TOSHIBA

8 Bit Microcontroller
TLCS-870/C Series

TMP86PM23UG

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	
2006/8/24	1	First Release
2007/10/19	2	Contents Revised

Table of Contents

TMP86PM23UG

19. Input/Output Circuitry

20. Electrical Characteristics

21. Package Dimensions

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

CMOS 8-Bit Microcontroller

TMP86PM23UG

The TMP86PM23UG is a single-chip 8-bit high-speed and high-functionality microcomputer incorporating 32768 bytes of One-Time PROM. It is pin-compatible with the TMP86CM23AUG (Mask ROM version). The TMP86PM23UG can realize operations equivalent to those of the TMP86CM23AUG by programming the on-chip PROM.

Product No.	ROM (EPROM)	RAM	Package	MASK ROM MCU	Emulation Chip
TMP86PM23UG	32768 bytes	1536 bytes	LQFP64-P-1010-0.50E	TMP86CM23AUG	TMP86C923XB

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. 20interrupt sources (External : 5 Internal : 15)
3. Input / Output ports (I/O : 48 pins Output : 3 pins)

   Large current output: 5pins (Typ. 20mA), LED direct drive
4. Prescaler
   - Time base timer
   - Divider output function
5. Watchdog Timer
6. 18-bit Timer/Counter : 1ch
   - Timer Mode
   - Event Counter Mode
   - Pulse Width Measurement Mode
   - Frequency Measurement Mode

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 : 1 ch
9. 8-bit SIO: 1 ch
10. 10-bit successive approximation type AD converter
    - Analog input: 8 ch
11. Key-on wakeup : 4 ch
12. LCD driver/controller
    - LCD direct drive capability (MAX 32 seg × 4 com)
    - 1/4,1/3,1/2duties or static drive are programmably selectable
13. Multiply accumulate unit (MAC)
    - Multiply or MAC mode are selectable
    - Signed or unsigned operation are selectable
14. Clock operation

    Single clock mode

    Dual clock mode
15. Low power consumption operation

    STOP mode: Oscillation stops. (Battery/Capacitor back-up.)

    SLOW1 mode: Low power consumption operation using low-frequency clock.(High-frequency clock stop.)

    SLOW2 mode: Low power consumption operation using low-frequency clock.(High-frequency clock oscillate.)

    IDLE0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using high frequency clock. Release by falling edge of the source clock which is set by 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.
16. Wide operation voltage:

    3.5 V to 5.5 V at 16MHz /32.768 kHz

    2.7 V to 5.5 V at 8 MHz /32.768 kHz

    1.8 V to 5.5 V at 4.2MHz /32.768 kHz

1.2 Pin Assignment

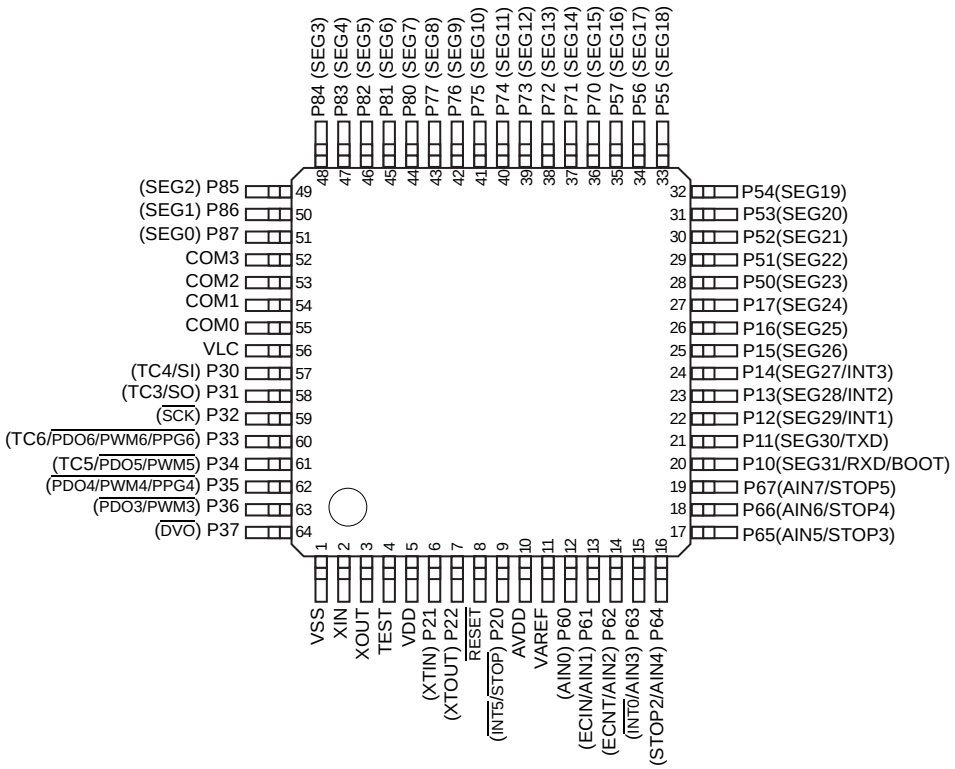


Figure 1-1 Pin Assignment

1.3 Block Diagram

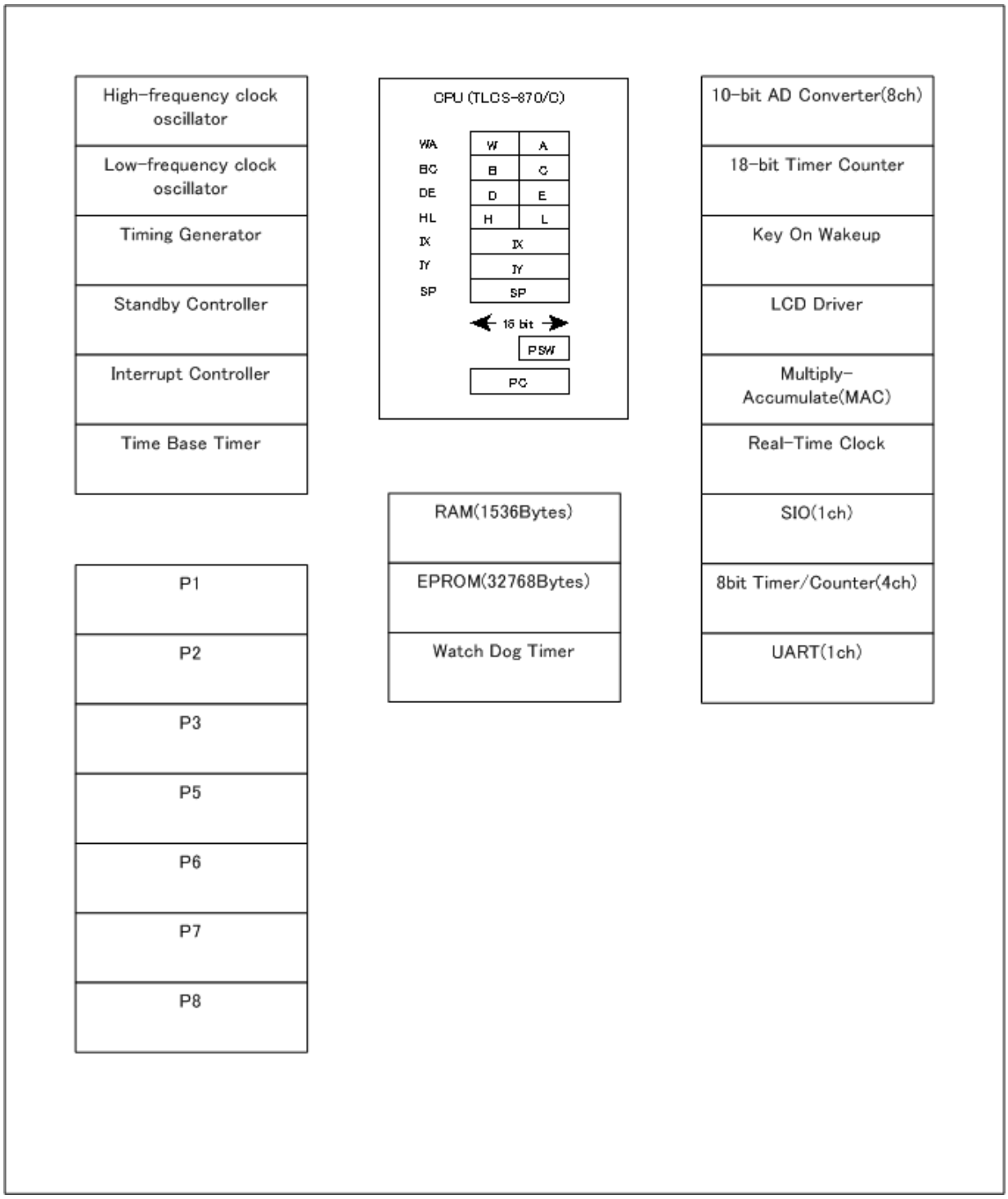


Figure 1-2 Block Diagram

1.4 Pin Names and Functions

The TMP86PM23UG has MCU mode and PROM mode. Table 1-1 shows the pin functions in MCU mode. The PROM mode is explained later in a separate chapter.

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

Pin Name	Pin Number	Input/Output	Functions
P17 SEG24	27	IO O	PORT17 LCD segment output 24
P16 SEG25	26	IO O	PORT16 LCD segment output 25
P15 SEG26	25	IO O	PORT15 LCD segment output 26
P14 SEG27 INT3	24	IO O I	PORT14 LCD segment output 27 External interrupt 3 input
P13 SEG28 INT2	23	IO O I	PORT13 LCD segment output 28 External interrupt 2 input
P12 SEG29 INT1	22	IO O I	PORT12 LCD segment output 29 External interrupt 1 input
P11 SEG30 TXD	21	IO O O	PORT11 LCD segment output 30 UART data output
P10 SEG31 RXD	20	IO O I	PORT10 LCD segment output 31 UART data input
P22 XTOUT	7	IO O	PORT22 Resonator connecting pins(32.768kHz) for inputting external clock
P21 XTIN	6	IO I	PORT21 Resonator connecting pins(32.768kHz) for inputting external clock
P20 $\overline{\text{STOP}}$ $\overline{\text{INT5}}$	9	IO I I	PORT20 STOP mode release signal input External interrupt 5 input
P37 $\overline{\text{DVO}}$	64	O O	PORT37 Divider Output
P36 $\overline{\text{PDO3/PWM3}}$	63	O O	PORT36 PDO3/PWM3 output
P35 $\overline{\text{PDO4/PWM4/PPG4}}$	62	O O	PORT35 PDO4/PWM4/PPG4 output
P34 $\overline{\text{PDO5/PWM5}}$ TC5	61	IO O I	PORT34 PDO5/PWM5 output TC5 input
P33 $\overline{\text{PDO6/PWM6/PPG6}}$ TC6	60	IO O I	PORT33 PDO6/PWM6/PPG6 output TC6 input
P32 $\overline{\text{SCK}}$	59	IO IO	PORT32 Serial Clock I/O

Table 1-1 Pin Names and Functions(2/3)

Pin Name	Pin Number	Input/Output	Functions
P31 SO TC3	58	IO O I	PORT31 Serial Data Output TC3 input
P30 SI TC4	57	IO I I	PORT30 Serial Data Input TC4 input
P57 SEG16	35	IO O	PORT57 LCD segment output 16
P56 SEG17	34	IO O	PORT56 LCD segment output 17
P55 SEG18	33	IO O	PORT55 LCD segment output 18
P54 SEG19	32	IO O	PORT54 LCD segment output 19
P53 SEG20	31	IO O	PORT53 LCD segment output 20
P52 SEG21	30	IO O	PORT52 LCD segment output 21
P51 SEG22	29	IO O	PORT51 LCD segment output 22
P50 SEG23	28	IO O	PORT50 LCD segment output 23
P67 AIN7 STOP5	19	IO I I	PORT67 Analog Input7 STOP5 input
P66 AIN6 STOP4	18	IO I I	PORT66 Analog Input6 STOP4 input
P65 AIN5 STOP3	17	IO I I	PORT65 Analog Input5 STOP3 input
P64 AIN4 STOP2	16	IO I I	PORT64 Analog Input4 STOP2 input
P63 AIN3 $\overline{\text{INT0}}$	15	IO I I	PORT63 Analog Input3 External interrupt 0 input
P62 AIN2 ECNT	14	IO I I	PORT62 Analog Input2 ECNT input
P61 AIN1 ECIN	13	IO I I	PORT61 Analog Input1 ECIN input
P60 AIN0	12	IO I	PORT60 Analog Input0
P77 SEG8	43	IO O	PORT77 LCD segment output 8
P76 SEG9	42	IO O	PORT76 LCD segment output 9

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

Pin Name	Pin Number	Input/Output	Functions
P75 SEG10	41	IO O	PORT75 LCD segment output 10
P74 SEG11	40	IO O	PORT74 LCD segment output 11
P73 SEG12	39	IO O	PORT73 LCD segment output 12
P72 SEG13	38	IO O	PORT72 LCD segment output 13
P71 SEG14	37	IO O	PORT71 LCD segment output 14
P70 SEG15	36	IO O	PORT70 LCD segment output 15
P87 SEG0	51	IO O	PORT87 LCD segment output 0
P86 SEG1	50	IO O	PORT86 LCD segment output 1
P85 SEG2	49	IO O	PORT85 LCD segment output 2
P84 SEG3	48	IO O	PORT84 LCD segment output 3
P83 SEG4	47	IO O	PORT83 LCD segment output 4
P82 SEG5	46	IO O	PORT82 LCD segment output 5
P81 SEG6	45	IO O	PORT81 LCD segment output 6
P80 SEG7	44	IO O	PORT80 LCD segment output 7
COM3	52	O	LCD common output 3
COM2	53	O	LCD common output 2
COM1	54	O	LCD common output 1
COM0	55	O	LCD common output 0
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	11	I	Analog Base Voltage Input Pin for A/D Conversion
AVDD	10	I	Analog Power Supply
VDD	5	I	+5V
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 TMP86PM23UG memory is composed OTP, RAM, DBR(Data buffer register) and SFR(Special function register). They are all mapped in 64-Kbyte address space. Figure 2-1 shows the TMP86PM23UG memory address map.

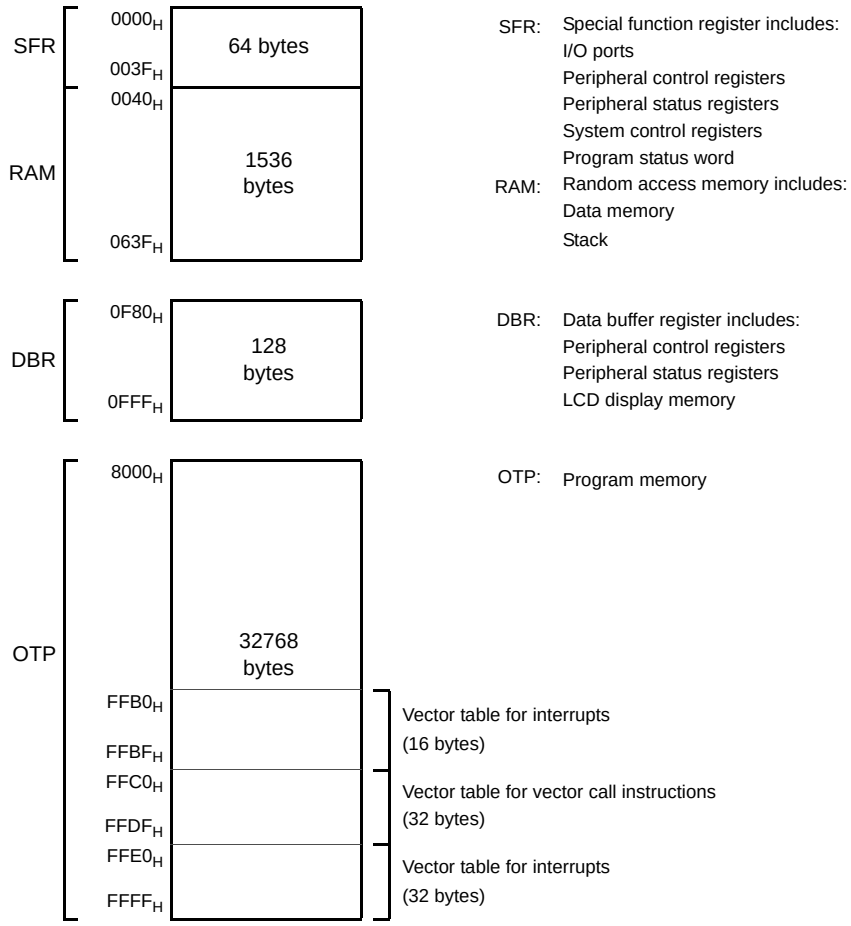


Figure 2-1 Memory Address Map

2.1.2 Program Memory (OTP)

The TMP86PM23UG has a 32768 bytes (Address 8000H to FFFFH) of program memory (OTP).

2.1.3 Data Memory (RAM)

The TMP86PM23UG has 1536bytes (Address 0040H to 063FH) 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”. (TMP86PM23UG)

```
        LD      HL, 0040H       ; Start address setup
        LD      A, H            ; Initial value (00H) setup
        LD      BC, 05FFH
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.

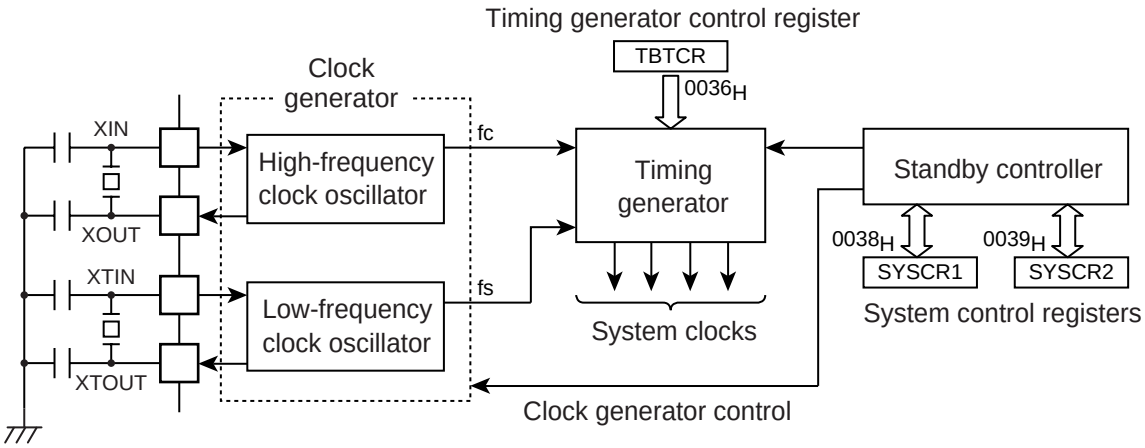


Figure 2-2 System Colck Control

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.

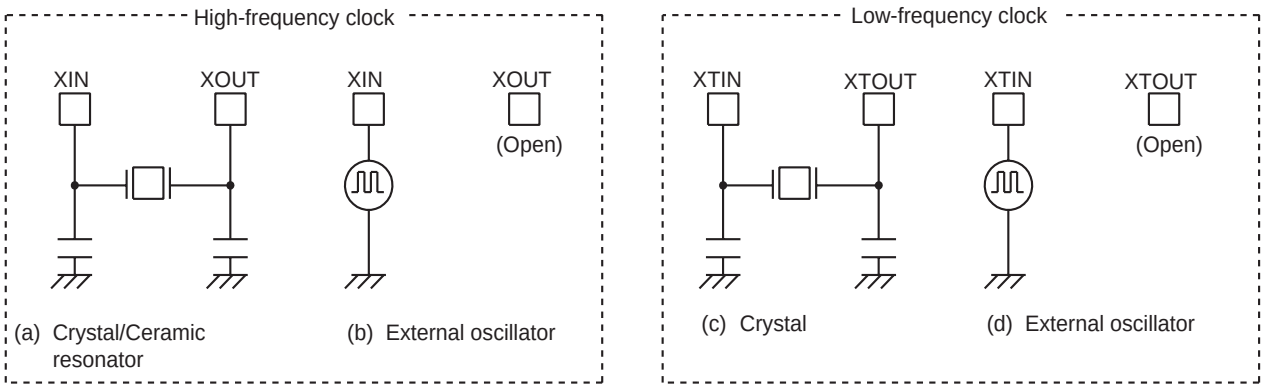


Figure 2-3 Examples of Resonator Connection

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 ($\overline{DVO}$) 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”.

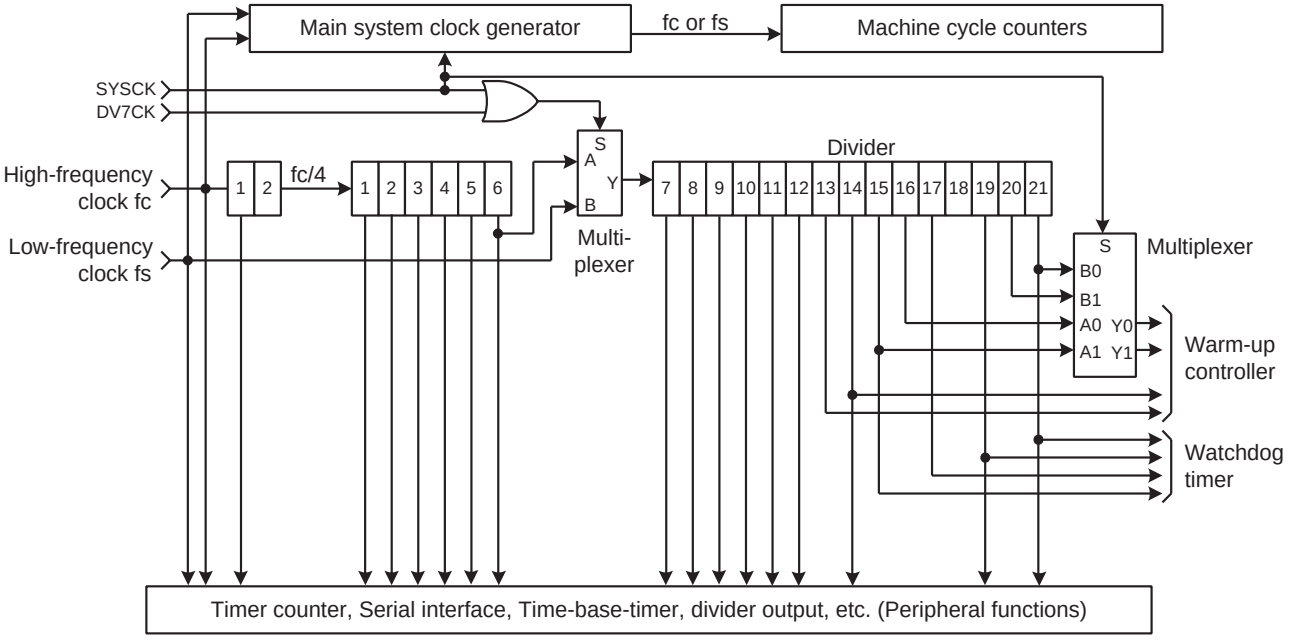


Figure 2-4 Configuration of Timing Generator

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.

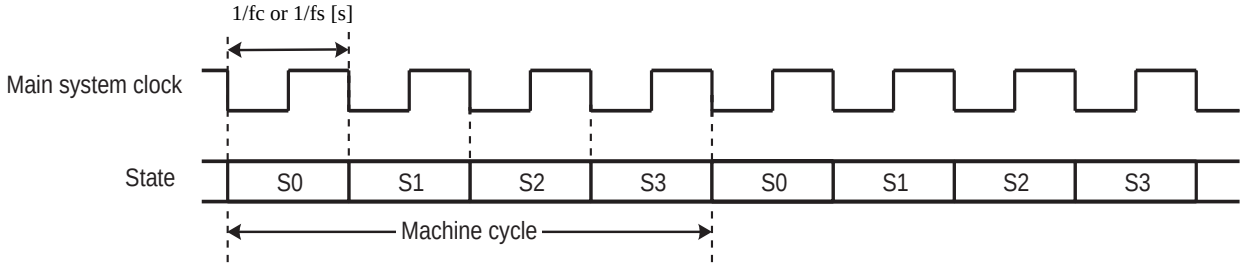


Figure 2-5 Machine Cycle

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 TMP86PM23UG 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>.

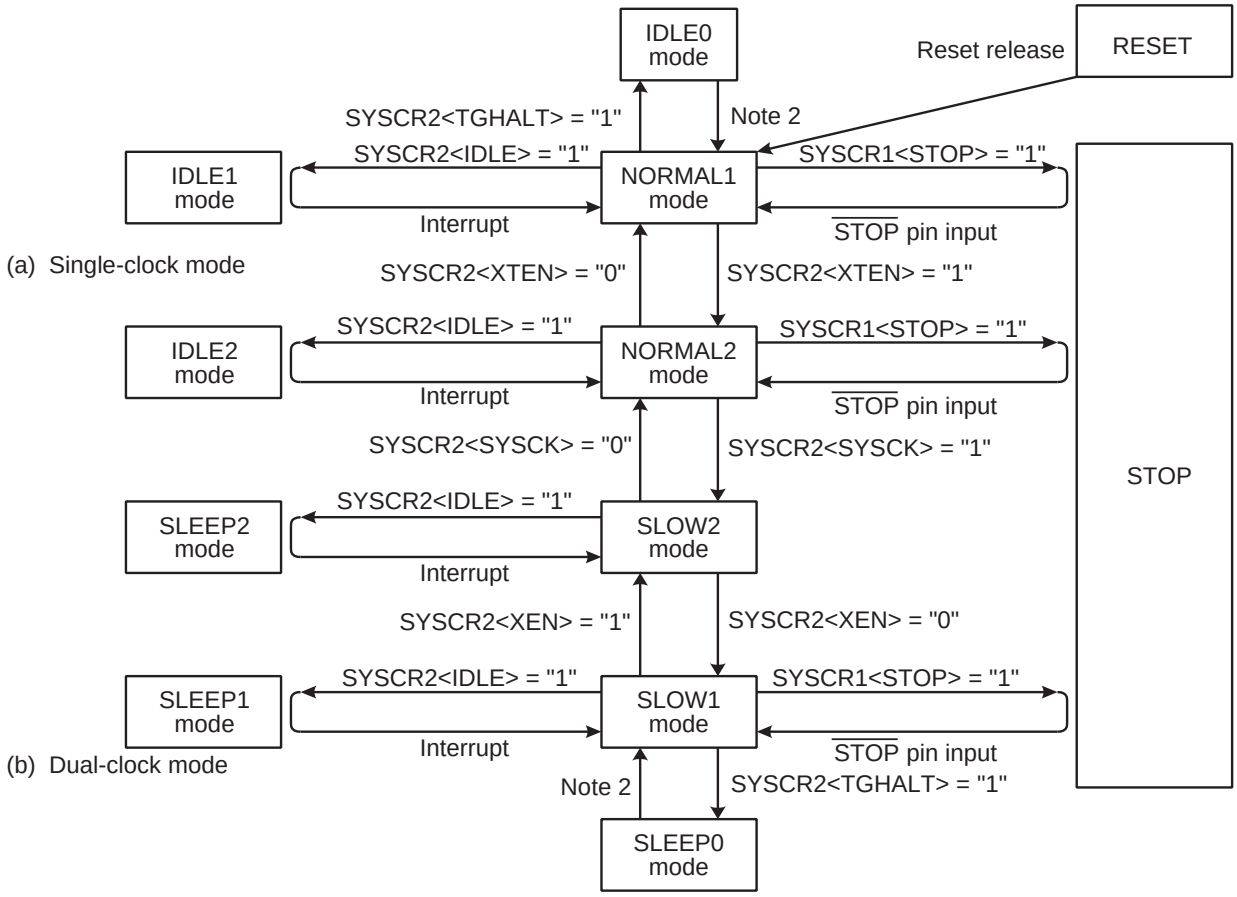
When SLEEP0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR<TBTCK>, the timing generator starts feeding the clock to all peripheral circuits.

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**)

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

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 $\overline{STOP}$ pin input and key-on wakeup input (STOP5 to STOP2) which is controlled by the STOP mode release control register (STOPCR).
The $\overline{STOP}$ pin is also used both as a port P20 and an $\overline{INT5}$ (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 $\overline{STOP}$ 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 $\overline{STOP}$ 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 $\overline{STOP}$ 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 $\overline{INT5}$ ($\overline{INT5}$ 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
```

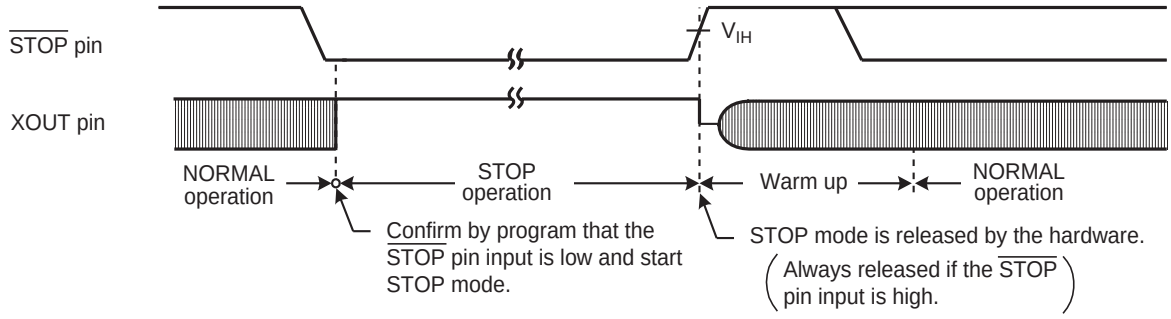


Figure 2-7 Level-sensitive Release Mode

Note 1: Even if the $\overline{\text{STOP}}$ pin input is low after warm-up start, the STOP mode is not restarted.

Note 2: In this case of changing to the level-sensitive mode from the edge-sensitive mode, the release mode is not switched until a rising edge of the $\overline{\text{STOP}}$ pin input is detected.

(2) Edge-sensitive release mode (RELM = “0”)

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

Example :Starting STOP mode from NORMAL mode

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

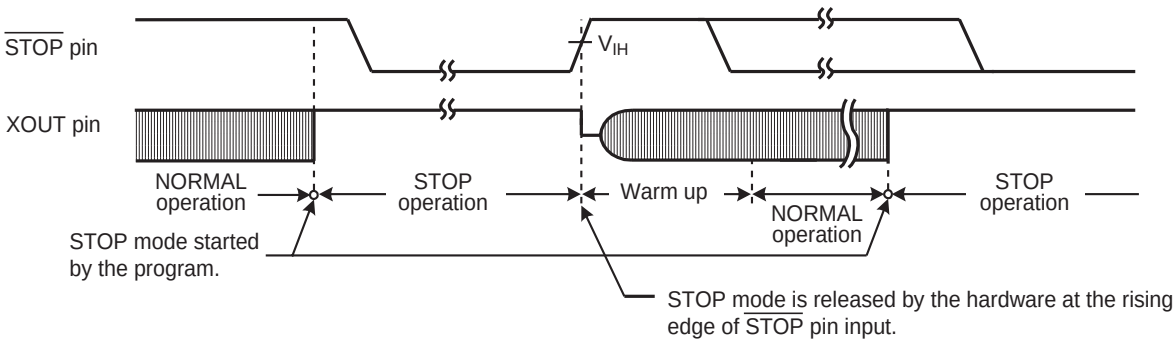


Figure 2-8 Edge-sensitive Release Mode

STOP mode is released by the following sequence.

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

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

Note 2: STOP mode can also be released by inputting low level on the $\overline{\text{RESET}}$ 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.

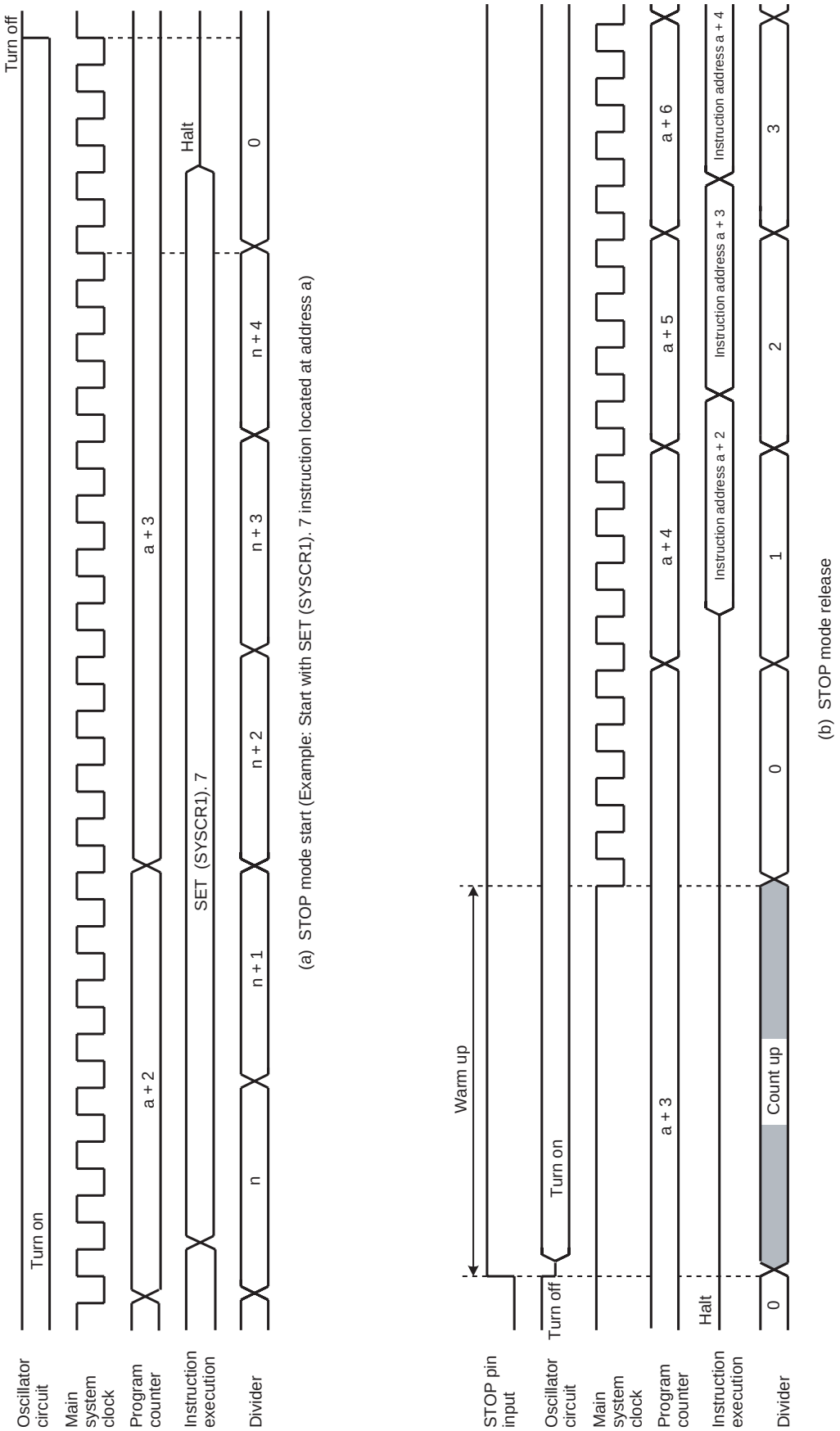


Figure 2-9 STOP Mode Start/Release

2.2.4.2 IDLE1/2 mode and SLEEP1/2 mode

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

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

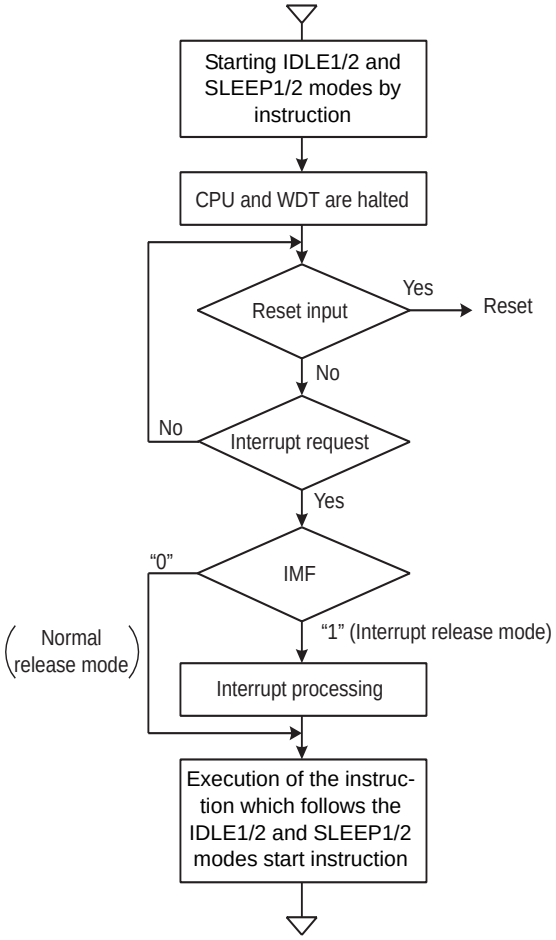


Figure 2-10 IDLE1/2 and SLEEP1/2 Modes

- Start the IDLE1/2 and SLEEP1/2 modes

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

- Release the IDLE1/2 and SLEEP1/2 modes

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

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

(1) Normal release mode (IMF = “0”)

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

(2) Interrupt release mode (IMF = “1”)

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

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

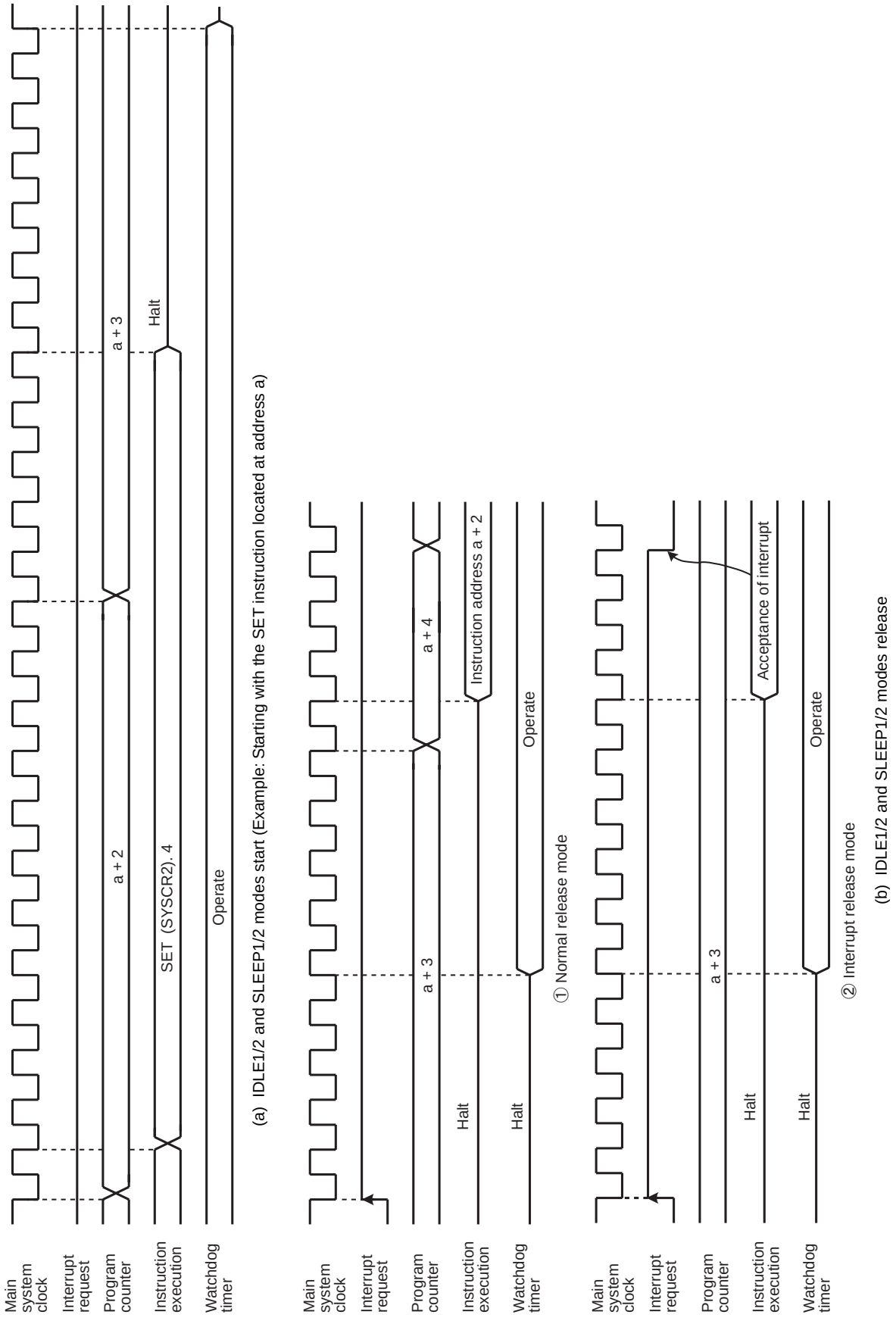


Figure 2-11 IDLE1/2 and SLEEP1/2 Modes Start/Release

2.2.4.3 IDLE0 and SLEEP0 modes (IDLE0, SLEEP0)

IDLE0 and SLEEP0 modes are controlled by the system control register 2 (SYSCR2) and the time base timer control register (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.

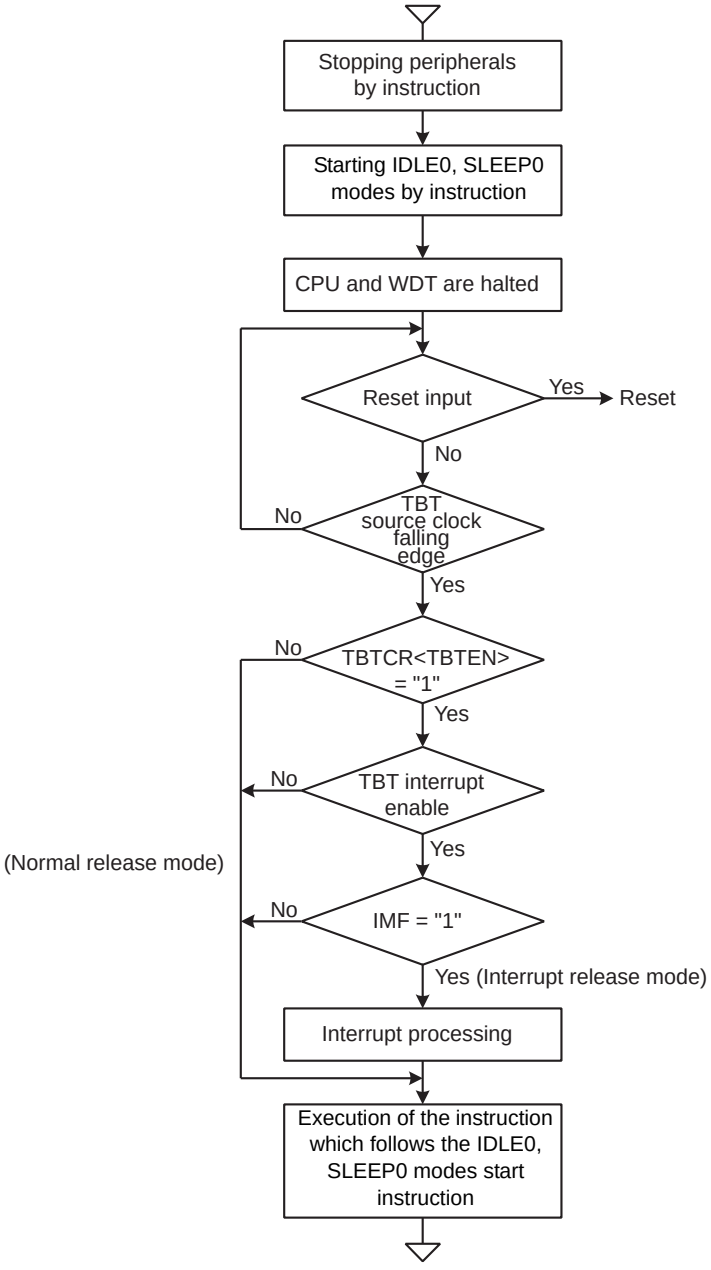


Figure 2-12 IDLE0 and SLEEP0 Modes

- Start the IDLE0 and SLEEP0 modes

  Stop (Disable) peripherals such as a timer counter.

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

- Release the IDLE0 and SLEEP0 modes

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

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

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

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

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

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

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

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

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

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

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

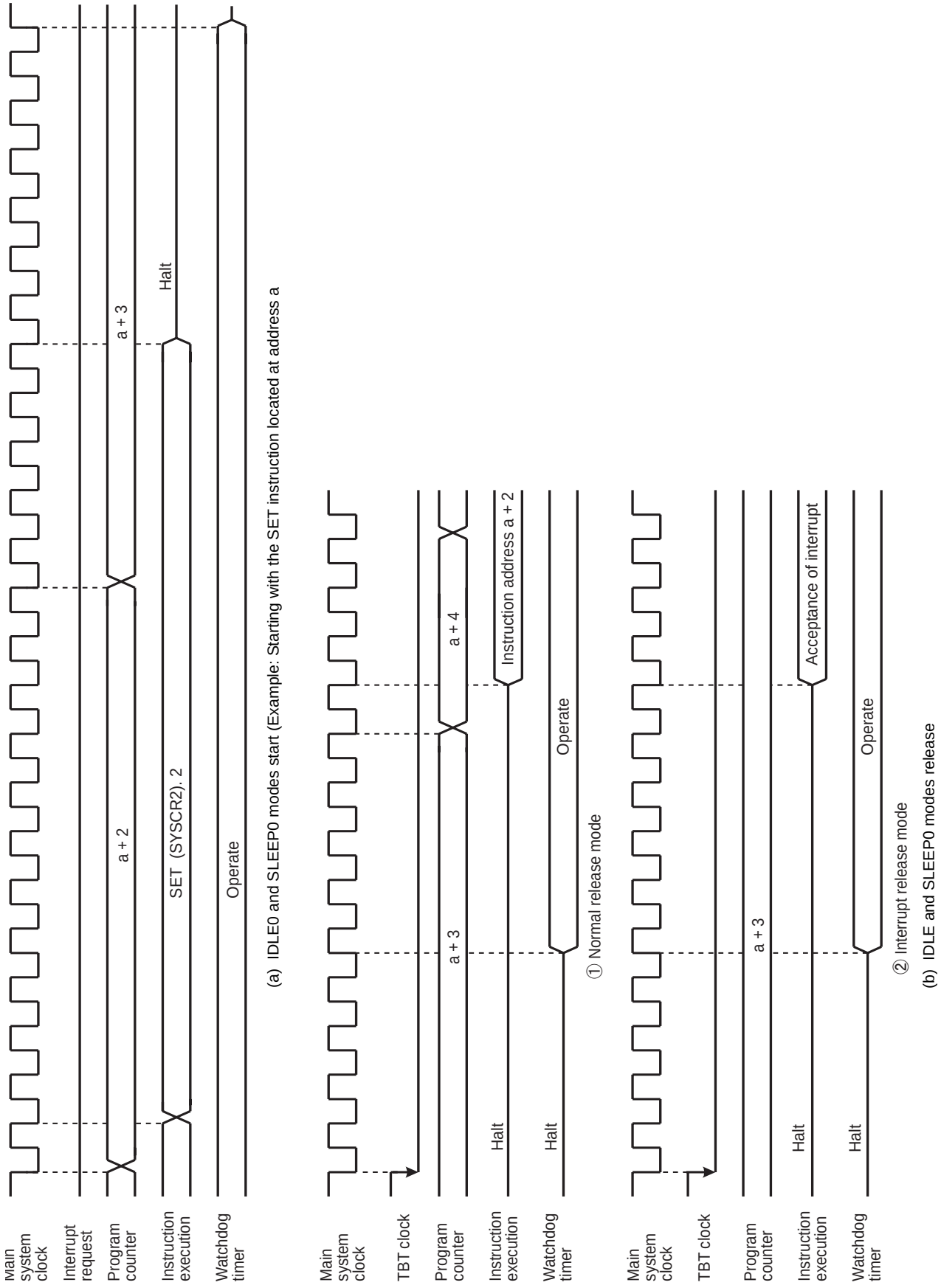


Figure 2-13 IDLE0 and SLEEP0 Modes Start/Release

2.2.4.4 SLOW mode

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

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

(1) Switching from NORMAL2 mode to SLOW1 mode

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

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

Example 1 :Switching from NORMAL2 mode to SLOW1 mode.

```
        SET     (SYSCR2). 5     ; SYSCR2<SYSCK> ← 1
                                  (Switches the main system clock to the low-frequency
                                  clock for SLOW2)
        CLR     (SYSCR2). 7     ; SYSCR2<XEN> ← 0
                                  (Turns off high-frequency oscillation)
```

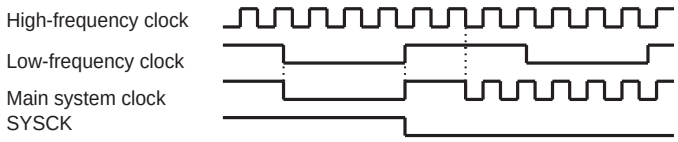
Example 2 :Switching to the SLOW1 mode after low-frequency clock has stabilized.

```
        SET     (SYSCR2). 6     ; SYSCR2<XTEN> ← 1
        LD      (TC3CR), 43H    ; Sets mode for TC4, 3 (16-bit mode, fs for source)
        LD      (TC4CR), 05H    ; Sets warming-up counter mode
        LDW     (TTREG3), 8000H ; Sets warm-up time (Depend on oscillator accompanied)
        DI                      ; IMF ← 0
        SET     (EIRH). 4       ; 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     (EIRH). 4           ; 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
```

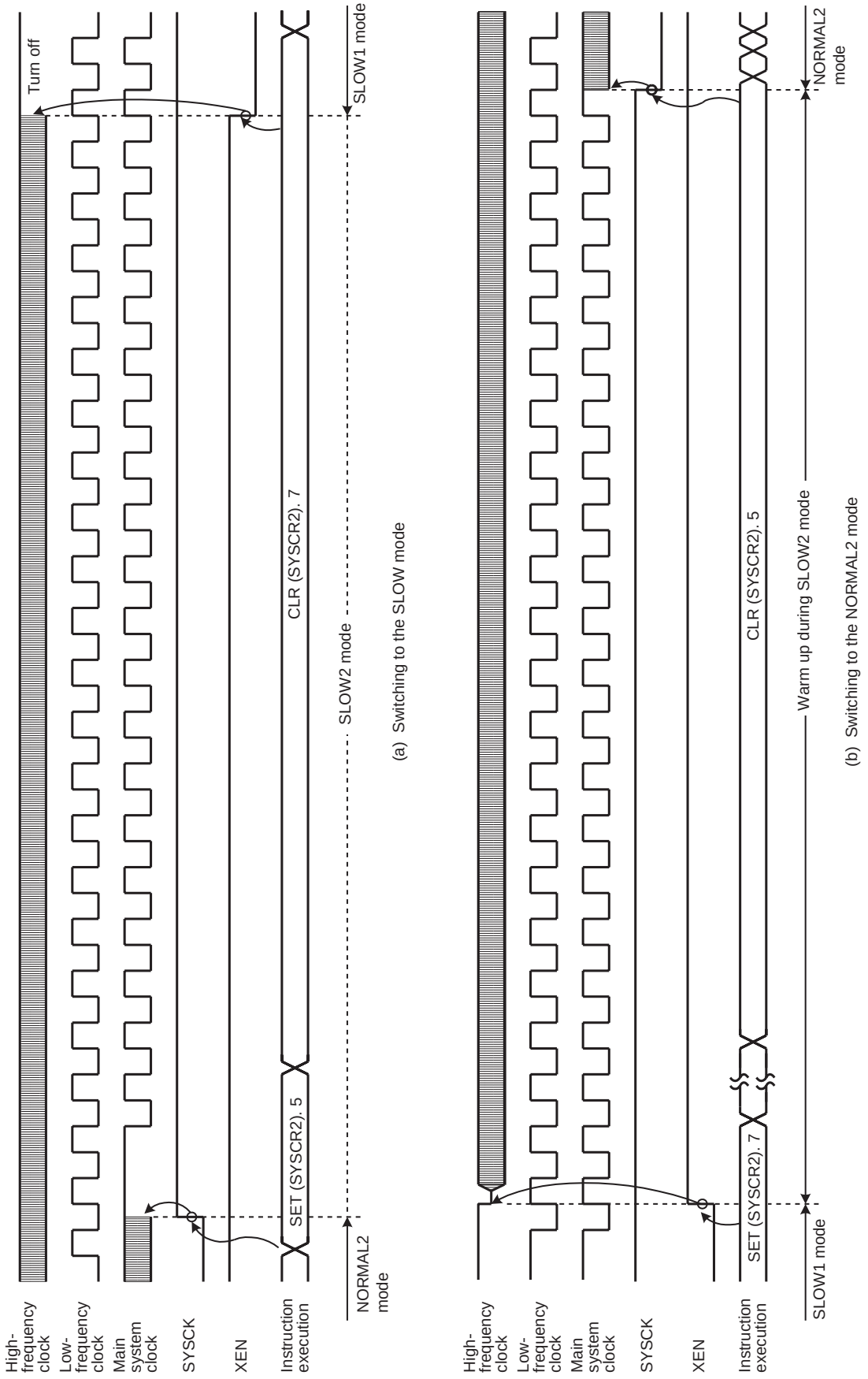


Figure 2-14 Switching between the NORMAL2 and SLOW Modes

2.3 Reset Circuit

The TMP86PM23UG 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.

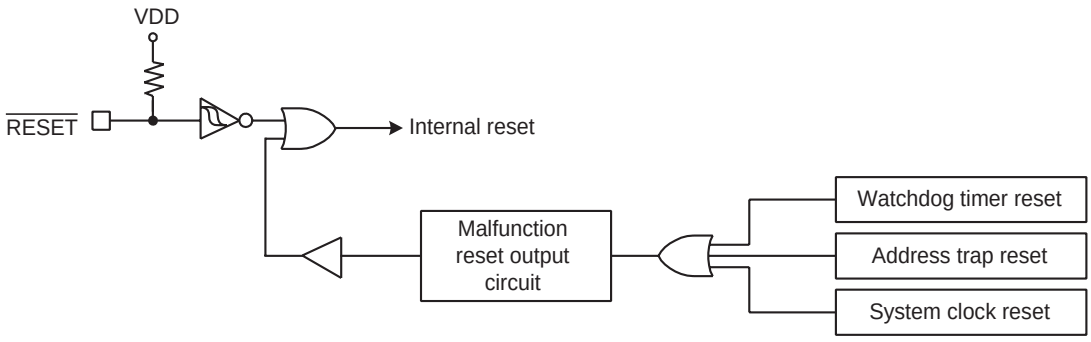
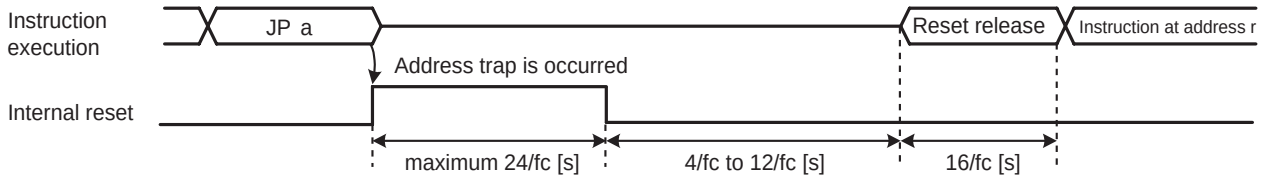


Figure 2-15 Reset Circuit

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 TMP86PM23UG has a total of 20 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	INTTC1	IMF• EF7 = 1	IL7	FFF0	8
Internal	INTSIO	IMF• EF8 = 1	IL8	FFEE	9
External	INT2	IMF• EF9 = 1	IL9	FFEC	10
Internal	INTRXD	IMF• EF10 = 1	IL10	FFEA	11
Internal	INTTXD	IMF• EF11 = 1	IL11	FFE8	12
Internal	INTTC4	IMF• EF12 = 1	IL12	FFE6	13
Internal	INTTC6	IMF• EF13 = 1	IL13	FFE4	14
Internal	INTRTC	IMF• EF14 = 1	IL14	FFE2	15
Internal	INTADC	IMF• EF15 = 1	IL15	FFE0	16
Internal	INTTC3	IMF• EF16 = 1	IL16	FFBE	17
External	INT3	IMF• EF17 = 1	IL17	FFBC	18
Internal	INTTC5	IMF• EF18 = 1	IL18	FFBA	19
External	$\overline{\text{INT5}}$	IMF• EF19 = 1	IL19	FFB8	20
-	Reserved	IMF• EF20 = 1	IL20	FFB6	21
-	Reserved	IMF• EF21 = 1	IL21	FFB4	22
-	Reserved	IMF• EF22 = 1	IL22	FFB2	23
-	Reserved	IMF• EF23 = 1	IL23	FFB0	24

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".

Note 3: If an INTADC interrupt request is generated while an interrupt with priority lower than the interrupt latch IL15 (INTADC) is being accepted, the INTADC interrupt latch may be cleared without the INTADC interrupt being processed. For details, refer to the corresponding notes in the chapter on the AD converter.

3.1 Interrupt latches (IL19 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, 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, 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 (EF19 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 (EF19 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: 00000000 000000**)

ILH,ILL (003DH, 003CH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IL15	IL14	IL13	IL12	IL11	IL10	IL9	IL8	IL7	IL6	IL5	IL4	IL3	IL2		

ILH (003DH) ILL (003CH)

(Initial value: ****0000)

ILE (002EH)

7	6	5	4	3	2	1	0
–	–	–	–	IL19	IL18	IL17	IL16

ILE (002EH)

IL19 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: 00000000 0000***0)

EIRH,EIRL (003BH, 003AH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EF15	EF14	EF13	EF12	EF11	EF10	EF9	EF8	EF7	EF6	EF5	EF4				IMF

EIRH (003BH) EIRL (003AH)

(Initial value: ****0000)

EIRE (002CH)

7	6	5	4	3	2	1	0
–	–	–	–	EF19	EF18	EF17	EF16

EIRE (002CH)

EF19 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".

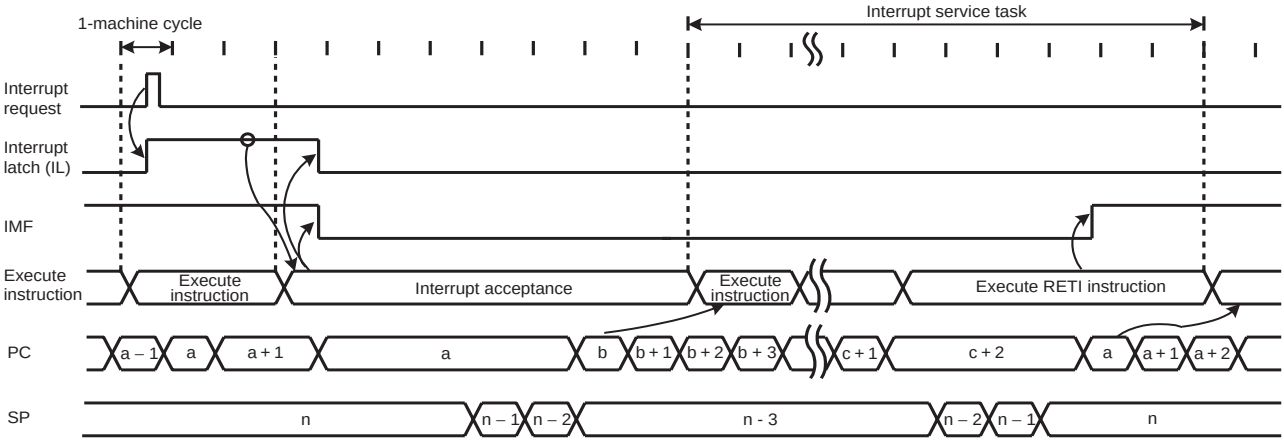
3.3 Interrupt Sequence

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

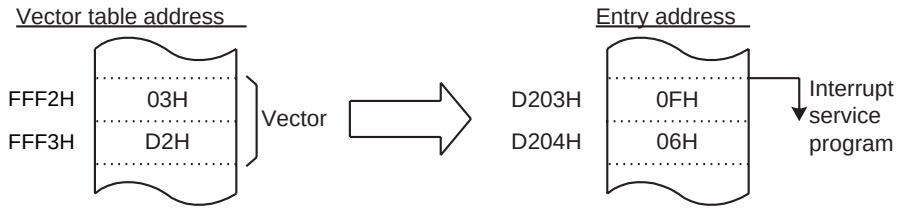


Figure 3-2 Vector table address,Entry address

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
```

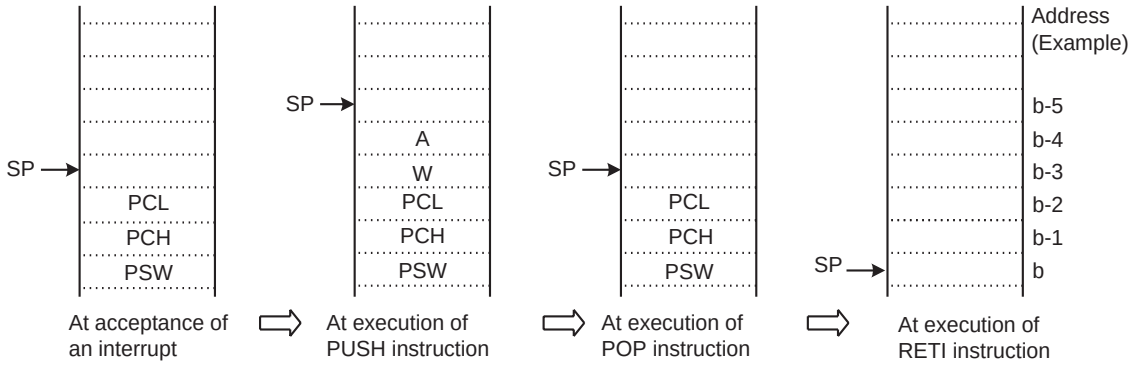


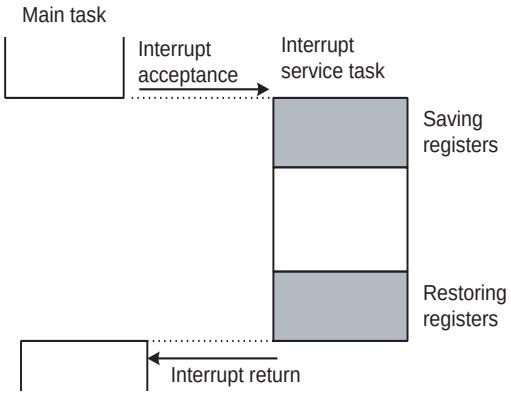
Figure 3-3 Save/store register using PUSH and POP instructions

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 TMP86PM23UG has 5 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 INT3. The $\overline{INT0}$/P63 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 $\overline{INT0}$/P63 pin function selection are performed by the external interrupt control register (EINTCR).

Source	Pin	Enable Conditions	Release Edge	Digital Noise Reject
INT0	$\overline{INT0}$	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 • EF9 = 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 • EF17 = 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.
INT5	$\overline{INT5}$	IMF • EF19 = 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 $\overline{INT0}$ 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	-	-	INT3ES	INT2ES	INT1ES		(Initial value: 00** 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	P63/$\overline{INT0}$ pin configuration	0: P63 input/output port 1: $\overline{INT0}$ pin (Port P63 should be set to an input mode)	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.

4. Special Function Register (SFR)

The TMP86PM23UG 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 0F80H to 0FFFH.

This chapter shows the arrangement of the special function register (SFR) and data buffer register (DBR) for TMP86PM23UG.

4.1 SFR

Address	Read	Write
0000H	Reserved	
0001H	P1DR	
0002H	P2DR	
0003H	P3DR	
0004H	P3OUTCR	
0005H	P5DR	
0006H	P6DR	
0007H	P7DR	
0008H	P8DR	
0009H	P1CR	
000AH	P5CR	
000BH	P6CR1	
000CH	P6CR2	
000DH	P7CR	
000EH	ADCCR1	
000FH	ADCCR2	
0010H	TREG1AL	
0011H	TREG1AM	
0012H	TREG1AH	
0013H	TREG1B	
0014H	TC1CR1	TC1CR
0015H	TC1CR2	
0016H	TC1SR	-
0017H	RTCCR	
0018H	TC3CR	
0019H	TC4CR	
001AH	TC5CR	
001BH	TC6CR	
001CH	TTREG3	
001DH	TTREG4	
001EH	TTREG5	
001FH	TTREG6	
0020H	ADCDR2	-
0021H	ADCDR1	-
0022H	Reserved	
0023H	Reserved	
0024H	P8CR	
0025H	UARTSR	UARTCR1

Address	Read	Write
0026H	-	UARTCR2
0027H	LCDCR	
0028H	PWREG3	
0029H	PWREG4	
002AH	PWREG5	
002BH	PWREG6	
002CH	EIRE	
002DH	Reserved	
002EH	ILE	
002FH	Reserved	
0030H	Reserved	
0031H	Reserved	
0032H	Reserved	
0033H	Reserved	
0034H	-	WDTCR1
0035H	-	WDTCR2
0036H	TBTCR	
0037H	EINTCR	
0038H	SYSCR1	
0039H	SYSCR2	
003AH	EIRL	
003BH	EIRH	
003CH	ILL	
003DH	ILH	
003EH	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
0F80H	SEG1/0	
0F81H	SEG3/2	
0F82H	SEG5/4	
0F83H	SEG7/6	
0F84H	SEG9/8	
0F85H	SEG11/10	
0F86H	SEG13/12	
0F87H	SEG15/14	
0F88H	SEG17/16	
0F89H	SEG19/18	
0F8AH	SEG21/20	
0F8BH	SEG23/22	
0F8CH	SEG25/24	
0F8DH	SEG27/26	
0F8EH	SEG29/28	
0F8FH	SEG31/30	
0F90H	SIOBR0	
0F91H	SIOBR1	
0F92H	SIOBR2	
0F93H	SIOBR3	
0F94H	SIOBR4	
0F95H	SIOBR5	
0F96H	SIOBR6	
0F97H	SIOBR7	
0F98H	-	SIOCR1
0F99H	SIOSR	SIOCR2
0F9AH	-	STOPCR
0F9BH	RDBUF	TDBUF
0F9CH	P2PRD	-
0F9DH	P3PRD	-
0F9EH	P1LCR	
0F9FH	P5LCR	

Address	Read	Write
0FA0H	P7LCR	
0FA1H	P8LCR	
0FA2H	Reserved	
0FA3H	Reserved	
0FA4H	MACCR	
0FA5H	MACSR	-
0FA6H	MPLDRL	
0FA7H	MPLDRH	
0FA8H	MPCDRL	
0FA9H	MPCDRH	
0FAAH	RCALDR1	MADDR1
0FABH	RCALDR2	MADDR2
0FACH	RCALDR3	MADDR3
0FADH	RCALDR4	MADDR4
0FAEH	Reserved	
0FAFH	Reserved	
0FB0H	Reserved	
0FB1H	Reserved	
0FB2H	Reserved	
0FB3H	Reserved	
0FB4H	Reserved	
0FB5H	Reserved	
0FB6H	Reserved	
0FB7H	Reserved	
0FB8H	Reserved	
0FB9H	Reserved	
0FBAH	Reserved	
0FBBH	Reserved	
0FBCH	Reserved	
0FBDH	Reserved	
0FBEH	Reserved	
0FBFH	Reserved	

Address	Read	Write
0FC0H	Reserved	
: :	: :	
0FDFH	Reserved	

Address	Read	Write
0FE0H	Reserved	
: :	: :	
0FFFH	Reserved	

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

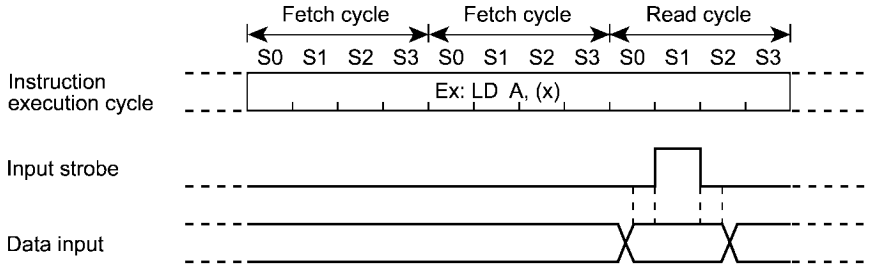
The TMP86PM23UG has 7 parallel input/output ports (48 pins) and output ports (3 pins) as follows.

	Primary Function	Secondary Functions
Port P1	8-bit I/O port	External interrupt input, UART input/output, and segment output.
Port P2	3-bit I/O port	Low-frequency resonator connections, external interrupt input, STOP mode release signal input.
Port P3	5-bit I/O port	Timer/counter input/output serial interface input/output and divider output.
	3-bit I/O port	Timer/counter input/output.
Port P5	8-bit I/O port	LCD segment output.
Port P6	8-bit I/O port	Analog input, external interrupt input, timer/counter input and STOP mode release signal input.
Port P7	8-bit I/O port	LCD segment output.
Port P8	8-bit I/O port	LCD segment output.

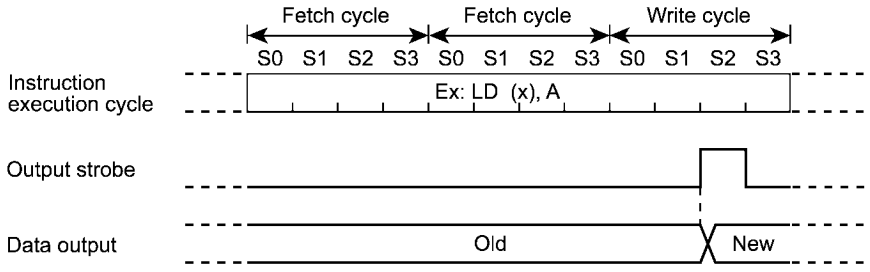
Each output port contains a latch, which holds the output data. All input ports do not have latches, so the external input data should be externally held until the input data is read from outside or reading should be performed several timer before processing. Figure 5-1 shows input/output timing examples.

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

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



(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 Timing (Example)

5.1 Port P1 (P17 to P10)

Port P1 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P1 is also used as a UART input/output, an external interrupt input and segment output of LCD. Input/output mode is specified by the P1 control register (P1CR).

When used as an input port or a secondary function input pins (UART input or external interrupt input), the corresponding bit of P1CR and P1LCR should be cleared to "0".

When used as an output port, the corresponding bit of P1CR should be set to "1", and the respective P1LCR bit should be cleared to "0". When used as an UART output pin, the corresponding bit of P1CR and the output latch (P1DR) should be set to "1", and the respective P1LCR bit should be cleared to "0".

When used as a segment pins of LCD, the respective bit of P1LCR should be set to "1".

During reset, the P1DR, P1CR and P1LCR are initialized to "0".

When the bit of P1CR and P1LCR is "0", the corresponding bit data by read instruction is a terminal input data.

When the bit of P1CR is "0" and that of P1LCR is "1", the corresponding bit data by read instruction is always "0".

When the bit of P1CR is "1", the corresponding bit data by read instruction is the value of P1DR.

Table 5-1 Register Programming for Multi-function Ports

Function	Programmed Value		
	P1DR	P1CR	P1LCR
Port input, UART input, and external interrupt input	*	"0"	"0"
Port "0" output	"0"	"1"	"0"
Port "1" output and UART output	"1"	"1"	"0"
LCD segment output	*	*	"1"

Note: Asterisk (*) indicates "1" or "0" either of which can be selected.

Table 5-2 Values Read from P1DR and Register Programming

Conditions		Values Read from P1DR
P1CR	P1LCR	
"0"	"0"	Terminal input data
"0"	"1"	"0"
"1"	"0"	Output latch contents
	"1"	

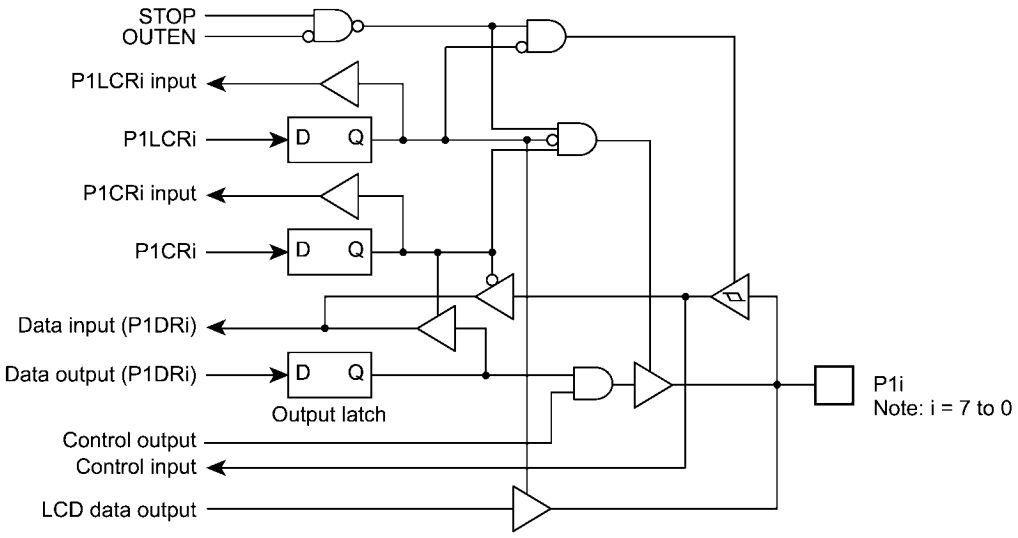


Figure 5-2 Port P1

P1DR (0001H) R/W

7	6	5	4	3	2	1	0
P17 SEG24	P16 SEG25	P15 SEG26	P14 SEG27 INT3	P13 SEG28 INT2	P12 SEG29 INT1	P11 SEG30 TXD	P10 SEG31 RXD

(Initial value: 0000 0000)

P1LCR (0F9EH)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

P1LCR	Port P1/segment output control (set for each bit individually)	0: P1 input/output port or secondary function (expect for segment) 1: LCD segment output	R/W

P1CR (0009H)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

P1CR	P1 port input/output control (set for each bit individually)	0: Input mode 1: Output mode	R/W

Note: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.

5.2 Port P2 (P22 to P20)

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

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

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

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

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

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

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

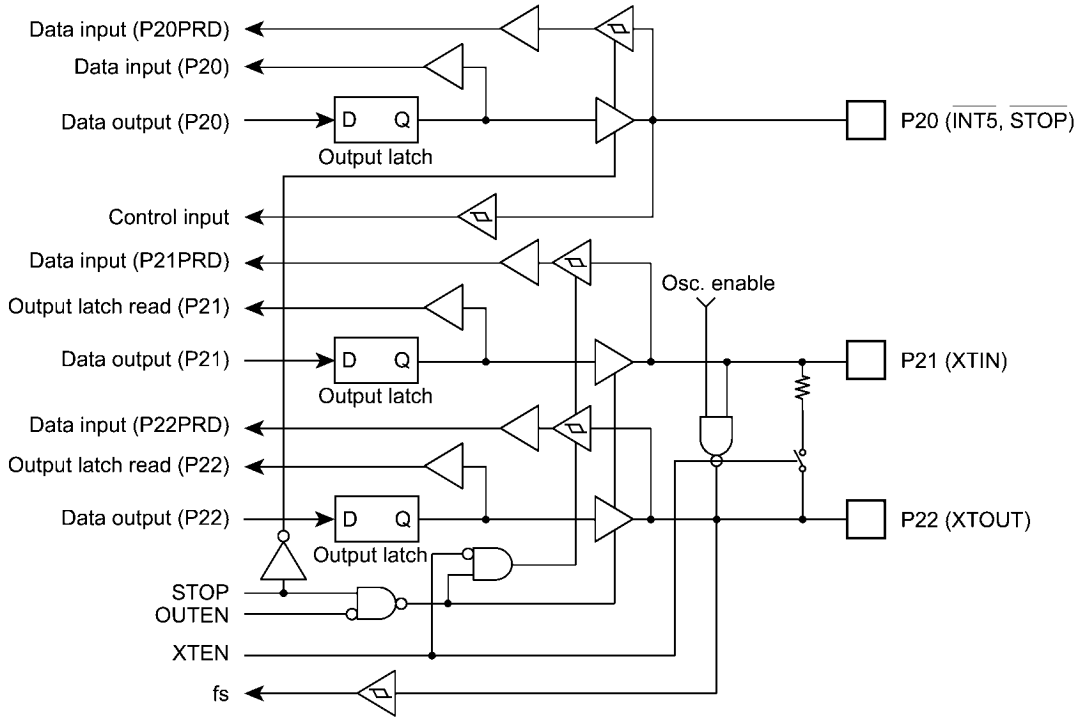


Figure 5-3 Port P2

	7	6	5	4	3	2	1	0	
P2DR (0002H) R/W						P22 XTOUT	P21 XTIN	P20 $\overline{\text{INT5}}$ $\overline{\text{STOP}}$	(Initial value: **** *111)
P2PRD (0F9CH) Read only	7	6	5	4	3	2	1	0	
						P22	P21	P20	

Note: 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.

5.3 Port P3 (P37 to P30)

Port P3 is a 3-bit output and a 5-bit input/output port.

It is also used as a timer/counter input/output, serial interface input/output or divider output.

When used as a timer/counter output, serial interface output or divider output, respective output latch (P3DR) should be set to "1".

It can be selected whether output circuit of P30 to P34 port is C-MOS output or a sink open drain individually, by setting P3OUTCR. When a corresponding bit of P3OUTCR is "0", the output circuit is selected to a sink open drain and when a corresponding bit of P3OUTCR is "1", the output circuit is selected to a C-MOS output. When used as an input port, serial interface input or timer/counter input, respective output control (P3OUTCR) should be set to "0" after P3DR is set to "1". During reset, the P3DR is initialized to "1", and the P3OUTCR is initialized to "0".

P3 port output latch (P3DR) and P3 port terminal input (P3PRD) are located on their respective address.

When read the output latch data, the P3DR should be read and when read the terminal input data, the P3PRD register should be read. If a read instruction is executed for the P3PRD and the P3OUTCR, read data of bits 7 to 5 are unstable.

Table 5-3 Register Programming for Multi-function ports (P34 to P30)

Function	Programmed Value	
	P3DR	P3OUTCR
Port input, serial interface input, or timer counter input	"1"	"0"
Port "0" output	"0"	Programming for each applications
Port "1" output, serial interface output, or timer counter output	"1"	

Table 5-4 Register Programming for Multi-function (P37 to P35)

Function	Programmed Value
	P3DR
Port "0" output	"0"
Port "1" output, timer counter output, or divider output	"1"

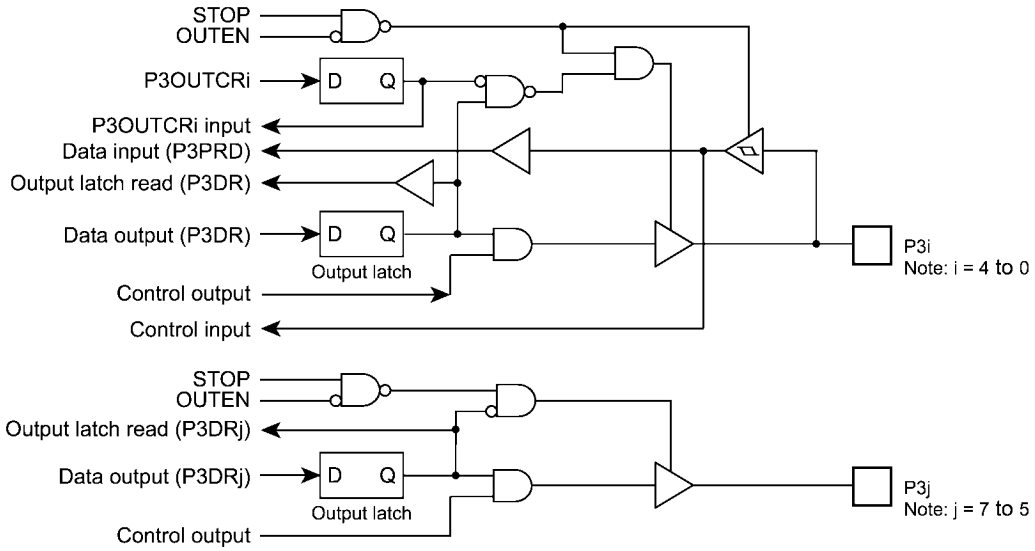


Figure 5-4 Port P3

P3DR (0003H) R/W

7	6	5	4	3	2	1	0
P37 $\overline{\text{DVO}}$	P36 $\overline{\text{PWM3}}$ $\overline{\text{PDO3}}$	P35 $\overline{\text{PWM4}}$ $\overline{\text{PDO4}}$ $\overline{\text{PPG4}}$	P34 $\overline{\text{PWM5}}$ $\overline{\text{PDO5}}$ TC5	P33 $\overline{\text{PWM6}}$ $\overline{\text{PDO6}}$ $\overline{\text{PPG6}}$ TC6	P32 $\overline{\text{SCK}}$	P31 SO TC3	P30 SI TC4

(Initial value: 1111 111)

P3OUTCR (0004H)

7	6	5	4	3	2	1	0

(Initial value: ***0 0000)

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

P3PRD (0F9DH) Read only

7	6	5	4	3	2	1	0
			P34	P33	P32	P31	P30

5.4 Port P5 (P57 to P50)

Port P5 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P5 is also used as a segment output of LCD. Input/output mode is specified by the P5 control register (P5CR).
When used as an input port, the corresponding bit of P5CR and P5LCR should be cleared to "0".
When used as an output port, the corresponding bit of P5CR should be set to "1", and the respective P5LCR bit should be cleared to "0".

When used as a segment pins of LCD, the respective bit of P5LCR should be set to "1".
During reset, the output latch (P5DR), P5CR and P5LCR are initialized to "0".
When the bit of P5CR and P5LCR is "0", the corresponding bit data by read instruction is a terminal input data.
When the bit of P5CR is "0" and that of P5LCR is "1", the corresponding bit data by read instruction is always "0".
When the bit of P5CR is "1", the corresponding bit data by read instruction is the value of P5DR.

Table 5-5 Register Programming for Multi-function Ports

Function	Programmed Value		
	P5DR	P5CR	P5LCR
Port input	*	"0"	"0"
Port "0" output	"0"	"1"	"0"
Port "1" output	"1"	"1"	"0"
LCD segment output	*	*	"1"

Note: Asterisk (*) indicates "1" or "0" either of which can be selected.

Table 5-6 Values Read from P5DR and Register Programming

Conditions		Values Read from P5DR
P5CR	P5LCR	
"0"	"0"	Terminal input data
"0"	"1"	"0"
"1"	"0"	Output latch contents
	"1"	

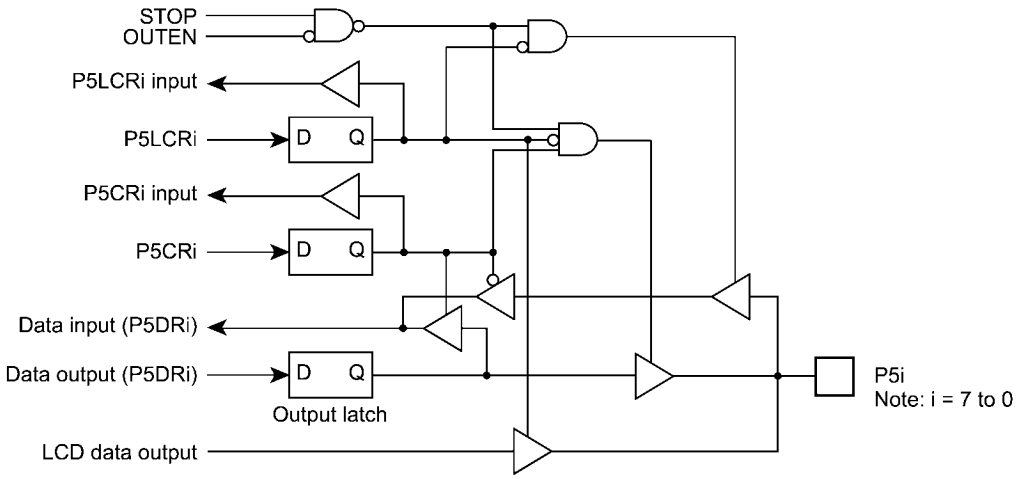


Figure 5-5 Port P5

P5DR (0005H) R/W

7	6	5	4	3	2	1	0	
P57 SEG16	P56 SEG17	P55 SEG18	P54 SEG19	P53 SEG20	P52 SEG21	P51 SEG22	P50 SEG23	(Initial value: 0000 0000)

P5LCR (0F9FH)

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

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

P5CR (000AH)

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

P5CR	P5 port input/output control (Set for each bit individually)	0: Input mode 1: Output mode	R/W

Note: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.

5.5 Port P6 (P67 to P60)

Port P6 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P6 is also used as an analog input, Key on Wake up input, timer/counter input and external interrupt input. Input/output mode is specified by the P6 control register (P6CR1) and input control register (P6CR2).

When used as an output port, the corresponding bit of P6CR1 should be set to "1".

When used as an input port, timer/counter input or an external interrupt input, the corresponding bit of P6CR1 should be cleared to "0", and then, the corresponding bit of P6CR2 should be set to "1".

When used as an analog input or key on wake up input, the corresponding bit of P6CR1 should be cleared to "0", and then, the corresponding bit of P6CR2 should be cleared to "0" .

The output latch of each digital input port with multiple functions should be set to "0" to prevent flow-through current. Therefore, the output latch of each port to be used for analog input should be preprogrammed to "0". The conversion input channel to be used is actually selected by ADCCR1<SAIN>.

During reset, the output latch (P6DR) and P6CR1 are initialized to 0", P6CR2 is initialized to "1".

When the bit of P6CR1 and P6CR2 is "0", the corresponding bit data by read instruction is always "0".

When the bit of P6CR1 is "0" and that of P6CR2 is "1", the corresponding bit data by read instruction is a terminal input data.

When the bit of P6CR1 is "1", the corresponding bit data by read instruction is the value of P6DR.

Table 5-7 Register Programming for Multi-function Ports

Function	Programmed Value		
	P6DR	P6CR1	P6CR2
Port input external interrupt input or timer counter input	*	"0"	"1"
Analog input or key-on wake-up input	*	"0"	"0"
Port "0" output	"0"	"1"	*
Port "1" output	"1"	"1"	*

Note: Asterisk (*) indicates "1" or "0" either of which can be selected.

Table 5-8 Values Read from P6DR and Register Programming

Conditions		Values Read from P6DR
P6CR1	P6CR2	
"0"	"0"	"0"
"0"	"1"	Terminal input data
"1"	"0"	Output latch contents
	"1"	

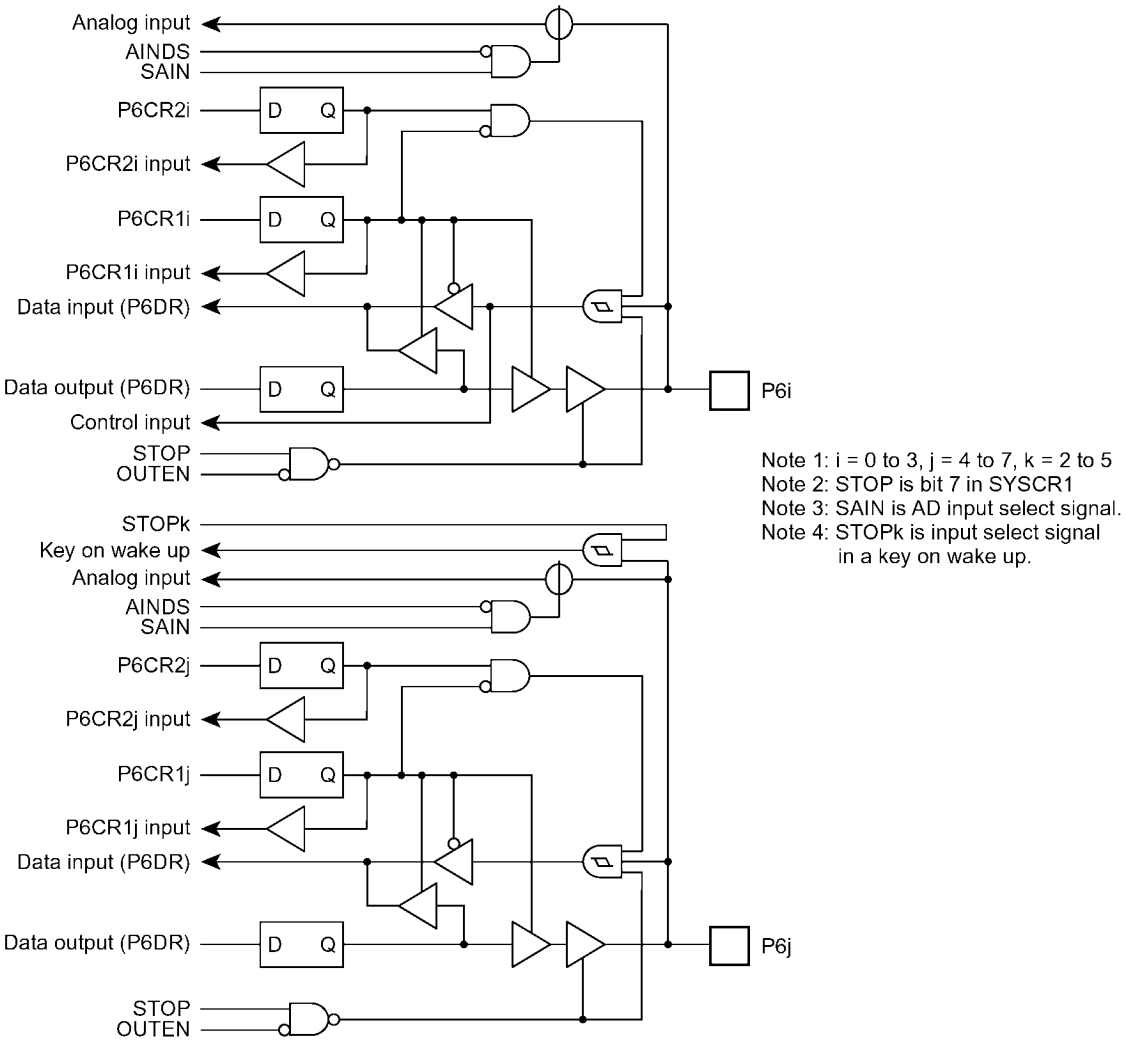


Figure 5-6 Port P6

Note 1: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.

Note 2: When used as an analog inport, be sure to clear the corresponding bit of P6CR2 to disable the port input.

Note 3: Do not set the output mode (P6CR1 = "1") for the pin used as a analog input pin.

Note 4: Pins not used for analog input can be used as I/O ports. During AD conversion, output instructions should not be executed to keep a precision. In addition, a variable signal should not be input to a port adjacent to the analog input during AD conversion.

P6DR (0006H) R/W

7	6	5	4	3	2	1	0
P67 AIN7 STOP5	P66 AIN6 STOP4	P65 AIN5 STOP3	P64 AIN4 STOP2	P63 AIN3 $\overline{INT0}$	P62 AIN2 ECNT	P61 AIN1 ECIN	P60 AIN0

(Initial value: 0000 0000)

P6CR1 (000BH)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

P6CR1	I/O control for port P6 (Specified for each bit)	0: Port input, Key on wake up input, Analog input, external interrupt input or timer counter input 1: Port output	R/W

P6CR2 (000CH)

7	6	5	4	3	2	1	0

(Initial value: 1111 1111)

P6CR2	P6 port input control (Specified for each bit)	0: Analog input or Key on wake up input 1: Port input , external interrupt input or timer counter input	R/W

5.6 Port P7 (P77 to P70)

Port P7 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P7 is also used as a segment output of LCD. Input/output mode is specified by the P7 control register (P7CR).

When used as an input port, the corresponding bit of P7CR and P7LCR should be cleared to "0".

When used as an output port, the corresponding bit of P7CR should be set to "1", and the respective P7LCR bit should be cleared to "0".

When used as a segment pins of LCD, the respective bit of P7LCR should be set to "1".

During reset, the output latch (P7DR), P7CR and P7LCR are initialized to "0".

When the bit of P7CR and P7LCR is "0", the corresponding bit data by read instruction is a terminal input data.

When the bit of P7CR is "0" and that of P7LCR is "1", the corresponding bit data by read instruction is always "0".

When the bit of P7CR is "1", the corresponding bit data by read instruction is the value of P7DR.

Table 5-9 Register Programming for Multi-function Ports

Function	Programmed Value		
	P7DR	P7CR	P7LCR
Port input	*	"0"	"0"
Port "0" output	"0"	"1"	"0"
Port "1" output	"1"	"1"	"0"
LCD segment output	*	*	"1"

Note: Asterisk (*) indicates "1" or "0" either of which can be selected.

Table 5-10 Values Read from P7DR and Register Programming

Conditions		Values Read from P7DR
P7CR	P7LCR	
"0"	"0"	Terminal input data
"0"	"1"	"0"
"1"	"0"	Output latch contents
	"1"	

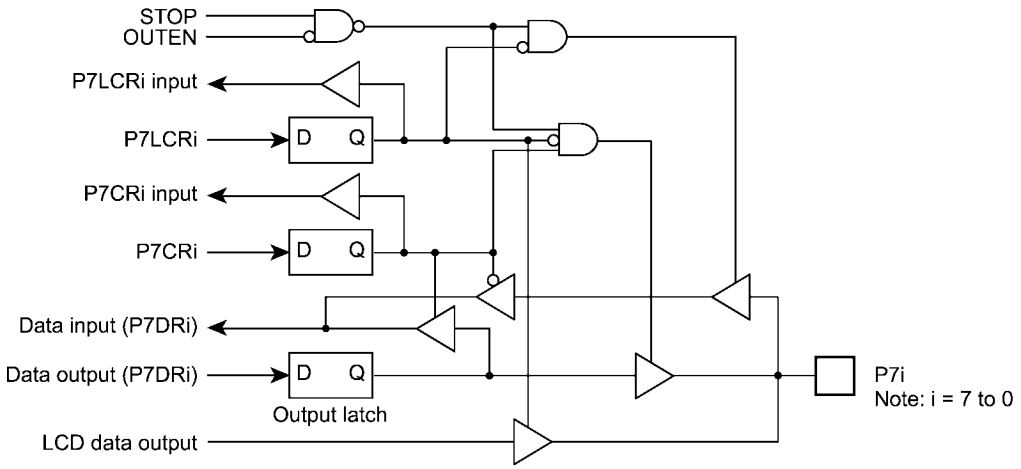


Figure 5-7 Port P7

P7DR (0007H) R/W

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

P7LCR (0FA0H)

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

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

P7CR (000DH)

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

P7CR	P7 port input/output control (set for each bit individually)	0: Input mode 1: Output mode	R/W

Note: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.

5.7 Port P8 (P87 to P80)

Port P8 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P8 is also used as a segment output of LCD. Input/output mode is specified by the P8 control register (P8CR).

When used as an input port, the corresponding bit of P8CR and P8LCR should be cleared to "0".

When used as an output port, the corresponding bit of P8CR should be set to "1", and the respective P8LCR bit should be cleared to "0".

When used as a segment pins of LCD, the respective bit of P8LCR should be set to "1".

During reset, the output latch (P8DR), P8CR and P8LCR are initialized to "0".

When the bit of P8CR and P8LCR is "0", the corresponding bit data by read instruction is a terminal input data.

When the bit of P8CR is "0" and that of P8LCR is "1", the corresponding bit data by read instruction is always "0".

When the bit of P8CR is "1", the corresponding bit data by read instruction is the value of P8DR.

Table 5-11 Register Programming for Multi-function ports

Function	Port Input		
	P8DR	P8CR	P8LCR
Port input	*	"0"	"0"
Port "0" output	"0"	"1"	"0"
Port "1" output	"1"	"1"	"0"
LCD segment output	*	*	"1"

Note: Asterisk (*) indicates "1" or "0" either of which can be selected.

Table 5-12 Values Read from P8DR and Register Programming

Conditions		Values Read from P8DR
P8CR	P8LCR	
"0"	"0"	Terminal input data
"0"	"1"	"0"
"1"	"0"	Output latch contents
	"1"	

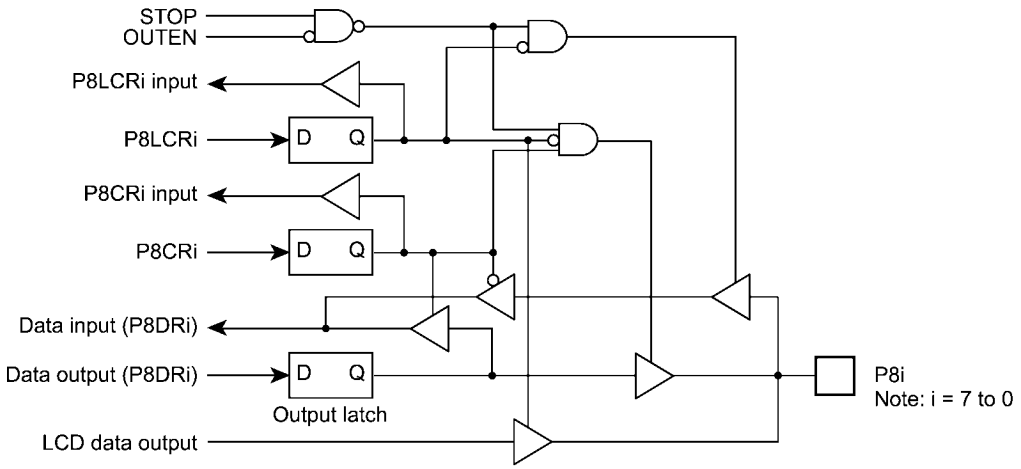


Figure 5-8 Port P8

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 (0FA1H)

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

P8LCR	P8 port segment output control (Specified for each bit)	0: Input/Output port 1: LCD segment output	R/W

P8CR (0024H)

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

P8CR	P8 port input/output control (Specified for each bit)	0: Input mode 1: Output mode	R/W

Note: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.

6. Time Base Timer (TBT)

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

6.1 Time Base Timer

6.1.1 Configuration

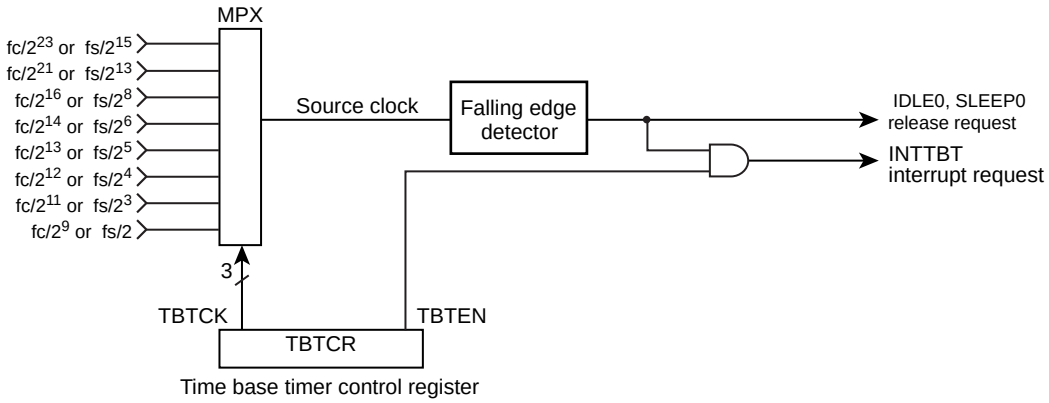


Figure 6-1 Time Base Timer configuration

6.1.2 Control

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

Time Base Timer Control Register

	7	6	5	4	3	2	1	0	
TBTCR (0036H)	(DVOEN)	(DVOCK)		(DV7CK)	TBTEN	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 6-1 Time Base Timer Interrupt Frequency (Example : fc = 16.0 MHz, fs = 32.768 kHz)

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

6.1.3 Function

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

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

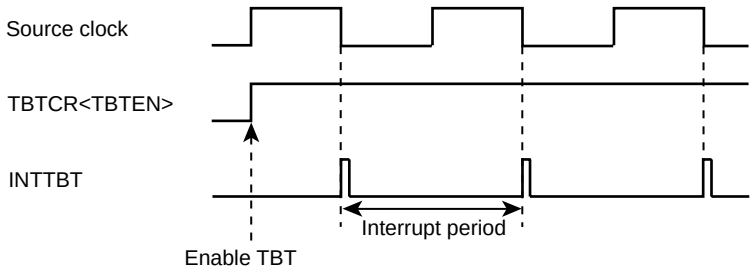


Figure 6-2 Time Base Timer Interrupt

6.2 Divider Output ($\overline{DVO}$)

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

6.2.1 Configuration

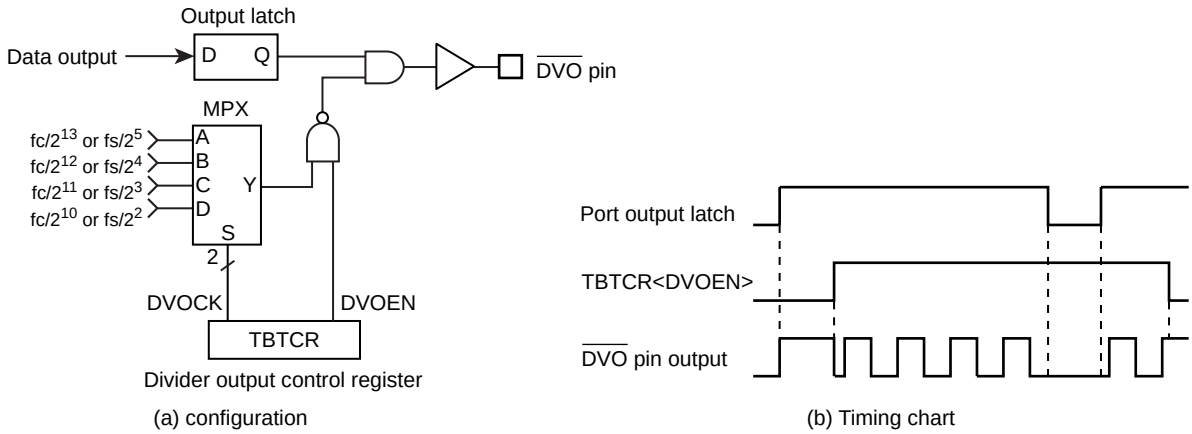


Figure 6-3 Divider Output

6.2.2 Control

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

Time Base Timer Control Register

	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 ($\overline{DVO}$) 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 6-2 Divider Output Frequency (Example : fc = 16.0 MHz, fs = 32.768 kHz)

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

7. Watchdog Timer (WDT)

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

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

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

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

7.1 Watchdog Timer Configuration

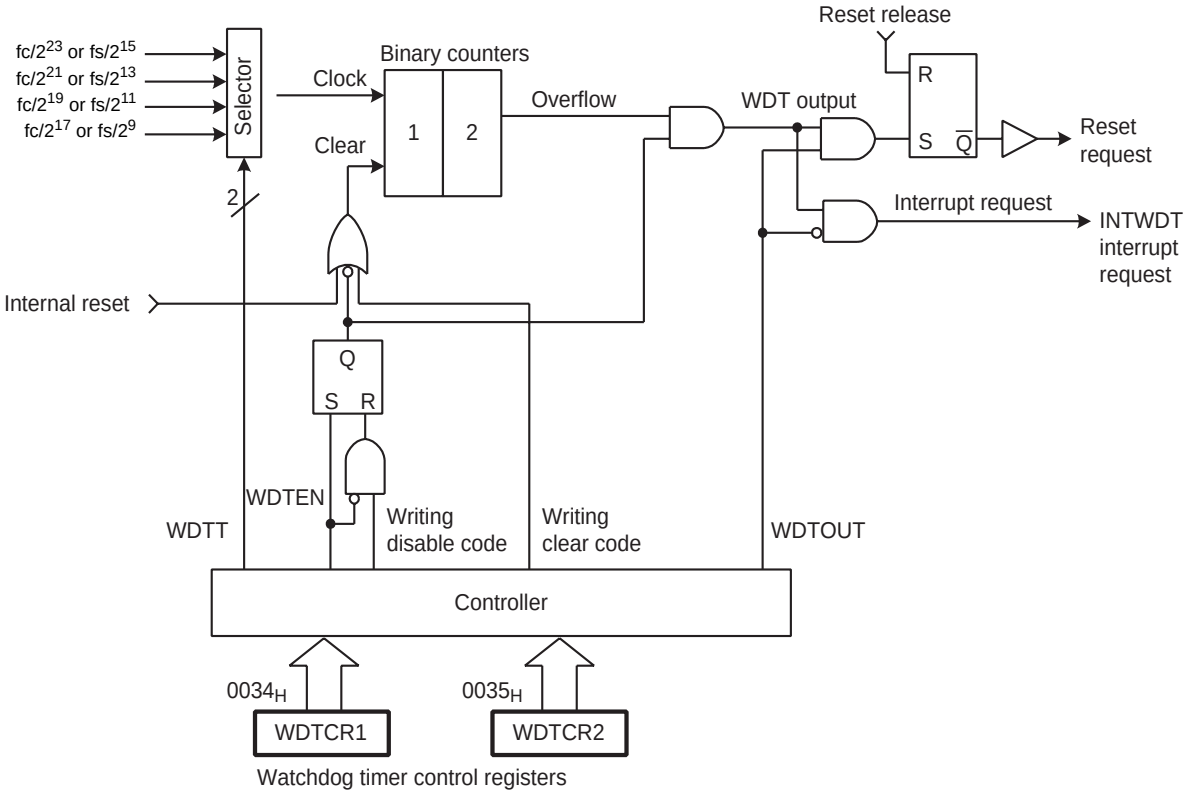


Figure 7-1 Watchdog Timer Configuration

7.2 Watchdog Timer Control

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

7.2.1 Malfunction Detection Methods Using the Watchdog Timer

The CPU malfunction is detected, as shown below.

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

If the CPU malfunctions such as endless loops or the deadlock conditions occur for some reason, the watchdog timer output is activated by the binary-counter overflow unless the binary counters are cleared. When WDTCR1<WDTOUT> is set to “1” at this time, the reset request is generated and 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
detect  LD      (WDTCR2), 4EH       : Clears the binary counters.
time
        :
Within  :
3/4 of
WDT     LD      (WDTCR2), 4EH       : Clears the binary counters.
detect
time
```

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 “7.2.3 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>.

7.2.2 Watchdog Timer Enable

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

7.2.3 Watchdog Timer Disable

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

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

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

Example :Disabling the watchdog timer

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

Table 7-1 Watchdog Timer Detection Time (Example: 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

7.2.4 Watchdog Timer Interrupt (INTWDT)

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

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

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

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

Example :Setting watchdog timer interrupt

```
LD      SP, 063FH           : Sets the stack pointer
LD      (WDTCR1), 00001000B : WDTOUT ← 0
```

7.2.5 Watchdog Timer Reset

When a binary-counter overflow occurs while WDTCR1<WDTOUT> is set to “1”, a watchdog timer reset request is generated. When a watchdog timer reset request is generated, the 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.

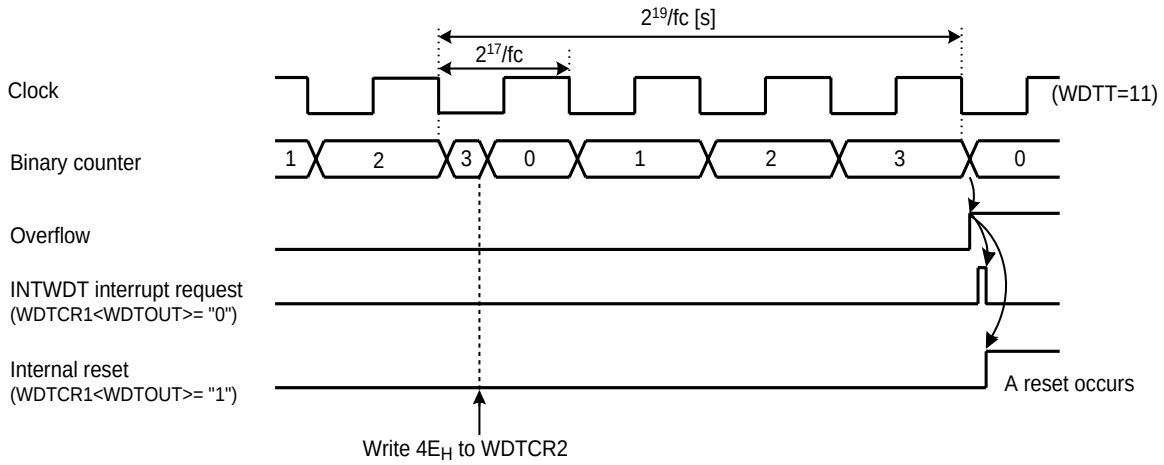


Figure 7-2 Watchdog Timer Interrupt

7.3 Address Trap

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

Watchdog Timer Control Register 1

WDTCR1 (0034H)

7	6	5	4	3	2	1	0
		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

7.3.1 Selection of Address Trap in Internal RAM (ATAS)

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

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

7.3.2 Selection of Operation at Address Trap (ATOUT)

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

7.3.3 Address Trap Interrupt (INTATRAP)

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

7.3.4 Address Trap Reset

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

8. 18-Bit Timer/Counter (TC1)

8.1 Configuration

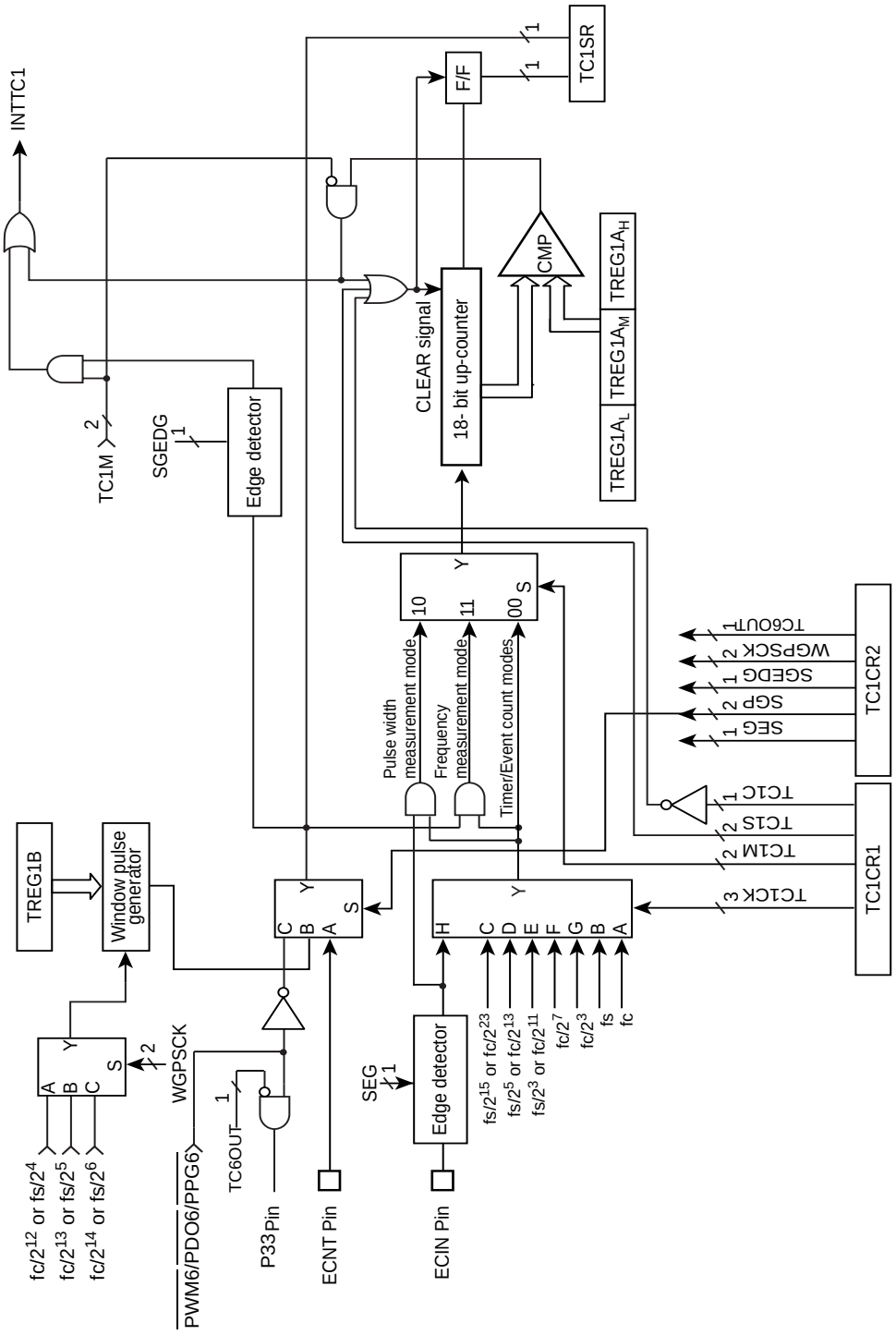
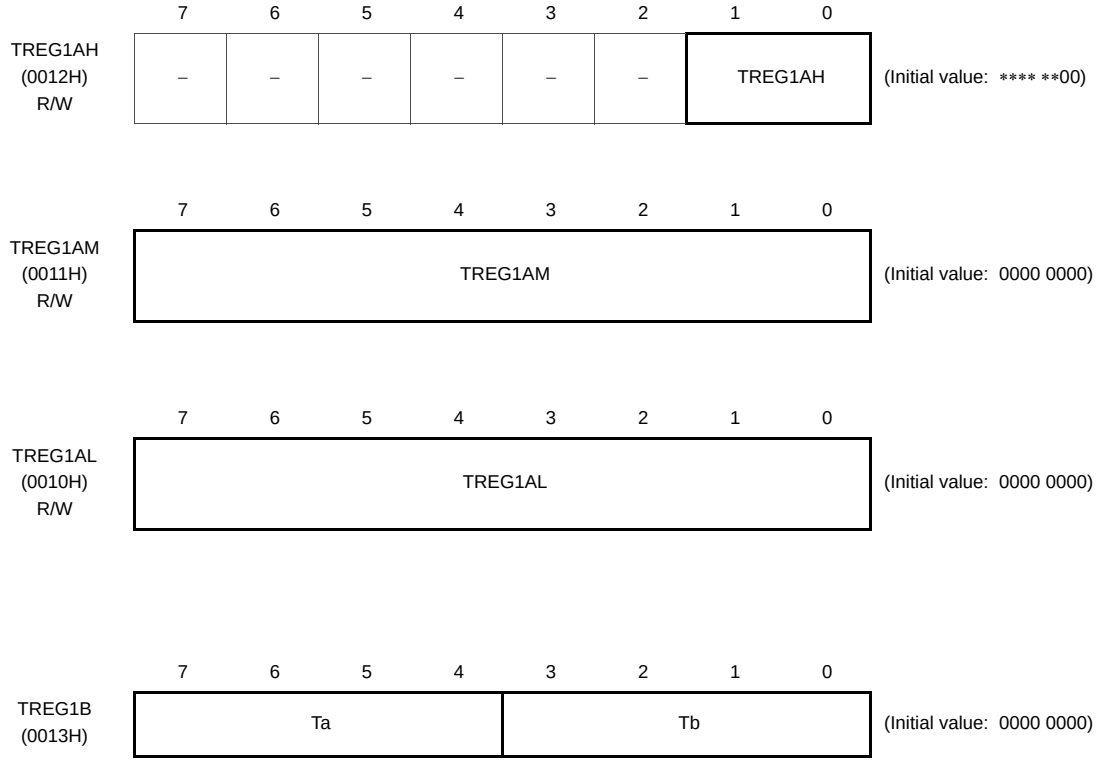


Figure 8-1 Timer/Counter1

8.2 Control

The Timer/counter 1 is controlled by timer/counter 1 control registers (TC1CR1/TC1CR2), an 18-bit timer register (TREG1A), and an 8-bit internal window gate pulse setting register (TREG1B).

Timer register



		WGPSCK	NORMAL1/2,IDLE1/2 modes		SLOW1/2, SLEEP1/2 modes	
			DV7CK=0	DV7CK=1		
Ta	Setting "H" level period of the window gate pulse	00 01 10	(16 - Ta) × 2^{12}/fc (16 - Ta) × 2^{13}/fc (16 - Ta) × 2^{14}/fc	(16 - Ta) × 2^{4}/fs (16 - Ta) × 2^{5}/fs (16 - Ta) × 2^{6}/fs	(16 - Ta) × 2^{4}/fs (16 - Ta) × 2^{5}/fs (16 - Ta) × 2^{6}/fs	R/W
Tb	Setting "L" level period of the window gate pulse	00 01 10	(16 - Tb) × 2^{12}/fc (16 - Tb) × 2^{13}/fc (16 - Tb) × 2^{14}/fc	(16 - Tb) × 2^{4}/fs (16 - Tb) × 2^{5}/fs (16 - Tb) × 2^{6}/fs	(16 - Tb) × 2^{4}/fs (16 - Tb) × 2^{5}/fs (16 - Tb) × 2^{6}/fs	

Timer/counter 1 control register 1

	7	6	5	4	3	2	1	0	
TC1CR1 (0014H)	TC1C	TC1S		TC1CK			TC1M		(Initial value: 1000 1000)

TC1C	Counter/overfow flag controll	0: Clear Counter/overflow flag (“1” is automatically set after clearing.) 1: Not clear Counter/overflow flag				R/W
TC1S	TC1 start control	00: Stop and counter clear and overflow flag clear 10: Start *1: Reserved				R/W
TC1CK	TC1 source clock select		NORMAL1/2,IDLE1/2 modes		SLOW1/2 mode	SLEEP1/2 mode	R/W
			DV7CK="0"	DV7CK="1"		
		000:	fc	fc	fc	fc
		001:	fs	fs	-	-
		010:	$fc/2^{23}$	$fs/2^{15}$	$fs/2^{15}$	$fs/2^{15}$
		011:	$fc/2^{13}$	$fs/2^{5}$	$fs/2^{5}$	$fs/2^{5}$
		100:	$fc/2^{11}$	$fs/2^{3}$	$fs/2^{3}$	$fs/2^{3}$
		101:	$fc/2^{7}$	$fc/2^{7}$	-	-
		110:	$fc/2^{3}$	$fc/2^{3}$	-	-
		111:	External clock (ECIN pin input)			
TC1M	TC1 mode select	00: Timer/Event counter mode 01: Reserved 10: Pulse width measurement mode 11: Frequency measurement mode				R/W

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

Note 2: Writing to the low-byte of the timer register 1A (TREG1AL, TREG1AM), the compare function is inhibited until the high-byte (TREG1AH) is written.

Note 3: Set the mode and source clock, and edge (selection) when the TC1 stops (TC1S=00).

Note 4: “fc” can be selected as the source clock only in the timer mode during SLOW mode and in the pulse width measurement mode during NORMAL 1/2 or IDLE 1/2 mode.

Note 5: When a read instruction is executed to the timer register (TREG1A), the counter immediate value, not the register set value, is read out. Therefore it is impossible to read out the written value of TREG1A. To read the counter value, the read instruction should be executed when the counter stops to avoid reading unstable value.

Note 6: Set the timer register (TREG1A) to ≥ 1.

Note 7: When using the timer mode and pulse width measurement mode, set TC1CK (TC1 source clock select) to internal clock.

Note 8: When using the event counter mode, set TC1CK (TC1 source clock select) to external clock.

Note 9: Because the read value is different from the written value, do not use read-modify-write instructions to TREG1A.

Note 10:$fc/2^{7}$, $fc/2^{3}$can not be used as source clock in SLOW/SLEEP mode.

Note 11:The read data of bits 7 to 2 in TREG1AH are always “0”. (Data “1” can not be written.)

Timer/Counter 1 control register 2

	7	6	5	4	3	2	1	0	
TC1CR2 (0015H)	SEG	SGP		SGEDG	WGPSCK		TC6OUT	"0"	(Initial value: 0000 000*)

SEG	External input clock (ECIN) edge select	0: Counts at the falling edge 1: Counts at the both (falling/rising) edges				R/W
SGP	Window gate pulse select	00: ECNT input 01: Internal window gate pulse (TREG1B) 10: $\overline{PWM6}/\overline{PDO6}/\overline{PPG6}$ (TC6)output 11: Reserved				R/W
SGEDG	Window gate pulse interrupt edge select	0: Interrupts at the falling edge 1: Interrupts at the falling/rising edges				R/W
WGPSCK	Window gate pulse source clock select		NORMAL1/2,IDLE1/2 modes		SLOW1/2 mode	SLEEP1/2 mode	R/W
			DV7CK="0"	DV7CK="1"		
		00:	2^{12}/fc	2^{4}/fs	2^{4}/fs	2^{4}/fs
		01:	2^{13}/fc	2^{5}/fs	2^{5}/fs	2^{5}/fs
		10:	2^{14}/fc	2^{6}/fs	2^{6}/fs	2^{6}/fs
		11:	Reserved	Reserved	Reserved	Reserved
TC6OUT	TC6 output ($\overline{PWM6}/\overline{PDO6}/\overline{PPG6}$) external output select	0: Output to P33 1: No output to P33				R/W

Note 1: fc; High-frequency clock [Hz] fs; Low-frequency clock [Hz] *; Don't care
Note 2: Set the mode, source clock, and edge (selection) when the TC1 stops (TC1S = 00).
Note 3: If there is no need to use $\overline{PWM6}/\overline{PDO6}/\overline{PPG6}$ as window gate pulse of TC1 always write "0" to TC6OUT.
Note 4: Make sure to write TC1CR2 "0" to bit 0 in TC1CR2.
Note 5: When using the event counter mode or pulse width measurement mode, set SEG to "0".

TC1 status register

	7	6	5	4	3	2	1	0	
TC1SR (0016H)	HECF	HEOVF	"0"	"0"	"0"	"0"	"0"	"0"	(Initial value: 0000 0000)

HECF	Operating Status monitor	0: Stop (during Tb) or disable 1: Under counting (during Ta)	Read only
HEOVF	Counter overflow monitor	0: No overflow 1: Overflow status	

8.3 Function

TC1 has four operating modes. The timer mode of the TC1 is used at warm-up when switching form SLOW mode to NORMAL2 mode.

8.3.1 Timer mode

In this mode, counting up is performed using the internal clock. The contents of TREGIA are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. Counting up resumes after the counter is cleared.

Table 8-1 Source clock (internal clock) of Timer/Counter 1

Source Clock				Resolution		Maximum Time Setting	
NORMAL1/2, IDLE1/2 Mode		SLOW Mode	SLEEP Mode	fc = 16 MHz	fs =32.768 kHz	fc = 16 MHz	fs =32.768 kHz
DV7CK = 0	DV7CK = 1						
$fc/2^{23}$ [Hz]	$fs/2^{15}$ [Hz]	$fs/2^{15}$ [Hz]	$fs/2^{15}$ [Hz]	0.52 s	1 s	38.2 h	72.8 h
$fc/2^{13}$	$fs/2^{5}$	$fs/2^{5}$	$fs/2^{5}$	512 ms	0.98 ms	2.2 min	4.3 min
$fc/2^{11}$	$fs/2^{3}$	$fs/2^{3}$	$fs/2^{3}$	128 ms	244 ms	0.6 min	1.07 min
$fc/2^{7}$	$fc/2^{7}$	-	-	8 ms	-	2.1 s	-
$fc/2^{3}$	$fc/2^{3}$	-	-	0.5 ms	-	131.1 ms	-
fc	fc	fc (Note)	-	62.5 ns	-	16.4 ms	-
fs	fs	-	-	-	30.5 ms	-	8 s

Note: When fc is selected for the source clock in SLOW mode, the lower bits 11 of TREG1A is invalid, and a match of the upper bits 7 makes interrupts.

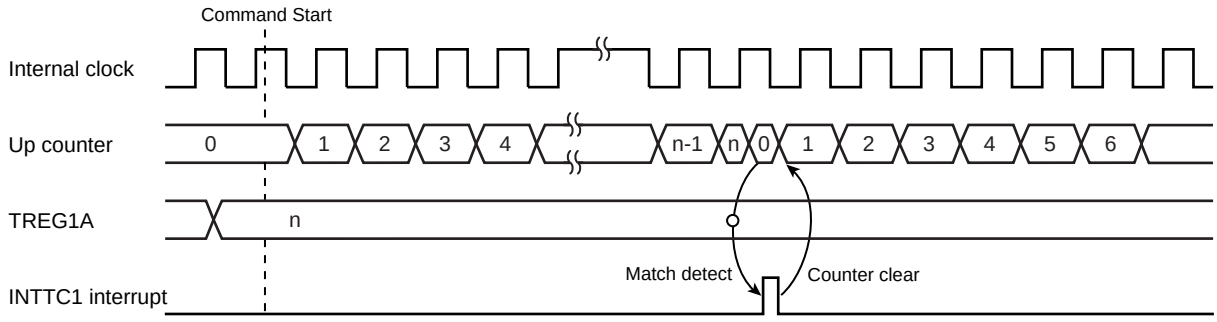


Figure 8-2 Timing chart for timer mode

8.3.2 Event Counter mode

It is a mode to count up at the falling edge of the ECIN pin input. When using this mode, set TC1CR1<TC1CK> to the external clock and then set TC1CR2<SEG> to "0" (Both edges can not be used).

The countents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. Counting up resumes for ECIN pin input edge each after the counter is cleared.

The maximum applied frequency is $fc/2^4$ [Hz] in NORMAL 1/2 or IDLE 1/2 mode and $fs/2^4$[Hz] in SLOW or SLEEP mode . Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.

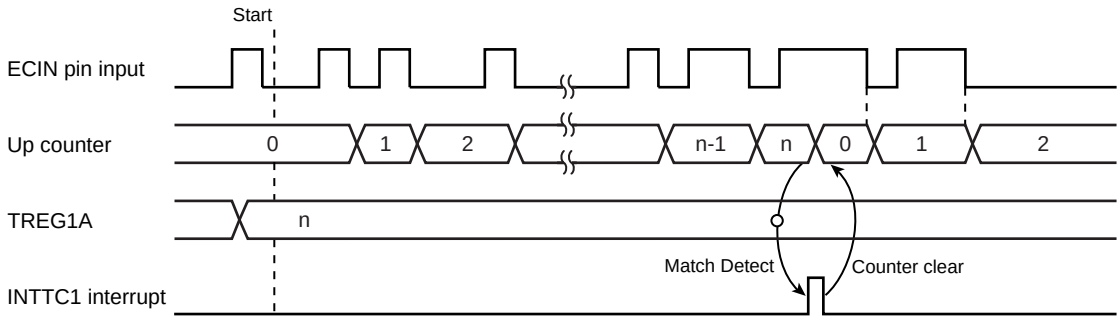


Figure 8-3 Event counter mode timing chart

8.3.3 Pulse Width Measurement mode

In this mode, pulse widths are counted on the falling edge of logical AND-ed pulse between ECIN pin input (window pulse) and the internal clock. When using this mode, set TC1CR1<TC1CK> to suitable internal clock and then set TC1CR2<SEG> to “0” (Both edges can not be used).

An INTTC1 interrupt is generated when the ECIN input detects the falling edge of the window pulse or both rising and falling edges of the window pulse, that can be selected by TC1CR2<SGEDG>.

The contents of TREG1A should be read while the count is stopped (ECIN pin is low), then clear the counter using TC1CR1<TC1C> (Normally, execute these process in the interrupt program).

When the counter is not cleared by TC1CR1<TC1C>, counting-up resumes from previous stopping value. When up counter is counted up from 3FFFFH to 00000H, an overflow occurs. At that time, TC1SR<HEOVF> is set to “1”. TC1SR<HEOVF> remains the previous data until the counter is required to be cleared by TC1CR1<TC1C>.

Note: In pulse width measurement mode, if TC1CR1<TC1S> is written to "00" while ECIN input is "1", INTTC1 interrupt occurs. According to the following step, when timer counter is stopped, INTTC1 interrupt latch should be cleared to "0".

Example :

```
TC1STOP :
        ¦                ¦
        DI                              ; Clear IMF
        CLR     (EIRL). 7               ; Clear bit7 of EIRL
        LD      (TC1CR1), 00011010B     ; Stop timer couter 1
        LD      (ILL), 01111111B        ; Clear bit7 of ILL
        SET     (EIRL). 7               ; Set bit7 of EIRL
        EI                              ; Set IMF
        ¦                ¦
```

Note 1: When SGEDG (window gate pulse interrupt edge select) is set to both edges and ECIN pin input is "1" in the pulse width measurement mode, an INTTC1 interrupt is generated by setting TC1S (TC1 start control) to "10" (start).

Note 2: In the pulse width measurement mode, HECF (operating status monitor) cannot used.

Note 3: Because the up counter is counted on the falling edge of logical AND-ed pulse (between ECIN pin input and the internal clock), if ECIN input becomes falling edge while internal source clock is "H" level, the up counter stops plus "1".

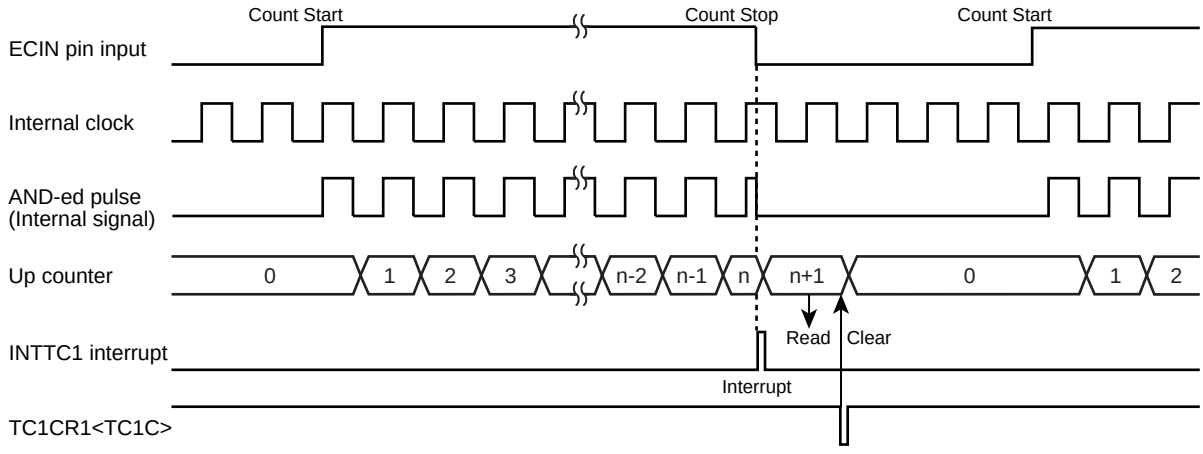


Figure 8-4 Pulse width measurement mode timing chart

8.3.4 Frequency Measurement mode

In this mode, the frequency of ECIN pin input pulse is measured. When using this mode, set TC1CR1<TC1CK> to the external clock.

The edge of the ECIN input pulse is counted during “H” level of the window gate pulse selected by TC1CR2<SGP>. To use ECNT input as a window gate pulse, TC1CR2<SGP> should be set to “00”.

An INTTC1 interrupt is generated on the falling edge or both the rising/falling edges of the window gate pulse, that can be selected by TC1CR2<SGEDG>. In the interrupt service program, read the contents of TREG1A while the count is stopped (window gate pulse is low), then clear the counter using TC1CR1<TC1C>. When the counter is not cleared, counting up resumes from previous stopping value.

The window pulse status can be monitored by TC1SR<HECF>.

When up counter is counted up from 3FFFFH to 00000H, an overflow occurs. At that time, TC1SR<HEOVF> is set to “1”. TC1SR<HEOVF> remains the previous data until the counter is required to be cleared by TC1CR1<TC1C>.

Using TC6 output ($\overline{\text{PWM6}}/\overline{\text{PDO6}}/\overline{\text{PPG6}}$) for the window gate pulse, external output of $\overline{\text{PWM6}}/\overline{\text{PDO6}}/\overline{\text{PPG6}}$ to P33 can be controlled using TC1CR2<TC6OUT>. Zero-clearing TC1CR2<TC6OUT> outputs $\overline{\text{PWM6}}/\overline{\text{PDO6}}/\overline{\text{PPG6}}$ to P33; setting 1 in TC1CR2<TC6OUT> does not output $\overline{\text{PWM6}}/\overline{\text{PDO6}}/\overline{\text{PPG6}}$ to P33. (TC1CR2<TC6OUT> is used to control output to P33 only. Thus, use the timer counter 6 control register to operate/stop $\overline{\text{PWM6}}/\overline{\text{PDO6}}/\overline{\text{PPG6}}$.)

When the internal window gate pulse is selected, the window gate pulse is set as follows.

Table 8-2 Internal window gate pulse setting time

		WGPSCK	NORMAL1/2,IDLE1/2 modes		SLOW1/2, SLEEP1/2 modes	
			DV7CK=0	DV7CK=1		
Ta	Setting "H" level period of the window gate pulse	00 01 10	(16 - Ta) × 2^{12}/fc (16 - Ta) × 2^{13}/fc (16 - Ta) × 2^{14}/fc	(16 - Ta) × 2^{4}/fs (16 - Ta) × 2^{5}/fs (16 - Ta) × 2^{6}/fs	(16 - Ta) × 2^{4}/fs (16 - Ta) × 2^{5}/fs (16 - Ta) × 2^{6}/fs	R/W
Tb	Setting "L" level period of the window gate pulse	00 01 10	(16 - Tb) × 2^{12}/fc (16 - Tb) × 2^{13}/fc (16 - Tb) × 2^{14}/fc	(16 - Tb) × 2^{4}/fs (16 - Tb) × 2^{5}/fs (16 - Tb) × 2^{6}/fs	(16 - Tb) × 2^{4}/fs (16 - Tb) × 2^{5}/fs (16 - Tb) × 2^{6}/fs	

The internal window gate pulse consists of “H” level period (Ta) that is counting time and “L” level period (Tb) that is counting stop time. Ta or Tb can be individually set by TREG1B. One cycle contains Ta + Tb.

Note 1: Because the internal window gate pulse is generated in synchronization with the internal divider, it may be delayed for a maximum of one cycle of the source clock (WGPSCK) immediately after start of the timer.

Note 2: Set the internal window gate pulse when the timer counter is not operating or during the Tb period. When Tb is overwritten during the Tb period, the update is valid from the next Tb period.

Note 3: In case of TC1CR2<SEG> = "1", if window gate pulse becomes falling edge, the up counter stops plus "1" regardless of ECIN input level. Therefore, if ECIN is always "H" or "L" level, count value becomes "1".

Note 4: In case of TC1CR2<SEG> = "0", because the up counter is counted on the falling edge of logical AND-ed pulse (between ECIN pin input and window gate pulse), if window gate pulse becomes falling edge while ECIN input is "H" level, the up counter stops plus "1". Therefore, if ECIN input is always "H" level, count value becomes "1".

Table 8-3 Table Setting Ta and Tb (WGPSCK = 10, fc = 16 MHz)

Setting Value	Setting time	Setting Value	Setting time
0	16.38ms	8	8.19ms
1	15.36ms	9	7.17ms
2	14.34ms	A	6.14ms
3	13.31ms	B	5.12ms
4	12.29ms	C	4.10ms
5	11.26ms	D	3.07ms
6	10.24ms	E	2.05ms
7	9.22ms	F	1.02ms

Table 8-4 Table Setting Ta and Tb (WGPSCK = 10, fs = 32.768 kHz)

Setting Valuen	Setting time	Setting Value	Setting time
0	31.25ms	8	15.63ms
1	29.30ms	9	13.67ms
2	27.34ms	A	11.72ms
3	25.39ms	B	9.77ms
4	23.44ms	C	7.81ms
5	21.48ms	D	5.86ms
6	19.53ms	E	3.91ms
7	17.58ms	F	1.95ms

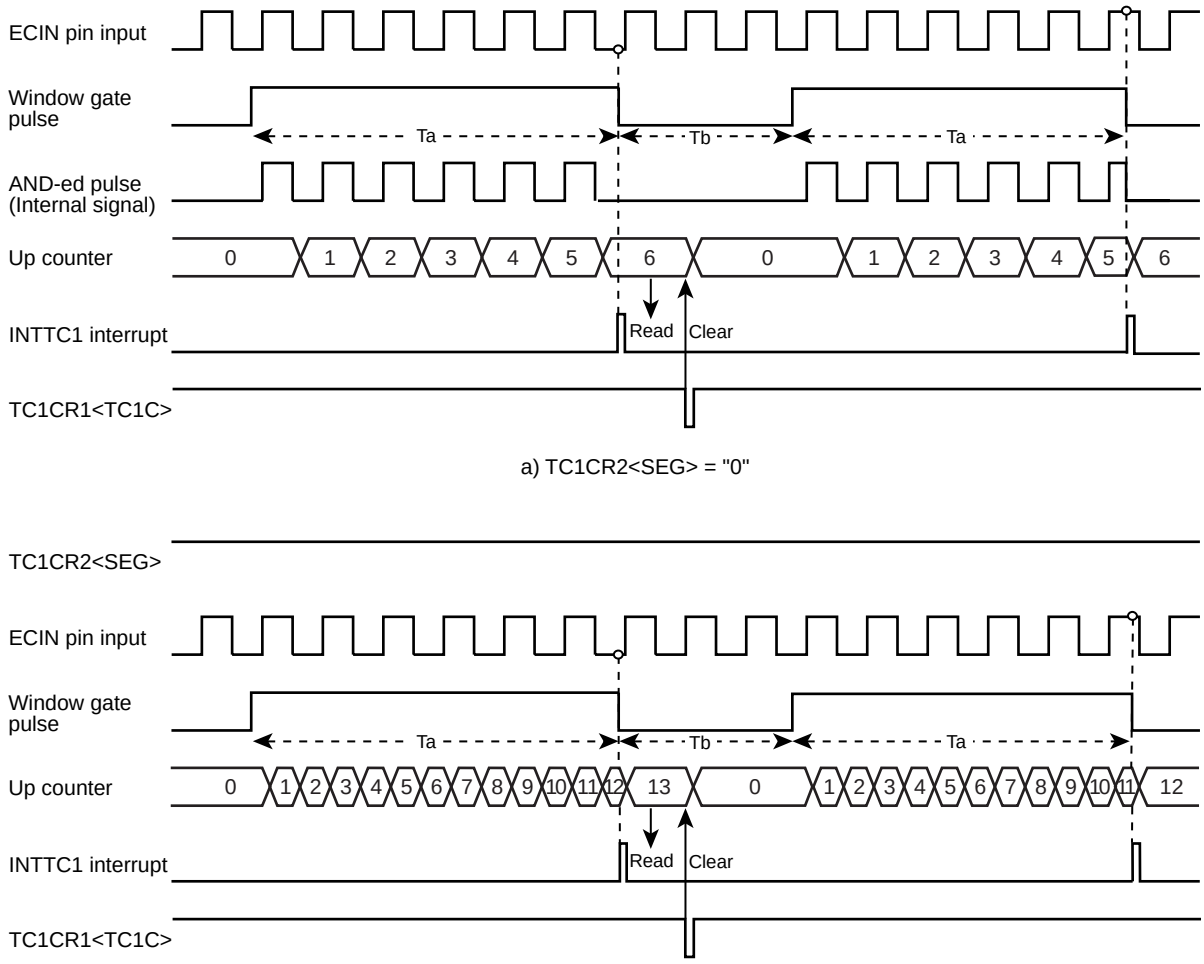


Figure 8-5 Timing chart for the frequency measurement mode (Window gate pulse falling interrupt)

9. 8-Bit TimerCounter (TC3, TC4)

9.1 Configuration

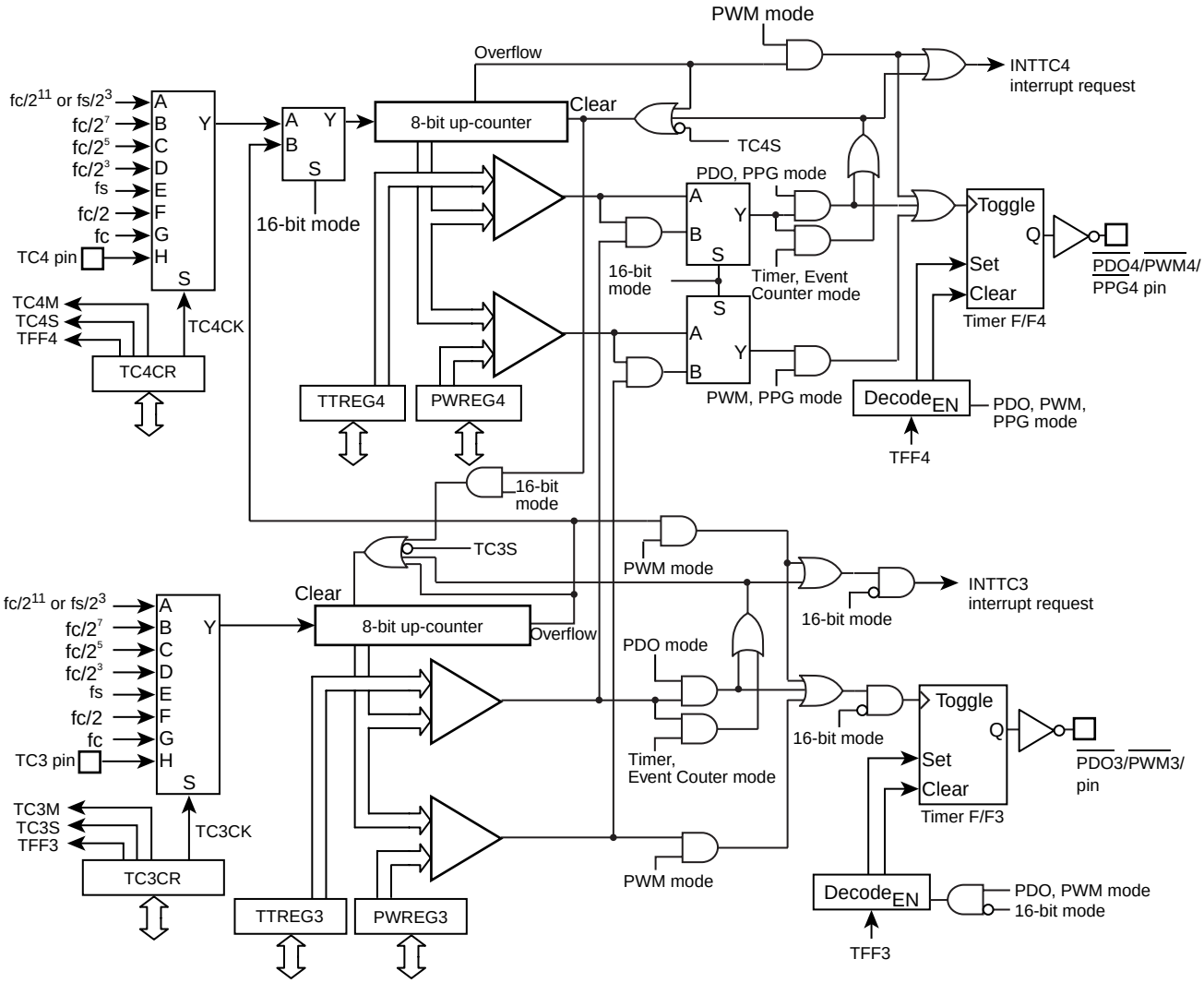


Figure 9-1 8-Bit TimerCounter 3, 4

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

	7	6	5	4	3	2	1	0	
TTREG3 (001CH) R/W									(Initial value: 1111 1111)
PWREG3 (0028H) R/W									(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 (0018H)	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

	7	6	5	4	3	2	1	0	
TTREG4 (001DH) R/W									(Initial value: 1111 1111)
PWREG4 (0029H) R/W									(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 (0019H)	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     (EIRH). 4           : 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.
```

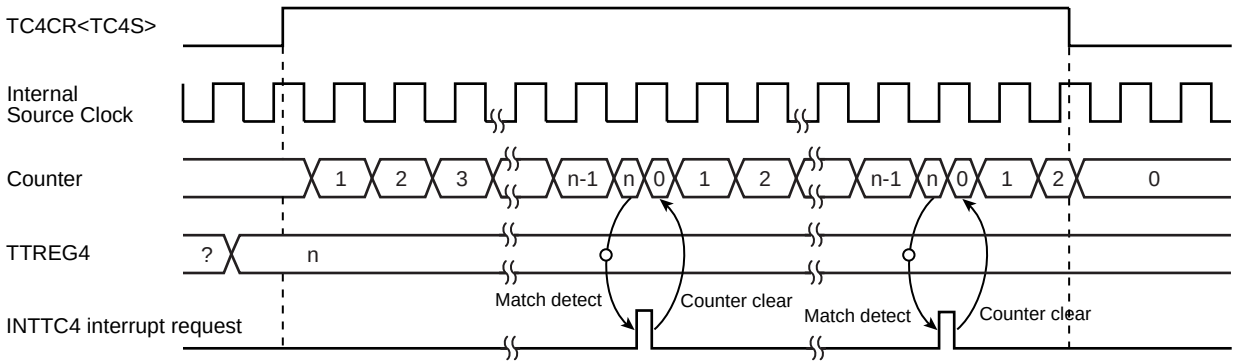


Figure 9-2 8-Bit Timer Mode Timing Chart (TC4)

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 $\overline{\text{PDOj, PWMj}}$ 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

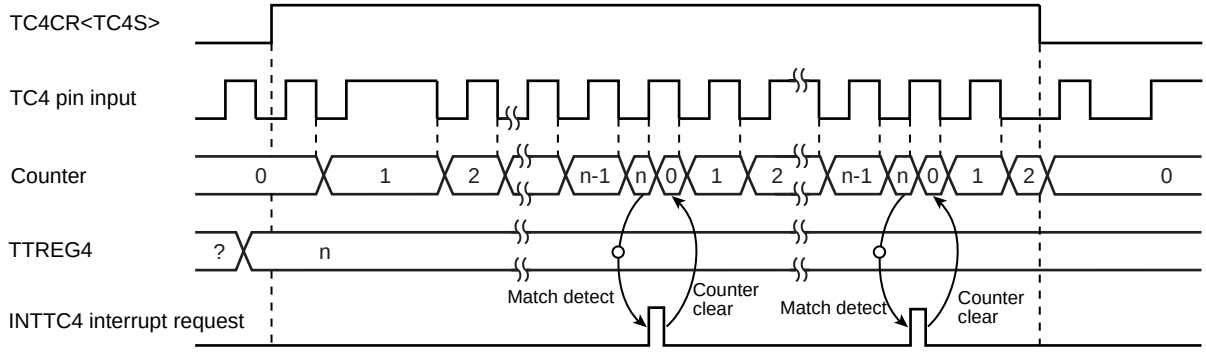


Figure 9-3 8-Bit Event Counter Mode Timing Chart (TC4)

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

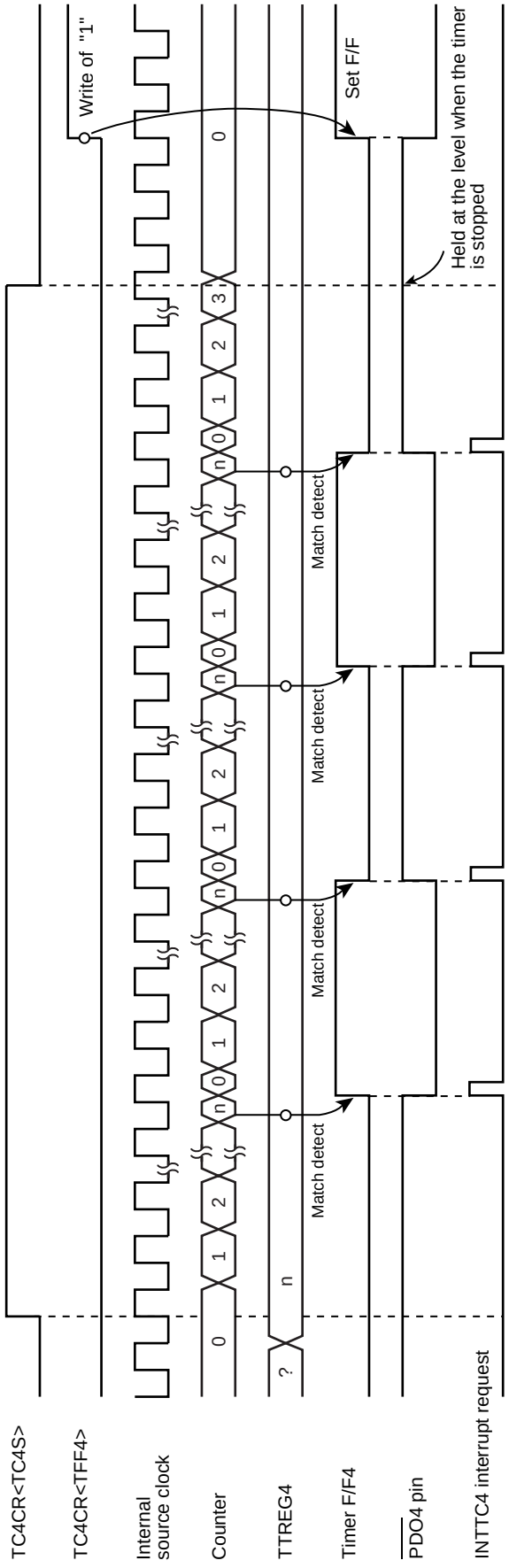


Figure 9-4 8-Bit PDO Mode Timing Chart (TC4)

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	–

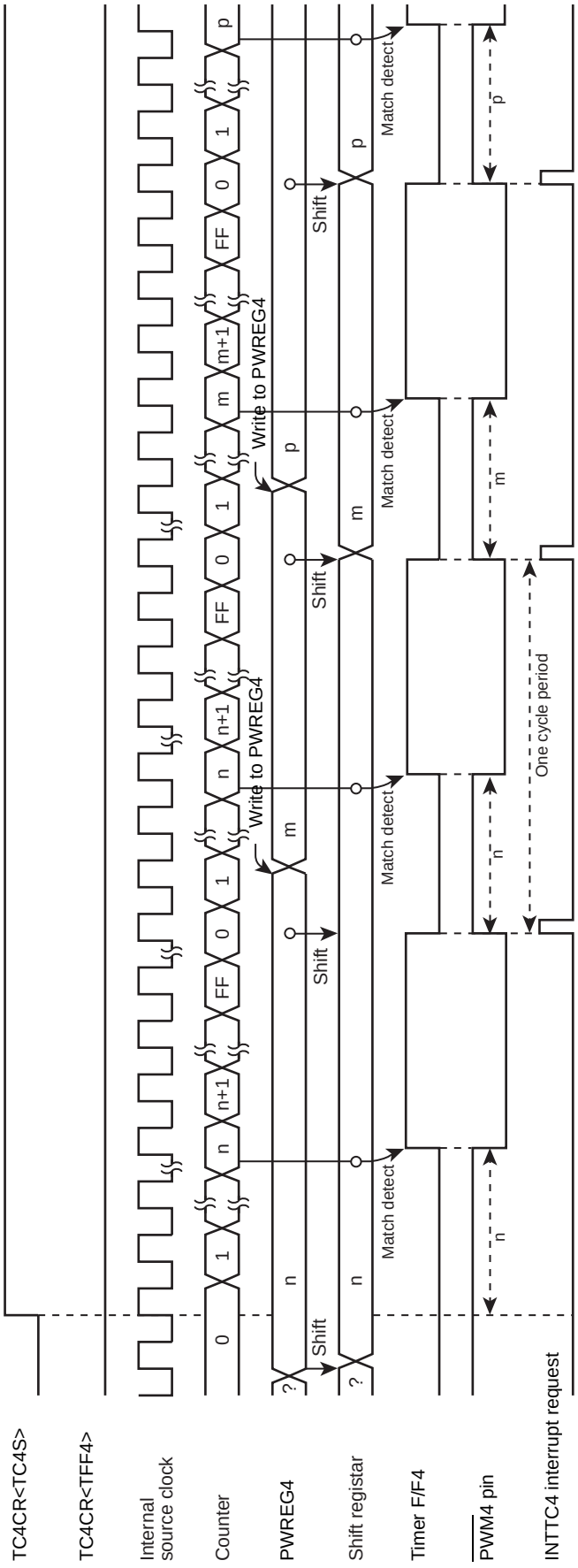


Figure 9-5 8-Bit PWM Mode Timing Chart (TC4)

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 $\overline{PDOj}$, $\overline{PWMj}$, and $\overline{PPGj}$ 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     (EIRH). 4           : 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.
```

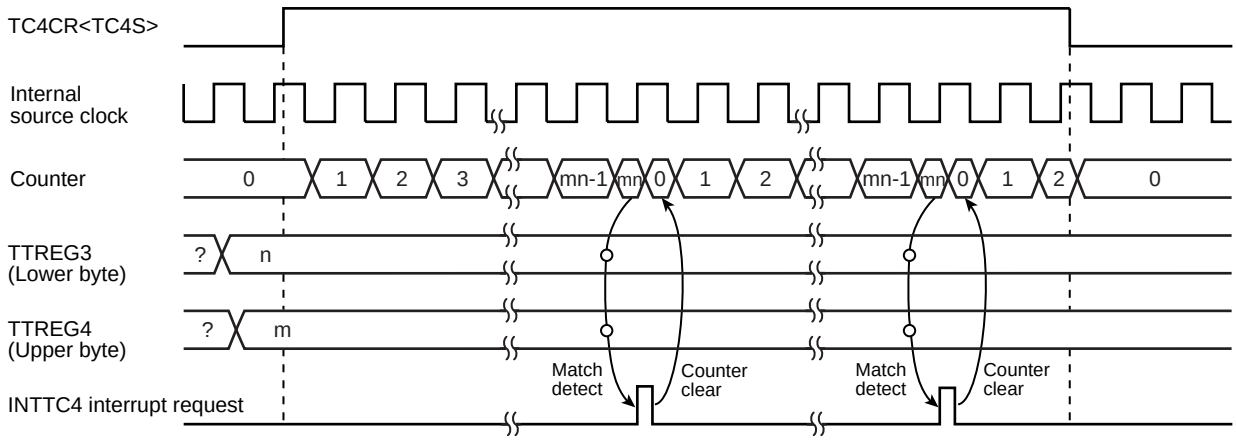


Figure 9-6 16-Bit Timer Mode Timing Chart (TC3 and TC4)

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 $\overline{PDOj}$, $\overline{PWMj}$ and $\overline{PPGj}$ pins may output pulses.
Note 2: In the event counter mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the event counter mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.
Note 3: j = 3, 4

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

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

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

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

Note 1: In the PWM mode, program the timer register PWREG4 and 3 immediately after the INTTC4 interrupt request is generated (normally in the INTTC4 interrupt service routine.) If the programming of PWREGj and the interrupt request occur at the same time, an unstable value is shifted, that may result in generation of pulse different from the programmed value until the next INTTC4 interrupt request is generated.
Note 2: When the timer is stopped during PWM output, the $\overline{PWM4}$ pin holds the output status when the timer is stopped. To change the output status, program TC4CR<TFF4> after the timer is stopped. Do not program TC4CR<TFF4> upon stopping of the timer.
Example: Fixing the $\overline{PWM4}$ pin to the high level when the TimerCounter is stopped

CLR (TC4CR).3: Stops the timer.
CLR (TC4CR).7 : Sets the $\overline{PWM4}$ pin to the high level.

Note 3: To enter the STOP mode, stop the timer and then enter the STOP mode. If the STOP mode is entered without stopping of the timer when fc, fc/2 or fs is selected as the source clock, a pulse is output from the $\overline{PWM4}$ 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.
```

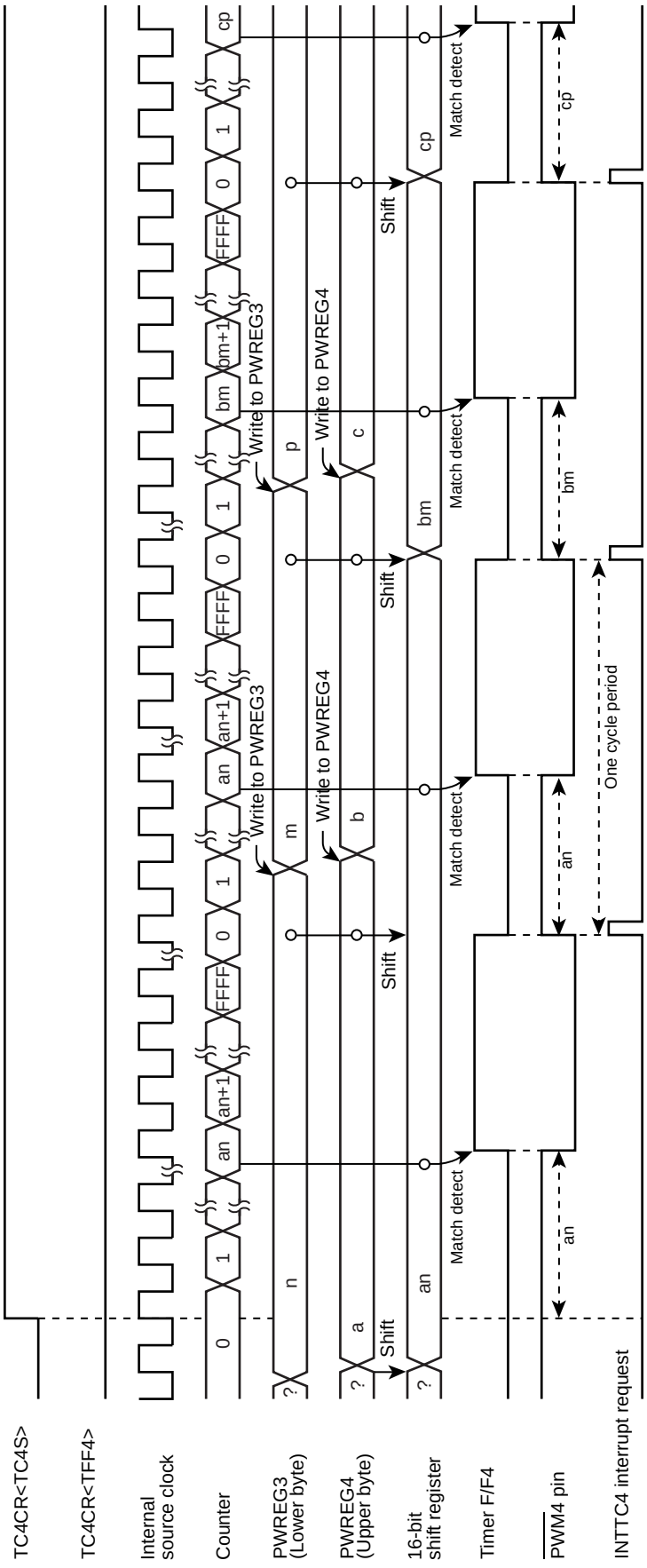


Figure 9-7 16-Bit PWM Mode Timing Chart (TC3 and TC4)

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 $\overline{PPG}4$ 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 $\overline{PPG}4$ 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 $\overline{PPG}4$ pin to the high level when the TimerCounter is stopped

```
CLR  (TC4CR).3: Stops the timer
CLR  (TC4CR).7: Sets the PPG4 pin to the high level
```

Note 3: i = 3, 4

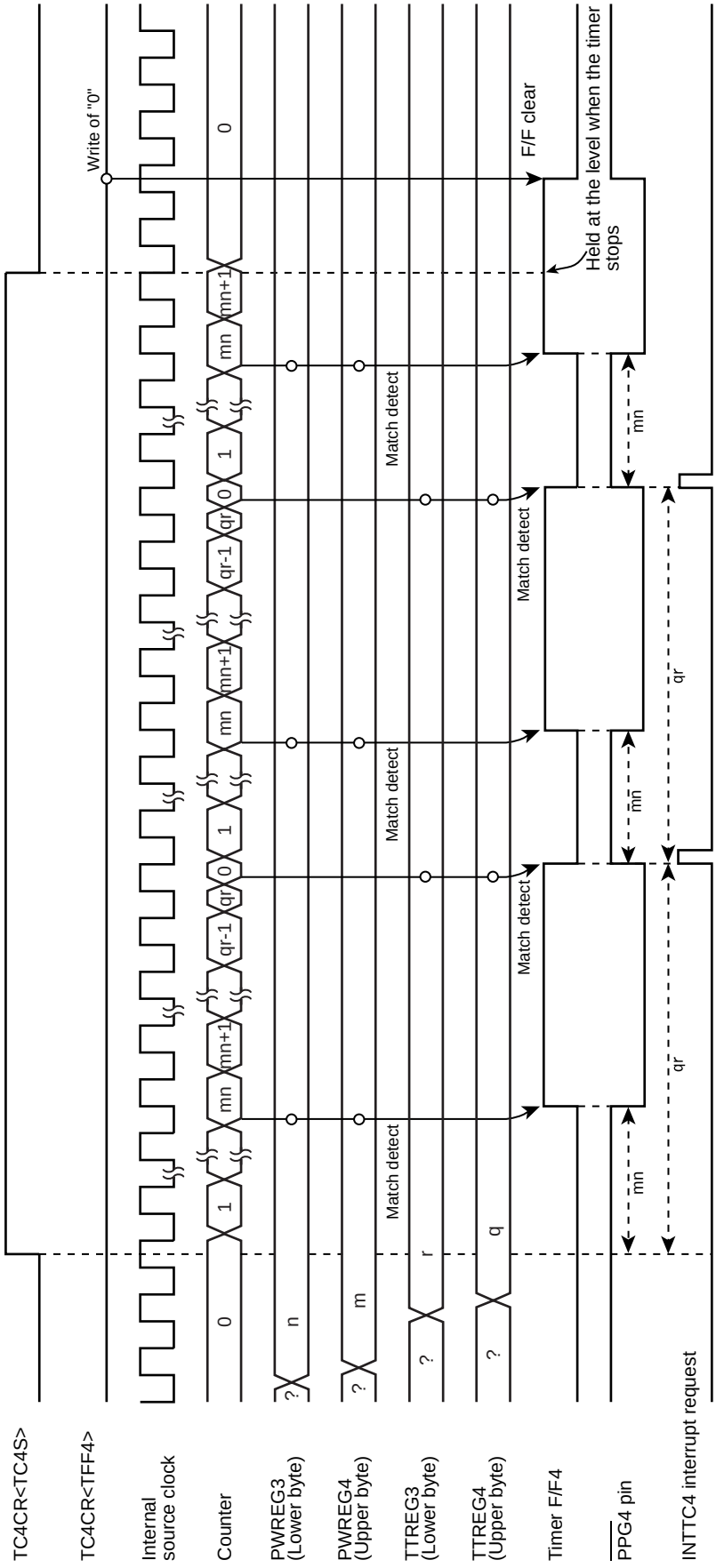


Figure 9-8 16-Bit PPG Mode Timing Chart (TC3 and TC4)

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     (EIRH). 4       : 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     (EIRH). 4           : 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

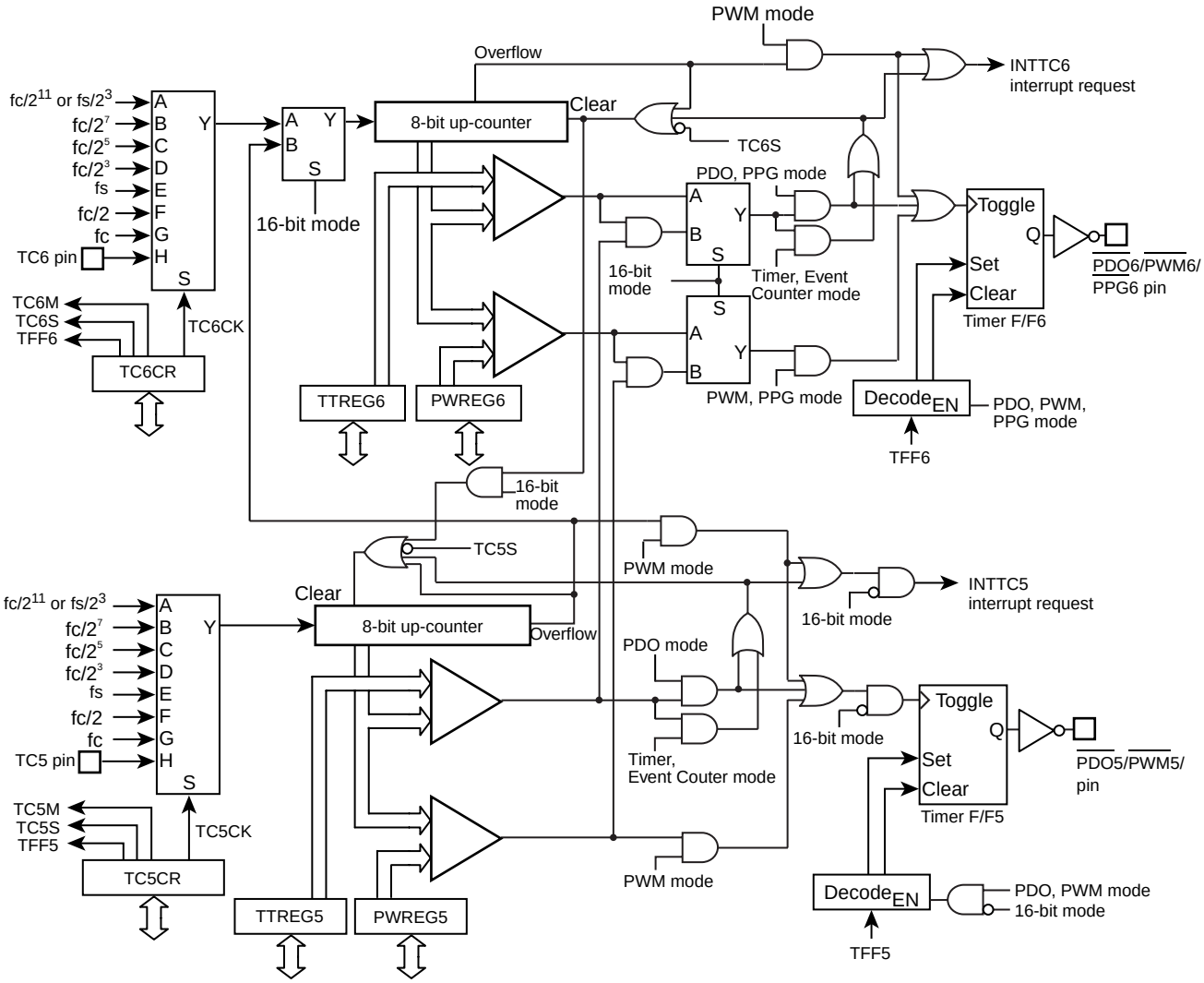


Figure 10-1 8-Bit TimerCounter 5, 6

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

	7	6	5	4	3	2	1	0	
TTREG5 (001EH) R/W									(Initial value: 1111 1111)
PWREG5 (002AH) 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 (001AH)	7	6	5	4	3	2	1	0	
	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

	7	6	5	4	3	2	1	0	
TTREG6 (001FH) R/W									(Initial value: 1111 1111)
PWREG6 (002BH) R/W									(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 (001BH)	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.

Note 9: To use the PDO, PWM or PPG mode, a pulse is not output from the timer output pin when TC1CR2<TC6OUT> is set to 1. To output a pulse from the timer output pin, clear TC1CR2<TC6OUT> to 0.

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     (EIRH). 5              : 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.
```

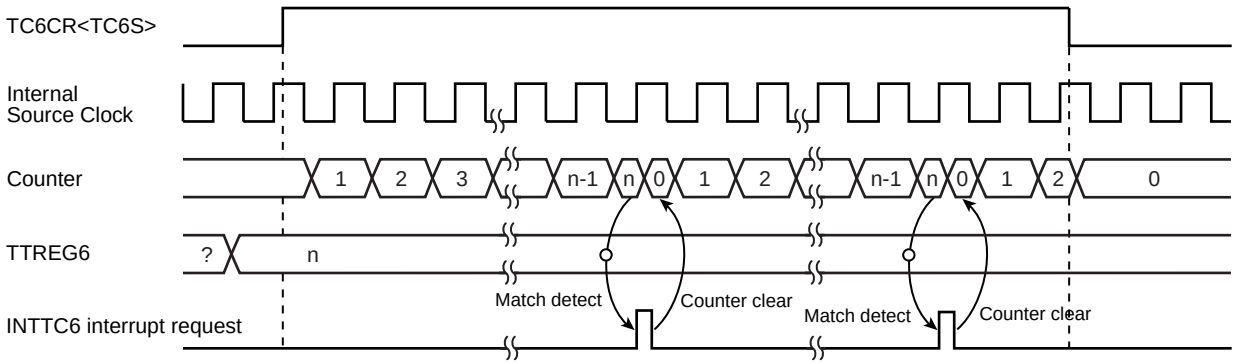


Figure 10-2 8-Bit Timer Mode Timing Chart (TC6)

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 $\overline{\text{PDOj, PWMj}}$ 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

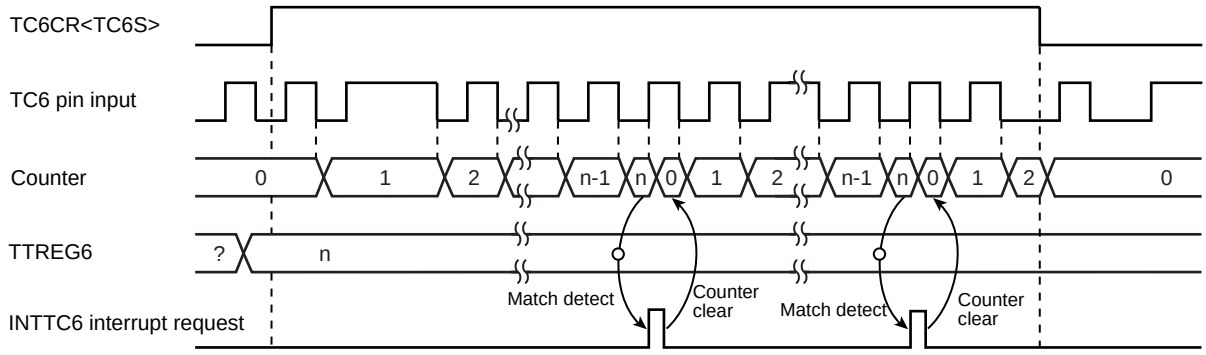


Figure 10-3 8-Bit Event Counter Mode Timing Chart (TC6)

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÷2^7/fc÷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

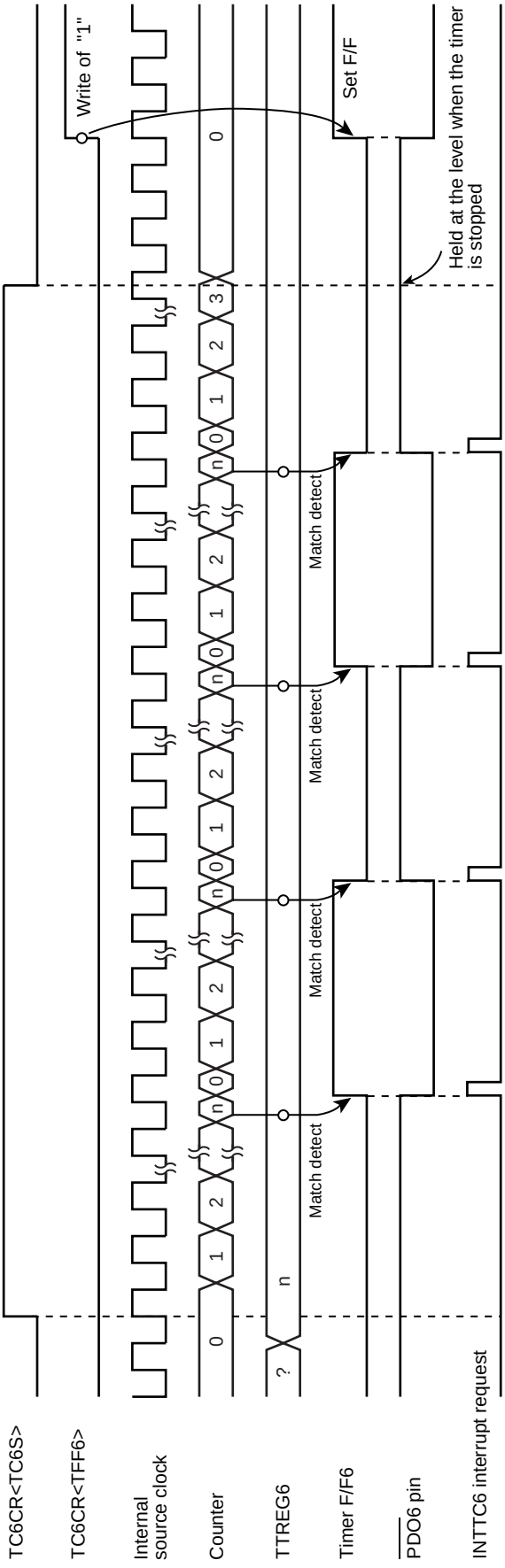


Figure 10-4 8-Bit PDO Mode Timing Chart (TC6)

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	–

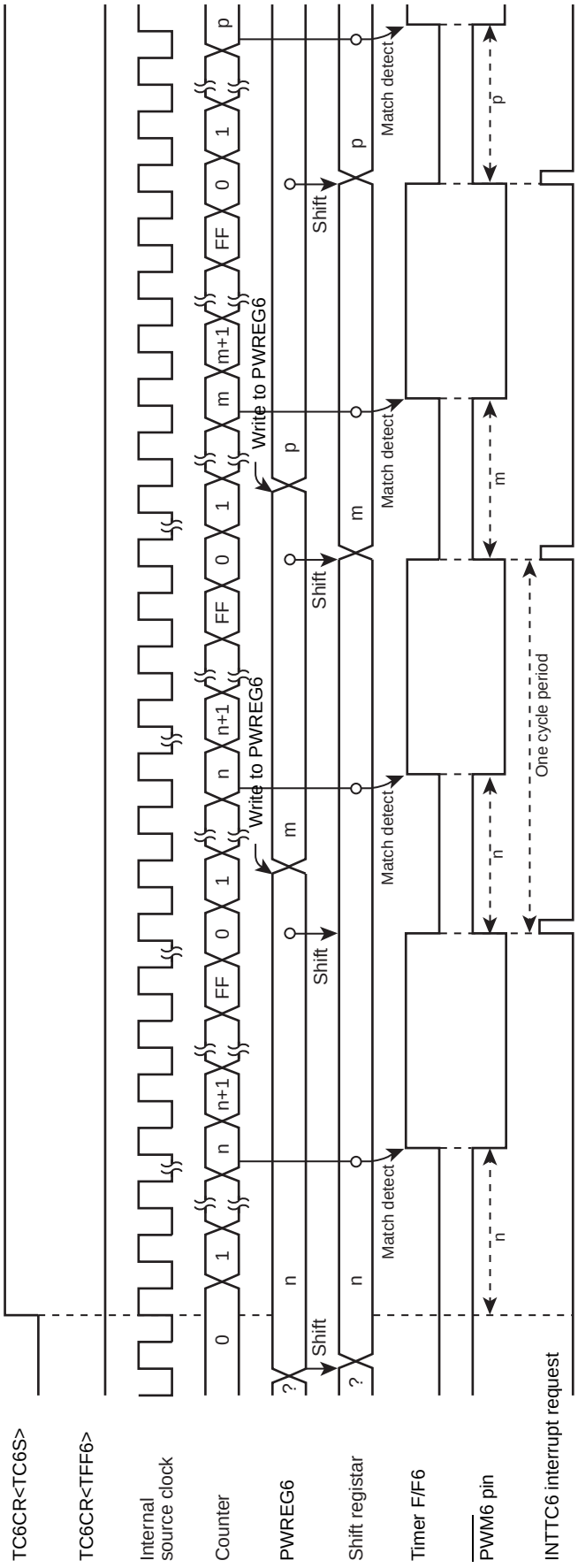


Figure 10-5 8-Bit PWM Mode Timing Chart (TC6)

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 $\overline{PDOj}$, $\overline{PWMj}$, and $\overline{PPGj}$ 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     (EIRH). 5           : 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.
```

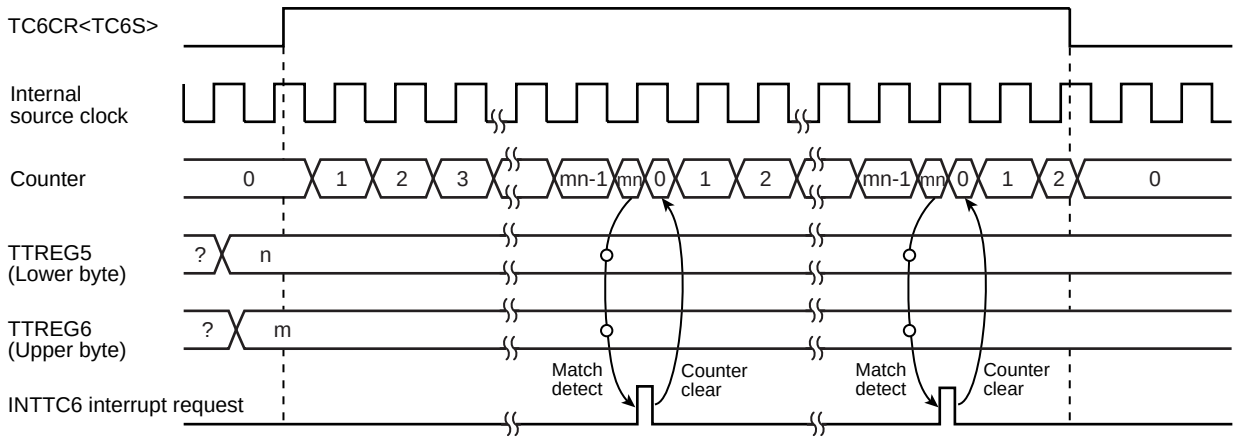


Figure 10-6 16-Bit Timer Mode Timing Chart (TC5 and TC6)

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 $\overline{PDOj}$, $\overline{PWMj}$ and $\overline{PPGj}$ pins may output pulses.
Note 2: In the event counter mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the event counter mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.
Note 3: j = 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 $\overline{PWM6}$ 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 $\overline{PWM6}$ 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 $\overline{PWM6}$ pin to the high level when the TimerCounter is stopped

CLR (TC6CR).3: Stops the timer.
CLR (TC6CR).7 : Sets the $\overline{PWM6}$ 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 $\overline{PWM6}$ 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.
```

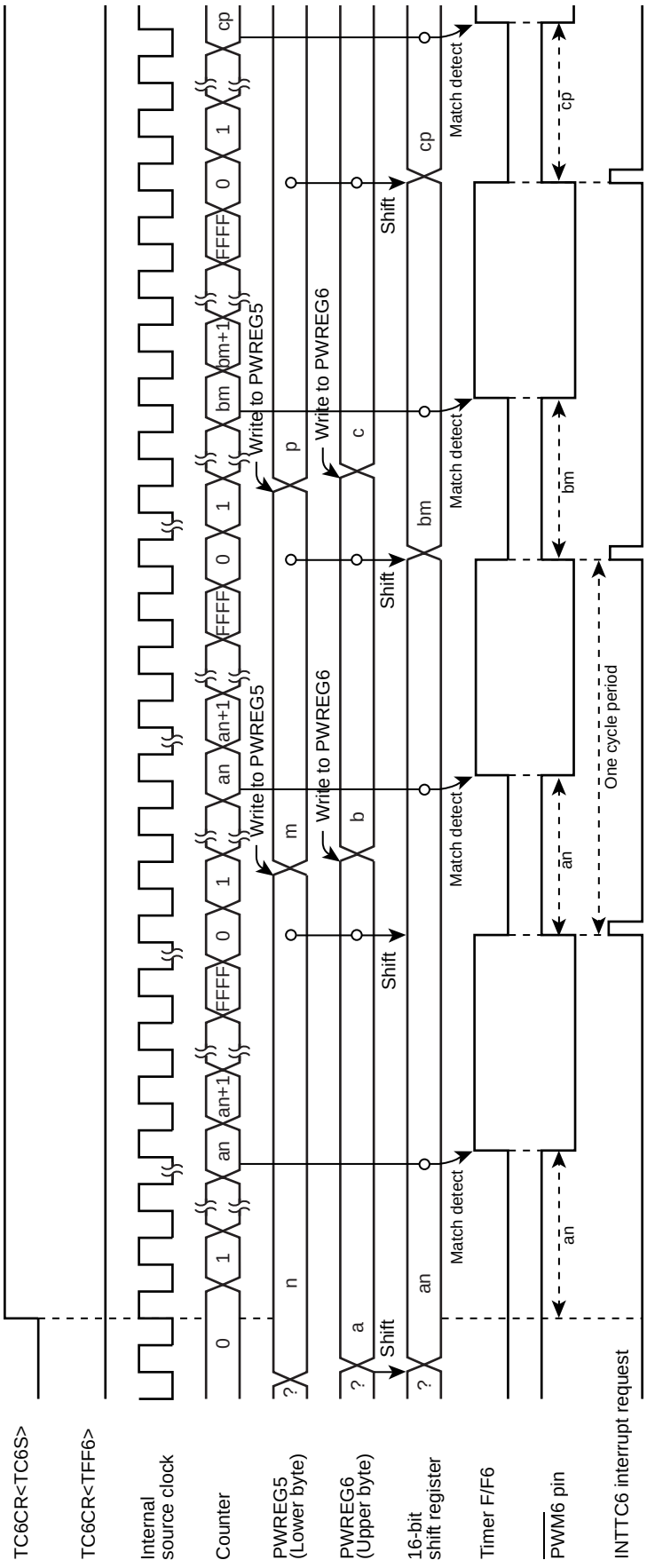


Figure 10-7 16-Bit PWM Mode Timing Chart (TC5 and TC6)

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{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{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{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

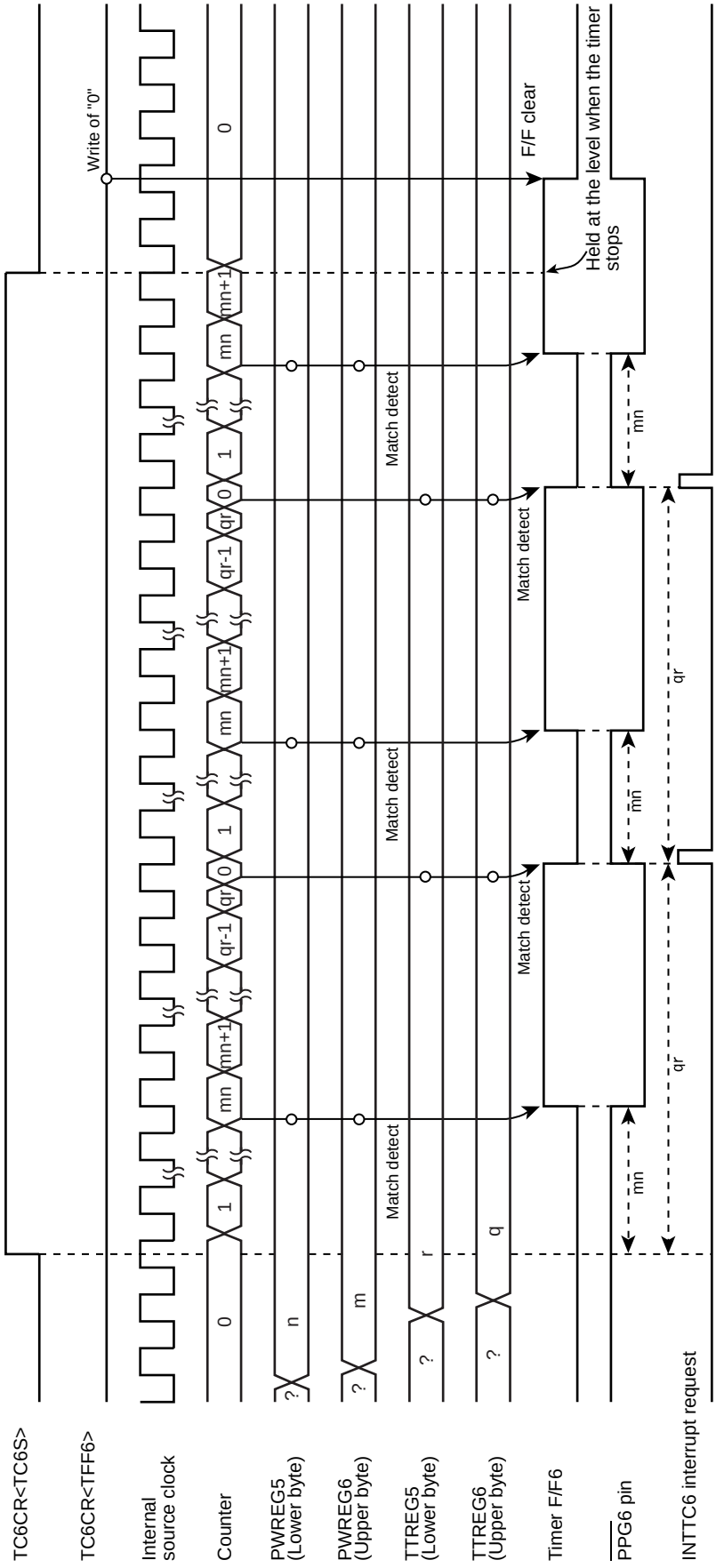


Figure 10-8 16-Bit PPG Mode Timing Chart (TC5 and TC6)

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     (EIRH). 5       : 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     (EIRH). 5           : 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. Asynchronous Serial interface (UART)

11.1 Configuration

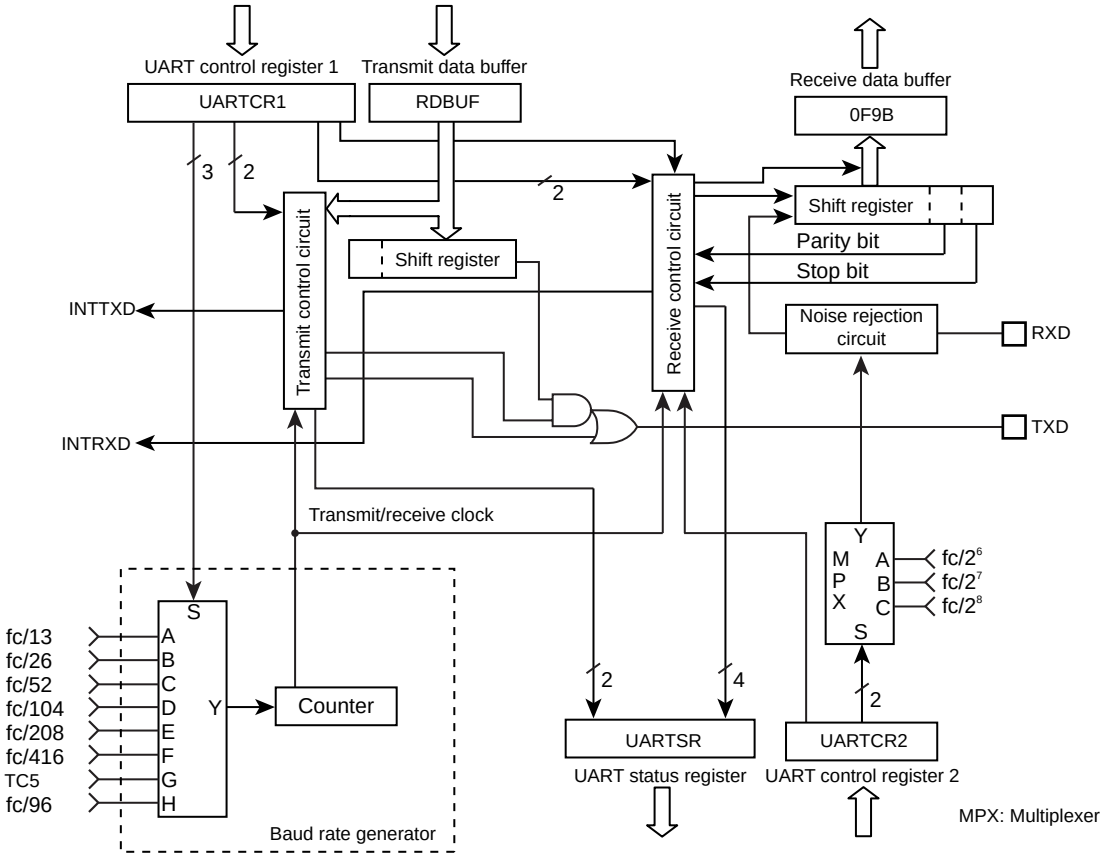


Figure 11-1 UART (Asynchronous Serial Interface)

11.2 Control

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

UART Control Register1

UARTCR1 (INTTC5H)

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: TTREG5 (Input TC5) 111: fc/96	

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

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

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

UART Control Register2

UARTCR2 (0025H)

7	6	5	4	3	2	1	0
					RXDNC	STOPBR	

(Initial value: **** *000)

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

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

UART Status Register

UARTSR (0026H)

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.

UART Receive Data Buffer

0F9B (0F9BH)

7	6	5	4	3	2	1	0

Read only

(Initial value: 0000 0000)

UART Transmit Data Buffer

RDBUF (0025H)

7	6	5	4	3	2	1	0

Write only

(Initial value: 0000 0000)

11.3 Transfer Data Format

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

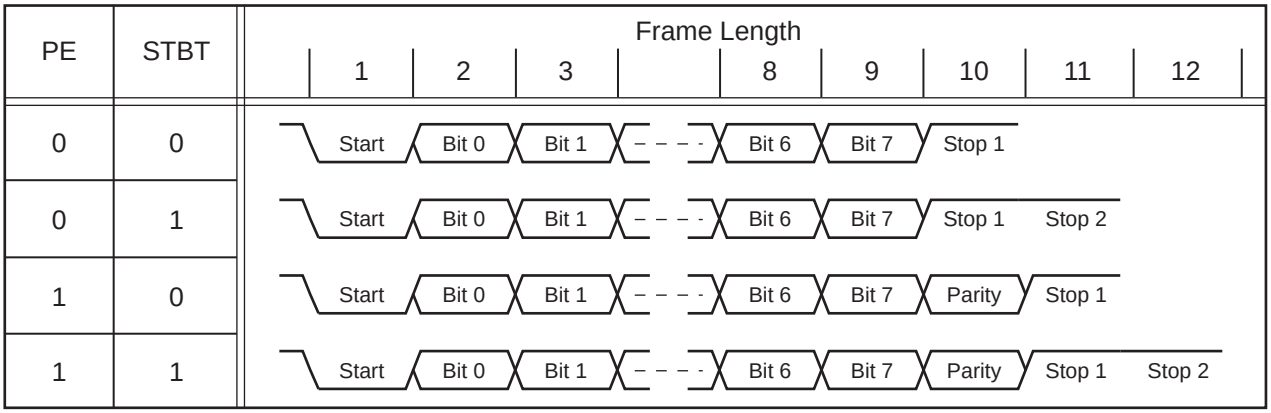


Figure 11-2 Transfer Data Format

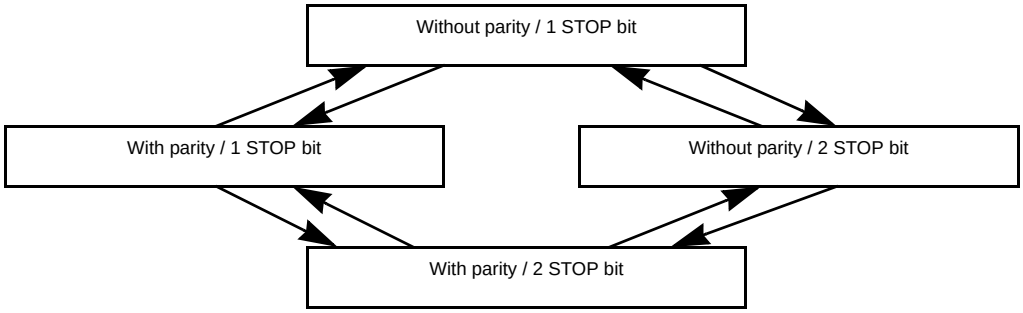


Figure 11-3 Caution on Changing Transfer Data Format

Note: In order to switch the transfer data format, perform transmit operations in the above Figure 11-3 sequence except for the initial setting.

11.4 Transfer Rate

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

Table 11-1 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 TTREG5 is used as the UART transfer rate (when UARTCR1<BRG> = “110”), the transfer clock and transfer rate are determined as follows:

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

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

11.5 Data Sampling Method

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

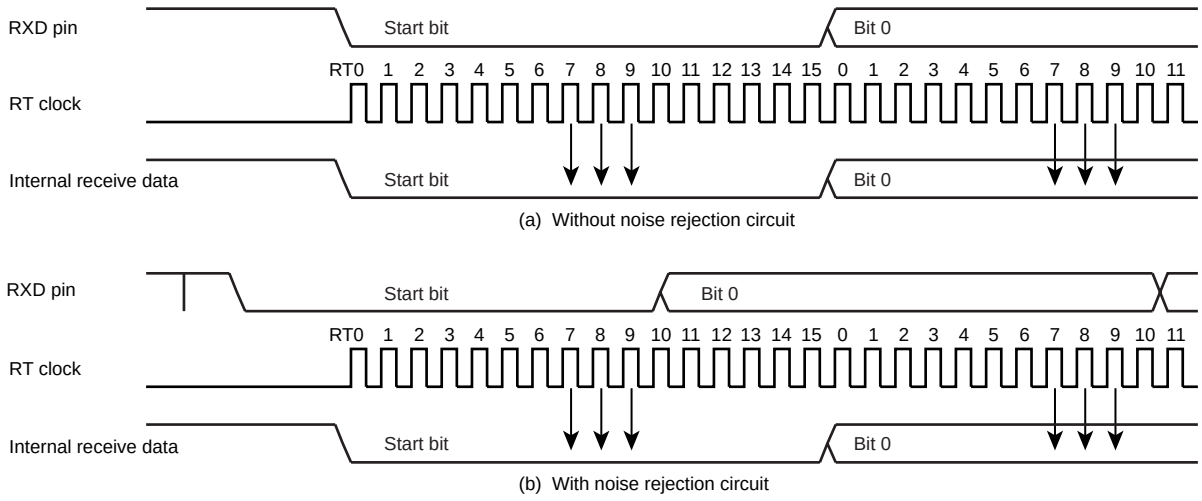


Figure 11-4 Data Sampling Method

11.6 STOP Bit Length

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

11.7 Parity

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

11.8 Transmit/Receive Operation

11.8.1 Data Transmit Operation

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

While UARTCR1<TXE> = “0” and from when “1” is written to UARTCR1<TXE> to when send data are written to RDBUF, the TXD pin is fixed at high level.
When transmitting data, first read UARTSR, then write data in RDBUF. Otherwise, UARTSR<TBEP> is not zero-cleared and transmit does not start.

11.8.2 Data Receive Operation

Set UARTCR1<RXE> to “1”. When data are received via the RXD pin, the receive data are transferred to 0F9B (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 0F9B (Receive data buffer). Then the receive buffer full flag UARTSR<RBFL> is set and an INTRXD interrupt is generated. Select the data transfer baud rate using UARTCR1<BRG>.

If an overrun error (OERR) occurs when data are received, the data are not transferred to 0F9B (Receive data buffer) but discarded; data in the 0F9B are not affected.

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

11.9 Status Flag

11.9.1 Parity Error

When parity determined using the receive data bits differs from the received parity bit, the parity error flag UARTSR<PERR> is set to “1”. The UARTSR<PERR> is cleared to “0” when the 0F9B is read after reading the UARTSR.

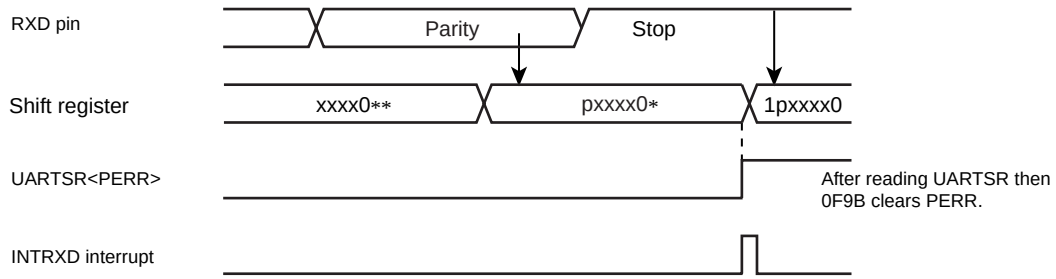


Figure 11-5 Generation of Parity Error

11.9.2 Framing Error

When “0” is sampled as the stop bit in the receive data, framing error flag UARTSR<FERR> is set to “1”. The UARTSR<FERR> is cleared to “0” when the 0F9B is read after reading the UARTSR.

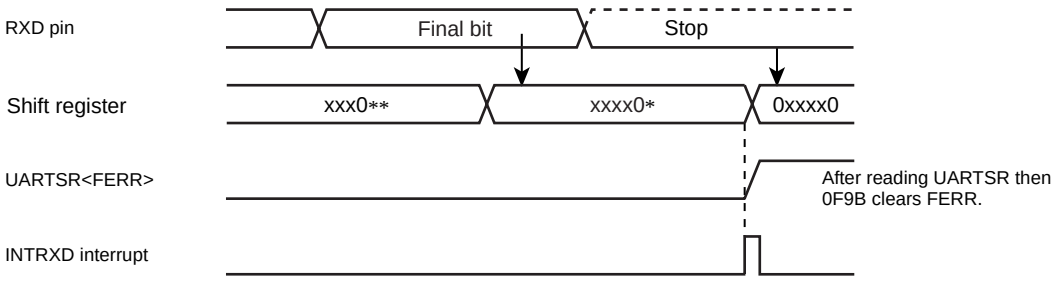


Figure 11-6 Generation of Framing Error

11.9.3 Overrun Error

When all bits in the next data are received while unread data are still in 0F9B, overrun error flag UARTSR<OERR> is set to “1”. In this case, the receive data is discarded; data in 0F9B are not affected. The UARTSR<OERR> is cleared to “0” when the 0F9B is read after reading the UARTSR.

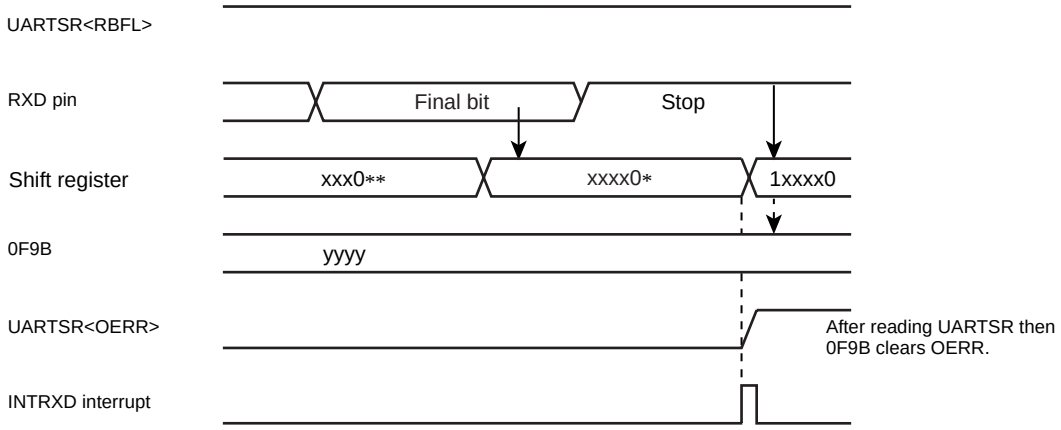


Figure 11-7 Generation of Overrun Error

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

11.9.4 Receive Data Buffer Full

Loading the received data in 0F9B sets receive data buffer full flag UARTSR<RBFL> to "1". The UARTSR<RBFL> is cleared to “0” when the 0F9B is read after reading the UARTSR.

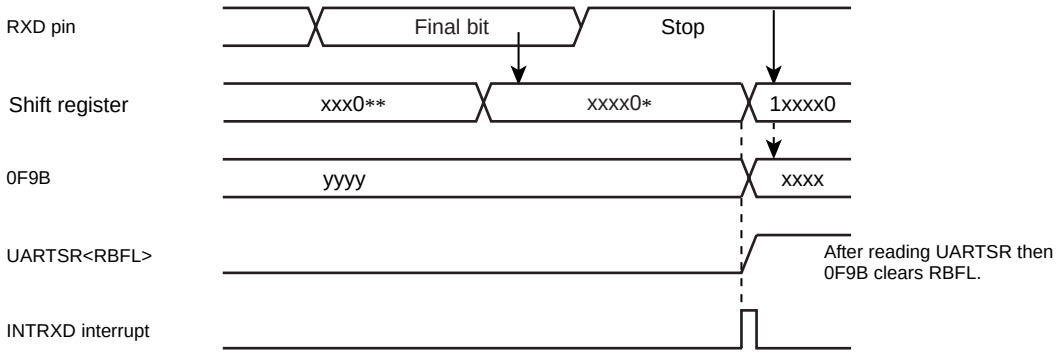


Figure 11-8 Generation of Receive Data Buffer Full

Note: If the overrun error flag UARTSR<OERR> is set during the period between reading the UARTSR and reading the 0F9B, it cannot be cleared by only reading the 0F9B. Therefore, after reading the 0F9B, read the UARTSR again to check whether or not the overrun error flag which should have been cleared still remains set.

11.9.5 Transmit Data Buffer Empty

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

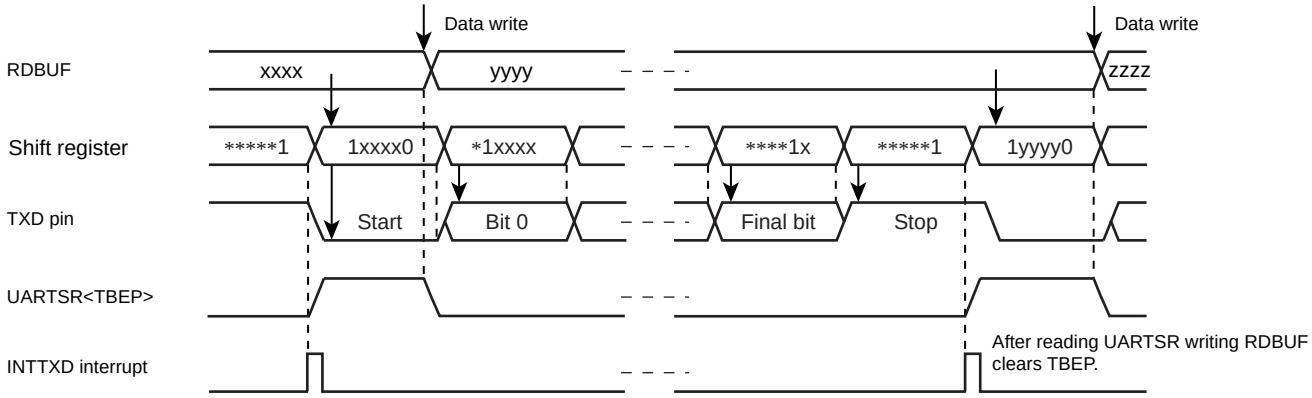


Figure 11-9 Generation of Transmit Data Buffer Empty

11.9.6 Transmit End Flag

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

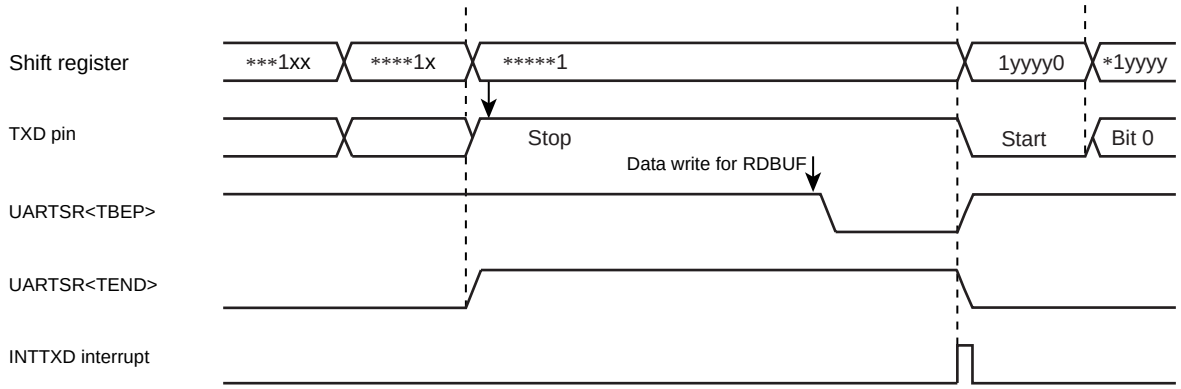


Figure 11-10 Generation of Transmit End Flag and Transmit Data Buffer Empty

12. Synchronous Serial Interface (SIO)

The TMP86PM23UG 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.

12.1 Configuration

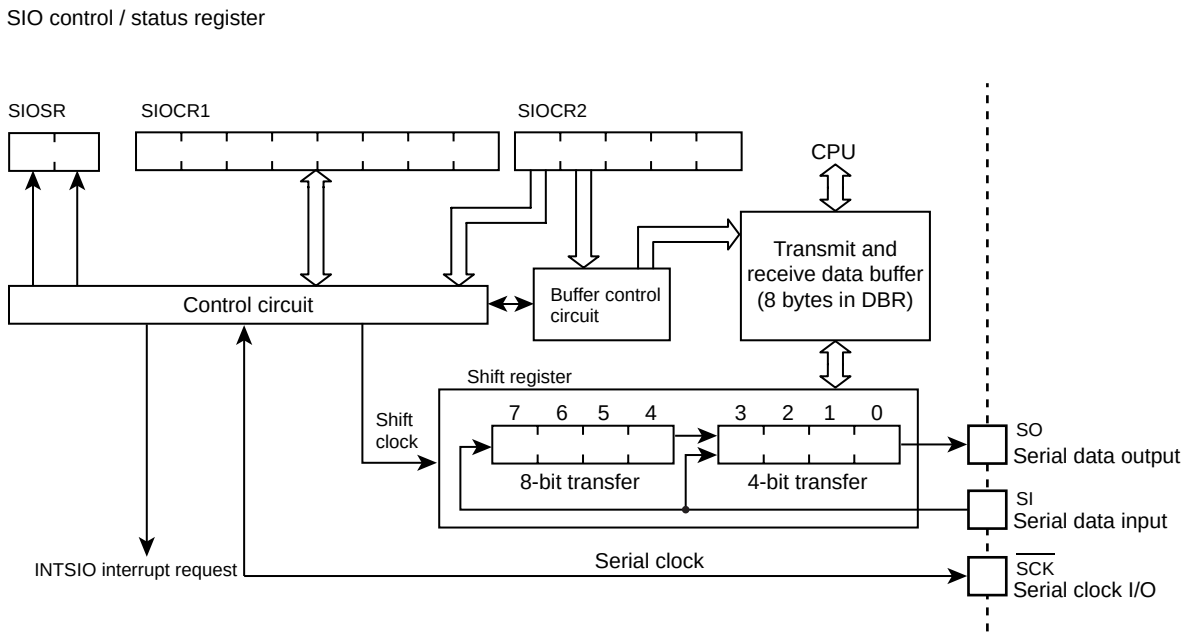


Figure 12-1 Serial Interface

12.2 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 0F90H to 0F97H 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

SIOCR1	7	6	5	4	3	2	1	0	
(0F98H)	SIOS	SIOINH	SIOM			SCK			(Initial value: 0000 0000)

SIOS	Indicate transfer start / stop	0: Stop 1: Start		Write only
SIOINH	Continue / abort transfer	0: Continuously transfer 1: Abort transfer (Automatically cleared after abort)		
SIOM	Transfer mode select	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		
SCK	Serial clock select		NORMAL1/2, IDLE1/2 mode: DV7CK = 0 / DV7CK = 1; SLOW1/2 SLEEP1/2 mode	Write only
		000	$fc/2^{13}$ / $fs/2^5$ / $fs/2^5$	
		001	$fc/2^8$ / $fc/2^8$ / -	
		010	$fc/2^7$ / $fc/2^7$ / -	
		011	$fc/2^6$ / $fc/2^6$ / -	
		100	$fc/2^5$ / $fc/2^5$ / -	
		101	$fc/2^4$ / $fc/2^4$ / -	
		110	Reserved	
		111	External clock (Input from SCK pin)	

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

SIOCR2	7	6	5	4	3	2	1	0	
(0F99H)				WAIT		BUF			(Initial value: ***0 0000)

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 0F90H 001: 2 words transfer 0F90H ~ 0F91H 010: 3 words transfer 0F90H ~ 0F92H 011: 4 words transfer 0F90H ~ 0F93H 100: 5 words transfer 0F90H ~ 0F94H 101: 6 words transfer 0F90H ~ 0F95H 110: 7 words transfer 0F90H ~ 0F96H 111: 8 words transfer 0F90H ~ 0F97H	

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 0F90H).

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
(0F99H)	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".

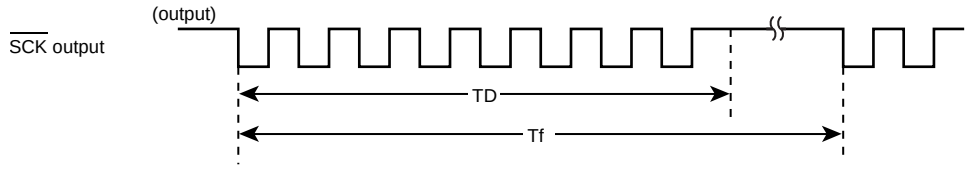


Figure 12-2 Frame time (T_f) and Data transfer time (T_D)

12.3 Serial clock

12.3.1 Clock source

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

12.3.1.1 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 12-1 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)

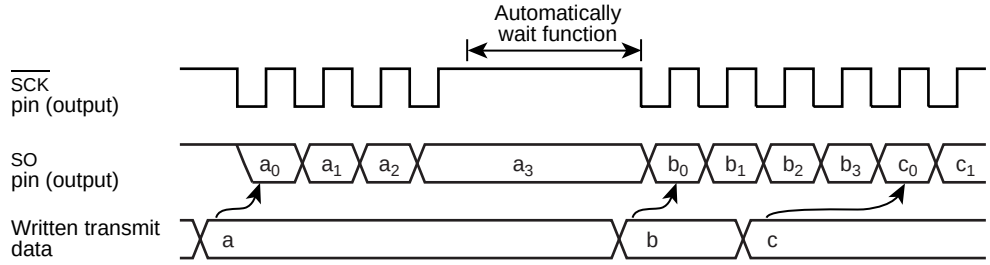


Figure 12-3 Automatic Wait Function (at 4-bit transmit mode)

12.3.1.2 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).

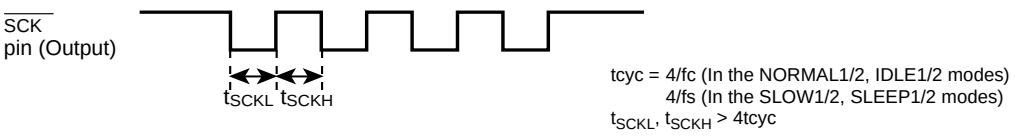


Figure 12-4 External clock pulse width

12.3.2 Shift edge

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

12.3.2.1 Leading edge

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

12.3.2.2 Trailing edge

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

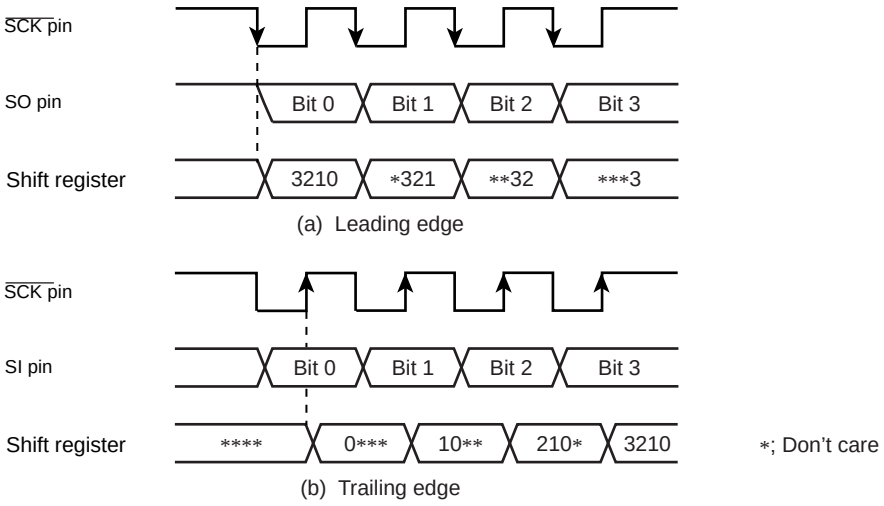


Figure 12-5 Shift edge

12.4 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).

12.5 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.

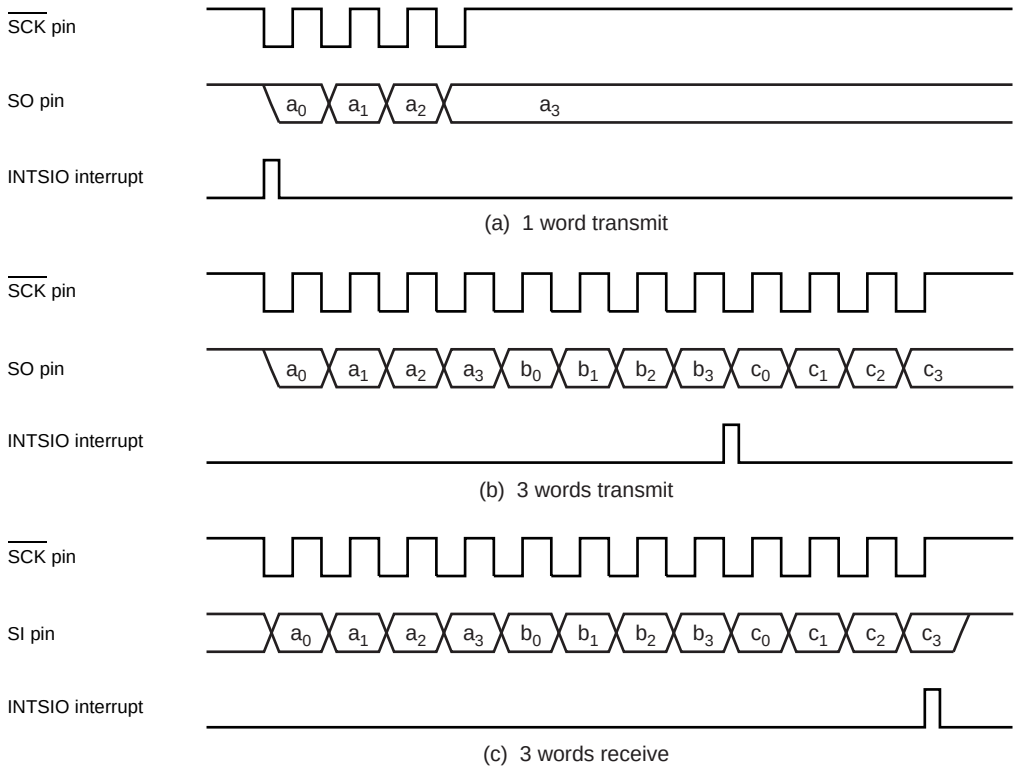


Figure 12-6 Number of words to transfer (Example: 1word = 4bit)

12.6 Transfer Mode

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

12.6.1 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”.

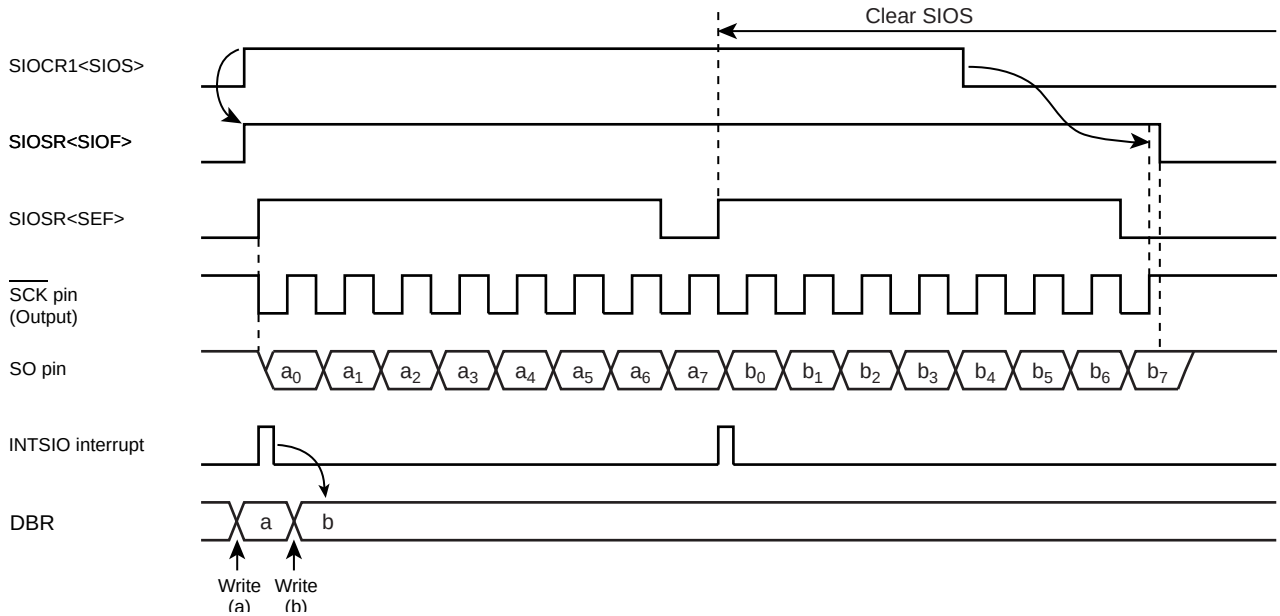


Figure 12-7 Transfer Mode (Example: 8bit, 1word transfer, Internal clock)

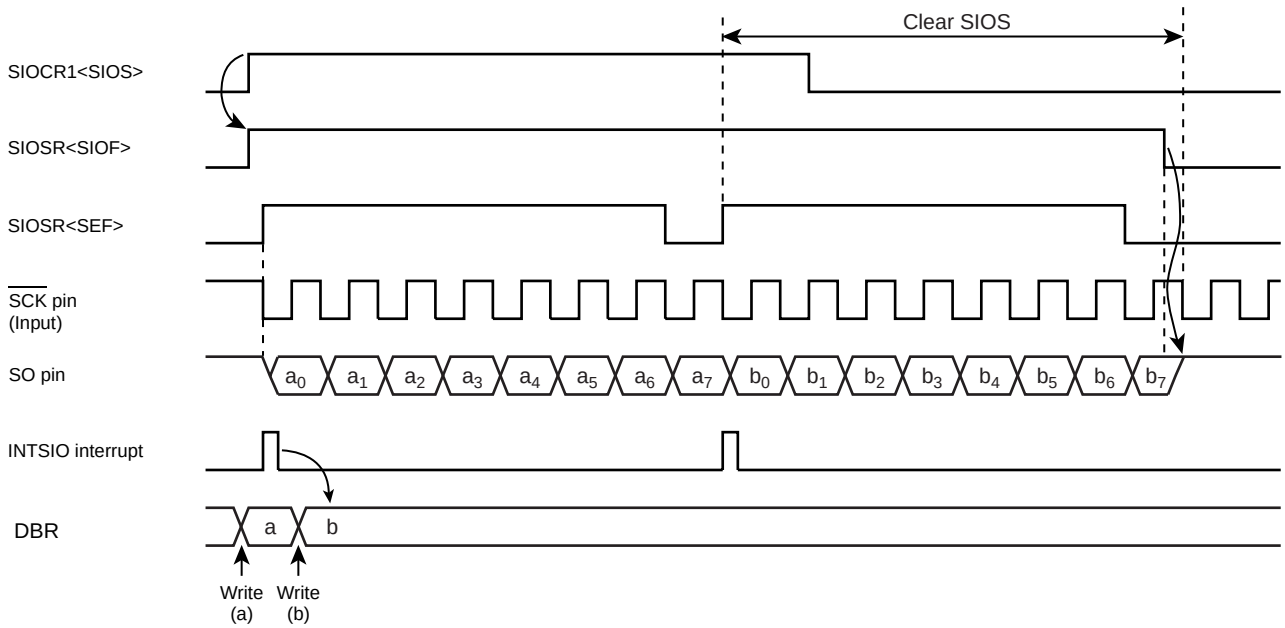


Figure 12-8 Transfer Mode (Example: 8bit, 1word transfer, External clock)

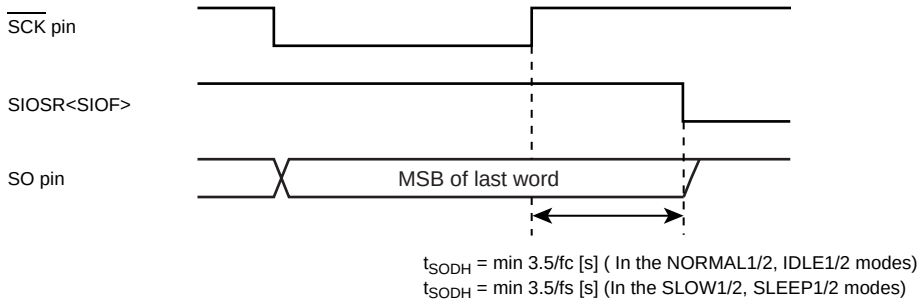


Figure 12-9 Transmiiied Data Hold Time at End of Transfer

12.6.2 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.

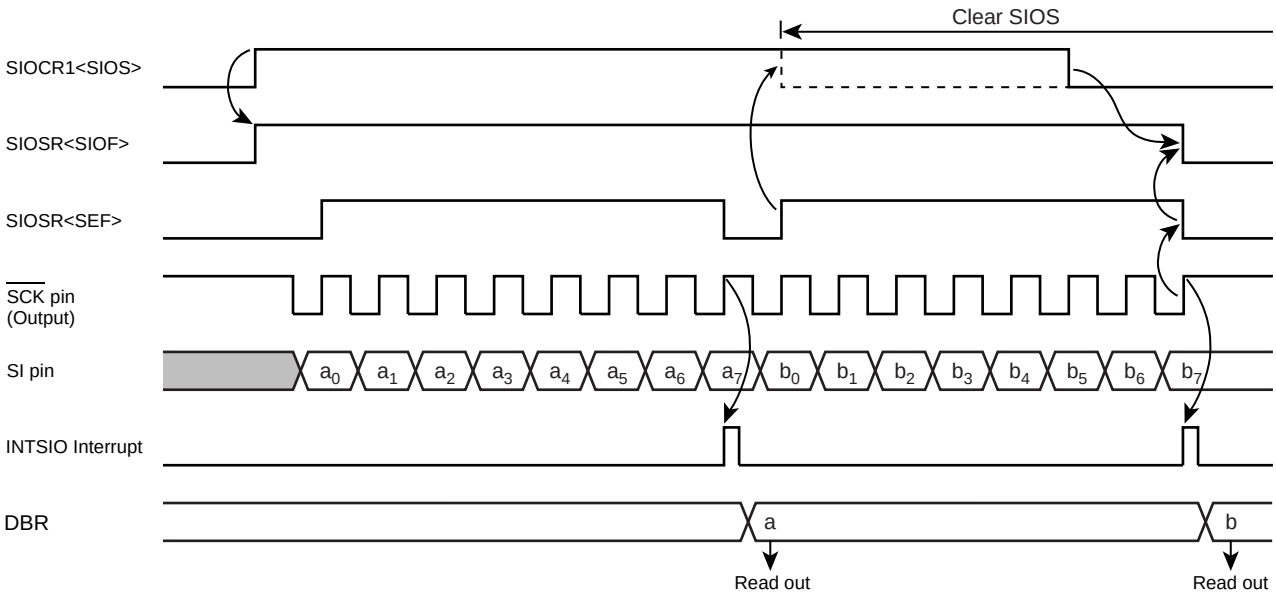


Figure 12-10 Receive Mode (Example: 8bit, 1word transfer, Internal clock)

12.6.3 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.

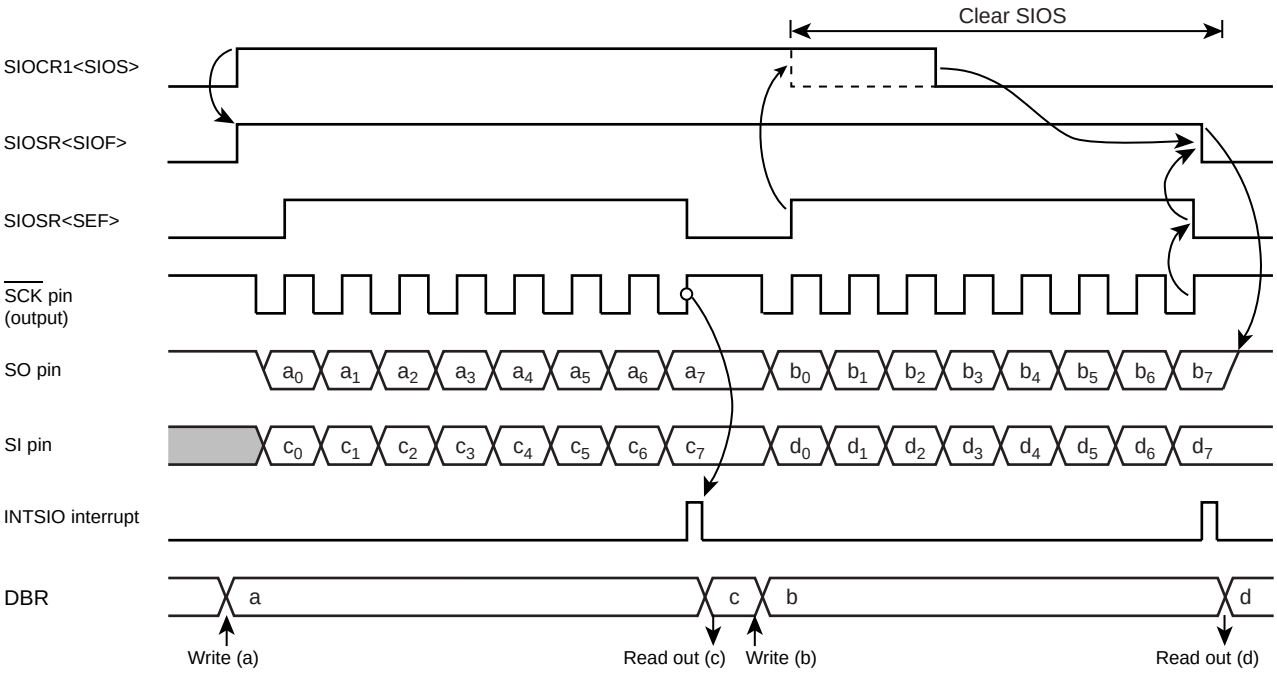


Figure 12-11 Transfer / Receive Mode (Example: 8bit, 1word transfer, Internal clock)

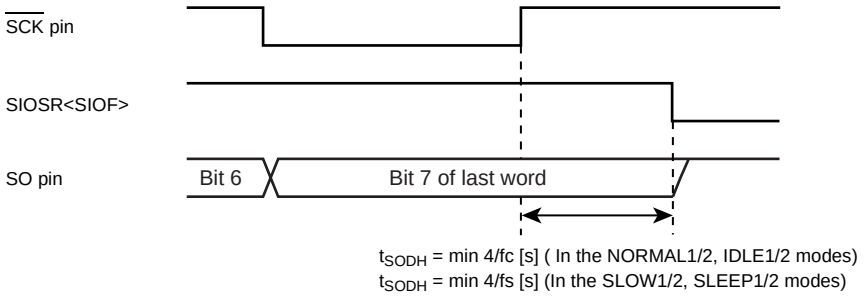


Figure 12-12 Transmitted Data Hold Time at End of Transfer / Receive

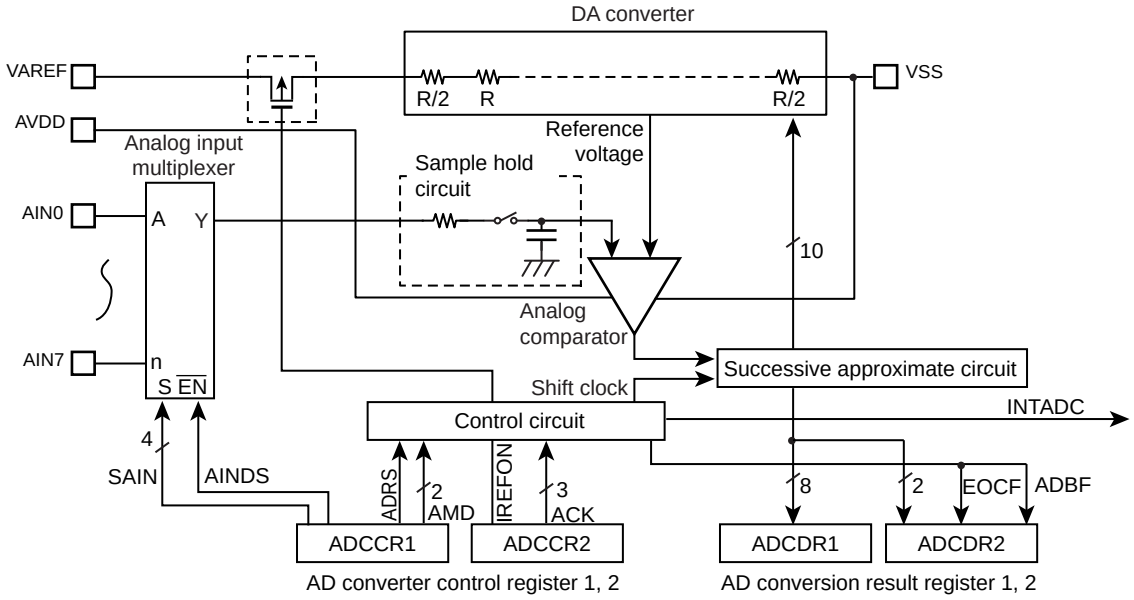
13. 10-bit AD Converter (ADC)

The TMP86PM23UG have a 10-bit successive approximation type AD converter.

13.1 Configuration

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

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



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

Figure 13-1 10-bit AD Converter

13.2 Register configuration

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCR1)

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

2. AD converter control register 2 (ADCCR2)

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

3. AD converted value register 1 (ADCDR1)

   This register used to store the digital value 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 (000EH)

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 (000FH)

7	6	5	4	3	2	1	0	
		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 13-1 ACK setting and Conversion time

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

Note 1: Setting for "–" in the above table are inhibited. fc: High Frequency oscillation clock [Hz]

Note 2: Set conversion time setting should be kept more than the following time by 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
- VAREF = 1.8 to 5.5 V 124.8 μs and more

AD Converted value Register 1

ADCDR1 (0021H)

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

AD Converted value Register 2

ADCDR2 (0020H)

7	6	5	4	3	2	1	0	
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.

13.3 Function

13.3.1 Software Start Mode

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

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

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

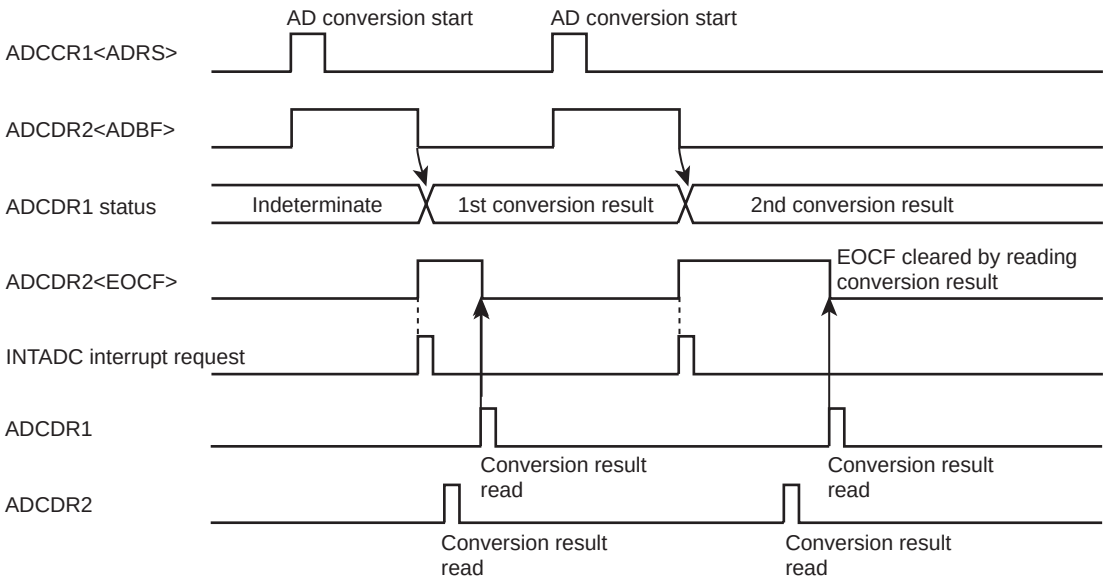


Figure 13-2 Software Start Mode

13.3.2 Repeat Mode

AD conversion of the voltage at the analog input pin specified by ADCCR1<SAIN> is performed repeatedly. In this mode, AD conversion is started by setting ADCCR1<ADRS> to “1” after setting ADCCR1<AMD> to “11” (Repeat mode).

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

In repeat mode, each time one AD conversion is completed, the next AD conversion is started. To stop AD conversion, set ADCCR1<AMD> to “00” (Disable mode) by writing 0s. The AD convert operation is stopped immediately. The converted value at this time is not stored in the AD converted value register.

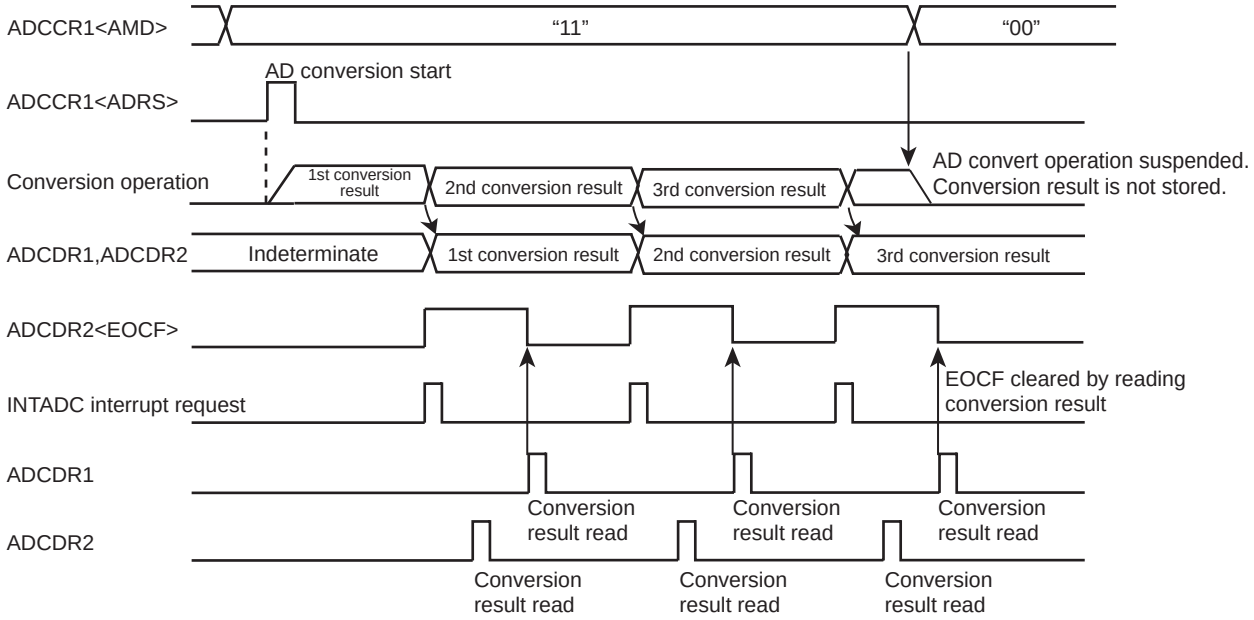


Figure 13-3 Repeat Mode

13.3.3 Register Setting

1. Set up the AD converter control register 1 (ADCCR1) as follows:
   - Choose the channel to AD convert using AD input channel select (SAIN).
   - Specify analog input enable for analog input control (AINDS).
   - Specify AMD for the AD converter control operation mode (software or repeat mode).
2. Set up the AD converter control register 2 (ADCCR2) as follows:
   - Set the AD conversion time using AD conversion time (ACK). For details on how to set the conversion time, refer to Figure 13-1 and AD converter control register 2.
   - Choose IREFON for DA converter control.
3. After setting up (1) and (2) above, set AD conversion start (ADRS) of AD converter control register 1 (ADCCR1) to "1". If software start mode has been selected, AD conversion starts immediately.
4. After an elapse of the specified AD conversion time, the AD converted value is stored in AD converted value register 1 (ADCDR1) and the AD conversion finished flag (EOCF) of AD converted value register 2 (ADCDR2) is set to "1", upon which time AD conversion interrupt INTADC is generated.
5. EOCF is cleared to "0" by a read of the conversion result. However, if reconverted before a register read, although EOCF is cleared the previous conversion result is retained until the next conversion is completed.

Example :After selecting the conversion time 19.5 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
```

13.4 STOP/SLOW Modes during AD Conversion

When standby mode (STOP or SLOW mode) is entered forcibly during AD conversion, the AD convert operation is suspended and the AD converter is initialized (ADCCR1 and ADCCR2 are initialized to initial value). Also, the conversion result is indeterminate. (Conversion results up to the previous operation are cleared, so be sure to read the conversion results before entering standby mode (STOP or SLOW mode).) When restored from standby mode (STOP or SLOW mode), AD conversion is not automatically restarted, so it is necessary to restart AD conversion. Note that since the analog reference voltage is automatically disconnected, there is no possibility of current flowing into the analog reference voltage.

13.5 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 10-bit digital value converted by the AD as shown in Figure 13-4.

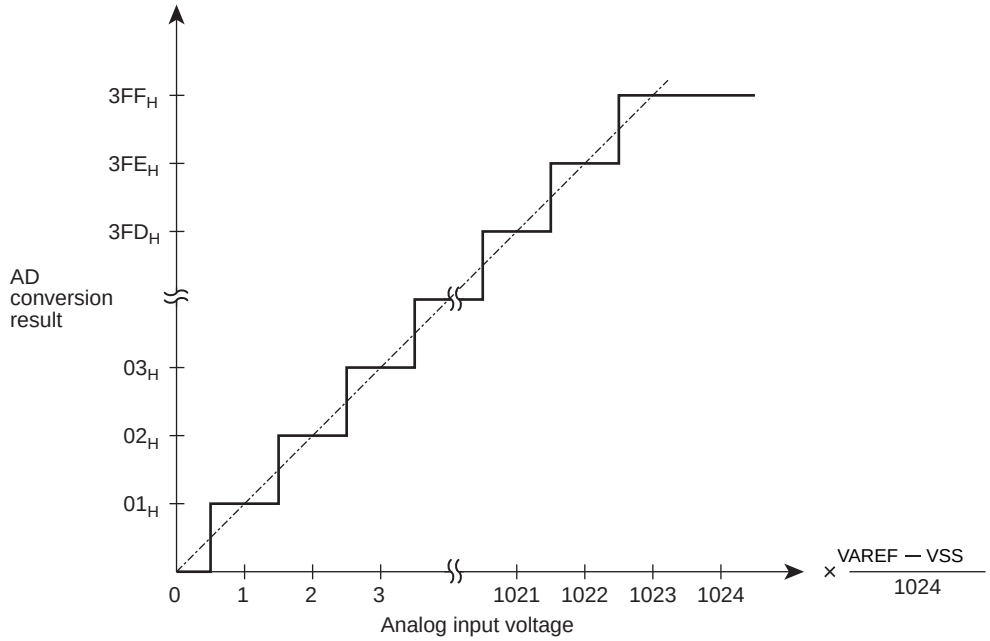


Figure 13-4 Analog Input Voltage and AD Conversion Result (Typ.)

13.6 Precautions about AD Converter

13.6.1 Restrictions for AD Conversion interrupt (INTADC) usage

When an AD interrupt is used, it may not be processed depending on program composition. For example, if an INTADC interrupt request is generated while an interrupt with priority lower than the interrupt latch IL15 (INTADC) is being accepted, the INTADC interrupt latch may be cleared without the INTADC interrupt being processed.

The completion of AD conversion can be detected by the following methods:

(1) Method not using the AD conversion end interrupt

Whether or not AD conversion is completed can be detected by monitoring the AD conversion end flag (EOCF) by software. This can be done by polling EOCF or monitoring EOCF at regular intervals after start of AD conversion.

(2) Method for detecting AD conversion end while a lower-priority interrupt is being processed

While an interrupt with priority lower than INTADC is being processed, check the AD conversion end flag (EOCF) and interrupt latch IL15. If IL15 = 0 and EOCF = 1, call the AD conversion end interrupt processing routine with consideration given to PUSH/POP operations. At this time, if an interrupt request with priority higher than INTADC has been set, the AD conversion end interrupt processing routine will be executed first against the specified priority. If necessary, we recommend that the AD conversion end interrupt processing routine be called after checking whether or not an interrupt request with priority higher than INTADC has been set.

13.6.2 Analog input pin voltage range

Make sure the analog input pins (AIN0 to AIN7) are used at voltages within VAREF to VSS. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain. The other analog input pins also are affected by that.

13.6.3 Analog input shared pins

The analog input pins (AIN0 to AIN7) are shared with input/output ports. When using any of the analog inputs to execute AD conversion, do not execute input/output instructions for all other ports. This is necessary to prevent the accuracy of AD conversion from degrading. Not only these analog input shared pins, some other pins may also be affected by noise arising from input/output to and from adjacent pins.

13.6.4 Noise Countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 13-5. The higher the output impedance of the analog input source, more easily they are susceptible to noise. Therefore, make sure the output impedance of the signal source in your design is 5 kΩ or less. Toshiba also recommends attaching a capacitor external to the chip.

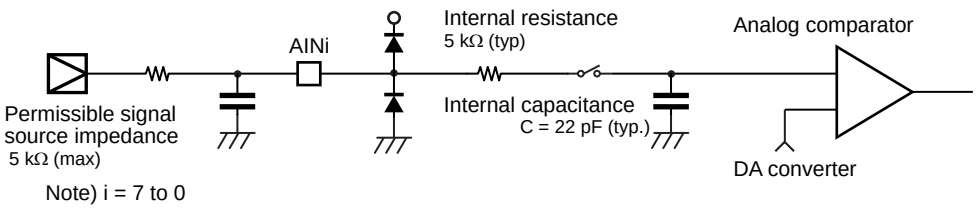


Figure 13-5 Analog Input Equivalent Circuit and Example of Input Pin Processing

14. Key-on Wakeup (KWU)

In the TMP86PM23UG, the STOP mode is released by not only P20($\overline{INT5}$/$\overline{STOP}$) pin but also four (STOP2 to STOP5) pins.

When the STOP mode is released by STOP2 to STOP5 pins, the $\overline{STOP}$ pin needs to be used.
In details, refer to the following section " 14.2 Control ".

14.1 Configuration

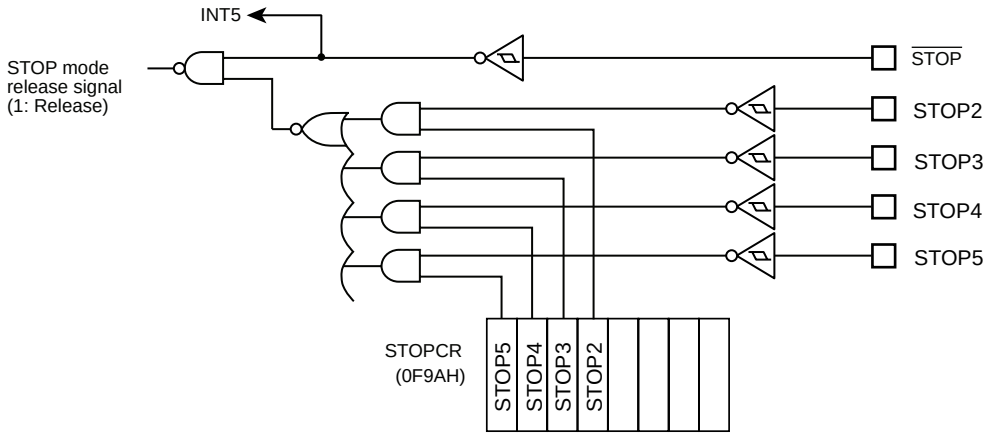


Figure 14-1 Key-on Wakeup Circuit

14.2 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	
(0F9AH)	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

14.3 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 14-2).

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 14-3).

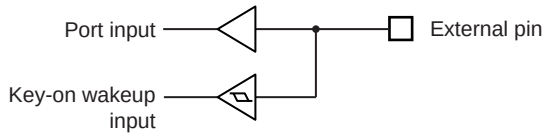


Figure 14-2 Key-on Wakeup Input and Port Input

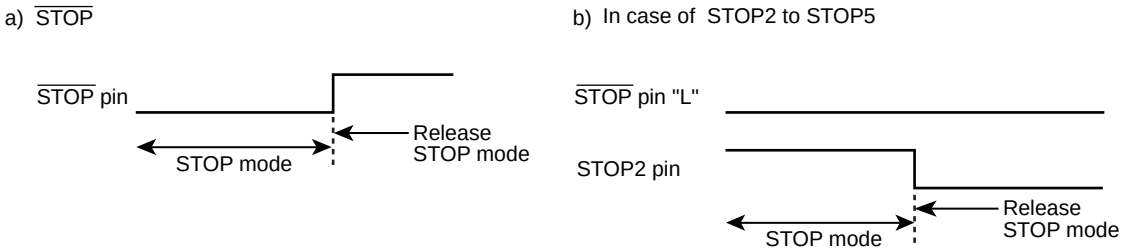


Figure 14-3 Priority of $\overline{\text{STOP}}$ pin and STOP2 to STOP5 pins

Table 14-1 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)

15. LCD Driver

The TMP86PM23UG 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 32 pins (SEG31 to SEG0)
2. Common output port 4 pins (COM3 to COM0)

In addition, VLC pin is provided for the LCD power supply.

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 128 Segments (8 segments × 16 digits)
2. 1/3 Duty (1/3 Bias) LCD Max 96 Segments (8 segments × 12 digits)
3. 1/3 Duty (1/2 Bias) LCD Max 96 Segments (8 segments × 12 digits)
4. 1/2 Duty (1/2 Bias) LCD Max 64 Segments (8 segments × 8 digits)
5. Static LCD Max 32 Segments (8 segments × 4 digits)

15.1 Configuration

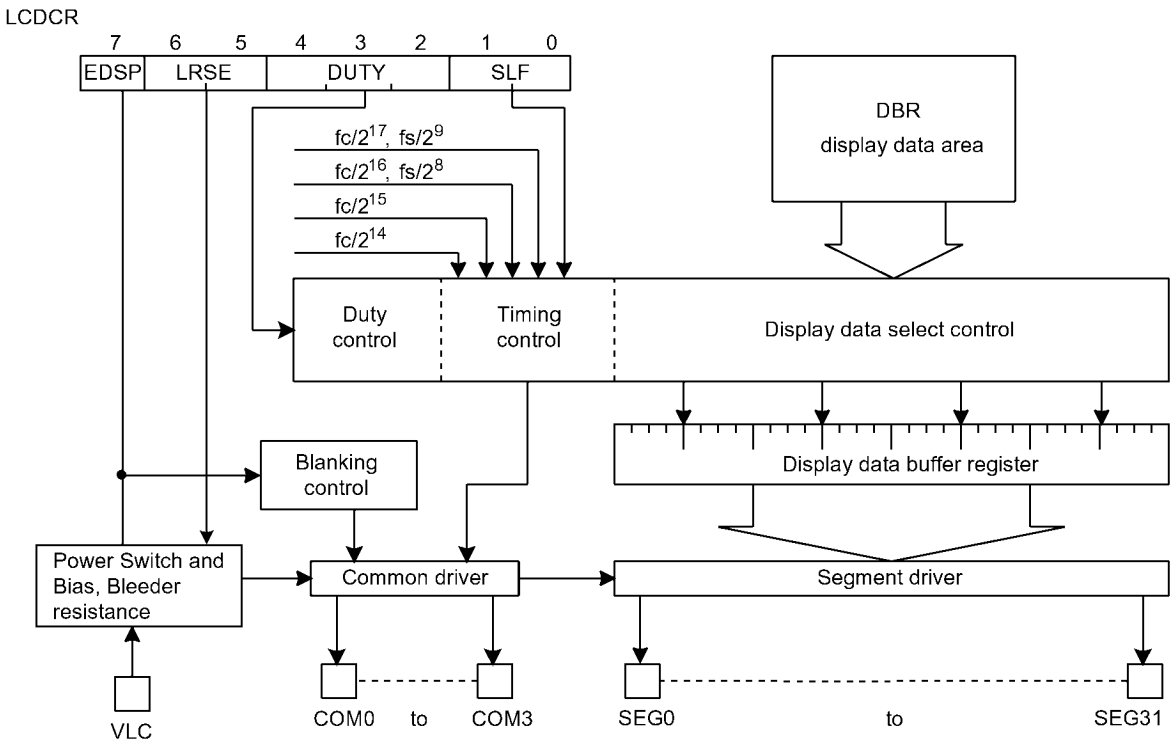


Figure 15-1 LCD Driver

15.2 Control

The LCD control register (LCDCR) controls the LCD driver. EDSP specifies whether to enable the LCD display. If EDSP is cleared to "0" for blanking, the power switch for the VLC pin is turned off. So, the COM pin and pin output selected with SEG enter GND level.

LCD Driver Control Register

LCDCR (0027H)

7	6	5	4	3	2	1	0
EDSP	LRSEL			DUTY		SLF	

(Initial value: 0000 0000)

EDSP	LCD display control	0: Blanking 1: Enables LCD display (Blanking is released)						R/W
LRSE	Period selection of enabling (turn on) of the low bleeder resistor (for implementing appropriate LCD panel drive capability)	NORMAL1/2, IDLE/1/2 mode					SLOW1/2, SLEEP1/2 mode	
		SLF Setting					SLF Setting	
			11	10	01	00	01	00
		00:	2^6/fc	2^7/fc	2^8/fc	2^9/fc	1/fs	2/fs
		01:	2^9/fc	2^{10}/fc	2^{11}/fc	2^{12}/fc	2^3/fs	2^4/fs
		10:	Always enabling					
		11:	Reserved					
DUTY	Selection of driving methods	000: 1/4 Duty (1/3 Bias) 001: 1/3 Duty (1/3 Bias) 010: 1/3 Duty (1/2 Bias) 011: 1/2 Duty (1/2 Bias) 100: Static 101: Reserved 110: Reserved 111: Reserved						
SLF	Selection of LCD frame frequency		NORMAL1/2, IDLE0/1/2 mode				SLOW1/2, SLEEP1/2 mode	
		00: 01: 10: 11:	fc/2^{17} [Hz] fc/2^{16} fc/2^{15} fc/2^{13}				fs/2^9 [Hz] fs/2^8 Reserved Reserved	

Note 1: The base-frequency (SLF) source clock is switched between high and low frequencies by the SYSCR2<SYSCK> programming. The base frequency does not depend on the TBTCR<DV7CK> programming.

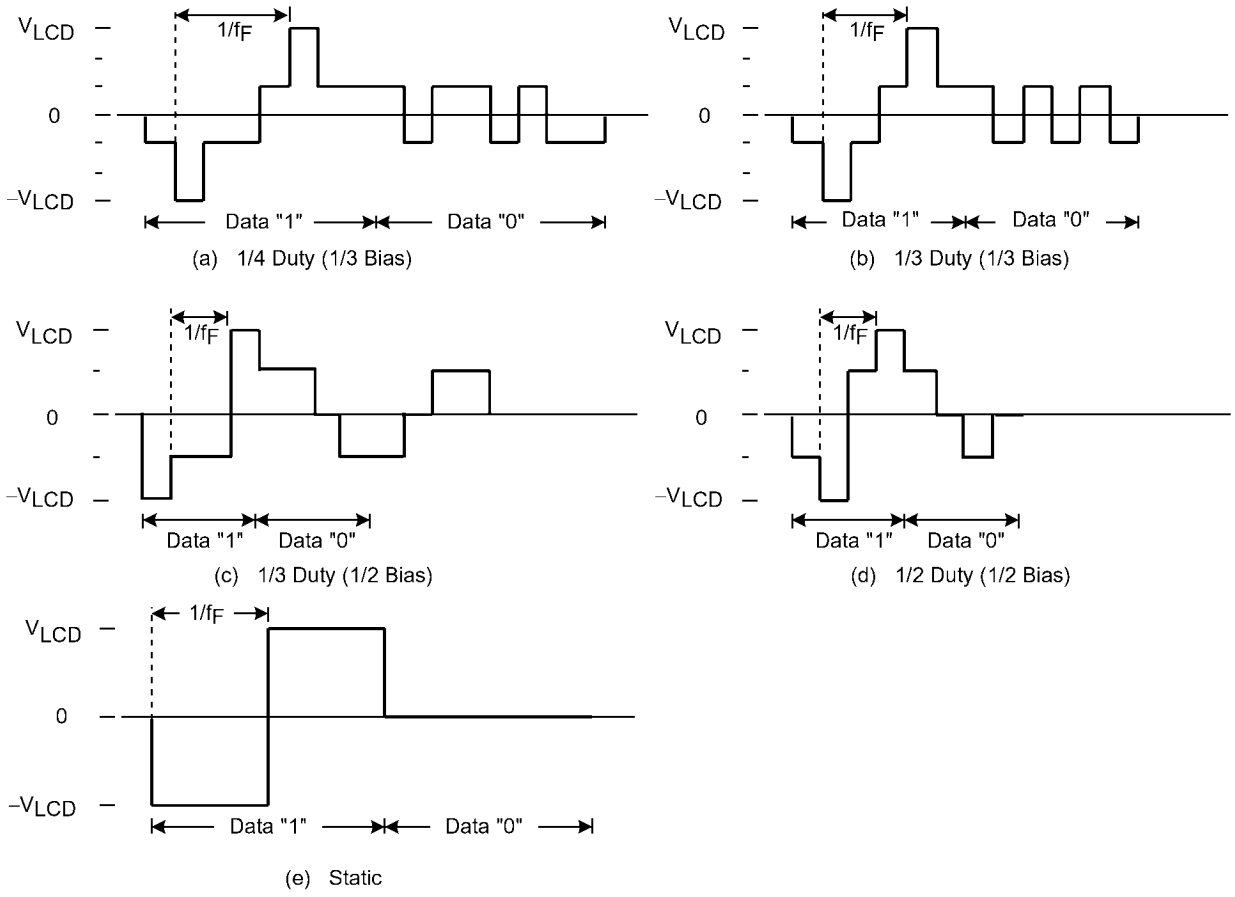
Note 2: If the setting of SYSCR2<SYSCK>is changed, be sure to turn off the LCD (clear EDSP to "0") to avoid the output of incorrect waveform.

Note 3: Programming LRSE properly according to the LCD panel used. As the LRSE programming increases (lengthen the period of enabling of the low resistor), the drive capability becomes higher while the power dissipation increases. Reversely, as the LRSE programming decreases shorten the period of enabling of the low resistor, the drive capability becomes lower while the power consumption decreases.

Note 4: If the IDLE0, SLEEP0, or STOP mode is activated when the display is enabled, LCDCR<EDSP> is automatically changed to "0" to blank the display.

15.2.1 LCD driving methods

As for LCD driving method, 5 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_{LCD}: LCD drive voltage (= $V_{DD} - V_{LC}$)

Figure 15-2 LCD Drive Waveform (COM - SEG pins)

15.2.2 Frame frequency

Frame frequency (f_F) is set according to driving method and base frequency as shown in the following Table 15-1. The base frequency is selected by LCDCR<SLF> according to the frequency fc and fs of the basic clock to be used.

Table 15-1 Setting of LCD Frame Frequency for high frequency clock

(a) At the SYSCR2<SYSCK> = "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	$\frac{fc}{2^{14}}$	$\frac{fc}{2^{14}}$	$\frac{4}{3}\bullet\frac{fc}{2^{14}}$	$\frac{4}{2}\bullet\frac{fc}{2^{14}}$	$\frac{fc}{2^{14}}$
	(fc = 2 MHz)	122	162	244	122
	(fc = 1 MHz)	61	81	122	61

Note: fc: High-frequency clock [Hz]

Table 15-2 Setting of LCD Frame Frequency for low frequency clock

(b) At the SYSCR2<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
1*	Reserved				

Note: fs: Low-frequency clock [Hz]

15.2.3 LCD drive voltage

LCD driving voltage VLCD is given as potential difference VDD – VLC between pins VDD and VLC. Therefore, when the CPU voltage and LCD drive voltage are the same, VLC pin will be connected to VSS pin. The LCD lights when the potential difference between segment output and common output is ±VLCD. Otherwise it turns off.

During reset, the power switch of LCD driver is automatically turned off, shutting off the VLC voltage.

After reset, if the P*LCR register (*; Port No.) for each port is set to "1" with LCDCR<EDSP> = "0", a GND level is output from the pin which can be used as segment.

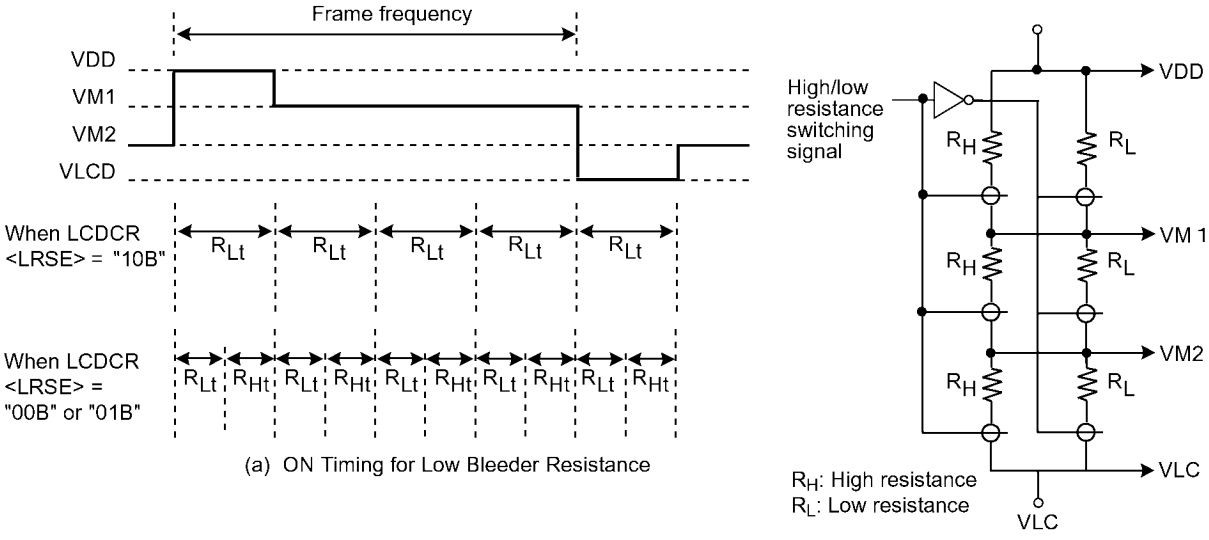
The power switch is turned on to supply VLC voltage to LCD driver by setting with LCDCR<EDSP> to "1".

If the IDLE0, SLEEP0, or STOP mode is activated, LCDCR<EDSP> is automatically changed to "0" to blank the display. To turn the display back on after releasing from the previous mode, set LCDCR<EDSP> to "1" again.

Note: During reset, the LCD common outputs (COM3 to COM0) are fixed "0" level. However, the multiplex port (input/output port or SEG output is selectable) becomes high impedance. Therefore, when the reset input is long remarkably, ghost problem may appear in LCD display.

15.2.4 Adjusting the LCD panel drive capability

The LCD panel drive capability can be adjusted by programming LCDCR<LRSE>. When the period of enabling of the low bleeder resistor is lengthened, the drive capability becomes higher while the power consumption increases. Reversely, when the period of enabling of the low bleeder resistor is shortened, the drive capability becomes lower while the power consumption decreases. If the drive capability is not enough, the LCD display might present a ghost problem. So, implement the optimum drive capability for the LCD panel used. The figure below shows the bleeder resistance timing and equivalent circuit for 1/4 duty and 1/3 bias.



R_{Lt}: Period during which resistance R_L is selected (Time specified with LCDCR<LRSE>)
R_{Ht}: Period during which resistance R_H is selected (Time specified with LCDCR<SLF> ÷ 4 – Time specified with LCDCR<LRSE>)

Figure 15-3 Bleeder Resistance Selection with LCDCR<LRSE> (for 1/4 duty and 1/3 bias)

15.3 LCD Display Operation

15.3.1 Display data setting

Display data is stored to the display data area (address 0F80H to 0F8FH,16 bytes) in the DBR. The display data 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 15-4 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 15-3).

Table 15-3 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 15-4 LCD Display Data Area (DBR)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0F80H	SEG1				SEG0			
0F81H	SEG3				SEG2			
0F82H	SEG5				SEG4			
0F83H	SEG7				SEG6			
0F84H	SEG9				SEG8			
0F85H	SEG11				SEG10			
0F86H	SEG13				SEG12			
0F87H	SEG15				SEG14			
0F88H	SEG17				SEG16			
0F89H	SEG19				SEG18			
0F8AH	SEG21				SEG20			
0F8BH	SEG23				SEG22			
0F8CH	SEG25				SEG24			
0F8DH	SEG27				SEG26			
0F8EH	SEG29				SEG28			
0F8FH	SEG31				SEG30			
	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0

15.3.2 Blanking

Blanking is enabled when LCDCR<EDSP> is cleared to "0".

To blank the LCD display and turn it off, a GND-level signal is output to the COM pin and the port which can be used as the segment by setting of P*LCR register (*; Port No.). At this time, the power switch of VLC pin is turned off.

15.4 Control Method of LCD Driver

15.4.1 Initial setting

Figure 15-4 shows the flowchart of initialization.

Example :To operate a 1/4 duty LCD of 32 segments × 4 com-mons at frame frequency fc/2^{16} [Hz],
The period of enabling of the low bleeder resistor: 2^8/fc

```
LD      (LCDCR), 00000001B      ; Sets LCD driving method, The period of enabling of low bleeder
                                  resistor and frame frequency.
LD      (P*LCR), 0FFH           ; Sets segment output control register. (*; Port No.)
 :       :
 :       :                      ; Sets the initial value of display data.
LD      (LCDCR), 10000001B      ; Display enable
```

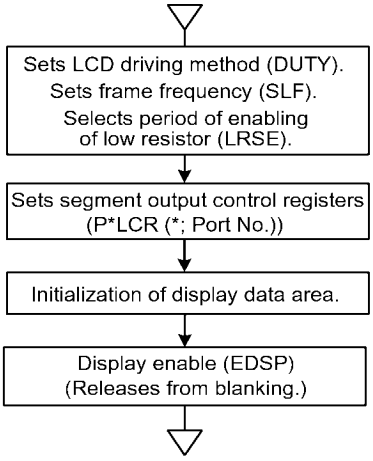


Figure 15-4 Initial Setting of LCD Driver

15.4.2 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 :(1) 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 15-5), display data become as shown in Table 15-5.

```
        LD      A, (80H)
        ADD     A, TABLE-$-7
        LD      HL, 0F85H
        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 definition instruction.

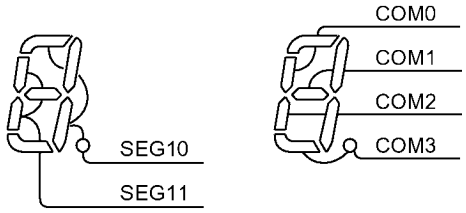


Figure 15-5 Example of COM, SEG Pin Connection (1/4 duty)

Table 15-5 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 15-6 shows an example of display data which are displayed using 1/2 duty LCD in the same way as Table 15-5. The connection between pins COM and SEG are the same as shown in Figure 15-6.

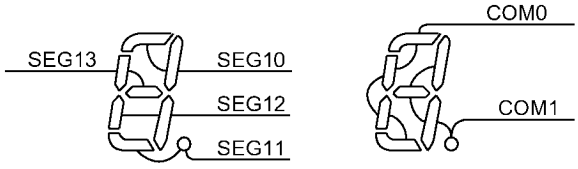


Figure 15-6 Example of COM, SEG Pin Connection

Table 15-6 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

15.4.3 Example of LCD driver output

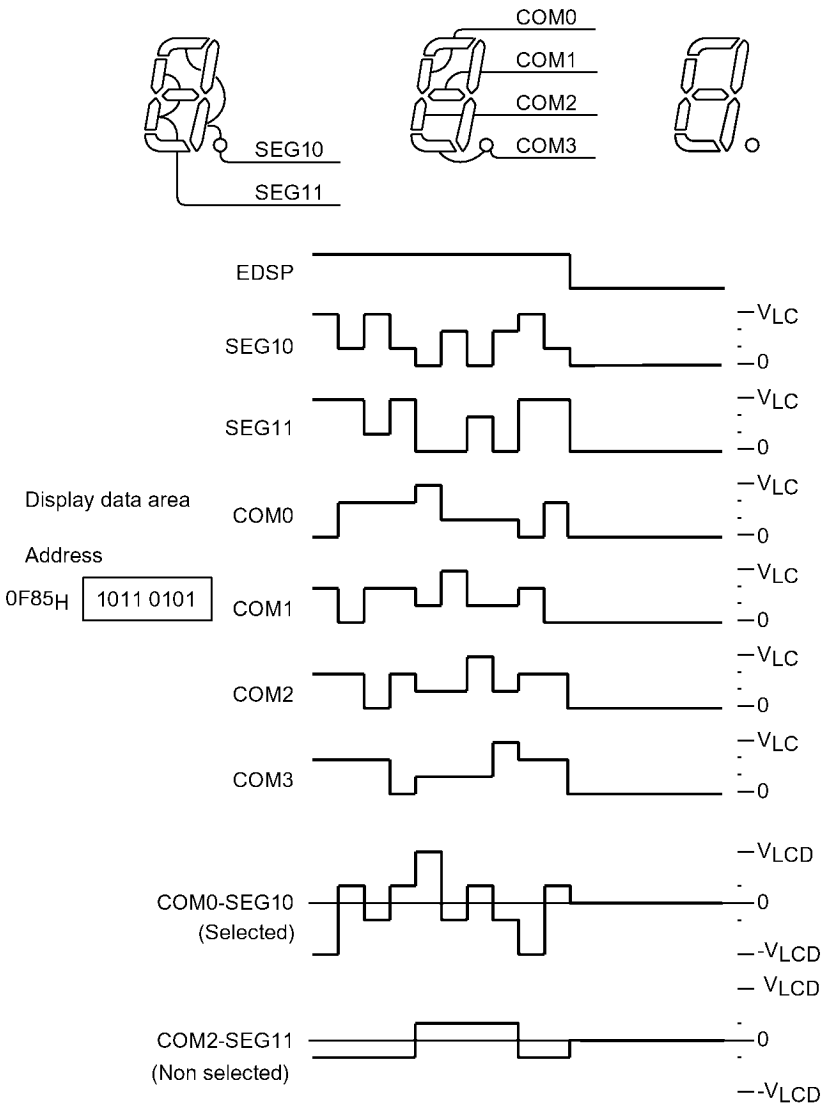


Figure 15-7 1/4 Duty (1/3 Bias) Drive

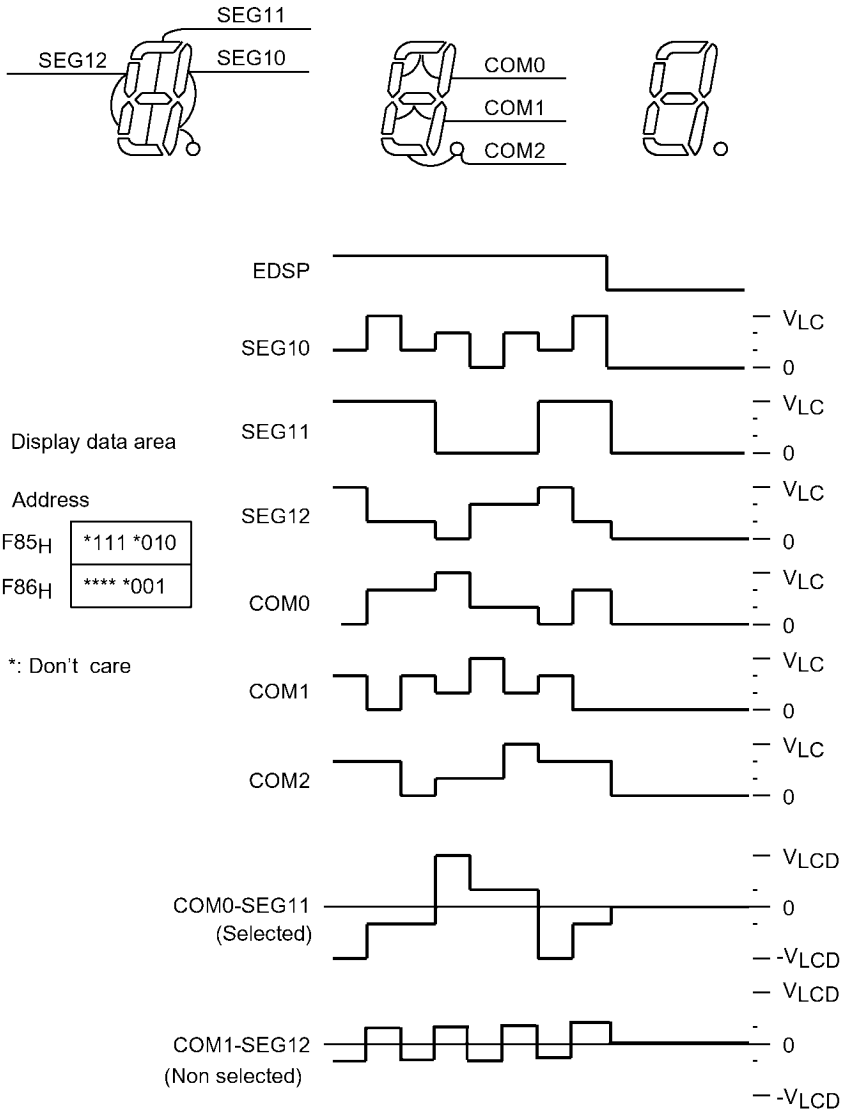


Figure 15-8 1/3 Duty (1/3 Bias) Drive

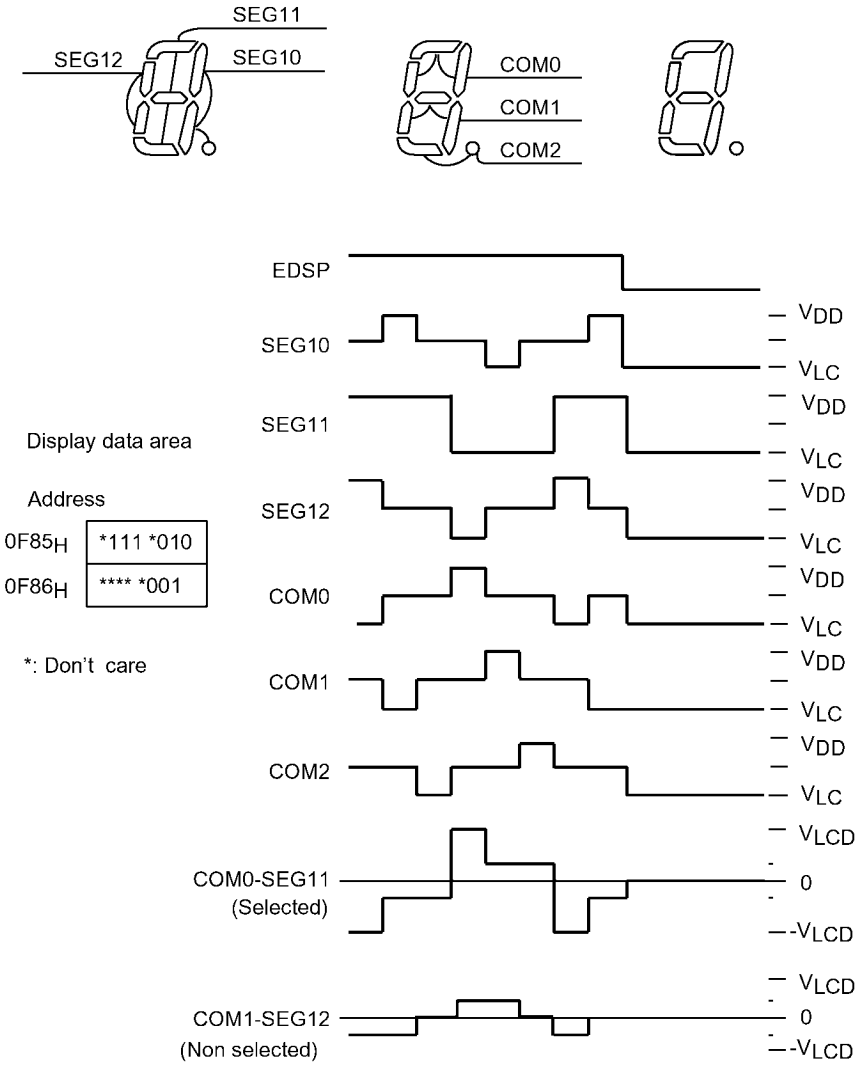


Figure 15-9 1/3 Duty (1/2 Bias) Drive

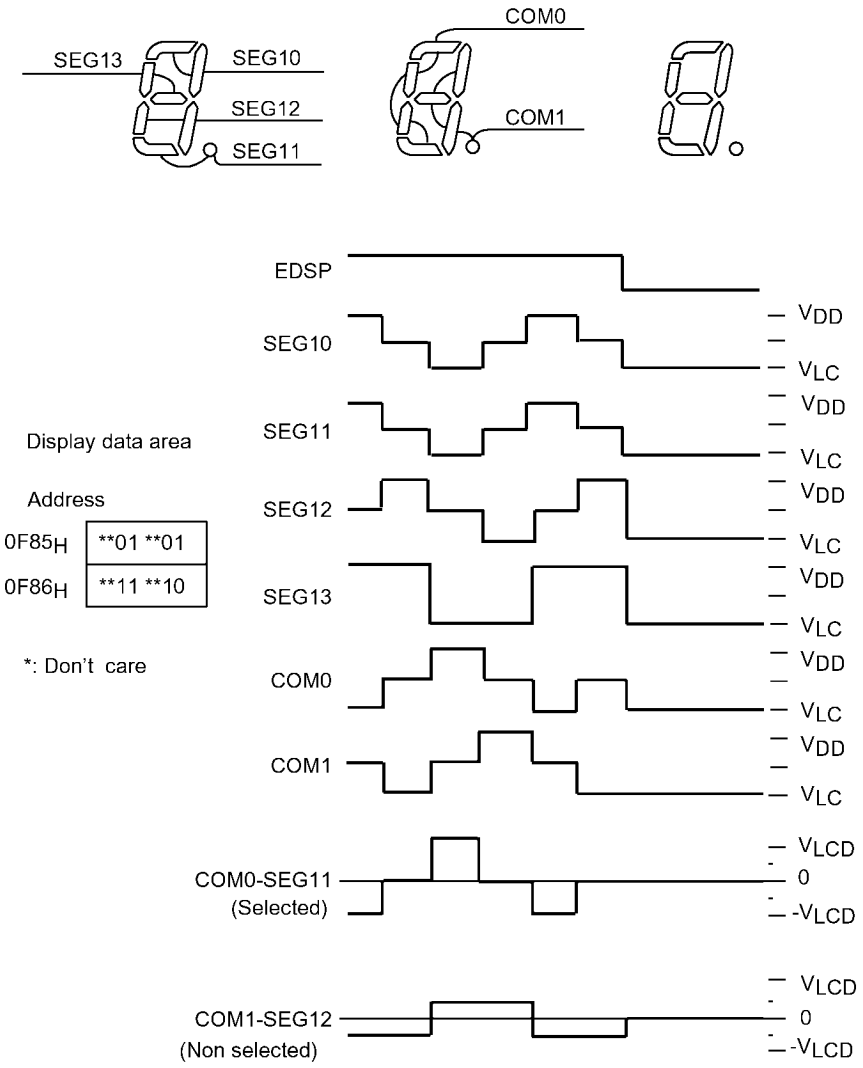


Figure 15-10 1/2 Duty (1/2 Bias) Drive

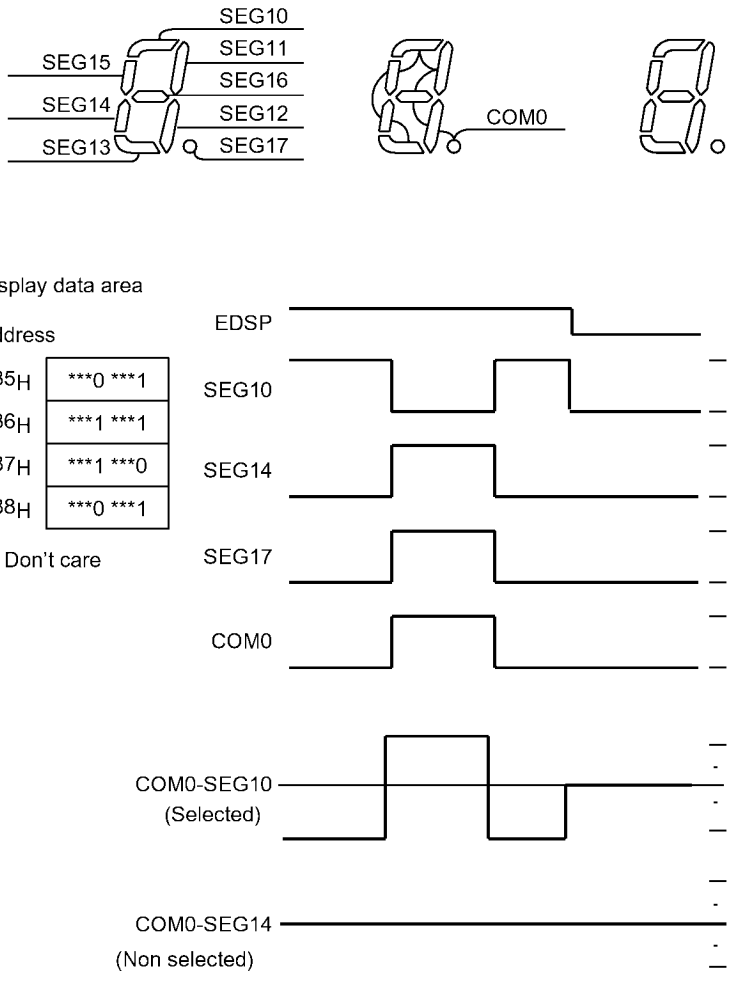


Figure 15-11 Static Drive

16. Real-Time Clock

The TMP86PM23UG include a real time counter (RTC). A low-frequency clock can be used to provide a periodic interrupt (0.0625[s],0.125[s],0.25[s],0.50[s]) at a programmed interval, implement the clock function. The RTC can be used in the mode in which the low-frequency oscillator is active (except for the SLEEP0 mode).

16.1 Configuration

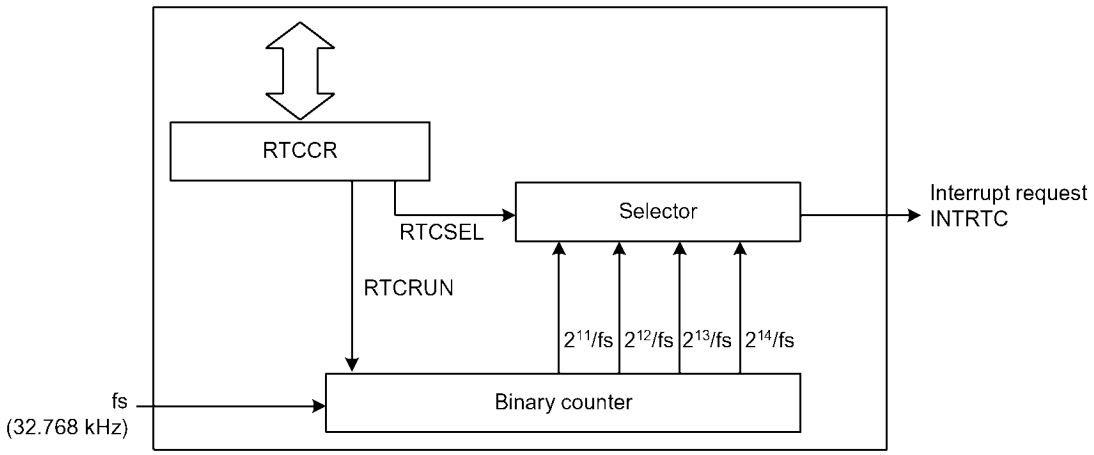


Figure 16-1 Configuration of the RTC

16.2 Control of the RTC

The RTC is controlled by the RTC control register (RTCCR).

RTC Control Register

RTCCR (0017H)	7	6	5	4	3	2	1	0	(Initial value: **** *000)
						RTCSEL	RTCRUN		

RTCSEL	Interrupt generation period (fs = 32.768 kHz)	00: 0.50 [s] 01: 0.25 [s] 10: 0.125 [s] 11: 0.0625 [s]	R/W
RTCRUN	RTC control	0: Stops and clears the binary counter. 1: Starts counting	

Note 1: Program the RTCCR during low-frequency oscillation (when SYSCR2<XTEN> = "1"). For selecting an interrupt generation period, program the RTCSEL when the timer is inactive (RTCRUN = "0"). During the timer operation, do not change the RTCSEL programming at the same moment the timer stops.

Note 2: The timer automatically stops, and this register is initialized (the timer's binary counter is also initialized) if one of the following operations is performed while the timer is active:
1. Stopping the low-frequency oscillation (with SYSCR2<XTEN> = "0")
2. When the TMP86PM23UG are put in STOP or SLEEP0 mode
Therefore, before activating the timer after releasing from STOP or SLEEP0 mode, reprogram the registers again.

Note 3: If a read instruction for RTCCR is executed, undefined value is set to bits 7 to 3.

Note 4: If break processing is performed on the debugger for the development tool during the timer operation, the timer stops counting (contents of the RTCCR isn't altered). When the break is cancelled, processing is restarted from the point at which it was suspended.

16.3 Function

The RTC counts up on the internal low-frequency clock. When RTCCR<RTCRUN> is set to "1", the binary counter starts counting up. Each time the end of the period specified with RTCCR<RTCSEL> is detected, an INTRTC interrupt is generated, and the binary counter is cleared. The timer continues counting up even after the binary counter is cleared.

17. Multiply-Accumulate (MAC) Unit

The TMP86PM23UG includes a multiply-accumulate (MAC) unit.

The MAC unit is capable of executing 16-bit × 16-bit multiplications and 16-bit × 16-bit + 32-bit multiply-accumulate operations.

The MAC unit supports only integer arithmetic, not fixed-point or floating-point arithmetic. Both signed and unsigned operations can be performed.

The MAC unit can only be used in NORMAL1 or NORMAL2 mode. All the registers of the MAC unit are initialized upon entering a mode other than NORMAL mode.

With development tools, if break mode is entered while the MAC unit is calculating, the calculation is continued but its result is unpredictable. In this case, the calculation must be re-executed after break mode is exited. Do not write to the multiplicand register in break mode. When the calculation is completed, it is possible to enter break mode and read the calculation result in break mode.

17.1 Configuration

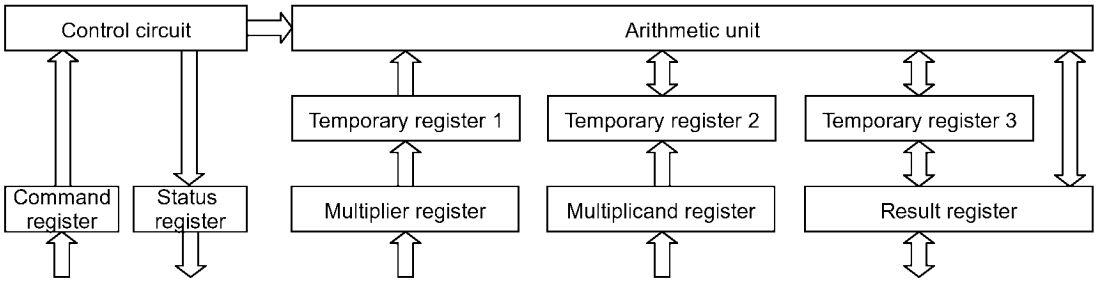


Figure 17-1 MAC Unit

17.2 Registers

The MAC unit consists of the following registers:

Table 17-1 Registers in the MAC Unit

Register	Address	Number of Bits
Command register (MACCR)	0FA4H	8 bits
Status register (MACSR)	0FA5H	8 bits
Multiplier data register (MPLDRH, MPLDRL)	0FA7H, 0FA6H	16 bits
Multiplicand data register (MPCDRH, MPCDRL)	0FA9H, 0FA8H	16 bits
Result register (RCALDR4 to RCALDR1)	0FAAH to 0FADH	32 bits
Addend register (MADDR4 to MADDR1)	0FAAH to 0FADH	32 bits

17.2.1 Command Register

The command register is used to enable and disable the MAC unit, specify the arithmetic mode, and clear the result register.

17.2.2 Status Register

The status register contains flags to indicate the operation status of the MAC unit and the calculation result.

17.2.3 Multiplier data Register

The data written to this register is calculated as a multiplier.

17.2.4 Multiplicand data Register

The data written to this register is calculated as a multiplicand.

17.2.5 Result Register

The calculation result is stored in this register.

17.2.6 Addend Register

The data written to this register is calculated as an addend in a multiply-accumulate operation. An addend must be written to this register while calculation is not being performed (CALC = "0").

17.3 Control

Command Register

MACCR (0FA4H)

7	6	5	4	3	2	1	0
RCLR	"1"	"1"	"1"	CMOD			EMAC

(Initial value: 0111 0000)

RCLR	Result register clear	0: -(Keeps the value of the result register.) 1: Clears the result register. (This bit is automatically cleared to "0" one machine cycle after it is set to "1".)	R/W
CMOD	Arithmetic mode	000: Unsigned multiply (16 bits × 16 bits) 001: Unsigned multiply-accumulate (16 bits × 16 bits + 32 bits) 010: Signed multiply (16 bits × 16 bits) 011: Signed multiply-accumulate (16 bits × 16 bits + 32 bits) 1**: Reserved	
EMAC	MAC unit control	0: Disables the MAC unit. 1: Enables the MAC unit.	

Note 1: Setting RCLR to "1" causes the result, addend, and status registers to be initialized. The multiplier, multiplicand, and command registers remain the same as before. (RCLR is automatically cleared to "0" one machine cycle after it is set to "1".)

Note 2: Writing to CMOD (including an overwrite) makes no changes to the status, multiplier, multiplicand, result, and addend registers.

Note 3: Before changing the arithmetic mode, be sure to check that calculation is not being performed (CALC = "0").

Note 4: Clearing the result register with RCLR is possible only when calculation is not being performed (CALC = "0"). (RCLR cannot be set to "1"during calculation.)

Note 5: Bits 6 to 4 are always read as "1". ("0" cannot be written.)

Status Register

MACSR (0FA5H)

7	6	5	4	3	2	1	0
"1"	"1"	"1"	CARF	ZERF	SIGN	OVRF	CALC

(Initial value: 1110 0000)

CARF	Carry flag	0: No carry occurred in multiply-accumulate operation. 1: Carry occurred in multiply-accumulate operation.	Read only
ZERF	Zero flag	0: Calculation resulst is other than "00000000H". 1: Calculation result is "00000000H".	
SIGN	Sign flag	0: Result register contents are positive or "00000000H". 1: Result register contents are negative.	
OVRF	Overflow flag	0: Overflow occurred. 1: No overflow occurred.	
CALC	Operation status flag	0: Calculation not in progress 1: Calculation in progress	

Note 1: The status register is initialized when the result register is cleared (RCLR = "1").

Note 2: CARF, ZERF, SIGN, and OVRF are programmed at the end of calculation. They are not affected by a read from the status register.

Note 3: ZERF and SIGN are not affected by a write to the addend register.

Note 4: In multiply mode, OVRF and CARF are always read as "0".

Note 5: Bit 7 to 5 are always read as "1".

Multiplier data Register

MPLDRH, MPLDRL (0FA7H, 0FA6H)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPLDRH (0FA7H)								MPLDRL (0FA6H)							

(Initial value: 0000 0000 0000 0000) R/W

Note: In signed arithmetic mode, bit 15 is treated as the sign bit.

Multiplicand data Register

MPCDRH, MPCDRL (0FA9H, 0FA8H)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPCDRH (0FA9H)								MPCDRL (0FA8H)							

(Initial value: 0000 0000 0000 0000) R/W

Note 1: In signed arithmetic mode, bit 15 is treated as the sign bit.

Note 2: Calculation can only be started by writing to both the lower byte (MPCDRL) and upper byte (MPCDRH) of the multiplicand register in this order.

Note 3: The multiplicand register can only be programmed when data is written in the order of lower byte and upper byte. If data is only written to the upper byte, the written data cannot be read out. (If data is only written to the lower byte, the written data can be read out.)

Result Register

RCALDR4, RCALDR3 (0FADH, 0FACH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCALDR4 (0FADH)								RCALDR3 (0FACH)							

(Initial value: 0000 0000 0000 0000) Read only

RCALDR2, RCALDR1 (0FABH, 0FAAH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCALDR2 (0FABH)								RCALDR1 (0FAAH)							

(Initial value: 0000 0000 0000 0000) Read only

Note: In signed arithmetic mode, bit 31 contains the sign of the calculation result.

Addend Register

MADDR4, MADDR3 (0FADH, 0FACH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MADDR4 (0FADH)								MADDR3 (0FACH)							

(Initial value: 0000 0000 0000 0000) Write only

MADDR2, MADDR1 (0FABH, 0FAAH)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MADDR2 (0FABH)								MADDR1 (0FAAH)							

(Initial value: 0000 0000 0000 0000) Write only

Note 1: In signed arithmetic mode, bit 31 is treated as the sign bit.

Note 2: Writing to the addend register changes the contents of the result register. Thus, read from the result register before writing to the addend register.

17.4 Register Description

17.4.1 EMAC

Setting MACCR<EMAC> to "1" enables the MAC unit. Once enabled, the MAC unit remains enabled until it is disabled.

17.4.2 CMOD

The MACCR<CMOD> is used to specify the arithmetic mode.

Calculation is started automatically when data is written to both the lower byte (MPCDRL) and upper byte (MPCDRH) of the multiplicand register in this order. Thus, the multiplier register (MPLDRH, MPLDRL) must be set before the multiplicand register. When calculation is completed, the result is stored in the result register (RCALDR4 to RCALDR1).

The arithmetic mode is valid until the CMOD field is changed. Note that if the operation mode is changed to IDLE0/1/2, SLOW1/2, or STOP mode, CMOD is initialized.

During calculation, the next data can be written to the multiplier and multiplicand registers only once. Do not write to these registers more than once. Whether or not calculation is in progress can be checked by reading the MACSR<CALC> flag.

Note 1: Before changing the arithmetic mode, ensure that calculation is not being performed (CALC = "0").

Note 2: Writing to the CMOD field (including an overwrite) makes no changes to the status, multiplier, multiplicand, result, and addend registers. Thus, to clear the status, result, and addend registers after a change of the arithmetic mode, set the RCLR bit to "1".

17.4.3 RCLR

When calculation is not being performed (CALC = "0"), setting MACCR<RCLR> to "1" causes the result, addend, and status registers to be initilized. (The multiplier and multiplicand registers remain the same as before.) RCLR is automatically cleared to "0" one machine cycle after it is set to "1"

Note: When calculation is in progress (CALC = "1"), RCLR cannot be set to "1". (The instruction to set it to "1" is invalid.)

As shown in Table 17-2, the state of each register changes when: the MAC unit is disabled (EMAC = "0"); the result register is cleared (RCLR = "1"); or the operation mode is changed.

Table 17-2 Effects of the EMAC and RCLR Bits on the MAC Registers

Register	EMAC = "0" (Disable)	RCLR = "1" (register clear)	IDLE0/1/2, SLOW1/2, or STOP Mode
Command register (MACCR)	Bits other than EMAC remain the same as before	Bits other than RCLR remain the same as before. RCLR is cleared to "0" after one machine cycle.	Initialized
Status register (MACSR)	Initialized	Initialized	Initialized
Multiplier data register (MPLDRH, MPLDRL)	Initialized	Remains the same as before	Initialized
Multiplicand data register (MPCDRH, MPCDRL)	Initialized	Remains the same as before	Initialized
Result register (RCALDR4 to RCALDR1)	Initialized	Initialized	Initialized
Addend register (MADDR4 toMADDR1)	Initialized	Initialized	Initialized

Note 1: The multiplier, multiplicand, and addend registers can be written to only when the MAC unit is enabled (EMAC = "1").

Note 2: When writing to the multiplicand register, be sure to write to the lower byte (MPCDRL) first and then to the upper byte (MPCDRH).

Note 3: RCLR can be written to only when calculation is not being performed (CALC = "0").

Note 4: When the MAC unit is enabled (EMAC = "1"), if the operation mode is changed to IDLE0/1/2, SLOW1/2, or STOP mode, the command register (MACCR) is initialized and its contents are discarded. Thus, program the MACCR again after each of these operation modes is exited.

17.5 Arithmetic Modes

The following four arithmetic modes are available:

1. Unsigned multiply (16 bits × 16 bits)
2. Signed multiply (16 bits × 16 bits)
3. Unsigned multiply-accumulate (16 bits × 16 bits + 32 bits)
4. Signed multiply-accumulate (16 bits × 16 bits + 32 bits)

17.5.1 Unsigned Multiply Mode

Setting the MACCR<CMOD> field in the command register to "000B" places the MAC unit in unsigned multiply mode. In this mode, the values of the multiplier and multiplicand registers are each treated as 16-bit data for calculation.

Calculation is started automatically by writing a multiplier to the multiplier register (MPLDRH, MPLDRL) and then writing a multiplicand to the lower byte (MPCDRL) and upper byte (MPCDRH) of the multiplicand register in this order. The calculation result is stored as 32-bit data in the result register (RCALDR4 to RCALDR1). (The previous calculation result is cleared.)

17.5.2 Signed Multiply Mode

Setting the MACCR<CMOD> field in the command register to "010B" places the MAC unit in signed multiply mode. In this mode, bit 15 in the multiplier and multiplicand registers is each treated as the sign bit.

Calculation is started automatically by writing a multiplier to the multiplier register (MPLDRH, MPLDRL) and then writing a multiplicand to the lower byte (MPCDRL) and upper byte (MPCDRH) of the multiplicand register in this order. The calculation result is stored as 32-bit data in the result register (RCALDR4 to RCALDR1). (Bit 31 contains the sign, and the previous calculation result is cleared.) The sign of the calculation result varies depending on the signs of the multiplier and multiplicand, as shown in Table 17-3.

Table 17-3 Signs Used in Singed Multiply Mode

Sign of Multiplier	Sign of Multiplicand	Sign of Calculation Result
0	0	0
0	1	1
1	0	1
1	1	0

17.5.3 Unsigned Multiply-Accumulate Mode

Setting the MACCR<CMOD> field in the command register to "001B" places the MAC unit in unsigned multiply-accumulate mode. In this mode, the values of the multiplier and multiplicand registers are each treated as 16-bit data for calculation.

Calculation is started automatically by writing a multiplier to the multiplier register (MPLDRH, MPLDRL) and then writing a multiplicand to the lower byte (MPCDRH) and upper byte (MPCDRH) of the multiplicand register in this order. First, the multiplier and multiplicand are multiplied. Then, the contents of the addend register are added to the product. The sum is stored as 32-bit data in the result register.

In unsigned multiply-accumulate mode, any addend can be written to the addend register when calculation is not being performed. If, for example, A × B is executed after arbitrary data C is written to the addend register, the result of A × B + C is stored in the result register (RCALDR4 to RCALDR1). Setting the RCLR bit to "1"

causes the result and addend registers to be cleared. After calculation is completed, the contents of the result register are automatically stored in the addend register. Thus, if the contents of the addend register are not changed, the result of the previous multiply-accumulate operation is used as an addend for the next calculation.

Note 1: Be sure to write to the addend register when calculation is not being performed (CALC = "0").
Note 2: Writing to the addend register changes the contents of the result register. Thus, read from the result register before writing to the addend register.

17.5.4 Signed Multiply-Accumulate Mode

Setting the MACCR<CMOD> field in the command register to "011B" places the MAC unit in signed multiply-accumulate mode. In this mode, bit 15 in the multiplier and multiplicand register is each treated as the sign bit.

Calculation is started automatically by writing a multiplier to the multiplier register (MPLDRH, MPLDRL) and then writing a multiplicand to the lower byte (MPCDRL) and upper byte (MPCDRH) of the multiplicand register in this order. First, the multiplicand and multiplicand are multiplied. Then, the contents of the addend register are added to the product. The sum is stored as signed 32-bit data in the result register (RCALDR4 to RCALDR1). The sign of the result varies as shown in Table 17-4. As in the case of unsigned multiply-accumulate mode, any addend can be written to the addend register when calculation is not being performed.

Note: In signed multiply-accumulate mode, bit 31 in the addend register is treated as the sign bit.

Table 17-4 Signs Used in Sgined Multiply-Accumulate Mode

Sign of Product	Sign of Addend	Sign (bit 31) of Calculation Result	
		When OVRF = "0"	When OVRF = "1"
0	0	0	1
0	1	"1" when sum < 0 "0" when sum ≥ 0	–
1	0	"1" when sum < 0 "0" when sum ≥ 0	–
1	1	1	0

17.5.5 Valid Numerical Ranges

Table 17-5 shows the numerical range that can be handled in each arithmetic mode.

Table 17-5 Valid Numerical Ranges in Decimal (with hexadecimal shown in brackets)

Mode	Multiplier/Multiplicand	Addend	Sum
Unsigned multiply	0 to 65535 (0000H to FFFFH)	–	0 to 4294836225 (00000000H to FFFE0001H)
Signed multiply	-32768 to 32767 (8000H to 7FFFH)	–	-1073709056 to 1073741824 (C0008000H to 40000000H)
Unsigned multiply-accumulate	0 to 65535 (0000H to FFFFH)	0 to 4294967295 (00000000H to FFFFFFFFH)	0 to 4294967295 (00000000H to FFFFFFFFH)
Signed multiply-accumulate	-32768 to 32767 (8000H to 7FFFH)	-2147483648 to 2147483647 (80000000H to 7FFFFFFFH)	-2147483648 to 2147483647 (80000000H~7FFFFFFFH)

17.6 Status Flags

The status register MACSR contains the following five flags. OVRF, CARF, SIGN, and ZERF are programmed when calculation is completed, and these flags are not affected by a read from the status register.

1. Operation status flag (CALC)
2. Overflow flag (OVRF)
3. Carry flag (CARF)
4. Sign flag (SIGN)
5. Zero flag (ZERF)

17.6.1 Operation Status Flag (CALC)

CALC indicates the status of the MAC unit. It is set to "1" when calculation is in progress and "0" when calculation is not in progress.

17.6.2 Overflow Flag (OVRF)

OVRF is set to "1" if the sum of positive values is negative or the sum of negative values is positive in signed multiply-accumulate mode. In other cases, it is cleared to "0".

Note: In multiply mode, OVRF is always read as "0".

17.6.3 Carry Flag (CARF)

CARF is set to "1" if a carry occurs in the highest-order bit (bit 31) in a multiply-accumulate operation. In other cases, it is cleared to "0".

Note: In multiply mode, CARF is always read as "0".

17.6.4 Sign Flag (SIGN)

SIGN contains the same data as the highest-order bit (bit 31) of the calculation result (regardless of whether calculation is performed in signed or unsigned mode).

Note: SIGN is programmed by the calculation result. This flag is not affected by a write to the addend register.

17.6.5 Zero Flag (ZERF)

ZERF is set to "1" if the result register contains "00000000H". In other cases, it is cleared to "0". It is also set to "1" if the result register contains "00000000H" after an overflow or carry has occurred.

Note: ZERF is programmed by the calculation result. This flag is not affected by a write to the addend register.

17.7 Example of Software Processing

The following shows an example of calculating $X = \alpha x + \beta y + \gamma z$. The calculation time is 3μs when fc = 8 MHz. The multiplier and multiplicand are separately stored in data RAM. The W and A registers are used as general-purpose registers. The general-purpose registers are not saved on the stack. The processing for enabling/disabling the MAC unit is not included.

Note 1: If the operation mode is changed by processing an interrupt during calculation, the correct calculation result may not be obtained. Thus, before starting calculation, be sure to execute the DI instruction to disable interrupts.

Note 2: Before reading the result register after calculation is started, check that the CALC flag in the MACSR register is "0" or wait for at least three machine cycles (e.g. NOP x 3).

Instruction			Processing Time
DI			(Disables interrupts.)
LD	WA, (RAM_Multiplier α)		6 cycles/3 μs
LD	(MPLDRL), WA	; Multiplier register	6 cycles/3 μs
LD	WA, (RAM_Multiplier x)		6 cycles/3 μs
LD	(MPCDRL), WA	; Multiplicand register	6 cycles/3 μs
; The next data can be written in succession.			

LD	WA, (RAM_Multiplier β)		6 cycles/3 μs
LD	(MPLDRL), WA	; Multiplier register	6 cycles/3 μs
LD	WA, (RAM_Multiplicand Z)		6 cycles/3 μs
LD	(MPCDRL), WA	; Multiplicand register	6 cycles/3 μs
; The first calculation is already completed. Thus, the next data can be written.			
LD	WA, (RAM_Multiplier y)		6 cycles/3 μs
LD	(MPLDRL), WA	; Multiplier register	6 cycles/3 μs
LD	WA, (RAM_Multiplicand Z)		6 cycles/3 μs
LD	(MPCDRL), WA	; Multiplicand register	6 cycles/3 μs
NOP		; Wait three machine cycles or longer.	
NOP		; (Note 2)	
NOP			
LD	WA, (RCALDR1)	; Low-order part of the result register	6 cycles/3 μs
LD	(RAM_Low-order part of result X), WA		6 cycles/3 μs
LD	WA, (RCALDR3)	; High-order part of the result register	6 cycles/3 μs
LD	(RAM_High-order part of result X), WA		6 cycles/3 μs
EI			(Enables interrupts.)
Processing time			51 ms
<CALL mn instruction>			6 cycles/3 μs
RET			6 cycles/3 μs
Total processing time			57 μs

18. OTP operation

This section describes the funstion and basic operationalblocks of TMP86PM23UG. The TMP86PM23UG has PROM in place of the mask ROM which is included in the TMP86CM23AUG. The configuration and function are the same as the TMP86CM23AUG. In addition, TMP86PM23UG operates as the single clock mode when releasing reset. When using the dual clock mode, oscillate a low-frequency clock by [SET. (SYSCR2). XTEN] command at the beginning of program.

18.1 Operating mode

The TMP86PM23UG has MCU mode and PROM mode.

18.1.1 MCU mode

The MCU mode is set by fixing the TEST/VPP pin to the low level. (TEST/VPP pin cannot be used open because it has no built-in pull-down resistor).

18.1.1.1 Program Memory

The TMP86PM23UG has 32K bytes built-in one-time-PROM (addresses 8000 to FFFFH in the MCU mode, addresses 0000 to 7FFFH in the PROM mode).
When using TMP86PM23UG for evaluation of mask ROM products, the program is written in the program storing area shown in Figure 18-1.

Since the TMP86PM23UG supports several mask ROM sizes, check the difference in memory size and program storing area between the one-time PROM and the mask ROM to be used.

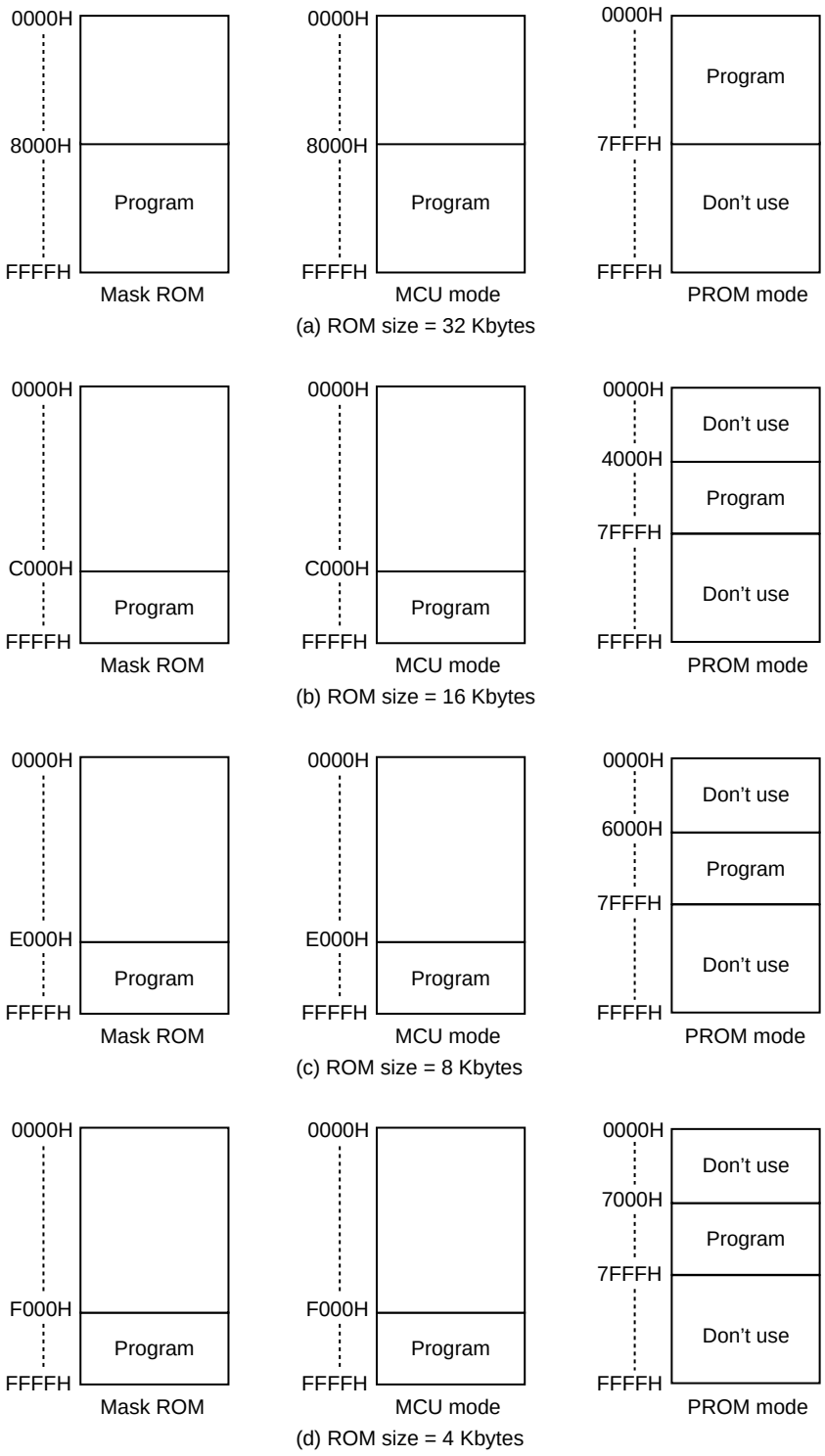


Figure 18-1 Program Memory Area

Note: The area that is not in use should be set data to FFH, or a general-purpose PROM programmer should be set only in the program memory area to access.

18.1.1.2 Data Memory

TMP86PM23UG has a built-in 1536 bytes Data memory (static RAM).

18.1.1.3 Input/Output Circuiry

1. Control pins

   The control pins of the TMP86PM23UG are the same as those of the TMP86CM23AUG except that the TEST pin does not have a built-in pull-down resistor.

2. I/O ports

   The I/O circuitries of the TMP86PM23UG I/O ports are the same as those of the TMP86CM23AUG.

18.1.2 PROM mode

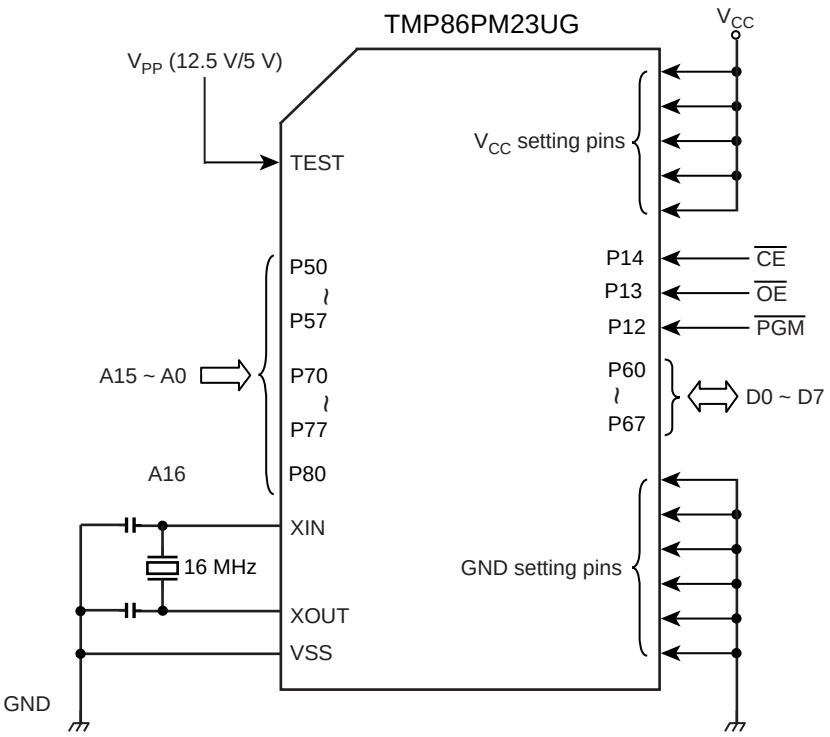
The PROM mode is set by setting the $\overline{\text{RESET}}$ pin, TEST pin and other pins as shown in Table 18-1 and Figure 18-2. The programming and verification for the internal PROM is acheived by using a general-purpose PROM programmer with the adaptor socket.

Table 18-1 Pin name in PROM mode

Pin name (PROM mode)	I/O	Function	Pin name (MCU mode)
A16	Input	Program memory address input	P80
A15 to A8	Input	Program memory address input	P77 to P70
A7 to A0	Input	Program memory address input	P57 to P50
D7 to D0	Input/Output	Program memory data input/output	P67 to P60
$\overline{\text{CE}}$	Input	Chip enable signal input	P14
$\overline{\text{OE}}$	Input	Output enable signal input	P13
$\overline{\text{PGM}}$	Input	Program mode signal input	P12
VPP	Power supply	+12.75V/5V (Power supply of program)	TEST
VCC	Power supply	+6.25V/5V	VDD
GND	Power supply	0V	VSS
VCC	Setting pin	Fix to "H" level in PROM mode	AVDD,P11,P21
GND	Setting pin	Fix to "L" level in PROM mode	VAREF,P15,P20,P22
$\overline{\text{RESET}}$	Setting pin	Fix to "L" level in PROM mode	$\overline{\text{RESET}}$
XIN (CLK)	Input	Set oscillation with resonator In case of external CLK input, set CLK to XIN and set XOUT to open.	XIN
XOUT	Output		XOUT

Note 1: The high-speed program mode can be used. The setting is different according to the type of PROM programmer to use, refer to each description of PROM programmer.
TMP86PM23UG does not support the electric signature mode, apply the ROM type of PROM programmer to TC571000D/AD.

Always set the adapter socket switch to the "N" side when using TOSHIBA's adaptor socket.



Note 1: EPROM adaptor socket (TC571000 • 1M bit EPROM)

Note 2: PROM programmer connection adaptor sockets
BM11698 for TMP86PM23UG

Note 3: Inside pin name for TMP86PM23UG
Outside pin name for EPROM

Figure 18-2 PROM mode setting

18.1.2.1 Programming Flowchart (High-speed program writing)

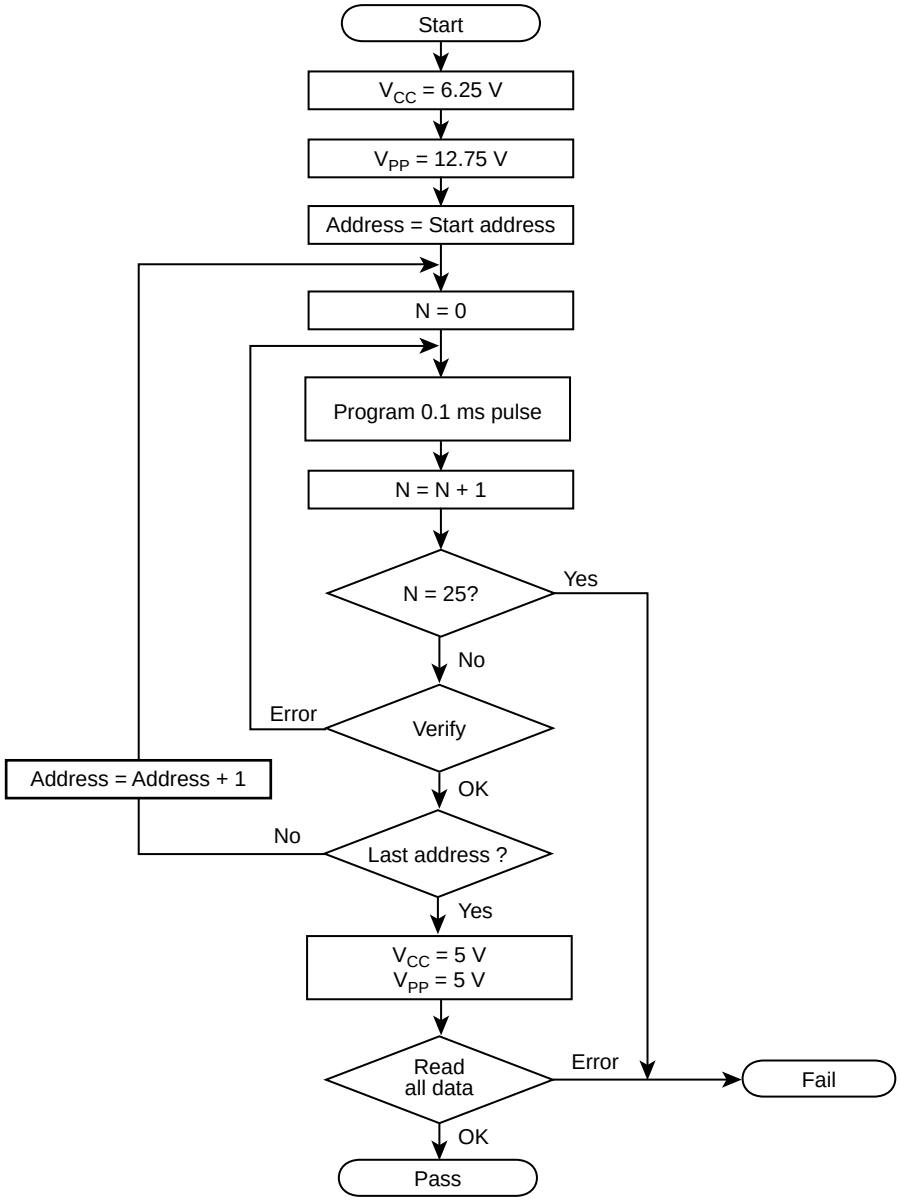


Figure 18-3 Programming Flowchart

The high-speed programming mode is set by applying Vpp=12.75V (programming voltage) to the Vpp pin when the Vcc = 6.25 V. After the address and data are fixed, the data in the address is written by applying 0.1[msec] of low level program pulse to $\overline{PGM}$ pin. Then verify if the data is written.
If the programmed data is incorrect, another 0.1[msec] pulse is applied to $\overline{PGM}$ pin. This programming procedure is repeated until correct data is read from the address (maximum of 25 times).
Subsequently, all data are programmed in all address. When all data were written, verfy all address under the condition Vcc=Vpp=5V.

18.1.2.2 Program Writing using a General-purpose PROM Programmer

1. Recommended OTP adaptor

   BM11698 for TMP86PM23UG

2. Setting of OTP adaptor

   Set the switch (SW1) to "N" side.

3. Setting of PROM programmer

   a. Set PROM type to TC571000D/AD.

      Vpp: 12.75 V (high-speed program writing mode)

   b. Data transmission (or Copy) (Note 1)

      The PROM of TMP86PM23UG is located on different address; it depends on operating mode: MCU mode and PROM mode. When you write the data of ROM for mask ROM products, the data shuold be transferred (or copied) from the address for MCU mode to that for PROM mode before writing operation is executed. For the applicable program areas of MCU mode and PROM mode are different, refer to TMP86PM23UG" Figure 18-1 Program Memory Area ".

      Example: In the block transfer (copy) mode, executed as below.

      32KB ROM capacity: 08000~0FFFFH → 00000~07FFFH

      16KB ROM capacity: 0C000~0FFFFH → 04000~07FFFH

      8KB ROM capacity : 0E000~0FFFFH → 06000~07FFFH

      4KB ROM capacity : 0F000~0FFFFH → 07000~07FFFH

   c. Setting of the program address (Note 1)

      Start address: 0000H (When 16 KB ROM capacity, start address is 4000H.
      When 8 KB ROM capacity, start address is 6000H.
      When 4KB ROM capacity, start address is 7000H.)

      End address: 7FFFH

4. Writting

   Write and verify according to the above procedure "Setting of PROM programmer".

Note 1: For the setting method, refer to each description of PROM programmer.
Make sure to set the data of address area that is not in use to FFH.

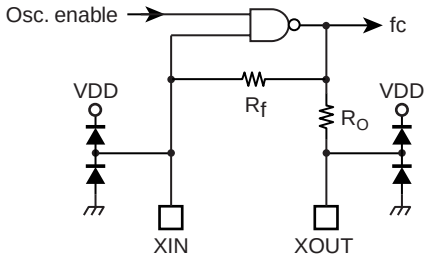
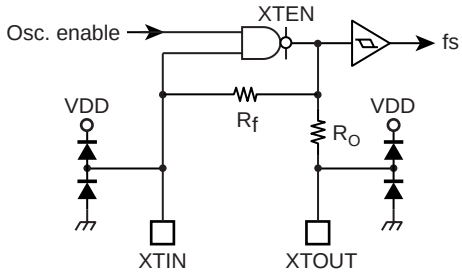
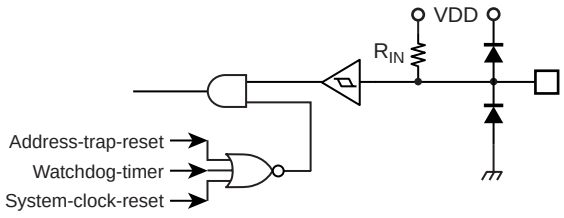
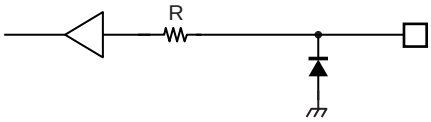
Note 2: When setting MCU to the adaptor or when setting the adaptor to the PROM programmer, set the first pin of the adaptor and that of PROM programmer socket matched. If the first pin is conversely set, MCU or adaptor or programmer would be damaged.

Note 3: The TMP86PM23UG does not support the electric signature mode.
If PROM programmer uses the signature, the device would be damaged because of applying voltage of 12±0.5V to pin 9(A9) of the address. Don't use the signature.

19. Input/Output Circuitry

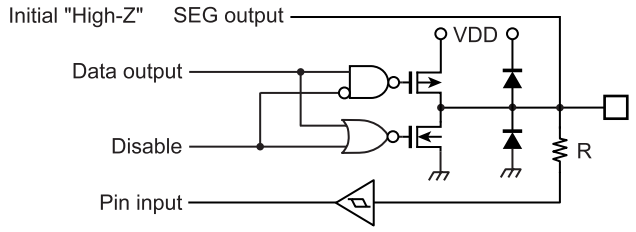
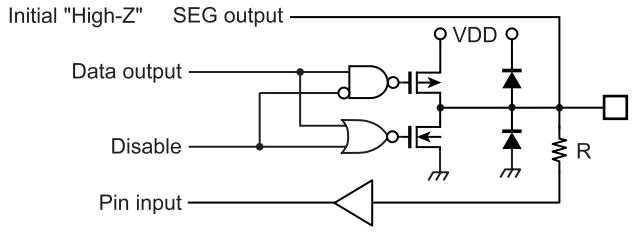
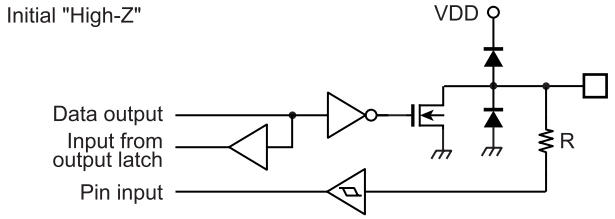
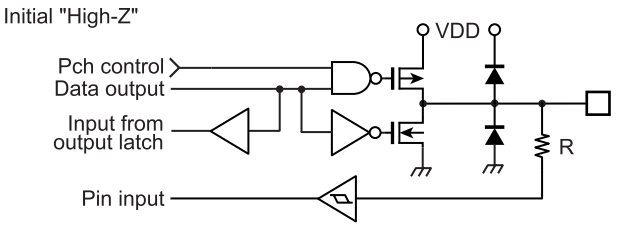
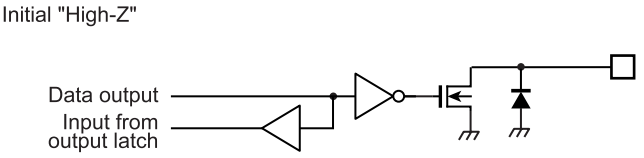
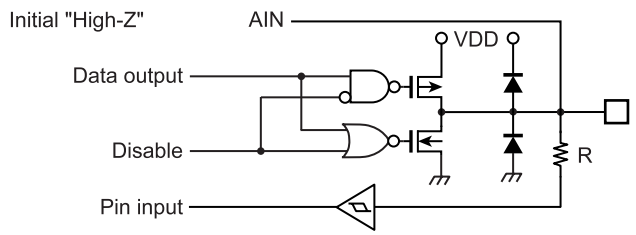
19.1 Control Pins

The input/output circuitries of the TMP86PM23UG 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		Resonator connecting pins (Low-frequency) R_f = 6 MΩ (typ.) R_O = 220 kΩ (typ.)
$\overline{\text{RESET}}$	Input		Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.)
TEST	Input		Without pull-down resistor R = 1 kΩ (typ.) Fix the TEST pin at low-level in MCU mode.

Note: The TEST pin of the TMP86PM23/PS23 does not have a pull-down resistor and protect diode (D_1). Fix the TEST pin at low-level in MCU mode.

19.2 Input/Output Ports

Port	I/O	Input/Output Circuitry	Remarks
P1	I/O		Tri-state I/O Hysteresis input R = 100 Ω (typ.) LCD segment output
P5 P7 P8	I/O		Tri-state I/O R = 100 Ω (typ.) LCD segment output
P2	I/O		Sink open drain output Hysteresis input R = 100 Ω (typ.)
P34 to P30	I/O		Sink open drain output or C-MOS output Hysteresis input High current output (Nch) (Only P33, P34 port) R = 100 Ω (typ.)
P37 to P35	Output		Sink open drain output High current output (Nch)
P6	I/O		Tri-state I/O Hysteresis input AIN input R = 100 Ω (typ.)

20. Electrical Characteristics

20.1 Absolute Maximum Ratings

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

(V_{SS} = 0 V)

Parameter	Symbol	Pins	Ratings	Unit
Supply voltage	V_{DD}		−0.3 to 6.5	V
Program voltage	V_{PP}	TEST/V_{PP}	−0.3 to 13.0	
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_{OUT1}	P1, P30 to P34, P5, P6, P7, P8 port	−1.8	mA
	I_{OUT2}	P1, P2, P30 to P32, P5, P6, P7, P8 port	3.2	
	I_{OUT3}	P33 to P37 port	30	
Output current (Total)	ΣI_{OUT1}	P1, P30 to P34, P5, P6, P7, P8 port	−30	
	ΣI_{OUT2}	P1, P2, P30 to P32, P5, P6, P7, P8 port	60	
	ΣI_{OUT3}	P33 to P37 port	80	
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	

20.2 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	3.5	5.5	V
				IDLE0, 1, 2 mode			
			fc = 8 MHz	NORMAL1, 2 mode	2.7		
				IDLE0, 1, 2 mode			
			fc = 4.2 MHz	NORMAL1, 2 mode	1.8		
				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} = 1.8 V to 5.5 V		1.0	4.2	MHz
			V_{DD} = 2.7 V to 5.5 V			8.0	
			V_{DD} = 3.5 V to 5.5 V			16.0	
	fs	XTIN, XTOUT			30.0	34.0	kHz

20.3 DC Characteristics

(V_{SS} = 0 V, Topr = –40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Hysteresis voltage	V_{HS}	Hysteresis input		–	0.9	–	V
Input current	I_{IN1}	TEST	V_{DD} = 5.5 V, V_{IN} = 5.5 V/0 V	–	–	±2	μA
	I_{IN2}	Sink open drain, Tri-state					
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$					
Input resistance	R_{IN1}	$\overline{RESET}$ pull-up		100	220	450	kΩ
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 and P3 port	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	–	–	0.4	
Output low current	I_{OL}	High current port (P33 to P37 port)	V_{DD} = 4.5 V, V_{OL} = 1.0 V	–	20	–	mA
Supply current in NORMAL 1, 2 mode	I_{DD}		V_{DD} = 5.5 V V_{IN} = 5.3/0.2 V fc = 16 MHz fs = 32.768 kHz	–	11.5	16.5	mA
Supply current in IDLE 0, 1, 2 mode				–	7.0	11.0	
Supply current in SLOW 1 mode			V_{DD} = 3.0 V V_{IN} = 2.8/0.2 V fs = 32.768 kHz LCD drive is not enable.	–	19.0	33.0	μA
Supply current in SLEEP 1 mode				–	13.0	26.0	
Supply current in SLEEP 0 mode				–	5.0	14.0	
Supply current in STOP mode			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V	–	0.5	10	μA
Segment output low resistance	R_{OS1}	SEG pin		–	20	–	kΩ
Common output low resistance	R_{OC1}	COM pin			20		
Segment output high resistance	R_{OS2}	SEG pin			200		
Common output high resistance	R_{OC2}	COM pin			200		
Segment/common output voltage	$V_{O2/3}$	SEG/COM pin	V_{DD} = 5.0 V V_{LC} = 2.0 V	3.8	–	4.2	V
	$V_{O1/2}$			3.3		3.7	
	$V_{O1/3}$			2.8		3.2	

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.

Note 5: Output resistors ROS and ROC indicate "ON" when switching levels.

Note 6: $V_{O2/3}$ indicates the output voltage at the 2/3 level when operating in the 1/4 or 1/3 duty mode.

Note 7: $V_{O1/2}$ indicates the output voltage at the 1/2 level when operating in the 1/2 duty or static mode.

Note 8: $V_{O1/3}$ indicates the output voltage at the 1/3 level when operating in the 1/4 or 1/3 duty mode.

Note 9: When using LCD, it is necessary to consider values of $R_{OS1/2}$ and $R_{OC1/2}$.

20.4 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}			
Analog reference voltage range (Note4)	ΔV_{AREF}		3.5	–	–	
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 5.5 V V_{SS} = 0.0 V	–	0.6	1.0	mA
Non linearity error		V_{DD} = A_{VDD} = 5.0 V V_{SS} = 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}			
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} = 0.0 V	–	0.5	0.8	mA
Non linearity error		V_{DD} = A_{VDD} = 2.7 V V_{SS} = 0.0 V V_{AREF} = 2.7 V	–	–	±2	LSB
Zero point error			–	–	±2	
Full scale error			–	–	±2	
Total error			–	–	±2	

(V_{SS} = 0.0 V, 2.0 V ≤ V_{DD} < 2.7 V, Topr = –40 to 85°C) (Note5)

(V_{SS} = 0.0 V, 1.8 V ≤ V_{DD} < 2.0 V, Topr = –10 to 85°C) (Note5)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} – 0.9	–	A_{VDD}	V
Power supply voltage of analog control circuit (Note6)	A_{VDD}		V_{DD}			
Analog reference voltage range (Note4)	ΔV_{AREF}	1.8 V ≤ V_{DD} < 2.0 V	1.8	–	–	
		2.0 V ≤ V_{DD} < 2.7 V	2.0	–	–	
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 2.7 V V_{SS} = 0.0 V	–	0.3	0.5	mA
Non linearity error		V_{DD} = A_{VDD} = 1.8 V V_{SS} = 0.0 V V_{AREF} = 1.8 V	–	–	±4	LSB
Zero point error			–	–	±4	
Full scale error			–	–	±4	
Total error			–	–	±4	

Note 1: The total error includes all errors except a quantization error, and is defined as a maximum deviation from the ideal conversion line.

Note 2: Conversion time is different in recommended value by power supply voltage.
About conversion time, please refer to “Register Configuration”.

Note 3: Please use input voltage to AIN input Pin in limit of V_{AREF} 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: When AD is used with $V_{DD} < 2.7$ V, the guaranteed temperature range varies with the operating voltage.

Note 6: The A_{VDD} pin should be fixed on the V_{DD} level even though AD converter is not used.

20.5 AC Characteristics

(V_{SS} = 0 V, V_{DD} = 3.5 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}					

(V_{SS} = 0 V, V_{DD} = 1.8 to 5.5 V, Topr = –40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcy	NORMAL1, 2 mode	0.95	–	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 = 4.2 MHz	–	119.05	–	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}					

20.6 Timer Counter 1 input (ECIN) Characteristics

(V_{SS} = 0 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition		Min	Typ.	Max	Unit
TC1 input (ECIN input)	t_{TC1}	Frequency measurement mode V_{DD} = 3.5 to 5.5 V	Single edge count	–	–	16	MHz
			Both edge count	–	–		
		Frequency measurement mode V_{DD} = 2.7 to 5.5 V	Single edge count	–	–	8	
			Both edge count	–	–		
		Frequency measurement mode V_{DD} = 1.8 to 5.5 V	Single edge count	–	–	4.2	
			Both edge count	–	–		

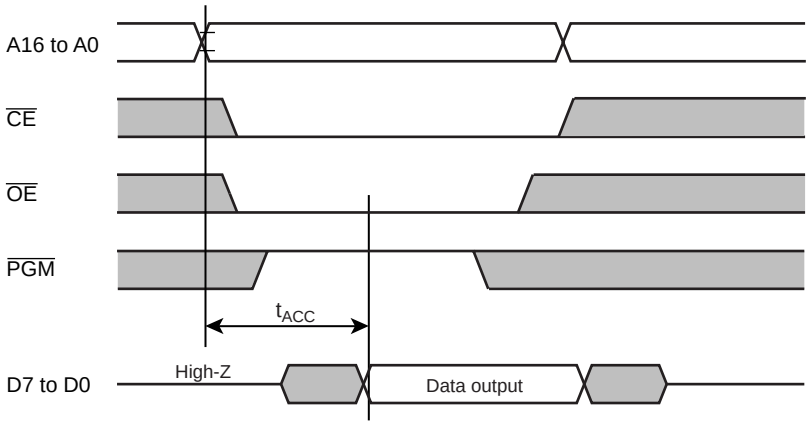
20.7 DC Characteristics, AC Characteristics (PROM mode)

20.7.1 Read operation in PROM mode

(V_{SS} = 0 V, Topr = –40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
High level input voltage (TTL)	V_{IH4}		2.2	–	V_{CC}	V
Low level input voltage (TTL)	V_{IL4}		0	–	0.8	
Power supply	V_{CC}		4.75	5.0	5.25	
Program supply of program	V_{PP}					
Address access time	t_{ACC}	V_{CC} = 5.0 ± 0.25 V	–	1.5tcyc + 300	–	ns

Note: tcyc = 500 ns at f_{CLK} = 8 MHz

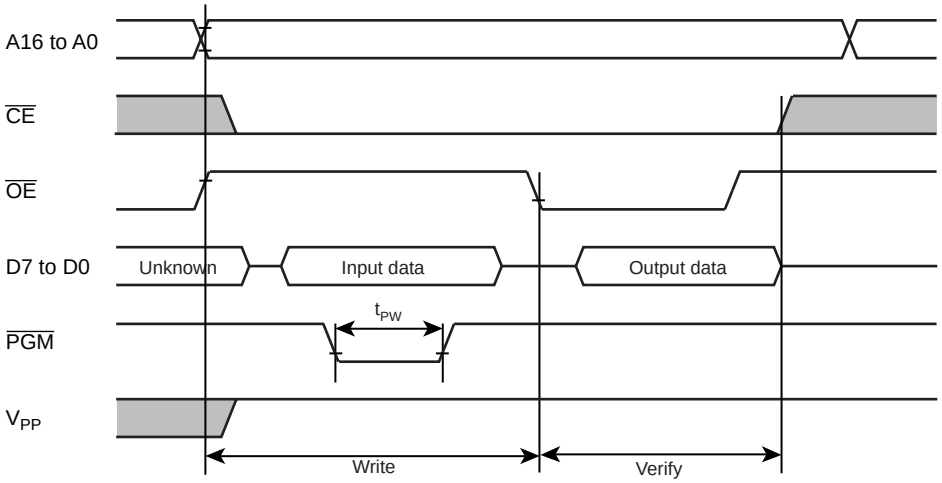


20.7.2 Program operation (High-speed)

(Topr = 25 ± 5 °C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
High level input voltage (TTL)	V_{IH4}		2.2	–	V_{CC}	V
Low level input voltage (TTL)	V_{IL4}		0	–	0.8	
Power supply	V_{CC}		6.0	6.25	6.5	
Program supply of program	V_{PP}		12.5	12.75	13.0	
Pulse width of initializing program	t_{PW}	V_{CC} = 6.0 V	0.095	0.1	0.105	ms

High-speed program writing

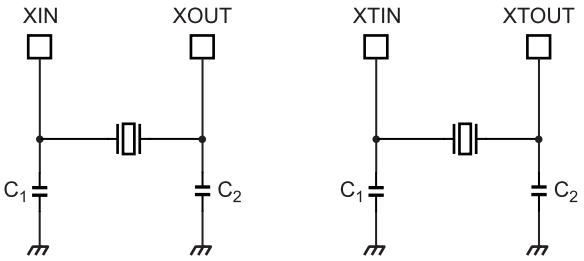


Note 1: The power supply of V_{PP} (12.75 V) must be set power-on at the same time or the later time for a power supply of V_{CC} and must be clear power-on at the same time or early time for a power supply of V_{CC}.

Note 2: The pull-up/pull-down device on the condition of V_{PP} = 12.75 V ± 0.25 V causes a damage for the device. Do not pull-up/pull-down at programming.

Note 3: Use the recommended adapter and mode.
Using other than the above condition may cause the trouble of the writting.

20.8 Recommended Oscillating Conditions



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

Note 1: A quartz resonator can be used for high-frequency oscillation only when V_{DD} is 2.7 V or above. If V_{DD} is below 2.7 V, use a ceramic resonator.

Note 2: 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 3: 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

20.9 Handling Precaution

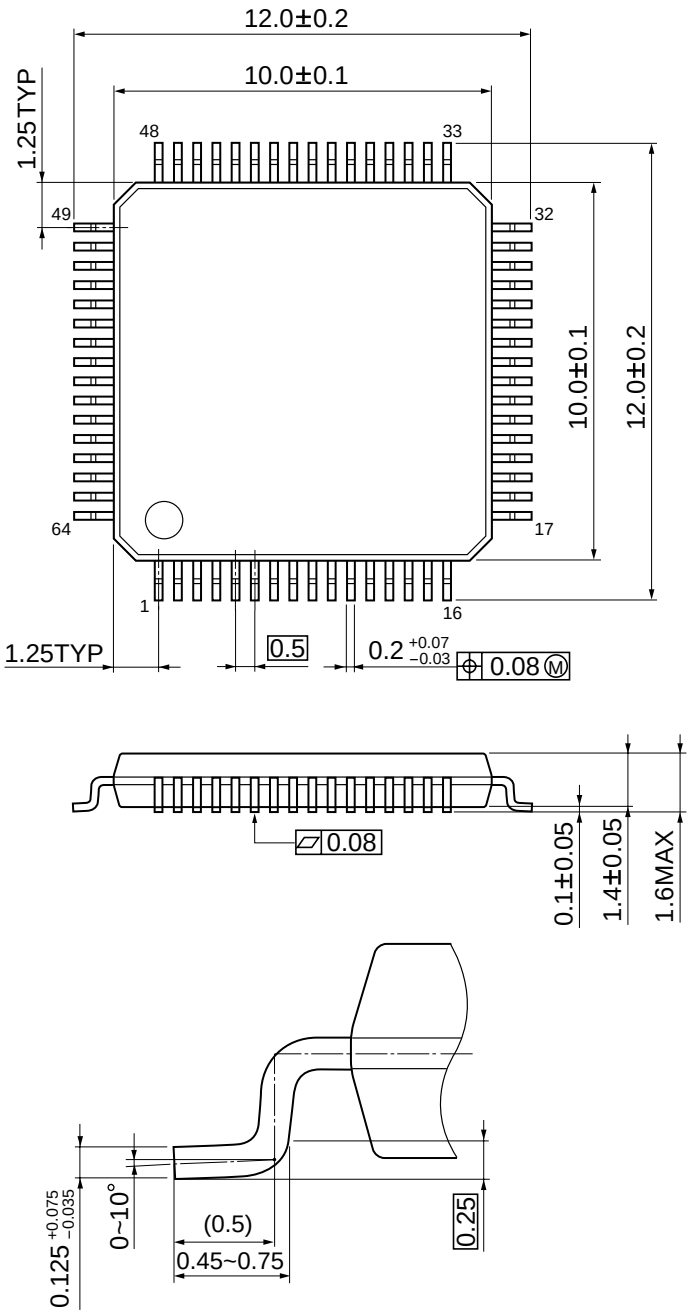
- The solderability test conditions for lead-free products (indicated by the suffix G in product name) are shown below.
  1. When using the Sn-37Pb solder bath
     Solder bath temperature = 230 °C
     Dipping time = 5 seconds
     Number of times = once
     R-type flux used
  2. When using the Sn-3.0Ag-0.5Cu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

  Note: The pass 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.

21. Package Dimensions

LQFP64-P-1010-0.50E Rev 02

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.