TOSHIBA

8 Bit Microcontroller
TLCS-870/C Series

TMP86P202MG

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

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

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

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

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

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

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

Revision History

Date	Revision	
2007/10/15	1	First Release

Table of Contents

This is a technical document that describes the operating functions and electrical

specifications of the 8-bit microcontroller series TLCS-870/C (LSI).

CMOS 8-Bit Microcontroller

TMP86P202MG

The TMP86P202MG is a single-chip 8-bit high-speed and high-functionality microcomputer incorporating 2048 bytes of One-Time PROM.

Product No.	ROM (EPROM)	RAM	Package	Emulation Chip
TMP86P202MG	2048 bytes	128 bytes	SOP20-P-300-1.27	TMP86C908XB

1.1 Features

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

     0.50 μs (at 8 MHz)
   - 132 types & 731 basic instructions
2. 11interrupt sources (External : 3 Internal : 8)
3. Input / Output ports (14 pins)

   Large current output: 2pins (Typ. 20mA), LED direct drive
4. Watchdog Timer
5. Prescaler
   - Time base timer
   - Divider output function
6. 8-bit timer counter : 2 ch
   - Timer, Event counter,
   - Programmable divider output (PDO),
   - Pulse width modulation (PWM) output,
   - Programmable pulse generation (PPG) modes
7. 8-bit successive approximation type AD converter (with sample hold)

   Analog inputs: 4ch
8. Low power consumption operation

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

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

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

9. Operation voltage:

3.3 V to 5.5 V at 8MHz

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

1.2 Pin Assignment

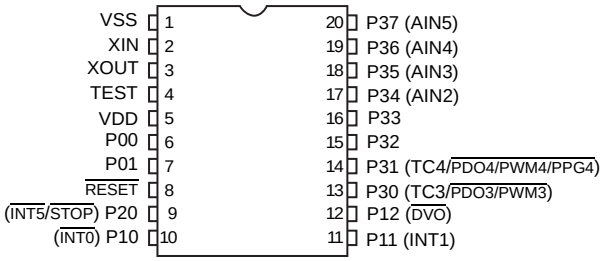


Figure 1-1 Pin Assignment

1.3 Block Diagram

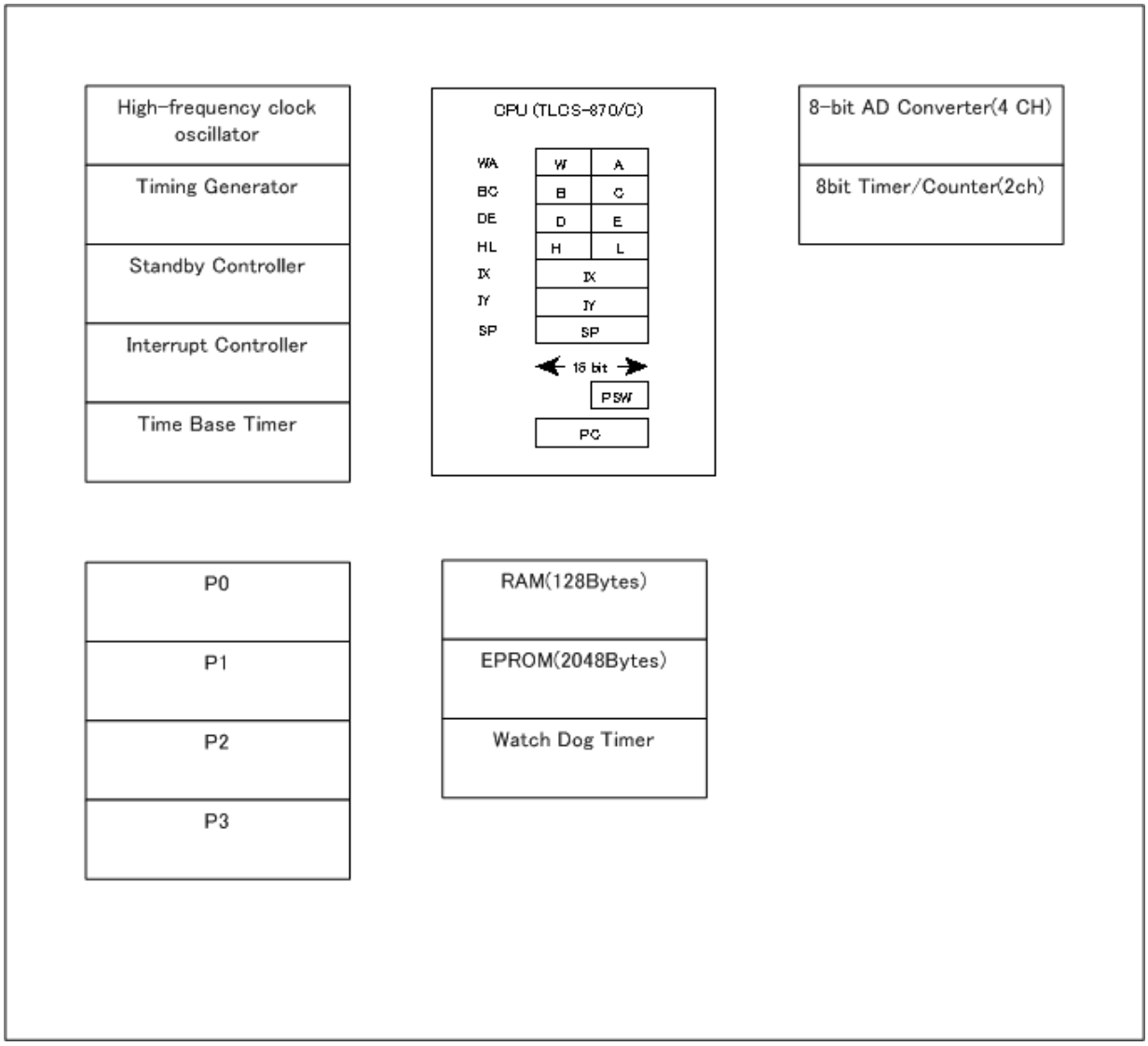


Figure 1-2 Block Diagram

1.4 Pin Names and Functions

The TMP86P202MG has MCU mode and PROM mode. Table 1-1 shows the pin functions in MCU mode. The PROM mode is explained later in a separate chapter.

Table 1-1 Pin Names and Functions

Pin Name	Pin Number	Input/Output	Functions
P01	7	IO	PORT01
P00	6	IO	PORT00
P12 $\overline{\text{DVO}}$	12	IO O	PORT12 Divider Output
P11 INT1	11	IO I	PORT11 External interrupt 1 input
P10 $\overline{\text{INT0}}$	10	IO I	PORT10 External interrupt 0 input
P20 $\overline{\text{STOP}}$ $\overline{\text{INT5}}$	9	IO I I	PORT20 STOP mode release signal input External interrupt 5 input
P37 AIN5	20	IO I	PORT37 AD converter analog input 5
P36 AIN4	19	IO I	PORT36 AD converter analog input 4
P35 AIN3	18	IO I	PORT35 AD converter analog input 3
P34 AIN2	17	IO I	PORT34 AD converter analog input 2
P33	16	IO	PORT33
P32	15	IO	PORT32
P31 TC4 $\overline{\text{PDO4/PWM4/PPG4}}$	14	IO I O	PORT31 TC4 input PDO4/PWM4/PPG4 output
P30 TC3 $\overline{\text{PDO3/PWM3}}$	13	IO I O	PORT30 TC3 input PDO3/PWM3 output
XIN	2	I	Resonator connecting pins for high-frequency clock
XOUT	3	O	Resonator connecting pins for high-frequency clock
$\overline{\text{RESET}}$	8	I	Reset signal
TEST	4	I	Test pin for out-going test. Normally, be fixed to low.
VDD	5	I	+5V
VSS	1	I	0(GND)

2. Operational Description

2.1 CPU Core Functions

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

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

2.1.1 Memory Address Map

The TMP86P202MG memory is composed OTP RAM, and SFR(Special function register). They are all mapped in 64-Kbyte address space. Figure 2-1 shows the TMP86P202MG memory address map.

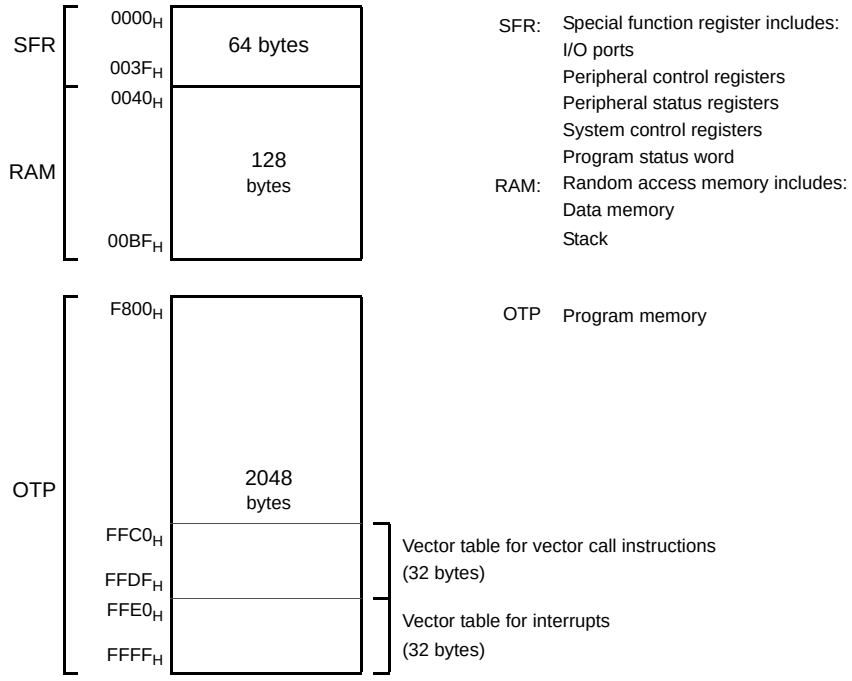


Figure 2-1 Memory Address Map

2.1.2 Program Memory (OTP)

The TMP86P202MG has a 2048 bytes (Address F800H to FFFFH) of program memory (OTP).

2.1.3 Data Memory (RAM)

The TMP86P202MG has 128 bytes (Address 0040H to 00BFH) of internal RAM. The internal RAM are located in the direct area; instructions with shorten operations are available against such an area.

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

Example :Clears RAM to “00H”. (TMP86P202MG)

```
                LD      HL, 0040H       ; Start address setup
                LD      A, H            ; Initial value (00H) setup
                LD      BC, 007FH
SRAMCLR:        LD      (HL), A
                INC     HL
                DEC     BC
                JRS     F, SRAMCLR
```

2.2 System Clock Controller

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

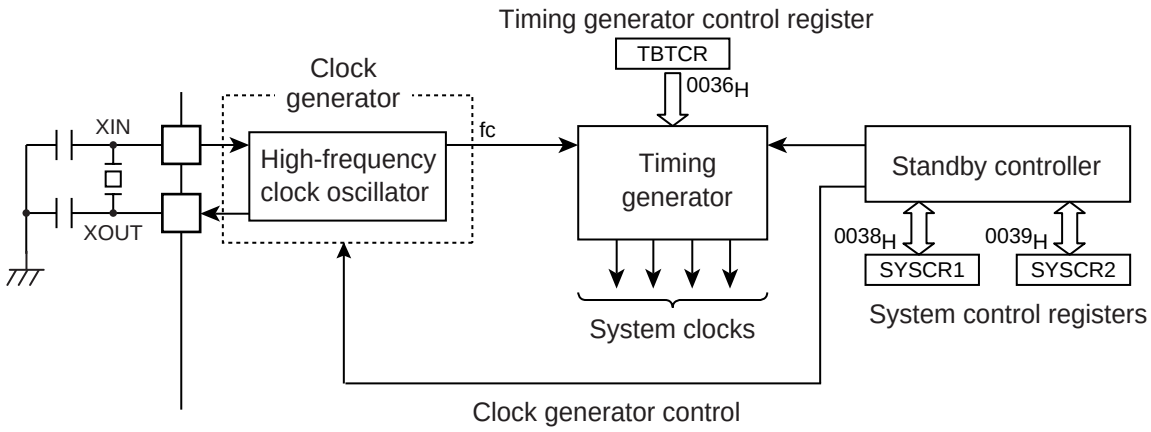


Figure 2-2 System Colck Control

2.2.1 Clock Generator

The clock generator generates the basic clock which provides the system clocks supplied to the CPU core and peripheral hardware.

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

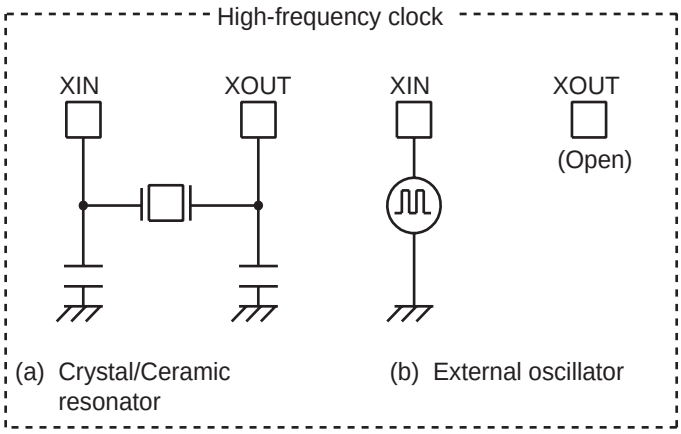


Figure 2-3 Examples of Resonator Connection

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

2.2.2 Timing Generator

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

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

2.2.2.1 Configuration of timing generator

The timing generator consists of a 2-stage prescaler, a 21-stage divider, a main system clock generator, and machine cycle counters.

As reset and STOP mode started/canceled, the prescaler and the divider are cleared to “0”.

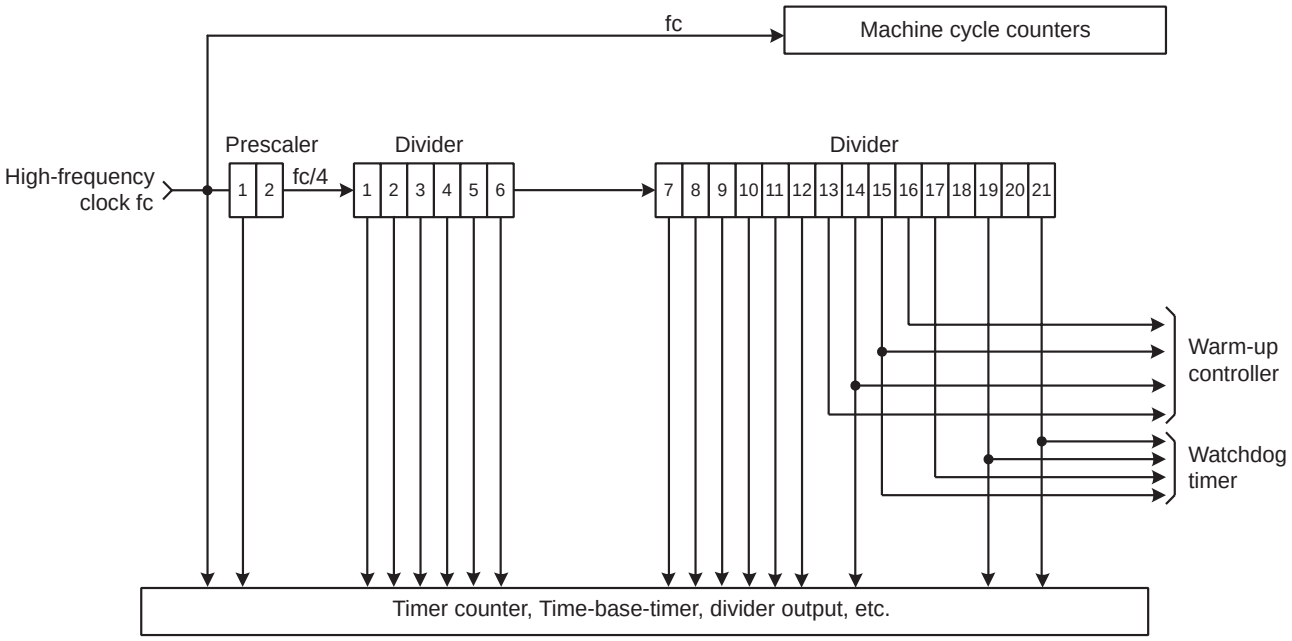


Figure 2-4 Configuration of Timing Generator

2.2.2.2 Machine cycle

Instruction execution and peripheral hardware operation are synchronized with the main system clock.

The minimum instruction execution unit is called an “machine cycle”. There are a total of 10 different types of instructions for the TLCS-870/C Series: Ranging from 1-cycle instructions which require one machine cycle for execution to 10-cycle instructions which require 10 machine cycles for execution. A machine cycle consists of 4 states (S0 to S3), and each state consists of one main system clock.

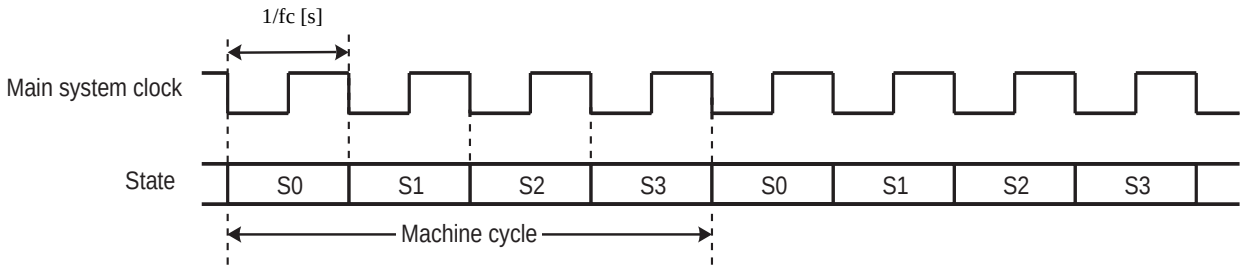


Figure 2-5 Machine Cycle

2.2.3 Operation Mode Control Circuit

The operation mode control circuit starts and stops the oscillation circuit for the high-frequency clock. There are two operating modes: Single clock mode and STOP mode. These modes are controlled by the system control registers (SYSCR1 and SYSCR2). Figure 2-6 shows the operating mode transition diagram.

2.2.3.1 Single-clock mode

The oscillation circuit for the high-frequency clock is used. The main-system clock is obtained from the high-frequency clock. In the single-clock mode, the machine cycle time is 4/fc [s].

(1) NORMAL1 mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock. The TMP86P202MG is placed in this mode after reset.

(2) IDLE1 mode

In this mode, the internal oscillation circuit remains active. The CPU and the watchdog timer are halted; however on-chip peripherals remain active (Operate using the high-frequency clock).

IDLE1 mode is started by SYSCR2<IDLE> = "1", and IDLE1 mode is released to NORMAL1 mode by an interrupt request from the on-chip peripherals or external interrupt inputs. When the IMF (Interrupt master enable flag) is “1” (Interrupt enable), the execution will resume with the acceptance of the interrupt, and the operation will return to normal after the interrupt service is completed. When the IMF is “0” (Interrupt disable), the execution will resume with the instruction which follows the IDLE1 mode start instruction.

(3) IDLE0 mode

In this mode, all the circuit, except oscillator and the timer-base-timer, stops operation.

This mode is enabled by SYSCR2<TGHALT> = "1".

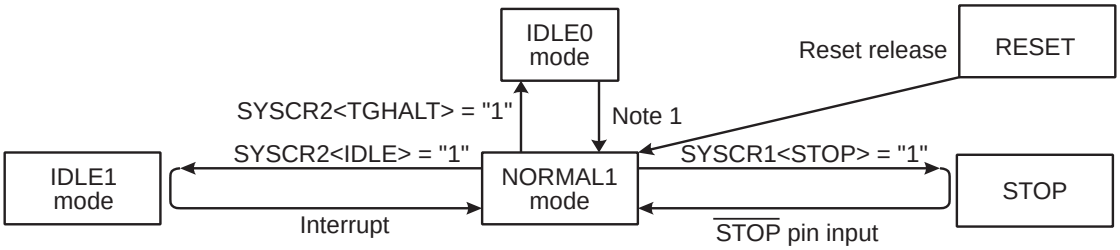
When IDLE0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR<TBTCK>, the timing generator starts feeding the clock to all peripheral circuits.

When returned from IDLE0 mode, the CPU restarts operating, entering NORMAL1 mode back again. IDLE0 mode is entered and returned regardless of how TBTCR<TBTEN> is set. When IMF = "1", EF6 (TBT interrupt individual enable flag) = "1", and TBTCR<TBTEN> = "1", interrupt processing is performed. When IDLE0 mode is entered while TBTCR<TBTEN> = "1", the INTTBT interrupt latch is set after returning to NORMAL1 mode.

2.2.3.2 STOP mode

In this mode, the internal oscillation circuit is turned off, causing all system operations to be halted. The internal status immediately prior to the halt is held with a lowest power consumption during STOP mode.

STOP mode is started by the system control register 1 (SYSCR1), and STOP mode is released by a inputting (Either level-sensitive or edge-sensitive can be programmably selected) to the $\overline{\text{STOP}}$ pin. After the warm-up period is completed, the execution resumes with the instruction which follows the STOP mode start instruction.



Note 1: The mode is released by falling edge of TBTCR<TBTCK> setting.

Figure 2-6 Operating Mode Transition Diagram

Table 2-1 Operating Mode and Conditions

Operating Mode		Oscillator: High Frequency	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		–

System Control Register 1

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

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

Note 1: When STOP mode is released with $\overline{\text{RESET}}$ pin input, a return is made to NORMAL1.

Note 2: fc: High-frequency clock [Hz], *; Don't care

Note 3: Bits 1 and 0 in SYSCR1 are read as undefined data when a read instruction is executed.

Note 4: As the hardware becomes STOP mode under OUTEN = "0", input value is fixed to "0"; therefore it may cause external interrupt request on account of falling edge.

Note 5: 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 6: Always set bit5 in SYSCR1 to "0".

Note 7: The warmig-up time should be set correctly for using oscillator.

System Control Register 2

SYSCR2	7	6	5	4	3	2	1	0	
(0039H)	XEN	"0"	"0"	IDLE		TGHALT			(Initial value: 1000 *0**)

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

Note 1: When SYSCR2<XEN> is cleard to "0", the device is reset.

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

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

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

Note 5: Because returning from IDLE0 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 6: When IDLE1 mode is released, IDLE is automatically cleared to "0".

Note 7: When IDLE0 mode is released, TGHALT is automatically cleared to "0".

Note 8: Always clear bit6 and 5 in SYSCR2 to "0".

Note 9: 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 mode is released.

2.2.4 Operating Mode Control

2.2.4.1 STOP mode

STOP mode is controlled by the system control register 1, the $\overline{\text{STOP}}$ pin input.
The $\overline{\text{STOP}}$ pin is also used both as a port P20 and an $\overline{\text{INT5}}$ (external interrupt input 5) pin. STOP mode is started by setting SYSCR1<STOP> to "1". During STOP mode, the following status is maintained.

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

STOP mode includes a level-sensitive mode and an edge-sensitive mode, either of which can be selected with the SYSCR1<RELM>.

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

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

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

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

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

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

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

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

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

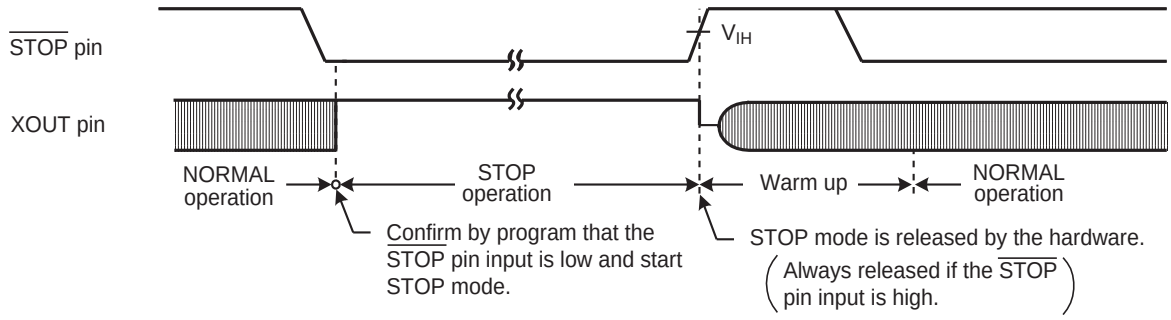


Figure 2-7 Level-sensitive Release Mode

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

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

(2) Edge-sensitive release mode (RELM = "0")

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

Example :Starting STOP mode from NORMAL mode

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

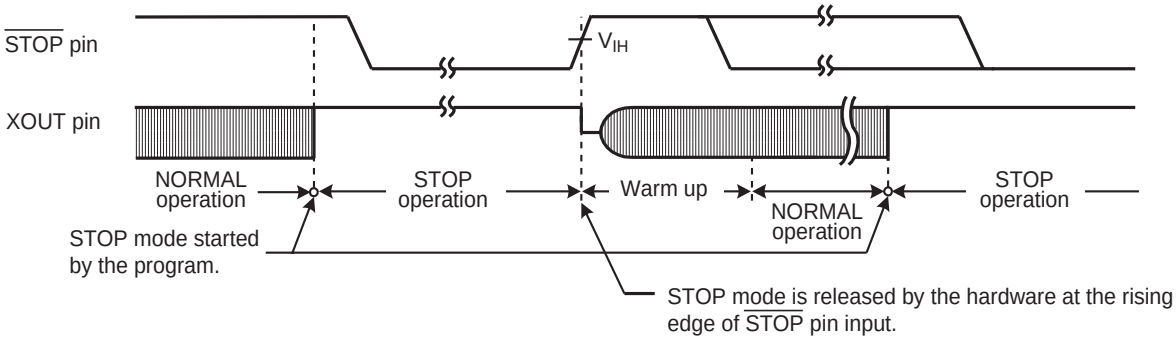


Figure 2-8 Edge-sensitive Release Mode

STOP mode is released by the following sequence.

1. The high-frequency clock oscillator is turned on.
2. A warm-up period is inserted to allow oscillation time to stabilize. During warm up, all internal operations remain halted. Four different warm-up times can be selected with the SYSCR1<WUT> in accordance with the resonator characteristics.
3. When the warm-up time has elapsed, normal operation resumes with the instruction following the STOP mode start instruction.

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

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

Note 3: When STOP mode is released with a low hold voltage, the following cautions must be observed. The power supply voltage must be at the operating voltage level before releasing STOP mode. The $\overline{\text{RESET}}$ pin input must also be “H” level, rising together with the power supply voltage. In this case, if an external time constant circuit has been connected, the $\overline{\text{RESET}}$ pin input voltage will increase at a slower pace than the power supply voltage. At this time, there is a danger that a reset may occur if input voltage level of the $\overline{\text{RESET}}$ pin drops below the non-inverting high-level input voltage (Hysteresis input).

Table 2-2 Warm-up Time Example (at fc = 8.0 MHz)

WUT	Warm-up Time [ms]
00	24.576
01	8.192
10	6.144
11	2.048

Note 1: The warm-up time is obtained by dividing the basic clock by the divider. Therefore, the warm-up time may include a certain amount of error if there is any fluctuation of the oscillation frequency when STOP mode is released. Thus, the warm-up time must be considered as an approximate value.

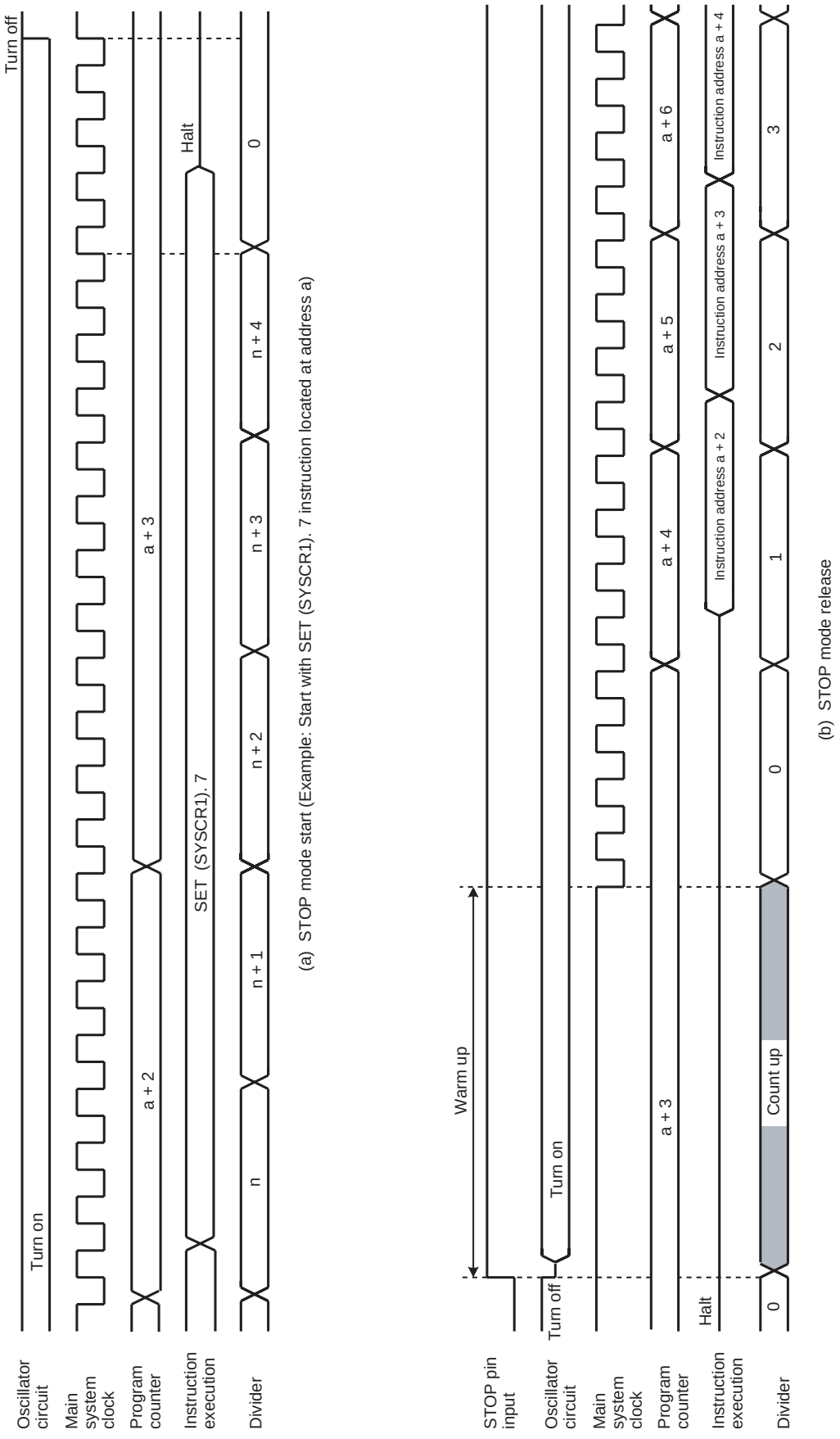


Figure 2-9 STOP Mode Start/Release

2.2.4.2 IDLE1 mode

IDLE1 mode is controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following status is maintained during this 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 this mode were entered.
3. The program counter holds the address 2 ahead of the instruction which starts this mode.

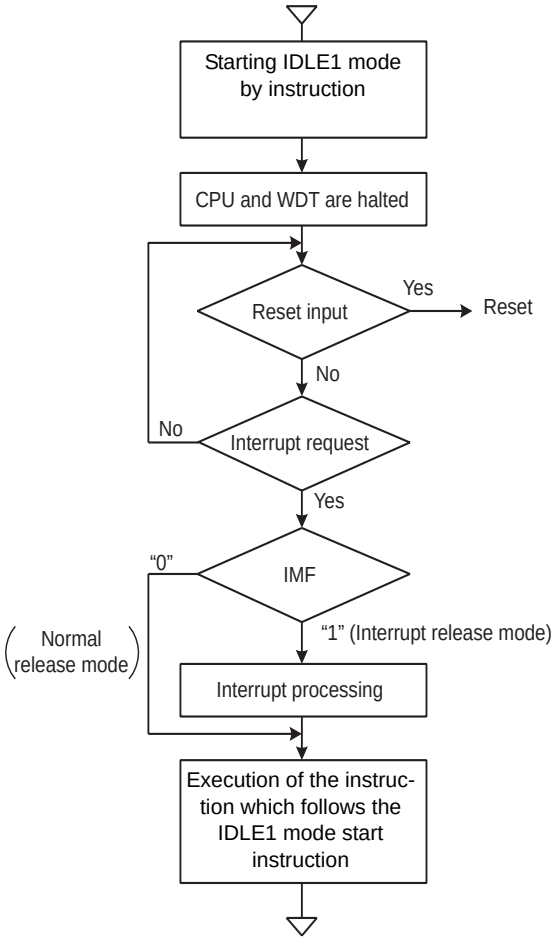


Figure 2-10 IDLE1 Mode

- Start the IDLE1 mode

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

- Release the IDLE1 mode

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

  IDLE1 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.

(1) Normal release mode (IMF = “0”)

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

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

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

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

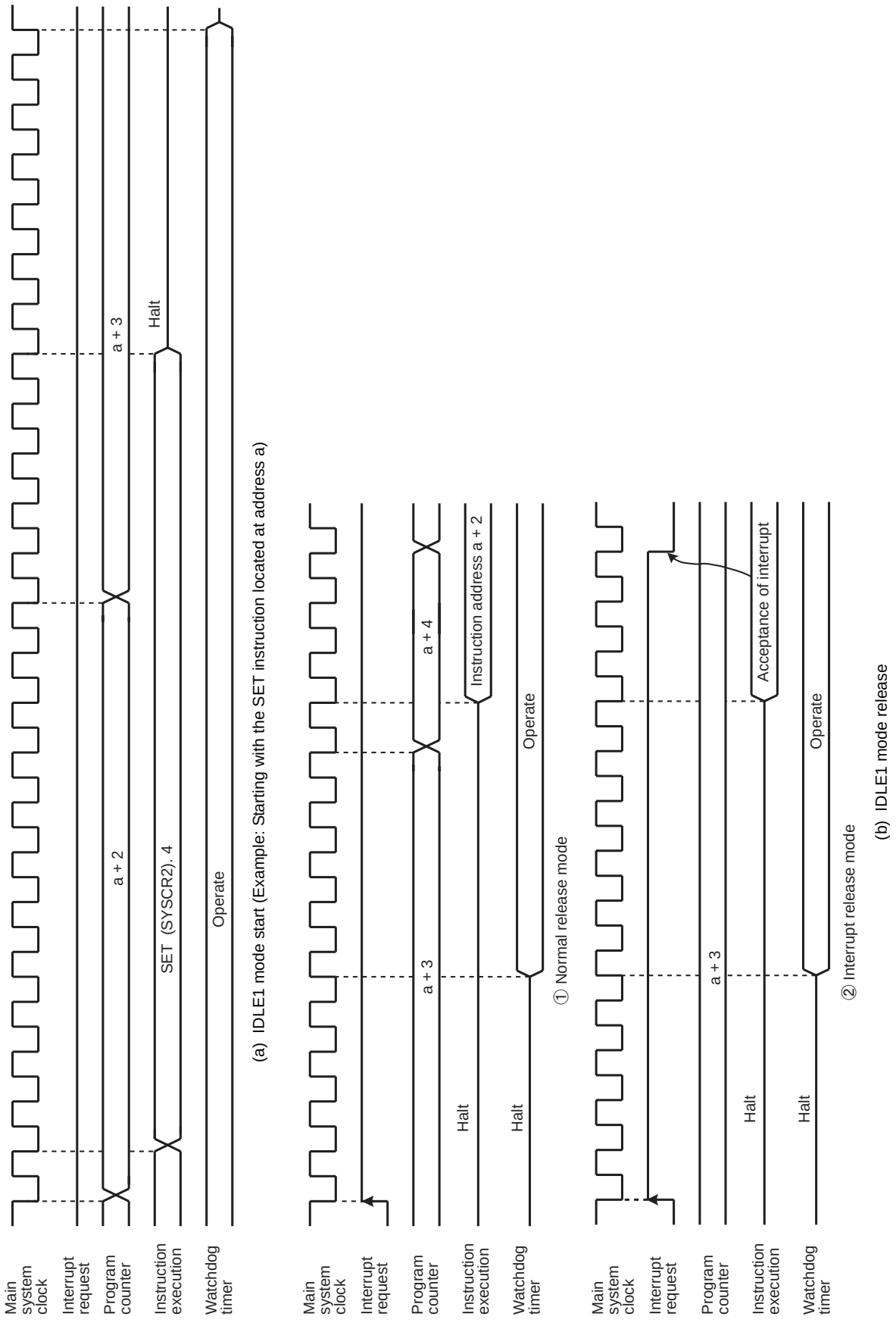


Figure 2-11 IDLE1 Mode Start/Release

2.2.4.3 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) peripherals.

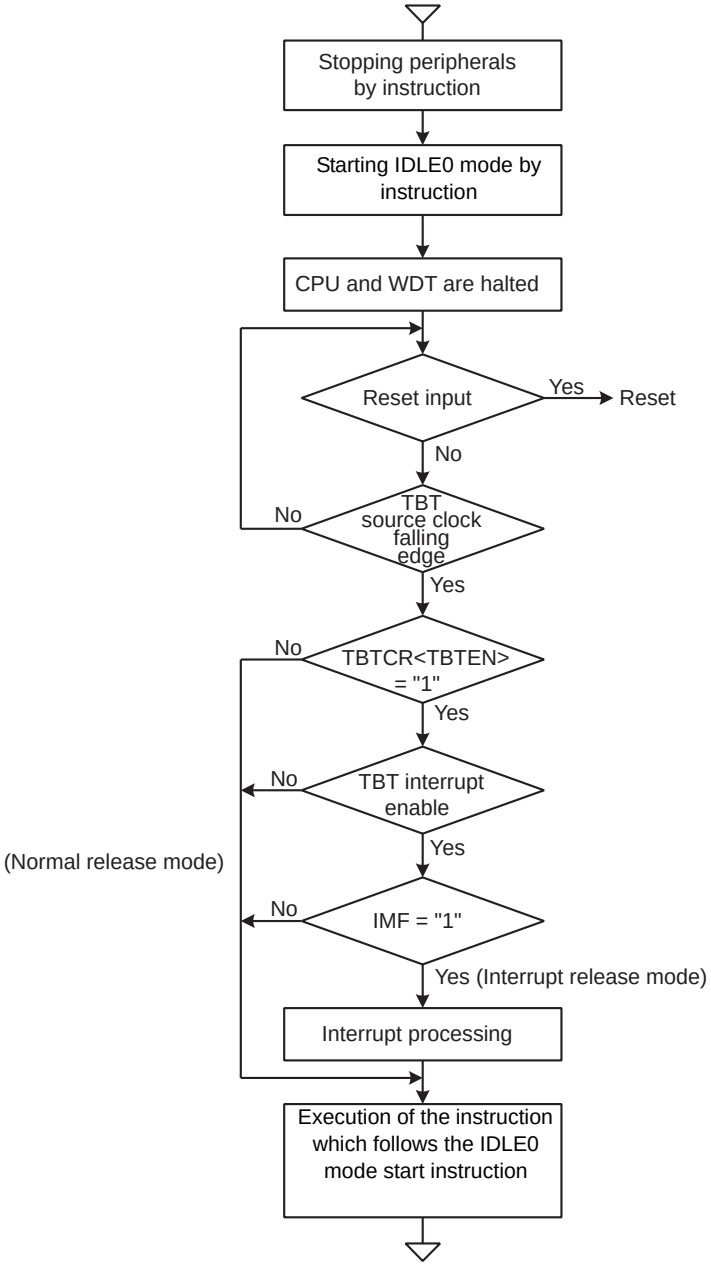


Figure 2-12 IDLE0 Mode

- Start the IDLE0 mode

  Stop (Disable) peripherals such as a timer counter.

  To start IDLE0 mode, 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 individual interrupt enable flag of TBT and TBTCR<TBTEN>.

  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 IDLE 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.

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

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

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

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

Note 1: Because returning from IDLE0 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.

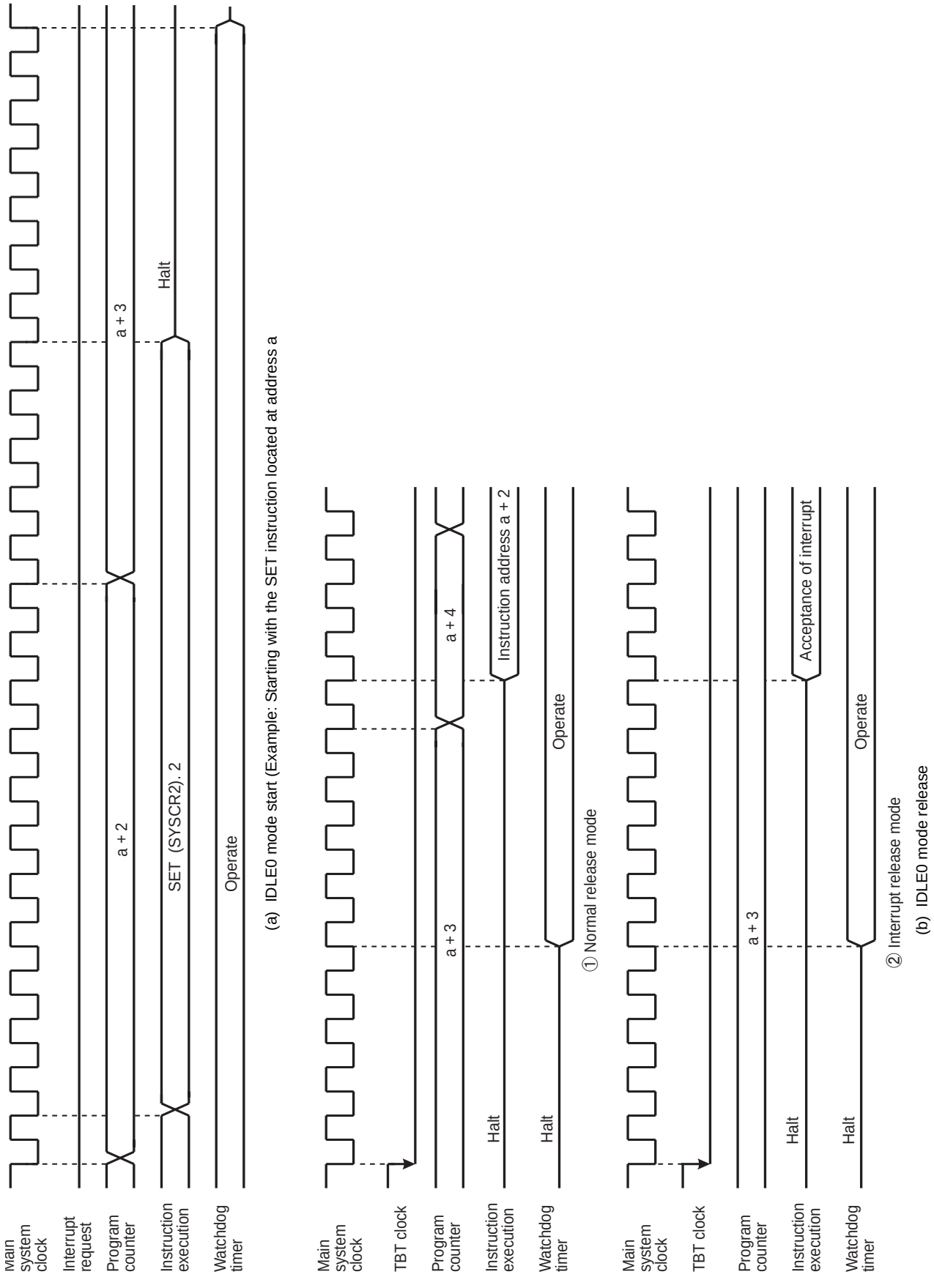


Figure 2-13 IDLE0 Mode Start/Release

2.3 Reset Circuit

The TMP86P202MG has four types of reset generation procedures: An external reset input, an address trap reset, a watchdog timer reset and a system clock reset. Of these reset, the address trap reset, the watchdog timer and the system clock reset are a malfunction reset. When the malfunction reset request is detected, reset occurs during the maximum 24/fc[s].

The malfunction reset circuit such as watchdog timer reset, address trap reset and system clock reset is not initialized when power is turned on. Therefore, reset may occur during maximum 24/fc[s] (3.0μs at 8.0 MHz) when power is turned on.

Table 1-3 shows on-chip hardware initialization by reset action.

Table 2-3 Initializing Internal Status by Reset Action

On-chip Hardware	Initial Value	On-chip Hardware	Initial Value
Program counter (PC)	(FFFEH)	Prescaler and divider of timing generator	0
Stack pointer (SP)	Not initialized		
General-purpose registers (W, A, B, C, D, E, H, L, IX, IY)	Not initialized		
Jump status flag (JF)	Not initialized	Watchdog timer	Enable
Zero flag (ZF)	Not initialized	Output latches of I/O ports	Refer to I/O port circuitry
Carry flag (CF)	Not initialized		
Half carry flag (HF)	Not initialized		
Sign flag (SF)	Not initialized		
Overflow flag (VF)	Not initialized		
Interrupt master enable flag (IMF)	0		
Interrupt individual enable flags (EF)	0	Control registers	Refer to each of control register
Interrupt latches (IL)	0		
		RAM	Not initialized

2.3.1 External Reset Input

The $\overline{\text{RESET}}$ pin contains a Schmitt trigger (Hysteresis) with an internal pull-up resistor.

When the $\overline{\text{RESET}}$ pin is held at “L” level for at least 3 machine cycles (12/fc [s]) with the power supply voltage within the operating voltage range and oscillation stable, a reset is applied and the internal state is initialized.

When the $\overline{\text{RESET}}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses FFFEH to FFFFH.

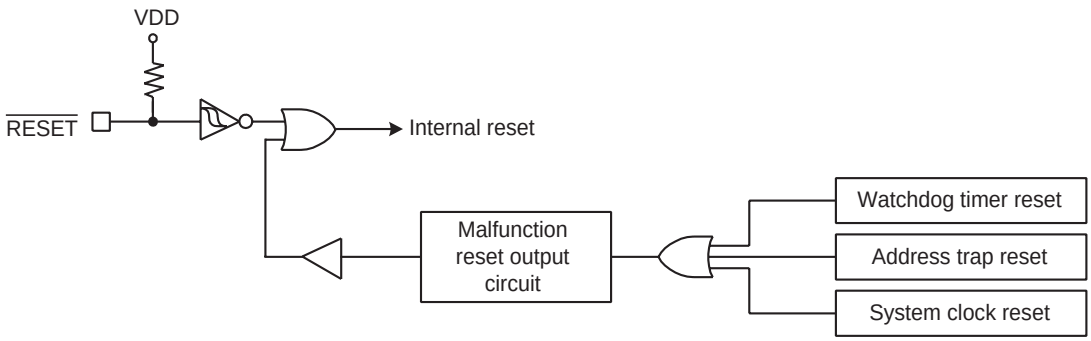
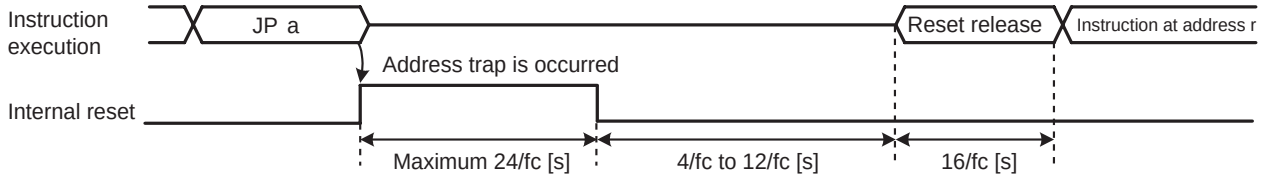


Figure 2-14 Reset Circuit

2.3.2 Address trap reset

If the CPU should start looping for some cause such as noise and an attempt be made to fetch an instruction from the on-chip RAM (when WDTCR1<ATAS> is set to “1”) or the SFR area, address trap reset will be generated. The reset time is maximum 24/fc[s] (3.0μs at 8.0 MHz).

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



Note 1: Address “a” is in the SFR or on-chip RAM (WDTCR1<ATAS> = “1”) space.

Note 2: During reset release, reset vector “r” is read out, and an instruction at address “r” is fetched and decoded.

Figure 2-15 Address Trap Reset

2.3.3 Watchdog timer reset

Refer to 2.4 “Watchdog Timer”.

2.3.4 System clock reset

If the condition as follows is detected, the system clock reset occurs automatically to prevent dead lock of the CPU. (The oscillation is continued without stopping.)

- In case of clearing SYSCR2<XEN> to “0”.

The reset time is maximum 24/fc (3.0 μs at 8.0 MHz).

3. Interrupt Control Circuit

The TMP86P202MG has a total of 11 interrupt sources excluding reset. Interrupts can be nested with priorities. Four of the internal interrupt sources are non-maskable while the rest are maskable.

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

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

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

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

3.1 Interrupt latches (IL15 to IL2)

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

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

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

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

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

interrupt service routine. However, if using multiple interrupt on interrupt service routine, manipulating EF or IL should be executed before setting IMF="1".

Example 1 :Clears interrupt latches

```
DI                              ; IMF ← 0
LDW     (ILL), 1110001110111111B ; IL12 to IL10 , IL6 ← 0
EI                              ; IMF ← 1
```

Example 2 :Reads interrupt latchess

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

Example 3 :Tests interrupt latches

```
TEST    (ILL). 6                ; if IL6 = 1 then jump
JR      F, SSET
```

3.2 Interrupt enable register (EIR)

The interrupt enable register (EIR) enables and disables the acceptance of interrupts, except for the non-maskable interrupts (Software interrupt, undefined instruction interrupt, address trap interrupt and watchdog interrupt). Non-maskable interrupt is accepted regardless of the contents of the EIR.

The EIR consists of an interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are located on address 003AH and 003BH in SFR area, and they can be read and written by an instructions (Including read-modify-write instructions such as bit manipulation or operation instructions).

3.2.1 Interrupt master enable flag (IMF)

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

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

3.2.2 Individual interrupt enable flags (EF15 to EF4)

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

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

Example 1 :Enables interrupts individually and sets IMF

```
DI                                  ; IMF ← 0
LDW     (EIRL), 1110100000100000B   ; EF15 to EF13, EF11, EF5 ← 1
  :                                   Note: IMF should not be set.
  :
EI                                  ; IMF ← 1
```

Example 2 :C compiler description example

```
unsigned  int  _io (3AH)  EIRL;     /* 3AH shows EIRL address */
_DI();
EIRL = 10100000B;
  :
_EI();
```

Interrupt Latches

(Initial value: 0**000** *00000**)

ILH,ILL (003DH, 003CH)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IL15	–	–	IL12	IL11	IL10	–	–	–	IL6	IL5	IL4	IL3	IL2		

ILH (003DH) ILL (003CH)

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

Note 1: To clear any one of bits IL6 to IL4, be sure to write "1" into IL2 and IL3.

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

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

Interrupt Enable Registers

(Initial value: 0**000** *000***0)

EIRH,EIRL (003BH, 003AH)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EF15	–	–	EF12	EF11	EF10	–	–	–	EF6	EF5	EF4				IMF

EIRH (003BH) EIRL (003AH)

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

Note 1: *: Don't care

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

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

3.3 Interrupt 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 (4.0 μs @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 3-1 shows the timing chart of interrupt acceptance processing.

3.3.1 Interrupt acceptance processing is packaged as follows.

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

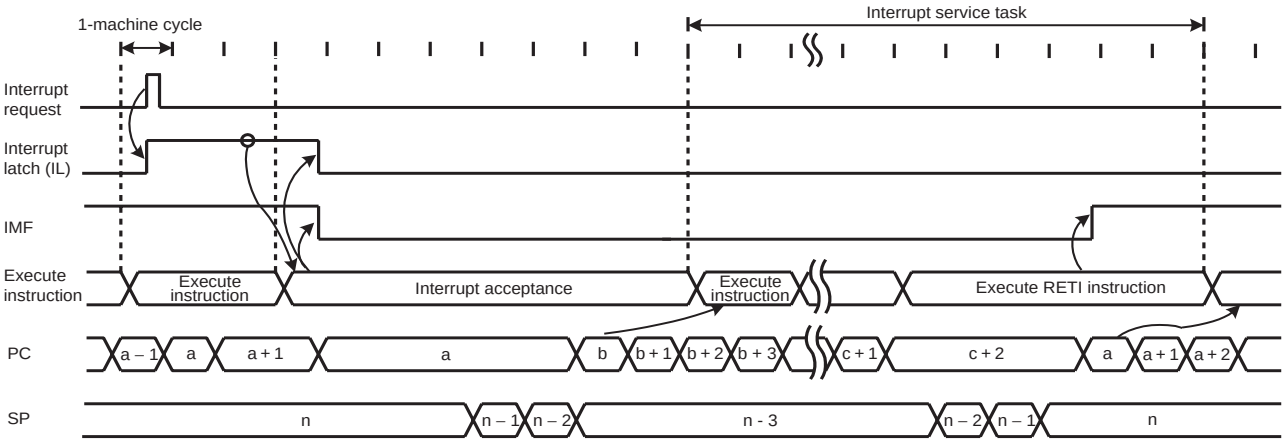
b. The interrupt latch (IL) for the interrupt source accepted is cleared to “0”.

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

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

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

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



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

Note 2: On condition that interrupt is enabled, it takes 38/fc [s] at maximum (If the interrupt latch is set at the first machine cycle on 10 cycle instruction) to start interrupt acceptance processing since its interrupt latch is set.

Figure 3-1 Timing Chart of Interrupt Acceptance/Return Interrupt Instruction

Example: Correspondence between vector table address for INTTBT and the entry address of the interrupt service program

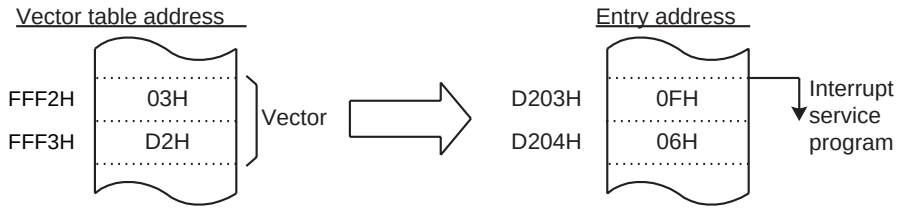


Figure 3-2 Vector table address, Entry address

A maskable interrupt is not accepted until the IMF is set to “1” even if the maskable interrupt higher than the level of current servicing interrupt is requested.

In order to utilize nested interrupt service, the IMF is set to “1” in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags.

To avoid overloaded nesting, clear the individual interrupt enable flag whose interrupt is currently serviced, before setting IMF to “1”. As for non-maskable interrupt, keep interrupt service shorten compared with length between interrupt requests; otherwise the status cannot be recovered as non-maskable interrupt would simply nested.

3.3.2 Saving/restoring general-purpose registers

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

3.3.2.1 Using PUSH and POP instructions

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

Example :Save/store register using PUSH and POP instructions

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

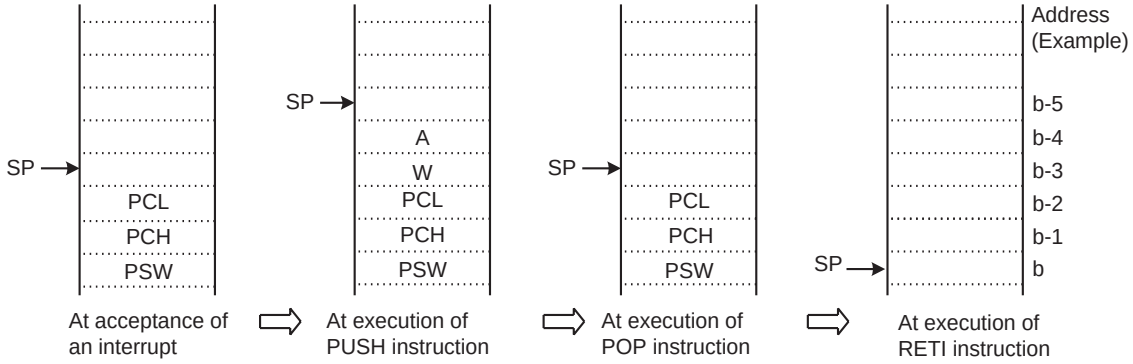


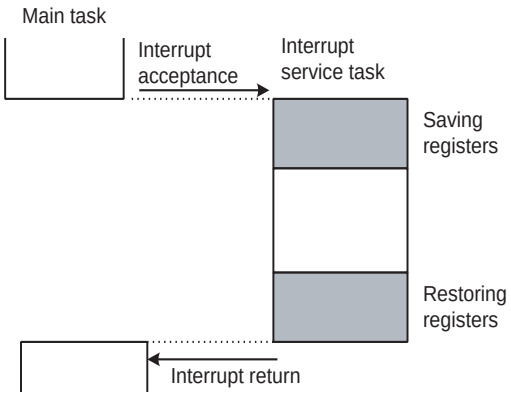
Figure 3-3 Save/store register using PUSH and POP instructions

3.3.2.2 Using data transfer instructions

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

Example :Save/store register using data transfer instructions

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



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

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

3.3.3 Interrupt return

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

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

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

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

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

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

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

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

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

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

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

3.4 Software Interrupt (INTSW)

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

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

3.4.1 Address error detection

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

3.4.2 Debugging

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

3.5 Undefined Instruction Interrupt (INTUNDEF)

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

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

3.6 Address Trap Interrupt (INTATRAP)

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

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

3.7 External Interrupts

The TMP86P202MG has 3 external interrupt inputs. These inputs are equipped with digital noise reject circuits (Pulse inputs of less than a certain time are eliminated as noise).

Edge selection is also possible with INT1. The $\overline{\text{INT0}}$/P10 pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

Edge selection, noise reject control and $\overline{\text{INT0}}$/P10 pin function selection are performed by the external interrupt control register (EINTCR).

Source	Pin	Enable Conditions	Release Edge	Digital Noise Reject
INT0	$\overline{\text{INT0}}$	IMF + EF4 + INT0EN=1	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals.
INT1	INT1	IMF + EF5 = 1	Falling edge or Rising edge	Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of 49/fc or 193/fc [s] or more are considered to be signals.
INT5	$\overline{\text{INT5}}$	IMF + EF15 = 1	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals.

Note 1: In NORMAL1 or IDLE1 mode, if a signal with no noise is input on an external interrupt pin, it takes a maximum of "signal establishment time + 6/fs[s]" from the input signal's edge to set the interrupt latch.

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

Note 3: When a pin with more than one function is used as an output and a change occurs in data or input/output status, an interrupt request signal is generated in a pseudo manner. In this case, it is necessary to perform appropriate processing such as disabling the interrupt enable flag.

External Interrupt Control Register

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

INT1NC	Noise reject time select	0: Pulses of less than 63/fc [s] are eliminated as noise 1: Pulses of less than 15/fc [s] are eliminated as noise	R/W
INT0EN	P10/$\overline{\text{INT0}}$ pin configuration	0: P10 input/output port 1: $\overline{\text{INT0}}$ pin (Port P10 should be set to an input mode)	R/W
INT1 ES	INT1 edge select	0: Rising edge 1: Falling edge	R/W

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

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

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

4. Special Function Register (SFR)

The TMP86P202MG adopts 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 0000H to 003FH.

This chapter shows the arrangement of the special function register (SFR) for TMP86P202MG.

4.1 SFR

Address	Read	Write
0000H	P0DR	
0001H	P1DR	
0002H	P2DR	
0003H	P3DR	
0004H	Reserved	
0005H	Reserved	
0006H	Reserved	
0007H	Reserved	
0008H	Reserved	
0009H	P1CR	
000AH	P3CR	
000BH	P0OUTCR	
000CH	P0PRD	-
000DH	P2PRD	-
000EH	ADCCR1	
000FH	ADCCR2	
0010H	Reserved	
0011H	Reserved	
0012H	Reserved	
0013H	Reserved	
0014H	Reserved	
0015H	Reserved	
0016H	Reserved	
0017H	Reserved	
0018H	Reserved	
0019H	Reserved	
001AH	TC3CR	
001BH	TC4CR	
001CH	TTREG3	
001DH	TTREG4	
001EH	PWREG3	
001FH	PWREG4	
0020H	ADCDR1	-
0021H	ADCDR2	-
0022H	Reserved	
0023H	Reserved	
0024H	Reserved	
0025H	Reserved	
0026H	Reserved	
0027H	Reserved	

Address	Read	Write
0028H	Reserved	
0029H	Reserved	
002AH	Reserved	
002BH	Reserved	
002CH	Reserved	
002DH	Reserved	
002EH	Reserved	
002FH	Reserved	
0030H	Reserved	
0031H	Reserved	
0032H	Reserved	
0033H	Reserved	
0034H	-	WDTCR1
0035H	-	WDTCR2
0036H	TBTCR	
0037H	EINTCR	
0038H	SYSCR1	
0039H	SYSCR2	
003AH	EIRL	
003BH	EIRH	
003CH	ILL	
003DH	ILH	
003EH	Reserved	
003FH	PSW	

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

Note 2: – ; Cannot be accessed.

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

5. I/O Ports

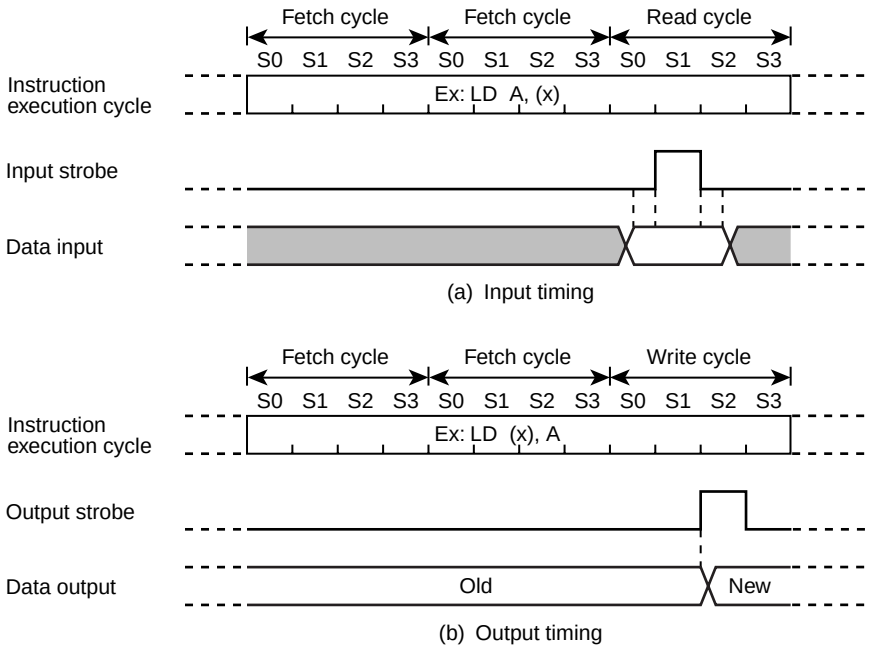
The TMP86P202MG has 4 parallel input/output ports as follows.

	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 5-1 shows input/output timing examples.

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

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



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

Figure 5-1 Input/Output Timing (Example)

5.1 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.

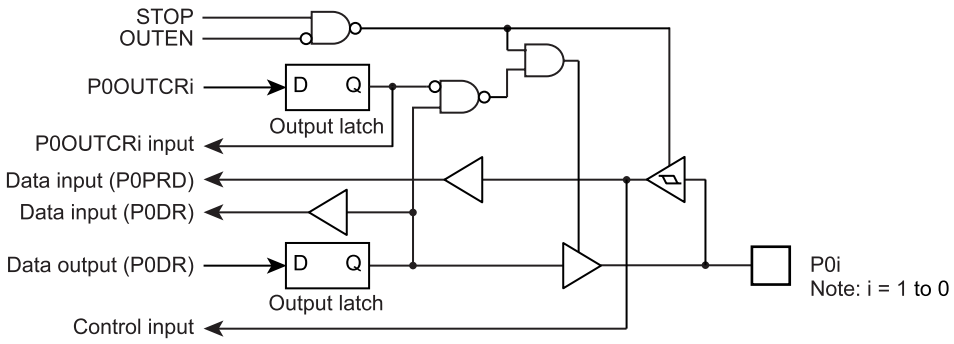


Figure 5-2 Port P0

	7	6	5	4	3	2	1	0	
P0DR (0000H) R/W							P01	P00	(Initial value: **** **11)
P0PRD (000CH) Read only							P01	P00	
P0OUTCR (000BH) R/W							P0OUTCR1	P0OUTCR0	(Initial value: **** **00)

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

5.2 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.

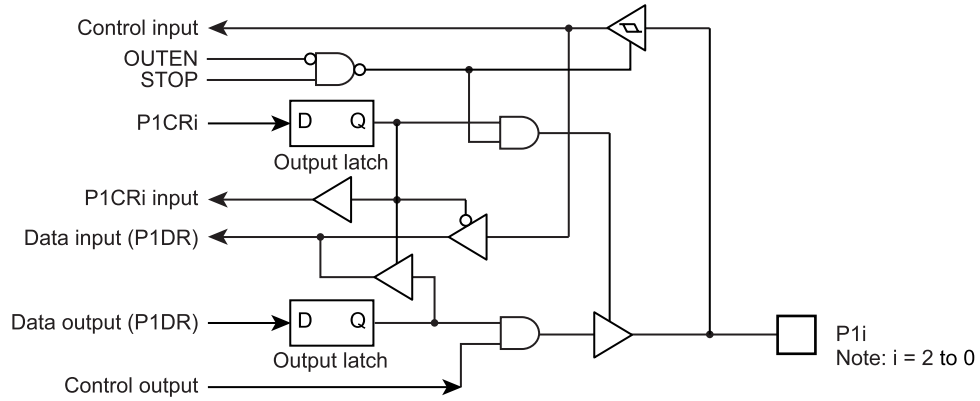


Figure 5-3 Port P1

P1DR (0001H) R/W

7	6	5	4	3	2	1	0
					P12 $\overline{\text{DVO}}$	P11 INT1	P10 $\overline{\text{INT0}}$

(Initial value: **** *000)

P1CR (0009H)

7	6	5	4	3	2	1	0

(Initial value: **** *000)

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

5.3 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.

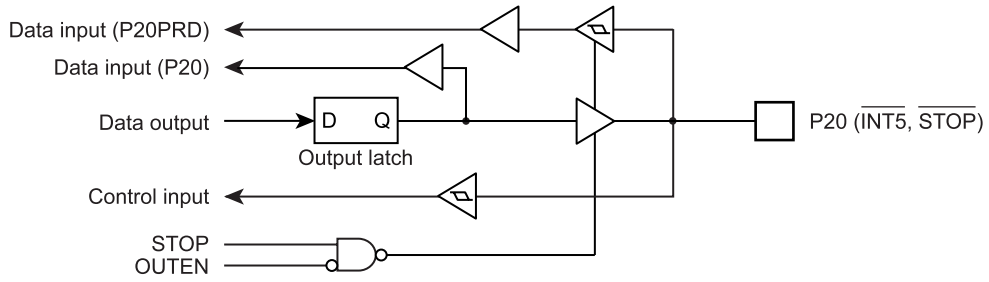


Figure 5-4 Port P2

	7	6	5	4	3	2	1	0	
P2DR (0002H) R/W								P20 INT5 STOP	(Initial value: **** ***1)
P2PRD (000DH) Read only	7	6	5	4	3	2	1	0	
								P20	

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

5.4 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.

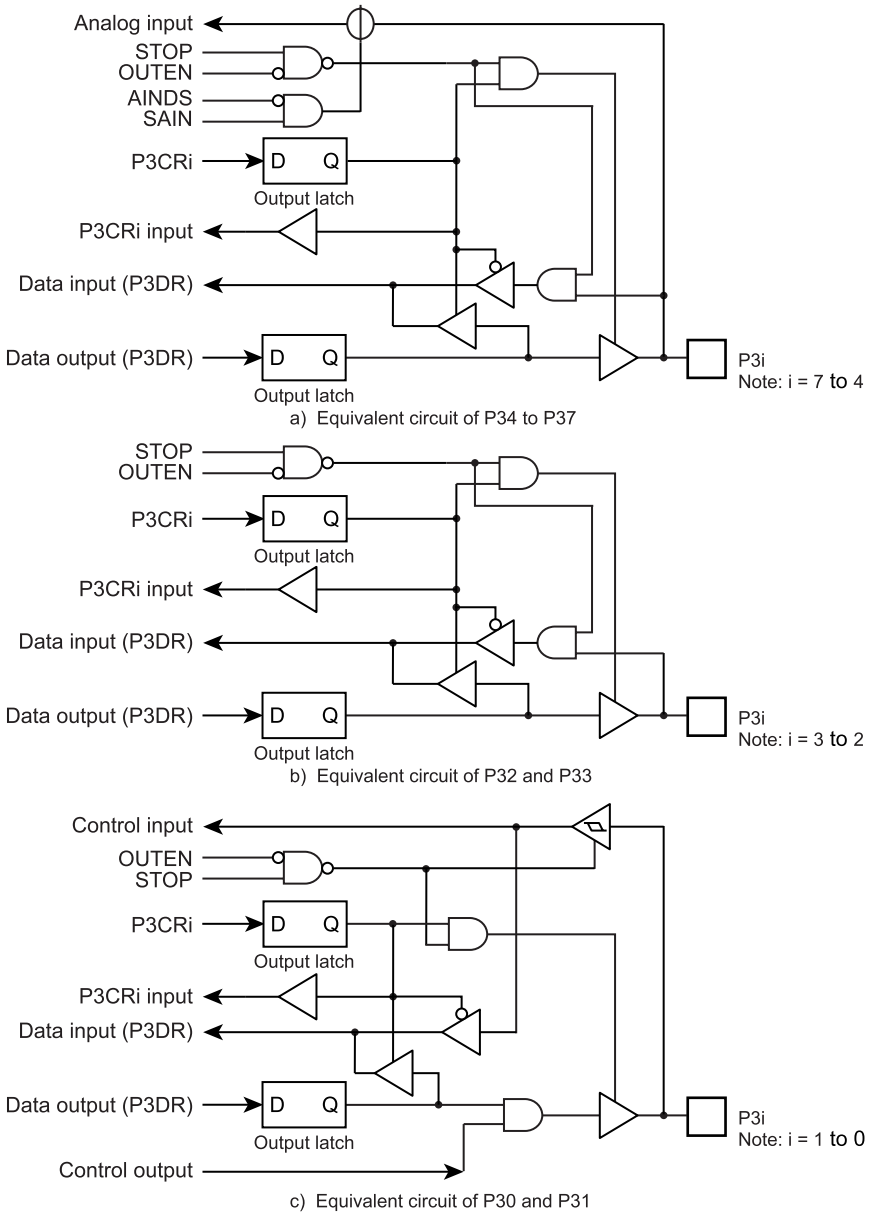


Figure 5-5 Port P3

	7	6	5	4	3	2	1	0	
P3DR (0003H) R/W	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 (000AH)	7	6	5	4	3	2	1	0	(Initial value: 0000 0000)

P3CR	Controls P3 port input/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.

6. Watchdog Timer (WDT)

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

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

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

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

6.1 Watchdog Timer Configuration

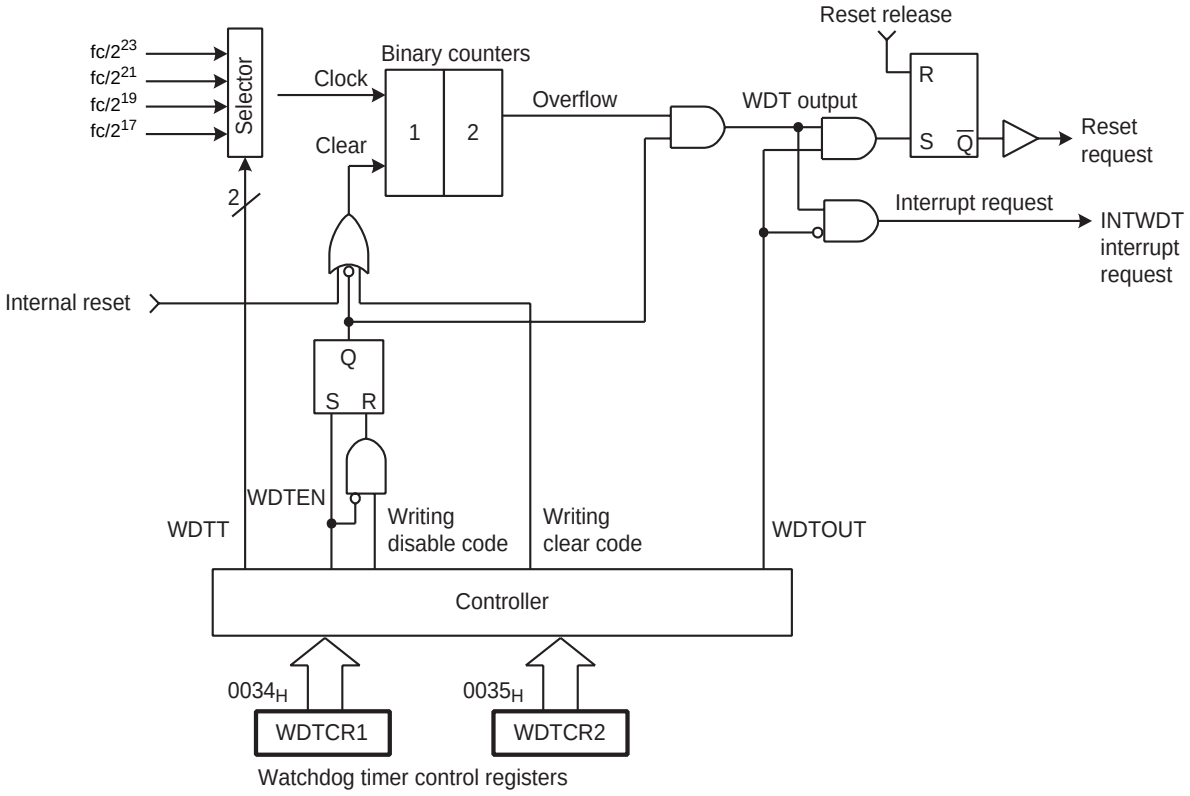


Figure 6-1 Watchdog Timer Configuration

6.2 Watchdog Timer Control

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

6.2.1 Malfunction Detection Methods Using the Watchdog Timer

The CPU malfunction is detected, as shown below.

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

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

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

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

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

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

Watchdog Timer Control Register 1

WDTCR1 (0034H)	7	6	5	4	3	2	1	0
			(ATAS)	(ATOUT)	WDTEN	WDTT		WDTOUT

(Initial value: **11 1001)

WDTEN	Watchdog timer enable/disable	0: Disable (Writing the disable code to WDTCR2 is required.) 1: Enable		Write only
WDTT	Watchdog timer detection time [s]		NORMAL1 mode	Write only
		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 request		Write only

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

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. If WDTCR1 is read, a don’t care is read.

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

Note 5: To clear WDTEN, set the register in accordance with the procedures shown in “6.2.3 Watchdog Timer Disable”.

Watchdog Timer Control Register 2

WDTCR2 (0035H)	7	6	5	4	3	2	1	0

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

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

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

Note 2: *: Don’t care

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

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

6.2.2 Watchdog Timer Enable

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

6.2.3 Watchdog Timer Disable

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

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

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

Example :Disabling the watchdog timer

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

Table 6-1 Watchdog Timer Detection Time (Example: fc = 8.0 MHz)

WDTT	Watchdog Timer Detection Time [s]
	NORMAL1 mode
00	4.194
01	1.048
10	262.144 m
11	65.536 m

6.2.4 Watchdog Timer Interrupt (INTWDT)

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

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

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

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

Example :Setting watchdog timer interrupt

```
LD      SP, 00BFH               : Sets the stack pointer
LD      (WDTCR1), 00001000B     : WDTOUT ← 0
```

6.2.5 Watchdog Timer Reset

When a binary-counter overflow occurs while WDTCR1<WDTOUT> is set to “1”, a watchdog timer reset request is generated. When a watchdog timer reset request is generated, the internal hardware is reset. The reset time is maximum 24/fc [s] (3.0 μs @ fc = 8.0 MHz).

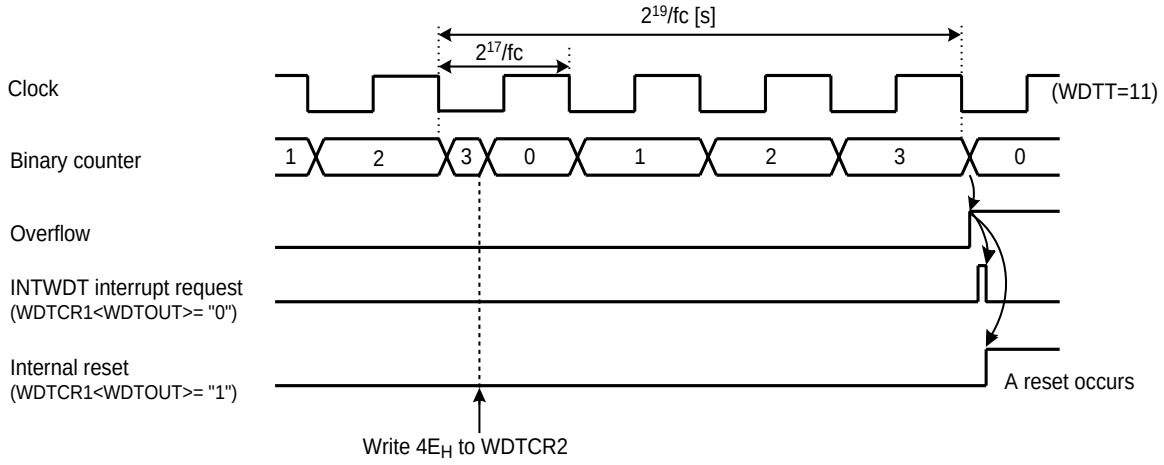


Figure 6-2 Watchdog Timer Interrupt/Reset

6.3 Address Trap

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

Watchdog Timer Control Register 1

WDTCR1 (0034H)	7	6	5	4	3	2	1	0
			ATAS	ATOUT	(WDTEN)	(WDTT)		(WDTOUT)

(Initial value: **11 1001)

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

Watchdog Timer Control Register 2

WDTCR2 (0035H)	7	6	5	4	3	2	1	0

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

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

6.3.1 Selection of Address Trap in Internal RAM (ATAS)

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

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

6.3.2 Selection of Operation at Address Trap (ATOUT)

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

6.3.3 Address Trap Interrupt (INTATRAP)

When a binary-counter overflow occurs during WDTCR1<ATOUT> set to "0", an address trap interrupt request (INTATRAP) is generated.

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

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

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

6.3.4 Address Trap Reset

While WDTCR1<ATOUT> is “1”, if the CPU should start looping for some cause such as noise and attempt be made to fetch an instruction from the on-chip RAM (while WDTCR1<ATAS> is “1”) or the SFR area, address trap reset will be generated. When an address trap reset request is generated, the internal hardware is reset. The reset time is maximum 24/fc [s] (3.0 μs @ fc = 8.0 MHz).

7. Time Base Timer (TBT)

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

7.1 Time Base Timer

7.1.1 Configuration

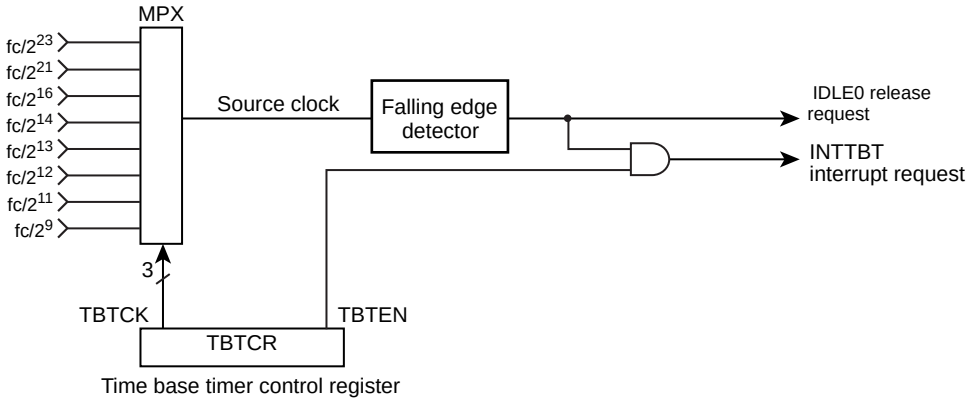


Figure 7-1 Time Base Timer configuration

7.1.2 Control

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

Time Base Timer Control Register

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

TBTEN	Time Base Timer enable / disable	0: Disable 1: Enable		
TBTCK	Time Base Timer interrupt Frequency select : [Hz]		NORMAL1 Mode	R/W
		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 care

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

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

```
LD      (TBTCR) ,  00000010B        ; TBTCK ← 010
LD      (TBTCR) ,  00001010B        ; TBTEN ← 1
DI                                  ; IMF ← 0
SET     (EIRL) . 6
EI
```

Table 7-1 Time Base Timer Interrupt Frequency (Example : fc = 8.0 MHz)

TBTCK	Time Base Timer Interrupt Frequency [Hz]
	NORMAL1 Mode
000	0.95
001	3.81
010	122.07
011	488.28
100	976.56
101	1953.12
110	3906.25
111	15625

7.1.3 Function

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

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

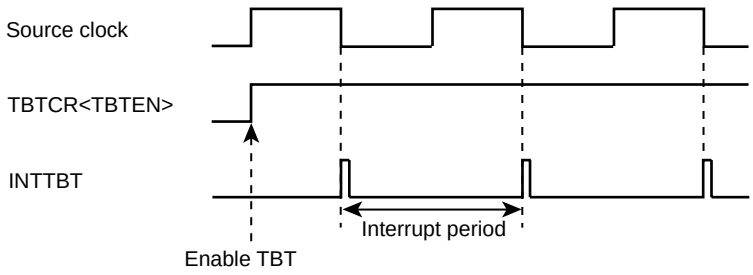


Figure 7-2 Time Base Timer Interrupt

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

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

7.2.1 Configuration

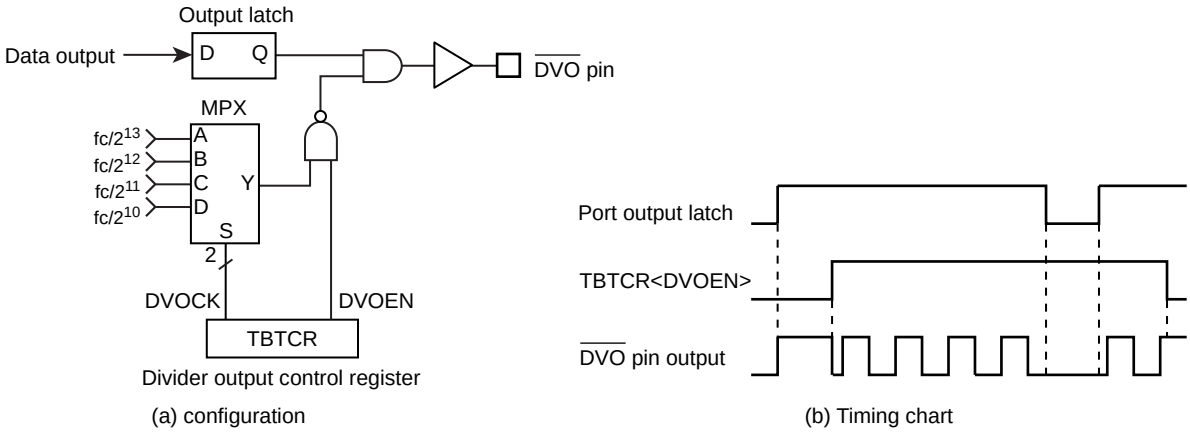


Figure 7-3 Divider Output

7.2.2 Control

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

Time Base Timer Control Register

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

DVOEN	Divider output enable / disable	0: Disable 1: Enable		R/W
DVOCK	Divider Output ($\overline{DVO}$) frequency selection: [Hz]		NORMAL1 Mode	R/W
		00	$fc/2^{13}$	
		01	$fc/2^{12}$	
		10	$fc/2^{11}$	
		11	$fc/2^{10}$	

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

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

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

Table 7-2 Divider Output Frequency (Example : fc = 8.0 MHz)

DVOCK	Divider Output Frequency [Hz]
	NORMAL1 Mode
00	0.977 k
01	1.953 k
10	3.906 k
11	7.813 k

8. 8-Bit TimerCounter (TC3, TC4)

8.1 Configuration

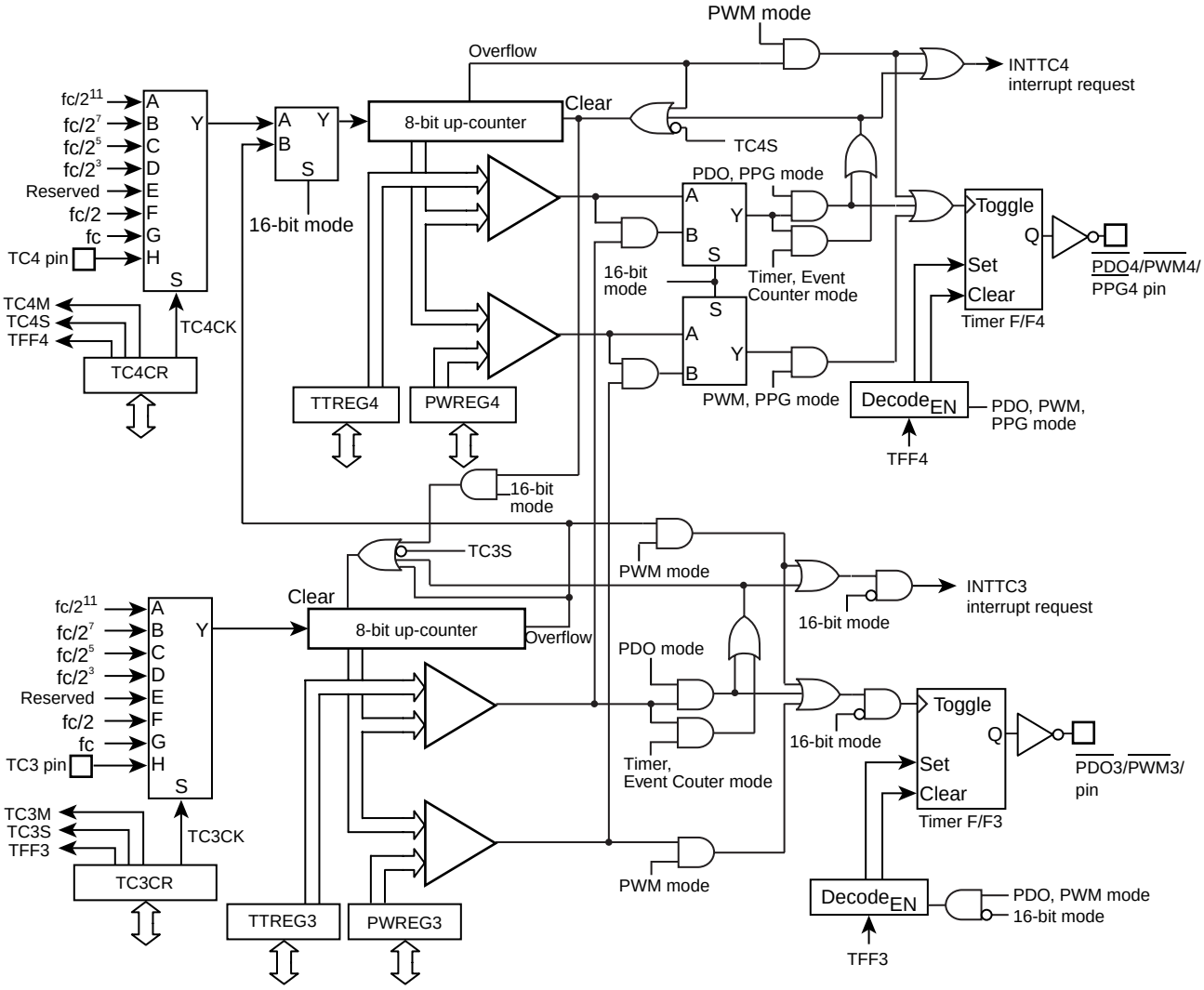


Figure 8-1 8-Bit TimerCouter 3, 4

8.2 TimerCounter Control

The TimerCounter 3 is controlled by the TimerCounter 3 control register (TC3CR) and two 8-bit timer registers (TTREG3, PWREG3).

TimerCounter 3 Timer Register

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

Note 1: Do not change the timer register (TTREG3) setting while the timer is running.

Note 2: Do not change the timer register (PWREG3) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

TimerCounter 3 Control Register

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

TFF3	Time F/F3 control	0: Clear 1: Set		R/W
TC3CK	Operating clock selection [Hz]	NORMAL1, IDLE1 mode		R/W
		000	$fc/2^{11}$	
		001	$fc/2^{7}$	
		010	$fc/2^{5}$	
		011	$fc/2^{3}$	
		100	Reserved	
		101	fc/2	
		110	fc	
		111	TC3 pin input	
TC3S	TC3 start control	0: Operation stop and counter clear 1: Operation start		R/W
TC3M	TC3M operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) output mode 011: 16-bit mode (Each mode is selectable with TC4M.) 1**: Reserved		R/W

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

Note 2: Do not change the TC3M, TC3CK and TFF3 settings while the timer is running.

Note 3: To stop the timer operation (TC3S= 1 → 0), do not change the TC3M, TC3CK and TFF3 settings. To start the timer operation (TC3S= 0 → 1), TC3M, TC3CK and TFF3 can be programmed.

Note 4: To use the TimerCounter in the 16-bit mode, set the operating mode by programming TC4CR<TC4M>, where TC3M must be fixed to 011.

Note 5: To use the TimerCounter in the 16-bit mode, select the source clock by programming TC3CK. Set the timer start control and timer F/F control by programming TC4CR<TC4S> and TC4CR<TFF4>, respectively.

Note 6: The operating clock settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 8-1.

Note 7: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 8-2.

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

TimerCounter 4 Timer Register

	7	6	5	4	3	2	1	0	
TTREG4 (001DH) R/W									(Initial value: 1111 1111)
PWREG4 (001FH) R/W									(Initial value: 1111 1111)

Note 1: Do not change the timer register (TTREG4) setting while the timer is running.

Note 2: Do not change the timer register (PWREG4) setting in the operating mode except the 8-bit and 16-bit PWM modes while the timer is running.

TimerCounter 4 Control Register

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

TFF4	Timer F/F4 control	0: Clear 1: Set		R/W
TC4CK	Operating clock selection [Hz]	NORMAL1, IDLE1 mode		R/W
		000	$fc/2^{11}$	
		001	$fc/2^{7}$	
		010	$fc/2^{5}$	
		011	$fc/2^{3}$	
		100	Reserved	
		101	fc/2	
		110	fc	
		111	TC4 pin input	
TC4S	TC4 start control	0: Operation stop and counter clear 1: Operation start		R/W
TC4M	TC4M operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) output mode 011: Reserved 100: 16-bit timer/event counter mode 101: Warm-up counter mode 110: 16-bit pulse width modulation (PWM) output mode 111: 16-bit PPG mode		R/W

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

Note 2: Do not change the TC4M, TC4CK and TFF4 settings while the timer is running.

Note 3: To stop the timer operation (TC4S= 1 → 0), do not change the TC4M, TC4CK and TFF4 settings.
To start the timer operation (TC4S= 0 → 1), TC4M, TC4CK and TFF4 can be programmed.

Note 4: When TC4M= 1** (upper byte in the 16-bit mode), the source clock becomes the TC4 overflow signal regardless of the TC3CK setting.

Note 5: To use the TimerCounter in the 16-bit mode, select the operating mode by programming TC4M, where TC3CR<TC3 M> must be set to 011.

Note 6: To the TimerCounter in the 16-bit mode, select the source clock by programming TC3CR<TC3CK>. Set the timer start control and timer F/F control by programming TC4S and TFF4, respectively.

Note 7: The operating clock settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 8-1.

Note 8: The timer register settings are limited depending on the timer operating mode. For the detailed descriptions, see Table 8-2.

Table 8-1 Operating Mode and Selectable Source Clock (NORMAL1 and IDLE1 Modes)

Operating mode	$fc/2^{11}$	$fc/2^{7}$	$fc/2^{5}$	$fc/2^{3}$	fc/2	fc	TC3 pin input	TC4 pin input
8-bit timer	O	O	O	O	–	–	–	–
8-bit event counter	–	–	–	–	–	–	O	O
8-bit PDO	O	O	O	O	–	–	–	–
8-bit PWM	O	O	O	O	O	O	–	–
16-bit timer	O	O	O	O	–	–	–	–
16-bit event counter	–	–	–	–	–	–	O	–
16-bit PWM	O	O	O	O	O	O	O	–
16-bit PPG	O	O	O	O	–	–	O	–

Note 1: For 16-bit operations (16-bit timer/event counter, warm-up counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bit (TC3CK).

Note 2: O : Available source clock

Table 8-2 Constraints on Register Values Being Compared

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

Note: n = 3 to 4

8.3 Function

The TimerCounter 3 and 4 have the 8-bit timer, 8-bit event counter, 8-bit programmable divider output (PDO), 8-bit pulse width modulation (PWM) output modes. The TimerCounter 3 and 4 (TC3, 4) are cascadable to form a 16-bit timer. The 16-bit timer has the operating modes such as the 16-bit timer, 16-bit pulse width modulation (PWM) output and 16-bit programmable pulse generation (PPG) modes.

8.3.1 8-Bit Timer Mode (TC3 and 4)

In the timer mode, the up-counter counts up using the internal clock. When a match between the up-counter and the timer register j (TTREGj) value is detected, an INTTCj interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting.

Note 1: In the timer mode, fix TCjCR<TFFj> to 0. If not fixed, the $\overline{\text{PDOj, PWMj}}$ and $\overline{\text{PPGj}}$ pins may output pulses.

Note 2: In the timer mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the timer mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

Table 8-3 Internal Source Clock for TimerCounter 3, 4 (Internal Clock)

Source Clock	Resolution	Maximum Setting time
NORMAL1, IDLE1 mode	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	255 μs

Example :Setting the timer mode with source clock $fc/2^7$ Hz and generating an interrupt 160 μs later (TimerCounter4, fc = 8.0 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 the operating cock to fc/2^7, and 8-bit timer mode.
LD      (TC4CR), 00011000B     : Starts TC4.
```

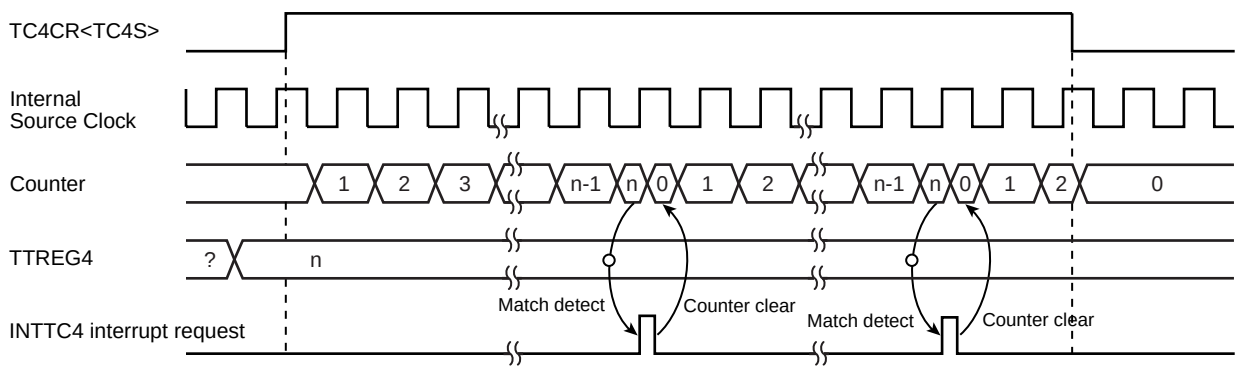


Figure 8-2 8-Bit Timer Mode Timing Chart (TC4)

8.3.2 8-Bit Event Counter Mode (TC3, 4)

In the 8-bit event counter mode, the up-counter counts up at the falling edge of the input pulse to the TCj pin. When a match between the up-counter and the TTREGj value is detected, an INTTCj interrupt is generated and the up-counter is cleared. After being cleared, the up-counter restarts counting at the falling edge of the input pulse to the TCj pin. Two machine cycles are required for the low- or high-level pulse input to the TCj pin. Therefore, a maximum frequency to be supplied is $fc/2^4$ Hz in the NORMAL1 or IDLE1 mode.

Note 1: In the event counter mode, fix TCjCR<TFFj> to 0. If not fixed, the $\overline{PDOj, PWMj}$ and $\overline{PPGj}$ pins may output pulses.

Note 2: In the event counter mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the event counter mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

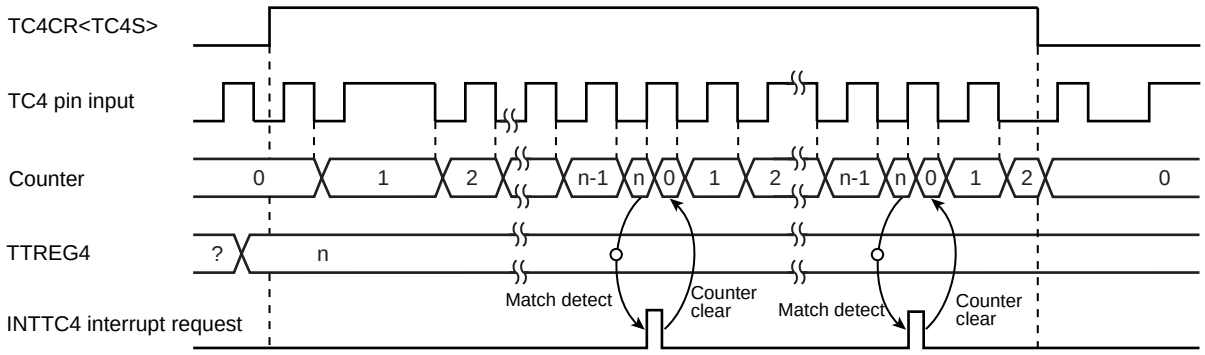


Figure 8-3 8-Bit Event Counter Mode Timing Chart (TC4)

8.3.3 8-Bit Programmable Divider Output (PDO) Mode (TC3, 4)

This mode is used to generate a pulse with a 50% duty cycle from the $\overline{PDOj}$ pin.

In the PDO mode, the up-counter counts up using the internal clock. When a match between the up-counter and the TTREGj value is detected, the logic level output from the $\overline{PDOj}$ pin is switched to the opposite state and the up-counter is cleared. The INTTCj interrupt request is generated at the time. The logic state opposite to the timer F/Fj logic level is output from the $\overline{PDOj}$ pin. An arbitrary value can be set to the timer F/Fj by TCjCR<TFFj>. Upon reset, the timer F/Fj value is initialized to 0.

To use the programmable divider output, set the output latch of the I/O port to 1.

Example :Generating 512 Hz pulse using TC4 (fc = 8.0 MHz)

```
        Setting port
LD      (TTREG4),  3DH          : 1/512÷2^7/fc÷2 = 3DH
LD      (TC4CR),   00010001B    : Sets the operating clock to fc/2^7, and 8-bit PDO mode.
LD      (TC4CR),   00011001B    : Starts TC4.
```

Note 1: In the programmable divider output mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the programmable divider output mode, the new value programmed in TTREGj is in effect immediately after programming. Therefore, if TTREGi is changed while the timer is running, an expected operation may not be obtained.

Note 2: When the timer is stopped during PDO output, the $\overline{PDOj}$ pin holds the output status when the timer is stopped. To change the output status, program TCjCR<TFFj> after the timer is stopped. Do not change the TCjCR<TFFj> setting upon stopping of the timer.
Example: Fixing the $\overline{PDOj}$ pin to the high level when the TimerCounter is stopped
CLR (TCjCR).3: Stops the timer.
CLR (TCjCR).7: Sets the $\overline{PDOj}$ pin to the high level.

Note 3: j = 3, 4

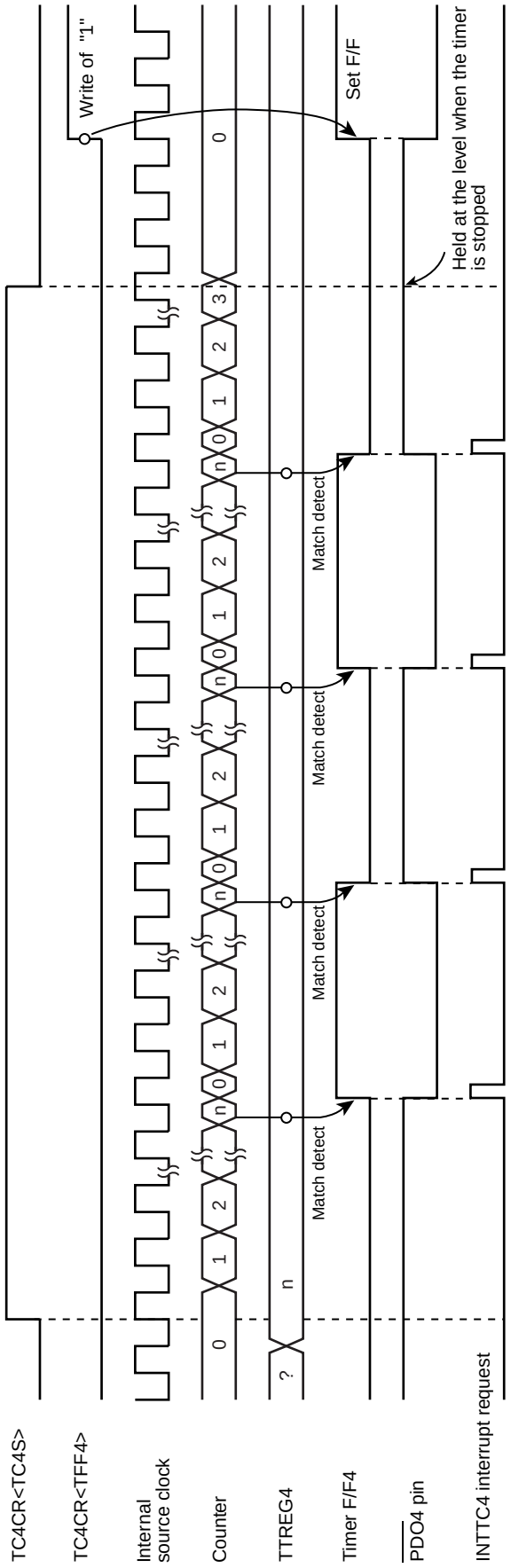


Figure 8-4 8-Bit PDO Mode Timing Chart (TC4)

8.3.4 8-Bit Pulse Width Modulation (PWM) Output Mode (TC3, 4)

This mode is used to generate a pulse-width modulated (PWM) signals with up to 8 bits of resolution. The up-counter counts up using the internal clock.

When a match between the up-counter and the PWREGj value is detected, the logic level output from the timer F/Fj is switched to the opposite state. The counter continues counting. The logic level output from the timer F/Fj is switched to the opposite state again by the up-counter overflow, and the counter is cleared. The INTTCj interrupt request is generated at this time.

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

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

Since PWREGj in the PWM mode is serially connected to the shift register, the value set to PWREGj can be changed while the timer is running. The value set to PWREGj during a run of the timer is shifted by the INTTCj interrupt request and loaded into PWREGj. While the timer is stopped, the value is shifted immediately after the programming of PWREGj. If executing the read instruction to PWREGj during PWM output, the value in the shift register is read, but not the value set in PWREGj. Therefore, after writing to PWREGj, the reading data of PWREGj is previous value until INTTCj is generated.

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

Note 1: In the PWM mode, program the timer register PWREGj immediately after the INTTCj interrupt request is generated (normally in the INTTCj interrupt service routine.) If the programming of PWREGj and the interrupt request occur at the same time, an unstable value is shifted, that may result in generation of the pulse different from the programmed value until the next INTTCj interrupt request is generated.

Note 2: When the timer is stopped during PWM output, the $\overline{\text{PWMj}}$ pin holds the output status when the timer is stopped. To change the output status, program TCjCR<TFFj> after the timer is stopped. Do not change the TCjCR<TFFj> upon stopping of the timer.
Example: Fixing the $\overline{\text{PWMj}}$ pin to the high level when the TimerCounter is stopped
CLR (TCjCR).3: Stops the timer.
CLR (TCjCR).7: Sets the $\overline{\text{PWMj}}$ pin to the high level.

Note 3: To enter the STOP mode during PWM output, stop the timer and then enter the STOP mode. If the STOP mode is entered without stopping the timer when fc or fc/2 is selected as the source clock, a pulse is output from the $\overline{\text{PWMj}}$ pin during the warm-up period time after exiting the STOP mode.

Note 4: j = 3, 4

Table 8-4 PWM Output Mode

Source Clock	Resolution	Repeated Cycle
NORMAL1, IDLE1 mode	fc = 8 MHz	fc = 8MHz
$fc/2^{11}$ [Hz]	256 μs	65.5 ms
$fc/2^{7}$	16 μs	4.1 ms
$fc/2^{5}$	4 μs	1.02 μs
$fc/2^{3}$	1 μs	256 μs
fc/2	250 ns	64 μs
fc	125 ns	32 μs

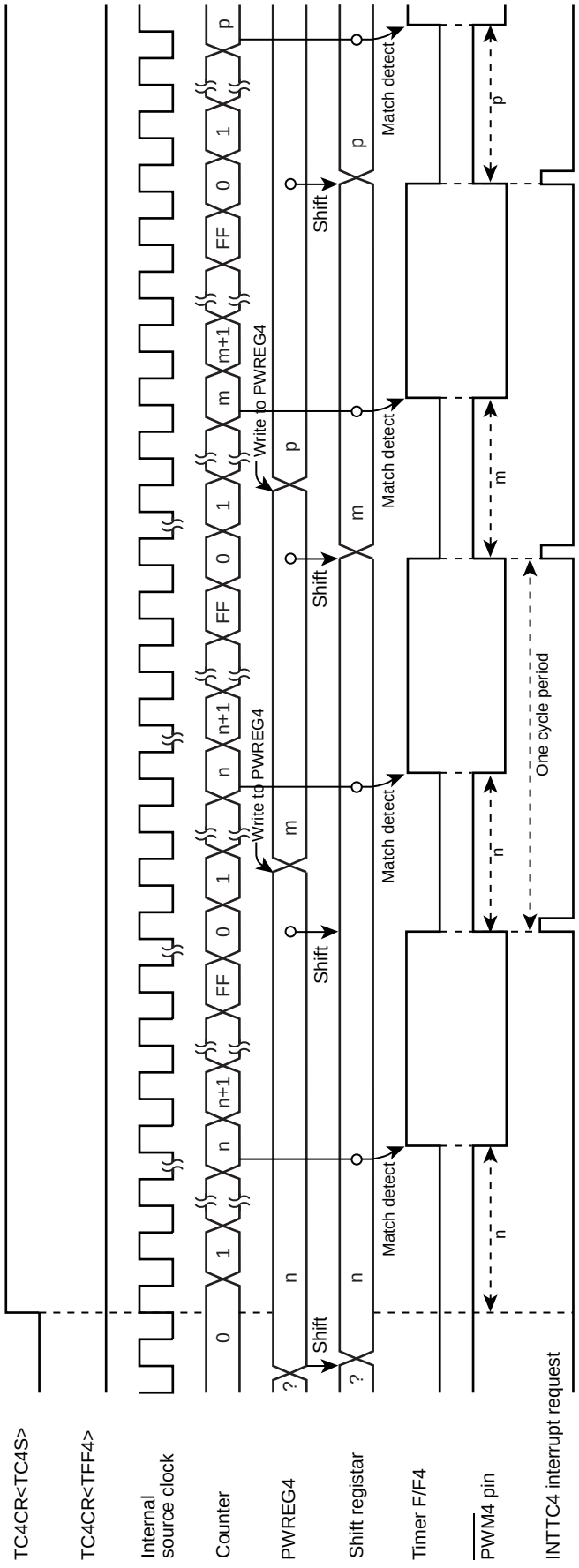


Figure 8-5 8-Bit PWM Mode Timing Chart (TC4)

8.3.5 16-Bit Timer Mode (TC3 and 4)

In the timer mode, the up-counter counts up using the internal clock. The TimerCounter 3 and 4 are cascadable to form a 16-bit timer.

When a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected after the timer is started by setting TC4CR<TC4S> to 1, an INTTC4 interrupt is generated and the up-counter is cleared. After being cleared, the up-counter continues counting. Program the lower byte and upper byte in this order in the timer register. (Programming only the upper or lower byte should not be attempted.)

Note 1: In the timer mode, fix TCjCR<TFFj> to 0. If not fixed, the $\overline{PDOj}$, $\overline{PWMj}$, and $\overline{PPGj}$ pins may output a pulse.

Note 2: In the timer mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the timer mode, the new value programmed in TTREGj is in effect immediately after programming of TTREGj. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.

Note 3: j = 3, 4

Table 8-5 Source Clock for 16-Bit Timer Mode

Source Clock	Resolution	Maximum Setting Time
NORMAL1, IDLE1 mode	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 :Setting the timer mode with source clock $fc/2^7$ Hz, and generating an interrupt 600 ms later (fc = 8.0 MHz)

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

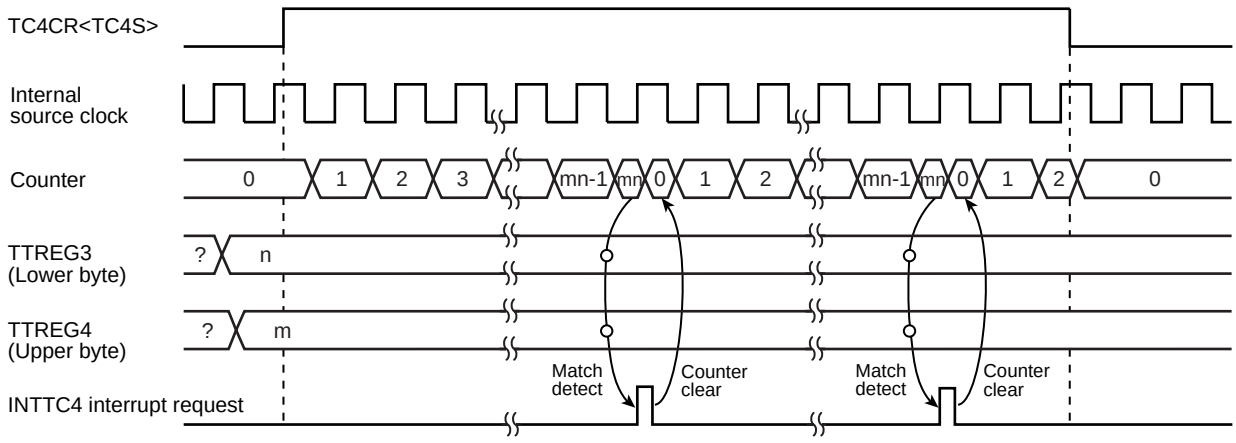


Figure 8-6 16-Bit Timer Mode Timing Chart (TC3 and TC4)

8.3.6 16-Bit Event Counter Mode (TC3 and 4)

In the event counter mode, the up-counter counts up at the falling edge to the TC3 pin. The TimerCounter 3 and 4 are cascadable to form a 16-bit event counter.

When a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected after the timer is started by setting TC4CR<TC4S> to 1, an INTTC4 interrupt is generated and the up-counter is cleared.

After being cleared, the up-counter restarts counting at the falling edge of the input pulse to the TC3 pin. Two machine cycles are required for the low- or high-level pulse input to the TC3 pin.

Therefore, a maximum frequency to be supplied is $fc/2^4$ Hz in the NORMAL1 or IDLE1 mode. Program the lower byte (TTREG3), and upper byte (TTREG4) in this order in the timer register. (Programming only the upper or lower byte should not be attempted.)

Note 1: In the event counter mode, fix TCjCR<TFFj> to 0. If not fixed, the $\overline{PDOj}$, $\overline{PWMj}$ and $\overline{PPGj}$ pins may output pulses.
Note 2: In the event counter mode, do not change the TTREGj setting while the timer is running. Since TTREGj is not in the shift register configuration in the event counter mode, the new value programmed in TTREGj is in effect immediately after the programming. Therefore, if TTREGj is changed while the timer is running, an expected operation may not be obtained.
Note 3: j = 3, 4

8.3.7 16-Bit Pulse Width Modulation (PWM) Output Mode (TC3 and 4)

This mode is used to generate a pulse-width modulated (PWM) signals with up to 16 bits of resolution. The TimerCounter 3 and 4 are cascadable to form the 16-bit PWM signal generator.

The counter counts up using the internal clock or external clock.

When a match between the up-counter and the timer register (PWREG3, PWREG4) value is detected, the logic level output from the timer F/F4 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F4 is switched to the opposite state again by the counter overflow, and the counter is cleared. The INTTC4 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC3 pin. Therefore, a maximum frequency to be supplied is $fc/2^4$ Hz in the NORMAL1 or IDLE1 mode.

Since the initial value can be set to the timer F/F4 by TC4CR<TFF4>, positive and negative pulses can be generated. Upon reset, the timer F/F4 is cleared to 0.

(The logic level output from the $\overline{PWM}4$ pin is the opposite to the timer F/F4 logic level.)

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

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

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

Note 1: In the PWM mode, program the timer register PWREG4 and 3 immediately after the INTTC4 interrupt request is generated (normally in the INTTC4 interrupt service routine.) If the programming of PWREGj and the interrupt request occur at the same time, an unstable value is shifted, that may result in generation of pulse different from the programmed value until the next INTTC4 interrupt request is generated.
Note 2: When the timer is stopped during PWM output, the $\overline{PWM}4$ pin holds the output status when the timer is stopped. To change the output status, program TC4CR<TFF4> after the timer is stopped. Do not program TC4CR<TFF4> upon stopping of the timer.
Example: Fixing the $\overline{PWM}4$ pin to the high level when the TimerCounter is stopped
CLR (TC4CR).3: Stops the timer.
CLR (TC4CR).7 : Sets the $\overline{PWM}4$ pin to the high level.

Note 3: To enter the STOP mode, stop the timer and then enter the STOP mode. If the STOP mode is entered without stopping of the timer when fc or fc/2 is selected as the source clock, a pulse is output from the $\overline{PWM4}$ pin during the warm-up period time after exiting the STOP mode.

Table 8-6 16-Bit PWM Output Mode

Source Clock	Resolution	Repeated Cycle
NORMAL1, IDLE1 mode	fc = 8 MHz	fc = 8 MHz
$fc/2^{11}$	256 μs	16.78 ms
$fc/2^{7}$	16 μs	1.05 ms
$fc/2^{5}$	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 :Generating a pulse with 2-ms high-level width and a period of 65.536 ms (fc = 8.0 MHz)

Setting ports

```
LDW   (PWREG3), 07D0H   : Sets the pulse width.
LD    (TC3CR), 33H      : Sets the operating clock to fc/2^3, and 16-bit PWM output
                          mode (lower byte).
LD    (TC4CR), 056H     : Sets TFF4 to the initial value 0, and 16-bit PWM signal
                          generation mode (upper byte).
LD    (TC4CR), 05EH     : Starts the timer.
```

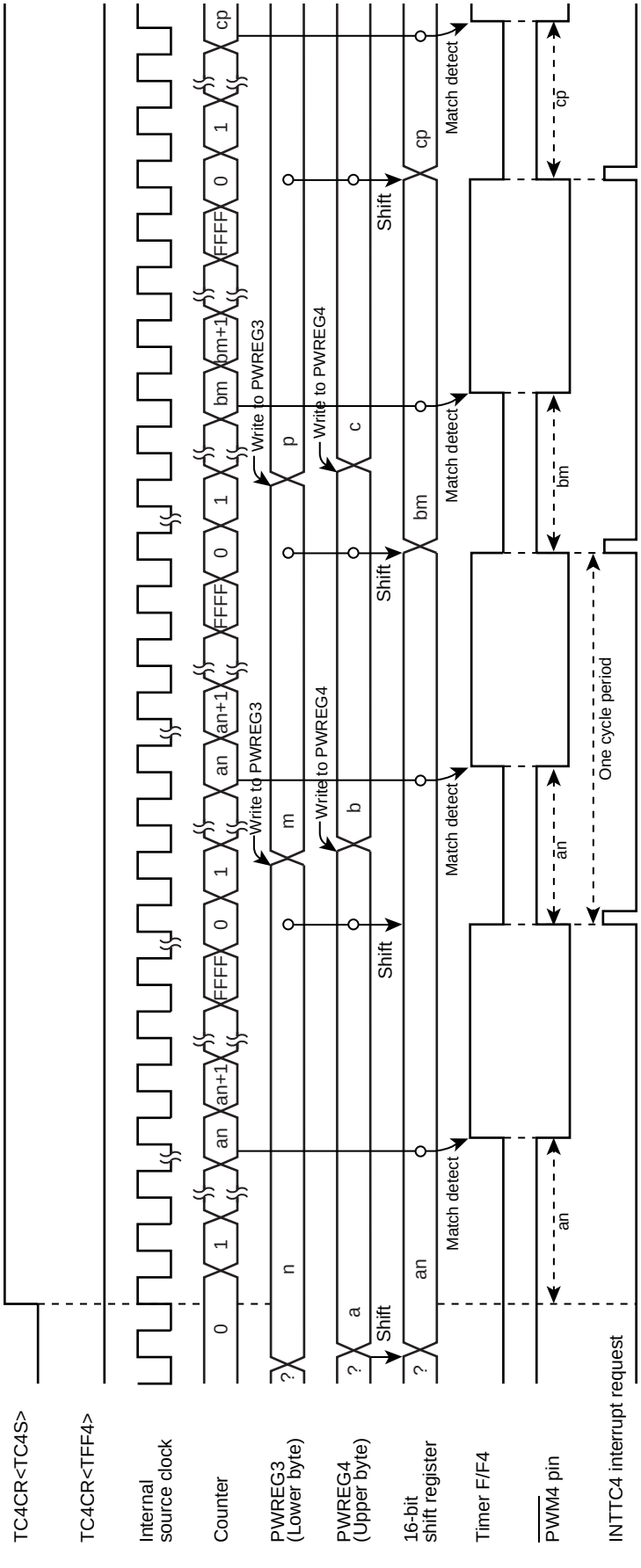


Figure 8-7 16-Bit PWM Mode Timing Chart (TC3 and TC4)

8.3.8 16-Bit Programmable Pulse Generate (PPG) Output Mode (TC3 and 4)

This mode is used to generate pulses with up to 16-bits of resolution. The timer counter 3 and 4 are cascadable to enter the 16-bit PPG mode.

The counter counts up using the internal clock or external clock. When a match between the up-counter and the timer register (PWREG3, PWREG4) value is detected, the logic level output from the timer F/F4 is switched to the opposite state. The counter continues counting. The logic level output from the timer F/F4 is switched to the opposite state again when a match between the up-counter and the timer register (TTREG3, TTREG4) value is detected, and the counter is cleared. The INTTC4 interrupt is generated at this time.

Two machine cycles are required for the high- or low-level pulse input to the TC3 pin. Therefore, a maximum frequency to be supplied is $fc/2^4$ Hz in the NORMAL1 or IDLE1 mode.

Since the initial value can be set to the timer F/F4 by TC4CR<TFF4>, positive and negative pulses can be generated. Upon reset, the timer F/F4 is cleared to 0.

(The logic level output from the $\overline{PPG}4$ pin is the opposite to the timer F/F4.)

Set the lower byte and upper byte in this order to program the timer register. (TTREG3 → TTREG4, PWREG3 → PWREG4) (Programming only the upper or lower byte should not be attempted.)

For PPG output, set the output latch of the I/O port to 1.

Example :Generating a pulse with 2-ms high-level width and a period of 32.770 ms (fc = 8.0 MHz)

```
        Setting ports
LDW     (PWREG3), 07D0H     : Sets the pulse width.
LDW     (TTREG3), 8002H     : Sets the cycle period.
LD      (TC3CR), 33H        : Sets the operating clock to fc/2^3, and16-bit PWM mode
                              (lower byte).
LD      (TC4CR), 057H       : Sets TFF4 to the initial value 0, and 16-bit
                              PWM mode (upper byte).
LD      (TC4CR), 05FH       : Starts the timer.
```

Note 1: In the PPG mode, do not change the PWREGi and TTREGi settings while the timer is running. Since PWREGi and TTREGi are not in the shift register configuration in the PPG mode, the new values programmed in PWREGi and TTREGi are in effect immediately after programming PWREGi and TTREGi. Therefore, if PWREGi and TTREGi are changed while the timer is running, an expected operation may not be obtained.

Note 2: When the timer is stopped during PPG output, the $\overline{PPG}4$ pin holds the output status when the timer is stopped. To change the output status, program TC4CR<TFF4> after the timer is stopped. Do not change TC4CR<TFF4> upon stopping of the timer.
Example: Fixing the $\overline{PPG}4$ pin to the high level when the TimerCounter is stopped

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

Note 3: i = 3, 4

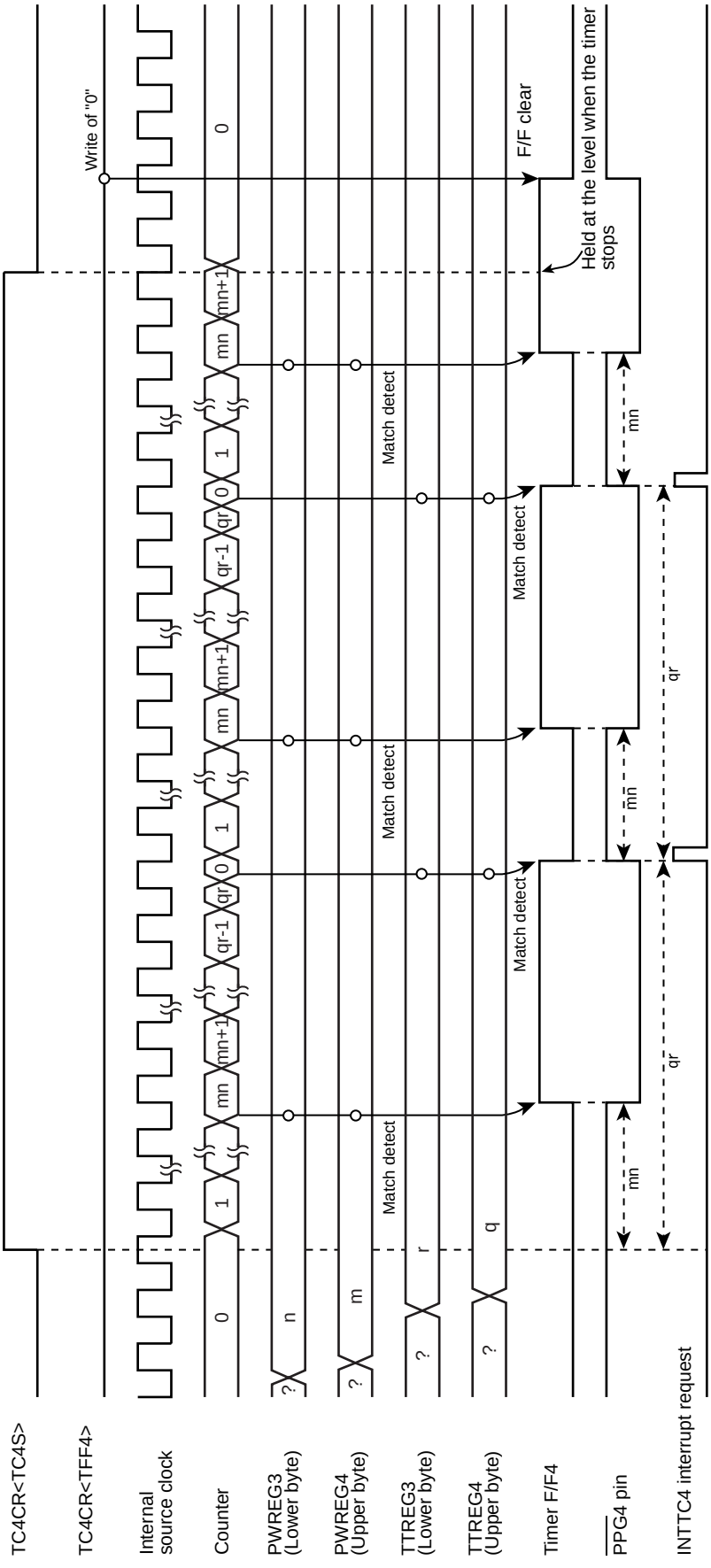


Figure 8-8 16-Bit PPG Mode Timing Chart (TC3 and TC40)

9. 8-Bit AD Converter (ADC)

The TMP86P202MG have a 8-bit successive approximation type AD converter.

Note: AD conversion characteristics are guaranteed with limited supply voltage range (4.5V to 5.5V).

If supply voltage is less than 4.5V then AD conversion accuracy can not be guaranteed.

9.1 Configuration

The circuit configuration of the 8-bit AD converter is shown in Figure 9-1.

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

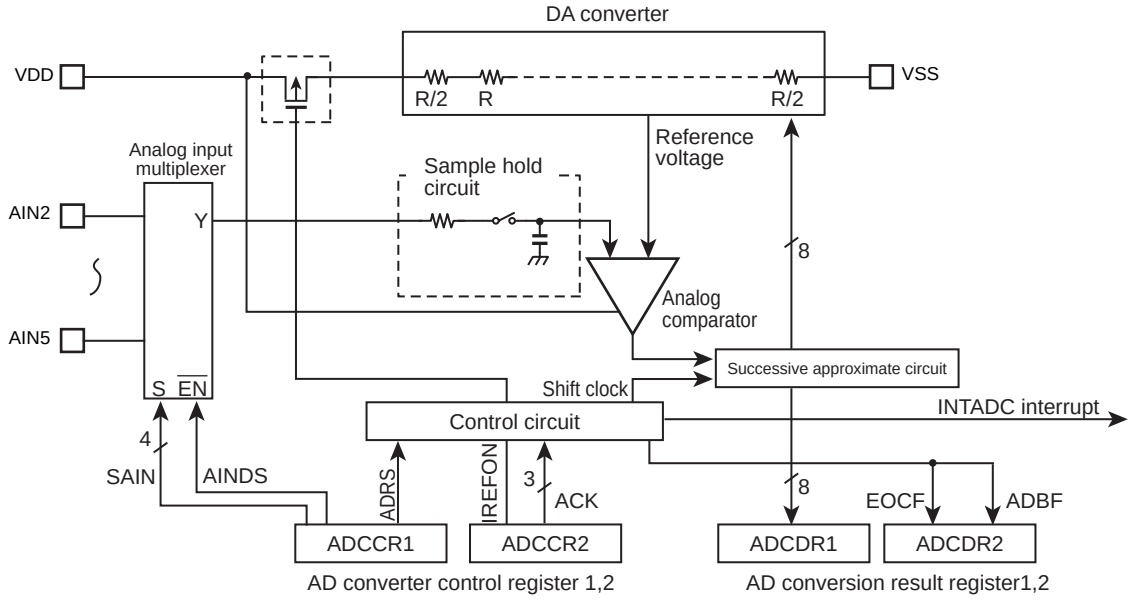


Figure 9-1 8-bit AD Converter (ADC)

9.2 Control

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCR1)

   This register selects the analog channels in which to perform AD conversion and controls the AD converter as it starts operating.

2. AD converter control register 2 (ADCCR2)

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

3. AD converted value register 1 (ADCDR1)

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

4. AD converted value register 2 (ADCDR2)

   This register monitors the operating status of the AD converter.

AD Converter Control Register 1

ADCCR1 (000EH)

7	6	5	4	3	2	1	0
ADRS	"0"	"1"	AINDS	SAIN			

(Initial value: 0001 0000)

ADRS	AD conversion start	0: – 1: Start	R/W
AINDS	Analog input control	0: Analog input enable 1: Analog input disable	
SAIN	Analog input channel select	0000: Reserved 0001: Reserved 0010: AIN2 0011: AIN3 0100: AIN4 0101: AIN5 0110: Reserved 0111: Reserved 1000: Reserved 1001: Reserved 1010: Reserved 1011: Reserved 1100: Reserved 1101: Reserved 1110: Reserved 1111: Reserved	

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

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

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

Note 4: The ADRS is automatically cleared to “0” after starting conversion.

Note 5: Do not set ADCCR1<ADRS> newly again during AD conversion. Before setting ADCCR1<ADRS> newly again, check ADCDR2<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

Note 6: After STOP mode is started, AD converter control register 1 (ADCCR1) is all initialized and no data can be written in this register. Therefore, to use AD converter again, set the ADCCR1 newly after returning to NORMAL1 or NORMAL2 mode.

Note 7: Although ADCCR1<SAIN> is initialized to "Reserved value" after reset, set the suitable analog input channel when using AD converter.

Note 8: Always set bit 5 in ADCCR1 to “1” and set bit 6 in ADCCR1 to “0”.

AD Converter Control Register 2

ADCCR2 (000FH)

7	6	5	4	3	2	1	0
		IREFON	"1"	ACK			"0"

(Initial value: **0* 000*)

IREFON	DA converter (ladder resistor) connection control	0: Connected only during AD conversion 1: Always connected	R/W
ACK	AD conversion time select	000: Reserved 001: Reserved 010: 78/fc 011: 156/fc 100: 312/fc 101: 624/fc 110: 1248/fc 111: Reserved	R/W

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

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

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

Table 9-1 ACK Setting and Conversion Time

Condition / ACK	Conversion time	8MHz	4 MHz	2 MHz
000	Reserved			
001	Reserved			
010	78/fc	-	19.5 μs	39.0 μs
011	156/fc	19.5 μs	39.0 μs	78.0 μs
100	312/fc	39.0 μs	78.0 μs	156.0 μs
101	624/fc	78.0 μs	156.0 μs	-
110	1248/fc	156.0 μs	-	-
111	Reserved			

Note 1: Settings for "–" in the above table are inhibited. fc: High-frequency clock [Hz]

Note 2: Set conversion time by Supply Voltage(VDD) as follows.

- VDD = 4.5 to 5.5 V (15.6 μs or more)

AD Converted Value Register1

ADCDR1 (0020H)

7	6	5	4	3	2	1	0
AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00

(Initial value: 0000 0000)

AD Converted Value Register2

ADCDR2 (0021H)

7	6	5	4	3	2	1	0
		EOCF	ADBF				

(Initial value: **00 ****)

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

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

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

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

9.3 Function

9.3.1 AD Converter Operation

When ADCCR1<ADRS> is set to "1", AD conversion of the voltage at the analog input pin specified by ADCCR1<SAIN> is thereby started.

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

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

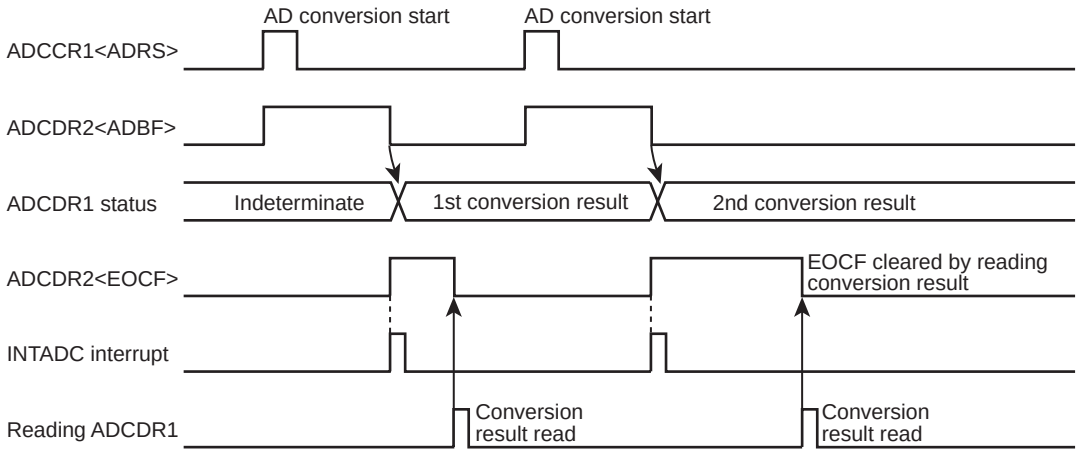


Figure 9-2 AD Converter Operation

9.3.2 AD Converter Operation

1. Set up the AD converter control register 1 (ADCCR1) as follows:
   - Choose the channel to AD convert using AD input channel select (SAIN).
   - Specify analog input enable for analog input control (AINDS).
2. Set up the AD converter control register 2 (ADCCR2) as follows:
   - Set the AD conversion time using AD conversion time (ACK). For details on how to set the conversion time, refer to Table 9-1.
   - Choose IREFON for DA converter control.
3. After setting up 1. and 2. above, set AD conversion start (ADRS) of AD converter control register 1 (ADCCR1) to “1”.
4. After an elapse of the specified AD conversion time, the AD converted value is stored in AD converted value register 1 (ADCDR1) and the AD conversion finished flag (EOCF) of AD converted value register 2 (ADCDR2) is set to “1”, upon which time AD conversion interrupt INTADC is generated.
5. EOCF is cleared to “0” by a read of the conversion result. However, if reconverted before a register read, although EOCF is cleared the previous conversion result is retained until the next conversion is completed.

Example :After selecting the conversion time of 39.0 μs at 8.0 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, read the converted value and store the 8-bit data in address 009FH on RAM.

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

9.3.3 STOP Mode during AD Conversion

When the STOPmode is entered forcibly during AD conversion, the AD convert operation is suspended and the AD converter is initialized (ADCCR1 and ADCCR2 are initialized to initial value.). Also, the conversion result is indeterminate. (Conversion results up to the previous operation are cleared, so be sure to read the conversion results before entering STOPmode.) When restored from STOPmode, AD conversion is not automatically restarted, so it is necessary to restart AD conversion. Note that the DA converter (Ladder resistor) is automatically disconnect.

9.3.4 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 8-bit digital value converted by the AD as shown in Figure 9-3.

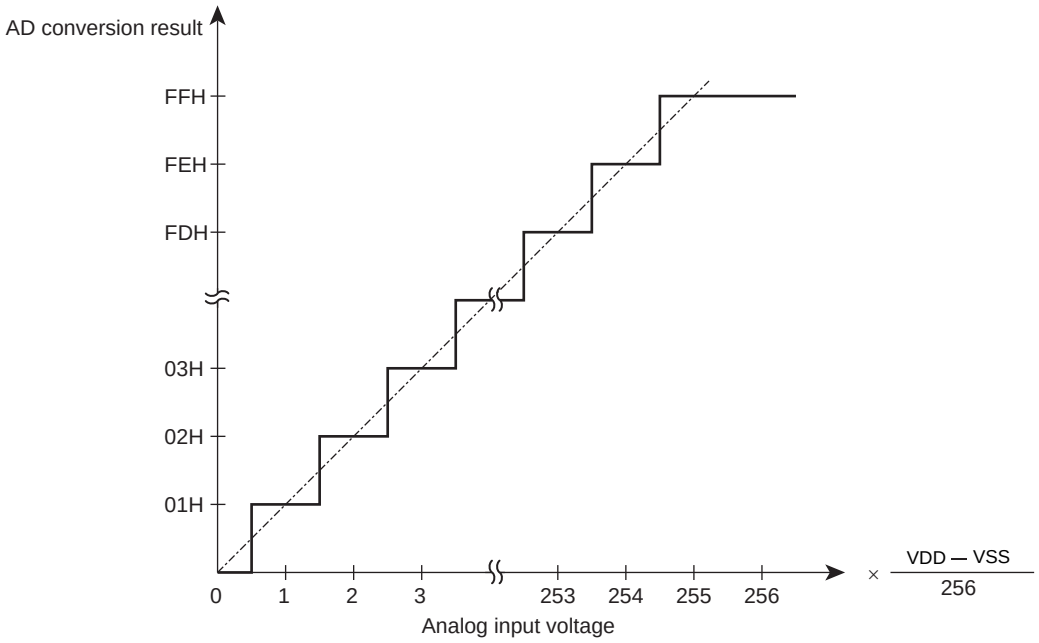


Figure 9-3 Analog Input Voltage and AD Conversion Result (typ.)

9.4 Precautions about AD Converter

9.4.1 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.

9.4.2 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.

9.4.3 Noise countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 9-4. The higher the output impedance of the analog input source, more easily they are susceptible to noise. Therefore, make sure the output impedance of the signal source in your design is 5 kΩ or less. Toshiba also recommends attaching a capacitor external to the chip.

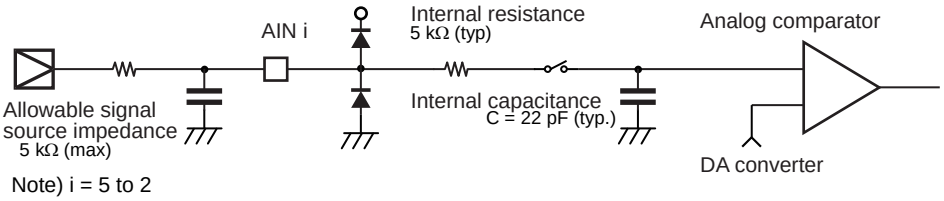


Figure 9-4 Analog Input Equivalent Circuit and Example of Input Pin Processing

10. OTP operation

10.1 Operating mode

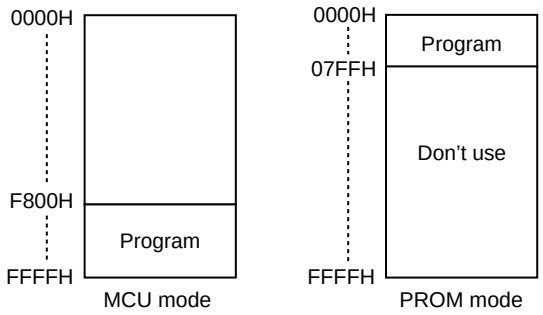
The TMP86P202MG has MCU mode and PROM mode.

10.1.1 MCU mode

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

10.1.1.1 Program Memory

The TMP86P202MG has 2K bytes built-in one-time-PROM (addresses F800 to FFFFH in the MCU mode, addresses 0000 to 07FFH in the PROM mode).



(a) ROM size = 2 Kbytes

Figure 10-1 Program Memory Area

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

10.1.1.2 Data Memory

TMP86P202MG has a built-in 128 bytes Data memory (static RAM).

10.1.2 PROM mode

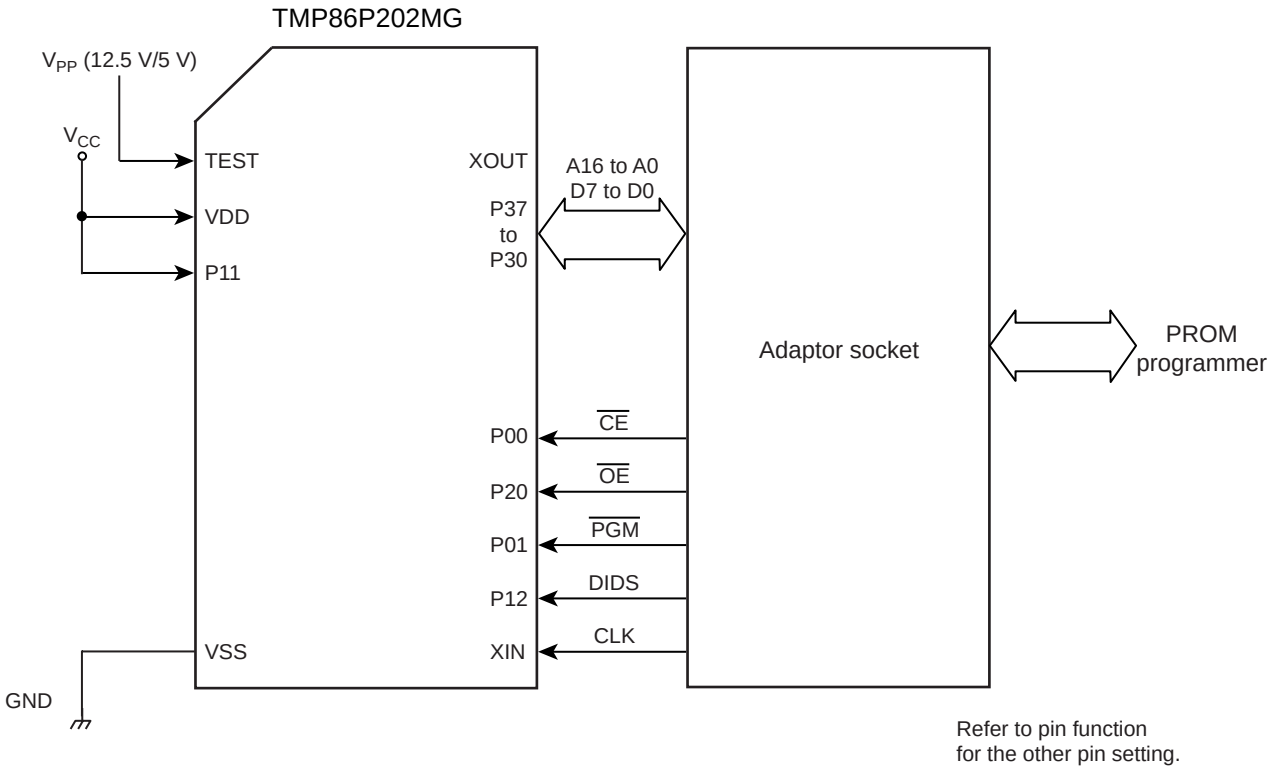
The PROM mode is set by setting the $\overline{\text{RESET}}$ pin, TEST pin and other pins as shown in Table 10-1 and Figure 10-2. The programming and verification for the internal PROM is acheived by using a general-purpose PROM programmer with the adaptor socket.

Table 10-1 Pin name in PROM mode

Pin name (PROM mode)	I/O	Function	Pin name (MCU mode)
A16	Input	Program memory address input	XOUT
A15 to A8	Input	Program memory address input	P37 to P30
A7 to A0	Input	Program memory address input	P37 to P30
D7 to D0	Input/Output	Program memory data input/output	P37 to P30
$\overline{CE}$	Input	Chip enable signal input	P00
$\overline{OE}$	Input	Output enable signal input	P20
$\overline{PGM}$	Input	Program mode signal input	P01
DIDS	Input	PROM mode control signal input	P12
VPP	Power supply	+12.75V/5V (Power supply of program)	TEST
VCC	Power supply	+6.25V/5V	VDD
GND	Power supply	0V	VSS
VCC	Input	Fix to "H" level in PROM mode	P11
$\overline{RESET}$	Input	Fix to "L" level in PROM mode	$\overline{RESET}$
CLK	Input	Input a clock from the outside	XIN

Note 1: The high-speed program mode can be used. The setting is different according to the type of PROM programmer to use, refer to each description of PROM programmer.
TMP86P202MG does not support the electric signature mode, apply the ROM type of PROM programmer to TC571000D/AD.

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



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

Note 2: PROM programmer connection adaptor sockets
BM11704 for TMP86P202MG

Note 3: The pin names written inside frame are for TMP86P202MG.
The pin names written outside frame except DIDS and CLK are for EPROM.

Figure 10-2 PROM mode setting

10.1.2.1 Programming Flowchart (High-speed program writing)

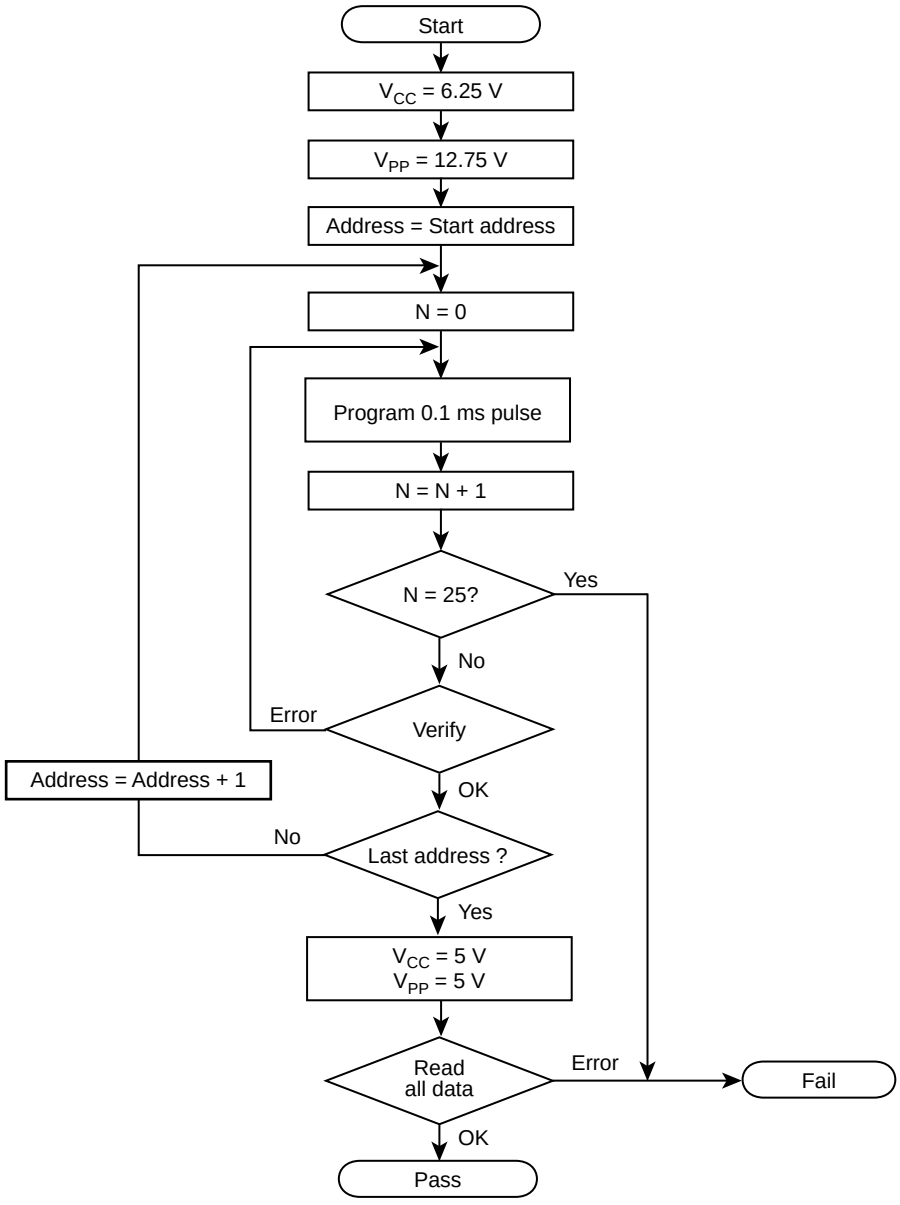


Figure 10-3 Programming Flowchart

The high-speed programming mode is set by applying Vpp=12.75V (programming voltage) to the Vpp pin when the Vcc = 6.25 V. After the address and data are fixed, the data in the address is written by applying 0.1[msec] of low level program pulse to $\overline{\text{PGM}}$ pin. Then verify if the data is written.

If the programmed data is incorrect, another 0.1[msec] pulse is applied to $\overline{\text{PGM}}$ pin. This programming procedure is repeated until correct data is read from the address (maximum of 25 times).

Subsequently, all data are programmed in all address. When all data were written, verfy all address under the condition Vcc=Vpp=5V.

10.1.2.2 Program Writing using a General-purpose PROM Programmer

(1) Recommended OTP adaptor

BM11704 for TMP86P202MG

(2) Setting of OTP adaptor

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

(3) Setting of PROM programmer

a. Set PROM type to TC571000D/AD.

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

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

The PROM of TMP86P202MG is located on different address; it depends on operating mode: MCU mode and PROM mode. When you write the data of ROM for mask ROM products, the data shuold be transferred (or copied) from the address for MCU mode to that for PROM mode before writing operation is executed. For the applicable program areas of MCU mode and PROM mode are different, refer to TMP86P202MG" Figure 10-1 Program Memory Area ".

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

2KB ROM capacity : 0F800~0FFFFH → 00000~007FFH

c. Setting of the program address (Note 1)

Start address: 0000H

End address: 07FFH

(4) Writting

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

(5) Security bit

The TMP86P202MG 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 (FFH data) in PROM mode.

How to program the security bit

The difference from the programming procedures described in section 10.1.2.2 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 1101H. Set the start address 1101H and the end address 1101H. Set the data FEH at the address 1101H.

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

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

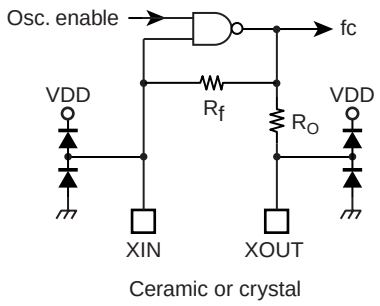
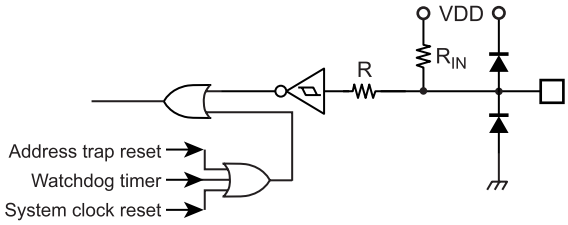
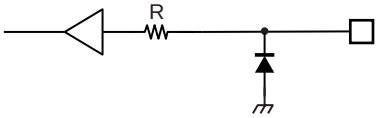
Note 3: The TMP86P202MG does not support the electric signature mode.
If PROM programmer uses the signature, the device would be damaged because of applying voltage of 12±0.5V to pin 9(A9) of the address. Don't use the signature.

Note 4: Do not alter the contents of register at 1101H after programming the security bit to 0.

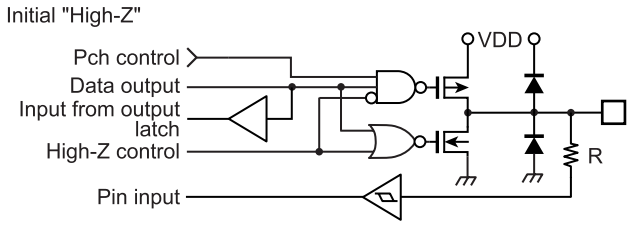
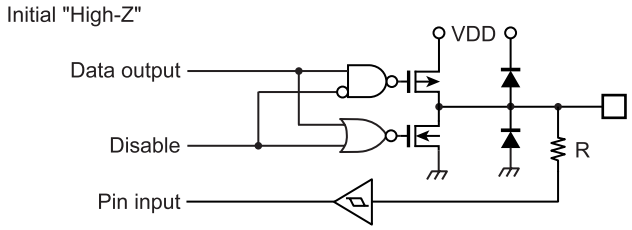
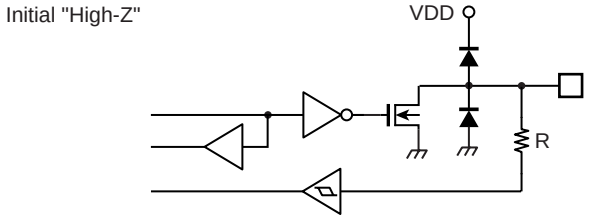
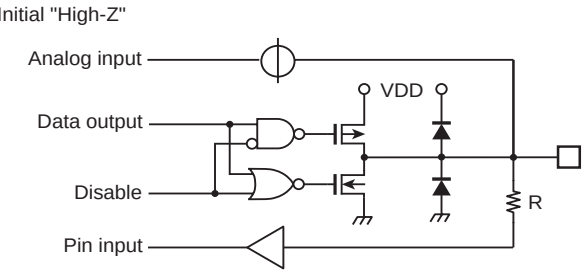
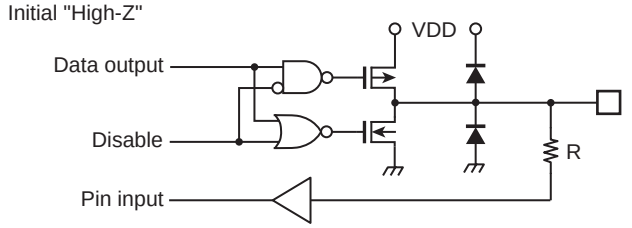
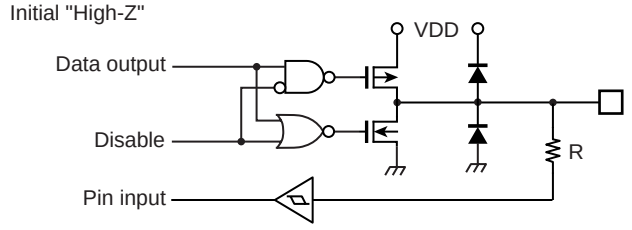
11. Input/Output Circuitry

11.1 Control Pins

The input/output circuitries of the TMP86P202MG control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remarks
XIN XOUT	XIN XOUT		Resonator connecting pins (High-frequency) R_f = 1.2 MΩ (typ.) R_O = 0.5 kΩ (typ.)
$\overline{\text{RESET}}$	Input		Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.) R = 1 kΩ (typ.)
TEST	Input		R = 1 kΩ (typ.) Fix the TEST pin at low-level in MCU mode.

11.2 Input/Output Ports

Port	I/O	Input/Output Circuitry	Remarks
P0	I/O		Sink open drain output or Push-pull output Hysteresis input High current output (Nch) (Programmable port option) R = 100 Ω (typ.)
P1	I/O		Tri-state I/O Hysteresis input R = 100 Ω (typ.)
P2	I/O		Sink open drain output Hysteresis input R = 100 Ω (typ.)
P37 P36 P35 P34	I/O		Tri-state I/O R = 100 Ω (typ.)
P33 P32	I/O		Tri-state I/O R = 100 Ω (typ.)
P31 P30	I/O		Tri-state I/O Hysteresis input R = 100 Ω (typ.)

Note: Input staturs 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.

12. Electrical Characteristics

12.1 Absolute Maximum Ratings

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

(V_{SS} = 0 V)

Parameter	Symbol	Pins	Ratings	Unit
Supply voltage	V_{DD}		−0.3 to 6.5	V
Program voltage	V_{PP}	TEST/V_{PP}	−0.3 to 13.0	
Input voltage	V_{IN}		−0.3 to V_{DD} + 0.3	
Output voltage	V_{OUT}		−0.3 to V_{DD} + 0.3	
Output current (Per 1 pin)	I_{OUT1}	P0, P1, P3 port	−1.8	mA
	I_{OUT2}	P1, P2, P3 port	12	
	I_{OUT3}	P0 port	30	
Output current (Total)	ΣI_{OUT1}	P0, P1, P3 port	−12	
	ΣI_{OUT2}	P1, P2, P3 port	40	
	ΣI_{OUT3}	P0 port	60	
Power dissipation [Topr = 85°C]	P_D	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	

12.2 Operating Condition

The Operating Conditions show the conditions under which the device be used in order for it to operate normally while maintaining its quality. If the device is used outside the range of Operating Conditions (power supply voltage, operating temperature range, or AC/DC rated values), it may operate erratically. Therefore, when designing your application equipment, always make sure its intended working conditions will not exceed the range of Operating Conditions.

(V_{SS} = 0 V, Topr = –40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Max	Unit
Supply voltage	V_{DD}		NORMAL1 mode	3.3	5.5	V
			IDLE0, 1 mode			
			STOP mode	2.0		
Input high level	V_{IH1}	Except hysteresis input	$V_{DD} \geq 4.5$ V	$V_{DD} \times 0.70$	V_{DD}	V
	V_{IH2}	Hysteresis input		$V_{DD} \times 0.75$		
	V_{IH3}		$V_{DD} < 4.5$ V	$V_{DD} \times 0.90$		
Input low level	V_{IL1}	Except hysteresis input	$V_{DD} \geq 4.5$ V	0	$V_{DD} \times 0.30$	
	V_{IL2}	Hysteresis input			$V_{DD} \times 0.25$	
	V_{IL3}		$V_{DD} < 4.5$ V		$V_{DD} \times 0.10$	
Clock frequency	fc	XIN, XOUT	V_{DD} = 3.3 V to 5.5 V	1.0	8.0	MHz

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.

12.3 DC Characteristics

(V_{SS} = 0 V, Topr = –40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Hysteresis voltage	V_{HS}	Hysteresis input		–	0.9	–	V
Input current	I_{IN1}	TEST	V_{DD} = 5.5 V, V_{IN} = 5.5 V/0 V	–	–	±2	μA
	I_{IN2}	Sink open drain, Tri-state port					
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$					
Input resistance	R_{IN}	$\overline{RESET}$ 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 V/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.

12.4 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 section of “Control” of the chapter "8-bit AD Converter".

Note 3: Please use input voltage to AIN input Pin in limit of V_{DD} to 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}.

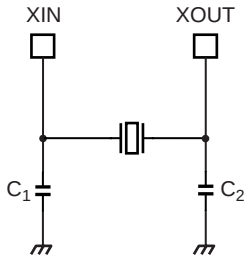
Note 5: AD conversion characteristics are guaranteed with limited supply voltage range 4.5 V to 5.5 V.
If supply voltage is less than 4.5 V then AD conversion accuracy can not be guaranteed.

12.5 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	NORMAL1 mode	0.5	–	4	μs
		IDLE0, 1 mode				
High level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 8 MHz	50	–	–	ns
Low level clock pulse width	t_{WCL}					

12.6 Recommended Oscillation Conditions



Ceramic, Crystal Oscillation

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

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

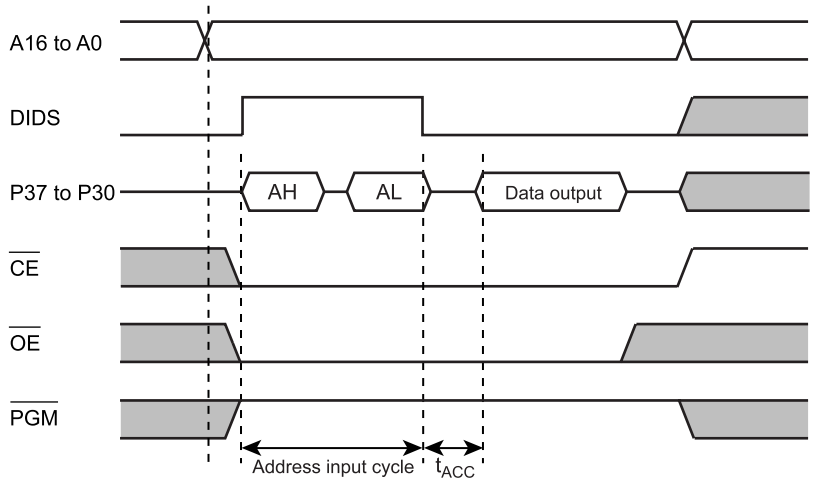
12.7 DC Characteristics, AC Characteristics (PROM mode)

12.7.1 Read operation in PROM mode

(V_{SS} = 0 V, Topr = –40 to 85°C)

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

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



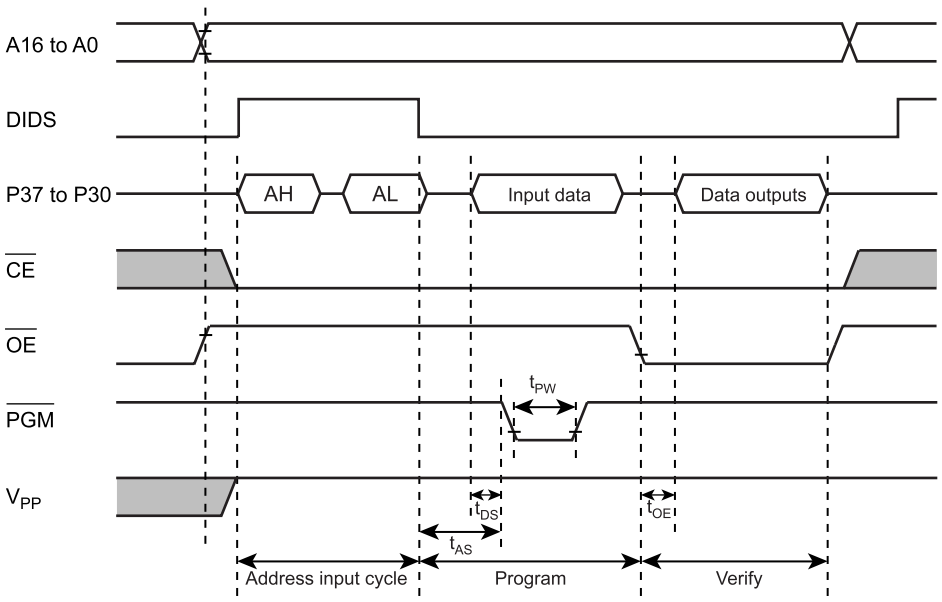
Note: DIDS and P37 to P30 are the signals for the TMP86P202MG.
All other signals are EPROM programmable.
AL: Address input (A0 to A7)
AH: Address input (A8 to A15)

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

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

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

High-speed program writing



Note: DIDS and P37 to P30 are the signals for the TMP86P202MG.
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 and mode.
Using other than the above condition may cause the trouble of the writting.

12.8 Handling Precaution

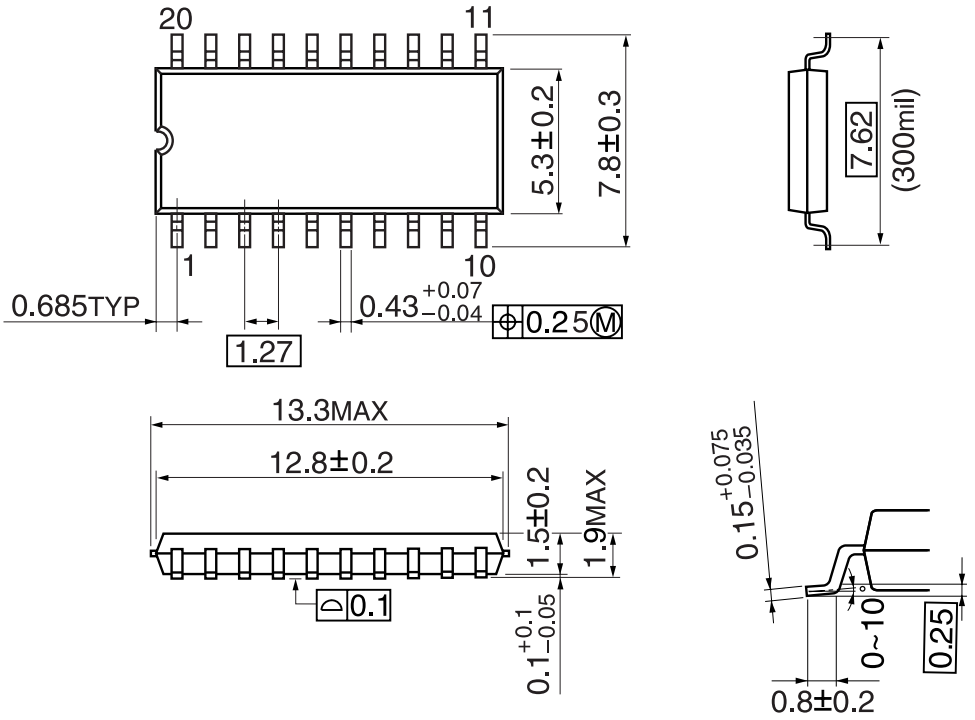
- The solderability test conditions for lead-free products (indicated by the suffix G in product name) are shown below.
  1. When using the Sn-37Pb solder bath
     Solder bath temperature = 230 °C
     Dipping time = 5 seconds
     Number of times = once
     R-type flux used
  2. When using the Sn-3.0Ag-0.5Cu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

  Note: The pass criteron of the above test is as follows:
  Solderability rate until forming ≥ 95 %
- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.

13. Package Dimensions

SOP20-P-300-1.27 Rev 01

Unit: mm



13. Package Dimensions

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

Toshiba provides a variety of development tools and basic software to enable efficient software development.

These development tools have specifications that support advances in microcomputer hardware (LSI) and can be used extensively. Both the hardware and software are supported continuously with version updates.

The recent advances in CMOS LSI production technology have been phenomenal and microcomputer systems for LSI design are constantly being improved. The products described in this document may also be revised in the future. Be sure to check the latest specifications before using.

Toshiba is developing highly integrated, high-performance microcomputers using advanced MOS production technology and especially well proven CMOS technology.

We are prepared to meet the requests for custom packaging for a variety of application areas.

We are confident that our products can satisfy your application needs now and in the future.