CMOS 8-Bit Microcontroller

TMP86P202P/M, TMP86P203P/M

The TMP86P202/203 are high-speed and high-performance 8-bit single chip microcomputers with small package. The MCU contain CPU core, ROM, RAM, multirole timer counter, and 8-bit AD converter, on a chip.

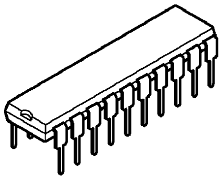
Product No.	ROM	RAM	Package	Resonator
TMP86P202P	2 K × 8 bits	128 × 8 bits	P-DIP20-300-2.54D	Ceramic, Crystal resonator
TMP86P202M*			P-SOP20-300-1.27	
TMP86P203P*			P-DIP20-300-2.54D	RC resonator
TMP86P203M*			P-SOP20-300-1.27	

*: Under development

Features

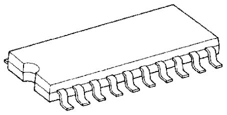
- ◆ 8-bit single chip microcomputer TLCS-870/C series
- ◆ Instruction execution time: 0.50 μs (fc = 8 MHz)
- ◆ 132 types and 731 basic instructions
- ◆ Interrupt sources: 11 factors (External: 3, Internal: 8)
- ◆ Input/Output ports: 14 pins
  - High-Current Output 2 pins (Typ. 20mA)
- ◆ 8-bit timer/counter: 2 ch
  - Timer, PDO output, Event counter, PWM output, PPG modes
  - Possible to use as 16-bit timer by connecting each other
- ◆ Time Base Timer
- ◆ Divider output function
- ◆ Watchdog Timer
  - Interrupt source/Internal Reset (programmable)
- ◆ 8-bit successive approximate type AD converter
  - Analog input: 4 ch

P-DIP20-300-2.54D



TMP86P202P
TMP86P203P*

P-SOP20-300-1.27



TMP86P202M*
TMP86P203M*

*: Under development

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.

◆ Power saving operating modes (3 modes)

- STOP mode: Oscillation stops. Battery/Capacitor back-up. Port output hold/High-impedance.
- IDLE 0 mode: CPU stops, and peripherals operate using high-frequency clock of Time-Base-Timer. Release by INTTBT interrupt.
- IDLE 1 mode: CPU stops, and peripherals operate using high-frequency clock. Release by interrupts.

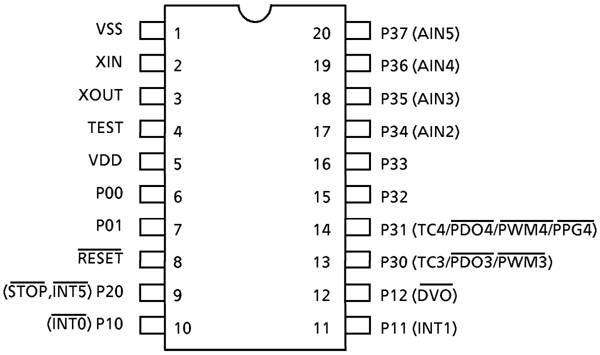
◆ Wide operating voltage: 3.3 to 5.5 V at 8 MHz (Ceramic resonator, Crystal resonator)
4.5 to 5.5 V at 2.5 MHz (RC resonator) (Under development)

Note: AD conversion characteristics are guaranteed with limited supply voltage range (4.5 V to 5.5 V). If supply voltage is less than 4.5 V then AD conversion accuracy can not be guaranteed.

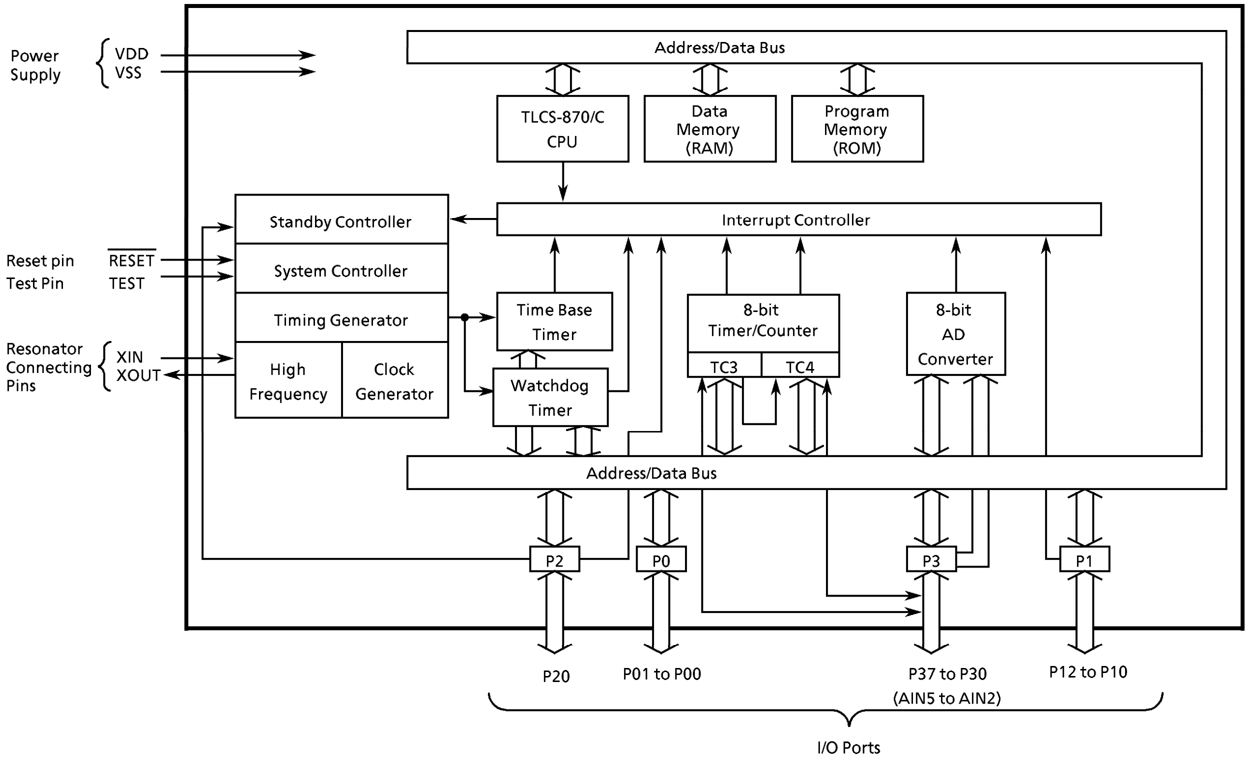
Pin Assignments (Top View)

P-DIP20-300-2.54D (for TMP86P202P and TMP86P203P*)
P-SOP20-300-1.27 (for TMP86P202M* and TMP86P203M*)

*: Under development


VSS 1
XIN 2
XOUT 3
TEST 4
VDD 5
P00 6
P01 7
$\overline{RESET}$ 8
($\overline{STOP}$,$\overline{INT5}$) P20 9
($\overline{INT0}$) P10 10
20 P37 (AIN5)
19 P36 (AIN4)
18 P35 (AIN3)
17 P34 (AIN2)
16 P33
15 P32
14 P31 (TC4/$\overline{PDO4}$/$\overline{PWM4}$/$\overline{PPG4}$)
13 P30 (TC3/$\overline{PDO3}$/$\overline{PWM3}$)
12 P12 ($\overline{DVO}$)
11 P11 (INT1)

Block Diagram



Pin Function

The TMP86P202/203 has two modes: MCU and PROM.

(1) MCU mode

Pin Name	Input/Output	Function	
P01	I/O	2-bit programmable input/output ports. Each bit of these ports can be individually configured as an input or output under software control. Nch open-drain output function.	
P00	I/O		
P12 ($\overline{\text{DVO}}$)	I/O (Output)	3-bit programmable input/output ports (tri-state). Each bit of these ports can be individually configured as an input or output under software control. When used as function, the latch must be set to 1.	Divider output
P11 (INT1)	I/O (Input)		External interrupt input 1
P10 ($\overline{\text{INT0}}$)	I/O (Input)		External interrupt input 0
P20 ($\overline{\text{INT5}}$, $\overline{\text{STOP}}$)	I/O (Input)	1-bit programmable input/output ports. When used as input port and function, the latch must be set to 1.	External interrupt input 5 or STOP mode release signal input
P37 (AIN5)	I/O (Input)	8-bit programmable input/output ports (tri-state). Each bit of these ports can be individually configured as an input or output under software control. When used as function and analog inputted latch must be set to 1.	AD converter analog input
P36 (AIN4)	I/O (Input)		
P35 (AIN3)	I/O (Input)		
P34 (AIN2)	I/O (Input)		
P33	I/O		
P32	I/O		
P31 (TC4, $\overline{\text{PDO4}}$, $\overline{\text{PWM4}}$, $\overline{\text{PPG4}}$)	I/O (I/O)		Timer/Counter 4 input, $\overline{\text{PDO}}$, $\overline{\text{PWM}}$, $\overline{\text{PPG}}$ output
P30 (TC3, $\overline{\text{PDO3}}$, $\overline{\text{PWM3}}$)	I/O (I/O)		Timer/Counter 3 input, $\overline{\text{PDO}}$, $\overline{\text{PWM}}$ output
XIN, XOUT	Input, Output	Resonator connecting pins for high-frequency clock. For inputting external clock, XIN is used and XOUT is opened.	
$\overline{\text{RESET}}$	Input	RESET signal input	
TEST	Input	TEST pin for out-going test. Be fixed to low.	
VDD, VSS	Power Supply	+ 5 V, 0 (GND)	

(2) PROM mode

Pin Name (PROM mode)	Input/Output	Functions	Pin name (MCU mode)
A16	Input	Program memory address inputs	XOUT
A15 to A8			P37 to P30
A7 to A0			P37 to P30
D7 to D0	I/O	Program memory data input/outputs	P37 to P30
$\overline{CE}$	Input	Chip enable signal input	P00
$\overline{OE}$		Output enable signal input	P20
$\overline{PGM}$		Program mode signal input	P01
DIDS		PROM mode control signal	P12
VPP	Power supply	+ 12.75 V/5 V (Program supply voltage)	TEST
VCC		+ 6.25 V/5 V	VDD
GND		0 V	VSS
P11	Input	PROM mode setting pins. Be fixed at high level.	
$\overline{RESET}$		PROM mode setting pins. Be fixed at low level.	
CLK		Input a clock from the outside.	XIN

Operational Description

1. CPU Core Functions

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

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

1.1 Memory Address Map

The TMP86P202/203 memory consist of 3 blocks: ROM, RAM and SFR (Special Function Register). They are all mapped in 64-Kbyte address space. Figure 1-1 shows the TMP86P202/203 memory address map. The general-purpose register banks are not assigned to the RAM address space.

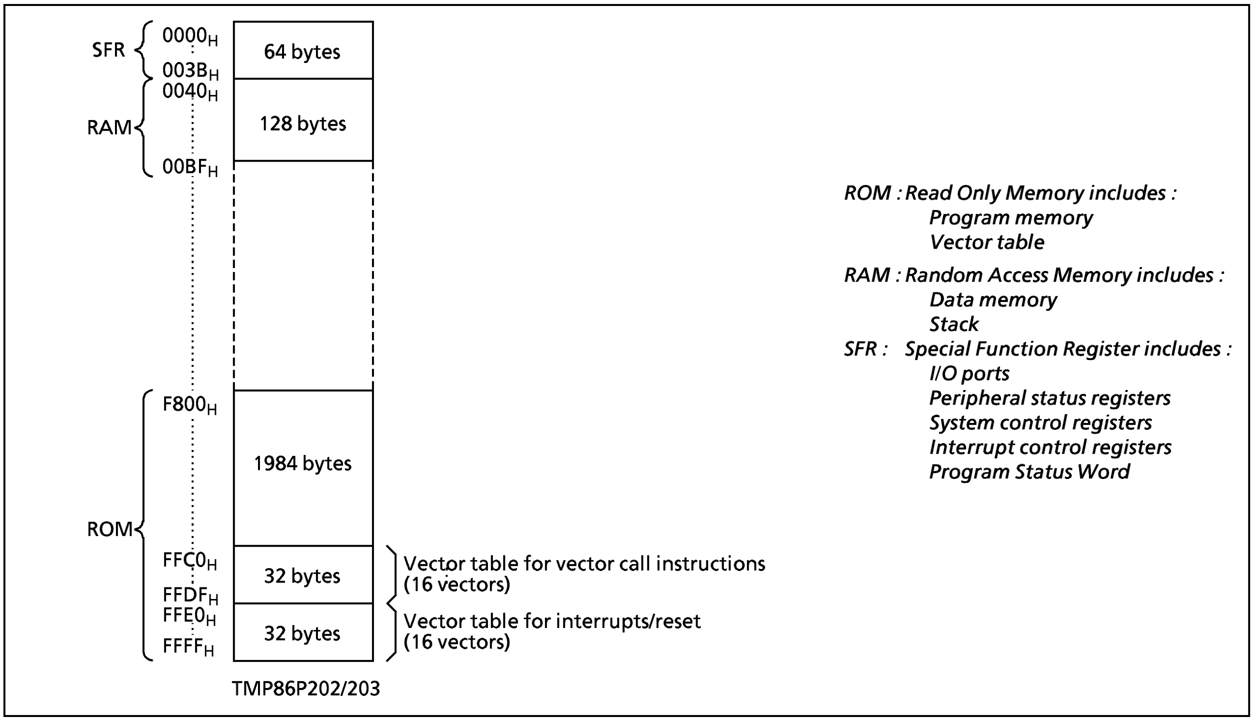

SFR
0000_H
$003B_H$
64 bytes
RAM
0040_H
$00BF_H$
128 bytes
ROM
$F800_H$
1984 bytes
$FFC0_H$
$FFDF_H$
32 bytes
Vector table for vector call instructions (16 vectors)
$FFE0_H$
$FFFF_H$
32 bytes
Vector table for interrupts/reset (16 vectors)
TMP86P202/203
ROM : Read Only Memory includes :
Program memory
Vector table
RAM : Random Access Memory includes :
Data memory
Stack
SFR : Special Function Register includes :
I/O ports
Peripheral status registers
System control registers
Interrupt control registers
Program Status Word

Figure 1-1. Memory Address Maps

1.2 Program Memory (ROM)

The TMP86P202/203 has a 2 K×8 bits (address $F800_H$ to $FFFF_H$), of program memory. However, placing program memory on the internal RAM is deregulated if a certain procedure is executed (See 2.4.5 Address Trap).

1.3 Data Memory (RAM)

The TMP86P202/203 have 128 bytes (0040 to 00BFH), of internal RAM. Internal RAM are located in the direct area; instructions with shorten operations are available against such an area.

Because the data memory contents become unstable upon power-up, execute the initialize routine to perform initial setting.

Example: RAM area of the TMP86P202/203

```
          LD   HL, 0040H  ;  Set start address
          LD   A, H       ;  Set initialization data (00H)
          LD   BC, 00FFH  ;  Set byte number ( − 1)
SRAMCLR:  LD   (HL), A    ;
          INC  HL
          DEC  BC
          JRS  F, SRAMCLR
```

1.4 System Clock Controller

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

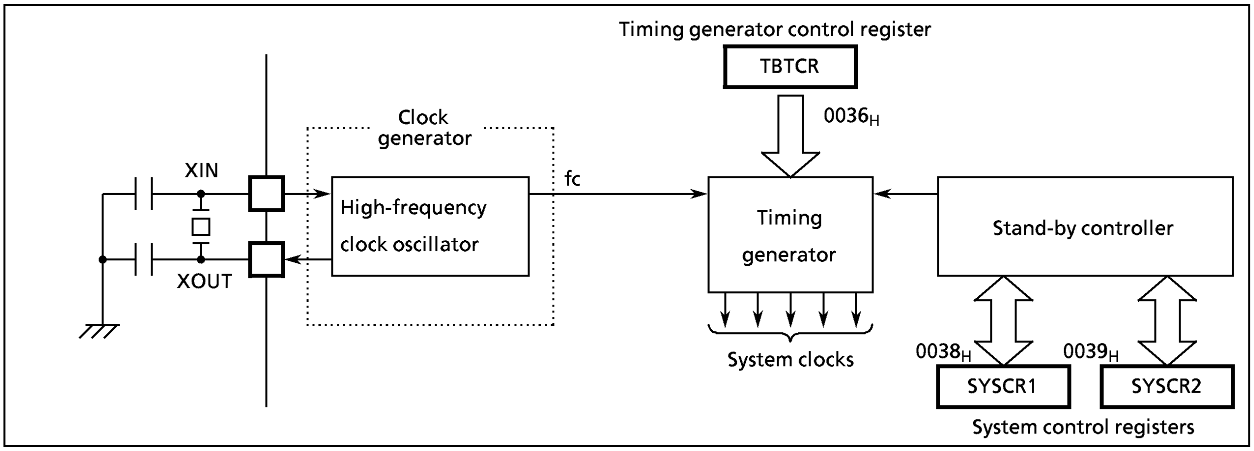


Figure 1-2. System Clock Control

1.4.1 Clock Generator

The Clock generator generates the basic clock which provides the system clocks supplied to the CPU core and peripheral hardware. It contains a oscillation circuit for the high-frequency clock.
The high-frequency (fc) clock can easily be obtained by connecting a resonator between the XIN and XOUT pins respectively. Clock input from an external oscillator is also possible. In this case, external clock is applied to XIN pin with XOUT pin not connected.

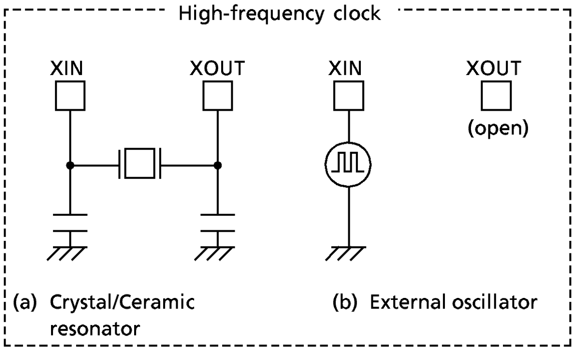


Figure 1-3. Examples of Resonator Connection

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.

1.4.2 Timing Generator

The timing generator generates the various system clocks supplied to the CPU core and peripheral hardware from the basic clock (fc). 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 2-stage prescaler, a 21-stage divider, a main system clock generator, and machine cycle counters.
As reset and STOP mode started/canceled, the prescaler and the divider are cleared to 0.

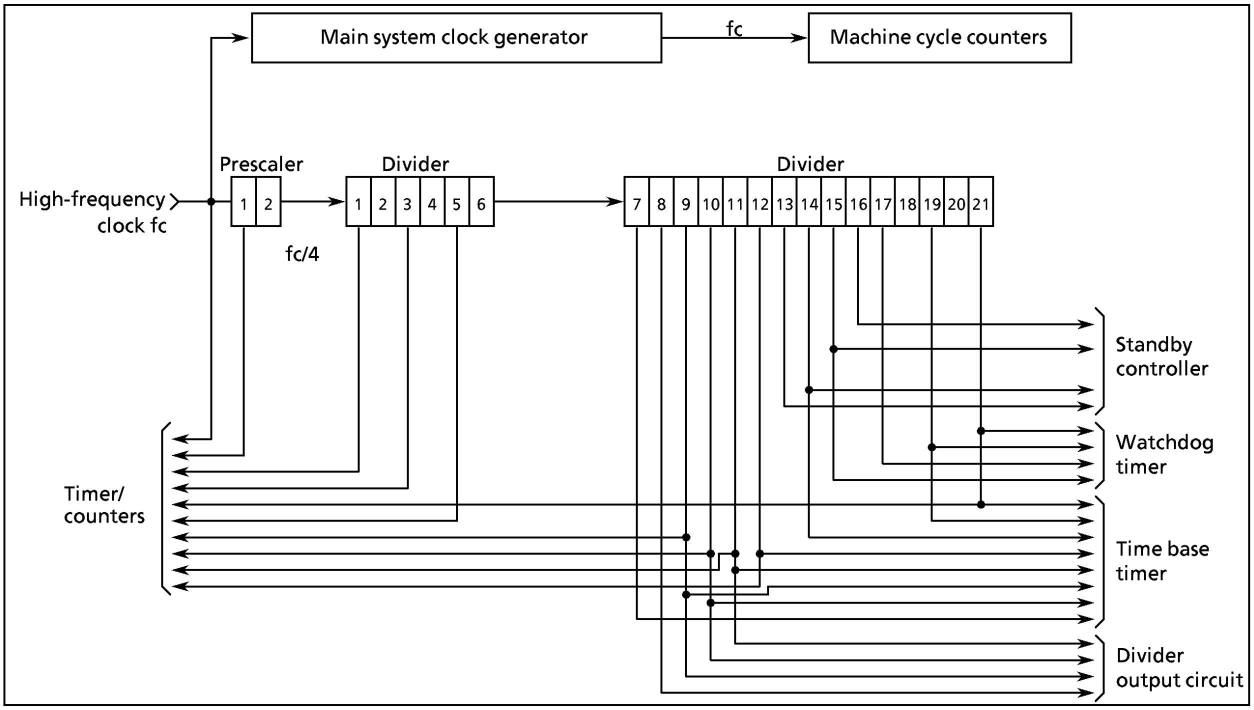


Figure 1-4. Configuration of Timing Generator

(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.

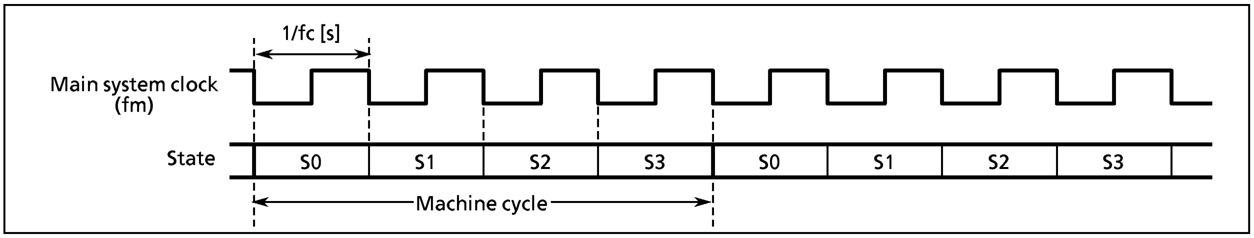


Figure 1-5. Machine Cycle

1.4.3 Standby Controller

The standby controller starts and stops the oscillation circuits for the high-frequency clock. Figure 1-6 shows the operating mode transition diagram and Figure 1-7 shows the system control registers. These modes are controlled by the system control registers (SYSCR1 and SYSCR2).

(1) Operating mode

① NORMAL1 mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock.
The TMP86P202/203 are 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.

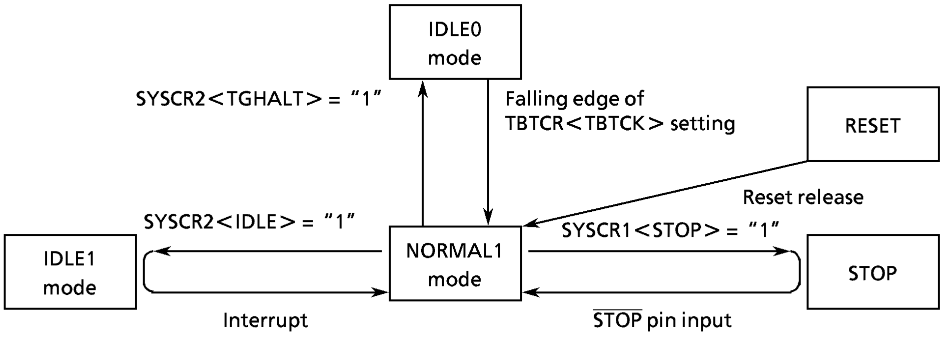
③ IDLE0 mode

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).
When IDLE0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR <TBTCK>, the timing generator starts feeding the clock to all peripheral circuits.
When returned from IDLE0 mode, the CPU restarts operating, entering NORMAL1 mode back again. IDLE0 mode is entered and returned regardless of how TBTCR <TBTEN> is set. When IMF = "1", EF_7 (TBT interrupt individual enable flag) = "1", and TBTCR <TBTEN> = "1", interrupt processing is performed. When IDLE0 mode is entered while TBTCR <TBTEN> = "1", the INTTBT interrupt latch is set after returning to NORMAL1 mode.

(2) 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.



Operating Mode		Oscillator	CPU core	TBT	Other Peripherals	Machine Cycle Time
Single Clock	RESET	Oscillation	Reset	Reset	Reset	4/fc [s]
	NORMAL1		Operate	Operate	Operate	
	IDLE1		Halt			
	IDLE0				Halt	—
	STOP	Stop		Halt		

Figure 1-6. Operating Mode Transition Diagram

System Control Register 1

SYSCR1 (0038_H) (Initial value: 00*0 00**)

7	6	5	4	3	2	1	0
STOP	RELM	"0"	OUTEN	WUT			

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 (Release by rising edge of $\overline{\text{STOP}}$ pin) 1: Level-sensitive release (Release by "H" level of $\overline{\text{STOP}}$ pin)	
OUTEN	Port output during STOP mode	0: High Impedance 1: Output Kept	
WUT	Warming-up time at releasing STOP mode	00: 3×2^{16}/fc 01: 2^{16}/fc 10: 3×2^{14}/fc 11: 2^{14}/fc	

Note 1: fc : High-frequency clock [Hz]
** : Don't care*
Note 2: Bits 1 and 0 in SYSCR1 are read as undefined data when a read instruction is executed.
Note 3: 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 4: Port P20 is used as $\overline{\text{STOP}}$ pin. Therefore, when stop mode is started, OUTEN does not affect to P20, and P20 becomes High-Z mode.
Note 5: Make sure to write "0" to bit 5 in SYSCR1.
Note 6: When STOP mode is released with $\overline{\text{RESET}}$ pin input, a return is made to NORMAL1.
Note 7: Before setting TGHALT to "1", be sure to stop peripherals. If peripherals are not stopped, the interrupt latch of peripherals may be set after IDLE0 or SLEEP0 mode is released.

System Control Register 2

SYSCR2 (0039_H) (Initial value: 1**0 *0**)

7	6	5	4	3	2	1	0
XEN	"0"	"0"	IDLE		TGHALT		

XEN	High-frequency oscillator control	0: Turn off oscillation 1: Turn on oscillation	R/W
IDLE	IDLE1 mode start	0: CPU and watchdog timer remain active 1: CPU and watchdog timer are stopped (start IDLE1 mode)	
TGHALT	IDLE0 mode start	0: TG provides source clock to all peripherals 1: TG stops source clock to peripherals except TBT (Start IDLE0 mode)	

Note 1: Make sure to write "0" to bit 5 and 6 in SYSCR2.
Note 2: A reset is applied, if XEN is cleared to "0".
*Note 3: * : Don't care*
Note 4: Bits 3, 1and 0 in SYSCR2 are always read as undefined value.
Note 5: Do not set SYSCR2<IDLE> and SYSCR2<TGHALT> to "1" at the same time.
Note 6: TG: Timing generator
Note 7: Because returning from IDLE0 to NORMAL1 is executed by the asynchronous internal clock, the period of IDLE0 mode might be shorter than the period setting by TBTCR<TBTCK>.
Note 8: When IDLE1 mode is released, IDLE is automatically cleared to "0".
Note 9: When IDLE0 mode is released, TGHALT is automatically cleared to "0".

Figure 1-7. System Control Registers

1.4.4 Operating Mode Control

(1) STOP mode

STOP mode is controlled by the system control register 1 (SYSCR1), 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 1: $\overline{\text{STOP}}$ pin doesn't have the control register. $\overline{\text{STOP}}$ pin should be fixed to low.
Note 2: 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 (P2PRD). 0             ;  Wait until the STOP pin input goes low level
             JRS   F, SSTOPH
             SET   (SYSCR1). 7            ;  Starts STOP mode
```

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

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

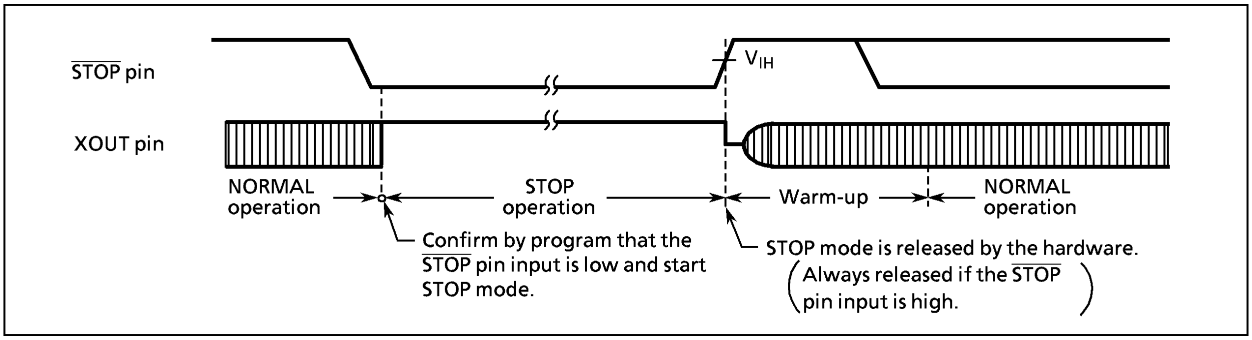

$\overline{\text{STOP}}$ pin
XOUT pin
V_{IH}
NORMAL operation
STOP operation
Warm-up
NORMAL operation
Confirm by program that the $\overline{\text{STOP}}$ pin input is low and start STOP mode.
STOP mode is released by the hardware.
(Always released if the $\overline{\text{STOP}}$ pin input is high.)

Figure 1-8. 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. <u>Edge-sensitive release mode</u> (RELM = 0)

In this mode, STOP mode is released by a rising edge of the $\overline{\text{STOP}}$ pin input. This is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (for example, a clock from a low-power consumption oscillator) is input to the $\overline{\text{STOP}}$ pin. In the edge-sensitive release mode, STOP mode is started even when the $\overline{\text{STOP}}$ pin input is high level.

Example: Starting STOP mode from NORMAL mode

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

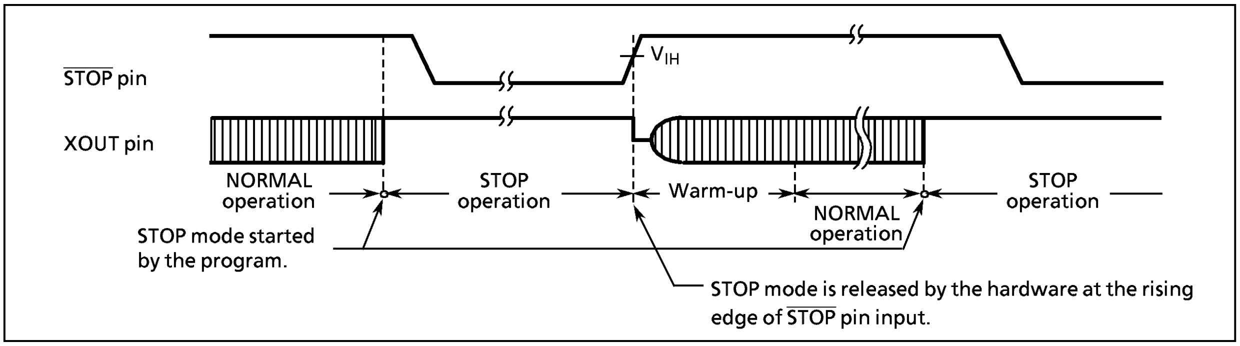

$\overline{\text{STOP}}$ pin
XOUT pin
V_{IH}
NORMAL operation
STOP operation
Warm-up
NORMAL operation
STOP operation
STOP mode started by the program.
STOP mode is released by the hardware at the rising edge of $\overline{\text{STOP}}$ pin input.

Figure 1-9. Edge-sensitive Release Mode

STOP mode is released by the following sequence.

① 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 1, 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.

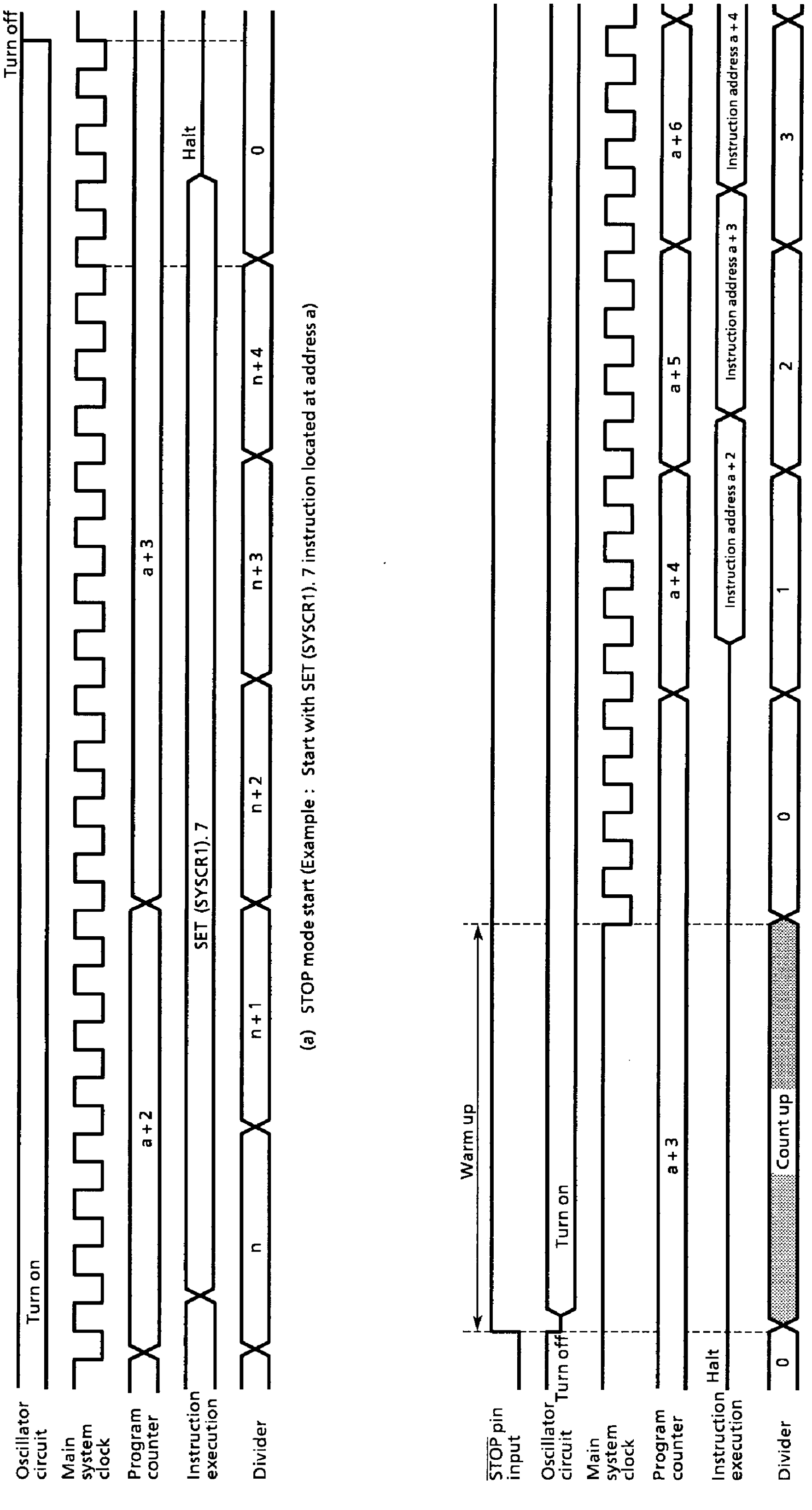
Table 1-1. Warm-up Time Example (at fc = 8.0 MHz)

WUT	Warm-up Time [ms]
	Return to NORMAL mode
000	24.576
010	8.192
100	6.144
110	2.048

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 1-10. STOP Mode Start/Release

(2) IDLE1 mode (IDLE1)

IDLE1 mode is controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following status is maintained during IDLE1 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 IDLE1 mode was entered.

③ The program counter holds the address 2 ahead of the instruction which starts IDLE1 mode.

Example: Starting IDLE1 mode.

```
            SET    (SYSCR2). 4     ; SYSCR2<IDLE> ←1
```

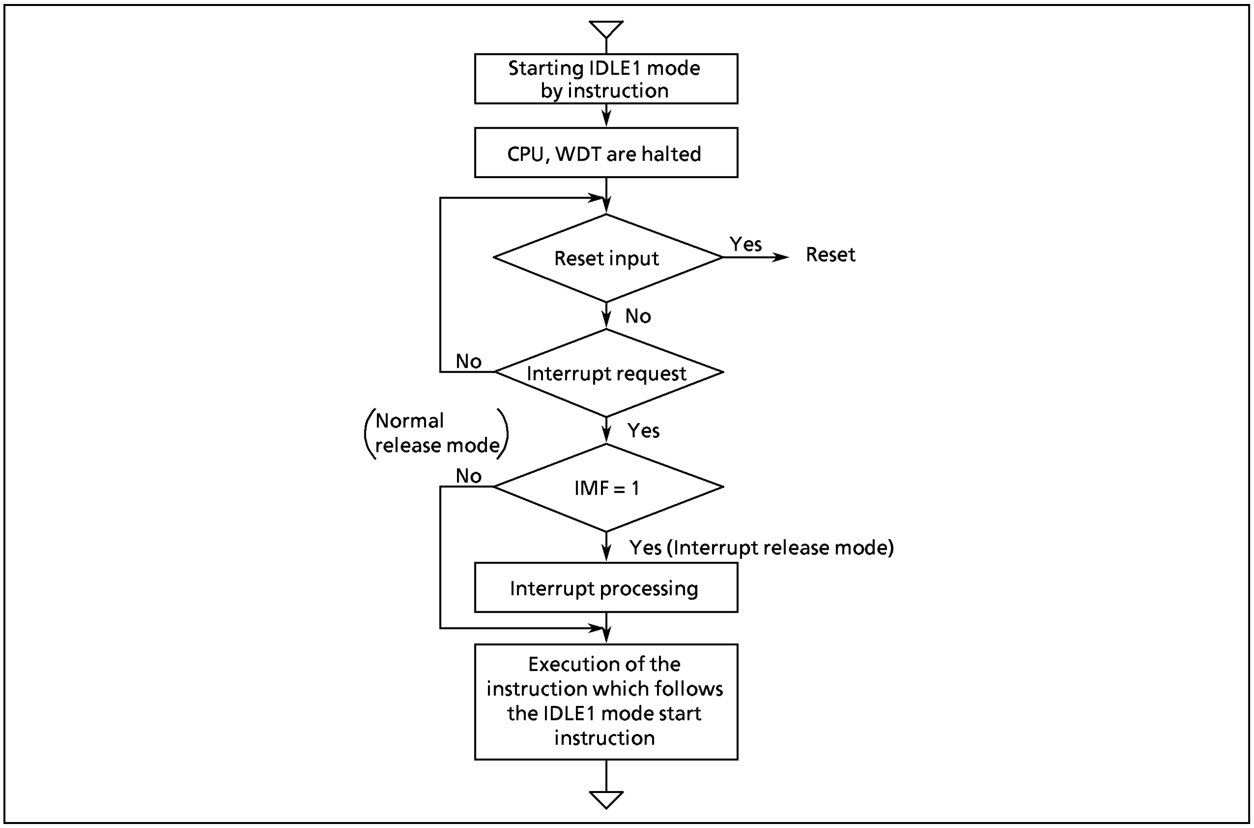

Starting IDLE1 mode by instruction
CPU, WDT are halted
Reset input
Yes
Reset
No
Interrupt request
No
Yes
(Normal release mode)
No
IMF = 1
Yes (Interrupt release mode)
Interrupt processing
Execution of the instruction which follows the IDLE1 mode start instruction

Figure 1-11. IDLE1 Mode

IDLE1 mode includes a normal release mode and an interrupt release mode. Selection is made with the interrupt master enable flag (IMF). Releasing IDLE1 mode returns from IDLE1 to NORMAL1.

IDLE1 mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin, which immediately performs the reset operation. After reset, the TMP86P202/203 are placed in NORMAL 1 mode.

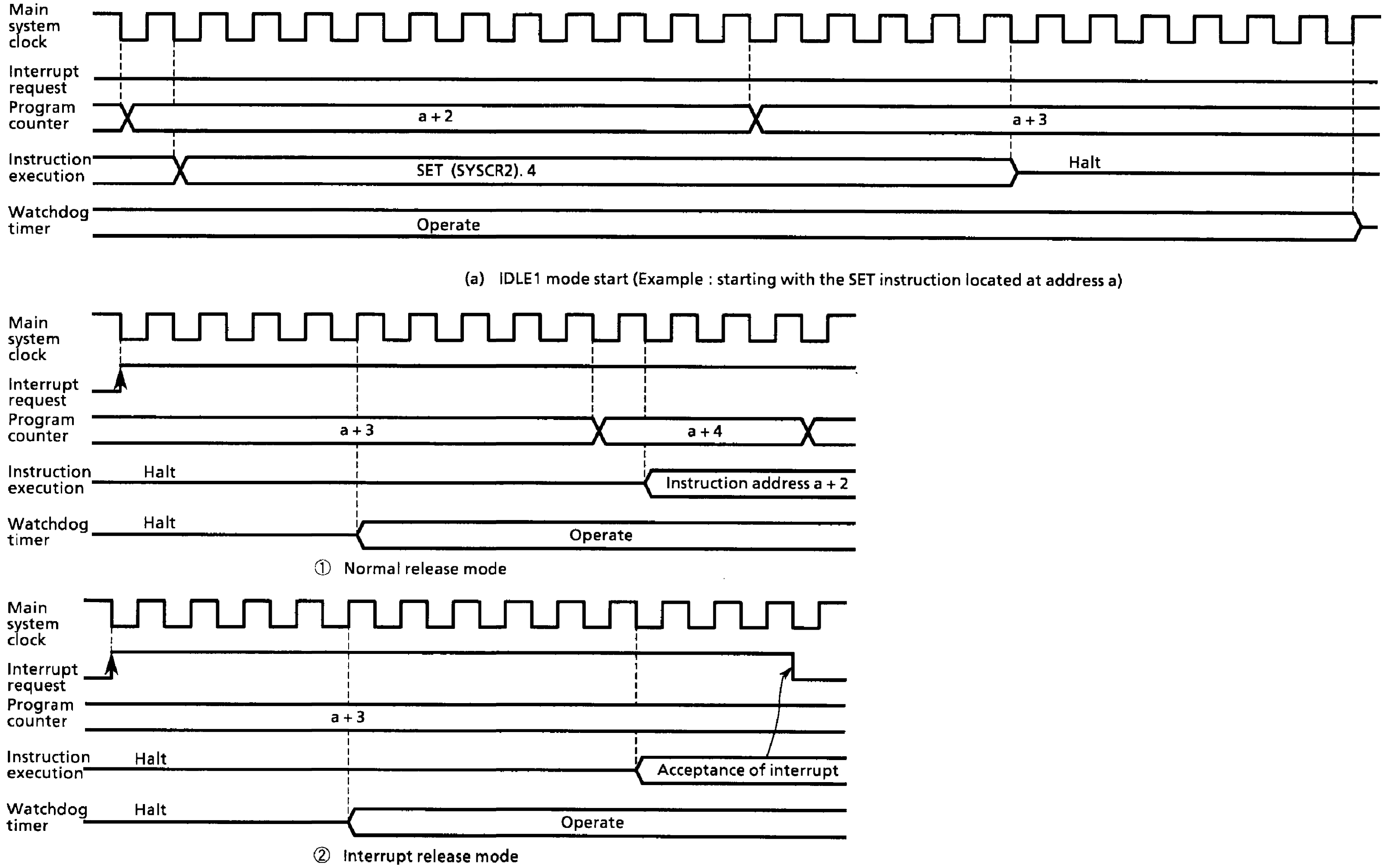
a. Normal release mode (IMF = "0")

IDLE1 mode is released by any interrupt source enabled by the individual interrupt enable flag (EF). Execution resumes with the instruction following the IDLE1 mode start instruction . The interrupt latches (IL) of the interrupt source used for releasing must be cleared to "0" by load instructions.

b. Interrupt release mode (IMF = "1")

IDLE1 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 IDLE1 mode.

Note: When a watchdog timer interrupt is generated immediately before IDLE1 mode is started, the watchdog timer interrupt will be processed but IDLE1 mode will not be started.



(a) IDLE1 mode start (Example : starting with the SET instruction located at address a)

① Normal release mode

② Interrupt release mode

(b) IDLE1 mode release

Figure 1-12. IDLE1 Mode Start/Release

(3) IDLE0 mode (IDLE0)

IDLE0 mode is controlled by the system control register 2 (SYSCR2) and the time base timer control register (TBTCR). The following status is maintained during IDLE0 mode.

① Timing generator stops feeding clock to peripherals except TBT.

② The data memory, CPU registers, program status word and port output latches are all held in the status in effect before IDLE0 mode was entered.

③ The program counter holds the address 2 ahead of the instruction which starts IDLE0 mode.

Note: Before starting IDLE0 mode, be sure to stop (disable) periperals.

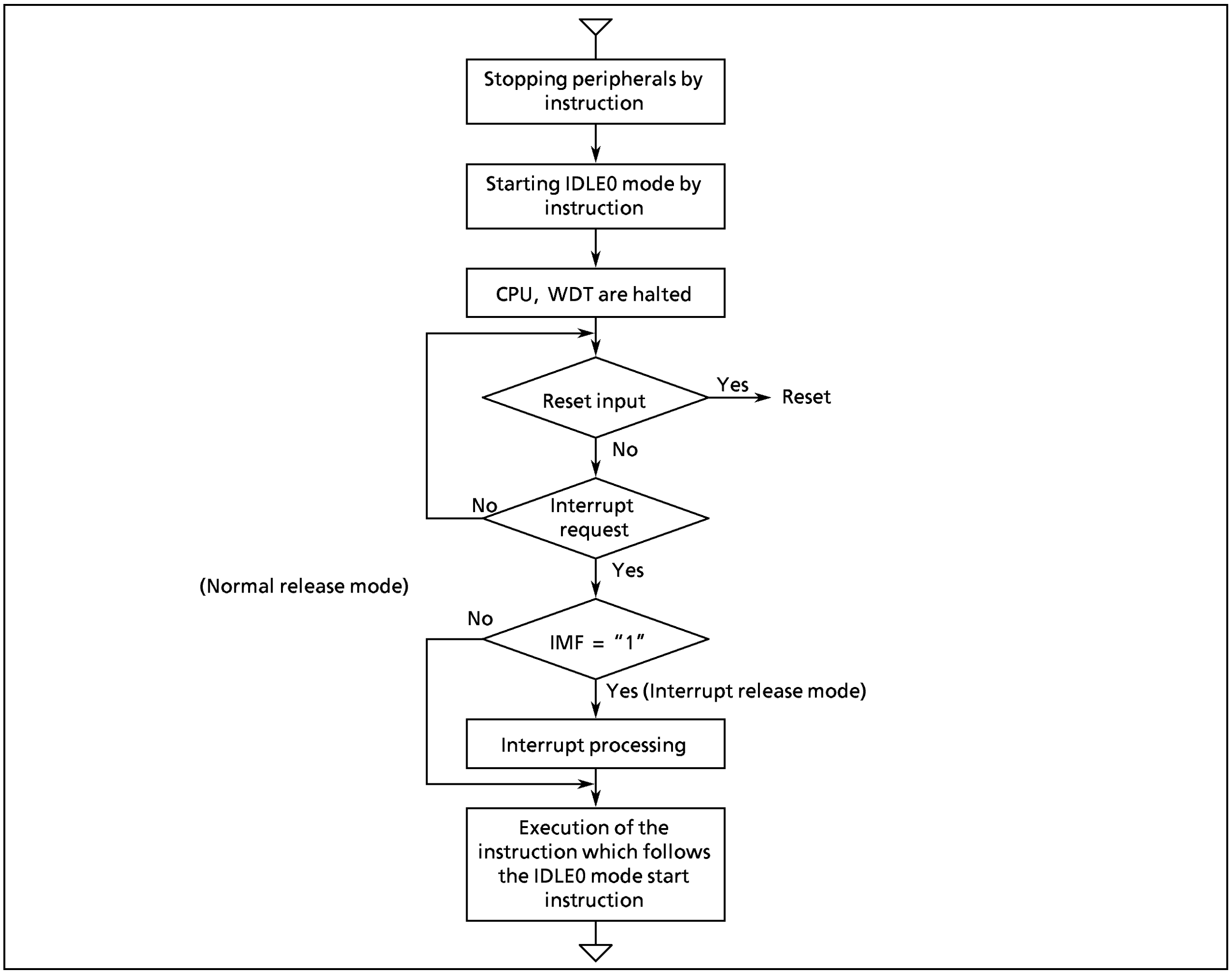

Stopping peripherals by instruction
Starting IDLE0 mode by instruction
CPU, WDT are halted
Reset input
Yes
Reset
No
No
Interrupt request
Yes
(Normal release mode)
No
IMF = "1"
Yes (Interrupt release mode)
Interrupt processing
Execution of the instruction which follows the IDLE0 mode start instruction

Figure 1-13. IDLE0 mode

- Start the IDLE0 mode

  Stop (disable) peripherals such as a timer counter.
  When IDLE0 mode starts, set SYSCR2<TGHALT> to "1".

- Release the IDLE0 mode

  IDLE0 mode include a normal release mode and an interrupt release mode.
  This mode is selected by interrupt master flag (IMF), the interrupt enable register 6 of INTTBT, and IMF.
  After releasing IDLE0 mode, the SYSCR2<TGHALT> is automatically cleared to "0" and the operation mode is returned to the mode preceding IDLE0 mode. Before starting the IDLE0 mode, when the TBTCR<TBTEN> is set to "1", INTTBT interrupt latch is set to "1".
  IDLE0 mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin. After releasing reset, the operation mode is started from NORMAL1 mode.

Note: IDLE0 mode start/release without reference to TBTCR<TBTEN> setting.

a. Normal release mode (IMF·EF_6·TBTCR<TBTEN> = "0")

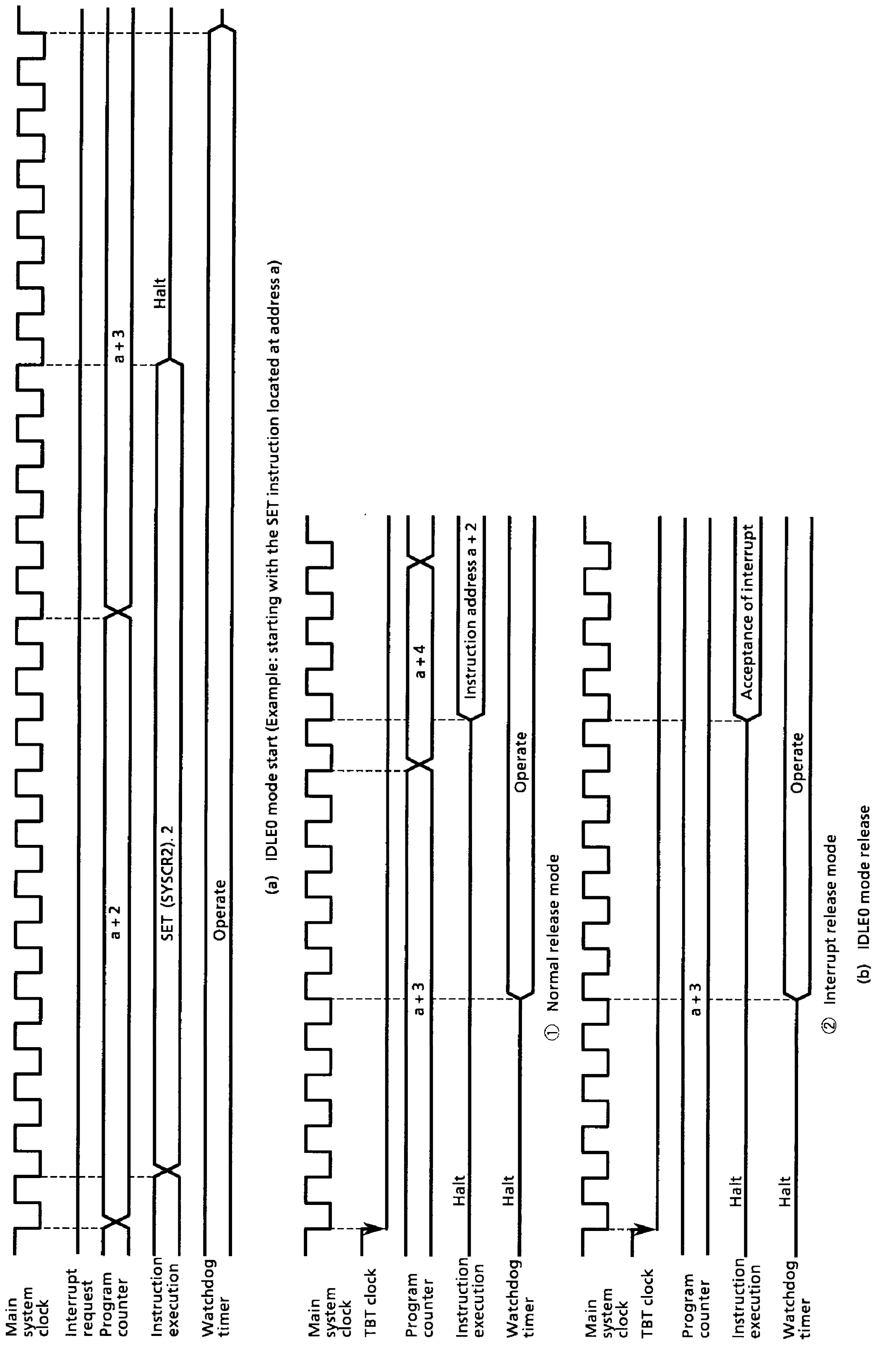
IDLE0 mode is released by the source clock falling edge, which is setting by the TBTCR<TBTCK>. After the falling edge is detected, the program operation is resumed from the instruction following the IDLE0 mode start instruction.

b. Interrupt release mode (IMF·EF_6·TBTCR<TBTEN> = "1")

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

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

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



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

① Normal release mode

② Interrupt release mode

(b) IDLE0 mode release

Figure 1-14. IDLE0 Mode Start/Release

1.5 Interrupt Control Circuit

The TMP86P202/203 are a total (Reset is excluded) of 11 interrupt sources. 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 1-2. Interrupt Sources

Interrupt Factors		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/ External	(Reset)	Non-maskable	–	FFFE_H	High 0
Internal	INTSWI (Software interrupt)	Non-maskable	–	FFFC_H	1
Internal	INTUNDEF (Executed the Undefined Instruction interrupt)	Non-maskable	–	FFFC_H	1
Internal	INTATRAP (Address Trap interrupt)	Non-maskable	IL_2	FFFA_H	2
Internal	INTWDT (Watchdog Timer interrupt)	Non-maskable	IL_3	FFF8_H	3
External	$\overline{\text{INT0}}$ (External interrupt 0)	$\text{IMF} \cdot \text{EF}_4 = 1$, $\text{INT0EN} = 1$	IL_4	FFF6_H	4
External	INT1 (External interrupt 1)	$\text{IMF} \cdot \text{EF}_5 = 1$	IL_5	FFF4_H	5
Internal	INTTBT (Time Base Timer interrupt)	$\text{IMF} \cdot \text{EF}_6 = 1$	IL_6	FFF2_H	6
Reserved		$\text{IMF} \cdot \text{EF}_7 = 1$	IL_7	FFF0_H	7
Reserved		$\text{IMF} \cdot \text{EF}_8 = 1$	IL_8	FFEE_H	8
Reserved		$\text{IMF} \cdot \text{EF}_9 = 1$	IL_9	FFEC_H	9
Internal	INTTC3 (TC3 interrupt)	$\text{IMF} \cdot \text{EF}_{10} = 1$	IL_{10}	FFEA_H	10
Internal	INTTC4 (TC4 interrupt)	$\text{IMF} \cdot \text{EF}_{11} = 1$	IL_{11}	FFE8_H	11
Internal	INTADC (AD converter interrupt)	$\text{IMF} \cdot \text{EF}_{12} = 1$	IL_{12}	FFE6_H	12
Reserved		$\text{IMF} \cdot \text{EF}_{13} = 1$	IL_{13}	FFE4_H	13
Reserved		$\text{IMF} \cdot \text{EF}_{14} = 1$	IL_{14}	FFE2_H	14
External	$\overline{\text{INT5}}$ (External interrupt)	$\text{IMF} \cdot \text{EF}_{15} = 1$	IL_{15}	FFE0_H	Low 15

Note 1: To use the address trap interrupt (INTATRAP), clear WDTCR1<ATOUT> to "0" (it is set for the "reset request" after reset is cancelled). For details, see 2.4.5 Address Trap.
Note 2: To use the watchdog timer interrupt (INTWDT), clear WDTCR1<ATOUT> to "0" (it is set for the "reset request" after reset is cancelled). For details, see 2.4 Watchdog Timer.

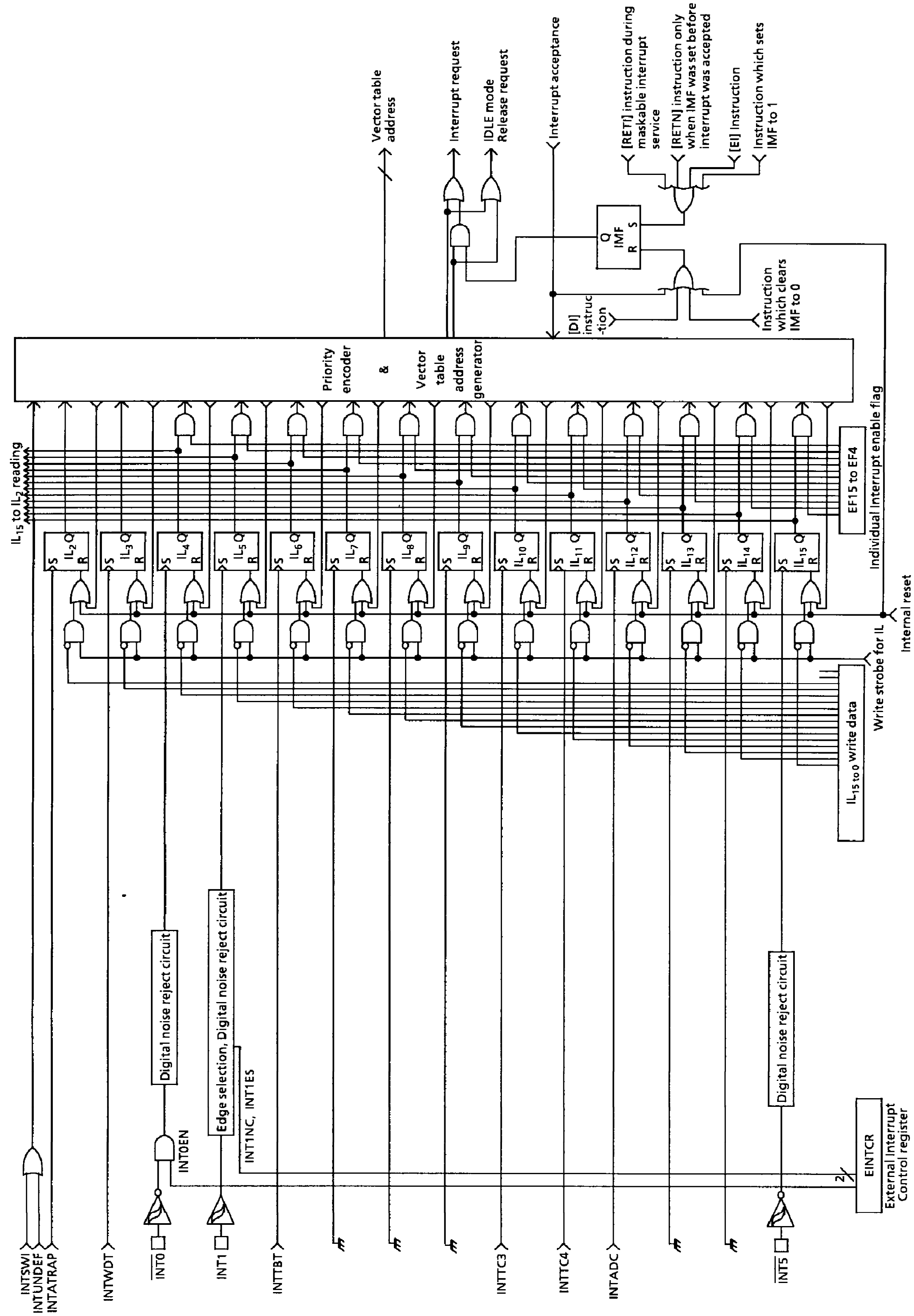

Vector table address
Interrupt request
IDLE mode Release request
Interrupt acceptance
[RETI] instruction during maskable interrupt service
[RETN] instruction only when IMF was set before interrupt was accepted
[EI] Instruction
Instruction which sets IMF to 1
Q IMF R S
[DI] instruc -tion
Instruction which clears IMF to 0
Priority encoder & Vector table address generator
IL_{15} to IL_2 reading
EF15 to EF4
Individual interrupt enable flag
S IL_2 Q R
S IL_3 Q R
S IL_4 Q R
S IL_5 Q R
S IL_6 Q R
S IL_7 Q R
S IL_8 Q R
S IL_9 Q R
S IL_{10} Q R
S IL_{11} Q R
S IL_{12} Q R
S IL_{13} Q R
S IL_{14} Q R
S IL_{15} Q R
Internal reset
Write strobe for IL
$IL_{15 to 0}$ write data
Digital noise reject circuit
Edge selection, Digital noise reject circuit
INT1NC, INT1ES
INT0EN
Digital noise reject circuit
2
EINTCR
External Interrupt Control register
INTSWI
INTUNDEF
INTATRAP
INTWDT
$\overline{INT0}$
INT1
INTTBT
INTTC3
INTTC4
INTADC
$\overline{INT5}$

Figure 1-15. 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.

Example 1: Clears interrupt latches

```
DI
LDW     (ILL), 1110001110111111B    ;  IL12 to IL10, IL6←0
LD      (ILH), 01111111B            ;  IL15←0
EI
```

Example 2: Reads interrupt latches

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

Example 3: Tests an interrupt latches

```
TEST    (ILL). 6                    ;  if IL6=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 $003A_H$ and $003B_H$ 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).

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.

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: Enables interrupts individually and sets IMF

```
DI                                ;   IMF ← 0
LDW (EIRL), 1000100000100000B     ;   EF15, EF11, EF5 ← 1
        :                             Note: IMF is not set.
        :
EI                                ;   IMF ← 1
```

Example 2: C compiler description example

```
unsigned int __io (3AH) EIRL;     ;   /* 3AH shows EIRL address */
__DI ( );
EIRL = 10100000B;
        :
__EI ( );
```

Interrupt Latches

IL ($003D_H$, $003C_H$)

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**)

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

Note: IL2 and IL3 are prohibited from clearing.

Interrupt Enable Registers

EIR ($003B_H$, $003A_H$)

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.

Note5: Do not permit EF_{14}, EF_{13}, EF_9, EF_8, and EF_7.

Figure 1-16. Interrupt Latch (IL) , Interrupt Enable Registers (EIR)

1.5.1 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 (4 μs at 8.0 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 1-17 shows the timing chart of interrupt acceptance processing.

(1) Interrupt acceptance processing is packaged as follows.

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

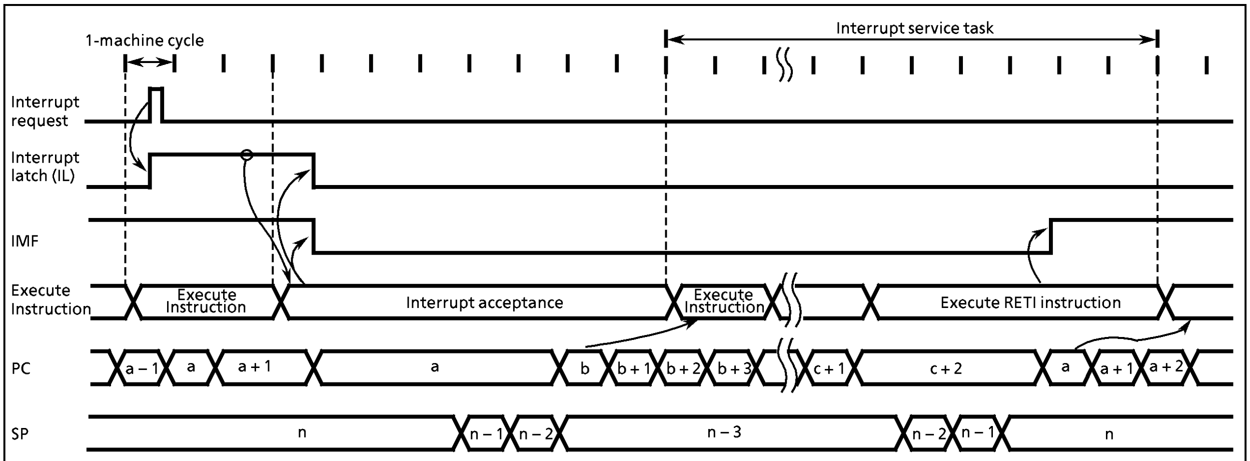
b) The interrupt latch (IL) for the interrupt source accepted is cleared to 0.

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

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

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

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

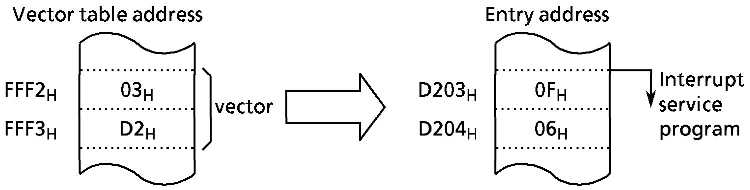


Note 1: a: Return address b: Entry address c: Address which RETI instruction is stored

Note 2: On condition that interrupt is enabled, it takes 38/fc [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 1-17. 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

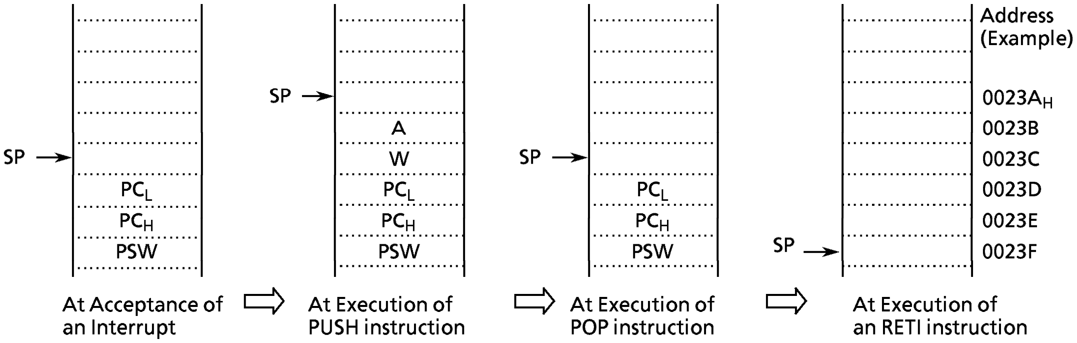
During interrupt acceptance processing, the program counter (PC) and the program status word (PSW, includes IMF) are automatically saved on the stack, but the accumulator and others are not. These registers are saved by software if necessary. When multiple interrupt services are nested, it is also necessary 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 or to save registers by a same interrupt request, 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
```



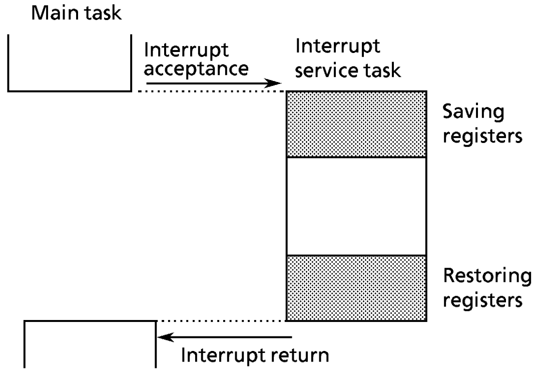
SP
PC_L
PC_H
PSW
At Acceptance of an Interrupt
SP
A
W
PC_L
PC_H
PSW
At Execution of PUSH instruction
SP
PC_L
PC_H
PSW
At Execution of POP instruction
SP
At Execution of an RETI instruction
Address (Example)
0023A_H
0023B
0023C
0023D
0023E
0023F

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 1-18. 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 times.

As for Address Trap interrupt (INTARTAP), 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.

Note: If the return instruction [RETN] is executed without rewriting the above data, the program returns to the address trap area and an address trap interrupt occurs again.
After interrupt acceptance processing has completed, the PC_L and PC_H values after returning from the interrupt are stored in addresses (SP + 1) and (SP + 2), respectively.

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

```
PINTxx  :  POP     WA                      ;  Recover SP by 2 times
           LD      WA, Return Address      ;  Store the restart address in the WA
                                              register
           PUSH    WA                      ;  Alter stacked data
           (interrupt processing)
           RETN                            ;  Non-maskable interrupt return
                                              instruction
```

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

```
PINTxx  :  INC     SP                      ;  Recover SP by 3 times
           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
```

Note: 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.

1.5.2 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.

1.5.3 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.

1.5.4 Address Trap Interrupt (INTATRAP)

If an instruction is fetched from any invalid area (address trap area) where instructions are not located, the device is reset or outputs an interrupt signal (INTATRAP). When an address trap interrupt is generated, the interrupt latch (IL2) is set and interrupt processing is entered. INTATRAP is accepted at any time even while other nonmaskable interrupts are being serviced, so that the current processing is stopped in the middle and INTATRAP interrupt processing starts immediately.
The address trap area can be chosen to be the SFR or RAM area or only the SFR area.

Note 1: Settings for address trap operation (reset operation or interrupt request) are set using the Watchdog Timer Control Register.

Note 2: If a SWI instruction located at the address immediately preceding the address trap area or an undefined instruction is executed, an address trap interrupt is accepted and serviced immediately after accepting a SWI/undefined instruction interrupt.

1.5.5 External Interrupts

The TMP86P202/203 have three external interrupt inputs. All of these interrupt inputs contain a digital noise rejection circuit (which eliminates narrow input pulses less than a predetermined duration as noise).
Also, the INT1 pins are edge-selectable.
The $\overline{\text{INT0}}$/P10 pin can be chosen to be an external interrupt input pin or an input/output port. When reset, this pin functions as an input port.
Edge selection, noise rejection control, and $\overline{\text{INT0}}$/P10 pin function selection all are set using the External Interrupt Control Register.

Table 1-3. External Interrupts

Interrupt Source	Pin Name	Shared Pin	Enable Condition	Edge	Digital Noise Rejection Circuit
INT0	$\overline{\text{INT0}}$	P10	IMF = 1, EF_4 = 1, INT0EN = 1	Falling edge	Pulses less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are always recognized as signal.
INT1	INT1	P11	IMF · EF_5 = 1	Falling or rising edge	Pulses less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of 49/fc or 193/fc [s] or more are always recognized as signal.
INT5	$\overline{\text{INT5}}$	P20/$\overline{\text{STOP}}$	IMF · EF_{15} = 1	Falling edge	Pulses less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are always recognized as signal.

*Note 1: When a noise-free signal is applied to the external interrupt input pin during **NORMAL 1** or IDLE 1 mode, the maximum amount of time from the active edge of the input signal to when the interrupt latch is set is as follows:*

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

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

Note 3: In cases where the shared pin is used as an output port and the data on it changes or the port is switched from output to input, a spurious interrupt request signal is generated. Therefore, some corrective measure such as to disable the interrupt enable flag is required.

Note 4: Be sure to disable external interrupts before changing operation modes or altering the value of the External Interrupt Control Register (EINTCR). Also, when enabling interrupts after changing operation modes or altering the value of the External Interrupt Control Register (EINTCR), be sure to clear the unnecessary interrupt latches in advance.

	7	6	5	4	3	2	1	0	
EINTCR (0037_H)	INT1NC	INT0EN					INT1ES		(Initial value: 00** **0*)

INT1NC	Selects noise rejection time for INT1	0: Eliminates pulses less than 63/fc [s] as noise 1: Eliminates pulses less than 15/fc [s] as noise	R/W
INT0EN	Selects P10/$\overline{\text{INT0}}$ function	0: P10 input/output port 1: $\overline{\text{INT0}}$ pin (P10 port must be set for input mode)	
INT1ES	Selects INT1 edge	0: Generates interrupt request at rising edge 1: Generates interrupt request at falling edge	

*Note: fc: The high-frequency clock [Hz] *: Don't care*

Figure 1-19. External Interrupt Control Register

1.6 Reset Circuit

The TMP86P202/203 have four types of reset generation procedures : an external reset input, an address trap reset, a watchdog timer reset and a system clock reset. Table 1-4 shows on-chip hardware initialization by reset action.
The malfunction reset 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 reset action at the maximum 24/fc[s] (3.0 μs at 8.0 MHz) when power is turned on.

Table 1-4. 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

1.6.1 External Reset Input

The $\overline{\text{RESET}}$ pin contains a Schmitt trigger (hysteresis) with an internal pull-up resistor.
When the $\overline{\text{RESET}}$ pin is held at L level for at least 3 machine cycles (12/fc [s]) with the power supply voltage within the operating voltage range and oscillation stable, a reset is applied and the internal state is initialized.
When the $\overline{\text{RESET}}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses FFFE_H to FFFF_H.

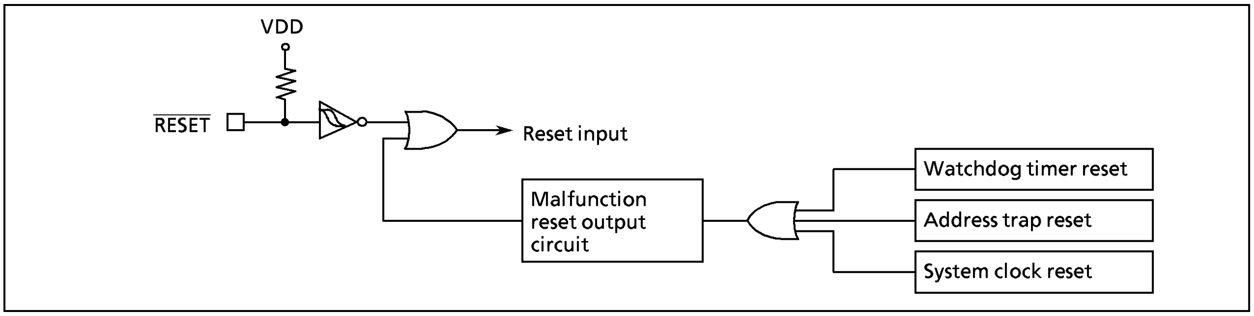
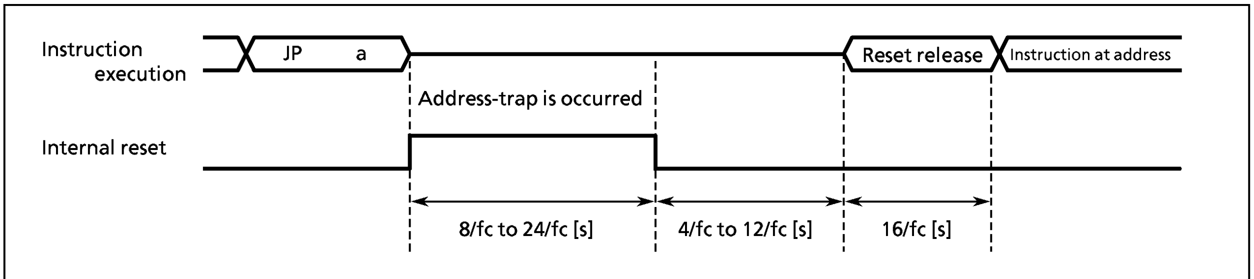


Figure 1-20. Reset Circuit

1.6.2 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. The reset time is about 8/fc to 24/fc [s] (1.0 to 3.0 μs at 8.0 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.

Figure 1-21. Address-Trap-Reset

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

1.6.3 Watchdog Timer Reset

Refer to Section "2.4 Watchdog Timer".

1.6.4 System-clock-reset

Clearing XEN (bits 7 in SYSCR2) to "0" stops system clock, and causes the microcomputer to deadlock. This can be prevented by automatically generating a reset signal whenever XEN = 0, is detected to continue the oscillation. The reset time is about 8/fc to 24/fc [s] (1.0 to 3.0 μs at 8.0 MHz).

2. On-Chip Peripherals Functions

2.1 Special Function Register (SFR)

The TMP86P202/203 adopt the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function register (SFR). The SFR is mapped on address 0000_H to $003F_H$.

Figure 2-1 indicate the special function register (SFR) for TMP86P202/203.

Address	Read	Write	Address	Read	Write
0000_H	P0DR (P0 Port Output Latch Control)		0020_H	ADCDR1 (AD Conversion Value Register 1)	–
01	P1DR (P1 Port Output Latch Control)		21	ADCDR2 (AD Conversion Value Register 2)	–
02	P2DR (P2 Port Output Latch Control)		22	Reserved	
03	P3DR (P3 Port Output Latch Control)		23	Reserved	
04	Reserved		24	Reserved	
05	Reserved		25	Reserved	
06	Reserved		26	Reserved	
07	Reserved		27	Reserved	
08	Reserved		28	Reserved	
09	P1CR (P1 Port Input/output Control)		29	Reserved	
0A	P3CR (P3 Port Input/output Control)		2A	Reserved	
0B	P0OUTCR (P0 Port Output Control)		2B	Reserved	
0C	P0PRD (P0 Pin Input)	–	2C	Reserved	
0D	P2PRD (P2 Pin Input)	–	2D	Reserved	
0E	ADCCR1 (AD Control Register 1)		2E	Reserved	
0F	ADCCR2 (AD Control Register 2)		2F	Reserved	
10	Reserved		30	Reserved	
11	Reserved		31	Reserved	
12	Reserved		32	Reserved	
13	Reserved		33	Reserved	
14	Reserved		34	–	WDTCR1 (Watchdog Timer Control 1)
15	Reserved		35	–	WDTCR2 (Watchdog Timer Control 2)
16	Reserved		36	TBTCR (TBT/TG/DVO Control)	
17	Reserved		37	EINTCR (External Interrupt Control)	
18	Reserved		38	SYSCR1 (System Control 1)	
19	Reserved		39	SYSCR2 (System Control 2)	
1A	TC3CR (Timer Counter 3 Control)		3A	EIRL (Interrupt Enable Register)	
1B	TC4CR (Timer Counter 4 Control)		3B	EIRH (Interrupt Enable Register)	
1C	TTREG3 (Timer Register 3)		3C	ILL (Interrupt Latch)	
1D	TTREG4 (Timer Register 4)		3D	ILH (Interrupt Latch)	
1E	PWREG3 (Timer Register 3)		3E	Reserved	
1F	PWREG4 (Timer Register 4)		3F	PSW (Program Status Word)	

Note 1: Do not access the reserved addresses in the program.

Note 2: – : Cannot be accessed.

Note 3: Read-modify-write instructions (bit manipulation instructions such as SET and CLR and arithmetic instructions such as AND and OR) cannot be used for the write-only registers and interrupt latches.

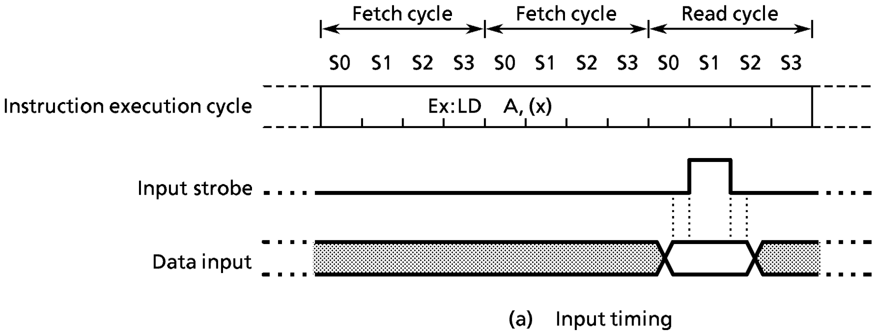
Figure 2-1. The Special Function Register (SFR) for TMP86P202/203

2.2 I/O Ports

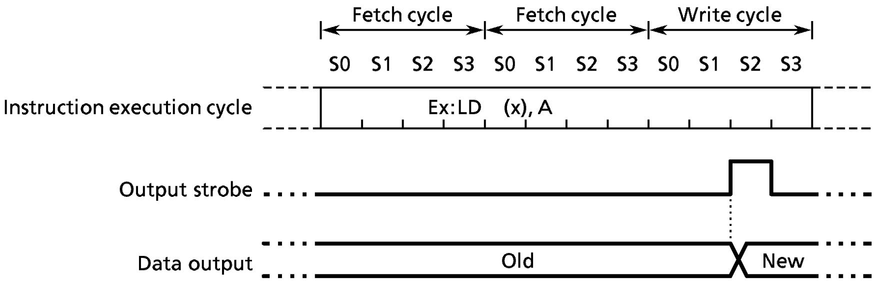
The TMP86P202/203 have 4 parallel input/output ports as follows.

	Primary Function	Secondary Functions
Port P0	2-bit I/O port	—
Port P1	3-bit I/O port	External interrupt input and divider output.
Port P2	1-bit I/O port	External interrupt input and STOP mode release signal input.
Port P3	8-bit I/O port	Analog input and Timer/Counter 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



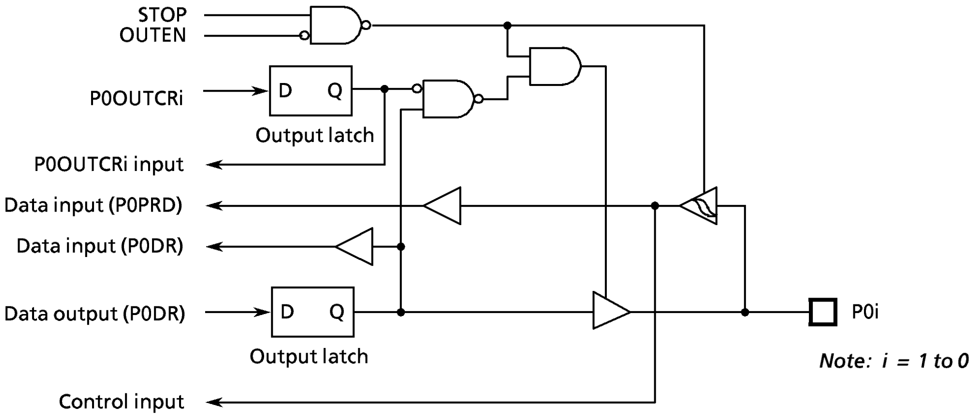
(b) Output timing

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 P0 (P01 to P00) Port (High Current)

The P0 port is an 2-bit input/output port. When using this port as an input port set the output latch to 1. When using this port as an output port, the output latch data (P0DR) is output to the P0 port.
When reset, the output latch (P0DR) and the push-pull control register (P0OUTCR) are initialized to 1 and 0, respectively.
The P0 port allows its output circuit to be selected between N-channel open-drain input/output or push-pull output by the P0OUTCR register.
When using this port as an input port, set the P0OUTCR register's corresponding bit to 0 after setting the P0DR to 1.
The P0 port has independent data input registers. To inspect the output latch status, read the P0DR register. To inspect the pin status, read the P0PRD register.



	7	6	5	4	3	2	1	0	
P0DR (0000_H) R/W							P01	P00	(Initial value: **** **11)

	7	6	5	4	3	2	1	0
P0PRD ($000C_H$) Read only							P01	P00

	7	6	5	4	3	2	1	0	
P0OUTCR ($000B_H$) R/W							P0OUTCR1	P0OUTCR0	(Initial value: **** **00)

P0OUTCR	Controls P0 port output	0: Sink open-drain input/output 1: Push-pull output	R/W

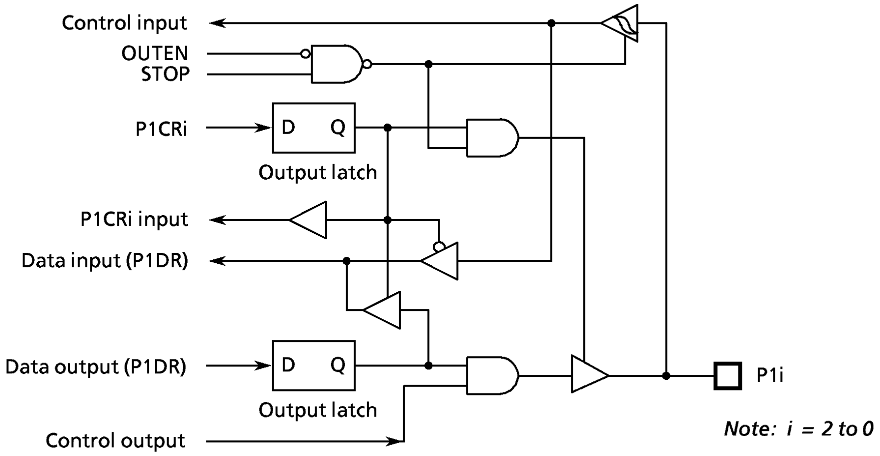
Figure 2-3. P0 Port and the P0 Port Input/output Control Register

2.2.2 P1 (P12 to P10) Port

The P1 port is a 3-bit input/output port that can be specified for input or output bitwise. The P1 Port Input/output Control Register (P1CR) is used to specify this port for input or output. When reset, the P1CR register is initialized to 0, with the P1 port set for input mode. The P1 port output latch is initialized to 0.

The P1 port is shared with external interrupt input and divider output. When using the P1 port as function pin, set its input pins for input mode. For the output pins, first set their output latches to 1 before setting the pins for output mode.

Note that the P11 pin is an external interrupt input. (When used as an output port, its interrupt latch is set at the rising or falling edge.) The P10 pin can be used as an input/output port or an external interrupt input by selecting its function with the External Interrupt Control Register (INT0EN). When reset, the P10 pin is chosen to be an input port.



	7	6	5	4	3	2	1	0	
P1DR (0001_H) R/W						P12 $\overline{\text{DVO}}$	P11 INT1	P10 $\overline{\text{INT0}}$	(Initial value: **** *000)

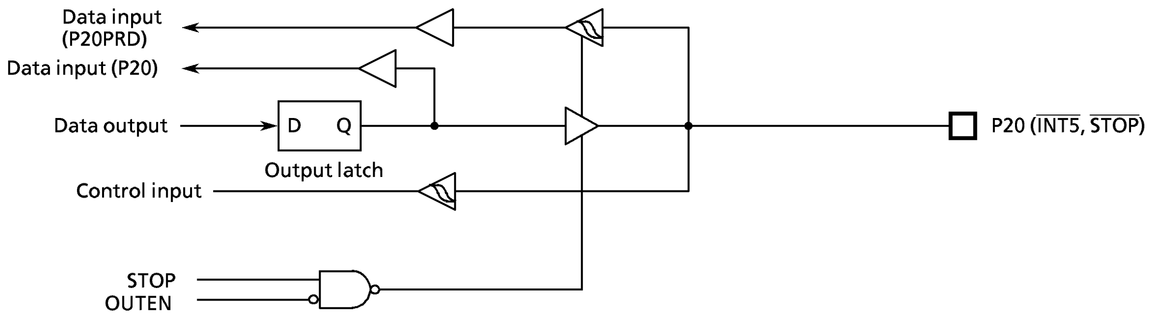
	7	6	5	4	3	2	1	0	
P1CR (0009_H)									(Initial value: **** *000)

P1CR	Controls P1 port input/output (specified bitwise)	0: Input mode 1: Output mode	R/W

Figure 2-4. P1 Port and the P1 Port Input/output Control Register

2.2.3 P2 (P20) Port

The P2 port is a 1-bit input/output port shared with external interrupt input, and STOP canceling signal input. When using this port as an input port or function pin, set the output latch to 1. The output latch is initialized to 1 when reset.
We recommend using the P20 pin for external interrupt input or STOP canceling signal input or as an input port. (When used as an output port, the interrupt latch is set by a falling edge.)
The P2 port has independent data input registers. To inspect the output latch status, read the P2DR register. To inspect the pin status, read the P2PRD register. When the P2DR or P2PRD read instruction is executed for the P2 port, the values read from bits 7 to 1 are indeterminate.



P2DR (0002$_H$) R/W	7	6	5	4	3	2	1	0	
								P20 $\overline{\text{INT5}}$ $\overline{\text{STOP}}$	(Initial value: **** ***1)

P2PRD (000D$_H$) Read only	7	6	5	4	3	2	1	0
								P20

Note: The P20 pin is shared with the $\overline{\text{STOP}}$ pin, so that when in STOP mode, its output goes to a High-Z state regardless of the OUTEN status.

Figure 2-5. P2 Port and the P2 Port Input/output Control Register

2.2.4 P3 (P37 to P30) Port

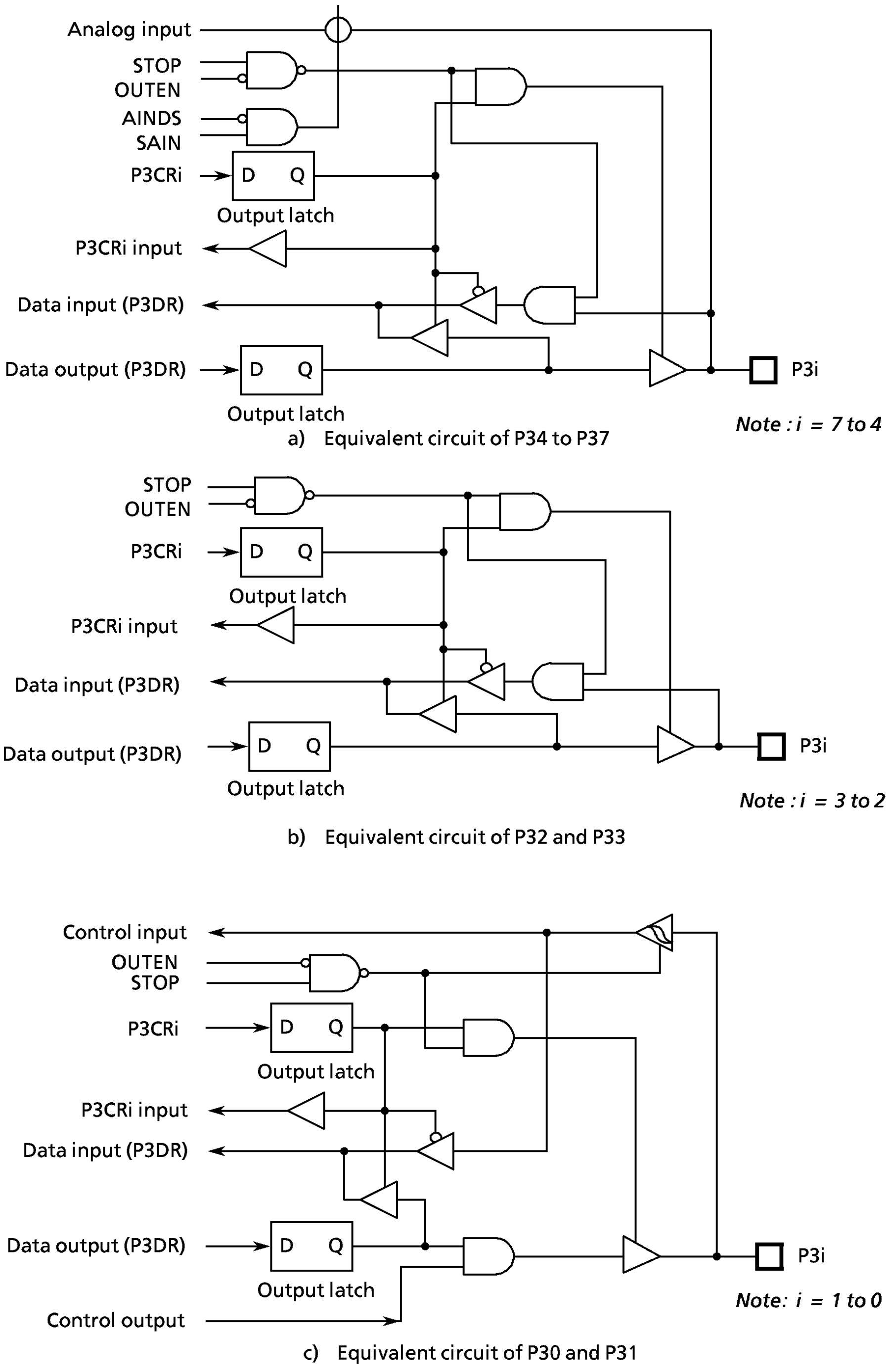
The P3 port is an 8-bit input/output port that can be specified for input or output bitwise, and is shared with analog input and 8-bit timer counter input/output. The P3 Port Input/output Control Register (P3CR) and AINDS (ADCCR1 register bit 4) are used to specify this port for input or output. When reset, the P3CR register and P3DR are cleared to 0, while AINDS is set to 1, so that P37 to P30 function as input port.

When using the P3 port as an input port, set AINDS = 1 while at the same time setting the P3CR register to 0.

When using the P3 port for analog input, set AINDS = 0 and the pins selected with SAIN (ADCCR1 register bits 3 to 0) are set for analog input no matter what values are set in the P3DR and P3CR. When using the P3 port as an output port, set the P3CR to 1 and the pin associated with that bit is set for output mode, so that P3DR (output latch data) is output from that pin.

When an input instruction is executed for the P3 port while using the AD converter, the pins selected for analog input read in the P3DR value into the internal circuit and those not selected for analog input read in a 1 or 0 according to the logic level on each pin. Even when an output instruction is executed, no latch data are forwarded to the pins selected for analog input.

Any pins of the P3 port which are not used for analog input can be used as input/output ports. During AD conversion, however, avoid executing output instructions on these ports, because this is necessary to maintain the accuracy of conversion. Also, during AD conversion, take care not to enter a rapidly changing signal to any port adjacent to analog input.



a) Equivalent circuit of P34 to P37

Note : i = 7 to 4

b) Equivalent circuit of P32 and P33

Note : i = 3 to 2

c) Equivalent circuit of P30 and P31

Note: i = 1 to 0

Figure 2-6. P3 Port and the P3 Port Input/output Control Register (1/2)

P3DR (0003_H) R/W

7	6	5	4	3	2	1	0
P37 AIN5	P36 AIN4	P35 AIN3	P34 AIN2	P33	P32	P31 TC4 $\overline{\text{PDO4}}$ $\overline{\text{PWM4}}$ $\overline{\text{PPG4}}$	P30 TC3 $\overline{\text{PDO3}}$ $\overline{\text{PWM3}}$

(Initial value: 0000 0000)

P3CR ($000A_H$)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

P3CR	Controls P3 port output (specified bitwise)	0: Input mode 1: Output mode	R/W

Note 1: P30 and P31 are hysteresis inputs.

Note 2: Input status on ports set for input mode are read in into the internal circuit. Therefore, when using the ports in a mixture of input and output modes, the contents of the output latches for the ports that are set for input mode may be rewritten by execution of bit manipulating instructions.

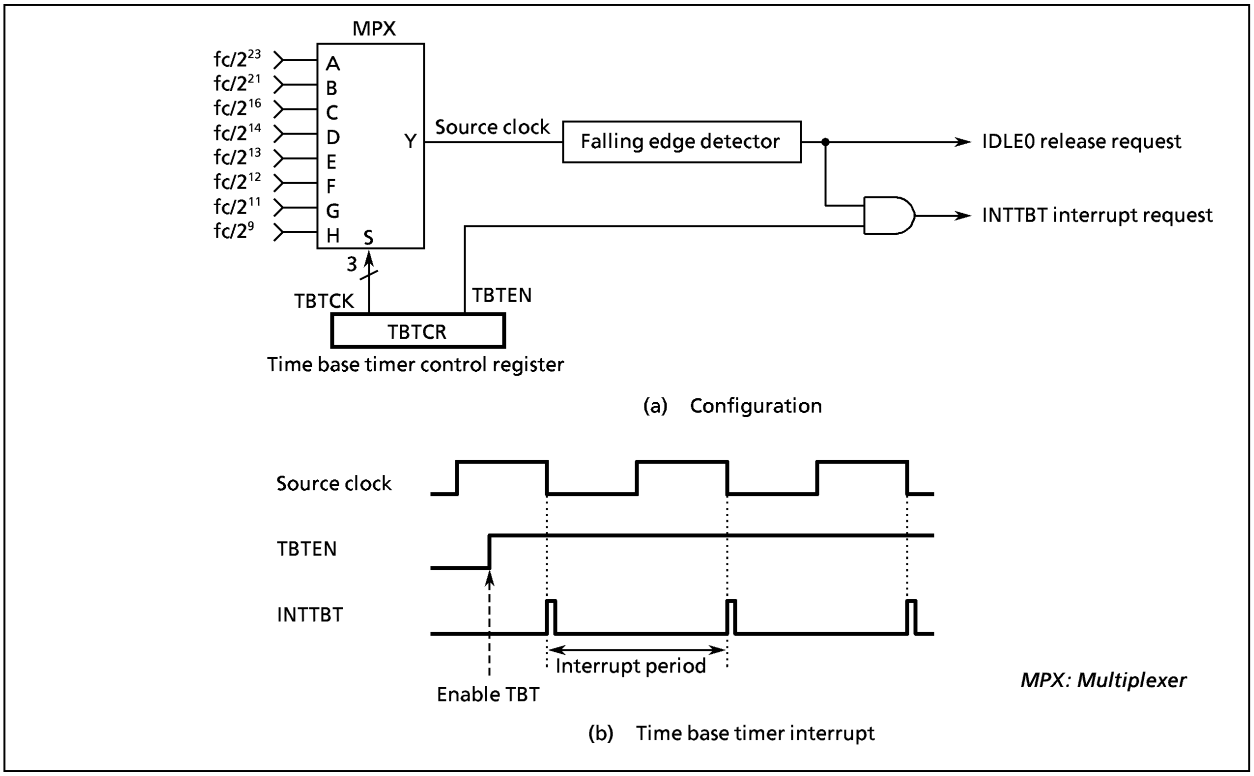
Figure 2-6. P3 Port and the P3 Port Input/output Control Registe (2/2)

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 selected by TBTCK) 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 2-7.(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.



(a) Configuration

(b) Time base timer interrupt

Figure 2-7. 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                              ; IMF ← “0”
SET   (EIRL). 6
EI                              ; IMF ← “1”
```

TBTCR
(0036$_H$)

7	6	5	4	3	2	1	0
(DVOEN)	(DVOCK)		“0”	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]	000: $fc/2^{23}$ 001: $fc/2^{21}$ 010: $fc/2^{16}$ 011: $fc/2^{14}$ 100: $fc/2^{13}$ 101: $fc/2^{12}$ 110: $fc/2^{11}$ 111: $fc/2^{9}$	

Note 1: fc : High-frequency clock [Hz] , ∗ : Don’t car
Note 2: Make sure to write ”0” to bit 4 in TBTCR.

Figure 2-8. Time Base Timer Control Register

Table 2-2. Time Base Timer Interrupt Frequency (Example: fc = 8.0 MHz)

TBTCK	Time base timer interrupt frequency [Hz]
000	0.95
001	3.81
010	122.07
011	488.28
100	976.56
101	1953.12
110	3906.25
111	15625

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 action or a non-maskable interrupt request. However, selection is possible only once after reset. After releasing a reset, the reset action 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: Care must be given in system design so as to protect the watchdog timer from disturbing noise. Otherwise the watchdog timer may not fully exhibit its functionality.

2.4.1 Watchdog Timer Configuration

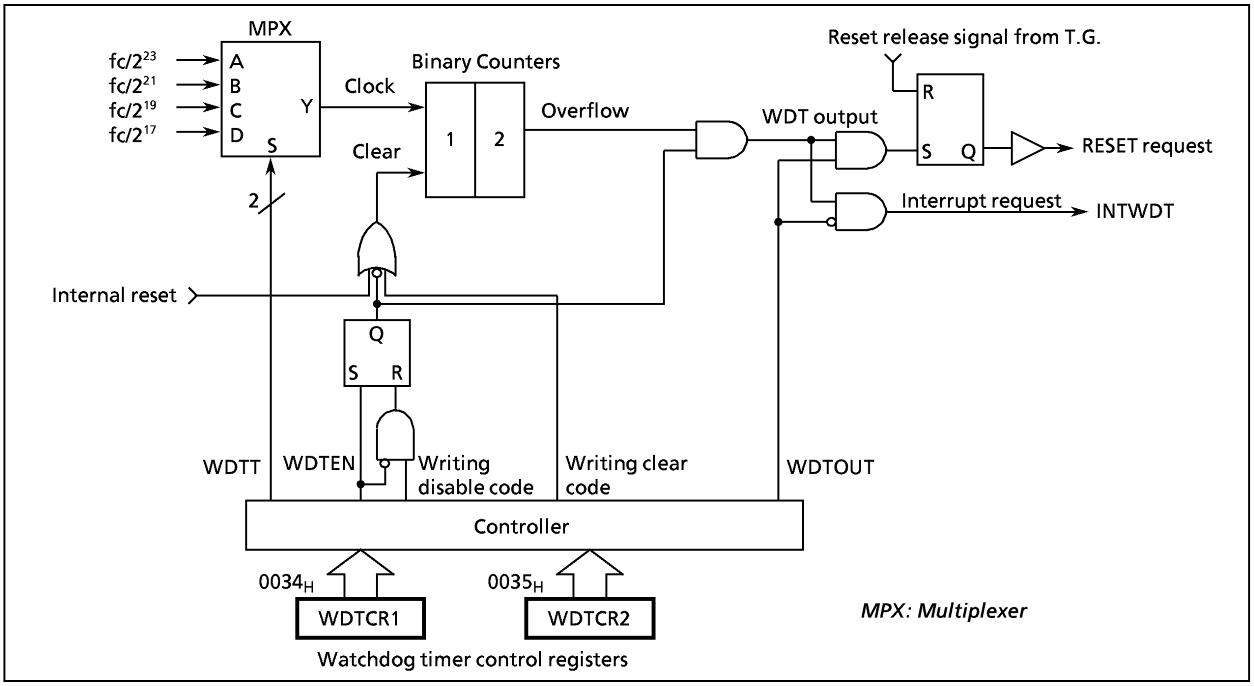


Figure 2-9. Watchdog Timer Configuration

2.4.2 Watchdog Timer Control

Figure 2-10 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

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, the internal hardware and the external circuit are reset. 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.

> *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>.*

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 after
detection time         │   :                         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

WDTCR1 (0034_H)

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

WDTEN	Watchdog timer enable/disable	0: Disable (It is necessary to write the disable code to WDTCR2) 1: Enable	Write-only
WDTT	Watchdog timer detection time [s]	00: 2^{25}/fc 01: 2^{23}/fc 10: 2^{21}/fc 11: 2^{19}/fc	
WDTOUT	Watchdog timer output select	0: Interrupt request 1: Reset action	

Note 1: WDTOUT cannot be set to "1" by program after clearing WDTOUT to "0".
*Note 2: fc: High-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: To disable the watchdog timer, always write "4EH" (clear code) to WDTCR2 for clearing the binary counter before writing "0" to WDTEN, and then write "B1H" (disable code) to WDTCR2. Also, immediately before these procedure, disable the interrupt mater flag (IMF) by DI instruction.

Watchdog Timer Register 2

WDTCR2 (0035_H)

7	6	5	4	3	2	1	0	
								(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) $D2_H$: 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

Figure 2-10. 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

Before disabling the watchdog timer, clear the interrupt master enable flag (IMF) to 0, write the clear code ($4E_H$) to WDTCR2, and then clear WDTEN (Bit 3 in WDTCR1) to 0. The watchdog timer can then be disabled by writing the disable code ($B1_H$) to WDTCR2. The watchdog timer cannot be disabled if the disable code is written to WDTCR2 before clearing WDTEN to 0. While the watchdog timer is disabled, the binary counters are cleared to 0.

Example: Disables the watchdog timer

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

Table 2-2. Watchdog Timer Detection Time (Example: fc = 8.0 MHz)

WDTT	Watchdog timer detection time [s]
00	4.194
01	1.048
10	262.144 m
11	65.536 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, 023FH              ;  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 to reset the internal hardware. The reset output time is about 8/fc to 24/fc [s] (1.0 to 3.0 μs at fc = 8.0 MHz).

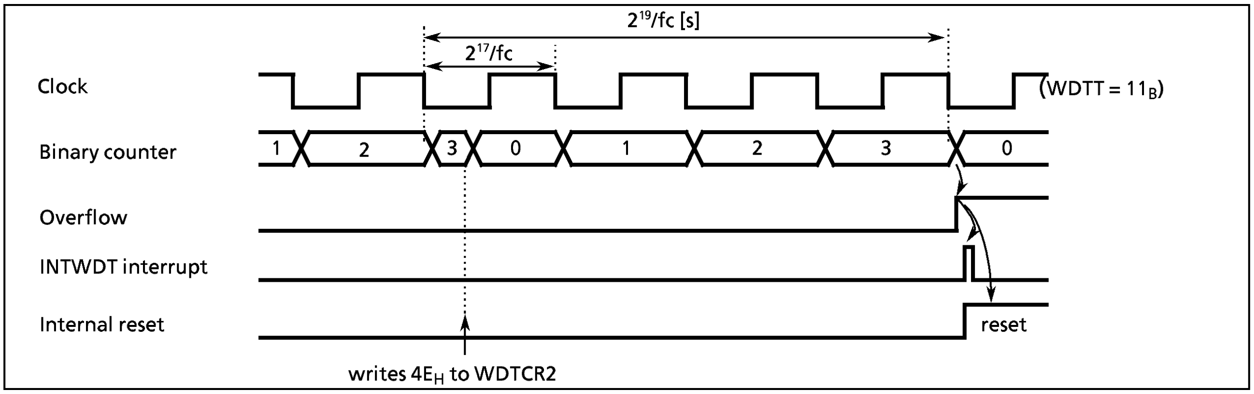


Figure 2-11. Watchdog Timer Interrupt/Reset

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 2-12. 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-action 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	(WDTEN)	(WDTT)		(WDTOUT)

(Initial value: **11 1001)

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

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

Figure 2-12. Watchdog Timer Control Registers

2.5 Divider Output (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 P12 ($\overline{\text{DVO}}$). The P12 output latch should be set to 1.

Note: Selection of divider output frequency must be made while divider output is disabled.

TBTCR (0036$_H$)

	7	6	5	4	3	2	1	0	
TBTCR (0036$_H$)	DVOEN	DVOCK		"0"	(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]	00: fc/2^{13} 01: fc/2^{12} 10: fc/2^{11} 11: fc/2^{10}	

*Note1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care*
Note2: Make sure to write "0" to bit 4 in TBTCR.

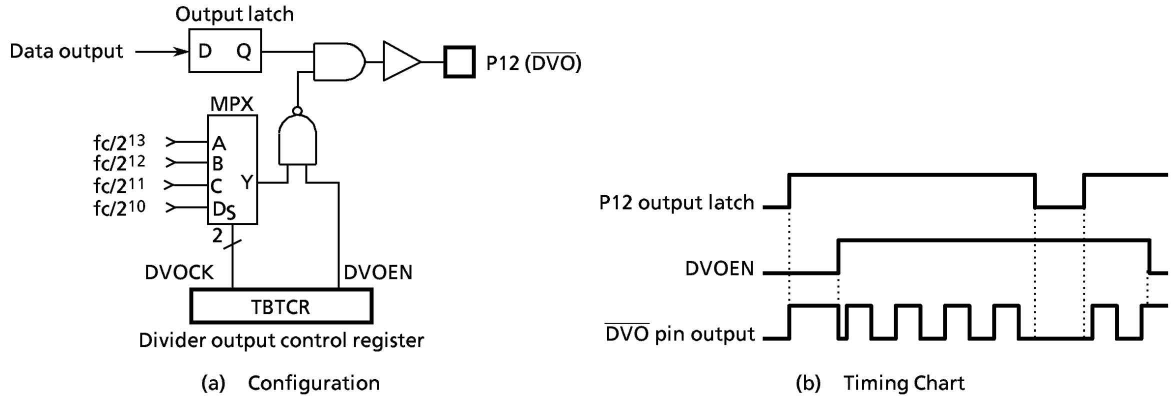
Figure 2-13. Divider Output Control Register

Example: 0.977 kHz pulse output (at fc = 8.0 MHz)

```
SET   (P1DR). 2               ;   P12 output latch←1
SET   (P1CR). 2               ;   Set P12 for output mode
LD    (TBTCR), 10000000B      ;   DVOEN←1, DVOCK←00
```

Table 2-3. Divider Output Frequency (Example: at fc = 8.0 MHz)

DVOCK	Divider output frequency [Hz]
00	0.977 k
01	1.953 k
10	3.906 k
11	7.813 k



(a) Configuration

(b) Timing Chart

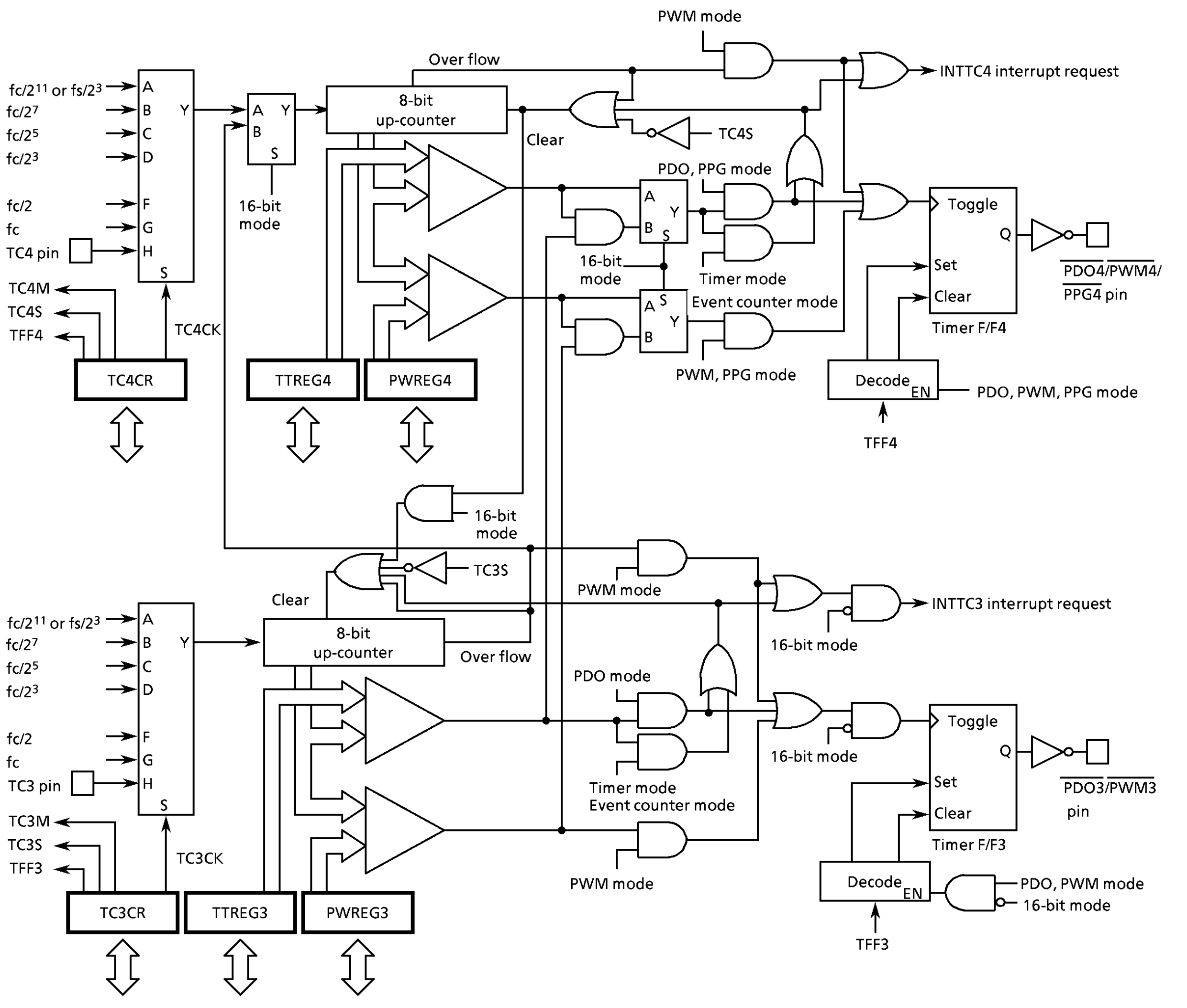
MPX: Multiplexer

Figure 2-14. Divider Output

2.6 8-Bit Timer/Counter (TC3, 4)

The TMP86P202/203 have two channels of 8-bit timer/counter (TC3, 4). These timer/counter are used as timer, event counter, PWM, PPG and PDO. These are also available as a 16-bit timer/counter by cascade connection.

2.6.1 Configuration



Note: No control input/output may be enabled depending on how the I/O ports are configured. See the chapter on the I/O ports for details.

Figure 2-15. 8-Bit Timer 3, 4

2.6.2 Control

The timer/counter 3 is controlled by a timer/counter 3 control register (TC3CR) and two 8-bit timer registers (TTREG3 and PWREG3).

Timer Register

	7	6	5	4	3	2	1	0	
TTREG3 ($001C_H$) R/W									(Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
PWREG3 ($001E_H$) R/W									(Initial value: 1111 1111)

Note 1: Do not change the timer register (TTREG3) while timer/counter is operating.
Note 2: Do not change the timer register (PWREG3) while timer/counter is operating, except 8-bit PWM and 16-bit PWM mode.

Timer/Counter 3 control register

	7	6	5	4	3	2	1	0	
TC3CR ($001A_H$)	TFF3	TC3CK			TC3S	TC3M			(Initial value: 0000 0000)

TFF3	Timer F/F3 contorl	0: Clear 1: Set	R/W
TC3CK	TC3 source clock select [Hz]	000: $fc/2^{11}$ 001: $fc/2^7$ 010: $fc/2^5$ 011: $fc/2^3$ 100: Reserved 101: fc/2 110: fc 111: External clock (TC3 pin input)	
TC3S	TC3 start control	0: Stop and counter clear 1: Command start	
TC3M	TC3 operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) mode 011: 16-bit mode (Mode selelction is controlled by TC4M) 1**: Reserved	

Note 1: fc: High-frequency clock[Hz]
Note 2: During TC3 operation, do not change TC3M, TC3CK and TFF3.
Note 3: When TC3 operation is stopped (TC3S = "1" → "0"), do not change TC3M, TC3CK and TFF3. But it is possible to change TC3M, TC3CK and TFF3 at the start timing (TC3S = "0" → "1").
Note 4: When used as 16-bit mode,the operating mode is selected by TC4M (bit 2 to 0 in TC4CR), and TC3M (bit 2 to 0 in TC3CR) should be set to "011".
Note 5: When used as 16-bit mode, only the source clock is selected by TC3CK, and start of operation and control of F/F are controlled by TC4S and TFF4 (bit 3 and 7 in TC4CR).
Note 6: Selecting source clock depends on the operating mode, refer to Table 2-4 for details.
Note 7: Value of timer register depends on the operating mode, refer to Table 2-5 for details.

Figure 2-16. Timer 3 Register and Timer/Counter 3 Control Register

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

Timer Register

	7	6	5	4	3	2	1	0	
TTREG4 ($001D_H$) R/W									(Initial value: 1111 1111)

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

Note 1: Do not change the timer register (TTREG4) while timer/counter is operating.
Note 2: Do not change the timer register (PWREG4) while timer/counter is operating, except 8-bit PWM and 16-bit PWM mode.

Timer/Counter 4 control register

	7	6	5	4	3	2	1	0	
TC4CR ($001B_H$)	TFF4	TC4CK			TC4S	TC4M			(Initial value: 0000 0000)

TFF4	Timer F/F4 contorl	0: Clear 1: Set	R/W
TC4CK	TC4 source clock select [Hz]	000: $fc/2^{11}$ 001: $fc/2^{7}$ 010: $fc/2^{5}$ 011: $fc/2^{3}$ 100: Reserved 101: fc/2 110: fc 111: External clock (TC4 pin input)	
TC4S	TC4 start control	0: Stop 1: Command start	
TC4M	TC4 operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) mode 011: Reserved 100: 16-bit timer/event counter mode 101: Reserved 110: 16-bit pulse width modulation (PWM) mode 111: 16-bit programmable pulse generate (PPG) output mode	

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

Note 2: During TC4 operation, do not change TC4M, TC4CK and TFF4.

Note 3: When TC4 operation is stopped (TC4S = "1" → "0"), do not change TC4M, TC4CK and TFF4. But it is possible to change TC4M, TC4CK and TFF4 at the start timing (TC4S = "0" → "1").

*Note 4: When TC4M is selected to "1**" (16-bit mode), the source clock is automatically selected to the overflowing signal of TC3 counter.*

Note 5: When used as 16-bit mode,the operating mode is selected by TC4M (bit 2 to 0 in TC4CR), and TC3M (bit 2 to 0 in TC3CR) should be set to "011".

Note 6: When used as 16-bit mode, only the source clock is selected by TC3CK, and start of operation and control of F/F are controlled by TC4S and TFF4 (bit 3 and 7 in TC4CR).

Note 7: Selecting source clock depends on the operating mode, refer to Table 2-4 for details.

Note 8: Value of timer register depends on the operating mode, refer to Table 2-5 for details.

Figure 2-17. Timer 4 Register and Timer/Counter 4 Control Register

Table 2-4. Operating Mode and Available Source Clock (NORMAL1, IDLE1 mode)

Operating Mode	fc/2^{11}	fc/2^7	fc/2^5	fc/2^3	fc/2	fc	TCi pin input
8-Bit Timer	○	○	○	○	–	–	–
8-Bit Event Counter	–	–	–	–	–	–	○
8-Bit PDO	○	○	○	○	–	–	–
8-Bit PWM	○	○	○	○	○	○	–
16-Bit Timer	○	○	○	○	–	–	–
16-Bit Event Counter	–	–	–	–	–	–	○
16-Bit PWM	○	○	○	○	○	○	○
16-Bit PPG	○	○	○	○	–	–	○

Note1: For 16-bit operation (16-bit Timer/Event Counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bits (TC3CK).

Note2: i = 3, 4 (8-bit mode)
i = 3 (16-bit mode)

Table 2-5. Restriction against the Rate for Comparing Registers

Operating Mode	Authorized Rate for Register
8-Bit Timer/Event Counter	1 ≦ (TTREGj) ≦ 255
8-Bit PDO	1 ≦ (TTREGj) ≦ 255
8-Bit PWM	2 ≦ (PWREGj) ≦ 254
16-Bit Timer/Event Counter	1 ≦ (TTREGm, n) ≦ 65535
16-Bit PWM	2 ≦ (PWREGm, n) ≦ 65534
16-Bit PPG	1 ≦ (PWREGm, n) < (TTREGm, n) ≦ 65535 ((PWREGm, n) + 1 < (TTREGm, n))

Note : j = 3, 4 m = 4, n = 3

2.6.3 Function

Timer/Counter 3 and 4 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 3 and 4 by cascade connection.

(1) 8-Bit Timer Mode (Timer/Counter 3 and 4)

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 request 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 TCjCR<TFFj> to “0”. If TFFj is set to “1”, unexpected pulse may be output from $\overline{PDOj}$ / $\overline{PWMj}$ / $\overline{PPGj}$ pin.
Note 2: In the timer mode, do not change the setting of timer registers (TTREGj) while timer/counter is operating. Since TTREGj is configured as one-stage register, a newly set value is immediately reflected on the timer register.
Note 3: j = 3 or 4

Table 2-6. Timer/Counter 3, 4 Source Clock (Internal Clock)

Source clock	Resolution	Maximum setting time
	fc = 8 MHz	fc = 8 MHz
$fc/2^{11}$ [Hz]	256 μs	65.2 ms
$fc/2^{7}$	16 μs	4.1 ms
$fc/2^{5}$	4 μs	1.0 ms
$fc/2^{3}$	1 μs	256.0 μs

Example: Sets the timer mode with source clock $fc/2^7$ [Hz] and generates an interrupt 160 μs later (at fc = 8 MHz).

```
LD    (TTREG4), 0AH        ;  Sets the timer register (160 μs÷2^7/fc=0AH)
DI
SET   (EIRH). 3            ;  Enables INTTC4 interrupt
EI
LD    (TC4CR), 00010000B   ;  Sets operating clock to fc/2^7, 8-bit timer mode.
LD    (TC4CR), 00011000B   ;  Starts TC4
```

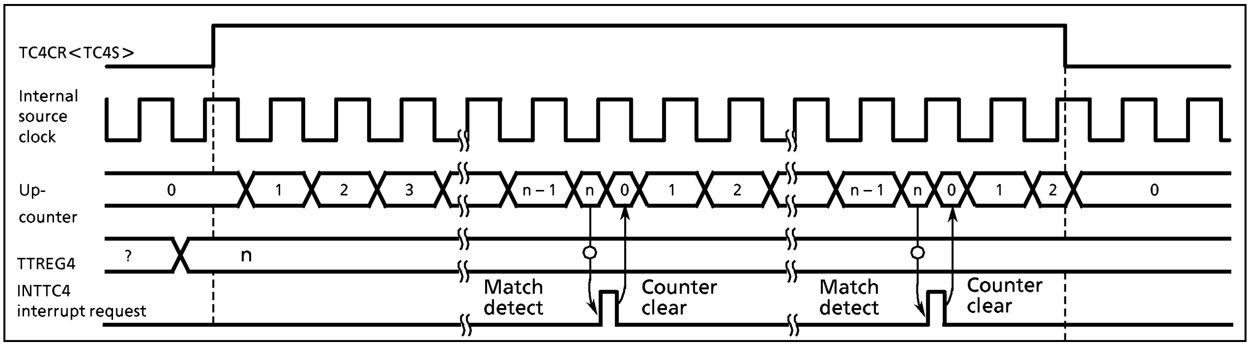


Figure 2-18. 8-Bit Timer Mode Timing Chart (In Case of Timer/Counter 4)

(2) 8-Bit Event Counter Mode (Timer/Counter 3 and 4)

In this mode, events are counted on the falling edge of TCj pin input. The contents of TTREGj are compared with the contents of up-counter. If a match is found, an INTTCj interrupt request is generated, and the counter is cleared. (The input clock width, whether high or low, requires at least two machine cycles.) The maximum applied frequency is fc/2^4 [Hz] in NORMAL1 or IDLE1 mode. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.

	NORMAL/IDLE1
"H" width	2^3/fc
"L" width	2^3/fc

Note 1: In the event counter mode, always write TCjCR<TFFj> to "0". If TFFj is set to "1", unexpected pulse may be output from $\overline{\text{PDOj}}$ / $\overline{\text{PWMj}}$ / $\overline{\text{PPGj}}$ pin.
Note 2: In the event counter mode, do not change the setting of timer registers (TTREGj) while timer/counter is operating. Since TTREGj is configured as one-stage register, a newly set value is immediately reflected on the timer register.
Note 3: j = 3 or 4

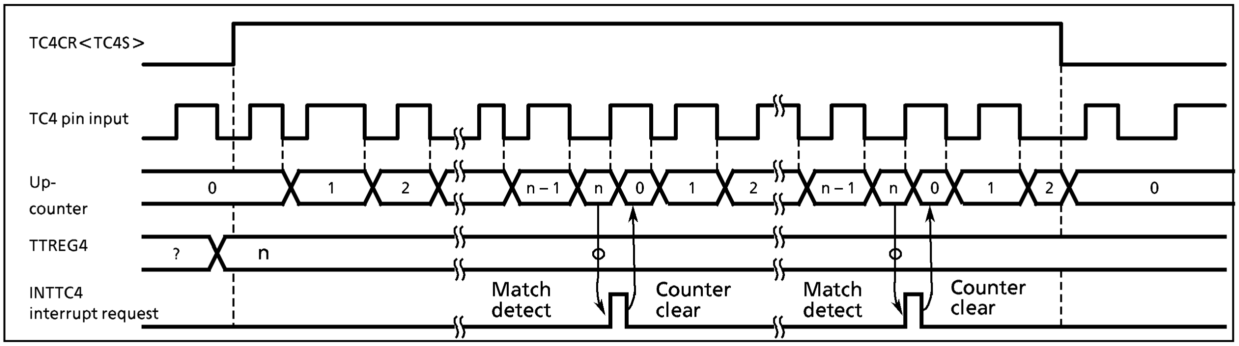


Figure 2-19. Event Counter Mode Timing Chart (In Case of Timer/Counter 4)

(3) 8-Bit Programmable Divider Output (PDO) Mode (Timer/Counter 3 and 4)

The internal clock is used for counting up. The contents of TTREGj are compared with the contents of the up-counter. Timer F/Fj output is toggled and the counter is cleared each time a match is found. Timer F/Fj output is inverted and output to the $\overline{\text{PDOj}}$ 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/Fj can be initialized by program, and it is initialized to "0" during reset. An INTTCj interrupt request is generated each time the $\overline{\text{PDOj}}$ output is toggled.

Example: Output 512 Hz pulses using TC4 (at fc = 8.0 MHz)

```
LD   (TTREG4), 3DH          ;  (1/1024 ÷ 2^7/fc) ÷ 2 = 3DH
LD   (TC4CR), 00010001B     ;  Sets operating clock to fc/2^7, 8-bit PDO mode
LD   (TC4CR), 00011001B     ;  Starts TC4.
```

Note 1: In the programmable divider output(PDO) mode, do not change the setting of timer registers (TTREGj) while timer/counter is operating. Since TTREGj 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 $\overline{\text{PDOj}}$ pin, modify TCjCR<TTFj> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFj simultaneously.

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

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

Note 3: j = 3 or 4

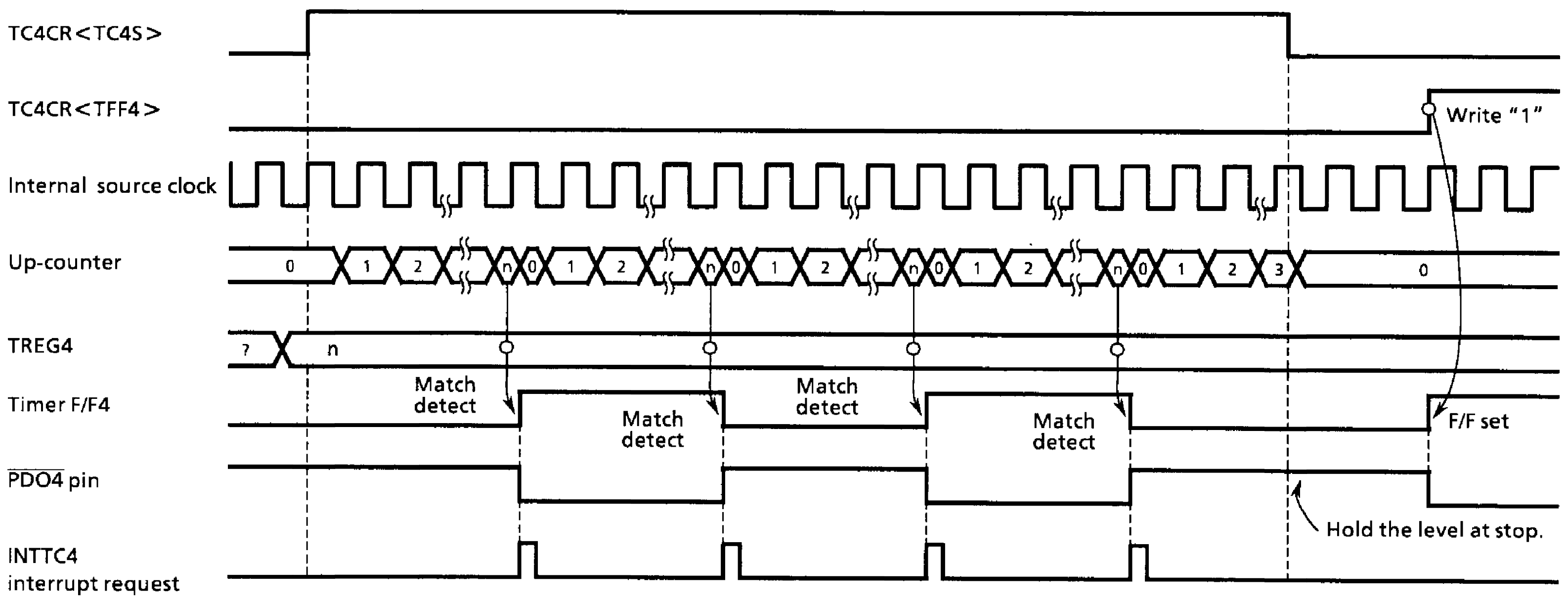


Figure 2-20. 8-Bit Timing Chart for PDO Mode (In Case of Timer/Counter 4)

(4) 8-Bit Pulse Width Modulation (PWM) Output Mode (Timer/Counter 3 and 4)

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/Fi output is toggled. The counter continues counting. And, when an overflow occurs, the timer F/Fi output is again toggled and the counter is cleared. Timer F/Fi output is inverted and output to the $\overline{\text{PWM i}}$ pin. An INTTCi interrupt request is generated when an overflow occurs.

In PWM mode, because PWREGi becomes a 2-stage registers with shift register, it is possible to change the setting value of PWREGi while timer/counter is operating. Therefore, output can be altered continuously. The shift operation of PWREGi to shift register is executed at the INTTCi timing. While timer/counter is operating, the data by read instruction is not a setting value of PWREGi but a value of shift register. Thereofre, after writing to PWREGi, the reading data of PWREGi is previous value till INTTCi is generated.

While timer/counter stops, written value to PWREGi is shifted to shift register immediately.

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 $\overline{\text{PWMi}}$, modify TCiCR<TTFi> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFi simultaneously.

Example: Fixes $\overline{\text{PWMi}}$ output at high level after timer/counter is stopped
CLR (TCiCR).3 ; Stops timer/counter.
CLR (TCiCR).7 ; Sets $\overline{\text{PWMi}}$ 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 = 3 or 4

Table 2-7. PWM Output Mode

Source clock	Resolution	Period of one cycle
	fc = 8 MHz	fc = 8 MHz
$fc/2^{11}$ [Hz]	256 μs	65.5 ms
$fc/2^{7}$	16 μs	4.1 ms
$fc/2^{5}$	4 μs	1.02 ms
$fc/2^{3}$	1 μs	256 μs
fc/2	250 ns	64 μs
fc	125 ns	32 μs

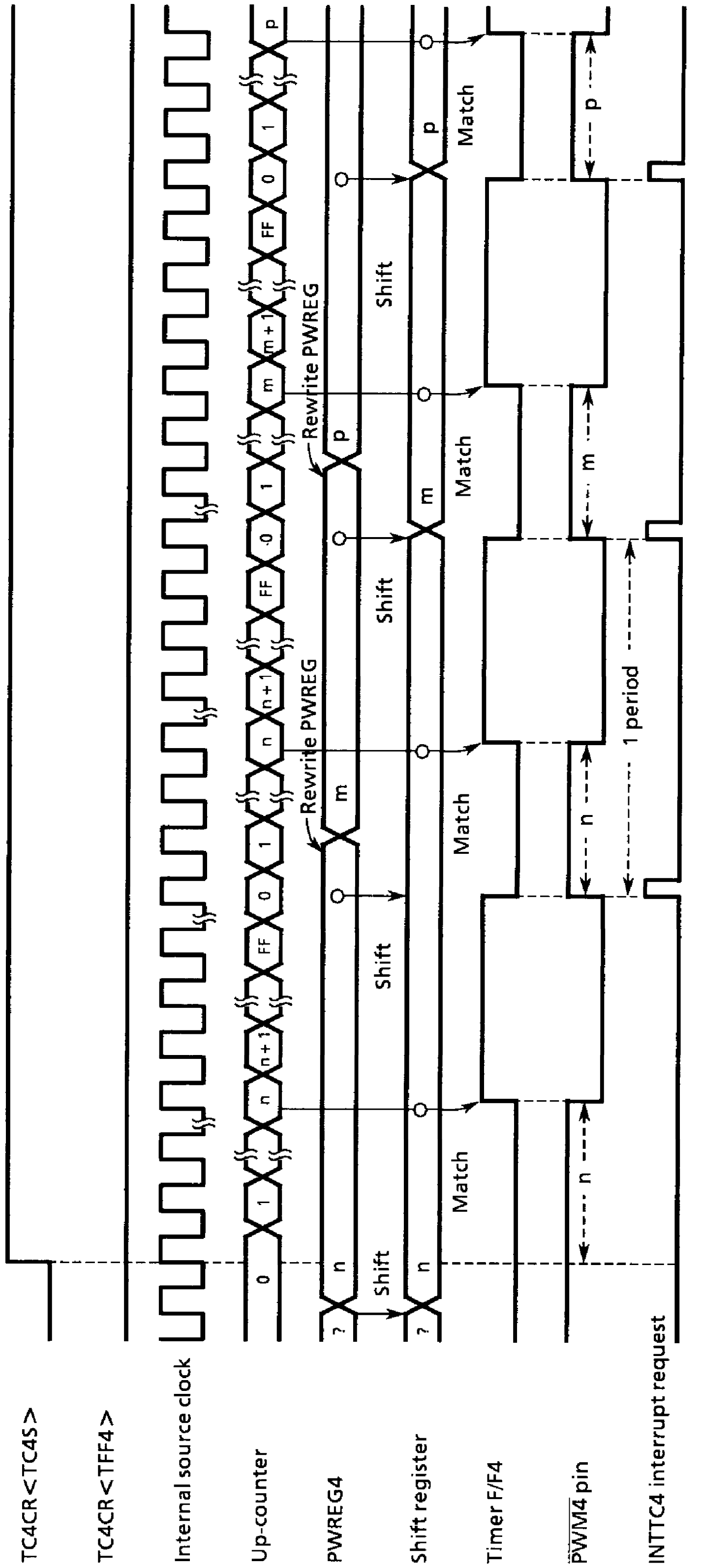


Figure 2-21. Timing Chart for PWM Mode (In Case of TC4)

(5) 16-Bit Timer Mode (Timer/Counter 3 and 4)

This is a timer mode in which the timers count up on the internal clock. The timer counters 3 and 4 can be cascaded to form a 16-bit timer.

After the timers start as specified by TC4S (TC4CR bit 3), an INTTC4 interrupt request is generated and the counter is cleared on each match between the counter value and the values of the timer registers (TTREG3 and TTREG4). Counting is continued after the counter is cleared. The timer registers must always be set in the order of the lower-order register (TTREG3) and higher-order register (TTREG4) (it is disallowed to set only the lower- or higher-order timer register).

Note 1: In the timer mode, always write TCjCR<TFFj> to "0". If TFFj is set to "1", unexpected pulse may be output from $\overline{PDOj}$/$\overline{PWMj}$/$\overline{PPGj}$ pin.
Note 2: In the timer mode, do not change the setting of timer registers (TTREGj) while timer/counter is operating. Since TTREGj is configured as one-stage register, a newly set value is immediately reflected on the timer register.
Note 3: j = 3 or 4

Table 2-8. 16-Bit Timer Mode

Source clock	Resolution	Maximum setting time
	fc = 8 MHz	fc = 8 MHz
$fc/2^{11}$	256 μs	16.78 s
$fc/2^{7}$	16 μs	1.05 s
$fc/2^{5}$	4 μs	262.1 ms
$fc/2^{3}$	1 μs	65.5 ms

Example: Set the 16-bit timer mode with source clock $fc/2^7$ [Hz] and generates an interrupt 600 [μs] later (at fc = 8 [MHz])

```
LDW    (TTREG3), 927CH    ;  Sets the timer register (600 ms ÷ 2^7/fc = 927CH)
DI
SET    (EIRH). 3          ;  Enable INTTC4 interrupt
EI
LD     (TC3CR), 13H       ;  Sets the 16-bit timer mode (lower) and source clock
LD     (TC4CR), 04H       ;  Sets the 16-bit timer mode (upper)
LD     (TC4CR), 0CH       ;  Starts timer/counter
```

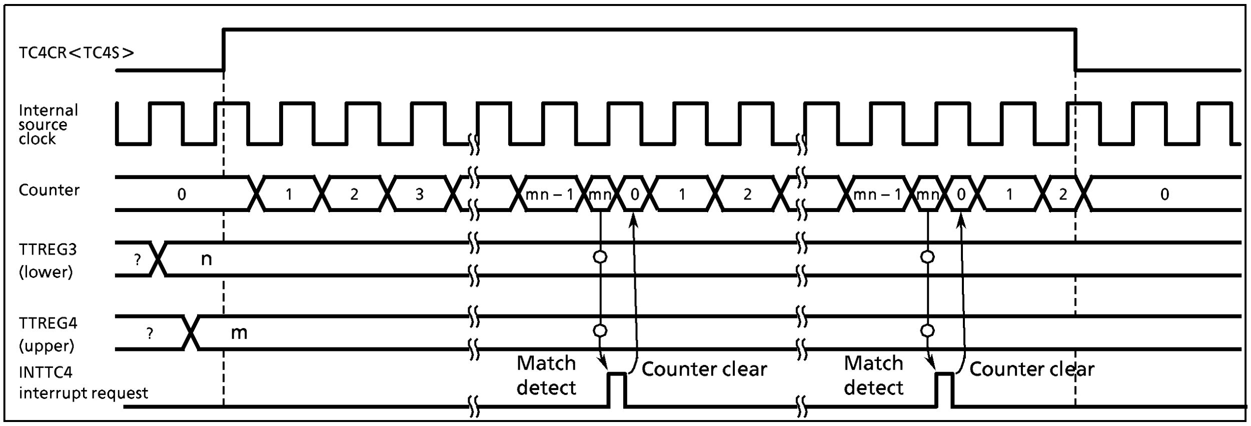


Figure 2-22. Timing Chart for 16-Bit Timer Mode (Timer/Counter 3 and 4)

(6) 16-Bit Event Counter Mode (Timer/Counter 3 and 4)

In this mode, event are counted on the falling edge of the TC3 pin input. Timer/counter 3 and 4 are also available as a 16-bit Event counter mode by cascade connection.

If a match is found, the INTTC4 interrupt request is generated and the counter is cleared to “0”. After the counter is cleared, counting up resumes every falling edge of TC3 input. The maximum applied frequency is fc/2^4 [Hz] in NORMAL1 or IDLE1 mode. Two or more machine cycles are required for both the “H” and “L” levels of the pulse width. The timer register should write to the TTREG3 more first than TTREG4. The timer register must not write only either TTREG3 or TTREG4.

	NORMAL/IDLE
“H” width	2^3/fc
“L” width	2^3/fc

The timer registers must always be set in the order of the lower-order register (TTREG3) and higher-order register (TTREG4) (it is disallowed to set only the lower- or higher-order timer register). When the timer registers are rewritten during counting, the new timer register values are reflected on the falling edge of the next input to the TC3 pin after the rewrite of the higher-order timer register (TTREG4) is completed (matches occur based on the previously set timer register values until the rewrite is completed). Initially, new timer register values are reflected when the 16-bit event counter is started by the TC4S bit after TTREG3 and TTREG4 being loaded with initial data.

Note 1: In the event counter mode, always write TCjCR<TFFj> to “0”. If TFFj is set to “1”, unexpected pulse may be output from $\overline{\text{PDOj}}$/$\overline{\text{PWMj}}$/$\overline{\text{PPGj}}$ pin.
Note 2: In the event counter mode, do not change the setting of timer registers (TTREGj) while timer/counter is operating. Since TTREGj is configured as one-stage register, a newly set value is immediately reflected on the timer register.
Note 3: j = 3 or 4

(7) 16-Bit Pulse Width Modulation (PWM) Output Mode (Timer/Counter 3 and 4)

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.

The contents of PWREG3/4 are compared with the contents of up-counter. If a match is found, the timer F/F4 output is toggled. The counter continues counting. And, when an overflow occurs, the timer F/F4 output is again toggled and the counter is cleared. Timer F/F4 output is inverted and output to the $\overline{\text{PWM4}}$ pin. An INTTC4 interrupt request is generated when an overflow occurs. When used as $\overline{\text{PWM4}}$ pin, respective output latch should be set to "1". In PWM mode, because PWREG4/3 each becomes a 2-stage registers with shift register, it is possible to change the setting value of PWREG4/3 while timer/counter is operating. Therefore, output can be altered continuously. The shift operation of PWREG4/3 to shift register is executed at the INTTC4 interrupt request timing. While timer/counter is operating, the data by read instruction is not a setting value of PWREG4/3 but a value of shift register. Therefore, after writing to PWREG4/3, the reading data of these registers is previous value till INTTC4 interrupt request is generated.
While timer/counter stops, written value to PWREG4/3 is shifted to shift register immediately. When writing to PWREG4/3, always write to the lower side (PWREG3) and then the upper side (PWREG4) in that order. Writing to only lower side (PWREG3) or the upper side (PWREG4) has no effect.

Note 1: In PWM mode, write to the timer register PWREGm,n immediately after an INTTCm interrupt is generated (normally during the INTTCm interrupt request service routine). If writing to PWREGm, n and INTTCm 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 INTTCm 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 $\overline{\text{PWMi}}$, modify TCiCR<TTFi> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFi simultaneously.

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

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

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

Note 4: m = 4, n = 3, i = 4

Table 2-9. 16-bit PWM Output Mode

Source clock	Resolution	Period of one cycle
	fc = 8 MHz	fc = 8 MHz
$fc/2^{11}$ [Hz]	256 μs	16.78 s
$fc/2^7$	16 μs	1.05 s
$fc/2^5$	4 μs	262.1 ms
$fc/2^3$	1 μs	65.5 ms
fc/2	250 ns	16.4 ms
fc	125 ns	8.2 ms

Example: Extract the pulse, whose term and "high" width is 65.536 ms and 2 ms respectively, from P31 width 16-bit PWM mode (at fc = 8 MHz = "0")

```
LDW  (PWREG3), 07D0H   ;  Sets pulse width
LD   (TC3CR), 33H      ;  Sets the 16-bit PWM mode (lower) and source clock
                          (fc/2^3)
LD   (TC4CR), 056H     ;  Sets the TFF4 to "0" and sets the 16-bit PWM mode
                          (upper)
LD   (TC4CR), 05EH     ;  Starts timer/counter
```

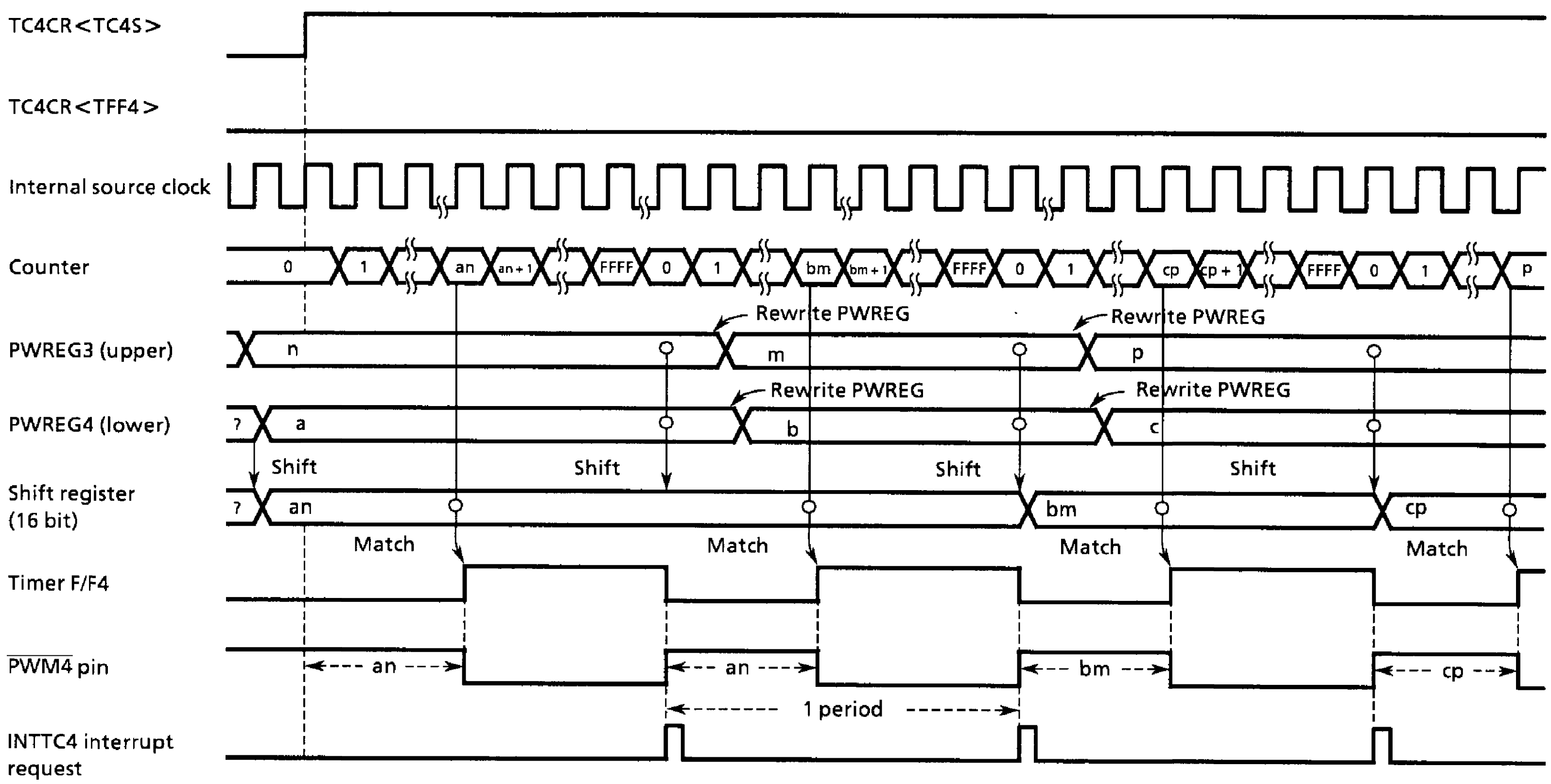


Figure 2-23. Timing Chart of 16-Bit PWM Mode (Timer/Counter 3 and 4)

(8) 16-Bit Programmable Pulse Generate (PPG) output mode (Timer/Counter 3 and 4)

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

First, the contents of PWREG3/4 are compared with the contents of up-counter. If a match is found, the timer F/F4 output is toggled. Next, timer F/F4 is again toggled and the counter is cleared by matching with TTREG3/4. The INTTC4 interrupt is generated at this time.
When used as $\overline{\text{PPG4}}$ pin, respective output latch should be set to “1”. During reset, the F/F4 is initialized to “0”.
The F/F4 output is configured by TC4CR<TFF4>. Therefore, the $\overline{\text{PPG4}}$ can output either output high or output low at first time. The timer register should write to the PWREG3/TTREG3 more first than PWREG4/TTREG4. The timer register must not write only either PWREG3/TTREG3 or PWREG4/TTREG4.

Example: Extract the pulse, whose term and “high” width is 32.770 ms and 2 ms respectively, from P31 with 16-bit PPG mode (at fc = 8 MHz)

```
LDW  (PWREG3), 07D0H    ;  Sets pulse width
LDW  (TTREG3), 8002H    ;  Sets pulse term
LD   (TC3CR), 33H       ;  Sets the 16-bit PPG mode (lower) and source clock (fc/2^3)
LD   (TC4CR), 057H      ;  Sets the TFF4 to “0” and sets the 16-bit PPG mode(upper)
LD   (TC4CR), 05FH      ;  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 $\overline{\text{PPGj}}$, modify TCiCR<TTFi> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFj simultaneously.

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

CLR (TCjCR).3 ; Stops timer/counter

CLR (TCjCR).7 ; Sets $\overline{\text{PPGj}}$ output to high level output

Note 3: j = 4, i = 3 or 4

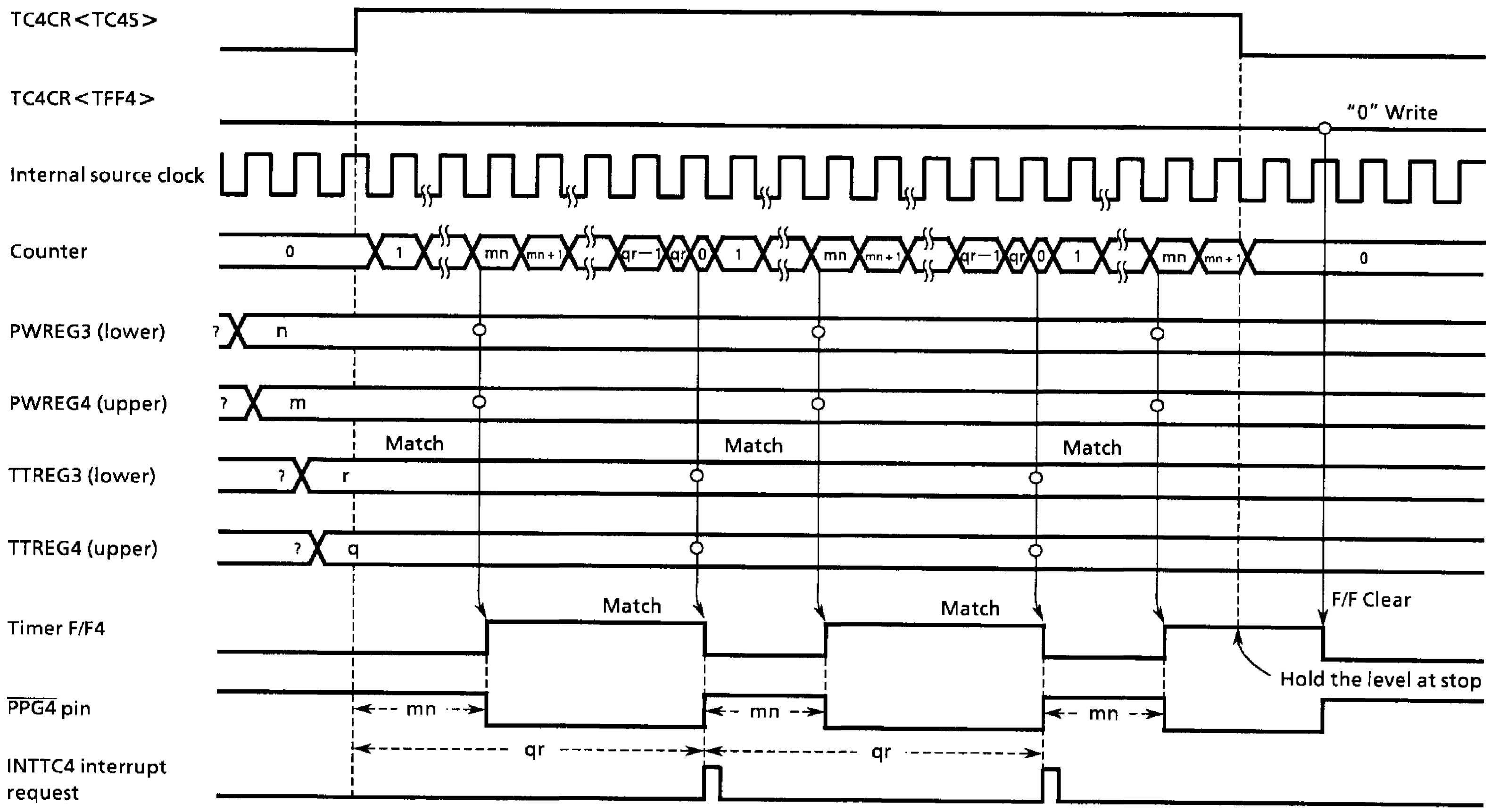


Figure 2-24. Timing Chart of 16-Bit PPG Mode (Timer/Counter 3 and 4)

2.7 8-Bit AD Converter (ADC)

The TMP86P202/203 have a 8-bit successive approximation type AD converter.

Note: AD conversion characteristics are guaranteed with limited supply voltage range (4.5 V to 5.5 V). If supply voltage is less than 4.5 V then AD conversion accuracy can not be guaranteed.

2.7.1 Configuration

The circuit configuration of the 8-bit AD converter is shown in Figure 2-25.
It consists of control registers ADCCR1 and ADCCR2, converted value registers ADCDR1 and ADCDR2, a DA converter, a sample-and-hold circuit, a comparator, and a successive comparison circuit.

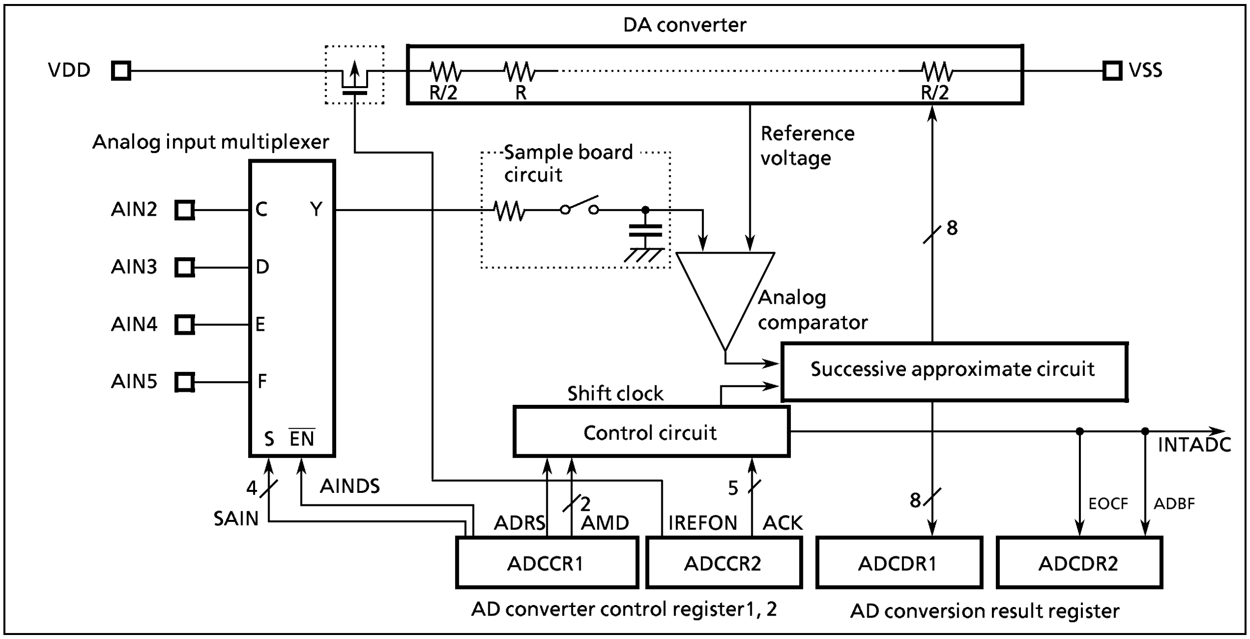


Figure 2-25. AD Converter (ADC)

2.7.2 Register Configuration

The AD converter consists of the following four registers:

- AD converter control register 1 (ADCCR1)
- AD converter control register 2 (ADCCR2)
- AD converted value register 1 (ADCDR1)
- AD converted value register 2 (ADCDR2)

(1) AD converter control register 1 (ADCCR1)

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

(2) AD converter control register 2 (ADCCR2)

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

(3) AD converted value register1 (ADCDR1)

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

(4) AD converted value register2 (ADCDR2)

This register monitors the operating status of the AD converter.

The AD converter control register configurations are shown in Figures 2-26 and 2-27.

AD Converter Control Register 1

	7	6	5	4	3	2	1	0	
ADCCR1 (000E$_H$)	ADRS	AMD		AINDS	SAIN				(Initial value: 0001 0000)

ADRS	AD conversion start	0: – 1: AD conversion start	R/W
AMD	AD operating mode select	00: Reserved 01: Software start mode 10: Reserved 11: Reserved	
AINDS	Analog input control	0: Analog input enable 1: Analog input disable	
SAIN	AD input channel select	0000: Reserved 0001: Reserved 0010: Selects AIN2 0011: Selects AIN3 0100: Selects AIN4 0101: Selects AIN5 0110: Reserved 0111: Reserved	

Note 1: Select analog input when AD converter stops.
Note 2: When the analog input is all use disabling, the AINDS should be set to 1.
Note 3: During conversion, do not perform output instruction to maintain a precision for all of the pins. And port near to analog input, do not input intense signaling of change.
Note 4: The ADRS is automatically cleared to 0 after starting conversion.
Note 5: Do not set ADRS (ADCCR1 bit 7) newly again during AD conversion. Before you set ADRS newly again, check EOCF (ADCDR bit 5) to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).
Note 6: When STOP mode is activated, AD converter control register 1 (ADCCR1) is all initialized. After NORMAL mode is resumed, .set the AD converter control register 1 (ADCCR1) again if necessary.

AD Converter Control Register 2

	7	6	5	4	3	2	1	0	
ADCCR2 (000F$_H$)			IREFON	"1"	ACK			"0"	(Initial value: **00 0000)

IREFON	Ladder resistor ON/OFF	0: ON only during AD conversion 1: Always ladder resistor ON					R/W
ACK	AD conversion time	ACK	Conversion time	fc = 8 MHz	fc = 4 MHz	fc = 2 MHz	
		000	Reserved				
		001	Reserved				
		010	78/fc [s]	–	19.5	39.0	
		011	156/fc [s]	19.5	39.0	78.0	
		100	312/fc [s]	39.0	78.0	156.0	
		101	624/fc [s]	78.0	156.0	–	
		110	1248/fc [s]	156.0	–	–	
		111	Reserved				

Note 1: Settings for – in the above table are inhibited.
Note 2: Set conversion time by analog reference voltage (V_{DD}) as follows.
V_{DD} = 4.5 to 5.5 V (15.6 μs or more)
Note 3: Always set bit 0 in ADCCR2 to 0 and set bit 4 in ADCCR2 to 1.
Note 4: When a read instruction for ADCCR2, bit 6 to 7 in ADCCR2 read in as undefined data.
Note 5: fc : High-frequency clock [Hz]
Note 6: When STOP mode is activated, AD converter control register 2 (ADCCR2) is all initialized. After NORMAL mode is resumed, set the AD converter control register 2 (ADCCR2) again if necessary.

Figure 2-26. AD Converter Control Register

2.7.3 AD Converter Operation

AD Conversion Result Register

	7	6	5	4	3	2	1	0	
ADCDR1 (0020$_H$)	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
ADCDR2 (0021$_H$)			EOCF	ADBF					(Initial value: **00 ****)

EOCF	AD conversion end flag	0: Under conversion or Before conversion 1: End of conversion	Read-only
ADBF	AD conversion busy flag	0: During stop of AD conversion 1: During AD conversion	

Note 1: The EOCF is cleared to 0 when reading the ADCDR1.
Therefore, the AD conversion result should be read to ADCDR2 more first than ADCDR1.
Note 2: ADBF is set to 1 when AD conversion starts and cleared to 0 when the AD conversion is finished. It also is cleared upon entering standby mode (STOP).
Note 3: If a read instruction is executed for ADCDR2, read data of bit 7, 6, 3 to 0 are unstable.

Figure 2-27. AD Converter Result Register

(1) Set up the AD converter control register 1 (ADCCR1) as follows:

- Choose the channel to AD convert using AD input channel select (SAIN).
- Specify analog input enable for analog input control (AINDS).
- Specify AMD for the AD converter control operation mode (software mode only).

(2) Set up the AD converter control register 2 (ADCCR2) as follows:

- Set the AD conversion time using AD conversion time (ACK). For details on how to set the conversion time, refer to Note 2 for AD converter control register 2.
- Choose IREFON for DA converter control.

(3) After setting up (1) and (2) above, set AD conversion start (ADRS) of AD converter control register 1 (ADCCR1) to 1. If software start mode has been selected, AD conversion starts immediately.

(4) After an elapse of the specified AD conversion time, the AD converted value is stored in AD converted value register 1 (ADCDR1) and the AD conversion finished flag (EOCF) of AD converted value register 2 (ADCDR2) is set to 1, upon which time AD conversion interrupt INTADC is generated.

(5) EOCF is cleared to 0 by a read of the conversion result. However, if reconverted before a register read, although EOCF is cleared the previous conversion result is retained until the next conversion is completed.

(6) When STOP mode starts, the AD Control Register 1 (ADCCR1) and AD Control Register 2 (ADCCR2) have all bits initialized. When using the AD converter after exiting STOP mode, set up the registers ADCCR1 and ADCCR2 back again.

2.7.4 AD Converter Operation Modes

(1) Software start mode

After setting AMD (ADCCR1 bits 5, 6) to 01 (software start mode), set ADRS (ADCCR1 bit 7) to 1. AD conversion of the voltage at the analog input pin specified by SAIN (ADCCR1 bits 0-3) is thereby started.
After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDR1) and at the same time EOCF (ADCDR2 bit 5) is set to 1, the AD conversion finished interrupt (INTADC) is generated.
ADRS is automatically cleared after AD conversion has started. Do not set ADRS (ADCCR1 bit 7) newly again (restart) during AD conversion. Before you set ADRS newly again, check EOCF (ADCDR bit 5) to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

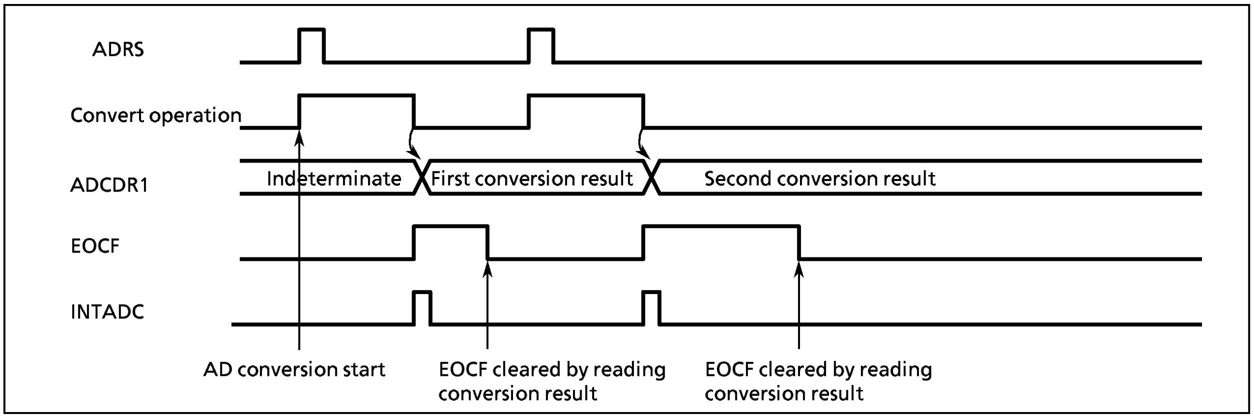


Figure 2-28. Operation in Software Start Mode

2.7.5 STOP Modes during AD

When standby mode (STOP mode) is forcibly entered during AD conversion, the AD conversion is stopped in the middle and the AD converter is initialized. In this case, the conversion result is indeterminate. (Because the results for up to the last conversion performed also are cleared, always read out the conversion results before entering standby mode.) Also, be sure to restart AD conversion after returning from standby mode, because AD conversion is not automatically restarted. Note that because the analog reference voltage is automatically turned off, there will be no currents flowing into the analog reference voltage.

2.7.6 Analog Input Voltage and AD Conversion Result

Example: After selecting the conversion time of 39.0 μs at 8 MHz and the analog input channel AIN4 pin, perform AD conversion once. After checking EOCF, read the converted value and store the 8-bit data in address 009F$_H$ on RAM. The operation mode is software start mode.

```
                                          ;  AIN SELECT
         LD     (P3DR), 00000000B         ;  P3DR bit 6 = 0
         LD     (P3CR), 00000000B         ;  P3CR bit 6 = 0
         LD     (ADCCR1), 00100100B       ;  Select AIN4
         LD     (ADCCR2), 11011000B       ;  Select conversion time (312/fc) and operation mode
                                          ;  AD CONVERT START
         SET    (ADCCR1). 7               ;  ADRS = 1
SLOOP:   TEST   (ADCDR2). 5               ;  EOCF = 1 ?
         JRS    T, SLOOP
                                          ;  RESULT DATA READ
         LD     A, (ADCDR1)
         LD     (9FH), A
```

The analog input voltage is corresponded to the 8-bit digital value converted by the AD as shown in Figure 2-29.

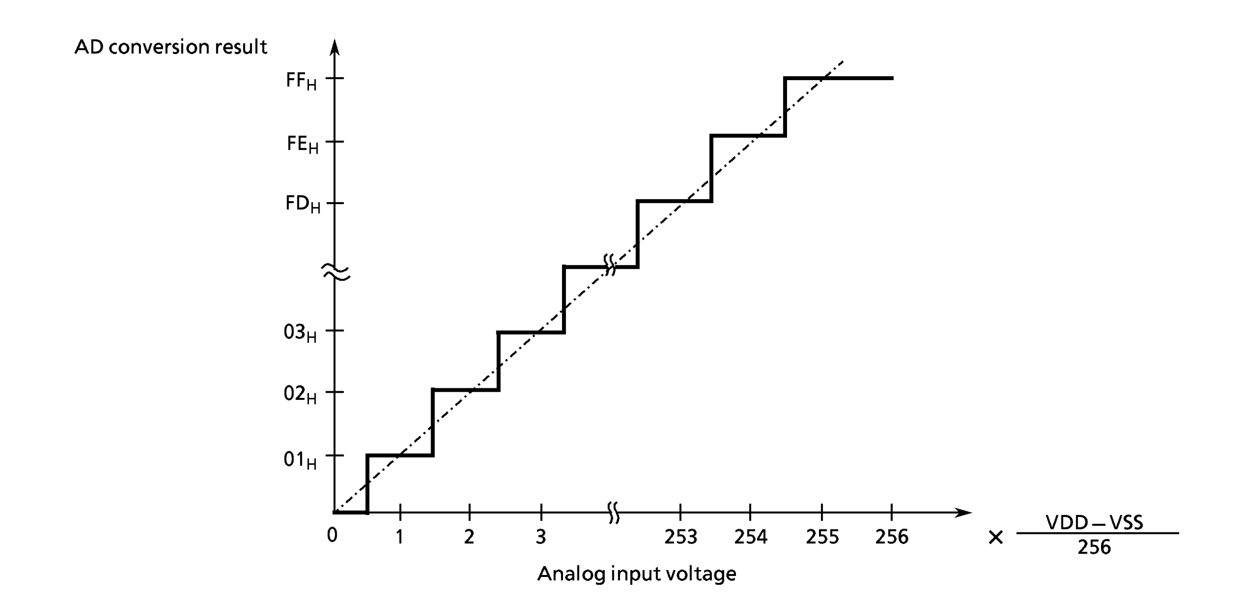


Figure 2-29. Analog Input Voltage and AD Conversion Result (typ.)

2.7.7 Precautions about AD Converter

(1) Analog input pin voltage range

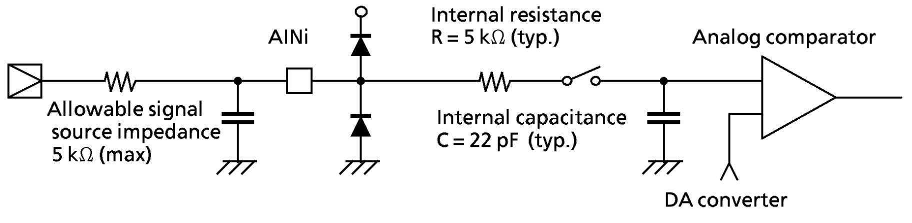
Make sure the analog input pins (AIN2 to AIN5) are used at voltages within VSS below VDD. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain. The other analog input pins also are affected by that.

(2) Analog input shared pins

The analog input pins (AIN2 to AIN5) are shared with input/output ports. When using any of the analog inputs to execute AD conversion, do not execute input/output instructions for all other ports. This is necessary to prevent the accuracy of AD conversion from degrading. Not only these analog input shared pins, some other pins may also be affected by noise arising from input/output to and from adjacent pins.

(3) Noise countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 2-30. The higher the output impedance of the analog input source, more easily they are susceptible to noise. Therefore, make sure the output impedance of the signal source in your design is 5 kΩ or less. Toshiba also recommends attaching a capacitor external to the chip.



Note: i = 2 to 5

Figure 2-30. Analog Input Equivalent Circuit and Example of Input Pin Processing

2.8 EPROM Mode

2.8.1 Program Memory (for EPROM Mode)

The TMP86P202/203 have an 2 Kbytes (addresses F800 to $FFFF_H$ in the MCU mode, addresses 0000 to $07FF_H$ in the PROM mode) one-time PROM.

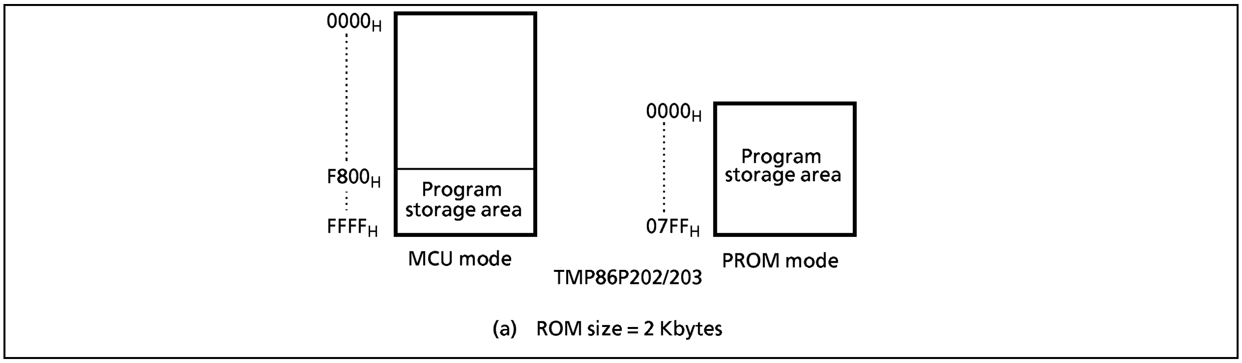


Figure 2-31. Program Memory Area

Note: Either write the data FF_H to the unused area or set the general-purpose PROM programmer to access only the program storage area

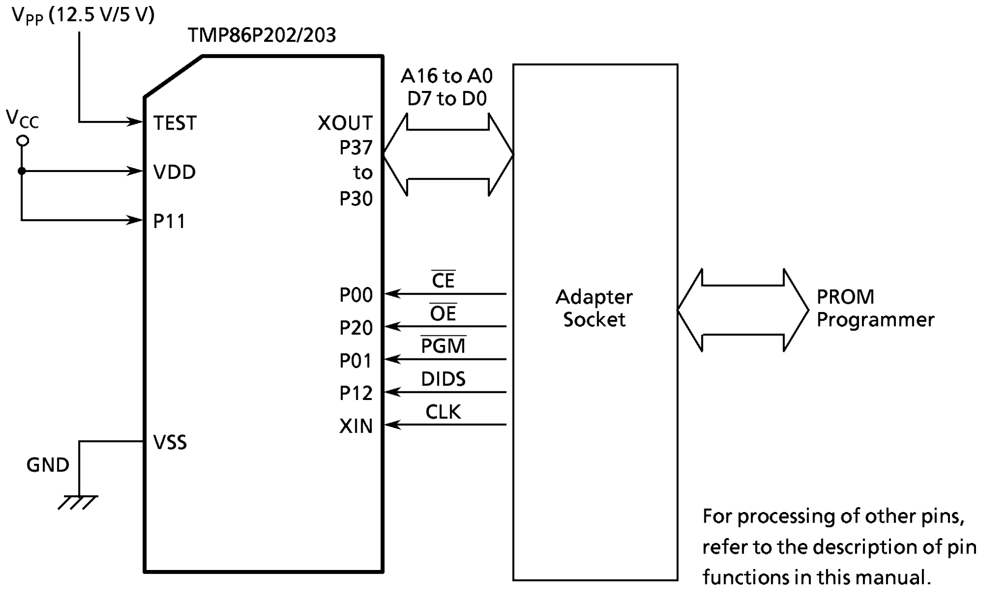
2.8.2 PROM Mode Control

PROM mode is entered by setting the $\overline{\text{RESET}}$ pin, P01, P00, P12, P11, P20, and P37 to P30 ports, and TEST as shown in Figure 2-32. In PROM mode, the ROM can be programmed and verified using a general-purpose PROM programmer.

> *Note: Fast program mode can be used. (Because settings vary with each PROM programmer used, refer to the user's manual for your PROM programmer.) The TMP86P202/203 does not have an electric signature function, so be sure to set the ROM type in your PROM programmer to TC571000D/AD or equivalent.*

When using one of the adapter sockets supported by Toshiba, set its switch to the "N" position.

EPROM adapter socket (TC571000 · 1 Mbit EPROM)



Note: Adapter sockets for connecting PROM programmer
BM11203 used for the TMP86P202/203P
BM11204 used for the TMP86P202/203M

Written inside the frame are the pin names on the TMP86P202/203 side.
Written outside the frame are the pin names on the EPROM side.

Figure 2-32. Setting Up PROM Mode

2.8.3 Programming Flowchart (Fast Program Mode)

Fast program mode is entered by applying a programming voltage, V_{PP} = 12.75 V, while V_{CC} = 6.25 V. After entering valid address and input data, apply a 0.1 ms programming (single) pulse to $\overline{\text{PGM}}$ input, and the data is written to the ROM. Verify whether the data has been written correctly; if not, apply a 0.1 ms programming pulse again. Repeat this (for up to 25 times) until data is written correctly. When programmed correctly at the set address, increment the address by one and write the next input data to the new address in the same way. When you finished writing data to all addresses, set V_{CC} = V_{PP} = 5 V and verify all programmed addresses.

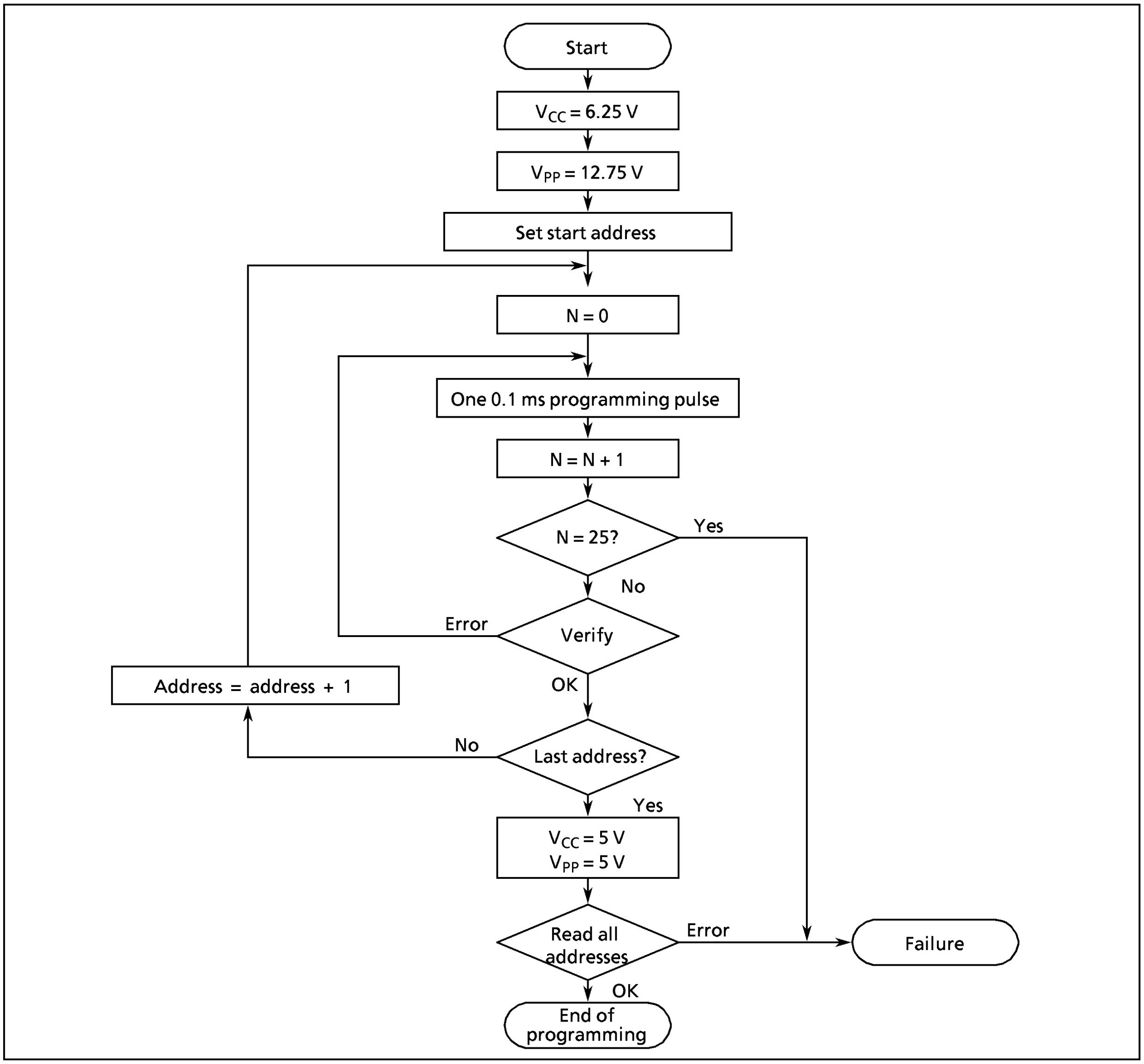


Figure 2-33. Programming Flowchart

2.8.4 Using Toshiba Adapter Socket along with General-purpose PROM Programmer

(1) Preparing the adapter

BM11203 used for the TMP86P202/203P
BM11204 used for the TMP86P202/203M (Under development)

(2) Setting up the adapter

Set the switch (SW1) to the N position.

(3) Setting up the PROM programmer

i) Set the PROM type to TC571000D/AD.
Programming voltage: 12.75 V (fast program mode)

ii) Transferring (or copying) data (Note 1)
The PROM of the TMP86P202/203 has different addresses to access when in MCU mode and when in PROM mode. Therefore, before the ROM contents to be mounted on the TMP86P202/203 can be written to the PROM of the TMP86P203, the data must be transferred (copied) from the addresses in MCU mode to the addresses in PROM mode. For the relationship of program areas between MCU and PROM modes, see Figure 1-1, “Program Memory Area” .
Example:Execute the following in block transfer (copy) mode.
When the ROM size is 2 KB: F800 to FFFF_H → 0000 to 07FF_H

iii) Set the write address. (Note 1)
Start address: 0000_H
End address: 07FF_H

(4) Programming

Program and verify the ROM following the operation procedure of your PROM programmer.

Note 1: For details on how to set, refer to the user's manual of your PROM programmer. Always make sure the data in unused address areas are set to FF_H.

Note 2: When attaching the MCU to the adapter or the adapter to the PROM programmer, make sure the No. 1 pin position on both sides are matched. If attached in reverse, the MCU, adapter, and PROM programmer may be damaged.

Note 3: The TMP86P202/203 does not support electric signature mode. If the PROM programmer is used in signature mode, a voltage of 12 V + 0.5 V is applied to No. 9 address pin (A9), which causes damage to the device. Therefore, do not use signature mode.

(5) Security bit

The TMP86P202/203 has a security bit in PROM cell.
If the security bit is programmed to 0, the content of the PROM is disable to be read (FF_H data) in PROM mode.

(How to program the security bit)

The difference from the programming procedures described in section 2.8.4 are follows.

1) Setting OTP adapter
Set the switch (SW1) to the S side.

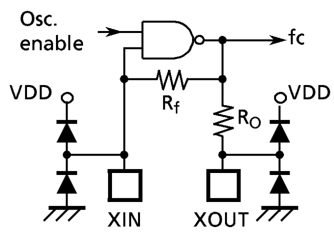
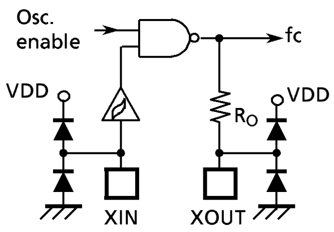
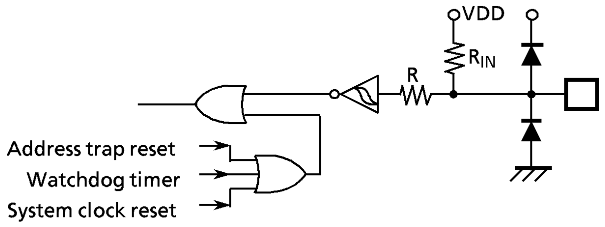
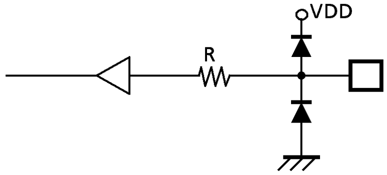
2) Setting PROM programmer
i) Setting of programming address
The security bit is in bit 0 of address 1101_H.
Set the start address 1101_H and the end address 1101_H.
Set the data FE_H at the address 1101_H.

Note: Do not alter the contents of the register at 1101_H after programming the security bit to 0.

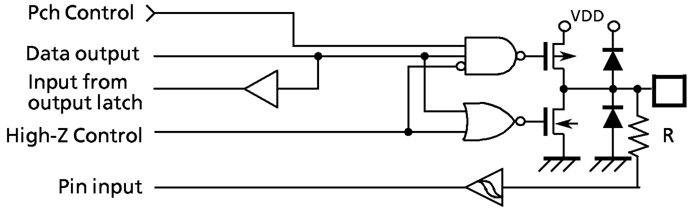
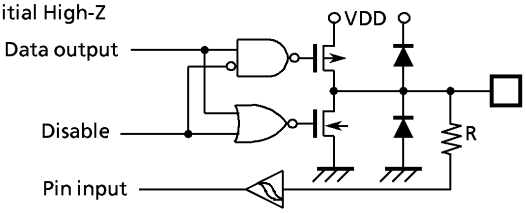
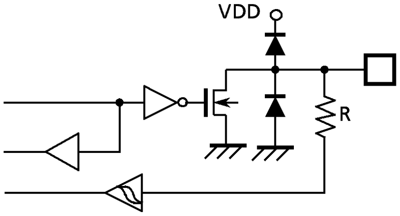
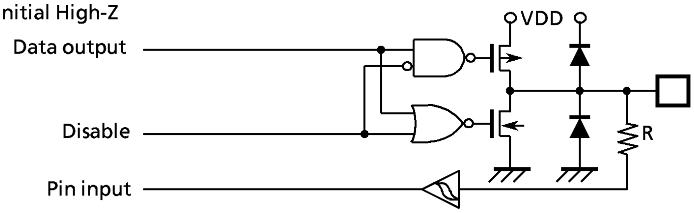
Input/Output Circuitry

(1) Control Pins

The input/output circuitries of the TMP86P202/203 control pins are shown below.

Control Pin	I/O	Input/Output Circuitry		Remarks
		TMP86P202	TMP86P203 (Under development)	
XIN XOUT	Input Output	 Osc. enable fc VDD R_f R_O VDD XIN XOUT Ceramic or crystal	 RC	Resonator connecting pins (High-frequency) R_f = 1.2 MΩ (typ.) R_O = 0.5 kΩ (typ.)
$\overline{\text{RESET}}$	Input	 VDD R_{IN} R Address trap reset Watchdog timer System clock reset		Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.) R = 100 Ω (typ.)
TEST	Input	 		R = 100 Ω (typ.)

(2) Input/Output Ports

Port	I/O	Input/Output Circuitry	Remarks
P0	I/O	Initial High-Z  Pch Control Data output Input from output latch High-Z Control Pin input VDD R	Sink open drain output or push-pull output Hysteresis input High current output (Nch) (Programmable port option) R = 100 Ω (typ.)
P1	I/O	Initial High-Z 	Tri-state I/O Hysteresis input R = 100 Ω (typ.)
P2	I/O	Initial High-Z 	Sink open drain output Hysteresis input R = 100 Ω (typ.)
P3	I/O	Initial High-Z  ∗ Refer to Figure 2-6	Tri-state I/O Hysteresis input R = 100 Ω (typ.)

Note: Input status on pins set for input mode are read in into the internal circuit. Therefore, when using the ports in a mixture of input and output modes, the contents of the output latches for the ports that are set for input mode may be rewritten by execution of bit manipulating instructions.

Electrical Characteristics

Absolute Maximum Ratings ($V_{SS} = 0$ V)

Parameter	Symbol	Pins	Rating	Unit
Supply Voltage	V_{DD}		– 0.3 to 6.5	V
Program voltage	V_{PP}	TEST/V_{PP}	– 0.3 to 13.0	
Input Voltage	V_{IN}		– 0.3 to V_{DD} + 0.3	
Output Voltage	V_{OUT1}	$\overline{\text{RESET}}$, Tri-state Port	– 0.3 to V_{DD} + 0.3	
	V_{OUT2}	P20, Sink Open Drain Port	– 0.3 to 5.5	
Output Current (Per 1 pin)	I_{OUT1} I_{OH}	P0, P1, P3 Port	– 1.8	mA
	I_{OUT2} I_{OL}	P1, P2, P3 Port	12	
	I_{OUT3} I_{OL}	P0 Port	30	
Output Current (Total)	Σ I_{OUT1}	P0, P1, P3 Port	– 12	
	Σ I_{OUT2}	P1, P2, P3 Port	40	
	Σ I_{OUT3}	P0 Port	60	
Power Dissipation [T_{opr} = 85℃]	PD	DIP	250	mW
		SOP	180	
Soldering Temperature (time)	Tsld		260 (10 s)	°C
Storage Temperature	Tstg		– 55 to 150	
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 Condition ($V_{SS} = 0$ V, Topr = – 40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Max	Unit
Supply Voltage	V_{DD}		NORMAL1 mode	3.3 (TMP86P202) 4.5 (TMP86P203)	5.5	V
			IDLE0, 1 mode			
			STOP mode	2.0		
Input high Level	V_{IH1}	Except Hysteresis input	V_{DD} ≧ 4.5 V	V_{DD} × 0.70	V_{DD}	
	V_{IH2}	Hysteresis input		V_{DD} × 0.75		
	V_{IH3}		V_{DD} < 4.5 V	V_{DD} × 0.90		
Input low Level	V_{IL1}	Except Hysteresis input	V_{DD} ≧ 4.5 V	0	V_{DD} × 0.30	
	V_{IL2}	Hysteresis input			V_{DD} × 0.25	
	V_{IL3}		V_{DD} < 4.5 V		V_{DD} × 0.10	
Clock Frequency	fc	XIN, XOUT	V_{DD} = 3.3 to 5.5 V (TMP86P202)	1.0	8.0	MHz
			V_{DD} = 4.5 to 5.5 V (TMP86P203)*	0.4	2.5	

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: AD conversion characteristics are guaranteed with limited supply voltage range (4.5 V to 5.5 V). If supply voltage is less than 4.5 V then AD conversion accuracy can not be guaranteed.

*: Under development

DC Standard

DC Characteristics (V_{SS} = 0 V, Topr = – 40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Hysteresis Voltage	V_{HS}	Hysteresis input		–	0.9	–	V
Input Current	I_{IN1}	TEST	V_{DD} = 5.5 V, V_{IN} = 5.5 V/0 V	–	–	± 2	μA
	I_{IN2}	Sink Open Drain, Tri-state Port					
	I_{IN3}	$\overline{\text{RESET}}$, $\overline{\text{STOP}}$					
Input Resistance	R_{IN2}	$\overline{\text{RESET}}$ Pull-Up		100	220	450	kΩ
Output Leakage Current	I_{LO}	Sink Open Drain, Tri-state Port	V_{DD} = 5.5 V, V_{OUT} = 5.5 V/0 V	–	–	± 2	μA
Output High Voltage	V_{OH}	P0, P1, P3 Port	V_{DD} = 4.5 V, I_{OH} = – 0.7 mA	4.1	–	–	V
Output Low Voltage	V_{OL}	P1, P2, P3 Port	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	–	–	0.4	
Output Low Current	I_{OL}	Middle Current Port (except XOUT, P0)	V_{DD} = 4.5 V, V_{OL} = 1.0 V	–	8	–	mA
Output Low Current	I_{OL}	High Current Port (P0 Port)	V_{DD} = 4.5 V, V_{OL} = 1.0 V	–	20	–	
Supply Current in NORMAL 1 mode	I_{DD}		V_{DD} = 5.5 V V_{IN} = 5.3/0.2 V fc = 8.0 MHz	–	3.0	5.5	
Supply Current in IDLE 0, 1 mode				–	1.9	4.0	
Supply Current in STOP mode			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V	–	0.5	10.0	μA

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

Note 2: Input current (I_{IN1}, I_{IN3}); The current through pull-up or pull-down resistor is not included.

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

AD Conversion Characteristics

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

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog Input Voltage	V_{AIN}		V_{SS}	–	V_{DD}	V
Power Supply Current of Analog Reference Voltage	I_{REF}	V_{DD} = 5.5 V V_{SS} = 0.0 V	–	0.6	1.0	mA
Non linearity Error		V_{DD} = 5.0 V, V_{SS} = 0.0 V	–	–	± 2	LSB
Zero Point Error			–	–	± 2	
Full Scale Error			–	–	± 2	
Total Error			–	–	± 4	

Note 1: The total error includes all errors except a quantization error, and is defined as a maximum deviation from the ideal conversion line.
Note 2: Conversion time is different in recommended value by power supply voltage.
About conversion time, please refer to "2.7.2 Register Configuration".
Note 3: Please use input voltage to AIN input Pin in limit of V_{DD} - $V_{SS.}$
When voltage of range outside is input, conversion value becomes unsettled and gives affect to other channel conversion value.
Note 4: The relevant pin for I_{REF} is $V_{DD,}$ so that the current flowing into V_{DD} is the power supply current I_{DD} + I_{REF}.
Note 5: AD conversion characteristics are guaranteed with limited supply voltage range (4.5 V to 5.5 V).
If supply voltage is less than 4.5 V then AD conversion accuracy can not be guaranteed.

AC Characteristics

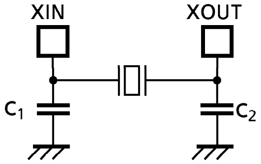
(V_{SS} = 0 V, V_{DD} = 3.3 to 5.5 V, Topr = – 40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine Cycle Time	tcy	NORMAL 1 mode IDLE 0, 1 mode	0.5	–	8	μs
High Level Clock Pulse Width	twcH	For external clock operation (XIN input) fc = 8 MHz	50	–	–	ns
Low Level Clock Pulse Width	twcL					

Recommended Oscillation Conditions

TMP86P202 (V_{SS} = 0 V, V_{DD} = 3.3 to 5.5 V, T_{opr} = – 40 to 85°C)

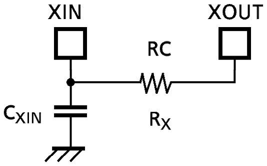
Parameter	Resonator	Oscillating Frequency	Recommended Resonator		Recommended Constant	
					C_1	C_2
High-frequency oscillation	Ceramic resonator	8 MHz	MURATA	CSA8.00MTZ CST8.00MTW	30 pF 30 pF (built-in)	30 pF 30 pF (built-in)
		4.19 MHz	MURATA	CSA4.19MG CST4.19MGW	30 pF 30 pF (built-in)	30 pF 30 pF (built-in)



(1) Ceramic, Crystal Oscillation

TMP86P203 (V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, T_{opr} = – 40 to 85°C) (Under development)

Parameter	Resonator	Oscillating Frequency	Recommended Constant	
			C_{XIN}	R_X
High-frequency oscillation	RC resonator	2 MHz	33 pF	10 kΩ
		400 kHz	100 pF	30 kΩ



(2) RC Oscillation

Note 1: When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.

Note 2: To ensure stable oscillation, the resonator position, load capacitance, etc. must be appropriate. Because there factors are greatly affected by board patterns, please be sure to evaluate operation on the board on which the device will actually be mounted.

Note 3: 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

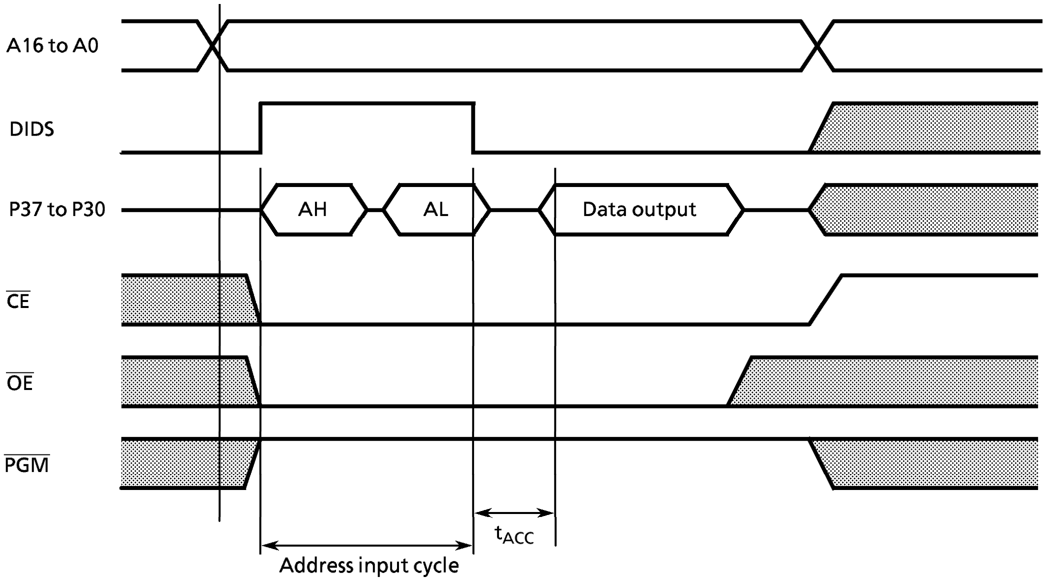
DC Characteristics, AC Characteristics (PROM mode)

(V_{SS} = 0 V, Topr = – 40 to 85°C)

(1) Read operation in PROM mode

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
High level input voltage	V_{IH4}		V_{CC} × 0.75	–	V_{CC}	V
Low leve input voltage	V_{IL4}		0	–	V_{CC} × 0.25	
Power supply	V_{CC}		4.75	5.0	5.25	
Power supply of program	V_{PP}					
Address access time	t_{ACC}	V_{CC} = 5.0 ± 0.25 V	–	–	1.5tcyc + 300	ns
Address input cycle	–		–	tcyc	–	

Note: tcyc = 250 ns at f_{CLK} = 16 MHz



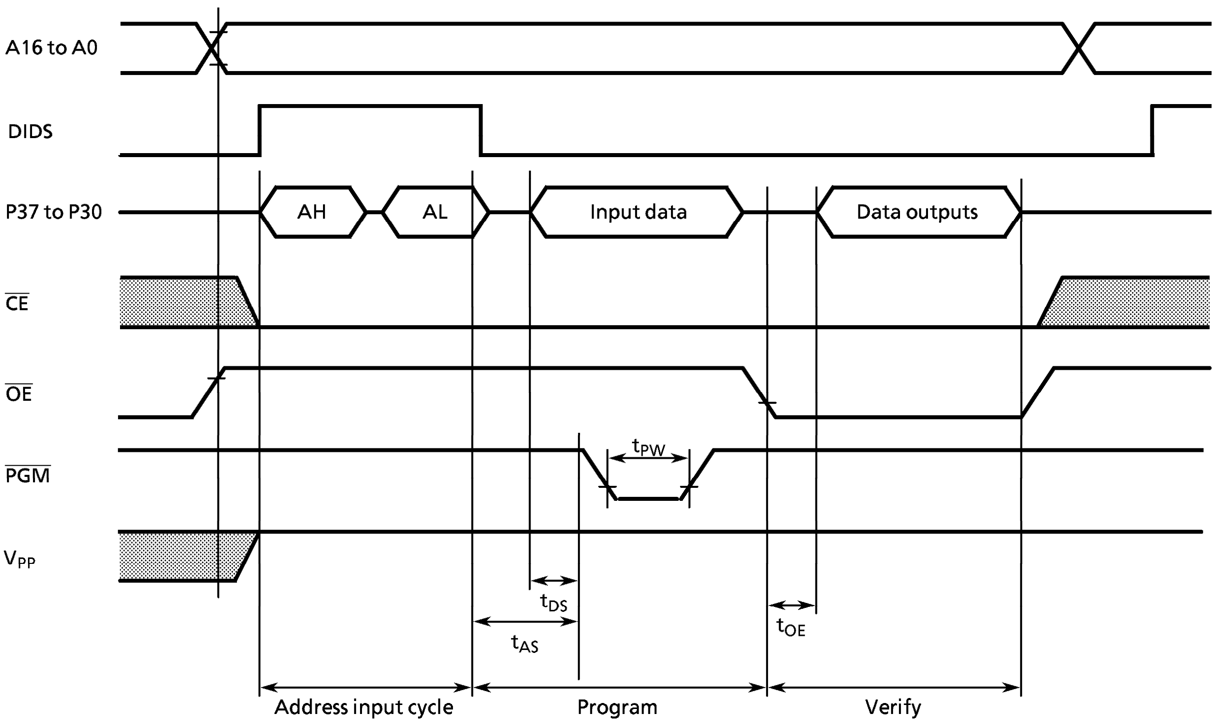
Note: DIDS and P37 to P30 are the signals for the TMP86P202/203.
All other signals are EPROM programmable.
AL: Address input (A0 to A7)
AH: Address input (A8 to A15)

(2) Program operation (High-speed) (Topr = 25 ± 5°C)

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
High level input voltage	V_{IH4}		V_{CC} × 0.75	–	V_{CC}	V
Low leve input voltage	V_{IL4}		0	–	V_{CC} × 0.25	
Power supply	V_{CC}		6.0	6.25	6.5	
Power supply of program	V_{PP}		12.5	12.75	13.0	
Pulse width of initializing program	t_{PW}	V_{CC} = 6.0 V	0.095	0.1	0.105	ms
Address set up time	t_{AS}		0.5tcyc	–	–	ns
Address input cycle	–		–	tcyc	–	
Data set up time	t_{DS}		1.5tcyc	–	–	
$\overline{\text{OE}}$ to valid output data	t_{OE}		–	–	1.5tcyc + 300	

Note: tcyc = 250 ns at f_{CLK} = 16 MHz

High-speed program writing



Note: DIDS and P37 to P30 are the signals for the TMP86P202/203.
All other signals are EPROM programmable.
AL: Address input (A0 to A7)
AH: Address input (A8 to A15)

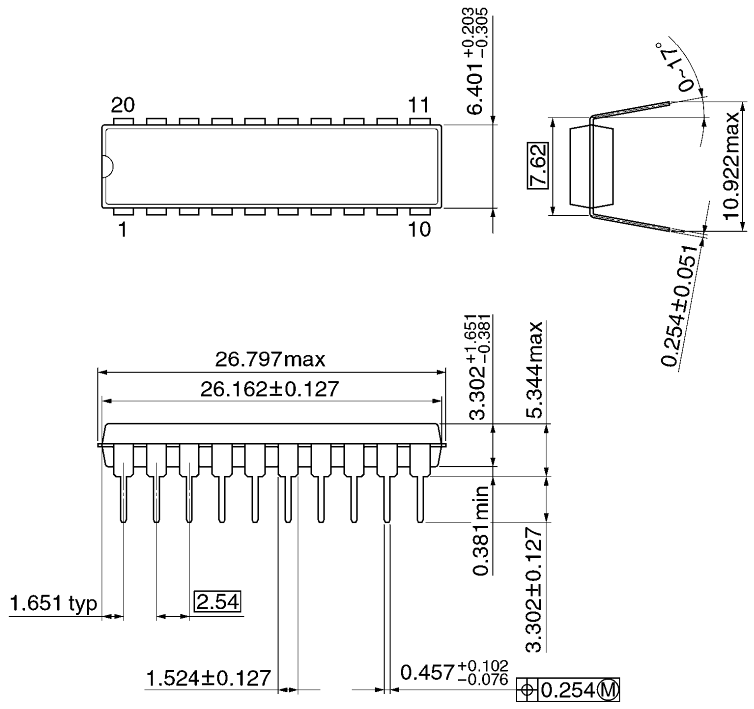
Note 1: The power supply of V_{PP} (12.75 V) must be set power-on at the same time or the later time for a power supply of V_{CC} and must be clear power-on at the same time or early time for a power supply of V_{CC}.

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

Note 3: Use the recommended adapter (see 1.2.2 (1)) and mode (see 1.2.2 (3) i).
Using other than the above condition may cause the trouble of the writting.

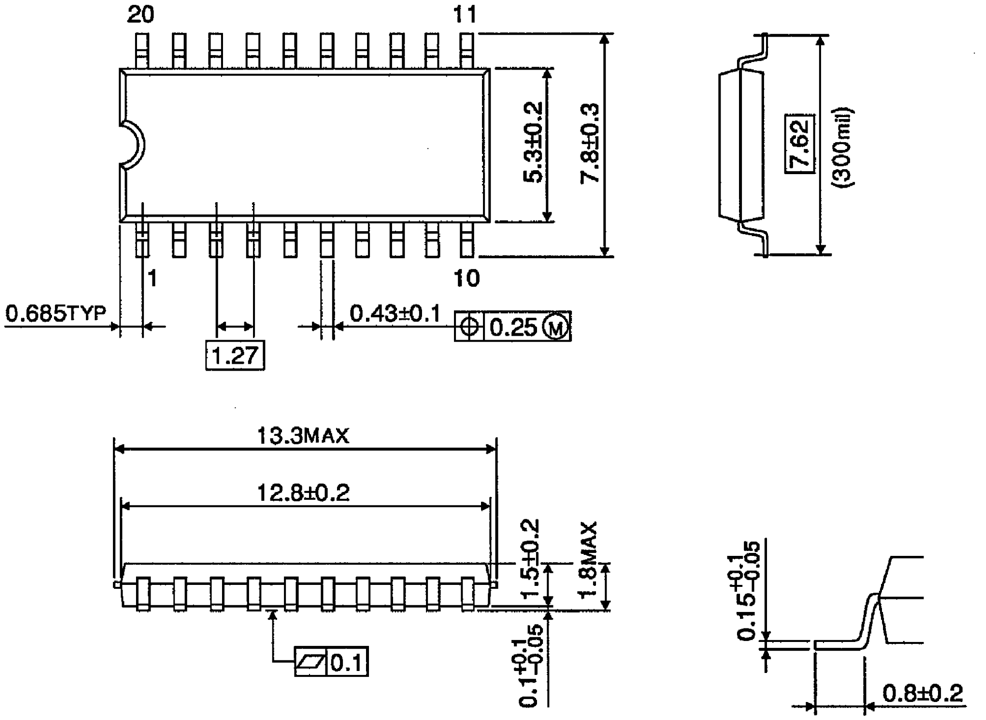
P-DIP20-300-2.54D

Unit: mm



P-SOP20-300-1.27

Unit: mm


20
11
1
10
5.3±0.2
7.8±0.3
7.62
(300mil)
0.685TYP
1.27
0.43±0.1
⊕ 0.25 Ⓜ
13.3MAX
12.8±0.2
1.5±0.2
1.8MAX
0.1
0.1 +0.1 −0.05
0.15 +0.1 −0.05
0.8±0.2