TOSHIBA

8 Bit Microcontroller

TLCS-870/C Series

TMP86CS28FG

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	
2006/12/5	1	First Release
2006/12/14	2	Contents Revised
2007/7/21	3	Contents Revised

Table of Contents

TMP86CS28FG

2. Operational Description

3. Interrupt Control Circuit

4. Special Function Register (SFR)

5. I/O Ports

6. Watchdog Timer (WDT)

7. Time Base Timer (TBT)

8. 16-Bit TimerCounter (TC10,TC11)

9. 8-Bit TimerCounter (TC3, TC4)

10. 8-Bit TimerCounter (TC5, TC6)

11. Synchronous Serial Interface (SIO)

12. Asynchronous Serial interface (UART1)

13. Asynchronous Serial interface (UART0)

14. 10-bit AD Converter (ADC)

15. Key-on Wakeup (KWU)

16. LCD Driver

17. Input/Output Circuitry

18. Electrical Characteristics

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

CMOS 8-Bit Microcontroller

TMP86CS28FG

Product No.	ROM (MaskROM)	RAM	Package	FLASH MCU	Emulation Chip
TMP86CS28FG	61440 bytes	2048 bytes	QFP80-P-1420-0.80B	TMP86FS28FG	TMP86C989XB

1.1 Features

1. 8-bit single chip microcomputer TLCS-870/C series
   - Instruction execution time :

     0.25 μs (at 16 MHz)

     122 μs (at 32.768 kHz)
   - 132 types & 731 basic instructions
2. 23interrupt sources (External : 6 Internal : 17)
3. Input / Output ports (62 pins)
4. Watchdog Timer
5. Prescaler
   - Time base timer
   - Divider output function
6. 16-bit timer counter: 2 ch
   - Timer, External trigger, Window, Pulse width measurement,

     Event counter, Programmable pulse generate (PPG) modes
7. 8-bit timer counter : 4 ch
   - Timer, Event counter, Programmable divider output (PDO),

     Pulse width modulation (PWM) output,

     Programmable pulse generation (PPG) modes
8. 8-bit UART/SIO: 1 ch
9. 8-bit UART : 1 ch

- 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 malfunctionor 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

10. 10-bit successive approximation type AD converter
    - Analog input: 8 ch
11. Key-on wakeup : 4 ch
12. LCD driver/controller

    Built-in voltage booster for LCD driver With display memory

    LCD direct drive capability (MAX 40 seg × 4 com)

    1/4,1/3,1/2duties or static drive are programmably selectable
13. Clock operation

    Single clock mode

    Dual clock mode
14. 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 TBTCR<TBTCK>.

    IDLE1 mode: CPU stops and peripherals operate using high frequency clock. Release by interruputs(CPU restarts).

    IDLE2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interruputs. (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 TBTCR<TBTCK>.

    SLEEP1 mode: CPU stops, and peripherals operate using low frequency clock. Release by interruput.(CPU restarts).

    SLEEP2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interruput.
15. Wide operation voltage:

    2.7 V to 5.5 V at 8MHz /32.768 kHz

    4.0 V to 5.5 V at 16 MHz /32.768 kHz

1.2 Pin Assignment

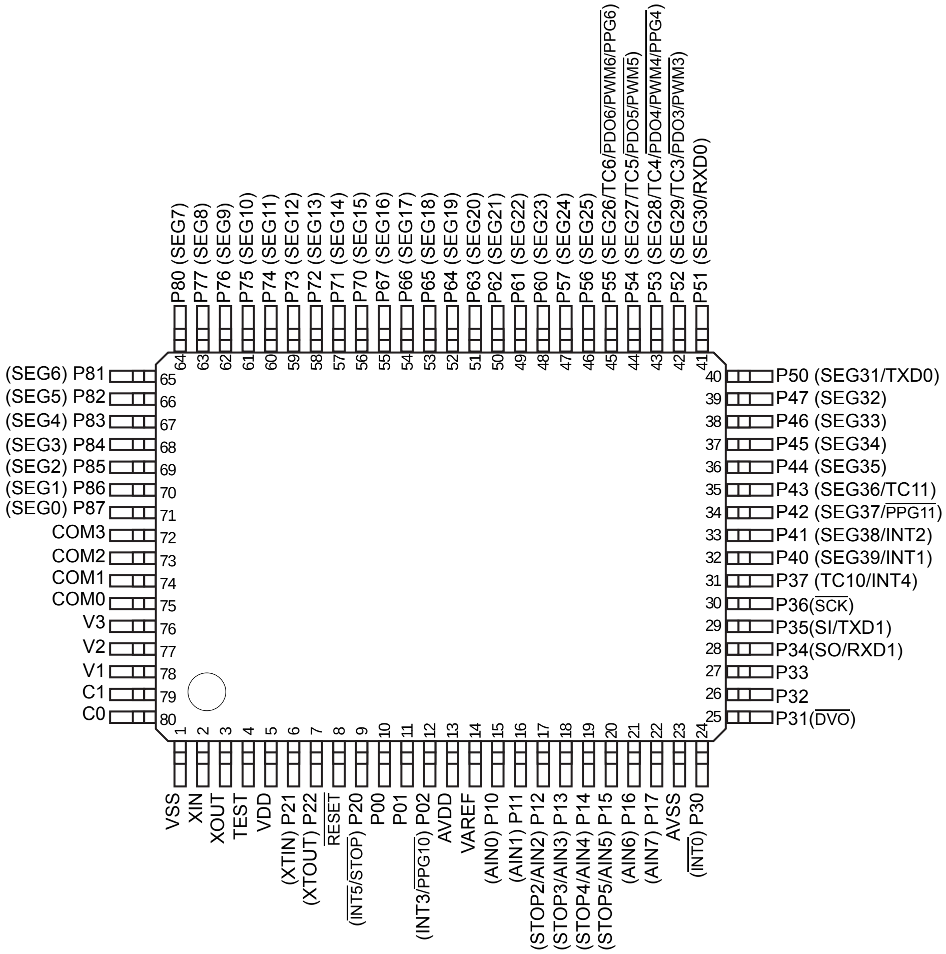


Figure 1-1 Pin Assignment

1.3 Block Diagram

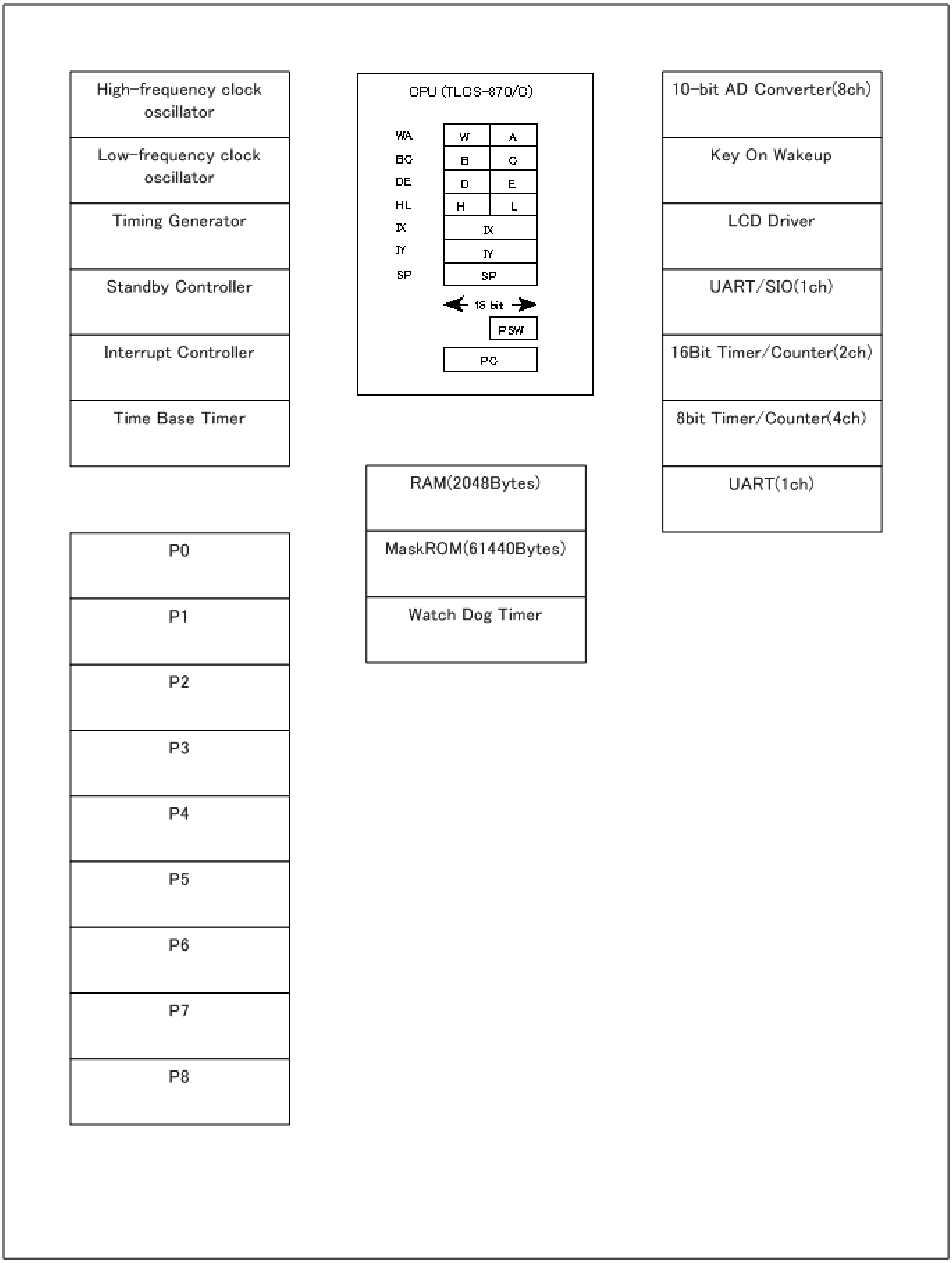


Figure 1-2 Block Diagram

1.4 Pin Names and Functions

Table 1-1 Pin Names and Functions(1/4)

Pin Name	Pin Number	Input/Output	Functions
P02 $\overline{\text{PPG10}}$ INT3	12	IO O I	PORT02 PPG10 output External interrupt 3 input
P01	11	IO	PORT01
P00	10	IO	PORT00
P17 AIN7	22	IO I	PORT17 Analog Input7
P16 AIN6	21	IO I	PORT16 Analog Input6
P15 AIN5 STOP5	20	IO I I	PORT15 Analog Input5 STOP5 input
P14 AIN4 STOP4	19	IO I I	PORT14 Analog Input4 STOP4 input
P13 AIN3 STOP3	18	IO I I	PORT13 Analog Input3 STOP3 input
P12 AIN2 STOP2	17	IO I I	PORT12 Analog Input2 STOP2 input
P11 AIN1	16	IO I	PORT11 Analog Input1
P10 AIN0	15	IO I	PORT10 Analog Input0
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 $\overline{\text{STOP}}$ $\overline{\text{INT5}}$	9	IO I I	PORT20 STOP mode release signal input External interrupt 5 input
P37 TC10 INT4	31	IO I I	PORT37 TC10 input External interrupt 4 input
P36 $\overline{\text{SCK}}$	30	IO IO	PORT36 Serial Clock I/O
P35 SI TXD1	29	IO I O	PORT35 Serial Data Input UART data output 1
P34 SO RXD1	28	IO O I	PORT34 Serial Data Output UART data input 1
P33	27	IO	PORT33

Table 1-1 Pin Names and Functions(2/4)

Pin Name	Pin Number	Input/Output	Functions
P32	26	IO	PORT32
P31 $\overline{\text{DVO}}$	25	IO O	PORT31 Divider Output
P30 $\overline{\text{INT0}}$	24	IO I	PORT30 External interrupt 0 input
P47 SEG32	39	IO O	PORT47 LCD segment output 32
P46 SEG33	38	IO O	PORT46 LCD segment output 33
P45 SEG34	37	IO O	PORT45 LCD segment output 34
P44 SEG35	36	IO O	PORT44 LCD segment output 35
P43 SEG36 TC11	35	IO O I	PORT43 LCD segment output 36 TC11 input
P42 SEG37 $\overline{\text{PPG11}}$	34	IO O O	PORT42 LCD segment output 37 PPG11 output
P41 SEG38 INT2	33	IO O I	PORT41 LCD segment output 38 External interrupt 2 input
P40 SEG39 INT1	32	IO O I	PORT40 LCD segment output 39 External interrupt 1 input
P57 SEG24	47	IO O	PORT57 LCD segment output 24
P56 SEG25	46	IO O	PORT56 LCD segment output 25
P55 SEG26 TC6 $\overline{\text{PDO6/PWM6/PPG6}}$	45	IO O I O	PORT55 LCD segment output 26 TC6 input PDO6/PWM6/PPG6 output
P54 SEG27 TC5 $\overline{\text{PDO5/PWM5}}$	44	IO O I O	PORT54 LCD segment output 27 TC5 input PDO5/PWM5 output
P53 SEG28 TC4 $\overline{\text{PDO4/PWM4/PPG4}}$	43	IO O I O	PORT53 LCD segment output 28 TC4 input PDO4/PWM4/PPG4 output
P52 SEG29 TC3 $\overline{\text{PDO3/PWM3}}$	42	IO O I O	PORT52 LCD segment output 29 TC3 input
P51 SEG30 RXD0	41	IO O I	PORT51 LCD segment output 30 UART data input 0
P50 SEG31 TXD0	40	IO O O	PORT50 LCD segment output 31 UART data output 0

Table 1-1 Pin Names and Functions(3/4)

Pin Name	Pin Number	Input/Output	Functions
P67 SEG16	55	IO O	PORT67 LCD segment output 16
P66 SEG17	54	IO O	PORT66 LCD segment output 17
P65 SEG18	53	IO O	PORT65 LCD segment output 18
P64 SEG19	52	IO O	PORT64 LCD segment output 19
P63 SEG20	51	IO O	PORT63 LCD segment output 20
P62 SEG21	50	IO O	PORT62 LCD segment output 21
P61 SEG22	49	IO O	PORT61 LCD segment output 22
P60 SEG23	48	IO O	PORT60 LCD segment output 23
P77 SEG8	63	IO O	PORT77 LCD segment output 8
P76 SEG9	62	IO O	PORT76 LCD segment output 9
P75 SEG10	61	IO O	PORT75 LCD segment output 10
P74 SEG11	60	IO O	PORT74 LCD segment output 11
P73 SEG12	59	IO O	PORT73 LCD segment output 12
P72 SEG13	58	IO O	PORT72 LCD segment output 13
P71 SEG14	57	IO O	PORT71 LCD segment output 14
P70 SEG15	56	IO O	PORT70 LCD segment output 15
P87 SEG0	71	IO O	PORT87 LCD segment output 0
P86 SEG1	70	IO O	PORT86 LCD segment output 1
P85 SEG2	69	IO O	PORT85 LCD segment output 2
P84 SEG3	68	IO O	PORT84 LCD segment output 3
P83 SEG4	67	IO O	PORT83 LCD segment output 4
P82 SEG5	66	IO O	PORT82 LCD segment output 5
P81 SEG6	65	IO O	PORT81 LCD segment output 6

Table 1-1 Pin Names and Functions(4/4)

Pin Name	Pin Number	Input/Output	Functions
P80 SEG7	64	IO O	PORT80 LCD segment output 7
COM3	72	O	LCD common output 3
COM2	73	O	LCD common output 2
COM1	74	O	LCD common output 1
COM0	75	O	LCD common output 0
V3	76	I	LCD voltage booster pin
V2	77	I	LCD voltage booster pin
V1	78	I	LCD voltage booster pin
C1	79	I	LCD voltage booster pin
C0	80	I	LCD voltage booster pin
XIN	2	I	Resonator connecting pins for high-frequency clock
XOUT	3	O	Resonator connecting pins for high-frequency clock
$\overline{\text{RESET}}$	8	I	Reset signal
TEST	4	I	Test pin for out-going test. Normally, be fixed to low.
VAREF	14	I	Analog Base Voltage Input Pin for A/D Conversion
AVDD	13	I	Analog Power Supply
AVSS	23	I	Analog Power Supply
VDD	5	I	Power Supply
VSS	1	I	0(GND)

2. Operational Description

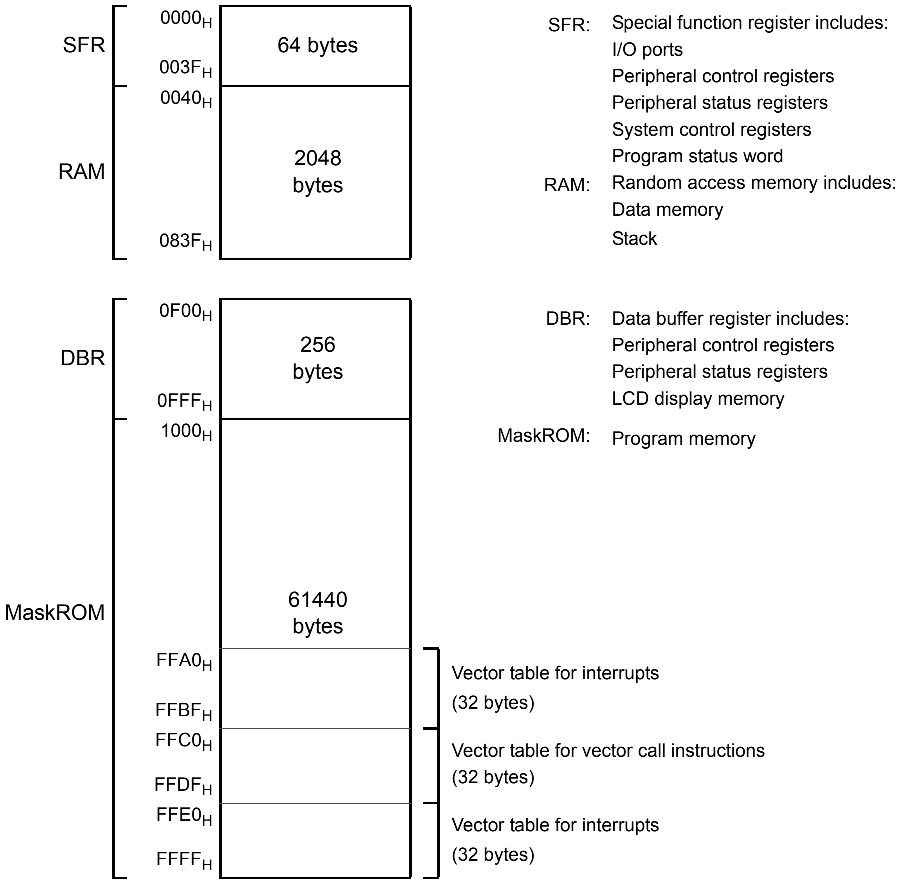
2.1 CPU Core Functions

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 TMP86CS28FG memory is composed MaskROM, RAM, DBR(Data buffer register) and SFR(Special function register). They are all mapped in 64-Kbyte address space. Figure 2-1 shows the TMP86CS28FG memory address map.



SFR
0000H
64 bytes
003FH
0040H
RAM
2048 bytes
083FH
DBR
0F00H
256 bytes
0FFFH
1000H
MaskROM
61440 bytes
FFA0H
FFBFH
FFC0H
FFDFH
FFE0H
FFFFH
Vector table for interrupts (32 bytes)
Vector table for vector call instructions (32 bytes)
Vector table for interrupts (32 bytes)
SFR: Special function register includes:
I/O ports
Peripheral control registers
Peripheral status registers
System control registers
Program status word
RAM: Random access memory includes:
Data memory
Stack
DBR: Data buffer register includes:
Peripheral control registers
Peripheral status registers
LCD display memory
MaskROM: Program memory

Figure 2-1 Memory Address Map

2.1.2 Program Memory (MaskROM)

The TMP86CS28FG has a 61440 bytes (Address 1000H to FFFFH) of program memory (MaskROM).

2.1.3 Data Memory (RAM)

The TMP86CS28FG has 2048 bytes (Address 0040H to 083FH) 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". (TMP86CS28FG)

```
                LD      HL, 0040H       ; Start address setup
                LD      A, H            ; Initial value (00H) setup
                LD      BC, 07FFH
    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.

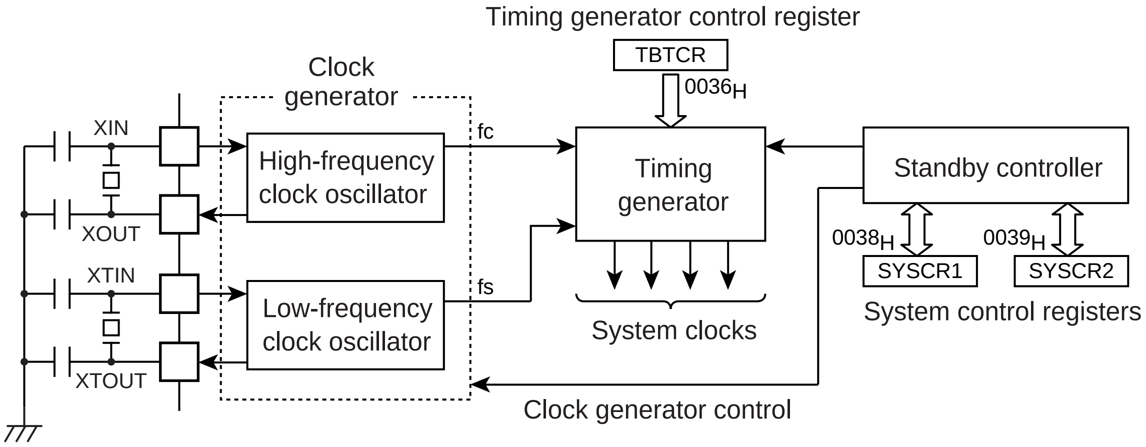


Figure 2-2 System Colck Control

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.

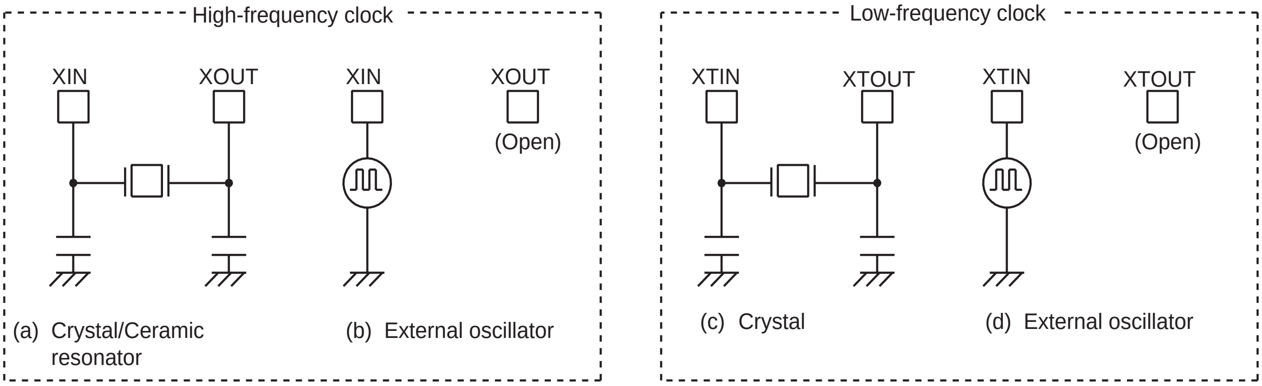


Figure 2-3 Examples of Resonator Connection

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.

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 (fc or fs). The timing generator provides the following functions.

1. Generation of main system clock
2. Generation of divider output ($\overline{\text{DVO}}$) 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
7. LCD

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, SYSCR2<SYSCK> and TBTCR<DV7CK>, that is shown in Figure 2-4. As reset and STOP mode started/canceled, the prescaler and the divider are cleared to “0”.

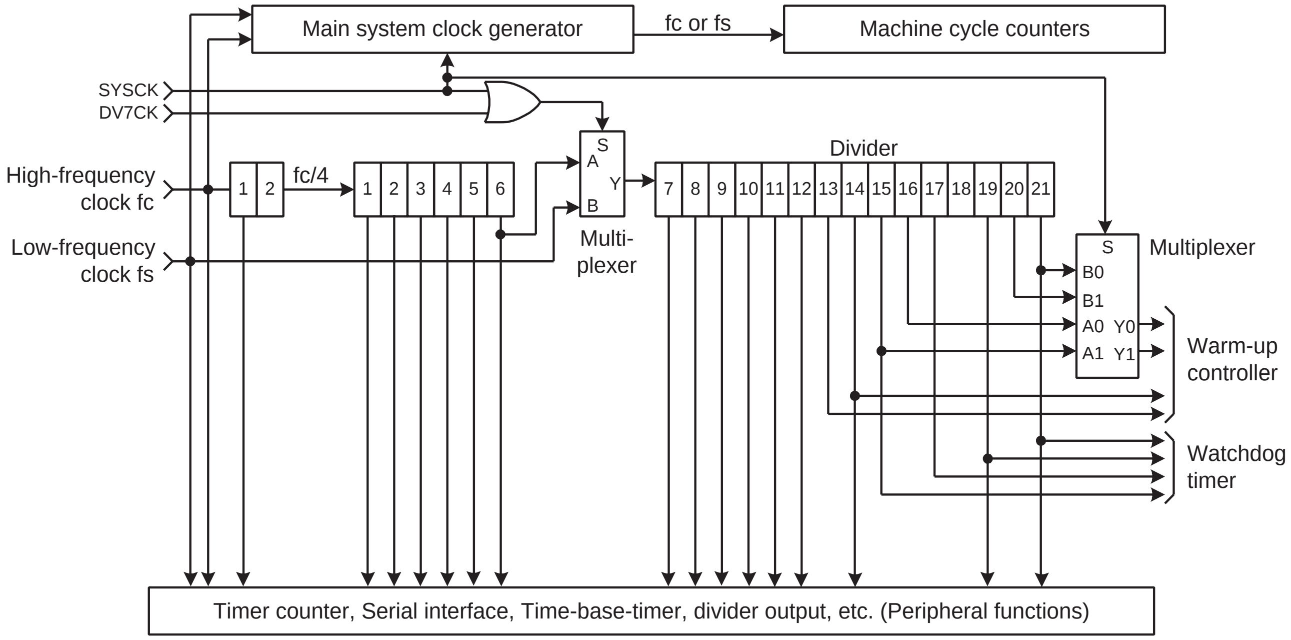


Figure 2-4 Configuration of Timing Generator

Timing Generator Control Register

TBTCR (0036H)	7	6	5	4	3	2	1	0	
	(DVOEN)	(DVOCK)		DV7CK	(TBTEN)	(TBTCK)			(Initial value: 0000 0000)

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 an “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.

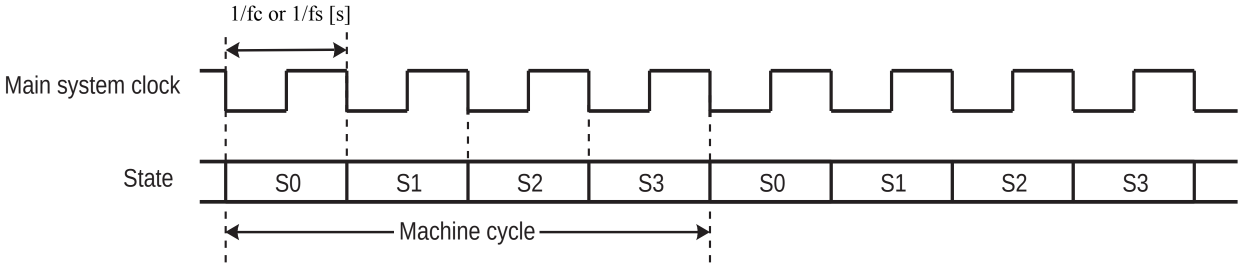


Figure 2-5 Machine Cycle

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 TMP86CS28FG 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/fc [s] in the NORMAL2 and IDLE2 modes, and 4/fs [s] (122 μs at fs = 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>.

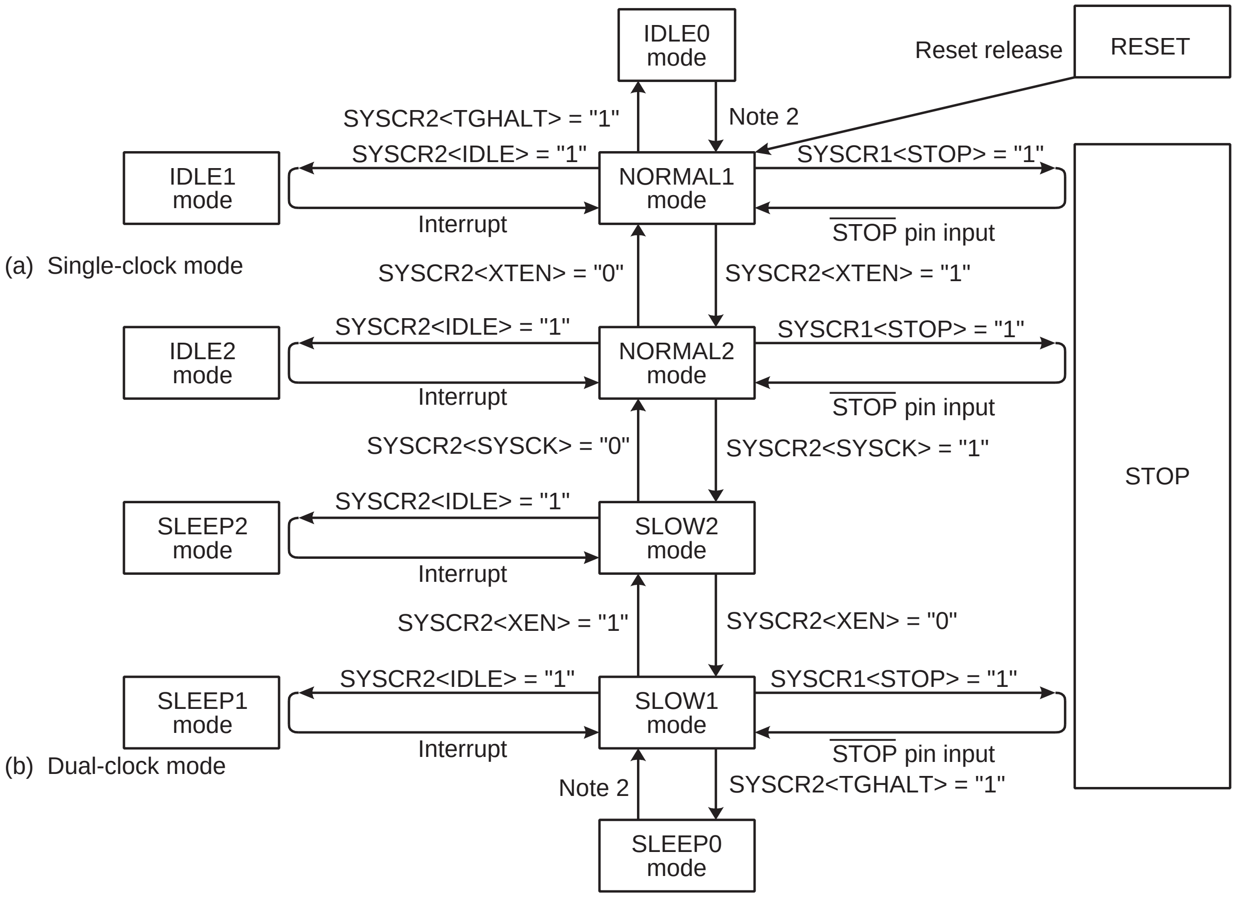
When SLEEP0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR<TBTCK>, the timing generator starts feeding the clock to all peripheral circuits.

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 $\overline{\text{STOP}}$ 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			
	IDLE0					Halt	
	STOP	Stop			Halt		–
Dual clock	NORMAL2	Oscillation	Oscillation	Operate with high frequency	Operate	Operate	4/fc [s]
	IDLE2			Halt			
	SLOW2			Operate with low frequency			4/fs [s]
	SLEEP2			Halt			
	SLOW1	Stop		Operate with low frequency			
	SLEEP1			Halt			
	SLEEP0					Halt	
	STOP		Stop		Halt		–

System Control Register 1

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

<table>
<tr><td>STOP</td><td>STOP mode start</td><td colspan="3">0: CPU core and peripherals remain active
1: CPU core and peripherals are halted (Start STOP mode)</td><td>R/W</td></tr>
<tr><td>RELM</td><td>Release method for STOP mode</td><td colspan="3">0: Edge-sensitive release
1: Level-sensitive release</td><td>R/W</td></tr>
<tr><td>RETM</td><td>Operating mode after STOP mode</td><td colspan="3">0: Return to NORMAL1/2 mode
1: Return to SLOW1 mode</td><td>R/W</td></tr>
<tr><td>OUTEN</td><td>Port output during STOP mode</td><td colspan="3">0: High impedance
1: Output kept</td><td>R/W</td></tr>
<tr><td rowspan="5">WUT</td><td rowspan="5">Warm-up time at releasing STOP mode</td><td></td><td>Return to NORMAL mode</td><td>Return to SLOW mode</td><td rowspan="5">R/W</td></tr>
<tr><td>00</td><td>3 x 2^{16}/fc</td><td>3 x 2^{13}/fs</td></tr>
<tr><td>01</td><td>2^{16}/fc</td><td>2^{13}/fs</td></tr>
<tr><td>10</td><td>3 x 2^{14}/fc</td><td>3 x 2^{6}/fs</td></tr>
<tr><td>11</td><td>2^{14}/fc</td><td>2^{6}/fs</td></tr>
</table>

Note 1: Always set RETM to “0” when transiting from NORMAL mode to STOP mode. Always set RETM to “1” when transiting from SLOW mode to STOP mode.

Note 2: When STOP mode is released with $\overline{\text{RESET}}$ 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 $\overline{\text{STOP}}$ 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)	
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)	R/W
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<TBTCK>.

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 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
```

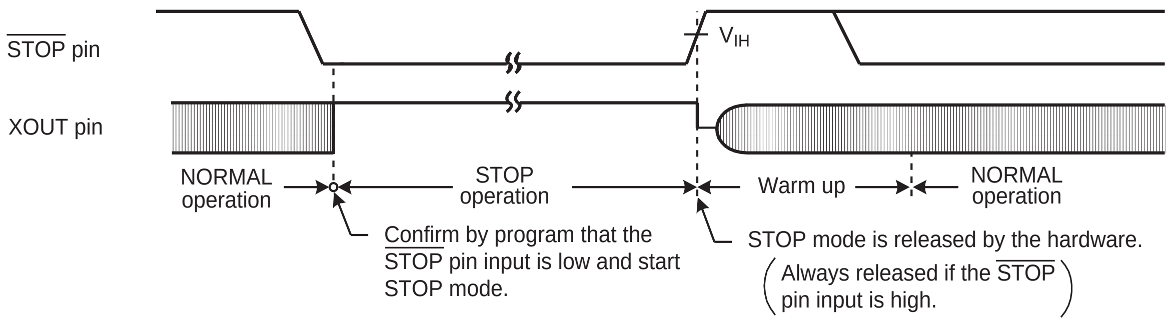

$\overline{\text{STOP}}$ pin
XOUT pin
V_{IH}
NORMAL operation
STOP operation
Warm up
NORMAL operation
Confirm by program that the $\overline{\text{STOP}}$ pin input is low and start STOP mode.
STOP mode is released by the hardware.
(Always released if the $\overline{\text{STOP}}$ pin input is high.)

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
```

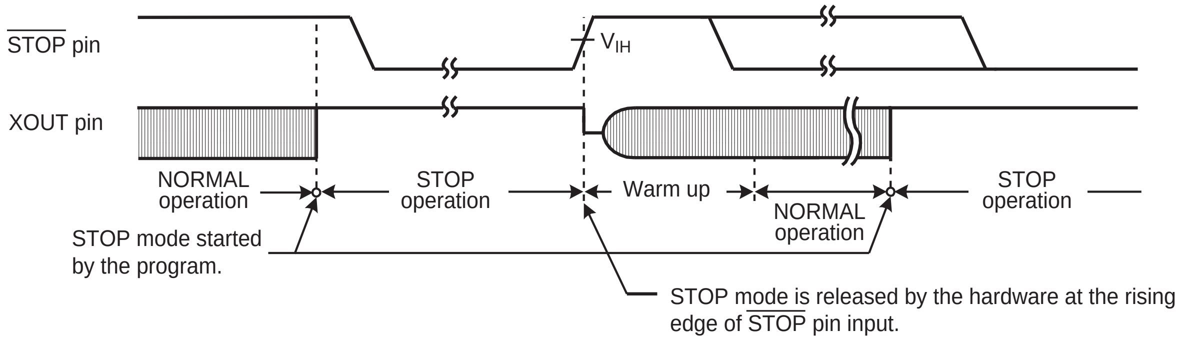

$\overline{\text{STOP}}$ pin
XOUT pin
V_{IH}
NORMAL operation
STOP operation
Warm up
NORMAL operation
STOP operation
STOP mode started by the program.
STOP mode is released by the hardware at the rising edge of $\overline{\text{STOP}}$ pin input.

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".

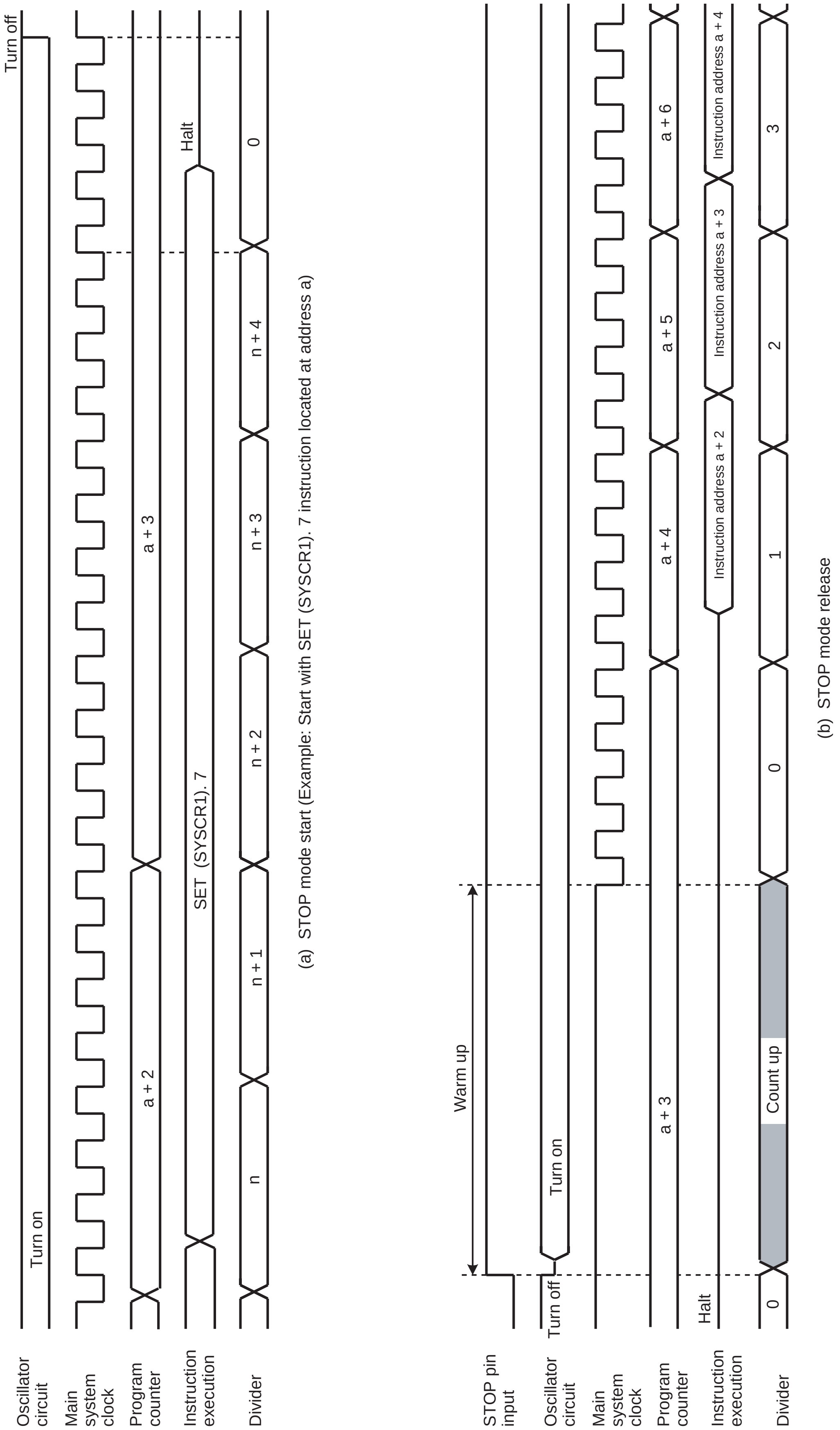
Note 2: STOP mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin, which immediately performs the normal reset operation.

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 $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ pin drops below the non-inverting high-level input voltage (Hysteresis input).

Table 2-2 Warm-up Time Example (at fc = 16.0 MHz, fs = 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.



(a) STOP mode start (Example: Start with SET (SYSCR1). 7 instruction located at address a)

(b) STOP mode release

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.

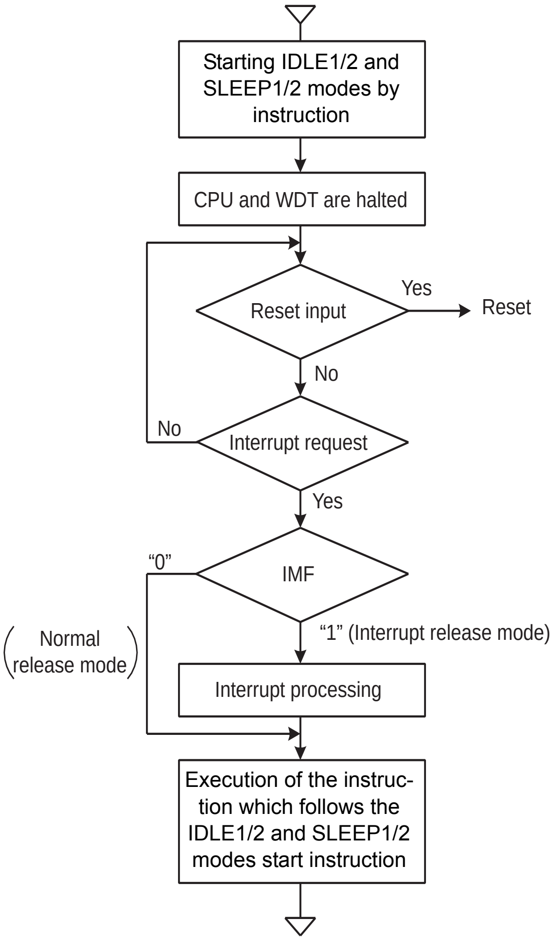

Starting IDLE1/2 and SLEEP1/2 modes by instruction
CPU and WDT are halted
Reset input
Yes
Reset
No
No
Interrupt request
Yes
"0"
IMF
"1" (Interrupt release mode)
(Normal release mode)
Interrupt processing
Execution of the instruction which follows the IDLE1/2 and SLEEP1/2 modes start instruction

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.

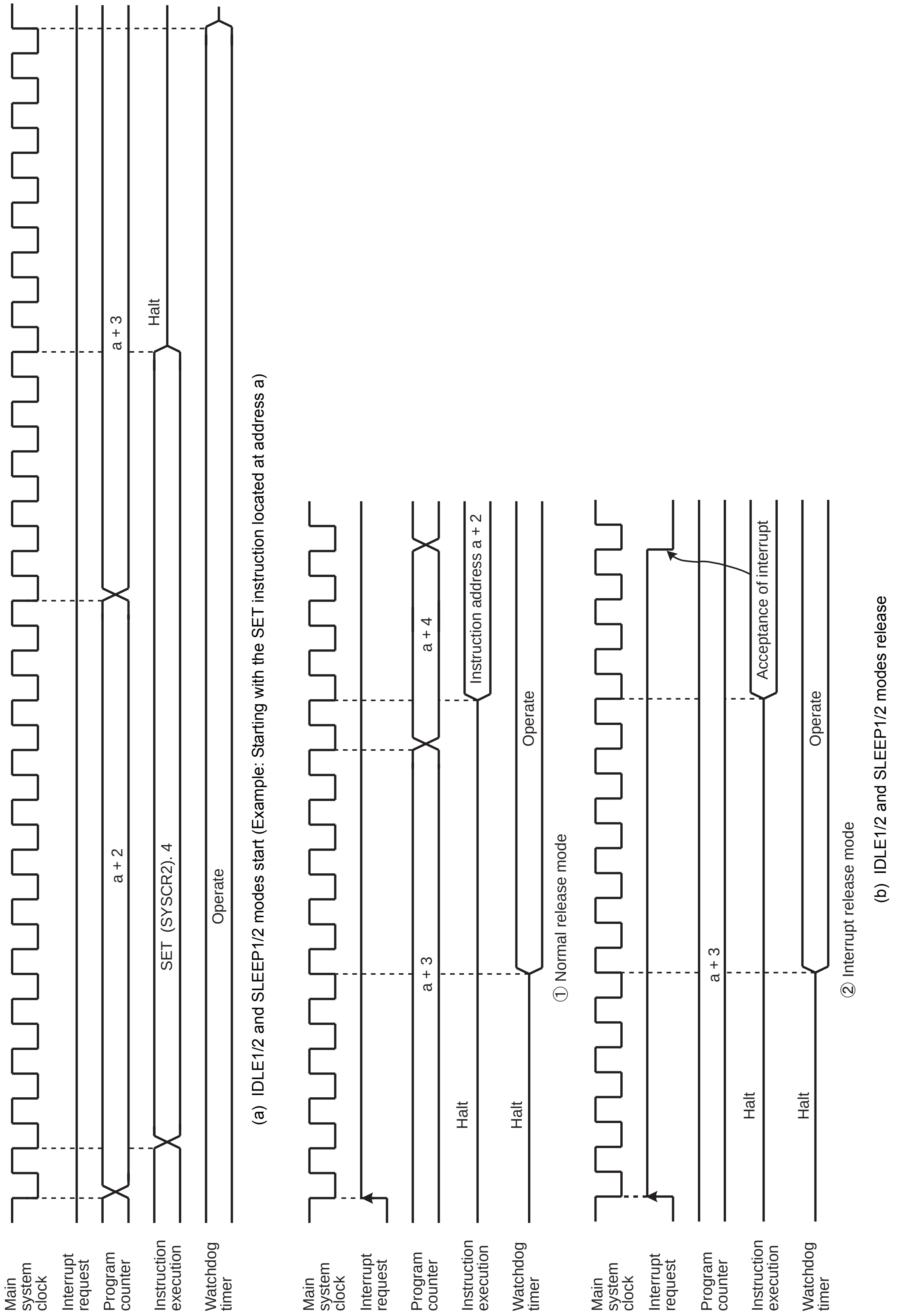
(1) Normal release mode (IMF = “0”)

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.



(a) IDLE1/2 and SLEEP1/2 modes start (Example: Starting with the SET instruction located at address a)

① Normal release mode

② Interrupt release mode

(b) IDLE1/2 and SLEEP1/2 modes release

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 (TBTCR). 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.

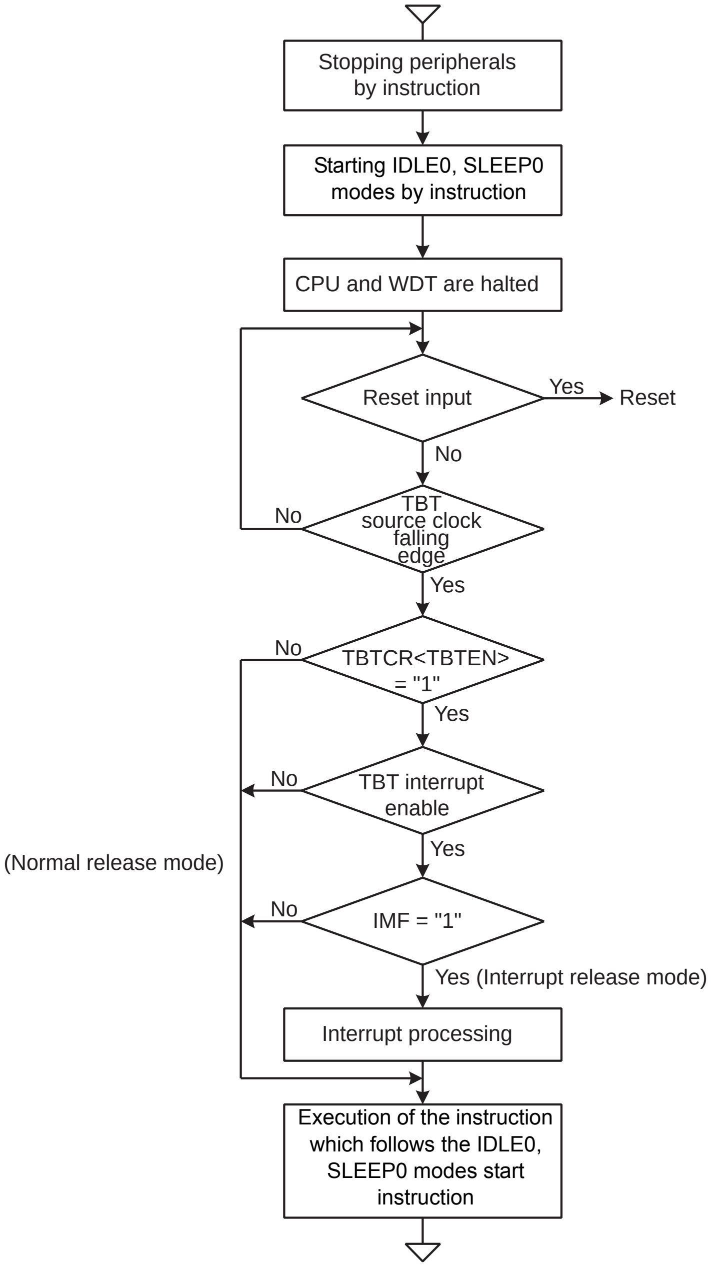

Stopping peripherals by instruction
Starting IDLE0, SLEEP0 modes by instruction
CPU and WDT are halted
Reset input
Yes
Reset
No
TBT source clock falling edge
No
Yes
TBTCR<TBTEN> = "1"
No
Yes
TBT interrupt enable
No
Yes
(Normal release mode)
IMF = "1"
No
Yes (Interrupt release mode)
Interrupt processing
Execution of the instruction which follows the IDLE0, SLEEP0 modes start instruction

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”)

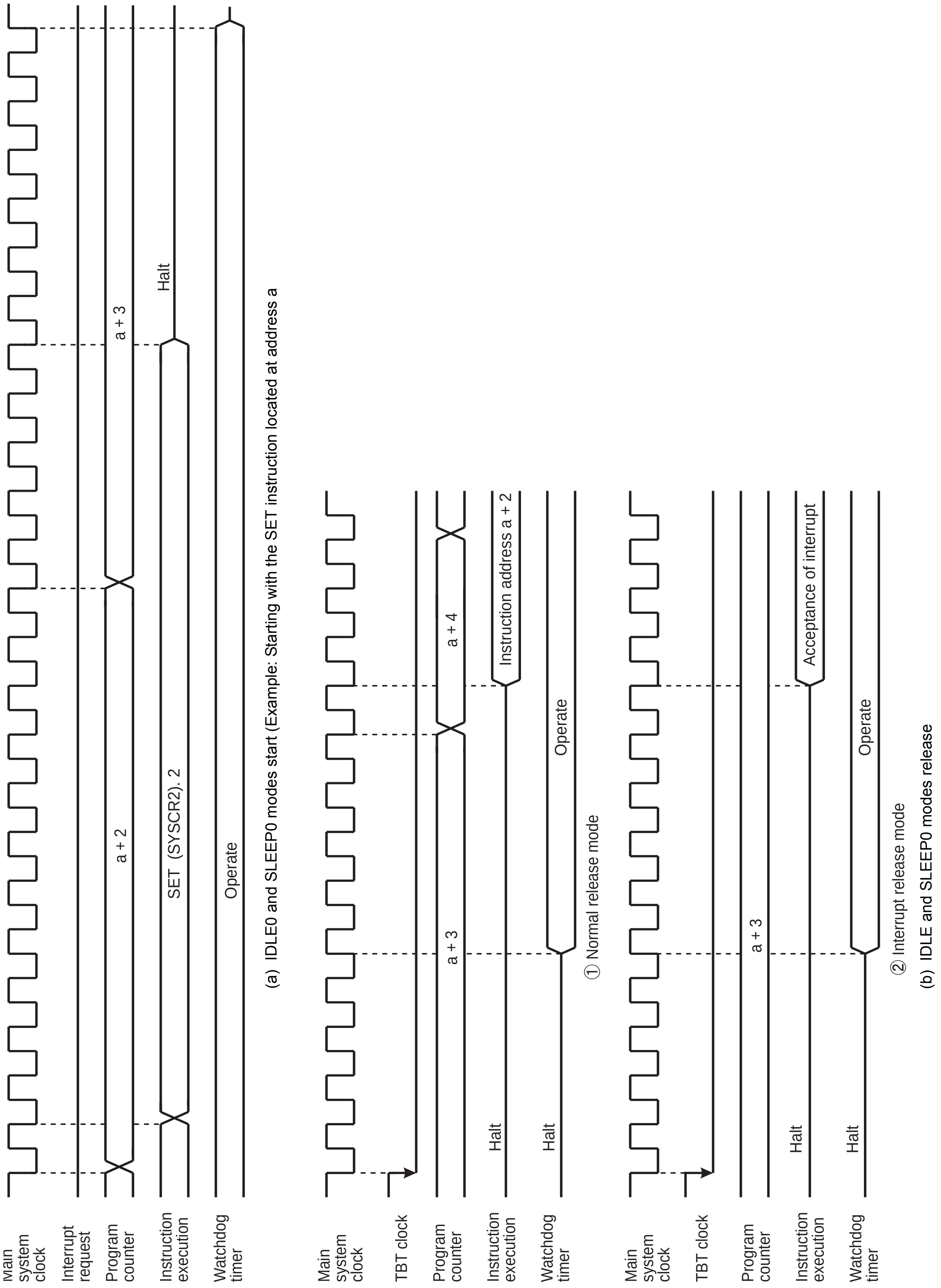
IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK>. After the falling edge is detected, the program operation is resumed from the instruction following the IDLE0 and SLEEP0 modes start instruction. Before starting the IDLE0 or SLEEP0 mode, when 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.



(a) IDLE0 and SLEEP0 modes start (Example: Starting with the SET instruction located at address a

① Normal release mode

② Interrupt release mode

(b) IDLE and SLEEP0 modes release

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)
```

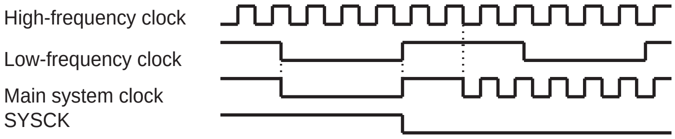
Example 2 :Switching to the SLOW1 mode after low-frequency clock has stabilized.

```
            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     (EIRE). 5            ; 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 continiued 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     (EIRE). 5           ; 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
```

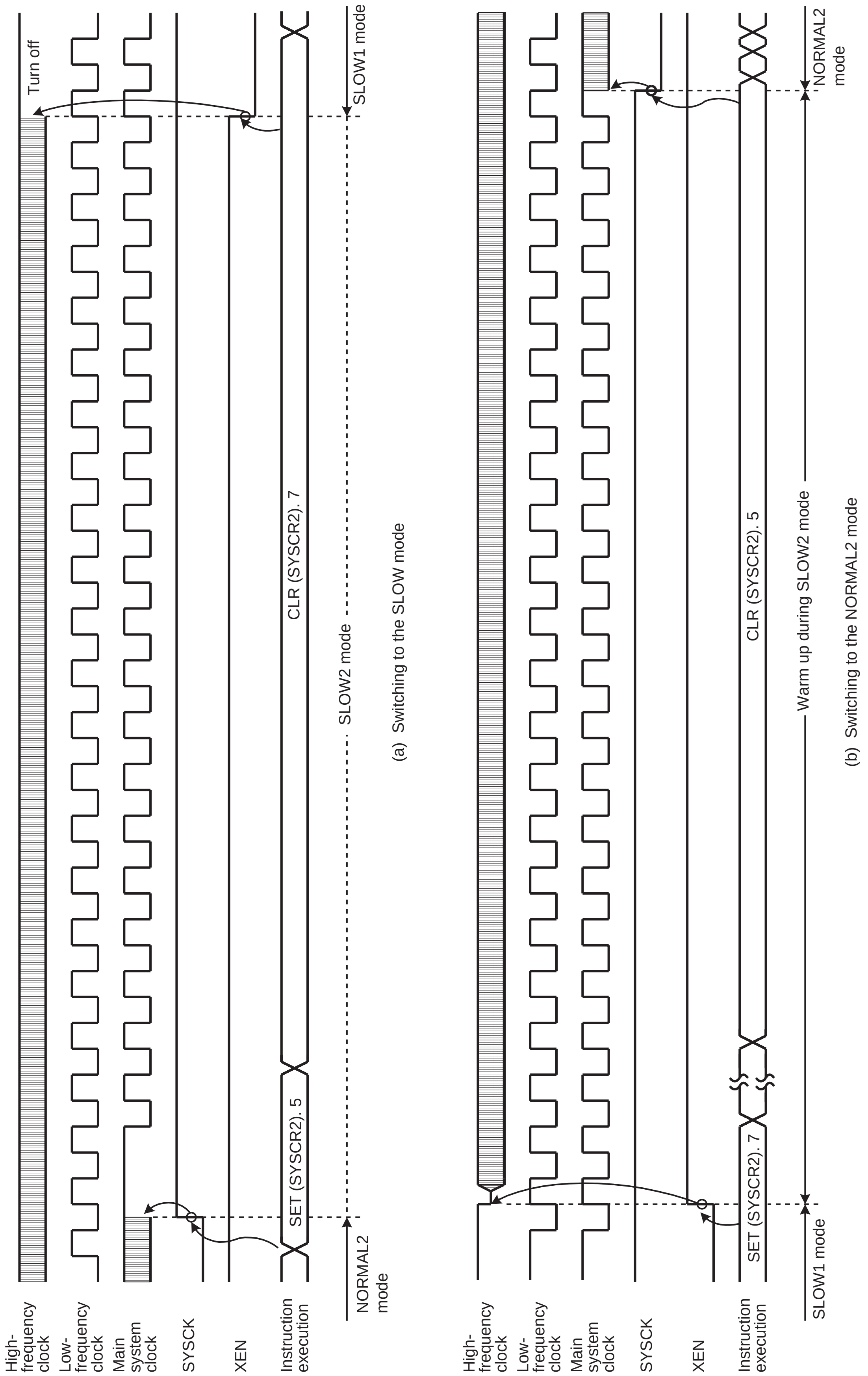


Figure 2-14 Switching between the NORMAL2 and SLOW Modes

2.3 Reset Circuit

The TMP86CS28FG 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/fc[s].

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/fc[s] (1.5µs at 16.0 MHz) when power is turned on.

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		
		LCD data buffer	Not initialized
		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/fc [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.

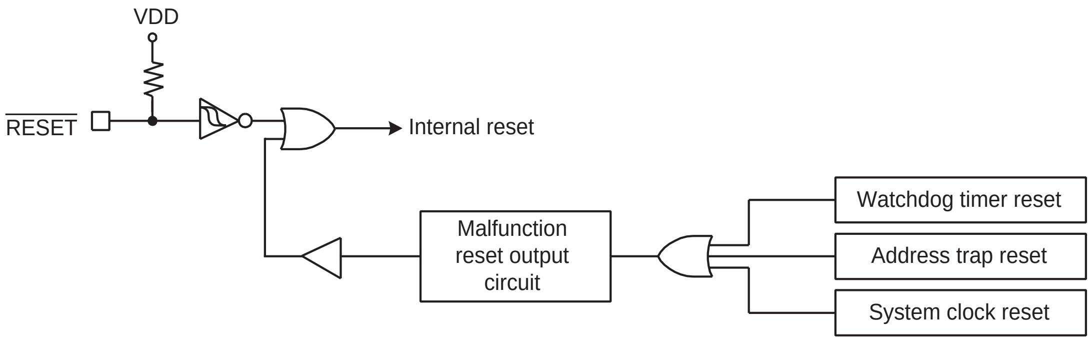
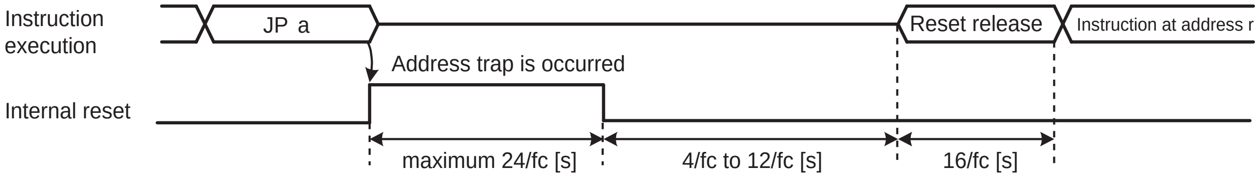


Figure 2-15 Reset Circuit

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 WDTCR1<ATAS> is set to “1”), DBR or the SFR area, address trap reset will be generated. The reset time is maximum 24/fc[s] (1.5μs at 16.0 MHz).

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, DBR or on-chip RAM (WDTCR1<ATAS> = “1”) space.
Note 2: During reset release, reset vector “r” is read out, and an instruction at address “r” is fetched and decoded.

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 SYSCR2<XEN> and SYSCR2<XTEN> simultaneously to “0”.
- In case of clearing SYSCR2<XEN> to “0”, when the SYSCR2<SYSCK> is “0”.
- In case of clearing SYSCR2<XTEN> to “0”, when the SYSCR2<SYSCK> is “1”.

The reset time is maximum 24/fc (1.5 μs at 16.0 MHz).

3. Interrupt Control Circuit

The TMP86CS28FG has a total of 23 interrupt sources excluding reset. 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	$\overline{\text{INT0}}$	IMF• EF4 = 1, INT0EN = 1	IL4	FFF6	5
External	INT1	IMF• EF5 = 1	IL5	FFF4	6
Internal	INTTBT	IMF• EF6 = 1	IL6	FFF2	7
Internal	INTTC10	IMF• EF7 = 1	IL7	FFF0	8
Internal	INTRXD0	IMF• EF8 = 1	IL8	FFEE	9
Internal	INTTXD0	IMF• EF9 = 1	IL9	FFEC	10
Internal	INTTC11	IMF• EF10 = 1	IL10	FFEA	11
External	INT2	IMF• EF11 = 1	IL11	FFE8	12
-	Reserved	IMF• EF12 = 1	IL12	FFE6	13
Internal	INTSIO	IMF• EF13 = 1	IL13	FFE4	14
-	Reserved	IMF• EF14 = 1	IL14	FFE2	15
-	Reserved	IMF• EF15 = 1	IL15	FFE0	16
-	Reserved	IMF• EF16 = 1	IL16	FFBE	17
-	Reserved	IMF• EF17 = 1	IL17	FFBC	18
-	Reserved	IMF• EF18 = 1	IL18	FFBA	19
-	Reserved	IMF• EF19 = 1	IL19	FFB8	20
Internal	INTTC3	IMF• EF20 = 1	IL20	FFB6	21
Internal	INTTC4	IMF• EF21 = 1	IL21	FFB4	22
External	INT3	IMF• EF22 = 1	IL22	FFB2	23
Internal	INTTC5	IMF• EF23 = 1	IL23	FFB0	24
Internal	INTTC6	IMF• EF24 = 1	IL24	FFAE	25
External	INT4	IMF• EF25 = 1	IL25	FFAC	26
External	$\overline{\text{INT5}}$	IMF• EF26 = 1	IL26	FFAA	27
Internal	INTRXD1	IMF• EF27 = 1	IL27	FFA8	28
Internal	INTTXD1	IMF• EF28 = 1	IL28	FFA6	29
Internal	INTADC	IMF• EF29 = 1	IL29	FFA4	30
-	Reserved	IMF• EF30 = 1	IL30	FFA2	31
-	Reserved	IMF• EF31 = 1	IL31	FFA0	32

Note 1: To use the address trap interrupt (INTATRAP), clear WDTCR1<ATOUT> to “0” (It is set for the “reset request” after reset is cancelled). For details, see “Address Trap”.

Note 2: To use the watchdog timer interrupt (INTWDT), clear WDTCR1<WDTOUT> to "0" (It is set for the "Reset request" after reset is released). For details, see "Watchdog Timer".

3.1 Interrupt latches (IL30 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 002EH, 002FH, 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 latchess

```
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 002CH, 002DH, 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 (EF30 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 (EF30 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 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 :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();
```

Interrupt Latches

(Initial value: **0*0000 000000**)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ILH,ILL (003DH, 003CH)	–	–	IL13	–	IL11	IL10	IL9	IL8	IL7	IL6	IL5	IL4	IL3	IL2		

ILH (003DH) ILL (003CH)

(Initial value: **000000 0000****)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ILD,ILE (002FH, 002EH)	–	–	IL29	IL28	IL27	IL26	IL25	IL24	IL23	IL22	IL21	IL20	–	–	–	–

ILD (002FH) ILE (002EH)

IL30 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

(Initial value: **0*0000 0000***0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIRH,EIRL (003BH, 003AH)	–	–	EF13	–	EF11	EF10	EF9	EF8	EF7	EF6	EF5	EF4				IMF

EIRH (003BH) EIRL (003AH)

(Initial value: **000000 0000****)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIRD,EIRE (002DH, 002CH)	–	–	EF29	EF28	EF27	EF26	EF25	EF24	EF23	EF22	EF21	EF20	–	–	–	–

EIRD (002DH) EIRE (002CH)

EF30 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 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.

3.3.1 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.

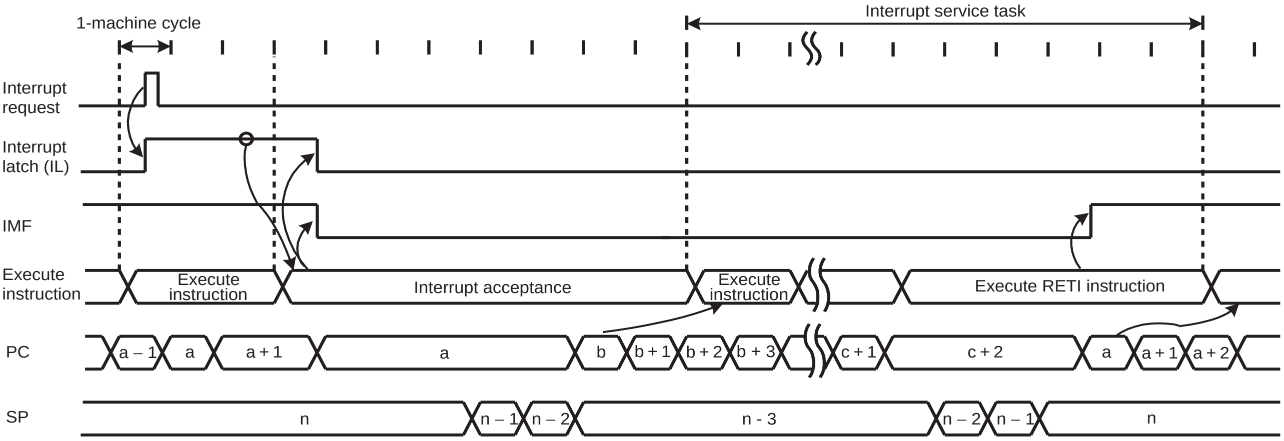
b. The interrupt latch (IL) for the interrupt source accepted is cleared to “0”.

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/fc [s] or 38/fs [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

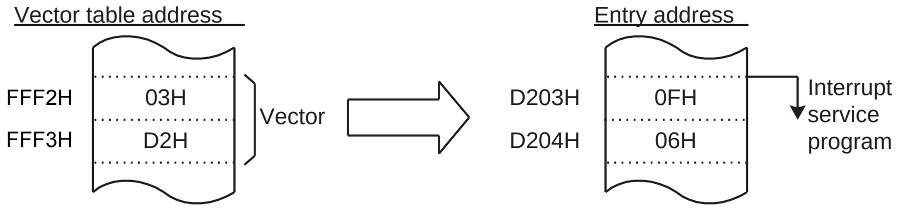


Figure 3-2 Vector table address,Entry address

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.

3.3.2 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.

3.3.2.1 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
```

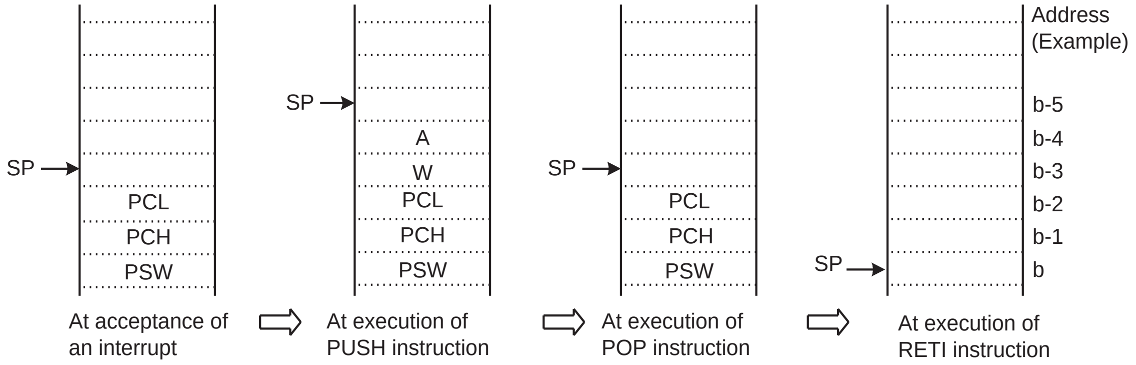

SP
PCL
PCH
PSW
At acceptance of an interrupt
SP
A
W
PCL
PCH
PSW
At execution of PUSH instruction
SP
PCL
PCH
PSW
At execution of POP instruction
SP
At execution of RETI instruction
Address (Example)
b-5
b-4
b-3
b-2
b-1
b

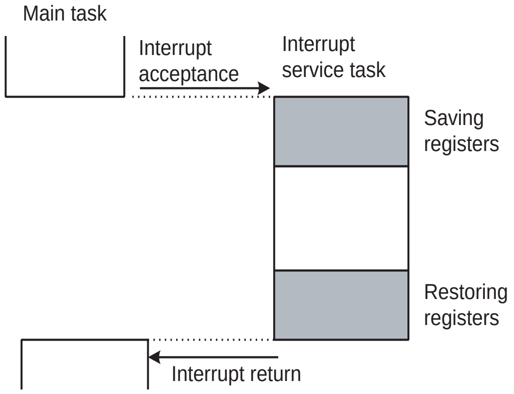
Figure 3-3 Save/store register using PUSH and POP instructions

3.3.2.2 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

3.3.3 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.

3.4 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.

3.4.1 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, DBR or SFR areas.

3.4.2 Debugging

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

3.5 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.

3.6 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).

3.7 External Interrupts

The TMP86CS28FG 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 $\overline{\text{INT0}}$/P30 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 $\overline{\text{INT0}}$/P30 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 • 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.
INT3	INT3	IMF • EF22 = 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 • EF25 = 1	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 • EF26 = 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.

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

EINTCR	7	6	5	4	3	2	1	0	
(0037H)	INT1NC	INT0EN	INT4ES		INT3ES	INT2ES	INT1ES		(Initial value: 0000 000*)

INT1NC	Noise reject time select	0: Pulses of less than 63/fc [s] are eliminated as noise 1: Pulses of less than 15/fc [s] are eliminated as noise	R/W
INT0EN	P30/$\overline{\text{INT0}}$ pin configuration	0: P30 input/output port 1: $\overline{\text{INT0}}$ pin (Port P30 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: fc: 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/fc.

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.

4. Special Function Register (SFR)

The TMP86CS28FG adopts the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function register (SFR) or the data buffer register (DBR). The SFR is mapped on address 0000H to 003FH, DBR is mapped on address 0F00H to 0FFFH.

This chapter shows the arrangement of the special function register (SFR) and data buffer register (DBR) for TMP86CS28FG.

4.1 SFR

Address	Read	Write
0000H	P0DR	
0001H	P1DR	
0002H	P2DR	
0003H	P3DR	
0004H	P4DR	
0005H	P5DR	
0006H	P6DR	
0007H	P7DR	
0008H	P8DR	
0009H	TC3CR	
000AH	TC4CR	
000BH	TC5CR	
000CH	TC6CR	
000DH	Reserved	
000EH	Reserved	
000FH	Reserved	
0010H	TC10DRAL	
0011H	TC10DRAH	
0012H	TC10DRBL	
0013H	TC10DRBH	
0014H	TC10CR	
0015H	TTREG3	
0016H	TTREG4	
0017H	TTREG5	
0018H	TTREG6	
0019H	PWREG3	
001AH	PWREG4	
001BH	PWREG5	
001CH	PWREG6	
001DH	Reserved	
001EH	Reserved	
001FH	Reserved	
0020H	TC11DRAL	
0021H	TC11DRAH	
0022H	TC11DRBL	
0023H	TC11DRBH	
0024H	TC11CR	
0025H	Reserved	

Address	Read	Write
0026H	Reserved	
0027H	Reserved	
0028H	Reserved	
0029H	Reserved	
002AH	Reserved	
002BH	P3OUTCR	
002CH	EIRE	
002DH	EIRD	
002EH	ILE	
002FH	ILD	
0030H	Reserved	
0031H	-	STOPCR
0032H	P0OUTCR	
0033H	Reserved	
0034H	-	WDTCR1
0035H	-	WDTCR2
0036H	TBTCR	
0037H	EINTCR	
0038H	SYSCR1	
0039H	SYSCR2	
003AH	EIRL	
003BH	EIRH	
003CH	ILL	
003DH	ILH	
003EH	Reserved	
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.).

4.2 DBR

Address	Read	Write
0F00H	Reserved	
: :	: :	
0F5FH	Reserved	

Address	Read	Write
0F60H	SIOBR0	
0F61H	SIOBR1	
0F62H	SIOBR2	
0F63H	SIOBR3	
0F64H	SIOBR4	
0F65H	SIOBR5	
0F66H	SIOBR6	
0F67H	SIOBR7	
0F68H	-	SIOCR1
0F69H	SIOSR	SIOCR2

Address	Read	Write
0F70H	Reserved	
: :	: :	
0F7FH	Reserved	

Address	Read	Write
0F80H	Reserved	
: :	: :	
0F9FH	Reserved	

Address	Read	Write
0FA0H	Reserved	
: :	: :	
0FBFH	Reserved	

Address	Read	Write
0FC0H	SEG1/0	
0FC1H	SEG3/2	
0FC2H	SEG5/4	
0FC3H	SEG7/6	
0FC4H	SEG9/8	
0FC5H	SEG11/10	
0FC6H	SEG13/12	
0FC7H	SEG15/14	
0FC8H	SEG17/16	
0FC9H	SEG19/18	
0FCAH	SEG21/20	
0FCBH	SEG23/22	
0FCCH	SEG25/24	
0FCDH	SEG27/26	
0FCEH	SEG29/28	
0FCFH	SEG31/30	
0FD0H	SEG33/32	
0FD1H	SEG35/34	
0FD2H	SEG37/36	
0FD3H	SEG39/38	
0FD4H	P4LCR	
0FD5H	P5LCR	
0FD6H	P6LCR	
0FD7H	P7LCR	
0FD8H	P8LCR	
0FD9H	LCDCR	
0FDAH	Reserved	
0FDBH	Reserved	
0FDCH	Reserved	
0FDDH	Reserved	
0FDEH	Reserved	
0FDFH	Reserved	

Address	Read	Write
0FE0H	ADCDR2	-
0FE1H	ADCDR1	-
0FE2H	ADCCR1	
0FE3H	ADCCR2	
0FE4H	Reserved	
0FE5H	UART0SR	UART0CR1
0FE6H	-	UART0CR2
0FE7H	RD0BUF	TD0BUF
0FE8H	UART1SR	UART1CR1
0FE9H	-	UART1CR2
0FEAH	RD1BUF	TD1BUF
0FEBH	Reserved	
0FECH	Reserved	
0FEDH	Reserved	
0FEEH	Reserved	
0FEFH	Reserved	
0FF0H	P0PRD	-
0FF1H	Reserved	
0FF2H	P2PRD	-
0FF3H	P3PRD	-
0FF4H	P4PRD	-
0FF5H	P5PRD	-
0FF6H	P6PRD	-
0FF7H	P7PRD	-
0FF8H	P8PRD	-
0FF9H	P1CR1	
0FFAH	P1CR2	
0FFBH	P4OUTCR	
0FFCH	P5OUTCR	
0FFDH	P6OUTCR	
0FFEH	P7OUTCR	
0FFFH	P8OUTCR	

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.).

5. I/O Ports

The TMP86CS28FG has 9 input/output ports (62 pins) as shown below.

Table 5-1 Port Functions

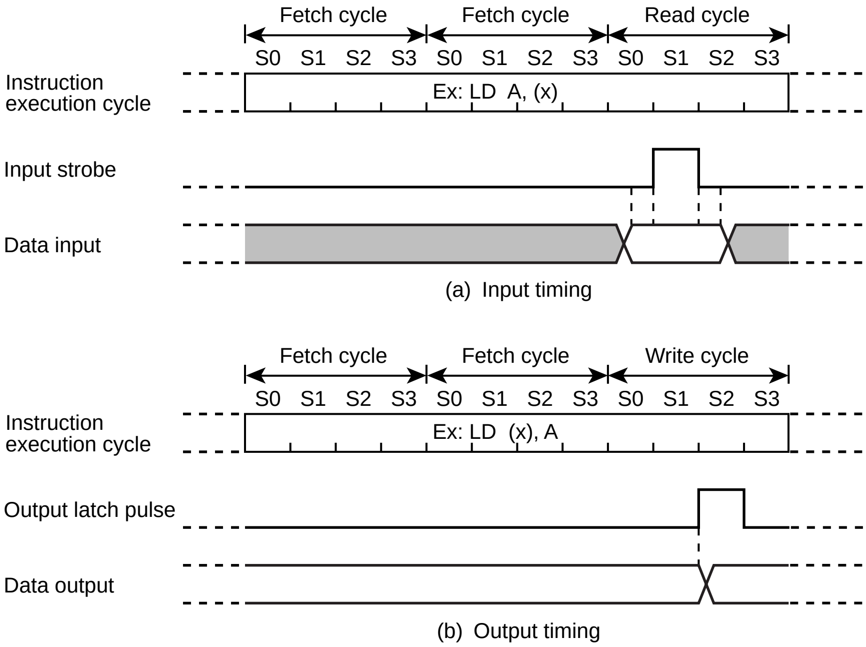
	Primary Function	Secondary Functions
Port P0	3-bit input/output port	External interrupt input, PPG output
Port P1	8-bit input/output port	Analog input, STOP mode release signal input
Port P2	3-bit input/output port	External interrupt input, low-frequency resonator connection, STOP mode release signal input
Port P3	8-bit input/output port	External interrupt input, timer/counter input, serial interface input/output, UART input/output, divider output
Port P4	8-bit input/output port	External interrupt input, timer/counter input, LCD segment output, PPG output
Port P5	8-bit input/output port	Timer/counter input/output, LCD segment output, UART input/output
Port P6	8-bit input/output port	LCD segment output
Port P7	8-bit input/output port	LCD segment output
Port P8	8-bit input/output port	LCD segment output

Table 5-2 Register List

Port	Latch	Read	Pch Control	CR1	CR2	LCD Control
P0	P0DR (0000H)	P0PRD (0FF0H)	P0OUTCR (0032H)	–	–	–
P1	P1DR (0001H)	–	–	P1CR1 (0FF9H)	P1CR2 (0FFAH)	–
P2	P2DR (0002H)	P2PRD (0FF2H)	–	–	–	–
P3	P3DR (0003H)	P3PRD (0FF3H)	P3OUTCR (002BH)	–	–	–
P4	P4DR (0004H)	P4PRD (0FF4H)	P4OUTCR (0FFBH)	–	–	P4LCR (0FD4H)
P5	P5DR (0005H)	P5PRD (0FF5H)	P5OUTCR (0FFCH)	–	–	P5LCR (0FD5H)
P6	P6DR (0006H)	P6PRD (0FF6H)	P6OUTCR (0FFDH)	–	–	P6LCR (0FD6H)
P7	P7DR (0007H)	P7PRD (0FF7H)	P7OUTCR (0FFEH)	–	–	P7LCR (0FD7H)
P8	P8DR (0008H)	P8PRD (0FF8H)	P8OUTCR (0FFFH)	–	–	P8LCR (0FD8H)

Each output port contains a latch for holding output data. All input ports do not have latches, making it necessary to externally hold input data until it is read externally or to read input data multiple times before it is processed. Figure 5-1 shows input/output timings.

External data is read from an input/output port in the S1 state of the read cycle in instruction execution. Since this timing cannot be recognized externally, transient input such as chattering must be processed by software. Data is output to an input/output port in the S2 state of the write cycle in instruction execution.



(a) Input timing

(b) Output timing

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

Figure 5-1 Input/Output Timings (Example)

5.1 Port P0 (P00 to P02)

Port P0 is a 3-bit input/output port that can also be used for external interrupt input or PPG output.

A reset initializes the output latch (P0DR) to “1” and the Pch control (P0OUTCR) to “0”.

To use a pin in Port P0 as an input port or external interrupt input, set P0DR to “1” and then set the corresponding bit in P0OUTCR to “0”.

To use a pin in Port P0 as a PPG output, set P0DR to “1”.

The output circuit of Port P0 can be set either as sink open-drain output (“0”) or CMOS output (“1”) individually for each bit in P0OUTCR.

Port P0 has a separate data input register. The output latch state can be read from the P0DR register, and the pin state can be read from the P0PRD register.

Table 5-3 Register Programming for Port P0 (P00 to P02)

Function	Programmed Value	
	P0DR	P0OUTCR
Port input, external interrupt input	“1”	“0”
Port "0" output	“0”	Set as appropriate.
Port "1" output, PPG output	“1”	

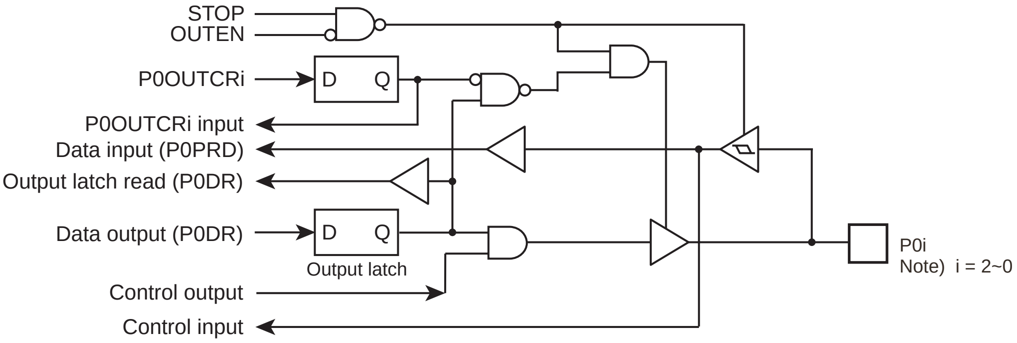


Figure 5-2 Port P0

P0DR (0000H) R/W	7	6	5	4	3	2	1	0	
						P02 $\overline{\text{PPG1}}$ INT3	P01	P00	(Initial value: **** *111)

P0OUTCR (0032H) R/W	7	6	5	4	3	2	1	0	
									(Initial value: **** *000)

P0OUTCR	Port P0 input/output control (set for each bit individually)	0: Sink open-drain output 1: CMOS output	R/W

P0PRD (0FF0H) Read only	7	6	5	4	3	2	1	0	
						P02	P01	P00	(Initial value: **** *000)

5.2 Port P1 (P10 to P17)

Port P1 is an 8-bit input/output port that can be configured as an input or an output on a bit basis. Port P1 is also used for analog input or key-on wake-up input.

The Port P1 input/output control register (P1CR1) and Port P1 input control register (P1CR2) are used to specify the function of each pin.

A reset initializes P1CR1 to “0”, P1CR2 to “1”, and the output latch (P1DR) to “0” so that Port P1 becomes an input port.

To use a pin in Port P1 as an input port, set P1CR1 to “0” and then set P1CR2 to “1”. To use a pin in Port P1 as an analog input or key-on wake-up input, set P1CR1 to “0” and then set P1CR2 to “0”.

To use a pin in Port P1 as an output port, set the corresponding bit in P1CR1 to “1”.

To read the output latch data, set P1CR1 to “1”and read P1DR. To read the pin state, set P1CR1 to “0” and P1CR2 to “1” and then read P1DR. When P1CR1 = “0” and P1CR2 = “0”, P1DR is read as “0”.

Bits not used as analog inputs are used as input/output pins. During AD conversion, however, output instructions must not be executed to ensure the accuracy of conversion results. Also, during AD conversion, do not input signals that fluctuate widely to pins near analog input pins.

Table 5-4 Register Programming for Port P1 (P10 to P17)

Function	Programmed Value		
	P1DR	P1CR1	P1CR2
Port input	*	“0”	“1”
Analog input, key-on wake-up input	*	“0”	“0”
Port “0” output	“0”	“1”	*
Port "1" output	“1”	“1”	*

Note: An asterisk (*) indicates that either “1” or “0” can be set.

Table 5-5 Values Read from P1DR according to Register Programming

Conditions		Values Read from P1DR
P1CR1	P1CR2	
“0”	“0”	“0”
“0”	“1”	Pin state
“1”	“0”	Output latch state
	“1”	

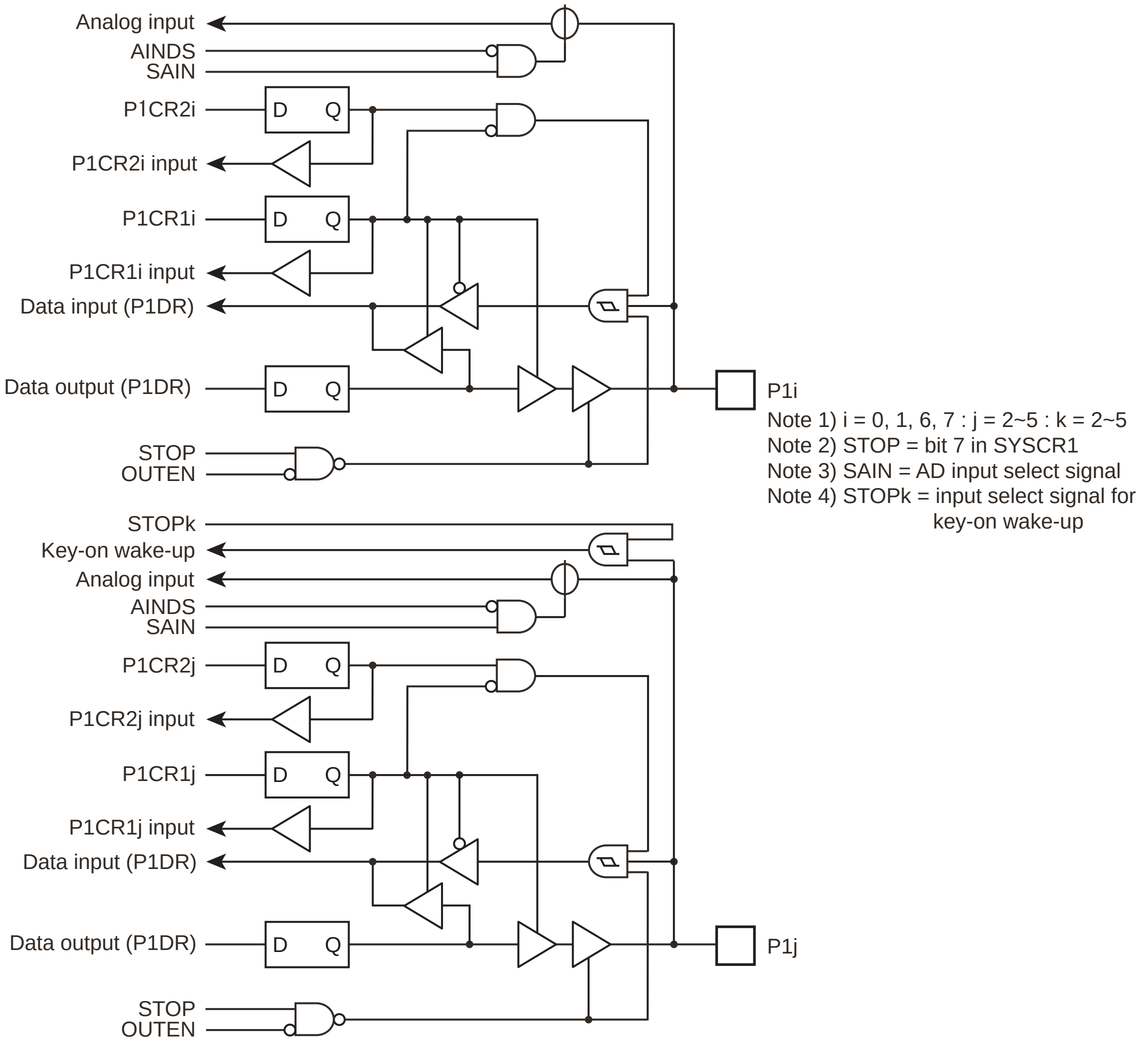


Figure 5-3 Port P1

Note 1: Pins set to input mode read the pin input data. Therefore, when both input and output modes are used in Port P1, the contents of the output latch of a pin set to input mode may be overwritten by a bit manipulation instruction.

Note 2: For a pin used as an analog input, be sure to clear the corresponding bit in P1CR2 to "0" to prevent flow-through current.

Note 3: For a pin used as an analog input, do not set P1CR1 to "1" (port output) to prevent the pin from becoming shorted with an external signal.

Note 4: Pins not used as analog inputs can be used as input/output pins. During AD conversion, however, output instructions must not be executed to ensure the accuracy of conversion results. Also, during AD conversion, do not input signals that fluctuate widely to pins near analog input pins.

P1DR (0001H) R/W

7	6	5	4	3	2	1	0	
P17 AIN7	P16 AIN6	P15 AIN5 STOP5	P14 AIN4 STOP4	P13 AIN3 STOP3	P12 AIN2 STOP2	P11 AIN1	P10 AIN0	(Initial value: 0000 0000)

P1CR1 (0FF9H)

7	6	5	4	3	2	1	0	
								(Initial value: 0000 0000)

P1CR1	Port P1 input/output control (set for each bit individually)	0: Port input, key-on wake-up input, analog input 1: Port output	R/W

P1CR2 (0FFAH)

7	6	5	4	3	2	1	0	
								(Initial value: 1111 1111)

P1CR2	Port P1 input control (set for each bit individually)	0: Analog input, key-on wake-up input 1: Port input	R/W

5.3 Port P2 (P20 to P22)

Port P2 is a 3-bit input/output port that can also be used for external interrupt input, STOP mode release signal input, or low-frequency resonator connection.

To use Port P2 as an input port or function pins, set the output latch (P2DR) to “1”. A reset initializes P2DR to “1”.

In the dual clock mode, pins P21 (XTIN) and P22 (XOUT) are connected with a low-frequency resonator (32.768 kHz). In the single clock mode, pins P21 and P22 can be used as normal input/output port pins.

It is recommended that pin P20 be used as an external interrupt input, STOP release signal input, or input port. (When P20 is used as an output port, the interrupt latch is set on the falling edge of the output pulse.)

Port P2 has a separate data input register. The output latch state can be read from the P2DR register, and the pin state can be read from the P2PRD register.

When a read instruction is executed on P2DR or P2PRD, bits 7 to 3 are read as undefined.

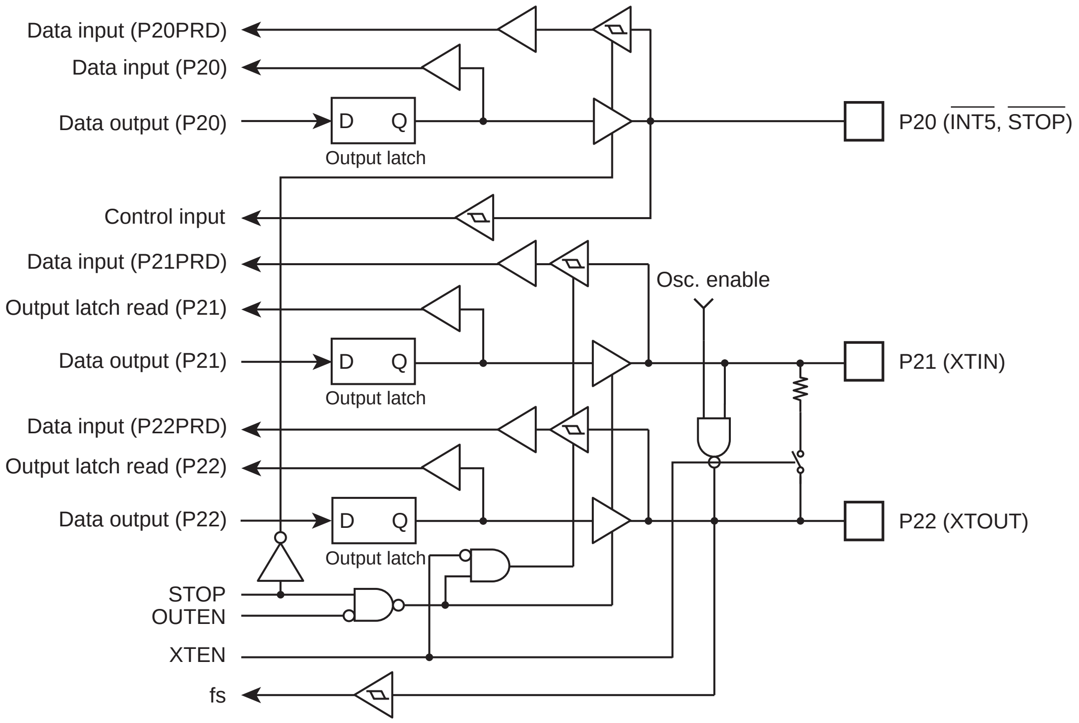


Figure 5-4 Port P2

	7	6	5	4	3	2	1	0	
P2DR (0002H) R/W						P22 XTOUT	P21 XTIN	P20 $\overline{INT5}$ $\overline{STOP}$	(Initial value: **** *111)

	7	6	5	4	3	2	1	0
P2PRD (0FF2H) Read only						P22	P21	P20

Note: Since pin P20 is also used as a $\overline{STOP}$ pin, the output of P20 becomes high-impedance in STOP mode regardless of the OUTEN state.

5.4 Port P3 (P30 to P37)

Port P3 is an 8-bit input/output port that can also be used for external interrupt input, divider output, timer/counter input, serial interface input/output, or UART input/output.

A reset initializes the output latch (P3DR) to “1” and the Pch control (P3OUTCR) to “0”.

To use a pin in Port P3 as an external interrupt input, timer/counter input, serial interface input, or UART input, set P3DR to “1” and then set the corresponding bit in P3OUTCR to “0”.

To use a pin in Port P3 as a divider output, serial interface output, or UART output, set P3DR to “1”.

Port 3 can be used for either SIO or UART, so be sure not to enable both of these functions at the same time.

The output circuit of Port P3 can be set either as sink open-drain output (“0”) or CMOS output (“1”) individually for each bit in P3OUTCR.

Port P3 has a separate data input register. The output latch state can be read from the P3DR register, and the pin state can be read from the P3PRD register.

Table 5-6 Register Programming for Port P3 (P30 to P37)

Function	Programmed Value	
	P3DR	P3OUTCR
Port input, external interrupt input, timer/counter input, serial interface input, UART input	“1”	“0”
Port “0” output	“0”	Set as appropriate.
Port “1” output, serial interface output, UART output, divider output	“1”	

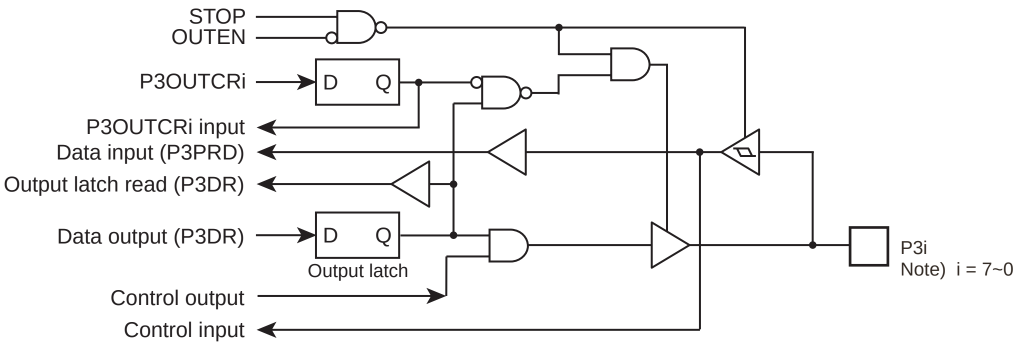


Figure 5-5 Port P3

P3DR (0003H) R/W	7	6	5	4	3	2	1	0	
	P37 TC10 INT4	P36 $\overline{\text{SCK}}$	P35 SI TXD1	P34 SO RXD1	P33	P32	P31 $\overline{\text{DVO}}$	P30 $\overline{\text{INT0}}$	(Initial value: 1111 1111)

P3OUTCR (002BH)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000)

P3OUTCR	Port P3 output circuit control (set for each bit individually)	0: Sink open-drain output 1: CMOS output	R/W

P3PRD (0FF3H) Read only	7	6	5	4	3	2	1	0
	P37	P36	P35	P34	P33	P32	P31	P30

5.5 Port P4 (P40 to P47)

Port P4 is an 8-bit input/output port that can also be used for external interrupt input, PPG output, timer/counter input, or LCD segment output.

A reset initializes the output latch (P4DR) to “1”, the Pch control (P4OUTCR) to “0”, and the LCD output control register (P4LCR) to “0”.

To use a pin in Port P4 as an input port, external interrupt input, or timer/counter input, set P4DR to “1” and then set the corresponding bit in P4LCR and P4OUTCR to “0”.

To use a pin in Port P4 as an LCD segment output, set the corresponding bit in P4LCR to “1”.

To use a pin in Port P4 as a PPG output, set P4DR to “1” and then set the corresponding bit in P4LCR to “0”.

The output circuit of Port P4 can be set either as sink open-drain outut (“0”) or CMOS output (“1”) individually for each bit in P4OUTCR.

Port P4 has a separate data input register. The output latch state can be read from the P4DR register, and the pin state can be read from the P4PRD register.

Table 5-7 Register Programming for Port P4 (P40 to P47)

Function	Programmed Value		
	P4DR	P4OUTCR	P4LCR
Port input, external interrupt input, timer/counter input	“1”	“0”	“0”
Port "0" output	“0”	Set as appropriate.	“0”
Port “1” output	“1”		“0”
PPG output	“1”		“0”
LCD segment output	*	*	“1”

Note: An asterisk (*) indicates that either “1” or “0” can be set.

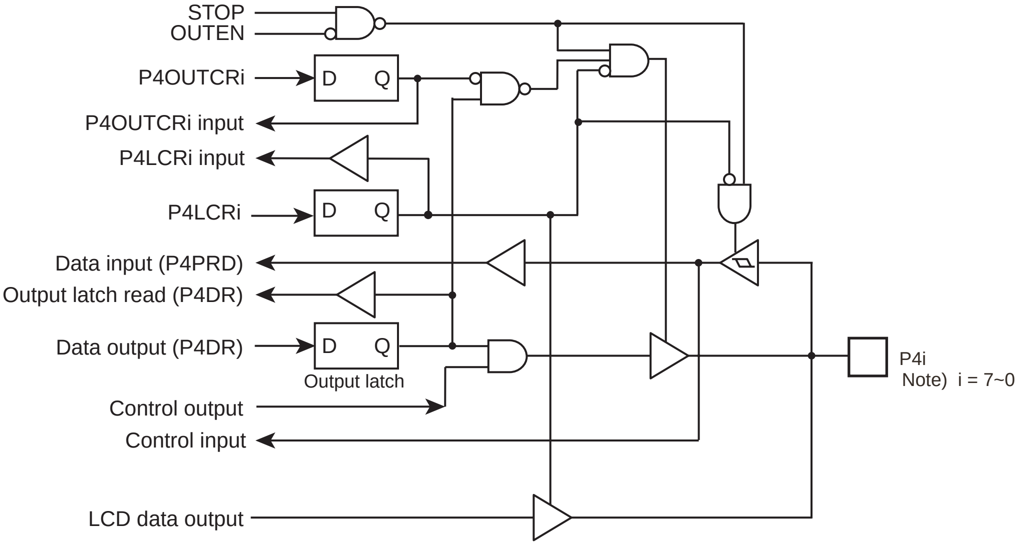


Figure 5-6 Port P4

	7	6	5	4	3	2	1	0	
P4DR (0004H) R/W	P47 SEG32	P46 SEG31	P45 SEG30	P44 SEG29	P43 SEG28 TC11	P42 SEG27 $\overline{\text{PPG1}}$	P41 SEG26 INT2	P40 SEG25 INT1	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P4LCR (0FD4H)									(Initial value: 0000 0000)

P4LCR	Port P4 segment output control (Set for each bit individually)	0: Input/output port 1: LCD segment output	R/W

	7	6	5	4	3	2	1	0	
P4OUTCR (0FFBH)									(Initial value: 0000 0000)

P4OUTCR	P4 output circuit control (Set for each bit individually)	0: Sink open-drain output 1: CMOS output	R/W

	7	6	5	4	3	2	1	0
P4PRD (0FF4H) Read only	P47	P46	P45	P44	P43	P42	P41	P40

5.6 Port P5 (P50 to P57)

Port P5 is an 8-bit input/output port that can also be used for timer/counter input/output, LCD segment output, or UART input/output.

A reset initializes the output latch (P5DR) to “1”, the Pch control (P5OUTCR) to “0”, and the LCD output control register (P5LCR) to “0”.

To use a pin in Port P5 as an input port, timer/counter input, or UART input, set P5DR to “1” and then set the corresponding bit in P5LCR and P5OUTCR to “0”.

To use a pin in Port P5 as an LCD segment output, set the corresponding bit in P5LCR to “1”.

To use a pin in Port P5 as a UART output or timer/counter output, set P5DR to "1" and then set the corresponding bit in P5LCR to “0”.

The output circuit of Port P5 can be set either as sink open-drain output (“0”) or CMOS otuput (“1”) individually for each bit in P5OUTCR.

Port P5 has a separate data input register. The output latch state can be read from the P5DR register, and the pin state can be read from the P5PRD register.

Table 5-8 Register Programming for Port P5 (P50 to P57)

Function	Programmed Value		
	P5DR	P5OUTCR	P5LCR
Port input, UART input, timer/counter input	“1”	“0”	“0”
Port “0” output	“0”	Set as appropriate.	“0”
Port “1” output, UART output	“1”		“0”
LCD segment output	*	*	“1”

Note: An asterisk (*) indicates that either “1” or “0” can be set.

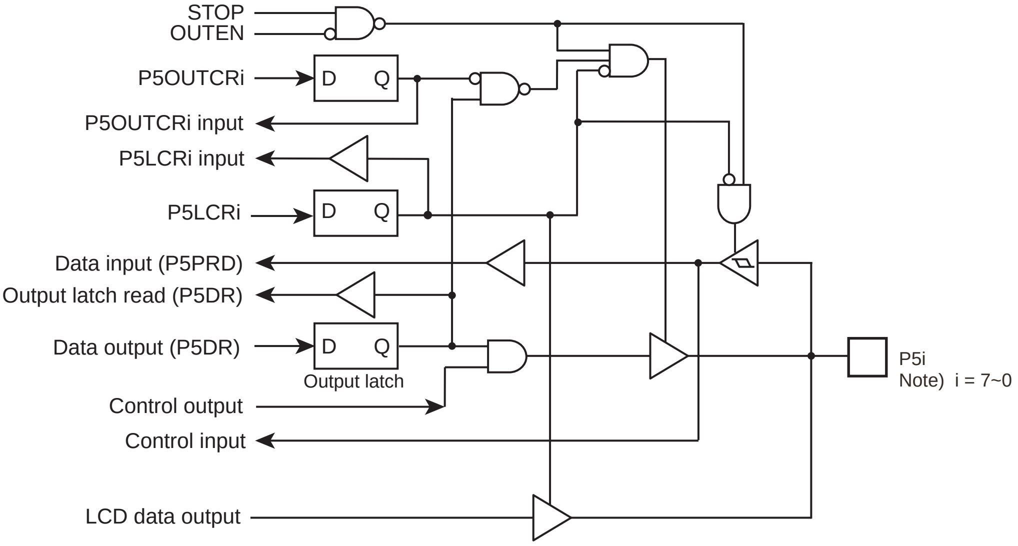


Figure 5-7 Port P5

	7	6	5	4	3	2	1	0	
P5DR (0005H) R/W	P57 SEG24	P56 SEG25	P55 SEG26 TC6 $\overline{\text{PWM6}}$ $\overline{\text{PDO6}}$	P54 SEG27 TC5 $\overline{\text{PWM5}}$ $\overline{\text{PDO5}}$	P53 SEG28 TC4 $\overline{\text{PWM4}}$ $\overline{\text{PDO4}}$	P52 SEG29 TC3 $\overline{\text{PWM3}}$ $\overline{\text{PDO3}}$	P51 SEG30 RXD0	P50 SEG31 TXD0	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P5LCR (0FD5H)									(Initial value: 0000 0000)

P5LCR	Port P5 segment output control (Set for each bit individually)	0: Input/output port 1: LCD segment output	R/W

	7	6	5	4	3	2	1	0	
P5OUTCR (0FFCH)									(Initial value: 0000 0000)

P5OUTCR	Port P5 input/output control (Set for each bit individually)	0: Sink open-drain output 1: CMOS output	R/W

	7	6	5	4	3	2	1	0
P5PRD (0FF5H) Read only	P57	P56	P55	P54	P53	P52	P51	P50

5.7 Port P6 (P60 to P67)

Port P6 is an 8-bit input/output port that can also be used for LCD segment output.

A reset initializes the output latch (P6DR) to “1”, the Pch control (P6OUTCR) to “0”, and the LCD output control register (P6LCR) to “0”.

To use a pin in Port P6 as an input port, set P6DR to “1” and then set the corresponding bit in P6LCR and P6OUTCR to “0”.

To use a pin in Port P6 as an LCD segment output, set the corresponding bit in P6LCR to “1”.

The output circuit of Port P6 can be set either as sink open-drain output (“0”) or CMOS output (“1”) individually for each bit in P6OUTCR.

Port P6 has a separate data input register. The outut latch state can be read from the P6DR register, and the pin state can be read from the P6PRD register.

Table 5-9 Register Programming for Port P6 (P60 to P67)

Function	Programmed Value		
	P6DR	P6OUTCR	P6LCR
Port input	“1”	“0”	“0”
Port "0" output	“0”	Set as appropriate.	“0”
Port “1” output	“1”		“0”
LCD segment output	*	*	“1”

Note: An asterisk (*) indicates that either “1” or “0” can be set.

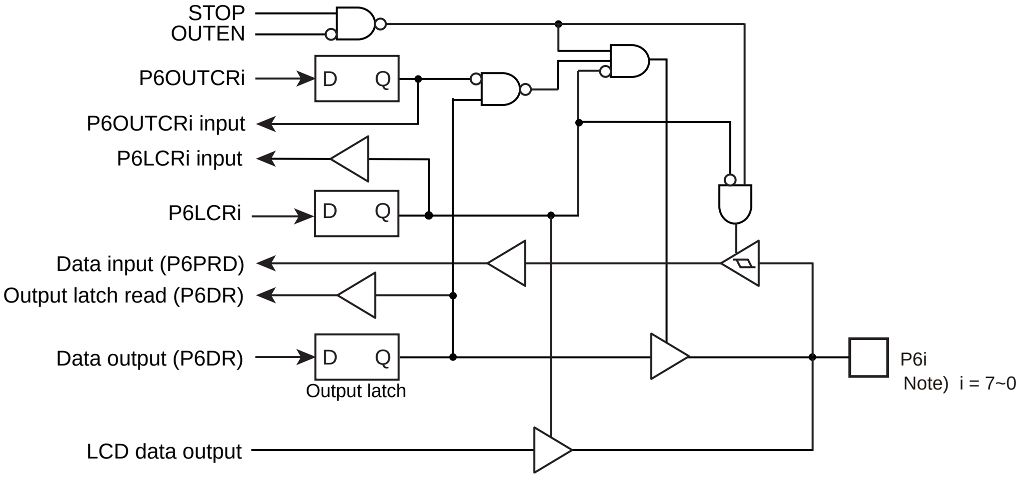


Figure 5-8 Port P6

	7	6	5	4	3	2	1	0	
P6DR (0006H) R/W	P67 SEG16	P66 SEG17	P65 SEG18	P64 SEG19	P63 SEG20	P62 SEG21	P61 SEG22	P60 SEG23	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P6LCR (0FD6H)									(Initial value: 0000 0000)

P6LCR	Port P6 segment output control (Set for each bit individually)	0: Input/output port 1: Segment output	R/W

	7	6	5	4	3	2	1	0	
P6OUTCR (0FFDH)									(Initial value: 1111 1111)

P6CR2	Port P6 input/output control (Set for each bit individually)	0: Sink open-drain output 1: CMOS output	R/W

	7	6	5	4	3	2	1	0
P6PRD (0FF6H) Read only	P67	P66	P65	P64	P63	P62	P61	P60

5.8 Port P7 (P70 to P77)

Port P7 is an 8-bit input/output port that can also be used for LCD segment output.

A reset initializes the output latch (P7DR) to “1”, the Pch control (P7OUTCR) to “0”, and the LCD output control register (P7LCR) to “0”.

To use a pin in Port P7 as an input port, set P7DR to “1” and then set the corresponding bit in P7LCR and P7OUTCR to “0”.

To use a pin in Port P7 as an LCD segment output, set the corresponding bit in P7LCR to “1”.

The output circuit of Port P7 can be set either as sink open-drain output (“0”) or CMOS output (“1”) individually for each bit in P7OUTCR.

Port P7 has a separate data input register. The output latch state can be read from the P7DR register, and the pin state can be read from the P7PRD register.

Table 5-10 Register Programming for Port P7 (P70 to P77)

Function	Programmed Value		
	P7DR	P7OUTCR	P7LCR
Port input	“1”	“0”	“0”
Port “0” output	“0”	Set as appropriate.	“0”
Port “1” output	“1”		“0”
LCD segment output	*	*	“1”

Note: An asterisk (*) indicates that either “1” or “0” can be set.

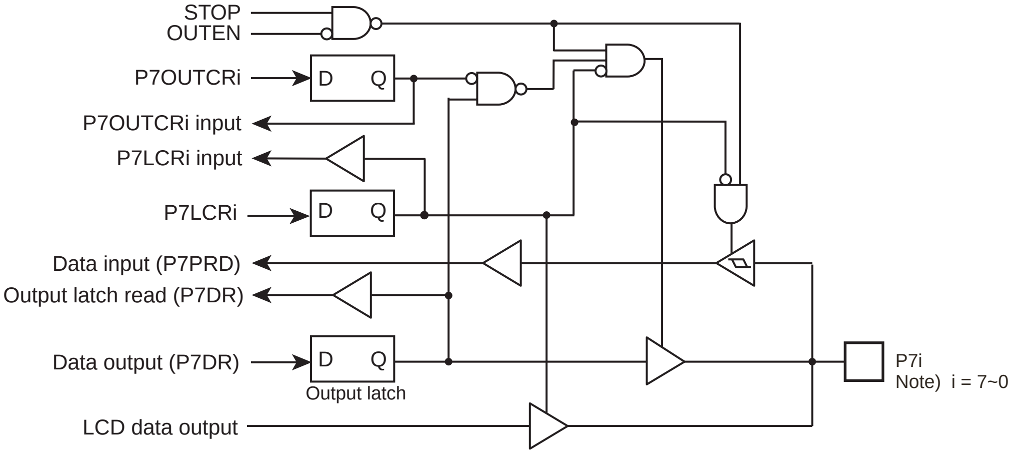


Figure 5-9 Port P7

P7DR	7	6	5	4	3	2	1	0	
(0007H) R/W	P77 SEG8	P76 SEG9	P75 SEG10	P74 SEG11	P73 SEG12	P72 SEG13	P71 SEG14	P70 SEG15	(Initial value: 0000 0000)

P7LCR	7	6	5	4	3	2	1	0	
(0FD7H)									(Initial value: 0000 0000)

P7LCR	Port P7 segment output control (set for each bit individually)	0: Input/output port 1: Segment output	R/W

P7OUTCR	7	6	5	4	3	2	1	0	
(0FFEH)									(Initial value: 0000 0000)

P7OUTCR	Port P7 input/output control (set for each bit individually)	0: Sink open-drain output 1: CMOS output	R/W

P7PRD	7	6	5	4	3	2	1	0
(0FF7H) Read only	P77	P76	P75	P74	P73	P72	P71	P70

5.9 Port P8 (P80 to P87)

Port P8 is an 8-bit input/output port that can also be used for LCD segment output.

A reset initializes the output latch (P8DR) to “1”, the Pch control (P8OUTCR) to “0”, and the LCD output control register (P8LCR) to “0”.

To use a pin in Port P8 as an input port, set P8DR to “1” and then set the corresponding bit in P8LCR and P8OUTCR to “0”.

To use a pin in Port P8 as an LCD segment output, set the corresponding bit in P8LCR to “1”.

The output circuit of Port P8 can be set either as sink open-drain output (“0”) or CMOS output (“1”) individually for each bit in P8OUTCR.

Port P8 has a separate data input register. The output latch state can be read from the P8DR register, and the pin state can be read from the P8PRD register.

Table 5-11 Register Programming for Port P8 (P80 to P87)

Function	Port Input		
	P8DR	P8OUTCR	P8LCR
Port input	“1”	“0”	“0”
Port “0” output	“0”	Set as appropriate.	“0”
Port “1” output	“1”		“0”
LCD segment output	*	*	“1”

Note: An asterisk (*) indicates that either “1” or “0” can be set.

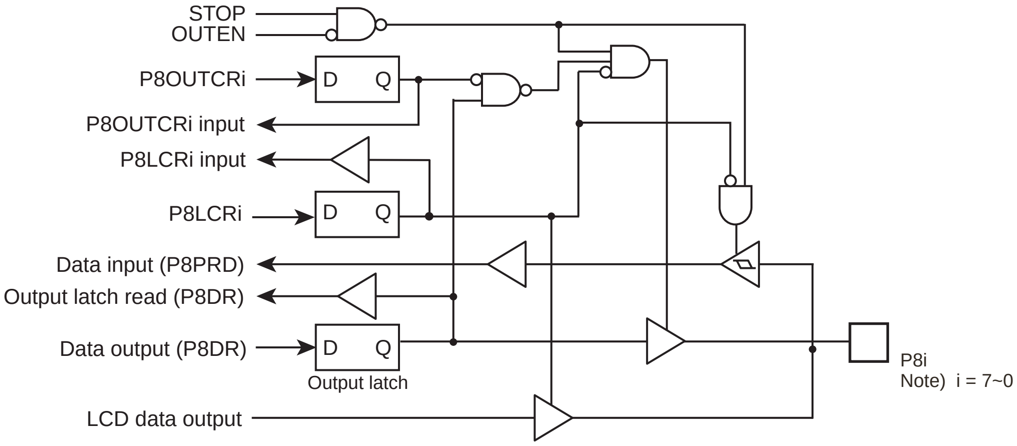


Figure 5-10 Port P8

P8DR (0008H) R/W	7	6	5	4	3	2	1	0	
	P87 SEG0	P86 SEG1	P85 SEG2	P84 SEG3	P83 SEG4	P82 SEG5	P81 SEG6	P80 SEG7	(Initial value: 0000 0000)

P8LCR (0FD8H)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000)

P8LCR	Port P8 segment output control (Set for each bit individually)	0: Input/output port 1: LCD segment output	R/W

P8OUTCR (0FFFH)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000)

P8OUTCR	Port P8 input/output control (Set for each bit individually)	0: Sink open-drain output 1: CMOS output	R/W

P8PRD (0FF8H) Read only	7	6	5	4	3	2	1	0
	P87	P86	P85	P84	P83	P82	P81	P80

6. 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.

6.1 Watchdog Timer Configuration

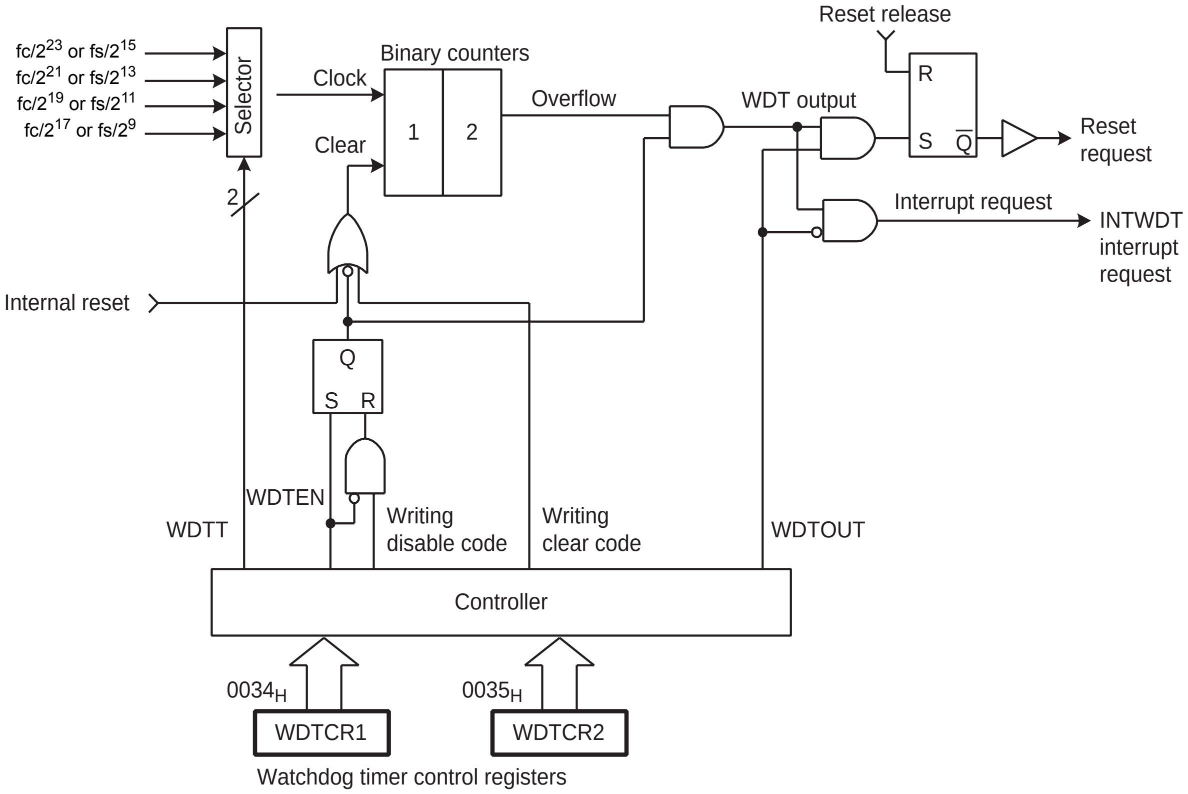


Figure 6-1 Watchdog Timer Configuration

6.2 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.

6.2.1 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 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}/fc [s], and resetting the CPU malfunction detection

```
                           LD      (WDTCR2), 4EH          : Clears the binary counters.

                           LD      (WDTCR1), 00001101B    : WDTT ← 10, WDTOUT ← 1

                         ┌ LD      (WDTCR2), 4EH          : Clears the binary counters (always clears immediately before and
                         │                                  after changing WDTT).
                         │  :
Within 3/4 of WDT        │
detection time           │  :
                         │
                         ├ LD      (WDTCR2), 4EH          : Clears the binary counters.
                         │
Within 3/4 of WDT        │  :
detection time           │
                         │  :
                         │ LD      (WDTCR2), 4EH          : Clears the binary counters.
                         └
```

Watchdog Timer Control Register 1

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

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 “6.2.3 Watchdog Timer Disable”.

Watchdog Timer Control Register 2

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

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>.

6.2.2 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.

6.2.3 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 6-1 Watchdog Timer Detection Time (Example: fc = 16.0 MHz, fs = 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

6.2.4 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, 083FH               : Sets the stack pointer
LD      (WDTCR1), 00001000B     : WDTOUT ← 0
```

6.2.5 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 internal hardware is reset. The reset time is maximum 24/fc [s] (1.5 µs @ 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.

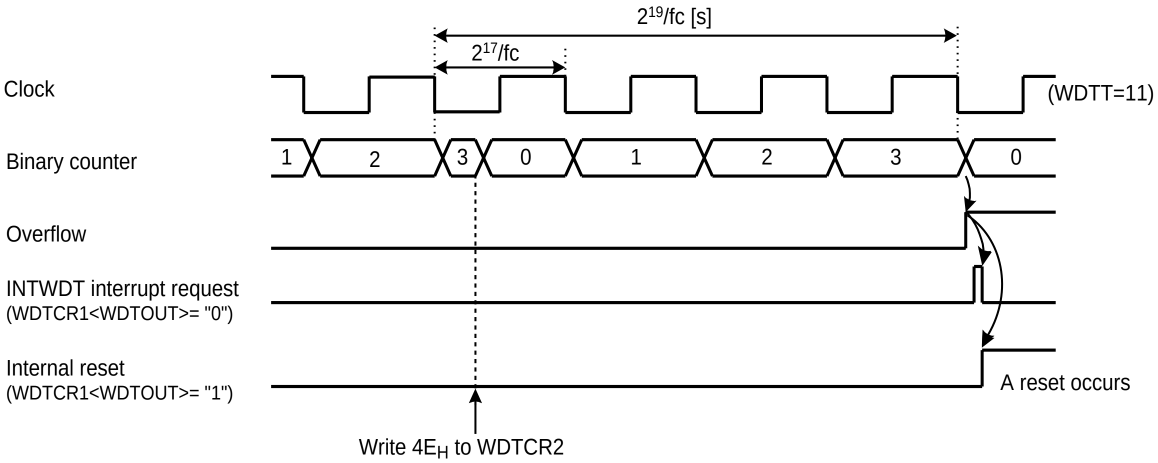


Figure 6-2 Watchdog Timer Interrupt

6.3 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	
			ATAS	ATOUT	(WDTEN)	(WDTT)		(WDTOUT)	(Initial value: **11 1001)

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

6.3.1 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 or DBR area generates an address trap unconditionally regardless of the setting in WDTCR1<ATAS>.

6.3.2 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>.

6.3.3 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”), DBR 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.

6.3.4 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”), DBR or the SFR area, address trap reset will be generated.

When an address trap reset request is generated, the internal hardware is reset. The reset time is maximum 24/fc [s] (1.5 µs @ fc = 16.0 MHz).

Note: When an address trap 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.

7. 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).

7.1 Time Base Timer

7.1.1 Configuration

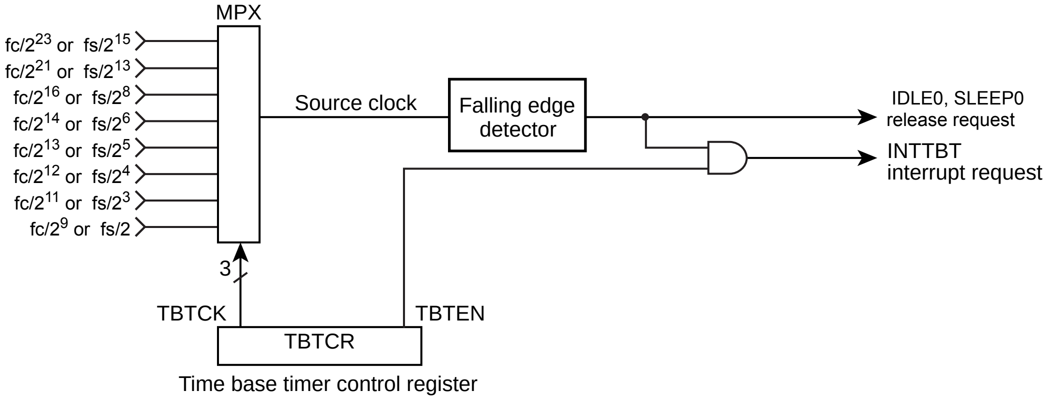


Figure 7-1 Time Base Timer configuration

7.1.2 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	TBTCK			(Initial Value: 0000 0000)

TBTEN	Time Base Timer enable / disable	0: Disable 1: Enable				
TBTCK	Time Base Timer interrupt Frequency select : [Hz]		NORMAL1/2, IDLE1/2 Mode		SLOW1/2 SLEEP1/2 Mode	R/W
			DV7CK = 0	DV7CK = 1		
		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 7-1 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	–

7.1.3 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 7-2).

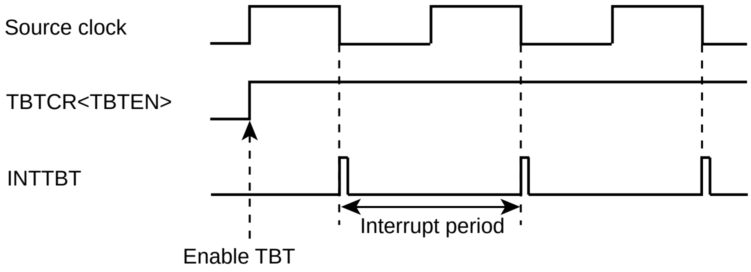
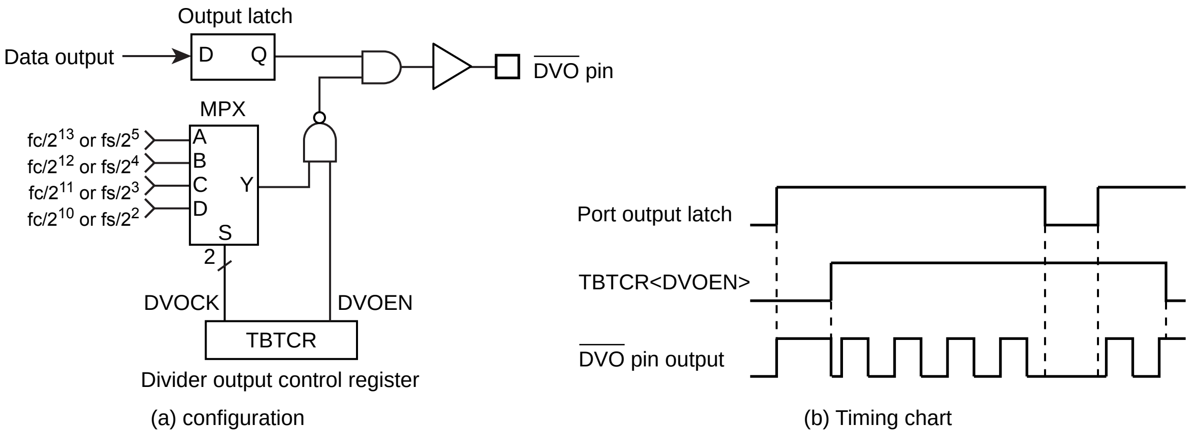


Figure 7-2 Time Base Timer Interrupt

7.2 Divider Output ($\overline{\text{DVO}}$)

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

7.2.1 Configuration


Output latch
Data output
D
Q
$\overline{\text{DVO}}$ pin
MPX
$fc/2^{13}$ or $fs/2^{5}$
$fc/2^{12}$ or $fs/2^{4}$
$fc/2^{11}$ or $fs/2^{3}$
$fc/2^{10}$ or $fs/2^{2}$
A
B
C
D
Y
S
2
DVOCK
DVOEN
TBTCR
Divider output control register
Port output latch
TBTCR<DVOEN>
$\overline{\text{DVO}}$ pin output

(a) configuration

(b) Timing chart

Figure 7-3 Divider Output

7.2.2 Control

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

Time Base Timer Control Register

	7	6	5	4	3	2	1	0	
TBTCR (0036H)	DVOEN	DVOCK		(DV7CK)	(TBTEN)	(TBTCK)			(Initial value: 0000 0000)

DVOEN	Divider output enable / disable	0: Disable 1: Enable				R/W
DVOCK	Divider Output ($\overline{\text{DVO}}$) frequency selection: [Hz]		NORMAL1/2, IDLE1/2 Mode		SLOW1/2 SLEEP1/2 Mode	R/W
			DV7CK = 0	DV7CK = 1		
		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 disable(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"
```

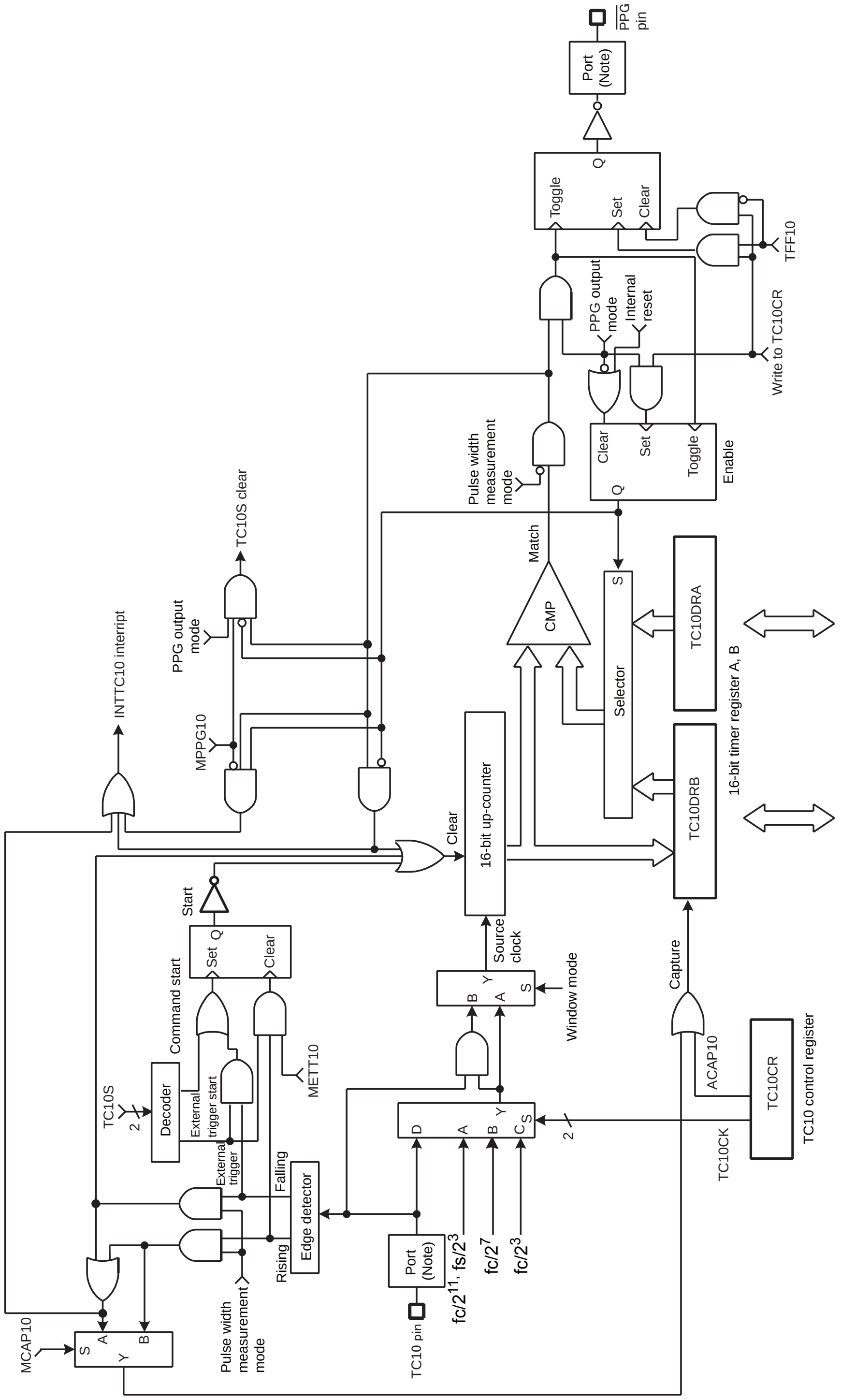
Table 7-2 Divider Output Frequency (Example : fc = 16.0 MHz, fs = 32.768 kHz)

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

8. 16-Bit TimerCounter (TC10,TC11)

8.1 16-Bit TimerCounter 10

8.1.1 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 10 (TC10)

8.1.2 TimerCounter Control

The TimerCounter 10 is controlled by the TimerCounter 10 control register (TC10CR) and two 16-bit timer registers (TC10DRA and TC10DRB).

Timer Register

	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
TC10DRA (0011H, 0010H)	TC10DRAH (0011H)	TC10DRAL (0010H)
	(Initial value: 1111 1111 1111 1111)	Read/Write
TC10DRB (0013H, 0012H)	TC10DRBH (0013H)	TC10DRBL (0012H)
	(Initial value: 1111 1111 1111 1111)	Read/Write (Write enabled only in the PPG output mode)

TimerCounter 10 Control Register

	7	6	5 4	3 2	1 0	
TC10CR (0014H)	TFF10	ACAP10 MCAP10 METT10 MPPG10	TC10S	TC10CK	TC10M	Read/Write (Initial value: 0000 0000)

TFF10	Timer F/F10 control	0: Clear	1: Set						R/W
ACAP10	Auto capture control	0:Auto-capture disable	1:Auto-capture enable						R/W
MCAP10	Pulse width measurement mode control	0:Double edge capture	1:Single edge capture						
METT10	External trigger timer mode control	0:Trigger start	1:Trigger start and stop						
MPPG10	PPG output control	0:Continuous pulse generation	1:One-shot						
TC10S	TC10 start control		Timer	Extrig-ger	Event	Win-dow	Pulse	PPG	R/W
		00: Stop and counter clear	O	O	O	O	O	O	
		01: Command start	O	–	–	–	–	O	
		10: Rising edge start (Ex-trigger/Pulse/PPG) Rising edge count (Event) Positive logic count (Window)	–	O	O	O	O	O	
		11: Falling edge start (Ex-trigger/Pulse/PPG) Falling edge count (Event) Negative logic count (Window)	–	O	O	O	O	O	
TC10CK	TC10 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 (TC10 pin input)						
TC10M	TC10 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 (TC10DRAH and TC10DRBH) 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 (TC10DRAL and TC10DRBL) 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 TC10CR1 during TC10S=00. Set the timer F/F10 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.
TC10DRA > TC10DRB > 1 (PPG output mode), TC10DRA > 1 (other modes)

Note 6: Set TFF10 to “0” in the mode except PPG output mode.

Note 7: Set TC10DRB after setting TC10M to the PPG output mode.

Note 8: When the STOP mode is entered, the start control (TC10S) is cleared to “00” automatically, and the timer stops. After the STOP mode is exited, set the TC10S to use the timer counter again.

Note 9: Use the auto-capture function in the operative condition of TC10. 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 TC10DRB by the source clock of up-counter after setting TC10CR<ACAP10> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC10DRB for the first time.

8.1.3 Function

TimerCounter 10 has six types of operating modes: timer, external trigger timer, event counter, window, pulse width measurement, programmable pulse generator output modes.

8.1.3.1 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 (TC10DRA) value is detected, an INTTC10 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting. Setting TC10CR<ACAP10> to “1” captures the up-counter value into the timer register 1B (TC10DRB) with the auto-capture function. Use the auto-capture function in the operative condition of TC10. 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 TC10DRB by the source clock of up-counter after setting TC10CR<ACAP10> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC10DRB for the first time.

Table 8-1 Internal Source Clock for TimerCounter 10 (Example: fc = 16 MHz, fs = 32.768 kHz)

TC10CK	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	–	–

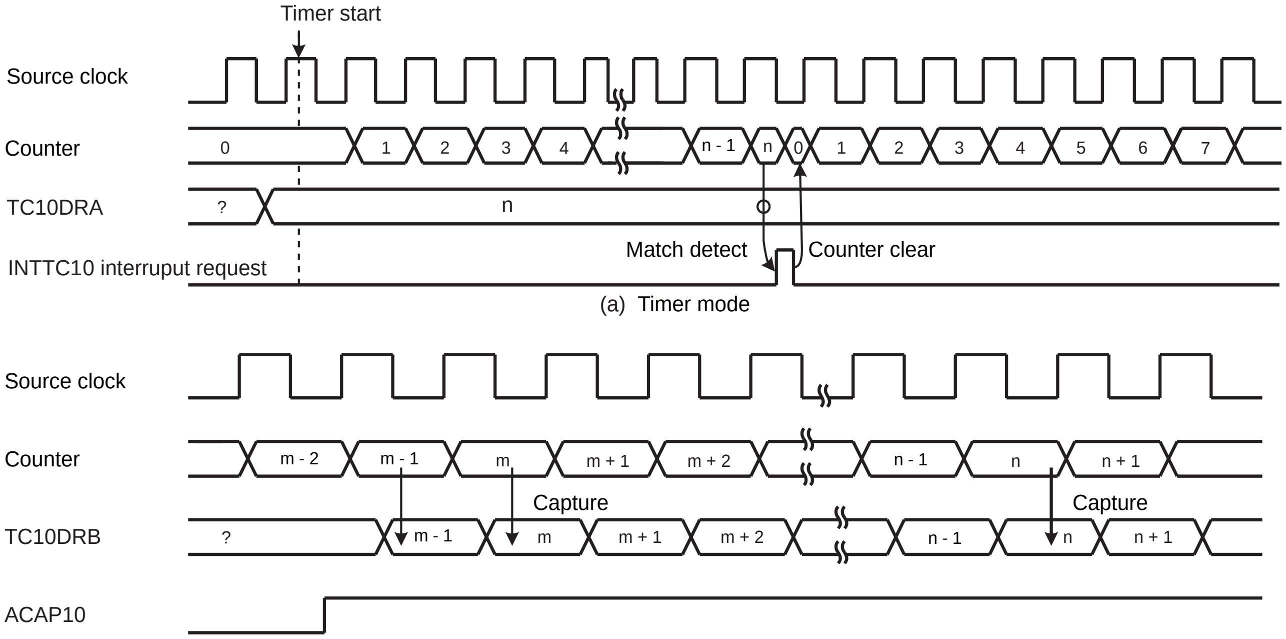
Example 1 :Setting the timer mode with source clock fc/2^{11} [Hz] and generating an interrupt 1 second later (fc = 16 MHz, TBTCR<DV7CK> = “0”)

```
LDW     (TC10DRA), 1E84H      ; Sets the timer register (1 s ÷ 2^11/fc = 1E84H)
DI                            ; IMF= “0”
SET     (EIRL). 7             ; Enables INTTC10
EI                            ; IMF= “1”
LD      (TC10CR), 00000000B   ; Selects the source clock and mode
LD      (TC10CR), 00010000B   ; Starts TC10
```

Example 2 :Auto-capture

```
LD          (TC10CR), 01010000B      ; ACAP10 ← 1
:           :
LD          WA, (TC10DRB)            ; Reads the capture value
```

Note: Since the up-counter value is captured into TC10DRB by the source clock of up-counter after setting TC10CR<ACAP10> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC10DRB for the first time.



(a) Timer mode

(b) Auto-capture

Figure 8-2 Timer Mode Timing Chart

8.1.3.2 External Trigger Timer Mode

In the external trigger timer mode, the up-counter starts counting by the input pulse triggering of the TC10 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 TC10CR<TC10S>.

- When TC10CR<METT10> is set to “1” (trigger start and stop)

  When a match between the up-counter and the TC10DRA value is detected after the timer starts, the up-counter is cleared and halted and an INTTC10 interrupt request is generated.

  If the edge opposite to trigger edge is detected before detecting a match between the up-counter and the TC10DRA, 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 TC10CR<METT10> is set to “0” (trigger start)

  When a match between the up-counter and the TC10DRA value is detected after the timer starts, the up-counter is cleared and halted and an INTTC10 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 TC10DRA.

Since the TC10 pin input has the noise rejection, pulses of 4/fc [s] or less are rejected as noise. A pulse width of 12/fc [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 TC10 pin (fc =16 MHz)

```
LDW     (TC10DRA), 007DH        ; 1ms ÷ 2^7/fc = 7DH
DI                              ; IMF= “0”
SET     (EIRL). 7               ; Enables INTTC10 interrupt
EI                              ; IMF= “1”
LD      (TC10CR), 00000100B     ; Selects the source clock and mode
LD      (TC10CR), 00100100B     ; Starts TC10 external trigger, METT10 = 0
```

Example 2 :Generating an interrupt when the low-level pulse with 4 ms or more width is input to the TC10 pin (fc =16 MHz)

```
LDW     (TC10DRA), 01F4H        ; 4 ms ÷ 2^7/fc = 1F4H
DI                              ; IMF= “0”
SET     (EIRL). 7               ; Enables INTTC10 interrupt
EI                              ; IMF= “1”
LD      (TC10CR), 00000100B     ; Selects the source clock and mode
LD      (TC10CR), 01110100B     ; Starts TC10 external trigger, METT10 = 0
```

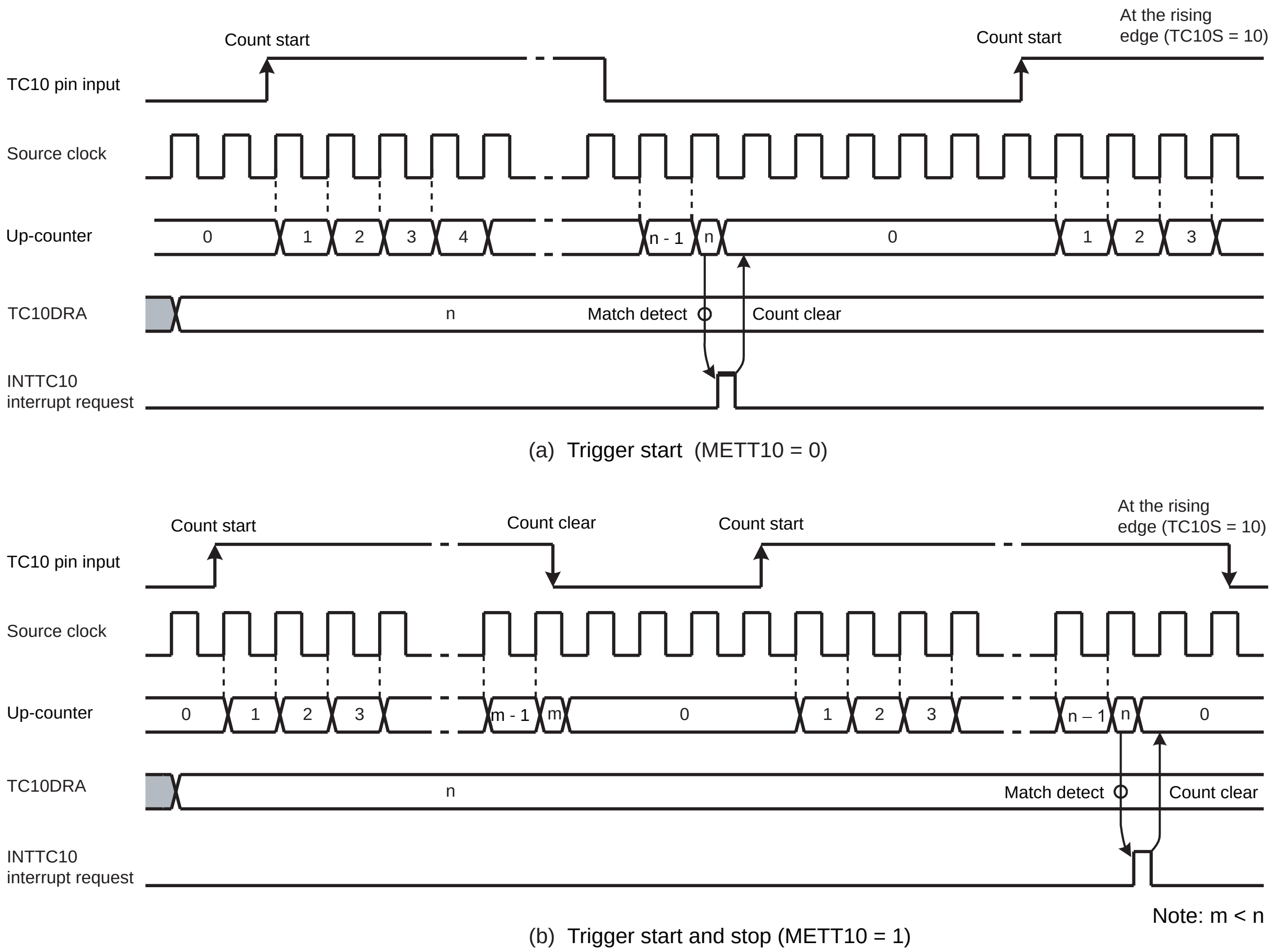


Figure 8-3 External Trigger Timer Mode Timing Chart

8.1.3.3 Event Counter Mode

In the event counter mode, the up-counter counts up at the edge of the input pulse to the TC10 pin. Either the rising or falling edge of the input pulse is selected as the count up edge in TC10CR<TC10S>.

When a match between the up-counter and the TC10DRA value is detected, an INTTC10 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 TC10 pin. Since a match between the up-counter and the value set to TC10DRA is detected at the edge opposite to the selected edge, an INTTC10 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 TC10 pin.

Setting TC10CR<ACAP10> to “1” captures the up-counter value into TC10DRB with the auto capture function. Use the auto-capture function in the operative condition of TC10. 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 TC10DRB by the source clock of up-counter after setting TC10CR<ACAP10> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC10DRB for the first time.

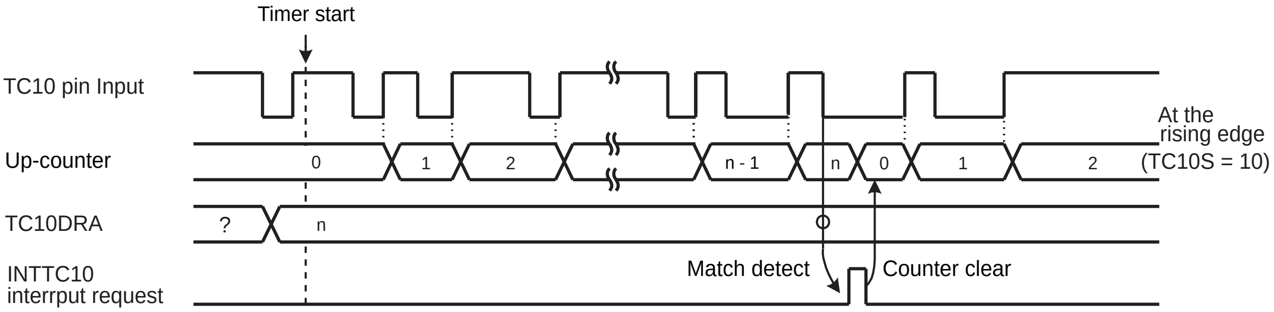


Figure 8-4 Event Counter Mode Timing Chart

Table 8-2 Input Pulse Width to TC10 Pin

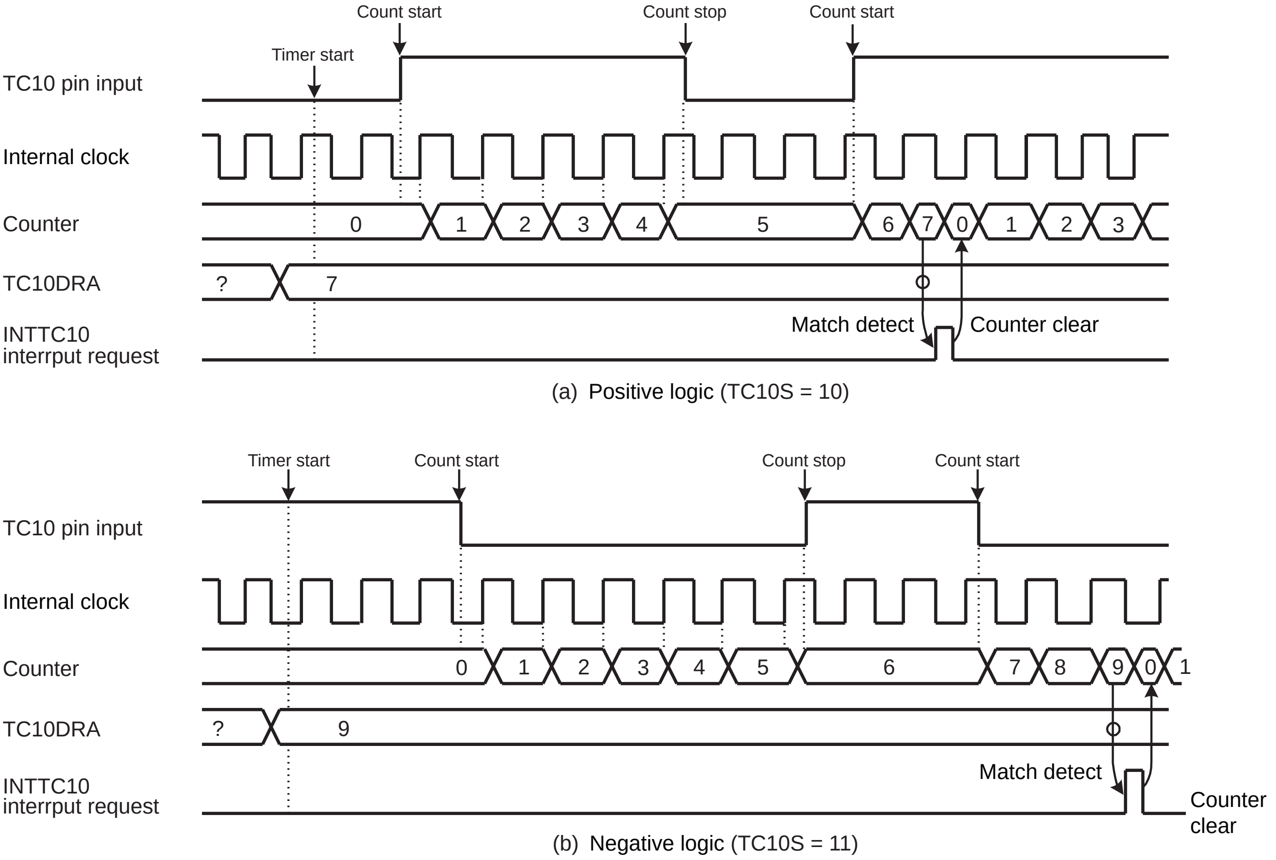
	Minimum Pulse Width [s]	
	NORMAL1/2, IDLE1/2 Mode	SLOW1/2, SLEEP1/2 Mode
High-going	2^3/fc	2^3/fs
Low-going	2^3/fc	2^3/fs

8.1.3.4 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 TC10 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 TC10DRA value is detected, an INTTC10 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 TC10CR<TC10CK>.



(a) Positive logic (TC10S = 10)

(b) Negative logic (TC10S = 11)

Figure 8-5 Window Mode Timing Chart

8.1.3.5 Pulse Width Measurement Mode

In the pulse width measurement mode, the up-counter starts counting by the input pulse triggering of the TC10 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 TC10CR<TC10S>. Either the single- or double-edge capture is selected as the trigger edge in TC10CR<MCAP10>.

- When TC10CR<MCAP10> 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 TC10CR<TC10S>. To measure the low-level input pulse width, set the falling edge to TC10CR<TC10S>.

  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 TC10DRB and generates an INTTC10 interrupt request. The up-counter is cleared at this time, and then restarts counting when detecting the trigger edge used to start counting.

- When TC10CR<MCAP10> 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 TC10CR<TC10S>. To measure the cycle starting with the low-going pulse, set the falling edge to TC10CR<TC10S>.

  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 TC10DRB and generates an INTTC10 interrupt request. The up-counter continues counting up, and captures the up-counter value into TC10DRB and generates an INTTC10 interrupt request when detecting the trigger edge used to start counting. The up-counter is cleared at this time, and then continues counting.

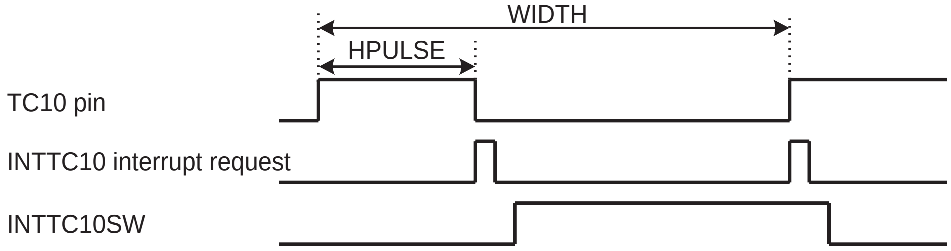
Note 1: The captured value must be read from TC10DRB until the next trigger edge is detected. If not read, the captured value becomes a don’t care. It is recommended to use a 16-bit access instruction to read the captured value from TC10DRB.

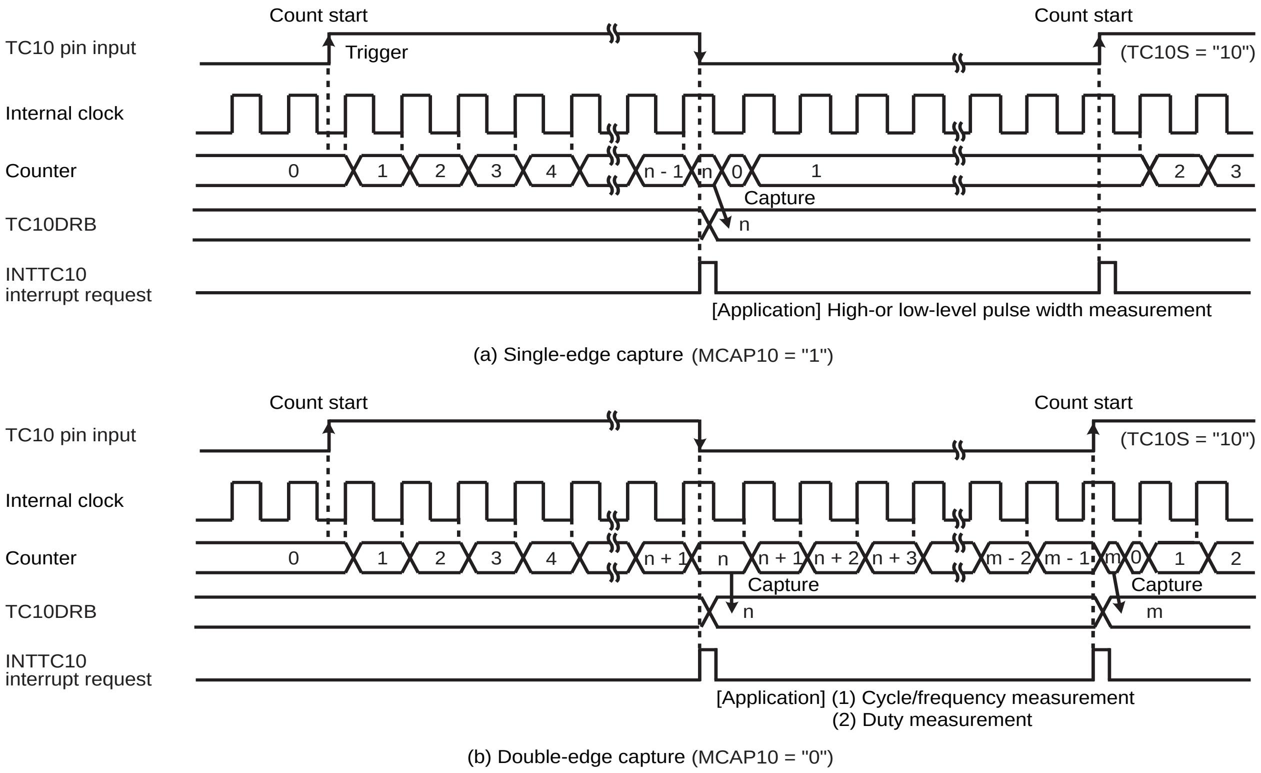
Note 2: For the single-edge capture, the counter after capturing the value stops at “1” until detecting the next edge. Therefore, the second captured value is “1” larger than the captured value immediately after counting starts.

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

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

```
              CLR     (INTTC10SW). 0          ; INTTC10 service switch initial setting
                                                Address set to convert INTTC10SW at each INTTC10
              LD      (TC10CR), 00000110B     ; Sets the TC10 mode and source clock
              DI                              ; IMF= "0"
              SET     (EIRL). 7               ; Enables INTTC10
              EI                              ; IMF= "1"
              LD      (TC10CR), 00100110B     ; Starts TC10 with an external trigger at MCAP10 = 0
               :
PINTTC10:     CPL     (INTTC10SW). 0          ; INTTC10 interrupt, inverts and tests INTTC10 service switch
              JRS     F, SINTTC10
              LD      A, (TC10DRBL)           ; Reads TC10DRB (High-level pulse width)
              LD      W,(TC10DRBH)
              LD      (HPULSE), WA            ; Stores high-level pulse width in RAM
              RETI
SINTTC10:     LD      A, (TC10DRBL)           ; Reads TC10DRB (Cycle)
              LD      W,(TC10DRBH)
              LD      (WIDTH), WA             ; Stores cycle in RAM
               :
              RETI                            ; Duty calculation
              :
VINTTC10:     DW      PINTTC10                ; INTTC10 Interrupt vector
```





(a) Single-edge capture (MCAP10 = "1")

(b) Double-edge capture (MCAP10 = "0")

Figure 8-6 Pulse Width Measurement Mode

8.1.3.6 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, TC10CR<TC10S> specifies either the edge of the input pulse to the TC10 pin or the command start. TC10CR<MPPG10> specifies whether a duty pulse is produced continuously or not (one-shot pulse).

- When TC10CR<MPPG10> is set to “0” (Continuous pulse generation)

  When a match between the up-counter and the TC10DRB value is detected after the timer starts, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC10 interrupt request is generated. The up-counter continues counting. When a match between the up-counter and the TC10DRA value is detected, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC10 interrupt request is generated. The up-counter is cleared at this time, and then continues counting and pulse generation.

  When TC10S is cleared to “00” during PPG output, the $\overline{\text{PPG}}$ pin retains the level immediately before the counter stops.

- When TC10CR<MPPG10> is set to “1” (One-shot pulse generation)

  When a match between the up-counter and the TC10DRB value is detected after the timer starts, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC10 interrupt request is generated. The up-counter continues counting. When a match between the up-counter and the TC10DRA value is detected, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC10 interrupt request is generated. TC10CR<TC10S> 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 TC10CR<TFF10> 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 TC10CR<TFF10> 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 TC10DRA or TC10DRB 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 TC10CR<TFF10> during a run of the timer. TC10CR<TFF10> can be set correctly only at initialization (after reset). When the timer stops during PPG, TC10CR<TFF10> 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 TC10CR<TFF10> 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 TC10CR<TC10M> to the timer mode (it is not required to start the timer mode), and then set the PPG mode. Set TC10CR<TFF10> at this time.

Note 3: In the PPG mode, the following relationship must be satisfied.
TC10DRA > TC10DRB

Note 4: Set TC10DRB after changing the mode of TC10M to the PPG mode.

Example :Generating a pulse which is high-going for 800 μs and low-going for 200 μs
(fc = 16 MHz)

```
                   Setting port
LD                 (TC10CR), 10000111B      ; Sets the PPG mode, selects the source clock
LDW                (TC10DRA), 007DH         ; Sets the cycle (1 ms ÷ 2^7/fc ms = 007DH)
LDW                (TC10DRB), 0019H         ; Sets the low-level pulse width (200 μs ÷ 2^7/fc = 0019H)
LD                 (TC10CR), 10010111B      ; Starts the timer
```

Example :After stopping PPG, setting the PPG pin to a high-level to restart PPG
(fc = 16 MHz)

```
                   Setting port
LD                 (TC10CR), 10000111B      ; Sets the PPG mode, selects the source clock
LDW                (TC10DRA), 007DH         ; Sets the cycle (1 ms ÷ 2^7/fc μs = 007DH)
LDW                (TC10DRB), 0019H         ; Sets the low-level pulse width (200 μs ÷ 2^7/fc = 0019H)
LD                 (TC10CR), 10010111B      ; Starts the timer
:                  :
LD                 (TC10CR), 10000111B      ; Stops the timer
LD                 (TC10CR), 10000100B      ; Sets the timer mode
LD                 (TC10CR), 00000111B      ; Sets the PPG mode, TFF10 = 0
LD                 (TC10CR), 00010111B      ; Starts the timer
```

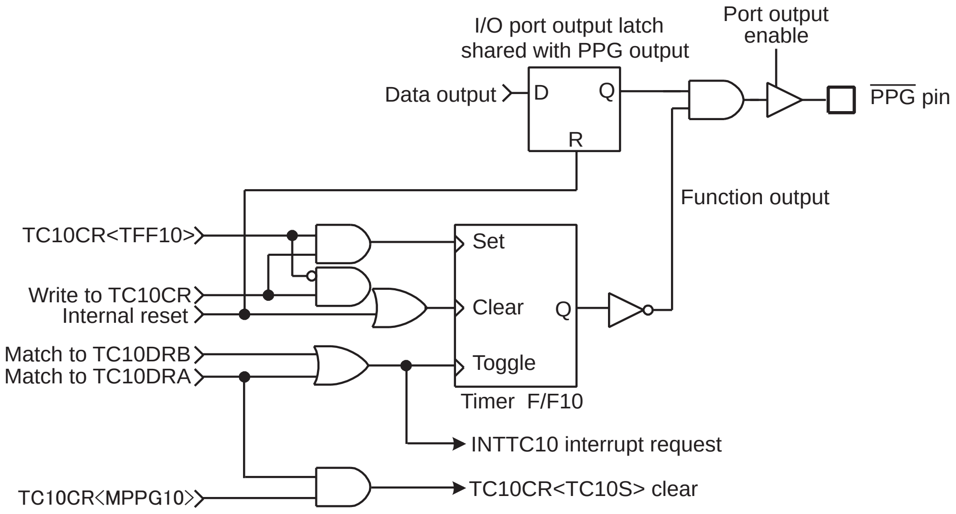
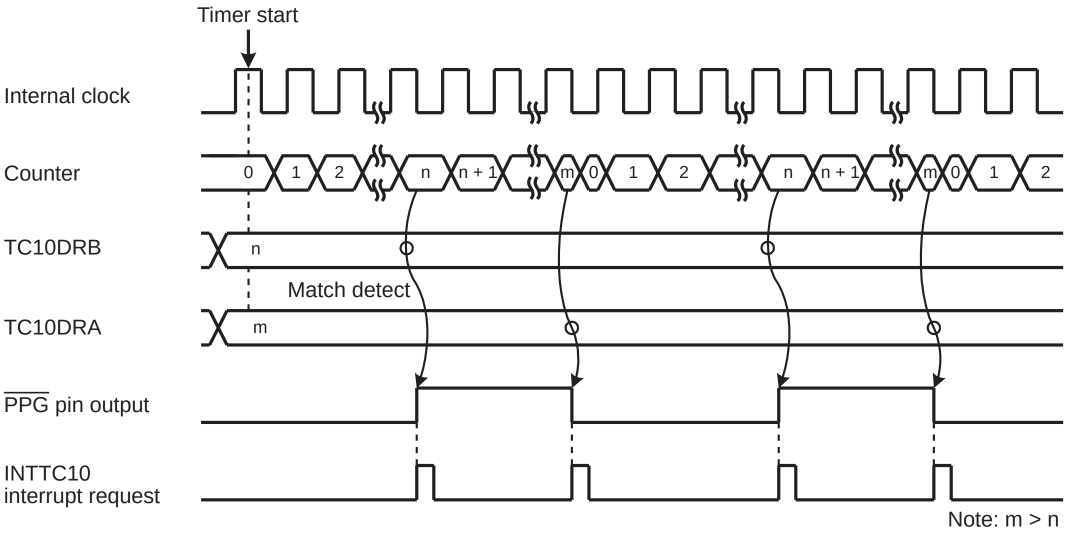
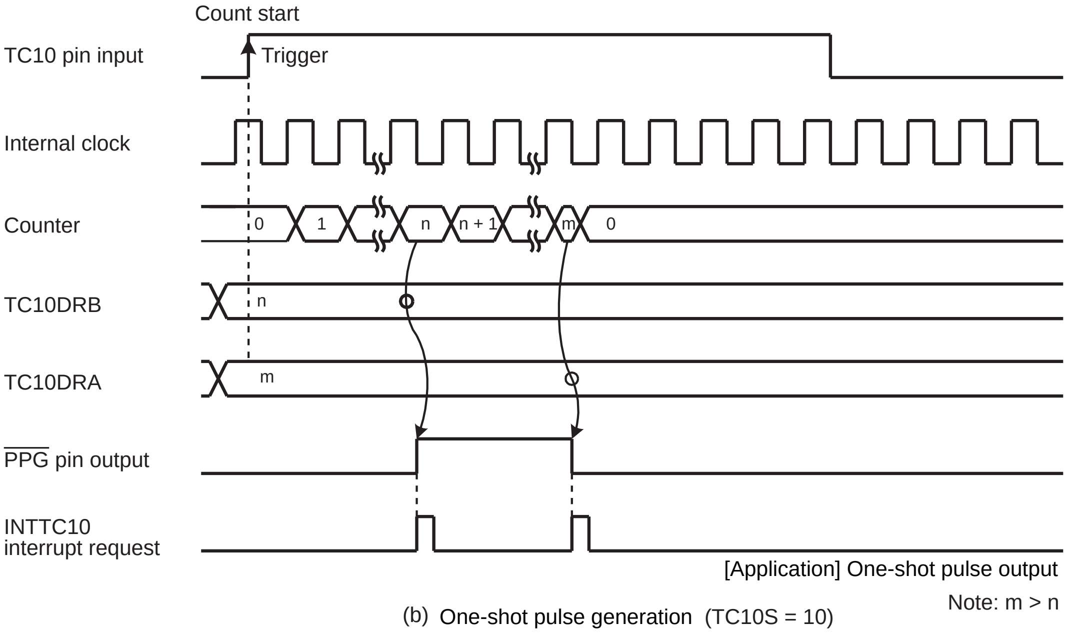


Figure 8-7 $\overline{\text{PPG}}$ Output



(a) Continuous pulse generation (TC10S = 01)

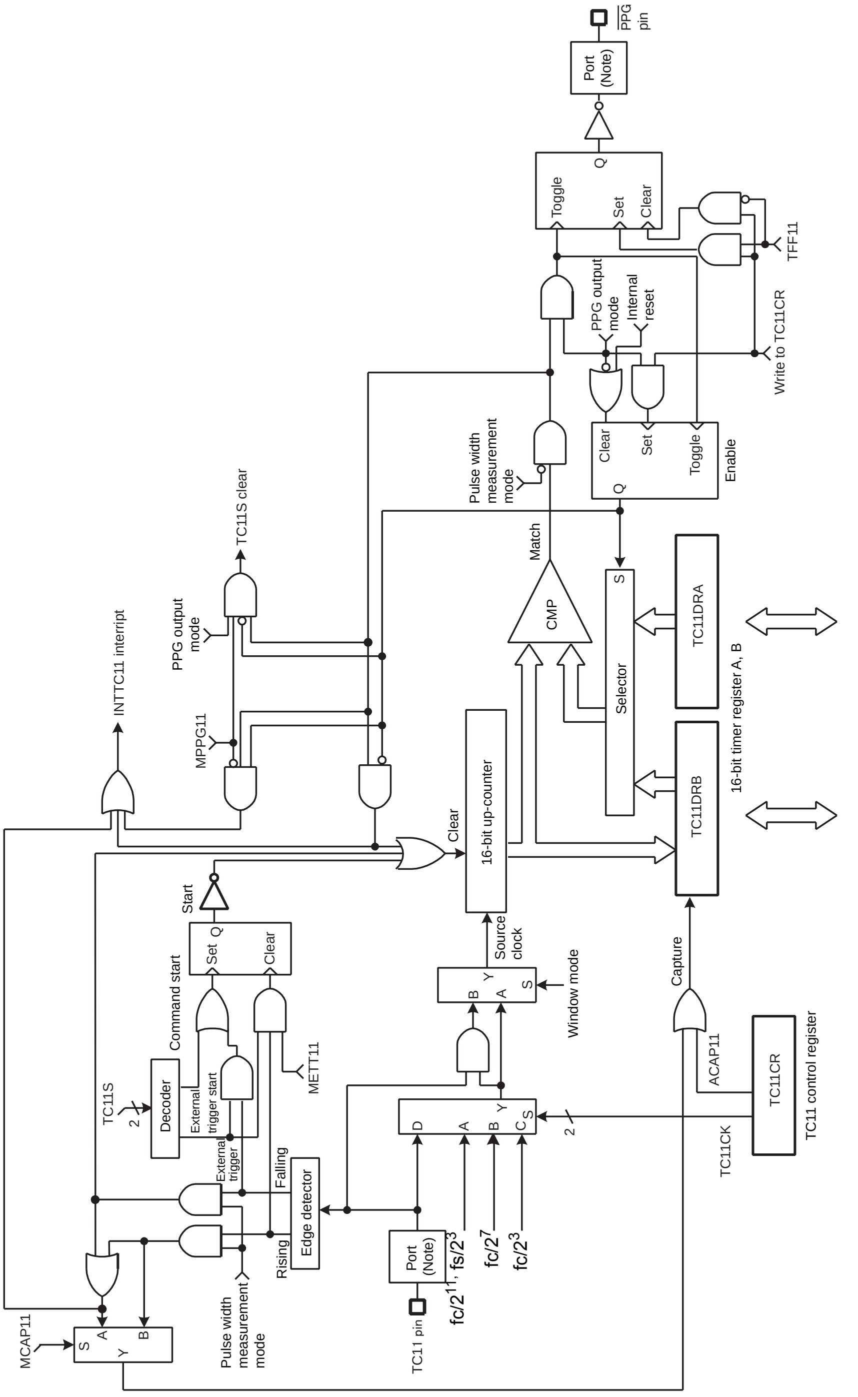


(b) One-shot pulse generation (TC10S = 10)

Figure 8-8 PPG Mode Timing Chart

8.2 16-Bit TimerCounter 11

8.2.1 Configuration



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

Figure 8-9 TimerCounter 11 (TC11)

8.2.2 TimerCounter Control

The TimerCounter 11 is controlled by the TimerCounter 11 control register (TC11CR) and two 16-bit timer registers (TC11DRA and TC11DRB).

Timer Register

<table>
<tr><td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>TC11DRA
(0021H, 0020H)</td><td colspan="8">TC11DRAH (0021H)</td><td colspan="8">TC11DRAL (0020H)</td></tr>
<tr><td></td><td colspan="8">(Initial value: 1111 1111 1111 1111)</td><td colspan="8">Read/Write</td></tr>
<tr><td>TC11DRB
(0023H, 0022H)</td><td colspan="8">TC11DRBH (0023H)</td><td colspan="8">TC11DRBL (0022H)</td></tr>
<tr><td></td><td colspan="8">(Initial value: 1111 1111 1111 1111)</td><td colspan="8">Read/Write (Write enabled only in the PPG output mode)</td></tr>
</table>

TimerCounter 11 Control Register

<table>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td>TC11CR
(0024H)</td><td>TFF11</td><td>ACAP11
MCAP11
METT11
MPPG11</td><td colspan="2">TC11S</td><td colspan="2">TC11CK</td><td colspan="2">TC11M</td><td>Read/Write
(Initial value: 0000 0000)</td></tr>
</table>

<table>
<tr><td>TFF11</td><td>Timer F/F11 control</td><td colspan="2">0: Clear</td><td colspan="6">1: Set</td><td>R/W</td></tr>
<tr><td>ACAP11</td><td>Auto capture control</td><td colspan="2">0:Auto-capture disable</td><td colspan="6">1:Auto-capture enable</td><td rowspan="4">R/W</td></tr>
<tr><td>MCAP11</td><td>Pulse width measurement mode control</td><td colspan="2">0:Double edge capture</td><td colspan="6">1:Single edge capture</td></tr>
<tr><td>METT11</td><td>External trigger timer mode control</td><td colspan="2">0:Trigger start</td><td colspan="6">1:Trigger start and stop</td></tr>
<tr><td>MPPG11</td><td>PPG output control</td><td colspan="2">0:Continuous pulse generation</td><td colspan="6">1:One-shot</td></tr>
<tr><td rowspan="5">TC11S</td><td rowspan="5">TC11 start control</td><td colspan="2"></td><td>Timer</td><td>Extrigger</td><td>Event</td><td>Window</td><td>Pulse</td><td>PPG</td><td rowspan="5">R/W</td></tr>
<tr><td colspan="2">00: Stop and counter clear</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr>
<tr><td colspan="2">01: Command start</td><td>O</td><td>–</td><td>–</td><td>–</td><td>–</td><td>O</td></tr>
<tr><td colspan="2">10: Rising edge start
(Ex-trigger/Pulse/PPG)
Rising edge count (Event)
Positive logic count (Window)</td><td>–</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr>
<tr><td colspan="2">11: Falling edge start
(Ex-trigger/Pulse/PPG)
Falling edge count (Event)
Negative logic count (Window)</td><td>–</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr>
<tr><td rowspan="6">TC11CK</td><td rowspan="6">TC11 source clock select [Hz]</td><td rowspan="2"></td><td colspan="5">NORMAL1/2, IDLE1/2 mode</td><td rowspan="2">Divider</td><td rowspan="2">SLOW, SLEEP mode</td><td rowspan="6">R/W</td></tr>
<tr><td>DV7CK = 0</td><td colspan="4">DV7CK = 1</td></tr>
<tr><td>00</td><td>$fc/2^{11}$</td><td colspan="4">$fs/2^3$</td><td>DV9</td><td>$fs/2^3$</td></tr>
<tr><td>01</td><td>$fc/2^7$</td><td colspan="4">$fc/2^7$</td><td>DV5</td><td>–</td></tr>
<tr><td>10</td><td>$fc/2^3$</td><td colspan="4">$fc/2^3$</td><td>DV1</td><td>–</td></tr>
<tr><td>11</td><td colspan="7">External clock (TC11 pin input)</td></tr>
<tr><td>TC11M</td><td>TC11 operating mode select</td><td colspan="8">00: Timer/external trigger timer/event counter mode
01: Window mode
10: Pulse width measurement mode
11: PPG (Programmable pulse generate) output mode</td><td>R/W</td></tr>
</table>

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 (TC11DRAH and TC11DRBH) 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 (TC11DRAL and TC11DRBL) 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 TC11CR1 during TC11S=00. Set the timer F/F10 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.
TC11DRA > TC11DRB > 1 (PPG output mode), TC11DRA > 1 (other modes)

Note 6: Set TFF11 to “0” in the mode except PPG output mode.

Note 7: Set TC11DRB after setting TC11M to the PPG output mode.

Note 8: When the STOP mode is entered, the start control (TC11S) is cleared to “00” automatically, and the timer stops. After the STOP mode is exited, set the TC11S to use the timer counter again.

Note 9: Use the auto-capture function in the operative condition of TC11. 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 TC11DRB by the source clock of up-counter after setting TC11CR<ACAP11> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC11DRB for the first time.

8.2.3 Function

TimerCounter 11 has six types of operating modes: timer, external trigger timer, event counter, window, pulse width measurement, programmable pulse generator output modes.

8.2.3.1 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 (TC11DRA) value is detected, an INTTC11 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting. Setting TC11CR<ACAP11> to “1” captures the up-counter value into the timer register 1B (TC11DRB) with the auto-capture function. Use the auto-capture function in the operative condition of TC11. 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 TC11DRB by the source clock of up-counter after setting TC11CR<ACAP11> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC11DRB for the first time.

Table 8-3 Internal Source Clock for TimerCounter 11 (Example: fc = 16 MHz, fs = 32.768 kHz)

TC11CK	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	–	–

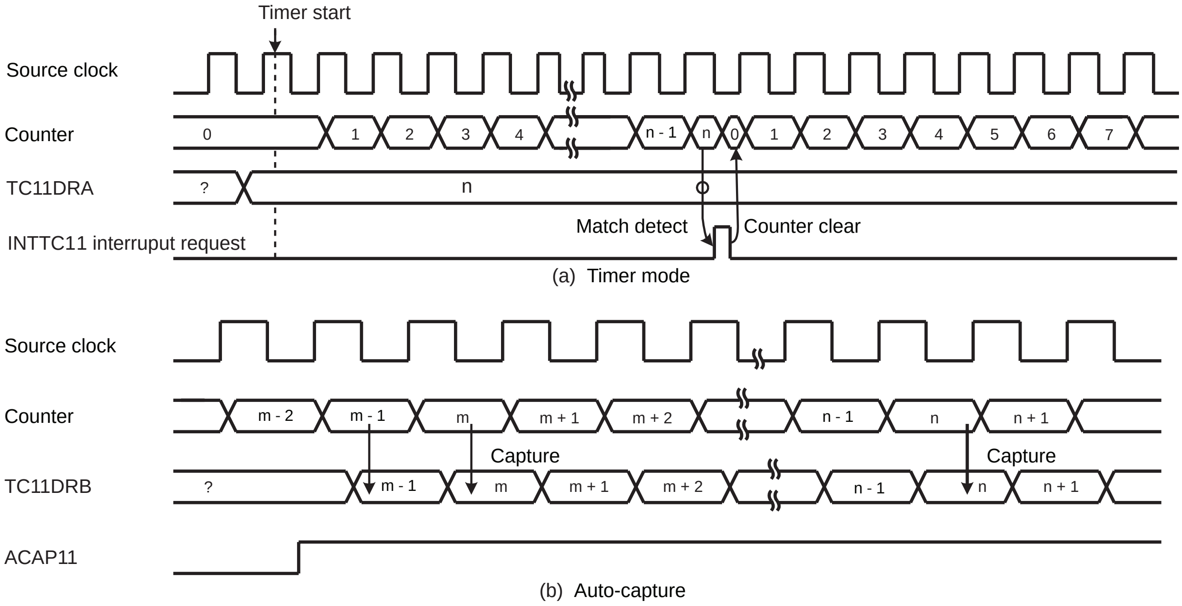
Example 1 :Setting the timer mode with source clock fc/2^{11} [Hz] and generating an interrupt 1 second later
(fc = 16 MHz, TBTCR<DV7CK> = “0”)

```
LDW     (TC11DRA), 1E84H      ; Sets the timer register (1 s ÷ 2^11/fc = 1E84H)
DI                            ; IMF= “0”
SET     (EIRL). 2             ; Enables INTTC11
EI                            ; IMF= “1”
LD      (TC11CR), 00000000B   ; Selects the source clock and mode
LD      (TC11CR), 00010000B   ; Starts TC11
```

Example 2 :Auto-capture

```
LD      (TC11CR), 01010000B   ; ACAP11 ← 1
:       :
LD      WA, (TC11DRB)         ; Reads the capture value
```

Note: Since the up-counter value is captured into TC11DRB by the source clock of up-counter after setting TC11CR<ACAP11> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC11DRB for the first time.



(a) Timer mode

(b) Auto-capture

Figure 8-10 Timer Mode Timing Chart

8.2.3.2 External Trigger Timer Mode

In the external trigger timer mode, the up-counter starts counting by the input pulse triggering of the TC11 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 TC11CR<TC11S>.

- When TC11CR<METT11> is set to “1” (trigger start and stop)

  When a match between the up-counter and the TC11DRA value is detected after the timer starts, the up-counter is cleared and halted and an INTTC11 interrupt request is generated.

  If the edge opposite to trigger edge is detected before detecting a match between the up-counter and the TC11DRA, 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 TC11CR<METT11> is set to “0” (trigger start)

  When a match between the up-counter and the TC11DRA value is detected after the timer starts, the up-counter is cleared and halted and an INTTC11 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 TC11DRA.

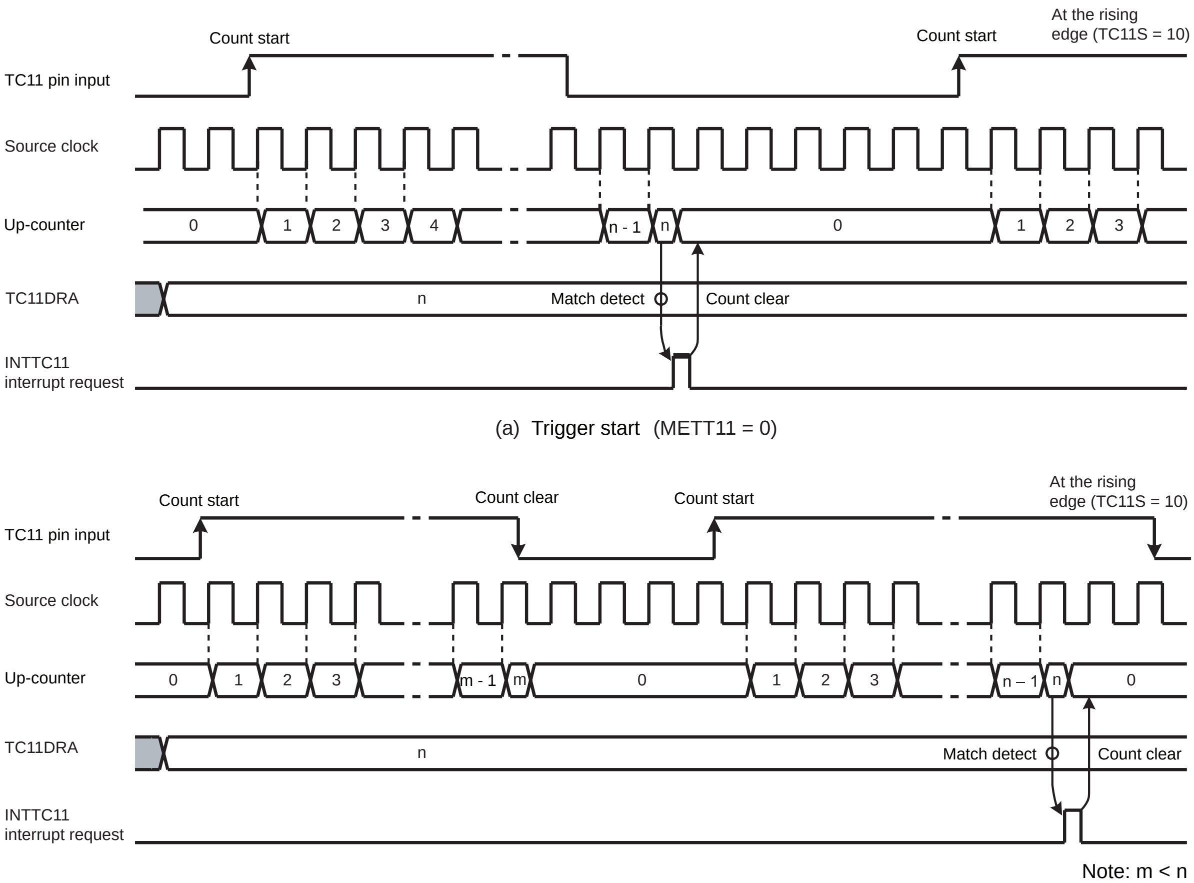
Since the TC11 pin input has the noise rejection, pulses of 4/fc [s] or less are rejected as noise. A pulse width of 12/fc [s] or more is required to ensure edge detection. The rejection circuit is turned off in the SLOW1/2 or SLEEP1/2 mode, but a pulse width of one machine cycle or more is required.

Example 1 :Generating an interrupt 1 ms after the rising edge of the input pulse to the TC11 pin (fc =16 MHz)

```
LDW     (TC11DRA), 007DH         ; 1ms ÷ 2^7/fc = 7DH
DI                               ; IMF= “0”
SET     (EIRL). 2                ; Enables INTTC11 interrupt
EI                               ; IMF= “1”
LD      (TC11CR), 00000100B      ; Selects the source clock and mode
LD      (TC11CR), 00100100B      ; Starts TC11 external trigger, METT11 = 0
```

Example 2 :Generating an interrupt when the low-level pulse with 4 ms or more width is input to the TC11 pin (fc =16 MHz)

```
LDW     (TC11DRA), 01F4H         ; 4 ms ÷ 2^7/fc = 1F4H
DI                               ; IMF= “0”
SET     (EIRL). 2                ; Enables INTTC11 interrupt
EI                               ; IMF= “1”
LD      (TC11CR), 00000100B      ; Selects the source clock and mode
LD      (TC11CR), 01110100B      ; Starts TC11 external trigger, METT11 = 0
```



(a) Trigger start (METT11 = 0)

(b) Trigger start and stop (METT11 = 1)

Figure 8-11 External Trigger Timer Mode Timing Chart

8.2.3.3 Event Counter Mode

In the event counter mode, the up-counter counts up at the edge of the input pulse to the TC11 pin. Either the rising or falling edge of the input pulse is selected as the count up edge in TC11CR<TC11S>.

When a match between the up-counter and the TC11DRA value is detected, an INTTC11 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 TC11 pin. Since a match between the up-counter and the value set to TC11DRA is detected at the edge opposite to the selected edge, an INTTC11 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 TC11 pin.

Setting TC11CR<ACAP11> to “1” captures the up-counter value into TC11DRB with the auto capture function. Use the auto-capture function in the operative condition of TC11. 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 TC11DRB by the source clock of up-counter after setting TC11CR<ACAP11> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC11DRB for the first time.

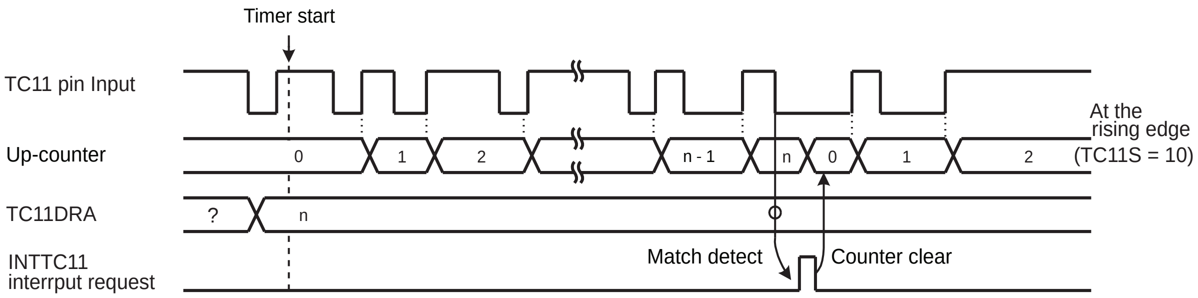


Figure 8-12 Event Counter Mode Timing Chart

Table 8-4 Input Pulse Width to TC11 Pin

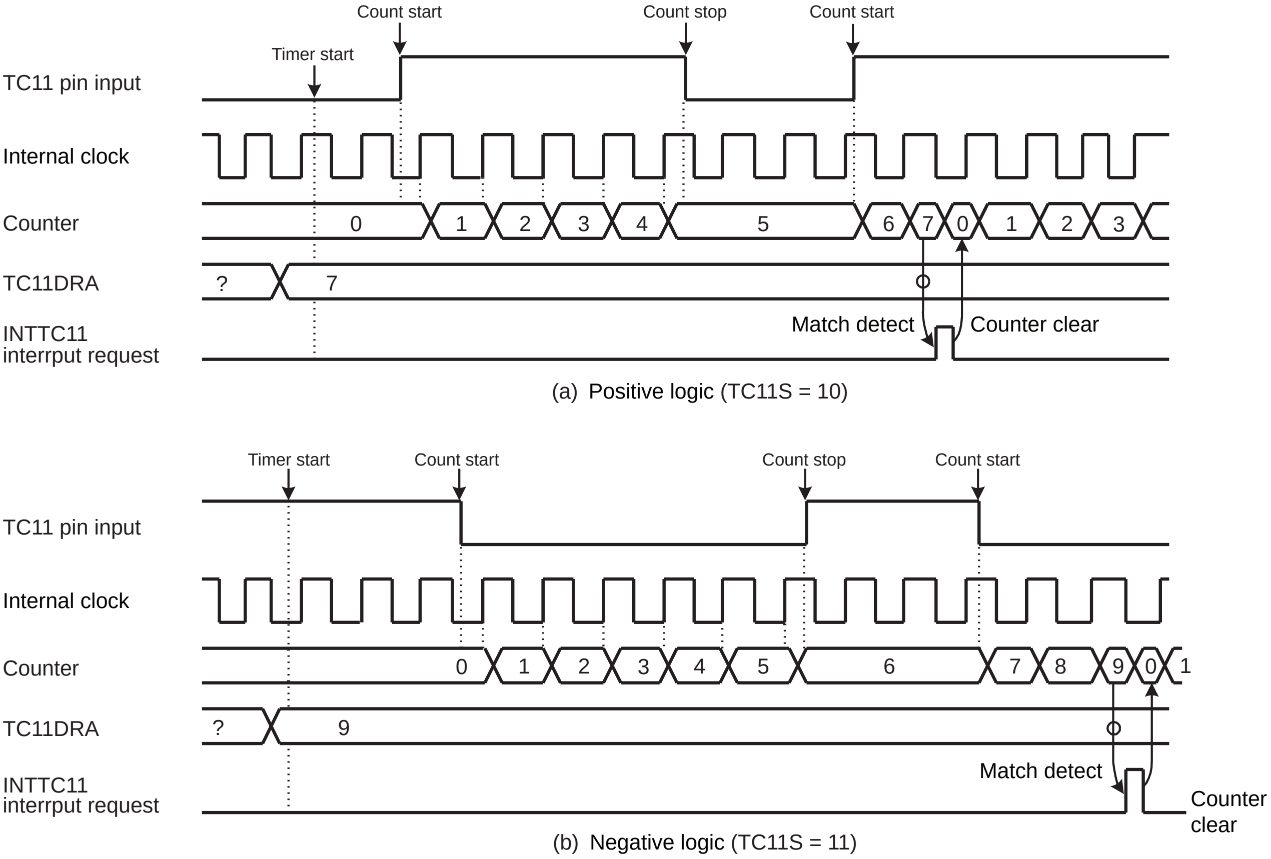
	Minimum Pulse Width [s]	
	NORMAL1/2, IDLE1/2 Mode	SLOW1/2, SLEEP1/2 Mode
High-going	2^3/fc	2^3/fs
Low-going	2^3/fc	2^3/fs

8.2.3.4 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 TC11 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 TC11DRA value is detected, an INTTC11 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 TC11CR<TC11CK>.



(a) Positive logic (TC11S = 10)

(b) Negative logic (TC11S = 11)

Figure 8-13 Window Mode Timing Chart

8.2.3.5 Pulse Width Measurement Mode

In the pulse width measurement mode, the up-counter starts counting by the input pulse triggering of the TC11 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 TC11CR<TC11S>. Either the single- or double-edge capture is selected as the trigger edge in TC11CR<MCAP11>.

- When TC11CR<MCAP11> 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 TC11CR<TC11S>. To measure the low-level input pulse width, set the falling edge to TC11CR<TC11S>.

  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 TC11DRB and generates an INTTC11 interrupt request. The up-counter is cleared at this time, and then restarts counting when detecting the trigger edge used to start counting.

- When TC11CR<MCAP11> 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 TC11CR<TC11S>. To measure the cycle starting with the low-going pulse, set the falling edge to TC11CR<TC11S>.

  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 TC11DRB and generates an INTTC11 interrupt request. The up-counter continues counting up, and captures the up-counter value into TC11DRB and generates an INTTC11 interrupt request when detecting the trigger edge used to start counting. The up-counter is cleared at this time, and then continues counting.

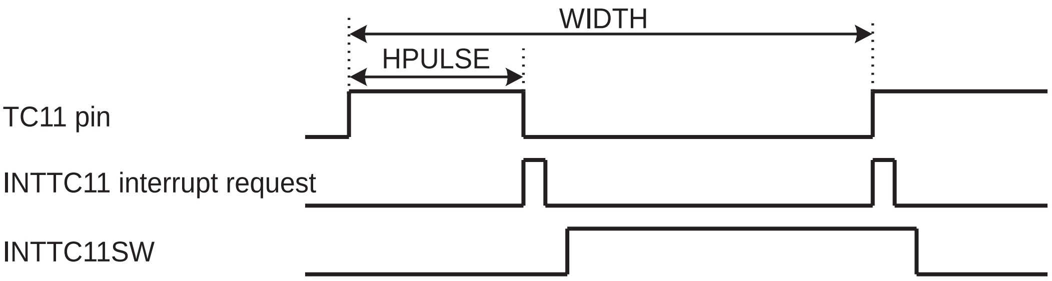
Note 1: The captured value must be read from TC11DRB until the next trigger edge is detected. If not read, the captured value becomes a don’t care. It is recommended to use a 16-bit access instruction to read the captured value from TC11DRB.

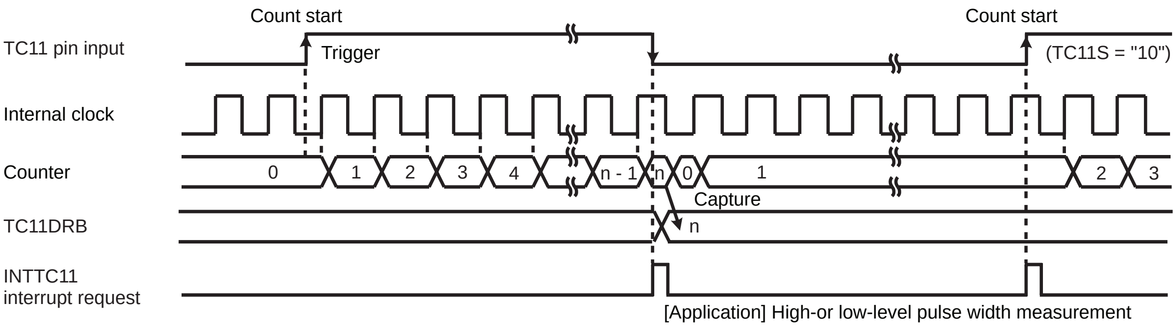
Note 2: For the single-edge capture, the counter after capturing the value stops at “1” until detecting the next edge. Therefore, the second captured value is “1” larger than the captured value immediately after counting starts.

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

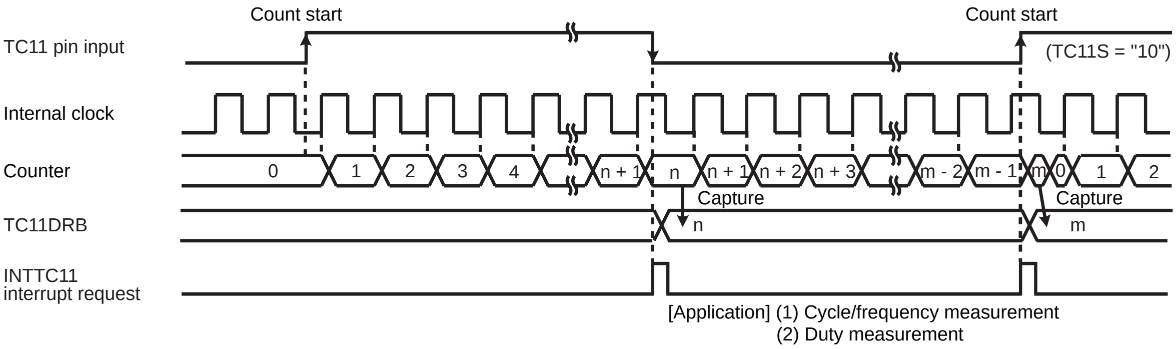
Example :Duty measurement (resolution fc/2^7 [Hz])

```
                    CLR     (INTTC11SW). 0        ; INTTC11 service switch initial setting
                                                    Address set to convert INTTC11SW at each INTTC11
                    LD      (TC11CR), 00000110B   ; Sets the TC11 mode and source clock
                    DI                            ; IMF= "0"
                    SET     (EIRH). 7             ; Enables INTTC11
                    EI                            ; IMF= "1"
                    LD      (TC11CR), 00100110B   ; Starts TC11 with an external trigger at MCAP11 = 0
                     :
        PINTTC11:   CPL     (INTTC11SW). 0        ; INTTC11 interrupt, inverts and tests INTTC11 service switch
                    JRS     F, SINTTC11
                    LD      A, (TC11DRBL)         ; Reads TC11DRB (High-level pulse width)
                    LD      W,(TC11DRBH)
                    LD      (HPULSE), WA          ; Stores high-level pulse width in RAM
                    RETI
        SINTTC11:   LD      A, (TC11DRBL)         ; Reads TC11DRB (Cycle)
                    LD      W,(TC11DRBH)
                    LD      (WIDTH), WA           ; Stores cycle in RAM
                     :
                    RETI                          ; Duty calculation
                    :
        VINTTC11:   DW      PINTTC11              ; INTTC11 Interrupt vector
```





(a) Single-edge capture (MCAP11 = "1")



(b) Double-edge capture (MCAP11 = "0")

Figure 8-14 Pulse Width Measurement Mode

8.2.3.6 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, TC11CR<TC11S> specifies either the edge of the input pulse to the TC11 pin or the command start. TC11CR<MPPG11> specifies whether a duty pulse is produced continuously or not (one-shot pulse).

- When TC11CR<MPPG11> is set to "0" (Continuous pulse generation)

  When a match between the up-counter and the TC11DRB value is detected after the timer starts, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC11 interrupt request is generated. The up-counter continues counting. When a match between the up-counter and the TC11DRA value is detected, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC11 interrupt request is generated. The up-counter is cleared at this time, and then continues counting and pulse generation.

  When TC11S is cleared to "00" during PPG output, the $\overline{\text{PPG}}$ pin retains the level immediately before the counter stops.

- When TC11CR<MPPG11> is set to "1" (One-shot pulse generation)

  When a match between the up-counter and the TC11DRB value is detected after the timer starts, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC11 interrupt request is generated. The up-counter continues counting. When a match between the up-counter and the TC11DRA value is detected, the level of the $\overline{\text{PPG}}$ pin is inverted and an INTTC11 interrupt request is generated. TC11CR<TC11S> 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 TC11CR<TFF11> 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 TC11CR<TFF11> 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 TC11DRA or TC11DRB 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 TC11CR<TFF11> during a run of the timer. TC11CR<TFF11> can be set correctly only at initialization (after reset). When the timer stops during PPG, TC11CR<TFF11> 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 TC11CR<TFF11> 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 TC11CR<TC11M> to the timer mode (it is not required to start the timer mode), and then set the PPG mode. Set TC11CR<TFF11> at this time.

Note 3: In the PPG mode, the following relationship must be satisfied.
TC11DRA > TC11DRB

Note 4: Set TC11DRB after changing the mode of TC11M to the PPG mode.

Example :Generating a pulse which is high-going for 800 µs and low-going for 200 µs
(fc = 16 MHz)

```
                Setting port

LD              (TC11CR), 10000111B       ; Sets the PPG mode, selects the source clock

LDW             (TC11DRA), 007DH          ; Sets the cycle (1 ms ÷ 2^7/fc ms = 007DH)

LDW             (TC11DRB), 0019H          ; Sets the low-level pulse width (200 µs ÷ 2^7/fc = 0019H)

LD              (TC11CR), 10010111B       ; Starts the timer
```

Example :After stopping PPG, setting the PPG pin to a high-level to restart PPG
(fc = 16 MHz)

```
                Setting port

LD              (TC11CR), 10000111B       ; Sets the PPG mode, selects the source clock

LDW             (TC11DRA), 007DH          ; Sets the cycle (1 ms ÷ 2^7/fc µs = 007DH)

LDW             (TC11DRB), 0019H          ; Sets the low-level pulse width (200 µs ÷ 2^7/fc = 0019H)

LD              (TC11CR), 10010111B       ; Starts the timer

:               :

LD              (TC11CR), 10000111B       ; Stops the timer

LD              (TC11CR), 10000100B       ; Sets the timer mode

LD              (TC11CR), 00000111B       ; Sets the PPG mode, TFF11 = 0

LD              (TC11CR), 00010111B       ; Starts the timer
```

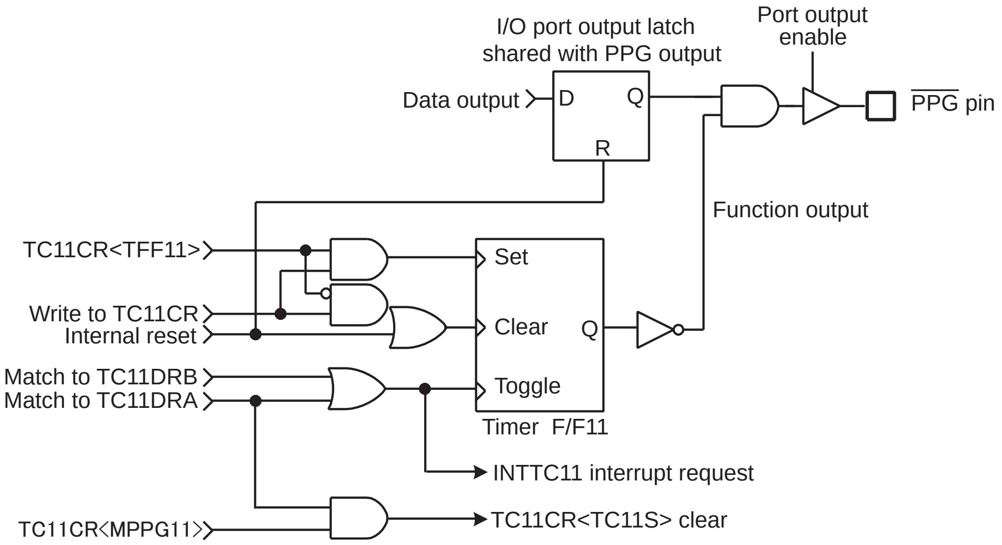
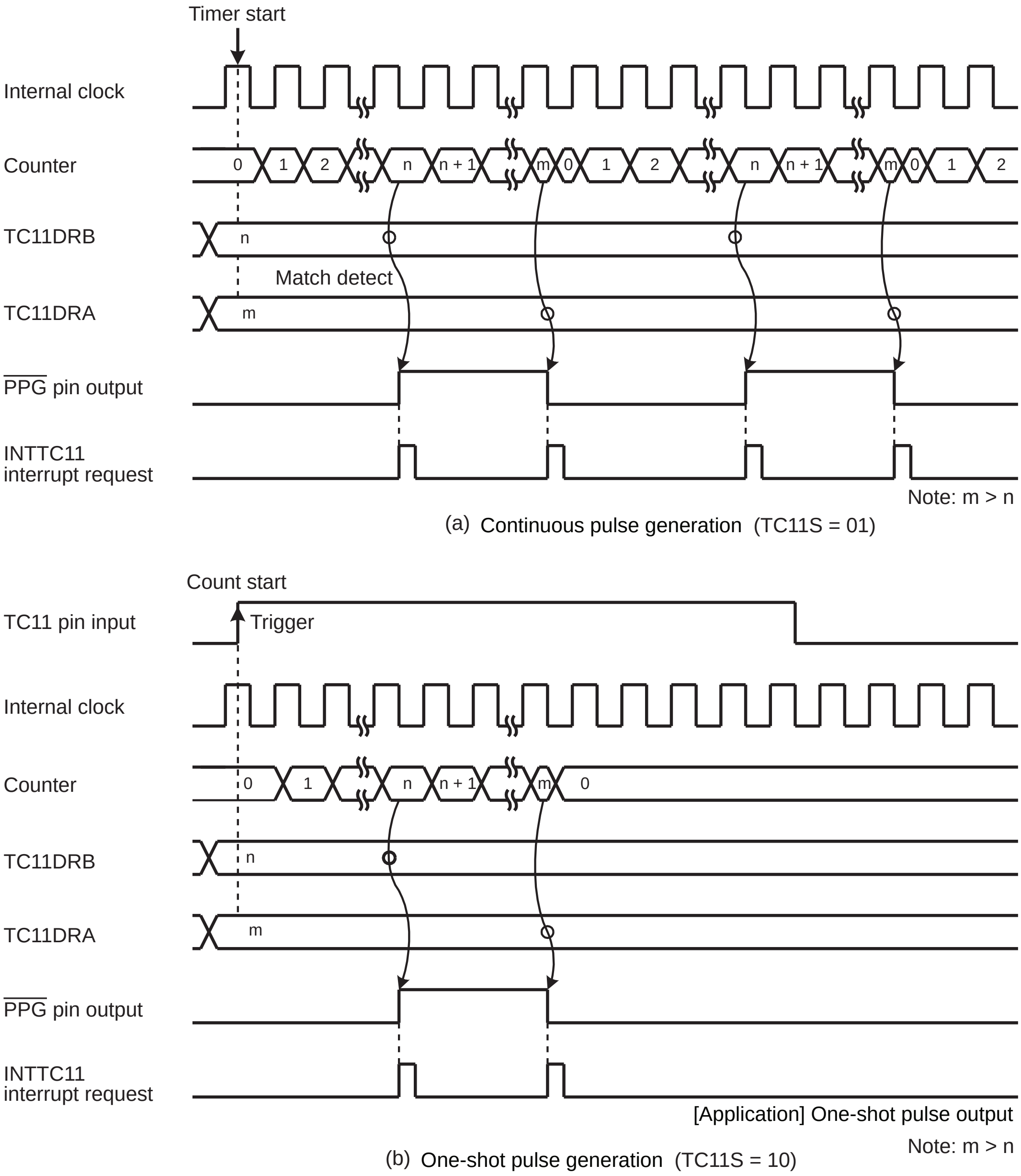


Figure 8-15 $\overline{\text{PPG}}$ Output



(a) Continuous pulse generation (TC11S = 01)

(b) One-shot pulse generation (TC11S = 10)

Figure 8-16 PPG Mode Timing Chart

9. 8-Bit TimerCounter (TC3, TC4)

9.1 Configuration

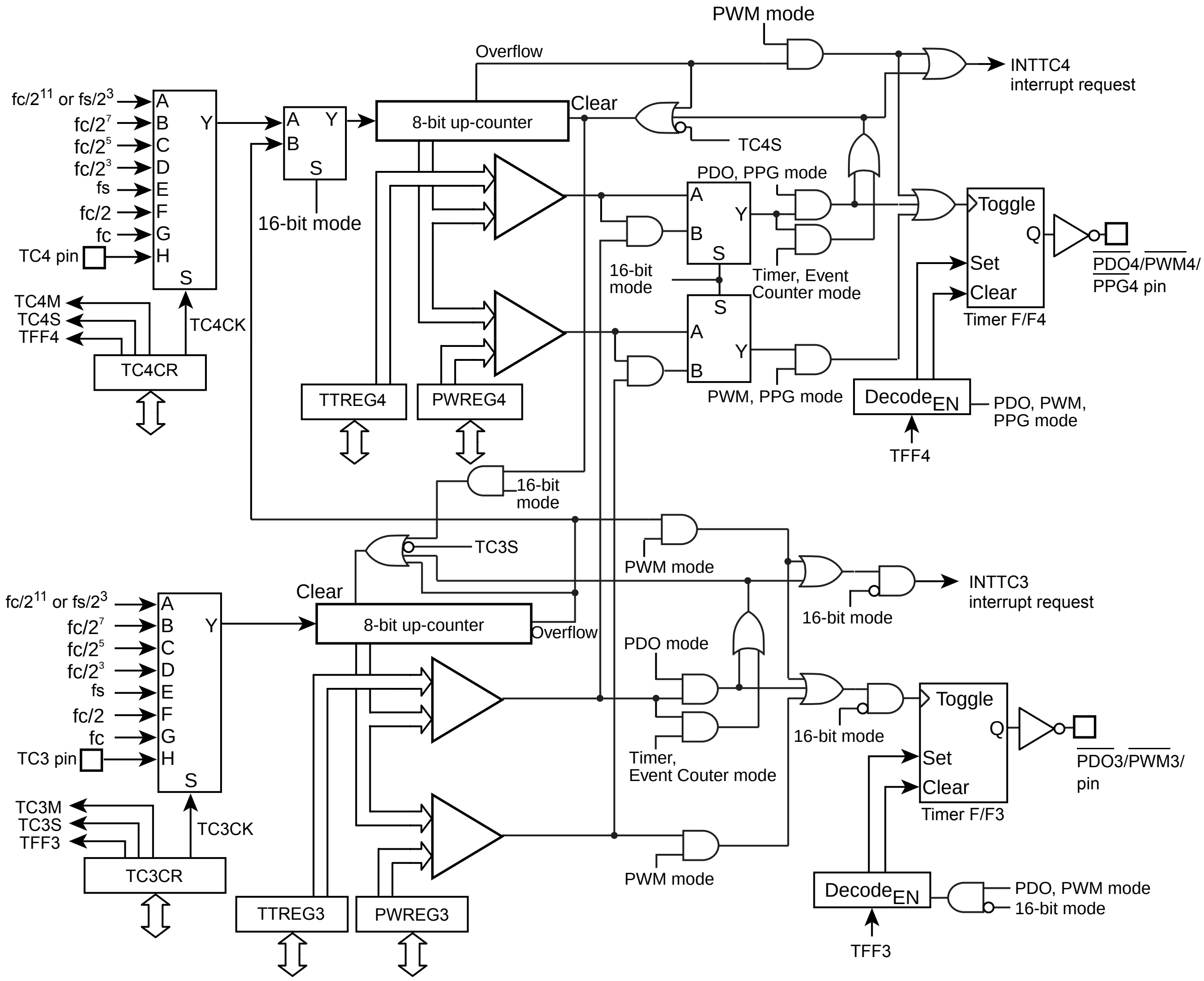


Figure 9-1 8-Bit TimerCounter 3, 4

9.2 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

TTREG3 (0015H) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

PWREG3 (0019H) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

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

TC3CR (0009H)	7	6	5	4	3	2	1	0	
	TFF3	TC3CK			TC3S	TC3M			(Initial value: 0000 0000)

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	R/W
			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 9-1 and Table 9-2.

Note 7: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 9-3.

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

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

TTREG4 (0016H) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

PWREG4 (001AH) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

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

TC4CR (000AH)	7	6	5	4	3	2	1	0	
	TFF4	TC4CK			TC4S	TC4M			(Initial value: 0000 0000)

TFF4	Timer F/F4 control	0: Clear 1: Set				R/W
TC4CK	Operating clock selection [Hz]		NORMAL1/2, IDLE1/2 mode		SLOW1/2 SLEEP1/2 mode	R/W
			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	–	
		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 9-1 and Table 9-2.

Note 8: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 9-3.

Table 9-1 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	O	O	O	O	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	O	O
8-bit PDO	O	O	O	O	–	–	–	–	–
8-bit PWM	O	O	O	O	O	O	O	–	–
16-bit timer	O	O	O	O	–	–	–	–	–
16-bit event counter	–	–	–	–	–	–	–	O	–
Warm-up counter	–	–	–	–	O	–	–	–	–
16-bit PWM	O	O	O	O	O	O	O	O	–
16-bit PPG	O	O	O	O	–	–	–	O	–

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: O : Available source clock

Table 9-2 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	O	–	–	–	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	O	O
8-bit PDO	O	–	–	–	–	–	–	–	–
8-bit PWM	O	–	–	–	O	–	–	–	–
16-bit timer	O	–	–	–	–	–	–	–	–
16-bit event counter	–	–	–	–	–	–	–	O	–
Warm-up counter	–	–	–	–	–	–	O	–	–
16-bit PWM	O	–	–	–	O	–	–	O	–
16-bit PPG	O	–	–	–	–	–	–	O	–

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: O : Available source clock

Table 9-3 Constraints on Register Values Being Compared

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

Note: n = 3 to 4

9.3 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.

9.3.1 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 $\overline{\text{PDOj}}$, $\overline{\text{PWMj}}$ and $\overline{\text{PPGj}}$ 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 9-4 Source Clock for TimerCounter 3, 4 (Internal Clock)

Source Clock			Resolution		Maximum Time Setting	
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	32.6 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/2^7 Hz and generating an interrupt 80 µs later (TimerCounter4, fc = 16.0 MHz)

```
LD     (TTREG4), 0AH          : Sets the timer register (80 µs÷2^7/fc = 0AH).
DI
SET    (EIRE). 5              : Enables INTTC4 interrupt.
EI
LD     (TC4CR), 00010000B     : Sets the operating clock to fc/2^7, and 8-bit timer mode.
LD     (TC4CR), 00011000B     : Starts TC4.
```

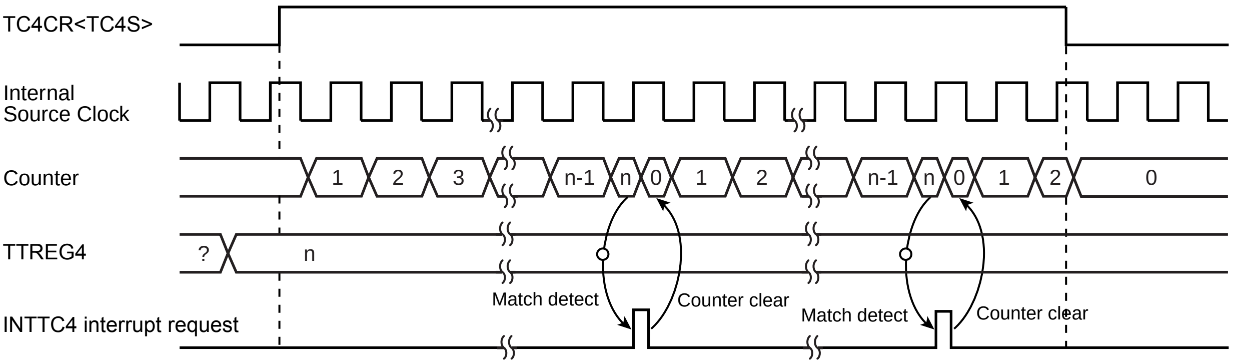


Figure 9-2 8-Bit Timer Mode Timing Chart (TC4)

9.3.2 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 $\overline{\text{PDOj}}$, $\overline{\text{PWMj}}$ and $\overline{\text{PPGj}}$ 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 TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

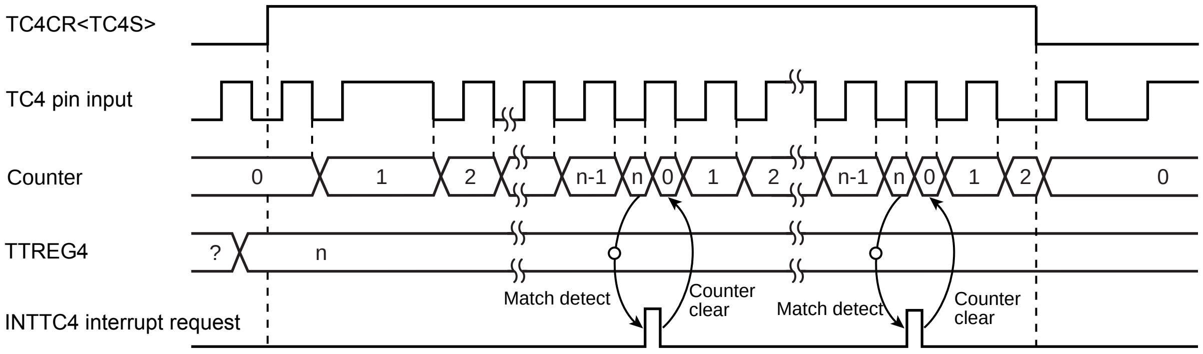


Figure 9-3 8-Bit Event Counter Mode Timing Chart (TC4)

9.3.3 8-Bit Programmable Divider Output (PDO) Mode (TC3, 4)

This mode is used to generate a pulse with a 50% duty cycle from the $\overline{\text{PDOj}}$ 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 $\overline{\text{PDOj}}$ 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 $\overline{\text{PDOj}}$ 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 (fc = 16.0 MHz)

```
          Setting port
LD        (TTREG4),  3DH               : 1/1024÷2^7/fc÷2 = 3DH
LD        (TC4CR),   00010001B         : Sets the operating clock to fc/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 $\overline{\text{PDOj}}$ 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 $\overline{\text{PDOj}}$ pin to the high level when the TimerCounter is stopped
CLR (TCjCR).3: Stops the timer.
CLR (TCjCR).7: Sets the $\overline{\text{PDOj}}$ pin to the high level.

Note 3: j = 3, 4

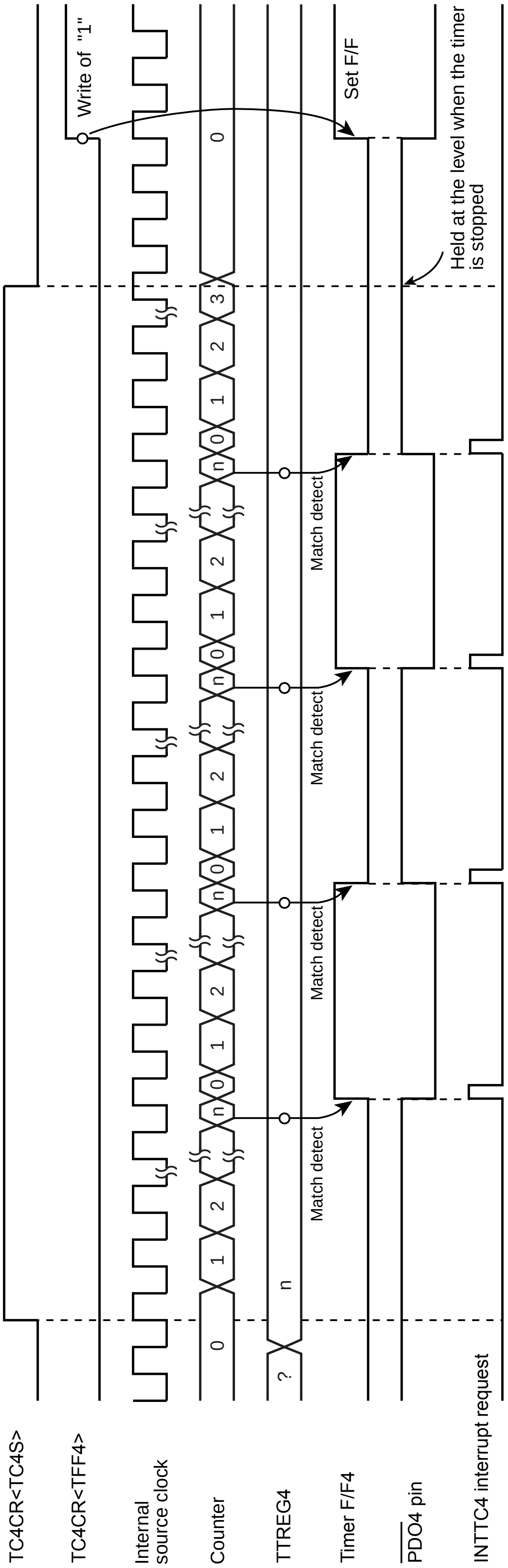


Figure 9-4 8-Bit PDO Mode Timing Chart (TC4)

9.3.4 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 $\overline{\text{PWMj}}$ 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 $\overline{\text{PWMj}}$ 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 $\overline{\text{PWMj}}$ pin to the high level when the TimerCounter is stopped
CLR (TCjCR).3: Stops the timer.
CLR (TCjCR).7: Sets the $\overline{\text{PWMj}}$ 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 $\overline{\text{PWMj}}$ pin during the warm-up period time after exiting the STOP mode.

Note 4: j = 3, 4

Table 9-5 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	–

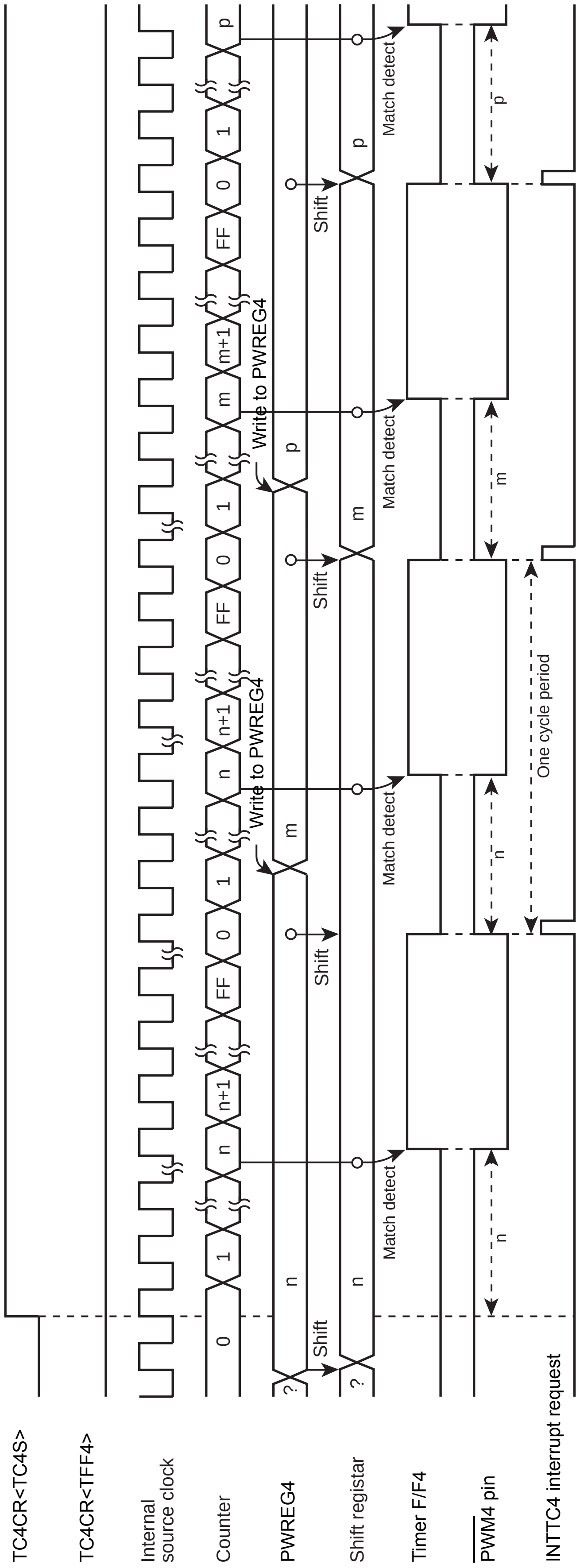


Figure 9-5 8-Bit PWM Mode Timing Chart (TC4)

9.3.5 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 cascadable 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 $\overline{\text{PDOj}}$, $\overline{\text{PWMj}}$, and $\overline{\text{PPGj}}$ 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 9-6 Source Clock for 16-Bit Timer Mode

Source Clock			Resolution		Maximum Time Setting	
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}	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/2^7 Hz, and generating an interrupt 300 ms later (fc = 16.0 MHz)

```
LDW     (TTREG3), 927CH     : Sets the timer register (300 ms÷2^7/fc = 927CH).
DI
SET     (EIRE). 5           : Enables INTTC4 interrupt.
EI
LD      (TC3CR), 13H        :Sets the operating clock to fc/2^7, and 16-bit timer mode
                             (lower byte).
LD      (TC4CR), 04H        : Sets the 16-bit timer mode (upper byte).
LD      (TC4CR), 0CH        : Starts the timer.
```

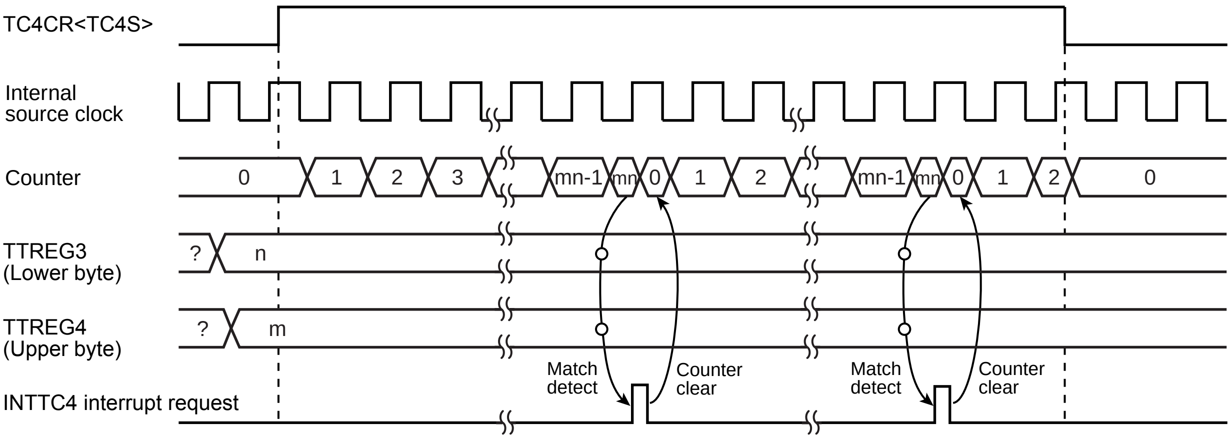


Figure 9-6 16-Bit Timer Mode Timing Chart (TC3 and TC4)

9.3.6 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 fc/2^4 Hz in the NORMAL1/2 or IDLE1/2 mode, and fs/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 $\overline{\text{PDOj}}$, $\overline{\text{PWMj}}$ and $\overline{\text{PPGj}}$ 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

9.3.7 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 fc/2^4 Hz in the NORMAL1/2 or IDLE1/2 mode, and fs/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{\text{PWM}}$4 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{\text{PWM}}$4 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{\text{PWM}}$4 pin to the high level when the TimerCounter is stopped

CLR (TC4CR).3: Stops the timer.
CLR (TC4CR).7 : Sets the $\overline{\text{PWM}}$4 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 fc, fc/2 or fs is selected as the source clock, a pulse is output from the $\overline{\text{PWM}}$4 pin during the warm-up period time after exiting the STOP mode.

Table 9-7 16-Bit 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					
$fc/2^{11}$	$fs/2^3$ [Hz]	$fs/2^3$ [Hz]	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	–
fs	fs	fs	30.5 μs	30.5 μs	2 s	2 s
fc/2	fc/2	–	125 ns	–	8.2 ms	–
fc	fc	–	62.5 ns	–	4.1 ms	–

Example :Generating a pulse with 1-ms high-level width and a period of 32.768 ms (fc = 16.0 MHz)

```
        Setting ports
LDW     (PWREG3), 07D0H        : Sets the pulse width.
LD      (TC3CR), 33H           : Sets the operating clock to fc/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.
```

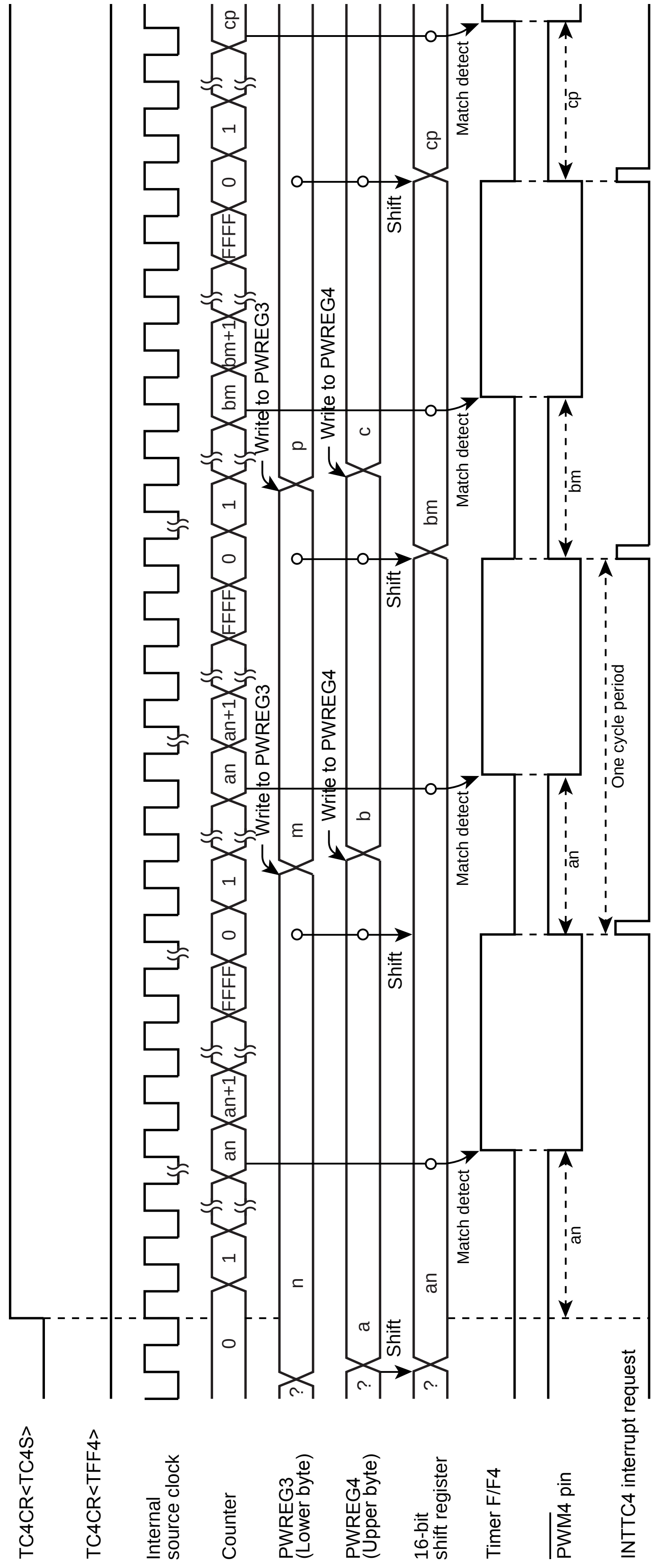


Figure 9-7 16-Bit PWM Mode Timing Chart (TC3 and TC4)

9.3.8 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 cascadable 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 fc/2^4 Hz in the NORMAL1/2 or IDLE1/2 mode, and fs/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{\text{PPG}}$4 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 (fc = 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 fc/2^3, and16-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{\text{PPG}}$4 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{\text{PPG}}$4 pin to the high level when the TimerCounter is stopped
CLR (TC4CR).3: Stops the timer
CLR (TC4CR).7: Sets the $\overline{\text{PPG}}$4 pin to the high level

Note 3: i = 3, 4

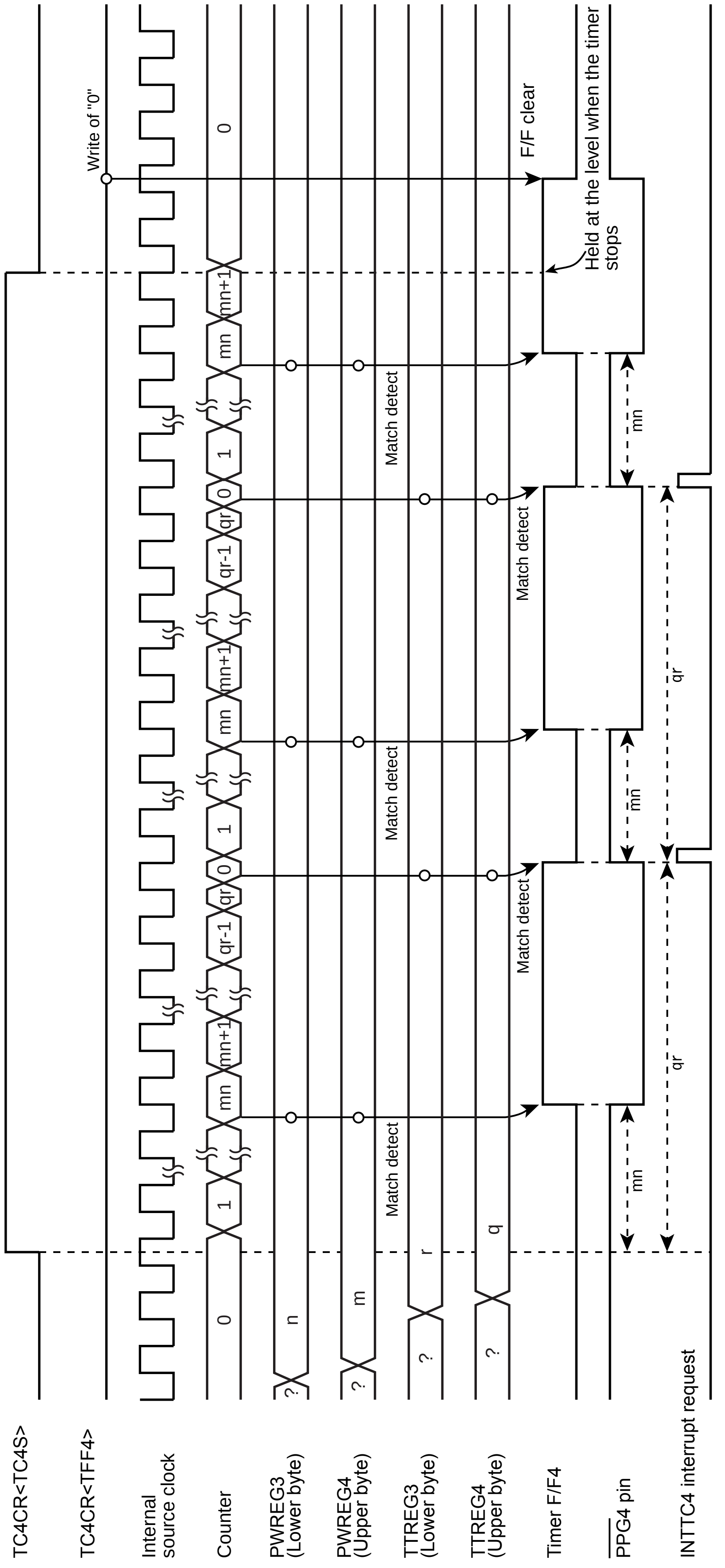


Figure 9-8 16-Bit PPG Mode Timing Chart (TC3 and TC4)

9.3.9 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 $\overline{\text{PDOi}}$, $\overline{\text{PWMi}}$ and $\overline{\text{PPGi}}$ 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

9.3.9.1 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 fs 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 9-8 Setting Time of Low-Frequency Warm-Up Counter Mode (fs = 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 fs, 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     (EIRE). 5          : 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
```

9.3.9.2 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 fc 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 9-9 Setting Time in High-Frequency Warm-Up Counter Mode

Minimum time Setting (TTREG4, 3 = 0100H)	Maximum time Setting (TTREG4, 3 = FF00H)
16 μs	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 fc, 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     (EIRE). 5           : 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
```

10. 8-Bit TimerCounter (TC5, TC6)

10.1 Configuration

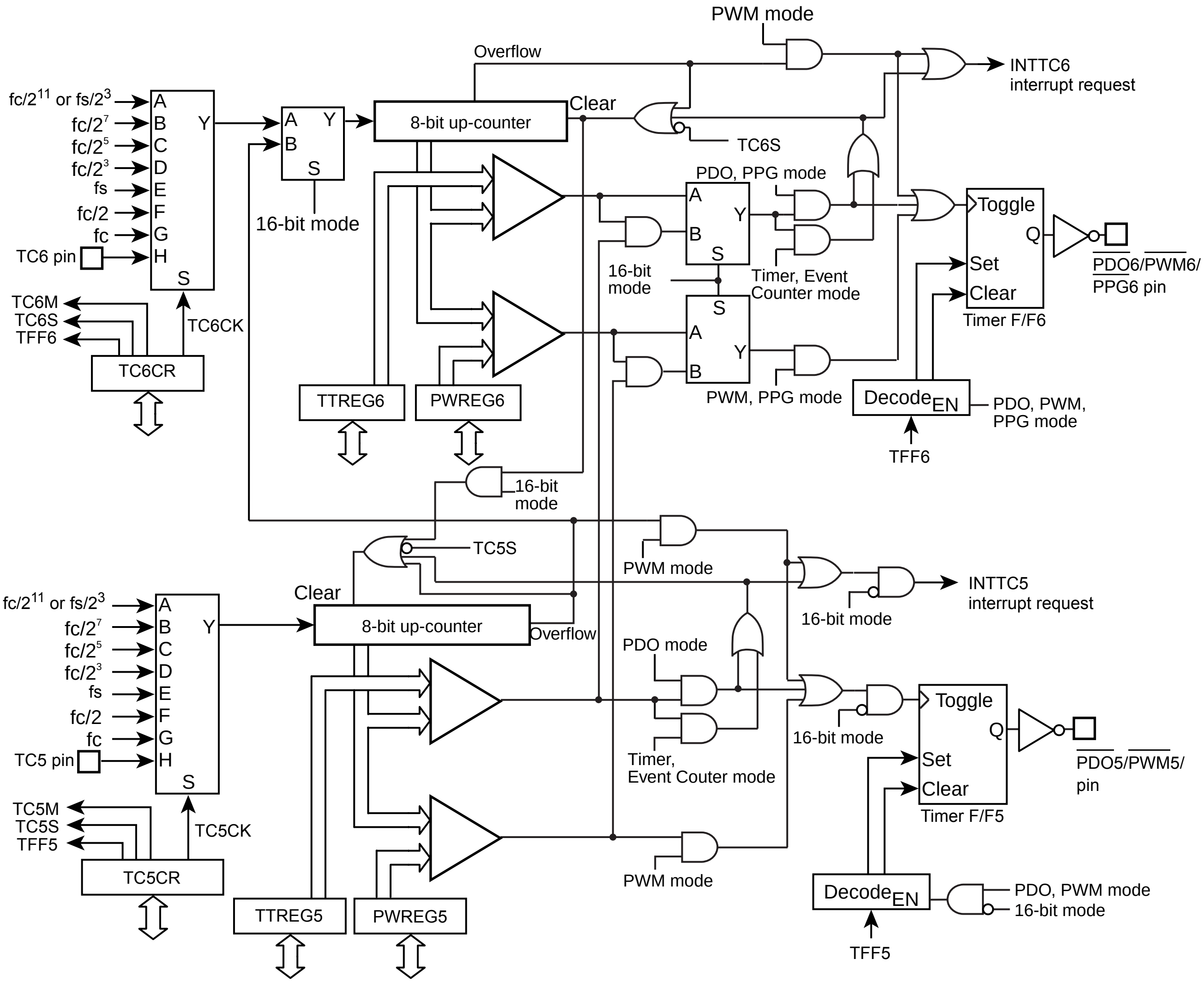


Figure 10-1 8-Bit TimerCounter 5, 6

10.2 TimerCounter Control

The TimerCounter 5 is controlled by the TimerCounter 5 control register (TC5CR) and two 8-bit timer registers (TTREG5, PWREG5).

TimerCounter 5 Timer Register

TTREG5	7	6	5	4	3	2	1	0	
(0017H) R/W									(Initial value: 1111 1111)

PWREG5	7	6	5	4	3	2	1	0	
(001BH) R/W									(Initial value: 1111 1111)

Note 1: Do not change the timer register (TTREG5) setting while the timer is running.

Note 2: Do not change the timer register (PWREG5) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

TimerCounter 5 Control Register

TC5CR	7	6	5	4	3	2	1	0	
(000BH)	TFF5	TC5CK			TC5S	TC5M			(Initial value: 0000 0000)

TFF5	Time F/F5 control	0: Clear 1: Set				R/W
TC5CK	Operating clock selection [Hz]		NORMAL1/2, IDLE1/2 mode		SLOW1/2 SLEEP1/2 mode	R/W
			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	TC5 pin input			
TC5S	TC5 start control	0: Operation stop and counter clear 1: Operation start				R/W
TC5M	TC5M 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 TC6M.) 1**: Reserved				R/W

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

Note 2: Do not change the TC5M, TC5CK and TFF5 settings while the timer is running.

Note 3: To stop the timer operation (TC5S= 1 → 0), do not change the TC5M, TC5CK and TFF5 settings. To start the timer operation (TC5S= 0 → 1), TC5M, TC5CK and TFF5 can be programmed.

Note 4: To use the TimerCounter in the 16-bit mode, set the operating mode by programming TC6CR<TC6M>, where TC5M must be fixed to 011.

Note 5: To use the TimerCounter in the 16-bit mode, select the source clock by programming TC5CK. Set the timer start control and timer F/F control by programming TC6CR<TC6S> and TC6CR<TFF6>, 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 fc in the SLOW or SLEEP mode can be used only as the high-frequency warm-up mode.

The TimerCounter 6 is controlled by the TimerCounter 6 control register (TC6CR) and two 8-bit timer registers (TTREG6 and PWREG6).

TimerCounter 6 Timer Register

TTREG6 (0018H) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

PWREG6 (001CH) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

Note 1: Do not change the timer register (TTREG6) setting while the timer is running.

Note 2: Do not change the timer register (PWREG6) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

TimerCounter 6 Control Register

TC6CR (000CH)	7	6	5	4	3	2	1	0	
	TFF6	TC6CK			TC6S	TC6M			(Initial value: 0000 0000)

TFF6	Timer F/F6 control	0: Clear 1: Set				R/W
TC6CK	Operating clock selection [Hz]		NORMAL1/2, IDLE1/2 mode		SLOW1/2 SLEEP1/2 mode	R/W
			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	–	
		111	TC6 pin input			
TC6S	TC6 start control	0: Operation stop and counter clear 1: Operation start				R/W
TC6M	TC6M 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 TC6M, TC6CK and TFF6 settings while the timer is running.

Note 3: To stop the timer operation (TC6S= 1 → 0), do not change the TC6M, TC6CK and TFF6 settings.
To start the timer operation (TC6S= 0 → 1), TC6M, TC6CK and TFF6 can be programmed.

Note 4: When TC6M= 1** (upper byte in the 16-bit mode), the source clock becomes the TC5 overflow signal regardless of the TC6CK setting.

Note 5: To use the TimerCounter in the 16-bit mode, select the operating mode by programming TC6M, where TC5CR<TC5M> must be set to 011.

Note 6: To the TimerCounter in the 16-bit mode, select the source clock by programming TC5CR<TC5CK>. Set the timer start control and timer F/F control by programming TC6S and TFF6, 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	TC5 pin input	TC6 pin input
8-bit timer	O	O	O	O	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	O	O
8-bit PDO	O	O	O	O	–	–	–	–	–
8-bit PWM	O	O	O	O	O	O	O	–	–
16-bit timer	O	O	O	O	–	–	–	–	–
16-bit event counter	–	–	–	–	–	–	–	O	–
Warm-up counter	–	–	–	–	O	–	–	–	–
16-bit PWM	O	O	O	O	O	O	O	O	–
16-bit PPG	O	O	O	O	–	–	–	O	–

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 (TC5CK).

Note 2: O : 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	TC5 pin input	TC6 pin input
8-bit timer	O	–	–	–	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	O	O
8-bit PDO	O	–	–	–	–	–	–	–	–
8-bit PWM	O	–	–	–	O	–	–	–	–
16-bit timer	O	–	–	–	–	–	–	–	–
16-bit event counter	–	–	–	–	–	–	–	O	–
Warm-up counter	–	–	–	–	–	–	O	–	–
16-bit PWM	O	–	–	–	O	–	–	O	–
16-bit PPG	O	–	–	–	–	–	–	O	–

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 (TC5CK).

Note2: O : Available source clock

Table 10-3 Constraints on Register Values Being Compared

Operating mode	Register Value
8-bit timer/event counter	1≤ (TTREGn) ≤255
8-bit PDO	1≤ (TTREGn) ≤255
8-bit PWM	2≤ (PWREGn) ≤254
16-bit timer/event counter	1≤ (TTREG6, 5) ≤65535
Warm-up counter	256≤ (TTREG6, 5) ≤65535
16-bit PWM	2≤ (PWREG6, 5) ≤65534
16-bit PPG	1≤ (PWREG6, 5) < (TTREG6, 5) ≤65535 and (PWREG6, 5) + 1 < (TTREG6, 5)

Note: n = 5 to 6

10.3 Function

The TimerCounter 5 and 6 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 5 and 6 (TC5, 6) 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 (TC5 and 6)

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 $\overline{\text{PDOj}}$, $\overline{\text{PWMj}}$ and $\overline{\text{PPGj}}$ 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 = 5, 6

Table 10-4 Source Clock for TimerCounter 5, 6 (Internal Clock)

Source Clock			Resolution		Maximum Time Setting	
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	32.6 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 $fc/2^7$ Hz and generating an interrupt 80 μs later (TimerCounter6, fc = 16.0 MHz)

```
LD      (TTREG6), 0AH           : Sets the timer register (80 µs÷2^7/fc = 0AH).
DI
SET     (EIRD). 0               : Enables INTTC6 interrupt.
EI
LD      (TC6CR), 00010000B      : Sets the operating clock to fc/2^7, and 8-bit timer mode.
LD      (TC6CR), 00011000B      : Starts TC6.
```

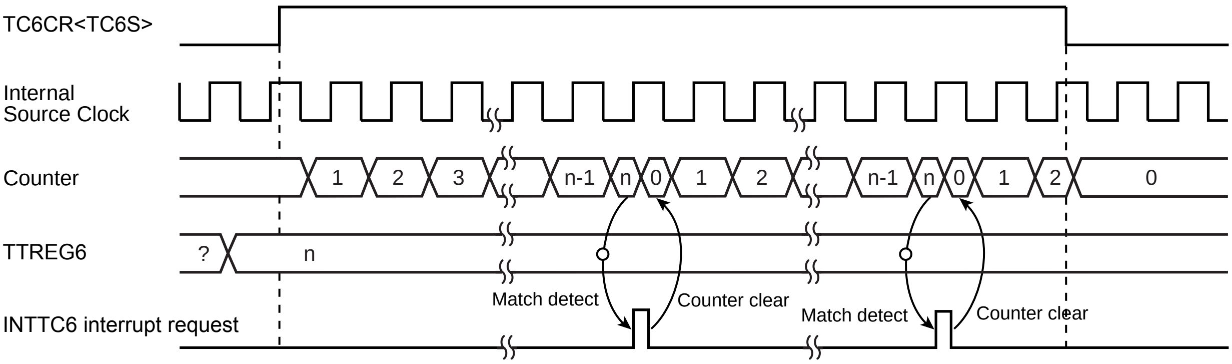


Figure 10-2 8-Bit Timer Mode Timing Chart (TC6)

10.3.2 8-Bit Event Counter Mode (TC5, 6)

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 $\overline{\text{PDOj}}$, $\overline{\text{PWMj}}$ and $\overline{\text{PPGj}}$ 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 TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 5, 6

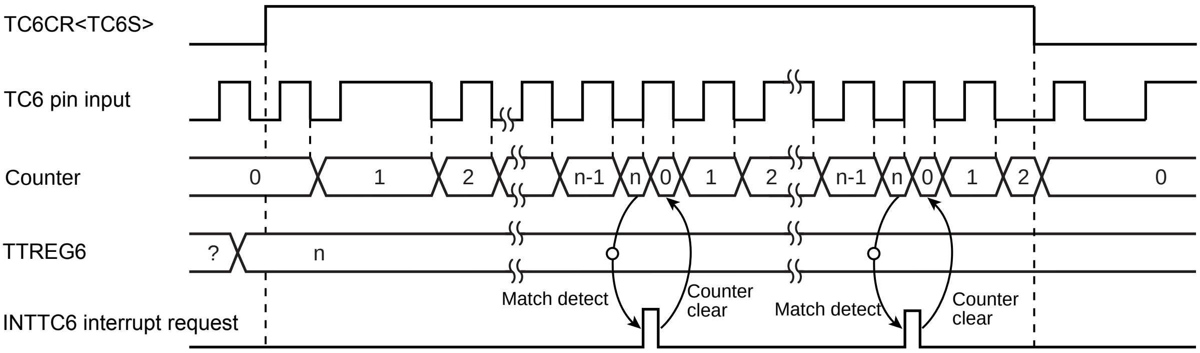


Figure 10-3 8-Bit Event Counter Mode Timing Chart (TC6)

10.3.3 8-Bit Programmable Divider Output (PDO) Mode (TC5, 6)

This mode is used to generate a pulse with a 50% duty cycle from the $\overline{\text{PDOj}}$ 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 $\overline{\text{PDOj}}$ 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 $\overline{\text{PDOj}}$ 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 TC6 (fc = 16.0 MHz)

Setting port

LD (TTREG6), 3DH : $1/1024 \div 2^7/\text{fc} \div 2$ = 3DH

LD (TC6CR), 00010001B : Sets the operating clock to fc/2^7, and 8-bit PDO mode.

LD (TC6CR), 00011001B : Starts TC6.

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 $\overline{\text{PDOj}}$ 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 $\overline{\text{PDOj}}$ pin to the high level when the TimerCounter is stopped
CLR (TCjCR).3: Stops the timer.
CLR (TCjCR).7: Sets the $\overline{\text{PDOj}}$ pin to the high level.

Note 3: j = 5, 6

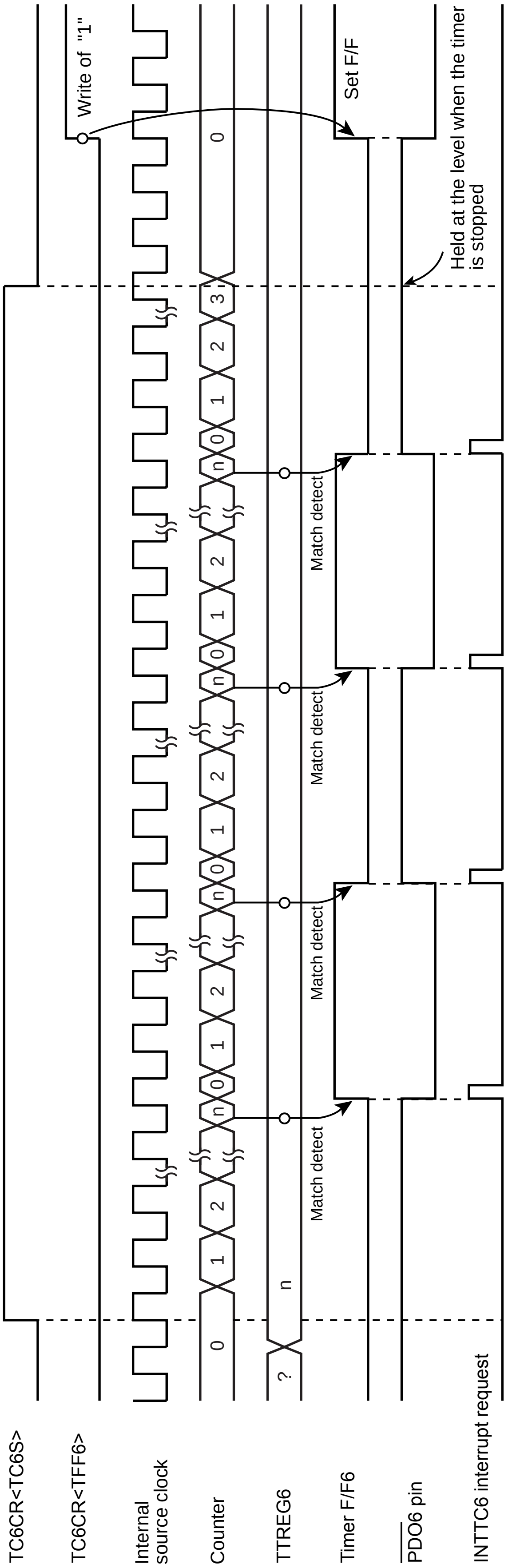


Figure 10-4 8-Bit PDO Mode Timing Chart (TC6)

10.3.4 8-Bit Pulse Width Modulation (PWM) Output Mode (TC5, 6)

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 $\overline{\text{PWMj}}$ 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 $\overline{\text{PWMj}}$ 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 $\overline{\text{PWMj}}$ pin to the high level when the TimerCounter is stopped
CLR (TCjCR).3: Stops the timer.
CLR (TCjCR).7: Sets the $\overline{\text{PWMj}}$ 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 $\overline{\text{PWMj}}$ pin during the warm-up period time after exiting the STOP mode.

Note 4: j = 5, 6

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	–

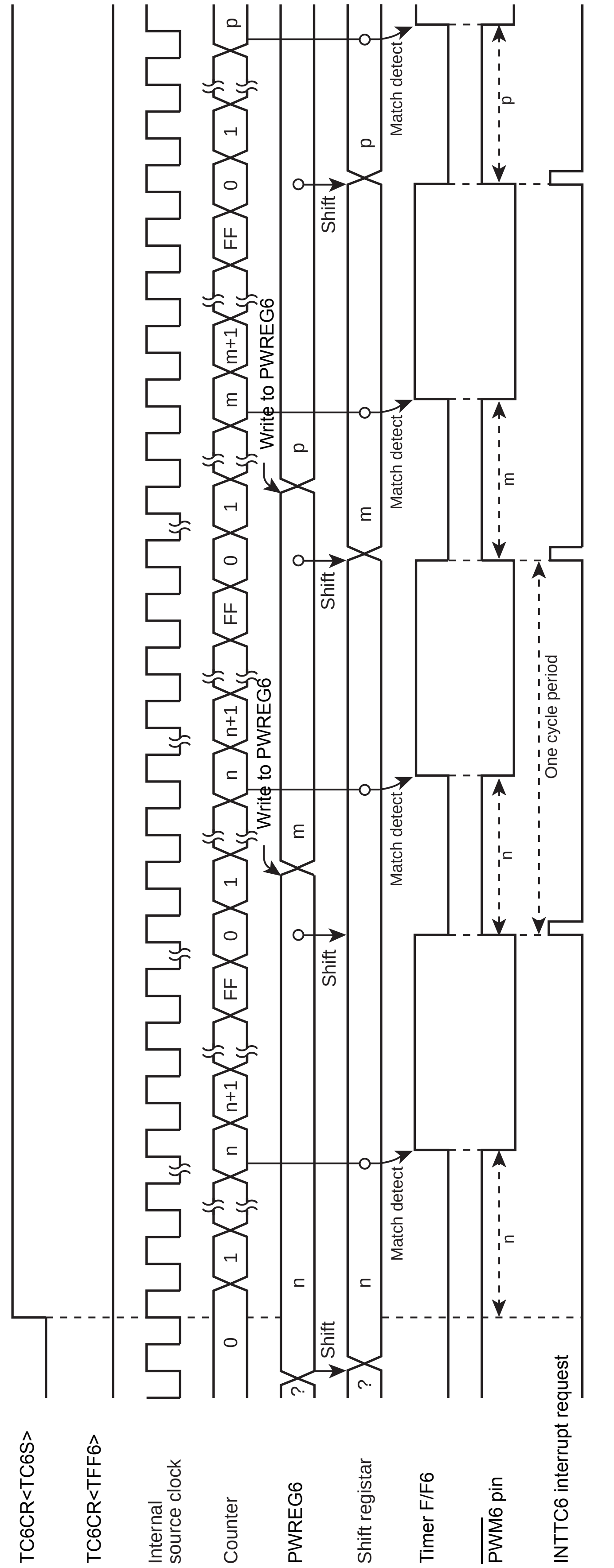


Figure 10-5 8-Bit PWM Mode Timing Chart (TC6)

10.3.5 16-Bit Timer Mode (TC5 and 6)

In the timer mode, the up-counter counts up using the internal clock. The TimerCounter 5 and 6 are cascadable to form a 16-bit timer.

When a match between the up-counter and the timer register (TTREG5, TTREG6) value is detected after the timer is started by setting TC6CR<TC6S> to 1, an INTTC6 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 $\overline{\text{PDOj}}$, $\overline{\text{PWMj}}$, and $\overline{\text{PPGj}}$ 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 = 5, 6

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

Source Clock			Resolution		Maximum Time Setting	
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					
$fc/2^{11}$	$fs/2^3$	$fs/2^3$	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 $fc/2^7$ Hz, and generating an interrupt 300 ms later (fc = 16.0 MHz)

```
LDW     (TTREG5), 927CH     : Sets the timer register (300 ms÷2^7/fc = 927CH).
DI
SET     (EIRD). 0           : Enables INTTC6 interrupt.
EI
LD      (TC5CR), 13H        :Sets the operating clock to fc/2^7, and 16-bit timer mode
                             (lower byte).
LD      (TC6CR), 04H        : Sets the 16-bit timer mode (upper byte).
LD      (TC6CR), 0CH        : Starts the timer.
```

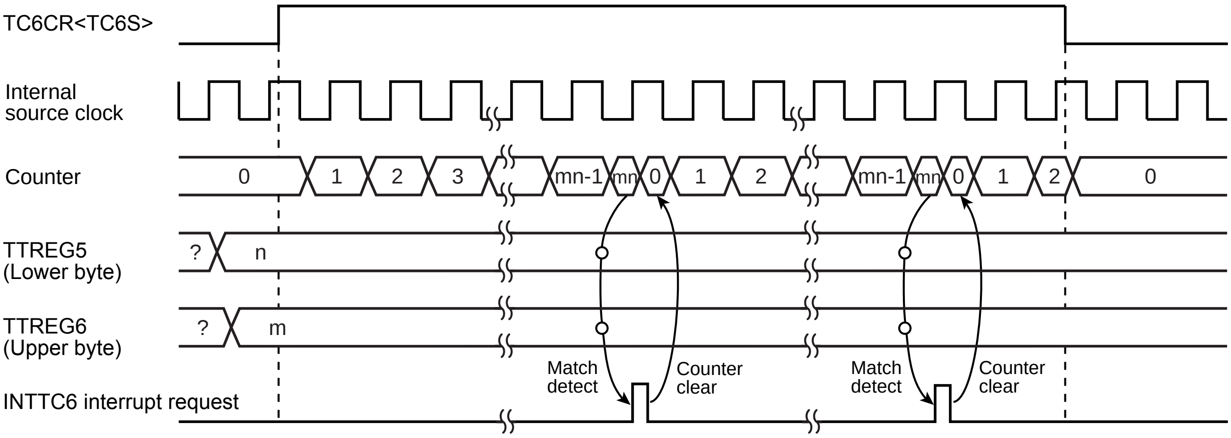


Figure 10-6 16-Bit Timer Mode Timing Chart (TC5 and TC6)

10.3.6 16-Bit Event Counter Mode (TC5 and 6)

In the event counter mode, the up-counter counts up at the falling edge to the TC5 pin. The TimerCounter 5 and 6 are cascadable to form a 16-bit event counter.

When a match between the up-counter and the timer register (TTREG5, TTREG6) value is detected after the timer is started by setting TC6CR<TC6S> to 1, an INTTC6 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 TC5 pin. Two machine cycles are required for the low- or high-level pulse input to the TC5 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 in the SLOW1/2 or SLEEP1/2 mode. Program the lower byte (TTREG5), and upper byte (TTREG6) 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 $\overline{\text{PDOj}}$, $\overline{\text{PWMj}}$ and $\overline{\text{PPGj}}$ 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 = 5, 6

10.3.7 16-Bit Pulse Width Modulation (PWM) Output Mode (TC5 and 6)

This mode is used to generate a pulse-width modulated (PWM) signals with up to 16 bits of resolution. The TimerCounter 5 and 6 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 (PWREG5, PWREG6) value is detected, the logic level output from the timer F/F6 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F6 is switched to the opposite state again by the counter overflow, and the counter is cleared. The INTTC6 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC5 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 to in the SLOW1/2 or SLEEP1/2 mode.

Since the initial value can be set to the timer F/F6 by TC6CR<TFF6>, positive and negative pulses can be generated. Upon reset, the timer F/F6 is cleared to 0.

(The logic level output from the $\overline{\text{PWM}}$6 pin is the opposite to the timer F/F6 logic level.)

Since PWREG6 and 5 in the PWM mode are serially connected to the shift register, the values set to PWREG6 and 5 can be changed while the timer is running. The values set to PWREG6 and 5 during a run of the timer are shifted by the INTTCj interrupt request and loaded into PWREG6 and 5. While the timer is stopped, the values are shifted immediately after the programming of PWREG6 and 5. Set the lower byte (PWREG5) and upper byte (PWREG6) in this order to program PWREG6 and 5. (Programming only the lower or upper byte of the register should not be attempted.)

If executing the read instruction to PWREG6 and 5 during PWM output, the values set in the shift register is read, but not the values set in PWREG6 and 5. Therefore, after writing to the PWREG6 and 5, reading data of PWREG6 and 5 is previous value until INTTC6 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 PWREG6 and 5 immediately after the INTTC6 interrupt request is generated (normally in the INTTC6 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 INTTC6 interrupt request is generated.

Note 2: When the timer is stopped during PWM output, the $\overline{\text{PWM}}$6 pin holds the output status when the timer is stopped. To change the output status, program TC6CR<TFF6> after the timer is stopped. Do not program TC6CR<TFF6> upon stopping of the timer.
Example: Fixing the $\overline{\text{PWM}}$6 pin to the high level when the TimerCounter is stopped

CLR (TC6CR).3: Stops the timer.
CLR (TC6CR).7 : Sets the $\overline{\text{PWM}}$6 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 fc, fc/2 or fs is selected as the source clock, a pulse is output from the $\overline{\text{PWM}}$6 pin during the warm-up period time after exiting the STOP mode.

Table 10-7 16-Bit 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					
$fc/2^{11}$	$fs/2^3$ [Hz]	$fs/2^3$ [Hz]	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	–
fs	fs	fs	30.5 μs	30.5 μs	2 s	2 s
fc/2	fc/2	–	125 ns	–	8.2 ms	–
fc	fc	–	62.5 ns	–	4.1 ms	–

Example :Generating a pulse with 1-ms high-level width and a period of 32.768 ms (fc = 16.0 MHz)

```
              Setting ports
LDW           (PWREG5), 07D0H        : Sets the pulse width.
LD            (TC5CR), 33H           : Sets the operating clock to fc/2^3, and 16-bit PWM output
                                       mode (lower byte).
LD            (TC6CR), 056H          : Sets TFF6 to the initial value 0, and 16-bit PWM signal
                                       generation mode (upper byte).
LD            (TC6CR), 05EH          : Starts the timer.
```

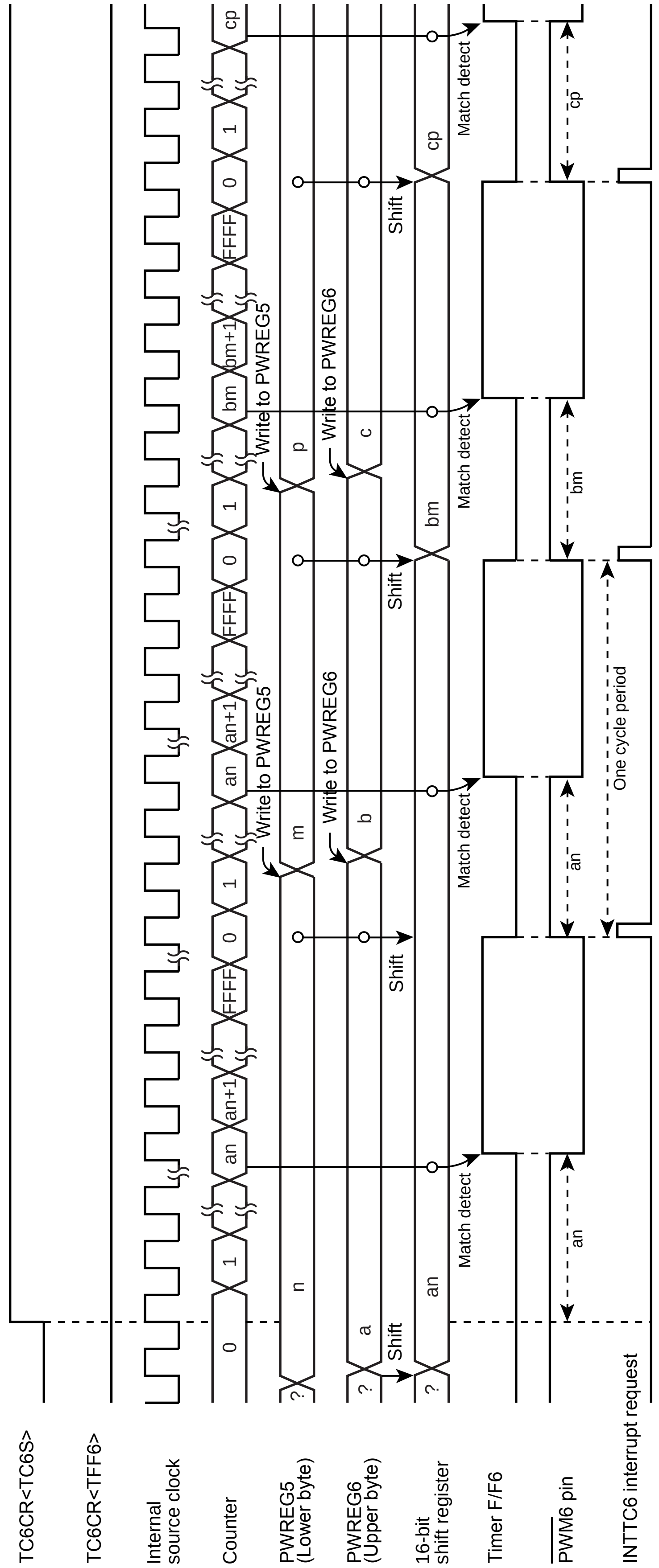


Figure 10-7 16-Bit PWM Mode Timing Chart (TC5 and TC6)

10.3.8 16-Bit Programmable Pulse Generate (PPG) Output Mode (TC5 and 6)

This mode is used to generate pulses with up to 16-bits of resolution. The timer counter 5 and 6 are cascadable 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 (PWREG5, PWREG6) value is detected, the logic level output from the timer F/F6 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F6 is switched to the opposite state again when a match between the up-counter and the timer register (TTREG5, TTREG6) value is detected, and the counter is cleared. The INTTC6 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC5 pin. Therefore, a maximum frequency to be supplied is $fc/2^4$ Hz in the NORMAL1/2 or IDLE1/2 mode, and $fs/2^4$ to in the SLOW1/2 or SLEEP1/2 mode.

Since the initial value can be set to the timer F/F6 by TC6CR<TFF6>, positive and negative pulses can be generated. Upon reset, the timer F/F6 is cleared to 0.

(The logic level output from the $\overline{\text{PPG}}$6 pin is the opposite to the timer F/F6.)

Set the lower byte and upper byte in this order to program the timer register. (TTREG5 → TTREG6, PWREG5 → PWREG6) (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 (fc = 16.0 MHz)

```
            Setting ports
LDW         (PWREG5), 07D0H        : Sets the pulse width.
LDW         (TTREG5), 8002H        : Sets the cycle period.
LD          (TC5CR), 33H           : Sets the operating clock to fc/2^3, and16-bit PPG mode
                                     (lower byte).
LD          (TC6CR), 057H          : Sets TFF6 to the initial value 0, and 16-bit
                                     PPG mode (upper byte).
LD          (TC6CR), 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{\text{PPG}}$6 pin holds the output status when the timer is stopped. To change the output status, program TC6CR<TFF6> after the timer is stopped. Do not change TC6CR<TFF6> upon stopping of the timer.
Example: Fixing the $\overline{\text{PPG}}$6 pin to the high level when the TimerCounter is stopped

```
    CLR  (TC6CR).3: Stops the timer
    CLR  (TC6CR).7: Sets the PPG6 pin to the high level
```

Note 3: i = 5, 6

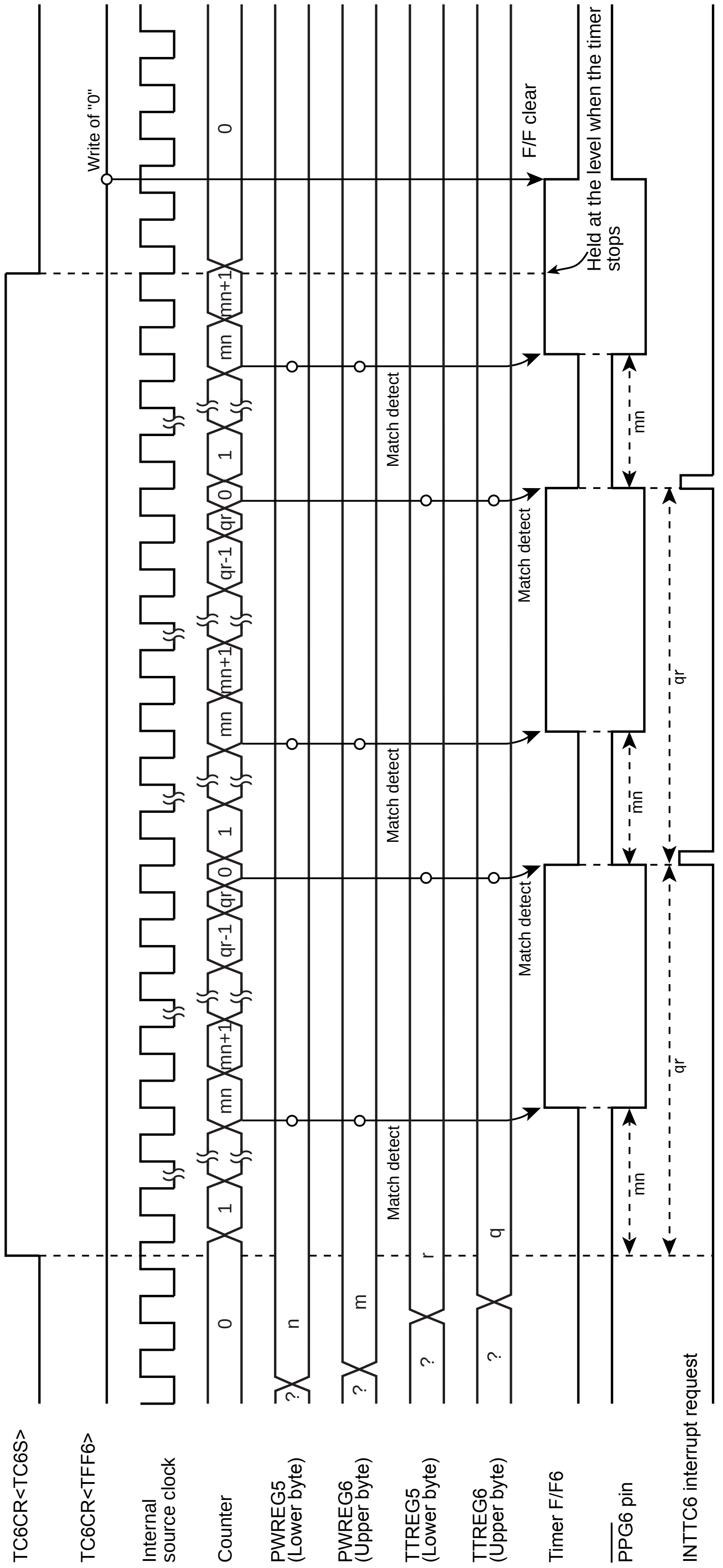


Figure 10-8 16-Bit PPG Mode Timing Chart (TC5 and TC6)

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 5 and 6 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 $\overline{\text{PDOi}}$, $\overline{\text{PWMi}}$ and $\overline{\text{PPGi}}$ pins may output pulses.

Note 2: In the warm-up counter mode, only upper 8 bits of the timer register TTREG6 and 5 are used for match detection and lower 8 bits are not used.

Note 3: i = 5, 6

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 fs 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 (TTREG6, 5) value is detected after the timer is started by setting TC6CR<TC6S> to 1, the counter is cleared by generating the INTTC6 interrupt request. After stopping the timer in the INTTC6 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 (fs = 32.768 kHz)

Minimum Time Setting (TTREG6, 5 = 0100H)	Maximum Time Setting (TTREG6, 5 = FF00H)
7.81 ms	1.99 s

Example :After checking low-frequency clock oscillation stability with TC6 and 5, switching to the SLOW1 mode

```
           SET     (SYSCR2).6        : SYSCR2<XTEN> ← 1
           LD      (TC5CR), 43H      : Sets TFF5=0, source clock fs, and 16-bit mode.
           LD      (TC6CR), 05H      : Sets TFF6=0, and warm-up counter mode.
           LD      (TTREG5), 8000H   : Sets the warm-up time.
                                       (The warm-up time depends on the oscillator characteristic.)
           DI                        : IMF ← 0
           SET     (EIRD). 0         : Enables the INTTC6.
           EI                        : IMF ← 1
           SET     (TC6CR).3         : Starts TC6 and 5.
             :           :
PINTTC6:   CLR     (TC6CR).3         : Stops TC6 and 5.
           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
             :           :
VINTTC6:   DW      PINTTC6           : INTTC6 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 fc 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 (TTREG6, 5) value is detected after the timer is started by setting TC6CR<TC6S> to 1, the counter is cleared by generating the INTTC6 interrupt request. After stopping the timer in the INTTC6 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 (TTREG6, 5 = 0100H)	Maximum time Setting (TTREG6, 5 = FF00H)
16 μs	4.08 ms

Example :After checking high-frequency clock oscillation stability with TC6 and 5, switching to the NORMAL1 mode

```
            SET     (SYSCR2).7          : SYSCR2<XEN> ← 1
            LD      (TC5CR), 63H        : Sets TFF5=0, source clock fc, and 16-bit mode.
            LD      (TC6CR), 05H        : Sets TFF6=0, and warm-up counter mode.
            LD      (TTREG5), 0F800H    : Sets the warm-up time.
                                          (The warm-up time depends on the oscillator characteristic.)
            DI                          : IMF ← 0
            SET     (EIRD). 0           : Enables the INTTC6.
            EI                          : IMF ← 1
            SET     (TC6CR).3           : Starts the TC6 and 5.
             :        :
PINTTC6:    CLR     (TC6CR).3           : Stops the TC6 and 5.
            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
             :        :
VINTTC6:    DW      PINTTC6             : INTTC6 vector table
```

11. Synchronous Serial Interface (SIO)

The TMP86CS28FG has a clocked-synchronous 8-bit serial interface. Serial interface has an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data.

Serial interface is connected to outside peripherl devices via SO, SI, SCK port.

11.1 Configuration

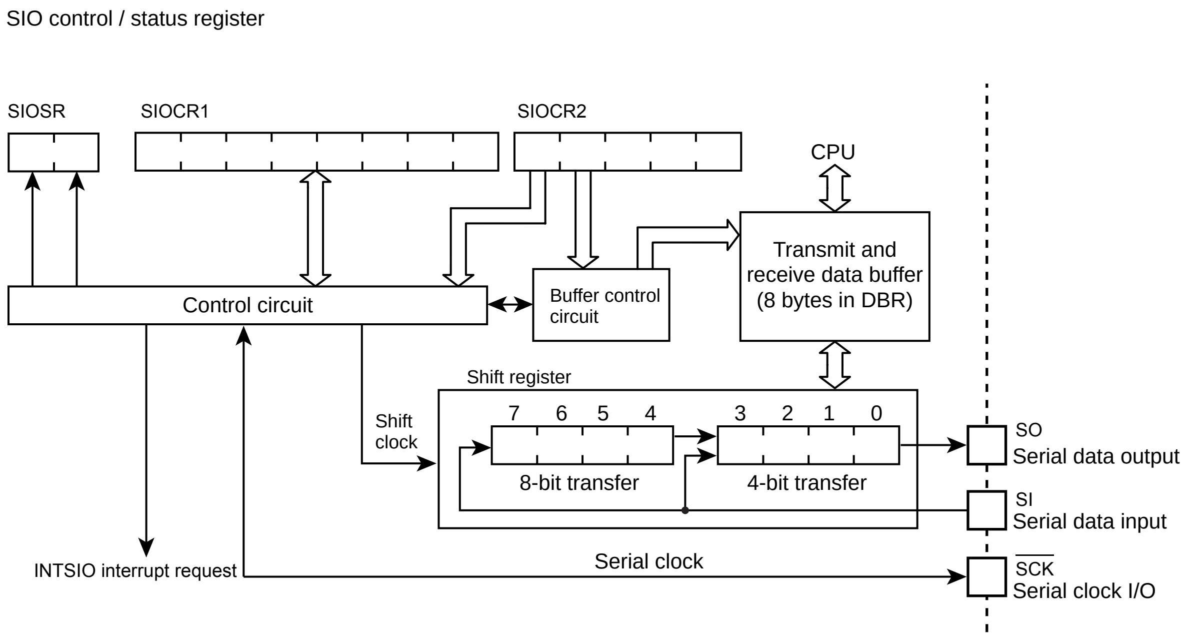


Figure 11-1 Serial Interface

11.2 Control

The serial interface is controlled by SIO control registers (SIOCR1/SIOCR2). The serial interface status can be determined by reading SIO status register (SIOSR).

The transmit and receive data buffer is controlled by the SIOCR2<BUF>. The data buffer is assigned to address 0F60H to 0F67H for SIO in the DBR area, and can continuously transfer up to 8 words (bytes or nibbles) at one time. When the specified number of words has been transferred, a buffer empty (in the transmit mode) or a buffer full (in the receive mode or transmit/receive mode) interrupt (INTSIO) is generated.

When the internal clock is used as the serial clock in the 8-bit receive mode and the 8-bit transmit/receive mode, a fixed interval wait can be applied to the serial clock for each word transferred. Four different wait times can be selected with SIOCR2<WAIT>.

SIO Control Register 1

<table>
<tr><td>SIOCR1</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td>(0F68H)</td><td>SIOS</td><td>SIOINH</td><td colspan="3">SIOM</td><td colspan="3">SCK</td><td>(Initial value: 0000 0000)</td></tr>
</table>

<table>
<tr><td>SIOS</td><td>Indicate transfer start / stop</td><td colspan="4">0: Stop
1: Start</td><td rowspan="3">Write only</td></tr>
<tr><td>SIOINH</td><td>Continue / abort transfer</td><td colspan="4">0: Continuously transfer
1: Abort transfer (Automatically cleared after abort)</td></tr>
<tr><td>SIOM</td><td>Transfer mode select</td><td colspan="4">000: 8-bit transmit mode
010: 4-bit transmit mode
100: 8-bit transmit / receive mode
101: 8-bit receive mode
110: 4-bit receive mode
Except the above: Reserved</td></tr>
<tr><td rowspan="10">SCK</td><td rowspan="10">Serial clock select</td><td rowspan="2"></td><td colspan="2">NORMAL1/2, IDLE1/2 mode</td><td rowspan="2">SLOW1/2
SLEEP1/2
mode</td><td rowspan="10">Write only</td></tr>
<tr><td>DV7CK = 0</td><td>DV7CK = 1</td></tr>
<tr><td>000</td><td>fc/2¹³</td><td>fs/2⁵</td><td>fs/2⁵</td></tr>
<tr><td>001</td><td>fc/2⁸</td><td>fc/2⁸</td><td>-</td></tr>
<tr><td>010</td><td>fc/2⁷</td><td>fc/2⁷</td><td>-</td></tr>
<tr><td>011</td><td>fc/2⁶</td><td>fc/2⁶</td><td>-</td></tr>
<tr><td>100</td><td>fc/2⁵</td><td>fc/2⁵</td><td>-</td></tr>
<tr><td>101</td><td>fc/2⁴</td><td>fc/2⁴</td><td>-</td></tr>
<tr><td>110</td><td colspan="3">Reserved</td></tr>
<tr><td>111</td><td colspan="3">External clock (Input from SCK pin)</td></tr>
</table>

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

Note 2: Set SIOS to "0" and SIOINH to "1" when setting the transfer mode or serial clock.

Note 3: SIOCR1 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO Control Register 2

<table>
<tr><td>SIOCR2</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td>(0F69H)</td><td></td><td></td><td></td><td colspan="2">WAIT</td><td colspan="3">BUF</td><td>(Initial value: ***0 0000)</td></tr>
</table>

WAIT	Wait control	Always sets "00" except 8-bit transmit / receive mode. 00: $T_f = T_D$(Non wait) 01: $T_f = 2T_D$(Wait) 10: $T_f = 4T_D$(Wait) 11: $T_f = 8T_D$ (Wait)	Write only
BUF	Number of transfer words (Buffer address in use)	000: 1 word transfer 0F60H 001: 2 words transfer 0F60H ~ 0F61H 010: 3 words transfer 0F60H ~ 0F62H 011: 4 words transfer 0F60H ~ 0F63H 100: 5 words transfer 0F60H ~ 0F64H 101: 6 words transfer 0F60H ~ 0F65H 110: 7 words transfer 0F60H ~ 0F66H 111: 8 words transfer 0F60H ~ 0F67H	

Note 1: The lower 4 bits of each buffer are used during 4-bit transfers. Zeros (0) are stored to the upper 4bits when receiving.

Note 2: Transmitting starts at the lowest address. Received data are also stored starting from the lowest address to the highest address. (The first buffer address transmitted is 0F60H).

Note 3: The value to be loaded to BUF is held after transfer is completed.

Note 4: SIOCR2 must be set when the serial interface is stopped (SIOF = 0).

Note 5: *: Don't care

Note 6: SIOCR2 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO Status Register

SIOSR	7	6	5	4	3	2	1	0
(0F69H)	SIOF	SEF						

SIOF	Serial transfer operating status monitor	0: Transfer terminated 1: Transfer in process	Read only
SEF	Shift operating status monitor	0: Shift operation terminated 1: Shift operation in process	

Note 1: T_f; Frame time, T_D; Data transfer time

Note 2: After SIOS is cleared to "0", SIOF is cleared to "0" at the termination of transfer or the setting of SIOINH to "1".

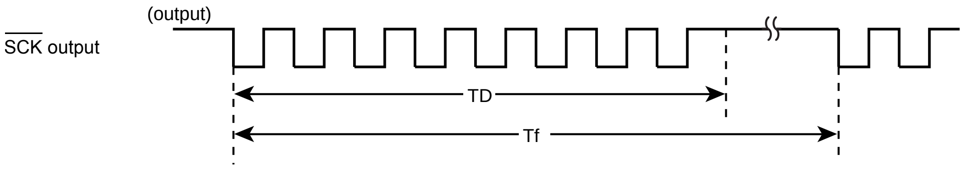


Figure 11-2 Frame time (T_f) and Data transfer time (T_D)

11.3 Serial clock

11.3.1 Clock source

Internal clock or external clock for the source clock is selected by SIOCR1<SCK>.

11.3.1.1 Internal clock

Any of six frequencies can be selected. The serial clock is output to the outside on the SCK pin. The SCK pin goes high when transfer starts.

When data writing (in the transmit mode) or reading (in the receive mode or the transmit/receive mode) cannot keep up with the serial clock rate, there is a wait function that automatically stops the serial clock and holds the next shift operation until the read/write processing is completed.

Table 11-1 Serial Clock Rate

	NORMAL1/2, IDLE1/2 mode				SLOW1/2, SLEEP1/2 mode	
	DV7CK = 0		DV7CK = 1			
SCK	Clock	Baud Rate	Clock	Baud Rate	Clock	Baud Rate
000	$fc/2^{13}$	1.91 Kbps	$fs/2^5$	1024 bps	$fs/2^5$	1024 bps
001	$fc/2^8$	61.04 Kbps	$fc/2^8$	61.04 Kbps	-	-
010	$fc/2^7$	122.07 Kbps	$fc/2^7$	122.07 Kbps	-	-
011	$fc/2^6$	244.14 Kbps	$fc/2^6$	244.14 Kbps	-	-
100	$fc/2^5$	488.28 Kbps	$fc/2^5$	488.28 Kbps	-	-
101	$fc/2^4$	976.56 Kbps	$fc/2^4$	976.56 Kbps	-	-
110	-	-	-	-	-	-
111	External	External	External	External	External	External

Note: 1 Kbit = 1024 bit (fc = 16 MHz, fs = 32.768 kHz)

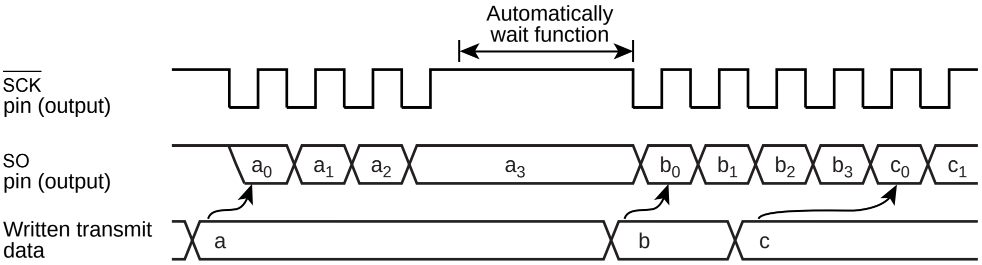
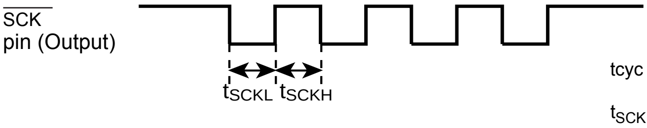


Figure 11-3 Automatic Wait Function (at 4-bit transmit mode)

11.3.1.2 External clock

An external clock connected to the $\overline{SCK}$ pin is used as the serial clock. In this case, output latch of this port should be set to "1". To ensure shifting, a pulse width of at least 4 machine cycles is required. This pulse is needed for the shift operation to execute certainly. Actually, there is necessary processing time for interrupting, writing, and reading. The minimum pulse is determined by setting the mode and the program. Therfore, maximum transfer frequency will be 488.3K bit/sec (at fc=16MHz).



tcyc = 4/fc (In the NORMAL1/2, IDLE1/2 modes)
4/fs (In the SLOW1/2, SLEEP1/2 modes)
t_{SCKL}, t_{SCKH} > 4tcyc

Figure 11-4 External clock pulse width

11.3.2 Shift edge

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

11.3.2.1 Leading edge

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

11.3.2.2 Trailing edge

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

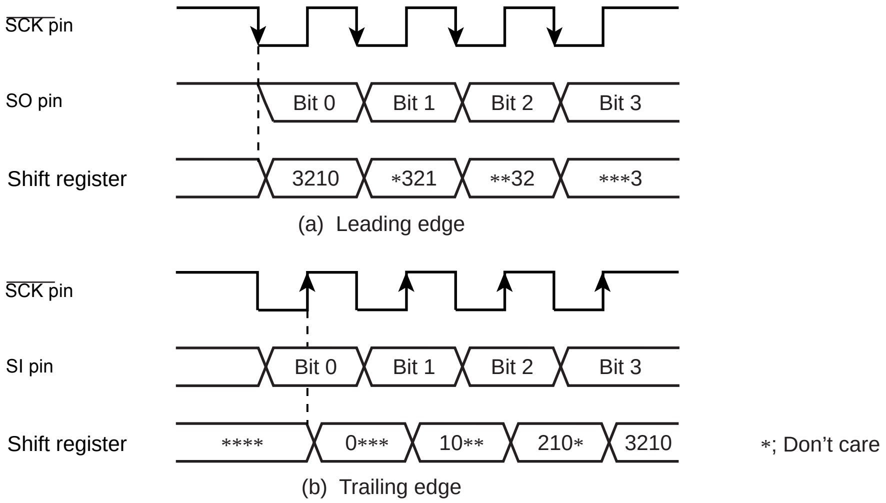


Figure 11-5 Shift edge

11.4 Number of bits to transfer

Either 4-bit or 8-bit serial transfer can be selected. When 4-bit serial transfer is selected, only the lower 4 bits of the transmit/receive data buffer register are used. The upper 4 bits are cleared to “0” when receiving.
The data is transferred in sequence starting at the least significant bit (LSB).

11.5 Number of words to transfer

Up to 8 words consisting of 4 bits of data (4-bit serial transfer) or 8 bits (8-bit serial transfer) of data can be transferred continuously. The number of words to be transferred can be selected by SIOCR2<BUF>.

An INTSIO interrupt is generated when the specified number of words has been transferred. If the number of words is to be changed during transfer, the serial interface must be stopped before making the change. The number of words can be changed during automatic-wait operation of an internal clock. In this case, the serial interface is not required to be stopped.

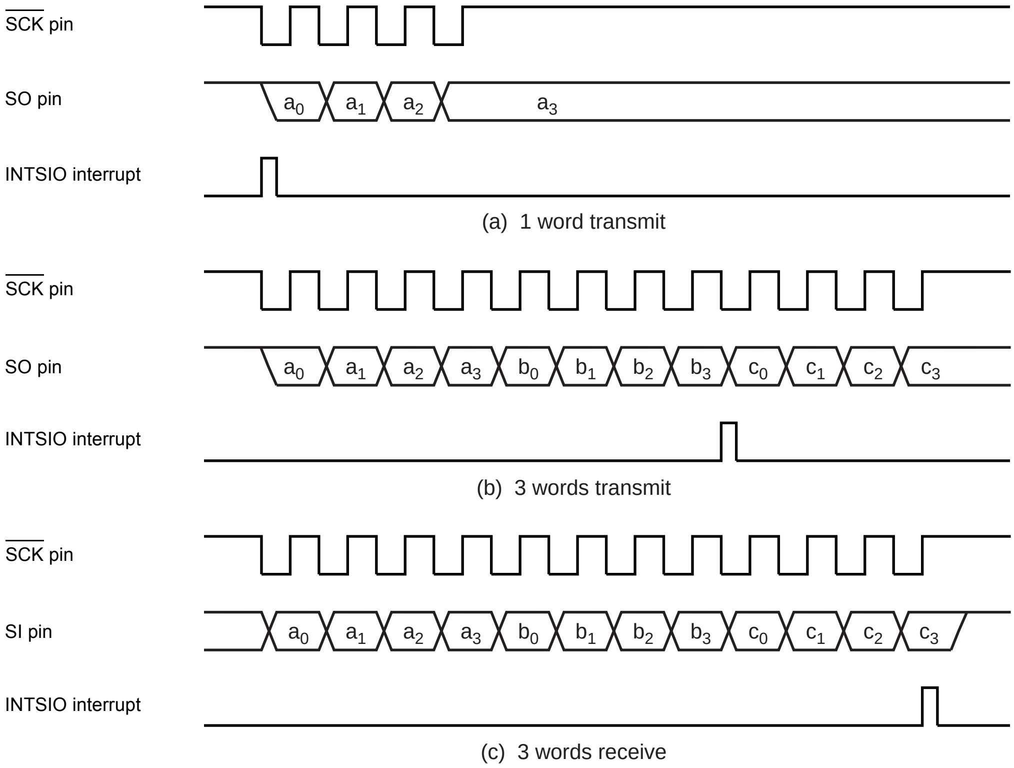


Figure 11-6 Number of words to transfer (Example: 1word = 4bit)

11.6 Transfer Mode

SIOCR1<SIOM> is used to select the transmit, receive, or transmit/receive mode.

11.6.1 4-bit and 8-bit transfer modes

In these modes, firstly set the SIO control register to the transmit mode, and then write first transmit data (number of transfer words to be transferred) to the data buffer registers (DBR).

After the data are written, the transmission is started by setting SIOCR1<SIOS> to “1”. The data are then output sequentially to the SO pin in synchronous with the serial clock, starting with the least significant bit (LSB). As soon as the LSB has been output, the data are transferred from the data buffer register to the shift register. When the final data bit has been transferred and the data buffer register is empty, an INTSIO (Buffer empty) interrupt is generated to request the next transmitted data.

When the internal clock is used, the serial clock will stop and an automatic-wait will be initiated if the next transmitted data are not loaded to the data buffer register by the time the number of data words specified with the SIOCR2<BUF> has been transmitted. Writing even one word of data cancels the automatic-wait; therefore, when transmitting two or more words, always write the next word before transmission of the previous word is completed.

Note: Automatic waits are also canceled by writing to a DBR not being used as a transmit data buffer register; therefore, during SIO do not use such DBR for other applications. For example, when 3 words are transmitted, do not use the DBR of the remained 5 words.

When an external clock is used, the data must be written to the data buffer register before shifting next data. Thus, the transfer speed is determined by the maximum delay time from the generation of the interrupt request to writing of the data to the data buffer register by the interrupt service program.

The transmission is ended by clearing SIOCR1<SIOS> to “0” or setting SIOCR1<SIOINH> to “1” in buffer empty interrupt service program.

SIOCR1<SIOS> is cleared, the operation will end after all bits of words are transmitted.

That the transmission has ended can be determined from the status of SIOSR<SIOF> because SIOSR<SIOF> is cleared to “0” when a transfer is completed.

When SIOCR1<SIOINH> is set, the transmission is immediately ended and SIOSR<SIOF> is cleared to “0”.

When an external clock is used, it is also necessary to clear SIOCR1<SIOS> to “0” before shifting the next data; If SIOCR1<SIOS> is not cleared before shift out, dummy data will be transmitted and the operation will end.

If it is necessary to change the number of words, SIOCR1<SIOS> should be cleared to “0”, then SIOCR2<BUF> must be rewritten after confirming that SIOSR<SIOF> has been cleared to “0”.

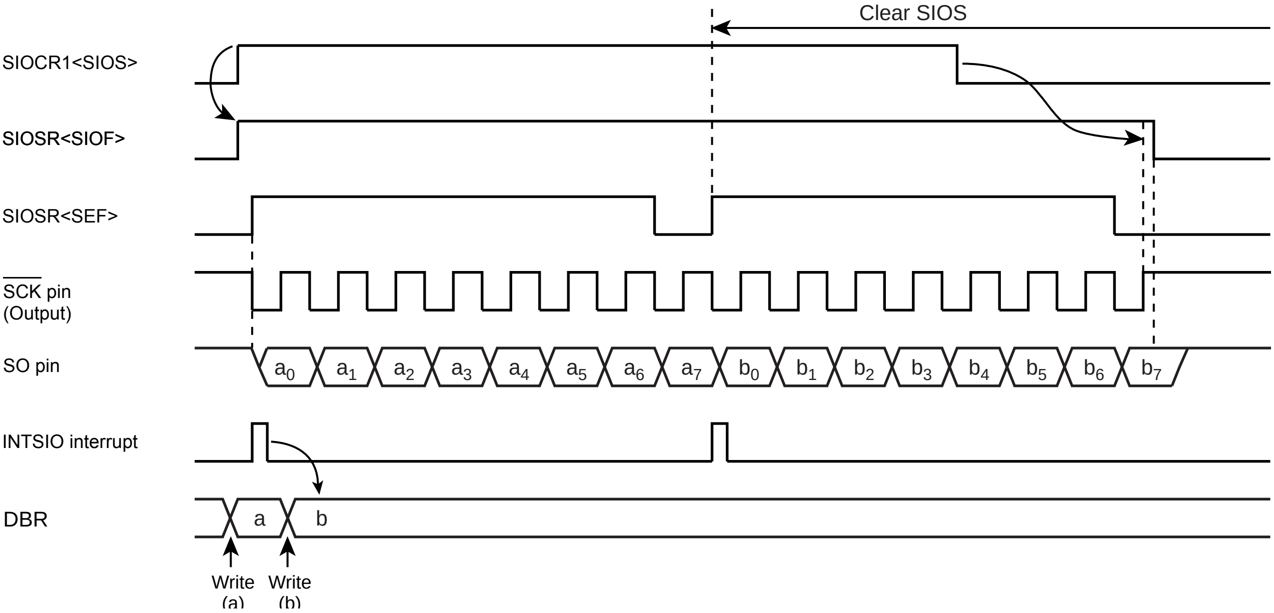


Figure 11-7 Transfer Mode (Example: 8bit, 1word transfer, Internal clock)

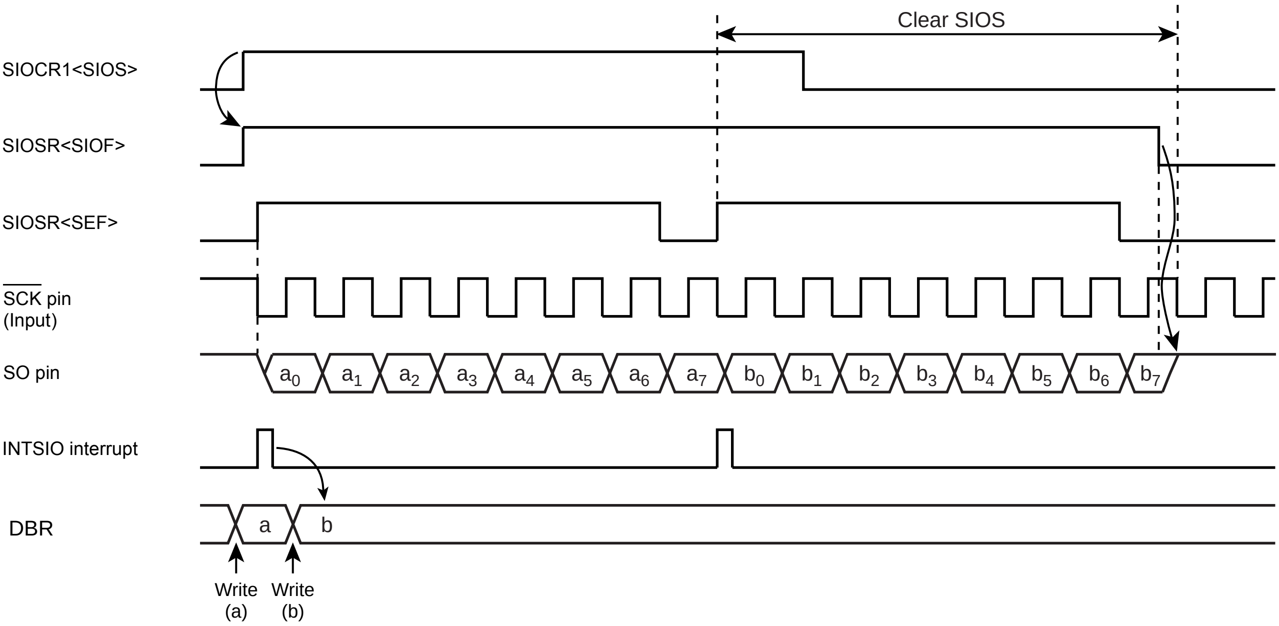


Figure 11-8 Transfer Mode (Example: 8bit, 1word transfer, External clock)

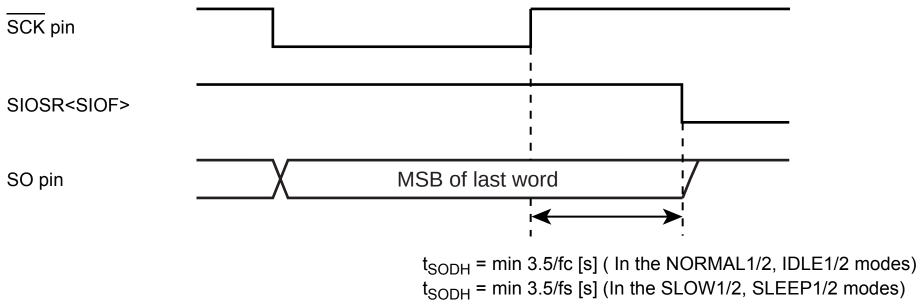

SCK pin
SIOSR<SIOF>
SO pin
MSB of last word
t_{SODH} = min 3.5/fc [s] (In the NORMAL1/2, IDLE1/2 modes)
t_{SODH} = min 3.5/fs [s] (In the SLOW1/2, SLEEP1/2 modes)

Figure 11-9 Transmiiied Data Hold Time at End of Transfer

11.6.2 4-bit and 8-bit receive modes

After setting the control registers to the receive mode, set SIOCR1<SIOS> to “1” to enable receiving. The data are then transferred to the shift register via the SI pin in synchronous with the serial clock. When one word of data has been received, it is transferred from the shift register to the data buffer register (DBR). When the number of words specified with the SIOCR2<BUF> has been received, an INTSIO (Buffer full) interrupt is generated to request that these data be read out. The data are then read from the data buffer registers by the interrupt service program.

When the internal clock is used, and the previous data are not read from the data buffer register before the next data are received, the serial clock will stop and an automatic-wait will be initiated until the data are read. A wait will not be initiated if even one data word has been read.

Note: Waits are also canceled by reading a DBR not being used as a received data buffer register is read; therefore, during SIO do not use such DBR for other applications.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, the previous data are read before the next data are transferred to the data buffer register. If the previous data have not been read, the next data will not be transferred to the data buffer register and the receiving of any more data will be canceled. When an external clock is used, the maximum transfer speed is determined by the delay between the time when the interrupt request is generated and when the data received have been read.

The receiving is ended by clearing SIOCR1<SIOS> to “0” or setting SIOCR1<SIOINH> to “1” in buffer full interrupt service program.
When SIOCR1<SIOS> is cleared, the current data are transferred to the buffer. After SIOCR1<SIOS> cleared, the receiving is ended at the time that the final bit of the data has been received. That the receiving has ended can be determined from the status of SIOSR<SIOF>. SIOSR<SIOF> is cleared to “0” when the receiving is ended. After confirmed the receiving termination, the final receiving data is read. When SIOCR1<SIOINH> is set, the receiving is immediately ended and SIOSR<SIOF> is cleared to “0”. (The received data is ignored, and it is not required to be read out.)

If it is necessary to change the number of words in external clock operation, SIOCR1<SIOS> should be cleared to “0” then SIOCR2<BUF> must be rewritten after confirming that SIOSR<SIOF> has been cleared to “0”. If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of data receiving, SIOCR2<BUF> must be rewritten before the received data is read out.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIOCR1<SIOS> to “0”, read the last data and then switch the transfer mode.

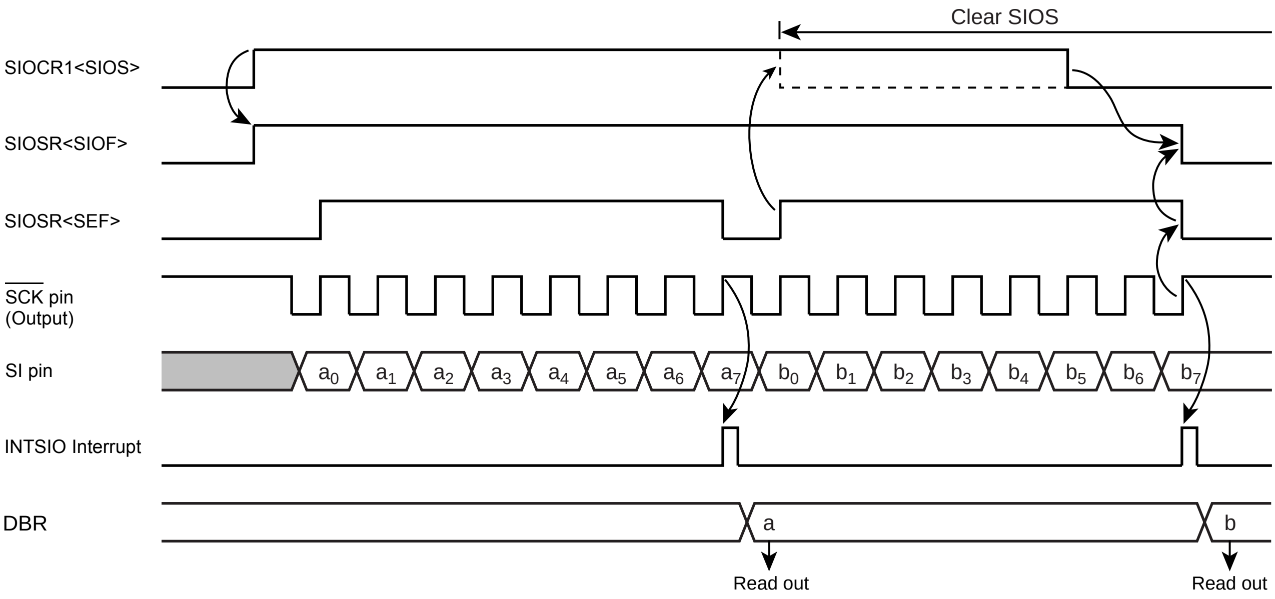


Figure 11-10 Receive Mode (Example: 8bit, 1word transfer, Internal clock)

11.6.3 8-bit transfer / receive mode

After setting the SIO control register to the 8-bit transmit/receive mode, write the data to be transmitted first to the data buffer registers (DBR). After that, enable the transmit/receive by setting SIOCR1<SIOS> to “1”. When transmitting, the data are output from the SO pin at leading edges of the serial clock. When receiving, the data are input to the SI pin at the trailing edges of the serial clock. When the all receive is enabled, 8-bit data are transferred from the shift register to the data buffer register. An INTSIO interrupt is generated when the number of data words specified with the SIOCR2<BUF> has been transferred. Usually, read the receive data from the buffer register in the interrupt service. The data buffer register is used for both transmitting and receiving; therefore, always write the data to be transmitted after reading the all received data.

When the internal clock is used, a wait is initiated until the received data are read and the next transfer data are written. A wait will not be initiated if even one transfer data word has been written.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, it is necessary to read the received data and write the data to be transmitted next before starting the next shift operation. When an external clock is used, the transfer speed is determined by the maximum delay between generation of an interrupt request and the received data are read and the data to be transmitted next are written.

The transmit/receive operation is ended by clearing SIOCR1<SIOS> to “0” or setting SIOCR1<SIOINH> to “1” in INTSIO interrupt service program.

When SIOCR1<SIOS> is cleared, the current data are transferred to the buffer. After SIOCR1<SIOS> cleared, the transmitting/receiving is ended at the time that the final bit of the data has been transmitted. That the transmitting/receiving has ended can be determined from the status of SIOSR<SIOF>. SIOSR<SIOF> is cleared to “0” when the transmitting/receiving is ended.

When SIOCR1<SIOINH> is set, the transmit/receive operation is immediately ended and SIOSR<SIOF> is cleared to “0”.

If it is necessary to change the number of words in external clock operation, SIOCR1<SIOS> should be cleared to “0”, then SIOCR2<BUF> must be rewritten after confirming that SIOSR<SIOF> has been cleared to “0”.

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of transmit/receive operation, SIOCR2<BUF> must be rewritten before reading and writing of the receive/transmit data.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIOCR1<SIOS> to “0”, read the last data and then switch the transfer mode.

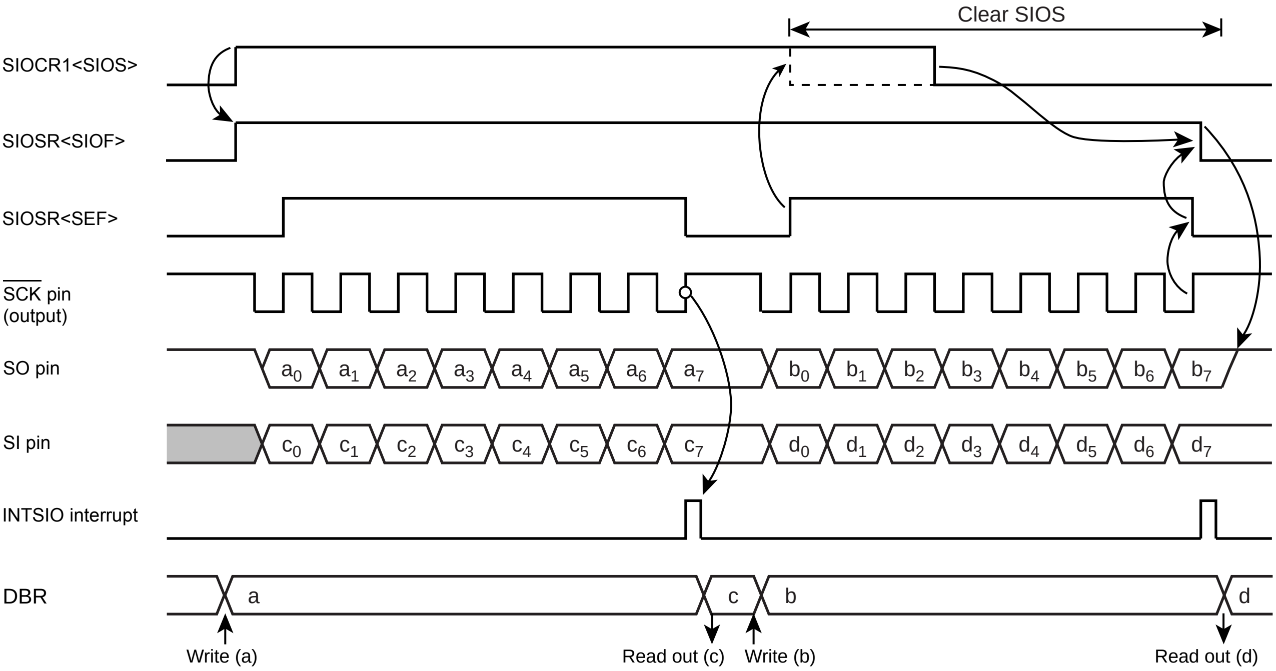
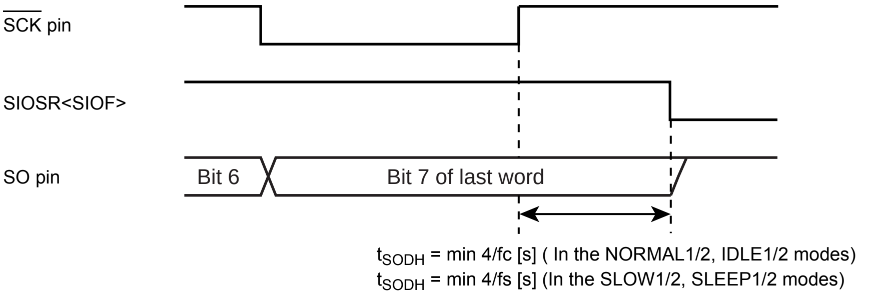


Figure 11-11 Transfer / Receive Mode (Example: 8bit, 1word transfer, Internal clock)



t_{SODH} = min 4/fc [s] (In the NORMAL1/2, IDLE1/2 modes)
t_{SODH} = min 4/fs [s] (In the SLOW1/2, SLEEP1/2 modes)

Figure 11-12 Transmitted Data Hold Time at End of Transfer / Receive

12. Asynchronous Serial interface (UART1)

12.1 Configuration

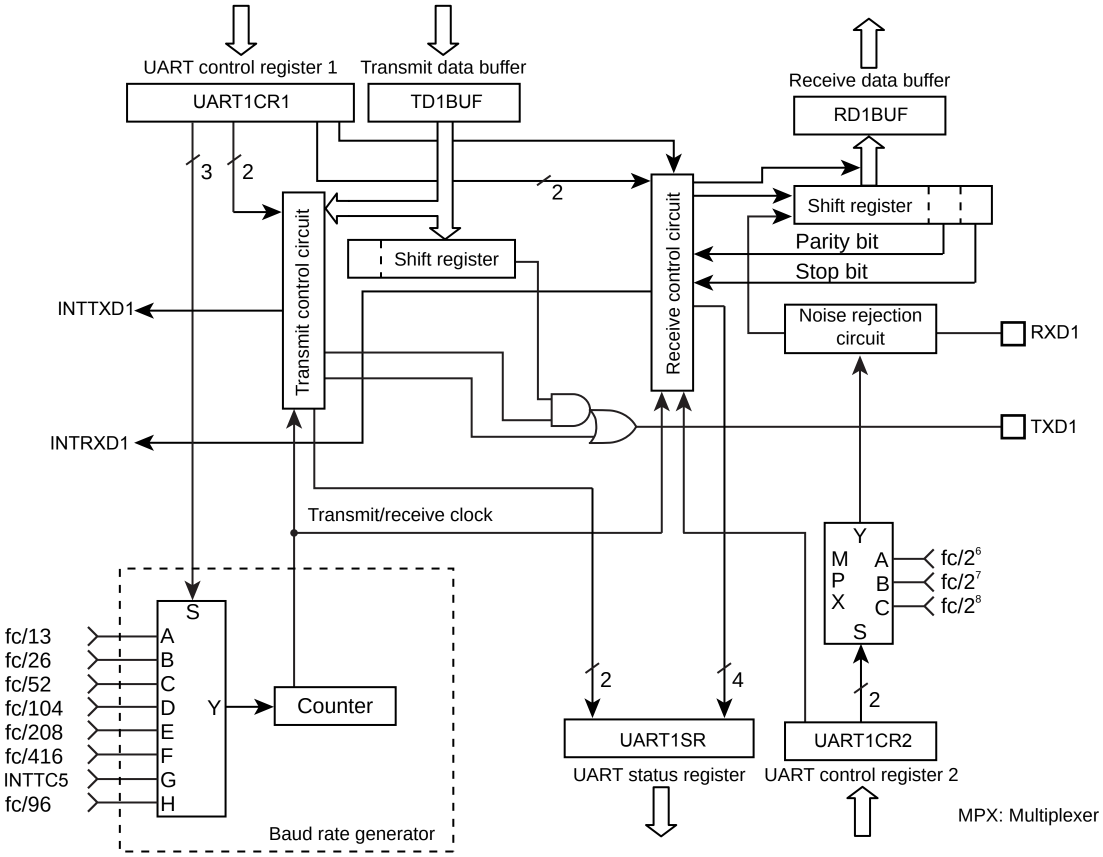


Figure 12-1 UART1 (Asynchronous Serial Interface)

12.2 Control

UART1 is controlled by the UART1 Control Registers (UART1CR1, UART1CR2). The operating status can be monitored using the UART status register (UART1SR).

UART1 Control Register1

	7	6	5	4	3	2	1	0	
UART1CR1 (0FE8H)	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: TC5 (Input INTTC5) 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: UART1CR1<RXE> and UART1CR1<TXE> should be set to “0” before UART1CR1<BRG> is changed.

UART1 Control Register2

	7	6	5	4	3	2	1	0	
UART1CR2 (0FE9H)						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 UART1CR2<RXDNC> = “01”, pulses longer than 96/fc [s] are always regarded as signals; when UART1CR2<RXDNC> = “10”, longer than 192/fc [s]; and when UART1CR2<RXDNC> = “11”, longer than 384/fc [s].

UART1 Status Register

UART1SR (0FE8H)	7	6	5	4	3	2	1	0	
	PERR	FERR	OERR	RBFL	TEND	TBEP			(Initial value: 0000 11**)

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.

UART1 Receive Data Buffer

RD1BUF (0FEAH)	7	6	5	4	3	2	1	0	Read only
									(Initial value: 0000 0000)

UART1 Transmit Data Buffer

TD1BUF (0FEAH)	7	6	5	4	3	2	1	0	Write only
									(Initial value: 0000 0000)

12.3 Transfer Data Format

In UART1, an one-bit start bit (Low level), stop bit (Bit length selectable at high level, by UART1CR1<STBT>), and parity (Select parity in UART1CR1<PE>; even- or odd-numbered parity by UART1CR1<EVEN>) are added to the transfer data. The transfer data formats are shown as follows.

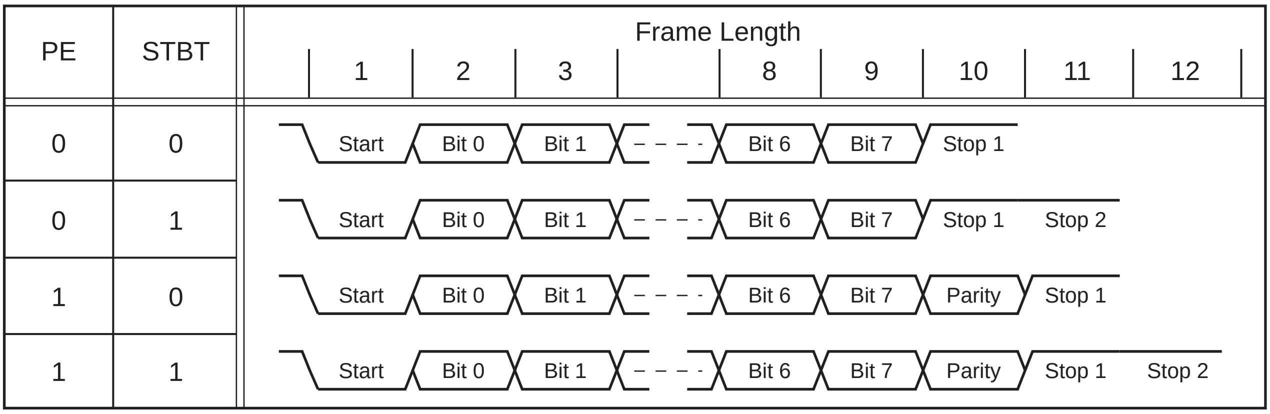


Figure 12-2 Transfer Data Format

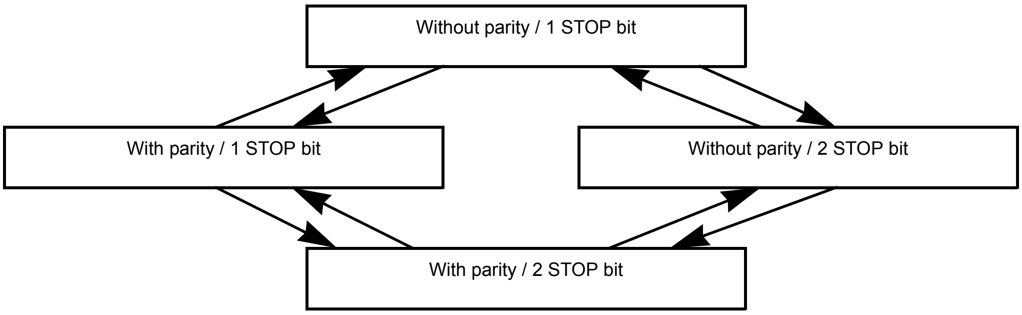


Figure 12-3 Caution on Changing Transfer Data Format

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 UART1 is set of UART1CR1<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

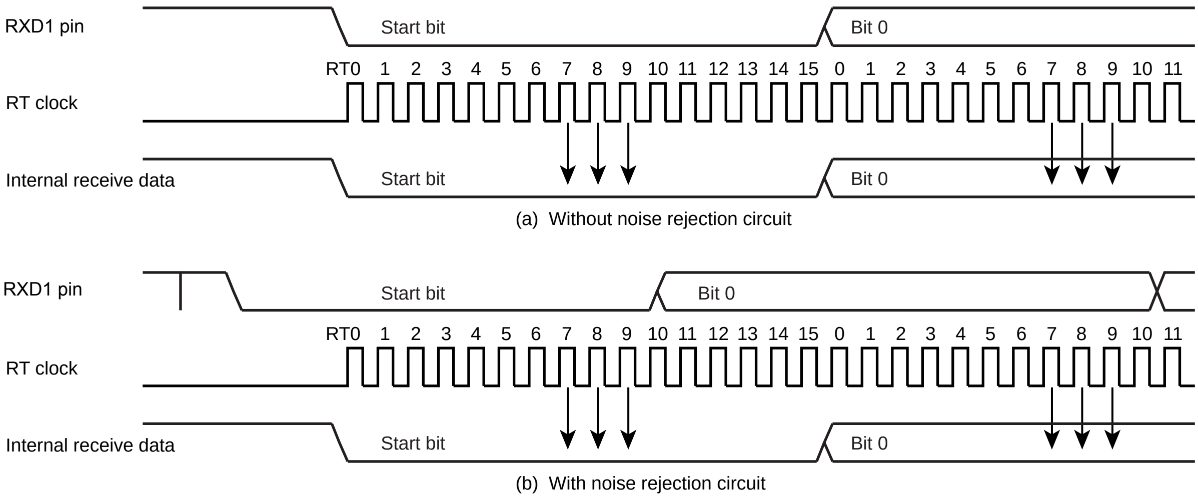
When TC5 is used as the UART1 transfer rate (when UART1CR1<BRG> = “110”), the transfer clock and transfer rate are determined as follows:

Transfer clock [Hz] = TC5 source clock [Hz] / TTREG5 setting value

Transfer Rate [baud] = Transfer clock [Hz] / 16

12.5 Data Sampling Method

The UART1 receiver keeps sampling input using the clock selected by UART1CR1<BRG> until a start bit is detected in RXD1 pin input. RT clock starts detecting “L” level of the RXD1 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).


RXD1 pin
Start bit
Bit 0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
Bit 0

(a) Without noise rejection circuit

RXD1 pin
Start bit
Bit 0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
Bit 0

(b) With noise rejection circuit

Figure 12-4 Data Sampling Method

12.6 STOP Bit Length

Select a transmit stop bit length (1 bit or 2 bits) by UART1CR1<STBT>.

12.7 Parity

Set parity / no parity by UART1CR1<PE> and set parity type (Odd- or Even-numbered) by UART1CR1<EVEN>.

12.8 Transmit/Receive Operation

12.8.1 Data Transmit Operation

Set UART1CR1<TXE> to “1”. Read UART1SR to check UART1SR<TBEP> = “1”, then write data in TD1BUF (Transmit data buffer). Writing data in TD1BUF zero-clears UART1SR<TBEP>, transfers the data to the transmit shift register and the data are sequentially output from the TXD1 pin. The data output include a one-bit start bit, stop bits whose number is specified in UART1CR1<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UART1CR1<BRG>. When data transmit starts, transmit buffer empty flag UART1SR<TBEP> is set to “1” and an INTTXD1 interrupt is generated.

While UART1CR1<TXE> = “0” and from when “1” is written to UART1CR1<TXE> to when send data are written to TD1BUF, the TXD1 pin is fixed at high level.
When transmitting data, first read UART1SR, then write data in TD1BUF. Otherwise, UART1SR<TBEP> is not zero-cleared and transmit does not start.

12.8.2 Data Receive Operation

Set UART1CR1<RXE> to “1”. When data are received via the RXD1 pin, the receive data are transferred to RD1BUF (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 RD1BUF (Receive data buffer). Then the receive buffer full flag UART1SR<RBFL> is set and an INTRXD1 interrupt is generated. Select the data transfer baud rate using UART1CR1<BRG>.

If an overrun error (OERR) occurs when data are received, the data are not transferred to RD1BUF (Receive data buffer) but discarded; data in the RD1BUF are not affected.

Note: When a receive operation is disabled by setting UART1CR1<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 UART1SR<PERR> is set to "1". The UART1SR<PERR> is cleared to "0" when the RD1BUF is read after reading the UART1SR.

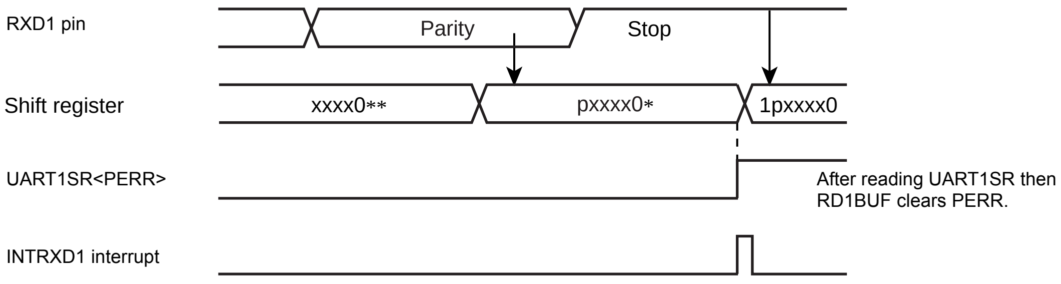


Figure 12-5 Generation of Parity Error

12.9.2 Framing Error

When "0" is sampled as the stop bit in the receive data, framing error flag UART1SR<FERR> is set to "1". The UART1SR<FERR> is cleared to "0" when the RD1BUF is read after reading the UART1SR.

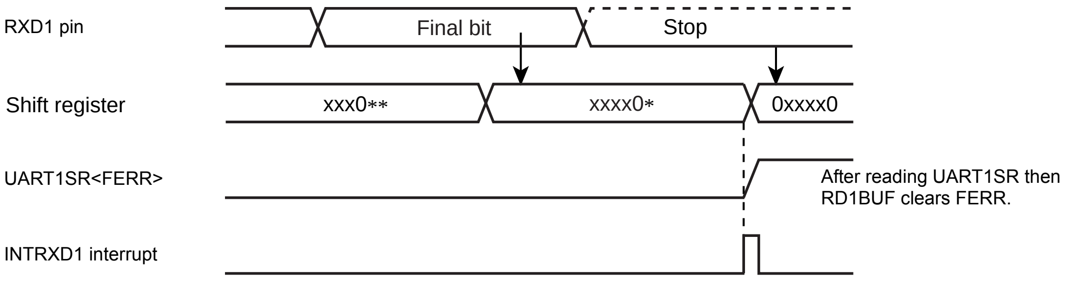


Figure 12-6 Generation of Framing Error

12.9.3 Overrun Error

When all bits in the next data are received while unread data are still in RD1BUF, overrun error flag UART1SR<OERR> is set to "1". In this case, the receive data is discarded; data in RD1BUF are not affected. The UART1SR<OERR> is cleared to "0" when the RD1BUF is read after reading the UART1SR.

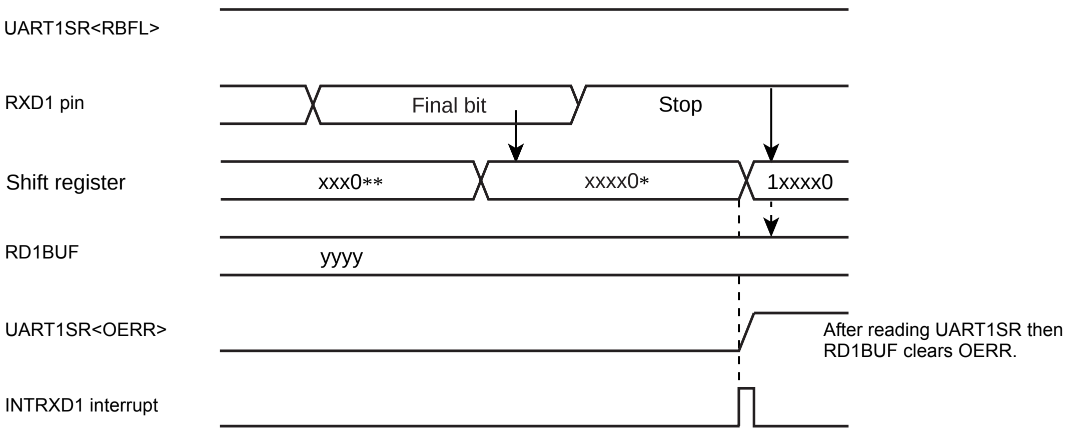


Figure 12-7 Generation of Overrun Error

Note: Receive operations are disabled until the overrun error flag UART1SR<OERR> is cleared.

12.9.4 Receive Data Buffer Full

Loading the received data in RD1BUF sets receive data buffer full flag UART1SR<RBFL> to "1". The UART1SR<RBFL> is cleared to “0” when the RD1BUF is read after reading the UART1SR.

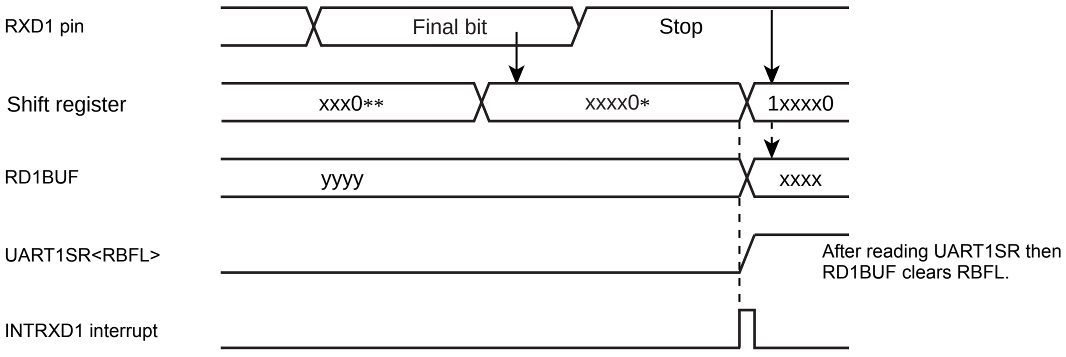


Figure 12-8 Generation of Receive Data Buffer Full

Note: If the overrun error flag UART1SR<OERR> is set during the period between reading the UART1SR and reading the RD1BUF, it cannot be cleared by only reading the RD1BUF. Therefore, after reading the RD1BUF, read the UART1SR 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 TD1BUF, that is, when data in TD1BUF are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag UART1SR<TBEP> is set to “1”. The UART1SR<TBEP> is cleared to “0” when the TD1BUF is written after reading the UART1SR.

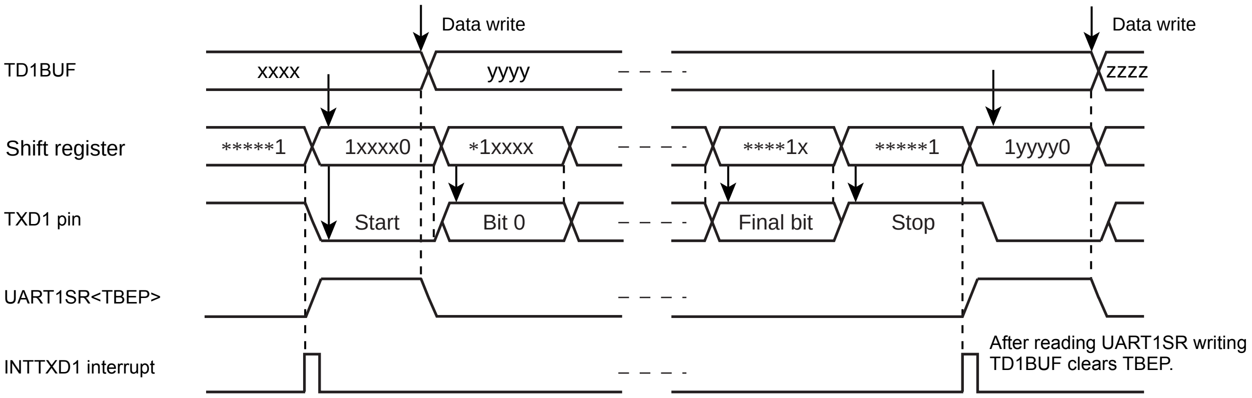


Figure 12-9 Generation of Transmit Data Buffer Empty

12.9.6 Transmit End Flag

When data are transmitted and no data is in TD1BUF (UART1SR<TBEP> = “1”), transmit end flag UART1SR<TEND> is set to “1”. The UART1SR<TEND> is cleared to “0” when the data transmit is started after writing the TD1BUF.

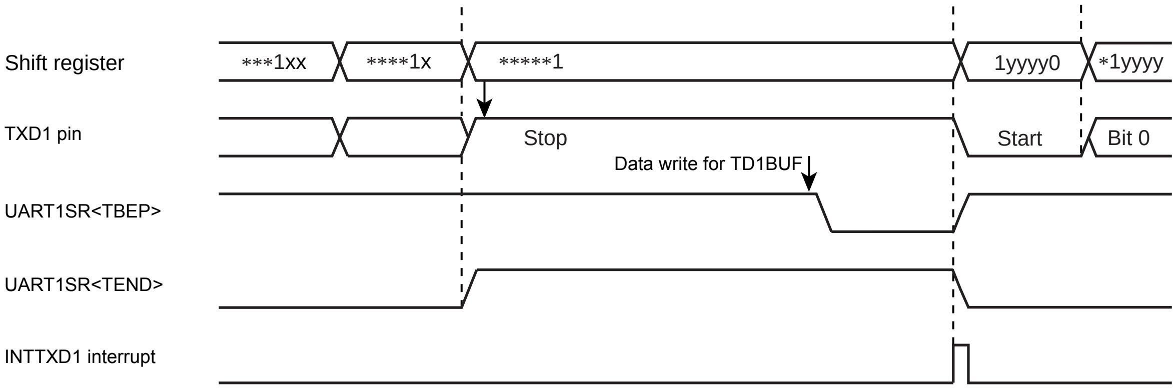


Figure 12-10 Generation of Transmit End Flag and Transmit Data Buffer Empty

13. Asynchronous Serial interface (UART0)

13.1 Configuration

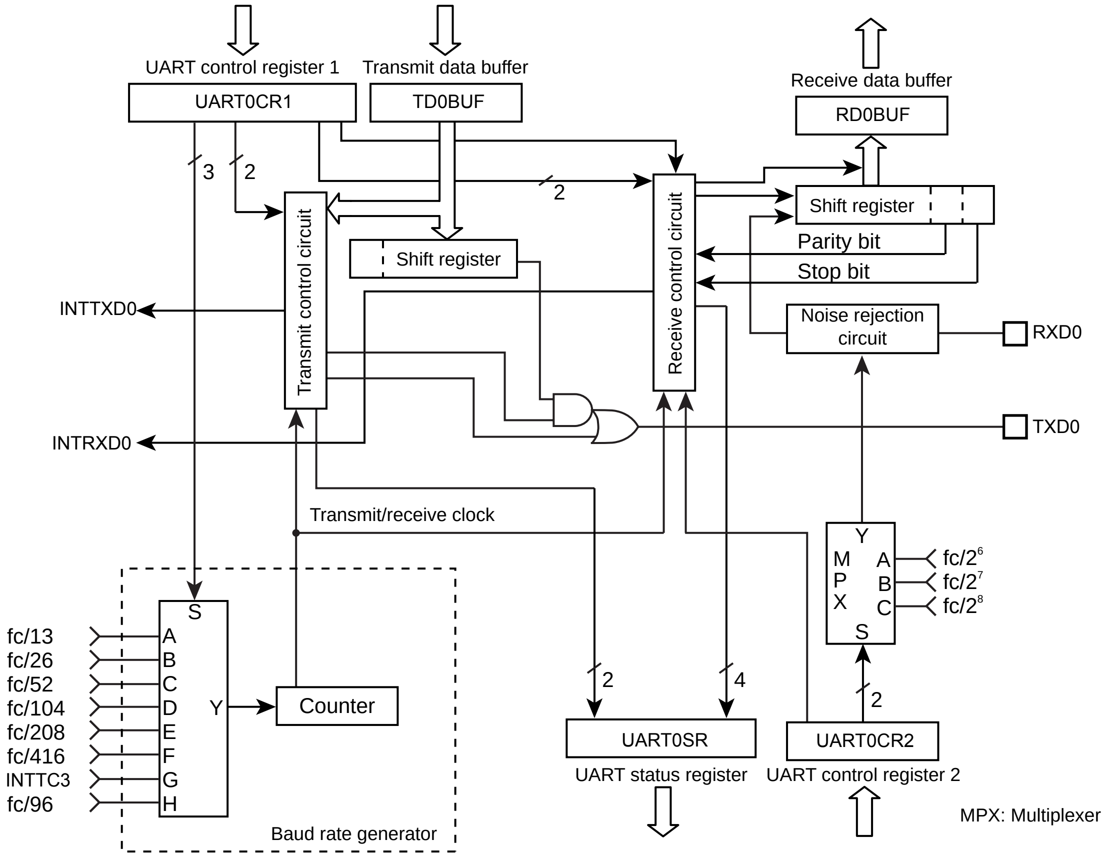


Figure 13-1 UART0 (Asynchronous Serial Interface)

13.2 Control

UART0 is controlled by the UART0 Control Registers (UART0CR1, UART0CR2). The operating status can be monitored using the UART status register (UART0SR).

UART0 Control Register1

UART0CR1 (0FE5H)	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: UART0CR1<RXE> and UART0CR1<TXE> should be set to “0” before UART0CR1<BRG> is changed.

UART0 Control Register2

UART0CR2 (0FE6H)	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 UART0CR2<RXDNC> = “01”, pulses longer than 96/fc [s] are always regarded as signals; when UART0CR2<RXDNC> = “10”, longer than 192/fc [s]; and when UART0CR2<RXDNC> = “11”, longer than 384/fc [s].

UART0 Status Register

	7	6	5	4	3	2	1	0	
UART0SR (0FE5H)	PERR	FERR	OERR	RBFL	TEND	TBEP			(Initial value: 0000 11**)

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.

UART0 Receive Data Buffer

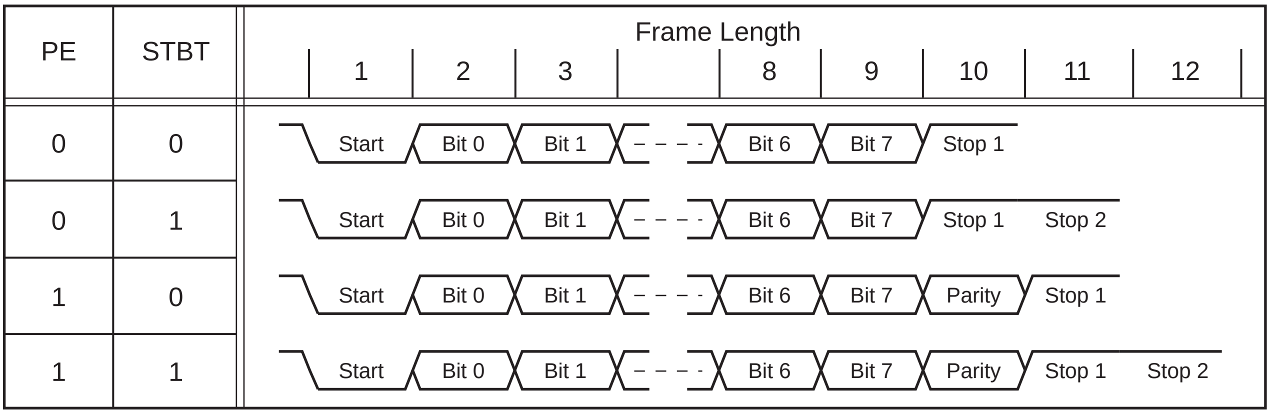
	7	6	5	4	3	2	1	0	Read only
RD0BUF (0FE7H)									(Initial value: 0000 0000)

UART0 Transmit Data Buffer

	7	6	5	4	3	2	1	0	Write only
TD0BUF (0FE7H)									(Initial value: 0000 0000)

13.3 Transfer Data Format

In UART0, an one-bit start bit (Low level), stop bit (Bit length selectable at high level, by UART0CR1<STBT>), and parity (Select parity in UART0CR1<PE>; even- or odd-numbered parity by UART0CR1<EVEN>) are added to the transfer data. The transfer data formats are shown as follows.



PE	STBT	Frame Length 1	2	3		8	9	10	11	12
0	0	Start	Bit 0	Bit 1	– – – -	Bit 6	Bit 7	Stop 1		
0	1	Start	Bit 0	Bit 1	– – – -	Bit 6	Bit 7	Stop 1	Stop 2	
1	0	Start	Bit 0	Bit 1	– – – -	Bit 6	Bit 7	Parity	Stop 1	
1	1	Start	Bit 0	Bit 1	– – – -	Bit 6	Bit 7	Parity	Stop 1	Stop 2

Figure 13-2 Transfer Data Format

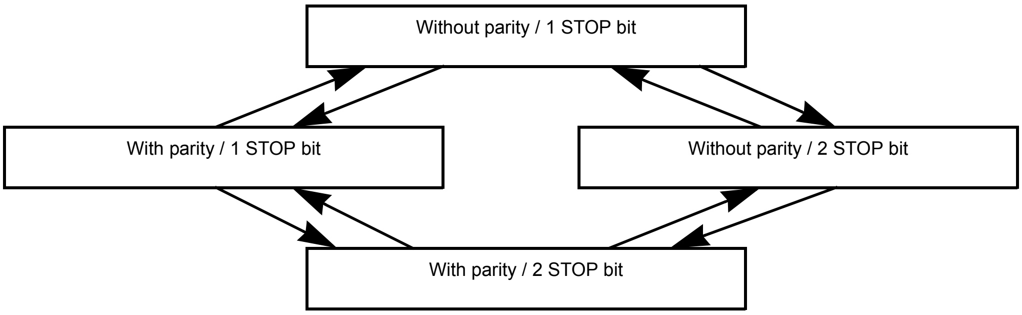


Figure 13-3 Caution on Changing Transfer Data Format

Note: In order to switch the transfer data format, perform transmit operations in the above Figure 13-3 sequence except for the initial setting.

13.4 Transfer Rate

The baud rate of UART0 is set of UART0CR1<BRG>. The example of the baud rate are shown as follows.

Table 13-1 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

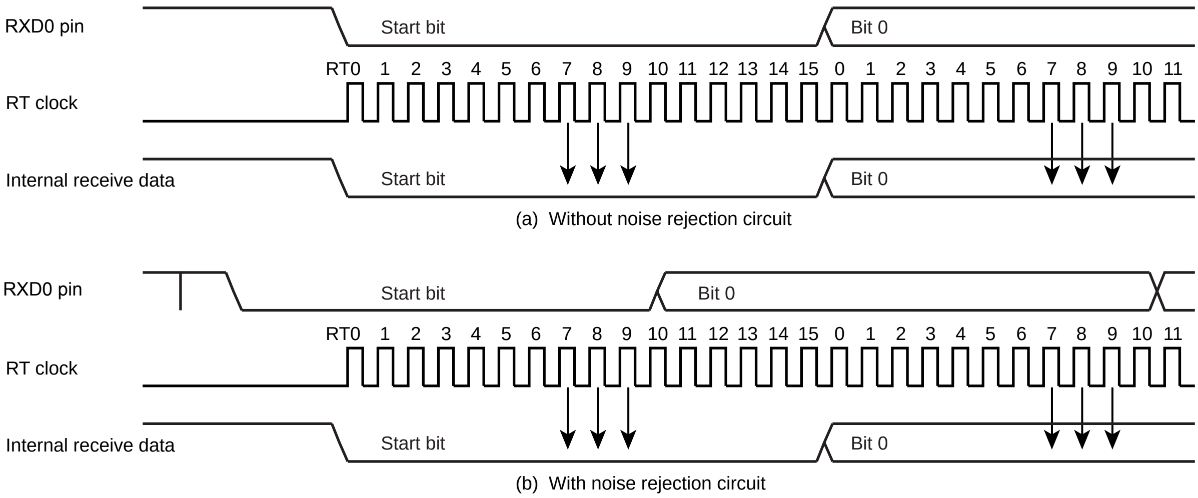
When TC3 is used as the UART0 transfer rate (when UART0CR1<BRG> = “110”), the transfer clock and transfer rate are determined as follows:

Transfer clock [Hz] = TC3 source clock [Hz] / TTREG3 setting value

Transfer Rate [baud] = Transfer clock [Hz] / 16

13.5 Data Sampling Method

The UART0 receiver keeps sampling input using the clock selected by UART0CR1<BRG> until a start bit is detected in RXD0 pin input. RT clock starts detecting “L” level of the RXD0 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).


RXD0 pin
Start bit
Bit 0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
Bit 0
RXD0 pin
Start bit
Bit 0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
Bit 0

(a) Without noise rejection circuit

(b) With noise rejection circuit

Figure 13-4 Data Sampling Method

13.6 STOP Bit Length

Select a transmit stop bit length (1 bit or 2 bits) by UART0CR1<STBT>.

13.7 Parity

Set parity / no parity by UART0CR1<PE> and set parity type (Odd- or Even-numbered) by UART0CR1<EVEN>.

13.8 Transmit/Receive Operation

13.8.1 Data Transmit Operation

Set UART0CR1<TXE> to “1”. Read UART0SR to check UART0SR<TBEP> = “1”, then write data in TD0BUF (Transmit data buffer). Writing data in TD0BUF zero-clears UART0SR<TBEP>, transfers the data to the transmit shift register and the data are sequentially output from the TXD0 pin. The data output include a one-bit start bit, stop bits whose number is specified in UART0CR1<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UART0CR1<BRG>. When data transmit starts, transmit buffer empty flag UART0SR<TBEP> is set to “1” and an INTTXD0 interrupt is generated.

While UART0CR1<TXE> = “0” and from when “1” is written to UART0CR1<TXE> to when send data are written to TD0BUF, the TXD0 pin is fixed at high level.
When transmitting data, first read UART0SR, then write data in TD0BUF. Otherwise, UART0SR<TBEP> is not zero-cleared and transmit does not start.

13.8.2 Data Receive Operation

Set UART0CR1<RXE> to “1”. When data are received via the RXD0 pin, the receive data are transferred to RD0BUF (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 RD0BUF (Receive data buffer). Then the receive buffer full flag UART0SR<RBFL> is set and an INTRXD0 interrupt is generated. Select the data transfer baud rate using UART0CR1<BRG>.

If an overrun error (OERR) occurs when data are received, the data are not transferred to RD0BUF (Receive data buffer) but discarded; data in the RD0BUF are not affected.

Note: When a receive operation is disabled by setting UART0CR1<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.

13.9 Status Flag

13.9.1 Parity Error

When parity determined using the receive data bits differs from the received parity bit, the parity error flag UART0SR<PERR> is set to "1". The UART0SR<PERR> is cleared to "0" when the RD0BUF is read after reading the UART0SR.

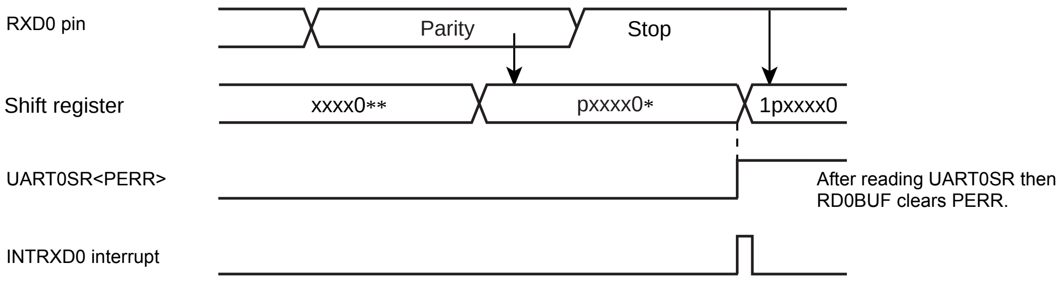


Figure 13-5 Generation of Parity Error

13.9.2 Framing Error

When "0" is sampled as the stop bit in the receive data, framing error flag UART0SR<FERR> is set to "1". The UART0SR<FERR> is cleared to "0" when the RD0BUF is read after reading the UART0SR.

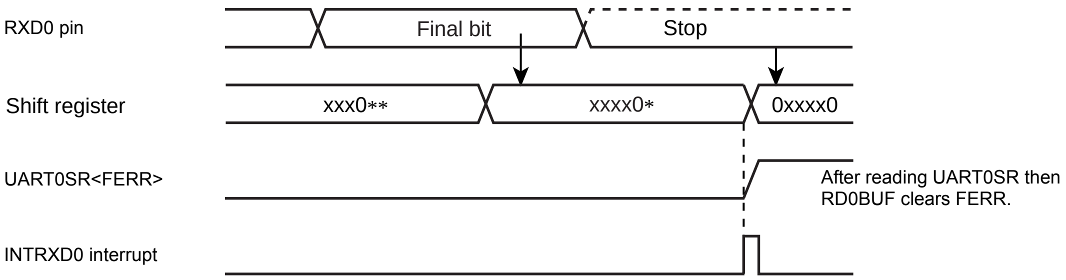


Figure 13-6 Generation of Framing Error

13.9.3 Overrun Error

When all bits in the next data are received while unread data are still in RD0BUF, overrun error flag UART0SR<OERR> is set to "1". In this case, the receive data is discarded; data in RD0BUF are not affected. The UART0SR<OERR> is cleared to "0" when the RD0BUF is read after reading the UART0SR.

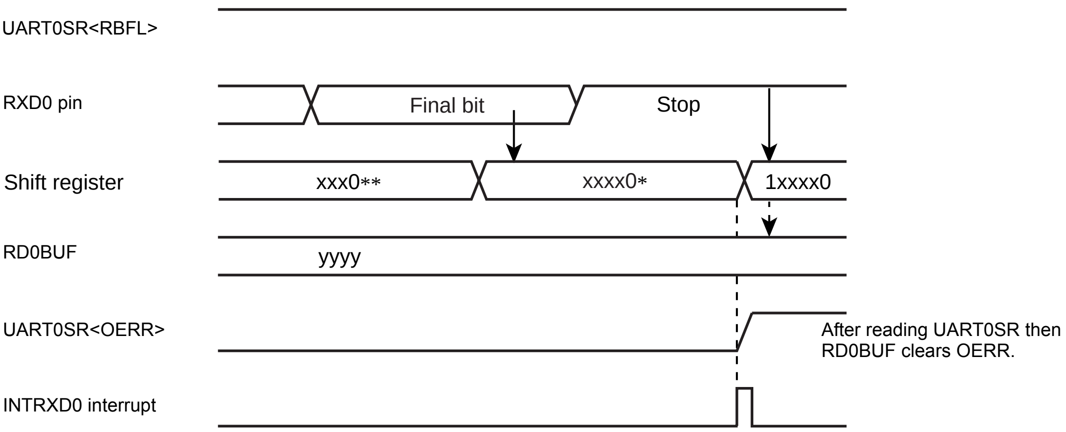


Figure 13-7 Generation of Overrun Error

Note: Receive operations are disabled until the overrun error flag UART0SR<OERR> is cleared.

13.9.4 Receive Data Buffer Full

Loading the received data in RD0BUF sets receive data buffer full flag UART0SR<RBFL> to "1". The UART0SR<RBFL> is cleared to “0” when the RD0BUF is read after reading the UART0SR.

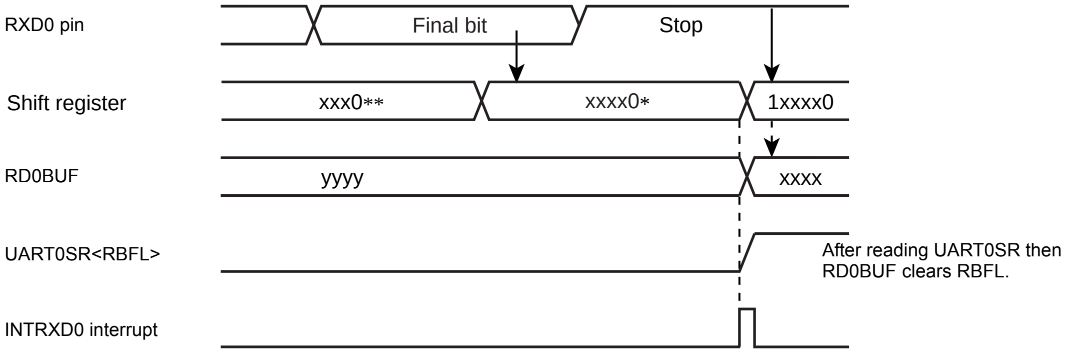


Figure 13-8 Generation of Receive Data Buffer Full

Note: If the overrun error flag UART0SR<OERR> is set during the period between reading the UART0SR and reading the RD0BUF, it cannot be cleared by only reading the RD0BUF. Therefore, after reading the RD0BUF, read the UART0SR again to check whether or not the overrun error flag which should have been cleared still remains set.

13.9.5 Transmit Data Buffer Empty

When no data is in the transmit buffer TD0BUF, that is, when data in TD0BUF are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag UART0SR<TBEP> is set to “1”. The UART0SR<TBEP> is cleared to “0” when the TD0BUF is written after reading the UART0SR.

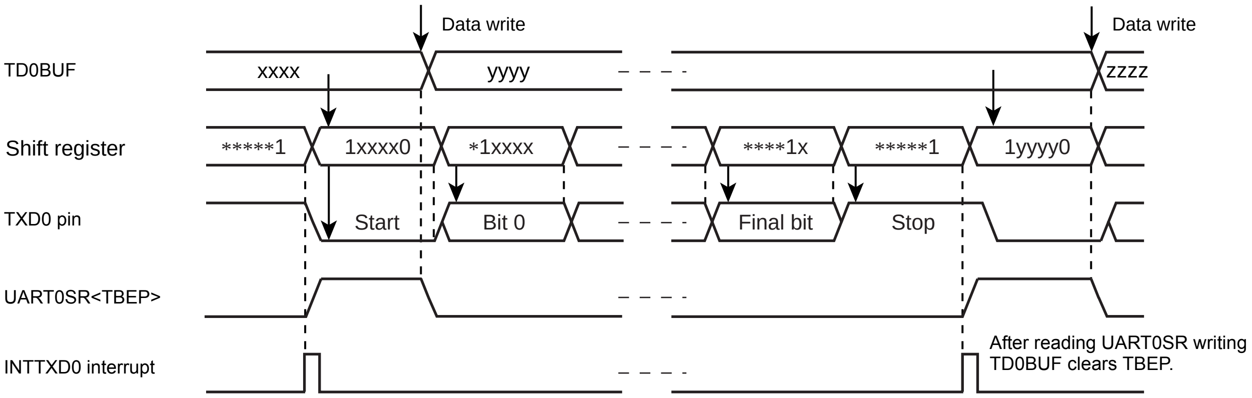


Figure 13-9 Generation of Transmit Data Buffer Empty

13.9.6 Transmit End Flag

When data are transmitted and no data is in TD0BUF (UART0SR<TBEP> = “1”), transmit end flag UART0SR<TEND> is set to “1”. The UART0SR<TEND> is cleared to “0” when the data transmit is started after writing the TD0BUF.

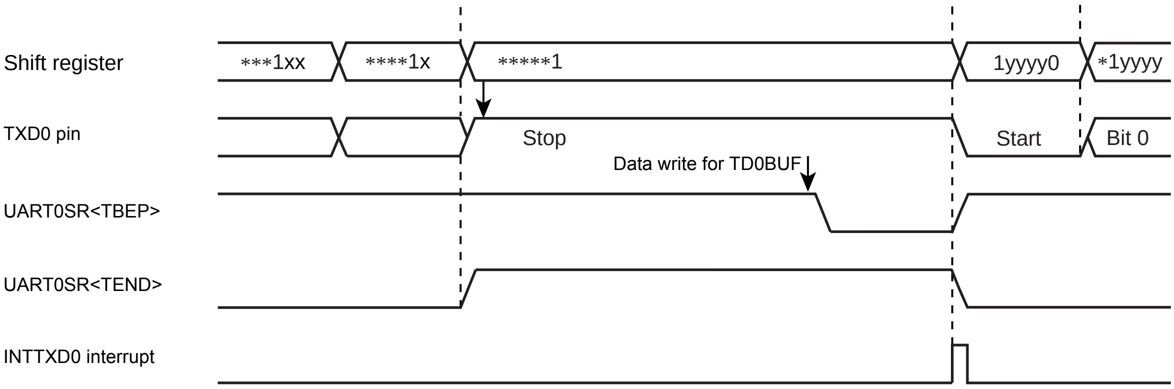


Figure 13-10 Generation of Transmit End Flag and Transmit Data Buffer Empty

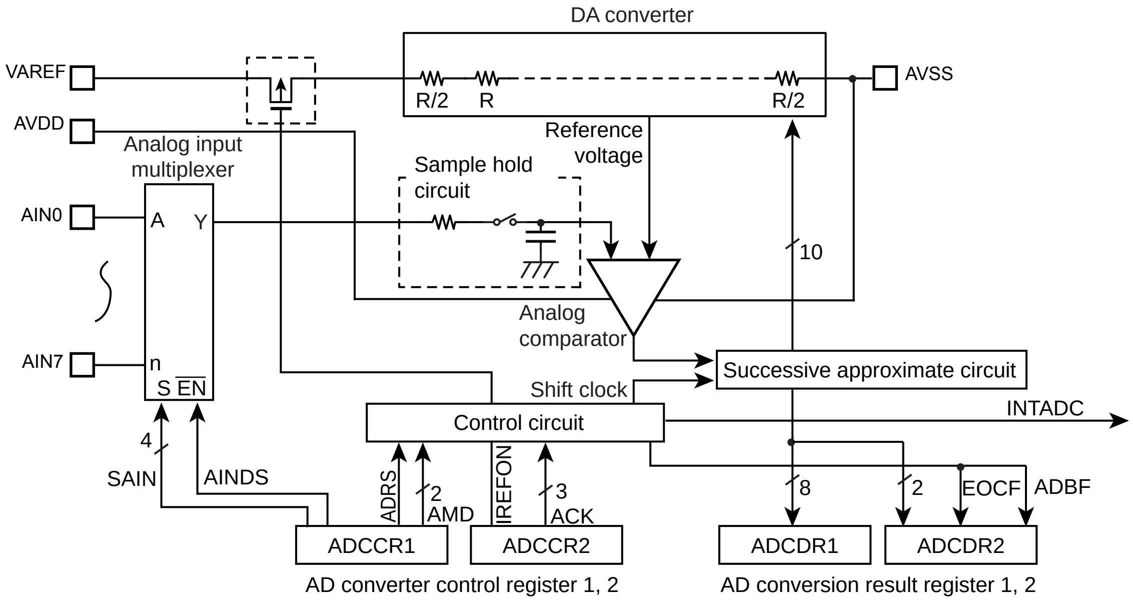
14. 10-bit AD Converter (ADC)

The TMP86CS28FG have a 10-bit successive approximation type AD converter.

14.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 14-1.

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 conbining a analog input port. For details, see the section on "I/O ports".

Figure 14-1 10-bit AD Converter

14.2 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 fter 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

ADCCR1 (0FE2H)	7	6	5	4	3	2	1	0	
	ADRS	AMD		AINDS	SAIN				(Initial value: 0001 0000)

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. Therfore, to use AD converter again, set the ADCCR1 newly after returning to NORMAL1 or NORMAL2 mode.

AD Converter Control Register 2

ADCCR2	7	6	5	4	3	2	1	0	
(0FE3H)			IREFON	"1"	ACK			"0"	(Initial value: **0* 000*)

IREFON	DA converter (Ladder resistor) connection control	0: Connected only during AD conversion 1: Always connected	R/W
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	

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. Therfore, to use AD converter again, set the ADCCR2 newly after returning to NORMAL1 or NORMAL2 mode.

Table 14-1 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 Analog reference voltage (VAREF) .

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

AD Converted value Register 1

ADCDR1	7	6	5	4	3	2	1	0	
(0FE1H)	AD09	AD08	AD07	AD06	AD05	AD04	AD03	AD02	(Initial value: 0000 0000)

AD Converted value Register 2

ADCDR2	7	6	5	4	3	2	1	0	
(0FE0H)	AD01	AD00	EOCF	ADBF					(Initial value: 0000 ****)

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. Therfore, 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.

14.3 Function

14.3.1 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).

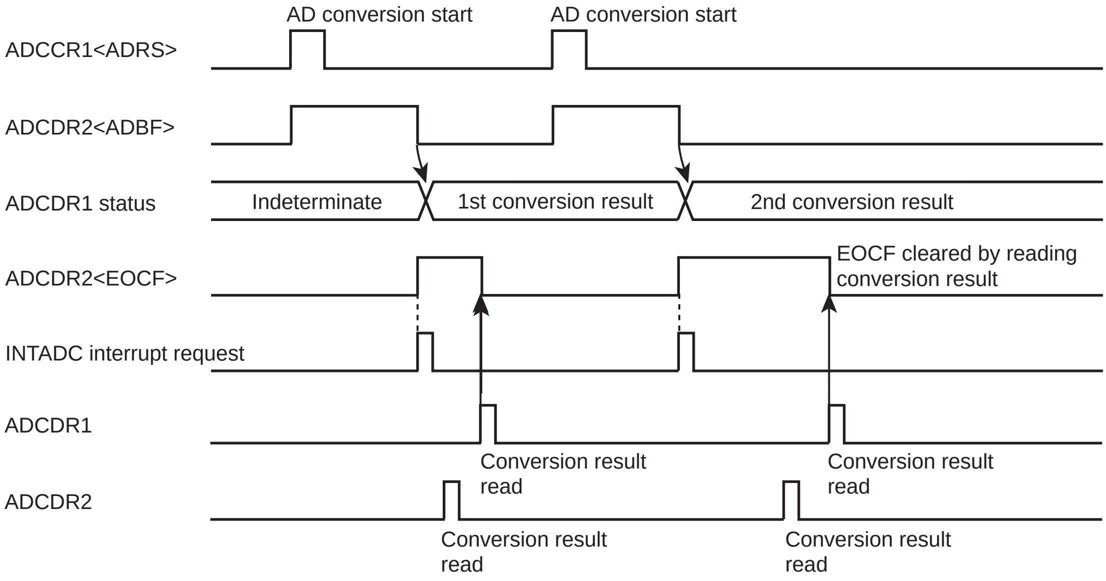


Figure 14-2 Software Start Mode

14.3.2 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.

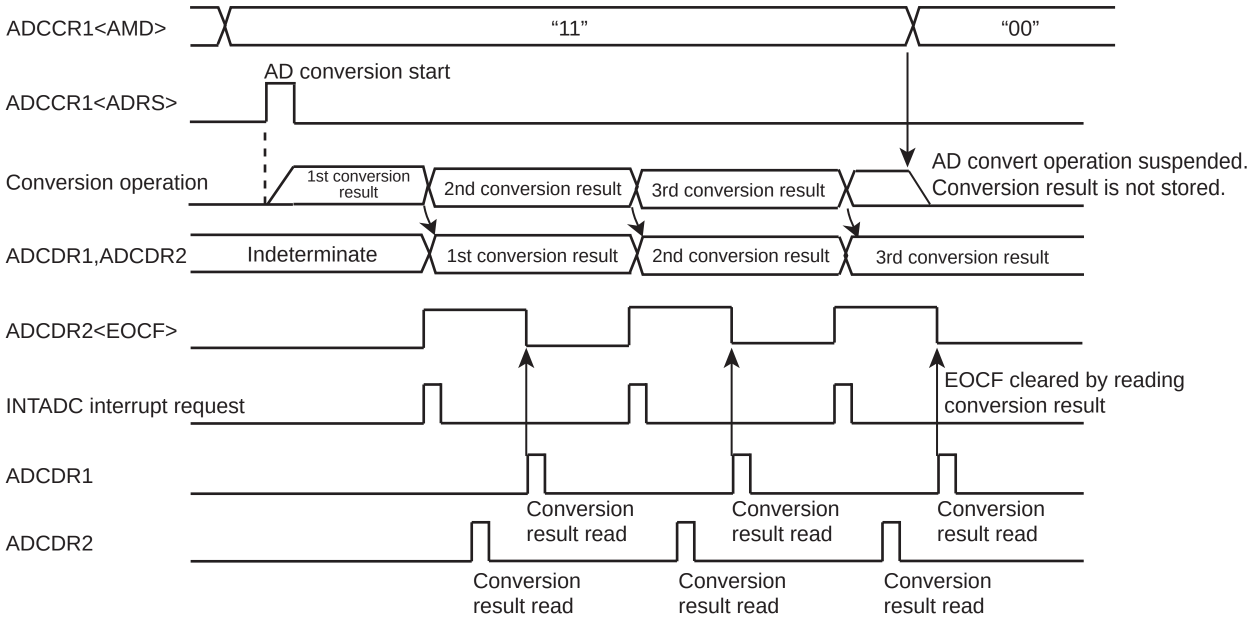


Figure 14-3 Repeat Mode

14.3.3 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 14-1 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 µs 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 nd store the upper 8 bits in address 0009FH in RAM. The operation mode is software start mode.

```
           : (port setting)   :                         ;Set port register approrriately 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               (ADCDR2) . 5              ; EOCF= 1 ?
           JRS                T, SLOOP

           LD                 A , (ADCDR2)              ; Read result data
           LD                 (9EH) , A
           LD                 A , (ADCDR1)              ; Read  result data
           LD                 (9FH), A
```

14.4 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.

14.5 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 14-4.

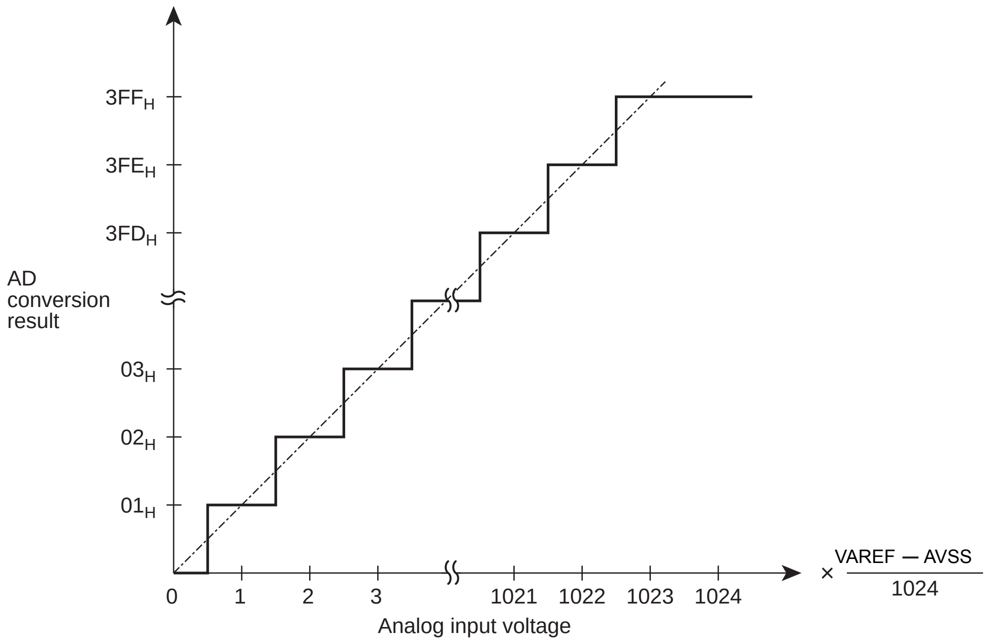


Figure 14-4 Analog Input Voltage and AD Conversion Result (Typ.)

14.6 Precautions about AD Converter

14.6.1 Analog input pin voltage range

Make sure the analog input pins (AIN0 to AIN7) are used at voltages within VAREF 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.

14.6.2 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.

14.6.3 Noise Countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 14-5. 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 kΩ or less. Toshiba also recommends attaching a capacitor external to the chip.

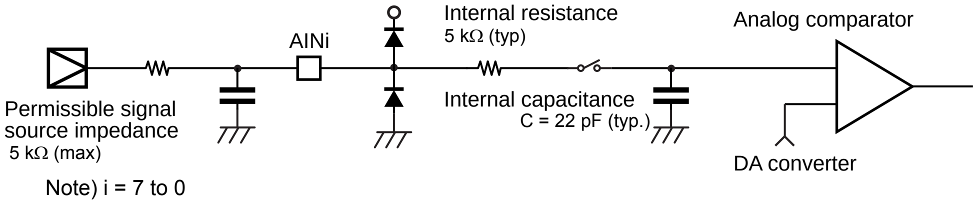


Figure 14-5 Analog Input Equivalent Circuit and Example of Input Pin Processing

15. Key-on Wakeup (KWU)

In the TMP86CS28FG, the STOP mode is released by not only P20($\overline{\text{INT5/STOP}}$) 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

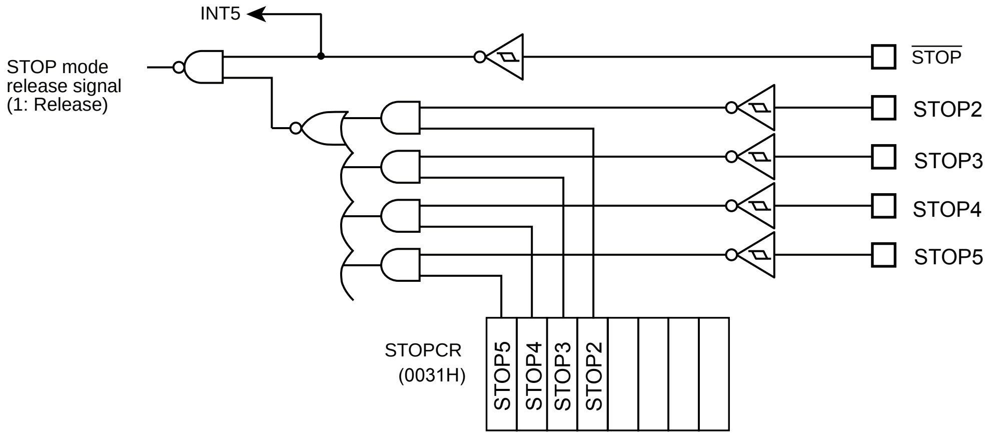


Figure 15-1 Key-on Wakeup Circuit

15.2 Control

STOP2 to STOP5 pins can 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 STOPCR 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 (STOPCR) 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 STOPCR 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 STOPCR, 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 (STOPCR) 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).

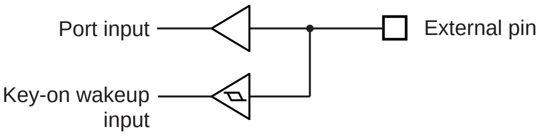


Figure 15-2 Key-on Wakeup Input and Port Input

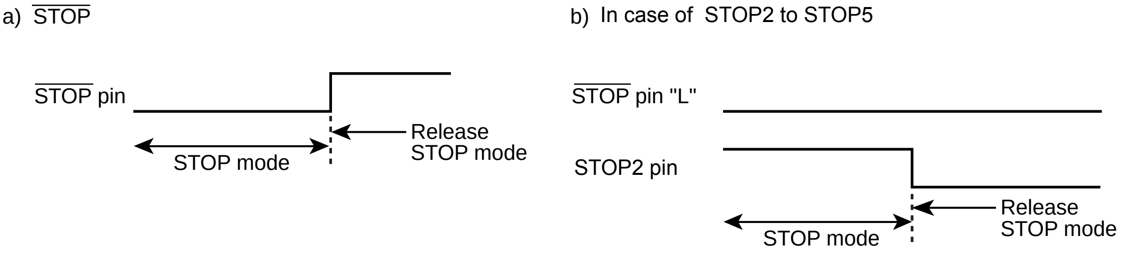

a) $\overline{\text{STOP}}$
$\overline{\text{STOP}}$ pin
STOP mode
Release STOP mode
b) In case of STOP2 to STOP5
$\overline{\text{STOP}}$ pin "L"
STOP2 pin
STOP mode
Release STOP mode

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. LCD Driver

The TMP86CS28FG has a driver and control circuit to directly drive the liquid crystal device (LCD). The pins to be connected to LCD are as follows:

1. Segment output port 40 pins (SEG39 to SEG0)
2. Common output port4 pins (COM3 to COM0)

In addition, C0, C1, V1, V2, V3 pin are provided for the LCD driver's booster circuit.

The devices that can be directly driven is selectable from LCD of the following drive methods:

1. 1/4 Duty (1/3 Bias) LCD	Max 160 Segments(8 segments × 20 digits)
2. 1/3 Duty (1/3 Bias) LCD	Max 120 Segments(8 segments × 15 digits)
3. 1/2 Duty (1/2 Bias) LCD	Max 80 Segments(8 segments × 10 digits)
4. Static LCD	Max 40 Segments(8 segments × 5 digits)

16.1 Configuration

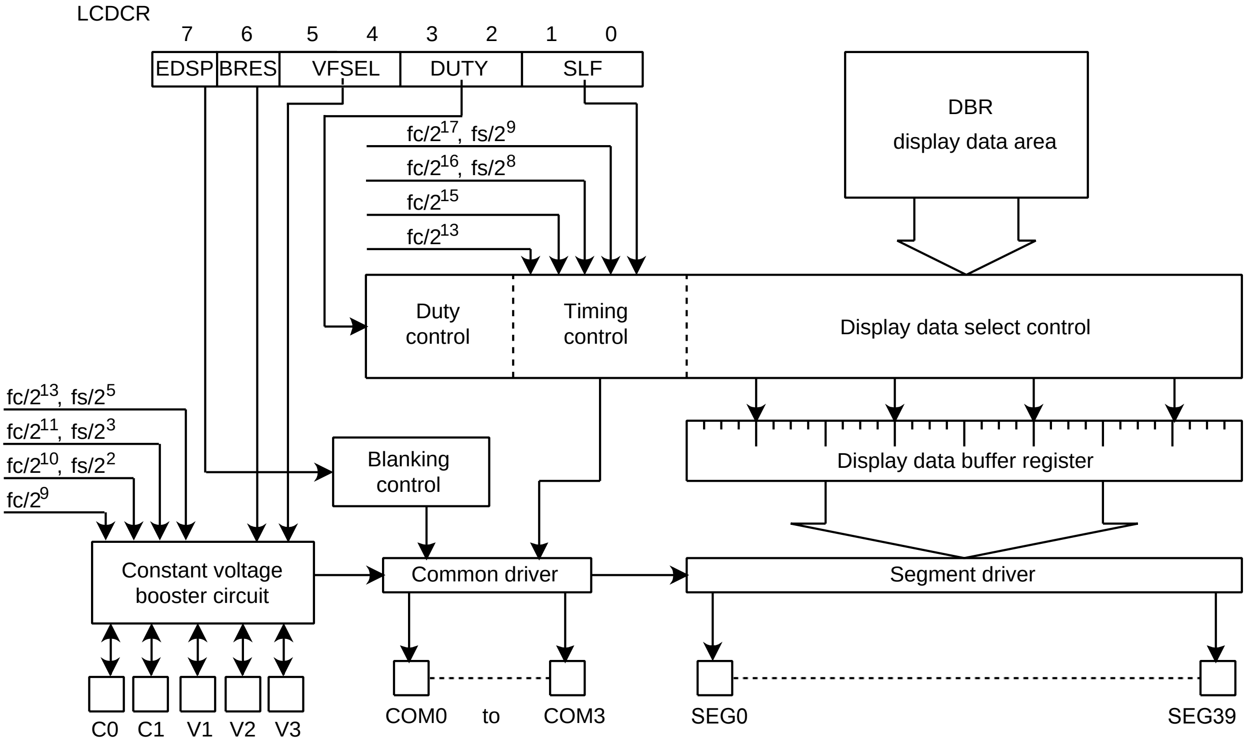


Figure 16-1 LCD Driver

Note: The LCD driver incorporates a dedicated divider circuit. Therefore, the break function of a debugger (development tool) will not stop LCD driver output.

16.2 Control

The LCD driver is controlled using the LCD control register (LCDCR). The LCD driver’s display is enabled using the EDSP.

LCD Driver Control Register

LCDCR (0FD9H)	7	6	5	4	3	2	1	0	
	EDSP	BRES	VFSEL		DUTY		SLF		(Initial value: 0000 0000)

EDSP	LCD Display Control	0: Blanking 1: Enables LCD display (Blanking is released)				R/W
BRES	Booster circuit control	0: Disable (use divider resistance) 1: Enable				
VFSEL	Selection of boost frequency		NORMAL1/2, IDLE/1/2 mode		SLOW1/2, SLEEP0/1/2 mode	
			DV7CK = 0	DV7CK = 1		
		00	$fc/2^{13}$	$fs/2^{5}$	$fs/2^{5}$	
		01	$fc/2^{11}$	$fs/2^{3}$	$fs/2^{3}$	
		10	$fc/2^{10}$	$fs/2^{2}$	$fs/2^{2}$	
		11	$fc/2^{9}$	$fc/2^{9}$	–	
DUTY	Selection of driving methods	00: 1/4 Duty (1/3 Bias) 01: 1/3 Duty (1/3 Bias) 10: 1/2 Duty (1/2 Bias) 11: Static				
SLF	Selection of LCD frame frequency		NORMAL1/2, IDLE/1/2 mode		SLOW1/2, SLEEP0/1/2 mode	
			DV7CK = 0	DV7CK = 1		
		00	$fc/2^{17}$	$fs/2^{9}$	$fs/2^{9}$	
		01	$fc/2^{16}$	fs/28	$fs/2^{8}$	
		10	$fc/2^{15}$	$fc/2^{15}$	–	
		11	$fc/2^{13}$	$fc/2^{13}$	–	

Note 1: When <BRES>(Booster circuit control) is set to “0”, $V_{DD} \geq V3 \geq V2 \geq V1 \geq V_{SS}$ should be satisfied.
When <BRES> is set to “1”, 5.5 [V] $\geq V3 \geq V_{DD}$ should be satisfied.
If these conditions are not satisfied, it not only affects the quality of LCD display but also may damage the device due to over voltage of the port.

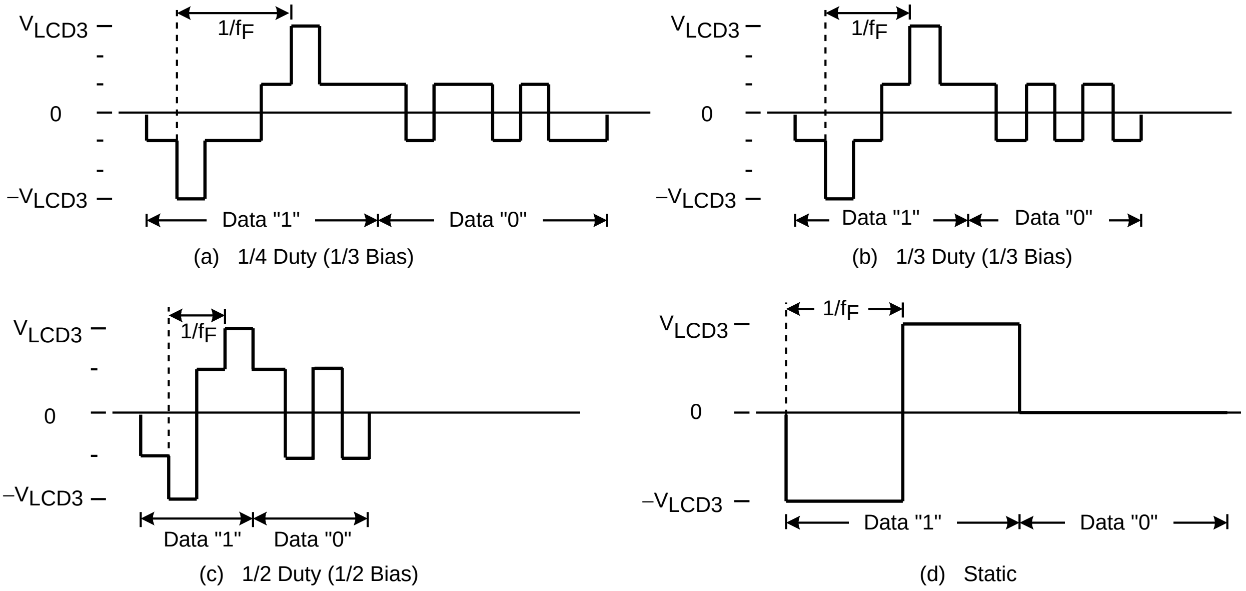
Note 2: When used as the booster circuit, bias should be composed to 1/3. Therefore, do not set LCDCR<DUTY> to "10" or "11" when the booster circuit is enable.

Note 3: Do not set SLF to “10” or “11” in SLOW1/2 modes.

Note 4: Do not set VFSEL to “11” SLOW1/2 modes.

16.2.1 LCD driving methods

As for LCD driving method, 4 types can be selected by LCDCR<DUTY>. The driving method is initialized in the initial program according to the LCD used.



Note 1: f_F: Frame frequency

Note 2: V_{LCD3}: LCD drive voltage

Figure 16-2 LCD Drive Waveform (COM-SEG pins)

16.2.2 Frame frequency

Frame frequency (f_F) is set according to driving method and base frequency as shown in the following Table 16-1. The base frequency is selected by LCDCR<SLF> according to the frequency fc and fs of the basic clock to be used.

Table 16-1 Setting of LCD Frame Frequency

(a) At the single clock mode. At the dual clock mode (DV7CK = 0).

SLF	Base frequency [Hz]	Frame frequency [Hz]			
		1/4 Duty	1/3 Duty	1/2 Duty	Static
00	$\frac{fc}{2^{17}}$	$\frac{fc}{2^{17}}$	$\frac{4}{3} \bullet \frac{fc}{2^{17}}$	$\frac{4}{2} \bullet \frac{fc}{2^{17}}$	$\frac{fc}{2^{17}}$
	(fc = 16 MHz)	122	163	244	122
	(fc = 8 MHz)	61	81	122	61
01	$\frac{fc}{2^{16}}$	$\frac{fc}{2^{16}}$	$\frac{4}{3} \bullet \frac{fc}{2^{16}}$	$\frac{4}{2} \bullet \frac{fc}{2^{16}}$	$\frac{fc}{2^{16}}$
	(fc = 8 MHz)	122	163	244	122
	(fc = 4 MHz)	61	81	122	61
10	$\frac{fc}{2^{15}}$	$\frac{fc}{2^{15}}$	$\frac{4}{3} \bullet \frac{fc}{2^{15}}$	$\frac{4}{2} \bullet \frac{fc}{2^{15}}$	$\frac{fc}{2^{15}}$
	(fc = 4 MHz)	122	163	244	122
	(fc = 2 MHz)	61	81	122	61
11	$\frac{fc}{2^{13}}$	$\frac{fc}{2^{13}}$	$\frac{4}{3} \bullet \frac{fc}{2^{13}}$	$\frac{4}{2} \bullet \frac{fc}{2^{13}}$	$\frac{fc}{2^{13}}$
	(fc = 1 MHz)	122	163	244	122

Note: fc: High-frequency clock [Hz]

Table 16-2

(b) At the dual clock mode (DV7CK = 1 or SYSCK = 1)

SLF	Base frequency [Hz]	Frame frequency [Hz]			
		1/4 Duty	1/3 Duty	1/2 Duty	Static
00	$\frac{fs}{2^{9}}$	$\frac{fs}{2^{9}}$	$\frac{4}{3} \bullet \frac{fs}{2^{9}}$	$\frac{4}{2} \bullet \frac{fs}{2^{9}}$	$\frac{fs}{2^{9}}$
	(fs = 32.768 kHz)	64	85	128	64
01	$\frac{fs}{2^{8}}$	$\frac{fs}{2^{8}}$	$\frac{4}{3} \bullet \frac{fs}{2^{8}}$	$\frac{4}{2} \bullet \frac{fs}{2^{8}}$	$\frac{fs}{2^{8}}$
	(fs = 32.768 kHz)	128	171	256	128

Note: fs: Low-frequency clock [Hz]

16.2.3 Driving method for LCD driver

In the TMP86CS28FG, LCD driving voltages can be generated using either an internal booster circuit or an external resistor divider. This selection is made in LCDCR<BRES>.

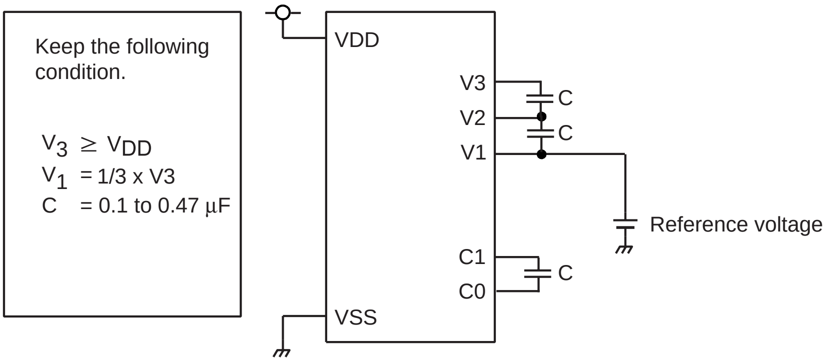
16.2.3.1 When using the booster circuit (LCDCR<BRES>="1")

When the reference voltage is connected to the V1 pin, the booster circuit boosts the reference voltage twofold (V2) or threefold (V3) to generate the output voltages for segment/common signals. When the reference voltage is connected to the V2 pin, it is reduced to 1/2 (V1) or boosted to 3/2 (V3). When the reference voltage is connected to the V3 pin, it is reduced to 1/3 (V1) or 2/3 (V2).

LCDCR<VFSEL> is used to select the reference frequency in the booster circuit. The faster the boosting frequency, the higher the segment/common drive capability, but power consumption is increased. Conversely, the slower the boosting frequency, the lower the segment/common drive capability, but power consumption is reduced. If the drive capability is insufficient, the LCD may not be displayed clearly. Therefore, select an optimum boosting frequency for the LCD panel to be used.

Table 16-3 shows the V3 pin current capacity and boosting frequency.

Note: When used as the booster circuit, bias should be composed to 1/3. Therefore, do not set LCDCR<DUTY> to "10" or "11" when the booster circuit is enable (LCDCR<BRES>="1").



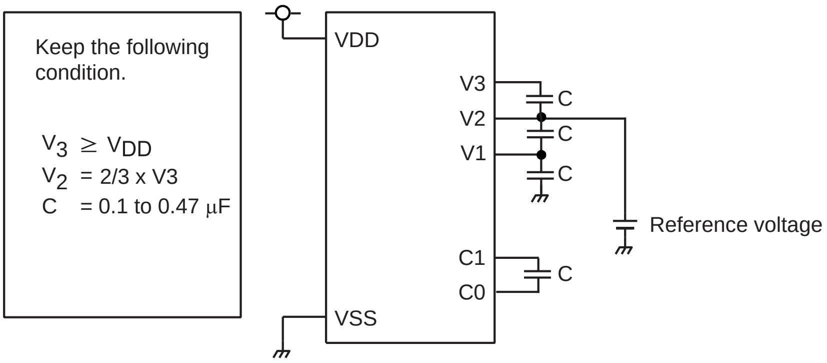
Keep the following condition.

$V_3 \geq V_{DD}$

V_1 = 1/3 x V3

C = 0.1 to 0.47 μF

a) Reference pin = V1



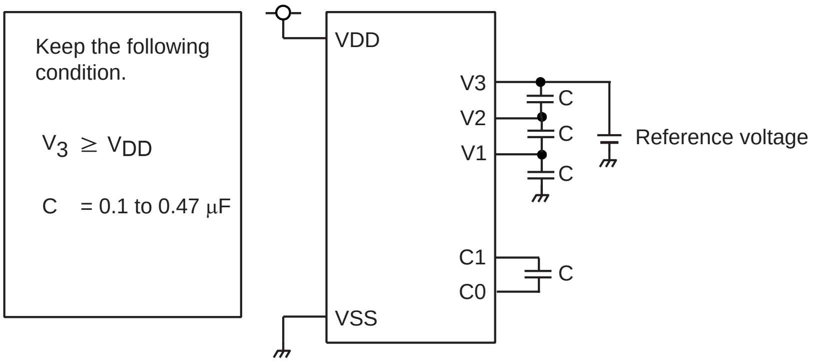
Keep the following condition.

$V_3 \geq V_{DD}$

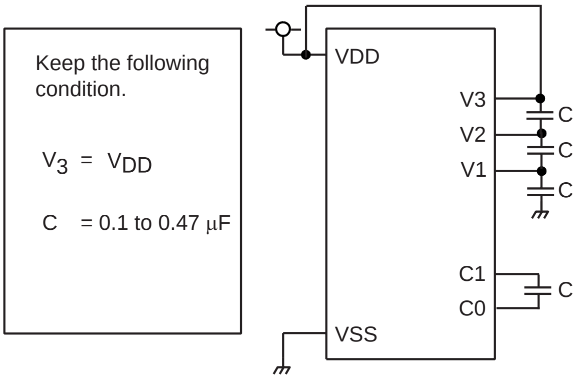
V_2 = 2/3 x V3

C = 0.1 to 0.47 μF

b) Reference pin = V2



c) Reference pin = V3



d) Reference pin = V3

Note 1: When the TMP86CS28FG uses the booster circuit to drive the LCD, the power supply and capacitor for the booster circuit should be connected as shown above.

Note 2: When the reference voltage is connected to a pin other than V1, add a capacitor between V1 and GND.

Note 3: The connection examples shown above are different from those shown in the datasheets of the previous version. Since the above connection method enhances the boosting characteristics, it is recommended that new boards be designed using the above connection method. (Using the existing connection method does not affect LCD display.)

Figure 16-3 Connection Examples When Using the Booster Circuit (LCDCR<BRES> = “1”)

Table 16-3 V3 Pin Current Capacity and Boosting Frequency (typ.)

VFSEL	Boosting frequency	fc = 16 MHz	fc = 8 MHz	fc = 4 MHz	fc = 32.768 MHz
00	fc/2^{13} or fs/2^5	−37 mV/ μA	−80 mV/ μA	−138 mV/ μA	−76 mV/ μA
01	fc/2^{11} or fs/2^3	−19 mV/ μA	−24 mV/ μA	−37 mV/ μA	−23 mV/ μA
10	fc/2^{10} or fs/2^2	−17 mV/ μA	−19 mV/ μA	−24 mV/ μA	−18 mV/ μA
11	fc/2^9	−16 mV/ μA	−17 mV/ μA	−19 mV/ μA	−

Note 1: The current capacity is the amount of voltage that falls per 1μA.

Note 2: The boosting frequency should be selected depending on your LCD panel.

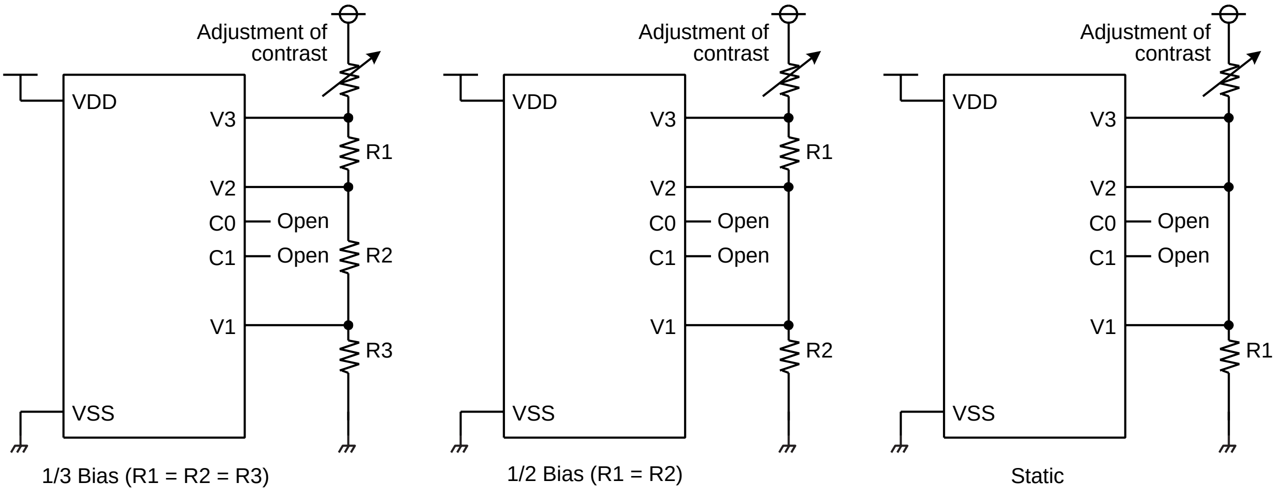
Note 3: For the reference pin V1 or V2, a current capacity ten times larger than the above is recommended to ensure stable operation.

For example, when the boosting frequency is fc/2^9 (at fc = 8 MHz), −1.7 mV/ μA or more is recommended for the current capacity of the reference pin V1.

16.2.3.2 When using an external resistor divider (LCDCR<BRES>="0")

When an external resistor divider is used, the voltage of an external power supply is divided and input on V1, V2, and V3 to generate the output voltages for segment/common signals.

The smaller the external resistor value, the higher the segment/common drive capability, but power consumption is increased. Conversely, the larger the external resistor value, the lower the segment/common drive capability, but power consumption is reduced. If the drive capability is insufficient, the LCD may not be displayed clearly. Therefore, select an optimum resistor value for the LCD panel to be used.



1/3 Bias (R1 = R2 = R3)

1/2 Bias (R1 = R2)

Static

Keep the following conditon.
$V_{DD} \geq V_3 \geq V_2 \geq V_1 \geq V_{SS}$

Figure 16-4 Connection Examples When Using an External Resistor Divider (LCDCR<BRES> = “0”)

16.3 LCD Display Operation

16.3.1 Display data setting

Display data is stored to the display data area (assigned to address 0FC0H to 0FD3H, 20bytes) in the DBR. The display data which are stored in the display data area is automatically read out and sent to the LCD driver by the hardware. The LCD driver generates the segment signal and common signal according to the display data and driving method. Therefore, display patterns can be changed by only over writing the contents of display data area by the program. Table 16-5 shows the correspondence between the display data area and SEG/COM pins.

LCD light when display data is “1” and turn off when “0”. According to the driving method of LCD, the number of pixels which can be driven becomes different, and the number of bits in the display data area which is used to store display data also becomes different.

Therefore, the bits which are not used to store display data as well as the data buffer which corresponds to the addresses not connected to LCD can be used to store general user process data (see Table 16-4).

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

Table 16-4 Driving Method and Bit for Display Data

Driving methods	Bit 7/3	Bit 6/2	Bit 5/1	Bit 4/0
1/4 Duty	COM3	COM2	COM1	COM0
1/3 Duty	–	COM2	COM1	COM0
1/2 Duty	–	–	COM1	COM0
Static	–	–	–	COM0

Note: –: This bit is not used for display data

Table 16-5 LCD Display Data Area (DBR)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0FC0H	SEG1				SEG0			
0FC1H	SEG3				SEG2			
0FC2H	SEG5				SEG4			
0FC3H	SEG7				SEG6			
0FC4H	SEG9				SEG8			
0FC5H	SEG11				SEG10			
0FC6H	SEG13				SEG12			
0FC7H	SEG15				SEG14			
0FC8H	SEG17				SEG16			
0FC9H	SEG19				SEG18			
0FCAH	SEG21				SEG20			
0FCBH	SEG23				SEG22			
0FCCH	SEG25				SEG24			
0FCDH	SEG27				SEG26			
0FCEH	SEG29				SEG28			
0FCFH	SEG31				SEG30			
0FD0H	SEG33				SEG32			
0FD1H	SEG35				SEG34			
0FD2H	SEG37				SEG36			
0FD3H	SEG39				SEG38			
	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0

16.3.2 Blanking

Blanking is enabled when EDSP is cleared to “0”.

Blanking turns off LCD through outputting a GND level to SEG/COM pin.

When in STOP mode, EDSP is cleared to “0” and automatically blanked. To redisplay ICD after exiting STOP mode, it is necessary to set EDSP back to “1”.

Note: During reset, the LCD common outputs are fixed “0” level. But the multiplex terminal of input/output port and LCD segment output becomes high impedance. Therefore, when the reset input is long remarkably, ghost problem may appear in LCD display.

16.4 Control Method of LCD Driver

16.4.1 Initial setting

Figure 16-5 shows the flowchart of initialization.

Example : To operate a 1/4 duty LCD of 40 segments × 4 com-mons at frame frequency fc/2^{16} [Hz], and booster frequency fc/2^{13} [Hz]

```
LD          (LCDCR), 01000001B       ; Sets LCD driving method and frame frequency. Boost frequency

LD          (P*LCR), 0FFH            ; Sets segment output control register. (*; Port No.)

 :            :

 :            :                      ; Sets the initial value of display data.

LD          (LCDCR), 11000001B       ; Display enable
```

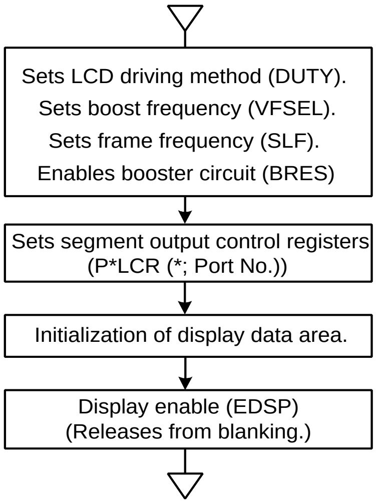

Sets LCD driving method (DUTY).
Sets boost frequency (VFSEL).
Sets frame frequency (SLF).
Enables booster circuit (BRES)
Sets segment output control registers (P*LCR (*; Port No.))
Initialization of display data area.
Display enable (EDSP) (Releases from blanking.)

Figure 16-5 Initial Setting of LCD Driver

16.4.2 Store of display data

Generally, display data are prepared as fixed data in program memory (ROM) and stored in display data area by load command.

Example :To display using 1/4 duty LCD a numerical value which corresponds to the LCD data stored in data memory at address 80H (when pins COM and SEG are connected to LCD as in Figure 16-6), display data become as shown in Table 16-6.

```
                LD          A, (80H)

                ADD         A, TABLE-$-7

                LD          HL, 0F80H

                LD          W, (PC + A)

                LD          (HL), W

                RET

TABLE:          DB          11011111B, 00000110B,
                            11100011B, 10100111B,
                            00110110B, 10110101B,
                            11110101B, 00010111B,
                            11110111B, 10110111B
```

Note: DB is a byte data difinition instruction.

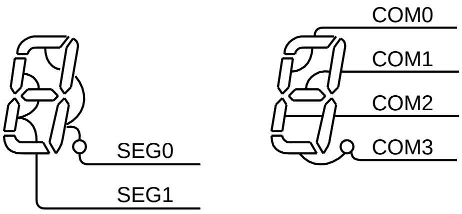
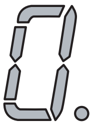
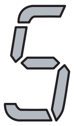
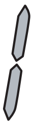
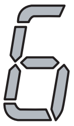
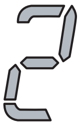
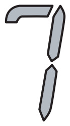
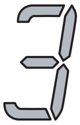
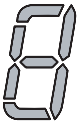
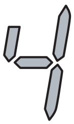
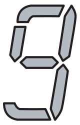


Figure 16-6 Example of COM, SEG Pin Connection (1/4 Duty)

Table 16-6 Example of Display Data (1/4 Duty)

No.	display	Display data	No.	display	Display data
0		11011111	5		10110101
1		00000110	6		11110101
2		11100011	7		00000111
3		10100111	8		11110111
4		00110110	9		10110111

Example 2: Table 16-6 shows an example of display data which are displayed using 1/2 duty LCD in the same way as Table 16-7. The connection between pins COM and SEG are the same as shown in Figure 16-7.

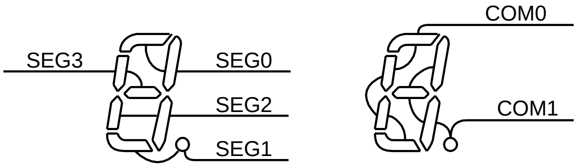


Figure 16-7 Example of COM, SEG Pin Connection

Table 16-7 Example of Display Data (1/2 Duty)

Number	Display data		Number	Display data	
	High order address	Low order address		High order address	Low order address
0	**01**11	**01**11	5	**11**10	**01**01
1	**00**10	**00**10	6	**11**11	**01**01
2	**10**01	**01**11	7	**01**10	**00**11
3	**10**10	**01**11	8	**11**11	**01**11
4	**11**10	**00**10	9	**11**10	**01**11

Note: *: Don’t care

16.4.3 Example of LCD drive output

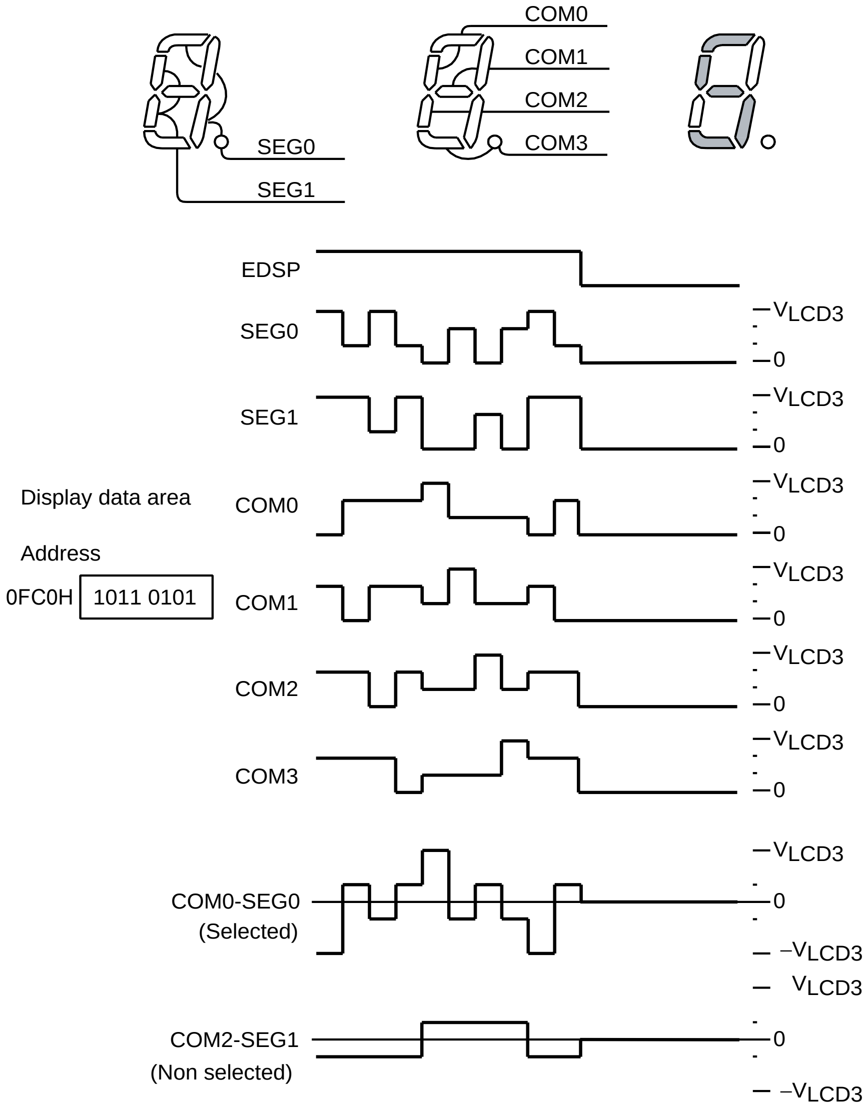
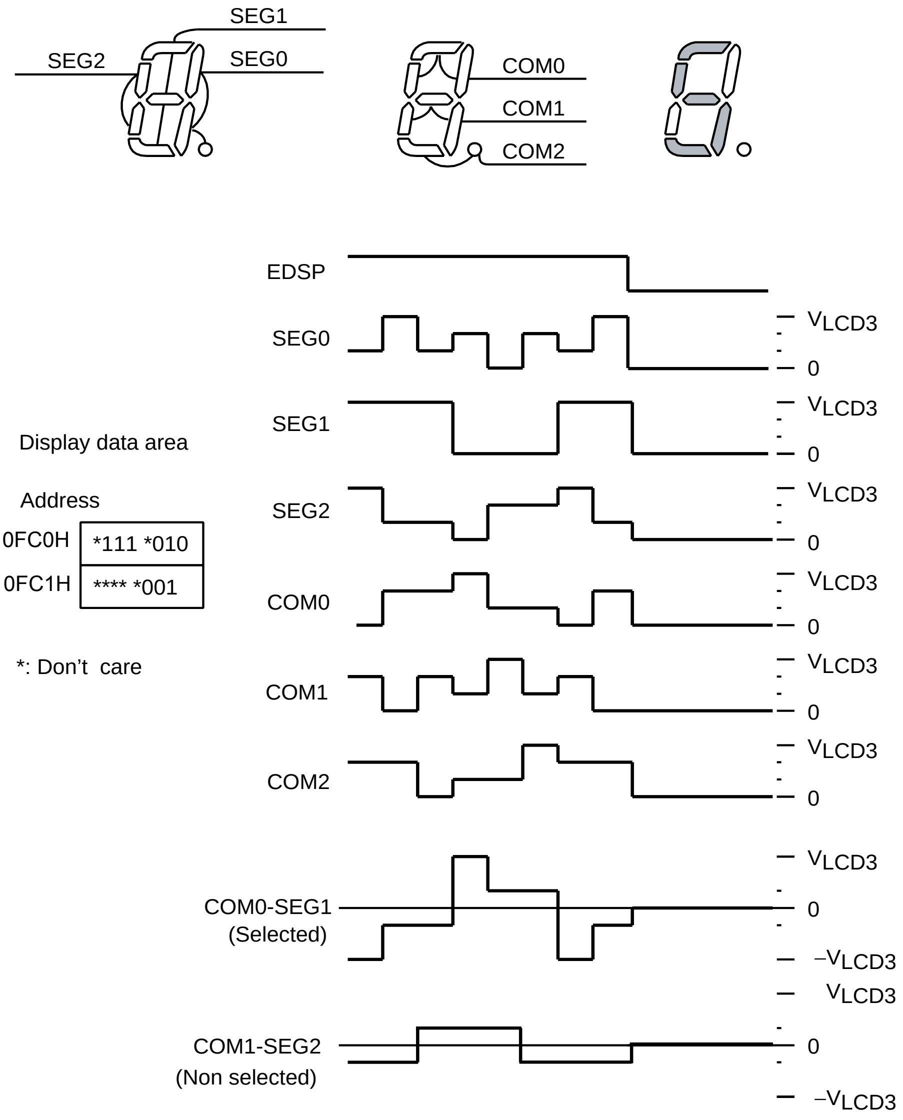


Figure 16-8 1/4 Duty (1/3 bias) Drive



Address	
0FC0H	*111 *010
0FC1H	**** *001

Figure 16-9 1/3 Duty (1/3 bias) Drive

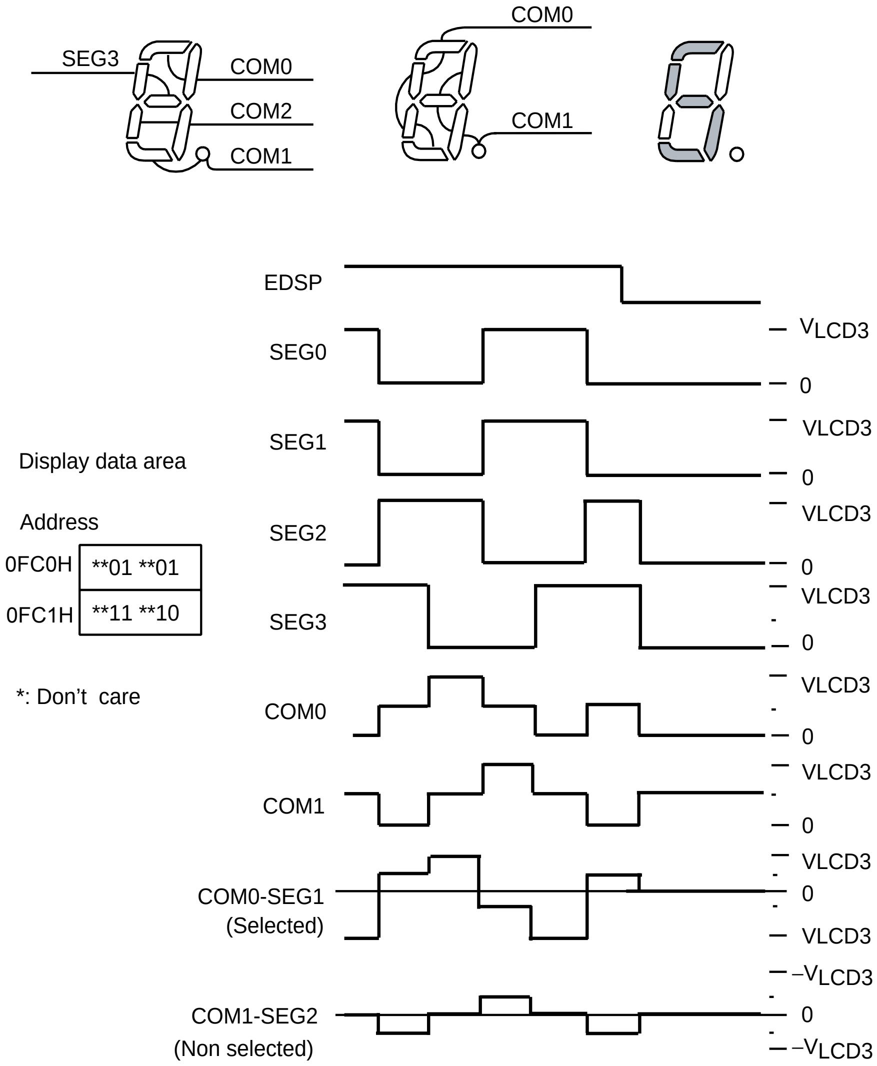
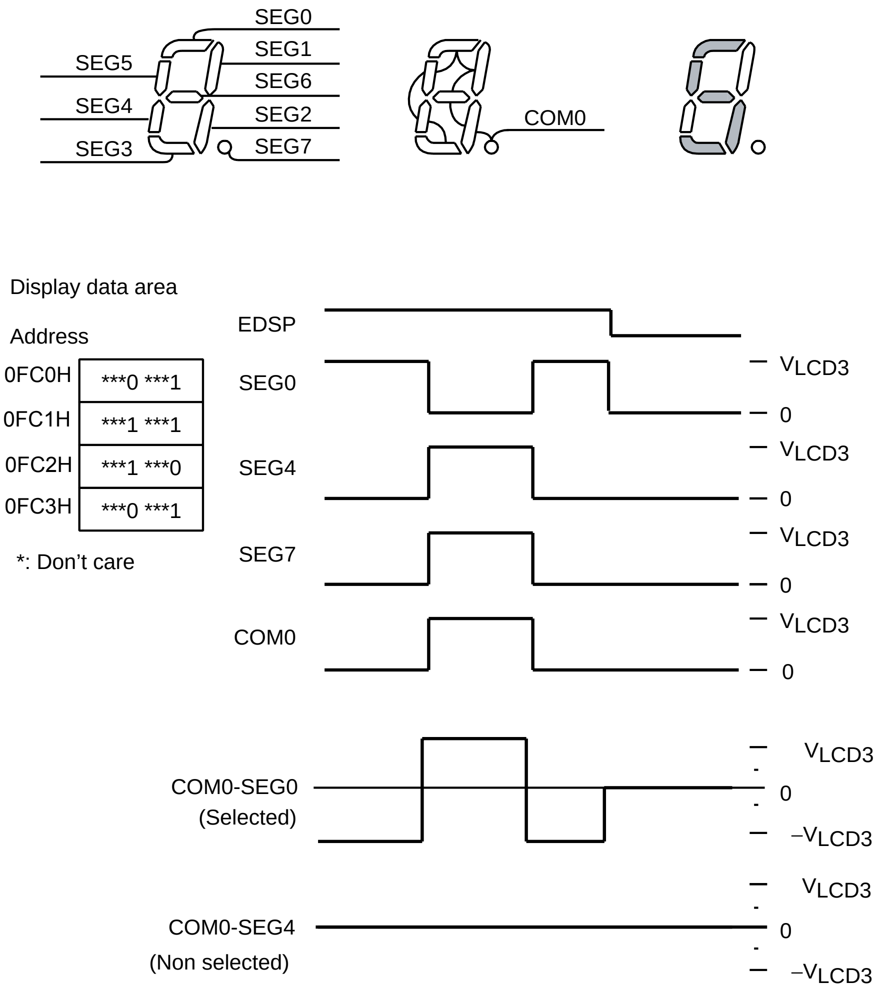


Figure 16-10 1/2 Duty (1/2 bias) Drive



Display data area

Address	
0FC0H	***0 ***1
0FC1H	***1 ***1
0FC2H	***1 ***0
0FC3H	***0 ***1

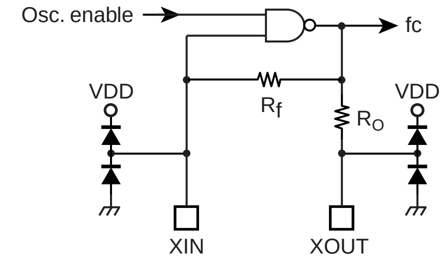
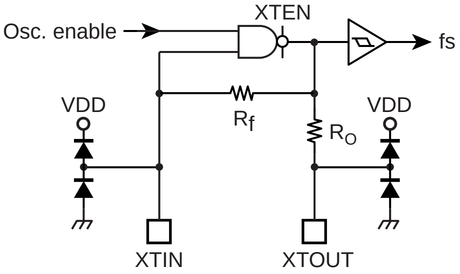
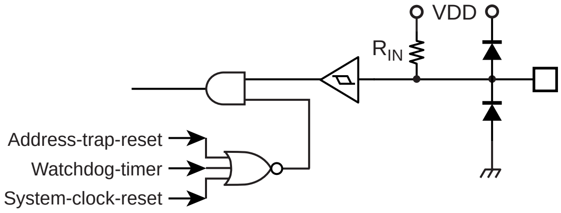
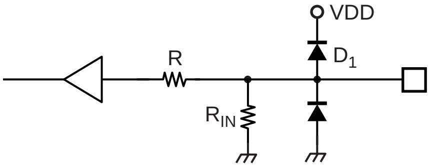
*: Don’t care

Figure 16-11 Static Drive

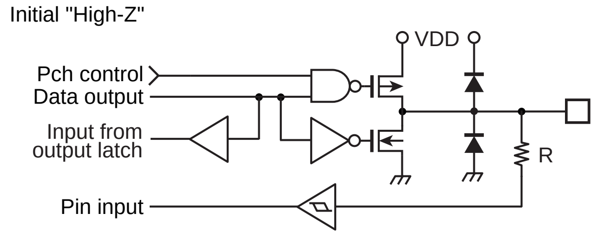
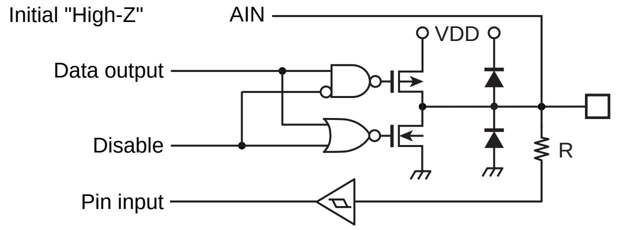
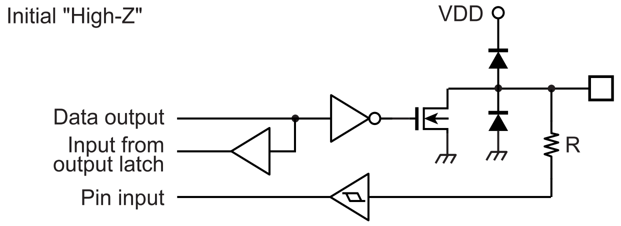
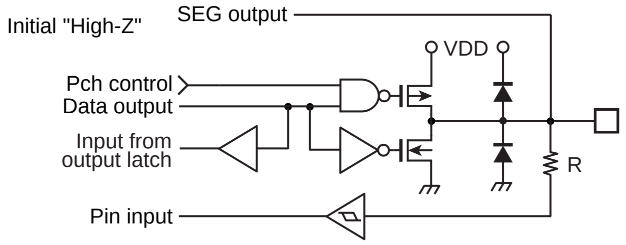
17. Input/Output Circuitry

17.1 Control Pins

The input/output circuitries of the TMP86CS28FG 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 MΩ (typ.) R_O = 0.5 kΩ (typ.)
XTIN XTOUT	Input Output	 XTEN Osc. enable fs VDD Rf RO VDD XTIN XTOUT	Resonator connecting pins (Low-frequency) R_f = 6 MΩ (typ.) R_O = 220 kΩ (typ.)
$\overline{\text{RESET}}$	Input	 VDD RIN Address-trap-reset Watchdog-timer System-clock-reset	Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.)
TEST	Input		Pull-down resistor R_{IN} = 70 kΩ (typ.) R = 1 kΩ (typ.)

17.2 Input/Output Ports

Port	I/O	Input/Output Circuitry	Remarks
P0,P3	Input Output	 Initial "High-Z" VDD Pch control Data output Input from output latch R Pin input	Sink open drain output or C-MOS output Hysteresis input R = 100 Ω (typ.)
P1	Input Output	 Initial "High-Z" AIN VDD Data output Disable R Pin input	Tri-state I/O Hysteresis input AIN input R = 100 W (typ.)
P2	Input Output	 Initial "High-Z" VDD Data output Input from output latch R Pin input	Sink open drain output Hysteresis input R = 100 Ω (typ.)
P4,P5,P6,P7,P8	Input Output	 Initial "High-Z" SEG output VDD Pch control Data output Input from output latch R Pin input	Sink open drain output or C-MOS output Hysteresis input R = 100 Ω (typ.) LCD segment output

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
Input voltage	V_{IN}		−0.3 to V_{DD} + 0.3	
Output voltage	V_{OUT}		−0.3 to V_{DD} + 0.3	
Output current (Per 1 pin)	I_{OL1}	P0,P1,P2,P3,P4,P5,P6,P7,P8 ports	3.2	mA
	I_{OH1}	P0,P1,P3,P4,P5,P6,P7,P8 ports	−1.8	
Output current (Total)	ΣI_{OL1}	P0,P1,P2,P3,P4,P5,P6,P7,P8 ports	80	
	ΣI_{OH1}	P0,P1,P3,P4,P5,P6,P7,P8 ports	−30	
Power dissipation [Topr = 85°C]	P_D		350	mW
Soldering temperature (Time)	Tsld		260 (10 s)	°C
Storage temperature	Tstg		−55 to 125	
Operating temperature	Topr		−40 to 85	

18.2 Operating Condition

The Operating Conditions show 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 V, Topr = –40 to 85°C)

Parameter	Symbol	Pins	Condition		Min	Max	Unit
Supply voltage	V_{DD}		fc = 16 MHz	NORMAL1, 2 mode	4.0	5.5	V
				IDLE0, 1, 2 mode			
			fc = 8 MHz	NORMAL1, 2 mode	2.7		
				IDLE0, 1, 2 mode			
			fs = 32.768 kHz	SLOW1, 2 mode			
				SLEEP0, 1, 2 mode			
				STOP mode			
Input high level	V_{IH1}	Except hysteresis input	$V_{DD} \geq 4.5$ V		$V_{DD} \times 0.70$	V_{DD}	
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
	V_{IH3}		$V_{DD} < 4.5$ V		$V_{DD} \times 0.90$		
Input low level	V_{IL1}	Except hysteresis input	$V_{DD} \geq 4.5$ V		0	$V_{DD} \times 0.30$	
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
	V_{IL3}		$V_{DD} < 4.5$ V			$V_{DD} \times 0.10$	
Clock frequency	fc	XIN, XOUT	V_{DD} = 2.7 V to 5.5 V		1.0	8.0	MHz
			V_{DD} = 4.0 V to 5.5 V			16.0	
	fs	XTIN, XTOUT	V_{DD} = 2.7 V to 5.5 V		30.0	34.0	kHz
LCD reference voltage range	V1		LCD booster circuit enable (V3 ≥ V_{DD})		0.9	1.8	V
Capacity for LCD booster circuit	C_{LCD}				0.1	0.47	uF

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}	Sink open drain, Tri-state	V_{DD} = 5.5 V, V_{IN} = 5.5 V/0 V	–	–	±2	μA
	I_{IN2}	$\overline{\text{RESET}}$, $\overline{\text{STOP}}$					
Input resistance	R_{IN1}	TEST pull-dwon		–	70	–	kΩ
	R_{IN2}	$\overline{\text{RESET}}$ pull-up		100	220	450	
Output leakage current	I_{LO}	Sink open drain, Tri-state	V_{DD} = 5.5 V, V_{OUT} = 5.5 V/0 V	–	–	±2	μA
Output high voltage	V_{OH}	C-MOS, Tri-st port	V_{DD} = 4.5 V, I_{OH} = −0.7 mA	4.1	–	–	V
Output low voltage	V_{OL}	Except XOUT	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	–	–	0.4	
LCD output voltage use LCD driver's booste	$V_{2\text{-}3OUT}$	V2 terminal	V3 ≥ V_{DD} Reference supply terminal :V1 SEG/COM terminal no load	–	V1 x 2	–	V
		V3 terminal		–	V1 x 3	–	
Supply current in NORMAL 1, 2 mode	I_{DD}		V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V fc = 16 MHz fs = 32.768 kHz	–	8.5	11.5	mA
Supply current in IDLE 0, 1, 2 mode				–	6	8.5	
Supply current in SLOW 1 mode			V_{DD} = 3.0 V V_{IN} = 2.8 V/0.2 V fs = 32.768 kHz	–	8.5	20	μA
Supply current in SLEEP 1 mode				–	6.1	15	
Supply current in SLEEP 0 mode				–	5	11	
Supply current in STOP mode			VDD = 5.5 V VIN = 5.3 V/0.2 V	–	0.5	10	

Note 1: Typical values show those at Topr = 25°C, V_{DD} = 5 V

Note 2: Input current (I_{IN1}, I_{IN2}); The current through pull-up or pull-down resistor is not included.

Note 3: I_{DD} does not include I_{REF} current.

Note 4: The supply currents of SLOW 2 and SLEEP 2 modes are equivalent to IDLE 0, 1, 2.

18.4 AD Conversion Characteristics

(V_{SS} = 0.0 V, 4.5 V ≤ V_{DD} ≤ 5.5 V, Topr = –40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} – 1.0	–	A_{VDD}	V
Power supply voltage of analog control circuit (Note6)	A_{VDD}		V_{DD}			
	A_{VSS}		V_{SS}			
Analog reference voltage range (Note4)	ΔV_{AREF}		3.5	–	–	
Analog input voltage	V_{AIN}		A_{VSS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 5.5 V V_{SS} = A_{VSS} = 0.0 V	–	0.6	1.0	mA
Non linearity error		V_{DD} = A_{VDD} = 5.0 V V_{SS} = A_{VSS} = 0.0 V V_{AREF} = 5.0 V	–	–	±2	LSB
Zero point error			–	–	±2	
Full scale error			–	–	±2	
Total error			–	–	±2	

(V_{SS} = 0.0 V, 2.7 V ≤ V_{DD} < 4.5 V, Topr = –40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} – 1.0	–	A_{VDD}	V
Power supply voltage of analog control circuit (Note6)	A_{VDD}		V_{DD}			
	A_{VSS}		V_{SS}			
Analog reference voltage range (Note4)	ΔV_{AREF}		2.5	–	–	
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 4.5 V V_{SS} = A_{VSS} = 0.0 V	–	0.5	0.8	mA
Non linearity error		V_{DD} = A_{VDD} = 2.7 V V_{SS} = A_{VSS} = 0.0 V V_{AREF} = 2.7 V	–	–	±2	LSB
Zero point error			–	–	±2	
Full scale error			–	–	±2	
Total error			–	–	±2	

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} to 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: $\Delta V_{AREF} = V_{AREF} - V_{SS}$

Note 5: The A_{VDD} pin should be fixed on the V_{DD} level even though AD converter is not used.

18.5 AC Characteristics

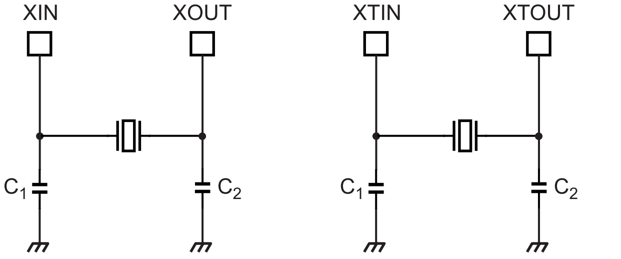
(V_{SS} = 0 V, V_{DD} = 4.0 to 5.5 V, Topr = –40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcy	NORMAL1, 2 mode	0.25	–	4	μs
		IDLE1, 2 mode				
		SLOW1, 2 mode	117.6	–	133.3	
		SLEEP1, 2 mode				
High level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 16 MHz	–	31.25	–	ns
Low level clock pulse width	t_{WCL}					
High level clock pulse width	t_{WCH}	For external clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	μs
Low level clock pulse width	t_{WCL}					

(V_{SS} = 0 V, V_{DD} = 2.7 to 5.5 V, Topr = –40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcy	NORMAL1, 2 mode	0.5	–	4	μs
		IDLE1, 2 mode				
		SLOW1, 2 mode	117.6	–	133.3	
		SLEEP1, 2 mode				
High level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 8 MHz	–	62.5	–	ns
Low level clock pulse width	t_{WCL}					
High level clock pulse width	t_{WCH}	For external clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	μs
Low level clock pulse width	t_{WCL}					

18.6 Recommended Oscillating Conditions



(1) High-frequency Oscillation (2) Low-frequency Oscillation

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

18.7 Handling Precaution

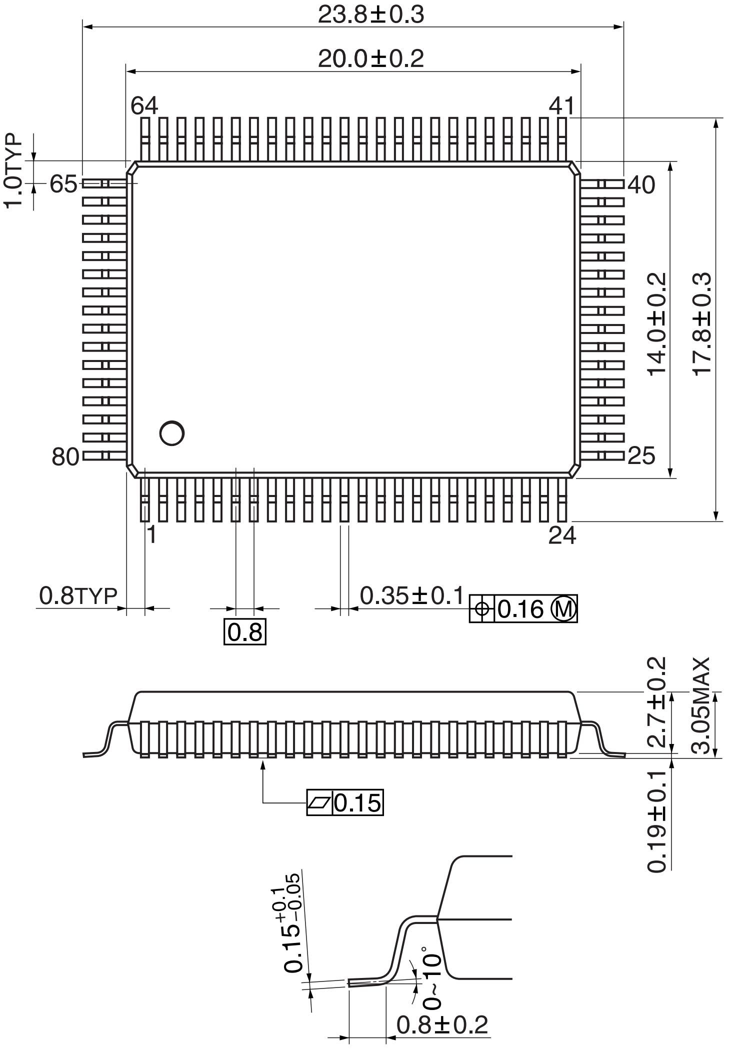
- The solderability test conditions for lead-free products (indicated by the suffix G in product name) are shown below.
  1. When using the Sn-37Pb solder bath
     Solder bath temperature = 230 °C
     Dipping time = 5 seconds
     Number of times = once
     R-type flux used
  2. When using the Sn-3.0Ag-0.5Cu solder bath
     Solder bath temperature = 245 °C
     Dipping time = 5 seconds
     Number of times = once
     R-type flux used

  Note: The pass criteron of the above test is as follows:
  Solderability rate until forming ≥ 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.

19. Package Dimensions

QFP80-P-1420-0.80B Rev 01

Unit: mm



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.