

T O S H I B A

TMP88CS43F data sheet (tentativeness version)

2002-2-25 R E V 0 . 2

Note 1) This product is underdevelopment, and there is the case that specification / a function is changed without a notice.

Note 2) This data sheet is tentativeness version, and there is a case changed in future. Toshiba Corporation disclaims all responsibilities for problems that may result from this data sheet. No part of this publication may be reproduced or distributed without the prior written permission of Toshiba Corporation.

Revisions in Data Books

Interrupts

Function	Page	Before revision	Notes	Revised
Interrupt Control (Interrupt latches, Interrupt enable register)	P25	Before you set EE, be sure to clear IMF (to disable interrupt).	Before you change each enable flag (EE) and/or each interrupt latch (IL), be sure to clear the interrupt master enable flag (IMF) to "0" (to disable interrupts). a. After a DI instruction is executed b. When an interrupt is accepted, IMF is automatically cleared to "0". However, to enable nested interrupts, change EF and/or IL before setting IMF to "1" (to enable interrupts). If the individual enable flags (EF) and interrupt latches (IL) are set under conditions other than the above, the proper operation cannot be guaranteed.	

Watchdog Timer

Function	Page	Before revision	Notes	Revised
Watchdog Timer	P54	(A note on disabling the watchdog timer is added.)	Just right before disabling the watchdog timer, disable the acceptance of interrupts (DI) and clear the watchdog timer. If the watchdog timer is disabled under conditions other than the above, the proper operation cannot be guaranteed. Example : DI LD (WDTCR2),4EH Clears the watchdog timer. LDW (WDTCR1),B100H Disables the watchdog timer. EI Enables interrupt acceptance.	

I/O Ports

Function	Page	Before revision	Notes	Revised
Port P1	P41	(A note on the P10 pin is added.)	When external memory is used, the P10 pin cannot be used as an external interrupt input (INT0) or an I/O port as CLK is output from this pin.	

AD Converter

Function	Page	Before revision	Notes	Revised
AD Converter (AD converter control registers ADCCR1 ADCCR2)	P158	(A note on the AD control register is added.)	When STOP or SLOW mode is activated, the AD control registers 1 (ADCCR1) and 2 (ADCCR2) are all initialized. After NORMAL mode is resumed, set both the AD control registers 1 and 2 (ADCCR1, ADCCR2) again if necessary.	

visions in Data Books

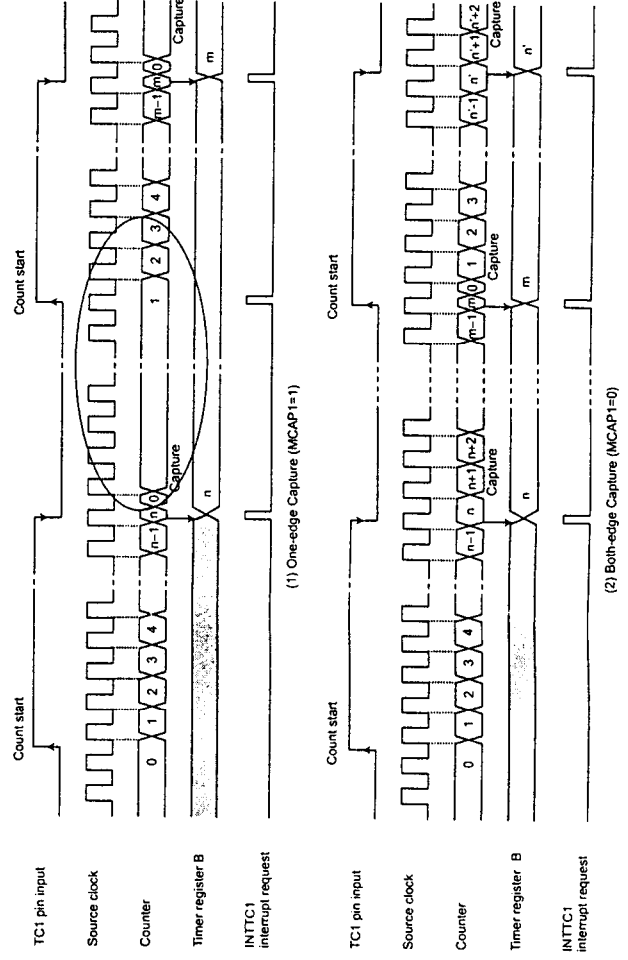
Function	Page	Notes	
		Before revision	Revised
Bit Timer/Counter 1 (TC1)	P58	<p>Note on writing to the timer registers (TC1DRA, TC1DRB) in the figure "Timer Registers and TC1 Control Register"</p> <p>Note: Writing to the timer registers is not executed until the next falling edge of the source clock.</p>	<p>Because the register configuration has been changed, this note is revised as shown below.</p> <p>Note: When writing to the timer registers, write first to the lower byte and then to the upper byte continuously. It is recommended that a 16-bit access instruction be used for write.</p> <p>The timer registers are configured as 2-stage shift registers, and the timer register value becomes valid at the next rising edge of the source clock after data is written to the upper byte (TC1DRA_H, TC1DRB_H). Note that writing only to the lower byte (TC1DRA_L, TC1DRB_L) does not make the register value valid.</p>
Bit Timer/Counter 1 (output mode)	P65	<p>"Programmable Pulse Generate (PPG) output mode"</p> <p>A match between the counter value and the value set in TC1DRB inverts timer F-F1. Continuous output (MPPG=0) generates INTTC1. The next match between the counter value and the value set in TC1DRB inverts timer F-F1 again and clears TC1. At the same time, INTTC1 is generated.</p>	<p>Because INTTC1 is generated by single output as well as by continuous output, this explanation is revised to include operation in the timer stop state.</p> <p>A match between the counter value and the value set in TC1DRB inverts timer F-F1 and at the same time generates INTTC1. The next match between the counter value and the value set in TC1DRA inverts timer F-F1 again, clears the counter, and generates INTTC1. In the case of single output (MPPG=1), TC1CS is automatically cleared to "00" and the timer stops counting. Writing "00" to TC1CS, while the timer is counting, stops timer operation and clears the counter. In this case PPG output keeps the same level just before the timer is stopped.</p>
	P65	<p>"Programmable Pulse Generate (PPG) output mode": F-F1 timer</p> <p>Since the timer F-F1 value can be set in TFF1 (Bit7 in TC1CR), either positive or negative AND pulses can be output.</p>	<p>An additional note on setting timer F-F1 by using TFF1 is included.</p> <p>Timer F-F1 is cleared to "0" during reset. Since the timer F-F1 value can be set in TFF1 (bit7 in TC1CR), either positive or negative pulse can be output.</p> <p>Note: Do not change timer F-F1 by TFF1 while TC1 is operating. The timer F-F1 value can be set or cleared only by initial setting. Setting or clearing the timer F-F1 value, while TC1 is counting or after TC1 stops counting, inverts the output. After TC1 stops, it is necessary to initialize timer F-F1 to change PPG output to a desired level. To initialize F-F1, change to the timer mode once (the timer mode need not be started) and then set to the PPG output mode also set TFF1 again.</p>
	P65	<p>(A note on the initialization of timer F-F1 in the PPG output mode is added.)</p>	<p>Note: To restart the PPG output mode, it is necessary to initialize timer F-F1. To initialize timer F-F1, change to the timer mode once (the timer mode need not be started), then set to the PPG output mode, and also set TFF1 again.</p>
	P65	<p>The section on "Programmable Pulse Generate (PPG) output mode" does not include a note on writing to the timer registers.</p>	<p>The following note on writing to the timer registers in the PPG output mode is added.</p> <p>Note: The timer register change should be done while TC1 is disabled (TC1S=00). When changing the timer register value while TC1 is counting, set a sufficiently larger value than the count value of the up-counter. If a smaller value than the count value of the up-counter is set to the timer register, comparison is not performed and it makes the up counter overflow.</p>

Revisions in Data Books

16-bit Timer/Counter 1 (TC1)

Function	Page	Notés	Revised
16-bit Timer/Counter 1 (Pulse width measurement mode)	P63	The section on "Pulse width measurement mode" does not contain a note on reading the capture value from the timer register B.	A note on reading the capture value is added. Note : The capture value of the timer register B has to be read by a 16-bit access instruction.
	P63	"Pulse width measurement mode": One-edge capture TC1 starts counting triggered by the rising (falling) edge of input to TC1 pin (set start by external trigger in TC1CR). The source clock is the internal clock. At the next falling (rising) edge, the counter value is loaded to TC1DRB and an interrupt is generated. If one-edge capture is set, the counter is cleared.	(1) A note on one-edge capture is added. When start by external trigger is set in TC1CR, the rising (falling) edge of the TC1 pin input triggers TC1 to start counting. (For the source clock, select the internal clock.) At the next falling (rising) edge, the counter value is loaded to TC1DRB and an INTTC1 request is generated. In the one-edge capture operation, the capture values from the 2nd capture onward increase by one compared with the capture value immediately after count start. (See the figure below).
	P63	Explanation of the revised figure	An INTTC1 request is generated at both the rising and falling edges of TC1 input also in the one-edge capture operation.
	P63	"Pulse width measurement mode": Both-edge capture If both-edge capture is set, the counter continues counting; at the next rising (falling) edge, the counter value is loaded to TC1DRB. If a capture value at a falling (rising) edge is required, data in TC1DRB must be read before a rising (falling) edge is detected.	(2) An explanation about both-edge capture is revised, and a note on the pulse width measurement mode is added. If both-edge capture is set, the counter continues counting and the counter value is loaded to TC1DRB again at the next rising (falling) edge. Note : Be sure to read the capture value from the timer register B before the next trigger edge is detected. If the capture value cannot be read, it becomes undefined.

Timing Chart for the Pulse Width Measurement Mode



CMOS 8-Bit Microcomputer

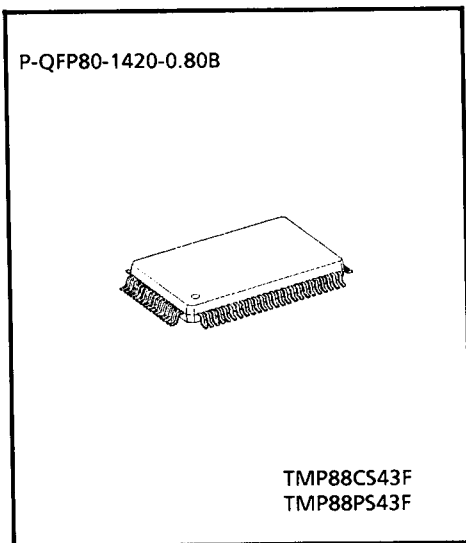
TMP88CS43F

The TMP88CS43F is a high-speed, high-function 8-bit microcomputer built around the TLCS-870/X series CPU core and incorporating sine wave drive PMD (Programmable Motor Drivers: PMD2), as well as a 10-bit AD converter, multifunction timer/counters, and synchronous/asynchronous serial interfaces.

Product Type Name	ROM	RAM	Package	With built-in OTP
TMP88CS43F	64 Kbytes	2 K + 128 bytes	P-QFP80-1420-0.80B	TMP88PS43F

Features

- ◆ 8-bit single-chip microcomputer TLCS-870/X series
- ◆ Minimum instruction execution time: 0.20 μ s
(when operating with 20.0 MHz)
- ◆ Fundamental machine instruction: 181 kinds, 842 instructions
- ◆ Interrupt sources 35 (6 external, 29 internal)
- ◆ Input/output port: 71 pins
 - Large-current output: 24 pins (typ. 20 mA),
capable of LED direct drive
- ◆ Watchdog Timer (WDT)
- ◆ Time Base Timer (TBT)
- ◆ Divider output function (DVO)
- ◆ 16-bit Timer/Counter: 2 channels (TC1, CTC)
 - TC1: Timer, external trigger timer, event counter,
window mode, pulse width measurement,
or PPG1 (Programmable Pulse) output
 - CTC: Timer, event counter,
or PPG2 (Programmable Pulse) output
- ◆ 8-bit Timer/Counter: 4 channels (TC3, TC4, TC5, TC6)
 - TC3: Timer, event counter, or capture
 - TC4: Timer, event counter, PDO (Programmable Divider Output), PWM (Pulse Width Modulation),
or UART baud rate
 - TC5, TC6: Two channels can be cascaded for use as a 16-bit timer Timer, event counter, PWM (Pulse
Width Modulation), PPG (Programmable Pulse) output, or PDO (Programmable Divider
Output)
- ◆ 10-bit successive approximation type AD converter (with sample-and-hold)
 - Analog input: 16 channels



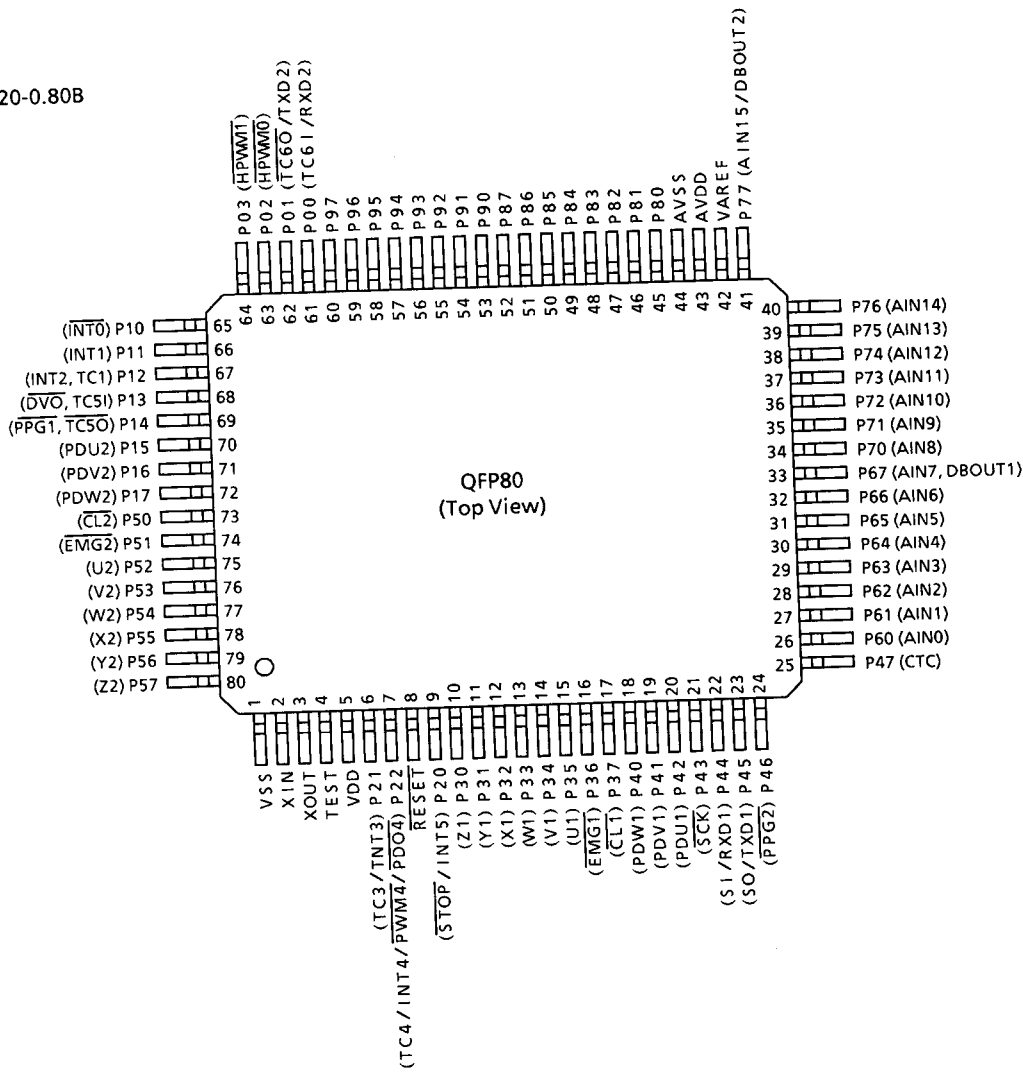
000707EBP1

- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance / Handling Precautions.
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc..
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer's own risk.
- The products described in this document are subject to the foreign exchange and foreign trade laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

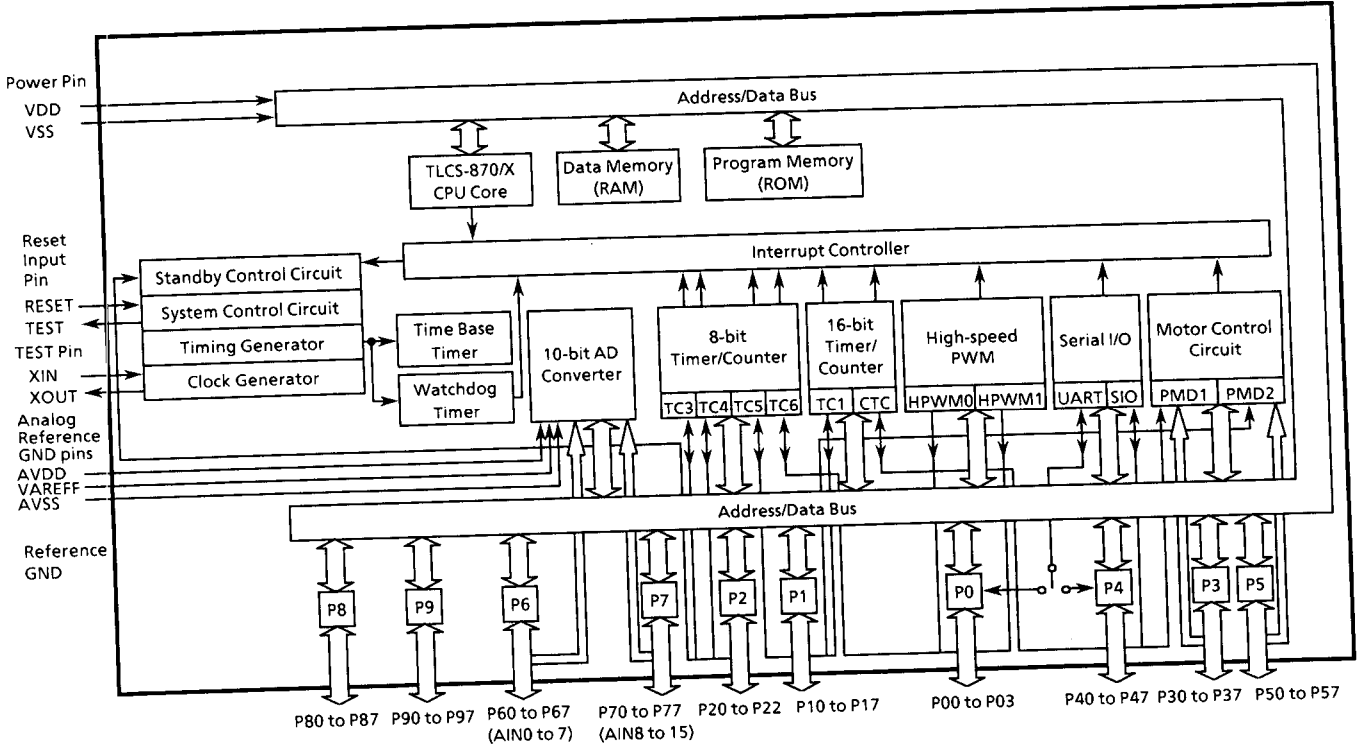
- ◆ Serial interface: Two channels (SIO, UART)
 - 8-bit SIO (synchronous): 1 channel
 - 8-bit UART (asynchronous): 1 channel (selectable pins to use)
- ◆ 8-bit high-speed PWM: 2 channels
- ◆ Programmable motor driver: 2 channels (PMD2)
 - Sine wave drive circuit (built-in sine wave data-only RAM)
 - Rotor position detect function
 - Motor control timer and capture function
 - Overload protective function
 - Auto current, auto position detection start
- ◆ Low power dissipation mode
 - STOP mode: Operation halted (battery)
 - IDLE mode: CPU halted and only peripheral hardware operating, returned normal by an interrupt (CPU restarts)
- ◆ Operating voltage: 4.5 to 5.5 V (at 8 to 20 MHz)

Pin Assignments (Top View)

P-QFP80-1420-0.80B



Block Diagram



Pin Function (1/3)

Pin Name	I/O	Function		
P00 (TC6I, RXD2)	I/O (input, input)	<ul style="list-style-type: none"> 4-bit programmable input/output port. (hysteresis input, tristate output/open-drain output) Can be set for input or output mode bitwise. Can be set for tristate or open-drain output bitwise. When using as timer/counter or UART input, set these pins for input mode. When using as timer/counter, UART, or PWM output, set these pins for output mode. 	Timer/counter input 6, UART input 2	
P01 (TC6O, TXD2)	I/O (output, output)		Timer/counter output 6, UART output 2	
P02 (HPWM0)	I/O (output)		High-speed PWM0 output	
P03 (HPWM1)			High-speed PWM1 output	
P10 (INT0)	I/O (input)	<ul style="list-style-type: none"> 8-bit programmable input/output port. (hysteresis input, tristate output) Can be set for input or output mode bitwise. When using as external interrupt, timer/counter, or position detection input, set these pins for input mode. When using as DVO output or PPG1 output of TC1, set these pins for output mode after setting the output latch to 1. 	External interrupt input 0	
P11 (INT1)			External interrupt input 1	
P12 (INT2, TC1)	I/O (input, input)		External interrupt 2 input or timer/counter 1 input	
P13 (TC5I, DVO)	I/O (input, output)		Timer/counter 5 input or divider output	
P14 (TC5O, PPG)	I/O (output, output)		Timer/counter 5 output or PPG output	
P15 (PDU2)	I/O (input)		Motor control position detection input (U2, V2, W2 phases)	
P16 (PDV2)				
P17 (PDW2)				
P20 (INT5, STOP)	I/O (input, input)		<ul style="list-style-type: none"> 3-bit input/output port. (hysteresis input, open-drain output) When using these pins as timer/counter, external interrupt, or STOP mode exiting input, set the output latch to 1. 	External interrupt 5 input, STOP mode exiting input
P21 (TC3, INT3)				Timer/counter 3 input or external interrupt 3 input
P22 (TC4, INT4, PWM4, PDO4)		Timer/counter 4 input or external interrupt 4 input, PWM4 output, PDO4 output		
P30 (Z1)	I/O (output)	<ul style="list-style-type: none"> 8-bit programmable input/output port. (hysteresis input, tristate output/open-drain output) Can be set for input or output mode bitwise. Can be set for tristate or open-drain output bitwise. Can directly drive LED with large current. When using motor control output, set these pins for output mode. Also, set the output latch to 1. When using error detection/overload protective input, set these pins for input mode. 	Motor control output (Z1, Y1, X1, W1, V1, U1 phases)	
P31 (Y1)				
P32 (X1)				
P33 (W1)				
P34 (V1)				
P35 (U1)				
P36 (EMG1)	I/O (input)	Motor control error detection input 1		
P37 (CL1)	I/O (input)	Motor control overload protective input 1		
P40 (PDW1)	I/O (input)	<ul style="list-style-type: none"> 8-bit programmable input/output port. (hysteresis input, tristate output/open-drain output) Can be set for input or output mode bitwise. Can be set for tristate or open-drain output bitwise. Can directly drive LED with large current. When using timer/counter, SIO, or position detection input, set these pins for input mode. Also, set the output latch to 1. When using SIO or UART output or as PPG2 output of CTC, set these pins for output mode. 	Motor control position detection input (W1, V1, U1 phases)	
P41 (PDV1)			I/O	SIO clock input/output
P42 (PDU1)				SIO input, UART data input 1
P43 (SCK)	I/O		SIO output, UART data output 1	
P44 (SI, RXD1)	I/O (input)			
P45 (SO, TXD1)	I/O (output)		PPG2 output	
P46 (PPG2)	I/O (input)		CTC input	
P47 (CTC)				

Pin Function (2/3)

Pin Name	I/O	Function	
P50 (CL2)	I/O (input)	<ul style="list-style-type: none"> 8-bit programmable input/output port. (hysteresis input, tristate output/open-drain output) Can be set for input or output mode bitwise. Can be set for tristate or open-drain output bitwise. Can directly drive LED with large current. When using motor control output, set these pins for output mode. Also, set the output latch to 1. When using error detection/overload protective input, set these pins for input mode. 	Motor control error detection input 2
P51 (EMG2)			Motor control overload protective input 2
P52 (U2)	I/O (output)		Motor control output (Z2, Y2, X2, W2, V2, U2 phases)
P53 (V2)			
P54 (W2)			
P55 (X2)			
P56 (Y2)			
P57 (Z2)			
P60 (AIN0)	I/O (input)	<ul style="list-style-type: none"> 8-bit programmable input/output port. (tristate output) Can be set for input or output mode bitwise. When using as analog input, set these pins for input mode. Also, set the output latch to 1. When using motor control output, set these pins for output mode. Also, set the output latch to 1. 	AD converter analog input
P61 (AIN1)			
P62 (AIN2)			
P63 (AIN3)			
P64 (AIN4)			
P65 (AIN5)			
P66 (AIN6)			
P67 (AIN7, DBOUT1)			
P70 (AIN8)	I/O (input)	<ul style="list-style-type: none"> 8-bit programmable input/output port. (tristate output) Can be set for input or output mode bitwise. When using as analog input, set these pins for input mode. Also, set the output latch to 1. When using motor control output, set these pins for output mode. Also, set the output latch to 1. 	AD converter analog input
P71 (AIN9)			
P72 (AIN10)			
P73 (AIN11)			
P74 (AIN12)			
P75 (AIN13)			
P76 (AIN14)			
P77 (AIN15, DBOUT2)			
P80	I/O	<ul style="list-style-type: none"> 8-bit programmable input/output port. (tristate output/open-drain output) Can be set for input or output mode bitwise. Can be set for tristate or open-drain output bitwise. 	—
P81			
P82			
P83			
P84			
P85			
P86			
P87			
P90	I/O	<ul style="list-style-type: none"> 8-bit programmable input/output port. (tristate output/open-drain output) Can be set for input or output mode bitwise. Can be set for tristate or open-drain output bitwise. 	—
P91			
P92			
P93			
P94			
P95			
P96			
P97			

Pin Function (3/3)

Pin Name	I/O	Function
TEST	Input	Used for shipping test. Fix this pin low.
$\overline{\text{RESET}}$	Input	Reset signal input (no watchdog timer, address trap, or system clock resets are output from this pin).
XIN	Input	High-frequency resonator connecting pins.
XOUT	Output	When using external clock input, feed it to XIN and leave XOUT open.
VSS	Power Supply	0.0 [V] (GND)
VDD		+ 5.0[V]
AVSS		AD conversion circuit GND
AVDD		AD conversion circuit power supply
VAREF		AD conversion analog reference voltage

Functional Description

1. Functions of the CPU Core

The CPU core consists mainly of the CPU, system clock control circuit, and interrupt control circuit. This chapter describes the CPU core, program memory, data memory, and reset circuit of the TMP88CS43.

1.1 Memory Address Map

The memory of the TMP88CS43 consists of four blocks: ROM, RAM, SFR (Special Function Registers), and DBR (Data Buffer Registers), which are mapped into one 1-Mbyte address space. Figure 1-1 shows a memory address map of the TMP88CS43. The general-purpose registers consist of 16 banks, which are mapped into the RAM address space.

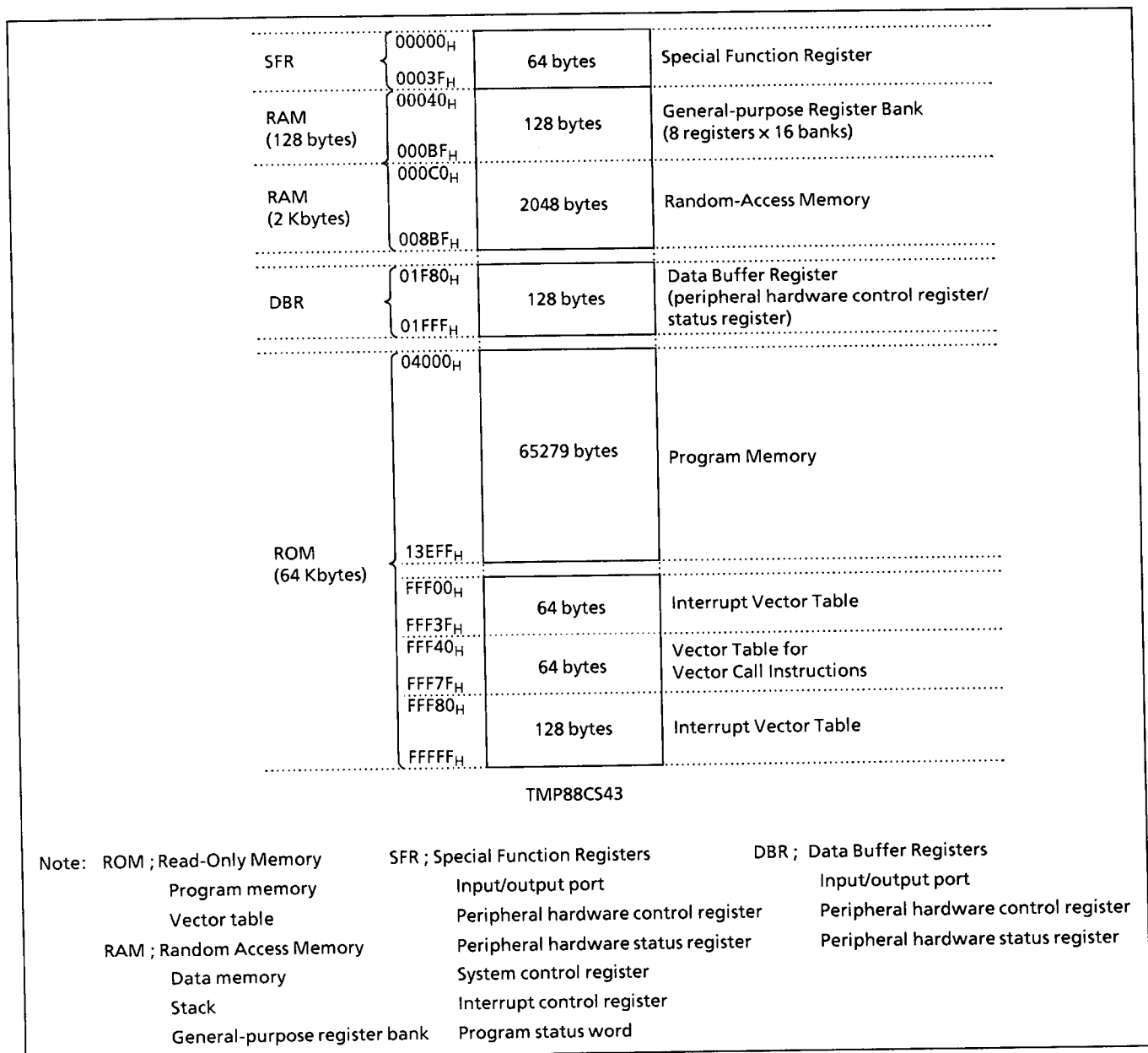


Figure 1-1. Memory address map

1.2 Program Memory (ROM)

The TMP88CS43 contains 64-Kbyte program memory (mask ROM) located at addresses 04000_H to 13EFF_H and addresses FFF00_H to FFFFF_H.

1.3 Data Memory (RAM)

The TMP88CS43 contains 2-Kbyte + 128-byte RAM. The first 128-byte location (00040 to 000BF_H) of the internal RAM is shared with a general-purpose register bank.

The content of the data memory is indeterminate at power-on, so be sure to initialize it in the initialize routine.

Example: Clearing the internal RAM of the TMP88CS43 (clear all RAM addresses to 0, except bank 0)

```
LD      HL, 0048H      ; Set the start address
LD      A, 00H         ; Set the initialization data (00H)
LD      BC, 0F7FH     ; Set byte counts (- 1)
SRAMCLR: LD      (HL+), A
        DEC     BC
        JRS    F, SRAMCLR
```

Note: Because general-purpose registers exist in the RAM, never clear the current bank address of RAM. In the above example, the RAM is cleared except bank 0.

1.4 System Clock Control Circuit

The System Clock Control Circuit consists of a clock generator, timing generator, and standby control circuit.

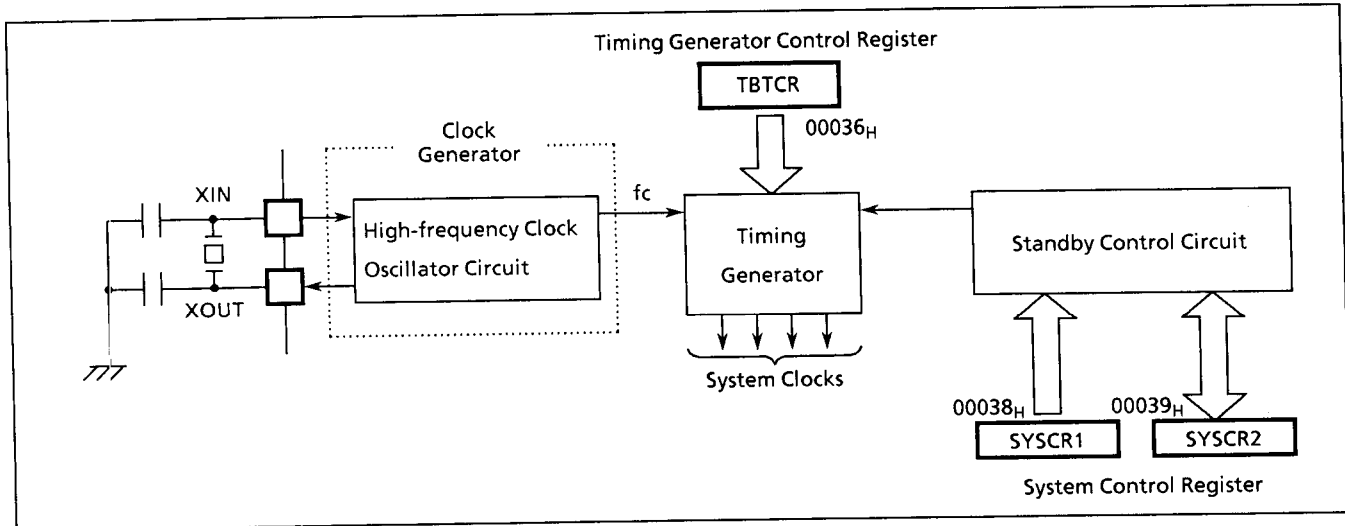


Figure 1-2. System Clock Control Circuit

1.4.1 Clock Generator

The Clock Generator generates the fundamental clock which serves as the reference for the system clocks supplied to the CPU core and peripheral hardware units.

The high-frequency clock (frequency f_c) can be obtained easily by connecting a resonator to the XIN and XOUT pins. Or a clock generated by an external oscillator can also be used. In this case, enter the external clock from the XIN pin and leave the XOUT pin open. The TMP88CS43 does not support the CR network that produces a time constant.

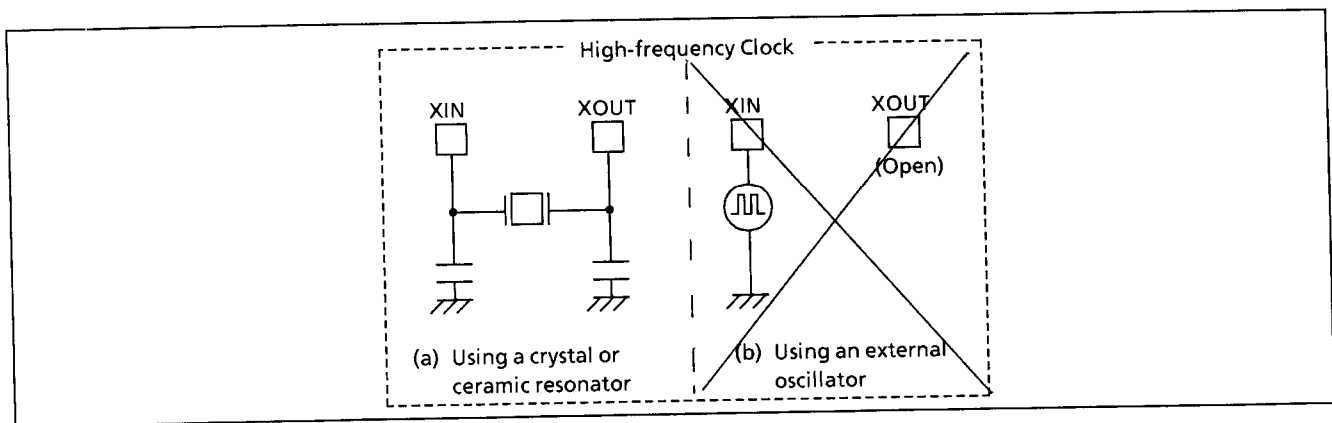


Figure 1-3. Example for Connecting a Resonator

Adjusting the oscillation frequency

Note: Although no hardware functions are provided that allow the fundamental clock to be monitored directly from the outside, the oscillation frequency can be adjusted by forwarding the pulse of a fixed frequency (e.g., clock output) to a port and monitoring it in a program while interrupts and the watchdog timer are disabled. For systems that require adjusting the oscillation frequency, an adjustment program must be created beforehand.

1.4.2 Timing Generator

The Timing Generator generates various system clocks from the fundamental clock that are supplied to the CPU core and peripheral hardware units. The Timing Generator has the following functions:

- ① Generate the main system clock fm
- ② Generate a divider output (\overline{DVO}) pulse
- ③ Generate the source clock for the time base timer
- ④ Generate the source clock for the watchdog timer
- ⑤ Generate the internal source clock for the timer counter
- ⑥ Generate a warming-up clock when exiting STOP mode

(1) Configuration of the Timing Generator

The Timing Generator has a two-stage prescaler, 21-stage dividers, and a machine cycle counter. When reset and when entering/exiting STOP mode, the prescaler and dividers are cleared to 0.

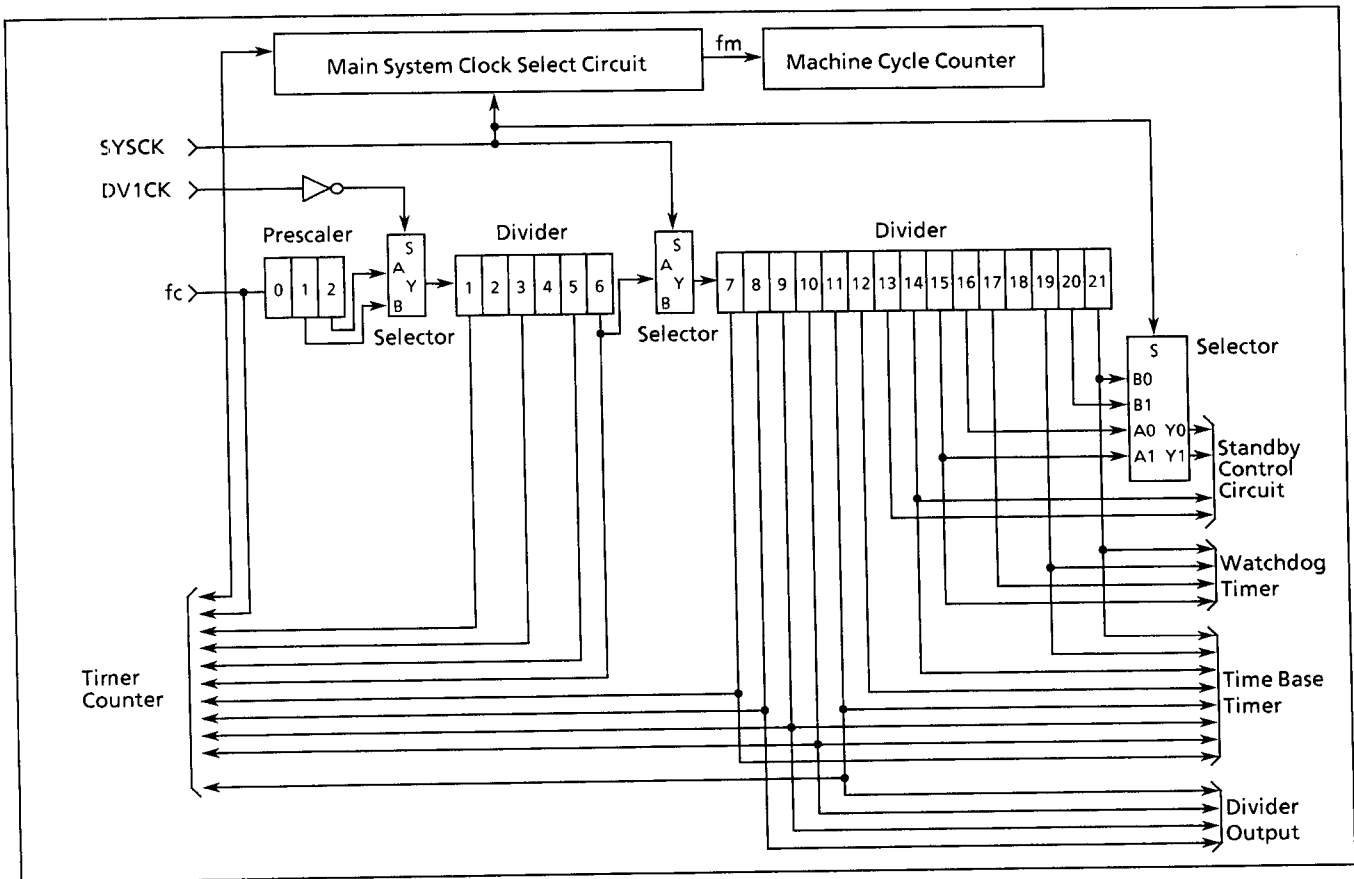


Figure 1-4. Configuration of the Timing Generator

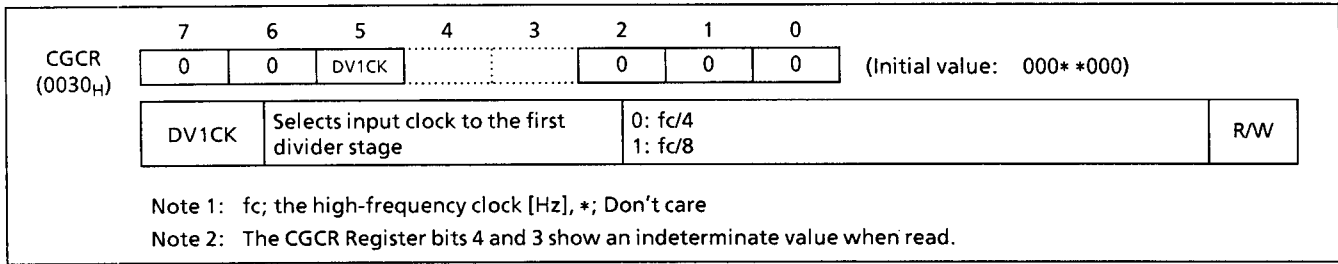


Figure 1-5. Divider Control Register

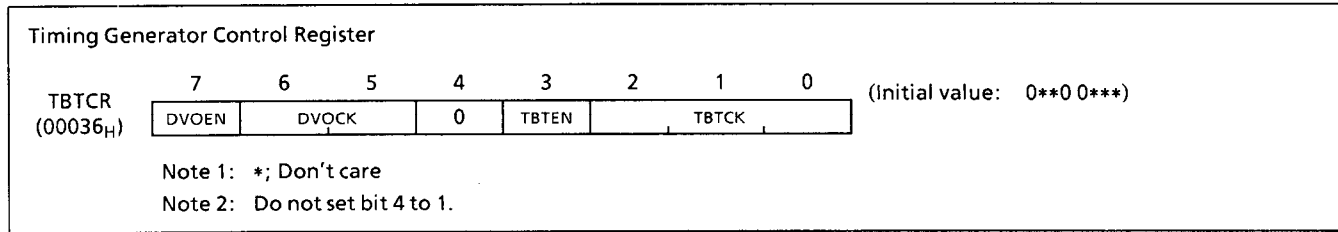


Figure 1-6. Timing Generator Control Register

(2) Machine cycle

Instruction execution and the internal hardware operations are synchronized to the system clocks. The minimum unit of instruction execution is referred to as the "machine cycle." The TLCS-870/X series has 15 types of instructions, from 1-cycle instructions which are executed in one machine cycle up to 15-cycle instructions that require a maximum of 15 machine cycles. A machine cycle consists of four states (S0 to S3), with each state comprised of one main system clock cycle.

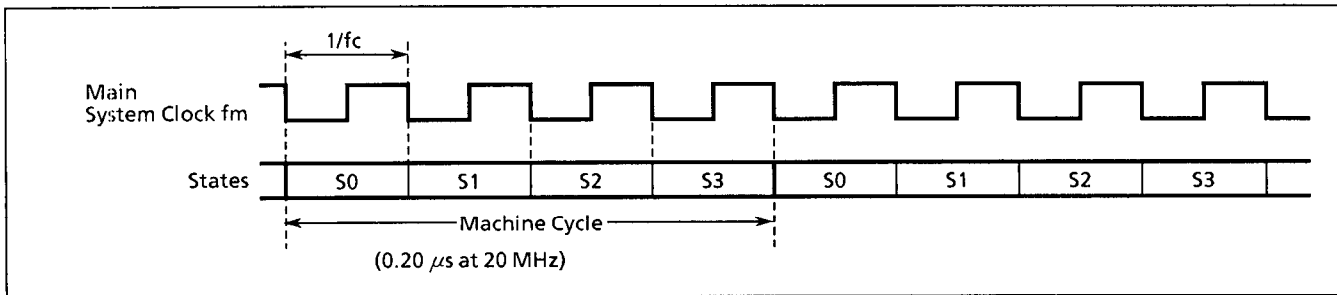


Figure 1-7. Machine Cycles

1.4.3 Standby Control Circuit

The Standby Control Circuit starts/stops the high-frequency clock oscillator circuit and selects the main system clock. The System Control Registers (SYSCR1, SYSCR2) are used to control operation modes of this circuit. Figure 1-8 shows an operation mode transition diagram. Figure 1-9 shows the System Control Registers.

(1) Single clock mode

Only the high-frequency clock oscillator circuit is used. Because the main system clock is generated from the high-frequency clock, the machine cycle time in single clock mode is $4/f_c$ [s].

① NORMAL1 mode

In this mode, the CPU core and peripheral hardware units are operated with the high-frequency clock. The TMP88CS43 enters this NORMAL1 mode after reset.

② IDLE1 mode

In this mode, the CPU and watchdog timer are turned off while the peripheral hardware units are operated with the high-frequency clock. IDLE1 mode is entered into by using System Control Register 2. The device is placed out of this mode and back into NORMAL1 mode by an interrupt from the peripheral hardware or an external interrupt. When IMF (interrupt master enable flag) = 1 (interrupt enabled), the device returns to normal operation after the interrupt has been serviced. When IMF = 0 (interrupt disabled), the device restarts execution beginning with the instruction next to one that placed it in IDLE1 mode.

③ STOP1 mode

The entire system operation including the oscillator circuit is halted, retaining the internal state immediately before being stopped, with a minimal amount of power consumed. STOP1 mode is entered into by using System Control Register 1, and is exited by STOP pin input (level or edge selectable). After an elapse of the warming-up time, the device restarts execution beginning with the instruction next to one that placed it in STOP1 mode.

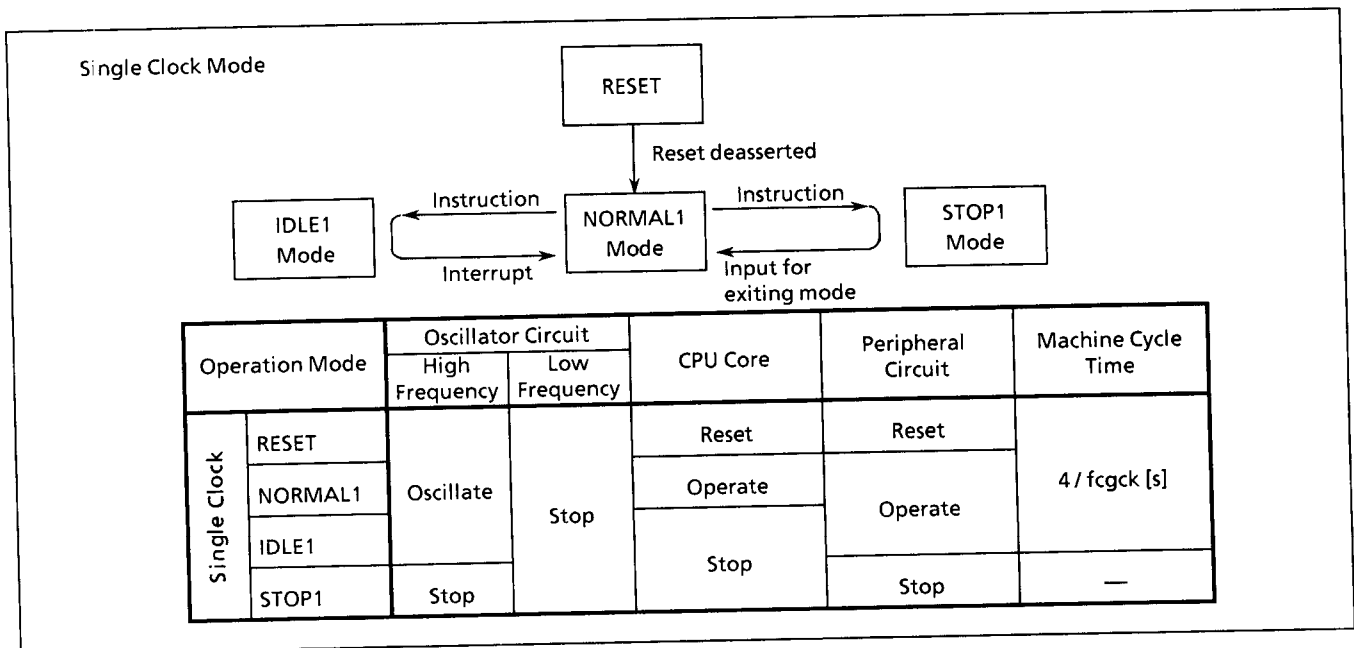


Figure 1-8. Operation Mode Transition Diagram

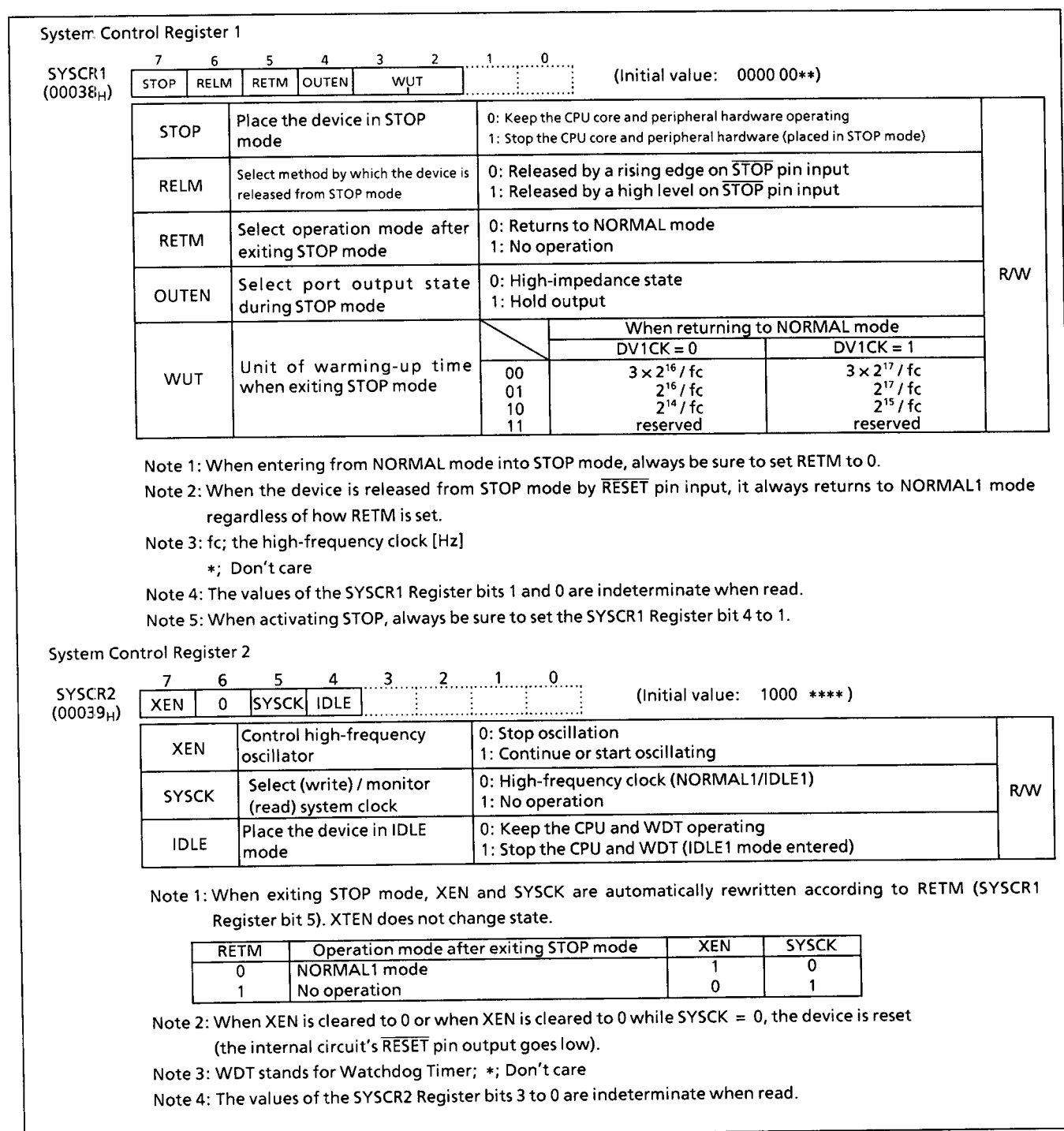


Figure 1-9. System Control Registers 1 and 2

1.4.4 Controlling Operation Modes

(1) STOP mode (STOP1)

STOP mode is controlled by System Control Register 1 (SYSCR1) and the $\overline{\text{STOP}}$ pin input. The $\overline{\text{STOP}}$ pin is shared with P20 port and $\overline{\text{INT5}}$ (external interrupt input 5). STOP mode is entered into by setting STOP (SYSCR1 Register bit 7) to 1. During STOP mode, the device retains the following state.

- ① Stop oscillation, thereby stopping operation of all internal circuits.
- ② The data memory, register, program status word, and port output latch hold the state in which they were immediately before entering STOP mode.
- ③ Clear the prescaler and divider for the timing generator to 0.
- ④ The program counter holds the instruction address two instructions ahead the one that placed the device in STOP mode (e.g., "SET (SYSCR1).7").

The device is released from STOP mode by the active level or edge on STOP pin input as selected by RELM (SYSCR1 Register bit 6).

Note: Before entering STOP mode, be sure to disable interrupts. This is because if the signal on an external interrupt pin changes state during STOP (from entering STOP mode till completion of warming-up) the interrupt latch is set to 1, so that the device may accept the interrupt immediately after exiting STOP mode. Also, when reenabling interrupts after exiting STOP mode, be sure to clear the unnecessary interrupt latches beforehand.

a. Released by level (when RELM = 1)

The device is released from STOP mode by a high level on $\overline{\text{STOP}}$ pin input.

Any instruction to place the device in STOP mode is ignored when executed while $\overline{\text{STOP}}$ pin input level is high, and the device immediately goes to a release sequence (warming-up) without entering STOP mode. Therefore, before STOP mode can be entered while RELM = 1, the $\overline{\text{STOP}}$ pin input must be verified to be low in a program. There are following methods to do this verification.

- ① Testing the port status
- ② INT5 interrupt (interrupt generated at a falling edge on $\overline{\text{INT5}}$ pin input)

Example 1: Entering STOP mode from NORMAL mode by testing P20 port

```
LD      (SYSCR1), 01010000B ; Select to be released from STOP mode
                                by level
SSTOPH : TEST  (P2) . 0      ; Wait until  $\overline{\text{STOP}}$  pin input goes low
JRS     F, SSTOPH
SET     (SYSCR1) . 7        ; Place the device in STOP mode
```

Example 2: Entering STOP mode from NORMAL mode by INT5 interrupt

```

PINT5 : TEST (P2) . 0 ; Do not enter STOP mode if P20 port
        JRS  F, SINT5  input level is high, to eliminate noise
        LD  (SYSCR1), 01010000B ; Select to be released from STOP mode by
                                level
        SET (SYSCR1) . 7 ; Place the device in STOP mode
SINT5 : RETI
    
```

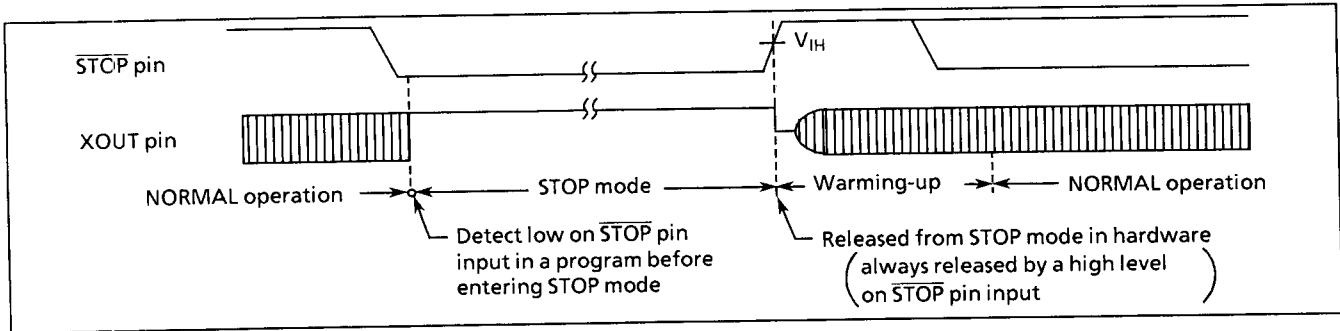


Figure 1-10. Released from STOP Mode by Level

Note 1: Once warming-up starts, the device does not return to STOP mode even when the \overline{STOP} pin input is pulled low again.

Note 2: If RELM is changed to 1 (level mode) after being set to 0 (edge mode), STOP mode remains unchanged unless a rising edge on \overline{STOP} pin input is detected.

b. Released by edge (when RELM = 0)

The device is released from STOP mode by a rising edge on \overline{STOP} pin input. This method is used in applications where a relatively short time of program processing is repeated at certain fixed intervals. Apply a fixed-period signal (e.g., clock from the low-power oscillating source) to the \overline{STOP} pin. When RELM = 0 (edge mode), the device is placed in STOP mode even when the \overline{STOP} pin input level is high.

Example: Entering STOP mode from NORMAL mode

```

LD (SYSCR1), 10010000B ; Set to be released by edge when entering STOP mode
    
```

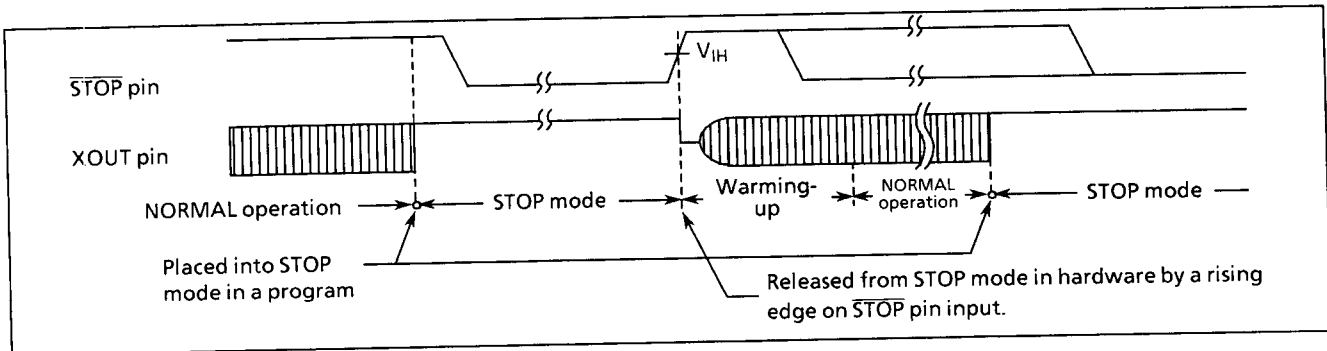


Figure 1-11. Released from STOP Mode by Edge

The device is released from STOP mode following the sequence described below.

- ① In single clock mode, only the high-frequency oscillator is oscillating.
- ② A warming-up time is inserted in order to allow for the clock oscillation to stabilize. During warming-up, the internal circuits remain idle. The warming-up time can be selected from three choices according to the oscillator characteristics by using WUT (SYSCR1 Register bits 3 and 2).
- ③ After an elapse of the warming-up time, the device restarts normal operation beginning with the instruction next to one that placed it in STOP mode. At this time, the prescaler and divider for the timing generator start from the zero-cleared state.

Table 1-1. Warming-up Time (Example: $f_c = 20$ MHz)

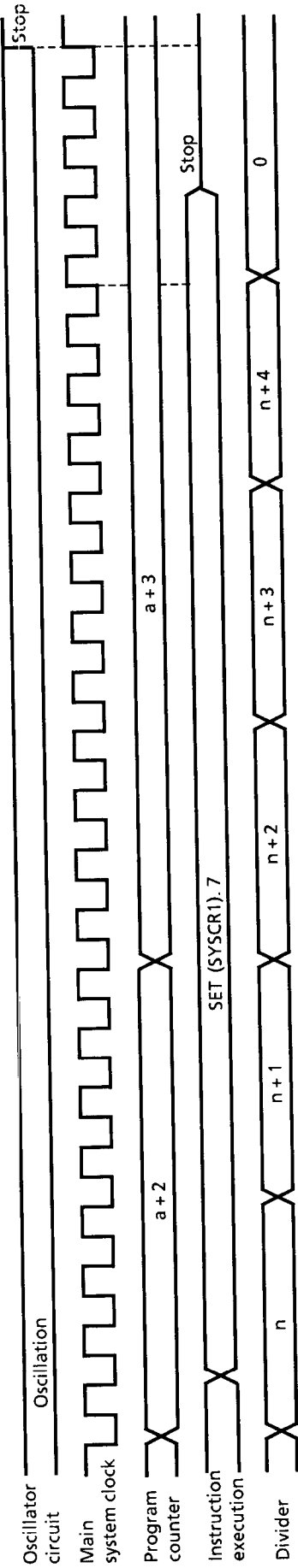
WUT	Warming-up time [ms]	
	When returning to NORMAL mode	
	DV1CK = 0	DV1CK = 1
00	9.831	19.662
01	3.277	6.554
10	0.819	1.638
11	reserved	reserved

Note: Because the warming-up time is obtained from the fundamental clock by dividing it, if the oscillation frequency fluctuates while exiting STOP mode, the warming-up time becomes to have some error. Therefore, the warming-up time must be handled as an approximate value.

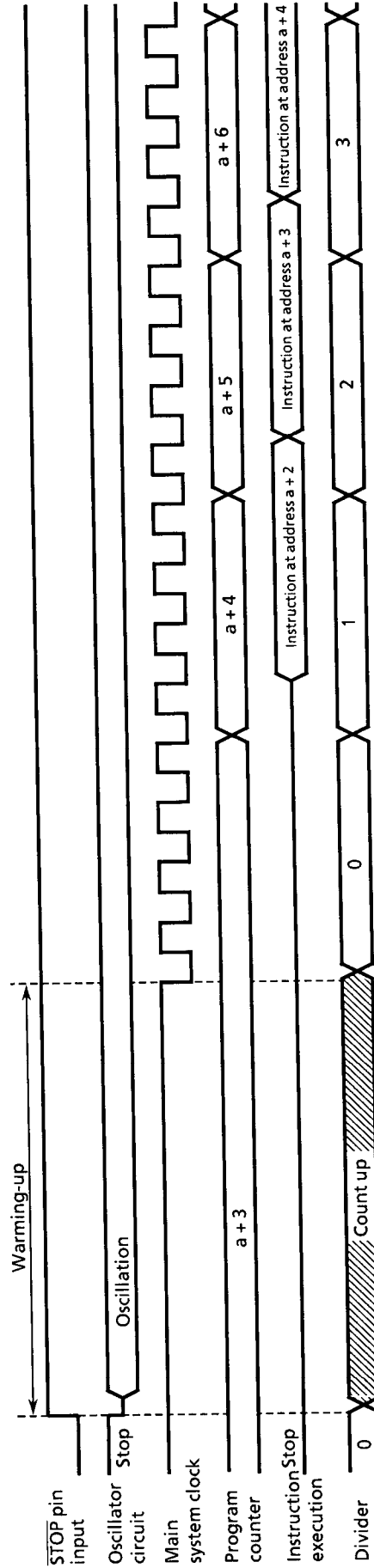
The device can also be released from STOP mode by pulling the $\overline{\text{RESET}}$ pin input low, in which case the device is immediately reset as is normally reset by RESET. After reset, the device starts operating from NORMAL1 mode.

Note: When exiting STOP mode while the device is retained at low voltage, the following caution is required.

Before exiting STOP mode, the power supply voltage must be raised to the operating voltage. At this time, the $\overline{\text{RESET}}$ pin level also is high and rises along with the power supply voltage. If the device has a time-constant circuit added external to the chip, the voltage on $\overline{\text{RESET}}$ pin input does not rise as fast as the power supply voltage. Therefore, if the voltage level on $\overline{\text{RESET}}$ pin input is below the $\overline{\text{RESET}}$ pin's noninverted, high-level input voltage (hysteresis input), the device may be reset.



(a) Entering STOP mode (Example: Entered into by the SET (SYSR1).7 instruction placed at address a)



(b) Exiting STOP mode

Figure 1-12. Entering and Exiting STOP Mode (when DV1CK = 0)

(2) IDLE mode (IDLE1)

IDLE mode is controlled by System Control Register 2 (SYSCR2) and a maskable interrupt. During IDLE mode, the device retains the following state.

- ① The CPU and watchdog timer stop operating.
The peripheral hardware continues operating.
- ② The data memory, register, program status word, and port output latch hold the state in which they were immediately before entering IDLE mode.
- ③ The program counter holds the instruction address two instructions ahead the one that placed the device in IDLE mode.

Example: Placing the device in IDLE mode

SET (SYSCR2) . 4

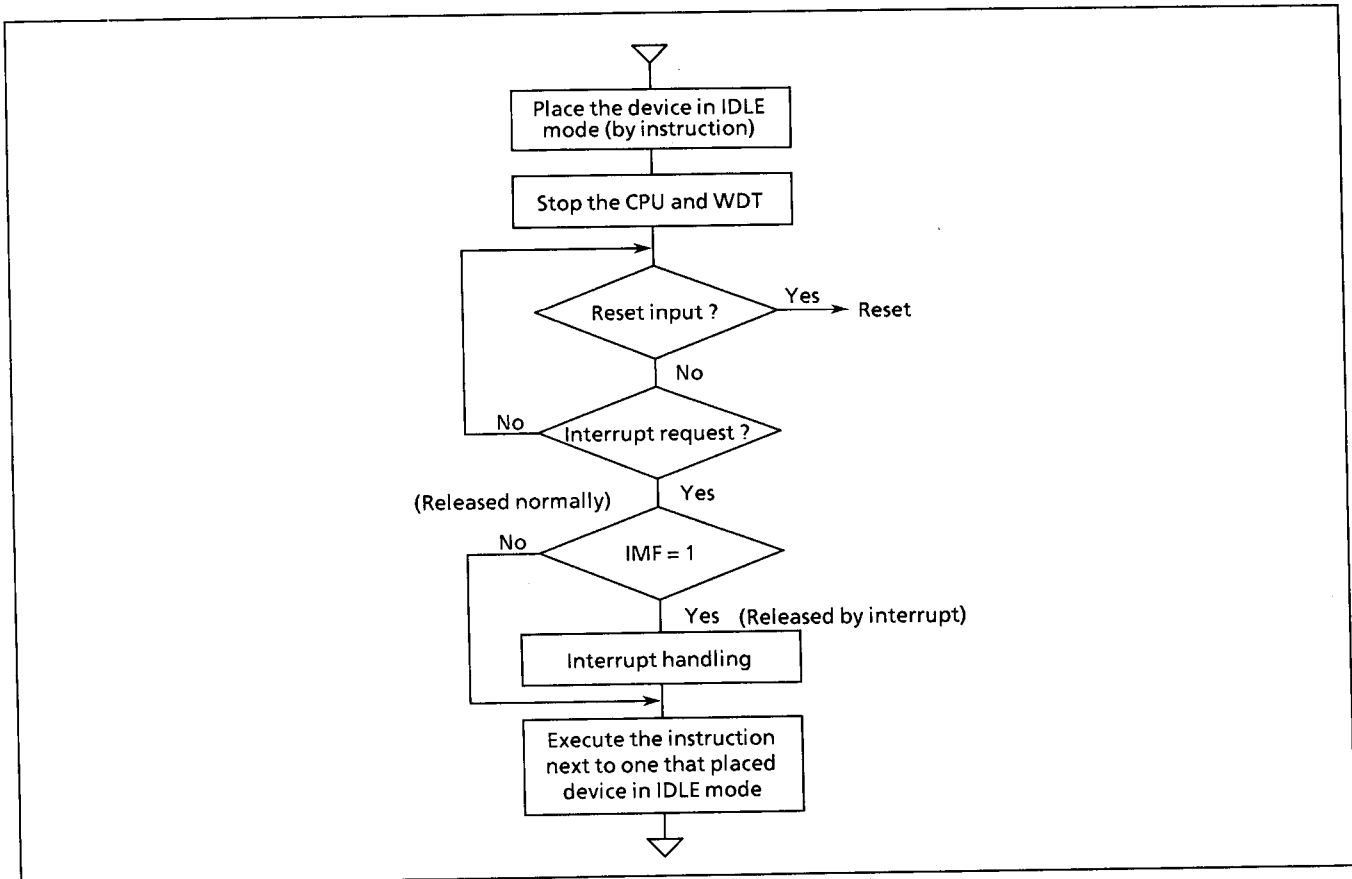


Figure 1-13. IDLE Mode

The device can be released from IDLE mode normally or by an interrupt as selected with the interrupt master enable flag (IMF).

a. Released normally (when IMF = 0)

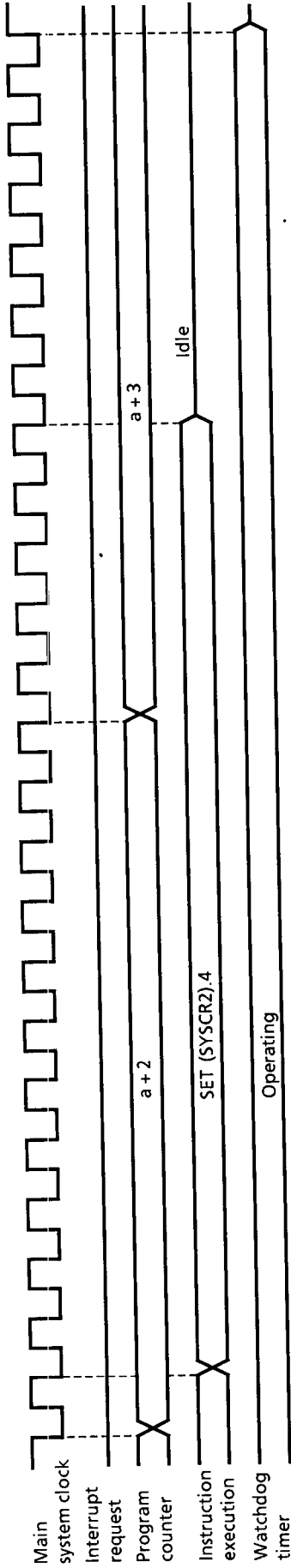
The device can be released from IDLE mode by the interrupt source enabled by the interrupt individual enable flag (EF), and restarts execution beginning with the instruction next to one that placed it in IDLE mode. The interrupt latch (IL) for the interrupt source used to exit IDLE mode normally needs to be cleared to 0 using a load instruction.

b. Released by interrupt (when IMF = 1)

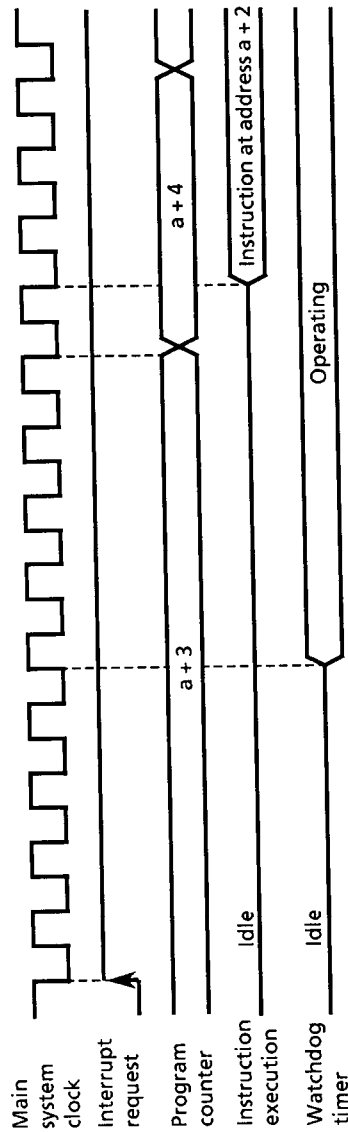
The device can be released from IDLE mode by the interrupt source enabled by the interrupt individual enable flag (EF), and enters interrupt handling. After interrupt handling, the device returns to the instruction next to one that placed it in IDLE mode.

The device can also be released from IDLE mode by pulling the $\overline{\text{RESET}}$ pin input low, in which case the device is immediately reset as is normally reset by $\overline{\text{RESET}}$. After reset, the device starts operating from NORMAL1 mode.

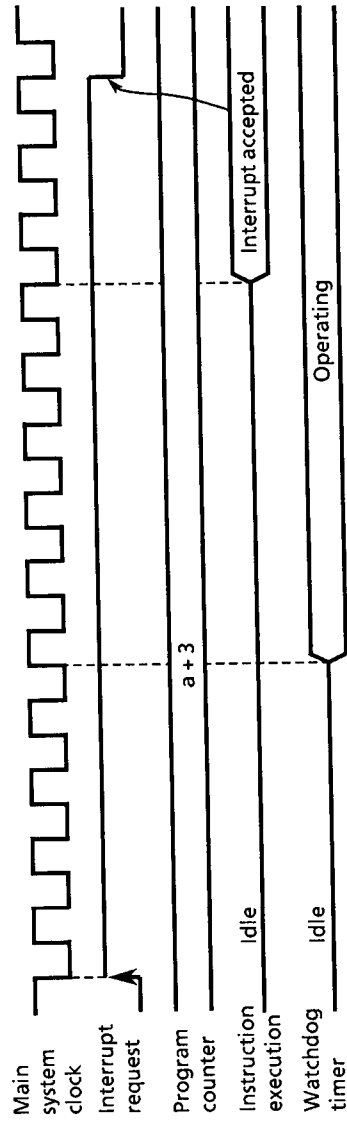
Note: If a watchdog timer interrupt occurs immediately before entering IDLE mode, the device processes the watchdog timer interrupt without entering IDLE mode.



(a) Entering IDLE mode (Example: Entered into by the SET instruction placed at address a)



① Released normally



② Released by interrupt

(b) Exiting IDLE mode

Figure 1-14. Entering and Exiting IDLE Mode

1.5 Interrupt Control Circuit

The TMP88CS43 has a total of 35 interrupt sources not including reset. Of the internal interrupt sources, two are pseudo-nonmaskable interrupts and all others maskable interrupts.

Table 1-2 lists the interrupt sources of the TMP88CS43.

Table 1-2. Interrupt Sources

Interrupt Source		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/ External	(Reset)	Nonmaskable	—	FFFFC _H	High 0
Internal	INTSW (Software interrupt)	Pseudo-nonmaskable	—	FFF8 _H	1
Internal	INTWDT (WDT interrupt)		IL ₂	FFF4 _H	2
External	INT0 (External interrupt 0)	IMF · EF ₃ = 1, INT0EN = 1	IL ₃	FFF0 _H	3
	reserved	IMF · EF ₄ = 1	IL ₄	FFFE _C _H	4
External	INT1 (External interrupt 1)	IMF · EF ₅ = 1	IL ₅	FFFE8 _H	5
Internal	INTTBT (Time base timer)	IMF · EF ₆ = 1	IL ₆	FFFE4 _H	6
	reserved	IMF · EF ₇ = 1	IL ₇	FFFE0 _H	7
Internal	INTEMG1 (ch1 error detection)	IMF · EF ₈ = 1	IL ₈	FFFD _C _H	8
Internal	INTEMG2 (ch2 error detection)	IMF · EF ₉ = 1	IL ₉	FFFD8 _H	9
Internal	INTCLM1 (ch1 overload protection)	IMF · EF ₁₀ = 1	IL ₁₀	FFFD4 _H	10
Internal	INTCLM2 (ch2 overload protection)	IMF · EF ₁₁ = 1	IL ₁₁	FFFD0 _H	11
Internal	INTTMR31 (ch1 timer 3)	IMF · EF ₁₂ = 1	IL ₁₂	FFFC _C _H	12
Internal	INTTMR32 (ch2 timer 3)	IMF · EF ₁₃ = 1	IL ₁₃	FFFC8 _H	13
	reserved	IMF · EF ₁₄ = 1	IL ₁₄	FFFC4 _H	14
External	INT5 (External interrupt 5)	IMF · EF ₁₅ = 1	IL ₁₅	FFFC0 _H	15
Internal	INTPDC1 (ch1 position detection)	IMF · EF ₁₆ = 1	IL ₁₆	FFFB _C _H	16
Internal	INTPDC2 (ch2 position detection)	IMF · EF ₁₇ = 1	IL ₁₇	FFFB8 _H	17
Internal	INTPWM1 (ch1 waveform generator)	IMF · EF ₁₈ = 1	IL ₁₈	FFFB4 _H	18
Internal	INTPWM2 (ch2 waveform generator)	IMF · EF ₁₉ = 1	IL ₁₉	FFFB0 _H	19
Internal	INTEDT1 (ch1 electrical angle timer)	IMF · EF ₂₀ = 1	IL ₂₀	FFFAC _H	20
Internal	INTEDT2 (ch2 electrical angle timer)	IMF · EF ₂₁ = 1	IL ₂₁	FFFA8 _H	21
Internal	INTTMR11 (ch1 timer 1)	IMF · EF ₂₂ = 1	IL ₂₂	FFFA4 _H	22
Internal	INTTMR12 (ch2 timer 1)	IMF · EF ₂₃ = 1	IL ₂₃	FFFA0 _H	23
Internal	INTTMR21 (ch1 timer 2)	IMF · EF ₂₄ = 1	IL ₂₄	FFF9 _C _H	24
Internal	INTTMR22 (ch2 timer 2)	IMF · EF ₂₅ = 1	IL ₂₅	FFF98 _H	25
Internal	INTTC1 (TC1: 16-bit timer/counter 1)	IMF · EF ₂₆ = 1	IL ₂₆	FFF94 _H	26
Internal	INTTC2 (CTC: 16-bit CTC counter)	IMF · EF ₂₇ = 1	IL ₂₇	FFF90 _H	27
Internal	INTTC6 (TC6: 8-bit timer/counter 6)	IMF · EF ₂₈ = 1	IL ₂₈	FFF8 _C _H	28
External	INT2 (External interrupt 2)	IMF · EF ₂₉ = 1	IL ₂₉	FFF88 _H	29
External	INT3 (External interrupt 3)	IMF · EF ₃₀ = 1	IL ₃₀	FFF84 _H	30
External	INT4 (External interrupt 4)	IMF · EF ₃₁ = 1	IL ₃₁	FFF80 _H	31
Internal	INTRX (UART reception)	IMF · EF ₃₂ = 1	IL ₃₂	FFF3 _C _H	32
Internal	INTTX (UART transmission)	IMF · EF ₃₃ = 1	IL ₃₃	FFF38 _H	33
Internal	INTSIO (SIO interrupt)	IMF · EF ₃₄ = 1	IL ₃₄	FFF34 _H	34
Internal	INTTC3 (TC3: 8-bit timer/counter 3)	IMF · EF ₃₅ = 1	IL ₃₅	FFF30 _H	35
Internal	INTTC4 (TC4: 8-bit timer/counter 4)	IMF · EF ₃₆ = 1	IL ₃₆	FFF2 _C _H	36
Internal	INTTC5 (TC5: 8-bit timer/counter 5)	IMF · EF ₃₇ = 1	IL ₃₇	FFF28 _H	37
Internal	INTADC (AD)	IMF · EF ₃₈ = 1	IL ₃₈	FFF24 _H	Low 38

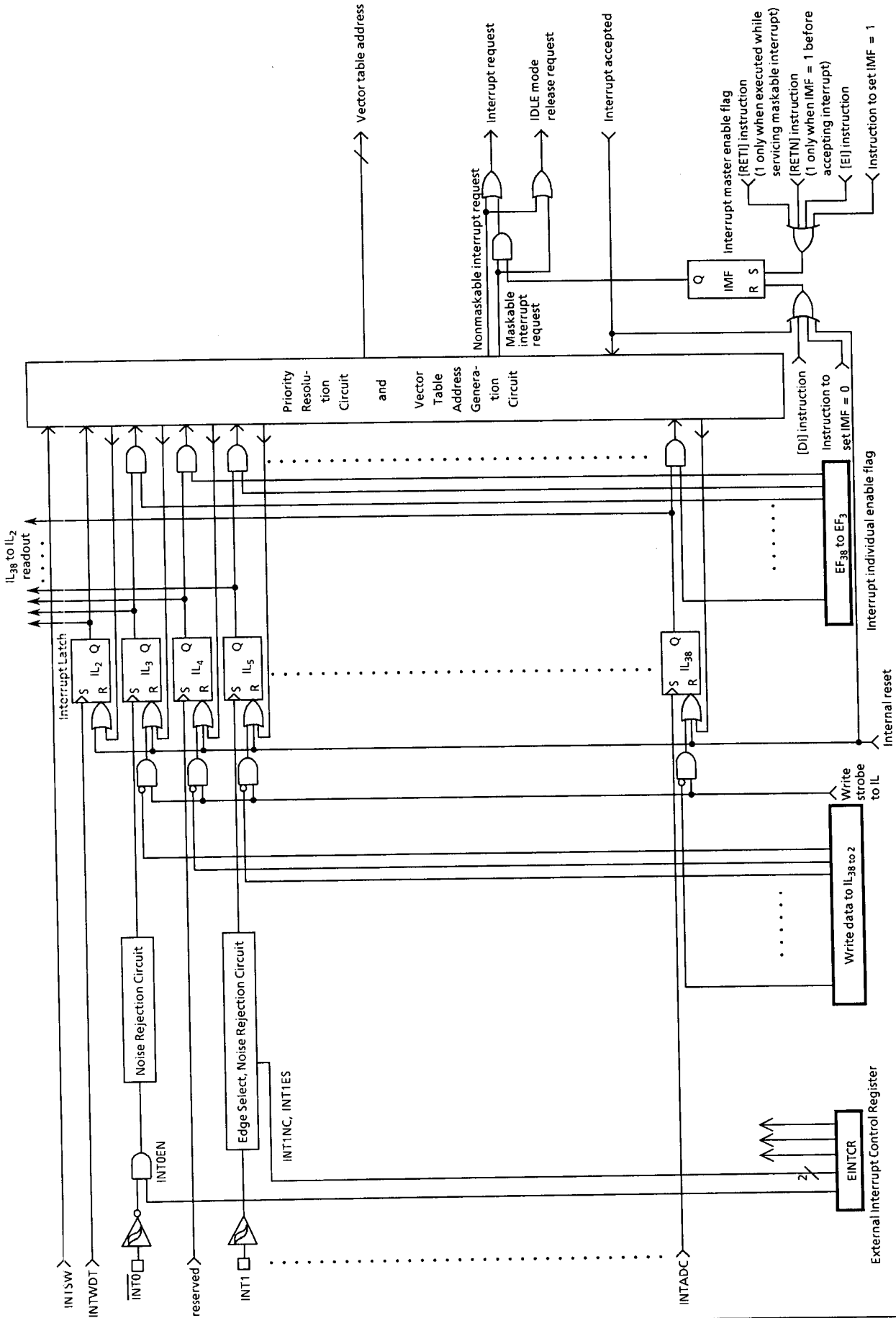


Figure 1-15. Interrupt Control Circuit

(1) Interrupt latches (IL₃₈ to IL₂)

The interrupt latches are provided one for each interrupt source except for software interrupts, and are set to 1 when an interrupt request is generated. When the interrupt has been enabled for acceptance, the interrupt controller requests the CPU to accept the interrupt. The interrupt latch is cleared to 0 immediately after the interrupt is accepted. When reset, all interrupt latches are initialized to 0.

The interrupt latches are allocated to addresses 0003C and 0003DH in the SFR area, and all interrupt latches except IL₂ can be individually cleared by an instruction. (However, read-modify-write instructions such as those used for bit manipulation and arithmetic operation cannot be used. This is because an interrupt request generated while executing a read-modify-write instruction may happen to be cleared.) Interrupt requests can be canceled or initialized in a program. The interrupt latches cannot be set directly by an instruction. Because the contents of the interrupt latches can be read out, interrupt requests can be tested in software.

Note 1: When the interrupt latch is read out while the external interrupt input pin (INT₀, INT₂, INT₃, INT₄, or INT₅) has an indeterminate value applied, an indeterminate value is read out.
 Note 2: When testing interrupt requests in software, make sure the external interrupt input pins have valid signals applied.

Example 1: Clearing the interrupt latches

```
LD    (ILL), 00000000B    ; IL7 to IL2 ← 0
LD    (ILH), 00000000B    ; IL8 to IL15 ← 0
LD    (ILE), 00000000B    ; IL16 to IL23 ← 0
LD    (ILD), 00000000B    ; IL24 to IL31 ← 0
LD    (ILC), 00000000B    ; IL32 to IL38 ← 0
```

Example 2: Reading the interrupt latches

```
LD    WA, (ILL)           ; W ← (ILH), A ← (ILL)
LD    BC, (ILE)           ; B ← (ILD), C ← (ILE)
LD    D, (ILC)            ; D ← (ILC)
```

Example 3: Testing the interrupt latches

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

(2) Interrupt Enable Register (EIR)

This register enables or disables the interrupt sources except pseudo-nonmaskable interrupts (software and watchdog timer interrupts) for or against acceptance. The pseudo-nonmaskable interrupts are always accepted no matter how this register is set. However, the pseudo-nonmaskable interrupts cannot themselves be nested one in another.

The Interrupt Enable Register consists of an interrupt master enable flag (IMF) and interrupt individual enable flags (EF). The Interrupt Enable Register is allocated to addresses 0003A and 0003BH in the SFR area, and can be read and written by an instruction (including bit manipulating and other read-modify-write instructions).

Note: However, do not execute any read-modify-write instruction on EIRL (address 0003AH) while servicing a pseudo-nonmaskable interrupt. If such an instruction is executed, the IMF flag cannot be set to 1 after RETN.

① Interrupt master enable flag (IMF)

This flag enables or disables all maskable interrupts for or against acceptance. All maskable interrupts are disabled against acceptance when this flag is cleared to 0, and are enabled for acceptance when the flag is set to 1.

When an interrupt is accepted, the interrupt master enable flag is cleared to 0, thereby temporarily disabling the subsequent maskable interrupts against acceptance. The flag is then set to 1 by the maskable interrupt return instruction [RETI] after executing the interrupt service routine, thereby reenabling maskable interrupts for acceptance. Therefore, if an interrupt request has already been received, the interrupt is accepted and serviced immediately after executing the [RETI] instruction.

For the pseudo-nonmaskable interrupts, the nonmaskable interrupt return instruction [RETN] is used to return from the interrupt to the main program. In this case, the interrupt master enable flag is set to 1 only when pseudo-nonmaskable interrupt processing is entered into while the interrupts are enabled for acceptance (IMF = 1). However, if the interrupt master enable flag was cleared to 0 in the interrupt service routine, it remains cleared.

The interrupt master enable flag is assigned to EIRL (address 0003AH in SFR) bit 0, and can be read and written by an instruction. The interrupt master enable flag normally is set and cleared using the [EI] and [DI] instructions. Note that when reset, the interrupt master enable flag is initialized to 0.

② Interrupt individual enable flags (EF₃₈ to EF₃)

These flags are used to individually enable or disable each maskable interrupt source, except external interrupt 0, for or against acceptance. An interrupt is enabled for acceptance when its corresponding interrupt individual enable flag is 1, and is disabled against acceptance when the flag is 0.

Note: Before manipulating the interrupt individual enable flags (EF_x), be sure to clear the interrupt master enable flag (IMF) (to disable interrupts against acceptance).
Do not set any interrupt individual enable flags (EF_x) and the IMF flag at the same time.

Example

```
DI                ; Disable interrupts (IMF = 0)
SET  (EIRL). 5    ; EF5 ← 1
CLR  (EIRL). 6    ; EF6 ← 0
CLR  (EIRH). 4    ; EF12 ← 0
CLR  (EIRD). 0    ; EF24 ← 0
EI                ; Enable interrupts (IMF = 1)
```

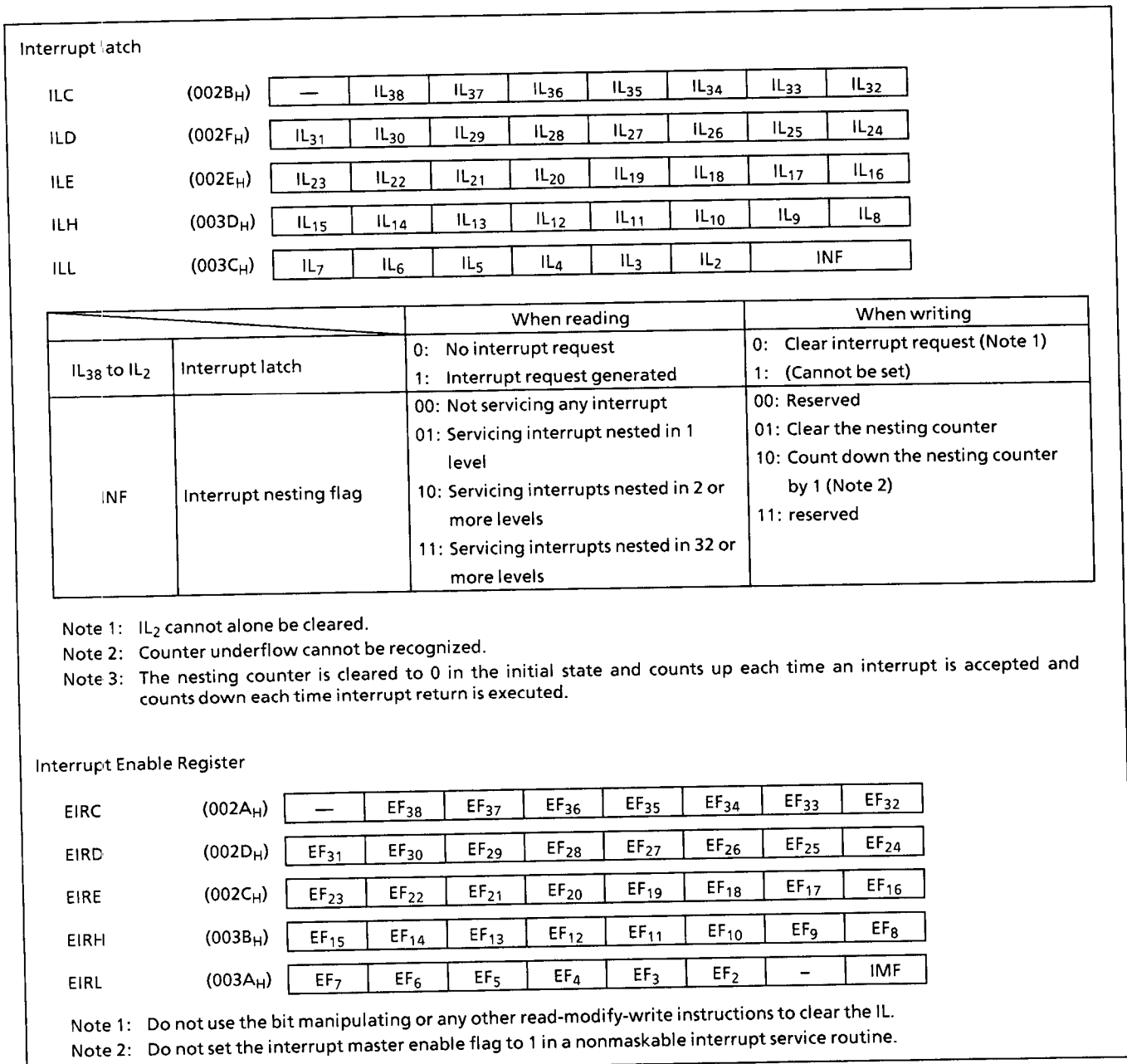


Figure 1-16. Interrupt Latches (IL) and Interrupt Enable Register (EIR)

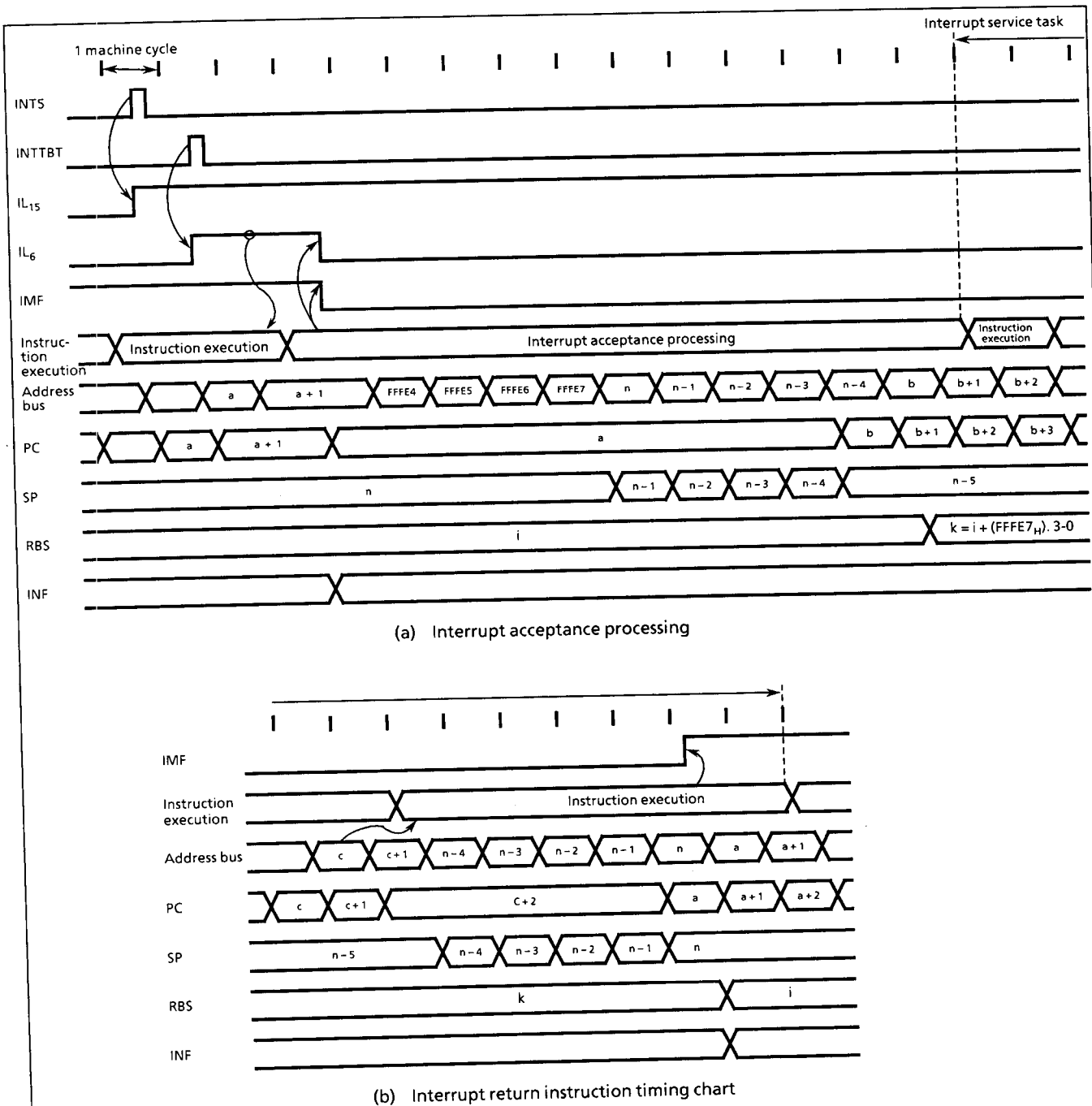
1.5.1 Interrupt Handling

An interrupt request is retained until the interrupt latch is cleared to 0 by accepting the interrupt or by a reset or instruction. Interrupt acceptance processing is executed in 12 machine cycles (2.4 μ s at 20.0 MHz) after the instruction being executed is finished. The interrupt service task is finished by executing the interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for pseudo-nonmaskable interrupts). Figure 1-17 shows a timing chart for interrupt acceptance processing.

(1) Interrupt acceptance processing

In interrupt acceptance processing, the following operations are performed automatically.

- ① Clear the interrupt master enable flag (IMF) to 0, thereby temporarily disabling the subsequent maskable interrupts against acceptance. If the interrupt being accepted is a nonmaskable interrupt, this operation also temporarily disables the subsequent nonmaskable interrupts against acceptance.
- ② Clear the interrupt latch for the accepted interrupt source to 0.
- ③ Save the contents of the Program Counter (PC) and Program Status Word (PSW) to the stack. (Pushed down in order of PSW_H, PSW_L, PC_E, PC_H, and PC_L.) The Stack Pointer (SP) is decremented five times.
- ④ From the vector table address corresponding to the accepted interrupt source, read the entry address (interrupt vector) of the interrupt service routine and set it in the program counter.
- ⑤ Read the RBS control code from the vector table and add its 4 low-order bits to the Register Bank Selector (RBS).
- ⑥ Let the interrupt nesting counter count up.
- ⑦ Proceed to execute the instruction stored at the interrupt service routine entry address.

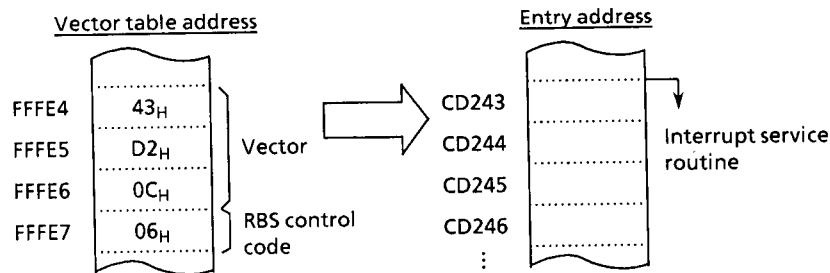


Note 1: The letter "a" denotes the return address, "b" denotes the entry address, and "c" denotes the address in which the RETI instruction is stored.

Note 2: The time from when the interrupt latch is set till when the interrupt acceptance processing starts, is $62/f_c$ [s] at maximum in cases when interrupts have been enabled (and no other interrupt requests are generated).

Figure 1-17. Interrupt Acceptance Processing and Interrupt Return Instruction Timing Chart

Example: Relationship between the vector table addresses and interrupt service routine entry addresses in INTTBT acceptance processing



Even when a maskable interrupt occurs which has higher priority than the interrupt being serviced, it is not accepted until the interrupt master enable flag is set to 1. Therefore, if multiple interrupts need to be handled, set the interrupt master enable flag to 1 in the interrupt service routine. At this time, be sure to selectively enable the interrupt sources that can be accepted by using the interrupt individual enable flags. However, do not execute any read-modify-write instruction on EIRL (address 0003A_H) during a pseudo-nonmaskable interrupt service task.

(2) Saving and restoring general-purpose registers

In interrupt acceptance processing, the program counter and program status word are automatically saved to the stack, but the accumulator and other registers are not automatically saved. If these registers need to be saved, do it in a program. However, if handling of multiple interrupts is desired, make sure the data memory areas in which the register contents are saved will not overlap.

There are following four methods to save the general-purpose registers.

① Saving and restoring general-purpose registers by automatic register bank switchover

The general-purpose registers can be saved at high speed by switching them to an unused register bank. Assign bank 0 to the main task and banks 1 to 15 to each interrupt service task. To increase the efficiency of data memory usage, assign a common bank to the interrupt sources which are not nested, i.e., not accepted one while servicing another.

The switched register banks are automatically restored by executing the interrupt return instruction [RETI] or [RETN]. Therefore, there is no need to save the RBS in a program.

Example: Register bank switchover

```

PINTxx :      Interrupt processing
              RETI
              ⋮
VINTxx :      DP      PINTxx
              DB      1          ; RBS←RBS + 1
    
```

② Saving and restoring general-purpose registers by register bank switchover

The general-purpose registers can be saved at high speed by switching them to an unused register bank. Use bank 0 for the main task and one of banks 1 to 15 for the interrupt service task.

Example: Register bank switchover

```

PINTxx :    LD      RBS, n
            ; Interrupt processing
            RETI    ; Restore bank and return from interrupt
            ;
VINTxx :    DP      PINTxx ; Interrupt service routine entry address
            ; Entry address
            DB      0      ;
    
```

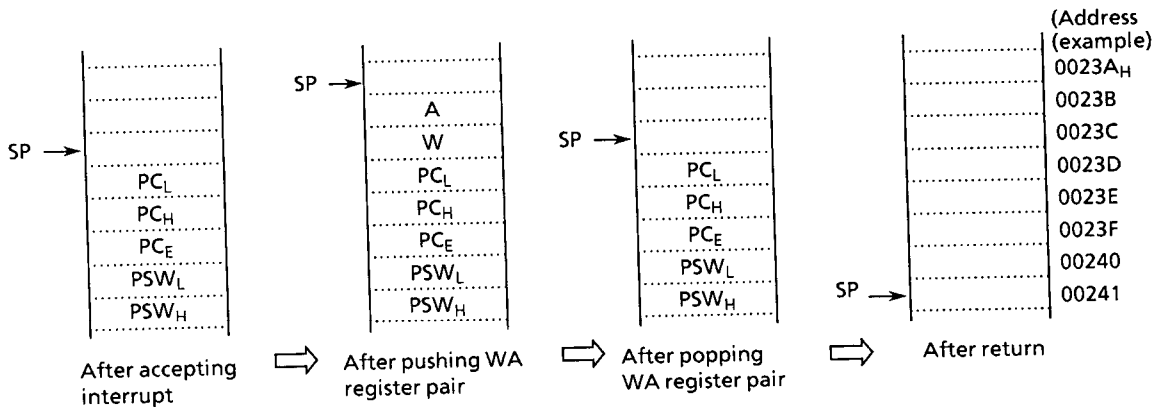
③ Saving and restoring general-purpose registers by push/pop instructions

When saving only specific registers or nesting interrupts from the same interrupt source, use the push/pop instructions to save and restore the general-purpose registers.

Example: Saving and restoring registers by push/pop instructions

```

PINTxx :    PUSH   WA      ; Save WA register pair to the stack
            ; Interrupt processing
            POP    WA      ; Restore WA register pair from the stack
            RETI    ; Return
    
```



④ Saving and restoring general-purpose registers by transfer instruction

When saving only specific registers in interrupt processing where interrupts are not nested, use an instruction for transfer to and from data memory to save and restore the general-purpose registers.

Example: Saving and restoring registers by data memory transfer instruction

```

PINTxx :    LD      (GSAVA), A ; Save register A
            ; Interrupt processing
            LD      A, (GSAVA) ; Restore register A
            RETI    ; Return
    
```

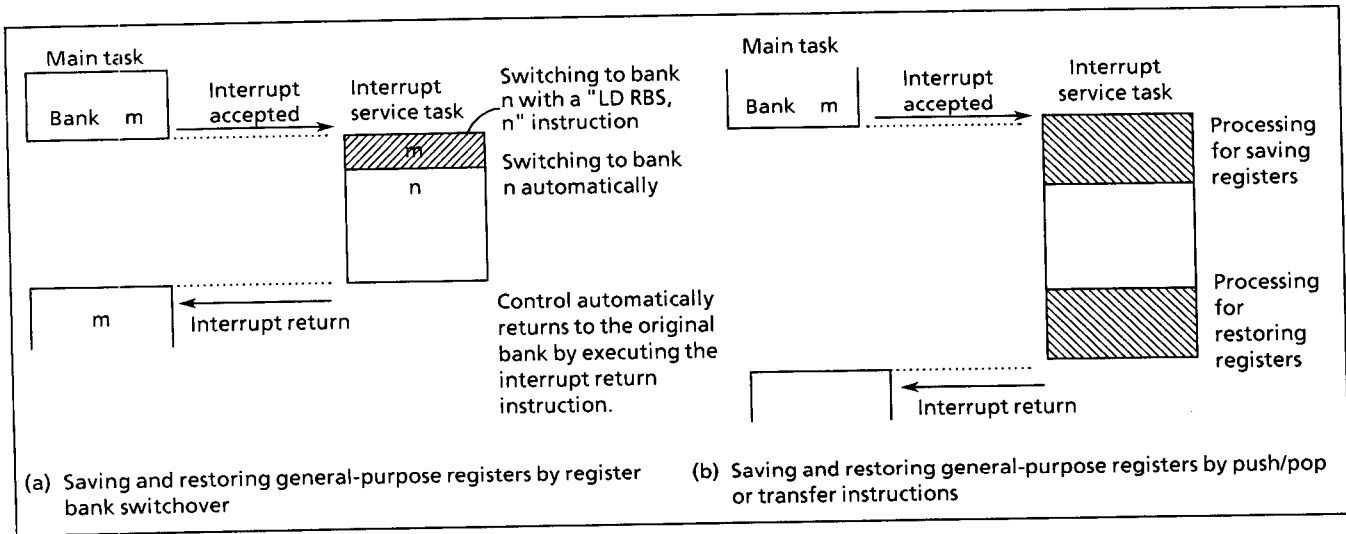


Figure 1-18. Saving and Restoring General-purpose Registers in Interrupt Processing

(3) Interrupt return

The interrupt return instructions perform the following operations.

[RETI] maskable interrupt return	[RETN] nonmaskable interrupt return
<ul style="list-style-type: none"> ① Restore the content of the program counter and that of the program status word from the stack. ② Increment the stack pointer 5 times. ③ Set the interrupt master enable flag to 1. 	<ul style="list-style-type: none"> ① Restore the content of the program counter and that of the program status word from the stack. ② Increment the stack pointer 5 times. ③ Set the interrupt master enable flag to 1 only when the nonmaskable interrupt was accepted while interrupts were enabled. However, if the interrupt master enable flag was cleared to 0 in the interrupt service routine, the flag remains cleared.
<ul style="list-style-type: none"> ④ The interrupt nesting counter is decremented and the interrupt nesting flag changes state. 	<ul style="list-style-type: none"> ④ The interrupt nesting counter is decremented and the interrupt nesting flag changes state.

Interrupt requests are sampled in the last cycle of the instruction being executed. Therefore, the next interrupt can be processed immediately after executing the interrupt return instruction.

Note: If the interrupt processing takes longer time than the duration for which an interrupt request is generated, only the interrupt service task is executed, and the main task is not executed.

1.5.2 Software Interrupt (INTSW)

A software interrupt is generated by executing the SWI instruction, and interrupt processing is immediately entered into (highest priority interrupt). However, if nonmaskable interrupt processing has already been entered into, no software interrupts are generated even by executing the SWI instruction, in which case the SWI instruction works in the same way as the NOP instruction.

The SWI instruction can only be used for the address error detection or debugging described below, and cannot be used for any other purposes.

① Address error detection

If the CPU fetches instructions from an address that does not exist in memory for some reasons (e.g., noise), FF_H is read out. Because code FF_H is the SWI instruction, a software interrupt is generated, thereby making it possible to detect address errors. Also, the address error detection range can further be expanded by filling all unused areas of the program memory with FF_H. If the RAM, SFR, or DBR area is accessed for instruction fetch, an address trap reset is generated.

② Debugging

Debugging efficiency can be increased by placing the SWI instruction at addresses where software breakpoints are set.

1.5.3 External Interrupts

The TMP88CS43 has six external interrupt inputs, all of which have a digital noise rejection circuit (which rejects input pulses shorter than a certain time as noise).

Also, the INT1 to INT4 pins allow the active edge to be selected.

The $\overline{\text{INT0}}$ /P10 pin can be used as an external interrupt input pin or an input/output port as selected. When reset, this pin is set for input mode.

The External Interrupt Control Register is used to select the active edge, control noise rejection, and select the $\overline{\text{INT0}}$ /P10 pin function.

Table 1-3. External Interrupts

Source	Pin Name	Shared Pin	Enable Condition	Edge	Digital Noise Rejection Circuit
INT0	$\overline{\text{INT0}}$	P10	IMF = 1, EF ₃ = 1, INT0EN = 1	Falling edge	Pulses less than 2/fc [s] in duration are rejected as noise. Those greater than 6/fc [s] in duration are always recognized as signal.
INT1	INT1	P11	IMF · EF ₅ = 1	Falling edge or rising edge	Pulses less than 15/fc or 63/fc [s] in duration are rejected as noise. Those greater than 48/fc or 192/fc [s] in duration are always recognized as signal. (When DV1CK = 0)
INT2	INT2	P12/TC1	IMF · EF ₂₉ = 1		Pulses less than 7/fc [s] in duration are rejected as noise. Those greater than 24/fc [s] in duration are always recognized as signal. (When DV1CK = 1)
INT3	INT3	P21/TC3	IMF · EF ₃₀ = 1		
INT4	INT4	P22/TC4	IMF · EF ₃₁ = 1		
INT5	$\overline{\text{INT5}}$	P20/ $\overline{\text{STOP}}$	IMF · EF ₁₅ = 1	Falling edge	Pulses less than 2/fc [s] in duration are rejected as noise. Those greater than 6/fc [s] in duration are always recognized as signal.

Note 1: When a noise-free signal is applied to the external interrupt input pin during NORMAL1 or IDLE1 mode, the maximum time before the interrupt latch is set after the active edge of the input signal is as follows.

- ① INT1 pin $49/f_c$ [s] (when INT1NC = 1), $193/f_c$ [s] (when INT1NC = 0)
- ② INT2 to 4 pins $25/f_c$ [s]

Note 2: When INT0EN = 0, the interrupt latch IL₃ is not set even by detecting a falling edge on $\overline{INT0}$ pin input.

Note 3: When using the shared pin as an output port and the data on it changes state or its direction is switched between input and output, an unsolicited interrupt request signal may be generated. Therefore, some corrective measure must be taken by, for example, disabling the interrupt enable flag.

EINTCR (00037 _H)		7	6	5	4	3	2	1	0	(Initial value: 0000 000*)
		INT1 NC	INT0 EN	INT4 ES	INT3 ES	INT2 ES	INT1 ES			
INT1NC	Select INT1 noise rejection time	0: Reject pulses less than $63/f_c$ [s] as noise 1: Reject pulses less than $15/f_c$ [s] as noise		R/W						
INT0EN	Select P10/ $\overline{INT0}$ pin function	0: P10 input/output port 1: $\overline{INT0}$ pin (P10 port must be set for input mode)								
INT4ES	Select INT4 edge	00: Generate interrupt request at rising edge 01: Generate interrupt request at falling edge 10: Generate interrupt request at both rising and falling edges 11: Generate interrupt request at high level								
INT3 ES INT2 ES INT1 ES	Select INT3 to INT1 edge	0: Generate interrupt request at rising edge 1: Generate interrupt request at falling edge								

Note: f_c ; the high-frequency clock [Hz], *; Don't care

Figure 1-19. External Interrupt Control Register

1.6 Reset Circuit

The TMP88CS43 has four ways to generate a reset: external reset input, address trap reset output, watchdog timer reset output, or system clock reset output.

Table 1-4 shows how the internal hardware is initialized by reset operation.

At power-on time, the internal cause reset output circuits (watchdog timer reset, address trap reset, and system clock reset) are not initialized.

Table 1-4. Internal Hardware Initialization by Reset Operation

Internal Hardware	Initial Value	Internal Hardware	Initial Value
Program Counter (PC)	(FFFFE _H to FFFFC _H)	Prescaler and divider for the timing generator	0
Stack Pointer (SP)	Not initialized		
General-purpose Registers (W, A, B, C, D, E, H, L)	Not initialized		
Register Bank Selector (RBS)	0	Watchdog timer	Enable
Jump Status Flag (JF)	1		
Zero Flag (ZF)	Not initialized	Output latch of input/output port	See description of each input/output port.
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	Control register	See description of each control register.
Interrupt Individual Enable Flag (EF)	0		
Interrupt Latch (IL)	0		
Interrupt Nesting Flag (INF)	0	RAM	Not initialized

1.6.1 External Reset Input

The $\overline{\text{RESET}}$ pin is a hysteresis input with a pull-up resistor included. By holding the $\overline{\text{RESET}}$ pin low for at least three machine cycles ($12/f_c$ [s]) or more while the power supply voltage is within the rated operating voltage range and the oscillator is oscillating stably, the device is reset and its internal state is initialized.

When the $\overline{\text{RESET}}$ pin input is released back high, the device is freed from reset and starts executing the program beginning with the vector address stored at addresses FFFFC to FFFF_{E_H}.

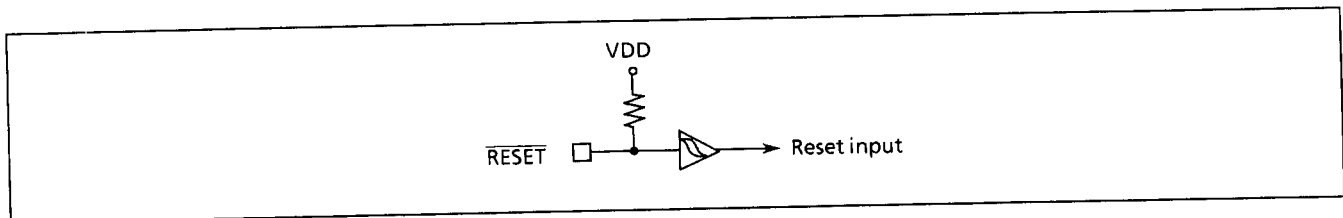


Figure 1-20. Reset Circuit

1.6.2 Address Trap Reset

If the CPU goes wild for reasons of noise, etc. and attempts to fetch instructions from the internal RAM, DBR, or SFR area, the device internally generates a reset.

1.6.3 Watchdog Timer Reset

Refer to Section 2.4, "Watchdog Timer."

1.6.4 System Clock Reset

When XEN (SYSCR2 Register bit 7) is cleared to 0 or when XEN is cleared to 0 while $SYSCK = 0$, the system clock is turned off, causing the CPU to become locked up. To prevent this problem, upon detecting $XEN = 0$, $XEN = SYSCK = 0$ or $SYSCK = 1$, the device automatically generates an internal reset signal to let the system clock continue oscillating.

2. Peripheral Hardware Functions

2.1 Special Function Registers (SFR) and Data Buffer Registers (DBR)

The TLCS-870/X series uses memory-mapped I/O method, and all peripheral hardware control signal and data transfers in it are performed via Special Function Registers (SFR) or Data Buffer Registers (DBR).

Figure 2-1(a) lists the Special Function Registers (SFR). Figure 2-1(b) lists the Data Buffer Registers (DBR).

Address	Read	Write	Address	Read	Write
0000		P0DR (P0 Port)	0020	TC5CR (Timer 5 Control)	
0001		P1DR (P1 Port)	0021	TC6CR (Timer 6 Control)	
0002		P2DR (P2 Port)	0022	TTREG5 (Timer 5 Period Register)	
0003		P3DR (P3 Port)	0023	TTREG6 (Timer 6 Period Register)	
0004		P4DR (P4 Port)	0024	PWREG5 (Timer 5 Pulse Width Register)	
0005		P5DR (P5 Port)	0025	PWREG6 (Timer 6 Pulse Width Register)	
0006		P6DR (P6 Port)	0026	ADCCRA (AD Control A)	
0007		P7DR (P7 Port)	0027	ADCCRB (AD Control B)	
0008		P8DR (P8 Port)	0028	ADCDRL (AD Conversion Value, 2 Low-order Bits)	-
0009		P9DR (P9 Port)	0029	ADCDRH (AD Conversion Value, 8 High-order Bits)	-
000A		P0CR (P0 Input/Output Control)	002A	EIRC (Extended Interrupt Enable, High)	
000B		P1CR (P1 Input/Output Control)	002B	ILC (Extended interrupt Latch, High)	
000C		HPWMCR (HPWM Control)	002C	EIRE (Extended Interrupt Enable, Low)	
000D		HPWMDR0 (HPWM0 Data)	002D	EIRD (Extended Interrupt Enable, Middle)	
000E		HPWMDR1 (HPWM1 Data)	002E	ILE (Extended Interrupt Latch, Low)	
000F		TC1CR (Timer 1 Control)	002F	ILD (Extended Interrupt Latch, Middle)	
0010		TC1DRAL (Timer 1 Register A, Lower)	0030	CGCR (First Divider Stage Input Clock Select)	
0011		TD1DRAH (Timer 1 Register A, Upper)	0031	reserved	
0012		TC1DRBL (Timer 1 Register B, Lower)	0032	reserved	
0013		TC1DRBH (Timer 1 Register B, Upper)	0033	reserved	
0014		CTC1CRL (CTC1 Control, Lower)	0034	-	WDTCT1 (WDT Control 1)
0015		CTC2CRH (CTC1 Control, Upper)	0035	-	WDTCT2 (WDT Control 2)
0016		CTC1DRL (CTC Compare Timer Register A, B, C, Lower)	0036	TBTCR (TBT/TG/Divider Output Control)	
0017		CTC1DRH (CTC Compare Timer Register A, B, C, Upper)	0037	EINTCR (External Interrupt Input Control)	
0018	reserved		0038	SYSCR1 (System Control 1)	
0019	reserved		0039	SYSCR2 (System Control 2)	
001A		TC4CR (Timer 4 Control)	003A	EIRL (Interrupt Enable, Lower)	
001B		TC4DR (Timer 4 Register)	003B	EIRH (Interrupt Enable, Upper)	
001C		TC3DRA (Timer 3 Register A)	003C	ILL (Interrupt Latch, Lower)	
001D		TC3DRB (Timer 3 Register B: TC3C)	003D	ILH (Interrupt Latch, Upper)	
001E		TC3CR (Timer 3 Control)	003E	PSWL (Program Status Word, Lower)	
001F		reserved	003F	PSWH (Program Status Word, Upper)	

Note 1: Do not access the reserved addresses in a program.
 Note 2: Marked with - cannot be accessed.
 Note 3: When defining address 0003F_H with a symbol, define it as GPSW/GRBS.
 Note 4: The write-only registers and interrupt latches cannot be operated on by using read-modify-write instructions (bit manipulating instructions such as SET or CLR, and arithmetic instructions such as AND or OR).

Figure 2-1 (a). Special Function Registers (SFR)

Address	Read	Write	Address	Read	Write
1F80	P0ODE (P0 Open-drain Control)		1F90	UARTSEL (Set UART pin)	
1F81	-		1F91	UARTSR (UART Status)	UARTCRA (UART Control 1)
1F82	-		1F92	-	
1F83	P3ODE (P3 Open-drain Control)		1F93	RDBUF (UART Receive Buffer)	TDBUF (UART Transmit Buffer)
1F84	P4ODE (P4 Open-drain Control)		1F94	-	
1F85	P5ODE (P5 Open-drain Control)		1F95	-	
1F86	P8ODE (P8 Open-drain Control)		1F96	-	
1F87	P9ODE (P9 Open-drain Control)		1F97	SIOSR (SIO Status)	SIOCR1 (SIO Control Register 1)
1F88	-		1F98	0	0
1F89	P3CR (P3 Input/output Control)		1F99	.	.
1F8A	P4CR (P4 Input/output Control)		1F9A	.	.
1F8B	P5CR (P5 Input/output Control)		1F9B	SIO Receive Buffer	SIO Transmit Buffer
1F8C	P6CR (P6 Input/output Control)		1F9C	.	.
1F8D	P7CR (P7 Input/output Control)		1F9D	.	.
1F8E	P8CR (P8 Input/output Control)		1F9E	.	.
1F8F	P9CR (P9 Input/output Control)		1F9F	7	7

Figure 2-1 (b). Data Buffer Registers (DBR) (1/2)

PMD Related (Ch 1)		Read	Write	PMD Related (Ch 2)		Read	Write
1FA0		PDCRA (Position Detection Control A)		1FD0		PDCRA (Position Detection Control A)	
1FA1		PDCRB (Position Detection Control B)		1FD1		PDCRB (Position Detection Control B)	
1FA2		PDCRC (Position Detection Control C)	-	1FD2		PDCRC (Position Detection Control C)	-
1FA3		SDREG (Sampling Delay Control)		1FD3		SDREG (Sampling Delay Control)	
1FA4		MTCRA (Mode Timer Control A)		1FD4		MTCRA (Mode Timer Control A)	
1FA5		MTCRB (Mode Timer Control B)		1FD5		MTCRB (Mode Timer Control B)	
1FA6		MCAPL (Mode Capture L)	-	1FD6		MCAPL (Mode Capture L)	-
1FA7		MCAPH (Mode Capture H)	-	1FD7		MCAPH (Mode Capture H)	-
1FA8		CMP1L (Compare Register 1L)		1FD8		CMP1L (Compare Register 1L)	
1FA9		CMP1H (Compare Register 1H)		1FD9		CMP1H (Compare Register 1H)	
1FAA		CMP2L (Compare Register 2L)		1FDA		CMP2L (Compare Register 2L)	
1FAB		CMP2H (Compare Register 2H)		1FDB		CMP2H (Compare Register 2H)	
1FAC		CMP3L (Compare Register 3L)		1FDC		CMP3L (Compare Register 3L)	
1FAD		CMP3H (Compare Register 3H)		1FDD		CMP3H (Compare Register 3H)	
1FAE		MDCRA (PMD Control A)		1FDE		MDCRA (PMD Control A)	
1FAF		MDCRB (PMD Control B)		1FDF		MDCRB (PMD Control B)	
1FB0		EMGCRA (EMG Control A)		1FE0		EMGCRA (EMG Control A)	
1FB1		EMGCRB (EMG Control B)		1FE1		EMGCRB (EMG Control B)	
1FB2		MDOUTL (PMD Output Register L)		1FE2		MDOUTL (PMD Output Register L)	
1FB3		MDOUTH (PMD Output Register H)		1FE3		MDOUTH (PMD Output Register H)	
1FB4		MDCNTL (PMD Counter L)	-	1FE4		MDCNTL (PMD Counter L)	-
1FB5		MDCNTH (PMD Counter H)	-	1FE5		MDCNTH (PMD Counter H)	-
1FB6		MDPRDL (PMD Period Register L)		1FE6		MDPRDL (PMD Period Register L)	
1FB7		MDPRDH (PMD Period Register H)		1FE7		MDPRDH (PMD Period Register H)	
1FB8		CMPUL (PMD Compare U Register L)		1FE8		CMPUL (PMD Compare U Register L)	
1FB9		CMPUH (PMD Compare U Register H)		1FE9		CMPUH (PMD Compare U Register H)	
1FBA		CMPVL (PMD Compare V Register L)		1FEA		CMPVL (PMD Compare V Register L)	
1FBB		CMPVH (PMD Compare V Register H)		1FEB		CMPVH (PMD Compare V Register H)	
1FBC		CMPWL (PMD Compare W Register L)		1FEC		CMPWL (PMD Compare W Register L)	
1FBD		CMPWH (PMD Compare W Register H)		1FED		CMPWH (PMD Compare W Register H)	
1FBE		DTR (Dead Time)		1FEE		DTR (Dead Time)	
1FBF		EMGREL (EMG Release Control)	-	1FEF		EMGREL (EMG Release Control)	-
1FC0		EDCR (Waveform Calculation Control)		1FF0		EDCR (Waveform Calculation Control)	
1FC1		-		1FF1		-	
1FC2		EDSETA (Waveform Calculation Period Control A)		1FF2		EDSETA (Waveform Calculation Period Control A)	
1FC3		EDSETB (Waveform Calculation Period Control B)		1FF3		EDSETB (Waveform Calculation Period Control B)	
1FC4		ELDEGA (Electrical Angle Set A)		1FF4		ELDEGA (Electrical Angle Set A)	
1FC5		ELDEGB (Electrical Angle Set B)		1FF5		ELDEGB (Electrical Angle Set B)	
1FC6		AMPA (Voltage Set A)		1FF6		AMPA (Voltage Set A)	
1FC7		AMPB (Voltage Set B)		1FF7		AMPB (Voltage Set B)	
1FC8		EDCAPA (Electrical Angle Capture Value A)		1FF8		EDCAPA (Electrical Angle Capture Value A)	
1FC9		EDCAPB (Electrical Angle Capture Value B)		1FF9		EDCAPB (Electrical Angle Capture Value B)	
1FCA		WFMDR (Sine Wave RAM Access)		1FFA		WFMDR (Sine Wave RAM Access)	
1FCB		-		1FFB		-	


 : Shows the registers associated with electrical angle timer and waveform calculation.

Figure 2-1 (b). Data Buffer Registers (DBR) (2/2)

2.2 Input/output Ports

The TMP88CS43 contains 10 input/output ports comprised of 71 pins.

- ① Port P0 ; 4-bit input/output port (timer/counter input, serial interface input/output, and high-speed PWM output)
- ② Port P1 ; 8-bit input/output port (external interrupt input, timer/counter input/output, divider output, and motor control circuit input)
- ③ Port P2 ; 3-bit input/output port (external interrupt input, timer/counter input/output, and STOP mode release signal input)
- ④ Port P3 ; 8-bit input/output port (motor control input/output)
- ⑤ Port P4 ; 8-bit input/output port (timer/counter output, serial interface input/output, motor control circuit input)
- ⑥ Port P5 ; 8-bit input/output port (motor control circuit input/output)
- ⑦ Port P6 ; 8-bit input/output port (analog input and motor control circuit output)
- ⑧ Port P7 ; 8-bit input/output port (analog input and motor control circuit output)
- ⑨ Port P8 ; 8-bit input/output port
- ⑩ Port P9 ; 8-bit input/output port

All output ports contain a latch, and the output data therefore are retained by the latch. But none of the input ports have a latch, so it is desirable that the input data be retained externally until it is read out, or read several times before being processed. Figure 2-2 shows input/output timing.

The timing at which external data is read in from input/output ports is S1 state in the read cycle of instruction execution. Because this timing cannot be recognized from the outside, transient input data such as chattering needs to be dealt with in a program. The timing at which data is forwarded to input/output ports is S2 state in the write cycle of instruction execution.

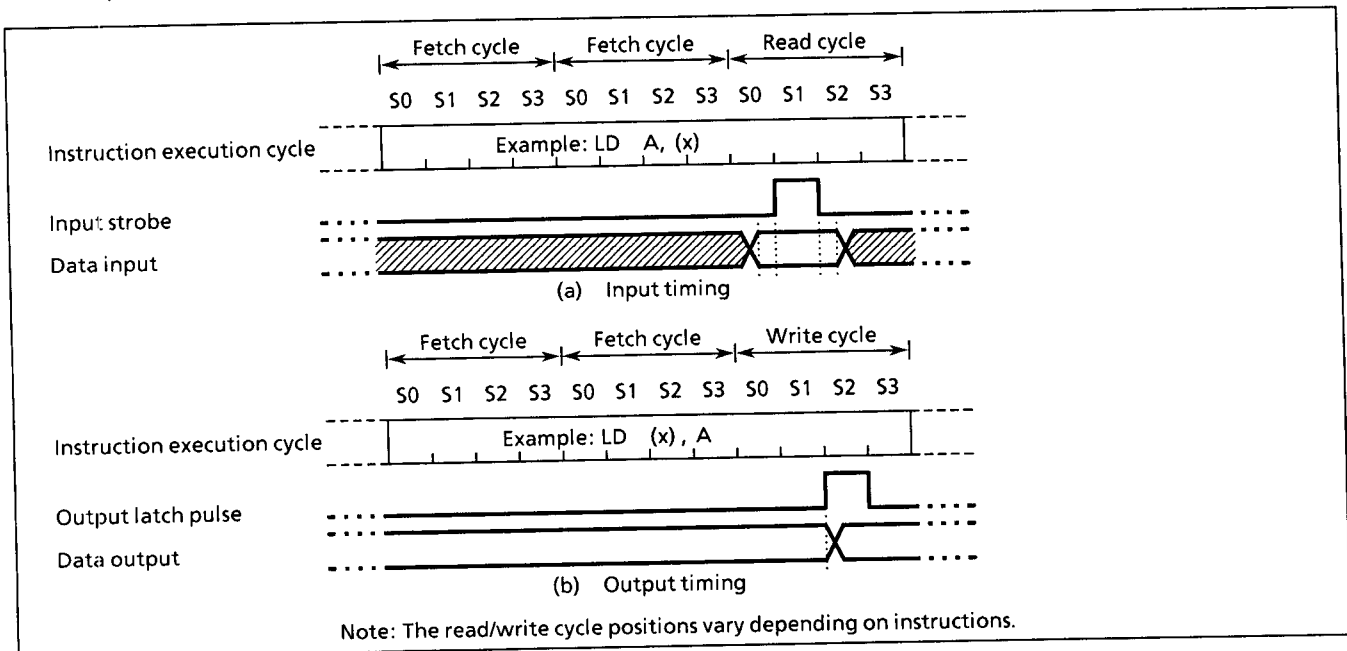


Figure 2-2. Example of Input/Output Timing

When an operation is performed for read from any input/output port except programmable input/output ports, whether the input value of the pin or the content of the output latch is read depends on the instruction executed, as shown below.

(1) Instructions which read the content of the output latch

- ① XCH r, (src)
- ② SET/CLR/CPL (src).b
- ③ SET/CLR/CPL (pp).g
- ④ LD (src).b, CF
- ⑤ LD (pp).b,CF
- ⑥ XCH CF, (src), b
- ⑦ ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), n
- ⑧ ADD/ADDC/SUB/SUBB/AND/OR/XOR (src),(HL) instructions, the (src) side thereof
- ⑨ MXOR (src), m

(2) Instructions which read the input value of the pin

Any instructions other than those listed above and ADD/ADDC/SUB/SUBB/AND/OR/XOR (src),(HL) instructions, the (HL) side thereof

2.2.1 Port P0 (P03 to P00)

Port P0 is a 4-bit input/output port shared with serial interface input/output. This port is switched between input and output modes using the P0 Port Input/Output Control Register (P0CR). When reset, the P0CR Register is initialized to 0, with the P0 port set for input mode. Also, the Output Latch (P0DR) is initialized to 0 when reset.

The P0 port contains bitwise programmable open-drain control. The P0 Port Open-drain Control Register (P0ODE) is used to select open-drain or tri-state mode for the port. When reset, the P0ODE Register is initialized to 0, with tri-state mode selected for the port.

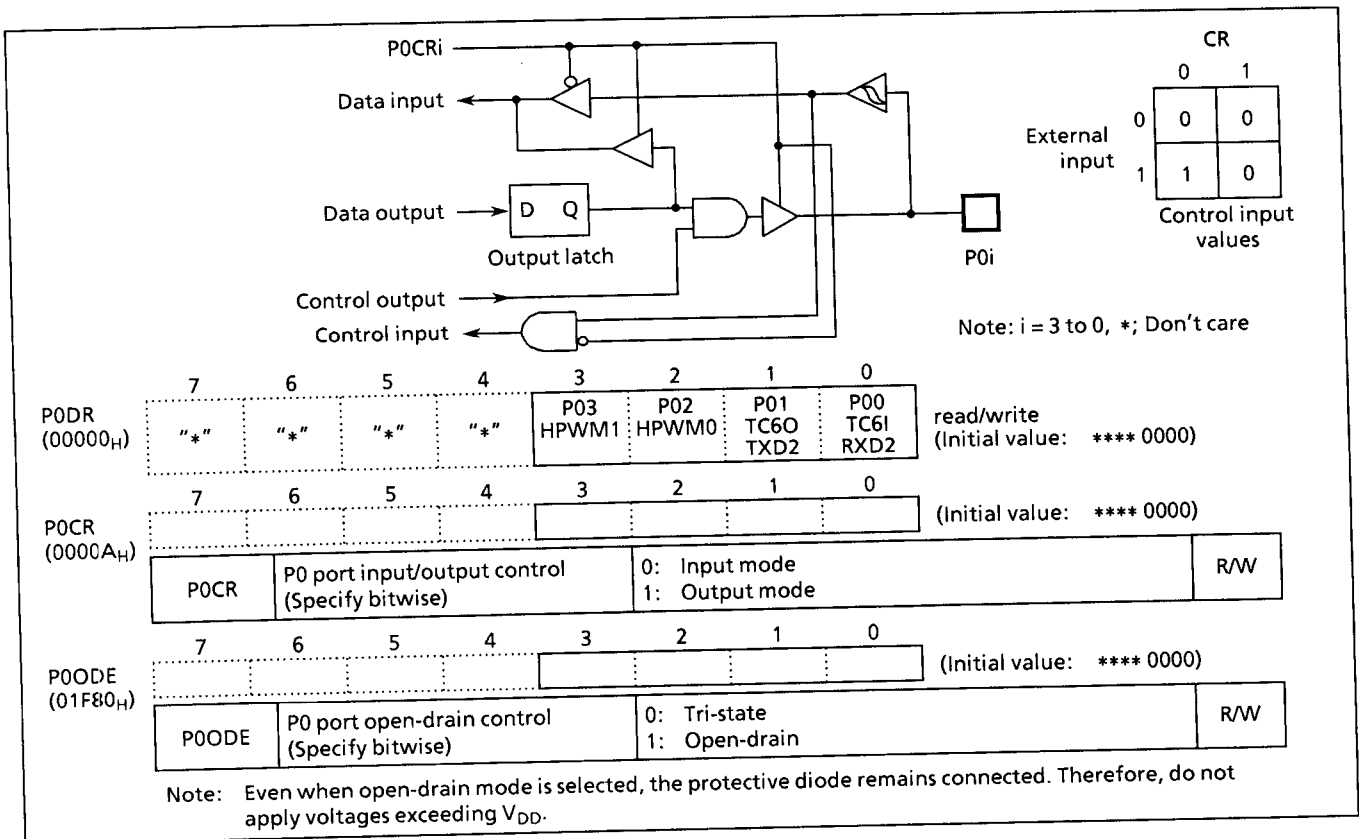


Figure 2-3. Port P0 and P0 Port Input/Output Registers

2.2.2 Port P1 (P17 to P10)

Port P1 is an 8-bit input/output port shared with external interrupt input, timer/counter input/output, and divider output. This port is switched between input and output modes using the P1 Port Input/output Control Register (P1CR). When reset, the P1CR Register is initialized to 0, with the P1 port set for input mode. Also, the Output Latch (P1DR) is initialized to 0 when reset.

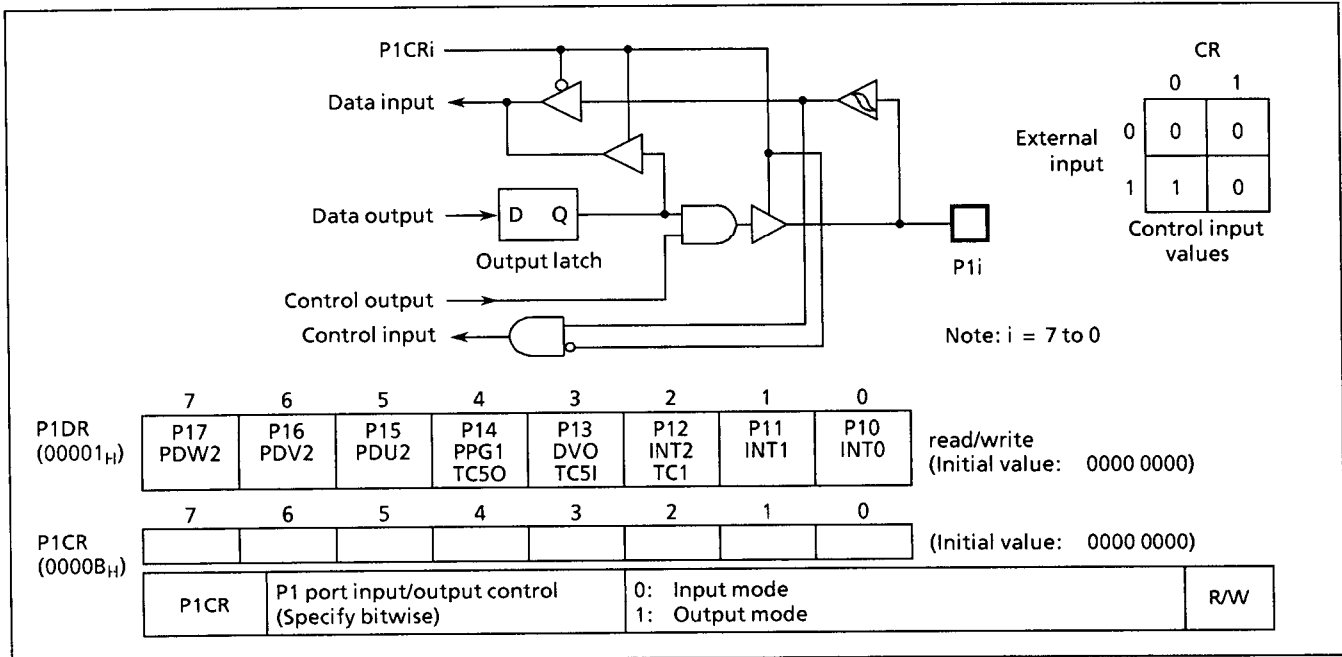


Figure 2-4. Port P1 and P1 Port Input/Output Register

2.2.3 Port P2 (P22 to P20)

Port P2 is a 3-bit input/output port shared with external interrupt input and STOP mode release signal. When using this port as these functional pins or an input port, set the output latch to 1. When reset, the output latch is initialized to 1.

We recommend using the P20 pin as external interrupt input, STOP mode release signal input, or input port. When using this port as an output port, note that the interrupt latch is set by a falling edge of output pulse. Note also that outputs on this port during STOP mode go to a high-impedance state.

When a read instruction is executed on P2 port, indeterminate values are read in from bits 7 to 3.

When any read-modify-write instruction in 2.2 (1) is executed on P2 port, the content of the output latch is read out. When any other instruction is executed, the external pin state is read out.

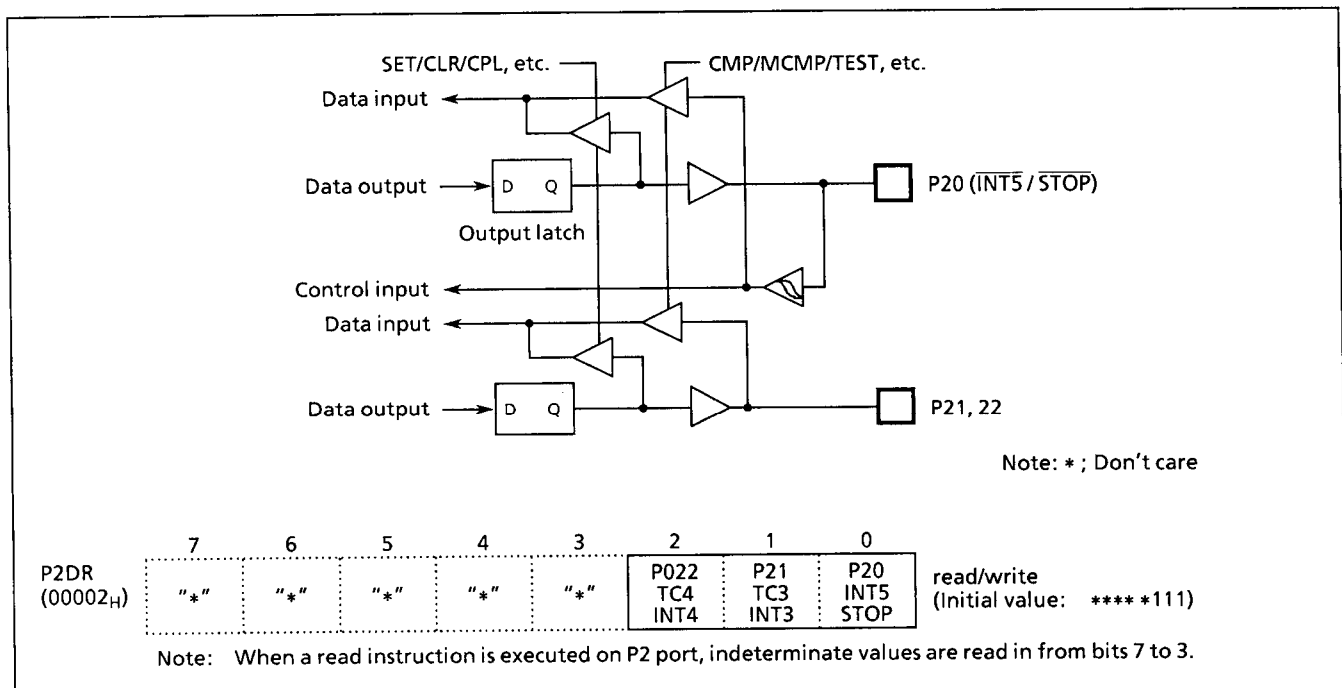


Figure 2-5. Port P2 and P2 Port Input/Output Register

2.2.4 Port P3 (P37 to P30)

Port P3 is an 8-bit input/output port. This port is switched between input and output modes using the P3 Port Input/output Control Register (P3CR). When reset, the P3CR Register is initialized to 0, with the P3 port set for input mode. Also, the Output Latch (P3DR) is initialized to 0 when reset.

The P3 port contains bitwise programmable open-drain control. The P3 Port Open-drain Control Register (P3ODE) is used to select open-drain or tri-state mode for the port. When reset, the P3ODE Register is initialized to 0, with tri-state mode selected for the port.

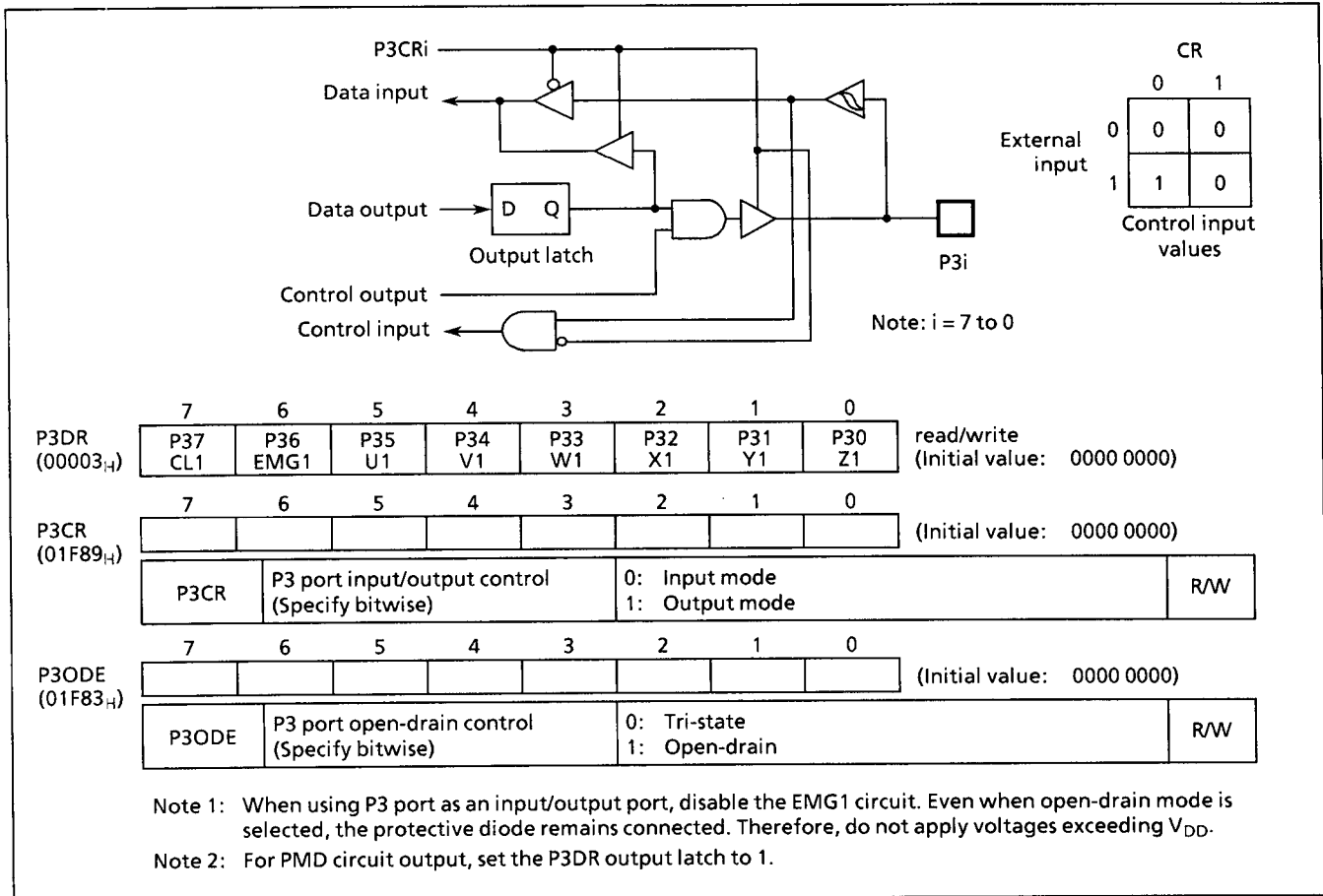


Figure 2-6. Port P3 and P3 Port Input/Output Registers

2.2.5 Port P4 (P46 to P40)

Port P4 is an 8-bit input/output port shared with serial interface input/output. This port is switched between input and output modes using the P4 Port Input/output Control Register (P4CR). When reset, the P4CR Register is initialized to 0, with the P4 port set for input mode. Also, the Output Latch (P4DR) is initialized to 0 when reset.

The P4 port contains bitwise programmable open-drain control. The P4 Port Open-drain Control Register (P4ODE) is used to select open-drain or tri-state mode for the port. When reset, the P4ODE Register is initialized to 0, with tri-state mode selected for the port.

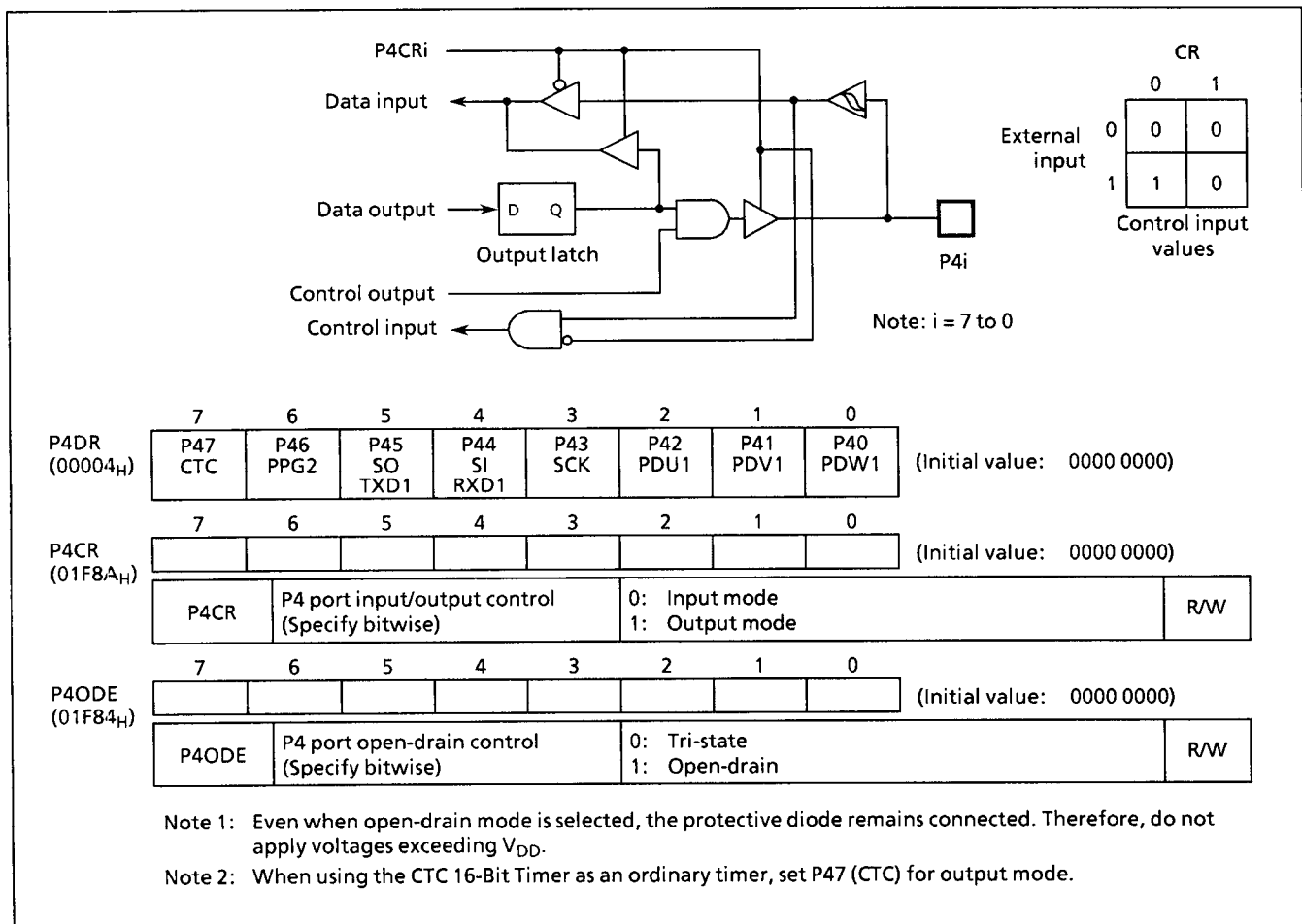


Figure 2-7. Port P4 and P4 Port Input/Output Registers

2.2.6 Port P5 (P57 to P50)

Port P5 is an 8-bit input/output port. This port is switched between input and output modes using the P5 Port Input/output Control Register (P5CR). When reset, the P5CR Register is initialized to 0, with the P5 port set for input mode. Also, the Output Latch (P5DR) is initialized to 0 when reset.

The P5 port contains bitwise programmable open-drain control. The P5 Port Open-drain Control Register (P5ODE) is used to select open-drain or tri-state mode for the port. When reset, the P5ODE Register is initialized to 0, with tri-state mode selected for the port.

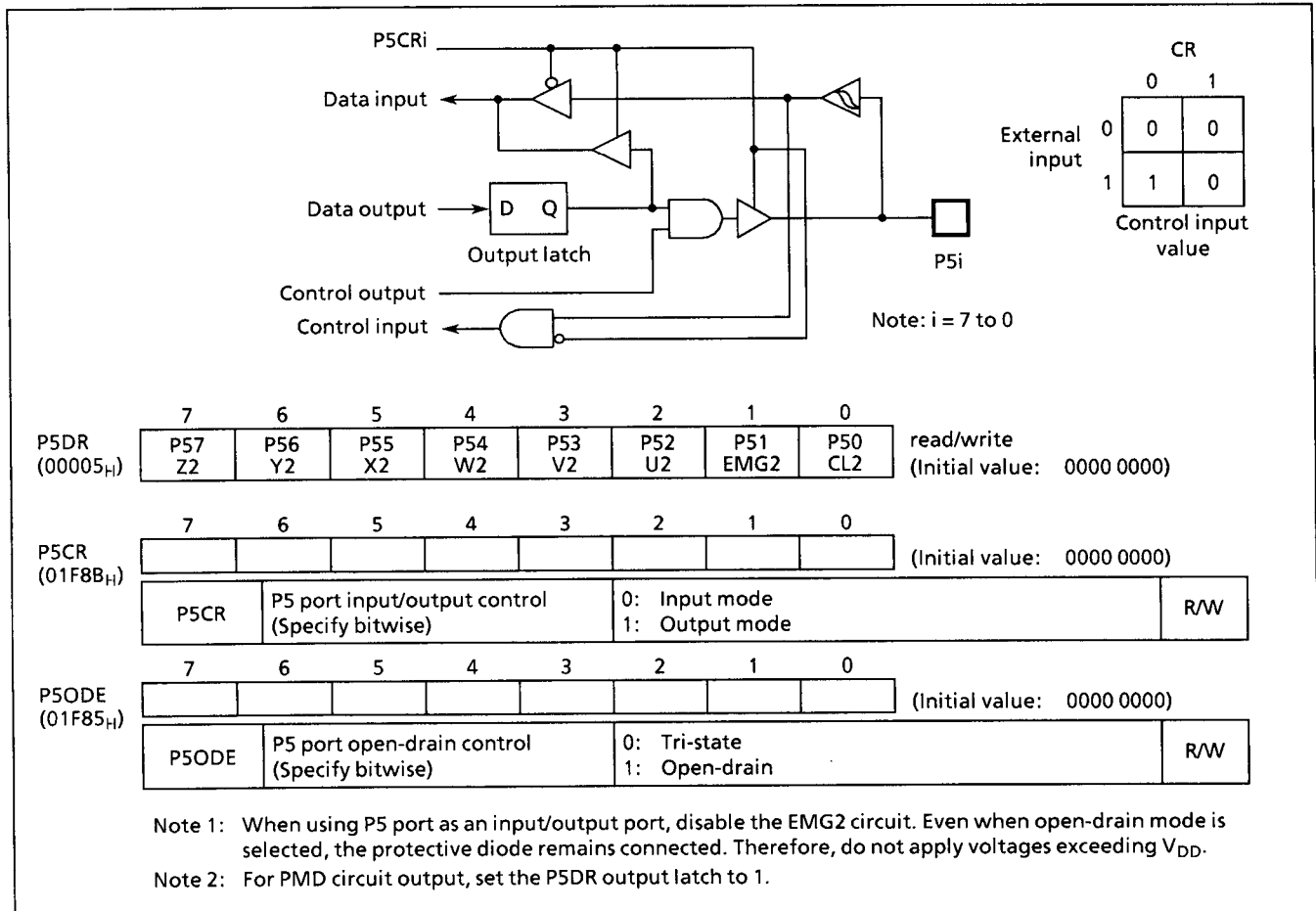


Figure 2-8. Port P5 and P5 Port Input/Output Registers

2.2.7 Port P6 (P67 to P60)

Port P6 is an 8-bit input/output port shared with AD converter analog input. This port is switched between input and output modes using the P6 Port Input/output Control Register (P6CR), P6 Port Output Latch (P6DR), and AINDS (ADCCRA Register bit 4). When reset, the P6CR Register and the P6DR Output Latch are initialized to 0 while AINDS is set to 1, so that P67 to P60 have their inputs fixed low (= 0). When using the P6 port as an input port, set the corresponding bits for input mode (P6CR = 0, P6DR = 1). When using the port as an output port, set the P6CR Register's corresponding bits to 1. When using the port for analog input, set the corresponding bits for analog input (P6CR = 0, P6DR = 0). Then set AINDS = 0, and AD conversion will start.

The reason why the output latch = 0 is because it is necessary to prevent current from flowing into the shared digital input circuit. Therefore, the ports used for analog input must have their output latches set to 0 beforehand. The actual input channels for AD conversion are selected using SAIN (ADCCRA Register bits 3 to 0).

Although the bits of P6 port not used for analog input can be used as input/output ports, do not execute output instructions on these ports during AD conversion. This is necessary to maintain the accuracy of AD conversion. Also, do not apply rapidly changing signals to ports adjacent to analog input during AD conversion.

If an input instruction is executed while the P6DR output latch is cleared to 0, data "0" is read in from said bits.

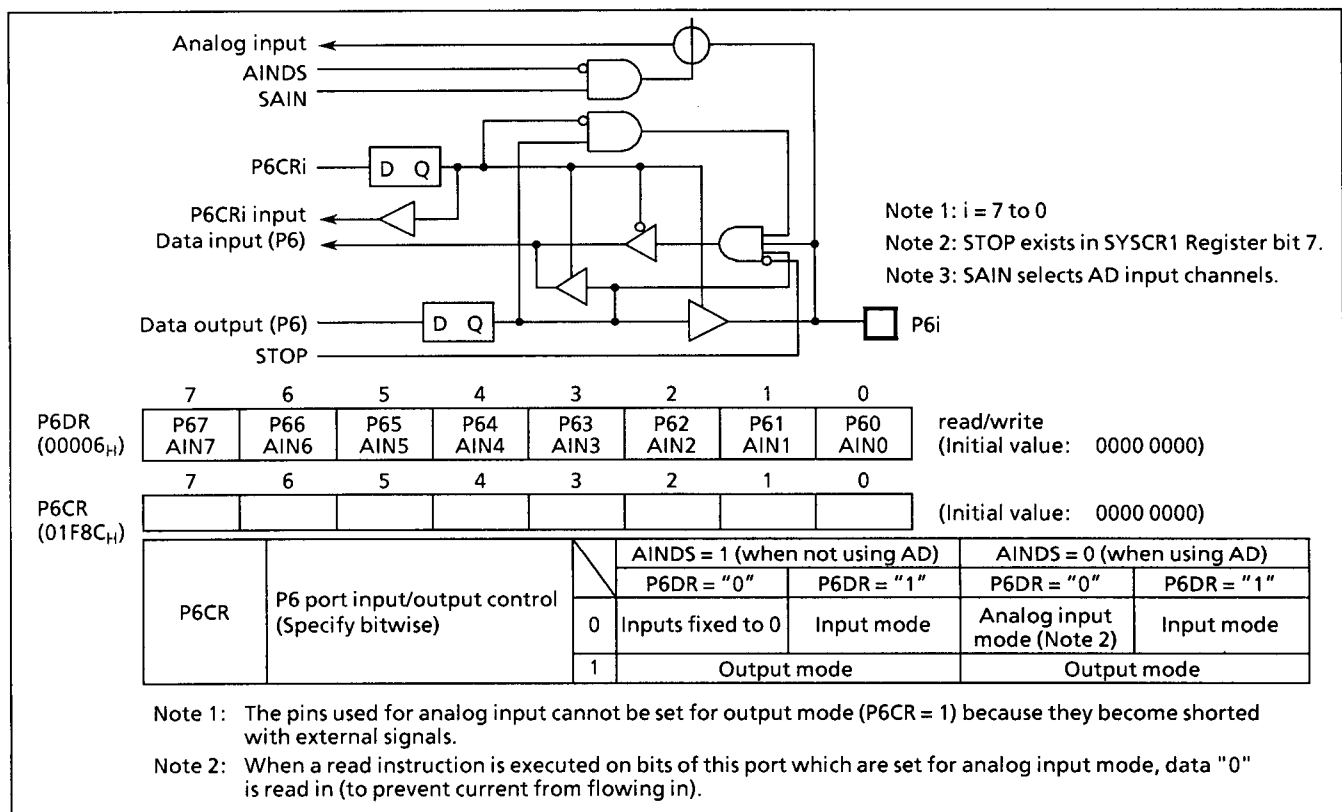


Figure 2-9. Port P6 and P6 Port Input/Output Registers

Note: When using this port in input mode (including analog input), do not use bit manipulating or other read-modify-write instructions. When a read instruction is executed on the bits of this port that are set for input, the contents of the pins are read in, so that if a read-modify-write instruction is executed, their output latches may be rewritten, making the pins unable to accept input. (A read-modify-write instruction first reads data from all of the eight bits and after modifying them (bit manipulation), writes data for all of the eight bits to the output latches.)

2.2.8 Port P7 (P77 to P70)

Port P7 is an 8-bit input/output port shared with AD converter analog input. This port is switched between input and output modes using the P7 Port Input/output Control Register (P7CR), P7 Port Output Latch (P7DR), and AINDS (ADCCRA Register bit 4). When reset, the P7CR Register and the P7DR Output Latch are initialized to 0 while AINDS is set to 1, so that P77 to P70 have their inputs fixed low (= 0). When using the P7 port as an input port, set the corresponding bits for input mode (P7CR = 0, P7DR = 1). When using the port as an output port, set the P7CR Register's corresponding bits to 1. When using the port for analog input, set the corresponding bits for analog input (P7CR = 0, P7DR = 0). Then set AINDS = 0, and AD conversion will start.

The reason why the output latch = 0 is because it is necessary to prevent current from flowing into the shared digital input circuit. Therefore, the ports used for analog input must have their output latches set to 0 beforehand. The actual input channels for AD conversion are selected using SAIN (ADCCRA Register bits 3 to 0).

Although the bits of P7 port not used for analog input can be used as input/output ports, do not execute output instructions on these ports during AD conversion. This is necessary to maintain the accuracy of AD conversion. Also, do not apply rapidly changing signals to ports adjacent to analog input during AD conversion.

If an input instruction is executed while the P7DR output latch is cleared to 0, data "0" is read in from said bits.

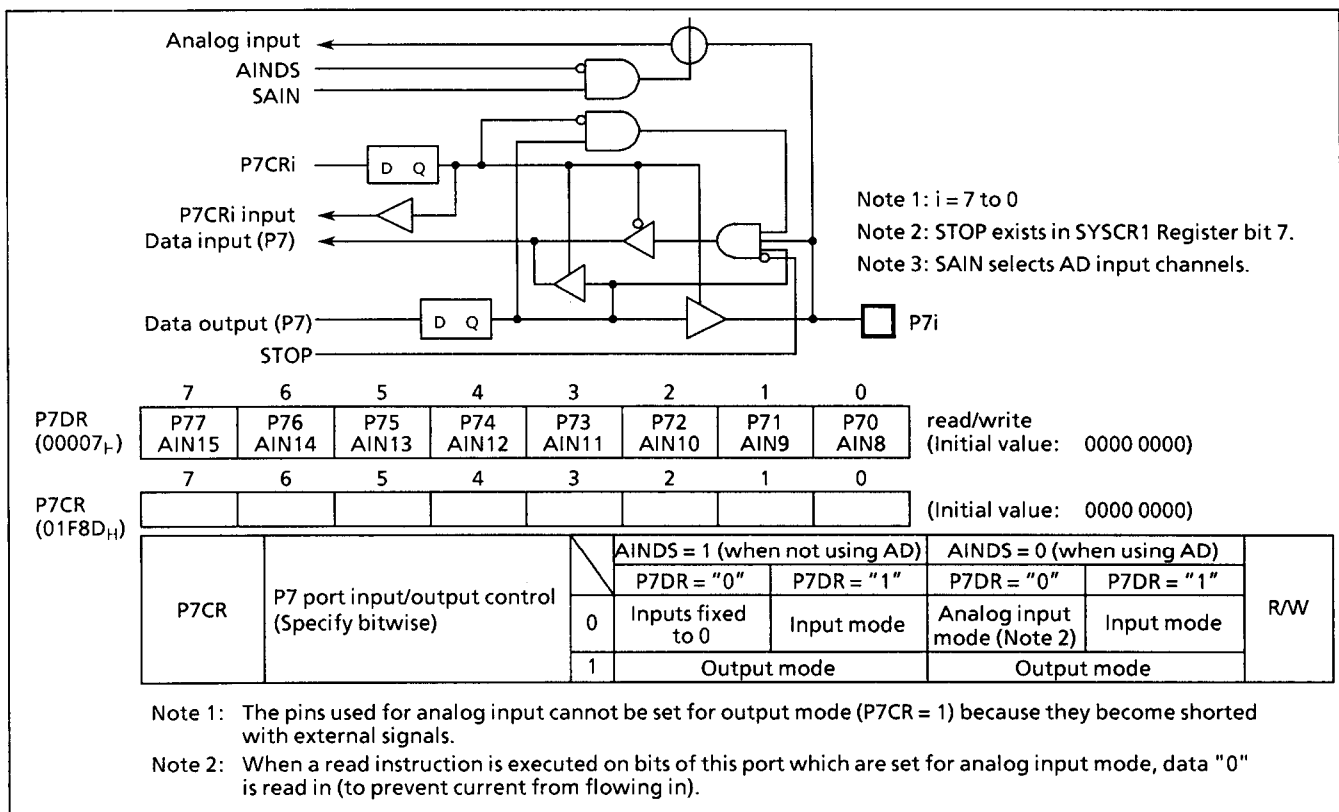


Figure 2-10. Port P7 and P7 Port Input/Output Registers

Note: When using this port in input mode (including analog input), do not use bit manipulating or other read-modify-write instructions. When a read instruction is executed on the bits of this port that are set for input, the contents of the pins are read in, so that if a read-modify-write instruction is executed, their output latches may be rewritten, making the pins unable to accept input. (A read-modify-write instruction first reads data from all of the eight bits and after modifying them (bit manipulation), writes data for all of the 8 bits to the output latches.)

2.2.9 Port P8 (P87 to P80)

Port P8 is an 8-bit input/output port. This port is switched between input and output modes using the P8 Port Input/output Control Register (P8CR). When reset, the P8CR Register is initialized to 0, with the P8 port set for input mode. Also, the Output Latch (P8DR) is initialized to 0 when reset.

The P8 port contains bitwise programmable open-drain control. The P8 Port Open-drain Control Register (P8ODE) is used to select open-drain or tri-state mode for the port. When reset, the P8ODE Register is initialized to 0, with tri-state mode selected for the port.

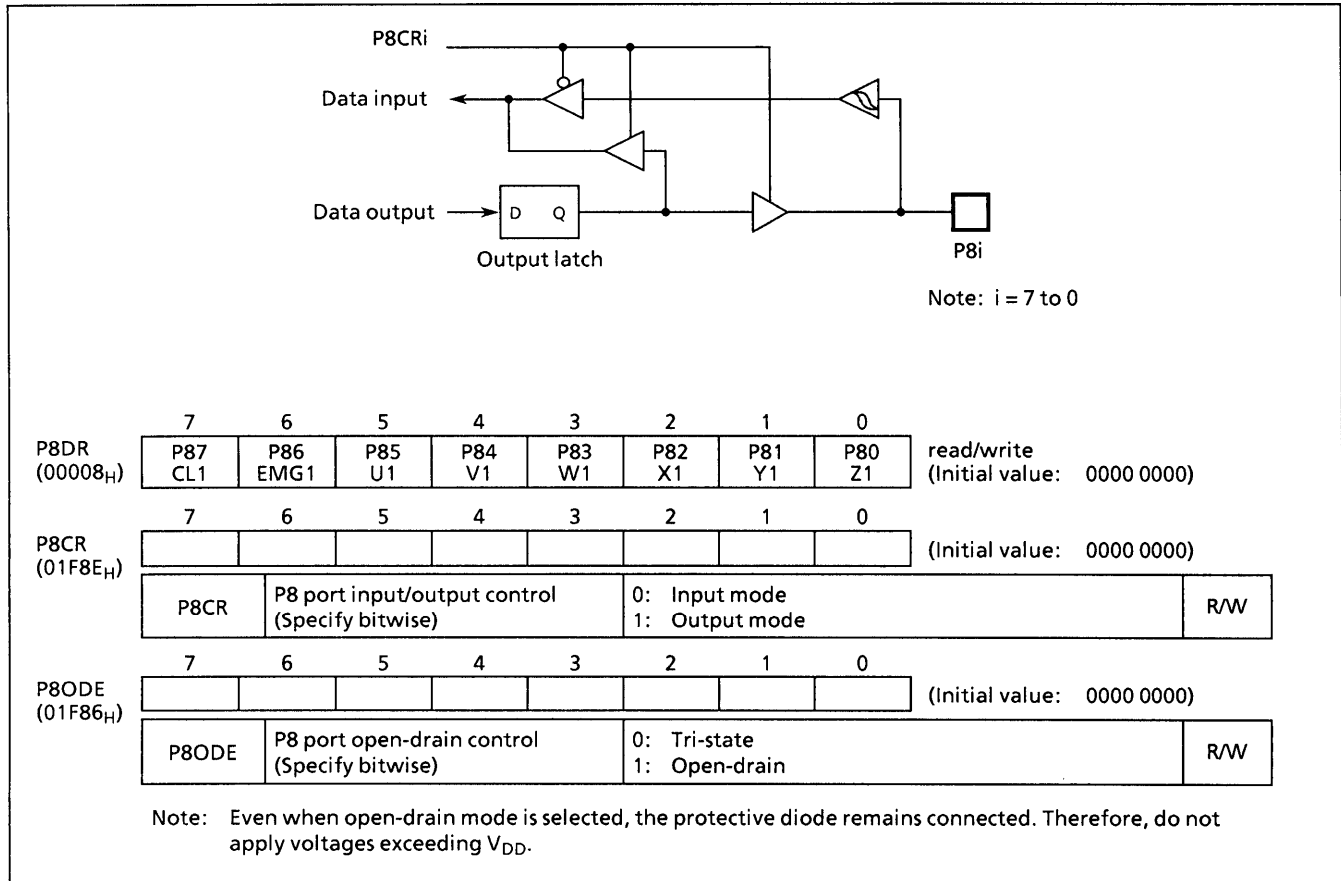


Figure 2-11. Port P8 and P8 Port Input/Output Registers

2.2.10 Port P9 (P97 to P90)

Port P9 is an 8-bit input/output port. This port is switched between input and output modes using the P9 Port Input/output Control Register (P9CR). When reset, the P9CR Register is initialized to 0, with the P9 port set for input mode. Also, the Output Latch (P9DR) is initialized to 0 when reset.

The P9 port contains bitwise programmable open-drain control. The P9 Port Open-drain Control Register (P9ODE) is used to select open-drain or tri-state mode for the port. When reset, the P9ODE Register is initialized to 0, with tri-state mode selected for the port.

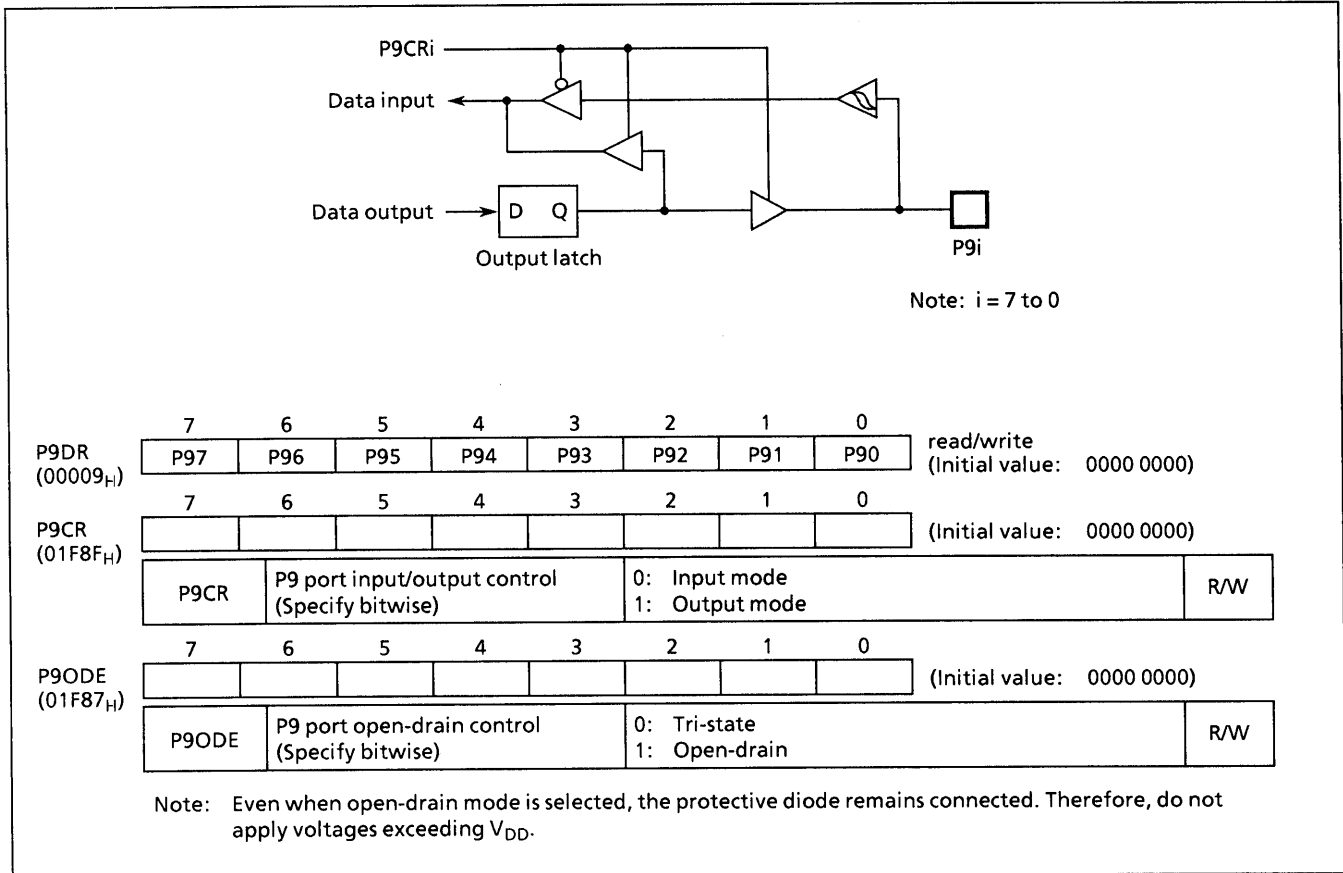


Figure 2-12. Port P9 and P9 Port Input/Output Registers

2.3 Time Base Timer (TBT)

The Time Base Timer is used to produce the reference time for key scan and dynamic display processing and for this purpose generates a time base timer interrupt (INTTBT) at fixed intervals.

A time base timer interrupt is generated beginning with the first rising edge of the source clock (the timing generator's divider output selected by TBTCK) after enabling the Time Base Timer. Because the divider is not cleared in a program, the first INTTBT interrupt, only the first one though, may be generated earlier than the set interrupt interval. (See Figure 2-13 (b).)

When selecting an interrupt frequency, make sure the Time Base Timer is disabled. (Do not change the set interrupt frequency when disabling the Time Base Timer while it is active.) It is possible to select an interrupt frequency while at the same time enabling the Time Base Timer.

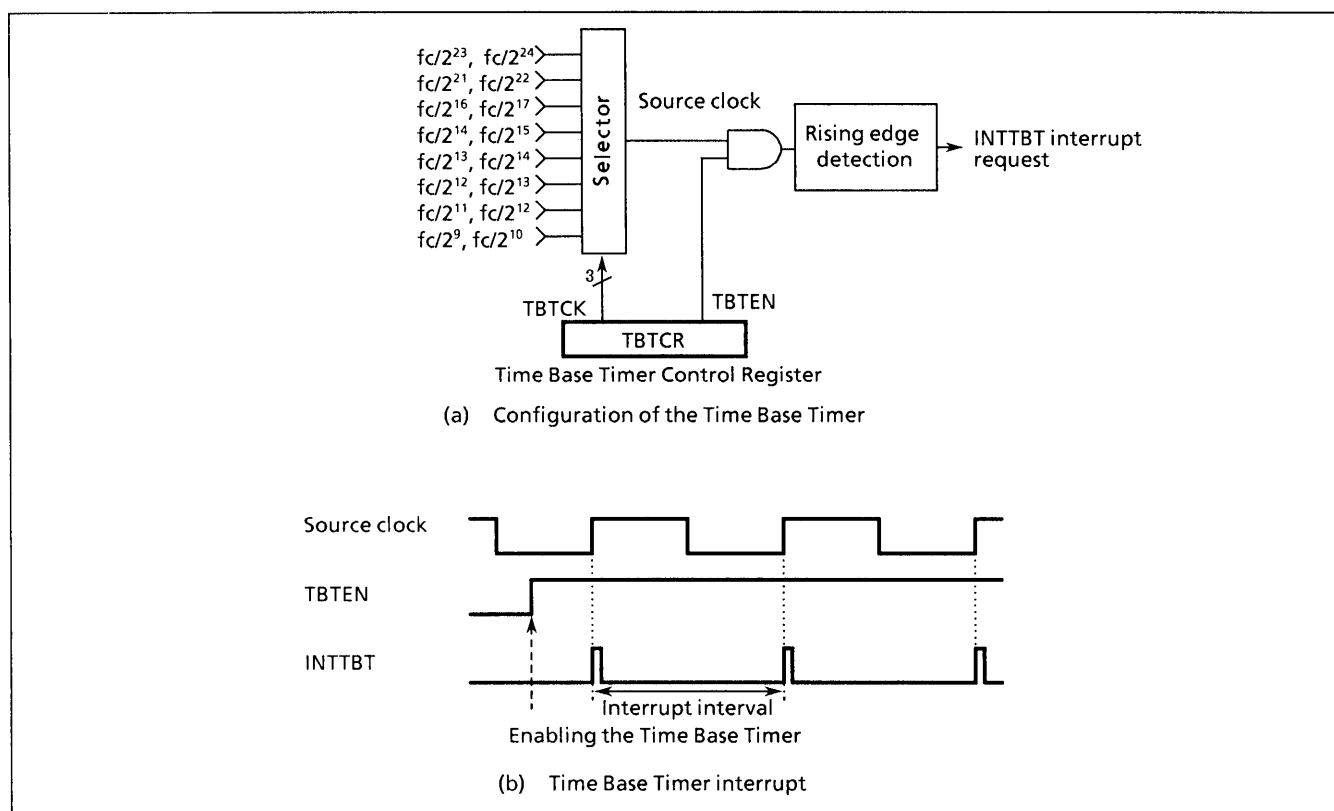


Figure 2-13. Time Base Timer

Example: Setting the Time Base Timer interrupt frequency to $fc/2^{16}$ [Hz] and enabling the INTTBT interrupt

```
LD (TBTCCR), 00001010B
SET (EIRL). 6
```

TBTCR (00036 _H)	7	6	5	4	3	2	1	0	(Initial value: 0**0 0***)
	(DVOEN)	(DVOCK)	0	TBTEN	TBTCR				
TBTEN	Enable/disable time base timer			0: Disable 1: Enable				R/W	
TBTCR	Select time base timer interrupt frequency Unit: Hz	/		NORMAL1 and IDLE1 modes					
				DV1CK = 0		DV1CK = 1			
		000	$fc/2^{23}$		$fc/2^{24}$				
		001	$fc/2^{21}$		$fc/2^{22}$				
		010	$fc/2^{16}$		$fc/2^{17}$				
		011	$fc/2^{14}$		$fc/2^{15}$				
		100	$fc/2^{13}$		$fc/2^{14}$				
		101	$fc/2^{12}$		$fc/2^{13}$				
		110	$fc/2^{11}$		$fc/2^{12}$				
		111	$fc/2^9$		$fc/2^{10}$				
Note: fc; the high-frequency clock [Hz], *; Don't care									

Figure 2-14. Time Base Timer Control Register

Table 2-1. Time Base Timer Interrupt Frequency (Example: fc = 20 MHz)

TBTCR	Time Base Timer Interrupt Frequency [Hz]	
	NORMAL1 and IDLE1 modes	
	DV1CK = 0	DV1CK = 1
000	2.38	1.20
001	9.53	4.78
010	305.18	153.50
011	1220.70	610.35
100	2441.40	1220.70
101	4882.83	2441.40
110	9765.63	4882.83
111	39063.00	19531.25

2.3.1 Divider Output (\overline{DVO})

By using the divider of the timing generator, it is possible to produce a 50% duty cycle pulse which can be used for buzzer drive, etc. The divider output is fed to the outside from the P13 (\overline{DVO}) pin. For the P13 port, set its output latch to 1 before setting it for output mode.

TBTCR (00036 _H)	7	6	5	4	3	2	1	0	(Initial value: 0**0 0***)
	(DVOEN)	(DVOCK)	0	(TBTEN)	(TBTCR)				
DVOEN	Enable/disable divider output			0: Disable 1: Enable				R/W	
DVOCK	Select divider output (\overline{DVO} pin) frequency			00: $fc/2^{13}$					
				01: $fc/2^{12}$					
				10: $fc/2^{11}$					
				01: $fc/2^{10}$					
Note: fc; the high-frequency clock [Hz], *; Don't care									

Figure 2-15. Time Base Timer Control Register

Example: Producing a 2.44 kHz pulse (when $f_c = 20$ MHz)

```

SET (P1).3 ; P13 output latch ← 1
LD (P1CR), 00001000B ; Set P13 for output mode
LD (TBTCCR), 10000000B ; DVOEN ← 1, DVOCK ← 00
    
```

Table 2-2. Divider Output Frequency

DVOCK	Divider Output Frequency	When $f_c = 20$ MHz
00	$f_c/2^{13}$	2.441 [kHz]
01	$f_c/2^{12}$	4.883 [kHz]
10	$f_c/2^{11}$	9.766 [kHz]
11	$f_c/2^{10}$	19.531 [kHz]

2.4 Watchdog Timer (WDT)

The Watchdog Timer is a fail-safe function which when the CPU operates erratically (or runs out of control) or becomes locked up for reasons of noise, etc., detects the fault condition as soon as possible and returns the CPU to normal condition.

The runaway detection signal to be output by the Watchdog Timer can be a reset output or a pseudo-nonmaskable interrupt request as selected in a program. However, this setting is effective only once. When reset, this setting is initialized and a reset output is selected.

When not using the Watchdog Timer for runaway detection, it can be used as a timer to generate an interrupt at fixed intervals.

2.4.1 Configuration of the Watchdog Timer

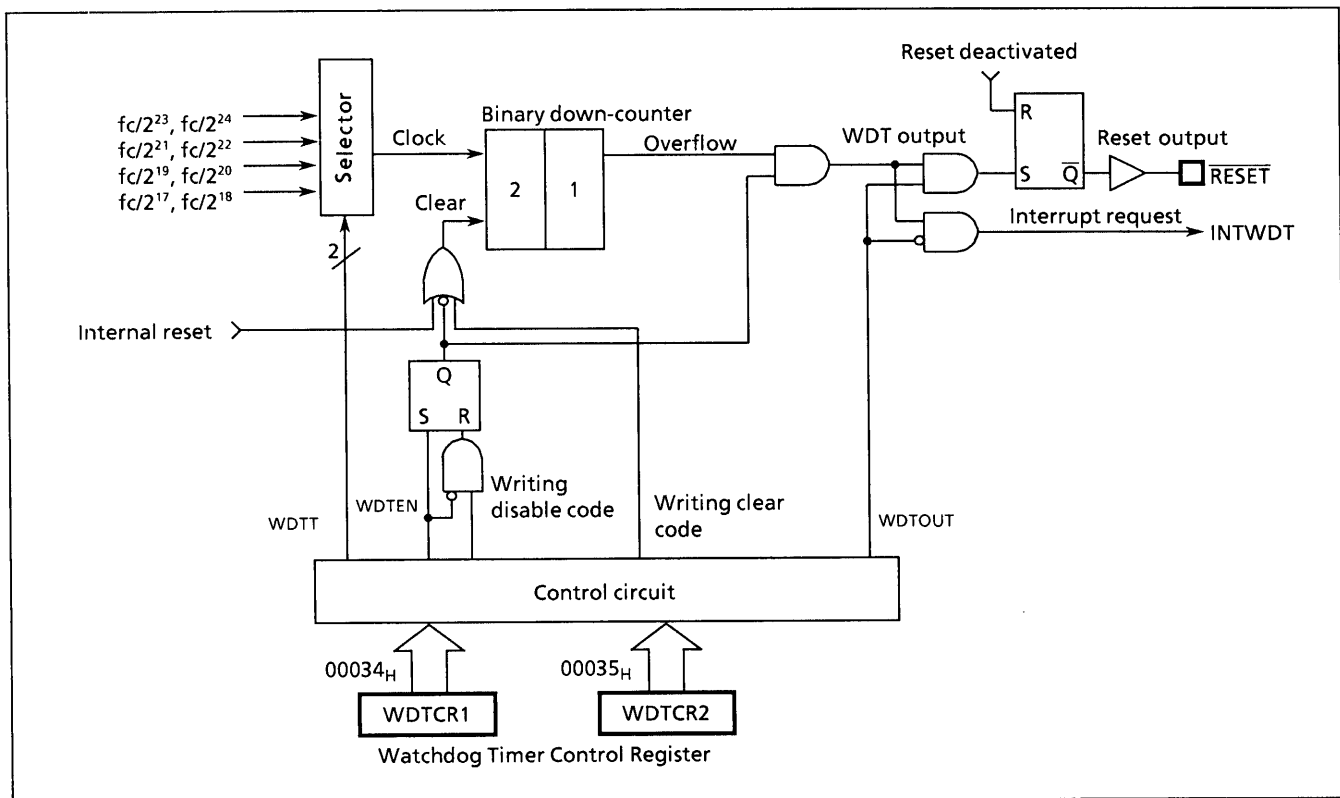


Figure 2-16. Configuration of the Watchdog Timer

2.4.2 Controlling the Watchdog Timer

The Watchdog Timer Control Registers are shown in Figure 2-17. After reset, the Watchdog Timer is disabled.

If affected by disturbing noise, etc., the Watchdog Timer may not be able to display its full function. Take this into consideration when designing your application system.

(1) Detecting runaway condition using the Watchdog Timer

To detect a runaway condition of the CPU, follow the procedure described below.

- ① Set the detection time, select output, and clear the binary counter.
- ② Repeatedly clear the binary counter within every detection time that is set.

If the CPU runs out of control or locks up for some reason and the binary counter cannot be cleared, an overflow signal from the binary counter activates the Watchdog Timer output. If $WDTOUT = 1$ at this time, the internal hardware is reset. If $WDTOUT = 0$, a watchdog timer interrupt (INTWDT) is generated.

During STOP mode (including warming-up) or IDLE mode, the Watchdog Timer temporarily stops counting up and after exiting STOP or IDLE mode, automatically restarts (continues counting up).

Example: Setting the Watchdog Timer detection time to $221/f_c$ [s] and resetting runaway detection

	LD	(WDTCR2), 4EH	; Clear the binary counter
	LD	(WDTCR1), 00001101B	; WDTT ← 10, WDTOUT ← 1
Within 3/4 of WDT detection time	[LD	(WDTCR2), 4EH ; Clear the binary counter (Always clear immediately before and after changing WDTT)
		LD	(WDTCR2), 4EH ; Clear the binary counter
Within 3/4 of WDT detection time	[LD	(WDTCR2), 4EH ; Clear the binary counter
		LD	(WDTCR2), 4EH ; Clear the binary counter

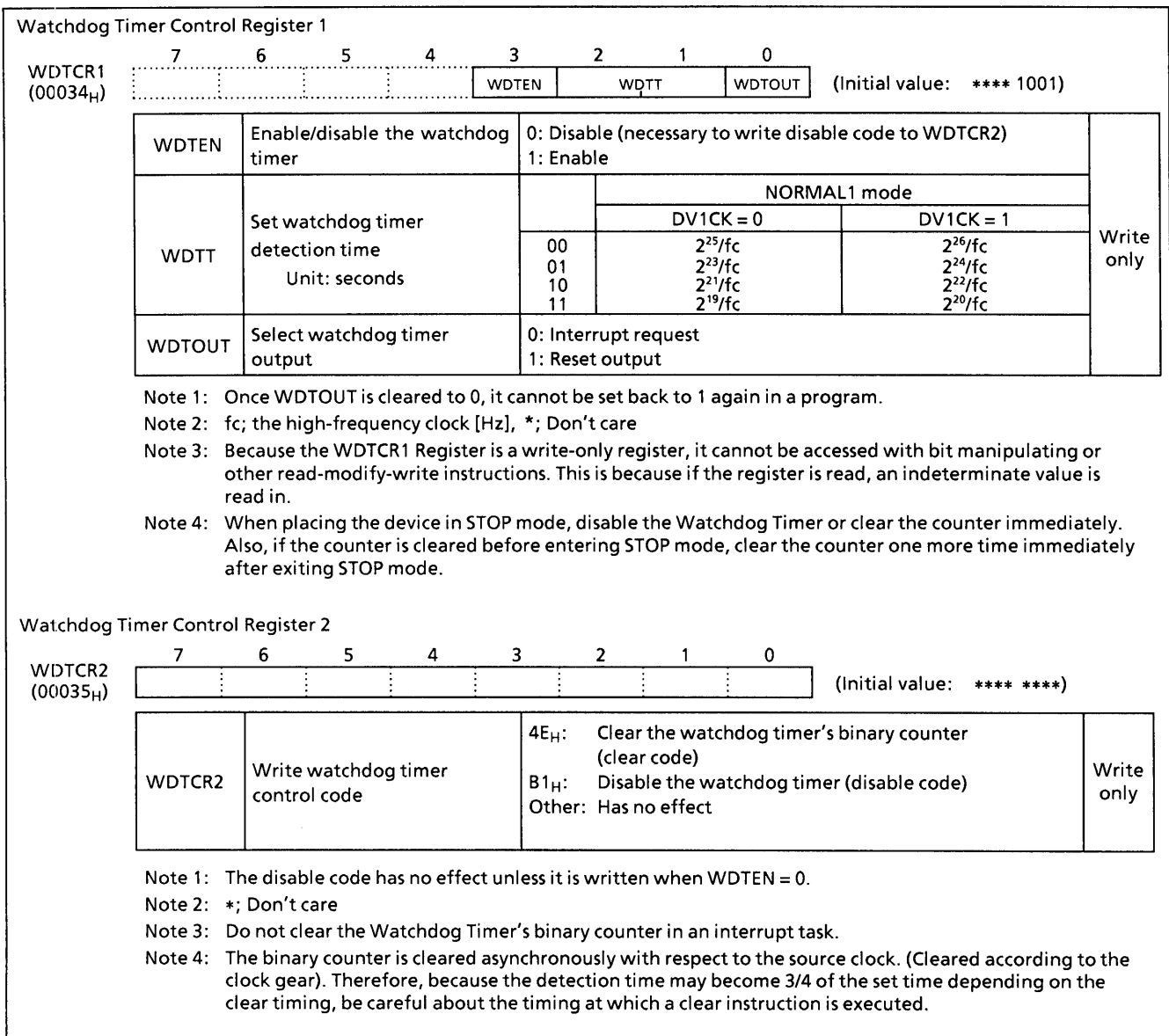


Figure 2-17. Watchdog Timer Control Registers

(2) Enabling the Watchdog Timer

Set WDTEN (WDTCR1 Register bit 3) to 1 to enable the Watchdog Timer. When reset, WDTEN is initialized to 1, so that the Watchdog Timer starts operating immediately after reset.

(3) Disabling the Watchdog Timer

After clearing WDTEN (WDTCR1 Register bit 3) to 0, write disable code (B1_H) to the WDTCR2 Register to disable the Watchdog Timer. Conversely, the Watchdog Timer cannot be disabled by writing disable code (B1_H) to the WDTCR2 Register before clearing WDTEN to 0. While the Watchdog Timer is disabled, its binary counter remains cleared.

Example: Disabling the Watchdog Timer

```
LDW (WDTCR1), 0B101H ; WDTEN←0, WDTCR2←disable code
```

Table 2-3. Watchdog Timer Detection Time (Example: $f_c = 20$ MHz)

WDT	Watchdog Timer Detection Time [s]	
	NORMAL1 mode	
	DV1CK = 0	DV1CK = 1
00	1.678	3.355
01	419.430 m	838.861 m
10	104.858 m	209.715 m
11	26.214 m	52.429 m

Note: If the Watchdog Timer is disabled during watchdog timer interrupt processing, the watchdog timer interrupt will never be cleared. Therefore, clear the Watchdog Timer before disabling it, or disable the Watchdog Timer a sufficient time before it overflows.

2.4.3 Watchdog Timer Interrupt (INTWDT)

This is a pseudo-nonmaskable interrupt which is always accepted no matter how the interrupt enable register is set. However, if this interrupt occurs while the preceding watchdog timer interrupt or a software interrupt is already being serviced, it is kept waiting for acceptance until the processing under way is finished (RETN instruction execution finished).

Note that before the watchdog timer output can be selected to be an interrupt request with WDTOUT, the stack pointer must first be set.

Example: Setting the watchdog timer interrupt

```
LD SP, 0023FH ; Set SP
LD (WDTCR1), 00001000B ; WDTOUT←0
```

2.4.4 Watchdog Timer Reset

This signal resets the internal hardware. The reset time is $8/f_c$ to $24/f_c$ [s] (16 to $48 \mu s$ at $f_c = 20$ MHz, $f_c = f_c/16$). The \overline{RESET} pin is a sink open-drain input/output with a pull-up resistor included.

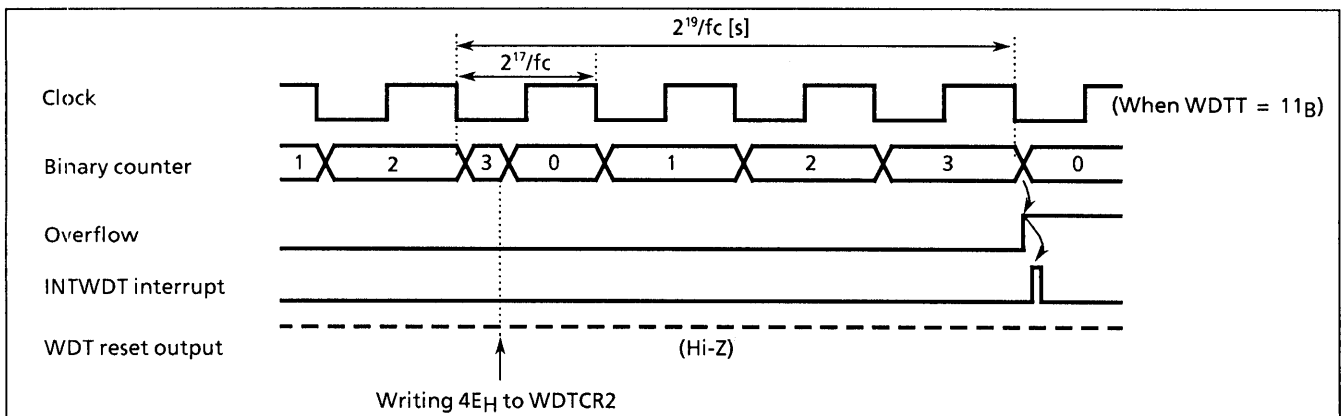


Figure 2-18. Watchdog Timer Interrupt and Reset

2.5 Divider Output (DVO)

By using the divider of the timing generator, it is possible to produce an approximately 50% duty cycle pulse which can be used for piezoelectric buzzer drive, etc. The divider output is fed to the outside from the P13 (DVO) pin.

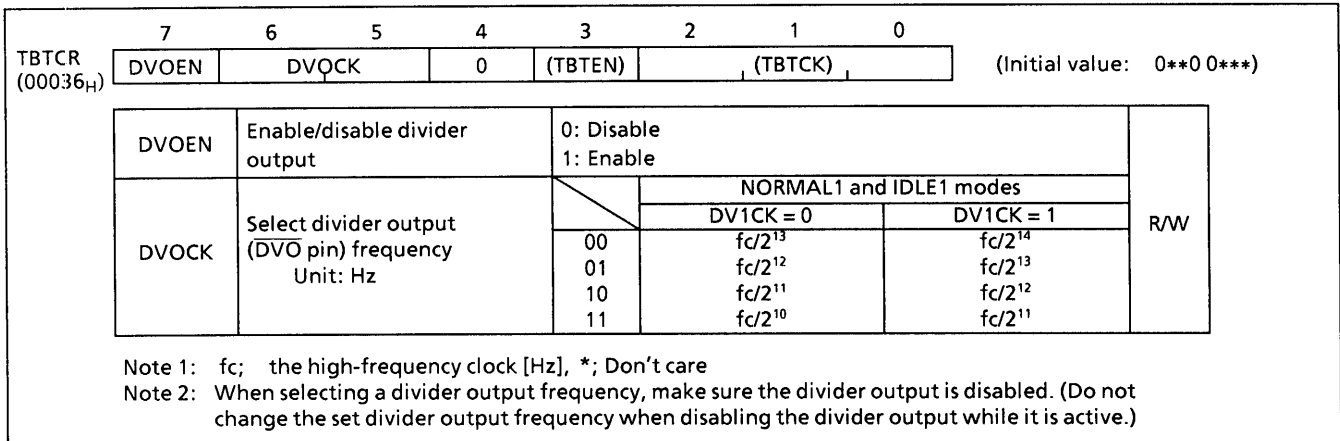


Figure 2-19. Divider Output Control Register

Example: When producing a 488.3 Hz pulse (when $fc = 20$ MHz, DV1CK = 1)

LD (TBTCR), 11000000B ; DVOEN←1, DVQCK←10

Table 2-4. Divider Output Frequency (Example: $fc = 20$ MHz)

DVQCK	Divider Output Frequency [Hz]	
	NORMAL1, IDLE1 Mode	
	DV1CK = 0	DV1CK = 1
00	2.4415 k	1.22075 k
01	4.8825 k	2.4415 k
10	9.765 k	4.8825 k
11	19.5325 k	9.765 k

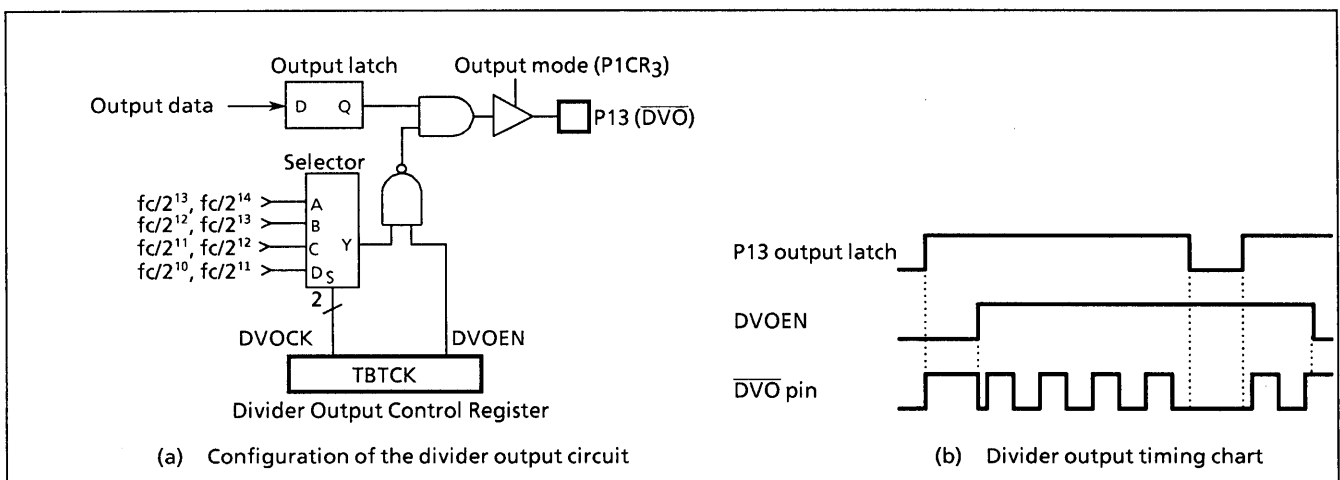


Figure 2-20. Divider Output

2.6 16-Bit Timer/Counter 1 (TC1)

2.6.1 Configuration

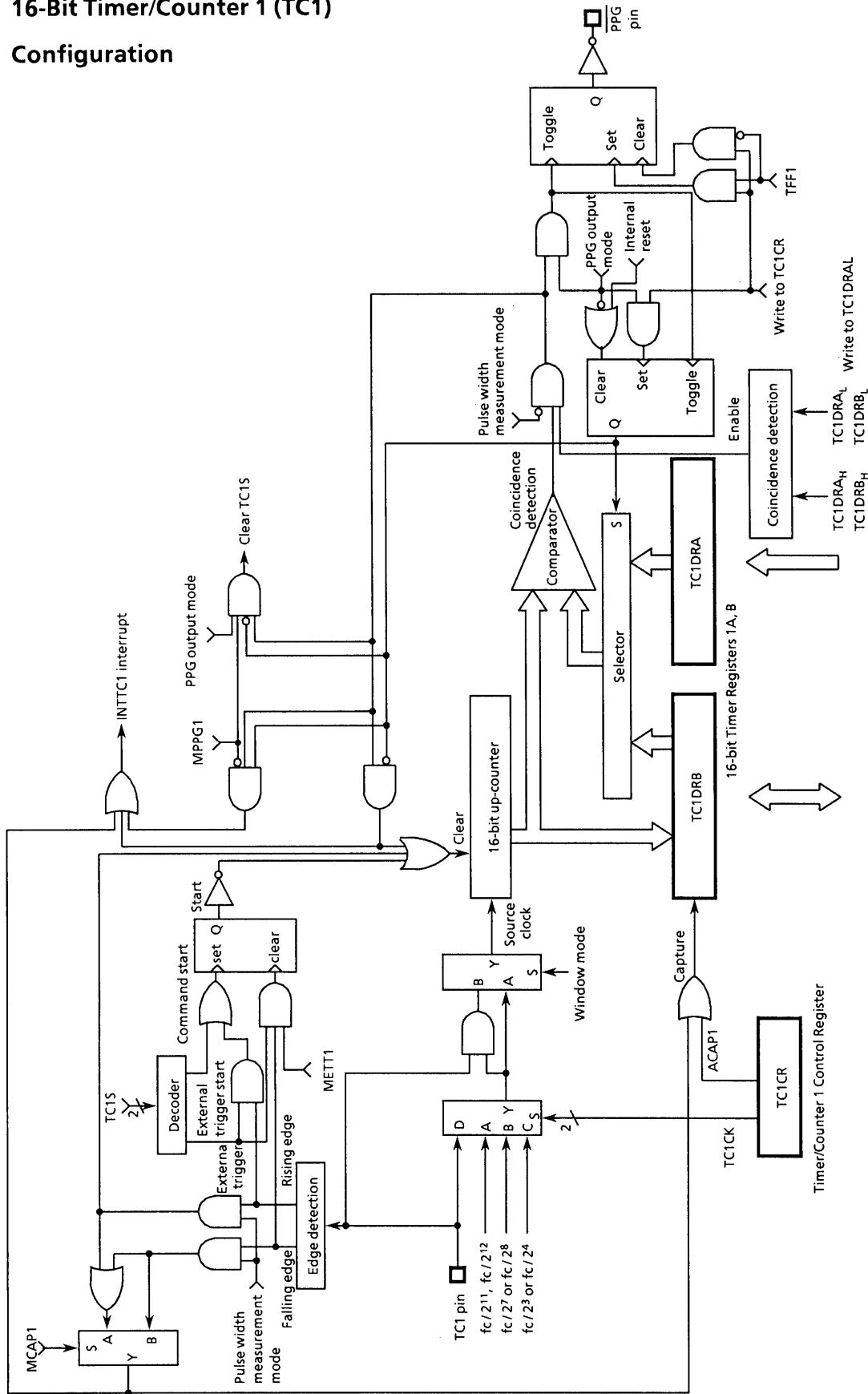


Figure 2-21. Timer/Counter 1 (TC1)