CMOS 8-Bit Microcontroller

TMP88CP77F, TMP88CS77F, TMP88CU77F

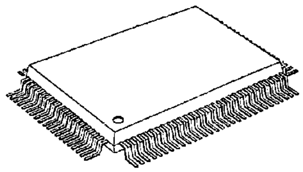
The TMP88CP77/S77/U77 are the high speed and high performance 8-bit single chip microcomputers. These MCU contain VFT (Vacuum Fluorescent Tube) driver, serial interface, 8-bit AD converter and multi-function timer/counter on a chip.

PART No.	ROM	RAM	PACKAGE	OTP MCU
TMP88CP77F	48K × 8 bit + 256 × 8 bit	1K × 8-bit	P-QFP100-1420-0.65A	TMP88PU77F
TMP88CS77F	64K × 8 bit + 256 × 8 bit	2K × 8-bit		
TMP88CU77F	96K × 8 bit + 256 × 8 bit	3K × 8-bit		

Features

- ◆8-bit single-chip microcomputer TLCS-870/X series microcomputer
- ◆interrupt sources: 20 (6 external, 14 internal)
- ◆I/O ports: 88 pins
- ◆Three 16-bit Timer/Counters
  - ● TC1: Timer, Event counter, Pulse width measurement, External trigger timer, Window modes
  - ● TC2: Timer, Event counter, Window modes
  - ● ETC1: Timer, Event counter, Window mode
    Minimum resolution:500 μs at 8 MHz
    Two capture inputs (edge-selectable)
    Two compare outputs
    (High/Low/Toggle/Steady output modes)
- ◆8-bit Timer/Counter
  - ● TC4: Timer, Event counter, PWM output, Programmable divider output modes
- ◆Time Base Timer (Interrupt frequency: 1 Hz to 16384 Hz)
- ◆Divider output function (frequency: 1 kHz to 8 kHz)

P-QFP100-1420-0.65A



TMP88CP77F
TMP88CS77F
TMP88CU77F
TMP88PU77F

070122EBP

- The information contained herein is subject to change without notice. 021023_D
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
  In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc. 021023_A
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer's own risk. 021023_B
- The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations. 060106_Q
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patents or other rights of TOSHIBA or the third parties. 070122_C
- The products described in this document are subject to foreign exchange and foreign trade control laws. 060925_E
- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions. 030619_S

◆Watchdog Timer

- Interrupt source/reset output (programmable)

◆Two 8-bit Serial Interfaces

- With 8 bytes and 32 bytes transmits/receive data buffer
- Internal/external serial clock, and 4/8-bit mode

◆Serial bus Interface

- I^{2}C-bus, 8-bit SIO modes

◆8-bit successive approximate type AD converter with sample and hold

- 12 analog inputs
- Conversion time: 23 μs at 8 MHz

◆Vacuum Fluorescent Tube Driver (automatic display)

- Programmable grid scan
- High breakdown voltage ports (max.40 V × 53 bits)

◆Dual clock operation

- Single/Dual-clock mode

◆Five Power saving operating modes

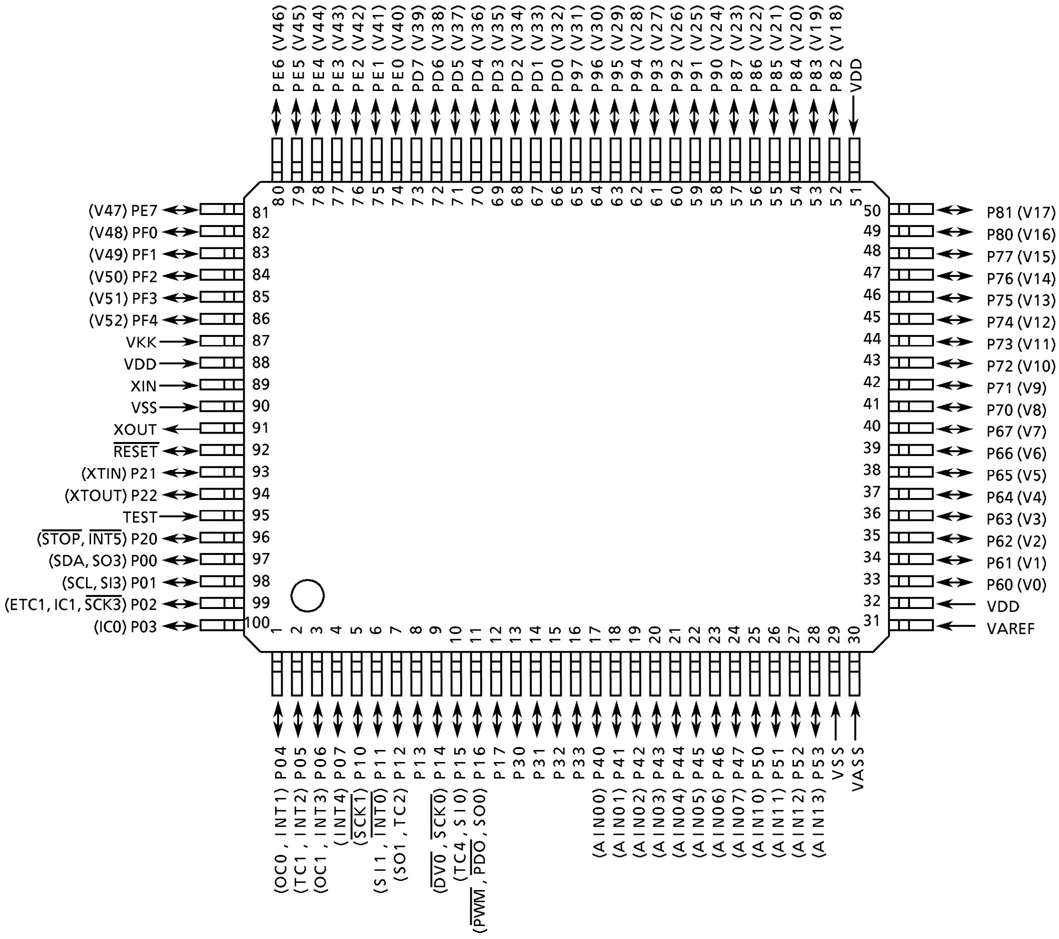
- STOP mode: Oscillation stops. Battery/Capacitor back-up. Port output hold/High-impedance.
- SLOW mode: Low power consumption operation using low-frequency clock (32.768 kHz)
- IDLE1 mode: CPU stops, and Peripherals operate using high-frequency clock. Release by interrupts.
- IDLE2 mode: CPU stops, and Peripherals operate using high-and low-frequency clock. Release by interruput.
- SLEEP mode: CPU stops, and Peripherals operate using low-frequency clock. Release by interrupts.

◆Wide operating voltage: 4.5 to 5.5 V at 12.5 MHz / 32.768 kHz, 2.7 to 5.5 V at 32.768 kHz

◆Emulation Pod: BM87CP77FOA

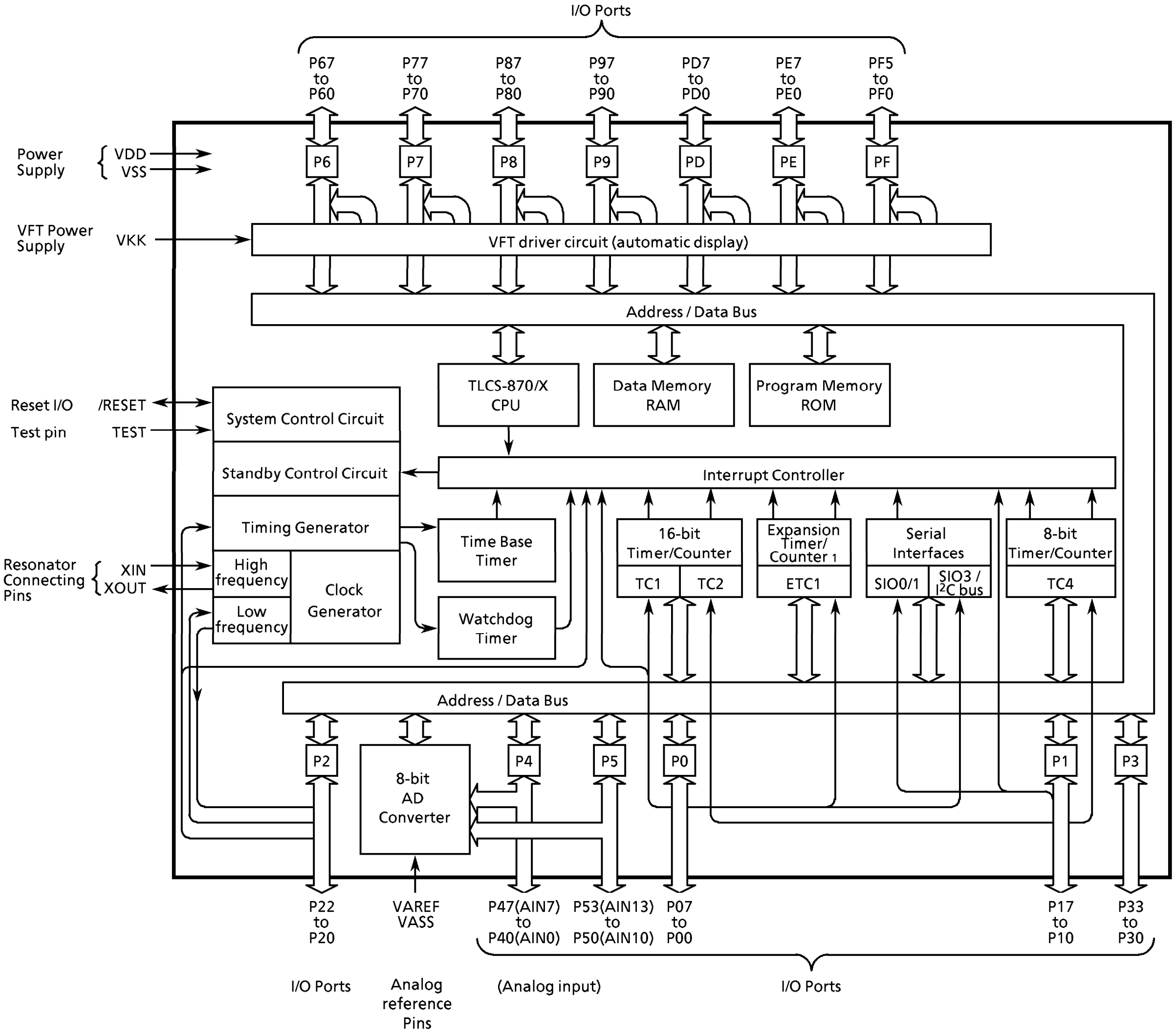
Pin Assignments (Top View)

P-QFP100-1420-0.65A



Note: All VDDs should be connected externally for keeping the same voltage level.

Block Diagram



Pin Function

Pin Name	I/O	Function	
P07 (INT4)	I/O (input)	8-bit input/output port with latch. When used as input port, multi function timer/counter, external interrupt input, serial bus interface, the latch must be set to "1".	External interrupt input 4
P06 (INT3, OC1)	I/O (I/O)		External interrupt input 3 or Expansion Timer/Counter compare output 1
P05 (TC1, INT2)	I/O (input)		Timer/Counter 1 input or External interrupt input 2
P04 (OC0, INT1)	I/O (I/O)		Expansion Timer/Counter compare output 0 or External interrupt input 1
P03 (IC0)	I/O (input)		Expansion Timer/Counter capture input 0
P02 (IC1, ETC1, $\overline{SCK3}$)	I/O (I/O)		Expansion Timer/Counter capture input 1 or Expansion Timer/Counter 1 input or SIO3 serial clock input/output
P01 (SCL, SI3)	I/O (I/O)		I²C bus serial clock input/output or SIO3 serial data input
P00 (SDA, SO3)	I/O (I/O)		I²C bus serial data input/output or SIO3 serial data output
P17	I/O	8-bit programmable input/output ports (tri-state). Each bit of this port can be individually configured as an input or an output under software control. When used as serial data input, external interrupt input, timer/counter input, the input mode is configured. When used as divider output, timer/counter output, serial data output, serial clock output, the latch must be set to "1" and the output mode is configured.	
P16 ($\overline{PWM}$, $\overline{PDO}$, SO0)	I/O (output)		8-bit PWM output or 8-bit programmable divider output or SIO0 serial data output
P15 (TC4, SI0)	I/O (input)		Timer/Counter 4 input or SIO0 serial data input
P14 ($\overline{DVO}$, $\overline{SCK0}$)	I/O (I/O)		Divider output or SIO0 serial clock input/output
P13	I/O		
P12 (SO1, TC2)	I/O (I/O)		SIO1 serial data output or Timer/Counter 2 input
P11 (SI1, $\overline{INT0}$)	I/O (input)		SIO1 serial data input or External interrupt 0 input
P10 ($\overline{SCK1}$)	I/O (I/O)		SIO1 serial clock input/output
P22 (XTOUT)	I/O (I/O)	3-bit input/output port with latch. When used as input port, external interrupt input, STOP mode release signal input, the input mode is configured.	Resonator connectiong pins (32.768 kHz). For inputting external clock, XTIN is used and XOUT is opened.
P21 (XTIN)	I/O (input)		
P20 ($\overline{INT5}$, $\overline{STOP}$)	I/O (input)		External interrupt input 5 or STOP mode release signal input
P33 to P30	I/O	4-bit programmable input/output ports(tri-state). Each bit of this port can be individually configured as an input or an output under software control.	
P47 (AIN07) to P40 (AIN00)	I/O (input)	8-bit programmable input/output ports (tri-state). Each bit of this port can be individually configured as an input or an output under software control. When used as analog input, the input mode is configured.	AD converter analog inputs
P53 (AIN13) to P50 (AIN10)	I/O (input)	4-bit programmable input/output ports (tri-state). Each bit of this port can be individually configured as an input or an output under software control. When used as analog input, the input mode is configured.	AD converter analog inputs

Pin Name	I/O	Function	
P67 (V7) to P60 (V0)	I/O (output)	Six 8-bit high breakdown voltage output ports with the latch. When used as a VFT driver output, the latch must be cleared to "0".	VFT driver output
P77 (V15) to P70 (V8)	I/O (output)		
P87 (V23) to P80 (V16)	I/O (output)		
P97 (V31) to P90 (V24)	I/O (output)		
PD7 (V39) to PD0(V31)	I/O (output)		
PE7 (V47) to PE0 (V40)	I/O (output)		
PF4 (V52) to PF0 (V48)	I/O (output)	6-bit high breakdown voltage output ports with the latch. When used as a VFT driver output, the latch must be cleared to "0".	
XIN, XOUT	Input, output	Resonator connecting pins for high-frequency clock. For inputting external clock, XIN is used and XOUT is opened.	
$\overline{\text{RESET}}$	I/O	Reset signal input or watchdog timer output/address-trap-reset output /sistem-clock-reset outputed.	
TEST	Input	Test pin for out-going test.Be tied to low.	
VDD, VSS (Note)	Power Supply	+ 5 V, 0 V (GND)	
VKK		VFT driver power supply	
VAREF, VASS		Analog reference voltae inputs (High, Low)	

Note: All VDDs should be connected externally for keeping the same voltage level.

Operational Description

1. CPU Core FUnctions

The CPU core consists of a CPU, a system clock controller, an interrupt controller, and a watchdog timer. This section provides a description of the CPU core, the program memory (ROM), the data memory (RAM), and the reset circuit.

1.1 Memory Address Map

TLCS-870/X Series, the memory is organized 4 address spaces (ROM, RAM, SFR, and DBR). Figure 1-1 shows the memory address maps of the 88CP77/S77/U77. It uses a memory mapped I/O system, and all I/O registers are mapped in the SFR/DBR address spaces. There are 16 banks of general-purpose registers.

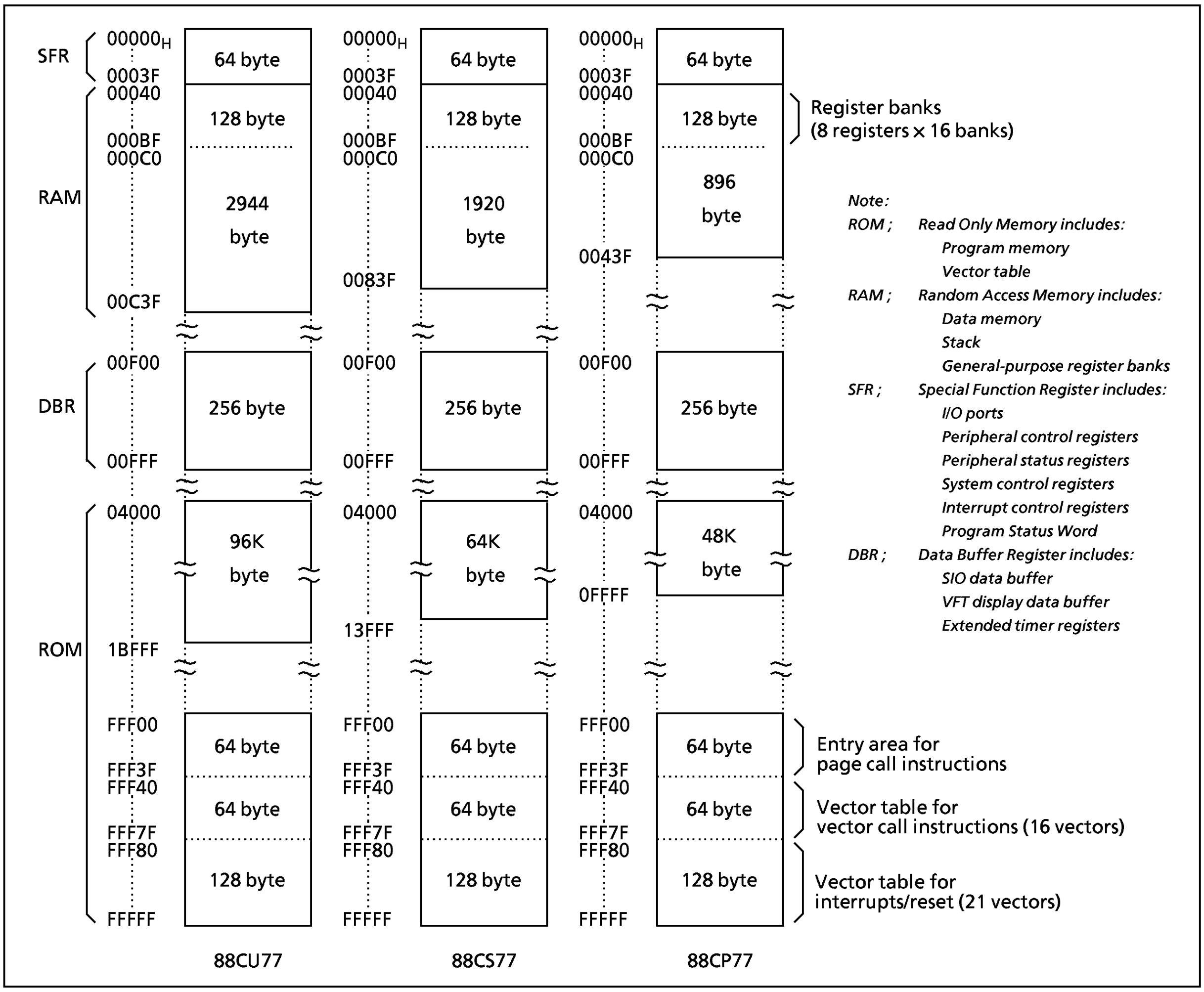

SFR
RAM
DBR
ROM
00000H
0003F
00040
000BF
000C0
00C3F
0083F
0043F
00F00
00FFF
04000
1BFFF
13FFF
0FFFF
FFF00
FFF3F
FFF40
FFF7F
FFF80
FFFFF
64 byte
128 byte
2944 byte
1920 byte
896 byte
256 byte
96K byte
64K byte
48K byte
64 byte
64 byte
128 byte
88CU77
88CS77
88CP77
Register banks (8 registers × 16 banks)
Entry area for page call instructions
Vector table for vector call instructions (16 vectors)
Vector table for interrupts/reset (21 vectors)
Note:
ROM ; Read Only Memory includes:
Program memory
Vector table
RAM ; Random Access Memory includes:
Data memory
Stack
General-purpose register banks
SFR ; Special Function Register includes:
I/O ports
Peripheral control registers
Peripheral status registers
System control registers
Interrupt control registers
Program Status Word
DBR ; Data Buffer Register includes:
SIO data buffer
VFT display data buffer
Extended timer registers

Figure 1-1. Memory Address Maps

1.2 Program Memory (ROM)

The 88CP77 has a 48 Kbytes (addresses 04000_H to $0FFFF_H$) and 256 bytes (addresses $FFF00_H$ to $FFFFF_H$), the 88CS77 has a 64 Kbytes (address 04000_H to $13FFF_H$) and 256 bytes (addresses $FFF00_H$ to $FFFFF_H$), the 88CU77 has a 96 Kbytes (address 04000_H to $1BFFF_H$) and 256 bytes (addresses $FFF00_H$ to $FFFFF_H$) of program memory (mask programmed ROM). Figure1-1 shown in Memory address maps.

Addresses $FFF00_H$ to $FFFFF_H$ in the program memory can also be used for special purposes.

1.3 Data Memory (RAM)

The 88CP77 has 1 Kbytes (address 00040_H to $0043F_H$), the 88CS77 has 2 Kbytes (address 00040_H to $0083F_H$), the 88CU77 has a 3 Kbytes (address 00040_H to $00C3F_H$) of data memory (static RAM). The first 128 bytes (00040_H to $000BF_H$) of the internal RAM are also used as general-purpose register banks.

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

Example1: Clears RAM to "00_H" except the bank 0 (88CP77).

```
          LD     HL, 00048H     ; Sets start address to HL register pair
          LD     A, H           ; Sets initial data (00H) to A register
          LD     BC, 03F7H      ; Sets number of byte to BC register pair
SRAMCLR:  LD     (HL+), A
          DEC    BC
          JRS    F, SRAMCLR
```

Example2: Clears RAM to "00_H" except the bank 0 (88CS77).

```
          LD     HL, 0048H      ; Sets start address to HL register pair
          LD     A, H           ; Sets initial data (00H) to A register
          LD     BC, 07F7H      ; Sets number of byte to BC register pair
SRAMCLR:  LD     (HL+), A
          DEC    BC
          JRS    F, SRAMCLR
```

Note: The general-purpose registers are mapped in the RAM; therefore, do not clear RAM at the current bank addresses.

1.4 System Clock Controller

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

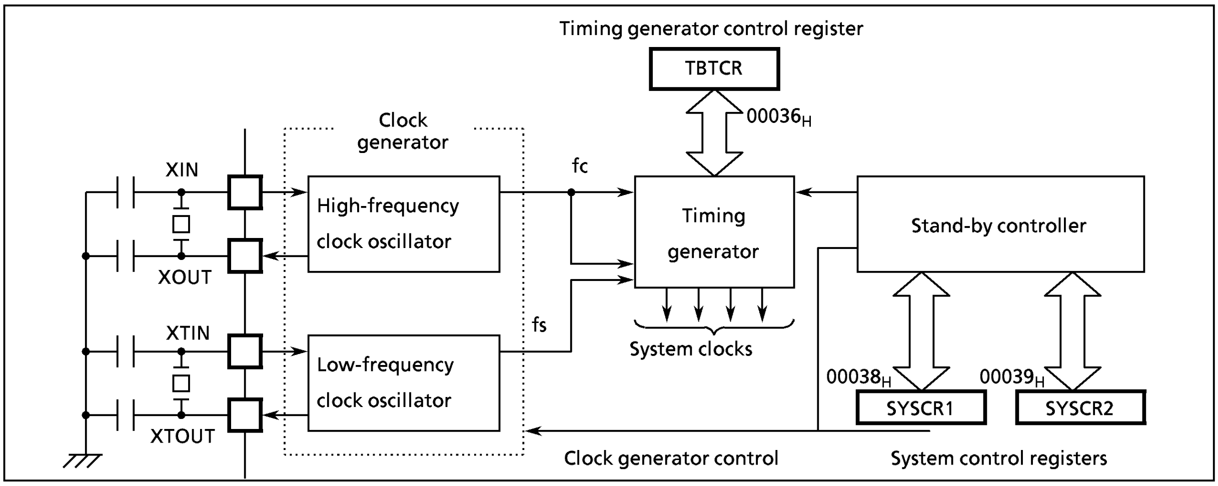


Figure 1-2. System Clock Controller

1.4.1 Clock generator

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

The high-frequency (fc) and low-frequency (fs) clocks can easily be obtained by connecting a resonator between the XIN / XOUT and XTIN / XTOUT pins, respectively. Clock input from an external oscillator is also possible.

In this case, external clock is applied to XIN / XTIN pin with XOUT / XTOUT pin not connected. The 88CP77/S77/U77 are not provided an RC oscillation.

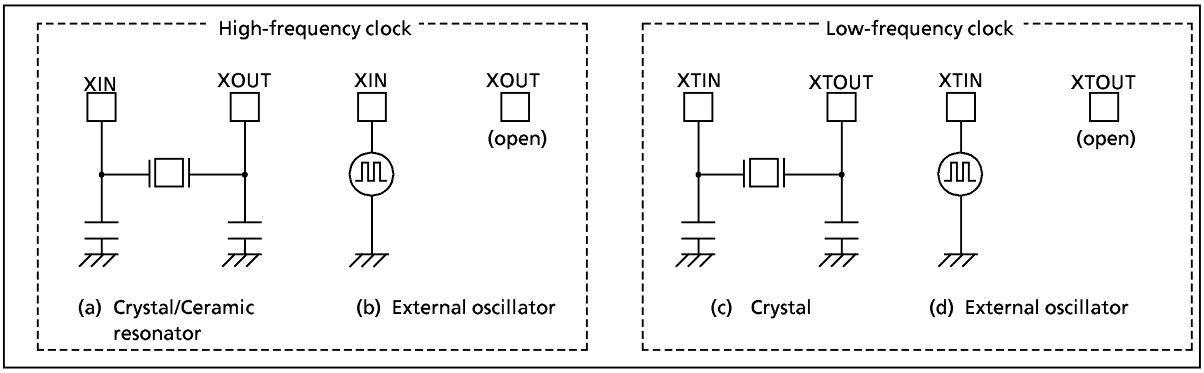


Figure 1-3. Examples of Resonator Connection

Note: Accurate Adjustment of the Oscillation Frequency

Although no hardware to externally and directly monitor the basic clock pulse is not provided, the oscillation frequency can be adjusted by making the program to output fixed frequency pulse to the port with disabling all interrupts and watchdog timers, and monitoring this pulse. With a system requiring adjustment of the oscillation frequency, the adjusting program must be created beforehand.

1.4.2 Timing generator

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

① Generation of main system clock (fm)
② Generation of divider output ($\overline{\text{DVO}}$) pulses
③ Generation of source clocks for time base timer
④ Generation of source clocks for watchdog timer
⑤ Generation of internal source clocks for timer / counters
⑥ Generation of internal clocks for serial interfaces SIO and HSO
⑦ Generation of source clocks for VFT driver circuit
⑧ Generation of warm-up clocks for releasing STOP mode
⑨ Generation of a clock for releasing reset output

(1) Configuration of timing generator

The timing generator consists of a 3-stage prescaler, a 21-stage divider, a main system clock generator, and machine cycle counters.

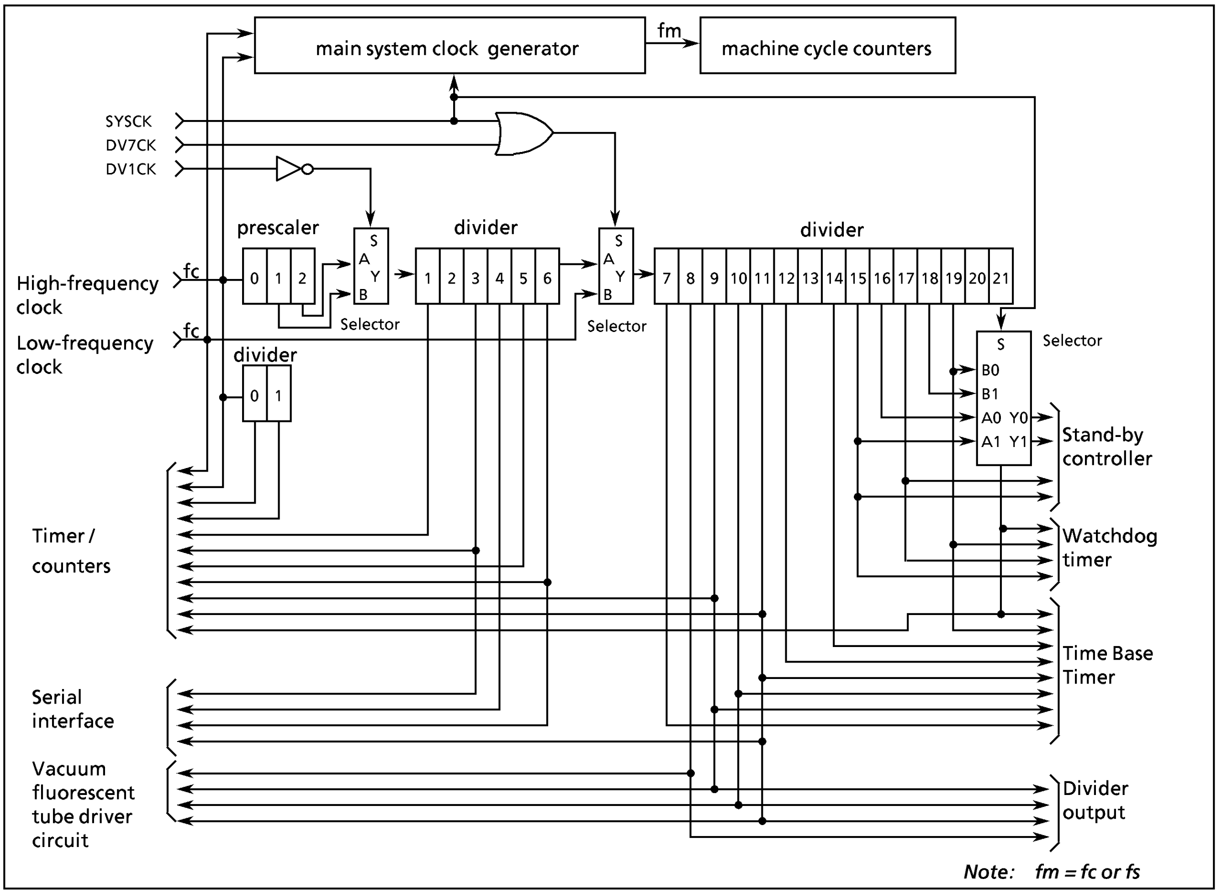
The clock fc/4 or fc/8, that is output from the 2nd stage or the 3rd stage of the prescaler, can be selected as the clock to input to the 1st stage of the divider by DV1CK (bit 5 in CGCR).

The DV1CK should be set the peripheral circuit prior to starting the peripheral circuits. Do not change the set value after setting.
An input clock to the 7th stage of the divider depends on the operating mode, DV1CK (bit 5 in CGCR), and DV7CK(bit 4 in TBTCR), that is shown in table 1-1. As reset and STOP mode started/canceled, The prescaler and the divider are cleared to "0".

Table1-1. Input Clock to 7th Stage of The Divider

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

Note 1: Do not set DV7CK to "1" in the single clock mode.
Note 2: In SLOW and SLEEP mode, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.



Note: fm = fc or fs

Figure 1-4. Configuration of Timing Generator

DVCR
(00030$_H$)

7	6	5	4	3	2	1	0
"0"	"0"	DV1CK					

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

DV1CK	Selection of input clock to the 1st stage of the divider	0: fc/4 1: fc/8	R/W

*Note 1: fc: High-frequency clock [Hz], *: don't care*
Note 2: Bit 4 to 0 in CGCR is always read in as "1" when a read instruction is executed.

Figure 1-5. Clock Gear Control Register

TBTCR
(00036$_H$)

7	6	5	4	3	2	1	0
(DVOEN)	(DVOCK)		DV7CK	(TBTEN)	(TBTCK)		

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

DV7CK	Selection of input clock to the 7 th stage of the divider	0: fc/2^8 or fc/2^9 [Hz] 1: fs	R/W

*Note 1: fc: High-frequency clock [Hz], *: don't care*
Note 2: Do not set DV7CK to "1" in the single clock mode.
Note 3: Do not set DV7CK to "1" before low-frequency clock is stable in the dual-clock mode.

Figure 1-6. Timing Generator Control Register

(2) Machine cycle

Instruction execution and peripheral hardware operation are synchronized with the main system clock. The minimum instruction execution unit is called an "machine cycle". There are a total of 15 different types of instructions for the TLCS-870/X Series: ranging from 1-cycle instructions which require one machine cycle for execution to 15-cycle instructions which require 15 machine cycles for execution.

A machine cycle consists of 4 states (S0 to S3), and each state consists of one main system clock.

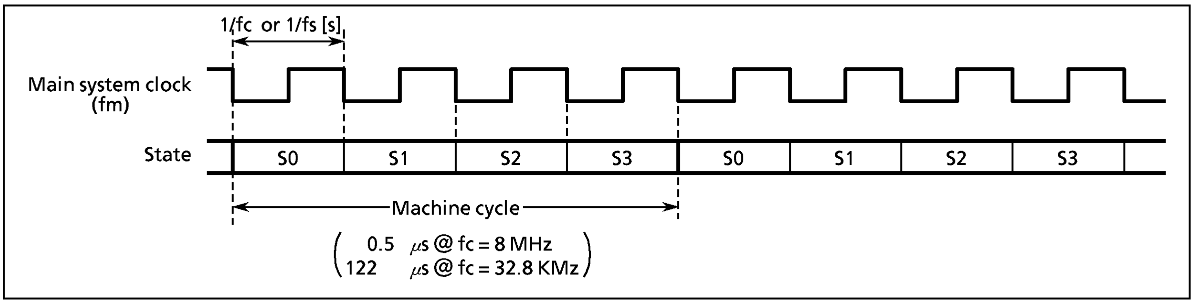


Figure 1-7. Machine Cycle

1.4.3 Stand-by controller

The stand-by controller starts and stops the oscillation circuits for the high-frequency and low-frequency clocks, and switches the main system clock. There are two operating modes: single-clock and dual-clock. These modes are controlled by the system control registers (SYSCR1and SYSCR2).

Figure 1-8 shows the operating mode transition diagram and Figure 1-9 shows the system control registers.

(1) **Single-clock mode**

Only the oscillation circuit for the high-frequency clock is used, and P21 (XTIN) and P22 (XTOUT) pins are used as input/output ports. In the single-clock mode, the machine cycle time is 4/fc [s] (0.5 μs @ fc = 8 MHz).

① NORMAL1 mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock. The 88CP77/S77/U77 are placed in this mode after reset.

② IDLE1 mode

In this mode, the internal oscillation circuit remains active. The CPU and the watchdog timer are halted; however on-chip peripherals remain active (operate using the high-frequency clock). IDLE1 mode is started by the system control register 2 (SYSCR2), and IDLE1 mode is released to NORMAL1 mode by an interrupt request from the on-chip peripherals or external interrupt inputs. When the IMF (interrupt master enable flag) is "1" (interrupt enable), the execution will resume with the acceptance of the interrupt, and the operation will return to normal after the interrupt service is completed. When the IMF is "0" (interrupt disable), the execution will resume with the instruction which follows the IDLE1 mode start instruction.

③ STOP1 mode

In this mode, the internal oscillation circuit is turned off, causing all system operations to be halted. The internal status immediately prior to the halt is held with a lowest power consumption during STOP1 mode.

STOP1 mode is started by the system control register 1 (SYSCR1), and STOP1 mode is released by an inputting (either level-sensitive or edge-sensitive can be programmably selected) to the $\overline{\text{STOP}}$ pin. After the warming-up period is completed, the execution resumes with the instruction which follows the STOP1 mode start instruction.

(2) **Dual-clock mode**

Both the high-frequency and low-frequency oscillation circuits are used in this mode. P21 (XTIN) and P22 (XTOUT) pins cannot be used as input / output ports. The main system clock is obtained from the high-frequency clock in NORMAL2 and IDLE2 modes, and is obtained from the low-frequency clock in SLOW and SLEEP modes. The machine cycle time is 4/fc [s] in the NORMAL2 and IDLE2 modes, and 4/fs [s] (122 μs at fs = 32.768 kHz) in the SLOW and SLEEP modes.

The TLCS-870/X is placed in the signal-clock mode during reset. To use the dual-clock mode, the low-frequency oscillator should be turned on by executing [SET (SYSCR2), XTEN] instruction.

① NORMAL2 mode

In this mode, the CPU core operates using the high-frequency clock. On-chip peripherals operate using the high-frequency clock and/or low-frequency clock.

② SLOW mode

This mode can be used to reduce power-consumption by turning off oscillation of the high-frequency clock. The CPU core and on-chip peripherals operate using the low-frequency clock. Switching back and forth between NORMAL2 and SLOW modes are performed by the system control register 2 (SYSCR2).

③ IDLE2 mode

In this mode, the internal oscillation circuit remain active. The CPU and the watchdog timer are halted; however, on-chip peripherals remain active (operate using the high-frequency clock and / or the low-frequency clock). Starting and releasing of IDLE2 mode are the same as for IDLE1 mode, except that operation returns to NORMAL2 mode.

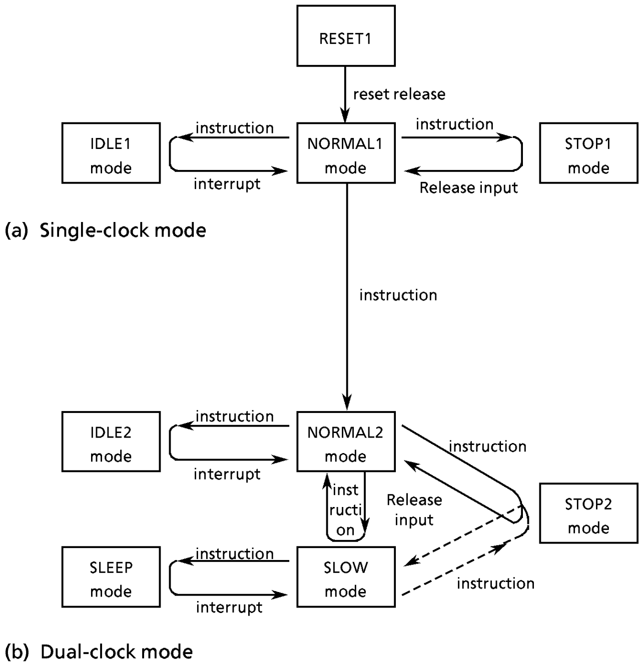
④ SLEEP mode

In this mode, the internal oscillation circuit of the low-frequency clock remains active. The CPU, the watchdog timer, and the internal oscillation circuit of the high-frequency clock are halted; however, on-chip peripherals remain active (operate using the low-frequency clock). Starting and releasing of SLEEP mode are the same as for IDLE1 mode, except that operation returns to SLOW mode.

⑤ STOP2 mode

As in STOP1 mode, all system operations are halted in this mode.

As in NORMAL2 mode at the start, the operating mode returns to NORMAL2 mode, and as in SLOW mode at the start, it returns to SLOW mode after release.



(a) Single-clock mode

(b) Dual-clock mode

Note: NORMAL1 and NORMAL2 modes are generically called NORMAL; STOP1 and STOP2 are called STOP; and IDLE1, IDLE2 and SLEEP are called IDLE.

Operating mode		Frequency		CPU core	On-chip Peripherals	Machine cycle time
		High-frequency	Low-frequency			
Single-Clock	RESET1	turning on oscillation	turning off oscillation	reset	reset	4/fc [s]
	NORMAL1			operate	operate	
	IDLE1			halt		
	STOP1	turning off oscillation			halt	—
Dual-Clock	NORMAL2	turning on oscillation	turning on oscillation	High-frequency	operate (High and/or Low)	4/fc [s]
	IDLE2			halt		
	SLOW	turning off oscillation		Low-frequency	Low-frequency	4/fs [s]
	SLEEP			halt		
	STOP2		turning off oscillation		halt	—

Figure 1-8. Operating Mode Transition Diagram

System Control Register 1

SYSCR1
(00038$_H$)

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

(Initial value: 0001 00**)

<table>
<tr><td>STOP</td><td>STOP mode start</td><td colspan="4">0: CPU core and peripherals remain active
1: CPU core and peripherals are halted (start STOP mode)</td><td rowspan="4">R/W</td></tr>
<tr><td>RELM</td><td>Release method for STOP mode</td><td colspan="4">0: Edge-sensitive release
1: Level-sensitive release</td></tr>
<tr><td>RETM</td><td>Operating mode after STOP mode</td><td colspan="4">0: Return to NORMAL mode
1: Return to SLOW mode</td></tr>
<tr><td rowspan="3">WUT</td><td rowspan="3">Warming-up time at releasing STOP mode</td><td rowspan="2"></td><td colspan="2">Return to NORMAL mode</td><td rowspan="2">Return to SLOW mode</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>00
01
10
11</td><td>3 x 2^{16} / fc
2^{16} / fc
3 x 2^{14} / fc
2^{14} / fc</td><td>3 x 2^{17} / fc
2^{17} / fc
3 x 2^{15} / fc
2^{15} / fc</td><td>3 x 2^{13} / fs
2^{13} / fs</td></tr>
</table>

Note 1: Always set RETM to "0" when transiting from NORMAL mode to STOP mode. Always set RETM to "1" when transiting from SLOW mode to STOP mode.

Note 2: When STOP mode is released with RESET pin input, a return is made to NORMAL mode regardless of the RETM contents.

Note 3: fc ; *High-frequency clock* [Hz]
fs ; *Low-frequency clock* [Hz]
* ; *Don't care*

Note 4: Bits 1 and 0 in SYSCR1 are read in as undefined data when a read instruction is executed.

Note 5: Always set bit 4 in SYSCR1 to "1" when STOP mode is started.

System Control Register 2

SYSCR2
(00039$_H$)

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

(Initial value: 1000 ****)

XEN	High-frequency oscillator control	0: Turn off oscillation 1: Turn on oscillation	R/W
XTEN	Low-frequency oscillator control	0: Turn off oscillation 1: Turn on oscillation	
SYSCK	Main system clock select (write)/main system clock monitor (read)	0: High-frequency clock 1: Low-frequency clock	
IDLE	IDLE mode start	0: CPU and watchdog timer remain active 1: CPU and watchdog timer are stopped (start IDLE1 mode)	

Note 1: XEN and SYSCK are automatically overwritten in accordance with the contents of RETM (bit 5 in SYSCR1) when STOP mode is released.

RETM	operating mode after STOP mode	XEN	SYSCK
0	NORMAL 1/2 mode	1	0
1	SLOW mode	0	1

Note 2: Do not clear XEN to "0" when SYSCK = 0, and do not clear XTEN to "0" when SYSCK = 1.

Note 3: A reset is applied (RESET pin output goes low) if both XEN and XTEN are cleared to "0".

*Note 4: * ; don't care*

Note 5: Bits 3 to 0 in SYSCR2 are always read in as "1" when a read instruction is executed.

Figure 1-9. System Control Registers1, 2

1.4.4 Operating mode control

(1) **STOP** mode (STOP1, STOP2)

STOP mode is controlled by the system control register 1 (SYSCR1) and the $\overline{\text{STOP}}$ pin input. The $\overline{\text{STOP}}$ pin is also used both as a port P20 and an $\overline{\text{INT5}}$ (external interrupt input 5) pin. STOP mode is started by setting STOP (bit 7 in SYSCR1) to "1". During STOP mode, the following status is maintained.

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

STOP mode includes a level-sensitive release mode and an edge-sensitive release mode, either of which can be selected with the RELM (bit 6 in SYSCR1).

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

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

① Testing a port P20.
② Using an external interrupt input $\overline{\text{INT5}}$ ($\overline{\text{INT5}}$ is a falling edge-sensitive input).

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

```
         LD (SYSCR1), 01010000B ;  Sets up the level-sensitive release mode
SSTOPH:  TEST  (P2) . 0         ;  Wait until the STOP pin input goes low level
         JRS   F, SSTOPH
         SET   (SYSCR1) . 7     ;  Starts STOP mode
```

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

```
PINT5:   TEST  (P2) . 0         ;  To reject noise, STOP mode does not start if
         JRS   F, SINT5            port P20 is at high
         LD (SYSCR1), 01010000B ;  Sets up the level-sensitive release mode.
         SET   (SYSCR1) . 7     ;  Starts STOP mode
SINT5:   RETI
```

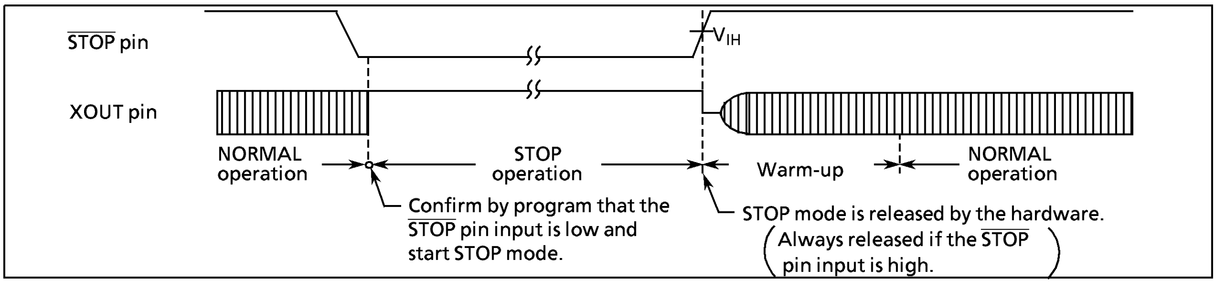

STOP pin
XOUT pin
V_IH
NORMAL operation
STOP operation
Warm-up
NORMAL operation
Confirm by program that the STOP pin input is low and start STOP mode.
STOP mode is released by the hardware.
(Always released if the STOP pin input is high.)

Figure 1-10. Level-sensitive release Mode

Note 1: Even if the STOP pin input is low after warming up start, the STOP mode is not restarted.
Note 2: In this case of changing to the level-sensitive mode from the edge-sensitive mode, the release mode is not switched until a rising edge of the STOP pin input is detected.

b. Edge-sensitive release mode (RELM = "0")

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

Example: Starting STOP mode from NORMAL mode

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

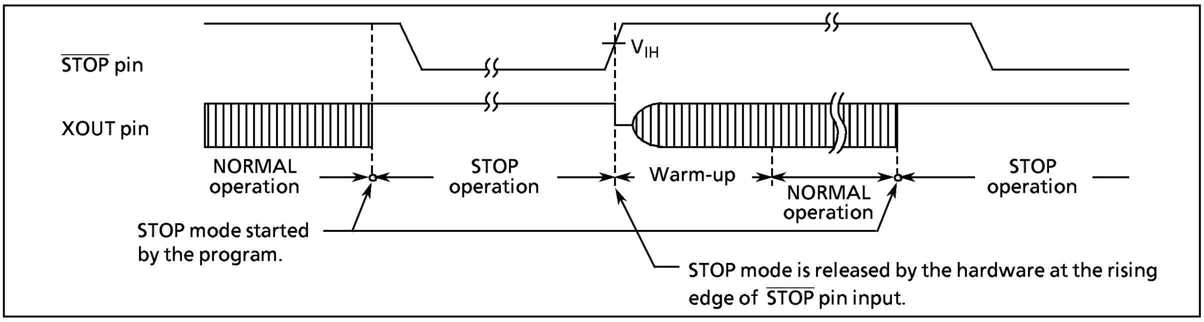

STOP pin
XOUT pin
V_IH
NORMAL operation
STOP operation
Warm-up
NORMAL operation
STOP operation
STOP mode started by the program.
STOP mode is released by the hardware at the rising edge of STOP pin input.

Figure 1-11. Edge-sensitive release Mode

STOP mode is released by the following sequence.

① In the dual-clock mode, when returning to NORMAL2, both the high-frequency and low-frequency clock oscillators are turned on ; when returning to SLOW mode, only the low-frequency clock oscillator is turned on. In the signal-clock mode, only the high-frequency clock oscillator is turned on.

② A warming-up period is inserted to allow oscillation time to stabilize. During warm-up, all internal operations remain halted. Four different warming-up times can be selected with the WUT (bits 2 and 3 in SYSCR1) in accordance with the resonator characteristics.

③ When the warming-up time has elapsed, normal operation resumes with the instruction following the STOP mode start instruction (e.g. [SET (SYSCR1). 7]). The start is made after the prescaler and the divider of the timing generator are cleared to "0".

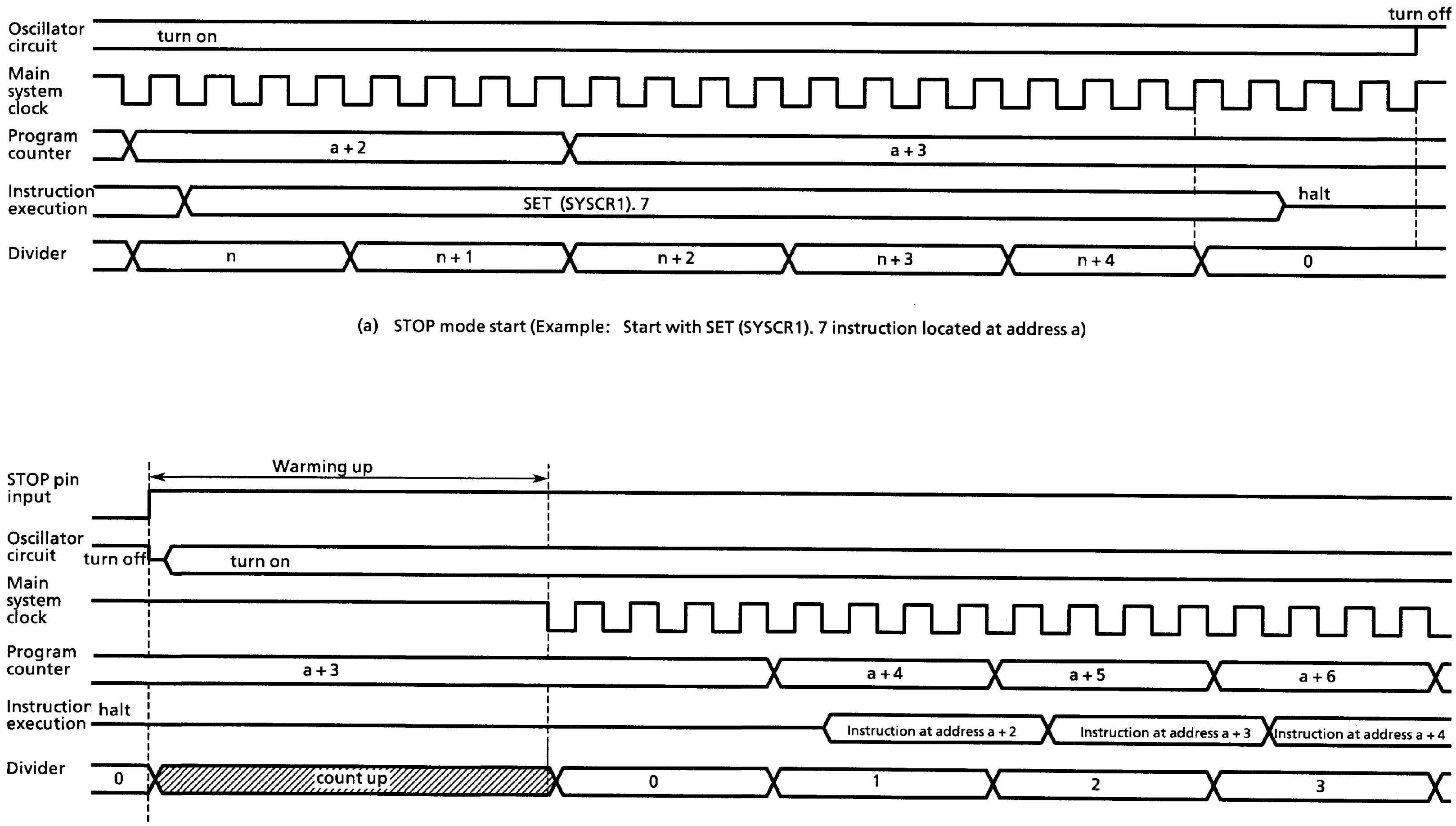
Table 1-2. Warming-up Time Example (at fc = 8 MHz, fs = 32.768 kHz)

WUT	Warming-up Time [ms]		
	Return to NORMAL mode		Return to SLOW mode
	DV1CK = 0	DV1CK = 0	
00	24.576	49.152	750
01	8.192	16.384	250
10	6.144	12.288	—
11	2.048	4.096	—

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

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

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



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

(b) STOP mode release

Figure 1-12. STOP Mode Start / Release

(2) **IDLE** mode (IDLE1, IDLE2, SLEEP)

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

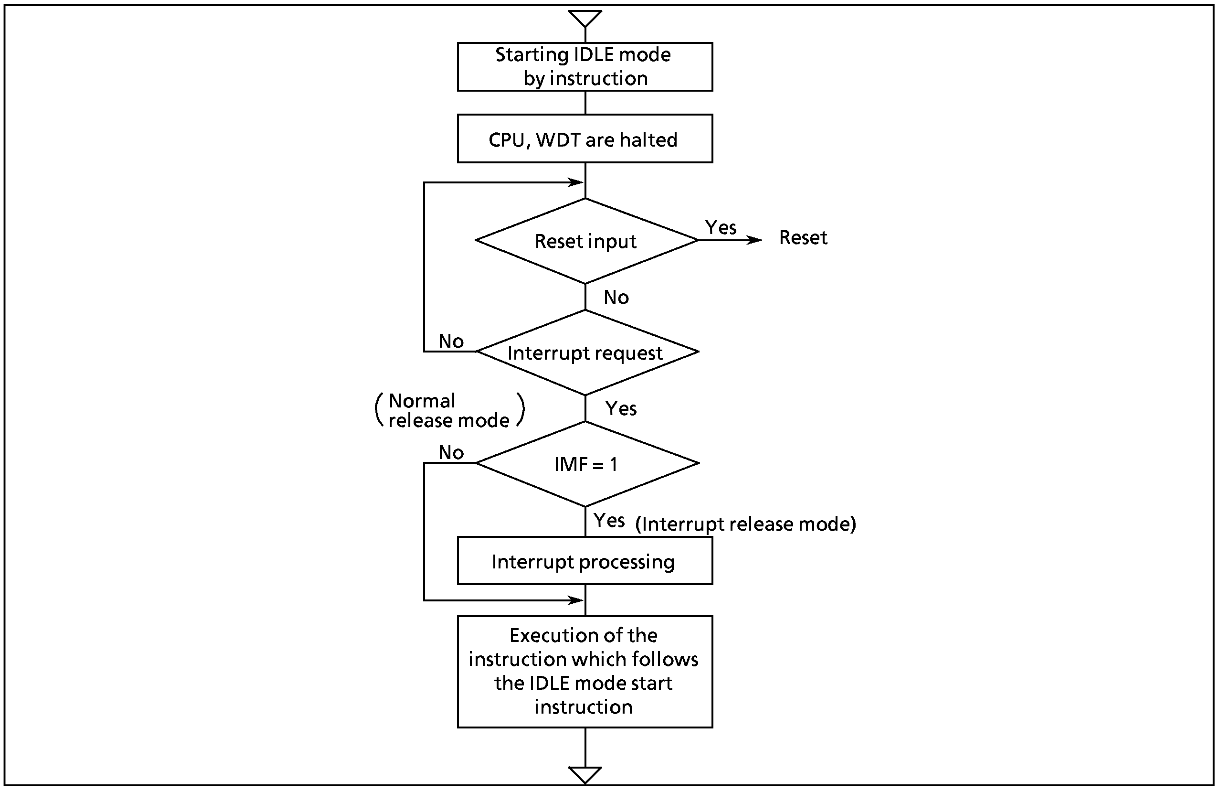
① Operation of the CPU and watchdog timer (WDT) is halted. On-chip peripherals continue to operate.

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

③ The program counter holds the address of the instruction but one to the instruction which starts IDLE mode.

Example: Starting IDLE mode.

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



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

Figure 1-13. IDLE Mode

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

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

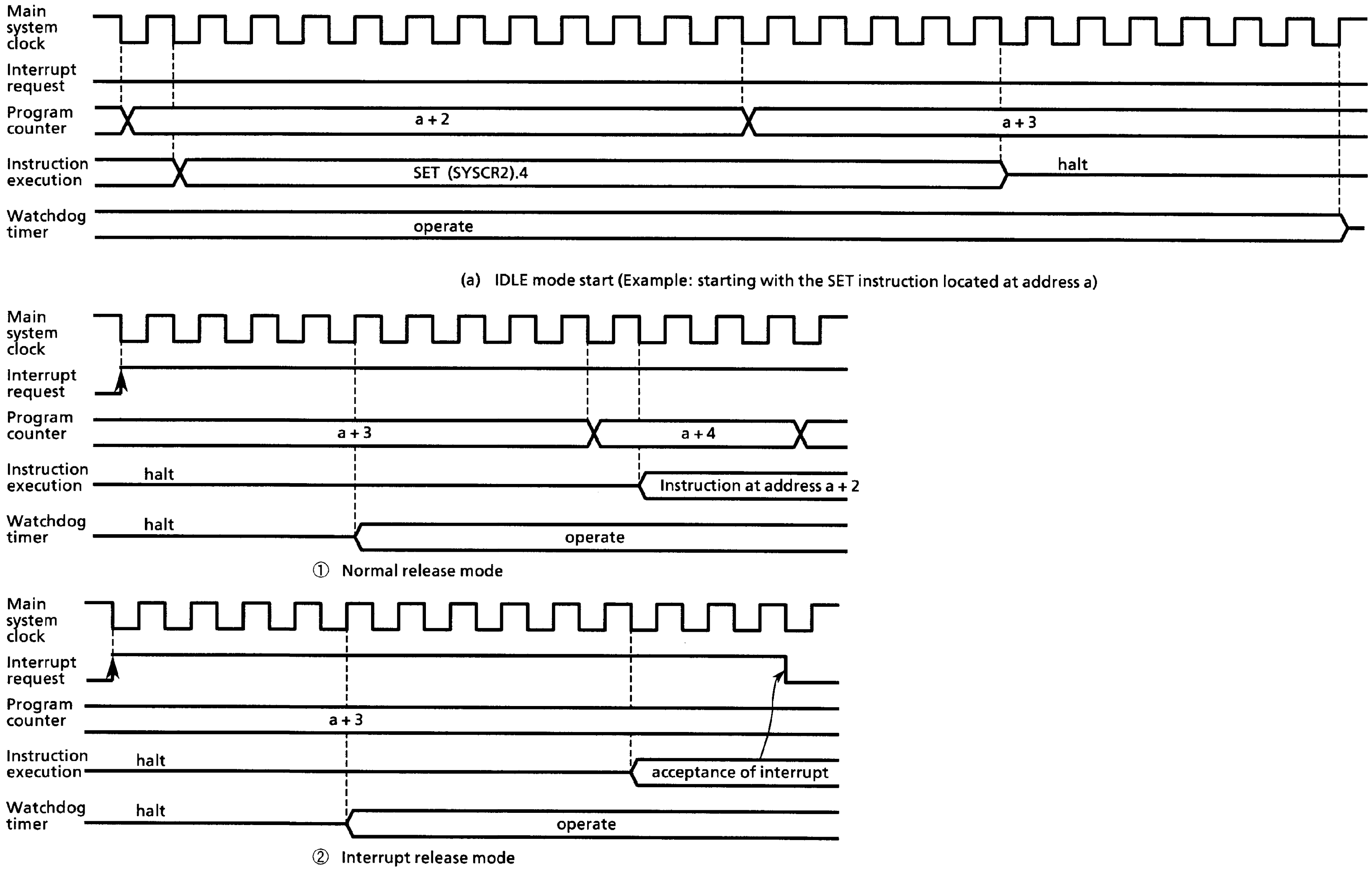
IDLE mode is released by any interrupt source enabled by the individual interrupt enable flag (EF) or an external interrupt 0 ($\overline{\text{INT0}}$ pin) request. Execution resumes with the instruction following the IDLE mode start instruction (e.g. [SET (SYSCR2), 4]. The interrupt latches (IL) of the interrupt source used for releasing must be cleared to "0" by load instructions.

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

IDLE mode is released and interrupt processing is started by any interrupt source enabled with the individual interrupt enable flag (EF) or an external interrupt 0 ($\overline{\text{INT0}}$ pin) request. After the interrupt is processed, the execution resumes from the instruction following the instruction which starts IDLE mode.

IDLE mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin, which immediately performs the reset operation. After reset, the 88CP77/S77/U77 is placed in NORMAL 1 mode.

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



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

① Normal release mode

② Interrupt release mode

(b) IDLE mode release

Figure 1-14. IDLE Mode Start / Release

(3) **SLOW** mode

SLOW mode is controlled by the system control register 2 (SYSCR2) and the timer/counter 2 (TC2).

a. Switching from NORMAL2 mode to SLOW mode

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

> *Note: The high frequency clock can be continued oscillation in order to return to* ***NORMAL2*** *mode from* ***SLOW*** *mode quickly.*
> *Always turn off oscillation of high frequency clock when switching from* ***SLOW*** *mode to* ***STOP*** *mode.*

When the low-frequency clock oscillation is unstable, wait until oscillation stabilizes before performing the above operations. The timer/counter 2 (TC2) can conveniently be used to confirm that low-frequency clock oscillation has stabilized.

Example1: Switching from NORMAL2 mode to SLOW mode.

```
                SET      (SYSCR2) . 5       ;  SYSCK←1
                                               (switches the main system clock to the low-
                                               frequency clock)
                CLR      (SYSCR2) . 7       ;  XEN←0
                                               (turns off high-frequency oscillation)
```

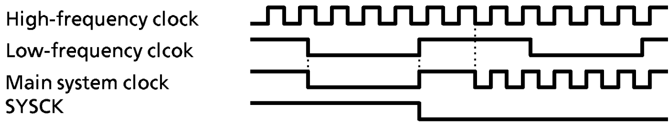
Example2: Switching to the SLOW mode after low-frequency clock oscillation has stabilized.

```
                LD       (TC2CR), 14H       ;  Sets TC2 mode
                                               (timer mode, source clock: fs)
                LDW      (TREG2), 8000H     ;  Sets warming-up time
                                               (according to Xtal characteristics)
                SET      (EIRH). EF8        ;  Enables INTTC2
                LD       (TC2CR), 34H       ;  Starts TC2
                 ⋮
      PINTTC2:  LD       (TC2CR), 10H       ;  Stops TC2
                SET      (SYSCR2) . 5       ;  SYSCK←1
                                               (switches the main system clock to the low-
                                               frequency clock)
                CLR      (SYSCR2) . 7       ;  XEN←0
                                               (turns off high-frequency oscillation)
                RETI
                 ⋮
      VINTTC2:  DL       PINTTC2            ;  INTTC2 vector table
```

b. Switching from SLOW mode to NORMAL2 mode

First, set XEN (bit 7 in SYSCR2) to turn on the high-frequency oscillation. When time for stabilization (warm-up) has been taken by the timer / counter 2 (TC2), clear SYSCK (bit 5 in SYSCR2) to switch the main system clock to the high-frequency clock.

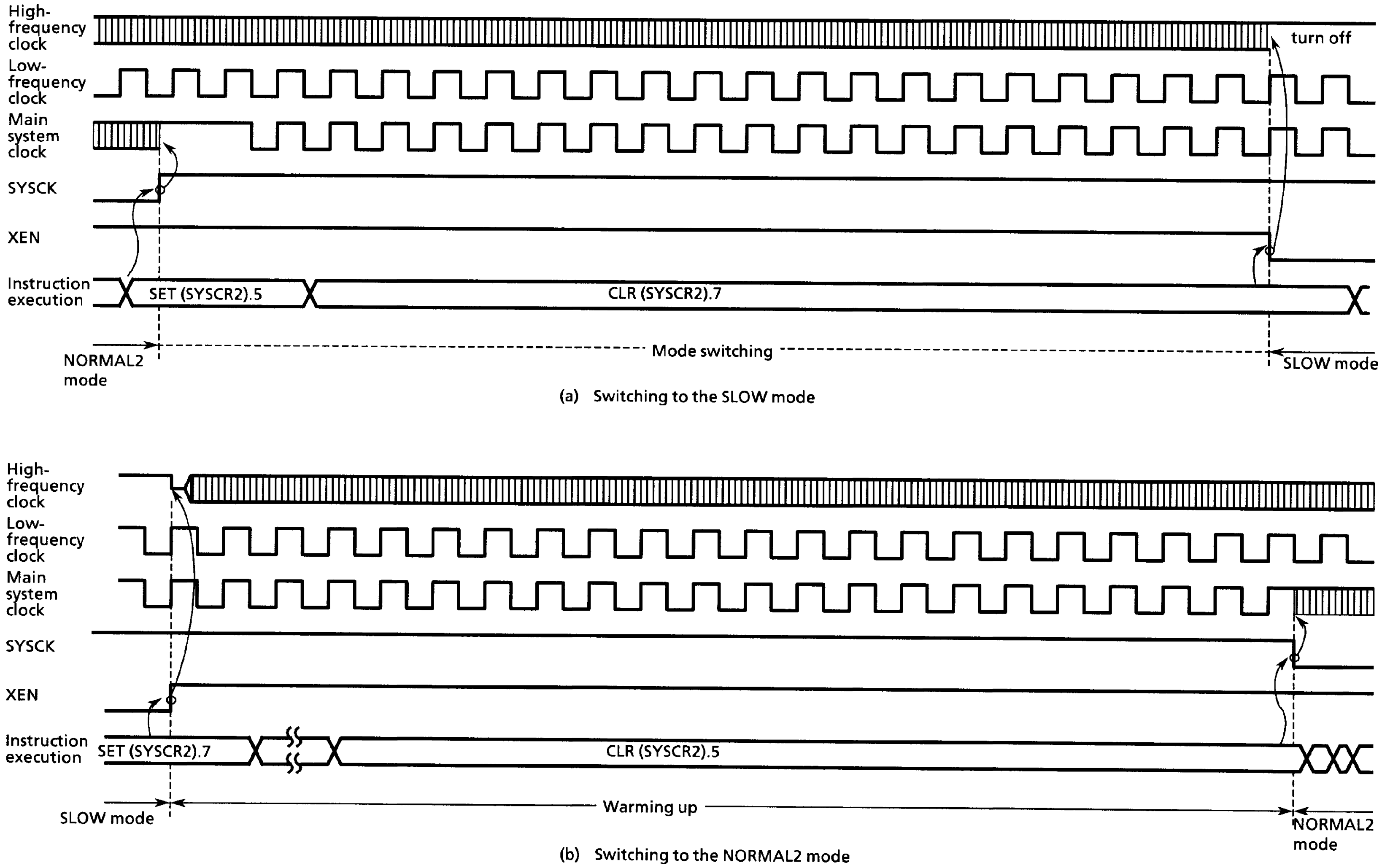
Note 1: After SYSCK is cleared to "0", executing the instructions is continued by the low-frequency clock for the period synchronized with low-frequency and high-frequency clocks.



Note 2: SLOW mode can also be released by inputting low level on the RESET pin, which immediately performs the reset operation. After reset, the 88CP77/S77/U77 are placed in NORMAL1 mode.

Example: Switching from SLOW mode to NORMAL2 mode (fc = 8 MHz, warming-up time is 7.94 ms).

```
          SET     (SYSCR2) . 7       ; XEN←1 (turns on high-frequency oscillation)
          LD      (TC2CR), 10H       ; Sets TC2 mode
                                       (timer mode, source clock: fc)
          LD      (TREG2 + 1), 0F8H; Sets the warming-up time
                                       (according to frequency and Xtal
                                       characteristics)
          SET     (EIRH). EF8        ; Enables INTTC2
          LD      (TC2CR), 30H       ; Starts TC2
           ⋮
PINTTC2:  LD      (TC2CR), 10H       ; Stops TC2
          CLR     (SYSCR2) . 5       ; SYSCK←0
                                       (switches the main system clock to the high-
                                       frequency clock)
          RETI
           ⋮
VINTTC2:  DL      PINTTC2            ; INTTC2 vector table
```



(a) Switching to the SLOW mode

(b) Switching to the NORMAL2 mode

Figure 1-15. Switching between the NORMAL2 and SLOW Modes

1.5 Interrupt Controller

The 88CP77/S77/U77 each have a total of 20 interrupt sources: 6 externals and 14 internals. Nested interrupt control with priorities is also possible. Two of the internal sources are pseudo non-maskable interrupts; the remainder are all maskable interrupts.

Interrupt latches (IL) that hold the interrupt requests are provided for interrupt sources. Each interrupt vector is independent.

The interrupt latch is set to "1" when an interrupt request is generated and requests the CPU to accept the interrupt. The acceptance of maskable interrupts can be selectively enabled and disabled by the program using the interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). When two or more interrupts are generated simultaneously, the interrupt is accepted in the highest priority order as determined by the hardware. Figure 1-16 shows the interrupt controller.

Table 1-3. Interrupt Sources

Interrupt Source		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/ External	(Reset)	Non-Maskable	—	FFFC$_H$	High 0
Internal	INTSW (Software interrupt)	Pseudo non-maskable	—	FFF8$_H$	1
Internal	INTWDT (Watchdog Timer interrupt)		IL_2	FFF4$_H$	2
External	INT0 (External interrupt 0)	IMF = 1, INT0EN = 1	IL_3	FFF0$_H$	3
Internal	INTTC1 (16-bit TC1 interrupt)	IMF · EF_4 = 1	IL_4	FFEC$_H$	4
External	INT1 (External interrupt 1)	IMF · EF_5 = 1	IL_5	FFE8$_H$	5
Internal	INTTBT (Time Base Timer interrupt)	IMF · EF_6 = 1	IL_6	FFE4$_H$	6
External	INT2 (External interrupt 2)	IMF · EF_7 = 1	IL_7	FFE0$_H$	7
Internal	INTTC2 (16-bit TC2 interrupt)	IMF · EF_8 = 1	IL_8	FFDC$_H$	8
Internal	INTSBI (Serial Bus Interface interrupt)	IMF · EF_9 = 1	IL_9	FFD8$_H$	9
Internal	INTTC4 (8-bit TC4 interrupt)	IMF · EF_{10} = 1	IL_{10}	FFD4$_H$	10
External	INT3 (External interrupt 3)	IMF · EF_{11} = 1	IL_{11}	FFD0$_H$	11
Internal	INTET1 (16-bit ETC1 interrupt)	IMF · EF_{12} = 1	IL_{12}	FFCC$_H$	12
Internal	INTSIO0 (Serial interface0 interrupt)	IMF · EF_{13} = 1	IL_{13}	FFC8$_H$	13
Internal	INTSIO1 (Serial interface1 interrupt)	IMF · EF_{14} = 1	IL_{14}	FFC4$_H$	14
External	INT5 (External interrupt 5)	IMF · EF_{15} = 1	IL_{15}	FFC0$_H$	15
Internal	INTIC0 (Input capture0 interrupt)	IMF · EF_{16} = 1	IL_{16}	FFBC	16
Internal	INTIC1 (Input capture1 interrupt)	IMF · EF_{17} = 1	IL_{17}	FFB8	17
Internal	INTOC0 (Output compare0 interrupt)	IMF · EF_{18} = 1	IL_{18}	FFB4	18
Internal	INTOC1 (Output compare1 interrupt)	IMF · EF_{19} = 1	IL_{19}	FFB0	19
External	INT4 (External interrupt4)	IMF · EF_{20} = 1	IL_{20}	FFFAC	Low 20

(1) **Interrupt Latches** (IL $_{20\text{ to }2}$)

Interrupt latches are provided for each source, except for a software interrupt. The latch is set to "1" when an interrupt request is generated, and requests the CPU to accept the interrupt. The latch is cleared to "0" just after the interrupt is accepted. All interrupt latches are initialized to "0" during reset.

Interrupt latches are assigned to addresses 0003C$_H$ and 0003D$_H$ in the SFR. Each latch can be cleared to "0" individually by an instruction; however, the read-modify-write instruction such as bit manipulation or operation instructions cannot be used. Thus, interrupt requests can be canceled and initialized by the program. Note that interrupt latches cannot be set to "1" by any instruction.

The contents of interrupt latches can be read out by an instruction. Therefore, testing interrupt requests by software is possible.

Example 1: Clears interrupt latches

```
LDW     (IL), 1110100000111111B   ;  IL12, IL10 to IL6←0
```

Example 2: Reads interrupt latches

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

Example 3: Tests an interrupt latch

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

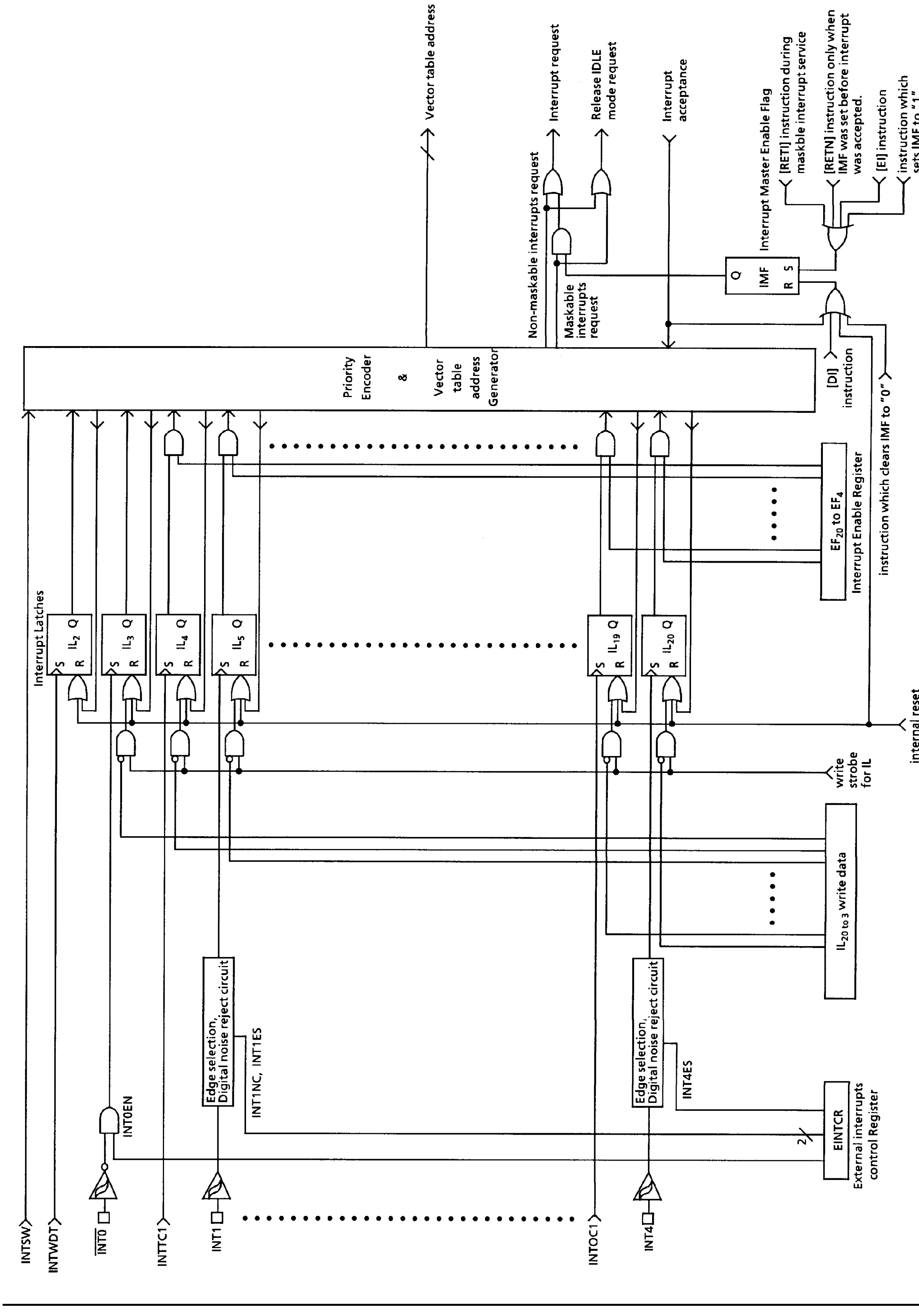

Vector table address
Interrupt request
Release IDLE mode request
Interrupt acceptance
Interrupt Master Enable Flag
[RETI] instruction during maskble interrupt service
[RETN] instruction only when IMF was set before interrupt was accepted.
[EI] instruction
instruction which sets IMF to "1"
Non-maskable interrupts request
Maskable interrupts request
Q
IMF
R S
Priority Encoder
&
Vector table address Generator
[DI] instruction
instruction which clears IMF to "0"
Interrupt Latches
S IL_2 Q
R
S IL_3 Q
R
S IL_4 Q
R
S IL_5 Q
R
S IL_{19} Q
R
S IL_{20} Q
R
EF_{20} to EF_4
Interrupt Enable Register
internal reset
write strobe for IL
$IL_{20 \text{ to } 3}$ write data
Edge selection, Digital noise reject circuit
INT1NC, INT1ES
Edge selection, Digital noise reject circuit
INT4ES
INT0EN
2
EINTCR
External interrupts control Register
INTSW
INTWDT
$\overline{\text{INT0}}$
INTTC1
INT1
INTOC1
INT4

Figure 1-16. Interrupt Controller Block Diagram

(2) **Interrupt Enable Register** (EIR)

The interrupt enable register (EIR) enables and disables the acceptance of interrupts, except for the pseudo non-maskable interrupts (software and watchdog timer interrupts). Pseudo non-maskable interrupts are accepted regardless of the contents of the EIR; however, the pseudo non-maskable interrupts cannot be nested more than once at the same time. For example, the watchdog timer interrupt is not accepted during the software interrupt service.

The EIR consists of an interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). This register is assigned to addresses $0003A_H$ and $0003B_H$ in the SFR, and can be read and written by an instruction (including read-modify-write instructions such as bit manipulation instructions).

① **Interrupt Master enable Flag** (IMF)

The interrupt master enable flag (IMF) enables and disables the acceptance of all interrupts, except for pseudo non-maskable interrupts. Clearing this flag to "0" disables the acceptance of all maskable interrupts. Setting to "1" enables the acceptance of interrupts.

When an interrupt is accepted, this flag is cleared to "0" to temporarily disable the acceptance of maskable interrupts. After execution of the interrupt service program, this flag is set to "1" by the maskable interrupt return instruction [RETI] to again enable the acceptance of interrupts. If an interrupt request has already been occurred, interrupt service starts immediately after execution of the [RETI] instruction.

Pseudo non-maskable interrupts are returned by the [RETN] instruction. In this case, the IMF is set to "1" only when pseudo non-maskable interrupt service is started with interrupt acceptance enabled (IMF = 1). Note that the IMF remains "0" when cleared by the interrupt service program.

The IMF is assigned to bit 0 at address $0003A_H$ in the SFR, and can be read and written by an instruction. The IMF is normally set and cleared by the [EI] and [DI] instructions, and the IMF is initialized to "0" during reset.

② **Individual interrupt Enable Flags** (EF_{20} to EF_4)

These flags enable and disable the acceptance of individual maskable interrupts, except for an external interrupt 0. Setting the corresponding bit of an individual interrupt enable flag to "1" enables acceptance of an interrupt, setting the bit to "0" disables acceptance.

Example 1: Sets EF for individual interrupt enable, and sets IMF to "1".

```
        LDW  (EIRL), 1110100010100001B     ;  EF15 to EF13, EF11, EF7, EF5, IMF←1
```

Example 2: Sets an individual interrupt enable flag to "1".

```
        SET  (EIRH).4                      ;  EF12←1
```

	23	22	21	20	19	18	17	16
IL_E ($0002E_H$)				IL_{20}	IL_{19}	IL_{18}	IL_{17}	IL_{16}

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IL (0003C, $0003D_H$)	IL_{15}	IL_{14}	IL_{13}	IL_{12}	IL_{11}	IL_{10}	IL_9	IL_8	IL_7	IL_6	IL_5	IL_4	IL_3	IL_2		
	IL_H ($0003D_H$)								IL_L ($0003C_H$)							

(Initial Value: ***00000 00000000 000000**)

	23	22	21	20	19	18	17	16
EIR_E ($0002C_H$)				EF_{20}	EF_{19}	EF_{18}	EF_{17}	EF_{16}

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIR (0003A, $0003B_H$)	EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_9	EF_8	EF_7	EF_6	EF_5	EF_4				IMF
	EIR_H ($0003B_H$)								EIR_L ($0003A_H$)							

(Initial Value: ***00000 00000000 0000***0)

Note 1: Do not use any read-modify-write instruction such as bit manipulation for clearing IL.
Note 2: Do not set IMF to "1" during non-maskable interrupt service program.
Note 3: Bits1 and 0 in IL_L, Bits 7, 6 and 5 in ILE are read in as undefined data when a read instruction is executed.
*Note 4: * ; don't care.*

Figure 1-17. Interrupt Latch (IL) and Interrupt Enable Register (EIR)

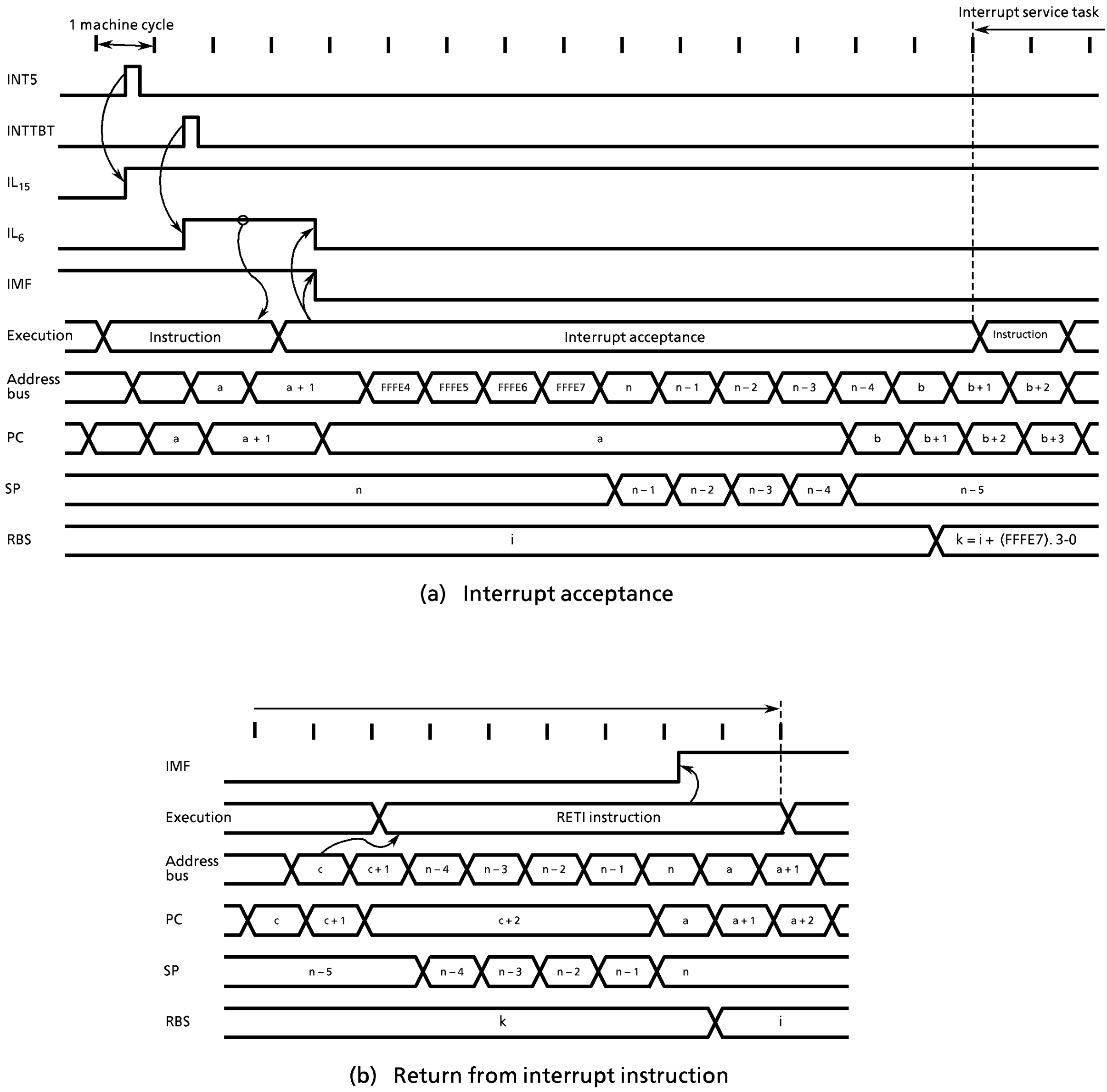
1.5.1 Interrupt sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to "0" by a reset or an instruction. Interrupt acceptance sequence requires 12 machine cycles (6 μs at fc = 8 MHz in the NORMAL mode) after the completion of the current instruction execution. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for pseudo non-maskable interrupts). Figure 1-18 shows the timing chart of interrupt acceptance processing.

(1) Interrupt acceptance

Interrupt acceptance processing is as follows.

① The interrupt master enable flag (IMF) is cleared to "0" to temporarily disable the acceptance of any following maskable interrupts. When a non-maskable interrupt is accepted, the acceptance of any following interrupts is temporarily disabled.
② The interrupt latch (IL) for the interrupt source accepted is cleared to "0".
③ The contents of the program counter (return address) and the program status word (PSW) are saved (pushed) on the stack in sequence of PSWH, PSWL, PCE, PCH, PCL. The stack pointer (SP) is decremented five times.
④ The entry address of the interrupt service program is read from the vector table, and set to the program counter.
⑤ The RBS control code is read from the vector table. The lower 4-bit of this code is added to the RBS.
⑥ The instruction stored at the entry address of the interrupt service program is executed.



(a) Interrupt acceptance

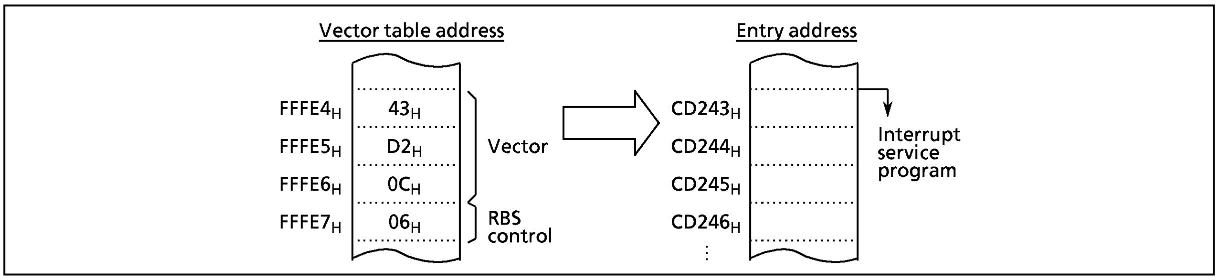
(b) Return from interrupt instruction

Note1: *a ; return address, b ; entry address, c ; address which the RETI instruction is stored*

Note2: *The maximum response time from when an IL is set until an interrupt acceptance processing starts is 62/fc [s] or 62/fs [s] with interrupt enabled..*

Figure 1-18. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

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



A maskable interrupt is not accepted until the IMF is set to "1" even if the maskable interrupt higher than the level of current servicing interrupt is occurred.
When nested interrupt service is necessary, the IMF is set to "1" in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags. However, an acceptance of external interrupt 0 cannot be disabled by the EF; therefore, if disable is necessary, either the external interrupt function of the $\overline{\text{INT0}}$ pin must be disabled with the INT0EN in the external interrupt control register (EINTCR) (the interrupt latch IL3 is not set at INT0EN = 0, therefore, the rising edge of $\overline{\text{INT0}}$ pin input can not be detected.) or an interrupt processing must be avoided by the program.

Example 1: Disables an external interrupt 0 using the INT0EN

```
        LD (EINTCR), 00000000B  ;   INT0EN←0
```

Example 2: Disables the processing of external interrupt 0 under the software control (using bit 0 at address 000F0H as the interrupt processing disable switch)

```
PINT0:  TEST  (000F0H) . 0     ;   Return without interrupt processing if (000F0H)0 = 1
        JRS   T, SINT0
        RETI
SINT0:  Interrupt processing
        RETI
          :
VINT0:  DL    PINT0
```

(2) Saving / Restoring general-purpose registers

During interrupt acceptance processing, the program counter (PC) and the program status word (PSW) are automatically saved on the stack, but not the accumulator and other registers. These registers are saved by the program if necessary. Also, when nesting multiple interrupt services, it is necessary to avoid using the same data memory area for saving registers.

The following method is used to save / restore the general-purpose registers.

① General-purpose register save/restore by automatic register bank changeover

The general-purpose registers can be saved at high-speed by switching to a register bank that is not in use. Normally, the bank 0 is used for the main task and the banks 1 to 15 are assigned to interrupt service tasks. To increase the efficiency of data memory utilization, the same bank is assigned for interrupt sources which are not nested.

The switched bank is automatically restored by executing an interrupt return instruction [RETI] or [RETN]. Therefore, it is not necessary for a program to save the RBS.

Example: Register bank changeover

```
PINTxx:   [ interrupt processing ]
          RETI
           ⋮
VINTxx:   DP        PINTxx
          DB        1                ; RBS ← RBS + 1
```

② General-purpose register save/restore by register bank changeover

The general-purpose registers can be saved at high-speed by switching to a register bank that is not in use. Normally, the bank 0 is used for the main task and the banks 1 to 15 are assigned to interrupt service tasks.

Example: Register bank changeover

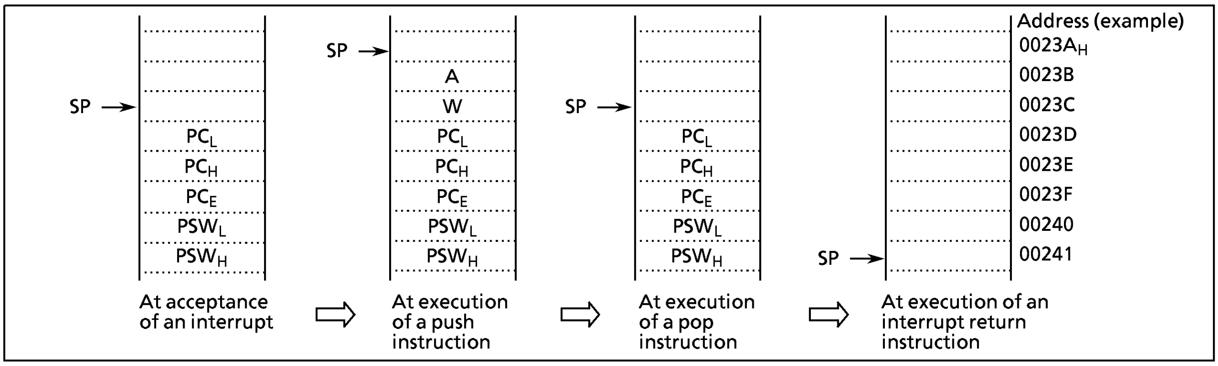
```
PINTxx:   LD        RBS, n
          [ interrupt processing ]
          RETI                     ;  Restores bank and Returns
           ⋮
VINTxx:   DP        PINTxx         ;  Interrupt sevice routine entry address
          DB        0
```

③ General-purpose registers save / restore using push and pop instructions

To save only a specific register, and when the same interrupt source occurs more than once, the general-purpose registers can be saved / restored using the push / pop instructions.

Example: Register save / restore using push and pop instructions

```
PINTxx:   PUSH     WA               ;  Save WA register pair
          interrupt processing
          POP      WA               ;  Restore WA register pair
          RETI                      ;  Return
```

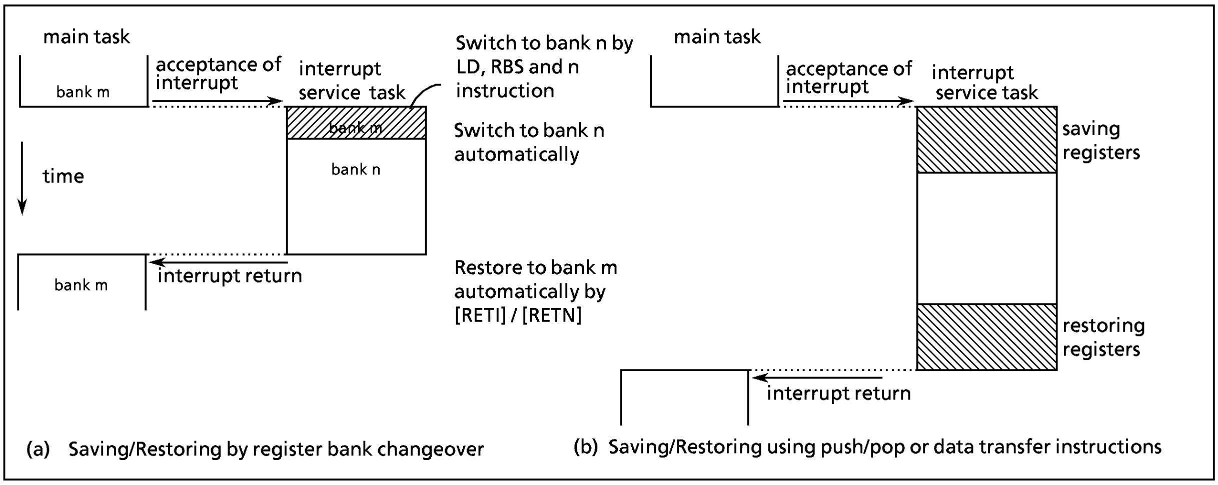

SP
PCL
PCH
PCE
PSWL
PSWH
At acceptance of an interrupt
SP
A
W
PCL
PCH
PCE
PSWL
PSWH
At execution of a push instruction
SP
PCL
PCH
PCE
PSWL
PSWH
At execution of a pop instruction
SP
At execution of an interrupt return instruction
Address (example)
0023AH
0023B
0023C
0023D
0023E
0023F
00240
00241

④ General-purpose registers save / restore using data transfer instructions

Data transfer instruction can be used to save only a specific general-purpose register during processing of single interrupt.

Example: Saving / restoring a register using data transfer instructions

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



(a) Saving/Restoring by register bank changeover

(b) Saving/Restoring using push/pop or data transfer instructions

Figure 1-19. Saving / Restoring General-purpose Registers

(3) Interrupt return

The interrupt return instructions [RETI] / [RETN] perform the following operations.

[RETI] Maskable interrupt return	[RETN] Non-maskable interrupt return
① The contents of the program counter and the program status word are restored from the stack.	① The contents of the program counter and program status word are restored from the stack.
② The stack pointer is incremented 5 times.	② The stack pointer is incremented 5 times.
③ The interrupt master enable flag is set to "1".	③ The interrupt master enable flag is set to "1" only when a non-maskable interrupt is accepted in interrupt enable status. However, the interrupt master enable flag remains at "0" when so clear by an interrupt service program.
④ The interrupt nesting counter is decremented, and the interrupt nesting flag is changed.	④ The interrupt nesting counter is decremented, and the interrupt nesting flag is changed.

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

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

1.5.2 Software interrupt (INTSW)

Executing the [SWI] instruction generates a software interrupt and immediately starts interrupt processing (INTSW is highest prioritized interrupt). However, if processing of a non-maskable interrupt is already underway, executing the SWI instruction will not generate a software interrupt but will result in the same operation as the [NOP] instruction.

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

Note: To use the SWI instruction for software break in the development tool, software interrupt always generates even if the non-maskable interrupt is in progress.

① Address error detection

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

② Debugging

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

1.5.3 External interrupts

The 88CP77/S77/U77 each have six external interrupt inputs ($\overline{\text{INT0}}$, INT1, INT2, INT3, INT4 and $\overline{\text{INT5}}$). Four of these are equipped with digital noise rejection circuits (pulse inputs of less than a certain time are eliminated as noise). Edge selection is also possible with INT1, INT2, INT3 and INT4.

The $\overline{\text{INT0}}$/P11 pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

Edge selection, noise rejection control and $\overline{\text{INT0}}$/P11 pin function selection are performed by the external interrupt control register (EINTCR). When INT0EN = 0, the IL_3 will not be set even if the falling edge of $\overline{\text{INT0}}$ pin input is detected.

Table 1-4. External Interrupts

Source	Pin	Secondary function pin	Enable conditions	Edge			Secondary function pin
				rising	falling	both	
INT0	$\overline{\text{INT0}}$	P11 / SI1	IMF = 1 INT0EN = 1	–	○	–	– (hysteresis input)
INT1	INT1	P04 / OC0	IMF · EF_5 = 1 INT1W = 0	INT1ES = 0	INT1ES = 1	–	For falling or rising edge, pulses of less than 7/fc [s] are eliminated as noise. Pulses of equal to or more than 24/fc [s] are regarded as signals.
			IMF · EF5 = 1 INT1W = 1	–	–	INT1W = 1 (Note 2)	Noise cancellation conditions are as listed in Table 1-5. They are applied to the INT1 pin when it is used for both edge interrupts.
INT2	INT2	P05 / TC1	IMF · EF_7 = 1 INT2W = 0	INT2ES = 0	INT2ES = 1	–	For falling or rising edge, pulses of less than 7/fc [s] are eliminated as noise. Pulses of equal to or more than 24/fc [s] are regarded as signals. Same applies to pin TC1 (at one edge)
			IMF · EF_7 = 1 INT2W = 1	–	–	INT2W = 1 (Note 2)	Noise cancellation conditions are as listed in Table 1-5. They are applied to the INT2 pin when it is used for both edge interrupts.
INT3	INT3	P06 / OC1	IMF · EF_{11} = 1 INT3W = 0	INT3ES = 0	INT3ES = 1	–	For falling or rising edge, pulses less than 7/fc [s] are cancelled as noise. Pulses equal to or more than 24/fc [s] are regarded as signals.
			IMF · EF_{11} = 1 INT3W = 1	–	–	INT3W = 1 (Note 2)	Noise cancellation conditions are as listed in Table 1-5. They are applied to the INT3 pin when it is used for both edge interrupts.
INT4	INT4	P07	IMF · EF_{20} = 1 INT4W = 0	INT4ES = 0	INT4ES = 1	–	For falling or rising edge, pulses less than 7/fc [s] are cancelled as noise. Pulses equal to or more than 24/fc [s] are regarded as signals.
			IMF · EF_{20} = 1 INT4W = 1	–	–	INT4W = 1 (Note 2)	Noise cancellation conditions are as listed in Table 1-5. They are applied to the INT4 pin is used for both edge interrupts.
INT5	$\overline{\text{INT5}}$	$\overline{\text{P20}}$/$\overline{\text{STOP}}$	IMF · EF_{15} = 1	–	○	–	– (hysteresis input)

Note 1: The noise rejection function is turned off for INT1, INT2, INT3 and INT4, used in SLOW and SLEEP modes. Also, the noise reject times are not constant for pulses input while transiting between operating modes (NORMAL2↔SLOW)

Note 2: To detect the edge at which an interrupt is generated, read bit 6 (INT1EDT) in EINTCR1, bit 6 (INT2EDT) in EINTCR2, bit 6 (INT3EDT) in EINTCR3 and bit 6 (INT4EDT) in EINTCR4, that is, at the beginning of the interrupt processing routine. INT1EDT, INT2EDT and INT3EDT and INT4EDT are valid only for both-edge interrupts (INT1W = 1, INT2W = 1, INT3W = 1 and INT4W = 1). INT1EDT, INT2EDT, INT3EDT and INT4EDT are set to 1 by an interrupt as the non-selected edge; cleared to 0 after read automatically.

When rising edge is selected (INT3ES = 0), bit 6 in INT3EDT is set to 1 when a falling edge is detected at the INT3 pin. (That is, remains 0 if rising edge is detected.)

When falling edge is selected (INT3ES = 1), bit 6 in INT3EDT is set to 1 when a rising edge is detected at the INT3 pin. (That is, remains 0 at falling edge.)

Note 3: The noise rejection function is also affected for timer/counter input (TC1 pin).

Note 4: Noise cancellation/pulse receive conditions for timer/counter are as described below:

① TC1 pin: When the P05 pin is used for TC1 input , INT2W must be cleared to "0". Do not change INT2W to "1".

Note 5: If a noiseless signal is input to the external interrupt pin in the NORMAL 1/2 or IDLE 1/2 mode, the maximum time from the edge of input signal until the IL is set is as follows:

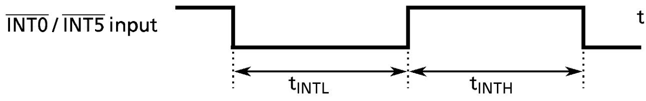
1. INT1 pin
25/fc [s] (when INT1W = 0, falling or rising edge)
25/fc [s] (when INT1W = 1, and NCS (0, 0, 0))
$(2^6/fc) \times 8.5 + 19/fc$ [s] (when INT1W = 1, and NCS (0, 0, 1))
$(2^7/fc) \times 8.5 + 19/fc$ [s] (when INT1W = 1, and NCS (0, 1, 0))
$(2^8/fc) \times 8.5 + 19/fc$ [s] (when INT1W = 1, and NCS (0, 1, 1))
$(2^9/fc) \times 8.5 + 19/fc$ [s] (when INT1W = 1, and NCS (1, 0, 0))
$(2^{10}/fc) \times 8.5 + 19/fc$ [s] (when INT1W = 1, and NCS (1, 0, 1))
$(2^{11}/fc) \times 8.5 + 19/fc$ [s] (when INT1W = 1, and NCS (1, 1, 0))
$(2^{12}/fc) \times 8.5 + 19/fc$ [s] (when INT1W = 1, and NCS (1, 1, 1))

2. INT2 pin
25/fc [s] (when INT2W = 0, falling or rising edge)
25/fc [s] (when INT2W = 1, and NCS (0, 0, 0))
$(2^6/fc) \times 8.5 + 19/fc$ [s] (when INT2W = 1, and NCS (0, 0, 1))
$(2^7/fc) \times 8.5 + 19/fc$ [s] (when INT2W = 1, and NCS (0, 1, 0))
$(2^8/fc) \times 8.5 + 19/fc$ [s] (when INT2W = 1, and NCS (0, 1, 1))
$(2^9/fc) \times 8.5 + 19/fc$ [s] (when INT2W = 1, and NCS (1, 0, 0))
$(2^{10}/fc) \times 8.5 + 19/fc$ [s] (when INT2W = 1, and NCS (1, 0, 1))
$(2^{11}/fc) \times 8.5 + 19/fc$ [s] (when INT2W = 1, and NCS (1, 1, 0))
$(2^{12}/fc) \times 8.5 + 19/fc$ [s] (when INT2W = 1, and NCS (1, 1, 1))

3. INT3 pin
25/fc [s] (when INT3W = 0, falling or rising edge)
25/fc [s] (when INT3W = 1, and NCS (0, 0, 0))
$(2^6/fc) \times 8.5 + 19/fc$ [s] (when INT3W = 1, and NCS (0, 0, 1))
$(2^7/fc) \times 8.5 + 19/fc$ [s] (when INT3W = 1, and NCS (0, 1, 0))
$(2^8/fc) \times 8.5 + 19/fc$ [s] (when INT3W = 1, and NCS (0, 1, 1))
$(2^9/fc) \times 8.5 + 19/fc$ [s] (when INT3W = 1, and NCS (1, 0, 0))
$(2^{10}/fc) \times 8.5 + 19/fc$ [s] (when INT3W = 1, and NCS (1, 0, 1))
$(2^{11}/fc) \times 8.5 + 19/fc$ [s] (when INT3W = 1, and NCS (1, 1, 0))
$(2^{12}/fc) \times 8.5 + 19/fc$ [s] (when INT3W = 1, and NCS (1, 1, 1))

4. INT4 pin 25/fc [s] (when INT4W = 0, falling or rising edge)
25/fc [s] (when INT4W = 1, and NCS (0, 0, 0))
$(2^{6}/fc) \times 8.5 + 19/fc$ [s] (when INT4W = 1, and NCS (0, 0, 1))
$(2^{7}/fc) \times 8.5 + 19/fc$ [s] (when INT4W = 1, and NCS (0, 1, 0))
$(2^{8}/fc) \times 8.5 + 19/fc$ [s] (when INT4W = 1, and NCS (0, 1, 1))
$(2^{9}/fc) \times 8.5 + 19/fc$ [s] (when INT4W = 1, and NCS (1, 0, 0))
$(2^{10}/fc) \times 8.5 + 19/fc$ [s] (when INT4W = 1, and NCS (1, 0, 1))
$(2^{11}/fc) \times 8.5 + 19/fc$ [s] (when INT4W = 1, and NCS (1, 1, 0))
$(2^{12}/fc) \times 8.5 + 19/fc$ [s] (when INT4W = 1, and NCS (1, 1, 1))

Note 6: The pulse width (both "H" and "L" level) for input to the $\overline{\text{INT0}}$ and $\overline{\text{INT5}}$ pins must be over 2 machine cycle.



t_{INTL}, t_{INTH} ≧ 2 tcyc (*Note: tcyc = 4/fm [s]*)

Note 7: When INT0EN = 0, interrupt latch IL_3 is not set even if a falling edge is detected for $\overline{\text{INT0}}$ pin input.

Table 1-5. Noise reject condition for INT1, INT2, INT3, INT4 (both-edge interrupt)

EINTCR1, EINTCR2, EINTCR3, EINTCR4			max. pulse width for noise reject	min. pulse width for immediate signal
NCS2 × 2	NCS1 × 1	NCS0 × 0		
0	0	0	– (histeresis input)	
0	0	1	$(2^{6}/fc) \times 7 - 6/fc$	$(2^{6}/fc) \times 8 + 5/fc$
0	1	0	$(2^{7}/fc) \times 7 - 6/fc$	$(2^{7}/fc) \times 8 + 5/fc$
0	1	1	$(2^{8}/fc) \times 7 - 6/fc$	$(2^{8}/fc) \times 8 + 5/fc$
1	0	0	$(2^{9}/fc) \times 7 - 6/fc$	$(2^{9}/fc) \times 8 + 5/fc$
1	0	1	$(2^{10}/fc) \times 7 - 6/fc$	$(2^{10}/fc) \times 8 + 5/fc$
1	1	0	$(2^{11}/fc) \times 7 - 6/fc$	$(2^{11}/fc) \times 8 + 5/fc$
1	1	1	$(2^{12}/fc) \times 7 - 6/fc$	$(2^{12}/fc) \times 8 + 5/fc$

Note: In SLOW mode, set (NCS × 2, 1, 0) = (0, 0, 0).
In SLOW mode, the digital noise filter in the above table is disabled.

EINTCR
(00037$_H$)

7	6	5	4	3	2	1	0
	INT0 EN		INT4 ES	INT3 ES	INT2 ES	INT1 ES	

(Initial value: *0*0 000*)

INT0EN	P10/$\overline{\text{INT0}}$ pin configuration	0: P10 input/output port 1: $\overline{\text{INT0}}$ pin (Port P10 should be set to an input mode)	R/W
INT4 ES INT3 ES INT2 ES INT1 ES	INT4 to INT1 edge select	0: Rising edge 1: Falling edge	

*Note 1: fc ; High-frequency clock [Hz] * ; don't care*

Note 2: Edge detection during switching edge selection is invalid.

Note 3: Do not change EINTCR only when IMF = 0. After changing EINTCR, interrupt latches of external interrupt inputs must be cleared to "0" using load instruction.

Note 4: In order to change of external interrupt input by rewriting the contents of INT1ES, INT2ES, INT3ES and INT4ES during NORMAL1/2 mode, clear interrupt latches of external interrupt inputs (INT1, INT2, INT3 and INT4) after 8 machine cycles from the time of rewriting. During SLOW mode, 3 machine cycles are required.

*Note 5: In order to change an edge of timer counter input by rewriting the contents of INT1ES, INT2ES, INT3ES and INT4ES during NORMAL1/2 mode, rewrite the contents after timer counter is stopped (TC*s = 0), that is, interrupt disable state. Then, clear interrupt laches of external interrupt inputs (INT1, INT2, INT3 and INT4) after 8 machine cycles from the time of rewriting to change to interrupt enable state. Finally, state timer counter. During SLOW mode, 3 machine cycles are required.*

Example: When changing TC1 pin inputs edge in external trigger timer mode from rising edge to falling edge.

```
                   LD (TC1CR),01001000B    ;  TC1S ← 00 (stop TC1)
                   DI                      ;  IMF ← 0 (disable interrupt service)
                   LD (EINTCR),00000100B   ;  INT2ES ← 1 (change edge selection)
    ──────────     NOP
   8 machine       to
     cycles        NOP
    ──────────     LD (ILL),01111111B      ;  IL7 ← 0 (clear interrupt latch)
                   EI                      ;  IMF ← 1 (enable interrupt service)
                   LD (TC1CR),01111000B    ;  TC1S ← 11 (start TC1)
```

Note 6: Bits 7, 5 and 0 in EINTCR are read in as undefined data when a read instruction is excuted.

Figure 1-20. External Interrupt Control Register (1)

External Interrupt Control Register 1

	7	6	5	4	3	2	1	0	
EINTCR1 (00024_H)	INT1W	INT1 EDT	NCS1			INT1 EDT			(initial value: 0000 00**)

INT1W	INT1 both edge selection	0: Refer to INT1ES 1: Both edge detection	R/W
INT1EDT	Flag indicating an interrupt at selected edge/non-selected edge, when INT1W = 1 (for both-edge interrupts)	0: Interrupt at selected edge or no interrupt 1: Interrupt at non-selected edge	R
NCS1	Noise cancellation time select for INT1 digital noise filter (valid only when INT1W = 1)	000: No noise cancellation 001: Cancels (2^6/fc × 7 – 6/fc) as noise. 010: Cancels (2^7/fc × 7 – 6/fc) as noise. 011: Cancels (2^8/fc × 7 – 6/fc) as noise. 100: Cancels (2^9/fc × 7 – 6/fc) as noise. 101: Cancels (2^{10}/fc × 7 – 6/fc) as noise. 110: Cancels (2^{11}/fc × 7 – 6/fc) as noise. 111: Cancels (2^{12}/fc × 7 – 6/fc) as noise.	R/W
INT1DET	INT1 interrupt detection flag	0: No interrupt 1: Interrupt	R

Note 1: INT1EDT and NCS1 are valid only when the INT1W bit in EINTCR1 is set to 1.
Therefore, when INT1W = 0, the digital noise filter set by the NCS1 bit is disabled.

Note 2: Do not changing the contents of INT1ES (bit 1 in EINTCR) when INT1W is set to 1 (both-edge detention).
If changing the contents of INT1ES during INT1W is set to 1, according to Note 3/4/5 at Figure 1-20 (1).

External Interrupt Control Register 2

	7	6	5	4	3	2	1	0	
EINTCR2 (00025_H)	INT2W	INT2 EDT	NCS2			INT2 EDT			(initial value: 0000 00**)

INT2W	INT2 both edge selection	0: Refer to INT2ES 1: Both edge detection	R/W
INT2EDT	Flag indicating an interrupt at selected edge/non-selected edge, when INT2W = 1 (for both-edge interrupts)	0: Interrupt at selected edge or no interrupt 1: Interrupt at non-selected edge	R
NCS2	Noise cancellation time select for INT2 digital noise filter (valid only when INT2W = 1)	000: No noise cancellation 001: Cancels (2^6/fc × 7 – 6/fc) as noise. 010: Cancels (2^7/fc × 7 – 6/fc) as noise. 011: Cancels (2^8/fc × 7 – 6/fc) as noise. 100: Cancels (2^9/fc × 7 – 6/fc) as noise. 101: Cancels (2^{10}/fc × 7 – 6/fc) as noise. 110: Cancels (2^{11}/fc × 7 – 6/fc) as noise. 111: Cancels (2^{12}/fc × 7 – 6/fc) as noise.	R/W
INT2DET	INT2 interrupt detection flag	0: No interrupt 1: Interrupt	R

Note 1: INT2EDT and NCS2 are valid only when the INT2W bit in EINTCR2 is set to 1.
Therefore, when INT2W = 0, the digital noise filter set by the NCS2 bit is disabled.

Note 2: Do not changing the contents of INT2ES (bit 2 in EINTCR) when INT2W is set to 1 (both-edge detention).
If changing the contents of INT2ES during INT2W is set to 1, according to Note 3/4/5 at Figure 1-20 (1).

External Interrupt Control Register 3

EINTCR3 (00026$_H$) (initial value: 0000 00**)

7	6	5	4	3	2	1	0
INT3W	INT3 EDT	NCS3			INT3 EDT		

INT3W	INT3 both edge selection	0: Refer to INT3ES 1: Both edge detection	R/W
INT3EDT	Flag indicating an interrupt at selected edge/non-selected edge, when INT3W = 1 (for both-edge interrupts)	0: Interrupt at selected edge or no interrupt 1: Interrupt at non-selected edge	R
NCS3	Noise cancellation time select for INT3 digital noise filter (valid only when INT3W = 1)	000: No noise cancellation 001: Cancels $(2^{6}/fc \times 7 - 6/fc)$ as noise. 010: Cancels $(2^{7}/fc \times 7 - 6/fc)$ as noise. 011: Cancels $(2^{8}/fc \times 7 - 6/fc)$ as noise. 100: Cancels $(2^{9}/fc \times 7 - 6/fc)$ as noise. 101: Cancels $(2^{10}/fc \times 7 - 6/fc)$ as noise. 110: Cancels $(2^{11}/fc \times 7 - 6/fc)$ as noise. 111: Cancels $(2^{12}/fc \times 7 - 6/fc)$ as noise.	R/W
INT3DET	INT3 interrupt detection flag	0: No interrupt 1: Interrupt	R

Note 1: INT3EDT and NCS3 are valid only when the INT3W bit in EINTCR3 is set to 1.
Therefore, when INT3W = 0, the digital noise filter set by the NCS3 bit is disabled.

Note 2: Do not changing the contents of INT3ES (bit 3 in EINTCR) when INT3W is set to 1 (both-edge detention).
If changing the contents of INT3ES during INT3W is set to 1, according to Note 3/4/5 at Figure 1-20 (1).

External Interrupt Control Register 4

EINTCR4 (00027$_H$) (initial value: 0000 00**)

7	6	5	4	3	2	1	0
INT4W	INT4 EDT	NCS4			INT4 EDT		

INT4W	INT4 both edge selection	0: Refer to INT4ES 1: Both edge detection	R/W
INT4EDT	Flag indicating an interrupt at selected edge/non-selected edge, when INT4W = 1 (for both-edge interrupts)	0: Interrupt at selected edge or no interrupt 1: Interrupt at non-selected edge	R
NCS4	Noise cancellation time select for INT4 digital noise filter (valid only when INT4W = 1)	000: No noise cancellation 001: Cancels $(2^{6}/fc \times 7 - 6/fc)$ as noise. 010: Cancels $(2^{7}/fc \times 7 - 6/fc)$ as noise. 011: Cancels $(2^{8}/fc \times 7 - 6/fc)$ as noise. 100: Cancels $(2^{9}/fc \times 7 - 6/fc)$ as noise. 101: Cancels $(2^{10}/fc \times 7 - 6/fc)$ as noise. 110: Cancels $(2^{11}/fc \times 7 - 6/fc)$ as noise. 111: Cancels $(2^{12}/fc \times 7 - 6/fc)$ as noise.	R/W
INT4DET	INT4 interrupt detection flag	0: No interrupt 1: Interrupt	R

Note 1: INT4EDT and NCS4 are valid only when the INT4W bit in EINTCR4 is set to 1.
Therefore, when INT4W = 0, the digital noise filter set by the NCS4 bit is disabled.

Note 2: Do not changing the contents of INT4ES (bit 4 in EINTCR) when INT4W is set to 1 (both-edge detention).
If changing the contents of INT4ES during INT4W is set to 1, according to Note 3/4/5 at Figure 1-20 (1).

Figure 1-21. External Interrupt Control Register (2)

1.6 Reset Circuit

The 88CP77/S77/U77 have four types of reset generation procedures: an external reset input, an address trap reset output, a watchdog timer reset output and a system clock reset output. Table 1-6 shows on-chip hardware initialization by reset action.

The malfunction reset output circuit such as watchdog timer reset, address trap reset and system clock reset is not initialized when power is turned on. The $\overline{\text{RESET}}$ pin can output level "L" at the maximum 24/fc [s] (3 μs at 8 MHz) when power is turned on.

Table 1-6. Initializing Internal Status by Reset Action

On-chip hardware	Initial value	On-chip hardwear	Initial value
Program counter (PC)	(FFFFE$_H$ to FFFFC$_H$)	Prescaler and Divider of timing generator	0
Stack pointer (SP)	not initialized		
General-purpose registers (WABCDEHL)	not initialized		
Register bank selector (RBS)	0	Watchdog timer	Enable
Jump status flag (JF)	1		
Zero flag (ZF)	not initialized	Output latches of I/O ports	Refer to I/O port circuitry
Carry flag (CF)	not initialized		
Half carry flag (HF)	not initialized		
Sign flag (SF)	not initialized		
Overflow flag (VF)	not initialized		
Interrupt master enable flag (IMF)	0		
Interrupt individinal enable flags (EF)	0	Control registers	Refer to each of control register
Interrut latchs (IL)	0		

1.6.1 External reset input

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

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

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

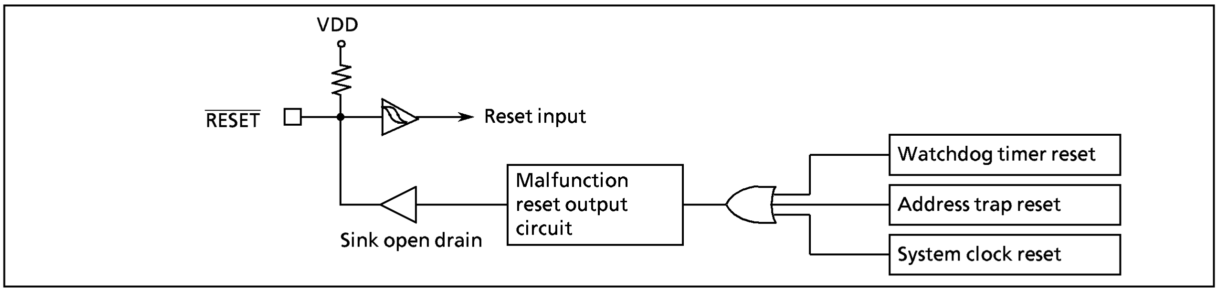
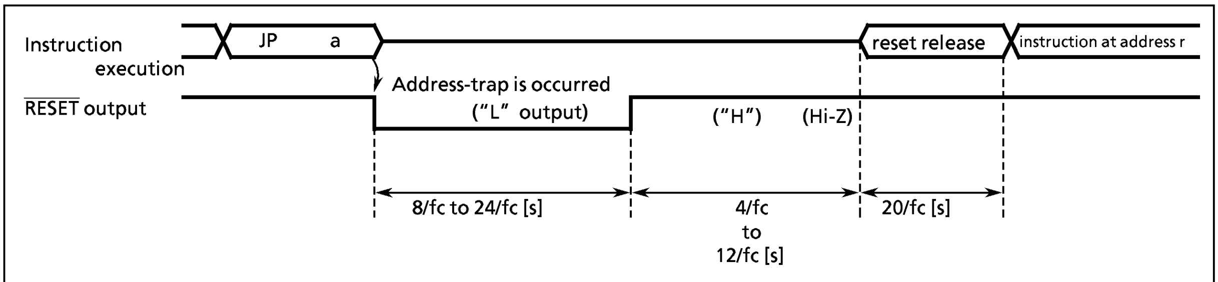


Figure 1-22. Reset Circuit

1.6.2 Address-trap-reset

If the CPU should start looping for some cause such as noise and an attempt be made to fetch an instruction from the on-chip RAM or the SFR area, an address-trap-reset will be generated. Then, the $\overline{\text{RESET}}$ pin output will go low. The reset time is about 8/fc to 24/fc [s] (1 to 3 μs at 8 MHz).



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

Figure 1-23. Address-Trap-Reset

1.6.3 Watchdog timer reset

Refer to Section "2.4 Watchdog Timer".

1.6.4 System-clock-reset

Clearing both XEN and XTEN (bits 7 and 6 in SYSCR2) to "0", clearing XEN to "0" when SYSCK = 0, or clearing XEN to "0" when SYSCK = 1 stops system clock, and causes the microcomputer to deadlock. This can be prevented by automatically generating a reset signal whenever XEN = XTEN = 0 is detected to continue the oscillation. Then, the $\overline{\text{RESET}}$ pin output goes low from high-impedance. The reset time is about 8/fc to 24/fc [s] (1 to 3 μs at 8 MHz).

2. On-Chip Peripheral Functions

2.1 Special Function Registers (SFR)

The 88CP77/S77/U77 use the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function registers (SFR).

The SFR are mapped to addresses 00000$_H$ to 0003F$_H$, and DBR are mapped to address 00F00$_H$ to 00FFF$_H$.

Figure 2-1. shows the 88CP77/S77/U77 SFR, DBR.

Address	Read	Write	Address	Read	Write
00000$_H$	P0 port (write)		00020$_H$	–	SBICR1 (SBI control 1)
01	P1 port		21	SBIDBR (SBI data buffer)	
02	P2 port		22	–	I^{2}CAR (I^{2}C bus address)
03	P3 port		23	SBRSR (SBI status register)	SBICR2 (SBI control 2)
04	P4 port		24	ENTCR1 (External interrupt control 1)	
05	P5 port		25	ENTCR2 (External interrupt control 2)	
06	P6 port		26	ENTCR3 (External interrupt control 3)	
07	P7 port		27	ENTCR4 (External interrupt control 4)	
08	P8 port		28	SIO1SR (SIO1 status register)	SIO1CR1 (SIO1 control 1)
09	P9 port		29	–	SIO1CR2 (SIO1 control 2)
0A	P3CR (P3 port I/O output control)		2A	VFTSR (VFT status register)	VFTCR1 (VFT control 1)
0B	P1CR (P1 port I/O output control)		2B	–	VFTCR2 (VFT control 2)
0C	P4CR (P4 port I/O output control)		2C	EIR$_E$ (Internal enable regoster)	
0D	P5CR (P5 port I/O output control)		2D	reserved	
0E	ADCCR (AD converter control)		2E	IL$_E$ (interrupt latch)	
0F	ADCDR (AD conv.result)	–	2F	reserved	
10	P0 port (read)	–	30	DVCR (DV1 control)	
11	reserved		31	reserved	
12	TREG1B$_L$ (Timer register 1B$_L$)	–	32	reserved	
13	TREG1B$_H$ (Timer register 1B$_H$)	–	33	reserved	
14	–	TC1CR (TC1 control)	34	–	WDTCR1 (Watch dog timer control 1)
15	–	TC2CR (TC2 control)	35	–	WDTCR2 (Watch dog timer control 2)
16	–	TREG1A$_L$ (Timer register 1A$_L$)	36	TBTCR (TBT / TG / DVO control)	
17	–	TREG1A$_H$ (Timer register 1A$_H$)	37	EINTCR (External interrupt control)	
18	–	TREG2$_L$ (Timer register 2$_L$)	38	SYSCR1 (System control 1)	
19	–	TREG2$_H$ (Timer register 2$_H$)	39	SYSCR2 (System control 2)	
1A	reserved		3A	EIR$_L$ (Interrupt enable register)	
1B	–	TREG4 (Timer register 4)	3B	EIR$_H$ (Interrupt enable register)	
1C	–	TC4CR (TC4 control)	3C	IL$_L$ (Internal latch)	
1D	PD port		3D	IL$_H$ (Internal latch)	
1E	PE port		3E	PSW$_L$ (Program status word)	
1F	PF port		3F	PSW$_H$ (Program status word)	

(a) Special function registers

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

Note 2: - ; Cannot be accessed.

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

Note 4: When defining address 0003F$_H$ with assembler symbols, use GRBS.
Address 0003E$_H$ must be GPSW/GFLAG.

Figure 2-1. (a) SFR & DBR

00Fxy

X Y	0 to 1	2 to 3	4 to 5	6 to 7	8 to 9		A to B		C to D		E to F
0	00F00	00F20	00F40	00F60	00F80		00FA0		00FC0		00FE0
1	00F01										
2	00F02										
3	00F03										
4	00F04										
5	00F05										
6	00F06										
7	00F07				VFT display data buffer (126 byte)						
8	00F08										SIO0 transmit data buffer (32 byte)
9	00F09										
A	00F0A										
B	00F0B										
C	00F0C										
D	00F0D										
E	00F0E										
F	00F0F										
0	00F10										
1	00F11										
2	00F12	00F32	00F52	00F72	00F92		00FB2		00FD2		
3	00F13						Read	Write			
4	00F14						OCR0$_L$ (Compare regoster 0$_L$)				
5	00F15						OCR0$_H$ (Compare regoster 0$_H$)		Read	Write	
6	00F16						OCR1$_L$ (Compare regoster 1$_L$)		SIO0SR (SIO0 status)	SIO0CR1 (SIO0 control1)	
7	00F17						OCR1$_H$ (Compare regoster 1$_H$)		reserved	SIO0CR2 (SIO0 control2)	
8	00F18						ICR0A$_L$ (Capter register)	–	SIO1 transmit data buffer (8 byte)		
9	00F19		reserved				ICR0A$_H$ (Capter register)	–			
A	00F1A				Read	Write	ICR0B$_L$ (Capter register)	–			
B	00F1B				ET1CR (ETC1 control)		ICR0B$_H$ (Capter register)	–			
C	00F1C				TMEN (Timer channel enable register)		ICR1A$_L$ (Capter register)	–			
D	00F1D				TMD (Capture mode register)		ICR1A$_H$ (Capter register)	–			
E	00F1E				–	ETREG1$_L$ (Extended timer register 1$_L$)	ICR1B$_L$ (Capter register)	–			
F	00F1F				–	ETREG1$_H$ (Extended timer register 1$_L$)	ICR1B$_H$ (Capter register)	–			00FFF

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

Note 2: – ; Cannot be accessed.

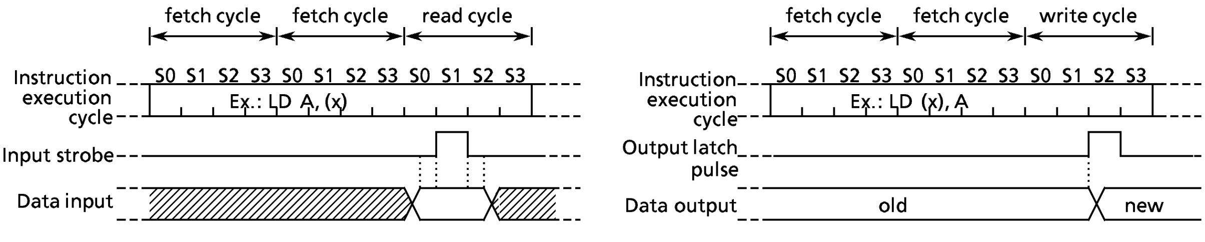
Figure 2-1. (b) SFR & DBR

2.2 I/O Ports

The 88CP77/S77/U77 each have 13 parallel input/output ports (88 pins) each as follows:

①	Port P0	8-bit I/O port	Serial bus interface input/output, Extended timer/counter input/output, timer/counter input and External interrupt input
②	Port P1	8-bit I/O port	External interrupt input, serial interface input/output, timer/ counter input/output, and divider output
③	Port P2	3-bit I/O port	Low-frequency resonator connections, external interrupt input, and STOP mode release signal input
④	Port P3	3-bit I/O port	
⑤	Port P4	8-bit I/O port	Anarog input
⑥	Port P5	4-bit I/O port	Anarog input
⑦	Port P6	8-bit I/O port	VFT output
⑧	Port P7	8-bit I/O port	VFT output
⑨	Port P8	8-bit I/O port	VFT output
⑩	Port P9	8-bit I/O port	VFT output
⑪	Port PD	8-bit I/O port	VFT output
⑫	Port PE	8-bit I/O port	VFT output
⑬	Port PF	5-bit I/O port	VFT output

Each output port contains a latch, which holds the output data. Input ports excluding do not have latches, so the external input data should either be held externally until read or reading should be performed several times before processing. Figure 2-2 shows input/output timing examples.
External data is read from an I/O port in the S1 state of the read cycle during execution of the read instruction. This timing can not be recognized from outside, so that transient input such as chattering must be processed by the program.
Output data changes in the S2 state of the write cycle during execution of the instruction which writes to an I/O port.



(a) Input Timing

(b) Output Timing

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

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

When reading an I/O port except programmable I/O ports P1, P3, P4 and P5, whether the pin input data or the output latch contents are read depends on the instructions, as shown below:

(a) Instructions that read the output latch contents

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

(b) Instructions that read the pin input data

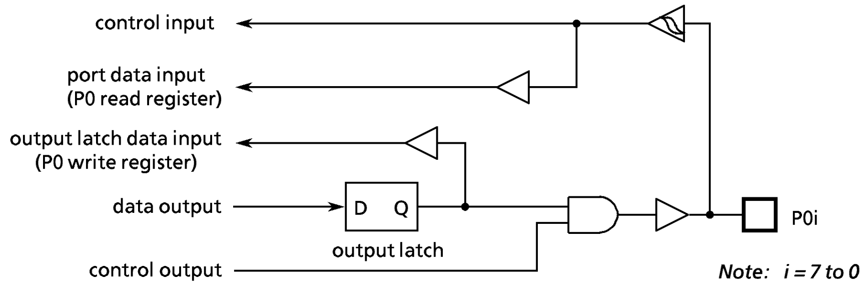
① (HL) side of ADD/ADDC/SUB/SUBB/AND/OR/XOR (src), (HL)

Port P0 has P0 read register and P0 write register. Port P0 can select the pin input data or the output latch contents by reading each registers.

2.2.1 Port P0 (P07 to P00)

Port P0 is an 8-bit input/output port, and is also used as external interrupt input, a serial bus interface input/output, a timer/counter input and extended timer/counter. When used as an input port or a secondary function pin, the output latch should be set to "1". The output latches are initialized to "1" during reset.

Port P0 has P0 read register and P0 write register. Port P0 can select the pin input data or the output latch contents by reading each registers.



Note: i = 7 to 0

	7	6	5	4	3	2	1	0	
P0 write (00000$_H$) R/W	P07 INT4	P06 INT3 OC1	P05 TC1 INT2	P04 OC0 INT1	P03 IC0	P02 IC1 ETC1 $\overline{\text{SCK3}}$	P01 SCL SI3	P00 SDA SO3	(Initial value: 1111 1111)

P0 read (00010$_H$) Read only	P07	P06	P05	P04	P03	P02	P01	P00

Figure 2-3. Port P0 and P0CR

The terminal of Sink open drain port needs to set it as follows.

Note1: Set output latch to "0" when an unused port is open.

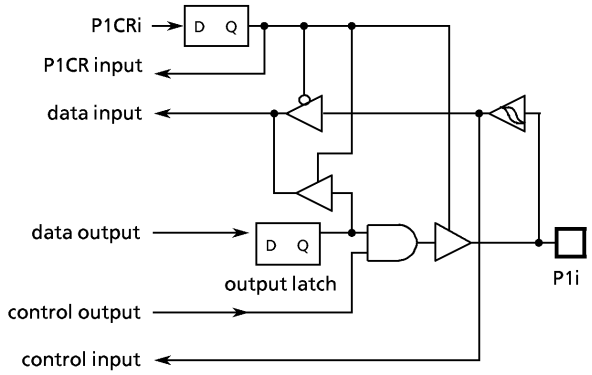
Note2: An input level of an input which is assigned as input should be kept high or low level in all operation mode.

Note3: Output level is kept during STOP mode. Set output latch to "0" before changing to the STOP mode in case cutting off power supply which is connected as pulls up a voltage level of the port through register.

2.2.2 Port P1 (P17 to P10)

Port P1 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P1 input/output control register (P1CR). Port P1 is configured as an input if its corresponding P1CR bit is cleared to "0", and as an output if its corresponding P1CR bit is set to "1". During reset, P1CR is initialized to "0", which configures port P1 as an input . The P1 output latches are also initialized to "0". Data is written into the output latch regardless of P1CR contents. Therfore initial output data should be written into the output latch before setting P1CR. Port P1 is also used as a serial interface input/output, an external interrupt input, a timer/counter input, and a divider output. When used as a secondary function pin, the input pins should be set to the input mode, and the output pins should be set to the output mode and beforehand the output latch should be set to "1".

Note: Port set to the input mode read the pin status. When input pin and output in exist in port P1 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.



Note: i = 7 to 0

P1 (00001_H)

7	6	5	4	3	2	1	0	
P17	P16 $\overline{\text{PWM}}$ $\overline{\text{PDO}}$ SO0	P15 TC4 SI0	P14 $\overline{\text{DVO}}$ $\overline{\text{SCK0}}$	P13	P12 SO1 TC2	P11 SI1 $\overline{\text{INT0}}$	P10 $\overline{\text{SCK1}}$	(Initial value: 0000 0000)

P1CR ($0000B_H$)

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

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

Figure 2-4. Port P1 and P1 I/O control register

The terminal of Tri-state port needs to set it as follows.

Note1: When an unused port is open, change the port mode to "output" after setting the output latch high or low.

Note2: An input level of an input which is assigned as input should be kept high or low level in all operation mode.

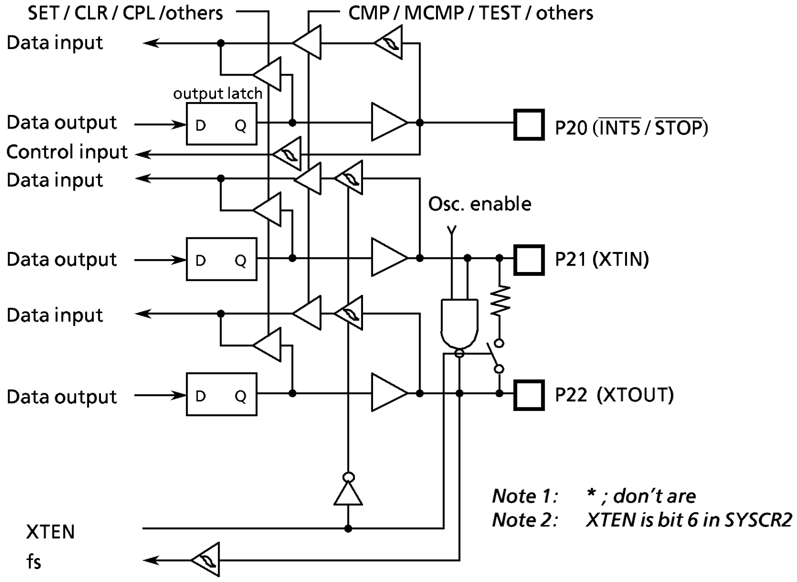
2.2.3 Port P2 (P22 to P20)

Port P2 is a 3-bit input/output port. It is also used as an external interrupt input, and low-frequency crystal connection pins When used as an input port, or the secondary function pin, the output latch should be set to “1”. During reset , the output latches are initialized to “1”.

A low-frequency crystal (32.768 kHz) is connected to pins P21 (XTIN) and P22 (XTOUT) in the dual-clock mode. In the single-clock mode, pins P21 and P22 can be used as normal input/output ports.

It is recommended that the P20 pin should be used as an external interrupt input, a STOP mode release signal input, or an input port. If used as an output port, the interrupt latch is set on the falling edge of the output pulse.

When a read instruction for port P2 is executed, bits 7 to 3 in P2 read in as undefined data.



*Note 1: * ; don’t are*
Note 2: XTEN is bit 6 in SYSCR2

	7	6	5	4	3	2	1	0
P2 (00002$_H$)						P22 XTOUT	P21 XTIN	P20 $\overline{\text{INT5}}$ STOP

(Initial value: **** *111)

Figure 2-5. Port P2

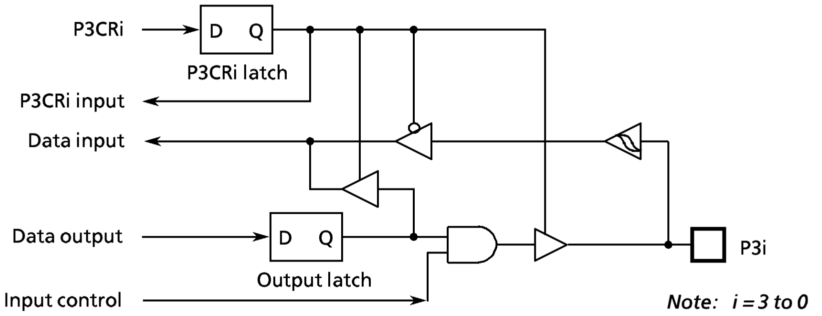
The terminal of Sink open drain port needs to set it as follows.

Note1: Set output latch to “0” when an unused port is open.
Note2: An input level of an input which is assigned as input should be kept high or low level in all operation mode.
Note3: Output level is kept during STOP mode. Set output latch to “0” before changing to the STOP mode in case cutting off power supply which is connected as pulls up a voltage level of the port through register.

2.2.4 Port P3 (P33 to P30)

Port P3 is an 4-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P3 input/output control register (P3CR). Port P3 is configured as an input if its corresponding P3CR bit is cleared to "0", and as an output if its corresponding P3CR bit is set to "1". During reset, P3CR is initialized to "0", which configures port P3 as an input . The P3 output latches are also initialized to "0". Data is written into the output latch regardless of P3CR contents. Therefore initial output data should be written into the output latch before setting P3CR.
When a read instruction for port P3 is executed, bit7 to 4 in P3 read in as undefined data.

Note: Port set to the input mode read the pin status. When input pin and output in exist in port P3 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.



Note: i = 3 to 0

	7	6	5	4	3	2	1	0	
P3 (00003$_H$) R/W					P33	P32	P31	P30	(Initial value: **** 0000)
P3CR (0000A$_H$)									(Initial value: **** 0000)

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

Figure 2-6. P3 and Port P3 I/O control register

The terminal of Tri-state port needs to set it as follows.

Note1: When an unused port is open, change the port mode to "output" after setting the output latch high or low.
Note2: An input level of an input which is assigned as input should be kept high or low level in all operation mode.

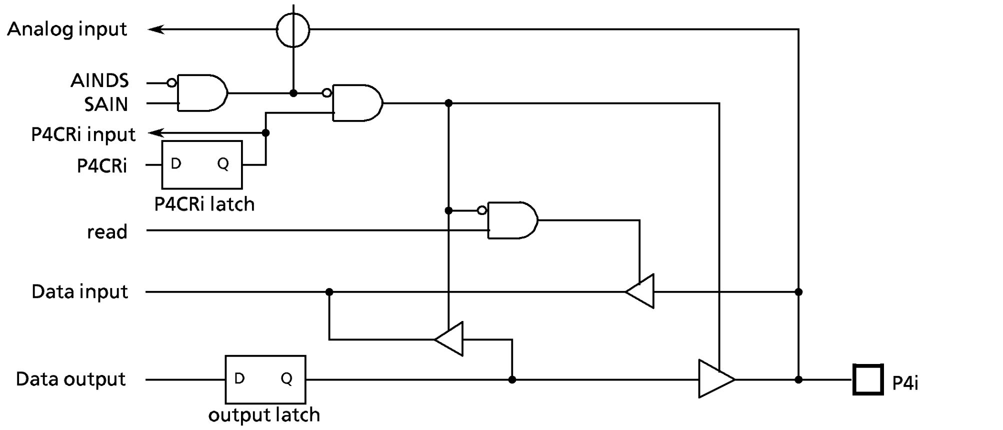
2.2.5 Port P4 (P47 to P40)

Ports P4 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P4 input/output control register (P4CR).

Port P4 is also used as an analog input for the AD converter. When used as an analog input, AINDS (bit 4 in the ADCCR) must be cleared to "0" and its corresponding P4CR bit must be set to "1". In this case, unuse pin as analog input is configured as only input port.

During reset, AINDS is initialized to "0" and all bits of P4CR are initialized to "1", which configures port P4 as analog input. The P4 output latches are initialized to "0". Data is written into the output latch regardless of the P4CR contents. Therefore initial output data should be written into the output latch before setting P4CR.

Note: Port set to the input mode read the pin status. When input pin and output in exist in port P4 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.



Note: i = 0 to 7

P4 (00004$_H$)

7	6	5	4	3	2	1	0
P47 AIN07	P46 AIN06	P45 AIN05	P44 AIN04	P43 AIN03	P42 AIN02	P41 AIN01	P40 AIN00

(Initial value: 0000 0000)

P4CR (0000C$_H$)

7	6	5	4	3	2	1	0

(Initial value: 0000 0000)

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

Figure 2-7. P4 and Port P4 control register

The terminal of Tri-state port needs to set it as follows.

Note1: When an unused port is open, change the port mode to "output" after setting the output latch high or low.

Note2: An input level of an input which is assigned as input should be kept high or low level in all operation mode.

2.2.6 Port P5 (P53 to P50)

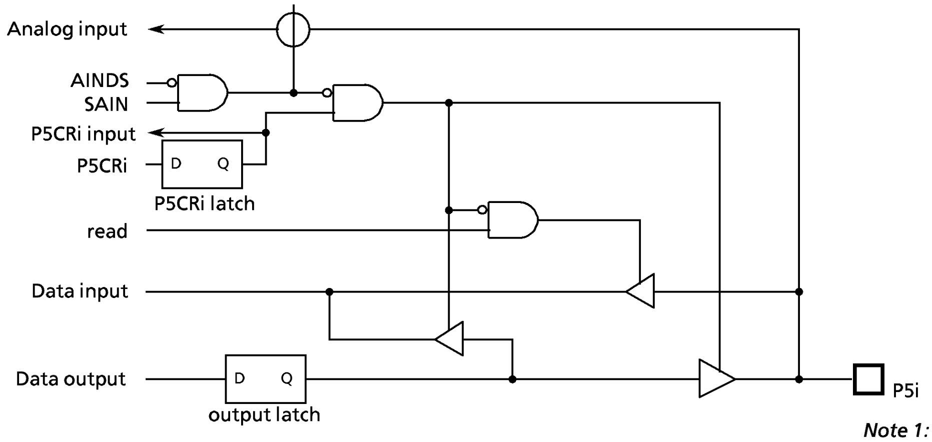
Ports P5 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit under software control. Input/output mode is specified by the corresponding bit in the port P5 input/output control register (P5CR).

Port P5 is also used as an analog input for the AD converter. When used as an analog input, AINDS (bit 4 in the ADCCR) must be cleared to "0" and its corresponding P5CR bit must be set to "1". In this case, unuse pin as analog input is configured as only input port.

During reset, AINDS is initialized to "0" and all bits of P5CR are initialized to "1", which configures port P5 as analog input. The P5 output latches are initialized to "0". Data is written into the output latch regardless of the P5CR contents. Therefore initial output data should be written into the output latch before setting P5CR.

When a read instruction for port P5 is executed, bit7 to 4 in P5 read in as undefined data.

Note: Port set to the input mode read the pin status. When input pin and output in exist in port P5 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.



Note 1: i = 0 to 3
*Note 2: * ; don't care*

P5 (00005$_H$)

7	6	5	4	3	2	1	0
				P53 AIN13	P52 AIN12	P51 AIN11	P50 AIN10

(Initial value: **** 0000)

P5CR (0000D$_H$)

7	6	5	4	3	2	1	0

(Initial value: **** 0000)

P5CR	Port P5 I/O control (Set for each bit individually)	0: Input mode 1: Output mode	R/W

Figure 2-8. P5 and Port P5 I/O control register

The terminal of Tri-state port needs to set it as follows.

Note1: When an unused port is open, change the port mode to "output" after setting the output latch high or low.

Note2: An input level of an input which is assigned as input should be kept high or low level in all operation mode.

2.2.7 Port 6 (P67 to P60), P7 (P77 to P70), P8 (P87 to P80), P9 (P97 to P90)

Port P6, P7, P8 and P9 are an 8-bit high-breakdown voltage input / output port, and also used as a VFT driver output, which can directly drive vacuum fluorescent tube (VFT).

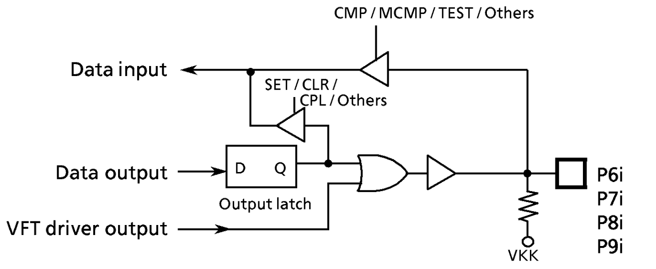
When used as an VFT driver output, the output latch should be cleared to "0".

Pins not used for VFT driver output can be used as I/O ports.

When use an VFT driver and normal input / output at the same time, VET driver output data buffer memory (DBR) need to cleared to "0".

The output latches are initialized to "0" during reset.

It recommends that port P6, P7, P8 and P9 shoud be used to drive directly drive vacuum fluorescent tube (VFT), since this port has a pull down resistance.



Note: i = 7 to 0

	7	6	5	4	3	2	1	0	
P6 (00006$_H$)	P67 V7	P66 V6	P65 V5	P64 V4	P63 V3	P62 V2	P61 V1	P60 V0	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P7 (00007$_H$)	P77 V15	P76 V14	P75 V13	P74 V12	P73 V11	P72 V10	P71 V9	P70 V8	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P8 (00008$_H$)	P87 V23	P86 V22	P85 V21	P84 V20	P83 V19	P82 V18	P81 V17	P80 V16	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P9 (00009$_H$)	P97 V31	P96 V30	P95 V29	P94 V28	P93 V27	P92 V26	P91 V25	P90 V24	(Initial value: 0000 0000)

Figure 2-9. P6, P7, P8, P9 ports

The terminal of Source open drain port needs to set it as follows.

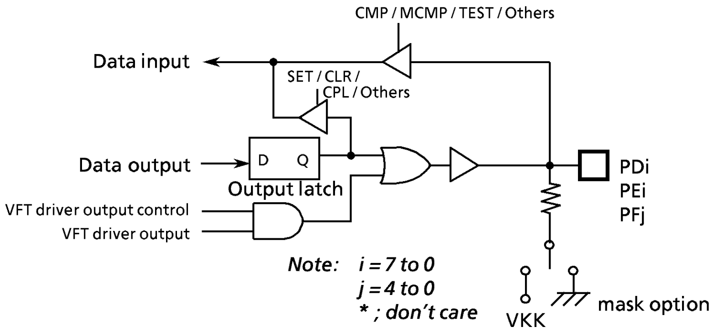
Note: When an unused port is open, clear corresponding DBR data for the port to "0" and set the output latch to "0". Vkk needs to be tied to power supply in this case.

2.2.8 PD (PD7 to PD0), PE (PE7 to PE0), PF (PF4 to PF0)

Port PD, PE and PF are high-breakdown voltage input / output port, and also used as a VFT driver output, which can directly drive vacuum fluorescent tube (VFT).

General-purpose or segment can be selected for each bit by VSEL (bit 4 to 0) in VFT driver control register 1 (VFTCR1). The VSEL is cleared to "0" during reset, which used as an input mode. When used as an input port or VFT driver output, the output latch set to "0". The output latches are initialized to "0" during reset.

When a read instruction for port PF is executed bit 7 to 5 in PF read in as undefined data.



Note: i = 7 to 0
j = 4 to 0
** ; don't care*

	7	6	5	4	3	2	1	0	
PD (0001D$_H$)	PD7 V39	PD6 V38	PD5 V37	PD4 V36	PD3 V35	PD2 V34	PD1 V33	PD0 V32	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
PE (0001E$_H$)	PE7 V47	PE6 V46	PE5 V45	PE4 V44	PE3 V43	PE2 V42	PE1 V41	PE0 V40	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
PF (0001F$_H$)				PF4 V52	PF3 V51	PF2 V50	PF1 V49	PF0 V48	(Initial value: ***0 0000)

Figure 2-10. PD, PE, PF Ports

The terminal of Source open drain port needs to set it as follows.

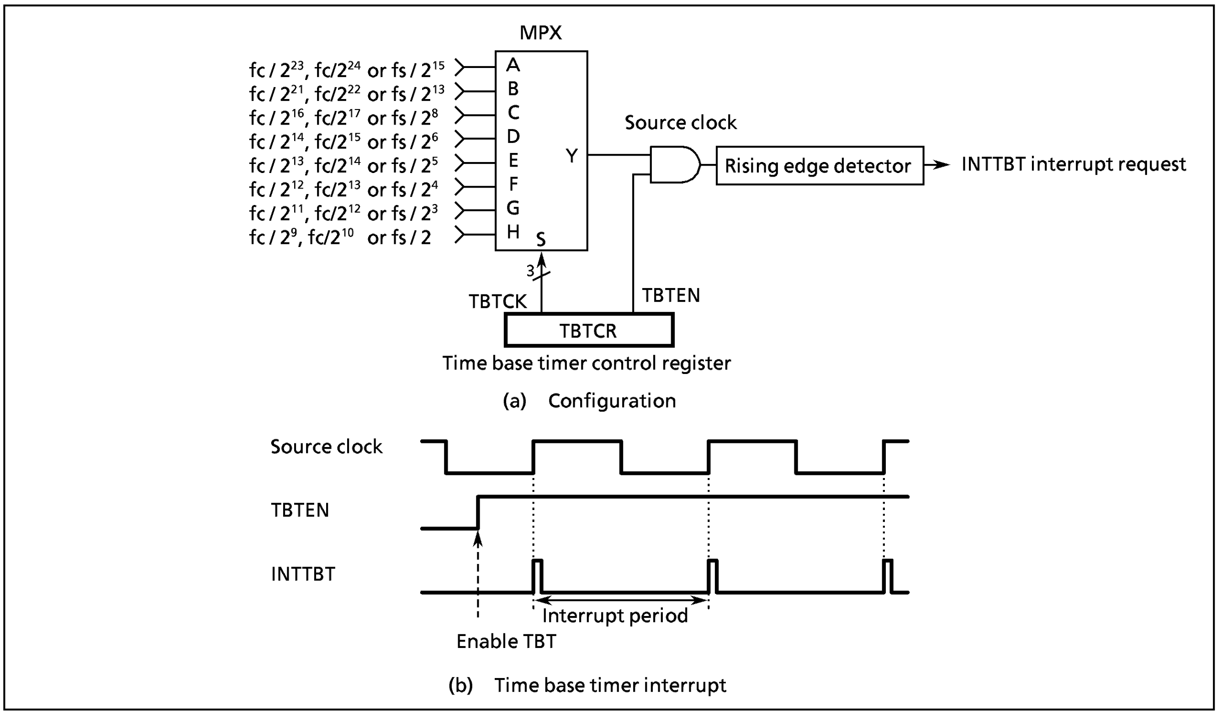
Note: When an unused port is open, clear corresponding DBR data for the port to "0" and set the output latch to "0". Vkk needs to be tied to power supply in this case. When the port without built-in Rk pull-down register is not used, clear corresponding DBR data for the port to "0", set the output latch to "0" and apply the GND level to the port.

2.3 Time Base Timer (TBT)

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

An INTTBT is generated on the first rising edge of source clock (the divider output of the timing generator) after the time base timer has been enabled. The divider is not cleared by the program; therefore, only the first interrupt may be generated ahead of the set interrupt period (Figure 2-11.(b)).

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


MPX
$fc/2^{23}$, $fc/2^{24}$ or $fs/2^{15}$
$fc/2^{21}$, $fc/2^{22}$ or $fs/2^{13}$
$fc/2^{16}$, $fc/2^{17}$ or $fs/2^{8}$
$fc/2^{14}$, $fc/2^{15}$ or $fs/2^{6}$
$fc/2^{13}$, $fc/2^{14}$ or $fs/2^{5}$
$fc/2^{12}$, $fc/2^{13}$ or $fs/2^{4}$
$fc/2^{11}$, $fc/2^{12}$ or $fs/2^{3}$
$fc/2^{9}$, $fc/2^{10}$ or $fs/2$
A
B
C
D
E
F
G
H
S
Y
3
Source clock
Rising edge detector
INTTBT interrupt request
TBTCK
TBTEN
TBTCR
Time base timer control register
Source clock
TBTEN
INTTBT
Interrupt period
Enable TBT

(a) Configuration

(b) Time base timer interrupt

Figure 2-11. Time Base Timer

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

```
                LD      (TBTCR) , 00000010B
                LD      (TBTCR) , 00001010B
                SET     (EIRL) . 6
```

TBTCR
(00036$_H$)

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

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

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

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

Figure 2-12. Time Base Timer Control Register

Table 2-1. Time Base Timer Interrupt Frequency (Example; fc = 8 MHz, fs = 32.768 kHz)

TBTCK	Time base timer interrupt frequency [Hz]				
	NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode
	DV7CK = 0		DV7CK = 1		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
000	0.95	0.48	1	1	1
001	3.81	1.91	4	4	4
010	122.07	61.04	128	128	–
011	488.28	244.14	512	512	–
100	976.56	488.28	1024	1024	–
101	1953.13	976.56	2048	2048	–
110	3906.25	1953.13	4096	4096	–
111	15625.00	7812.50	16384	16384	–

2.4 Watchdog Timer (WDT)

The watchdog timer is a fail-safe system to rapidly detect the CPU malfunctions such as endless looping caused by noise or the like, or deadlock and resume the CPU to the normal state.

The watchdog timer signal for detecting malfunction can be selected either a reset output or a pseudo non-maskable interrupt request. However, selection is possible only once after reset. At first the reset output is selected.

When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals.

2.4.1 Watchdog timer configuration

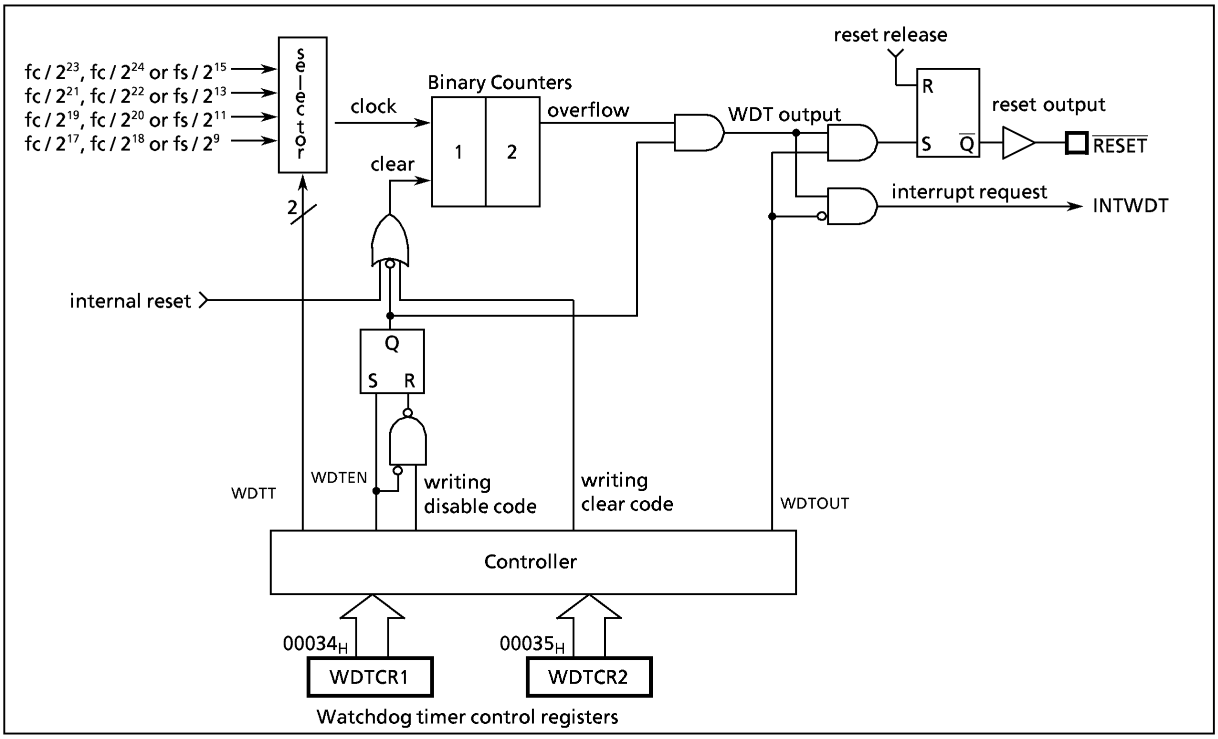


Figure 2-13. Watchdog Timer Configuration

2.4.2 Watchdog timer control

Figure 2-14 shows the watchdog timer control registers (WDTCR1, WDTCR2). The watchdog timer is automatically enabled after reset.

(1) Malfunction detection methods using the watchdog timer

The CPU malfunction is detected at follows.

① Setting the detection time, selecting output, and clearing the binary counter.

② Repeatedly clearing the binary counter within the setting detection time

If the CPU malfunctions such as endless looping or deadlock occur for any cause, the watchdog timer output will become active at the rising of an overflow from the binary counters unless the binary counters are cleared. At this time, when WDTOUT = 1 a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the internal hardware and the external circuit. When WDTOUT = 0, a watchdog timer interrupt(INTWDT) is generated.

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

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

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

Watchdog Timer Register 1

WDTCR1 (00034$_H$)

7	6	5	4	3	2	1	0
				WDT EN	WDTT		WDT OUT

(Initial value: **** 1001)

<table>
<tr><td rowspan="1">WDTEN</td><td>Watchdog timer enable / disable</td><td colspan="6">0: Disable (It is necessary to write the disable code to WDTCR2)
1: Enable</td><td rowspan="8">write only</td></tr>
<tr><td rowspan="6">WDTT</td><td rowspan="6">Watchdog timer detection time [s]</td><td rowspan="3"></td><td colspan="4">NORMA1/2 mode</td><td rowspan="3">SLOW mode</td></tr>
<tr><td colspan="2">DV7CK = 0</td><td colspan="2">DV7CK = 1</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>00
01</td><td>2^{25} / fc
2^{23} / fc</td><td>2^{26} / fc
2^{24} / fc</td><td>2^{17} / fs
2^{15} / fs</td><td>2^{17} / fs
2^{15} / fs</td><td>2^{17} / fs
2^{15} / fs</td></tr>
<tr><td>10</td><td>2^{21} / fc</td><td>2^{22} / fc</td><td>2^{13} / fs</td><td>2^{13} / fs</td><td>2^{13} / fs</td></tr>
<tr><td>11</td><td>2^{19} / fc</td><td>2^{20} / fc</td><td>2^{11} / fs</td><td>2^{11} / fs</td><td>2^{11} / fs</td></tr>
<tr><td>WDTOUT</td><td>Watchdog timer output select</td><td colspan="6">0: Interrupt request
1: Reset output</td></tr>
</table>

Note 1: WDTOUT cannot be set to "1" by program after clearing WDTOUT to "0".
*Note 2: fc; High-frequency clock [Hz], fs; Low-frequency clock [Hz], *; don't care*
Note 3: WDTCR1 is a write-only register and must not be used with any of read-modify-write instructions.
Note 4: Disable the watchdog timer or clear the counter just before switching to STOP mode.
When the counter is cleared just before switching to STOP mode, clear the counter again subsequently to releasing STOP mode.

Watchdog Timer Register 2

WDTCR2 (00035$_H$)

7	6	5	4	3	2	1	0

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

WDTCR2	Watchdog timer control code write register	4E$_H$: Watchdog timer binary counter clear (clear code) B1$_H$: Watchdog timer disable (disable code) others: Invalid	write only

Note 1: The disable code is invalid unless written when WDTEN = 0.
*Note 2: *; don't care*
Note 3: The binary counter of the watchdog timer must not be cleared by the interrupt task.

Figure 2-14. Watchdog Timer Control Registers

(2) Watchdog timer enable

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

Example: Enables watchdog timer

(3) Watchdog timer disable

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

Example: Disables watchdog timer

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

Table 2-2. Watchdog Timer Detection Time (Example: fc = 8 MHz, fs = 32.768 kHz)

WDTT	Watchdog timer detection time [s]				
	NORMAL1/2 mode				SLOW mode
	DV7CK = 0		DV7CK = 1		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
00	4.194	8.389	4	4	4
01	1.048	2.097	1	1	1
10	262.144 m	524.288 m	250 m	250 m	250 m
11	65.536 m	131.072 m	62.5 m	62.5 m	62.5 m

2.4.3 Watchdog timer interrupt (INTWDT)

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

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

Example: Watchdog timer interrupt setting up

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

2.4.4 Watchdog timer reset

If the watchdog timer output becomes active, a reset is generated, which drives the $\overline{\text{RESET}}$ pin (sink open drain input / output with pull-up) low to reset the internal hardware and external circuits. The reset output time is about 8/fc to 24/fc [s] (1 to 3 μs at fc = 8 MHz, fcgck = fc).

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

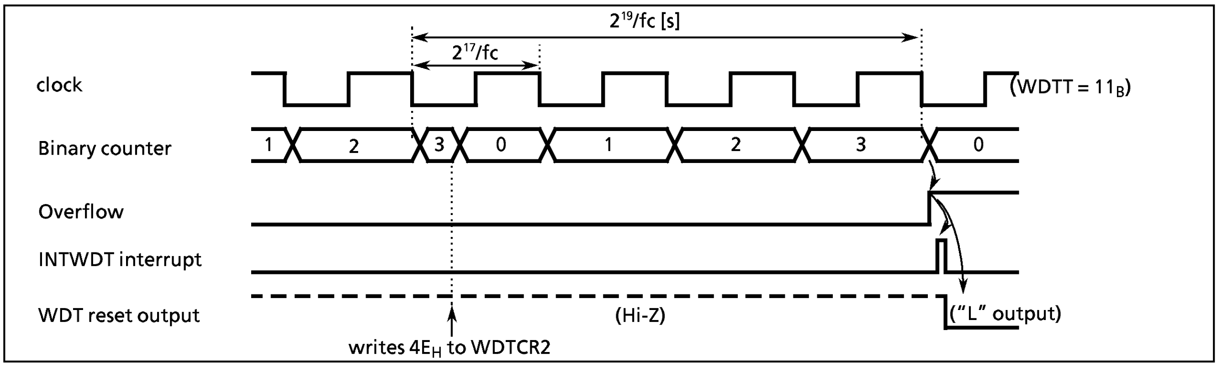


Figure 2-15. Watchdog Timer Interrupt / Reset

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

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

TBTCR (00036_H)

7	6	5	4	3	2	1	0
DVOEN	DVOCK		(DV7CK)	(TBTEN)	(TBTCK)		

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

			NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode	
DVOEN	Divider output enable / disable	0: Disable 1: Enable						R/W
			DV7CK = 0		DV7CK = 1			
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1		
DVOCK	Divider output ($\overline{\text{DVO}}$) frequency selection [Hz]	00	$fc/2^{13}$	$fc/2^{14}$	$fs/2^5$	$fs/2^5$	$fs/2^5$	
		01	$fc/2^{12}$	$fc/2^{13}$	$fs/2^4$	$fs/2^4$	$fs/2^4$	
		10	$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	$fs/2^3$	
		11	$fc/2^{10}$	$fc/2^{11}$	$fs/2^2$	$fs/2^2$	$fs/2^2$	

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

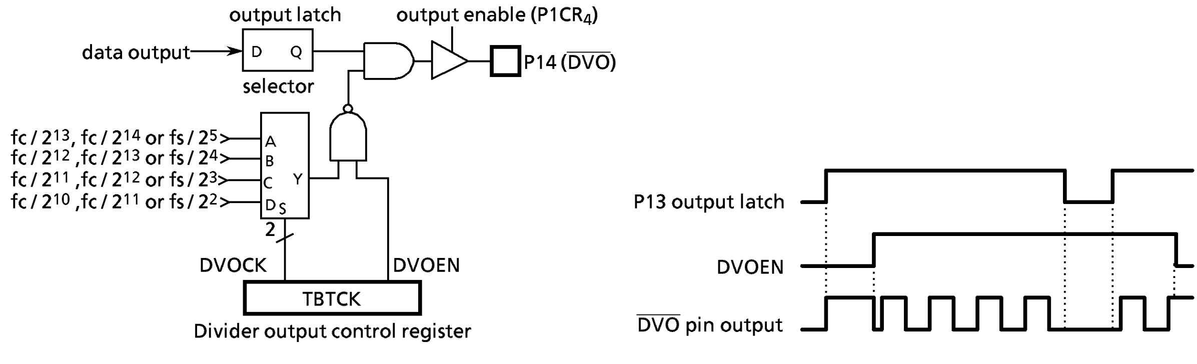
Figure 2-16. Divider Output Control Register

Example: 976 Hz pulse output (at fc = 8 MHz, DV1CK = 0, DV7CK = 0,)

```
SET  (P1).4              ;  P14 output latch←1
LD (P1CR), 00010000B     ;  Configures P14 as an output mode
LD (TBTCR), 10000000B    ;  DVOEN←1, DVOCK←00
```

Table 2-3. Divider Output Frequency (Example: at fc = 8 MHz, fs = 32.768 kHz)

DVOCK	Divider output frequency [kHz]				
	NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode
	DV7CK = 0		DV7CK = 1		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
00	0.976	0.488	1.024	1.024	1.024
01	1.953	0.976	2.048	2.048	2.048
10	3.906	1.953	4.096	4.096	4.096
11	7.812	3.906	8.192	8.192	8.192


output latch
output enable ($P1CR_4$)
data output
D Q
selector
P14 ($\overline{\text{DVO}}$)
fc / 2¹³, fc / 2¹⁴ or fs / 2⁵
fc / 2¹², fc / 2¹³ or fs / 2⁴
fc / 2¹¹, fc / 2¹² or fs / 2³
fc / 2¹⁰, fc / 2¹¹ or fs / 2²
A
B
C Y
D S
2
DVOCK
DVOEN
TBTCK
Divider output control register
P13 output latch
DVOEN
$\overline{\text{DVO}}$ pin output

(a) Configuration of divider output circuit (b) Divider output timing chart

Figure 2-17. Divider Output

2.6 16-bit Timer / Counter 1 (TC1)

2.6.1 Configuration

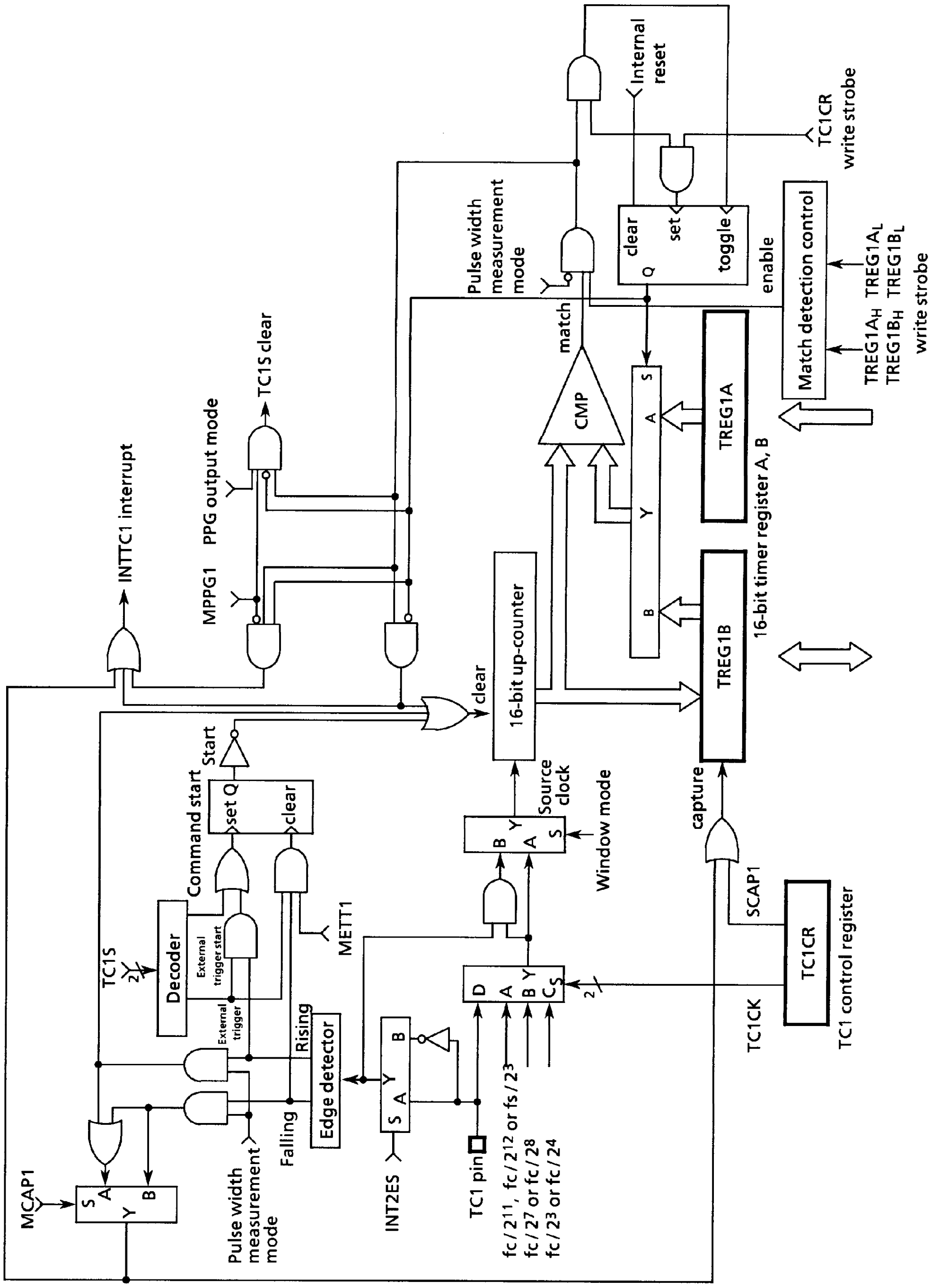


Figure 2-18. Timer / Counter 1 (TC1)

2.6.2 Control

The timer / counter 1 is controlled by a timer / counter 1 control register (TC1CR) and two 16-bit timer registers (TREG1A and TREG1B). Reset does not affect TREG1A and TREG1B.

Timer Register 1

TREG1A
(00016, 00017H)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREG1AH (00017H)								TREG1AL (00016H)							

Write only

TREG1B
(00012, 00013H)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREG1BH (00013H)								TREG1BL (00012H)							

Read only

Timer Counter 1 Control Register

TC1CR
(00014H)

7	6	5	4	3	2	1	0
"1"	SCAP1 MCAP1 METT1	TC1S		TC1CK		TC1M	

(Initial value: *000 0000)

<table>
<tr><td>TC1M</td><td>TC1 operating mode select</td><td colspan="6">00: Timer / external trigger timer / event counter mode
01: Window mode
10: Pulse width measurement mode
11: reserved</td><td rowspan="12">Write only</td></tr>
<tr><td rowspan="7">TC1CK</td><td rowspan="7">TC1 source clock select [Hz]</td><td rowspan="3"></td><td colspan="4">NORMAL1/2, IDLE1/2 mode</td><td rowspan="3">SLOW, SLEEP mode</td></tr>
<tr><td colspan="2">DV7CK = 0</td><td colspan="2">DV7CK = 1</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>00</td><td>$fc/2^{11}$</td><td>$fc/2^{12}$</td><td>$fs/2^{3}$</td><td>$fs/2^{3}$</td><td>$fs/2^{3}$</td></tr>
<tr><td>01</td><td>$fc/2^{7}$</td><td>$fc/2^{8}$</td><td>$fc/2^{7}$</td><td>$fc/2^{8}$</td><td>–</td></tr>
<tr><td>10</td><td>$fc/2^{3}$</td><td>$fc/2^{4}$</td><td>$fc/2^{3}$</td><td>$fc/2^{4}$</td><td>–</td></tr>
<tr><td>11</td><td colspan="5">External clock (TC1 pin input)</td></tr>
<tr><td>TC1S</td><td>TC1 start control</td><td colspan="6">00: Stop & counter clear
01: Command start
10: Reserved
11: External trigger start</td></tr>
<tr><td>SCAP1</td><td>Software capture control</td><td colspan="3">0: –</td><td colspan="3">1: Software capture trigger</td></tr>
<tr><td>MCAP1</td><td>Pulse width measurement mode control</td><td colspan="3">0: Double edge capture</td><td colspan="3">1: Single edge capture</td></tr>
<tr><td>METT1</td><td>External trigger timer mode control</td><td colspan="3">0: Trigger start</td><td colspan="3">1: Trigger start & stop</td></tr>
</table>

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

Note 2: Writing to the lower byte of the timer registers (TREG1AL, TREG1BL), the comparison is inhibited until the upper byte(TREG1AH, TREG1BH) is written. Only the lower byte of the timer registers can not be changed. After writing to the upper byte, any match during 1 machine cycle (instruction execution cycle) is ignored.

Note 3: Set the mode, source clock, edge (including INT2ES) when TC1 stops (TC1S=00).

Note 4: Software capture can be used in only timer and event counter modes. SCAP1 is automatically cleared to "0" after capturing.

Note 5: Values to be loaded to timer registers must satisfy the following condition.
TREG1A>0

Note 6: TC1CR and TREG1A are write-only registers and must not be used with any of the read-modify-write instructions such as SET, CLR, etc.

Note 7: Please use the auto-capture function in the operative condition of TC1.
A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Please read the capture value in a capture enabled condition.

Note 8: Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

Figure 2-19. Timer Registers and TC1 Control Register

2.6.3 Function

Timer / counter 1 has five operating modes: timer, external trigger timer, event counter, window, pulse width measurement.

(1) **Timer** mode

In this mode, counting up is performed using the internal clock. The contents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to "0". Counting up resumes after the counter is cleared. The current contents of up-counter can be transferred to TREG1B by setting SCAP1 (bit 6 in TC1CR) to "1"(software capture function). SCAP1 is automatically cleared after capturing.

Table 2-4. Source Clock (internal clock) for Timer / Counter 1 (Example: at fc = 8 MHz, fs = 32.768 kHz)

TC1CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum time setting	Resolution [μs]	Maximum time setting	Resolution [μs]	Maximum time setting	Resolution [μs]	Maximum time setting
00	256.00	16.77	512.00	33.55	244.14	16.0	244.14	16.0
01	16.00	1.04	32.00	2.09	16.00	1.04	32.00	2.09
10	1.00	65.53 m	2.00	131.07 m	1.00	65.53 m	2.00	131.07 m

TC1CK	SLOW, SLEEP mode	
	Resolution [μs]	Maximum time setting [s]
00	244.14	16.0
01	–	–
10	–	–

Example 1: Sets the timer mode with source clock fs/2^3 [Hz] and generates an interrupt 1 later (at fs = 32.768 kHz).

LDW (TREG1A), 1000H ; Sets the timer register (1 s ÷ 2^3 / fs = 1000_H)
SET (EIRL). EF4 ; Enable INTTC1
EI
LD (TC1CR), 00010000B ; Starts TC1

Note: TC1CR is a wire-only register and must not be used with [SET(TC1CR).4] instruction.

Example 2: Software capture

```
LD    (TC1CR) , 01010000B  ;   SCAP1←1 (Captures)
LD    WA,  (TREG1B)        ;   Reads the capture value
```

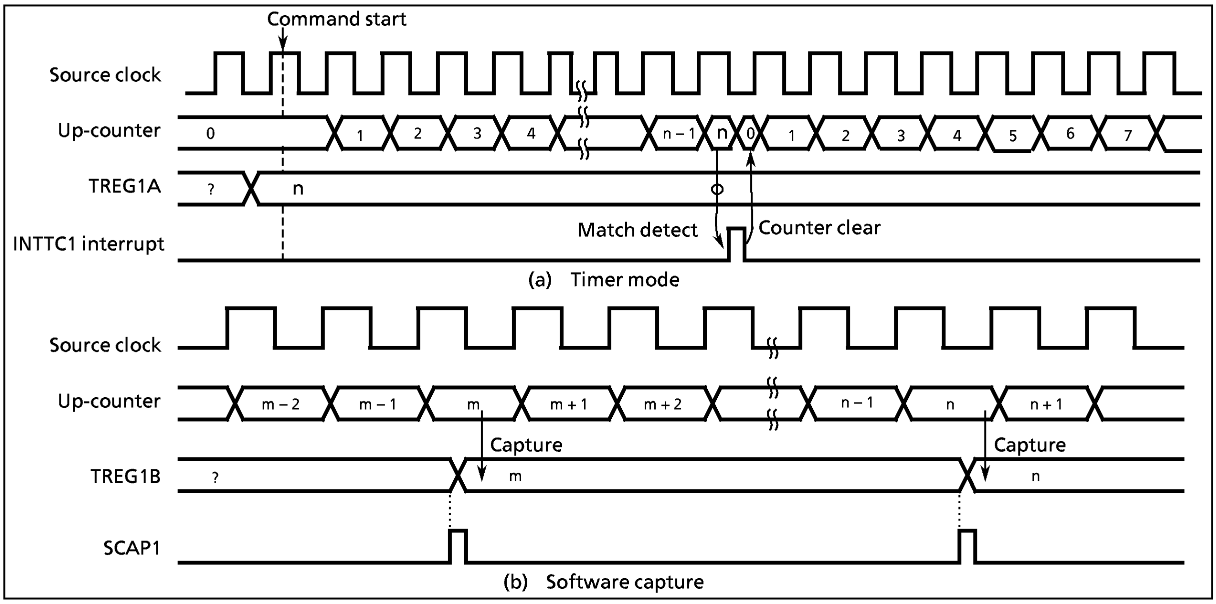


Figure 2-20. Timer Mode Timing Chart

(2) **External trigger timer** mode

In this mode, counting up is started by an external trigger. This trigger is the edge of the TC1 pin input. Either the rising or falling edge can be selected with INT2ES in EINTCR. Edge selection is the same as for INT2 pin. Source clock is an internal clock selected with TC1CK. The contents of TREG1A is compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to "0" and halted. The counter is restarted by the selected edge of the TC1 pin input.

When METT1 (bit 6 in TC1CR) is "1", inputting the edge to the reverse direction of the trigger edge to start counting clears the counter, and the counter is stopped. Inputting a constant pulse width can generate interrupts. When METT1 is "0", the reverse directive edge input is ignored. The TC1 pin input edge before a match detection is also ignored.

The TC1 pin input has the same noise rejection as the INT2 pin; therefore, pulses of 7/fc [s] or less are rejected as noise. A pulse width of 24/fc [s] or more is required for edge detection in NORMAL1, 2 or IDLE1, 2 mode. The noise rejection circuit is turned off in SLOW and SLEEP modes. But, a pulse width of one machine cycle or more is required.

Example 1: Detects rising edge in TC1 pin input and generates an interrupt 100 μs later. (at fc = 8 MHz, DV1CK = 1)

```
LD    (EINTCR), 00000000B  ;  INT2ES←0 (rising edge)
LDW   (TREG1A), 0032H      ;  100 μs ÷ 2^4 / fc = 32H
SET   (EIRL).EF4           ;  INTTC1 interrupt enable
EI
LD    (TC1CR), 00111000B   ;  TC1 external trigger start, METT1 = 0
```

Example 2: Generates an interrupt, inputting "L" level pulse (pulse width: 4 ms or more) to the TC1 pin. (at fc = 8 MHz, DV1CK = 1)

```
LD    (EINTCR), 00000100B  ;  INT2ES←1 ("L" level)
LDW   (TREG1A), 007DH      ;  4 ms ÷ 2^8 / fc = 7DH
SET   (EIRL).EF4           ;  INTTC1 interrupt enable
EI
LD    (TC1CR), 01110100B   ;  TC1 external trigger start, METT1 = 1
```

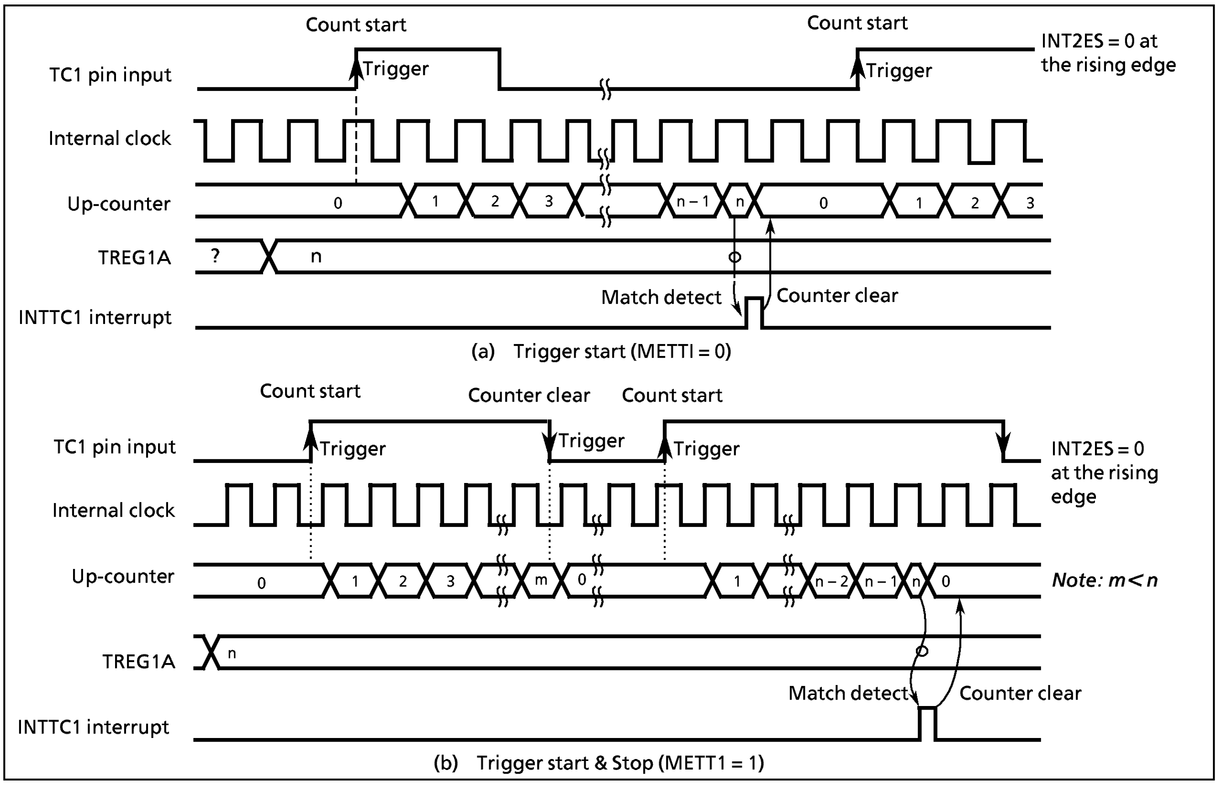


Figure 2-21. External Trigger Timer Mode Timing Chart

(3) **Event counter** mode

In this mode, events are counted at the edge of the TC1 pin input. Either the rising or falling edge can be selected with INT2ES in EINTCR. Edge selection is the same as for INT2 pin. The contents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. This maximum applied frequency is shown in table 2-5.

Setting SCAP1 to “1” transfers the current contents of up-counter to TREG1B (software capture function). SCAP1 is automatically cleared after capturing.

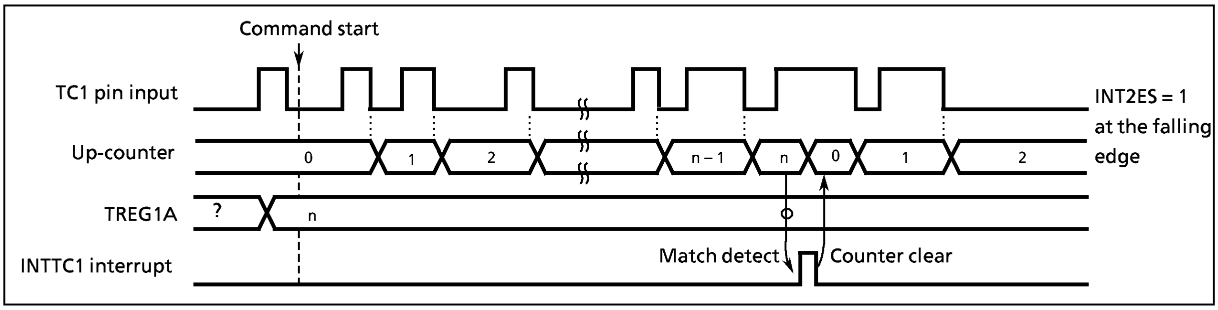


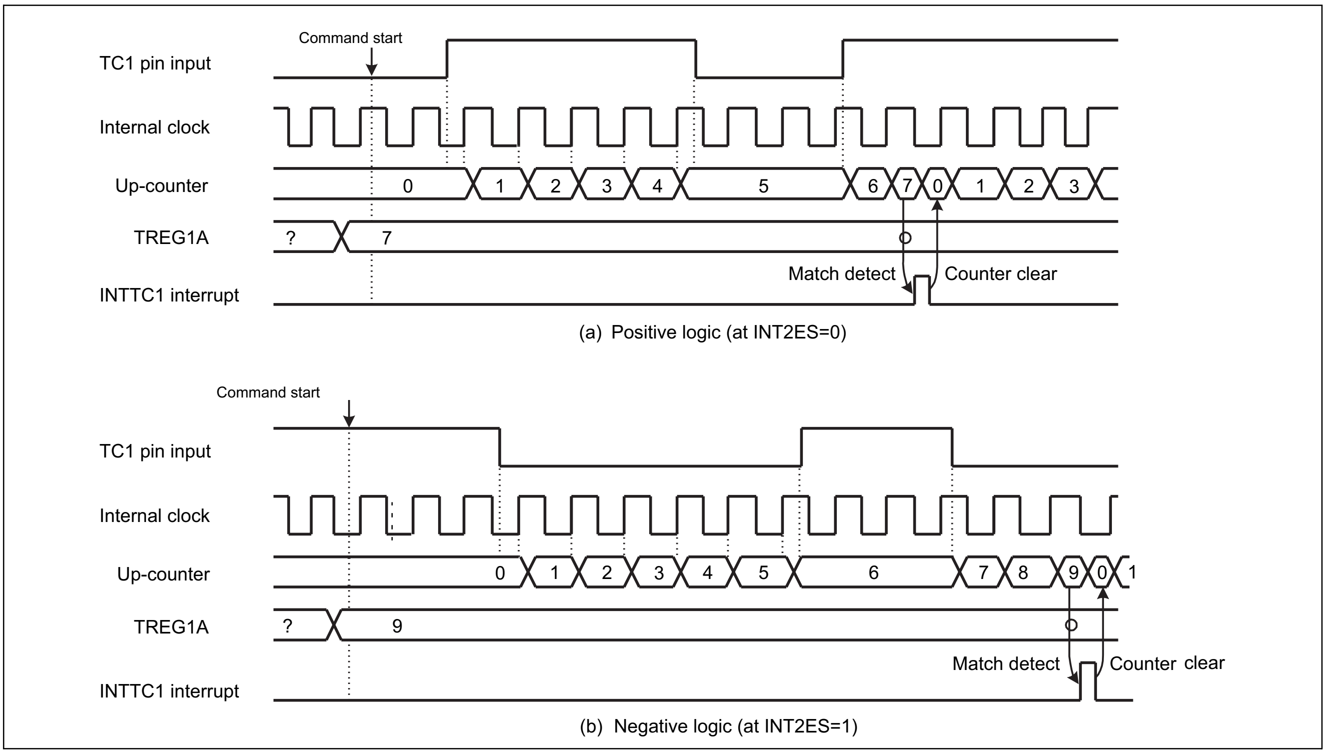
Figure 2-22. Event Counter Mode Timing Chart

Table 2-5. Timer / Counter 1 External Clock Source

Maximum applied frequency [Hz]	
NORMAL1/2, IDLE1/2 mode	SLOW, SLEEP mode
$fc/2^4$	$fs/2^4$

(4) Window mode

Counting up is performed on the rising edge of the pulse that is the logical AND-ed product of the TC1 pin input (window pulse) and an internal clock. The contents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. Positive or negative logic for the TC1 pin input can be selected. Edge selection is the same as for INT2 pin. Setting SCAP1 to "1" transfers the current contents of up-counter to TREG1B.
It is necessary that the maximum applied frequency be such that the counter value can be analyzed by the program. That is; the frequency must be considerably slower than the selected internal clock.



(a) Positive logic (at INT2ES=0)

(b) Negative logic (at INT2ES=1)

Figure 2-23. Window Mode Timing Chart

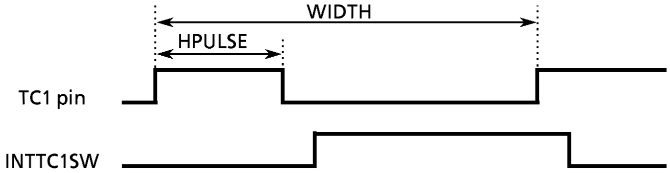
(5) Pulse width measurement mode

Counting is started by the external trigger (set to external trigger start by TC1S). The trigger can be selected either the rising or falling edge of the TC1 pin input. The source clock is used an internal clock. On the next falling (rising) edge, the counter contents are transferred to TREG1B and an INTTC1 interrupt is generated. The counter is cleared when the single edge capture mode is set. When double edge capture is set, the counter continues and, at the next rising (falling) edge, the counter contents are again transferred to TREG1B. If a falling (rising) edge capture value is required, it is necessary to read out TREG1B contents until a rising (falling) edge is detected. Falling or rising edge is selected with INT2ES, and single edge or double edge is selected with MCAP1 (bit 6 in TC1CR).

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

Example: Duty measurement (resolution fc/2^7 [Hz] DV1CK = 0)

```
                 CLR    (INTTC1SW). 0          ;  INTTC1 service switch initial setting
                 LD     (EINTCR), 00000000B    ;  Sets the rise edge at the INT2 edge
                 LD     (TC1CR), 00000110B     ;  Sets the TC1 mode and source clock
                 SET    (EIRL). EF4            ;  Enables INTTC1
                 EI
                 LD     (TC1CR), 00110110B     ;  Starts TC1 with an external trigger at MCAP1 = 0
                              ⋮
PINTTC1:    CPL    (INTTC1SW). 0          ;  Complements INTTC1 service switch
            JRS    F, SINTTC1
            LD     (HPULSE), (TREG1BL)    ;  Reads TREG1B ("H" level pulse width)
            LD     (HPULSE + 1), (TREG1BH)
            RETI
SINTTC1:    LD     (WIDTH), (TREG1BL)     ;  Reads TREG1B (Period)
            LD     (WIDTH + 1), (TREG1BH)
                     ⋮                    ;  Duty calculation
            RETI
                     ⋮
VINTTC1:    DW     PINTTC1
```



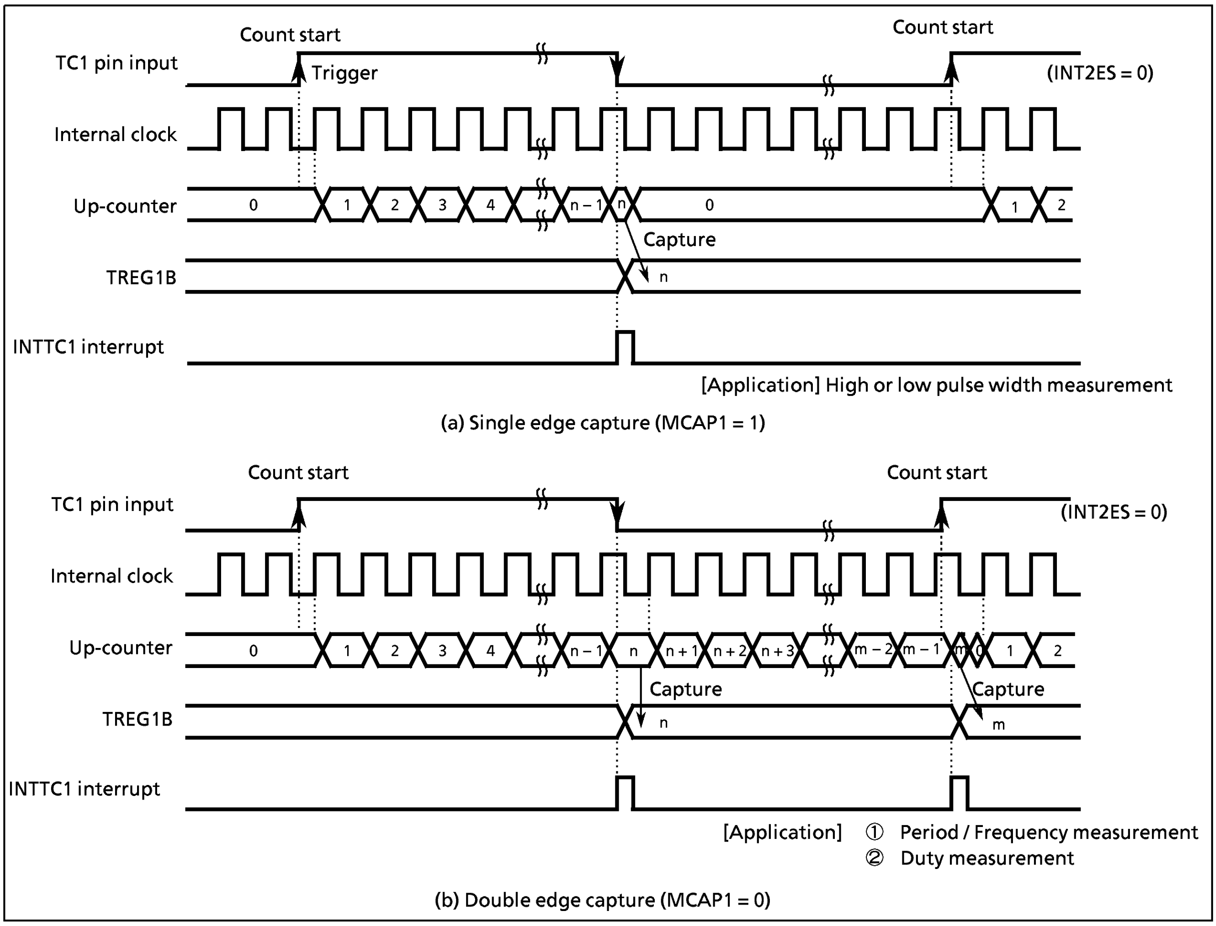


Figure 2-24. Pulse Width Measurement Mode Timing Chart

2.7 16-bit Timer / Counter 2 (TC2)

2.7.1 Configuration

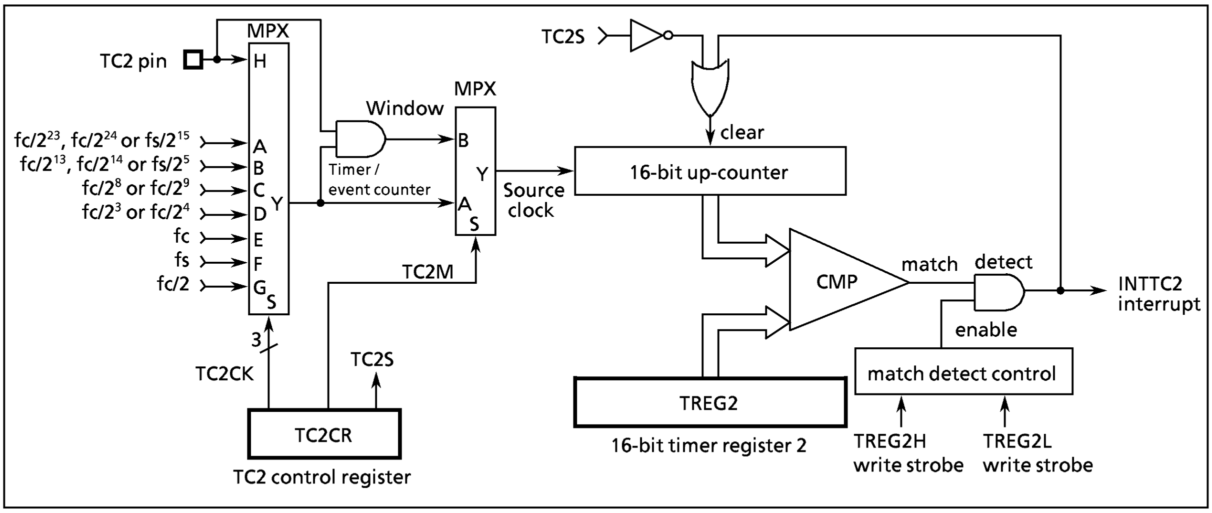


Figure 2-25. Timer / Counter 2 (TC2)

2.7.2 Control

The timer / counter 2 is controlled by a timer/counter 2 control register (TC2CR) and a 16-bit timer register 2 (TREG2). Reset does not affect TREG2.

Timer Register 2

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREG2 (00018, 00019$_H$)	TREG2$_H$ (00019$_H$)								TREG2$_L$ (00018$_H$)							

Write only

Timer Counter 2 Control Register

	7	6	5	4	3	2	1	0
TC2CR (00015$_H$)			TC2S	TC2CK				TC2M

(Initial value: **00 00*0)

TC2M	TC2 operating mode select	0: Timer / event counter mode 1: Window mode							Write only
TC2CK	TC2 source clock select [Hz]		NORMAL1/2, IDLE1/2 mode				SLOW mode	SLEEP mode	
			DV7CK = 0		DV7CK = 1				
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1			
		000	fc / 2^{23}	fc / 2^{24}	fs / 2^{15}	fs / 2^{15}	fs / 2^{15}	fs / 2^{15}	
		001	fc / 2^{13}	fc / 2^{14}	fs / 2^{5}	fs / 2^{5}	fs / 2^{5}	fs / 2^{5}	
		010	fc / 2^{8}	fc / 2^{9}	fc / 2^{8}	fc / 2^{9}	–	–	
		011	fc / 2^{3}	fc / 2^{4}	fc / 2^{3}	fc / 2^{4}	–	–	
		100	–	–	–	–	fc	–	
		101	fs	fs	fs	fs	–	–	
		110	–	–	–	–	fc / 2	–	
		111	External clock (TC2 pin input)						
TC2S	TC2 start control	0: Stop and counter clear 1: Start							

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

Note 2: Writing to the lower byte of timer register 2 (TREG2$_L$), the comparison is inhibited until the upper byte (TREG2$_H$) is written. After writing to the upper byte, any match during 1 machine cycle (instruction execution cycle) is ignored.

Note 3: Set the mode and source clock when the TC2 stops (TC2S = 0).

Note 4: Values to be loaded to the timer register must satisfy the following condition.
TREG2 > 0 (TREG2$_{15 \text{ to } 11}$ > 0 at warm-up)

Note 5: "fc" and "fc/2" can be selected as the source clock only in the timer mode during the SLOW mode.

Note 6: TC2CR and TREG2 are write-only registers and must not be used with any of the read-modify-write instructions such as SET, CLR, etc.

Figure 2-26. Timer Register 2 and TC2 Control Register

2.7.3 Function

The timer / counter 2 has three operating modes: timer, event counter and window modes. Also timer / counter 2 is used for warm-up when switching from SLOW mode to NORMAL2 mode.

(1) **Timer** mode

In this mode, the internal clock is used for counting up. The contents of TREG2 are compared with the contents of up-counter. If a match is found, a timer / counter 2 interrupt (INTTC2) is generated, and the counter is cleared. Counting up is resumed after the counter is cleared.

Also, when "fc" is selected as the source clock during SLOW mode, the lower 11 bits of TREG2 are ignored and an INTTC2 interrupt is generated by matching the upper 5 bits. Thus, in this case, only the $TREG2_H$ setting is necessary.

Table 2-6. Source Clock (internal clock) for Timer / Counter 2 (at fc = 8 MHz, fs = 32.768 kHz)

TC2CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution	Maximum time setting	Resolution	Maximum time setting	Resolution	Maximum time setting	Resolution	Maximum time setting
000	1.04 s	19.08 h	2.90 s	38.17 h	1 s	18.2 h	1 s	18.2 h
001	1.02 ms	67.10 s	2.04 ms	2.23 min	0.98 ms	1.07 min	0.98 ms	1.07 min
010	32 μs	2.09 s	64.00 μs	4.19 s	32 μs	2.09 s	64 μs	4.19 s
011	1 μs	65.53 ms	2.00 μs	131.07 ms	1 μs	65.53 ms	2 μs	131.07 ms
100	–	–	–	–	–	–	–	–
101	30.5 μs	2 s	30.5 μs	2 s	30.5 μs	2 s	30.5 μs	2 s
110	–	–	–	–	–	–	–	–

TC2CK	SLOW mode		SLEEP mode	
	Resolution [s]	Maximum time setting	Resolution [s]	Maximum time setting
000	1 s	18.2 h	1 s	18.2 h
001	0.98ms	1.07 min	0.98 ms	1.07 min
01*	–	–	–	–
100	125 ns (Note)	–	–	–
101	–	–	–	–
110	250 ns (Note)	–	–	–

*Note: fc and fc/2 can be used only in the timer mode. It is used for warm-up when switching from **SLOW** mode to **NORMAL2** mode.*

Example: Sets the timer mode with source clock $fc/2^4$ [Hz] and generates an interrupt every 25 ms (at fc = 8 MHz, DV1CK = 1).

```
LDW  (TREG2), 30D4H      ;  Sets TREG2 (25 ms ÷ 2^4 / fc = , 30D4H)
SET  (EIRH).EF8          ;  Enables INTTC2 interrupt
EI
LD (TC2CR), 00101100B    ;  Starts TC2
```

(2) **Event counter** mode

In this mode, events are counted on the rising edge of the TC2 pin input. The contents of TREG2 are compared with the contents of the up-counter. If a match is found, an INTTC2 interrupt is generated, and the counter is cleared. The maximum frequency applied to the TC2 pin is shown in table 2-7. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.

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

```
LDW   (TREG2), 640           ;  Sets TREG2
SET   (EIRH).EF8             ;  Enables INTTC2 interrupt
EI
LD    (TC2CR), 00111100B     ;  Starts TC2
```

Table 2-7. Timer / Counter 2 External Clock Source

Maximum applied frequency [Hz]	
NORMAL1/2, IDLE1/2 mode	SLOW, SLEEP mode
$fc / 2^4$	$fs / 2^4$

(3) **Window** mode

In this mode, counting up is performed on the rising edge of an internal clock during TC2 external pin input(window pulse) is "H" level. The contents of TREG2 are compared with the contents of up-counter. If a match is found, an INTTC2 interrupt is generated, and the up-counter is cleared.

The maximum applied frequency (TC2 input) must be considerably slower than the selected internal clock.

Example: Generates an interrupt, inputting "H" level pulse width of 120 ms or more.
(at fc = 8 MHz, DV1CK = 1)

```
LDW   (TREG2), 003AH         ;  Sets TREG2 (120 ms ÷ 2^14/fc = 003A_H)
SET   (EIRH).EF8             ;  Enables INTTC2 interrupt
EI
LD    (TC2CR), 00100101B     ;  Starts TC2
```

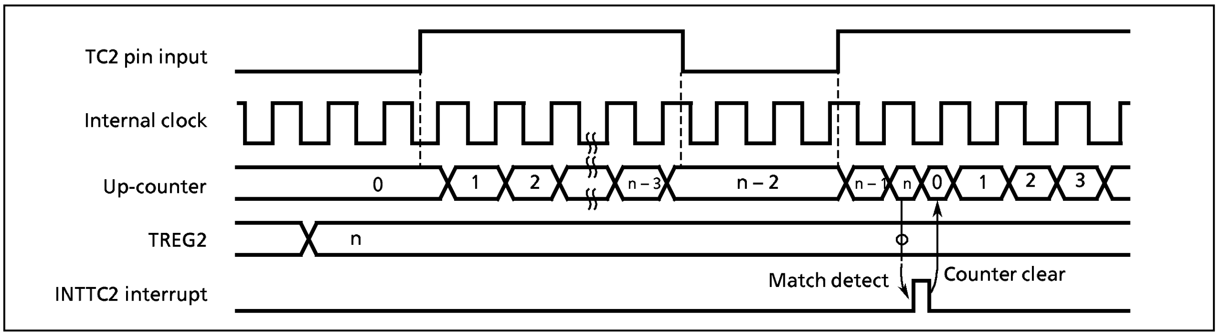


Figure 2-27. Window Mode Timing Chart

2.8 8-bit Timer / Counter 4 (TC4)

2.8.1 Configuration

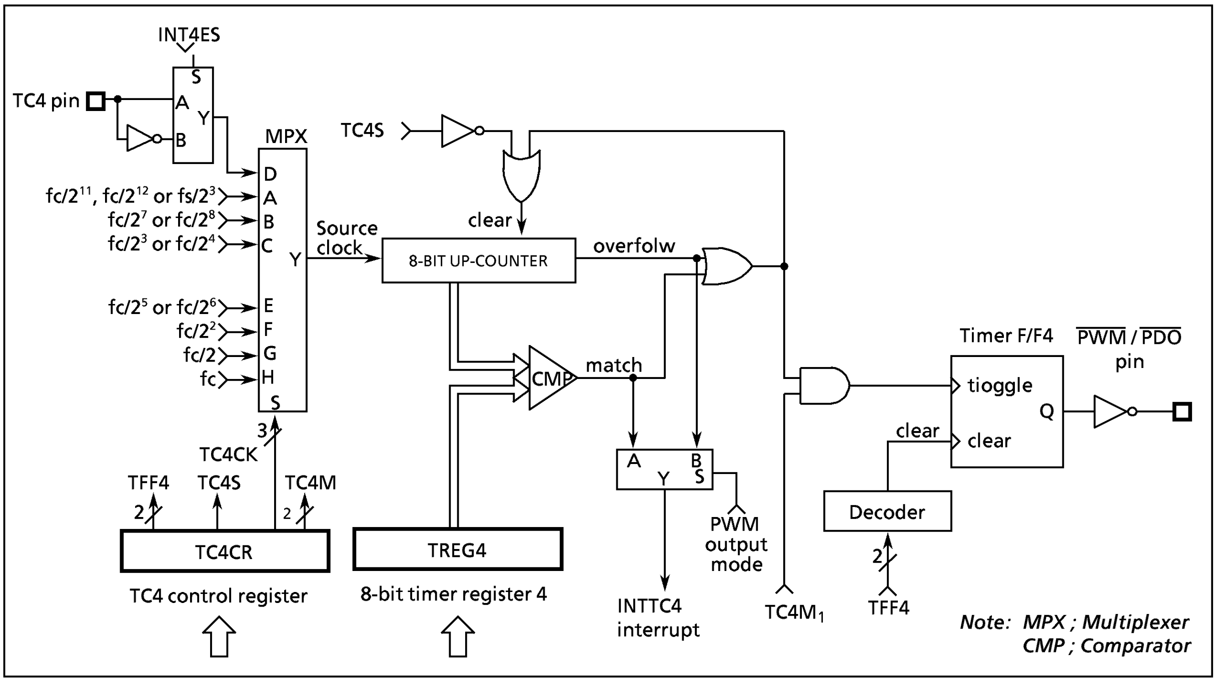


Figure 2-28. Timer / Counter 4 (TC4)

2.8.2 Control

The timer / counter 4 is controlled by a timer / counter 4 control register (TC4CR) and an 8-bit timer register 4 (TREG4). Reset does not affect TREG4.

Timer Register 4

	7	6	5	4	3	2	1	0	
TREG4 ($0001B_H$)									Write only

Timer Counter 4 Control Register

	7	6	5	4	3	2	1	0	
TC4CR ($0001C_H$)	TFF4		TC4CK	TC4S	TC4CK		TC4M		(Initial value: 0000 0000)

<table>
<tr><td>TC4M</td><td>TC4 operating mode select</td><td colspan="6">00: Timer / event counter mode
01: reserved
10: Programmable divider output (PDO) mode
11: Pulse width modulation (PWM) output mode</td><td rowspan="14">Write only</td></tr>
<tr><td rowspan="11">TC4CK</td><td rowspan="11">TC4 source clock select [Hz]
(in sequence from bit 5, 3, 2)</td><td rowspan="3"></td><td colspan="4">NORMAL1/2, IDLE1/2 mode</td><td rowspan="3">SLOW, SLEEP mode</td></tr>
<tr><td colspan="2">DV7CK = 0</td><td colspan="2">DV7CK = 1</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>000</td><td>$fc/2^{11}$</td><td>$fc/2^{12}$</td><td>$fs/2^{3}$</td><td>$fs/2^{3}$</td><td>$fs/2^{3}$</td></tr>
<tr><td>001</td><td>$fc/2^{7}$</td><td>$fc/2^{8}$</td><td>$fc/2^{7}$</td><td>$fc/2^{8}$</td><td>–</td></tr>
<tr><td>010</td><td>$fc/2^{3}$</td><td>$fc/2^{4}$</td><td>$fc/2^{3}$</td><td>$fc/2^{4}$</td><td>–</td></tr>
<tr><td>011</td><td colspan="5">External clock (TC4 pin input)</td></tr>
<tr><td>100</td><td>$fc/2^{5}$</td><td>$fc/2^{6}$</td><td>$fs/2^{5}$</td><td>$fs/2^{6}$</td><td>–</td></tr>
<tr><td>101</td><td>$fc/2^{2}$</td><td>$fc/2^{2}$</td><td>$fc/2^{2}$</td><td>$fc/2^{2}$</td><td>–</td></tr>
<tr><td>110</td><td>fc/2</td><td>fc/2</td><td>fc/2</td><td>fc/2</td><td>–</td></tr>
<tr><td>111</td><td>fc</td><td>fc</td><td>fc</td><td>fc</td><td>–</td></tr>
<tr><td>TC4S</td><td>TC4 start control</td><td colspan="6">0: Stop &amp; clear clear
1: Start</td></tr>
<tr><td>TFF4</td><td>Timer F/F4 control</td><td colspan="6">00: Clear
01: reserved
10: reserved
11: – Note 3</td></tr>
</table>

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

Note 2: Set the operating mode, the source clock selection, the edge selection (INT4ES) and timer F/F4 control when the TC4 stops (TC4S = 0)

Note 3: Set TFF4 to "11" in the timer and event counter mode and PWM mode.

Note 4: Values to be loaded to the timer register must satisfy the following condition.
(a) 5< TREG4< 251 in PWM output mode
(b) 0< TREG4 in others

Note 5: The source clock $fc/2^{2}$, fc/2 and fc must be used in only PWM output mode.

Note 6: TC4CR and TREG4 are a write-only register and must not be used with any of the read-modify-write instructions such as SET, CLR. etc.

Figure 2-29. Timer Register 4 and TC4 Control Register

2.8.3 Function

The timer / counter 4 has four operating modes: timer, event counter, programmable divider output, and PWM output mode.

(1) **Timer** mode

In this mode, the internal clock is used for counting up. The contents of TREG4 are compared with the contents of up-counter. If a match is found, an INTTC4 interrupt is generated and the up-counter is cleared to "0". Counting up resumes after the up-counter is cleared.

Table 2-8. Source Clock (internal clock) for Timer / Counter 4 (Example: at fc = 8 MHz, fs = 32.768 kHz)

TC4CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]
000	256	65.28 m	512	130.56 m	244.14	62.2 m	244.14	62.2 m
001	16	4.08 m	32	8.16 m	10.24	2.6 m	20.48	5.2 m
010	1	255 μ	2	510 μ	0.64	163.2 μ	1.28	326 μ
100	4	1.02 m	8	2.04 μ	2.56	653 μ	5.12	1306 μ

TC4CK	SLOW, SLEEP mode	
	Resolution [μs]	Maximum setting time [s]
000	244.14	62.2 m
001	—	—
010	—	—
100	—	—

(2) **Event counter** mode

In this mode, the TC4 pin input (external clock) pulse is used for counting up.

The contents of the TREG4 are compared with the contents of the up-counter. If a match is found, an INTTC4 interrupt is generated and the counter is cleared. Counting up resumes after the up-counter is cleared. The maximum applied frequency is shown in Table 2-12. Two or more machine cycles are required for both the high and low levels of the pulse width.

Table 2-9. Timer / Counter 4 External Clock Source

Maximum applied frequency [Hz]	
NORMAL1/2, IDEL1/2 mode	SLOW, SLEEP mode
$fc/2^4$	$fs/2^4$

(3) **Programmable Divider Output** (PDO) mode

The internal clock is used for counting up. The contents of TREG4 are compared with the contents of the up-counter. If a match is found, the timer F/F4 output is toggled and the counter is cleared. Timer F/F4 output is inverted and output to the P16 ($\overline{\text{PDO}}$) pin. When programmable divider output is executed, P16 output latch is set to "1" and the output mode is configured. This mode can be used for approximate 50% duty pulse output. Timer F/F4 can be initialized by program, and it is initialized to "0" during reset. An INTTC4 interrupt is generated each time the $\overline{\text{PDO}}$ output is toggled.

Example: Output a 1024 Hz pulse (at fc = 4.194304 MHz)

```
        SET    (P1).6                 ;   P16 output latch←1
        LD     (P1CR), 01000000B      ;   P16 output mode
        LD     (TREG4),  10H          ;   (1/1024 ÷ 2^7/fc) ÷ 2 = 10H
        LD     (TC4CR),  00010110B    ;   Stats TC4
```

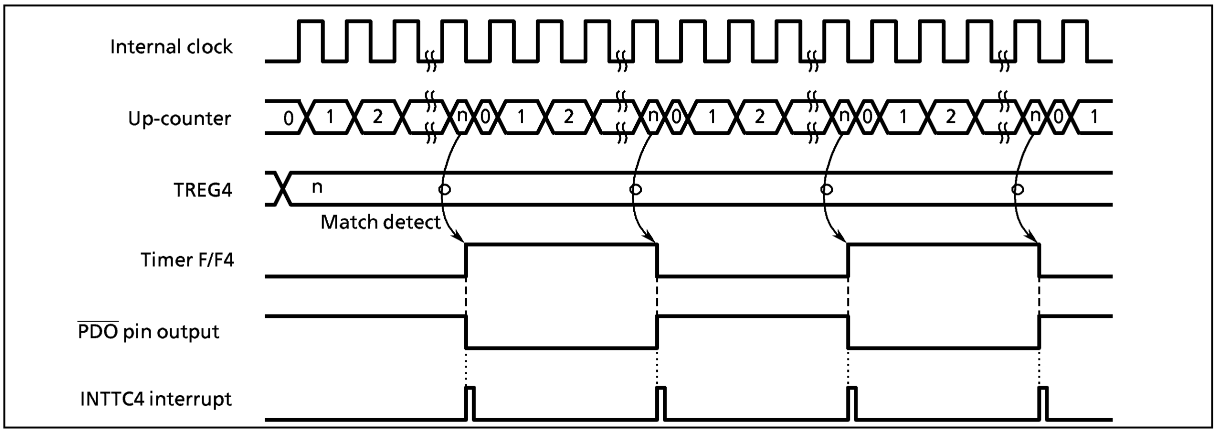


Figure 2-30. PDO Mode Timing Chart

(4) **Pulse Width Modulation** (PWM) **output** mode

PWM output with a resolution of 8 bits is possible. The internal clock is used for counting up. The contents of TREG4 are compared with the contents of up-counter. If a match is found, the timer F/F4 output is toggled. Counting up resumes. And, when an overflow occurs, the timer F/F4 output is again toggled and the counter is cleared. Timer F/F4 output is inverted and output to the P16 ($\overline{\text{PWM}}$) pin. When pulse width modulation output is executed, P16 output latch is set to "1". An INTTC4 interrupt is generated when an overflow occurs.

TREG4 is configured a 2-stage shift register and, during output, will not switch until one output cycle is completed even if TREG4 is overwritten; therefore, output can be altered continuously. Also, the first time, TREG4 is shifted by setting TC4S (bit 4 in TC4CR) to "1" after data are loaded to TREG4.

Note: Do not rewrite the contents of TREG4 at only an INTTC4 interrupt generation cycle. The contents of TREG4 is rewritten by the INTTC4 interrupt service routine.

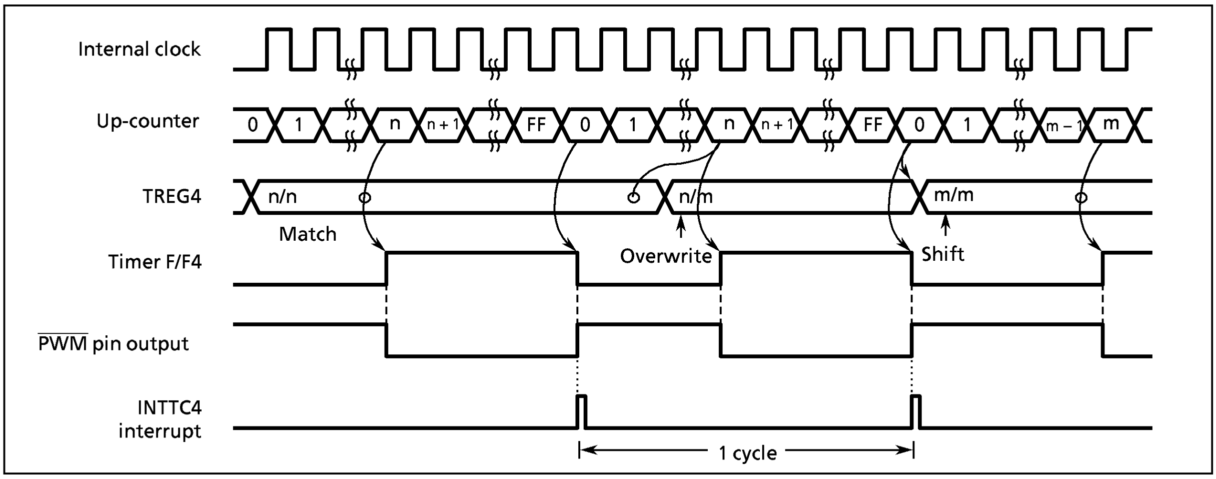


Figure 2-31. PWM Output Mode Timing Chart

Table 2-10. PWM Output Mode (Example: fc = 8 MHz, fs = 32.768 kHz)

TC4CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [ns]	Repeat cycle [μs]	Resolution [ns]	Repeat cycle [μs]	Resolution [ns]	Repeat cycle [μs]	Resolution [ns]	Repeat cycle [μs]
000	256 μs	65.5 ms	512 μs	131.0 ms	244.1 μs	62.5 ms	244.1 μs	62.5 ms
001	16 μs	4.09 ms	32 μs	8.19 ms	16 μs	4.09 ms	32 μs	8.19 ms
010	1 μs	256	2 μs	512	1 μs	256	2 μs	512
100	4 μs	1.02 μs	8 μs	2.04 ms	4 μs	1.02 μs	4 μs	1.02 μs
101	500	128	500	128	500	128	500	128
110	250	64	250	64	250	34	250	64
111	125	32	125	32	125	32	125	32

2.9 Extended Timer-Counter (ETC1)

2.9.1 Structure

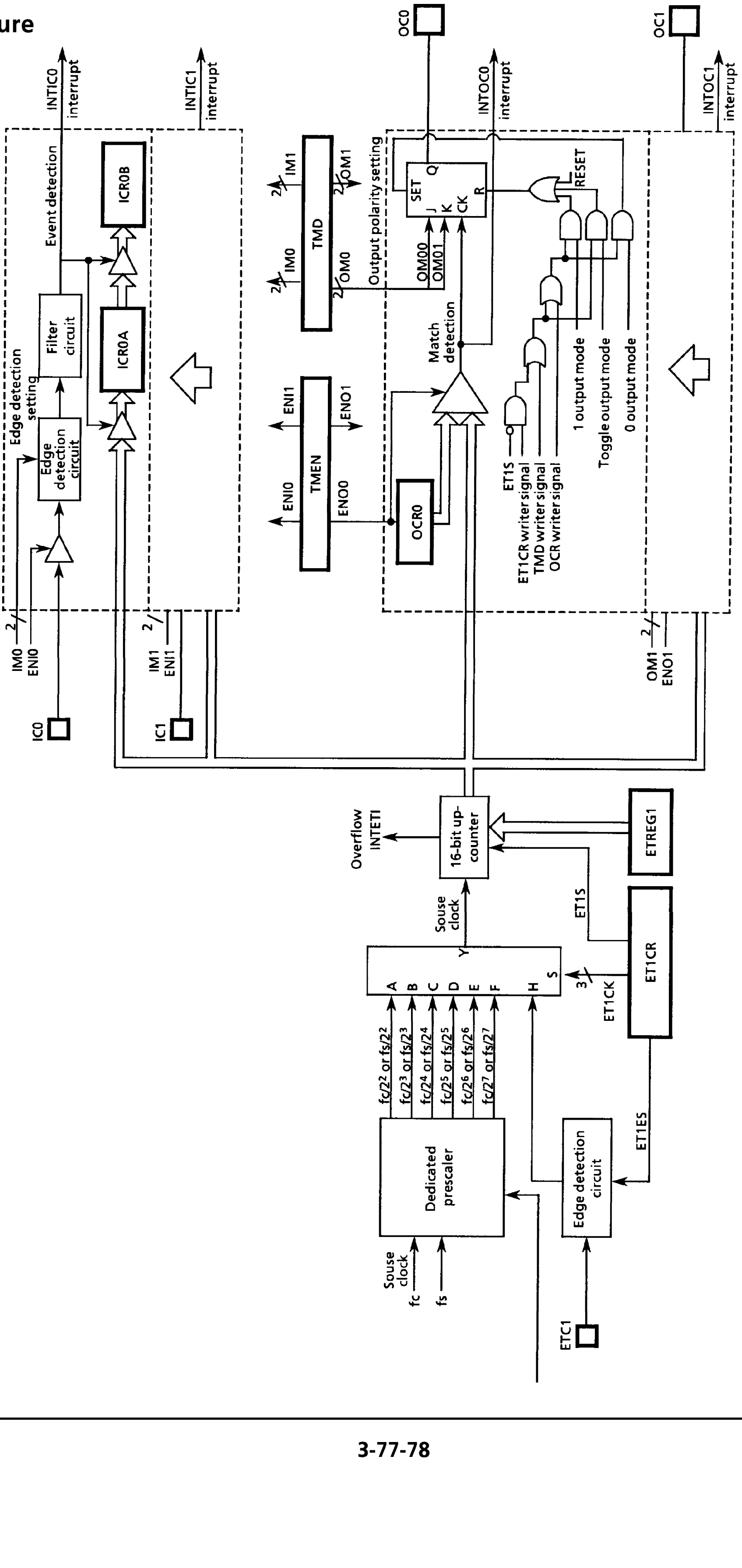


Figure 2-32. Extended Timer-Counter (ETC1)

2.9.2 Control

Extended timer-counter 1 is controlled by the following registers.
The following shows the structure of the registers. Pay special attention to Notes 1 to 4.

Extended timer registers

These 16-bit timer register set and record the ETC1 counter values.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETREG1 (00F9E, F9F$_H$)	ETREG1$_H$ (00F9F$_H$)								ETREG1$_L$ (00F9E$_H$)							

Write-only (Initial value: 0000 0000 0000 0000)

Note 1: On writing to the lower byte of the timer register, the value is not set in the counter until a write to the upper byte. Writing to the lower byte only is not possible. A one-machine cycle wait is generated after writing to the upper byte.

Note 2: ETREG1 is write-only register and must not be used with any of the read-modify-write instructions such as SET, CLR, etc.

Note 3: Writing FFFF to the timer register generates an immediate interrupt request signal.

Extended timer control registers

These register set the ETC1 operating modes.

	7	6	5	4	3	2	1	0
ET1CR (00F9B$_H$)	ET1S				ET1ES	ET1CK		

(Initial value: 0*** 0000)

ET1S	ETREG1 start control	0: Stop 1: Start command			R / W
ET1ES	Count edge selection for event counter mode	0: Rising 1: Falling			
ET1CK	ETC1 source clock selection		NORMAL1 / 2 or IDLE1 / 2 modes	SLOW or SLEEP modes	
		000	fc / 2^2	fs / 2^2	
		001	fc / 2^3	fs / 2^3	
		010	fc / 2^4	fs / 2^4	
		011	fc / 2^5	fs / 2^5	
		100	fc / 2^6	fs / 2^6	
		101	fc / 2^7	fs / 2^7	
		110	reserved	reserved	
		111	External clock (event counter mode)		

*: don't care

Timer channel enable register

This register specifies whether to use the capture input control or match detection control.

	7	6	5	4	3	2	1	0
TMEN (00F9C$_H$)	ENO1	ENO0	ENI1	ENI0				

(Initial value: 0000 ****)

ENI0	IC0 capture input control	0:Disable 1:Enable	R / W
ENI1	IC1 capture input control	0:Disable 1:Enable	
ENO0	OC0 match detection control	0:Disable 1:Enable	
ENO1	OC1 match detection control	0:Disable 1:Enable	

Note: When used as the output compare NOP mode, the ENOi should be set to "1". (i = 1, 0)

Figure 2-33. Extended Timer-Counter 1 Register (1 / 3)

Capture / Compare mode register

This register specifies the operating modes for the input capture channels.

	7	6	5	4	3	2	1	0	
TMD (00F9D$_H$)	IM1		IM0		OM1		OM0		(Initial value: 0000 0000)

IMi	ICi capture edge control	00: Detects rising edges 01: Detects falling edges 10: Detects rising or falling (either) edges 11: reserved	R / W
OMi		00: NOP, output unchanged (steady output) 01: 1 output 10: 0 output 11: Toggel output (output invreted)	

Note 1: i = 1 to 0

Note 2: When writteng to 16-bit registers, always write the lower byte first, followed by the upper byte.

Note 3: When reading from 16-bit registers, always read the lower byte first, followed by the upper byte.

Note 4: When the OMi used as NOP mode, the compare output pin can be used as normal I/O ports. The ENOi of TMEN should be set to "1" in case.

Capture registers

These 16-bit registers record the time when a capture pin input changed.

Channels IC0 and IC1 have two registers each.

Channel	Capture registers
IC0	ICR0A (00FB8, 00FB9$_H$), ICR0B (00FBA, 00FBB$_H$)
IC1	ICR1A (00FBC, 00FBD$_H$), ICR1B (00FBE, 00FBF$_H$)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICR0A (00FB8, FB9$_H$)	ICR0A$_H$ (00FB9$_H$)								ICR0A$_L$ (00FB8$_H$)							

Read only (Initial value: 0000 0000 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICR0B (00FBA, FBB$_H$)	ICR0B$_H$ (00FBB$_H$)								ICR0B$_L$ (00FBA$_H$)							

Read only (Initial value: 0000 0000 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICR1A (00FBC, FBD$_H$)	ICR1A$_H$ (00FBD$_H$)								ICR1A$_L$ (00FBC$_H$)							

Read only (Initial value: 0000 0000 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICR1B (00FBE, FBF$_H$)	ICR1B$_H$ (00FBF$_H$)								ICR1B$_L$ (00FBE$_H$)							

Read only (Initial value: 0000 0000 0000 0000)

Note: On reading the registers, the lower byte read operation latches the buffer value. Therefore, always read the lower byte first followed by the upper byte.

Figure 2-33. Extended Timer-Counter 1 Register (2 / 3)

Compare registers

These 16-bit registers set the time at which to change the compare pin output. Each channel has one register.

Channel	Compare register
OC0	OCR0 (00FB4, $00FB5_H$)
OC1	OCR1 (00FB6, $00FB7_H$)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCR0 (00FB4, $FB5_H$)	$OCR0_H$ ($00FB5_H$)								$OCR0_L$ ($00FB4_H$)							

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCR1 (00FB6, $FB7_H$)	$OCR1_H$ ($00FB7_H$)								$OCR1_L$ ($00FB6_H$)							

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

Note 1: On writing to the byte of a compare register, the data value is not set until a write to the upper byte. Writing to the lower byte only is not possible. A one-machine cycle wait is generated after writing to the upper byte.

Note 2: On reading the registers, the lower byte read operation latches the timer-counter value. Therefore, always read the lower byte first followed by the upper byte.

Figure 2-33. Extended Timer-Counter 1 Register (3 / 3)

2.9.3 Outline

TMP88CP77/S77/U77 each have three internal 16-bit timer-counter channels (TC1, TC2, ETC1). The following is an outline of the extended timer (ETC1).

(1) Free-running timer mode and event counter mode

The ETC1 timer-counter has two operating modes: free-running timer mode and event counter mode. Free-running timer mode counts up on an internal clock generated by a dedicated prescaler. Event counter mode counts up on input from the ETC1 (P02) pin.

Table 2-11. Free-Running Mode and Event Counter Mode (When fc = 8 MHz, fs = 32.768 kHz)

	Count clock	Edge	Resolution	
			(fc)	(fs)
Free-running timer	Internal clock	Rising edge	500ns to 16μs	122μs to 3.906ms
Event counter mode	ETC1 pin	Rising or falling	1μs or more	244μs or more

(2) Input capture function and output compare function

ETC1 has two input capture pin channels (P03 (IC0) and P02 (IC1)) and two output compare pin channels (P04 (OC0) and P06 (OC1)). Each channel can be used as a standard input / output port when not being used by ETC1.

Table 2-12. Input Capture Function and Output Compare Function

	Pin	Function	Resolution	Data register
Input capture mode	P03 P02	Rising edge Falling edge Either edge	Depends on specified timer	Two for P03 and P02
Output compare mode	P04 P06	Outputs 1 Outputs 0 Inverts output No change		One per channel

2.9.4 Outline of timer modes

ETC1 has two operating modes: free-running timer mode and event counter mode.

(1) Free-running timer mode

This mode counts up on an internal clock. When the counter value reaches FFFFH, an INTET1 / INTET2 interrupt is generated. This does not halt the count.

The ET1CK bit of the extended timer-counter 1 control register (ET1CR) selects the source clock. Setting ET1CK to 111B sets event counter mode.

To generate an interrupt after a specific number of counts, subtract the desired count from FFFFH and set in extended timer register 1 (ETREG1). To clear the counter, write 0000H. When writing to the ETREG1, write the lower byte first, followed by the upper byte. As the trigger for the counter to load the data value is the completion of the write to the upper byte, writing the lower byte only does not load the data value. After a write to the upper byte, a one-machine cycle wait is generated before the count starts.

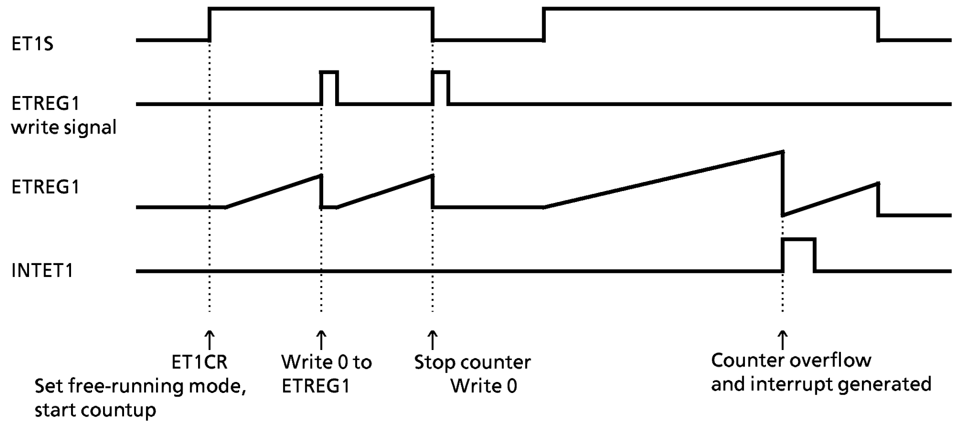
ETREG1 is write-only register and must not be used with any of the read-modify-write instructions such as SET, CLR, etc.

Example 1: Generate an ETC1 interrupt after 4s using the fs / 2^2 source clock. (fs = 32.768 kHz)

```
LD      (ET1CR), 00000000B    ; stops the counter, sets the source clock to fs / 2^2
LDW     (ETREG1), 07FFFH      ; sets the timer register (4 s ÷ 122 μs = 07FFFH)
SET     (EIRH).EF12           ; enables interrupt INTET1
EI
SET     (ET1CR).7             ; starts ETC1
```

Example 2: Generate an ETC1 interrupt after 256 μs using the fc / 2^3 source clock. (fc = 8 MHz, DV1CK = 0)

```
LD      (ET1CR), 00000001B    ; stops the counter, sets the source clock to fc / 2^3
LDW     (ETREG1), 0FEFFH      ; sets the timer register (0FFFFH to 100H)
SET     (EIRH) .EF12          ; enables interrupt INTET1
EI
SET     (ET1CR).7             ; starts ETC1
```


ET1S
ETREG1 write signal
ETREG1
INTET1
ET1CR Set free-running mode, start countup
Write 0 to ETREG1
Stop counter Write 0
Counter overflow and interrupt generated

Note: ET1S is a bit of ET1CR.

Figure 2-34. Example of Free-Running Mode Operation

Table 2-13. ETC1 Internal Clock Source (Example: When fc = 8 MHz, fs = 32.768 kHz)

ET1CK	NORMAL1 / 2 or IDLE1 / 2 modes		SLOW mode or SLEEP modes	
	Resolution	Max. setting time	Resolution	Max. setting time
000	500 ns	32.77 ms	122.0 μs	8 s
001	1 ns	65.54 ms	244.1 μs	16 s
010	2 μs	131.07 ms	488.3 μs	32 s
011	4 μs	262.14 ms	976.6 μs	64 s
100	8 μs	524.29 ms	1.953 ms	128 s
101	16 μs	1.05 s	3.906 ms	256 s
110	–	–	–	–

Note: ET1CK is the ETC1 source clock setting bit.

(2) Event counter mode

Event counter mode counts up on an edge input to the ETC1 pin. ET1CK = 111B sets event counter mode. When used as ETC1 (P02), the P02 latch must be set to "1".

This mode functions the same as free-running mode, except that the source clock changes from an internal clock to a pin input.

The maximum applied frequency is $fc/2^3$ [Hz] in NORMAL 1/2 or IDLE 1/2 mode and $fs/2^3$ [Hz] in SLOW or SLEEP mode. Therefore, the external clock of two machine cycle or less is not guaranteed.

Example 1: Generate an interrupt after counting 100 (64H) rising edges on the ETC1 pin.

```
SET     (P0 write). 2         ; sets P02 (ETC1) output latch to "1"
LD      (ET1CR), 00000111B    ; stops the count, sets rising edge, sets ungated
                                event counter mode
LDW     (ETREG1), 0FF9BH      ; sets the timer register (0FFFFH to 0064H)
SET     (EIRH).EF12           ; enables INTET1 interrupt
EI
SET     (ET1CR).7             ; starts ETC1
```

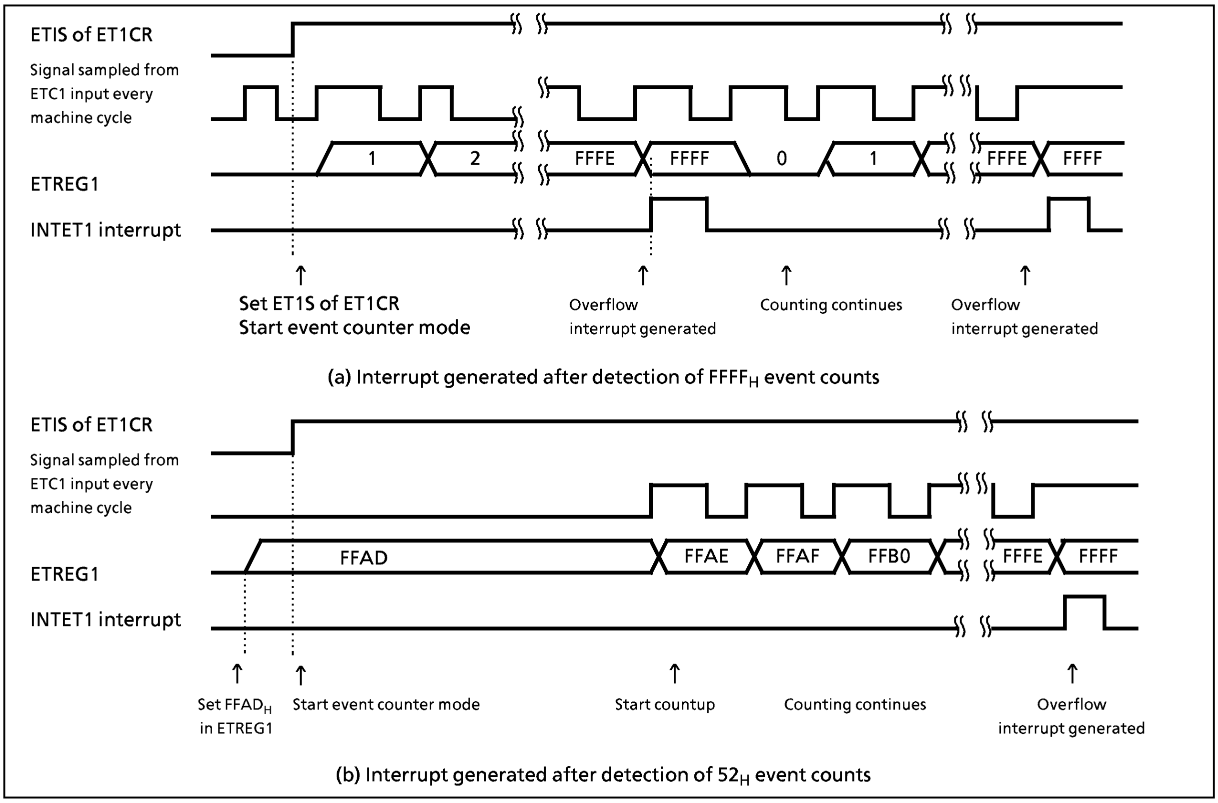
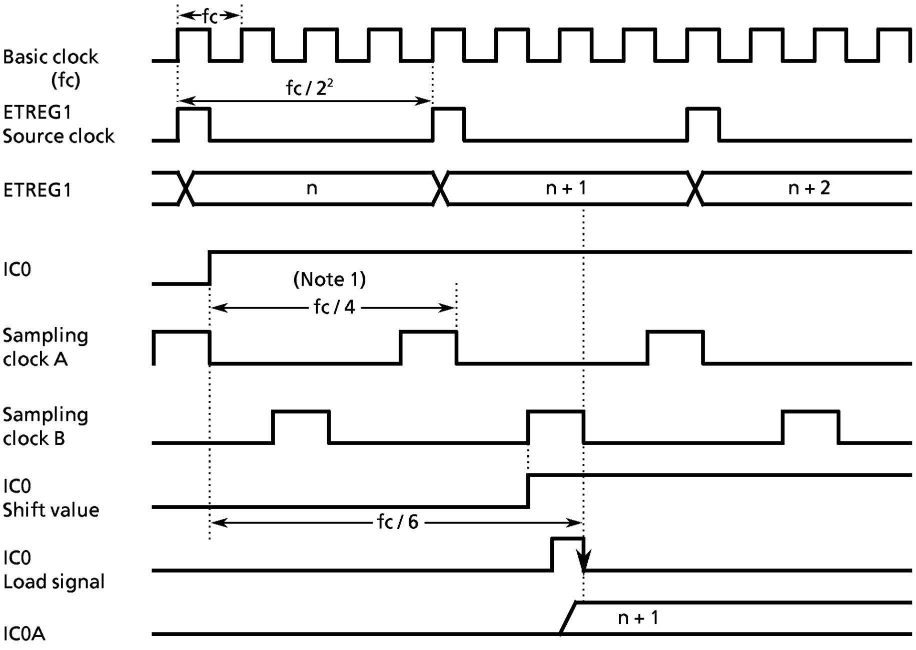

ETIS of ET1CR
Signal sampled from ETC1 input every machine cycle
ETREG1
INTET1 interrupt
1
2
FFFE
FFFF
0
1
FFFE
FFFF
Set ET1S of ET1CR
Start event counter mode
Overflow interrupt generated
Counting continues
Overflow interrupt generated
(a) Interrupt generated after detection of FFFFH event counts
ETIS of ET1CR
Signal sampled from ETC1 input every machine cycle
ETREG1
INTET1 interrupt
FFAD
FFAE
FFAF
FFB0
FFFE
FFFF
Set FFADH in ETREG1
Start event counter mode
Start countup
Counting continues
Overflow interrupt generated
(b) Interrupt generated after detection of 52H event counts

Figure 2-35. Example of Event Counter Mode Operation (ETC1 Rising Edge Detection)

2.9.5 Pin Functions

(1) Channel pin
ETC1 has two input capture (IC) pin channels (P03 (IC0) and P02 (IC1)), and two output compare (OC) pin channels (P04 (OC0) and P06 (OC1)).
Use the timer channel enable register (TMEN) to select whether to use a pin as an ETC channel pin or as a standard I/O port.
When using a pin as an ETC channel pin, be sure to set the output data latch in P0 write.
For example, when using P04 as OC0, set P0 write (P0 write, bit 4) to 1.

(2) Input capture function
This function can be used to measure such parameters as pulse width, period, and duty.
The function loads the timer value into the capture register at the capture input change timing.
The capture input circuit has an internal digital noise canceller circuit. Inputs shorter than a maximum of fc / 6 may be interpreted as noise and not be accepted as an input signal. Therefore, the width of the capture input must always be longer than fc / 6.
Since this mode loads the timer value to the capture register after the capture input is sampled by the fc / 4 sampling clock, a detection delay of between fc / 2 (min.) and fc / 6 (max.) occurs.
The timer value is loaded into capture register iA (ICRiA) when the first edge is detected and an interrupt generated. Further capture operations are disabled until ICRiA is read. Always read the lower byte of the ICRiA register first, followed by the upper byte.
IC0, IC1 shift the ICRiA value (at previous edge detection) to ICRiB when the next edge is detected, load the timer value (at current edge detection) to ICRiA, and generate an interrupt.
Since the timer continues to count up if an overflow occurs before edge detection, the capture operation proceeds as normal. *Note: i = 0, 1*



Note: The IC input sampling clock is fc / 4 regardless of the source clock.

Figure 2-36. Example of Capture Function Sampling
(Capture 0 Rising Edge Detection, ETREG1 Source Clock = fc / 2^2)

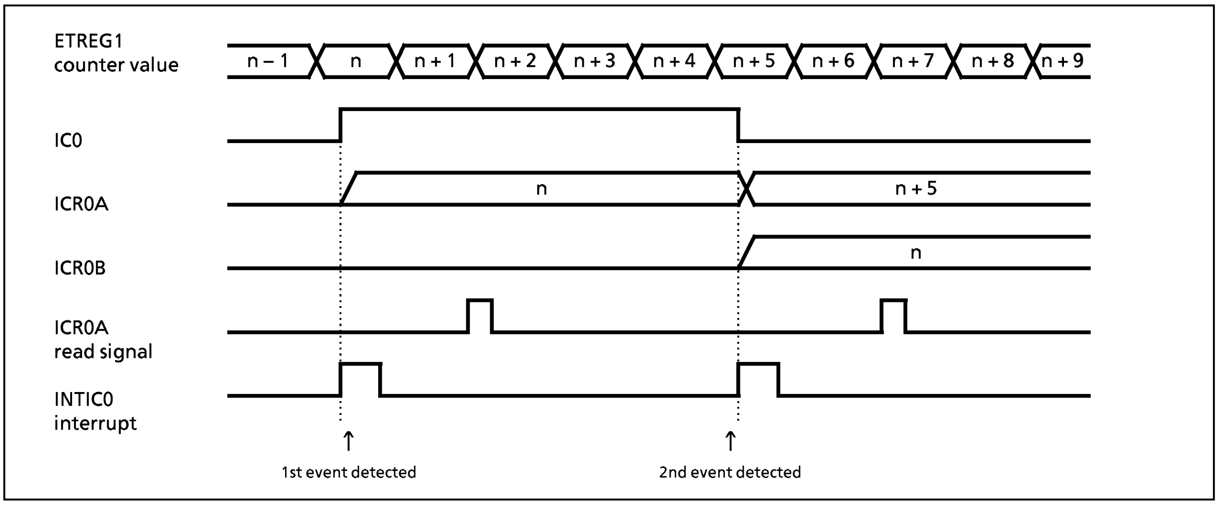


Figure 2-37. Example of Capture Mode Operation (Channel 0 Either-Edge Detection)

```
Example 1: Detect rising or falling edges on IC0.
                SET     (P0 write). 3           ;  sets P03 output data latch to “1”
                LD      (TMD), 20H              ;  sets IC0 to either-edge detection
                SET     (TMEN). 4               ;  sets P03 to capture mode (IC0)
Example 2:  Measure the width of positive pulses on IC1.
                SET     (P0 write). 2           ;  sets P02 to input
                LD      (TMD), 00H              ;  sets IC1 to rising edge detection
   LP:          TEST    ILE.1                   ;  waits for IC1 rising edge detection (sets IL17)
                JR      T, LP
                LD      (ILE), FDH              ;  clears IL17
                LD      L, (ICR1AL)             ; reads ICR1A
                LD      H, (ICR1AH)
                LD      (TMD), 40H              ;  sets IC1 to falling edge detection
   LP2:         TEST    ILE.1                   ;  waits for IC1 falling edge detection (sets IL17)
                JR      T, LP2
                LD      A, (ICR1AL)
                LD      W, (ICR1AH)
                SUB     WA, HL                  ;  calculates pulse width, stores in WA register
                LD      (ILE), 0FDH             ;  clears IL17
```

(3) Output compare mode

This mode can set the output to any desired level at any desired timing. Bit OMi of the compare mode register (TMD) selects the output mode. The modes are 1 output, 0 output, toggle output, and NOP.

The compare register (OCRi) is used to set the output change timing. Always write to the lower byte of OCRi first, followed by the upper byte. Since the trigger for OCRi to load the data value is the completion of the write to the upper byte, writing the lower byte only does not load the data value.

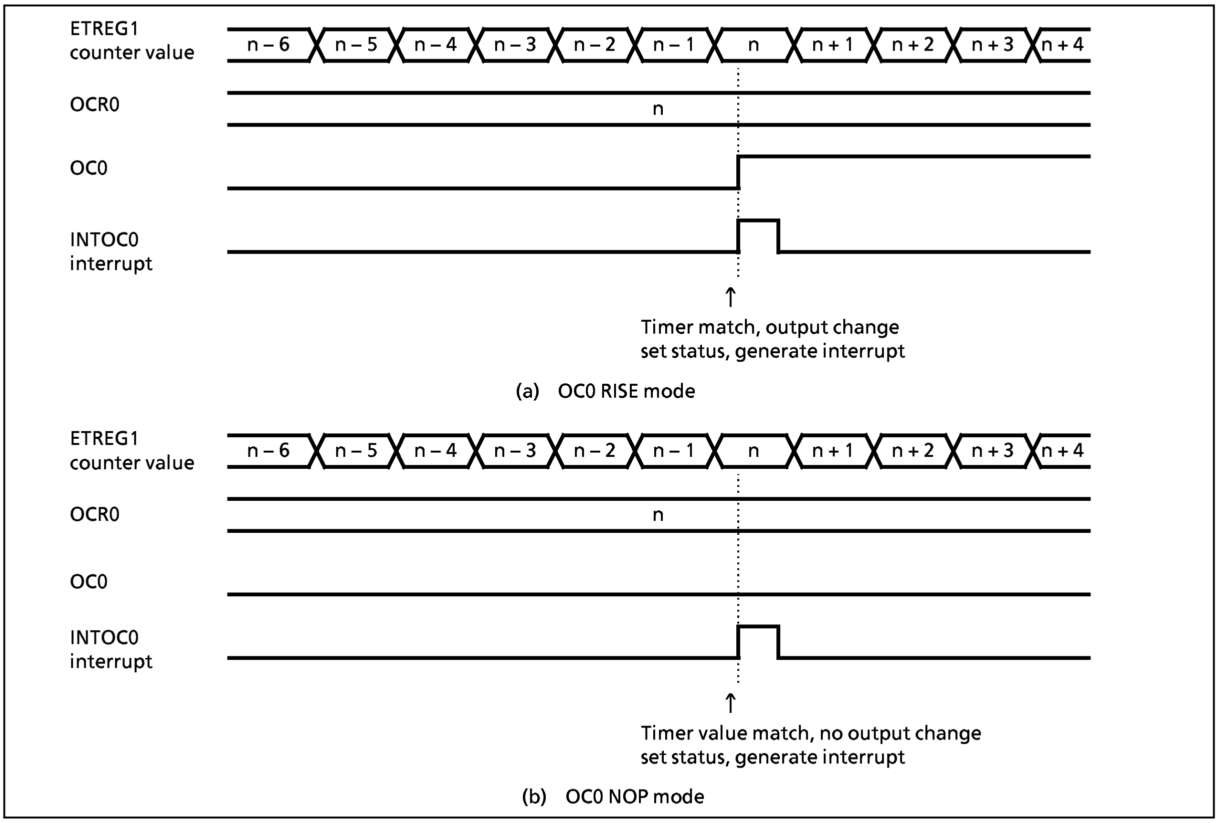
A match between the values of the selected timer and OCRi outputs the mode set in OMi to P0k.

The timer generates an interrupt request (INTOCi) after performing the output change operation. Further output compare operations are disabled until writing to the upper byte of OCRi completes. However, writing 0 to ET1S enables another compare operation.

If the output mode is NOP, an interrupt only is generated when the set time is reached and the output value does not change. This can be used as a time-set interrupt with no change in the output.

When OMi = 11, the output value inverts.

Note: i = 0 , 1
k = 6, 4



Timer match, output change
set status, generate interrupt

(a) OC0 RISE mode

Timer value match, no output change
set status, generate interrupt

(b) OC0 NOP mode

Figure 2-38. Example of Compare Mode Operation

Example 1: Change the output of OC0 from 0 to 1 when ETREG1 = 0C80H (1 output mode).

```
            SET     (P0 write). 4           ; sets P04 output data latch to "1"
            LD      (TMD), 01H              ; sets OC0 to 1 output
            LD      (OCR0), 0C80H           ; sets output change time
            LD      (TMEN), 40H             ; sets P04 to compare mode (OC0)
```

Example 2: Change OC0 output from 0 to 1 when ETREG1 = 0C80H (toggle output mode).

```
            SET     (P0 write). 4           ; sets P04 output data latch to "1"
            LD      (TMD), 03H              ; sets OC0 to toggle output
            LD      (OCR0), 0C80H           ; sets output change timing
            LD      (TMEN), 40H             ; sets P04 to compare mode (OC0)
```

Example 3: Output pulses from OC0 with a 0 width of 100μs and a 1 width of 200 μs.
(fc = 8 MHz, DV1CK = 0)

```
            LD      (ET1CR), 02H            ; sets source clock to fc / 2^3 (1 μs)
            LDW     (ETREG1), 0000H
            LDW     BC, 064H                ; sets 0 width setting time (100 μs)
            LDW     DE, 0C8H                ; sets 1 width setting time (200 μs)
            LD      WA, 064H
  ;
            SET     (P0 write). 4           ; sets P04 output data latch to "1"
            LD      (TMD), 03H              ; sets OC0 to toggle output
            LD      (OCR0), BC              ; sets 0-to-1 output change time
            LD      (TMEN),  40H            ; sets P04 to compare mode (OC0)
            SET     (ET1CR). 7              ; starts ETC1
  0OUT:
            ADD     WA, DE                  ; sets 1-to-0 output change time
  LP:       TEST    ILE.2                   ; waits for OC0 1-to-0 change (sets IL18)
            JR      T, LP
            LD      (ILE), 0FBH             ; clears IL18
            LD      (OCR0L), A              ; sets 1-to-0 output change time
            LD      (OCR0H), W
  1OUT:
            ADD     WA, BC                  ; sets 0-to-1 output change time
  LP2:      TEST    ILE.2                   ; waits for OC0 0-to-1 change (sets IL18)
            JR      T, LP
            LD      (ILE), 0FBH             ; clears IL18
            LD      (OCR0L), A              ; sets 0-to-1 output change time
            LD      (OCR0H), W
            JP      0OUT
```

2.9.6 Interrupts

The timer block has a total of five interrupt requests: one extended timer interrupts (INTET1), two input capture pin interrupts (INTIC0 and INTIC1), and two output compare pin interrupts (INTOC0 and INTOC1).

(1) INTET1
Generated on counter overflow. The counter continues counting up after the overflow.

(2) INTIC0 and INTIC1
Input capture pin interrupts
The interrupt is generated when the timer value is loaded to the capture register at edge detection.

(3) INTOC0 and INTOC1
Output compare pin interrupts
The interrupt is generated at a match between the compare register and timer value.

2.10 Serial Bus Interface (SBI-ver.C)

The 88CP77/S77/U77 each have a 1-channel serial bus interface which employs a clocked-synchronous 8-bit serial bus interface and an I^{2}C bus (a bus system by Philips).

The serial bus interface is connected to an external device through P00 (SDA) and P01 (SCL) in the I^{2}C bus mode; and through P02 ($\overline{\text{SCK3}}$), P00 (SO3) and P01 (SI3) in the clocked-synchronous 8-bit SIO mode.

The serial bus interface pins are also used for the P0 port. When used for serial bus interface pins, set the P0 output latches of these pins to "1", and control inputs and outputs of these pins by the I/O control register. When not used for serial bus interface pins, the pin is used as a normal I/O port.

2.10.1 Configuration

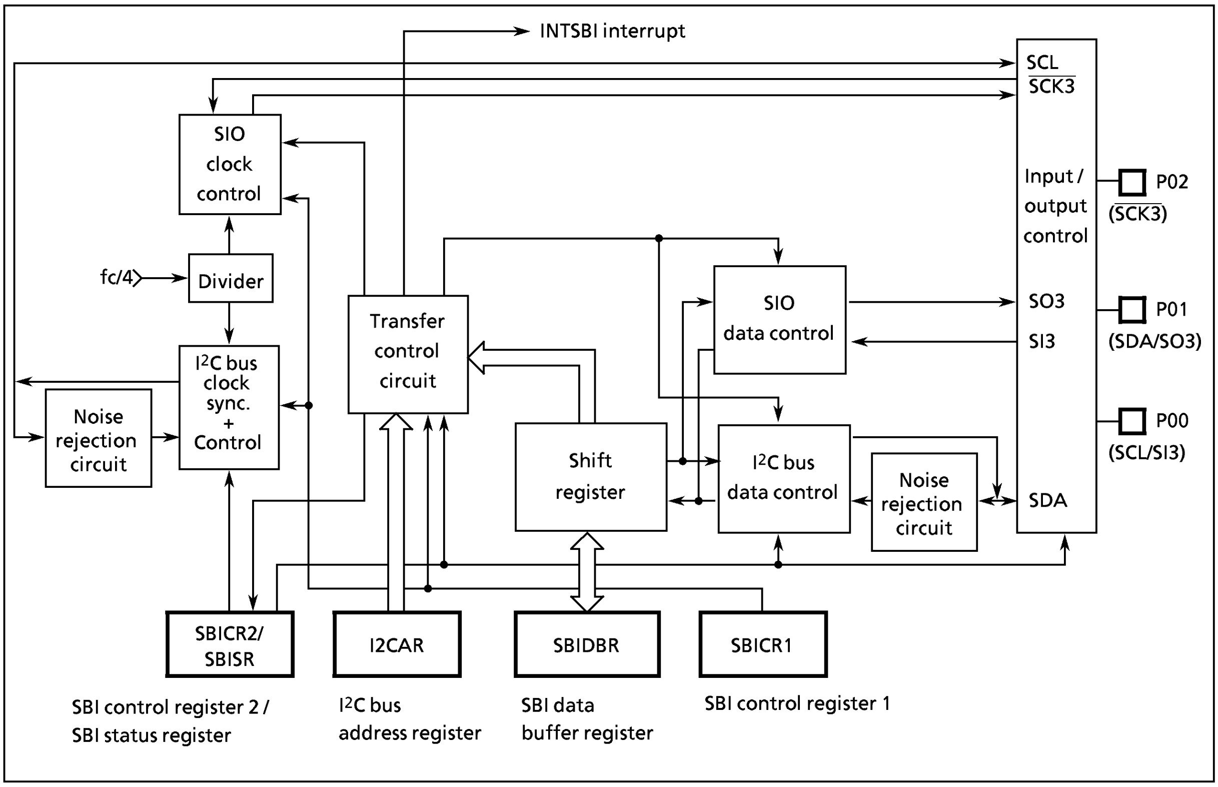


Figure 2-39. Serial Bus Interface (SBI-ver.C)

2.10.2 Control

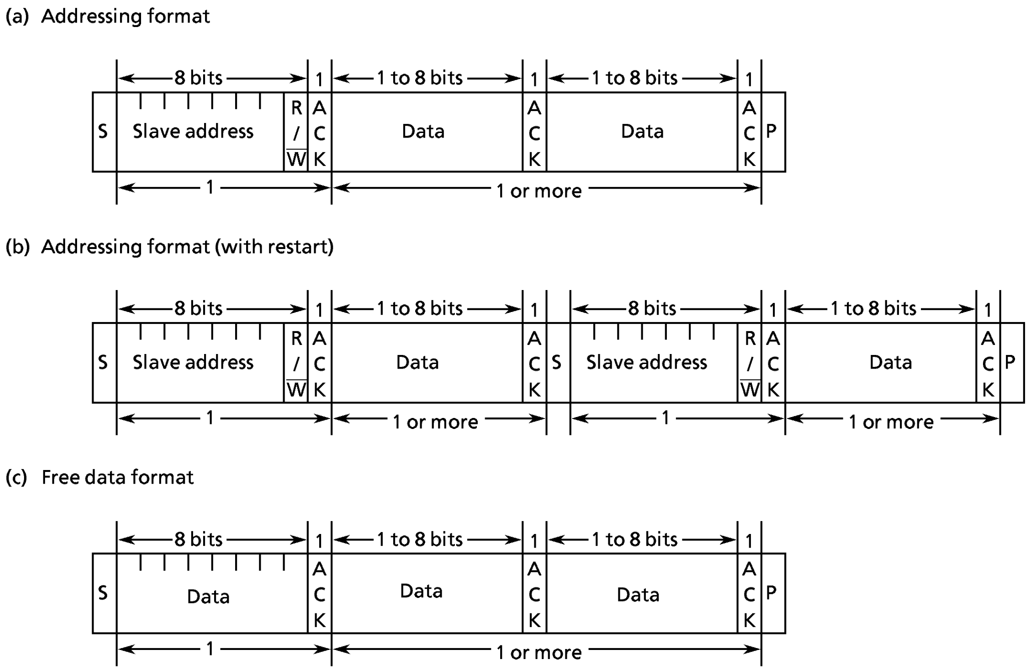
The following registers are used to control the serial bus interface and monitor the operation status.

- Serial bus interface control register 1 (SBICR1)
- Serial bus interface control register 2 (SBICR2)
- Serial bus interface data buffer register (SBIDBR)
- I^{2}C bus address register (I2CAR)
- Serial bus interface status register (SBISR)

The above registers differ depending on an mode to be used. Refer to Section "2.10.4 I^{2}C bus mode control" and "2.10.6 Clocked-synchronous 8-bit SIO mode control".

2.10.3 The data format in the I^2C bus mode

The data format in the I^2C bus mode are shown in Figure 2-40.



Notes: *S: Start condition*

R/$\overline{W}$: Direction bit

ACK: Acknowledge bit

P: Stop condition

Figure 2-40. Data Format in I^2C Bus Mode

2.10.4 I^{2}C bus mode control

The following registers are used to control the serial bus interface (SBI-ver.C) and monitor the operation status in the I^{2}C bus mode.

Serial Bus Interface Control Register 1

SBICR1	7	6	5	4	3	2	1	0	
(00020H)	BC			ACK	SWRST	SCK			(Initial value: 0000 0000)

		BC	ACK = 0		ACK = 1		
			Number of Clock	Bits	Number of Clock	Bits	
BC	Number of transferred bits	000	8	8	9	8	Write only
		001	1	1	2	1	
		010	2	2	3	2	
		011	3	3	4	3	
		100	4	4	5	4	
		101	5	5	6	5	
		110	6	6	7	6	
		111	7	7	8	7	
ACK	Acknowledge mode specification	0: Acknowledge not returned to transmitter. 1: Acknowledge returned to transmitter.					Read/ Write
SWRST	Initiate a internal of SBI	0: - 1: Initialized (Clearing “0” after initialized)					Read/ Write
SCK	Serial clock selection	000:Reserved (Note4) 001:Reserved (Note4) 010: 58.8 kHz 011: 30.3 kHz 100: 15.4 kHz 101: 7.75 kHz 110: 3.89 kHz 111:Reserved	at fc = 8 MHz (Output on SCL pin)				Write only

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

Note 2: Set the BC to “000” before switching to 8-bit SIO bus mode.

Note 3: SBICR1 is write-only registers, which cannot be used with any of read-modify-write instruction such as bit manipulation, etc.

Note 4: This I^{2}C bus circuit does not support the Fast mode. It supports the Standard mode only. Although the I^{2}C bus circuit itself allows the setting of a baud rate over 100 kbps, the compliance with the I^{2}C specification is not guaranteed in that case.

Serial Bus Interface Data Buffer Register

SBIDBR	7	6	5	4	3	2	1	0	
(00021H)									Read / Write

Note 1: For writing transmitted data, start from the MSB (bit 7).

Note 2: Cannot read the data which was written into SBIDBR, since a write data buffer and a read data buffer are independent in SBIDBR. Therefore, SBIDBR cannot be used with any of read-modify-write instructions such as bit manipulation, etc.

Note 3: The data which was written into SBIDBR is cleared to “0” when INTSBI is generated.

I^{2}C bus Address Register

I2CAR	7	6	5	4	3	2	1	0	
	Slave address							ALS	
(00022H)	SA6	SA5	SA4	SA3	SA2	SA1	SA0		(Initial value: 0000 0000)

SA	88CP77/S77/U77 slave address selection		Write only
ALS	Address recognition mode specification	0: Slave address recognition 1: Non slave address recognition	

Note: I2CAR is write-only register, which cannot be used with any of read-modify-write instruction such as bit manipulation, etc.

Figure 2-41.Serial Bus Interface Control Register 1, Serial Bus Interface Data Buffer Register and I^{2}C Bus Address Register In The I^{2}C Bus Mode

Serial Bus Interface Control Register 2

SBICR2
(00023$_H$)

7	6	5	4	3	2	1	0
MST	TRX	BB	PIN	SBIM		"0"	"0"

(Initial value: 0001 00**)

MST	Master / slave selection	0: Slave 1: Master	Write only
TRX	Transmitter / receiver selection	0: Receiver 1: Transmitter	
BB	Start / stop generation	0: Stop condition 1: Start condition	
PIN	Cancel interrupt service request	0: – (cannot be cleared to "0") 1: Cancel interrupt service request	
SBIM	Serial bus interface operating mode selection	00: Port mode (serial bus interface output disable) 01: Clocked-synchronous 8-bit SIO mode 10: I^{2}C bus mode 11: Reserved	

*Note 1: * ; don't care*

Note 2: Switch a mode to port after confirming that the bus is free.

Note 3: Switch a mode to I2Cbus mode after confiming that input signals via port are high level.

Note 4: SBICR2 has write-only register bits, which can not be used with any of read-modify-write instructions such as bit manipulation, etc.

Note 5: Clear bits 1 and 0 in SBICR2 to "0".

SBISR
(00023$_H$)

7	6	5	4	3	2	1	0
MST	TRX	BB	PIN	AL	AAS	AD0	LRB

(Initial value: 0001 0000)

MST	Master / slave selection status monitor	0: Slave 1: Master	Read only
TRX	Transmitter / receiver selection status monitor	0: Receiver 1: Transmitter	
BB	Bus status monitor	0: Bus free 1: Bus busy	
PIN	Interrupt service request status monitor	0: INTSBI occurs 1: INTSBI not occurs	
AL	Arbitration loss detection monitor	0: Arbitration loss undetected 1: Arbitration loss detected	
AAS	Slave address match detection monitor	0: Slave address unmatch or "GENERAL CALL" undetected 1: Slave address match or "GENERAL CALL" detected	
AD0	"GENERAL CALL" detection monitor	0: "GENERAL CALL" undetected 1: "GENERAL CALL" detected	
LRB	Last received bit monitor	0: Last received bit "0" 1: Last received bit "1"	

Figure 2-42. Serial Bus Interface Control Register 2 and Serial Bus Interface Status Register In The I^{2}C Bus Mode

(1) Acknowledge mode specification

Set the ACK (bit 4 in SBICR1) to “1” for operation in the acknowledge mode. The 88CP77/S77/U77 generates an additional clock pulse for an acknowledge signal when operating in the master mode. In the transmitter mode during the clock pulse cycle, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low level in order to generate the acknowledge signal.
Reset the ACK for operation in the non-acknowledge mode. The 88CP77/S77/U77 do not generate a clock pulse for the acknowledge signal when operating in the master mode.
In the acknowledge mode, the 88CP77/S77/U77 counts a clock pulse for the acknowledge signal when operating in the slave mode. During the clock pulse, when the received slave address is the same as the value set at the I2CAR or when a GENERAL CALL is received, the SDA pin is set to the low level in order to generate the acknowledge signal.
In the transmitter mode during the clock pulse cycle after matching the slave addresses or receiving a GENERAL CALL, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low level in order to generate the acknowledge signal.
In non-acknowledge mode, the 88CP77/S77/U77 does not count a clock pulse for the acknowledge signal when operating in the slave mode.

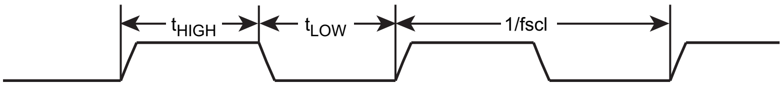
(2) Number of transfer bits

The BC (bits 7 to 5 in SBICR1) is used to select a number of bits for transmitting and receiving data. Since the BC is cleared to “000” as a start condition, a slave address and direction bit transmissions are always executed in 8 bits. Other than these, the BC retains a specified value.

(3) Serial clock

a. Clock source

The SCK (bits 2 to 0 in SBICR1) is used to select a maximum transfer frequency output from the SCL pin in the master mode. Set a communication baud rate that meets the I^{2}C bus specification, such as the shortest pulse width of tLOW, based on the equations shown below.In both master mode and slave mode, a pulse width of at least 4 machine cycles is required for both high and low levels.



$t_{LOW} = 2^n/fc$
$t_{HIGH} = 2^n/fc + 8/fc$
$fscl = 1/(t_{LOW} + t_{HIGH})$

Note: fc ; High-frequency clock

SCK (bits 2 to 0 in the SBICR1)	n
000	4
001	5
010	6
011	7
100	8
101	9
110	10

Figure 2-43. Clock Source

b. Clock synchronization

In the I2C bus mode, in order to drive a bus with a wired AND, a master device which pulls down a clock pulse to low will, in the first place, invalidate a clock pulse of another master device which generates a high-level clock pulse. The master device with a high-level clock pulse needs to detect the situation and implement the following procedure.

The 88CP77/S77/U77 have a clock synchronization function for normal data transfer even when more than one master exists on a bus.

The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

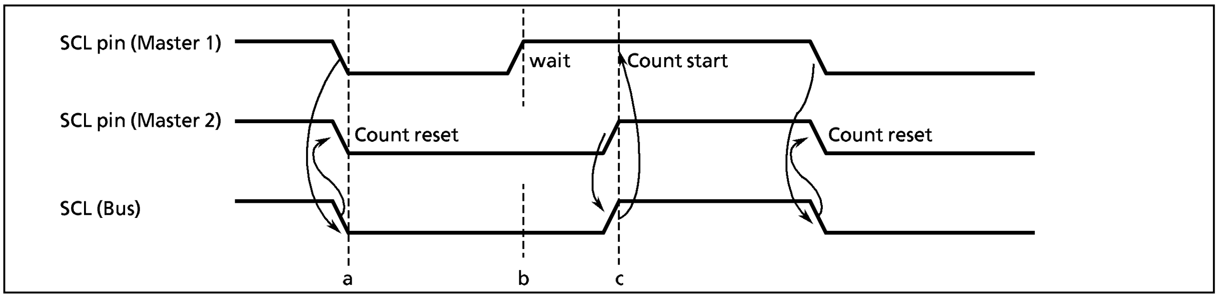


Figure 2-44. Clock Synchronization

As Master 1 pulls down the SCL pin to the low level at point “a”, the SCL line of the bus becomes the low level. After detecting this situation, Master 2 resets counting a clock pulse in the high level and sets the SCL pin to the low level.

Master 1 finishes counting a clock pulse in the low level at point “b”and sets the SCL pin to the high level. Since Master 2 holds the SCL line of the bus at the low level, Master 1 waits for counting a clock pulse in the high level. After Master 2 sets a clock pulse to the high level at point “c” and detects the SCL line of the bus at the high level, Master 1 starts counting a clock pulse in the high level.

The clock pulse on the bus is deteminded by the master device with the shortest high-level period and the master device with the longest low-level period from among those master devices connected to the bus.

(4) Slave address and address recognition mode specification

When the serial bus interface circuit is used with an addressing format to recognize the slave address, clear the ALS (bit 0 in I2CAR) to “0”, and set the SA (bits 7 to 1 in I2CAR) to the slave address.

When the serial bus interface circuit is used with a free data format not to recognize the slave address, set the ALS to “1”. With a free data format, the slave address and the direction bit are not recognized, and they are processed as data from immediately after start condition.

(5) Master / slave selection

Set the MST (bit 7 in SBICR2) to “1” for operating the 88CP77/S77/U77 as a master device.

Reset the MST for operation as a slave device. The MST is cleared to “0” by the hardware after a stop condition on the bus is detected or arbitration is lost.

(6) Transmitter / receiver selection

Set the TRX (bit 6 in SBICR2) to "1" for operating the 88CP77/S77/U77 as a transmitter. Reset the TRX for operation as a receiver. When data with an addressing format is transferred in the slave mode, the TRX is set to "1" if the direction bit (R/$\overline{W}$) sent from the master device is "1", and is cleared to "0" if the bit is "0". In the master mode, after an acknowledge signal is returned from the slave device with the hardware, the TRX is set to "0" if a transmitted direction bit is "1", and set to "1" if it is "0". When an acknowledge signal is not returned, the current condition is maintained. The TRX is cleared to "0" by the hardware after a stop condition on the bus is detected or arbitration is lost.

The following table shows TRX changing conditions and TRX value after changing.

Mode	Direction bit	Conditions	TRX after changing
Slave mode	0	When the received slave address is the same as I2CAR	0
	1		1
Master mode	0	When the ACK signal is returned	1
	1		0

When the serial bus interface circuit is used with a free data format, the TRX is not changed by hardware since the slave address and the direction bit are not recognized, and they are processed as data from immediately after start condition.

(7) Start / stop condition generation

A start condition and 8-bit data are output on the bus by writing "1" to the MST, TRX and BB when the BB (bit 5 in SBICR2) is "0". It is necessary to set the transmitting data to the data buffer register and "1" to ACK beforehand.

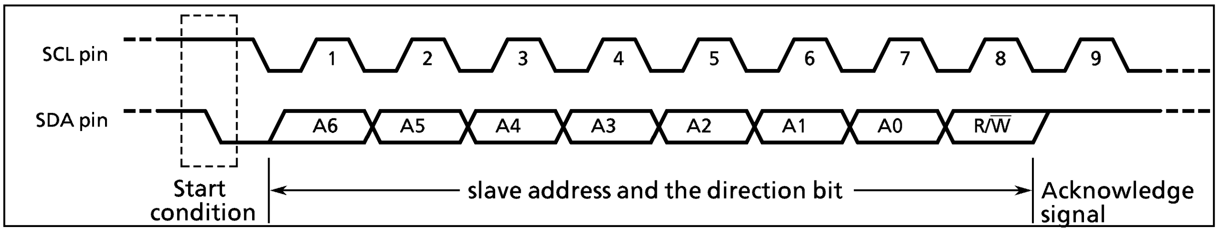


Figure 2-45. Start Condition Generation and Slave Address Generation

When the BB is "1", sequence of generating a stop condition is started by writeng "1" to the MST, TRX, and PIN, and "0" to the BB. Do not modify the contents of MST, TRX, BB and PIN until a stop condition is generated on a bus.

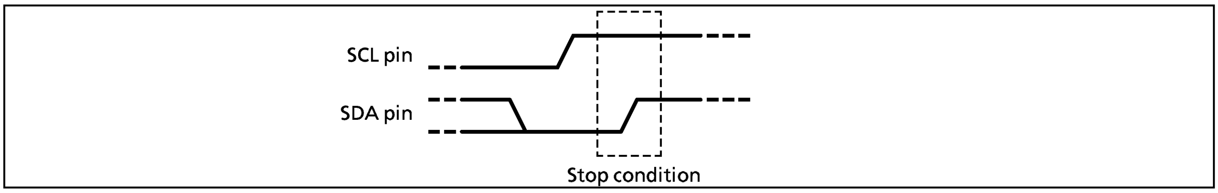


Figure 2-46. Stop Condition Generation

The bus condition can be indicated by reading the contetns of the BB (bit 5 in SBISR). The BB is set to "1" when a start condition on a bus is detected and is cleared to "0" when a stop condition is detected.

(8) Interrupt service request cancel

In the master mode, a serial bus interface interrupt request (INTSBI) occurs after the number of clocks which is specified by the BC and ACK has been transmitted.

In the slave mode, when the received slave address is the same as the value set at the I2CAR, after outputting the acknowledge signal when a GENERAL CALL is received, or when data transfer is complete after matching the slave addresses or receiving a GENERAL CALL, an INTSBI interrupt request occurs.

When a serial bus interface interrupt request occurs, the PIN (bit 4 in SBISR) is cleared to "0". During the time that the PIN is "0", the SCL pin is pulled down to the low level.

Either writing / reading data to / from the SBIDBR sets the PIN to "1".

The time from the PIN being set to "1" until the SCL pin is released takes t_{LOW}.

Although the PIN (bit 4 in SBICR2) can be set to "1" by the program, the PIN is not set to "0" when "0" is written.

(9) Serial bus interface operating mode

The SBIM (bits 3 , 2 in SBICR2) is used to specify the serial bus interface operation mode. Set the SBIM to "10" after confirming that the serial bus interface pin is set to high level when used in the I2C bus mode.

Switch a mode to port after making sure that a bus is free.

(10) Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus in the I2C bus mode, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.

Data on the SDA line is used for bus arbitration of the I2C bus.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on the bus. Master 1 and Master 2 output the same data until point "a". After Master 1 outputs "1" and Master 2, "0", the SDA line of the bus is wired AND and the SDA line is pulled down to the low level by Master 2. When the SCL line of the bus is pulled up at point "b", the slave device reads data on the SDA line, that is, data in Master 2. Data transmitted from Master 1 becomes invalid. The state in Master 1 is called "arbitration lost". A master device which loses arbitration releases the SDA pin and the SCL pin in order not to effect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

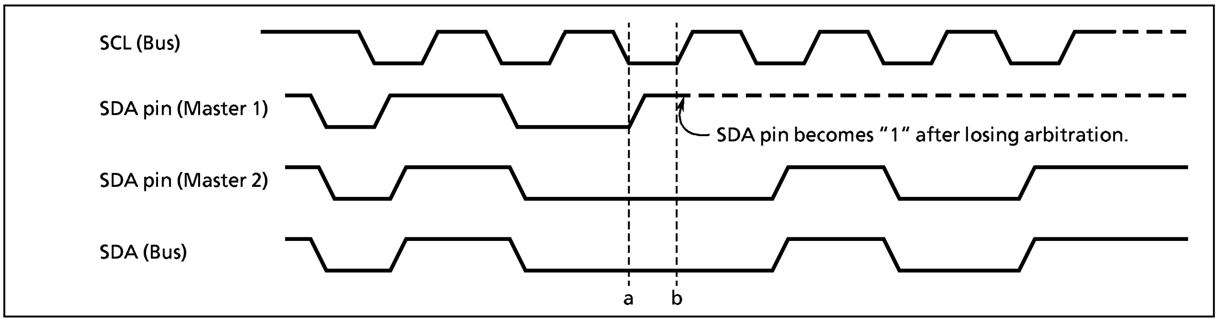


Figure 2-47. Arbitration Lost

The 88CP77/S77/U77 compares levels of the SDA line of the bus with those of the 88CP77/S77/U77 SDA pin at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and the AL (bit 3 in SBISR) is set to "1".

When the AL is set to "1", the MST and TRX are reset to "0" and the mode is switched to a slave receiver mode.

The AL is reset to "0" by writing/reading data to / from the SBIDBR or writing data to the SBICR2.

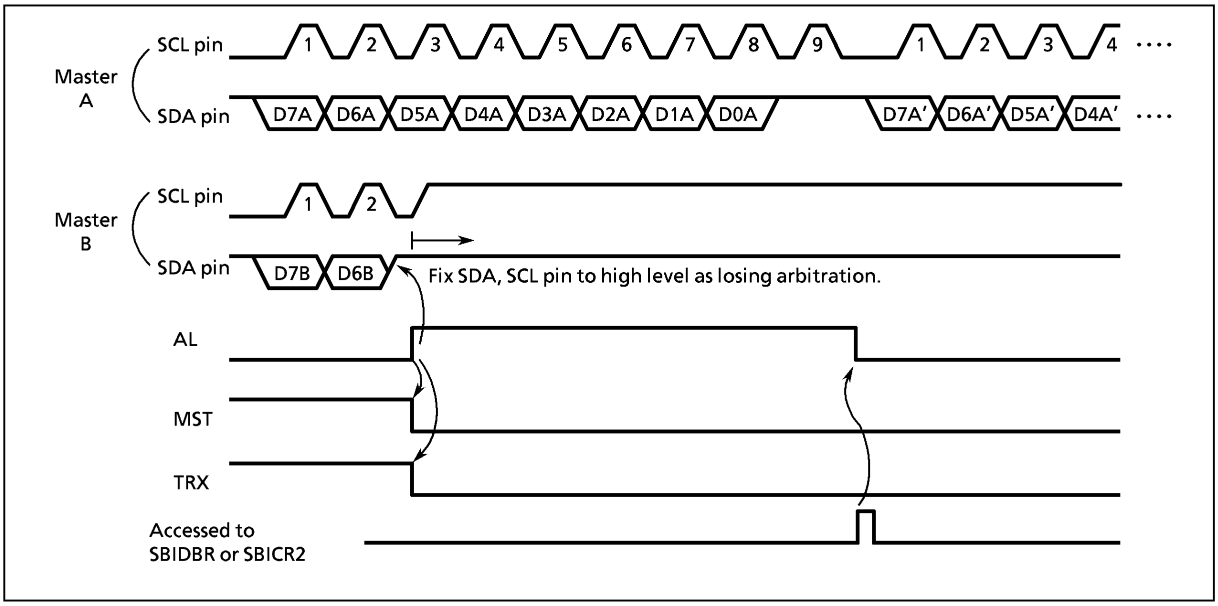

Master A
SCL pin
1 2 3 4 5 6 7 8 9 1 2 3 4 ····
SDA pin
D7A D6A D5A D4A D3A D2A D1A D0A D7A' D6A' D5A' D4A' ····
Master B
SCL pin
1 2
SDA pin
D7B D6B
Fix SDA, SCL pin to high level as losing arbitration.
AL
MST
TRX
Accessed to SBIDBR or SBICR2

Figure 2-48. Example of when 88CP77/S77/U77 are Master B

(11) Slave address match detection monitor

The AAS (bit 2 in SBISR) is set to "1" in the slave mode, in the address recognition mode (ALS = 0), or when receiving a slave address with the same value that sets a GENERAL CALL or I2CAR. When the ALS is "1", the AAS is set to "1" after receiving the first 1-word of data. The AAS is cleared to "0" by after writing / reading data to / from a data buffer register.

(12) GENERAL CALL detection monitor

The AD0 (bit 1 in SBISR) is set to "1" in the slave mode, when all 8-bit data received immediately after a start condition are "0". The AD0 is cleared to "0" when a start or stop condition is detected on the bus.

(13) Last received bit monitor

The SDA value stored at the rising edge of the SCL line is set to the LRB (bit 0 in SBISR). When the contents of the LRB are read immediately after an INTSBI interrupt request is generated in the acknowledge mode, and ACK signal is read.

(14) Software Reset Function

Software reset function is used to initialize SBI, when SBI is rocked by external noise, etc.

SWRST (bit 0 in SBICR) is set to "1", internal reset signal pulse is generated and inputted into SBI circuit.

All command registers and status registers are initialized to an initial value.

SWRST is automatically cleared to "0" after initializing SBI circuit.

2.10.5 Data transfer in I^2C bus mode

(1) **Device initialization**

First, set the ACK in the SBICR1 to "1", the BC to "000", and the data length to 8-bit to count a clock pulse for the acknowledge signal. In addition, set the transmit frequency to the SCK.

Next, set the slave address to the SA in the I2CAR. Clear the ALS to "0" to set the addressing format. After confirming that the serial bus interface pin is high level, for specifying the default setting to a slave receiver mode, clear "0" to the MST, TRX, and BB in the SBICR2; "1" to the PIN; "10" to the SBIM; and "0" to bits 1 and 0.

Note: To initialize the serial bus interface circuit, a constant period that the start conditions are not generated for any device is required after all devices which are connected to the bus are initialized. Then, the initialization must be completed during the period. If not, other devices may start transmitting data before the serial bus interface circuit has been initialized. Thus, data can not be normally received.

(2) **Start condition and slave address generation**

Confirm a bus free status (when BB = 0).

Set the ACK to "1" and specify a slave address and a direction bit to be transmitted to the SBIDBR.

When the BB is "0", the start condition are generated and the slave address and the direction bit which are set to the SBIDBR are output on a bus by writing "1" to the MST, TRX, BB, and PIN. An INTSBI interrupt request occurs at the 9th falling edge of the SCL clock cycle, and the PIN is cleared to "0". The SCL pin is pulled down to the low level while the PIN is "0". When an interrupt request occurs, the TRX changes by the hardware according to the direction bit only when an acknowledge signal is returned from the slave device.

Note 1: The slave address to be output to the SBIDBR must be set after the bus free is detected by software. If setting of the slave address is executed before detection bus free, the current output data may be corrupted.

Note 2: The bus free must be confirmed by software within 98.0 μs (the shortest transmitting time according to the I2C bus standard) after setting of the slave address to be output. Only when the bus free is confirmed, set "1" to the MST, TRX, BB, and PIN to generate the start conditions. If the start conditions are generated without writing "1" to them, transferring may be executed by other masters between the time when the slave address to be output to the SBIDBR is written and the time when "1" is written to the MST, TRX, BB, and PIN in the SBICR2. Thus, the slave address may be corrupted.

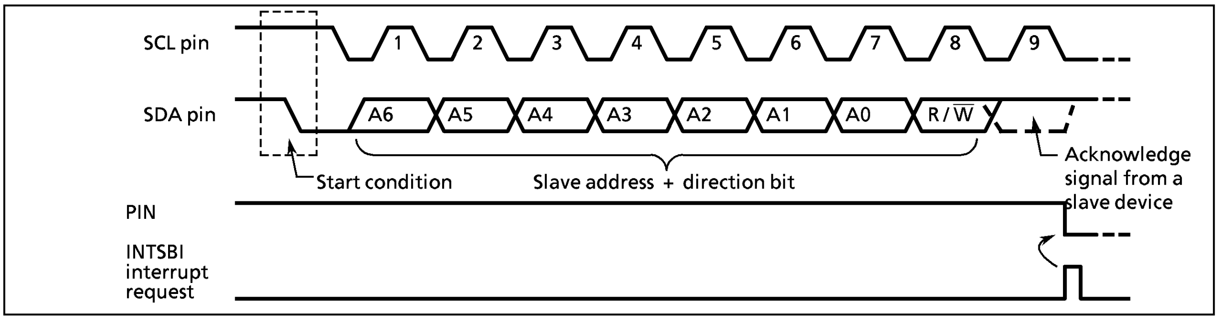


Figure 2-49. Start Condition Generation and Slave Address Transfer

(3) **1-word data transfer**

Check the MST by the INTSBI interrupt process after a 1-word data transfer is completed, and determine whether the mode is a master or slave.

a. When the MST is "1" (Master mode)

Check the TRX and determine whether the mode is a transmitter or receiver.

① When the TRX is "1" (Transmitter mode)

Check the LRB. When the LRB is "1", a receiver does not request data. Implement the process to generate a stop condition and terminate data transfer.

When the LRB is "0", the receiver requests new data. When the next transmitted data is other than 8 bits, set the BC and write the transmitted data to the SBIDBR after setting ACK to "1". After writing the data, the PIN becomes "1", a serial clock pulse is generated for transferring a new 1-word of data from the SCL pin, and then the 1-word data is transmitted. After the data is transmitted, an INTSBI interrupt request occurs. The PIN becomes "0" and the SCL pin is pulled down to the low level. If the data to be transferred is more than one word in length, repeat the procedure from the LRB checking above.

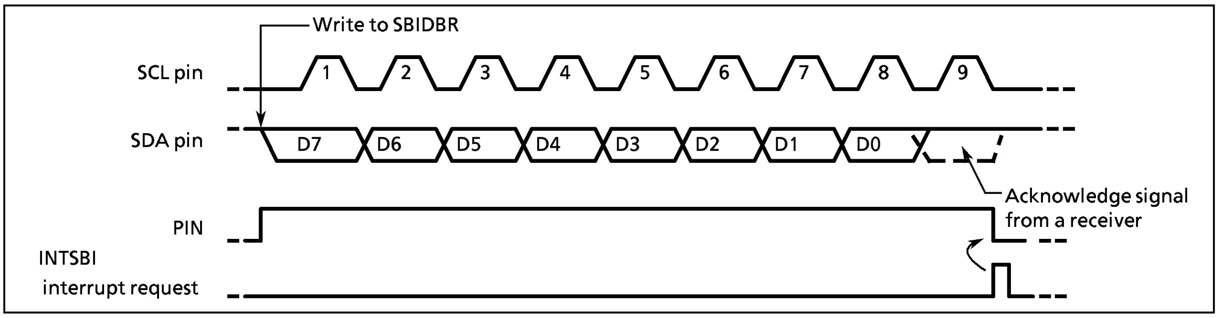


Figure 2-50. Example when BC = "000", ACK = "1"

② When the TRX is "0" (Receiver mode)

When the next transmitted data is other than 8 bits, set the BC again. Set the ACK to "1" and read the received data from the SBIDBR (data which is read immediately after a slave address is sent is undefined). After the data is read, the PIN becomes "1". The 88CP77/S77/U77 outputs a serial clock pulse to the SCL to transfer new 1-word of data and sets the SDA pin to "0" at the acknowledge signal timing.

An INTSBI interrupt request occurs and the PIN becomes "0". The 88CP77/S77/U77 outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the SBIDBR.

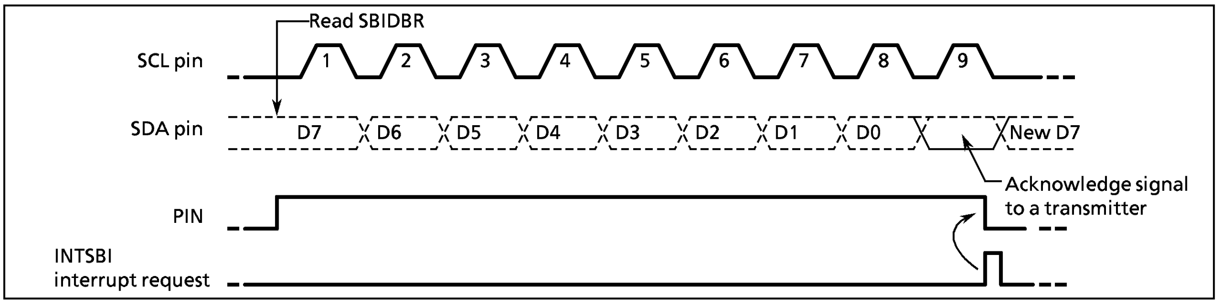


Figure 2-51. Example when BC = "000", ACK = "1"

In order to terminate transmitting data to a transmitter, clear the ACK to "0" before reading data which is 1 word before the last data to be received. The last data does not generate a clock pulse for the acknowledge signal. After the data is transmitted and an interrupt request has occurred, set the BC to "001" and read the data. The 88CP77/S77/U77 generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA line of the bus keeps the high level. The transmitter receives the high-level signal as an ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, the 88CP77/S77/U77 generates a stop condition and terminates data transfer.

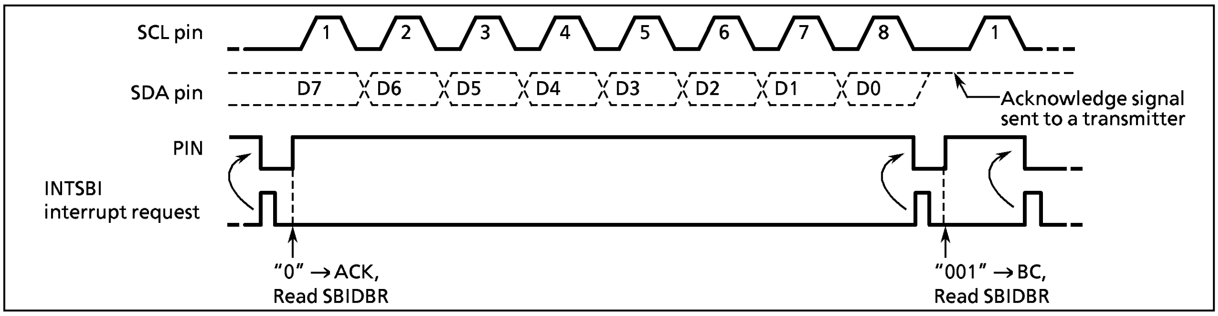


Figure 2-52. Termination of Data Transfer in Master Receiver Mode

b. When the MST is "0" (Slave mode)

In the slave mode, the 88CP77/S77/U77 operates either in normal slave mode or in slave mode after losing arbitration.

In the slave mode, an INTSBI interrupt request occurs when the 88CP77/S77/U77 receives a slave address or a GENERAL CALL from the master device, or when a GENERAL CALL is received and data transfer is complete after matching a received slave address. In the master mode, the 88CP77/S77/U77 operates in a slave mode if it is losing arbitration. An INTSBI interrupt request occurs when word data transfer terminates after losing arbitration. When an INTSBI interrupt request occurs, the PIN (bit 4 in the SBICR2) is reset, and the SCL pin is pulled down to the low level. Either reading/writing from/to the SBIDBR or setting the PIN to "1" releases the SCL pin after taking t_{LOW} time.

Check the AL (bit 3 in the SBISR), the TRX (bit 6 in the SBISR), the AAS (bit 2 in the SBISR), and the AD0 (bit 1 in the SBISR) and implements processes according to conditions listed in the next table.

Table 2-14. Operation in the Slave Mode

TRX	AL	AAS	AD0	Conditions	Process
1	1	1	0	The 88CP77/S77/U77 loses arbitration when transmitting a slave address and receives a slave address of which the value of the direction bit sent from another master is "1".	Set the number of bits in 1 word to the BC and write transmitted data to the SBIDBR.
	0	1	0	In the slave receiver mode, the 88CP77/S77/U77 receives a slave address of which the value of the direction bit sent from the master is "1".	
		0	0	In the slave transmitter mode, 1-word data is transmitted.	Check the LRB. If the LRB is set to "1", set the PIN to "1" since the receiver does not request next data. Then, clear the TRX to "0" release the bus. If the LRB is cleared to "0", set the number of bits in a word to the BC and write transmitted data to the SBIDBR since the receiver requests next data.
0	1	1	1/0	The 88CP77/S77/U77 loses arbitration when transmitting a slave address and receives a slave address or GENERAL CALL of which the value of the direction bit sent from another master is "0".	Read the SBIDBR for setting the PIN to "1" (reading dummy data) or write "1" to the PIN.
		0	0	The 88CP77/S77/U77 loses arbitration when transmitting a slave address or data and terminates transferring word data.	
	0	1	1/0	In the slave receiver mode, the 88CP77/S77/U77 receives a slave address or GENERAL CALL of which the value of the direction bit sent from the master is "0".	
		0	1/0	In the slave receiver mode, the 88CP77/S77/U77 terminates receiving of 1-word data.	Set the number of bits in a word to the BC and read received data from the SBIDBR.

(4) **Stop condition generation**

When the BB is "1", a sequence of generating a stop condition is started by writing "1" to the MST, TRX, and PIN, and "0" to the BB. Do not modify the contents of the MST, TRX, BB, PIN until a stop condition is generated on a bus.

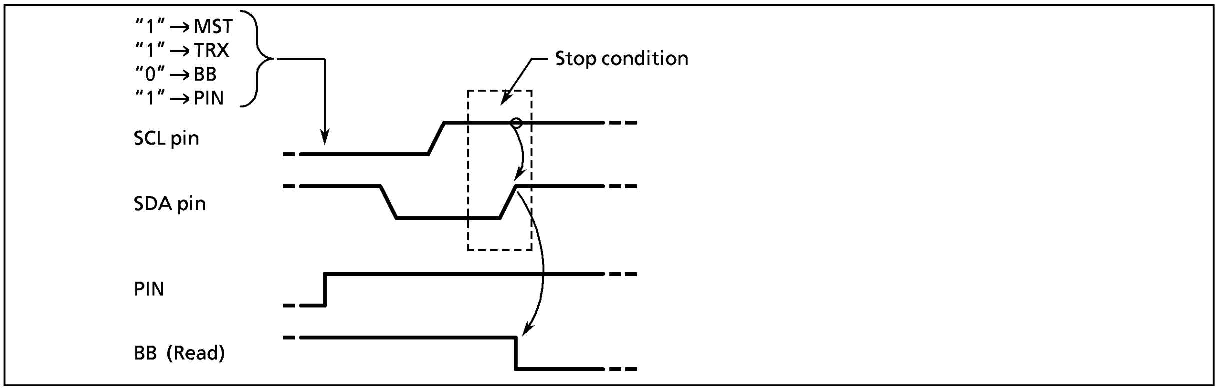


Figure 2-53. Stop Condition Generation

(5) **Restart**

Restart is used to change the direction of data transfer between a master device and a slave device during transferring data. The following explains how to restart when the 88CP77/S77/U77 are in the master mode.

Specify "0" to the MST, TRX, and BB and "1" to the PIN and release the bus. The SDA pin retains the high level and the SCL pin is released. Since a stop condition is not generated on a bus, a bus is assumed to be in a busy state from other devices. Check the BB until it becomes "0" to check that the SCL pin of the 88CP77/S77/U77 are released. Check the LRB until it becomes "1" to check that the SCL line of a bus is not pulled down to the low level by other devices. After confirming that a bus stays in a free state, generate a start condition with procedure (2).

In order to meet setup time when restarting, take at least 4.7 [μs] of waiting time by software from the time of restarting to confirm that the bus is free until the time to generate the start condition.

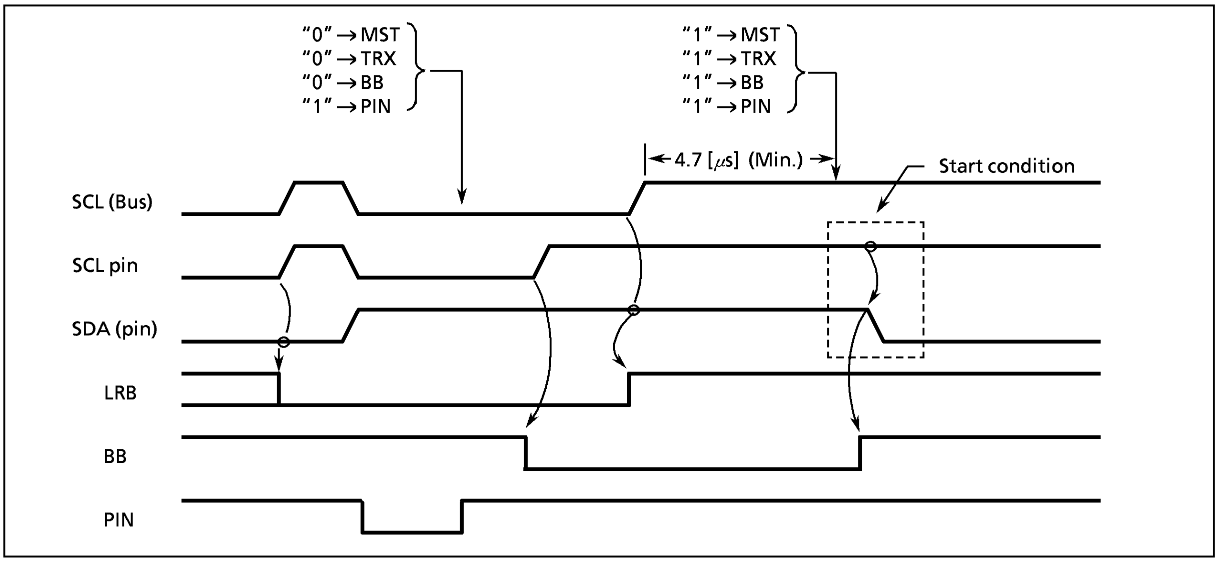


Figure 2-54. Timing Diagram when Restarting the 88CP77/S77/U77

2.10.6 Clocked-synchronous 8-bit SIO Mode Control

The following registers are used to control the serial bus interface (SBI-ver. C) and monitor the operation status in the clocked-synchronous 8-bit SIO mode.

Serial Bus Interface Control Register 1

SBICR1 (00020_H)

7	6	5	4	3	2	1	0
SIOS	SIO INH	SIOM		"0"	SCK		

(Initial value: 0000 *000)

SIOS	Indicate transfer start/stop	0: Stop 1: Start	Write only
SIOINH	Continue/abort transfer	0: Continue transfer 1: Abort transfer (automatically cleared after abort)	
SIOM	Transfer mode select	00: 8-bit transmit mode 01: reserved 10: 8-bit transmit / receive mode 11: 8-bit receive mode	
SCK	Serial clock select	000: $fc/2^5$ (250 kHz) 001: $fc/2^6$ (125 kHz) 010: $fc/2^7$ (62.5 kHz) 011: $fc/2^8$ (31.25 kHz) 100: $fc/2^9$ (15.62 kHz) 101: $fc/2^{10}$ (7.8 kHz) 110: $fc/2^{11}$ (3.90 kHz) } at fc = 8 MHz (Output on $\overline{\text{SCK3}}$ pin) 111: External clock (input from $\overline{\text{SCK3}}$ pin)	

*Note 1: * ; don't care*
Note 2: Set the SIOS to "0" when setting the transfer mode or serial clock.
Note 3: SBICR1 is write-only register, which cannot be used with any of read-modify-write instruction such as bit manipulation, etc.

Serial Bus Interface Data Buffer Register

SBIDBR (00021_H)

7	6	5	4	3	2	1	0

Read / Write

Note: Cannot read the data which was written into SBIDBR, since a write data buffer and a read buffer are independent in SBIDBR. Therefore, SBIDBR cannot be used with any in read-modify-write instruction such as bit manipulation, etc.

Serial Bus Interface Control Register 2

SBICR2 (00023_H)

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

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

SBIM	Serial bus interface operation mode selection	00: Port mode (serial bus interface output disable) 01: Clocked-synchronous 8-bit SIO mode 10: I^2C bus mode 11: reserved	Write only

*Note 1: * ; don't care*
Note 2: Switch a mode to port after data transfer is complete.
Note 3: Switch a mode to I^2C bus mode or clocked-synchronous 8-bit SIO mode after confirming that input signal via port is high level.
Note 4: SBICR2 is write-only register, which cannot be used with any of read-modify-write instruction such as bit manipulation, etc.
Note 5: Clear bits 7 to 5 and bits 1 to 0 in SBICR2 to "0", and set bit 4 to "1".

Serial Bus Interface Status Register

SBISR (00023_H)

7	6	5	4	3	2	1	0
"1"	"1"	"1"	"1"	SIOF	SEF	"1"	"1"

SIOF	Serial transfer operating status monitor	0: Transfer terminated 1: Transfer in process	Read only
SEF	Shift operating status monitor	0: Shift operation terminated 1: Shift operation in process	

Figure 2-55. Control Register / Data Buffer Register / Status Register in SIO Mode

(1) Serial clock

a. Clock source

The SCK(bits 2 to 0 in SBICR1) is used to select the following functions.

① Internal clock

In an internal clock mode, any of seven frequencies can be selected. The serial clock is output to the outside on the $\overline{\text{SCK3}}$ pin. The $\overline{\text{SCK3}}$ pin becomes a high level when data transfer starts. When writing (in the transmit mode) or reading (in the receive mode) data cannot follow the serial clock rate, an automatic-wait function is executed to stop the serial clock automatically and hold the next shift operation until reading or writing is complete.

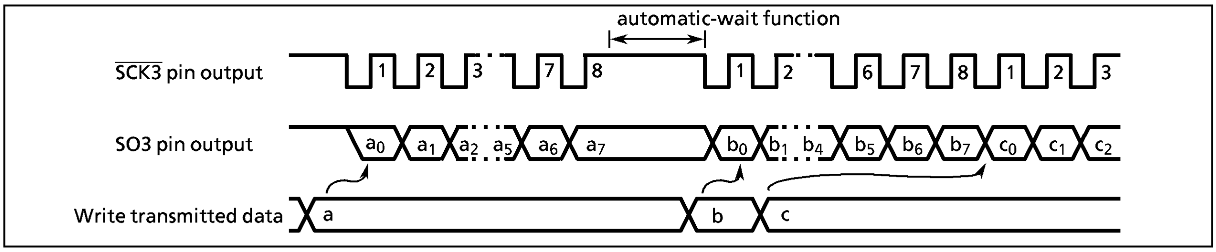


Figure 2-56. Automatic Wait Function

② External clock (SCK = "111")

An external clock supplied to the $\overline{\text{SCK3}}$ pin is used as the serial clock. In order to ensure shift operation, a pulse width of at least 4 machine cycles is required for both high and low levels in the serial clock. The maximum data transfer frequency is 250 kHz (fc = 8 MHz).

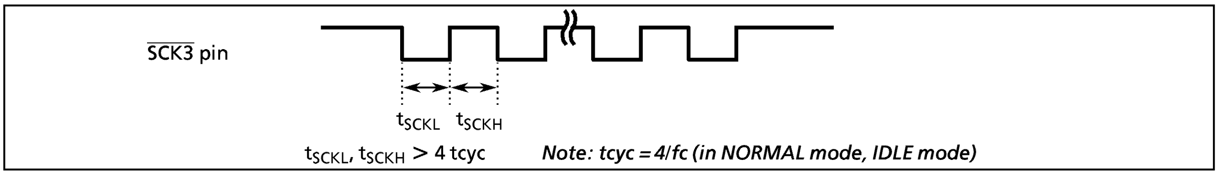

SCK3 pin
t_{SCKL}
t_{SCKH}
t_{SCKL}, t_{SCKH} > 4 tcyc
Note: tcyc = 4/fc (in NORMAL mode, IDLE mode)

Figure 2-57. The Maximum Data Transfer Frequency in The External Clock Input

b. Shift edge

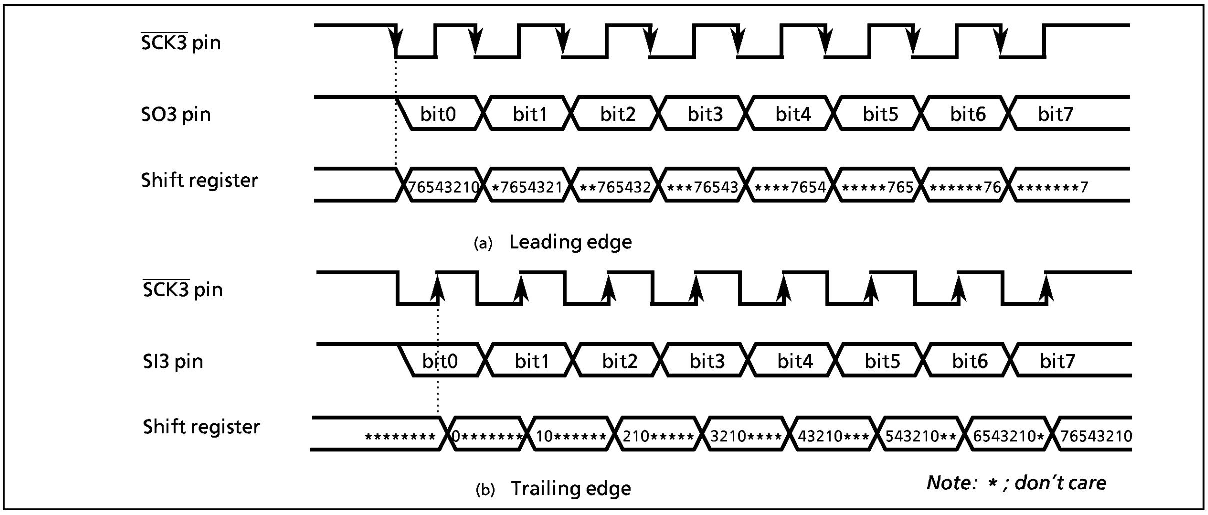
The leading edge is used to transmit data, and the trailing edge is used to receive data.

① Leading edge

Data is shifted on the leading edge of the serial clock (at a falling edge of the $\overline{\text{SCK3}}$ pin input / output).

② Trailing edge

Data is shifted on the trailing edge of the serial clock (at a rising edge of the $\overline{\text{SCK3}}$ pin input / output).



(a) Leading edge

(b) Trailing edge

*Note: * ; don't care*

Figure 2-58. Shift Edge

(2) Transfer mode

The SIOM (bits 5 and 4 in SIO1CR) is used to select a transmit, receive, or transmit/receive mode.

a. 8-bit transmit mode

Set a control register to a transmit mode and write data to the SBIDBR.

After the data is written, set the SIOS to "1" to start data transfer. The transmitted data is transferred from the SBIDBR to the shift register and output to the SO3 pin in synchronous with the serial clock, starting from the least significant bit (LSB). When the data is transferred to the shift register, the SBIDBR becomes empty. The INTSBI (buffer empty) interrupt request is generated to request new data.

When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated if new data is not loaded to the data buffer register after the specified 8-bit data is transmitted. When new data is written, automatic-wait function is canceled.

When the external clock is used, data should be written to the SBIDBR before new data is shifted. The transfer speed is determined by the maximum delay time between the time when an interrupt request is generated and the time when data is written to the SBIDBR by the interrupt service program.

When the transmit is started, the same value as the final bit of the last data is output until the falling edge of the $\overline{\text{SCK3}}$ after the SIOF goes "1".

Transmitting data is ended by cleaning the SIOS to "0" by the buffer empty interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, the transmitted mode ends when all data is output. In order to confirm if data is surely transmitted by the program, set the SIOF (bit 3 in the SBISR) to be sensed. The SIOF is cleared to "0" when transmitting is complete. When the SIOINH is set, transmitting data stops. The SIOF turns "0".

When the external clock is used, it is also necessary to clear the SIOS to "0" before new data is shifted; otherwise, dummy data is transmitted and operation ends.

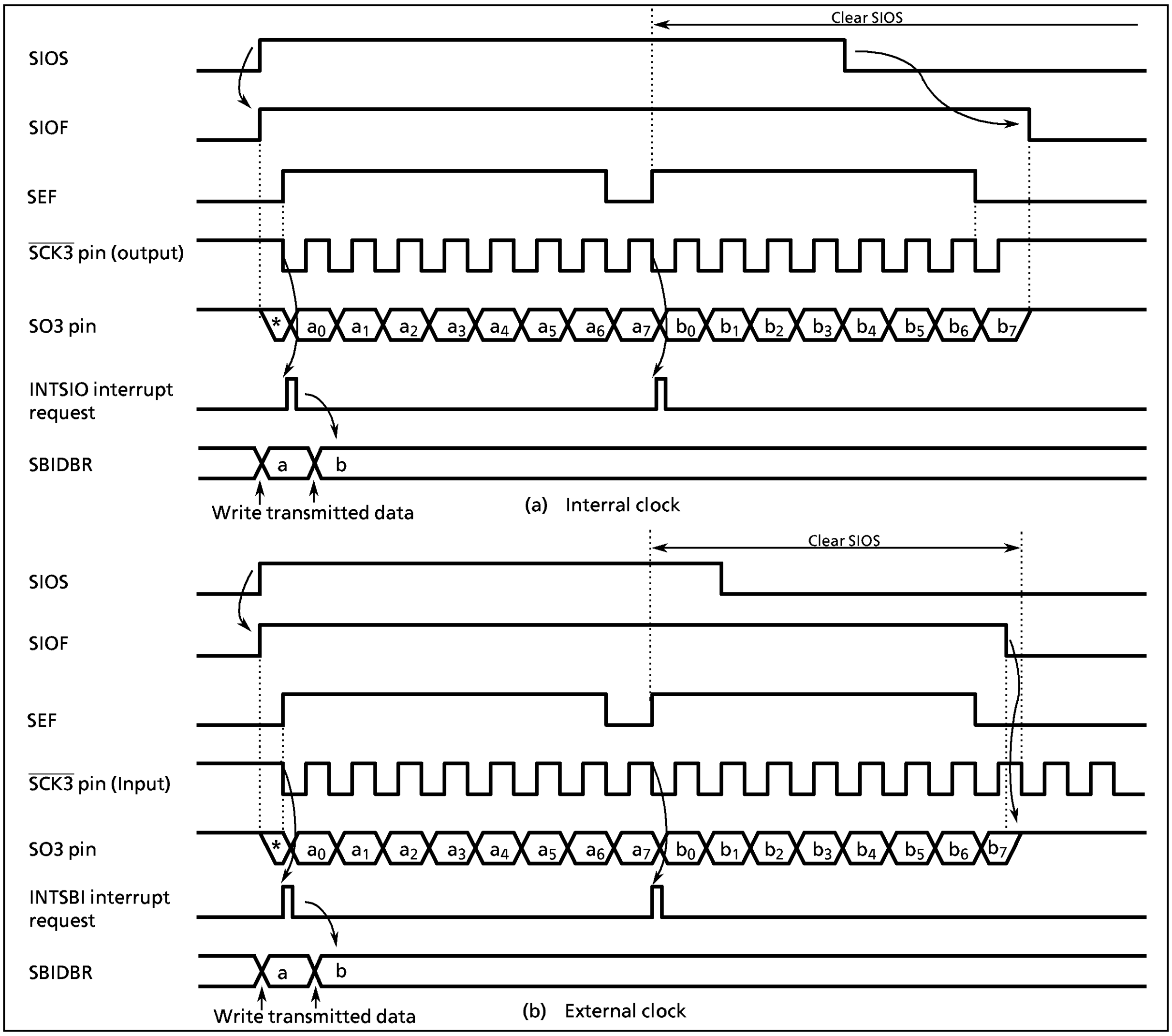


Figure 2-59. Transfer Mode

Example: SIO0 transfer end command (External clock)

```
STEST1:  TEST   (SBISR) . SEF            ;  If SEF = 1 then loop
         JRS    F , STEST1
STEST2:  TEST   (P0 read) . 2            ;  If SCK3 = 0 then loop
         JRS    T , STEST2
         LD     (SBICR1) , 00000111B     ;  SIOS ← 0
```

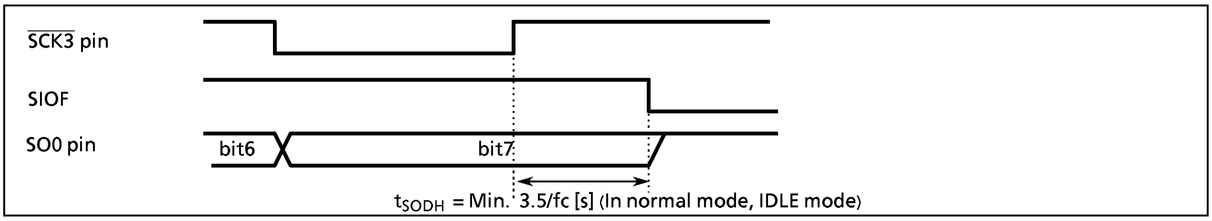

$\overline{\text{SCK3}}$ pin
SIOF
SO0 pin
bit6
bit7
t_{SODH} = Min. 3.5/fc [s] (In normal mode, IDLE mode)

Figure 2-60. Transmitted Data Hold Time at End of Transmit

b. 8-bit receive mode

Set a control register to a receive mode and the SIOS to "1" for switching to a receive mode. Data is received from the SI0 pin to the shift register in synchronous with the serial clock, starting from the least significant bit (LSB). When the 8-bit data is received, the data is transferred from the shift register to the SBIDBR. The INTSBI (buffer full) interrupt request is generated to request of reading the received data. The data is then read from the SBIDBR by the interrupt service program.

When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated until the received data is read from the SBIDBR.

When the external clock is used, since shift operation is synchronized with the clock pulse provided externally, the received data should be read before new data is transferred to the SBIDBR. If the received data is not read, further data to be received is canceled. The maximum transfer speed when the external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when received data is read.

Receiving data is ended by clearing the SIOS to "0" by the buffer full interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, received data is transferred to the SBIDBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm if data is surely received by the program, set the SIOF (bit 3 in SBIDBR) to be sensed. The SIOF is cleared to "0" when receiving is complete. After confirming that receiving has ended, the last data is read. When the SIOINH is set, receiving data stops. The SIOF turns "0" (the received data becomes invalid, therefore no need to read it).

Note: When the transfer mode is switched, the ***SBIDBR*** *contents are lost. In case that the mode needs to be switched, conclude receiving data by clearing the* ***SIOS*** *to "0", read the last data, and then switch the mode.*

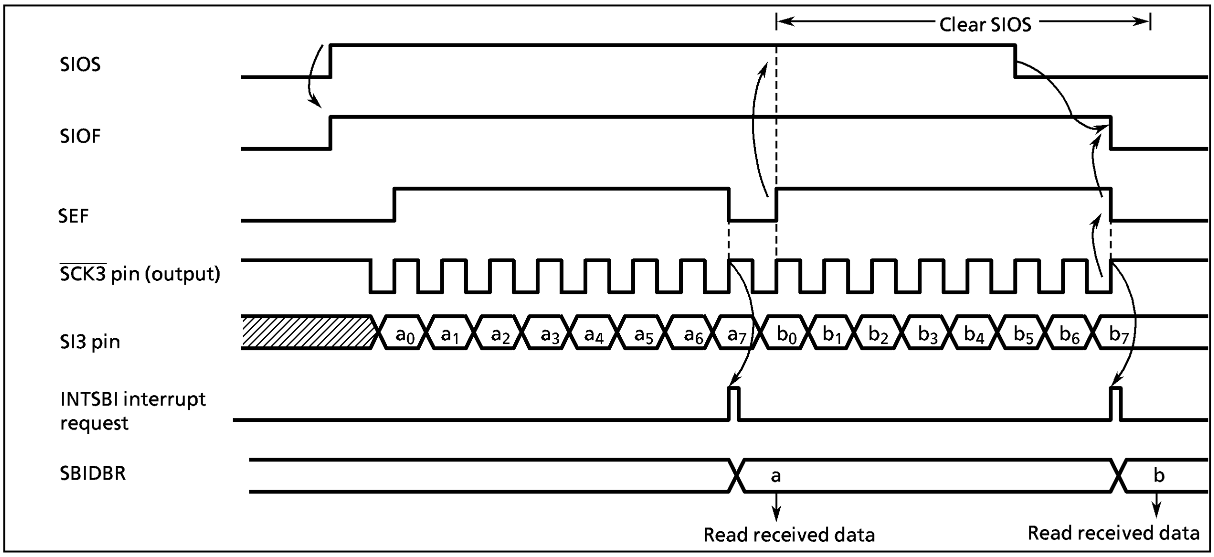


Figure 2-61. Receive Mode (Example: Internal clock)

c. 8-bit transmit / receive mode

Set a control register to a transmit / receive mode and write data to the SBIDBR. After the data is written, set the SIOS to "1" to start transmitting / receiving. When transmitting, the data is output from the SO3 pin on the leading edges in synchronous with the serial clock, starting from the least significant bit (LSB). When receiving, the data is input to the SI3 pin on the trailing edges of the serial clock. 8-bit data is transferred from the shift register to the SBIDBR, and the INTSBI interrupt request occurs. The interrupt service program reads the received data from the data buffer register and writes data to be transmitted. The SBIDBR is used for both transmitting and receiving. Transmitted data should always be written after received data is read.

When the internal clock is used, automatic-wait function is initiated until received data is read and next data is written.

When the external clock is used, since the shift operation is synchronized with the serial clock provided externally, received data is read and transmitted data is written before new shift operation is executed. The maximum transfer speed when the external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when received data is read and transmitted data is written.

When the transmit is started, the same value as the final bit of the last data is output until the falling edge of the $\overline{SCK3}$ after the SIOF goes "1".

Transmitting / receiving data is ended by cleaning the SIOS to "0" by the INTSBI interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, received data is transferred to the SBIDBR in complete blocks. The transmit / receive mode ends when the transfer is complete. In order to confirm if data is surely transmitted / received by the program, set the SIOF (bit 3 in SBISR) to be sensed. The SIOF becomes "0" after transmitting / receiving is complete. When the SIOINH is set, transmitting / receiving data stops. The SIOF turns "0".

Note: When the transfer mode is switched, the SBIDBR contents are lost. In case that the mode needs to be switched, conclude transmitting/receiving data by clearing the SIOS to '0", read the last data, and then switch the transfer mode.

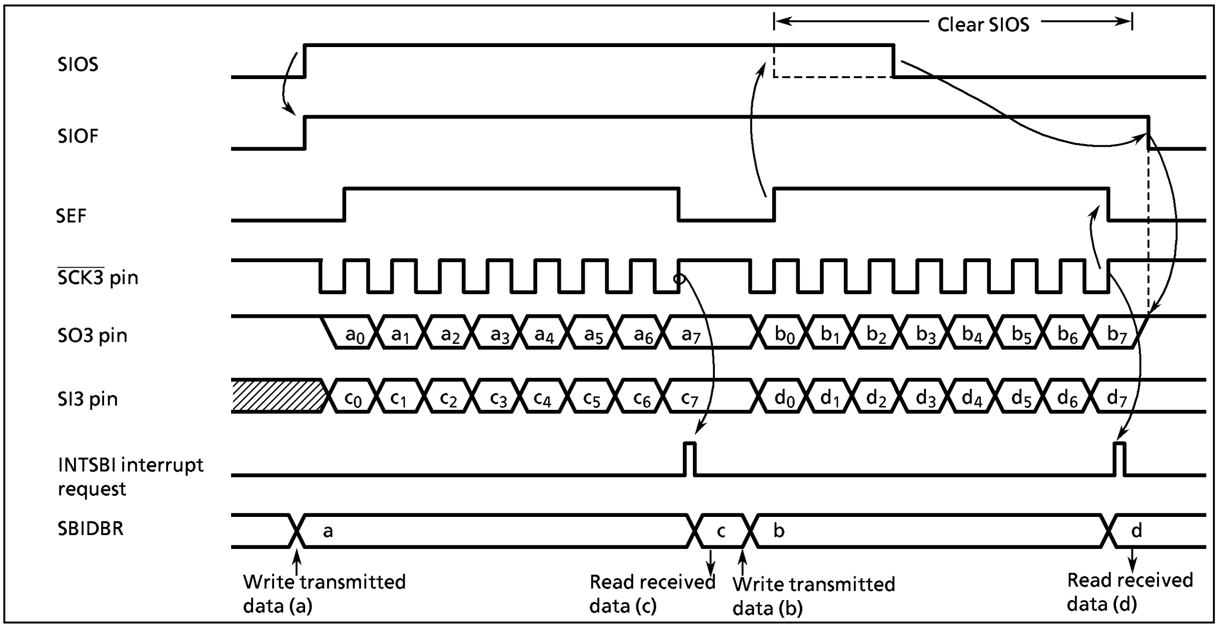


Figure 2-62. Transmit / Receive Mode (Example: Internal clock)

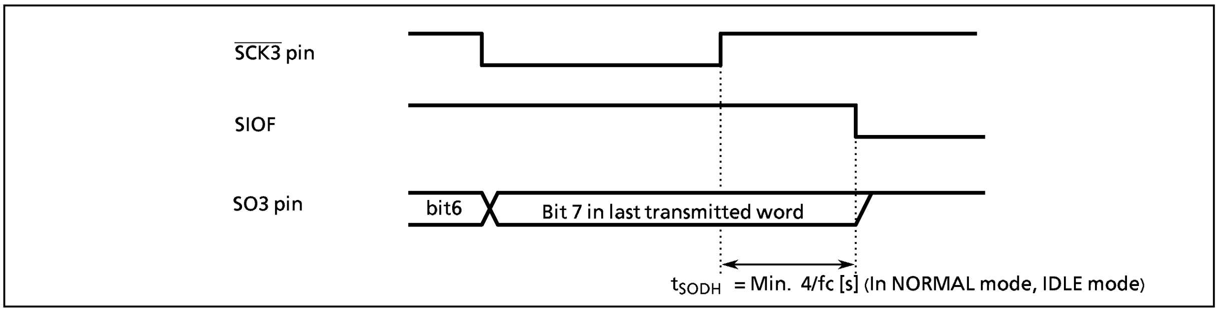

$\overline{\text{SCK3}}$ pin
SIOF
SO3 pin
bit6
Bit 7 in last transmitted word
t_{SODH} = Min. 4/fc [s] (In NORMAL mode, IDLE mode)

Figure 2-63. Transmitted Data Hold Time at End of Transmit / Receive

2.11 Serial Interface (SIO0, SIO1)

The 88CP77/S77/U77 each have two built-in 8-bit serial interface channels that operate in sync with the clock: SIO0 and SIO1. SIO0 has a 32-byte send/receive data buffer. SIO1 has an 8-byte send/receive data buffer. SIO0 can consecutively transfer up to 256 bits of data automatically; SIO1 up to 64 bits.

Serial interface is connected to external devices via the following pins:

(1) P16 (SO0), P15 (SI0), P14 ($\overline{\text{SCK0}}$)
(2) P12 (SO1), P11 (SI1), P10 ($\overline{\text{SCK1}}$)

Serial interface pins are also used as port P1. When used as serial interface pins, set the output latches of port P1 to 1. In send mode, SIx pins: P15 and P11 can be used as general-purpose I/O ports; in receive mode, SOx pins P16 and P12.

2.11.1 Configuration

The SIO0 and SIO1 have the same configuration, except for the addresses/bit positions of the control/status registers and buffer registers.

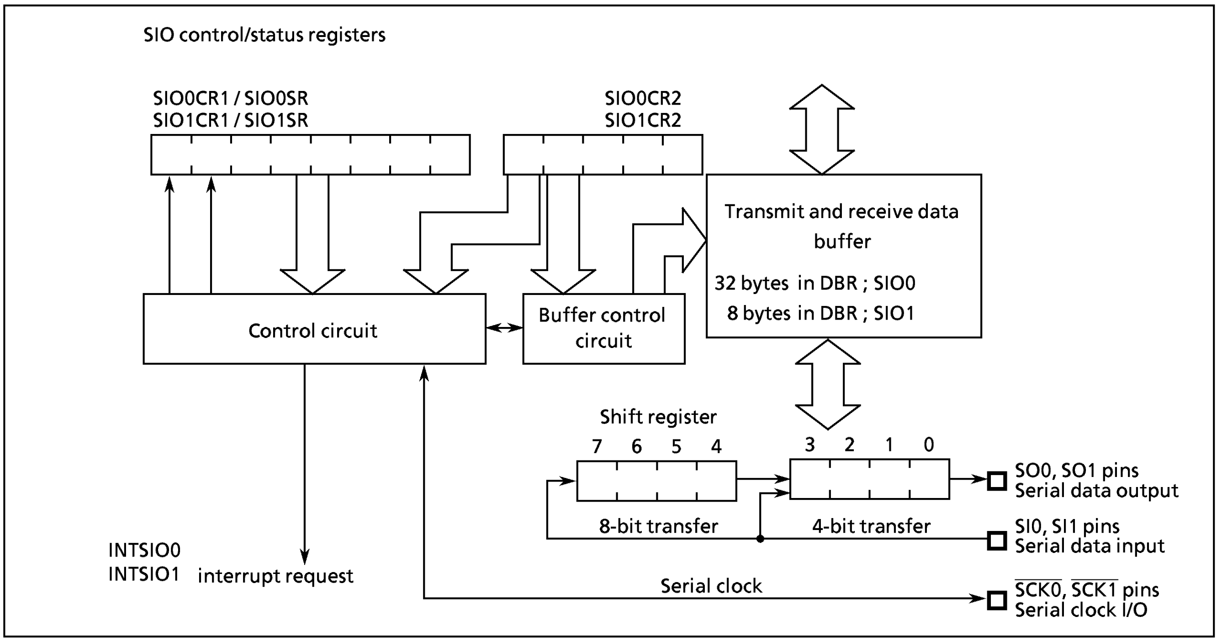


Figure 2-64. Serial Interfaces

2.11.2 Control

SIO0 and SIO1 are controlled by SIO0CR1/SIO0CR2 and SIO1CR1/SIO1CR2. SIO states can be determined by reading SIO0SR and SIO1SR.

The send/receive data buffer is controlled by BUF (bits 4 to 0 in SIO0CR2) or (bits 2 to 0 in SIO1CR2). The send/receive data buffer for SIO0 is assigned to addresses 0FE0 to $0FFF_H$ in DBR; for SIO1, addresses 0FD8 to $0FDF_H$. The buffer for SIO0 can consecutively transfer up to 32 words at a time; the buffers for SIO1, up to 8 words. When transfer of the set number of words is complete, a buffer empty (in send mode) or buffer full (in receive or send/receive mode) interrupt (INTSIO0 or INTSIO1) is generated.

When using the internal clock as the serial clock in 8-bit send/receive or 8-bit receive mode, a fixed-time wait can be inserted to the serial clock every time one word is transferred. One of the four wait times can be selected by WAIT (bits 6 and 5 in SIO0CR2) or (bits 4 and 3 in SIO1CR2).

SIO0, SIO1 Control Registers 1

	7	6	5	4	3	2	1	0	
SIO1CR1 (00028_H) **SIO0CR1** ($00FD6_H$)	SIOS	SIO INH	SIOM			SCK			(Initial value: 0000 0000)

SIOS	Indicate transfer start/stop	0: Stop 1: Start	write only
SIOINH	Continue/abort transfer	0: Continue transfer 1: Abort transfer (automatically cleared after abort)	
SIOM	Transfer mode select	000: 8-bit transmit mode 010: 4-bit transmit mode 100: 8-bit transmit/receive mode 110: 8-bit receive mode 110: 4-bit receive mode	
SCK	Serial clock select	000: Internal clock fc / 2^{13} or fs / 2^5 [Hz] 001: Internal clock fc / 2^8 010: Internal clock fc / 2^6 011: Internal clock fc / 2^5 (Output on $\overline{\text{SCK}}$ pin) 111: External clock (input from $\overline{\text{SCK}}$ pin)	

Note 1: fc ; High-frequency clock [Hz], fs ; Low-frequency clock [Hz]
Note 2: Set SIOS to "0" and SIOINH to "1" when setting the transfer mode or serial clock

SIO0, SIO1 Status Registers

	7	6	5	4	3	2	1	0
SIO1SR (00028_H) **SIO0SR** ($00FD6_H$)	SIOF	SEF	"1"	"1"	"1"	"1"	"1"	"1"

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

SIO0 Control Registers 2

	7	6	5	4	3	2	1	0	
SIO0CR2 ($00FD7_H$)		WAIT		BUF					(initial value: *000 0000)

WAIT	Wait Control	Please set "00", except for 8-bit transmit/receive mode or receive mode 00: T_f = T_D (Non wait) 01: T_f = $2T_D$ 10: T_f = $4T_D$ (wait) 11: T_f = $8T_D$	write only
BUF	Number of transfer words	Buffer address used 00000: 1 word transfer0FE0 00001: 2 word transfer0FE0 to $0FE1_H$ 00010: 3 word transfer0FE0 to $0FE2_H$ ∫ ∫ 11101: 30 word transfer0FE0 to $0FFD_H$ 11110: 31 word transfer0FE0 to $0FFF_H$ 11111: 32 word transfer0FE0 to $0FFF_H$	

Note: SIO1CR1, SIO0CR2, SIO0CR1 are write-only registers, which cannot access any of in read-modify-write instruction such as bit operate, etc.

Figure 2-65. SIO Control Registers and Status Registers

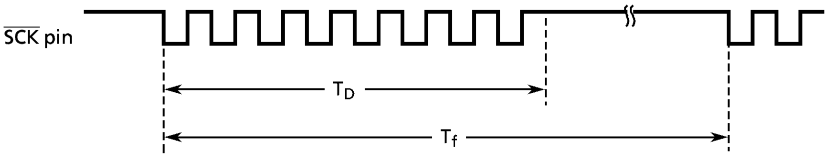
SIO1 Control Registers 2

	7	6	5	4	3	2	1	0	
SIO1CR2 (00029_H)				WAIT		BUF			(initial value: ***0 0000)

WAIT	Wait Control	Please set "00", except for 8-bit transmit/receive mode or receive mode 00: T_f = T_D (Non wait) 01: T_f = $2T_D$ 10: T_f = $4T_D$ (wait) 11: T_f = $8T_D$	write only
BUF	Number of transfer words	Buffer address used (SIO1) 000:1 word transfer 0FD8$_H$ 001:2 word transfer 0FD8 to 0FD9H 010:3 word transfer 0FD8 to 0FDA$_H$ 011:4 word transfer 0FD8 to 0FDB$_H$ 100:5 word transfer 0FD8 to 0FDC$_H$ 101:6 word transfer 0FD8 to 0FDD$_H$ 110:7 word transfer 0FD8 to 0FDE$_H$ 111:8 word transfer 0FD8 to 0FDF$_H$	

Note 1: SIO1CR2 is write-only registers, which cannot access any of in read- modify-write instruction such as bit operate, etc.

Note 2: T_f; frame time, T_D ; data transfer time



Note 3: The lower 4 bits of each buffer are used during 4-bit transfers. Zeros (0) are stored to the upper 4bits when receiving.

Note 4: Transmitting starts at the lowest address. Received data are also stored starting from the lowest address to the highest address. For example, in the case of SIO0, the first buffer address transmitted is 0FE0$_H$.

Note 5: The value to be loaded to BUF is held after transfer is completed.

Note 6: Set the SIOS to "0" when setting the transfer mode or serial clock.

*Note 7: *; don't care*

Note 8: WAIT is valid only in the 8-bit transmit / receive and 8-bit receive modes.

Figure 2-66. SIO Control Register

(1) Serial clock

a. Clock Source

SCK (bits 2 to 0 in SIO0CR1/SIO1CR1) is able to select the following:

① Internal clock

Any of four frequencies can be selected. The serial clock is output to the outside on the $\overline{SCK0}$ / $\overline{SCK1}$ pin. The $\overline{SCK0}$ and $\overline{SCK1}$ pins goes high when transfer starts.

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

Table 2-15. Serial Clock Rate

Serial clock			Maximum transfer rate	
NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode		
DV7CK = 0	DV7CK = 1		At fc = 8 MHz	At fs = 32.768 kHz
$fc / 2^{13}$ [Hz]	$fs / 2^5$ [Hz]	$fs / 2^5$ [Hz]	0.95 Kbit/s	1 Kbit/s
$fc / 2^8$	$fc / 2^8$	–	30.5	
$fc / 2^6$	$fc / 2^6$	–	122	
$fc / 2^5$	$fc / 2^5$	–	244	

Note: 1 Kbit = 1024 bit

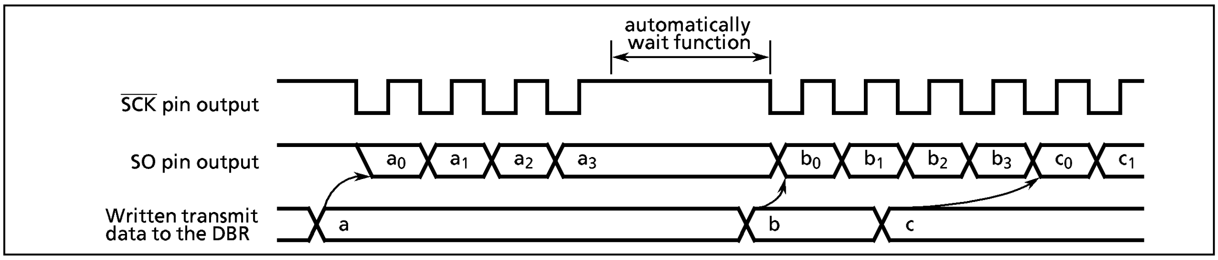
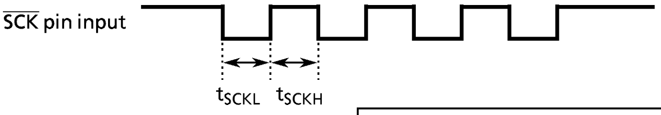

automatically wait function
$\overline{SCK}$ pin output
SO pin output
a0 a1 a2 a3 b0 b1 b2 b3 c0 c1
Written transmit data to the DBR
a
b
c

Figure 2-67. Clock Source (Internal Clock)

② External clock

An external clock connected to the $\overline{SCK0}$ / $\overline{SCK1}$ pins are used as the serial clock. In this case, the P14 ($\overline{SCK0}$) / P10 ($\overline{SCK1}$)output latch must be set to "1". To ensure shifting, a pulse width of at least 4 machine cycles is required. Thus, the maximum transfer speed is 244K-bit/s. (at fc = 8MHz).



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

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

b. Shift edge

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

① Leading edge

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

② Trailing edge

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

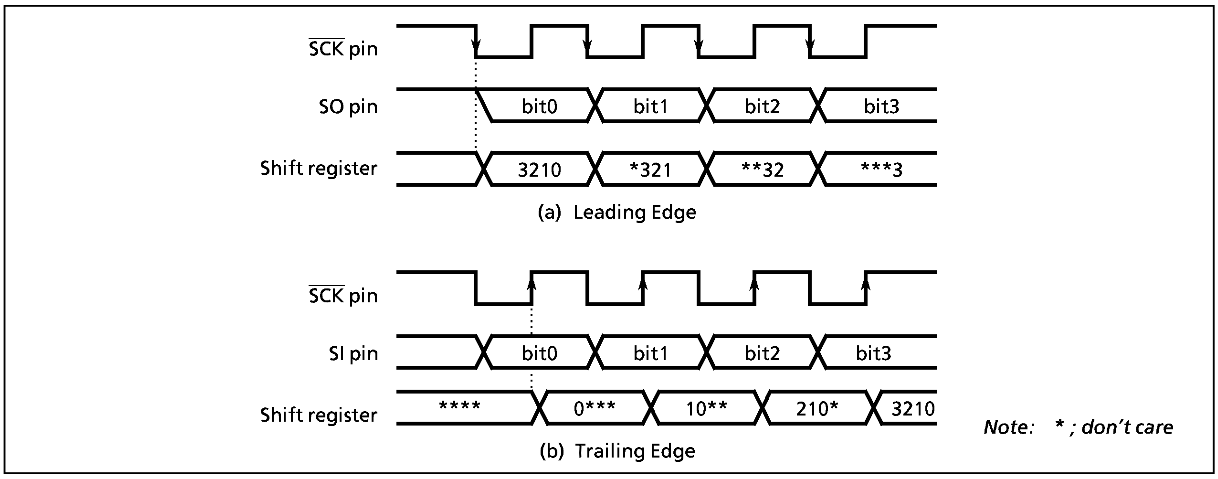


Figure 2-68. Shift Edge

(2) Number of bits to transfer

Either 4-bit or 8-bit serial transfer can be selected. When 4-bit serial transfer is selected, only the lower 4 bits of the transmit/receive data buffer register are used. The upper 4 bits are cleared to “0” when receiving.

The data is transferred in sequence starting at the least significant bit (LSB).

(3) Number of words to transfer

Up to 8 words consisting of 4 bits of data (4-bit serial transfer) or 8 bits (8-bit serial transfer) of data can be transferred continuously. The number of words to be transferred is loaded to BUF in SIO0CR2 and SIO1CR2.

An INTSIO interrupt is generated when the specified number of words has been transferred. If the number of words is to be changed during transfer, the serial interface must be stopped before making the change. The number of words can be changed during automatic-wait operation of an internal clock. In this case, the serial interface is not required to be stopped.

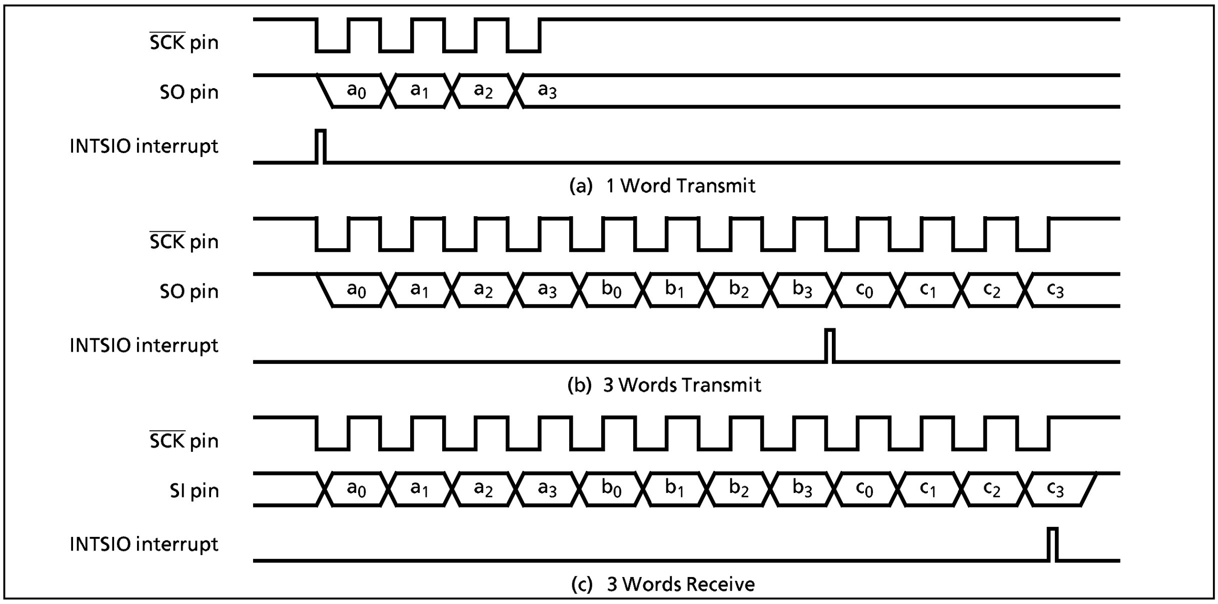


Figure 2-69. Number of Bits to Transfer (Example: 4-bit serial transfer)

2.11.3 Transfer mode

SIOM (bits 5 - 3 in SIO0CR1/SIO1CR1) is used to select the transmit, receive, or transmit/receive mode.

(1) **4-bit and 8-bit transmit** modes

In these modes, the SIO0CR1/SIO1CR1 is set to the transmit mode and then the data to be transmitted first are written to the data buffer registers (DBR). After the data are written, the transmission is started by setting SIOS to "1". The data are then output sequentially to the SO pin in synchronous with the serial clock, starting with the least significant bit (LSB). As soon as the LSB has been output, the data are transferred from the data buffer register to the shift register. When the final data bit has been transferred and the data buffer register is empty, an INTSIO (buffer empty) interrupt is generated to request the next transmitted data.

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

Note: Waits are also canceled by writing to a DBR not being used as a transmit data buffer register; therefore, during SIO do not use such DBR for other applications.
For example (SIO1), when 3words are transmitted, do not use the DBR of the remained 5words.

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

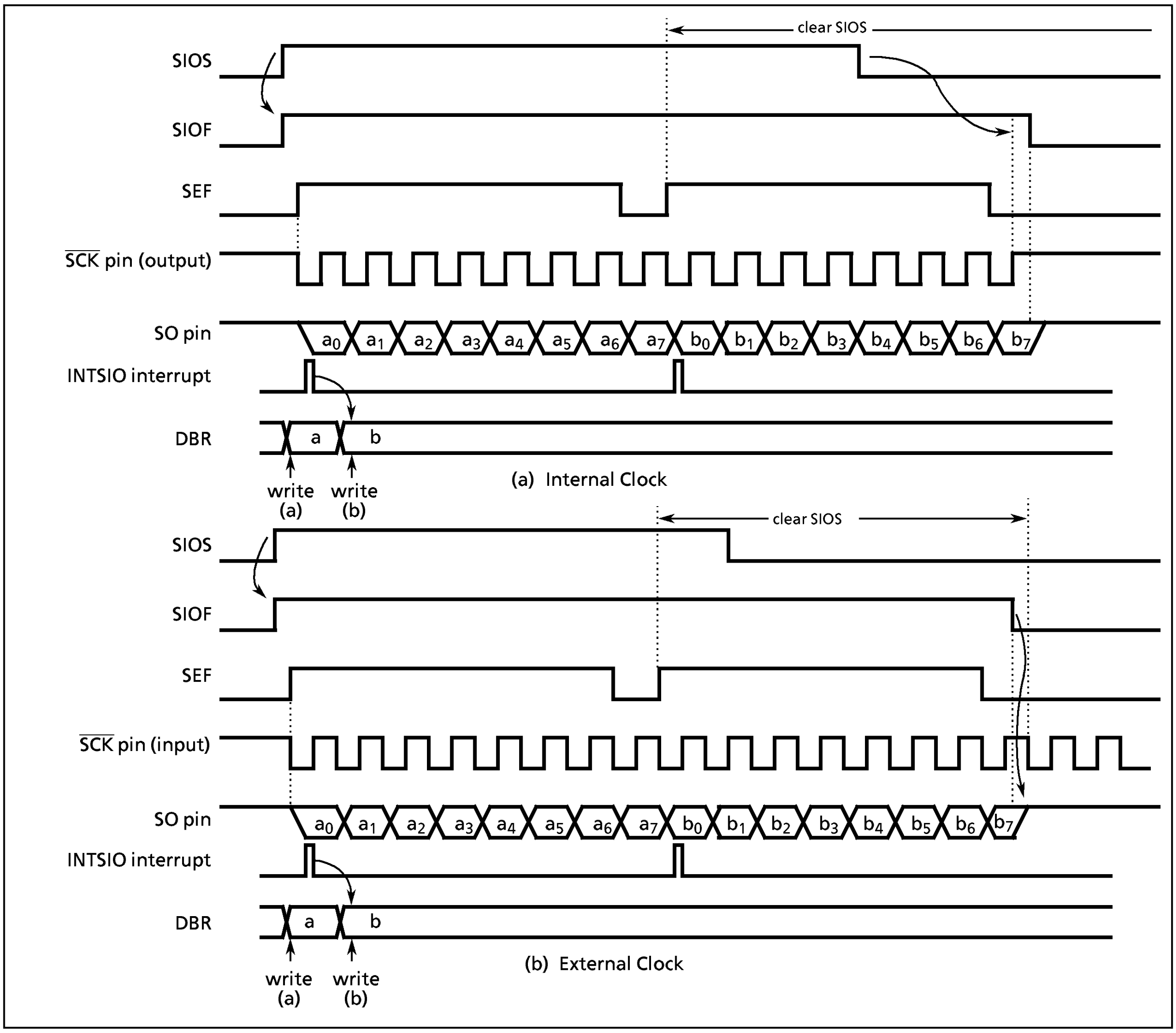
When the transmit is started, after the SIOF goes "1" output from the SO pin holds final bit of the last data until falling edge of the $\overline{\text{SCK}}$.

The transmission is ended by clearing SIOS to "0" or setting SIOINH to "1" in buffer empty interrupt service program. That the transmission has ended can be determined from the status of SIOF (bit 7 in SIO0SR/SIO1SR) because SIOF is cleared to "0" when a transfer is completed.

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

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

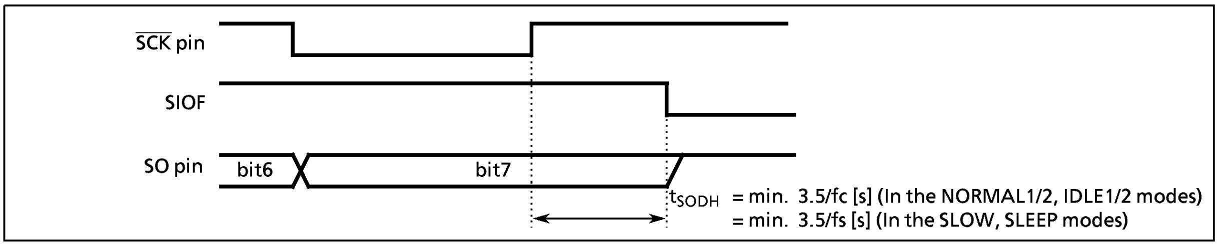
If it is necessary to change the number of words, SIOS should be cleared to "0", then BUF must be rewritten after confirming that SIOF has been cleared to "0".



(a) Internal Clock

(b) External Clock

Figure 2-70. Transfer Mode (Example: 8-bit, 1 Word Transfer)



t_{SODH} = min. 3.5/fc [s] (In the NORMAL1/2, IDLE1/2 modes)
= min. 3.5/fs [s] (In the SLOW, SLEEP modes)

Figure 2-71. Transmitted Data Hold Time at End of Transmit

(2) **4-bit and 8-bit receive** modes

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

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

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

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

When the transmit is started, after the SIOF goes "1" output from the SO pin holds final bit of the last data until falling edge of the $\overline{\text{SCK}}$.

The receiving is ended by clearing SIOS to "0" or setting SIOINH to "1" in buffer full interrupt service program. When SIOINH is set, the receiving is immediately ended SIOF is cleared to "0" receiving. When SIOS is cleared, the current data are transferred to the buffer in 4-bit or 8-bit blocks. The receiving mode ends when the transfer is completed. SIOF is cleared to "0" when receiving is ended and thus can be sensed by program to confirm that receiving has ended. If it is necessary to change the number of words in external clock operation, SIOS should be cleared to "0" then BUF must be rewritten after confirming that SIOF has been cleared to "0".

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

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

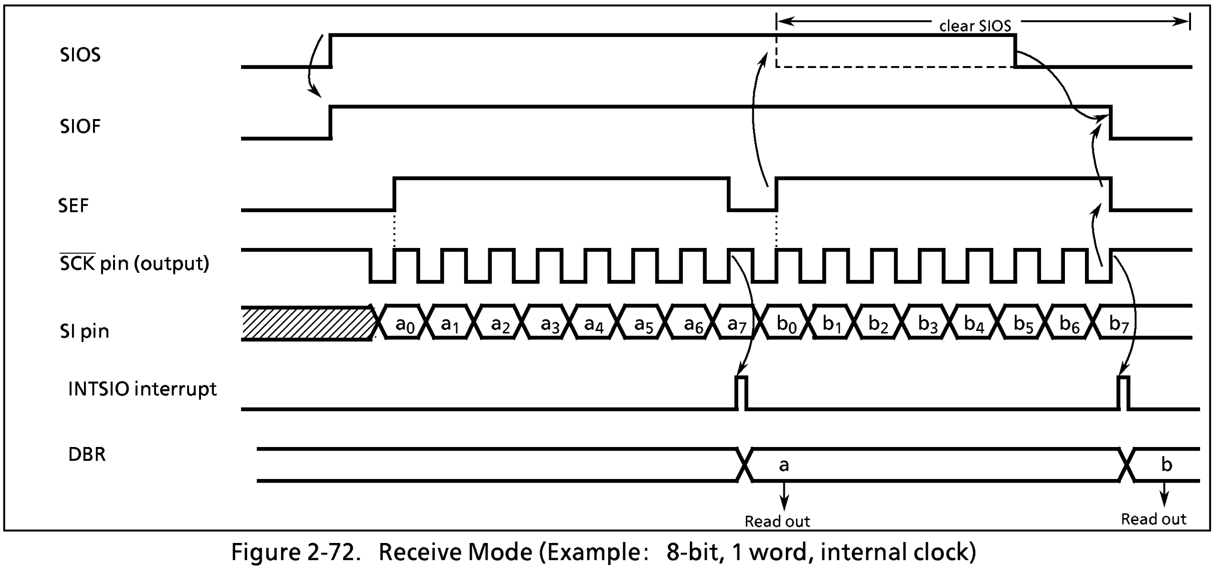

SIOS
SIOF
SEF
$\overline{\text{SCK}}$ pin (output)
SI pin
INTSIO interrupt
DBR
clear SIOS
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7
a
b
Read out
Read out

Figure 2-72. Receive Mode (Example: 8-bit, 1 word, internal clock)

(3) **8-bit transmit/receive** mode

After setting the control registers to the 8-bit transmit/receive mode, write the data to be transmitted first to the data buffer registers (DBR). After that, enable transceiving by setting SIOS to "1". When transmitting, the data are output from the SO pin at leading edges of the serial clock. When receiving, the data are input to the SI pin at the trailing edges of the serial clock. 8-bit data are transferred from the shift register to the data buffer register. An INTSIO interrupt is generated when the number of data words specified with the BUF has been transferred. The interrupt service program reads the received data from the data buffer register and then writes the data to be transmitted. The data buffer register is used for both transmitting and receiving; therefore, always write the data to be transmitted after reading the received data.

When the internal clock is used, a wait is initiated until the received data are read and the next data are written. A wait will not be initialed if even one data ward has been written.

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

When an external clock is used, the shift operation is synchronized with the external clock; therefore, it is necessary to read the received data and write the data to be transmitted next before starting the next shift operation. When an external clock is used, the transfer speed is determined by the maximum delay between generation of an interrupt request and the received data are read and the data to be transmitted next are written.

The transmit/receive operation is ended by clearing SIOS to "0" or setting SIO1NH to "1" in interrupt service program. When SIOS is cleared, the current data are transferred to the data buffer register in 8-bit blocks. The transmit mode ends when the transfer is completed. SIOF is cleared to "0" when receiving is ended and thus can be sensed by program to confirm that receiving has ended.

When SIOINH is set, the transmit/receive operation is immediately ended and SIOF is cleared to "0".

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

If it is necessary to change the number of words in internal clock, during automatic-wait which operation which occurs after completion of transmit/receive operation, BUF must be rewritten before reading and writing of the receive/transmit data.

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

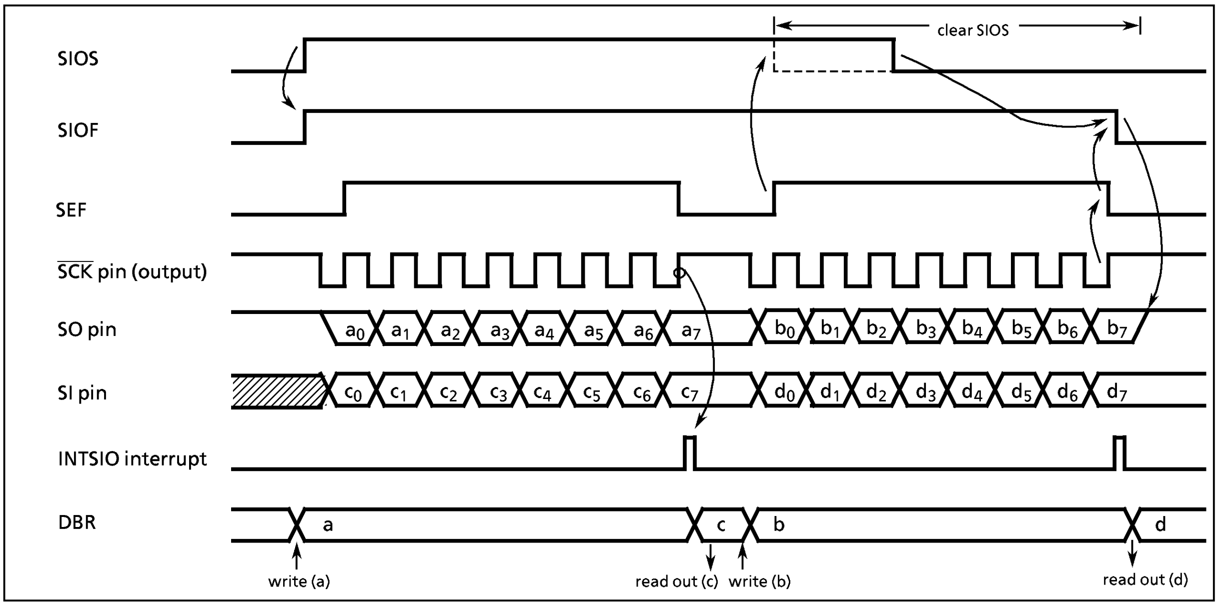
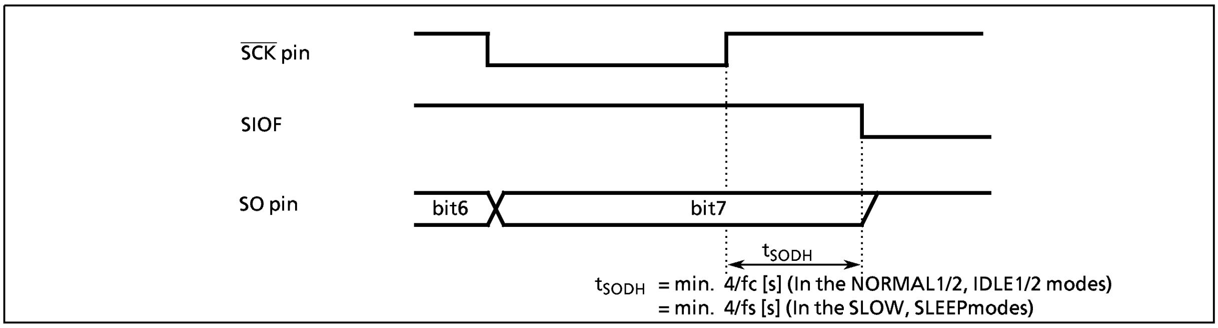

clear SIOS
SIOS
SIOF
SEF
$\overline{SCK}$ pin (output)
SO pin
SI pin
INTSIO interrupt
DBR
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7
b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7
c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7
d_0 d_1 d_2 d_3 d_4 d_5 d_6 d_7
a
c
b
d
write (a)
read out (c)
write (b)
read out (d)

Figure 2-73. Transmit/Receive Mode (Example: 8-bit, 1word, internal clock)



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

Figure 2-74. Transmitted Data Hold Time at End of Transmit/Receive

2.12 8-bit AD Converter (ADC)

The 88CP77/S77/U77 each have an 8-channel multiplexed-input 8-bit successive approximate type AD converter with sample and hold.

2.12.1 Configuration

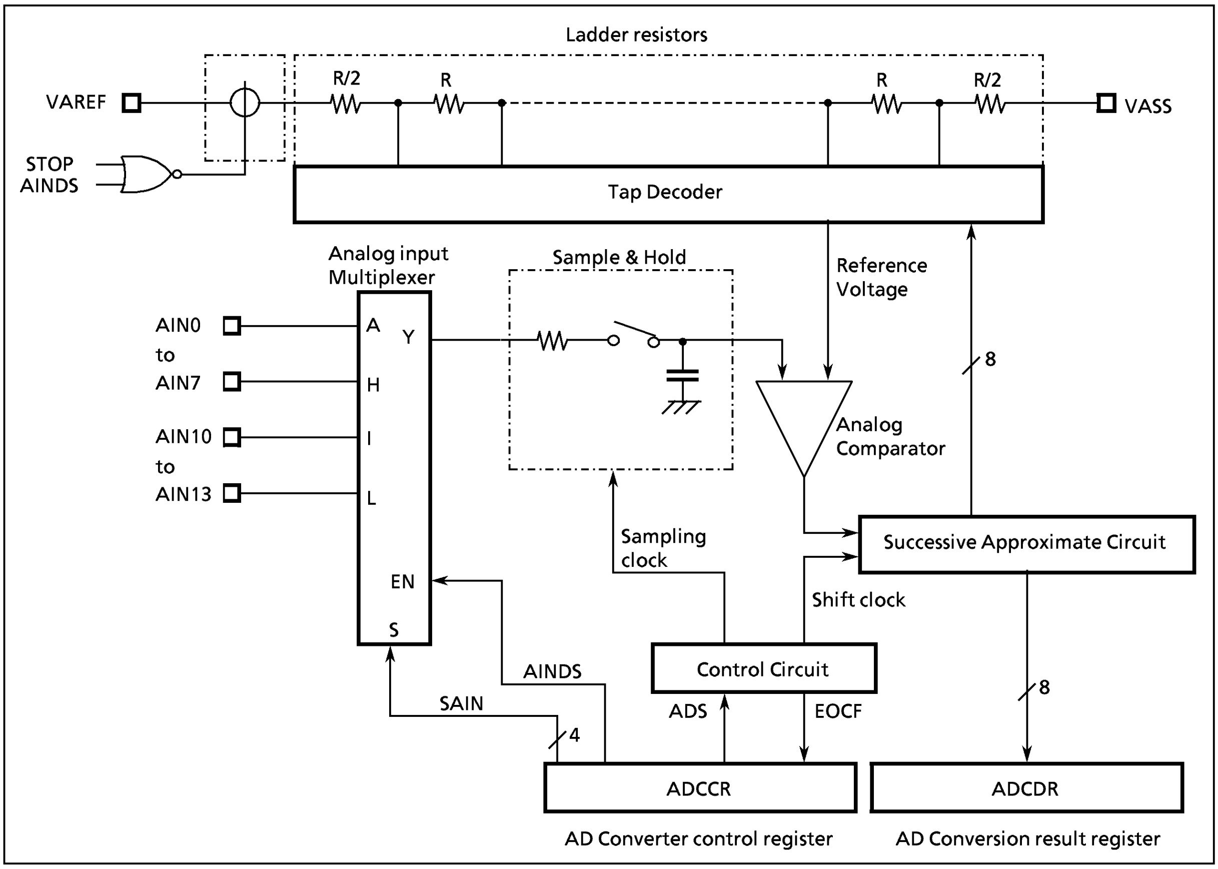


Figure 2-75. AD Converter

2.12.2 Control

The AD converter is controlled by the AD converter control register (ADCCR).

The operating state of the AD converter is confirmed by reading EOCF in ADCCR. The AD conversion value is confirmed by reading the contents of AD conversion value registers.

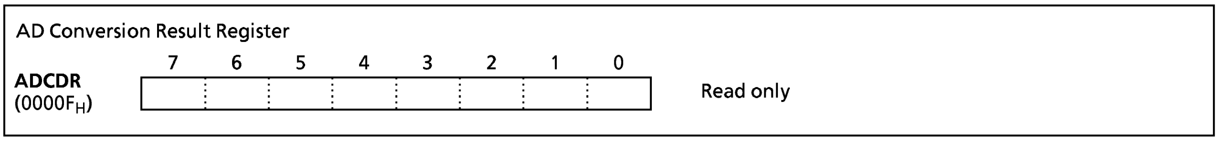


Figure 2-76. AD Conversion Result Register

AD Converter Control Register

ADCCR (0000E$_H$) (Initial value: 0000 0000)

7	6	5	4	3	2	1	0
EOCF	ADS	ACK	AINDS	SAIN			

SAIN	Analog input selection	0000: AIN0 select 0001: AIN1 select 0010: AIN2 select 0011: AIN3 select 0100: AIN4 select 0101: AIN5 select 0110: AIN6 select 0111: AIN7 select 1000: AIN10 select 1001: AIN11 select 1010: AIN12 select 1011: AIN13 select 1100: reserved 1101: reserved 1110: reserved 1111: reserved	R/W
AINDS	Analog input control	0: Enable 1: Disable	
ACK	Conversion time selection	0: 184/fc [s]: 14.72 μs (fc = 12.5 MHz) 1: 736/fc [s]: 58.88 μs (fc = 12.5 MHz)	
ADS	AD conversion start	0: – 1: AD conversion start	
EOCF	End of AD conversion flag	0: Under conversion or Before conversion 1: End of conversion	R

Note 1: Select analog input when AD converter stops.
Note 2: The ADS is automatically cleared to "0" after starting conversion.
Note 3: The EOCF is cleared to "0" when reading the ADCDR.
Note 4: The EOCF is read-only.

Figure 2-77. AD converter control register and AD conversion result register

2.12.3 Operation

Analog reference voltage on high side is applied to the VAREF pin ; on the low side, to the VASS pin. The reference voltage between VAREF and VASS is divided a ladder resistor and compared with the analog voltage input for AD conversion.

(1) Start of AD conversion

First, set the corressponding P4CR and P5CR bit to "0" for analog input. Clear the AINDS (bit 4 in ADCCR) to "0" and select one of twelve analog input AIN13-AIN0 with the SAIN (bits 3-0 in ADCCR).

Note: The pin that is not used as an analog input can be used as regular input/output pins. During conversion, do not perform output instruction to maintain a precision for all of the pins.

AD conversion is started by setting the ADS (bit 6 in ADCCR) to "1".
Conversion is accomplished in 46 machine cycles (184/fc [s]).
The EOCF (bit 7 in ADCCR) is set to "1" at end of conversion.
When setting the ADS to "1" under AD conversion, the AD converter circuit is initialized and the AD conversion try again from start.
The sampling of the analog input voltage is excuted at 4 machine cycles after setting the ADS to "1".

Note: The circuit of sample and hold is included in a capacitor of (12 pF (try.)) through a register (5 kΩ (try)). Therefore, until 4 machine cycles is over, this capacitor must be charged.

(2) Reading of AD conversion result

After the end of conversion, read the conversion result from the ADCDR.
The EOCF is automatically cleared to "0" when reading the ADCDR.

(3) AD conversion in STOP mode

When the MCU places in the STOP mode during the AD conversion, the conversion is terminated and the ADCDR contents become indefinite.

However, if the STOP mode is started after the end of conversion (EOCF = 1), the ADCDR contents are held.

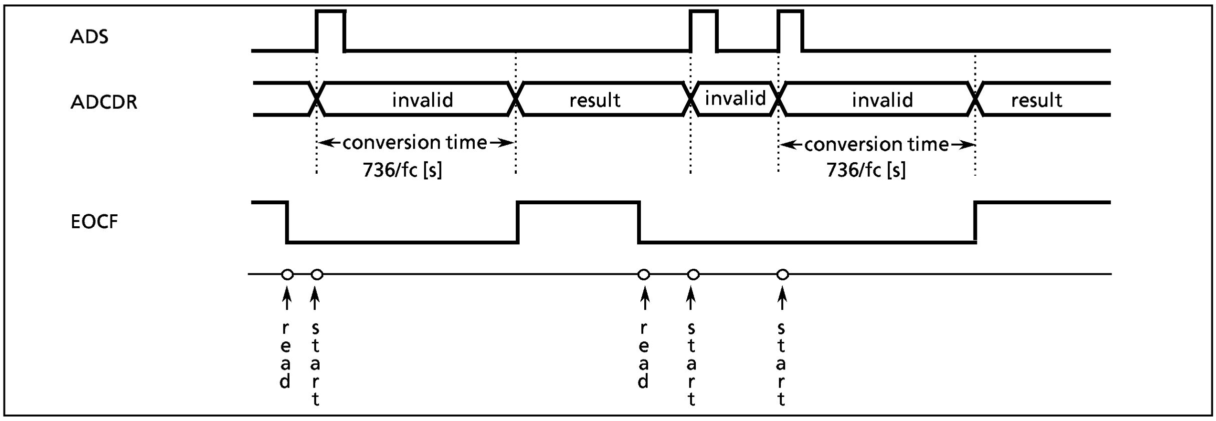


Figure 2-78. AD Conversion Timing Chart (ACK = 1)

Example:

```
           ; AIN SELECT
           LD      (ADCCR) , 00100100B   ;  Selects AIN4, ACK = 1
           ; AD CONVERT START
           SET     (ADCCR) . 6           ;  ADS = 1
SLOOP:     TEST    (ADCCR) . 7           ;  EOCF = 1 ?
           JRS     T, SLOOP
           ; RESULT DATA READ
           LD      (9EH), (ADCDR)
```

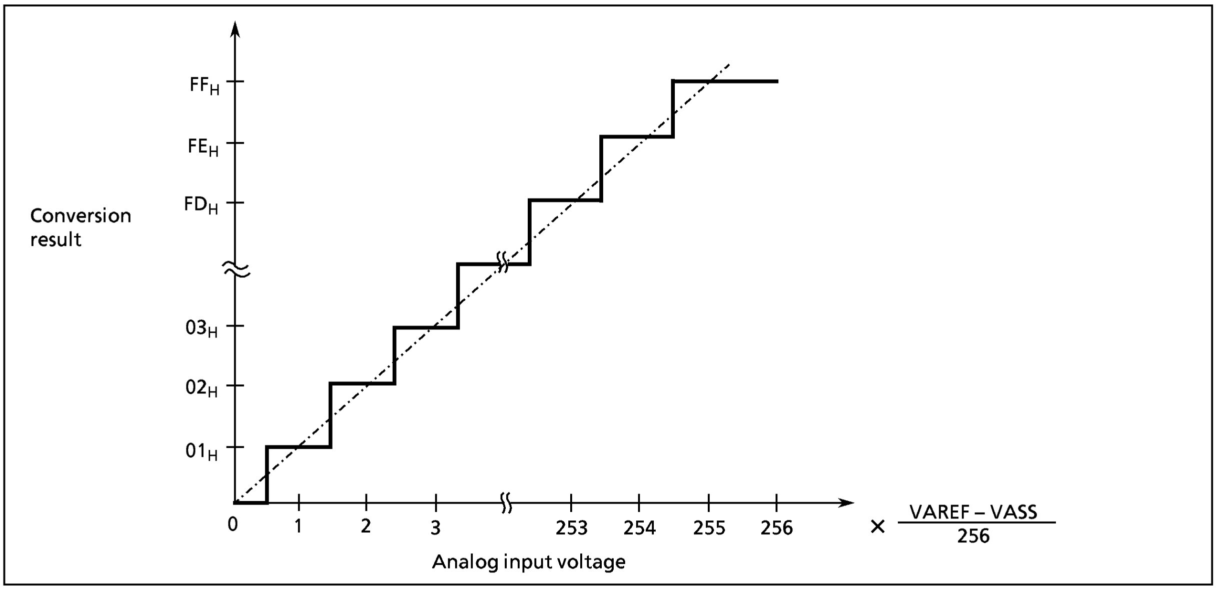


Figure2-79. Analog Input Voltage vs AD Conversion Result (typ.)

2.13 Vacuum Fluorescent Tube (VFT) Driver Circuit

The 88CP77/S77/U77/U77 features built-in high-breakdown voltage output buffers for directly driving fluorescent tubes, and a display control circuit used to automatically transfer display data to the output port.

2.13.1 Functions

(1) 53 high-breakdown voltage output buffers built-in.

- Large current output pin (V0 to V17)
- Middle current output pin (V18 to V52)

There is also the VKK pin used for the VFT drive power supply.

(2) The dynamic lighting system makes it possible to select 1 to 18 digits (T0 to T17) by program.

(3) Pins not used for VFT driver can be used as general-purpose ports.

- Pins can be selected using the VSEL (bits 4 to 0) in VFT control register1 bit by bit.

(4) Display data (126 bytes in DBR) are automatically transferred to the VFT output pin.

(5) Brightness level can be adjusted in 7 steps using the dimmer function.

(6) Table 2-16. shown in setting of display time.

Table 2-16. Setting of display time

DV1CK = 0 [s]	DV1CK = 1 [s]
2^{10}/fc	2^{10}/fc
2^{11}/fc	2^{11}/fc
2^{12}/fc	2^{12}/fc
2^{13}/fc	2^{13}/fc

2.13.2 Configuration

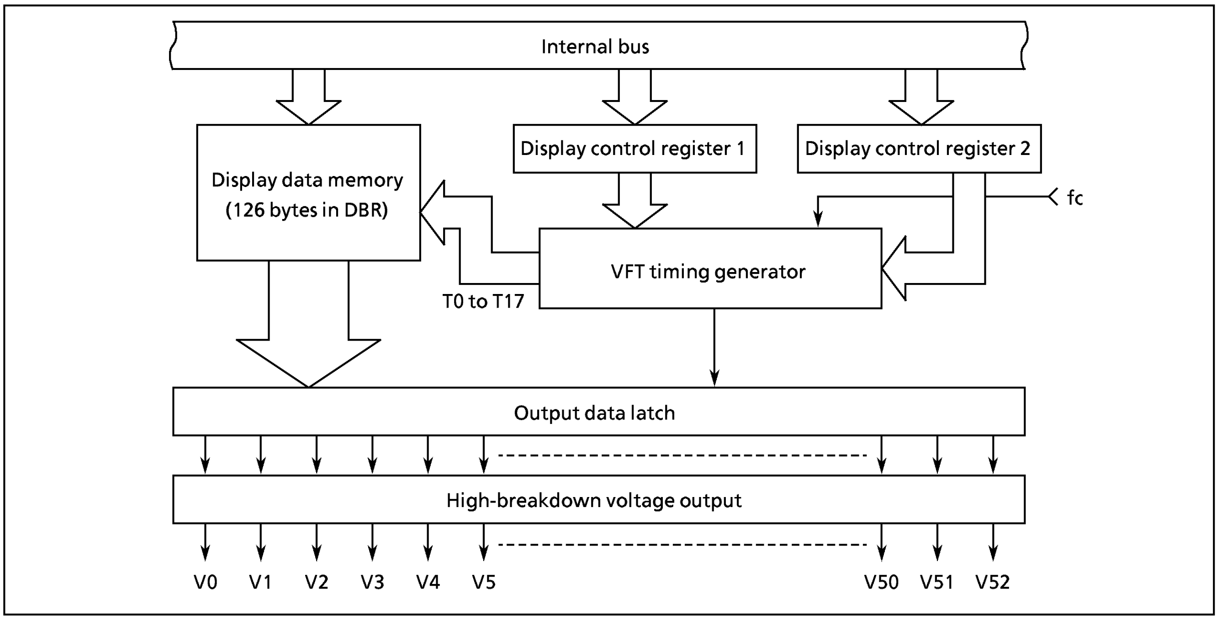


Figure 2-80. Vacuum Fluorescent Tube (VFT) Driver Circuit

2.13.3 Control

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

Switching the mode from NORMAL1/2 to SLOW or STOP puts the VFT driver circuit into blanking state (BLK is set to "1"; values set in the VFT control registers except BLK and EKEY are maintained), and sets segment outputs and digit outputs are cleared to "0". Thus, ports P6 to P9, and PD to PF function as general-purpose output ports with pull-down.010000

VFT control register 1

	7	6	5	4	3	2	1	0	
VFTCR1 (0002A$_H$)	BLK	SDT		VSEL					(initial value: 0000 0000)

BLK	VFT display control	0: Display enable 1: Disable		write only
SDT	Display time select (tdisp) (Display time of 1 digit)		DV1CK = 0 [s]	DV1CK = 1 [s]
		00	2^{10}/fc	2^{10}/fc
		01	2^{11}/fc	2^{11}/fc
		10	2^{12}/fc	2^{12}/fc
		11	2^{13}/fc	2^{13}/fc
VSEL	Automatic display select (When using VFT driver (automatic display), V31 to V0 are only used to output VFT.) Pins which are not selected by the output pins other than the above-mentioned pins can be used as general-purpose input/output pins. (When using as a general-purpose input/output pin, the display data which corresponds to the pin must be set to "0")	00000: 32 (V31 to V0) 00001: 33 (V32 to V0) 00010: 34 (V33 to V0) 00011: 35 (V34 to V0) 00100: 36 (V35 to V0) 00101: 37 (V36 to V0) 00110: 38 (V37 to V0) 00111: 39 (V38 to V0) 01000: 40 (V39 to V0) 01001: 41 (V40 to V0) 01010: 42 (V41 to V0)	01011: 43 (V42 to V0) 01100: 44 (V43 to V0) 01101: 45 (V44 to V0) 01110: 46 (V45 to V0) 01111: 47 (V46 to V0) 10000: 48 (V47 to V0) 10001: 49 (V48 to V0) 10010: 50 (V49 to V0) 10011: 51 (V50 to V0) 10100: 52 (V51 to V0) 10101: 53 (V52 to V0) others: reserved	

Note 1: fc ; high frequency clock

Note 2: VFTCR1 is write-only register, which cannot use any of in read-modify-write instruction such as bit operate, etc.

Note 3: During display enable, when display time (SDT) or automatic display (VSEL) is changed, BLK is should be set to "1".

Note 4: reserved ; can not be accessed.

	7	6	5	4	3	2	1	0
VFTSR (0002A$_H$)	WAIT							

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

Note 1: During reset, WAIT is initialized to "1".

Note 2: During display disable, when BLK of VFTCR1 is written to "0", WAIT becomes "0" from next tdisp start timing and VFT starts display action.

Note 3: During display enable, when BLK of VFTCR1 is written to "1", after having finished timing display of present, VFT display becomes disable.

Note 4: After display disable, display enable setting to VFTCR1 needs to set after validation of WAIT = "1".

Figure 2-81. VFT Control Register 1, VFT status register

VFT control register 2

VFTCR2 (0002B$_H$)	7	6	5	4	3	2	1	0
	DIM			STA				

(initial value: 0000 0000)

DIM	Dimmer time select	000: reserved 001: (14/16) × tdisp (s) 010: (12/16) × tdisp (s) 011: (10/16) × tdisp (s) 100: (8/16) × tdisp (s) 101: (6/16) × tdisp (s) 110: (4/16) × tdisp (s) 111: (2/16) × tdisp (s)	write only
STA	Number of state (display)	00000: 1 display mode (T0) 00001: 2 display mode (T1 to T0) 00010: 3 display mode (T2 to T0) 00011: 4 display mode (T3 to T0) 00100: 5 display mode (T4 to T0) 00101: 6 display mode (T5 to T0) 00110: 7 display mode (T6 to T0) 00111: 8 display mode (T7 to T0) 01000: 9 display mode (T8 to T0) 01001: 10 display mode (T9 to T0) 01010: 11 display mode (T10 to T0) 01011: 12 display mode (T11 to T0) 01100: 13 display mode (T12 to T0) 01101: 14 display mode (T13 to T0) 01110: 15 display mode (T14 to T0) 01111: 16 display mode (T15 to T0) 10000: 17 display mode (T16 to T0) 10001: 18 display mode (T17 to T0) others: reserved	

Note 1: VFTCR2 is write-only register, which cannot use any of in read-modify-write instruction such as bit operate, etc.

Note 2: Even if a number of the display digit is set a pin which is equal to the digit dose not output. It is necessary to write data to the data buffer which corresponds to the digit according to the display timing (T0 to T17).

*Note 3: * ; don't care*

Note 4: DIM of VFTCR2 cannot use initial value.

Note 5: The pin which DIM is valid, it is assigned from P60 to P81 by setting of STA.

Figure 2-82. VFT control Register 2

(1) Setting of display mode

VFT display mode is set by VFT control register 1 (VFTCR1) and VFT control register 2 (VFTCR2). VFT control register 1 (VFTCR1) sets 1 display time (tdisp) and the number of display lines (VSEL), and VFT control register 2 (VFTCR2) sets dimmer timer (DIM) and state (STA). (BLK of VFTCR1 must be set to "1".) The pin which DIM is valid, it is assigned from P60 to P81 by setting of STA.
For example, when STA is 8 display mode, the DIM output pins are assigned from P60 to P67.
Therefore, one display time of other pins are tdisp (s).
(See Display operation in section 2.13.4 for display timing and data setting procedures.)

(2) Display data setting

Data are converted into VFT display data by instructions. The converted data stored in the display data buffer (addresses 00F00 to 00FD1 in DBR) are automatically transferred to the VFT driver circuit, then transferred to the high-breakdown voltage output buffer. Thus, to change the display pattern, just change the data in the display data buffer.

Bits in the VFT segment (dot) and display data area correspond one to one. When data are set to 1, the segments corresponding to the bits light. The display data buffer is assigned to the DBR area shown in Figure 2-83. (The display data buffer can not be used as data memory)

bit	0 to 7	0 to 7	0 to 7	0 to 7	0 to 7	0 to 7	0 to 4	Timing
	00F00	00F20	00F40	00F60	00F80	00FA0	00FC0	T0
	00F01	00F21	00F41	00F61	00F81	00FA1	00FC1	T1
	00F02	00F22	00F42	00F62	00F82	00FA2	00FC2	T2
	00F03	00F23	00F43	00F63	00F83	00FA3	00FC3	T3
	00F04	00F24	00F44	00F64	00F84	00FA4	00FC4	T4
	00F05	00F25	00F45	00F65	00F85	00FA5	00FC5	T5
	00F06	00F26	00F46	00F66	00F86	00FA6	00FC6	T6
	00F07	00F27	00F47	00F67	00F87	00FA7	00FC7	T7
	00F08	00F28	00F48	00F68	00F88	00FA8	00FC8	T8
	00F09	00F29	00F49	00F69	00F89	00FA9	00FC9	T9
	00F0A	00F2A	00F4A	00F6A	00F8A	00FAA	00FCA	T10
	00F0B	00F2B	00F4B	00F6B	00F8B	00FAB	00FCB	T11
	00F0C	00F2C	00F4C	00F6C	00F8C	00FAC	00FCC	T12
	00F0D	00F2D	00F4D	00F6D	00F8D	00FAD	00FCD	T13
	00F0E	00F2E	00F4E	00F6E	00F8E	00FAE	00FCE	T14
	00F0F	00F2F	00F4F	00F6F	00F8F	00FAF	00FCF	T15
	00F10	00F30	00F50	00F70	00F90	00FB0	00FD0	T16
	00F11	00F31	00F51	00F71	00F91	00FB1	00FD1	T17
output	V0 to V7	V8 to V15	V16 to V23	V24 to V31	V32 to V39	V40 to V47	V48 to V52	

Note: Writing "0" in 7 to 5 bit of address 00FC0$_H$ to 00FDI$_H$ in DBR.

Figure 2-83. VFT Display Data Buffer Memory (DBR)

Note: The data memory contents become unstable when the power supply is turned on ; therefore, the data memory should be initialized.

2.13.4 Display operation

After setting of the display timing for the number of digits according to the using VFT and storing the segment and digit data according to the respective timings, clearing BLK in VFTCR1 to 0 starts VFT display.

Figure 2-84. shows the VFT drive pulse and Figure 2-85, 86 show the display operation.

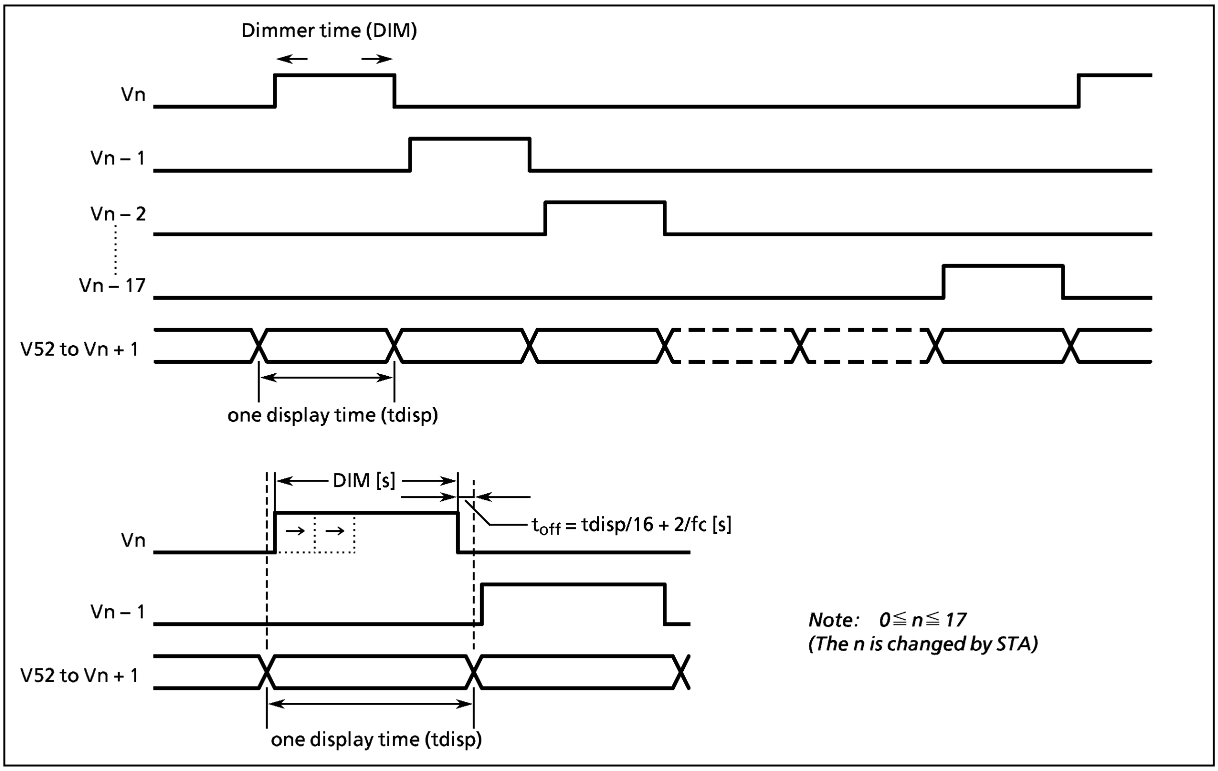
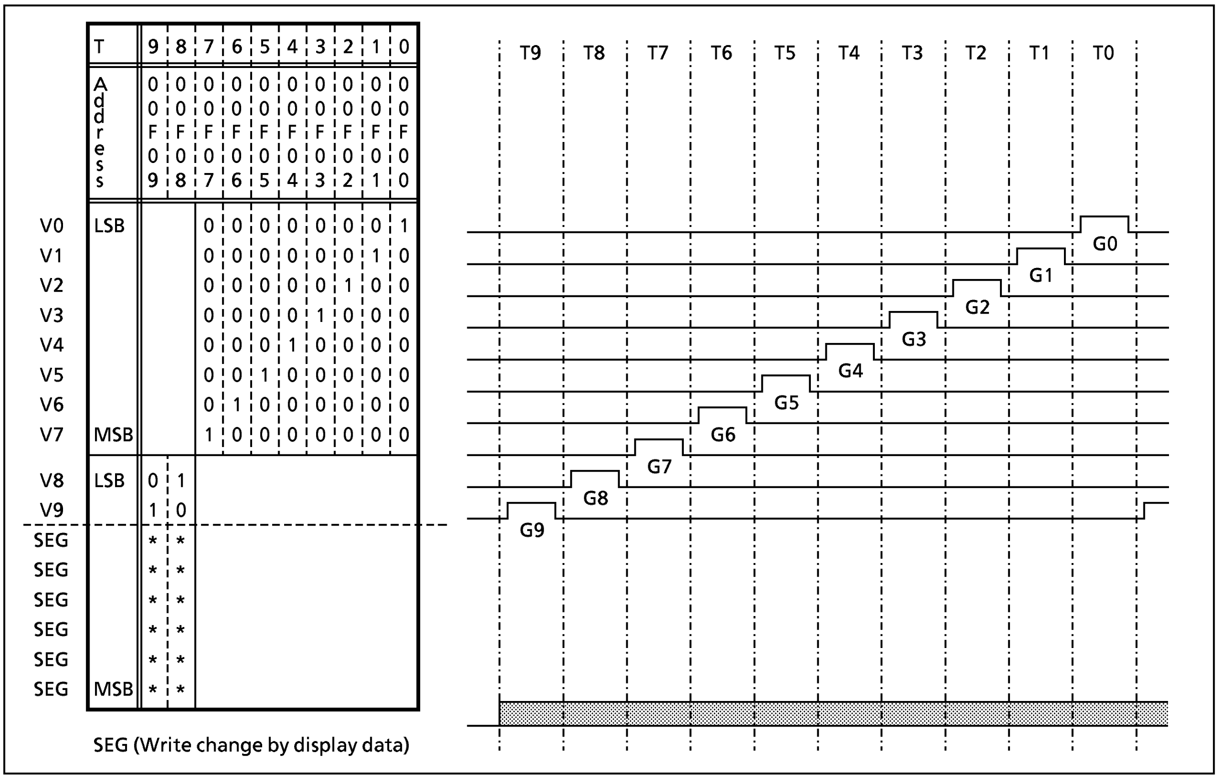


Figure 2-84. VFT Drive Waveform and Display timing

2.13.5 Example of display operation

(1) For Conventional type VFT

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



	T	9	8	7	6	5	4	3	2	1	0
	Address	00F09	00F08	00F07	00F06	00F05	00F04	00F03	00F02	00F01	00F00
V0	LSB			0	0	0	0	0	0	0	1
V1				0	0	0	0	0	0	1	0
V2				0	0	0	0	0	1	0	0
V3				0	0	0	0	1	0	0	0
V4				0	0	0	1	0	0	0	0
V5				0	0	1	0	0	0	0	0
V6				0	1	0	0	0	0	0	0
V7	MSB			1	0	0	0	0	0	0	0
V8	LSB	0	1								
V9		1	0								
SEG		*	*								
SEG		*	*								
SEG		*	*								
SEG		*	*								
SEG		*	*								
SEG	MSB	*	*								

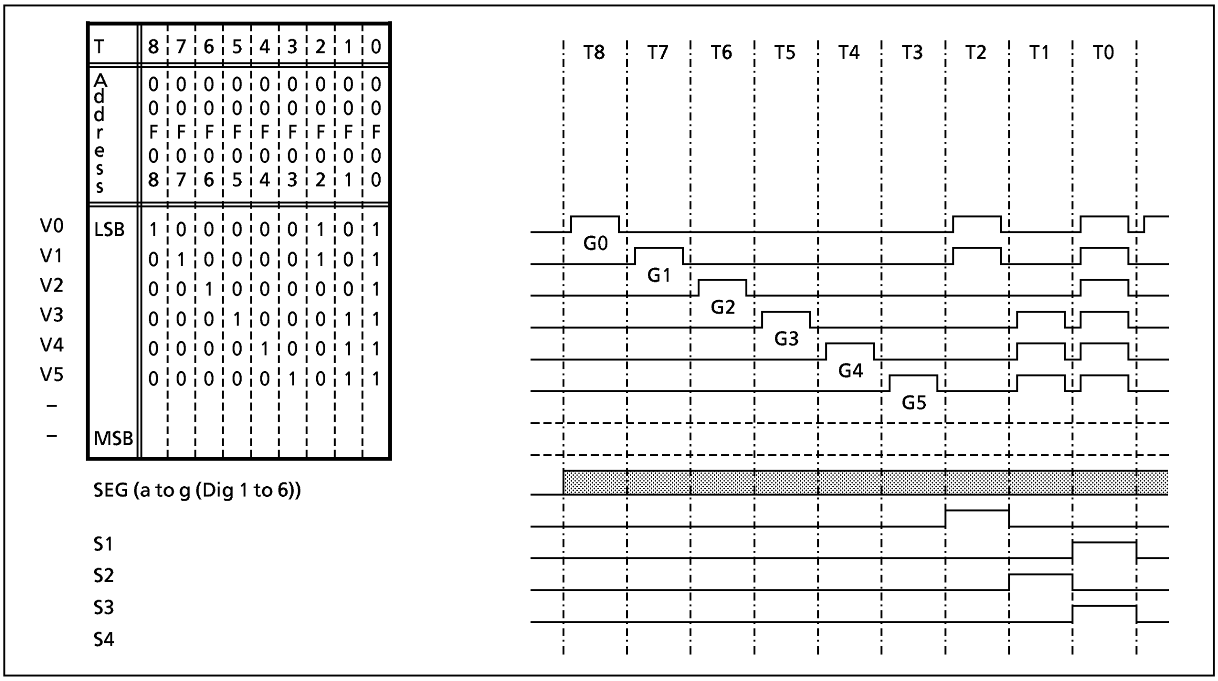
SEG (Write change by display data)

Figure 2-85. Example of Conventional type VFT driver pulse

(2) For Grid scan type VFT

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

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



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

Figure 2-86. Grid Scan Type Display Vacuum Fluorescent Tube Ware

2.13.6 Port function

(1) High-breakdown voltage buffer

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

It is recommended that ports P6, P7, P8 and P9 should be used as VFT driver output. Precaution for using as general-purpose I/O pins are follows.

> *Note: When not using a pin which is pulled down to pin V_{KK} (RK = typ. 80 kΩ), it must be set to open. It is necessary to clear the port output latch and the data buffer memory (DBR) to "0".*

① Ports P6 to P9

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

② Port PD to PF

VFT output and usual input/output are controlled by VSEL of VFT control register in bits. When a pin which is pulled down to pin V_{KK} is used as usual output or input, the following cautions are required.

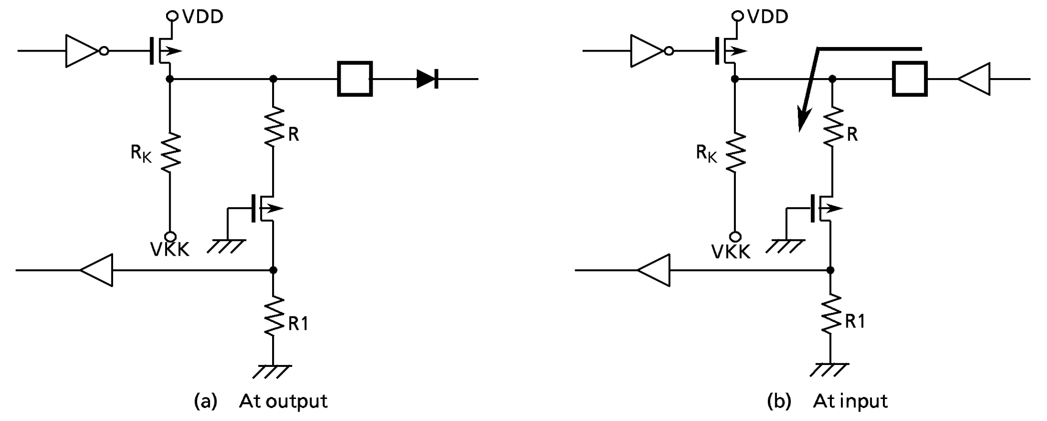
(a) When outputting

When level "L" is output, a port which is pulled down to pin V_{KK} is pin V_{KK} voltage. Such processes as clamping with the diode as shown in figure 2-87. (a) are necessary to prevent pin V_{KK} voltage applying to the external circuit.

(b) When inputting

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

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

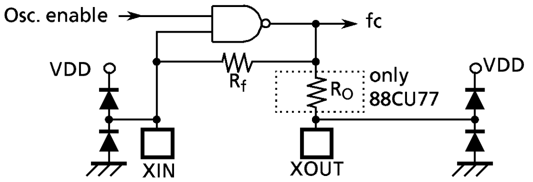
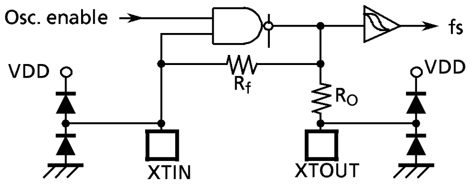
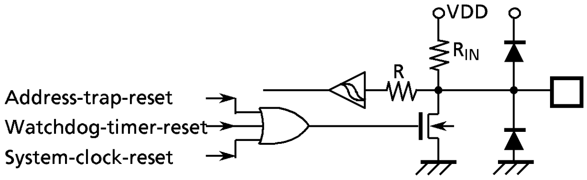
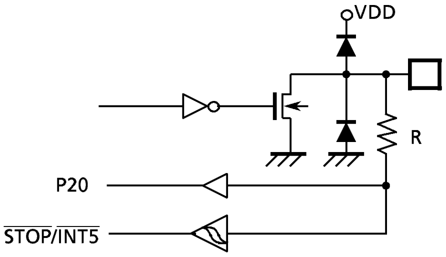
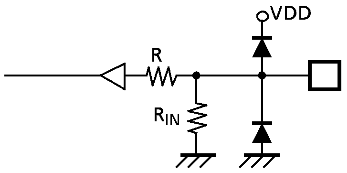


(a) At output (b) At input

Figure 2-87. External Circuit Interface

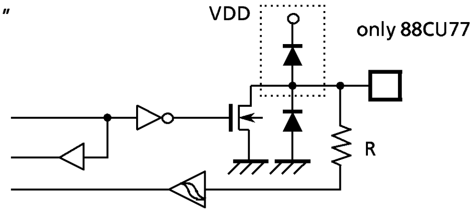
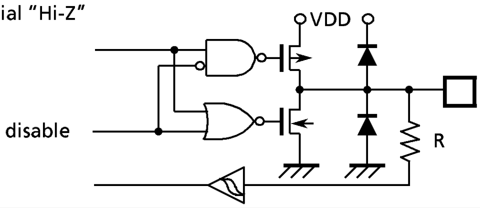
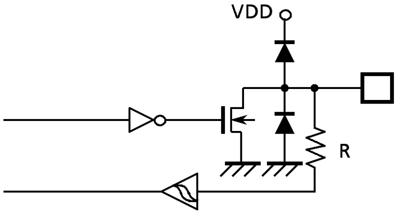
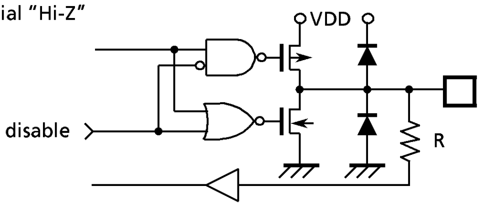
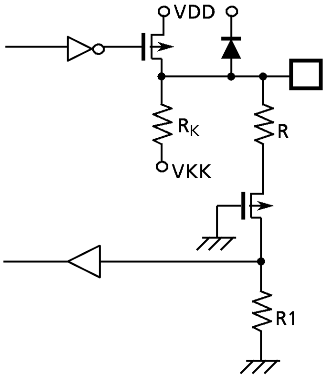
Input / Output Circuitry

(1) Control pins
The input/output circuitries of the 88CP77/S77/U77/U77 control pins are shown below.

Control Pin	I/O	Circuitry	Remarks
XIN XOUT	Input Output	 Osc. enable fc VDD R_f R_O only 88CU77 VDD XIN XOUT	Resonator connecting pins (high-frequency) R_f = 1.2 MΩ (typ.) R_O = 0.5 kΩ (typ.)
XTIN XTOUT	Input Output	 Osc. enable fs VDD R_f R_O VDD XTIN XTOUT	Resonator connecting pins (low-frequency) R_f = 6 MΩ (typ.) R_O = 220 kΩ (typ.)
RESET	I/O	 VDD R_{IN} R Address-trap-reset Watchdog-timer-reset System-clock-reset	Hysteresis input R_{IN} = 220 kΩ (typ.) R = 1 kΩ (typ.)
STOP / INT5	Input		Hysteresis input R = 1 kΩ (typ.)
TEST	Input		Pull-down resistor R_{IN} = 70 kΩ (typ.) R = 1 kΩ (typ.)

Note 1: The TEST pin of the 88PU77 does not have a pull-down resistor. Fix the test pin at low-level.
Note 2: The XOUT pin of the 88CU77 has a R_O resistor unlike the 88PU77/CS77/CP77.

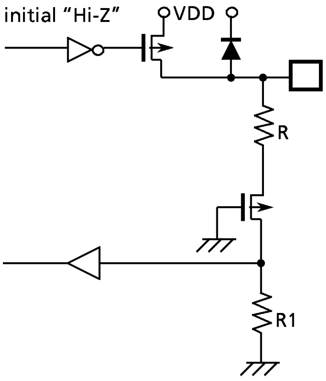
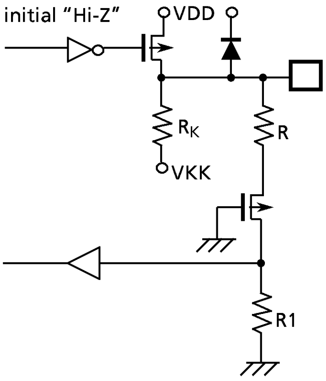
(2) - ① Input/Output Ports

Port	I/O	Input/Output Circuitry	Remarks
P0	I/O	initial “Hi-Z” 	Sink open drain output Hysteresis input R = 1 kΩ (typ.)
P1 P3	I/O	initial “Hi-Z” 	Tri-state I/O Hysteresis input R = 1 kΩ (typ.)
P2	I/O	initial “Hi-Z” 	Sink open drain output Hysteresis input R = 1 kΩ
P4 P5	I/O	initial “Hi-Z” 	Tri-state I/O R = 1 kΩ (typ.)
P6 P7 P8 P9	I/O	initial “Hi-Z” 	Source open drain I/O High-breakdown voltage R_K = 80 kΩ (typ.) R = 1 kΩ (typ.) R1 = 200 kΩ (typ.)

Note: The P0 port of 88CU77 is different from 88PU77/CS77/CP77 and has guard diodes built-in VDD side. Therefore, absolute maximum rating of input voltage is ” – 0.3 to VDD + 0.3” volts.

(2) - ② Input / output Ports

Port	I/O	Input / Output Circuitry and Code		Remarks
		A	D	Source open drain I/O

PD
PE
PF
	I/O			High-breakdown voltage
				R_K = 80 kΩ (typ.)
				R = 1 kΩ (typ.)
				R1 = 200 kΩ (typ.)

Electrical Characteristics

Absolute Maximum Ratings

(V_{SS} = 0 V)

PARAMETER	SYMBOL	PINS	RATINGS	UNIT
Supply Voltage (* Note4)	V_{DD}		– 0.3 to 6.5	V
Input Voltage	V_{IN1}	P1, P2, P3, P4, P5, XOUT, $\overline{\text{RESET}}$	– 0.3 to V_{DD} + 0.3	V
	V_{IN2}	P0 port (* Note 3)	– 0.3 to 5.5 V	
Output Voltage	V_{OUT1}	P1, P2, P3, P4, P5, XOUT, $\overline{\text{RESET}}$	– 0.3 to V_{DD} + 0.3	V
	V_{OUT2}	P0 port (* Note 3)	– 0.3 to 5.5 V	
	V_{OUT3}	Source open drain ports	V_{DD} – 40 to V_{DD} + 0.3	
Output Current (Per 1 pin)	I_{OUT1}	P0, P1, P2, P3, P4, P5 ports	3.2	mA
	I_{OUT2}	P6, P7, P80, 81 Ports	– 25	
	I_{OUT3}	P82 to P87, P9, PD, PE, PF ports	– 12	
Output Current (Total)	$\Sigma\ I_{OUT1}$	P1, P3, P4, P5 ports	– 40	mA
	$\Sigma\ I_{OUT2}$	P0, P1, P2, P3, P4, P5 ports	60	
	$\Sigma\ I_{OUT3}$	P6, P7, P8, P9, PD, PE, PF ports	– 120	
Power Dissipation [Topr = 25°C]	PD Note		1200	mW
Soldering Temperature (time)	Tsld		260 (10 s)	°C
Storage Temperature	Tstg		– 55 to + 125	°C
Operating Temperature	Topr		– 30 to + 70	°C

Note 1: The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

Note 2: Power Dissipation (PD) ; For PD, it is necessary to decrease -14.3 mw/°C.

Note 3: The P0 port of TMP88CU77 is different from TMP88PU77/CS77/CP77 and has guard diodes built-in VDD side. Therefore, absolute maximum rating of input voltage is " – 0.3 to VDD + 0.3" volts.

Note 4: All VDDs should be connected externally for keeping the same voltage level.

Recommended Operating Conditions

(V_{SS} = 0 V, Topr = – 30 to 70°C)

PARAMETER	SYMBOL	PINS	CONDITIONS		Min	Max	UNIT
Supply Voltage	V_{DD}		fc = 12.5 MHz	NORMAL 1, 2 modes	4.5	5.5	V
				IDLE1, 2 modes			
			fs = 32.768 kHz	SLOW mode			
				SLEEP mode			
				STOP mode	2.0		
Input High Voltage	V_{IH1}	Except hysteresis input	$V_{DD} \geqq 4.5$ V		$V_{DD} \times 0.70$	V_{DD}	V
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
	V_{IH3}		$V_{DD} < 4.5$ V		$V_{DD} \times 0.90$		
Input Low Voltage	V_{IL1}	Except hysteresis input	$V_{DD} \geqq 4.5$ V		0	$V_{DD} \times 0.30$	V
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
	V_{IL3}		$V_{DD} < 4.5$ V			$V_{DD} \times 0.10$	
Clock Frequency	fc	XIN, XOUT	V_{DD} = 4.5 V to 5.5 V		1.0	12.5	MHz
	fs	XTIN, XTOUT	V_{DD} = 2.7 V to 5.5 V		30.0	34.0	kHz

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

How to calculate power consumption.

With the TMP88CP77/S77/U77, a pull-down resistor (Rk = 80 kΩ typ.) can be built into a VFT driver using mask option (port by port). The share of VFT driver loss (VFT driver output loss + pull-down resistor (Rk) loss) in power consumption Pmax is high. When using a fluorescent display tube with a large number of segments, the maximum power consumption Pd must not be exceeded.

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

Where,
operating power consumption: VDD x IDD
normal power consumption: Σ Iout2 x 0.4
VFT driver loss: VFT driver output loss + pull-down resistor (Rk) loss

Example:
When Ta = 10 to 50°c and a fluorescent display tube with segment output = 3 mA, digit output = 15 mA, Vxx = -25 V is used.
Operating conditions: VDD = 5 V ± 10%, fc = 8 MHz, VFT dimmer time (DIM) = (14/16) x tseg:

Power consumption Pmax = (1) + (2) + (3)
Where, segments pin = X grid pin = Y, Y = 2
(1) Operating power consumption: $V_{DD} \times I_{DD} = 5.5\ V \times 22\ mA = 121\ mW$
(2) Normal output port loss: Σ Iout2 x 0.4 V = 60 mA × 0.4 V = 24 mW
(3) VFT driver loss: segment pin = 3 mA x 2 V x number of segments X = 6 mW x X
digit pin = 15 mA x 2 V x 14/16 (DIM) x number of gridsY
= 52.5 mW
Rk loss = $(5.5 + 25\ V)^2$ / 50 kΩ x (number of segments X + number of digits Y) = 18.605 mW x (X + 2)

Therefore, Pmax = 121 mW + 24 mW + 6 mW × X + 52.5 mW + 18.605 mWx (X + 2) = 234.71 + 24.605X

Maximum power consumption Pd when Ta = 50°C is determined by the following equation:
PD = 1200 mW-(14.3 x 25) = 842.5 mW
The number of segments X which can be lit is:
PD > Pmax
842.5 mW > 234.71 mW + 24.605 X
24.7 > X

Thus, a fluorescent display tube with less than 24 segments can be used. If a fluorescent display tube with 24 segments or more is used, either a pull-down resistor must be attached externally , or the number of segments to be lit must be kept to less than 24 by software.

D.C. Characteristics

(V_{SS} = 0 V, Topr = – 30 to 70°C)

PARAMETER	SYMBOL	PINS	CONDITIONS	Min	Typ.	Max	UNIT
Hysteresis Voltage	V_{HS}	Hysteresis input		–	0.9	–	V
Input Current	I_{IN1}	TEST	V_{DD} = 5.5 V V_{IN} = 5.5 V / 0 V	–	–	± 2	μA
	I_{IN2}	Open drain ports, Tri-state ports					
	I_{IN3}	$\overline{\text{RESET}}$, $\overline{\text{STOP}}$					
	I_{IN4}	PD, PE, PF ports (Note3)		–	–	80	
Input Resistance	R_{IN2}	RESET		100	220	450	kΩ
Pull-down Resistance	R_K	Source open drain ports	V_{DD} = 5.5 V, V_{KK} = – 30 V	50	80	110	
Output Leakage Current	I_{LO1}	Sink open drain ports	V_{DD} = 5.5 V, V_{OUT} = 5.5 V	–	–	2	μA
	I_{LO2}	Source open drain ports	V_{DD} = 5.5 V, V_{OUT} = – 32 V	–	–	– 2	
	I_{LO3}	Tri-state ports	V_{DD} = 5.5 V, V_{OUT} = 5.5 V / 0 V	–	–	2	
Output High Voltage	V_{OH2}	Tri-state ports	V_{DD} = 4.5 V, I_{OH} = – 0.7 mA	4.1	–	–	V
	V_{OH3}	P82 to P87, P9, PD, PE, PF ports	V_{DD} = 4.5 V, I_{OH} = – 8 mA	2.4	–	–	
Output Low Voltage	V_{OL}	Except XOUT	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	–	–	0.4	V
Output High current	I_{OH1}	P6, P7, P80, P81 ports	V_{DD} = 4.5 V, V_{OH} = 2.4 V	–	– 30	–	mA
	I_{OH2}	P82 to P87, P9, PD, PE, PF ports		–	– 15	–	
Supply Current in NORMAL 1, 2 modes	I_{DD}		V_{DD} = 5.5 V V_{IN} = 5.3 V / 0.2 V fc = 12.5 MHz fs = 32.768 kHz	–	15	22	mA
modesSupply Current in IDLE 1, 2 modes				–	6	12	
Supply Current in SLOW mode			V_{DD} = 3.0 V V_{IN} = 2.8 V / 0.2 V fs = 32.768 kHz	–	30	60	μA
Supply Current in SLEEP mode				–	15	30	
Supply Current in STOP mode			V_{DD} = 5.5 V V_{IN} = 5.3 V / 0.2 V	–	0.5	10	

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

Note 2: Input Current I_{IN1}, I_{IN3} ; The current through resistor is not included, when the input resistor (pull-up/pull-down) is contained.

Note 3: Input Current I_{IN4} ; The current when the pull-down register (Rk) is not connected by the mask option.

AD Conversion Characteristics

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

PARAMETER	SYMBOL	CONDITIONS	Min	Typ.	Max	UNIT
Analog Reference Voltage	V_{AREF}		4.5	–	V_{DD}	V
	V_{ASS}		V_{SS}			
Analog Input Voltage	V_{AIN}		V_{ASS}	–	V_{AREF}	V
Analog Supply Current	I_{REF}	V_{AREF} = 5.5 V, V_{ASS} = 0.0 V	–	0.5	1.0	mA
Nonlinearity Error		V_{DD} = 5.0 V, V_{SS} = 0.0 V V_{AREF} = 5.000 V V_{ASS} = 0.000 V	–	–	± 1	LSB
Zero Point Error			–	–	± 1	
Full Scale Error			–	–	± 1	
Total Error			–	–	± 2	

Note: Total errors includes all errors, except quantization error.

A.C. Characteristics

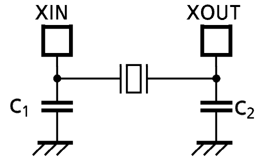
(V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = – 30 to 70°C)

PARAMETER	SYMBOL	CONDITIONS	Min	Typ.	Max	UNIT
Machine Cycle Time	tcy	In NORMAL1, 2 modes	0.32	–	10	μs
		In IDLE 1, 2 modes				
		In SLOW mode	117.6	–	133.3	
		In SLEEP mode				
High Level Clock Pulse Width	t_{WCH}	For external clock operation (XIN input), fc = 12.5 MHz	32	–	–	ns
Low Level Clock Pulse Width	t_{WCL}					
High Level Clock Pulse Width	t_{WSH}	For external clock operation (XTIN input), fs = 32.768 kHz	15.2	–	–	μs
Low Level Clock Pulse Width	t_{WSL}					

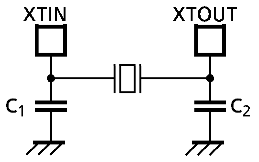
Recommended Oscillating Conditions

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

PARAMETER	Oscillator	Oscillation Frequency	Recommended Oscillator		Recommended Constant	
					C_1	C_2
High-frequency Oscillation	Ceramic Resonator	12.5 MHz	Murata	CSA12.5MTZ	30 pF	30 pF
		8 MHz	Murata	CSA8.00MTZ	30 pF	30 pF
	Crystal Oscillator	12.5 MHz	NDK	AT-51	10 pF	10 pF
Low-frequency Oscillation	Crystal Oscillator	32.768 kHz	NDK	MX-38T	15 pF	15 pF



(1) High-frequency Oscillation



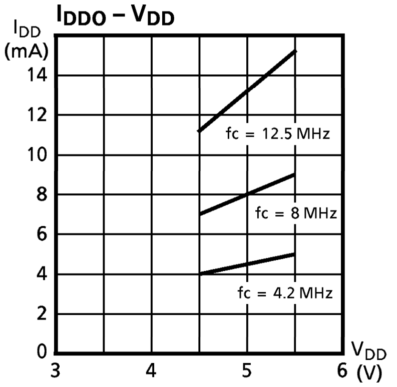
(2) Low-frequency Oscillation

Note: An electrical shield by metal shield plate on the surface of IC package should be recommendable in order to prevent the device from the high electric fieldstress applied from CRT (Cathode Ray Tube) for continuous reliable operation.

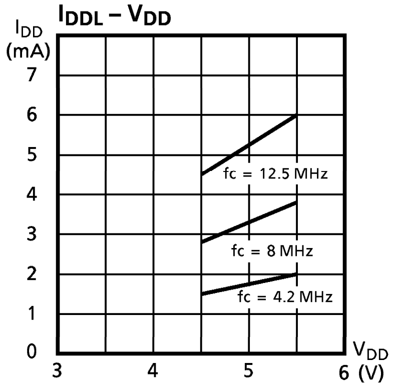
Typical Characteristics

(Ta = 25°C)

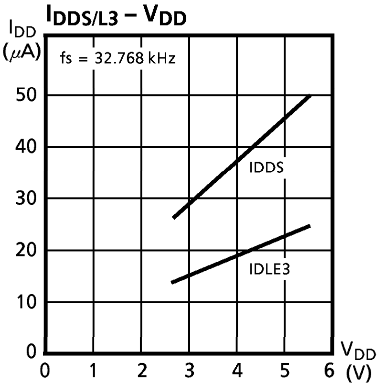
I_{DDO} – V_{DD}



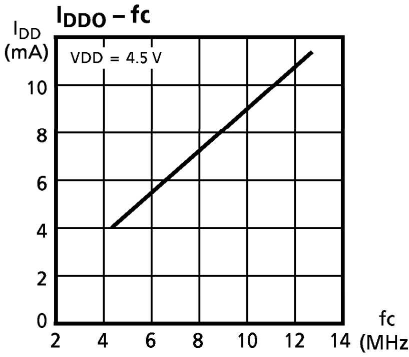
I_{DDL} – V_{DD}



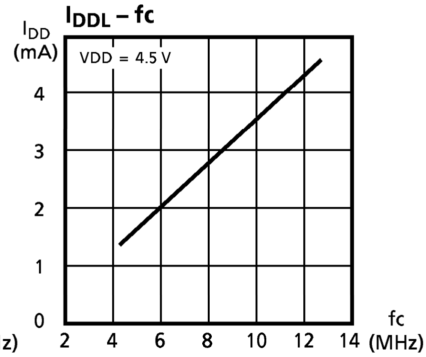
$I_{DDS/L3}$ – V_{DD}



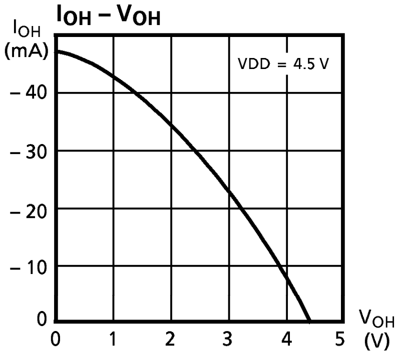
I_{DDO} – fc



I_{DDL} – fc

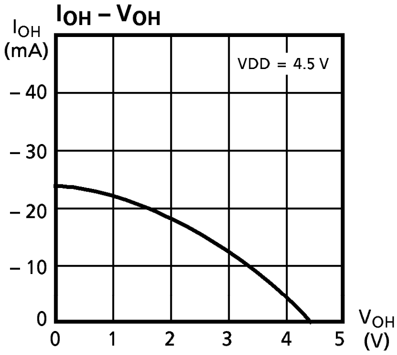


I_{OH} – V_{OH}



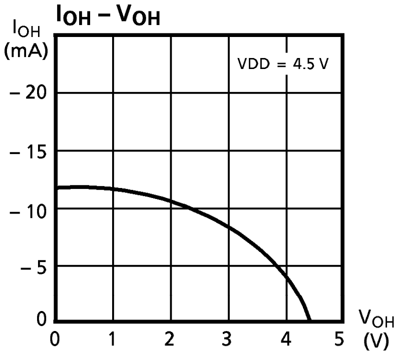
P6, P7, P80, P81 ports

I_{OH} – V_{OH}



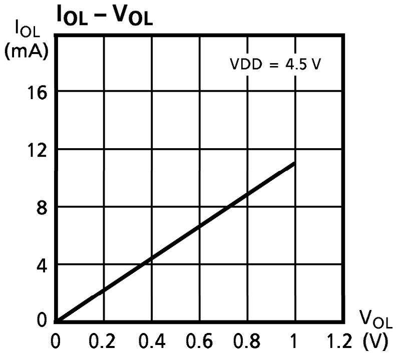
P82 to P87, P9, PD, PE, PF ports

I_{OH} – V_{OH}



P1, P3, P4, P5 ports

I_{OL} – V_{OL}



P0, P1, P2, P3, P4, P5 ports