TOSHIBA

8 Bit Microcontroller

TLCS-870/C Series

TMP86CK74AFG

TOSHIBA CORPORATION

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

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

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

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

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

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

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

Revision History

Date	Revision	
2007/10/9	1	First Release

Table of Contents

4. Special Function Register (SFR)

5. I/O Ports

6. Watchdog Timer (WDT)

7. Time Base Timer (TBT)

8. 16-Bit TimerCounter 1 (TC1)

9. 16-Bit Timer/Counter2 (TC2)

10. 8-Bit TimerCounter 3 (TC3)

11. 8-Bit TimerCounter 4 (TC4)

12. Synchronous Serial Interface (SIO)

18. 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

TMP86CK74AFG

Product No.	ROM (MaskROM)	RAM	Package	OTP MCU	Emulation Chip
TMP86CK74AFG	24576 bytes	1024 bytes	QFP80-P-1420-0.80M	TMP86PM74AFG	TMP86C974XB

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. 17interrupt sources (External : 6 Internal : 11)
3. Input / Output ports (70 pins)

   Large current output: 2pins (Typ. 20mA), LED direct drive

4. Watchdog Timer
5. Prescaler
   - Time base timer
   - Divider output function
6. 16-bit timer counter: 1 ch
   - Timer, External trigger, Window, Pulse width measurement,

     Event counter, Programmable pulse generate (PPG) modes
7. 16-bit timer counter: 1 ch
   - Timer, Event counter, Window modes
8. 8-bit timer counter : 1 ch
   - Timer, Event counter, Capture modes
9. 8-bit timer counter : 1 ch
   - Timer, Event counter, Pulse width modulation (PWM) output,

Programmable divider output (PDO) modes

10. Serial Interface
    - 8-bit SIO :1 channel (32 bytes Buffer)
11. 8-bit successive approximation type AD converter (with sample hold)

    Analog inputs: 8ch
12. Key-on wakeup : 4 ch
13. Vacuum flouorescent tube driver (automatic display)
    - Programmable grid scan
    - High breakdown voltage ports(MAX 40 V × 37 bits)
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:

    4.5 V to 5.5 V at 16MHz /32.768 kHz

    2.7 V to 5.5 V at 8 MHz /32.768 kHz

1.2 Pin Assignment

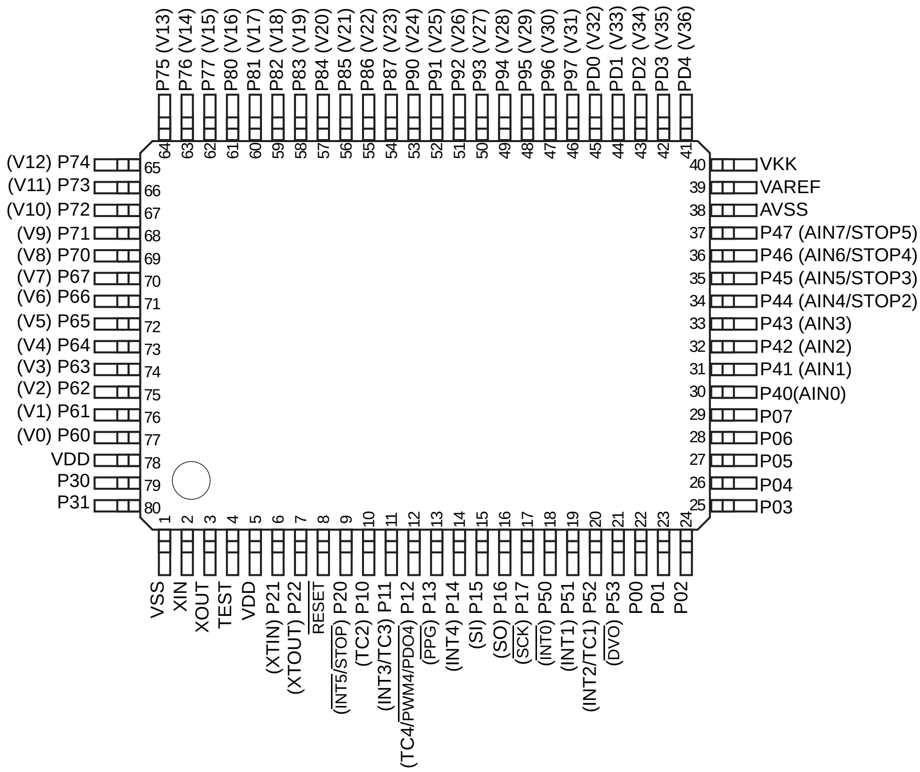


Figure 1-1 Pin Assignment

1.3 Block Diagram

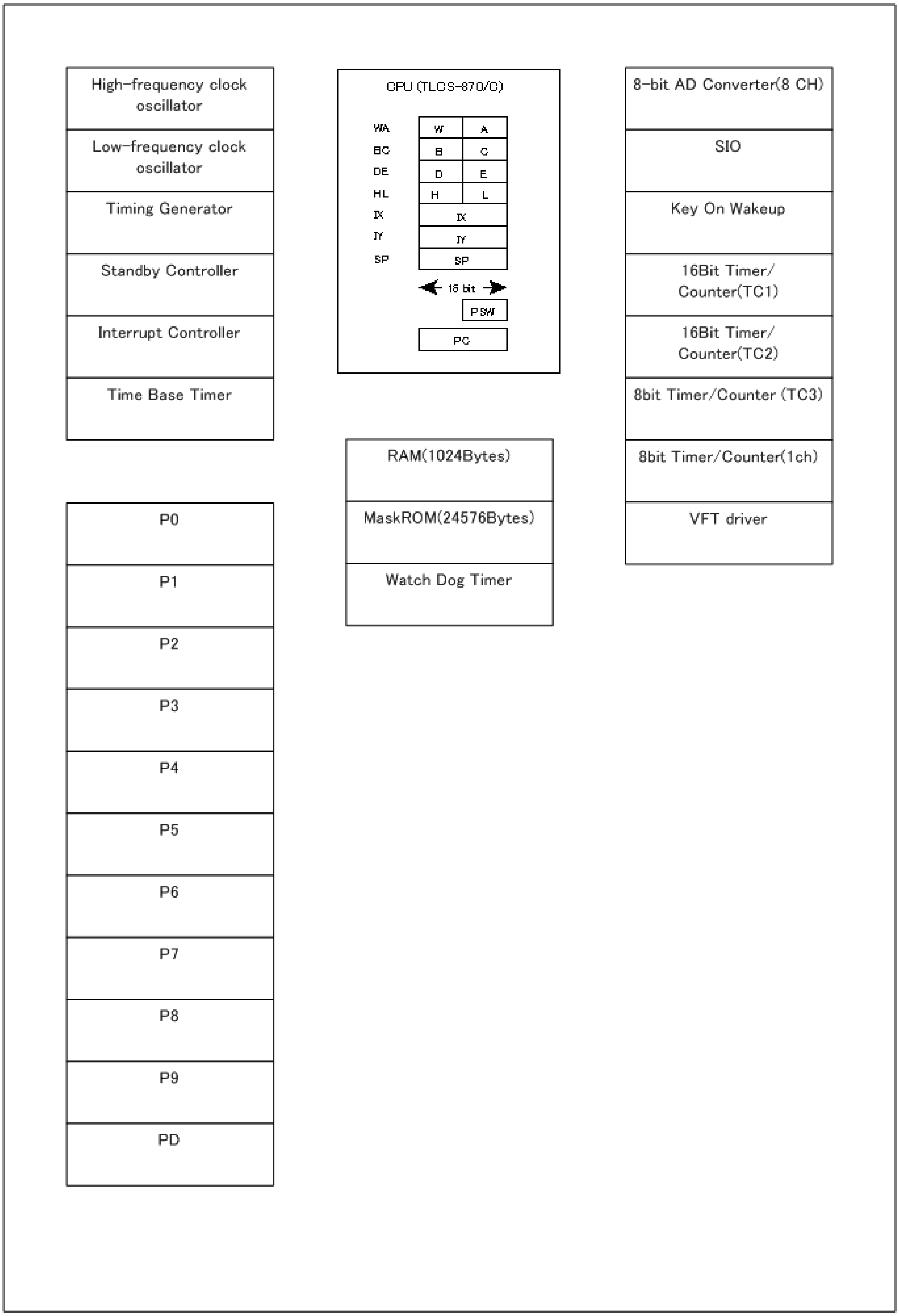


Figure 1-2 Block Diagram

1.4 Pin Names and Functions

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

Pin Name	Pin Number	Input/Output	Functions
P07	29	IO	PORT07
P06	28	IO	PORT06
P05	27	IO	PORT05
P04	26	IO	PORT04
P03	25	IO	PORT03
P02	24	IO	PORT02
P01	23	IO	PORT01
P00	22	IO	PORT00
P17 $\overline{\text{SCK}}$	17	IO IO	PORT17 Serial clock input/output
P16 SO	16	IO O	PORT16 Serial data output
P15 SI	15	IO I	PORT15 Serial data input
P14 INT4	14	IO I	PORT14 External interrupt 4 input
P13 $\overline{\text{PPG}}$	13	IO O	PORT13 PPG output
P12 $\overline{\text{PWM4/PDO4}}$ TC4	12	IO O I	PORT12 PWM4/PDO4 output TC4 input
P11 TC3 INT3	11	IO I I	PORT11 TC3 pin input External interrupt 3 input
P10 TC2	10	IO I	PORT10 TC2 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
P31	80	IO	PORT31
P30	79	IO	PORT30
P47 AIN7 STOP5	37	IO I I	PORT47 AD converter analog input 7 STOP5 input
P46 AIN6 STOP4	36	IO I I	PORT46 AD converter analog input 6 STOP4 input

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

Pin Name	Pin Number	Input/Output	Functions
P45 AIN5 STOP3	35	IO I I	PORT45 AD converter analog input 5 STOP3 input
P44 AIN4 STOP2	34	IO I I	PORT44 AD converter analog input 4 STOP2 input
P43 AIN3	33	IO I	PORT43 AD converter analog input 3
P42 AIN2	32	IO I	PORT42 AD converter analog input 2
P41 AIN1	31	IO I	PORT41 AD converter analog input 1
P40 AIN0	30	IO I	PORT40 AD converter analog input 0
P53 $\overline{\text{DVO}}$	21	IO O	PORT53 Divider Output
P52 TC1 INT2	20	IO I I	PORT52 TC1 input External interrupt 2 input
P51 INT1	19	IO I	PORT51 External interrupt 1 input
P50 $\overline{\text{INT0}}$	18	IO I	PORT50 External interrupt 0 input
P67 V7	70	IO O	PORT67 Grid output7
P66 V6	71	IO O	PORT66 Grid output6
P65 V5	72	IO O	PORT65 Grid output5
P64 V4	73	IO O	PORT64 Grid output4
P63 V3	74	IO O	PORT63 Grid output3
P62 V2	75	IO O	PORT62 Grid output2
P61 V1	76	IO O	PORT61 Grid output1
P60 V0	77	IO O	PORT60 Grid output0
P77 V15	62	IO O	PORT77 Grid output15
P76 V14	63	IO O	PORT76 Grid output14
P75 V13	64	IO O	PORT75 Grid output13
P74 V12	65	IO O	PORT74 Grid output12

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

Pin Name	Pin Number	Input/Output	Functions
P73 V11	66	IO O	PORT73 Grid output11
P72 V10	67	IO O	PORT72 Grid output10
P71 V9	68	IO O	PORT71 Grid output9
P70 V8	69	IO O	PORT70 Grid output8
P87 V23	54	IO O	PORT87 Segment output23
P86 V22	55	IO O	PORT86 Segment output22
P85 V21	56	IO O	PORT85 Segment output21
P84 V20	57	IO O	PORT84 Segment output20
P83 V19	58	IO O	PORT83 Segment output19
P82 V18	59	IO O	PORT82 Segment output18
P81 V17	60	IO O	PORT81 Segment output17
P80 V16	61	IO O	PORT80 Segment output16
P97 V31	46	IO O	PORT97 Segment output31
P96 V30	47	IO O	PORT96 Segment output30
P95 V29	48	IO O	PORT95 Segment output29
P94 V28	49	IO O	PORT94 Segment output28
P93 V27	50	IO O	PORT93 Segment output27
P92 V26	51	IO O	PORT92 Segment output26
P91 V25	52	IO O	PORT91 Segment output25
P90 V24	53	IO O	PORT90 Segment output24
PD4 V36	41	IO O	PORTD4 Segment output36
PD3 V35	42	IO O	PORTD3 Segment output35
PD2 V34	43	IO O	PORTD2 Segment output34

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

Pin Name	Pin Number	Input/Output	Functions
PD1 V33	44	IO O	PORTD1 Segment output33
PD0 V32	45	IO O	PORTD0 Segment output32
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	39	I	Analog reference voltage input (High)
AVSS	38	I	AD circuit power supply
VDD	78	I	Power Supply
VSS	1	I	0V(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 TMP86CK74AFG memory is composed MaskROM, RAM, DBR(Data buffer register) and SFR(Special function register). They are all mapped in 64-Kbyte address space. Figure 2-1 shows the TMP86CK74AFG memory address map.

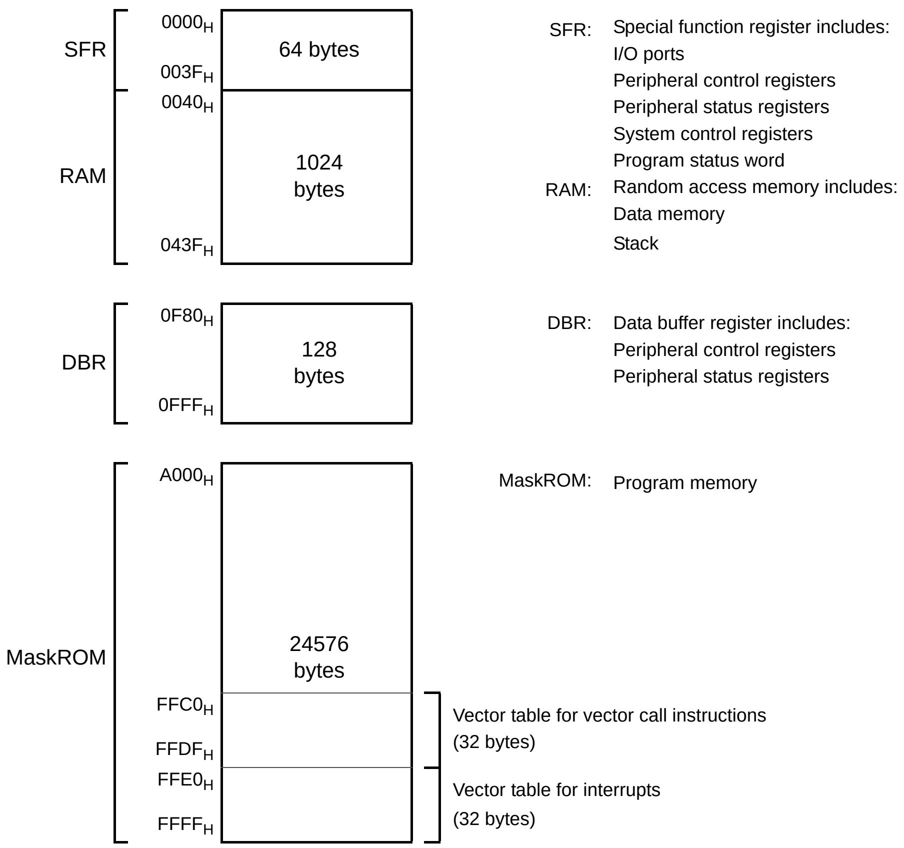


Figure 2-1 Memory Address Map

2.1.2 Program Memory (MaskROM)

The TMP86CK74AFG has a 24576 bytes (Address A000H to FFFFH) of program memory (MaskROM).

2.1.3 Data Memory (RAM)

The TMP86CK74AFG has 1024bytes (Address 0040H to 043FH) of internal RAM. The first 192 bytes (0040H to 00FFH) of the internal RAM are located in the direct area; instructions with shorten operations are available against such an area.

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

Example :Clears RAM to “00H”. (TMP86CK74AFG)

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

2.2 System Clock Controller

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

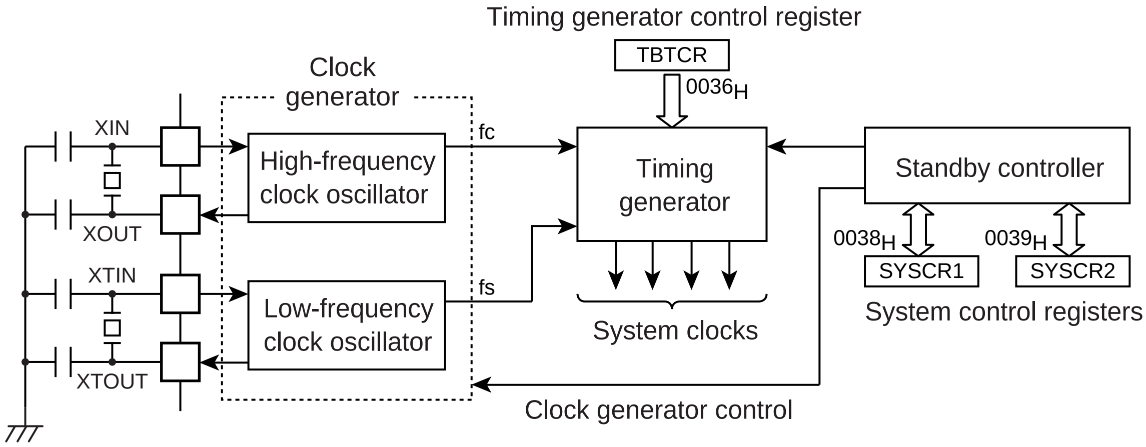


Figure 2-2 System Colck Control

2.2.1 Clock Generator

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

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

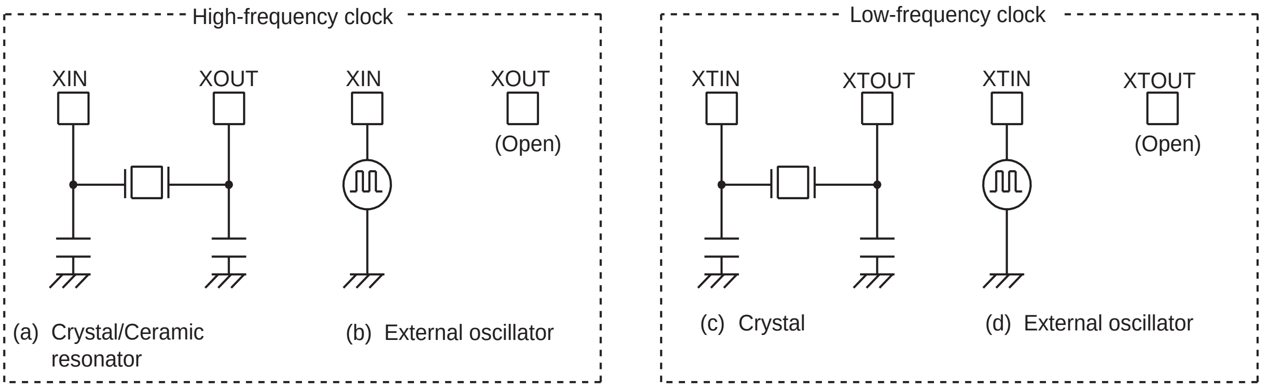


Figure 2-3 Examples of Resonator Connection

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

2.2.2 Timing Generator

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

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

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”.

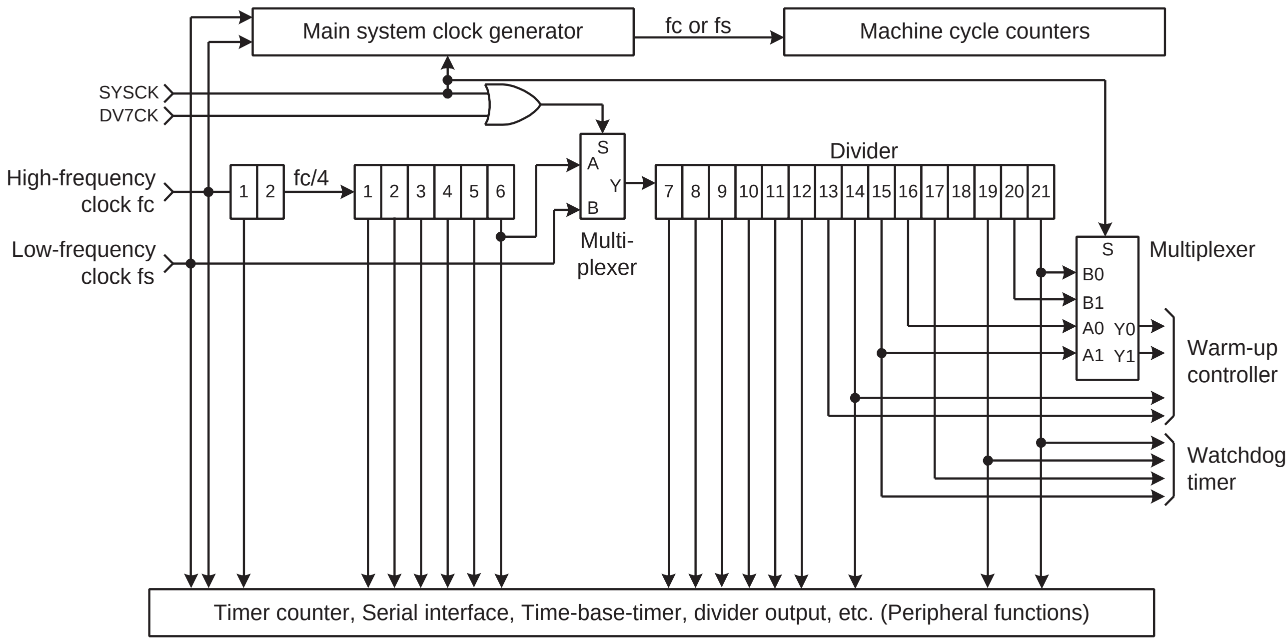


Figure 2-4 Configuration of Timing Generator

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.

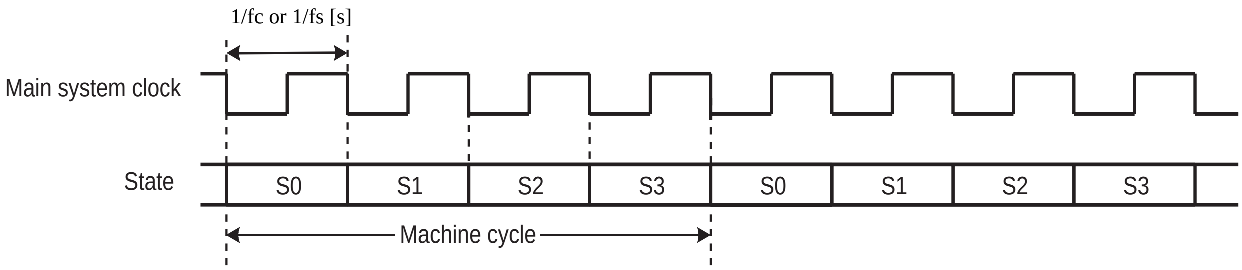


Figure 2-5 Machine Cycle

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 TMP86CK74AFG 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”, EF7 (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>.

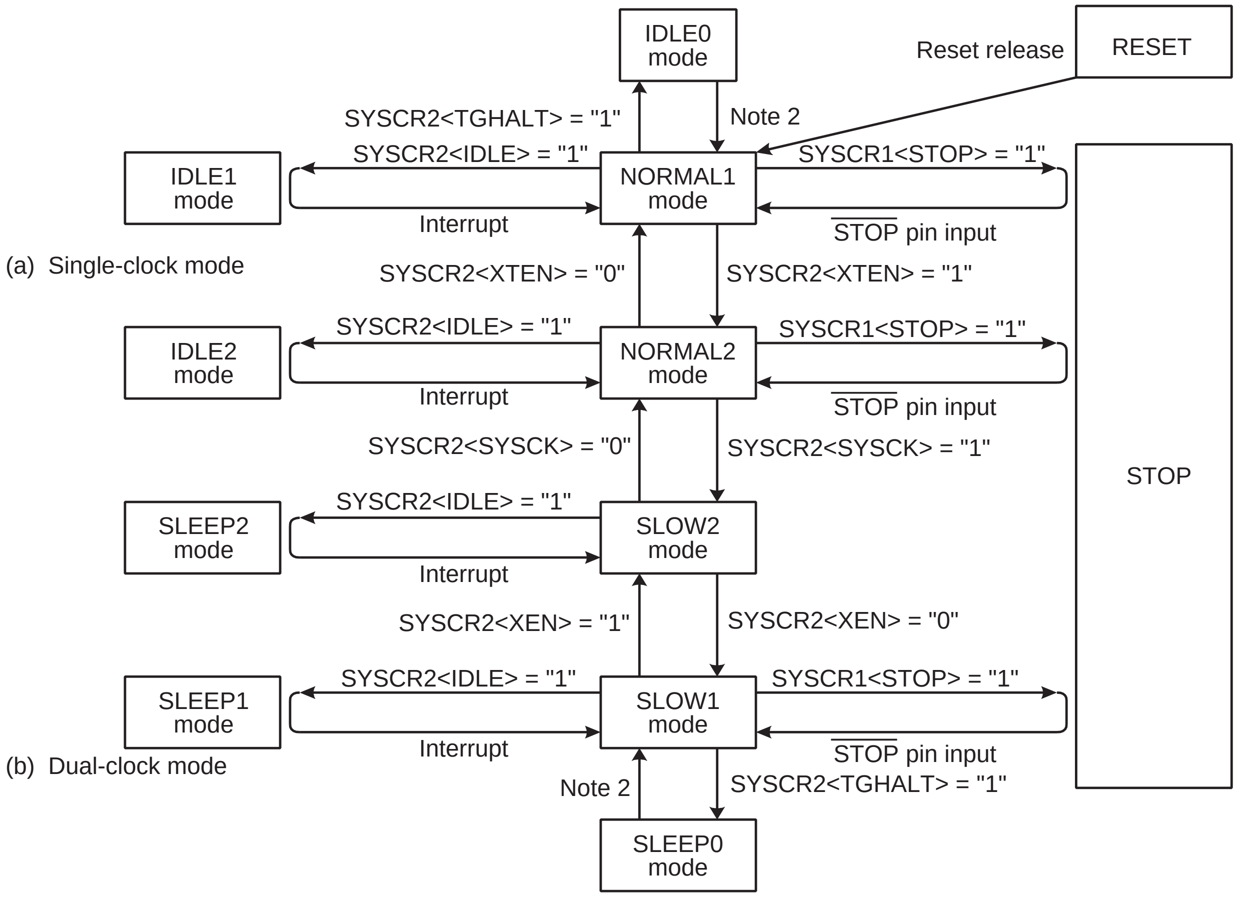
When SLEEP0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR<TBTCK>, the timing generator starts feeding the clock to all peripheral circuits.

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”, EF7 (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

<table>
<tr><th colspan="2" rowspan="2">Operating Mode</th><th colspan="2">Oscillator</th><th rowspan="2">CPU Core</th><th rowspan="2">TBT</th><th rowspan="2">Other Peripherals</th><th rowspan="2">Machine Cycle Time</th></tr>
<tr><th>High Frequency</th><th>Low Frequency</th></tr>
<tr><td rowspan="5">Single clock</td><td>RESET</td><td rowspan="4">Oscillation</td><td rowspan="5">Stop</td><td>Reset</td><td>Reset</td><td>Reset</td><td rowspan="4">4/fc [s]</td></tr>
<tr><td>NORMAL1</td><td>Operate</td><td rowspan="3">Operate</td><td rowspan="2">Operate</td></tr>
<tr><td>IDLE1</td><td rowspan="3">Halt</td></tr>
<tr><td>IDLE0</td><td rowspan="2">Halt</td></tr>
<tr><td>STOP</td><td>Stop</td><td>Halt</td><td>–</td></tr>
<tr><td rowspan="8">Dual clock</td><td>NORMAL2</td><td rowspan="4">Oscillation</td><td rowspan="7">Oscillation</td><td>Operate with high frequency</td><td rowspan="7">Operate</td><td rowspan="6">Operate</td><td rowspan="2">4/fc [s]</td></tr>
<tr><td>IDLE2</td><td>Halt</td></tr>
<tr><td>SLOW2</td><td>Operate with low frequency</td><td rowspan="5">4/fs [s]</td></tr>
<tr><td>SLEEP2</td><td>Halt</td></tr>
<tr><td>SLOW1</td><td rowspan="4">Stop</td><td>Operate with low frequency</td></tr>
<tr><td>SLEEP1</td><td rowspan="3">Halt</td></tr>
<tr><td>SLEEP0</td><td rowspan="2">Halt</td></tr>
<tr><td>STOP</td><td>Stop</td><td>Halt</td><td>–</td></tr>
</table>

System Control Register 1

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

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

Note 1: Always set RETM to “0” when 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	R/W
SYSCK	Main system clock select (Write)/main system clock monitor (Read)	0: High-frequency clock (NORMAL1/NORMAL2/IDLE1/IDLE2) 1: Low-frequency clock (SLOW1/SLOW2/SLEEP1/SLEEP2)	R/W
IDLE	CPU and watchdog timer control (IDLE1/2 and SLEEP1/2 modes)	0: CPU and watchdog timer remain active 1: CPU and watchdog timer are stopped (Start IDLE1/2 and SLEEP1/2 modes)	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)	R/W

Note 1: A reset is applied if both XEN and XTEN are cleared to “0”, XEN is cleared to “0” when SYSCK = “0”, or XTEN is cleared to “0” when SYSCK = “1”.

Note 2: *: Don’t care, TG: Timing generator, *; Don’t care

Note 3: Bits 3, 1 and 0 in SYSCR2 are always read as undefined value.

Note 4: Do not set IDLE and TGHALT to “1” simultaneously.

Note 5: Because returning from IDLE0/SLEEP0 to NORMAL1/SLOW1 is executed by the asynchronous internal clock, the period of IDLE0/SLEEP0 mode might be shorter than the period setting by TBTCR<TBTCK>.

Note 6: When IDLE1/2 or SLEEP1/2 mode is released, IDLE is automatically cleared to “0”.

Note 7: When IDLE0 or SLEEP0 mode is released, TGHALT is automatically cleared to “0”.

Note 8: Before setting TGHALT to “1”, be sure to stop peripherals. If peripherals are not stopped, the interrupt latch of peripherals may be set after IDLE0 or SLEEP0 mode is released.

2.2.4 Operating Mode Control

2.2.4.1 STOP mode

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

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

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

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

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

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

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

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

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

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

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

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

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

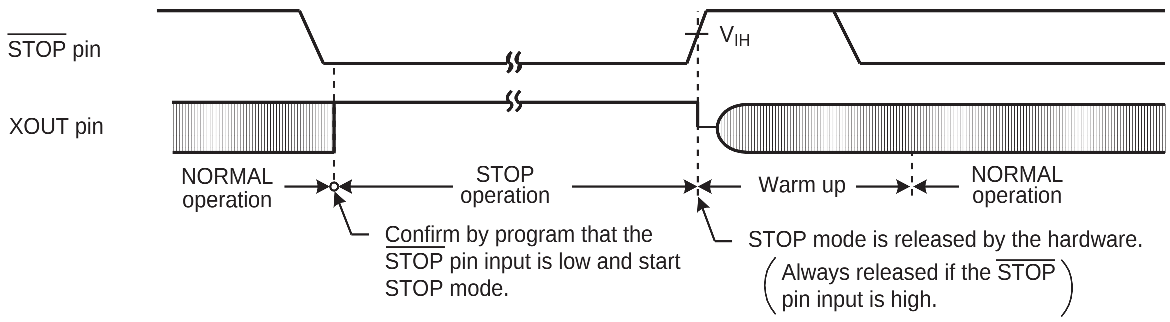


Figure 2-7 Level-sensitive Release Mode

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
```

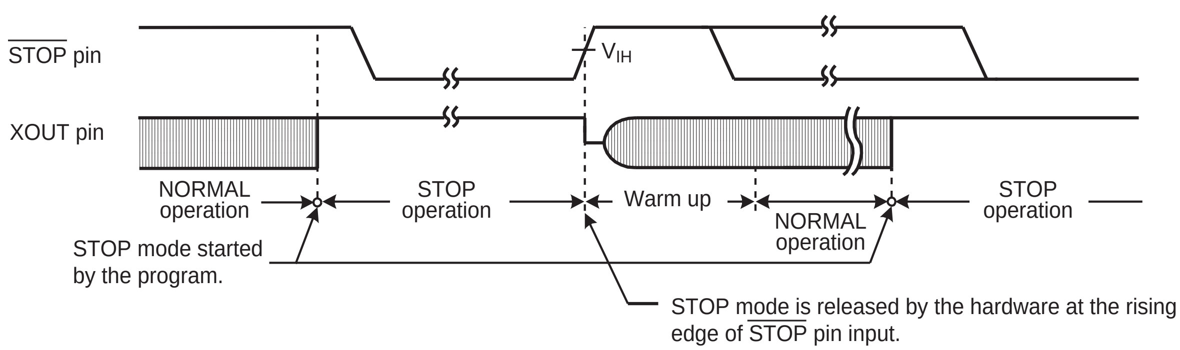


Figure 2-8 Edge-sensitive Release Mode

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.

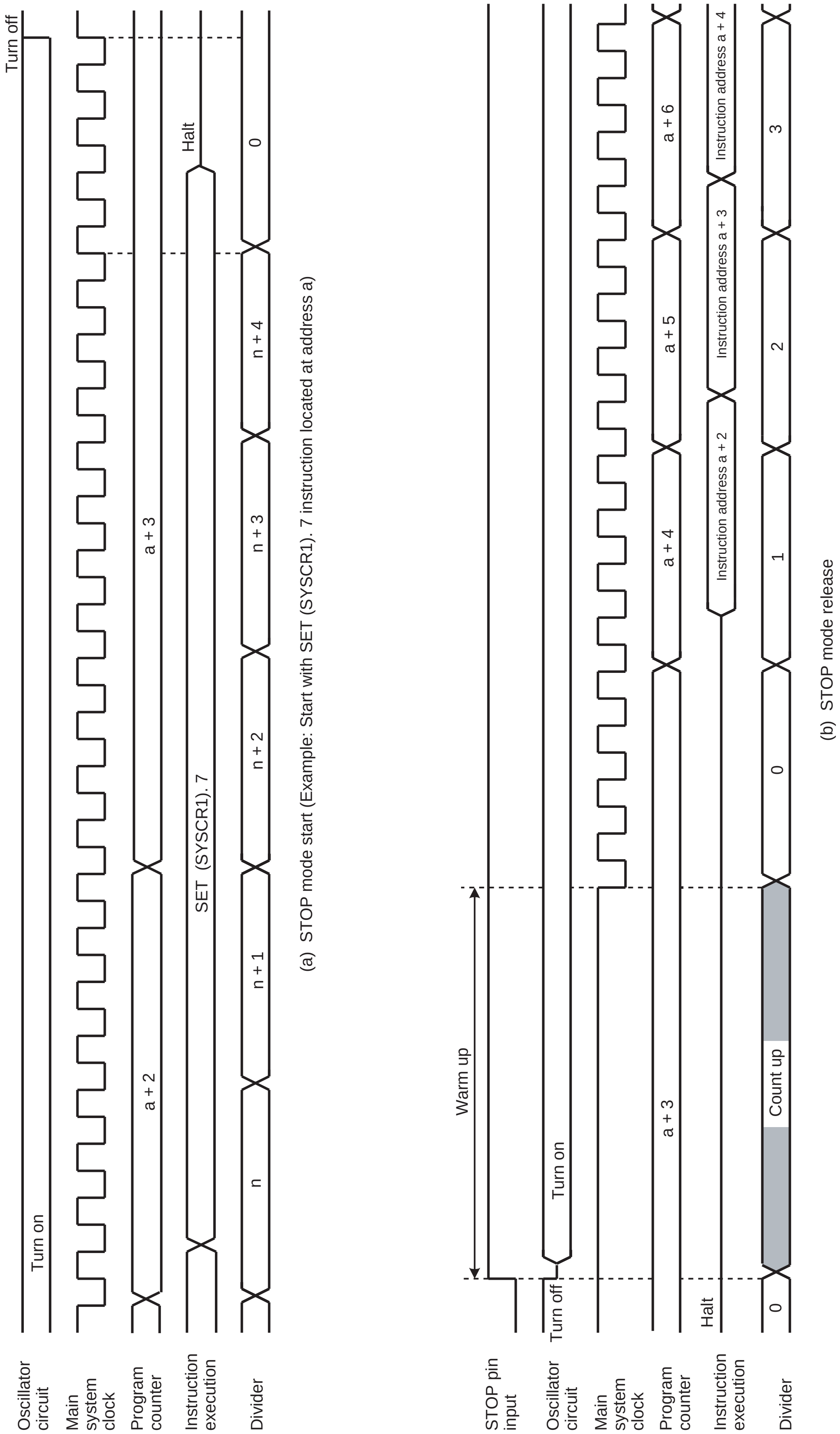


Figure 2-9 STOP Mode Start/Release

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.

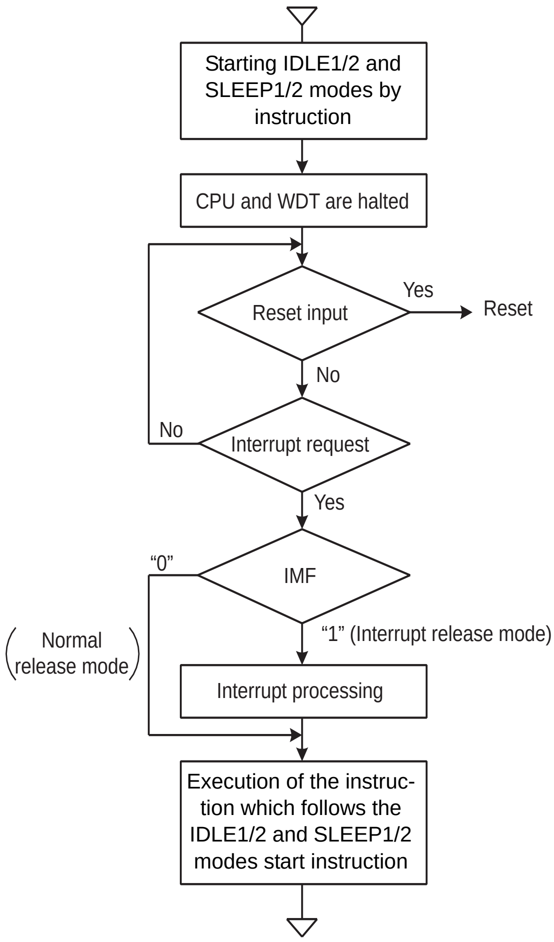


Figure 2-10 IDLE1/2 and SLEEP1/2 Modes

• 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.

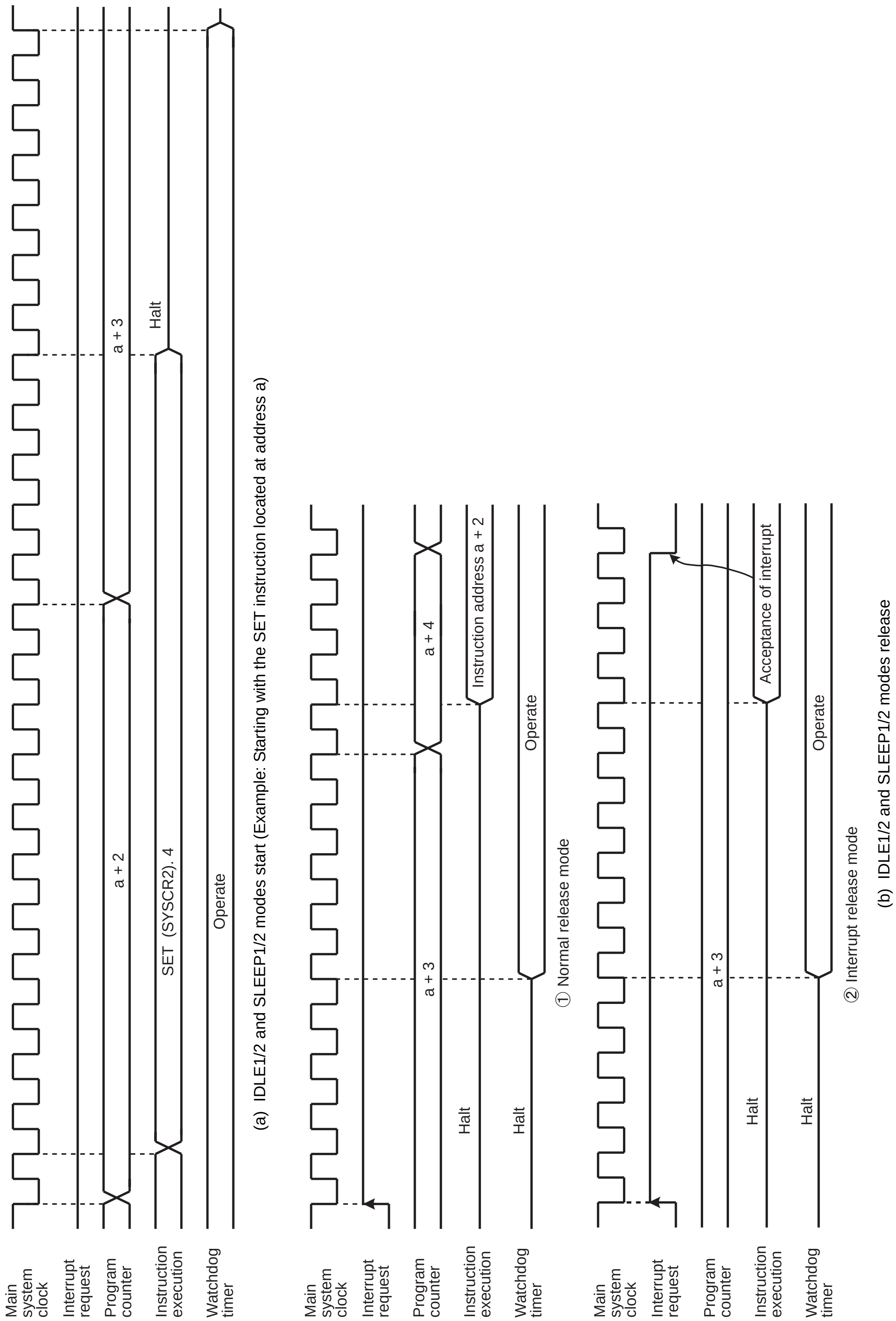


Figure 2-11 IDLE1/2 and SLEEP1/2 Modes Start/Release

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.

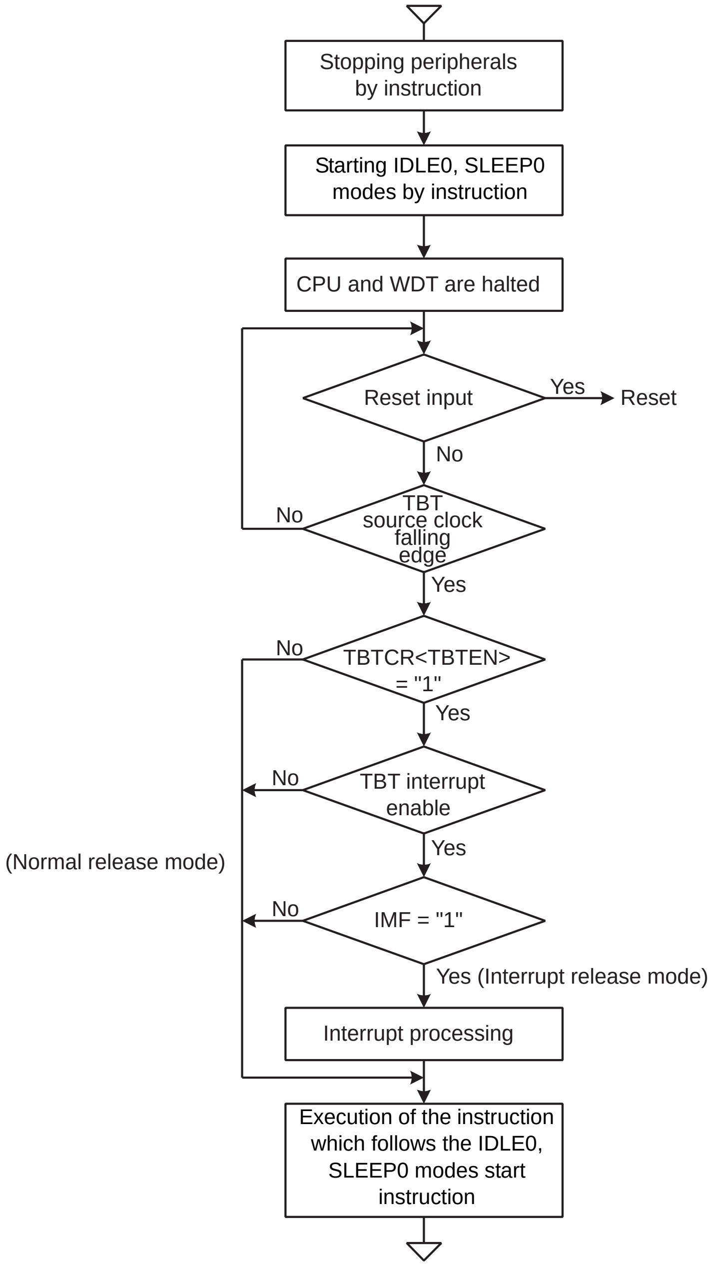


Figure 2-12 IDLE0 and SLEEP0 Modes

- 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•EF7•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•EF7•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.

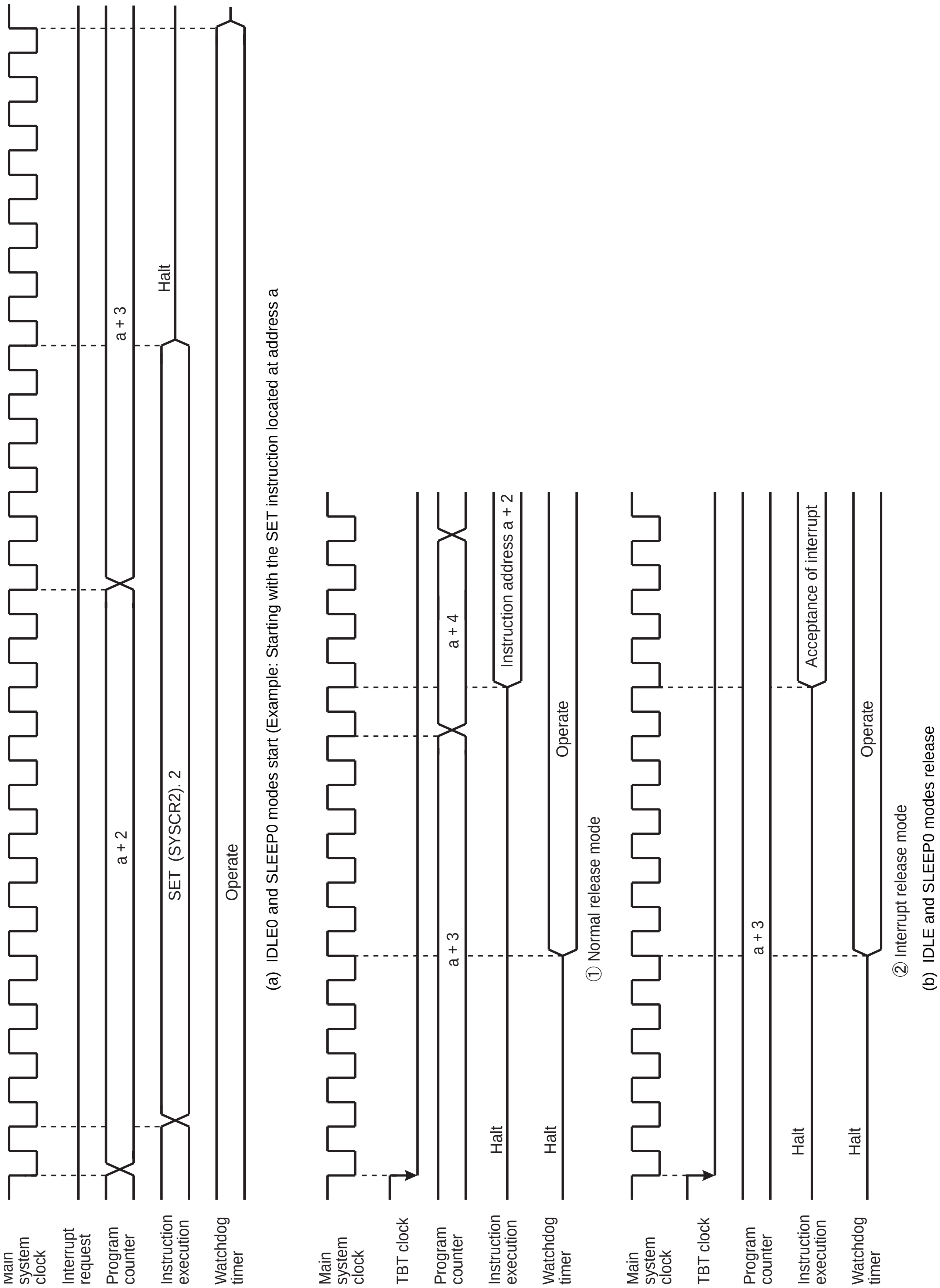


Figure 2-13 IDLE0 and SLEEP0 Modes Start/Release

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)
```

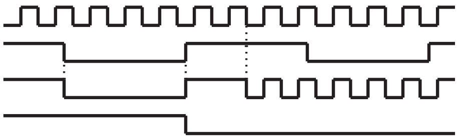
Example 2 :Switching to the SLOW1 mode after low-frequency clock has stabilized.

```
            SET     (SYSCR2). 6         ; SYSCR2<XTEN> ← 1
            LD      (TC2CR), 14H        ; Sets mode for TC2 (fs for source)
            LDW     (TC2DRL), 8000H     ; Sets warm-up time (Depend on oscillator accompanied)
            DI                          ; IMF ← 0
            SET     (EIRH). 5           ; Enables INTTC2
            EI                          ; IMF ← 1
            SET     (TC2CR). 5          ; Starts TC2
              :
PINTTC2:    CLR     (TC2CR). 5          ; Stops TC2
            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
              :
VINTTC2:    DW      PINTTC2             ; INTTC2 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 (TC2), 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      (TC2CR), 10H      ; Sets mode for TC2 (fc for source)
            LD      (TC2DRH), 0F8H    ; Sets warm-up time
            DI                        ; IMF ← 0
            SET     (EIRH). 5         ; Enables INTTC2
            EI                        ; IMF ← 1
            SET     (TC2CR). 5        ; Starts TC2
             :
PINTTC2:    CLR     (TC2CR). 5        ; Stops TC2
            CLR     (SYSCR2). 5       ; SYSCR2<SYSCK> ← 0
                                        (Switches the main system clock to the high-frequency clock)
            RETI
             :
VINTTC2:    DW      PINTTC2           ; INTTC2 vector table
```

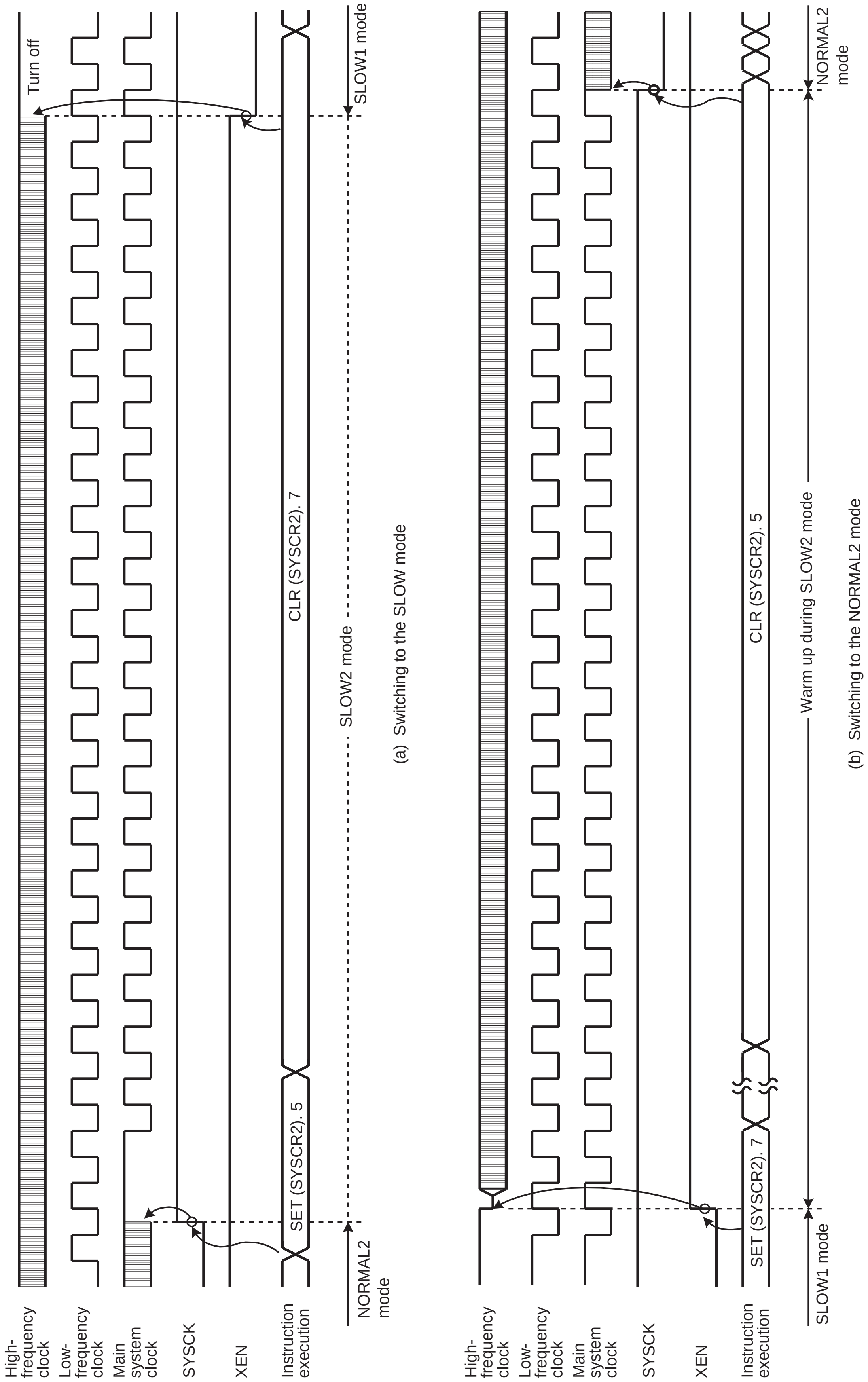


Figure 2-14 Switching between the NORMAL2 and SLOW Modes

2.3 Reset Circuit

The TMP86CK74AFG 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		
		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.

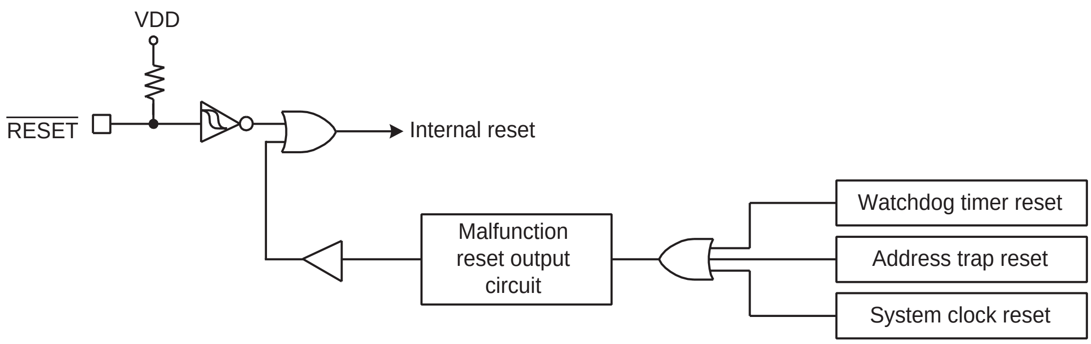
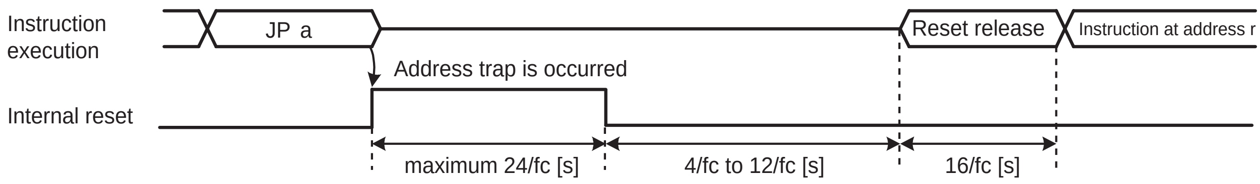


Figure 2-15 Reset Circuit

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 TMP86CK74AFG has a total of 17 interrupt sources excluding reset, of which 1 source levels are multiplexed. Interrupts can be nested with priorities. Four of the internal interrupt sources are non-maskable while the rest are maskable.

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

Interrupt Factors		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/External	(Reset)	Non-maskable	–	FFFE	1
Internal	INTSWI (Software interrupt)	Non-maskable	–	FFFC	2
Internal	INTUNDEF (Executed the undefined instruction interrupt)	Non-maskable	–	FFFC	2
Internal	INTATRAP (Address trap interrupt)	Non-maskable	IL2	FFFA	2
Internal	INTWDT (Watchdog timer interrupt)	Non-maskable	IL3	FFF8	2
External	$\overline{\text{INT0}}$	IMF• EF4 = 1, INT0EN = 1	IL4	FFF6	5
Internal	INTTC1	IMF• EF5 = 1	IL5	FFF4	6
External	INT1	IMF• EF6 = 1	IL6	FFF2	7
Internal	INTTBT	IMF• EF7 = 1	IL7	FFF0	8
Internal	INTTC3	IMF• EF8 = 1	IL8	FFEE	9
Internal	INTSIO	IMF• EF9 = 1	IL9	FFEC	10
Internal	INTTC4	IMF• EF10 = 1	IL10	FFEA	11
External	INT3	IMF• EF11 = 1	IL11	FFE8	12
External	INT4	IMF• EF12 = 1	IL12	FFE6	13
Internal	INTTC2	IMF• EF13 = 1	IL13	FFE4	14
External	$\overline{\text{INT5}}$	IMF• EF14 = 1	IL14	FFE2	15
Internal	INTADC	IMF• EF15 = 1, IL15ER = 0	IL15	FFE0	16
External	INT2	IMF• EF15 = 1, IL15ER = 1			

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

Note 2: To use the address trap interrupt (INTATRAP), clear WDTCR1<ATOUT> to “0” (It is set for the “reset request” after reset is cancelled). For details, see “Address Trap”.

Note 3: To use the watchdog timer interrupt (INTWDT), clear WDTCR1<WDTOUT> to "0" (It is set for the "Reset request" after reset is released). For details, see "Watchdog Timer".

3.1 Interrupt latches (IL15 to IL2)

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

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

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

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

Note: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)
In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

Example 1 :Clears interrupt latches

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

Example 2 :Reads interrupt 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 003AH and 003BH in SFR area, and they can be read and written by an instructions (Including read-modify-write instructions such as bit manipulation or operation instructions).

3.2.1 Interrupt master enable flag (IMF)

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

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

3.2.2 Individual interrupt enable flags (EF15 to EF4)

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

Note: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)
In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute 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)

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

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

Note 2: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)
In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

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

Interrupt Enable Registers

(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)

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

Note 1: *: Don’t care

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

Note 3: In main program, before manipulating the interrupt enable flag (EF) or the interrupt latch (IL), be sure to clear IMF to "0" (Disable interrupt by DI instruction). Then set IMF newly again as required after operating on the EF or IL (Enable interrupt by EI instruction)
In interrupt service routine, because the IMF becomes "0" automatically, clearing IMF need not execute normally on interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

3.3 Interrupt Source Selector (INTSEL)

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

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

1. INTADC and INT2 share the interrupt source level whose priority is 16.

Interrupt source selector

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

IL15ER	Selects INTADC or INT2	0: INTADC 1: INT2	R/W

3.4 Interrupt Sequence

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

3.4.1 Interrupt acceptance processing is packaged as follows.

a. The interrupt master enable flag (IMF) is cleared to “0” in order to disable the acceptance of any following interrupt.

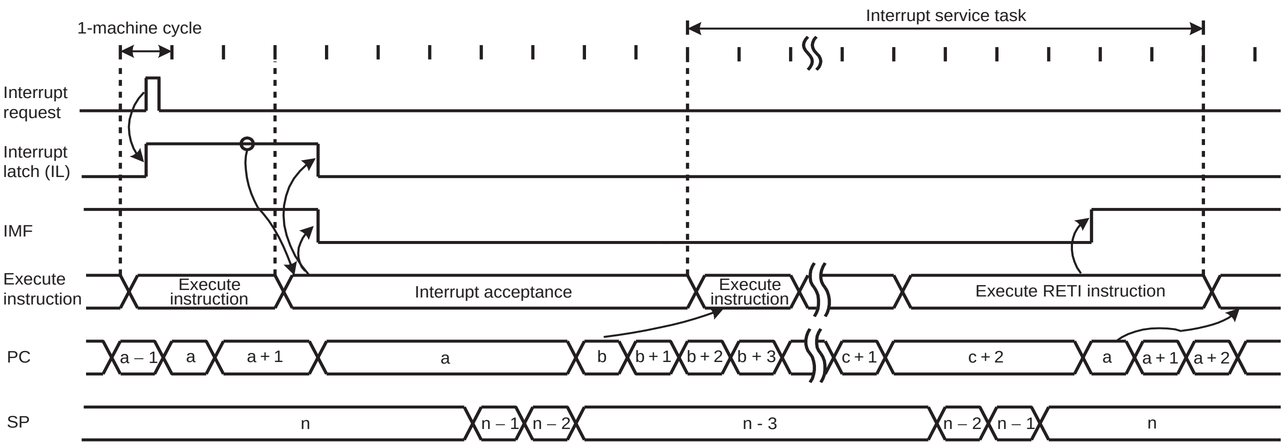
b. The interrupt latch (IL) for the interrupt source accepted is cleared to “0”.

c. The contents of the program counter (PC) and the program status word, including the interrupt master enable flag (IMF), are saved (Pushed) on the stack in sequence of PSW + IMF, PCH, PCL. Meanwhile, the stack pointer (SP) is decremented by 3.

d. The entry address (Interrupt vector) of the corresponding interrupt service program, loaded on the vector table, is transferred to the program counter.

e. The instruction stored at the entry address of the interrupt service program is executed.

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



Note 1: a: Return address entry address, b: Entry address, c: Address which RETI instruction is stored
Note 2: On condition that interrupt is enabled, it takes 38/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

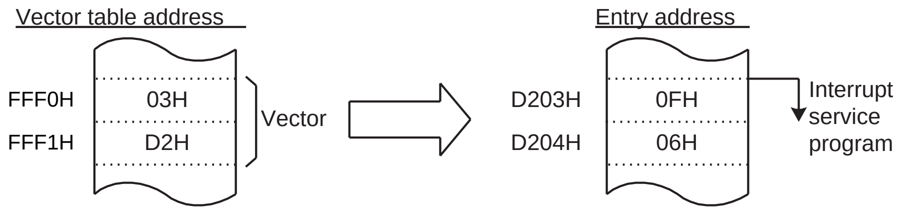


Figure 3-2 Vector table address,Entry address

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.

3.4.2 Saving/restoring general-purpose registers

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

3.4.2.1 Using PUSH and POP instructions

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

Example :Save/store register using PUSH and POP instructions

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

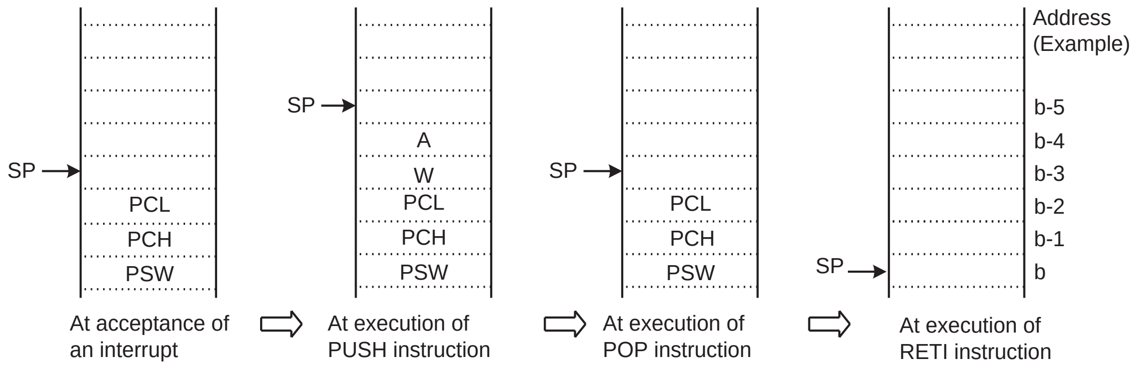


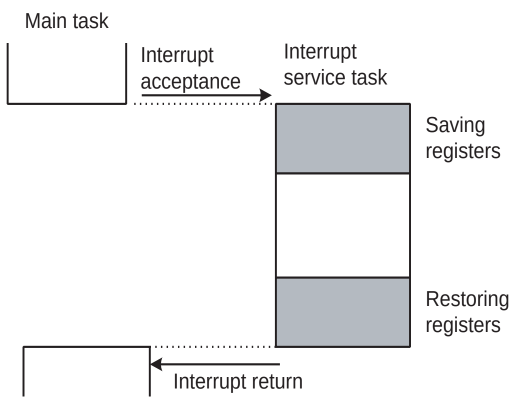
Figure 3-3 Save/store register using PUSH and POP instructions

3.4.2.2 Using data transfer instructions

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

Example :Save/store register using data transfer instructions

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



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

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

3.4.3 Interrupt return

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

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

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

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

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

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

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

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

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

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

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

3.5 Software Interrupt (INTSW)

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

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

3.5.1 Address error detection

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

3.5.2 Debugging

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

3.6 Undefined Instruction Interrupt (INTUNDEF)

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

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

3.7 Address Trap Interrupt (INTATRAP)

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

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

3.8 External Interrupts

The TMP86CK74AFG has 6 external interrupt inputs. These inputs are equipped with digital noise reject circuits (Pulse inputs of less than a certain time are eliminated as noise).

Edge selection is also possible with INT1 to INT4. The $\overline{\text{INT0}}$/P50 pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

Edge selection, noise reject control and $\overline{\text{INT0}}$/P50 pin function selection are performed by the external interrupt control register (EINTCR).

Source	Pin	Enable Conditions	Release Edge (level)	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 • EF6 = 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 • EF15 = 1 and IL15ER=1	Falling edge or Rising edge	Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT3	INT3	IMF • EF11 = 1	Falling edge or Rising edge	Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT4	INT4	IMF • EF12 = 1	Falling edge, Rising edge, Falling and Rising edge or H level	Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT5	$\overline{INT5}$	IMF • EF14 = 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	INT4ES		INT3ES	INT2ES	INT1ES		(Initial value: 0000 000*)

INT1NC	Noise reject time select	0: Pulses of less than 63/fc [s] are eliminated as noise 1: Pulses of less than 15/fc [s] are eliminated as noise	R/W
INT0EN	P50/$\overline{\text{INT0}}$ pin configuration	0: P50 input/output port 1: $\overline{\text{INT0}}$ pin (Port P50 should be set to an input mode)	R/W
INT4 ES	INT4 edge select	00: Rising edge 01: Falling edge 10: Rising edge and Falling edge 11: H level	R/W
INT3 ES	INT3 edge select	0: Rising edge 1: Falling edge	R/W
INT2 ES	INT2 edge select	0: Rising edge 1: Falling edge	R/W
INT1 ES	INT1 edge select	0: Rising edge 1: Falling edge	R/W

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

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

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

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

4. Special Function Register (SFR)

The TMP86CK74AFG 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 TMP86CK74AFG.

4.1 SFR

Address	Read	Write
0000H	P0DR	
0001H	P1DR	
0002H	P2DR	
0003H	P3DR	
0004H	P4DR	
0005H	P5DR	
0006H	P6DR	
0007H	P7DR	
0008H	P8DR	
0009H	P9DR	
000AH	P0CR	
000BH	P1OUTCR	
000CH	P4CR1	
000DH	P5CR	
000EH	ADCCR1	
000FH	ADCCR2	
0010H	TC3DRA	
0011H	TC3DRB	-
0012H	TC3CR	
0013H	TC2CR	
0014H	TC4CR	
0015H	P1PRD	-
0016H	P2PRD	-
0017H	P3PRD	-
0018H	TC4DR	
0019H	SIOCR1	
001AH	SIOCR2	
001BH	SIOSR	-
001CH	SIOBUF	
001DH	PDDR	
001EH	Reserved	
001FH	Reserved	
0020H	TC1DRAL	
0021H	TC1DRAH	
0022H	TC1DRBL	
0023H	TC1DRBH	
0024H	TC2DRL	
0025H	TC2DRH	

Address	Read	Write
0026H	ADCDR2	-
0027H	ADCDR1	-
0028H	P4CR2	
0029H	TC3SEL	
002AH	VFTCR1	
002BH	VFTCR2	
002CH	VFTCR3	
002DH	VFTSR	-
002EH	Reserved	
002FH	Reserved	
0030H	Reserved	
0031H	-	STOPCR
0032H	TC1CR	
0033H	Reserved	
0034H	-	WDTCR1
0035H	-	WDTCR2
0036H	TBTCR	
0037H	EINTCR	
0038H	SYSCR1	
0039H	SYSCR2	
003AH	EIRL	
003BH	EIRH	
003CH	ILL	
003DH	ILH	
003EH	INTSEL	
003FH	PSW	

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

Note 2: – ; Cannot be accessed.

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

4.2 DBR

Address	Read	Write
0F80H	VFTDBR(T0,V7 to V0)	
0F81H	VFTDBR(T1,V7 to V0)	
0F82H	VFTDBR(T2,V7 to V0)	
0F83H	VFTDBR(T3,V7 to V0)	
0F84H	VFTDBR(T4,V7 to V0)	
0F85H	VFTDBR(T5,V7 to V0)	
0F86H	VFTDBR(T6,V7 to V0)	
0F87H	VFTDBR(T7,V7 to V0)	
0F88H	VFTDBR(T8,V7 to V0)	
0F89H	VFTDBR(T9,V7 to V0)	
0F8AH	VFTDBR(T10,V7 to V0)	
0F8BH	VFTDBR(T11,V7 to V0)	
0F8CH	VFTDBR(T12,V7 to V0)	
0F8DH	VFTDBR(T13,V7 to V0)	
0F8EH	VFTDBR(T14,V7 to V0)	
0F8FH	VFTDBR(T15,V7 to V0)	
0F90H	VFTDBR(T0,V15 to V8)	
0F91H	VFTDBR(T1,V15 to V8)	
0F92H	VFTDBR(T2,V15 to V8)	
0F93H	VFTDBR(T3,V15 to V8)	
0F94H	VFTDBR(T4,V15 to V8)	
0F95H	VFTDBR(T5,V15 to V8)	
0F96H	VFTDBR(T6,V15 to V8)	
0F97H	VFTDBR(T7,V15 to V8)	
0F98H	VFTDBR(T8,V15 to V8)	
0F99H	VFTDBR(T9,V15 to V8)	
0F9AH	VFTDBR(T10,V15 to V8)	
0F9BH	VFTDBR(T11,V15 to V8)	
0F9CH	VFTDBR(T12,V15 to V8)	
0F9DH	VFTDBR(T13,V15 to V8)	
0F9EH	VFTDBR(T14,V15 to V8)	
0F9FH	VFTDBR(T15,V15 to V8)	

Address	Read	Write
0FA0H	VFTDBR(T0,V23 to V16)	
0FA1H	VFTDBR(T1,V23 to V16)	
0FA2H	VFTDBR(T2,V23 to V16)	
0FA3H	VFTDBR(T3,V23 to V16)	
0FA4H	VFTDBR(T4,V23 to V16)	
0FA5H	VFTDBR(T5,V23 to V16)	
0FA6H	VFTDBR(T6,V23 to V16)	
0FA7H	VFTDBR(T7,V23 to V16)	
0FA8H	VFTDBR(T8,V23 to V16)	
0FA9H	VFTDBR(T9,V23 to V16)	
0FAAH	VFTDBR(T10,V23 to V16)	
0FABH	VFTDBR(T11,V23 to V16)	
0FACH	VFTDBR(T12,V23 to V16)	
0FADH	VFTDBR(T13,V23 to V16)	
0FAEH	VFTDBR(T14,V23 to V16)	
0FAFH	VFTDBR(T15,V23 to V16)	
0FB0H	VFTDBR(T0,V31 to V24)	
0FB1H	VFTDBR(T1,V31 to V24)	
0FB2H	VFTDBR(T2,V31 to V24)	
0FB3H	VFTDBR(T3,V31 to V24)	
0FB4H	VFTDBR(T4,V31 to V24)	
0FB5H	VFTDBR(T5,V31 to V24)	
0FB6H	VFTDBR(T6,V31 to V24)	
0FB7H	VFTDBR(T7,V31 to V24)	
0FB8H	VFTDBR(T8,V31 to V24)	
0FB9H	VFTDBR(T9,V31 to V24)	
0FBAH	VFTDBR(T10,V31 to V24)	
0FBBH	VFTDBR(T11,V31 to V24)	
0FBCH	VFTDBR(T12,V31 to V24)	
0FBDH	VFTDBR(T13,V31 to V24)	
0FBEH	VFTDBR(T14,V31 to V24)	
0FBFH	VFTDBR(T15,V31 to V24)	

Address	Read	Write
0FC0H	VFTDBR(T0,V36 to V32)	
0FC1H	VFTDBR(T1,V36 to V32)	
0FC2H	VFTDBR(T2,V36 to V32)	
0FC3H	VFTDBR(T3,V36 to V32)	
0FC4H	VFTDBR(T4,V36 to V32)	
0FC5H	VFTDBR(T5,V36 to V32)	
0FC6H	VFTDBR(T6,V36 to V32)	
0FC7H	VFTDBR(T7,V36 to V32)	
0FC8H	VFTDBR(T8,V36 to V32)	
0FC9H	VFTDBR(T9,V36 to V32)	
0FCAH	VFTDBR(T10,V36 to V32)	
0FCBH	VFTDBR(T11,V36 to V32)	
0FCCH	VFTDBR(T12,V36 to V32)	
0FCDH	VFTDBR(T13,V36 to V32)	
0FCEH	VFTDBR(T14,V36 to V32)	
0FCFH	VFTDBR(T15,V36 to V32)	
0FD0H	Reserved	
0FD1H	Reserved	
0FD2H	Reserved	
0FD3H	Reserved	
0FD4H	Reserved	
0FD5H	Reserved	
0FD6H	Reserved	
0FD7H	Reserved	
0FD8H	Reserved	
0FD9H	Reserved	
0FDAH	Reserved	
0FDBH	Reserved	
0FDCH	Reserved	
0FDDH	Reserved	
0FDEH	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.).

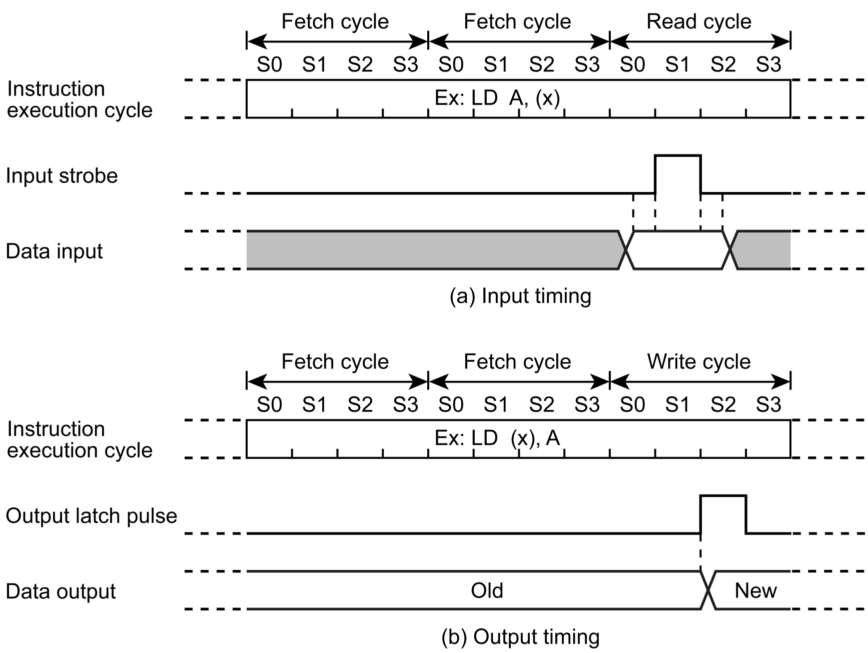
5. I/O Ports

The TMP86CK74AFG has 11 parallel input/output ports (70 pins) as follows.

	Primary Function	Secondary Functions
Port P0	8-bit I/O port	–
Port P1	8-bit I/O port	External interrupt input, timer/counter input/output, Serial interface input/output
Port P2	3-bit I/O port	Low-frequency resonator connections, external interrupt input/output, STOP mode release signal Input
Port P3	2-bit I/O port	–
Port P4	8-bit I/O port	Analog input, STOP mode release signal input
Port P5	4-bit I/O port	External interrupt input
Port P6	8-bit I/O port	VFT output
Port P7	8-bit I/O port	VFT output
Port P8	8-bit I/O port	VFT output
Port P9	8-bit I/O port	VFT output
Port PD	5-bit I/O port	VFT 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.



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

Figure 5-1 Input/Output Timing (Example)

5.1 Port P0 (P07 to P00)

Port P0 is an 8-bit general-purpose input/output port which can be configured as an input or an output in one-bit unit. Each bit of the port can be configured for either input or output separately, using the P0 port input/output control register (P0CR). A reset clears the P0CR to "0", placing port P0 in input mode. A reset also initializes the P0 port output latch (P0DR) to "0".

Note: If the port is in input mode, it senses the state of an input to its pins. If some pins of the port are in input mode, and others are in output mode, the content of the output latch related to a port pin that is in input mode may be changed when a bit manipulation instruction is executed on the port.

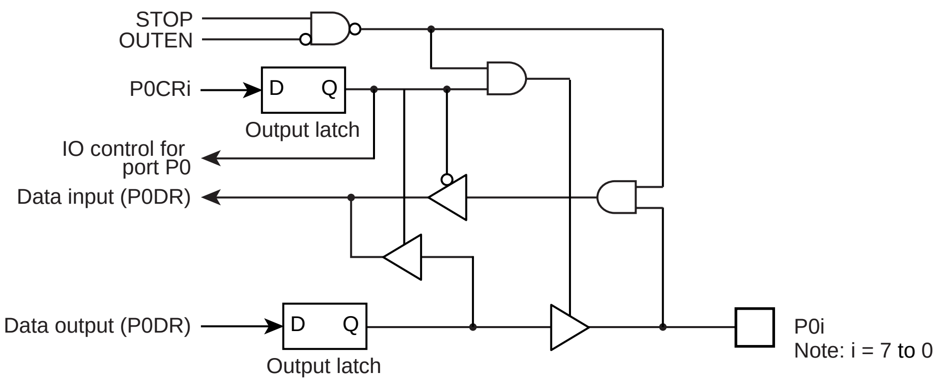


Figure 5-2 Port P0

	7	6	5	4	3	2	1	0	
P0DR (0000H) R/W	P07	P06	P05	P04	P03	P02	P01	P00	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P0CR (000AH)									(Initial value: 0000 0000)

P0CR	I/O control for port P0 (This register can be set on bit basis.)	0: Input mode 1: Output mode	R/W

5.2 Port P1 (P17 to P10)

Port P1 is an 8-bit input/output port, and also used as a timer counter input/output, external interrupt input, and serial interface input/output. To use port P1 as an input port or secondary-function pins, set its output latch (P1DR) to “1”. A reset sets the output latch to “1” and clears the push-pull control register (P1OUTCR) to “0”.

The P1OUTCR can be used to select Nch open-drain output or CMOS output for the output circuit of port P1. To use port P1 as an input port, set the P1DR to “1”, and then clear the corresponding bit of the P1OUTCR to “0”.

Port P1 has separate data input registers. To sense the state of the output latch, read the P1DR. To sense the state of the pins the port, read the P1 port input data (P1PRD) register.

The input waveform of a TC3 input can be inverted in terms of phase, using the Timer Counter3 input control (TC3SEL) register.

P10, P11, P12, P13, and P14 can work not only as a port but also as, respectively, the TC2, TC3/INT3, $\overline{\text{PWM4}}$/$\overline{\text{PDO4}}$/TC4, $\overline{\text{PPG}}$, and INT4 functions. To use the TC2, TC3, INT3, TC4, and INT4 functions, place the respective pins in input mode. To use the $\overline{\text{PWM4}}$, $\overline{\text{PDO4}}$, and $\overline{\text{PPG}}$ functions, place the respective pins in output mode.

P15, P16, and P17 can work not only as a port but also as, respectively, the SI, SO, and $\overline{\text{SCK}}$ functions. To use these functions, place the pin corresponding to the SI function in input mode, the pin corresponding to the SO function in output mode, and the pin corresponding to the $\overline{\text{SCK}}$ function in either input or output mode.

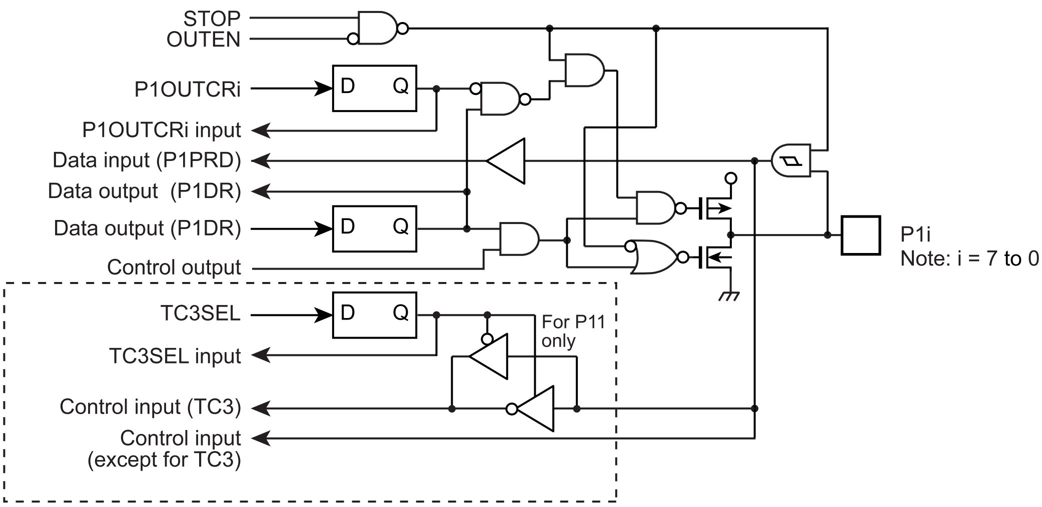


Figure 5-3 Port P1

P1DR (0001H) R/W

7	6	5	4	3	2	1	0
P17 $\overline{\text{SCK}}$	P16 SO	P15 SI	P14 INT4	P13 $\overline{\text{PPG}}$	P12 $\overline{\text{PWM4}}$ $\overline{\text{PDO4}}$ TC4	P11 TC3 INT3	P10 TC2

(Initial value: 1111 1111)

P1OUTCR (000BH)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

P1OUTCR	I/O control for port P1 (This register can be set on bit basis.)	0: Nch open-drain output 1: CMOS output	R/W

P1PRD (0015H) Read only

7	6	5	4	3	2	1	0
P17	P16	P15	P14	P13	P12	P11	P10

TC3SEL (0029H)

7	6	5	4	3	2	1	0
							TC3INV

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

TC3INV	TC3 input control	0: Normal input 1: Inverted input	R/W

P1OUTCR	P1DR	Function
0	0	Low output
0	1	Input, open-drain output, or control input
1	0	Low output
1	1	High output or control output

5.3 Port P2 (P22 to P20)

Port P2 is a 3-bit input/output port. It can work not only as a port but also as external input, STOP mode release signal input, and low-frequency resonator connection pins. To use it as an input port or the secondary-function pins, set the output latch (P2DR) to “1”. A reset initializes the P2DR to “1”. To run the device in dual clock mode, connect a low-frequency resonator (32.768 kHz) to pins P21 (XTIN) and P22 (XTOUT). When the device runs in single clock mode, P21 and P22 can be used as an ordinary input/output port. It is recommended that pin P20 be used for external interrupt input, STOP release signal input, or as an input port (if it is used as an output port, it is set with the content of the interrupt latch at the negative-going edge of the signal.)

Port P2 has separate data input registers. To sense the state of the output latch, read the P2DR. To sense the state of the pins of the port, read the P2 port input data (P2PRD) register.

If a read instruction is executed for the P2DR or P2PRD on port P2, the sensed state of bits 7 to 3 is undefined.

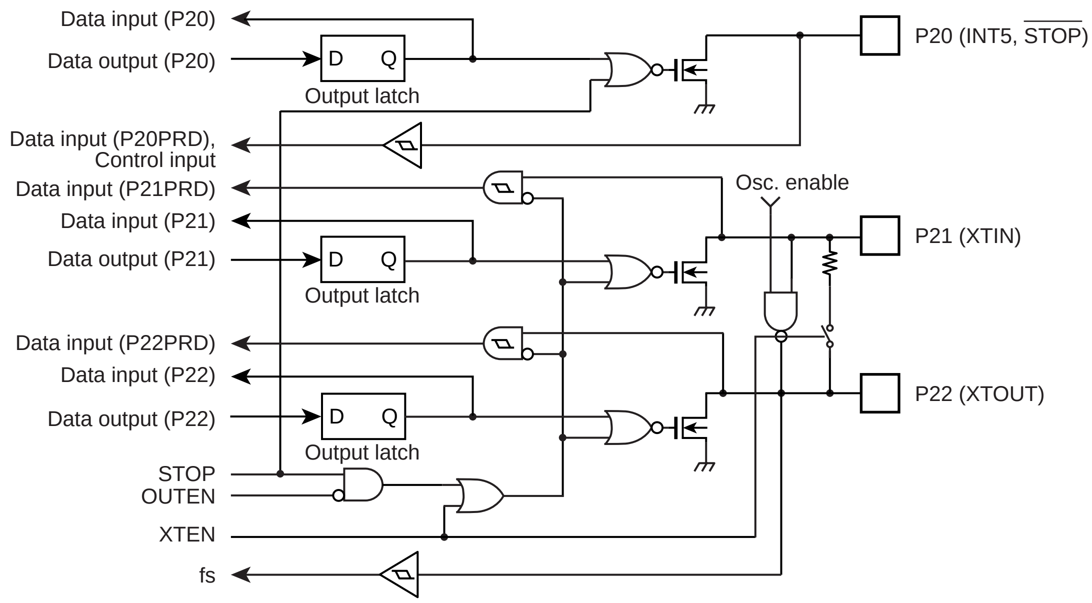


Figure 5-4 Port P2

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

	7	6	5	4	3	2	1	0
P2PRD (0016H) Read only						P22	P21	P20

Note: Because pin P20 is used also as the $\overline{STOP}$ pin, its output high impedance becomes high when it enters the STOP mode regardless of the state of OUTEN.

5.4 Port P3 (P31 to P30)

Port P3 is a 2-bit input/output port. To use it as an input port, set the output latch (P3DR) to “1”. A reset initializes the output latch to “1”.

Port P3 has separate data input registers. To sense the state of the output latch, read the P3DR. To sense the state of the pins of the port, read the P3PRD register.

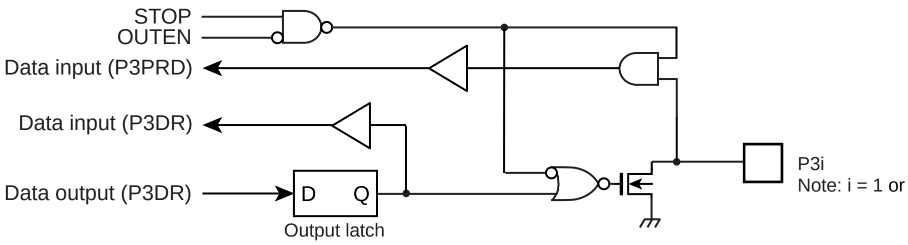


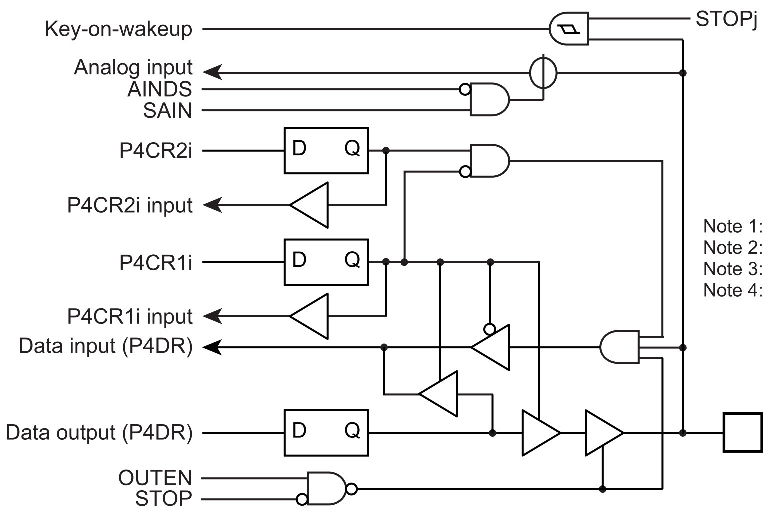
Figure 5-5 Port P5

	7	6	5	4	3	2	1	0	
P3DR (0003H) R/W							P31	P30	(Initial value: **** **11)

	7	6	5	4	3	2	1	0
P3PRD (0017H) Read only							P31	P30

5.5 Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port. Each bit of the port can be configured for either input or output separately, using the P4 port input/output control register (P4CR1). These pins can work not only as a port but also for analog input and key-on-wakeup input. To use each bit for output, set the corresponding bit of the P4CR1 to “1” to place them in output mode. To use them in input mode, clear the corresponding bit of the P4CR1 to “0”, then set the P4CR2 to “1”. To use the bits for analog input and key-on-wakeup input, clear the P4CR1 and P4CR2 to “0” in the stated order (then, for analog input, clear the ADCCR1<AINDS> to “0”, and start the AD). A reset initializes the P4CR1 and P4CR2, respectively, to “0” and “1”, thereby placing port P4 in input mode. A reset also clears the P4 port output latch (P4DR) to “0”.



Note 1: i = 7 to 0, and j = 5 to 2
Note 2: STOP is SYSCR1 bit 7.
Note 3: SAIN is AD input seletion signal.
Note 4: STOPj is an input selection signal for the key-on-wakeup function.

Figure 5-6 Port P4

P4DR (0004H) R/W

7	6	5	4	3	2	1	0
P47 AIN7 STOP5	P46 AIN6 STOP4	P45 AIN5 STOP3	P44 AIN4 STOP2	P43 AIN3	P42 AIN2	P41 AIN1	P40 AIN0

(Initial value: 0000 0000)

P4CR1 (000CH)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

P4CR1	I/O control for port P4 (This register can be set on bit basis.)	0: Input mode or analog input/key-on-wakeup input 1: Output mode	R/W

P4CR2 (0028H)

7	6	5	4	3	2	1	0

(Initial value: 1111 1111)

P4CR2	I/O control for port P4 (This register can be set on bit basis.)	0: Analog input/key-on-wakeup input 1: Input mode	R/W

Note 1: If a port is in input mode, it senses the state of an input to its pins. If some pins of the port are in input mode, and others are in output mode, the content of the output latch related to a port pin that is in input mode may be changed when a bit manipulation instruction is executed on the port.

Note 2: The P4CR2 controls the input gate of pins used for analog input. In analog input mode, clear the P4CR2 to “0” to fix the input gate, thereby protecting it from through current. In input mode, set the P4CR2 to “1”. When using the key-on-wakeup function, clear the P4CR2 to “0”, because the inputs are received separately. If the P4CR2 is “0”, read-accessing the P4CR2 yields “0”.

5.6 Port P5 (P53 to P50)

Port P5 is a 4-bit general-purpose input/output port. Each bit of the port can be configured for either input or output separately, using the P5 port input/output control register (P5CR). A reset clears the P5CR to “0”, placing port P5 in input mode. A reset also initializes the P5 port output latch (P5DR) to “0”.

P50, P51, and P52 can work not only as an input/output port but also, respectively, for the $\overline{\text{INT0}}$, INT1, INT2 and TC1 functions. To use these functions, place the corresponding pins in input mode.

P53 can work not only as a port but also as, respectively, the $\overline{\text{DVO}}$ function. To use the $\overline{\text{DVO}}$ function, place the respective pin in output mode.

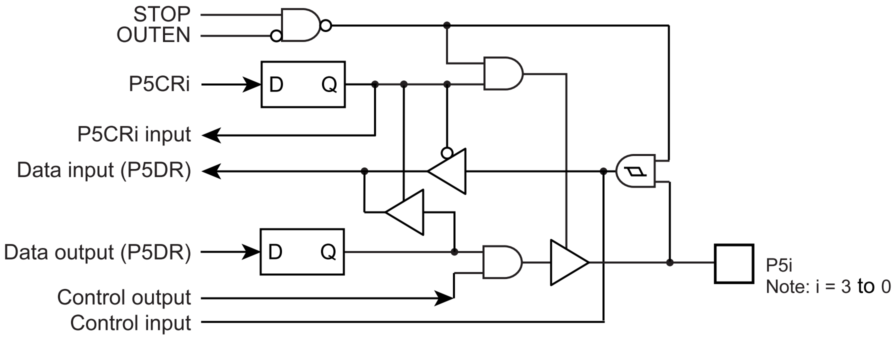


Figure 5-7 Port P5

	7	6	5	4	3	2	1	0	
P5DR (0005H)					P53 $\overline{\text{DVO}}$	P52 INT2 TC1	P51 INT1	P50 $\overline{\text{INT0}}$	(Initial value: **** 0000)

	7	6	5	4	3	2	1	0	
P5CR (000DH)									(Initial value: **** 0000)

P5CR	I/O control for port P5 (This register can be set on bit basis.)	0: Input mode 1: Output mode	R/W

Note: If a port is in input mode, it senses the state of an input to its pins. If some pins of the port are in input mode, and others are in output mode, the content of the output latch related to a port pin that is in input mode may be changed when a bit manipulation instruction is executed on the port.

5.7 Ports P6 (P67 to P60), P7 (P77 to P70), P8 (P87 to P80), and P9 (P97 to P90)

Ports P6, P7, P8, and P9 are 8-bit high-breakdown voltage input/output ports. They can work not only as a port but also for VFT driver output. They can drive directly a vacuum fluorescent tube (VFT). To use them as an input port or VFT driver, clear the output latch to “0”.

Pins not set up for VFT driver output can be used as an input/output port. To use a pin for ordinary input/output when a VFT driver is used, clear the VFT driver output data buffer memory (DBR) for the pin to “0”. A reset initializes the output latch to “0”.

It is recommended that ports P6, P7, P8, and P9 be used to drive a VFT because they have a built-in pull-down resistor.

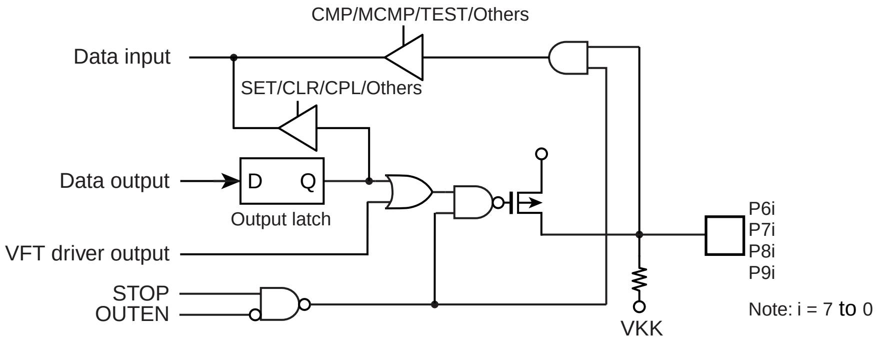


Figure 5-8 Port P6, P7, P8, P9

	7	6	5	4	3	2	1	0	
P6DR (0006H) R/W	P67	P66	P65	P64	P63	P62	P61	P60	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P7DR (0007H) R/W	P77	P76	P75	P74	P73	P72	P71	P70	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P8DR (0008H) R/W	P87	P86	P85	P84	P83	P82	P81	P80	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P9DR (0009H) R/W	P97	P96	P95	P94	P93	P92	P91	P90	(Initial value: 0000 0000)

5.8 Port PD (PD4 to PD0)

Port PD is a high-breakdown voltage input/output port. It can work not only as a port but also for VFT driver output. It can drive directly a VFT. Each bit of the port can be configured for a segment or input/output separately, using the VFTCR1<VSEL> of VFT driver control register 1 (VFTCR1). A reset clears the VSEL to “0”, causing the port to work as an input/output port. To use it as an input port, clear the output latch to “0”. A reset initializes the output latch to “0”.

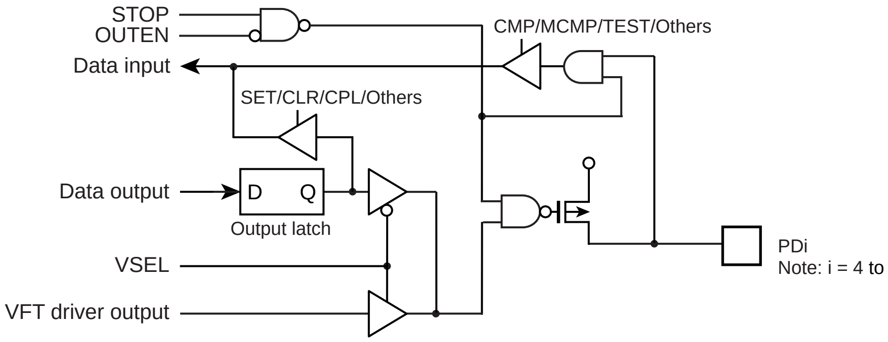


Figure 5-9 Port PD

PDDR (001DH) R/W	7	6	5	4	3	2	1	0	
				PD4	PD3	PD2	PD1	PD0	(Initial value: ***0 0000)

6. Watchdog Timer (WDT)

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

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

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

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

6.1 Watchdog Timer Configuration

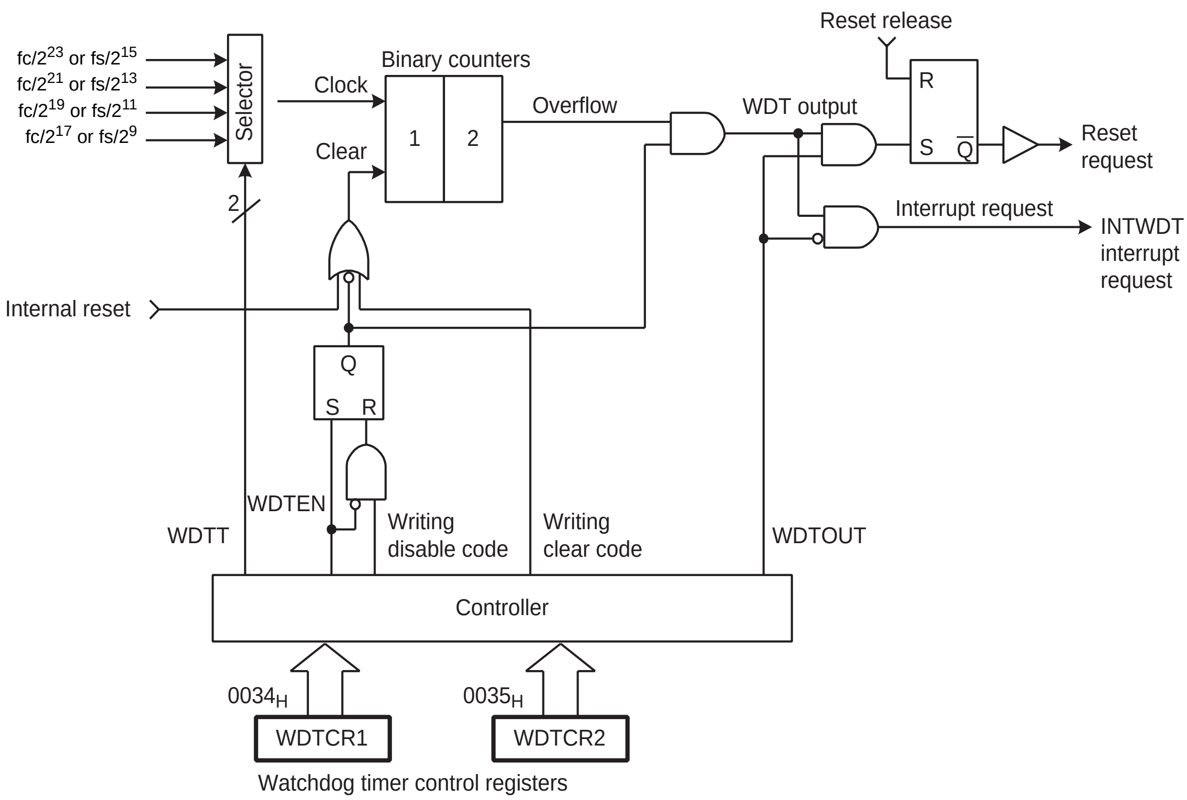


Figure 6-1 Watchdog Timer Configuration

6.2 Watchdog Timer Control

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

6.2.1 Malfunction Detection Methods Using the Watchdog Timer

The CPU malfunction is detected, as shown below.

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

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

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

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

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

```
                         LD    (WDTCR2), 4EH          : Clears the binary counters.
                         LD    (WDTCR1), 00001101B    : WDTT ← 10, WDTOUT ← 1
                       ┌ LD    (WDTCR2), 4EH          : Clears the binary counters (always clears immediately before and
                       │                                after changing WDTT).
Within 3/4 of WDT      │  :
detection time         │  :
                       ├ LD    (WDTCR2), 4EH          : Clears the binary counters.
Within 3/4 of WDT      │  :
detection time         │  :
                       └ LD    (WDTCR2), 4EH          : Clears the binary counters.
```

Watchdog Timer Control Register 1

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

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

Note 1: After clearing WDTOUT to “0”, the program cannot set it to “1”.

Note 2: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don’t care

Note 3: WDTCR1 is a write-only register and must not be used with any of read-modify-write instructions. If WDTCR1 is read, a don’t care is read.

Note 4: To activate the STOP mode, disable the watchdog timer or clear the counter immediately before entering the STOP mode. After clearing the counter, clear the counter again immediately after the STOP mode is inactivated.

Note 5: To clear WDTEN, set the register in accordance with the procedures shown in “1.2.3 Watchdog Timer Disable”.

Watchdog Timer Control Register 2

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

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

Note 1: The disable code is valid only when WDTCR1<WDTEN> = 0.

Note 2: *: Don’t care

Note 3: The binary counter of the watchdog timer must not be cleared by the interrupt task.

Note 4: Write the clear code 4EH using a cycle shorter than 3/4 of the time set in WDTCR1<WDTT>.

6.2.2 Watchdog Timer Enable

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

6.2.3 Watchdog Timer Disable

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

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

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

Example :Disabling the watchdog timer

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

Table 6-1 Watchdog Timer Detection Time (Example: fc = 16.0 MHz, fs = 32.768 kHz)

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

6.2.4 Watchdog Timer Interrupt (INTWDT)

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

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

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

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

Example :Setting watchdog timer interrupt

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

6.2.5 Watchdog Timer Reset

When a binary-counter overflow occurs while WDTCR1<WDTOUT> is set to “1”, a watchdog timer reset request is generated. When a watchdog timer reset request is generated, the internal hardware is reset. The reset time is maximum 24/fc [s] (1.5 µs @ fc = 16.0 MHz).

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

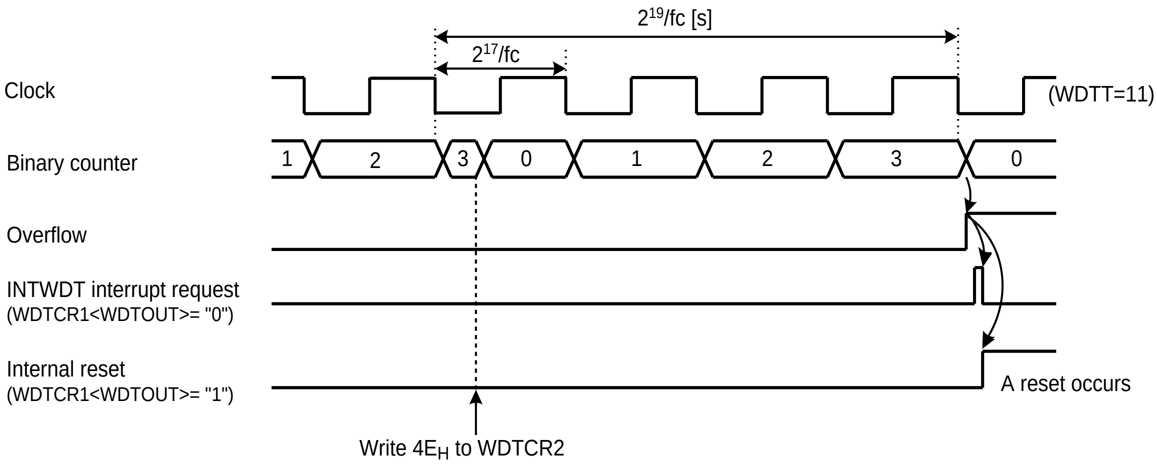


Figure 6-2 Watchdog Timer Interrupt

6.3 Address Trap

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

Watchdog Timer Control Register 1

WDTCR1 (0034H)

7	6	5	4	3	2	1	0
		ATAS	ATOUT	(WDTEN)	(WDTT)		(WDTOUT)

(Initial value: **11 1001)

ATAS	Select address trap generation in the internal RAM area	0: Generate no address trap 1: Generate address traps (After setting ATAS to “1”, writing the control code D2H to WDTCR2 is reguired)	Write only
ATOUT	Select opertion at address trap	0: Interrupt request 1: Reset request	

Watchdog Timer Control Register 2

WDTCR2 (0035H)

7	6	5	4	3	2	1	0

(Initial value: **** ****)

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

6.3.1 Selection of Address Trap in Internal RAM (ATAS)

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

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

6.3.2 Selection of Operation at Address Trap (ATOUT)

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

6.3.3 Address Trap Interrupt (INTATRAP)

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

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

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

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

6.3.4 Address Trap Reset

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

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

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

7. Time Base Timer (TBT)

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

7.1 Time Base Timer

7.1.1 Configuration

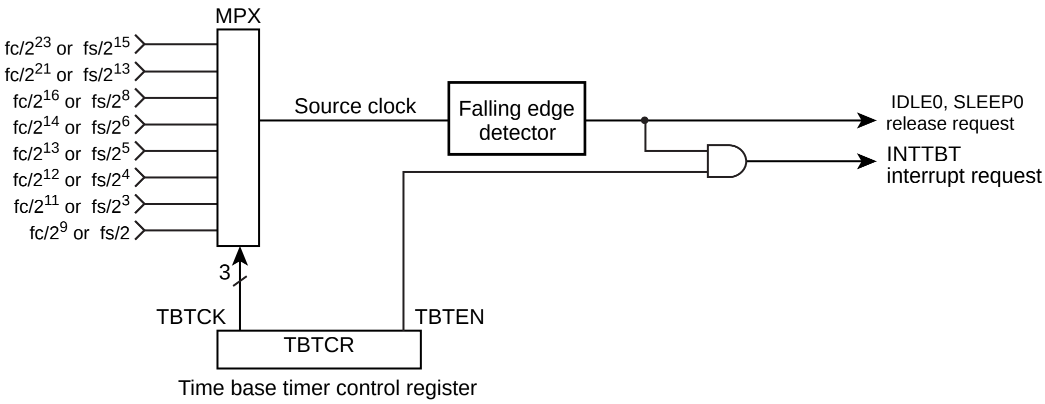


Figure 7-1 Time Base Timer configuration

7.1.2 Control

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

Time Base Timer Control Register

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

TBTEN	Time Base Timer enable / disable	0: Disable 1: Enable				
TBTCK	Time Base Timer interrupt Frequency select : [Hz]		NORMAL1/2, IDLE1/2 Mode		SLOW1/2 SLEEP1/2 Mode	R/W
			DV7CK = 0	DV7CK = 1		
		000	$fc/2^{23}$	$fs/2^{15}$	$fs/2^{15}$	
		001	$fc/2^{21}$	$fs/2^{13}$	$fs/2^{13}$	
		010	$fc/2^{16}$	$fs/2^{8}$	–	
		011	$fc/2^{14}$	$fs/2^{6}$	–	
		100	$fc/2^{13}$	$fs/2^{5}$	–	
		101	$fc/2^{12}$	$fs/2^{4}$	–	
		110	$fc/2^{11}$	$fs/2^{3}$	–	
		111	$fc/2^{9}$	fs/2	–	

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

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

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

```
LD      (TBTCR) ,  00000010B      ; TBTCK ← 010
LD      (TBTCR) ,  00001010B      ; TBTEN ← 1
DI                                ; IMF ← 0
SET     (EIRL) . 7
```

Table 7-1 Time Base Timer Interrupt Frequency (Example : fc = 16.0 MHz, fs = 32.768 kHz)

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

7.1.3 Function

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

The divider is not cleared by the program; therefore, only the first interrupt may be generated ahead of the set interrupt period (Figure 7-2).

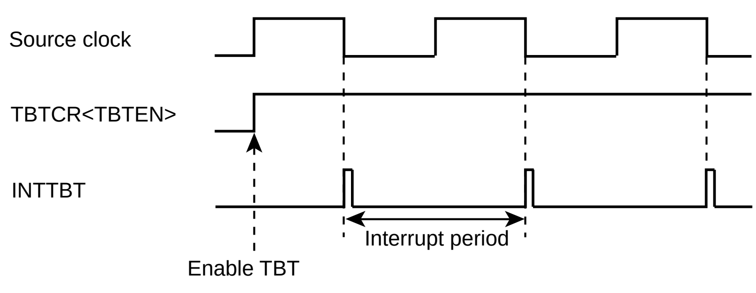
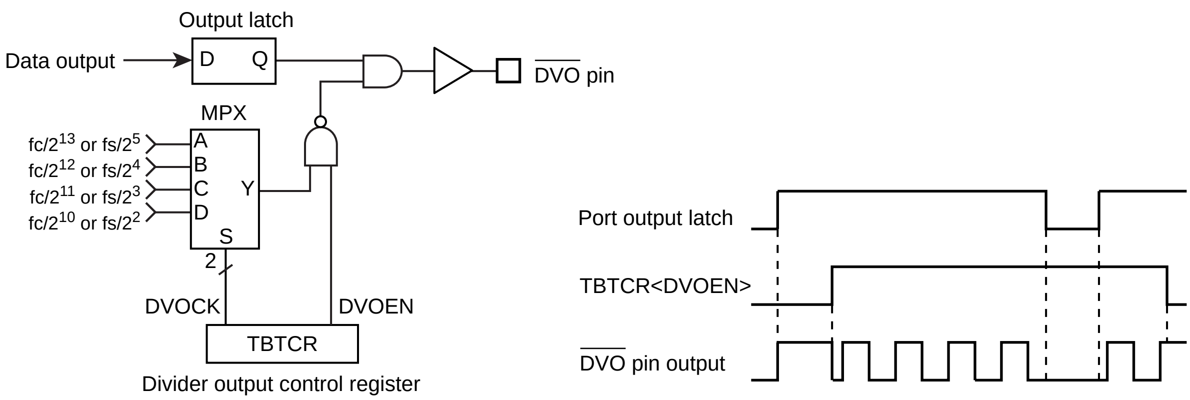


Figure 7-2 Time Base Timer Interrupt

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

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.

7.2.1 Configuration



(a) configuration (b) Timing chart

Figure 7-3 Divider Output

7.2.2 Control

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

Time Base Timer Control Register

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

DVOEN	Divider output enable / disable	0: Disable 1: Enable				R/W
DVOCK	Divider Output ($\overline{DVO}$) frequency selection: [Hz]		NORMAL1/2, IDLE1/2 Mode DV7CK = 0	NORMAL1/2, IDLE1/2 Mode DV7CK = 1	SLOW1/2 SLEEP1/2 Mode	R/W
		00	$fc/2^{13}$	$fs/2^5$	$fs/2^5$	
		01	$fc/2^{12}$	$fs/2^4$	$fs/2^4$	
		10	$fc/2^{11}$	$fs/2^3$	$fs/2^3$	
		11	$fc/2^{10}$	$fs/2^2$	$fs/2^2$	

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

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

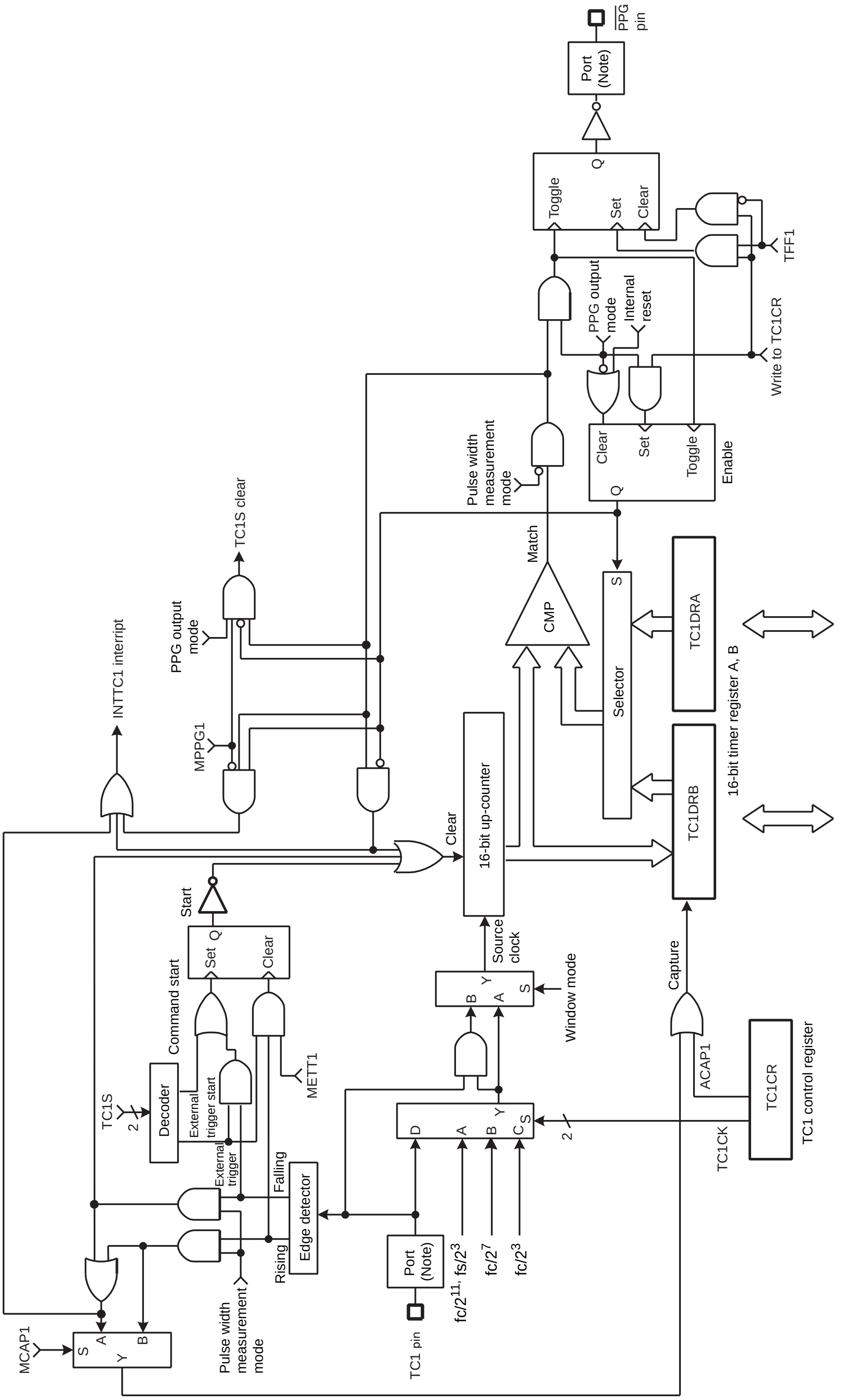
```
LD      (TBTCR) ,  00000000B        ; DVOCK ← "00"
LD      (TBTCR) ,  10000000B        ; DVOEN ← "1"
```

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

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

8. 16-Bit TimerCounter 1 (TC1)

8.1 Configuration



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

Figure 8-1 TimerCounter 1 (TC1)

8.2 TimerCounter Control

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

Timer Register

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC1DRA (0021H, 0020H)	TC1DRAH (0021H)								TC1DRAL (0020H)							
	(Initial value: 1111 1111 1111 1111)								Read/Write							
TC1DRB (0023H, 0022H)	TC1DRBH (0023H)								TC1DRBL (0022H)							
	(Initial value: 1111 1111 1111 1111)								Read/Write (Write enabled only in the PPG output mode)							

TimerCounter 1 Control Register

	7	6	5	4	3	2	1	0	
TC1CR (0032H)	TFF1	ACAP1 MCAP1 METT1 MPPG1	TC1S		TC1CK		TC1M		Read/Write (Initial value: 0000 0000)

TFF1	Timer F/F1 control	0: Clear	1: Set	R/W
ACAP1	Auto capture control	0:Auto-capture disable	1:Auto-capture enable	R/W
MCAP1	Pulse width measurement mode control	0:Double edge capture	1:Single edge capture	R/W
METT1	External trigger timer mode control	0:Trigger start	1:Trigger start and stop	R/W
MPPG1	PPG output control	0:Continuous pulse generation	1:One-shot	R/W

TC1S — TC1 start control (R/W)

	Timer	Extrigger	Event	Window	Pulse	PPG
00: Stop and counter clear	O	O	O	O	O	O
01: Command start	O	–	–	–	–	O
10: Rising edge start (Ex-trigger/Pulse/PPG) Rising edge count (Event) Positive logic count (Window)	–	O	O	O	O	O
11: Falling edge start (Ex-trigger/Pulse/PPG) Falling edge count (Event) Negative logic count (Window)	–	O	O	O	O	O

TC1CK — TC1 source clock select [Hz] (R/W)

	NORMAL1/2, IDLE1/2 mode DV7CK = 0	NORMAL1/2, IDLE1/2 mode DV7CK = 1	Divider	SLOW, SLEEP mode
00	$fc/2^{11}$	$fs/2^3$	DV9	$fs/2^3$
01	$fc/2^7$	$fc/2^7$	DV5	–
10	$fc/2^3$	$fc/2^3$	DV1	–
11	External clock (TC1 pin input)			

TC1M — TC1 operating mode select (R/W)

00: Timer/external trigger timer/event counter mode
01: Window mode
10: Pulse width measurement mode
11: PPG (Programmable pulse generate) output mode

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

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

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

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

Note 5: To set the timer registers, the following relationship must be satisfied.
TC1DRA > TC1DRB > 1 (PPG output mode), TC1DRA > 1 (other modes)

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

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

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

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

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

8.3 Function

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

8.3.1 Timer mode

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

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

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

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

```
LDW     (TC1DRA), 1E84H        ; Sets the timer register (1 s ÷ 2^11/fc = 1E84H)
DI                             ; IMF= "0"
SET     (EIRL). 5              ; Enables INTTC1
EI                             ; IMF= "1"
LD      (TC1CR), 00000000B     ; Selects the source clock and mode
LD      (TC1CR), 00010000B     ; Starts TC1
```

Example 2 :Auto-capture

```
LD      (TC1CR), 01010000B     ; ACAP1 ← 1
:       :
LD      WA, (TC1DRB)           ; Reads the capture value
```

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

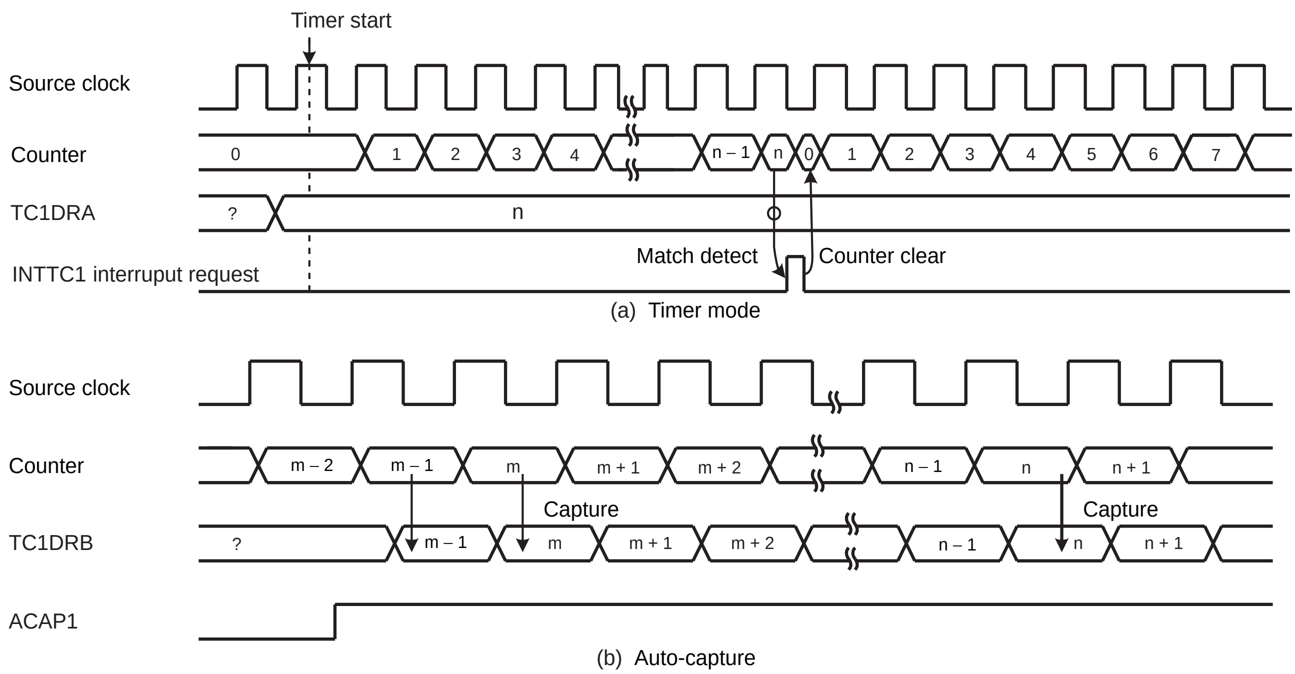


Figure 8-2 Timer Mode Timing Chart

8.3.2 External Trigger Timer Mode

In the external trigger timer mode, the up-counter starts counting by the input pulse triggering of the TC1 pin, and counts up at the edge of the internal clock. For the trigger edge used to start counting, either the rising or falling edge is defined in TC1CR<TC1S>.

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

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

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

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

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

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

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

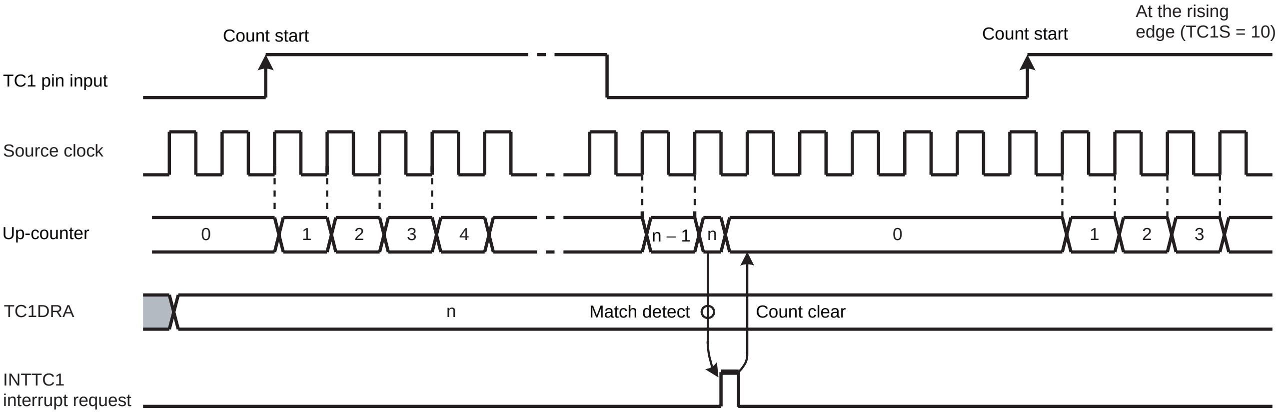
Since the TC1 pin input has the noise rejection, pulses of 4/fc [s] or less are rejected as noise. A pulse width of 12/fc [s] or more is required to ensure edge detection. The rejection circuit is turned off in the SLOW1/2 or SLEEP1/2 mode, but a pulse width of one machine cycle or more is required.

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

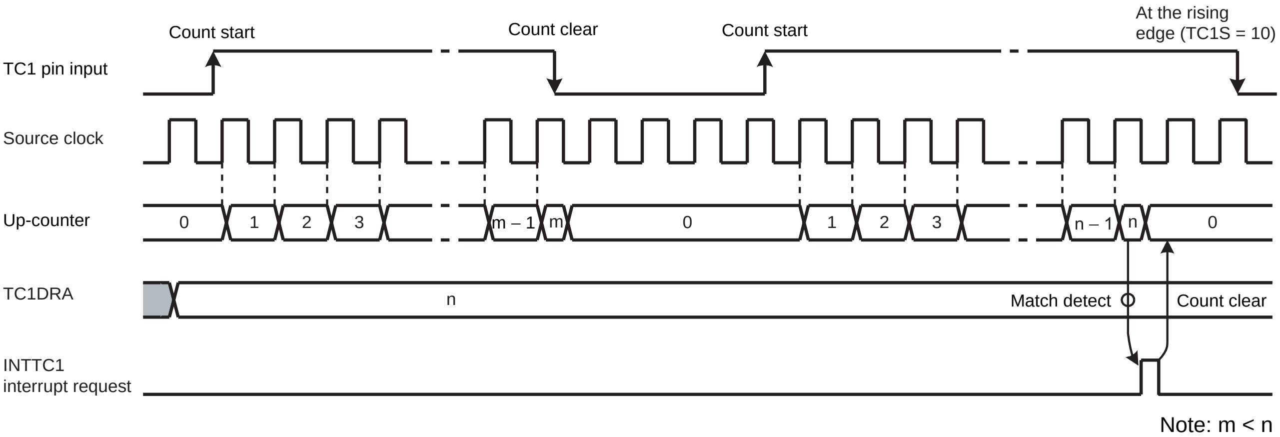
```
LDW    (TC1DRA), 007DH      ; 1ms ÷ 2^7/fc = 7DH
DI                          ; IMF= “0”
SET    (EIRL). 5            ; Enables INTTC1 interrupt
EI                          ; IMF= “1”
LD     (TC1CR), 00000100B   ; Selects the source clock and mode
LD     (TC1CR), 00100100B   ; Starts TC1 external trigger, METT1 = 0
```

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

```
LDW    (TC1DRA), 01F4H      ; 4 ms ÷ 2^7/fc = 1F4H
DI                          ; IMF= “0”
SET    (EIRL). 5            ; Enables INTTC1 interrupt
EI                          ; IMF= “1”
LD     (TC1CR), 00000100B   ; Selects the source clock and mode
LD     (TC1CR), 01110100B   ; Starts TC1 external trigger, METT1 = 1
```



(a) Trigger start (METT1 = 0)



(b) Trigger start and stop (METT1 = 1)

Figure 8-3 External Trigger Timer Mode Timing Chart

8.3.3 Event Counter Mode

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

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

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

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

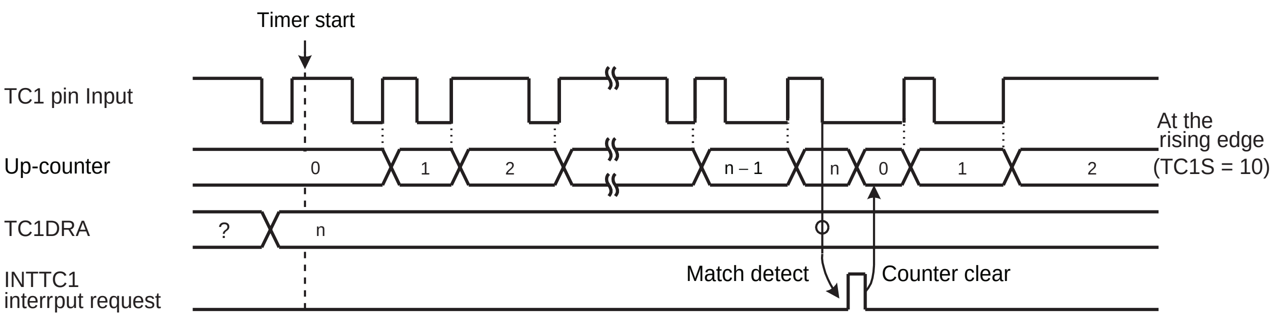


Figure 8-4 Event Counter Mode Timing Chart

Table 8-2 Input Pulse Width to TC1 Pin

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

8.3.4 Window Mode

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

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

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

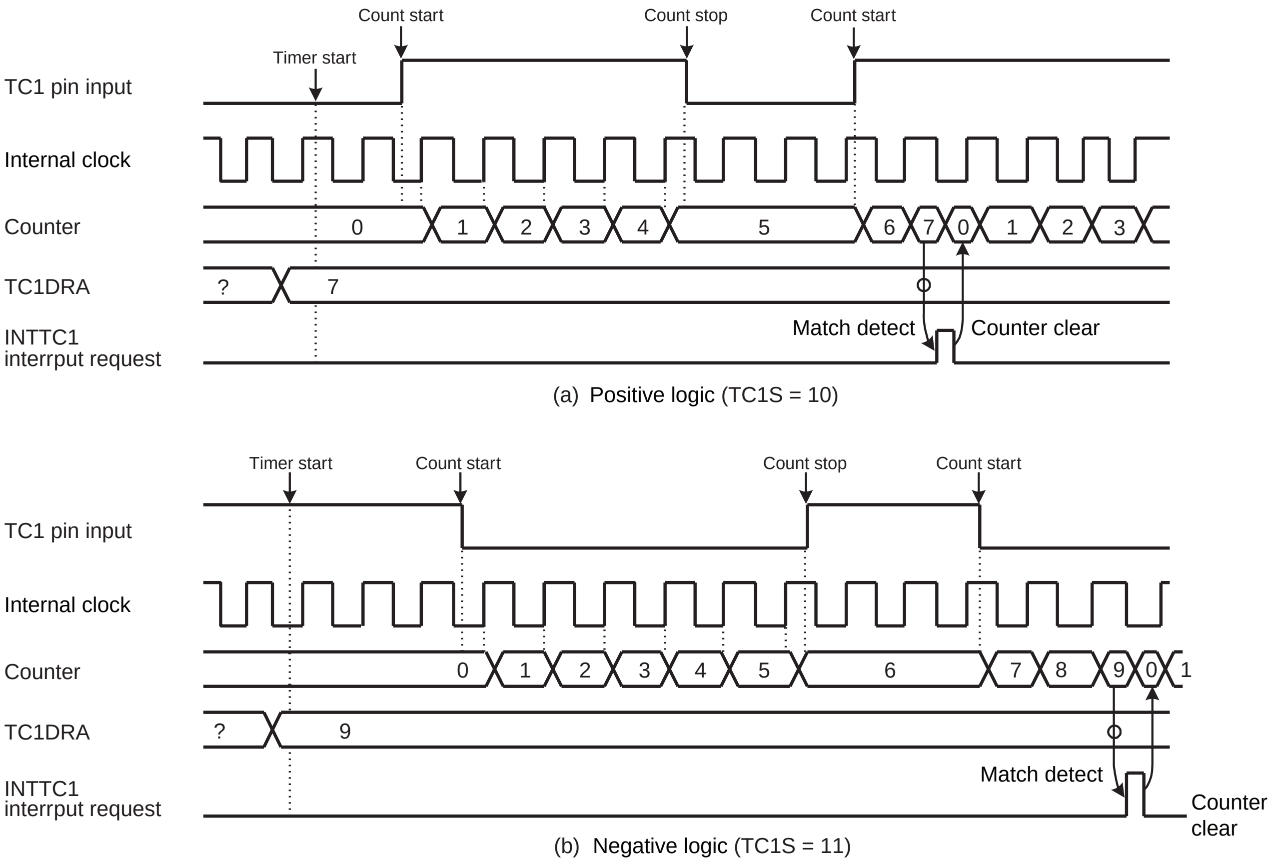


Figure 8-5 Window Mode Timing Chart

8.3.5 Pulse Width Measurement Mode

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

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

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

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

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

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

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

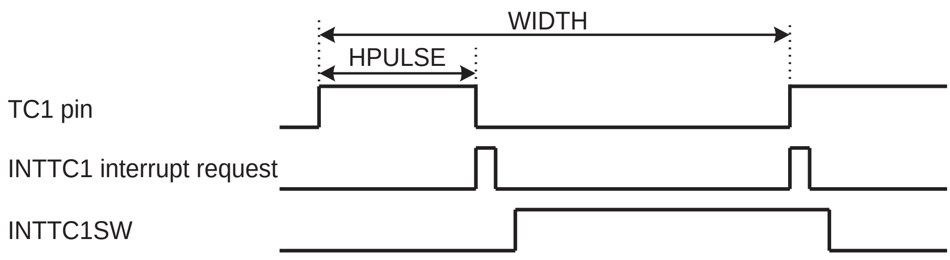
Note 1: The captured value must be read from TC1DRB until the next trigger edge is detected. If not read, the captured value becomes a don’t care. It is recommended to use a 16-bit access instruction to read the captured value from TC1DRB.

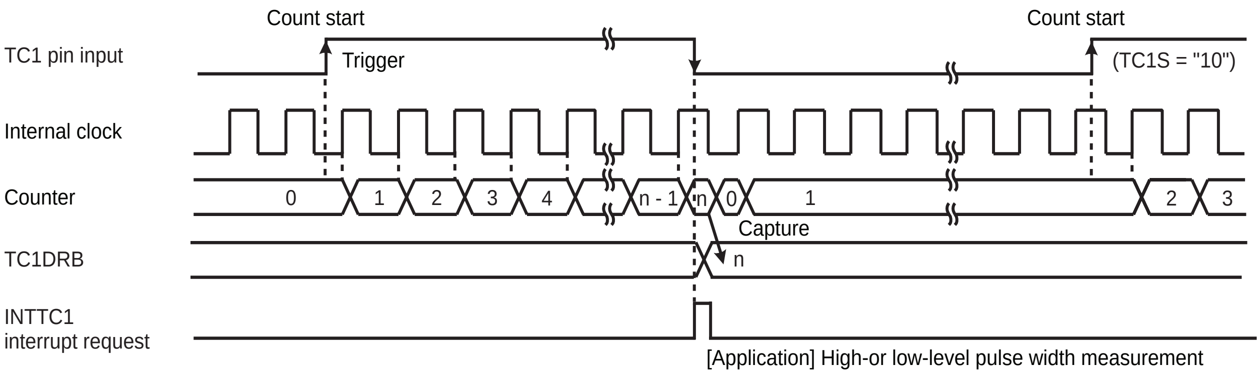
Note 2: For the single-edge capture, the counter after capturing the value stops at “1” until detecting the next edge. Therefore, the second captured value is “1” larger than the captured value immediately after counting starts.

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

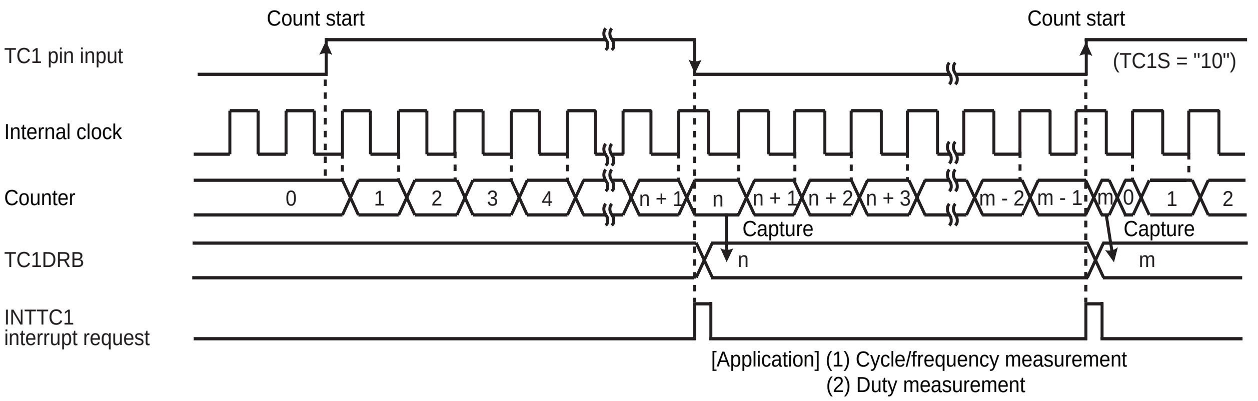
Example :Duty measurement (resolution fc/2^7 [Hz])

```
            CLR     (INTTC1SW). 0         ; INTTC1 service switch initial setting
                                            Address set to convert INTTC1SW at each INTTC1
            LD      (TC1CR), 00000110B    ; Sets the TC1 mode and source clock
            DI                            ; IMF= "0"
            SET     (EIRL). 5             ; Enables INTTC1
            EI                            ; IMF= "1"
            LD      (TC1CR), 00100110B    ; Starts TC1 with an external trigger at MCAP1 = 0
             :
PINTTC1:    CPL     (INTTC1SW). 0         ; INTTC1 interrupt, inverts and tests INTTC1 service switch
            JRS     F, SINTTC1
            LD      A, (TC1DRBL)          ; Reads TC1DRB (High-level pulse width)
            LD      W,(TC1DRBH)
            LD      (HPULSE), WA          ; Stores high-level pulse width in RAM
            RETI
SINTTC1:    LD      A, (TC1DRBL)          ; Reads TC1DRB (Cycle)
            LD      W,(TC1DRBH)
            LD      (WIDTH), WA           ; Stores cycle in RAM
             :
            RETI                          ; Duty calculation
            :
VINTTC1:    DW      PINTTC1               ; INTTC1 Interrupt vector
```





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



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

Figure 8-6 Pulse Width Measurement Mode

8.3.6 Programmable Pulse Generate (PPG) Output Mode

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

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

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

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

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

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

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

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

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

Note 3: In the PPG mode, the following relationship must be satisfied.
TC1DRA > TC1DRB

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

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

```
                    Setting port
LD      (TC1CR), 10000111B      ; Sets the PPG mode, selects the source clock
LDW     (TC1DRA), 007DH         ; Sets the cycle (1 ms ÷ 2^7/fc ms = 007DH)
LDW     (TC1DRB), 0019H         ; Sets the low-level pulse width (200 µs ÷ 2^7/fc = 0019H)
LD      (TC1CR), 10010111B      ; Starts the timer
```

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

```
                    Setting port
LD      (TC1CR), 10000111B      ; Sets the PPG mode, selects the source clock
LDW     (TC1DRA), 007DH         ; Sets the cycle (1 ms ÷ 2^7/fc µs = 007DH)
LDW     (TC1DRB), 0019H         ; Sets the low-level pulse width (200 µs ÷ 2^7/fc = 0019H)
LD      (TC1CR), 10010111B      ; Starts the timer
:       :
LD      (TC1CR), 10000111B      ; Stops the timer
LD      (TC1CR), 10000100B      ; Sets the timer mode
LD      (TC1CR), 00000111B      ; Sets the PPG mode, TFF1 = 0
LD      (TC1CR), 00010111B      ; Starts the timer
```

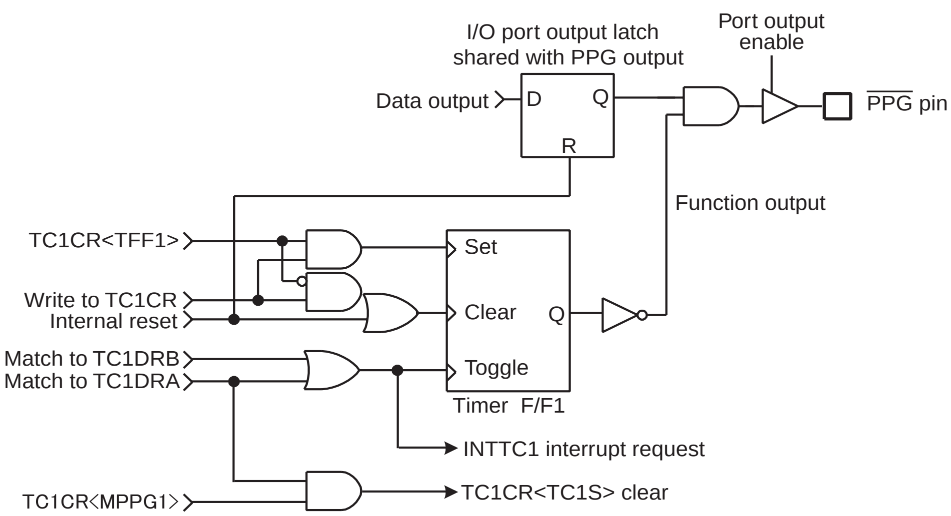
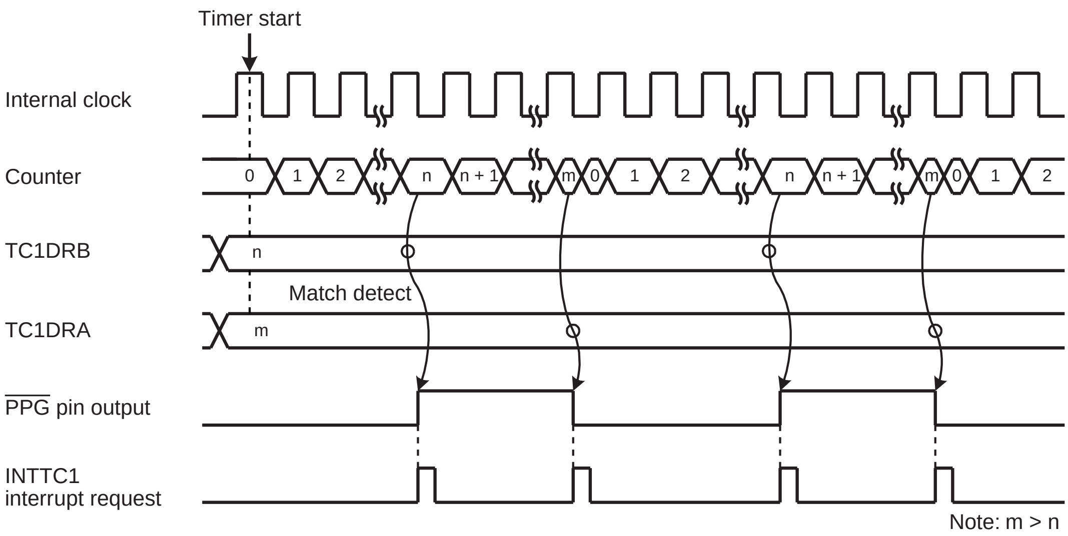
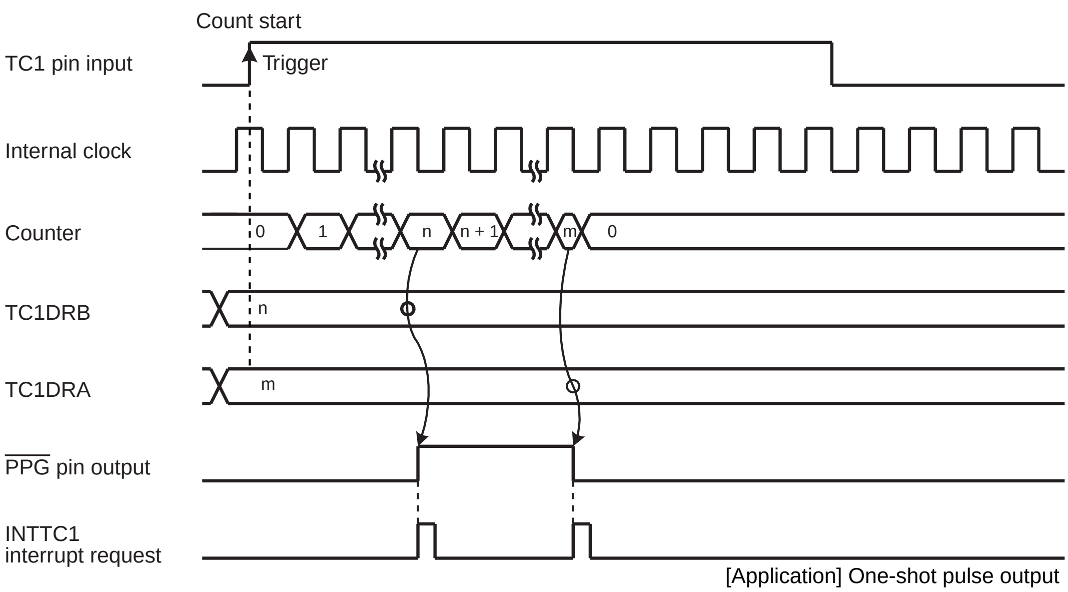


Figure 8-7 $\overline{\text{PPG}}$ Output



(a) Continuous pulse generation (TC1S = 01)

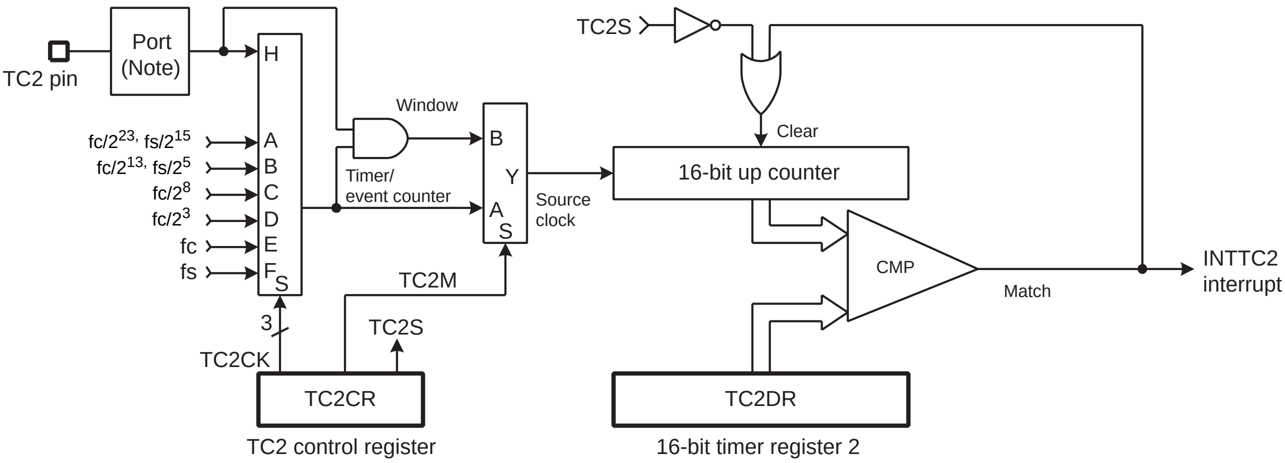


(b) One-shot pulse generation (TC1S = 10)

Figure 8-8 PPG Mode Timing Chart

9. 16-Bit Timer/Counter2 (TC2)

9.1 Configuration



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

Figure 9-1 Timer/Counter2 (TC2)

9.2 Control

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

TC2DR (0025H, 0024H)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC2DRH (0025H)								TC2DRL (0024H)							

(Initial value: 1111 1111 1111 1111) R/W

TC2CR (0013H)

7	6	5	4	3	2	1	0
		TC2S	TC2CK				TC2M

(Initial value: **00 00*0)

TC2S	TC2 start control	0:Stop and counter clear 1:Start						R/W
TC2CK	TC2 source clock select Unit : [Hz]		NORMAL1/2, IDLE1/2 mode		Divider	SLOW1/2 mode	SLEEP1/2 mode	R/W
			DV7CK = 0	DV7CK = 1				
		000	fc/2^{23}	fs/2^{15}	DV21	fs/2^{15}	fs/2^{15}	
		001	fc/2^{13}	fs/2^{5}	DV11	fs/2^{5}	fs/2^{5}	
		010	fc/2^{8}	fc/2^{8}	DV6	–	–	
		011	fc/2^{3}	fc/2^{3}	DV1	–	–	
		100	–	–	–	fc (Note7)	–	
		101	fs	fs	–	–	–	
		110	Reserved					
		111	External clock (TC2 pin input)					
TC2M	TC2 operating mode select	0:Timer/event counter mode 1:Window mode						R/W

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

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

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

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

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

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

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

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

9.3 Function

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

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

9.3.1 Timer mode

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

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

Table 9-1 Source Clock (Internal clock) for Timer/Counter2 (at fc = 16 MHz, DV7CK=0)

TC2CK	NORMAL1/2, IDLE1/2 mode				SLOW1/2 mode		SLEEP1/2 mode	
	DV7CK = 0		DV7CK = 1					
	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting
000	524.29 [ms]	9.54 [h]	1 [s]	18.2 [h]	1 [s]	18.2 [h]	1 [s]	18.2 [h]
001	512.0 [ms]	33.55 [s]	0.98 [ms]	1.07 [min]	0.98 [ms]	1.07 [min]	0.98 [ms]	1.07 [min]
010	16.0 [ms]	1.05 [s]	16.0 [ms]	1.05 [s]	–	–	–	–
011	0.5 [ms]	32.77 [ms]	0.5 [ms]	32.77 [ms]	–	–	–	–
100	–	–	–	–	62.5 [ns]	–	–	–
101	30.52 [ms]	2 [s]	30.52 [ms]	2 [s]	–	–	–	–

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

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

```
LDW     (TC2DR), 061AH        ; Sets  TC2DR (25 ms ³ 2^8/fc = 061AH)
DI                            ; IMF= "0"
SET     (EIRH). 5             ; Enables INTTC2 interrupt
EI                            ; IMF= "1"
LD      (TC2CR), 00001000B    ; Source clock / mode select
LD      (TC2CR), 00101000B    ; Starts Timer
```

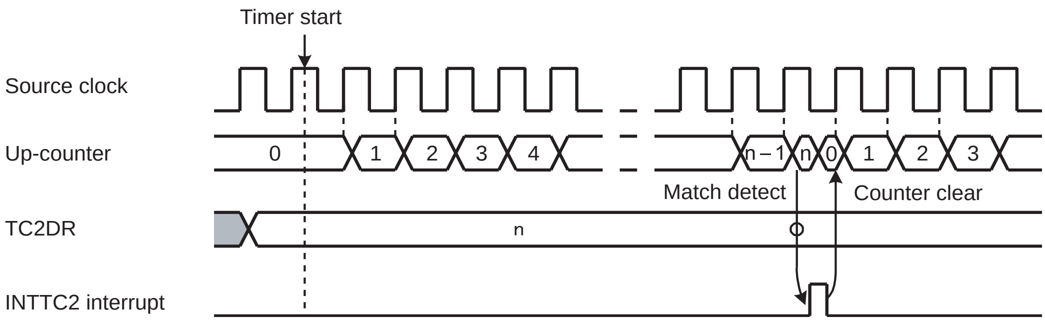


Figure 9-2 Timer Mode Timing Chart

9.3.2 Event counter mode

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

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

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

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

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

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

	Minimum Input Pulse Width [s]	
	NORMAL1/2, IDLE1/2 mode	SLOW1/2, SLEEP1/2 mode
“H” width	2^3/fc	2^3/fs
“L” width	2^3/fc	2^3/fs

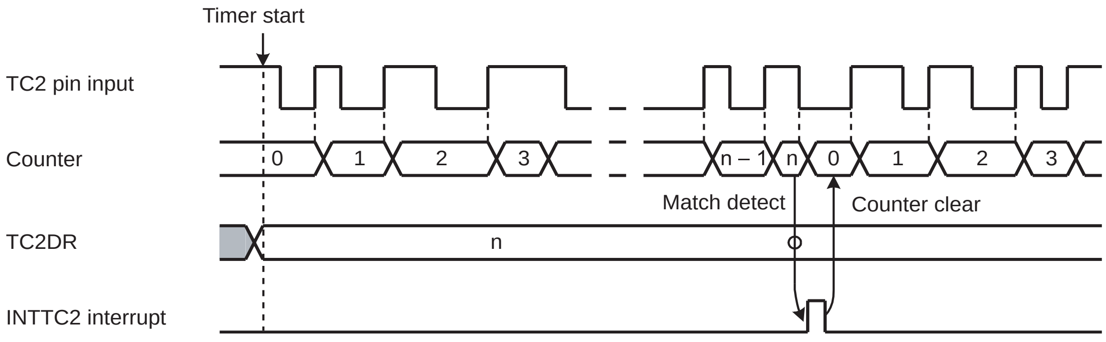


Figure 9-3 Event Counter Mode Timing Chart

9.3.3 Window mode

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

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

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

Example :Generates an interrupt, inputting “H” level pulse width of 120 ms or more. (at fc = 16 MHz, TBTCR<DV7CK> = “0”)

```
LDW     (TC2DR), 00EAH        ; Sets TC2DR (120 ms ³ 2^13/fc = 00EAH)
DI                            ; IMF= “0”
SET     (EIRH). 5             ; Enables  INTTC2 interrupt
EI                            ; IMF= “1”
LD      (TC2CR), 00000101B    ; TC2sorce clock / mode select
LD      (TC2CR), 00100101B    ; Starts TC2
```

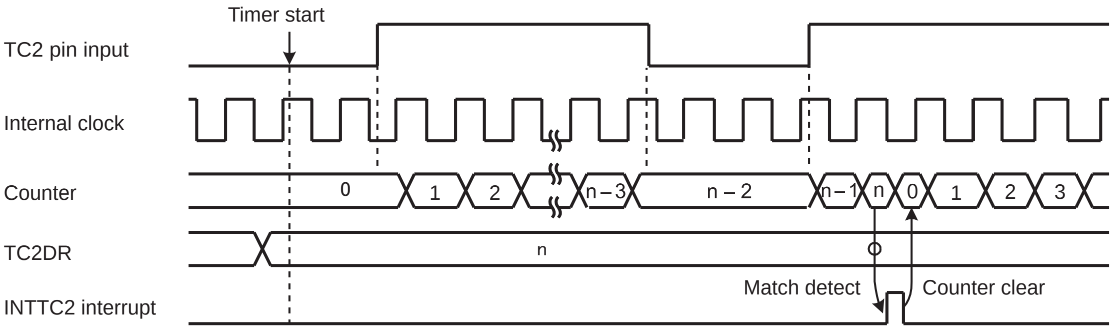
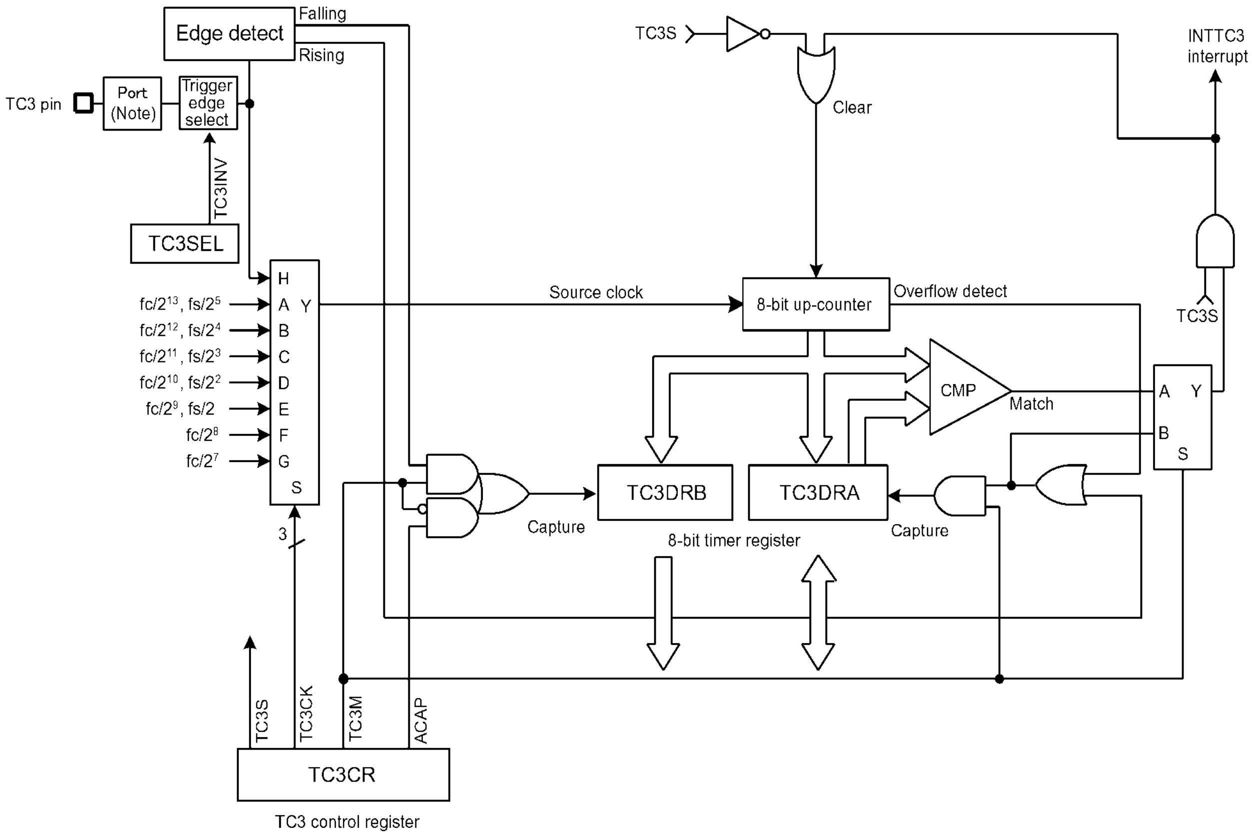


Figure 9-4 Window Mode Timing Chart

10. 8-Bit TimerCounter 3 (TC3)

10.1 Configuration



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

Figure 10-1 TimerCounter 3 (TC3)

10.2 TimerCounter Control

The TimerCounter 3 is controlled by the TimerCounter 3 control register (TC3CR) and two 8-bit timer registers (TC3DRA and TC3DRB).

Timer Register and Control Register

	7	6	5	4	3	2	1	0	
TC3DRA (0010H)									Read/Write (Initial value: 1111 1111)
TC3DRB (0011H)									Read only (Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
TC3CR (0012H)		ACAP		TC3S	TC3CK			TC3M	(Initial value: *0*0 0000)

ACAP	Auto capture control	0: – 1: Auto capture					R/W
TC3S	TC3 start control	0: Stop and counter clear 1: Start					R/W
TC3CK	TC3 source clock select [Hz]		NORMAL1/2, IDLE1/2 mode		Divider	SLOW1/2, SLEEP1/2 mode	R/W
			DV7CK = 0	DV7CK = 1			
		000	$fc/2^{13}$	$fs/2^{5}$	DV11	$fs/2^{5}$	
		001	$fc/2^{12}$	$fs/2^{4}$	DV10	$fs/2^{4}$	
		010	$fc/2^{11}$	$fs/2^{3}$	DV9	$fs/2^{3}$	
		011	$fc/2^{10}$	$fs/2^{2}$	DV8	$fs/2^{2}$	
		100	$fc/2^{9}$	fs/2	DV7	fs/2	
		101	$fc/2^{8}$	$fc/2^{8}$	DV6	–	
		110	$fc/2^{7}$	$fc/2^{7}$	DV5	–	
		111	External clock (TC3 pin input)				
TC3M	TC3 operating mode select	0: Timer/event counter mode 1: Capture mode					R/W

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care
Note 2: Set the operating mode and source clock when TimerCounter stops (TC3S = 0).
Note 3: To set the timer registers, the following relationship must be satisfied.
TC3DRA > 1 (Timer/event counter mode)
Note 4: Auto-capture (ACAP) can be used only in the timer and event counter modes.
Note 5: When the read instruction is executed to TC3CR, the bit 5 and 7 are read as a don't care.
Note 6: Do not program TC3DRA when the timer is running (TC3S = 1).
Note 7: When the STOP mode is entered, the start control (TC3S) is cleared to 0 automatically, and the timer stops. After the STOP mode is exited, TC3S must be set again to use the timer counter.

TimerCounter 3 Input Control Register

	7	6	5	4	3	2	1	0	
TC3SEL (0029H)								TC3INV	Read/Write (Initial value: **** ***0)

TC3INV	TC3 input control		Event counter mode	Capture mode	R/W
		0: 1:	Count at the rising edge Count at the falling edge	An interrupt is generated at the rising edge. An interrupt is generated at the falling edge.	

Note: When the read instruction is executed to TC3SEL, the bit 7 to 1 are read as a don't care.

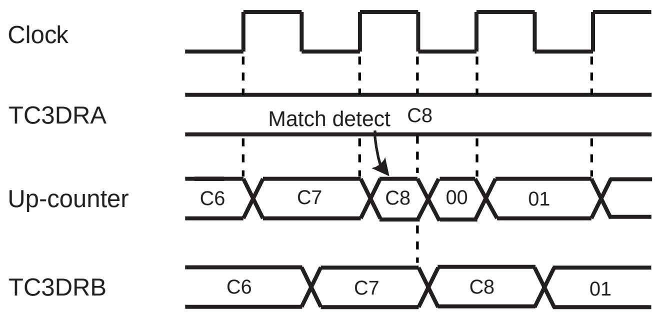
10.3 Function

TimerCounter 3 has three types of operating modes: timer, event counter and capture modes.

10.3.1 Timer mode

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the timer register 3A (TC3DRA) value is detected, an INTTC3 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting. Setting TC3CR<ACAP> to 1 captures the up-counter value into the timer register B (TC3DRB) with the auto-capture function. The count value during timer operation can be checked by executing the read instruction to TC3DRB.

Note: 00H which is stored in the up-counter immediately after detection of a match is not captured into TC3DRB. (Figure 10-2)

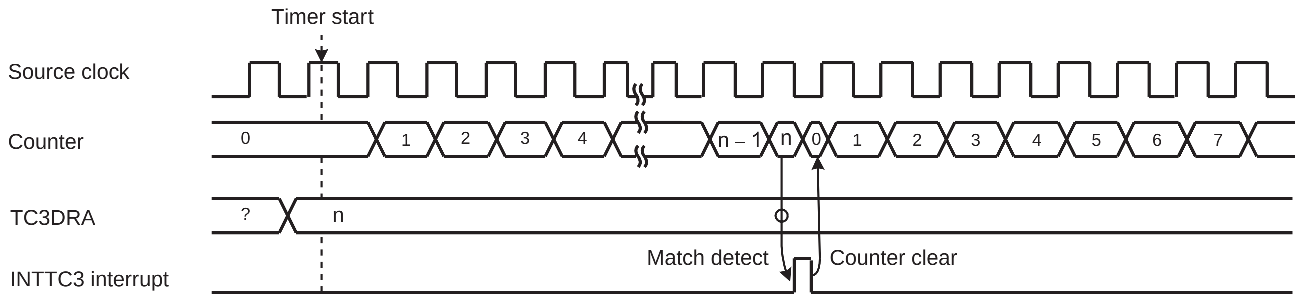


Note: In the case that TC3DRB is C8H

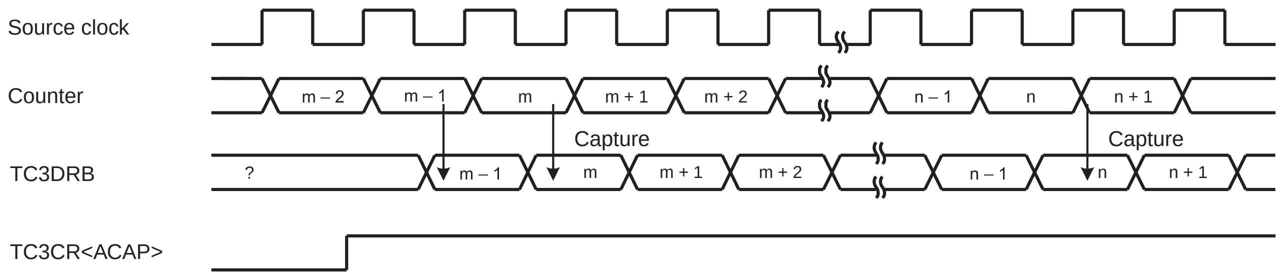
Figure 10-2 Auto-Capture Function

Table 10-1 Source Clock for TimerCounter 3 (Example: fc = 16 MHz, fs = 32.768 kHz)

TC3CK	NORMAL1/2, IDLE1/2 mode				SLOW1/2, SLEEP1/2 mode	
	DV7CK = 0		DV7CK = 1			
	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Reso-lution [μs]	Maximum Time Setting [ms]
000	512	130.6	976.56	249.0	976.56	249.0
001	256	65.3	488.28	124.5	488.28	124.5
010	128	32.6	244.14	62.3	244.14	62.3
011	64	16.3	122.07	31.1	122.07	31.1
100	32	8.2	61.01	15.6	61.01	15.6
101	16	4.1	16.0	4.1	–	–
110	8	2.0	8.0	2.0	–	–



(a) Timer mode



(b) Auto capture

Figure 10-3 Timer Mode Timing Chart

10.3.2 Event Counter Mode

In the event counter mode, the up-counter counts up at the edge of the input pulse to the TC3 pin. Either the rising or falling edge of the input pulse is programmed as the count up edge in TC3SEL<TC3INV>.

When a match between the up-counter and TC3DRA value is detected, an INTTC3 interrupt is generated and up-counter is cleared. After being cleared, the up-counter restarts counting at each edge of the input pulse to the TC3 pin. Since a match between the up-counter and TC3DRA value is detected at the edge opposite to the selected edge, an INTTC3 interrupt request is generated at the edge opposite to the selected edge immediately after the up-counter reaches the value set in TC3DRA.

The maximum applied frequencies are shown in Table 10-2. The pulse width larger than one machine cycle is required for high-going and low-going pulses.

Setting TC3CR<ACAP> to 1 captures the up-counter value into TC3DRB with the auto-capture function. The count value during a timer operation can be checked by the read instruction to TC3DRB.

Note: 00H which is stored in the up-counter immediately after detection of a match is not captured into TC3DRB. (Figure 10-2)

Example :Inputting 50 Hz pulse to TC3, and generating interrupts every 0.5 s

```
LD      (TC3SEL), 00000000B     : Selects the count-up edge.
LD      (TC3CR), 00001110B      : Sets the clock mode
LD      (TC3DRA), 19H           : 0.5 s ÷ 1/50 = 25 = 19H
LD      (TC3CR), 00011110B      : Starts TC3.
```

Table 10-2 Maximum Frequencies Applied to TC3

	Minimum Pulse Width	
	NORMAL1/2, IDLE1/2 mode	SLOW1/2, SLEEP1/2 mode
High-going	2^2/fc	2^2/fs
Low-going	2^2/fc	2^2/fs

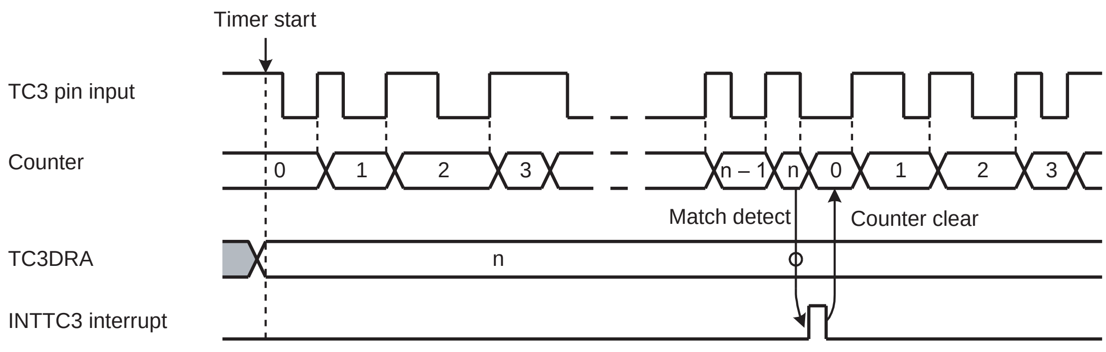


Figure 10-4 Event Counter Mode Timing Chart (TC3SEL<TC3INV> = 0)

10.3.3 Capture Mode

In the capture mode, the pulse width, frequency and duty cycle of the pulse input to the TC3 pin are measured with the internal clock. The capture mode is used to decode remote control signals, and identify AC50/60 Hz.

Either the rising or falling edge is programmed in TC3SEL<TC3IVN> as the INTTC3 interrupt generation edge. Typically, program TC3SEL<TC3INV> = 0 when the first capture is performed at the falling edge, and TC3SEL<TC3INV> = 1 when performed at the rising edge.

- When TC3SEL<TC3INV> = 0

  When the falling edge of the TC3 input is detected after the timer starts, the up-counter value is captured into TC3DRB. Hereafter, whenever the rising edge is detected, the up-counter value is captured into TC3DRA and the INTTC3 interrupt request is generated. The up-counter is cleared at this time. Generally, read TC3DRB and TC3DRA during INTTC3 interrupt processing. After the up-counter is cleared, counting is continued and the next up-counter value is captured into TC3DRB.

  When the rising edge is detected immediately after the timer starts, the up-counter value is captured into TCDRA only, but not into TC3DRB. The INTTC3 interrupt request is generated. When the read instruction is executed to TC3DRB at this time, the value at the completion of the last capture (FF immediately after a reset) is read.

- When TC3SEL<TC3INV> = 1

  When the rising edge of the TC3 input is detected after the timer starts, the up-counter value is captured into TC3DRB. Hereafter, whenever the falling edge is detected, the up-counter value is captured into TC3DRA and the INTTC3 interrupt request is generated. The up-counter is cleared at this time. Generally, read TC3DRB and TC3DRA during INTTC3 interrupt processing. After the up-counter is cleared, counting is continued and the next up-counter value is captured into TC3DRB.

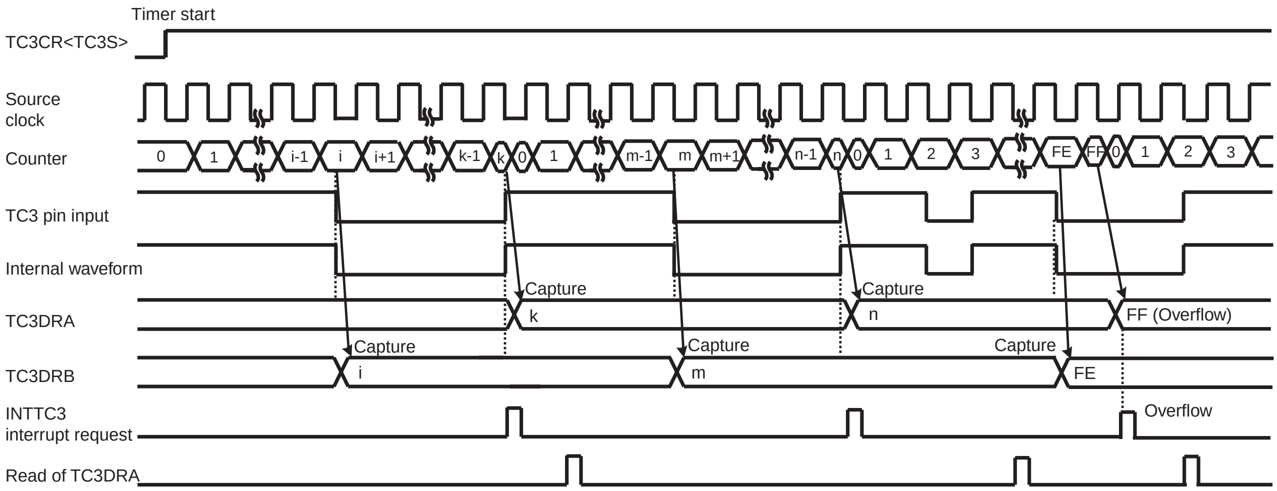
  When the falling edge is detected immediately after the timer starts, the up-counter value is captured into TC3DRA only, but not into TC3DRB. The INTTC3 interrupt request is generated. When the read instruction is executed to TC3DRB at this time, the value at the completion of the last capture (FF immediately after a reset) is read.

Table 10-3 Trigger Edge Programmed in TC3SEL<TC3INV>

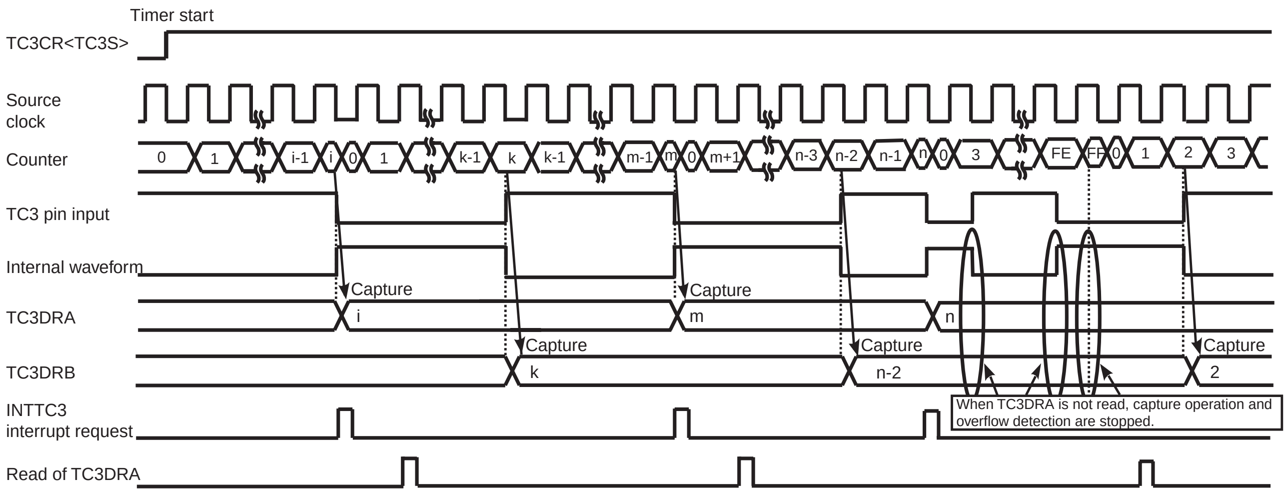
TC3SEL<TC3INV>	Capture into TC3DRB	Capture into TC3DRA	INTTC3 Interrupt Request
0	Falling edge	Rising edge	
1	Rising edge	Falling edge	

The minimum input pulse width must be larger than one cycle width of the source clock programmed in TC3CR<TC3CK>.

The INTTC3 interrupt request is generated if the up-counter overflow (FFH) occurs during capture operation before the edge is detected. TC3DRA is set to FFH and the up-counter is cleared. Counting is continued by the up-counter, but capture operation and overflow detection are stopped until TC3DRA is read. Generally, read TC3DRB first because capture operation and overflow detection resume by reading TC3DRA.



a) TC3SEL<TC3INV>=0

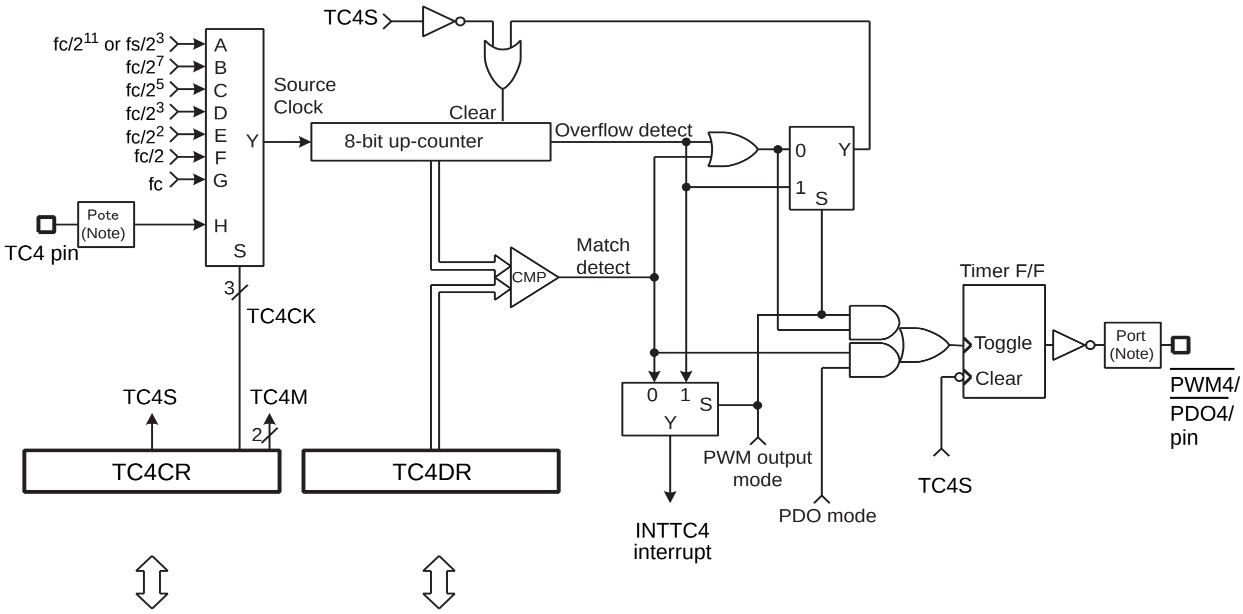


b) TC3SEL<TC3INV>=1

Figure 10-5 Capture Mode Timing Chart

11. 8-Bit TimerCounter 4 (TC4)

11.1 Configuration



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

Figure 11-1 TimerCounter 4 (TC4)

11.2 TimerCounter Control

The TimerCounter 4 is controlled by the TimerCounter 4 control register (TC4CR) and timer registers 4 (TC4DR).

Timer Register and Control Register

	7	6	5	4	3	2	1	0	
TC4DR (0018)									Read/Write (Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
TC4CR (0014)			TC4S	TC4CK			TC4M		Read/Write (Initial value: **00 0000)

TC4S	TC4 start control	0: Stop and counter clear 1: Start					R/W
TC4CK	TC4 source clock select [Hz]		NORMAL1/2, IDLE1/2 mode		Divider	SLOW1/2, SLEEP1/2 mode	R/W
			DV7CK = 0	DV7CK = 1			
		000	$fc/2^{11}$	$fs/2^3$	DV9	$fs/2^3$	
		001	$fc/2^7$	$fc/2^7$	DV5	–	
		010	$fc/2^5$	$fc/2^5$	DV3	–	
		011	$fc/2^3$	$fc/2^3$	DV1	–	
		100	$fc/2^2$	$fc/2^2$	–	–	
		101	fc/2	fc/2	–	–	
		110	fc	fc	–	–	
		111	External clock (TC4 pin input)				
TC4M	TC4 operating mode select	00: Timer/event counter mode 01: Reserved 10: Programmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode					R/W

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

Note 2: To set the timer registers, the following relationship must be satisfied.
$1 \leq TC4DR \leq 255$

Note 3: To start timer operation (TC4S = 0 → 1) or disable timer operation (TC4S = 1→ 0), do not change the TC4CR<TC4M, TC4CK> setting. During timer operation (TC4S = 1→ 1), do not change it, either. If the setting is programmed during timer operation, counting is not performed correctly.

Note 4: The event counter and PWM output modes are used only in the NOMAL1/2 and IDLE1/2 modes.

Note 5: When the STOP mode is entered, the start control (TC4S) is cleared to "0" automatically.

Note 6: The bit 6 and 7 of TC4CR are read as a don't care when these bits are read.

Note 7: In the timer, event counter and PDO modes, do not change the TC4DR setting when the timer is running.

Note 8: When the high-frequency clock fc exceeds 10 MHz, do not select the source clock of TC4CK = 110.

Note 9: The operating clock fs can not be used in NORMAL1 or IDEL1 mode (when low-frequency oscillation is stopped.)

Note 10:For available source clocks depending on the operation mode, refer to the following table.

		Timer Mode	Event Counter Mode	PDO Mode	PWM Mode
TC4CK	000	O	–	O	–
	001	O	–	O	–
	010	O	–	O	–
	011	O	–	–	O
	100	–	–	–	O
	101	–	–	–	O
	110	–	–	–	O
	111	–	O	–	×

Note: O : Available source clock

11.3 Function

TimerCounter 4 has four types of operating modes: timer, event counter, programmable divider output (PDO), and pulse width modulation (PWM) output modes.

11.3.1 Timer Mode

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the TC4DR value is detected, an INTTC4 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting.

Table 11-1 Source Clock for TimerCounter 4 (Example: fc = 16 MHz, fs = 32.768 kHz)

TC4CK	NORMAL1/2, IDLE1/2 Mode				SLOW1/2, SLEEP1/2 Mode	
	DV7CK = 0		DV7CK = 1			
	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]
000	128.0	32.6	244.14	62.2	244.14	62.2
001	8.0	2.0	8.0	2.0	–	–
010	2.0	0.510	2.0	0.510	–	–
011	0.5	0.128	0.5	0.128	–	–

11.3.2 Event Counter Mode

In the event counter mode, the up-counter counts up at the rising edge of the input pulse to the TC4 pin.

When a match between the up-counter and the TC4DR value is detected, an INTTC4 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at rising edge of the TC4 pin. Since a match is detected at the falling edge of the input pulse to the TC4 pin, the INTTC4 interrupt request is generated at the falling edge immediately after the up-counter reaches the value set in TC4DR.

The minimum pulse width applied to the TC4 pin are shown in Table 11-2. The pulse width larger than two machine cycles is required for high- and low-going pulses.

Note: The event counter mode can not used in the SLOW1/2 and SLEEP1/2 modes since the external clock is not supplied in these modes.

Table 11-2 External Source Clock for TimerCounter 4

	Minimum Pulse Width
	NORMAL1/2, IDLE1/2 mode
High-going	2^3/fc
Low-going	2^3/fc

11.3.3 Programmable Divider Output (PDO) Mode

The programmable divider output (PDO) mode is used to generated a pulse with a 50% duty cycle by counting with the internal clock.

When a match between the up-counter and the TC4DR value is detected, the logic level output from the PDO4 pin is switched to the opposite state and INTTC4 interrupt request is generated. The up-counter is cleared at this time and then counting is continued. When a match between the up-counter and the TC4DR value is detected, the logic level output from the PDO4 pin is switched to the opposite state again and INTTC4 interrupt request is generated. The up-counter is cleared at this time, and then counting and PDO are continued.

When the timer is stopped, the PDO4 pin is high. Therefore, if the timer is stopped when the PDO4 pin is low, the duty pulse may be shorter than the programmed value.

Example :Generating 1024 Hz pulse (fc = 16.0 Mhz)

```
LD    (TC4CR), 00000110B    : Sets the PDO mode. (TC4M = 10, TC4CK = 001)
LD    (TC4DR), 3DH          : 1/1024 ÷ 2^7/fc ÷ 2 (half cycle period) = 3DH
LD    (TC4CR), 00100110B    : Start TC4
```

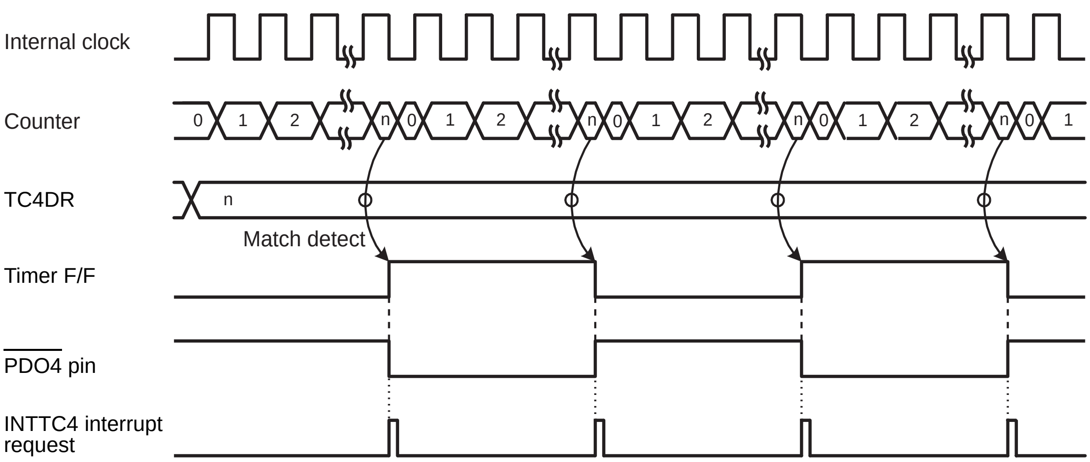


Figure 11-2 PDO Mode Timing Chart

11.3.4 Pulse Width Modulation (PWM) Output Mode

The pulse width modulation (PWM) output mode is used to generate the PWM pulse with up to 8 bits of resolution by an internal clock.

When a match between the up-counter and the TC4DR value is detected, the logic level output from the $\overline{\text{PWM}}$4 pin becomes low. The up-counter continues counting. When the up-counter overflow occurs, the $\overline{\text{PWM}}$4 pin becomes high. The INTTC4 interrupt request is generated at this time.

When the timer is stopped, the PWM4 pin is high. Therefore, if the timer is stopped when the PWM4 pin is low, one PMW cycle may be shorter than the programmed value.

TC4DR is serially connected to the shift register. If TC4DR is programmed during PWM output, the data set to TC4DR is not shifted until one PWM cycle is completed. Therefore, a pulse can be modulated periodically. For the first time, the data written to TC4DR is shifted when the timer is started by setting TC4CR<TC4S> to 1.

Note 1: The PWM output mode can be used only in the NORMAL1/2 and IDEL 1/2 modes.

Note 2: In the PWM output mode, program TC4DR immediately after the INTTC4 interrupt request is generated (typically in the INTTC4 interrupt service routine.) When the programming of TC4DR and the INTTC4 interrupt 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 issued.

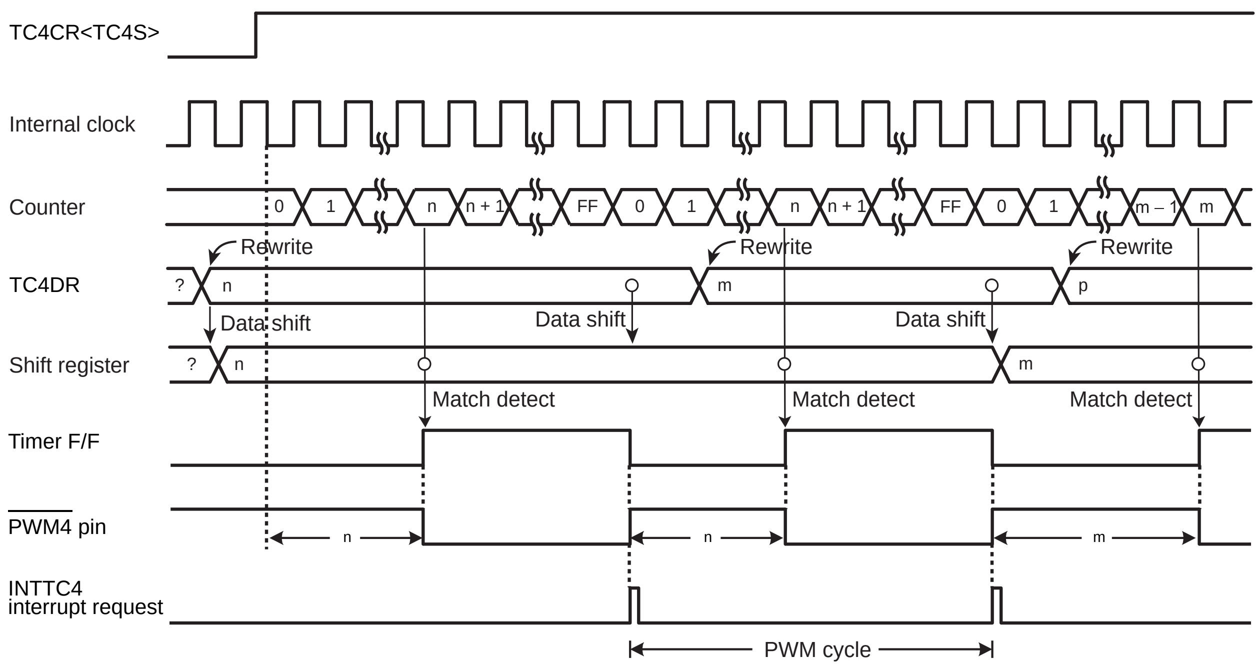


Figure 11-3 PWM output Mode Timing Chart (TC4)

Table 11-3 PWM Mode (Example: fc = 16 MHz)

TC4CK	NORMAL1/2, IDLE1/2 Mode			
	DV7CK = 0		DV7CK = 1	
	Resolution [ns]	Cycle [μs]	Resolution [ns]	Cycle [μs]
000	–	–	–	–
001	–	–	–	–
010	–	–	–	–
011	500	128	500	128
100	250	64	250	64
101	125	32	125	32
110	–	–	–	–

12. Synchronous Serial Interface (SIO)

The TMP86CK74AFG contain one SIO (synchronous serial interface) channel. It is connected to external devices via the SI, SO and $\overline{\text{SCK}}$ pins. The SI pin is used also as the P15 pin, the SO pin is used also as the P16 pin, and the $\overline{\text{SCK}}$ pin is used also as the P17 pin. Using these pins for serial interfacing requires setting the output latches of the each port to “1”.

SIO Functions.

- Transfer mode (8 bit)
- Receive mode (8 bit)
- Transfer/Receive mode (8 bit)
- Internal /External clock selection
- 32 bytes Buffer conbining Transfer and Receive

Table 12-1 lists the SIO1 register addresses.

Table 12-1 Control Registers

	SIO1	
	Register name	Address
SIO control register 1	SIOCR1	0019H
SIO control register 2	SIOCR2	001AH
SIO status register	SIOSR	001BH
SIO data buffer	SIOBUF	001CH

12.1 Configuration

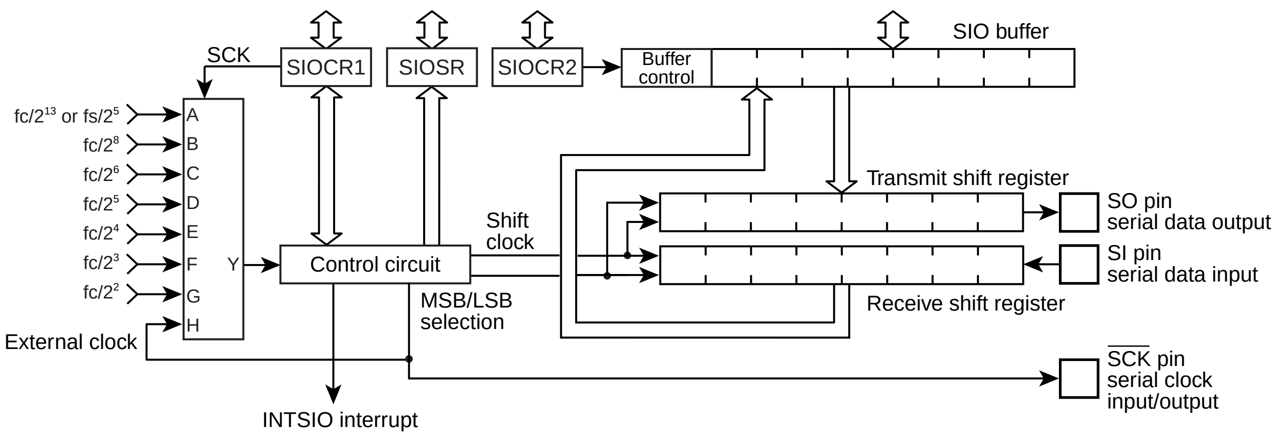


Figure 12-1 Configuration of the Serial Interface

12.2 Control

SIO is controlled using Serial Interface Control Register 1 (SIOCR1) and Serial Interface Control Register 2 (SIOCR2). The operating status of the serial interface can be determined by reading the Serial Interface Status Register (SIOSR).

Serial Interface Control Register 1

SIOCR1 (0019H)	7	6	5	4	3	2	1	0	
	SIOS	SIONH	SIOM		SIODIR	SCK			(Initial value: 0000 0000)

SIOS	Start/Stop a transfer.	0: Stop 1: Start					R/W
SIOINH	Continue/Abort a transfer (Note 1)	0: Continue transfer. 1: Abort transfer (automatically cleared after abort).					R/W
SIOM	Select transfer mode.	00: Transmit mode 01: Receive mode 10: Transmit/receive mode 11: Reserved					R/W
SIODIR	Select direction of transfer	0: MSB (transfer beginning with bit 7) 1: LSB (transfer beginning with bit 0)					R/W
SCK	Select a serial clock. (Note 2)		NORMAL1/2, IDLE1/2 mode		Source clock	SLOW1/2, SLEEP1/2 mode	R/W
			DV7CK = 0	DV7CK = 1			
		000	$fc/2^{13}$	$fs/2^5$	DV11	$fs/2^5$	
		001	$fc/2^8$	$fc/2^8$	DV6	–	
		010	$fc/2^6$	$fc/2^6$	DV4	–	
		011	$fc/2^5$	$fc/2^5$	DV3	–	
		100	$fc/2^4$	$fc/2^4$	DV2	–	
		101	$fc/2^3$	$fc/2^3$	DV1	–	
		110	$fc/2^2$	$fc/2^2$	$fc/2^2$	–	
		111	External clock (supplied from the $\overline{SCK}$ pin)	External clock (supplied from the $\overline{SCK}$ pin)	–	–	

Note 1: If SIOCR1<SIOINH> is set, SIOCR1<SIOS>, SIOSR<SIOF>, SIOSR<SEF>, SIOSR<TXF>, SIOSR<RXF>, SIOSR<TXERR>, and SIOSR<RXERR> are initialized.

Note 2: When selecting a serial clock, do not make such a setting that the serial clock rate will exceed 1 Mbps.

Note 3: Before setting SIOCR1<SIOS> to “1” or setting SIOCR1<SIOM>, SIOCR1<SIODIR>, or SIOCR1<SCK> to any value, make sure the SIO is idle (SIOSR<SIOF> = “0”).

Note 4: Reserved: Setting prohibited

Serial Interface Control Register 2

SIOCR2 (001AH)

7	6	5	4	3	2	1	0
"0"	"0"	"0"	SIORXD				

(Initial value: ***0 0000)

SIORXD	Set the number of data bytes to transmit/receive.	00H: 1-byte transfer 01H: 2-byte transfer 02H: 3-byte transfer 03H: 4-byte transfer : 1FH: 32-byte transfer	R/W

Note 1: Before setting the number of data bytes to transfer, make sure the SIO is idle (SIOSR<SIOF> = "0").

Note 2: The number of data bytes to transfer is used for transmit and receive operations in common.

Note 3: Always write "0" to bits 7 to 5.

Serial Interface Status Register

SIOSR (001BH)

7	6	5	4	3	2	1	0
SIOF	SEF	TXF	RXF	TXERR	RXERR		

(Initial value: 0010 00**)

SIOF	Monitor the operating status of serial transfer.	0: Transfer ended (Note1) 1: Transfer in process	Read only
SEF	Shift operation status flag	0: Shift ended 1: Shift in process	
TXF	Transmit buffer flag	0: The transmit buffer contains data. 1: The transmit buffer contains no data.	
RXF	Receive buffer flag	0: The receive buffer contains no data. 1: As many data bytes specified in SIORXD have been received. (The flag is reset to "0" when as many data bytes as specified in SIORXD have been read.)	
TXERR	Transmit error flag (Note2)	0: Transmit operation was normal. 1: Error occurred during transmission.	
RXERR	Receive error flag (Note2)	0: Receive operation was normal. 1: Error occurred during reception.	

Note 1: The SIOSR<SIOF> bit is cleared to "0" by clearing SIOCR1<SIOS> to stop transferring or by setting SIOCR1<SIOINH> to "1" to abort transfer.

Note 2: Neither the SIOSR<TXERR> nor SIOSR<RXERR> bit can be cleared when transfer ends on SIOCR1<SIOS> = "0". To clear them, set SIOCR1<SIOINH> to "1"

Note 3: Do not write to the SIOSR register.

Serial Interface Data Buffer

SIOBUF (001CH)

7	6	5	4	3	2	1	0

(Initial value: **** ****)

SIOBUF	Transmit/receive data buffer	Transmit data are set, or received data are stored.	R/W

Note 1: Setting SIOCR1<SIOINH> causes the contents of SIOBUF to be lost.

Note 2: When setting transmit data or storing received data, be sure to handle as many bytes as specified in SIOCR2<SIORXD> at a time.

12.3 Function

12.3.1 Serial clock

12.3.1.1 Clock source

One of the following clocks can be selected using SIOCR1<SCK>.

(1) Internal clock

A clock having the frequency selected with SIOCR1<SCK> (except for “111”) is used as the serial clock. The $\overline{SCK}$ pin output goes high when transfer starts or ends.

Table 12-2 Serial Clock Rate

SCK	Clock	Baud Rate	
		fc = 16 MHz	fc = 8 MHz
000	$fc/2^{13}$	1.91 Kbps	0.95 Kbps
001	$fc/2^{8}$	61.04 Kbps	30.51 Kbps
010	$fc/2^{6}$	244.14 Kbps	122.07 Kbps
011	$fc/2^{5}$	488.28 Kbps	244.14 Kbps
100	$fc/2^{4}$	976.56 Kbps	488.28 Kbps
101	$fc/2^{3}$	–	976.56 Kbps
110	$fc/2^{2}$	–	–
111	External	External	External

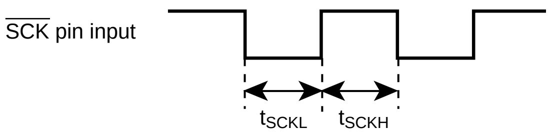
(1 Kbit = 1,024 bit)

Note: Do not make such a setting that the serial clock rate will exceed 1 Mbps.

(2) External clock

Setting SIOCR1<SCK> to “111” causes an external clock to be selected. A clock supplied to the $\overline{SCK}$ pin is used as the serial clock.

For a shift operation to be performed securely, both the high and low levels of the serial clock pulse must be at least 4/fc. If fc = 8 MHz, therefore, the maximum available transfer rate is 976.56 Kbps.



t_{SCKL}, t_{SCKH} ≥4/fc (High-frequency clock mode)
t_{SCKL}, t_{SCKH} ≥4/fs (Low-frequency clock mode)

Figure 12-2 External Clock

12.3.1.2 Shift edges

The SIO uses leading-edge shift for transmission and trailing-edge shift for reception.

(1) Leading-edge shift

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

(2) Trailing-edge shift

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

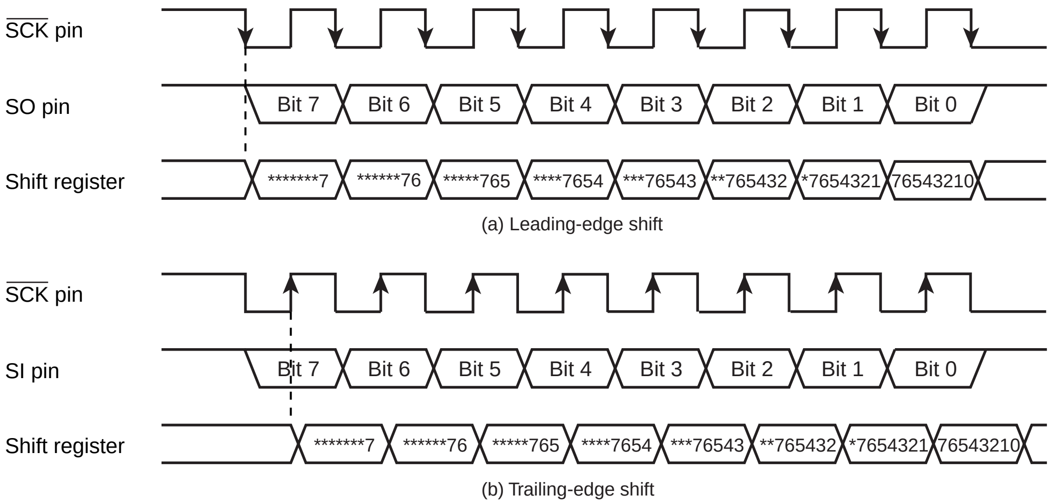


Figure 12-3 Shift Edges

12.3.2 Transfer bit direction

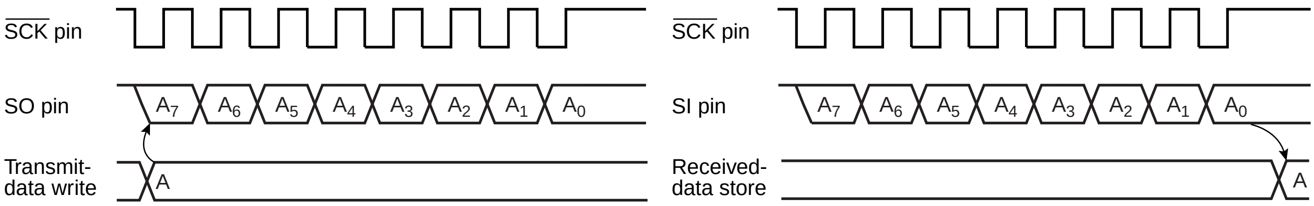
The direction in which 8-bit serial data are transferred can be selected using SIOCR1<SIODIR>. The direction of data transfer applies in common to both transmission and reception, and cannot be set individually.

12.3.2.1 MSB transfer

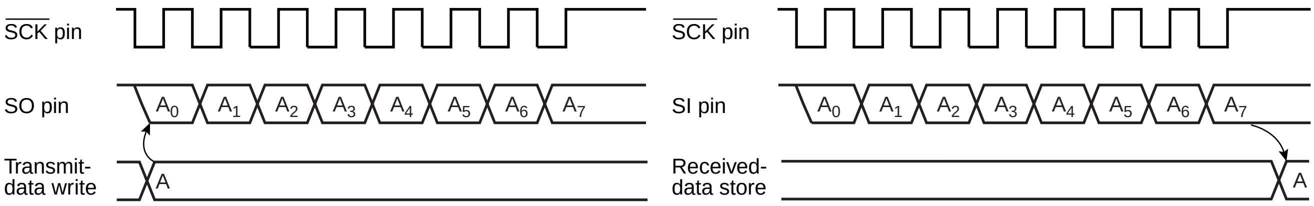
MSB transfer is assumed by clearing SIOCR1<SIODIR> to “0”. In MSB transfer, data are transferred sequentially beginning with the most significant bit (MSB). As for received data, the first data bit to receive is stored as the MSB.

12.3.2.2 LSB transfer

LSB transfer is assumed by setting SIOCR1<SIODIR> to “1”. In LSB transfer, data are transferred sequentially beginning with the least significant bit (LSB). As for received data, the first data bit to receive is stored as the LSB.



(a) MSB transfer (when SIODIR = "0")



(b) LSB transfer (when SIODIR = "1")

Figure 12-4 Transfer Bit Direction

12.3.3 Transfer modes

SIOCR1<SIOM> is used to select a transfer mode (transmit, receive, or transmit/receive mode).

12.3.3.1 Transmit mode

Transmit mode is assumed by setting SIOCR1<SIOM> to “00”.

(1) Causing the SIO to start transmitting

1. Set the transmit mode, serial clock rate, and transfer direction, respectively, in SIOCR1<SIOM>, SIOCR1<SCK>, and SIOCR1<SIODIR>.
2. Set the number of data bytes to transfer in SIOCR2<SIORXD>.
3. Set, in SIOBUF, as many transmit data bytes as specified in SIOCR2<SIORXD>.

4. SIOCR1<SIOS> to “1”.

If the selected serial clock is an internal clock, the SIO immediately starts transmitting data sequentially in the direction selected using SIOCR1<SIODIR>.

If the selected serial clock is an external clock, the SIO immediately starts transmitting data, upon external clock input, sequentially in the direction selected using SIOCR1<SIODIR>.

(2) Causing the SIO to stop transferring

1. When as many data bytes as specified in SIOCR2<SIORXD> have been transmitted, be sure to clear SIOCR1<SIOS> to “0” to halt the SIO. Clearing of SIOCR1<SIOS> should be executed within the INTSIO service routine or should be executed after confirmation of SIOSR<TXF> = “1”. Before starting to transfer the next data, make sure SIOSR<SIOF> = “0” and SIOSR<TXERR>= “0”, write the data to be transferred, and then set SIOCR1<SIOS> = “1”.

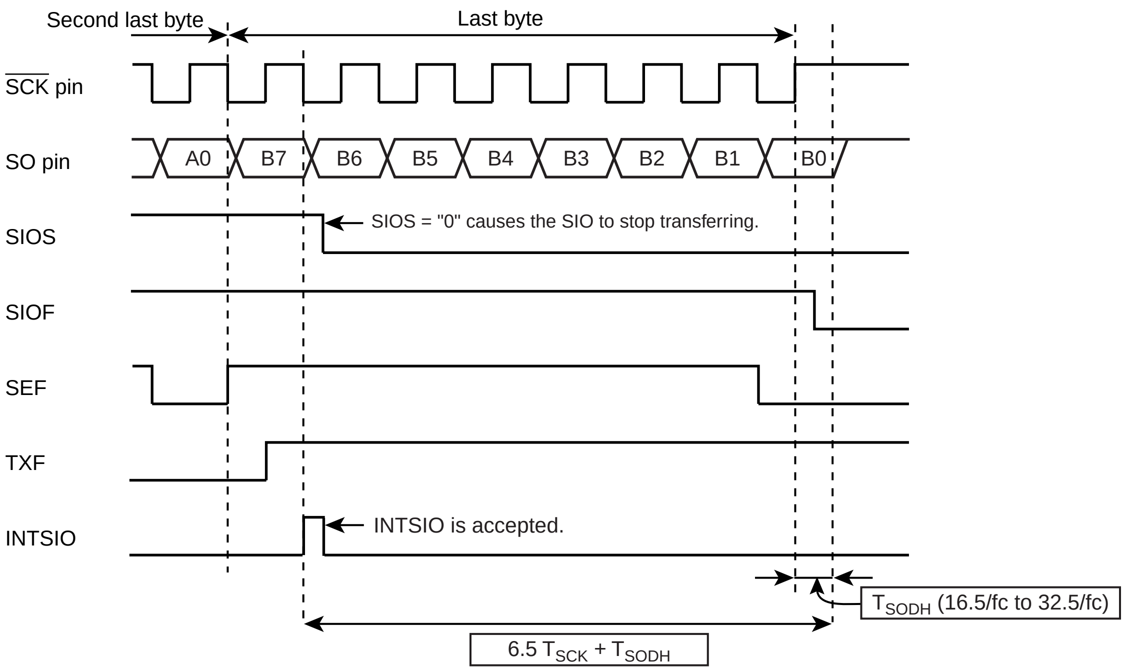


Figure 12-5 Time from INTSIO Occurrence to Transfer End (SIOSR<SIOF> = “0”) when the SIO is Directed to Stop Transferring (SIOCR1<SIOS> = “0”) upon the Occurrence of a Transmit Interrupt

Note 1: Be sure to write as many bytes as specified in SIOCR2<SIORXD> to SIOBUF. If the number of data bytes to be written to SIOBUF is not equal to the value specified in SIOCR2<SIORXD>, the SIO fails to work normally.

Note 2: Before starting the SIO, be sure to write as many data bytes as specified in SIOCR2<SIORXD> to SIOBUF.

Note 3: In the transmit mode, an INTSIO interrupt occurs when the transmission of the second bit of the last byte begins.

Note 4: If an attempt is made to write SIOCR1<SIOS> = “0” within the INTSIO interrupt service routine, the SIO stops transferring (SIOSR<SIOF> = “0”) after the last data byte is transmitted (the signal at the SCK pin rises).

Note 5: Be sure to write to SIOBUF in the condition SIO stop status (SIOSR<SIOF>="0"). If write to SIOBUF during SIO working status (SIOSR<SIOF>="1"), the SIO fails to work normally.

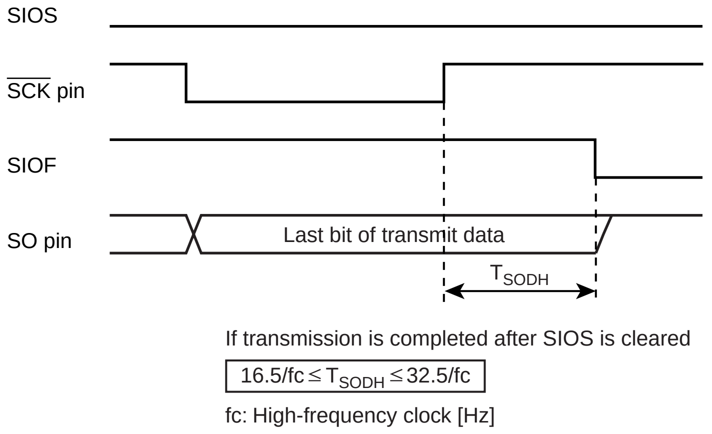


Figure 12-6 Last-Bit Hold Time

- Setting SIOCR1<SIOINH> to “1” causes the SIO to immediately stop a transmission sequence even if any byte is being transmitted.

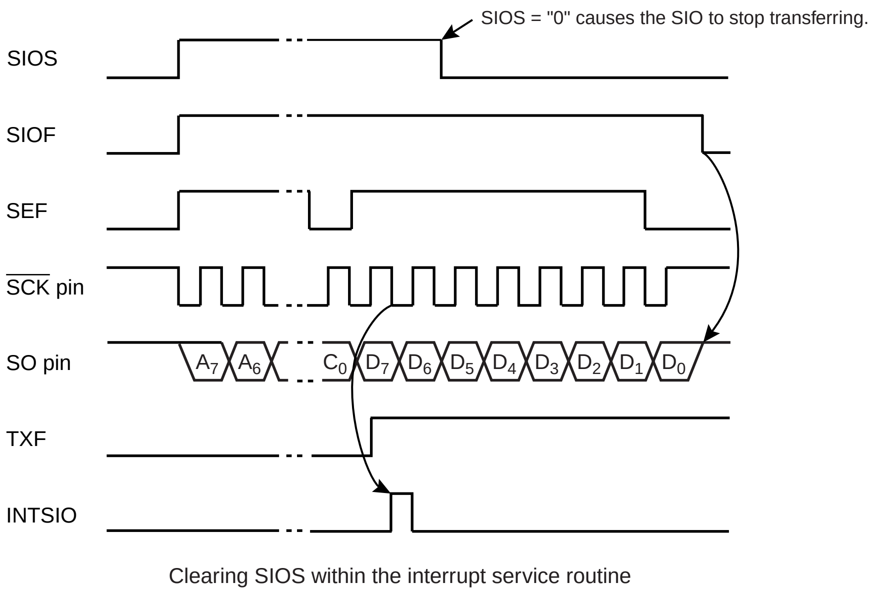


Figure 12-7 SIOCR1<SIOS> Clear Timing

12.3.3.2 Transmit error

During operation on an external clock, the following case may be detected as a transmit error, causing the transmit error flag (SIOSR<TXERR>) to be set to “1”. If a transmit error occurs, the SO pin goes high.

- If the $\overline{\text{SCK}}$ pin goes low when the SIO is running (SIOSR<SIOF> = “1”) but there is no transmit data in SIOBUF (SIOSR<TXF> = “1”).

If a transmit error is detected, be sure to set SIOCR1<SIOINH> to “1” to force the SIO to halt. Setting SIOCR1<SIOINH> to “1” initializes the SIOCR1<SIOS> and SIOSR registers; no other registers or bits are initialized.

Example :Example of setting the transmit mode (transmit mode, external clock, and 32-byte transfer)

```
                          Port setting
              DI                                   ; IMF ← 0
              LDW         (EIRL), ******1********0B ; Enables INTSIO (EF9).
              EI                                   ; Enables interrupts.
              LD          (SIOCR1), 01******B      ; Initializes the SIO (forces the SIO halt).
WAIT:         TEST        (SIOSR). 7               ; Checks to see if the SIO has halted (SIOF = 0).
              JRS         F, WAIT                  ; Jumps to START if the SIO is already at a halt.
START:        LD          (SIOCR1), 00000111B      ; Sets the transmit mode, selects the direction of transfer,
                                                     and sets a serial clock.
              LD          (SIOCR2), 00011111B      ; Sets the number of bytes (32 bytes) to transfer.
                            :
                          Transmit data setting
                            :
              LD          (SIOCR1), 10000111B      ; Directs the SIO to start transferring.
INTSIO
(INTSIO ser-
vice routine):
              LD          (SIOCR1), 00000111B      ; Directs the SIO to stop transferring.
              TEST        (SIOSR). 3               ; Checks TXERR.
              JRS         T, NOERR
              LD          (SIOCR1), 01000111B      ; Forces the SIO to halt (clears TXERR).
                            :
                          Error handling
                            :
NOERR:
END:                                               ; End of transfer
```

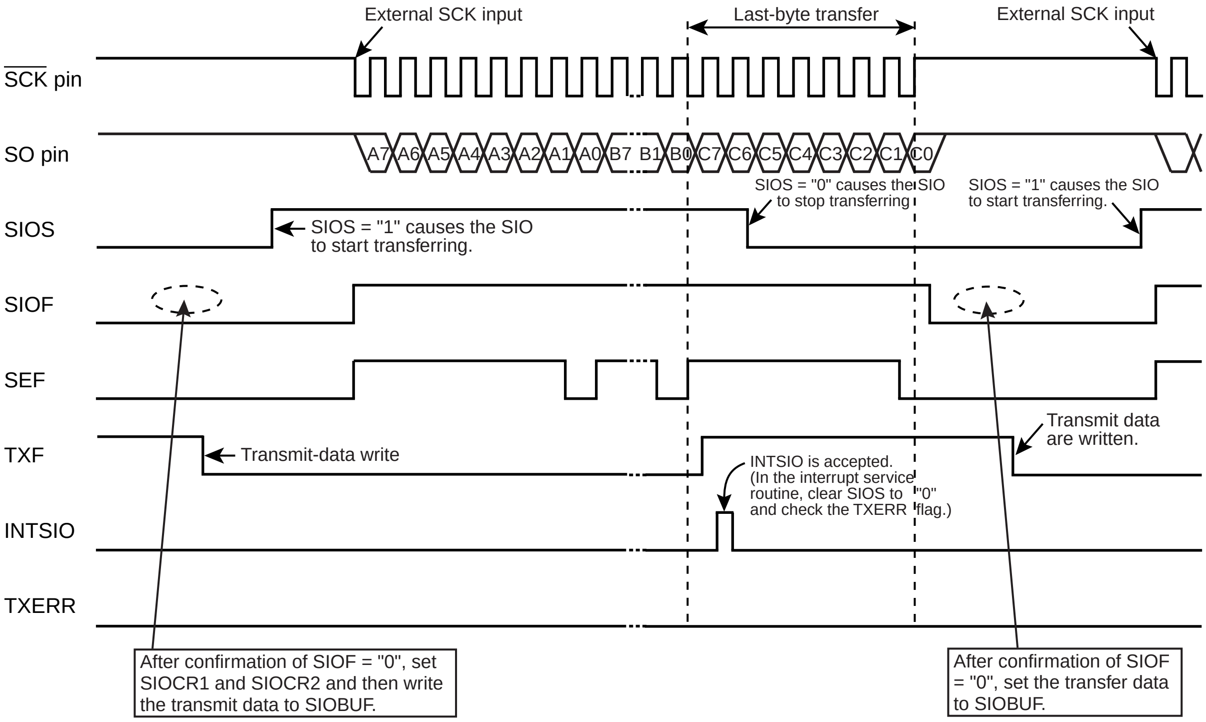


Figure 12-8 Transmit Mode Operation
(where 3 bytes are transferred on an external source clock)

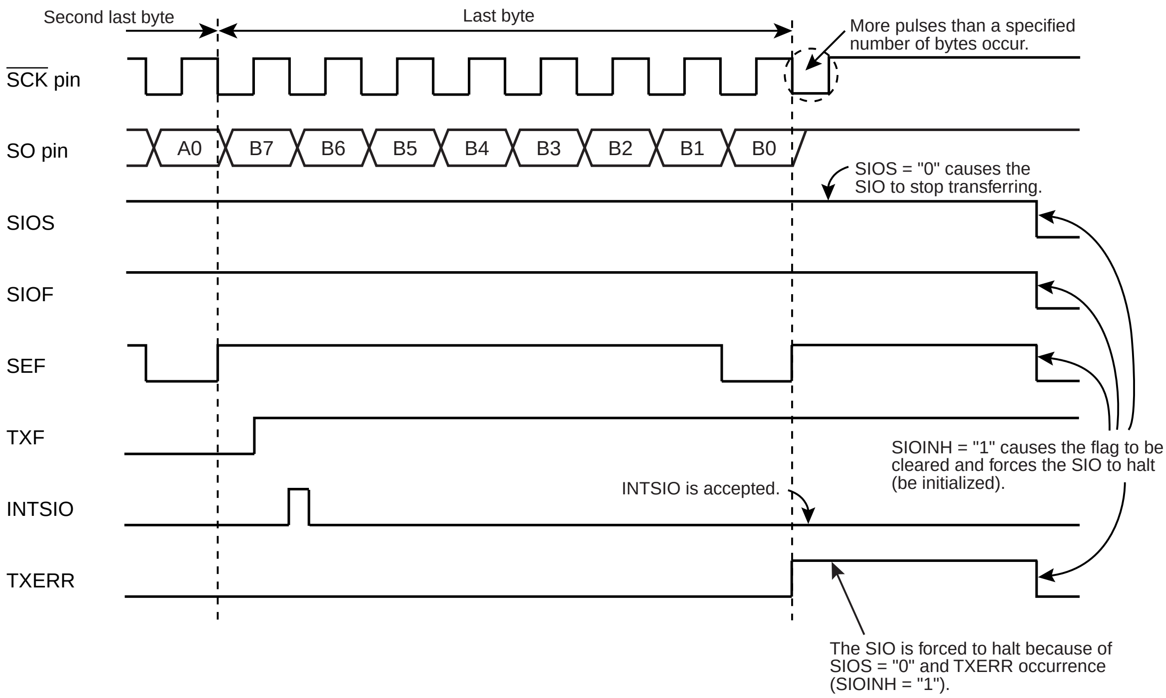


Figure 12-9 Occurrence of Transmit Error (where, before the SIO is directed to stop transferring (SIOCR1<SIOS> = “0” is written), the transfer of the last byte is completed and more pulses than a specified number of bytes occur)

Note: When the SIO is running (SIOSR<SIOF> = “1”), do not supply more transfer clock pulses than the number of bytes specified in SIOCR2<SIORXD> to the SCK pin.

12.3.3.3 Receive mode

Receive mode is assumed by setting SIOCR1<SIOM> to “01”.

(1) Causing the SIO to start receiving

1. Set the receive mode, serial clock rate, and transfer direction, respectively, in SIOCR1<SIOM>, SIOCR1<SCK>, and SIOCR1<SIODIR>.
2. Set the number of data bytes to transfer in SIOCR2<SIORXD>.
3. Set SIOCR1<SIOS> to “1”.

   If the selected serial clock is an internal clock, the SIO immediately starts receiving data sequentially in the direction selected using SIOCR1<SIODIR>.

   If the selected serial clock is an external clock, the SIO immediately starts receiving data, upon external clock input, sequentially in the direction selected using SIOCR1<SIODIR>.

(2) Causing the SIO to stop receiving

1. When as many data bytes as specified in SIOCR2<SIORXD> have been received, be sure to clear SIOCR1<SIOS> to “0” to halt the SIO. Clearing of SIOCR1<SIOS> should be executed within the INTSIO service routine or should be executed after confirmation of SIOSR<RXF> = “1”.

   Setting SIOCR1<SIOINH> to “1” causes the SIO to immediately stop a reception sequence even if any byte is being received.

(3) Received-data read timing

Before reading received data, be sure to make sure SIOBUF is full (SIOSR<RXF> = “1”) or clear SIOCR1<SIOS> to “0” to halt the SIO in the INTSIO interrupt service routine.

To read the received data after SIOCR1<SIOS> to “0”, make sure SIOSR<SIOF> = “0” and SIOSR<RXERR> = “0”. SIOSR<RXF> is cleared to “0” when as many received data bytes as specified in SIOCR2<SIORXD> are read.

To transfer the next data after SIOCR1<SIOS> to “0”, first read the received data, make sure SIOSR<SIOF> = “0”, and set SIOCR1<SIOS> = “1” to start receiving data.

Note 1: Be sure to read, from SIOBUF, as many received data bytes as specified in SIOCR2<SIORXD>. If the number of data bytes to be read from SIOBUF is not equal to the value specified in SIOCR2<SIORXD>, the SIO fails to work normally.

Note 2: If an attempt is made to read data before the end of reception (SIOSR<RXF> = “0”), the SIO fails to work normally.

Note 3: In the receive mode, an INTSIO interrupt occurs when the reception of the last bit of the last data byte is completed.

Note 4: If an attempt is made to start transferring after a receive error has been detected, the SIO fails to work normally. Before starting transferring, set SIOCR1<SIOINH> = “1” to force the SIO to halt.

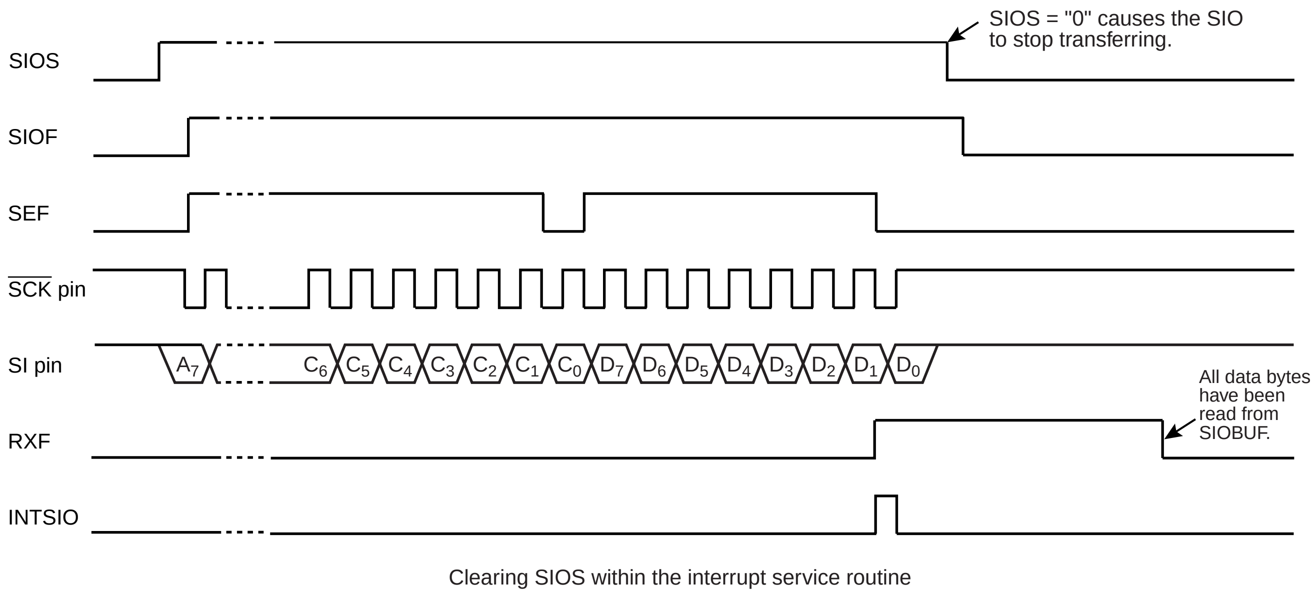


Figure 12-10 SIOCR1<SIOS> Clear Timing

12.3.3.4 Receive error

During operation on an external clock, the following case is detected as a receive error, causing the receive error flag (SIOSR<RXERR>) to be set to "1". If a receive error occurs, discard all data from the receive buffer.

- If the reception of the next data byte ends with SIOBUF full (SIOSR<RXF> = "1") (if eight clock pulses are supplied to the $\overline{SCK}$ pin)

If a receive error is detected, be sure to set SIOCR1<SIOINH> to "1" to force the SIO to halt. Setting SIOCR1<SIOINH> to "1" initializes the SIOCR1<SIOS> and SIOSR registers; no other registers or bits are initialized.

Note: When the SIO is running on an external clock, it becomes impossible to read the content of the receive data buffer (SIOBUF) correctly if the $\overline{SCK}$ pin goes low before as many data bytes as specified in SIOCR2<SIORXD> are read.
A receive error flag (SIOSR<RXF>) can be set only after eight clock pulses are input upon completion of reception. If only one to seven transfer clock pulses (including noise) are input to the $\overline{SCK}$ pin, therefore, it becomes impossible to determine whether the pulses at the pin are those unnecessary. So, it is recommended that the system employ a backup method such as checksum-based verification.
Before restarting reception, be sure to force the SIO to halt (SIOCR1<SIOINH> = "1").

Example :Example of setting the receive mode (receive mode, external clock, and 32-byte transfer)

		Port setting	
	DI		; IMF ← 0
	LDW	(EIRL), ******1********0B	; Enables INTSIO (EF9)
	EI		; Enables interrupts.
	LD	(SIOCR1), 01******B	; Initializes the SIO (Forces the SIO halt).
WAIT:	TEST	(SIOSR). 7	; Checks to see if the SIO has halted (SIOF = 0).
	JRS	F, WAIT	; Jumps to START if the SIO is already at a halt.
START:			
	LD	(SIOCR1), 00010111B	; Sets the receive mode, selects the direction of transfer, and sets a serial clock.
	LD	(SIOCR2), 00011111B	; Sets the number of bytes to transfer.
	LD	(SIOCR1), 10010111B	; Directs the SIO to start transferring.
INTSIO (INTSIO service routine):			
	LD	(SIOCR1), 00010111B	; Directs the SIO to stop transferring.
		:	
		Receive data reading Checks a checksum or the like to see if the received data are normal.	
		:	
	LD	(SIOCR1), 01010111B	; Forces the SIO to halt.
END:			; End of transfer

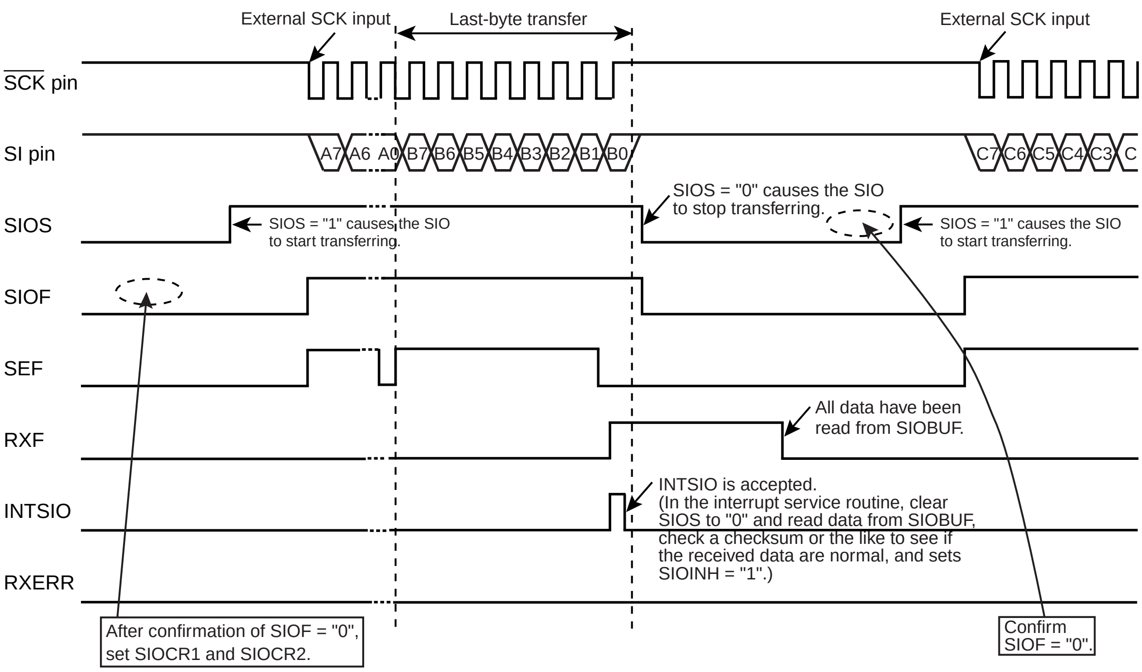


Figure 12-11 Receive Mode Operation
(where 2 bytes are transferred on an external source clock)

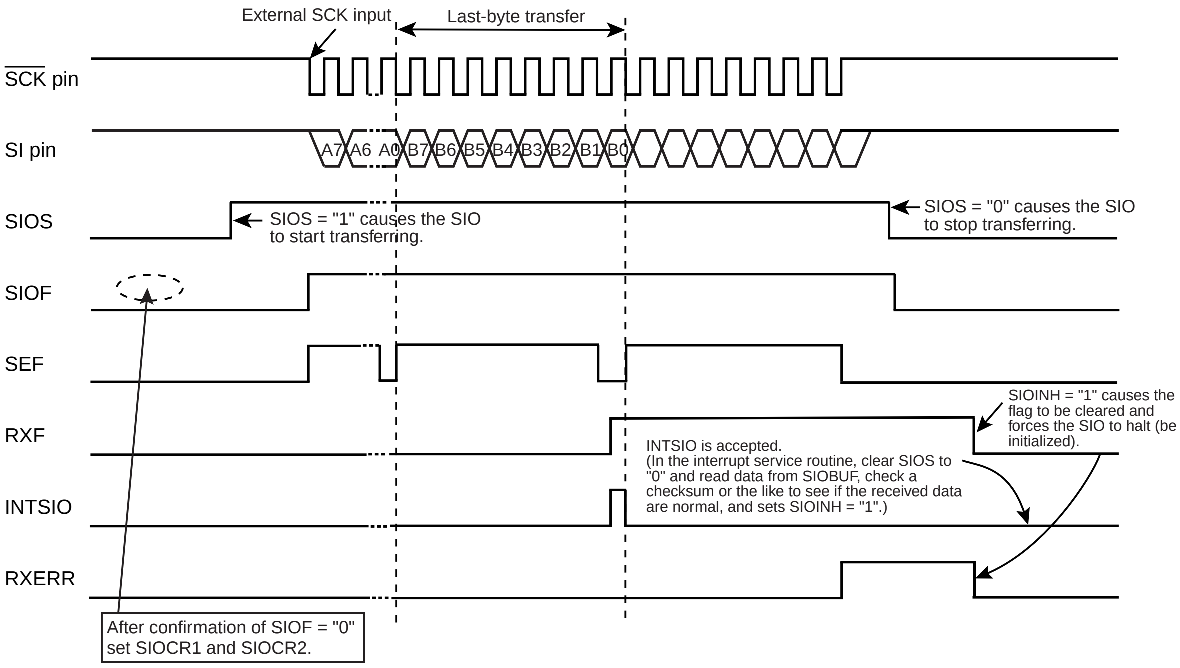


Figure 12-12 Occurrence of Receive Error
(2 bytes are transferred on an external source clock)

Note 1: When the SIO is running (SIOSR<SIOF> = “1”), do not supply more transfer clock pulses than the number of bytes specified in SIOCR2<SIORXD> at $\overline{\text{SCK}}$ pin.

Note 2: After data reception is completed, a receive error occurs if eight clock pulses are supplied to the $\overline{\text{SCK}}$ pin before a direction to stop the SIO becomes valid (SIOCR1<SIOS> = “0”). Figure 12-8 shows a case in which a receive error occurs when eight clock pulses are supplied to the $\overline{\text{SCK}}$ pin before the INTSIO interrupt service routine writes SIOCR1<SIOS> = “0”.

12.3.3.5 Transmit/receive mode

Transmit/receive mode is assumed by setting SIOCR1<SIOM> to “10”.

(1) Causing the SIO to start transmitting/receiving

1. Set the transmit/receive mode, serial clock rate, and transfer direction, respectively, in SIOCR1<SIOM>, SIOCR1<SCK>, and SIOCR1<SIODIR>.
2. Set the number of data bytes to transfer in SIOCR2<SIORXD>.
3. Set, in SIOBUF, as many transmit data bytes as specified in SIOCR2<SIORXD>.
4. Set SIOCR1<SIOS> to “1”.

If the selected serial clock is an internal clock, the SIO immediately starts transmitting/receiving data sequentially in the direction selected using SIOCR1<SIODIR>.

If the selected serial clock is an external clock, the SIO starts transmitting/receiving data, in synchronization with a clock input to the $\overline{\text{SCK}}$ pin sequentially in the direction selected using SIOCR1<SIODIR>.

Note 1: SIOCR2<SIORXD>, SIOCR1<SIODIR>, and SIOCR1<SCK> are used in common to both transmission and reception. They cannot be set individually.

Note 2: Transmit data are output in synchronization with the falling edge of a signal at the $\overline{\text{SCK}}$ pin. The data are received in synchronization with the rising edge of a signal at the $\overline{\text{SCK}}$ pin.

(2) Causing the SIO to stop transmitting/receiving

1. When as many data bytes as specified in SIOCR2<SIORXD> have been transmitted and received, be sure to clear SIOCR1<SIOS> to “0” to halt the SIO. Clearing of SIOCR1<SIOS> should be executed within the INTSIO service routine or should be executed after confirmation of SIOSR<RXF> = “1”.

   Setting SIOCR1<SIOINH> to “1” causes the SIO to immediately stop the transmission/reception sequence even if any byte is being transmitted or received.

(3) Received-data read and transmit-data set timing

After as many bytes as specified in SIOCR2<SIORXD> have been transmitted and received, reading the received data and writing the next transmit data should be executed after confirmation of SIOSR<RXF> = “1” or should be executed after SIOCR1<SIOS> is cleared to “0” in the INTSIO interrupt service routine. To re-start transferring the next data after SIOCR1<SIOS> to “0”, first make sure SIOSR<SIOF> = “0”, SIOSR<TXERR> = “0” and SIOSR<RXERR> = “0”, and read the received data, and then write the transmit data and set SIOCR1<SIOS> = “1” to start transferring.

Note 1: An INTSIO interrupt occurs when the last bit of the last data byte is received.

Note 2: When writing to and reading from SIOBUF, make sure that the number of data bytes to transfer is as specified in SIOCR2<SIORXD>. If the number is not equal to the value specified in SIOCR2<SIORXD>, the SIO does not run normally.

Note 3: When as many data bytes as specified in SIOCR2<SIORXD> are read, SIOSR<RXF> is cleared to “0”.

Note 4: In the transmit/receive mode, setting SIOCR1<SIOINH> to “1” to force the SIO to halt will cause received data to be discarded.

Note 5: If a transfer sequence is started after a transmit or receive error has been detected, the SIO does not run normally. Before starting transferring, set SIOCR1<SIOINH> = “1” to force the SIO to halt.

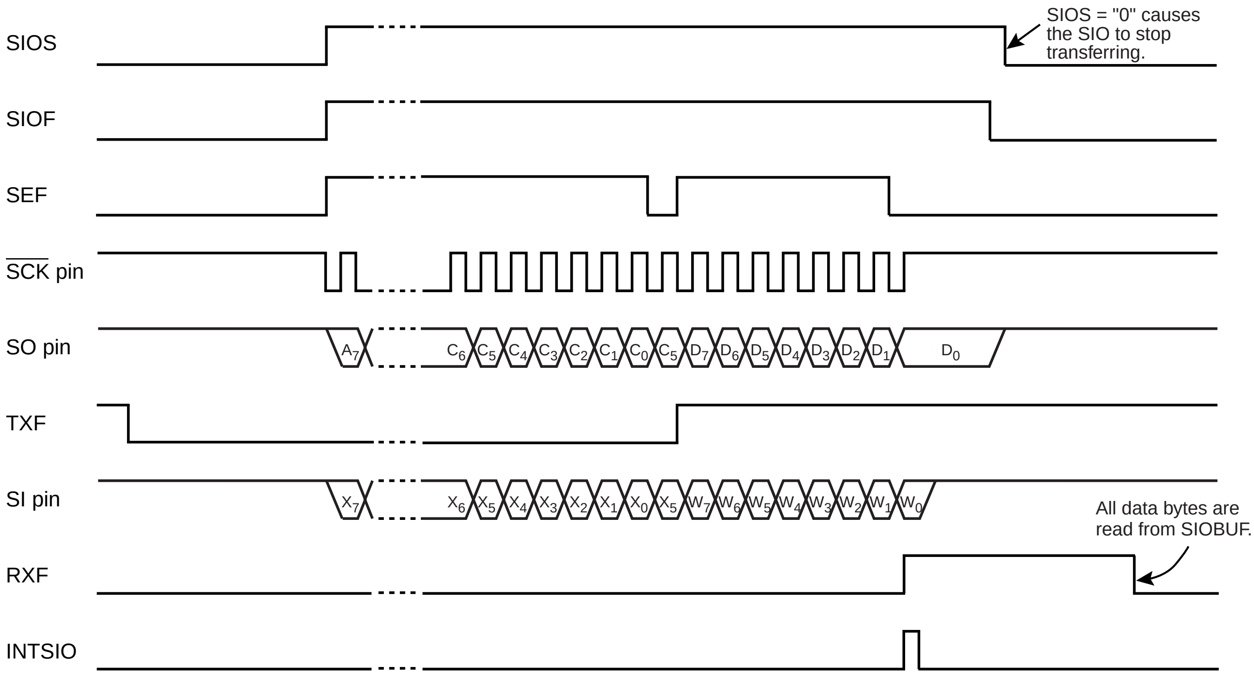


Figure 12-13 SIOCR1<SIOS> Clear Timing (Transmit/Receive Mode)

12.3.3.6 Transmit/receive error

During operation on an external clock, the following cases may be detected as a transmit or receive error, causing an error flag (SIOSR<TXERR> or SIOSR<RXERR>) to be set. If an error occurs, the transmit data go high.

- If the $\overline{SCK}$ pin goes low when the SIO is running (SIOSR<SIOF> = “1”) but there is no transmit data in SIOBUF (SIOSR<TXF> = “1”).
- If the reception of the next data byte is completed when the SIO is running (SIOSR<SIOF> = “1”) and SIOBUF is full (SIOSR<RXF> = “1”) (if eight clock pulses are supplied to the $\overline{SCK}$ pin) (SIOSR<RXERR>)

If a transmit or receive error is detected, be sure to set SIOCR1<SIOINH> to “1” to force the SIO to halt.

Note: When the SIO is running on an external clock, it becomes impossible to read the content of the receive data buffer (SIOBUF) correctly if the $\overline{SCK}$ pin goes low before as many data bytes as specified in SIOCR2<SIORXD> are read.
A receive error flag (SIOSR<RXF>) can be set only after eight clock pulses are input upon completion of reception. If one to seven transfer clock pulses (including noise) are input to the $\overline{SCK}$ pin, therefore, it becomes impossible to determine whether the pulses at the pin are those unnecessary. So, it is recommended that the system employ a backup method such as checksum-based verification.
Before restarting transmitting/receiving, be sure to force the SIO to halt (SIOCR1<SIOINH> = “1”).

Example :Example of setting the transmit/receive mode
(transmit/receive mode, external clock, and 32-byte transfer)

```
                    Port setting
          DI                                   ; IMF ← 0
          LDW       (EIRL), ******1********0B  ; Enables INTSIO (EF9)
          EI                                   ; Enables interrupts.
          LD        (SIOCR1), 01******B        ; Initializes the SIO (forces the SIO halt).
WAIT:     TEST      (SIOSR). 7                 ; Checks to see if the SIO has halted (SIOF = 0).
          JRS       F, WAIT                    ; Jumps to START if the SIO is already at a halt.
```

Example :Example of setting the transmit/receive mode
(transmit/receive mode, external clock, and 32-byte transfer)

```
START:
            LD      (SIOCR1), 00100111B    ; Sets the transmit/receive mode, selects the direction of transfer,
                                             and sets a serial clock.
            LD      (SIOCR2), 00011111B    ; Sets the number of bytes (32 bytes) to transfer.
Transmit data set-
ting:                                      ;
            LD      (SIOCR1), 10100111B    ; Starts transferring.
INTSIO (INTSIO
service routine):
            LD      (SIOCR1), 00100111B    ; Directs the SIO to stop transferring.
            TEST    (SIOSR). 3             ; Checks TXERR.
            JRS     T, TXNOERR
            LD      (SIOCR1), 01100111B    ; Forces the SIO to halt (clears TXERR).
                      :
                    Error handling
                      :
            JR      END
TXNOER:
                      :
                    Receive-data reading
                    Checks a checksum or the
                    like to see if the received
                    data are correct.
                      :
            LD      (SIOCR1), 01100111B    ; Forces the SIO to halt.
END:                                       ; End of transfer
```

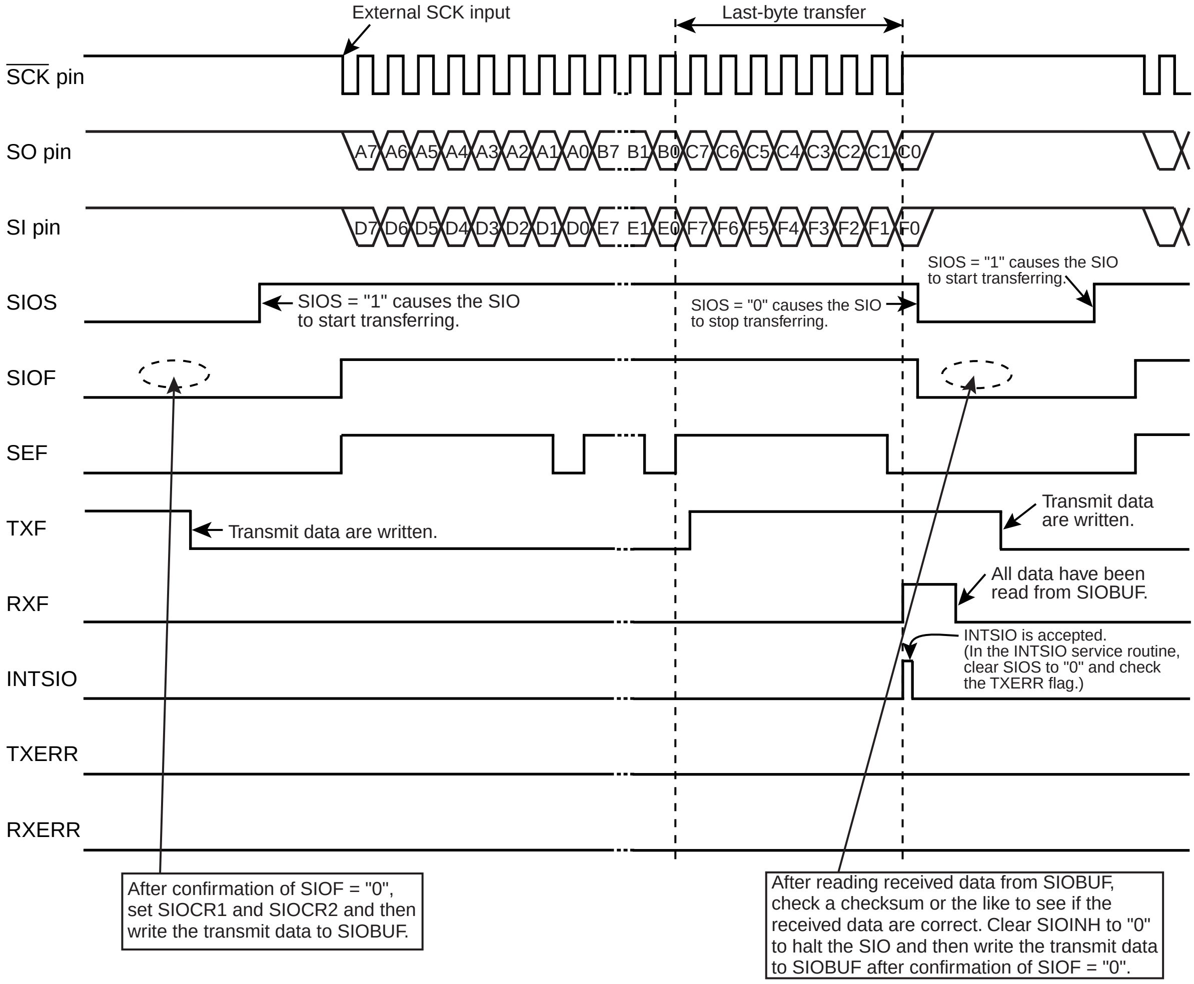


Figure 12-14 Transmit/Receive Mode Operation
(where 3 bytes are transferred on an external source clock)

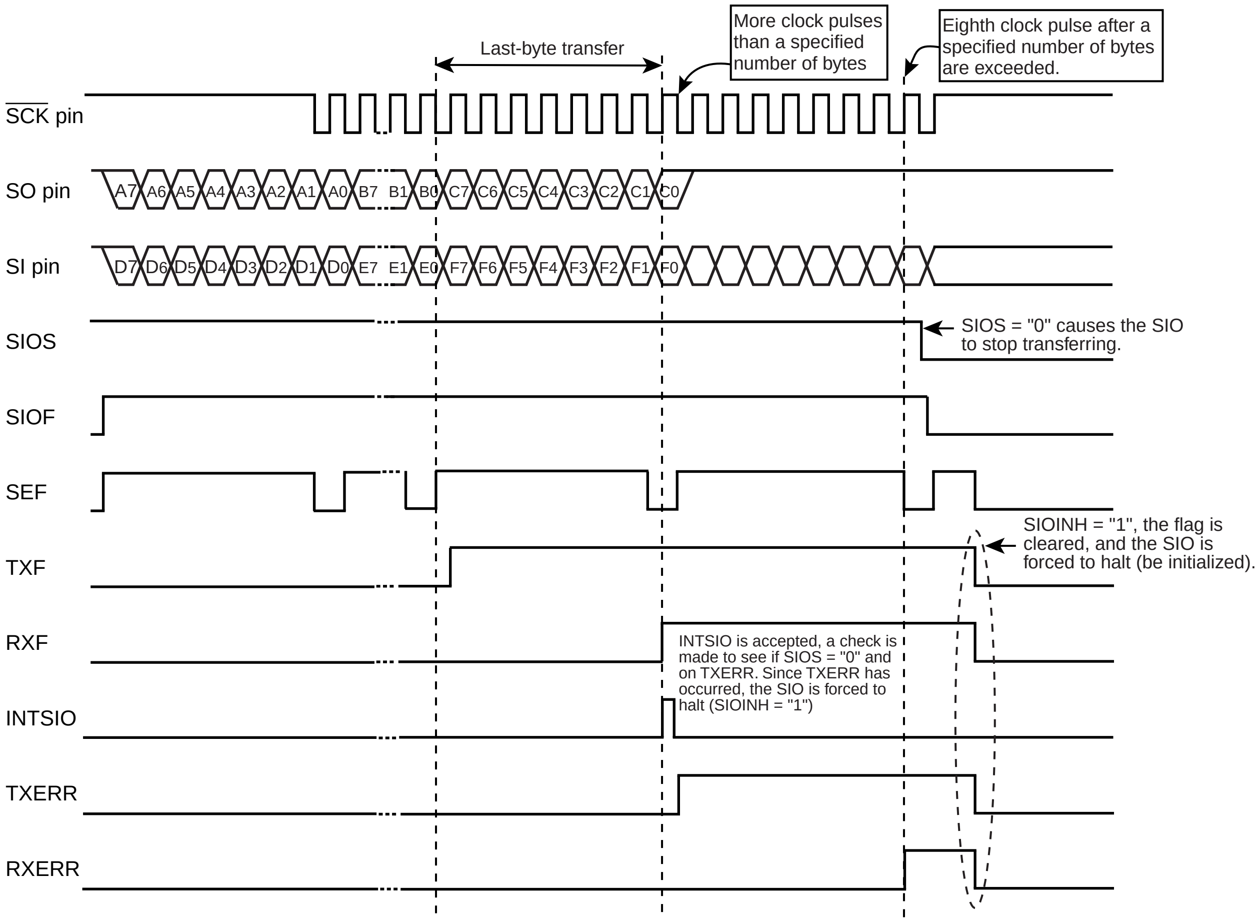


Figure 12-15 Occurrence of Transmit/Receive Error
(3 bytes are transferred on an external source clock)

Note: When the SIO is running (SIOSR<SIOF> = “1”), do not supply more transfer clock pulses than the number of bytes specified in SIOCR2<SIORXD> to the SCK pin.

13. 8-Bit AD Converter (ADC)

The TMP86CK74AFG have a 8-bit successive approximation type AD converter.

13.1 Configuration

The circuit configuration of the 8-bit AD converter is shown in Figure 13-1.

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

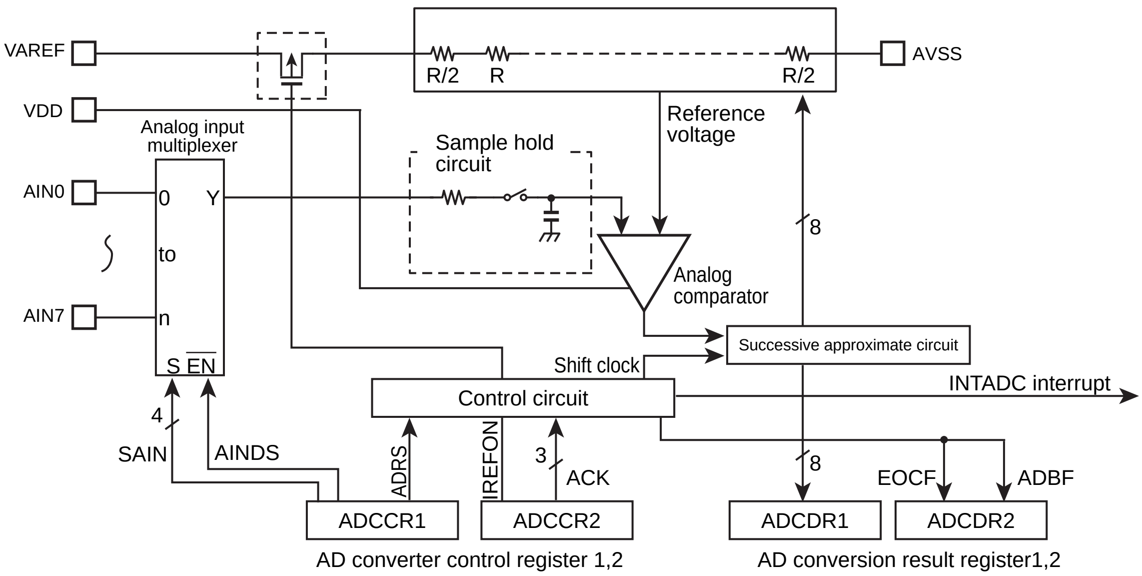


Figure 13-1 8-bit AD Converter (ADC)

13.2 Control

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCR1)

   This register selects the analog channels 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 (ADCDR1)

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

4. AD converted value register (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	"0"	"1"	AINDS	SAIN				(Initial value: 0001 0000)

ADRS	AD conversion start	0: – 1: Start	R/W
AINDS	Analog input control	0: Analog input enable 1: Analog input disable	R/W
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	R/W

Note 1: Select analog input when AD converter stops (ADCDR2<ADBF> = “0”).

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

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

Note 4: The 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 register 1 (ADCCR1) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCR1 newly after returning to NORMAL1 or NORMAL2 mode.

Note 7: Always set bit 5 in ADCCR1 to “1” and set bit 6 in ADCCR1 to “0”.

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	000: 39/fc 001: Reserved 010: 78/fc 011: 156/fc 100: 312/fc 101: 624/fc 110: 1248/fc 111: Reserved	R/W

Note 1: Always set bit 0 in ADCCR2 to "0" and set bit 4 in ADCCR2 to "1".

Note 2: When a read instruction for ADCCR2, bit 6 to 7 in ADCCR2 read in as undefined data.

Note 3: After STOP or SLOW/SLEEP mode are started, AD converter control register 2 (ADCCR2) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCR2 newly after returning to NORMAL1 or NORMAL2 mode.

Table 13-1 Conversion Time according to ACK Setting and Frequency

Condition / ACK	Conbersion time'	16MHz	8MHz	4 MHz	2 MHz	10MHz	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: Settings for "–" in the above table are inhibited.

Note 2: Set conversion time by Analog Reference Voltage (V_{AREF}) as follows.

- V_{AREF} = 4.5 to 5.5 V (15.6 μs or more)
- V_{AREF} = 2.7 to 5.5 V (31.2 μs or more)

AD Conversion Result Register

ADCDR1 (0027H)	7	6	5	4	3	2	1	0	
	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00	(Initial value: 0000 0000)

AD Conversion Result Register

ADCDR2 (0026H)	7	6	5	4	3	2	1	0	
			EOCF	ADBF					(Initial value: **00 ****)

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

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

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

Note 3: If a read instruction is executed for ADCDR2, read data of bits 7, 6 and 3 to 0 are unstable.

13.3 Function

13.3.1 AD Conveter Operation

When ADCCR1<ADRS> is set 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) and at the same time ADCDR2<EOCF> is set to “1”, the AD conversion finished interrupt (INTADC) is generated.

ADCCR1<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 ADCDR<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

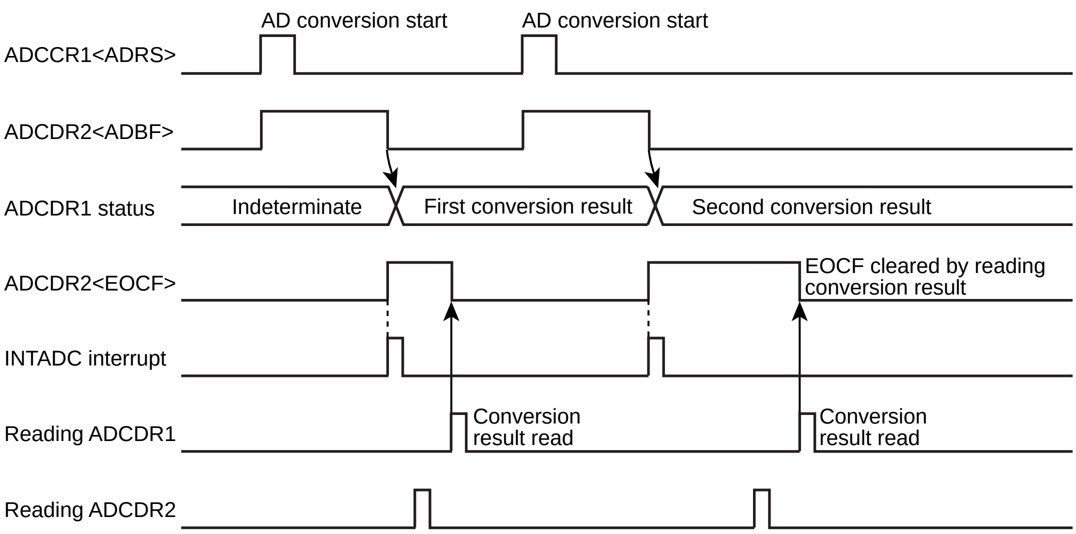


Figure 13-2 AD Converter Operation

13.3.2 AD Converter Operation

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).
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 Table 13-1.
   - 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”.
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 of 19.5 μs at 16 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, read the converted value and store the 8-bit data in address 009FH on RAM.

```
        ; AIN SELECT
        :       :                    ; Before setting the AD converter register, set each port reg-
        :       :                    ister suitably (For detail, see chapter of I/O port.)
        LD      (ADCCR1), 00100011B  ; Select AIN3
        LD      (ADCCR2), 11011000B  ; Select conversion time (312/fc) and operation mode
        ; AD CONVERT START
        SET     (ADCCR1). 7          ; ADRS = 1
SLOOP:  TEST    (ADCDR2). 5          ; EOCF = 1 ?
        JRS     T, SLOOP
        ; RESULT DATA READ
        LD      A, (ADCDR1)
        LD      (9FH), A
```

13.3.3 STOP and SLOW Mode during AD Conversion

When the 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 STOP or SLOW mode.) When restored from 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.

13.3.4 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 8-bit digital value converted by the AD as shown in Figure 13-3.

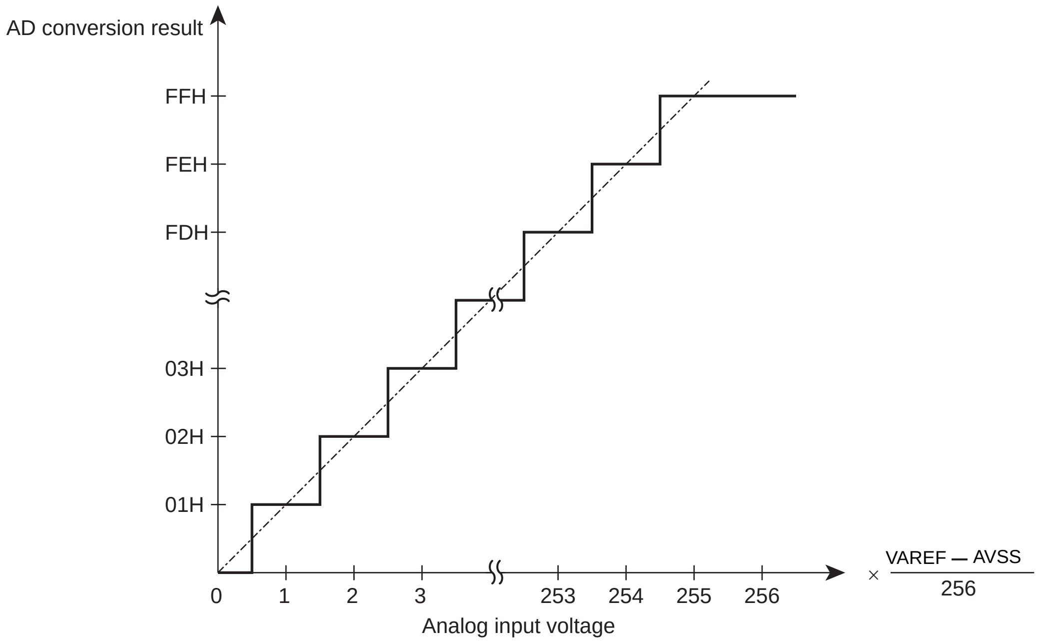


Figure 13-3 Analog Input Voltage and AD Conversion Result (typ.)

13.4 Precautions about AD Converter

13.4.1 Analog input pin voltage range

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

13.4.2 Analog input shared pins

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

13.4.3 Noise countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 13-4. 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.

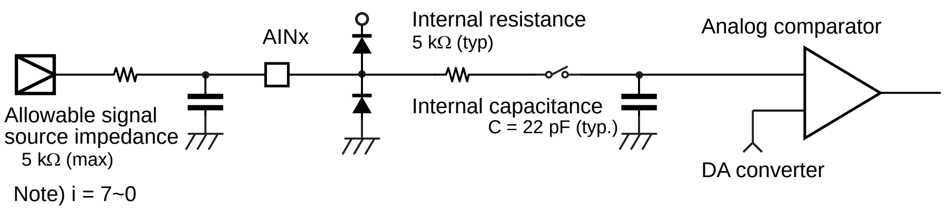


Figure 13-4 Analog Input Equivalent Circuit and Example of Input Pin Processing

14. Key-on Wakeup (KWU)

In the TMP86CK74AFG, 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

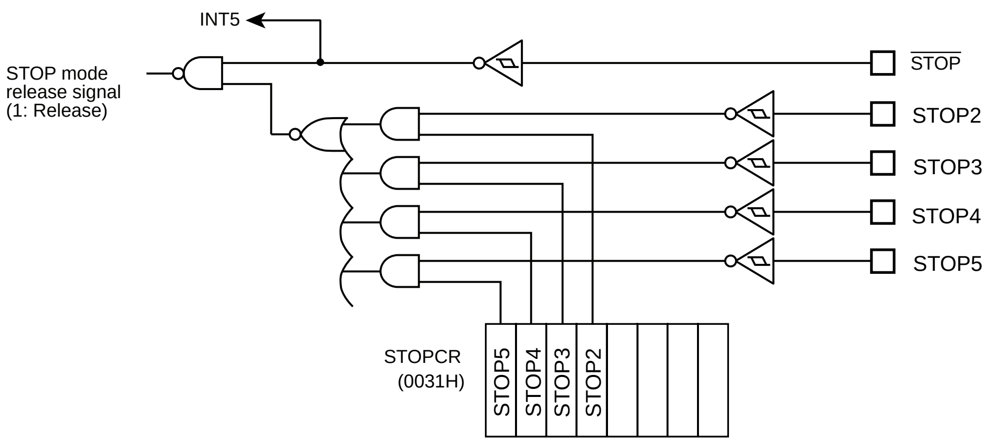


Figure 14-1 Key-on Wakeup Circuit

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	
(0031H)	STOP5	STOP4	STOP3	STOP2					(Initial value: 0000 ****)

STOP5	STOP mode released by STOP5	0:Disable 1:Enable	Write only
STOP4	STOP mode released by STOP4	0:Disable 1:Enable	Write only
STOP3	STOP mode released by STOP3	0:Disable 1:Enable	Write only
STOP2	STOP mode released by STOP2	0:Disable 1:Enable	Write only

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).

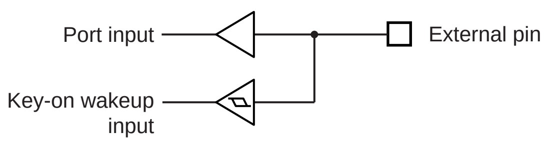


Figure 14-2 Key-on Wakeup Input and Port Input

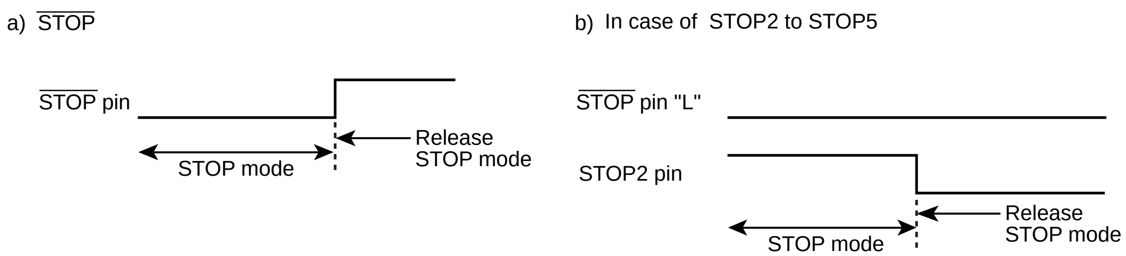


Figure 14-3 Priority of $\overline{\text{STOP}}$ pin and STOP2 to STOP5 pins

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. Vacuum Fluorescent Tube (VFT) Driver Circuit

The TMP86CK74AFG features built-in high-breakdown voltage output buffers for directly driving fluorescent tubes, and a display control circuit used to automatically transfer display data to the output port.

The segment and the digit, as it is the VFT drive circuit which included in the usual products, are not allocated. The segment and the digit can be freely allocated in the timing (T0 to T15) which is specified according to the display tube types and the layout.

15.1 Functions

1. 37 high-breakdown voltage output buffers built-in.
   - Large current output pin 16 (V0 to V15)
   - Middle current output pin 21 (V16 to V36)

   There is also the VKK pin used for the VFT drive power supply.
2. The dynamic lighting system makes it possible to select 1 to 16 digits (T0 to T15) by program.
3. Pins not used for VFT driver can be used as general-purpose ports (PD).

   Pins can be selected using the VSEL (bits 4 to 0) in VFT control register1 bit by bit.
4. Display data (80 bytes in DBR) are automatically transferred to the VFT output pin.
5. Brightness level can be adjusted in 7 steps using the dimmer function.
6. Display time are shown in Table 15-1.

Table 15-1 tdisp Time setting

SDT1	SDT2	tdisp Time	at 16 MHz	at 8 MHz	at 4 MHz	at 2 MHz	at 1 MHz
00	0	2^9/fc [s]	32 µs	64 µs	128 µs	256 µs	512 µs
01		2^{10}/fc [s]	64 µs	128 µs	256 µs	512 µs	1024 µs
10		2^{11}/fc [s]	128 µs	256 µs	512 µs	1024 µs	2048 µs
11		2^{12}/fc [s]	256 µs	512 µs	1024 µs	2048 µs	4096 µs
00	1	2^8/fc [s]	16 µs	32 µs	64 µs	128 µs	256 µs
01		2^9/fc [s]	32 µs	64 µs	128 µs	256 µs	512 µs
10		2^{10}/fc [s]	64 µs	128 µs	256 µs	512 µs	1024 µs
11		2^{11}/fc [s]	128 µs	256 µs	512 µs	1024 µs	2048 µs

15.2 Configuration

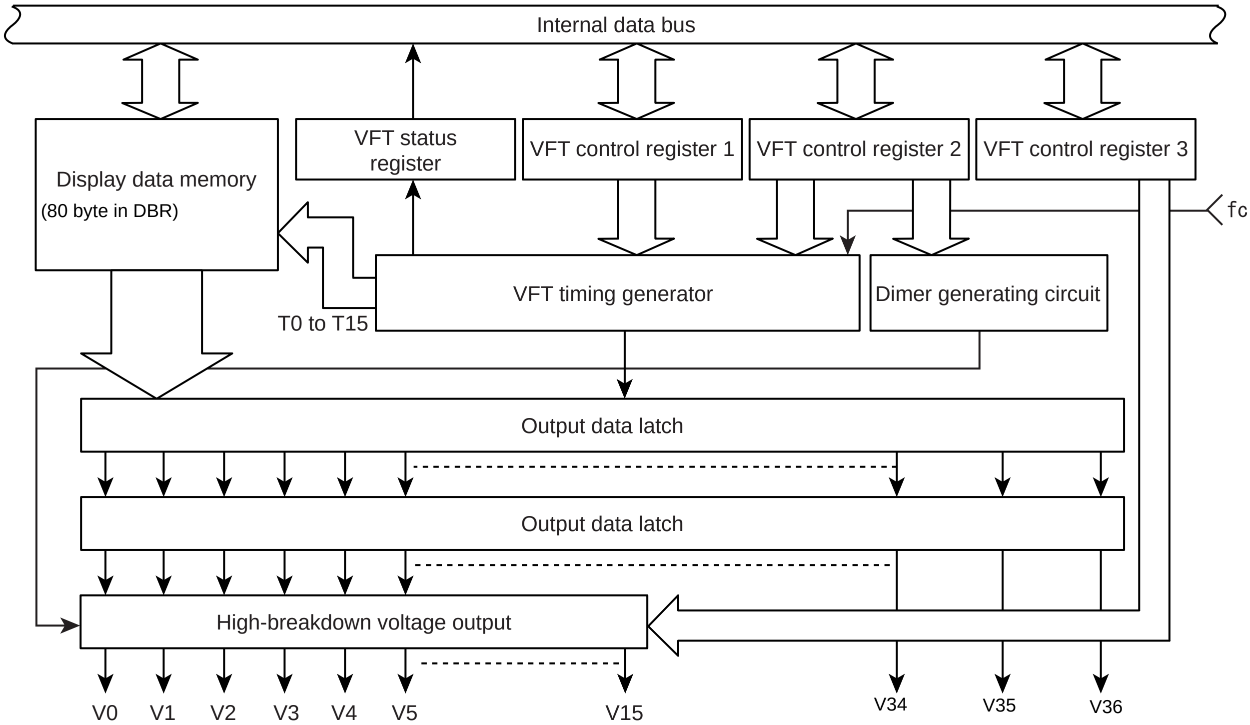


Figure 15-1 Vacuum Fluorescent Display (VFT) Circuit

15.3 Control

The VFT driver circuit is controlled by the VFT control registers (VFTCR1, VFTCR2, VFTCR3). Reading VFT status register (VFTSR) determines the VFT operating status.

Switching the mode from NORMAL1/2 to SLOW or STOP puts the VFT driver circuit into blanking state (BLK is set to “1” ; values set in the VFT control registers except BLK is maintained), and sets segment outputs and digit outputs are cleared to “0”. Thus, ports P6 to P9, and PD function as general-purpose output ports with pull-down.

VFT control register 1

VFTCR1	7	6	5	4	3	2	1	0	
(002AH)	BLK	SDT1		VSEL					(Initial value: 1000 0000)
(002AH)	BLK	SDT1		"0"					(Initial value: 1000 0000)

BLK	VFT display control	0: Display enable 1: Disable			R/W
SDT1	Display time select1 (tdisp) (Display time of 1 digit)		SDT2 = 0	SDT2 = 1	R/W
		00	2^9/fc	2^8/fc	
		01	2^{10}/fc	2^9/fc	
		10	2^{11}/fc	2^{10}/fc	
		11	2^{12}/fc	2^{11}/fc	
VSEL	Automatic display select (When using VFT driver (automatic display), V31 to V0 are only used to output VFT.) Pins which are not selected by the output pins other than the above-mentioned pins can be used as general-purpose input/output pins. (When using as a general-purpose input/output pin, the display data which corresponds to the pin must be set to “0”)	00000: 32 (V31 to V0) 00001: 33 (V32 to V0) 00010: 34 (V33 to V0) 00011: 35 (V34 to V0) 00100: 36 (V35 to V0) 00101: 37 (V36 to V0) Other: Reserved			R/W

Note 1: fc: High frequency clock [Hz]

Note 2: It is necessary to set diplay blanking staus by setting VFTCR1<BLK> to "1", when you would like to change display time(SDT1) and automatic display number (VSEL) on VFT display operation. At the same time, please make sure not to modify SDT1 and VSEL.

Note 3: Reserved: Can not access.

VFTSR	7	6	5	4	3	2	1	0	
(002DH)	WAIT								(Initial value: 1000 0000)

WAIT	VFT operational status monitor	0: VFT display in operation 1: VFT display operation disabled	Read only

Note 1: VFTSR<WAIT> is initialized to 1 after resetting.

Note 2: When VFTCR1<BLK> is cleared to 0, WAIT flag is cleared to 0 at an end of display timing. And a VFT driving circuit is enabled at an end of next display timing.

Note 3: During a VFT driving circuit is enabled, it is disabled just after an end of display timing (tdisp) by setting VFTCR1<BLK> to 1. And WAIT flag is set to 1 simultaneously.

Note 4: When a VFT driving circuit is enabled again, it is necessary that VFTCR1<BLK> is set to 1 after confirming VFTSR<WAIT> is 1.

VFT control register 2

VFTCR2 (002BH)	7	6	5	4	3	2	1	0	
	DIM			STA					(Initial value: 0010 0000)

DIM	Dimmer time select	000: Reserved 001: (14/16) × tdisp (s) 010: (12/16) × tdisp (s) 011: (10/16) × tdisp (s) 100: (8/16) × tdisp (s) 101: (6/16) × tdisp (s) 110: (4/16) × tdisp (s) 111: (2/16) × tdisp (s)	R/W
STA	Number of state (display)	00000: 1 display mode (T0) 00001: 2 display mode (T1 to T0) 00010: 3 display mode (T2 to T0) 00011: 4 display mode (T3 to T0) 00100: 5 display mode (T4 to T0) 00101: 6 display mode (T5 to T0) 00110: 7 display mode (T6 to T0) 00111: 8 display mode (T7 to T0) 01000: 9 display mode (T8 to T0) 01001: 10 display mode (T9 to T0) 01010: 11 display mode (T10 to T0) 01011: 12 display mode (T11 to T0) 01100: 13 display mode (T12 to T0) 01101: 14 display mode (T13 to T0) 01110: 15 display mode (T14 to T0) 01111: 16 display mode (T15 to T0) Others: Reserved	

Note 1: Even if a number of the display digit is set a pin which is equal to the digit dose not output.
It is necessary to write data to the data buffer which corresponds to the digit according to the display timing (T0 to T15).

VFT control register 3

VFTCR3 (002CH)	7	6	5	4	3	2	1	0	
	OWSEL					HVTR1-	HVTR0	SDT2	(Initial value: 0000 0000)

SDT2	Display time select 2 (tdisp) (Display time of 1 digit)		SDT1 = "00"	SDT1 = "01"	SDT1 = "10"	SDT1 = "11"	R/W
		0	2^9/fc [s]	2^{10}/fc [s]	2^{11}/fc [s]	2^{12}/fc [s]	
		1	2^8/fc [s]	2^9/fc [s]	2^{10}/fc [s]	2^{11}/fc [s]	
HVTR0	P6 to P9 Ports Tr time select	0	Tr normal mode typ. 150 ns (VDD = 3 V, Vkk = −35 V)				R/W
		1	Tr increment mode typ. 3 μs (VDD = 3 V, Vkk = −35 V)				
HVTR1	PD Ports Tr time select	0	(Note1) Tr normal mode (Note1) typ. 150 ns (VDD = 3 V, Vkk = −35 V)				R/W
		1	(Note1) Tr increment mode (Note1) typ. 3 μs (VDD = 3 V, Vkk = −35 V)				
OWSEL	Output waveform select (Select grid or segment)		GRID output (Dimmer enable)		SEG output		R/W
		00000	P60		P61 to PD4P97		
		00001	P60 to P61		P62 to PD4P97		
		00010	P60 to P62		P63 to PD4P97		
		00011	P60 to P63		P64 to PD4P97		
		00100	P60 to P64		P65 to PD4P97		
		00101	P60 to P65		P66 to PD4P97		
		00110	P60 to P66		P67 to PD4P97		
		00111	P60 to P67		P70 to PD4P97		
		01000	P60 to P70		P71 to PD4P97		
		01001	P60 to P71		P72 to PD4P97		
		01010	P60 to P72		P73 to PD4P97		
		01011	P60 to P73		P74 to PD4P97		
		01100	P60 to P74		P75 to PD4P97		
		01101	P60 to P75		P76 to PD4P97		
		01110	P60 to P76		P77 to PD4P97		
		01111	P60 to P77		P80 to PD4P97		
		10000	Reserved		Reserved		
		to	to		to		
		11111	Reserved		Reserved		

Note 1: A rising time of Port D is measured when Port D is connected with pull-down resistor (about 80kΩ) to VKK pin.

Note 2: It is possible to reduce the VFT port noise by using Tr increment mode. When Tr increment mode is enabled, a time of Tr is increased and also Tf. Therefore, the display time and dimmer value should be decided with the stray capacitor on a PCB. Otherwise the switching timing between grid and segment is overlapped each other and a VFT display is dimmed. Please confirm a VFT display with your set.

15.3.1 Setting of Display mode

VFT display mode is set by VFT control register 1 (VFTCR1), VFT control register 2 (VFTCR2) and VFT control register 3 (VFTCR3). VFT control register 1 (VFTCR1) sets 1 display time (tdisp) and the number of display lines (VSEL), VFT control register 2 (VFTCR2) sets dimmer timer (DIM) and state (STA) and VFT control register 3 (VFTCR3) sets Port Tr mode (HVTR0/1). (BLK of VFTCR1 must be set to “1”.) The segments and the digits are not fixed, so that they can be freely allocated. However the number of states must be specified according to the number of digits of VFT which you use. Thought the layout of VFT display mode is freely allocated, the followings are recommended; usually, large current output (V0 to V15) is used for a digit, and middle current output (V16 to V36) is used for a segment.

In case of changeing the setting of dimmer time (DIM) in display-on, it is available to change whenever the BLK status is "0".

15.3.2 Display data setting

Data are converted into VFT display data by instructions. The converted data stored in the display data buffer (addresses 0F80H to 0FCFH in DBR) are automatically transferred to the VFT driver circuit (V0 to V36), then transferred to the high-breakdown voltage output buffer. Thus, to change the display pattern, just change the data in the display data buffer.

Bits in the VFT segment (dot) and display data area correspond one to one. When data are set to 1, the segments corresponding to the bits light. The display data buffer is assigned to the DBR area shown in Figure 15-2. (The display data buffer can not be used as data memory)

Bit	0 to 7	0 to 7	0 to 7	0 to 7	0 to 4	Timing
	0F80	0F90	0FA0	0FB0	0FC0	T0
	0F81	0F91	0FA1	0FB1	0FC1	T1
	0F82	0F92	0FA2	0FB2	0FC2	T2
	0F83	0F93	0FA3	0FB3	0FC3	T3
	0F84	0F94	0FA4	0FB4	0FC4	T4
	0F85	0F95	0FA5	0FB5	0FC5	T5
	0F86	0F96	0FA6	0FB6	0FC6	T6
	0F87	0F97	0FA7	0FB7	0FC7	T7
	0F88	0F98	0FA8	0FB8	0FC8	T8
	0F89	0F99	0FA9	0FB9	0FC9	T9
	0F8A	0F9A	0FAA	0FBA	0FCA	T10
	0F8B	0F9B	0FAB	0FBB	0FCB	T11
	0F8C	0F9C	0FAC	0FBC	0FCC	T12
	0F8D	0F9D	0FAD	0FBD	0FCD	T13
	0F8E	0F9E	0FAE	0FBE	0FCE	T14
	0F8F	0F9F	0FAF	0FBF	0FCF	T15
Output pin	V0 to V7	V8 to V15	V16 to V23	V24 to V31	V32 to V36	

Bit	0 to 7	0 to 7	0 to 7	0 to 7	Timing
	0F80	0F90	0FA0	0FB0	T0
	0F81	0F91	0FA1	0FB1	T1
	0F82	0F92	0FA2	0FB2	T2
	0F83	0F93	0FA3	0FB3	T3
	0F84	0F94	0FA4	0FB4	T4
	0F85	0F95	0FA5	0FB5	T5
	0F86	0F96	0FA6	0FB6	T6
	0F87	0F97	0FA7	0FB7	T7
	0F88	0F98	0FA8	0FB8	T8
	0F89	0F99	0FA9	0FB9	T9
	0F8A	0F9A	0FAA	0FBA	T10
	0F8B	0F9B	0FAB	0FBB	T11
	0F8C	0F9C	0FAC	0FBC	T12
	0F8D	0F9D	0FAD	0FBD	T13
	0F8E	0F9E	0FAE	0FBE	T14
	0F8F	0F9F	0FAF	0FBF	T15
Output pin	V0 to V7	V8 to V15	V16 to V23	V24 to V31	

Figure 15-2 VFT Display Data Buffer Memory (DBR)

Note: Contents in data memory is cleared (unknown data) after power-on.

15.4 Display Operation

As the above-mentioned, the segment and the digit are not allocated. After setting of the display timing for the number of digits according to the using VFT and storing the segment and digit data according to the respective timings, clearing VFTCR1<BLK> to 0 starts VFT display.

Figure 15-3 shows the VFT drive pulse and Figure 15-4, Figure 15-5 show the display operation.

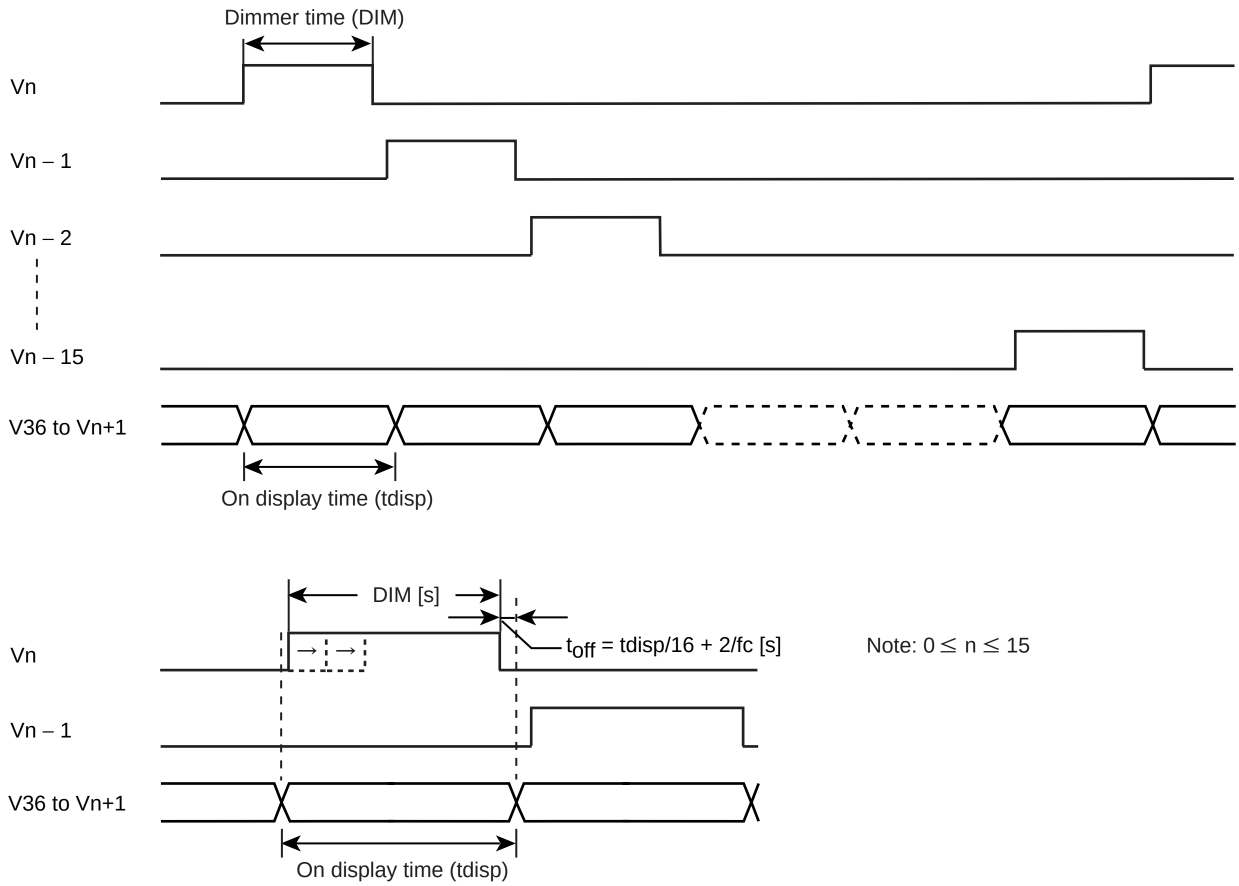


Figure 15-3 VFT Drive Wafeform and Display Timing

15.5 Example of Display Operation

15.5.1 For Conventional type VFT

When using the conventional type VFT, the output timing of the digits is specified to output 1 digit for 1 timing. Data must be set to output the pins which are specified to the digit in sequence. The following figure shows a data allocation of the display data buffer (DBR) and the output timing when VFT of 10 digits is used and V0 to V9 pins are allocated as the digit outputs. (When data is first written by the data buffer which corresponds to the digit pin, it is unnecessary to rewrite the data later.)

	T	9	8	7	6	5	4	3	2	1	0
	Address	0F99	0F98	0F87	0F86	0F85	0F84	0F83	0F82	0F81	0F80
V0	LSB			0	0	0	0	0	0	0	1
V1				0	0	0	0	0	0	1	0
V2				0	0	0	0	0	1	0	0
V3				0	0	0	0	1	0	0	0
V4				0	0	0	1	0	0	0	0
V5				0	0	1	0	0	0	0	0
V6				0	1	0	0	0	0	0	0
V7	MSB			1	0	0	0	0	0	0	0
V8	LSB	0	1								
V9		1	0								
SEG		*	*								
SEG		*	*								
SEG		*	*								
SEG		*	*								
SEG		*	*								
SEG	MSB	*	*								

SEG (Write change by display data)

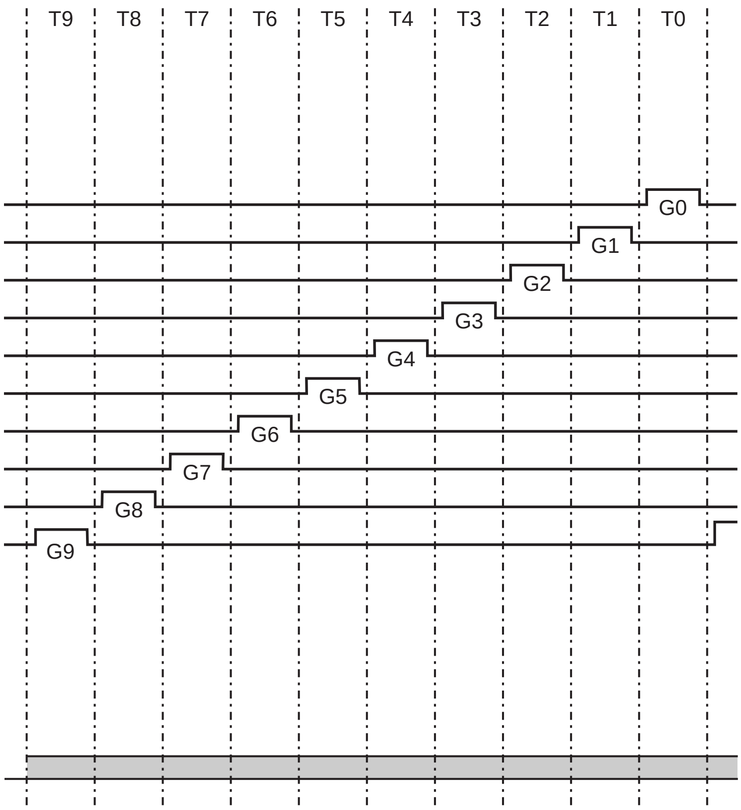


Figure 15-4 Example of Conventional type VFT driver pulse

15.5.2 For Grid scan type VFT

When using the grid scan type VFT, two or more grids must be simultaneously selected to turn the display pattern which contains two or more grids on. Additionally, the timing and the data must be determined to set the grid scan mode as follows.

- When the display pattern which is fully set in the respective grids is turned on, only the grids which correspond as ever must be scanned in sequence to turn on the display pattern. (timing of T8 to T3 in the following figure)
- When the display pattern which contains two or more grids is turned on, two or more corresponding grids are simultaneously selected to turn on the display pattern. (timing of T2 to T0 in the following figure)

	T	8	7	6	5	4	3	2	1	0
	Address	0F88 8	0F88 7	0F88 6	0F88 5	0F88 4	0F88 3	0F88 2	0F88 1	0F88 0
V0	LSB	1	0	0	0	0	0	1	0	1
V1		0	1	0	0	0	0	1	0	1
V2		0	0	1	0	0	0	0	0	1
V3		0	0	0	1	0	0	0	1	1
V4		0	0	0	0	1	0	0	1	1
V5		0	0	0	0	0	1	0	1	1
–										
–	MSB									

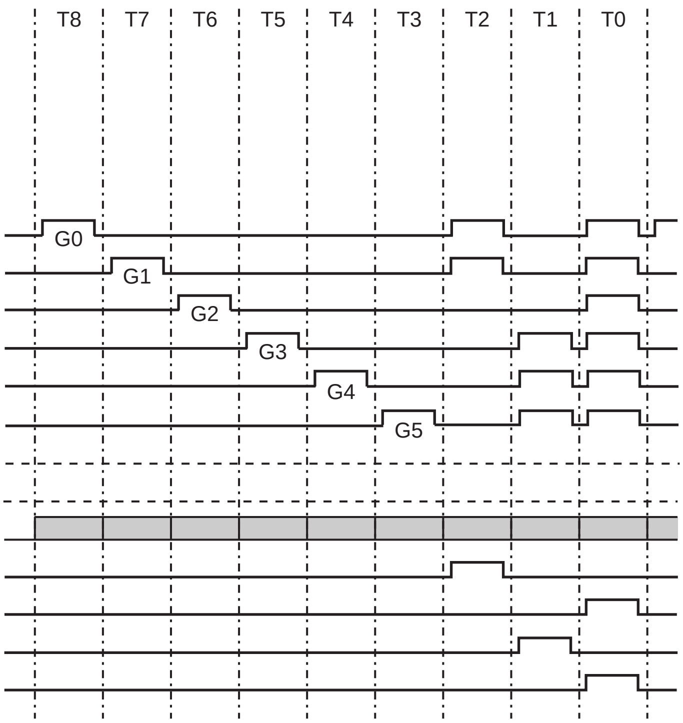


Figure 15-5 Grid Scan Type Display Vacuum Fluorescent Tube Ware

15.6 Port Function

15.6.1 High-breakdown voltage buffer

To drive fluorescent display tube, clears the port output latch to “0”. The port output latch is initialized to 0 at reset.

Precaution for using as general-purpose I/O pins are follows.

Note: When not using a pin which is pulled down (R_K = typ. 80 kΩ) to pin VKK , it must be set to open. It is necessary to clear the port output latch and the data buffer memory (DBR) to “0”.

15.6.1.1 Ports P6 to P9

When a part of P6 to P9 is used as the input/output pin (VFT driver in operation), the data buffer memory (DBR) of the segment which is also used as the input/output pin must be cleared to “0”.

15.6.1.2 Port PD

VFT output and usual input/output are controlled by VFTCR1<VSEL> in bits.

15.6.2 Caution

When a pin which is pulled down to pin VKK is used as usual output or input, the following cautions are required.

15.6.2.1 When outputting

When level “L” is output, a port which is pulled down to pin VKK is pin VKK voltage. Such processes as clamping with the diode as shown in Figure 15-6 (a) are necessary to prevent pin VKK voltage applying to the external circuit.

15.6.2.2 When inputting

When the external data is input, the port output latch is cleared to “0”.

The input threshold is the same as that of the other usual input/output port. However it is necessary to drive R_K(typ. 80 kΩ) sufficiently because of pulled down to pin VKK.

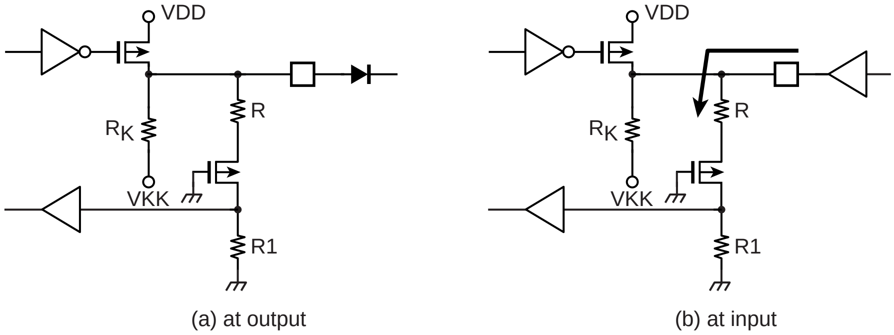
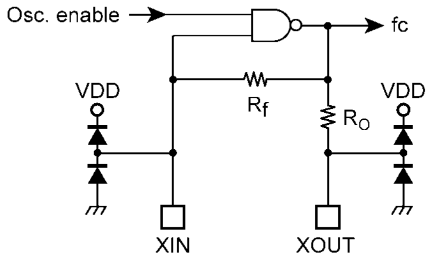
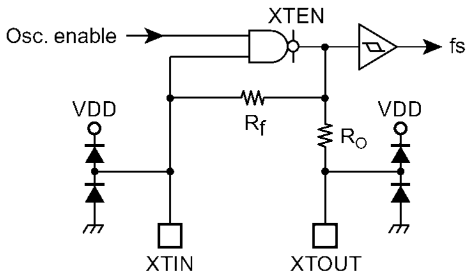
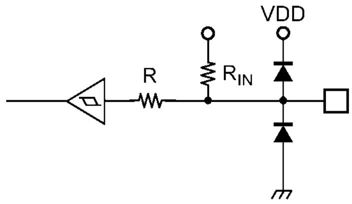
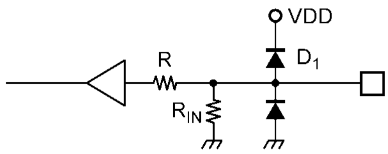


Figure 15-6 External Circuit Interface

16. Input/Output Circuitry

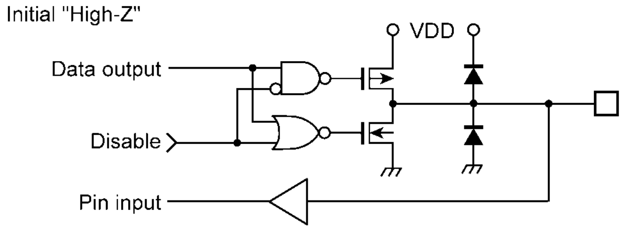
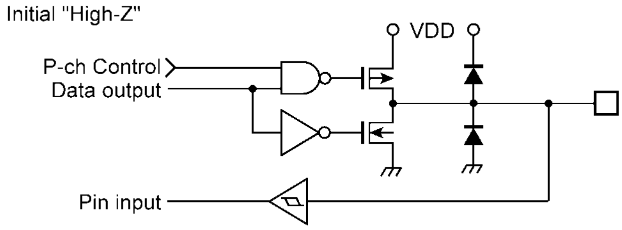
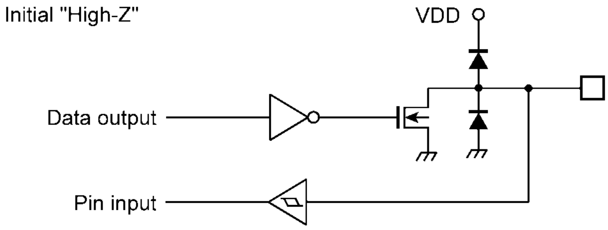
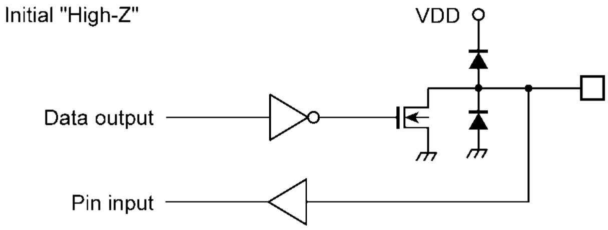
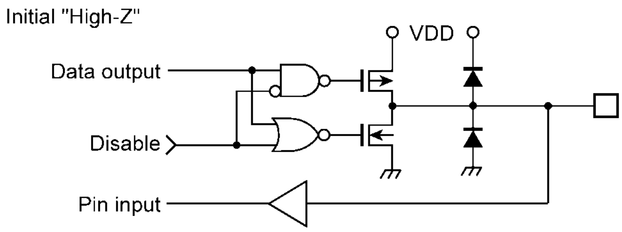
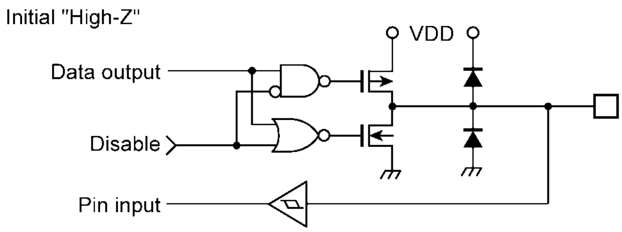
16.1 Control Pins

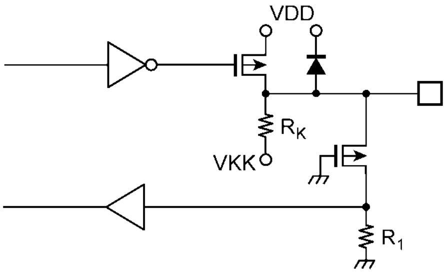
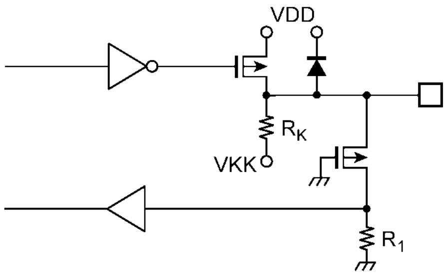
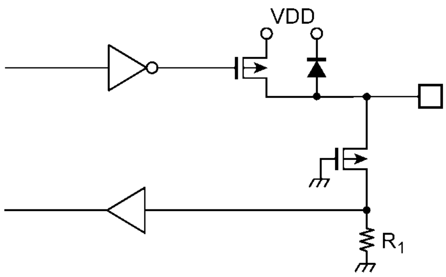
The input/output circuitries of the TMP86CK74AFG control pins are shown below.

<table>
<tr><th>Control Pin</th><th>I/O</th><th>Input/Output Circuitry</th><th>Remarks</th></tr>
<tr><td>XIN
XOUT</td><td>Input
Output</td><td>
</td><td>Resonator connecting pins (High-frequency)
R_f = 1.2 MΩ (typ.)
R_O = 500 Ω (typ.)</td></tr>
<tr><td>XTIN
XTOUT</td><td>Input
Output</td><td>
</td><td>Resonator connecting pins (Low-frequency)
R_f = 6 MΩ (typ.)
R_O = 220 kΩ (typ.)</td></tr>
<tr><td>$\overline{\text{RESET}}$</td><td>Input</td><td>
</td><td>Hysteresis input
Pull-up resistor
R_{IN} = 220 kΩ (typ.)
R = 1 kΩ (typ.)</td></tr>
<tr><td>TEST</td><td>Input</td><td>
</td><td>Pull-down resistor
R_{IN} = 70 kΩ (typ.)
R = 1 kΩ (typ.)</td></tr>
</table>

Note: The TEST pin of TMP86PM74 does not have a pull-down resistor and protect diode(D1). Fix the TEST pin at low-level in MCU mode.

16.2 Input/Output Ports

Control Pin	I/O	Input/Output Circuitry	Remarks
P0	I/O		Tri-state I/O
P1	I/O		Programmable Open drain ouput Hysteresis input
P2	I/O		Sink open drain output Hysteresis input
P3	I/O		Sink open drain output Large current ouput
P4	I/O		Tri-state I/O
P5	I/O		Tri-state I/O Hysteresis input

Control Pin	I/O	Input/Output Circuitry	Remarks
P6 P7	I/O		Source open drain I/O High breakdown voltage (Large current) R_K = 80 kΩ (typ.) R1 = 200 kΩ (typ.)
P8 P9	I/O		Source open drain I/O High breakdown voltage (Middle current) R_K = 80 kΩ (typ.) R_1 = 200 kΩ (typ.)
PD	I/O		Source open drain I/O High breakdown voltage (Middle current) R_1 = 200 kΩ (typ.)

17. Electrical Characteristics

17.1 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 ratings is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products, which include this device, ensure that no absolute maximum rating value will ever be exceeded.

(V_{SS} = 0 V)

Parameter		Symbol	Pins	Ratings	Unit
Supply voltage		V_{DD}		−0.3 to 6.5	V
Input voltage		V_{IN}		−0.3 to V_{DD} + 0.3	V
Output voltage		V_{OUT1}		−0.3 to V_{DD} + 0.3	V
		V_{OUT2}	Sink open drain port	V_{DD} − 41 to V_{DD} + 0.3	V
Output current (Per 1 pin)	IOL	I_{OUT1}	P0, P01, P2, P4, P5 ports	5	mA
		I_{OUT2}	P3 port	40	
	IOH	I_{OUT3}	P0, P1, P4, P5 ports	−3	
		I_{OUT4}	P6, P7 ports	−30	
		I_{OUT5}	P8, P9 P_D ports	−20	
Output current (Total)	IOL	ΣI_{OUT1}	P0, P01, P2, P4, P5 ports	120	
	IOH	ΣI_{OUT2}	P6, P7, P8, P9 P_D ports	−120	
Power dissipation [Topr = 25°C]		P_D		1200	mW
Soldering temperture (Time)		Tsld		260 (10 s)	°C
Storage temperature		Tstg		−55 to 125	
Operating temperature		Topr		−30 to 70	

Note 1: All V_{DDs} should be connected externally for keeping the same voltage level.

Note 2: Power Dissipation (P_D); For P_D, it is necessary to decrease −14.3 mW/°C.

17.2 Operating Conditions

The Operating Conditions shows the conditions under which the device be used in order for it to operate normally while maitaining 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.

Parameter	Symbol	Pins	Condition		Min	Max	Unit
Supply voltage	V_{DD}		fc = 16 MHz	NORMAL1, 2 modes	4.5	5.5	V
				IDLE0, 1, 2 modes			
			fc = 8 MHz	NORMAL1, 2 modes	2.7		
				IDLE0, 1, 2 modes			
			fs = 32.768 kHz	SLOW1, 2 modes			
				SLEEP0, 1, 2 modes			
				STOP mode			
Output voltage	V_{OUT3}	Source open drain pins			V_{DD} – 38	V_{DD}	
Input high voltage	V_{IH1}	Except hysteresis input			$V_{DD} \times 0.70$	V_{DD}	
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
Input low voltage	V_{IL1}	Except hysteresis input			0	$V_{DD} \times 0.30$	
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
Clock frequency	fc	XIN, XOUT	V_{DD} = 2.7 V to 5.5 V		1.0	8.0	MHz
			V_{DD} = 4.5 V to 5.5 V			16.0	
	fs	XTIN, XTOUT			30.0	34.0	kHz

17.3 How to Calculate Power Consumption

The share of VFT driver loss (VFT driver output loss + pull-down resistor (RK) loss) in power consumption Pmasx of TMP86CK74AFG is high. When using a fluorescent display tube with a large number of segments, the maximum power consumption Pd must not be exceeded.

17.3.1 Power consumption Pmax = operating power consumption + normal output port loss + VFT driver loss

1. Operating power consumption: VDD × IDD
2. Normal output port loss: $\Sigma I_{OUT1} \times 0.4$
3. VFT driver loss: VFT driver output loss + pull-down resistor (R_K) loss

Example: When Ta = –10°C to 50°C

(When using a fluorescent display tube with a conventional type which can use only one grid output at the same time.) and a fluorescent display tube with segment output = 3mA, digit output = 12mA, V_{KK} = –34.5 V is used.

Operating conditions; VDD = 5 V ± 10%, fc = 8 MHz, VFT dimmer time (DIM) = (14/16) × t_{SEG},

Power consumption Pmax = (1) + (2) + (3)

1. Operating power consumption: VDD × IDD = 5.5 V × 10 mA = 55 mW
2. Normal output port loss: $\Sigma I_{OUT1} \times 0.4$ = 60 mA × 0.4 V = 24 mW
3. VFT driver loss:

   Segment pin = 3 mA × 2 V × number of segments X = 6 mW × X

   Grid pin = 12 mA × 2 V × 14/16 (DIM)) × number of grids Y = 21 mW × Y

   R_K loss = $(5.5\text{ V} + 34.5\text{ V})^2$ / 50 kΩ × (number of segments X + number of grids Y)

   = 32 mW × (X + Y

Therefore, Pmax = 55 mW + 24 mW + 6 mW × X + 21 mW + 32 mW × (X + Y)

= 132 mW + 38 mWX

Maximum power consumption Pd when Ta = 50°C is determined by the following equation ;

PD = 1200 mW – (14.3 mW × 25°C) = 842.5 mW

The number of segments X that can be lit is:

PD > Pmax

842.5 mW > 132 + 38X

18.69 < X

Thus, a fluorescent display tube with less than 18 segments can be used. If a fluorescent display tube with 18 segments or more is used, either a pull-down resistor must be attached externally, or the number of segments to be lit must be kept to less than 18 by software.

17.4 DC Characteristics

17.4.1 DC Characteristics (1) (V_{DD} = 5 V)

[Condition] V_{DD} = 5.0 V ± 10%, V_{SS} = A_{VSS} = 0 V, Topr = −30 to 70°C (Typ.: V_{DD} = 5.0 V, Topr = 25°C, Vin = 5.0 V/0 V)

<table>
<tr><th>Parameter</th><th>Symbol</th><th>Pins</th><th colspan="2">Condition</th><th>Min</th><th>Typ.</th><th>Max</th><th>Unit</th></tr>
<tr><td>Hysteresis voltage</td><td>V_{HS}</td><td>Hysteresis input</td><td colspan="2"></td><td>–</td><td>0.9</td><td>–</td><td>V</td></tr>
<tr><td rowspan="3">Input current</td><td>I_{IN1}</td><td>TEST</td><td colspan="2" rowspan="3">V_{DD} = 5.5 V, V_{IN} = 5.5 V/0 V</td><td rowspan="3">–</td><td rowspan="3">–</td><td rowspan="3">±2</td><td rowspan="3">μA</td></tr>
<tr><td>I_{IN2}</td><td>Sink open drain, Tri-st</td></tr>
<tr><td>I_{IN3}</td><td>$\overline{RESET}$, STOP</td></tr>
<tr><td>Input resistance</td><td>R_{IN}</td><td>$\overline{RESET}$ pull-up</td><td colspan="2"></td><td>100</td><td>220</td><td>450</td><td rowspan="2">kΩ</td></tr>
<tr><td>Pull-down resistance (Note1)</td><td>R_K</td><td>Sink open drain</td><td colspan="2">V_{DD} = 5.5 V, V_{KK} = −30 V</td><td>50</td><td>–</td><td>110</td></tr>
<tr><td rowspan="2">Output leakage current</td><td>I_{LO1}</td><td>Sink open drain, Tri-st</td><td colspan="2">V_{DD} = 5.5 V, V_{OUT} = 5.5 V</td><td>–</td><td>–</td><td>±2</td><td rowspan="2">μA</td></tr>
<tr><td>I_{LO2}</td><td>Sink open drain</td><td colspan="2">V_{DD} = 5.5 V, V_{KK} = −32 V</td><td>–</td><td>–</td><td>±2</td></tr>
<tr><td>Output high voltage</td><td>V_{OH}</td><td>Tri-st</td><td colspan="2">V_{DD} = 4.5 V, I_{OH} = −0.7 mA</td><td>4.1</td><td>–</td><td>–</td><td rowspan="2">V</td></tr>
<tr><td>Output low voltage</td><td>V_{OL}</td><td>Except XOUT, P3 port</td><td colspan="2">V_{DD} = 4.5 V, I_{OL} = 1.6 mA</td><td>–</td><td>–</td><td>0.4</td></tr>
<tr><td rowspan="2">Output high current</td><td>I_{OH1}</td><td>P6, P7 port</td><td colspan="2">V_{DD} = 4.5 V, V_{OH} = 2.4 V</td><td>−18</td><td>−28</td><td>–</td><td rowspan="9">mA</td></tr>
<tr><td>I_{OH2}</td><td>P8, P9 P_D port</td><td colspan="2">V_{DD} = 4.5 V, V_{OH} = 2.4 V</td><td>−9</td><td>−14</td><td>–</td></tr>
<tr><td>Output low current</td><td>I_{OL}</td><td>High-current (P3 port)</td><td colspan="2">V_{DD} = 4.5 V, V_{OL} = 1.0 V</td><td>–</td><td>30</td><td>–</td></tr>
<tr><td rowspan="2">Supply current in NORMAL1, 2 modes</td><td rowspan="8">I_{DD}</td><td rowspan="8"></td><td>fc = 16.0 MHz
fs = 32.768 kHz</td><td rowspan="4">AD converter disable (IREF off)</td><td>–</td><td>12</td><td>18</td></tr>
<tr><td>fc = 8.0 MHz
fs = 32.768 kHz</td><td>–</td><td>6</td><td>9</td></tr>
<tr><td rowspan="2">Supply current in IDLE0, 1, 2 modes</td><td>fc = 16.0 MHz
fs = 32.768 kHz</td><td>–</td><td>6</td><td>9</td></tr>
<tr><td>fc = 8.0 MHz
fs = 32.768 kHz</td><td>–</td><td>3</td><td>4.5</td></tr>
<tr><td rowspan="2">Supply current in NORMAL1, 2 modes</td><td>fc = 16.0 MHz
fs = 32.768 kHz</td><td rowspan="2">AD converter enable</td><td>–</td><td>13</td><td>19</td></tr>
<tr><td>fc = 8.0 MHz
fs = 32.768 kHz</td><td>–</td><td>7</td><td>10</td></tr>
<tr><td rowspan="2">Supply current in STOP mode</td><td>Topr = to 50°C</td><td rowspan="2">AD converter disable</td><td>–</td><td rowspan="2">0.5</td><td>5</td><td rowspan="2">μA</td></tr>
<tr><td>Topr = to 70°C</td><td>–</td><td>10</td></tr>
</table>

Note 1: Topr = −10 to 70°C

Note 2: Typical values show those at Topr = 25°C, V_{DD} = 5 V

Note 3: Input current (I_{IN1}, I_{IN3}); The current through pull-up or pull-down resistor is not included.

Note 4: I_{DD} does not include I_{REF} current.

17.4.2 DC Characteristics (2) (V_{DD} = 3 V)

[Condition] V_{DD} = 3.0 V ± 10%, V_{SS} = A_{VSS} = 0 V, Topr = –30 to 70°C (Typ.: V_{DD} = 3.0 V, Topr = 25°C, Vin = 3.0 V/0 V)

Parameter	Symbol	Pins	Condition		Min	Typ.	Max	Unit
Hysteresis voltage	V_{HS}	Hysteresis input			–	0.4	–	V
Input current	I_{IN1}	TEST	V_{DD} = 3.3 V, V_{IN} = 3.3 V/0 V		–	–	±2	μA
	I_{IN2}	Sink open drain, Tri-st						
	I_{IN3}	$\overline{RESET}$, STOP						
Input resistance	R_{IN}	$\overline{RESET}$ pull-up			100	220	450	kΩ
Pull-down resistance	R_K	Sink open drain	V_{DD} = 3.3 V, V_{KK} = –30 V		45	–	105	
Output leakage current	I_{LO1}	Sink open drain, Tri-st	V_{DD} = 3.3 V, V_{OUT} = 3.3 V/0 V		–	–	±2	μA
	I_{LO2}	Sink open drain	V_{DD} = 3.3 V, V_{KK} = –32 V		–	–	±2	
Output high voltage	V_{OH}	Tri-st	V_{DD} = 2.7 V, I_{OH} = –0.6 mA		2.3	–	–	V
Output low voltage	V_{OL}	Except XOUT, P3 port	V_{DD} = 2.7 V, I_{OL} = 0.9 mA		–	–	0.4	
Output high current	I_{OH1}	P6, P7 port	V_{DD} = 2.7 V, V_{OH} = 1.5 V		–5.5	–8	–	mA
	I_{OH2}	P8, P9, P_D port	V_{DD} = 2.7 V, V_{OH} = 1.5 V		–3	–4.5	–	
Output low current	I_{OL}	High-current (P3 port)	V_{DD} = 2.7 V, V_{OL} = 1.0 V		–	6	–	
Supply current in NORMAL1, 2 modes	I_{DD}		fc = 8.0 MHz fs = 32.768 kHz	AD converter disable (IREF off)	–	3	4.5	
Supply current in IDLE0, 1, 2 modes			fc = 8.0 MHz fs = 32.768 kHz		–	2	2.5	
Supply current in NORMAL1, 2 modes			fc = 8.0 MHz fs = 32.768 kHz	AD converter enable	–	3.5	5	
Supply current in SLOW1, 2 modes			fs = 32.768 kHz	AD converter disable	–	30	60	μA
Supply current in SLEEP0, 1, 2 modes					–	15	30	
Supply current in STOP mode			Topr = to 50°C		–	0.5	5	
			Topr = to 70°C		–		10	

Note 1: Typical values show those at Topr = 25°C, V_{DD} = 3 V

Note 2: Input current (I_{IN1}, I_{IN3}); 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 SLOW2 and SLEEP2 modes are equivalent to IDLE0, 1, 2.

17.5 AD Characteristics

(V_{SS} = 0 V, 4.5 V ≤ V_{DD} ≤ 5.5 V, Topr = –30 to 70°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		V_{DD} – 1.5	–	V_{DD}	V
Analog reerence voltage range	ΔV_{AREF}		3.0	–	–	
Analog input voltage	V_{AIN}		0	–	V_{AREF}	
Analog supply current	I_{REF}	V_{DD} = V_{AREF} = 5.5 V, V_{SS} = 0.0 V	–	0.6	1.0	mA
Non linearity error		V_{DD} = V_{AREF} = 4.5 to 5.5 V, V_{SS} = 0.0 V	–	–	±1	LSB
Zero point error			–	–	±1	
Full scale error			–	–	±1	
Total error			–	–	±2	

(V_{SS} = 0 V, 2.7 V ≤ V_{DD} < 4.5 V, Topr = –30 to 70°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		V_{DD} – 1.5	–	V_{DD}	V
Analog reerence voltage range	ΔV_{AREF}		2.5	–	–	
Analog input voltage	V_{AIN}		0	–	V_{AREF}	
Analog supply current	I_{REF}	V_{DD} = V_{AREF} = 4.5 V, V_{SS} = 0.0 V	–	0.5	0.8	mA
Non linearity error		V_{DD} = V_{AREF} = 2.7 to 4.5 V, V_{SS} = 0.0 V	–	–	±1	LSB
Zero point error			–	–	±1	
Full scale error			–	–	±1	
Total error			–	–	±2	

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

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

Note 3: Please use input voltage to AIN input pin in limit of V_{AREF} – V_{SS}. When voltage of range outside is input, conversion value becomes unsettled and gives affect to other channel conversion value.

Note 4: Analog Reference Voltage Range: $\Delta V_{AREF} = V_{AREF} - V_{SS}$

17.6 AC Characteristics

(V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = –30 to 70°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcyc	NORMAL1, 2 mode	0.25	–	4	μs
		IDLE0, 1, 2 mode				
		SLOW1, 2 mode	117.6	–	133.3	
		SLEEP0, 1, 2 mode				
High level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 16 MHz	–	31.25	–	ns
Low level clock pulse width	t_{WCL}					
High level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	μs
Low level clock pulse width	t_{WSL}					

(V_{SS} = 0 V, V_{DD} = 2.7 to 4.5 V, Topr = –30 to 70°C)

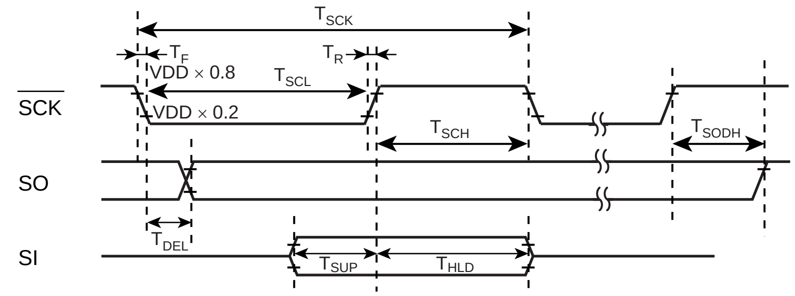
Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcyc	NORMAL1, 2 mode	0.5	–	8	μs
		IDLE0, 1, 2 mode				
		SLOW1, 2 mode	117.6	–	133.3	
		SLEEP0, 1, 2 mode				
High level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 8 MHz	–	62.5	–	ns
Low level clock pulse width	t_{WCL}					
High level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	μs
Low level clock pulse width	t_{WSL}					

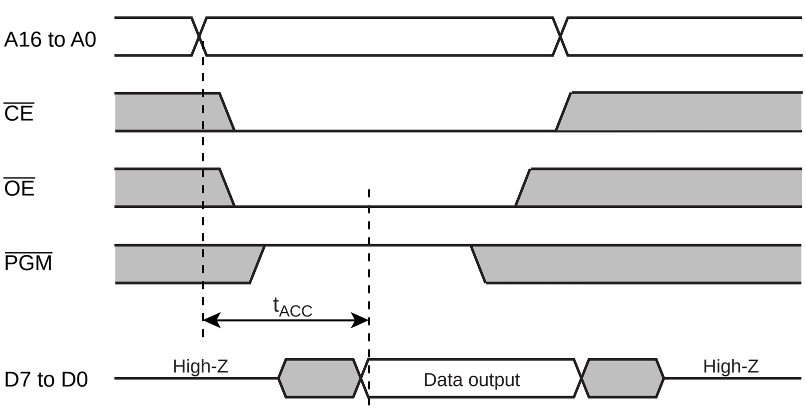
17.7 HSIO AC Characteristics

(V_{SS} = 0 V, V_{DD} = 2.7 to 5.5 V, Topr = –30 to 70°C)

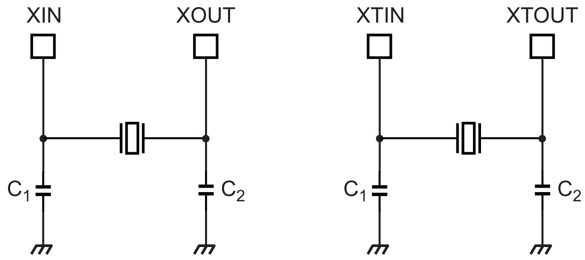
Parameter	Symbol	Condition	Min	Typ.	Max	Unit
$\overline{SCK}$ output period (Internal clock)	T_{SCK1}	8 MHz < fc ≤ 16 MHz V_{DD} = 4.5 V to 5.5 V	16/fc	–	–	s
$\overline{SCK}$ output low width (Internal clock)	T_{SCL1}		8/fc – 100 ns	–	–	
$\overline{SCK}$ output high width (Internal clock)	T_{SCH1}		8/fc – 100 ns	–	–	
$\overline{SCK}$ output period (Internal clock)	T_{SCK2}	4 MHz < fc ≤ 8 MHz V_{DD} = 2.7 V to 5.5 V	8/fc	–	–	
$\overline{SCK}$ output low width (Internal clock)	T_{SCL2}		4/fc – 100 ns	–	–	
$\overline{SCK}$ output high width (Internal clock)	T_{SCH2}		4/fc – 100 ns	–	–	
$\overline{SCK}$ output period (Internal clock)	T_{SCK3}	fc ≤ 4 MHz V_{DD} = 2.7 V to 5.5 V	4/fc	–	–	
$\overline{SCK}$ output low width (Internal clock)	T_{SCL3}		2/fc – 100 ns	–	–	
$\overline{SCK}$ output high width (Internal clock)	T_{SCH3}		2/fc – 100 ns	–	–	
$\overline{SCK}$ input period (External clock)	T_{SCK4}	fc ≤ 8 MHz (V_{DD} = 2.7 V to 5.5 V) fc ≤ 16 MHz (V_{DD} = 4.4 V to 5.5 V)	1000	–	–	ns
$\overline{SCK}$ input low width (External clock)	T_{SCL4}		400	–	–	
$\overline{SCK}$ input low width (External clock)	T_{SCH4}		400	–	–	
SI input setup time	T_{SUP}		200	–	–	
SI input hold time	T_{HLD}		200	–	–	
SO output delay time	T_{DEL}		–	–	200	
Rising time	T_R	V_{DD} = 3.0 V, CL ≤ 50 pF (Note)	–	–	100	
Falling time	T_F		–	–	100	
SO last bit hold time	T_{SODH}		16.5/fc	–	32.5/fc	

Note: CL, External Capacitance





17.8 Recommended Oscillating Conditions



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

Note 1: To ensure stable oscillation, the resonator position, load capacitance, etc. must be appropriate. Because these factors are greatly affected by board patterns, please be sure to evaluate operation on the board on which the device will actually be mounted.

Note 2: For the resonators to be used with Toshiba microcontrollers, we recommend ceramic resonators manufactured by Murata Manufacturing Co., Ltd.
For details, please visit the website of Murata at the following URL:
http://www.murata.com

17.9 Handling Precaution

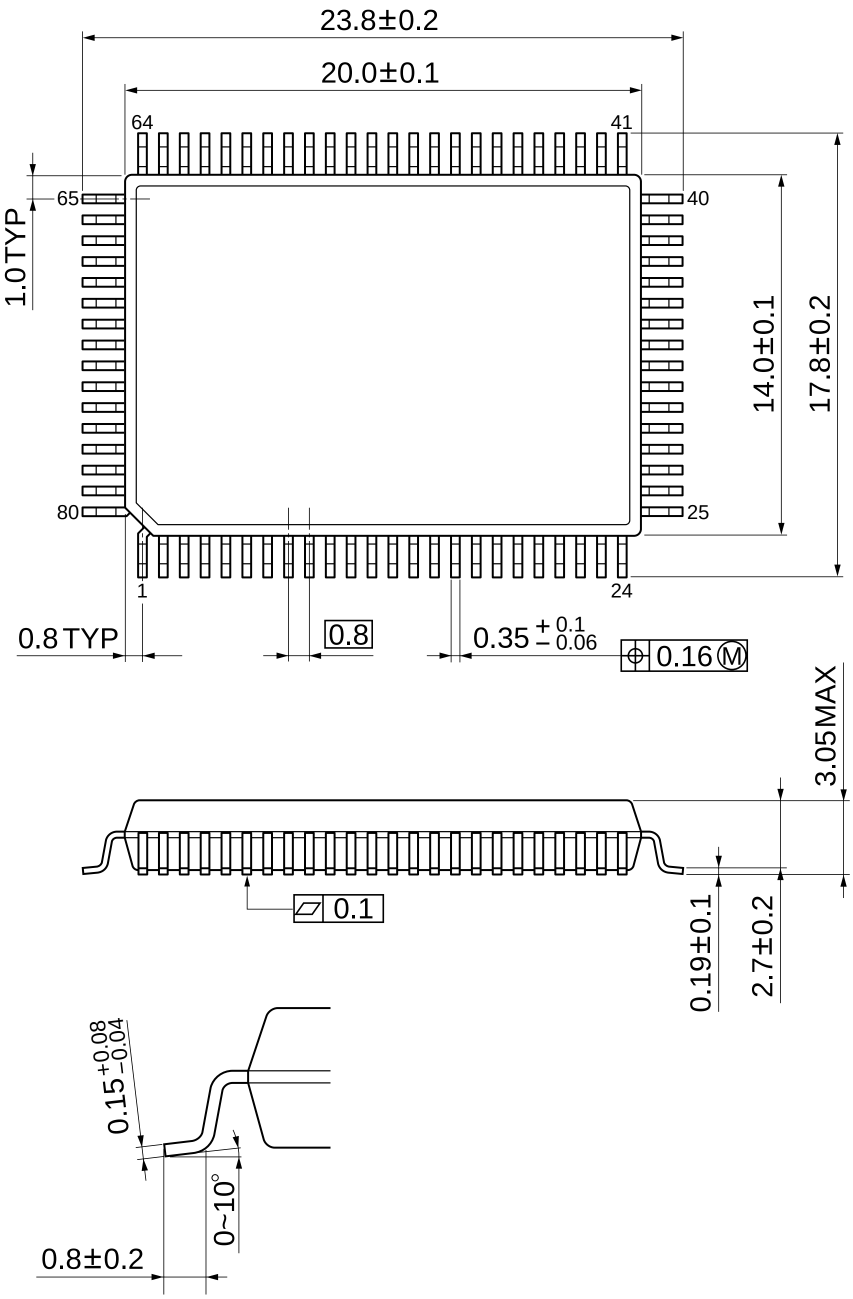
- The solderability test conditions for lead-free products (indicated by the suffix G in product name) are shown below.
  1. When using the Sn-37Pb solder bath
     Solder bath temperature = 230 °C
     Dipping time = 5 seconds
     Number of times = once
     R-type flux used
  2. When using the Sn-3.0Ag-0.5Cu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

18. Package Dimensions

QFP80-P-1420-0.80M Rev 02

Unit: mm



18. Package Dimensions

TMP86CK74AFG

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.