figure 2-3 Examples of Resonator Connection

Note: The function to monitor the basic clock directly at external is not provided for hardware, however, with disabling all interrupts and watchdog timers, the oscillation frequency can be adjusted by monitoring the pulse which the fixed frequency is outputted to the port by the program.
 The system to require the adjustment of the oscillation frequency should create the program for the adjustment in advance.

Not Recommended for New Design

2.2.2 Timing Generator

The timing generator generates the various system clocks supplied to the CPU core and peripheral hardware from the basic clock (f_c or f_s). The timing generator provides the following functions.

1. Generation of main system clock
2. Generation of divider output ($\overline{DV0}$) pulses
3. Generation of source clocks for time base timer
4. Generation of source clocks for watchdog timer
5. Generation of internal source clocks for timer/counters
6. Generation of warm-up clocks for releasing STOP mode

2.2.2.1 Configuration of timing generator

The timing generator consists of a 2-stage prescaler, a 21-stage divider, a main system clock generator, and machine cycle counters.

An input clock to the 7th stage of the divider depends on the operating mode, $\text{SYSCK} < \text{SYSCK} >$ and $\text{TBTCR} < \text{DV7CK} >$, that is shown in Figure 2-4. As reset and STOP mode started/canceled, the prescaler and the divider are cleared to "0".

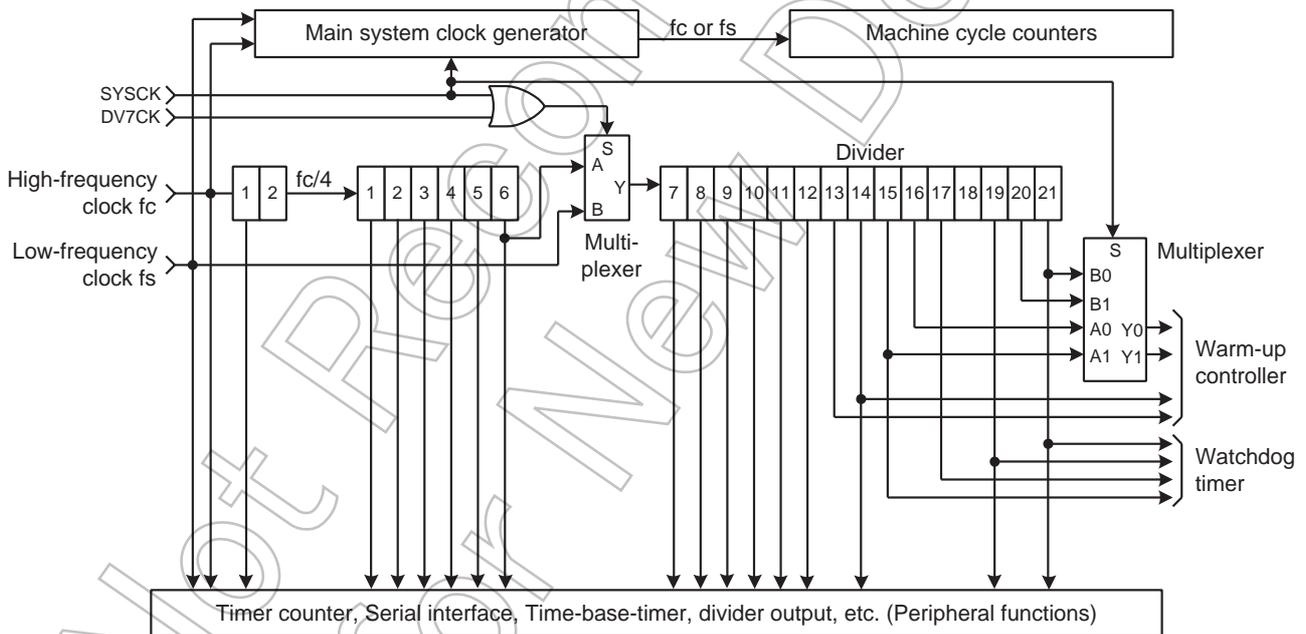


Figure 2-4 Configuration of Timing Generator

Timing Generator Control Register

| | | | | | | | | | |
|------------------|--|---------|---------------------------|---------|---------|---|---|---|----------------------------|
| TBTCR (0036H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: 0000 0000) |
| | (DVOEN) | (DVOCK) | DV7CK | (TBTEN) | (TBTCK) | | | | |
| DV7CK | Selection of input to the 7th stage of the divider | | 0: $fc/2^8$ [Hz] 1: fs | | | | | | R/W |

- Note 1: In single clock mode, do not set DV7CK to "1".
- Note 2: Do not set "1" on DV7CK while the low-frequency clock is not operated stably.
- Note 3: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care
- Note 4: In SLOW1/2 and SLEEP1/2 modes, the DV7CK setting is ineffective, and fs is input to the 7th stage of the divider.
- Note 5: When STOP mode is entered from NORMAL1/2 mode, the DV7CK setting is ineffective during the warm-up period after release of STOP mode, and the 6th stage of the divider is input to the 7th stage during this period.

2.2.2.2 Machine cycle

Instruction execution and peripheral hardware operation are synchronized with the main system clock.

The minimum instruction execution unit is called a "machine cycle". There are a total of 10 different types of instructions for the TLCS-870/C Series: Ranging from 1-cycle instructions which require one machine cycle for execution to 10-cycle instructions which require 10 machine cycles for execution. A machine cycle consists of 4 states (S0 to S3), and each state consists of one main system clock.

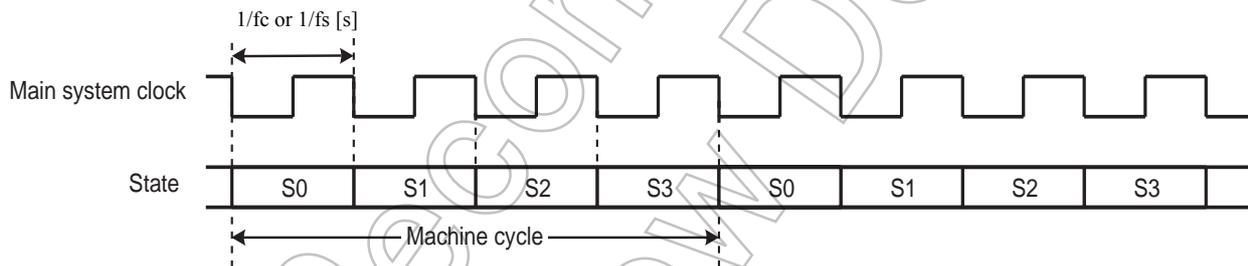


Figure 2-5 Machine Cycle

2.2.3 Operation Mode Control Circuit

The operation mode control circuit starts and stops the oscillation circuits for the high-frequency and low-frequency clocks, and switches the main system clock. There are three operating modes: Single clock mode, dual clock mode and STOP mode. These modes are controlled by the system control registers (SYSCR1 and SYSCR2). Figure 2-6 shows the operating mode transition diagram.

2.2.3.1 Single-clock mode

Only the oscillation circuit for the high-frequency clock is used, and P21 (XTIN) and P22 (XTOUT) pins are used as input/output ports. The main-system clock is obtained from the high-frequency clock. In the single-clock mode, the machine cycle time is $4/fc$ [s].

(1) NORMAL1 mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock. The TMP86PS44UG is placed in this mode after reset.

(2) IDLE1 mode

In this mode, the internal oscillation circuit remains active. The CPU and the watchdog timer are halted; however on-chip peripherals remain active (Operate using the high-frequency clock).

IDLE1 mode is started by `SYSCR2<IDLE> = "1"`, and IDLE1 mode is released to NORMAL1 mode by an interrupt request from the on-chip peripherals or external interrupt inputs. When the IMF (Interrupt master enable flag) is "1" (Interrupt enable), the execution will resume with the acceptance of the interrupt, and the operation will return to normal after the interrupt service is completed. When the IMF is "0" (Interrupt disable), the execution will resume with the instruction which follows the IDLE1 mode start instruction.

(3) IDLE0 mode

In this mode, all the circuit, except oscillator and the timer-base-timer, stops operation.

This mode is enabled by `SYSCR2<TGHALT> = "1"`.

When IDLE0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with `TBTCR<TBTCK>`, the timing generator starts feeding the clock to all peripheral circuits.

When returned from IDLE0 mode, the CPU restarts operating, entering NORMAL1 mode back again. IDLE0 mode is entered and returned regardless of how `TBTCR<TBTEN>` is set. When `IMF = "1"`, `EF6` (TBT interrupt individual enable flag) = "1", and `TBTCR<TBTEN> = "1"`, interrupt processing is performed. When IDLE0 mode is entered while `TBTCR<TBTEN> = "1"`, the `INTTBT` interrupt latch is set after returning to NORMAL1 mode.

2.2.3.2 Dual-clock mode

Both the high-frequency and low-frequency oscillation circuits are used in this mode. P21 (XTIN) and P22 (XTOUT) pins cannot be used as input/output ports. The main system clock is obtained from the high-frequency clock in NORMAL2 and IDLE2 modes, and is obtained from the low-frequency clock in SLOW and SLEEP modes. The machine cycle time is $4/f_c$ [s] in the NORMAL2 and IDLE2 modes, and $4/f_s$ [s] (122 μ s at $f_s = 32.768$ kHz) in the SLOW and SLEEP modes.

The TLCS-870/C is placed in the signal-clock mode during reset. To use the dual-clock mode, the low-frequency oscillator should be turned on at the start of a program.

(1) NORMAL2 mode

In this mode, the CPU core operates with the high-frequency clock. On-chip peripherals operate using the high-frequency clock and/or low-frequency clock.

(2) SLOW2 mode

In this mode, the CPU core operates with the low-frequency clock, while both the high-frequency clock and the low-frequency clock are operated. As the `SYSCR2<SYSCK>` becomes "1", the hardware changes into SLOW2 mode. As the `SYSCR2<SYSCK>` becomes "0", the hardware changes into NORMAL2 mode. As the `SYSCR2<XEN>` becomes "0", the hardware changes into SLOW1 mode. Do not clear `SYSCR2<XTEN>` to "0" during SLOW2 mode.

(3) SLOW1 mode

This mode can be used to reduce power-consumption by turning off oscillation of the high-frequency clock. The CPU core and on-chip peripherals operate using the low-frequency clock.

Switching back and forth between SLOW1 and SLOW2 modes are performed by SYSCR2<XEN>. In SLOW1 and SLEEP modes, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

(4) IDLE2 mode

In this mode, the internal oscillation circuit remain active. The CPU and the watchdog timer are halted; however, on-chip peripherals remain active (Operate using the high-frequency clock and/or the low-frequency clock). Starting and releasing of IDLE2 mode are the same as for IDLE1 mode, except that operation returns to NORMAL2 mode.

(5) SLEEP1 mode

In this mode, the internal oscillation circuit of the low-frequency clock remains active. The CPU, the watchdog timer, and the internal oscillation circuit of the high-frequency clock are halted; however, on-chip peripherals remain active (Operate using the low-frequency clock). Starting and releasing of SLEEP mode are the same as for IDLE1 mode, except that operation returns to SLOW1 mode. In SLOW1 and SLEEP1 modes, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

(6) SLEEP2 mode

The SLEEP2 mode is the idle mode corresponding to the SLOW2 mode. The status under the SLEEP2 mode is same as that under the SLEEP1 mode, except for the oscillation circuit of the high-frequency clock.

(7) SLEEP0 mode

In this mode, all the circuit, except oscillator and the timer-base-timer, stops operation. This mode is enabled by setting "1" on bit SYSCR2<TGHALT>.

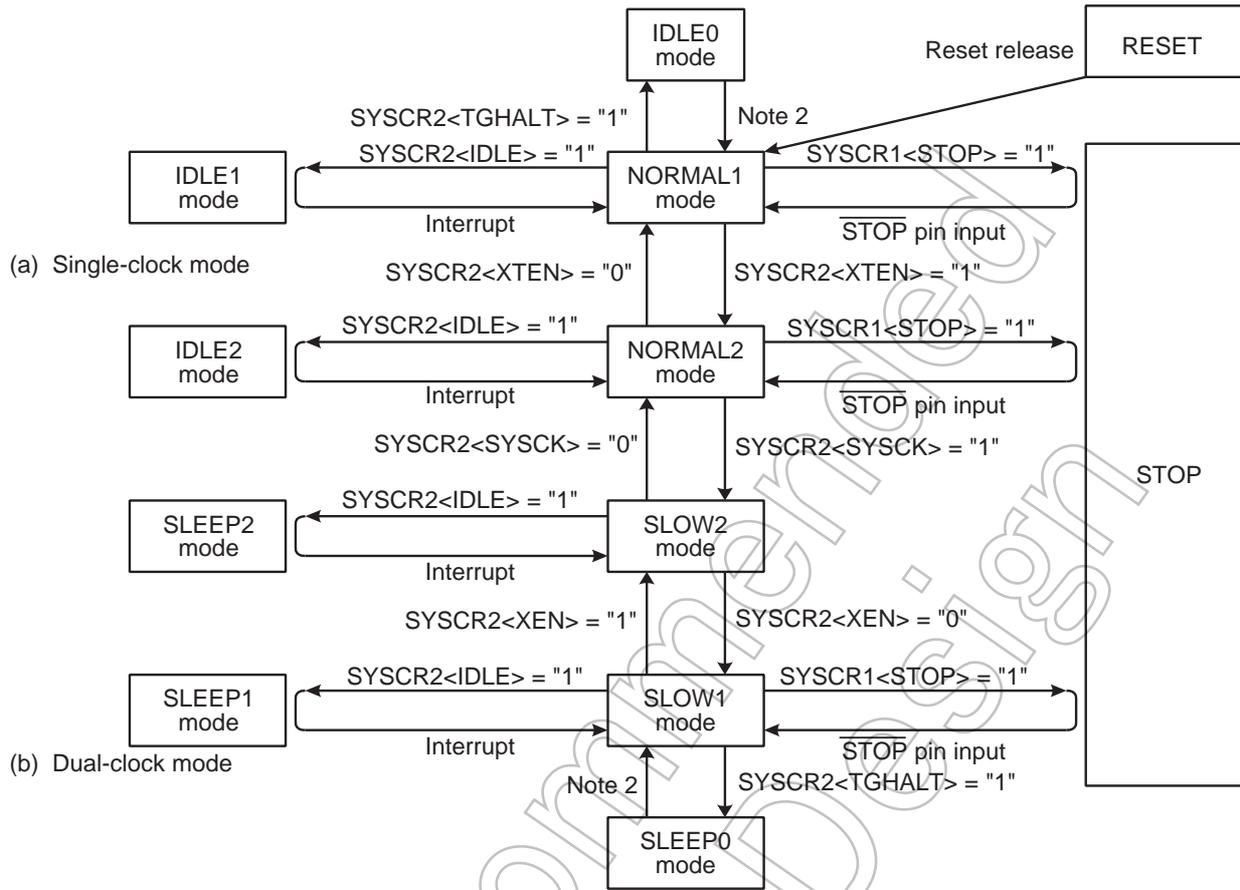
When SLEEP0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR<TBTK>, the timing generator starts feeding the clock to all peripheral circuits.

When returned from SLEEP0 mode, the CPU restarts operating, entering SLOW1 mode back again. SLEEP0 mode is entered and returned regardless of how TBTCR<TBTEN> is set. When IMF = "1", EF6 (TBT interrupt individual enable flag) = "1", and TBTCR<TBTEN> = "1", interrupt processing is performed. When SLEEP0 mode is entered while TBTCR<TBTEN> = "1", the INTTBT interrupt latch is set after returning to SLOW1 mode.

2.2.3.3 STOP mode

In this mode, the internal oscillation circuit is turned off, causing all system operations to be halted. The internal status immediately prior to the halt is held with a lowest power consumption during STOP mode.

STOP mode is started by the system control register 1 (SYSCR1), and STOP mode is released by a inputting (Either level-sensitive or edge-sensitive can be programmably selected) to the STOP pin. After the warm-up period is completed, the execution resumes with the instruction which follows the STOP mode start instruction.



Note 1: NORMAL1 and NORMAL2 modes are generically called NORMAL; SLOW1 and SLOW2 are called SLOW; IDLE0, IDLE1 and IDLE2 are called IDLE; SLEEP0, SLEEP1 and SLEEP2 are called SLEEP.

Note 2: The mode is released by falling edge of TBTCR<TBTCk> setting.

Figure 2-6 Operating Mode Transition Diagram

Table 2-1 Operating Mode and Conditions

| Operating Mode | | Oscillator | | CPU Core | TBT | Other Peripherals | Machine Cycle Time | | |
|----------------|---------|----------------|---------------|-----------------------------|---------|-------------------|--------------------|---------|----------|
| | | High Frequency | Low Frequency | | | | | | |
| Single clock | RESET | Oscillation | Stop | Reset | Reset | Reset | 4/fc [s] | | |
| | NORMAL1 | | | Operate | Operate | Operate | | | |
| | IDLE1 | | | Halt | | | | Operate | Halt |
| | IDLE0 | | | | Halt | Halt | | | |
| | STOP | | | Stop | | | Halt | Halt | - |
| Dual clock | NORMAL2 | Oscillation | Oscillation | Operate with high frequency | Operate | Operate | 4/fc [s] | | |
| | IDLE2 | | | Halt | | | | | |
| | SLOW2 | | | Operate with low frequency | | | | | |
| | SLEEP2 | | | Halt | | | | | |
| | SLOW1 | Stop | Oscillation | Operate with low frequency | | | Operate | Operate | 4/fs [s] |
| | SLEEP1 | | | Halt | | | | | |
| | SLEEP0 | | | | | | | | |
| | STOP | | | Stop | | | | | |

System Control Register 1

| | | | | | | | | | |
|---------|------|------|------|-------|-----|---|---|---|----------------------------|
| SYSCR1 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| (0038H) | STOP | RELM | RETM | OUTEN | WUT | | | | (Initial value: 0000 00**) |

| | | | | | |
|-------|-------------------------------------|---|-----------------------|----------------------|-----|
| STOP | STOP mode start | 0: CPU core and peripherals remain active 1: CPU core and peripherals are halted (Start STOP mode) | | R/W | |
| RELM | Release method for STOP mode | 0: Edge-sensitive release 1: Level-sensitive release | | R/W | |
| RETM | Operating mode after STOP mode | 0: Return to NORMAL1/2 mode 1: Return to SLOW1 mode | | R/W | |
| OUTEN | Port output during STOP mode | 0: High impedance 1: Output kept | | R/W | |
| WUT | Warm-up time at releasing STOP mode | | Return to NORMAL mode | Return to SLOW mode | R/W |
| | | 00 | $3 \times 2^{16}/fc$ | $3 \times 2^{13}/fs$ | |
| | | 01 | $2^{16}/fc$ | $2^{13}/fs$ | |
| | | 10 | $3 \times 2^{14}/fc$ | $3 \times 2^6/fs$ | |
| | | 11 | $2^{14}/fc$ | $2^6/fs$ | |

- Note 1: Always set RETM to "0" when transitioning from NORMAL mode to STOP mode. Always set RETM to "1" when transitioning from SLOW mode to STOP mode.
- Note 2: When STOP mode is released with \overline{RESET} pin input, a return is made to NORMAL1 regardless of the RETM contents.
- Note 3: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care
- Note 4: Bits 1 and 0 in SYSCR1 are read as undefined data when a read instruction is executed.
- Note 5: As the hardware becomes STOP mode under OUTEN = "0", input value is fixed to "0"; therefore it may cause external interrupt request on account of falling edge.
- Note 6: When the key-on wakeup is used, RELM should be set to "1".
- Note 7: Port P20 is used as \overline{STOP} pin. Therefore, when stop mode is started, OUTEN does not affect to P20, and P20 becomes High-Z mode.
- Note 8: The warmig-up time should be set correctly for using oscillator.

System Control Register 2

| | | | | | | | | | |
|---------|-----|------|-------|------|---|--------|---|---|----------------------------|
| SYSCR2 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| (0039H) | XEN | XTEN | SYSCK | IDLE | | TGHALT | | | (Initial value: 1000 *0**) |

| | | | | |
|--------|---|--|--|-----|
| XEN | High-frequency oscillator control | 0: Turn off oscillation 1: Turn on oscillation | | R/W |
| XTEN | Low-frequency oscillator control | 0: Turn off oscillation 1: Turn on oscillation | | |
| SYSCK | Main system clock select (Write)/main system clock monitor (Read) | 0: High-frequency clock (NORMAL1/NORMAL2/IDLE1/IDLE2) 1: Low-frequency clock (SLOW1/SLOW2/SLEEP1/SLEEP2) | | R/W |
| IDLE | CPU and watchdog timer control (IDLE1/2 and SLEEP1/2 modes) | 0: CPU and watchdog timer remain active 1: CPU and watchdog timer are stopped (Start IDLE1/2 and SLEEP1/2 modes) | | |
| TGHALT | TG control (IDLE0 and SLEEP0 modes) | 0: Feeding clock to all peripherals from TG 1: Stop feeding clock to peripherals except TBT from TG. (Start IDLE0 and SLEEP0 modes) | | |

- Note 1: A reset is applied if both XEN and XTEN are cleared to "0", XEN is cleared to "0" when SYSCK = "0", or XTEN is cleared to "0" when SYSCK = "1".
- Note 2: *: Don't care, TG: Timing generator, *: Don't care
- Note 3: Bits 3, 1 and 0 in SYSCR2 are always read as undefined value.
- Note 4: Do not set IDLE and TGHALT to "1" simultaneously.
- Note 5: Because returning from IDLE0/SLEEP0 to NORMAL1/SLOW1 is executed by the asynchronous internal clock, the period of IDLE0/SLEEP0 mode might be shorter than the period setting by $TBTCR < TBTCR >$.
- Note 6: When IDLE1/2 or SLEEP1/2 mode is released, IDLE is automatically cleared to "0".
- Note 7: When IDLE0 or SLEEP0 mode is released, TGHALT is automatically cleared to "0".
- Note 8: Before setting TGHALT to "1", be sure to stop peripherals. If peripherals are not stopped, the interrupt latch of peripherals may be set after IDLE0 or SLEEP0 mode is released.

2.2.4 Operating Mode Control

2.2.4.1 STOP mode

STOP mode is controlled by the system control register 1, the $\overline{\text{STOP}}$ pin input and key-on wakeup input (STOP5 to STOP2) which is controlled by the STOP mode release control register (STOPCR). The $\overline{\text{STOP}}$ pin is also used both as a port P20 and an $\overline{\text{INT5}}$ (external interrupt input 5) pin. STOP mode is started by setting SYSCR1<STOP> to “1”. During STOP mode, the following status is maintained.

1. Oscillations are turned off, and all internal operations are halted.
2. The data memory, registers, the program status word and port output latches are all held in the status in effect before STOP mode was entered.
3. The prescaler and the divider of the timing generator are cleared to “0”.
4. The program counter holds the address 2 ahead of the instruction (e.g., [SET (SYSCR1).7]) which started STOP mode.

STOP mode includes a level-sensitive mode and an edge-sensitive mode, either of which can be selected with the SYSCR1<RELM>. Do not use any key-on wakeup input (STOP5 to STOP2) for releasing STOP mode in edge-sensitive mode.

Note 1: The STOP mode can be released by either the STOP or key-on wakeup pin (STOP5 to STOP2). However, because the STOP pin is different from the key-on wakeup and can not inhibit the release input, the STOP pin must be used for releasing STOP mode.

Note 2: During STOP period (from start of STOP mode to end of warm up), due to changes in the external interrupt pin signal, interrupt latches may be set to “1” and interrupts may be accepted immediately after STOP mode is released. Before starting STOP mode, therefore, disable interrupts. Also, before enabling interrupts after STOP mode is released, clear unnecessary interrupt latches.

(1) Level-sensitive release mode (RELM = “1”)

In this mode, STOP mode is released by setting the $\overline{\text{STOP}}$ pin high or setting the STOP5 to STOP2 pin input which is enabled by STOPCR. This mode is used for capacitor backup when the main power supply is cut off and long term battery backup.

Even if an instruction for starting STOP mode is executed while $\overline{\text{STOP}}$ pin input is high or STOP5 to STOP2 input is low, STOP mode does not start but instead the warm-up sequence starts immediately. Thus, to start STOP mode in the level-sensitive release mode, it is necessary for the program to first confirm that the $\overline{\text{STOP}}$ pin input is low or STOP5 to STOP2 input is high. The following two methods can be used for confirmation.

1. Testing a port.
2. Using an external interrupt input $\overline{\text{INT5}}$ ($\overline{\text{INT5}}$ is a falling edge-sensitive input).

Example 1 :Starting STOP mode from NORMAL mode by testing a port P20.

```

LD          (SYSCR1), 01010000B    ; Sets up the level-sensitive release mode
SSTOPH:    TEST      (P2PRD), 0      ; Wait until the  $\overline{\text{STOP}}$  pin input goes low level
           JRS       F, SSTOPH
           DI                          ; IMF ← 0
           SET      (SYSCR1), 7      ; Starts STOP mode
    
```

Example 2 :Starting STOP mode from NORMAL mode with an INT5 interrupt.

```

PINT5:      TEST      (P2PRD). 0           ; To reject noise, STOP mode does not start if
           JRS        F, SINT5           port P20 is at high
           LD         (SYSCR1), 01010000B ; Sets up the level-sensitive release mode.
           DI         ; IMF ← 0
           SET       (SYSCR1). 7         ; Starts STOP mode

SINT5:      RETI
    
```

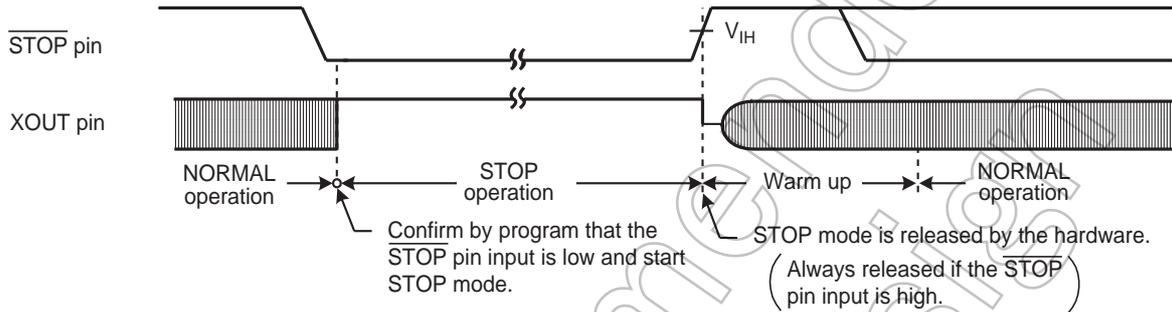


Figure 2-7 Level-sensitive Release Mode

- Note 1: Even if the $\overline{\text{STOP}}$ pin input is low after warm-up start, the STOP mode is not restarted.
- Note 2: In this case of changing to the level-sensitive mode from the edge-sensitive mode, the release mode is not switched until a rising edge of the $\overline{\text{STOP}}$ pin input is detected.

(2) Edge-sensitive release mode (RELM = "0")

In this mode, STOP mode is released by a rising edge of the $\overline{\text{STOP}}$ pin input. This is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (for example, a clock from a low-power consumption oscillator) is input to the $\overline{\text{STOP}}$ pin. In the edge-sensitive release mode, STOP mode is started even when the $\overline{\text{STOP}}$ pin input is high level. Do not use any STOP5 to STOP2 pin input for releasing STOP mode in edge-sensitive release mode.

Example :Starting STOP mode from NORMAL mode

```

DI         ; IMF ← 0
LD         (SYSCR1), 10010000B ; Starts after specified to the edge-sensitive release mode
    
```

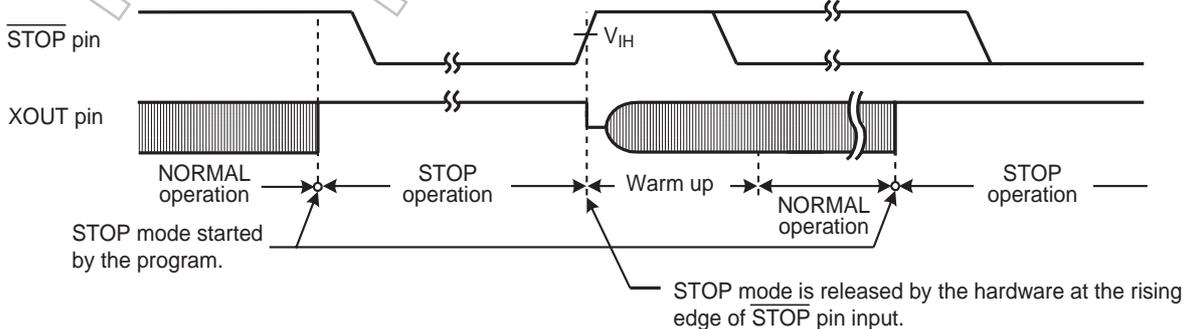


Figure 2-8 Edge-sensitive Release Mode

STOP mode is released by the following sequence.

1. In the dual-clock mode, when returning to NORMAL2, both the high-frequency and low-frequency clock oscillators are turned on; when returning to SLOW1 mode, only the low-frequency clock oscillator is turned on. In the single-clock mode, only the high-frequency clock oscillator is turned on.
2. A warm-up period is inserted to allow oscillation time to stabilize. During warm up, all internal operations remain halted. Four different warm-up times can be selected with the SYSCR1<WUT> in accordance with the resonator characteristics.
3. When the warm-up time has elapsed, normal operation resumes with the instruction following the STOP mode start instruction.

Note 1: When the STOP mode is released, the start is made after the prescaler and the divider of the timing generator are cleared to "0".

Note 2: STOP mode can also be released by inputting low level on the RESET pin, which immediately performs the normal reset operation.

Note 3: When STOP mode is released with a low hold voltage, the following cautions must be observed. The power supply voltage must be at the operating voltage level before releasing STOP mode. The RESET pin input must also be "H" level, rising together with the power supply voltage. In this case, if an external time constant circuit has been connected, the RESET pin input voltage will increase at a slower pace than the power supply voltage. At this time, there is a danger that a reset may occur if input voltage level of the RESET pin drops below the non-inverting high-level input voltage (Hysteresis input).

Table 2-2 Warm-up Time Example (at $f_c = 16.0$ MHz, $f_s = 32.768$ kHz)

| WUT | Warm-up Time [ms] | |
|-----|-----------------------|---------------------|
| | Return to NORMAL Mode | Return to SLOW Mode |
| 00 | 12.288 | 750 |
| 01 | 4.096 | 250 |
| 10 | 3.072 | 5.85 |
| 11 | 1.024 | 1.95 |

Note 1: The warm-up time is obtained by dividing the basic clock by the divider. Therefore, the warm-up time may include a certain amount of error if there is any fluctuation of the oscillation frequency when STOP mode is released. Thus, the warm-up time must be considered as an approximate value.

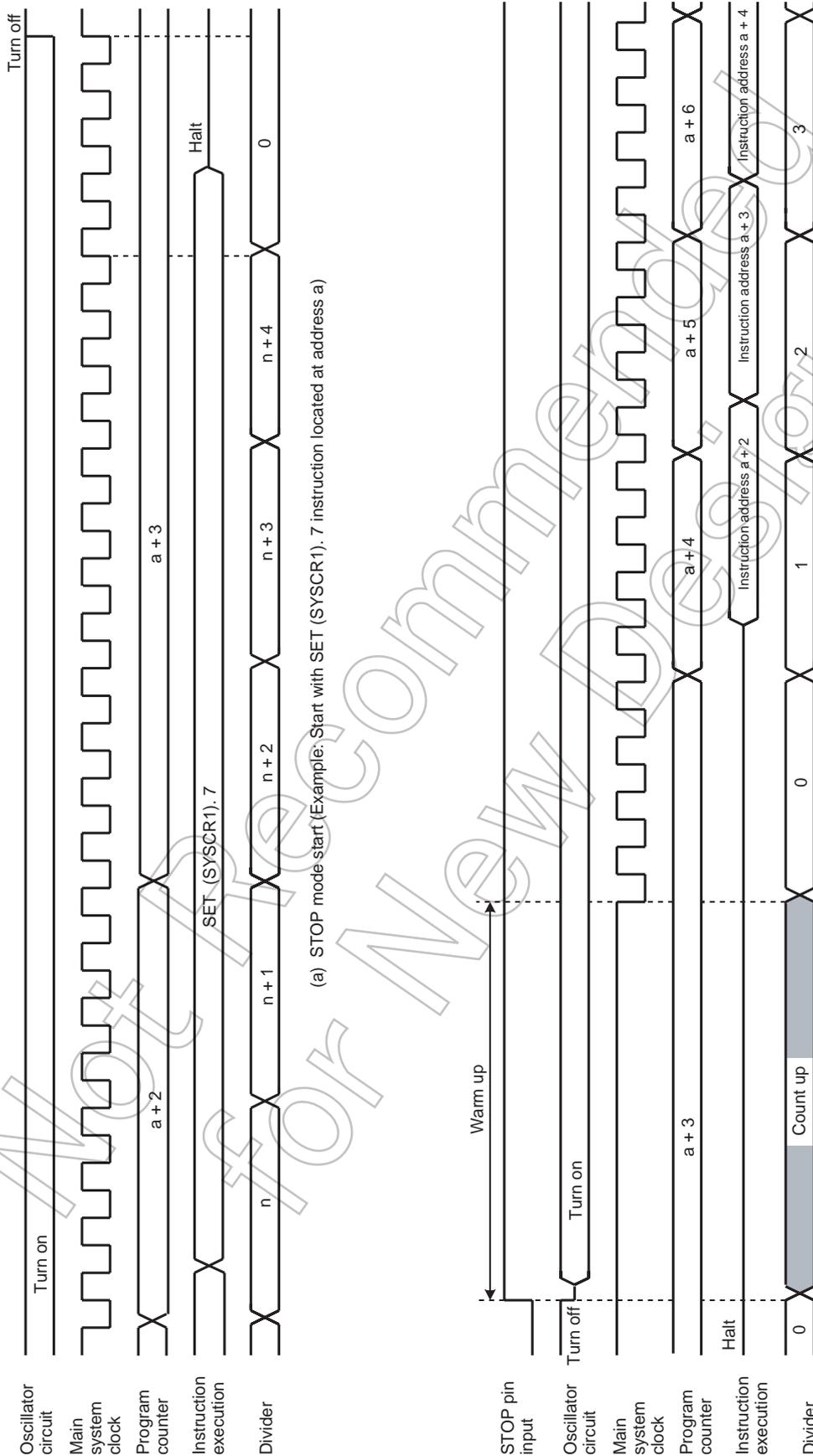


Figure 2-9 STOP Mode Start/Release

2.2.4.2 IDLE1/2 mode and SLEEP1/2 mode

IDLE1/2 and SLEEP1/2 modes are controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following status is maintained during these modes.

1. Operation of the CPU and watchdog timer (WDT) is halted. On-chip peripherals continue to operate.
2. The data memory, CPU registers, program status word and port output latches are all held in the status in effect before these modes were entered.
3. The program counter holds the address 2 ahead of the instruction which starts these modes.

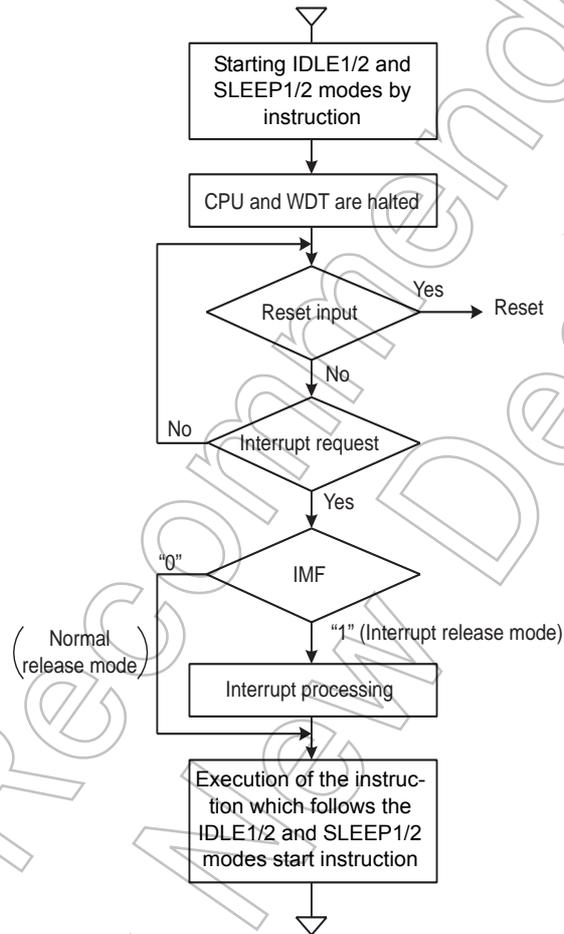


Figure 2-10 IDLE1/2 and SLEEP1/2 Modes

- Start the IDLE1/2 and SLEEP1/2 modes

After IMF is set to "0", set the individual interrupt enable flag (EF) which releases IDLE1/2 and SLEEP1/2 modes. To start IDLE1/2 and SLEEP1/2 modes, set SYSCR2<IDLE> to "1".

- Release the IDLE1/2 and SLEEP1/2 modes

IDLE1/2 and SLEEP1/2 modes include a normal release mode and an interrupt release mode. These modes are selected by interrupt master enable flag (IMF). After releasing IDLE1/2 and SLEEP1/2 modes, the SYSCR2<IDLE> is automatically cleared to "0" and the operation mode is returned to the mode preceding IDLE1/2 and SLEEP1/2 modes.

IDLE1/2 and SLEEP1/2 modes can also be released by inputting low level on the $\overline{\text{RESET}}$ pin. After releasing reset, the operation mode is started from NORMAL1 mode.

(1) Normal release mode (IMF = "0")

IDLE1/2 and SLEEP1/2 modes are released by any interrupt source enabled by the individual interrupt enable flag (EF). After the interrupt is generated, the program operation is resumed from the instruction following the IDLE1/2 and SLEEP1/2 modes start instruction. Normally, the interrupt latches (IL) of the interrupt source used for releasing must be cleared to "0" by load instructions.

(2) Interrupt release mode (IMF = "1")

IDLE1/2 and SLEEP1/2 modes are released by any interrupt source enabled with the individual interrupt enable flag (EF) and the interrupt processing is started. After the interrupt is processed, the program operation is resumed from the instruction following the instruction, which starts IDLE1/2 and SLEEP1/2 modes.

Note: When a watchdog timer interrupts is generated immediately before IDLE1/2 and SLEEP1/2 modes are started, the watchdog timer interrupt will be processed but IDLE1/2 and SLEEP1/2 modes will not be started.

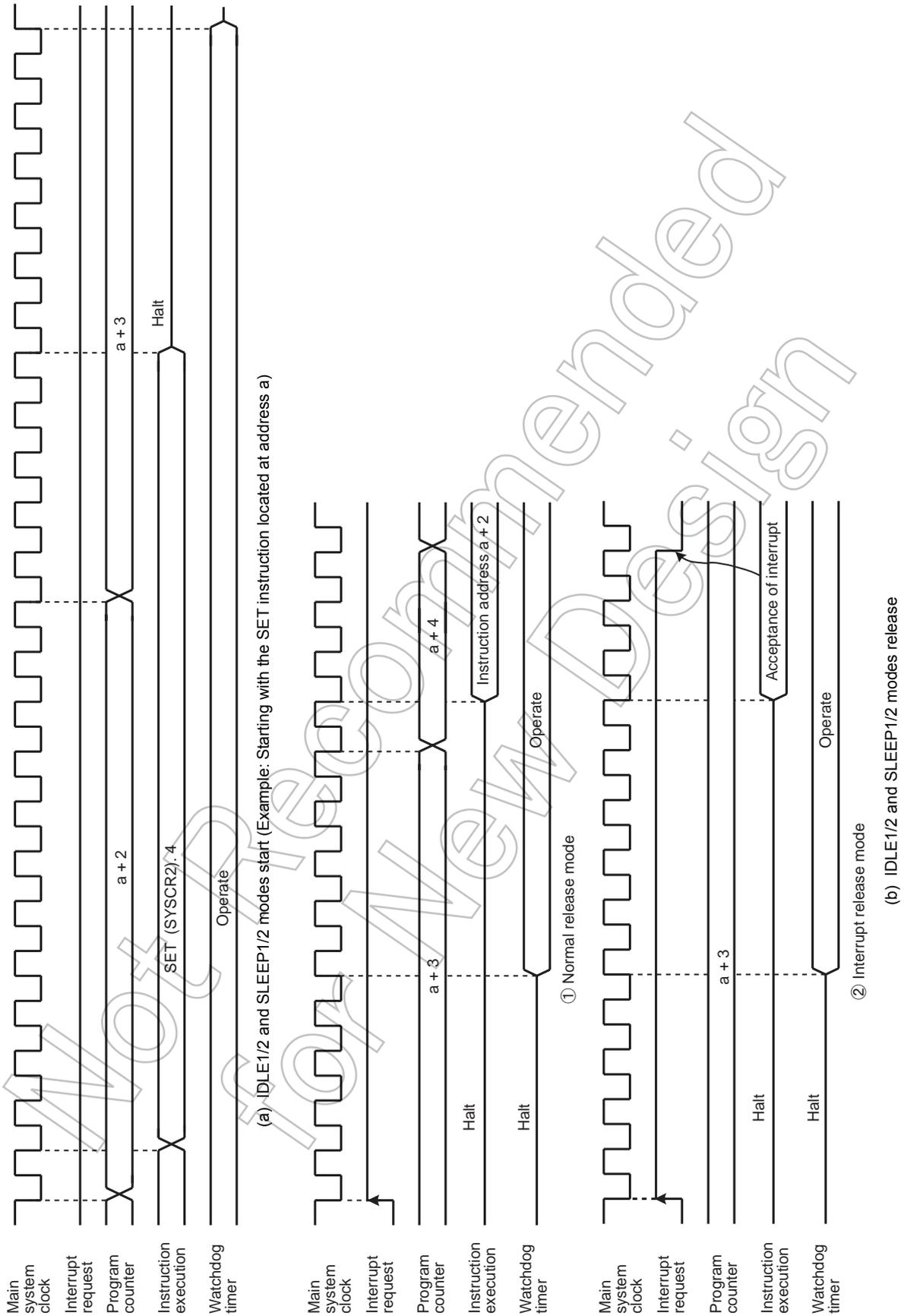


Figure 2-11 IDLE1/2 and SLEEP1/2 Modes Start/Release

2.2.4.3 IDLE0 and SLEEP0 modes (IDLE0, SLEEP0)

IDLE0 and SLEEP0 modes are controlled by the system control register 2 (SYSCR2) and the time base timer control register (TBTCCR). The following status is maintained during IDLE0 and SLEEP0 modes.

1. Timing generator stops feeding clock to peripherals except TBT.
2. The data memory, CPU registers, program status word and port output latches are all held in the status in effect before IDLE0 and SLEEP0 modes were entered.
3. The program counter holds the address 2 ahead of the instruction which starts IDLE0 and SLEEP0 modes.

Note: Before starting IDLE0 or SLEEP0 mode, be sure to stop (Disable) peripherals.

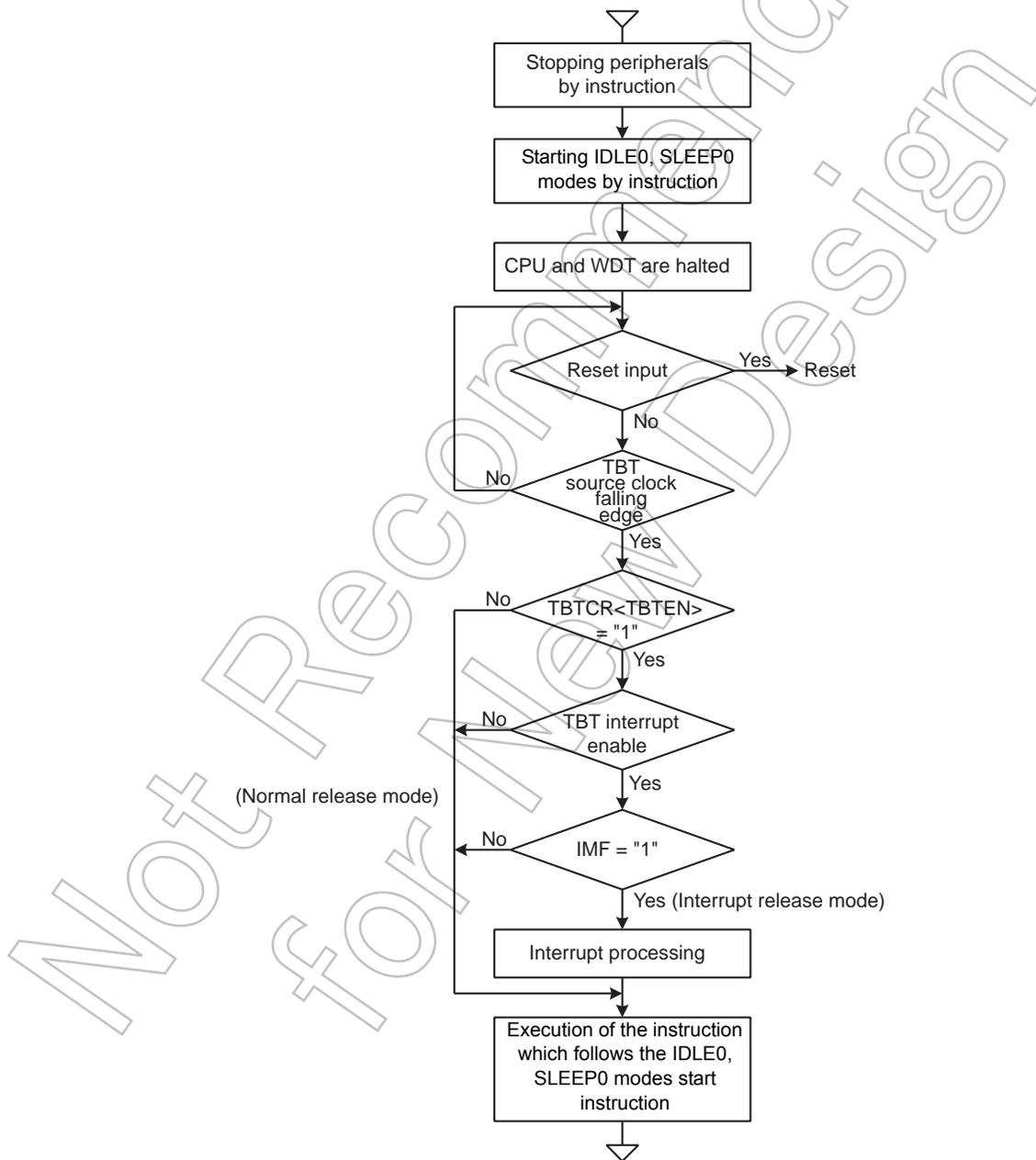


Figure 2-12 IDLE0 and SLEEP0 Modes

- Start the IDLE0 and SLEEP0 modes

Stop (Disable) peripherals such as a timer counter.

To start IDLE0 and SLEEP0 modes, set SYSCR2<TGHALT> to “1”.

- Release the IDLE0 and SLEEP0 modes

IDLE0 and SLEEP0 modes include a normal release mode and an interrupt release mode.

These modes are selected by interrupt master flag (IMF), the individual interrupt enable flag of TBT and TBTCR<TBTEN>.

After releasing IDLE0 and SLEEP0 modes, the SYSCR2<TGHALT> is automatically cleared to “0” and the operation mode is returned to the mode preceding IDLE0 and SLEEP0 modes. Before starting the IDLE0 or SLEEP0 mode, when the TBTCR<TBTEN> is set to “1”, INTTBT interrupt latch is set to “1”.

IDLE0 and SLEEP0 modes can also be released by inputting low level on the $\overline{\text{RESET}}$ pin. After releasing reset, the operation mode is started from NORMAL1 mode.

Note: IDLE0 and SLEEP0 modes start/release without reference to TBTCR<TBTEN> setting.

- (1) Normal release mode (IMF•EF6•TBTCR<TBTEN> = “0”)

IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK>. After the falling edge is detected, the program operation is resumed from the instruction following the IDLE0 and SLEEP0 modes start instruction. Before starting the IDLE0 or SLEEP0 mode, when the TBTCR<TBTEN> is set to “1”, INTTBT interrupt latch is set to “1”.

- (2) Interrupt release mode (IMF•EF6•TBTCR<TBTEN> = “1”)

IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK> and INTTBT interrupt processing is started.

Note 1: Because returning from IDLE0, SLEEP0 to NORMAL1, SLOW1 is executed by the asynchronous internal clock, the period of IDLE0, SLEEP0 mode might be the shorter than the period setting by TBTCR<TBTCK>.

Note 2: When a watchdog timer interrupt is generated immediately before IDLE0/SLEEP0 mode is started, the watchdog timer interrupt will be processed but IDLE0/SLEEP0 mode will not be started.

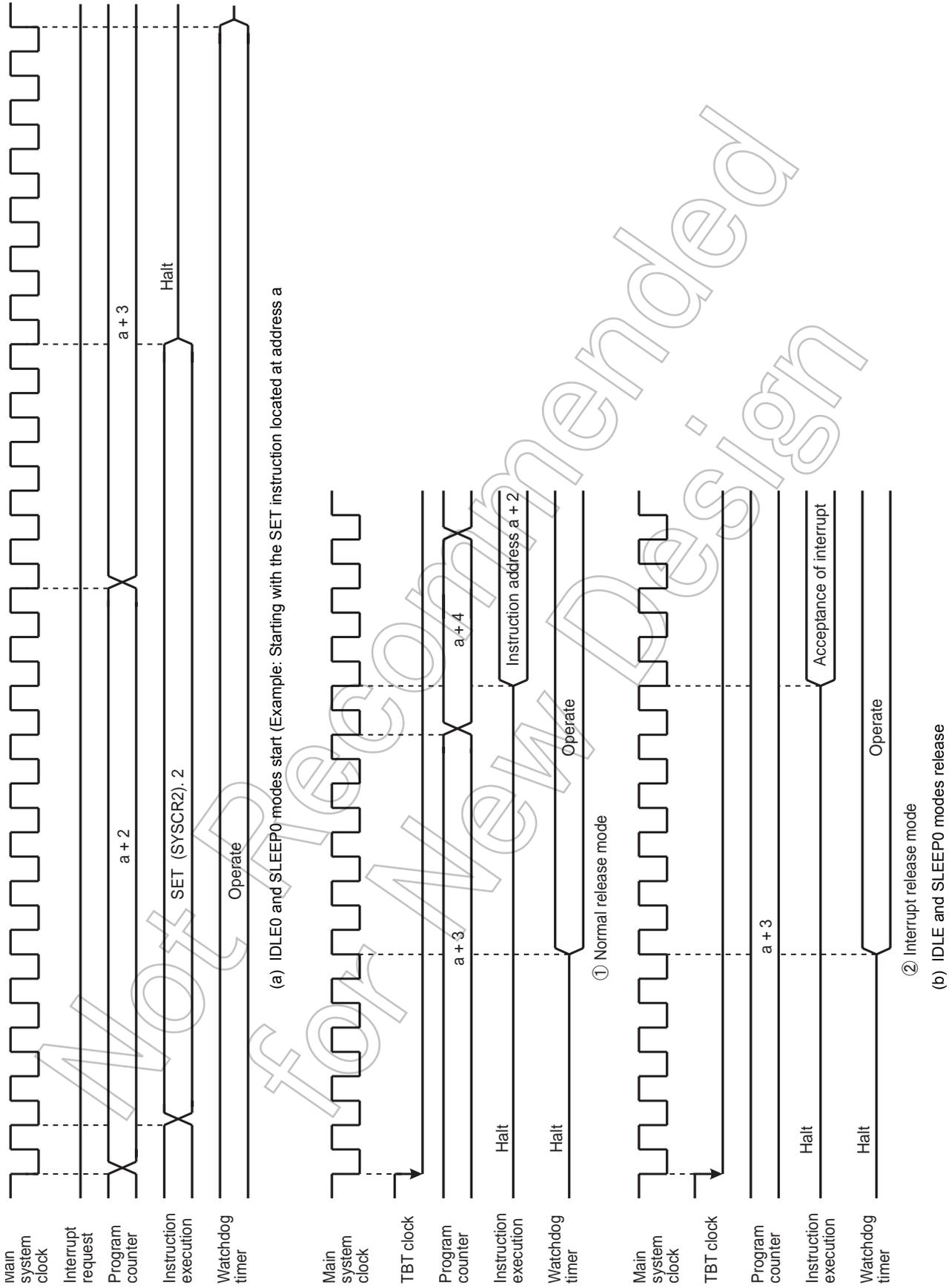


Figure 2-13 IDLE0 and SLEEP0 Modes Start/Release

2.2.4.4 SLOW mode

SLOW mode is controlled by the system control register 2 (SYSCR2).

The following is the methods to switch the mode with the warm-up counter.

(1) Switching from NORMAL2 mode to SLOW1 mode

First, set SYSCR2<SYSCK> to switch the main system clock to the low-frequency clock for SLOW2 mode. Next, clear SYSCR2<XEN> to turn off high-frequency oscillation.

Note: The high-frequency clock can be continued oscillation in order to return to NORMAL2 mode from SLOW mode quickly. Always turn off oscillation of high-frequency clock when switching from SLOW mode to stop mode.

Example 1 :Switching from NORMAL2 mode to SLOW1 mode.

```

SET      (SYSCR2). 5      ; SYSCR2<SYSCK> ← 1
                        ; (Switches the main system clock to the low-frequency
                        ; clock for SLOW2)

CLR      (SYSCR2). 7      ; SYSCR2<XEN> ← 0
                        ; (Turns off high-frequency oscillation)

```

Example 2 :Switching to the SLOW1 mode after low-frequency clock has stabilized.

```

SET      (SYSCR2). 6      ; SYSCR2<XTEN> ← 1

LD       (TC3CR), 43H     ; Sets mode for TC4, 3 (16-bit mode, fs for source)

LD       (TC4CR), 05H     ; Sets warming-up counter mode

LDW     (TTREG3), 8000H   ; Sets warm-up time (Depend on oscillator accompanied)

DI       ; IMF ← 0

SET      (EIRH). 1       ; Enables INTTC4

EI       ; IMF ← 1

SET      (TC4CR). 3       ; Starts TC4, 3

:

PINTTC4: CLR      (TC4CR). 3 ; Stops TC4, 3

SET      (SYSCR2). 5      ; SYSCR2<SYSCK> ← 1
                        ; (Switches the main system clock to the low-frequency clock)

CLR      (SYSCR2). 7      ; SYSCR2<XEN> ← 0
                        ; (Turns off high-frequency oscillation)

RETI

:

VINTTC4: DW       PINTTC4 ; INTTC4 vector table

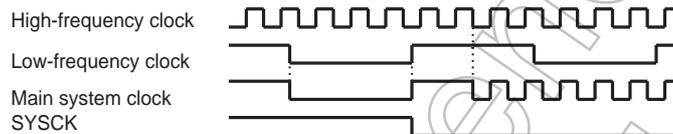
```

(2) Switching from SLOW1 mode to NORMAL2 mode

First, set SYSCR2<XEN> to turn on the high-frequency oscillation. When time for stabilization (Warm up) has been taken by the timer/counter (TC4,TC3), clear SYSCR2<SYSCK> to switch the main system clock to the high-frequency clock.

SLOW mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin. After releasing reset, the operation mode is started from NORMAL1 mode.

Note: After SYSCK is cleared to "0", executing the instructions is continued by the low-frequency clock for the period synchronized with low-frequency and high-frequency clocks.



Example :Switching from the SLOW1 mode to the NORMAL2 mode (fc = 16 MHz, warm-up time is 4.0 ms).

```

SET      (SYSCR2). 7      ; SYSCR2<XEN> ← 1 (Starts high-frequency oscillation)
LD       (TC3CR), 63H     ; Sets mode for TC4, 3 (16-bit mode, fc for source)
LD       (TC4CR), 05H     ; Sets warming-up counter mode
LD       (TTREG4), 0F8H   ; Sets warm-up time
DI       ; IMF ← 0
SET      (EIRH). 1       ; Enables INTTC4
EI       ; IMF ← 1
SET      (TC4CR). 3      ; Starts TC4, 3
:
PINTTC4: CLR      (TC4CR). 3      ; Stops TC4, 3
          CLR      (SYSCR2). 5     ; SYSCR2<SYSCK> ← 0
          ; (Switches the main system clock to the high-frequency clock)
          RETI
:
VINTTC4: DW       PINTTC4      ; INTTC4 vector table
    
```

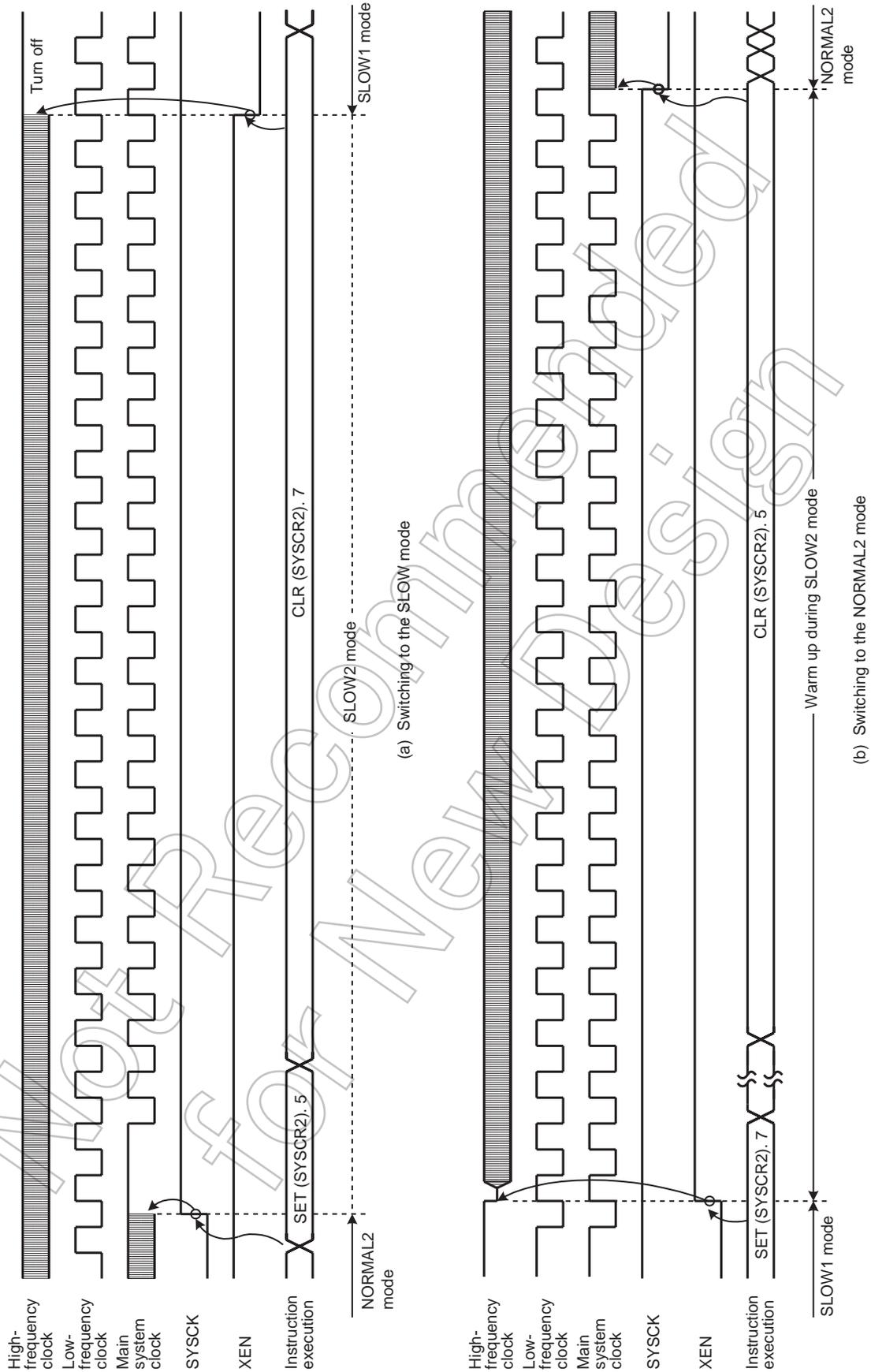


Figure 2-14 Switching between the NORMAL2 and SLOW Modes

2.3 Reset Circuit

The TMP86PS44UG has four types of reset generation procedures: An external reset input, an address trap reset, a watchdog timer reset and a system clock reset. Of these reset, the address trap reset, the watchdog timer and the system clock reset are a malfunction reset. When the malfunction reset request is detected, reset occurs during the maximum $24/f_c[s]$ (The $\overline{\text{RESET}}$ pin outputs "L" level).

The malfunction reset circuit such as watchdog timer reset, address trap reset and system clock reset is not initialized when power is turned on. Therefore, reset may occur during maximum $24/f_c[s]$ ($1.5\mu s$ at 16.0 MHz) when power is turned on. $\overline{\text{RESET}}$ pin outputs "L" level during maximum $24/f_c[s]$ ($1.5\mu s$ at 16.0MHz).

Table 2-3 shows on-chip hardware initialization by reset action.

Table 2-3 Initializing Internal Status by Reset Action

| On-chip Hardware | Initial Value | On-chip Hardware | Initial Value |
|--|-----------------|---|-----------------------------------|
| Program counter (PC) | (FFFEH) | Prescaler and divider of timing generator | 0 |
| Stack pointer (SP) | Not initialized | | |
| General-purpose registers (W, A, B, C, D, E, H, L, IX, IY) | Not initialized | | |
| Jump status flag (JF) | Not initialized | Watchdog timer | Enable |
| Zero flag (ZF) | Not initialized | Output latches of I/O ports | Refer to I/O port circuitry |
| Carry flag (CF) | Not initialized | | |
| Half carry flag (HF) | Not initialized | | |
| Sign flag (SF) | Not initialized | | |
| Overflow flag (VF) | Not initialized | | |
| Interrupt master enable flag (IMF) | 0 | | |
| Interrupt individual enable flags (EF) | 0 | Control registers | Refer to each of control register |
| Interrupt latches (IL) | 0 | | |
| | | RAM | Not initialized |

2.3.1 External Reset Input

The $\overline{\text{RESET}}$ pin contains a Schmitt trigger (Hysteresis) with an internal pull-up resistor.

When the $\overline{\text{RESET}}$ pin is held at "L" level for at least 3 machine cycles ($12/f_c [s]$) with the power supply voltage within the operating voltage range and oscillation stable, a reset is applied and the internal state is initialized.

When the $\overline{\text{RESET}}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses FFFEh to FFFFh.

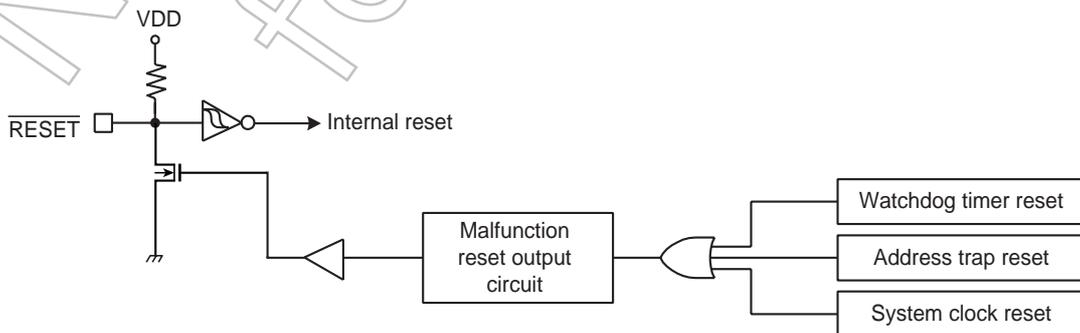
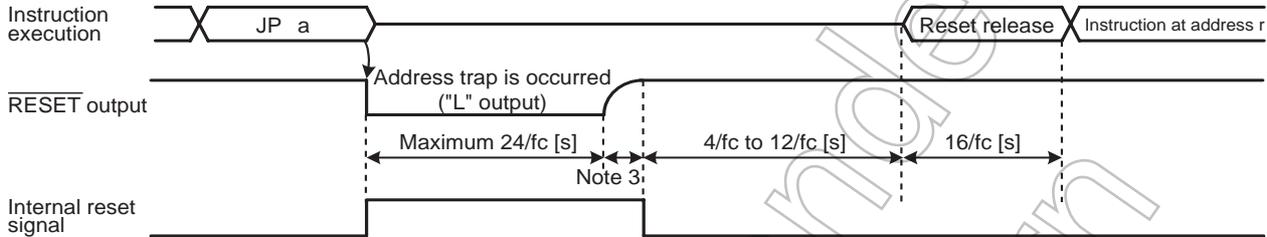


Figure 2-15 Reset Circuit

2.3.2 Address trap reset

If the CPU should start looping for some cause such as noise and an attempt be made to fetch an instruction from the on-chip RAM (when $WDT\overline{CR}1\langle ATAS \rangle$ is set to "1") or the SFR area, address trap reset will be generated. The reset time is maximum $24/f_c$ [s] (1.5 μ s at 16.0 MHz). Then, the \overline{RESET} pin outputs "L" level during maximum $24/f_c$ [s].

Note: The operating mode under address trapped is alternative of reset or interrupt. The address trap area is alternative.



Note 1: Address "a" is in the SFR or on-chip RAM ($WDT\overline{CR}1\langle ATAS \rangle = "1"$) space.

Note 2: During reset release, reset vector "r" is read out, and an instruction at address "r" is fetched and decoded.

Note 3: Varies on account of external condition: voltage or capacitance

Figure 2-16 Address Trap Reset

2.3.3 Watchdog timer reset

Refer to Section "Watchdog Timer".

2.3.4 System clock reset

If the condition as follows is detected, the system clock reset occurs automatically to prevent dead lock of the CPU. (The oscillation is continued without stopping.)

- In case of clearing $SY\overline{SCR}2\langle XEN \rangle$ and $SY\overline{SCR}2\langle XTEN \rangle$ simultaneously to "0".
- In case of clearing $SY\overline{SCR}2\langle XEN \rangle$ to "0", when the $SY\overline{SCR}2\langle SY\overline{SCK} \rangle$ is "0".
- In case of clearing $SY\overline{SCR}2\langle XTEN \rangle$ to "0", when the $SY\overline{SCR}2\langle SY\overline{SCK} \rangle$ is "1".

The reset time is maximum $24/f_c$ (1.5 μ s at 16.0 MHz). Then, the \overline{RESET} pin outputs "L" level during maximum $24/f_c$ [s] (1.5 μ s at 16.0MHz).

Not Recommended
for New Design

Not Recommended
for New Design

3. Interrupt Control Circuit

The TMP86PS44UG has a total of 19 interrupt sources excluding reset, of which 3 source levels are multiplexed. Interrupts can be nested with priorities. Four of the internal interrupt sources are non-maskable while the rest are maskable.

Interrupt sources are provided with interrupt latches (IL), which hold interrupt requests, and independent vectors. The interrupt latch is set to “1” by the generation of its interrupt request which requests the CPU to accept its interrupts. Interrupts are enabled or disabled by software using the interrupt master enable flag (IMF) and interrupt enable flag (EF). If more than one interrupts are generated simultaneously, interrupts are accepted in order which is dominated by hardware. However, there are no prioritized interrupt factors among non-maskable interrupts.

| Interrupt Factors | | Enable Condition | Interrupt Latch | Vector Address | Priority |
|-------------------|---|--------------------------|-----------------|----------------|----------|
| Internal/External | (Reset) | Non-maskable | – | FFFE | 1 |
| Internal | INTSWI (Software interrupt) | Non-maskable | – | FFFC | 2 |
| Internal | INTUNDEF (Executed the undefined instruction interrupt) | Non-maskable | – | FFFC | 2 |
| Internal | INTATRAP (Address trap interrupt) | Non-maskable | IL2 | FFFA | 2 |
| Internal | INTWDT (Watchdog timer interrupt) | Non-maskable | IL3 | FFF8 | 2 |
| External | $\overline{INT0}$ | IMF•EF4 = 1, INTOEN = 1 | IL4 | FFF6 | 5 |
| External | INT1 | IMF•EF5 = 1 | IL5 | FFF4 | 6 |
| Internal | INTTBT | IMF•EF6 = 1 | IL6 | FFF2 | 7 |
| Internal | INTTC1 | IMF•EF7 = 1 | IL7 | FFF0 | 8 |
| External | INT2 | IMF•EF8 = 1 | IL8 | FFEE | 9 |
| Internal | INTTC4 | IMF•EF9 = 1 | IL9 | FFEC | 10 |
| Internal | INTTC3 | IMF•EF10 = 1 | IL10 | FFEA | 11 |
| External | INT3 | IMF•EF11 = 1 | IL11 | FFE8 | 12 |
| Internal | INTSIO | IMF•EF12 = 1 | IL12 | FFE6 | 13 |
| Internal | INTRXD | IMF•EF13 = 1, IL13ER = 0 | IL13 | FFE4 | 14 |
| Internal | INTTC2 | IMF•EF13 = 1, IL13ER = 1 | | | |
| External | INT4 | IMF•EF14 = 1, IL14ER = 0 | IL14 | FFE2 | 15 |
| Internal | INTTXD | IMF•EF14 = 1, IL14ER = 1 | | | |
| External | $\overline{INT5}$ | IMF•EF15 = 1, IL15ER = 0 | IL15 | FFE0 | 16 |
| Internal | INTADC | IMF•EF15 = 1, IL15ER = 1 | | | |

Note 1: The INTSEL register is used to select the interrupt source to be enabled for each multiplexed source level (see 3.3 Interrupt Source Selector (INTSEL)).

Note 2: To use the address trap interrupt (INTATRAP), clear WDTTCR1<ATOUT> to “0” (It is set for the “reset request” after reset is cancelled). For details, see “Address Trap”.

Note 3: To use the watchdog timer interrupt (INTWDT), clear WDTTCR1<WDTOUT> to “0” (It is set for the “Reset request” after reset is released). For details, see “Watchdog Timer”.

3.1 Interrupt latches (IL15 to IL2)

An interrupt latch is provided for each interrupt source, except for a software interrupt and an executed the undefined instruction interrupt. When interrupt request is generated, the latch is set to “1”, and the CPU is requested to accept the interrupt if its interrupt is enabled. The interrupt latch is cleared to “0” immediately after accepting interrupt. All interrupt latches are initialized to “0” during reset.

The interrupt latches are located on address 003CH and 003DH in SFR area. Each latch can be cleared to “0” individually by instruction. However, IL2 and IL3 should not be cleared to “0” by software. For clearing the interrupt latch, load instruction should be used and then IL2 and IL3 should be set to “1”. If the read-modify-write instructions such as bit manipulation or operation instructions are used, interrupt request would be cleared inadequately if interrupt is requested while such instructions are executed.

Interrupt latches are not set to “1” by an instruction.

Since interrupt latches can be read, the status for interrupt requests can be monitored by software.

Note: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to “0” (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)

In interrupt service routine, because the IMF becomes “0” automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF=“1”.

Example 1 :Clears interrupt latches

```
DI                                ; IMF ← 0
LDW    (ILL), 1110100000111111B  ; IL12, IL10 to IL6 ← 0
EI                                ; IMF ← 1
```

Example 2 :Reads interrupt latches

```
LD    WA, (ILL)                  ; W ← ILH, A ← ILL
```

Example 3 :Tests interrupt latches

```
TEST    (ILL), 7                  ; if IL7 = 1 then jump
JR      F, SSET
```

3.2 Interrupt enable register (EIR)

The interrupt enable register (EIR) enables and disables the acceptance of interrupts, except for the non-maskable interrupts (Software interrupt, undefined instruction interrupt, address trap interrupt and watchdog interrupt). Non-maskable interrupt is accepted regardless of the contents of the EIR.

The EIR consists of an interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are located on address 003AH and 003BH in SFR area, and they can be read and written by an instructions (Including read-modify-write instructions such as bit manipulation or operation instructions).

3.2.1 Interrupt master enable flag (IMF)

The interrupt enable register (IMF) enables and disables the acceptance of the whole maskable interrupt. While IMF = “0”, all maskable interrupts are not accepted regardless of the status on each individual interrupt enable flag (EF). By setting IMF to “1”, the interrupt becomes acceptable if the individuals are enabled. When an interrupt is accepted, IMF is cleared to “0” after the latest status on IMF is stacked. Thus the maskable interrupts which follow are disabled. By executing return interrupt instruction [RETI/RETN], the stacked data, which was the status before interrupt acceptance, is loaded on IMF again.

The IMF is located on bit0 in EIRL (Address: 003AH in SFR), and can be read and written by an instruction. The IMF is normally set and cleared by [EI] and [DI] instruction respectively. During reset, the IMF is initialized to “0”.

3.2.2 Individual interrupt enable flags (EF15 to EF4)

Each of these flags enables and disables the acceptance of its maskable interrupt. Setting the corresponding bit of an individual interrupt enable flag to “1” enables acceptance of its interrupt, and setting the bit to “0” disables acceptance. During reset, all the individual interrupt enable flags (EF15 to EF4) are initialized to “0” and all maskable interrupts are not accepted until they are set to “1”.

Note: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to “0” (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)
 In interrupt service routine, because the IMF becomes “0” automatically, clearing IMF need not execute nor-

mally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

Example 1 :Enables interrupts individually and sets IMF

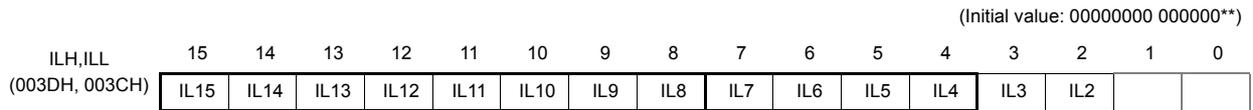
```
DI ; IMF ← 0
LDW (EIRL), 1110100010100000B ; EF15 to EF13, EF11, EF7, EF5 ← 1
: ; Note: IMF should not be set.
:
EI ; IMF ← 1
```

Example 2 :C compiler description example

```
unsigned int _io (3AH) EIRL; /* 3AH shows EIRL address */
_DI();
EIRL = 10100000B;
:
_EI();
```

Not Recommended for New Design

Interrupt Latches



ILH (003DH)

ILL (003CH)

| | | | | |
|-------------|-------------------|--|--|-----|
| IL15 to IL2 | Interrupt latches | at RD 0: No interrupt request 1: Interrupt request | at WR 0: Clears the interrupt request 1: (Interrupt latch is not set.) | R/W |
|-------------|-------------------|--|--|-----|

Note 1: To clear any one of bits IL7 to IL4, be sure to write "1" into IL2 and IL3.

Note 2: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)

In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

Note 3: Do not clear IL with read-modify-write instructions such as bit operations.

Interrupt Enable Registers



EIRH (003BH)

EIRL (003AH)

| | | | |
|-------------|--|---|-----|
| EF15 to EF4 | Individual-interrupt enable flag (Specified for each bit) | 0: Disables the acceptance of each maskable interrupt. 1: Enables the acceptance of each maskable interrupt. | R/W |
| IMF | Interrupt master enable flag | 0: Disables the acceptance of all maskable interrupts 1: Enables the acceptance of all maskable interrupts | |

Note 1: *: Don't care

Note 2: Do not set IMF and the interrupt enable flag (EF15 to EF4) to "1" at the same time.

Note 3: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)

In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

3.3 Interrupt Source Selector (INTSEL)

Each interrupt source that shares the interrupt source level with another interrupt source is allowed to enable the interrupt latch only when it is selected in the INTSEL register. The interrupt controller does not hold interrupt requests corresponding to interrupt sources that are not selected in the INTSEL register. Therefore, the INTSEL register must be set appropriately before interrupt requests are generated.

The following interrupt sources share their interrupt source level; the source is selected on the register INTSEL.

1. INTRXD and INTTC2 share the interrupt source level whose priority is 14.
2. INT4 and INTTXD share the interrupt source level whose priority is 15.
3. $\overline{\text{INT5}}$ and INTADC share the interrupt source level whose priority is 16.

Interrupt source selector

| | | | | | | | | | |
|-------------------|---|---|---|---|---|--------|--------|--------|----------------------------|
| INTSEL (003EH) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: **** *000) |
| | - | - | - | - | - | IL13ER | IL14ER | IL15ER | |

| | | | |
|--------|--|--|-----|
| IL13ER | Selects INTRXD or INTTC2 | 0: INTRXD 1: INTTC2 | R/W |
| IL14ER | Selects INT4 or INTTXD | 0: INT4 1: INTTXD | R/W |
| IL15ER | Selects $\overline{\text{INT5}}$ or INTADC | 0: $\overline{\text{INT5}}$ 1: INTADC | R/W |

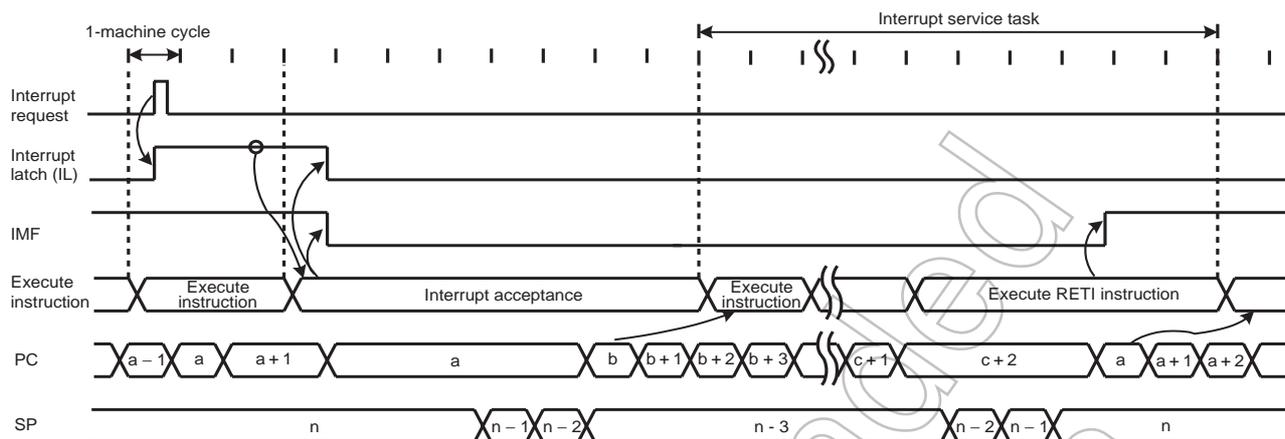
3.4 Interrupt Sequence

An interrupt request, which raised interrupt latch, is held, until interrupt is accepted or interrupt latch is cleared to “0” by resetting or an instruction. Interrupt acceptance sequence requires 8 machine cycles (2 μs @16 MHz) after the completion of the current instruction. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for non-maskable interrupts). Figure 3-1 shows the timing chart of interrupt acceptance processing.

3.4.1 Interrupt acceptance processing is packaged as follows.

- a. The interrupt master enable flag (IMF) is cleared to “0” in order to disable the acceptance of any following interrupt.
- b. The interrupt latch (IL) for the interrupt source accepted is cleared to “0”.
- c. The contents of the program counter (PC) and the program status word, including the interrupt master enable flag (IMF), are saved (Pushed) on the stack in sequence of PSW + IMF, PCH, PCL. Meanwhile, the stack pointer (SP) is decremented by 3.
- d. The entry address (Interrupt vector) of the corresponding interrupt service program, loaded on the vector table, is transferred to the program counter.
- e. The instruction stored at the entry address of the interrupt service program is executed.

Note: When the contents of PSW are saved on the stack, the contents of IMF are also saved.



Note 1: a: Return address entry address, b: Entry address, c: Address which RETI instruction is stored
 Note 2: On condition that interrupt is enabled, it takes $38/f_c$ [s] or $38/f_s$ [s] at maximum (If the interrupt latch is set at the first machine cycle on 10 cycle instruction) to start interrupt acceptance processing since its interrupt latch is set.

Figure 3-1 Timing Chart of Interrupt Acceptance/Return Interrupt Instruction

Example: Correspondence between vector table address for INTTBT and the entry address of the interrupt service program

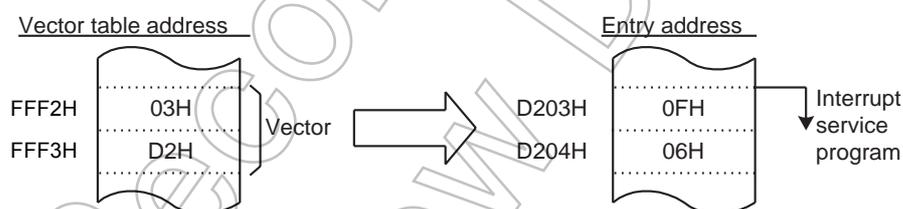


Figure 3-2 Vector table address,Entry address

A maskable interrupt is not accepted until the IMF is set to "1" even if the maskable interrupt higher than the level of current servicing interrupt is requested.

In order to utilize nested interrupt service, the IMF is set to "1" in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags.

To avoid overloaded nesting, clear the individual interrupt enable flag whose interrupt is currently serviced, before setting IMF to "1". As for non-maskable interrupt, keep interrupt service shorten compared with length between interrupt requests; otherwise the status cannot be recovered as non-maskable interrupt would simply nested.

3.4.2 Saving/restoring general-purpose registers

During interrupt acceptance processing, the program counter (PC) and the program status word (PSW, includes IMF) are automatically saved on the stack, but the accumulator and others are not. These registers are saved by software if necessary. When multiple interrupt services are nested, it is also necessary to avoid using the same data memory area for saving registers. The following methods are used to save/restore the general-purpose registers.

3.4.2.1 Using PUSH and POP instructions

If only a specific register is saved or interrupts of the same source are nested, general-purpose registers can be saved/restored using the PUSH/POP instructions.

Example :Save/store register using PUSH and POP instructions

```
PINTxx:    PUSH    WA           ; Save WA register
           (interrupt processing)
           POP     WA           ; Restore WA register
           RETI    ; RETURN
```

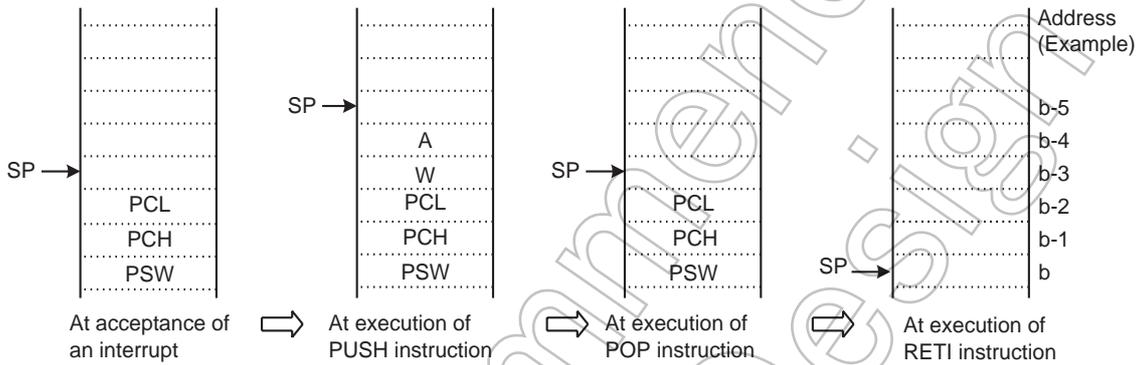


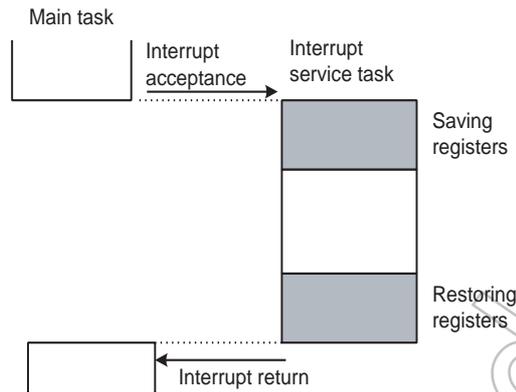
Figure 3-3 Save/store register using PUSH and POP instructions

3.4.2.2 Using data transfer instructions

To save only a specific register without nested interrupts, data transfer instructions are available.

Example :Save/store register using data transfer instructions

```
PINTxx:    LD      (GSAVA), A   ; Save A register
           (interrupt processing)
           LD      A, (GSAVA)   ; Restore A register
           RETI    ; RETURN
```



Saving/Restoring general-purpose registers using PUSH/POP data transfer instruction

Figure 3-4 Saving/Restoring General-purpose Registers under Interrupt Processing

3.4.3 Interrupt return

Interrupt return instructions [RETI]/[RETN] perform as follows.

| [RETI]/[RETN] Interrupt Return |
|--|
| 1. Program counter (PC) and program status word (PSW, includes IMF) are restored from the stack. |
| 2. Stack pointer (SP) is incremented by 3. |

As for address trap interrupt (INTATRAP), it is required to alter stacked data for program counter (PC) to restarting address, during interrupt service program.

Note: If [RETN] is executed with the above data unaltered, the program returns to the address trap area and INTATRAP occurs again. When interrupt acceptance processing has completed, stacked data for PCL and PCH are located on address (SP + 1) and (SP + 2) respectively.

Example 1 :Returning from address trap interrupt (INTATRAP) service program

```
PINTxx:    POP        WA                ; Recover SP by 2
           LD         WA, Return Address ;
           PUSH       WA                ; Alter stacked data
           (interrupt processing)
           RETN      ; RETURN
```

Example 2 :Restarting without returning interrupt
(In this case, PSW (Includes IMF) before interrupt acceptance is discarded.)

```
PINTxx:    INC        SP                ; Recover SP by 3
           INC        SP                ;
           INC        SP                ;
           (interrupt processing)
           LD         EIRL, data        ; Set IMF to "1" or clear it to "0"
           JP         Restart Address   ; Jump into restarting address
```

Interrupt requests are sampled during the final cycle of the instruction being executed. Thus, the next interrupt can be accepted immediately after the interrupt return instruction is executed.

Note 1: It is recommended that stack pointer be return to rate before INTATRAP (Increment 3 times), if return interrupt instruction [RETN] is not utilized during interrupt service program under INTATRAP (such as Example 2).

Note 2: When the interrupt processing time is longer than the interrupt request generation time, the interrupt service task is performed but not the main task.

3.5 Software Interrupt (INTSW)

Executing the SWI instruction generates a software interrupt and immediately starts interrupt processing (INTSW is highest prioritized interrupt).

Use the SWI instruction only for detection of the address error or for debugging.

3.5.1 Address error detection

FFH is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address during single chip mode. Code FFH is the SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing FFH to unused areas of the program memory. Address trap reset is generated in case that an instruction is fetched from RAM or SFR areas.

3.5.2 Debugging

Debugging efficiency can be increased by placing the SWI instruction at the software break point setting address.

3.6 Undefined Instruction Interrupt (INTUNDEF)

Taking code which is not defined as authorized instruction for instruction causes INTUNDEF. INTUNDEF is generated when the CPU fetches such a code and tries to execute it. INTUNDEF is accepted even if non-maskable interrupt is in process. Contemporary process is broken and INTUNDEF interrupt process starts, soon after it is requested.

Note: The undefined instruction interrupt (INTUNDEF) forces CPU to jump into vector address, as software interrupt (SWI) does.

3.7 Address Trap Interrupt (INTATRAP)

Fetching instruction from unauthorized area for instructions (Address trapped area) causes reset output or address trap interrupt (INTATRAP). INTATRAP is accepted even if non-maskable interrupt is in process. Contemporary process is broken and INTATRAP interrupt process starts, soon after it is requested.

Note: The operating mode under address trapped, whether to be reset output or interrupt processing, is selected on watchdog timer control register (WDTCR).

3.8 External Interrupts

The TMP86PS44UG has 6 external interrupt inputs. These inputs are equipped with digital noise reject circuits (Pulse inputs of less than a certain time are eliminated as noise).

Edge selection is also possible with INT1 to INT4. The $\overline{\text{INT0}}/\text{P00}$ pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

Edge selection, noise reject control and $\overline{\text{INT0}}/\text{P00}$ pin function selection are performed by the external interrupt control register (EINTCR).

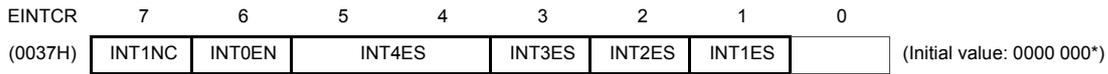
| Source | Pin | Enable Conditions | Release Edge (level) | Digital Noise Reject |
|--------|--------------------------|-----------------------------------|---|---|
| INT0 | $\overline{\text{INT0}}$ | IMF • EF4 • INT0EN=1 | Falling edge | Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals. |
| INT1 | INT1 | IMF • EF5 = 1 | Falling edge or Rising edge | Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of 49/fc or 193/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals. |
| INT2 | INT2 | IMF • EF8 = 1 | Falling edge or Rising edge | Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals. |
| INT3 | INT3 | IMF • EF11 = 1 | Falling edge or Rising edge | Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals. |
| INT4 | INT4 | IMF • EF14 = 1 and IL14ER=0 | Falling edge, Rising edge, Falling and Rising edge or H level | Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals. |
| INT5 | $\overline{\text{INT5}}$ | IMF • EF15 = 1 and IL15ER=0 | Falling edge | Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals. |

Note 1: In NORMAL1/2 or IDLE1/2 mode, if a signal with no noise is input on an external interrupt pin, it takes a maximum of "signal establishment time + 6/fs[s]" from the input signal's edge to set the interrupt latch.

Note 2: When INT0EN = "0", IL4 is not set even if a falling edge is detected on the $\overline{\text{INT0}}$ pin input.

Note 3: When a pin with more than one function is used as an output and a change occurs in data or input/output status, an interrupt request signal is generated in a pseudo manner. In this case, it is necessary to perform appropriate processing such as disabling the interrupt enable flag.

External Interrupt Control Register



| | | | |
|---------|---|--|-----|
| INT1NC | Noise reject time select | 0: Pulses of less than $63/f_c$ [s] are eliminated as noise 1: Pulses of less than $15/f_c$ [s] are eliminated as noise | R/W |
| INT0EN | P00/ $\overline{\text{INT0}}$ pin configuration | 0: P00 input/output port 1: $\overline{\text{INT0}}$ pin (Port P00 should be set to an input mode) | R/W |
| INT4 ES | INT4 edge select | 00: Rising edge 01: Falling edge 10: Rising edge and Falling edge 11: H level | R/W |
| INT3 ES | INT3 edge select | 0: Rising edge 1: Falling edge | R/W |
| INT2 ES | INT2 edge select | 0: Rising edge 1: Falling edge | R/W |
| INT1 ES | INT1 edge select | 0: Rising edge 1: Falling edge | R/W |

Note 1: f_c : High-frequency clock [Hz], *: Don't care

Note 2: When the system clock frequency is switched between high and low or when the external interrupt control register (EINTCR) is overwritten, the noise canceller may not operate normally. It is recommended that external interrupts are disabled using the interrupt enable register (EIR).

Note 3: The maximum time from modifying INT1NC until a noise reject time is changed is $2^6/f_c$.

Note 4: In case $\overline{\text{RESET}}$ pin is released while the state of INT4 pin keeps "H" level, the external interrupt 4 request is not generated even if the INT4 edge select is specified as "H" level. The rising edge is needed after $\overline{\text{RESET}}$ pin is released.

Not Recommended for New Design

Not Recommended
for New Design

4. Special Function Register (SFR)

The TMP86PS44UG adopts the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function register (SFR). The SFR is mapped on address 0000H to 003FH.

This chapter shows the arrangement of the special function register (SFR) for TMP86PS44UG.

4.1 SFR

| Address | Read | Write |
|---------|--------|----------|
| 0000H | | P0DR |
| 0001H | | P1DR |
| 0002H | | P2DR |
| 0003H | | P3DR |
| 0004H | | P4DR |
| 0005H | | Reserved |
| 0006H | | Reserved |
| 0007H | | Reserved |
| 0008H | P0PRD | - |
| 0009H | | Reserved |
| 000AH | P2PRD | - |
| 000BH | | Reserved |
| 000CH | | Reserved |
| 000DH | | P1CR |
| 000EH | | P3CR |
| 000FH | | P4CR |
| 0010H | | TC1DRAL |
| 0011H | | TC1DRAH |
| 0012H | | TC1DRBL |
| 0013H | | TC1DRBH |
| 0014H | | TC1CR |
| 0015H | | Reserved |
| 0016H | | TC3CR |
| 0017H | | TC4CR |
| 0018H | | TTREG3 |
| 0019H | | TTREG4 |
| 001AH | | PWREG3 |
| 001BH | | PWREG4 |
| 001CH | | ADCCR1 |
| 001DH | | ADCCR2 |
| 001EH | ADCDR2 | - |
| 001FH | ADCDR1 | - |
| 0020H | UARTSR | UARTCR1 |
| 0021H | - | UARTCR2 |
| 0022H | RDBUF | TDBUF |
| 0023H | | TC2CR |
| 0024H | | TC2DRL |
| 0025H | | TC2DRH |
| 0026H | | SIOCR1 |
| 0027H | | SIOSR |

| Address | Read | Write |
|---------|--------|----------|
| 0028H | SIORDB | SIOTDB |
| 0029H | | Reserved |
| 002AH | | Reserved |
| 002BH | | Reserved |
| 002CH | | Reserved |
| 002DH | | Reserved |
| 002EH | | DACCR |
| 002FH | | DACDR |
| 0030H | | Reserved |
| 0031H | - | STOPCR |
| 0032H | | Reserved |
| 0033H | | Reserved |
| 0034H | - | WDTCR1 |
| 0035H | - | WDTCR2 |
| 0036H | | TBTCR |
| 0037H | | EINTCR |
| 0038H | | SYSCR1 |
| 0039H | | SYSCR2 |
| 003AH | | EIRL |
| 003BH | | EIRH |
| 003CH | | ILL |
| 003DH | | ILH |
| 003EH | | INTSEL |
| 003FH | | PSW |

Note 1: Do not access reserved areas by the program.

Note 2: - ; Cannot be accessed.

Note 3: Write-only registers and interrupt latches cannot use the read-modify-write instructions (Bit manipulation instructions such as SET, CLR, etc. and logical operation instructions such as AND, OR, etc.).

Not Recommended for New Design

5. I/O Ports

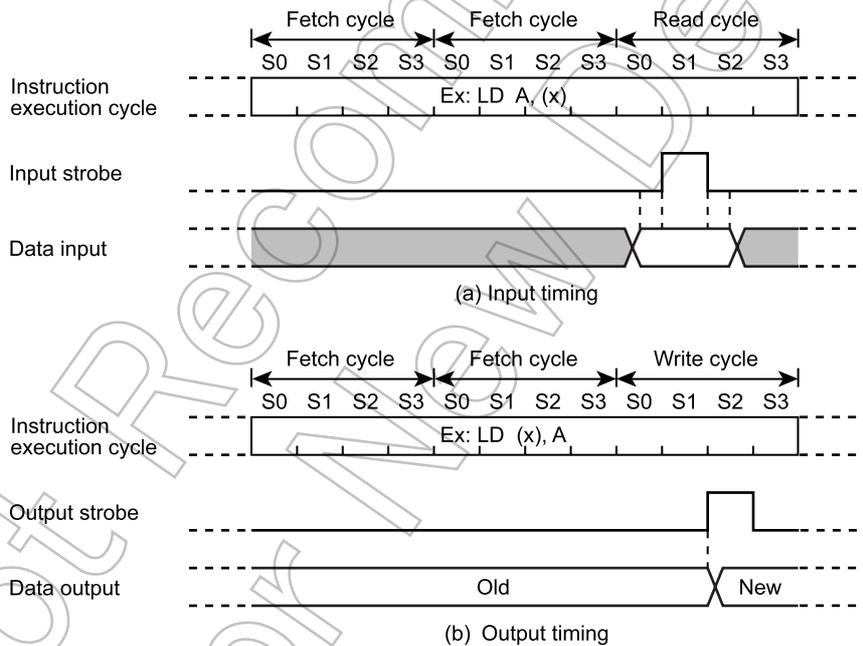
The TMP86PS44UG has 5 parallel input/output ports (35 pins) as follows.

| | Primary Function | Secondary Functions |
|---------|------------------|--|
| Port P0 | 8-bit I/O port | External interrupt input, serial and timer/counter input/output |
| Port P1 | 8-bit I/O port | External interrupt input, timer/counter input/output, and divider output |
| Port P2 | 3-bit I/O port | External interrupt input, and STOP mode release signal input |
| Port P3 | 8-bit I/O port | Analog input, and STOP mode release signal input |
| Port P4 | 8-bit I/O port | |

Each output port contains a latch, which holds the output data. All input ports do not have latches, so the external input data should be externally held until the input data is read from outside or reading should be performed several times before processing. Figure 5-1 shows input/output timing examples.

External data is read from an I/O port in the S1 state of the read cycle during execution of the read instruction. This timing cannot be recognized from outside, so that transient input such as chattering must be processed by the program.

Output data changes in the S2 state of the write cycle during execution of the instruction which writes to an I/O port.



Note: The positions of the read and write cycles may vary, depending on the instruction.

Figure 5-1 Input/Output Timing (Example)

5.1 Port P0 (P07 to P00)

Port P0 is an 8-bit input/output port which is also used as an external interrupt input, serial interface input/output and timer/counter input/output.

When used as an input port or a secondary function pins, the respective output latch (P0DR) should be set to “1”. When used as an output port, the respective P0DR bit should be set data. During reset, the output latch is initialized to “1”.

P0 port output latch (P0DR) and P0 port terminal input (P0PRD) are located on their respective address.

When read the output latch data, the P0DR should be read and when read the terminal input data, the P0PRD register should be read. P00 port ($\overline{\text{INT0}}$) can be configured as either an I/O port or as external interrupt input with INT0EN (bit 6 in EINTCR). During reset, P00 port ($\overline{\text{INT0}}$) is configured as an input port.

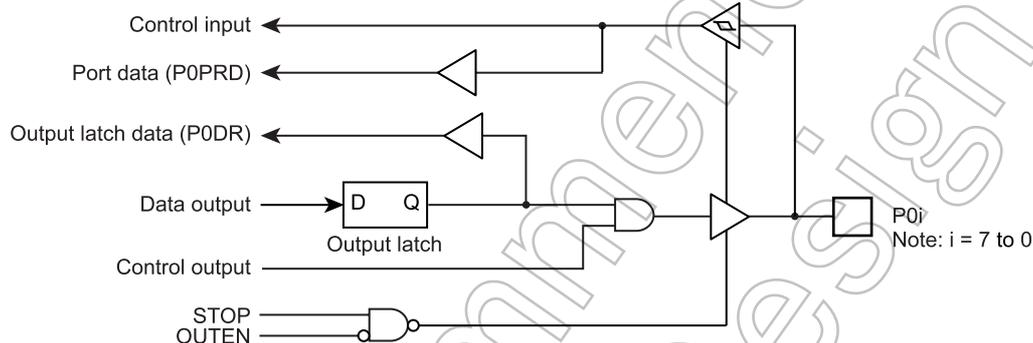


Figure 5-2 Port P0

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------------------------------|-------------|------------|-----------|-----------|------------|------------|------------------------------------|---------------------------------|----------------------------|
| P0DR (0000H) R/W | P07 INT4 | P06 SCK | P05 SI | P04 SO | P03 TXD | P02 RXD | P01 PWM4 PDO4 PPG4 TC4 | P00 $\overline{\text{INT0}}$ | (Initial value: 1111 1111) |
| P0PRD (0008H) Read only | P07 | P06 | P05 | P04 | P03 | P02 | P01 | P00 | |

5.3 Port P2 (P22 to P20)

Port P2 is a 3-bit input/output port.

It is also used as an external interrupt, a STOP mode release signal input, and low-frequency crystal oscillator connection pins. When used as an input port or a secondary function pins, respective output latch (P2DR) should be set to "1".

During reset, the P2DR is initialized to "1".

A low-frequency crystal oscillator (32.768 kHz) is connected to pins P21 (XTIN) and P22 (XTOUT) in the dual-clock mode. In the single-clock mode, pins P21 and P22 can be used as normal input/output ports.

It is recommended that pin P20 should be used as an external interrupt input, a STOP mode release signal input, or an input port. If it is used as an output port, the interrupt latch is set on the falling edge of the output pulse.

P2 port output latch (P2DR) and P2 port terminal input (P2PRD) are located on their respective address.

When read the output latch data, the P2DR should be read and when read the terminal input data, the P2PRD register should be read. If a read instruction is executed for port P2, read data of bits 7 to 3 are unstable.

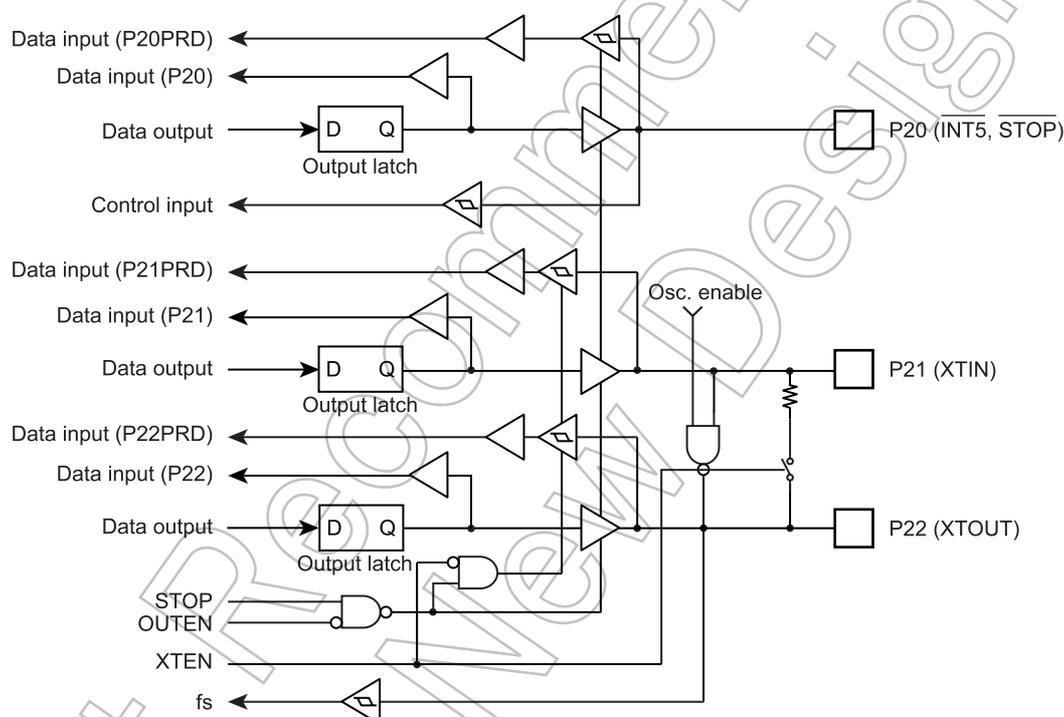


Figure 5-4 Port P2

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------------------------------|---|---|---|---|---|--------------|-------------|---------------------|----------------------------|
| P2DR (0002H) R/W | | | | | | P22 XTOUT | P21 XTIN | P20 INT5 STOP | (Initial value: **** *111) |
| P2PRD (000AH) Read only | | | | | | P22 | P21 | P20 | |

5.4 Port P3 (P37 to P30)

Port P3 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Port P3 is also used as an analog input, key on wake up input. Input/output mode is specified by the corresponding bit in the port P3 input/output control register (P3CR), and AINDS (bit 4 in ADCCR1). During reset, P3CR are initialized to “0” and AINDS is set to “1”, therefore port P3 is configured as an input.

When used as an analog input, set an analog input channel to SAIN (bit 0, 1, 2 in ADCCR1) and clear AINDS to “0”. When AINDS is “0”, the pin which is specified as an analog input is used as analog input independent on the value of P3CR and P3DR.

When used as an input port or key on wake up input, the corresponding bit of P3CR is cleared to “0” without specifying as an analog input.

When the AD converter is enabled (AINDS is “0”), the data of port which is selected as an analog input is read “0”. and the data of port which is not selected as an analog input is read “0” or “1”, depend on the voltage level.

When used as an output port, the corresponding bit of P3CR is set to “1” without specifying as an analog input. Data can be written into the output latch regardless of P3CR contents, therefore initial output data should be written into the output latch before setting P3CR.

The pins not used as analog input can be used as an input/output port. But output instructions should not be executed to keep a precision. In addition, a variable signal should not be input to an adjacent port to the analog input during AD conversion.

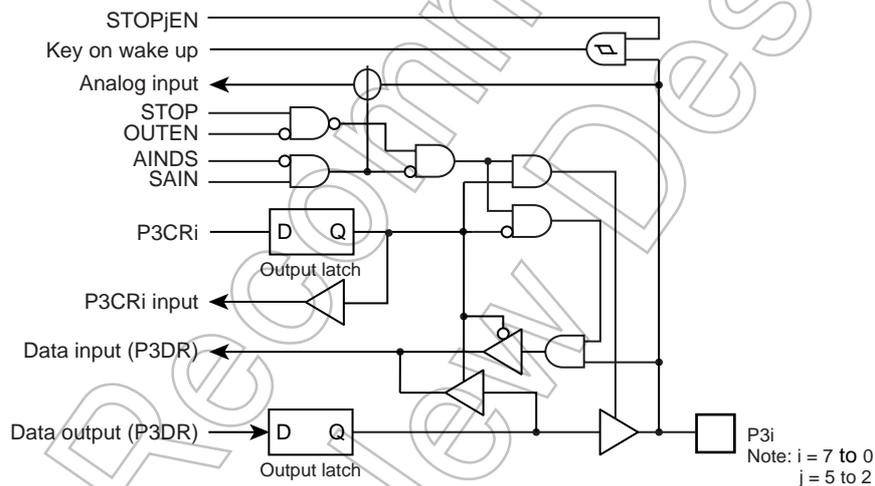


Figure 5-5 Port P3

| | | | | | | | | | |
|------------------------|----------------------|----------------------|----------------------|----------------------|-------------|-------------|-------------|-------------|----------------------------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| P3DR (0003H) R/W | P37 AIN7 STOP5 | P36 AIN6 STOP4 | P35 AIN5 STOP3 | P34 AIN4 STOP2 | P33 AIN3 | P32 AIN2 | P31 AIN1 | P30 AIN0 | (Initial value: 0000 0000) |

| | | | | | | | | | |
|-----------------|---|---|---|---|---|---|---|---|----------------------------|
| P3CR (000EH) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: 0000 0000) |
|-----------------|---|---|---|---|---|---|---|---|----------------------------|

| | | | |
|------|---|---------------------------------|-----|
| P3CR | I/O control (Specified for each bit) | 0: Input mode 1: Output mode | R/W |
|------|---|---------------------------------|-----|

Note: Ports set to the input mode read the pin states. Ports set to the output mode read the output latch. When input pin and output pin exist in port P3 together, the contents of the output latch which is specified as an input mode may be rewritten by executing the bit manipulation instructions.

5.5 Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P4 input/output control register (P4CR). Port P4 is configured as an input if its corresponding P4CR bit is cleared to “0”, and as an output if its corresponding P4CR bit is set to “1”.

During reset, the P4CR is initialized to “0” and port P4 is input mode. The P4 output latches are also initialized to “0”.

When used as an input port, the corresponding bit of P4CR is cleared to “0”.

When used as an output port, the corresponding bit of P4CR is set to “1”. Data can be written into the output latch regardless of P4CR contents, therefore initial output data should be written into the output latch before setting P4CR.

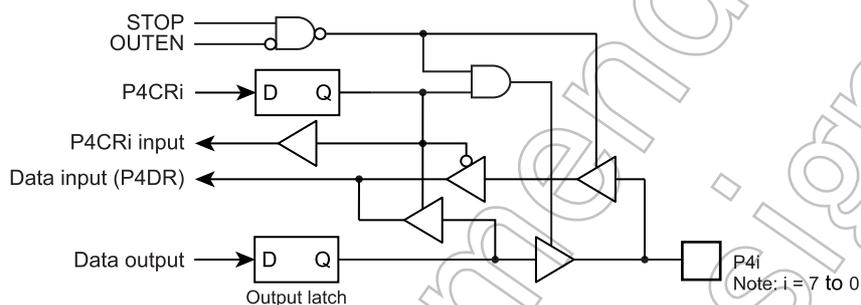


Figure 5-6 Port P4

| | | | | | | | | | |
|------------------------|-----|-----|-----|-----|-----|-----|-----|-----|----------------------------|
| P4DR (0004H) R/W | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: 0000 0000) |
| | P47 | P46 | P45 | P44 | P43 | P42 | P41 | P40 | |

| | | | | | | | | | |
|-----------------|---|---|---|---|---|---|---|---|----------------------------|
| P4CR (000FH) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: 0000 0000) |
| | | | | | | | | | |

| | | | |
|------|--|---------------------------------|-----|
| P4CR | I/O control for port P4 (Specified for each bit) | 0: Input mode 1: Output mode | R/W |
|------|--|---------------------------------|-----|

Note: Ports set to the input mode read the pin states. Ports set to the output mode read the output latch. When input pin and output pin exist in port P4 together, the contents of the output latch which is specified as an input mode may be rewritten by executing the bit manipulation instructions.

6. Time Base Timer (TBT)

The time base timer generates time base for key scanning, dynamic displaying, etc. It also provides a time base timer interrupt (INTTBT).

6.1 Time Base Timer

6.1.1 Configuration

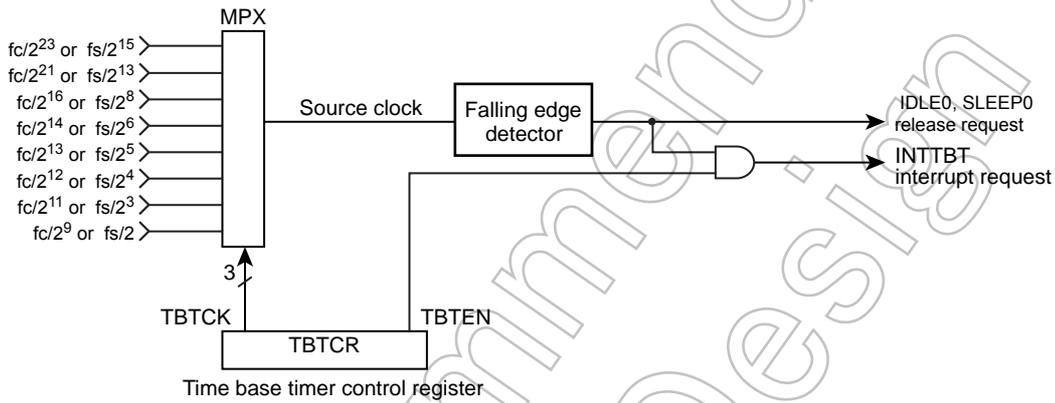


Figure 6-1 Time Base Timer configuration

6.1.2 Control

Time Base Timer is controlled by Time Base Timer control register (TBTCR).

Time Base Timer Control Register

| | | | | | | | | | |
|------------------|---------|---------|---------|-------|--------|---|---|---|----------------------------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| TBTCR (0036H) | (DVOEN) | (DVOCK) | (DV7CK) | TBTEN | TBTKCK | | | | (Initial Value: 0000 0000) |

| TBTEN | Time Base Timer enable / disable | 0: Disable 1: Enable | | | | |
|--------|---|---------------------------|-------------|----------------------------|-------------|-------------|
| | | NORMAL 1/2, IDLE 1/2 Mode | | SLOW 1/2 SLEEP 1/2 Mode | | |
| TBTKCK | Time Base Timer interrupt Frequency select : [Hz] | DV7CK = 0 | DV7CK = 1 | R/W | | |
| | | 000 | $fc/2^{23}$ | | $fs/2^{15}$ | $fs/2^{15}$ |
| | | 001 | $fc/2^{21}$ | | $fs/2^{13}$ | $fs/2^{13}$ |
| | | 010 | $fc/2^{16}$ | | $fs/2^8$ | – |
| | | 011 | $fc/2^{14}$ | | $fs/2^6$ | – |
| | | 100 | $fc/2^{13}$ | | $fs/2^5$ | – |
| | | 101 | $fc/2^{12}$ | | $fs/2^4$ | – |
| | | 110 | $fc/2^{11}$ | | $fs/2^3$ | – |
| 111 | $fc/2^9$ | $fs/2$ | – | | | |

Note 1: fc; High-frequency clock [Hz], fs; Low-frequency clock [Hz], *, Don't care

Note 2: The interrupt frequency (TBTCK) must be selected with the time base timer disabled (TBTEN="0"). (The interrupt frequency must not be changed with the disable from the enable state.) Both frequency selection and enabling can be performed simultaneously.

Example :Set the time base timer frequency to $fc/2^{16}$ [Hz] and enable an INTTBT interrupt.

```
LD      (TBTCR) , 00000010B      ; TBTCK ← 010
LD      (TBTCR) , 00001010B      ; TBTEN ← 1
DI      ; IMF ← 0
SET     (EIRL) . 6
```

Table 6-1 Time Base Timer Interrupt Frequency (Example : $fc = 16.0$ MHz, $fs = 32.768$ kHz)

| TBTCK | Time Base Timer Interrupt Frequency [Hz] | | |
|-------|--|-------------------------|------------------------|
| | NORMAL1/2, IDLE1/2 Mode | NORMAL1/2, IDLE1/2 Mode | SLOW1/2, SLEEP1/2 Mode |
| | DV7CK = 0 | DV7CK = 1 | |
| 000 | 1.91 | 1 | 1 |
| 001 | 7.63 | 4 | 4 |
| 010 | 244.14 | 128 | - |
| 011 | 976.56 | 512 | - |
| 100 | 1953.13 | 1024 | - |
| 101 | 3906.25 | 2048 | - |
| 110 | 7812.5 | 4096 | - |
| 111 | 31250 | 16384 | - |

6.1.3 Function

An INTTBT (Time Base Timer Interrupt) is generated on the first falling edge of source clock (The divider output of the timing generator which is selected by TBTCK.) after time base timer has been enabled.

The divider is not cleared by the program; therefore, only the first interrupt may be generated ahead of the set interrupt period (Figure 6-2).

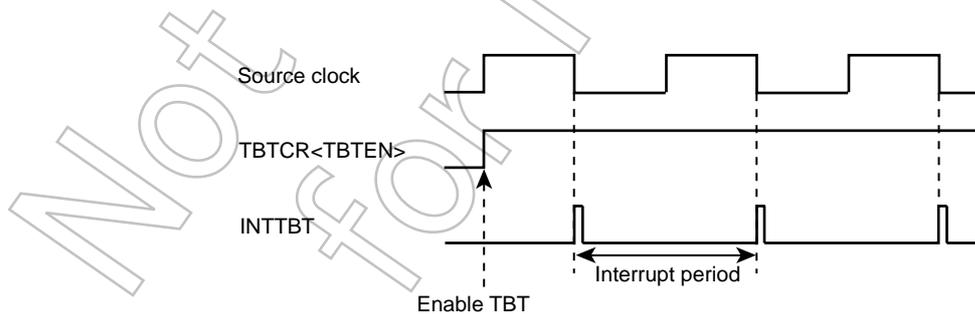


Figure 6-2 Time Base Timer Interrupt

6.2 Divider Output (\overline{DVO})

Approximately 50% duty pulse can be output using the divider output circuit, which is useful for piezoelectric buzzer drive. Divider output is from \overline{DVO} pin.

6.2.1 Configuration

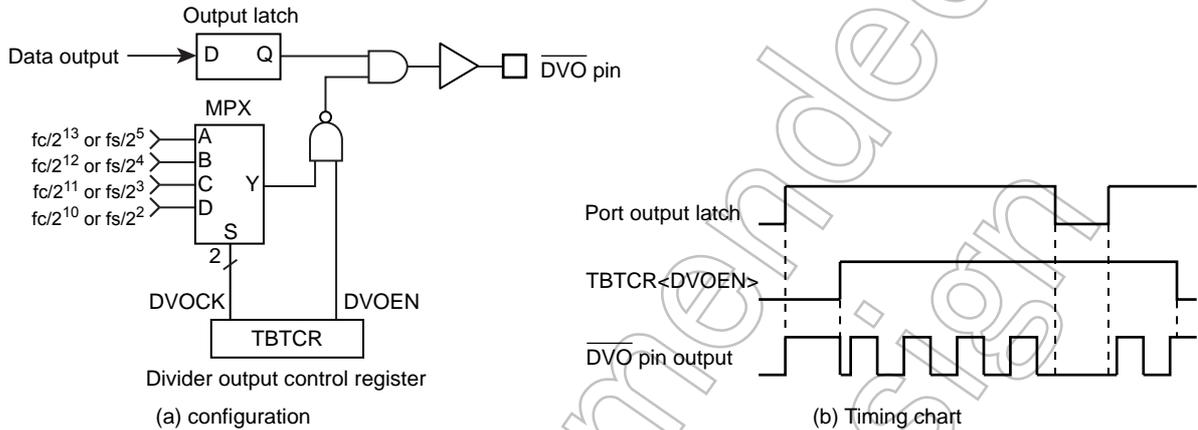
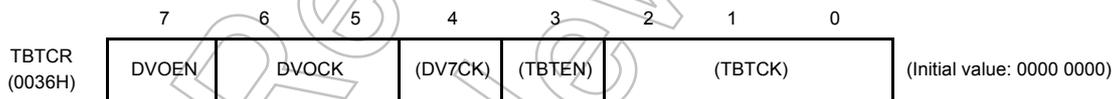


Figure 6-3 Divider Output

6.2.2 Control

The Divider Output is controlled by the Time Base Timer Control Register.

Time Base Timer Control Register



| DVOEN | Divider output enable / disable | 0: Disable 1: Enable | | | R/W | |
|-------|---|---------------------------|-------------|-------------------------|----------|----------|
| | | NORMAL 1/2, IDLE 1/2 Mode | | SLOW 1/2 SLEEP 1/2 Mode | | |
| DVOCK | Divider Output (\overline{DVO}) frequency selection: [Hz] | DV7CK = 0 | DV7CK = 1 | | R/W | |
| | | 00 | $fc/2^{13}$ | $fs/2^5$ | | $fs/2^5$ |
| | | 01 | $fc/2^{12}$ | $fs/2^4$ | | $fs/2^4$ |
| | | 10 | $fc/2^{11}$ | $fs/2^3$ | | $fs/2^3$ |
| | | 11 | $fc/2^{10}$ | $fs/2^2$ | $fs/2^2$ | |

Note: Selection of divider output frequency (DVOCK) must be made while divider output is disabled (DVOEN="0"). Also, in other words, when changing the state of the divider output frequency from enabled (DVOEN="1") to disabled(DVOEN="0"), do not change the setting of the divider output frequency.

Example : 1.95 kHz pulse output (fc = 16.0 MHz)

```
LD      (TBTCR), 00000000B      ; DVOCK ← "00"
LD      (TBTCR), 10000000B      ; DVOEN ← "1"
```

Table 6-2 Divider Output Frequency (Example : fc = 16.0 MHz, fs = 32.768 kHz)

| DVOCK | Divider Output Frequency [Hz] | | |
|-------|-------------------------------|-----------|------------------------|
| | NORMAL1/2, IDLE1/2 Mode | | SLOW1/2, SLEEP1/2 Mode |
| | DV7CK = 0 | DV7CK = 1 | |
| 00 | 1.953 k | 1.024 k | 1.024 k |
| 01 | 3.906 k | 2.048 k | 2.048 k |
| 10 | 7.813 k | 4.096 k | 4.096 k |
| 11 | 15.625 k | 8.192 k | 8.192 k |

Not Recommended for New Design

7. Watchdog Timer (WDT)

The watchdog timer is a fail-safe system to detect rapidly the CPU malfunctions such as endless loops due to spurious noises or the deadlock conditions, and return the CPU to a system recovery routine.

The watchdog timer signal for detecting malfunctions can be programmed only once as “reset request” or “interrupt request”. Upon the reset release, this signal is initialized to “reset request”.

When the watchdog timer is not used to detect malfunctions, it can be used as the timer to provide a periodic interrupt.

Note: Care must be taken in system design since the watchdog timer functions are not be operated completely due to effect of disturbing noise.

7.1 Watchdog Timer Configuration

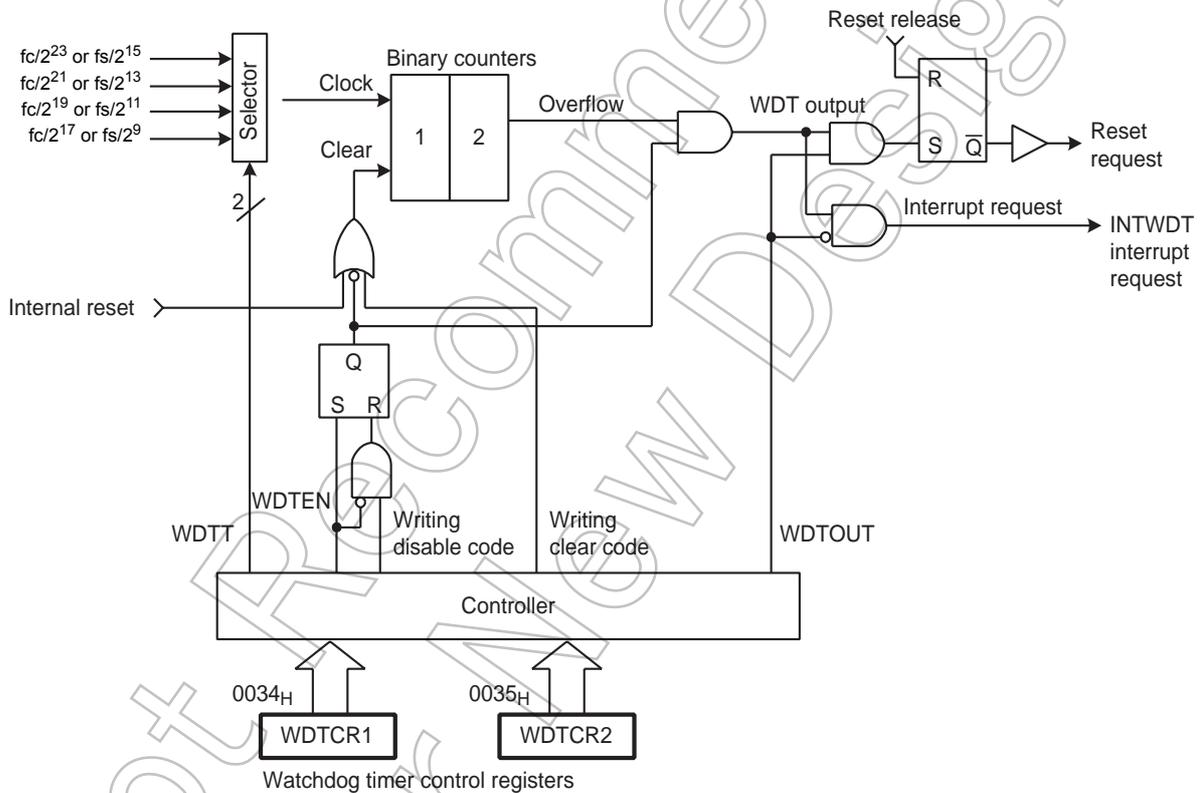


Figure 7-1 Watchdog Timer Configuration

7.2 Watchdog Timer Control

The watchdog timer is controlled by the watchdog timer control registers (WDTCR1 and WDTCR2). The watchdog timer is automatically enabled after the reset release.

7.2.1 Malfunction Detection Methods Using the Watchdog Timer

The CPU malfunction is detected, as shown below.

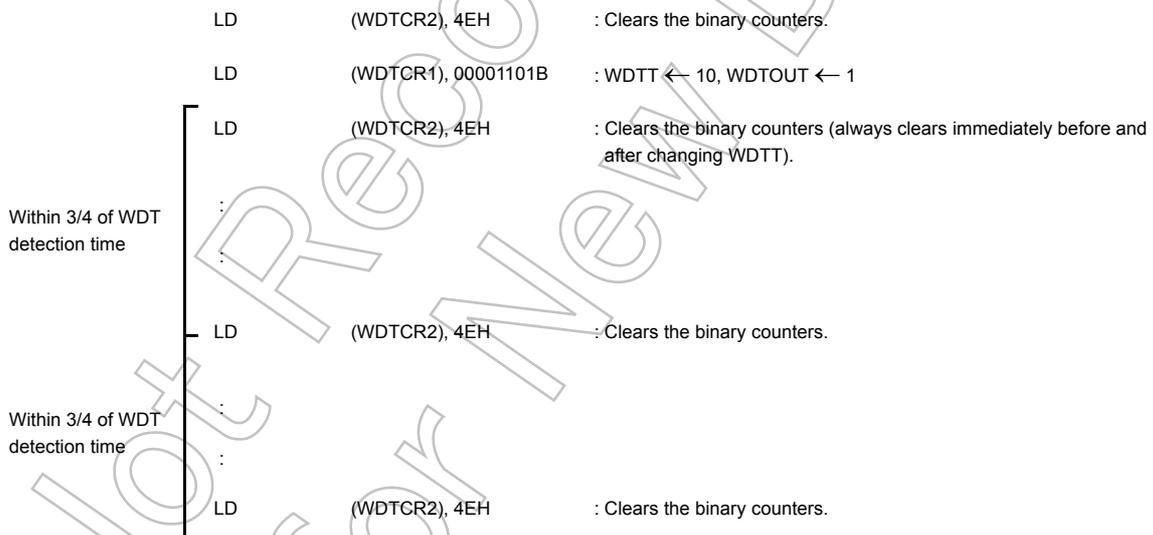
1. Set the detection time, select the output, and clear the binary counter.
2. Clear the binary counter repeatedly within the specified detection time.

If the CPU malfunctions such as endless loops or the deadlock conditions occur for some reason, the watchdog timer output is activated by the binary-counter overflow unless the binary counters are cleared. When WDTCR1<WDTOUT> is set to “1” at this time, the reset request is generated and the RESET pin outputs a low-level signal, then internal hardware is initialized. When WDTCR1<WDTOUT> is set to “0”, a watchdog timer interrupt (INTWDT) is generated.

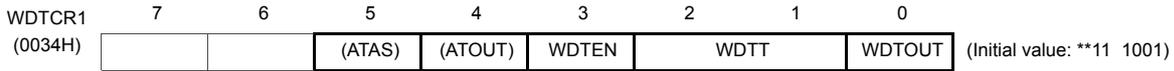
The watchdog timer temporarily stops counting in the STOP mode including the warm-up or IDLE/SLEEP mode, and automatically restarts (continues counting) when the STOP/IDLE/SLEEP mode is inactivated.

Note: The watchdog timer consists of an internal divider and a two-stage binary counter. When the clear code 4EH is written, only the binary counter is cleared, but not the internal divider. The minimum binary-counter overflow time, that depends on the timing at which the clear code (4EH) is written to the WDTCR2 register, may be 3/4 of the time set in WDTCR1<WDTT>. Therefore, write the clear code using a cycle shorter than 3/4 of the time set to WDTCR1<WDTT>.

Example :Setting the watchdog timer detection time to $2^{21}/f_c$ [s], and resetting the CPU malfunction detection



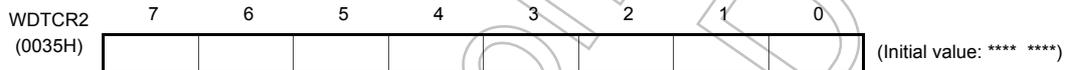
Watchdog Timer Control Register 1



| | | | | | | | |
|--------|-----------------------------------|---|-------------|--------------|------------|-------------|-------------|
| WDTEN | Watchdog timer enable/disable | 0: Disable (Writing the disable code to WDTCR2 is required.) 1: Enable | Write only | | | | |
| WDTT | Watchdog timer detection time [s] | NORMAL1/2 mode | | SLOW1/2 mode | Write only | | |
| | | | DV7CK = 0 | | | DV7CK = 1 | |
| | | 00 | $2^{25}/fc$ | | | $2^{17}/fs$ | $2^{17}/fs$ |
| | | 01 | $2^{23}/fc$ | | | $2^{15}/fs$ | $2^{15}/fs$ |
| | | 10 | $2^{21}/fc$ | | | $2^{13}/fs$ | $2^{13}/fs$ |
| | 11 | $2^{19}/fc$ | $2^{11}/fs$ | $2^{11}/fs$ | | | |
| WDTOUT | Watchdog timer output select | 0: Interrupt request 1: Reset request | Write only | | | | |

- Note 1: After clearing WDTOUT to "0", the program cannot set it to "1".
- Note 2: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care
- Note 3: WDTCR1 is a write-only register and must not be used with any of read-modify-write instructions. If WDTCR1 is read, a don't care is read.
- Note 4: To activate the STOP mode, disable the watchdog timer or clear the counter immediately before entering the STOP mode. After clearing the counter, clear the counter again immediately after the STOP mode is inactivated.
- Note 5: To clear WDTEN, set the register in accordance with the procedures shown in "7.2.3 Watchdog Timer Disable".

Watchdog Timer Control Register 2



| | | | |
|--------|-----------------------------------|---|------------|
| WDTCR2 | Write Watchdog timer control code | 4EH: Clear the watchdog timer binary counter (Clear code) B1H: Disable the watchdog timer (Disable code) D2H: Enable assigning address trap area Others: Invalid | Write only |
|--------|-----------------------------------|---|------------|

- Note 1: The disable code is valid only when WDTCR1<WDTEN> = 0.
- Note 2: *: Don't care
- Note 3: The binary counter of the watchdog timer must not be cleared by the interrupt task.
- Note 4: Write the clear code 4EH using a cycle shorter than 3/4 of the time set in WDTCR1<WDTT>.

7.2.2 Watchdog Timer Enable

Setting WDTCR1<WDTEN> to "1" enables the watchdog timer. Since WDTCR1<WDTEN> is initialized to "1" during reset, the watchdog timer is enabled automatically after the reset release.

7.2.3 Watchdog Timer Disable

To disable the watchdog timer, set the register in accordance with the following procedures. Setting the register in other procedures causes a malfunction of the microcontroller.

1. Set the interrupt master flag (IMF) to “0”.
2. Set WDTCR2 to the clear code (4EH).
3. Set WDTCR1<WDTEN> to “0”.
4. Set WDTCR2 to the disable code (B1H).

Note: While the watchdog timer is disabled, the binary counters of the watchdog timer are cleared.

Example :Disabling the watchdog timer

```
DI          : IMF ← 0
LD          (WDTCR2), 04EH      : Clears the binary counter
LDW        (WDTCR1), 0B101H    : WDTEN ← 0, WDTCR2 ← Disable code
```

Table 7-1 Watchdog Timer Detection Time (Example: $f_c = 16.0$ MHz, $f_s = 32.768$ kHz)

| WDTT | Watchdog Timer Detection Time[s] | | |
|------|----------------------------------|-----------|-----------|
| | NORMAL1/2 mode | | SLOW mode |
| | DV7CK = 0 | DV7CK = 1 | |
| 00 | 2.097 | 4 | 4 |
| 01 | 524.288 m | 1 | 1 |
| 10 | 131.072 m | 250 m | 250 m |
| 11 | 32.768 m | 62.5 m | 62.5 m |

7.2.4 Watchdog Timer Interrupt (INTWDT)

When WDTCR1<WDTOUT> is cleared to “0”, a watchdog timer interrupt request (INTWDT) is generated by the binary-counter overflow.

A watchdog timer interrupt is the non-maskable interrupt which can be accepted regardless of the interrupt master flag (IMF).

When a watchdog timer interrupt is generated while the other interrupt including a watchdog timer interrupt is already accepted, the new watchdog timer interrupt is processed immediately and the previous interrupt is held pending. Therefore, if watchdog timer interrupts are generated continuously without execution of the RETN instruction, too many levels of nesting may cause a malfunction of the microcontroller.

To generate a watchdog timer interrupt, set the stack pointer before setting WDTCR1<WDTOUT>.

Example :Setting watchdog timer interrupt

```
LD          SP, 043FH          : Sets the stack pointer
LD          (WDTCR1), 00001000B : WDTOUT ← 0
```

7.2.5 Watchdog Timer Reset

When a binary-counter overflow occurs while $WDTCR1<WDTOUT>$ is set to "1", a watchdog timer reset request is generated. When a watchdog timer reset request is generated, the \overline{RESET} pin outputs a low-level signal and the internal hardware is reset. The reset time is maximum $24/fc$ [s] ($1.5 \mu s$ @ $fc = 16.0$ MHz).

Note: When a watchdog timer reset is generated in the SLOW1 mode, the reset time is maximum $24/fc$ (high-frequency clock) since the high-frequency clock oscillator is restarted. However, when crystals have inaccuracies upon start of the high-frequency clock oscillator, the reset time should be considered as an approximate value because it has slight errors.

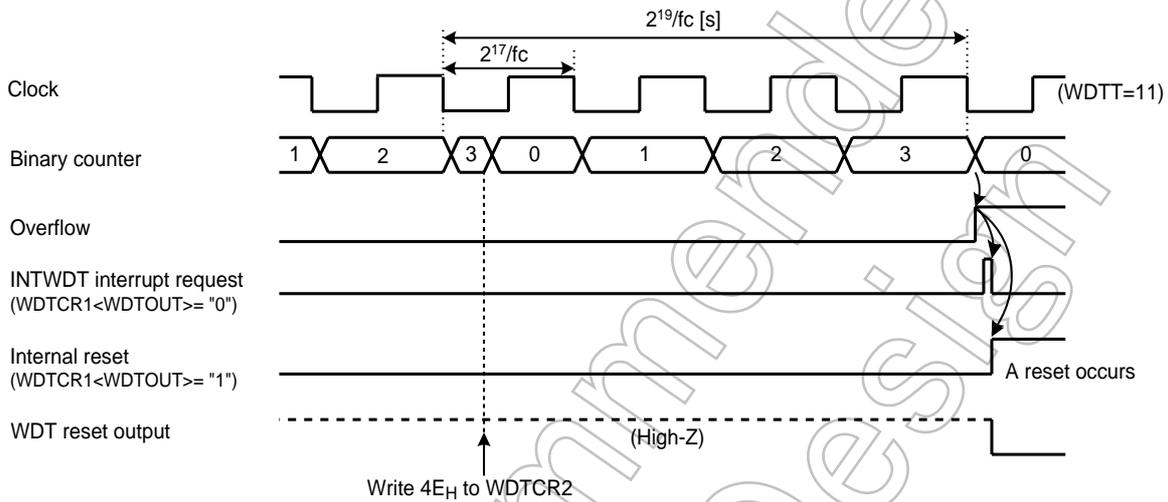


Figure 7-2 Watchdog Timer Interrupt/Reset

Not Recommended for New Design

7.3 Address Trap

The Watchdog Timer Control Register 1 and 2 share the addresses with the control registers to generate address traps.

Watchdog Timer Control Register 1

| | | | | | | | | | |
|-------------------|---|---|------|-------|---------|--------|----------|---|----------------------------|
| WDTCR1 (0034H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: **11 1001) |
| | | | ATAS | ATOUT | (WDTEN) | (WDTT) | (WDTOUT) | | |

| | | | |
|-------|---|--|------------|
| ATAS | Select address trap generation in the internal RAM area | 0: Generate no address trap 1: Generate address traps (After setting ATAS to "1", writing the control code D2H to WDTCR2 is required) | Write only |
| ATOUT | Select operation at address trap | 0: Interrupt request 1: Reset request | |

Watchdog Timer Control Register 2

| | | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|---|---------------------------|
| WDTCR2 (0035H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: **** ***) |
| | | | | | | | | | |

| | | | |
|--------|--|--|------------|
| WDTCR2 | Write Watchdog timer control code and address trap area control code | D2H: Enable address trap area selection (ATRAP control code) 4EH: Clear the watchdog timer binary counter (WDT clear code) B1H: Disable the watchdog timer (WDT disable code) Others: Invalid | Write only |
|--------|--|--|------------|

7.3.1 Selection of Address Trap in Internal RAM (ATAS)

WDTCR1<ATAS> specifies whether or not to generate address traps in the internal RAM area. To execute an instruction in the internal RAM area, clear WDTCR1<ATAS> to "0". To enable the WDTCR1<ATAS> setting, set WDTCR1<ATAS> and then write D2H to WDTCR2.

Executing an instruction in the SFR area generates an address trap unconditionally regardless of the setting in WDTCR1<ATAS>.

7.3.2 Selection of Operation at Address Trap (ATOUT)

When an address trap is generated, either the interrupt request or the reset request can be selected by WDTCR1<ATOUT>.

7.3.3 Address Trap Interrupt (INTATRAP)

While WDTCR1<ATOUT> is "0", if the CPU should start looping for some cause such as noise and an attempt be made to fetch an instruction from the on-chip RAM (while WDTCR1<ATAS> is "1") or the SFR area, address trap interrupt (INTATRAP) will be generated.

An address trap interrupt is a non-maskable interrupt which can be accepted regardless of the interrupt master flag (IMF).

When an address trap interrupt is generated while the other interrupt including an address trap interrupt is already accepted, the new address trap is processed immediately and the previous interrupt is held pending. Therefore, if address trap interrupts are generated continuously without execution of the RETN instruction, too many levels of nesting may cause a malfunction of the microcontroller.

To generate address trap interrupts, set the stack pointer beforehand.

7.3.4 Address Trap Reset

While WDTCR1<ATOUT> is “1”, if the CPU should start looping for some cause such as noise and an attempt be made to fetch an instruction from the on-chip RAM (while WDTCR1<ATAS> is “1”) or the SFR area, address trap reset will be generated.

When an address trap reset request is generated, the $\overline{\text{RESET}}$ pin outputs a low-level signal and the internal hardware is reset. The reset time is maximum $24/f_c$ [s] ($1.5 \mu\text{s}$ @ $f_c = 16.0 \text{ MHz}$).

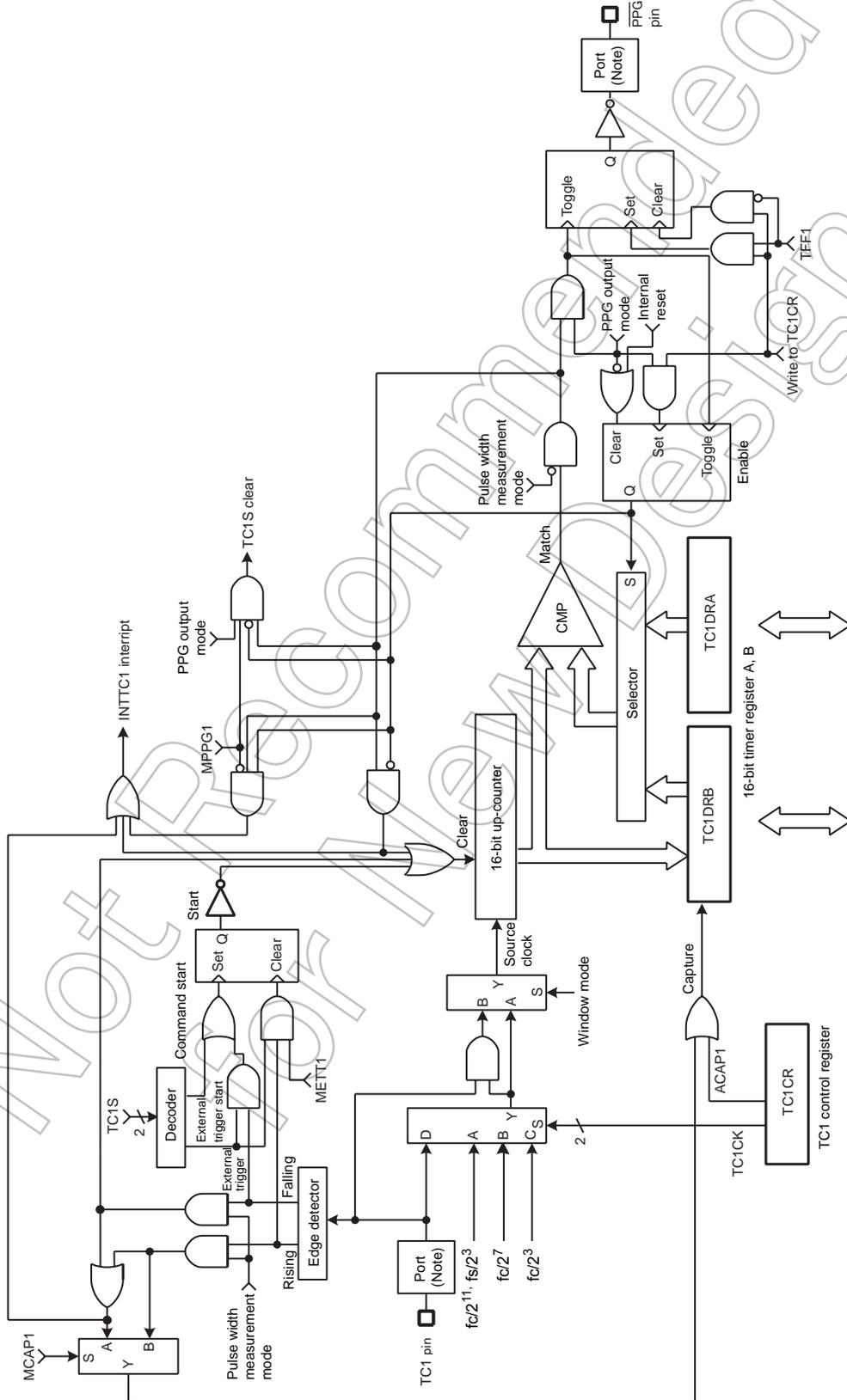
Note: When an address trap reset is generated in the SLOW1 mode, the reset time is maximum $24/f_c$ (high-frequency clock) since the high-frequency clock oscillator is restarted. However, when crystals have inaccuracies upon start of the high-frequency clock oscillator, the reset time should be considered as an approximate value because it has slight errors.

Not Recommended
for New Design

Not Recommended
for New Design

8. 16-Bit TimerCounter 1 (TC1)

8.1 Configuration



Note: Function I/O may not operate depending on I/O port setting. For more details, see the chapter "I/O Port".

Figure 8-1 TimerCounter 1 (TC1)

8.2 TimerCounter Control

The TimerCounter 1 is controlled by the TimerCounter 1 control register (TC1CR) and two 16-bit timer registers (TC1DRA and TC1DRB).

Timer Register

| | | | | | | | | | | | | | | | | |
|--------------------------|---|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TC1DRA (0011H, 0010H) | TC1DRAH (0011H) (Initial value: 1111 1111 1111 1111) | | | | | | | | TC1DRAL (0010H) Read/Write | | | | | | | |
| TC1DRB (0013H, 0012H) | TC1DRBH (0013H) (Initial value: 1111 1111 1111 1111) | | | | | | | | TC1DRBL (0012H) Read/Write (Write enabled only in the PPG output mode) | | | | | | | |

TimerCounter 1 Control Register

| | | | | | | | | |
|------------------|------|----------------------------------|------|-------|------|--|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TC1CR (0014H) | TFF1 | ACAP1 MCAP1 METT1 MPPG1 | TC1S | TC1CK | TC1M | Read/Write (Initial value: 0000 0000) | | |

| | | | | | | | | | | |
|-------|--------------------------------------|---|--------------------------------|----------|------------|-------|----------|------------------|-------|-----|
| TFF1 | Timer F/F1 control | 0: Clear | 1: Set | | | | | R/W | | |
| ACAP1 | Auto capture control | 0:Auto-capture disable | 1:Auto-capture enable | | | | | R/W | | |
| MCAP1 | Pulse width measurement mode control | 0:Double edge capture | 1:Single edge capture | | | | | | | |
| METT1 | External trigger timer mode control | 0:Trigger start | 1:Trigger start and stop | | | | | | | |
| MPPG1 | PPG output control | 0:Continuous pulse generation | 1:One-shot | | | | | | | |
| TC1S | TC1 start control | | | Timer | Extrig-ger | Event | Win-dow | | Pulse | PPG |
| | | 00: Stop and counter clear | | 0 | 0 | 0 | 0 | 0 | 0 | |
| | | 01: Command start | | 0 | - | - | - | - | 0 | |
| | | 10: Rising edge start (Ex-trigger/Pulse/PPG) Rising edge count (Event) Positive logic count (Window) | | - | 0 | 0 | 0 | 0 | 0 | |
| | | 11: Falling edge start (Ex-trigger/Pulse/PPG) Falling edge count (Event) Negative logic count (Window) | | - | 0 | 0 | 0 | 0 | 0 | |
| TC1CK | TC1 source clock select [Hz] | NORMAL1/2, IDLE1/2 mode | | | | | Divider | SLOW, SLEEP mode | R/W | |
| | | DV7CK = 0 | | | DV7CK = 1 | | | | | |
| | | 00 | $fc/2^{11}$ | $fs/2^3$ | | DV9 | $fs/2^3$ | | | |
| | | 01 | $fc/2^7$ | $fc/2^7$ | | DV5 | - | | | |
| | | 10 | $fc/2^3$ | $fc/2^3$ | | DV1 | - | | | |
| | | 11 | External clock (TC1 pin input) | | | | | | | |
| TC1M | TC1 operating mode select | 00: Timer/external trigger timer/event counter mode 01: Window mode 10: Pulse width measurement mode 11: PPG (Programmable pulse generate) output mode | | | | | | R/W | | |

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz]

Note 2: The timer register consists of two shift registers. A value set in the timer register becomes valid at the rising edge of the first source clock pulse that occurs after the upper byte (TC1DRAH and TC1DRBH) is written. Therefore, write the lower byte and the upper byte in this order (it is recommended to write the register with a 16-bit access instruction). Writing only the lower byte (TC1DRAL and TC1DRBL) does not enable the setting of the timer register.

Note 3: To set the mode, source clock, PPG output control and timer F/F control, write to TC1CR during TC1S=00. Set the timer F/ F1 control until the first timer start after setting the PPG mode.

Note 4: Auto-capture can be used only in the timer, event counter, and window modes.

Note 5: To set the timer registers, the following relationship must be satisfied.

TC1DRA > TC1DRB > 1 (PPG output mode), TC1DRA > 1 (other modes)

Note 6: Set TFF1 to "0" in the mode except PPG output mode.

Note 7: Set TC1DRB after setting TC1M to the PPG output mode.

Note 8: When the STOP mode is entered, the start control (TC1S) is cleared to "00" automatically, and the timer stops. After the STOP mode is exited, set the TC1S to use the timer counter again.

Note 9: Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition.

Note 10: Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

Not Recommended
for New Design

8.3 Function

TimerCounter 1 has six types of operating modes: timer, external trigger timer, event counter, window, pulse width measurement, programmable pulse generator output modes.

8.3.1 Timer mode

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the timer register 1A (TC1DRA) value is detected, an INTTC1 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting. Setting TC1CR<ACAP1> to "1" captures the up-counter value into the timer register 1B (TC1DRB) with the auto-capture function. Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition. Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

Table 8-1 Internal Source Clock for TimerCounter 1 (Example: $f_c = 16$ MHz, $f_s = 32.768$ kHz)

| TC1CK | NORMAL1/2, IDLE1/2 mode | | | | SLOW, SLEEP mode | |
|-------|-------------------------|--------------------------|-----------------|--------------------------|------------------|--------------------------|
| | DV7CK = 0 | | DV7CK = 1 | | Resolution [μs] | Maximum Time Setting [s] |
| | Resolution [μs] | Maximum Time Setting [s] | Resolution [μs] | Maximum Time Setting [s] | | |
| 00 | 128 | 8.39 | 244.14 | 16.0 | 244.14 | 16.0 |
| 01 | 8.0 | 0.524 | 8.0 | 0.524 | - | - |
| 10 | 0.5 | 32.77 m | 0.5 | 32.77 m | - | - |

Example 1 :Setting the timer mode with source clock $f_c/2^{11}$ [Hz] and generating an interrupt 1 second later ($f_c = 16$ MHz, TBTCR<DV7CK> = "0")

```
LDW      (TC1DRA), 1E84H      ; Sets the timer register ( $1\text{ s} \div 2^{11}/f_c = 1\text{E84H}$ )
DI       ; IMF="0"
SET      (EIRL), 7           ; Enables INTTC1
EI       ; IMF="1"
LD       (TC1CR), 00000000B   ; Selects the source clock and mode
```

```
LD       (TC1CR), 00010000B   ; Starts TC1
```

Example 2 :Auto-capture

```
LD       (TC1CR), 01010000B   ; ACAP1 ← 1
```

```
:
```

```
LD       WA, (TC1DRB)        ; Reads the capture value
```

Note: Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

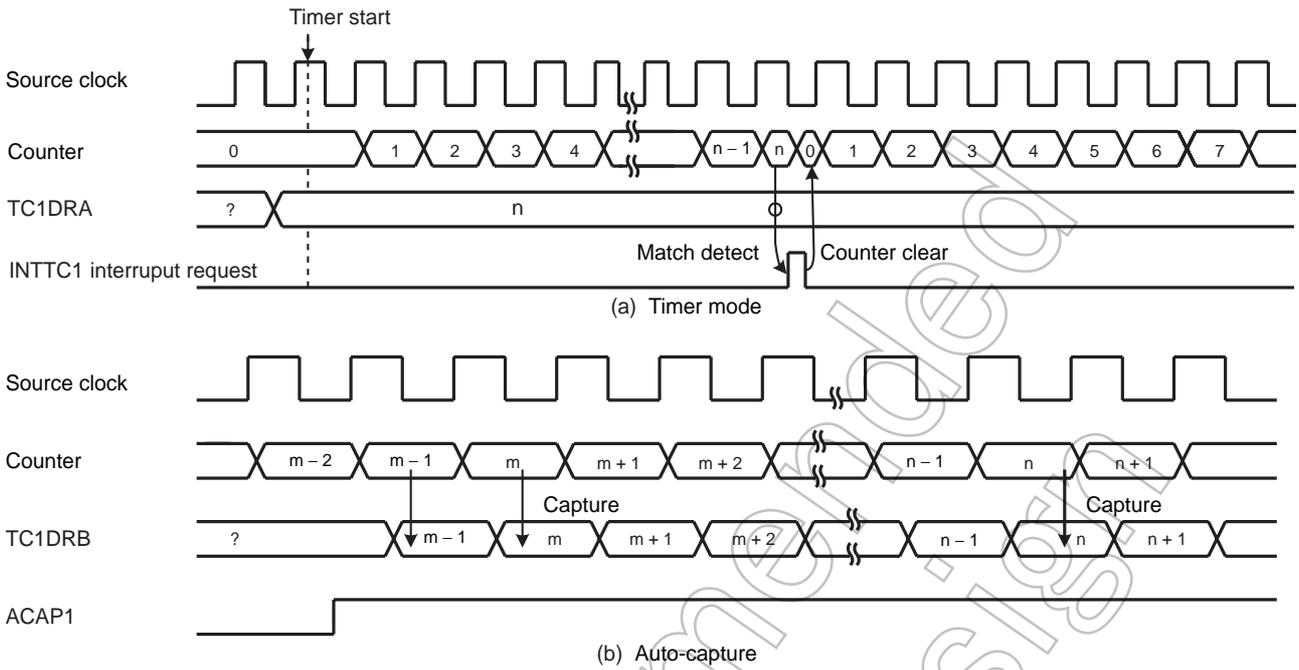


Figure 8-2 Timer Mode Timing Chart

Not Recommended for New Design

8.3.2 External Trigger Timer Mode

In the external trigger timer mode, the up-counter starts counting by the input pulse triggering of the TC1 pin, and counts up at the edge of the internal clock. For the trigger edge used to start counting, either the rising or falling edge is defined in TC1CR<TCIS>.

- When TC1CR<METT1> is set to “1” (trigger start and stop)

When a match between the up-counter and the TC1DRA value is detected after the timer starts, the up-counter is cleared and halted and an INTTC1 interrupt request is generated.

If the edge opposite to trigger edge is detected before detecting a match between the up-counter and the TC1DRA, the up-counter is cleared and halted without generating an interrupt request. Therefore, this mode can be used to detect exceeding the specified pulse by interrupt.

After being halted, the up-counter restarts counting when the trigger edge is detected.

- When TC1CR<METT1> is set to “0” (trigger start)

When a match between the up-counter and the TC1DRA value is detected after the timer starts, the up-counter is cleared and halted and an INTTC1 interrupt request is generated.

The edge opposite to the trigger edge has no effect in count up. The trigger edge for the next counting is ignored if detecting it before detecting a match between the up-counter and the TC1DRA.

Since the TC1 pin input has the noise rejection, pulses of $4/f_c$ [s] or less are rejected as noise. A pulse width of $12/f_c$ [s] or more is required to ensure edge detection. The rejection circuit is turned off in the SLOW1/2 or SLEEP1/2 mode, but a pulse width of one machine cycle or more is required.

Example 1 :Generating an interrupt 1 ms after the rising edge of the input pulse to the TC1 pin
($f_c = 16$ MHz)

```
LDW      (TC1DRA), 007DH      ; 1ms ÷ 27/fc = 7DH
DI       ; IMF= "0"
SET      (EIRL). 7           ; Enables INTTC1 interrupt
EI       ; IMF= "1"
LD       (TC1CR), 00000100B   ; Selects the source clock and mode
LD       (TC1CR), 00100100B   ; Starts TC1 external trigger, METT1 = 0
```

Example 2 :Generating an interrupt when the low-level pulse with 4 ms or more width is input to the TC1 pin
($f_c = 16$ MHz)

```
LDW      (TC1DRA), 01F4H      ; 4 ms ÷ 27/fc = 1F4H
DI       ; IMF= "0"
SET      (EIRL). 7           ; Enables INTTC1 interrupt
EI       ; IMF= "1"
LD       (TC1CR), 00000100B   ; Selects the source clock and mode
LD       (TC1CR), 01110100B   ; Starts TC1 external trigger, METT1 = 1
```

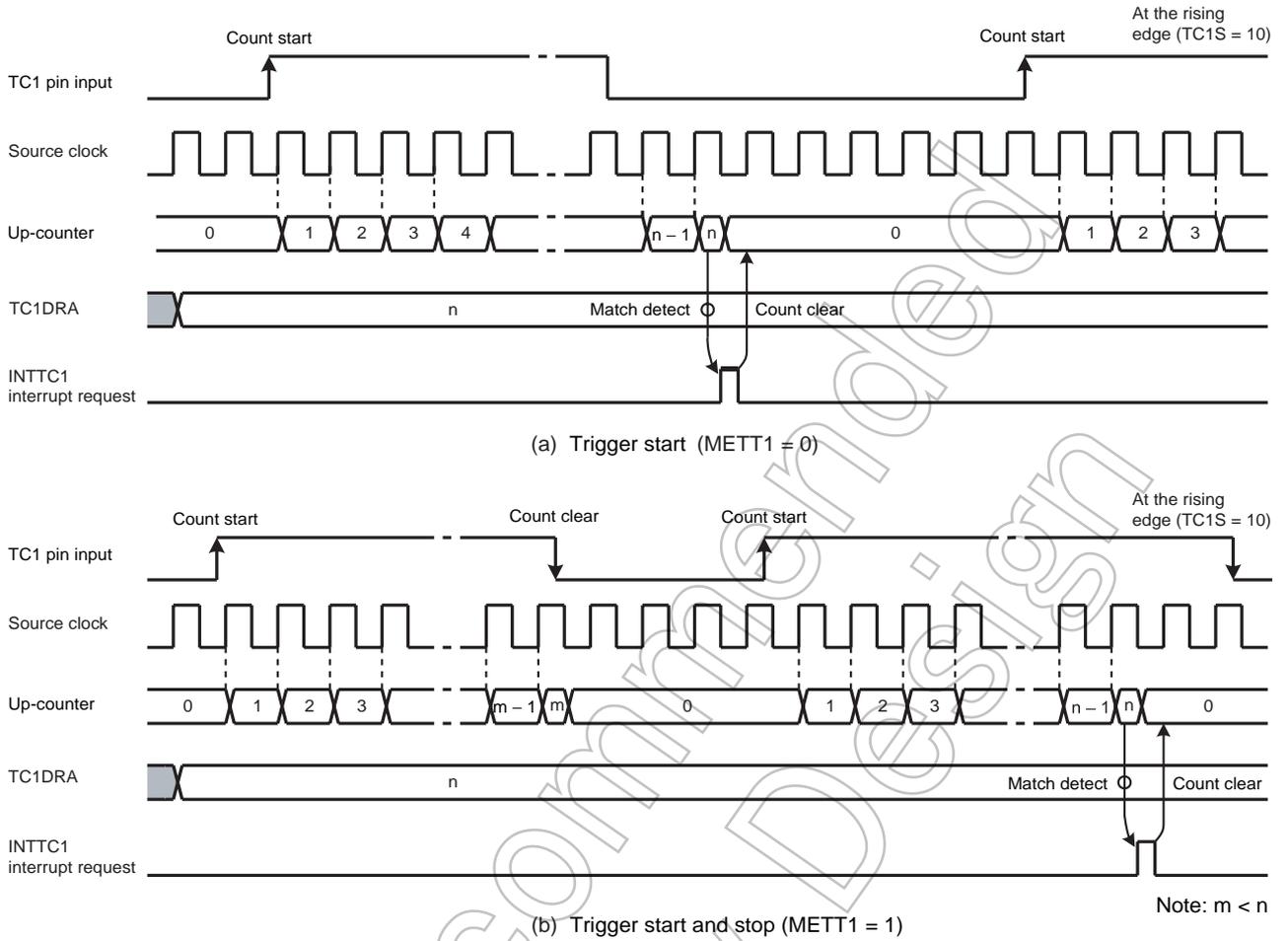


Figure 8-3 External Trigger Timer Mode Timing Chart

8.3.3 Event Counter Mode

In the event counter mode, the up-counter counts up at the edge of the input pulse to the TC1 pin. Either the rising or falling edge of the input pulse is selected as the count up edge in TC1CR<TC1S>.

When a match between the up-counter and the TC1DRA value is detected, an INTTC1 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at each edge of the input pulse to the TC1 pin. Since a match between the up-counter and the value set to TC1DRA is detected at the edge opposite to the selected edge, an INTTC1 interrupt request is generated after a match of the value at the edge opposite to the selected edge.

Two or more machine cycles are required for the low-or high-level pulse input to the TC1 pin.

Setting TC1CR<ACAP1> to "1" captures the up-counter value into TC1DRB with the auto capture function. Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition. Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

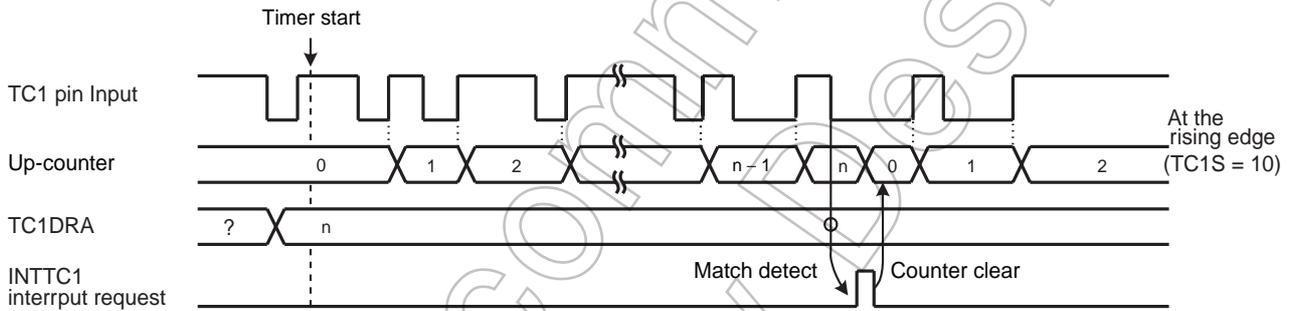


Figure 8-4 Event Counter Mode Timing Chart

Table 8-2 Input Pulse Width to TC1 Pin

| | Minimum Pulse Width [s] | |
|------------|-------------------------|------------------------|
| | NORMAL1/2, IDLE1/2 Mode | SLOW1/2, SLEEP1/2 Mode |
| High-going | $2^3/f_c$ | $2^3/f_s$ |
| Low-going | $2^3/f_c$ | $2^3/f_s$ |

8.3.4 Window Mode

In the window mode, the up-counter counts up at the rising edge of the pulse that is logical ANDed product of the input pulse to the TC1 pin (window pulse) and the internal source clock. Either the positive logic (count up during high-going pulse) or negative logic (count up during low-going pulse) can be selected.

When a match between the up-counter and the TC1DRA value is detected, an INTTC1 interrupt is generated and the up-counter is cleared.

Define the window pulse to the frequency which is sufficiently lower than the internal source clock programmed with TC1CR<TC1CK>.

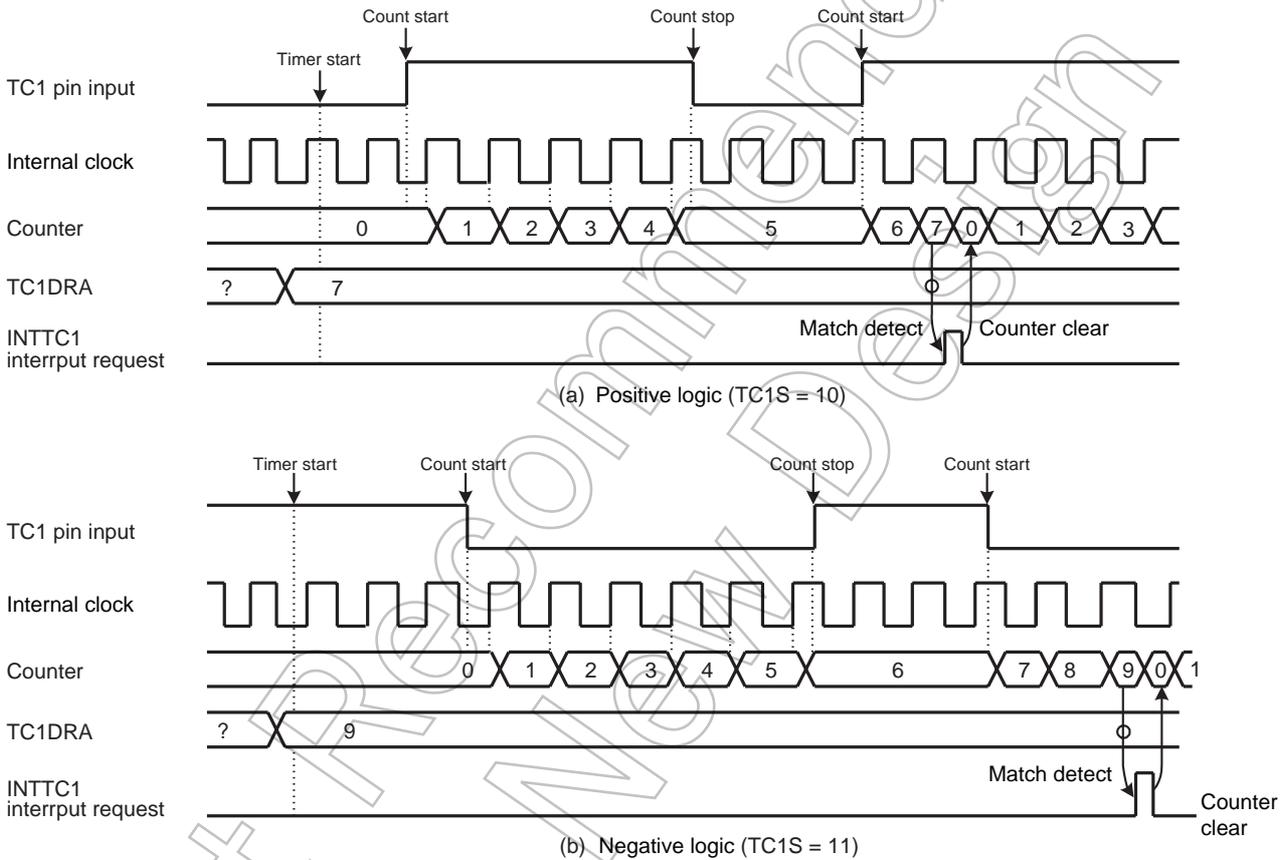


Figure 8-5 Window Mode Timing Chart

8.3.5 Pulse Width Measurement Mode

In the pulse width measurement mode, the up-counter starts counting by the input pulse triggering of the TC1 pin, and counts up at the edge of the internal clock. Either the rising or falling edge of the internal clock is selected as the trigger edge in TC1CR<TC1S>. Either the single- or double-edge capture is selected as the trigger edge in TC1CR<MCAP1>.

- When TC1CR<MCAP1> is set to “1” (single-edge capture)

Either high- or low-level input pulse width can be measured. To measure the high-level input pulse width, set the rising edge to TC1CR<TC1S>. To measure the low-level input pulse width, set the falling edge to TC1CR<TC1S>.

When detecting the edge opposite to the trigger edge used to start counting after the timer starts, the up-counter captures the up-counter value into TC1DRB and generates an INTTC1 interrupt request. The up-counter is cleared at this time, and then restarts counting when detecting the trigger edge used to start counting.

- When TC1CR<MCAP1> is set to “0” (double-edge capture)

The cycle starting with either the high- or low-going input pulse can be measured. To measure the cycle starting with the high-going pulse, set the rising edge to TC1CR<TC1S>. To measure the cycle starting with the low-going pulse, set the falling edge to TC1CR<TC1S>.

When detecting the edge opposite to the trigger edge used to start counting after the timer starts, the up-counter captures the up-counter value into TC1DRB and generates an INTTC1 interrupt request. The up-counter continues counting up, and captures the up-counter value into TC1DRB and generates an INTTC1 interrupt request when detecting the trigger edge used to start counting. The up-counter is cleared at this time, and then continues counting.

Note 1: The captured value must be read from TC1DRB until the next trigger edge is detected. If not read, the captured value becomes a don't care. It is recommended to use a 16-bit access instruction to read the captured value from TC1DRB.

Note 2: For the single-edge capture, the counter after capturing the value stops at “1” until detecting the next edge. Therefore, the second captured value is “1” larger than the captured value immediately after counting starts.

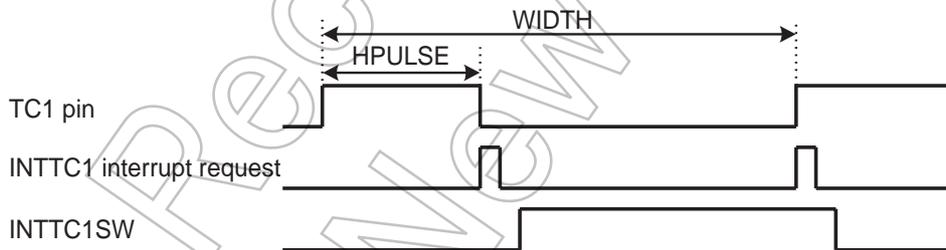
Note 3: The first captured value after the timer starts may be read incorrectly, therefore, ignore the first captured value.

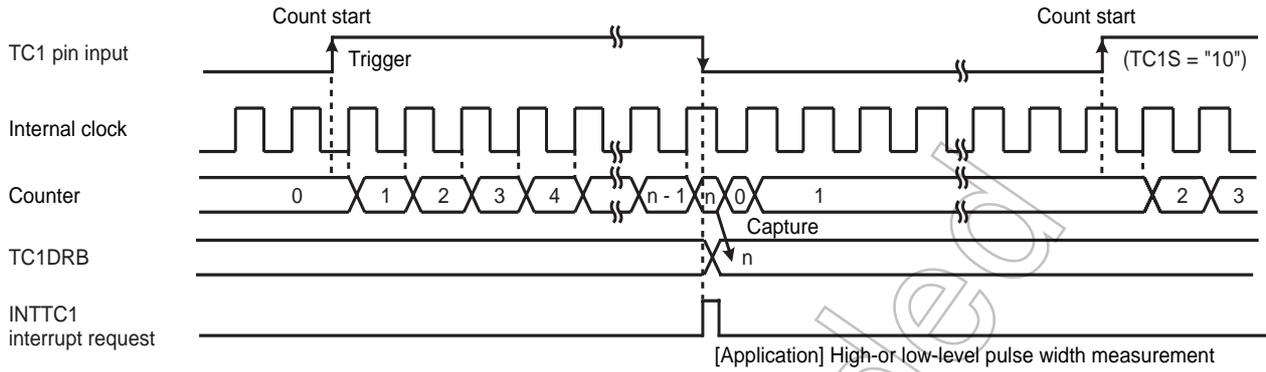
Example :Duty measurement (resolution $f_c/2^7$ [Hz])

```

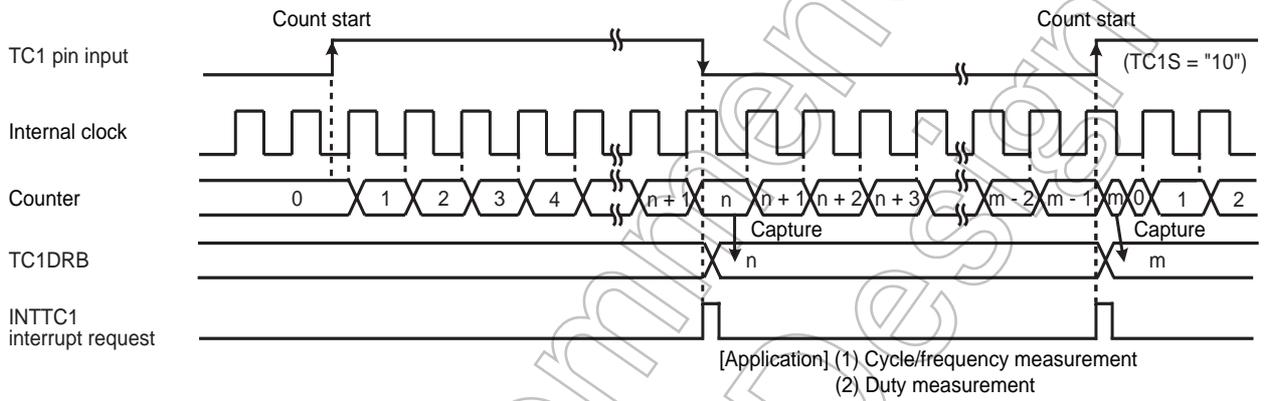
CLR      (INTTC1SW). 0      ; INTTC1 service switch initial setting
                          Address set to convert INTTC1SW at each INTTC1

LD      (TC1CR), 00000110B  ; Sets the TC1 mode and source clock
DI      ; IMF= "0"
SET     (EIRL). 7          ; Enables INTTC1
EI      ; IMF= "1"
LD      (TC1CR), 00100110B  ; Starts TC1 with an external trigger at MCAP1 = 0
:
PINTTC1: CPL      (INTTC1SW). 0      ; INTTC1 interrupt, inverts and tests INTTC1 service switch
          JRS     F, SINTTC1
          LD      A, (TC1DRBL)      ; Reads TC1DRB (High-level pulse width)
          LD      W,(TC1DRBH)
          LD      (HPULSE), WA      ; Stores high-level pulse width in RAM
          RETI
SINTTC1: LD      A, (TC1DRBL)      ; Reads TC1DRB (Cycle)
          LD      W,(TC1DRBH)
          LD      (WIDTH), WA      ; Stores cycle in RAM
          :
          RETI      ; Duty calculation
          :
VINTTC1: DW      PINTTC1          ; INTTC1 Interrupt vector
    
```





(a) Single-edge capture (MCAP1 = "1")



(b) Double-edge capture (MCAP1 = "0")

Figure 8-6 Pulse Width Measurement Mode

Not Recommended for New Designs

8.3.6 Programmable Pulse Generate (PPG) Output Mode

In the programmable pulse generation (PPG) mode, an arbitrary duty pulse is generated by counting performed in the internal clock. To start the timer, TC1CR<TC1S> specifies either the edge of the input pulse to the TC1 pin or the command start. TC1CR<MPPG1> specifies whether a duty pulse is produced continuously or not (one-shot pulse).

- When TC1CR<MPPG1> is set to “0” (Continuous pulse generation)

When a match between the up-counter and the TC1DRB value is detected after the timer starts, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC1 interrupt request is generated. The up-counter continues counting. When a match between the up-counter and the TC1DRA value is detected, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC1 interrupt request is generated. The up-counter is cleared at this time, and then continues counting and pulse generation.

When TC1S is cleared to “00” during PPG output, the $\overline{\text{PPG}}$ pin retains the level immediately before the counter stops.

- When TC1CR<MPPG1> is set to “1” (One-shot pulse generation)

When a match between the up-counter and the TC1DRB value is detected after the timer starts, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC1 interrupt request is generated. The up-counter continues counting. When a match between the up-counter and the TC1DRA value is detected, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC1 interrupt request is generated. TC1CR<TC1S> is cleared to “00” automatically at this time, and the timer stops. The pulse generated by PPG retains the same level as that when the timer stops.

Since the output level of the $\overline{\text{PPG}}$ pin can be set with TC1CR<TFF1> when the timer starts, a positive or negative pulse can be generated. Since the inverted level of the timer F/F1 output level is output to the $\overline{\text{PPG}}$ pin, specify TC1CR<TFF1> to “0” to set the high level to the $\overline{\text{PPG}}$ pin, and “1” to set the low level to the $\overline{\text{PPG}}$ pin. Upon reset, the timer F/F1 is initialized to “0”.

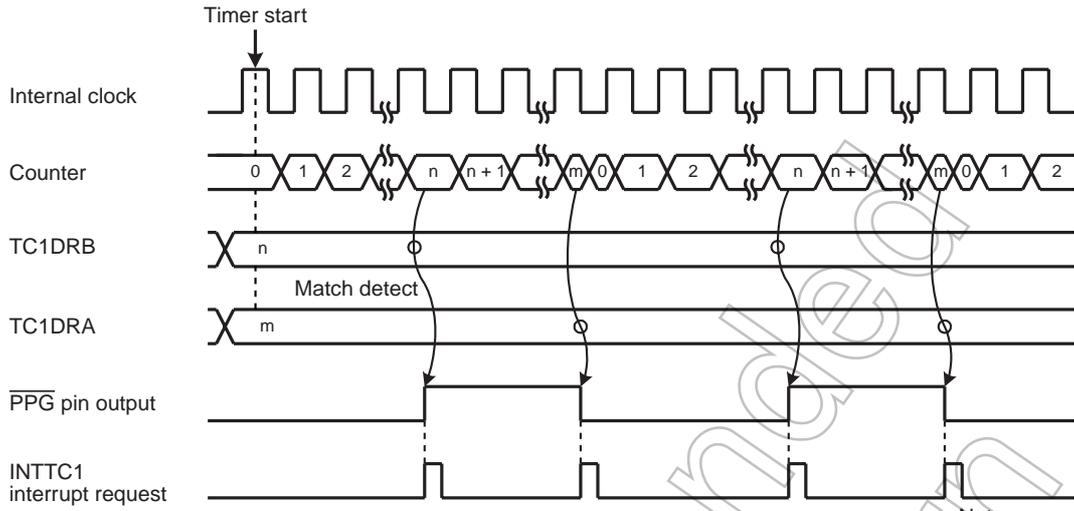
Note 1: To change TC1DRA or TC1DRB during a run of the timer, set a value sufficiently larger than the count value of the counter. Setting a value smaller than the count value of the counter during a run of the timer may generate a pulse different from that specified.

Note 2: Do not change TC1CR<TFF1> during a run of the timer. TC1CR<TFF1> can be set correctly only at initialization (after reset). When the timer stops during PPG, TC1CR<TFF1> can not be set correctly from this point onward if the PPG output has the level which is inverted of the level when the timer starts. (Setting TC1CR<TFF1> specifies the timer F/F1 to the level inverted of the programmed value.) Therefore, the timer F/F1 needs to be initialized to ensure an arbitrary level of the PPG output. To initialize the timer F/F1, change TC1CR<TC1M> to the timer mode (it is not required to start the timer mode), and then set the PPG mode. Set TC1CR<TFF1> at this time.

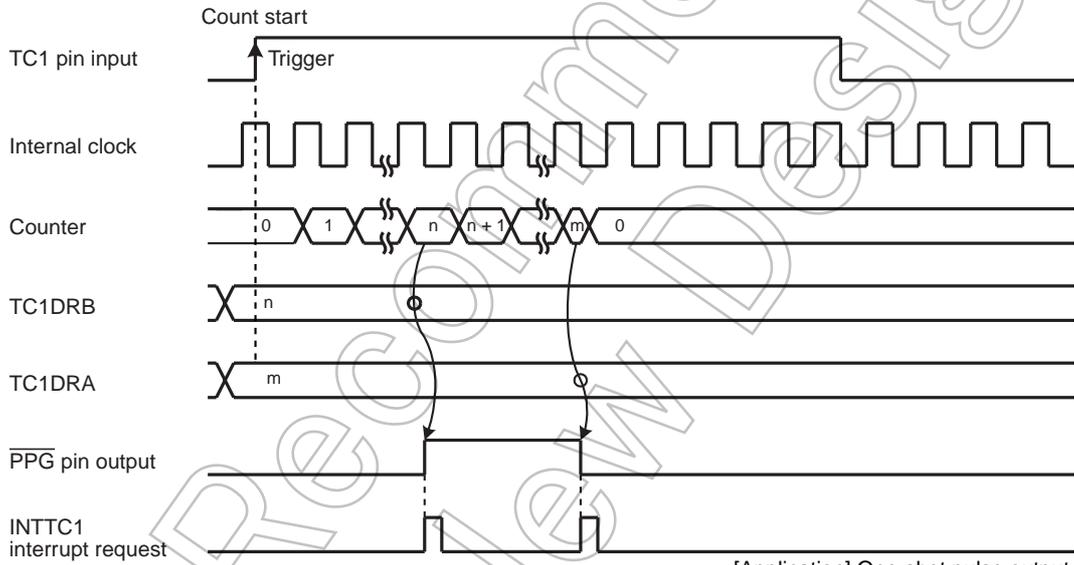
Note 3: In the PPG mode, the following relationship must be satisfied.

$$\text{TC1DRA} > \text{TC1DRB}$$

Note 4: Set TC1DRB after changing the mode of TC1M to the PPG mode.



(a) Continuous pulse generation (TC1S = 01)



[Application] One-shot pulse output

(b) One-shot pulse generation (TC1S = 10)

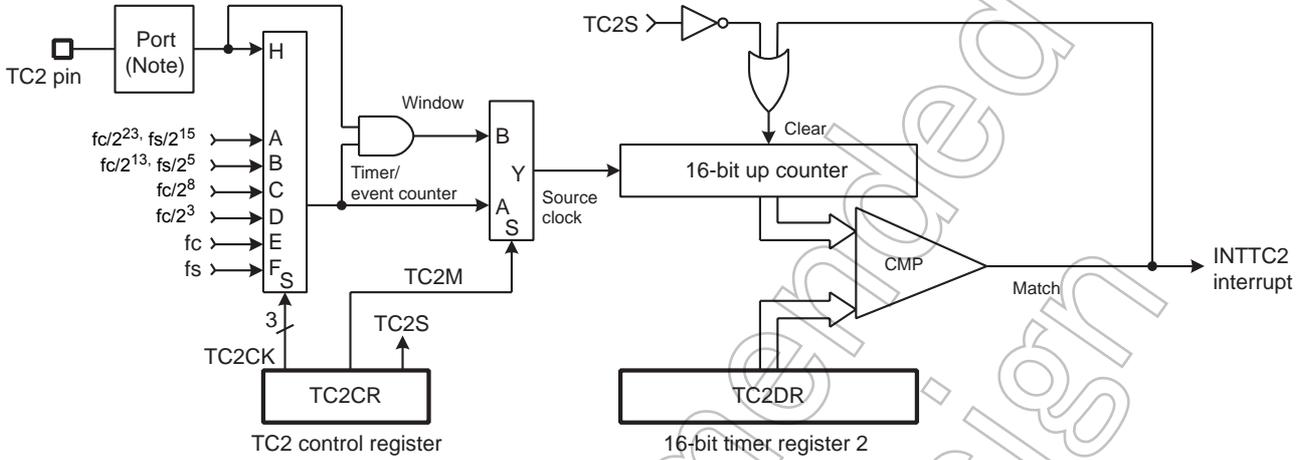
Note: $m > n$

Figure 8-8 PPG Mode Timing Chart

Not Recommended
for New Design

9. 16-Bit Timer/Counter2 (TC2)

9.1 Configuration

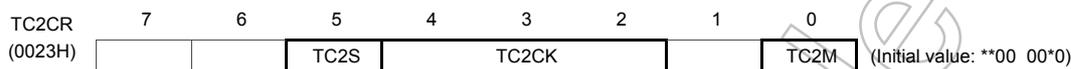
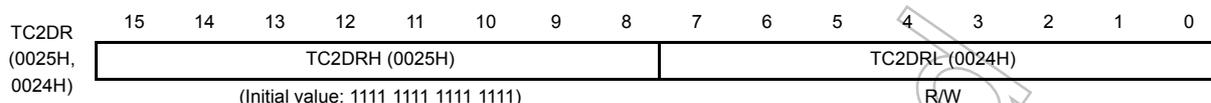


Note: When control input/output is used, I/O port setting should be set correctly. For details, refer to the section "I/O ports".

Figure 9-1 Timer/Counter2 (TC2)

9.2 Control

The timer/counter 2 is controlled by a timer/counter 2 control register (TC2CR) and a 16-bit timer register 2 (TC2DR).



| TC2S | TC2 start control | 0: Stop and counter clear 1: Start | R/W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------|--|---|-------------------------|-------------|---------|--------------|---------------|--------------|---------------|-----|-------------|-------------|------|-------------|-----|-------------|----------|------|----------|-----|----------|----------|-----|---|-----|----------|----------|-----|---|-----|---|---|---|------------|-----|----|----|---|---|-----|----------|--|--|--|-----|--------------------------------|--|--|--|-----|
| TC2CK | TC2 source clock select Unit : [Hz] | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">NORMAL1/2, IDLE1/2 mode</th> <th rowspan="2" style="text-align: center;">Divider</th> <th rowspan="2" style="text-align: center;">SLOW1/2 mode</th> <th rowspan="2" style="text-align: center;">SLEEP1/2 mode</th> </tr> <tr> <th style="text-align: center;">DV7CK = 0</th> <th style="text-align: center;">DV7CK = 1</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">000</td> <td style="text-align: center;">$fc/2^{23}$</td> <td style="text-align: center;">$fs/2^{15}$</td> <td style="text-align: center;">DV21</td> <td style="text-align: center;">$fs/2^{15}$</td> </tr> <tr> <td style="text-align: center;">001</td> <td style="text-align: center;">$fc/2^{13}$</td> <td style="text-align: center;">$fs/2^5$</td> <td style="text-align: center;">DV11</td> <td style="text-align: center;">$fs/2^5$</td> </tr> <tr> <td style="text-align: center;">010</td> <td style="text-align: center;">$fc/2^8$</td> <td style="text-align: center;">$fc/2^8$</td> <td style="text-align: center;">DV6</td> <td style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">011</td> <td style="text-align: center;">$fc/2^3$</td> <td style="text-align: center;">$fc/2^3$</td> <td style="text-align: center;">DV1</td> <td style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">100</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> <td style="text-align: center;">fc (Note7)</td> </tr> <tr> <td style="text-align: center;">101</td> <td style="text-align: center;">fs</td> <td style="text-align: center;">fs</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> </tr> <tr> <td style="text-align: center;">110</td> <td colspan="3" style="text-align: center;">Reserved</td> <td></td> </tr> <tr> <td style="text-align: center;">111</td> <td colspan="3" style="text-align: center;">External clock (TC2 pin input)</td> <td></td> </tr> </tbody> </table> | NORMAL1/2, IDLE1/2 mode | | Divider | SLOW1/2 mode | SLEEP1/2 mode | DV7CK = 0 | DV7CK = 1 | 000 | $fc/2^{23}$ | $fs/2^{15}$ | DV21 | $fs/2^{15}$ | 001 | $fc/2^{13}$ | $fs/2^5$ | DV11 | $fs/2^5$ | 010 | $fc/2^8$ | $fc/2^8$ | DV6 | - | 011 | $fc/2^3$ | $fc/2^3$ | DV1 | - | 100 | - | - | - | fc (Note7) | 101 | fs | fs | - | - | 110 | Reserved | | | | 111 | External clock (TC2 pin input) | | | | R/W |
| | | NORMAL1/2, IDLE1/2 mode | | Divider | | | | SLOW1/2 mode | SLEEP1/2 mode | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | DV7CK = 0 | DV7CK = 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 000 | $fc/2^{23}$ | $fs/2^{15}$ | DV21 | $fs/2^{15}$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 001 | $fc/2^{13}$ | $fs/2^5$ | DV11 | $fs/2^5$ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 010 | $fc/2^8$ | $fc/2^8$ | DV6 | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 011 | $fc/2^3$ | $fc/2^3$ | DV1 | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 100 | - | - | - | fc (Note7) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 101 | fs | fs | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 110 | Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 111 | External clock (TC2 pin input) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TC2M | TC2 operating mode select | 0: Timer/event counter mode 1: Window mode | R/W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Note 2: When writing to the Timer Register 2 (TC2DR), always write to the lower side (TC2DRL) and then the upper side (TC2DRH) in that order. Writing to only the lower side (TC2DRL) or the upper side (TC2DRH) has no effect.

Note 3: The timer register 2 (TC2DR) uses the value previously set in it for coincidence detection until data is written to the upper side (TC2DRH) after writing data to the lower side (TC2DRL).

Note 4: Set the mode and source clock when the TC2 stops (TC2S = 0).

Note 5: Values to be loaded to the timer register must satisfy the following condition.
 $TC2DR > 1$ ($TC2DR_{15}$ to $TC2DR_{11} > 1$ at warm up)

Note 6: If a read instruction is executed for TC2CR, read data of bit 7, 6 and 1 are unstable.

Note 7: The high-frequency clock (fc) can be selected only when the time mode at SLOW2 mode is selected.

Note 8: On entering STOP mode, the TC2 start control (TC2S) is cleared to "0" automatically. So, the timer stops. Once the STOP mode has been released, to start using the timer counter, set TC2S again.

9.3 Function

The timer/counter 2 has three operating modes: timer, event counter and window modes.

And if f_c or f_s is selected as the source clock in timer mode, when switching the timer mode from SLOW1 to NORMAL2, the timer/counter2 can generate warm-up time until the oscillator is stable.

9.3.1 Timer mode

In this mode, the internal clock is used for counting up. The contents of TC2DR are compared with the contents of up counter. If a match is found, a timer/counter 2 interrupt (INTTC2) is generated, and the counter is cleared. Counting up is resumed after the counter is cleared.

When f_c is selected for source clock at SLOW2 mode, lower 11-bits of TC2DR are ignored and generated a interrupt by matching upper 5-bits only. Though, in this situation, it is necessary to set TC2DRH only.

Table 9-1 Source Clock (Internal clock) for Timer/Counter2 (at $f_c = 16 \text{ MHz}$, DV7CK=0)

| TC2C K | NORMAL1/2, IDLE1/2 mode | | | | SLOW1/2 mode | | SLEEP1/2 mode | |
|-----------|-------------------------|---------------------------|------------|---------------------------|-----------------|---------------------------------|-----------------|---------------------------------|
| | DV7CK = 0 | | DV7CK = 1 | | Resolu- tion | Maxi- mum Time Setting | Resolu- tion | Maxi- mum Time Setting |
| | Resolution | Maximum Time Set- ting | Resolution | Maximum Time Set- ting | | | | |
| 000 | 524.29 [ms] | 9.54 [h] | 1 [s] | 18.2 [h] | 1 [s] | 18.2 [h] | 1 [s] | 18.2 [h] |
| 001 | 512.0 [ms] | 33.55 [s] | 0.98 [ms] | 1.07 [min] | 0.98 [ms] | 1.07 [min] | 0.98 [ms] | 1.07 [min] |
| 010 | 16.0 [ms] | 1.05 [s] | 16.0 [ms] | 1.05 [s] | - | - | - | - |
| 011 | 0.5 [ms] | 32.77 [ms] | 0.5 [ms] | 32.77 [ms] | - | - | - | - |
| 100 | - | - | - | - | 62.5 [ns] | - | - | - |
| 101 | 30.52 [ms] | 2 [s] | 30.52 [ms] | 2 [s] | - | - | - | - |

Note: When f_c is selected as the source clock in timer mode, it is used at warm-up for switching from SLOW1 mode to NORMAL2 mode.

Example :Sets the timer mode with source clock $f_c/2^3$ [Hz] and generates an interrupt every 25 ms (at $f_c = 16 \text{ MHz}$)

```
LDW      (TC2DR), 061AH      ; Sets TC2DR (25 ms * 28/fc = 061AH)
DI       ; IMF= "0"
SET      (EIRH), 5          ; Enables INTTC2 interrupt
EI       ; IMF= "1"
LD       (TC2CR), 00001000B ; Source clock / mode select
LD       (TC2CR), 00101000B ; Starts Timer
```

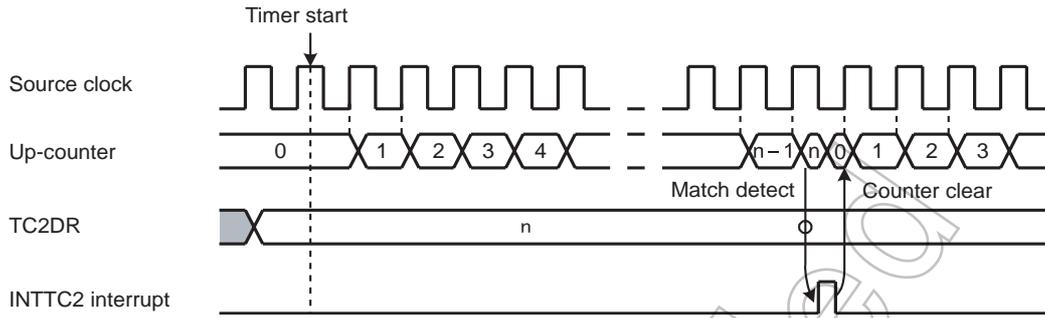


Figure 9-2 Timer Mode Timing Chart

Not Recommended for New Design

9.3.2 Event counter mode

In this mode, events are counted on the rising edge of the TC2 pin input. The contents of TC2DR are compared with the contents of the up counter. If a match is found, an INTTC2 interrupt is generated, and the counter is cleared. Counting up is resumed every the rising edge of the TC2 pin input after the up counter is cleared.

Match detect is executed on the falling edge of the TC2 pin. Therefore, an INTTC2 interrupt is generated at the falling edge after the match of TC2DR and up counter.

The minimum input pulse width of TC2 pin is shown in Table 9-2. Two or more machine cycles are required for both the “H” and “L” levels of the pulse width.

Example :Sets the event counter mode and generates an INTTC2 interrupt 640 counts later.

```
LDW      (TC2DR), 640      ; Sets TC2DR
DI       ; IMF= "0"
SET      (EIRH), 5        ; Enables INTTC2 interrupt
EI       ; IMF= "1"
LD       (TC2CR), 00011100B ; TC2 source vclock / mode select
LD       (TC2CR), 00111100B ; Starts TC2
```

Table 9-2 Timer/Counter 2 External Input Clock Pulse Width

| | Minimum Input Pulse Width [s] | |
|-----------|-------------------------------|--------------------------|
| | NORMAL 1/2, IDLE 1/2 mode | SLOW 1/2, SLEEP 1/2 mode |
| “H” width | $2^3/f_c$ | $2^3/f_s$ |
| “L” width | $2^3/f_c$ | $2^3/f_s$ |

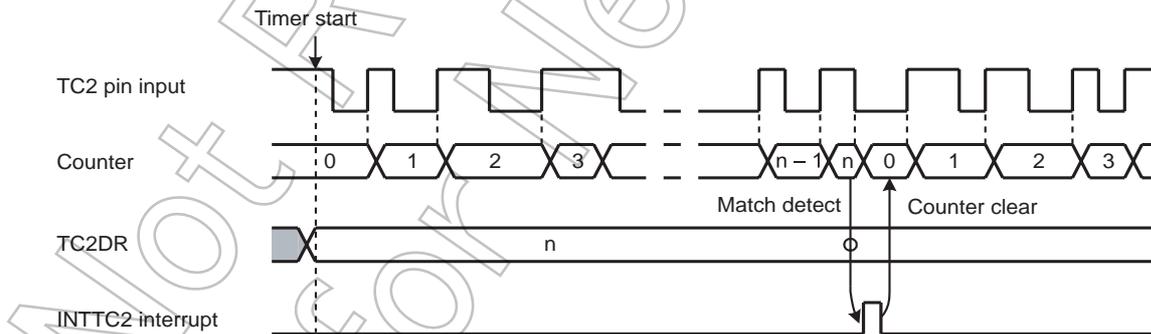


Figure 9-3 Event Counter Mode Timing Chart

9.3.3 Window mode

In this mode, counting up performed on the rising edge of an internal clock during TC2 external pin input (Window pulse) is “H” level. The contents of TC2DR are compared with the contents of up counter. If a match found, an INTTC2 interrupt is generated, and the up-counter is cleared.

The maximum applied frequency (TC2 input) must be considerably slower than the selected internal clock by the TC2CR<TC2CK>.

Note: It is not available window mode in the SLOW/SLEEP mode. Therefore, at the window mode in NORMAL mode, the timer should be halted by setting TC2CR<TC2S> to "0" before the SLOW/SLEEP mode is entered.

Example :Generates an interrupt, inputting “H” level pulse width of 120 ms or more. (at $f_c = 16 \text{ MHz}$, $\text{TBTCR}\langle\text{DV7CK}\rangle = \text{“0”}$)

```
LDW      (TC2DR), 00EAH      ; Sets TC2DR ( $120 \text{ ms} \cdot 2^{13}/f_c = 00EAH$ )
DI       ; IMF= “0”
SET      (EIRH), 5          ; Enables INTTC2 interrupt
EI       ; IMF= “1”
LD       (TC2CR), 00000101B ; TC2sorce clock / mode select
LD       (TC2CR), 00100101B ; Starts TC2
```

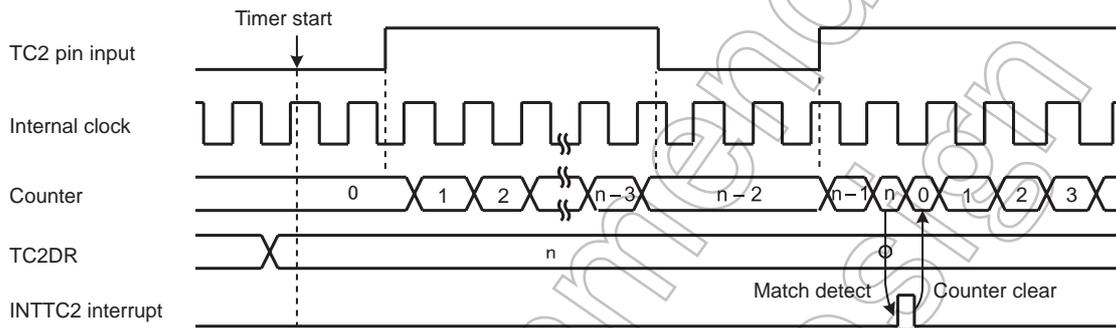


Figure 9-4 Window Mode Timing Chart

10. 8-Bit TimerCounter (TC3, TC4)

10.1 Configuration

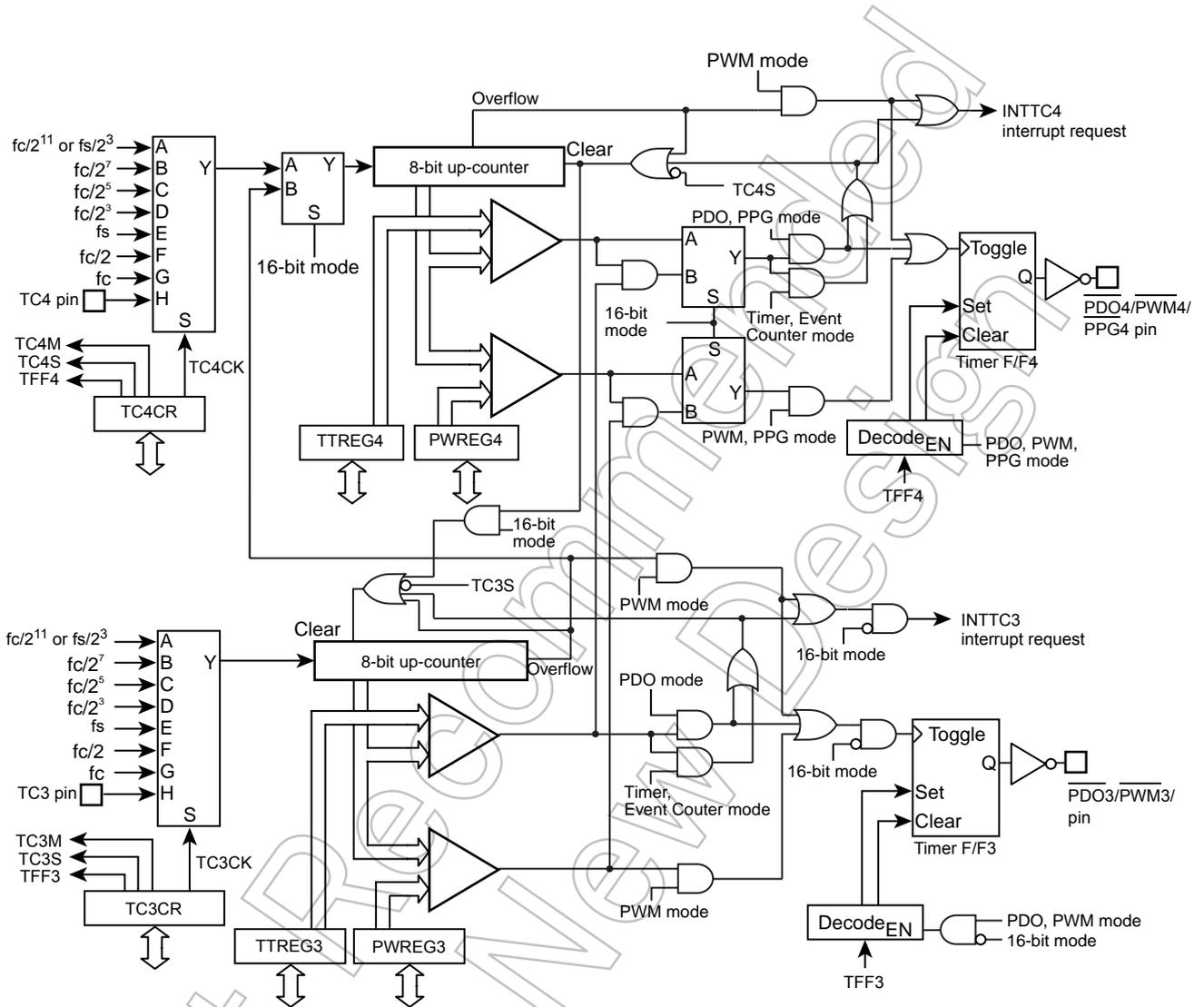
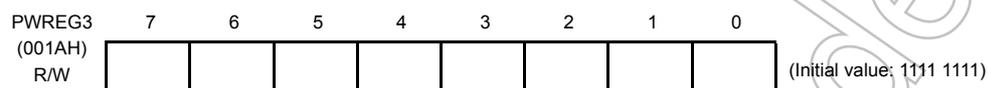
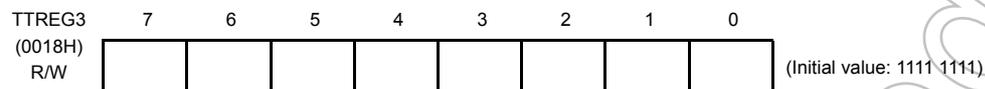


Figure 10-1 8-Bit TimerCounter 3, 4

10.2 TimerCounter Control

The TimerCounter 3 is controlled by the TimerCounter 3 control register (TC3CR) and two 8-bit timer registers (TTREG3, PWREG3).

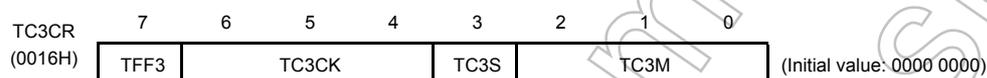
TimerCounter 3 Timer Register



Note 1: Do not change the timer register (TTREG3) setting while the timer is running.

Note 2: Do not change the timer register (PWREG3) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

TimerCounter 3 Control Register



| TFF3 | Time F/F3 control | 0: Clear 1: Set | R/W | | |
|-------|--------------------------------|--|-------------|----------|----------|
| TC3CK | Operating clock selection [Hz] | NORMAL1/2, IDLE1/2 mode | | | |
| | | SLOW1/2 SLEEP1/2 mode | | | |
| | | DV7CK = 0 | DV7CK = 1 | | |
| | | 000 | $fc/2^{11}$ | $fs/2^3$ | $fs/2^3$ |
| | | 001 | $fc/2^7$ | $fc/2^7$ | – |
| | | 010 | $fc/2^5$ | $fc/2^5$ | – |
| | | 011 | $fc/2^3$ | $fc/2^3$ | – |
| | | 100 | fs | fs | fs |
| 101 | fc/2 | fc/2 | – | | |
| 110 | fc | fc | fc (Note 8) | | |
| 111 | TC3 pin input | | | | |
| TC3S | TC3 start control | 0: Operation stop and counter clear 1: Operation start | R/W | | |
| TC3M | TC3M operating mode select | 000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) output mode 011: 16-bit mode (Each mode is selectable with TC4M.) 1**: Reserved | R/W | | |

Note 1: fc: High-frequency clock [Hz] fs: Low-frequency clock[Hz]

Note 2: Do not change the TC3M, TC3CK and TFF3 settings while the timer is running.

Note 3: To stop the timer operation (TC3S= 1 → 0), do not change the TC3M, TC3CK and TFF3 settings. To start the timer operation (TC3S= 0 → 1), TC3M, TC3CK and TFF3 can be programmed.

Note 4: To use the TimerCounter in the 16-bit mode, set the operating mode by programming TC4CR<TC4M>, where TC3M must be fixed to 011.

Note 5: To use the TimerCounter in the 16-bit mode, select the source clock by programming TC3CK. Set the timer start control and timer F/F control by programming TC4CR<TC4S> and TC4CR<TFF4>, respectively.

Note 6: The operating clock settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-1 and Table 10-2.

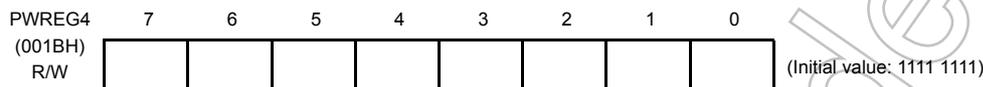
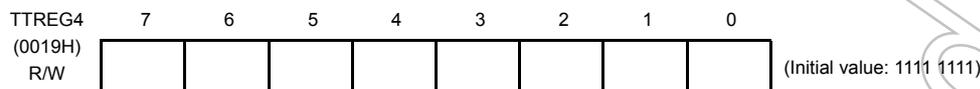
Note 7: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-3.

Note 8: The operating clock f_c in the SLOW or SLEEP mode can be used only as the high-frequency warm-up mode.

Not Recommended
for New Design

The TimerCounter 4 is controlled by the TimerCounter 4 control register (TC4CR) and two 8-bit timer registers (TTREG4 and PWREG4).

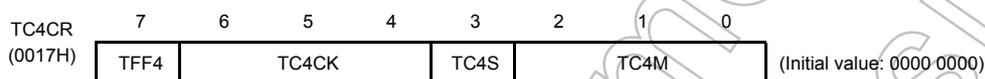
TimerCounter 4 Timer Register



Note 1: Do not change the timer register (TTREG4) setting while the timer is running.

Note 2: Do not change the timer register (PWREG4) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

TimerCounter 4 Control Register



| TFF4 | Timer F/F4 control | 0: Clear 1: Set | R/W | | |
|-------|--------------------------------|---|--------------------|-------------------|-------------------|
| TC4CK | Operating clock selection [Hz] | NORMAL 1/2, IDLE 1/2 mode | | | |
| | | SLOW 1/2 SLEEP 1/2 mode | | | |
| | | DV7CK = 0 | DV7CK = 1 | | |
| | | 000 | fs/2 ¹¹ | fs/2 ³ | fs/2 ³ |
| | | 001 | fc/2 ⁷ | fc/2 ⁷ | – |
| | | 010 | fc/2 ⁵ | fc/2 ⁵ | – |
| | | 011 | fc/2 ³ | fc/2 ³ | – |
| | | 100 | fs | fs | fs |
| 101 | fc/2 | fc/2 | – | | |
| 110 | fc | fc | – | | |
| 111 | TC4 pin input | | | | |
| TC4S | TC4 start control | 0: Operation stop and counter clear 1: Operation start | R/W | | |
| TC4M | TC4M operating mode select | 000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) output mode 011: Reserved 100: 16-bit timer/event counter mode 101: Warm-up counter mode 110: 16-bit pulse width modulation (PWM) output mode 111: 16-bit PPG mode | R/W | | |

Note 1: fc: High-frequency clock [Hz] fs: Low-frequency clock [Hz]

Note 2: Do not change the TC4M, TC4CK and TFF4 settings while the timer is running.

Note 3: To stop the timer operation (TC4S= 1 → 0), do not change the TC4M, TC4CK and TFF4 settings. To start the timer operation (TC4S= 0 → 1), TC4M, TC4CK and TFF4 can be programmed.

Note 4: When TC4M= 1** (upper byte in the 16-bit mode), the source clock becomes the TC3 overflow signal regardless of the TC4CK setting.

Note 5: To use the TimerCounter in the 16-bit mode, select the operating mode by programming TC4M, where TC3CR<TC3M> must be set to 011.

Note 6: To the TimerCounter in the 16-bit mode, select the source clock by programming TC3CR<TC3CK>. Set the timer start control and timer F/F control by programming TC4S and TFF4, respectively.

Note 7: The operating clock settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-1 and Table 10-2.

Note 8: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 10-3.

Table 10-1 Operating Mode and Selectable Source Clock (NORMAL1/2 and IDLE1/2 Modes)

| Operating mode | fc/2 ¹¹ or fs/2 ³ | fc/2 ⁷ | fc/2 ⁵ | fc/2 ³ | fs | fc/2 | fc | TC3 pin input | TC4 pin input |
|----------------------|---|-------------------|-------------------|-------------------|----|------|----|------------------|------------------|
| 8-bit timer | 0 | 0 | 0 | 0 | - | - | - | - | - |
| 8-bit event counter | - | - | - | - | - | - | - | 0 | 0 |
| 8-bit PDO | 0 | 0 | 0 | 0 | - | - | - | - | - |
| 8-bit PWM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | - |
| 16-bit timer | 0 | 0 | 0 | 0 | - | - | - | - | - |
| 16-bit event counter | - | - | - | - | - | - | - | 0 | - |
| Warm-up counter | - | - | - | - | 0 | - | - | - | - |
| 16-bit PWM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - |
| 16-bit PPG | 0 | 0 | 0 | 0 | - | - | - | 0 | - |

Note 1: For 16-bit operations (16-bit timer/event counter, warm-up counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bit (TC3CK).

Note 2: 0 : Available source clock

Table 10-2 Operating Mode and Selectable Source Clock (SLOW1/2 and SLEEP1/2 Modes)

| Operating mode | fc/2 ¹¹ or fs/2 ³ | fc/2 ⁷ | fc/2 ⁵ | fc/2 ³ | fs | fc/2 | fc | TC3 pin input | TC4 pin input |
|----------------------|---|-------------------|-------------------|-------------------|----|------|----|------------------|------------------|
| 8-bit timer | 0 | - | - | - | - | - | - | - | - |
| 8-bit event counter | - | - | - | - | - | - | - | 0 | 0 |
| 8-bit PDO | 0 | - | - | - | - | - | - | - | - |
| 8-bit PWM | 0 | - | - | - | 0 | - | - | - | - |
| 16-bit timer | 0 | - | - | - | - | - | - | - | - |
| 16-bit event counter | - | - | - | - | - | - | - | 0 | - |
| Warm-up counter | - | - | - | - | - | - | 0 | - | - |
| 16-bit PWM | 0 | - | - | - | 0 | - | - | 0 | - |
| 16-bit PPG | 0 | - | - | - | - | - | - | 0 | - |

Note1: For 16-bit operations (16-bit timer/event counter, warm-up counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bit (TC3CK).

Note2: 0 : Available source clock

Table 10-3 Constraints on Register Values Being Compared

| Operating mode | Register Value |
|----------------------------|---|
| 8-bit timer/event counter | $1 \leq (TTREGn) \leq 255$ |
| 8-bit PDO | $1 \leq (TTREGn) \leq 255$ |
| 8-bit PWM | $2 \leq (PWREGn) \leq 254$ |
| 16-bit timer/event counter | $1 \leq (TTREG4, 3) \leq 65535$ |
| Warm-up counter | $256 \leq (TTREG4, 3) \leq 65535$ |
| 16-bit PWM | $2 \leq (PWREG4, 3) \leq 65534$ |
| 16-bit PPG | $1 \leq (PWREG4, 3) < (TTREG4, 3) \leq 65535$ and $(PWREG4, 3) + 1 < (TTREG4, 3)$ |

Note: n = 3 to 4

Not Recommended for New Design

10.3 Function

The TimerCounter 3 and 4 have the 8-bit timer, 8-bit event counter, 8-bit programmable divider output (PDO), 8-bit pulse width modulation (PWM) output modes. The TimerCounter 3 and 4 (TC3, 4) are cascadable to form a 16-bit timer. The 16-bit timer has the operating modes such as the 16-bit timer, 16-bit event counter, warm-up counter, 16-bit pulse width modulation (PWM) output and 16-bit programmable pulse generation (PPG) modes.

10.3.1 8-Bit Timer Mode (TC3 and 4)

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the timer register j (TTREGj) value is detected, an INTTCj interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting.

Note 1: In the timer mode, fix TCjCR<TFFj> to 0. If not fixed, the \overline{PDO}_j , \overline{PWM}_j and \overline{PPG}_j pins may output pulses.

Note 2: In the timer mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the timer mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

Table 10-4 Source Clock for TimerCounter 3, 4 (Internal Clock)

| Source Clock | | SLOW1/2, SLEEP1/2 mode | Resolution | | Maximum Time Setting | |
|-------------------------|------------------------|------------------------|-------------|-----------------|----------------------|-----------------|
| NORMAL1/2, IDLE1/2 mode | | | fc = 16 MHz | fs = 32.768 kHz | fc = 16 MHz | fs = 32.768 kHz |
| DV7CK = 0 | DV7CK = 1 | | | | | |
| fc/2 ¹¹ [Hz] | fs/2 ³ [Hz] | fs/2 ³ [Hz] | 128 μs | 244.14 μs | 32.6 ms | 62.3 ms |
| fc/2 ⁷ | fc/2 ⁷ | – | 8 μs | – | 2.0 ms | – |
| fc/2 ⁵ | fc/2 ⁵ | – | 2 μs | – | 510 μs | – |
| fc/2 ³ | fc/2 ³ | – | 500 ns | – | 127.5 μs | – |

Example :Setting the timer mode with source clock fc/2⁷ Hz and generating an interrupt 80 μs later (TimerCounter4, fc = 16.0 MHz)

```
LD (TTREG4), 0AH : Sets the timer register (80 μs÷27/fc = 0AH).
DI
SET (EIRH), 1 : Enables INTTC4 interrupt.
EI
LD (TC4CR), 0001000B : Sets the operating clock to fc/27, and 8-bit timer mode.
LD (TC4CR), 00011000B : Starts TC4.
```

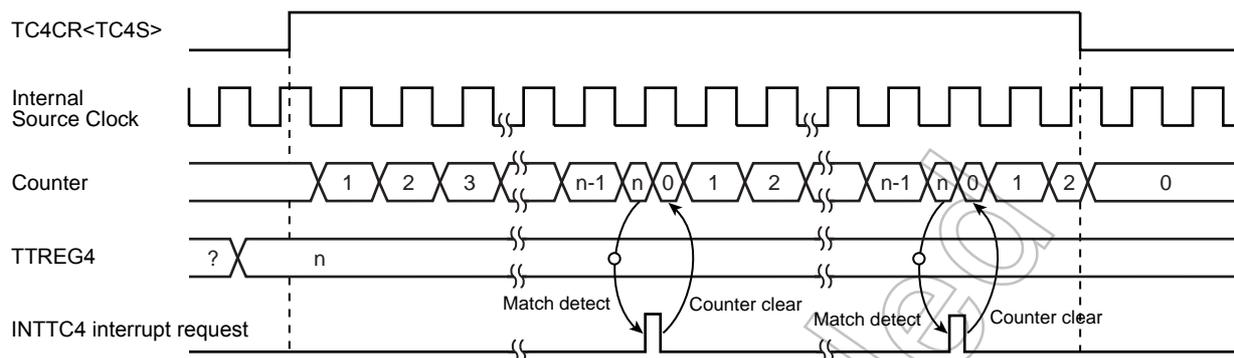


Figure 10-2 8-Bit Timer Mode Timing Chart (TC4)

10.3.2 8-Bit Event Counter Mode (TC3, 4)

In the 8-bit event counter mode, the up-counter counts up at the falling edge of the input pulse to the TCj pin. When a match between the up-counter and the TTREGj value is detected, an INTTCj interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at the falling edge of the input pulse to the TCj pin. Two machine cycles are required for the low- or high-level pulse input to the TCj pin. Therefore, a maximum frequency to be supplied is $fc/2^4$ Hz in the NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ Hz in the SLOW1/2 or SLEEP1/2 mode.

Note 1: In the event counter mode, fix TCjCR<TFFj> to 0. If not fixed, the \overline{PDOj} , $PWMj$ and \overline{PPGj} pins may output pulses.

Note 2: In the event counter mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the event counter mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

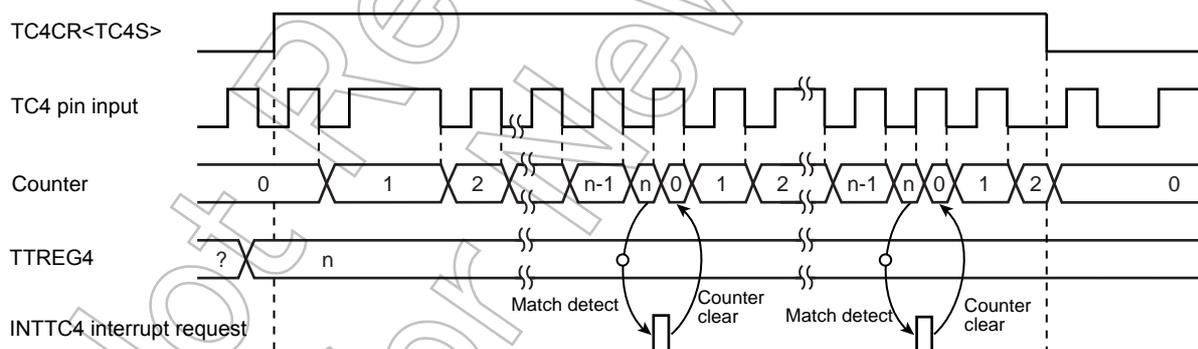


Figure 10-3 8-Bit Event Counter Mode Timing Chart (TC4)

10.3.3 8-Bit Programmable Divider Output (PDO) Mode (TC3, 4)

This mode is used to generate a pulse with a 50% duty cycle from the \overline{PDOj} pin.

In the PDO mode, the up-counter counts up using the internal clock. When a match between the up-counter and the TTREGj value is detected, the logic level output from the \overline{PDOj} pin is switched to the opposite state and the up-counter is cleared. The INTTCj interrupt request is generated at the time. The logic state opposite to the timer F/Fj logic level is output from the \overline{PDOj} pin. An arbitrary value can be set to the timer F/Fj by TCjCR<TFFj>. Upon reset, the timer F/Fj value is initialized to 0.

To use the programmable divider output, set the output latch of the I/O port to 1.

Example :Generating 1024 Hz pulse using TC4 ($f_c = 16.0$ MHz)

| Setting port | | |
|--------------|--------------------|---|
| LD | (TTREG4), 3DH | : $1/1024 \div 2^7 / f_c \div 2 = 3DH$ |
| LD | (TC4CR), 00010001B | : Sets the operating clock to $f_c/2^7$, and 8-bit PDO mode. |
| LD | (TC4CR), 00011001B | : Starts TC4. |

Note 1: In the programmable divider output mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the programmable divider output mode, the new value programmed in TTREGj is in effect immediately after programming. Therefore, if TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 2: When the timer is stopped during PDO output, the \overline{PDOj} pin holds the output status when the timer is stopped. To change the output status, program TCjCR<TFFj> after the timer is stopped. Do not change the TCjCR<TFFj> setting upon stopping of the timer.

Example: Fixing the \overline{PDOj} pin to the high level when the TimerCounter is stopped

CLR (TCjCR).3: Stops the timer.

CLR (TCjCR).7: Sets the \overline{PDOj} pin to the high level.

Note 3: j = 3, 4

Not Recommended for New Design

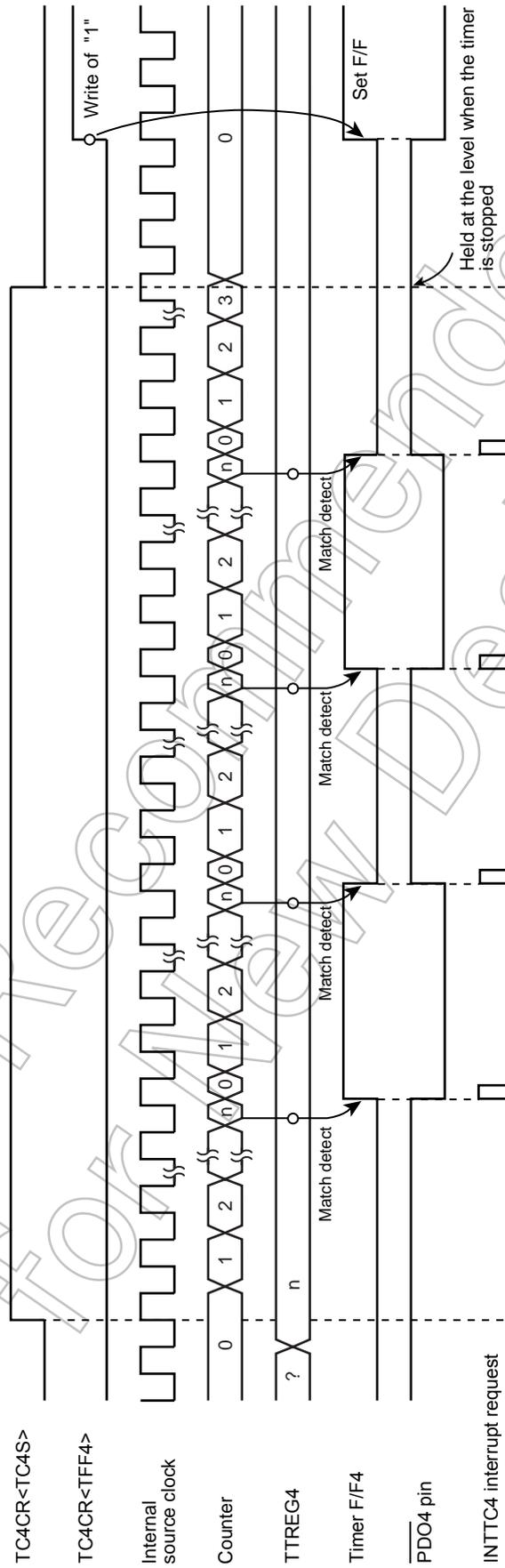


Figure 10-4 8-Bit PDO Mode Timing Chart (TC4)

10.3.4 8-Bit Pulse Width Modulation (PWM) Output Mode (TC3, 4)

This mode is used to generate a pulse-width modulated (PWM) signals with up to 8 bits of resolution. The up-counter counts up using the internal clock.

When a match between the up-counter and the PWREGj value is detected, the logic level output from the timer F/Fj is switched to the opposite state. The counter continues counting. The logic level output from the timer F/Fj is switched to the opposite state again by the up-counter overflow, and the counter is cleared. The INTTCj interrupt request is generated at this time.

Since the initial value can be set to the timer F/Fj by TCjCR<TFFj>, positive and negative pulses can be generated. Upon reset, the timer F/Fj is cleared to 0.

(The logic level output from the $\overline{\text{PWM}}_j$ pin is the opposite to the timer F/Fj logic level.)

Since PWREGj in the PWM mode is serially connected to the shift register, the value set to PWREGj can be changed while the timer is running. The value set to PWREGj during a run of the timer is shifted by the INTTCj interrupt request and loaded into PWREGj. While the timer is stopped, the value is shifted immediately after the programming of PWREGj. If executing the read instruction to PWREGj during PWM output, the value in the shift register is read, but not the value set in PWREGj. Therefore, after writing to PWREGj, the reading data of PWREGj is previous value until INTTCj is generated.

For the pin used for PWM output, the output latch of the I/O port must be set to 1.

Note 1: In the PWM mode, program the timer register PWREGj immediately after the INTTCj interrupt request is generated (normally in the INTTCj interrupt service routine.) If the programming of PWREGj and the interrupt request occur at the same time, an unstable value is shifted, that may result in generation of the pulse different from the programmed value until the next INTTCj interrupt request is generated.

Note 2: When the timer is stopped during PWM output, the $\overline{\text{PWM}}_j$ pin holds the output status when the timer is stopped. To change the output status, program TCjCR<TFFj> after the timer is stopped. Do not change the TCjCR<TFFj> upon stopping of the timer.

Example: Fixing the $\overline{\text{PWM}}_j$ pin to the high level when the TimerCounter is stopped

CLR (TCjCR).3: Stops the timer.

CLR (TCjCR).7: Sets the $\overline{\text{PWM}}_j$ pin to the high level.

Note 3: To enter the STOP mode during PWM output, stop the timer and then enter the STOP mode. If the STOP mode is entered without stopping the timer when fc, fc/2 or fs is selected as the source clock, a pulse is output from the $\overline{\text{PWM}}_j$ pin during the warm-up period time after exiting the STOP mode.

Note 4: j = 3, 4

Table 10-5 PWM Output Mode

| Source Clock | | | Resolution | | Repeated Cycle | |
|-------------------------|------------------------|------------------------|-------------|-----------------|----------------|-----------------|
| NORMAL1/2, IDLE1/2 mode | | SLOW1/2, SLEEP1/2 mode | fc = 16 MHz | fs = 32.768 kHz | fc = 16 MHz | fs = 32.768 kHz |
| DV7CK = 0 | DV7CK = 1 | | | | | |
| fc/2 ¹¹ [Hz] | fs/2 ³ [Hz] | fs/2 ³ [Hz] | 128 μs | 244.14 μs | 32.8 ms | 62.5 ms |
| fc/2 ⁷ | fc/2 ⁷ | — | 8 μs | — | 2.05 ms | — |
| fc/2 ⁵ | fc/2 ⁵ | — | 2 μs | — | 512 μs | — |
| fc/2 ³ | fc/2 ³ | — | 500 ns | — | 128 μs | — |
| fs | fs | fs | 30.5 μs | 30.5 μs | 7.81 ms | 7.81 ms |
| fc/2 | fc/2 | — | 125 ns | — | 32 μs | — |
| fc | fc | — | 62.5 ns | — | 16 μs | — |

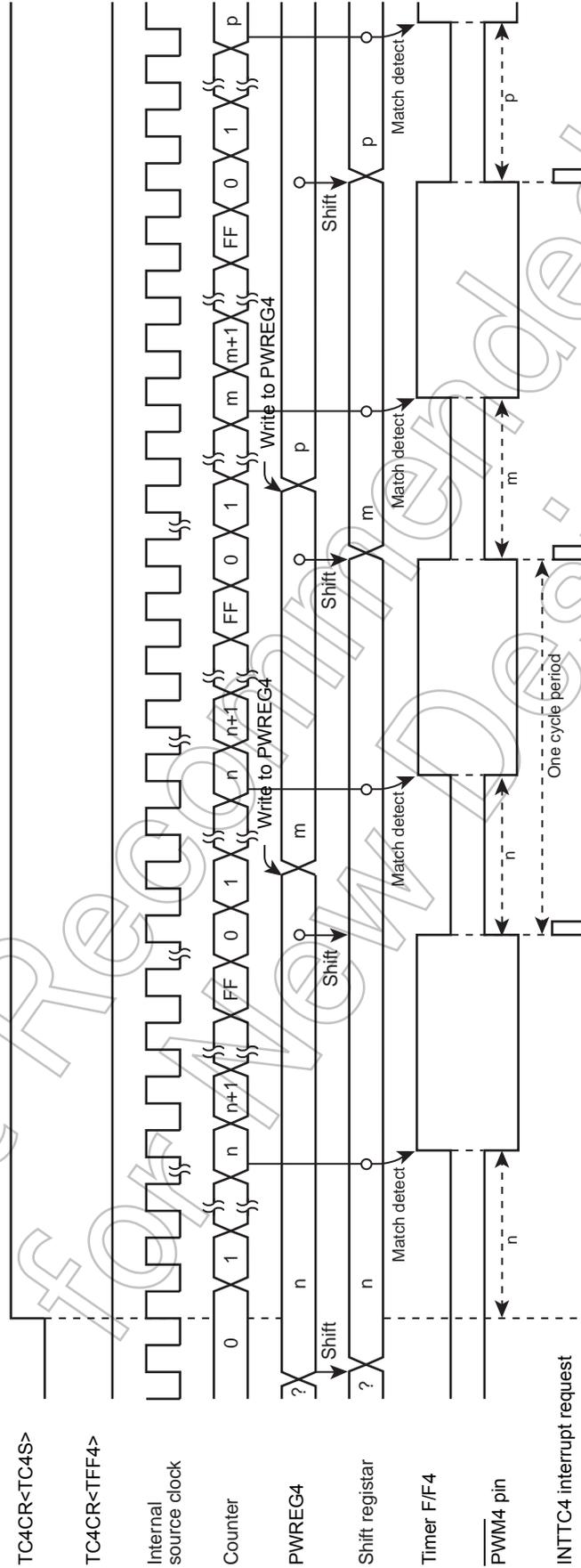


Figure 10-5 8-Bit PWM Mode Timing Chart (TC4)

10.3.5 16-Bit Timer Mode (TC3 and 4)

In the timer mode, the up-counter counts up using the internal clock. The TimerCounter 3 and 4 are cascaded to form a 16-bit timer.

When a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected after the timer is started by setting TC4CR<TC4S> to 1, an INTTC4 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter continues counting. Program the lower byte and upper byte in this order in the timer register. (Programming only the upper or lower byte should not be attempted.)

Note 1: In the timer mode, fix TCjCR<TFFj> to 0. If not fixed, the $\overline{P\bar{D}O_j}$, $\overline{P\bar{W}M_j}$, and $\overline{P\bar{P}G_j}$ pins may output a pulse.

Note 2: In the timer mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the timer mode, the new value programmed in TTREGj is in effect immediately after programming of TTREGj. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

Table 10-6 Source Clock for 16-Bit Timer Mode

| Source Clock | | SLOW1/2, SLEEP1/2 mode | Resolution | | Maximum Time Setting | |
|-------------------------|-------------------|------------------------------|-------------|-----------------|----------------------|-----------------|
| NORMAL1/2, IDLE1/2 mode | | | fc = 16 MHz | fs = 32.768 kHz | fc = 16 MHz | fs = 32.768 kHz |
| DV7CK = 0 | DV7CK = 1 | | | | | |
| fc/2 ¹¹ | fs/2 ³ | fs/2 ³ | 128 μs | 244.14 μs | 8.39 s | 16 s |
| fc/2 ⁷ | fc/2 ⁷ | – | 8 μs | – | 524.3 ms | – |
| fc/2 ⁵ | fc/2 ⁵ | – | 2 μs | – | 131.1 ms | – |
| fc/2 ³ | fc/2 ³ | – | 500 ns | – | 32.8 ms | – |

Example :Setting the timer mode with source clock fc/2⁷ Hz, and generating an interrupt 300 ms later (fc = 16.0 MHz)

- LDW (TTREG3), 927CH : Sets the timer register (300 ms=2⁷/fc = 927CH).
- DI
- SET (EIRH), 1 : Enables INTTC4 interrupt.
- EI
- LD (TC3CR), 13H : Sets the operating clock to fc/2⁷, and 16-bit timer mode (lower byte).
- LD (TC4CR), 04H : Sets the 16-bit timer mode (upper byte).
- LD (TC4CR), 0CH : Starts the timer.

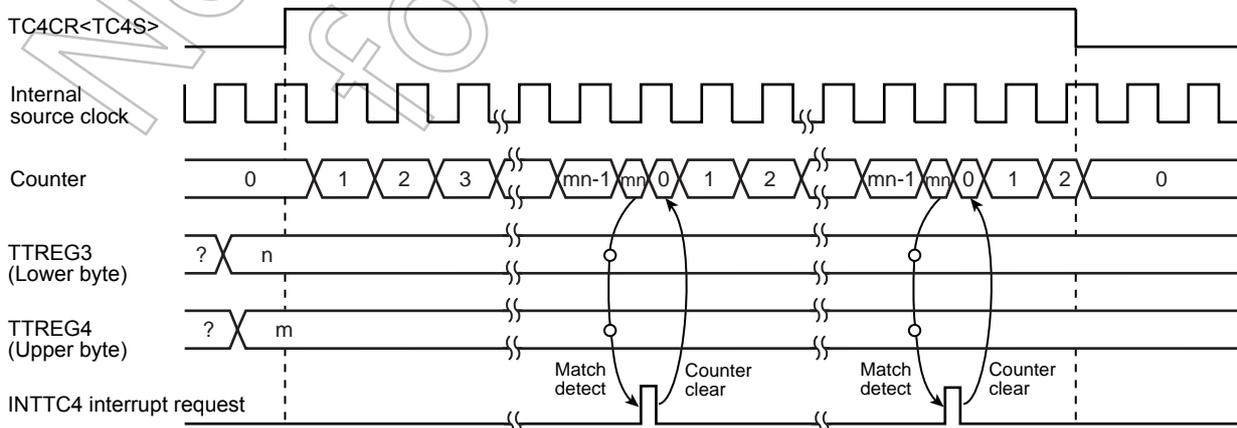


Figure 10-6 16-Bit Timer Mode Timing Chart (TC3 and TC4)

10.3.6 16-Bit Event Counter Mode (TC3 and 4)

In the event counter mode, the up-counter counts up at the falling edge to the TC3 pin. The TimerCounter 3 and 4 are cascadable to form a 16-bit event counter.

When a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected after the timer is started by setting TC4CR<TC4S> to 1, an INTTC4 interrupt is generated and the up-counter is cleared.

After being cleared, the up-counter restarts counting at the falling edge of the input pulse to the TC3 pin. Two machine cycles are required for the low- or high-level pulse input to the TC3 pin.

Therefore, a maximum frequency to be supplied is $f_c/2^4$ Hz in the NORMAL1/2 or IDLE1/2 mode, and $f_s/2^4$ in the SLOW1/2 or SLEEP1/2 mode. Program the lower byte (TTREG3), and upper byte (TTREG4) in this order in the timer register. (Programming only the upper or lower byte should not be attempted.)

Note 1: In the event counter mode, fix TCjCR<TFFj> to 0. If not fixed, the \overline{PDOj} , \overline{PWMj} and \overline{PPGj} pins may output pulses.

Note 2: In the event counter mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the event counter mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

10.3.7 16-Bit Pulse Width Modulation (PWM) Output Mode (TC3 and 4)

This mode is used to generate a pulse-width modulated (PWM) signals with up to 16 bits of resolution. The TimerCounter 3 and 4 are cascadable to form the 16-bit PWM signal generator.

The counter counts up using the internal clock or external clock.

When a match between the up-counter and the timer register (PWREG3, PWREG4) value is detected, the logic level output from the timer F/F4 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F4 is switched to the opposite state again by the counter overflow, and the counter is cleared. The INTTC4 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC3 pin. Therefore, a maximum frequency to be supplied is $f_c/2^4$ Hz in the NORMAL1/2 or IDLE1/2 mode, and $f_s/2^4$ to in the SLOW1/2 or SLEEP1/2 mode.

Since the initial value can be set to the timer F/F4 by TC4CR<TFF4>, positive and negative pulses can be generated. Upon reset, the timer F/F4 is cleared to 0.

(The logic level output from the $\overline{PWM4}$ pin is the opposite to the timer F/F4 logic level.)

Since PWREG4 and 3 in the PWM mode are serially connected to the shift register, the values set to PWREG4 and 3 can be changed while the timer is running. The values set to PWREG4 and 3 during a run of the timer are shifted by the INTTCj interrupt request and loaded into PWREG4 and 3. While the timer is stopped, the values are shifted immediately after the programming of PWREG4 and 3. Set the lower byte (PWREG3) and upper byte (PWREG4) in this order to program PWREG4 and 3. (Programming only the lower or upper byte of the register should not be attempted.)

If executing the read instruction to PWREG4 and 3 during PWM output, the values set in the shift register is read, but not the values set in PWREG4 and 3. Therefore, after writing to the PWREG4 and 3, reading data of PWREG4 and 3 is previous value until INTTC4 is generated.

For the pin used for PWM output, the output latch of the I/O port must be set to 1.

Note 1: In the PWM mode, program the timer register PWREG4 and 3 immediately after the INTTC4 interrupt request is generated (normally in the INTTC4 interrupt service routine.) If the programming of PWREGj and the interrupt request occur at the same time, an unstable value is shifted, that may result in generation of pulse different from the programmed value until the next INTTC4 interrupt request is generated.

Note 2: When the timer is stopped during PWM output, the $\overline{PWM4}$ pin holds the output status when the timer is stopped. To change the output status, program TC4CR<TFF4> after the timer is stopped. Do not program TC4CR<TFF4> upon stopping of the timer.

Example: Fixing the $\overline{PWM4}$ pin to the high level when the TimerCounter is stopped

CLR (TC4CR).3: Stops the timer.
 CLR (TC4CR).7 : Sets the PWM4 pin to the high level.

Note 3: To enter the STOP mode, stop the timer and then enter the STOP mode. If the STOP mode is entered without stopping of the timer when f_c , $f_c/2$ or f_s is selected as the source clock, a pulse is output from the PWM4 pin during the warm-up period time after exiting the STOP mode.

Table 10-7 16-Bit PWM Output Mode

| Source Clock | | SLOW1/2, SLEEP1/2 mode | Resolution | | Repeated Cycle | |
|-------------------------|------------------------|------------------------------|------------------------|----------------------------|------------------------|----------------------------|
| NORMAL1/2, IDLE1/2 mode | | | $f_c = 16 \text{ MHz}$ | $f_s = 32.768 \text{ kHz}$ | $f_c = 16 \text{ MHz}$ | $f_s = 32.768 \text{ kHz}$ |
| DV7CK = 0 | DV7CK = 1 | | | | | |
| $f_c/2^{11}$ | $f_s/2^3 \text{ [Hz]}$ | $f_s/2^3 \text{ [Hz]}$ | 128 μs | 244.14 μs | 8.39 s | 16 s |
| $f_c/2^7$ | $f_c/2^7$ | – | 8 μs | – | 524.3 ms | – |
| $f_c/2^5$ | $f_c/2^5$ | – | 2 μs | – | 131.1 ms | – |
| $f_c/2^3$ | $f_c/2^3$ | – | 500 ns | – | 32.8 ms | – |
| f_s | f_s | f_s | 30.5 μs | 30.5 μs | 2 s | 2 s |
| $f_c/2$ | $f_c/2$ | – | 125 ns | – | 8.2 ms | – |
| f_c | f_c | – | 62.5 ns | – | 4.1 ms | – |

Example :Generating a pulse with 1-ms high-level width and a period of 32.768 ms ($f_c = 16.0 \text{ MHz}$)

- Setting ports
- LDW (PWREG3), 07D0H : Sets the pulse width.
 - LD (TC3CR), 33H : Sets the operating clock to $f_c/2^3$, and 16-bit PWM output mode (lower byte).
 - LD (TC4CR), 056H : Sets TFF4 to the initial value 0, and 16-bit PWM signal generation mode (upper byte).
 - LD (TC4CR), 05EH : Starts the timer.

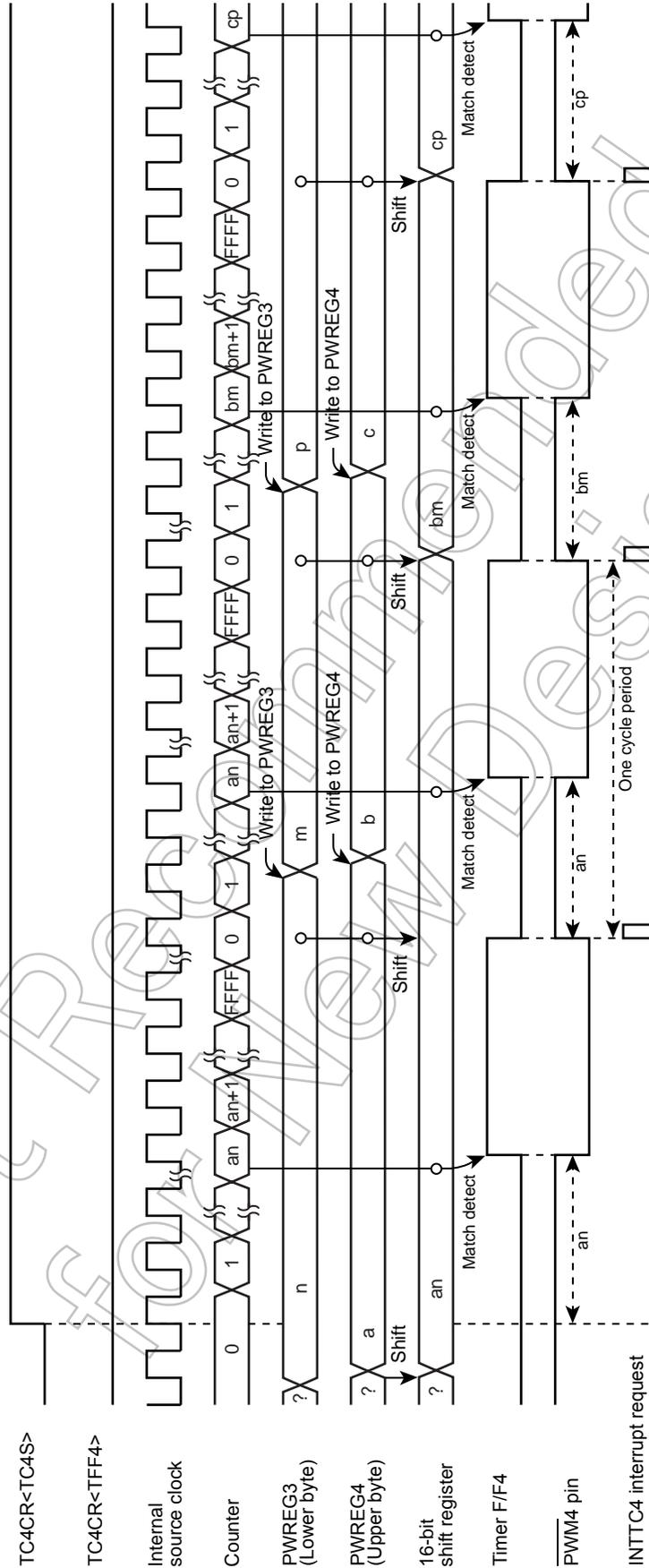


Figure 10-7 16-Bit PWM Mode Timing Chart (TC3 and TC4)

10.3.8 16-Bit Programmable Pulse Generate (PPG) Output Mode (TC3 and 4)

This mode is used to generate pulses with up to 16-bits of resolution. The timer counter 3 and 4 are cascaded to enter the 16-bit PPG mode.

The counter counts up using the internal clock or external clock. When a match between the up-counter and the timer register (PWREG3, PWREG4) value is detected, the logic level output from the timer F/F4 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F4 is switched to the opposite state again when a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected, and the counter is cleared. The INTTC4 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC3 pin. Therefore, a maximum frequency to be supplied is $f_c/2^4$ Hz in the NORMAL1/2 or IDLE1/2 mode, and $f_s/2^4$ to in the SLOW1/2 or SLEEP1/2 mode.

Since the initial value can be set to the timer F/F4 by TC4CR<TFF4>, positive and negative pulses can be generated. Upon reset, the timer F/F4 is cleared to 0.

(The logic level output from the $\overline{PPG4}$ pin is the opposite to the timer F/F4.)

Set the lower byte and upper byte in this order to program the timer register. (TTREG3 → TTREG4, PWREG3 → PWREG4) (Programming only the upper or lower byte should not be attempted.)

For PPG output, set the output latch of the I/O port to 1.

Example :Generating a pulse with 1-ms high-level width and a period of 16.385 ms ($f_c = 16.0$ MHz)

| Setting ports | | |
|---------------|-----------------|---|
| LDW | (PWREG3), 07D0H | : Sets the pulse width. |
| LDW | (TTREG3), 8002H | : Sets the cycle period. |
| LD | (TC3CR), 33H | : Sets the operating clock to $f_c/2^3$, and 16-bit PPG mode (lower byte). |
| LD | (TC4CR), 057H | : Sets TFF4 to the initial value 0, and 16-bit PPG mode (upper byte). |
| LD | (TC4CR), 05FH | : Starts the timer. |

Note 1: In the PPG mode, do not change the PWREGi and TTREGi settings while the timer is running. Since PWREGi and TTREGi are not in the shift register configuration in the PPG mode, the new values programmed in PWREGi and TTREGi are in effect immediately after programming PWREGi and TTREGi. Therefore, if PWREGi and TTREGi are changed while the timer is running, an expected operation may not be obtained.

Note 2: When the timer is stopped during PPG output, the $\overline{PPG4}$ pin holds the output status when the timer is stopped. To change the output status, program TC4CR<TFF4> after the timer is stopped. Do not change TC4CR<TFF4> upon stopping of the timer.

Example: Fixing the $\overline{PPG4}$ pin to the high level when the TimerCounter is stopped

CLR (TC4CR).3: Stops the timer

CLR (TC4CR).7: Sets the $\overline{PPG4}$ pin to the high level

Note 3: i = 3, 4

10.3.9 Warm-Up Counter Mode

In this mode, the warm-up period time is obtained to assure oscillation stability when the system clocking is switched between the high-frequency and low-frequency. The timer counter 3 and 4 are cascadable to form a 16-bit TimerCounter. The warm-up counter mode has two types of mode; switching from the high-frequency to low-frequency, and vice-versa.

Note 1: In the warm-up counter mode, fix TCiCR<TFFi> to 0. If not fixed, the $\overline{PD0i}$, \overline{PWMi} and \overline{PPGi} pins may output pulses.

Note 2: In the warm-up counter mode, only upper 8 bits of the timer register TTREG4 and 3 are used for match detection and lower 8 bits are not used.

Note 3: i = 3, 4

10.3.9.1 Low-Frequency Warm-up Counter Mode (NORMAL1 → NORMAL2 → SLOW2 → SLOW1)

In this mode, the warm-up period time from a stop of the low-frequency clock f_s to oscillation stability is obtained. Before starting the timer, set SYSCR2<XTEN> to 1 to oscillate the low-frequency clock. When a match between the up-counter and the timer register (TTREG4, 3) value is detected after the timer is started by setting TC4CR<TC4S> to 1, the counter is cleared by generating the INTTC4 interrupt request. After stopping the timer in the INTTC4 interrupt service routine, set SYSCR2<SYSCK> to 1 to switch the system clock from the high-frequency to low-frequency, and then clear of SYSCR2<XEN> to 0 to stop the high-frequency clock.

Table 10-8 Setting Time of Low-Frequency Warm-Up Counter Mode ($f_s = 32.768$ kHz)

| Minimum Time Setting (TTREG4, 3 = 0100H) | Maximum Time Setting (TTREG4, 3 = FF00H) |
|---|---|
| 7.81 ms | 1.99 s |

Example :After checking low-frequency clock oscillation stability with TC4 and 3, switching to the SLOW1 mode

```

SET      (SYSCR2).6      : SYSCR2<XTEN> ← 1
LD       (TC3CR), 43H    : Sets TFF3=0, source clock  $f_s$ , and 16-bit mode.
LD       (TC4CR), 05H    : Sets TFF4=0, and warm-up counter mode.
LD       (TTREG3), 8000H : Sets the warm-up time.
                                     (The warm-up time depends on the oscillator characteristic.)
DI       : IMF ← 0
SET      (EIRH). 1      : Enables the INTTC4.
EI       : IMF ← 1
SET      (TC4CR).3      : Starts TC4 and 3.
PINTTC4: CLR      (TC4CR).3 : Stops TC4 and 3.
SET      (SYSCR2).5      : SYSCR2<SYSCK> ← 1
                                     (Switches the system clock to the low-frequency clock.)
CLR      (SYSCR2).7      : SYSCR2<XEN> ← 0 (Stops the high-frequency clock.)
RETI
:       :
VINTTC4: DW      PINTTC4  : INTTC4 vector table
    
```

10.3.9.2 High-Frequency Warm-Up Counter Mode (SLOW1 → SLOW2 → NORMAL2 → NORMAL1)

In this mode, the warm-up period time from a stop of the high-frequency clock f_c to the oscillation stability is obtained. Before starting the timer, set SYSCR2<XEN> to 1 to oscillate the high-frequency clock. When a match between the up-counter and the timer register (TTREG4, 3) value is detected after the timer is started by setting TC4CR<TC4S> to 1, the counter is cleared by generating the INTTC4 interrupt request. After stopping the timer in the INTTC4 interrupt service routine, clear SYSCR2<SYSCK> to 0 to switch the system clock from the low-frequency to high-frequency, and then SYSCR2<XTEN> to 0 to stop the low-frequency clock.

Table 10-9 Setting Time in High-Frequency Warm-Up Counter Mode

| Minimum time Setting (TTREG4, 3 = 0100H) | Maximum time Setting (TTREG4, 3 = FF00H) |
|---|---|
| 16 μ s | 4.08 ms |

Example :After checking high-frequency clock oscillation stability with TC4 and 3, switching to the NORMAL1 mode

```

SET      (SYSCR2).7      : SYSCR2<XEN> ← 1
LD       (TC3CR), 63H    : Sets TFF3=0, source clock  $f_c$ , and 16-bit mode.
LD       (TC4CR), 05H    : Sets TFF4=0, and warm-up counter mode.
LD       (TTREG3), 0F800H : Sets the warm-up time.
                               (The warm-up time depends on the oscillator characteristic.)
DI       : IMF ← 0
SET      (EIRH). 1      : Enables the INTTC4.
EI       : IMF ← 1
SET      (TC4CR).3      : Starts the TC4 and 3.
:       :
PINTTC4: CLR      (TC4CR).3 : Stops the TC4 and 3.
        CLR      (SYSCR2).5 : SYSCR2<SYSCK> ← 0
                               (Switches the system clock to the high-frequency clock.)
        CLR      (SYSCR2).6 : SYSCR2<XTEN> ← 0
                               (Stops the low-frequency clock.)
        RETI
:       :
VINTTC4: DW       PINTTC4  : INTTC4 vector table
    
```


11.2 Control

The SIO is controlled using the serial interface control register (SIOCR1). The operating status of the serial interface can be inspected by reading the status register (SIOCR1).

Serial Interface Control Register

| | | | | | | | | | |
|-------------------|------|--------|------|--------|-----|---|---|---|----------------------------|
| SIOCR1 (0026H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | SIOS | SIOINH | SIOM | SIODIR | SCK | | | | (Initial value: 0000 0000) |

| | | | | | |
|--------|-------------------------------------|--|------------------------|------------------------|-------------------|
| SIOS | Specify start/stop of transfer | 0: Stop 1: Start | | | R/W |
| SIOINH | Forcibly stops transfer (Note 1) | 0: – 1: Forcibly stop (Automatically cleared to "0" after stopping) | | | |
| SIOM | Selects transfer mode | 00: Transmit mode 01: Receive mode 10: Transmit/receive mode 11: Reserved | | | |
| SIODIR | Selects direction of transfer | 0: MSB (Transfer beginning with bit7) 1: LSB (Transfer beginning with bit0) | | | |
| SCK | Selects serial clock | NORMAL 1/2 or IDLE 1/2 modes | | SLOW/SLEEP mode | |
| | | | TBTCR <DV7CK> = "0" | TBTCR <DV7CK> = "1" | |
| | | 000 | fc/2 ¹² | fs/2 ⁴ | fs/2 ⁴ |
| | | 001 | fc/2 ⁸ | fc/2 ⁸ | Reserved |
| | | 010 | fc/2 ⁷ | fc/2 ⁷ | Reserved |
| | | 011 | fc/2 ⁶ | fc/2 ⁶ | Reserved |
| | | 100 | fc/2 ⁵ | fc/2 ⁵ | Reserved |
| | | 101 | fc/2 ⁴ | fc/2 ⁴ | Reserved |
| 110 | fc/2 ³ | fc/2 ³ | Reserved | | |
| 111 | External clock (Input from SCK pin) | | | | |

Note 1: When SIOCR1<SIOINH> is set to "1", SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

Note 2: Transfer mode, direction of transfer and serial clock must be select during the transfer is stopping (when SIOSR<SIOF> "0").

Note 3: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Serial Interface Status Register

| | | | | | | | | | |
|------------------|------|-----|-----|-----|-------|-------|---|---|----------------------------|
| SIOSR (0027H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: 0010 00**) |
| | SIOF | SEF | TXF | RXF | TXERR | RXERR | | | |

| | | | |
|-------|--|--|-----------|
| SIOF | Serial transfer operation status monitor | 0: Transfer finished 1: Transfer in progress | Read only |
| SEF | Number of clocks monitor | 0: 8 clocks 1: 1 to 7 clocks | |
| TXF | Transmit buffer empty flag | 0: Data exists in transmit buffer 1: No data exists in transmit buffer | |
| RXF | Receive buffer full flag | 0: No data exists in receive buffer 1: Data exists in receive buffer | |
| TXERR | Transfer operation error flag | Read 0: – (No error exist) 1: Transmit buffer under run occurs in an external clock mode Write 0: Clear the flag 1: – (A write of "1" to this bit is ignored) | R/W |
| RXERR | Receive operation error flag | Read 0: – (No error exist) 1: Receive buffer over run occurs in an external clock mode Write 0: Clear the flag 1: – (A write of "1" to this bit is ignored) | |

Note 1: The operation error flag (TXERR and RXERR) are not automatically cleared by stopping transfer with SIOCR1<SIOS> "0". Therefore, set these bits to "0" for clearing these error flag. Or set SIOCR1<SIOINH> to "1".

Note 2: *: Don't care

Receive buffer register

| | | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|---|---|
| SIORDB (0028H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Read only (Initial value: 0000 0000) |
| | | | | | | | | | |

Transmit buffer register

| | | | | | | | | | |
|-------------------|---|---|---|---|---|---|---|---|---------------------------------------|
| SIOTDB (0028H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Write only (Initial value: **** *) |
| | | | | | | | | | |

Note 1: SIOTDB is write only register. A bit manipulation should not be performed on the transmit buffer register using a read-modify-write instruction.

Note 2: The SIOTDB should be written after checking SIOSR<TXF> "1". When SIOSR<TXF> is "0", the writing data can't be transferred to SIOTDB even if write instruction is executed to SIOTDB

Note 3: *: Don't care

11.3 Function

11.3.1 Serial clock

11.3.1.1 Clock source

The serial clock can be selected by using SIOCR1<SCK>. When the serial clock is changed, the writing instruction to SIOCR1<SCK> should be executed while the transfer is stopped (when SIOSR<SIOF> “0”)

(1) Internal clock

Setting the SIOCR1<SCK> to other than “111B” outputs the clock (shown in " Table 11-1 Serial Clock Rate ($f_c = 16\text{ MHz}$, $f_s = 32.768\text{kHz}$) ") as serial clock outputs from $\overline{\text{SCK}}$ pin. At the before beginning or finishing of a transfer, $\overline{\text{SCK}}$ pin is kept in high level.

When writing (in the transmit mode) or reading (in the receive mode) data can not follow the serial clock rate, an automatic-wait function is executed to stop the serial clock automatically and hold the next shift operation until reading or writing is completed (shown in " Figure 11-2 Automatic-wait Function (Example of transmit mode) "). The maximum time from releasing the automatic-wait function by reading or writing a data is 1 cycle of the selected serial clock until the serial clock comes out from $\overline{\text{SCK}}$ pin.

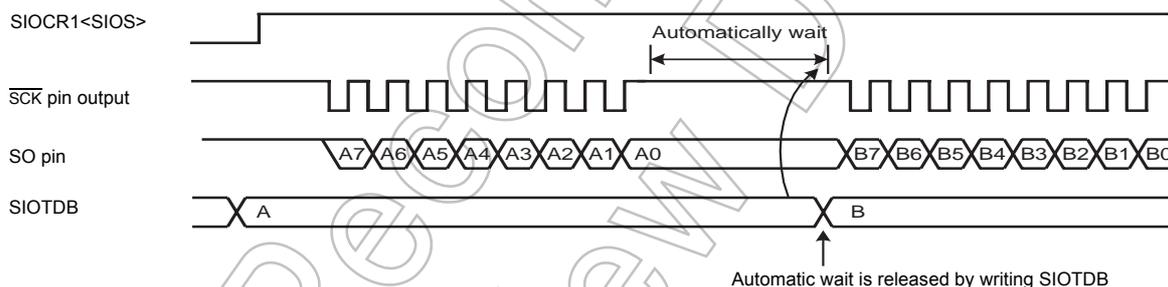


Figure 11-2 Automatic-wait Function (Example of transmit mode)

Table 11-1 Serial Clock Rate ($f_c = 16\text{ MHz}$, $f_s = 32.768\text{kHz}$)

| SCK | NORMAL1/2, IDLE1/2 Mode | | | | SLOW1/2, SLEEP1/2 Mode | |
|-----|-------------------------|------------|--------------|-----------|------------------------|-----------|
| | Serial Clock | Baud Rate | Serial Clock | Baud Rate | Serial Clock | Baud Rate |
| 000 | $f_c/2^{12}$ | 3.906 kbps | $f_s/2^4$ | 2048 bps | $f_s/2^4$ | 2048 bps |
| 001 | $f_c/2^8$ | 62.5 kbps | $f_c/2^8$ | 62.5 kbps | Reserved | — |
| 010 | $f_c/2^7$ | 125 kbps | $f_c/2^7$ | 125 kbps | Reserved | — |
| 011 | $f_c/2^6$ | 250 kbps | $f_c/2^6$ | 250 kbps | Reserved | — |
| 100 | $f_c/2^5$ | 500 kbps | $f_c/2^5$ | 500 kbps | Reserved | — |
| 101 | $f_c/2^4$ | 1.00 Mbps | $f_c/2^4$ | 1.00 Mbps | Reserved | — |
| 110 | $f_c/2^3$ | 2.00 Mbps | $f_c/2^3$ | 2.00 Mbps | Reserved | — |

(2) External clock

When an external clock is selected by setting SIOCR1<SCK> to “111B”, the clock via the $\overline{\text{SCK}}$ pin from an external source is used as the serial clock.

To ensure shift operation, the serial clock pulse width must be $4/f_c$ or more for both “H” and “L” levels.

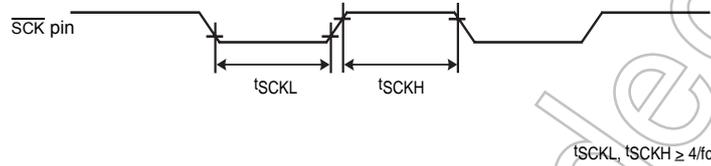


Figure 11-3 External Clock

11.3.1.2 Shift edge

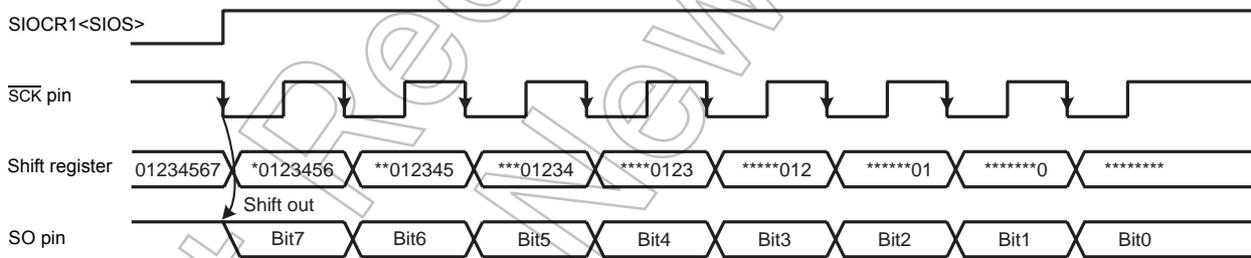
The leading edge is used to transmit data, and the trailing edge is used to receive data.

(1) Leading edge shift

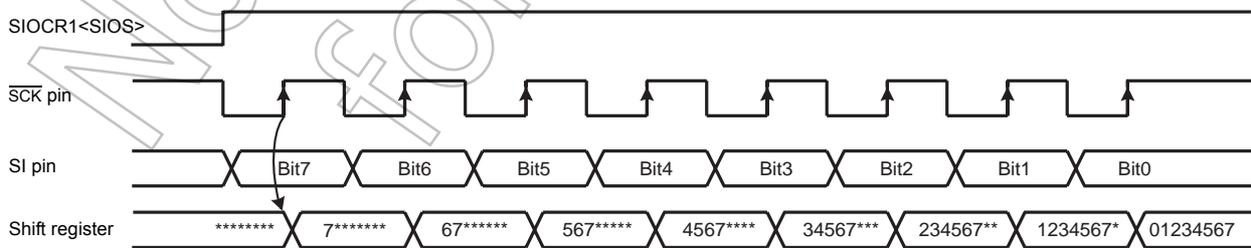
Data is shifted on the leading edge of the serial clock (falling edge of the $\overline{\text{SCK}}$ pin input/output).

(2) Trailing edge shift

Data is shifted on the trailing edge of the serial clock (rising edge of the $\overline{\text{SCK}}$ pin input/output).



(a) Leading edge shift (Example of MSB transfer)



(b) Trailing edge shift (Example of MSB transfer)

Figure 11-4 Shift Edge

11.3.2 Transfer bit direction

Transfer data direction can be selected by using SIOCR1<SIODIR>. The transfer data direction can't be set individually for transmit and receive operations.

When the data direction is changed, the writing instruction to SIOCR1<SIODIR> should be executed while the transfer is stopped (when SIOCR1<SIOF>= "0")

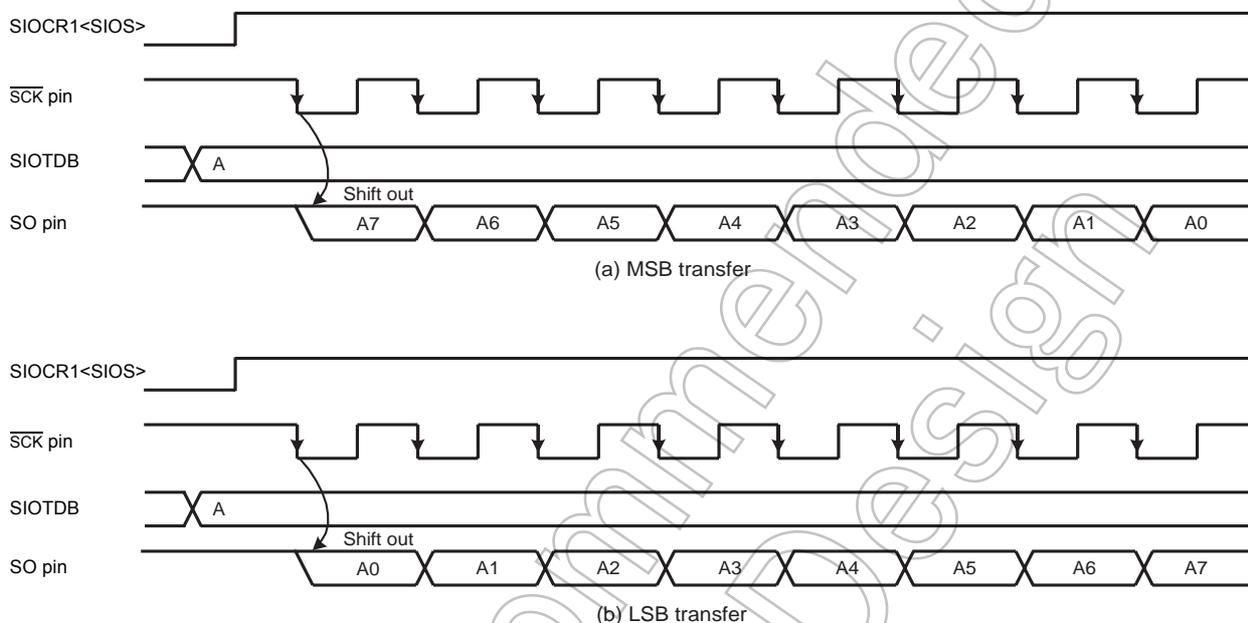


Figure 11-5 Transfer Bit Direction (Example of transmit mode)

11.3.2.1 Transmit mode

(1) MSB transmit mode

MSB transmit mode is selected by setting SIOCR1<SIODIR> to "0", in which case the data is transferred sequentially beginning with the most significant bit (Bit7).

(2) LSB transmit mode

LSB transmit mode is selected by setting SIOCR1<SIODIR> to "1", in which case the data is transferred sequentially beginning with the least significant bit (Bit0).

11.3.2.2 Receive mode

(1) MSB receive mode

MSB receive mode is selected by setting SIOCR1<SIODIR> to "0", in which case the data is received sequentially beginning with the most significant bit (Bit7).

(2) LSB receive mode

LSB receive mode is selected by setting SIOCR1<SIODIR> to “1”, in which case the data is received sequentially beginning with the least significant bit (Bit0).

11.3.2.3 Transmit/receive mode

(1) MSB transmit/receive mode

MSB transmit/receive mode are selected by setting SIOCR1<SIODIR> to “0” in which case the data is transferred sequentially beginning with the most significant bit (Bit7) and the data is received sequentially beginning with the most significant (Bit7).

(2) LSB transmit/receive mode

LSB transmit/receive mode are selected by setting SIOCR1<SIODIR> to “1”, in which case the data is transferred sequentially beginning with the least significant bit (Bit0) and the data is received sequentially beginning with the least significant (Bit0).

11.3.3 Transfer modes

Transmit, receive and transmit/receive mode are selected by using SIOCR1<SIOM>.

11.3.3.1 Transmit mode

Transmit mode is selected by writing “00B” to SIOCR1<SIOM>.

(1) Starting the transmit operation

Transmit mode is selected by setting “00B” to SIOCR1<SIOM>. Serial clock is selected by using SIOCR1<SCK>. Transfer direction is selected by using SIOCR1<SIODIR>.

When a transmit data is written to the transmit buffer register (SIOTDB), SIOSR<TXF> is cleared to “0”.

After SIOCR1<SIOS> is set to “1”, SIOSR<SIOF> is set synchronously to “1” the falling edge of $\overline{\text{SCK}}$ pin.

The data is transferred sequentially starting from SO pin with the direction of the bit specified by SIOCR1<SIODIR>, synchronizing with the $\overline{\text{SCK}}$ pin's falling edge.

SIOSR<SEF> is kept in high level, between the first clock falling edge of $\overline{\text{SCK}}$ pin and eighth clock falling edge.

SIOSR<TXF> is set to “1” at the rising edge of pin after the data written to the SIOTDB is transferred to shift register, then the INTSIO interrupt request is generated, synchronizing with the next falling edge on $\overline{\text{SCK}}$ pin.

Note 1: In internal clock operation, when SIOCR1<SIOS> is set to “1”, transfer mode does not start without writing a transmit data to the transmit buffer register (SIOTDB).

Note 2: In internal clock operation, when the SIOCR1<SIOS> is set to “1”, SIOTDB is transferred to shift register after maximum 1-cycle of serial clock frequency, then a serial clock is output from $\overline{\text{SCK}}$ pin.

Note 3: In external clock operation, when the falling edge is input from $\overline{\text{SCK}}$ pin after SIOCR1<SIOS> is set to “1”, SIOTDB is transferred to shift register immediately.

(2) During the transmit operation

When data is written to SIOTDB, SIOSR<TXF> is cleared to “0”.

In internal clock operation, in case a next transmit data is not written to SIOTDB, the serial clock stops to “H” level by an automatic-wait function when all of the bit set in the SIOTDB has been transmitted. Automatic-wait function is released by writing a transmit data to SIOTDB. Then, transmit operation is restarted after maximum 1-cycle of serial clock.

When the next data is written to the SIOTDB before termination of previous 8-bit data with SIOSR<TXF> “1”, the next data is continuously transferred after transmission of previous data.

In external clock operation, after SIOSR<TXF> is set to “1”, the transmit data must be written to SIOTDB before the shift operation of the next data begins.

If the transmit data is not written to SIOTDB, transmit error occurs immediately after shift operation is started. Then, INTSIO interrupt request is generated after SIOSR<TXERR> is set to “1”.

(3) Stopping the transmit operation

There are two ways for stopping transmits operation.

- The way of clearing SIOCR1<SIOS>. When SIOCR1<SIOS> is cleared to “0”, transmit operation is stopped after all transfer of the data is finished. When transmit operation is finished, SIOSR<SIOF> is cleared to “0” and SO pin is kept in high level. In external clock operation, SIOCR1<SIOS> must be cleared to “0” before SIOSR<SEF> is set to “1” by beginning next transfer.
- The way of setting SIOCR1<SIOINH>. Transmit operation is stopped immediately after SIOCR1<SIOINH> is set to “1”. In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

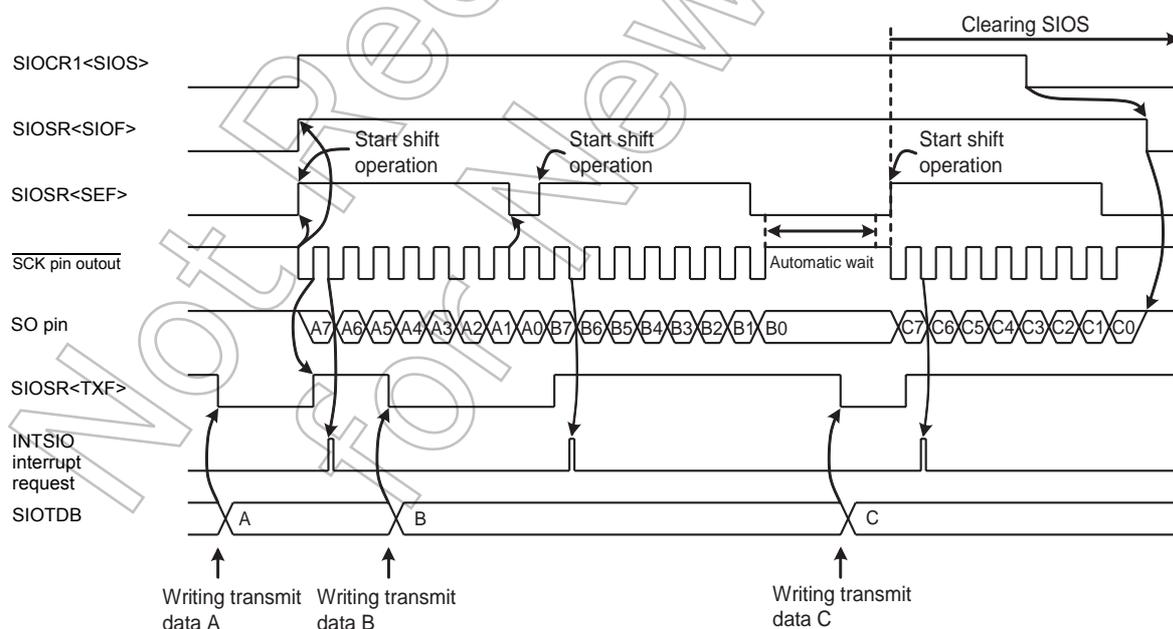


Figure 11-6 Example of Internal Clock and MSB Transmit Mode

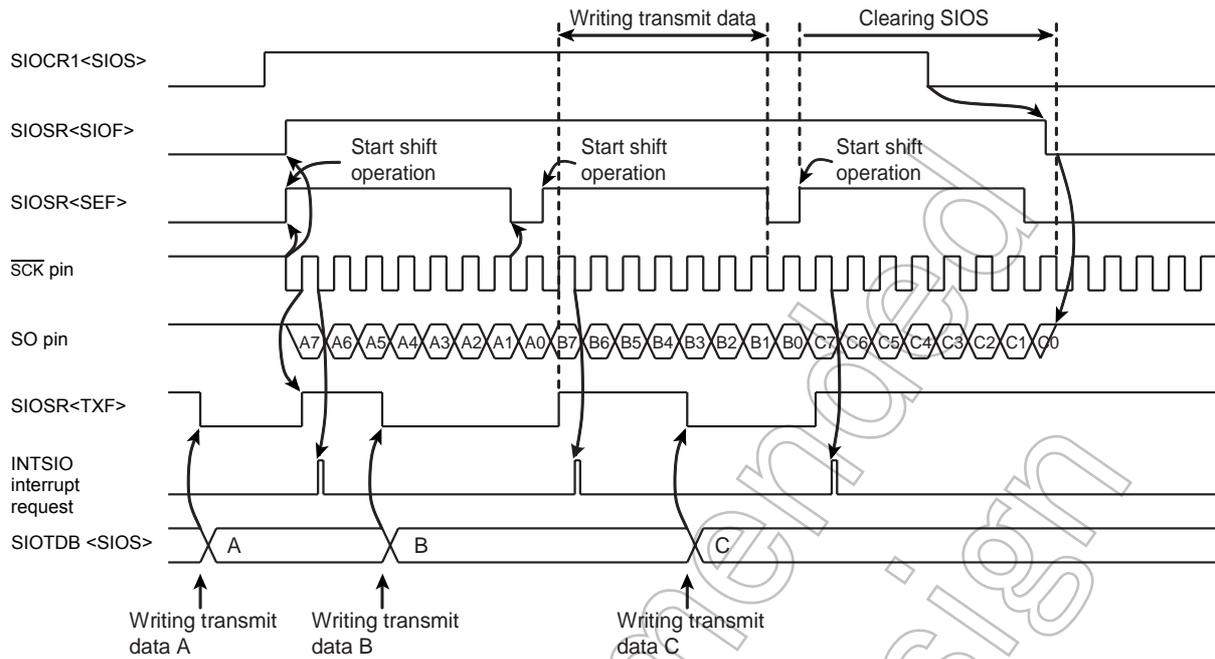


Figure 11-7 Exaple of External Clock and MSB Transmit Mode

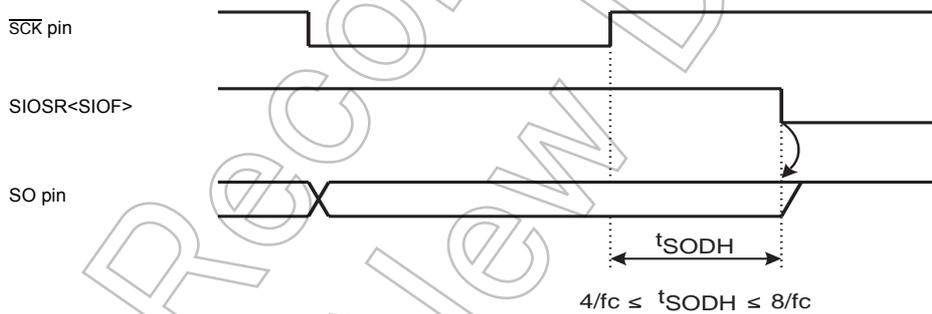


Figure 11-8 Hold Time of the End of Transmit Mode

(4) Transmit error processing

Transmit errors occur on the following situation.

- Shift operation starts before writing next transmit data to SIOTDB in external clock operation.

If transmit errors occur during transmit operation, SIOSR<TXERR> is set to “1” immediately after starting shift operation. Synchronizing with the next serial clock falling edge, INTSIO interrupt request is generated.

If shift operation starts before writing data to SIOTDB after SIOCR1<SIOS> is set to “1”, SIOSR<TXERR> is set to “1” immediately after shift operation is started and then INTSIO interrupt request is generated.

SIO pin is kept in high level when SIOSR<TXERR> is set to “1”. When transmit error occurs, transmit operation must be forcibly stop by writing SIOCR1<SIOINH> to “1”. In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

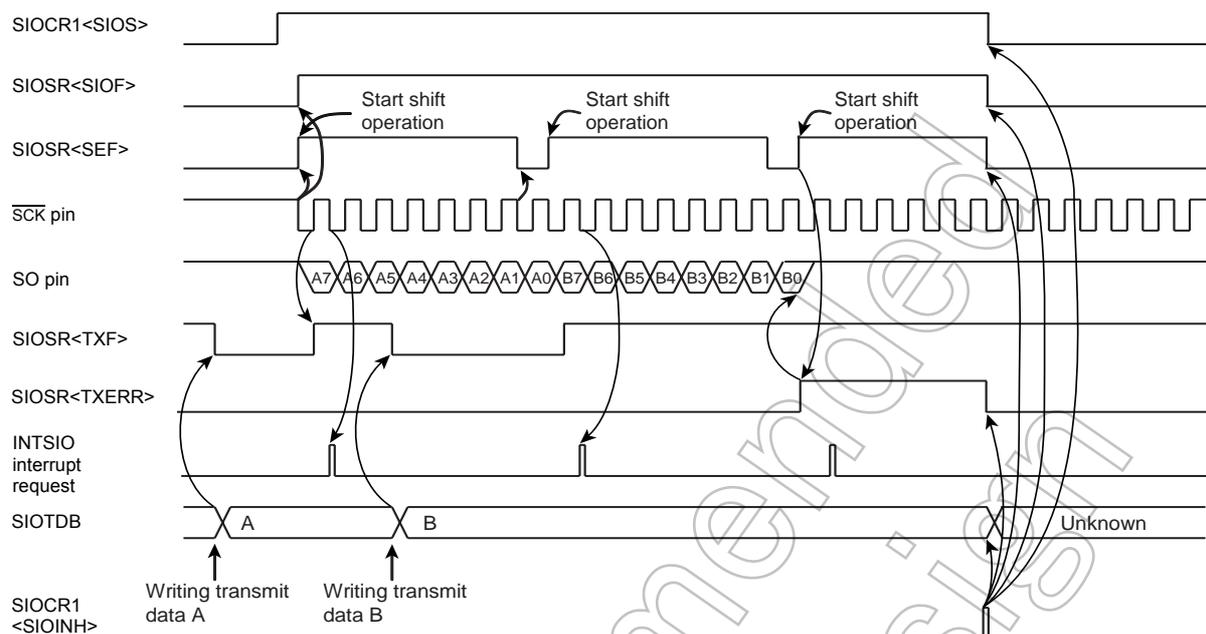


Figure 11-9 Example of Transmit Error Processing

11.3.3.2 Receive mode

The receive mode is selected by writing “01B” to SIOCR1<SIOM>.

(1) Starting the receive operation

Receive mode is selected by setting “01” to SIOCR1<SIOM>. Serial clock is selected by using SIOCR1<SCK>. Transfer direction is selected by using SIOCR1<SIODIR>.

After SIOCR1<SIOS> is set to “1”, SIOSR<SIOF> is set synchronously to “1” the falling edge of $\overline{\text{SCK}}$ pin.

Synchronizing with the $\overline{\text{SCK}}$ pin's rising edge, the data is received sequentially from SI pin with the direction of the bit specified by SBIDIR<SIODIR>.

SIOSR<SEF> is kept in high level, between the first clock falling edge of $\overline{\text{SCK}}$ pin and eighth clock falling edge.

When 8-bit data is received, the data is transferred to SIORDB from shift register. INTSIO interrupt request is generated and SIOSR<RXF> is set to “1”

Note: In internal clock operation, when the SIOCR1<SIOS> is set to “1”, the serial clock is generated from $\overline{\text{SCK}}$ pin after maximum 1-cycle of serial clock frequency.

(2) During the receive operation

The SIOSR<RXF> is cleared to “0” by reading a data from SIORDB.

In the internal clock operation, the serial clock stops to “H” level by an automatic-wait function when the all of the 8-bit data has been received. Automatic-wait function is released by reading a received data from SIORDB. Then, receive operation is restarted after maximum 1-cycle of serial clock.

In external clock operation, after SIOSR<RXF> is set to “1”, the received data must be read from SIORDB, before the next data shift-in operation is finished.

If received data is not read out from SIORDB receive error occurs immediately after shift operation is finished. Then INTSIO interrupt request is generated after SIOSR<RXERR> is set to "1".

(3) Stopping the receive operation

There are two ways for stopping the receive operation.

- The way of clearing SIOCR1<SIOS>.

When SIOCR1<SIOS> is cleared to "0", receive operation is stopped after all of the data is finished to receive. When receive operation is finished, SIOSR<SIOF> is cleared to "0". In external clock operation, SIOCR1<SIOS> must be cleared to "0" before SIOSR<SEF> is set to "1" by starting the next shift operation.
- The way of setting SIOCR1<SIOINH>.

Receive operation is stopped immediately after SIOCR1<SIOINH> is set to "1". In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

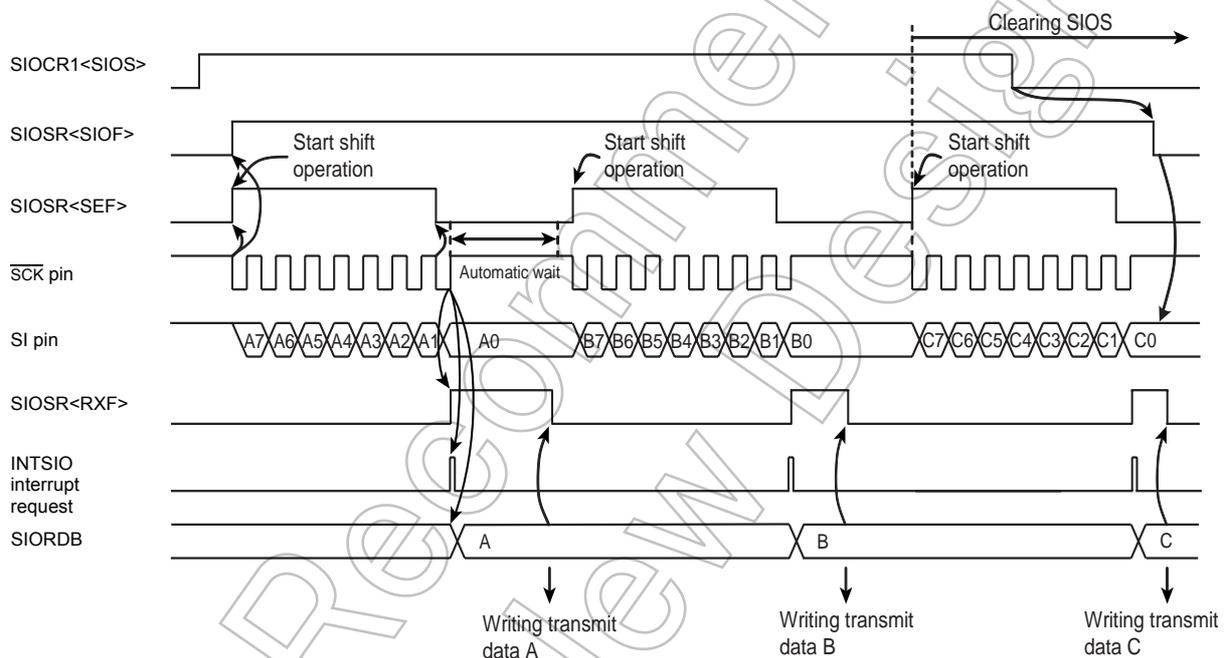


Figure 11-10 Example of Internal Clock and MSB Receive Mode

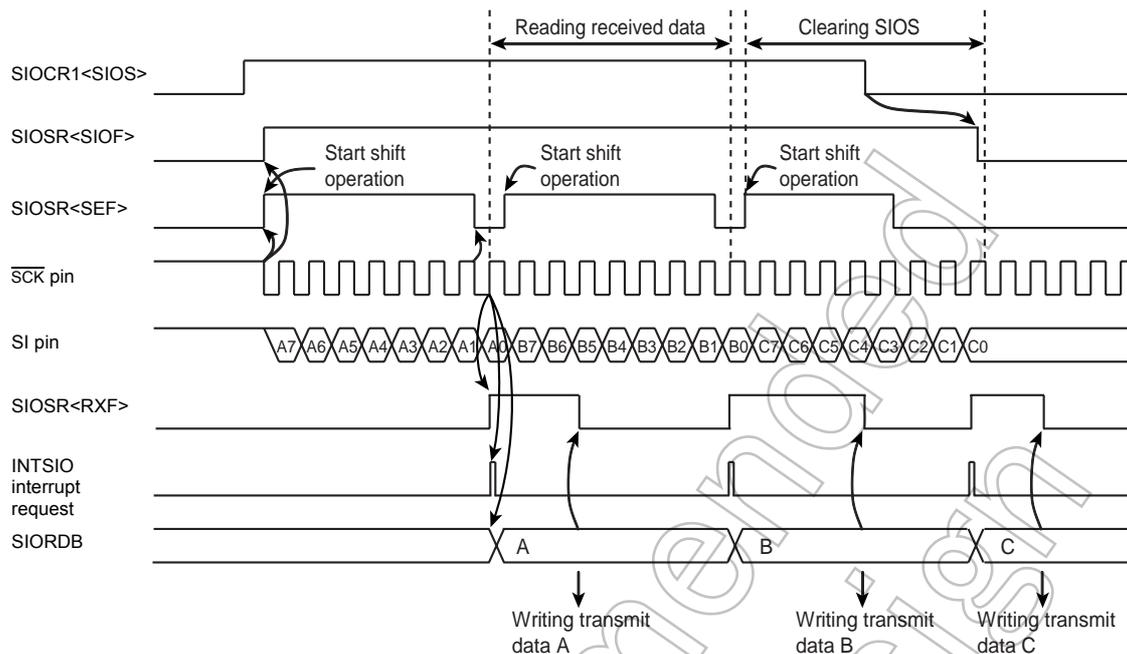


Figure 11-11 Example of External Clock and MSB Receive Mode

(4) Receive error processing

Receive errors occur on the following situation. To protect SIORDB and the shift register contents, the received data is ignored while the SIOSR<RXERR> is “1”.

- Shift operation is finished before reading out received data from SIORDB at SIOSR<RXF> is “1” in an external clock operation.

If receive error occurs, set the SIOCR1<SIOS> to “0” for reading the data that received immediately before error occurrence. And read the data from SIORDB. Data in shift register (at errors occur) can be read by reading the SIORDB again.

When SIOSR<RXERR> is cleared to “0” after reading the received data, SIOSR<RXF> is cleared to “0”.

After clearing SIOCR1<SIOS> to “0”, when 8-bit serial clock is input to $\overline{\text{SCK}}$ pin, receive operation is stopped. To restart the receive operation, confirm that SIOSR<SIOF> is cleared to “0”.

If the receive error occurs, set the SIOCR1<SIOINH> to “1” for stopping the receive operation immediately. In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

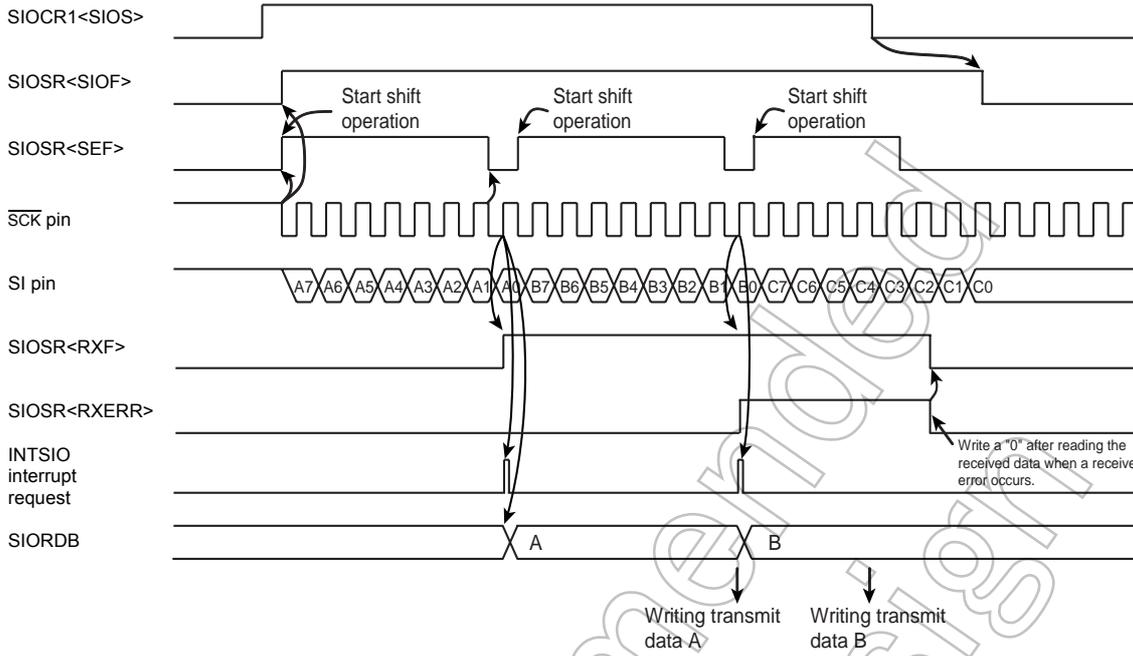


Figure 11-12 Example of Receive Error Processing

Note: If receive error is not corrected, an interrupt request does not generate after the error occurs.

11.3.3.3 Transmit/receive mode

The transmit/receive mode are selected by writing “10” to SIOCR1<SIOM>.

(1) Starting the transmit/receive operation

Transmit/receive mode is selected by writing “10B” to SIOCR1<SIOM>. Serial clock is selected by using SIOCR1<SCK>. Transfer direction is selected by using SIOCR1<SIODIR>.

When a transmit data is written to the transmit buffer register (SIOTDB), SIOSR<TXF> is cleared to “0”.

After SIOCR1<SIOS> is set to “1”, SIOSR<SIOF> is set synchronously to the falling edge of $\overline{\text{SCK}}$ pin.

The data is transferred sequentially starting from SO pin with the direction of the bit specified by SIOCR1<SIODIR>, synchronizing with the $\overline{\text{SCK}}$ pin's falling edge. And receiving operation also starts with the direction of the bit specified by SIOCR1<SIODIR>, synchronizing with the $\overline{\text{SCK}}$ pin's rising edge.

SIOSR<SEF> is kept in high level between the first clock falling edge of $\overline{\text{SCK}}$ pin and eighth clock falling edge.

SIOSR<TXF> is set to “1” at the rising edge of $\overline{\text{SCK}}$ pin after the data written to the SIOTDB is transferred to shift register. When 8-bit data has been received, the received data is transferred to SIORDB from shift register, then the INTSIO interrupt request occurs, synchronizing with setting SIOSR<RXF> to “1”.

Note 1: In internal clock operation, when the SIOCR1<SIOS> is set to “1”, SIOTDB is transferred to shift register after maximum 1-cycle of serial clock frequency, then a serial clock is output from $\overline{\text{SCK}}$ pin.

Note 2: In external clock operation, when the falling edge is input from $\overline{\text{SCK}}$ pin after SIOCR1<SIOS> is set to “1”, SIOTDB is transferred to shift register immediately. When the rising edge is input from $\overline{\text{SCK}}$ pin, receive operation also starts.

(2) During the transmit/receive operation

When data is written to SIOTDB, SIOSR<TXF> is cleared to “0” and when a data is read from SIORDB, SIOSR<RXF> is cleared to “0”.

In internal clock operation, in case of the condition described below, the serial clock stops to “H” level by an automatic-wait function when all of the bit set in the data has been transmitted.

- Next transmit data is not written to SIOTDB after reading a received data from SIORDB.
- Received data is not read from SIORDB after writing a next transmit data to SIOTDB.
- Neither SIOTDB nor SIORDB is accessed after transmission.

The automatic wait function is released by writing the next transmit data to SIOTDB after reading the received data from SIORDB, or reading the received data from SIORDB after writing the next data to SIOTDB.

Then, transmit/receive operation is restarted after maximum 1 cycle of serial clock.

In external clock operation, reading the received data from SIORDB and writing the next data to SIOTDB must be finished before the shift operation of the next data begins.

If the transmit data is not written to SIOTDB after SIOSR<TXF> is set to “1”, transmit error occurs immediately after shift operation is started. When the transmit error occurred, SIOSR<TXERR> is set to “1”.

If received data is not read out from SIORDB before next shift operation starts after setting SIOSR<RXF> to “1”, receive error occurs immediately after shift operation is finished. When the receive error has occurred, SIOSR<RXERR> is set to “1”.

(3) Stopping the transmit/receive operation

There are two ways for stopping the transmit/receive operation.

- The way of clearing SIOCR1<SIOS>.
When SIOCR1<SIOS> is cleared to “0”, transmit/receive operation is stopped after all transfer of the data is finished. When transmit/receive operation is finished, SIOSR<SIOF> is cleared to “0” and SO pin is kept in high level.
In external clock operation, SIOCR1<SIOS> must be cleared to “0” before SIOSR<SEF> is set to “1” by beginning next transfer.
- The way of setting SIOCR1<SIOINH>.
Transmit/receive operation is stopped immediately after SIOCR1<SIOINH> is set to “1”. In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

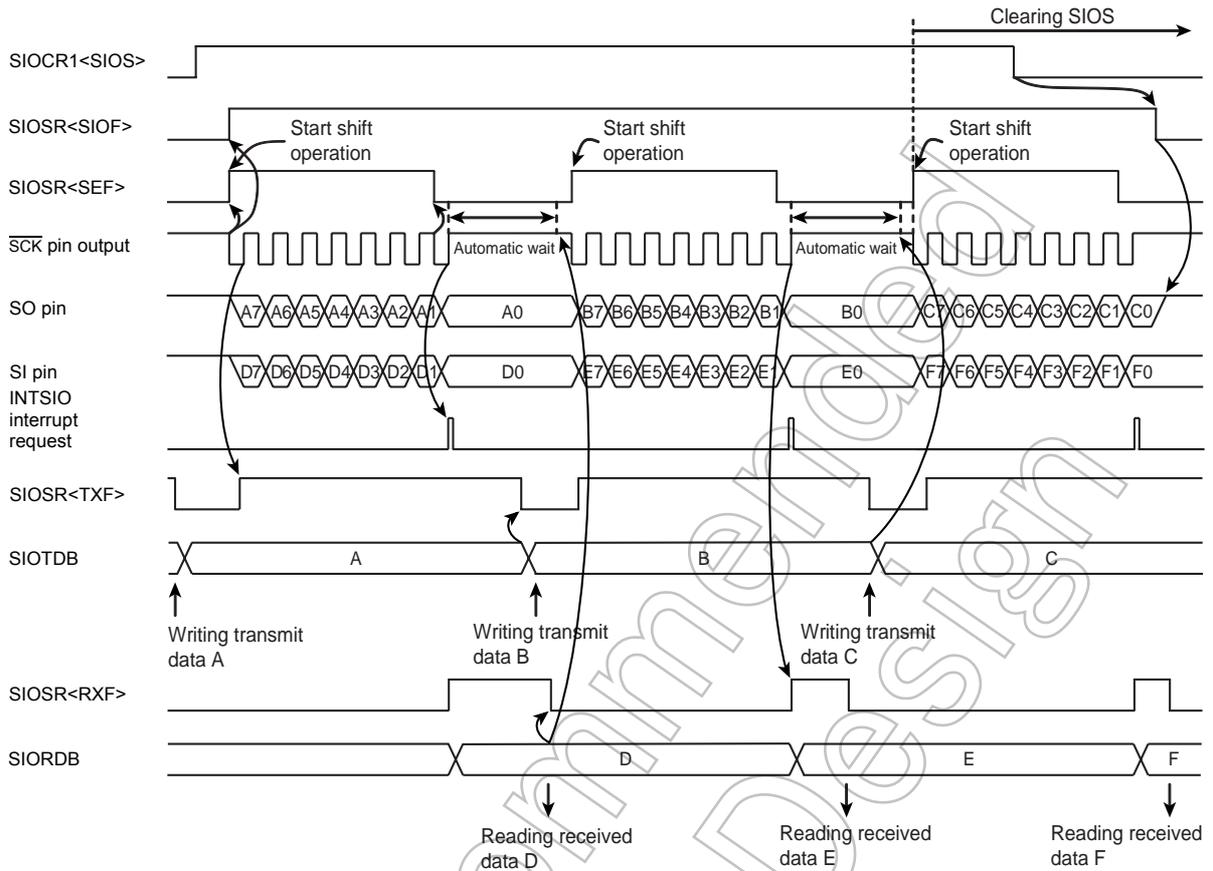


Figure 11-13 Example of Internal Clock and MSB Transmit/Receive Mode

Not Recommended for New

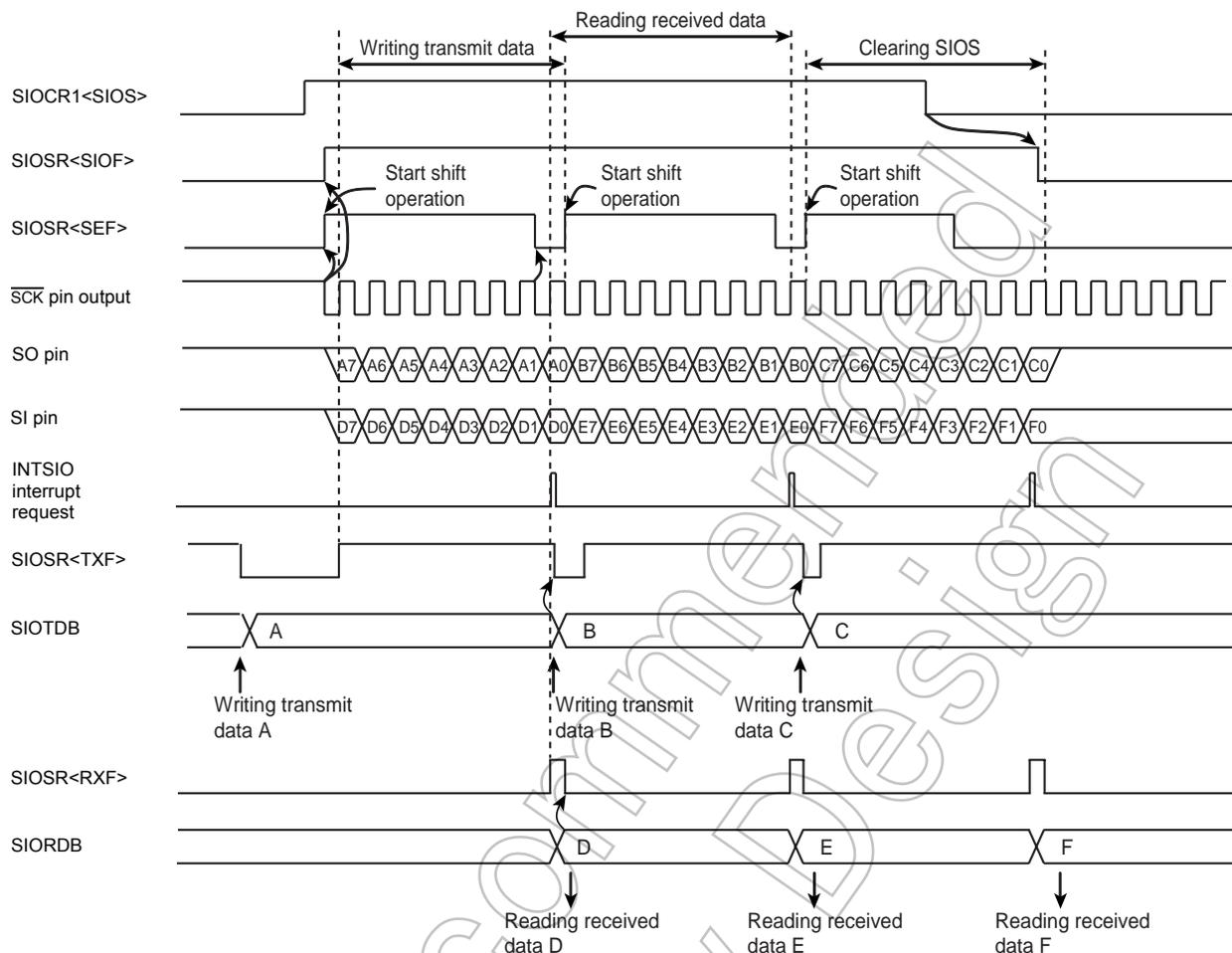


Figure 11-14 Example of External Clock and MSB Transmit/Receive Mode

(4) Transmit/receive error processing

Transmit/receive errors occur on the following situation. Corrective action is different, which errors occur transmits or receives.

(a) Transmit errors

Transmit errors occur on the following situation.

- Shift operation starts before writing next transmit data to SIOTDB in external clock operation.

If transmit errors occur during transmit operation, SIOSR<TXERR> is set to “1” immediately after starting shift operation. And INTSIO interrupt request is generated after all of the 8-bit data has been received.

If shift operation starts before writing data to SIOTDB after SIOCR1<SIOS> is set to “1”, SIOSR<TXERR> is set immediately after starting shift operation. And INTSIO interrupt request is generated after all of the 8-bit data has been received.

SO pin is kept in high level when SIOSR<TXERR> is set to “1”. When transmit error occurs, transmit operation must be forcibly stop by writing SIOCR1<SIOINH> to “1” after the received data is read from SIORDB. In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

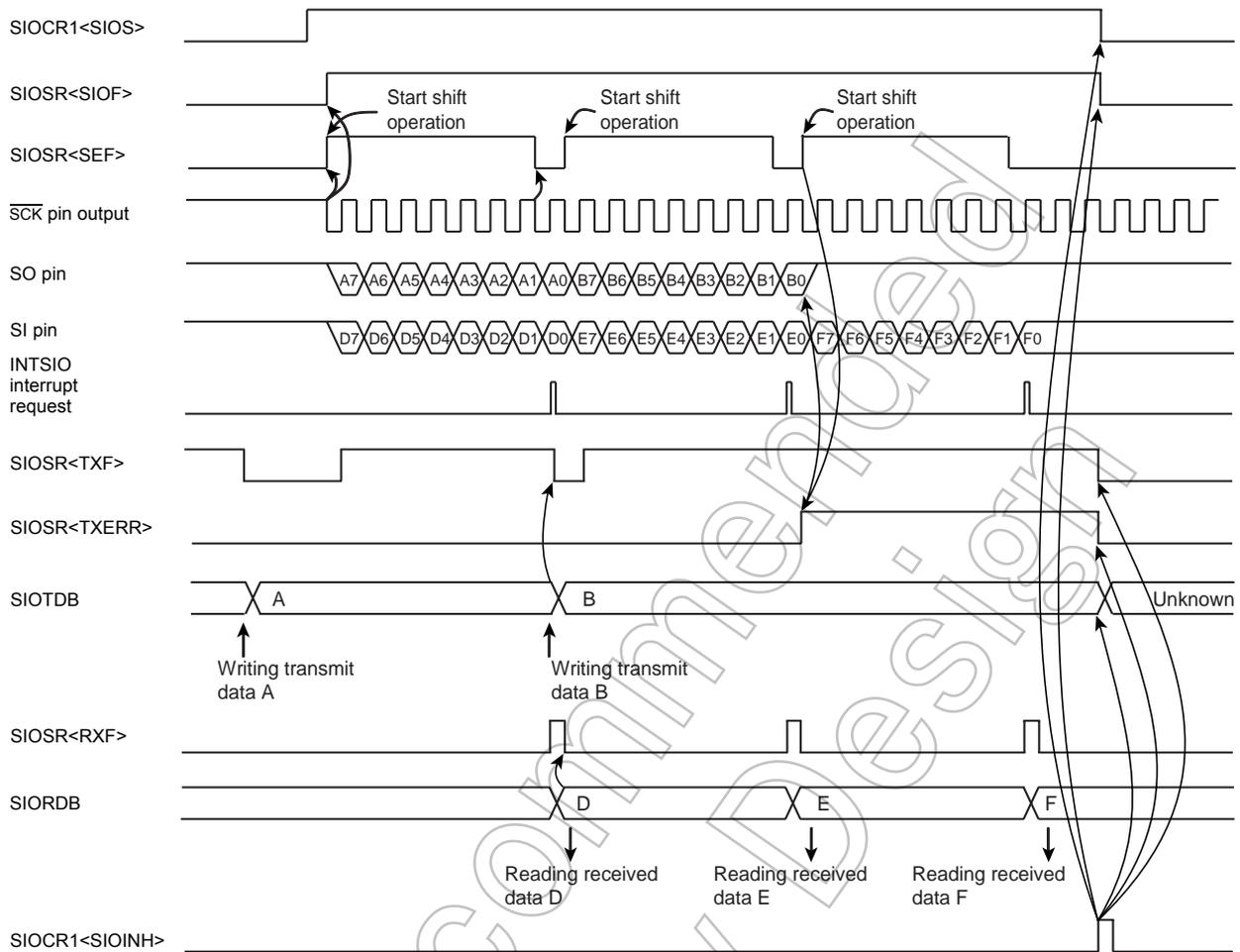


Figure 11-15 Example of Transmit/Receive (Transmit) Error Processing

(b) Receive errors

Receive errors occur on the following situation. To protect SIORDB and the shift register contents, the received data is ignored while the SIOSR<RXERR> is "1".

- Shift operation is finished before reading out received data from SIORDB at SIOSR<RXF> is "1" in an external clock operation. If receive error occurs, set the SIOCR1<SIOS> to "0" for reading the data that received immediately before error occurrence. And read the data from SIORDB. Data in shift register (at errors occur) can be read by reading the SIORDB again. When SIOSR<RXERR> is cleared to "0" after reading the received data, SIOSR<RXF> is cleared to "0". After clearing SIOCR1<SIOS> to "0", when 8-bit serial clock is input to $\overline{\text{SCK}}$ pin, receive operation is stopped. To restart the receive operation, confirm that SIOSR<SIOF> is cleared to "0". If the received error occurs, set the SIOCR1<SIOINH> to "1" for stopping the receive operation immediately. In this case, SIOCR1<SIOS>, SIOSR register, SIORDB register and SIOTDB register are initialized.

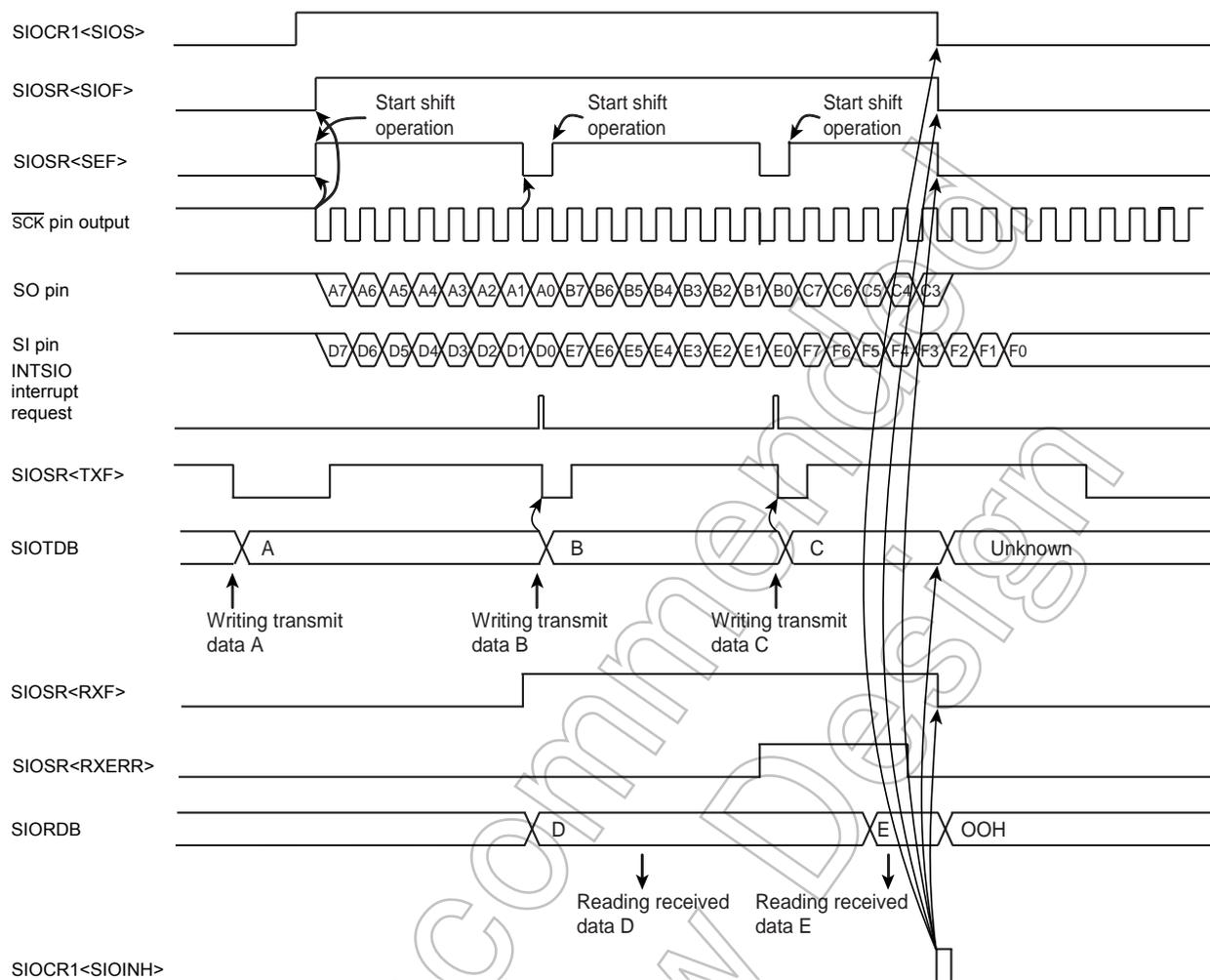


Figure 11-16 Example of Transmit/Receive (Receive) Error Processing

Note: If receive error is not corrected, an interrupt request does not generate after the error occurs.

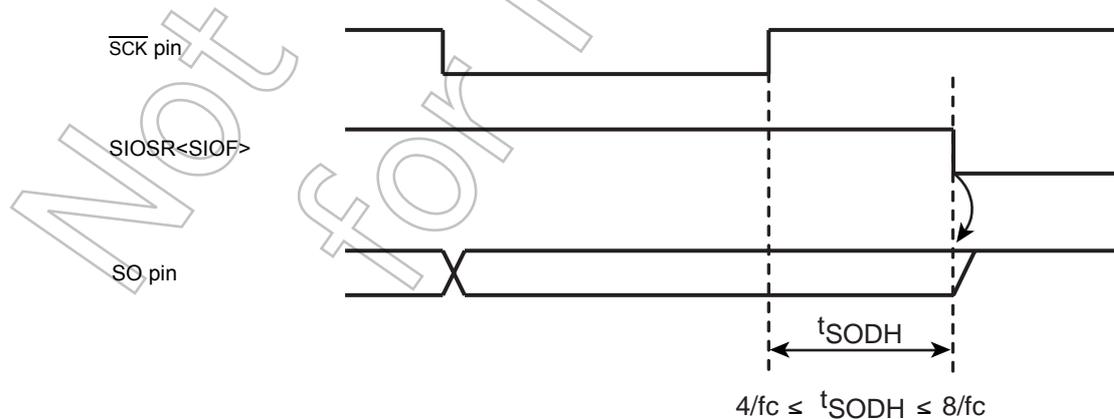


Figure 11-17 Hold Time of the End of Transmit/Receive Mode

12. Asynchronous Serial interface (UART)

12.1 Configuration

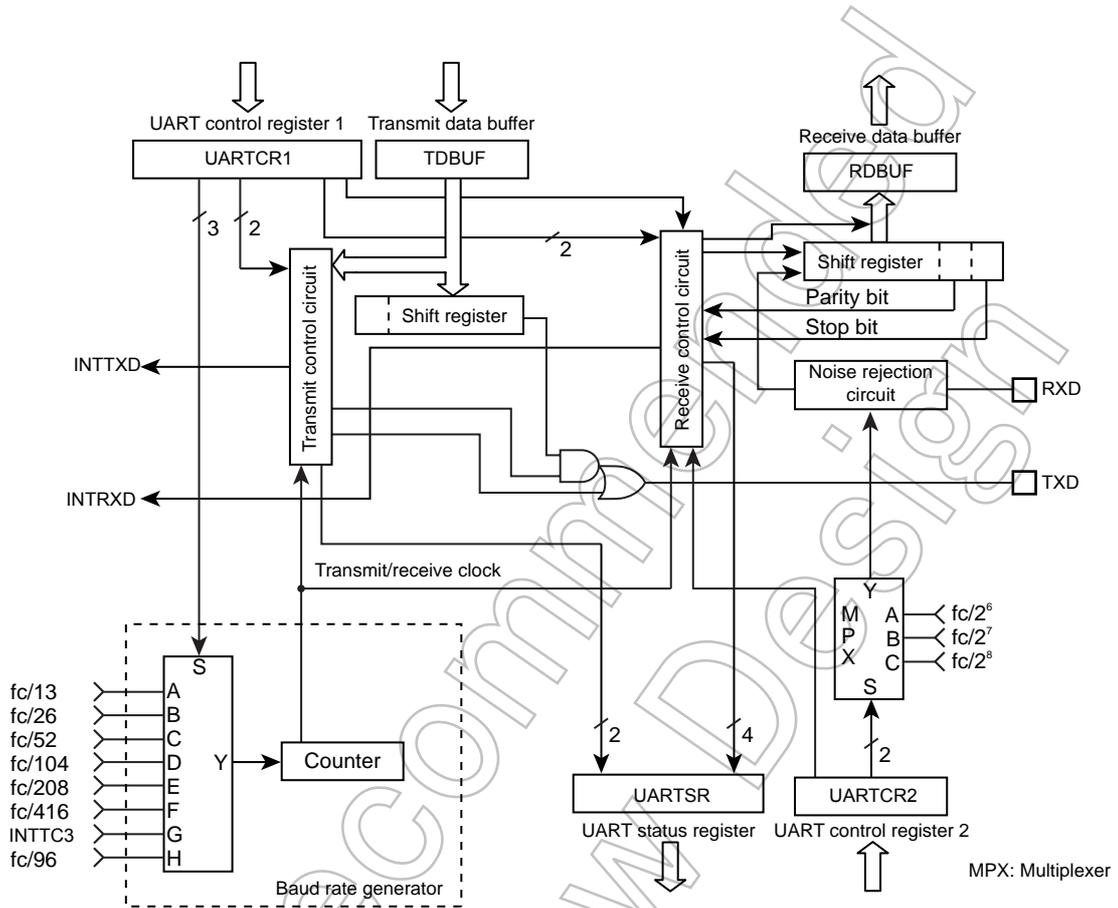


Figure 12-1 UART (Asynchronous Serial Interface)

12.2 Control

UART is controlled by the UART Control Registers (UARTCR1, UARTCR2). The operating status can be monitored using the UART status register (UARTSR).

UART Control Register1

| | | | | | | | | | |
|--------------------|-----|-----|------|------|----|-----|---|---|----------------------------|
| UARTCR1 (0020H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | TXE | RXE | STBT | EVEN | PE | BRG | | | (Initial value: 0000 0000) |

| | | | |
|------|--------------------------|--|------------|
| TXE | Transfer operation | 0: Disable 1: Enable | Write only |
| RXE | Receive operation | 0: Disable 1: Enable | |
| STBT | Transmit stop bit length | 0: 1 bit 1: 2 bits | |
| EVEN | Even-numbered parity | 0: Odd-numbered parity 1: Even-numbered parity | |
| PE | Parity addition | 0: No parity 1: Parity | |
| BRG | Transmit clock select | 000: fc/13 [Hz] 001: fc/26 010: fc/52 011: fc/104 100: fc/208 101: fc/416 110: TC3 (Input INTTC3) 111: fc/96 | |

Note 1: When operations are disabled by setting TXE and RXE bit to "0", the setting becomes valid when data transmit or receive complete. When the transmit data is stored in the transmit data buffer, the data are not transmitted. Even if data transmit is enabled, until new data are written to the transmit data buffer, the current data are not transmitted.

Note 2: The transmit clock and the parity are common to transmit and receive.

Note 3: UARTCR1<RXE> and UARTCR1<TXE> should be set to "0" before UARTCR1<BRG> is changed.

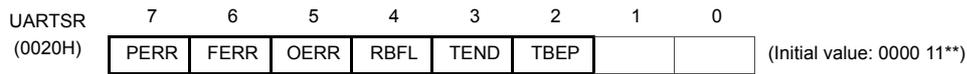
UART Control Register2

| | | | | | | | | | |
|--------------------|---|---|---|---|---|-------|--------|---|----------------------------|
| UARTCR2 (0021H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | | | | | | RXDNC | STOPBR | | (Initial value: **** *000) |

| | | | |
|--------|---|--|------------|
| RXDNC | Selection of RXD input noise rejection time | 00: No noise rejection (Hysteresis input) 01: Rejects pulses shorter than 31/fc [s] as noise 10: Rejects pulses shorter than 63/fc [s] as noise 11: Rejects pulses shorter than 127/fc [s] as noise | Write only |
| STOPBR | Receive stop bit length | 0: 1 bit 1: 2 bits | |

Note: When UARTCR2<RXDNC> = "01", pulses longer than 96/fc [s] are always regarded as signals; when UARTCR2<RXDNC> = "10", longer than 192/fc [s]; and when UARTCR2<RXDNC> = "11", longer than 384/fc [s].

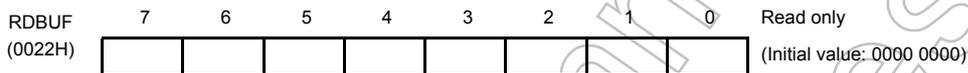
UART Status Register



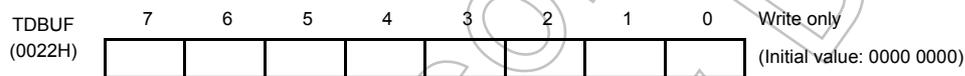
| | | | |
|------|---------------------------------|---|-----------|
| PERR | Parity error flag | 0: No parity error 1: Parity error | Read only |
| FERR | Framing error flag | 0: No framing error 1: Framing error | |
| OERR | Overrun error flag | 0: No overrun error 1: Overrun error | |
| RBFL | Receive data buffer full flag | 0: Receive data buffer empty 1: Receive data buffer full | |
| TEND | Transmit end flag | 0: On transmitting 1: Transmit end | |
| TBEP | Transmit data buffer empty flag | 0: Transmit data buffer full (Transmit data writing is finished) 1: Transmit data buffer empty | |

Note: When an INTTXD is generated, TBEP flag is set to "1" automatically.

UART Receive Data Buffer



UART Transmit Data Buffer



Not Recommended for New Design

12.3 Transfer Data Format

In UART, an one-bit start bit (Low level), stop bit (Bit length selectable at high level, by UARTCR1<STBT>), and parity (Select parity in UARTCR1<PE>; even- or odd-numbered parity by UARTCR1<EVEN>) are added to the transfer data. The transfer data formats are shown as follows.

| PE | STBT | Frame Length | | | | | | | | | |
|----|------|--------------|---|---|---|---|---|---|---|---|----|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 0 | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | |
| 1 | 1 | | | | | | | | | | |

Figure 12-2 Transfer Data Format

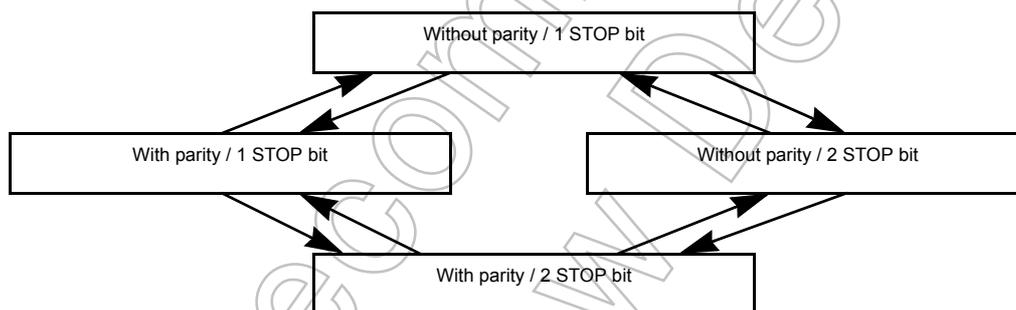


Figure 12-3 Caution on Changing Transfer Data Format

Note: In order to switch the transfer data format, perform transmit operations in the above Figure 12-3 sequence except for the initial setting.

12.4 Transfer Rate

The baud rate of UART is set of UARTCR1<BRG>. The example of the baud rate are shown as follows.

Table 12-1 Transfer Rate (Example)

| BRG | Source Clock | | |
|-----|--------------|--------------|--------------|
| | 16 MHz | 8 MHz | 4 MHz |
| 000 | 76800 [baud] | 38400 [baud] | 19200 [baud] |
| 001 | 38400 | 19200 | 9600 |
| 010 | 19200 | 9600 | 4800 |
| 011 | 9600 | 4800 | 2400 |
| 100 | 4800 | 2400 | 1200 |
| 101 | 2400 | 1200 | 600 |

When TC3 is used as the UART transfer rate (when UARTCR1<BRG> = “110”), the transfer clock and transfer rate are determined as follows:

$$\text{Transfer clock [Hz]} = \text{TC3 source clock [Hz]} / \text{TTREG3 setting value}$$

$$\text{Transfer Rate [baud]} = \text{Transfer clock [Hz]} / 16$$

12.5 Data Sampling Method

The UART receiver keeps sampling input using the clock selected by UARTCR1<BRG> until a start bit is detected in RXD pin input. RT clock starts detecting “L” level of the RXD pin. Once a start bit is detected, the start bit, data bits, stop bit(s), and parity bit are sampled at three times of RT7, RT8, and RT9 during one receiver clock interval (RT clock). (RT0 is the position where the bit supposedly starts.) Bit is determined according to majority rule (The data are the same twice or more out of three samplings).

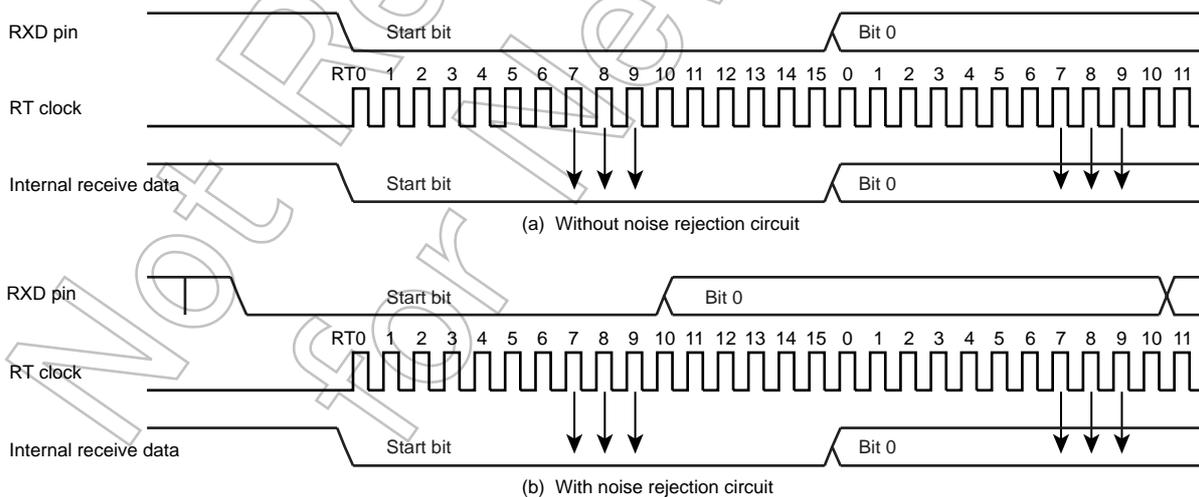


Figure 12-4 Data Sampling Method

12.6 STOP Bit Length

Select a transmit stop bit length (1 bit or 2 bits) by UARTCR1<STBT>.

12.7 Parity

Set parity / no parity by UARTCR1<PE> and set parity type (Odd- or Even-numbered) by UARTCR1<EVEN>.

12.8 Transmit/Receive Operation

12.8.1 Data Transmit Operation

Set UARTCR1<TXE> to “1”. Read UARTSR to check UARTSR<TBEP> = “1”, then write data in TDBUF (Transmit data buffer). Writing data in TDBUF zero-clears UARTSR<TBEP>, transfers the data to the transmit shift register and the data are sequentially output from the TXD pin. The data output include a one-bit start bit, stop bits whose number is specified in UARTCR1<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UARTCR1<BRG>. When data transmit starts, transmit buffer empty flag UARTSR<TBEP> is set to “1” and an INTTXD interrupt is generated.

While UARTCR1<TXE> = “0” and from when “1” is written to UARTCR1<TXE> to when send data are written to TDBUF, the TXD pin is fixed at high level.

When transmitting data, first read UARTSR, then write data in TDBUF. Otherwise, UARTSR<TBEP> is not zero-cleared and transmit does not start.

12.8.2 Data Receive Operation

Set UARTCR1<RXE> to “1”. When data are received via the RXD pin, the receive data are transferred to RDBUF (Receive data buffer). At this time, the data transmitted includes a start bit and stop bit(s) and a parity bit if parity addition is specified. When stop bit(s) are received, data only are extracted and transferred to RDBUF (Receive data buffer). Then the receive buffer full flag UARTSR<RBFL> is set and an INTRXD interrupt is generated. Select the data transfer baud rate using UARTCR1<BRG>.

If an overrun error (OERR) occurs when data are received, the data are not transferred to RDBUF (Receive data buffer) but discarded; data in the RDBUF are not affected.

Note: When a receive operation is disabled by setting UARTCR1<RXE> bit to “0”, the setting becomes valid when data receive is completed. However, if a framing error occurs in data receive, the receive-disabling setting may not become valid. If a framing error occurs, be sure to perform a re-receive operation.

12.9 Status Flag

12.9.1 Parity Error

When parity determined using the receive data bits differs from the received parity bit, the parity error flag UARTSR<PERR> is set to “1”. The UARTSR<PERR> is cleared to “0” when the RDBUF is read after reading the UARTSR.

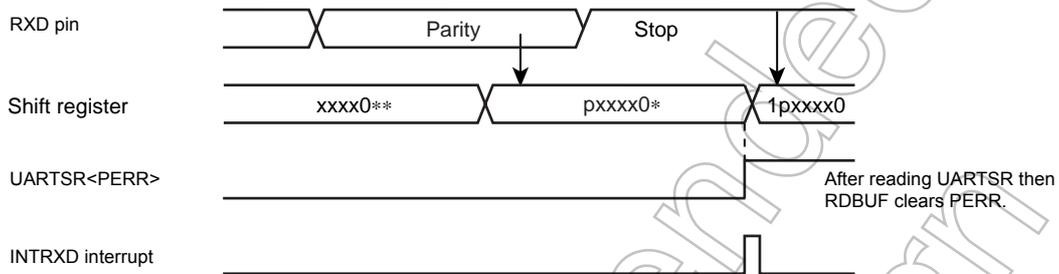


Figure 12-5 Generation of Parity Error

12.9.2 Framing Error

When “0” is sampled as the stop bit in the receive data, framing error flag UARTSR<FERR> is set to “1”. The UARTSR<FERR> is cleared to “0” when the RDBUF is read after reading the UARTSR.

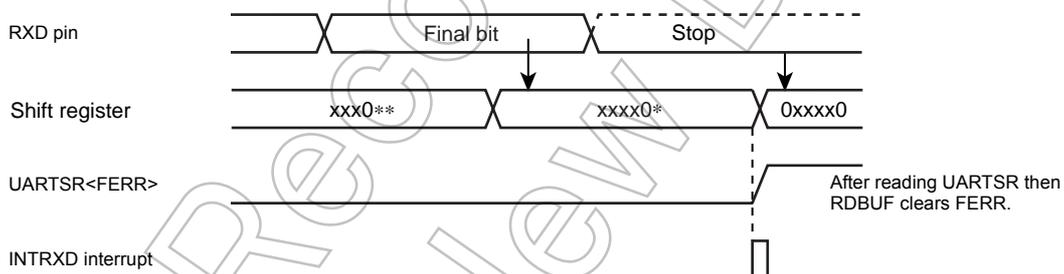


Figure 12-6 Generation of Framing Error

12.9.3 Overrun Error

When all bits in the next data are received while unread data are still in RDBUF, overrun error flag UARTSR<OERR> is set to “1”. In this case, the receive data is discarded; data in RDBUF are not affected. The UARTSR<OERR> is cleared to “0” when the RDBUF is read after reading the UARTSR.

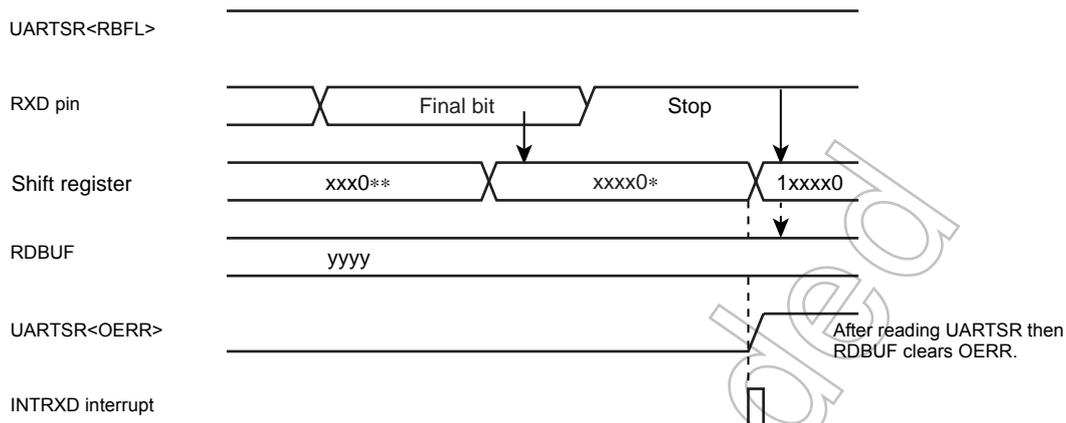


Figure 12-7 Generation of Overrun Error

Note: Receive operations are disabled until the overrun error flag UARTSR<OERR> is cleared.

12.9.4 Receive Data Buffer Full

Loading the received data in RDBUF sets receive data buffer full flag UARTSR<RBFL> to "1". The UARTSR<RBFL> is cleared to "0" when the RDBUF is read after reading the UARTSR.

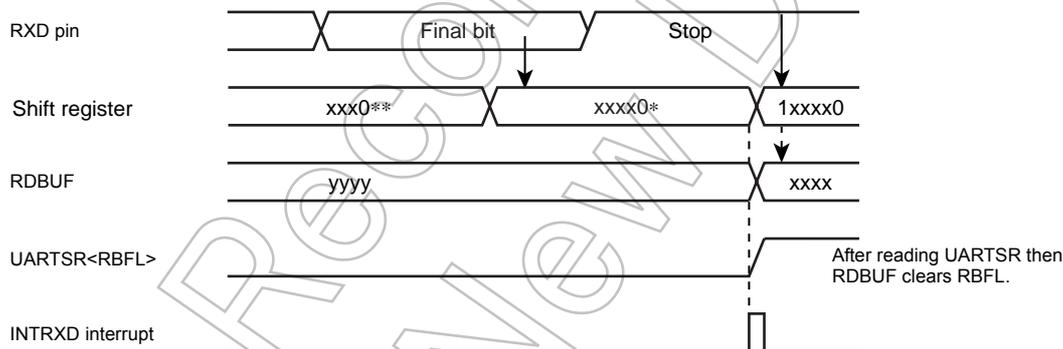


Figure 12-8 Generation of Receive Data Buffer Full

Note: If the overrun error flag UARTSR<OERR> is set during the period between reading the UARTSR and reading the RDBUF, it cannot be cleared by only reading the RDBUF. Therefore, after reading the RDBUF, read the UARTSR again to check whether or not the overrun error flag which should have been cleared still remains set.

12.9.5 Transmit Data Buffer Empty

When no data is in the transmit buffer TDBUF, that is, when data in TDBUF are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag UARTSR<TBEP> is set to "1". The UARTSR<TBEP> is cleared to "0" when the TDBUF is written after reading the UARTSR.

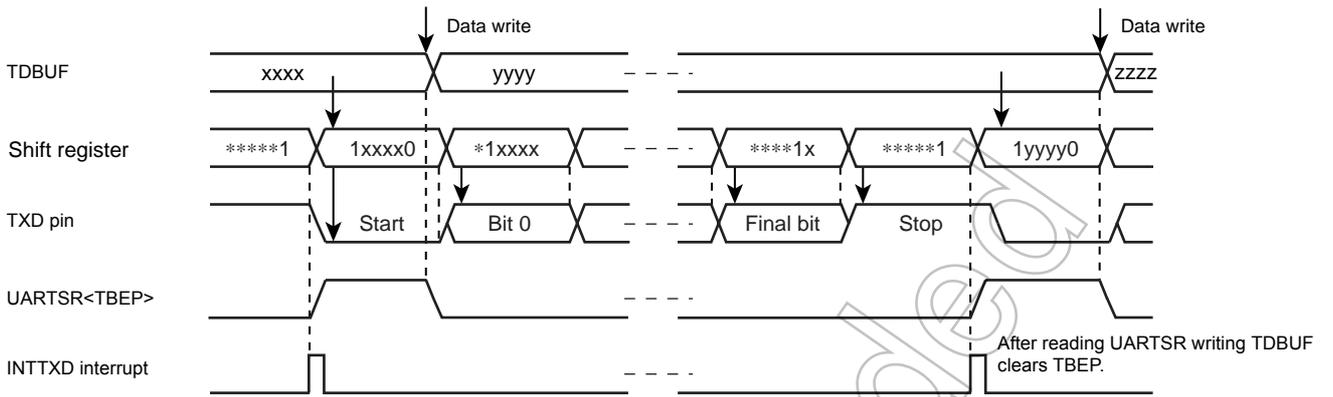


Figure 12-9 Generation of Transmit Data Buffer Empty

12.9.6 Transmit End Flag

When data are transmitted and no data is in TDBUF (UARTSR<TBEP> = “1”), transmit end flag UARTSR<TEND> is set to “1”. The UARTSR<TEND> is cleared to “0” when the data transmit is started after writing the TDBUF.

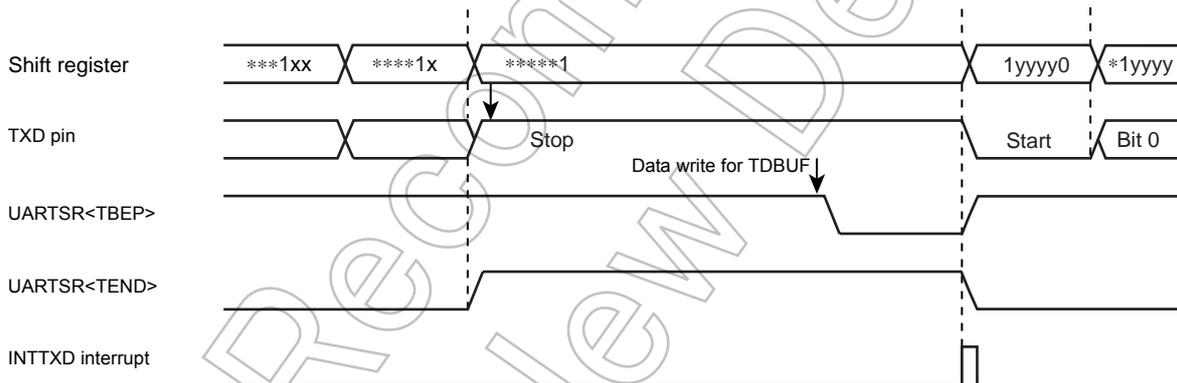


Figure 12-10 Generation of Transmit End Flag and Transmit Data Buffer Empty

Not Recommended
for New Design

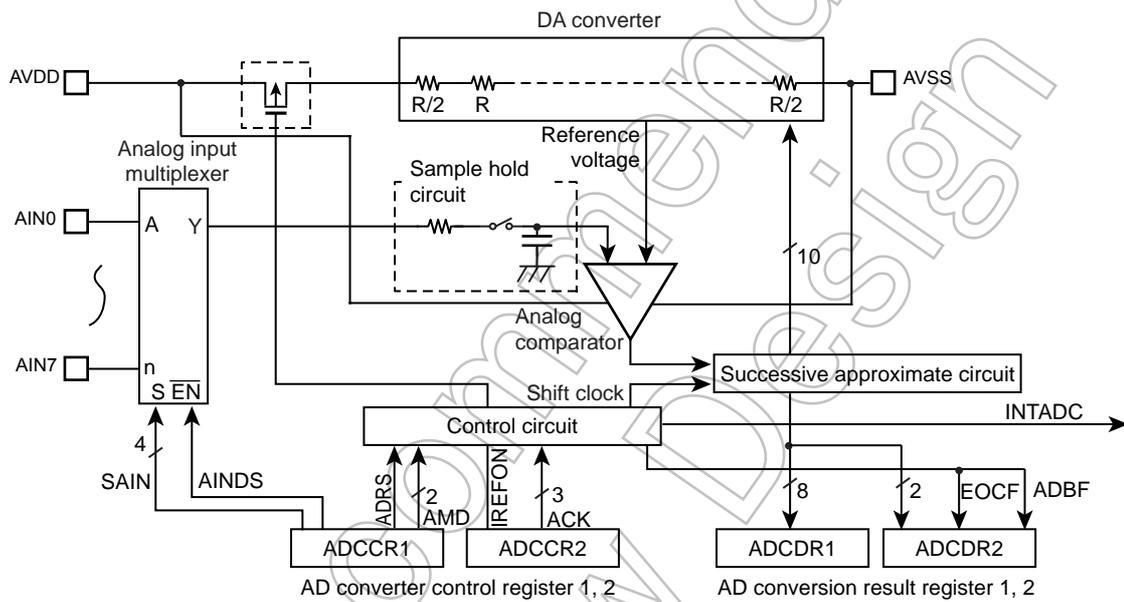
13. 10-bit AD Converter (ADC)

The TMP86PS44UG have a 10-bit successive approximation type AD converter.

13.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 13-1.

It consists of control register ADCCR1 and ADCCR2, converted value register ADCDR1 and ADCDR2, a DA converter, a sample-hold circuit, a comparator, and a successive comparison circuit.



Note: Before using AD converter, set appropriate value to I/O port register combining a analog input port. For details, see the section on "I/O ports".

Figure 13-1 10-bit AD Converter

13.2 Register configuration

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCR1)

This register selects the analog channels and operation mode (Software start or repeat) in which to perform AD conversion and controls the AD converter as it starts operating.

2. AD converter control register 2 (ADCCR2)

This register selects the AD conversion time and controls the connection of the DA converter (Ladder resistor network).

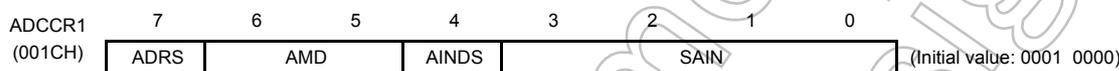
3. AD converted value register 1 (ADCDR1)

This register used to store the digital value after being converted by the AD converter.

4. AD converted value register 2 (ADCDR2)

This register monitors the operating status of the AD converter.

AD Converter Control Register 1



| | | | |
|-------|-----------------------------|--|-----|
| ADRS | AD conversion start | 0: - 1: AD conversion start | R/W |
| AMD | AD operating mode | 00: AD operation disable 01: Software start mode 10: Reserved 11: Repeat mode | |
| AINDS | Analog input control | 0: Analog input enable 1: Analog input disable | |
| SAIN | Analog input channel select | 0000: AIN0 0001: AIN1 0010: AIN2 0011: AIN3 0100: AIN4 0101: AIN5 0110: AIN6 0111: AIN7 1000: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Reserved 1101: Reserved 1110: Reserved 1111: Reserved | |

Note 1: Select analog input channel during AD converter stops (ADCDR2<ADBF> = "0").

Note 2: When the analog input channel is all use disabling, the ADCCR1<AINDS> should be set to "1".

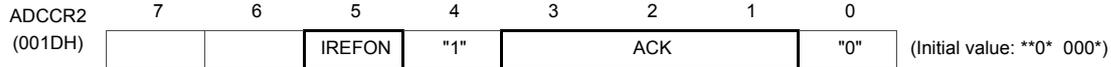
Note 3: During conversion, Do not perform port output instruction to maintain a precision for all of the pins because analog input port use as general input port. And for port near to analog input, Do not input intense signaling of change.

Note 4: The ADCCR1<ADRS> is automatically cleared to "0" after starting conversion.

Note 5: Do not set ADCCR1<ADRS> newly again during AD conversion. Before setting ADCCR1<ADRS> newly again, check ADCDR2<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

Note 6: After STOP or SLOW/SLEEP mode are started, AD converter control register1 (ADCCR1) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCR1 newly after returning to NORMAL1 or NORMAL2 mode.

AD Converter Control Register 2



| | | | |
|--------|---|---|-----|
| IREFON | DA converter (Ladder resistor) connection control | 0: Connected only during AD conversion 1: Always connected | |
| ACK | AD conversion time select (Refer to the following table about the conversion time) | 000: 39/fc 001: Reserved 010: 78/fc 011: 156/fc 100: 312/fc 101: 624/fc 110: 1248/fc 111: Reserved | R/W |

- Note 1: Always set bit0 in ADCCR2 to "0" and set bit4 in ADCCR2 to "1".
- Note 2: When a read instruction for ADCCR2, bit6 to 7 in ADCCR2 read in as undefined data.
- Note 3: After STOP or SLOW/SLEEP mode are started, AD converter control register2 (ADCCR2) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCR2 newly after returning to NORMAL1 or NORMAL2 mode.

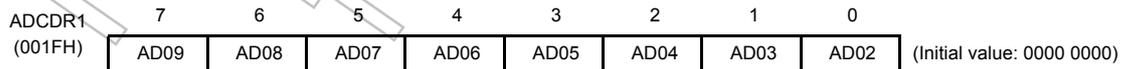
Table 13-1 ACK setting and Conversion time

| Condition ACK | Conversion time | 16 MHz | 8 MHz | 4 MHz | 2 MHz | 10 MHz | 5 MHz | 2.5 MHz |
|------------------|-----------------|---------|----------|----------|----------|----------|----------|----------|
| 000 | 39/fc | - | - | - | 19.5 μs | - | - | 15.6 μs |
| 001 | Reserved | | | | | | | |
| 010 | 78/fc | - | - | 19.5 μs | 39.0 μs | - | 15.6 μs | 31.2 μs |
| 011 | 156/fc | - | 19.5 μs | 39.0 μs | 78.0 μs | 15.6 μs | 31.2 μs | 62.4 μs |
| 100 | 312/fc | 19.5 μs | 39.0 μs | 78.0 μs | 156.0 μs | 31.2 μs | 62.4 μs | 124.8 μs |
| 101 | 624/fc | 39.0 μs | 78.0 μs | 156.0 μs | - | 62.4 μs | 124.8 μs | - |
| 110 | 1248/fc | 78.0 μs | 156.0 μs | - | - | 124.8 μs | - | - |
| 111 | Reserved | | | | | | | |

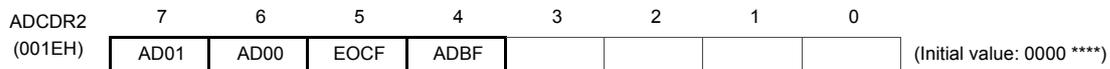
- Note 1: Setting for "-" in the above table are inhibited. fc: High Frequency oscillation clock [Hz]
- Note 2: Set conversion time setting should be kept more than the following time by Power supply voltage(VDD) .

- VDD = 4.5 to 5.5 V 15.6 μs and more
- VDD = 2.7 to 5.5 V 31.2 μs and more

AD Converted value Register 1



AD Converted value Register 2



| | | | |
|------|-------------------------|--|-----------|
| EOCF | AD conversion end flag | 0: Before or during conversion 1: Conversion completed | Read only |
| ADBF | AD conversion BUSY flag | 0: During stop of AD conversion 1: During AD conversion | |

Note 1: The ADCDR2<EOCF> is cleared to "0" when reading the ADCDR1. Therefore, the AD conversion result should be read to ADCDR2 more first than ADCDR1.

Note 2: The ADCDR2<ADBF> is set to "1" when AD conversion starts, and cleared to "0" when AD conversion finished. It also is cleared upon entering STOP mode or SLOW mode .

Note 3: If a read instruction is executed for ADCDR2, read data of bit3 to bit0 are unstable.

Not Recommended for New Design

13.3 Function

13.3.1 Software Start Mode

After setting ADCCR1<AMD> to “01” (software start mode), set ADCCR1<ADRS> to “1”. AD conversion of the voltage at the analog input pin specified by ADCCR1<SAIN> is thereby started.

After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDR1, ADCDR2) and at the same time ADCDR2<EOCF> is set to 1, the AD conversion finished interrupt (INTADC) is generated.

ADRS is automatically cleared after AD conversion has started. Do not set ADCCR1<ADRS> newly again (Restart) during AD conversion. Before setting ADRS newly again, check ADCDR2<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

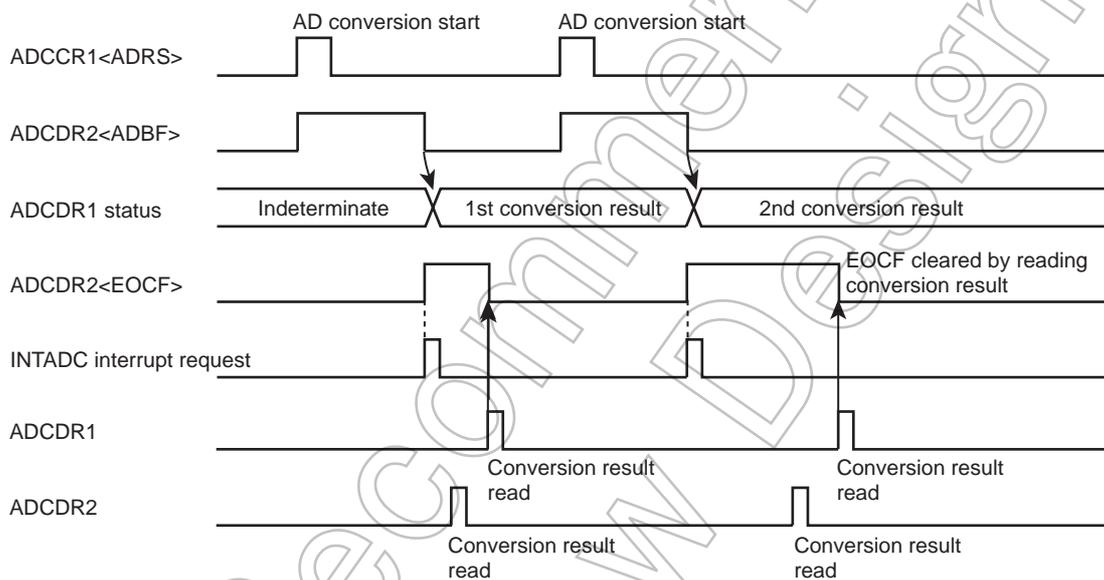


Figure 13-2 Software Start Mode

13.3.2 Repeat Mode

AD conversion of the voltage at the analog input pin specified by ADCCR1<SAIN> is performed repeatedly. In this mode, AD conversion is started by setting ADCCR1<ADRS> to “1” after setting ADCCR1<AMD> to “11” (Repeat mode).

After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDR1, ADCDR2) and at the same time ADCDR2<EOCF> is set to 1, the AD conversion finished interrupt (INTADC) is generated.

In repeat mode, each time one AD conversion is completed, the next AD conversion is started. To stop AD conversion, set ADCCR1<AMD> to “00” (Disable mode) by writing 0s. The AD convert operation is stopped immediately. The converted value at this time is not stored in the AD converted value register.

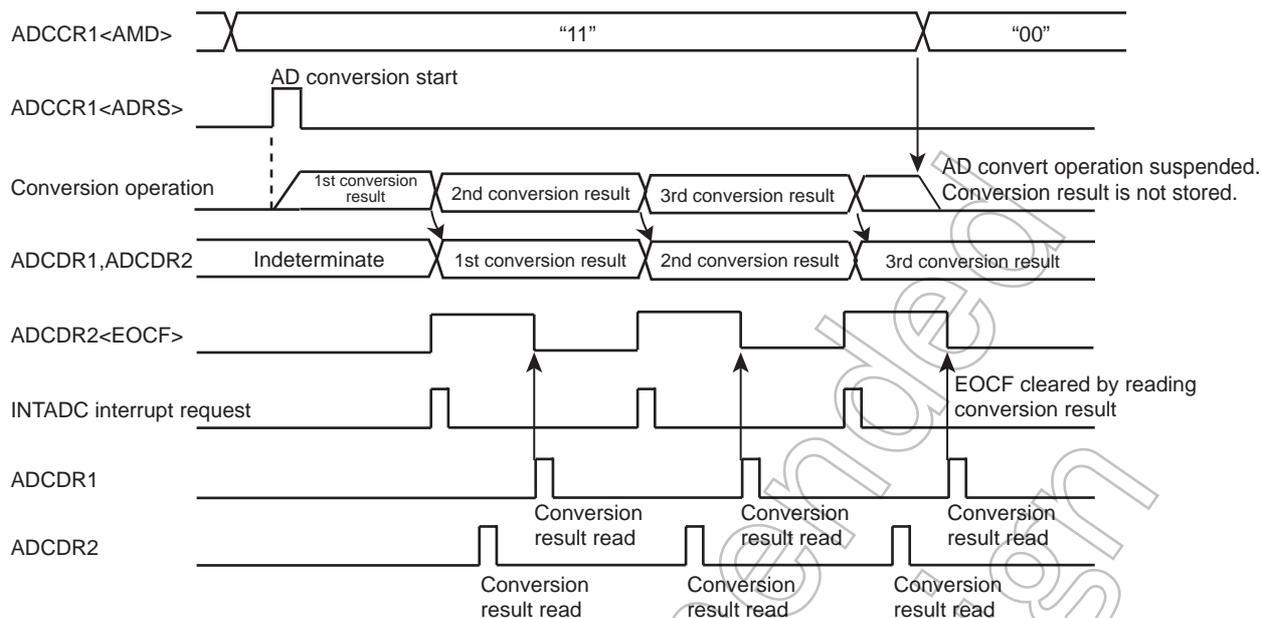


Figure 13-3 Repeat Mode

13.3.3 Register Setting

1. Set up the AD converter control register 1 (ADCCR1) as follows:
 - Choose the channel to AD convert using AD input channel select (SAIN).
 - Specify analog input enable for analog input control (AINDS).
 - Specify AMD for the AD converter control operation mode (software or repeat mode).
2. Set up the AD converter control register 2 (ADCCR2) as follows:
 - Set the AD conversion time using AD conversion time (ACK). For details on how to set the conversion time, refer to Figure 13-1 and AD converter control register 2.
 - Choose IREFON for DA converter control.
3. After setting up (1) and (2) above, set AD conversion start (ADRS) of AD converter control register 1 (ADCCR1) to "1". If software start mode has been selected, AD conversion starts immediately.
4. After an elapse of the specified AD conversion time, the AD converted value is stored in AD converted value register 1 (ADCDR1) and the AD conversion finished flag (EOCF) of AD converted value register 2 (ADCDR2) is set to "1", upon which time AD conversion interrupt INTADC is generated.
5. EOCF is cleared to "0" by a read of the conversion result. However, if reconverted before a register read, although EOCF is cleared the previous conversion result is retained until the next conversion is completed.

Example :After selecting the conversion time 19.5 μ s at 16 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, read the converted value, store the lower 2 bits in address 0009EH and store the upper 8 bits in address 0009FH in RAM. The operation mode is software start mode.

```

; (port setting)
;
; Set port register appropriately before setting AD
; converter registers.
;
; (Refer to section I/O port in details)
LD (ADCCR1), 00100011B ; Select AIN3
LD (ADCCR2), 11011000B ; Select conversion time(312/fc) and operation
mode
SET (ADCCR1), 7 ; ADRS = 1(AD conversion start)
SLOOP: TEST (ADCCR2), 5 ; EOCF= 1 ?
JRS T, SLOOP
LD A, (ADCDR2) ; Read result data
LD (9EH), A
LD A, (ADCDR1) ; Read result data
LD (9FH), A
    
```

13.4 STOP/SLOW Modes during AD Conversion

When standby mode (STOP or SLOW mode) is entered forcibly during AD conversion, the AD convert operation is suspended and the AD converter is initialized (ADCCR1 and ADCCR2 are initialized to initial value). Also, the conversion result is indeterminate. (Conversion results up to the previous operation are cleared, so be sure to read the conversion results before entering standby mode (STOP or SLOW mode).) When restored from standby mode (STOP or SLOW mode), AD conversion is not automatically restarted, so it is necessary to restart AD conversion. Note that since the analog reference voltage is automatically disconnected, there is no possibility of current flowing into the analog reference voltage.

13.5 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 10-bit digital value converted by the AD as shown in Figure 13-4.

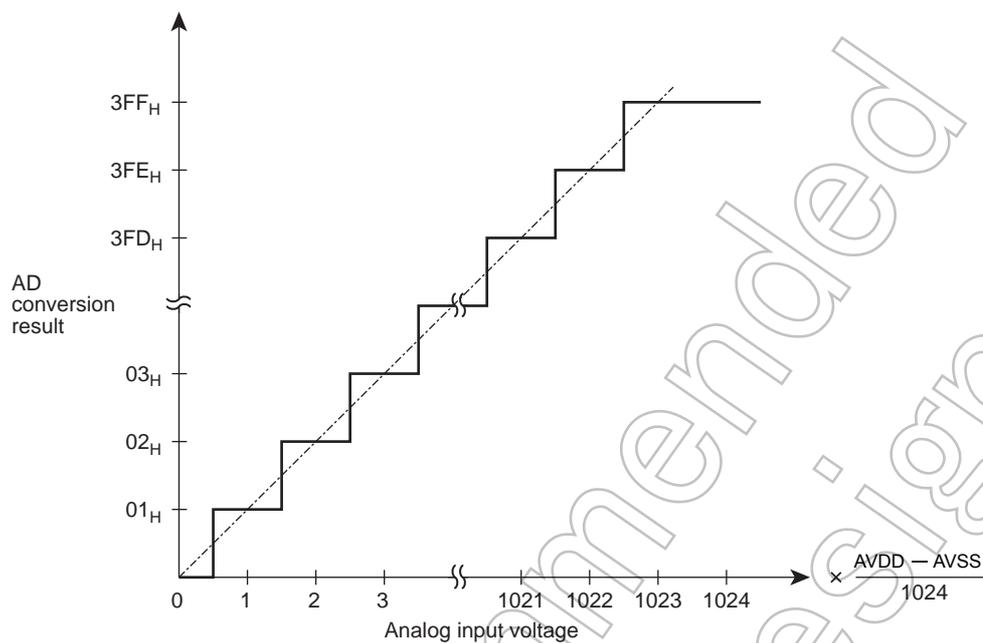


Figure 13-4 Analog Input Voltage and AD Conversion Result (Typ.)

Not Recommended for New Design

13.6 Precautions about AD Converter

13.6.1 Analog input pin voltage range

Make sure the analog input pins (AIN0 to AIN7) are used at voltages within AVDD to AVSS. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain. The other analog input pins also are affected by that.

13.6.2 Analog input shared pins

The analog input pins (AIN0 to AIN7) are shared with input/output ports. When using any of the analog inputs to execute AD conversion, do not execute input/output instructions for all other ports. This is necessary to prevent the accuracy of AD conversion from degrading. Not only these analog input shared pins, some other pins may also be affected by noise arising from input/output to and from adjacent pins.

13.6.3 Noise Countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 13-5. The higher the output impedance of the analog input source, more easily they are susceptible to noise. Therefore, make sure the output impedance of the signal source in your design is 5 k Ω or less. Toshiba also recommends attaching a capacitor external to the chip.

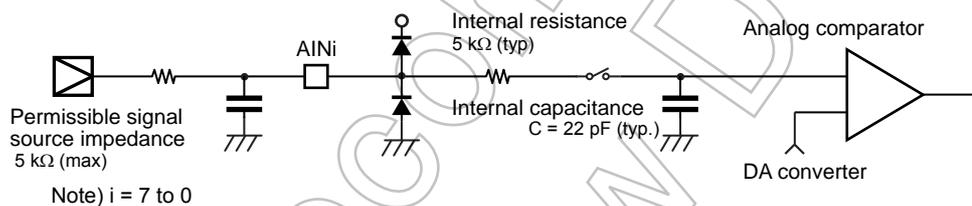


Figure 13-5 Analog Input Equivalent Circuit and Example of Input Pin Processing

Not Recommended
for New Design

14. 8-Bit DA Converter (DAC)

The TMP86PS44UG has 1-channel of 8-bit Digital to Analog (DA) converter.

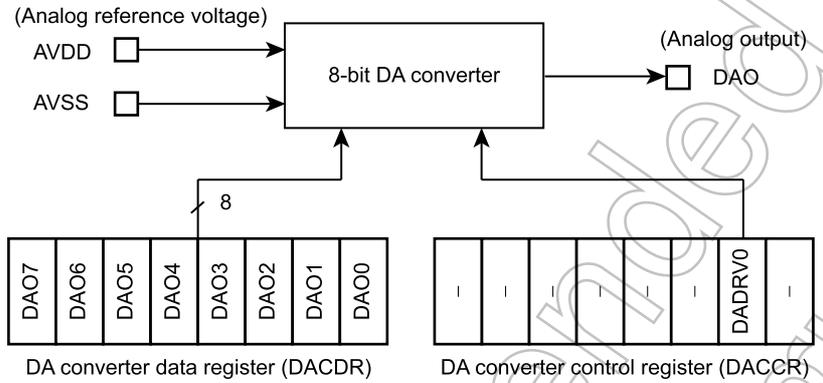


Figure 14-1 8-bit DA converter (DAC)

14.1 Register Configuration

The DA converter are controlled by 2 registers; shown below.

1. DA converter control register (DACCR)
This register controls the analog output. It can be read and written.
2. DA converter data register (DACDR)
This register sets up the digital value conversion by DA converter. It can be read and written.

DA converter control register

| | | | | | | | | | |
|------------------|---|-----|---|---|---|---|--------|---|----------------------------|
| DACCR (002EH) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: 0100 0000) |
| | - | "1" | - | - | - | - | DADRVO | - | |

| | | | |
|--------|----------------------------------|--|-----|
| DADRVO | Register conversion value output | 0: 0 V output (Power down) 1: DACDR conversion value output | R/W |
|--------|----------------------------------|--|-----|

Note: It is set to power down by setting DACCR<DADRVO> to "0".

DA converter data register

| | | | | | | | | | |
|------------------|------|------|------|------|------|------|------|------|----------------------------|
| DACDR (002FH) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | (Initial value: 0000 0000) |
| | DAO7 | DAO6 | DAO5 | DAO4 | DAO3 | DAO2 | DAO1 | DAO0 | |

| | | |
|--------------|---------------------------|-----|
| DAO7 to DAO0 | DA converter digital data | R/W |
|--------------|---------------------------|-----|

14.1.1 DA Converter Operation

An analog voltage reference is $DAREF = (A_{VDD} - A_{VSS})$.

1. Starting DA conversion operation

DA converter operation is activated by setting $DACCR<DADRV0>$ to "1". The digital value in $DACDR$ is converted into an analog value and is output from the DAO pin. After resetting, $DACCR$ and $DACDR$ are cleared to "0". The DA converter goes the power down. (DAO analog level is 0 V.)

2. Logical expression for converted value

The relation between a digital value and analog output voltage in $DACDR$ can be expressed in the following equation.

$$VOA = \frac{32 + 2^0 \times DAO0 + 2^1 \times DAO1 + 2^2 \times DAO2 + 2^3 \times DAO3 + 2^4 \times DAO4 + 2^5 \times DAO5 + 2^6 \times DAO6 + 2^7 \times DAO7}{304} \times DAREF$$

* $DAREF = A_{VDD} - A_{VSS}$

3. The relation between DA conversion result and Analog output voltage.

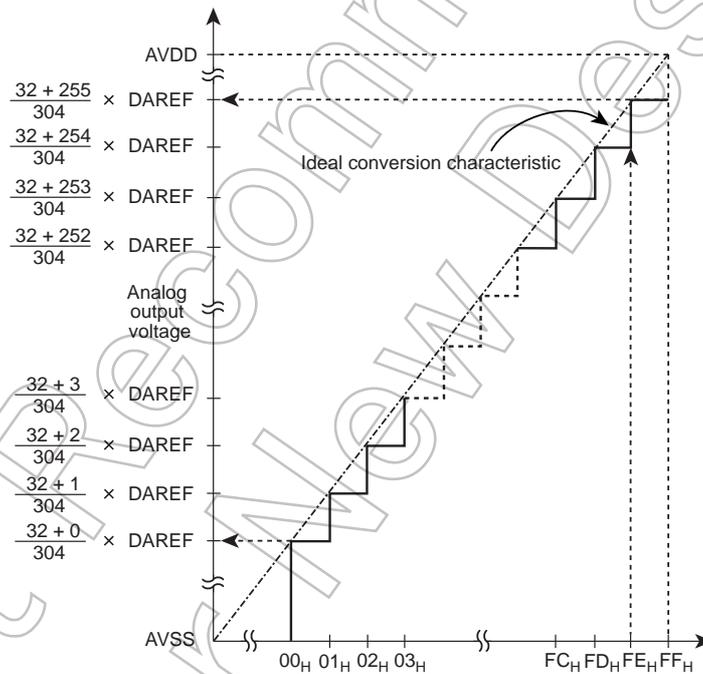


Figure 14-2 DA conversion result (8-bit digital) vs Analog output voltage

Note 1: Analog reference voltage

An analog voltage between A_{VDD} and A_{VSS} ($= DAREF$) is output in 8-bit resolution. Make sure the analog reference pins (A_{VDD} , A_{VSS}) are used at recommended operating condition. It needs that these reference voltage are supplied from a stable power supply.

Note 2: Analog output pin

The analog output pin (DAO) is dedicated to analog output to ensure the precision and characteristics of DA conversion. While DA conversion is being performed, do not execute input/output instructions on the adjacent port. Otherwise, the precision of DA conversion may be adversely affected. Also note that DAO is susceptible to noise (especially power supply noise) in some cases.

Note 3: Anti-noise measure

To ensure the characteristics of DA conversion, place a bypass capacitor between the A_{VDD} and A_{VSS} pins as near to the IC as possible. It is recommended to use a tantalum capacitor (10 μF) and multilayer ceramic capacitor (0.1 μF) in combination.

15. Key-on Wakeup (KWU)

In the TMP86PS44UG, the STOP mode is released by not only P20($\overline{\text{INT5}}/\overline{\text{STOP}}$) pin but also four (STOP2 to STOP5) pins.

When the STOP mode is released by STOP2 to STOP5 pins, the $\overline{\text{STOP}}$ pin needs to be used. In details, refer to the following section " 15.2 Control ".

15.1 Configuration

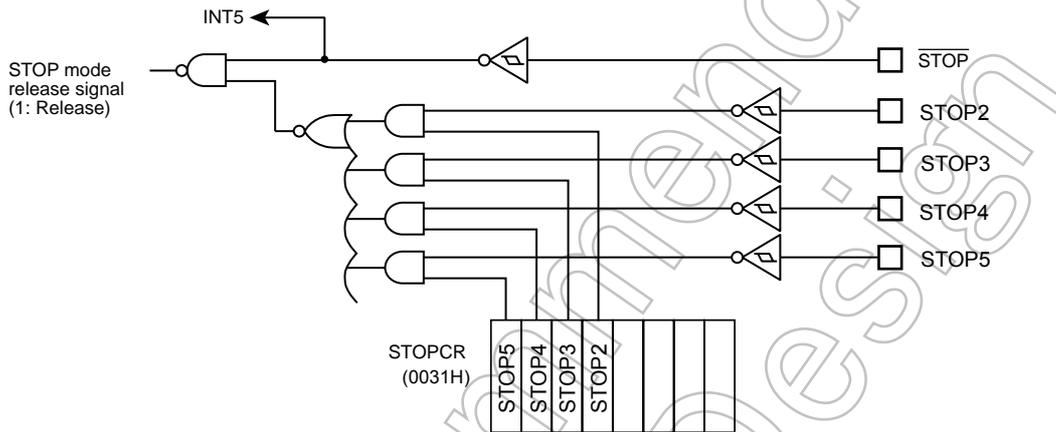


Figure 15-1 Key-on Wakeup Circuit

15.2 Control

STOP2 to STOP5 pins can be controlled by Key-on Wakeup Control Register (STOPCR). It can be configured as enable/disable in 1-bit unit. When those pins are used for STOP mode release, configure corresponding I/O pins to input mode by I/O port register beforehand.

Key-on Wakeup Control Register

| | | | | | | | | | |
|---------|-------|-------|-------|-------|---|---|---|---|----------------------------|
| STOPCR | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| (0031H) | STOP5 | STOP4 | STOP3 | STOP2 | | | | | (Initial value: 0000 ****) |

| | | | |
|-------|-----------------------------|-----------------------|------------|
| STOP5 | STOP mode released by STOP5 | 0:Disable 1:Enable | Write only |
| STOP4 | STOP mode released by STOP4 | 0:Disable 1:Enable | Write only |
| STOP3 | STOP mode released by STOP3 | 0:Disable 1:Enable | Write only |
| STOP2 | STOP mode released by STOP2 | 0:Disable 1:Enable | Write only |

15.3 Function

Stop mode can be entered by setting up the System Control Register (SYSCR1), and can be exited by detecting the "L" level on STOP2 to STOP5 pins, which are enabled by STOPCR, for releasing STOP mode (Note1).

Also, each level of the STOP2 to STOP5 pins can be confirmed by reading corresponding I/O port data register, check all STOP2 to STOP5 pins "H" that is enabled by STOPPCR before the STOP mode is started (Note2,3).

Note 1: When the STOP mode released by the edge release mode (SYSCR1<RELM> = "0"), inhibit input from STOP2 to STOP5 pins by Key-on Wakeup Control Register (STOPPCR) or must be set "H" level into STOP2 to STOP5 pins that are available input during STOP mode.

Note 2: When the $\overline{\text{STOP}}$ pin input is high or STOP2 to STOP5 pins input which is enabled by STOPPCR is low, executing an instruction which starts STOP mode will not place in STOP mode but instead will immediately start the release sequence (Warm up).

Note 3: The input circuit of Key-on Wakeup input and Port input is separated, so each input voltage threshold value is different. Therefore, a value comes from port input before STOP mode start may be different from a value which is detected by Key-on Wakeup input (Figure 15-2).

Note 4: $\overline{\text{STOP}}$ pin doesn't have the control register such as STOPPCR, so when STOP mode is released by STOP2 to STOP5 pins, $\overline{\text{STOP}}$ pin also should be used as STOP mode release function.

Note 5: In STOP mode, Key-on Wakeup pin which is enabled as input mode (for releasing STOP mode) by Key-on Wakeup Control Register (STOPPCR) may generate the penetration current, so the said pin must be disabled AD conversion input (analog voltage input).

Note 6: When the STOP mode is released by STOP2 to STOP5 pins, the level of $\overline{\text{STOP}}$ pin should hold "L" level (Figure 15-3).

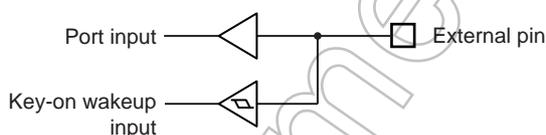


Figure 15-2 Key-on Wakeup Input and Port Input

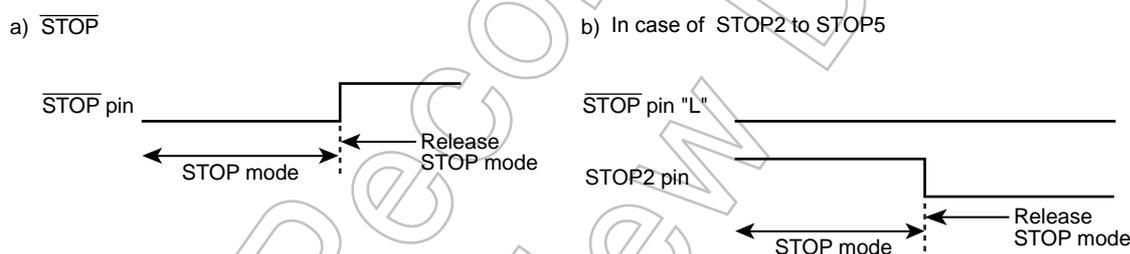


Figure 15-3 Priority of $\overline{\text{STOP}}$ pin and STOP2 to STOP5 pins

Table 15-1 Release level (edge) of STOP mode

| Pin name | Release level (edge) | |
|--------------------------|-----------------------------|-------------------|
| | SYSCR1<RELM>="1" (Note2) | SYSCR1<RELM>="0" |
| $\overline{\text{STOP}}$ | "H" level | Rising edge |
| STOP2 | "L" level | Don't use (Note1) |
| STOP3 | "L" level | Don't use (Note1) |
| STOP4 | "L" level | Don't use (Note1) |
| STOP5 | "L" level | Don't use (Note1) |

16. OTP operation

This section describes the function and basic operational blocks of TMP86PS44UG. The TMP86PS44UG has PROM in place of the mask ROM which is included in the TMP86CS44UG. The configuration and function are the same as the TMP86CS44UG. In addition, TMP86PS44UG operates as the single clock mode when releasing reset. When using the dual clock mode, oscillate a low-frequency clock by [SET. (SYSCR2). XTEN] command at the beginning of program.

16.1 Operating mode

The TMP86PS44UG has MCU mode and PROM mode.

16.1.1 MCU mode

The MCU mode is set by fixing the TEST/VPP pin to the low level. (TEST/VPP pin cannot be used open because it has no built-in pull-down resistor).

16.1.1.1 Program Memory

The TMP86PS44UG has 60K bytes built-in one-time-PROM (addresses 1000 to FFFFH in the MCU mode, addresses 0000 to EFFFH in the PROM mode).

When using TMP86PS44UG for evaluation of mask ROM products, the program is written in the program storing area shown in Figure 16-1.

Since the TMP86PS44UG supports several mask ROM sizes, check the difference in memory size and program storing area between the one-time PROM and the mask ROM to be used.

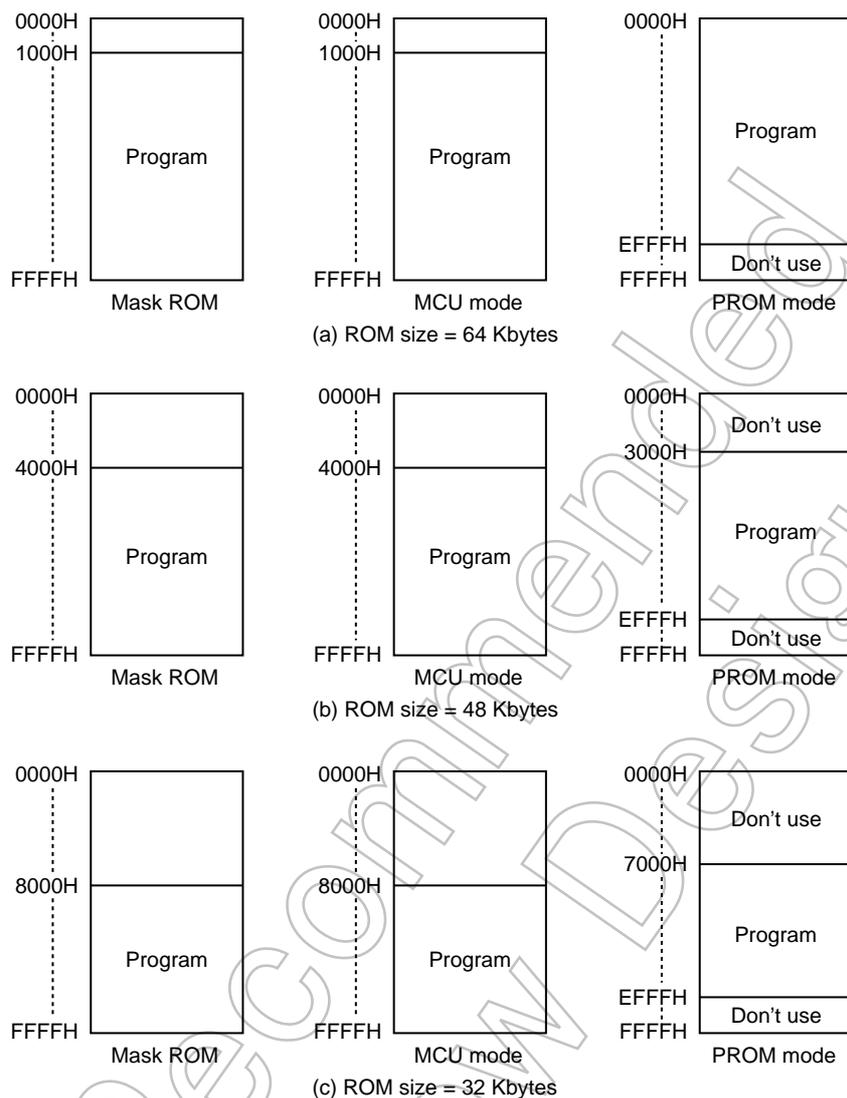


Figure 16-1 Program Memory Area

Note: The area that is not in use should be set data to FFH, or a general-purpose PROM programmer should be set only in the program memory area to access.

16.1.1.2 Data Memory

TMP86PS44UG has a built-in 1024 bytes Data memory (static RAM).

16.1.1.3 Input/Output Circuitry

1. Control pins

The control pins of the TMP86PS44UG are the same as those of the TMP86CS44UG except that the TEST pin does not have a built-in pull-down resistor.

2. I/O ports

The I/O circuitries of the TMP86PS44UG I/O ports are the same as those of the TMP86CS44UG.

16.1.2 PROM mode

The PROM mode is set by setting the $\overline{\text{RESET}}$ pin, TEST pin and other pins as shown in Table 16-1 and Figure 16-2. The programming and verification for the internal PROM is achieved by using a general-purpose PROM programmer with the adaptor socket.

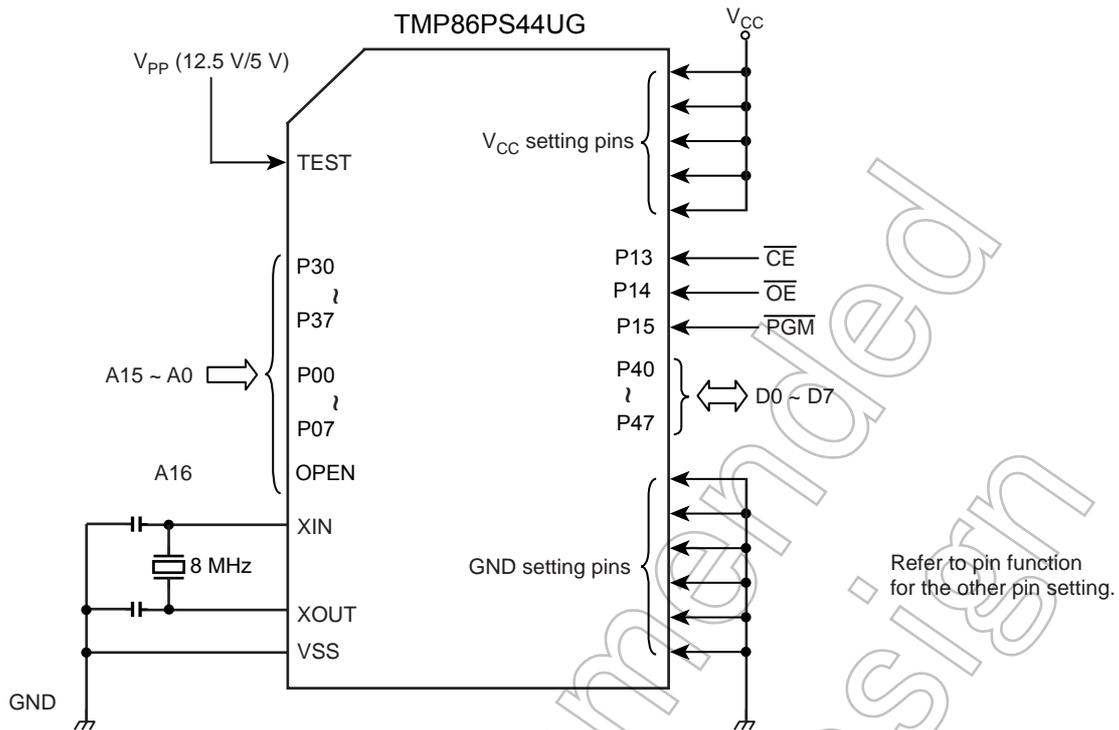
Table 16-1 Pin name in PROM mode

| Pin name (PROM mode) | I/O | Function | Pin name (MCU mode) |
|---------------------------|--------------|---|---------------------------|
| A15 to A8 | Input | Program memory address input | P07 to P00 |
| A7 to A0 | Input | Program memory address input | P37 to P30 |
| D7 to D0 | Input/Output | Program memory data input/output | P47 to P40 |
| $\overline{\text{CE}}$ | Input | Chip enable signal input | P13 |
| $\overline{\text{OE}}$ | Input | Output enable signal input | P14 |
| $\overline{\text{PGM}}$ | Input | Program mode signal input | P15 |
| VPP | Power supply | +12.75V/5V (Power supply of program) | TEST |
| VCC | Power supply | +6.25V/5V | VDD |
| GND | Power supply | 0V | VSS |
| VCC | Setting pin | Fix to "H" level in PROM mode | AVDD,P11,P21 |
| GND | Setting pin | Fix to "L" level in PROM mode | AVSS,P10,P12,P20,P22 |
| $\overline{\text{RESET}}$ | Setting pin | Fix to "L" level in PROM mode | $\overline{\text{RESET}}$ |
| XIN (CLK) | Input | Set oscillation with resonator In case of external CLK input, set CLK to XIN | XIN |
| XOUT | Output | and set XOUT to open. | XOUT |

Note 1: The high-speed program mode can be used. The setting is different according to the type of PROM programmer to use, refer to each description of PROM programmer.
TMP86PS44UG does not support the electric signature mode, apply the ROM type of PROM programmer to TC571000D/AD.

Note 2: No pin is applied to A16 pin of TC571000D/AD(Open) in PROM mode.

Always set the adapter socket switch to the "N" side when using TOSHIBA's adaptor socket.



Note 1: EPROM adaptor socket (TC571000 • 1M bit EPROM)

Note 2: PROM programmer connection adaptor sockets
 BM11687 for TMP86PS44UG

Note 3: Inside pin name for TMP86PS44UG
 Outside pin name for EPROM

Figure 16-2 PROM mode setting

Not Recommended for New Design

16.1.2.1 Programming Flowchart (High-speed program writing)

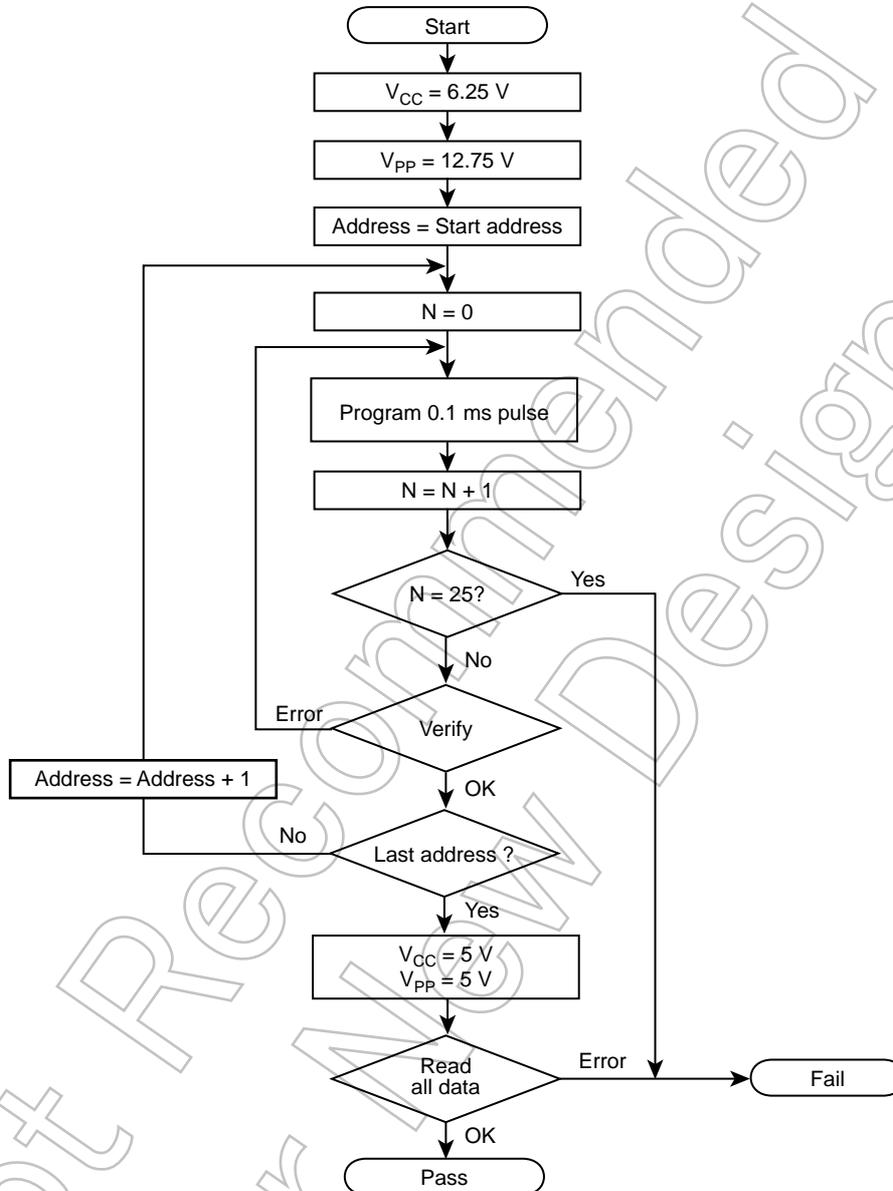


Figure 16-3 Programming Flowchart

The high-speed programming mode is set by applying $V_{pp}=12.75V$ (programming voltage) to the V_{pp} pin when the $V_{cc} = 6.25 V$. After the address and data are fixed, the data in the address is written by applying 0.1[msec] of low level program pulse to \overline{PGM} pin. Then verify if the data is written. If the programmed data is incorrect, another 0.1[msec] pulse is applied to \overline{PGM} pin. This programming procedure is repeated until correct data is read from the address (maximum of 25 times). Subsequently, all data are programmed in all address. When all data were written, verify all address under the condition $V_{cc}=V_{pp}=5V$.

16.1.2.2 Program Writing using a General-purpose PROM Programmer

1. Recommended OTP adaptor

BM11687 for TMP86PS44UG

2. Setting of OTP adaptor

Set the switch (SW1) to "N" side.

3. Setting of PROM programmer

a. Set PROM type to TC571000D/AD.

Vpp: 12.75 V (high-speed program writing mode)

b. Data transmission (or Copy) (Note 1)

The PROM of TMP86PS44UG is located on different address; it depends on operating mode: MCU mode and PROM mode. When you write the data of ROM for mask ROM products, the data should be transferred (or copied) from the address for MCU mode to that for PROM mode before writing operation is executed. For the applicable program areas of MCU mode and PROM mode are different, refer to TMP86PS44UG" Figure 16-1 Program Memory Area".

Example: In the block transfer (copy) mode, executed as below.

60KB ROM capacity: 01000~0FFFFH → 00000~0EFFFH

48 KB ROM capacity: 04000~0FFFFH → 03000~0EFFFH

32KB ROM capacity: 08000~0FFFFH → 07000~0EFFFH

c. Setting of the program address (Note 1)

Start address: 0000H (When 48KB ROM capacity, start address is 3000H.

When 32 KB, ROM capacity, start address is 7000H.)

End address: EFFFH

4. Writing

Write and verify according to the above procedure "Setting of PROM programmer".

Note 1: For the setting method, refer to each description of PROM programmer.

Make sure to set the data of address area that is not in use to FFH.

Note 2: When setting MCU to the adaptor or when setting the adaptor to the PROM programmer, set the first pin of the adaptor and that of PROM programmer socket matched. If the first pin is conversely set, MCU or adaptor or programmer would be damaged.

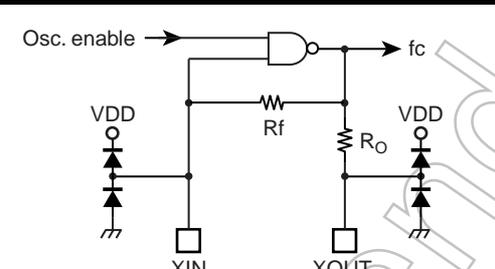
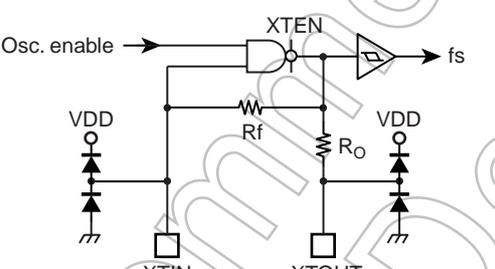
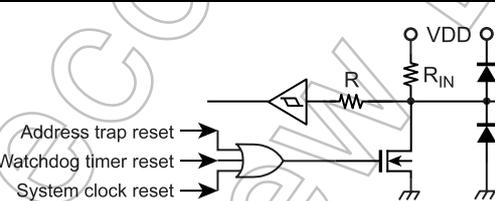
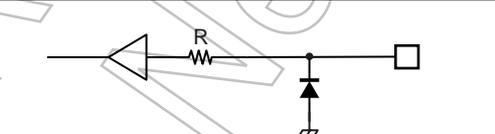
Note 3: The TMP86PS44UG does not support the electric signature mode.

If PROM programmer uses the signature, the device would be damaged because of applying voltage of 12±0.5V to pin 9(A9) of the address. Don't use the signature.

17. Input/Output Circuitry

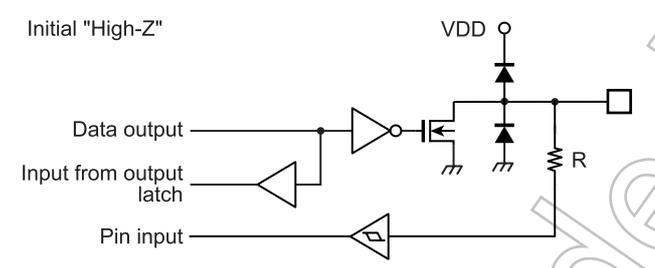
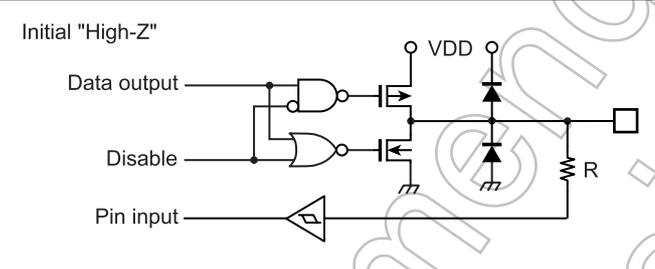
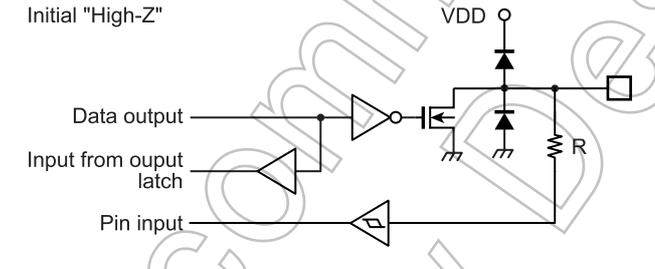
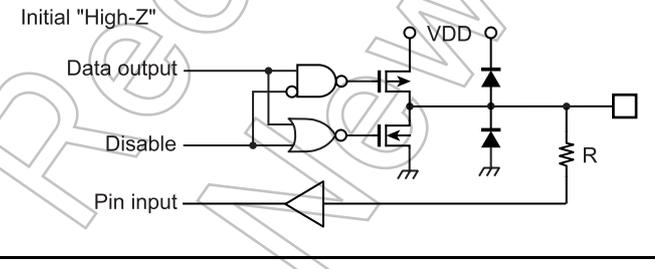
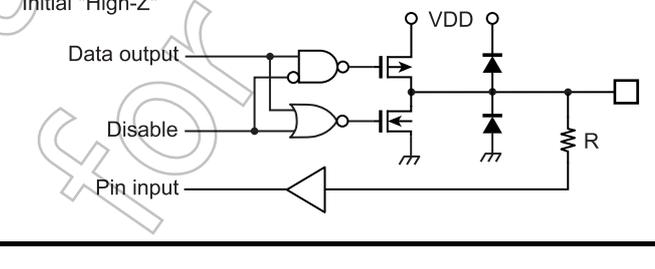
17.1 Control Pins

The input/output circuitries of the TMP86PS44UG control pins are shown below.

| Control Pin | I/O | Input/Output Circuitry | Remarks |
|---------------------------|-----------------|--|---|
| XIN XOUT | Input Output |  | Resonator connecting pins (high-frequency) $R_f = 1.2\text{ M}\Omega$ (typ.) $R_o = 1.5\text{ k}\Omega$ (typ.) |
| XTIN XTOUT | Input Output |  | Resonator connecting pins (Low-frequency) $R_f = 6\text{ M}\Omega$ (typ.) $R_o = 220\text{ k}\Omega$ (typ.) |
| $\overline{\text{RESET}}$ | Input Output |  | Hysteresis input Pull-up resistor $R_{IN} = 220\text{ k}\Omega$ (typ.) $R = 1\text{ k}\Omega$ (typ.) |
| TEST | Input |  | Without pull-down resistor $R = 1\text{ k}\Omega$ (typ.) Fix the TEST pin at low-level in MCU mode. |

Note: The TEST pin of the TMP86PS44 does not have a pull-down resistor and protect diode (D1). Fix the TEST pin at low-level in MCU mode.

17.2 Input/Output Ports

| Port | I/O | Input/Output Circuitry | Remarks |
|------|-----|--|--|
| P0 | I/O | <p>Initial "High-Z"</p>  | <p>Sink open drain output High current output Hysteresis input R = 100 Ω (typ.)</p> |
| P1 | I/O | <p>Initial "High-Z"</p>  | <p>Tri-state I/O Hysteresis input R = 100 Ω (typ.)</p> |
| P2 | I/O | <p>Initial "High-Z"</p>  | <p>Sink open drain output High current output Hysteresis input R = 100 Ω (typ.)</p> |
| P3 | I/O | <p>Initial "High-Z"</p>  | <p>Tri-state I/O R = 100 Ω (typ.)</p> |
| P4 | I/O | <p>Initial "High-Z"</p>  | <p>Tri-state I/O High current output (Nch) R = 100 Ω (typ.)</p> |

18. Electrical Characteristics

18.1 Absolute Maximum Ratings

The absolute maximum ratings are rated values, which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

(V_{SS} = 0 V)

| Parameter | Symbol | Pins | Ratings | Unit |
|---|---------------------|--------------------------|-------------------------------|------|
| Supply voltage | V _{DD} | | -0.3 to 6.5 | V |
| Program voltage | V _{PP} | TEST/V _{PP} pin | -0.3 to 13.0 | V |
| Input voltage | V _{IN} | | -0.3 to V _{DD} + 0.3 | V |
| Output voltage | V _{OUT} | | -0.3 to V _{DD} + 0.3 | V |
| Output current (Per 1 pin) | I _{OUT1} | P1, P3, P4 port | -1.8 | mA |
| | I _{OUT2} | P1, P3 port | 3.2 | |
| | I _{OUT3} | P0, P2, P4 port | 30 | |
| Output current (Total) | Σ I _{OUT1} | P1, P3 port | 60 | |
| | Σ I _{OUT2} | P0, P2, P4 port | 80 | |
| Power dissipation [T _{opr} = 85°C] | PD | | 250 | mW |
| Soldering temperature (Time) | T _{sld} | | 260 (10 s) | °C |
| Storage temperature | T _{stg} | | -55 to 125 | |
| Operating temperature | T _{opr} | | -40 to 85 | |

18.2 Operating Condition

The Operating Conditions shows the conditions under which the device be used in order for it to operate normally while maintaining its quality. If the device is used outside the range of Operating Conditions (power supply voltage, operating temperature range, or AC/DC rated values), it may operate erratically. Therefore, when designing your application equipment, always make sure its intended working conditions will not exceed the range of Operating Conditions.

($V_{SS} = 0\text{ V}$, $T_{opr} = -40\text{ to }85^{\circ}\text{C}$)

| Parameter | Symbol | Pins | Condition | Min | Max | Unit | |
|---------------------------|-------------------|-------------------------|---|-------------------------|-------------------------|------|----------------------|
| Supply voltage | V_{DD} | | $f_c = 16\text{ MHz}$ | NORMAL1, 2 mode | 4.5 | 5.5 | V |
| | | | | IDLE0, 1, 2 mode | | | |
| | | | $f_c = 8\text{ MHz}$ | NORMAL1, 2 mode | 2.7 | | |
| | | | | IDLE0, 1, 2 mode | | | |
| | | | $f_c = 4.2\text{ MHz}$ | NORMAL1, 2 mode | 2.0 | | |
| | | | | IDLE0, 1, 2 mode | | | |
| $f_s = 32.768\text{ kHz}$ | SLOW1, 2 mode | 2.0 | | | | | |
| | SLEEP0, 1, 2 mode | | | | | | |
| | | STOP mode | 2.0 | | | | |
| Input high level | V_{IH1} | Except hysteresis input | $V_{DD} \geq 4.5\text{ V}$ | $V_{DD} \times 0.70$ | V_{DD} | | |
| | V_{IH2} | Hysteresis input | | $V_{DD} \times 0.75$ | | | |
| | V_{IH3} | | | $V_{DD} < 4.5\text{ V}$ | | | $V_{DD} \times 0.90$ |
| Input low level | V_{IL1} | Except hysteresis input | $V_{DD} \geq 4.5\text{ V}$ | 0 | $V_{DD} \times 0.30$ | | |
| | V_{IL2} | Hysteresis input | | | $V_{DD} \times 0.25$ | | |
| | V_{IL3} | | | | $V_{DD} < 4.5\text{ V}$ | | $V_{DD} \times 0.10$ |
| Clock frequency | f_c | XIN, XOUT | $V_{DD} = 2.7\text{ V to }5.5\text{ V}$ | 1.0 | 8.0 | MHz | |
| | | | $V_{DD} = 4.5\text{ V to }5.5\text{ V}$ | | 16.0 | | |
| | f_s | XTIN, XTOUT | | 30.0 | 34.0 | kHz | |

18.3 DC Characteristics

(V_{SS} = 0 V, Topr = -40 to 85°C)

| Parameter | Symbol | Pins | Condition | Min | Typ. | Max | Unit |
|-----------------------------------|------------------|--|--|-----|------|-----|------|
| Hysteresis voltage | V _{HS} | Hysteresis input | | - | 0.9 | - | V |
| Input current | I _{IN1} | TEST | V _{DD} = 5.5 V, V _{IN} = 5.5 V/0 V | - | - | ±2 | μA |
| | I _{IN2} | Sink open drain, Tri-state port | | | | | |
| | I _{IN3} | $\overline{\text{RESET}}$, $\overline{\text{STOP}}$ pull-up | | | | | |
| Input resistance | R _{IN} | $\overline{\text{RESET}}$ pull-up | | 100 | 200 | 450 | kΩ |
| Output leakage current | I _{LO1} | Sink open drain port | V _{DD} = 5.5 V, V _{OUT} = 5.5 V | - | - | 2 | μA |
| | I _{LO2} | Tri-state port | V _{DD} = 5.5 V, V _{OUT} = 5.5 V/0 V | - | - | ±2 | |
| Output high voltage | V _{OH1} | Tri-state port | V _{DD} = 4.5 V, I _{OH} = -0.7 mA | 4.1 | - | - | V |
| Output low voltage | V _{OL} | Except XOUT, P0, P2, P4 port | V _{DD} = 4.5 V, I _{OL} = 1.6 mA | - | - | 0.4 | |
| Output low current | I _{OL} | High current port (P0, P2, P4 port) | V _{DD} = 4.5 V, V _{OL} = 1.0 V | - | 20 | - | mA |
| Supply current in NORMAL1, 2 mode | I _{DD} | | V _{DD} = 5.5 V V _{IN} = 5.3 V/0.2 V f _c = 16 MHz f _s = 32.768 kHz | - | 10.5 | 12 | |
| Supply current in IDLE0,1, 2 mode | | | - | 5.5 | 6.5 | | |
| Supply current in SLOW1 mode | | | - | 8 | 20 | | |
| Supply current in SLEEP1 mode | | | - | 5 | 15 | | |
| Supply current in SLEEP0 mode | | | - | 4 | 13 | | |
| Supply current in STOP mode | | | - | 0.5 | 10 | | |
| | | | V _{DD} = 3.0 V V _{IN} = 2.8 V/0.2 V f _s = 32.768 kHz | - | - | - | μA |
| | | | V _{DD} = 5.5 V V _{IN} = 5.3 V/0.2 V | - | - | - | |

Note 1: Typical values show those at Topr = 25°C, V_{DD} = 5 V.

Note 2: Input current (I_{IN3}); The current through pull-up resistor is not included.

Note 3: I_{DD} does not include I_{REF} current.

Note 4: The supply currents of SLOW2 and SLEEP2 modes are equivalent to IDLE 0,1,2.

18.4 AD Conversion Characteristics

($V_{SS} = 0.0\text{ V}$, $4.5\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $T_{opr} = -40\text{ to }85^\circ\text{C}$)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|--|------------|--|-----------|------|-----------|------|
| Analog reference voltage | V_{AREF} | $V_{AREF} = A_{VDD} - A_{VSS}$ | A_{VDD} | – | A_{VDD} | V |
| Power supply voltage of analog control circuit | A_{VDD} | $A_{VSS} = V_{SS}$ | V_{DD} | | | |
| Analog input voltage | V_{AIN} | $A_{VSS} = V_{SS}$, $A_{VDD} = V_{DD}$ | V_{SS} | – | V_{DD} | |
| Power supply current of analog reference voltage | I_{REF} | $V_{DD} = A_{VDD} = 5.5\text{ V}$ $V_{SS} = A_{VSS} = 0.0\text{ V}$ | – | 0.6 | 1.0 | mA |
| Non linearity error | | $V_{DD} = A_{VDD} = 5.0\text{ V}$ $V_{SS} = A_{VSS} = 0.0\text{ V}$ $V_{AREF} = A_{VDD} - A_{VSS}$ | – | – | ± 2 | LSB |
| Zero point error | | | – | – | ± 2 | |
| Full scale error | | | – | – | ± 2 | |
| Total error | | | – | – | ± 4 | |

($V_{SS} = 0.0\text{ V}$, $2.7\text{ V} \leq V_{DD} < 4.5\text{ V}$, $T_{opr} = -40\text{ to }85^\circ\text{C}$)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|--|------------|--|-----------|------|-----------|------|
| Analog reference voltage | V_{AREF} | $V_{AREF} = A_{VDD} - A_{VSS}$ | A_{VDD} | – | A_{VDD} | V |
| Power supply voltage of analog control circuit | A_{VDD} | $A_{VSS} = V_{SS}$ | V_{DD} | | | |
| Analog input voltage | V_{AIN} | $A_{VSS} = V_{SS}$, $A_{VDD} = V_{DD}$ | V_{SS} | – | V_{DD} | |
| Power supply current of analog reference voltage | I_{REF} | $V_{DD} = A_{VDD} = 4.5\text{ V}$ $V_{SS} = A_{VSS} = 0.0\text{ V}$ | – | 0.5 | 0.8 | mA |
| Non linearity error | | $V_{DD} = A_{VDD} = 2.7\text{ V}$ $V_{SS} = A_{VSS} = 0.0\text{ V}$ $V_{AREF} = A_{VDD} - A_{VSS}$ | – | – | ± 2 | LSB |
| Zero point error | | | – | – | ± 2 | |
| Full scale error | | | – | – | ± 2 | |
| Total error | | | – | – | ± 4 | |

Note 1: The total error includes all errors except a quantization error, and is defined as a maximum deviation from the ideal conversion line.

Note 2: Conversion time is different in recommended value by power supply voltage.
About conversion time, please refer to "Register Configuration".

Note 3: Please use input voltage to AIN input Pin in limit of $V_{AREF} - V_{SS}$.
When voltage of range outside is input, conversion value becomes unsettled and gives affect to other channel conversion value.

Note 4: Analog Reference Voltage Range: $V_{AREF} = A_{VDD} - A_{VSS}$

18.5 DA Conversion Characteristics

($V_{SS} = 0\text{ V}$, $V_{DD} = 4.5\text{ to }5.5\text{ V}$, $T_{opr} = -40\text{ to }85^\circ\text{C}$)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|---|------------|---|----------|------|----------|---------------|
| Analog reference voltage | A_{VDD} | $A_{VDD} = V_{DD}$ | 4.5 | – | 5.5 | V |
| | A_{VSS} | $A_{VSS} = V_{SS}$ | V_{SS} | – | V_{SS} | |
| Resolution | | | – | – | 8 | Bits |
| Linearity error (Note1) | | $A_{VDD} = V_{DD} = 4.5\text{ to }5.5\text{ V}$ $A_{VSS} = V_{SS} = 0\text{ V}$ Load conditions 5 pF, 10 M Ω | – | – | ± 5 | LSB |
| Settling time | | | | | 4 | ms |
| Reference current (DA power consumption) (Note2) | I_{DREF} | No load, $V_{DD} = 5.5\text{ V}$, $f_c = 16\text{ MHz}$ | – | 350 | 500 | μA |
| DAO output current (Note3) | I_{DAO} | $A_{VDD} = V_{DD} = 4.5\text{ V}$ $A_{VSS} = V_{SS} = 0\text{ V}$ No load 0.5 V bias when inputting 7FH | 14 | 19 | – | |

($V_{SS} = 0\text{ V}$, $V_{DD} = 2.7\text{ to }4.5\text{ V}$, $T_{opr} = -40\text{ to }85^\circ\text{C}$)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit | |
|---|------------|---|--|------|----------|---------------|-----|
| Analog reference voltage | A_{VDD} | $A_{VDD} = V_{DD}$ | 2.7 | – | 4.5 | V | |
| | A_{VSS} | $A_{VSS} = V_{SS}$ | V_{SS} | – | V_{SS} | | |
| Resolution | | | – | – | 8 | Bits | |
| Linearity error (Note1) | | $A_{VDD} = V_{DD} = 2.7\text{ to }4.5\text{ V}$ $A_{VSS} = V_{SS} = 0\text{ V}$ Load conditions 5 pF, 10 M Ω | $-10^\circ\text{C} \leq T_a \leq 85^\circ\text{C}$ | – | – | +5 | LSB |
| | | | $-40^\circ\text{C} \leq T_a < -10^\circ\text{C}$ | – | – | +5 | |
| Settling time | | | | | 6 | ms | |
| Reference current (DA power consumption) (Note2) | I_{DREF} | No load, $V_{DD} = 2.7\text{ V}$, $f_c = 8\text{ MHz}$ | – | 200 | 300 | μA | |
| DAO output current (Note3) | I_{DAO} | $A_{VDD} = V_{DD} = 2.7\text{ V}$ $A_{VSS} = V_{SS} = 0\text{ V}$ No load 0.5 V bias when inputting 7FH | 0.4 | 5.3 | – | | |

Note 1: In linearity error measurements, the first code (00H, 01H) and the last code (FEH, FFH) are excluded.

Note 2: DA converter power consumption

Note 3: DAO current drive capability

18.6 AC Characteristics

($V_{SS} = 0\text{ V}$, $V_{DD} = 4.5\text{ to }5.5\text{ V}$, $T_{opr} = -40\text{ to }85^\circ\text{C}$)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|------------------------------|------------------|---|-------|-------|-------|------|
| Machine cycle time | t _{cy} | NORMAL1, 2 mode | 0.25 | – | 4 | μs |
| | | IDLE0, 1, 2 mode | | | | |
| | | SLOW1, 2 mode | 117.6 | – | 133.3 | |
| | | SLEEP0, 1, 2 mode | | | | |
| High level clock pulse width | t _{WCH} | For external clock operation (XIN input) | – | 31.25 | – | ns |
| Low level clock pulse width | t _{WCL} | fc = 16 MHz | | | | |
| High level clock pulse width | t _{WSH} | For external clock operation (XTIN input) | – | 15.26 | – | μs |
| Low level clock pulse width | t _{WSL} | fs = 32.768 kHz | | | | |

($V_{SS} = 0\text{ V}$, $V_{DD} = 2.7\text{ to }4.5\text{ V}$, $T_{opr} = -40\text{ to }85^\circ\text{C}$)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|------------------------------|------------------|---|-------|-------|-------|------|
| Machine cycle time | t _{cy} | NORMAL1, 2 mode | 0.5 | – | 4 | μs |
| | | IDLE0, 1, 2 mode | | | | |
| | | SLOW1, 2 mode | 117.6 | – | 133.3 | |
| | | SLEEP0, 1, 2 mode | | | | |
| High level clock pulse width | t _{WCH} | For external clock operation (XIN input) | – | 62.5 | – | ns |
| Low level clock pulse width | t _{WCL} | fc = 8 MHz | | | | |
| High level clock pulse width | t _{WSH} | For external clock operation (XTIN input) | – | 15.26 | – | μs |
| Low level clock pulse width | t _{WSL} | fs = 32.768 kHz | | | | |

Not Recommended for New

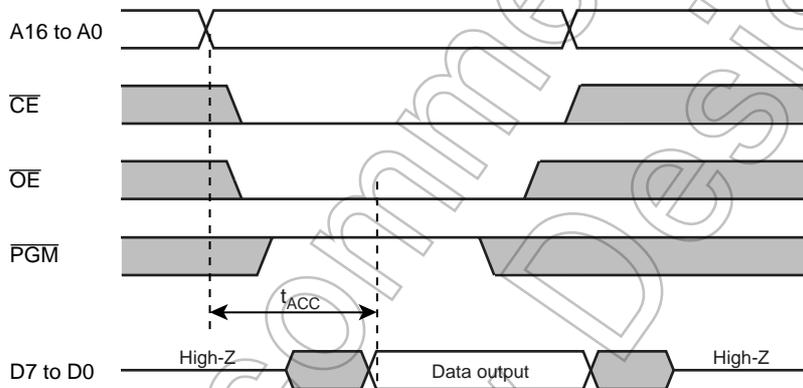
18.7 DC Characteristics, AC Characteristics (PROM mode)

18.7.1 Read operation in PROM mode

($V_{SS} = 0\text{ V}$, $T_{opr} = -40\text{ to }85^\circ\text{C}$)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|---------------------------|-----------|----------------------------------|------|--------------------|----------|------|
| High level input voltage | V_{IH4} | | 2.2 | – | V_{CC} | V |
| Low level input voltage | V_{IL4} | | 0 | – | 0.8 | |
| Power supply | V_{CC} | | 4.75 | 5.0 | 5.25 | |
| Program supply of program | V_{PP} | | | | | |
| Address access time | t_{ACC} | $V_{CC} = 5.0 \pm 0.25\text{ V}$ | – | $1.5t_{cyc} + 300$ | – | ns |

Note: $t_{cyc} = 500\text{ ns}$, $f_{CLK} = 8\text{ MHz}$

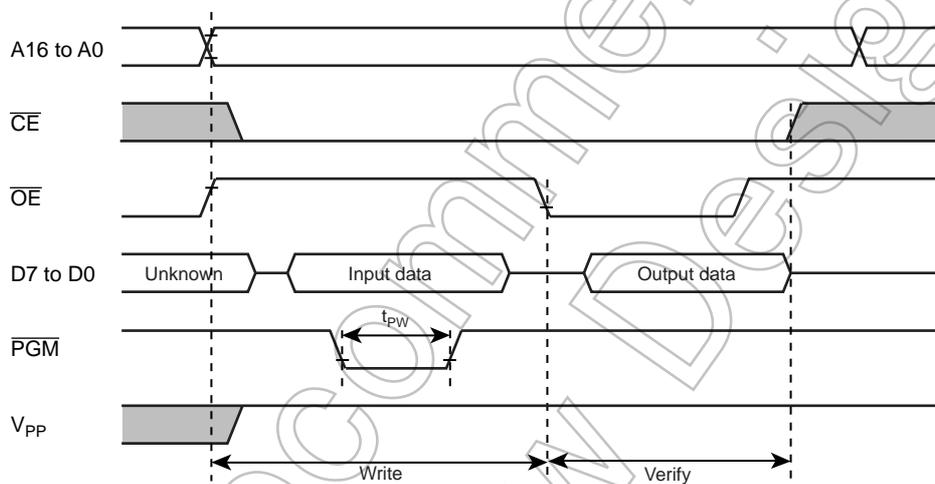


Not Recommended for New Design

18.7.2 Program operation (High-speed) ($T_{opr} = 25 \pm 5^{\circ}\text{C}$)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|-------------------------------------|-----------|-------------------------|-------|-------|----------|------|
| High level input voltage | V_{IH4} | | 2.2 | – | V_{CC} | V |
| Low level input voltage | V_{IL4} | | 0 | – | 0.8 | |
| Power supply | V_{CC} | | 6.0 | 6.25 | 6.5 | |
| Program supply of program | V_{PP} | | 12.5 | 12.75 | 13.0 | |
| Pulse width of initializing program | t_{PW} | $V_{CC} = 6.0\text{ V}$ | 0.095 | 0.1 | 0.105 | ms |

High-speed program writing

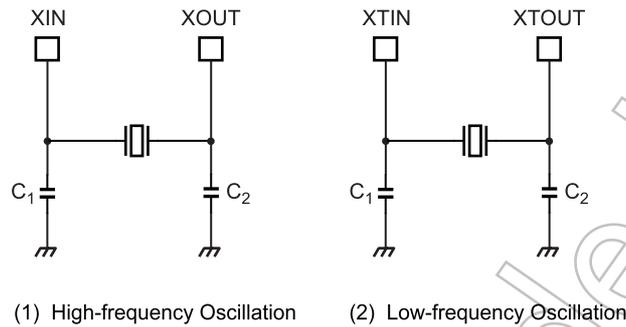


Note 1: The power supply of V_{PP} (12.75 V) must be set power-on at the same time or the later time for a power supply of V_{CC} and must be clear power-on at the same time or early time for a power supply of V_{CC} .

Note 2: The pulling up/down device causes a damage for the device while the voltage supplied to device (condition of especially $V_{PP} = 12.75\text{ V} \pm 0.25\text{ V}$). Do not pull up/down at programming.

Note 3: Use the recommended adapter and mode.
Using other than the above condition may cause the trouble of the writing.

18.8 Recommended Oscillating Conditions



Note 1: To ensure stable oscillation, the resonator position, load capacitance, etc. must be appropriate. Because these factors are greatly affected by board patterns, please be sure to evaluate operation on the board on which the device will actually be mounted.

Note 2: For the resonators to be used with Toshiba microcontrollers, we recommend ceramic resonators manufactured by Murata Manufacturing Co., Ltd.

For details, please visit the website of Murata at the following URL:

<http://www.murata.com>

18.9 Handling Precaution

- The solderability test conditions for lead-free products (indicated by the suffix G in product name) are shown below.

1. When using the Sn-37Pb solder bath
 - Solder bath temperature = 230 °C
 - Dipping time = 5 seconds
 - Number of times = once
 - R-type flux used
2. When using the Sn-3.0Ag-0.5Cu solder bath
 - Solder bath temperature = 245 °C
 - Dipping time = 5 seconds
 - Number of times = once
 - R-type flux used

Note: The pass criterion of the above test is as follows:

Solderability rate until forming $\geq 95\%$

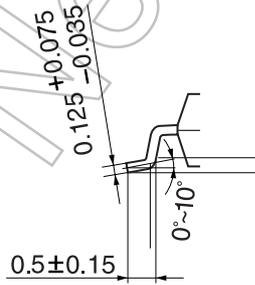
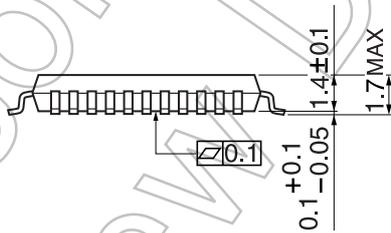
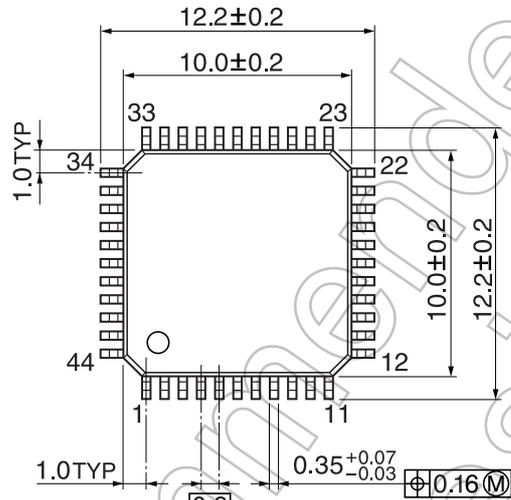
- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.

Not Recommended
for New Design

19. Package Dimensions

LQFP44-P-1010-0.80A Rev 02

Unit: mm



Not Recommended for New Design

Not Recommended
for New Design

This is a technical document that describes the operating functions and electrical specifications of the 8-bit microcontroller series TLCS-870/C (LSI).

Toshiba provides a variety of development tools and basic software to enable efficient software development.

These development tools have specifications that support advances in microcomputer hardware (LSI) and can be used extensively. Both the hardware and software are supported continuously with version updates.

The recent advances in CMOS LSI production technology have been phenomenal and microcomputer systems for LSI design are constantly being improved. The products described in this document may also be revised in the future. Be sure to check the latest specifications before using.

Toshiba is developing highly integrated, high-performance microcomputers using advanced MOS production technology and especially well proven CMOS technology.

We are prepared to meet the requests for custom packaging for a variety of application areas. We are confident that our products can satisfy your application needs now and in the future.

Not Recommended
for New Design

