CMOS 8-Bit Microcontroller

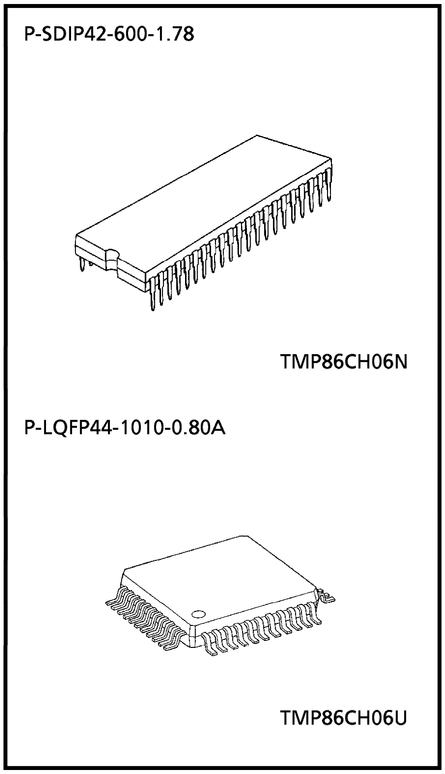
TMP86CH06N/TMP86CH06U

The TMP86CH06 is the 8-bit single chip microcomputer, which contains ROM, RAM, multi-function timer/counters, serial interface (UART/SIO) with high speed, high performance and low power consumption.

Product No.	ROM	RAM	Package	OTP MCU
TMP86CH06N	16K × 8-bit	512 × 8-bit	P-SDIP42-600-1.78	TMP86PH06N
TMP86CH06U			P-LQFP44-1010-0.80A	TMP86PH06U

Features

- ◆ 8-bit single chip microcomputer TLCS-870/C Series
- ◆ Minimum Instruction execution time: 0.25 μs (at 16.0 MHz)
  - ● Power consumption is reducible by means of conversion of instruction execution time
    0.25 μs, 0.50 μs, 1.0 μs, 2.0 μs, 4.0 μs, 8.0 μs, 122 μs
    (at 16.0 MHz, 32.768 kHz operation)
- ◆ External Bus Interface
  - ● up to 64 Kbytes (for both Program and Data memory)
  - ● Multiplexed between Lower Address-bus and Data-bus
- ◆ 21 Interrupt factors (6 for External, 15 for Internal)
- ◆ Input/Output Ports: 35 pins
  - ● High-Current Output (Typ. 20 mA, LED direct drive): 8 pins
- ◆ 16-bit Timer/Counter: 1 channel
  - ● Timer, Event counter, Pulse Width measurement and External-triggered timer



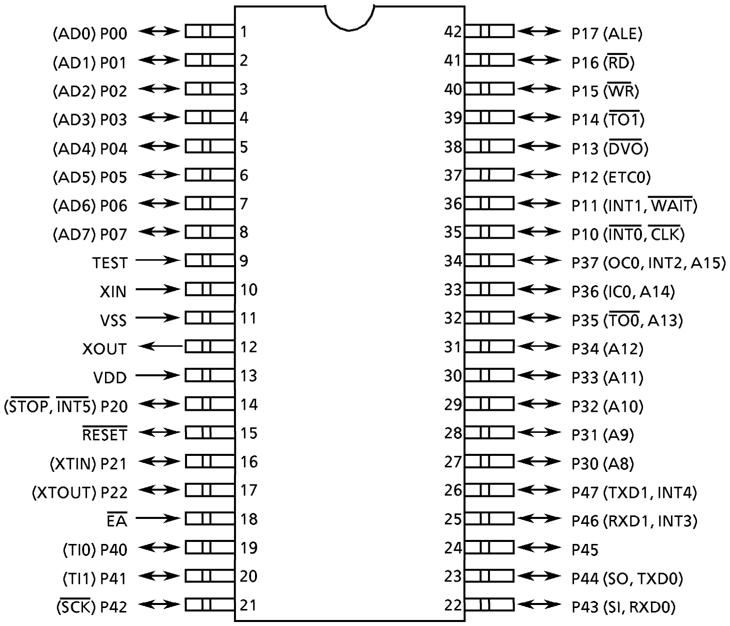
000707EBP1

- ● 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.
- ● 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..
- ● 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.
- ● The products described in this document are subject to the foreign exchange and foreign trade laws.
- ● The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- ● The information contained herein is subject to change without notice.

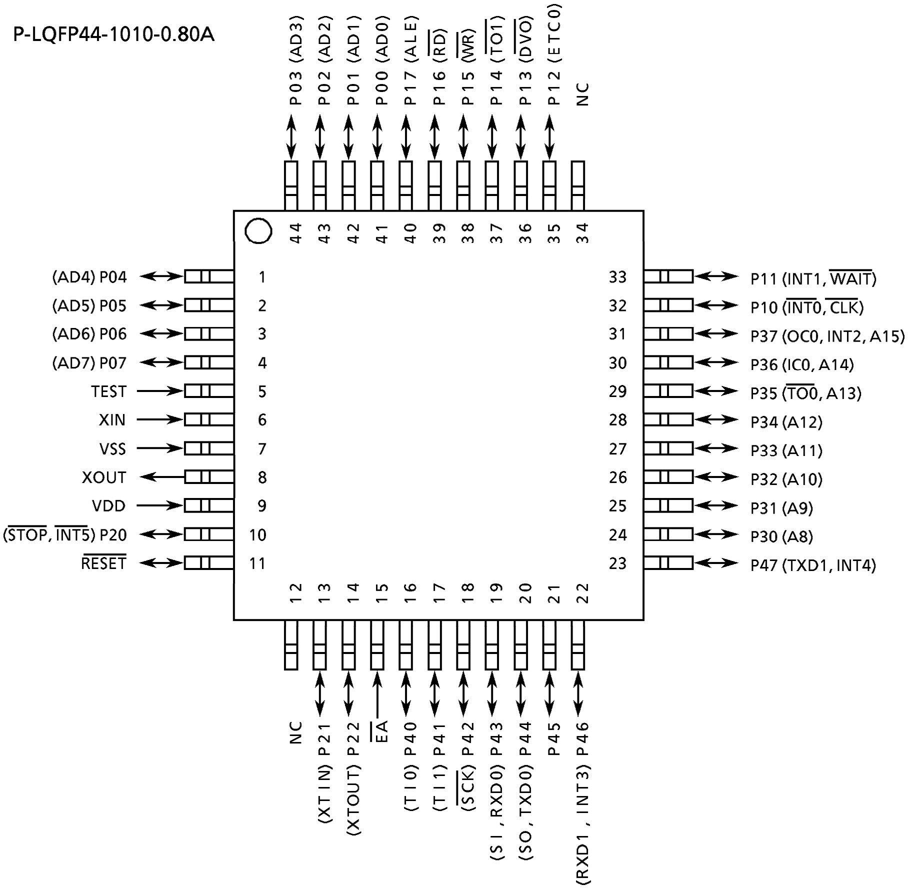
- 8-bit Timer/Counter: 2 channels
  - Available for 16-bit timer by cascade connection
  - Timer, Event counter, Pulse Width Modulation, Programmed Pulse Generator and Programmable Divider Output, Warming-up Counter
- Serial Interface
  - 8-bit UART: 2 channels
  - 8-bit SIO (synchronized): 1 channel
- Clock Oscillation circuit: 2 units
  - For Single or Dual clock mode
- Nine power saving operating modes
  - STOP mode: Oscillation stops. Battery/Capacitor back-up. Port output hold/High-impedance.
  - SLOW 1, 2 mode: Low power consumption operation using low-frequency clock (32.768 kHz)
  - IDLE 0 mode: CPU stops, and peripherals operate using high-frequency clock of Time-Base-Timer. Release by falling edge of TBTCR<TBTCK> setting.
  - IDLE 1 mode: CPU stops, and peripherals operate using high-frequency clock. Release by interruputs.
  - IDLE 2 mode: CPU stops, and peripherals operate using high and low frequency clock. Release by interruputs.
  - SLEEP 0 mode: CPU stops, and peripherals operate using low-frequency clock of Time-Base-Timer. Release by falling edge of TBTCR<TBTCK> setting.
  - SLEEP 1 mode: CPU stops, and peripherals operate using low-frequency clock. Release by interrupts.
  - SLEEP 2 mode: CPU stops, and peripherals operate using high and low frequency clock. Release by interrupts.
- Operating Voltage: 4.5 to 5.5 V at 16.0 MHz/32.768 kHz, 2.7 to 5.5 V at 8 MHz/32.768 kHz,
  1.8 to 5.5 V at 4.2 MHz/32.768 kHz.

Pin Assignments (Top View)

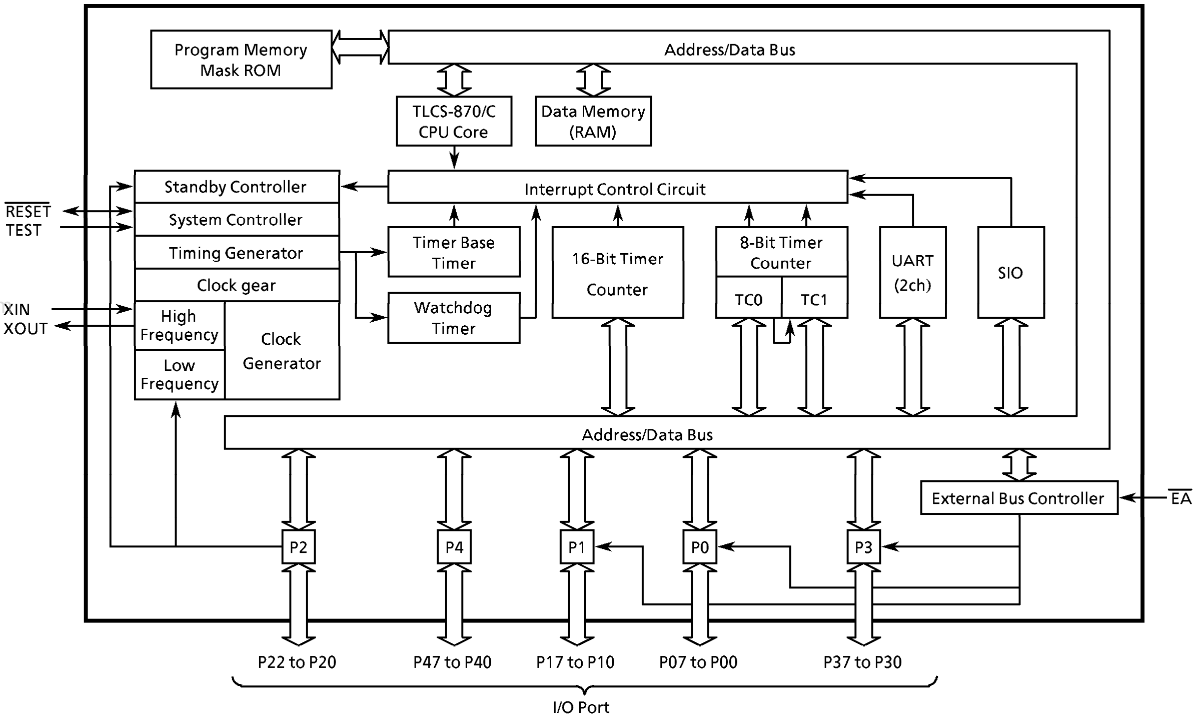
P-SDIP42-600-1.78


(AD0) P00 1
(AD1) P01 2
(AD2) P02 3
(AD3) P03 4
(AD4) P04 5
(AD5) P05 6
(AD6) P06 7
(AD7) P07 8
TEST 9
XIN 10
VSS 11
XOUT 12
VDD 13
($\overline{\text{STOP}}$, $\overline{\text{INT5}}$) P20 14
$\overline{\text{RESET}}$ 15
(XTIN) P21 16
(XTOUT) P22 17
$\overline{\text{EA}}$ 18
(TI0) P40 19
(TI1) P41 20
($\overline{\text{SCK}}$) P42 21
42 P17 (ALE)
41 P16 ($\overline{\text{RD}}$)
40 P15 ($\overline{\text{WR}}$)
39 P14 ($\overline{\text{TO1}}$)
38 P13 ($\overline{\text{DVO}}$)
37 P12 (ETC0)
36 P11 (INT1, $\overline{\text{WAIT}}$)
35 P10 ($\overline{\text{INT0}}$, $\overline{\text{CLK}}$)
34 P37 (OC0, INT2, A15)
33 P36 (IC0, A14)
32 P35 ($\overline{\text{TO0}}$, A13)
31 P34 (A12)
30 P33 (A11)
29 P32 (A10)
28 P31 (A9)
27 P30 (A8)
26 P47 (TXD1, INT4)
25 P46 (RXD1, INT3)
24 P45
23 P44 (SO, TXD0)
22 P43 (SI, RXD0)

P-LQFP44-1010-0.80A


P03 (AD3) 44
P02 (AD2) 43
P01 (AD1) 42
P00 (AD0) 41
P17 (ALE) 40
P16 ($\overline{\text{RD}}$) 39
P15 ($\overline{\text{WR}}$) 38
P14 ($\overline{\text{TO1}}$) 37
P13 ($\overline{\text{DVO}}$) 36
P12 (ETC0) 35
NC 34
(AD4) P04 1
(AD5) P05 2
(AD6) P06 3
(AD7) P07 4
TEST 5
XIN 6
VSS 7
XOUT 8
VDD 9
($\overline{\text{STOP}}$, $\overline{\text{INT5}}$) P20 10
$\overline{\text{RESET}}$ 11
33 P11 (INT1, $\overline{\text{WAIT}}$)
32 P10 ($\overline{\text{INT0}}$, $\overline{\text{CLK}}$)
31 P37 (OC0, INT2, A15)
30 P36 (IC0, A14)
29 P35 ($\overline{\text{TO0}}$, A13)
28 P34 (A12)
27 P33 (A11)
26 P32 (A10)
25 P31 (A9)
24 P30 (A8)
23 P47 (TXD1, INT4)
12 NC
13 P21 (XTIN)
14 P22 (XTOUT)
15 $\overline{\text{EA}}$
16 P40 (TI0)
17 P41 (TI1)
18 P42 ($\overline{\text{SCK}}$)
19 P43 (SI, RXD0)
20 P44 (SO, TXD0)
21 P45
22 P46 (RXD1, INT3)

Block Diagram



Pin Names And Functions (1/3)

Pin Name	No. of Pins	Input/Output	Functions
P00 to P07 AD0 to AD7	8	I/O 3-states	Port 0: 8-bit I/O port that allows selection of input/output on bit basis. Address/Data bus: Functions as 8-bit bidirectional address/data bus for external memory (TTL input).
P10 $\overline{INT0}$ $\overline{CLK}$	1	I/O Input Output	Port 10: I/O port that allows selection of input/output on bit basis. (Schmidt Input) External Interrupt Request 0 Clock: Clock output
P11 INT1 $\overline{WAIT}$	1	I/O Input Input	Port 11: I/O port that allows selection of input/output on bit basis. (Schmidt Input) External Interrupt Request 1 Wait: Wait request from external Memory
P12 ETC0	1	I/O Input	Port 12: I/O port that allows selection of input/output on bit basis. (Schmidt Input) Extended Timer Input 0
P13 $\overline{DVO}$	1	I/O Output	Port 13: I/O port that allows selection of input/output on bit basis. (Schmidt Input) Divider Output
P14 $\overline{TO1}$	1	I/O Output	Port 14: I/O port that allows selection of input/output on bit basis. (Schmidt Input) 8-bit Timer 1 Output
P15 $\overline{WR}$	1	I/O Output	Port 15: I/O port that allows selection of input/output on bit basis. (Schmidt Input) Write: Generates strobe signal to write data on External Memory.
P16 $\overline{RD}$	1	I/O Output	Port 16: I/O port that allows selection of input/output on bit basis. (Schmidt Input) Read: Generates strobe signal to read data from External Memory.
P17 ALE	1	I/O Output	Port 17: I/O port that allows selection of input/output on bit basis. (Schmidt Input) Address Latch Enable: The negative edge of ALE supplies an address latch timing on AD0 to AD7 for External Memory.
P20 $\overline{STOP}$ $\overline{INT5}$	1	I/O Input Input	Port 20: I/O port that allows selection of input/output on bit basis. Open-Drain Output. (Schmidt Input) STOP Releasing: The rising edge or High Level Releases STOP mode. External Interrupt Request 5
P21 XTIN	1	I/O Input	Port 21: I/O port that allows selection of input/output on bit basis. Open-Drain Output. (Schmidt Input) Low Frequency Clock Input
P22 XTOUT	1	I/O Output	Port 22: I/O port that allows selection of input/output on bit basis. Open-Drain Output. (Schmidt Input) Low Frequency Clock Output
P30 to P34 A8 to A12	5	I/O Output	Port 30 to 34: I/O port that allows selection of input/output on bit basis. (Schmidt Input) Address Bus 8 to 12

Pin Names And Functions (2/3)

Pin Name	No. of Pins	Input/Output	Functions
P35	1	I/O	Port 35: I/O port that allows selection of input/output on bit basis. (Schmidt Input)
$\overline{\text{TO0}}$		Output	8-bit Timer Output 0
A13		Output	Address Bus 13
P36	1	I/O	Port 36: I/O port that allows selection of input/output on bit basis. (Schmidt Input)
IC0		Input	Capture Input 0 for Extended Timer
A14		Output	Address Bus 14
P37	1	I/O	Port 37: I/O port that allows selection of input/output on bit basis. (Schmidt Input)
OC0		Output	Output Compare 0 for Extended Timer
INT2		Input	External Interrupt Request 2 (Schmidt Input)
A15		Output	Address Bus 15
P40	1	I/O	Port 40: I/O port that allows selection of input/output on bit basis. Programmable Open-Drain Output. (Schmidt Input)
TI0		Input	8-bit Timer Input 0
P41	1	I/O	Port 41: I/O port that allows selection of input/output on bit basis. Programmable Open-Drain Output. (Schmidt Input)
TI1		Input	8-bit Timer Input 1
P42	1	I/O	Port 42: I/O port that allows selection of input/output on bit basis. Programmable Open-Drain Output. (Schmidt Input)
$\overline{\text{SCK}}$		I/O	Clock input/output for SIO
P43	1	I/O	Port 43: I/O port that allows selection of input/output on bit basis. Programmable Open-Drain Output. (Schmidt Input)
SI, RXD0		Input	Data Input for UART/SIO channel 0
P44	1	I/O	Port 44: I/O port that allows selection of input/output on bit basis. (Schmidt Input)
$\overline{\text{SO}}$, TXD0		Output	Data Output for UART/SIO channel 0. Programmable Open-Drain Output.
P45	1	I/O	Port 45: I/O port that allows selection of input/output on bit basis. Programmable Open-Drain Output.
P46	1	I/O	Port 46: I/O port that allows selection of input/output on bit basis. Programmable Open-Drain Output. (Schmidt Input)
RXD1		Input	Data Input for UART channel 1
INT3		Input	External Interrupt Request 3
P47	1	I/O	Port 47: I/O port that allows selection of input/output on bit basis. (Schmidt Input)
TXD1		Output	Data Output for UART channel 1. Programmable Open-Drain Output.
INT4		Input	External Interrupt Request 4 (Schmidt Input)

Pin Names And Functions (3/3)

Pin Name	No. of Pins	Input/Output	Functions
$\overline{\text{EA}}$	1	Input	External Access: Fix to HIGH level to utilize internal ROM. Fix to LOW level to utilize external memory.
TEST	1	Input	to be fixed to LOW level
$\overline{\text{RESET}}$	1	I/O	RESET signal input or watchdog timer output/address trap output/system-clock-reset-output
XIN/XOUT	2	I/O	High-frequency resonator is to be connected.
VSS	1	Input	Ground
VDD	1	Input	Power Supply

Operational Description

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, the external memory interface, and the reset circuit.

1.1 Memory Address Maps

The TMP86CH06 memory consists of 3 blocks: ROM, RAM and SFR (Special Function Register). They are all mapped in 64K-byte address space. Figure 1-1 shows the TMP86CH06 memory address maps. The general-purpose register banks are not assigned to the RAM address space.

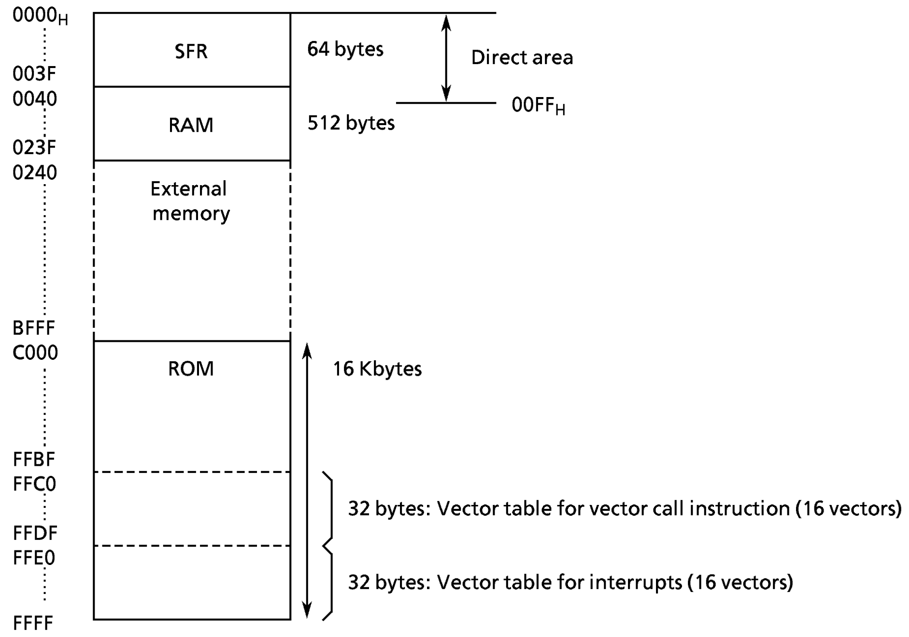


Figure 1-1. Memory Address Maps

1.2 Program Memory (ROM)

The TMP86CH06 can address up to 64 Kbytes of external program memory space except the SFR area and the internal RAM. However, placing program memory on the internal RAM is deregulated if a certain procedure is executed (See 2.4.5 Address trap). The TMP86CH06 contains a 16-Kbyte program memory (mask ROM) at addresses from $C000_H$ to $FFFF_H$.

1.3 Data Memory (RAM)

The TMP86CH06 is available up to 64 Kbytes of data memory area. Data memory consists of internal data memory (internal ROM or RAM) and external data memory (ROM or RAM). The TMP86CH06 has 512 bytes of internal RAM. The first 192 bytes (0040_H to $00FF_H$) of the internal RAM are located in the direct area; instructions with shorten operations are available against such an area.

1.4 External Memory Interfaces

The TMP86CH06 can be connected with the external memory through address bus, a data bus, and the control bus. It is available up to 64 Kbytes of data memory area except the area where the internal RAM and SFR are located. Data bus and the lower bits of address bus are multiplexed.

- The operation mode is selected from the following three by setting $\overline{\text{EA}}$ terminal and control register EXPCR.

  1) Single chip mode: ($\overline{\text{EA}}$ = 1, EXPCR<RDOE, WROE> = "0, 0")
  The TMP86CH06 operates within internal memory (ROM, RAM and SFR).
  Each terminal for connecting external memory is available for input/output port.

  2) Expansion mode: ($\overline{\text{EA}}$ = 1, EXPCR<either or both RDOE, WROE> = "1")
  After releasing reset, the TMP86CH06 is starts to operate within internal memory (ROM, RAM and SFR).
  If either or both RDOE and WROE are enabled, terminals for external memory start to function as ALE and AD7 to 0.

  3) ROM-less mode: ($\overline{\text{EA}}$ = 0)
  After releasing reset, the TMP86CH06 operates with external memory instead of internal ROM. The internal RAM and SFR are still be used.

- Wait Control

  The operation mode is selected from the following three: Non wait, 1-cycle wait or (1 + n)-cycle wait mode.

- Internal ROM security function

  The TMP86CH06 can keep CPU from reading the data on internal ROM, while its read instruction is located in external memory.

Note: The transfer destination data of the vector table (located from FFC0H to FFFFH) is not protected.

1.4.1 Controlling

The external memory interfaces are controlled by expansion control register (EXPCR) and wait control register (WAITCR).
EXPCR directly manages $\overline{\text{RD}}$, $\overline{\text{WR}}$ and address bus output. And, assigning terminals to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ subsequently provides AD7 to 0 and ALE. In order to utilize external memory, port directions whether to be input or output is determined regardless of each bit on control register. After reset, EXPCR is a register which can be written only once. Therefore, the second and the following write operation cannot modify the data on EXPCR.

External Access Control Register

	7	6	5	4	3	2	1	0	
EXPCR (0031_H)	ABUSEN			–	RDOE	WROE	–	–	(Initial value: 111* 11**) $\overline{\text{EA}}$ ← "0" (Initial value: 000* 00**) $\overline{\text{EA}}$ ← "1"

ABUSEN	Upper bits on address bus output enable [A15 to A8]	000: input/output port 001: A9 to A8 output 010: A10 to A8 output 011: A11 to A8 output 100: A12 to A8 output 101: A13 to A8 output 110: A14 to A8 output 111: A15 to A8 output (001 to 111) other terminals on P3 port keep their input/output mode.	WR 1 time
RDOE	$\overline{\text{RD}}$ strobe output enable	0: $\overline{\text{RD}}$ strobe output Disable 1: $\overline{\text{RD}}$ strobe output Enable	Write only
WROE	$\overline{\text{WR}}$ strobe output enable	0: $\overline{\text{WR}}$ strobe output Disable 1: $\overline{\text{WR}}$ strobe output Enable	Write only

Note 1: Once either RDOE or WROE or both are enabled to "1", the following terminals are utilized for external memory interface; therefore they cannot be used for general-purpose input/output ports.

- *AD7 to 0 (P07 to P00)*
- *ALE (P17)*
- *$\overline{\text{RD}}$ (P16: if RDOE = "1") or $\overline{\text{WR}}$ (P15:if WROE = "1")*
- *CLK (P10)*

Note 2: EXPCR is a write only register and must not be used with any of the read-modify-write instructions.

WAITCR manages security for internal ROM, number of waiting cycles and waiting clock output. When control waiting cycles, WAITCR selects waiting mode from three: no-wait, 1-state wait or (1 + n)-state wait.

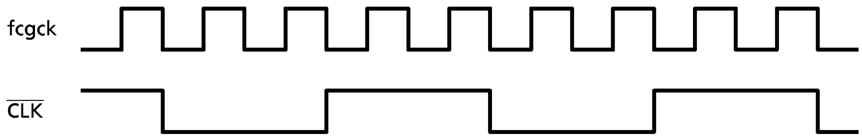
WAITCR selects output mode from the following: High output by controlling waiting clock output when an external bus is used, $\overline{\text{CLK}}$ output during waiting cycles, $\overline{\text{CLK}}$ output at all times. When the internal ROM is disabled while programs on the external memory and the internal RAM are operating, data in C000H to FFBFH on the ROM cannot be read under software control. WAITCR is a register which can be written only once. Therefore, the second and the following write operation cannot modify this bit.

WAIT Control Register

WAITCR (0032$_H$)	7	6	5	4	3	2	1	0	
	WAIT		–	"0"	CLKV		–	RDMEM	(Initial value: 01*0 11*1) $\overline{\text{EA}}$ ← "0" (Initial value: 00*0 00*1) $\overline{\text{EA}}$ ← "1"

WAIT	Control waiting cycles	00: Wait Disable 01: (Fixed) 1-cycle wait Enable 10: (1 + n)-cycle wait Enable ($\overline{\text{WAIT}}$ pin) 11: Wait Disable	Write only
CLKV	$\overline{\text{CLK}}$ output mode	0* : "H" output ($\overline{\text{CLK}}$ output disable) 10: $\overline{\text{CLK}}$ output enable during waiting cycles 11: always $\overline{\text{CLK}}$ output Enable	R/W
RDMEM	security for internal ROM reading out	0: Read out Disable 1: Read out Enable	WR 1 time

Note 1: The timing chart during $\overline{\text{CLK}}$ output enable is described below.



Note 2: P11 terminal is assigned for $\overline{\text{WAIT}}$ input when WAIT is "10"; in the condition, it cannot be used for general-purpose input/output port.

Note 3: WAITCR include a write only register and must not be used with any of the read-modify-write instructions.

Note 4: Make sure to set "0" on bit 4.

Note 5: When external memory is used, the P10 pin cannot be used as an external interrupt input ($\overline{\text{INT0}}$) or an I/O port as $\overline{\text{CLK}}$ is output from this pin.

1.4.2 External Bus Controller

The TMP86CH06 is interfaced to an external memory via following buses.

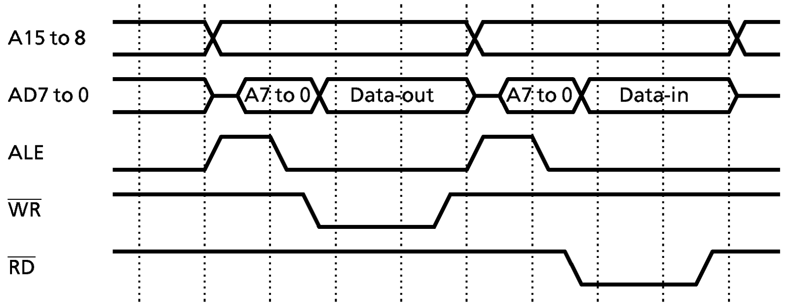
(1) Address/Databus: A15 to 8, AD7 to 0

There are 16-bit output address bus and 8-bit bidirectional databus. The lower 8 bits for address bus and 8 bits for databus are multiplexed. All of the terminals, AD15 to 8 and AD7 to 0, are used also for port. These ports functions as address/databus by setting register EXPCR. Upper bits of address bus can be output either partially or totally.

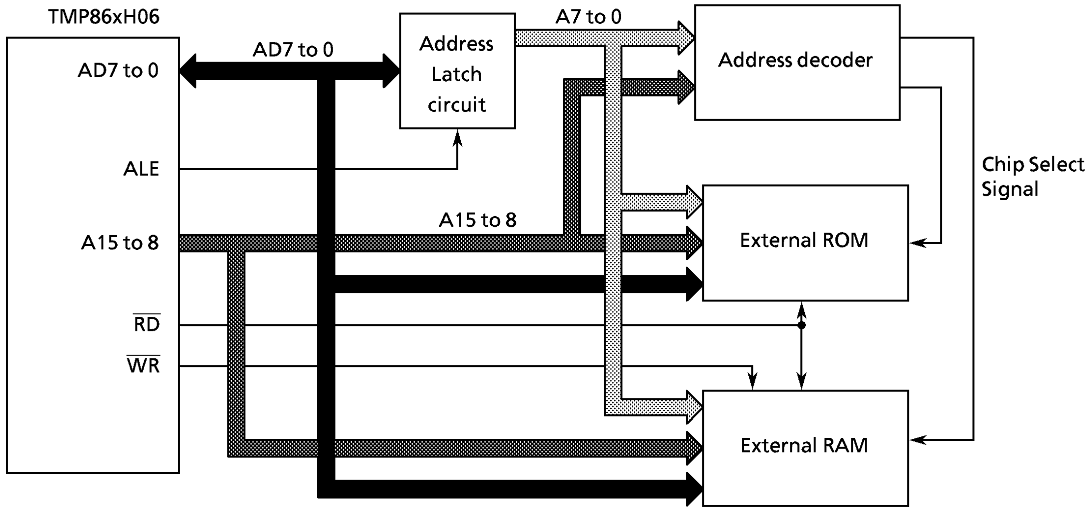
(2) Control signals: $\overline{RD}$, $\overline{WR}$, ALE

$\overline{RD}$ pulse indicates for reading, $\overline{WR}$ pulse indicates for writing, and ALE pulse indicates for address latch enable. These terminals are used also for timing for port, and these ports operate as such control signals. $\overline{RD}$ becomes "L" level on external memory read cycle, and $\overline{WR}$ becomes "L" level on external memory write cycle. Both $\overline{RD}$ and $\overline{WR}$ keep "H" level on both dummy cycle and internal memory read/write cycle.

a) Timing chart on external memory interface



b) Example for connecting external ROM/RAM

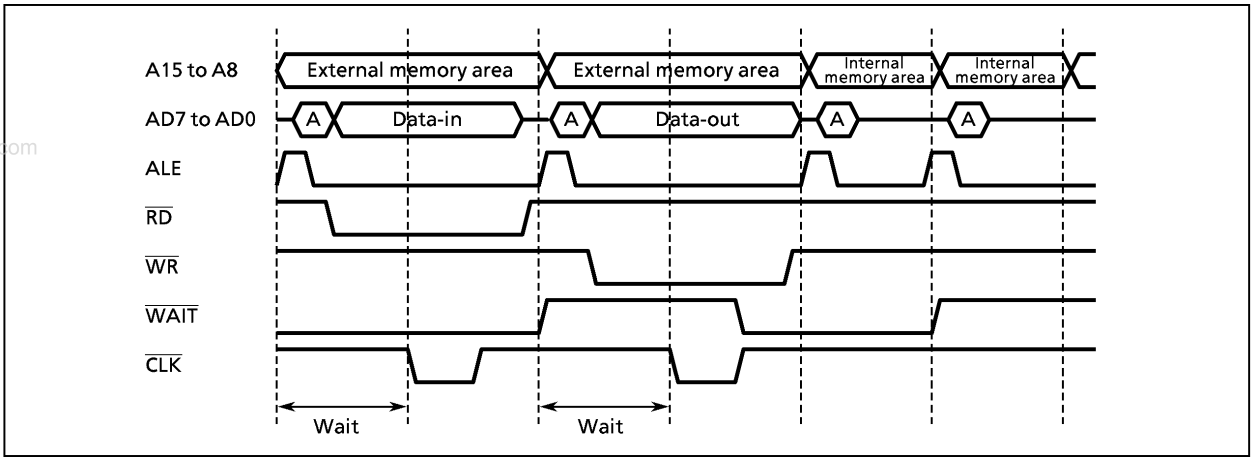


1.4.3 Wait Controller

The number of wait cycles is selected from three: no wait, fixed 1-cycle wait and (1 + n)-cycle wait.

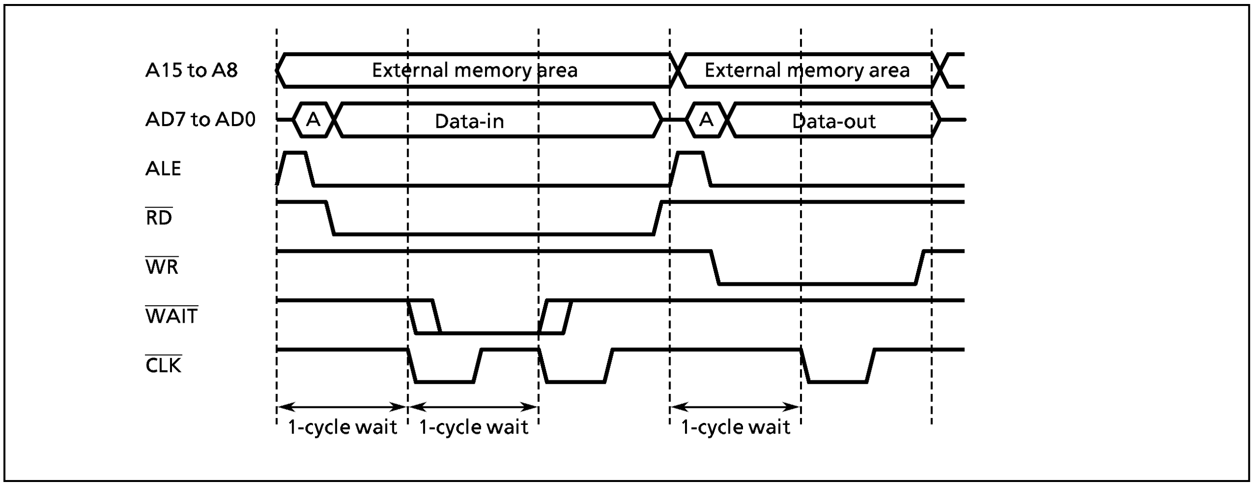
(1) Fixed 1-cycle wait

1-state wait operation is executed regardless of $\overline{\text{WAIT}}$ terminal condition. The following diagram indicates the wait cycle, under WAIT = “01” and /CLK outputs during waiting cycles.



(2) (1 + n)-cycle wait

Continuous wait operation is executed as far as “L” input is detected from $\overline{\text{WAIT}}$ terminal, after 1-cycle wait operation is executed. The following diagram indicates the wait cycle, under WAIT = “10” and $\overline{\text{CLK}}$ outputs during waiting cycles.



1.5 System Clock Controller

The system clock controller consists of a clock generator, a clock gear, a timing generator, and a stand-by controller.

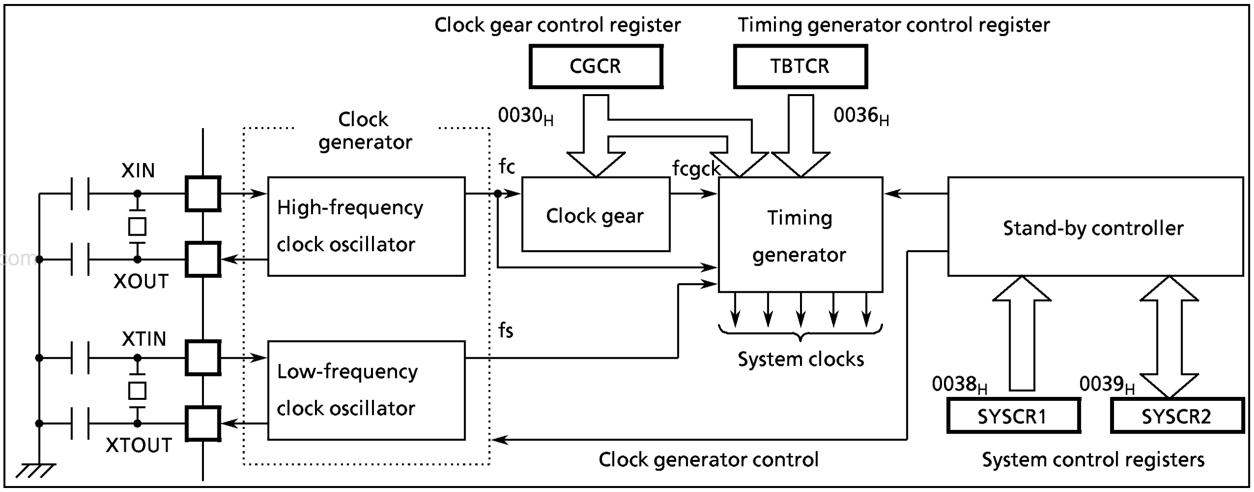


Figure 1-2. System Clock Control

1.5.1 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) and low-frequency (fs) clocks 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/XOUT pin not connected.

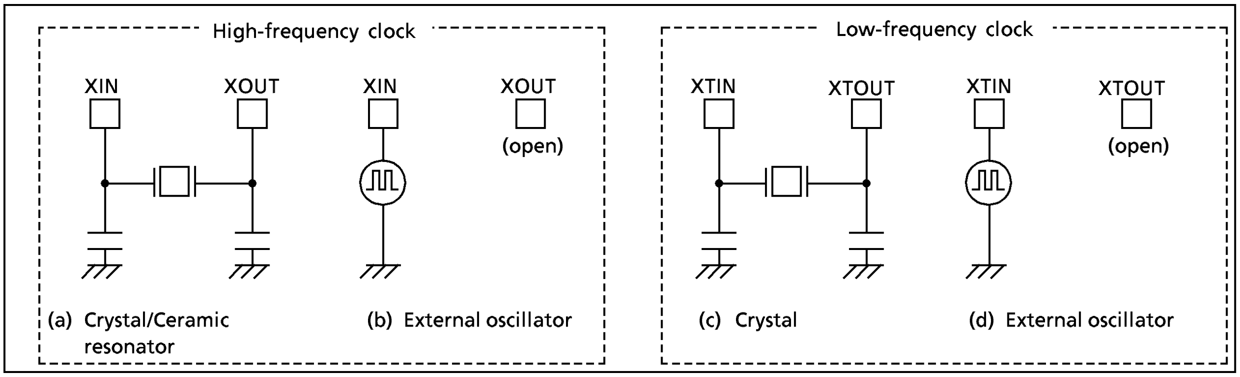


Figure 1-3. Examples of Resonator Connection

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.

1.5.2 Clock gear

The clock gear is a circuit to select a gear clock fcgck, that is the basis of the main system clock supplied to the timing generator, from high-frequency clock fc, or the divided clock fc/2, fc/4 or fc/8 and fc/16. Power consumption can be reduced by switching of the high-frequency from fc to fc/2, fc/4, fc/8 and fc/16 with the clock gear using.
The clock gear consists of a 4-stage prescaler with a multiplexer.

*Note: The **TMP86CH06** starts the reset release or initialization status from fc*

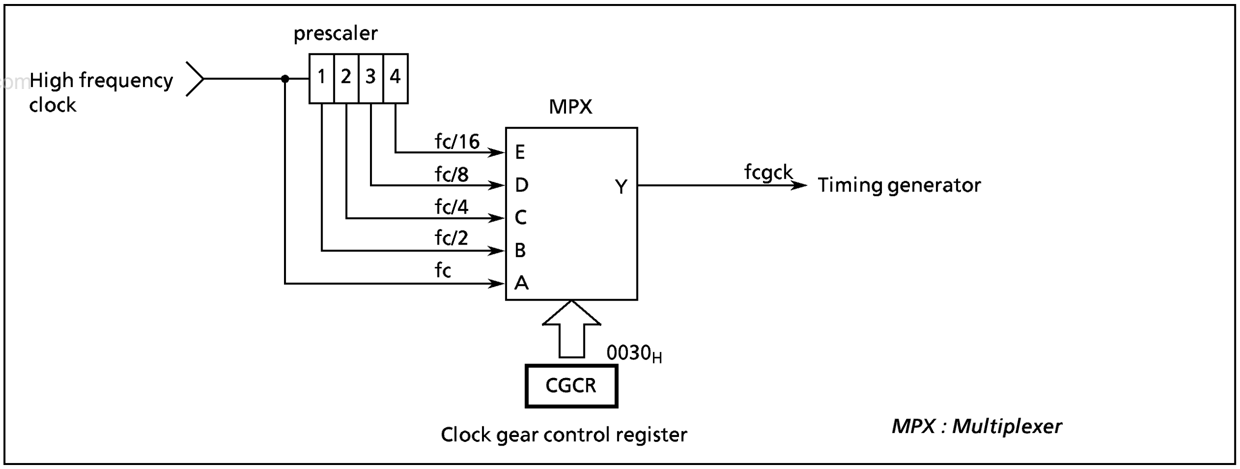


Figure 1-4. Configuration of Clock Gear

Clock Gear Control Register

	7	6	5	4	3	2	1	0	
CGCR (0030_H)	(DVCK)		(DV1CK)			FCGCK			(Initial value: 000∗ ∗000)

FCGCK	Gear clock selection	000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: Reserved 11∗: Reserved [Hz]	R/W

Note 1: fc : High-frequency clock, ∗ : Don't care
Note 2: Bit 4 and in CGCR are always read as undefined value.

Figure 1-5. Clock Gear Control Register

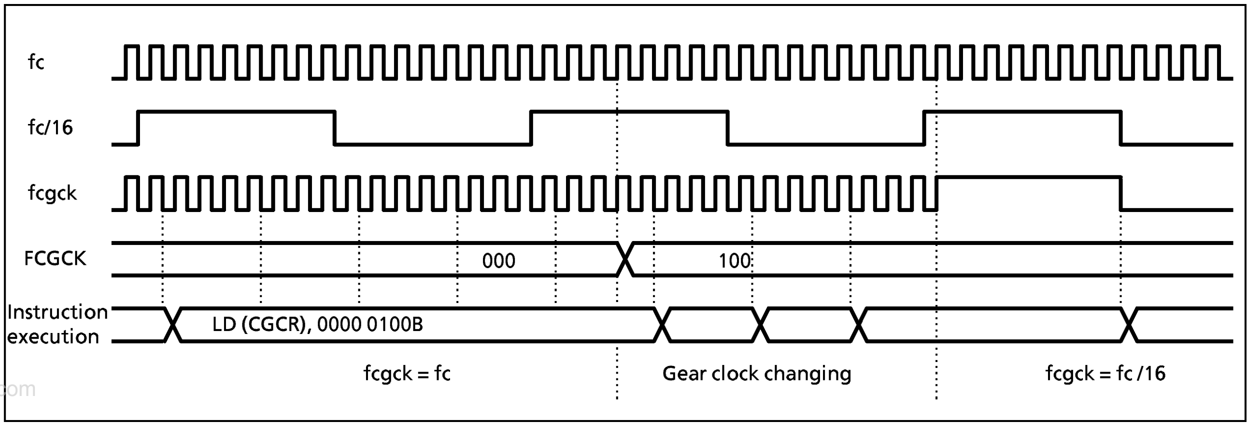


Figure 1-6. Example of Clock Exchangeable Timing by Clock Gear

1.5.3 Timing Generator

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

① Generation of main system clock (fm)
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

(1) Configuration of timing generator

The timing generator consists of a 3-stage prescaler, a 21-stage divider, a main system clock generator, and machine cycle counters.
The clock fc/4 or fc/8, that is output from the 2nd stage or the 3rd stage of the prescaler, can be selected as the clock to input to the 1st stage of the divider by DV1CK (bit 5 in CGCR). Inputting fc/8 to the 1st stage of the divider operates the peripheral circuit without the setting change when the operation clock is multiplied by 2. (Example: 8 MHz to 16 MHz)
Even if the main system clock is changed by the clock gear, the output from the divider is not changed.
However, the peripheral circuits utilizing high-speed divider output (first to fourth stage) cannot be used if the main system clock slows down on the clock gear. In such a case, setting DVCK (bit 7, 6 in CGCR) enables reducing speed, while high-speed operating parts remain their paces. DVCK should be set in accordance with the divider output providing for peripheral circuits and the system should be kept the minimum operating frequency. DVCK should be set before the peripheral circuits start operating. Do not change the set value after DVCK is set.
An input clock to the 7th stage of the divider depends on the operating mode, DV1CK (bit 5 in CGCR), and DV7CK (bit 4 in TBTCR), that is shown in Table 1-2. As reset and STOP mode started/canceled, the prescaler and the divider are cleared to “0”.

Table 1-1. Divider Output Capability

DVCK	Gear Clock frequency	Dinder Output Capability									
		DV1CK = 0					DV1CK = 1				
		DV1G	DV2G	DV3G	DV4	DV5	DV1G	DV2G	DV3G	DV4	DV5
00	fcgck = fc	$fc/2^{3}$	$fc/2^{4}$	$fc/2^{5}$	$fc/2^{6}$	$fc/2^{7}$	$fc/2^{4}$	$fc/2^{5}$	$fc/2^{6}$	$fc/2^{7}$	$fc/2^{8}$
		×	○	○	○	○	○	○	○	○	○
	fcgck = fc/2	×	×	○	○	○	×	○	○	○	○
	fcgck = fc/4	×	×	×	○	○	×	×	○	○	○
	fcgck = fc/8	×	×	×	×	○	×	×	×	○	○
	fcgck = fc/16	×	×	×	×	×	×	×	×	×	○
01	fcgck = fc	$fc/2^{4}$	$fc/2^{4}$	$fc/2^{5}$	$fc/2^{6}$	$fc/2^{7}$	$fc/2^{5}$	$fc/2^{5}$	$fc/2^{6}$	$fc/2^{7}$	$fc/2^{8}$
		○	○	○	○	○	○	○	○	○	○
	fcgck = fc/2	×	×	○	○	○	○	○	○	○	○
	fcgck = fc/4	×	×	×	○	○	×	×	○	○	○
	fcgck = fc/8	×	×	×	×	○	×	×	×	○	○
	fcgck = fc/16	×	×	×	×	×	×	×	×	×	○
10	fcgck = fc	$fc/2^{5}$	$fc/2^{5}$	$fc/2^{5}$	$fc/2^{6}$	$fc/2^{7}$	$fc/2^{6}$	$fc/2^{6}$	$fc/2^{6}$	$fc/2^{7}$	$fc/2^{8}$
		○	○	○	○	○	○	○	○	○	○
	fcgck = fc/2	○	○	○	○	○	○	○	○	○	○
	fcgck = fc/4	×	×	×	○	○	○	○	○	○	○
	fcgck = fc/8	×	×	×	×	○	×	×	×	○	○
	fcgck = fc/16	×	×	×	×	×	×	×	×	×	○
11	fcgck = fc	$fc/2^{6}$	$fc/2^{6}$	$fc/2^{6}$	$fc/2^{6}$	$fc/2^{7}$	$fc/2^{7}$	$fc/2^{7}$	$fc/2^{7}$	$fc/2^{7}$	$fc/2^{8}$
		○	○	○	○	○	○	○	○	○	○
	fcgck = fc/2	○	○	○	○	○	○	○	○	○	○
	fcgck = fc/4	○	○	○	○	○	○	○	○	○	○
	fcgck = fc/8	×	×	×	×	○	○	○	○	○	○
	fcgck = fc/16	×	×	×	×	×	×	×	×	×	○

Note: When selecting DV1G, DV2G, DV3G, DV4 or DV5 for the operating clock of timer, SIO, etc, check Table 1-1 to see that the divider output is possible (marked with ○). If the divider output is not possible (marked with ×), do not select them for the operating clock.

Table 1-2. Input Clock to 7th Stage of The Divider

Single-clock mode		Dual-clock mode			
NORMAL1, IDLE1 mode		NORMAL2, IDLE2 mode (SYSCK = 0)			SLOW, SLEEP mode (SYSCK = 1)
DV1CK = 0	DV1CK = 1	DV7CK = 0		DV7CK = 1	
		DV1CK = 0	DV1CK = 1		
$fc/2^{8}$	$fc/2^{9}$	$fc/2^{8}$	$fc/2^{9}$	fs	fs

Note 1: Do not set "1" on DV7CK while the low-frequency clock is not operated stably.

Note 2: In SLOW and SLEEP mode, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

Note 3: Do not set DV7CK to "1" when single clock mode is selected.

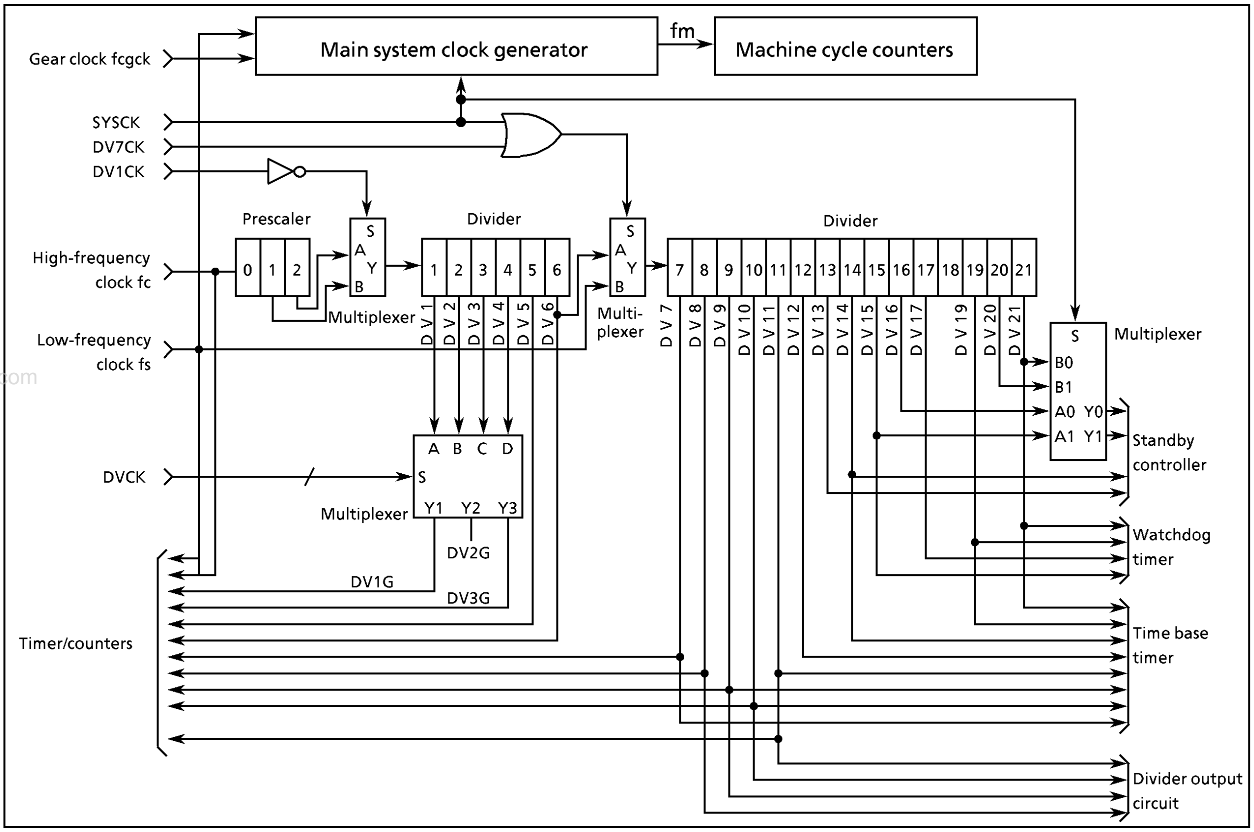


Figure 1-7. Configuration of Timing Generator

Table 1-3. Division Ratio of Divider

	DV7CK = 0		DV7CK = 1	
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1
DV1	fc/2^3	fc/2^4	fc/2^3	fc/2^4
DV2	fc/2^4	fc/2^5	fc/2^4	fc/2^5
DV3	fc/2^5	fc/2^6	fc/2^5	fc/2^6
DV4	fc/2^6	fc/2^7	fc/2^6	fc/2^7
DV5	fc/2^7	fc/2^8	fc/2^7	fc/2^8
DV6	fc/2^8	fc/2^9	fc/2^8	fc/2^9
DV7	fc/2^9	fc/2^{10}	fs/2	
DV8	fc/2^{10}	fc/2^{11}	fs/2^2	
DV9	fc/2^{11}	fc/2^{12}	fs/2^3	
DV10	fc/2^{12}	fc/2^{13}	fs/2^4	
DV11	fc/2^{13}	fc/2^{14}	fs/2^5	
DV12	fc/2^{14}	fc/2^{15}	fs/2^6	
DV13	fc/2^{15}	fc/2^{16}	fs/2^7	
DV14	fc/2^{16}	fc/2^{17}	fs/2^8	
DV15	fc/2^{17}	fc/2^{18}	fs/2^9	
DV16	fc/2^{18}	fc/2^{19}	fs/2^{10}	
DV17	fc/2^{19}	fc/2^{20}	fs/2^{11}	
DV18	fc/2^{20}	fc/2^{21}	fs/2^{12}	
DV19	fc/2^{21}	fc/2^{22}	fs/2^{13}	
DV20	fc/2^{22}	fc/2^{23}	fs/2^{14}	
DV21	fc/2^{23}	fc/2^{24}	fs/2^{15}	

	7	6	5	4	3	2	1	0	
CGCR (0030_H)	DVCK		DV1CK			(FCGCK)			(initial value: 000* *000)

DVCK	Selection for the divider output: DV1G to DV3G	DV1G DV2G DV3G 00: DV1 DV2 DV3 01: DV2 DV2 DV3 10: DV3 DV3 DV3 11: DV4 DV4 DV4	R/W
DV1CK	Selection of input clock to the 1st stage of the divider	0: fc/4 1: fc/8	

*Note 1: fc : High-frequency clock [Hz] * : Don't care*
Note 2: The bit 4 and 3 on CGCR are always read as undefined value.

Figure 1-8. Clock Gear Control Register

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

DV7CK	Selection of input to the 7th stage of the divider	0: fc/2^8 or fc/2^9 [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: During warming-up after STOP mode is released or in IDLE0 mode, output of the 6th stage of the divider is input to the 7th stage of the divider regardless of DV7CK.

Figure 1-9. Timing Generator Control Register

(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.

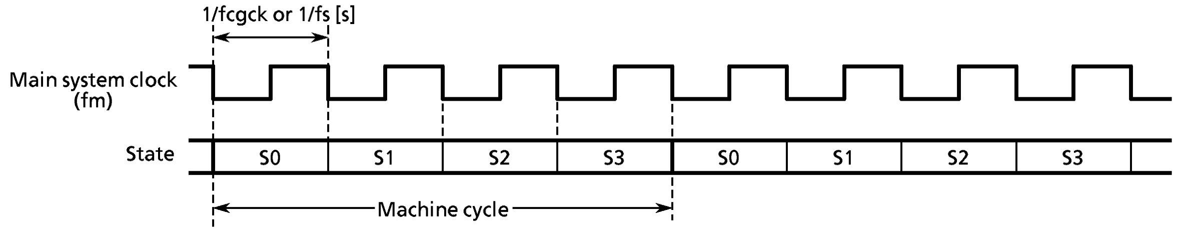


Figure 1-10. Machine Cycle

Table 1-4. Example of Machine Cycle

Frequency		Machine cycle				
		fcgck = fc	fcgck = fc/2	fcgck = fc/4	fcgck = fc/8	fcgck = fc/16
High-frequency clock	fc = 16.0 MHz	0.25 μs	0.5 μs	1.0 μs	2.0 μs	4.0 μs
	fc = 12.5 MHz	0.32 μs	0.64 μs	1.28 μs	2.56 μs	–
	fc = 4.2 MHz	0.95 μs	1.9 μs	3.8 μs	–	–
	fc = 1 MHz	4 μs	–	–	–	–
Low-frequency clock	fs = 32.8 kHz	122 μs				

Note: "–" is movement outside guaranteed range.

1.5.4 Standby Controller

The standby controller starts and stops the oscillation circuits for the high-frequency and low-frequency clocks, and switches the main system clock. There are two operating modes: single-clock and dual-clock. These modes are controlled by the system control registers (SYSCR1 and SYSCR2).
Figure 1-11 shows the operating mode transition diagram and Figure 1-12 shows the system control registers.

(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/fcgck [s].

① NORMAL1 mode
In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock.
The TMP86CH06 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 the system control register 2 (SYSCR2), 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 setting "1" on bit TGHALT on the system control register 2 (SYSCR2). As IDLE0 mode is enabled, the timing generator halts (the system clock halts) and the CPU and other peripherals are stopped.
As IDLE0 mode is enabled, the count on the Time-Base-Timer is cleared and the Time-Base-Timer restart counting. When the count reaches the rate on TBTCR, SLEEP0 releasing signal, which restarts the timing generator, is generated and the hardware restores NORMAL1 mode.

(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/fcgck [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 after reset is released. To use the dual-clock mode, the low-frequency oscillator should be turned on at the start of a program.

Note: With the TLCS-870/C Series, no option is provided to select the clock mode after release of reset. When reset is released, it becomes the single mode.

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. On-chip peripherals are triggered by the low-frequency clock. As the SYSCK on SYSCR2 becomes “0”, the hardware changes into NORMAL2 mode. As the XEN on SYSCR2 becomes “0”, the hardware changes into SLOW1 mode. Do not clear 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 XEN bit on the system control register 2 (SYSCR2). In SLOW1 and SLEEP1 mode, 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 SLEEP1 mode are the same as for IDLE1 mode, except that operation returns to SLOW1 mode. In SLOW1 and SLEEP1 mode, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

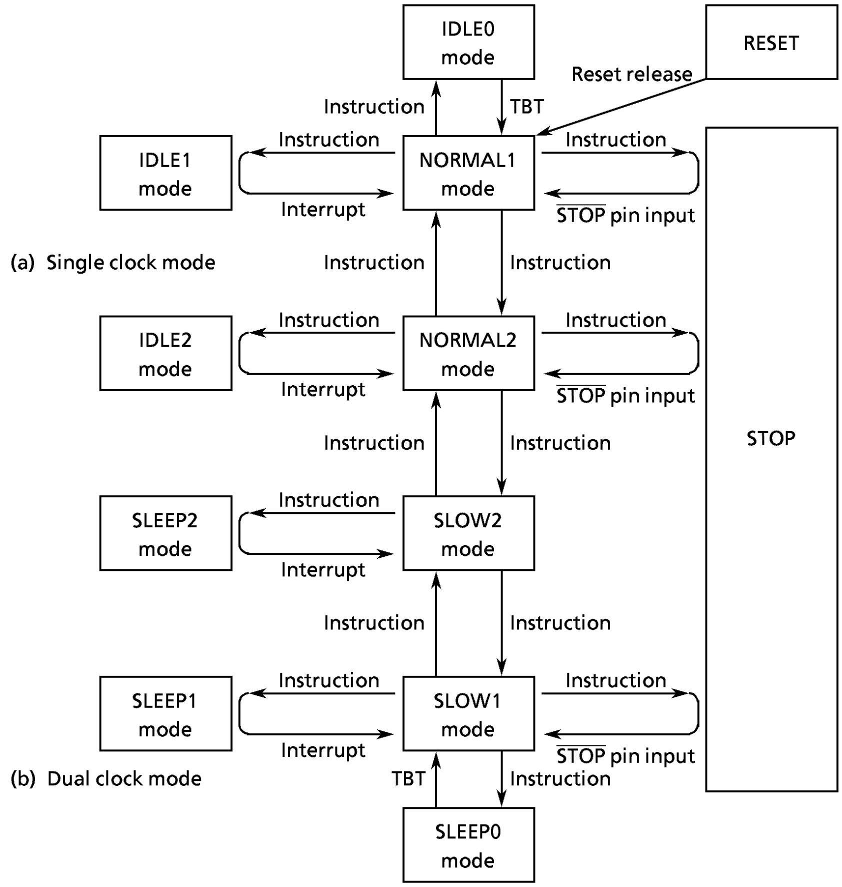
⑥ SLEEP2 mode
The SLEEP2 mode is the idle mode corresponding to the SLOW2 mode. The status under the SLEEP2 mode is same as that under the SLEEP1 mode, except for the oscillation circuit of the high-frequency clock.

⑦ SLEEP0 mode

In this mode, All the circuit, except oscillator and the Time-Base-Timer, stops operation.
This mode is enabled by setting "1" on bit TGHALT on the system control register2 (SYSCR2) during SLOW1. As SLEEP0 mode is enabled, the timing generator halts (the system clock halts) and the CPU and other peripherals are stopped.
As SLEEP0 mode is enabled, the count on the Time-Base-Timer is cleared and the Time-Base-Timer restart counting. When the count reaches the rate on TBTCR, SLEEP0 releasing signal, which restarts the timing generator, is generated and the hardware restores SLOW1 mode.

(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 warming-up period is completed, the execution resumes with the instruction which follows the STOP mode start instruction.



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

Operating Mode		Oscillator		CPU core	TBT	Other peripherals	Machine cycle time
		High-freq.	Low-freq.				
Single Clock	RESET	Oscilla-tion	Stop	Reset	Reset	Reset	4/fcgck [s]
	NORMAL1			Operate	Operate	Operate	
	IDLE1			Halt			
	IDLE0					Halt	
	STOP	Stop			Halt		—
Dual Clock	NORMAL2	Oscilla-tion	Oscilla-tion	Operate with High Freq.	Operate	Operate	4/fcgck [s]
	IDLE2			halt			
	SLOW2			Operate with Low Freq.			4/fs [s]
	SLEEP2			Halt			
	SLOW1	Stop		Operate with Low Freq.			
	SLEEP1			Halt			
	SLEEP0					Halt	
	STOP		Stop		Halt		—

Note: After reset is released, it becomes the single clock mode.

Figure 1-11. Operating Mode Transition Diagram

System Control Register 1

SYSCR1 (0038_H)

7	6	5	4	3	2	1	0
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 ($\overline{\text{STOP}}$ pin) 1: Level-sensitive release ($\overline{\text{STOP}}$ pin)	
RETM	Operating mode after STOP mode	0: Return to NORMAL mode 1: return to SLOW mode	
OUTEN	Port output during STOP mode	0: High Impedance 1: Output kept	
WUT	Warming-up time at releasing STOP mode		

	Return to NORMAL mode		Return to SLOW mode
	DV1CK = 0	DV1CK = 1	
00	$3 \times 2^{16}/fc$	$3 \times 2^{17}/fc$	$3 \times 2^{13}/fs$
01	$2^{16}/fc$	$2^{17}/fc$	$2^{13}/fs$
10	$3 \times 2^{14}/fc$	$3 \times 2^{15}/fc$	$3 \times 2^{6}/fs$
11	$2^{14}/fc$	$2^{15}/fc$	$2^{6}/fs$

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 interrupt request on account of falling edge.

Note 6: As the hardware becomes STOP mode under OUTEN = "0" from external bus mode, P17, P16, P15 keep their output and they do not become high impedance.

Note 7: The P20 pin is shared with the STOP pin, so that when it enters STOP mode, its output goes to a high-impedance state regardless of how OUTEN is set.

System Control Register 2

SYSCR2 (0039_H)

7	6	5	4	3	2	1	0
XEN	XTEN	SYSCK	IDLE		TGHALT		

(Initial value: 1000 *0**)

XEN	High-frequency oscillator control	0: Turn off oscillation 1: Turn on oscillation	R/W
XTEN	Low-frequency oscillator control	0: Turn off oscillation 1: Turn on oscillation	
SYSCK	Main system clock select (write)/main system clock monitor (read)	0: High-frequency clock (NORMAL1/NORMAL2/IDLE1/IDLE2) 1: Low-frequency clock (SLOW/SLEEP)	
IDLE	IDLE mode start	0: CPU and watchdog timer remain active 1: CPU and watchdog timer are stopped (start IDLE mode)	
TGHALT	TG stops (SLEEP0, IDLE0 mode)	0: TG keeps its operation 1: TG stops (SLEEP0, IDLE0 mode)	

Note 1: A reset is applied ($\overline{\text{RESET}}$ pin output goes low) 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 = "0".

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

Note 3: Bits 3, 1and 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".

Figure 1-12. System Control Registers

1.5.5 Operating Mode Control

(1) STOP mode

STOP mode is controlled by the system control register 1 (SYSCR1) and the $\overline{\text{STOP}}$ pin input. 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 STOP (bit 7 in SYSCR1) 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 release mode and an edge-sensitive release mode, either of which can be selected with the RELM (bit 6 in SYSCR1).

Note: During STOP period (from start of STOP mode to end of warming-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.

a. Level-sensitive release mode (RELM = "1")

In this mode, STOP mode is released by setting the $\overline{\text{STOP}}$ pin high. This mode is used for capacitor back-up when the main power supply is cut off and long term battery back-up.
When the $\overline{\text{STOP}}$ pin input is high, executing an instruction which starts STOP mode will not place in STOP mode but instead will immediately start the release sequence (warm-up). 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. The following two methods can be used for confirmation.

① Testing a port P20.
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  (P2R). 0                 ;  Wait until the STOP pin input goes low level
          JRS   F, SSTOPH
          DI
          SET   (SYSCR1). 7           ;  Starts STOP mode
```

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

```
PINT5:    TEST  (P2R). 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
          SET   (SYSCR1). 7           ;  Starts STOP mode
SINT5:    RETI
```

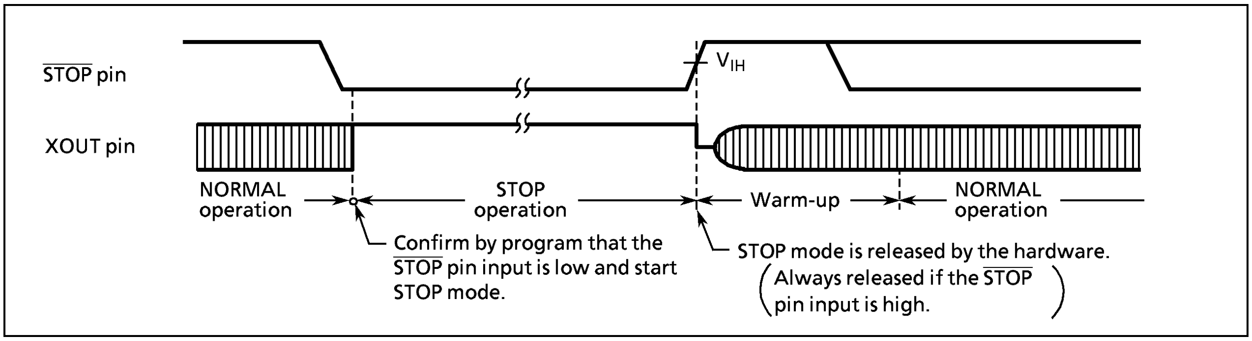

$\overline{\text{STOP}}$ pin
XOUT pin
V_{IH}
NORMAL operation
STOP operation
Warm-up
NORMAL operation
Confirm by program that the $\overline{\text{STOP}}$ pin input is low and start STOP mode.
STOP mode is released by the hardware.
(Always released if the $\overline{\text{STOP}}$ pin input is high.)

Figure 1-13. Level-sensitive release mode

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

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

b. 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, $\overline{\text{STOP}}$ mode is started even when the $\overline{\text{STOP}}$ pin input is high level.

Example: Starting STOP mode from NORMAL mode

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

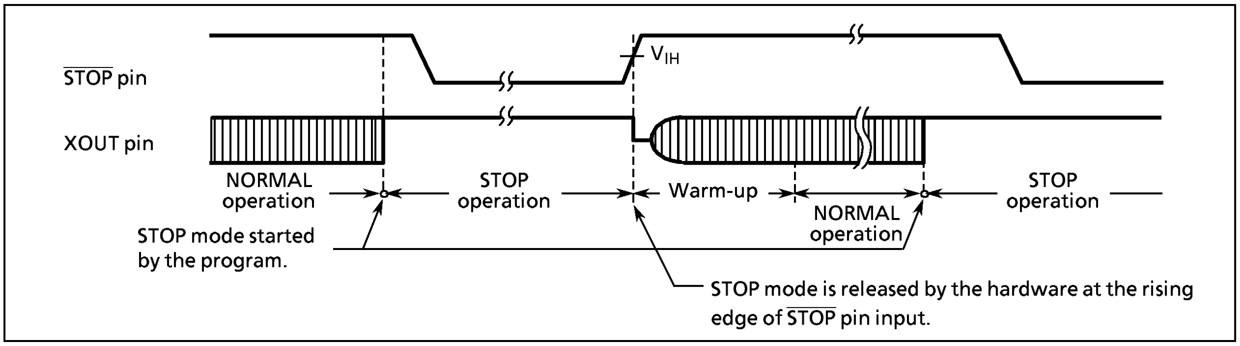

$\overline{\text{STOP}}$ pin
XOUT pin
V_{IH}
NORMAL operation
STOP operation
Warm-up
NORMAL operation
STOP operation
STOP mode started by the program.
STOP mode is released by the hardware at the rising edge of $\overline{\text{STOP}}$ pin input.

Figure 1-14. Edge-sensitive release mode

STOP mode is released by the following sequence.

① In the dual-clock mode, when returning to NORMAL2 or SLOW2, 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 signal-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 WUT (bits 2 and 3 in SYSCR1) 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. The start is made after the prescaler and the divider of the timing generator are cleared to "0".

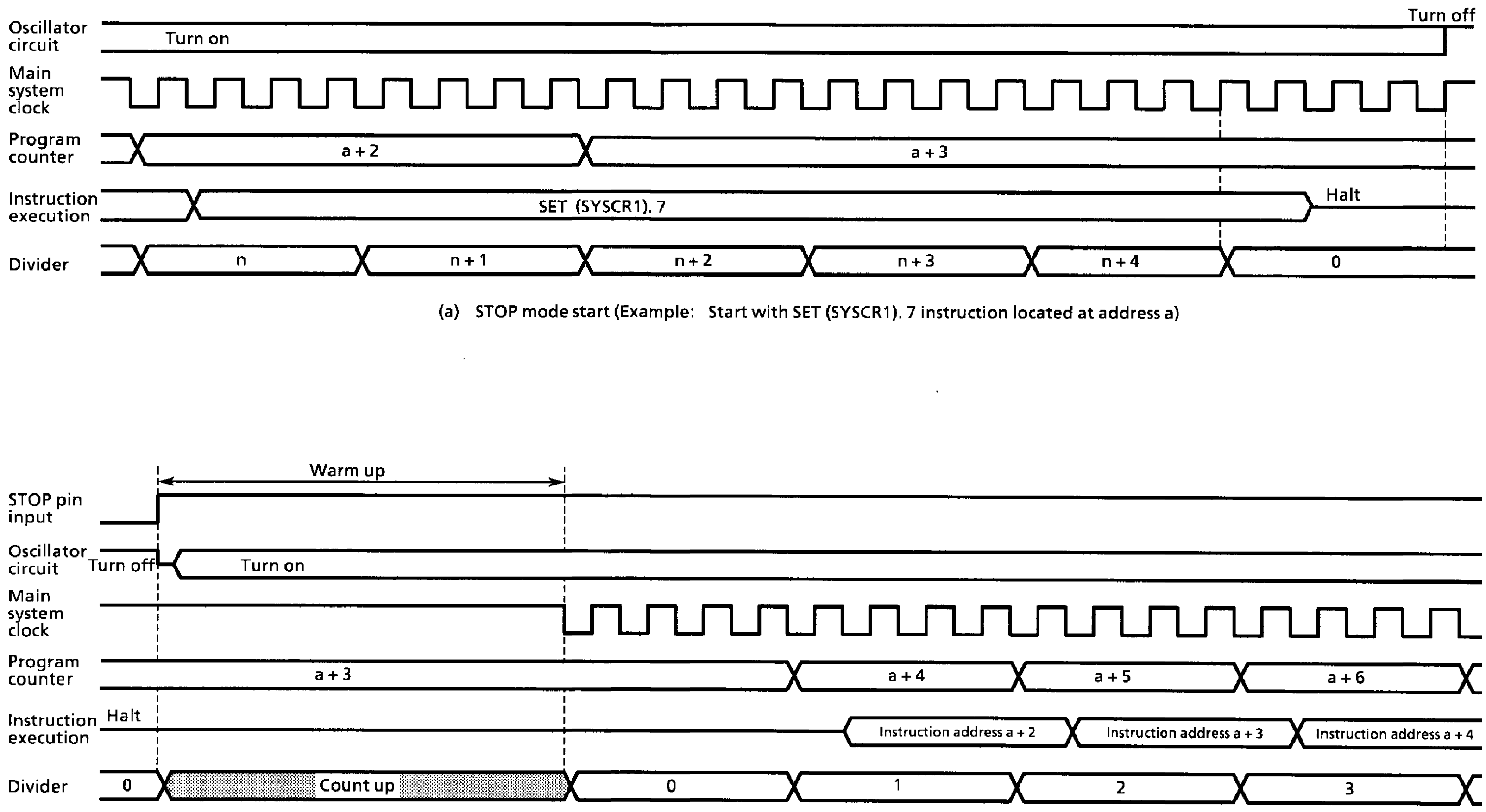
Table 1-5. Warm-up Time Example (at fc = 16.0 MHz, fs = 32.768 kHz)

WUT	Warm-up Time [ms]		
	Return to NORMAL mode		Return to SLOW mode
	DV1CK = 0	DV1CK = 1	
00	12.288	24.576	750
01	4.096	8.192	250
10	3.072	6.144	5.9
11	1.024	2.048	2.0

Note: 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 an approximate value.

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

Note: 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).



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

(b) STOP mode release

Figure 1-15. STOP Mode Start/Release (DV1CK = 0)

(2) IDLE mode (IDLE1, IDLE2, SLEEP1, SLEEP2)

IDLE mode is controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following status is maintained during IDLE mode.

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 IDLE mode was entered.
③ The program counter holds the address 2 ahead of the instruction which starts IDLE mode.

```
Example:    Starting IDLE mode.
                SET    (SYSCR2). 4          ;   IDLE ← 1
```

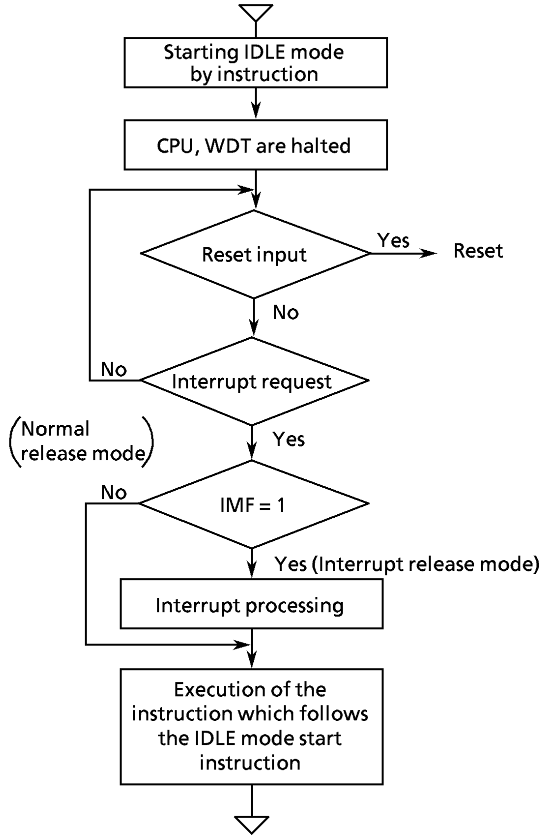

Starting IDLE mode by instruction
CPU, WDT are halted
Reset input
Yes
Reset
No
Interrupt request
No
Yes
(Normal release mode)
IMF = 1
No
Yes (Interrupt release mode)
Interrupt processing
Execution of the instruction which follows the IDLE mode start instruction

Figure 1-16. IDLE Mode

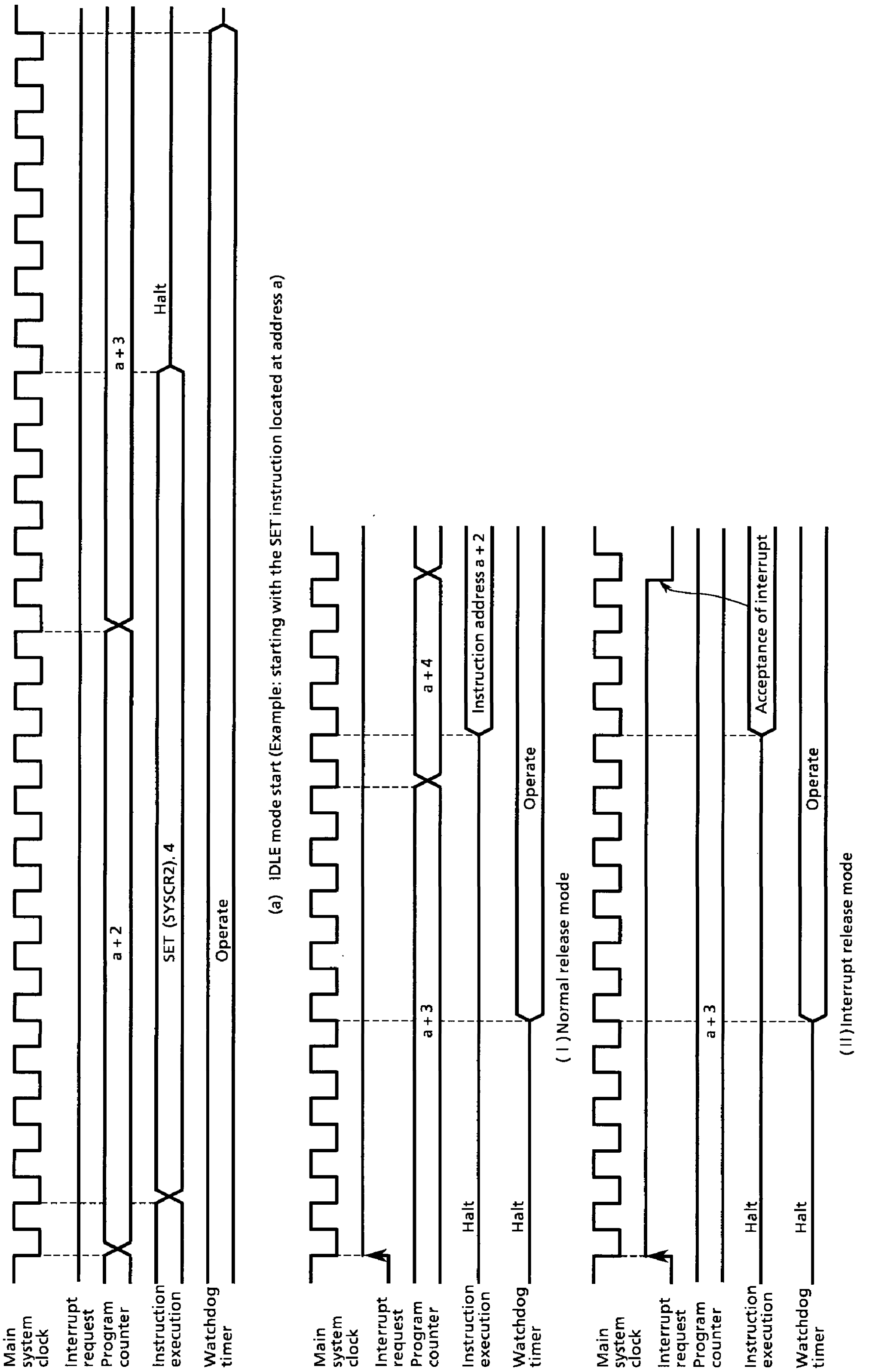
IDLE mode includes a normal release mode and an interrupt release mode. Selection is made with the interrupt master enable flag (IMF). Releasing IDLE mode returns from IDLE1 to NORMAL1, from IDLE2 to NORMAL2, from SLEEP1 to SLOW1 mode, and from SLEEP2 to SLOW2 mode.

(Ⅰ) Normal release mode (IMF = "0")
IDLE mode is released by any interrupt source enabled by the individual interrupt enable flag (EF). Execution resumes with the instruction following the IDLE mode start instruction . The interrupt latches (IL) of the interrupt source used for releasing must be cleared to "0" by load instructions. However, only TBT interrupt can release IDLE0/SLEEP0 mode.

(Ⅱ) Interrupt release mode (IMF = "1")
IDLE mode is released and interrupt processing is started by any interrupt source enabled with the individual interrupt enable flag (EF). After the interrupt is processed, the execution resumes from the instruction following the instruction which starts IDLE mode. However, only TBT interrupt can release IDLE0/SLEEP0 mode.

IDLE mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin, which immediately performs the reset operation. After reset, the TMP86CH06 is placed in NORMAL 1 mode.

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



(a) IDLE mode start (Example: starting with the SET instruction located at address a)

(I) Normal release mode

(II) Interrupt release mode

(b) IDLE mode release

Figure 1-17. IDLE Mode Start/Release

(3) IDLE0/SLEEP0 mode (IDLE0,SLEEP0)

IDLE0/SLEEP0 mode is controlled by the system control register 2 (SYSCR2) and TBT. The following status is maintained during IDLE0 mode/SLEEP0 mode.

① All the circuit, except oscillator and the TBT, stops operation.
② The data memory, CPU registers, program status word and port output latches are all held in the status in effect before IDLE0/SLEEP0 mode was entered.
③ The program counter holds the address 2 ahead of the instruction which starts IDLE0/SLEEP0 mode.

Example: Starting IDLE mode.

```
         SET   (SYSCR2) . 2
```

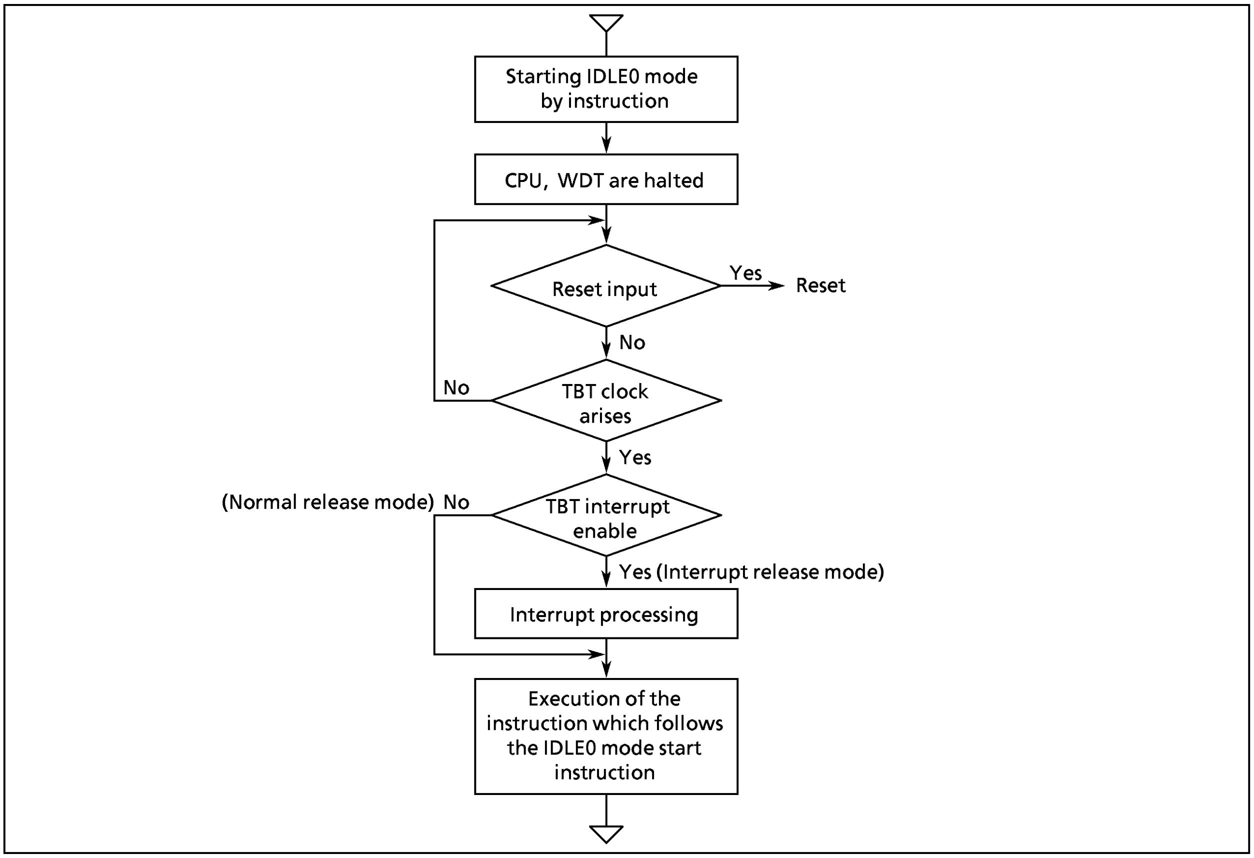
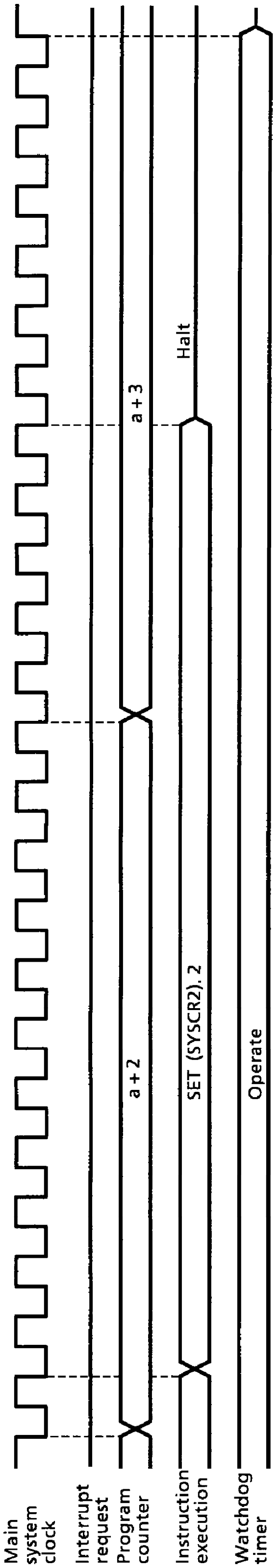

Starting IDLE0 mode by instruction
CPU, WDT are halted
Reset input
Yes
Reset
No
TBT clock arises
No
Yes
(Normal release mode) No
TBT interrupt enable
Yes (Interrupt release mode)
Interrupt processing
Execution of the instruction which follows the IDLE0 mode start instruction

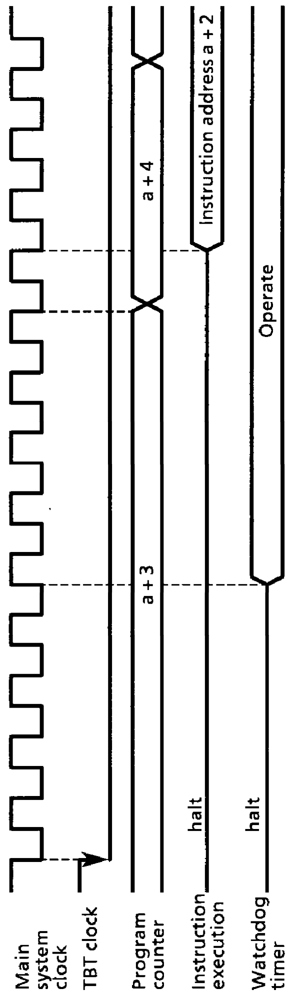
Figure 1-18. IDLE mode

When the count reaches the rate stated on TBTCR after starting IDLE0/SLEEP0 mode, this mode is released, and the device restore the operation.
On restarting, the interrupt processing is accepted if the TBT interrupt is enable. If its interrupt is disable, the instruction, which follows the instruction to enable IDLE0 mode, is executed.
IDLE0 mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin, which immediately performs the reset operation. After reset, the TMP86CH06 is placed in NORMAL1 mode.

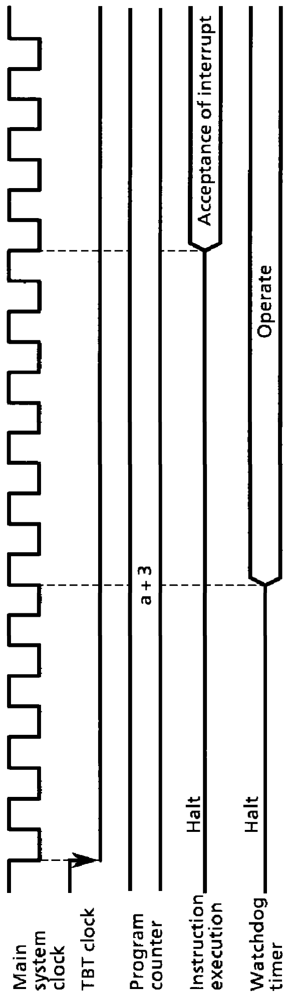
Note: When a watchdog timer interrupt is generated immediately before IDLE0/SLEEP0 mode is started, the watchdog timer interrupt will be processed but IDLE0/SLEEP0 mode will not be started.



(a) IDLE0, SLEEP0 mode start (Example: starting with the SET instruction located at address a)



① Normal release mode



② Interrupt release mode

(b) IDLE0, SLEEP0 mode release

Figure 1-19. IDLE0, SLEEP0 Mode Start/Release

(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 warming-up counter (TC1, 0).

a. Switching from NORMAL2 mode to SLOW1 mode

First, set SYSCK (bit 5 in SYSCR2) to switch the main system clock to the low-frequency clock for SLOW2 mode.
Next, clear XEN (bit 7 in SYSCR2) to turn off high-frequency oscillation.

> *Note: The high-frequency clock oscillation can be continued to return quickly to NORMAL2 mode. However, halt high frequency clock in order to enable STOP mode from SLOW mode.*

When the low-frequency clock oscillation is unstable, wait until oscillation stabilizes before performing the above operations. The timer/counter 1, 0 (TC1, TC0) can conveniently be used to confirm that low-frequency clock oscillation has stabilized.

Example1: Switching from NORMAL2 mode to SLOW1 mode.

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

            CLR   (SYSCR2). 7         ;  XEN ← 0
                                         (turns off high-frequency oscillation)
```

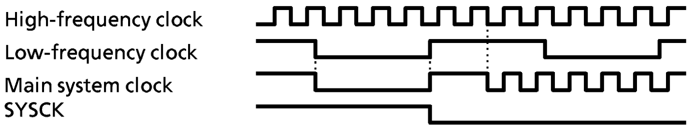
Example2: Switching to the SLOW1 mode after low-frequency clock has stabilized.

```
            LD    (TC0CR), 43H        ;  Sets mode for TC0, TC1 (16-bit TC, fs for source)

            LD    (TC1CR), 05H

            LDW   (TTREG0), 8000H     ;  Sets warming-up time
                                         (depend on oscillator accompanied)

            DI

            SET   (EIRL). 7           ;  Enables INTTC1

            EI

            SET   (TC1CR). 3          ;  Starts TC1, 0
              ⋮

  PINTTC1:  CLR   (TC1CR). 3          ;  Stops TC1, 0
            SET   (SYSCR2). 5         ;  SYSCK ← 1 (Switches the main system clock to the
                                         low-frequency clock)

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

            RETI
              ⋮

  VINTTC1:       DW    PINTTC1        ;  INTTC1 vector table
```

b. Switching from SLOW1 mode to NORMAL2 mode

First, set XEN (bit 7 in SYSCR2) to turn on the high-frequency oscillation. When time for stabilization (warm-up) has been taken by the timer/counter 1, 0 (TC1, 0), clear SYSCK (bit 5 in SYSCR2) to switch the main system clock to the high-frequency clock.

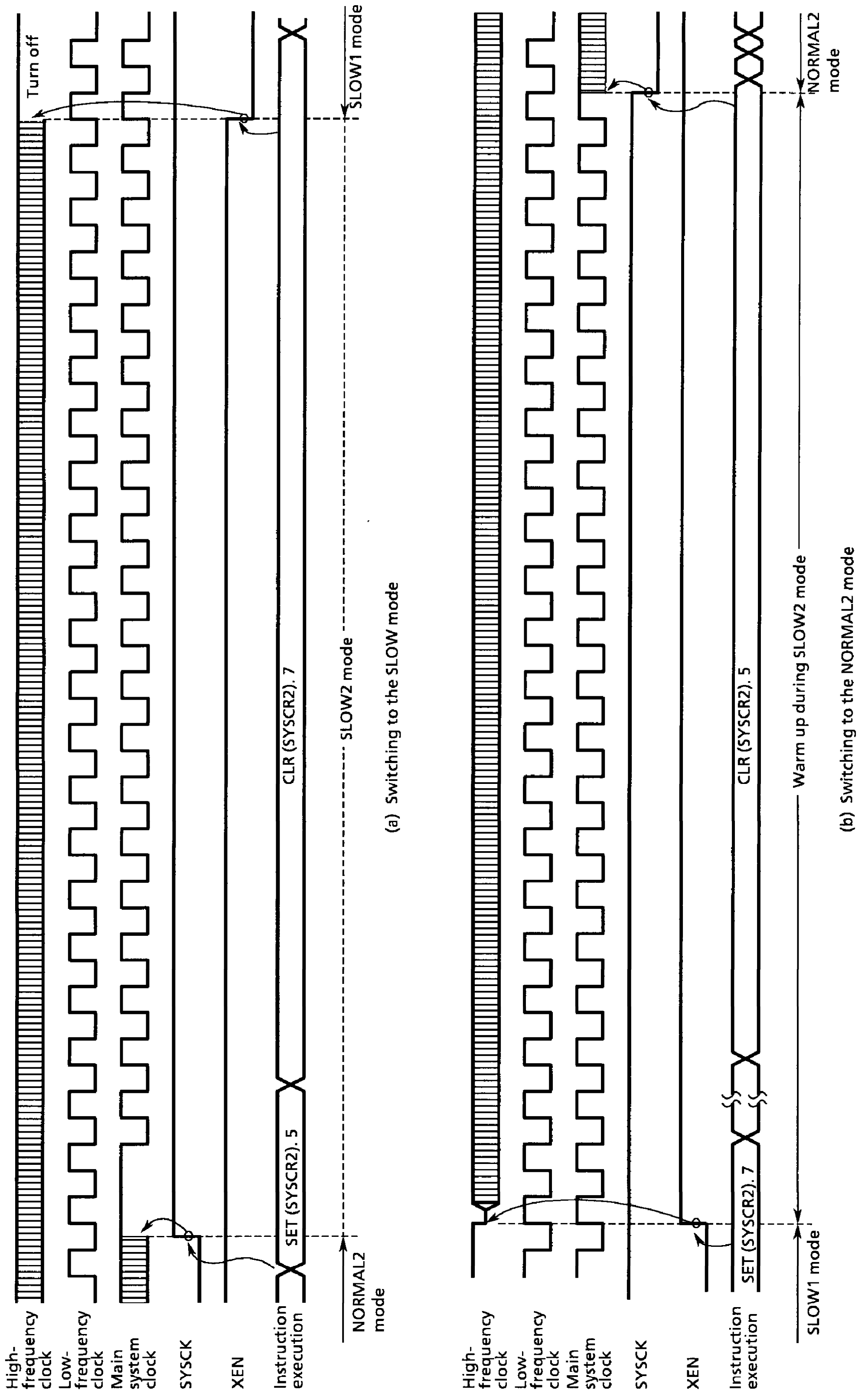
Note 1: After SYSCK is cleared to "0", executing the instructions is continued by the low-frequency clock for the period synchronized with low-frequency and high-frequency clocks.



Note 2: SLOW mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin, which immediately performs the reset operation. After reset, the TMP86CH06 is placed in NORMAL1 mode.

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

```
          SET  (SYSCR2). 7       ;  XEN ← 1 (Starts high-frequency oscillation)
          LD   (TC0CR), 63H      ;  Sets mode for TC0, TC1 (16-bit TC, fc for source)
          LD   (TC1CR), 05H

          LD   (TTREG1), 0F8H    ;  Sets warming-up time

          DI

          SET  (EIRL). 7         ;  Enables INTTC1

          EI

          SET  (TC1CR). 3        ;  Starts TC1, 0
           :

PINTTC1:  CLR  (TC1CR). 3        ;  Stops TC1, 0

          CLR  (SYSCR2). 5       ;  SYSCR←0 (Switches the main system clock to the
                                    high-frequency clock)

          RETI
           :

VINTTC1:     DW   PINTTC1        ;  INTTC1 vector table
```



(a) Switching to the SLOW mode

(b) Switching to the NORMAL2 mode

Figure 1-20. Switching between the NORMAL2 and SLOW Modes

1.6 Interrupt Control Circuit

The TMP86CH06 has a total (Reset is excluded) of 21 interrupt sources; 5 of the sources are multiplexed. Multiple interrupt with priorities is available. 4 of the internal factors are non-maskable interrupts, and the rest of them are maskable interrupts.

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.

Table 1-6. Interrupt Sources

Interrupt Factors		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/ External	(Reset)	Non-maskable	—	FFFE$_H$	High 1
Internal	INTSW (Software interrupt)	Non-maskable	—	FFFC$_H$	2
Internal	INTUNDEF (Executed the Undefined Instruction interrupt)	Non-maskable	—	FFFC$_H$	2
Internal	INTATRAP (Address Trap interrupt)	Non-maskable	IL$_2$	FFFA$_H$	2
Internal	INTWDT (Watchdog Timer interrupt)	Non-maskable	IL$_3$	FFF8$_H$	2
External	INT0 (External interrupt0)	IMF = 1, EF$_4$ = 1, INT0EN = 1	IL$_4$	FFF6$_H$	5
External	INT1 (External interrupt1)	IMF = 1, EF$_5$ = 1	IL$_5$	FFF4$_H$	6
Internal	INTTBT (Time Base Timer interrupt)	IMF = 1, EF$_6$ = 1	IL$_6$	FFF2$_H$	7
Internal	INTTC1 (8-bit Timer1 (16-bit mode) interrupt)	IMF = 1, EF$_7$ = 1	IL$_7$	FFF0$_H$	8
Internal, Internal	INTRXD0 (UART (ch0) Received interrupt), INTSIO (SIO interrupt)	IMF = 1, EF$_8$ = 1	IL$_8$	FFEE$_H$	9
Internal	INTTXD0 (UART (ch0) Transmitted interrupt)	IMF = 1, EF$_9$ = 1	IL$_9$	FFEC$_H$	10
Internal	INTET0 (ETC0 Full-Count interrupt)	IMF = 1, EF$_{10}$ = 1	IL$_{10}$	FFEA$_H$	11
Internal	INTIC0 (ETC0 Input Capture interrupt)	IMF = 1, EF$_{11}$ = 1	IL$_{11}$	FFE8$_H$	12
Internal, External	INTOC0 (ETC0 Output Compare interrupt), INT2 (External interrupt2)	IMF = 1, EF$_{12}$ = 1	IL$_{12}$	FFE6$_H$	13
Internal, External	INT3 (External interrupt3), INTRXD1 (UART (ch1) Received interrupt)	IMF = 1, EF$_{13}$ = 1	IL$_{13}$	FFE4$_H$	14
Internal, External	INT4 (External interrupt4), INTTXD1 (UART (ch1) Transmitted interrupt)	IMF = 1, EF$_{14}$ = 1	IL$_{14}$	FFE2$_H$	15
Internal, External	INTTC0 (8-bit Timer0 interrupt), INT5 (External interrupt5)	IMF = 1, EF$_{15}$ = 1	IL$_{15}$	FFE0$_H$	Low 16

Note 1: The following interrupt factors share their interrupt source; the factor is selected on the register INTSEL.

1) *INTRXD0 and INTSIO share the source whose priority is 9.*
2) *INTOC0 and INT2 share the source whose priority is 13.*
3) *INT3 and INTRXD1 share the source whose priority is 14.*
4) *INT4 and INTTXD1 share the source whose priority is 15.*
5) *INTTC0 and INT5 share the source whose priority is 16.*

Note 2: 2 alternatives are to be chosen in case INTATRAP (Address Trap interrupt) is executed: interrupt or reset. (for detail, see 1.6.4 Address trap interrupt (INTATRAP)

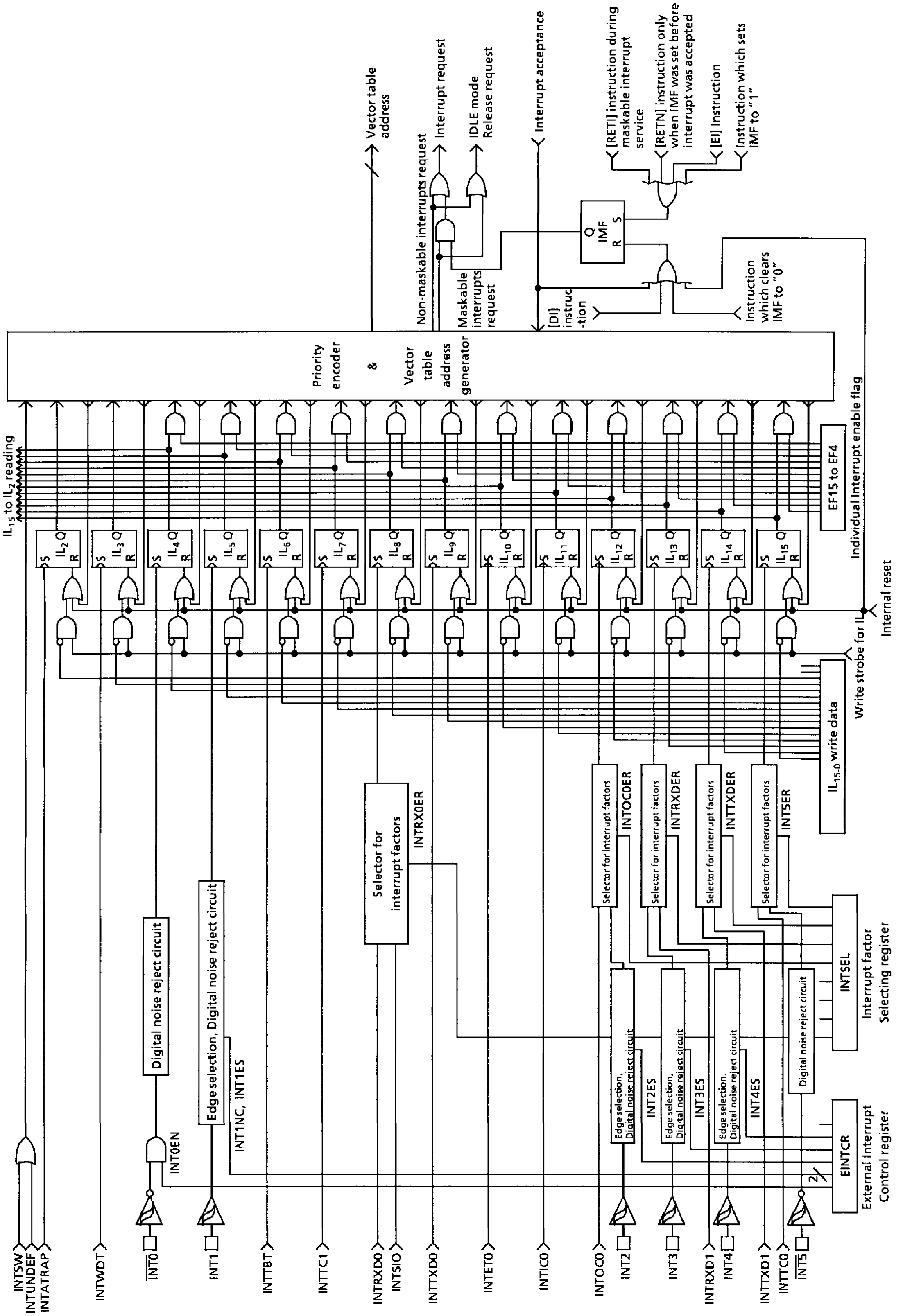

INTSW
INTUNDEF
INTATRAP
INTWDT
INT0
INT0EN
Digital noise reject circuit
INT1
Edge selection, Digital noise reject circuit
INT1NC, INT1ES
INTTBT
INTTC1
INTRXD0
INTSIO
Selector for interrupt factors
INTRX0ER
INTTXD0
INTET0
INTIC0
INTOC0
INT2
INT3
INTRXD1
INT4
INTTXD1
INTTC0
INT5
Edge selection, Digital noise reject circuit
INT2ES
INT3ES
INT4ES
Digital noise reject circuit
EINTCR
External Interrupt Control register
2
Selector for interrupt factors
INTOC0ER
INTRXDER
INTTXDER
INT5ER
INTSEL
Interrupt factor Selecting register
IL15-0 write data
Write strobe for IL
Internal reset
EF15 to EF4
Individual Interrupt enable flag
IL15 to IL2 reading
S IL2 Q R
S IL3 Q R
S IL4 Q R
S IL5 Q R
S IL6 Q R
S IL7 Q R
S IL8 Q R
S IL9 Q R
S IL10 Q R
S IL11 Q R
S IL12 Q R
S IL13 Q R
S IL14 Q R
S IL15 Q R
Priority encoder & Vector table address generator
Vector table address
Non-maskable interrupts request
Interrupt request
IDLE mode Release request
Maskable interrupts request
Interrupt acceptance
[DI] instruc-tion
Instruction which clears IMF to "0"
Q IMF R S
[RETI] instruction during maskable interrupt service
[RETN] instruction only when IMF was set before interrupt was accepted
[EI] Instruction
Instruction which sets IMF to "1"

Figure 1-21. Interrupt Controller Block Diagram

(1) Interrupt Latches (IL_{15} to IL_2)

An interrupt latch is provided for each interrupt source, except for a software 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. All interrupt latches are initialized to “0” during reset.

The interrupt latches are located on address $003C_H$ and $003D_H$ in SFR area. Except for IL_3 and IL_2, each latch can be cleared to “0” individually by instruction (However, the read-modify-write instructions such as bit manipulation or operation instructions cannot be used. Interrupt request would be cleared inadequately if interrupt is requested while such instructions are executed.). Thus interrupt request can be canceled /initialized by software.

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: Before you change each enable flag (EF) and/or each interrupt latch (IL), be sure to clear the interrupt master enable flag IMF) to “0” (to disable interrupts).
a. After a DI instruction is executed.
b. When an interrupt is accepted, IMF is automatically cleared to “0”. However, to enable nested interrupts, change EF and/or IL before setting IMF to “1” (to enable interrupts).
If the individual enable flags (EF) and interrupt latches (IL) are set under conditions other than the above, the proper operation cannot be guaranteed.

Example 1: Clears interrupt latches

```
DI                                  ; IMF←0
LDW     (ILL), 1110100000111111B    ; IL12, IL10 to IL6←0
LD      (ILH), 11111000B            ; IL10 to IL8←0
EI                                  ; IMF←1
```

Example 2: Reads interrupt latches

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

Example 3: Tests an interrupt latches

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

(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 $003C_H$ and $003D_H$ 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).

Note: Before you change each enable flag (EF) and/or each interrupt latch (IL), be sure to clear the interrupt master enable flag IMF) to “0” (to disable interrupts).
a. After a DI instruction is executed.
b. When an interrupt is accepted, IMF is automatically cleared to “0”. However, to enable nested interrupts, change EF and/or IL before setting IMF to “1” (to enable interrupts).
If the individual enable flags (EF) and interrupt latches (IL) are set under conditions other than the above, the proper operation cannot be guaranteed.

a) 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: $003A_H$ 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", and maskable interrupts are not accepted until it is set to "1".

b) Individual interrupt enable flags (EF_{15} to EF_4)

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. The individual interrupt enable flags (EF_{15} to EF_4) are located on EIRL to EIRH (address: $003A_H$ to $003B_H$ in SFR), and can be read and written by an instruction. During reset, all the individual interrupt enable flags (EF_{15} to EF_4) are initialized to "0" and all maskable interrupts are not accepted until they are set to "1".

Example 1: Sets individual interrupt enable flags

```
DI                                    ; IMF ← 0
LDW   (EIRL), 1110100010100000B       ; EF15 to 13, EF11, EF7, EF5 ← 1
                                        Note: Do not set IMF
 ⋮
EI                                    ; IMF ← 1
```

Example 2: Example of description in C

```
unsigned int  __io (3AH)   EIRL ;       /* 3AH: address for EIRL */

__DI ( ) ;

EIRL = 10100000B ;

 ⋮

__EI ( ) ;
```

Interrupt Latches

IL (003C, 003D$_H$)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IL_{15}	IL_{14}	IL_{13}	IL_{12}	IL_{11}	IL_{10}	IL_9	IL_8	IL_7	IL_6	IL_5	IL_4	IL_3	IL_2		

IL_H (003D$_H$) IL_L (003C$_H$)

(Initial value: 00000000 000000**)

		At RD	At WR	
IL_{15} to IL_2	Interrupt Latches	0: No interrupt request 1: Interrupt request	0: Clears the interrupt request (Note 1) 1: (Interrupt Latch is not set.)	R/W

Note 1: IL2 and IL3 are prohibited from clearing.
Note 2: Before you change each enable flag (EF) and/or each interrupt latch (IL), be sure to clear the interrupt master enable flag IMF) to "0" (to disable interrupts).
a. After a DI instruction is executed.
b. When an interrupt is accepted, IMF is automatically cleared to "0". However, to enable nested interrupts, change EF and/or IL before setting IMF to "1" (to enable interrupts).
If the individual enable flags (EF) and interrupt latches (IL) are set under conditions other than the above, the proper operation cannot be guaranteed.

Interrupt Enable Registers

EIR (003A, 003B$_H$)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_9	EF_8	EF_7	EF_6	EF_5	EF_4				IMF

EIR_H (003B$_H$) EIR_L (003A$_H$)

(Initial value: 00000000 0000***0)

Note 1: Do not clear IL with read-modify-write instructions such as bit operations
*Note 2: * : Don't care*
Note 3: Before you change each enable flag (EF) and/or each interrupt latch (IL), be sure to clear the interrupt master enable flag IMF) to "0" (to disable interrupts).
a. After a DI instruction is executed.
b. When an interrupt is accepted, IMF is automatically cleared to "0". However, to enable nested interrupts, change EF and/or IL before setting IMF to "1" (to enable interrupts).
If the individual enable flags (EF) and interrupt latches (IL) are set under conditions other than the above, the proper operation cannot be guaranteed.
Note 4: Do not set IMF to 1 simultaneously with EF.

Figure 1-22 (1). Interrupt Latch (IL) , Interrupt Enable Registers (EIR)

(3) Selecting interrupt factor (INTSEL)

Each interrupt factor, that shares its interrupt source with other factors, enables its interrupt latch (IL) only if it is selected on INTSEL. The interrupt controller does not hold the interrupt request, while the factor generates the interrupt request is not selected on INTSEL. Therefore, set INTSEL appropriately before interrupt factors arises.

Interrupt Selecting Register

INTSEL (003E$_H$)

7	6	5	4	3	2	1	0
INTRX0ER	–	–	–	INTOC0ER	INTRX1ER	INTTXDER	INT5ER

(Initial value: 0*** 0000)

INTRX0ER	Alternative of INTRXD0 or INTSIO	0: INTRXD0 1: INTSIO	R/W
INTOC0ER	Alternative of INTOC0 or INT2	0: INTOC0 1: INT2	
INTRX1ER	Alternative of INT3 or INTRXD1	0: INT3 1: INTRXD1	
INTTXDER	Alternative of INT4 or INTTXD1	0: INT4 1: INTTXD1	
INT5ER	Alternative of INTTC0 or $\overline{\text{INT5}}$	0: INTTC0 1: $\overline{\text{INT5}}$	

*Note: * : Don't care*

Figure 1-22 (2). Interrupt Selecting Register (INTSEL)

1.6.1 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 at 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 1-23 shows the timing chart of interrupt acceptance processing.

(1) Interrupt acceptance processing is packaged as follows.

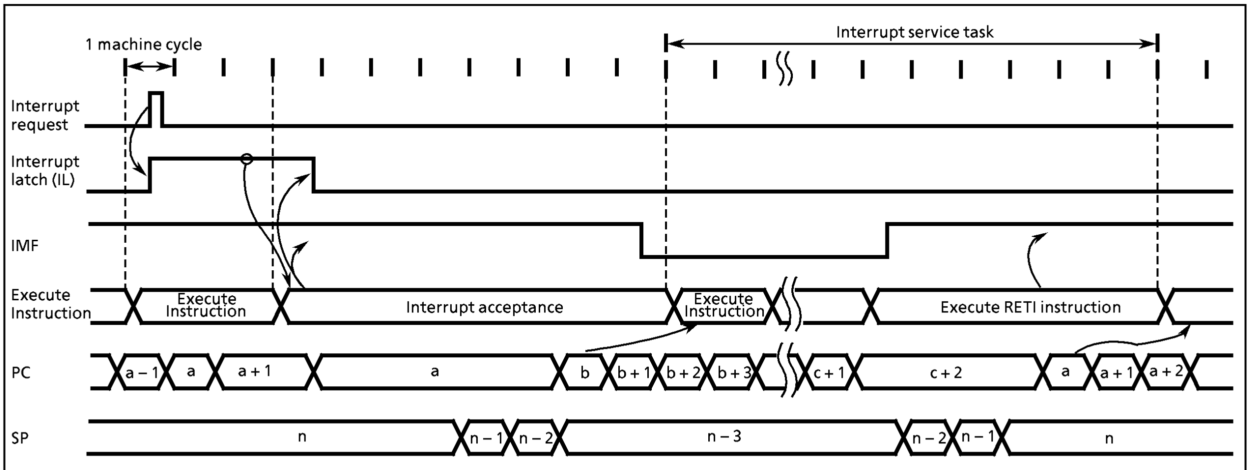
a) The interrupt latch (IL) for the interrupt source accepted is cleared to "0".

b) 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.

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

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.

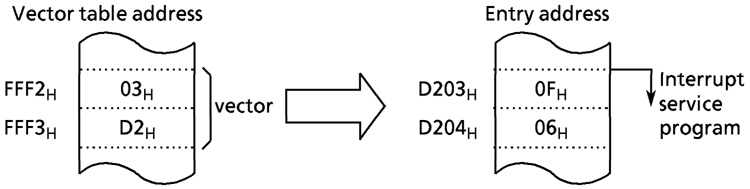


Note 1: a : return address entry address b : entry address c : address which RETI instruction is stored

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 1-23. 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



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.

(2) Saving/Restoring general -purpose registers

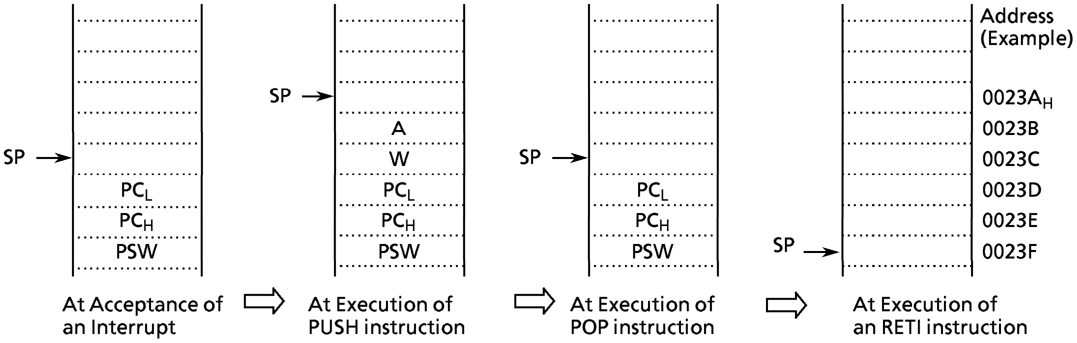
During interrupt acceptance processing, the program counter (PC) and the program status word (PSW, includes IMF) are automatically saved on the stack, but the accumulator and others are not. These registers are saved by software if necessary. When multiple interrupt services are nested, it is also necessary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

a) Using PUSH and POP instructions

To save only a specific register, PUSH and POP instructions are available.

Example: save/store register using PUSH and POP instructions

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

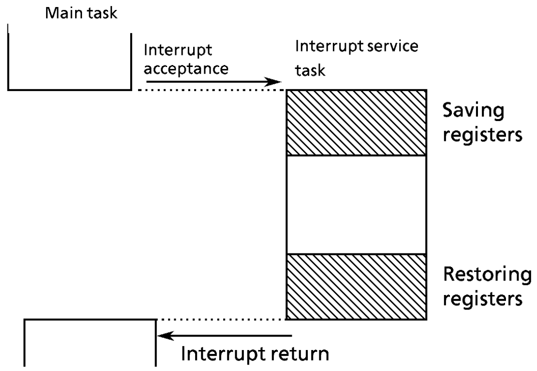

SP
PCL
PCH
PSW
At Acceptance of an Interrupt
SP
A
W
PCL
PCH
PSW
At Execution of PUSH instruction
SP
PCL
PCH
PSW
At Execution of POP instruction
Address (Example)
0023AH
0023B
0023C
0023D
0023E
0023F
SP
At Execution of an RETI instruction

b) 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 instruction

Figure 1-24. Saving/Restoring general-purpose registers under interrupt processing

(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. Otherwise returning interrupt causes INTATRAP again. When interrupt acceptance processing has completed, stacked data for PC_L and PC_H 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
```

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).

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

1.6.2 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.

① Address error detection

FF_H 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 FF_H 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 FF_H to unused areas of the program memory. Address-trap reset is generated in case that an instruction is fetched from RAM or SFR areas.

② Debugging

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

1.6.3 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.

1.6.4 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.
The address trapped area is alternative: SFR and RAM, or SFR only.

Note 1: The operating mode under address trapped, whether to be reset-output or interrupt processing, is selected on watchdog timer control register (WDTCR).
Note 2: When the SWI instruction or undefined instruction located in address immediately preceding the address trap area is executed, address trap interrupt accept processing is executed immediately after the SWI/undefined instruction interrupt is accepted.

1.6.5 External Interrupts

The TMP86CH06 has six 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 $\overline{\text{INT0}}$/P10 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 INT0/P10 pin function selection are performed by the external interrupt control register (EINTCR).

Table 1-7. External Interrupts

Source	Pin	Secondary function pin	Enable conditions	Edge	Digital noise reject
INT0	$\overline{\text{INT0}}$	P10	IMF = 1, EF$_4$ = 1, 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	NT1	P11	IMF · EF$_5$ = 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	P37/OC0	IMF · EF$_{12}$ = 1 INTOC0ER = 1		
INT3	INT3	P46/SI1/RXD1	IMF · EF$_{13}$ = 1 INTRXDER = 0		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	P47/SO1/TXD1	IMF · EF$_{14}$ = 1 INTTXDER = 0		
INT5	$\overline{\text{INT5}}$	P20/$\overline{\text{STOP}}$	IMF · EF$_{15}$ = 1 INT5ER = 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: If a noiseless signal is input to the external interrupt pin in the NORMAL 1/2 or IDLE 1/2 mode, the maximum time from the edge of input signal until the IL is set is as follows:

① INT1 pin 55/fc [s] (INT1NC = 1), 199/fc [s] (INT1NC = 0)

② INT2 to 4 pin 31/fc [s]

Note 2: Even if the falling edge of $\overline{\text{INT0}}$ pin input is detected at INT0EN = 0, the interrupt latch IL_4 is not set.

Note 3: When data changed and did a change of I/O when used external interrupt ports as a normal ports, interrupt request signal occurs incorrectly. Handling of prohibition of interrupt enable register (EIR) is necessary.

Note 4: The maximum time from modifying INT1NC until a noise reject time is changed is 2^6/fc.

	7	6	5	4	3	2	1	0	
EINTCR (0037$_H$)	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	P10/$\overline{\text{INT0}}$ pin configuration	0: P10 input/output port 1: $\overline{\text{INT0}}$ pin (Port P10 should be set to an input mode)	
INT4 ES	INT4 edge select	00: Rising edge 01: Falling edge 10: Both edges 11: High level	
INT3 ES INT2 ES INT1 ES	INT3 to INT1 edge select	0: Rising edge 1: Falling edge	

*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).

Figure 1-25. External Interrupt Control Register

1.7 Reset Circuit

The TMP86CH06 has four types of reset generation procedures: an external reset input, an address trap reset output, a watchdog timer reset output and a system clock reset output. Table 1-8 shows on-chip hardware initialization by reset action.

The malfunction reset output circuit such as watchdog timer reset, address trap reset and system clock reset is not initialized when power is turned on. The $\overline{\text{RESET}}$ pin can output level "L" at the maximum 24/fcgck[s] (1.5 μs at 16.0 MHz) when power is turned on.

Table 1-8. Initializing Internal Status by Reset Action

On-chip hardware	Initial value	On-chip hardware	Initial value
Program counter (PC)	(FFFE$_H$)	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

1.7.1 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/fcgck [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 FFFE to FFFF$_H$.

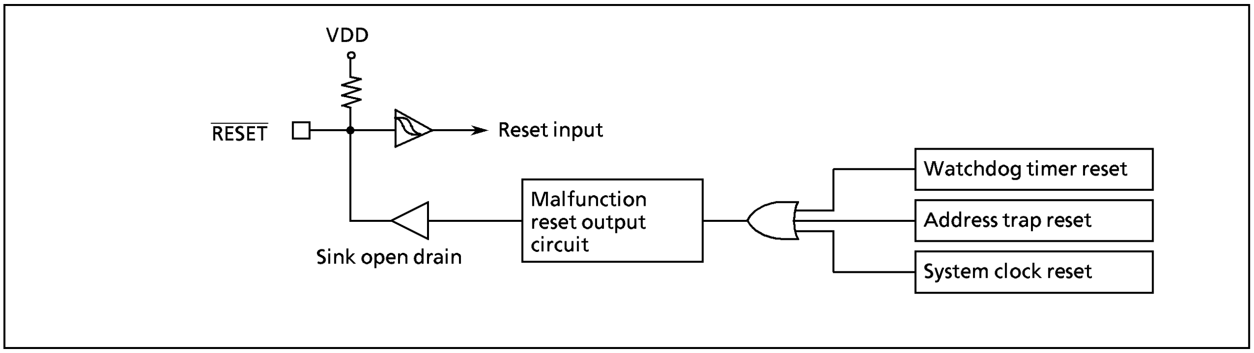
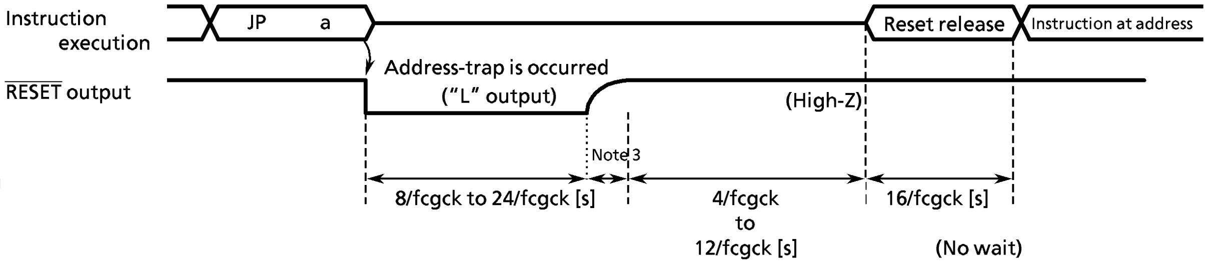


Figure 1-26. Reset Circuit

1.7.2 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 or the SFR area, address-trap-reset will be generated. Then, the $\overline{\text{RESET}}$ pin output will go low. The reset time is about 8/fcgck to 24/fcgck [s] (0.5 to 1.5 μs at 16.0 MHz).



Note 1: Address "a" is in the SFR or on-chip RAM space.
Note 2: During reset release, reset vector "r" is read out, and an instruction at address "r" is fetched and decoded.
Note 3: Varies on account of external condition: voltage or Capacitance

Figure 1-27. Address-Trap-Reset

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

1.7.3 Watchdog Timer Reset

Refer to Section "2.4 Watchdog Timer".

1.7.4 System-Clock-Reset

Clearing both XEN and XTEN (bits 7 and 6 in SYSCR2) to "0", clearing XEN to "0" when SYSCK = 0, or clearing XTEN to "0" when SYSCK = 1 stops system clock, and causes the microcomputer to deadlock. This can be prevented by automatically generating a reset signal whenever XEN = XTEN = 0, XEN = SYSCK = 0, or XTEN = 0/SYSCK = 1 is detected to continue the oscillation. The, the $\overline{\text{RESET}}$ pin output goes low from high-impedance. The reset time is about 8/fcgck to 24/fcgck [s] (0.5 to 1.5 μs at 16.0 MHz).

2.1 Special Function Register (SFR)

The TMP86CH06 adopts the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function register (SFR). A series of SFR is mapped on address 0000_H to $003F_H$.
Figure 2-1 (a) to 2-1 (b) indicate the special function register (SFR) for TMP86CH06.

Address	Read	Write
0000_H	P0DR (P0 Port output latch)	
0001_H	P1DR (P1 Port output latch)	
0002_H	P2DR (P2 Port output latch)	
0003_H	P3DR (P3 Port output latch)	
0004_H	P4DR (P4 Port output latch)	
0005_H	Reserved	
0006_H	Reserved	
0007_H	Reserved	
0008_H	P0CR (P0 Port input/output Control Register)	
0009_H	P1CR (P1 Port input/output Control Register)	
$000A_H$	P3CR (P3 Port input/output Control Register)	
$000B_H$	P4CR (P4 Port input/output Control Register)	
$000C_H$	P4ODE (P4 Port Open-Drain control Register)	
$000D_H$	P2R (P2 Port terminal input)	——
$000E_H$	P4R (P4 Port terminal input)	——
$000F_H$	Reserved	
0010_H	ET0CR (ETC0 Control Register)	
0011_H	ET0MIO (ETC0 Capture/Compare Mode Register)	
0012_H	ET0RL (ETC0 Register L)	
0013_H	ET0RH (ETC0 Register H)	
0014_H	ET0ICAL (ETC0 Capture Register AL)	——
0015_H	ET0ICAH (ETC0 Capture Register AH)	——
0016_H	ET0ICBL (ETC0 Capture Register BL)	PME (Pulse Measure Enable)
0017_H	ET0ICBH (ETC0 Capture Register BH)	——
0018_H	ET0OCRL (ETC0 Compare Register L)	
0019_H	ET0OCRH (ETC0 Compare Register H)	
$001A_H$	UART0SR (UART0 status register)	UART0CRA (UART0 control register A)
$001B_H$	——	UART0CRB (UART0 control register B)
$001C_H$	RD0BUF	TD0BUF
$001D_H$	RD1BUF	TD1BUF
$001E_H$	UART1SR (UART1 status register)	UART1CRA (UART1 control register A)
$001F_H$	——	UART1CRB (UART1 control register B)

Figure 2-1 (a). the special function register (SFR) for TMP86CH06

Address	Read	Write
0020$_H$	TC0CR (Timer0 Control Register)	
0021$_H$	TC1CR (Timer1 Control Register)	
0022$_H$	TTREG0 (Timer Term register0)	
0023$_H$	TTREG1 (Timer Term register1)	
0024$_H$	PWREG0 (Pulse Width register0)	
0025$_H$	PWREG1 (Pulse Width register1)	
0026$_H$	——	SIOCR1 (SIO Control Register 1)
0027$_H$	SIOSR (SIO Status Register)	SIOCR2 (SIO Control Register 2)
0028$_H$	SIODBR (SIO Receive/Transmit Data Buffer (8 byte)	
0029$_H$		
002A$_H$		
002B$_H$		
002C$_H$		
002D$_H$		
002E$_H$		
002F$_H$		
0030$_H$	CGCR (Clock Gear Control Register)	
0031$_H$	——	EXPCR (External Access)
0032$_H$	WAITCR (Wait Control register)	
0033$_H$	Reserved	
0034$_H$	——	WDTCR1 (WDT Control 1)
0035$_H$	——	WDTCR2 (WDT Control 2)
0036$_H$	TBTCR (TBT/TG/DVO control register)	
0037$_H$	EINTCR (External Interrupt Control Register)	
0038$_H$	SYSCR1 (System Control 1)	
0039$_H$	SYSCR2 (System Control 2)	
003A$_H$	EIRL (The lower of interrupt enable register)	
003B$_H$	EIRH (The upper of interrupt enable register)	
003C$_H$	ILL (The lower of interrupt latch)	
003D$_H$	ILH (The upper of interrupt latch)	
003E$_H$	INTSEL (Interrupt Factor Selector)	
003F$_H$	PSW (Program Status Word)	

Figure 2-1 (b). the special function register (SFR) for TMP86CH06

Note 1: Do not access reserved areas by software.
Note 2: – : Cannot be accessed.
Note 3: Write only register and interrupt latches cannot use the read-modify-write instructions (bit manipulation instructions such as SET, CLR, etc. and logical operations such as AND, OR, etc.).
Note 4: Address 0032$_H$: Only bit 2 and 3 on WAITCR can be read.

2.2 I/O Ports

The TMP86CH06 has 5 parallel input/output ports (35 pins) as follows.

① Port P0 ; 8-bit I/O port (utilized also for Address/Data bus)

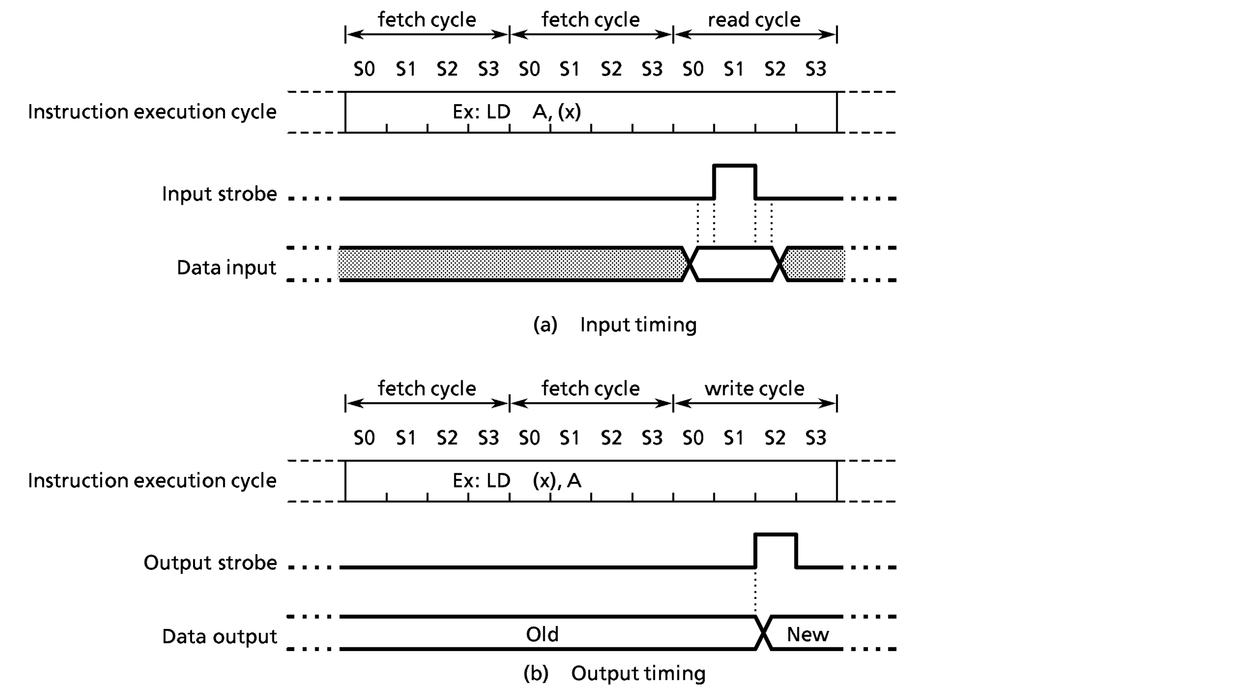
② Port P1 ; 8-bit I/O port (utilized also for External interrupt input, Timer input/output and External memory management output)

③ Port P2 ; 3-bit I/O port (utilized also for Low frequency resonator connections, External interrupt input and STOP mode releasing signal input)

④ Port P3 ; 8-bit I/O port (utilized also for Address bus output, Timer input/output and External interrupt input)

⑤ Port P4 ; 8-bit I/O port (utilized also for Timer input and Serial interface input/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 2-2 shows input/output timing examples.
External data is read from an I/O port in the S1 state of the read cycle during execution of the read instruction. This timing cannot be recognized from outside, so that transient input such as chattering must be processed by the program.
Output data changes in the S2 state of the write cycle during execution of the instruction which writes to an I/O port.



(a) Input timing

(b) Output timing

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

Figure 2-2. Input/Output Timing (Example)

2.2.1 Port P0

Port P0 is the 8-bit I/O port that allows selection of input/output on bit basis. Input/output mode is specified on the P0 port input/output control register (P0CR). During reset, all the bits on P0CR are initialized to "0" and P0 becomes input port. Reset operation also initializes all the bits on P0 port output latch (P0DR) to "0".
Besides input/output port, Port P0 functions as multiplexed Address/Data bus (AD7 to 0). Port P0 becomes 8-bit bidirectional Address/Data bus (AD7 to 0) when the CPU accesses to the external memory. When it functions as data bus, the judgment on its input rate is based on TTL level.

Note: Input status is read while the port is input mode. Therefore the contents of output latch, that belongs to the terminal for input, may alter if both input and output are mixed in P0 port.

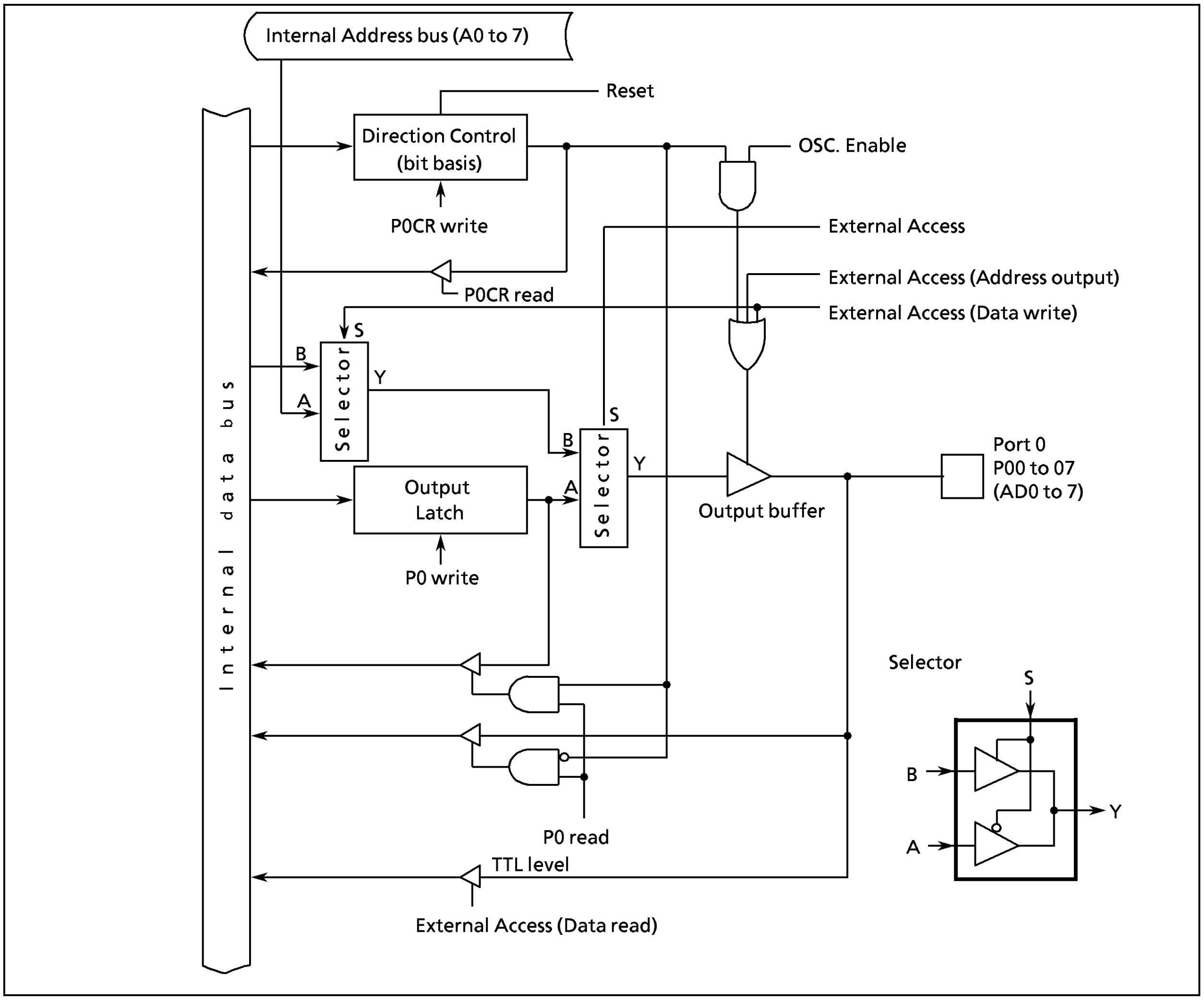


Figure 2-3. Port P0 (1/2)

P0DR (0000_H)	7	6	5	4	3	2	1	0	
	P07	P06	P05	P04	P03	P02	P01	P00	(Initial Value: 0000 0000)

P0CR (0008_H)	7	6	5	4	3	2	1	0	
	P0CR7	P0CR6	P0CR5	P0CR4	P0CR3	P0CR2	P0CR1	P0CR0	(Initial Value: 0000 0000)

P0CR	I/O control for port P0 (Set for each bit individually)	0: Input mode 1: Output mode	R/W

Figure 2-3. Port P0 (2/2)

2.2.2 Port P1

Port P1 is the 8-bit I/O port that allows selection of input/output on bit basis. Input/output mode is specified on the P1 port input/output control register (P1CR). During reset, all the bits on P1CR are initialized to "0" and P1 becomes input port. Reset operation also initializes all the bits on P1 port output latch (P1DR) to "0".

Besides input/output port, the terminals in Port P1 functions as follows.

P15, P16 and P17 have $\overline{\text{WR}}$, $\overline{\text{RD}}$ and ALE function respectively. These terminals output signals for $\overline{\text{WR}}$, $\overline{\text{RD}}$ and ALE when the CPU accesses to the external memory. In order to utilize $\overline{\text{WR}}$ and $\overline{\text{RD}}$ functions, set WROE and RDOE, located on the EXPCR, respectively. If WROE, RDOE or both are enabled, P17 functions as ALE. If the device is released from reset while $\overline{\text{EA}}$ terminal is low, $\overline{\text{CLK}}$ signal is uttered every machine cycle.

P10, P11 and P12 have $\overline{\text{INT0}}$, INT1, and ETC0 function respectively. In order to utilize these functions, the terminal should be set for input.

Furthermore, pll has the function of $\overline{\text{WAIT}}$ while the external memory is begin connected. If the $\overline{\text{WAIT}}$ function is not necessary as external memory is utilized, clear the WAIT bits (bit 7, 6) on WAITCR to "00".

P13 and P14 have $\overline{\text{DVO}}$ and TO1 function respectively. In order to utilize these functions, the output latch belongs to each terminal should be set to "1" before the terminal is set for output.

Note1: Input status is read while the port is input mode. Therefore the contents of output latch, that belongs to the terminal for input, may alter if both input and output are mixed in P1 port.
Note2: When the external memory is used, P10 pin cannot be used as external interrupt pin ($\overline{\text{INT0}}$) or input/output port because $\overline{\text{CLK}}$ is output from P10.

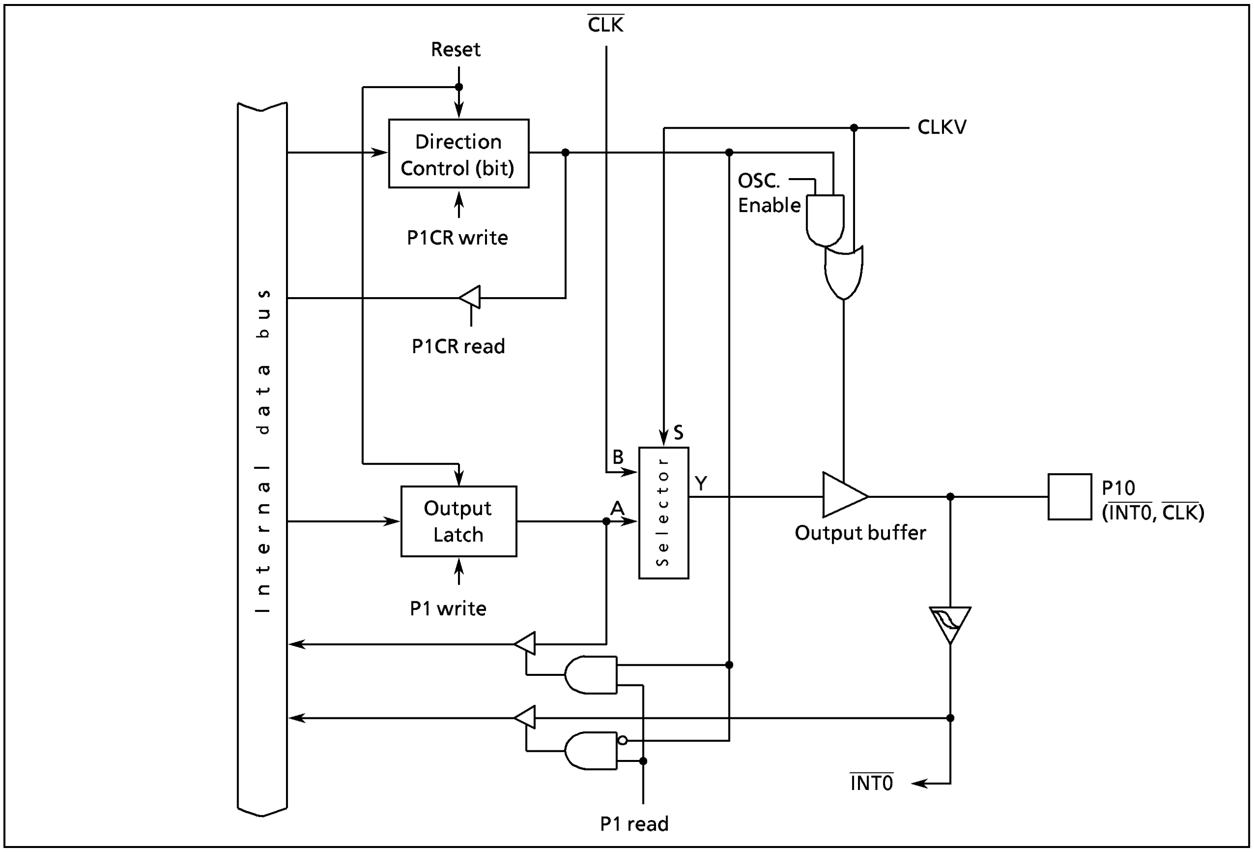


Figure 2-4. P10

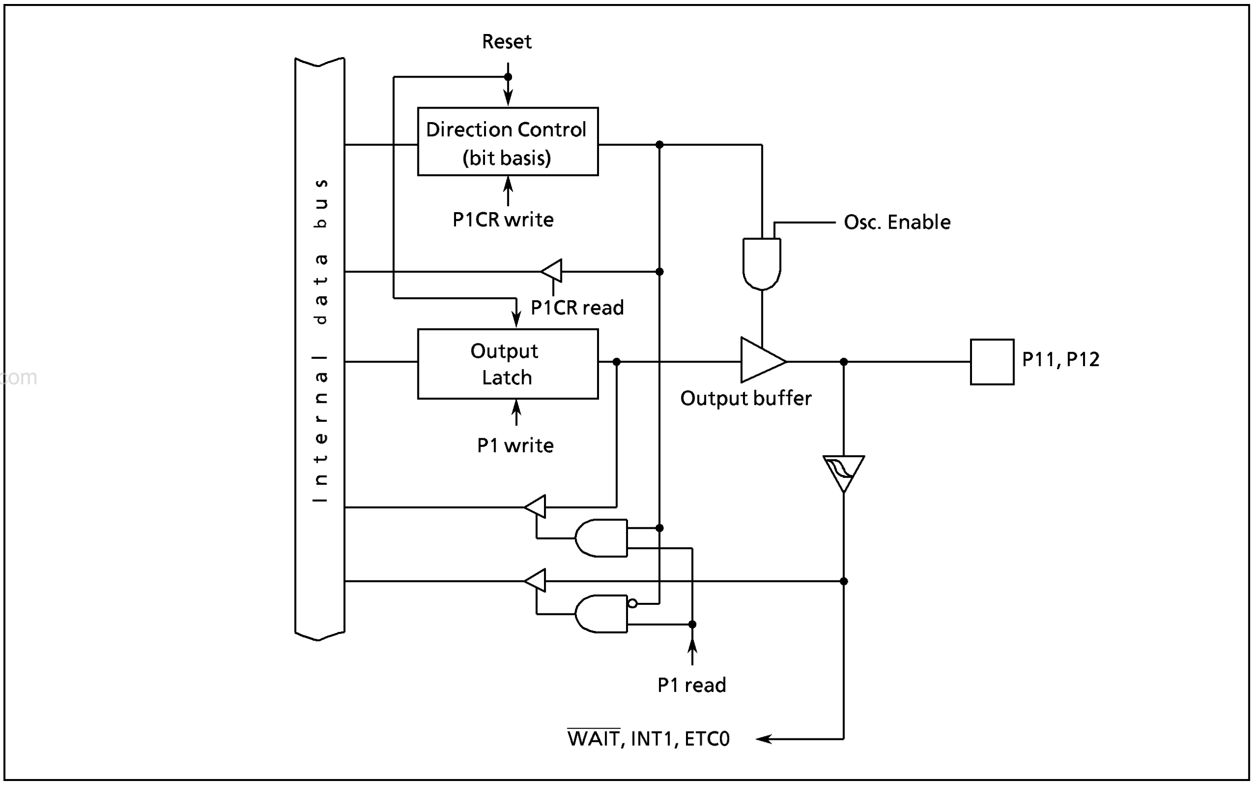


Figure 2-5. P11, P12

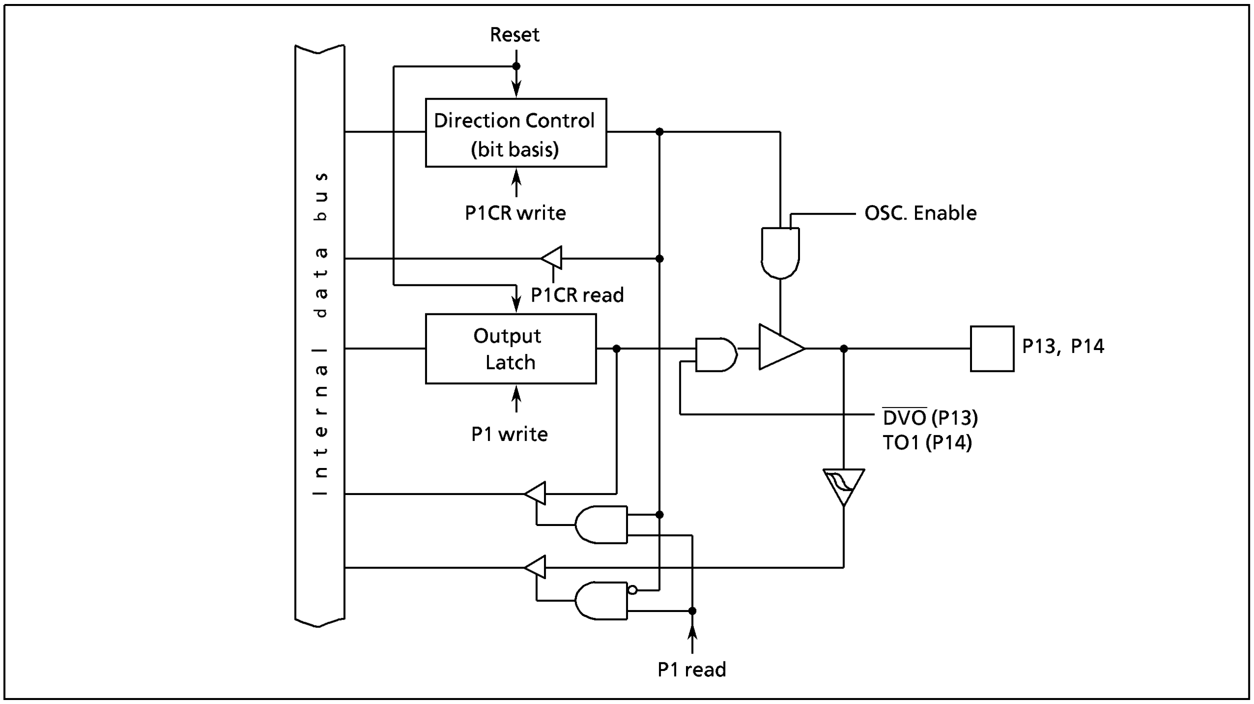


Figure 2-6. P13, P14

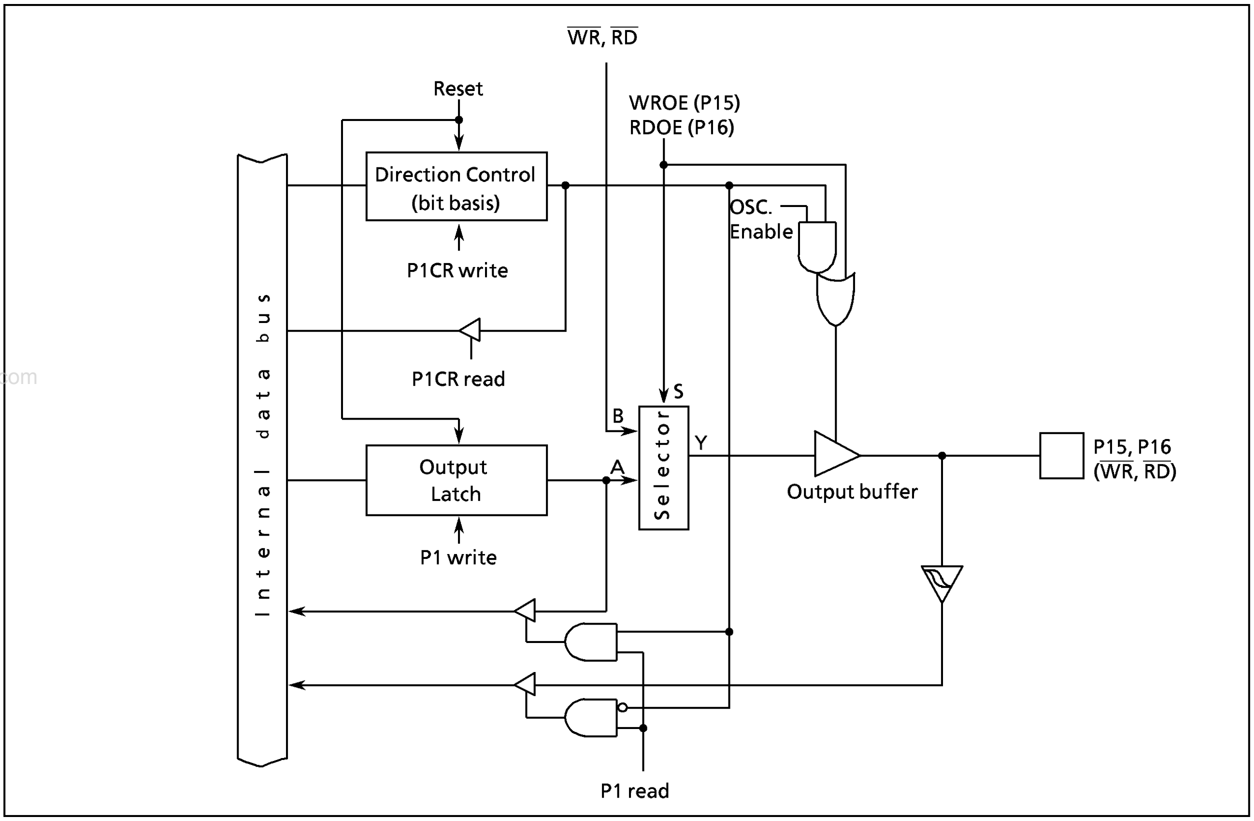


Figure 2-7. P15, P16

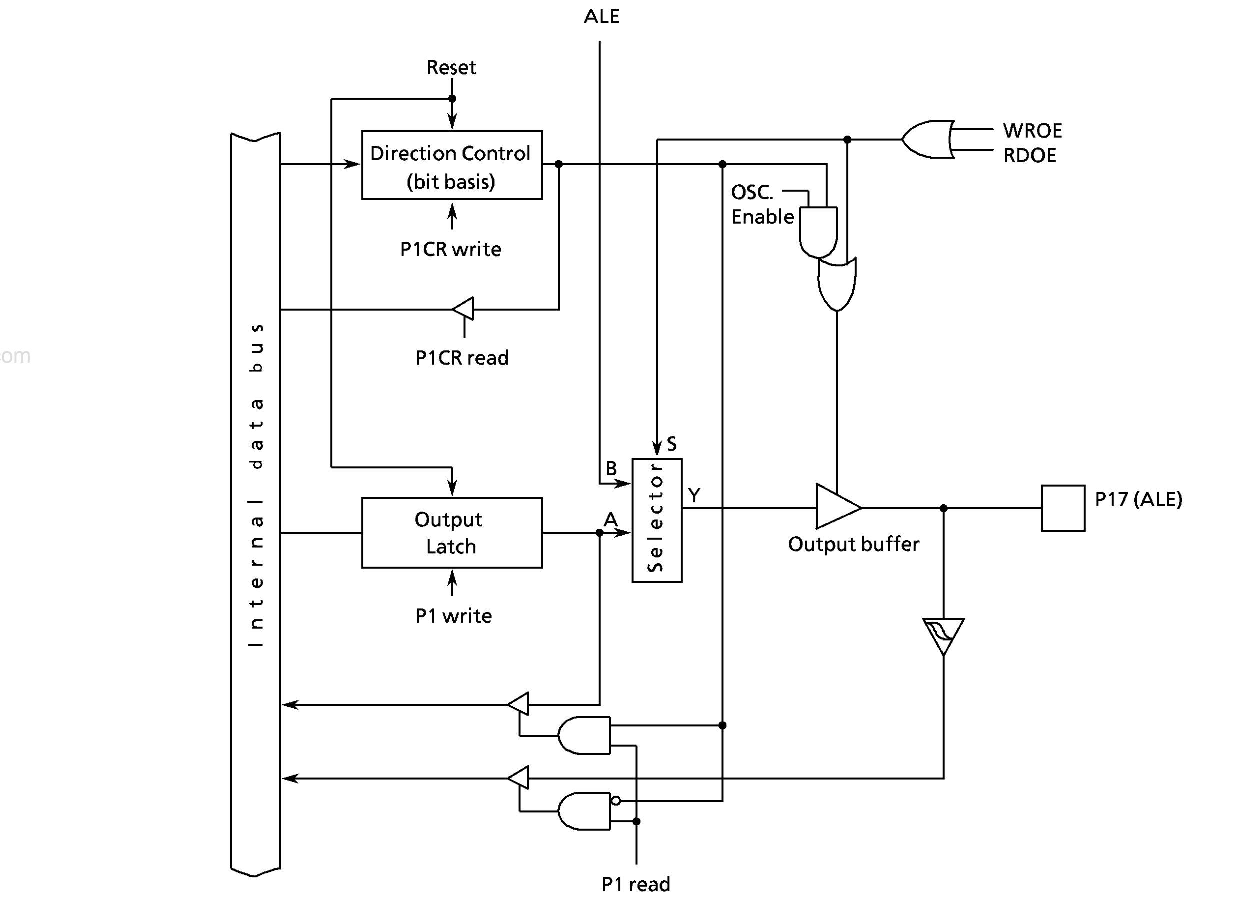


Figure 2-8. P17

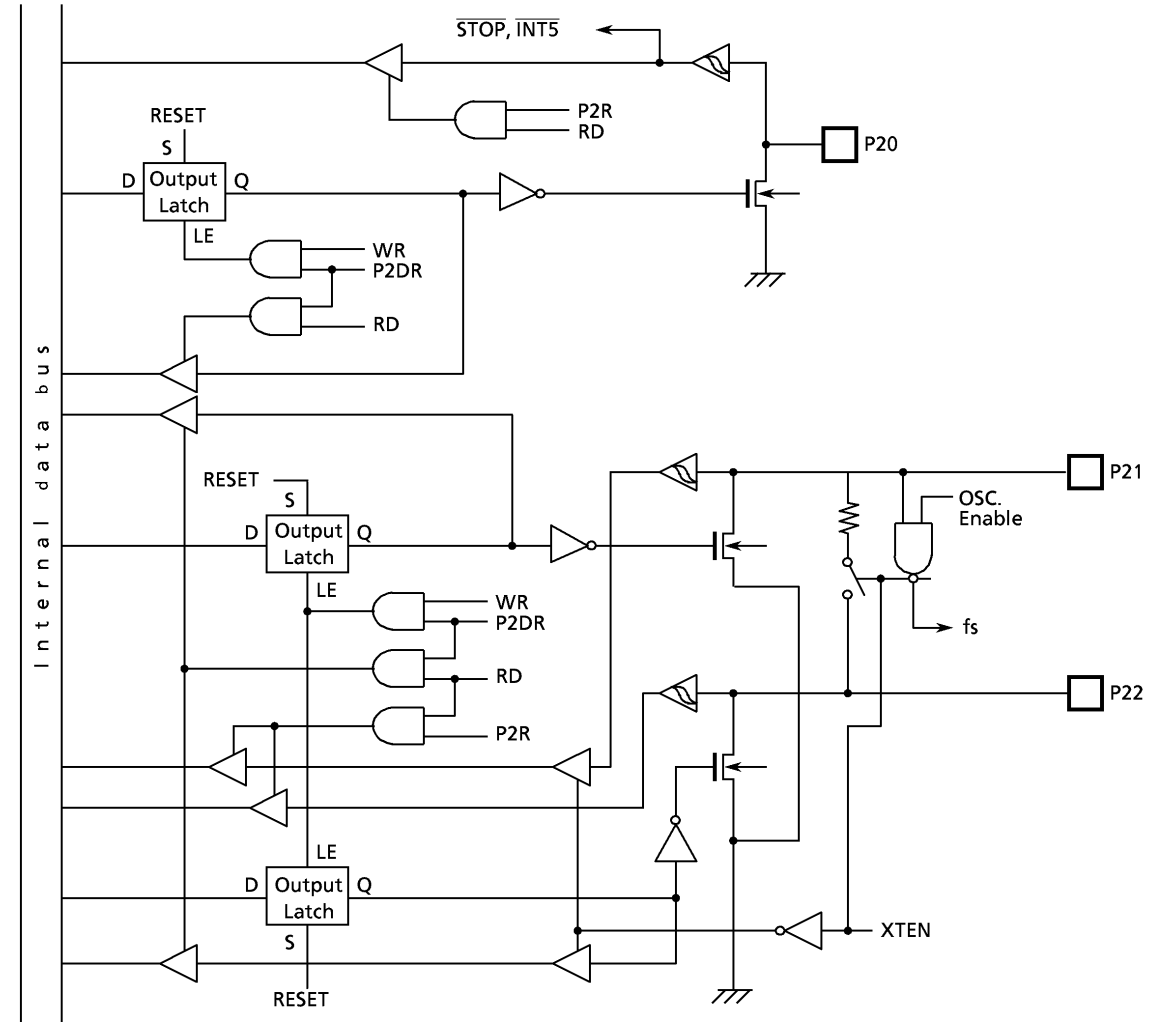
	7	6	5	4	3	2	1	0	
P1DR (0001$_H$)	P17	P16	P15	P14	P13	P12	P11	P10	(Initial Value: 0000 0000)
P1CR (0009$_H$)	P1CR7	P1CR6	P1CR5	P1CR4	P1CR3	P1CR2	P1CR1	P1CR0	(Initial Value: 0000 0000)

P1CR	I/O control for port P1 (Set for each bit individually)	0: Input mode 1: Output mode	R/W

Figure 2-9. Port P1DR and P1CR register

2.2.3 Port P2

Port P2 is a 3-bit input/output port. It is also used as an external interrupt, a STOP mode release signal input, and low-frequency Xtal connection pins. When they are used as an input port or a secondary function pins, the respective output latch should be set to "1".
A low-frequency Xtal (32.768 kHz) is connected to pins P21 (XTIN) and P22 (XTOUT) in the dual-clock mode. In the single-clock mode, pins P21 and P22 can be used as normal input/output ports.
It is recommended that pin P20 should be used as an external interrupt input, a STOP mode release signal input, or an input port. If it is used as an output port, the interrupt latch is set on the falling edge of the output pulse.
When a read instruction is executed for port P2, bits 7 to 3 are read as undefined values.
P2 port output latch (P2DR) and P2 port terminal input (P2R) are located on their respective addresses. Therefore, if input and output pins are mixed in P2 port, Read Write Modify instructions do not affect the output latch belongs to the terminal for input.



P2DR (0002$_H$)	7	6	5	4	3	2	1	0	
	*	*	*	*	*	P22	P21	P20	(Initial Value: 1111 1111)

P2R (000D$_H$)	7	6	5	4	3	2	1	0	
	*	*	*	*	*	P22IN	P21IN	P20IN	(Initial Value: 1111 1- - -) Read-only

Figure 2-10. Port P2

Note1: 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.
Note2: Bit 7 through bit 3 in P2R contain unstable values.
*Note3: *: Don't care.*

2.2.4 Port P3

Port P3 is the 8-bit I/O port that allows selection of input/output on bit basis. Input/output mode is specified on the P3 port input/output control register (P3CR). During reset, all the bits on P3CR are initialized to "0" and P3 becomes input port. Reset operation also initializes all the bits on P3 port output latch (P3DR) to "0".

Besides input/output port, the terminals in Port P3 functions as follows.

P36 and P37 have IC0 and INT2 function respectively. In order to utilize these functions, the terminal should be set for input.

P35 and P37 have TO0 and OC0 function respectively. In order to utilize these functions, the output latch belongs to each terminal should be set to "1" before the terminal is set for output.

Port P3 also functions as address bus (A15 to 8). Port P3 becomes 8-bit bidirectional address bus (A15 to 8) when the CPU accesses to the external memory. In order to utilize address bus (A15 to 8) function, set ABUSEN, located on the EXPCR.

Note: Input status is read while the port is input mode. Therefore the contents of output latch, that belongs to the terminal for input, may alter if both input and output are mixed in P3 port.

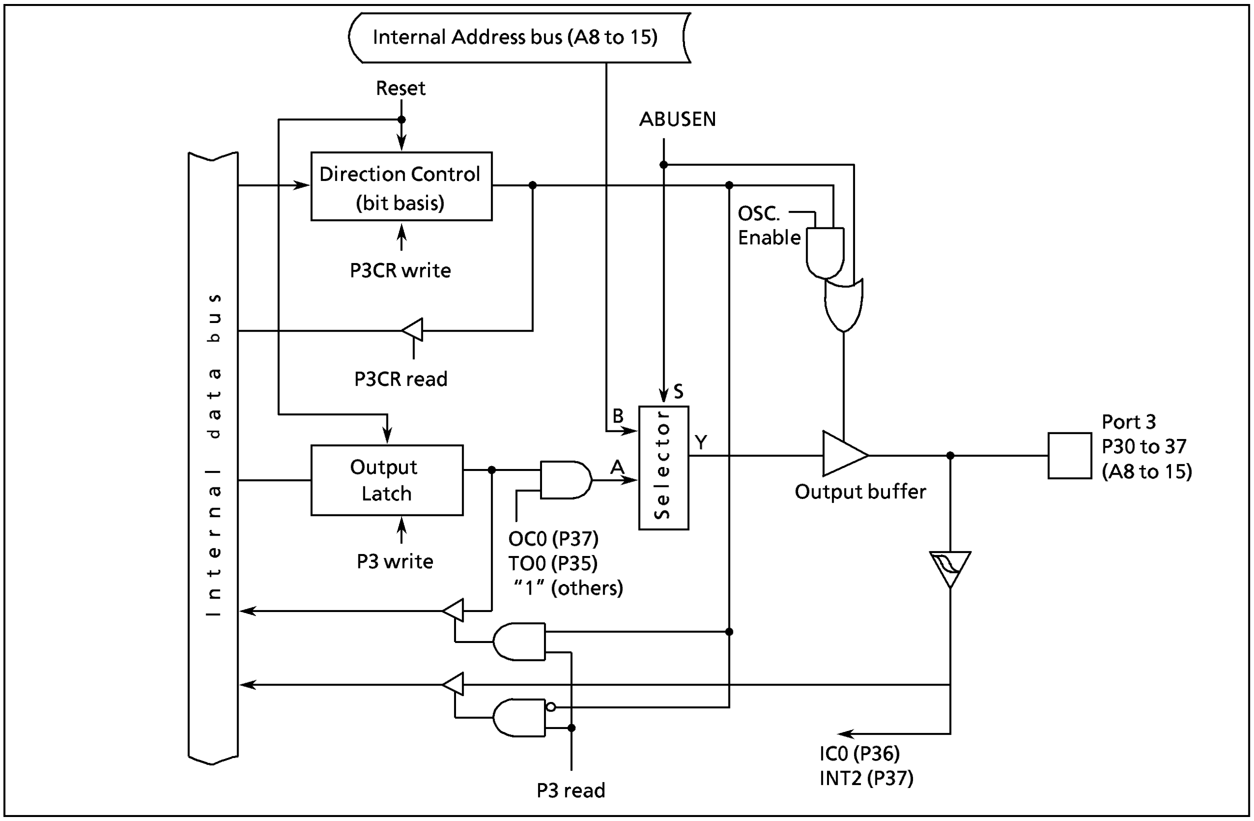


Figure 2-11. Port P3 (1/2)

P3DR
(0003_H)

7	6	5	4	3	2	1	0
P37	P36	P35	P34	P33	P32	P31	P30

(Initial Value: 0000 0000)

P3CR
($000A_H$)

7	6	5	4	3	2	1	0
P3CR7	P3CR6	P3CR5	P3CR4	P3CR3	P3CR2	P3CR1	P3CR0

(Initial Value: 0000 0000)

P3CR	I/O control for port P3 (Set for each bit individually)	0: Input mode 1: Output mode	R/W

Figure 2-11. Port P3 (2/2)

2.2.5 Port P4

Port P4 is the 8-bit I/O port that allows selection of input/output on bit basis. Input/output mode is specified on the P4 port input/output control register (P4CR). During reset, all the bits on P4CR are initialized to “0” and P4 becomes input port. Reset operation also initializes all the bits on P4 port output latch (P4DR) to “1”.

Port P4 has programmable open-drain output function. The data on P4 port open-drain control register (P4ODE) determines whether open-drain output is enabled or disabled on bit basis. During reset, all the bits on P4ODE are initialized to “0”, and under the circumstances P4 becomes CMOS output port if P4CR is set to “1”.

Besides input/output port, the terminals in Port P4 functions as follows.

P40 and P41 have TI0 and TI1 function respectively: both TI0 and TI1 are for 8-bit timers. In order to utilize these functions, the terminal should be set for input.

P46 and P47 have INT3 and INT4 function respectively: both INT3 and INT4 are for interrupts. In order to utilize these functions, the terminal should be set for input.

The bundles, one consists of P42, P43 and P44 and the other consists of P46 and P47, have serial interface function respectively.

P4 output latch (P4DR) and P4 port terminal input (P4R) are located on their respective addresses. Therefore, if input and output pins are mixed in P4 port, Read Write Modify instructions do not affect the output latch belongs to the terminal for input.

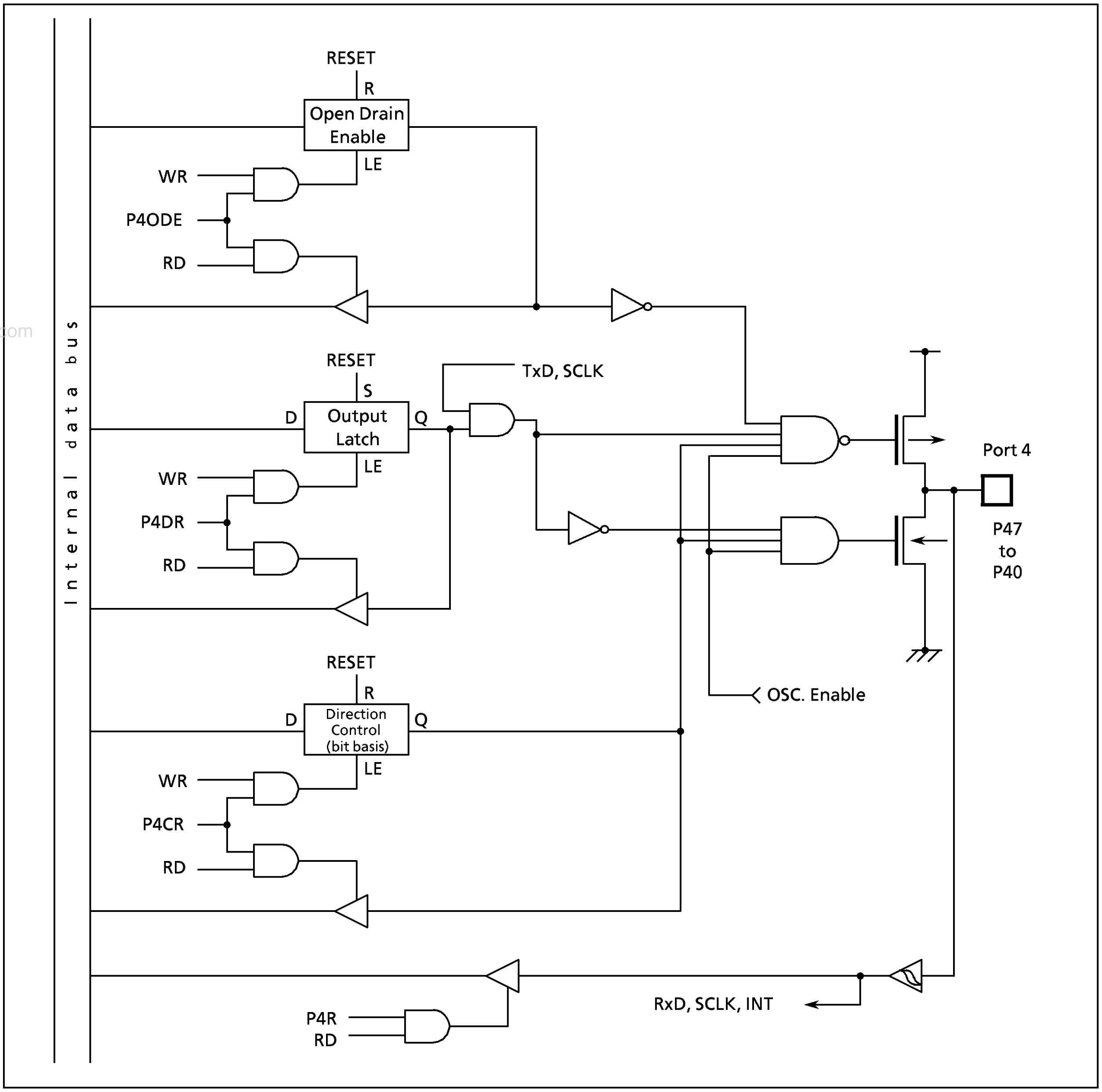


Figure 2-12. Port P4 (1/2)

P4DR (0004_H)	7	6	5	4	3	2	1	0	
	P47	P46	P45	P44	P43	P42	P41	P40	(Initial Value: 1111 1111)

P4R ($000E_H$)	7	6	5	4	3	2	1	0	
	P47IN	P46IN	P45IN	P44IN	P43IN	P42IN	P41IN	P40IN	(Initial Value: - - - - - - - -) Read only

P4CR ($000B_H$)	7	6	5	4	3	2	1	0	
	P4CR7	P4CR6	P4CR5	P4CR4	P4CR3	P4CR2	P4CR1	P4CR0	(Initial Value: 0000 0000)

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

P4ODE ($000C_H$)	7	6	5	4	3	2	1	0	
	P4ODE7	P4ODE6	P4ODE5	P4ODE4	P4ODE3	P4ODE2	P4ODE1	P4ODE0	(Initial Value: 0000 0000)

P4ODE	I/O control for port P4 (Set for each bit individually)	0: 3-state output mode 1: Nch O.D. output mode	R/W

Figure 2-12. Port P4 (2/2)

2.3 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).

An INTTBT is generated on the first falling edge of source clock (the divider output of the timing generator) after the 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 2-9.(b)).

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

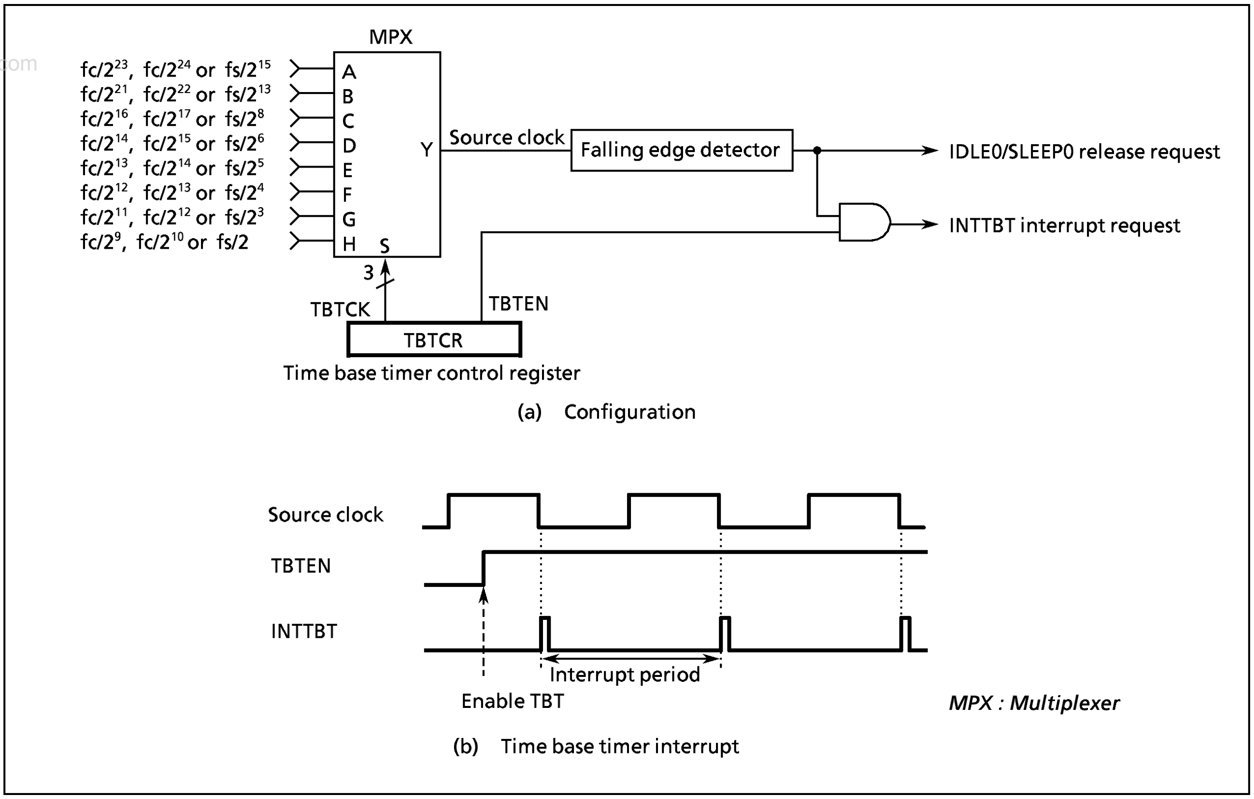

MPX
fc/2^23, fc/2^24 or fs/2^15
fc/2^21, fc/2^22 or fs/2^13
fc/2^16, fc/2^17 or fs/2^8
fc/2^14, fc/2^15 or fs/2^6
fc/2^13, fc/2^14 or fs/2^5
fc/2^12, fc/2^13 or fs/2^4
fc/2^11, fc/2^12 or fs/2^3
fc/2^9, fc/2^10 or fs/2
A
B
C
D
E
F
G
H
S
Y
3
Source clock
Falling edge detector
IDLE0/SLEEP0 release request
INTTBT interrupt request
TBTCK
TBTEN
TBTCR
Time base timer control register
(a) Configuration
Source clock
TBTEN
INTTBT
Interrupt period
Enable TBT
MPX : Multiplexer
(b) Time base timer interrupt

Figure 2-13. Time Base Timer

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

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

	7	6	5	4	3	2	1	0	
TBTCR (00036$_H$)	(DVOEN)	(DVOCK)		(DV7CK)	TBTEN	TBTCK			(Initial value: 0000 0000)

TBTEN	Time base timer enable/disable	0: Disable 1: Enable							R/W
TBTCK	Time base timer interrupt frequency select [Hz]		NORMAL1/2, IDLE1/2 mode					SLOW, SLEEP mode	
			DV7CK = 0		DV7CK = 1				
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1			
		000	$fc/2^{23}$	$fc/2^{24}$	$fs/2^{15}$	$fs/2^{15}$	DV21	$fs/2^{15}$	
		001	$fc/2^{21}$	$fc/2^{22}$	$fs/2^{13}$	$fs/2^{13}$	DV19	$fs/2^{13}$	
		010	$fc/2^{16}$	$fc/2^{17}$	$fs/2^{8}$	$fs/2^{8}$	DV14	–	
		011	$fc/2^{14}$	$fc/2^{15}$	$fs/2^{6}$	$fs/2^{6}$	DV12	–	
		100	$fc/2^{13}$	$fc/2^{14}$	$fs/2^{5}$	$fs/2^{5}$	DV11	–	
		101	$fc/2^{12}$	$fc/2^{13}$	$fs/2^{4}$	$fs/2^{4}$	DV10	–	
		110	$fc/2^{11}$	$fc/2^{12}$	$fs/2^{3}$	$fs/2^{3}$	DV9	–	
		111	$fc/2^{9}$	$fc/2^{10}$	fs/2	fs/2	DV7	–	

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

Figure 2-14. Time Base Timer Control Register

Table 2-1. 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				SLOW, SLEEP mode
	DV7CK = 0		DV7CK = 1		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
000	1.91	0.95	1	1	1
001	7.63	3.81	4	4	4
010	244.14	122.07	128	128	–
011	976.56	488.28	512	512	–
100	1953.13	976.56	1024	1024	–
101	3906.25	1953.13	2048	2048	–
110	7812.5	3906.25	4096	4096	–
111	31250	15625	16384	16384	–

2.4 Watchdog Timer (WDT)

The watchdog timer is a fail-safe system to rapidly detect the CPU malfunctions such as endless looping caused by noise or the like, or deadlock and resume the CPU to the normal state.
The watchdog timer signal for detecting malfunction can be selected either a reset output or a non-maskable interrupt request. However, selection is possible only once after reset. At first the reset output is selected.
When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals.

Note: The functions of the watchdog timer may not be fully realized due to disturbance noise, etc. Therefore, careful consideration is required in the designing stage.

2.4.1 Watchdog Timer Configuration

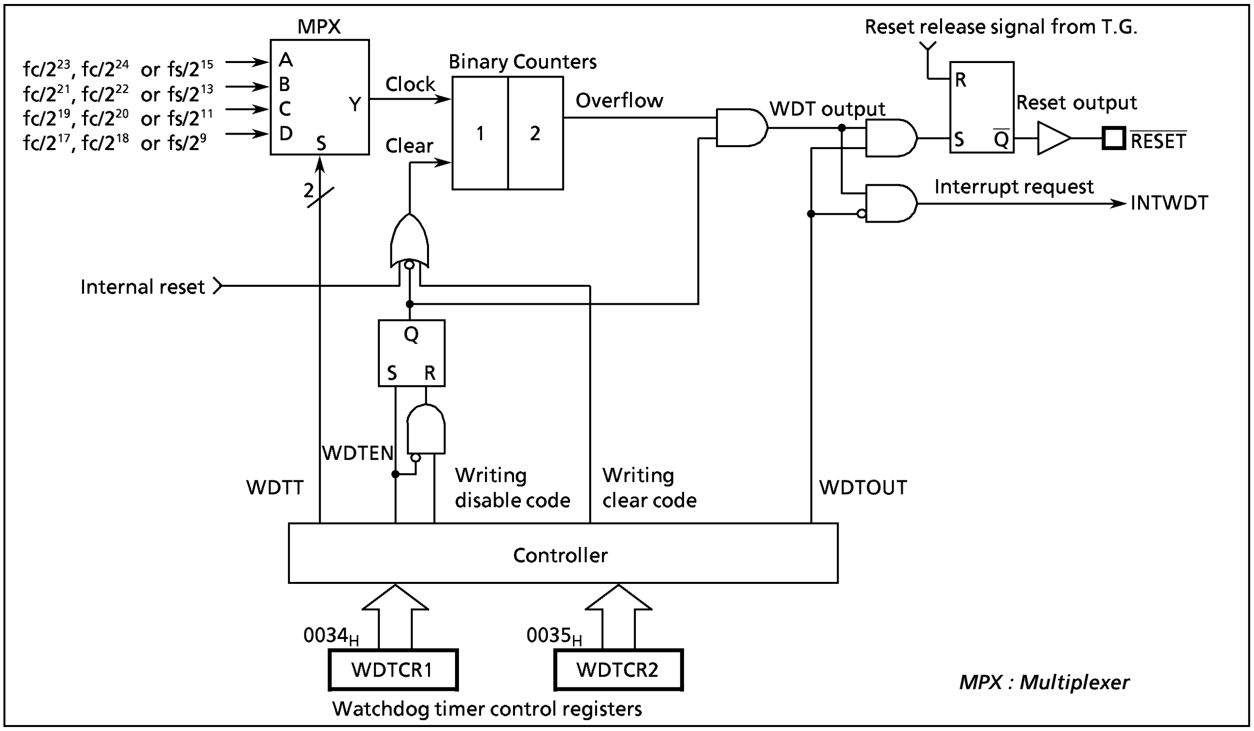


Figure 2-15. Watchdog Timer Configuration

2.4.2 Watchdog Timer Control

Figure 2-16 shows the watchdog timer control registers (WDTCR1, WDTCR2). The watchdog timer is automatically enabled after reset.

(1) Malfunction detection methods using the watchdog timer

The CPU malfunction is detected at follows.

① Setting the detection time, selecting output, and clearing the binary counter.
② Repeatedly clearing the binary counter within the setting detection time

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

If the CPU malfunctions such as endless looping or deadlock occur for any cause, the watchdog timer output will become active at the rising of an overflow from the binary counters unless the binary counters are cleared. At this time, when WDTOUT = 1 a reset is generated, which drivers the $\overline{\text{RESET}}$ pin low to reset the internal hardware and the external circuit. When WDTOUT = 0, a watchdog timer interrupt (INTWDT) is generated.
The watchdog timer temporarily stops counting in STOP mode including warm-up or IDLE mode, and automatically restarts (continues counting) when the STOP/IDLE mode is released.

Example: Sets the watchdog timer detection time to 2^{21}/fc [s] and resets the CPU malfunction.

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

Watchdog Timer Register 1

	7	6	5	4	3	2	1	0	
WDTCR1 (0034_H)			(ATAS)	(ATOUT)	WDT EN	WDTT		WDT OUT	(Initial value: **11 1001)

<table>
<tr><td rowspan="2">WDTEN</td><td rowspan="2">Watchdog timer enable/disable</td><td colspan="6">0: Disable (It is necessary to write the disable code to WDTCR2)
1: Enable</td><td rowspan="10">Write only</td></tr>
<tr></tr>
<tr><td rowspan="7">WDTT</td><td rowspan="7">Watchdog timer detection time [s]</td><td rowspan="3"></td><td colspan="4">NORMAL1/2 mde</td><td rowspan="3">SLOW mode</td></tr>
<tr><td colspan="2">DV7CK = 0</td><td colspan="2">DV7CK = 1</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>00</td><td>2^{25}/fc</td><td>2^{26}/fc</td><td>2^{17}/fs</td><td>2^{17}/fs</td><td>2^{17}/fs</td></tr>
<tr><td>01</td><td>2^{23}/fc</td><td>2^{24}/fc</td><td>2^{15}/fs</td><td>2^{15}/fs</td><td>2^{15}/fs</td></tr>
<tr><td>10</td><td>2^{21}/fc</td><td>2^{22}/fc</td><td>2^{13}/fs</td><td>2^{13}/fs</td><td>2^{13}/fs</td></tr>
<tr><td>11</td><td>2^{19}/fc</td><td>2^{20}/fc</td><td>2^{11}/fs</td><td>2^{11}/fs</td><td>2^{11}/fs</td></tr>
<tr><td>WDTOUT</td><td>Watchdog timer output select</td><td colspan="6">0: Interrupt request
1: Reset output</td></tr>
</table>

Note 1: WDTOUT cannot be set to "1" by program after clearing WDTOUT to "0".
*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.
Note 4: The watchdog timer must be disabled or the counter must be cleared immediately before entering to the STOP mode. When the counter is cleared, the counter must be cleared again immediately after releasing the STOP mode.
Note 5: Just right before disabling the watchdog timer, disable the acceptance of interrupts (DI) and clear the watchdog timer.
If the watchdog timer is disabled under conditions other than the above, the proper operation cannot be guaranteed.

```
Example :  DI                               Disables interrupt acceptance.
           LD     (WDTCR2), 4EH             Clears the watchdog timer.
           LDW    (WDTCR1), 0B101H          Disables the watchdog timer.
           EI                               Enables interrupt acceptance.
```

Watchdog Timer Register 2

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

WDTCR2	Watchdog timer control code write register	$4E_H$: Watchdog timer binary counter clear (clear code) $B1_H$: Watchdog timer disable (disable code) D2: Enable assigning address trap area Others: Invalid	Write only

Note 1: The disable code is invalid unless written when 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: Clears the binary counter does not clear the source clock.
It is recommended that the time to clear is set to 3/4 of the detecting time
Note 5: WDTCR2 is a write-only register and must not be used with any of the read-modify-write instructions.

Figure 2-16. Watchdog Timer Control Registers

(2) Watchdog timer enable

The watchdog timer is enabled by setting WDTEN (bit 3 in WDTCR1) to "1". WDTEN is initialized to "1" during reset, so the watchdog timer operates immediately after reset is released.

(3) Watchdog timer disable

The watchdog timer is disabled by writing the disable code ($B1_H$) to WDTCR2 after clearing WDTEN (bit 3 in WDTCR1) to "0", The watchdog timer is not disabled if this procedure is reversed and the disable code is written to WDTCR2 before WDTEN is cleared to "0". During disabling the watchdog timer, the binary counters are cleared to "0".

Example: Disables watchdog timer

```
        DI                    ;  IMF ← 0
        LD(WDTCR2), 04EH      ;  Clear the binary counter
        LDW (WDTCR1) , 0B101H ;  WDTEN ← 0, WDTCR2 ← Disable code
        EI                    ;  IMF ← 1
```

Table 2-2. 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		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
00	2.097	4.194	4	4	4
01	524.288 m	1.049	1	1	1
10	131.072 m	262.144 m	250 m	250 m	250 m
11	32.768 m	65.536 m	62.5 m	62.5 m	62.5 m

2.4.3 Watchdog Timer Interrupt (INTWDT)

This is a pseudo non-maskable interrupt which can be accepted regardless of the contents of the EIR. If a watchdog timer interrupt or a software interrupt is already accepted, however, the new watchdog timer interrupt waits until the previous interrupt processing is completed (the end of the [RETN] instruction execution).

The stack pointer (SP) should be initialized before using the watchdog timer output as an interrupt source with WDTOUT.

Example: Watchdog timer interrupt setting up

```
        LD   SP, 0023FH             ;  Sets the stack pointer
        LD   (WDTCR1), 00001000B    ;  WDTOUT ← 0
```

2.4.4 Watchdog Timer Reset

If the watchdog timer output becomes active, a reset is generated, which drivers the $\overline{\text{RESET}}$ pin (sink open drain input/output with pull-up) low to reset the internal hardware. The reset output time is about 8/fcgck to 24/fcgck [s] (0.5 to 1.5 μs at fc = 16.0 MHz).

Note: The high-frequency clock oscillator also turns on when a watchdog timer reset is generated in SLOW mode. The reset output time is 8/fcgck to 24/fcgck [s]. Therefore, the reset time may include a certain amount of error if there is any fluctuation of the oscillation frequency at starting the high-frequency clock oscillation. Thus, the reset time must be considered an approximated value.

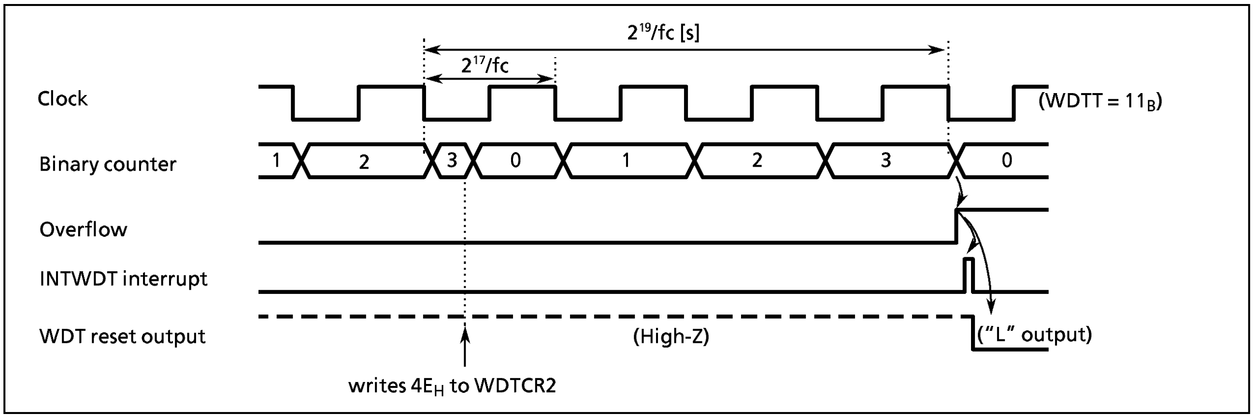


Figure 2-17. Watchdog Timer Interrupt/Reset

2.4.5 Address trap

The Watchdog Timer Control Register 1, 2 shares its addresses with the control registers in case of address trap. These control registers for address trap are shown on Figure 2-18. Whether internal RAM area should be trapped or not is selected on bit ATAS on WDTCR1. The written data becomes valid after the control code D2H is written on WDTCR2. If the instructions are to be placed on internal RAM area, internal RAM area should be excepted from the area to be trapped before such instructions are executed.

The operating mode under address trapped, whether to be reset-output or interrupt processing, is selected on bit ATOUT on WDTCR1.

Example: Setting in order that the CPU normally executes instruction in internal RAM area and the address trap causes interrupt

```
LD      (WDTCR1), 0D200H
```

Watchdog Timer Control Register 1

WDTCR1 (0034$_H$)	7	6	5	4	3	2	1	0	
	–	–	ATAS	ATOUT	(WDT EN)	(WDTT)		(WDT OUT)	(Initial value: **11 1001)

ATAS	Area for address trap	0: SFR area only 1: SFR, RAM and MEM1 area (Valid if control code is written on WDTCR2)	Write only
ATOUT	Operation in case of address trap	0: Interrupt 1: Reset	

Watchdog Timer Control Register 2

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

WDTCR2	Watchdog timer control code and Address trapped area control code	D2: Address trapped area valid to set (ATRAP control code) 4E: Watchdog timer binary counter clear (WDT clear code) B1$_H$: Watchdog timer disable (WDT disable code) Others: Invalid	Write only

Figure 2-18. Watchdog Timer Control Registers

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

Approximately 50% duty pulse can be output using the divider output circuit, which is useful for piezoelectric buzzer drive. Divider output is from pin P13 ($\overline{\text{DVO}}$). The P13 output latch (P1DR bit 3 = 1) should be set to "1" and then the P13 should be configured as an output mode (P1CR bit 3 = 1).

> *Note: The divider output frequency should be selected with divider output disabled. (The divider output frequency should not also be changed when divider output is changed from enabled state to disabled state.)*

TBTCR (0036_H)

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

DVOEN	Divider output enable/disable	0: Disable 1: Enable	R/W
DVOCK	Divider output ($\overline{\text{DVO}}$) frequency selection [Hz]		

	NORMAL1/2, IDLE1/2 mode					SLOW, SLEEP mode
	DV7CK = 0		DV7CK = 1			
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1		
00	$fc/2^{13}$	$fc/2^{14}$	$fs/2^{5}$	$fs/2^{5}$	DV11	$fs/2^{5}$
01	$fc/2^{12}$	$fc/2^{13}$	$fs/2^{4}$	$fs/2^{4}$	DV10	$fs/2^{4}$
10	$fc/2^{11}$	$fc/2^{12}$	$fs/2^{3}$	$fs/2^{3}$	DV9	$fs/2^{3}$
11	$fc/2^{10}$	$fc/2^{11}$	$fs/2^{2}$	$fs/2^{2}$	DV8	$fs/2^{2}$

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

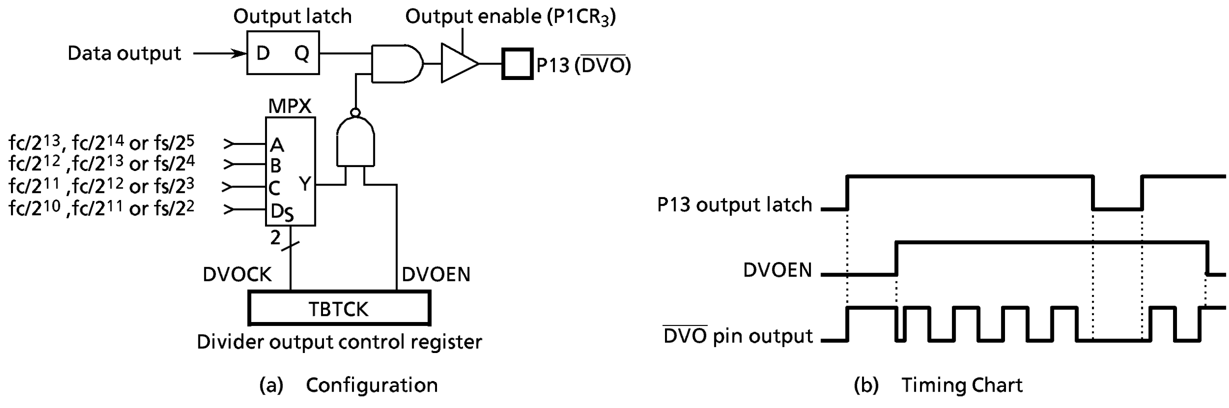
Figure 2-19. Divider Output Control Register

Example: 1.95 kHz pulse output (at fc = 16 MHz, DV1CK = 1)

```
SET   (P1DR). 3            ;   P13 output latch←1
SET   (P1CR). 3            ;   Configures P13 as an output mode
LD    (TBTCR), 10000000B   ;   DVOEN←1, DVOCK←00
```

Table 2-3. Divider Output Frequency (Example: at fc = 16.0 MHz, fs = 32.768 kHz)

DVOCK	Divider output frequency [Hz]				
	NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode
	DV7CK = 0		DV7CK = 1		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
00	1.953 k	976.6	1.024 k	1.024 k	1.024 k
01	3.906 k	1.953 k	2.048 k	2.048 k	2.048 k
10	7.813 k	3.906 k	4.096 k	4.096 k	4.096 k
11	15.625 k	7.813 k	8.192 k	8.192 k	8.192 k


Output latch
Output enable (P1CR3)
Data output
D Q
P13 ($\overline{\text{DVO}}$)
MPX
fc/2^13, fc/2^14 or fs/2^5
fc/2^12 ,fc/2^13 or fs/2^4
fc/2^11 ,fc/2^12 or fs/2^3
fc/2^10 ,fc/2^11 or fs/2^2
A
B
C
D
Y
S
2
DVOCK
DVOEN
TBTCK
Divider output control register
P13 output latch
DVOEN
$\overline{\text{DVO}}$ pin output

(a) Configuration

(b) Timing Chart

MPX : Multiplexer

Figure 2-20. Divider Output

2.6 8-bit Timer/Counter (TC0, 1)

The TMP86CH06 contains 2 channels of 8-bit timer/counter (TC0, 1). These are available for a 16-bit timer/counter by cascade connection. This timer/counter is available for timer, event counter, PWM, PPG, PDO, or warming-up counter.

2.6.1 Configuration

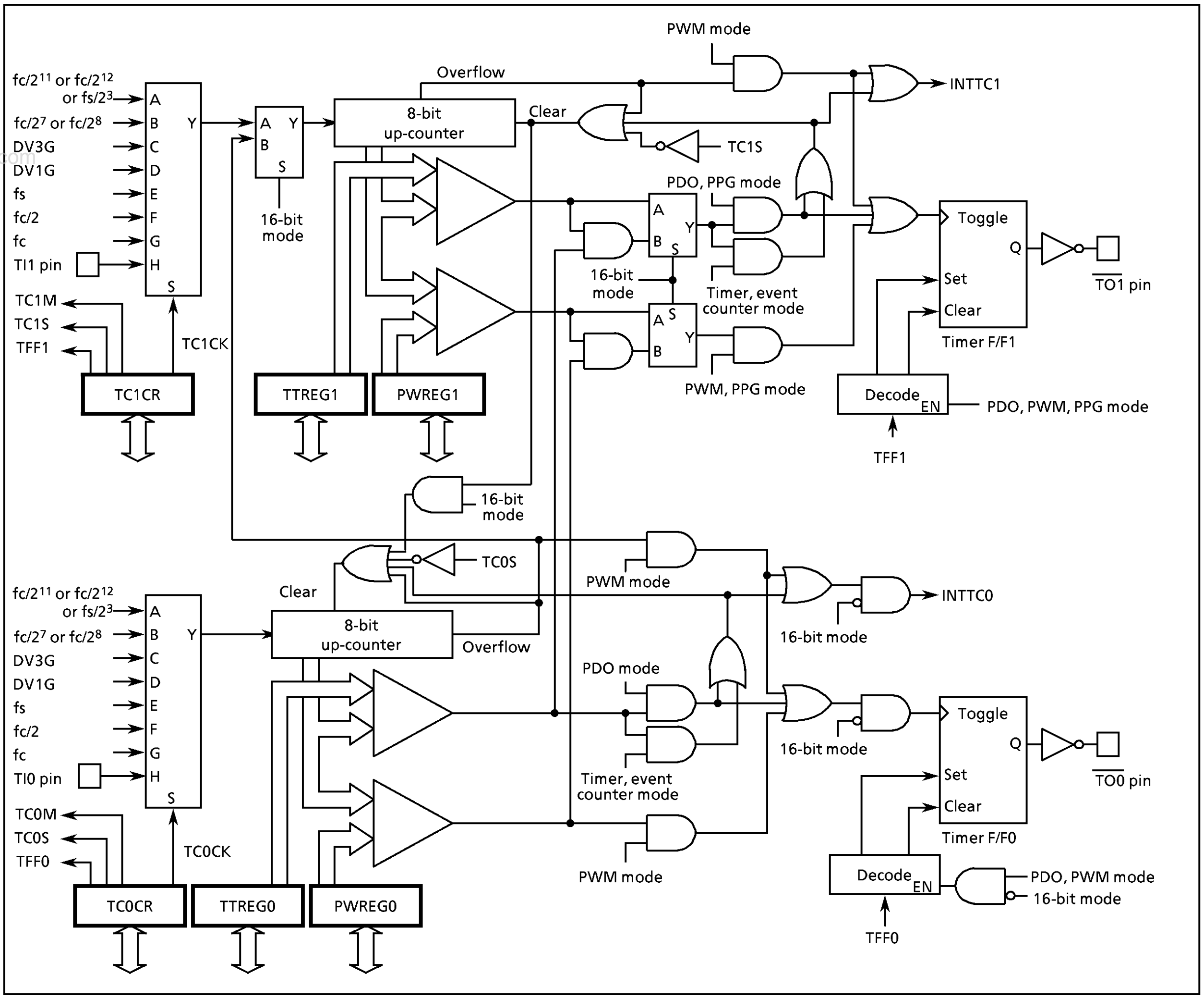


Figure 2-21. 8-bit Timer 0, 1

2.6.2 Control

The timer/counter 0 is controlled by a timer/counter 0 control register (TC0CR) and two 8-bit timer registers (TTREG0 and PWREG0).

Timer 0 Register

	7	6	5	4	3	2	1	0	
TTREG0 (0022_H)									(Initial Value: 1111 1111)
PWREG0 (0024_H)									(Initial Value: 1111 1111)

TC0 Control Register

	7	6	5	4	3	2	1	0	
TC0CR (0020_H)	TFF0	TC0CK			TC0S	TC0M			(Initial Value: 0000 0000)

<table>
<tr><td>TFF0</td><td>Timer F/F 0 management</td><td colspan="7">0: Clear to "0"
1: Set to "1"</td><td rowspan="14">R/W</td></tr>
<tr><td rowspan="11">TC0CK</td><td rowspan="11">Source clock</td><td rowspan="3"></td><td colspan="4">NORMAL1/2, IDLE1/2 mode</td><td rowspan="3">(Note 12)</td><td rowspan="3">SLOW, SLEEP mode</td></tr>
<tr><td colspan="2">DV7CK = 0, DVCK = 00 (Note 11)</td><td colspan="2">DV7CK = 1, DVCK = **</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>000</td><td>$fc/2^{11}$</td><td>$fc/2^{12}$</td><td>$fs/2^{3}$</td><td>$fs/2^{3}$</td><td>DV9</td><td>$fs/2^{3}$</td></tr>
<tr><td>001</td><td>$fc/2^{7}$</td><td>$fc/2^{8}$</td><td>$fc/2^{7}$</td><td>$fc/2^{8}$</td><td>DV5</td><td>–</td></tr>
<tr><td>010</td><td>$fc/2^{5}$</td><td>$fc/2^{6}$</td><td>–</td><td>–</td><td>DV3G</td><td>–</td></tr>
<tr><td>011</td><td>–</td><td>$fc/2^{4}$</td><td>–</td><td>–</td><td>DV1G</td><td>–</td></tr>
<tr><td>100</td><td>fs (Note 9)</td><td>fs (Note 9)</td><td>fs (Note 9)</td><td>fs (Note 9)</td><td>fs (Note 9)</td><td>fs (Note 9)</td></tr>
<tr><td>101</td><td>fc/2</td><td>fc/2</td><td>fc/2</td><td>fc/2</td><td>fc/2</td><td>fc/2</td></tr>
<tr><td>110</td><td>fc</td><td>fc</td><td>fc</td><td>fc</td><td>fc</td><td>fc (Note 10)</td></tr>
<tr><td>111</td><td>TI0 pin input</td><td>TI0 pin input</td><td>TI0 pin input</td><td>TI0 pin input</td><td>TI0 pin input</td><td>TI0 pin input</td></tr>
<tr><td>TC0S</td><td>Timer start/stop</td><td colspan="7">0: Stop operation (Note 13)
1: Start operation</td></tr>
<tr><td>TC0M</td><td>Operating mode</td><td colspan="7">000: 8-bit TC
001: 8-bit PDO
010: 8-bit PWM
011: 16-bit TC/PWM/PPG lower bits
100: Reserved
101: Reserved
110: Reserved
111: Reserved</td></tr>
</table>

Note 1: fc: High-frequency clock [Hz], fs: Low frequency clock [Hz]
Note 2: During TC0 operation, do not change TC0M, TC0CK and TFF0.
Note 3: When TC0 operation is stopped (TC0S = "1" → "0"), do not change TC0M, TC0CK and TFF0. But it is possible to change TC0M, TC0CK and TFF0 at the start timing (TC0S = "0" → "1").
Note 4: The value of TTREG0 which determines interrupt period has to be longer than the period of interrupt routine.
Note 5: When used as 16-bit mode, the operating mode is selected by TC1M (bit 2 to 0 in TC1CR), and TC0M should be set to "011".
Note 6: When used as 16-bit mode, only the source clock is selected by TC0CK, and start of operation and control of F/F are controlled by TC1S and TFF1 (bit 3 and 7 in TC1CR).
Note 7: Selecting source clock depends on the operating mode, refer to Table 2-4 and 2-5 for details.
Note 8: Value of timer register depends on the operating mode, refer to Table 2-6 for details.
Note 9: When used as the NORMAL1 and IDLE1 modes, the TC0 source clock can not use the "fs".
Note 10: When used as the SLOW and SLEEP modes, the "fc" of TC0 source clock can use only "fc warming-up counter" mode.
Note 11: When DVCK is other than "00", see Table 1-1.
Note 12: When selecting DV1G, DV3G, or DV5 for the operating clock, check Table 1-1 in Section 1.5.3, "Timing Generator," to see that the divider output is possible (marked with ○). If the divider output is impossible (marked with ×), do not choose the operating clock.
Note 13: The up counter is cleared when stopped by a command.
Note 14: In the timer, event counter, PDO output, and PPG output modes, do not change the settings of the timer registers (TTREGi, PWREGi) while TCi is operating. Since each of TTREGi and PWREGi is constructed from one-stage register in the timer, event counter, PDO output, and PPG output modes, a newly set values is immediately reflected on the corresponding timer register. Note: i = 0, 1
Note 15: If the timer counter is stopped during output operation in PDO, PWM or PPG mode, the output state is maintained at the state immediately before time/counter is stopped. For changing the level of pin, modify TTFi (bit 7 of the timer counter control register (TCiCR)) after time/counter has been stopped. Do not execute halt of timer/counter and modification of TFFi simultaneously.

Example : Fixes $\overline{TOi}$ output at high level after timer/counter is stopped. Note: i = 0, 1

```
CLR   (TCiCR). 3   ;  Stops timer/counter
CLR   (TCiCR). 7   ;  Sets output to high level output
```

Figure 2-22. Timer 0 Register and Timer/Counter 0 Control Register

The timer/counter 1 is controlled by a timer/counter 1 control register (TC1CR) and two 8-bit timer registers (TTREG1 and PWREG1).

Timer 1 Register

TTREG1 (0023$_H$)		(Initial Value: 1111 1111)
PWREG1 (0025$_H$)		(Initial Value: 1111 1111)

TC1 Control Register

	7	6	5	4	3	2	1	0	
TC1CR (0021$_H$)	TFF0	TC1CK			TC1S	TC1M			(Initial Value: 0000 0000)

<table>
<tr><td>TFF1</td><td>Timer F/F 1 management</td><td colspan="7">0: Clear to “0”
1: Set to “1”</td><td rowspan="14">R/W</td></tr>
<tr><td rowspan="11">TC1CK</td><td rowspan="11">Source clock</td><td rowspan="3"></td><td colspan="4">NORMAL1/2, IDLE1/2 mode</td><td rowspan="3">(Note 12)</td><td rowspan="3">SLOW, SLEEP mode</td></tr>
<tr><td colspan="2">DV7CK = 0, DVCK = 00 (Note 11)</td><td colspan="2">DV7CK = 1, DVCK = **</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>000</td><td>fc/2^{11}</td><td>fc/2^{12}</td><td>fs/2^3</td><td>fs/2^3</td><td>DV9</td><td>fs/2^3</td></tr>
<tr><td>001</td><td>fc/2^7</td><td>fc/2^8</td><td>fc/2^7</td><td>fc/2^8</td><td>DV5</td><td>–</td></tr>
<tr><td>010</td><td>fc/2^5</td><td>fc/2^6</td><td>–</td><td>–</td><td>DV3G</td><td>–</td></tr>
<tr><td>011</td><td>–</td><td>fc/2^4</td><td>–</td><td>–</td><td>DV1G</td><td>–</td></tr>
<tr><td>100</td><td>fs (Note 9)</td><td>fs (Note 9)</td><td>fs (Note 9)</td><td>fs (Note 9)</td><td>fs (Note 9)</td><td>fs (Note 9)</td></tr>
<tr><td>101</td><td>fc/2</td><td>fc/2</td><td>fc/2</td><td>fc/2</td><td>fc/2</td><td>fc/2</td></tr>
<tr><td>110</td><td>fc</td><td>fc</td><td>fc</td><td>fc</td><td>fc</td><td>fc (Note 10)</td></tr>
<tr><td>111</td><td>TI1 pin input</td><td>TI1 pin input</td><td>TI1 pin input</td><td>TI1 pin input</td><td>TI1 pin input</td><td>TI1 pin input</td></tr>
<tr><td>TC1S</td><td>Timer start/stop</td><td colspan="7">0: Stop operation (Note 13)
1: Start operation</td></tr>
<tr><td>TC1M</td><td>Operating mode</td><td colspan="7">000: 8-bit TC
001: 8-bit PDO
010: 8-bit PWM
011: Reserved
100: 16-bit TC upper bits
101: 16-bit TC upper bits (for fc Warming-up Counter)
110: 16-bit PWM upper bits
111: 16-bit PPG upper bits</td></tr>
</table>

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

Note 2: During TC1 operation, do not change TC1M, TC1CK and TFF1.

Note 3: When TC1 operation is stopped (TC1S = "1" → "0"), do not change TC1M, TC1CK and TFF1. But it is possible to change TC1M, TC1CK and TFF1 at the start timing (TC1S = "0" → "1").

Note 4: The value of TTREG1 which determines interrupt period has to be longer than the period of interrupt routine.

*Note 5: When TC1M is selected to "1**" (16-bit mode), the source clock is automatically selected to the over-flowing signal of TC0 counter.*

Note 6: When used as 16-bit mode, the operating mode is selected by TC1M (bit 2 to 0 in TC1CR), and TC0M should be set to "011".

Note 7: When used as 16-bit mode, only the source clock is selected by TC0CK, and start of operation and control of F/F are controlled by TC1S and TFF1 (bit 3 and 7 in TC1CR).

Note 8: Selecting source clock depends on the operating mode, refer to Table 2-4 and 2-5 for details.

Note 9: Value of timer register depends on the operating mode, refer to Table 2-6 for details.

Note 10: When used as the NORMAL1 and IDLE1 modes, the TC1 source clock can not use the "fs".

Note 11: When DVCK = "00", fc/2^3 cannot be selected for the operating clock.

Note 12: When selecting DV1G, DV3G, or DV5 for the operating clock, check Table 1-1 in Section 1.5.3, "Timing Generator," to see that the divider output is possible (marked with ○). If the divider output is impossible (marked with ×), do not choose the operating clock.

Note 13: The up counter is cleared when stopped by a command.

Note 14: In the timer, event counter, PDO output, and PPG output modes, do not change the settings of the timer registers (TTREGi, PWREGi) while TCi is operating. Since each of TTREGi and PWREGi is constructed from one-stage register in the timer, event counter, PDO output, and PPG output modes, a newly set values is immediately reflected on the corresponding timer register. Note: i = 0, 1

Note 15: If the timer counter is stopped during output operation in PDO, PWM or PPG mode, the output state is maintained at the state immediately before time/counter is stopped. For changing the level of pin, modify TTFi (bit 7 of the timer counter control register (TCiCR)) after time/counter has been stopped. Do not execute halt of timer/counter and modification of $\overline{\text{TFFi}}$ simultaneously.

Example : Fixes $\overline{\text{TOi}}$ output at high level after timer/counter is stopped. Note: i = 0, 1

```
CLR    (TCiCR). 3    ;  Stops timer/counter
CLR    (TCiCR). 7    ;  Sets output to high level output
```

Figure 2-23. Timer 1 Register and Timer/Counter 1 Control Register

Table 2-4. Operating mode and available source clock (NORMAL1/2, IDLE1/2 mode)

TCnCK Operating Mode	000	001	010	011	100	101	110	111
8-bit Timer	○	○	○	○	○	—	—	—
8-bit Event Counter	—	—	—	—	—	—	—	○
8-bit PDO	○	○	○	○	○	—	—	○
8-bit PWM	○	○	○	○	○	○	○	○
16-bit Timer	○	○	○	○	○	—	—	—
16-bit Event Counter	—	—	—	—	—	—	—	○
fs Warming-up Counter	—	—	—	—	○	—	—	—
16-bit PWM	○	○	○	○	○	○	○	○
16-bit PPG	○	○	○	○	○	—	—	○

Note 1: For 16-bit operation (16-bit Timer/Event Counter, fc Warming-up Counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bits (TC0CK).

Note 2: n = 0, 1

Note 3: When used as the NORMAL1 and IDLE1 modes, the source clock can not use the "fs".

Table 2-5. Operating mode and available source clock (Under SLOW1/2 mode, SLEEP1/2 mode)

TCnCK Operating Mode	000	001	010	011	100	101	110	111
8-bit Timer	○	—	—	—	—	—	—	—
8-bit Event Counter	—	—	—	—	—	—	—	○
8-bit PDO	○	—	—	—	—	—	—	○
8-bit PWM	○	—	—	—	○	—	—	○
16-bit Timer	○	—	—	—	—	—	—	—
16-bit Event Counter	—	—	—	—	—	—	—	○
fc Warming-up Counter	—	—	—	—	—	—	○	—
16-bit PWM	○	—	—	—	○	—	—	○
16-bit PPG	○	—	—	—	—	—	—	○

Note 1: For 16-bit operation (16-bit Timer/Event Counter, fc Warming-up Counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bits (TC0CK).

Note 2: n = 0, 1

Table 2-6. Restriction against the Rate for Comparing Registers

Operating Mode	Authorized Rate for Register
8-bit Timer/Event Counter	1 ≦ (TTREGn) ≦ 255
8-bit PDO	1 ≦ (TTREGn) ≦ 255
8-bit PWM	2 ≦ (PWREGn) ≦ 254
16-bit Timer/Event Counter	1 ≦ (TTREG1, 0) ≦ 65535
fc Warming-up Counter	256 ≦ (TTREG1, 0) ≦ 65535 (Note 2)
16-bit PWM	2 ≦ (PWREG1, 0) ≦ 65534
16-bit PPG	1 ≦ (PWREG1, 0) < (TTREG1, 0) ≦ 65535

Note 1: n = 0, 1

Note 2: When used at the "fc Warming-up counter" mod, the timer operating mode must use the 16-bit timer mode. That time, the timer register uses higher 8-bits and the lower 8-bits are ignored. But the timer register must be set both of high and low bits.

2.6.3 Function

Timer/Counter 0 and 1 have eight operating modes: 8-bit timer, 8-bit external trigger timer, 8-bit programmable divider output mode, 8-bit pulse width modulation output mode, 16-bit timer, 16-bit external trigger timer, 16-bit pulse width modulation output mode, 16-bit programmable pulse generator output mode.
16-bit timer mode can use Timer counter 0 and 1 by cascade connection.

(1) 8-bit Timer Mode (Timer/Counter 0 and 1)

In this mode, counting up is performed using the internal clock. The contents of TTREGi are compared with the contents of up-counter. If a match is found, an INTTCi interrupt is generated, and the counter is cleared to "0". Counting up resumes after the counter is cleared.

Note 1: In the timer mode, always write TCiCR<TFFi> to "0". If TFFi is set to "1", unexpected pulse may be output from $\overline{\text{TOi}}$ pin.
Note 2: In the timer mode, do not change the setting of timer registers (TTREGi) while timer/counter is operating. Since TTREGi is configured as one-stage register, a newly set value is immediately reflected on the timer register.
Note 3: i = 0, 1

Table 2-7. Timer/Counter 0 and 1 Source Clock (Internal Clock, DV1CK = 0, DVCK = 01)

Source clock			Resolution		Maximum time setting	
NORMAL1/2, IDLE1/2 modes		SLOW1/2, SLEEP1/2 modes	At fcgck = 16 MHz	At fs = 32.768 kHz	At fcgck = 16 MHz	At fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
fc/2^{11} [Hz]	fs/2^3 [Hz]	fs/2^3 [Hz]	128 μs	244.14 μs	32.8 ms	62.5 ms
fc/2^7 (DV5)	–	–	8 μs	–	2.0 ms	–
fc/2^5 (DV3G)	–	–	2 μs	–	510 μs	–
fc/2^4 (DV1G)	–	–	1 μs	–	255 μs	–

Example : Sets the timer mode with source clock fc/2^7 [Hz] and generates an interrupt 80 μs later (at fc = 16 MHz).

```
LDW   (TTREG1), 0AH          ;  Sets the timer register (80 μs ÷ 2^7/fc = 0AH)
DI
SET   (EIRL). EF7            ;  Enables INTTC1 interrupt
EI
LD    (TC1CR), 00010000B     ;  Sets the 8-bit timer mode and source clock (fc/2^7)
LD    (TC1CR), 00011000B     ;  Starts TC1
```

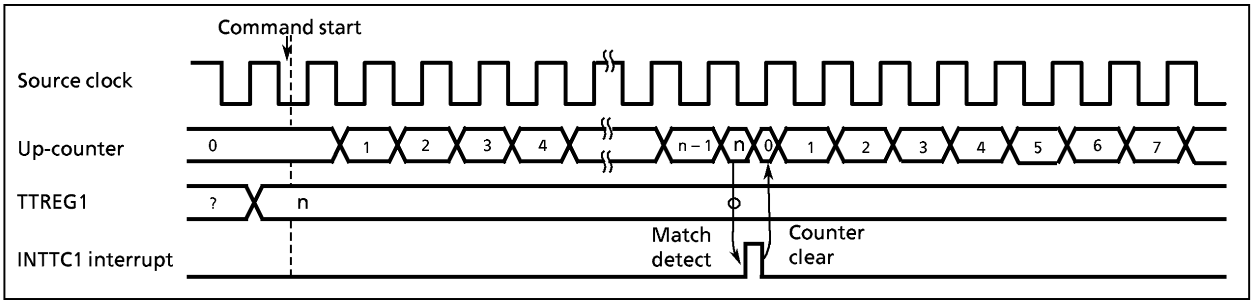


Figure 2-24. 8-bit Timer Mode Timing Chart (In Case of Timer/Counter 1)

(2) 8-bit Event Counter Mode (Timer/Counter 0 and 1)

In this mode, events are counted on the falling edge of TIi pin input. The contents of TTREGi are compared with the contents of up-counter. If a match is found, an INTTCi interrupt is generated, and the counter is cleared. The maximum applied frequency is fcgck/2^4 [Hz] in NORMAL 1/2 or IDLE 1/2 mode, and fs/2^4 [Hz] in SLOW 1/2 or SLEEP 1/2 mode.
Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.

	NORMAL/IDLE	SLOW/SLEEP
High level width	fcgck/2^3	fs/2^3
Low level width	fcgck/2^3	fs/2^3

Note 1: In the event counter mode, always write TCiCR<TFFi> to "0". If TFFi is set to "1", unexpected pulse may be output from $\overline{\text{TOi}}$ pin.
Note 2: In the event counter mode, do not change the setting of timer registers (TTREGi) while timer/counter is operating. Since TTREGi is configured as one-stage register, a newly set value is immediately reflected on the timer register.
Note 3: i = 0, 1

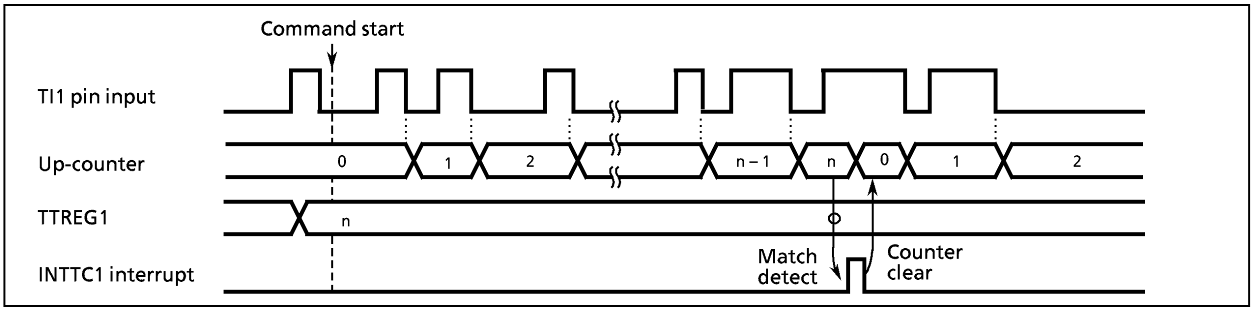


Figure 2-25. Event Counter Mode Timing Chart (In Case of Timer/Counter 1)

(3) Programmable Divider Output (PDO) Mode (Timer/Counter 0 and 1)

The internal clock is used for counting up. The contents of TTREGi are compared with the contents of the up-counter. Timer F/F i output is toggled and the counter is cleared each time a match is found. Timer F/F i output is inverted and output to the $\overline{\text{TOi}}$ pin. When used as a this mode, respective output latch should be set to "1". This mode can be used for 50% duty pulse output. Timer F/F i can be initialized by program, and it is initialized to "0" during reset. An INTTCi interrupt is generated each time the $\overline{\text{TOi}}$ output is toggled.

Note 1: In the programmable divider output (PDO) mode, do not change the setting of timer registers (TTREGi) while timer/counter is operating. Since TTREGi is configured as one-stage register, a newly set value is immediately reflected on the timer register.
Note 2: If PDO output is stopped during output operation, the output state is maintained at the state immediately before timer/counter is stopped. For changing the level of $\overline{\text{TOi}}$ pin, modify TCiCR<TFFi> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFi simultaneously.

Example: Fixes $\overline{\text{TOi}}$ output at high level after timer/counter is stopped
CLR (TCiCR).3 ; Stops timer/counter.
CLR (TCiCR).7 ; Sets $\overline{\text{TOi}}$ output to high level output

Note 3: i = 0, 1

Example : Output a 1024 Hz pulse (at fc = 16 MHz, in case of TC1)

```
SET  (P1DR). 4              ;  P14 output latch ← 1
SET  (P1CR). 4
LD   (TTREG1), 7AH          ;  (1/1024 ÷ 2^7/fc) ÷ 2 = 7A_H
LD   (TC1CR), 00010001B     ;  Sets the 8-bit PDO mode and source clock (fc/2^7)
LD   (TC1CR), 00011001B     ;  Starts TC1
```

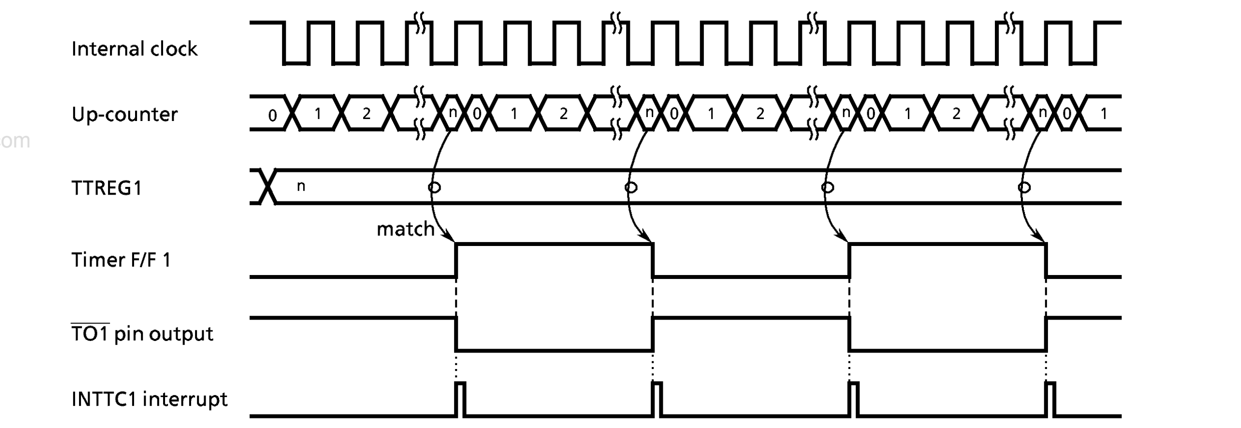


Figure 2-26. 8-bit Timing Chart for PDO Mode (In Case of Tmer/Counter 1)

(4) 8-bit Pulse Width Modulation (PWM) Output Mode (Timer/Counter 0 and 1)

PWM output with a resolution of 8 bits is possible. The internal clock is used for counting up. The contents of PWREGi are compared with the contents of up-counter. If a match is found, the timer F/F i output is toggled. The counter continues counting. And, when an overflow occurs, the timer F/F i output is again toggled and the counter is cleared. Timer F/F i output is inverted and output to the $\overline{\text{TOi}}$ pin. An INTTCi interrupt is generated when an overflow occurs.

PWREGi is configured a 2-stage shift register and, during output, will not switch until one output cycle is completed even if PWREGi is overwritten; therefore, output can be altered continuously. Also, the first time, PWREGi is shifted by setting TCiS (bit 4 in TCiCR) to "1" after data are loaded to PWREGi.

Note 1: In PWM mode, write to the timer register PWREGi immediately after an INTTCi interrupt is generated (normally during the INTTCi interrupt service routine). If writing to PWREGi and INTTCi interrupt occur at the same time, the unstable value being written is shifted. This may cause pulses different from the set value to be output until the next INTTCi interrupt is generated.

Note 2: If PWM output is stopped during output operation, the output state is maintained at the state immediately before timer/counter is stopped. For changing the level of $\overline{\text{TOi}}$, modify TCiCR<TTFi> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFi simultaneously.

Example: Fixes $\overline{\text{TOi}}$ output at high level after timer/counter is stopped
CLR (TCiCR).3 ; Stops timer/counter.
CLR (TCiCR).7 ; Sets $\overline{\text{TOi}}$ output to high level output

Note 3: Before starting STOP mode, disable PWM output. When the timer/counter is enabled and fc, fc/2 or fs is selected as the source clock, pulse is output from PWM pin during warming-up after releasing STOP mode.

Note 4: i = 0, 1

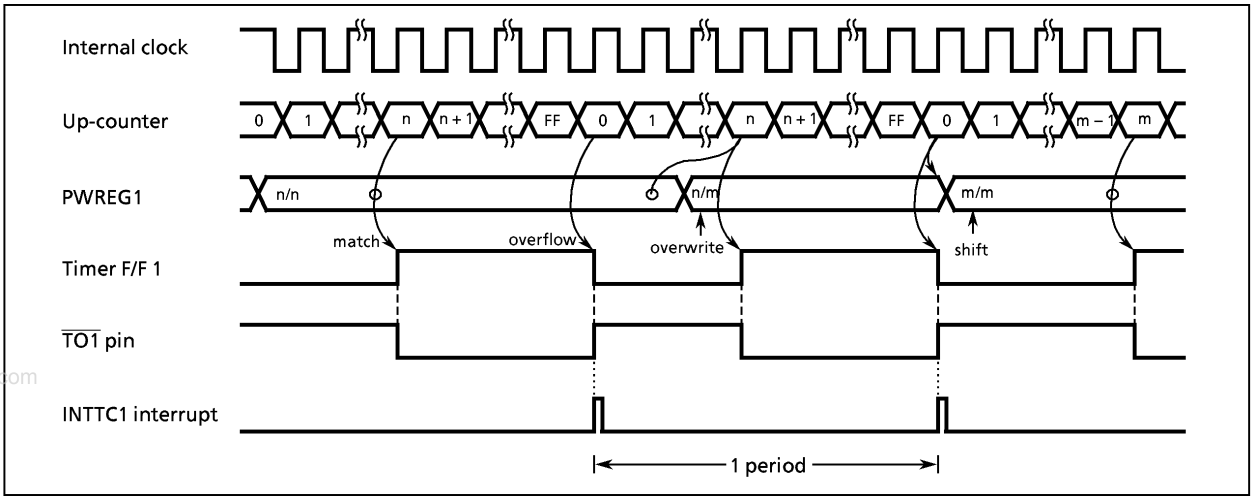


Figure 2-27. Timing Chart for PWM Mode (In Case of TC1)

Table 2-8. PWM Output Mode (DV1CK = 0, DVCK = 01)

Source clock			Resolution		Maximum setting time	
NORMAL1/2, IDLE1/2 mode		SLOW1/2, SLEEP1/2 mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
fc/2^{11} [Hz]	fs/2^3 [Hz]	fs/2^3 [Hz]	128 μs	244.14 μs	32.8 ms	62.5 ms
fc/2^7 (DV5)	–	–	8 μs	–	2.05 ms	–
fc/2^5 (DV3G)	–	–	2 μs	–	512 μs	–
fc/2^4 (DV1G)	–	–	1 μs	–	256 μs	–
fs	fs	fs	30.5 μs	30.5 μs	7.81 ms	7.81 ms
fc/2	fc/2	–	125 ns	–	32 μs	–
fc	fc	–	62.5 ns	–	16 μs	–

(5) 16-bit Timer Mode (Timer/counter 0 and 1)

In this mode, counting up is performed using the internal clock.
Timer/counter 0 and 1 are also available as a 16-bit timer mode by cascade connection.

16 bit timer mode of Timer/counter 0 and 1

If a match is found, the INTTC1 interrupt is generated and the counter is cleared to “0”. Counting up resumes after the counter is cleared. The timer register should write to the TTREG0 more first than TTREG1. The timer register must not write only either TTREG0 or TTREG1.

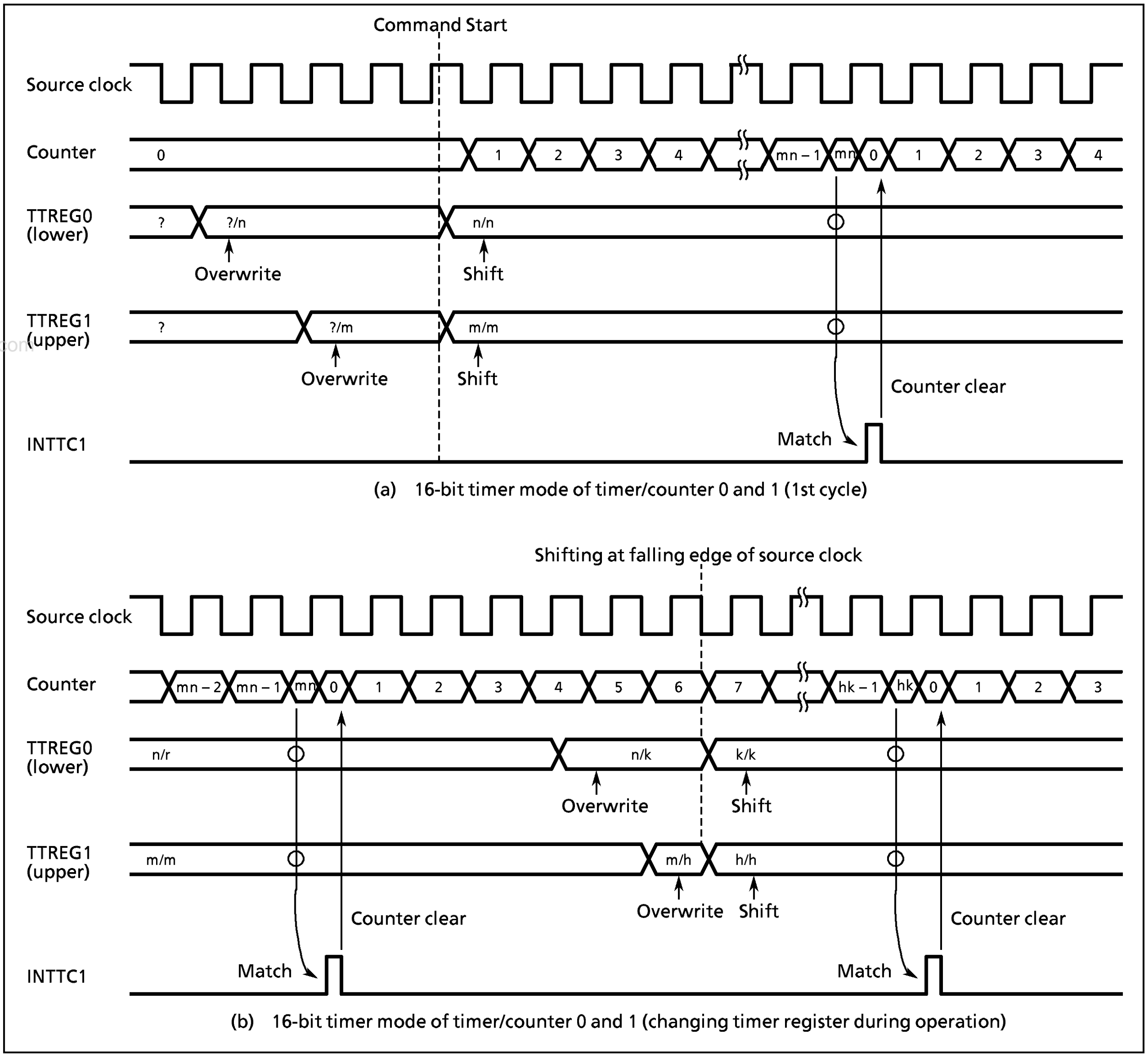
Note 1: In the 16-bit timer mode, always write TCiCR<TFFi> to “0”. If TFFi is set to “1”, unexpected pulse may be output from $\overline{\text{TO}i}$ pin.
Note 2: In the timer mode, do not change the setting of timer registers (TTREGi) while timer/counter is operating. Since TTREGi is configured as one-stage register, a newly set value is immediately reflected on the timer register.
Note 3: i = 0, 1

Table 2-9. Source Clock of 16-bit Timer Mode (DV1CK = 0, DVCK = 01)

Source clock			Resolution		Maximum setting time	
NORMAL1/2, IDLE1/2 mode		SLOW1/2, SLEEP1/2 mode	At fc = 16 MHz	At fs = 32.768 kHz	At fc = 16 MHz	At fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
fc/2^{11}	fs/2^3 [Hz]	fs/2^3	128 μs	244.1 μs	8.38 s	16 s
fc/2^7 (DV5)	–	–	8 μs	–	524.3 ms	–
fc/2^5 (DV3G)	–	–	2 μs	–	131.1 ms	–
fc/2^4 (DV1G)	–	–	1 μs	–	65.5 ms	–
–	–	–	–	–	–	–
fc/2	fc/2	–	125 ns	–	8.2 ms	–
fc	fc	–	62.5 ns	–	4.1 ms	–

Example: Set the 16-bit timer mode with source clock fc/2^7 (DV5) [Hz] and generates an interruput 300 [ms] later (at fc = 16 [MHz])

```
LDW    (TTREG0), 927CH    ;  Sets the timer register (300 ms ÷ 2^7/fc = 927C_H)
DI
SET    (EIRL). EF7        ;  Enable INTTC1 interrupt
EI
LD     (TC0CR), 13H       ;  Sets the TC0 mode and source clock
LD     (TC1CR), 04H       ;  Set the 16-bit timer mode (upper)
LD     (TC1CR), 0CH       ;  Starts timer/counter
```



(a) 16-bit timer mode of timer/counter 0 and 1 (1st cycle)

(b) 16-bit timer mode of timer/counter 0 and 1 (changing timer register during operation)

Figure 2-28. Timing Chart for 16-bit Timer Mode (Timer/Counter 0 and 1)

(6) 16-bit Event Counter Mode (Timer/counter 0 and 1)

In this mode, event are counted on the falling edge of the TI0 pin input. Timer/counter 0 and 1 are also available as a 16-bit Event counter mode by cascade connection.

16 bit event counter mode of Timer/counter 0 and 1

If a match is found, the INTTC1 interrupt is generated and the counter is cleared to "0". After the counter is cleared, counting up resumes every falling edge of TI0 input. The timer register should write to the TTREG0 more first than TTREG1. The timer register must not write only either TTREG0 or TTREG1. When the TTREG1 is written during counter continuation, the timer register is changed by next falling edge of TI0 input at TTREG0 and TTREG1 same time. The timer/counter is counting by value of last timer register until the TTREG1 is written. Also, the first time, the timer register is changed by the TC1S (bit 3 in TC1CR) set to "1" after data are loaded to TTREG0/1.

Note 1: In the event counter mode, always write TC0CR<TFF0> to "0". If TFF0 is set to "1", unexpected pulse may be output from $\overline{\text{TOi}}$ pin.
Note 2: In the event counter mode, do not change the setting of timer registers (TTREGi) while timer/counter is operating. Since TTREGi is configured as one-stage register, a newly set value is immediately reflected on the timer register.
Note 3: i = 0, 1

(7) 16-bit Pulse Width Modulation (PWM) Output Mode (Timer/counter 0 and 1)

PWM output with a resolution of 16 bits is possible. Timer/counter 3 and 4 (5 and 6) are also available as a 16-bit PWM output mode by cascade connection

16 bit PWM output mode of Timer/counter 0 and 1

The contents of PWREG0/1 are compared with the contents of up-counter. If a match is found, the timer F/F 1 output is toggled. The counter continues counting. And, when an overflow occurs, the timer F/F 1 output is again toggled and the counter is cleared. Timer F/F 1 output is inverted and output to the $\overline{\text{TO1}}$ pin. An INTTC1 interrupt is generated when an overflow occurs. When used as $\overline{\text{TO1}}$ pin, respective output latch should be set to "1". PWREG0/1 are configured a 2-stage shift register and, during output, will not switch until one output cycle is completed even if PWREG0/1 are overwritten. Therefore, output can be altered continuously. Also, the first time, the timer register is changed by the TC1S (bit 3 in TC1CR) set to "1" after data are loaded to PWREG0/1.

The timer register should write to the PWREG0 more first than PWREG1. The timer register must not write only either PWREG0 or PWREG1.

Note 1: In PWM mode, write to the timer register PWREG1, 0 immediately after an INTTC1 interrupt is generated (normally during the INTTC1 interrupt service routine). If writing to PWREG1, 0 and INTTC1 interrupt occur at the same time, the unstable value being written is shifted. This may cause pulses different from the set value to be output until the next INTTC1 interrupt is generated.
Note 2: If PWM output is stopped during output operation, the output state is maintained at the state immediately before timer/counter is stopped. For changing the level of $\overline{\text{TO1}}$ modify TC1CR<TTF1> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFF1 simultaneously.

Example: Fixes $\overline{\text{TO1}}$ output at high level after timer/counter is stopped
CLR (TC1CR).3 ; Stops timer/counter
CLR (TC1CR).7 ; Sets $\overline{\text{TO1}}$ output to high level output

Note 3: Before starting STOP mode, disable PWM output. When the timer/counter is enabled and fc, fc/2 or fs is selected as the source clock, pulse is output from $\overline{\text{TO1}}$ pin during warming-up after releasing STOP mode.

Table 2-10. 16-bit PWM Output Mode (DV1CK = 0, DVCK = 01)

Source clock			Resolution		Maximum setting time	
NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode	At fc = 16 MHz	At fs = 32.768 kHz	At fc = 16 MHz	At fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
fc/2^{11}	fs/2^3 [Hz]	fs/2^3	128 μs	244.1 μs	8.38 s	16 s
fc/2^7 (DV5)	–	–	8 μs	–	524.3 ms	–
fc/2^5 (DV3G)	–	–	2 μs	–	131.1 ms	–
fc/2^4 (DV1G)	–	–	1 μs	–	65.5 ms	–
fs	fs	fs	30.5 μs	30.5 μs	2 s	2 s
fc/2	fc/2	–	125 ns	–	8.2 ms	–
fc	fc	–	62.5 ns	–	4.1 ms	–

Example: Extract the pulse, whose term and "high" width is 131.1 ms and 4 ms respectively, from P14 width 16-bit PWM mode (at fc = 16 MHz, DV7CK = 0)

```
SET     (P1DR). 4           ;  Sets P14 output data latch to "1"
SET     (P1CR). 4
LDW     (PWREG0), 07D0H     ;  Sets pulse width
LD      (TC0CR), 23H        ;  Sets the mode and source clock (DV3G)
LD      (TC1CR), 56H        ;  Sets the TFF1 to "1" and sets the 16-bit PWM mode
                               (upper)
LD      (TC1CR), 5EH        ;  Starts timer/counter
```

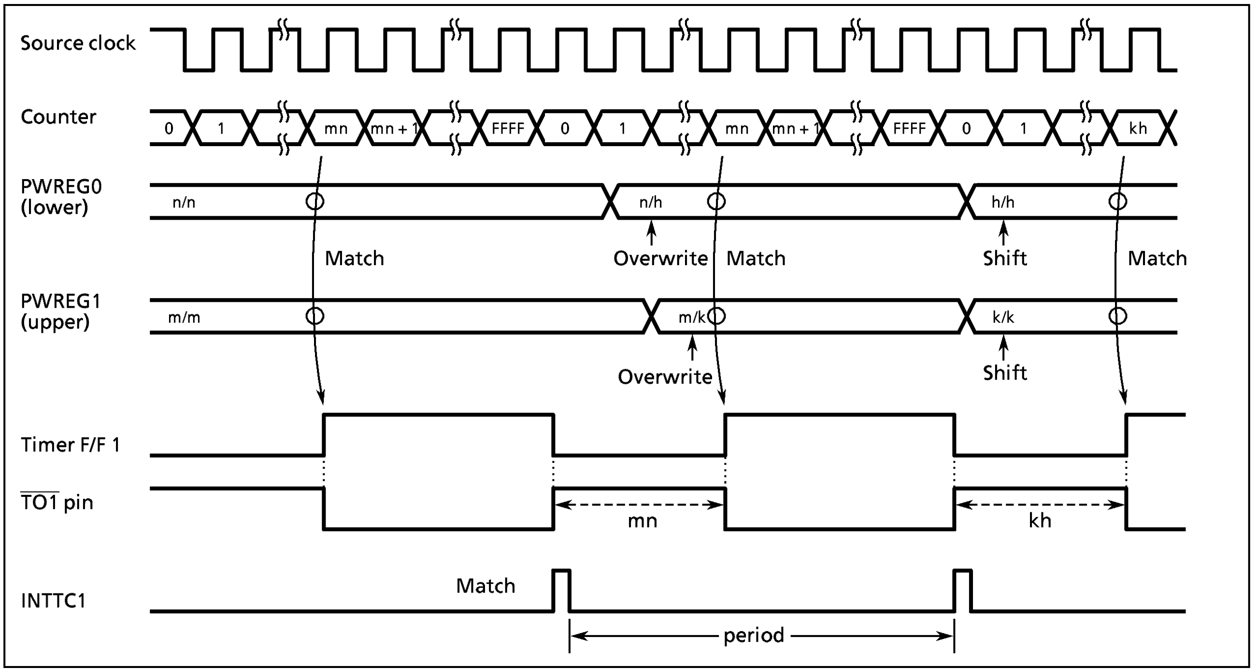


Figure 2-29. Timing Chart of 16-bit PWM Mode (Timer/Counter 3 and 4)

(8) 16-bit Programmable Pulse Generate (PPG) output mode (TC0 + 1)

PPG output with a resolution of 16 bits is possible. Timer/counter 0 and 1 are also available as a 16-bit PPG output mode by cascade connection.

16 bit PPG output mode of Timer/counter 0 and 1

First, the contents of PWREG0/1 are compared with the contents of up-counter. If a match is found, the timer F/F 1 output is toggled. The INTTC1 interrupt is generated at this time. Next, timer F/F 1 is again toggled and the counter is cleared by matching with TTREG0/1. The INTTC1 interrupt is again generated when an overflow occurs.

When used as $\overline{\text{PPG1}}$ pin, respective output latch should be set to "1". During reset, the F/F 1 is initialized to "0".

The F/F 1 output is configured by TFF1 (bit 7 in TC1CR). Therefore, the $\overline{\text{TO1}}$ can output either output high or output low at first time. The timer register should write to the PWREG0/TTREG0 more first than PWREG1/TTREG1. The timer register must not write only either PWREG0/TTREG0 or PWREG1/TTREG1.

When the TTREG1/PWREG1 are written during counter continuation, the timer register is changed by next falling edge of source clock at PWREG0/TTREG0 and PWREG1/TTREG1 same time. The timer/counter is counting by value of last timer register until the PWREG1/TTREG1 is written.

Example: Extract the pulse, whose term and "high" width is 65.54 ms and 4 ms respectively, from P14 with 16-bit PPG mode (at fc = 16 MHz, DV7CK = 0)

```
SET     (P1DR). 4           ;  Sets P14 output data latch to "1"
SET     (P1CR). 4
LDW     (PWREG0), 07D0H     ;  Sets pulse width
LDW     (TTREG0), 8002H     ;  Sets pulse term
LD      (TC0CR), 23H        ;  Sets the mode and source clock (DV3G)
LD      (TC1CR), 57H        ;  Sets the TFF1 to "1" and sets the 16-bit PPG mode
                               (upper)
LD      (TC1CR), 5FH        ;  Starts timer/counter
```

Note 1: In the programmable pulse generate (PPG) mode, do not change the setting of timer registers (PWREGi, TTREGi) while timer/counter is operating. Since PWREGi, TTREGi are configured as one-stage register, a newly set value is immediately reflected on the timer register.

Note 2: If PPG output is stopped during output operation, the output state is maintained at the state immediately before timer/counter is stopped. For changing the level of $\overline{\text{TO1}}$, modify TCiCR<TTFi> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFi simultaneously.

Example: Fixes $\overline{\text{TO1}}$ output at high level after timer/counter is stopped

```
CLR (TC1CR).3  ;  Stops timer/counter
CLR (TC1CR).7  ;  Sets TO1 output to high level output
```

Note 3: i = 0, 1

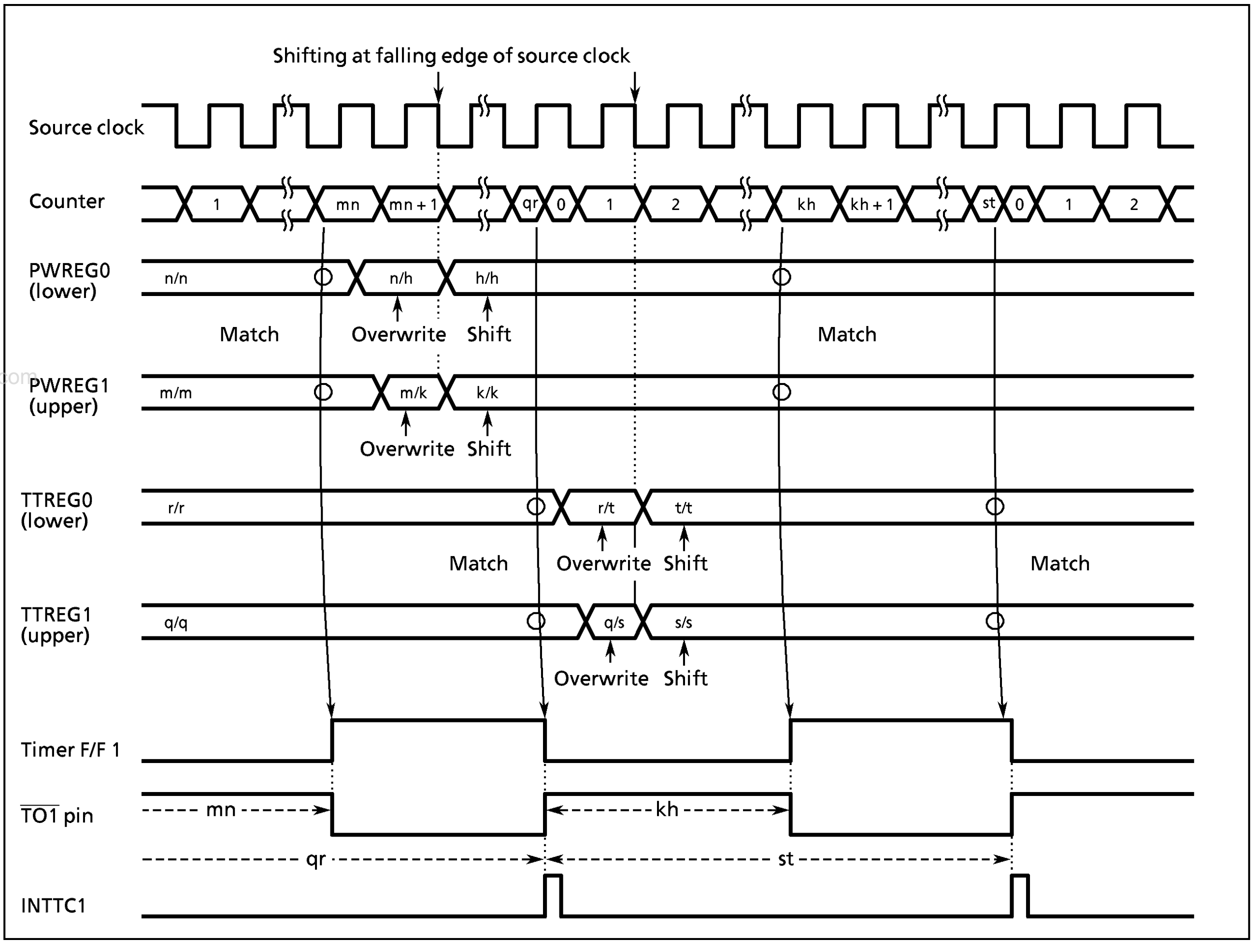


Figure 2-30. Timing Chart of 16-bit PPG Mode (Timer/Counter 0 and 1)

(9) Warming-up counter mode

In this mode, the warming-up period for switching the main system clock can be generated. Timer/counter 0 and 1 are used as a 16-bit timer by cascade connection.
There are 2 modes in warming-up counter mode, one is a mode from NORMAL to SLOW and the other is a mode from SLOW to NORMAL.

Note 1: In the warming-up mode, always write TCiCR<TFFi> to "0". If TFFi is set to "1", unexpected pulse may be output from $\overline{TOi}$ pin.
Note 2: In the warming-up mode, the lower 8 bits of TTREG1, 0 are ignored and an interrupt is generated by matching the upper 8 bits.
Note 3: i = 0, 1

a. Warming-up counter mode for low-frequency (NORMAL1→NORMAL2→SLOW2→SLOW1)

In this mode, it can obtain the warming-up period till the oscillation for low-frequency (fs) is stabilized.
Before timer/counter is started, turn on low-frequency oscillation by setting SYSCR2<XTEN> to "1".
After timer/counter is started by setting TC1CR<TC1S>, the contents of TTREG1, 0 are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to "0".
In the interrupt service program, stop the timer/counter and change the system clock to low-frequency clock by setting SYSCR2<SYSCK> to "1".
After that, halt the high-frequency oscillation by clearing SYSCR2<XEN> to "0".

Table 2-13. Warming-up period for low-frequency oscillation
(at fs = 32.768 kHz)

Min (at TTREGm, n = 0100H)	Max (at TTREGm, n = FF00H)
7.8 ms	1.99 s

Example: Switching to the SLOW1 mode after low-frequency clock has stabilized by using TC1, 0.

```
          SET   (SYSCR2).6       ; SYSCR2<XTEN> ← "1"
          LD    (TC0CR), 43H     ; TFF0 = "0", fs for source clock, sets 16-bit mode
          LD    (TC1CR), 05H     ; TFF1 = "0", sets warming-up counter mode
          LD    (TTREG0), 8000H  ; sets warming-up time (depend on oscillator characteristics)
          DI                     ; IMF ← "0"
          SET   (EIRL).7         ; Enables INTTC1
          EI                     ; IMF ← "1"
          SET   (TC1CR). 3       ; Starts TC1, 0
           :         :
PINTTC1:  CLR   (TC1CR).3        ; Stops TC1, 0
          SET   (SYSCR2).5       ; SYSCR2<SYSCK> ← "1"
                                   (Switches the main system clock to the low-frequency clock)
          CLR   (SYSCR2).7       ; SYSCR2<XEN> ← "0" (Turns off low-frequency oscillation)
          RETI
           :         :
VINTTC1:  DW    PINTTC1          ; INTTC1 vector table
```

b. Warming-up counter mode for high-frequency (SLOW1→SLOW2→NORMAL2→NORMAL1)

In this mode, it can obtain the warming-up period till the oscillation for high-frequency (fc) is stabilized.
Before timer/counter is started, turn on high-frequency oscillation by setting SYSCR2<XEN> to "1".
After timer/counter is started by setting TC1CR<TC1S>, the contents of TTREG1, 0 are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to "0".
In the interrupt service program, stop the timer/counter and change the system clock to high-frequency clock by clearing SYSCR2<SYSCK> to "0".
After that, halt the low-frequency oscillation by clearing SYSCR2<XTEN> to "0".

Table 2-12. Warming-up period for high-frequency (at fc = 16 MHz)

Min (at TTREGm, n = 0100H)	Max (at TTREGm, n = FF00H)
16 μs	4.08 ms

Esample: Switching to the NORMAL1 mode after high-frequency clock has stabilized by using TC1, 0.

```
          SET   (SYSCR2).7        ; SYSCR2<XEN> ← "1"
          LD    (TC0CR), 63H      ; TFF0 = "0" , fc for source clock, sets 16-bit mode
          LD    (TC1CR), 05H      ; TFF1 = "0" , sets warming-up counter mode
          LD    (TTREG0), 0F800H  ; Sets warming-up time (depend on oscillator characteristics)
          DI                      ; IMF ← "0"
          SET   (EIRL).7          ; Enables INTTC1
          EI                      ; IMF ← "1"
          SET   (TC1CR).3         ; Starts TC1, 0
           :       :
PINTTC1:  CLR   (TC1CR).3         ; Stops TC1, 0
          CLR   (SYSCR2).5        ; SYSCR2<SYSCK> ← "0"
                                    (Switches the main system clock to the high-frequency clock)
          CLR   (SYSCR2).6        ; SYSCR2<XTEN> ← "0"
                                    (Turns off high-frequency oscillation)
          RETI
           :       :
VINTTC1:  DW    PINTTC1           ; INTTC1 vector table
```

2.7 Extended Timer-Counter (ETC0)

The TMP86CH06 contains 16-bit extended timer-counter, which is accompanied by terminals for capturing input and outputting comparison. The alternative of internal or external input is to be source clock.

2.7.1 Configuration

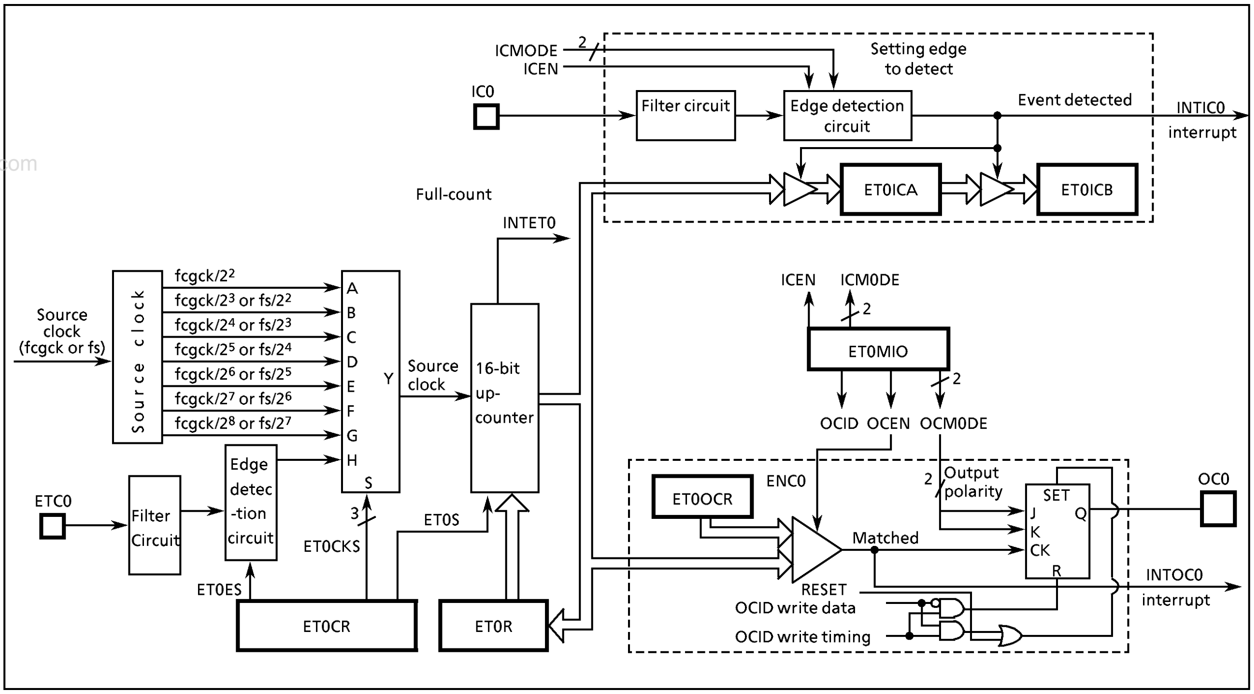


Figure 2-31. Extended Timer-Counter 0 (ETC0)

2.7.2 Controlling

The extended timer-counter (ETC0) is controlled on the following registers.

Extended Timer-Counter Register

This 16-bit timer register is to set and record the ETC0 counter value.

It is able to read while counting.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ET0R (0012, 13_H)	$ET0R_H$ (0013_H)								$ET0R_L$ (0012_H)							

Read/Write (Initial value: 0000 0000 0000 0000)

Note 1: Data written on address (0012_H), does not reach lower byte of the ETC0 register until data on upper byte of it is set because of buffering. Writing only to lower byte is not allowed.

Note 2: On reading the register, the lower byte read operation latches the timer-counter value. Therefore, always read the lower byte first, and the upper byte last.

Note 3: Writing $FFFF_H$ to the ETC0 register generates an immediate interrupt request signal.

Note 4: The lower byte of the ETC0 register and the lower byte of the ETC0 compare register share their buffering register. Therefore, do not execute write instruction against the ETC0 compare register, between writing on lower byte of the ETC0 register and upper byte of it.

External Timer-counter Control Register

Principal mode register for ETC0

	7	6	5	4	3	2	1	0
ET0CR (0010_H)	ET0S				ET0ES	ET0CKS		

(Initial value: 0∗∗∗ 0000)

ET0S	ETC0 start/stop	0: Stop 1: Start			R/W
ET0ES	Count edge selection for event counter mode	0: Rising edge 1: Falling edge			
ET0CKS	Source clock for ETC0		NORMAL1/2, IDLE1/2 mode	SLOW, SLEEP mode	
		000	fcgck/2^2	–	
		001	fcgck/2^3	fs/2^2	
		010	fcgck/2^4	fs/2^3	
		011	fcgck/2^5	fs/2^4	
		100	fcgck/2^6	fs/2^5	
		101	fcgck/2^7	fs/2^6	
		110	fcgck/2^8	fs/2^7	
		111	External clock (event counter mode)		

∗ : Don't care

Note: Altering source clock and edge type under event counter mode is allowed only if the extended timer has been stopped. They can be set concurrently with enabling (start instruction).

Pulse Measure Enable Register

Mode register for capturing in order to calculate time difference or effective counting

	7	6	5	4	3	2	1	0
PME (0016_H)								PME

(Initial value: ∗∗∗∗ ∗∗∗0)

PME	edge which causes interrupt request	0: of earliest detection 1: of second earliest detection	Write only

Note 1: If the address, PME locates, is read, the lower bits on ET0ICB is extracted.

Note 2: As writing on PME by software, only the data "1" is accepted. PME is cleared to "0" by hardware when the edge is detected or the device is totally reset.

Note 3: PME is a write only register and must not be used with any of the read-modify-write instructions.

Figure 2-32. Registers on Extended Timer-Counter 0 (1/2)

ETC0 Capture/Compare Mode Register

The register for both capturing and comparing

	7	6	5	4	3	2	1	0	
ET0MIO (0011$_H$)	OCIDEN	ICEN	ICMODE		OCID	OCEN	OCMODE		(Initial value: 1000 1000)

OCIDEN	Forcing OCID to initialize	0: Disable 1: Enable	R/W
ICEN	Assigning capture to P36	0: Disable (used for port) 1: Enable (used for capturing)	
ICMODE	Edge detection during capturing	00: No detection 01: Detect rising edge 10: Detect falling edge 11: Detect both edges	
OCID	Initial value for comparator	0: Initialize to “0” 1: Initialize to “1” * Valid when OCIDEN = “1”	
OCEN	Assigning compare output to P37	0: Disable (used for port) 1: Enable (used for comparator output)	
OCMODE	Comparator output when matched	00: NOP, output unchanged (steady output) 01: “1” output 10: “0” output 11: TOGGLE output (invert output)	

ETC0 Capture Register A/B

These 16-bit registers record the time when the capture terminal input changed. At maximum, 2 latest data are recorded; after detection, the capture register A holds the latest.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ET0ICA (0014, 15$_H$)	ET0ICA$_H$ (0015$_H$)								ET0ICA$_L$ (0014$_H$)							

Read only (Initial value: 0000 0000 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ET0ICB (0016, 17$_H$)	ET0ICB$_H$ (0017$_H$)								ET0ICB$_L$ (0016$_H$)							

Read only (Initial value: 0000 0000 0000 0000)

Note: *Since capturing is restored by reading the upper byte of the capture register A, read capture register B before reading capture register A. Read capture register A from lower byte to upper byte.*

ETC0 Compare Register

The 16-bit register is to set the time for changing output on comparator.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ET0OCR (0018, 19$_H$)	ET0OCR$_H$ (0019$_H$)								ET0OCR$_L$ (0018$_H$)							

Read/Write (Initial value: 0000 0000 0000 0000)

Note 1: *Data written on address (0018$_H$), does not reach lower byte of the ETC0 compare register until data on upper byte of it is set because of buffering. Writing only to lower bite is not allowed.*

Note 2: *The lower byte of the ETC0 register and the lower byte of the ETC0 compare register share their buffering register. Therefore, do not execute write instruction against the ETC0 register, between writing on lower byte of the ETC0 compare register and upper byte of it.*

Figure 2-33. Registers on Extended Timer-Counter 0 (2/2)

2.7.3 Source Clock

The source clock for the extended timer-counter (ETC0) has 2 sorts of operating mode: the free-running mode whose source clock is provided from internal clock, and event-counter mode whose source clock is provided from external clock (through terminal ETC0).

Source Clock and Accuracy for ETC0 (at fcgck = 8 MHz, fs = 32.768 kHz)

Operating Mode	Source Clock	Edge	Accuracy	
			(fcgck)	(fs)
Free Running Timer	Internal	Rising	500 ns to 32 μs	122 μs to 3.91 ms
Event Counter	ETCO terminal	Rising or falling	1 μs or more	244 μs or more

(1) Free Running Timer Mode

The internal clock increase the extended timer-counter (ETC0). The interrupt INTET0 is requested when the counter reaches $\mathrm{FFFF_H}$, meanwhile the counter is still counting up. The source clock is selected on the bit ET0CKS in ETC0 control register (ET0CR). The counter becomes event-counter mode if ET0CKS = "111". In order to generate interrupt after a certain period, subtract a rate from $\mathrm{FFFF_H}$ and set its result on the ETC0 register (ET0R). The ET0R is cleared if $\mathrm{0000_H}$ is written on ETC0R. Both writing and reading against the ET0R are available, while the counter is in motion or halt. By executing read instruction against the ET0R, the current rate on the counter is extracted. Notice that reading and writing against the ET0R should be done from lower byte to upper byte consecutively.

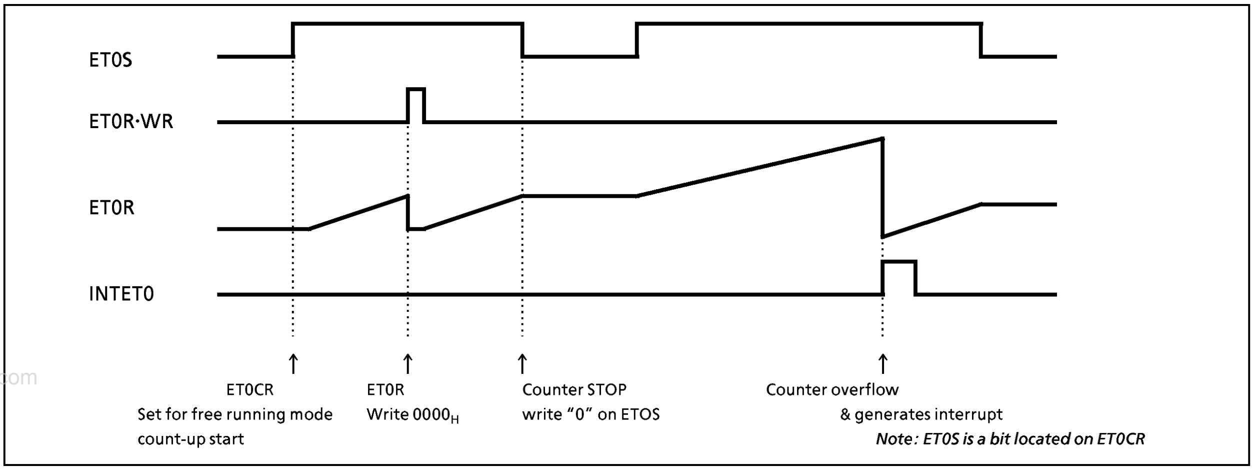
Since the counter extract the whole data for the ET0R as data is written on upper byte, the exclusive writing against lower byte of ET0R is not valid. As for reading, data on the counter is latched on reading against lower byte.

Example 1: Generate interrupt of ETC0 in 8 s, from source clock of fs/4 (fs = 32.768 kHz)

```
LD      (ET0CR), 00000001B     ;  Stop the counter, and assign fs/2^2 to source clock
LD      (ET0R), 0000H          ;  Set timer register (8 s ÷ 122 μs = FFFFH)
DI
SET     (EIRH).2               ;  Enable interrupt INTET0
EI
SET     (ET0CR).7              ;  ETC0 start
```

Example 2: Generate interrupt of ETC0 in 256 μs, from source clock of fcgck/8 (fcgck = 8 MHz, DV1CK = 0)

```
LD      (ET0CR), 00000001B     ;  Stop the counter, and assign fcgck/8 to source clock
LDW     (ET0R), 0FEFFH         ;  Set timer register (FFFFH − 100H)
DI
SET     (EIRH) .2              ;  Enable interrupt INTET0
EI
SET     (ET0CR).7              ;  ETC0 start
```



Note: ET0S is a bit located on ET0CR

Figure 2-34. Example for Free-running mode operation

Table 2-13. ETC0 Internal clock (fc = 8 MHz, FCGCK = 000, fs = 32.768 kHz)

ET0CKS	NORMAL 1/2, IDLE 1/2 mode		SLOW, SLEEP mode	
	Accuracy	Range	Accuracy	Range
000	500 ns	32.77 ms	–	–
001	1 μs	65.54 ms	122.0 μs	8 s
010	2 μs	131.07 ms	244.1 μs	16 s
011	4 μs	262.14 ms	488.3 μs	32 s
100	8 μs	524.29 ms	976.6 μs	64 s
101	16 μs	1.05 s	1.953 ms	128 s
110	32 μs	2.10 s	3.906 ms	256 s

Note: ET0CKS is a group of bits, which manages the source clock of ETC0.

(2) Event-counter Mode

The edge of ETC0 terminal input increase the extended timer-counter (ETC0). The counter becomes event-counter mode if ET0CKS = "111".
In order to utilize event-counter mode, set ETC0 terminal to input mode on P1CR. The type of capturing edge, either rising or falling, can be selected on ET0ES located on bit 3 on ET0CR. The counter operates similar to free running counter mode, except for the source clock: internal clock or terminal input.

Example: Generate interrupt of ETC0 after counting 100 (64_H) times of rising edge

```
CLR     (ET0CR). 7      ;  stops the counter
CLR     (P1CR). 2       ;  sets P12 (ETC0) for input
LD      (ET0CR), 07H    ;  event counter mode, detecting rising edge
LDW     (ET0R), 0FF9BH  ;  sets timer register (FFFFH to 0064H)
DI
SET     (EIRH). 2       ;  enables interrupt INTET0
EI                      ;
SET     (ET0CR). 7      ; starts the counter
```

Note: Altering source clock and edge type under event counter mode is allowed only if the extended timer has been stopped.

Since ETC0 input terminal has digital noise cancellor, only pulse width of 2 machine cycles or longer is accepted. Pulse width of 1 machine cycle or shorter is neglected. It is unstable to acknowledge pulse width of between 1 and 2, as valid input.

(input pulse width) < (1 machine cycle) → not counted
(1 machine cycle) ≦ (input pulse width) < (2 machine cycles) → counted or not counted (depend on timing)
(2 machine cycle) ≦ (input pulse width) → counted
4/fcgck = 1 machine cycle (0.5 μs at fcgck = 8 MHz)

After detecting a rising (falling) edge, it is required to detect falling (rising) edge before detecting another rising (falling) edge.

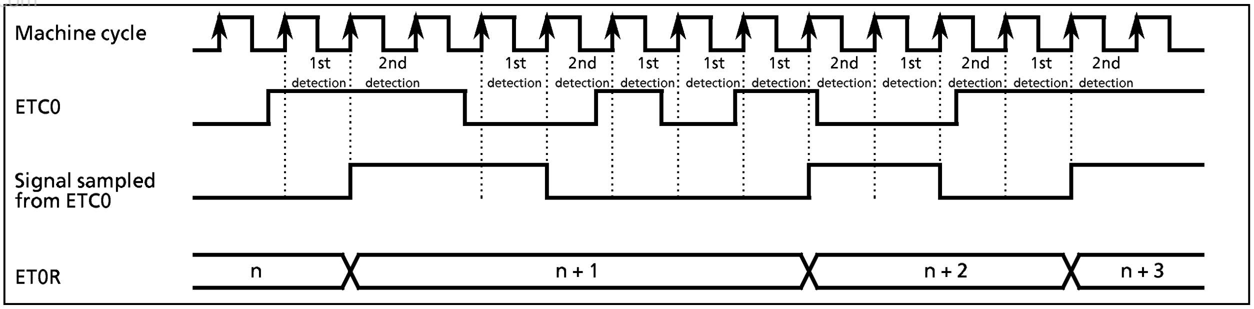
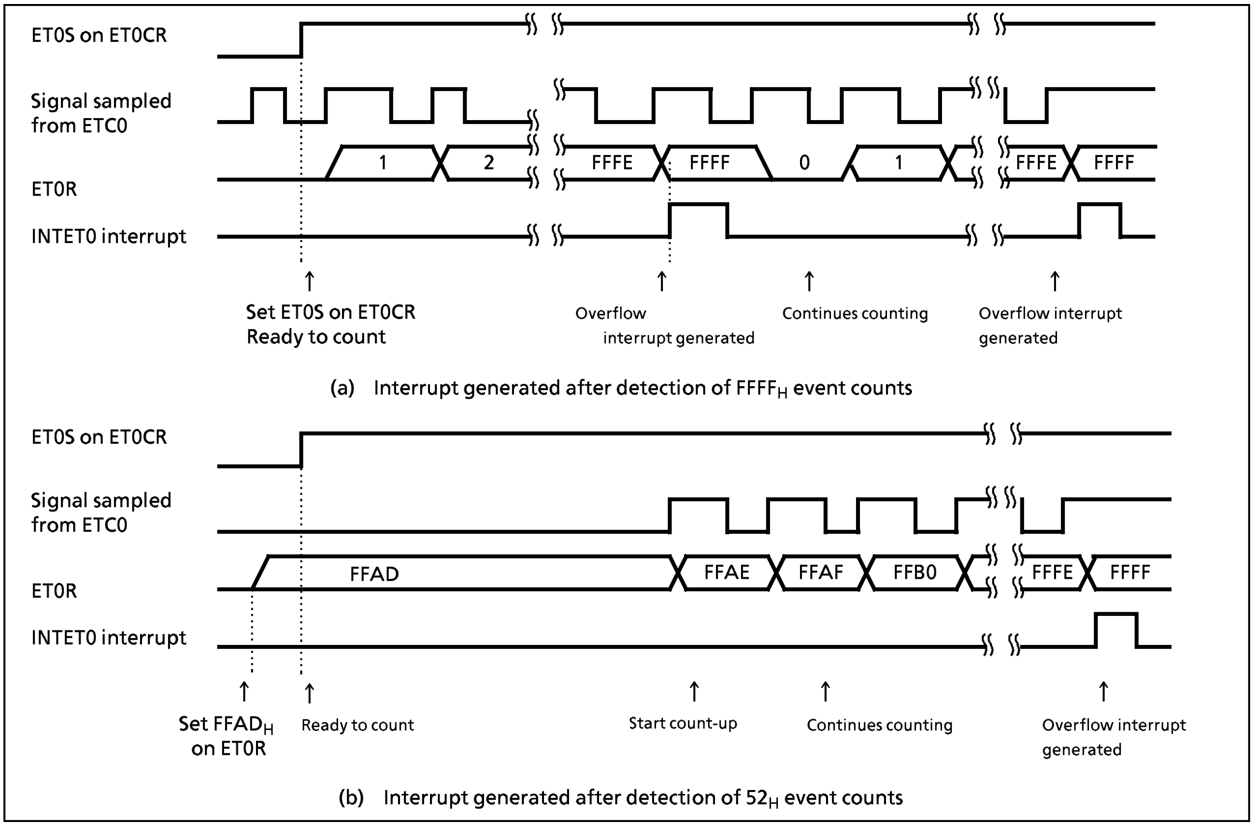


Figure 2-35. Edge Filtering in Event Counter Mode (Rising Edge Detection)



(a) Interrupt generated after detection of FFFF$_H$ event counts

(b) Interrupt generated after detection of 52$_H$ event counts

Figure 2-36. Example of Event Counter Mode Operation (ETC0 Rising Edge Detection)

2.7.4 Capturing input, Output comparing

The extended timer-counter (ETC0) involves terminals for capturing input IC0 (P36) and output comparing OC0 (P37). Both IC0 and OC0 are available for general input/output ports if they are not involved in the extended timer-counter (ETC0).

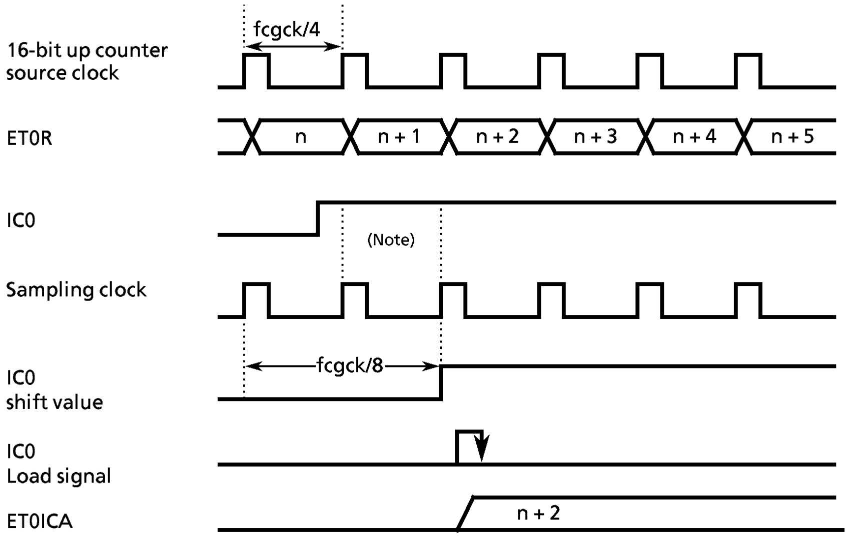
Capturing input and Output comparing Functions

	Shared function	Operation	Accuracy	Data register
Capture input	P36	Rising edge Falling edge Both edges	Depend on timer	2
Output compare	P37	"1" output "0" output toggle NOP		1

When capturing input function is to be used, assign input mode to P36 (set "0" on bit6 on P3CR). When output comparing function is to be used, assign output mode to P37 (set "1" on bit7 on P3CR) after P37 output latch (bit7 on P3DR) is set to "1".

(1) Capturing input

It is the function to measure matters, such as pulse width, frequency or duty.
At the time the capturing acknowledged input changed, the contemporary rate on the extended timer-counter (ETC0) is loaded on the capture register. The capturing contains a digital noise-cancellor circuit in its block. A stable pulse of fcgck/8 or longer is required in order to inform the hardware of input, otherwise the pulse would not be acknowledged normally. Since the rate on the extended timer-counter (ETC0) is loaded on the capture register after the capturing samples every fcgck/4 with sampling clock, there is a lag of fcgck/4 to fcgck/8 between terminal input and capture register record.
After detecting edge, the following capture operation is prohibited, until the data on the capture register ET0ICA. As for reading ET0ICA, read lower byte first then read upper byte.
The capturing operates normally regardless of overflow of the counter, since the counter is still counting-up after overflow.



Note: Sampling clock for IC0 input is fcgck/4, regardless of its source clock.

Figure 2-37. Example of Event Counter Mode Operation (IC0 rising edge detection, Source clock of fcgck/4)

Example 1: Capturing IC0 on rising edge with ETC0

```
        CLR     (P3CR).6                ;  Assigns input mode to P36
        LD      (ET0MIO),0101****B      ;  Enables capturing on rising edge
```

Example 2: Capturing positive pulse width IC0, with ETC0 (without using ET0ICB)

```
        CLR     (P3CR). 6               ;  Assigns input mode to P36
        DI
        SET     (EIRH). 3               ;  Enables individual interrupt enable flag for
                                           INTIC0
        EI
        LD      (ET0MIO), 0111****B     ;  Enables capturing on both edges
LP:     TEST    (ILH).3                 ;  Waits until IC0 rises (IL11 rises)
        JR      T, LP
        DI
        LD      (ILH), F7H              ;  Clears IL11
        EI
        LD      A, (ET0ICAL)            ;  Reads ET0ICA
        LD      W, (ET0ICAH)
LP2:    TEST    (ILH).3                 ;  Waits until IC0 falls (IL11 rises)
        JR      T, LP2
        SUB     WA, (ET0ICA)            ;  Calculates pulse width and loads it on HL register
        LD      HL, 0000H
        SUB     HL, WA
        DI
        LD      (ILH), F7H              ;  Clears IL11
        EI
```

If the data “1” is set on register PME located on the address same as ET0ICBL, the interrupt INTIC0 is requested as the second valid edge is detected. As this interrupt is requested, the first detected time (loaded on ET0ICB) and the second (loaded on ET0ICA) can be read consecutively. PME is cleared to “0” by hardware, after the edge is detected. Therefore, after the interrupt INTIC0 is requested, it is required to set “1” on PME again in order to continue extracting the 2-word data on capture register. As the second detected time (loaded on ET0ICA) is read, the capture operation is enabled again.
As for extracting the 2-word data on capture register, read the first detected time (loaded on ET0ICB) before reading the second (loaded on ET0ICA). In order to extract the 2-word data normally, the data “1” should be set on PME before the first edge detection, if the capture operation has already been enabled.

Example 3: Calculate (Second risen time) - (First risen time) and then calculate (Fourth risen time) - (Third risen time), as for IC0 input

```
        CLR   (P3CR).6               ;  Assigns IC0 to P36
        SET   (PME).0                ;  Rises PME (interrupt after second edge detection)
        DI
        SET   (EIRH).3               ;  Enables individual interrupt enable flag for
                                        INTIC0
        LD    (ET0MIO),0101----B     ;  Enables capturing on rising edge
        EI
LP:     JR    LP
IC0:    SET   (ET0ICBL).0            ;  Raises PME (interrupt after second edge detection)
        LD    HL,(ET0ICBL)           ;  Reads the first (third) detected time
        LD    BC,(ET0ICAL)           ;  Reads the second (fourth) detected time,
                                     ;  Enables capturing again
        SUB   BC,HL                  ;  Calculates time
        RETI
```

(2) Output Comparing

The output level alters at the time stated. The output mode is selected on the bits OCMODE on register ET0MIO whether to be output "1", output "0", invert or NOP.

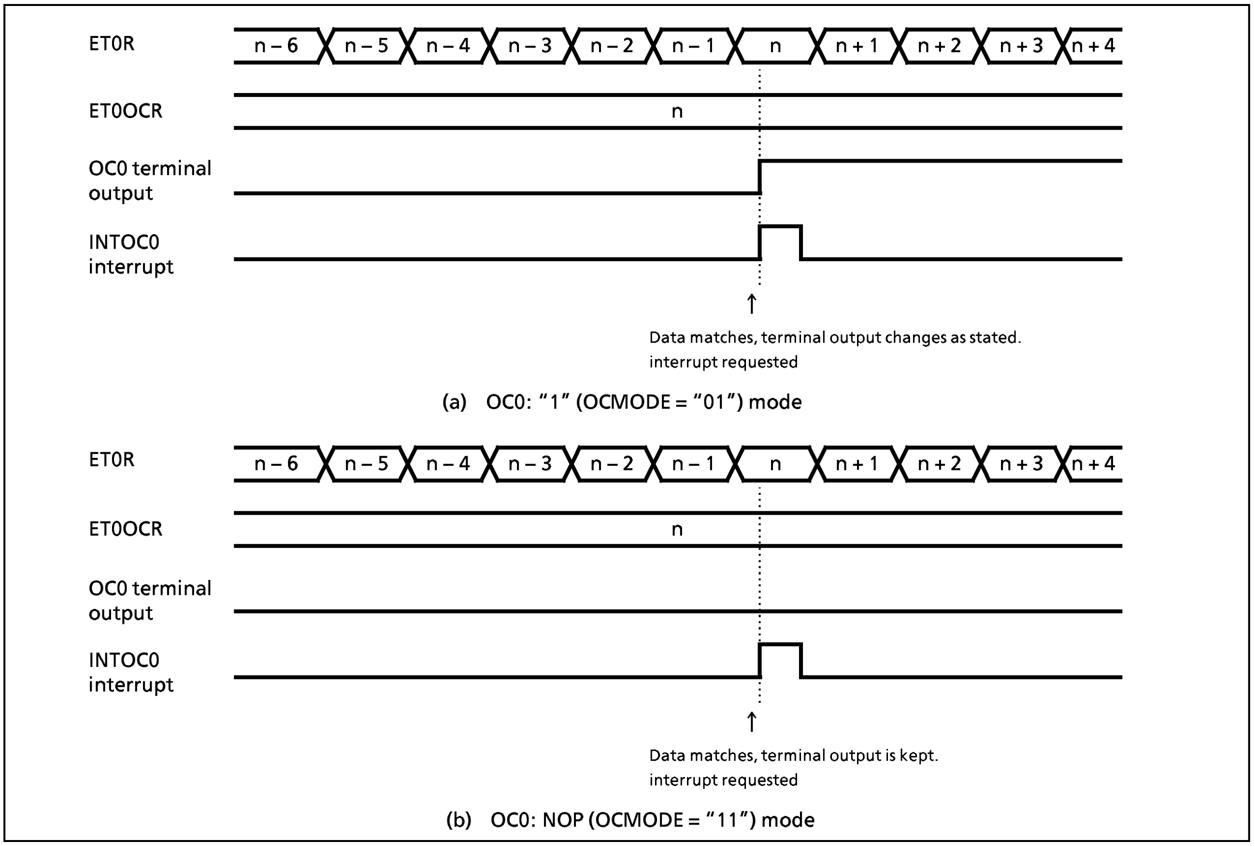
The initial value that the comparator outputs is set on bit OCID in register ET0MIO. Once the comparator is enabled, namely during comparison, bit OCIDEN should be cleared to "0" for fear that other than data matching alter OC0 output.

The time to alter output is set on the ETC0 compare register (ETC0OCR). This register should be consecutively written from the lower byte to the upper. As the upper byte is set, the 2-byte data is loaded on the ETC0OCR effectively. Writing only to the lower byte is prohibited.

As the rate of the timer-counter and comparator matches, the stated data depends on OCMODE is output on P37, and the interrupt INTOC0 is requested.

If OCMODE has been set for NOP, the terminal keeps its value and only the interrupt request is required. If OCMODE has been "11", the terminal inverts its output value.

Note: While the timer-counter is not in motion (ET0S = "0"), the comparator outputs "1" regardless of the rate on register ET0MIO.



(a) OC0: "1" (OCMODE = "01") mode

(b) OC0: NOP (OCMODE = "11") mode

Figure 2-38. Examples for comparing operation

Example 1: Change OC0 output from “0” to “1” when ET0R = 0C80$_H$. (“1” output mode)

```
SET  (P3DR).7
SET  (P3CR).7           ;  assigns OC0 output mode to P37
LDW  (ET0OCRL), 0C80H   ;  sets time for changing output
LD   (ET0MIO), 1---0101B ; initializes OC0 to “0” and assign “1” output mode to it
CLR  (ET0MIO).7         ;  prohibits OCID revising
```

Example 2: Change OC0 output from “1” to “0” when ET0R = 0C80$_H$. (toggle output mode)

```
SET  (P3DR).7
SET  (P3CR).7           ;  assigns OC0 output mode to P37
LDW  (ET0OCRL), 0C80H   ;  sets time for changing output
LD   (ET0MIO), 1---1111B ; initializes OC0 to “1” and assigns toggle output mode to it
CLR  (ET0MIO).7         ;  prohibits OCID revising
```

Example 3: Output the pulse with “0” for 1ms and “1” for 2ms, through OC0 terminal. (at fcgck = 8[MHz])

```
        LD   (ET0CR), 01H       ;  selects fcgck/8 (1μs) for source clock
        LDW  (ET0R), 0000H      ;  clears the counter
        LD   BC, 03E8H          ;  substitutes “0” width (1 ms) for BC
        LD   DE, 07D0H          ;  substitutes “1” width (2 ms) for DE
        LD   WA, 03E8H          ;  substitutes “initial value” width (1 ms) for WA
        SET  (P3DR).7
        SET  (P3CR).7           ;  assigns OC0 output mode to P37
        LD   (ET0OCRL), A
        LD   (ET0OCRH), W       ;  sets time for changing output
        LD   (ET0MIO), 1---0111B ; initializes OC0 to “0” and assigns toggle output mode to it
        CLR  (ET0MIO).7         ;  prohibits OCID revising
        SET  (ET0CR).7          ;  ETC0 starts counting
OUT0:
        ADD  WA, DE             ;  calculates time to change output from “1” to “0”
LP:
        TEST (ILH).4            ;  waits for changing output from “0” to “1” (sets IL12)
        JR   T, LP
        DI
        LD   (ILH), 0EFH        ;  clears IL12
        EI
        LD   (ET0OCRL), A
        LD   (ET0OCRH), W       ;  sets time for changing output from “1” to “0”
OUT1:
        ADD  WA,BC              ;  calculates time to change output from “0” to “1”
LP1:
        TEST (ILH).4            ;  waits for changing output from “1” to “0” (sets IL12)
        JR   T, LP1
        DI
        LD   (ILH), 0EFH        ;  clears IL12
        EI
        LD   (ET0OCRL), A
        LD   (ET0OCRH), W       ;  sets time for changing output from “0” to “1”
        JR   OUT0
```

Example 4: Generate only interrupt request when ET0R = 0C80$_H$: without OC0 terminal output.

```
LDW  (ET0OCRL), 0C80H   ;  sets time for interrupt request
LD   (ET0MIO),1---1100B ;  initializes OC0 to “1” and assigns NOP output mode to it
CLR  (ET0MIO).7         ;  prohibits OCID revising
```

2.7.5 Interrupting

There are 3 sorts of interrupt requesting: for extended timer (INTET0), for capture input (INTIC0) and for output compare (INTOC0).

(1) INTET0

The interrupt requesting is generated as the counter reaches $FFFF_H$; meanwhile the counter is still counting up.

(2) INTIC0

Interrupt requesting is triggered by capture input pin (IC0). The interrupt is requested when the timer value at edge detection is loaded on the capture register.

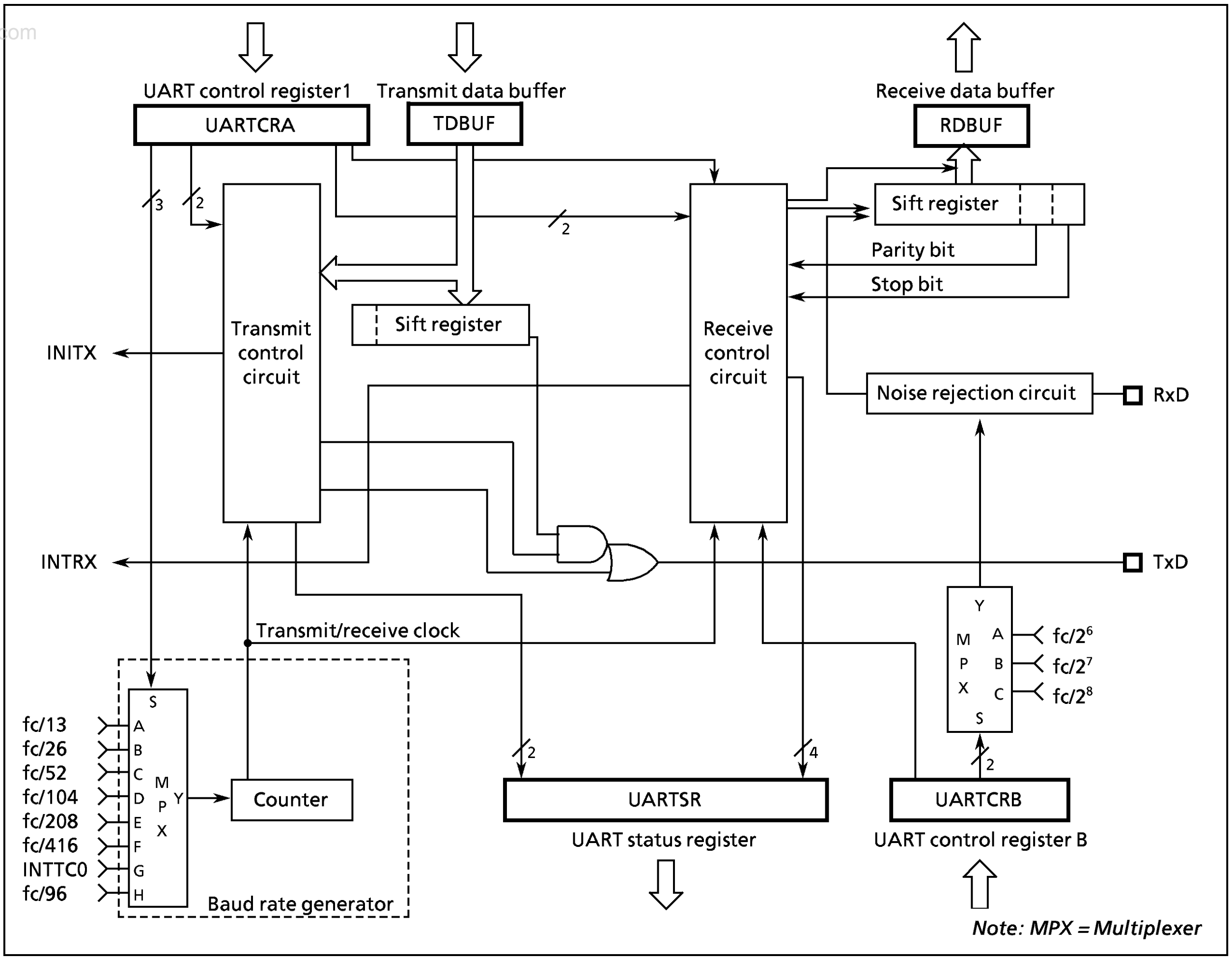
(3) INTOC0

Interrupt requesting is triggered by comparator. The interrupt is requested when the rate of the timer-counter and comparator matches.

2.8 UART (Asynchronous serial interface)

The TMP86CH06 has 2 channels of UART (asynchronous serial interface).
The UART is connected to external devices via RxD and TxD. RxD0 and RxD1 are also used as P43 and P46, respectively ; TxD0 and TxD1 as P44 and P47 respectively. To use P44 or P47 as the TxD pin, set corresponding bit in P4 port output latches to 1 and set the terminal to output mode.
To use P43 or P46 as the RxD pin, set the terminal to input mode respectively.
Notice that P43 (RxD0) and P44 (TxD0) are also used as syncronized serial interface (SIO) ; the ch.0 of UART is not available when the syncronized serial interface is utilized.

2.8.1 Configuration



Note: MPX = Multiplexer

Figure 2-39. UART

2.8.2 Control

UART is controlled by the UART control registers (UARTCRA, UARTCRB). The operating status can be monitored using the UART status register (UART0SR, UART1SR).

UART Control Register

	7	6	5	4	3	2	1	0	
UART0CRA ($001A_H$)	TXE	RXE	STBT	EVEN	PE	BRG			(Initial value: 0000 0000)

UART1CRA ($001E_H$)

BRG	Transmit clock select	000: fc/13 [Hz] 001: fc/26 010: fc/52 011: fc/104 100: fc/208 101: fc/416 110: INTTC0 (8-bit timer counter TC0 interrupt) 111: fc/96	Write only
PE	Parity addition	0: No parity 1: Parity	
EVEN	Even-numbered parity	0: Odd-numbered parity 1: Even-numbered parity	
STBT	Transmit stop bit length	0: 1 bit 1: 2 bit	
RXE	Receive operation	0: Disable 1: Enable	
TXE	Transfer operation	0: Disable 1: Enable	

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

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

Note 3: UART0CRA and UART1CRA are a write only register and must not be used with any of the read-modify-write instructions.

Note 4: Do not switch the transmit clock during transmit operation.

	7	6	5	4	3	2	1	0	
UART0CRB ($001B_H$)						RXDNC		STOPBR	(Initial value: **** *000)

UART1CRB ($001F_H$)

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

Note 5: When RxDNC = 01, pulses longer than 96/fc [s] are always regarded as signals ; when RxDNC = 10, longer than 192/fc [s] ; and when RxDNC = 11, longer than 384/fc [s]

Note 6: The following types of clocks are available for transmitting, while the clock gear is utilized.

Note 7: UART0CRB and UART1CRB are a write only register and must not be used with any of the read-modify-write instructions.

BRG	Clock Rate	fcgck [Hz]					
		fc	fc/2	fc/4	fc/8	fc/16	fc/32
000	fc/13	Enable	Disable	Disable	Disable	Disable	Disable
001	fc/26	Enable	Enable	Disable	Disable	Disable	Disable
010	fc/52	Enable	Enable	Enable	Disable	Disable	Disable
011	fc/104	Enable	Enable	Enable	Enable	Disable	Disable
100	fc/208	Enable	Enable	Enable	Enable	Enable	Disable
101	fc/416	Enable	Enable	Enable	Enable	Enable	Enable
111	fc/96	Enable	Enable	Enable	Enable	Disable	Disable

Figure 2-40. UART control register

UART Status Register

UART0SR ($001A_H$)
UART1SR ($001E_H$)

7	6	5	4	3	2	1	0
PERR	FERR	OERR	RBFL	TEND	TBEP		

(Initial value: 0000 11**)

TBEP	Transmit data buffer empty flag	0: – 1: Transmit data buffer empty	Read only
TEND	Transmit end flag	0: Transmitting 1: Transmit end	
RBFL	Receive data buffer full flag	0: – 1: Receive data buffer full	
OERR	Overrun error flag	0: No overrun error 1: Overrun error	
FERR	Framing error flag	0: No framing error 1: Framing error	
PERR	Parity error flag	0: No parity error 1: Parity error	

UART Receive Data Buffer

RD0BUF ($001C_H$)
RD1BUF ($001D_H$)

7	6	5	4	3	2	1	0

(Initial value: - - - - - - - -) Read-only

UART Transmit Data Buffer

TD0BUF ($001C_H$)
TD1BUF ($001D_H$)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000) Write-only

Note: TD0BUF and TD1BUF are a write only register and must not be used with any of the read-modify-write instructions.

Figure 2-41. UART status register and Data Buffer Registers

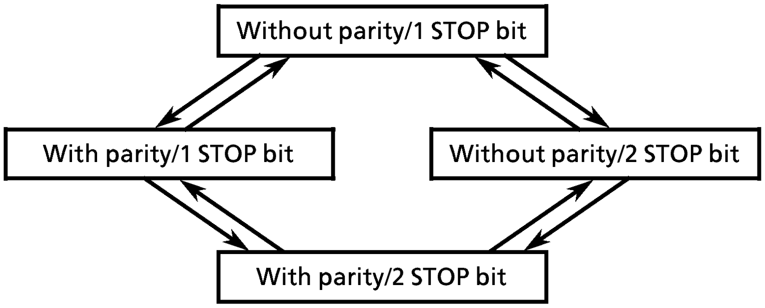
2.8.3 Transfer Data Format

In UART, a one-bit start bit (low level) , stop bit (bit length selectable at high level, by STBT), and parity (select parity in PE ; even-or odd-numbered parity by EVEN) are added to the transfer data. The transfer data formats are shown as follow.

Table 2-14. Transfer Data Format

PE	STBT	Frame length								
		1	2	3	-------	8	9	10	11	12
0	0	Start	Bit 0	Bit 1	-----	Bit 6	Bit 7	Stop1		
0	1	Start	Bit 0	Bit 1	-----	Bit 6	Bit 7	Stop1	Stop2	
1	0	Start	Bit 0	Bit 1	-----	Bit 6	Bit 7	Parity	Stop1	
1	1	Start	Bit 0	Bit 1	-----	Bit 6	Bit 7	Parity	Stop1	Stop2

Note: To switch the transfer data format at times other than initial setting, follow the transition sequence shown below as a deadlock may be caused in data transfer.



2.8.4 Transfer Rate

The baud rate of UART is set of BRG (bit 0, 1, and 2 in UARTCR1) . The example of the baud rate shown as follows.

Table 2-15. Transfer Rate (Example)

BRG	Sourse clock		
	16 MHz	8 MHz	4 MHz
000	76800 [baud]	38400 [baud]	19200 [baud]
001	38400	19200	9600
010	19200	9600	4800
011	9600	4800	2400
100	4800	2400	1200
101	2400	1200	600

When INTTC0 is used as the UART transfer rate (when BRG = 110), the transfer clock and transfer rate are detarmined as follows:

$$\text{Transfer clock} = \frac{\text{TC0 source clock}}{\text{TTREG0 set value}}$$

$$\text{Transfer rate} = \frac{\text{Transfer clock}}{16}$$

2.8.5 Data Sampling

The UART receiver keeps sampling input using the clock selected by BRG (bit 0, 1, and 2 in UARTCRA) until a start bit is detected in RxD pin input. RT clock starts at the falling edge of the RxD pin. Once a start bit is detected, the start bit, data bits, stop bit(s), and parity bit are sampled at three times of RT7, RT8, and RT9 during one receiver clock interval (RT clock). (RT0 is the position where the bit supposedly starts) . Bit is determined according to majority rule (the data are the same twice or more out of three samplings) .

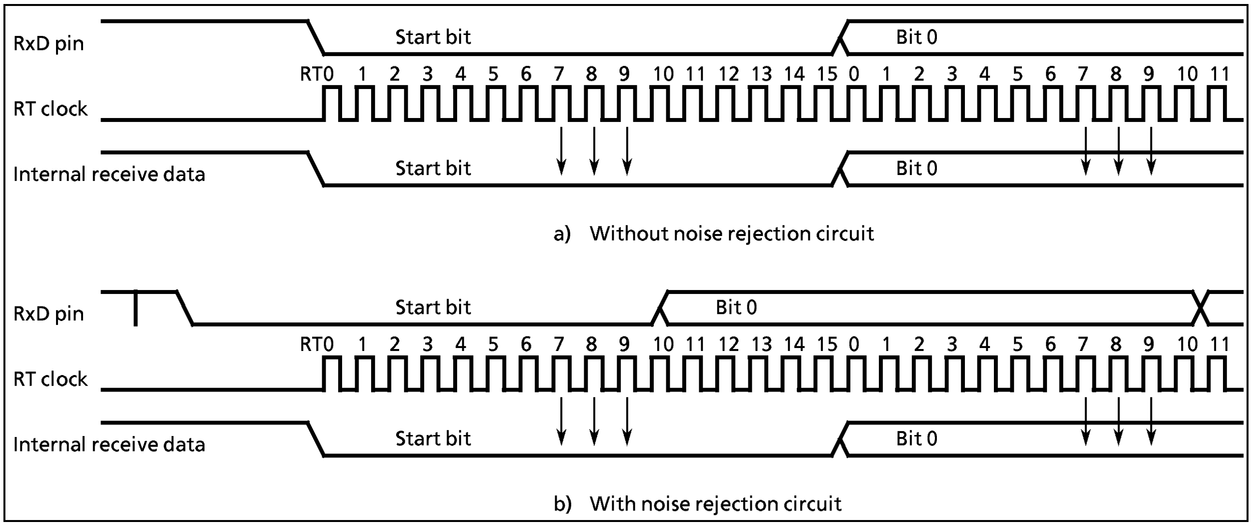

RxD pin
Start bit
Bit 0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
Bit 0
a) Without noise rejection circuit
RxD pin
Start bit
Bit 0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
Bit 0
b) With noise rejection circuit

Figure 2-42. Data Sampling

2.8.6 STOP Bit Length

Select a transmit stop bit length (1 or 2 bits) by STBT (bit5 in UARTCR1)

2.8.7 Parity

Set parity/no parity by PE ; set parity type (odd-or even-numbered) by EVEN (bit 4 in UARTCRA).

2.8.8 Transmit/Receive

(1) Data transmit

Set TXE (bit 7 in UARTCRA) to 1. Read UARTSR to check TBEP=1, then write data in TDBUF (transmit data buffer). Writing data in TDBUF zero-clears TBEP, transfers the data to the transmit shift register and the data are sequentially output from the TxD pin. The data output include a one-bit start bit, stop bits whose number is specified in STBT (bit 5 in UARTCRA) and a parity bit if parity addition is specified. Select the data transfer baud rate using bits 0 to 2 in UARTCRA. When data transmit starts, transmit buffer empty flag TBEP is set to 1 and an INTTX interrupt is generated.
When transmitting data, first read UARTSR, then write data in TDBUF. Otherwise, TBEP is not zero-cleared and transmit does not start.

(2) Data receive

Set RXE (bit 6 in UARTCRA) to 1. When data are received via the RxD pin, the receive data are transferred to RDBUF (receive data buffer). At this time, the data transmitted include a start bit and stop bit(s) and a parity bit if parity addition is specified. When stop bit(s) are received, data only are extracted and transferred to RDBUF (receive data buffer). Then the receive buffer full flag RBFL is set and an INTRX interrupt is generated. Select the data transfer baud rate using bits 0 to 2 in UARTCRA. If an overrun error (OERR) occurs when data are received, the data are not transferred to RDBUF (receive data buffer) but discarded; data in the RDBUF are not affected.

Note: Setting RXE bit to "0" causes data receive to stop after the data transfer currently being received has completed. However, if a flaming error occurs in this last transfer, the data receive stop state may not be entered. Therefore, if a flaming error occurs, be sure to perform the receive operation again.

2.8.9 Status Flag/Interrupt Signal

(1) Parity error

When parity determined using the receive data bits differs from the received parity bit, the parity error flag PERR is set in UARTSR. Reading UARTSR then RDBUF clears PERR.

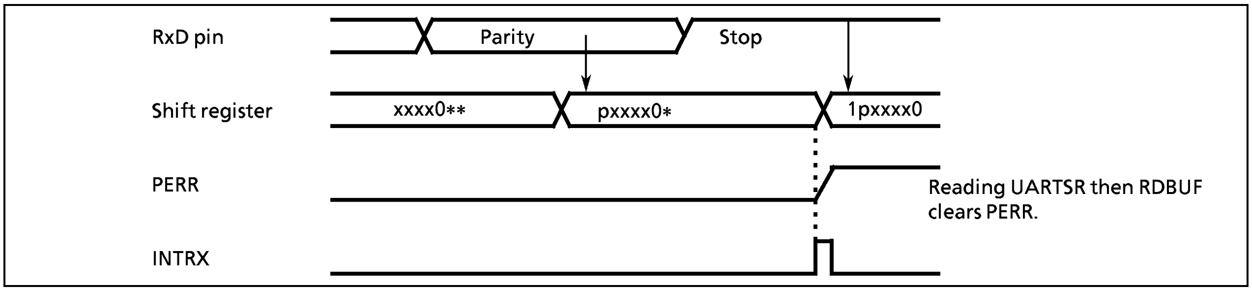


Figure 2-43. Generation of Parity Error

(2) Framing error

When 0 is sampled as the stop bit in the receive data, framing error flag FERR is set. Reading UARTSR then RDBUF clears FERR.

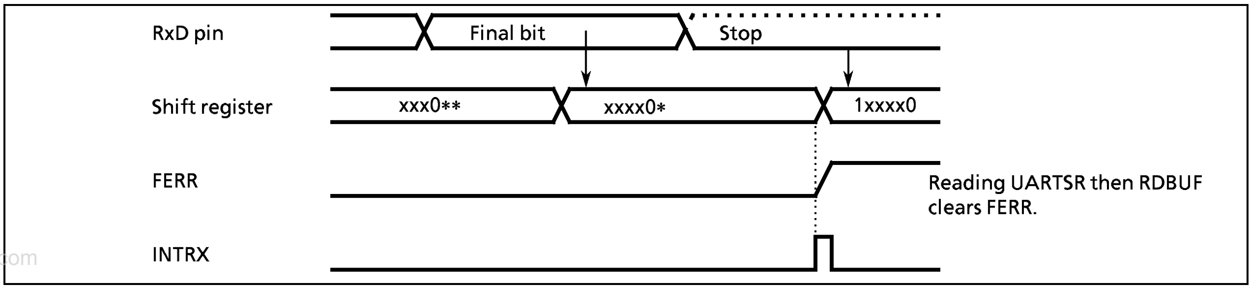


Figure 2-44. Generation of Framing Error

(3) Overrun error

When all bits in the next data are received while unread data are still in RDBUF, overrun error flag OERR is set. In this case, the receive data is discarded ; data in RDBUF are not affected. Reading UARTSR then RDBUF clears OERR.

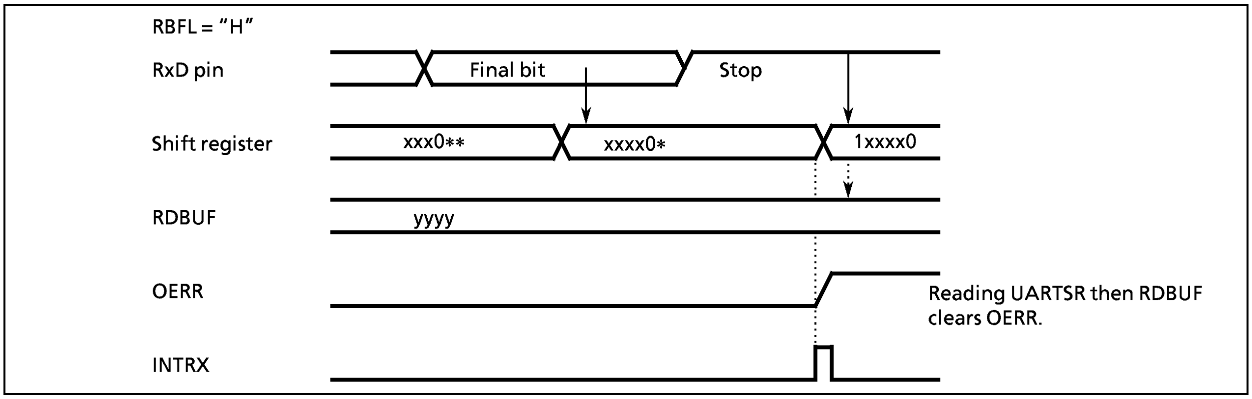


Figure 2-45. Generation of Overrun Error

(4) Receive data buffer full

Loading the received data in RDBUF sets receive data buffer full flag RBFL. Reading UARTSR then RDBUF clears the RBFL.

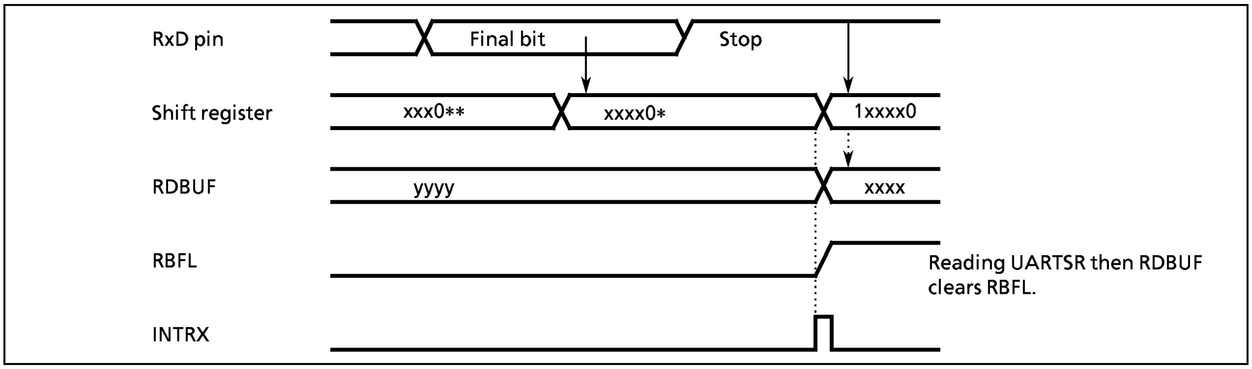


Figure 2-46. Generation of Receive Buffer Full

(5) Transmit data buffer empty

When no data is in the transmit buffer TDBUF, TBEP is set, that is, when data in TDBUF are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag TBEP is set. Reading UARTSR then writing the data to TDBUF clears TBEP.

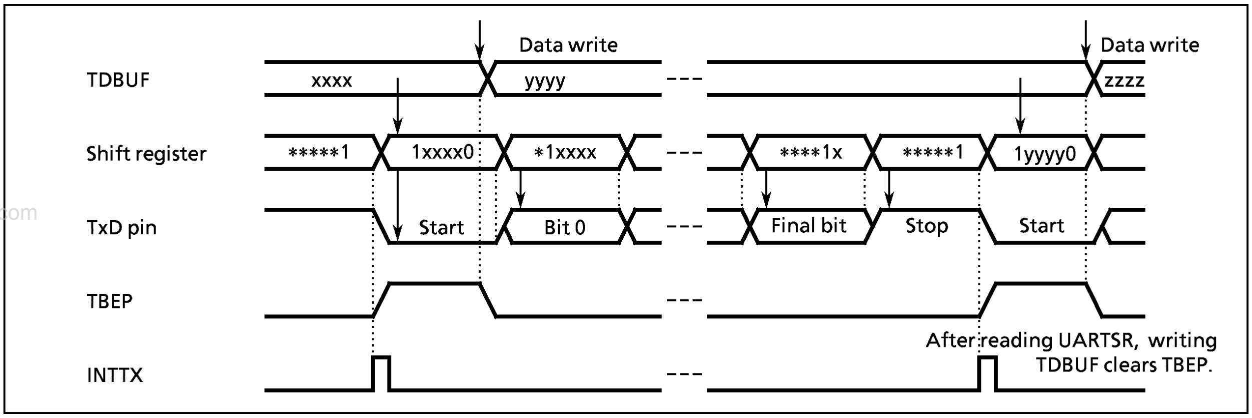


Figure 2-47. Generation of Transmit Buffer Empty

(6) Transmit end flag

When data are transmitted and no data is in TDBUF (TBEP = 1), transmit end flag TEND is set. Writing data to TDBUF then staring data transmit clears TEND.

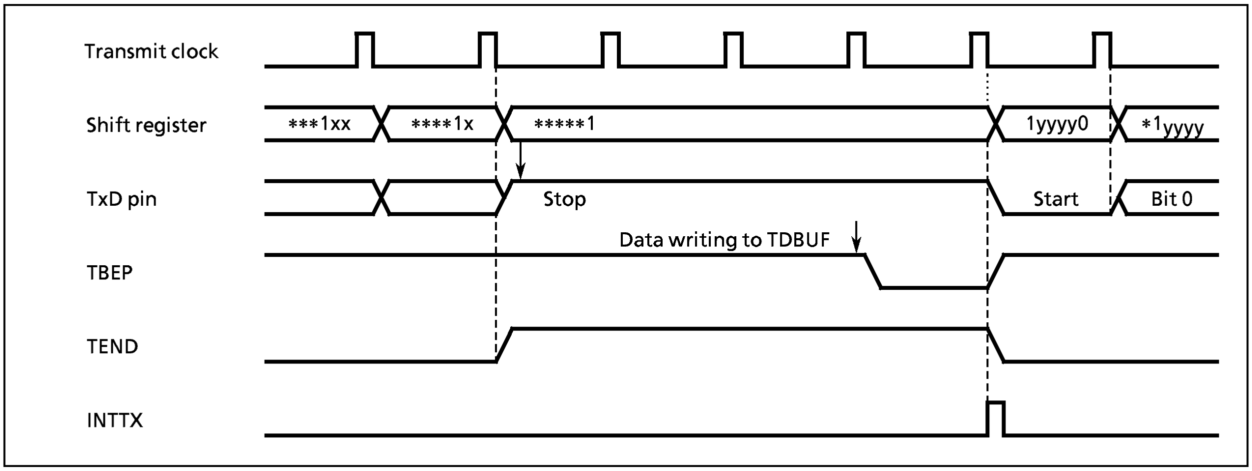


Figure 2-48. Generation of Transmit Buffer Empty

2.9 Serial Interface (SIO)

The TMP86CH06 has one clocked-synchronous 8-bit serial interface. Serial interface have an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data. The serial interfaces are connected to external devices via pins P44 (SO), P43 (SI), P42 ($\overline{\text{SCK}}$). The serial interface pins are also used as port P4. When used as input pins, the input pins should be set to the input mode. When used as output pins, beforehand the output pins should be set to the output mode, and output latch should be set to "1". In the transmit mode, pin P43 can be used as normal I/O port, and in the receive mode, the pin P44 can be used as normal I/O ports.

2.9.1 Configuration

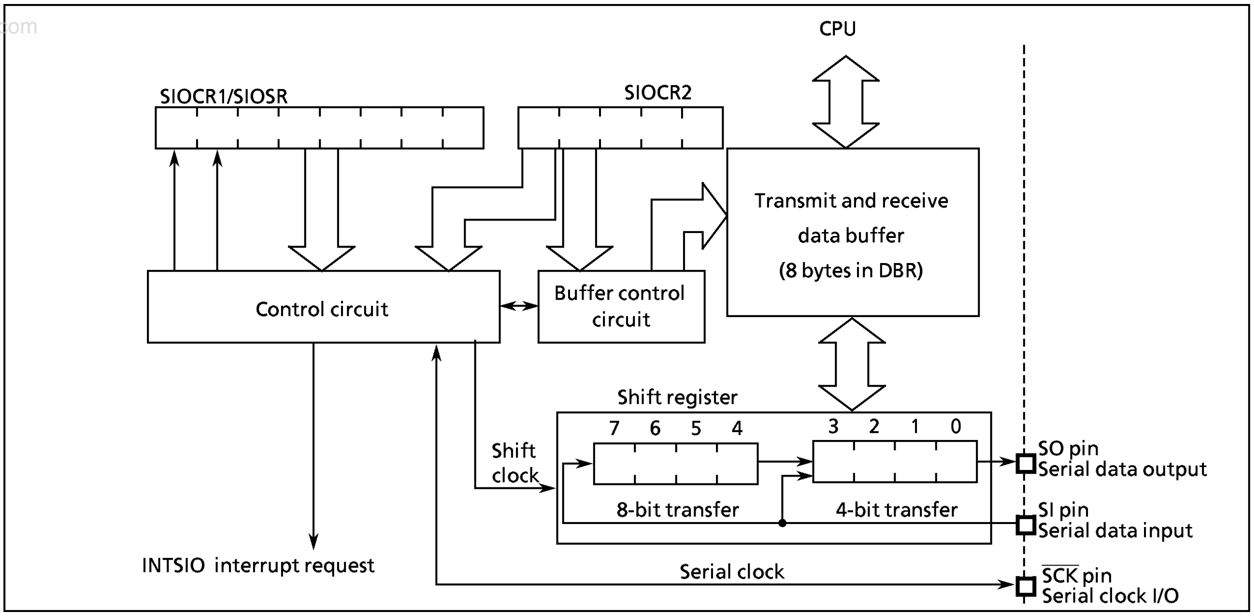


Figure 2-49. Serial Interfaces

2.9.2 Control

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

The transmit and receive data buffer is controlled by the BUF (bits 2 to 0 in SIOCR2). The data buffer is assigned to address 0028_H to $002F_H$ for SIO in the DBR area, and can continuously transfer up to 8 words (bytes or nibbles) at one time. When the specified number of words has been transferred, a buffer empty (in the transmit mode) or a buffer full (in the receive mode or transmit/receive mode) interrupt (INTSIO) is generated.

When the internal clock is used as the serial clock in the 8-bit receive mode and the 8-bit transmit/receive mode, a fixed interval wait can be applied to the serial clock for each word transferred. Four different wait times can be selected with WAIT (bits 4 and 3 in SIOCR2).

SIO Control Register 1

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

<table>
<tr><td>SIOS</td><td>Indicate transfer start/stop</td><td>0: Stop
1: Start</td><td>Write only</td></tr>
<tr><td>SIOINH</td><td>Continue/abort transfer</td><td>0: Continue transfer
1: Abort transfer (automatically cleared after abort)</td><td></td></tr>
<tr><td>SIOM</td><td>Transfer mode select</td><td>000: 8-bit transmit mode
010: 4-bit transmit mode
100: 8-bit transmit/receive mode
101: 8-bit receive mode
110: 4-bit receive mode</td><td></td></tr>
<tr><td>SCK</td><td>Serial clock select</td><td>
<table>
<tr><td rowspan="3"></td><td colspan="4">NORMAL1/2, IDLE1/2 mode</td><td rowspan="3">(Note 4)</td><td rowspan="3">SLOW, SLEEP mode</td></tr>
<tr><td colspan="2">DV7CK = 0, DVCK = 00</td><td colspan="2">DV7CK = 1, DVCK = 00</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>000</td><td>fc/2^13</td><td>fc/2^14</td><td>fs/2^5</td><td>fs/2^5</td><td>DV10/2</td><td>fs/2^5</td></tr>
<tr><td>001</td><td>fc/2^8</td><td>fc/2^9</td><td>fc/2^8</td><td>fc/2^9</td><td>DV5/2</td><td>–</td></tr>
<tr><td>010</td><td>fc/2^7</td><td>fc/2^8</td><td>fc/2^7</td><td>fc/2^8</td><td>DV4/2</td><td>–</td></tr>
<tr><td>011</td><td>fc/2^6</td><td>fc/2^7</td><td>fc/2^6</td><td>fc/2^7</td><td>DV3G/2</td><td>–</td></tr>
<tr><td>100</td><td>fc/2^5</td><td>fc/2^6</td><td>fc/2^5</td><td>fc/2^6</td><td>DV2G/2</td><td>–</td></tr>
<tr><td>101</td><td>(fc/2^4 (Note 5))</td><td>fc/2^5</td><td>(fc/2^4 (Note 5))</td><td>fc/2^5</td><td>DV1G/2</td><td>–</td></tr>
<tr><td>110</td><td colspan="6">Reserved</td></tr>
<tr><td>111</td><td colspan="6">External clock (input from $\overline{\text{SCK}}$ pin)</td></tr>
</table>
</td><td></td></tr>
</table>

Note 1: fc : High-frequency clock [Hz], fs : Low-frequency clock [Hz]
Note 2: Set SIOS to "0" and SIOINH to "1" when setting the transfer mode or serial clock.
Note 3: SIOCR1 is a write-only register and must not be used with any of the read-modify-write instructions.
Note 4: When selecting DV1G, DV3G, DV4, or DV5 for the serial clock, check Table 1-1 in Section 1, 5, 3, "Timing Generator," to see that the divider output is possible (marked with ○). If the divider output is impossible (marked with ×), do not choose the serial clock.
Note 5: When DVCK = "00", fc/2^4 cannot be selected for the operating clock.

SIO Status Register

	7	6	5	4	3	2	1	0	
SIOSR (0027_H)	SIOF	SEF	"1"	"1"	"1"	"1"	"1"	"1"	(Initial value: 0011 1111)

SIOF	Serial transfer operating status monitor	0: Transfer terminated 1: Transfer in process (After SIOS is cleared to "0", SIOF is cleared to "0" at the termination of transfer or setting of SIOINH.)	Read only
SEF	Shift operating status monitor	0: Shift operation terminated 1: Shift operation in process	

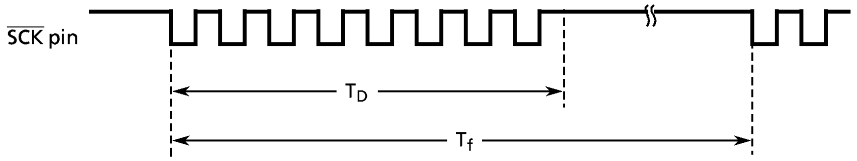
SIO Control Register 2

	7	6	5	4	3	2	1	0	
SIOCR2 (0027_H)				WAIT		BUF			(Initial value: ***0 0000)

WAIT	Wait control	Always sets "00" except 8-bit transmit/receive mode. 00: $T_f = T_D$ (non-wait) 01: $T_f = 2T_D$ (wait) 10: $T_f = 4T_D$ (wait) 11: $T_f = 8T_D$ (wait)	Write only
BUF	Number of transfer words	Buffer address used SIO 000: 1 word transfer 0028_H 001: 2 words transfer 0028_H to 0029_H 010: 3 words transfer 0028_H to $002A_H$ 011: 4 words transfer 0028_H to $002B_H$ 100: 5 words transfer 0028_H to $002C_H$ 101: 6 words transfer 0028_H to $002D_H$ 110: 7 words transfer 0028_H to $002E_H$ 111: 8 words transfer 0028_H to $002F_H$	

Figure 2-50. SIO Control Registers and Status Registers (1/2)

Note 1: T_f: frame time, T_D : data transfer time



Note 2: The lower 4 bits of each buffer are used during 4-bit transfers. Zeros (0) are stored to the upper 4bits when receiving.

Note 3: Transmitting starts at the lowest address. Received data are also stored starting from the lowest address to the highest address. For example, in the case of SIO, the first buffer address transmitted is $0FF0_H$.

Note 4: The value to be loaded to BUF is held after transfer is completed.

Note 5: SIOCR2 must be set when the serial interface is stopped (SIOF = 0).

Note 6: SIOCR2 is a write-only register and must not be used with any of the read-modify-write instructions.

Note 7: ∗: Don't care

Figure 2-50. SIO Control Registers and Status Registers (2/2)

(1) Serial Clock

a. Clock Source

SCK (bits 2 to 0 in SIOCR) is able to select the following:

① Internal Clock

Any of four frequencies can be selected. The serial clock is output to the outside on the $\overline{\text{SCK}}$ pin. The $\overline{\text{SCK}}$ pin goes high when transfer starts.

In order to extract internal clock, set the output latch of P42 to "1" and set the terminal to output mode.

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

Table 2-16. Serial Clock Rate

(DVCK = "01", fc = 16 MHz, fs = 32.768 kHz)

	NORMAL1/2, IDLE1/2 Mode				SLOW, SLEEP Mode	
	DV7CK = 0		DV7CK = 1			
SCK	Clock	Baud rate	Clock	Baud rate	Clock	Baud rate
000	$fc/2^{13}$	1.91 Kbps	$fs/2^{5}$	1024 bps	$fs/2^{5}$	1024 bps
001	$fc/2^{8}$	61.04 Kbps	$fc/2^{8}$	61.04 Kbps	—	—
010	$fc/2^{7}$	122.07 Kbps	$fc/2^{7}$	122.07 Kbps	—	—
011	$fc/2^{6}$	244.14 Kbps	$fc/2^{6}$	244.14 Kbps	—	—
100	$fc/2^{5}$	488.28 Kbps	$fc/2^{5}$	488.28 Kbps	—	—
101	$fc/2^{5}$	488.28 Kbps	$fc/2^{5}$	488.28 Kbps	—	—
110	—	—	—	—	—	—
111	External		External		External	

Note: 1 Kbps = 1024 bps

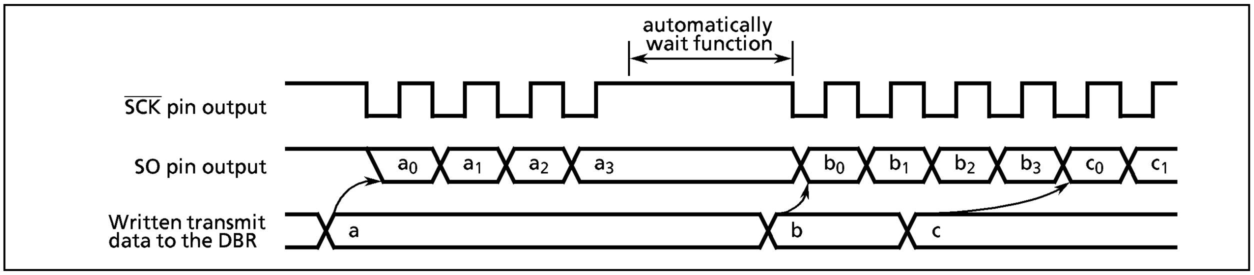
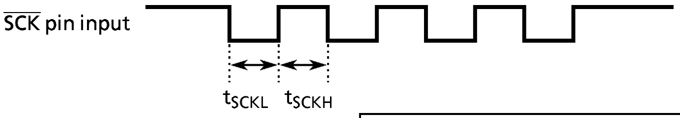


Figure 2-51. Clock Source (Internal Clock)

② External Clock

An external clock connected to the $\overline{SCK}$ pin is used as the serial clock. In this case, the P42 ($\overline{SCK}$) must be set to the input mode. To ensure shifting, a pulse width of at least 4 machine cycles is required. This pulse is needed for the shift operation to execute certainly. Actually, there is necessary processing time for interrupting, writing, and reading. The minimum pulse is determined by setting the mode and the program.



t_{SCKL}, t_{SCKH} > 4 tcyc

Note: tcyc = 4/fcgck (In NORAML1/2, IDLE1/2 modes)
4/fs (In SLOW, SLEEP modes)

b. Shift edge

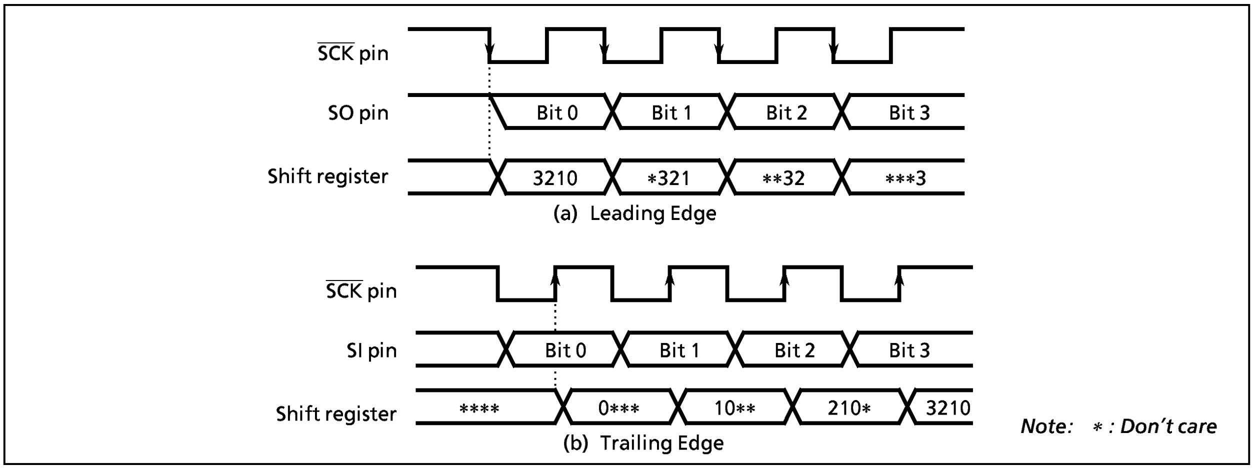
The leading edge is used to transmit, and the trailing edge is used to receive.

① Leading Edge

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

② Trailing Edge

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



(a) Leading Edge

(b) Trailing Edge

*Note: * : Don't care*

Figure 2-52. Shift Edge

(2) Number of Bits to Transfer

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

(3) Number of Words to Transfer

Up to 8 words consisting of 4 bits of data (4-bit serial transfer) or 8 bits (8-bit serial transfer) of data can be transferred continuously. The number of words to be transferred is loaded to BUF.
An INTSIO interrupt is generated when the specified number of words has been transferred. If the number of words is to be changed during transfer, the serial interface must be stopped before making the change. The number of words can be changed during automatic-wait operation of an internal clock. In this case, the serial interface is not required to be stopped.

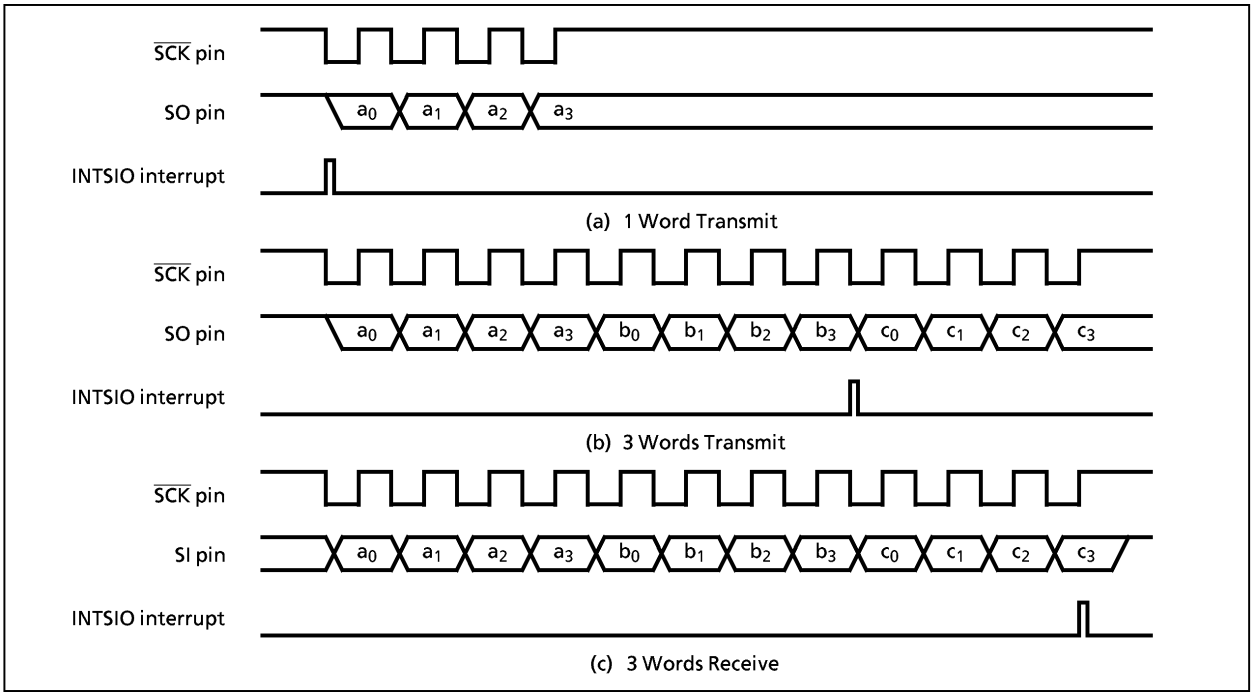


Figure 2-53. Number of Bits to Transfer (Example: 4-bit serial transfer)

(4) Transfer Mode

SIOM (bits 5 to 3 in SIOCR1) is used to select the transmit, receive, or transmit/receive mode.

a) 4-bit and 8-bit Transmit Modes

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

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

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

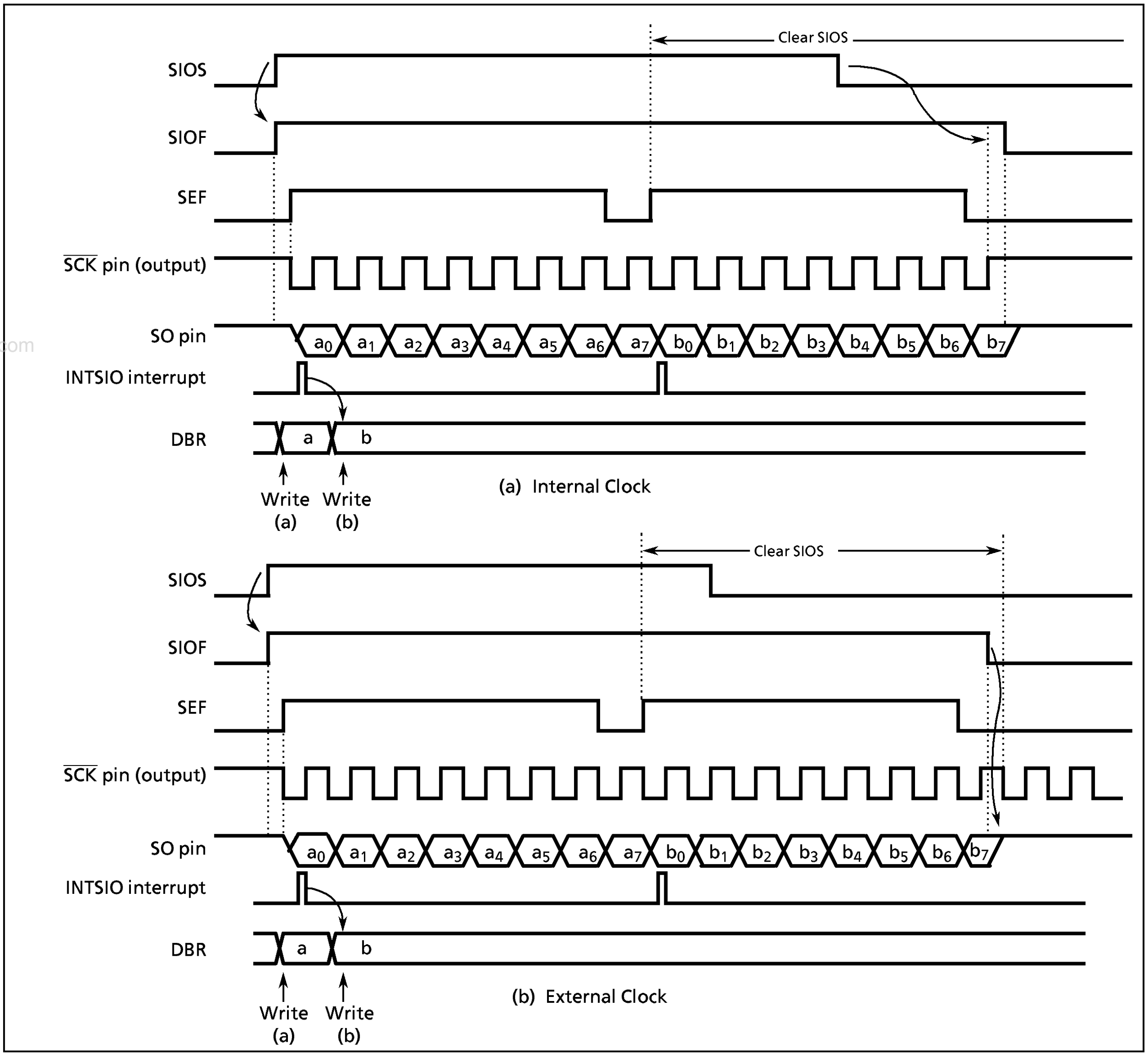
When an external clock is used, the data must be written to the data buffer register before shifting next data. Thus, the transfer speed is determined by the maximum delay time from the generation of the interrupt request to writing of the data to the data buffer register by the interrupt service program.

The transmission is ended by clearing SIOS to “0” or setting SIOINH to “1” in buffer empty interrupt service program. That the transmission has ended can be determined from the status of SIOF (bit 7 in SIOSR) because SIOF is cleared to “0” when a transfer is completed.

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

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

When an external clock is used, it is also necessary to clear SIOS to “0" before shifting the next data; otherwise, dummy data will be transmitted and the operation will end.



(a) Internal Clock

(b) External Clock

Figure 2-54. Transfer Mode (Example: 8-bit, 1 Word Transfer)

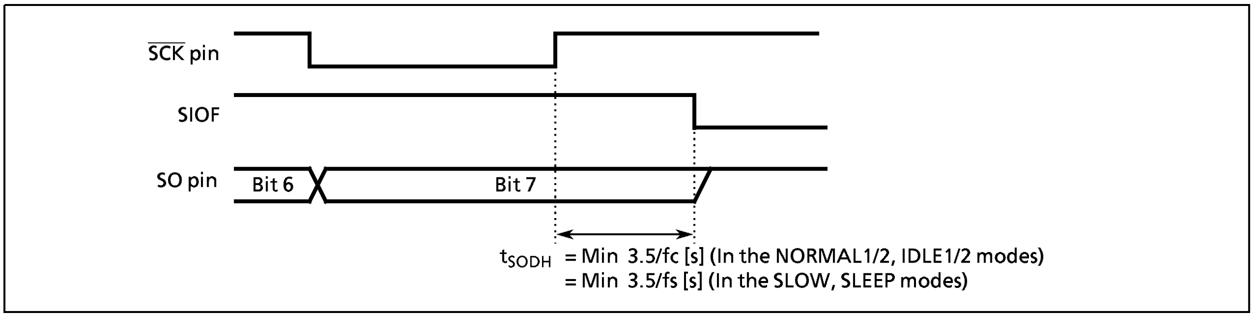

SCK pin
SIOF
SO pin
Bit 6
Bit 7
tSODH = Min 3.5/fc [s] (In the NORMAL1/2, IDLE1/2 modes)
= Min 3.5/fs [s] (In the SLOW, SLEEP modes)

Figure 2-55. Transmitted Data Hold Time at End of Transmit

b) 4-bit and 8-bit Receive Modes

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

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

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

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

The receiving is ended by clearing SIOS to “0” or setting SIOINH to “1” in buffer full interrupt service program. When SIOINH is set, the receiving is immediately ended and SIOF is cleared to “0”. When SIOS is cleared, the current data are transferred to the buffer. After SIOS cleared, the transmissions is ended at the time that the final bit of the data being shifted has been output. That the transmission has ended can be determined from the status of SIOF (bit 7 in SIOSR). SIOF is cleared to “0” when the transmission is ended.

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

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of data receive, BUF must be rewritten before the received data is read out.

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

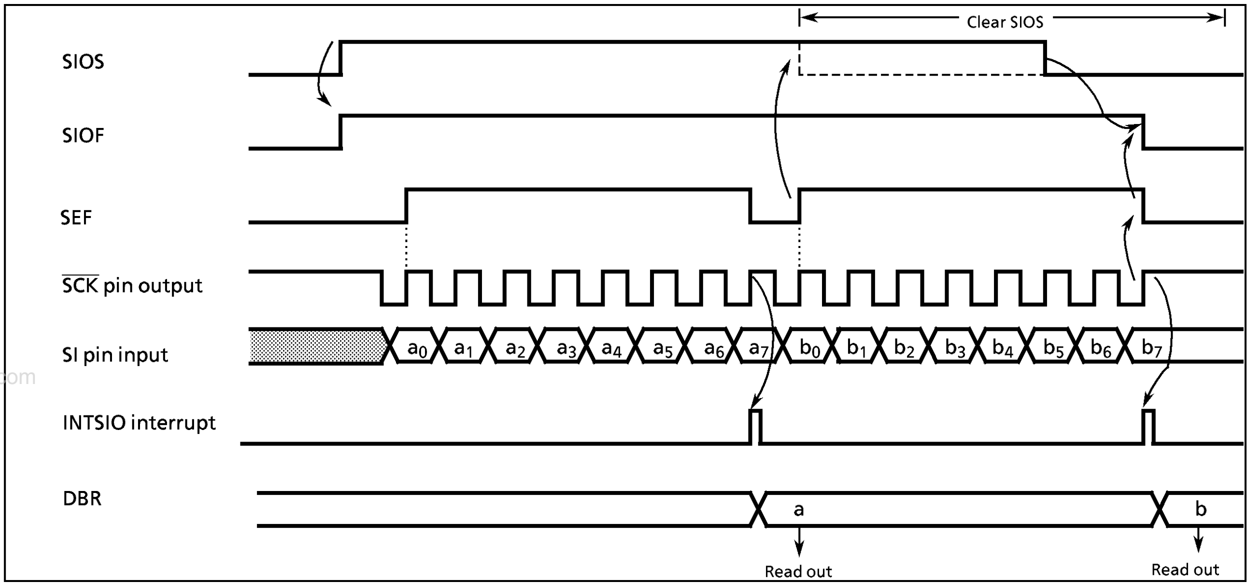


Figure 2-56. Receive Mode (Example: 8-bit, 1 word, internal clock)

(3) 8-bit Transmit/Receive Mode

After setting the control registers to the 8-bit transmit/receive mode, write the data to be transmitted first to the data buffer registers (DBR). After that, enable transceiving by setting SIOS to “1”. When transmitting, the data are output from the SO pin at leading edges of the serial clock. When receiving, the data are input to the SI pin at the trailing edges of the serial clock. 8-bit data are transferred from the shift register to the data buffer register. An INTSIO interrupt is generated when the number of data words specified with the BUF has been transferred. The interrupt service program reads the received data from the data buffer register and then writes the data to be transmitted. The data buffer register is used for both transmitting and receiving; therefore, always write the data to be transmitted after reading the received data.
When the internal clock is used, a wait is initiated until the received data are read and the next data are written. A wait will not be initiated if even one data word has been written.

Note: An automatic-wait is also canceled by writing to a DBR not being used as a transmit data buffer registers; therefore, during SIO do not use such DBR for other applications.

When an external clock is used, the shift operation is synchronized with the external clock; therefore, it is necessary to read the received data and write the data to be transmitted next before starting the next shift operation. When an external clock is used, the transfer speed is determined by the maximum delay between generation of an interrupt request and the received data are read and the data to be transmitted next are written.
Clear SIOS to “0” to SIOINH to “1” in INTSIO interrupt service program to end transmit/receive mode. The transmit/receive is ended at the time that the final bit of the data being shifted has been output.

The end of transmit/receive can be determined from the status of SIOF (bit 7 in SIOSR). SIOF is cleared to "0" when the transmit/receive is ended. When SIOINH is set, the transmit/receive is immediately ended and SIOF is cleared to "0".
If the number of words is to be changed during transfer, SIOS must be cleared to "0" and BUF is rewritten after confirmed that SIOF clearing to "0". The number of words can be changed during automatic-wait operation of an internal clock. In this case, BUF must be rewritten before the final transmitted/received data is read out.

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

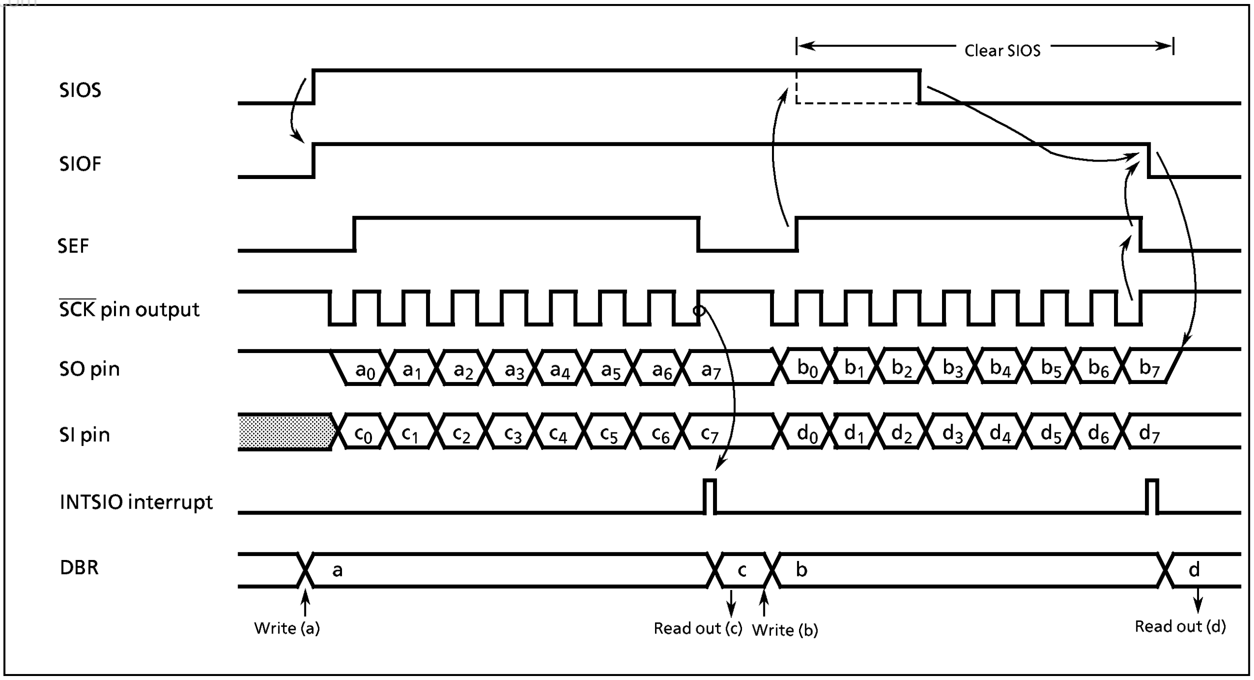
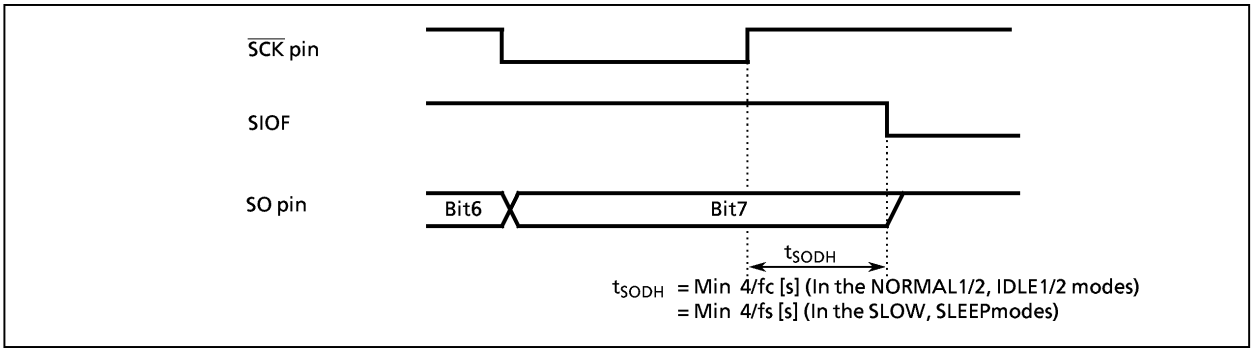


Figure 2-57. Transmit/Receive Mode (Example: 8-bit, 1word, internal clock)



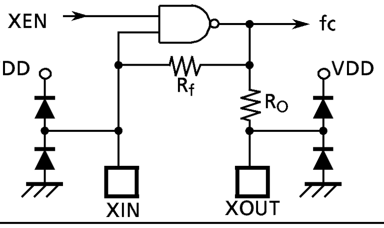
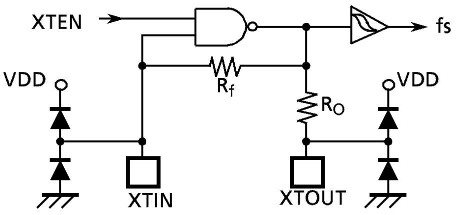
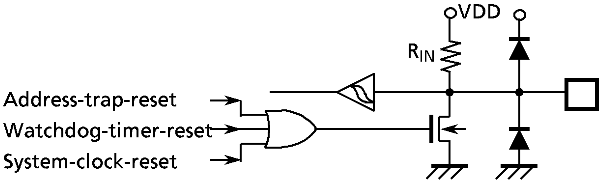
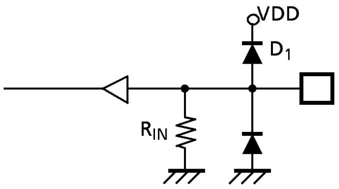
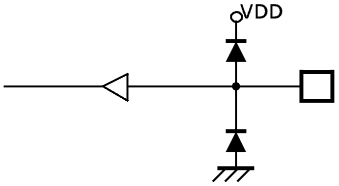
t_{SODH} = Min 4/fc [s] (In the NORMAL1/2, IDLE1/2 modes)
= Min 4/fs [s] (In the SLOW, SLEEPmodes)

Figure 2-58. Transmitted Data Hold Time at End of Transmit/Receive

Input/Output Circuitry

(1) Control Pins

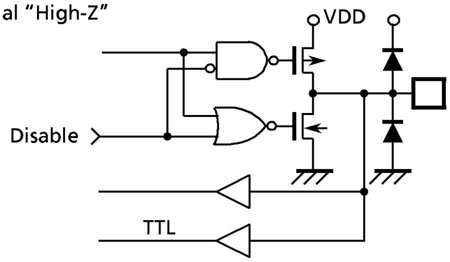
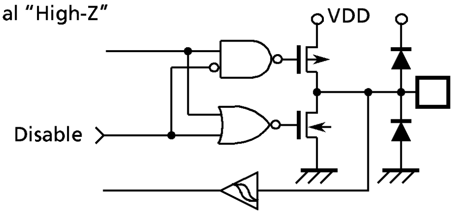
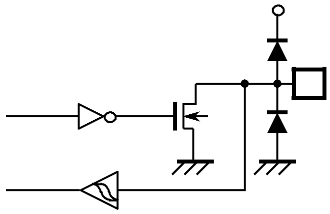
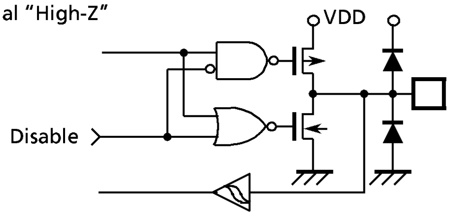
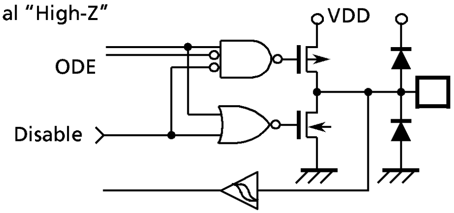
The input/output circuitries of the TMP86CH06 control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remarks
XIN XOUT	Input Output		Resonator connecting pins (high-frequency) R_f = 1.2 MΩ (typ.) R_O = 1 kΩ (typ.)
XTIN XTOUT	Input Output		Resonator connecting pins (low-frequency) R_f = 6 MΩ (typ.) R_O = 220 kΩ (typ.)
$\overline{\text{RESET}}$	I/O	 VDD R_{IN} Address-trap-reset Watchdog-timer-reset System-clock-reset	Sink open drain output Hysteresis input Built in Pull-up resistor R_{IN} = 220 kΩ (typ.)
TEST	Input		Built in Pull-down resistor R_{IN} = 70 kΩ (typ.)
$\overline{\text{EA}}$	Input		

Note: The TMP86PH06 does not have a pull-down resistor (R_{IN}) and a diode (D_1) for TEST pin. Be sure to fix the TEST pin to low level in MCU mode.

(2) Input/Output Ports

The input/output circuitries of the TMP86CH06 input/output ports are shown below.

Port	I/O	Input/Output Circuitry	Remarks
P0	I/O	Initial "High-Z" 	Tri-state I/O Nch. High-currency output
P1	I/O	Initial "High-Z" 	Tri-state I/O Hysteresis input
P2	I/O	Initial "High-Z" 	Sink open drain output Hysteresis input
P3	I/O	Initial "High-Z" 	Tri-state I/O Hysteresis input
P4	I/O	Initial "High-Z" 	Tri-state I/O Hysteresis input Programmable Open Drain output

Electrical Characteristics

Absolute Maximum Rating (V_{SS} = 0 V)

Parameter	Symbol	Pins	Rating	Unit
Supply Voltage	V_{DD}		– 0.3 to 6.5	V
Input Voltage	V_{IN}		– 0.3 to V_{DD} + 0.3	
Output Voltage	V_{OUT}		– 0.3 to V_{DD} + 0.3	
Output Current	I_{OUT1}	P1 to P4	3.2	mA
	I_{OUT3}	P0	30	
Output Current	ΣI_{OUT1}		80	
	ΣI_{OUT3}		120	
Power Dissipation (Topr = 85°C)	PD	TMP86CH06N	600	mW
		TMP86CH06U	350	
Soldering Temperature (Time)	Tsld		260 (10 s)	°C
Storage Temperature	Tstg		– 55 to 125	
Operating Temperature	Topr		– 40 to 85	

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

Recommended Operating Conditions

(V_{SS} = 0 V, Topr = – 40 to 85°C)

Parameter	Symbol	Pins	Conditions		Min	Max	Unit
Supply Voltage	V_{DD}		fc = 16 MHz	NORMAL1, 2 mode	4.5	5.5	V
				IDLE0, 1, 2 mode			
			fc = 8 MHz	NORMAL1, 2 mode	2.7		
				IDLE0, 1, 2 mode			
			fc = 4.2 MHz	NORMAL1, 2 mode	1.8		
				IDLE0, 1, 2 mode			
			fs = 32.768 kHz	SLOW1, 2 mode			
				SLEEP0, 1, 2 mode			
				STOP mode	1.8		
Input High Voltage	V_{IH1}	Except hysteresis and TTL input	$V_{DD} \geqq 4.5$ V		$V_{DD} \times 0.70$	V_{DD}	V
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
	V_{IH3}	Except TTL input	$V_{DD} < 4.5$ V		$V_{DD} \times 0.90$		
	V_{IH4}	TTL input (Data bus)	$V_{DD} = 5$ V		2.2		
	V_{IH5}		$V_{DD} = 1.8$ V		$V_{DD} - 0.2$		
Input Low Voltage	V_{IL1}	Except hysteresis and TTL input	$V_{DD} \geqq 4.5$ V		0	$V_{DD} \times 0.30$	V
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
	V_{IL3}	Except TLL input	$V_{DD} < 4.5$ V			$V_{DD} \times 0.10$	
	V_{IL4}	TTL input (Data bus)	$V_{DD} = 5$ V			0.8	
	V_{IL5}		$V_{DD} = 1.8$ V			0.2	
Clock Frequency	fc	XIN, XOUT	V_{DD} = 4.5 V to 5.5 V		1.0	16	MHz
			V_{DD} = 2.7 V to 5.5 V			8	
			V_{DD} = 1.8 V to 5.5 V			4.2	
	fs	XTIN, XTOUT			30.0	34.0	kHz

Note 1: The recommended operating conditions for a device are operating conditions under which it can be guaranteed that the device will operate as specified. If the device is used under operating conditions other than the recommended operating conditions (supply voltage, operating temperature range, specified AC/DC values etc.), malfunction may occur. Thus, when designing products which include this device, ensure that the recommended operating conditions for the device are always adhered to.

Note 2: Clock Frequency fc ; The condition of supply voltage range is the value under NORMAL1/2 and IDLE0/1/2 mode.

Note 3: The minimum fc with clock gear is calculated as following formula with the ratio on divider n.
(Min fc) = (ratio on divider n) × 1 [MHz]

DC Characteristics

(V_{SS} = 0 V, Topr = – 40 to 85°C)

Parameter	Symbol	Pins	Conditions	Min	Typ.	Max	Unit
Hysteresis Voltage	V_{HS}	Hysteresis input		–	0.9	–	V
Input Current	I_{IN1}	TEST, $\overline{\text{EA}}$	V_{DD} = 5.5 V V_{IN} = 5.5 V/0 V	–	–	± 2	μA
	I_{IN2}	Sink Open Drain, Tri-state Port					
	I_{IN3}	$\overline{\text{RESET}}$, $\overline{\text{STOP}}$					
Input Resistance	R_{IN2}	$\overline{\text{RESET}}$		100	220	450	kΩ
	R_{IN3}	TEST		–	70	–	
OSC. Feedback Resistance	Rfx	XIN-XOUT		–	1.2	–	MΩ
	Rfxt	XTIN-XTOUT		–	6	–	
Output Leakage Current	I_{LO1}	Sink Open Drain Port	V_{DD} = 5.5 V, V_{OUT} = 5.5 V	–	–	2	μA
	I_{LO2}	Tri-state Port	V_{DD} = 5.5 V, V_{OUT} = 5.5 V/0 V	–	–	± 2	
"H" Output Voltage	V_{OH2}	Tri-state Port	V_{DD} = 4.5 V, I_{OH} = – 0.7 mA	4.1	–	–	V
"L" Output Voltage	V_{OL3}	Except P0 and XOUT	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	–	–	0.4	V
"L" Output Current	I_{OL1}	Except P0 and XOUT	V_{DD} = 4.5 V, V_{OL} = 0.4 V	1.6	–	–	mA
	I_{OL3}	P0	V_{DD} = 4.5 V, V_{OL} = 1.0 V	–	20	–	
Supply Current under NORMAL1, 2 mode	I_{DD}		V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V fc = 16 MHz fs = 32.768 kHz	–	5.5	7.0	mA
Supply Current under IDLE1, 2 mode				–	2.8	3.5	
Supply Current under NORMAL1, 2 mode			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V fc = 8 MHz fs = 32.768 kHz	–	4.0	5.0	mA
Supply Current under IDLE1, 2 mode				–	2.0	2.5	
Supply Current under SLOW1 mode			V_{DD} = 3.0 V V_{IN} = 2.8 V/0.2 V fs = 32.768 kHz	–	14	25	μA
Supply Current under SLEEP1 mode				–	7.0	15	μA
Supply Current under SLEEP0 mode				–	6.0	15	μA
Supply Current under STOP mode			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V	–	0.5	10	μA

Note 1: Typical values are shown under T_{opr} = 25°C, V_{DD} = 5 V, while conditions are not stated.
Note 2: Input current I_{IN1}, I_{IN3}: The current through pull-up or pull-down resistor is not included.

AC Characteristics

(1) (V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = – 40 to 85°C)

① CLOCK

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
Machine Cycle Time	tcy	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}	External clock operation (XIN input) fc = 16 MHz	25	–	–	ns
Low Level Clock Pulse Width	t_{WCL}					
High Level Clock Pulse Width	t_{WSH}	External clock operation (XTIN input) fs = 32.768 kHz	14.7	–	–	μs
Low Level Clock Pulse Width	t_{WSL}					

② External Memory Interface (Multiplexed Bus) at V_{DD} = 4.5 to 5.5

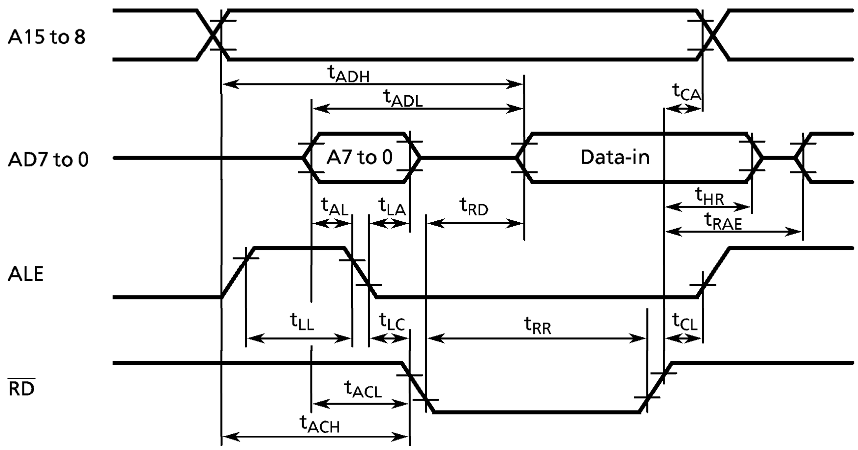
No.	Symbol	Parameter	Variable		16 MHz		Unit
			Min	Max	Min	Max	
1	t_{AL}	A7 to 0 effective → ALE	0.5t – 15		16		ns
2	t_{LA}	ALE fall → A7 to 0 hold	0.5t – 20		11		ns
3	t_{LL}	ALE pulse width	t – 40		22		ns
4	t_{LC}	ALE fall → $\overline{RD}$, $\overline{WR}$ fall	0.5t – 25		6		ns
5	t_{CL}	$\overline{RD}$, $\overline{WR}$ rise → ALE rise	0.5t – 20		11		ns
6	t_{ACL}	A7 to 0 effective → $\overline{RD}$, $\overline{WR}$ fall	t – 25		37		ns
7	t_{ACH}	A15 to 8 effective → $\overline{RD}$, $\overline{WR}$ fall	1.5t – 25		68		ns
8	t_{CA}	$\overline{RD}$, $\overline{WR}$ rise → A15 to 8 hold	0.5t – 20		11		ns
9	t_{ADL}	A7 to 0 effective → D7 to 0 input		3t – 55		132	ns
10	t_{ADH}	A15 to 8 effective → D7 to 0 input		3.5t – 65		153	ns
11	t_{RD}	$\overline{RD}$ fall → D7 to 0 input		2t – 60		65	ns
12	t_{RR}	$\overline{RD}$ pulse width	2t – 40		85		ns
13	t_{HR}	$\overline{RD}$ rise → D7 to 0 hold	0		0		ns
14	t_{RAE}	$\overline{RD}$ rise → A7 to 0 effective	t – 15		47		ns
15	t_{WW}	$\overline{WR}$ pulse width	2t – 40		85		ns
16	t_{DW}	D7 to 0 effective → $\overline{WR}$ rise	2t – 40		85		ns
17	t_{WD}	$\overline{WR}$ rise → D7 to 0 hold	0.5t – 15		16		ns

Note: t = tcy/4 (t = 62.5 ns at fcgck = 16 MHz)

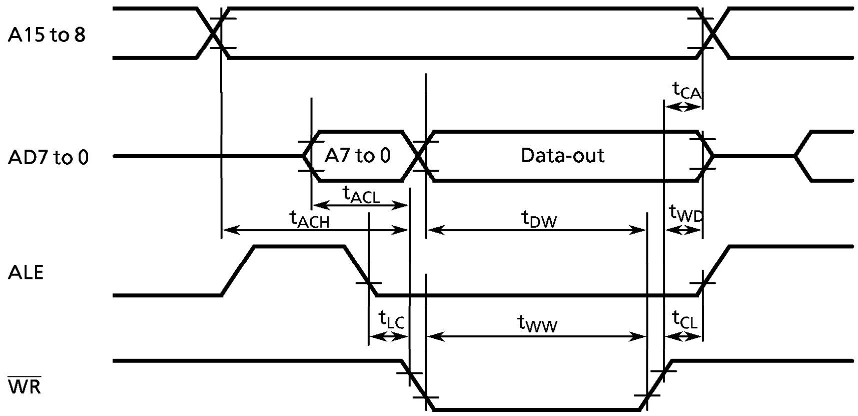
A.C.Measurement Condition

Output Level:	High 2.2 V/Low 0.8 V, CL = 50 pF
Input Level:	High 2.4 V/Low 0.4 V (D7 to D0)
	High 0.8 VDD/Low 0.2 VDD (Except D7 to D0)

Read Cycle



Write Cycle



Recommended Oscillating Conditions - 1

(V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = – 40 to 85°C)

Parameter	Oscillator	Oscillation Frequency	Recommended Oscillator		Recommended Constant	
					C_1	C_2
High-frequency Oscillation	Ceramic Resonator	16 MHz	MURATA	CSA16.00MXZ040	10 pF	10 pF
		8 MHz	MURATA	CSA8.00MTZ CST8.00MTW	30 pF 30 pF (built-in)	30 pF 30 pF (built-in)
		4.19 MHz	MURATA	CSA4.19MG CST4.19MGW	30 pF 30 pF (built-in)	30 pF 30 pF (built-in)
Low-frequency Oscillation	Crystal Oscillator	32.768 kHz	SII	VT-200	6 pF	6 pF

Recommended Oscillating Conditions - 2

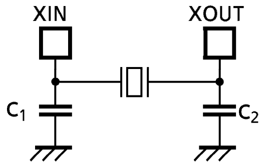
(V_{SS} = 0 V, V_{DD} = 2.7 to 5.5 V, Topr = – 40 to 85°C)

Parameter	Oscillator	Oscillation Frequency	Recommended Oscillator		Recommended Constant	
					C_1	C_2
High-frequency Oscillation	Ceramic Resonator	8 MHz	MURATA	CSA8.00MTZ CST8.00MTW	30 pF 30 pF (built-in)	30 pF 30 pF (built-in)
		4.19 MHz	MURATA	CSA4.19MG CST4.19MGW	30 pF 30 pF (built-in)	30 pF 30 pF (built-in)

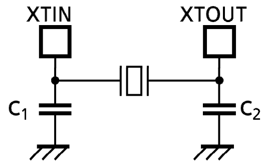
Recommended Oscillating Conditions - 3

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

Parameter	Oscillator	Oscillation Frequency	Recommended Oscillator		Recommended Constant	
					C_1	C_2
High-frequency Oscillation	Ceramic Resonator	4.19 MHz	MURATA	CSA4.19MG CST4.19MGW	30 pF 30 pF (built-in)	30 pF 30 pF (built-in)



(1) High-frequency Oscillation



(2) Low-frequency Oscillation

Note 1: An electrical shield by metal shield plate on the surface of IC package is recommended in order to protect the device from the high electric field stress applied from CRT (Cathodic Ray Tube) for continuous reliable operation.

Note 2: The product numbers and specifications of the resonators by Murata Manufacturing Co., Ltd. are subject to change. For up-to-date information, please refer to the following URL;

http://www.murata.co.jp/search/index.html